Report No. UT-20.09

Impact of Mobility as a Service on Transit Access

Prepared For:

Utah Department of Transportation
Research & Innovation Division

Submitted By:

RSG, Inc.

Authored By:

Robert Chamberlin, P.E., PTOE
Nima Haghighi, Ph.D.

Final Report

July 2020

RESEARCH

UDOT Keeping Utah Moving

Utah Department of Transportation - Research Division
4501 South 2700 West - P.O. Box 148410 - SLC,UT 84114-8410

DISCLAIMER

The authors alone are responsible for the preparation and accuracy of the information, data, analysis, discussions, recommendations, and conclusions presented herein. The contents do not necessarily reflect the views, opinions, endorsements, or policies of the Utah Department of Transportation or the U.S. Department of Transportation. The Utah Department of Transportation makes no representation or warranty of any kind and assumes no liability, therefore.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Utah Department of Transportation (UDOT) for funding this research, and the following individuals from UDOT serving on the Technical Advisory Committee (TAC) for helping to guide the research:

- Richard Brockmyer (UDOT)
- Khaisy Vonarath (UDOT)

The authors would also like to acknowledge Bert Granberg and Chad Worthen of WFRC, Tim Hereth of MAG, and Shana Lindsay a consultant working with UDOT Research who served as Project Manager.

TECHNICAL REPORT ABSTRACT

<table>
<tr><td>1. Report No.
UT- 20.09</td><td>2. Government Accession No.
N/A</td><td colspan="2">3. Recipient's Catalog No.
N/A</td></tr>
<tr><td colspan="2" rowspan="2">4. Title and Subtitle
The Impact of Mobility as a Service on Transit Access</td><td colspan="2">5. Report Date
July 2020</td></tr>
<tr><td colspan="2">6. Performing Organization Code</td></tr>
<tr><td colspan="2">7. Author(s)
Robert Chamberlin, PE/PTOE, Nima Naghighi, PhD</td><td colspan="2">8. Performing Organization Report No.
RSG, Inc.</td></tr>
<tr><td colspan="2" rowspan="2">9. Performing Organization Name and Address
RSG, Inc.
307W 200S, Suite 2004
Salt Lake City, UT 84101</td><td colspan="2">10. Work Unit No.</td></tr>
<tr><td colspan="2">11. Contract or Grant No.
208256</td></tr>
<tr><td colspan="2" rowspan="2">12. Sponsoring Agency Name and Address
Utah Department of Transportation
4501 South 2700 West, PO Box 148410
Salt Lake City, UT 84114-8410</td><td colspan="2">13. Type of Report & Period Covered
Final, March – July 2020</td></tr>
<tr><td colspan="2">14. Sponsoring Agency Code
PIC No UT 19.502</td></tr>
</table>

16. Abstract

The emergence of a new mode, Mobility as a Service, has, to date, been most often characterized as the ridehailing mode provided by companies such as Uber and Lyft. If and when these private services evolve to incorporate autonomous vehicles, costs for this service will decrease, making it ever more competitive with other modes, both motorized and non-motorized. A potential future vision of Mobility as a Service in Utah was the topic of a prior UTRAC study, The Impact of Shared Autonomous Vehicles on VMT in Utah (Report No. UT-19.10). That work concluded that the existence of this new mode would lead to a 1-7% increase in trip making and a 4-9% increase in Vehicle Miles Traveled. The application of MaaS as a transit access mode was not addressed in that work.

This phase 2 study focuses on MaaS as a transit access mode. This mode is also referred to as microtransit. For this research, it was proposed to use the Wasatch Front Travel Model as the analytical tool to test how attractive MaaS (microtransit) would be to access the fixed guideway rail systems, FrontRunner, and TRAX. The research describes the technical modeling steps required to account for this new transit access mode.

Given that the mode share model within the Wasatch Front Travel Demand Model is being refactored in a project jointly supported by UDOT, UTA, and WFRC, this research has concluded by detailing the steps involved so that this work can be carried forward as part of that parallel project.

<table>
<tr><td colspan="2">17. Key Words
Mobility as a Service, Travel Demand Modeling, Transit Access</td><td colspan="2">18. Distribution Statement
Not restricted. Available through:
UDOT Research Division, 4501 South 2700 West, P.O. Box 148410, Salt Lake City, UT 84114-8410</td><td rowspan="2">23. Registrant's Seal
N/A</td></tr>
<tr><td>19. Security Classification
Unclassified</td><td>20. Security Classification
Unclassified</td><td>21. No. of Pages
120</td><td>22. Price
N/A</td></tr>
</table>

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This research project was conducted as follow-up research to another UTRAC project, The Impact of Shared Autonomous Vehicles on Vehicle Miles Travelled in Utah (UDOT Research Report No. UT-19.10). A key implementation strategy from that research was to investigate the modeling process that would be necessary to model Mobility as a Service (MaaS) as an access mode to transit.

This research investigated the recent research in this area and found several studies that had researched various aspects of MaaS, but mostly as a stand-alone mode (e.g. Uber/Lyft) as opposed to an access mode to transit. Simultaneous with this research is a pilot project initiated by the Utah Transit Authority (UTA) in partnership with Via to provide an on-demand ride to LRT and commuter rail stations within a prescribed area. This pilot is ongoing as of the writing of this report.

Two other ongoing or recent efforts are notable. First, UTA conducted an on-board survey concluding in November 2019 (prior to the commencing of the Via pilot) that identified zero trips using MaaS to access transit. Second, UDOT and WFRC have initiated a project to refactor the mode-split model of the Wasatch Front Travel Model. The findings and scripts created under this research (detailed in this report) can be used to support this latter effort to refactor the mode-split model.

It was concluded that a full development of the scripts necessary to represent Maas to Transit was beyond the resources of the UTRAC grant.

1. INTRODUCTION

1.1 Problem Statement

The past few years have witnessed an explosive growth in new transportation modes. These new modes are leveraging digital technologies to offer efficient and convenient mobility services. The new mobility services include a wide variety of on-demand services, also referred to as "Mobility as a Service" (MaaS), ranging from short-term rentals of cars, bikes, and scooters to the peer-to-peer provisioning of transportation services. Although the share of these new mobility services is still small, a significant increase in their popularity can be expected due to reduced waiting times, competitive fares, and the promise of a one-seat ride. Transportation Network Companies (TNCs) such as Uber and Lyft are among the most successful of these mobility services. In 2018, 4 billion TNC rides were estimated in the United States compared to 0.5 billion taxi rides. Uber operates in over 785 metropolitan areas worldwide and is estimated to have 110 million users across the world (Statista, 2020).

The success and growing popularity of the TNCs largely owe to a favorable regulatory environment, advancement of digital technologies, and innovative business models. Most TNCs are designed around smartphone platforms that allow travelers to request a ride at the push of a button, map estimated waiting time, and automatically pay by credit card. The platform estimates the travel cost based on travel distance, travel time, travel period, driver supply, and customer demand. Smartphone apps, treating drivers as independent contractors, and requiring drivers to buy and maintain their vehicles and personal insurance, help to cut costs when compared to conventional taxi service. As a result, TNC fees are generally competitive with fares offered by traditional taxi service. Recently, new shared-ride services offered by TNCs such as UberPool and LyftLine are making TNCs competitive with traditional public transportation modes. One study estimated similar ridership for TNCs and public transit at the end of 2018 (Schaller, 2018).

