Transit Oriented Communities Tool

Final Report

Miami-Dade Transportation Planning Organization

June 2021

RENAISSANCE PLANNING

TPO Miami-Dade Transportation Planning Organization

Contents

TABLE OF FIGURES

Table of Tables

Executive Summary

Executive Summary

The Transit Oriented Communities Tool (TOC Tool) is an accessible resource devoted to transit oriented communities (TOC) in Miami-Dade County, specifically in corridors and station areas defined by the Strategic Miami Area Rapid Transit (SMART) Plan. The TOC Tool offers introductory information to help site visitors understand TOC, learn about design conventions in station area walksheds, and become acquainted with key topics for measuring and monitoring TOC implementation progress. The key topics reported in the TOC Tool are Economic Development, Urban Form, Regional Travel, and Local Access.

Within each reporting topic, elements of effective TOC are defined as follows:

TOC is effective when....	
Economic Development • regional growth is directed to SMART Plan corridors and station areas	**Regional Travel** • multimodal travel is increasing • the transportation system provides access to jobs and essential services by a variety of modes • SMART Plan corridors and station areas generate fewer and shorter vehicular trips than the rest of the region
Urban Form • development patterns promote multimodal travel • expansion of the region's development footprint is modest and consistent with local and regional plans	**Local Access** • SMART Plan corridors and station areas offer opportunities for work, shopping, school, and recreational trips by non-motorized modes • SMART Plan stations are reachable by walking from most parcels in the station areas • parks and other public spaces are reachable by walking from most parcels in the station areas

For each element of effective TOC, the TOC Tool offers interactive dashboards with web maps, data summaries, and graphic visualizations to track TOC implementation progress over time. Users can select reporting areas from drop-down menus in the dashboards, specifying corridors and station areas. The selection results are combined to filter the dashboard contents and focus reporting on the selected location. The options allow users to explore metrics for individual station areas, entire corridors, or the entire SMART Plan system.

Finally, the TOC Tool provides additional information pertaining to TOC planning in Miami-Dade County, links to relevant related planning efforts in the region, and a review of academic and industry literature pertaining to TOC and similar development paradigms.

Project Working Group

The Miami-Dade Transportation Planning Organization (TPO) developed the TOC Tool in consultation with a Project Working Group (PWG), consisting of transportation and land use planners representing partner agencies and local jurisdictions in SMART Plan corridors. The PWG reviewed documents informing and framing the development of the TOC Tool; participated in regular meetings to facilitate data acquisition and review potential TOC-related metrics; preview site organization and content; and discuss prospective applications of the TOC Tool for SMART Plan implementation.

Summary of Report Contents

The ultimate product of the TOC Tool development effort is the TOC Tool website[1], which reports current conditions and trends for a variety of TOC-related metrics across the SMART Plan system by corridor and station area. This document compiles resources that describe TOC and its role in Miami-Dade County, the use of the TOC Tool website, and detailed background information about its development. A brief overview of contents is offered below.

Transit Oriented Communities

This introductory section describes what TOC is and some key topics for measuring and monitoring progress toward TOC goals and objectives. It describes TOC's role in Miami-Dade County's long-term strategic land use and transportation plans. Finally, it summarizes TOC-related literature from applications in other regions and academic studies. These resources provide a primer for orienting the reader to TOC and the contents of the TOC Tool website. The remainder of the document focuses on the use and development of the TOC Tool website.

TOC Tool User Guide

The TOC Tool User Guide (User Guide) describes how the TOC Tool website is organized and how users can most effectively use the site. It begins with an introduction to the TOC Tool and describes the details of the site organization and navigation options. This section includes an outline of the four primary topic areas for TOC reporting: Economic Development, Urban Form, Regional Travel, and Local Access. Within each topic, summaries of the reporting dashboard(s) supporting each are provided. The User Guide then offers an overview of the standard TOC dashboard organization with tips for dashboard navigation and interpretation.

TOC Tool Technical Guide

The TOC Tool Technical Guide (Technical Guide) describes how the TOC Tool website is constructed and maintained. This includes an overview of the scripts used to obtain, process, and generate TOC Tool data and metrics; the TOC data processing organization; and the TOC Tool resources in ArcGIS, including notes on configuration, updating, and reproduction. The Technical Guide also includes detailed appendix materials covering the methods and data sources used to formulate TOC metrics, comprehensive documentation of all TOC Tool scripting, and a proposed schedule and pattern for maintaining and augmenting the TOC Tool.

Attachments

In addition to the User Guide and Technical Guide documents described above, this document includes two interim reports that shaped the development of the final TOC Tool. These reports use some terminology that has changed since the early phases of the TOC Tool's development. They also use their own respective numbering system for figures and tables.

- **Literature Review**: This report lays the foundation for understanding key concepts in transit oriented development (TOD) and community perspectives through a review of TOD and TOC reporting

[1] https://www.miamidade.gov/global/TPO/transit-oriented-communities/home.page

efforts in peer regions across the country. It also describes the relationship between TOD and current planning priorities expressed in key plans in Miami Dade County.

- **Initial Framework**: This report outlines the objectives of the TOC Tool in supporting the SMART Plan and identifies the elements of effective TOC/TOD based on those objectives and the literature review. These elements defined the organization of the TOC Tool data, reporting, and interface development. The framework report also defines the temporal and spatial reporting scope of the TOC reporting. Some potential metrics included in the initial framework report were revised or dropped in the final TOC Tool due to data, processing, or other constraints. While some of the language and content of this report evolved over the course of developing the TOC Tool, this report is included to record the framing principles, intent and aspirations for potential updates and enhancements to the TOC Tool.

Disclaimer

The Miami-Dade TPO is not endorsing any manufacturers, products, or services cited herein and any trade name that may appear in the work has been included only because it is essential to the contents of the work.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the Author(s) and do not necessarily reflect the view of the Miami-Dade TPO.

Transit Oriented Communities

Transit Oriented Communities (TOC) are a major focus of Miami-Dade County's regional growth management and transportation strategy. TOCs focus new residential, commercial, office and institutional investments in areas served by premium transit. The TOC Tool website provides helpful information on TOC development in the county across four key topics:

- Economic Development
- Urban Form
- Regional Travel
- Local Access

This report offers a brief overview of what TOC is, describes how it aligns with ongoing major planning efforts in Miami-Dade County, and summarizes literature pertaining to TOC and related development approaches, including transit oriented development (TOD).

What Is TOC?

Transit Oriented Communities (TOC) are neighborhoods and business districts designed around rapid transit stations. TOC integrates economic development and urban design principles with regional mobility via rapid transit. TOC is characterized by medium-to-high density, mixed-use development and a well-connected network of streets providing local connectivity for pedestrians and bicyclists. These elements combine to create thriving communities supporting non-motorized travel, providing connected streets enhancing automobile travel and stimulating development that is easily connected via transit.

A growing number of regions around the country are implementing TOC approaches because they support both concentrated growth and development in compact areas and enhanced transit and non-motorized connections. This results in benefits like:

- Increased access to jobs and essential services like health care, fresh food and schools
- Increased opportunities for shorter trips which may reduce vehicle miles traveled, potentially mitigating congestion
- Maximized impacts of public investments in rapid transit
- Improved conditions for recreational and non-work walking and biking
- Increased property values and reinvestment in the community

TOC promotes development that generates these and other benefits while establishing a strong foundation for rapid transit investments, even accounting for the uncertainty inherent in large-scale infrastructure projects. TOC affirms sound planning and urban design principles, guiding efficient and sustainable growth and development supporting community goals.

TOC supports concentrated areas of development that are effectively and efficiently served by useful transit service. TOC can accommodate anything from existing fixed route local bus service and is flexible enough to evolve, accommodating future rapid transit technologies. TOC fosters land use and trip-making patterns that generate the ridership needed for successful rapid transit projects, laying the groundwork for cost-effective projects.

Characteristics of TOC

Effective TOC emphasizes both regional mobility and local accessibility, providing travel choices, supporting regional economic development goals and contributing to local livability goals. TOC primarily focuses on the density and mix of development near transit stations, relative to local context. Zoning for TOC promotes mixed uses within a neighborhood (horizontal mixed use) and within individual buildings, such as a multifamily residential building with retail on the ground floor (vertical mixed use). Walking and biking connections are prioritized, and regional transit connections are accessible. This results in more destinations (jobs, stores, banks, restaurants, etc.) and housing units in a connected and compact area.

Being near something does not always mean it is easily accessed. Design and urban form are critical elements of TOC, ensuring that houses and destinations are well-connected to each other and to transit services. In a thoughtfully designed TOC area, the strong connectivity between land uses and a transit station promotes increased local walking and biking trips and more transit trips for longer distance travel. In areas with "transit adjacent" development, travel options are often limited.

In Figure 1, development in the upper right example is connected by a compact street network and the transit station is easily accessed. This is well-designed TOC. Development in the lower left example is characterized by large lots, circuitous connections over a sparse network of streets and a large parking lot separating jobs and residents from the transit station. This type of development may be near transit, but it is not effective TOC.

Figure 1 - Transit-Oriented vs. Transit-Adjacent Development Patterns[2]

[2] Source: National Transit Institute https://www.ntionline.com/transit-oriented-development/

Defining the Station Area

TOC is implemented over an entire transit station area rather than on a site-by-site basis. A station area is typically defined as the land area within a specified distance of a given transit station (usually one-quarter mile or one-half mile). More complex transit station area definitions may account for walking and/or bicycling times to the transit station, barriers to accessing the station or other factors affecting the ease with which the transit station can be reached from nearby developments.

The TOC Tool defines station areas as the half-mile radius around each station, except where alternative station area boundaries have been set through local planning processes. Locations within the half-mile radius of multiple stations are included for all station areas they fall within. Although the station area is defined in this relatively simplistic way, walking and bicycling access metrics for each station support a more realistic analysis of "on-the-ground" accessibility, helping understand how easy it is to get to stations on foot or by bike.

Aspects of TOC

Like all land development processes, TOC evolves gradually over time. No two station areas are the same, nor will they evolve in the same way. Each station area has a distinct context and vision for future development; differing levels of market readiness; and varying relationships to the regional rapid transit network. Monitoring development trends and community characteristics in station areas over time helps localities assess the effectiveness of their adopted TOC implementation strategies, make local policy adjustments and coordinate with regional partners and/or neighboring jurisdictions to manage growth and support local and regional planning priorities. It is important to identify a clear set of TOC goals, objectives and supporting metrics to monitor TOC trends over time.

The TOC Tool focuses on four key aspects of TOC:

- **Economic Development**: There must be development demand in TOC areas, aligning with regional economic development goals. Established TOC areas should provide a mix of jobs, while emerging TOC areas should offer developable land in accessible locations. Successful TOCs present attractive and resilient investment opportunities that sustain property value appreciation over time. The TOC Tool provides data about development trends in SMART Plan corridors and station areas.
- **Built Environment**: TOCs offer land development patterns that support multimodal transportation investments, creating stations areas with a vibrant mix of uses at transit-supportive densities with a well-connected network of local streets. It is important to note that transit-supportive densities vary with community context and transit technology. The TOC Tool provides data about development density, land use mix, and street connectivity in SMART Plan station areas and corridors.
- **Regional Travel**: The SMART Plan envisions investment in six new rapid transit corridors in addition to Miami-Dade's existing Metrorail and Metromover lines. The regional rapid transit network will provide access to jobs, shopping, schools, and services by transit from station areas. TOCs will bolster the ridership potential in each corridor and manage growth in vehicle miles of travel (VMT) across the region. The TOC Tool tracks regional accessibility, ridership, VMT, average trip lengths, and related statistics describing regional travel in SMART Plan corridors and station areas.
- **Local Accessibility**: The well-connected, compact development patterns associated with TOC present opportunities to make short trips by walking and biking for work, shopping, school, and recreation. Local access to key destinations complements the regional travel impacts of TOCs. The TOC Tool provides data about walking and biking access to jobs and key destinations, parks and

public spaces, last-mile access to SMART Plan stations, and bicycle and pedestrian facilities that make local travel safe and comfortable.

The Role of TOC in Miami-Dade County

TOC is an important component of the Miami-Dade region's growth management strategy. Several key planning documents emphasize development paradigms accommodating continued regional growth supported by adequate urban infrastructure, including multimodal transportation facilities.

These include Miami-Dade County's Comprehensive Development Master Plan[3] (CDMP); Miami-Dade TPO's 2045 Long Range Transportation Plan[4] (LRTP); and the Strategic Miami Area Rapid Transit[5] (SMART) Plan.

Comprehensive Development Master Plan

The planning principles set forth by the CDMP, and supported by regulations in the County code, present a clear vision of development concentrated in urban centers and defined by multimodal accessibility (provided by a variety of transportation modes such as walking, biking and transit) in addition to maintaining vehicular mobility. Designations such as the Urban Development Boundary and Urban Centers concentrate future development in multimodal areas and encourage infill development and redevelopment in areas that are already urbanized.

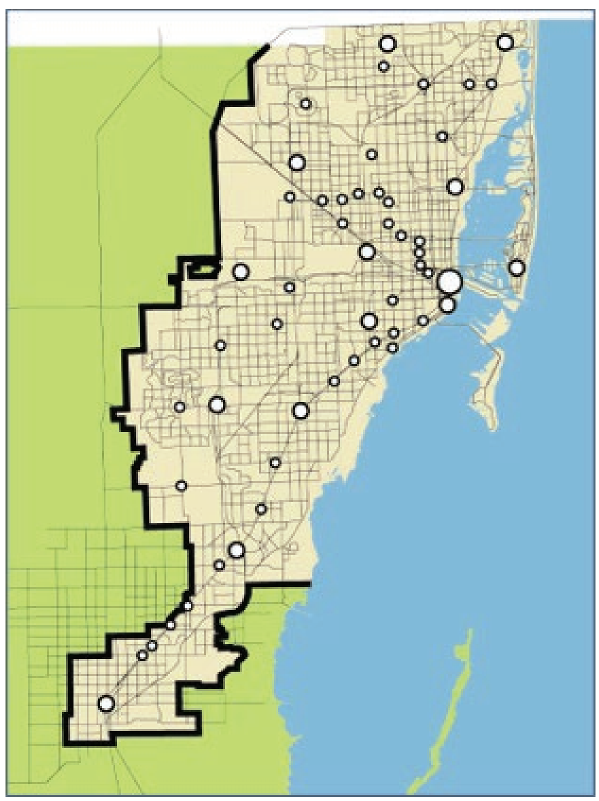


Figure 2 - The CDMP aims to manage regional growth by focusing development in Urban Centers

---

[3] https://www.miamidade.gov/planning/cdmp.asp
[4] https://www.metro.net/projects/lrtp/
[5] http://www.miamidadetpo.org/smartplan.asp

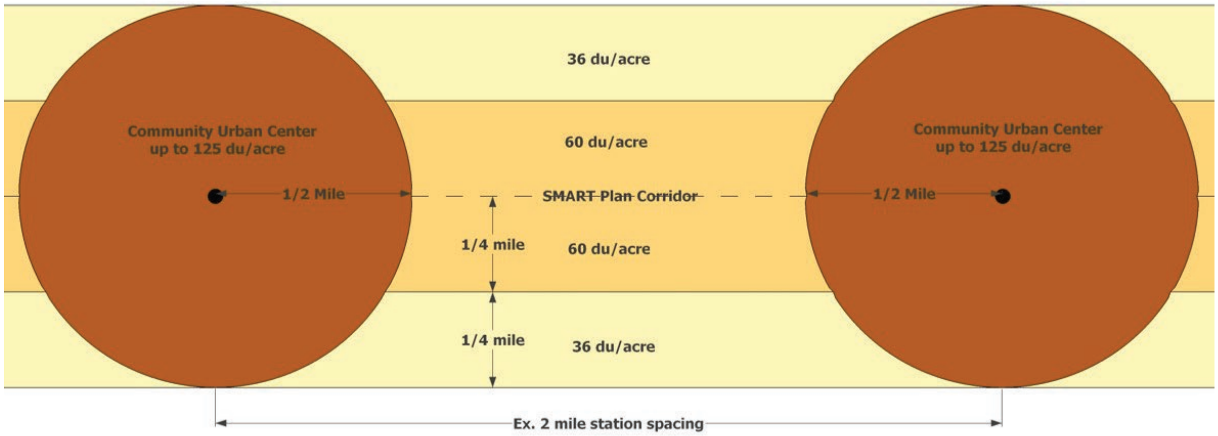


Figure 3 - The CDMP establishes station-area and corridor density targets for transit-supportive development

Long Range Transportation Plan

The goals established for the future of the transportation network in Miami-Dade TPO's 2045 LRTP align with the multimodal vision for urban development set forth in the CDMP. While the LRTP is a comprehensive regional transportation plan establishing needed projects and funding priorities for all modes, it places a heavy emphasis on transit and the implementation of the SMART Plan stating, "The highest priorities of the TPO Governing Board are transit and projects that support transit services."

Examples of goals and objectives listed in the LRTP that relate to TOC include:

- GOAL: Maximize Mobility Choices Systemwide
    - Objective: Provide a comprehensive transportation network for dependable and reliable transportation options.
    - Objective: Increase mobility choices throughout the county.
- GOAL: Support Economic Vitality
    - Objective: Provide affordable housing.
    - Objective: Improve access to employment centers.
    - Objective: Provide access to tourist destinations - seaports, airport, beaches, etc.
- GOAL: Protect and Preserve the Environment and Quality of Life and Promote Energy Conservation
    - Objective: Preserve agricultural land or critical habitat consumed by transportation projects.
    - Objective: Minimize and mitigate air and water quality impacts of transportation facilities, services, and operations.
    - Objective: Promote projects that support urban infill and densification.

      major activity centers (i.e., healthcare, recreation, education, employment, and cultural facilities).
    - Objective: Improve Quality of Life for all ages and abilities.

These excerpts from the LRTP resonate with many of the key elements of successful TOC, reinforcing the importance of TOC in the region's coordinated growth management and transportation planning strategy.

They also clarify topics that warrant monitoring as the SMART Plan is implemented and TOC matures over time to understand how TOC is helping the region achieve these and similar goals and objectives.

SMART Plan

The SMART Plan identifies six rapid transit corridors extending the region's existing rapid transit network. Figure 4 below shows a map of each corridor's extents and relationship to the existing rapid transit network. All six SMART Plan corridors and additional Bus Express Rapid Transit (BERT) corridors are located within the Urban Development Boundary and many of the zoned Urban Centers are located along SMART Plan corridors. The areas targeted by the County for concentrated growth are planned to be along premium transit routes, exemplifying the integration of land use and transportation planning that is essential to TOC.

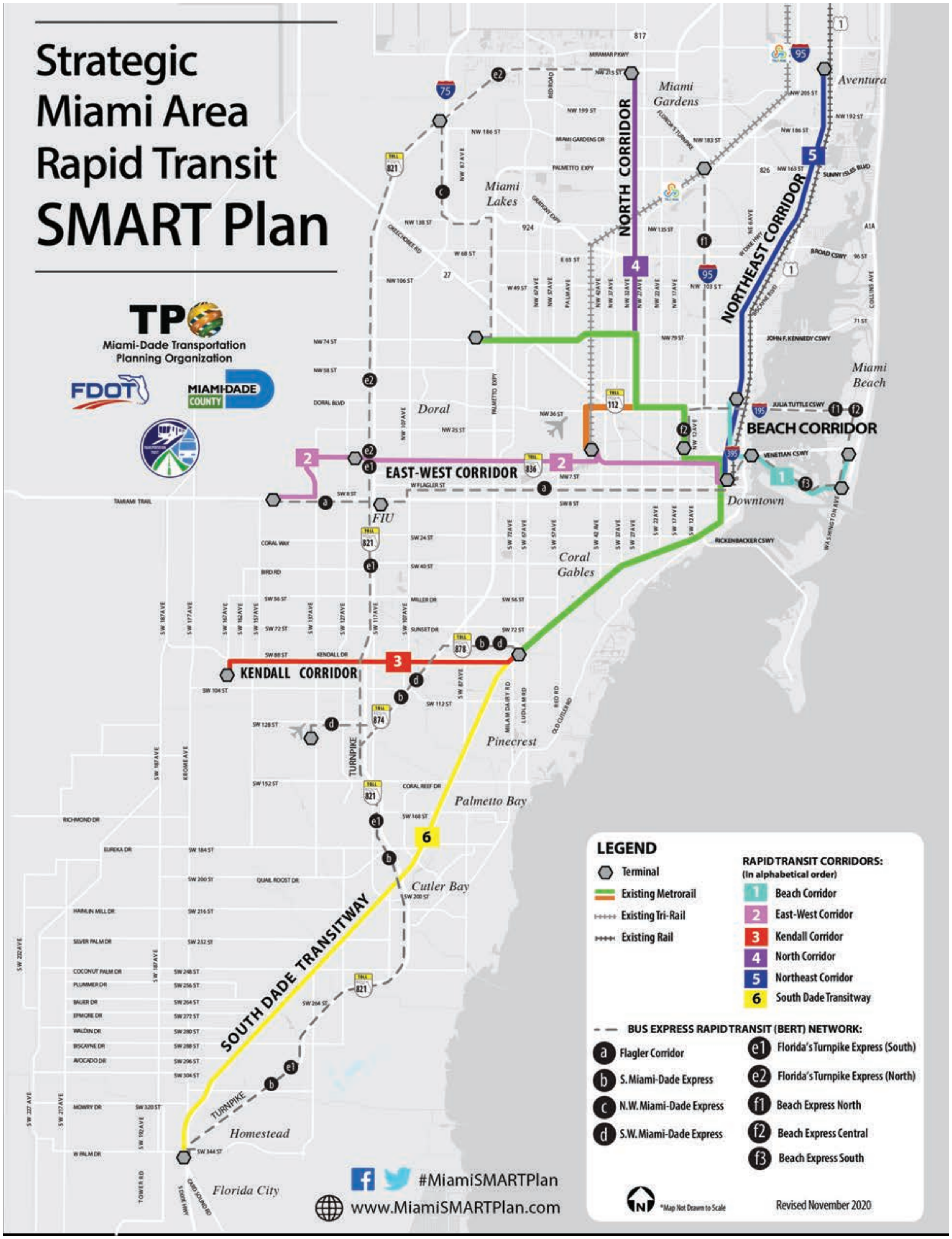


Figure 4 - SMART Plan: Map of six planned rapid transit corridors

Current SMART Plan efforts are focused on studying transit technologies, operations, costs, ridership, environmental impacts, etc. for each of the six corridors. This includes visioning studies and Project Development and Environmental (PD&E) studies that address key questions of land use planning, transportation system engineering, environmental impacts, community visions for growth and development and more.

Literature Pertaining to Transit Oriented Communities

This section summarizes several studies and examples of TOC (or similar) applications that offer additional orientation to this development and planning paradigm.

Los Angeles TOC Guidelines[6]

Los Angeles' Transit Oriented Communities (TOC) Incentive Program was passed by voters in 2016. The program contains two key provisions – 1) A package of new incentives for building affordable housing near public transit; and 2) a requirement that Los Angeles City Planning create TOC guidelines for all housing developments within a half-mile radius of a major transit stop. Los Angeles's program is specifically focused on stimulating the development of affordable housing near transit and less focused on cultivating an overall urban form that better supports transit. Therefore, the LA program does not promote recommendations for urban design or policy guidelines addressing the multiple facets of successful transit oriented urban form.

Los Angeles's TOC Programs[7]

In recent years, municipalities throughout California have struggled to meet housing needs, and construction of new housing units in the state has not kept pace with demand, resulting in increased housing costs that rank among the highest in the nation. At the same time, California faces pressure to achieve ambitious greenhouse gas (GHG) reduction goals in the relatively near term. Meeting those goals will require significant decreases in transportation sector emissions, which represent about 40 percent of the state's GHG emissions. Particularly impacted by both the affordability and climate change crises are low-income Californians, whose communities suffer disproportionate impacts from lack of housing availability and vulnerability to climate change—and who also are California's most reliable transit riders.

Developing TOC for Better Accessibility and Affordability[8]

The development of transit-oriented communities (TOC) is a central element in the promotion of accessibility in the Metro Vancouver Region (MVR). The entity is building on its wealth of experience in public transport development in the region, which has contributed to its high levels of well-being and economic progress. TOC aims to incentivize people to drive less and walk, cycle and take transit more. A solid culture of community engagement, the existence of a coordinating body for transport planning, and the links between transport and land-use policy are the main assets MVR has to enhance accessibility. However, tackling the affordable housing deficit around transport hubs remains a challenge for local authorities. The purpose of this paper is to draw lessons from the MVR's experience in transit-oriented initiatives to contribute to the development of compact, connected and green urban centers.

[6] https://planning.lacity.org/plans-policies/transit-oriented-communities-incentive-program

[7] https://escholarship.org/uc/item/5x83x4w6

[8] https://www.oecd-ilibrary.org/urban-rural-and-regional-development/developing-transit-oriented-communities-for-better-accessibility-and-affordability_f2bb60fc-en;jsessionid=7DG49JvBAEYXK0hriNrideqe.ip-10-240-5-142

Does TOD Need the T?[9]

Transit-oriented developments (TODs) often consist of new housing near rail stations. Channeling urban growth into such developments is intended in part to reduce the climate change, pollution and congestion caused by driving. But new housing might be expected to attract more affluent households that drive more, and rail access might have smaller effects on auto ownership and use than housing tenure and size, parking availability, and the neighborhood and sub-regional built environments.

Cost of a Ride: The Effects of Densities on Fixed-Guideway Transit Ridership and Costs[10]

Seeking to identify cost-effective transit investments and density thresholds for station areas, researchers explore relationships between transit ridership, transit investments, population and employment densities, and costs. After controlling for a variety of factors, researchers identify an inverse relationship between density and capital costs. They develop density thresholds to guide planners.

Do TODs Make a Difference?[11]

The authors of this report seek to identify how TODs perform relative to the metropolitan area that they are situated within. To this end, researchers reviewed 23 fixed guideway transit systems in 17 metropolitan areas in the South and West. They find that TODs do perform differently from the metropolitan area.

Building Transit Oriented Development in Established Communities[12]

The report identifies several approaches, such as the use of transit oriented design, focusing transit oriented development (TOD) around park-and-ride lots, making changes to land development regulations, parking management, offering development incentives, coordinating stakeholders, incorporating transit into future development/redevelopment, crafting TOD design guidelines, predesignating transit corridors, ensuring pedestrian and bicycle access, adapting transit services to the needs of suburban-style communities, offering location efficient mortgages and ideas for dealing with community resistance toward applying transit friendly measures to car oriented communities.

Incentivizing TOD: Case Studies of Regional Programs Throughout the United States[13]

With a goal of reviewing diverse approaches to the implementation of TOD, this report is comprised of case studies from 1994 to 2006. Each case study reviews a variety of topic areas, including the sponsoring organization, the TOD program that created it, activities around creating the TOD, funding, and lessons learned.

Histories of Transit-Oriented Development[14]

This paper looks at the development of the TOD vision from multiple perspectives. First, the paper looks at a long history of transit-served real estate development. Second, it evaluates TOD's position in a long history of social and urban theory. Lastly, it tracks Calthorpe's TOD concept through his lifetime of work up to the publication of "The New American Metropolis." This paper is meant to serve as a foundation for further work

---

[9] https://www.tandfonline.com/doi/abs/10.1080/01944363.2013.791008
[10] https://www.tandfonline.com/doi/abs/10.1080/01944363.2011.589767
[11] https://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1006&context=trec_reports
[12] https://www.nctr.usf.edu/pdf/473-135.pdf
[13] https://www.psrc.org/sites/default/files/incentivizingtodcasestudies.pdf
[14] https://escholarship.org/uc/item/7wm9t8r6

that will seek to define the term TOD in its many uses across the fields of planning, design, transportation, and policy.

TOC Tool User Guide

Overview

This document provides a user guide to the TOC Tool website, orienting users to the site's organization and illustrating how to navigate the TOC Tool dashboards and web maps.

The Transit Oriented Communities Tool (TOC Tool) is an accessible resource devoted to transit oriented communities (TOC) in Miami-Dade County, specifically in corridors and station areas defined by the Strategic Miami Area Rapid Transit (SMART) Plan. The TOC Tool offers introductory information to help site visitors understand TOC, learn about design conventions in station area walksheds, and become acquainted with key topics for measuring and monitoring TOC implementation progress. The key topics reported in the TOC Tool are Economic Development, Urban Form, Regional Travel, and Local Access.

Within each reporting topic, elements of effective TOC are defined as follows:

TOC IS EFFECTIVE WHEN....	
Economic Development • regional growth is directed to SMART Plan corridors and station areas	**Regional Travel** • multimodal travel is increasing • the transportation system provides access to jobs and essential services by a variety of modes • SMART Plan corridors and station areas generate fewer and shorter vehicular trips than the rest of the region
Urban Form • development patterns promote multimodal travel • expansion of the region's development footprint is modest and consistent with local and regional plans	**Local Access** • SMART Plan corridors and station areas offer opportunities for work, shopping, school, and recreational trips by non-motorized modes • SMART Plan stations are reachable by walking from most parcels in the station areas • Parks and other public spaces are reachable by walking from most parcels in the station areas

For each element of effective TOC, the TOC Tool offers interactive dashboards with web maps, data summaries, and graphic visualizations to track TOC implementation progress over time. Users can select reporting areas from drop-down menus in the dashboards, specifying corridors and station areas. The selection results are combined to filter the dashboard contents and focus reporting on the selected location. The options allow users to explore metrics for individual station areas, entire corridors, or the entire SMART Plan system.

Finally, the TOC Tool provides additional information pertaining to TOC planning in Miami-Dade County, links to relevant related planning efforts in the region, and a review of academic and industry literature pertaining to TOC and similar development paradigms.

Guide to the Site

Site Layout

The TOC Tool is divided into a series of sections consisting of multiple pages. A structural outline and brief description of each page is included below, with dashboard pages indicated by **blue** text.

- Home Page
  - The default page when first visiting the site. Covers the purpose of the tool, tool layout and a dashboard displaying the location and status of the SMART corridors and stations in the region.
- TOC Basics
  - Provides a brief overview of the TOC concept, including station areas and describes how transportation and land use are integrated. Also introduces the tool's four main topics: Economic Development, Urban Form, Regional Travel, and Local Access.
- Economic Development
  - *Overview*
    - Explains the importance of monitoring economic development in the region, the relationship between economic development and TOCs, and how economic development is measured in the dashboard. Also includes links to additional resources outside the tool that address economic development.
  - *New Development*
    - The New Development dashboard covers the existing, historical, and near-term future time frames, reporting on residential units (by single family or multi-family) employment (by generalized industry sector), and floor area (by land use). See Guide to the Dashboards for dashboard navigation help.
- Urban Form
  - *Overview*
    - Explains the importance of urban form considerations in effective TOCs, and how urban form is measured in the dashboards. Also includes link to additional resources outside the tool that address urban form and resiliency.
  - *Development Patterns*
    - The Development Patterns dashboard covers the existing and historical time frames, reporting on development density (residential units per acre, jobs per acre, and floor area ratio), land use mix (jobs-housing balance, land use mix index), and network connectivity (grid density, network connectivity ratio). See Guide to the Dashboards for dashboard navigation help.
  - *Urban Footprint*
    - The Urban Footprint dashboard covers the existing and historical time frames reporting on impervious surface coverage (existing time frame only) and floor area by developed land cover category. See Guide to the Dashboards for dashboard navigation help.
- Regional Travel
  - *Overview*
    - Explains the importance of monitoring regional travel in Miami Dade County, the relationship between regional travel trends and TOCs, and how regional travel is measured in the dashboards. Also includes link to additional resources outside the tool that covers regional travel and transportation system performance.
  - *Multimodal Travel*

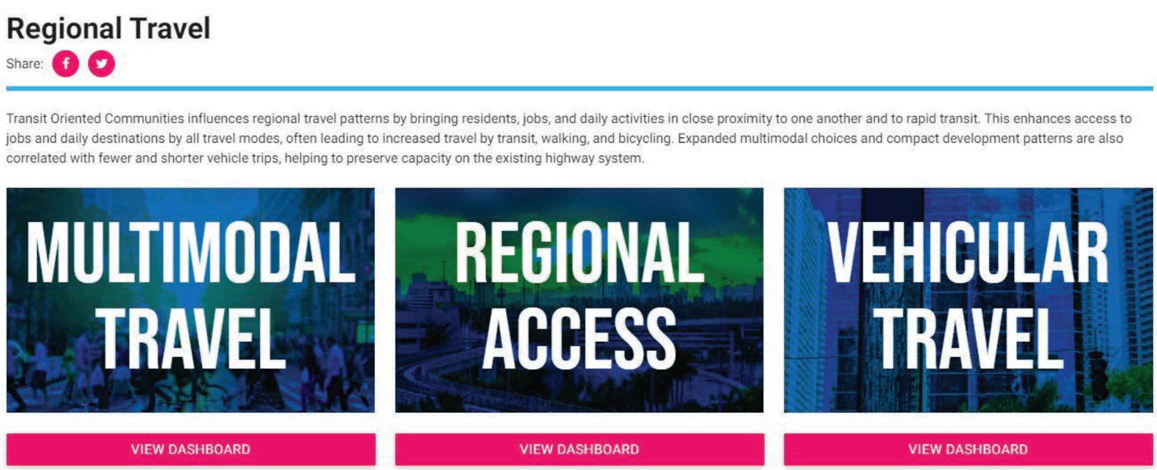
        - The Multimodal Travel dashboard covers the existing and historical time frames, reporting commute mode share and transit ridership statistics. See Guide to the Dashboards for dashboard navigation help.
    - *Regional Access*
        - The Regional Access dashboards covers the existing and historical time frames, reporting on access to jobs, housing, and essential services provided by the walk, bike, transit, and auto travel modes. See Guide to the Dashboards for dashboard navigation help.
    - *Vehicular Travel*
        - The Vehicular Travel dashboard covers the existing and historical time frames, reporting vehicle miles of travel (VMT) and average trip length for each time frame. See Guide to the Dashboards for dashboard navigation help.
- Local Access
    - *Overview*
        - Explains the importance of local access and connectivity to TOC station area design and how local access is measured in the dashboards.
    - *Non-Motorized Opportunities*
        - The Non-Motorized Opportunities dashboard covers the existing and historical time frames, reporting on bicycle facilities (existing only) and walkable and bikeable opportunities for each time frame. See Guide to the Dashboards for dashboard navigation help.
    - *Access to Transit*
        - The Access to Transit dashboard covers the existing and historical time frames, reporting on last-mile accessibility to SMART Plan stations (parcels within 15-minute walk of a station, walk time to a station, and walk directness to a station). See Guide to the Dashboards for dashboard navigation help.
    - *Access to parks*
        - The Access to Transit dashboard covers the existing and historical time frames, reporting on park acreage per 1,000 residents (existing only) and local accessibility to parks (parcels within 15-minute walk of a park, walk time to a park, and walk directness to a park). See Guide to the Dashboards for dashboard navigation help.
- Additional Reading
    - This section contains resources a site user may find useful to their overall understanding of TOC and how the TOC Tool can be used but are not necessary for effectively using the TOC Tool site.
        - *TOC in Miami*
            - Provides an overview of the history and plans for TOC in Miami Dade County. Includes links to the Comprehensive Development Master Plan, Long Range Transportation Plan, and the SMART Plan. Also links to a variety of research papers covering TOC and related development paradigms in more depth.

Site Navigation

The Home Page

TOC Tool pages are accessible from the Home page, either through direct hyperlinks or through topic overview pages. The ribbon of images on the home page (shown below) provides links to the pages indicated by the label overlaying each image.

Clicking on a TOC topic ("Regional Travel," e.g.) will take the user to that topic's "Overview" page, which provides a similar ribbon of images to link the user to the corresponding dashboard pages. In the example below (snipped from the Regional Travel Overview page), clicking on "Multimodal Travel" will load the Multimodal Travel dashboard, presenting information about commute mode shares and transit utilization in SMART Plan corridors and station areas.



Dashboard Layout

While each TOC Tool dashboard is unique and designed to best present its distinctive data, the dashboards used throughout the tool share common elements that allow for consistent, predictable, and intuitive patterns of use. The common elements of each dashboard are diagrammed below.

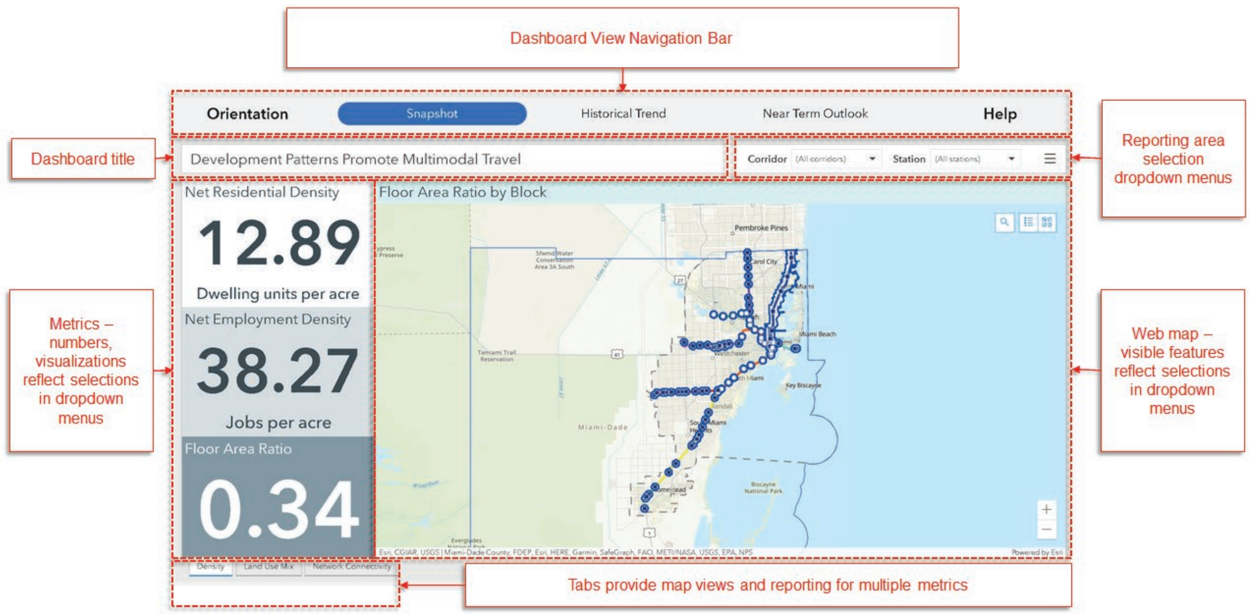


Figure 5 - Typical Organization of TOC Tool Dashboards

Dashboard View Navigation Bar

Each dashboard page includes a view navigation bar above the dashboard. The view navigation bar allows the user to see TOC-focused reporting for different time frames (usually a **snapshot** of current conditions or a **historical trend**), review orientation text describing the contents of the dashboard, or review help on how to use the dashboards. The **Help** button opens a page in a new tab whose contents are comparable to this user's guide, providing guidance on dashboard navigation. The **Orientation** button loads a popup window with descriptions of the various metrics used in the dashboard, including how they were processed/collected, where the data comes from, and what they mean in the context of TOC.

By default, the dashboard page will show a snapshot of conditions for the most recent available data year. Historical trends and short-term forecasts based on permitted development (where available) can be viewed by selecting the appropriate time frame using the bar above the dashboard. Loading a new time frame may take a few seconds, and corridor and station area selections must be made independently for each time frame.

Reporting Area Selection Dropdown Menus

Users can select reporting areas from drop-down menus in the dashboards, specifying corridors and station areas. The selection results are combined to filter the dashboard contents and focus reporting on the selected location. The options allow users to explore metrics for individual station areas, entire corridors, or the entire SMART Plan system. The dropdown menu selection options are listed below. All combinations of corridor selections and station area selections are possible.

Table 1 - Dashboard Corridor and Station Selection Menu Options

CORRIDOR SELECTION MENU	STATION AREA SELECTION MENU
• All corridors • Beach • East-West • Green • Kendall • Metromover • North • Northeast • Orange • South	• All stations *(combined station areas in selected corridor)* • Entire corridor *(area within and outside stations areas in selected corridor)* • Outside station areas *(area of selected corridor not in a station area)* • Individual stations by name *(selectable station names vary by selected corridor)*

See the Map Navigation section for graphics demonstrating how the reporting area selection menus work.

WEB MAPS

Map Essentials

Maps are provided for visual reference to highlight the location of selected SMART Plan Corridors and Station Areas and to reveal the fine-grained data details summarized in the metrics reported on the left side of the dashboard display. To understand the content shown in each map, open its legend by clicking on the icon in the upper right-hand corner of the map, as shown in the graphic below. The resulting pop up will show the legend for all active map layers.

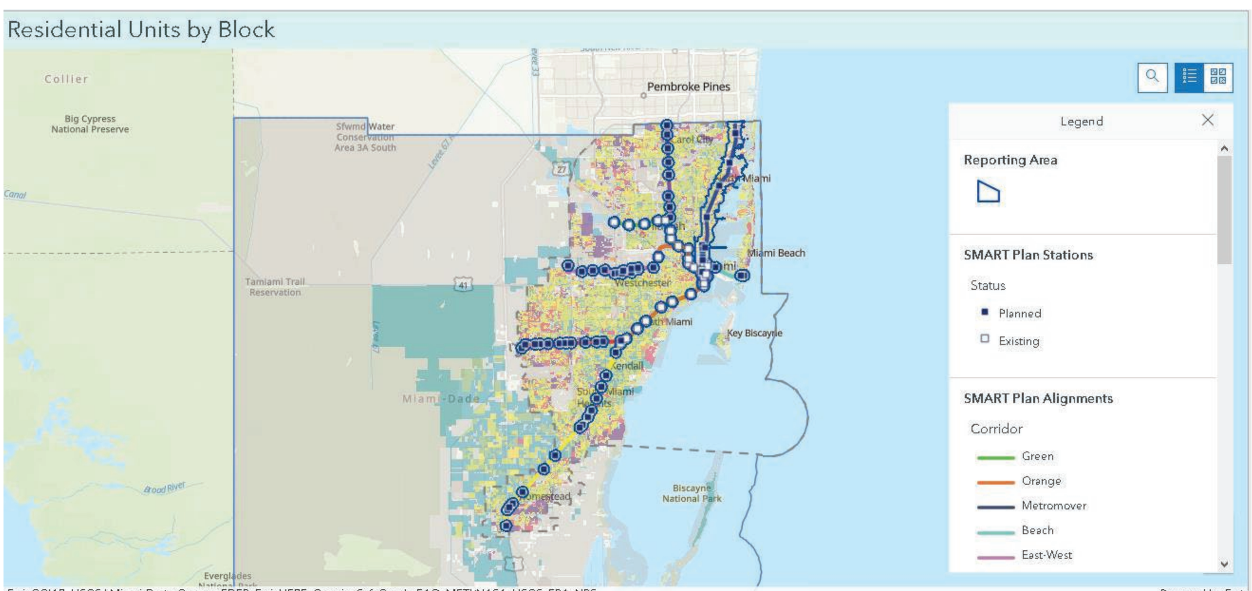


Figure 2 - Revealing the Legend in a Dashboard Webmap

The base map can be changed by clicking on the icon in the upper right-hand corner of the map, as shown in the graphic below. A variety of base maps are available including street maps, topographic maps, and aerial imagery. The default base maps is “Topographic”.

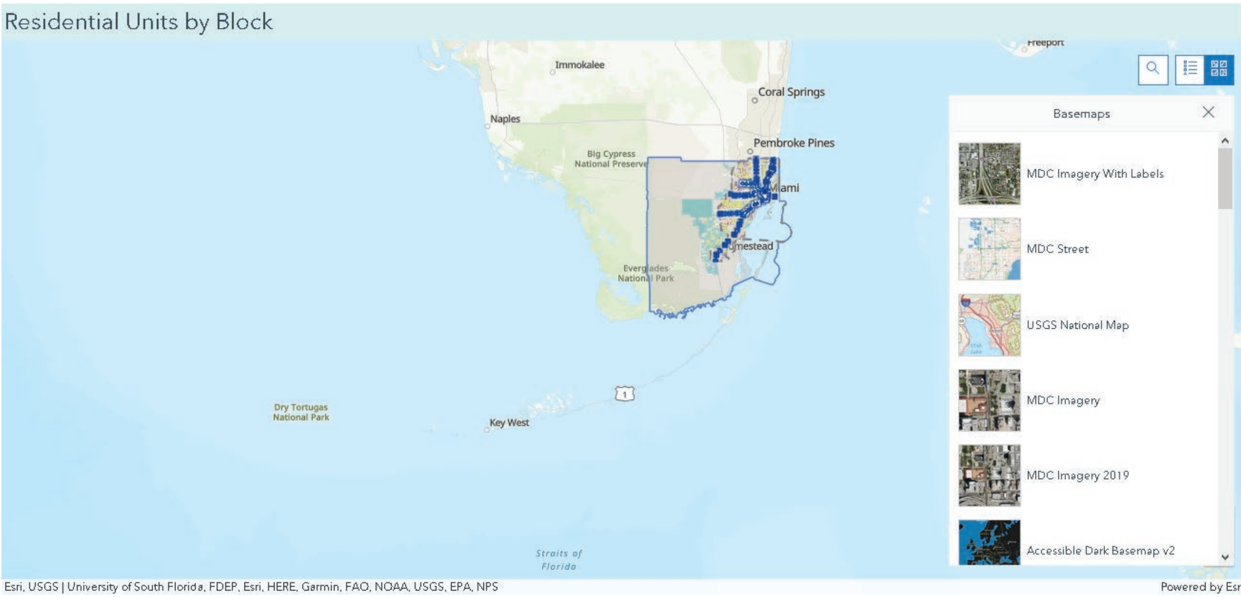


Figure 7 - Loading Alternative Basemaps in a Dashboard Webmap

Any street address can be located by clicking on the icon in the upper right-hand corner of the map to activate the address location text box. Enter an address in the text box to zoom the map extents to the respective location, as shown in the graphic below. Entering an address does not change reporting selections but can help users identify nearby corridors and stations to guide selections.

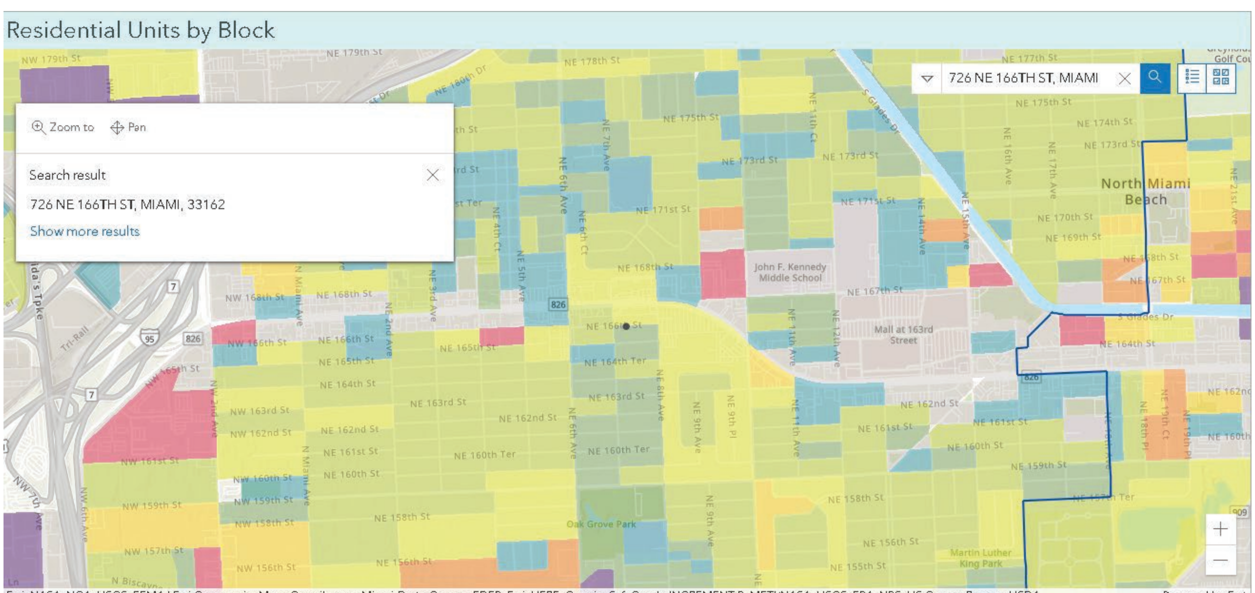


Figure 8 - Finding an Address Location on a Dashboard Webmap

Map Navigation

Each dashboard contains a dropdown menu in the upper right to select a SMART Plan corridor and view data for only the selected corridor. This will automatically zoom the map to the corridor and update the metrics reported on the left of the dashboard. If "(All Corridors)" is selected, the dashboard presents information for all SMART Plan corridors combined.

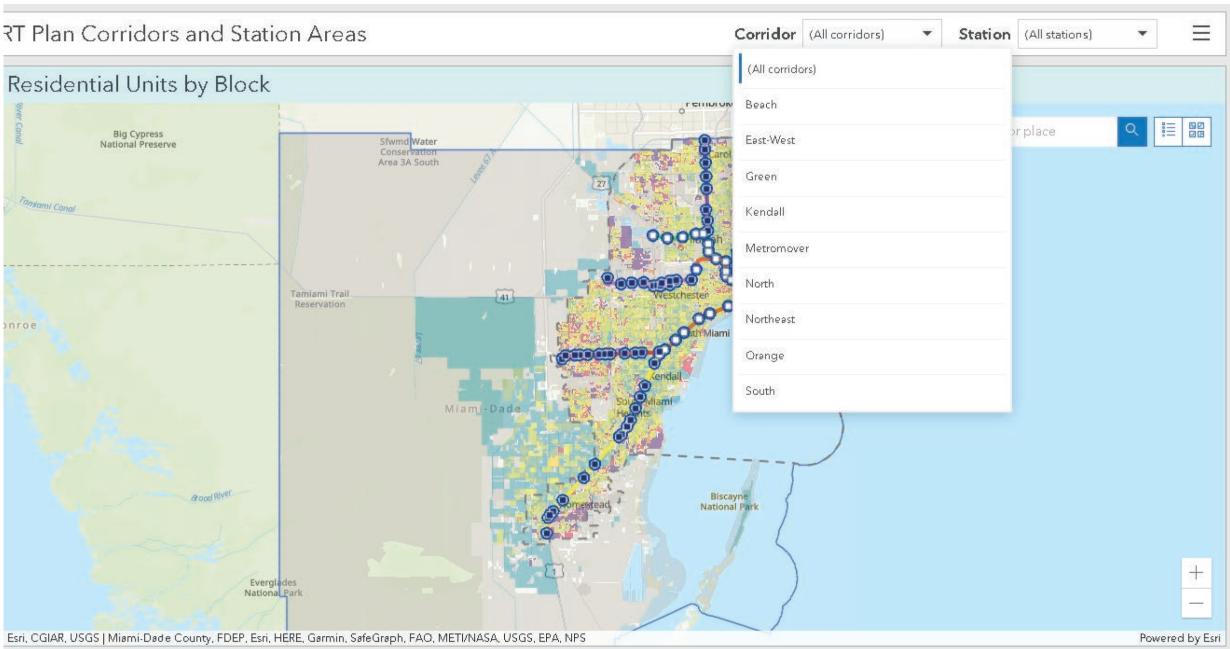


Figure 9 - Updating Dashboard Views with Corridor Selections

Each dashboard contains a second drop-down menu in the upper right to report information for a specific SMART Plan station area. The default selection is "(All stations)" within the selected corridor. In addition to specific station areas, the "(Entire corridor)" and "(Outside station areas)" options present data at a corridor-wide scale.

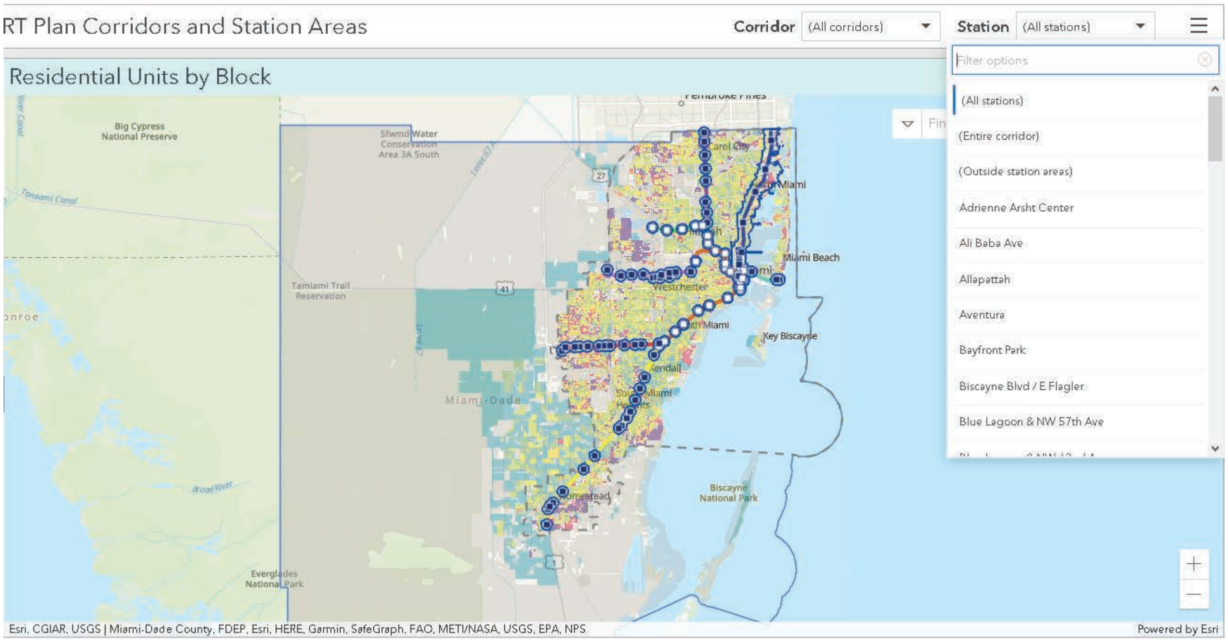


Figure 10 - Updating Dashboard Views with Station Area Selections

Metrics and Tabs

Each dashboard includes a metrics panel on the left side of the dashboard display. Numeric and graphical displays provide insight into conditions for the geographic areas set by the selections made in the reporting area selection dropdown menus.

Dashboards report multiple metrics to fully describe conditions in SMART Plan stations and corridors for a given element of effective TOC's. Browse each metric group using the tabs in the bottom left corner of the dashboard. Corridor and station area selections are retained across metric groups.