Considering additional benefits and convenience that these new on-demand mobility services offer, they have the promise to significantly alter the transportation landscape as we know it. To date, most long-range transportation plans in Utah have not accounted for new mobility services and their impact on public transportation. This highlights the need for research to shed light on the impact of MaaS on transit ridership. This research attempts to address this gap by investigating the impact of TNCs on public transportation use in Utah in the 2040 horizon year. This research follows on research commenced in 2017 (completed in 2019) which evaluated the impact of MaaS (termed "Shared Autonomous Vehicles" in that research) on vehicle miles traveled in Utah (UDOT Research Report No. UT-19.10).

1.2 Objectives

The objective of this UTRAC research is to follow up the 2017 study with an effort to effectively model MaaS as one of the first mile-last mile options for accessing transit. Transit access options incorporated into the current Wasatch Front model are park and ride, kiss and ride, and non-motorized). Addressing this important aspect of MaaS will give a more complete answer to the question of how this mode will affect travel demand in the future.

The current research extends the research cited above, completed in 2019, by extending the MaaS model as an access mode to transit. Some transit professionals refer to this mode as "microtransit". At the time of the writing of this report, the Utah Transit Authority (UTA) has initiated a microtransit pilot in partnership with Via, a TNC that focuses on connecting people with transit services. Restated then, the objective of this research is to develop a modeling approach to capture this type of microtransit service, so that it can be applied within the planning modeling frameworks used for estimating travel demand.

1.3 Outline of Report

This report has the following sections:

- Literature Review
- Approach to Modeling Mobility as a Service
- Conclusions
- Recommendations and Implementation

2. LITERATURE REVIEW

Deployment of smartphone technologies in recent years has provided the technology platform for shared mobility services. These new services have changed the transportation landscape by creating a new mode – Mobility as a Service, or MaaS – which competes with conventional transportation modes. MaaS as a viable mode is new and the research community has just begun to investigate the supply and demand characteristics of this mode. Microtransit, as a variation of MaaS, is newer still, and very little research investigation has yet to happen.

Several studies have investigated the impacts of users' characteristics, geographic context, and built environment factors on the adoption rate of MaaS. For example, Clewlow and Mishra (2017) conducted a comprehensive travel and residential survey in seven major U.S. cities with a representative sample of their urban and suburban populations. They found that the adoption rate of TNC services is approximately double among college-educated individuals compared to those without any college degree. Survey results revealed that 29% of Americans living in urban areas had used ridehailing services in comparison with 15% of those living in suburban areas.

Kooti et al. (2017) reported while younger TNC riders tend to take frequent, shorter rides, older ones are more likely to take infrequent, longer rides.

In another study, Alemi et al. (2018) studied the lifestyle of TNC users to identify the factors impacting the adoption rate of ridehailing services. They revealed that highly educated independent millennials who live in core urban areas without owning personal vehicles and without children have the highest adoption rate. They also reported positive correlation between adoption rate of these services and the urbanization level of the neighborhood.

Dias et al. (2017) estimated a bivariate Ordered Probit model to investigate the use of MaaS based on a survey dataset derived from a 2014-2015 Puget Sound Regional Travel Study. They found that users of such services are likely to be young, well-educated, higher income, employed, and residing in higher-density neighborhoods. The presence of children is found to reduce ridehailing and carsharing usage among low- and middle-income households. Results also revealed that households owning vehicles are less likely to use carsharing services and households residing in a high-density location are more likely to use both ridehailing and carsharing services compared to their counterparts residing in low-density areas. Several other studies also reported that ridehailing users are more likely to own fewer cars (Conway et al., 2018; Hampshire at al., 2017).

In more recent studies, Yu and Peng (2019) indicated that population density along with road and sidewalk densities significantly impact the demand for TNC services. Xie et al. (2019) developed a nested framework to model the behavior of a local on-demand mobility service in the Boston-Cambridge region considering impacts of subscription services, service access, menu options, and opt-out choices and their connections. The proposed framework is utilized to model the demand of the Tripod, an on-demand service which offers incentives for more energy-efficient travel options through a real-time travel menu (Azevedo et al., 2018). They reported higher Tripod market penetration in lower-income population segments. Moreover, Tripod's usage is found to be more associated with trips that have lower time constraints.

Firnkorn and Muller (2011) attempted to analyze the impacts of carsharing on mobility and car ownership using a survey conducted in Germany. They found that more than one-quarter of the respondents would be willing to forego a vehicle purchase if car2go, a carsharing service that allows users to take and leave vehicles at any point within city limits, was offered permanently.

Clewlow (2016) studied characteristics of carsharing service users using the 2010-2012 California Household Travel Survey in the San Francisco Bay Area. It found that carsharing service users residing in urban areas own significantly fewer vehicles than non-users. Carsharing service users living in suburban areas drive less than their non-user counterparts. They also reported that carsharing service users who own vehicles tend to own alternative-fueled vehicles, (e.g. hybrid, plug-in hybrid electric, and battery electric) showing more environmental concerns regarding their travel mode.

In summary, early adopters of on-demand mobility services are found to be well-educated individuals living in urban areas. These services are considerably more popular among young adults who are heavy users of smartphone technology and related apps.

Considering additional benefits that MaaS offers, it can draw significant share away from conventional transportation modes (Haghighi et al., 2019). On the other hand, MaaS has the potential to increase transit ridership by filling first/last mile gaps in transit use. Several studies attempted to unveil whether MaaS competes or complements transit use in urban areas. However, there is no consensus among researchers on the role that MaaS plays in serving public transportation. For example, while some studies reported that carsharing can complement the use of public transit (Firnkorn and Muller, 2011; Costain et al., 2012), another study showed that one-way carsharing can be a substitute for public transportation (Le vin et al., 2014).

Several studies conducted surveys to explore how ridehailing services compete with conventional transportation modes. As a part of the survey, respondents were asked what they would have used if ridehailing services were not available for their last trip. For example, Alemi et al. (2018) and Rayle et al. (2016) reported that 12-27% (Alemi) and 33% (Rayle) of ridehailing users would have traveled by transit if ridehailing services were not available. Clewlow and Mishra (2017) found that ridehailing services can substitute 6% and 3% of bus and light rail trips, respectively in major U.S. cities. Moreover, they reported that ridehailing services can play a complementary role for commuter rail and estimated a 3% increase in its use. Alemi et al. (2018) reported that approximately 10% of respondents increased their public transportation use due to improved accessibility provided by ridehailing services. However, for most respondents ridehailing substituted for the use of public transportation. Rayle et al. (2016) found that only 5% of ridehailing users in their dataset used ridehailing service to connect to public transportation. In another recent study in Utah, Haghighi et al. (2019) modeled Shared Autonomous Vehicles (SAVs) as a MaaS mode with a lower operating cost due to a driverless service. They reported that the introduction of a new MaaS mode shifts 12-17% of trips from auto (SOV and HOV) and 1.2% from non-motorized modes.