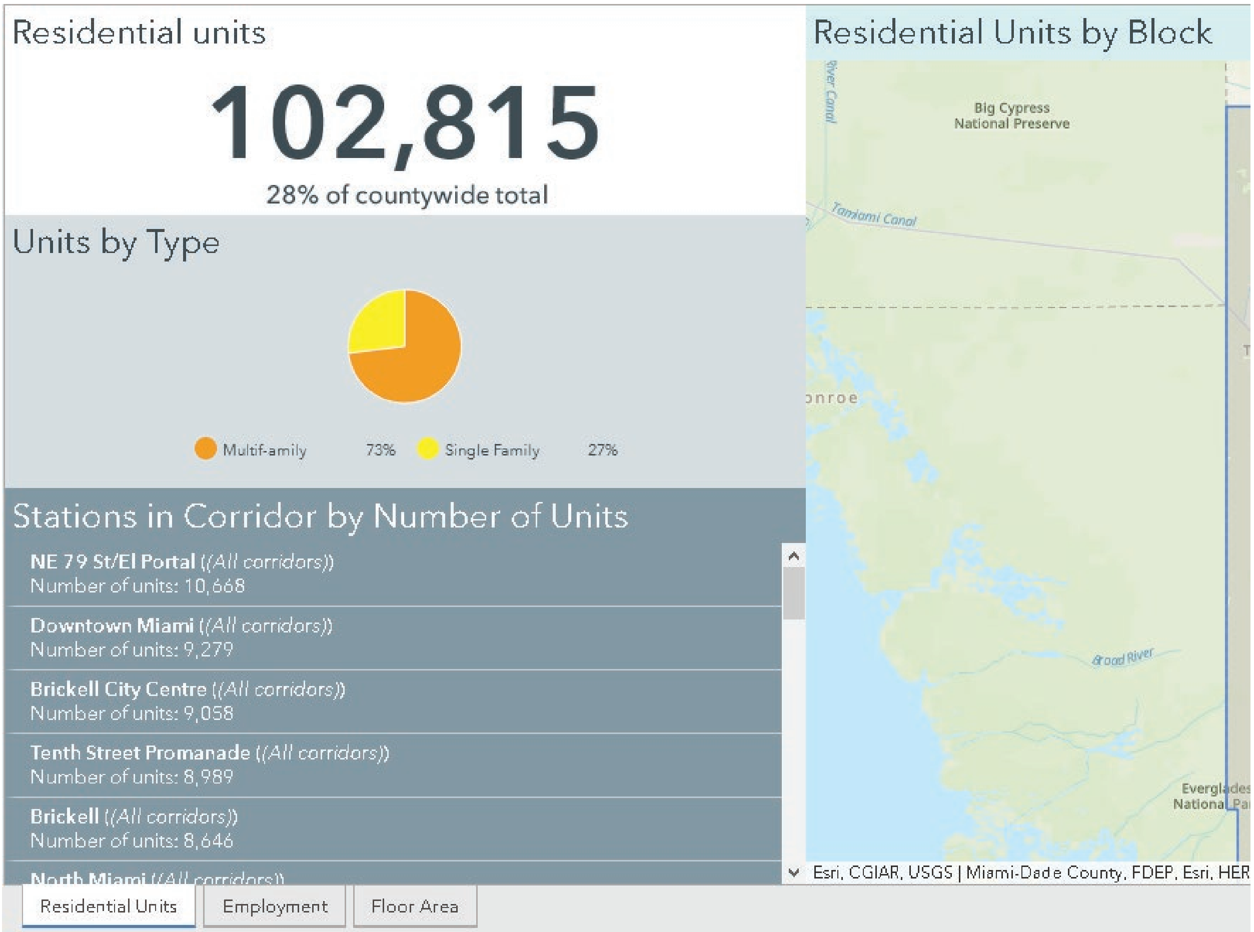


Figure 11 - Tabs Present Different Metrics in the Same TOC Element

TOC Tool Technical Guide

Overview

This document provides a technical guide to the TOC Tool website, the scripts used to develop TOC Tool metrics, the organization and construction of web resources in ArcGIS Online, and the documentation of methodologies and data resources used in its development.

The Transit Oriented Communities Tool (TOC Tool) is a website devoted to transit oriented communities (TOC) in Miami-Dade County, specifically in corridors and station areas defined by the Strategic Miami Area Rapid Transit Plan (SMART Plan). The TOC Tool offers introductory information to help site visitors understand what TOC is, learn about design conventions in station area walksheds, and become acquainted with key topics for measuring and monitoring TOC implementation progress. The four key topics reported in the TOC Tool are Economic Development, Urban Form, Regional Travel, and Local Access.

Within each reporting topic, elements of effective TOC are defined as follows:

TOC is effective when....	
Economic Development • regional growth is directed to SMART Plan corridors and station areas	**Regional Travel** • multimodal travel is increasing • the transportation system provides access to jobs and essential services by a variety of modes • SMART Plan corridors and station areas generate fewer and shorter vehicular trips than the rest of the region
Urban Form • development patterns promote multimodal travel • expansion of the region's development footprint is modest and consistent with local and regional plans	**Local Access** • SMART Plan corridors and station areas offer opportunities for work, shopping, school, and recreational trips by non-motorized modes • SMART Plan stations are reachable by walking from most parcels in the station areas • Parks and other public spaces are reachable by walking from most parcels in the station areas

For each element of effective TOC, the TOC Tool offers interactive dashboards with web maps, data summaries, and graphic visualizations to track TOC implementation progress over time. Users can select reporting areas from dropdown menus in the dashboards, specifying corridors and station areas. The selection results are combined to filter the dashboard contents and focus reporting on the selected location. The options allow users to explore metrics for individual station areas, entire corridors, or the whole SMART Plan system.

Finally, the TOC Tool provides additional information pertaining to TOC planning in Miami-Dade County, links to relevant related planning efforts in the region, and a review of academic and industry literature pertaining to TOC and similar development paradigms.

This document presents detailed technical information about the scripts, data, web resources, and documentation procedures used to construct the TOC Tool. High-level overviews of scrips and data organization are provided in the following sections with rigorous detail presented in appendices. Key information for exploring and maintaining the TOC Tool's web resources in the ArcGIS Online (AGOL) platform is also presented in the "AGOL" section. Finally, the "Documents" section addresses the process used to build and maintain documents that report the methodological details and metadata for all metrics reported by the TOC Tool.

SCRIPTS

The TOC Tool tracks and reports numerous metrics pertaining to TOC in Miami-Dade County. These metrics depend on an array of data inputs and on consistent, reliable methods to generate findings. For this reason, most of the work required to build and maintain the TOC Tool is codified in a series of scripts written primarily in the Python (version 3.7 or later) programming language, in a customized environment that leverages arcpy (via ArcPro version 2.7 or later) and a host of supporting packages. Some scripts written in the R language (version 4.0.2 or later) are also provided to support TOC Tool documentation needs.

This section provides technical guidance for configuring the development environment so that TOC Tool procedures can be executed. It then outlines the organization of the scripts used to develop the TOC Tool, running from project configuration to data download, processing, final reporting, and documentation. Each set of scripts is summarized to highlight its role in the TOC Tool development process. Details of script methods are available in Appendix B.

DEVELOPMENT ENVIRONMENT CONFIGURATION

A set of custom python environments are used to manage the process of downloading, preparing and building the data sets utilized in the TOC Tool. Due to compatibility issues with the OSMnx library, the download procedure requires its own environment. The instructions below outline the process needed to stand up the necessary python environments. Note however, that if ArcGIS Pro is updated to a new major or minor version (ie X.Y.Z, where X is the incremented major version number, Y is the incremented minor version, and Z is a patch or bug fix version) you will need to rebuild the *pmt_tools* environments from scratch as the underlying python dependencies used by *arcpy* will likely have changed. *pmt_download* does not depend on arcpy.

pmt_download (conda environment):

1. Select Windows Start
2. Navigate to 'Python Command Prompt' under ArcGIS folder --> Open
3. In the command window run the below commands:
4. *Create a new environment*
    - `>>> conda create --name %LocalAppData%\ESRI\conda\envs\pmt_download`
5. *Activate the newly created environment*
    - `>>> activate pmt_download`
6. *Install conda packages using **conda-forge** channel*

- `>>> conda install geopandas pandas numpy scipy rtree=0.9.4 -y`

7. Install pip packages
   - `>>> pip install osmnx censusdata`

pmt_tools (conda environment):

1. Select Windows Start
2. Navigate to 'Python Command Prompt' under ArcGIS folder --> Open
3. In the command window run the below commands:
4. *Create a new environment*
   - `>>> conda create --clone arcgispro-py3 --name %LocalAppData%\ESRI\conda\envs\pmt_tools`
5. *Activate the newly created environment*
   - `>>> activate pmt_tools`
6. *Install conda packages using **conda-forge** channel*
   - `>>> conda install -c conda-forge momepy sphinx dask -y`
7. Install pip packages
   - `>>> pip install simpledbf`

Script Organization

Config

Configuration files have been provided to define parameters and settings of the Download, Prepare, and Build procedures. Each process has its own configuration specific to that process, however the configurations provide standardization of information across processes allowing for consistent usages between them. Localizing these parameters allows a user to quickly locate hard coded values and make changes easily. Configurations are provided as python files for straightforward import to the procedures.

Download

The download module performs all automated download procedures for raw data used in the tool. These include Census geographies and economic/demographic tabular data, OpenSteetMap networks and building footprints, NLCD Impervious surface data, and a series of supporting vector layers available via REST endpoints. Where appropriate these data are organized by year to reflect changing conditions (ie Parcel records, Census demographics, etc). All data downloaded will reside in the RAW folder within the project ROOT directory.

Note there are a series of RAW datasets that cannot be downloaded and require coordination with other agencies or manually acquiring those datasets.

Step by step:
- Select Windows Start
- Navigate to 'Python Command Prompt' under ArcGIS folder --> Open
- `>>> activate pmt_downloads`
- `>>> python /path/to/downloader.py`

Prepare

The prepare module standardizes and formats all raw datasets into a common storage pattern, normalizing or separating geospatial data from tabular data where possible to decrease overall file sizes. The standardized databases include PMT_BasicFeatures and PMT_YYYY (where YYYY = the relevant year of data). Standardization includes removing unnecessary attributes, renaming attributes for readability, merging data where needed, and placing outputs in a common geodatabase structure. In addition to standardization, much of the analytical processing is performed via this module. Outputs of the prepare process are stored in the CLEANED folder within the project ROOT directory.

Step by step:

- Select Windows Start
- Navigate to 'Python Command Prompt' under ArcGIS folder --> Open
- `>>> activate pmt_tools`
- `>>> python /path/to/preparer.py`

Build

The build module generates the final time series geodatabases used in the Experience Builder site. For each PMT_XXX geodatabase, a Snapshot summary is generated, creating wide tables and calculating new attributes by aggregating data up from lower spatial scales. Tables long on categorical information are also created. Trend and NearTerm (trend) geodatabases are created, both utilizing the same procedure and creating tables long on year and calculating difference values for start and end time frames (Trend: start = 2014, end=current year; NearTerm: start = current year, end = forecasted near term from permits) These pre-canned calculations and transformations allow for more robust and speedy indicators and widgets in the site. Outputs of the build process are stored in the BUILD folder within the project ROOT directory.

Step by step:

- Select Windows Start
- Navigate to 'Python Command Prompt' under ArcGIS folder --> Open
- `>>> activate pmt_tools`
- `>>> python /path/to/builder.py`

Docs

R Markdown files that use pandoc and the Metric Inventory and Data Inventory spreadsheets to create .docx files included as Appendix A to this Technical Guide. These files need to be updated if methodologies described in the markdown file are updated.

Step by step:

- Start R Studio
- Open *PMT_Methodology.rmd*
- Click “Knit” to open a dropdown menu -> select “Knit to Word” to create a .docx output file.

All scripts include detailed internal documentation in the form of script headers and class/function docstrings. These annotations are used to develop the comprehensive script documentation included as Appendix B to this Technical Guide. This process relies on the sphinx Python module and is automated by calling the `make html` command from the root directory.

DATA ORGANIZATION

All TOC Tool supporting data are stored in a standard directory structure described below. TOC Tool scripts populate and update files in each section as follows:

- downloader.py: RAW
- preparer.py: CLEANED
- builder.py: BUILD

ROOT

Each TOC Tool script utilizes the PMT_tools package. The package is organized according to the processing phases described in the TOC Tool scripts section, all of which define a root directory where raw, cleaned, and build data outputs are stored. **IMPORTANT! The user defines the root in the PMT_tools.PMT.py file. Make sure this file is updated to reflect your workspace.**

RAW

Raw data are populated by the downloader.py script and manually by the user. Raw data are sourced from a variety of agencies, some of which required inter-departmental coordination. For details on data sources used in the TOC Tool, see Appendix A (data inventory section). For information on inter-departmental coordination, consult the TOC Tool Maintenance Plan (Appendix C).

CLEANED

TOC Tool metrics are calculated by processing and combining RAW data, applying the methods described in Appendix A through the preparer.py script. The resulting data are stored in standardized geodatabase outputs with the name PMT_{year}. Data are processed and organized by year to support the development of trend and near-term analyses downstream. Each output generated by preparer.py is stored as one or more distinct feature classes or tables within a given year's geodatabase to support selective process and/or data updates without re-running the entire preparer process.

BUILD

The TOC Tool data products are the geodatabases housed in the BUILD folder. These are created by builder.py. They are the feature classes and tables consumed in all TOC Tool ArcPro projects and the sources for all AGOL feature services. A "snapshot" geodatabase is generated for every analysis year in the CLEANED folder. The most recent year's resulting geodatabase is labeled as *Current_Snapshot.gdb*. Snapshots of earlier years and of a simulated near-term outlook (conditions with permitted development) provide the bases for developing additional geodatabases that record changes over time: *Trend.gdb* and *NearTerm.gdb.*

AGOL

ORGANIZATION OF SERVICES

This section describes the general outline and organization of feature and tile services utilized in the tool. Hosted feature services have been created for each time frame, separated by features and tables. Tile services have also been created to increase the drawing efficiency of the maps they support. Within ArcGIS Online the services have been organized into five folders:

- PMT [Feature Service Layers]
- PMT [Table Service Layers]
- PMT [Tile Service Layers] – SNAP
- PMT [Tile Service Layers] – TREND
- PMT [Tile Service Layers] – NEAR

There are three individual ArcGIS Pro projects associated with the TOC tool: Snapshot, Trend, NearTerm. Feature, table, and tile services are managed through these projects as maps within the project.

Feature Services

For organizational purposes, all features were combined into a single service with multiple layers. Any new data layers added to a given service should be appended to the end as they are indexed based on the order in the map. The BasicFeatures service is managed within the Snapshot ArcPro project.

Table 2 - Feature Service Layer list by Time Frame

Snapshot Service Layers	Trend Service Layers*	NearTerm Service Layers
Bike comfort	No trend feature layers	BuildingPermits
Census Block Groups		Census_Blocks_diff
Census Blocks		MAZ_diff
Intersections		SummaryAreas_diff
Park_Points		TAZ_diff
MAZ		
Station area walk shed		
TAZ		
Transit Stops		
Walk Time to Parks		
Walk Time to SMART Plan Stations		
Bike Facilities		

*Trend services were combined into a single "table" service (features and tables), as we originally didn't intend to use the features generated for any dashboard work, only as Tile Services.

Table Services

The table services include at minimum one feature service to allow for overwriting existing data in AGOL. Currently hosted table services do not appear to be over-writeable via the Share functionality of ArcPRO.

Table 3 - Table Service Layer list by Time Frame

Snapshot Table Layers	Trend table Layers*	NearTerm table Layers
ActivityByTime_Auto	ActivityByTime_Auto_byYear	ActivityByTime_Auto_byYear
ActivityByTime_Bike	ActivityByTime_Auto_diff	ActivityByTime_Auto_diff
ActivityByTime_Transit	ActivityByTime_Bike_byYear	ActivityByTime_Bike_byYear
ActivityByTime_Walk	ActivityByTime_Bike_diff	ActivityByTime_Bike_diff
AreaByDevStatus	ActivityByTime_Transit_byYear	ActivityByTime_Transit_byYear
BikeFacilityMilesByTier	ActivityByTime_Transit_diff	SummaryAreas_diff
CommutesByMode	ActivityByTime_Walk_byYear	
JobsBySector	ActivityByTime_Walk_diff	
TransitByTimeOfDay	AreaByDevStatus_byYear	
WalkTimeToParks	AreaByDevStatus_diff	
WalkTimeToStations	AttrByLU_byYear	
SummaryAreas	AttrByLU_diff	
	CommutesByMode_byYear	
	CommutesByMode_diff	
	JobsBySector_byYear	
	JobsBySector_diff	
	SummaryAreas_byYear	
	TransitByTimeOfDay_byYear	
	TransitByTimeOfDay_diff	
	WalkTimeToParks_byYear	
	WalkTimeToParks_diff	
	WalkTimeToStations_byYear	
	WalkTimeToStations_diff	
	BikeFacilityMilesByTier_byYear	
	BikeFacilityMilesByTier_diff	
	Census_Blocks_diff	
	MAZ_diff	
	SummaryAreas_diff	
	TAZ_diff	

Tile Services

The tile services are generated to increase drawing efficiency of maps associated with dashboards. Traditional Tile services have been used in this iteration of the tool as the symbology for the tile service doesn't show up in web maps legend, rendering the symbology un-useful.

Table 4 - Tile Service Layer list by Time Frame

Snapshot Tile Layers	Trend Tile Layers*	NearTerm Tile Layers
Bike Facilities	Change in jobs:housing balance of Walkshed (since 2014)	Change in Estimated Employment (permitted development)
Daily Transit Boardings and Alightings	Change in jobs:housing balance of Bikeshed (since 2014)	Change in FAR by block (permitted development)
Daily Vehicle Miles of Travel Generated by TAZ	Change in jobs within a 30 minute Walk (since 2014)	Change in jobs-housing ratio by block
Estimated Market Value ($) per Square Foot by Block	Change in jobs within 60 minutes by Transit (since 2014)	Change in Residential Units by Block (permitted development)
Floor Area Ratio by Block	Change in jobs within a 30 minute Bike (since 2014)	Change in Total Living Area (permitted developement)
Generalized Land Use by Parcel	Change in jobs within 60 minutes by Auto (since 2014)	
Impervious Surface Coverage by Block	Change in daily transit boardings and alightings (since 2014)	
Jobs by Block	Change in Effective Developable Area by Block (since 2014)	
Jobs within 30 minutes by bike	Change in Estimated Market Value ($) per Square Foot (since 2014)	
Jobs within 30 minutes by walking	Change in Floor Area Ratio by Block (since 2014)	
Jobs within 60 minutes by car	Change in Jobs by Block (since 2014)	
Jobs within 60 minutes by transit	Change in Jobs within 30 minutes by Walking (since 2014)	
Jobs:Housing Balance by Block	Change in Jobs:Housing Balance by Block (since 2014)	
Jobs:Housing Balance in a 30-minute Bike Ride	Change in Non-Auto Mode Share (since 2014)	
Jobs:Housing Balance in a 30-minute Walk	Change in Residential Units by Block (since 2014)	
Non-Auto Mode Share by Block	Change in Taxable Value ($) per Square Foot (since 2014)	
Number of workers working from home by block	Change in Total Floor Area by Block (since 2014)	
Residential Units by Block	Change in Working from Home (since 2014)	
Street Connectivity Index by Intersection		
Streets by Bicycle Comfort Level		
Taxable Value ($) per Square Foot by Block		
Total Developable Area by Block		
Vehicular Trip Length by TAZ (from)		
Walk Time to Nearest Park		
Walk Time to Nearest SMART Plan Station		

Service Update Procedure

Feature/Table Services (Overwrite a web feature layer—ArcGIS Pro | Documentation)
This is a simple workflow that only requires the existing services to be overwritten. If a back-up copy is desired, make a copy of the existing service before overwriting.

- Open the ArcPro project for a given time period (Snapshot, Trend, NearTerm)
- Navigate to the map of interest
- Select the Share tab
- Follow the Overwrite Feature Layer procedure outlined in the link above
    a. Web Layer → Overwrite Web Layer
    b. Locate the feature layer in your AGOL organization to be overwritten
    c. Select OK in modal window that opens
    d. In "Sharing" panel:
        i. Summary and tags information should already be present and stored in the feature/table service. Only update if these data have changed in some way.
        ii. Run Analyze to verify share settings
        iii. Run Publish
- The process will need to be repeated for other time periods as needed.
- The Basic Features service is managed within the "Snapshot" ArcPro project.

Map/Tile Services
This is a two-stage process. The updated data layer must first be published as a new Tile Service, then the old service is replaced in the second step.

- Open the ArcPro project for a given time period (Snapshot, Trend, NearTerm)
- Navigate to the Tile Service Map within the project (ex: 'Snapshot Tile Services')
- (STEP 1) Follow the Share a web tile layer process (Share a web tile layer—ArcGIS Pro | Documentation)
    - Toggle on the layer to be updated
    - Right-click the layer → Sharing → Share As Web Layer
    - In the `Share As Web Layer` dialog that opens:
        - General
            - Toggle 'Tile' under Layer Type
            - Define a location, suggest a well named folder (ex: MapTiles_folder)
            - Select AGOL Groups to share the tile layer with
        - Configuration
            - Tiling Scheme – ArcGIS Online / Bing Maps / Google Maps (ensures data are projected to Web Mercator)
            - Levels of Detail – suggest 9 to 17 (counties to city block)
            - Options
                - Cache Locally (builds tile package on local machine and reduces charges/credit use in AGOL) – navigate to a local location on the machine

    - Run Analyze to verify share settings
    - Run Publish (this will take some time depending on the scale and detail of the data)
- (STEP 2) Follow the Replace Web Tile Layer process (Replace a web tile layer—ArcGIS Pro | Documentation)
    - Select Share tab
    - Select Replace Web Layer
    - In `Geoprocessing` window:
        - Target Layer – locate the existing Tile service in AGOL to be overwritten
        - Archive Layer Name – leave the default created by ArcPro
            - <target layer>_archive_<timestamp>
        - Update Layer – locate the new Tile service previously created
        - Run the tool

These steps would need to be repeated for all Tile Services to be updated. Each portion of the workflow could be completed in batches (i.e. Share web tile layer for all layers needing to be updated, then Replace web tile layer for these layers).

MAP CONVENTIONS (SNAP, TREND/NT)

This section describes the map conventions used for each map associated for each "element of effective TOC," separate maps were created for each element.

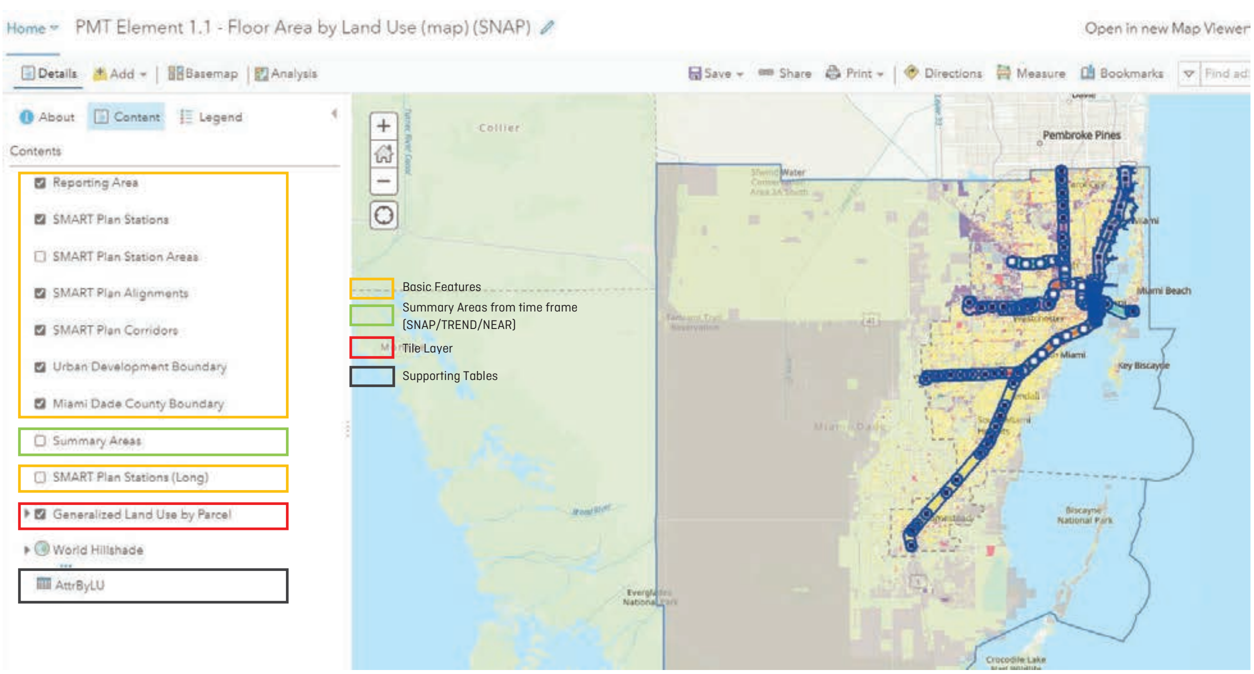


Figure 12 - TOC tool map layer organization

Layers and tables to include

The supporting maps are built using a consistent pattern of layers and supporting tables. At the base each map includes the ESRI World Topographic Map and the layers from the BasicFeatures feature service, providing a consistent look and feel to each map. A Summary Areas layer from an associated time frame

(SNAP/TREND/NEAR) is included to power the summarization information used by dashboards, utilizing the tabular data. Additionally, any supporting long form tabular data associated with the sub-element topic is included for dashboard widgets.

Symbology

Symbology of the layers in the BasicFeatures feature service as well as Tile Services are defined and applied in ArcPRO prior to creating the services. BasicFeature styling for the alignments, corridors adhere to the Strategic Miami Area Rapid Transit SMART Plan color palette. Layer files for all Tile layers and Basic Features are included in maps are provided as within the project folders.

Dashboard conventions

This section describes the content, layout, and interactions of the TOC Tool element dashboards. For each "element of effective TOC," separate dashboards were constructed for each reporting time frame (Snapshot, Historical Trend, or Near Term) as shown in Table 1 below. There are 19 element dashboards, and a separate simple SMART Plan Overview dashboard, making 20 total dashboards.

Table 5 - Dashboards by Element and Time Frame

Topic	Element	Time Frames
Economic Development	Growth Monitoring	Snapshot, Trend, Near Term
Urban Form	Development Patterns	Snapshot, Trend
Urban Form	Urban Footprint	Snapshot, Trend
Regional Travel	Multimodal Travel	Snapshot, Trend
Regional Travel	Regional Access	Snapshot, Trend
Regional Travel	Vehicular Travel	Snapshot, Trend
Local Access	Non-Motorized Opportunities	Snapshot, Trend
Local Access	Access to Transit	Snapshot, Trend
Local Access	Access to Parks	Snapshot, Trend

Layout

All TOC Tool dashboards conform to a standard layout, as shown in Figure 13 below. The layout includes as the dominant visual element the web maps from which the data that drive dashboard widgets are sourced. Maps are pinned to the lower right corner of the dashboard. Reporting widgets are assembled on the left side of the dashboard, filling the same vertical space occupied by the map. The widgets report specific metrics. For dashboards that report multiple metric sets, each set is stacked as a combination of map-and-widgets in the tabs displayed in the lower right of the dashboard. The dashboard title bar is pinned to the top of the dashboard above the maps and widgets. The title bar usually[1] includes two dropdown selector menus to drive dashboard navigation based on SMART Plan corridor and station names.

[1] The lone exception is the SMART Plan Overview dashboard, which has only a corridor selection dropdown menu.

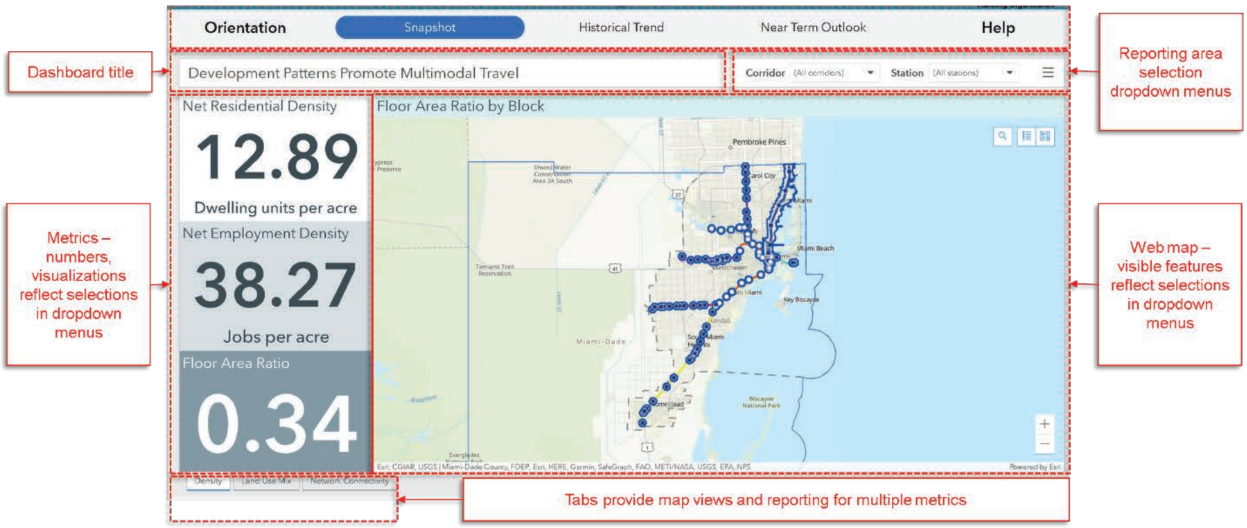


Figure 13 - Standard TOC Tool Dashboard Components and Layout

Maps

Maps are added to the dashboard by clicking the widget selector [+ ▾] icon in the dashboard editing interface and choosing "Map" from the dropdown menu, which opens the map selection window. Find and select the map to add to the dashboard. Once a map has been added to a dashboard it cannot be replaced by a different map. The process to replace involves deleting the old map and adding the replacement map following the steps defined above. Any widgets that use data in the existing map will be deleted or will lose their connection to underlying data when the map is deleted, so map replacement can sometimes be an onerous task. However, see the Potential Enhancements section below for emerging AGOL capabilities that may alleviate this burden.

Selectors

The reporting area selection dropdown menus in the upper right corner of each dashboard are the primary controllers of dashboard views. The dropdown menus determine which features are visible on the maps as well as the geography of the data reported in the metric widgets. The details of how each dropdown menu is configured are outlined below.

To add a selector dropdown menu to a new dashboard

With the dashboard opened in editing mode, Hover over the blue rectangle in the upper left corner of the title bar. This loads a small menu of actions that affect the title bar. Choose "add category selector" to create a dropdown menu in the title bar.

Configuring selector dropdown menus

Configuring a selector involves connecting to a data source, setting display options, and defining actions that follow when users click on items in the menu.

The "Corridor" selector dropdown menu:

- Data
    - Source: *SMART Plan Corridors* feature layer (*PMT_BasicFeatures_feat_Service*)
    - Filter: None

    - Line item template: *{Corridor}* or blank
    - Maximum categories: 50
    - Sort by: *Corridor* (ascending)
- Selector options
    - Label: *Corridor*
    - Selection: Single
    - Operator: equal
    - Preferred display type: dropdown
    - Display type threshold: 10
    - None option: disabled
    - Default selection: First
- Actions
    - Filters:
        - "Station" selector dropdown menu, where Source and Target fields are "Corridor"
        - "SMART Plan Corridor" feature layers in all dashboard webmaps, where Source and Target fields are "Corridor"
        - Any list widgets in the dashboard, where Source and Target fields are "Corridor"

The "Station" selector dropdown menu:

- Data
    - Source: *Reporting Area* feature layer ("SummaryAreas" for feature service matching the dashboard's reporting time frame)
    - Filter: None
    - Line item template: *{Name}* or blank
    - Maximum categories: 50
    - Sort by: *Name* (ascending)
- Selector options
    - Label: *Station*
    - Selection: Single
    - Operator: equal
    - Preferred display type: dropdown
    - Display type threshold: 10
    - None option: disabled
    - Default selection: First
- Actions
    - Filters:
        - "Reporting Area" feature layers in all dashboard webmaps, where Source and Target fields are "RowID"
        - All relevant non-list widgets in the dashboard, where Source and Target fields are "RowID"
    - Zoom: All maps
    - Flash: All maps

Widgets

TOC Tool dashboard widgets expose data from webmap feature layers or tables, reporting metrics for the geographic area defined by the selector dropdown menus. Widgets typically have a title set under the "General" widget options. Titles are typically formatted as the "Heading 3" style. Notes are provided in normal text style in the Description section. To help visually distinguish among widgets when viewing the dashboard in the TOC Tool Experience Builder app, each widget is given a different background/foreground(text) color scheme.

- Top widget:
  - Background color: #FFFFFF
  - Text color: #3E4D54
- Middle widget:
  - Background color: #D6DDE0
  - Text color: #3E4D54
- Bottom widget:
  - Background color: #8298A4
  - Text color: #FFFFFF

The following common widget types and configuration conventions are used throughout the TOC Tool dashboards:

- **Indicators**: These widgets simply display numerical data with light annotation. They typically refer to data in the "Reporting Area" feature layer of a given webmap. Indicators are typically filtered to reflect features selected in the Station selector dropdown menu. In rare cases, the indicator may use a "reference value" that is unfiltered.
- **Lists**: Display customized feature attributes in an ordered list. They typically refer to data in the "Reporting Area" feature layer of a given webmap. Lists are always filtered to reflect features selected in the Corridor selector dropdown menu. The content of a list always follows the following convention:
  - Data:
    - Filter: *Name != (All stations) AND Name != (Entire corridor) AND (Name != (Outside station area)*
    - Maximum features displayed: 50
    - Sort by: {the key field being displayed in the list}, usually in descending order, but this varies by topic.
  - List options
    - Line item template:

      {Name} ({Corridor})
      Description of metric to report: {*Field name*}

    - Line item icon: None
    - Text color: #FFFFFF
    - Background color: #8298A4
    - Separator color: #CCCCCC

    - Selection color: #3e4d54
    - Selection text color: #FFFFFF
- **Serial Charts**: Serial charts display data that are subdivided into classes or series. They typically refer to a table that records metrics by category, such as floor area by land use or jobs by industry sector. The most common serial chart used in the TOC Tool is the bar chart, but line and area charts follow similar conventions. Serial Charts are always filtered to reflect features selected in the Station selector dropdown menu. Most serial charts include a legend
- **Pie Charts**: Pie charts are a special type of serial chart that focus on relative shares of values by class or series rather than absolute values. They typically refer to a table that records metrics by category, such as dwelling units by unit type. Pie Charts are always filtered to reflect features selected in the Station selector dropdown menu.

Potential Enhancements

This section identifies and describes potential near-term enhancements to the TOC Tool dashboard development and maintenance processes.

- All TOC Tool dashboard widgets source data from webmaps in their respective dashboards, consuming feature layers and table views in the maps. Recently, AGOL has added capability to source widget data directly from feature and table services rather than referencing items in a webmap. Doing so would likely mitigate the risk of losing widgets or severing connections between widgets and data sources when webmaps are modified or deleted.

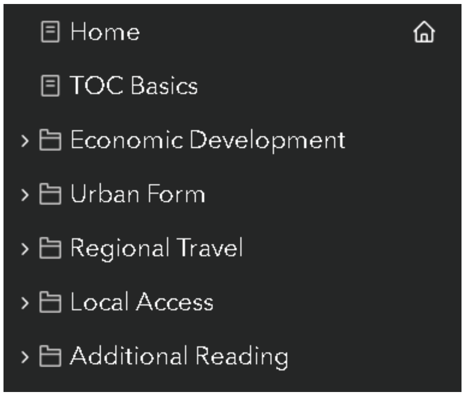
EXPERIENCE BUILDER

The TOC Tool dashboards are embedded as web pages within and Experience Builder website in AGOL. This is necessary to allow for the time frame navigation for each TOC element. Experience Builder is a what-you-see-is-what-you-get (WYSIWYG) application for developing websites and exposing related AGOL data, maps, and dashboards through built-in widgets. Web page creation, editing and organization is well documented in ESRI's help for the Experience Builder product. There are existing security issues related to the Content-Security-Policy headers for the out of the box version of Experience Builder. These vulnerabilities cannot be addressed for the version housed solely in AGOL. However, there is potential to migrate the application to a Developer version of the tool, whereby the Miami-Dade IT team would be able to serve the application from their own internal server and have control over the header variables used.

To update the Experience Builder website, open the Miami-Dade Transit Oriented Communities Tool application in AGOL in Edit mode. Make changes in the edit session, saving along the way by clicking on the icon. Any saved changes remain unpublished but viewable to authorized users in an edit session. When changes are finalized and ready to be exposed to the public, click the "Publish" button in the upper right corner of the Experience Builder application.

Organization

To view the TOC Tool site organization, click on the icon in the panel pinned to the left side of the Experience Builder application. This will load the page organization panel, as shown below.



Pages can be free-standing or organized within folders. To hide content find a page or folder in the organizational panel, highlight it, and click the icon to hide. Follow the same procedure to make hidden comments visible again. The dashboards displayed in the current TOC Tool website are all free-standing pages with names reflecting the TOC element and the word "Fullscreen" ("Development Patterns Fullscreen", e.g.).

- To create a new page, click the icon at the top of the organizational panel.
- To rename a page, find it in the organizational panel, click the three dots next to the item, and select "Rename." Type the new name and hit enter.
- To copy a page, find it in the organizational panel, click the three dots next to the item, and select "Duplicate."
- To delete a page, find it in the organizational panel, click the three dots next to the item, and select "Delete." You will not be prompted to confirm the deletion, so exercise caution. If a page or folder is accidently deleted, use the undo button in the Experience Builder applications top ribbon to restore the deleted item.
- To update a page's contents, highlight it in the organizational panel to load it in the Experience Builder applications main window. Select items on the page by pointing and clicking or highlight them in the page outline panel beneath the organizational panel. Options for modifying the style and content of any selected item are presented in the panel pinned to the right side of the Experience Builder application.

DOCUMENTS

METRIC INVENTORY

The Metric Inventory spreadsheet (*Metric_Inventory.xlsx*) records methodological information about each metric reported in the TOC Tool. This spreadsheet is read by the script *PMT_Methodology.R* to construct detailed methodological documentation. It consists of the following tabs:

- **Topics and Elements**: names and enumerates topics and elements of effective TOC reported in the TOC Tool. Each element is assigned to a single topc.
- **MetricMethodology**: lists each metric reported in the TOC Tool. This is the primary worksheet used to develop methodological documentation (see Appendix A - Metrics). Important column descriptions are provided below:

- **Key**: Indicates which element the metric belongs to, referring to the Code established in the "Topics and Elements" tab.
- **Include**: Some metrics were explored as part of the TOC Tool's development. Details of these metrics remain in the spreadsheet, but they are marked with a value of 0 (zero) in the "include" column. Metrics that are fully developed and reported in the TOC Tool are marked with a value of 1.
- **Tab Num**: Specifies the tab ordering in TOC Tool dashboards for this and related metrics (see "Tab Name" below).
- **Tab Name**: Specifies which tab this metric is reported on within its "Element" dashboard. Similar metrics are reported on the same tab. Tabs are used to organize dashboards for legibility.
- **Topic**: The name of the topic within which the metric is reported. Used for filtering the spreadsheet and formulaically looked up based on "Key". Each topic consists of one or more elements.
- **Element**: The name of the element within which the metric is reported. Used for filtering the spreadsheet and formulaically looked up based on "Key". Each element has its own dashboard in the TOC.
- **Metric**: The name of the metric.
- **Map Dataset**: If the metric is portrayed in the map pane of its "Element" dashboard, the data source for the mapped data is named here.
- **Map Attribute**: If the metric is portrayed in the map pane of its "Element" dashboard, the field within "Map Dataset" is named here.
- **Widget Dataset**: If the metric is portrayed in widget within its "Element" dashboard, the data source for the data reported by the widget is named here.
- **Widget Attribute**: If the metric is portrayed in widget within its "Element" dashboard, the field within "Widget Dataset" reported by the widget is named here.
- **Summary**: A brief description of the metric to guide interpretation.
- **Source**: The raw input data source(s) used to develop the metric. If multiple data sources are relied on, they are listed in a semi-colon-separated character string. Values provided are expected to correspond to values in the "Name" column of the Data Inventory spreadsheet.
- **Methodology**: Describes the calculation of the metric. In some cases, the metric derivation depends on complex analytical procedures that support multiple metrics. Rather than repeat those details for each metric, references are made to procedures by name using markdown syntax. This means referring to a document heading in square brackets "[ ]". The headings to go within the brackets must match a heading in the *PMT_Methodology.R* script.
- **Notes**: This column offers the opportunity to cite literature or otherwise provide insight as to why the metric is included, its relationship to TOC, and/or how metric values may be interpreted. Notes are not included for every metric. References to other resources (literature, e.g.) are provided in markdown hyperlink syntax: [text to display](link/to/url.pdf).

- **GDB specs**: offers detailed information about all datasets and fields in the TOC Tool geodatabases.

Data Inventory

The Data Inventory spreadsheet (*Data_Inventory.xlsx*) records information about each raw input dataset used to support metric development for the TOC Tool. This spreadsheet is read by the script

PMT_Methodology.R to construct a detailed data inventory. It consists of multiple tabs, but one key tab ("Data") is used to construct the documentation of data sources. Key columns in the "Data" tab include:

- **Name:** A common name by which to refer to the source data. Corresponds to values in the "Source" column of the "MetricMethodology" tab in the Metric Inventory.
- **Description:** A brief description of the dataset, what it represents, and what kinds of information it includes.
- **Type:** "Spatial" or "Tabular"
- **Shape Type:** If "Type" is "Spatial", the geometry type ("Point", "Line", "Polygon", e.g.)
- **Geographic Scope:** The geographic area for which data are available from this source.
- **Spatial Unit:** The geographic unit represented by each row in the dataset.
- **Latest Vintage:** Provides a version number or similar indication of which edition of this dataset was used in the TOC Tool.
- **Data Published:** When the dataset was published – for some datasets this may be a specific date, for others it may be the year.
- **Update Schedule:** Defines the frequency with which the dataset is updated.
- **Publisher/Provider:** Identifies the agency/entity who publishes the dataset.
- **Mechanism to Obtain:** Describes how data are obtained for use in the TOC Tool. There are two primary modes of data acquisition. For data that can be obtained through a "Download", urls to sources of data or descriptions of the procedure used to fetch data are provided. For data that require "direct coordination", agency and contact information are provided.
- **Key Fields:** Highlights key fields in each dataset that identify features and/or relate them to other features. In some cases, fields that provide distinctive insight for TOC analysis are also listed in this column.
- **Related Tables:** Identifies other datasets that are related to this dataset.
- **Quality Concerns:** Acknowledges potential concerns pertaining to data quality or usage.

Appendix A: Metrics, Methods, and Data

Introduction

The Transit Oriented Communities Tool (the TOC Tool) is a web-based data visualization tool to track progress towards local and regional planning and development goals along major rapid transit corridors in the Miami-Dade region. The TOC Tool aims to provide insight into how transit station areas and corridors change over time through public and private investments in housing, commercial buildings, recreational space, transportation infrastructure and more. The TOC Tool focuses on transit-oriented development (TOD) areas, which combine the region's growth management emphasis on urban centers and nodal community development with areawide investment in rapid transit and multimodal infrastructure. Trends in growth and development, multimodal travel, urban design, accessibility, and more are summarized and tracked for rapid transit corridors and station areas. The TOC Tool is a collaborative product led by the Miami-Dade Transportation Planning Organization (the TPO) in conjunction with local jurisdictions and partner agencies.

The TOC Tool development effort is guided by the following goals, which aim to maximize its relevance and value as a data and analysis resource for policy planners and decision-makers.

- Focus on the goals and objectives of the Strategic Miami Area Rapid Transit (SMART) Plan that can be addressed through TOC.
- Develop performance metrics that clearly indicate the region's performance relative to SMART Plan TOC goals across multiple topics.
- Track trends over time by summarizing baseline performance and routinely updating data and metrics.
- Provide a simple interface for users to summarize the story of TOC at various scales.
- Explore detailed metrics and data.

This document provides a detailed inventory of metrics reported in the TOC Tool and visualized in web maps and dashboards. Metrics are organized by reporting topic. Details pertaining to data sources, metric development (methodology), etc. are listed for each metric. More information pertaining to data sources listed for each metric may be found in the Data Inventory section of this document. Finally, some metrics rely on complex procedures that combine multiple datasets. For relevant metrics, these procedures are referenced among the methodological information provided, but the details of the procedures are found in the Analytical Procedures section of this document.

The TPO developed an organizational framework defining appropriate topics and metrics for TOC reporting and monitoring in conjunction with the Project Working Group (PWG), a panel of planning professionals representing many of the TPO's local and regional partners in implementing the SMART Plan. The framework was informed by a review of industry literature pertaining to TOC and similar reporting/monitoring efforts in peer regions across the country. The inventory of prospective TOC metrics provided below synthesizes the results of these combined efforts.

Metrics

Economic Development

Regional growth is directed to SMART Plan corridors and station areas

- **Number of Residential Units**
  - NO_RES_UNTS
    - *Summary:* The count of residential units as shown in the parcel record
    - *Source(s):* Florida DOR
    - *Method:* This attribute comes directly from the Miami Dade parcel record.
    - *Notes:* The FDOT TOD Framework suggests residential targets for station areas (see pages 38-43)
- **Share of Total Residential Units in County**
  - SHR_RES_UNTS
    - *Summary:* This summary area's share of countywide residential units
    - *Source(s):* Florida DOR
    - *Method:* This attribute comes directly from the Miami Dade parcel record.
    - *Notes:* NA
- **Units by Type**
  - GN_VA_LU; NO_RES_UNTS
    - *Summary:* The number of residential units by land use type (multifamily or single-family)
    - *Source(s):* Florida DOR
    - *Method:* Unit counts come directly form the Miami Dade parcel record; land use codes are generalized based on the detailed codes provided in the parcel record.
    - *Notes:* NA
- **Number of Jobs**
  - Total_Employment
    - *Summary:* LODES jobs estimates disaggregated to parcels.
    - *Source(s):* Florida DOR; LODES
    - *Method:* LODES jobs estimates at the block group scale are disaggregated to the parcel level based on parcel land use codes and floor area estimates. See Demographic and Employment Estimation and Allocation analytical procedure
    - *Notes:* The FDOT TOD Framework suggests employment targets for station areas (see pages 38-43)
- **Share of Total Jobs in County**
  - SHR_Employment
    - *Summary:* This summary area's share of countywide jobs
    - *Source(s):* Florida DOR; LODES

    - *Method:* LODES jobs estimates at the block group scale are disaggregated to the parcel level based on parcel land use codes and floor area estimates. See Demographic and Employment Estimation and Allocation analytical procedure
    - *Notes:* NA
- **Jobs by industry sector**
  - Sector;Jobs
    - *Summary:* Disaggregated LODES data summarized by industry group
    - *Source(s):* Florida DOR; LODES
    - *Method:* LODES jobs estimates at the block group scale are disaggregated to the parcel level based on parcel land use codes and floor area estimates. See Demographic and Employment Estimation and Allocation analytical procedure. Jobs by the 20 NAICS categories reported in LODES are aggregated into 7 sectors: Resource jobs (Ag/mining); Industrial jobs (manufacture, distribution, wholesale); Consumer jobs (retail, accommodation, food services, entertainment); Office jobs (finance, real estate, professional, admin, e.g.); Health care jobs; education jobs; and other jobs
    - *Notes:* NA
- **Total Floor Area**
  - TOT_LVG_AREA
    - *Summary:* Total building floor area as shown in the parcel record
    - *Source(s):* Florida DOR
    - *Method:* This attribute comes directly from the Miami Dade parcel record.
    - *Notes:* NA
- **Share of Total Floor Area in County**
  - SHR_LVG_AREA
    - *Summary:* This summary area's share of countywide total floor area
    - *Source(s):* Florida DOR
    - *Method:* This attribute comes directly from the Miami Dade parcel record.
    - *Notes:* NA
- **Square Footage by Use type**
  - TOT_LVG_AREA;GN_VA_LU
    - *Summary:* Total building floor area by parcel land use code
    - *Source(s):* Florida DOR
    - *Method:* Floor area estimates come from the Miami Dade parcel record (TOT_LVG_AR); land use codes are generalized from the DOR_UC field to 7 groups: (1.)Vacant/Undeveloped; (2.) Single Family Residential; (3.) Multi-Family Residential; (4.) Industrial/Manufacturing; (5.)Commercial/Retail; (6.) Office; (7.) Other

- *Notes:* Transit-oriented communities encourage a mix of land uses at both the neighborhood and the corridor scale. A mix of land uses helps to create complete, walkable, and diverse neighborhoods around transit stations and stops and increases non-motorized travel (TOC Guidelines)

Urban Form

Development Patterns promote multimodal travel

- **Net Residential Density**
  - RES_DENS
    - *Summary:* Number of residential units per acre of residential land
    - *Source(s):* Florida DOR
    - *Method:* Dwelling unit estimates come from the Miami Dade parcel record (NO_RES_UNTS);residential land area estimates come from the Miami Dade parcel record (LND_SQFOOT) as the sum of values for all parcels having a positive number of residential units.
    - *Notes:* The FDOT TOD Framework suggests residential density targets for station areas (see pages 38-43). Optimal TOCs contain the highest density within a quarter-mile of the station area.
- **Employment Density**
  - EMP_DENS
    - *Summary:* Number of employees per acre of non-residential land
    - *Source(s):* Florida DOR; LODES
    - *Method:* LODES jobs estimates at the block group scale are disaggregated to the parcel level based on parcel land use codes and floor area estimates. See Demographic and Employment Estimation and Allocation analytical procedure; non-residential land area estimates come from the Miami Dade parcel record (LND_SQFOOT) as the sum of values for all parcels having a positive number of jobs estimated.
    - *Notes:* The FDOT TOD Framework suggests employment density targets for station areas (see pages 38-43)
- **Floor Area Ratio**
  - FAR
    - *Summary:* Total building floor area (square footage) per land area (square footage)
    - *Source(s):* Florida DOR
    - *Method:* Floor area estimates come from the Miami Dade parcel record (TOT_LVG_AR); land area estimates come from the Miami Dade Parcel record (LND_SQFOOT)
    - *Notes:* The FDOT TOD Framework suggests FAR targets for station areas (see pages 38-43). Optimal TOCs contain the highest density within a quarter-mile of the station area.

- **Jobs:Housing Ratio**
  - JHRatio
    - *Summary:* Number of jobs per housing unit in a summary area
    - *Source(s):* Florida DOR; LODES
    - *Method:* This attribute is derived from the "NO_RES_UNTS" and "Total_Employment" variables, summarized from the parcel level to the station area and corridor scales, as Total Employment in the summary area divided by the Number of Residential Units in the summary area.
    - *Notes:* A balanced jobs to housing ratio within a station area creates efficiencies for transit service and increases the likelihood of people being able to access a range of destinations (FDOT TOD Framework)
- **Mix of housing and jobs**
  - NO_RES_UNTS;Total_Employment
    - *Summary:* A graph comparing the proportions of total residential and non-residential activity in this summary area.
    - *Source(s):* Florida DOR; LODES
    - *Method:* Referenced attributes are derived from the "NO_RES_UNTS" and "Total_Employment" variables, summarized from the parcel level to the station area and corridor scales.
    - *Notes:* NA
- **Land use mix**
  - ENP
    - *Summary:* A land use mix index that counts present land uses weighted by their abundance in a summary area
    - *Source(s):* Florida DOR
    - *Method:* Mathematically, the reciprocal of the sum over all present land uses of the proportional abundance squared. The proportional abundance of a land use is the ratio of the floor area for all buildings of that land use over the total floor area (for all land uses) in the station area or corridor. The formula is: $\frac{1}{\sum_{LU} p_{LU}^2}$. It is scaled to [0,1] for interpretability using its fixed minimum (1) and maximum (number of possible land uses). The ENP metric is essentially a weighted count of land uses, where a land uses with higher proportional abundance in an area have greater influence on the resulting index. This metric has roots in political science but is used in many fields.
    - *Notes:* Transit-oriented communities encourage a mix of land uses at both the neighborhood and the corridor scale. A mix of land uses helps to create complete, walkable neighborhoods around transit stations and stops and create a mix of trip generators that can be reached by a variety of modes (TOC Guidelines)

- **Grid density**
  - GRID_DENS
    - *Summary:* Blocks per square mile
    - *Source(s):* Census Blocks
    - *Method:* The number of block features in each summary area (station area or corridor) divided by area in square miles.
    - *Notes:* Higher street grid density and smaller block sizes, combined with appropriately scaled and permeable building frontages, improve street level activity, pedestrian connectivity, and accessibility (FDOT TOD Framework)
- **Street connectivity index**
  - CentIdx
    - *Summary:* An intersection centrality index that describes how dense and well-connected streets are in a summary area
    - *Source(s):* OSM Networks
    - *Method:* For each node in the countywide bicycle network, a network-based distance is evaluated to all other nodes within a 1-mile radius. The centrality index is calculated as the number of other nodes reachable divide by the average distance required to reach those nodes. Higher scores indicate more nodes reachable over a shorter average distances. This calculation is performed for all nodes whose centrality index values are assigned to nearby parcels. Parcel values are then summarized to the station area and corridor scales using a weighted-average formula (based on building floor area) to track trends in floor area relative to network connectivity over time. This summarization addresses two concerns associated with a naive averaging of node-based scores to stations areas and corridors: (1.) street connectivity's impact on development patterns and travel behaviors is most relevant in areas where buildings are located, and (2.) since changes to street networks occur slowly over time, the floor-area-weighted average allows recent trends to be portrayed in the TOC tool based on whether building area is added in well-connected or poorly-connected areas.
    - *Notes:* Street connectivity and block length have strong relationships with walking and transit use. Short blocks and well-connected streets contribute to a higher-quality pedestrian experience and often occur in places where other elements of good design (sidewalks, etc.) are in place. (TOC Guidelines)

Expansion of the region's development footprint is modest and consistent with local and regional plans

- **Impervious surface coverage**
  - IMP_PCT
    - *Summary:* The mean impervious percentage derived from NLCD raster cells in a station area or corridor.
    - *Source(s):* NLCD Impervious Surface

- *Method:* Census block boundaries are overlaid on the NLCD raster estimating impervious surface percentages for a localized (30 meter x 30 meter) area. The mean impervious percentage estimate for the block is summarized for mapping; the mean impervious percentage estimate for each station area and corridor is summarized for reporting.
- *Notes:* Impervious surface coverage affects stormwater runoff volumes and non-point source pollutants. As a growth management strategy, TOD can help mitigate the loss of permeable land covers by focusing development in locations that are already impervious. This supports resiliency for transportation and other systems.

- **Floor area by development intensity**
  - DevStatus;FA_SHR
    - *Summary:* Breakdown of total floor area in each summary area by underlying development intensity: (1.) undeveloped, (2.) developed open space, (3.) developed low intensity, (4.) developed medium intensity, (5.) developed high intensity
    - *Source(s):* NLCD Impervious Surface; Census Blocks; Florida DOR
    - *Method:* Census block boundaries are overlaid on the NLCD raster estimating impervious surface percentages for a localized (30 meter x 30 meter) area. The total area of raster cells is summarized, grouping by threshold of impervious surface coverage: undeveloped = 0%; developed open space = >0% to <20%; developed low intensity = 20% to <50%; developed medium intensity = 50% to <80%; developed high intensity = 80% to 100%. Blocks also have estimates of total floor area derived form underlying parcel data. For station areas and corridors, the total floor area in each development intensity category is summarized from the block level. This allows the TOC Tool to track where building area is being added over time relative to prevailing development intensities.
    - *Notes:* Impervious surface coverage affects stormwater runoff volumes and non-point source pollutants. As a growth management strategy, TOD can help mitigate the loss of permeable land covers by focusing development in locations that are already impervious. This supports resiliency for transportation and other systems.

Regional Travel

Multimodal travel is increasing

- **Weekday boardings and alightings**
  - ON;OFF
    - *Summary:* Average weekday transit boardings and alightings in this summary area.
    - *Source(s):* Transit Ridership

        - *Method:* For a given station area or corridor, Boardings (ON) and Alightings (OFF) are summarized by time of day. Boarding/alighting instances are associated with station areas or corridors based on simple geographic overlay with observed boardings and alightings at transit stops.
        - *Notes:* NA
- **Transit utilization by time of day**
    - ON;OFF;TimePeriod
        - *Summary:* Boardings and alightings by stop location by time of day for six time periods: (1.) early morning: 3-6 AM; (2.) morning peak: 6-8:30 AM; (3.) mid-day: 8:30 AM-3 PM; (4.) afternoon peak: 3-6 PM; (5.) evening: 6-8 PM; (6.) late night: 8 PM - 3AM
        - *Source(s):* Transit Ridership
        - *Method:* For a given station area or corridor, Boardings (ON) and Alightings (OFF) are summarized by time of day. Boarding/alighting instances are associated with station areas or corridors based on simple geographic overlay with observed boardings and alightings at transit stops.
        - *Notes:* NA
- **Commutes by mode**
    - CommMode;Commutes
        - *Summary:* Share of total daily commutes by each mode: Drove alone; transit; carpool; walk/bike; other
        - *Source(s):* Florida DOR; ACS, Commute
        - *Method:* [
        - *Notes:* Non-auto mode shares are higher in TOCs (TOC Guidelines). Work-from-home is excluded from this modal breakdown since no trips are generated.
- **Non-auto mode share**
    - SHR_NONAUTO
        - *Summary:* Share of total daily commutes in this summary area made by non-auto modes (transit, walk/bike, or other)
        - *Source(s):* Florida DOR; ACS, Commute
        - *Method:* ACS commuting trips estimated at the block group scale are disaggregated to the parcel level based on land use (commutes are assumed to originate from residential parcels) and number of residential buildings. See Demographic and Employment Estimation and Allocation analytical procedure.
        - *Notes:* Non-auto mode shares are higher in TOCs (TOC Guidelines). Work-from-home is excluded from this mode share estimate since no trips are generated.