Henao (2017) analyzed the impacts of ridesourcing services on travel behaviors in Denver, Colorado. He became a driver for both Uber and Lyft to obtain real travel data from 416 rides (Lyft, UberX, LyftLine, and UberPool) and socio-demographics of 311 passengers. Survey results revealed that 22.2% of respondents would have traveled with public transportation and 12.2% would not have traveled if Lyft/Uber was not an option.

Some studies investigated agency- or city-level impacts of MaaS on transit ridership. For example, Sadowsky and Nelson (2017) investigated the impacts of Uber and Lyft on public transportation use in the US's largest urban areas using a discontinuity regression model. Monthly public transit data provided by the Federal Transit Authority was used for transit ridership analysis. Results revealed that the introduction of the first ridehailing company (Uber) served as a complement to public transit and increased transit ridership. However, the introduction of a second company (Lyft) served as a substitute to transit service, consequently decreasing public transit use in the study area. They also found that this substitution process strengthened over time. Moreover, they reported that the introduction of ridehailing services affected bus rider behavior differently than rail rider behavior in cities where both bus and rail public transportation systems are available. They found that a ridehailing service never became a substitute for bus users largely due to the lower fare cost of the bus. In another study, Smith (2015) found a strong correlation between Uber and public transit use, reporting that 25-40% of Uber drop-offs and pick-ups are located near public transit stations in several cities.

Young et al. (2019) investigated the degree to which ridehailing trips compete or complement public transit in Toronto. They attempted to unveil certain factors that make travelers choose ridehailing services instead of transit while transit is still a reasonably viable alternative. Trip characteristics are found to have more significant impact on competition than user attributes. They reported that 30.6% of ridehailing trips in the study sample have transit alternatives with approximately similar duration and can compete directly with public transport services. On the other hand, 26.9% of ridehailing trips are found to have poor transit alternatives and consequently are less competitive. They recommended imposing an additional tax on those ridehailing trips with viable transit alternatives to keep transit modes competitive.

Dias et al. (2018) analyzed one million trips by RideAustin, a local nonprofit ridesharing company launched in June 2016 in Austin. They found that individuals living in neighborhoods with poor transit access tend to use ridehailing services more frequently than transit service; however, there is a synergy between transit and ridehailing service usage in other areas. In another study, Barber and Burtch (2019) examined the impact that ridehailing services have had on the use of different transit modes in the United States. They indicated that, on average, while ridehailing services have significantly reduced the utilization of local bus services, they have increased utilization of commuter rail services.

Hall et al. (2018) investigated whether Uber is a substitute or a complement for public transit using a difference-in-differences approach across US metropolitan areas. They found that Uber is a complement for an average transit agency and it can increase public transit use. Moreover, they reported growth in transit use over time after Uber entry. However, while Uber can increase transit usage, it could still exacerbate congestion by increasing the number of trips by travelers and Uber drivers looking for a fare.

To complement public transportation use, transit agencies have been investing in park-and-ride lots, deployment of shuttle and feeder buses, and micromobility-sharing services to address the first-mile/last-mile gap that is an inconvenience for public transit users. Recently several transit agencies across the country have started to assess potential partnerships with Transportation Network Companies (TNCs) and other private service providers such as bike-sharing systems, car-sharing services that involve short-term car rentals, and private shuttle and bus services. These arrangements are so-called Public-Private Partnerships (P3), where a government agency contracts with a private partner to renovate, construct, operate, maintain, and/or manage a facility or system, in whole or in part. Under these arrangements, the agency may retain ownership of the public facility or system, but the private party generally invests its own capital to design and develop the properties. Typically, each partner shares in income resulting from the partnership.

From August 2016 to February 2017, the City of Centennial, Colorado implemented the "Go Centennial" project to provide first-mile/last-mile connections to transit through partnership with Lyft Line, Lyft's car-pooling service. Before partnering with Lyft, the city used to offer the call-n-ride system to fill first-mile/last-mile gaps in the area. Under the call-n-ride system, Centennial owned and operated several vans that had to be booked at least two hours in advance. Residents of Centennial had expressed concerns regarding booking trips during peak hours. Moreover, officials estimated that each ride costs $21 on average. The system used to charge $2.60 per ride but included free transfers to and from the light rail. To make Go Centennial competitive, the city made the new program entirely free but without a transfer pass for public transit. The city reported a successful pilot program implementation. Each Lyft Line ride cost $4.75 on average compared to the $21 under the previous dial-n-ride system to the city. Results revealed 11 riders per day for the new service compared to 50 riders per day for the original call-n-ride service. The smaller ridership might be explained by limited awareness about service availability (Blodgett et al., 2017).

Uber also started its first partnership with the City of Altamonte Springs, Florida in 2016 and more recently, launched partnerships with several transit agencies and transit apps including Masab, San Joaquin Regional Transit District (Stockton, California), TransLoc, and Moovit. Such partnerships allow travelers to make efficient trip planning using various transportation modes. Through these partnerships each traveler is able to input his destination in the Rider app and receive a personalized journey that uses an optimal combination of walking, transit, and Uber.

The Utah Transit Authority (UTA) has entered a partnership with Via, launched in November 2019, to provide a microtransit pilot service serving seven TRAX and FrontRunner stations within a 65-square mile area in south Salt Lake County (UTA on Demand by Via, 2019). This pilot is evaluating the effectiveness of this type of service as a complement to transit, as the Via service is designed to collect transit riders for the TRAX service (at the S. Jordan Parkway, Daybreak Parkway, Draper Town Center, Kimball's Lane, and Crescent View TRAX stations) and for the FrontRunner commuter rail service (at the Draper and South Jordan stations).

This pilot constitutes real-time research into the effectiveness of MaaS as a transit access mode. The user cost of a trip is $2.50 which can also be covered using a UTA pass. UTA has published evaluation reports providing summary data on Key Performance Indicators (https://www.rideuta.com/Services/UTA-on-Demand-by-Via/Microtransit-Evaluation-Reports). For the first quarter of 2020, the average cost per rider was $19.10, indicating that the price structuring of the UTA-Via pilot incorporates a cost subsidy for microtransit users. Weekday ridership averaged 316 per day during Q1 2020, which is evaluated as "approaching 6-month target, on track." The Q2 evaluation report cites the impacts from COVID-19 and shows the cost per rider increasing to $34.40 with an average weekday ridership of 169. The report notes that wheelchair customers have used the service at record rates.

Findings from the Literature Review Relevant to This Research

This review of recent research largely focuses on sociodemographic and urban form factors and their impact on the usage of MaaS and related forms of ridesharing. There is no recent published research focused on the key question which is the subject of this research, namely, how Mobility as a Service may serve as a first-mile/last-mile access mode to transit.

The current architecture of the Wasatch Front Travel Model may be able to partially address some of the factors discussed in the recent research, such as the impact of urban form and household composition on trip generation. Generally, however, the current model form is not able to address the impact of household income, household composition, and urban form on mode choice.

The research team will focus on making appropriate changes to the current Wasatch Front mode-choice model to accommodate MaaS within the transit branch in the nested structure of the model. These modifications will create a new transit access mode – MaaS – that will compete with the current transit access modes in the model – drive, kiss and ride, and walk/bike.

The most relevant information that can inform this project is the ongoing UTA pilot project with Via. This microtransit pilot provides the most relevant data for modeling this service for planning purposes. Results of the Via pilot may provide calibration data for the mode-choice model modifications proposed in this research.

3. APPROACH TO MODELING MOBILITY AS A SERVICE

This section summarizes various modifications to the current version of the model to accommodate MaaS as a mode for accessing transit. Proposed modifications to the mode choice model would create a new modal option – MaaS – for accessing transit on the origin end of a trip (e.g. home-to-transit station) or on the destination end of a trip (e.g. transit station to work).

3.1 Mode Choice Modifications

In the previous UTRAC research (UDOT Research Report No. UT-19.10), a new modal option was introduced to the motorized branch mode-choice model representing MaaS (Figure 1). To represent MaaS as an access mode to transit, or microtransit, a new modal option is introduced in the transit access sub-nest. Within the transit access sub-nest, a MaaS option is added to the options already programmed into the model-choice model: walk/bike, park & ride, kiss & ride. A simplifying assumption is that the MaaS access mode would be for fixed guideway modes only (LRT [shown], commuter rail [shown], BRT [not shown]).

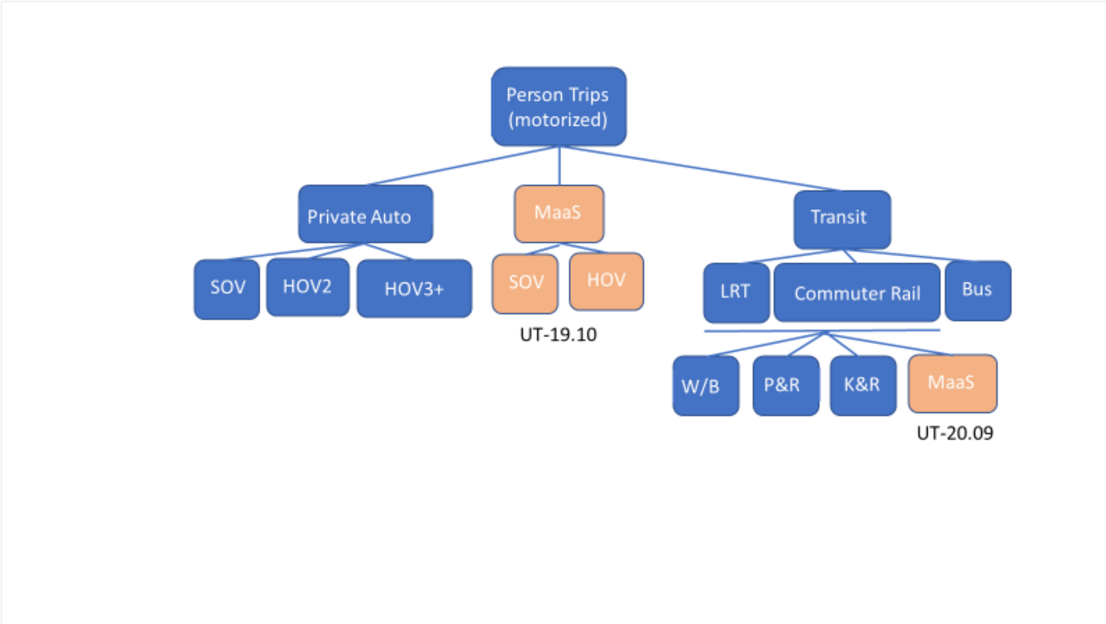

Figure 1 Schematic of Modal Options in the Wasatch Front Travel Model

The MaaS-to-Transit utility function would be very similar to the Park-&-Ride utility function, with additional time added to the trip for initial wait time (IWT). The MaaS-to-Transit utility function would also include a fare and possibly a new term reflecting the disutility of sharing a ride with strangers.

In 2019, UTA conducted an on-board ridership survey. This survey was completed before the UTA-Via microtransit pilot commenced and did not identify any trips using an Uber/Lyft type of service for accessing transit.

3.2 Steps to Implement Mode-Choice Modifications

To implement the new access mode, "Maas to Transit", several steps are necessary, each of which involves a custom script to be called by the main travel-model control script. These steps are depicted in Table 1.

Conduct for Origin and Destination End of Each Trip	Step 1	Create Drive Links for MaaS Access to Transit (Appendix A)
	Step 2	Create Transit Skim to Include MaaS Drive Access Link (Appendix B)
	Step 3	Include MaaS to Transit Nest for Selected Trip Types (Appendix C)
	Step 4	Calculate Mode Utilities for Selected Trip Types
	Step 5	Revise Mode Choice Script Block to Include MaaS to Transit
	Step 6	Update Mode Choice Working Matrices Script Block - Final Output
	Step 7	Update Model Run Control Script

Table 1 Steps to Implement MaaS to Transit in the Wasatch Front Travel Model

Step 1: Create a drive access link to each transit station where MaaS to Transit is a viable mode. A simplifying assumption is to limit drive access links to fixed guideway transit only and to enforce a distance constraint whereby TAZ centroids outside a specified radius are considered impractical for this mode. A second simplifying assumption is for the drive access links to be calculated as straight-line distances (centroid to centroid). The script for this step is provided in Appendix A.

Step 2: Create a transit skim to include MaaS-to-Transit Access Link. This script determines the distance and cost for accessing transit stations using MaaS and assigns those costs to the overall transit trip. The script for this step is provided in Appendix B.

Step 3: This script incorporates MaaS to Transit within the Transit Nest for Selected Trip Types. The current script is written for the Home-Based-Work and Home-Based-Other trip types, though other trip types might be considered for MaaS to Transit. (Appendix C).

Step 4: Calculate Modal Utilities for Selected Trip Types. This script calculates the mode-specific utilities, which would include the utilities for transit trips using MaaS as an access mode. Typical inputs to modal utilities are travel time and travel fare. For MaaS to Transit, an Initial Wait Time would be added to the utility equation. This script was not completed as part of the research.

Step 5: This script would calculate the revised mode choice to include MaaS to Transit if this sub-mode was determined to be the most competitive in the transit next. This script was not completed as part of the research.

Step 6: This is the final Mode-Choice output script which would need to include MaaS-to-Transit trips. This script was not completed as part of the research.

Step 7: The Cube-model control script would need to be updated to run the six new scripts.

All steps would need to address MaaS to Transit on both the origin and destination end of each trip.

4. CONCLUSIONS

Mobility as a Service (MaaS) is a new mode of travel that encompasses several concepts. As applied in this research, MaaS refers to an on-call service that would give patrons access to fixed-rail transit stations. Modeling Mobility as a Service as an access mode to transit involves seven steps, described in this research, which must be conducted for both the origin (accessing transit on the origin end of the trip) and the destination (accessing the final destination from the transit stop on the destination end of the trip).

Within the current framework of the Wasatch Front Travel Model, including MaaS to Transit involves custom scripting using the Cube-software scripting language. Of the seven steps described, three have been scripted and are provided as a product of this research.