- **Daily commutes**
  - Total_Commutes
    - *Summary:* Total commute trips originating in this summary area
    - *Source(s):* Florida DOR; ACS, Commute
    - *Method:* ACS commuting trips estimated at the block group scale are disaggregated to the parcel level based on land use (commutes are assumed to originate from residential parcels) and number of residential buildings. See Demographic and Employment Estimation and Allocation analytical procedure.
    - *Notes:* NA

The transportation system provides access to jobs and essential services by multiple modes

- **Access to essential activities by travel time (by mode)**
  - Activity;TimeBin;value
    - *Summary:* Estimates of the number of activities reachable within travel time tiers. Activities assessed include: Total jobs; consumer jobs; health care jobs; school enrollment (adult learners); school enrollment (K-12); residential units. Travel time tiers are reported as: Less than 15 minutes; 15-30 minutes; 30-45 minutes; 45-60 minutes. Modes reported include: Highway, Transit, Bicycle, Walk.
    - *Source(s):* SERPM TAZ; SERPM MAZ; SERPM Transit Skims; SERPM Auto Skims; OSM Networks
    - *Method:* Travel times between SERPM MAZ and TAZ features are looked up from mode-specific skim matrices. For each zone (MAZ's for walking and biking, TAZ's for auto and transit), the number of destination-end activities (jobs, e.g.) are joined to the skim, and the number of destination activities by type are grouped into 15-minute bands (jobs with 0-15 minutes, jobs within 15-30 minutes, e.g.). See Network Analysis - Zone Skims analytical procedure.
    - *Notes:* NA

SMART Plan corridors and station areas generate fewer and shorter vehicular trips than the rest of the region

- **Daily vehicle miles of travel**
  - VMT_ALL
    - *Summary:* The expected VMT generated by trips traveling from/to zones in a summary area based on SERPM trip tables
    - *Source(s):* SERPM Auto Trip Table; SERPM Auto Skims; SERPM TAZ
    - *Method:* The SERPM trip table provides estimates of daily vehicle trips traveling among regional TAZs. The SERPM skim estimates travel times and distance among regional TAZs. For each TAZ, a baseline estimate of VMT per household and VMT per job is generated using total trip mileage indicated

by the trip tables and skims, normalized by the number of households and jobs reflected in SERPM TAZ socio-economic data. These rates are then multiplied by the TOC tool's estimate of residential units and jobs in each year to estimate total VMT from and to each TAZ. These estimates are averaged into a total VMT estimate for trips to and from each TAZ.

    - *Notes:* Transit-oriented communities contribute to improved environmental sustainability by generating fewer and shorter automobile trips (TOC Guidelines)

- **Daily vehicle trips per activity**
  - TRIPS_PER_ACT_FROM;TRIPS_PER_ACT_TO
    - *Summary:* The expected number of daily vehicle trips produced by/attracted to all zones in a summary area based on SERPM trip tables and activity estimates
    - *Source(s):* SERPM Auto Trip Table; SERPM Auto Skims; SERPM TAZ
    - *Method:* The SERPM trip table provides estimates of daily vehicle trips traveling among regional TAZs. The SERPM skim estimates travel times and distance among regional TAZs. For each destination TAZ, the estimate of the number of trips coming from other TAZs is multiplied by the estimated distance to travel from each destination, and the result is summed to yield an estimate of total VMT generated to each zone.
    - *Notes:* Transit-oriented communities contribute to improved environmental sustainability by generating fewer and shorter automobile trips (TOC Guidelines)
- **Average vehicle trip length (origin end)**
  - AVG_DIST_FROM
    - *Summary:* The mean distance traveled for each trip from a summary area.
    - *Source(s):* SERPM Auto Trip Table; [SERPM Auto Skim]
    - *Method:* The VMT estimates from each zone are divided by the total number of trips originating in the TAZ. This provides an estimate of average trip lengths for trips beginning in each TAZ. For each summary area, the average trip length estimate for underlying TAZs is reported.
    - *Notes:* Transit-oriented communities contribute to improved environmental sustainability by generating fewer and shorter automobile trips (TOC Guidelines)
- **Average vehicle trip duration (origin end)**
  - AVG_TIME_FROM
    - *Summary:* The mean time traveled for each trip from a summary area.
    - *Source(s):* SERPM Auto Trip Table; [SERPM Auto Skim]
    - *Method:* The VHT (vehicle hours of travel) estimates from each zone are divided by the total number of trips originating in the TAZ. This provides an estimate of average trip durations for trips beginning in each TAZ. For each

summary area, the average trip duration estimate for underlying TAZs is reported.
- *Notes:* NA

Local Access

SMART Plan corridors and station areas offer opportunities for trips by non-motorized modes

- **Balanced accessibility by walking between jobs and households**
  - Walk_JHBal
    - *Summary:* An index based on jobs reachable by walking and households reachable walking to indicate the extent to which a summary area offers opportunities to make complete trips by walking
    - *Source(s):* OSM Networks; SERPM MAZ; LODES
    - *Method:* Travel times between SERPM MAZ features are looked up from a walking travel time matrix developed using OpenStreetMap features. For each MAZ, the number of jobs and residential units reachable within 15 minutes is summarized. The ratio is calculated for each MAZ and averaged for all MAZs with any portion of their area within a summary area (station area or corridor). See Network Analysis - Zone Skims analytical procedure.
    - *Notes:* Most of the traffic reduction benefits of transit-oriented communities occur not because of increased transit ridership but, rather, because of increased walking for the 80% of household travel that is not commute-related (TOC Guidelines). The number and share of trips made by walking and biking depends on the number and balance of trip productions and attractions within comfortable walking and biking distance.
- **Jobs in 30 minutes by walking**
  - TotalJobsin30W
    - *Summary:* The number of jobs reachable within 30 minutes by walking (on average) from this summary area
    - *Source(s):* OSM Networks; Florida DOR; LODES
    - *Method:* Parcel-based estimated of jobs (see Demographic and Employment Estimation and Allocation analytical procedure) are summarized to the SERPM MAZ level. Travel times by walking among MAZ's are estimated using OSM walk network (see Network Analysis - Zone Skims analytical procedure). For each MAZ, the number jobs reachable within 30 minutes is estimated. For a summary area, the average jobs reachable estimates for all underlying MAZs is reported.
    - *Notes:* Most of the traffic reduction benefits of transit-oriented communities occur not because of increased transit ridership but, rather, because of increased walking for the 80% of household travel that is not commute-related (TOC Guidelines). The number and share of trips made by walking and biking depends on the number and balance of trip productions and attractions within comfortable walking and biking distance.

- **Residential units in 30 minutes by walking**
  - HHin30W
    - *Summary:* The number of residential units reachable within 30 minutes by walking (on average) from this summary area
    - *Source(s):* OSM Networks; Florida DOR; LODES
    - *Method:* Parcel-based estimated of residential units are summarized to the SERPM MAZ level. Travel times by walking among MAZ's are estimated using OSM walk network (see Network Analysis - Zone Skims analytical procedure). For each MAZ, the number units reachable within 30 minutes is estimated. For a summary area, the average units reachable estimates for all underlying MAZs is reported.
    - *Notes:* Most of the traffic reduction benefits of transit-oriented communities occur not because of increased transit ridership but, rather, because of increased walking for the 80% of household travel that is not commute-related (TOC Guidelines). The number and share of trips made by walking and biking depends on the number and balance of trip productions and attractions within comfortable walking and biking distance.
- **Balanced accessibility by biking between jobs and households**
  - Bike_JHBal
    - *Summary:* An index based on jobs reachable by biking and households reachable by biking to indicate the extent to which a summary area offers opportunities to make complete trips by biking
    - *Source(s):* OSM Networks; Florida DOR; LODES
    - *Method:* Travel times between SERPM MAZ features are looked up from a biking travel time matrix developed using OpenStreetMap features. For each MAZ, the number of jobs and residential units reachable within 15 minutes is summarized. The ratio is calculated for each MAZ and averaged for all MAZs with any portion of their area within a summary area (station area or corridor). See Network Analysis - Zone Skims analytical procedure.
    - *Notes:* Most of the traffic reduction benefits of transit-oriented communities occur not because of increased transit ridership but, rather, because of increased walking for the 80% of household travel that is not commute-related (TOC Guidelines). The number and share of trips made by walking and biking depends on the number and balance of trip productions and attractions within comfortable walking and biking distance.
- **Jobs in 30 minutes bicycling**
  - TotalJobsin30B
    - *Summary:* The number of jobs reachable within 30 minutes by bicycling (on average) from this summary area
    - *Source(s):* OSM Networks; Florida DOR; LODES
    - *Method:* Parcel-based estimated of jobs (see Demographic and Employment Estimation and Allocation analytical procedure) are summarized to the

SERPM MAZ level. Travel times by bicycling among MAZ's are estimated using OSM bike network (see Network Analysis - Zone Skims analytical procedure). For each MAZ, the number jobs reachable within 30 minutes is estimated. For a summary area, the average jobs reachable estimates for all underlying MAZs is reported.

- *Notes:* Most of the traffic reduction benefits of transit-oriented communities occur not because of increased transit ridership but, rather, because of increased walking for the 80% of household travel that is not commute-related (TOC Guidelines). The number and share of trips made by walking and biking depends on the number and balance of trip productions and attractions within comfortable walking and biking distance.

- **Residential units in 30 minutes by bicycling**
  - HHin30B
    - *Summary:* The number of residential units reachable within 30 minutes by bicycling (on average) from this summary area
    - *Source(s):* OSM Networks; Florida DOR; LODES
    - *Method:* Parcel-based estimated of residential units are summarized to the SERPM MAZ level. Travel times by bicycling among MAZ's are estimated using OSM bike network (see Network Analysis - Zone Skims analytical procedure). For each MAZ, the number units reachable within 30 minutes is estimated. For a summary area, the average units reachable estimates for all underlying MAZs is reported.
    - *Notes:* Most of the traffic reduction benefits of transit-oriented communities occur not because of increased transit ridership but, rather, because of increased walking for the 80% of household travel that is not commute-related (TOC Guidelines). The number and share of trips made by walking and biking depends on the number and balance of trip productions and attractions within comfortable walking and biking distance.
- **Miles of bike facilities**
  - Bike_Miles
    - *Summary:* Total length of bicycle facilities of any type within this summary area's boundaries.
    - *Source(s):* Miami Dade Bicycle Facilities
    - *Method:* Bicycle facility line features are intersected with summary area (station areas, corridors) polygons to summarize the total linear distance of facilities within the summary area boundary. Only the lengths of facilities that fall inside the boundary are included in their reported figure.
    - *Notes:* NA
- **Facility mileage by type**
  - Bike_Fac;Bike_Miles
    - *Summary:* Share of bicycle facility miles by facility type: Sidepath; paved path; sidewalk; bike lane; wide curb lane; paved shoulder

- *Source(s):* Miami Dade Bicycle Facilities
- *Method:* Bicycle facility line features are intersected with summary area (station areas, corridors) polygons to summarize the total linear distance of facilities within the summary area boundary. Only the lengths of facilities that fall inside the boundary are included in their reported figures, stratified by facility type.
- *Notes:* NA

SMART Plan transit stations are reachable by walking from most parcels in the station area

- **Percent of parcels within 15-minute walk of transit station**
  - Prop_Stn15
    - *Summary:* For all parcels in this summary area, the proportion that are within 15-minute walk of a station.
    - *Source(s):* Florida DOR; OSM Networks
    - *Method:* The OpenStreetMap walk network is used to measure travel time from each parcel to each SMART Plan station. Parcels are assigned walk time values in 5 minute intervals up to 30 min (this supports bar charts of parcels in each time interval). For a given summary area (station area or corridor), the total number of parcels and the number within 15 minutes of a transit station are summarized. The latter number is then divided by the former to get the share of parcels within 15 minutes of a station area.
    - *Notes:* When deciding whether to use transit, one of the most important factors people consider is the distance between their origin and a transit passenger facility (stop, exchange, or station) and again to their destination. What matters for the traveler is not the straight-line or 'as the crow flies' distance but, rather, the actual walking distance using the available streets and paths. In an area with long blocks and dead-end streets, the walking distance can be much further than the straight-line distance. Some destinations that are physically very close to a transit stop or station may still require a long walk. (TOC Guidelines)
- **Walk time to transit stations**
  - MinTimeStn_walk
    - *Summary:* The average walk time from each parcel to the nearest SMART Plan transit station
    - *Source(s):* Florida DOR; OSM Networks
    - *Method:* The OpenStreetMap walk network is used to measure travel time from each parcel to each SMART Plan station. Each parcel's walk time to the nearest (in terms of network distance) station is recorded. The average time to walk from parcels to the nearest station is then summarized for each summary area (station area or corridor).
    - *Notes:* When deciding whether to use transit, one of the most important factors people consider is the distance between their origin and a transit

passenger facility (stop, exchange, or station) and again to their destination. What matters for the traveler is not the straight-line or 'as the crow flies' distance but, rather, the actual walking distance using the available streets and paths. In an area with long blocks and dead-end streets, the walking distance can be much further than the straight-line distance. Some destinations that are physically very close to a transit stop or station may still require a long walk. (TOC Guidelines)

- **Walking route directness to transit stations**
  - DirIdxStn_walk
    - *Summary:* The minimum walk time to a transit station relative to the hypothetical minimum time
    - *Source(s):* Florida DOR; OSM Networks
    - *Method:* A hypothetical walk time from each parcel to each SMART Plan station is calculated based on the simple linear distance between the parcel and the station location and an assumed walking speed of 3 miles per hour. The network-based walk time between each parcel and each station (see Minimum walk time to transit stations) is then divided by the hypothetical walk time to assess the directness of the connection. Higher values indicate less direct connections, meaning the walk network offers only circuitous routings between parcels and stations.
    - *Notes:* When deciding whether to use transit, one of the most important factors people consider is the distance between their origin and a transit passenger facility (stop, exchange, or station) and again to their destination. What matters for the traveler is not the straight-line or 'as the crow flies' distance but, rather, the actual walking distance using the available streets and paths. In an area with long blocks and dead-end streets, the walking distance can be much further than the straight-line distance. Some destinations that are physically very close to a transit stop or station may still require a long walk. (TOC Guidelines)

Parks and public spaces are reachable by walking from most parcels in the station area

- **Park acreage**
  - Park_Acres
    - *Summary:* Total park acreage per capita within a summary area
    - *Source(s):* Parks
    - *Method:* Within a summary area (station area or corridor), total park acreage is summarized and divided by the estimated population. See Demographic and Employment Estimation and Allocation analytical procedure
    - *Notes:* At the neighborhood scale, a mix of land uses (including parks) in close proximity creates an environment where many needs of daily life can be met within a short walk from home, work or transit (TOC Guidelines)

- **Share of parcels within 15-minute walk of public spaces**
  - Prop_Walk15
    - *Summary:* For all parcels in a given area, the proportion that are within 15-minute walk of any park/public space facility.
    - *Source(s):* Florida DOR; OSM Networks
    - *Method:* The OpenStreetMap walk network is used to measure travel time from each parcel to each park location. Parcels are assigned walk time values in 5 minute intervals up to 30 min (this supports bar charts of parcels in each time interval). For a given summary area (station area or corridor), the total number of parcels and the number within 15 minutes of a transit station are summarized. The latter number is then divided by the former to get the share of parcels within 15 minutes of a station area.
    - *Notes:* NA
- **Walk time to public spaces**
  - BinStn_walk;NO_RES_UNTS
    - *Summary:* The average walk time from each parcel to a park/public space facility
    - *Source(s):* Florida DOR; OSM Networks
    - *Method:* The OpenStreetMap walk network is used to measure travel time from each parcel to each park facility. Each parcel's walk time to the nearest (in terms of network distance) park is recorded. The average time to walk from parcels to the nearest station is then summarized for each summary area (station area or corridor).
    - *Notes:* NA
- **Walking route directness to public spaces**
  - DirIdxPark_walk
    - *Summary:* The minimum walk time from each parcel to any park/public space facility relative to the hypothetical minimum time
    - *Source(s):* Florida DOR; OSM Networks
    - *Method:* A hypothetical walk time from each parcel to each park facility is calculated based on the simple linear distance between the parcel and the park location and an assumed walking speed of 3 miles per hour. The network-based walk time between each parcel and each park (see Minimum walk time to parks) is then divided by the hypothetical walk time to assess the directness of the connection. Higher values indicate less direct connections, meaning the walk network offers only circuitous routings between parcels and parks.
    - *Notes:* NA

Data Inventory

Miami Dade Bicycle Facilities

- **Description:** A polyline feature class of bicycle facilities by type within Miami-Dade County.
- **Type:** Spatial
- **Shape Type:** Line
- **Geographic Scope:** MD County
- **Spatial Unit:** Segment
- **Latest Vintage:** latest available data
- **Date Published:** 43619
- **Update Schedule:** Annually
- **Publisher/Provider:** Miami Dade County
- **Mechanism to Obtain:** Download:
  - Bike_Lanes: https://opendata.arcgis.com/datasets/b874dd0e2d0941a689c56f54ae72d67c_0.geojson,
  - Paved_Path: https://opendata.arcgis.com/datasets/5ee76f3de89a4510871f7943ee20a80d_0.geojson,
  - Paved_Shoulder: https://opendata.arcgis.com/datasets/76ac2f796a6341a6b5d45f66e42788d1_0.geojson”,
  - Wide_Curb_Lane: https://opendata.arcgis.com/datasets/b0b330209d244850ae5f89768edc3271_0.geojson
- **Key Fields:** FacID: ID for each row Bike_Fac: bike facility type Bike_Miles: mileage of each line segment/row in dataset
- **Related Tables:** (None)
- **Quality Concerns:** (None)

Miami Dade County Boundary

- **Description:** A polygon feature class of the Miami-Dade County outer boundaries.
- **Type:** Spatial
- **Shape Type:** Polygon
- **Geographic Scope:** MD County
- **Spatial Unit:** County
- **Latest Vintage:** latest available data
- **Date Published:** 43417
- **Update Schedule:** none planned
- **Publisher/Provider:** Miami Dade County

- **Mechanism to Obtain:** Download: https://opendata.arcgis.com/datasets/cec575982ea64ef7a11e587e532c6b6a_0.geojson
- **Key Fields:** (None)
- **Related Tables:** (None)
- **Quality Concerns:** (None)

Urban Development Boundary

- **Description:** A polygon feature class of Miami-Dade County's Urban Development Boundary(UDB) boundaries. The boundary was adopted by the Board of County Commissioners (BCC) as per recommendation. This boundary identifies the area where urban development may occur through the year 2020. Development orders permitting urban development will generally be approved within the UDB at some time through the year 2020 provided that level-of-service standards for necessary public facilities will be met. Adequate countywide capacity will be maintained within the UDB by increasing development densities or intensities inside the UDB or by expanding the UDB when the need for such change is determined to be necessary through the amendment process.
- **Type:** Spatial
- **Shape Type:** Polyline
- **Geographic Scope:** MD County
- **Spatial Unit:** County
- **Latest Vintage:** latest available data
- **Date Published:** 41655
- **Update Schedule:** Annually
- **Publisher/Provider:** Miami Dade County
- **Mechanism to Obtain:** Download: https://opendata.arcgis.com/datasets/a468dc11c02f4467ade836947627554b_0.geojson
- **Key Fields:** (None)
- **Related Tables:** (None)
- **Quality Concerns:** Data come as a line feature and are processed with the MD County boundary polygon to generate a polygon feature class. The assumes all area within the created polygon are within the UDB

Census Blocks

- **Description:** Statistical areas bounded by visible features such as roads, streams, and railroad tracks, and by nonvisible boundaries such as property lines, city, township, school district, county limits and short line-of-sight extensions of roads.
- **Type:** Spatial
- **Shape Type:** Polygon
- **Geographic Scope:** US, limited to Miami-Dade County
- **Spatial Unit:** Census block
- **Latest Vintage:** latest available data

- **Date Published:** 43685
- **Update Schedule:** once per decade
- **Publisher/Provider:** US Census
- **Mechanism to Obtain:** Download: utilizing the Census FTP site and custom python script to automatically pull the geometries.
- **Key Fields:** GEOID10
- **Related Tables:** ACS, LODES: GEOID10
- **Quality Concerns:** NA

ACS, Commute

- **Description:** Table of commute (journey to work) data [table B08301] by Census block group
- **Type:** Tabular
- **Shape Type:** NA
- **Geographic Scope:** US, limited to MD County
- **Spatial Unit:** Block group
- **Latest Vintage:** 2018 (latest available data)
- **Date Published:** 43818
- **Update Schedule:** 5-year ACS tables are released with a 1 year lag (e.g. 2019 5-year tables were released 12/10/2020)
- **Publisher/Provider:** US Census
- **Mechanism to Obtain:** Downloaded via the US Census API using the "censusdata" python module
- **Key Fields:** GEOID10
- **Related Tables:** BlockGroup: GEOID10
- **Quality Concerns:** NA

ACS, Race

- **Description:** Table of population race and ethnicity variables [table B03002] by Census block group
- **Type:** Tabular
- **Shape Type:** NA
- **Geographic Scope:** US, limited to MD County
- **Spatial Unit:** Block group
- **Latest Vintage:** 2018 (latest available data)
- **Date Published:** 43818
- **Update Schedule:** 5-year ACS tables are released with a 1 year lag (e.g. 2019 5-year tables were released 12/10/2020)
- **Publisher/Provider:** US Census
- **Mechanism to Obtain:** Downloaded via the US Census API using the "censusdata" python module

- **Key Fields:** GEOID10
- **Related Tables:** BlockGroup: GEOID11
- **Quality Concerns:** NA

LODES

- **Description:** A table of employment in each of 20 NAICS sectors by Census block group
- **Type:** Tabular
- **Shape Type:** NA
- **Geographic Scope:** US, limited to MD County
- **Spatial Unit:** Census block
- **Latest Vintage:** 2017 (latest available)
- **Date Published:** originally uploaded 2019
- **Update Schedule:** data for year y become available in year y+2 (e.g. 2018 was released at the end of 2020)
- **Publisher/Provider:** Census LEHD (Longitudinal Employer-Household Dynamics)
- **Mechanism to Obtain:** Download: programmatically from archive at: https://lehd.ces.census.gov/data/lodes/LODES7/
- **Key Fields:** GEOID10: block identifier; CNSXX: estimated employment in NAICS sector XX
- **Related Tables:** NA
- **Quality Concerns:** data is aggregated up from the blocks to the block group to combat inaccuracies of estimation at such a small geographic level

Florida DOR

- **Description:** A set of tax assessor use parcels
- **Type:** Spatial
- **Shape Type:** Polygon
- **Geographic Scope:** MD County
- **Spatial Unit:** Parcel
- **Latest Vintage:** 2019
- **Date Published:** 44054
- **Update Schedule:** Final parcel data are published yearly in October after certification
- **Publisher/Provider:** Florida Department of Revenue
- **Mechanism to Obtain:** Download:
  - Map Data: ftp://sdrftp03.dor.state.fl.us/Map%20Data/
  - Attributes: ftp://sdrftp03.dor.state.fl.us/Tax%20Roll%20Data%20Files/
- **Key Fields:** PARCEL_ID, DOR_UC
- **Related Tables:** Shape and Attributes are stored separately but are downloaded joined in the same script
- **Quality Concerns:** Previous year data is only available upon request, additionally for automating the URLs on their ftp do not follow common naming conventions.

NLCD Impervious Surface

- **Description:** A raster of percent imperviousness
- **Type:** Spatial
- **Shape Type:** Raster
- **Geographic Scope:** Contiguous 48 states, limited to MD County
- **Spatial Unit:** ~30m x 30m grid cell
- **Latest Vintage:** 2016 (latest available)
- **Date Published:** January of 2019
- **Update Schedule:** every 5 years, with ~3 year lag (i.e. 2021 data should be expected in 2024)
- **Publisher/Provider:** Multi-Resolution Land Characteristics (MRLC) Consortium
- **Mechanism to Obtain:** Download: programmatically from download links available at: https://www.mrlc.gov/data
- **Key Fields:** (None)
- **Related Tables:** (None)
- **Quality Concerns:** (None)

OSM Buildings

- **Description:** A polygon feature class of the building footprints
- **Type:** Spatial
- **Shape Type:** Polygon
- **Geographic Scope:** World, limited to MD county
- **Spatial Unit:** building
- **Latest Vintage:** OSM pulls are current as of the download date
- **Date Published:** OSM is open source, so the data used has no fixed pub date
- **Update Schedule:** OSM is open source, so the data can be updated at any time
- **Publisher/Provider:** OpenStreetMap (OSM)
- **Mechanism to Obtain:** Downloaded through the OSM API available in python
- **Key Fields:** NA
- **Related Tables:** NA
- **Quality Concerns:** OSM is open source, so there's always the potential for missing/inaccurate data; however, reliability is historically high in urban areas such as MDC

OSM Networks

- **Description:** A line feature class of streets
- **Type:** Spatial
- **Shape Type:** Polyline
- **Geographic Scope:** World, limited to MD county
- **Spatial Unit:** street

- **Latest Vintage:** OSM pulls are current as of the download date
- **Date Published:** OSM is open source, so the data used has no fixed pub date
- **Update Schedule:** OSM is open source, so the data can be updated at any time
- **Publisher/Provider:** OpenStreetMap (OSM)
- **Mechanism to Obtain:** NA
- **Key Fields:** highway: road type (motorway, primary, residential, etc.)
- **Related Tables:** NA
- **Quality Concerns:** OSM is open source, so there's always the potential for missing/inaccurate data; however, reliability is historically high in urban areas such as MDC

Parks

- **Description:** Park polygon features as well as park facilities as point locations
- **Type:** Spatial
- **Shape Type:** Polygon and Point
- **Geographic Scope:** MD County
- **Spatial Unit:** Park
- **Latest Vintage:** latest available data
- **Date Published:** Facilities-10/22/20 County Parks - 4/3/20 National/State Parks 8/20/19 Municipal - 3/26/19
- **Update Schedule:** NA BI-Annually Annually Bi-Annually
- **Publisher/Provider:** MiamiDade County
- **Mechanism to Obtain:** Download:
  - muni_url = https://opendata.arcgis.com/datasets/16fe02a1defa45b28bf14a29fb5f0428_0.geojson
  - county_url = https://opendata.arcgis.com/datasets/aca1e6ff0f634be282d50cc2d534a832_0.geojson
  - state_fed_url = https://opendata.arcgis.com/datasets/fa11a4c0a3554467b0fd5bc54edde4f9_0.geojson
  - park_fac_url = https://opendata.arcgis.com/datasets/8c9528d3e1824db3b14ed53188a46291_0.geojson
- **Key Fields:** NA
- **Related Tables:** NA
- **Quality Concerns:** NA

Building Permits

- **Description:** Building permit data from the Miami Dade Dept of Regulatory and Economic Resources Road Impact Fee database.
- **Type:** Tabular
- **Shape Type:** Table
- **Geographic Scope:** MD County
- **Spatial Unit:** Parcel
- **Latest Vintage:** 2019
- **Date Published:** 44113
- **Update Schedule:** Annually, via direct request
- **Publisher/Provider:** RER
- **Mechanism to Obtain:** Direct Coordination: Contact with Keith Richardson of Miami-Dade, Department of Regulatory and Economic Resources; Impact Fee Section and request a CSV Road Impact Fee Collection Report for most recent year. Place data in the PMT/Data/Raw folder for further processing.
- **Key Fields:** PROC_NUM, FOLIO_NUM, SITE_ADDR, ASSE_DATE, STATUS_CODE, COL_CON, COL_ADM, ASD_CATTHRES_CATC_CODE, ASSD_BASSIS_QTY
- **Related Tables:** Parcels: PARCLNO
- **Quality Concerns:** NA

Transit Ridership

- **Description:** Daily Transit ridership by time of day including boardings and alightings at each location, averaged for the month of April for consistent year-over-year comparisons (from Dept of Transportation and Public Works)
- **Type:** Tabular
- **Shape Type:** Table
- **Geographic Scope:** MD County
- **Spatial Unit:** Point
- **Latest Vintage:** 2020 (April)
- **Date Published:** NA
- **Update Schedule:** Annually, via direct request
- **Publisher/Provider:** DTPW
- **Mechanism to Obtain:** Direct Coordination: Contact with Esther Frometa-Spring of Miami-Dade, Department of Transportation and Public Works. The report may require DTPW staff to coordinate with their consultant team.
- **Key Fields:** DAY_OF_WEEK, TIME_PERIOD, ROUTE, DIRECTION, UNIQUE_STOP_NO, STOP_NAME, ON, OFF, Year
- **Related Tables:** NA
- **Quality Concerns:** NA

SERPM Auto Skims

- **Description:** A matrix in the FSUTMS standard estimating travel times and distances among TAZs for auto (SOV, HOV, etc.) trips
- **Type:** Tabular
- **Shape Type:** Table
- **Geographic Scope:** Southeast Florida Regional Planning Model area (Miami-Dade, Broward, and Palm Beach Counties)
- **Spatial Unit:** TAZ
- **Latest Vintage:** v. 8.0
- **Date Published:** 2020
- **Update Schedule:** NA
- **Publisher/Provider:** FDOT
- **Mechanism to Obtain:** Download SERPM model inputs from FSUTMS Online; run SERPM model to generate outputs; export auto skim to csv table format for PMT processing
- **Key Fields:** NA
- **Related Tables:** TAZ
- **Quality Concerns:** NA

SERPM Transit Skims

- **Description:** A matrix in the FSUTMS standard estimating travel times and distances among TAZs for transit trips
- **Type:** Tabular
- **Shape Type:** Table
- **Geographic Scope:** Southeast Florida Regional Planning Model area (Miami-Dade, Broward, and Palm Beach Counties)
- **Spatial Unit:** TAZ
- **Latest Vintage:** v. 8.0
- **Date Published:** 2020
- **Update Schedule:** NA
- **Publisher/Provider:** FDOT
- **Mechanism to Obtain:** Download SERPM model inputs from FSUTMS Online; run SERPM model to generate outputs; export transit skim to csv table format for PMT processing
- **Key Fields:** NA
- **Related Tables:** TAZ
- **Quality Concerns:** NA

SERPM Auto Trip Table

- **Description:** A matrix in the FSUTMS standard estimating trips made by personal vehicles among TAZs
- **Type:** Tabular
- **Shape Type:** Table

- **Geographic Scope:** Southeast Florida Regional Planning Model area (Miami-Dade, Broward, and Palm Beach Counties)
- **Spatial Unit:** TAZ
- **Latest Vintage:** v. 8.0
- **Date Published:** 2020
- **Update Schedule:** NA
- **Publisher/Provider:** FDOT
- **Mechanism to Obtain:** Download SERPM model inputs from FSUTMS Online; run SERPM model to generate outputs; export auto trip table to csv table format for PMT processing
- **Key Fields:** NA
- **Related Tables:** TAZ
- **Quality Concerns:** NA

SERPM MAZ

- **Description:** SERPM inputs that include geographic features (polygons) representing micro analysis zones, a sub-TAZ geography populated with rich demographic and economic details in related tables (csv).
- **Type:** Spatial
- **Shape Type:** Polygon
- **Geographic Scope:** Southeast Florida Regional Planning Model area (Miami-Dade, Broward, and Palm Beach Counties)
- **Spatial Unit:** MAZ
- **Latest Vintage:** v. 8.0
- **Date Published:** 2020
- **Update Schedule:** NA
- **Publisher/Provider:** FDOT
- **Mechanism to Obtain:** Download SERPM model inputs from FSUTMS Online
- **Key Fields:** mgra; TAZ
- **Related Tables:** NA
- **Quality Concerns:** NA

SERPM TAZ

- **Description:** SERPM inputs that include geographic features (polygons) representing travel analysis zones. Demographic and economic data may be summarized from the MAZ scale to the TAZ scale.
- **Type:** Spatial
- **Shape Type:** Polygon
- **Geographic Scope:** Southeast Florida Regional Planning Model area (Miami-Dade, Broward, and Palm Beach Counties)
- **Spatial Unit:** TAZ

- **Latest Vintage:** v. 8.0
- **Date Published:** 2020
- **Update Schedule:** NA
- **Publisher/Provider:** FDOT
- **Mechanism to Obtain:** Download SERPM model inputs from FSUTMS Online
- **Key Fields:** TAZ
- **Related Tables:** NA
- **Quality Concerns:** NA

ANALYTICAL PROCEDURES

Demographic and Employment Estimation and Allocation

Allocation of Census data to parcels for the TOC Tool requires two broad steps. The first is estimating Census data at the block group level. The second is applying rules based on parcel land uses to estimate parcel-level values from block group-level totals.

Census data to be allocated includes the following activity categories:

- Population data
    - *Description*: population counts by race and ethnicity
    - *Source*: American Community Survey (ACS)
    - *Level*: block group
    - *PMT years available at publication*: 2014 - 2018
    - *PMT years unavailable at publication*: 2019
- Employment data
    - *Description*: employment counts by NAICS sector
    - *Source*: LEHD Origin-Destination Employment Statistics (LODES)
    - *Level*: block (allocated up to the block group for consistency with ACS data)
    - *PMT years available at publication*: 2014 - 2017
    - *PMT years unavailable at publication*: 2018 - 2019
- Commute data
    - *Description*: commute counts by mode
    - *Source*: American Community Survey (ACS)
    - *Level*: block group
    - *PMT years available at publication*: 2014 - 2018
    - *PMT years unavailable at publication*: 2019

Once the requisite data is collected, block group-level estimation proceeds according to the steps below.

- Modeling
    - Calculate total population, total employment, and total commutes in observed years by summing over the relevant activity sub-categories (racial/ethnic subgroups, NAICS sectors, and commute modes, respectively).
    - Fill any missing values for the above calculations with "0"; this indicates that block group had no population, employment, and/or commutes.

- Using a fixed set of explanatory variables, fit linear models to predict total population, employment, and commutes across the set of observed years.
    - For each of the three response variables, the predictive model includes all explanatory variables that showed a statistically significant correlation – positive or negative – with the response.
- Calculating shares
  - Fill any missing values with "0" for all population, employment, and commute sub-categories.
  - For each activity category, calculate the block group share for the activity sub-categories in each year by dividing the sub-category total by the category total ("drove alone" commutes as a share of all commutes, e.g.).
    - If any block groups have no data for a given category, try to take the block group share for the sub-categories as the mean of the block group share for the sub-categories in all neighboring block groups that had data for that category.
    - If any block groups still have no data for a given category after trying the above, take the block group share for the sub-categories as the block group share for the sub-categories in the nearest block group with data.
- Forecasting shares
  - For un-observed years of each category, apply a "constant-shares" approach to estimate sub-category shares in unobserved years: the shares stay the same as they were in the most recently observed year (refer to the data section to see what years this affects).
- Estimating
  - Apply the linear models to estimate total population, employment, and commutes in **all** PMT years.
    - Note that model results are used *even in observed years* to improve the year-over-year consistency of activity estimates.
  - Multiply the category estimate for a year by the block group shares for the sub-categories in that year to get the final results.

The outputs of the estimation process are block-group level estimates of all activity categories and sub-categories in all PMT years. With this data in place, allocation proceeds according to the steps below. The steps describe the process of allocation for one year only, but the same process is used to allocate from the block group to the parcel for all years.

- Spatial processing
  - Convert the parcels to centroids
  - Intersect the parcel centroids with the block group estimation results to match parcels (and their attributes) to a single block group
- Initializing allocation
  - For each sub-category, identify land uses that *would be expected* to house units of that sub-category. This is called the "Land Use Mask"
    - e.g. population and commutes should only associated with residential land uses
    - e.g. jobs in Agriculture should primarily be associated with agricultural land uses
  - Make a secondary mask called the "Non-residential Mask", which comprises all land uses in the "Land Use Mask" for any employment sub-category.

- Make a tertiary mask called the "All-developed Mask", which comprises all land uses in the "Land Use Mask", regardless of category.
- For each sub-category in each block group, estimate a "block group total" of available building area by summing the building area in all parcels with relevant land uses according to the "Land Use Mask". **It is possible that the Land Use Mask will fail because it sub-category-specific land uses are not found**. So, if this results in 0 building area available for a sub-category, follow through the steps below until some amount of area is obtained:
  - *For employment sub-categories only*, sum building area in the block group for all parcels with land uses in the "Non-residential Mask". This sort of search is not relevant for population or commutes sub-categories, because the initial search occurs over all residential areas; for residence-based categories, begin at the step below.
  - If there is still no building area estimated, try to sum building area in the block group for all parcels with land uses in the "All-developed Mask".
  - If there is still a building area estimate of 0.0, sum building area in the block group for all parcels, regardless of land use.
  - If there is still a building area estimate of 0.0, total land area in the block group across all parcels, regardless of land use. This step is implemented as a last resort, and is guaranteed to work since all parcels have land area (actual instances of land-area-based allocation are rare).
- Depending on what step above yielded the block group total, identify the "eligible parcels" that are candidates to receive units of each sub-category in each block group
  - e.g. if use of the Land Use Mask produced the block group total for block group A for NAICS sector Agriculture, only parcels with agricultural land uses as specified by the Mask are eligible to receive Agriculture jobs in block group A

- Allocating
  - For each sub-category in each block group, divide the area (building or land, depending on how the block group total was calculated) of each eligible parcel by the block group total to get a "share" for each parcel
    - Ineligible parcels are given a share of 0
  - Multiply the block group estimate for a sub-category by the sub-category share in that block group to achieve a parcel level estimate.

The results of the allocation process are parcel-level estimates of all activity categories and sub-categories in all PMT years. These estimates are used to inform localized analysis and visualizations of demographic data in Miami-Dade County.

Developable Area and Contiguity

Contiguity and developable area for a parcel are assessed based on the *Contiguity Index*, or $CONTIG$, landscape metric developed and released by the University of Massachusetts Landscape Ecology Lab as a part of the *FRAGSTATS* software . In the context of urban development, contiguity can be interpreted as the degree to which non-developed land is contiguous; developable area can be interpreted as the total area of non-developed land

For more information on *FRAGSTATS*, see the *FRAGSTATS* homepage. For mathematical specifics and a detailed description of the $CONTIG$ metric, see the $CONTIG$ documentation.

For the PMT, contiguity and developable area calculations requires two datasets:

- parcel boundaries (obtained from FDOR)
- building polygons (obtained from OSM)

Once these data are collected, contiguity and developable area are calculated according to the following procedure:

- Take the spatial difference of the parcels and the buildings. This yields a shape of *all* un-built, or developable, area. Each polygon in the resulting shape will be tagged with a parcel ID that can be used to assign an area to a parcel.
- Split any multi-part polygons in the difference shape into unique single-part polygons. The unique parts of a multi-part polygon are inherently non-contiguous, so $CONTIG$ will have to be calculated for each individual polygon and summarized back up to the parcel level. Each polygon in the resulting shape will be tagged with a sub-parcel ID that can be used to assign a polygon to a parcel.
- Rasterize the single-part difference shape with a user-defined cell size. The values of these raster cells will be either a sub-parcel ID or a null value (meaning that cell is not developable).
  - This cell size should be large enough that a new building could reasonably fit within a single cell. As a default, the cell size is 40 feet (associated with a 1600 square foot building footprint).
- Count the number of cells with sub-parcels belonging to each parcel, and multiply by the cell size. This is the **developable area** of a parcel.
- For each raster cell, identify its "valid neighbors". Valid neighbors meet the following criteria:
  - The neighboring cell must be developable.
  - The neighboring cell must belong to the same sub-parcel as the cell of interest.
- Using a pre-set weighting system, sum the weight for each cell based on it's neighbors to get a "weight total".
  - The weighting system is intended to quantify the value of being contiguous to a particular neighbor (the value of the cell itself is 1 regardless of system). As a default, contiguity in the development context is assessed with *Nearest neighbors (NN)* weighting, where all neighbors are worth 1. A user is allowed to specify whatever system they want, but some other common options include:
    - *Rook*: horizontal and vertical neighbors are worth 1, diagonal neighbors are worth 0
    - *Queen*: horizontal and vertical neighbors are worth 2, diagonal neighbors are worth 1
- Sum the weight totals by sub-parcel ID, and divide this by the count of cells with that sub-parcel ID. This is become the basis for the $CONTIG$ calculation.
- *Complete the calculation of* $CONTIG$ *for a sub-parcel by subtracting 1 from the value calculated above, and dividing by the maximum weight total minus 1 (which, in the case of the default is 9-1 = 8).*
- *Using a pre-set fun*ction, summarize the sub-parcel contiguity scores to the parcel. This is the **contiguity index** of a parcel.
  - This can be any function that return as a scalar value as a summary of a set of values. As a default, *mean*, *median*, *minimum*, and *maximum* are all used to summarize sub-parcel contiguity to the parcel, but the *median* is likely the most robust single value in the development context.

- As an added step, a user could multiply the final contiguity index by developable area to get a "value of developable area" metric.
  - This is a natural extension because $CONTIG$ is naturally constrained to [0,1] to define minimum and maximum contiguity.
  - Note, however, that this approach assumes that the value of developing a parcel is dependent on the contiguity of *all* developable area, not a subset of interest.

The result of this analysis are parcel-level assessments of contiguity of developable area and total developable area. These metrics are used to inform localized analysis and visualizations of development potential in Miami-Dade County.

Network Analysis - Zone Skims

Many TOC-related metrics focus on network connectivity and access to jobs, housing, and other trip generators by various modes.

For motorized modes (auto and transit), the Southeast Florida Regional Planning Model (SERPM) is used to generate matrices of travel times and distances among travel analysis zones (TAZ). In SERPM, OD data for transit travel opportunities are reported for transit access points (TAPs). To get TAZ-to-TAZ data, two sub-networks are created from the OD matrix: the TAP-to-TAP matrix from SERPM and a TAZ-to-TAP matrix using OSM walk networks (see non-motorized modes below). These sub-networks are combined and solved to estimate the shortest walk-access/egress-to-transit travel times among TAZs.

For non-motorized travel modes (walking and biking), digital representations of the local network are obtained from OpenStreetMap using the OSMNX python package. These data are used to construct mode-specific network datasets using ESRI's network template tools. These network datasets are then used to evaluate travel times among micro-analysis zones (MAZ) features, which are smaller than TAZ's, using ESRI's OD cost matrix network solver.

These origin-destination (OD) matrices are used to estimate the number of activities reachable from a given zone, where destination-end activities (jobs, e.g.) are related to the matrix and summarized for each zone of origin. For example, if $ZoneA$ can reach $ZoneB$ and $ZoneC$, and $ZoneB$ has 20 jobs and $ZoneC$ has 30 jobs, then $ZoneA$ can reach 50 jobs. These summaries are generated for each respective mode using consistent travel time ranges for reporting: 0-15 minutes, 15-30 minutes, 30-45 minutes, and 45-60 minutes.

Additionally, non-motorized networks are used to generate service areas defining the geographic areas and individual street segments within 30 minutes of parks and transit stations areas using ESRI's service area network solver. Parcel features are intersected with the resulting service areas to approximate the time it takes to walk from each parcel to the nearest park and to the nearest rapid transit station. This allows parcel attributes to be summarized in terms of proximity to transit and park spaces (floor area by walk time to transit, e.g.).

APPENDIX B: SCRIPT DOCSTRINGS

INTRODUCTION AND SETUP

Transit Oriented Communities (TOC) are a major focus of Miami-Dade County's regional growth management and transportation strategy. TOCs focus new residential, commercial, office and institutional investments in areas served by premium transit. The site created by these tools provide helpful information on TOC development in the county across four key topics: economic development, urban form, regional travel, and local access.

The scripts download data resources and generate new data to monitor development, transportation infrastructure, and travel patterns over time. The following project documents the environment, scripts and usage for the Miami-Dade TPO's TOD Performance Monitoring Toolkit.

INSTALLATION AND AUTOMATED DOWNLOADS

Environment Setup

Assumptions

- ArcGIS Pro is installed in a standard location (ex: C:Program FilesArcGISPro)
- ArcGIS Pro version >= 2.7
- Familiarity with command line interactions

Building python CONDA environment

`pmt_download (used for download procedures only)`

Manual Install

1. Select Windows Start
2. Navigate to 'Python Command Prompt' under ArcGIS folder –> Open
3. In the command window run the below commands:
    a. Create a new environment

    ```
    conda create --name %LocalAppData%\\ESRI\\conda\\envs\\pmt_downlo
    ad
    ```

    b. Activate the newly created environment

    ```
    activate pmt_download
    ```

    c. Install conda packages

    ```
    conda install geopandas pandas numpy scipy rtree=0.9.4
    * enter 'y/yes' and return when asked
    ```

    d. Install pip packages

```
pip install censusdata osmnx
```

Semi-Automated Install

Assumes Anaconda or Miniconda is installed

1. Select Windows Start
2. Navigate to or search for Anaconda Prompt
3. Change directory to pmt_code project

   ```
   cd /path/to/code
   ```

4. Run the following commands

   ```
   conda install -c conda-forge mamba
   mamba env create -f environment_download.yml
   ```

pmt_tools

Manual Install

1. Select Windows Start
2. Navigate to 'Python Command Prompt' under ArcGIS folder –> Open
3. In the command window run the below commands:
   a. Clone the existing ArcGIS python default environment (*arcgispro-py3*)

      ```
      conda create --clone arcgispro-py3 --name %LocalAppData%\\ESRI\\c
      onda\\envs\\pmt_tools
      ```

   b. Activate the newly created environment

      ```
      activate pmt_tools
      ```

   c. Install conda packages using **conda-forge** channel

      ```
      conda install -c conda-forge momepy sphinx dask
      * enter 'y/yes' and return when asked
      * spyder is optional if you want to install a Data Science focuse
      d IDE
      ```

   d. Install pip packages

      ```
      pip install simpledbf
      ```

Semi-Automated Install

Assumes Anaconda or Miniconda is installed

1. Select Windows Start
2. Navigate to or search for Anaconda Prompt
3. Change directory to pmt_code project

   ```
   cd /path/to/code_dir
   ```

4. Run the following commands

```
conda install -c conda-forge mamba
mamba env create -f environment_processing.yml
```

WARNING:¶

If you have recently updated ArcGIS Pro to a new Major Version, you will need to remove the existing environment and recreate it using steps 4-7 again if using the manual install method. If using the semi-automated method, check that the python version in environment_process.yml matches the installation of Pro

- Remove env

  conda env remove -n pmt_tools

- Follow the above steps to recreate the environment

pmt_docs (used for generating documentation updates due to process enhancements or code changes)

Semi-Automated Install

Assumes Anaconda or Miniconda is installed

1. Select Windows Start
2. Navigate to or search for Anaconda Prompt
3. Change directory to pmt_code project

   ```
   cd /path/to/code_dir
   ```

4. Run the following commands

   ```
   conda install -c conda-forge mamba
   mamba env create -f environment_docs.yml
   ```

Tool Usage

Download Tools

1. follow steps 1-3 of manual environment setup processes to open correct command prompt
2. open PMT_toolsconfigdownload_config.py
   - verify all existing configuration variables are ready to use
3. open PMT_toolsutils.py
   - verify DATA_ROOT variable is set correctly (todo: allow setting DATA_ROOT in executable)
4. activate pmt_download environment

   ```
   conda activate pmt_download
   ```

5. run downloader script

   ```
   Usage: python downloader.py
   ```

   ```
    download all available datasources automagically and place them in the
    RAW folder by data category
   ```

```
If flags are provided, individual download procedures will be run
 -s: setup_download_folder is run, building the base folder structure (
--setup)
 -u: download_urls is run, grabbing all data available directly from a
URL endpoint (--urls)
 -o: download_osm_data is run, pulling osm_networks and osm_builidng_fo
otprints (--osm)
 -g: download_census_geo is run, pulling census geography data used in
the tool (--census_geo)
 -c: download_commutes_data is run, pulling commute data for the tool (
--commutes)
 -r: download_race_data is run, pulling race data for the tool (--race)
 -l: download_lodes_data is run, pulling jobs data for the tool (--lode
s)

Example Usage:

    python downloader.py -s -u [setup download folder and download url
endpoints]****
```

TOC Tool Configuration Scripts

Download Configuration

The download_config module specifies data endpoints for automated download of TOC tool supporting data. Some variables defined by the module also specified key fields to retain and renaming dictionaries to make downloaded data more intuitive and legible.

Prepare Configuration

The prepare_config module establishes settings, field references, analysis constants, etc. to facilitate data cleaning and analysis via the preparer module. No methods or functions are defined, but some settings are assigned as custom classes defined in PMT.

Build Configuration

The build_config module defines how feature classes are overlaid, summarized, and reported in the TOC tool. All analysis and reporting executed by builder is driven by the settings defined here. Most variables defined in this module focus on one of the following objectives:

- Providing table or feature class specifications (path, key fields, etc.)
- Defining feature intersection and summarization parameters using supporting classes defined in PMT
- Defining field calculation formulas and code blocks

TOC Tool Download Scripts

Downloader

The downloader module handles downloading source data to the “Raw” data folder for downstream processing and analysis. Data are organized into subfolders defined by the RAW_FOLDERS variable in the download_config module. Purpose-specific download functions are defined here that utilize methods defined more abstractly in supporting modules, including census_geo, census, helper, and open_street_map.

`download.downloader.download_census_geo`(***overwrite=True***)
Download census data

- downloads and unzips the census block and blockgroup shapefiles
- downloads and writes out to table the ACS race and commute data
- downloads LODES data to table

Inputs:

- RAW//temp_downloads (folder path)
- RAW//CENSUS (extract path)
- CENSUS_GEO_TYPES (list of geographies)

Outputs:

- RAW//CENSUS//BG (block groups geogrpahies)
- RAW//CENSUS//TABBLOCK (block geographies)

`download.downloader.download_commute_data`(***overwrite=True***)
Downloads ACS commute data of interest

Inputs:

- RAW//CENSUS (root census folder)

Outputs:

- RAW//CENSUS//ACS_{year}_commute.csv

`download.downloader.download_lodes_data`(***overwrite=True***)
Download LODES data for job counts

- downloads lodes files by year and optionally aggregates to a coarser geographic area

Inputs:

- RAW//LODES (root lodes folder)

Outputs:

- RAW//LODES//fl_wac_S000_JT00_{year}_blk.csv.gz
- RAW//LODES//fl_wac_S000_JT00_{year}_bgrp.csv.gz
- RAW//LODES//fl_xwalk.csv.gz

`download.downloader.download_osm_data`(***overwrite=True***)

Download osm data - networks and buildings

- downloads networks as nodes.shp and edges.shp
- downloads all buildings, subset to poly/multipoly features
- both functions will create the output folder if not there

Inputs:

- RAW//Miami-Dade_County_Boundary.geojson (used as AOI to define area of needed data)
- RAW//OPEN_STREET_MAP

Outputs: (generally suffix will take the form q{1-4}_{year} where q indicates the quarter of the year)

- RAW//OPEN_STREET_MAP//bike_{suffix) [network]
- RAW//OPEN_STREET_MAP//buildings_{suffix)[builidng footprints]
- RAW//OPEN_STREET_MAP//drive_{suffix) [network]
- RAW//OPEN_STREET_MAP//walk_{suffix) [network]

`download.downloader.download_race_data`(***overwrite=True***)

Downloads ACS race data of interest

Inputs:

- RAW//CENSUS (root census folder)

Outputs:

- RAW//CENSUS//ACS_{year}_race.csv

`download.downloader.download_urls`(***overwrite=True***)

Downloads raw data that are easily accessible via web `request' at a url endpoint

Inputs:

- DOWNLOAD_URL_DICT (dictionary of output_name: url found in config.download_config)

Outputs: (11 files)

- RAW//{output_name} –> ['Imperviousness', 'MD_Urban_Growth_Boundary', 'Miami-Dade_County_Boundary',

'Municipal_Parks', 'County_Parks', 'Federal_State_Parks', 'Park_Facilities', 'Bike_Lanes', 'Paved_Path', 'Paved_Shoulder', 'Wide_Curb_Lane']

`download.downloader.setup_download_folder`(***dl_folder='RAW'***)

Creates a download folder if it doesn't already exist and populates with necessary subfolders for remaining download work

Parameters: **dl_folder** (*str*) – path download ROOT folder

Returns:	None

Download support

helper module

The helper module provides generalized methods to acquire data given a url endpoint, along with some purpose-built methods to obtain and/or clean ACS, OSM, and other datasets used in TOC analysis.

`download.helper.census_geoindex_to_columns`***(pd_idx, gen_geoid=True, geoid='GEOID10')***
Given an index of censusgeo objects, return a dataframe with columns reflecting the geographical hierarchy and identifying discrete features.

Parameters:

- **pd_idx** (*idx*) – Index, A pandas Index of censusgeo objects.
- **gen_geoid** (*bool*) – Boolean, default=True; If True, the geographical hierarchy will be concatenated into a geoid field. If False, only the geographicl hierarchy fields are returned.
- **geoid** (*str*) – String, default="GEOID10"; The name to assign the geoid column if gen_geoid is True.

Returns: **DataFrame; A data frame with columns reflecting the geographical hierachy of**
index, identifying discrete geographic features. This data frame has index as its index.

Return type: geo_cols (pandas.DataFrame)

`download.helper.download_commute_vars`***(year, acs_dataset='acs5', state='12', county='086', table=None, columns=None)***
Downloads commute (journey to work) data from available ACS data in table B08301.

Parameters:

- **year** – Int
- **acs_dataset** (*str, default="acs5"*) – Which ACS dataset to download (3-year, 5-year, e.g.)
- **state** (*str, default="12"*) – Which state FIPS code to download data for ("12" is Florida)
- **county** – (str, defult="086"): Which county FIPS code to download data for ("086" is Miami-Dade)
- **table** (*str*) – string code for the Census table of interest ex: "B03002"
- **columns** (*dict*) – key, value pairs of Census table columns and rename (ex: {"001E": "Total_Commutes", "003E": "Drove_alone"})

Returns: **commute_data (pandas.DataFrame)**
A data frame with columns showing commute statistics by mode

Raises: **ValueError** – If the table is not found (i.e. the requested year's data are not available)

`download.helper.download_file_from_url`(***url***, ***save_path***)
Downloads file resources directly from a url endpoint to a folder

Parameters:
- **url** (*str*) – String; path to resource
- **save_path** (*str*) – String; path to output file

Returns: None

`download.helper.download_race_vars`(***year***, ***acs_dataset='acs5'***, ***state='fl'***, ***county='086'***, ***table=None***, ***columns=None***)
Downloads population race and ethnicity variables from available ACS data in table B03002.

Parameters:
- **year** (*int*) – year of interest
- **acs_dataset** (*str*) – String, default="acs5"; Which ACS dataset to download (3-year, 5-year, e.g.)
- **state** (*str*) – String, default="12"; Which state FIPS code to download data for (12 is Florida)
- **county** (*str*) – String, defult="086"; Which county FIPS code to download data for (086 is Miami-Dade)
- **table** (*str*) – string code for the Census table of interest ex: "B03002"
- **columns** (*dict*) – key, value pairs of Census table columns and rename (ex: {"002E": "Total_Non_Hisp", "012E": "Total_Hispanic")

Returns: **A data frame with columns showing population by race (white, black,** Asian, 2 or more, or other) and ethnicity (Hispanic, non-Hispanic) for block groups in the specified state-county.

Return type: race_data (pandas.DataFrame)

Raises: **ValueError** – If the table is not found (i.e. the requested year's data are not available)

`download.helper.fetch_acs`(***year***, ***acs_dataset***, ***state***, ***county***, ***table***, ***columns***)
Internal function to hit the CENSUS api and extract a pandas DataFrame for the requested Table, State, County

Parameters:
- **year** (*int*) – year of interest

- **acs_dataset** (*str*) – Census data source: 'acs1' for ACS 1-year estimates, 'acs5' for ACS 5-year estimates, 'acs3' for ACS 3-year estimates, 'acsse' for ACS 1-year supplemental estimates, 'sf1' for SF1 data.
- **state** (*str*) – two letter state abbreviation
- **county** (*str*) – three digit FIPS code as a string
- **table** (*str*) – string code for the Census table of interest ex: "B03002"
- **columns** (*dict*) – key, value pairs of Census table columns and rename (ex: {"002E": "Total_Non_Hisp", "012E": "Total_Hispanic")

Returns: **Data frame with columns corresponding to designated variables, and row** index of censusgeo objects representing Census geographies.