5. RECOMMENDATIONS AND IMPLEMENTATION

5.1 Recommendations

It is recommended that the UDOT Planning Division coordinate with the technical modeling community, including WFRC and MAG, to identify the best way to incorporate the concepts, ideas, and solutions presented in this body of work. As of the writing of this report, WFRC and UDOT are engaged in a project to refactor the mode-split model such that emerging trends and modes, such as MaaS, may be more fully integrated.

The scripts provided in this research provide a solid foundation for including MaaS as an access mode to transit. The prior research, The Impact of Shared Autonomous Vehicles on VMT in Utah, is also available to inform the mode-split model refactoring project.

It is also recommended that the modeling community and the refactoring project review the performance of the UTA-Via pilot, which will provide essential validation data.

5.2 Implementation Plan

It is recommended that this report and the associated model scripts in Appendices A-C be made available to the project team that is currently refactoring the mode-split model of the Wasatch Front Travel Model.

Appendix A: Script for Creating MaaS-to-Transit Links

```
;System
    ;file to halt the model run if model crashes
    *(ECHO 'model crashed' > 04_Create_drive_access_links.txt)


;print time stamp
RUN PGM=MATRIX
    ZONES = 1
    ScriptStartTime = currenttime()
    PRINT FILE='@ParentDir@@ScenarioDir@_Log\RunTime.txt',
        APPEND=T,
        LIST='\n    Create Drive Access Links     ', formatdatetime(ScriptStartTime, 40, 0, 'yyyy-
mm-dd,  hh:nn:ss')
ENDRUN



;script specific variables
    ;MaaS mode
    ;Mode20
    Mode20_MaxDist     = 10      ;mile
    Mode20_MaxDist_SL  = 10
    Mode20_MaxDist_UT  = 10


    ;KNR mode
    ;Mode30
```

```
Mode30_MaxDist     = 1.5      ;mile
Mode30_MaxDist_SL  = 1.5
Mode30_MaxDist_UT  = 1.5

;bus modes
;Mode40
Mode40_NumCon      = 3      ;max number of drive access link
Mode40_EligiblePNR = '40,50,60,70,80,90'
Mode40_MaxTime     = 6      ;min
Mode40_MaxTime_SL  = 5
Mode40_MaxTime_UT  = 6


;BRT
;Mode50
Mode50_NumCon      = 2      ;max number of drive access link
Mode50_EligiblePNR = '40,50,60,70,80,90'
Mode50_MaxTime     = 8      ;min
Mode50_MaxTime_SL  = 5
Mode50_MaxTime_UT  = 8

;Mode90
Mode90_NumCon      = 2      ;max number of drive access link
Mode90_EligiblePNR = '40,50,60,70,80,90'
```

```
Mode90_MaxTime     = 6      ;min
Mode90_MaxTime_SL  = 3
Mode90_MaxTime_UT  = 6
;EXPRESS BUS
;Mode60
Mode60_NumCon      = 2      ;max number of drive access link
Mode60_EligiblePNR = '40,50,60,70,80,90'
Mode60_MaxTime     = 12     ;min
Mode60_MaxTime_SL  = 7
Mode60_MaxTime_UT  = 12


;rail modes
;Mode70
Mode70_NumCon      = 2      ;max number of drive access link
Mode70_EligiblePNR = '40,50,60,70,80,90'
Mode70_MaxTime     = 25     ;min
Mode70_MaxTime_SL  = 20
Mode70_MaxTime_UT  = 25
PNR_notfor_DAWE    =20125 ; DA and WE residents do not want to drive to U and PNR and use Trax
;CRT
;Mode80
Mode80_NumCon      = 2      ;max number of drive access link
Mode80_EligiblePNR = '40,50,60,70,80,90'
Mode80_MaxTime     = 30     ;min
```

```
   Mode80_MaxTime_SL  = 25

   Mode80_MaxTime_UT  = 35
```

```
;CREATE KNR DRIVE ACCESS LINKS
===================================================================
======================================

;calculate MODE30 drive access links

RUN PGM=MATRIX  MSG='Mode Choice 4: calculate KNR (mode 30) drive access links'

   FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf',
AUTOARRAY=ALLFIELDS

   FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsAllModes.dbf',
AUTOARRAY=ALLFIELDS


   FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode30.NTL'


   ;parameters

   ZONES = 1


   PRINT PRINTO=1, LIST=';;<<PT>>;; \n'


   ;loop through TAZ nodes
```

```
LOOP TAZ_Node=1,@UsedZones@
   procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
   PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'

   ;calculate TAZ X & Y coordinates, TAZ area in sqare miles
   TAZ_X  = DBA.1.X[TAZ_Node]
   TAZ_Y  = DBA.1.Y[TAZ_Node]
   County = DBA.1.COUNTY[TAZ_Node]

   ;calculate max drive distance (min)
   if (County=@SL_ID@)
      maxdrive = @Mode30_MaxDist_SL@
   elseif (County=@UT_ID@)
      maxdrive = @Mode30_MaxDist_UT@
   else
      maxdrive = @Mode30_MaxDist@
   endif

   ;loop through stop nodes
   LOOP StopRec=1,DBI.2.NUMRECORDS
      ;calculate stop node number and X & Y coordinates
      Stop_Node = DBA.2.N[StopRec]
      Stop_X    = DBA.2.X[StopRec]
      Stop_Y    = DBA.2.Y[StopRec]
```

```
      ;calculate distance & from TAZ to stop node in miles
      xydist = SQRT( (TAZ_X-Stop_X)^2 + (TAZ_Y-Stop_Y)^2 ) / 1609.344    ;convert meters to miles


      ;print walk access support link if <= 1 max drive distance
      if (xydist<=maxdrive)
        ;print out link as NT leg
        PRINT PRINTO=1,
          LIST='NT LEG=', TAZ_Node(6.0),' -', Stop_Node(6.0), ',  MODE=30,  COST=', xydist/25*60(6.2), ',  DIST=', xydist(6.2), ',  ONEWAY=T,  SPEED=25.0'
      endif


    ENDLOOP  ;loop through stop records


  ENDLOOP  ;loop through TAZ centroids


ENDRUN






;CREATE LOCAL BUS DRIVE ACCESS LINKS ============================================================================================================


;calculate MODE40 drive access links
RUN PGM=MATRIX  MSG='Mode Choice 4: calculate local bus (mode 40) drive access links'
```

```
   FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_PNR_nodes.dbf',
AUTOARRAY=ALLFIELDS

   FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf',
AUTOARRAY=ALLFIELDS

   FILEI DBI[3] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode4.dbf',
AUTOARRAY=ALLFIELDS


   FILEI MATI[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_Pk.mtx'


   FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode40.NTL'



   ;parameters

   ZONES = 1



   PRINT PRINTO=1, LIST=';;<<PT>>;; \n'



   ;define arrays

   ARRAY STOPS_NODE     = 99999,


      Closest_PNR    = 1000,

      Closest_Time   = 1000,

      Closest_Dist   = 1000,
```

```
      Closest_XYDist = 1000


    ;assign stop node array based on node number as index (reduce need to loop in script)
    LOOP recnum=1,DBI.3.NUMRECORDS