Return type: pandas.DataFrame

***download.helper.get_filename_from_header*(*url*)**
Grabs a filename provided in the url object header

Parameters:	**url** (*str*) – string, url path to file on server
Returns:	filename as string
Return type:	filename (str)

***download.helper.trim_components*(*graph, min_edges=2, message=True*)**
Remove connected components less than a certain size (in number of edges) from a graph.

Parameters:

- **graph** (*nx.Graph*) – the networkx graph from which to remove small components
- **min_edges** (*int, optional, default=2*) – the minimum number of edges required for a component to remain in the network; any component with FEWER edges will be removed.
- **message** (*bool, optional, default=True*) – if True, prints a message indicating the number of components removed from graph

Returns: **a modified copy of the original graph with connected components smaller than min_edges** removed

Return type: G (nx.Graph)

census module

The census modules provides several helper methods to acquire and aggregate LODES data for use in the TOC tool.

exception `download.census.LodesFileTypeError`

`download.census.aggregate_lodes_data`***(geo_crosswalk_path, lodes_path, file_type, agg_geo)***

Aggregate LODES data to desired geographic levels, and save the created files [to .csv.gz]

Parameters:
- **geo_crosswalk_path** (*str*) – path to geographic crosswalk CSV
- **lodes_path** (*str*) – file path to gzipped lodes data file
- **file_type** (*str*) – shorthand for type of jobs summarization
- **agg_geo** (*str*) – shorthand for the geographic scale to aggregate data to

Returns: pd.Dataframe; data aggregated up to provided aggregate geography

`download.census.download_aggregate_lodes`***(output_dir, file_type, state, segment, part, job_type, year, agg_geog=None, overwrite=False)***

Helper function to fetch lodes data and aggregate to another census geography if one is provided

Parameters:
- **output_dir** (*str*) – path to location downloaded files should end up
- **file_type** (*str*) – one of three LODES groupings [‘od’, ‘rac’, ‘wac’] - OD: Origin-Destination data, totals are associated with both a home Census Block and a work Census Block - RAC: Residence Area Characteristic data, jobs are totaled by home Census Block - WAC: Workplace Area Characteristic data, jobs are totaled by work Census Block
- **state** (*str*) – The two-character postal abbreviation for the state
- **segment** (*str*) – Segment of the workforce, can have the values of [“S000”, “SA01”, “SA02”, “SA03”, “SE01”, “SE02”, “SE03”, “SI01”, “SI02”, “SI03”, “”]
- **part** (*str*) – Part of the state file, can have a value of either “main” or “aux”. Complimentary parts of the state file, the main part includes jobs with both workplace and residence in the state and the aux part includes jobs with the workplace in the state and the residence outside of the state.
- **job_type** (*str*) – LODES job types (“JT00” for All Jobs, “JT01” for Primary Jobs, “JT02” for All Private Jobs, “JT03” for Private Primary Jobs, “JT04” for All Federal Jobs, or “JT05” for Federal Primary Jobs).
- **year** (*int*) – year of LODES data to download
- **agg_geog** (*str*) – census geographies to aggregate lodes data to
- **overwrite** (*bool*) – if set to True, delete the existing copy of the LODES data

Returns: writes csv tables of aggregated lodes data in output_dir

Return type: None

***download.census.validate_aggregate_geo_inputs*(values, valid)**

Helper functions to validate that a given geo is a string or list and is found among a set of valid values.

Parameters:	• **values** (*str or list*) – one or more (string) values to validate. If not a string type or members are not string types, raises ValueError • **valid** (*list*) – a list valid string values. If value or any member of value is not found among the valid values, raises ValueError
Returns:	flag indicating if values is valid
Return type:	bool

***download.census.validate_lodes_download*(file_type, state, segment, part, job_type, year, agg_geos)**

Validates the download inputs to the dl_lodes method, combining other validation methods (file_type, state, segment, part, job_type, and year)

Parameters:	• **file_type** (*str*) – The LODES file type to download ("OD", "RAC", or "WAC") • **state** (*str*) – The two-character postal abbreviation for the state • **segment** (*str*) – Segment of the workforce, can have the values of ["S000", "SA01", "SA02", "SA03", "SE01", "SE02", "SE03", "SI01", "SI02", "SI03", ""] • **part** (*str*) – Part of the state file, can have a value of either "main" or "aux". Complimentary parts of the state file, the main part includes jobs with both workplace and residence in the state and the aux part includes jobs with the workplace in the state and the residence outside of the state. • **job_type** (*str*) – LODES job types ("JT00" for All Jobs, "JT01" for Primary Jobs, "JT02" for All Private Jobs, "JT03" for Private Primary Jobs, "JT04" for All Federal Jobs, or "JT05" for Federal Primary Jobs). • **year** (*str*) – year of LODES data to download • **agg_geos** (*str*) – census geographies to aggregate lodes data to
Returns:	flag indicating if parameters are valid
Return type:	bool

See also: download_aggregate_lodes

***download.census.validate_string_inputs*(value, valid_inputs)**

Helper function to validate that a given value is a string and is found in the list of valid inputs

Parameters:	• **value** (*var*) – a value to validate. If not a string type, raises ValueError

- **valid_inputs** (*list or str*) – one or more valid string values. If value is not found among the valid values, raises ValueError

Returns:	flag indicating if value is valid
Return type:	bool

download.census.validate_year(***year***)
Confirm year is correct datatype and part of expected years available

download.census.validate_year_input(***year***, ***state***)
Validate the provided year is in the possible set for a given state

census_geo module
Comment:
much of this code has been borrowed from: https://github.com/censusreporter/census-shapefile-utils

- *fetch_shapefiles.py*

or can be used to fetch files for a single state and/or single geography type. Pass an -s argument to limit by state, pass a -g argument to limit to a single geography type, and/or pass a -y argument to change the year from 2012 to something else (e.g. 2015).

>> python fetch_shapefiles.py >> python fetch_shapefiles.py -s WA >> python fetch_shapefiles.py -g place >> python fetch_shapefiles.py -y 2015 >> python fetch_shapefiles.py -s WA -g place -y 2015

If you use the -s argument to fetch files for a single state, the script will also download the national county, state and congressional district files that include data for your chosen state.

The script will create DOWNLOAD_DIR and EXTRACT_DIR directories if necessary, fetch a zipfile or set of zipfiles from the Census website, then extract the shapefiles from each zipfile retrieved.

DISABLE_AUTO_DOWNLOADS will prevent certain geography types from being automatically downloaded if no -g argument is passed to fetch_shapefiles.py. This may be useful because certain files, such as those for Zip Code Tabulation Areas, are extremely large. You can still target any geography in GEO_TYPES_LIST specifically, however. So to fetch the ZCTA data:

>> python fetch_shapefiles.py -g zcta5

download.census_geo.download_files_in_list(***filename_list***, ***download_dir***, ***force=False***)
Helper function to download list of files

Parameters:

- **filename_list** (*list*) – list of files
- **download_dir** (*str*) – path to download directory
- **force** (*bool*) – flag to force download of files in list

Returns: list of files downloaded

Return type: list

download.census_geo.extract_downloaded_file(***filename*, *extract_dir*, *unzip_dir*, *remove_on_error=True***)

Helper function to extract file from zip to a new directory

Parameters:
- **filename** (*str*) – path to zipped file
- **extract_dir** (*str*) – path to directory where zipped file will be written out when extracted
- **unzip_dir** (*str*) – name of the directory files are unzipped to
- **remove_on_error** (*bool*) – flag to delete failed unzipped files

Returns: None

download.census_geo.get_all_geo_types(***state=None*, *year='2019'***)

Helper function to fetch all valid census geographies

Parameters:
- **state** (*str*) – two character state abbreviation, if none provided data will be pulled for entire US
- **year** (*str*) – year of census geography, defaults is 2019

Returns: None

download.census_geo.get_content_length(***url***)

Helper function to determine how large the item to be downloaded is

Parameters: **url** (*str*) – url path

Returns: integer value of the content size

Return type: int

download.census_geo.get_filename_list_from_ftp(***target*, *state***)

Helper function to extract a list of files available from the provided FTP folder (target) by state

Parameters:
- **target** (*str*) – path to FTP site
- **state** (*str*) – two character state abbreviation

Returns: list of filenames matching state

Return type: filename_list (list)

`download.census_geo.get_one_geo_type`(*geo_type*, *download_dir=None*, *extract_dir=None*, *state=None*, *year='2019'*)
Helper function to fetch a single geographic dataset

Parameters:
- **geo_type** (*str*) – one of the valid census geographies (see GEO_TYPES_DICT keys in __init__)
- **download_dir** (*str*) – path to download directory
- **extract_dir** (*str*) – path to directory geographic data are extracted to
- **state** (*str*) – two character state abbreviation
- **year** (*str*) – base year for TIGER geography, default is 2019 (latest appropriate year for ACS and LODES)

Returns: None

open_street_map module

The open_street_map module provides generalized methods to acquire network data and building footprint polygons from Open Street Map. It depends on the osmnx module to download and clean mode-specific networks.

`download.open_street_map.calc_osm_bbox`(*gdf*)
Given a polygon GeoDataFrame, returns an appropriately formatted bbox dict for OSM

Parameters: **gdf** (*gpd.GeoDataFrame, gpd.Geoseries*) – GeoDataFrame object

Returns: **dictionary of coordinates representing the bbox for an area of interest,** formatted to work with osmnx

Return type: bbox (dict)

`download.open_street_map.download_osm_buildings`(*output_dir*, *polygon=None*, *bbox=None*, *data_crs=None*, *keep_fields=['osmid', 'building', 'name', 'geometry']*, *suffix=''*, *overwrite=False*)
Uses an Overpass query to fetch the OSM building polygons within a specified bounding box or the bounding box of a provided shapefile.

Parameters:
- **output_dir** (*str*) – Path to output directory.
- **polygon** (*str*) – path to a shapefile or geojson object readable by geopandas
- **bbox** (*dict*) – default=None; A dictionary with keys 'south', 'west', 'north', and 'east' of EPSG:4326-style coordinates, defining a bounding box for the area from which to fetch OSM features. Only required when study_area_polygon_path is not provided. See module notes for performance and suggestions on usage.

- **data_crs** (*int*) – integer value representing an EPSG code
- **keep_fields** (*list*) – list of fields to keep in output dataset
- **suffix** (*str*) – string value to be added to the end of the output folder
- **overwrite** (*bool*) – if set to True, delete the existing copy of buildings

Returns: **A gdf of OSM building features. By default, the CRS of** the gdf will be EPSG:4326 unless a tranformation is specified using transfor_epsg or a shape file with a differnt CRS is provided as study_area_polygon_path.

Return type: buildings_gdf (gpd.GeoDataFrame)

Notes

OSM building polygons features will automatically be saved in the output_dir`s `OSM_Buildings_{YYYYMMDDHHMMSS}.shp where YYYYMMDDHHMMSS is the date and time at which the Overpass query was pushed. This is done for record keeping purposes.

download.open_street_map.download_osm_networks(***output_dir***, ***polygon=None***, ***bbox=None***, ***data_crs=None***, ***net_types=['drive', 'walk', 'bike']***, ***pickle_save=False***, ***suffix=''***, ***overwrite=False***)

Download an OpenStreetMap network within the area defined by a polygon feature class or a bounding box.

Parameters:

- **output_dir** (*str*) – Path, Path to output directory. Each modal network (specified by net_types) is saved to this directory within an epoynmous folder as a shape file. If pickle_save is True, pickled graph objects are also stored in this directory in the appropriate subfolders.
- **polygon** (*str*) – Path, default=None; Path to study area polygon(s) shapefile. If provided, the polygon features define the area from which to fetch OSM features and bbox is ignored. See module notes for performance and suggestions on usage.
- **bbox** (*dict*) – default=None; A dictionary with keys ‘south’, ‘west’, ‘north’, and ‘east’ of EPSG:4326-style coordinates, defining a bounding box for the area from which to fetch OSM features. Only required when study_area_polygon_path is not provided. See module notes for performance and suggestions on usage.
- **data_crs** (*int*) – integer value representing an EPSG code
- **net_types** (*list*) – [String,…], default=[“drive”, “walk”, “bike”] A list containing any or all of “drive”, “walk”, or “bike”, specifying the desired OSM network features to be downloaded.

- **pickle_save** (*bool*) – default=False; If True, the downloaded OSM networks are saved as python networkx objects using the pickle module. See module notes for usage.
- **suffix** (*str*) – default=""; Downloaded datasets may optionally be stored in folders with a suffix appended, differentiating networks by date, for example.
- **overwrite** (*bool*) – if set to True, delete the existing copy of the network(s)

Returns: **A dictionary of networkx graph objects. Keys are mode names based on** net_types; values are graph objects.

Return type: G (dict)

`download.open_street_map.validate_bbox`(***bbox***)

Given a dictionary defining a bounding box, confirm its values are valid. North/south values must be between -90 and 90 (latitude); east/west values must be between -180 and 180 (longitude).

Parameters: **bbox** (*dict*) – A dictionary with keys 'south', 'west', 'north', and 'east' of EPSG:4326-style coordinates

Returns: None, This function simply raises exceptions if an invalid bounding box is provided

Raises: **ValueError** –

- If required keys are not found - If provided coordinate values are not within valid ranges

`download.open_street_map.validate_inputs`(***study_area_poly=None***, ***bbox=None***, ***data_crs=3857***)

Validation method for input downloading osm data via osmnx, converts polygon to bbox if provided

Parameters:

- **study_area_poly** (*str*) – path to a valid geospatial data file readable by geopandas
- **bbox** (*dict*) – A dictionary with keys 'south', 'west', 'north', and 'east' of EPSG:4326-style coordinates
- **data_crs** (*int*) – valid EPSG code in the projected coordinates

Returns: dictionary of 'north', 'south', 'east', 'west' coordinates

Return type: bbox (dict)

***download.open_street_map.validate_network_types*(network_types)**

Validation method to confirm whether provided network string matches osmnx types

Parameters:	**network_types** (*str or list*) – string or list of strings
Returns:	return list of network types
Return type:	network_types (list)
Raises:	**ValueError** – • if no item is in valid types

parcel_ftp module

The parcel_ftp module provides generalized methods to acquire Florida Department of Revenue parcel data (GIS features and NAL tables). It depends on the ftplib and re modules to download and save consistent versions of parcel data for specified counties and years.

***download.parcel_ftp.extract_county_name*(url, regex)**

Extract a Florida county name from a url string

Parameters:	• **url** (*str*) – url to a data file • **regex** (*str*) – regular expression to find a county
Returns:	**The FIRST county found; if the regex might find more than 1 unique county,** consider a different regex
Return type:	str

***download.parcel_ftp.extract_county_number*(url)**

Extract a Florida DOR county number from a url string

Parameters:	**url** (*str*) – url to a data file
Returns:	**The FIRST county number found; if the regex might find more than 1 unique** county, consider a different regex
Return type:	str

***download.parcel_ftp.extract_year*(url, regex='[0-9]{4}')**

Extract a year from a url string

Parameters:	• **url** (*str*) – url to a data file • **regex** (*str*) – regular expression to find a year. The default is [0-9]{4}, or 4 numbers in a row (which works for the Florida DOR FTP site)
Returns:	**The FIRST year found; if the regex might find more than 1 unique year,** consider a different regex
Return type:	int

`download.parcel_ftp.fdor_availability`(***df***, ***year=None***, ***county=None***)
Filter all available FDOR GIS and Tax Roll data according to desired specifications

Parameters:
- **df** (*pandas.DataFrame*) – output of fdor_gis_and_tax() (a dataframe of available data)
- **year** (*int or list, optional*) – year(s) for which data is desired. The default is None, no filtering will be completed on year
- **county** (*str or list, optional*) – county(ies) for which data is desired. The default is None, no filtering will be completed on county

Notes

For searching by county, note the following spelling specifications:

1. DeSoto [not Desoto]
2. Miami-Dade [not Dade or Miami Dade]
3. St. Johns [not St Johns or Saint Johns]
4. St. Lucie [not St Lucie or Saint Lucie]

If the spelling of a county does not the fdor_gis_and_tax results, this function will return no data, so please be careful with spelling!

Returns:	dataframe of available data according to the provided specs
Return type:	pandas.DataFrame
Raises:	**ValueError** – if the specs result in no available data

`download.parcel_ftp.fdor_gis_and_tax`(***save_path=None***)
Parse (and optionally save) the download link for all GIS and Tax Roll data available on the FDOR FTP site.

Parameters: **save_path** (*str*) – Path to which to save the results. If None, urls are not committed to disk and only a pandas dataframe of urls is returned.

Returns:
- file: either “GIS” for GIS data, or “TaxRoll” for tax roll data
- year: the year associated with the data (or 9999 if there is none)
- county: the county associated with the data (or ‘’ if there is none)
- url: link to the data

Return type: pandas.DataFrame with 4 columns

`download.parcel_ftp.flatten`(***d***, ***parent_key=''***, ***sep='/'***)
Flatten nested dictionaries into a single dictionary with combined keys

Parameters:
- **d** (*dict*) – dictionary to flatten

- **parent_key** (*str, optional*) – string to paste to the front of every key. The default is ‘’, keys get no prefix
- **sep** (*str, optional*) – string used to separate nested keys. The default is ‘/’ [for usage with directories/urls]

Returns: a flattened dictionary, with nested keys joined by sep

Return type: dict

`download.parcel_ftp.florida_counties`**()**

Produce a dataframe of Florida counties and all of their possible representations

Returns: **with 4 columns:**

- county: county name
- co_no: Florida DOR county number
- fips: 3-digit county FIPS code
- format: a possible representation of the county name (case insensitive)

Return type: pandas.DataFrame

`download.parcel_ftp.format_florida_counties`**(*county*)**

Produce all possible representations of a county name for the state of Florida (representations are case insensitive)

Parameters: **county** (*str*) – county name

Returns: **options for representing county names. in particular, spaces** will be condensed to “_” or “”, and the word “saint” will be tried as “saint”, “st”, and “st.”

Return type: list

`download.parcel_ftp.get_ftp_files`**(*folder_connection, ftp_site='sdrftp03.dor.state.fl.us'*)**

Get file paths to all files within an FTP folder

Parameters:

- **folder_connection** (*str*) – Folder name in an FTP directory. If the folder is not at the level of the main FTP site, it must be specified using pathing, e.g. folder/subfolder/subsubfolder
- **ftp_site** (*str, optional*) – FTP main site. The default is “sdrftp03.dor.state.fl.us”, which is the FDOR FTP main site

Returns: file paths to FTP files

Return type: list

`download.parcel_ftp.traverse`**(ftp, depth=None)**
Produce a recursive listing of an ftp server contents (starting from the current directory)

Parameters:	• **ftp** (*ftplib.FTP object*) – FTP connection • **depth** (*None*) – controls depth to which searching is completed; ignored if provided by user, searching always begins at the folder connection of the FTP object but required to support recursive searches of ftp nodes.
Returns:	**recursive dictionary, where each key contains the contents of the** subdirectory or None if it corresponds to a file
Return type:	level (dict)

TOC Tool Prepare Scripts

Preparer

The preparer module standardizes and formats all raw datasets into a common storage pattern, normalizing or separating geospatial data from tabular data where possible to decrease overall file sizes. The standardized databases include PMT_BasicFeatures and PMT_YYYY (where YYYY = the relevant year of data). Standardization includes removing unnecessary attributes, renaming attributes for readability, merging data where needed, and placing outputs in a common geodatabase structure. In addition to standardization, much of the analytical processing is performed via this module.

Functions defined in this module are purpose-built for TOC analysis in Miami-Dade County's TOC Toolkit. They lean on more abstract functions and classes defined in prepare_helpers, PMT and other supporting modules.

Functions:

- process_normalized_geometries()
- process_basic_features()
- process_parks()
- process_udb()
- process_transit()
- process_parcels()
- process_permits()
- enrich_block_groups()
- process_parcel_land_use()
- process_imperviousness()
- process_osm_networks()

- process_bg_apply_activity_models()
- process_allocate_bg_to_parcels()
- process_model_se_data()
- process_osm_skims()
- process_model_skims()
- process_osm_service_areas()
- process_centrality()
- process_walk_times()
- process_ideal_walk_times()
- process_access()
- process_contiguity()
- process_bike_miles()
- process_travel_stats()
- process_walk_to_transit_skim()
- process_serpm_transit()

`prepare.preparer.enrich_block_groups`**(*overwrite=True*)**
YEAR by YEAR, enrich block group with parcel data and race/commute/jobs data as table

- if Year == "NearTerm", process as normal (parcel data have been updated to include permit updates)

Inputs:

- CLEANED//PMT_{year}.gdb//Polygons//Parcels
- CLEANED//PMT_{year}.gdb//Polygons//Census_BlockGroups
- RAW//CENSUS//ACS_{year}_race.csv
- RAW//CENSUS//ACS_{year}_commute.csv
- RAW//LODES//fl_wac_S000_JT00_{year}_bgrp.csv.gz

Output:

- CLEANED//PMT_{year}.gdb//Enrichment_census_blockgroups (table)

`prepare.preparer.process_access`**()**
Summarizes activities (jobs, school enrollments, housing units, etc.) reachable from zone features (MAZs for non-motorized modes, TAZs for motorized modes) by alternative travel modes (walk, bike, transit, auto).

Inputs

- CLEANED//osm_networks//{mode})Skim_{vintage}.csv
- CLEANED//SERPM//SERPM_OD_{model_year}.csv
- CLEANED//PMT_{year}.gdb//EconDemog_MAZ
- CLEANED//PMT_{year}.gdb//EconDemog_TAZ

Outputs

- CLEANED//PMT_{year}.gdb//Access_maz_Bike
- CLEANED//PMT_{year}.gdb//Access_maz_Walk
- CLEANED//PMT_{year}.gdb//Access_taz_Auto
- CLEANED//PMT_{year}.gdb//Access_taz_Transit

`prepare.preparer.process_allocate_bg_to_parcels`(***overwrite=True***)
Disaggregation of modeled block group data back to parcels, for the NearTerm time period, only parcels with permits are allocated to, as the other parcels are considered to be relatively static and carry over data from the allocation

Inputs:
parcel geometry, block group geometry, modeled block group tables - CLEANED//PMT_YYYY.gdb//Polygons//Parcels - CLEANED//PMT_YYYY.gdb//Polygons//Census_BlockGroups - CLEANED//PMT_YYYY.gdb//Modeled_blockgroups

Outputs:

- CLEANED//EconDemog_parcels (EconDemog_parcels: table of disaggregated economic and demographic data at the parcel level)

`prepare.preparer.process_basic_features`(***overwrite=True***)
Utilizing the basic features, StationAreas, Corridors are genrated and used to generate SummaryAreas for the project

Inputs:

- BASIC_FEATURES = makePath(CLEANED, "PMT_BasicFeatures.gdb", "BasicFeatures")
- CLEANED//BASIC_FEATURES//StationArea_presets
- CLEANED//BASIC_FEATURES//Corridor_presets
- CLEANED//BASIC_FEATURES//SMARTplanStations
- CLEANED//BASIC_FEATURES//SMARTplanAlignments

Outputs:

- CLEANED//BASIC_FEATURES//StationAreas
- CLEANED//BASIC_FEATURES//Corridors
- CLEANED//BASIC_FEATURES//StationsLong
- CLEANED//BASIC_FEATURES//SummaryAreas

`prepare.preparer.process_bg_apply_activity_models`(***overwrite=True***)
Using existing LODES and Census demographic data, a linear model is fitted to the data at the block group level. Modeled results are used in all years, even in those with observed data, so that there are clearer relationships and trends over time (mixing observed and modeled results can yield unexpected patterns at the temporal boundary between observed and estimated data).

Inputs:

- enriched Block group data (LODES, demographics) with parcel summarizations

Outputs:

- modeled version of the original enriched data

`prepare.preparer.process_bike_facilities`(***overwrite=True***)
Combines and formats bike facility data layers previously downloaded from the MD Open Data Portal. A copy of the combined dataset is placed in each yearly geodatabase

Inputs:

- RAW//Bike_lane.geojson
- RAW//Paved_Path.geojson
- RAW//Paved_Shoulder.geojson
- RAW//Wide_Curb_Lane.geojson

Outputs:

- RAW//bike_facilities.shp
- CLEANED//PMT_YYYY.gdb//Network//bike_facilities

`prepare.preparer.process_bike_miles`(***overwrite=True***)
Intersects Summary area polygons with bike facilities and summarizes each facility type by miles within the summary area.

Inputs:

- CLEANED//PMT_YYYY.gdb//Networks//bike_facilities
- CLEANED//PMT_YYYY.gdb//Polygons//SummaryAreas

Outputs:

- CLEANED//PMT_YYYY.gdb//BikeFac_summaryareas

`prepare.preparer.process_centrality`(**)**
For each analysis year, analyze network centrality for all nodes in the bike network. Assigns a centrality score to parcels based on nearby network nodes.

Inputs:

- CLEANED//osm_networks//bike_{vintage}.gdb
- CLEANED//PMT_{year}.gdb//Polygons//parcels

Outputs:

- CLEANED//PMT_{year}.gdb//Networks//nodes_bike
- CLEANED//PMT_{year}.gdb//Centrality_parcels

`prepare.preparer.process_contiguity`(***overwrite=True***)
Estimates contiguity of developable land year over year by removing building footprints and other non-developable areas from the parcel layer and calculating the area of the remaining space on each parcel

Inputs:

- CLEANED//BASIC_FEATURES//MiamiDadeCountyBoundary
- CLEANED//PMT_YYYY.gdb//Polygons//Parcels
- RAW//ENVIRONMENTAL_FEATURES//NHDPLUS_H_0309_HU4_GDB.gdb//NHDWaterbody
- RAW//ENVIRONMENTAL_FEATURES//PADUS2_0FL.gdb//PADUS2_0Combined_DOD_Fee_Designation_Easement_FL
- RAW//OpenStreetMap//buildings_{prefix_qX_YYYY}//OSM_Buildings_{YYYYMMDDTTTT}.shp

Outputs:

- CLEANED//PMT_{year}.gdb//Congiguity_parcels

***prepare.preparer.process_ideal_walk_times*(*overwrite=True*)**
Estimates hypothetical walk times from parcels to stations and parcels to parks based on spatial relationships among parcel features and stations, parks. Assumes a constant walk speed.

Inputs

- CLEANED//PMT_{year}.gdb//Polygons//parcels
- CLEANED//Park_Points.shp
- BASIC_FEATURES//SMARTplanStations

Outputs

- CLEANED//PMT_{year}.gdb//WalkTimeIdeal_parcels

***prepare.preparer.process_imperviousness*(*overwrite=True*)**
Calculates impervious percentage by Census Block, and generates area estimates for NonDev, DevOS, DevLow, DevMed, and DevHigh intensity classes

Inputs:

- RAW//Imperviousness.zip
- BASIC_FEATURES//MiamiDadeCountyBoundary

Outputs:

- CLEANED//PMT_{year}.gdb//Imperviousness_census_blocks

***prepare.preparer.process_lu_diversity*(*overwrite=True*)**
YEAR OVER YEAR: calculates land use diversity within aggregate geometries using parcels

The diversity measures are defined as followed:

- **Simpson index**: mathematically, the probability that a random draw of one unit of land use A would be followed by a random draw of one unit of land use B. Ranges from 0 (only one land use present) to 1 (all land uses present in equal abundance)
- **Shannon index**: borrowing from information theory, Shannon quantifies the uncertainty in predicting the land use of a random one unit draw. The higher the uncertainty, the higher the diversity. Ranges from 0 (only one land use present) to -log(1/|land uses|) (all land uses present in equal abundance)
- **Berger-Parker index**: the maximum proportional abundance, giving a measure of dominance. Ranges from 1 (only one land use present) to 1/|land uses| (all land uses present in equal abundance). Lower values indicate a more even spread, while high values indicate the dominance of one land use.
- **Effective number of parties (ENP):** a count of land uses, as weighted by their proportional abundance. A land use contributes less to ENP if it is relatively rare, and more if it is relatively common. Ranges from 1 (only one land use present) to |land uses| (all land uses present in equal abunance)
- **Chi-squared goodness of fit**: the ratio of an observed chi-squared goodness of fit test statistic to a "worst case scenario" chi-squared goodness of fit test statistic. The goodness of fit test requires the definition of an "optimal" land use distribution ("optimal" is assumed to be equal abundance of all land uses, but can be specified by the user). The "worst case scenario" defines the highest possible chi-squared statistic that could be observed under the optimal land use distribution. In practice, this "worst case scenario" is the equivalent of the least likely land use [according to the optimal distribution] comprising the entire area. Ranges from 0 (all land uses present in equal abundance) to 1 (only one land use present)

Inputs:

- CLEANED//PMT_BasicFeatures.gdb//SummaryAreas (geometry)
- REF//Land_Use_Recode.csv
- CLEANED//PMT_YYYY.gdb//Polygons//Parcels

Outputs:

- CLEANED//PMT_YYYY.gdb//Diversity_summaryareas

`prepare.preparer.process_model_se_data`(***overwrite=True***)

Summarizing parcel level data up to MAZ and TAZ, and including SERPM model data for variables that cannot be readily estimated from parcel records (school enrollment, e.g.)

Inputs: (parcel geometry, MAZ geometry (includes TAZ geometry))

- CLEANED//PMT_YYYY.gdb//Polygons//MAZ
- CLEANED//PMT_YYYY.gdb//Polygons//Parcels
- CLEANED//PMT_YYYY.gdb//EconDemog_parcels
- RAW//SERPM//maz_data_2015.csv

Outputs: (parcel and SERPM data summarized up to MAZ and TAZ)

- CLEANED//PMT_YYYY.gdb//EconDemog_MAZ
- CLEANED//PMT_YYYY.gdb//EconDemog_TAZ

`prepare.preparer.process_model_skims`**()**

For each SERPM model year, combine transit skims for local and premium transit into one table. Get best available transit time, eliminating false connections.

:: Assumes transit and auto skims have same fields.

Inputs:

- RAW//SERPM//AM_HWY_SKIMS_{model_year}.csv
- RAW//SERPM//DLY_VEH_TRIPS_{model_year}.csv
- CLEANED//SERPM//TAZ_to_TAZ_local_{model_year}.csv
- CLEANED//SERPM//TAZ_to_TAZ_prem_{model_year}.csv

Outputs:

- CLEANED//SERPM/SERPM_OD_{model_year}.csv

`prepare.preparer.process_normalized_geometries`**(*overwrite=True*)**

YEAR BY YEAR:

- Sets up Year GDB, and 'Polygons' feature dataset
- Adds MAZ, TAZ, Census_Blocks, Census_BlockGroups, SummaryAreas
- for each geometry type, the year is added as an attribute
- for NearTerm, year is set to 9998

Inputs:

- RAW//CENSUS//..//{census_blocks.shp}
- RAW//CENSUS//..//{census_blockgroups.shp}
- RAW//TAZ.shp
- RAW//MAZ_TAZ.shp
- CLEANED//BASIC_FEATURES//SummaryAreas

Outputs:

- CLEANED//PMT_{year}.gdb//PolygonsCensus_Blocks;Census_Blockgroups;TAZ;MAZ;SummaryAreas

`prepare.preparer.process_osm_networks`**()**

Creates bicycle and walk networks from osm-downloaded shape files and a network dataset template.

Inputs:

- RAW//OpenStreetMap//{mode}_{vintage}//edges.shp
- REF//osm_{mode}_template.xml

Outputs:

- CLEANED//osm_networks//{mode}_{vintage}.gdb
- CLEANED//PMT_{year}.gdb//Networks//edges_bike

prepare.preparer.process_osm_service_areas()
Estimates service area lines and polygons defining the 30-minute walkshed around transit stations and park facilities.

Inputs

- BASIC_FEATURES//SMARTplanStations
- CLEANED//Park_Points.shp
- CLEANED//osm_networks//walk_{vintage}.gdb

Outputs

- CLEANED//PMT_{year}.gdb//Networks//walk_to_parks_MERGE
- CLEANED//PMT_{year}.gdb//Networks//walk_to_parks_NO_MERGE
- CLEANED//PMT_{year}.gdb//Networks//walk_to_parks_NON_OVERLAP
- CLEANED//PMT_{year}.gdb//Networks//walk_to_parks_OVERLAP
- CLEANED//PMT_{year}.gdb//Networks//walk_to_stn_MERGE
- CLEANED//PMT_{year}.gdb//Networks//walk_to_stn_NO_MERGE
- CLEANED//PMT_{year}.gdb//Networks//walk_to_stn_NON_OVERLAP
- CLEANED//PMT_{year}.gdb//Networks//walk_to_stn_OVERLAP

prepare.preparer.process_osm_skims()
Estimated travel time by walking and biking between all MAZ origin-destination pairs and store in a long csv table.

Inputs

- CLEANED//osm_networks//{mode}_{vintage}.gdb
- CLEANED//PMT_{year}.gdb//PolygonsMAZ

Outputs

- CLEANED//osm_networks//{mode}_Skim_{vintage}.csv

prepare.preparer.process_parcel_land_use(***overwrite=True***)
Generates a table mapping parcels to a human readable land use category (multiple) using the DOR_UC attribute

Inputs:

- CLEANED//PMT_{year}.gdb//Polygons//Parcels
- REF//Land_Use_Recode.csv

Output:

- CLEANED//PMT_{year}.gdb//LandUseCodes_parcels (table) `prepare.preparer.process_parcels`(***overwrite=True***)

YEAR by YEAR

- cleans geometry, joins parcels from DOR to NAL table keeping appropriate columns
- **if year == NearTerm:**
- previous year parcels are copied in to NearTerm gdb

Inputs:

- RAW//Parcels//Miami_{year}.shp
- RAW//Parcels//NAL_{year}.shp

:: these data are acquired from a download via the FDOR ftp site

Outputs:

- CLEANED//PMT_{year}.shp//Polygons//Parcels

`prepare.preparer.process_parks`(***overwrite=True***)

Parks - merges park polygons into one and formats both poly and point park data.

YEAR by YEAR:

- sets up Points FDS and year GDB (unless they exist already)
- copies Park_Points in to each year gdb under the Points FDS
- treat NEAR_TERM like any other year

Inputs:

- RAW//Municipal_Parks.geojson (polygons)
- RAW//Federal_State_Parks.geojson (polygons)
- RAW//County_Parks.geojson (polygons)
- RAW//Park_Facilities.geojson (points)

Outputs:

- CLEANED//Park_points.shp; Park_Polys.shp
- CLEANED//PMT_{year}.gdb//Points//Park_points

`prepare.preparer.process_permits`(***overwrite=True***)

Processes Road Impact Fee report into point feature class, where points represent the parcel locations of active permits. The permit points are then used to generate a NearTerm parcel layer updating various parcel level metrics

Inputs:

- RAW//BUILDING_PERMITS//Road Impact Fee Collection Report – {year}.csv
- CLEANED//PMT_{year}.gdb//Polygons//Parcels

Outputs:

- CLEANED//PMT_NearTerm.gdb//Points//BuildingPermits
- CLEANED//PMT_NearTerm.gdb//Polygons//Parcels (updated)

***prepare.preparer.process_serpm_transit*()**

Combines estimates of TAP to TAP and TAZ to TAP travel times (in minutes) to create an OD table of TAZ to TAZ travel time estimates.

Inputs:

- RAW//SERPM//SERPM_TAZ_Centroids.shp
- RAW//SERPM//TAP_to_TAP_{skim version}_{model_year}.csv
- CLEANED//SERPM//TAZ_to_TAP_{net suffix for analysis year}.csv

Outputs:

- CLEANED//SERPM//TAP_to_TAP_{skim version}_{model_year}_clean.csv
- CLEANED//SERPM//TAZ_to_TAZ_{skim version}_{model_year}.csv

***prepare.preparer.process_transit*(*overwrite=True*)**

Converts a list of transit ridership files to points with attributes cleaned.

YEAR by YEAR:

- cleans and consolidates transit data into Year POINTS FDS
- **if YEAR == NearTerm:**
    - most recent year is copied over

NOTE: transit folder reflects current location, needs update to reflect cleaner structure

Inputs:

- RAW//TRANSIT//TransitRidership_byStop//AVERAGE_RIDERSHIP_PER_STOP_PER_TRIP_WEEKDAY_{hhmm_YYYY_MMM}_standard_format.XLS

:: these files must be acquired from DTPW

Outputs:

- CLEANED//PMT_{year}.gdb//Points//TransitRidership

***prepare.preparer.process_travel_stats*(*overwrite=True*)**

Estimates rates of vehicle miles of travel (VMT) per capita and per job based on vehicle trip estimates, estimated travel distances, and jobs and housing estimates from SERPM. These estimates are then applied year over year to parcel-based estimates of jobs and housing to approximate daily vehicle trips and VMT generated by each TAZ.

Inputs:

- CLEANED//SERPM//SERPM_OD_{model_year}.csv
- CLEANED//PMT_YYYY.gdb//EconDemog_TAZ

Outputs:

- CLEANED//PMT_YYYY.gdb//TripStates_TAZ

***prepare.preparer.process_udb*(overwrite=True)**
Converts Urban Development Boundary line feature class to a polygon.

Inputs:

- RAW//MD_Urban_Growth_Boundary.geojson
- RAW//Miami-Dade_County_Boundary.geojson

Outputs:

- CLEANED//UrbanDevelopmentBoundary.shp
- CLEANED//PMT_BasicFeatures.gdb//BasicFeatures//UrbanGrowthBoundary

***prepare.preparer.process_walk_times*()**
Estimates walk times from parcels to stations and parcels to parks based on spatial relationships among parcel features and service area lines.

Inputs

- CLEANED//PMT_{year}.gdb//Polygons//parcels
- CLEANED//PMT_{year}.gdb//Networks//walk_to_stn_MERGE
- CLEANED//PMT_{year}.gdb//Networks//walk_to_parks_MERGE

Outputs

- CLEANED//PMT_{year}.gdb//WalkTime_parcels

***prepare.preparer.process_walk_to_transit_skim*()**
Estimates the time required to walk between SERPM TAZ centroids and SERPM TAP (transit access point) nodes using the OSM walk network (for Miami-Dade County) or based on simple spatial relationships (coarser estimate outside MDC).

Inputs:

- RAW//SERPM//SERPM_TAZ_Centroids.shp
- RAW//SERPM//SERPM_TAP_Nodes.shp
- NETS_DIR//Walk{net suffix for analysis year}.gdb//osm/osm_ND

Outputs:

- CLEANED//SERPM/TAZ_to_TAP{net suffix for analysis year}.csv

Preparer support

prepare_helpers module

The prepare_helpers module defines a host of functions that support preparer procedures. Much of the heavy lifting for TOC analysis occurs here by methods that are more abstract and parameterized than the purpose-built methods in preparer.

`prepare.prepare_helpers.add_xy_from_poly`**(*poly_fc*, *poly_key*, *table_df*, *table_key*)**
Calculates x,y coordinates for a given polygon feature class and returns as new columns of a data

Parameters:
- **poly_fc** (*str*) – path to polygon feature class
- **poly_key** (*str*) – primary key from polygon feature class
- **table_df** (*pd.DataFrame*) – pandas dataframe
- **table_key** (*str*) – primary key of table df

Returns: updated table_df with XY centroid coordinates appended

Return type: pandas.DataFrame

`prepare.prepare_helpers.agg_to_zone`**(*parcel_fc*, *agg_field*, *zone_fc*, *zone_id*)**
Aggregate parcel data up to a zone feature class, limited to one field for aggregation

Parameters:
- **parcel_fc** (*str*) – parcel feature class path
- **agg_field** (*str*) – field to be aggregated
- **zone_fc** (*str*) – feature class data will be summarized to
- **zone_id** (*str*) – primary key of zone fc

Returns: dataframe of zoneID and summarized attribute

Return type: pandas.DataFrame

`prepare.prepare_helpers.allocate_bg_to_parcels`**(*bg_modeled_df*, *bg_geom*, *bg_id_field*, *parcel_fc*, *parcels_id='FOLIO'*, *parcel_wc=''*, *parcel_lu='DOR_UC'*, *parcel_liv_area='TOT_LVG_AREA'*)**
Allocate block group data to parcels using relative abundances (proportions of total among all parcels) of parcel building square footage

Parameters:
- **bg_modeled_df** (*pd.DataFrame*) – pandas DataFrame of modeled block group job, population, and commute data for allocation
- **bg_geom** (*str*) – Path; path to feature class of block group polygons
- **bg_id_field** (*str*) – block group key
- **parcel_fc** (*str*) – Path to shape of parcel polygons, containing at a minimum a unique ID field, land use field, and total living area field (Florida DOR)
- **parcels_id** (*str, default="FOLIO"*) – unique ID field in the parcels shape
- **parcel_wc** (*str, default=""*) – where clause to select out parcels and limit allocation to only the selected parcels (as when allocating NearTerm permitted parcels)
- **parcel_lu** (*str, default="DOR_UC"*) – land use code field in the parcels shape

- **parcel_liv_area** (*str, default="TOT_LVG_AREA"*) – building square footage field in the parcels shape

Returns: dataframe of the resultant allocation based on model

Return type: intersect_df (pd.DataFrame)

***`prepare.prepare_helpers.analyze_imperviousness`*(raster_points, rast_cell_area, zone_fc, zone_id_field)**

Summarize percent impervious surface cover in each of a collection of zones

Parameters:

- **raster_points** (*str*) – Path to clipped/transformed imperviousness raster as points (see the prep_imperviousness function)
- **rast_cell_area** (*float*) – numeric value for pixel area on the ground
- **zone_fc** (*str*) – Path to polygon geometries to which imperviousness will be summarized
- **zone_id_field** (*str*) – id field in the zone geometries

Returns: table of impervious percent within the zone geometries

Return type: df (pandas dataframe)

***`prepare.prepare_helpers.apply_blockgroup_model`*(year, bg_enrich_path, bg_geometry_path, bg_id_field, model_coefficients, shares_from=None)**

Predict block group-level total employment, population, and commutes using
pre-fit linear models, and apply a shares-based approach to subdivide totals into relevant subgroups

Parameters:

- **year** (*int*) – year of the bg_enrich data
- **bg_enrich_path** (*str*) – path to enriched block group data; this is the data to which the models will be applied
- **bg_geometry_path** (*str*) – path to geometry of block groups underlying the data
- **bg_id_field** (*str*) – block group unique id column
- **model_coefficients** (*pd.DataFrame*) – pandas.DataFrame of model coefficients
- **shares_from** (*dict*) – optional if the year of interest does not have observed data for either LODES or ACS, provide other files from which subgroup shares can be calculated (with the keys “LODES” and “ACS”, respectively). For example, imagine that you are applying the models to a year where ACS data was available but LODES data was not. Then, shares_from = {“LODES”: “path_to_most_recent_bg_enrich_file_with_LODES”}. A separate file

does not need to be referenced for ACS because the data to which the models are being applied already reflects shares for ACS variables. The default is None, which assumes LODES and ACS data are available for the year of interest in the provided `bg_enrich file

Returns:	pd.DataFrame of model application results
Return type:	alloc (pd.DataFrame)

`prepare.prepare_helpers.assign_features_to_agg_area`(***in_features***, ***agg_features=None***, ***buffer=None***, ***in_fields='*'***, ***as_df=False***)

Generates a feature class or table that relates disaggregate features to those in an aggregate area. Optionally, the aggregate areas can be generated from the disaggregate features using a buffer, creating a "floating zone" around each.

Disaggregate features are always assigned to aggregate areas based on the intersection of their centroids with the aggreagte area features.

Parameters:

- **in_features** (*str*) – path to a feature class of disaggregate features to be related to features in agg_features
- **agg_features** (*str*) – path to a feature class of aggreagate areas. If none, buffer must be provided.
- **buffer** (*str*) – If agg_features is not provided, a buffer may be provided to create floating zones around each feature in in_features. May be a linear distance that includes distance and units ("2640 Feet", e.g.) or a field in in_features that specifies a linear distance for each feature.
- **in_fields** (*list, default="*"*) – Field(s) in in_features to retain in the result. By default, all fields are retained, but a list can be specified to return only named fields to make the output more concise.
- **as_df** (*bool*) – If True, returns a data frame (table) of the intersect between in_features centroids and agg_features or floating zones. Otherwise returns the path to a temporary feature class with intersection geometries and attributes.

Returns: fc path or DataFrame (see as_df arg)

See also

PMT.intersectFeatures

`prepare.prepare_helpers.berger_parker_diversity`(***in_df***, ***group_col***, ***weight_col=None***, ***total_col=None***, ***pct_col=None***, ***count_lu=None***, ****kwargs***)

Berger-Parker index: the maximum proportional abundance, giving a
measure of dominance.

Ranges from 1 (only one land use present) to 1/|land uses| (all land uses present in equal abundance). Lower values indicate a more even spread, while high values indicate the dominance of one land use.

This function is not intended to be run directly. Use lu_diversity.

`prepare.prepare_helpers.build_short_term_parcels`(***parcel_fc, parcels_id_field, parcels_lu_field, parcels_living_area_field, parcels_land_value_field, parcels_total_value_field, parcels_buildings_field, permit_fc, permits_ref_df, permits_id_field, permits_lu_field, permits_units_field, permits_values_field, permits_cost_field, units_field_match_dict={}***)
Using current parcel data and current permits, generate a near-term estimate of parcel changes as a temp feature class.

Parameters:

- **parcel_fc** (*str*) – Path to current parcel feature class
- **parcels_id_field** (*str*) – Primary key for parcel data
- **parcels_lu_field** (*str*) – Land use code attribute
- **parcels_living_area_field** (*str*) – Building floor area field
- **parcels_land_value_field** (*str*) – Parcel land value field
- **parcels_total_value_field** (*str*) – Combined building and land value field
- **parcels_buildings_field** (*str*) – Count of buildings per parcel field
- **permit_fc** (*str*) – Path to permitted development that has been spatialized
- **permits_ref_df** (*pandas.DataFrame*) – Table of reference units to map values from permits to parcel
- **permits_id_field** (*str*) – Foreign key in permits layer tying parcels and permits
- **permits_lu_field** (*str*) – Permits land use field
- **permits_units_field** (*str*) – Permits unit type field (ex: sq.ft)
- **permits_values_field** (*str*) – Permit unit value field (ex: 2526 –> reference to permits_units_field)
- **permits_cost_field** (*str*) – Combined administrative and construction cost field in permit data
- **units_field_match_dict** (*dict*) – k,v pair of parcel field and the unit list in permit_units_field that match

Returns: path to in_memory feature class with updated parcel data

Return type: temp_parcels (str)

`prepare.prepare_helpers.calculate_contiguity_index`(***quadrats_fc*, *parcels_fc*, *mask_fc*, *parcels_id_field*, *cell_size=40*, *weights='nn'***)
Calculate contiguity of developable area

Parameters:

- **quadrats_fc** (*str*) – path to fishnet of chunks for processing
- **parcels_fc** (*str*) – path to parcel polygons; contiguity will be summarized and reported at this scale
- **mask_fc** (*str*) – path to mask polygons used to eliminate areas that are not developable
- **parcels_id_field** (*str*) – name of parcel primary key field
- **cell_size** (*int, default=40*) – cell size for raster over which contiguity will be calculated. (in the units of the input data crs)
- **weights** (*str or dict, default="nn"*) – weights for neighbors in contiguity calculation (see notes for how to specify weights)

Returns: table of polygon-level (sub-parcel) contiguity indices

Return type: pd.DataFrame

Notes

Weights can be provided in one of two ways:

1. one of three defaults: "rook", "queen", or "nn".
   - "nn" (nearest neighbor) weights give all neighbors a weight of 1, regardless of orientation.
   - "rook" weights give all horizontal/vertical neighbors a weight of 1, and all diagonal neighbors a weight of 0
   - "queen" weights give all horizontal/vertical neighbors a weight of 2,and all diagonal neighbors a weight of 1
2. a dictionary of weights for each of 9 possible neighbors. This dictionary must have the keys ["top_left", "top_center", "top_right", "middle_left", "self", "middle_right", "bottom_left", "bottom_center", "bottom_right"]. If providing weights as a dictionary, a good strategy is to set "self"=1, and then set other weights according to a perceived relative importance to the cell itself. Cell-specific rates rarely need to be specified.

Raises: **ValueError** – If weights are an invalid string or a dictionary with invalid keys (see Notes)

`prepare.prepare_helpers.calculate_contiguity_summary`(***full_results_df*, *parcels_id_field*, *summary_funcs=['min', 'max', 'median', 'mean']*, *area_scaling=True***)
Summarize contiguity/developable area results from
analyze_contiguity_index from sub-parcel to parcel

Parameters:

- **full_results_df** (*pandas.DataFrame*) – dataframe output of analyze_contiguity_index
- **parcels_id_field** (*str*) – name of a field used to identify the parcels in the summarized parcel results
- **summary_funcs** (*list*) – [str, …] functions to be used to summarize contiguity to the parcel; available options include min, max, mean, and median Default is all options
- **area_scaling** (*bool*) – should a scaled version of developable area be calculated? If True, a "scaled_area" statistic will be calculated as developable area times contiguity index (at the parcel level). The default is True

Returns: **a table of summarized results attributed with:**

- A parcel identifier (as specified in analyze_contiguity_index when the feature class was initialized)
- Parcel developable area (summed to the parcel)
- {fun}-summarized contiguity, for each function in summary_funs
- {fun}-scaled area, for each of {fun}-summarized contiguity, if area_scaling = True

Return type: pandas dataframe

`prepare.prepare_helpers.clean_and_drop`***(feature_class, use_cols=None, rename_dict=None)***

Remove and rename fields provided for a feature class to format as desired

Parameters:

- **feature_class** (*str*) – path to feature class
- **use_cols** (*list*) – list of columns to keep
- **rename_dict** (*dict*) – key, value pairs of columns to keep and new column names

Returns: None

`prepare.prepare_helpers.clean_parcel_geometry`***(in_features, fc_key_field, new_fc_key, out_features=None)***

Cleans parcel geometry (repair faulty geometries, drop records with empty geometries, dissolve identical polygons, etc.) and sets common key to user supplied new_fc_key. Output features may have non-unique key field values if the original geometry is multi-part.

Parameters:

- **in_features** (*str*) – path to raw parcel shapefile
- **fc_key_field** (*str*) – primary key of raw shapefile data
- **new_fc_key** (*str*) – new primary key name used throughout processing
- **out_features** (*str*) – path to output parcel feature class

Returns: None

`prepare.prepare_helpers.clean_permit_data`***(permit_csv, parcel_fc, permit_key, poly_key, rif_lu_tbl, dor_lu_tbl, out_file, out_crs*)**

Reformat and clean RER road impact permit data, specific to the TOC tool

Parameters:

- **permit_csv** (*str*) – path to permit csv
- **parcel_fc** (*str*) – path to parcel feature class; should be most recent parcel year
- **permit_key** (*str*) – foreign key of permit data that ties to parcels (“FOLIO”)
- **poly_key** (*str*) – primary key of parcel data that ties to permits (“FOLIO”)
- **rif_lu_tbl** (*str*) – path to road_impact_fee_cat_codes table (maps RIF codes to more standard LU codes)
- **dor_lu_tbl** (*str*) – path to dor use code table (maps DOR LU codes to more standard and generalized categories)
- **out_file** (*str*) – path to output permit point feature class
- **out_crs** (*int*) – EPSG code

Returns: None

`prepare.prepare_helpers.clean_skim_csv`***(in_file, out_file, imp_field, drop_val=0, renames={}, node_offset=0, node_fields=['F_TAP', 'T_TAP'], chunksize=100000, **kwargs*)**

Reads an OD table and drops rows where imp_field = drop_val is true. Optionallly renumbers nodes by applying (adding) an offset to the original values. Saves a new csv containing key columns (node_fields and imp_field)

Parameters:

- **in_file** (*str*) – Path to a long OD table in csv format
- **out_file** (*str*) – Path to a new (shortened) OD table to store function outputs
- **imp_field** (*str*) – The name of the field in in_file containing impedance (time, distance, cost) values between OD pairs. If this field is renamed using renames, the new name should be provided here.
- **drop_val** (*int/float, default=0*) – Rows where imp_field is equal to drop_val are dropped from the skim
- **renames** (*dict*) – Keys are column names in in_file, values are new names for those columns in`out_file`
- **node_offset** (*int, default=0*) – If origin and destiantion nodes need to be renumbered, this value will be added to the original values in node_fields. (This is used when the multiple skims are being used to create a network and node number collisions need to be handled.)

- **node_fields** (*str*) – [String,…] List of fields containing node values. At a minimum, there should be two fields listed: the origin and destiantion fields. All fields listed will have the node_offset (if given) applied. If columns are renamed used renames, give the new column names, not the old ones.
- **chunksize** (*int, default=10000*) – Number of rows to process at a given time. More rows are faster but require more memory.
- **kwargs** – Keywords to use when loading in_file with pd.read_csv.

Returns: path to output cleaned skim

Return type: `out_file

`prepare.prepare_helpers.combine_csv_dask`**(*merge_fields*, *out_table*, **tables*, *suffixes=None*, *col_renames={}*, *how='inner'*, ***kwargs*)**

Merges two or more csv tables into a single table based on key common columns. All other columns from the original tables are included in the output csv.

Parameters:

- **merge_fields** (*list*) – One or more column names common to all tables on which they will be merged
- **out_table** (*str*) – Path to the csv file to be created to store combined outputs
- **tables** (*list*) – Paths to the tables to be combined.
- **suffixes** (*list, default=None*) – If any tables have common column names (other than merge_fields) that would create naming collisions, the user can specify the suffixes to apply to each input table. If None, name collisions may generate unexpected colums in the output, so it is recommended to provide specific suffixes, especially if collisions are expected. Length of the list must match the number of tables.
- **col_renames** (*dict*) – A dictionary for renaming columns in any of the provided tables. Keys are old column names, values are new column names.
- **how** (*str, "inner" or "outer", default="inner"*) – If “inner” combined csv tables will only include rows with common values in merge_fields across all tables. If “outer”, all rows are retained with nan values for unmatched pairs in any table.
- **kwargs** – Any keyword arguments are passed to the dask dataframes read_csv method.

`prepare.prepare_helpers.consolidate_cols`**(*df*, *base_fields*, *consolidations*)**

Use the PMT.Consolidation class to combine columns and
return a clean data frame.

Parameters:	• **df** (*pd.DataFrame*) – pandas Dataframe • **base_fields** (*list*) – [String, …]; Field(s) in df that are not subject to consolidation but which are to be retained in the returned data frame. • **consolidations** (*iterable*) – [Consolidation, …]; Specifications for output columns that consolidate columns found in df.
Returns:	A new data frame with columns reflecting base_field and consolidations.
Return type:	clean_df (pd.DataFrame)

See also

PMT.Consolidation

`prepare.prepare_helpers.copy_net_result`**(*source_fds*, *target_fds*, *fc_names*)**
Since some PMT years use the same OSM network, a solved network analysis
can be copied from one year to another to avoid redundant processing. This function is a helper function called by PMT wrapper functions. It is not intended to be run independently.

Parameters:	• **source_fds** (*str*) – Path to source FeatureDataset • **target_fds** (*str*) – Path to destination FeatureDataset • **fc_names** (*str, list*) – [String, …] The feature class(es) to be copied from an already-solved analysis. Provide the names only (not paths).
Returns:	**Copies network service area feature classes to the target year** output from a source year using the same OSM data. Any existing features in the feature dataset implied by the target year are overwritten.
Return type:	None

`prepare.prepare_helpers.create_permits_units_reference`**(*parcels*, *permits*, *lu_key*, *parcels_living_area_key*, *permit_value_key*, *permits_units_name='sq. ft.'*, *units_match_dict=None*)**
Creates a reference table by which to convert various units provided in
the Florida permits_df data to building square footage

Parameters:	• **parcels** (*str*) – path to MOST RECENT parcels_df data (see notes) • **permits** (*str*) – path to the building permits_df data • **lu_key** (*str*) – field name for a land use field present in BOTH the parcels_df and permits_df data • **parcels_living_area_key** (*str*) – field name in the parcels_df for the total living area (building area) • **permit_value_key** (*str*) – field name in the permits_df for the unit of measurement for permit types

- **permits_units_name** (*str*) – unit name for building area in the permits_units_field
- **units_match_dict** (*dict*) – a dictionary of the format {unit_name: parcel_field, …}, where the unit_name is one of the unit names present in the permits_df data's permits_units_field, and the parcel_field is the field name in the parcels_df corresponding to that unit. It should be used to identify non-building area fields in the parcels_df for which we can calculate a building area/unit for the multiplier. parcel_field can also take the form of a basic function (+,-,/,*) of a column, see Notes for specifications

Notes

The most up-to-date parcels_df data available should be used, because units multipliers for the short term should be based on the most current data

To specify a function for the units_match_field, use the format "{field} {function sign} {number}". So, for example, to map an 'acre' unit in the permits_df to a 'land_square_foot' field in the parcels_df, you'd use the dictionary entry 'acre': 'land_square_foot' / 43560

Returns: a table of units multipliers/overwrites by land use

Return type: pandas dataframe

***`prepare.prepare_helpers.csv_to_df`*(*csv_file*, *use_cols*, *rename_dict*)**
Helper function to convert CSV file to pandas dataframe, and drop unnecessary columns assumes any strings with comma (,) should have those removed and dtypes infered

Parameters:
- **csv_file** (*str*) – path to csv file
- **use_cols** (*list*) – list of columns to keep from input csv
- **rename_dict** (*dict*) – dictionary mapping existing column name to standardized column names

Returns: Pandas.DataFrame

***`prepare.prepare_helpers.enp_diversity`*(*in_df*, *group_col*, *weight_col=None*, *total_col=None*, *pct_col=None*, *count_lu=None*, ***kwargs*)**
Effective number of parties (ENP): a count of land uses, as weighted
by their proportional abundance.

A land use contributes less to ENP if it is relatively rare, and more if it is relatively common. Ranges from 1 (only one land use present) to |land uses| (all land uses present in equal abunance).

This function is not intended to be run directly. Use lu_diversity.

`prepare.prepare_helpers.enrich_bg_with_econ_demog`**(*tbl_path*, *tbl_id_field*, *join_tbl*, *join_id_field*, *join_fields*)**

Adds data from another raw table as new columns based on the fields provided. Handles compressed inputs and data typing on-the-fly.

Parameters:

- **tbl_path** (*str*) – path table being updated
- **tbl_id_field** (*str*) – table primary key
- **join_tbl** (*str*) – path to table being joined with tbl_path
- **join_id_field** (*str*) – join table foreign key
- **join_fields** (*list*) – [String, …]; list of fields to include in update

Returns: None

`prepare.prepare_helpers.enrich_bg_with_parcels`**(*bg_fc*, *parcels_fc*, *sum_crit=None*, *bg_id_field=None*, *par_id_field=None*, *par_lu_field=None*, *par_bld_area=None*, *par_sum_fields=None*)**

Relates parcels to block groups based on parcel centroid location and summarizes key parcel fields to the block group level, including building floor area by potential activity type (residential, jobs by type, e.g.).

Parameters:

- **bg_fc** (*str*) – path to block group feature class
- **parcels_fc** (*str*) – path to parcel feature class
- **sum_crit** (*dict*) – Dictionary whose keys reflect column names to be generated to hold sums of parcel-level data in the output block group data frame, and whose values consist of at least one PMT comparator class (Comp, And). These are used to map parcel land use codes to LODES variables, e.g. An iterable of comparators in a value implies an Or operation.
- **par_sum_fields** (*list*) –

  [String, …], default=[“LND_VAL”, “LND_SQFOOT”, “JV”,
  ”NO_BULDNG”, “NO_RES_UNTS”, “TOT_LVG_AREA]
  If provided, these parcel fields will also be summed to the block-group level.

- **bg_id_field** (*str*) – block group primary key attribute default=”GEOID”
- **par_id_field** (*str*) – parcel primary key attribute, default=”PARCELNO”
- **par_lu_field** (*str*) – parcel land use attribute, default=”DOR_UC”
- **par_bld_area** (*str*) – parcel building area attribute, default=”TOT_LVG_AREA”

Returns: DataFrame of block group ids and related/summarized parcel data

Return type:	bg_df (pd.DataFrame)

*`prepare.prepare_helpers.estimate_maz_from_parcels`***(*par_fc*, *par_id_field*, *maz_fc*, *maz_id_field*, *taz_id_field*, *se_data*, *se_id_field*, *agg_cols*, *consolidations*)**
Estimate jobs, housing, etc. at the MAZ level based on underlying parcel data.

Parameters:

- **par_fc** (*str*) – Path to Parcel features
- **par_id_field** (*str*) – Field identifying each parcel feature
- **maz_fc** (*str*) – Path to MAZ features
- **maz_id_field** (*str*) – Field identifying each MAZ feature
- **taz_id_field** (*str*) – Field in maz_fc that defines which TAZ the MAZ feature is in.
- **se_data** (*str*) – Path to a gdb table containing parcel-level socio-economic/demographic estimates.
- **se_id_field** (*str*) – Field identifying each parcel in se_data
- **agg_cols** (*object*) – [PMT.AggColumn, …] Columns to summarize to MAZ level
- **consolidations** (*object*) – [PMT.Consolidation, …] Columns to consolidated into a single statistic and then summarize to TAZ level.

Returns: DataFrame

See also

PMT.AggColumn PMT.Consolidation

*`prepare.prepare_helpers.field_mapper`***(*in_fcs*, *use_cols*, *rename_dicts*)**
Create a field mapping for one or more feature classes

Parameters:

- **in_fcs** (*list,str*) – list of feature classes
- **use_cols** (*list, str*) – list or tuple of lists of column names to keep
- **rename_dicts** (*dict*) – dict or tuple of dicts to map field names

Returns: arcpy.FieldMappings

*`prepare.prepare_helpers.generate_chunking_fishnet`***(*template_fc*, *out_fishnet_name*, *chunks=20*)**
Generates a fishnet feature class that minimizes the rows and columns based on number of chunks and template_fc proportions.

Parameters:

- **template_fc** (*str*) – path to template feature class
- **out_fishnet_name** (*str*) – name of output file
- **chunks** (*int*) – number of chunks to be used for processing

Returns: path to fishnet generated for chunking

Return type: quadrat_fc (str)

`prepare.prepare_helpers.generate_od_table`***(origin_pts, origin_name_field, dest_pts, dest_name_field, in_nd, imped_attr, cutoff, net_loader, out_table, restrictions=None, use_hierarchy=False, uturns='ALLOW_UTURNS', o_location_fields=None, d_location_fields=None, o_chunk_size=None)***

Creates and solves an OD Matrix problem for a collection of origin and
destination points using a specified network dataset. Results are exported as a csv file.

Parameters:

- **origin_pts** (*str*) – Path to point feature class representing origin locations
- **origin_name_field** (*str*) – Unique ID field of point data
- **dest_pts** (*str*) – Path to point feature class representing destination locations
- **dest_name_field** (*str*) – unique ID of destination points
- **in_nd** (*str*) – Path to network dataset
- **imped_attr** (*str*) – String; impedance attribute
- **cutoff** (*int*) – numeric
- **net_loader** (*class*) – NetLoader object defininig how our network is loaded/configured
- **out_table** (*str*) – Path to output table
- **restrictions** (*list*) – [String, …], default=None List of restriction attributes to apply during the analysis.
- **use_hierarchy** (*bool*) – Boolean, default=False
- **uturns** (*str*) – String, default="ALLOW_UTURNS"
- **o_location_fields** (*list, default=None*) – [String, …], if origin_pts have pre-calculated network location fields, list the fields in order ("SourceOID", "SourceID", "PosAlong", "SideOfEdge", "SnapX", "SnapY", "Distance",). This speeds up processing times since spatial analysis to load locations on the network is not needed.
- **d_location_fields** (*list, default=None*) – [String, …], same as o_location_fields but for destination points.
- **o_chunk_size** (*int, default=None*) – if given, origin locations will be analyzed in chunks of the specified length to avoid memory errors.

`prepare.prepare_helpers.geojson_to_feature_class_arc`***(geojson_path, geom_type, encoding='utf8', unique_id=None)***

Converts geojson to feature class in memory and adds unique_id attribute if provided

Parameters:

- **geojson_path** (*str*) – path to geojson file

- **geom_type** (*str*) – The geometry type to convert from GeoJSON to features. OPTIONS: POINT, MULTIPOINT, POLYLINE, POLYGON
- **encoding** (*str*) – name of the encoding used to decode or encode the file
- **unique_id** (*str*) – name of unique id column, Default is None

Returns: path to temporary feature class

Return type: temp_feature (str)

prepare.prepare_helpers.get_field_dtype(***in_table***, ***field***)
Helper function to map data types from arcgis type to pandas type

Parameters:
- **in_table** (*str*) – path to table
- **field** (*str*) – field of interest

Returns: dtype of input field in format pandas can use

prepare.prepare_helpers.get_filename(***file_path***)
Helper function to extract a file name from a file path

prepare.prepare_helpers.get_raster_file(***folder***)
Get the name of the raster from within the zip (the .img file), there should be only one

Parameters: **folder** (*str*) – path to folder containing raster data

Returns: path to raster file

prepare.prepare_helpers.is_gz_file(***filepath***)
Test if a file is zipped

Parameters: **filepath** (*str*) – path to file of interest

Returns: bool

prepare.prepare_helpers.lines_to_centrality(***line_features***, ***impedance_attribute***)
Using the "lines" layer output from an OD matrix problem, calculate
node centrality statistics and store results in a csv table.

Parameters:
- **line_features** (*str*) – ODMatrix/Lines feature layer
- **impedance_attribute** (*str*) – field

Returns: summarized df with centrality measure appended

Return type: sum_df (pd.DataFrame)

`prepare.prepare_helpers.lu_diversity`(***in_df, groupby_field, lu_field, div_funcs, weight_field=None, count_lu=None, regional_comp=False***)
Wrapper function to run the four land use diversity methods (Simpson, Shannon, Berger-Parker, or ENP)

Parameters:

- **in_df** (*pandas.DataFrame*) – Dataframe containing the data to be analyzed
- **groupby_field** (*str*) – The field in in_df used to group the data
- **lu_field** (*str*) – The field in in_df containing the land use categorical values
- **div_funcs** (*list*) – List of the functions to generate diversity indices
- **weight_field** (*str*) – The field used to generate weights
- **count_lu** (*int*) – Count of relevant land use classes
- **regional_comp** (*bool, default=False*) – Boolean indicating whether to perform a region-wide comparison

Returns: panadas.DataFrame

Notes

If weight_field is none, a field called "COUNT" containing row counts by groupby_field values is created and used for weighting.

`prepare.prepare_helpers.make_basic_features`(***bf_gdb, stations_fc, stn_id_field, stn_diss_fields, stn_corridor_fields, alignments_fc, align_diss_fields, align_corridor_name, stn_buff_dist='2640 Feet', align_buff_dist='2640 Feet', stn_areas_fc='Station_Areas', corridors_fc='Corridors', long_stn_fc='Stations_Long', preset_station_areas=None, preset_station_id=None, preset_corridors=None, preset_corridor_name=None, rename_dict={}, overwrite=False***)
In a geodatabase with basic features (station points and corridor alignments), create polygon feature classes used for standard mapping and summarization. The output feature classes include:

- buffered corridors,
- buffered station areas,
- a long file of station points, where each station/corridor combo is represented as a separate feature.

Parameters:

- **bf_gdb** (*str*) – Path to a geodatabase with key basic features, including stations and alignments
- **stations_fc** (*str*) – A point feature class in`bf_gdb` with station locations and columns indicating belonging in individual corridors (i.e., the column names reflect corridor names and flag whether the station is served by that corridor).

- **stn_id_field** (*str*) – A field in stations_fc that identifies stations (common to a single station area)
- **stn_diss_fields** (*list*) – Field(s) on which to dissolve stations when buffering station areas. Stations that reflect the same location by different facilities may be dissolved by name or ID, e.g. This may occur at intermodal locations. For example, where metro rail meets commuter rail - the station points may represent the same basic station but have slightly different geolocations.
- **stn_corridor_fields** (*list*) – The columns in stations_fc that flag each stations belonging in various corridors.
- **alignments_fc** (*str*) – Path to a line feature class in bf_gdb reflecting corridor alignments
- **align_corridor_name** (*str*) – A field in alignments_fc that identifies the corridor it belongs to.
- **align_diss_fields** (*list*) – Field(s) on which to dissolve alignments when buffering corridor areas.
- **stn_buff_dist** (*str, default="2640 Feet"*) – A linear unit by which to buffer station points to create station area polygons.
- **align_buff_dist** (*str, default="2640 Feet"*) – A linear unit by which to buffer alignments to create corridor polygons
- **stn_areas_fc** (*str, default="Station_Areas"*) – The name of the output feature class to hold station area polygons
- **corridors_fc** (*str, default="Corridors"*) – The name of the output feature class to hold corridor polygons
- **long_stn_fc** (*str, default="Stations_Long"*) – The name of the output feature class to hold station features, elongated based on corridor belonging (to support dashboard menus)
- **preset_station_areas** (*path or feature layer, default=None*) – Features that pre-define station areas will supplant simple station buffers where preset_station_id matches stn_id_field
- **preset_station_id** (*str, default=None*) – Key field for preset_station_areas to lookup and replace geometries in the stn_areas_fc output
- **preset_corridors** (*path or feature layer, default=None*) – Features that pre-define corridor areas will supplant simple alignment buffers where preset_corridor_name matches stn_id_field
- **preset_corridor_name** – (str, default=None): Key field for preset_corridors to lookup and replace geometries in the corridors_fc output
- **rename_dict** (*dict, default={}*) – If given, stn_corridor_fields can be relabeled before pivoting to create long_stn_fc, so that the values reported in the output “Corridor” column are not the column names,

but values mapped on to the column names (changing “EastWest” column to “East-West”, e.g.)

- **overwrite** (*bool*) – default=False

***`prepare.prepare_helpers.make_summary_features`*(bf_gdb, long_stn_fc, stn_areas_fc, stn_id_field, corridors_fc, cor_name_field, out_fc, stn_buffer_meters=804.672, stn_name_field='Name', stn_status_field='Status', stn_cor_field='Corridor', overwrite=False)**

Creates a single feature class for data summarization based on station area and corridor geographies. The output feature class includes each station area, all combined station areas, the entire corridor area, and the portion of the corridor that is outside station areas.

Parameters:

- **bf_gdb** (*str*) – Path to basic features gdb
- **long_stn_fc** (*str*) – path to long station points feature class (deprecated in lieu of patched features)
- **stn_areas_fc** (*str*) – path to station area polygons feature class
- **stn_id_field** (*str*) – id field linking stn_areas_fc and long_stn_fc
- **corridors_fc** (*str*) – path to corridors feature class
- **cor_name_field** (*str*) – name field for corridor feature class
- **out_fc** (*str*) – path to output feature class
- **stn_buffer_meters** (*num, default=804.672 [1/2 mile]*) –
- **stn_name_field** (*str, default="Name"*) – station name field
- **stn_status_field** (*str, default="Status"*) – status of station
- **stn_cor_field** (*str, default="Corridor*) – corridor field
- **overwrite** (*bool, default=False*) – flag indicating whether to overwrite existing copy

***`prepare.prepare_helpers.match_units_fields`*(d)**

Helper function to match units to a field

Parameters:	**d** (*dict*) – a dictionary of the format {unit_name: parcel_field, …}, where the unit_name is one of the unit names present in the permits_df data’s permits_units_field, and the parcel_field is the field name in the parcels_df corresponding to that unit. It should be used to identify non-building area fields in the parcels_df for which we can calculate a building area/unit for the multiplier. parcel_field can also take the form of a basic function (+,-,/,*) of a column, see Notes for specifications
Returns:	listed outputs of field and functions
Return type:	match_fields, match_functions (lists)

`prepare.prepare_helpers.merge_and_subset`(***feature_classes***, ***subset_fc***)
Helper function to merge a list of feature classes and subset them based on a provided area of interest subset feature class

Parameters:	• **feature_classes** (*list*) – list of (paths to) feature classes • **subset_fc** (*str*) – Path to subset feature class polygon
Returns:	path to an in-memory feature class of all combined fcs
Return type:	merge_fc (str)

`prepare.prepare_helpers.model_blockgroup_data`(***data_path***, ***bg_enrich_tbl_name***, ***bg_key***, ***fields='*'***, ***acs_years=None***, ***lodes_years=None***)
Fit linear models to block group-level total employment, population, and
commutes at the block group level, and save the model coefficients for future prediction

Parameters:	• **data_path** (*str*) – path to enriched block group data, with a fixed-string wild card for year (see Notes) • **bg_enrich_tbl_name** (*str*) – name of enriched block group table data see Notes • **bg_key** (*str*) – name of primary key to block group data • **fields** (*list*) – list of fields to use for processing • **acs_years** (*list*) – list of int years for which ACS variables (population, commutes) are present in the data • **lodes_years** (*list*) – list of int years for which LODES variables (employment) are present in the data

Notes

in bg_enrich_path, replace the presence of a year with the string "{year}". For example, if your enriched block group data for 2010-2015 is stored at "Data_2010.gdb/enriched", "PMT_2019.gdb/Enrichment_census_blockgroups", …, then bg_enrich_tbl_name = "Enrichment_census_blockgroups".

Returns:	Data frame of model coefficients
Return type:	coeffs (pandas.DataFrame)

`prepare.prepare_helpers.network_centrality`(***in_nd***, ***in_features***, ***net_loader***, ***name_field='OBJECTID'***, ***impedance_attribute='Length'***, ***cutoff='1609'***, ***restrictions=''***, ***chunk_size=1000***)
Uses Network Analyst to create and iteratively solve an OD matrix problem
to assess connectivity among point features.

The evaluation analyses how many features can reach a given feature and
what the total and average travel impedances are. Results are reported for traveling TO each feature (i.e. features as destinations), which may be significant if oneway or similar restrictions are honored.

Parameters:

- **in_nd** (*str*) – Path to NetworkDataset
- **in_features** (*str*) – Path to Feature Class or Feature Layer A point feature class or feature layer that will serve as origins and destinations in the OD matrix
- **net_loader** (*object*) – NetLoader; Provide network location loading preferences using a NetLoader instance.
- **name_field** (*str, default="OBJECTID"*) – A field in in_features that identifies each feature. Generally this should be a unique value.
- **impedance_attribute** (*str, default="Length"*) – The attribute in in_nd to use when solving shortest paths among features in in_features.
- **cutoff** (*str, default="1609"*) – A number (as a string) that establishes the search radius for evaluating node centrality. Counts and impedances from nodes within this threshold are summarized. Units are implied by the impedance_attribute.
- **restrictions** (*str, default=""*) – If in_nd includes restriction attributes, provide a semi-colon-separated string listing which restrictions to honor in solving the OD matrix.
- **chunk_size** (*int, default=1000*) – Destination points from in_features are loaded iteratively in chunks to manage memory. The chunksize determines how many features are analyzed simultaneously (more is faster but consumes more memory).

Returns: DataFrame

Return type: centrality_df

`prepare.prepare_helpers.parcel_ideal_walk_time`(***parcels_fc***, ***parcel_id_field***, ***target_fc***, ***target_name_field***, ***radius***, ***target_name***, ***overlap_type='HAVE_THEIR_CENTER_IN'***, ***sr=None***, ***assumed_mph=3***)
Estimate walk time between parcels and target features (stations, parks,
e.g.) based on a straight-line distance estimate and an assumed walking speed.

Parameters:

- **parcels_fc** (*str*) – Path to parcel feature class
- **parcel_id_field** (*str*) – A field that uniquely identifies features in parcels_fc
- **target_fc** (*str*) – Path to a feature class used to estimate straight line distance from parcels

- **target_name_field** (*str*) – A field that uniquely identifies features in target_fc
- **radius** (*str*) – A “linear unit” string for spatial selection (‘5280 Feet’, e.g.)
- **target_name** (*str*) – A string suffix included in output field names.
- **overlap_type** (*str, default="HAVE_THEIR_CENTER_IN"*) – A string specifying selection type (see ArcGIS)
- **sr** (*arcpy.SpatialReference*) – A spatial reference code, string, or object to ensure parcel and target features are projected consistently. If None, the spatial reference from parcels_fc is used.
- **assumed_mph** (*int/float*) – default=3; The assumed average walk speed expressed in miles per hour.

Returns: **DataFrame**
A data frame with columns storing ideal walk time data: nearest_{target_name}, min_time_{target_name}, n_{target_name}

Return type: walk_time_fc

`prepare.prepare_helpers.parcel_walk_time_bin`(***in_table*, *bin_field*, *time_field*, *code_block***)

Adds a field to create travel time categories in a new bin_field
based on the walk times recorded in a time_field and an extended if/else code_block that defines a simple function assignBin().

Parameters:

- **in_table** (*str*) – path to walk time table
- **bin_field** (*str*) – Name of bin field
- **time_field** (*str*) – Name of time field
- **code_block** (*str*) – Code block encapsulated as a string Defines a python function assignBin() with if/else statements to group times in time_field into bins to be stored as string values in bin_field.

`prepare.prepare_helpers.parcel_walk_times`(***parcel_fc*, *parcel_id_field*, *ref_fc*, *ref_name_field*, *ref_time_field*, *preselect_fc*, *target_name***)

For features in a parcel feature class, summarize walk times reported
in a reference features class of service area lines. Generates fields recording the nearest reference feature, walk time to the nearest reference feature, number of reference features within the service area walk time cutoff, and a minimum walk time “category” field.

Parameters:

- **parcel_fc** (*str*) – Path to the parcel features to which walk time estimates will be appended.
- **parcel_id_field** (*str*) – The field in parcel_fc that uniquely identifies each feature.

- **ref_fc** (*str*) – Path to a feature class of line features with travel time estimates from/to key features (stations, parks, etc.)
- **ref_name_field** (*str*) – A field in ref_fc that identifies key features (which station, e.g.)
- **ref_time_field** (*str*) – A field in ref_fc that reports the time to walk from each line feature from/to key features.
- **preselect_fc** (*str*) – A feature class used to subset parcel_fc prior to spatial join
- **target_name** (*str*) – A string suffix included in output field names.

Returns: **DataFrame**
A data frame with columns storing walk time data: nearest_{target_name}, min_time_{target_name}, n_{target_name}

Return type: walk_time_df

See also

parcel_ideal_walk_time

*`prepare.prepare_helpers.patch_basic_features`***(*station_areas_fc*, *corridors_fc*, *preset_station_areas=None*, *preset_station_id_field=None*, *existing_station_id_field=None*, *preset_corridors=None*, *preset_corridor_name_field=None*, *existing_corridor_name_field=None*)**
Modifies the basic features database to updata station area and/or corridor geometries based on provided preset features (i.e., custom areas that deviate from the simple buffers created by make_basice_features)

Parameters:

- **station_areas_fc** (*str or feature layer*) –
- **corridors_fc** (*str or feature layer*) –
- **preset_station_areas** (*str or feature layer, default=None*) – If provided, these geometries will be used to update station area features (StationAreas and SummaryAreas)
- **preset_station_id_field** (*str, default=None*) – The field in preset_station_areas that corresponds to "basic_features/StationAreas.Id." It is used to map new geometries into the StationAreas and SummaryAreas feature classes.
- **existing_station_id_field** (*str, default=None*) – The field in station_areas_fc represnting the primary key
- **preset_corridors** (*str or feature layer, default=None*) – If provided, these geometries will be used to update corridor features (Corridors and SummaryAreas)
- **preset_corridor_name_field** (*str, default=None*) – The field in preset_corridors that corresponds to

"basic_features/Corridors.Corridor." It is used to map new geometries into the Corridors and SummaryAreas feature classes.

- **existing_corridor_name_field** (*str, default=None*) – The field in corridors_fc representing the primary key

See also

make_basic_features make_summary_features

`prepare.prepare_helpers.patch_local_regional_maz`(***maz_par_df, maz_par_key, maz_df, maz_key***)

Create a region wide MAZ socioeconomic/demographic data frame based on parcel-level and MAZ-level data. Where MAZ features do not overlap with parcels, use MAZ-level data.

Parameters:	• **maz_par_df** (*pd.DatarFrame*) – df of parcel data aggregated up to MAZ level • **maz_par_key** (*str*) – MAZ common id key • **maz_df** (*pd.DataFrame*) – df of MAZ data • **maz_key** (*str*) – MAZ common id key
Returns:	MAZ data patched in with parcel level data rolled up to MAZ
Return type:	pd.DataFrame

`prepare.prepare_helpers.prep_feature_class`(***in_fc, geom, out_fc, use_cols=None, rename_dict=None, unique_id=None***)

Tidies up a provided feature class, removing unnecessary fields and mapping existing fields to prefered values

Parameters:	• **in_fc** (*str*) – path to input feature class • **geom** (*str*) – geometry type of input features • **out_fc** (*str*) – path to output feature class • **use_cols** (*list*) – list of column names to keep • **rename_dict** (*list*) – list of dictionaries mapping attributes by feature class • **unique_id** (*str*) – name of new field to add as unique identifier
Returns:	None

`prepare.prepare_helpers.prep_imperviousness`(***zip_path, clip_path, out_dir, transform_crs=None***)

Clean a USGS impervious surface raster by clipping to the bounding box of a study area and transforming the clipped raster to a desired CRS

Parameters:

- **zip_path** (*str*) – Path to a .zip folder of downloaded imperviousness raster (see the dl_imperviousness function)
- **clip_path** (*str*) – Path of study area polygon(s) whose bounding box will be used to clip the raster
- **out_dir** (*str*) – Path to a save location for the clipped and transformed raster
- **transform_crs** (*any type accepted by arcpy.SpatialReference(), default=None*) – Identifier of spatial reference to which to transform the clipped raster.

Returns: **File will be clipped, transformed, and saved to the save directory; the** save path will be returned upon completion

Return type: out_raster (str)

`prepare.prepare_helpers.prep_parcel_land_use_tbl`**(*parcels_fc*, *parcel_lu_field*, *parcel_fields*, *lu_tbl*, *tbl_lu_field*, *null_value=None*, *dtype_map=None*, ***kwargs*)**
Generate a table that combines parcels having detailed DOR use codes with generalized land use classifications.

Parameters:

- **parcels_fc** (*str*) – path to parcel feature class
- **parcel_lu_field** (*str*) – String; The column in parcels_fc with each parcel's DOR use code.
- **parcel_fields** – [String, …]; Other columns in parcels_fc (such as an ID field, e.g.) to retain alongside land use codes.
- **lu_tbl** – Path; A csv table of land use groupings to associated with each parcel, keyed by DOR use code.
- **tbl_lu_field** – String; The column in lu_tbl with DOR use codes.
- **null_value** – var or dict; default values for nulls in parcels_fc. If a dict is given, the keys are column names and values are default values to assign.
- **dtype_map** – {string: type}; Data type for data or columns. If any data type specifications are needed to properly parse lu_tbl provide them as a dictionary.
- ****kwargs** – Any other keyword arguments given are passed to the pd.read_csv method when reading lu_tbl.

Returns: DataFrame

Return type: par_df

`prepare.prepare_helpers.prep_parcels`(***in_fc*, *in_tbl*, *out_fc*, *fc_key_field=None*, *new_fc_key_field=None*, *tbl_key_field=None*, *tbl_renames=None*, ****kwargs***)

Starting with raw parcel features and raw parcel attributes in a table,
clean features by repairing invalid geometries, deleting null geometries, and dissolving common parcel ID's. Then join attribute data based on the parcel ID field, managing column names in the process.

Parameters:

- **in_fc** (*str*) – Path or feature layer; A collection of raw parcel features (shapes)
- **in_tbl** (*str*) – Path to a table of raw parcel attributes.
- **out_fc** (*str*) – The path to the output feature class that will contain clean parcel geometries with attribute columns joined.
- **fc_key_field** (*str, default="PARCELNO"*) – The field in in_fc that identifies each parcel feature.
- **new_fc_key_field** (*str, default=None*) – parcel common key used throughout downstream processing
- **tbl_key_field** (*str, default=None*) – The field in in_csv that identifies each parcel feature.
- **tbl_renames** (*dict, default=None*) – Dictionary for renaming columns from in_csv. Keys are current column names; values are new column names.
- **kwargs** – Keyword arguments for reading csv data into pandas (dtypes, e.g.)

Returns: None, updates parcel data

`prepare.prepare_helpers.prep_park_polys`(***in_fcs*, *out_fc*, *geom='POLYGON'*, *use_cols=None*, *rename_dicts=None*, *unique_id=None***)

Merges park polygon data into single feature class, creating a unique ID for each polygon and reformatting the table

Parameters:

- **in_fcs** (*list*) – list of paths to park feature classes
- **out_fc** (*str*) – path to output dataset
- **geom** (*str*) – geometry type of input features
- **use_cols** (*list*) – list of column names to keep
- **rename_dicts** (*list*) – list of dictionaries mapping attributes by feature class
- **unique_id** (*str*) – name of new field to add as unique identifier (defined in prepare_config)

Returns: None

`prepare.prepare_helpers.prep_transit_ridership`(***in_table***, ***rename_dict***, ***unique_id***, ***shape_fields***, ***from_sr***, ***to_sr***, ***out_fc***)

Converts transit ridership data to a feature class and reformats attributes.

Parameters:

- **in_table** (*str*) – xls file path
- **rename_dict** (*dict*) – dictionary of {existing: new attribute} names
- **unique_id** (*str*) – name of id field added
- **shape_fields** (*list*) – columns to be used as shape field (x,y coords)
- **from_sr** (*SpatialReference*) – the spatial reference definition for coordinates listed in 'shape_fields'
- **to_sr** (*SpatialReference*) – the spatial reference definition for output features
- **out_fc** (*str*) – path to the output feature class

Returns: Path

Return type: out_fc

`prepare.prepare_helpers.read_transit_file_tag`(***file_path***)

Extracts a tag from the file path that is used in the naming of "ON" and "OFF" counts per stop

Parameters: **file_path** (*str*) – Path to transit ridership file

Returns: String

Return type: tag

`prepare.prepare_helpers.read_transit_xls`(***xls_path***, ***sheet=None***, ***head_row=None***, ***rename_dict=None***)

Reads in the provided xls file and concatenates the sheets into a single data frame to resolve formatting issues and make the file's contents code-readable.

XLS File Desc: Sheet 1 contains header and max rows for a sheet (65536), data continue on subsequent sheets without header.

Parameters:

- **xls_path** (*str*) – Path to xls file
- **sheet** (*str, int, or list, default=None*) – sheet(s) in xls_path to concatentate. If None, read and concatenate all shees in the workbook.
- **head_row** (*int or list, default=None*) – Row(s) in xls_path that serve as table headers. Must be the same for all `sheet`(s). If None, all rows are read without specifying a header.
- **rename_dict** (*dict, default=None*) – Dictionary to map existing column names to more readable names

Returns: consolidated dataframe of data from xls

Return type: pd.Dataframe

`prepare.prepare_helpers.shannon_diversity`(***in_df*, *group_col*, *weight_col=None*, *total_col=None*, *pct_col=None*, *count_lu=None*, ***kwargs***)
Shannon index: borrowing from information theory, Shannon quantifies
the uncertainty in predicting the land use of a random one unit draw.

The higher the uncertainty, the higher the diversity. Ranges from 0 (only one land use present) to -log(1/|land uses|) (all land uses present in equal abundance).

This function is not intended to be run directly. Use lu_diversity.

`prepare.prepare_helpers.simpson_diversity`(***in_df*, *group_col*, *weight_col=None*, *total_col=None*, *pct_col=None*, *count_lu=None*, ***kwargs***)
Simpson index: mathematically, the probability that a random draw of
one unit of land use A would be followed by a random draw of one unit of land use B.

Ranges from 0 (only one land use present) to 1 (all land uses present in equal abundance)

This function is not intended to be run directly. Use lu_diversity.

`prepare.prepare_helpers.skim_to_graph`(***in_csv*, *source*, *target*, *attrs*, *create_using=<class 'networkx.classes.digraph.DiGraph'>*, *renames={}*, ***kwargs***)
Converts a long OD table from a csv into a networkx graph, such that each OD row becomes an edge in the graph, with its origin and destination added as nodes.

Parameters:

- **in_csv** (*str*) – Path to skim csv file
- **source** (*str*) – The origin field. If fields are renamed used renames, give the new name.
- **target** (*str*) – The destination field. If fields are renamed used renames, give the new name.
- **attrs** (*list*) – [String,…]; Column names containing values to include as edge attributes
- **create_using** (*networx graph constructor, default=nx.DiGraph*) – The type of graph to build from the csv.
- **renames** (*dict*) – Keys are column names in in_file, values are new names for those columns to appear as edge attributes.
- **kwargs** – Keywords to use when loading in_file with pd.read_csv.

Returns: networkx.Graph

`prepare.prepare_helpers.split_date`**(*df, date_field, unix_time=False*)**
Ingest date attribute and splits it out to DAY, MONTH, YEAR

Parameters:	• **df** (*pd.DataFrame*) – DataFrame with a date field • **date_field** (*str*) – column name • **unix_time** (*str*) – unix time stamp
Returns:	DataFrame reformatted to include split day, month and year
Return type:	df (pd.DataFrame)

`prepare.prepare_helpers.summarize_access`**(*skim_table, o_field, d_field, imped_field, se_data, id_field, act_fields, imped_breaks, units='minutes', join_by='D', chunk_size=100000, **kwargs*)**
Reads an origin-destination skim table, joins activity data,
and summarizes activities by impedance bins.

Parameters:	• **skim_table** (*str*) – Path to SKIM table • **o_field** (*str*) – Origin field • **d_field** (*str*) – Destination field • **imped_field** (*str*) – Impedance field • **se_data** (*str*) – Path to socioeconomic data table • **id_field** (*str*) – se_data's id field • **act_fields** (*list*) – activity type fields (job types, e.g.) • **imped_breaks** (*int/float*) – list of break points by time • **units** (*str, default="minutes"*) – cost units of imped_field • **join_by** (*str, default="D"*) – join se_data based on o_field ("O") or d ("D") • **chunk_size** (*int, default=100000*) – number of rows to process simultaneously (larger values will finish faster by require more memory) • **kwargs** – Keyword arguments for reading the skim table
Returns:	Path
Return type:	out_table

`prepare.prepare_helpers.symmetric_difference`**(*target_fc, update_fc, out_fc_name*)**
If Advanced arcpy license not available this will calculate the
symmetrical difference of two sets of polygons.

Parameters:

- **target_fc** (*str*) – path to input feature class
- **update_fc** (*str*) – path to update feature class (same geometry type required)
- **out_fc_name** (*str*) – name of output feature class

Returns: path to output file in in_memory space

Return type: out_fc (str)

`prepare.prepare_helpers.taz_travel_stats`**(*od_table, o_field, d_field, veh_trips_field, auto_time_field, dist_field, taz_df, taz_id_field, hh_field, jobs_field, chunksize=100000, **kwargs*)**
Calculate rates of vehicle trip generation, vehicle miles of travel, and
average trip length.

Parameters:

- **od_table** (*str*) – Path to a csv table containing origin-destination information, including number of vehicle trips, travel time by car, travel distance by car, and travel time by transit
- **o_field** (*str*) – Field identifying origins in od_tables
- **d_field** (*str*) – Field identifying destinations in od_tables
- **veh_trips_field** (*str*) – Field recording number of vehicle trips in od_tables
- **auto_time_field** (*str*) – Field recording highway travel time in od_tables
- **dist_field** (*str*) – Field recording highway travel distance (miles) in od_tables
- **taz_df** (*pandas.DataFrame*) – DataFrame of TAZ economic and demographic data
- **taz_id_field** (*str*) – Field identifying TAZs in taz_df (corresponds to o_field and d_field)
- **hh_field** (*str*) – Field recording number of households in taz_df
- **jobs_field** (*str*) – Field recording number of jobs in taz_df
- **chunksize** (*int, default=100000*) – Number of OD rows to process at one time. More is faster but users more memory.
- **kwargs** – Keyword arguments to pass to pandas.read_csv method when loading od data

Returns: **Table of vehicle trip generation rates, trip legnths, and VMT**
estimates by TAZ.

Return type: taz_stats_df (pd.DataFrame)

`prepare.prepare_helpers.transit_skim_joins`***(taz_to_tap, tap_to_tap, out_skim, o_col='OName', d_col='DName', imp_col='Minutes', origin_zones=None, destination_zones=None, total_cutoff=inf)***
Creates a full skim from TAZ to TAZ by transit based on TAP to TAP skims and TAZ to TAP access/egress skims. TAP = transit access point.

This function assumes taz_to_tap and tap_to_tap have identical column headings for key fields.

Parameters:

- **taz_to_tap** (*str*) – Path to a csv OD table with estimated impedances between TAZs and accessible TAPs. This represents travel time outside the transit vehicle.
- **tap_to_tap** (*str*) – Path to a csv OD table with estimated impdedances between TAPs. This represents travel time in the transit vehicle.
- **out_skim** (*str*) – Path to an output csv OD table with estimated impedances between TAZs, inclusive of in-vehicle and out-of-vehicle impedance.
- **o_col** (*str, default="OName"*) – the column in taz_to_tap and tap_to_tap that identifies the origin-end feature of each record.
- **d_col** (*strdefault="OName"*) – the column in taz_to_tap and tap_to_tap that identifies the destination-end feature of each record.
- **imp_col** (*str, default="Minutes"*) – the column in taz_to_tap and tap_to_tap that records the impedance for each OD pair.
- **origin_zones** (*list, default=None*) – To limit the output OD table to specific origin locations, provide the origin TAZ id's in a list.
- **destination_zones** (*list, default=None*) – same as origin_zones except for destination TAZs.
- **total_cutoff** (*numeric, default=np.inf*) – If given, the output OD table will be truncated to include only OD pairs having estimated impedances less than or equal to the cutoff.

Returns: results are stored in a new csv table at out_skim

Return type: None

`prepare.prepare_helpers.udb_line_to_polygon`***(udb_fc, county_fc, out_fc)***
Uses the urban development boundary line to bisect the county boundary and generate two polygon output features.

During processing the UDB line features are dissolved into a single feature - this assumes all polylines in the shape file touch one another such that a single cohesive polyline feature results.

This function also assumes that the UDB will only define a simple bi-section of the county boundary. If the UDB geometry becomes more complex over time, modifications to this function may be needed.

Parameters:
- **udb_fc** (*str*) – Path to the udb line features.
- **county_fc** (*str*) – Path to the county boundary polygon. This is expected to only include a single polygon encompassing the entire county.
- **out_fc** (*str*) – Path to the output feature class.

Returns: Path to the output feature class

Return type: out_fc (str)

`prepare.prepare_helpers.update_dict`(***d***, ***values***, ***tag***)

Adds new key:value pairs to dictionary given a set of values and
tags in the keys. Only valuable in support of the transit ridership analaysis

Parameters:
- **d** (*dict*) – existing dictionary
- **values** (*list*) – output value names
- **tag** (*str*) – unique tag found in a column

Returns: updated with new key value pairs

Return type: d (dict)

`prepare.prepare_helpers.update_transit_times`(***od_table***, ***out_table***, ***competing_cols=[]***, ***out_col=None***, ***replace_vals={}***, ***chunksize=100000***, *****kwargs***)

Opens and updates a csv table containing transit travel time estimates between origin-destination pairs as indicated by the provided parameters.

Parameters:
- **od_table** (*str*) – path to a csv of OD data that includes transit travel time estimates
- **out_table** (*str*) – path to the new output csv table containing updated values
- **competing_cols** (*list, default=[]*) – The minimum value among competing columns will be written to out_col
- **out_col** (*str, default=None*) – A new column to be populated with updated transit travel time estimates based on competing_cols comparisons and value replacement indicated by replace_vals
- **replace_vals** (*dict, default={}*) – A dict whose keys indicate old values (those to be replaced) and whose values indicate new values.
- **chunksize** (*int, default=100000*) – The number of rows to process at a time (to accomodate large files)
- **kwargs** – Keyword arguments to pass to the pandas read_csv method

Returns: A new csv table with updated transit times is stored at the path specified

Return type: out_table (str)

`prepare.prepare_helpers.validate_json`**(*json_file*, *encoding='utf8'*)**
Check for valid json file

Parameters:
- **json_file** (*str*) – path to file
- **encoding** (*str*) – name of the encoding used to decode or encode the file

Returns: json deserialized to python object

`prepare.prepare_helpers.validate_weights`**(*weights*)**
Helper function to validate weights provided to contiguity index calcs are properly formatted

prepare_osm_networks module

The prepare_osm_networks module is similar to prepare_helpers except that the classes and functions defined here focus specificalluy on creating network datasets from raw OSM line features obtained using the downloader module. This module establishes a functional approach to standardize the components of network dataset development and configuration. Consistent schemas allow network dataset templates to be used for building similar network datasets with different OSM download vintages. This makes analytical replication as well as refinements and updates to network attributes and parameters over time relatively simple.

`prepare.prepare_osm_networks.classify_bikability`**(*bike_edges*)**
Adds two fields to cleaned bike edge features: "bikability" and "cycleway". The former assigns a "level of traffic stress" (LTS) score to each facility based on its facility type (least comfortable facilties score lowest on a range from 1 to 4). The latter field tags facilities with cycleway facilities on them. These details are used in building and solving the biking network dataset.

Parameters: **bike_edges** (*str*) – Path to bike network edge features.

Returns: bike_edges (str)

`prepare.prepare_osm_networks.import_OSM_shape`**(*osm_fc*, *to_feature_dataset*, *fc_name=None*, *overwrite=False*, *wc=None*, *field_mapping=None*)**
A simple function to facilitate the transfer of osm features in a shapefile to a feature dataset for use in a network dataset.

Parameters:
- **osm_fc** (*str*) – The path to the osm features to be added to the feature dataset.

- **to_feature_dataset** (*str*) – The path to the output feature dataset where network features are stored and a network dataset will eventually be constructed.
- **fc_name** (*str, default=None*) – The name of the output feature class to be created. If None, the name will match that if osm_fc.
- **overwrite** (*bool, default=False*) – If True, if there is already a feature class in to_feature_dataset with the same fc_name (as provided or implied by osm_fc), it will be deleted and replaced.
- **wc** – (str, default=None): A where clause to only transfer select features from osm_fc to the output feature class.
- **field_mapping** (*arcpy.FieldMappings, default=None*) – An arcpy FieldMappings object that handles field naming, merging, etc.

Returns: **Path to a new feature class in to_feature_dataset, transferring** features from osm_fc.

Return type: net_source_fc (str)

***prepare.prepare_osm_networks.make_network_dataset*(*template_xml, out_feature_dataset, net_name='osm_ND'*)**

Make a network dataset from a template xml file. The features in out_feature_dataset must be consistent with those used in the network dataset used to creat the xml template.

Parameters:

- **template_xml** (*str*) – Path to a network dataset xml template
- **out_feature_dataset** (*str*) – Path to the feature dataset where network source features are stored and where the network dataset will be created.
- **net_name** (*str, default="osm_ND"*) – The name of the network dataset to be created.

Returns: **Creates a network dataset in out_feature_dataset based on the** specifications in template_xml.

Return type: None

See also

makeNetworkDatasetTemplate

***prepare.prepare_osm_networks.make_network_dataset_template*(*from_nd, template_xml*)**

Make a network dataset template from an existing network dataset. The template can be used to construct new network datasets using the same specifications later.

Parameters:	• **from_nd** (*str*) – Path to the existing network dataset from which the template will be created. • **template_xml** (*str*) – Path to an output template (xml) file.	
Returns:	**Creates a network dataset xml template at the path specified by** template_xml	
Return type:	nd_template (xml)	

See also

makeNetworkDataset

TOC Tool Build Scripts

Builder

The 'builder' module serves as the final processing tool in developing the TOC toolkit databases.

This module performs the heavy lifting to build out the PMT geodatabases by timeframe for ingestion by AGOL for mapping and reporting in the Experience Builder website.

For each PMT_YYYY geodatabase, a Snapshot summary is generated, creating wide tables and calculating new attributes by aggregating data up from lower spatial scales. Tables long on categorical information are created as well. Trend and NearTerm geodatabases from the yearly geodatabases, utilizing the similar procedures and creating tables long on year and calculating difference values for start and end time frames (Trend: start = 2014, end=current year; NearTerm: start = current year, end = forecasted near term from permits) These procedures generate all metrics reported in the TOC tool website as a prior calculations and transformations, supporting more performant mapping and reporting within the site.

This module depends on the build_helper and build_config modules primarily.

Functions:
year_to_snapshot:

- process/task to take year gdb, making the geometries wide where needed and metrics long on categorical data

snapshot_to_trend: (used to process Trend and NearTerm geodatabases with alterations to inputs)

- process/task to stack snapshot metrics, making them long on year, ref_YEAR_Snapshot
- data also generate difference values by metric within the summary geometries, blocks, MAZ, TAZ

`build.builder.process_all_snapshots`(years)
Helper function to iterate all years and generate snapshot databases for list of years provided

`build.builder.process_year_to_snapshot`**(*year*)**
Process cleaned yearly data to a Snapshot database

Procedure:

1. copies feature datasets into a temporary geodatabase
2. performs a series of permanent joins of tabular data onto feature classes making wide tables
3. calculates a series of new fields in the existing feature classes
4. calculates a dataframe of region wide parcel level statistics
5. Intersects a series of geometries together, allowing us to aggregate and summarize data from higher to lower spatial scales
6. Enrichment of existing feature class tables with the information from higher spatial resolution, in effect widening the tables (ex: roll parcel level data up to blocks, or parcel level data up to Station Areas)
7. Generate new tables that are long on categorical information derived from the intersections (ex: pivot TOT_LVG_AREA on Land Use, taking the sum of living area by land use)
8. Create separate access by mode tables (bike, walk, transit, auto)
9. Calculate new attributes based on region wide summaries
10. Calculate additional attributes for dashboards that require all previous steps to be run
11. If successful, replace existing copy of Snapshot with newly processed version.

Returns: None

`build.builder.process_years_to_trend`**(*years*, *tables*, *long_features*, *diff_features*, *base_year=None*, *snapshot_year=None*, *out_gdb_name=None*)**
Utilizing a base and snapshot year, trend data are generated for the associated time period.

Procedure:

1. creates a a blank output workspace with necessary feature dataset categories uniquely named
2. generates tables long on year for all tabular data and summary areas
3. **generated difference tables for all tabular data summary features**
4. (Summary Areas, Census Blocks, MAZ, and TAZ)
5. upon completion, replace existing copy of Trend/NearTerm gdb with newly processed version.

Builder support

build_helpers module

The build_helper module provides a handful of functions to facilitate consistent applications of geoprocessing methods and table operations to combine feature classes, consolidate columns, calculate derivative values, elongate tables to support serial reporting, compare similar tables across time periods and calculate differences, etc. It directly supports the builder module, and many of the arguments taken by functions defined here assume properly-formatted objects as specified in the build_config module.

`build.build_helper.add_year_columns`(***in_gdb*, *year***)
Helper function ensuring the year attribute is present in all layers/tables

Parameters:
- **in_gdb** (*str*) – path to geodatabase
- **year** (*int*) – value to be calculated as year

Returns: None

`build.build_helper.alter_fields`(***table_list*, *field*, *new_field_name***)
Helper function to rename a field found in multiple tables created to handle RowID used for summary features

Parameters:
- **table_list** (*list*) – list of paths to tables containing the field of interest
- **field** (*str*) – current field name
- **new_field_name** (*str*) – desired field name to replace existing

Returns: None

`build.build_helper.apply_field_calcs`(***gdb*, *new_field_specs*, *recalculate=False***)
Helper function that applies field calculations, adding a new field to a table

Parameters:
- **gdb** (*str*) – path to geodatabase containing table to have new calc added
- **new_field_specs** (*list*) – list of dictionaries specifying table(s), new_field, field_type, expr, code_block
- **recalculate** (*bool*) – flag to rerun a calculation if the field already exsits in the table; currently unused

Returns: None

`build.build_helper.build_access_by_mode`(***sum_area_fc*, *modes*, *id_field*, *out_gdb*, *year_val***)
Helper function to generate access tables by mode.

Parameters:
- **sum_area_fc** (*str*) – path to summary area feature class
- **modes** (*list*) – modes of travel
- **id_fields** (*list*) – fields to be used as index
- **out_gdb** (*str*) – path to output geodatabase
- **year_val** (*int*) – value to insert for year

Returns: None

`build.build_helper.build_enriched_tables`**(*gdb*, *fc_dict*, *specs*)**

Helper function used to enrich and/or elongate data for a summarization area. Enrichment is based on intersection of disaggregate features with summarization area features. Elongation melts tables for serial reporting (square footage by land use per summarization area, e.g.)

Parameters:

- **gdb** (*str*) – path to geodatabase where outputs are written
- **fc_dict** (*dict*) – dictionary returned from build_intersections
- **specs** (*list of dicts*) – list of dictionaries specifying sources, grouping, aggregations, consolidations, melts/elongations, and an output table (this is used by the try/except clause to make a new table (elongation) or append to an existing feature class (widening)

Returns: None

`build.build_helper.build_intersections`**(*gdb*, *enrich_specs*)**

Helper function that performs a batch intersection of polygon feature classes

Parameters:

- **gdb** (*str*) – path to geodatabase that contains the source data
- **enrich_specs** (*list*) – list of dictionaries specifying source data, groupings, aggregations, consolidations, melt/elongation, and boolean for full geometry or centroid use in intersection

Returns: Dictionary of the format {summ fc: {disag_fc: path/to/intersection}} will return multiple results for each summ_fc if more than one intersection is made against it.

Return type: dict

`build.build_helper.finalize_output`**(*intermediate_gdb*, *final_gdb*)**

Takes an intermediate GDB path and the final GDB path for that data and replaces the existing GDB if it exists, otherwise it makes a copy of the intermediate GDB and deletes the original

Parameters:

- **intermediate_gdb** (*str*) – path to file geodatabase
- **final_gdb** (*str*) – path to file geodatabase, cannot be the same as intermediate

Returns: None

`build.build_helper.join_attributes`**(*to_table*, *to_id_field*, *from_table*, *from_id_field*, *join_fields='*'*, *null_value=0.0*, *renames=None*, *drop_dup_cols=False*)**

Helper function to join attributes of one table to another

Parameters:
- **to_table** (*str*) – path to table being extended
- **to_id_field** (*str*) – primary key
- **from_table** (*str*) – path to table being joined
- **from_id_field** (*str*) – foreign key
- **join_fields** (*list/str*) – list of fields to be added to to_table; Default: "*", indicates all fields are to be joined
- **null_value** (*int/str*) – value to insert for nulls
- **renames** (*dict*) – key/value pairs of existing field names/ new field names
- **drop_dup_cols** (*bool*) – flag to eliminate duplicated fields

Returns: None

***build.build_helper.list_fcs_in_gdb*()**
Generator to iterate over all feature classes ina geodatabase. Assumes you set your arcpy.env.workspace to a gdb before calling

Yields: path to feature class

***build.build_helper.make_reporting_gdb*(*out_path*, *out_gdb_name=None*, *overwrite=False*)**
Helper function to create a temporary geodatabase to hold data as its procesed

Parameters:
- **out_path** (*str*) – path to folder
- **out_gdb_name** (*str*) – name of geodatabase, Default is None, resulting in a unique name
- **overwrite** (*bool*) – flag to delete an existing geodatabase

Returns (str):
path to output geodatabase

***build.build_helper.make_snapshot_template*(*in_gdb*, *out_path*, *out_gdb_name=None*, *overwrite=False*)**
Helper function to copy yearly feature classes into a reporting geodatabase; copies all feature datasets from corresponding clean data workspace

Parameters:
- **in_gdb** (*str*) – path to clean data workspace
- **out_path** (*str*) – path where snapshot template gdb is written
- **out_gdb_name** (*str*) – optional name of output gdb
- **overwrite** (*bool*) – boolean flag to overwrite an existing copy of the out_gdb_name

Returns (str):
path to the newly created reporting geodatabase

`build.build_helper.make_trend_template`(***out_path*, *out_gdb_name=None*, *overwrite=False***)

Helper function to generate a blank output workspace with necessary feature dataset categories

Parameters:
- **out_path** (*str*) – path where trend template gdb is written
- **out_gdb_name** (*str*) – optional name of output gdb
- **overwrite** (*bool*) – boolean flag to overwrite an existing copy of the out_gdb_name

Returns (str):

path to the newly created reporting geodatabase

`build.build_helper.post_process_databases`(***basic_features_gdb*, *build_dir***)

Copies in basic features gdb to build dir and cleans up FCs and Tables with SummID to RowID. Finally deletes the TEMP folder generated in the build process

Parameters:
- **basic_features_gdb** (*str*) – path to the basic features geodatabase
- **build_dir** (*str*) – path to the build directory

Returns: None

`build.build_helper.process_joins`(***in_gdb*, *out_gdb*, *fc_specs*, *table_specs***)

Joins feature classes to associated tabular data from year set and appends to FC in output gdb, making single feature classes with wide attribute tables for each geographic unit of analysis.

Parameters:
- **in_gdb** (*str*) – path to geodatabase containing input feature classes and tables
- **out_gdb** (*str*) – path to geodatabase that will hold output feature classes
- **fc_specs** (*list*) – list of tuples defining the feature classes from in_gdb to retain in out_gdb. Each tuple consists of the feature class name, unique ID attribute, and the feature dataset (within in_gdb) where it is located.
- **table_specs** (*list*) – list of tuples defining the tables from in_gdb to join to feature classes in out_gdb. Each tuple consists of table name, unique ID attribute, fields to retain ("*" = all fields), and a dictionary of field renaming specs ({} must be given if no renaming is desired). Columns from each table are joined to the FC associated with the table name (based on the presence of the FC name in the table name)

Returns: List of paths to joined feature classes ordered as Blocks, Parcels, MAZ, TAZ, SummaryAreas, NetworkNodes

Return type: list

`build.build_helper.sum_parcel_cols`(***gdb*, *par_spec*, *columns***)
Helper function to summarize a provided list of columns for the parcel layer, creating region wide statistics

Parameters:
- **gdb** (*str*) – path to geodatabase that parcel layer exists in
- **par_spec** (*tuple*) – tuple of format (fc name, unique id column, feature dataset location)
- **columns** (*list*) – string list of fields/columns needing summarization

Returns: pandas.Dataframe

`build.build_helper.summarize_attributes`(***in_fc*, *group_fields*, *agg_cols*, *consolidations=None*, *melt_col=None***)
Helper function to perform summarizations of input feature class defined by the group, agg, consolidate, and melt columns/objects provided

Parameters:
- **in_fc** (*str*) – path to feature class, typically this will be the result of an intersection of a summary fc and disaggregated fc
- **group_fields** (*list*) – list of Column objects with optional rename attribute
- **agg_cols** (*list*) – list of AggColumn objects with optional agg_method and rename attributes
- **consolidations** (*list*) – list of Consolidation objects with optional consolidation method attribute
- **melt_col** (*list*) – list of MeltColumn objects with optional agg_method, default value, and DomainColumn object

Returns: pandas.Dataframe object with all data summarized according to specs

`build.build_helper.tag_filename`(***filename*, *tag***)
Helper method to add a suffix to the end of a filename

Parameters:
- **filename** (*str*) – path to file or filename string
- **tag** (*str*) – string suffix to append to end of filename

Returns: updated filepath or filename string with suffix appended

Return type: str

`build.build_helper.unique_values`(***table*, *field***)
Helper function to return all unique values for a provided field/column

Parameters:
- **table** (*str*) – path to table of interest
- **field** (*str*) – field name of interest

Returns: sorted unique values from the field

TOC Tool Core Module (PMT)

The PMT module provides a centralized repository of commonly used classes and functions supporting the development of the TOC toolkit. Many of these focus on file management and mundane but critical procedural support. It also sets constant variables for relative file locations and analysis parameters such as the years of data to be analyzed and reported.

***class* `PMT.AggColumn`(*name, agg_method=<built-in function sum>, default=0.0, rename=None, domain=None*)**
A Column that will be aggregated in a downstream group by/agg process.

`name`
the name of the column to aggregate

Type: str

`agg_method`
the aggregation method to apply (see pandas.DataFrame.agg)

Type: str or callable

`default`
values to replace missing/nan records with prior to aggregation

Type: numeric, default=0.0

`rename`
see Column

Type: str, default=None

`domain`
see Column, DomainColumn

Type: object, default=None

***class* `PMT.And`(*criteria*)**
A stack of Comp objects that define conditions that must all evaluate to True when applied against a given value.

`criteria`
a list of Comp objects.

Type: list

`eval`(***vals**)
Evaluate the comparison of this value(s) against all values in self.criteria using the comparison methods specified for each criterion.

Parameters:
- ***vals** – one more values to evaluate against self.criteria. Returns a boolean
- **that has True values only where all criteria are met.** (*vector*) –

class *`PMT.CollCollection`*(***name, input_cols, agg_method=<built-in function sum>, default=0.0, domain=None***)
A building-block class that is a child of AggColumn (i.e., it anticipates a downstream aggregateion process). The CollCollection specifies multiple column parameters to facilitate various column procedures. This class is never initialized. See Consolitation, MeltColumn.