       idx           = DBA.3.N[recnum]
       STOPS_NODE[idx] = recnum
    ENDLOOP


    ;calculate drive access links
    ;loop through TAZ
    LOOP TAZ_Node=1,@Usedzones@
       procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
       PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'


       TAZ_X      = DBA.2.X[TAZ_Node]
       TAZ_Y      = DBA.2.Y[TAZ_Node]
       TAZ_COUNTY = DBA.2.COUNTY[TAZ_Node]


       ;initialize PNR variables
       cnt_EligiblePNR = 0
       SET VAL=9999, VARS=Closest_Time    ;sets all elements in array to 9999 so 0 values in
array are sorted to back in the array


       ;populate closest PNR arrays
       LOOP PNR_recnum=1,DBI.1.NUMRECORDS
```

```
;assign PNR variables
PNR_Node   = DBA.1.N[PNR_recnum]
PNR_X      = DBA.1.X[PNR_recnum]
PNR_Y      = DBA.1.Y[PNR_recnum]
PNR_Code   = DBA.1.PNR[PNR_recnum]
PNR_TAZID  = DBA.1.TAZID[PNR_recnum]

;check for eligible PNR node
if (PNR_Code=@Mode40_EligiblePNR@ & STOPS_NODE[PNR_Node]>0)
   ;count eligible PNRs
   cnt_EligiblePNR = cnt_EligiblePNR + 1

   ;calculate distance from TAZ to PNR node (not used - calculated for comparison to over-net dist)
   xydist = SQRT( (TAZ_X-PNR_X)^2 + (TAZ_Y-PNR_Y)^2 ) / 1609.344    ;convert meters to miles

   ;lookup AM time and distance skim values based on PNR's TAZID
   OverNetDist = MATVAL(1, 11, TAZ_Node, PNR_TAZID, 0)   ;MATVAL(file#, matrix#, I, J, ReturnCode if error)
   IVT         = MATVAL(1, 5,  TAZ_Node, PNR_TAZID, 0)
   OVT         = MATVAL(1, 1,  TAZ_Node, PNR_TAZID, 0)

   ;calculate total time = in-vehicle time + out-of-vehicle time
   TotalTime   = IVT ;+ OVT
```

```
            ;check for zero values
            if (IVT=0 | OverNetDist=0)
              TotalTime   = 9999
              OverNetDist = 999
            endif

            ;assign initial array element
            Closest_PNR[PNR_recnum]    = PNR_Node
            Closest_Time[PNR_recnum]   = TotalTime
            Closest_Dist[PNR_recnum]   = OverNetDist
            Closest_XYDist[PNR_recnum] = xydist

          endif  ;check for eligble PNR node

        ENDLOOP  ;populate closest PNR arrays

        ;sort ascending based on shortest time, then shortest over-net distance, then shortest xy
    distance, then PNR node number
        SORT ARRAY='+Closest_Time','+Closest_Dist','+Closest_XYDist','+Closest_PNR'

        ;check array
        ;if (TAZ_Node=11)
        ;    LOOP chk_array=1,cnt_EligiblePNR
        ;       if (chk_array=1)
        ;           PRINT CSV=T,
    FILE='@ParentDir@@ScenarioDir@Temp\4_ModeChoice\_check_closestPNRsorting.csv',
```

```
;               LIST='    TAZ', '   INDEX', '     PNR', '    TIME', '    DIST', '  XYDIST'

;       endif

;

;       PRINT CSV=T, FORM=8.0,
FILE='@ParentDir@@ScenarioDir@Temp\4_ModeChoice\_check_closestPNRsorting.csv',

;           LIST=TAZ_Node, chk_array, Closest_PNR[chk_array],
Closest_Time[chk_array](8.2), Closest_Dist[chk_array](8.2), Closest_XYDist[chk_array](8.2)

;    ENDLOOP

;endif


;print drive access links

LOOP Conct_num=1,@Mode40_NumCon@

   ;assign closest nth eligible PNR node and calculate avg over the network speed

   PNR_Node = Closest_PNR[Conct_num]

   PNR_Time = Closest_Time[Conct_num]

   PNR_Dist = Closest_Dist[Conct_num]


   OverNetSpeed = 60 * PNR_Dist / PNR_Time      ;in mph



   ;add dummy link in case no transit mode in scenario (need at least 1 to keep from
crashing)

   if (print_dummy_40=0)

      PRINT PRINTO=1, LIST='NT LEG=1-1,  MODE=40,  COST=2.40,  DIST=1.00,
ONEWAY=F,  SPEED=25.0'

      print_dummy_40=1

   endif
```

```
;print drive access links if less than MaxTime
if (TAZ_COUNTY=@SL_ID@)
   ;use SL county MaxTime
   if (PNR_Time<=@Mode40_MaxTime_SL@)
      ;print out link as NT leg
      PRINT PRINTO=1,
         LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=40,  COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
   endif

elseif (TAZ_COUNTY=@UT_ID@)
   ;use UT county MaxTime
   if (PNR_Time<=@Mode40_MaxTime_UT@)
      ;print out link as NT leg
      PRINT PRINTO=1,
         LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=40,  COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
   endif

else
   ;use general MaxTime
   if (PNR_Time<=@Mode40_MaxTime@)
      ;print out link as NT leg
      PRINT PRINTO=1,
```

```
            LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=40,  COST=', 
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

        endif

      endif  ;print drive access links if less than MaxTime


    ENDLOOP  ;print drive access links


  ENDLOOP  ;loop through TAZ


ENDRUN





;CREATE BRT DRIVE ACCESS LINKS
==============================================================================
============================================


;calculate MODE50 drive access links

RUN PGM=MATRIX  MSG='Mode Choice 4: calculate BRT (mode 50) drive access links'

  FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_PNR_nodes.dbf', 
AUTOARRAY=ALLFIELDS

  FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf', 
AUTOARRAY=ALLFIELDS

  FILEI DBI[3] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode5.dbf', 
AUTOARRAY=ALLFIELDS
```

```
FILEI MATI[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_Pk.mtx'


FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode50.NTL'


;parameters
ZONES = 1


PRINT PRINTO=1, LIST=';;<<PT>>;; \n'



;define arrays
ARRAY STOPS_NODE     = 99999,


      Closest_PNR    = 1000,
      Closest_Time   = 1000,
      Closest_Dist   = 1000,
      Closest_XYDist = 1000



;assign stop node array based on node number as index (reduce need to loop in script)
LOOP recnum=1,DBI.3.NUMRECORDS
    idx            = DBA.3.N[recnum]
    STOPS_NODE[idx] = recnum
ENDLOOP
```

```
;calculate drive access links
;loop through TAZ
LOOP TAZ_Node=1,@Usedzones@
  procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
  PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'

  TAZ_X      = DBA.2.X[TAZ_Node]
  TAZ_Y      = DBA.2.Y[TAZ_Node]
  TAZ_COUNTY = DBA.2.COUNTY[TAZ_Node]

  ;initialize PNR variables
  cnt_EligiblePNR = 0
  SET VAL=9999, VARS=Closest_Time    ;sets all elements in array to 9999 so 0 values in
array are sorted to back in the array

  ;populate closest PNR arrays