`Name`
see AggColumn

Type: str

`input_cols`
Names of columns that will be used to generate a new column

Type: list

`agg_method`
see AggColumn

Type: str or callabe

`default`
One or more default values to apply to input_cols to replace missing/nan values. If multiple defaults are given, specify them as a dictionary whose keys are input_cols and whose values are corresponding defaults.

Type: var or dict

`defaultsDict`()
If self.

class *`PMT.Column`*(***name, default=0.0, rename=None, domain=None***)
A class that defines key attributes of an existing or prospective column to facilitate processing

`name`
The original name of the column

Type: str

`default`
The default value to fill missing/nan values. Type depends on column data type.

Type: var

rename

If given, this column will be renamed in a dataframe.

Type: str, default=None

domain

DomainColumn; If given, values in this Column will be used to define a new column that assings them to an Ordinal domain. This is primarily useful to enforce ordering of rows in a dataframe by category.

Type: obj, default=None

***apply_domain*(*df, col=None*)**

Method to create the domain column associated with this Column in the given dataframe.

Parameters:
- **df** (*pd.DataFrame*) – a dataframe having this Column as a member
- **col** (*str, default=None*) – if None, this Column's values are mapped onto its Domain attribute to generate a new column self.Domain.name. This argument is generally not needed, but some child classes rely on the ability to specify another column as the domain reference.

***class PMT.Comp*(*comp_method, v*)**

A naive class that allows string-based specification of comparison operators for process configuration support purposes.

comp_method

Comparison methods provided as strings: - "==" is 'equals' or __eq__() - "!=" is 'not equal to' or __ne__() - "<" is 'less than' or __lt__() - "<=" is 'is less than or equal to' or __le__() - ">" is 'greater than' or __gt__() - ">=" is 'greater than or equal to' or __ge__()

Type: str

v

the value to compare other values against using the comp_method

Type: var

***eval*(*val*)**

Evaluate the comparison of this value against self.v

Parameters: **val** (*var*) – the value to comparea against self.v using self.comp_method

***class PMT.Consolidation*(*name, input_cols, cons_method=<built-in function sum>, agg_method=<built-in function sum>, default=0.0*)**

A column collection that collapses multiple columns into a single column through a row-wise aggregation.

name

Name of the new column to be created when input_cols are consolidated.

Type: str

input_cols

Names of columns that will be consolidated into a new column

Type: list

cons_method

the aggregation method to apply row-wise to generate a new column from input_cols (see pandas.DataFrame.agg)

Type: str or callable

agg_method

see AggColumn

Type: str or callabe

default

see CollCollection

Type: var or dic

class *PMT.DomainColumn*(***name, default=- 1, domain_map={}***)

A Column that applies a domain mapping (original values to ordinal values) to another Column.

name

the name of this DomainColumn when added to any data frame

Type: str

default

the value to apply to missing values (those not found in domain_map)

Type: var

domain_map

key-value pairs that relate original values in a reference Column to ordinal values recorded in this DomainColumn.

Type: dict

class *PMT.MeltColumn*(***label_col, val_col, input_cols, agg_method=<built-in function sum>, default=0.0, domain=None***)

A column collection that collapses multiple columns into a single column through table elongation (melting).

label_col

Name of the new column to be created to store input_cols headings when the table is melted.

Type: str

`val_col`
Name of the new column to be created to store input_cols values when the table is melted.

Type: str

`input_cols`
Names of columns that will be melted into the new columns label_col and val_col.

Type: list

`agg_method`
see AggColumn

Type: str or callabe

`default`
see CollCollection

Type: var or dic

`domain`
see Column, DomainColumn

Type: object, default=None

*`apply_domain`***(*df*)**
DomainColumn specifications are applied in the same way as other Column objects, but for the MeltColumn class, self.label_col is used for mapping the domain.

class *`PMT.NetLoader`***(*search_tolerance*, *search_criteria*, *match_type='MATCH_TO_CLOSEST'*, *append='APPEND'*, *snap='NO_SNAP'*, *offset='5 Meters'*, *exclude_restricted='EXCLUDE'*, *search_query=None*)**
A naive class for specifying network location loading preferences. Simplifies network functions by passing loading specifications as a single argument. This class does no validation of assigned preferences.

All attributes correspond to arcpy attribute specifications defined here: https://pro.arcgis.com/en/pro-app/latest/tool-reference/network-analyst/add-locations.htm

`search_tolerance`
Type: str

`search_criteria`
Type: str

`match_type`
Type: str

`append`
Type: str

`snap`

Type: str

`offset`

Type: str

`exclude_restricted`

Type: str

`search_query`

Type: str

class *`PMT.Or`*(***vector, criteria***)

A stack of Comp or 'And' objects that define conditions, any of which must evaluate to True when applied agaianst a vector of values.

`vector`

A vector of values to test against self.criteria.

Type: np.array-like

`criteria`

[Comp, And, …] Criteria that will be applied to check if values in vector meet any.

Type: list

*`eval`***()**

Returns a boolean vector like self.vector that has True values where the values in self.vector evaluate to True for any criterion given in self.criteria.

class *`PMT.ServiceAreaAnalysis`*(***name, network_dataset, facilities, name_field, net_loader***)

Specifies elements of a Network Analyst Service Area Problem and

provides a method for solving and exporting service arae lines and polygons.

Parameters:

- **name** (*str*) – Name of the service area problem to generate
- **network_dataset** (*str*) – Path to the network dataset that will determine the service area
- **facilities** (*str or Feature Layer*) – Path to point features representing locations for which service areas will be created.
- **name_field** (*str*) – A field in facilities that identifies each facility (can tag a group of facilities)
- **net_loader** (*object*) – NetLoader; specifications for how to relate facilities to features in the network_dataset.

`solve`(***imped_attr, cutoff, out_ws, restrictions='', use_hierarchy=False, net_location_fields=''***)

Create service area lines and polygons for this object's facilities.

Parameters:	• **imped_attr** (*str*) – The impedance attribute in this object's network_dataset to use in estimating service areas. • **cutoff** (*numeric*) – The size of the service area to create (in units corresponding to those used by imped_attr). • **out_ws** (*str*) – Path to a workspace where service area feature class oupututs will be stored. • **restrictions** (*str*) – A semi-colon-separated list of restriction attributes in self.network_dataset to honor when creating service areas. • **use_hierarchy** (*bool, default=False*) – • **net_location_fields** (*str, default=""*) – if self.facilities have pre-calculated network location fields, list the fields in order ("SourceOID", "SourceID", "PosAlong", "SideOfEdge", "SnapX", "SnapY", "Distance",). This speeds up processing times since spatial analysis to load locations on the network is not needed.

`PMT.add_unique_id`(***feature_class, new_id_field='ProcessID'***)
Adds a unique incrementing integer value to a feature class and returns that name

Parameters:	• **feature_class** (*str*) – Path to a feature class • **new_id_field** (*str, default="ProcessID"*) – Name of new id field.
Returns:	Name of new id field
Return type:	new_id_field (str)

`PMT.col_multi_index_to_names`(***columns, separator='_'***)
For a collection of columns in a data frame, collapse index levels to flat column names. Index level values are joined using the provided separator.

Parameters:	• **columns** (*pandas.Index*) – The columns to flatten (i.e, df.columns) • **separator** (*str*) – The string value used to flatten multi-level column names
Returns:	pd.Index
Return type:	flat_columns

`PMT.copy_features`(***in_fc, out_fc, drop_columns=[], rename_columns={}***)
Copy features from a raw directory to a cleaned directory.
During copying, columns may be dropped or renamed.

Parameters:	• **in_fc** (*str*) – Path to input feature class • **out_fc** (*str*) – Path to output feature class

- **drop_columns** (*list*) – [String,…]; A list of column names to drop when copying features.
- **rename_columns** (*dict*) – {String: String,…} A dictionary with keys that reflect raw column names and values that assign new names to these columns.

Returns: Path to the file location for the copied features.

Return type: out_fc (str)

`PMT.count_rows`(***in_table*, *groupby_field=None*, *out_field=None*, *skip_nulls=False*, *null_value=None*, *inplace=True***)

Counts rows in a table.

Parameters:

- **in_table** (*str, feature layer, table view or DataFrame*) – (Path to) the table for which to return a row count
- **groupby_field** (*list, default=None*) – If given, the number of rows in the table with unique combinations of specified fields is returned.
- **out_field** (*str, default=None*) – If given, the count is added to features in the table
- **skip_nulls** (*bool, default=False*) – Control whether records using nulls are skipped.
- **null_value** (*var or dict*) – Replaces null values from the input with a new value. Can be provided as a dict to set null replacement values for specific columns.
- **inplace** (*bool, default=True*) – Only applies when out_field is provided. If True, in_table is updated in-place with a new field.

Returns: Int, DataFrame, or None

`PMT.dbf_to_df`(***dbf_file***)

Reads in dbf file and returns Pandas DataFrame object

Parameters: **dbf_file** – String; path to dbf file

Returns: pandas DataFrame object

`PMT.df_to_table`(***df*, *out_table*, *overwrite=False***)

Use a pandas data frame to export an arcgis table.

Parameters:

- **df** (*pandas.DataFrame*) – DataFrame
- **out_table** (*str*) – Path to output table
- **overwrite** (*bool, default=False*) –

Returns: Path

Return type: out_table (str)

`PMT.extend_table_df`(***in_table*, *table_match_field*, *df*, *df_match_field*, ***kwargs***)
Use a pandas data frame to extend (add columns to) an existing table based through a join on key columns. Key values in the existing table must be unique.

Parameters:
- **in_table** (*str, feature layer, or table view*) – Path to the existing table to be extended
- **table_match_field** (*str*) – The field in in_table on which to join values from df
- **df** (*pandas.DataFrame*) – The data frame whose columns will be added to in_table
- **df_match_field** (*str*) – The field in df on which join values to in_table
- **kwargs** – Optional keyword arguments to be passed to arcpy.da.ExtendTable.

Returns: None; in_table is modified in place

`PMT.featureclass_to_df`(***in_fc*, *keep_fields='*'*, *skip_nulls=False*, *null_val=0***)
Converts feature class/feature layer to pandas DataFrame object, keeping
only a subset of fields if provided and dropping all spatial data

Parameters:
- **in_fc** (*str*) – Path to a feature class
- **keep_fields** (*list, default="*"*) – Field names to return in the dataframe (“*” will return all fields)
- **skip_nulls** (*bool*) – Control whether records using nulls are skipped.
- **null_val** (*int, float, or dict*) – value to be used for nulls found in the data. Can be given as a dict of default values by field

Returns: pandas.Dataframe

`PMT.gen_sa_lines`(***facilities*, *name_field*, *in_nd*, *imped_attr*, *cutoff*, *net_loader*, *out_fc*, *from_to='TRAVEL_FROM'*, *overlap='OVERLAP'*, *restrictions=''*, *use_hierarchy=False*, *uturns='ALLOW_UTURNS'*, *net_location_fields=None***)
Creates service area lines around given facilities using a network dataset.

Parameters:
- **facilities** (*str or feature layer*) – The facilities for which service areas will be generated.
- **name_field** (*str*) – The field in facilities that identifies each location.
- **in_nd** (*str*) – Path to the network dataset
- **imped_attr** (*str*) – The name of the impedance attribute to use when solving the network and generating service area lines

- **cutoff** (*numeric*) – The search radius (in units of imped_attr) that defines the limits of the service area. If a list is given, the highest value defines the cutoff and all other values are used as break points, which are used to split output lines.
- **net_loader** (*obj*) – NetLoader; Location loading preferences
- **out_fc** (*str*) – Path to service area polygon feature class to be created
- **from_to** (*str, default="TRAVEL_FROM"*) – If “TRAVEL_FROM”, service areas reflect the reach of the network from facilities; if “TRAVEL_TO”, service areas reflec the reach of the network to the facilities. If not applying one-way restrictions, the outcomes are effectively equivalent.
- **overlap** (*str, deault="OVERLAP"*) – If “OVERLAP”, individual sets of line features for each facility will be generated; if “SPLIT”, line service area features are assigned to the nearest facility.
- **restrictions** (*str, default=None*) – Specify restriction attributes (oneway, e.g.) to honor when generating service area lines. If the restrictions are parameterized, default parameter values are used in the solve.
- **uturns** (*str, default="ALLOW_UTURNS"*) – Options are “ALLOW_UTURNS”, “NO_UTURNS”, “ALLOW_DEAD_ENDS_ONLY”, “ALLOW_DEAD_ENDS_AND_INTERSECTIONS_ONLY”
- **use_hierarchy** (*bool, default=False*) – If a hierarchy is defined for in_nd, it will be applied when solving the network if use_hierarchy is True; otherwise a simple, non-hierarchical solve is executed.
- **net_location_fields** (*list, default=None*) – If provided, list the fields in the facilities attribute table that define newtork loading locations. Fields must be provided in the following order: SourceID, SourceOID, PosAlong, SideOfEdge, SnapX, SnapY, Distance.

Returns: Path

Return type: out_fc (str)

See also

NetLoader

`PMT.gen_sa_polys`(***facilities*, *name_field*, *in_nd*, *imped_attr*, *cutoff*, *net_loader*, *out_fc*, *from_to='TRAVEL_FROM'*, *merge='NO_MERGE'*, *nesting='RINGS'*, *restrictions=None*, *use_hierarchy=False*, *uturns='ALLOW_UTURNS'*, *net_location_fields=None***)

Creates service area polygons around given facilities using a network dataset.

Parameters:

- **facilities** (*str or feature layer*) – The facilities for which service areas will be generated.
- **name_field** (*str*) – The field in facilities that identifies each location.

- **in_nd** (*str*) – Path to the network dataset
- **imped_attr** (*str*) – The name of the impedance attribute to use when solving the network and generating service area lines
- **cutoff** (*numeric*) – The search radius (in units of imped_attr) that defines the limits of the service area. If a list is given, the highest value defines the cutoff and all other values are used as break points, which are used to split output lines.
- **net_loader** (*obj*) – NetLoader; Location loading preferences
- **out_fc** (*str*) – Path to service area polygon feature class to be created
- **from_to** (*str, default="TRAVEL_FROM"*) – If “TRAVEL_FROM”, service areas reflect the reach of the network from facilities; if “TRAVEL_TO”, service areas reflec the reach of the network to the facilities. If not applying one-way restrictions, the outcomes are effectively equivalent.
- **restrictions** (*str, default=None*) – Specify restriction attributes (oneway, e.g.) to honor when generating service area lines. If the restrictions are parameterized, default parameter values are used in the solve.
- **uturns** (*str, default="ALLOW_UTURNS"*) – Options are “ALLOW_UTURNS”, “NO_UTURNS”, “ALLOW_DEAD_ENDS_ONLY”, “ALLOW_DEAD_ENDS_AND_INTERSECTIONS_ONLY”
- **use_hierarchy** (*bool, default=False*) – If a hierarchy is defined for in_nd, it will be applied when solving the network if use_hierarchy is True; otherwise a simple, non-hierarchical solve is executed.
- **net_location_fields** (*list, default=None*) – If provided, list the fields in the facilities attribute table that define newtork loading locations. Fields must be provided in the following order: SourceID, SourceOID, PosAlong, SideOfEdge, SnapX, SnapY, Distance.

Returns:	Path
Return type:	out_fc (str)

See also

NetLoader

`PMT.intersect_features`***(summary_fc, disag_fc, disag_fields='*', as_df=False, in_temp_dir=False, full_geometries=False)***

Creates a temporary intersected feature class for disaggregation of data

Parameters:

- **summary_fc** (*str*) – Path to path to polygon feature class with data to be disaggregated from
- **disag_fc** (*str*) – Path to polygon feature class with data to be disaggregated to

- **disag_fields** (*list*) – List of fields to pass over to intersect function
- **as_df** (*bool*) – If True, returns a data frame (table) of the resulting intersect. Otherwise returns the path to a temporary feature class
- **in_temp_dir** (*bool*) – If True, intersected features are stored in a temp directory, otherwise they are stored in memory
- **full_geometries** (*bool*) – If True, intersections are run against the complete geometries of features in disag_fc, otherwise only centroids are used.

Returns: Path to temp intersected feature class

Return type: int_fc (str)

`PMT.is_multipart`(***polygon_fc***)

Helper function to determine if polygon FC has multipart features

Parameters: **polygon_fc** (*str*) – Path to polygon feature class

Returns: bool

`PMT.iter_rows_as_chunks`(***in_table*, *chunksize=1000***)

A generator to iterate over chunks of a table for arcpy processing.

This method cannot be reliably applied to a table view or feature
layer with a current selection as it alters selections as part of the chunking process.

Parameters:

- **in_table** (*Table View or Feature Layer*) – The rows/features to process
- **chunksize** (*int, default=1000*) – The number of rows/features to process at a time

Returns: **Table View of Feature Layer**
in_table is returned with iterative selections applied

Return type: in_table

`PMT.json_to_featureclass`(***json_obj*, *out_file*, *new_id_field='ROW_ID'*, *exclude=None*, *sr=4326*, *overwrite=False***)

Creates a feature class or shape file from a json object.

Parameters:

- **json_obj** (*dict*) –
- **out_fc** (*str*) –
- **new_id_field** (*str*) – name of new field to be added
- **exclude** (*List; [String,...]*) – list of columns to exclude

- **sr** (*spatial reference*) – A spatial reference specification. Authority/factory code, WKT, WKID, ESRI name, path to .prj file, etc.
- **overwrite** (*bool*) – True/False whether to overwrite an existing dataset

Returns: Path

Return type: out_fc

See also

gdfToFeatureClass jsonToTable

`PMT.json_to_table`(***json_obj*, *out_file***)

Creates an ArcGIS table from a json object. Assumes potentially a geoJSON object.

Parameters:
- **json_obj** (*dict*) – dict
- **out_file** (*str*) – Path to output table

Returns: Path

Return type: out_file (str)

SeeAlso:

jsonToFeatureClass

`PMT.make_inmem_path`(***file_name=None***)

Generates an in_memory path usable by arcpy that is unique to avoid any overlapping names. If a file_name is provided, the in_memory file will be given that name with an underscore appended to the beginning.

Returns: String; in_memory path

Raises: **ValueError, if file_name has been used already** –

`PMT.make_path`(***in_folder*, **subnames***)

Dynamically set a path (e.g., for iteratively referencing year-specific geodatabases).

{in_folder}/{subname_1}/../{subname_n}

Parameters:
- **in_folder** (*str*) – String or Path
- **subnames** (*list/tuple*) – A list of arguments to join in making the full path

Returns (str):

str: String path

`PMT.multipolygon_to_polygon_arc`(***file_path***)

Convert multipolygon geometries in a single row into multiple rows of simple polygon geometries, returns original file if not multipart

Parameters:	**file_path** (*str*) – Path to input Poly/MultiPoly feature class
Returns:	path to in_memory Polygon feature class
Return type:	str

`PMT.polygons_to_points`(***in_fc, out_fc, fields='*', skip_nulls=False, null_value=0***)

Convenience function to dump polygon features to centroids and save as a new feature class.

Parameters:

- **in_fc** (*str*) – Path to input feature class
- **out_fc** (*str*) – Path to output point fc
- **fields** (*str or list, default="*"*) – [String,...] fields to include in conversion
- **skip_nulls** (*bool, default=False*) – Control whether records using nulls are skipped.
- **null_value** – Replaces null values from the input with a new value.

`PMT.table_difference`(***this_table, base_table, idx_cols, fields='*', **kwargs***)

Helper function to calculate the difference between this_table and base_table

ie… this_table minus base_table

Parameters:

- **this_table** (*str*) – path to a snapshot table
- **base_table** (*str*) – path to a previous year's snapshot table
- **idx_cols** (*list*) – column names used to generate a common index
- **fields** (*list*) – if provided, a list of fields to calculate the difference on; Default: "*" indicates all fields
- ****kwargs** – keyword arguments for featureclass_to_df

Returns:	df like this_table but containing difference values
Return type:	pandas.Dataframe

`PMT.table_to_df`(***in_tbl, keep_fields='*', skip_nulls=False, null_val=0***)

Converts a table to a pandas dataframe

Parameters:

- **in_tbl** (*str*) – Path to input table
- **keep_fields** (*list, default="*"*) – list of fields to keep ("*" = keep all fields)
- **skip_nulls** (*bool*) – Control whether records using nulls are skipped.
- **null_val** (*int*) – Replaces null values from the input with a new value.

Returns: pandas dataframe of the table

Return type: df (pd.DataFrame)

***`PMT.validate_directory`*(directory)**

Helper function to check if a directory exists and create if not

***`PMT.validate_feature_dataset`*(fds_path, sr, overwrite=False)**

Validate that a feature dataset exists and is the correct sr,
otherwise create it and return the path

Parameters:
- **fds_path** (*str*) – String; path to existing or desired feature dataset
- **sr** (*spatial reference*) – arcpy.SpatialReference object

Returns: String; path to existing or newly created feature dataset

Return type: fds_path (str)

***`PMT.validate_geodatabase`*(gdb_path, overwrite=False)**

Helper function to check if a geodatabase exists, and create if not

***`PMT.which_missing`*(table, field_list)**

Returns a list of the fields that are missing from a given table

Parameters:
- **table** (*str*) – Path to a table
- **field_list** (*list*) – List of fields

Returns: list

Appendix C: Maintenance Schedule

Overview

This document provides a proposed plan for maintaining the TOC-Tool website and supporting data.

The Transit Oriented Communities Tool (TOC Tool) is a website devoted to transit oriented communities (TOC) in Miami-Dade County, specifically in corridors and station areas defined by the Strategic Miami Area Rapid Transit Plan (SMART Plan). The TOC Tool website offers:

- introductory information to help site visitors understand what TOC is, learn about design conventions in station area walksheds, and become acquainted with key topics for measuring and monitoring TOC implementation progress;
- interactive dashboards with web maps, data summaries, and graphic visualizations to track TOC implementation progress over time;
- literature pertaining to TOC planning in Miami-Dade County, links to relevant related planning efforts in the region, and a review of academic and industry literature pertaining to TOC and similar development paradigms.

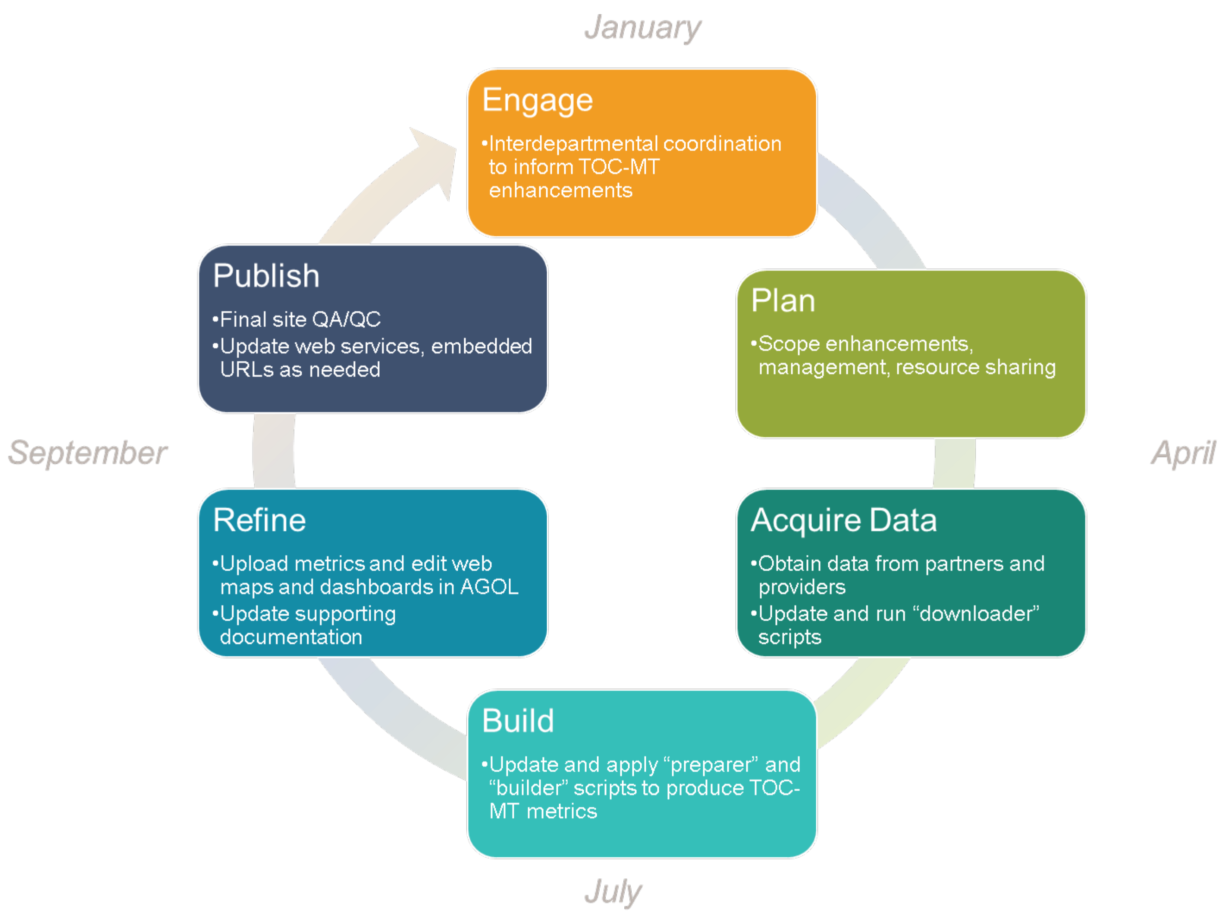
The TOC-Tool was developed by the Miami-Dade Transportation Planning Organization (TPO) but will be hosted and maintained by Miami Dade County, with support of the County's IT and GIS staff. To keep the content of the TOC Tool fresh and relevant over time, this document provides a maintenance plan identifying key steps, timelines, and departmental responsibilities (coordinated among TPO and GIS staff).

TOC Tool Components

The TOC Tool is a value-added data, analysis, and reporting resource that consumes data from diverse sources, executes scripted geoprocessing tasks, and facilitates updates to web-based services (features, tiles, maps, dashboards, and website) in the Miami-Dade County ArcGIS Online portal. Managing the TOC Tool over time requires routine maintenance to two principal components: site content and site data.

Site content includes the text and images that communicate key information pertaining to TOC concepts, TOC-related planning efforts, TOC Tool use (website help page and User Guide documents), and documentation of the calculation of TOC Tool metrics. Site data includes data acquisition, processing, and publication, which in turn involves configuring and executing the scripts used to develop TOC Tool content and maintaining the tool's Data Inventory and Technical Guide documents. The remaining sections of this document provide an outline of the maintenance tasks associated with site content and site data, identifying the lead agency responsible for each task, how frequently updates are likely to occur, the level of effort associated with each task, and expected coordination needs.

This maintenance plan envisions updating the TOC Tool on an annual basis, with final publication of version updates completed near the end of each calendar year. The key phases and timing of the update cycle are illustrated in the graphic below.



Primary Responsibility: TPO

Component	Lead agency	Update frequency	Time required	Coordination needs
Site text	TPO	As needed	Minimal	MD County Communications
Images	TPO	As needed	Minimal	MD County Communications
TOC-related plans ("Additional Reading" page)	TPO	As plans are updated	Low	Lead planning agency
Literature ("Additional Reading" page	TPO	As needed	Minimal	None
User Guide	TPO	As needed	Variable	Any contributors to site contents/ enhancements
Metric methodology documentation	TPO	As needed	Variable	Any contributors to site contents/ enhancements for content updates; GIS staff to run autodoc scripts.

Site Data Maintenance

Primary Responsibility: GIS

Component	Lead agency	Update frequency	Time required	Coordination needs
Source data (automated downloads)	GIS	Annually	Hours	TPO for "Download" script support as needed.
Source data (non-automated)	TPO	Annually	Days to weeks	Various departments as specified in Data Inventory document
Integrate novel data	TPO	Annually	Weeks	Any contributors to site contents/ enhancements
Local processing data	GIS	Annually	Days to weeks	TPO for "Prepare" and "Build" script support as needed.
Feature service updates	GIS	Annually	1-2 Days	None
Tile service updates	GIS	Annually	1-2 Days	None
Web maps	GIS	Annually	Minimal*	*Should be automatic unless there are methodology or visualization changes prompted by TPO or partners.
Dashboards	GIS	Annually	Minimal*	*Should be automatic unless there are methodology or visualization changes prompted by TPO or partners.
Experience builder website	GIS	Annually	2 weeks	*Should be automatic unless embedded url's change; recommend QA/QC review with TPO
Scripts/toolboxes	TPO	As needed.	Variable	Any contributors to site contents/ enhancements
Data inventory	TPO	As needed.	Variable	Contributors of new (undocumented) data
Technical Guide	GIS	As needed.	Minimal	TPO for "autodoc" scripts support.

Recommended Process for Updating Web Maps and Dashboards

When existing TOC Tool content is updated with new data, or when stylistic changes to maps or dashboards are envisioned, the updates should take place in copies of the maps and dashboards currently embedded in the TOC Tool Experience Builder website. This allows the existing content to be retained and displayed as-is until the updates have been reviewed and approved.

- Copy or create new web map.
    - The 'Template Map' has been provided to serve as a starting point for new maps and new metrics, with styling and zoom visibility preset to match all other maps.
    - Changes to existing maps are best handled by opening in ArcPRO, make updates, and Save Map up to AGOL, leaving the default to save a backup copy with a timestamp.
- Copy or create new dashboard.
    - The "Template Dashboard" has been provided to serve as a starting point for new dashboards and can be copied to build new metric dashboards.

    - Updating maps requires the use of AGO-Assistant (https://ago-assistant.esri.com/), and replacing the map widget ID with the ID of the map to be ported in.
    - NOTE: changing the map used in the dashboard will break the embedded widgets, therefore the recommended procedure is to use the template as a guide for style and basic formatting only.
- Build updated dashboard reports, following conventions defined in Technical Guide.
- When updated content is reviewed and ready for publication:
    - Ensure all updated content is shared publicly.
    - Embed updated dashboard's URL in Experience Builder site on the appropriate page.
    - Publish changes to Experience Builder site.
- Update offline TOC Tool documents
    - Update Metric Inventory spreadsheet with metrics and methodologies in new material.
    - Update Data Inventory spreadsheet with data sources used in new material.
    - Apply autodoc scripts to generate an updated methodology document.

Recommended Process for Adding New Topics to TOC-Tool

As the TOC Tool matures, new topics, elements, metrics, reporting time frames, etc. may be added. These additions will often require new or updated web maps and dashboards (see procedures above). Additionally, they will often require new pages in the Experience Builder site.

- Clone pages in Experience Builder to create space for new content
    - Topic pages provide an overview of major topics in the TOC Tool. If a new topic is added, clone a topic page.
    - Element pages include embedded dashboards in a View Navigator widget. If a new element is added, clone an element page. If the element uses different time frames than those in the cloned page, update the View Navigator widget.
- Keep new Experience Builder pages hidden until ready for publication. Toggle page visibility in the "Pages" menu of the Experience Builder application.
- Update upstream pages as needed
    - If a new topic is added:
        - Update the TOC Tool Home page to show the new topic among other topics in the ribbon of images under the site title.
        - Update the TOC Basics page to identify and describe the new topic.
    - If a new element is added:
        - Update the topic page to which the element belongs to show the new element and its relationship to TOC effectiveness.
    - Update the Metric Index table
- Update offline TOC Tool documents
    - Update Metric Inventory spreadsheet with metrics and methodologies in new material.
    - Update Data Inventory spreadsheet with data sources used in new material.
    - Apply autodoc scripts to generate an updated methodology document.

Methodological Additions or Revisions

As the TOC Tool matures, the scripts supporting its development will evolve with it. A version control system and code repository to manage versions, test code updates on side branches, raise and resolve issues collaboratively, etc. should be instituted. Git is an example of a widely used version control system. Sites like Github or Bitbucket provide web-based code repositories. The version control system will be administered by Miami-Dade County's IT Department. Decisions regarding what tools will be used, which users will be added with what privileges, conventions for version numbering, etc. will be coordinated with IT staff.

Attachment:

TOC Tool Peer Review Memo

1 Introduction

The Transit Oriented Development Performance Monitoring Toolkit (the PMT) is a web-based data visualization tool to track progress towards local and regional planning and development goals along existing and planned rapid transit corridors in the Miami-Dade region. The PMT aims to provide insight into how current and potential transit station areas and corridors change over time through public and private investments in housing, commercial buildings, recreational space, transportation infrastructure and more. The PMT focuses on transit-oriented development (TOD) areas, which combine the region's growth management emphasis on urban centers and nodal community development with areawide investment in rapid transit and multimodal infrastructure. Trends in growth and development, multimodal travel, urban design, accessibility, and more are tracked at a variety of geographic scales. The PMT is a collaborative product led by the Miami-Dade Transportation Planning Organization (the TPO) in conjunction with local jurisdictions and partner agencies.

This document begins with an overview of TOD, including what it is and how the community-oriented development patterns of TOD can offer diverse benefits. It then reviews several key planning documents that outline growth management and transportation policy priorities in the Miami-Dade region. These documents offer insight into potential key measures to include in the PMT. Finally, this document closes with a review of TOD monitoring programs in peer regions across the country, highlighting key topics in measuring TOD's effectiveness over time.

2 What is TOD?

Transit-oriented development (TOD) refers to neighborhoods and business districts designed around rapid transit stations. TOD is a type of community-oriented development (COD) that couples economic development and urban design principles with regional mobility via rapid transit. COD is characterized by compact, mixed-use development and a well-connected network of streets that provide local connectivity for pedestrians and bicyclists. These elements combine to create livable, economically thriving communities that offer opportunities for non-motorized travel, network redundancy for automobile travel, and nodal development patterns that can be easily connected using transit.

A growing number of regions around the country are planning for C/TOD because these development approaches can provide numerous benefits, including:

- Increasing access to jobs and essential services;
- Limiting the vehicle trips and/or vehicle miles of travel generated by new jobs and residents, which can also help improve air quality;
- Enhancing the productivity of rapid transit investments;
- Augmenting the role of walking and biking for recreational and utilitarian travel, which promotes healthy communities;
- Bolstering land values and retaining local property taxes.

COD establishes a development paradigm that offers these expected benefits, paves the way for rapid transit investments, and accounts for the inherent uncertainty associated with large-scale infrastructure projects. Whereas a TOD strategy may hinge on the complex factors informing whether and when a rapid transit project is built (grant funding applications, engineering and design timelines, project phasing, cost escalations, etc.), COD affirms sound planning and urban design principles to guide growth and development in a sustainable manner that supports community goals. These principles ensure that the loss or delay of a large prestige infrastructure project does not result in a total loss for the local community.

Moreover, a COD development paradigm creates nodes of development that can be effectively and efficiently served by transit - starting with fixed route bus services and potentially including future rapid transit technologies. These nodes of development foster land use and trip-making patterns that are likely to yield favorable ridership forecasts when planning for and designing rapid transit projects. Thus, COD can lay the groundwork for developing cost-effective rapid transit projects, and the advent of rapid transit planning and implementation allow a COD neighborhood to further evolve into a TOD neighborhood.

A short list of additional reading addressing COD as a development paradigm is provided at the end of this document.

The PMT focuses on existing and potential TOD areas in Miami-Dade's SMART Plan rapid transit corridors. For this reason, the remainder of this document focuses primarily on TOD specifically, rather than COD more generally. However, both COD and TOD are important components of the growth management strategy for the Miami-Dade region.

2.1 Elements of TOD

Planning for TOD often focuses on the **density** and **mix** of development near transit stations. Zoning code revisions and/or overlays in TOD areas typically allow for or promote mixed uses among multiple buildings within a neighborhood (horizontal mixed use) or within individual buildings, such as a multi-family residential building with retail on the ground floor (vertical mixed use). Moreover, parcels in TOD areas can usually be developed to higher densities than is typical in non-TOD areas. This combination of density and diversity places numerous destinations - jobs, stores, banks, restaurants, etc. - and housing units in a compact area that promotes walking and biking within the station area as well as the use of transit for travel to and from other parts of the region. Strategic planning for TOD emphasizes both regional mobility and local accessibility to offer diverse travel choices while supporting regional economic development goals and local livability goals.

Design is also important to TOD, ensuring that households and destinations are well-connected to each other and to transit services, not simply near to each other. Figure 14 offers an illustrative example. Development in the upper right is connected by a tight street network with uses easily accessible to the transit station. This is well-designed TOD. The lower left of the figure is characterized by large lots, with circuitous connections over a sparse network of streets and a large parking lot separating jobs and residents from the transit station. This type of development may be "transit adjacent" but is not well-designed TOD.

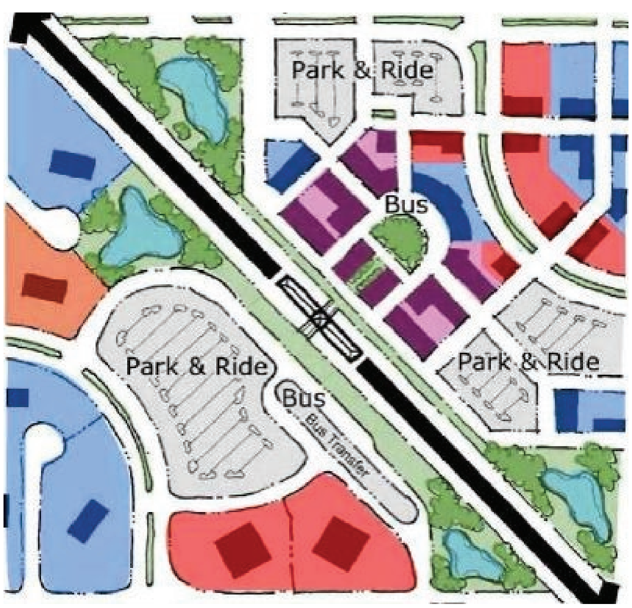


Figure 14: Appropriate (upper right) vs inappropriate (lower left) station area design for TOD (source: NTI)

In a well-designed TOD area, the strong connectivity among land uses and the transit station allows a relatively high proportion of trips to be made by walking or biking locally and by transit for longer-distance travel. In areas with "transit adjacent" development, these travel options are limited due to poor non-auto connectivity.

2.2 Defining the Station Area

In the planning and development industries, TOD is frequently evaluated over an entire "transit station area" rather than on a site-by-site basis. The station area is usually defined as the land area within a specified distance of a given transit station ($\frac{1}{4}$ mile or $\frac{1}{2}$ mile). More complex transit station-area definitions may account for walking and/or bicycling times to the transit station, barriers to accessing the station, or other factors that affect the ease with which the transit station can be reached from nearby developments.

For the PMT, it is important to establish a consistent station area definition to ensure TOD metrics are reported at similar scales and reliably represent the areas in which TOD plans are or could be implemented.

3 What makes TOD successful?

Like all land development processes, TOD is implemented gradually over time. Each station area has a distinct existing context, future vision, market readiness, and relationship to the regional rapid transit network. Processes for monitoring development trends and community characteristics in station areas can help localities assess the effectiveness of their TOD implementation strategies, make local policy adjustments, and coordinate with regional partners and/or neighboring jurisdictions to manage growth and support local and regional planning priorities over time. Whether TOD efforts are progressing in the right direction and whether they are successful will depend on how success has been defined locally and regionally. For this reason, it is important to identify a clear set of goals and objectives for TOD along with supporting metrics to be reported and monitored over time.

Some **common goals and objectives for TOD** are listed by key topic below. This list reflects general priorities expressed in local and regional planning documents and topics monitored by peer communities in their ongoing TOD reporting.

- **Coordinate Transportation and Land Use Planning**: TOD is an integrated transportation and land use strategy that directs growth and development to areas with well-connected, high capacity, multimodal transportation facilities to ensure regional mobility and accessibility. Common transportation and land use objectives for TOD include:
  - Maximizing return on transportation infrastructure investments by enhancing system utilization and productivity;
  - Providing multimodal travel choices to a significant proportion of regional residents, businesses, and visitors;
  - Investing in transportation facilities that support local and regional growth and development goals.
- **Economic Development**: TOD can often support local and regional economic development strategies by better connecting the workforce to jobs, strengthening connections among complementary industries, and constructing new or refitted building stock tailored to attract investment from firms in industries targeted by economic development policies. Common economic development objectives for TOD include:
  - Capturing a significant proportion of new jobs in station areas;
  - Increasing the value of real estate within station areas;
  - Enhancing workforce access to jobs regionally and within station areas;
  - Enhancing consumer access to goods and services regionally and within station areas.
- **Fiscal impact** In conjunction with economic development, investments in station areas often result in appreciating property values. For local jurisdictions, this can bolster tax revenues while more compact development patterns can minimize the additional costs of providing services to new residents and jobs. Examples of fiscal impact objectives for TOD include:

- Increasing the taxable value of properties in station areas;
- Increasing tax revenues from property, taxable sales, and other activities within station areas;
- Lowering per capita and/or per job expenditures within station areas;
- Utilizing taxes generated within station areas for public investments within the station area.

- **Access**: Growth and development in station areas should provide benefits to everyone, reducing overall cost burdens associated with combined housing and transportation expenses and improving access to jobs, education, recreation, and essential goods and services. Examples of objectives related to increased access include:
    - Reducing estimated combined costs of housing and transportation for all populations regionally and within station areas;
    - Improving access to jobs, education, recreation, and essential goods and services for all population segments.
- **Healthy Living**: The compact development patterns and connectivity that are essential to TOD can offer physical and mental health benefits by create opportunities for a variety of short trips that can be made by walking or bicycling and increasing social connections and interactions. Objectives related to healthy living may include:
    - Increasing the number or share of trips made by walking or bicycling within station areas;
    - Improving access to open space and recreational facilities;
    - Enhancing access to healthy food.
- **Environmental Stewardship**: TOD can help mitigate the impacts of human settlement and activity on the natural environment by creating development patterns that improve air and water quality, protect natural habitats, and preserve agricultural lands. Potential objectives for TOD related to environmental stewardship include:
    - Limiting growth in transportation-related emissions to protect regional air quality;
    - Limiting the expansion of impervious surfaces;
    - Lowering non-point source pollutants entering water sources;
    - Preserving natural areas and agricultural lands in the region.

4 What is the role of TOD in the Miami-Dade Region?

COD and TOD are important components of the Miami-Dade region's growth management strategy. Several key planning documents emphasize these development paradigms to accommodate continued regional growth supported by adequate urban infrastructure, including multimodal transportation facilities. Key planning documents include:

- **Miami-Dade County's Comprehensive Development Master Plan (CDMP)** [1]
- **The Miami-Dade TPO's 2045 Long Range Transportation Plan (LRTP)** [2]
- **The Strategic Miami Area Rapid Transit (SMART) Plan** [3]

The CDMP and LRTP are long-term planning documents that set goals and establish strategies for accommodating future growth in the region and guiding investments in transportation system expansions and improvements. The SMART Plan carries forward priorities expressed in the CDMP, LRTP, and other plans, focusing on six rapid transit corridors. The TPO is leading a multi-agency coordination effort to produce engineering and environmental studies for each corridor. These studies are important components of state and federal funding processes for major transportation investments.

Each of these key planning efforts is described briefly below. While these documents provide insight into regional planning priorities and growth and investment strategies, it should be noted that the Miami-Dade region consists of numerous local jurisdictions. Each of these local jurisdictions has its own growth policies and development regulations. It is important to understand these local priorities alongside those expressed in the plans discussed below. Successful COD and TOD implementation balances regional planning visions with local preferences and policies.

4.1 The Comprehensive Development Master Plan

The planning principles set forth by the CDMP, and supported by regulations in the County code, present a clear vision of development that is focused in **urban centers** and defined by accessibility provided by a **variety of transportation modes** (walk, bike, transit) in addition to vehicular mobility. Designations such as the Urban Development Boundary and Urban Centers serve to concentrate future development in multimodal areas and to encourage infill development and redevelopment in areas that are already urbanized. Figure 15 indicates the locations of Urban Center development areas and the Urban Development Boundary.

[1] https://www.miamidade.gov/planning/cdmp.asp
[2] https://miamidade2045lrtp.com/
[3] http://www.miamidadetpo.org/smartplan.asp

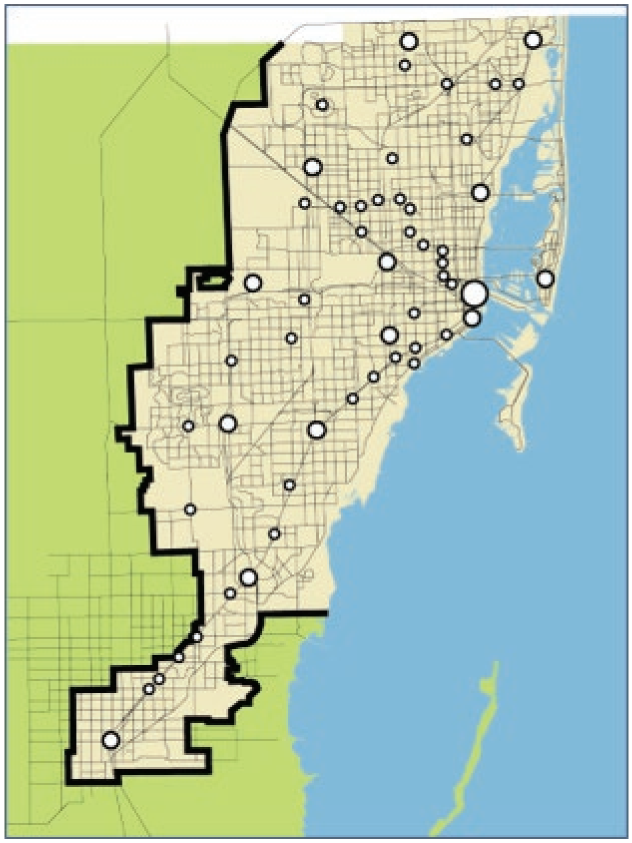


Figure 15: The CDMP aims to manage regional growth by focusing development in Urban Centers

The development vision for all designated Urban Centers reflects COD characteristics - dense, diverse development with strong neighborhood design and multimodal access to destinations. Many Urban Centers are also located along the existing MetroRail lines or in SMART Plan corridors, marking them as existing or potential TOD nodes. Additional planning the in the SMART Plan corridors may generate additional TOD areas along each corridor.

The CDMP sets general density targets for SMART Plan corridors and station areas, stratified by station area typologies. Table 6 shows the general specifications for a three-tiered station-typology, including station-area size, residential density maximums and floor area ratio minimums. Figure 16 then offers a simple illustration of targeted densities in station areas and along rapid transit corridors. These general guidelines apply to areas within $\frac{1}{2}$ mile of existing MetroRail and SMART Plan corridors, and within 1 mile of the East-West Corridor. However, density targets and related policies do not apply within incorporated areas - where local jurisdictions exercise planning and development approval authority - and are subject to zoning approval.

Table 6: Urban Center types and density thresholds

	Regional Station	Metropolitan Station	Community Station
Size	1-mile radius from station	1/4 to 1/2-mile radius from station/stop	700 to 1800-foot radius from stop
Residential Density (maximums)	500 Dwellings/Acre	250 Dwellings/Acre	125 Dwellings/Acre
Floor Area Ratio (minimums)	4.0 in core 2.0 in edge	3.0 in core 0.75 in edge	1.5 in core 0.5 in edge

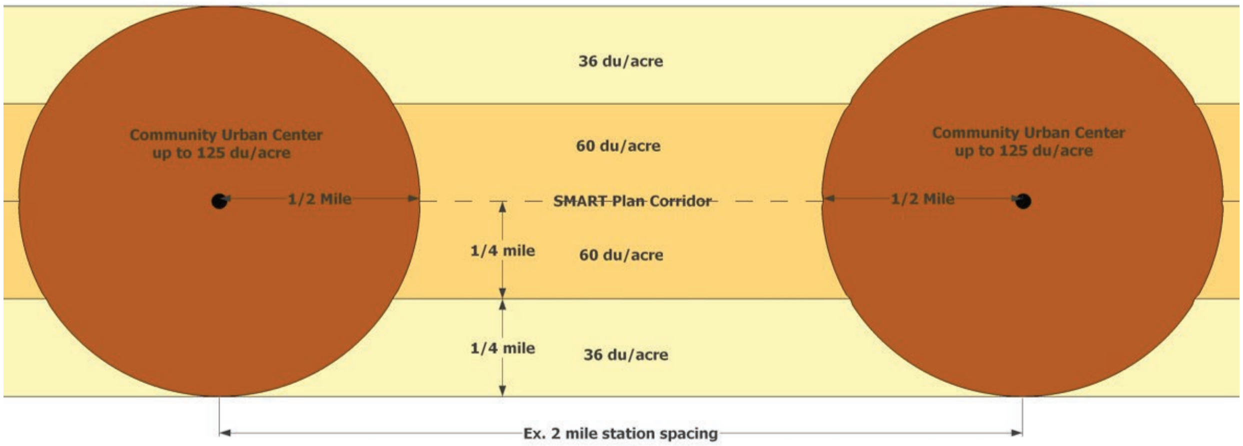


Figure 16: The CDMP establishes station-area and corridor density targets for transit-supportive development.

Overall, the development patterns envisioned in the CDMP are intended to support the goals of **decreas-ing impact to the environment** (both by minimizing encroachment on nearby agricultural land and by reducing carbon emissions by creating a multimodal network), **developing a robust transit system** that makes the region internationally competitive, maximizing efficiency by **integrating land use and transportation planning**, and **optimizing efficiency in public service delivery**. The elements of the plan presented above may provide useful benchmarks for TOD reporting and monitoring at the station area and corridor scales and for a variety of station area types.

4.2 The 2045 Long Range Transportation Plan

The goals established for the future of the transportation network in the Miami-Dade TPO’s 2045 LRTP align with the multimodal vision for urban development set forth in the CDMP and other local plans. While the LRTP is a comprehensive regional transportation plan that establishes needed projects and funding priorities for all modes, it places a heavy emphasis on transit and the implementation of the SMART Plan. According to the LRTP, “The highest priorities of the TPO Governing Board are transit and projects that support transit services.” Given this emphasis, the LRTP aims to develop the regional transit system as an essential strategy to promote economic growth and job creation and increase the region’s international competitiveness.

Examples of goals and objectives listed in the LRTP that relate to TOD include:

- **GOAL: Maximize Mobility Choices Systemwide**
  - Objective: Provide a comprehensive transportation network for dependable and reliable transportation options.
  - Objective: Increase mobility choices throughout the county.
- **GOAL: Support Economic Vitality**
  - Objective: Provide affordable housing.
  - Objective: Improve access to employment centers.
  - Objective: Provide access to tourist destinations - seaports, airport, beaches, etc.
- **GOAL: Protect and Preserve the Environment and Quality of Life and Promote Energy Conservation**
  - Objective: Preserve agricultural land or critical habitat consumed by transportation projects.
  - Objective: Minimize and mitigate air and water quality impacts of transportation facilities, services, and operations.
  - Objective: Promote projects that support urban infill and densification.
  - Objective: Provide affordable transit service from identified Communities of Concern to major activity centers (i.e., healthcare, recreation, education, employment, and cultural facilities).
  - Objective: Improve Quality of Life for all ages and abilities.

These excerpts from the 2045 LRTP resonate with many of the key elements of successful TOD, reinforcing the importance of TOD in the region's coordinated growth management and transportation planning strategy. They also clarify topics that warrant monitoring as the SMART Plan is implemented and TOD matures over time to understand how TOD is helping the region achieve these and similar goals and objectives.

4.3 The SMART Plan

The SMART Plan identifies six rapid transit corridors that extend the region's existing rapid transit network. Figure 17 shows a map of each corridor's extents and relationship to the existing rapid transit network.

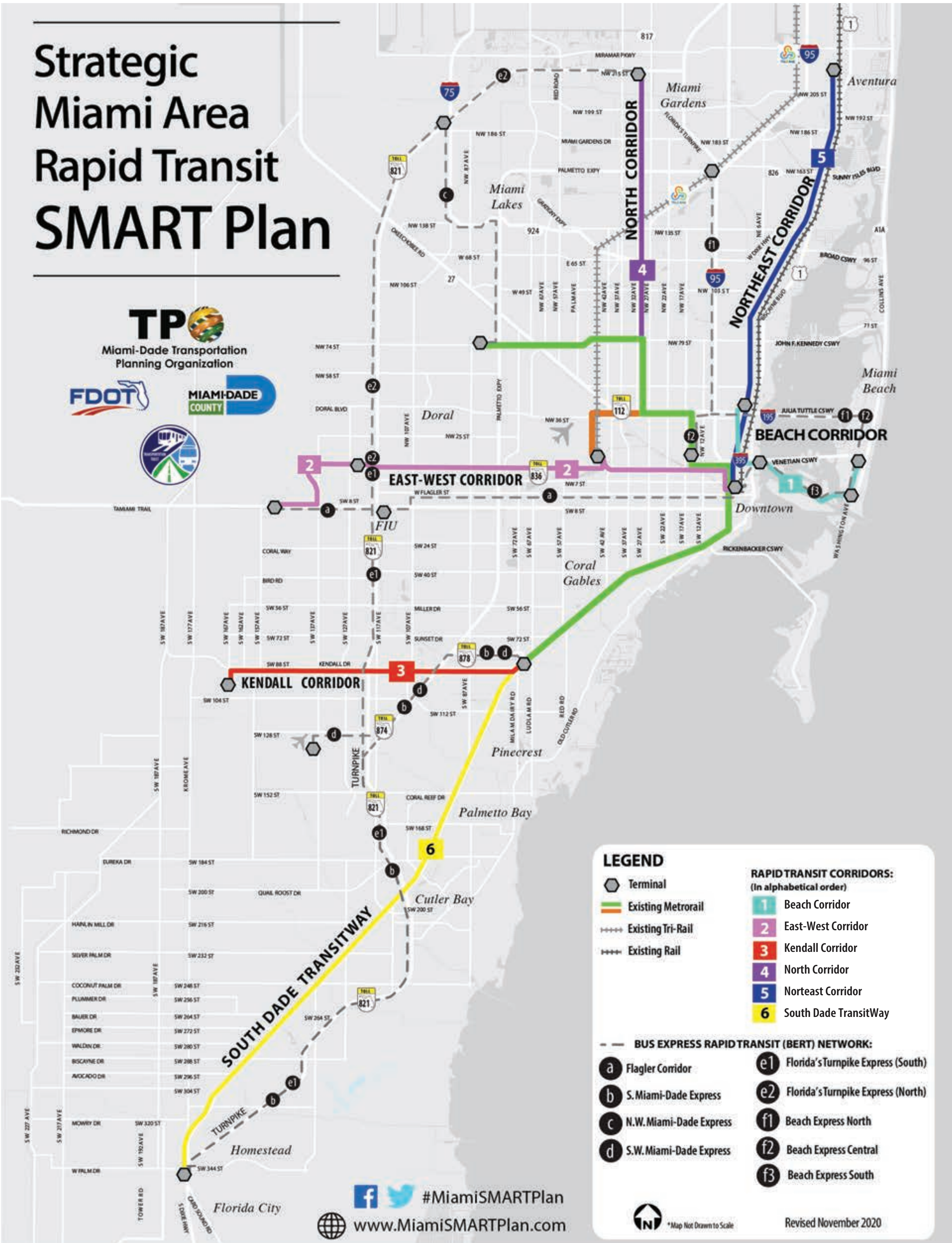


Figure 17: SMART Plan: Map of six planned rapid transit corridors

All six SMART Plan corridors and additional Bus Express Rapid Transit (BERT) corridors are located within the Urban Development Boundary and many of the zoned Urban Centers[4] are located along SMART Plan corridors. The areas targeted by the County for concentrated growth are planned to be along premium transit, exemplifying the integration of land use and transportation planning that is essential to TOD.

Current SMART Plan efforts are focused on studying transit technologies, operations, costs, ridership, environmental impacts, etc. for each of the six corridors. This includes visioning studies and Project Development and Environmental (PD&E) studies that address key questions of land use planning, transportation system engineering, environmental impacts, community visions for growth and development and more. Table 7 provides a snapshot of current planning progress for each corridor. The findings of these corridor-specific studies will help define detailed station areas, set benchmarks for station area and corridor-level performance monitoring, and provide essential data to develop the PMT.

Table 7: SMART Plan corridor planning progress

Corridor	**PD&E Status**	**Land Use and Vision Planning Study**
Beach	In Progress	Anticipated 2020
East-West	In Progress	In Progress
Kendall	In Progress	In Progress
North	In Progress	Complete
Northeast	In Progress	Anticipated 2020
South Dade Transitway	Complete – awarding construction (Fall 2022)	Complete

[4] SMART Plan station locations that are not already zoned as urban centers will be zoned by 2025 pursuant to CDMP Policy LU-9F.

5 How are other regions measuring TOD progress and success?

Across the country, regions with populations, transit system development, and/or growth dynamics comparable to those in the Miami-Dade region have been planning for TOD and tracking progress across a diverse collection of metrics. Specific metrics, methods, and media vary based on each peer region's planning priorities, goals for TOD, and monitoring objectives. This section summarizes TOD performance monitoring efforts in five peer regions:

- Denver
- Dallas
- Minneapolis
- Seattle
- Salt Lake City

For each region's monitoring effort, reports, tools, maps, and related documents are described with the aim of understanding best practices, identifying key metrics, and recognizing strengths and weaknesses in methodology, presentation, interpretability, etc. The following questions merit consideration when reviewing TOD modeling efforts in each peer region:

- What topics/measures are reported?
- What scale are they reported at?
- What time frames are they reported for?
- How could this resource inform the development of the PMT?

5.1 Denver RTD

Resources reviewed:

- Denver RTD TOD Development Database [5]
- Denver RTD 2013 Transit Oriented Development Status Report Tool [6]

The Regional Transportation District (RTD) finalized its TOD Strategic Plan in 2010, outlining the main goals of the region's TOD strategy, which include:

- More compact and dense development in station areas than outside of station areas.
- Mixed uses.
- Walkable urban design and streetscapes.

Since the development of the initial TOD Strategic Plan, RTD has released two reports (one in 2013 and one in 2018) cataloging new development in station areas. The two reports draw on data from two sources: the **RTD Development Database**, which is maintained by RTD staff, and a **TOD Trend Tool**, which uses property data from CoStar Group, a commercial property industry tracking firm. The RTD database defines TOD more narrowly than the TOD Trend Tool. To be included in the RTD database a development must meet the following criteria:

- It is situated within a half-mile walk of a transit station.
- It adheres to TOD principles.
- It is a large scale project.
- It is located outside of the following locations: downtown Boulder, downtown Denver (except Union Station), Auraria, Anscutz (no office TOD), and Denver Tech Center.

The TOD Trend Tool, on the other hand, includes all developments within a half-mile radius of a station, regardless of design or scale.

These two data sources allow RTD to catalog the **growth in number of multifamily, office, and retail developments near station areas**, tracking trends over time and at multiple geographic scales (station area, route, and region).

In the TOD Trend Tool, information is summarized at the corridor level and station area. At the corridor level, data are summarized to show the number of residential units, hotel rooms, and non-residential floor area by use type (retail, office, government, medical, etc...) completed by year. Planned developments are also inventoried. At the station level, specific development projects are discussed in addition to information about structured parking. The report also estimates total real estate development value in dollars.

[5] https://www.rtd-denver.com/projects/transit-oriented-development-projects
[6] https://www.rtd-denver.com/sites/default/files/files/2017-06/rtd-tod-status-report-2013_0.pdf

The RTD Development Database serves as an inventory of projects. Each site includes information about the project, including: use, income designation, rental vs. owned, number of units, non-residential square footage by type, year built or planned. The database can be viewed as a table or as web-based interactive map.

The metrics presented in by the Denver region's TOD monitoring efforts provide examples for tracking and reporting details of land development in TOD areas, along rapid transit corridors, and across the region. The Denver region does not use performance benchmarks to monitor progress toward specific TOD goals in these reports but focuses on a robust inventory of development approvals that meet TOD criteria.

Several excerpts from the TOD Trend Tool and the RTD Development Database are presented in the figures below.

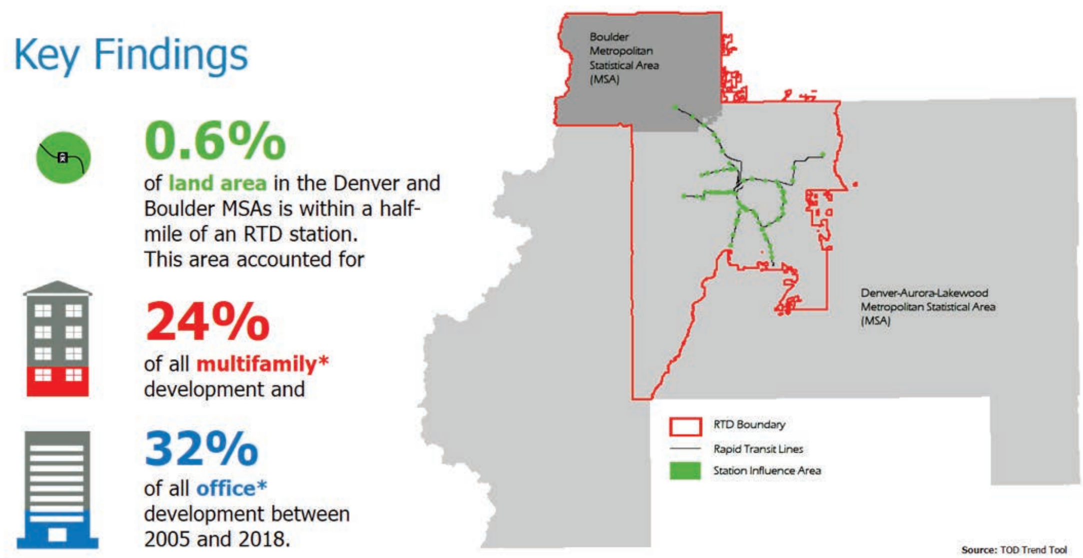


Figure 18: This graphic from the Transit Oriented Development Status Report identifies overall key findings across the region.

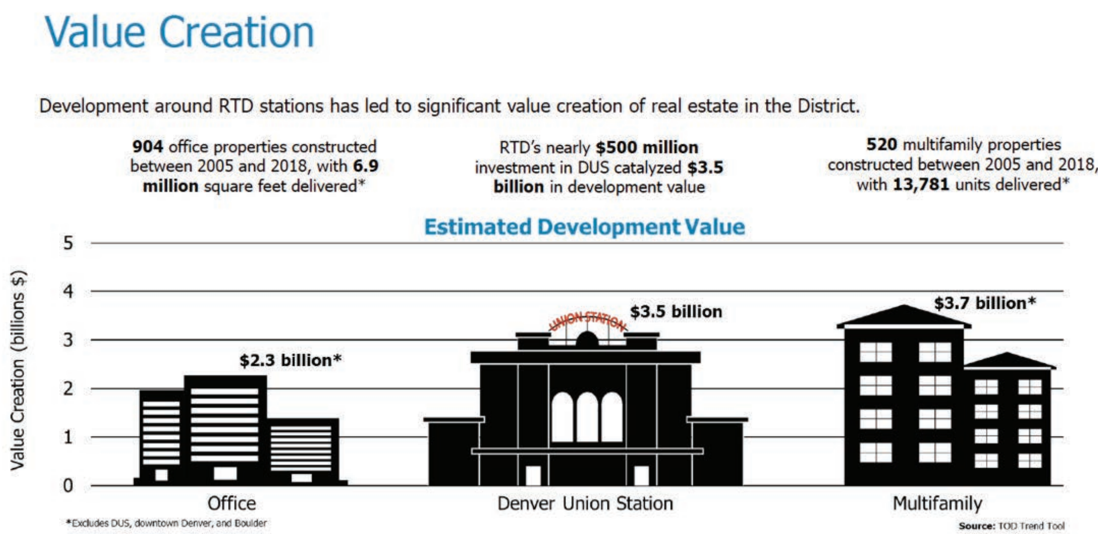


Figure 19: Value creation by development type is illustrated across the region in this graphic from the Transit Oriented Development Status Report.

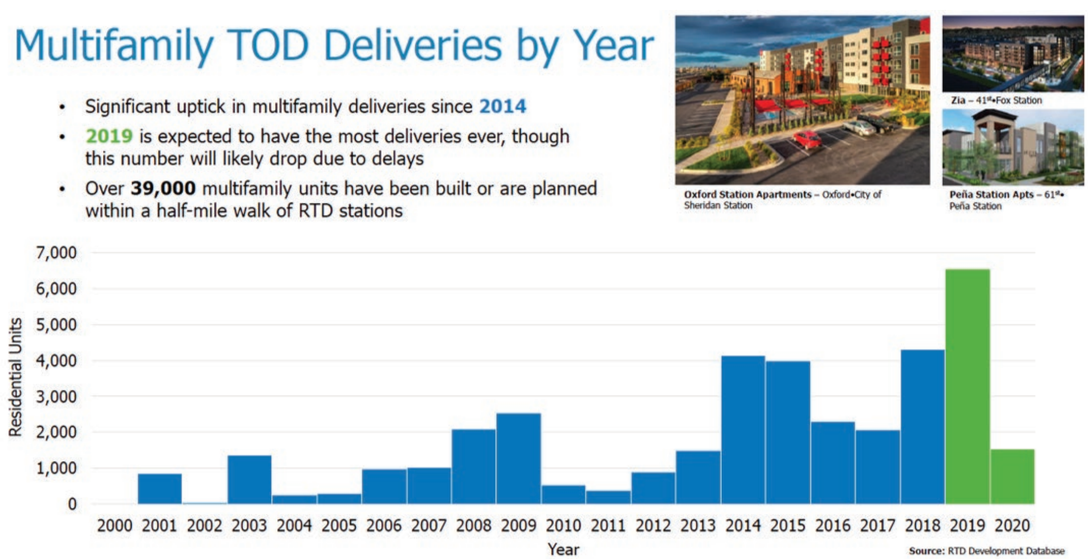


Figure 20: This chart from the Transit Oriented Development Status Report shows the number of residential TOD units by year, including planned project, for the RTD region.

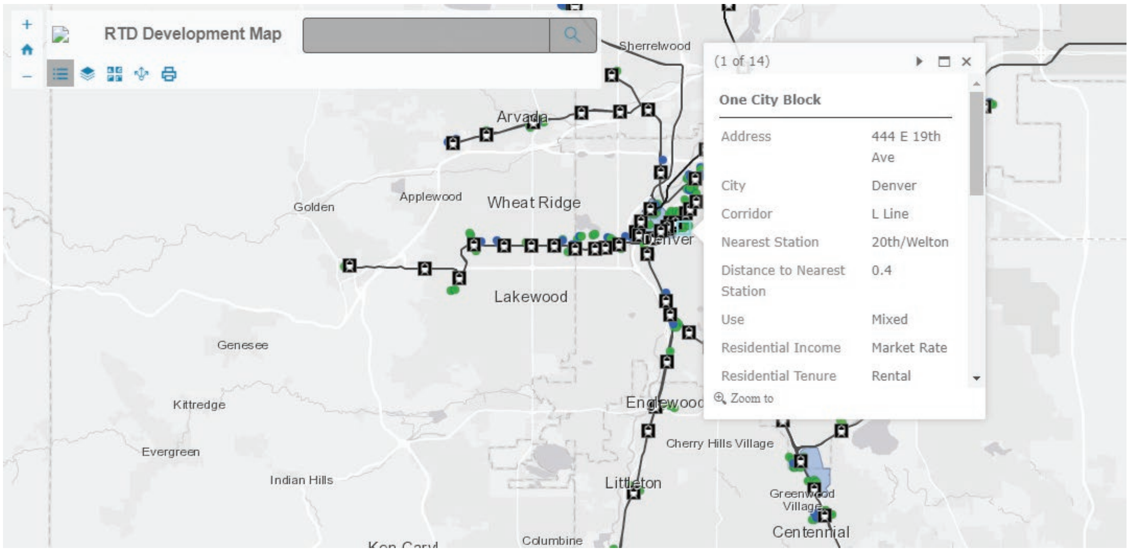


Figure 21: The RTD Development Database can be viewed as a table or online web-based map.

Top 10 Stations for TOD*

	Office	Total TOD (sq ft)	Number of Properties
1	Union Station	1,836,191	8
2	Arapahoe at Village Center	1,417,000	5
3	Sky Ridge	700,000	1
4	38th/Blake	525,000	4
5	Dry Creek	506,587	4
6	Boulder Junction	440,000	3
7	Belleview	390,000	2
8	Central Park	220,000	1
9	Colorado	220,000	1
10	Lincoln	197,000	1

	Residential	Total TOD (units)	Number of Properties
1	20th/Welton	3,253	16
2	Union Station	2,316	7
3	Dry Creek	1,977	8
4	Belleview	1,924	7
5	Lincoln	1,805	7
6	US 36 & Broomfield	1,717	6
7	25th/Welton	1,406	8
8	Sky Ridge	1,142	5
9	I-25/Broadway	1,072	5
10	Boulder Junction	848	6

- The Southeast Corridor and Downtown Denver have seen the greatest development

* Excludes Downtown Denver and Downtown Boulder Stations

Source: RTD Development Database

Figure 22: These tables from the Transit Oriented Development Status Report use data from the RTD Development Database to rank stations by office and residential TOD development.

5.2 Dallas DART

Resources reviewed:

- Dallas – The Economic and Fiscal Impacts of Development Near DART Stations (May 2017) [7]

Dallas Area Rapid Transit (DART) has worked with the University of North Texas since 1999 to track the **fiscal impact of development near light rail stations** in the Dallas-Forth Worth area. The first two studies (completed in 1999 and 2002) looked at **property value change over time** for properties within a quarter mile of light rail stations and compared these to the property value change over the same time period for properties in comparable areas.

In 2005, the research method switched to focusing on newspaper articles and community announcements about developments within a quarter mile of stations and summarizing the total amount of new real estate investment. This same method was used in the 2007 study in addition to the use of field work to identify projects not found in local publications. In 2009, researchers totaled the economic impact of capital projects to build out additional rail lines, including expenditures, labor income, employment, and indirect business taxes. A 2013 report again evaluated the value over time and consequent tax revenue generation of properties within a quarter mile of stations against those of control properties in similar areas, this time from 1996-2013. Finally, a 2017 report summarizes development by two main time-frames: 1999-2013 and 2014-2015.

The report catalogs projects that are either "completed or under construction" or "planned or proposed." All types of projects are included in the analysis and categorized as either non-residential, multi-family, single-family, and health care, indicating that a strict definition of TOD is not applied. Projects are named individually in table form. Figures representing direct impact (dollars), total impact (dollars), labor income (dollars), employment (jobs) and state and local taxes (dollars) are presented for each project type, aggregating all projects within station areas in the DART network.

Each of these studies was heavily focused on totaling development values and the fiscal impact generated by those values, in terms of the economic impact in employment and tax revenue generation at the local and state level. This demonstrates an emphasis in the region on using **TOD for economic development and revenue generation**. The methods used to estimate development values and the holistic economic and fiscal impact of these developments can provide a framework for reporting similar measures in the PMT. Figure 23 provides examples of system-level reporting on the economic and fiscal impact of TOD in the Dallas region.

[7] https://www.dart.org/about/economicdevelopment/May2017DARTEconomicandFiscalImpacts.pdf

Table 5: Potential Fiscal Impacts of Existing & Proposed TOD	
Description	***Value***
Announced Value	**$ 4,902,800,000**
Announced Value Attributable to DART	**$ 4,255,700,000**
Cities (Annually)	
Taxable Property Value	$ 2,843,779,000
Property Tax Revenues	$ 16,785,000
Taxable Retail Sales	$ 665,552,000
Sales Tax Revenues	$ 6,656,000
Total Revenue to Cities	**$ 23,531,000**
Total State and Local Tax Revenues	**$ 127,095,000**

Table 6: Economic & Fiscal Impacts from DART System Buildout (All Lines)	
Description	***Impact***
Total Expenditures	$2,337,000,000
Economic Activity	$ 4,059,672,000
Labor Income	$ 1,520,157,000
Employment	32,095
Other Property Income	$ 432,140,000
Indirect Business Taxes	$ 99,986,000

Figure 23: The Economic and Fiscal Impacts of Development Near DART Stations (2013 and 2017)

5.3 Minneapolis-St. Paul Metro Transit

Resources reviewed:

- Minneapolis MetroTransit – Development Along Transit: Regional Growth Near High Frequency Transit in the Twin Cities (2019) [8]

Metro Transit in Minnesota's Twin Cities region focuses its TOD reporting on tracking **trends in permitted and planned developments along high frequency transit corridors**. This effort is documented in *Development Trends Along Transit: Regional Growth Near High Frequency Transit in the Twin Cities*, which draws from data in an annual regional Building Permit Survey for permitted projects and an inventory of planned projects built by tracking development announcements in news media publications.

Development projects are categorized as multifamily residential, commercial, industrial, or public and institutional. Mixed Use is also included as a category for planned projects. Development trends are tracked over two different time frames, depending on the type of project. For multifamily residential projects, development trends are tracked from 2009 through 2018. Other development projects are tracked from the years 2003 to 2018.

Metrics are reported at three different scales: region-wide, areas with high frequency transit services, and at the route level. In addition to graphs and tables, the report includes maps plotting development projects from 2003-2018 by the permit value in dollars.

The report limits the tracking of development projects along transit to locations with high frequency transit service. This is defined in the report as service that qualifies as either Light Rail Transit (LRT), Bus Rapid Transit (BRT), or high frequency local bus. To qualify as high frequency service, route headways must be 15 minutes or shorter on weekdays and Saturday during daytime hours. A half-mile proximity is used to define areas proximate to high frequency service.

The Twin Cities example is similar to others that focus primarily on development trends, but it adds a criterion for TOD based on transit service frequency regardless of technology or supporting infrastructure. This criterion highlights the need to specify the transportation attributes that qualify a location to be considered a TOD area. For the PMT, this may suggest the need to develop transportation metrics and/or targets as part of the TOD monitoring framework.

Three figures below offer excerpts from the Twin Cities TOD monitoring effort. Figure 24 shows regional reporting of the value of permitted developments in TOD areas and the share of regional development (by number of units and by value) captured within TOD areas. Figure 25 provides an example chart that highlights route-level development impacts over time. Figure 26 plots the location and values of all commercial developments in the region, highlighting those in TOD locations.

[8] https://www.metrotransit.org/Data/Sites/1/media/tod/devtrendsreport_2019.pdf

Year	Units	Permit Value	% of Region Units	% of Region Permit Value
2009	544	$62,421,676	25.1%	27.7%
2010	950	$93,362,624	29.0%	28.0%
2011	1,398	$123,580,901	34.5%	38.5%
2012	4,618	$511,893,249	59.0%	62.0%
2013	3,616	$608,217,713	45.9%	50.2%
2014	1,962	$277,538,961	30.6%	32.5%
2015	3,300	$566,979,633	42.0%	46.0%
2016	3,375	$587,405,883	37.3%	41.9%
2017	3,801	$573,663,158	37.5%	40.9%
2018	4,386	$863,685,057	37.7%	41.8%
Total	**27,950**	**$4,268,748,855**	**43.2%**	

Figure 24: Example table from Development Trends Along Transit report showing share of regional resi-dential development occurring within proximity to high frequency transit throughout the region by year.

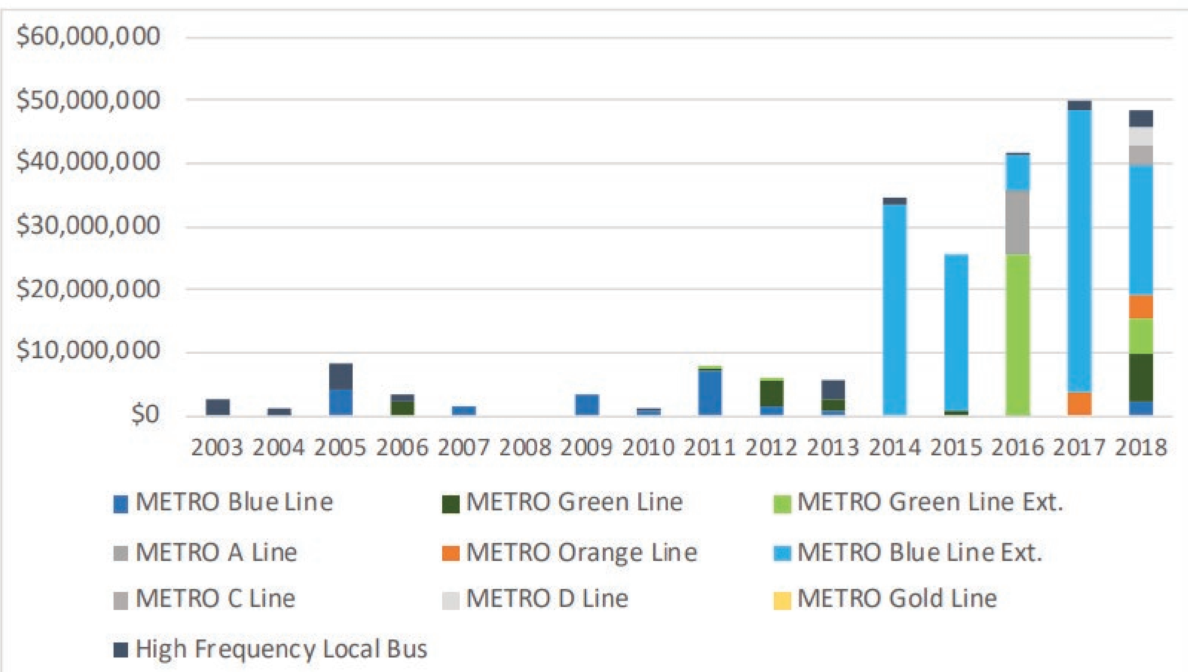


Figure 25: Example chart from Development Trends Along Transit report showing industrial permit value by high frequency corridor.

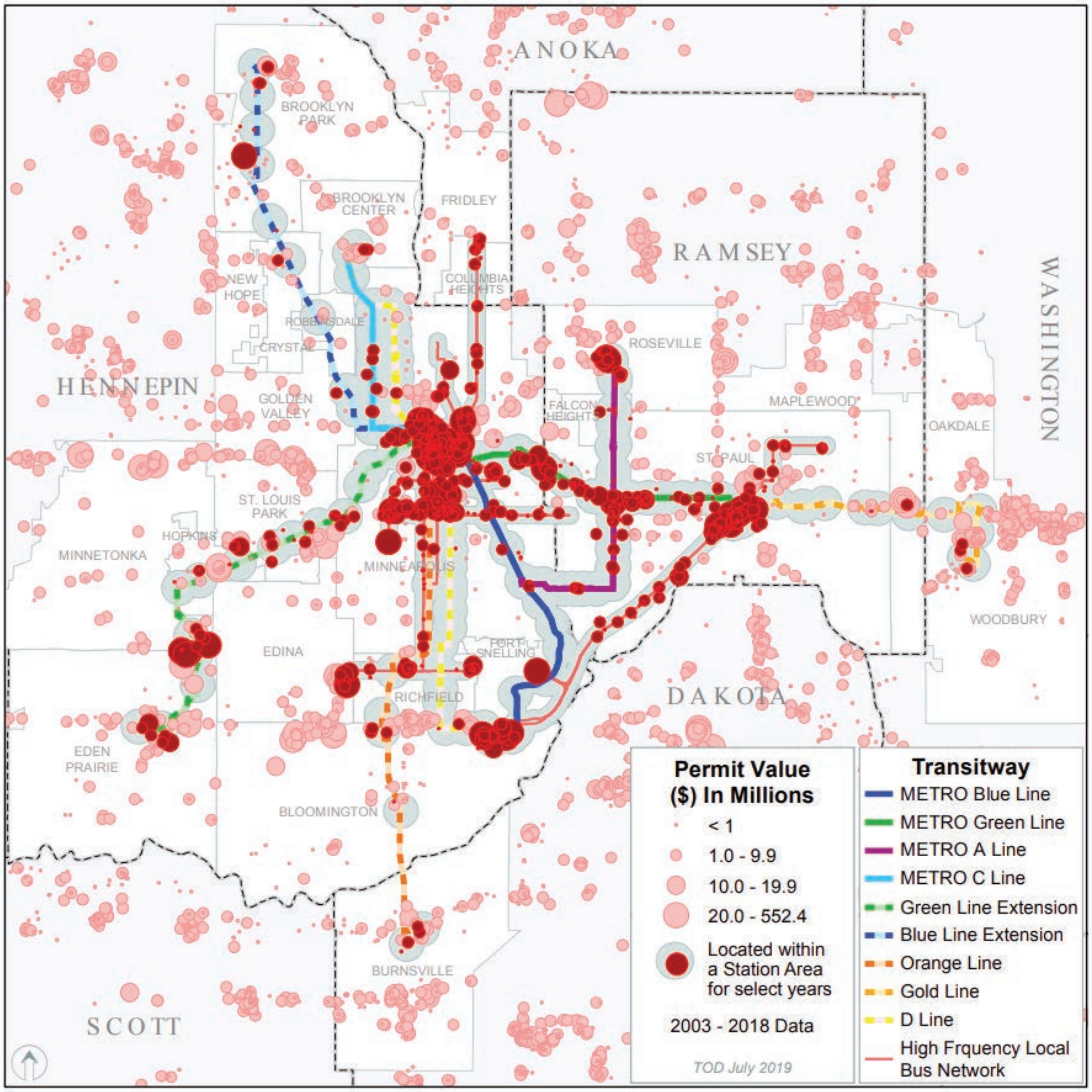


Figure 26: Example map from Development Trends Along Transit report illustrating commercial develop-ment near high frequency transit from 2013-2018.

5.4 Seattle PSRC

Resources reviewed:

- Puget Sound Growing Transit Communities Monitoring Report (Dec 2017) [9]
- The Growing Transit Communities Strategy (2013) [10]

The Puget Sound Regional Council (PSRC) has identified TOD as a key strategy for achieving the region's transportation, economic, and environmental goals. PSRC's Growing Transit Communities Strategy (2013) sets forth the following main goals for TOD in the region:

- Attract residential and employment growth to transit communities.
- Provide housing choices near high capacity transit that are affordable to a full range of incomes.
- Increase access to opportunity for residents of transit communities.

Within this framework, the 2017 Growing Transit Communities Monitoring Report presents key data on a comprehensive set of factors. The report reviews **trends in demographics, housing, employment, transit ridership, and TOD-related funding and policies** in an effort to comprehensively monitor the implementation of TOD across the region. The report is divided into into two main sections: performance monitoring and implementation monitoring.

The *performance monitoring* section explores measures of population, housing, affordable housing, employment, and ridership. Metrics are reported for each station area and are compared to regional totals or other station areas and tracked over time. The report does not use a standardized format for reporting and providing comparisons, but rather highlights notable trends and comparisons for each station area. Station areas are defined by the type of transit service available. A $\frac{1}{2}$-mile buffer is used for light rail, commuter rail, and ferries. BRT and streetcar stations are defined by a $\frac{1}{4}$ mile buffer. The analysis excludes downtown Seattle stations, noting that they are situated in a unique context relative to other station areas in the region.

The *implementation monitoring* section focuses on reviewing **policies and actions taken to facilitate the implementation of TOD** in the region. The report covers municipal zoning efforts, affordable housing projects and incentives, funding, long range transportation plans, the regional transportation plan, and parking tools. In some cases the report provides measures that attempt to quantify implementation progress, such as the portion of station areas with a small area plan that has been adopted or is in development. In other cases, policies are highlighted but not measured quantitatively. For example, the plan notes the proportion of jurisdictions in the region that have adopted affordable housing incentives, and highlights one specific city's incentives.

[9] https://www.psrc.org/sites/default/files/tod201712-pres-gtcmonitoringreport.pdf
[10] https://www.psrc.org/sites/default/files/gtcstrategy.pdf

TOD reporting in the Seattle region offers a template for monitoring a comprehensive array of topics related to TOD at the station area and regional scales. In addition to tracking development trends, it highlights transportation, access, and equity dimensions of TOD for performance monitoring. It also acknowledges the key role of development policies in different jurisdictions. These considerations are important in framing potential topics and metrics to incorporate in the PMT.

Excerpts from the *Growing Transit Communities Monitoring Report* are presented below. Figure 27 presents a table of TOD topic areas and metrics used to monitor station area performance and track progress across the region. Figure 15 provides an example of tracking growth trends at stations through-out the region and assessing the effectiveness of TOD in managing regional growth. Figure 28 shows an example of monitoring housing affordability in TOD areas. Figure 29 tracks regional ridership trends over time by transit technology.

Population/ Demographics	Percent of regional population growth that occurred in corridor areas
	Change in percent of population within corridor areas as percent of total population
	Variation in population among corridor areas
	Variation in population growth among corridor areas
	Variation in percentage of residents who identify as non-white among corridor areas
Housing and Affordability	Percentage of new regional housing permits issued in corridor areas
	Change of housing in corridor areas as percent of regional housing
	Percentage of housing units in corridor areas that are affordable to households making 80% AMI or less
	Variation in affordability by corridor area
	Cost burden by corridor area
Jobs	Percent of regional job growth that occurred in corridor areas
	Change of jobs within corridor areas as a percent of total jobs in the region
	Variation in job growth among corridor areas
Transit Ridership	Trends in transit ridership over time
	Boardings at stations with High Capacity Transit compared to those without
	Average weekday boardings by station
	Trends in transit use and walking/biking mode share over time
	Transit commute share by service

Figure 27: *Measures included in the Growing Transit Communities Monitoring Report (December 2017)*

Figure 28: Population change by station area is summarize in this graphic from the Growing Transit Communities Monitoring Report (December 2017)

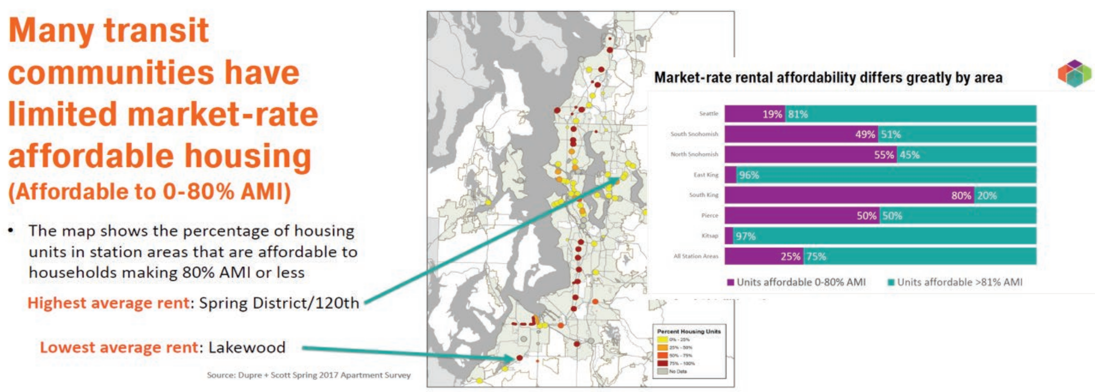


Figure 29: This graphic from the implementation monitoring portion of the Growing Transit Communities Monitoring Report (December 2017) provides information about the presence of market-rate affordable housing in the region.

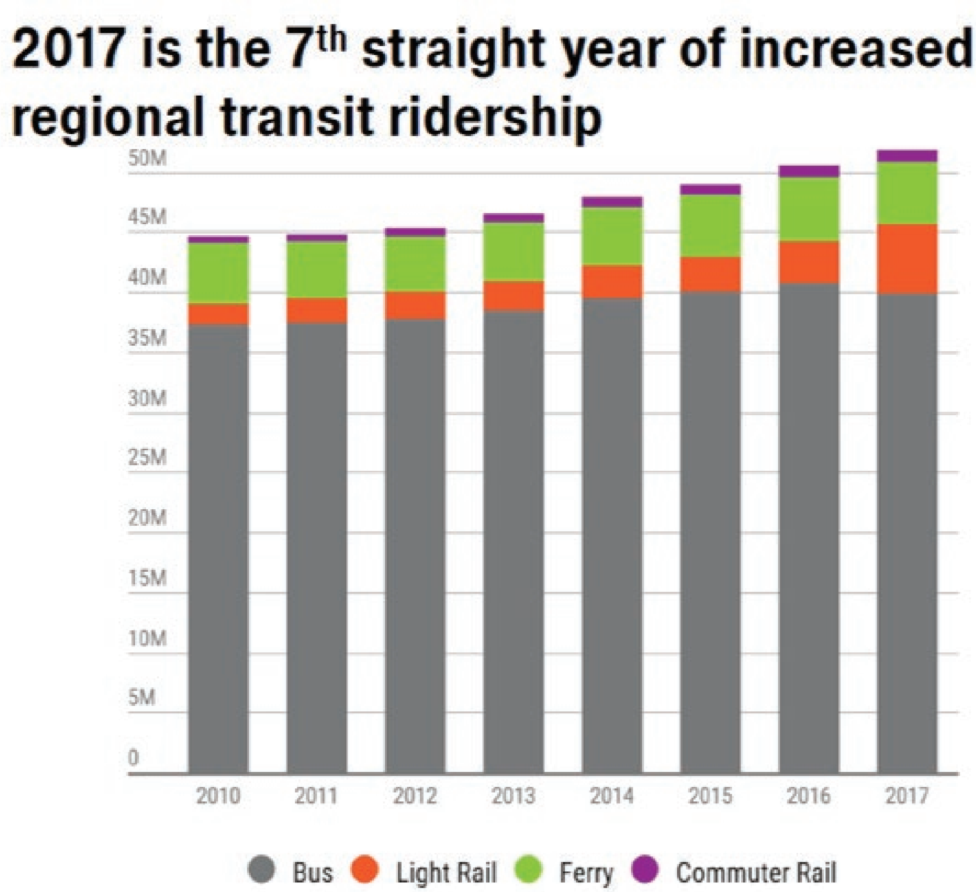


Figure 30: Ridership by mode across the transportation system is plotted in this chart from the Growing Transit Communities Monitoring Report (December 2017).

5.5 Salt Lake City UTA

Resources reviewed:

- Utah Transit Authority TOD Web Tool [11]

The Utah Transit Authority (UTA) sees TOD as an opportunity to **decrease traffic congestion**, **improve air quality and public health**, **lower cost of living**, **increase accessibility**, and **generate economic activity** by increasing land use intensity. UTA maintains an interactive online database and map called the *TOD Web Tool*, where users can access information about individual station areas by clicking on them in an interactive map. The *TOD Web Tool* gives each station area three different scores:

- **Overall TOD score:** assesses market strength, municipal support for TOD, and accessibility to provide an overall assessment for market TOD to be successful.
- **Growth Opportunity:** uses market strength, growth projections, and need for TOD as indicators for highlighting stations that are strong candidates for TOD.
- **Affordable Housing:** denotes station areas where affordable housing as part of TOD is most appropriate.

These three main scores and the metrics that go into calculating them are displayed on the map of stations in the *TOD Web Tool*. In addition to the map, there is a tab on the web tool that allows a user to **compare a station's score** with the maximum possible score and the top station score, by one or all corridors. A separate tab provides data on measures that describe **how a station can improve its score**. Measures gauging a station's potential to improve its score are provided across five different TOD-enabling factors. This allows the user to both understand the underlying components of the composite score, as well as to better understand what types of actions could be taken to increase the overall score.

- Zone for Mixed Use
- Zone for Higher Density
- Plan for Moderate Income Housing
- Adopt TOD Supportive Parking Requirements
- Adopt TIF Districts

The *TOD Web Tool* provides examples of comprehensive regional benchmarking, comparing each station's TOD progress relative to maximum potential and observed scores. It also highlights the importance of market factors in achieving transit-oriented patterns of development. The PMT may seek to incorporate a similar understanding of TOD readiness in monitoring TOD in the Miami-Dade region.

Several snapshots of the *TOD Web Tool* are provided below, highlighting how it presents station area scores and allows the user to drill down for additional details.

[11] https://www.rideuta.com/Doing-Business/Transit-Oriented-Development/TOD-Web-Tool

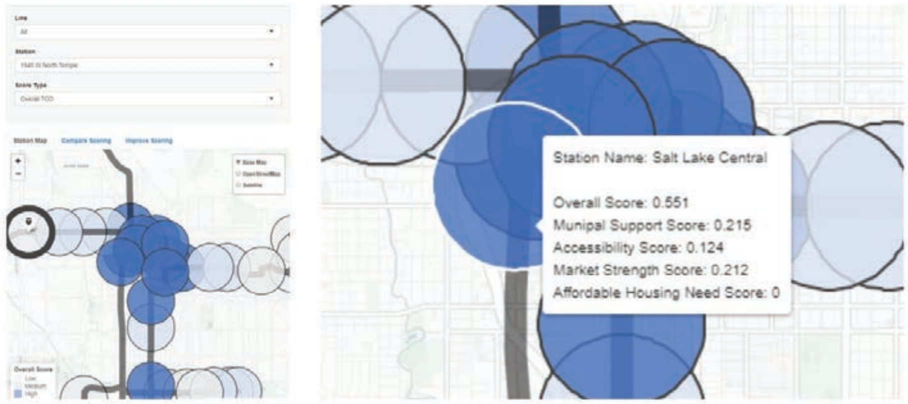


Figure 31: Example of UTA TOD Web Tool Web Interface showing individual station measures.

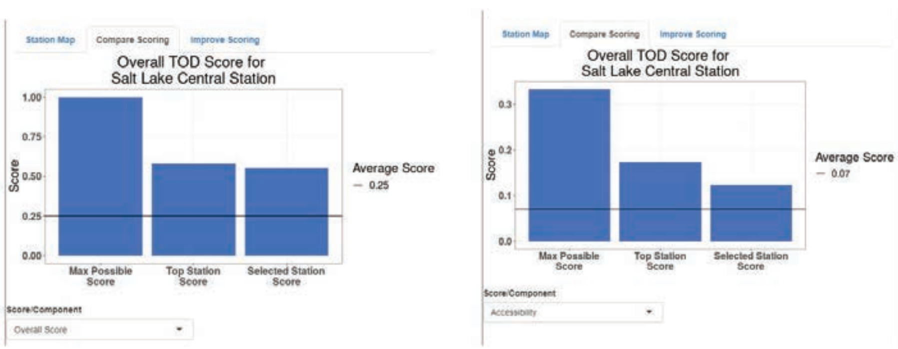


Figure 32: Example of UTA TOD Web Tool Web Interface ability to compare a station's score to other stations in the region or corridor.

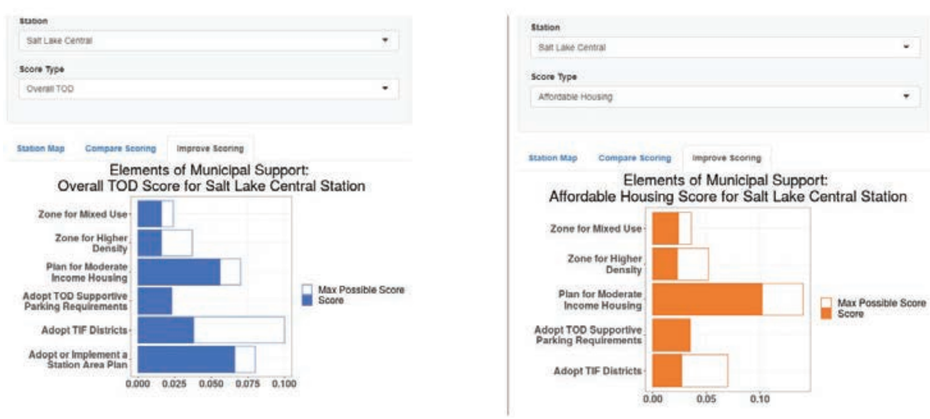


Figure 33: Example of UTA TOD Web Tool displaying individual scores relative to maximum possible scores for different measures, providing a measure of how much opportunity there is for improvement for each factor.

6 Additional Reading

Below are brief synopses of several articles focused on community-oriented development (COD) and transit-oriented development (TOD). While COD is not an established term in the academic or industry literature, these articles address the same planning principles described as COD in this document.

"Community-Oriented Development" Resources

Building Transit Oriented Development in Established Communities

Center for Urban Transportation Research, University of South Florida. October 2002[12]

Summary: Through reviewing case studies where auto-oriented communities have successfully re-imagined themselves as transit-oriented communities, researchers identify a series of approaches intended to guide the implementing of transit-oriented development in auto-oriented areas.

From the report: This report provides a synthesis of the **steps that established car-oriented communities have taken to transform into transit-oriented communities.** The report identifies several approaches, such as the use of transit oriented design, focusing transit-oriented development (TOD) **around park-and-ride lots**, making changes to land development regulations, **parking management**, offering development incentives, coordinating stakeholders, incorporating transit into future development/redevelopment, crafting TOD design guidelines, predesignating transit corridors, **ensuring pedestrian and bicycle access**, adapting transit services to the needs of suburban style communities, offering location efficient mortgages and **ideas for dealing with community resistance** toward applying transit friendly measures to car oriented communities.

Does TOD need the T? On the Importance of Factors Other Than Rail Access

Journal of the American Planning Association. May 9, 2013[13]

Summary: Seeking to identify relationships between auto ownership and use and household characteristics near rail station, researchers collect and analyze survey data in New Jersey. They find that rail access explains less than factors such as parking affability, tenure, density, and bus service. The authors conclude that "TOD" development can be effective even if crated beyond rail stations.

From the report: Transit-oriented developments (TODs) often consist of new housing near rail stations. Channeling urban growth into such developments is intended in part to reduce the climate change, pollution, and congestion caused by driving. But new housing might be expected to attract more affluent households that drive more, and rail access might have smaller effects on auto ownership and use than housing tenure and size, parking availability, and the neighborhood and subregional built environments.

I surveyed households in northern New Jersey living within two miles of 10 rail stations about their housing age and type, access to off-street parking, work and non-work travel patterns, demographics,

[12] https://www.nctr.usf.edu/pdf/473-135.pdf
[13] https://www.tandfonline.com/doi/abs/10.1080/01944363.2013.791008

and reasons for choosing their neighborhoods. The survey data were geocoded and joined to on-street parking data from a field survey, along with neighborhood and subregional built environment measures. I analyzed how these factors were correlated with automobile ownership and use as reported in the survey.

Auto ownership, commuting, and grocery trip frequency were substantially lower among households living in new housing near rail stations compared to those in new households farther away. But rail access does little to explain this fact. Housing type and tenure, local and subregional density, bus service, and particularly off- and on-street parking availability, play a much more important role.

Transportation and land use planners should broaden their efforts to develop dense, mixed-use, low-parking housing beyond rail station areas. This could be both more influential and less expensive than a development policy oriented around rail.

Cost of a Ride: The Effects of Densities on Fixed-Guideway Transit Ridership and Costs

Journal of the American Planning Association. June 27, 20[14]

Summary: Seeking to identify cost-effective transit investments and density thresholds for station areas, researchers explore relationships between transit ridership, transit investments, population and employment densities, and costs. After controlling for a variety of factors, researchers identify an inverse relationship between density and capital costs. They develop density thresholds to guide planners.

From the report: High costs and low ridership are the bane of fixed-guideway transit investments. The net capital and operating cost per passenger mile of recent investments ranged from $/0.22tooover/$10 in 2008. **A better understanding of characteristics of the most successful transit investments can help inform future investment policy and improve the performance of existing transit systems**. We evaluated the ridership, operating costs, and capital costs of recent transit investments and identified job and population densities that can support more cost-effective fixed-guideway transit service.

Controlling for neighborhood, regional, and transit service attributes, population and job density are positively correlated with both ridership and capital costs. As density increases, so do capital costs and ridership. Density, however, has an inverse relationship to capital cost per rider and total costs per passenger mile. Higher densities tend to improve transit's cost effectiveness, in spite of higher capital costs.

Job and population densities around transit stations are frequently below minimum thresholds needed for cost-effective transit investments and operation. This contributes to high costs per passenger mile on many transit systems. We generate **density guidelines for cities and towns to use as a point of comparison and a potential target for zoning** around existing and proposed transit stations based on actual or projected capital costs.

[14] https://www.tandfonline.com/doi/abs/10.1080/01944363.2011.589767

Do TODs Make a Difference?

National Institute for Transportation and Communities (NITC). December 2015[15]

Summary: The authors of this report seek to identify how TODs perform relative to the metropolitan area that they are situated within. To this end, researchers reviewed 23 fixed guideway transit systems in 17 metropolitan areas in the South and West. They find that TODs do perform different from the metropolitan area. Research findings across seven areas are reported below.

From the report: In this report, we present research that measures the outcomes of TOD areas in relation to their metropolitan area controls with respect to (1) jobs by sector, (2) housing choice for household types based on key demographic characteristics, (3) housing affordability based on transportation costs, and (4) job-worker balance as a measure of accessibility. Prior literature has not systematically evaluated TOD outcomes in these respects with respect to light rail transit (LRT), commuter rail transit (CRT), bus rapid transit (BRT), and streetcar transit (SCT) systems.

We find: (1) **most TOD areas gained jobs in the office, knowledge, education, health care and entertainment sectors**, adding more than $100 billion in wages capitalized over time; (2) in assessing economic resilience associated with LRT systems, jobs continued to shift away from TOD areas before the Great Recession, the pace slowed during the Recession, but reversed during recovery leading us to **speculate that LRT TOD areas may have transformed metropolitan economies served by LRT systems**; (3) rents for offices, retail stores and apartments were higher when closer to SCT systems, had mixed results with respect LRT systems, but were mostly lower with respect to CRT systems (our BRT sample was too small to evaluate); (4) **SCT systems performed best in terms of increasing their TOD area shares of metropolitan population, households and householders by age, housing units, and renters with BRT systems performing less well while LRT and CRT systems experienced a much smaller shift in the share of growth**; (5) **household transportation costs as a share of budgets increase with respect to distance from LRT transit stations to seven miles suggesting the proximity to LRT stations reduces total household transportation costs**; (6) emerging trends that may favor higher-wage jobs locating in transit TOD areas over time than lower or middle wage jobs perhaps because TOD areas attract more investment which requires more productive, higher-paid labor to justify the investment; and (7) the share of workers who commute 10 minutes or less to work increases nearly one-half of one percent for each half-mile their resident block group is to an LRT transit station, capping at a gain of 1.3 percent, which is not a trivial gain.

How Does Transportation Affordability Vary Among TODs, TADs, and Other Areas?

National Institute for Transportation and Communities (NITC). August 2017[16]

Summary: This report seeks to compare and contrast travel behaviors and affordability between transit-oriented developments (TODs), transit-adjacent developments (TAD) and hybrids. After classifying station areas based on several built-environment factors, the researchers found that TOD tends to increase transit and non-motorized forms of transportation and decrease VMT. The researchers also conclude

[15] https://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1006&context=trec_reports
[16] https://nitc.trec.pdx.edu/research/project/859

that TODs tend to result in transportation costs being a lower burden in households versus TADs.

From the report: Transit-oriented development (TOD) has gained popularity worldwide as a sustainable form of urbanism; it concentrates development near a transit station so as to reduce auto-dependency and increase ridership. Existing travel behavior studies in the context of TOD, however, are limited in terms of small sample size, inconsistent TOD classification methods, and failure to control for residential self-selection. Thus, this study has three research questions. First, how can we distinguish between Transit-oriented development (TOD) and Transit-adjacent development (TAD)? Second, how do travel behaviors vary between TODs and TADs? Third, how does transportation affordability vary between TODs and TADs? This study utilizes cluster analysis to classify station area types and propensity score matching to control residential self-selection.

From cluster analysis with built-environment factors—density, diversity, and walkability—in a half-mile buffer, this study classifies existing station areas as TOD, TAD or Hybrid types. After controlling for residential self-selection, **it shows that a TOD motivates its residents to walk more and take transit more while using personal vehicles less.** The significant difference between TOD and TAD in both VMT and the number of auto trips demonstrates that TODs make the personal vehicle trips shorter and fewer. **Travel behavior in the Hybrid type demonstrates the possibility of gradual and practical change.** Finally, the percentage of household income spent on transportation is lower in TOD households than TAD households. This shows that a TOD household is likely to save enough money on vehicle ownership and use that, while it likely spends more on transit, the final result is a **significantly lower financial burden from transportation**.

Incentivizing TOD: Case Studies of Regional Programs Throughout the United States

Puget Sound Regional Council Growing Transit Communities Partnership. October 8, 2012[17]

Summary: With a goal review diverse approaches to the implementation of TOD, this report is comprised of case studies from 1994 to 2006. Each case study review a variety of topic areas, including the sponsoring org nation, the TOD program that created it, activities around creating the TOD, funding, and lessons learned.

From the report: This report illustrates and assesses different approaches to supporting transit-oriented development, through six case studies of regional programs throughout the country. Through describing the structure, funding and implementation of these programs, the report provides a broad but detailed survey of strategies employed by regional organizations working in concert with local actors to support the balanced growth of communities near transit. Particular attention is paid to the ways in which the programs support equitable TOD.

Histories of Transit-Oriented Development: Perspectives on the Development of the TOD Concept

Berkeley Institute of Urban and Regional Development. Fall 2007.[18]

[17] https://todresources.org/app/uploads/sites/2/2016/06/incentivizing-tod.pdf
[18] https://escholarship.org/uc/item/7wm9t8r6

Summary: This paper reviews the concert of the "TOD" through time. It explores the topic through identify historical development patterns and identifying economic and urban theories behind them.

From the report: This paper looks at the development of the TOD vision from multiple perspectives. First, the paper looks at a long history of **transit served real estate development**. Second, it evaluates TOD's position in a long history of social and urban theory. Lastly, it tracks Calthorpe's TOD concept through his lifetime of work up to the publication of "The New American Metropolis." This paper is meant to serve as a foundation for further work that will seek to **define the term TOD** in its many uses across the fields of planning, design, transportation, and policy.

Seven American TODs: Good practices for urban design in Transit-Oriented Development projects

Journal of Transport and Land Use. November 11, 2011[19]

Summary: What constitutes good urban design in a TOD? This publication develops a series of case studies in order to identify a series of 12 guiding principals. The authors relate each principal to either TOD "process," "place," or "facilities." The study also identifies pitfalls to avoid.

From the report: In the past few decades, Transit-Oriented Development (TOD) has emerged as a popular and influential planning concept in the United States. Physical design is an important aspect of making TOD projects work as it is a crucial means of coordinating relatively intensive land uses and multiple transportation modes. This paper **analyzes seven American TOD projects in terms of urban design** and concludes with a discussion of "good practices" for future TOD projects focusing on development processes, place-making, and facilities. This research supplements prior scholarship on TOD that has tended to focus on policy issues such as regulation and financing.

Weaving Together Vibrant Communities through Transit-Oriented Development If You Build Rail Transit in Suburbs, Will Development Come?

Community Investments. Summer 2010[20]

Summary: How does rail transit effect residential development in suburban communities over a long period of time? This longitudinal study explores development at suburban rail stations in Portland over a 10-year period. The author identifies preconditions that may need to exist for redevelopment to occur. These include vacant land, presence of non-residential uses, and compatible zoning. Further, the author also notes that many studies likely do not explore a long enough period of time to capture land use changes.

From the report: There is little consensus about the effects of rail transit on residential development, especially in suburbs, despite high expectations. I revisit this issue by examining a "best case": I test the impacts of rail transit on housing development around suburban rail stations in Portland (OR), a region with multiple supportive transit-oriented development policies. conducted longitudinal analyses of housing development in quarter mile catchment areas around 57 suburban rail transit stations in sub-

[19] https://www.jtlu.org/index.php/jtlu/article/view/67
[20] http://www.reconnectingamerica.org/assets/Uploads/abrookstod082010.pdf

urban Portland from 2004 to 2014, a longer time period than many previous studies, evaluating factors that helped station areas attract residential development. Housing developments, most multifamily, grew much faster in the quarter mile catchment areas around suburban stations put into service in or before 2004 than the regional average. Areas around stations opened after 2004 have few residential developments. More residential development is associated with higher system ridership, more vacant land zoned for residential and mixed-use purposes, greater shares of nonresidential land, and higher shares of the White population. Faster density increases were associated with more vacant land zoned for high-density and mixed-use purposes.

Rail systems may need to be in operation longer to affect land use than the study periods in some previous research. Rail transit can guide suburban residential development in neighborhoods with enough vacant land, some nonresidential destinations, and appropriate zoning. **Multiple supportive public policies and incentives appear to have a major impact, but may not be effective alone unless these preconditions are met.**

Reasons for Living in a Transit-Oriented Development, and Associated Transit Use

Journal of the American Planning Association, November 26, 2007[21]

Summary: The authors of this study use survey data to identify why individuals chose to relocate to a TOD. While surveys were conducted for TODs in California, the author notes that the results are generalize.

From the report: Cities and regions throughout the U.S. are promoting transit-oriented development (TOD) near rail stations to increase both transit use and the number and range of housing opportunities. This paper reports the results of a survey of households who moved to TODs within the last 5 years, finding a wide range of motivations. **About one-third of respondents reported access to transit as one of the top three reasons for choosing to live in a TOD.** They were equally or more likely to cite lower housing cost or the quality of the neighborhood. Those who reported that their choice of residence location was motivated in part by access to transit were more likely to use transit than those who did not.

[21] https://www.tandfonline.com/doi/abs/10.1080/01944360608976757

Attachment:

TOC Tool Development Framework Memo

Introduction

The Transit Oriented Development Performance Monitoring Toolkit (the PMT) is a web-based data visualization tool to track progress towards local and regional planning and development goals along major rapid transit corridors in the Miami-Dade region. The PMT aims to provide insight into how transit station areas and corridors change over time through public and private investments in housing, commercial buildings, recreational space, transportation infrastructure and more. The PMT focuses on transit-oriented development (TOD) areas, which combine the region's growth management emphasis on urban centers and nodal community development with areawide investment in rapid transit and multimodal infrastructure. Trends in growth and development, multimodal travel, urban design, accessibility, and more are tracked at a variety of geographic scales. The PMT is a collaborative product led by the Miami-Dade Transportation Planning Organization (the TPO) in conjunction with local jurisdictions and partner agencies.

The PMT development effort is guided by the following goals, which aim to maximize its relevance and value as a data and analysis resource for policy planners and decision-makers.

- Focus on the goals and objectives of the Strategic Miami Area Rapid Transit (SMART) Plan that can be addressed through TOD
- Develop performance metrics that clearly indicate the region's performance relative to SMART Plan TOD goals across multiple topics
- Track trends over time by summarizing baseline performance and routinely updating data and metrics
    - Assess the effectiveness of TOD and related policies and investments
    - Understand the factors that drive development demand in TOD areas to support long-range forecasting
- Provide a simple interface for users
    - Summarize the story of TOD (what's working, what can be improved) at various scales
- Explore detailed metrics and data

This document presents the PMT analysis framework, which discusses the underlying logic and organization for the eventual toolkit design and implementation. The purpose of this framework is to detail the ways in which TOD best practices relate to and facilitate the attainment of several key goals expressed in the SMART Plan, as well as to summarize the methodological approach and data that will be utilized during the creation of the final toolkit.

Framework Objectives:

1. Connect the PMT to SMART Plan objectives
2. Define the elements of effective TOD
3. Describe temporal and spatial frames of analysis
4. Organize reporting for insight at multiple scales

When completed, the toolkit interface will provide a user-friendly dashboard to track progress towards SMART Plan goals at multiple geographic scales and across several different time frames. A well-designed PMT is expected to reinforce multi-agency and jurisdictional collaboration in SMART Plan implementation by providing data-driven insights that can inform organizational actions and resource prioritization over time.