  LOOP PNR_recnum=1,DBI.1.NUMRECORDS
    ;assign PNR variables
    PNR_Node   = DBA.1.N[PNR_recnum]
    PNR_X      = DBA.1.X[PNR_recnum]
    PNR_Y      = DBA.1.Y[PNR_recnum]
    PNR_Code   = DBA.1.PNR[PNR_recnum]
    PNR_TAZID  = DBA.1.TAZID[PNR_recnum]
```

```
;check for eligible PNR node
if (PNR_Code=@Mode50_EligiblePNR@ & STOPS_NODE[PNR_Node]>0)
   ;count eligible PNRs
   cnt_EligiblePNR = cnt_EligiblePNR + 1

   ;calculate distance from TAZ to PNR node
   xydist = SQRT( (TAZ_X-PNR_X)^2 + (TAZ_Y-PNR_Y)^2 ) / 1609.344    ;convert
meters to miles

   ;lookup AM time and distance skim values based on PNR's TAZID
   OverNetDist = MATVAL(1, 11, TAZ_Node, PNR_TAZID, 0)   ;MATVAL(file#,
matrix#, I, J, ReturnCode if error)
   IVT         = MATVAL(1, 5,  TAZ_Node, PNR_TAZID, 0)
   OVT         = MATVAL(1, 1,  TAZ_Node, PNR_TAZID, 0)

   ;calculate total time = in-vehicle time + out-of-vehicle time
   TotalTime   = IVT ;+ OVT

   ;check for zero values
   if (IVT=0 | OverNetDist=0)
      TotalTime   = 9999
      OverNetDist = 999
   endif

   ;assign initial array element
```

```
            Closest_PNR[PNR_recnum]    = PNR_Node
            Closest_Time[PNR_recnum]   = TotalTime
            Closest_Dist[PNR_recnum]   = OverNetDist
            Closest_XYDist[PNR_recnum] = xydist

         endif  ;check for eligble PNR node

      ENDLOOP  ;populate closest PNR arrays

      ;sort ascending based on shortest time, then shortest over-net distance, then shortest xy distance, then PNR node number
      SORT ARRAY='+Closest_Time','+Closest_Dist','+Closest_XYDist','+Closest_PNR'

      ;print drive access links
      LOOP Conct_num=1,@Mode50_NumCon@
         ;assign closest nth eligible PNR node and calculate avg over the network speed
         PNR_Node = Closest_PNR[Conct_num]
         PNR_Time = Closest_Time[Conct_num]
         PNR_Dist = Closest_Dist[Conct_num]

         OverNetSpeed = 60 * PNR_Dist / PNR_Time      ;in mph

         ;add dummy link in case no transit mode in scenario (need at least 1 to keep from crashing)
         if (print_dummy_50=0)
```

```
        PRINT PRINTO=1, LIST='NT LEG=1-1,  MODE=50,  COST=2.40,  DIST=1.00,
ONEWAY=F,  SPEED=25.0'

        print_dummy_50=1

    endif



    ;print drive access links if less than MaxTime

    if (TAZ_COUNTY=@SL_ID@)

        ;use SL county MaxTime

        if (PNR_Time<=@Mode50_MaxTime_SL@)

            ;print out link as NT leg

            PRINT PRINTO=1,

                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=50,  COST=',
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

        endif



    elseif (TAZ_COUNTY=@UT_ID@)

        ;use UT county MaxTime

        if (PNR_Time<=@Mode50_MaxTime_UT@)

            ;print out link as NT leg

            PRINT PRINTO=1,

                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=50,  COST=',
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

        endif



    else

        ;use general MaxTime
```

```
        if (PNR_Time<=@Mode50_MaxTime@)
          ;print out link as NT leg
          PRINT PRINTO=1,
            LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=50,  COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
        endif
      endif  ;print drive access links if less than MaxTime

    ENDLOOP  ;print drive access links

  ENDLOOP  ;loop through TAZ

ENDRUN

;CREATE BRT 9 DRIVE ACCESS LINKS
=====================================================================================================================

;calculate MODE90 drive access links
RUN PGM=MATRIX  MSG='Mode Choice 4: calculate BRT 9 (mode 90) drive access links'
  FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_PNR_nodes.dbf', AUTOARRAY=ALLFIELDS
  FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf', AUTOARRAY=ALLFIELDS
  FILEI DBI[3] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode9.dbf', AUTOARRAY=ALLFIELDS
```

```
FILEI MATI[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_Pk.mtx'


FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode90.NTL'


;parameters
ZONES = 1
PRINT PRINTO=1, LIST=';;<<PT>>;; \n'


;define arrays
ARRAY STOPS_NODE     = 99999,


      Closest_PNR    = 1000,
      Closest_Time   = 1000,
      Closest_Dist   = 1000,
      Closest_XYDist = 1000



;assign stop node array based on node number as index (reduce need to loop in script)
LOOP recnum=1,DBI.3.NUMRECORDS
    idx            = DBA.3.N[recnum]
    STOPS_NODE[idx] = recnum
ENDLOOP
```

```
;calculate drive access links
;loop through TAZ
LOOP TAZ_Node=1,@Usedzones@
    procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
    PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'

    TAZ_X      = DBA.2.X[TAZ_Node]
    TAZ_Y      = DBA.2.Y[TAZ_Node]
    TAZ_COUNTY = DBA.2.COUNTY[TAZ_Node]

    ;initialize PNR variables
    cnt_EligiblePNR = 0
    SET VAL=9999, VARS=Closest_Time    ;sets all elements in array to 9999 so 0 values in
array are sorted to back in the array

    ;populate closest PNR arrays
    LOOP PNR_recnum=1,DBI.1.NUMRECORDS
        ;assign PNR variables
        PNR_Node   = DBA.1.N[PNR_recnum]
        PNR_X      = DBA.1.X[PNR_recnum]
        PNR_Y      = DBA.1.Y[PNR_recnum]
        PNR_Code   = DBA.1.PNR[PNR_recnum]
        PNR_TAZID  = DBA.1.TAZID[PNR_recnum]

        ;check for eligible PNR node
```

```
if (PNR_Code=@Mode90_EligiblePNR@ & STOPS_NODE[PNR_Node]>0)
   ;count eligible PNRs
   cnt_EligiblePNR = cnt_EligiblePNR + 1

   ;calculate distance from TAZ to PNR node
   xydist = SQRT( (TAZ_X-PNR_X)^2 + (TAZ_Y-PNR_Y)^2 ) / 1609.344    ;convert meters to miles

   ;lookup AM time and distance skim values based on PNR's TAZID
   OverNetDist = MATVAL(1, 11, TAZ_Node, PNR_TAZID, 0)   ;MATVAL(file#, matrix#, I, J, ReturnCode if error)
   IVT         = MATVAL(1, 5,  TAZ_Node, PNR_TAZID, 0)
   OVT         = MATVAL(1, 1,  TAZ_Node, PNR_TAZID, 0)

   ;calculate total time = in-vehicle time + out-of-vehicle time
   TotalTime   = IVT ;+ OVT

   ;check for zero values
   if (IVT=0 | OverNetDist=0)
     TotalTime   = 9999
     OverNetDist = 999
   endif

   ;assign initial array element
   Closest_PNR[PNR_recnum]    = PNR_Node
   Closest_Time[PNR_recnum]   = TotalTime
```

```