In developing the framework, this document describes several key topics in TOD performance measurement and expresses high-level goal language distilled from SMART Plan documents and ongoing efforts. These are used to define the PMT's analytical needs, which are organized at three reporting levels:

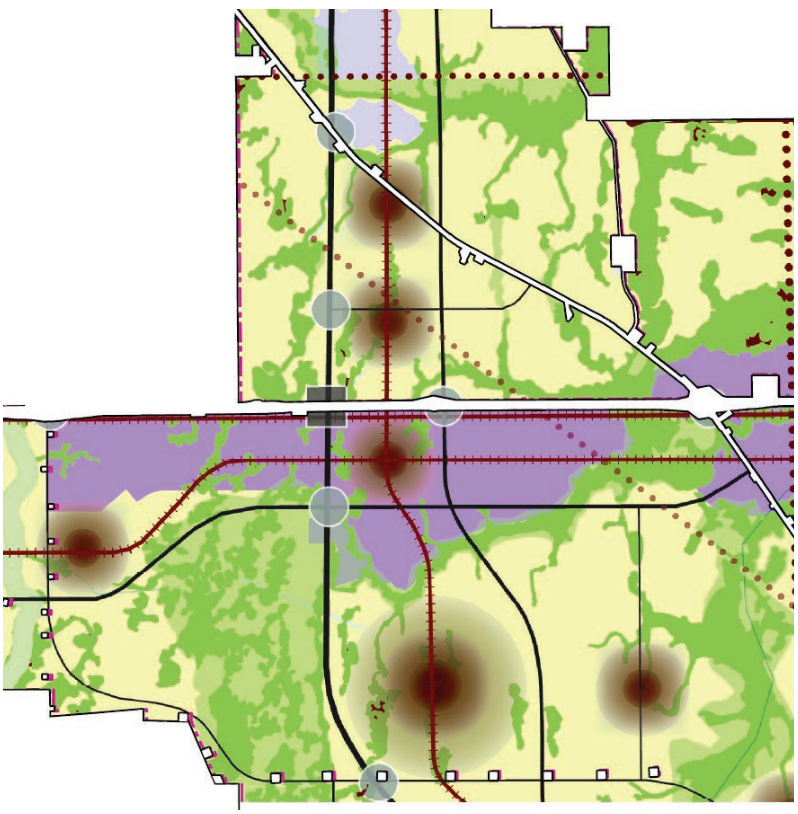
- **Topics** represent the most basic reporting level, summarizing detailed information to indicate the extent to which regional planning goals are being met.
- **Elements** of TOD effectiveness provide an intermediate level of reporting in which insights about the impacts of TOD can be understood within topic areas.
- **Performance metrics** offer the most fine-grained level of reporting, focusing on specific impacts of TOD that comprise various elements.

Consideration of each reporting level helps establish the time horizons and geographic summary levels to be reported in the PMT interface as well as the specific analysis and data needs for building the toolkit.

Framing TOD Performance

Key Topics in TOD Performance

TOD refers to neighborhoods and business districts designed around rapid transit stations. TOD is a type of community-oriented development (COD) that couples economic development and urban design principles with regional mobility via rapid transit. COD is characterized by compact, mixed-use development and a well-connected network of streets that provide local connectivity for pedestrians and bicyclists. These elements combine to create livable, economically thriving communities that offer opportunities for non-motorized travel, network redundancy for automobile travel, and nodal development patterns that can be easily connected using transit, as shown in Figure 34.



Community-oriented development (COD) focuses on nodal development with a dense center of diverse land uses transitioning to lower density edges characterized by residential buildings. This development pattern promotes the organization of distinct districts that can be easily connected by transit. When rapid transit is added to a COD area, it becomes a transit-oriented development (TOD). Although not every COD area will become a TOD, each supports the region's growth management strategy and lays a foundation for developing cost-effective rapid transit projects.

Figure 34: COD Development Patterns - Linear Nodes

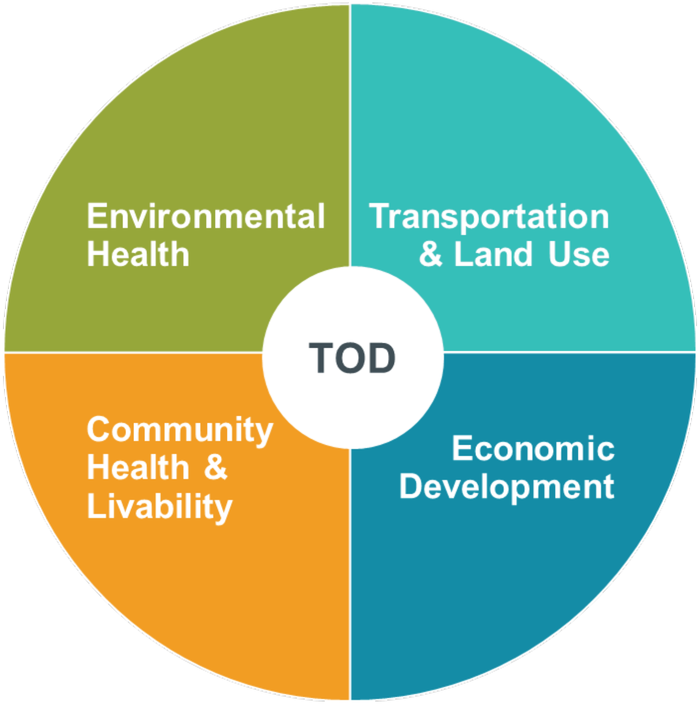


Figure 35: Key Topics in C/TOD Performance

COD, like any land development pattern, is a process that happens gradually over time. Regularly monitoring development trends and the characteristics of communities near transit can help localities assess the effectiveness of their COD implementation strategies and adjust as needed. COD's become TOD's with the advent of rapid transit.

Drawing on the previously completed Literature Review, Figure 35 identifies several key topics addressed by C/TOD, including transportation and land use, economic development, community health and livability, and environmental health. These topics organize the PMT framework, focusing on existing or potential TOD areas in the six SMART Plan rapid transit corridors. The next section expresses goals distilled from SMART Plan documents related to each topic, setting the stage for performance evaluation and reporting objectives for the PMT.

SMART Plan Goals and Performance Monitoring

The SMART Plan identifies six rapid transit corridors for expansion of the existing regional rapid transit network and aims to manage regional growth in concert with local plans by focusing new development in urban centers that are or may become rapid transit station areas. Through this integration of land use and transportation planning, the SMART Plan aims to optimize development and travel patterns to support continued economic development, foster healthy lifestyles, and minimize the impacts of growth on the natural environment.

While the SMART Plan does not enumerate explicit goals related to TOD in rapid transit corridors, its guiding themes and principles indicate an alignment with common TOD strategies and objectives. Figure 36 offers paraphrased SMART Plan goals that can be addressed through TOD, organized by the key topics listed above. These goals express in general terms the desired outcomes of a TOD growth strategy. Therefore, effective TOD policies and implementation will support the attainment of these goals locally and regionally.

SMART PLAN GOAL	KEY TOPIC	TOOLKIT NEED
Optimize where we live, work, shop, and play and how we travel	Transportation and Land Use	Monitor growth and development in TOD areas, utilization of non-auto travel modes, and productivity of the transportation system
Attract investment through strategic infrastructure and operations enhancements	Economic Development	Understand the characteristics of added jobs, businesses, and residents alongside transportation investments
Provide safe streets and public spaces that promote active lifestyles and offer community gathering places	Community Health and Livability	Assess walkability, bike-ability, access to recreation and public spaces alongside crime and safety data
Minimize adverse impacts to the natural world	Environmental Health	Estimate air and water quality, energy consumption, and urban footprint impacts of new development

Figure 36: SMART Plan Goals related to Key TOD Performance Metrics

Elements of Effective TOD

Alongside the SMART Plan goals, Figure 36 identifies key analysis and reporting needs for the PMT. This organization provides a clear structure for performance measurement, monitoring, and reporting by prompting consideration of criteria that can indicate the extent to which the stated goals are being achieved. In the PMT framework, these criteria are referred to as "elements" of effective TOD. They provide insight into where and how TOD is or is not directly contributing to the attainment of SMART Plan goals. These insights can then guide policy actions and prompt jurisdictional collaboration to maximize the potential benefits envisioned by the SMART Plan.

Each identified TOD topic is broken down into several elements that derive from the expressed SMART Plan goals and related toolkit needs, as shown in Figure 37. The elements are intended to indicate the effectiveness or successfulness of TOD in a given area and for a given time frame (see Geographic Summary Scales for Reporting and The Timing of TOD Effectiveness). Therefore, to the extent that each element is achieved, TOD is engendering desirable outcomes. By virtue of being organized into topic areas, an assessment can be made regarding the attainment of each goal.

For example, since the SMART Plan aims to optimize where we live, work, shop and play and how we travel, the PMT needs to monitor growth and development in TOD areas as well as the utilization of non-auto travel modes and the productivity of the transportation system. To the extent that…

- A significant proportion of regional growth occurs in SMART Plan corridors and station areas…
- Development patterns promote multimodal travel…
- Multimodal travel and productivity are increasing…

…TOD is helping the region attain the goals set forth in the SMART Plan related to the transportation and land use topic. If one of the elements is lacking, it may point to specific policy and/or investment needs at a regional or local scale. A short list of elements is envisioned for each topic area to keep reporting simple and offer clear insight into the factors associated with effective versus underperforming station areas and corridors.

Key Topic	Toolkit Need	Elements of Effective TOD
Transportation and Land Use	Monitor growth and development in TOD areas, utilization of non-auto travel modes, and productivity of the transportation system	• A significant proportion of regional growth is in SMART Plan corridors and station areas • Development patterns promote multimodal travel • Multimodal travel and productivity are increasing. • All residents enjoy similar access to jobs and essential goods and services.
Economic Development	Understand the characteristics of added jobs, businesses, and residents alongside transportation investments	• SMART Plan corridors and station areas attract jobs in targeted industry clusters. • New development in SMART Plan corridors and station areas generates a neutral or positive fiscal impact for local jurisdictions. • SMART Plan corridors and station areas offer marketable development and investment opportunities
Community Health and Livability	Assess walkability, bike-ability, access to recreation and public spaces alongside crime and safety data	• SMART Plan corridors and station areas offer opportunities for utilitarian/recreational trips by non-motorized modes. • SMART Plan corridors and station areas are safe and secure for active transportation. • Public spaces and SMART Plan transit stations are reachable by walking from most parcels in the station area.
Environmental Health	Estimate air and water quality, energy consumption, and urban footprint impacts of new development	• SMART Plan corridors and station areas generate fewer and shorter vehicular trips than the rest of the region. • Expansion of the region's development footprint is modest and consistent with local and regional plans. • Buildings in SMART Plan corridors and station areas consume less energy on average than those in the rest of the region.

Figure 37: Elements of Effective TOD by Topic Area

Determining whether (or to what extent) each element is achieved depends on underlying performance metrics driven by available data (see PMT Metrics and Data below). Since the PMT is envisioned to track trends over time and report on the effectiveness of TOD at multiple geographic scales, the specific derivation of supporting metrics varies based on the reporting time frame and location in focus. Before addressing these details, the next section outlines the time frames envisioned for PMT reporting. Geographic scales are then considered in the discussion of The PMT Interface.

The Timing of TOD Effectiveness

A stated goal of the PMT is to track TOD effectiveness over time with the intention of providing insight into several distinct planning questions. Each reporting time frame provides distinctive insight that can inform planning decisions. Four reporting time frames are envisioned, as shown in Figure 38.

TIME FRAME	KEY PLANNING QUESTIONS	REFERENCES	PRECISION
Snapshot of current conditions	• Where and to what extent is the region attaining to SMART Plan goals? • What gaps or impediments need to be addressed and which agencies/jurisdictions can partner to meet goals?	• Current year data	High
Historical Trends	• Where and to what extent is the region tracking toward SMART Plan goal attainment? • What factors have facilitated or impeded progress toward goals?	• Current year data • Baseline data	High
Near-term outlook	• Where and to what extent can near-term projects and developments be expected to boost attainment of SMART Plan goals? • What opportunities are available to influence near-term trends to boost progress toward goals?	• Current year data • Committed projects data	Moderate
Long-term outlook	• Where and to what extent do long-range forecasts and cost-feasible transportation priorities indicate stronger attainment of SMART Plan goals?	• Long-range forecast data • Long-range travel networks	Low

Figure 38: PMT Reporting Time Frames

Current year data will be used to offer a snapshot of TOD performance at a moment in time. This illuminates the status quo and can help identify corridors or station areas where policy action or public investment may be needed to attain SMART Plan goals.

The creation of baseline performance metrics that are updated at regular intervals provides the ability to assess the effectiveness of TOD and related policies over time. This historical trend information provides additional insight not available in the snapshot report. An underperforming station area, for example, may be trending in the desired direction, indicating that past policies and investments will lead to goal attainment if given additional time to mature. Longitudinal reporting is also expected to provide insight into the factors that drive development demand in TOD areas to support of long-range forecasting.

For both the snapshot and historical time frames that rely on observed data, the precision and reliability of metrics and element/topic findings is expected to be high. For forward-looking time frames, this is not the case. A near-term outlook is expected to be available based on information pertaining to committed developments and projects. This helps shape understanding of where ongoing development momentum is occurring and potentially impact planning in these areas. While there is always uncertainty regarding future development, this near-term outlook should offer a moderate level of precision and reliability. For the long-term outlook, based on regional forecast data, only a loose sense of TOD effectiveness can be provided. This perspective, however, can offer a preview of a mature TOD strategy in light of overall regional growth.

Limits of PMT Development Effort

While the PMT is intended to provide planners across the region with data and insight to answer key planning questions, the PMT is not envisioned to answer these questions directly. The extent to which the PMT may be used to inform policy, prioritize resources, or otherwise influence planning and programming is left to the discretion of relevant agencies and jurisdictions. Moreover, the datasets maintained within the PMT can be used for modeling and forecasting applications, but any such application will be separate from the PMT and its development and/or maintenance. In this way the PMT will serve as one of many tools for assisting in the region's long-term planning efforts.

The PMT Interface

The PMT will provide a simple interface for users, summarizing the story of TOD at various scales and in various time frames. The PMT will report on TOD effectiveness locally and regionally to learn what is working and what can be improved. While many users may use the PMT to obtain high-level reports on specific stations or corridors, the interface will facilitate the investigation of detailed metrics and data related to each TOD topic and element. Three data views – TOD Report Cards, TOD Effectiveness Ratings, and Performance Metrics – will give users the option of viewing a high-level snapshot or a detailed data analysis of a particular geography organized by topic or element.

The example images provided below are not intended to be a preview of the finalized interface visualizations. These examples are included to help demonstrate the overall organization of various interface elements and the possible ways in which each interface relates to other pieces of the PMT.

TOD Report Cards

TOD Report Cards provide a summary of TOD performance for each temporal reporting period at a glance. The vision for each Report Card is to create a visual representation of the performance of each geographic area that is easily understood and requires little interpretation. Figure 39 shows a very basic example of how the Report Cards could be organized. The gauge icons indicate performance levels for each TOD topic and across each time period; in this way a user could quickly understand the current state of TOD as well as how the current state compares to historical trends and future outlooks.

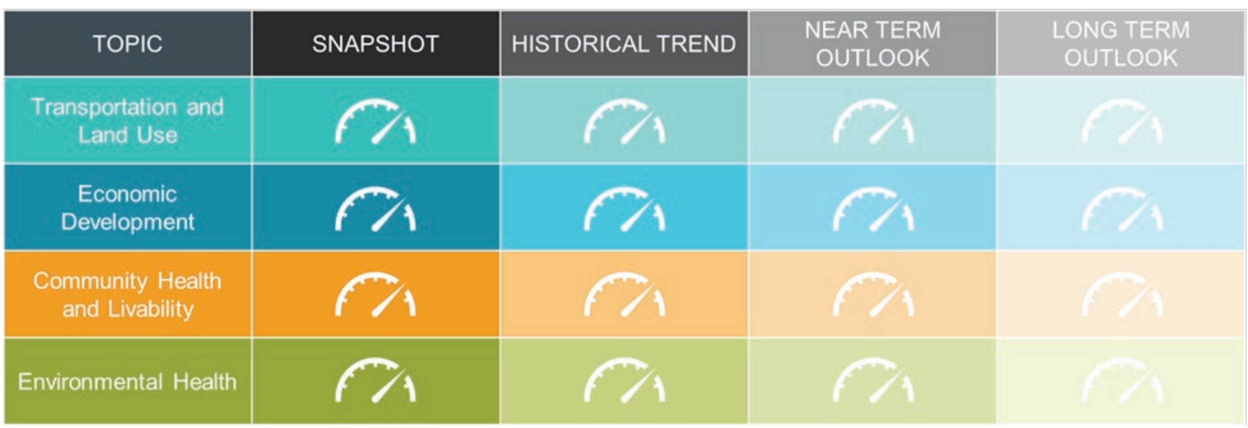


Figure 39: PMT Interface – TOD Report Card

TOD Effectiveness Ratings

A TOD Effectiveness Rating interface (see mock-up in Figure 40) will be generated if a user is interested in learning more about a particular TOD topic and time frame. By clicking on a cell of the TOD Report Card, for example, a topic-specific interface will be generated, providing insight into the performance rating indicated in the report card. The TOD Effectiveness Rating interface will address the elements of effective TOD, providing basic visualizations (represented as gauge icons here) that offer insight into the ways TOD is already working and how its effectiveness can be strengthened. In the example shown in Figure 40, it would be important to understand that a sub-optimal rating for the “Transportation and Land Use” topic is a result of development patterns that inadequately promote multimodal travel rather than a lack of development momentum. Exploration at this intermediate level of detail can provide rich insight into where and how TOD is effectively helping the region attain the SMART Plan goals.

TOPIC	HISTORICAL TREND	
Transportation and Land Use	A significant proportion of regional growth is in SMART Plan corridors and station areas.	
	Development patterns promote multimodal travel.	
	Multimodal travel and productivity are increasing.	

Figure 40: PMT Interface – TOD Effectiveness Ratings

Performance Metrics

Each element of effective TOD (reported in the TOD Effectiveness Ratings interface) is comprised of several performance metrics that are tailored to the geography and time frame being reported. The Performance Metrics interface features the most detailed information available in the PMT, drilling down into these metrics and data. Figure 41 shows a mock-up of the Performance Metrics interface. This example provides details on trends over time, indicating the extent to which each trend supports the statement: "Multimodal travel and productivity are increasing." Exploration at this level may reveal the underlying causes of sub-optimal TOD performance or provide important context related to performance. For example, travel trends often vary with national economic trends, so viewing detail related to travel data by year may highlight the influence of an economic downturn on the effectiveness of TOD related to transportation performance.

TOPIC	ELEMENT	HISTORICAL TREND	
Transportation and Land Use	Multimodal travel and productivity are increasing.	Change in unlinked passenger trips by transit since baseline year	
		Change in non-auto journey to work mode share since baseline year	
		Etc…	

Figure 41: PMT Interface – Performance Metrics

Geographic Summary Scales for Reporting

TOD Report Cards will be generated at multiple geographic scales. All three reporting levels – TOD Report Cards, TOD Effectiveness Ratings, and Performance Metrics – will be available regionwide, by SMART Plan Corridor, and by SMART Plan Station Area. As noted above, the formulation of specific metrics and data used to determine level of performance are likely to vary based on geographic resolution.

Visualizations

The PMT interface will include interactive maps to display topic, element, and metric-level details at corridor or station level throughout the region. Additionally, each interface may include interactive visualizations such as tables and charts, that reveal trends and forecasts, corridor or station rankings, lists of notable features, etc. Finally, selective side-by-side comparisons among features at the corridor or station area scales may be incorporated into the interface to highlight important trends and key differences.

PMT Metrics and Data

Key topics in TOD and elements of effective TOD are evaluated based on reliable and valid metrics that describe underlying system performance and reveal local and regional dynamics such as development and travel trends that shape and respond to TOD. Given the number of topics, elements, time frames, and geographic scales envisioned in the PMT, it is important to identify a limited number of critical metrics rather than a multiplicity of approximations or indicators.

This section outlines the characteristics of ideal metrics for TOD evaluation and highlights factors that may limit the extent to which any PMT metric exhibits all of these attributes. Having a sense of what constitutes an ideal metric, however, can guide the selection of appropriate metrics to include in the PMT. It then provides an initial list of potential metrics to include in the PMT. These will be refined as the PMT beta application is developed. Finally, the section closes with potential sources of data to support metric development.

Ideal Metrics and Limitations

To provide a focused reporting tool, the PMT will seek to establish strong metrics for each topic and element of TOD, exhibiting the following characteristics:

- Directly and completely reflects (an aspect of) a system's functional performance
- Meaningful for all uses/users of a system
- Provides meaningful comparisons for arbitrary aggregations of features
- Allows tracking of changes over time such that observed changes provide clear indications of performance improvements or degradations
- Provides insight into the underlying causes of unsatisfactory performance

Potential metrics associated with effective TOD often entail complexity. In some cases, they may be developed through the combination of multiple datasets. In others, they may require complex calculations, application of iterative methods, and/or estimation of key information. In such cases, an additional criterion can be stipulated for an ideal metric: that all supporting data and processes are consistent in their temporal and/or spatial boundaries. For example, the calculation of access to jobs by transit requires data showing where jobs are located and a transit network that can be used in route solving applications. General Transit Feed Specification (GTFS) feeds are often used to represent transit networks and schedules. Ideally, GTFS feeds representing distinct agencies (Miami-Dade Transit, South Florida Regional Transportation Authority, e.g.) would represent identical time periods, reflecting scheduled transit services that are typically active at the same time. However, agencies may not always publish feeds covering overlapping date ranges. In most cases, this would not preclude the combination of feeds from different agencies in analyzing transit accessibility, but it would undermine the accuracy of the metric in unquantifiable ways.

Given the expected complexity of PMT metrics, it is unlikely that most selected metrics will exhibit all the characteristics of an ideal measure. Factors that curtail the ability to generate ideal metrics for the PMT may include:

- **Data availability, accuracy, precision, and completeness**. Data development intrinsically requires balancing accuracy, precision, and complete feature representation and/or attribute detail. Datasets that offer all of these may entail substantial development and quality assurance protocols that elevate costs, limiting availability.
- **Process**: The development of many metrics will be defined by specific processing steps. These steps are generally informed by reasonable assumptions and well-documented methodologies. However, it is rare that these processing steps offer complete certainty with respect to the validity of derived outputs, and exceptions to the driving assumptions can often be postulated.
- **Knowledge**: Metrics that rely on correlations, statistical models, or typical rates in their calculation are estimates that may account for many influential factors but fundamentally cannot account for all potential sources of variance in the output value. The unexplained variance can be described as a knowledge gap, where a more perfect understanding of the variable being estimated could yield a better metric.
- **Efficiency:** Some metrics may be available through detailed modeling, data collection, or simulation, for which the needed computational power, processing time, and/or algorithmic efficiency precludes their development as part of the PMT.

Potential Metrics in the PMT

Keeping in mind the criteria for an ideal metric (and potential limitations), several measures have been identified for potential use in the PMT, listed in the tables below.

Land Use and Transportation

ELEMENT	METRIC
A significant proportion of regional growth is in SMART Plan corridors and station areas	Number of dwelling units
	Number of jobs
	Square footage by use type
Development patterns promote multimodal travel	Jobs:housing balance
	Land use diversity/interaction
	Number of off-street public parking spaces
	Network density, regularity, centrality
Multimodal travel and productivity are increasing.	Transit ridership
	Non-auto mode share
	Utilization of local shared mobility services
	Utilization of off-street public parking
	Complete person trips per minute
All residents enjoy access to jobs and essential goods and services.	Multimodal access to jobs by population group
	Multimodal access to essential goods and services by population group

Economic Development

ELEMENT	METRIC
SMART Plan corridors and station areas attract jobs in targeted industry clusters.	Jobs by industry cluster
New development in SMART Plan corridors and station areas generates a neutral or positive fiscal impact for local jurisdictions.	Estimated tax revenues
	Estimated costs of public services
SMART Plan corridors and station areas offer marketable development and investment opportunities	Vacant and re-developable acreage
	Fragmentation of unbuilt areas
	Competitive access among workers and jobs
	Estimated market value of parcels

Community Health and Livability

ELEMENT	METRIC
SMART Plan corridors and station areas offer opportunities for utilitarian and recreational trips by non-motorized modes.	Balanced accessibility by walking and biking between jobs and households, non-work destinations and households
	Acres of parks and public spaces accessible by walking per capita
SMART Plan corridors and station areas are safe and secure for active transportation.	Bicycle and pedestrian crashes
	Violent crime incidents
	Level of traffic stress/quality of service scores for walking and biking networks
Public spaces and SMART Plan transit stations are reachable by walking from most parcels in the station area.	Share of parcels within 15-minute walk of transit station, public spaces
	Minimum walk time to transit stations, public spaces
	Walking route directness to transit stations, public spaces

Environmental Health

ELEMENT	METRIC
SMART Plan corridors and station areas generate fewer and shorter vehicular trips than the rest of the region.	Vehicle miles of travel
	Average vehicle trip length
Expansion of the region's development footprint is modest and consistent with local and regional plans.	Impervious surface area
	Share of building square footage beyond growth boundary
Buildings in SMART Plan corridors and station areas consume less energy on average than those in the rest of the region.	Building energy consumption

Data Needs and Availability

PMT metric development will ultimately rely on readily available datasets that are routinely updated to ensure metrics can be reliably generated over the course of several years in order to track TOD effectiveness over time. Some data are likely to be sourced from local jurisdictions and partner agencies, which will require coordination between the TPO and the respective data providers to ensure local data are consistent in content and reliably available for use in the PMT. Figure 42 identifies the key data needed for developing the metrics listed above, in terms of content rather than source. These needs help tailor data acquisition efforts supporting the PMT and may prove useful in future years of PMT updating if data availability changes and alternative sources are deployed to meet analysis needs.

TOPIC AREA	DATA NEED
Land use, activity, socio-economic and demographic	**Population (stratified by demographics)**
	Households (stratified by socio-economic characteristics)
	Jobs (stratified by industry and socio-economic characteristics of workers)
	Land use (stratified by generalized types)
	Land Cover
	Number of dwelling units (stratified by building type, subsidy)
	Land value
	Dwelling unit/building market value
	Taxable value
	Vacancy
	Parks, recreational facilities, public spaces
	Essential goods and services outlets
	Building footprints and total square footage (stratified by land use)
	Off street public parking facilities and spaces
Travel networks, attributes, counts	**Costs of traversal (time, distance, monetary, etc.) by time of day**
	Transit station locations
	Rate of utilization (counts) by time of day
	Availability/quality of service by time of day
	Origin-destination travel activity
Miscellaneous	**Growth boundary**
	Energy consumption rates by building typology
	Industry/economic cluster affiliations
	Crime and safety (crash) locations/statistics
	Parking utilization statistics
	Unit costs of government services
Temporal groupings	Existing
	Committed
	Forecasted
	Target

Figure 42 Data Needs for PMT Metric Development

Figure 43 provides a data inventory template in which basic metadata associated PMT data sources will be recorded. A data inventory is provided as an appendix to this report. The data inventory is expected to evolve over the course of the PMT development process. Examples of expected sources for PMT data include:

Open data hubs and APIs

- Miami-Dade County
- FDOT
- Census/ACS
- LODES

Local/regional/state data

- RER
- FDOR
- FSUTMS Online

Collaboration with local and regional partners

- Development pipeline/building permits
- Licensed vendor data

TYPE	**TABULAR, SPATIAL**
Shape type (if spatial)	Polygon, Polyline, Point
Geographic scope	Coverage, jurisdictions included/excluded, etc.
Geographic scale/precision	Unit of geography for each feature
Latest Vintage	Epoch reflected in data
Date published	
Update schedule	How frequently are data generated?
Publisher/provider	
Mechanism to obtain	API, download, direct coordination, etc.
Key Fields	**Field name**: description **Field name**: description …
Related tables	**Table**: Key Field **Table**: Key Field …
Observed quality concerns	Describe limitations or data quality issues worth noting

Figure 43 Data Inventory Template

APPENDIX: DATA INVENTORY

Census Total Population Data

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract
Latest Vintage	2018 5-Year
Date published	-
Update schedule	December each year. 2019 data will be available December 2020.
Publisher/provider	US Census Bureau
Mechanism to obtain	TableID: B01003. **Population API -** https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0001E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510
Key Fields	**DP05_0001E**: Total population count
Related tables	**-**
Observed quality concerns	-

Census Total Population-Race Data

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract
Latest Vintage	2018 5-Year
Date published	-
Update schedule	Annually
Publisher/provider	US Census Bureau
Mechanism to obtain	TableID: B02001. Data is available at the tract and block group level. APIs below represent count data from selected racial categories and may not sum to 100 percent. **Race – White:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0037E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 **Race – "Latin or Hispanic as Race":** https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0070E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510

	Race – Black: https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0038E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 **Race – Native American:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0039E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 **Race – Asian:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0044E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 **Race – Some other listed race:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP05_0057E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510
Key Fields	Main field ID will identify racial identity associated with header list.
Related tables	-
Observed quality concerns	-

Census Total Means of Transportation to Work

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract
Latest Vintage	2018 5-Year
Date published	-
Update schedule	December each year. 2019 data will be available December 2020.
Publisher/provider	US Census Bureau
Mechanism to obtain	TableID: B08301. Data is available at the tract and block group level. APIs below represent count data from selected commute categories and may not sum to 100 percent. Means of Transportation – **Drove a Car**, Truck, or Van to Work Alone, for 16+ workers, to work: https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0019E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 Means of Transportation – **Public Transportation**, for 16+ workers, to work: https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0021E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 Means of Transportation – **Walked**, for 16+ workers, to work:

	https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0022E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 Means of Transportation – **Worked from home**, for 16+ workers: https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0024E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510
Key Fields	First-named field corresponds to keyword in description.
Related tables	-
Observed quality concerns	-

Census Median Household Income

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract
Latest Vintage	2018 5-Year
Date published	-
Update schedule	December each year. 2019 data will be available December 2020.
Publisher/provider	US Census Bureau
Mechanism to obtain	TableID: B19013. Data is available at the tract and block group level. API: https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0062E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510
Key Fields	Includes self-evident count fields of median household income for each geography unit.
Related tables	-
Observed quality concerns	-

Census Total Housing Units

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract

Latest Vintage	2018 5-Year
Date published	-
Update schedule	December each year. 2019 data will be available December 2020.
Publisher /provider	US Census Bureau
Mechanism to obtain	TableID: B25001: **Housing unit estimate API:** *https://api.census.gov/data/2018/acs/acs5/profile?get=DP02_0001E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510*
Key Fields	DP02_0001E: Housing units in tract.
Related tables	-
Observed quality concerns	[Will be interesting to see how this interacts with MDC-produced housing unit data.]

Census Housing Units in / by Structure

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract
Latest Vintage	2018 5-Year
Date published	-
Update schedule	December each year. 2019 data will be available December 2020.
Publisher/provider	US Census Bureau
Mechanism to obtain	TableID: B25024. Data is available at the tract and block group level. The APIS below contain count data on units and units by structure. **Totals Units:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP04_0001E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 **Detached Units in Structure 1:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP04_0007E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff8155988f57795c11fa4510 **Units in Structure 5 to 9:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP04_0011E&f

	or=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff815598 8f57795c11fa4510 **Units in Structure 10 to 19:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0012E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff815598 8f57795c11fa4510 **Units in Structure 20+:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0013E&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff815598 8f57795c11fa4510
Key Fields	Total units provide count data by geography.
Related tables	-
Observed quality concerns	[Will be interesting to see how this interacts with MDC-produced housing unit data.]

Census Family Poverty Status in the Past 12 Months by Family Type

5-year estimates

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Miami-Dade County
Geographic scale/precision	Census block group / Census Tract
Latest Vintage	2018 5-Year
Date published	-
Update schedule	December each year. 2019 data will be available December 2020.
Publisher/provider	US Census Bureau
Mechanism to obtain	TableID: B17010. Data is available at the tract and block group level.
Key Fields	Main fields of interest **Income in the past 12 months below poverty level:** (1) children under 5 living in HH (2) children under 5, plus children 5 – 17 living in HH (3) children between 5 – 17 living in HH (4) and HHs with no children **An API with percentage data on the "percent of families and people whose income in the past 12 months is below the poverty level" is available from:** https://api.census.gov/data/2018/acs/acs5/profile?get=DP03_0119PE&for=tract:*&in=state:12&in=county:086&key=335a884a466eb18cff81559 88f57795c11fa4510
Related tables	-
Observed quality concerns	On the API tool, some tracts have values that are null representations (66666,66).

Building Permits

Building permits issued in Miami-Dade County within the last 3 years

Type	Spatial
Shape type (if spatial)	Point
Geographic scope	Miami Dade County (unincorporated only)
Geographic scale/precision	Address
Latest Vintage	May 28, 2020
Date published	May 28, 2020
Update schedule	Weekly (Fridays)
Publisher/provider	Miami Dade County (unincorporated)
Mechanism to obtain	API *https://opendata.arcgis.com/datasets/31cd319f45544648b59f0418aea60091_0.geojson*
Key Fields	**TYPE**: Building permit type **APPTYPE**: Foreign key to Tbl Bldg Permit App Type **PROPUSE**: Foreign key to Tbl Bldg Permit Prop Use **DESC1**: Building permit description **ISSUDATE**: When the permit was issued (some dates are very old, suggesting this either isn't the last 3 years or perhaps the field is related to something other than the building permit??) **RESCOMM**: whether the permit is for residential or commercial property **ESTVALUE**: Dollar value of the permitted activity/improvement?
Related tables	**Tbl Bldg Permit App Type**: CODE **Tbl Bldg Permit Prop Use**: CODE
Observed quality concerns	• Since the building permits data contains information on new developments, there will always be rejects in the street geocoding process as there is always a lag between the time new developments (streets, properties) are entered in the Building Department's database and the time when the same information is available in the coverages ADADE and LPROP.

Tbl Bldg Permit App Type

A table of Miami-Dade County Building permits application types. To identify the code and code descriptions of Building Permits application types.

Type	Tabular
Shape type (if spatial)	NA
Geographic scope	NA
Geographic scale/precision	NA
Latest Vintage	6/7/2020
Date published	6/7/2020
Update schedule	Weekly (Fridays)
Publisher/provider	Miami Dade County (unincorporated)
Mechanism to obtain	API *https://opendata.arcgis.com/datasets/cf047c70557a49e1a6848a62081bf93c_0.geojson*
Key Fields	**CODE**: Primary Key **CODEDESC**: plain language description of application type
Related tables	
Observed quality concerns	• Potential need to generalize application type codes; unclear if codes are a static set or if new codes could be added at a later date.

Tbl Bldg Permit Prop Use

A table of Miami-Dade County Building property uses. To identify the code and code descriptions of Building permits property use.

Type	Tabular
Shape type (if spatial)	NA
Geographic scope	NA
Geographic scale/precision	NA
Latest Vintage	6/7/2020
Date published	6/7/2020
Update schedule	Weekly (Fridays)
Publisher/provider	Miami Dade County (unincorporated)
Mechanism to obtain	API *https://opendata.arcgis.com/datasets/5c1986f08a5a40b4be8a55418c833da8_0.geojson*
Key Fields	**CODE**: Primary Key **CODEDESC**: plain language description of property use
Related tables	
Observed quality concerns	• Potential need to generalize property use codes; unclear if use codes are a static set or if new codes could be added at a later date.

Statewide Land Use Cover: Florida DEP

This is a table from the Florida Department of Environmental Protection covering land use covering information.

Type	Tabular
Shape type (if spatial)	Polygon
Geographic scope	Cross-section of Florida including Miami-Dade County
Geographic scale/precision	-
Latest Vintage	2014-16
Date published	N/A
Update schedule	N/A
Publisher/provider	Florida Department of Environmental Protection
Mechanism to obtain	API: https://opendata.arcgis.com/datasets/2f0e5f9a180a412fbd77dc5628f28de3_3.geojson. URL access: https://geodata.dep.state.fl.us/datasets/2f0e5f9a180a412fbd77dc5628f28de3_3/geoservice?geometry=-80.208%2C25.767%2C-80.180%2C25.774.
Key Fields	**LEVEL2LANDUSECODE**: Keyed land use code hierarchy **LANDUSE DESCRIPTION**: plain language description of LEVEL2 code type
Related tables	-
Observed quality concerns	-

National Land Use Cover: MRLC

This is a table from the Florida Department of Environmental Protection covering land use covering information.

Type	Spatial
Shape type (if spatial)	Polygon
Geographic scope	CONUS
Geographic scale/precision	-
Latest Vintage	2016
Date published	2017-
Update schedule	N/A
Publisher/provider	Multi-Resolution Land Characteristics Consortium
Mechanism to obtain	URL access: https://www.mrlc.gov/data/nlcd-2016-land-cover-conus.

Key Fields	Classification_description: see "related tables."
Related tables	Translation table for classification description is available from: https://www.mrlc.gov/data/legends/national-land-cover-database-2016-nlcd2016-legend.
Observed quality concerns	-

Building Footprints

2015-LIDAR data from the Open Data Hub.

Type	Spatial
Shape type (if spatial)	Polygon
Geographic scope	Area within the UGB
Geographic scale/precision	-
Latest Vintage	2015
Date published	-
Update schedule	-
Publisher/provider	Miami-Dade County
Mechanism to obtain	API: https://opendata.arcgis.com/datasets/d511e9ebc5aa4f49a23ff5fa2fb99786_0.geojson.
Key Fields	**Source:** {L, P} where L = LiDAR, P = MDC Planimetric **Bld_type** : {S, L} where S = Small Buildings, L = Large Buildings
Related tables	-
Observed quality concerns	There is a *z* field for height, but I could not preview the data. The title is "2D," so height data may not be complete.

Urban Growth Boundary (Urban Development Boundary)

2015-LIDAR data from the Open Data Hub.

Type	Spatial
Shape type (if spatial)	Polyline
Geographic scope	MDC
Geographic scale/precision	-
Latest Vintage	2020
Date published	2014
Update schedule	"UDB to be reviewed sometime in 2020"
Publisher/provider	Miami-Dade County

Mechanism to obtain	API: https://opendata.arcgis.com/datasets/a468dc11c02f4467ade836947627554b_0.geojson
Key Fields	-
Related tables	-
Observed quality concerns	-

Taxable Values - DOR

Type	Tabular
Shape type (if spatial)	Polygon
Geographic scope	Miami-Dade County
Geographic scale/precision	-
Latest Vintage	March 2019
Date published	7/9/20
Update schedule	Files published annually during summer.
Publisher/provider	Florida Department of Revenue
Mechanism to obtain	Navigate to the Florida DOR Data Hub: https://floridarevenue.com/property/Pages/DataPortal.aspx > Download Tax Data > Download Tax Roll Directory > Select desired year for "NAL – SDF – DBF" > Select county number 23 for Miami-Dade.
Key Fields	-
Related tables	-
Observed quality concerns	-

Taxable and assessed values - MDC Property Appraiser

Miami-Dade property appraiser values.

Type	Tabular*
Shape type (if spatial)	-
Geographic scope	Countywide
Geographic scale/precision	Parcel
Latest Vintage	2019
Date published	2020
Update schedule	Annually, following tax season
Publisher/provider	Miami Dade County
Mechanism to obtain	The Property Appraiser produces reports, but does not seem to openly produce a file containing the values associated with each parcel. It may be available through a $50 "commonly requested items" package from: http://bbs.miamidade.gov/, but there is no description of what is included in this package.
Key Fields	-
Related tables	-
Observed quality concerns	-

Parcel shapes

Parcel shapes from MDC Data

Type	Spatial
Shape type (if spatial)	Polyline
Geographic scope	Countywide
Geographic scale/precision	-
Latest Vintage	6/22/2020
Date published	6/22/2020
Update schedule	Monday – Saturday
Publisher/provider	Miami Dade County
Mechanism to obtain	API *https://opendata.arcgis.com/datasets/347bce97227c4b54b04a3e626b558950_0.geojson*
Key Fields	-
Related tables	-
Observed quality concerns	-

County land use

Type	Spatial

Shape type (if spatial)	Polygon
Geographic scope	Countywide
Geographic scale/precision	Parcel
Latest Vintage	Appears to be regularly updated, but is not stamped
Date published	Appears to be regularly updated, but is not stamped
Update schedule	-
Publisher/provider	Miami Dade County
Mechanism to obtain	API: https://opendata.arcgis.com/datasets/244e956692d442c3beaa8a89259e3bd9_0.geojson
Key Fields	**LU**: Land use code **Descr**: Land use code description.
Related tables	-
Observed quality concerns	No update schedule posted, but Memo claims that it is "updated weekly" and contains data since 1994.

Certificates of Use

Type	Tabular
Shape type (if spatial)	N/A
Geographic scope	Unincorporated Miami-Dade County
Geographic scale/precision	Address data
Latest Vintage	6/22/2020
Date published	6/23/2020
Update schedule	Weekly (Fridays)
Publisher/provider	Miami Dade County
Mechanism to obtain	API *https://opendata.arcgis.com/datasets/fcdc6019414c45898715103dcbc4b5c7_0.geojson*
Key Fields	**CertOccIssueDate**: Date on which certificate of use (CU) was granted **CoProcessDBAName**: The recognizable name of the business **CoProcessBusAddrProp**: Site address line **CoDataBusCity**: Site address, city line **CoDataBusZipCode**: Site address, Zip Code **ProcessBusCodeDesc**: Plain-English **top-level** description of businesses **CoDataBusUseComment**: Plain-English **detailed** description of business
Related tables	
Observed quality concerns	The web viewer does not display any shapes at all, and the "about" field is marked as "non-GIS." However, the data fields do contain address lines, so address matching should be possible. Data appears regularly updated, and dates back to 2003.

Certificates of Occupancy

Type	Tabular
Shape type (if spatial)	N/A
Geographic scope	Unincorporated Miami-Dade County
Geographic scale/precision	Address data
Latest Vintage	6/22/2020
Date published	6/23/2020
Update schedule	Weekly (Fridays)
Publisher/provider	Miami Dade County
Mechanism to obtain	API *https://opendata.arcgis.com/datasets/c2d2450697d44d418c8fac19c1b7547f_0.geojson*
Key Fields	**IssueDate**: Date on which certificate of occupancy (CO) was granted **Property_Address**: Site address line **Number_Of_Occupants_Per_Floor**: The number of allowable occupants allowed under the CO **Square_Footage**: Square footage of the building footprint.
Related tables	
Observed quality concerns	The web viewer does not display any shapes at all, and the "about" field is marked as "non-GIS." The data fields do contain address lines, so address matching should be possible. Data appears regularly updated, and date back to 2003. Load_Per_Floor & Occupants_Per_Floor seem to both have what look like erroneous values; for instance, displaying 99,999. Square_Footage may also have this affliction: reading values of 9,999,999. The Square_Footage example is possible certainly, whereas the occupants result would not be. Both instances tend to be for much older COs...

Energy Consumption of Buildings

2015-LIDAR data from the Open Data Hub.

Type	Spatial
Shape type (if spatial)	Polygon
Geographic scope	Area within the UGB
Geographic scale/precision	-
Latest Vintage	2015
Date published	-
Update schedule	-
Publisher/provider	Miami-Dade County

Mechanism to obtain	API: https://opendata.arcgis.com/datasets/d511e9ebc5aa4f49a23ff5fa2fb99786_0.geojson.
Key Fields	**Source:** {L, P} where L = LiDAR, P = MDC Planimetric **Bld_type** : {S, L} where S = Small Buildings, L = Large Buildings
Related tables	-
Observed quality concerns	There is a *z* field for height, but I could not preview the data. The title is "2D," so height data may not be complete.

New Businesses: InfoGroup

Type	Spatial
Shape type (if spatial)	Polyline
Geographic scope	Miami-Dade County
Geographic scale/precision	Address
Latest Vintage	Current
Date published	Current
Update schedule	Weekly(?)
Publisher/provider	MPO / Miami GIS
Mechanism to obtain	TBD. Data is publicly available from https://gisweb.miamidade.gov/businesstracker/, but no download link is clear.
Key Fields	**New Businesses** **Closed Businesses**
Related tables	**TBD-**
Observed quality concerns	TBD-

Bike facilities

A line feature class of Miami-Dade County, Metropolitan Planning Organization (MPO) existing bike lanes.

Type	Spatial
Shape type (if spatial)	Polyline
Geographic scope	Miami-Dade County
Geographic scale/precision	Road?
Latest Vintage	March 2019
Date published	May 5, 2019
Update schedule	Annually
Publisher/provider	MPO / Miami GIS
Mechanism to obtain	https://opendata.arcgis.com/datasets/b874dd0e2d0941a689c56f54ae72d67c_0.geojson
Key Fields	**Limits:** Describe bike lane boundaries with street name endpoints. **Length:** Length in miles. **Shape_Length:** Length in meters.
Related tables	**None.**
Observed quality concerns	Does not indicate bike lane directionality –may be an issue if bike lanes exist on only one direction of a roadway. Maybe a rare drawback.

MDT Routes, Stops

Type	Spatial
Shape type (if spatial)	Polyline, point
Geographic scope	Miami-Dade County
Geographic scale/precision	Road / Lower than address (for stops)
Latest Vintage	N/A
Date published	May 27, 2020
Update schedule	As needed, no regularized schedule
Publisher/provider	MPO / Miami GIS
Mechanism to obtain	http://www.miamidade.gov/transit/googletransit/current/google_transit.zip - there may be an API
Key Fields	**TrainStations** **MoverStations** **BusRouteStops**
Related tables	**None.**
Observed quality concerns	

FDOT 5-Year Work Plan

Type	Spatial
Shape type (if spatial)	Polyline
Geographic scope	State of Florida
Geographic scale/precision	-
Latest Vintage	July 2018
Date published	July 1, 2019
Update schedule	Yearly, but not updated at this location.
Publisher/provider	FDOT
Mechanism to obtain	No GeoJSON Link, but option to Open in ESRI apps: https://fdot.maps.arcgis.com/home/item.html?id=f37669322ceb43e19a2b56eabf4c74bf#overview. Future year versions will need to be obtained through the FDOT portal.
Key Fields	**District**: FDOT District **County_Code**: Two-digit designation (87 is Miami-Dade) **Description**: Description of project extents **Project_Type**: Description of main project purpose (lighting, sidewalk).
Related tables	
Observed quality concerns	None

Financially Assisted Units (Subsidized Housing)

Type	Tabular
Shape type (if spatial)	-

Geographic scope	Miami-Dade County (includes incorporated areas)
Geographic scale/precision	Address
Latest Vintage	2019 (latest data in set)
Date published	N/A
Update schedule	Annually
Publisher/provider	The Shimberg Center
Mechanism to obtain	Direct request and download from Shimberg data page; http://flhousingdata.shimberg.ufl.edu/. Query: Miami-Dade County | Assisted Housing Inventory
Key Fields	**Street Address:** Street address line **City**: City line for address field **Zip Code**: Address zip **Housing Programs**: The Federal / State programs that contribute to the housing project **Assisted Units**: The number of units in the project that receive some sort of assistance (contrast with total units) **Affordability Start Date**: The first year in which assisted units came online **Overall year of subsidy expiration**: The year in which *current* assistance ends, or in which legal covenants protecting affordability associated with assistance expire
Related tables	Fields are in "plain English," but full definitions are still available at: http://flhousingdata.shimberg.ufl.edu/AHI-user-guide
Observed quality concerns	None

Parks, Incorporated Areas

Type	Spatial
Shape type (if spatial)	Polygon
Geographic scope	Incorporated areas of Miami-Dade
Geographic scale/precision	Address
Latest Vintage	N/A
Date published	March 26, 2019
Update schedule	Claims bi-annually; since this has not been updated in 1.2 years, "bi-annually" may mean every *two* years rather than *twice per* year.
Publisher/provider	MPO / Miami GIS
Mechanism to obtain	https://opendata.arcgis.com/datasets/16fe02a1defa45b28bf14a29fb5f0428_0.geojson
Key Fields	**City**: Governing municipality **Class**: Description of park type ("mini-park," "path," e.g.) **Developed**: Seems to indicate level of completeness with respect to the park's planning intent.
Related tables	**None.**
Observed quality concerns	

Parks, Unincorporated Areas

Type	Spatial
Shape type (if spatial)	Polygon

Geographic scope	Unincorporated areas of Miami-Dade
Geographic scale/precision	Address
Latest Vintage	N/A
Date published	March 26, 2019
Update schedule	Claims bi-annually; since this has not been updated in 1.2 years, "bi-annually" may mean every *two* years rather than *twice per* year.
Publisher/provider	MPO / Miami GIS
Mechanism to obtain	https://opendata.arcgis.com/datasets/aca1e6ff0f634be282d50cc2d534a832_0.geojson
Key Fields	**City**: Field refers to Miami-Dade County. **Class**: Description of park type ("mini-park," "path," e.g.)
Related tables	-
Observed quality concerns	-

USDA Food Atlas (Food Desert Locator)

Type	Tabular
Shape type (if spatial)	-
Geographic scope	United States
Geographic scale/precision	Census tract
Latest Vintage	2015 (latest data in set)
Date published	2019
Update schedule	N/A
Publisher/provider	USDA
Mechanism to obtain	Data is available as an .xls file from https://www.ers.usda.gov/data-products/food-access-research-atlas/. Census-tract data.
Key Fields	**County:** Useful to filter data to Miami-Dade **LILATracts_1and10:** Low-income (LI) and Low-access populations without access to supermarkets at 1-mile travel distance for urban areas. This is the most common statistic from this dataset.
Related tables	.xls file contains a worksheet with a translation table.
Observed quality concerns	Somewhat out of date.

Crash Locations

Type	Spatial
Shape type (if spatial)	Point
Geographic scope	State of Florida
Geographic scale/precision	7-decimal degree lat/long.
Latest Vintage	2017
Date published	2017
Update schedule	Lag of three+ years, I think recent crash locations are generally protected and difficult to distribute under Florida Statues.
Publisher/provider	FDOT
Mechanism to obtain	Yearly .zip files (ostensibly?) contain plotted crash points along with severity indicators. They can be downloaded at: https://ubr.fdot.gov/basemaps/category/52, or from the FDOT Unified Base Roads site. The folders seem to contain GIS files.

Key Fields	**County_TXT:** Displays county in which crash occurred, numbered. 87 is Miami-Dade. **Crash_Dade:** Date on which the crash occurred, according to the crash report. **MOST_HARM_:** A set of codes describing the nature of the collision, i.e. Code #10 – Collision w/ pedestrian
Related tables	.zip download contains a .pdf explainer document, including a translation table for the MOST_HARM_: field.
Observed quality concerns	The 2017 dataset indicates that it was "only 70% finished" when published… Sure has been a while since then.

Air quality attainment (EPA)

Type	Tabular
Shape type (if spatial)	-
Geographic scope	Floridian Counties
Geographic scale/precision	Countywide
Latest Vintage	20
Date published	May 2020
Update schedule	N/A
Publisher/provider	US-EPA
Mechanism to obtain	https://www3.epa.gov/airquality/greenbook/anayo_fl.html - data is available in .xls and.dbf downloadable formats.
Key Fields	**Nonattainment In Year:** displays whether the county is non-attaining, if the county appears
Related tables	-
Observed quality concerns	-

Air quality (local monitoring sites)

Type	Spatial
Shape type (if spatial)	Point
Geographic scope	Miami-Dade
Geographic scale/precision	7-digit lat/long(?)
Latest Vintage	Current as of writing
Date published	Current as of writing
Update schedule	Hourly/daily (beacon data)
Publisher/provider	Florida Department of Environmental Protection (FDEP)
Mechanism to obtain	https://opendata.arcgis.com/datasets/bd163c857f1b48c2b1f1a3a15dc8163d_4.geojson
Key Fields	**AQI_VALUE**: Air quality index (0 – 200) **AQI_Category**: Standards for AQI index **O3_**: There are a variety of indicators for ozone. **NO2:** There are a variety of indicators for nitrogen dioxide. **SO2:** There are a variety of indicators for sulfur dioxide. **PM25:** There are some indicators for particular matter.
Related tables	-

Observed quality concerns	Not all stations report on the same data, and this is an aggregator.

Energy consumption by building typology

Type	-
Shape type (if spatial)	-
Geographic scope	-
Geographic scale/precision	-
Latest Vintage	-
Date published	-
Update schedule	-
Publisher/provider	-
Mechanism to obtain	-
Key Fields	-
Related tables	-
Observed quality concerns	-

Common Travel Elasticities in Broward, Florida

Paper published in Urban Studies with Broward-specific elasticities.

Type	-
Shape type (if spatial)	-
Geographic scope	Broward, but likely still applies to MDC.
Geographic scale/precision	-
Latest Vintage	-
Date published	-
Update schedule	-
Publisher/provider	UrbanStudies, Thomson, Brown, et al.,
Mechanism to obtain	"What Really Matters for Increasing Transit Ridership: Understanding the Determinants of Transit Ridership Demand in Broward County, Florida Gregory Thompson, Jeffrey Brown and Torsha Bhattacharya" https://journals.sagepub.com/doi/pdf/10.1177/0042098012443864?casa_token=kMrz6LYUajAAAAAA:NRMubUZCF2aDlHGm0GKWOYs6geiD4FTYyH7nFbsI5rKzdVSHWU_WYzaBE9QnUkszfjZyJa_At5g. See page 3340, Table 3.
Key Fields	-
Related tables	-
Observed quality concerns	-

H+T Index on the Costs of Commuting

Paper published in Urban Studies with Broward-specific elasticities.

Type	Tabular

Shape type (if spatial)	-
Geographic scope	Florida
Geographic scale/precision	Census Tracts
Latest Vintage	2018?
Date published	2020
Update schedule	N/A
Publisher/provider	Center for Neighborhood Technology
Mechanism to obtain	With a CNT Account (Free), go to the Data Download Section, and download the .xls for Florida Tracts. https://htaindex.cnt.org/download/data.php.
Key Fields	**HT_80ami**: Housing + Transportation Costs % Income for the Regional Moderate Household. *This is the key measure from the CNT.* **T_80ami**: Housing Costs % Income for the Regional Moderate Household **T_ami**: Transportation Costs % Income for the Regional Typical Household **Vmt_per_hh_ami**: Annual Vehicle Miles Traveled per Household for the Regional Typical Household **Vmt_per_hh_80ami**: Annual Vehicle Miles Traveled per Household for the Regional Moderate Household
Related tables	-
Observed quality concerns	Data timing opaque.

Public Parking Facilities (Miami Parking Authority)

The Miami Parking Authority (maybe) has data on public parking facilities throughout Miami-Dade County, including street and structured parking, and a request to them may reveal the number and nature of the parking facilities that they manage.

Type	Spatial
Shape type (if spatial)	Point
Geographic scope	Miami-Dade County, unclear if all municipalities participate
Geographic scale/precision	7-digit latitude/longitude
Latest Vintage	?
Date published	?
Update schedule	?
Publisher/provider	Miami Parking Authority
Mechanism to obtain	Coordination / public records request. https://commerce.miamiparking.com/facility/findparking.
Key Fields	**?**
Related tables	-
Observed quality concerns	Unsure if this data really exists, but their website does have a map showing parking facilities, *with pricing.*

Public Parking Facilities (Maintained by Miami Internal Services)

The Miami Parking Authority (maybe) has data on public parking facilities throughout Miami-Dade County, including street and structured parking, and a request to them may reveal the number and nature of the parking facilities that they manage.

Type	Tabular
Shape type (if spatial)	Address
Geographic scope	Miami-Dade County
Geographic scale/precision	Address
Latest Vintage	?
Date published	November 2019
Update schedule	?
Publisher/provider	Miami-Dade County
Mechanism to obtain	A posted list on the Miami website listing parking lots and the number of parking stalls, along with pricing, located at: https://www.miamidade.gov/facilities/parking-facilities.asp.
Key Fields	-
Related tables	-
Observed quality concerns	-

Parking Facilities (via OSM database query)

The Miami Parking Authority (maybe) has data on public parking facilities throughout Miami-Dade County, including street and structured parking, and a request to them may reveal the number and nature of the parking facilities that they manage.

Type	Spatial
Shape type (if spatial)	Point / polygon
Geographic scope	Miami-Dade County
Geographic scale/precision	Latitude / longitude
Latest Vintage	-
Date published	-
Update schedule	-
Publisher/provider	OpenStreetMap
Mechanism to obtain	A list of parking tags that may be queried from from an OSM API. **Amenity key = parking:** Includes surface, multi-storey (story), underground, rooftop, sheds, carport, and garage-boxes. **Amenity key = Park_ride**: parking facilities with park and ride designations from transit **Amenity key = Capacity**: number of stalls **Amenity key = parking:lane**=*:- shows parking facilities along major roadways **Amenity key = bicycle_parking:** Locations with bicycle parking facilities

Key Fields	-
Related tables	-
Observed quality concerns	-

Miami Dade Crime Map

Miami Dade seems to maintain spatial data for their crime reports and upload it to a "Crime Mapping.Com" for public review. Thre is no API or download options from this site. Coordination with MDC may be helpful

Type	Spatial
Shape type (if spatial)	Point
Geographic scope	MDC Police Dept
Geographic scale/precision	Latitude / longitude
Latest Vintage	-
Date published	-
Update schedule	Daily
Publisher/provider	Miami-Dade Police Department(?)
Mechanism to obtain	View public-facing information at: https://www.crimemapping.com/map/fl/miami-dadecounty.
Key Fields	-
Related tables	-
Observed quality concerns	-