Closest_Dist[PNR_recnum]   = OverNetDist
Closest_XYDist[PNR_recnum] = xydist

endif  ;check for eligble PNR node

ENDLOOP  ;populate closest PNR arrays

;sort ascending based on shortest time, then shortest over-net distance, then shortest xy distance, then PNR node number
SORT ARRAY='+Closest_Time','+Closest_Dist','+Closest_XYDist','+Closest_PNR'

;print drive access links
LOOP Conct_num=1,@Mode90_NumCon@
;assign closest nth eligible PNR node and calculate avg over the network speed
PNR_Node = Closest_PNR[Conct_num]
PNR_Time = Closest_Time[Conct_num]
PNR_Dist = Closest_Dist[Conct_num]

OverNetSpeed = 60 * PNR_Dist / PNR_Time      ;in mph

;add dummy link in case no transit mode in scenario (need at least 1 to keep from crashing)
if (print_dummy_90=0)
PRINT PRINTO=1, LIST='NT LEG=1-1,  MODE=90,  COST=2.40,  DIST=1.00, ONEWAY=F,  SPEED=25.0'
print_dummy_90=1
```

```
endif



;print drive access links if less than MaxTime
if (TAZ_COUNTY=@SL_ID@)
   ;use SL county MaxTime
   if (PNR_Time<=@Mode90_MaxTime_SL@)
      ;print out link as NT leg
      PRINT PRINTO=1,
         LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=90,  COST=',
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
   endif

elseif (TAZ_COUNTY=@UT_ID@)
   ;use UT county MaxTime
   if (PNR_Time<=@Mode90_MaxTime_UT@)
      ;print out link as NT leg
      PRINT PRINTO=1,
         LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=90,  COST=',
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
   endif

else
   ;use general MaxTime
   if (PNR_Time<=@Mode90_MaxTime@)
      ;print out link as NT leg
```

```
PRINT PRINTO=1,

    LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=90,  COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

        endif

      endif  ;print drive access links if less than MaxTime


    ENDLOOP  ;print drive access links


  ENDLOOP  ;loop through TAZ


ENDRUN



;CREATE EXPRESS BUS DRIVE ACCESS LINKS
======================================================================================================================


;calculate MODE60 drive access links

RUN PGM=MATRIX  MSG='Mode Choice 4: calculate express bus (mode 60) drive access links'

  FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_PNR_nodes.dbf', AUTOARRAY=ALLFIELDS

  FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf', AUTOARRAY=ALLFIELDS

  FILEI DBI[3] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode6.dbf', AUTOARRAY=ALLFIELDS
```

```
    FILEI MATI[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_Pk.mtx'


    FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode60.NTL'


    ;parameters
    ZONES = 1


    PRINT PRINTO=1, LIST=';;<<PT>>;; \n'


    ;define arrays
    ARRAY STOPS_NODE     = 99999,


          Closest_PNR    = 1000,
          Closest_Time   = 1000,
          Closest_Dist   = 1000,
          Closest_XYDist = 1000



    ;assign stop node array based on node number as index (reduce need to loop in script)
    LOOP recnum=1,DBI.3.NUMRECORDS
      idx            = DBA.3.N[recnum]
      STOPS_NODE[idx] = recnum
    ENDLOOP
```

```
;calculate drive access links
;loop through TAZ
LOOP TAZ_Node=1,@Usedzones@
  procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
  PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'

  TAZ_X      = DBA.2.X[TAZ_Node]
  TAZ_Y      = DBA.2.Y[TAZ_Node]
  TAZ_COUNTY = DBA.2.COUNTY[TAZ_Node]

  ;initialize PNR variables
  cnt_EligiblePNR = 0
  SET VAL=9999, VARS=Closest_Time    ;sets all elements in array to 9999 so 0 values in
array are sorted to back in the array

  ;populate closest PNR arrays
  LOOP PNR_recnum=1,DBI.1.NUMRECORDS
    ;assign PNR variables
    PNR_Node   = DBA.1.N[PNR_recnum]
    PNR_X      = DBA.1.X[PNR_recnum]
    PNR_Y      = DBA.1.Y[PNR_recnum]
    PNR_Code   = DBA.1.PNR[PNR_recnum]
    PNR_TAZID  = DBA.1.TAZID[PNR_recnum]
```

```
;check for eligible PNR node
if (PNR_Code=@Mode60_EligiblePNR@ & STOPS_NODE[PNR_Node]>0)
    ;count eligible PNRs
    cnt_EligiblePNR = cnt_EligiblePNR + 1

    ;calculate distance from TAZ to PNR node
    xydist = SQRT( (TAZ_X-PNR_X)^2 + (TAZ_Y-PNR_Y)^2 ) / 1609.344    ;convert
meters to miles

    ;lookup AM time and distance skim values based on PNR's TAZID
    OverNetDist = MATVAL(1, 11, TAZ_Node, PNR_TAZID, 0)   ;MATVAL(file#,
matrix#, I, J, ReturnCode if error)
    IVT         = MATVAL(1, 5,  TAZ_Node, PNR_TAZID, 0)
    OVT         = MATVAL(1, 1,  TAZ_Node, PNR_TAZID, 0)

    ;calculate total time = in-vehicle time + out-of-vehicle time
    TotalTime   = IVT ;+ OVT

    ;check for zero values
    if (IVT=0 | OverNetDist=0)
       TotalTime   = 9999
       OverNetDist = 999
    endif

    ;assign initial array element
    Closest_PNR[PNR_recnum]    = PNR_Node
```

```
            Closest_Time[PNR_recnum]   = TotalTime
            Closest_Dist[PNR_recnum]   = OverNetDist
            Closest_XYDist[PNR_recnum] = xydist

        endif  ;check for eligble PNR node

    ENDLOOP  ;populate closest PNR arrays

    ;sort ascending based on shortest time, then shortest over-net distance, then shortest xy distance, then PNR node number
    SORT ARRAY='+Closest_Time','+Closest_Dist','+Closest_XYDist','+Closest_PNR'

    ;print drive access links
    LOOP Conct_num=1,@Mode60_NumCon@
        ;assign closest nth eligible PNR node and calculate avg over the network speed
        PNR_Node = Closest_PNR[Conct_num]
        PNR_Time = Closest_Time[Conct_num]
        PNR_Dist = Closest_Dist[Conct_num]

        OverNetSpeed = 60 * PNR_Dist / PNR_Time      ;in mph



        ;add dummy link in case no transit mode in scenario (need at least 1 to keep from crashing)
        if (print_dummy_60=0)
```

```
        PRINT PRINTO=1, LIST='NT LEG=1-1,  MODE=60,  COST=2.40,  DIST=1.00,
ONEWAY=F,  SPEED=25.0'

        print_dummy_60=1

    endif




    ;print drive access links if less than MaxTime

    if (TAZ_COUNTY=@SL_ID@)

        ;use SL county MaxTime

        if (PNR_Time<=@Mode60_MaxTime_SL@)

            ;print out link as NT leg

            PRINT PRINTO=1,

                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=60,  COST=',
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

        endif


    elseif (TAZ_COUNTY=@UT_ID@)

        ;use UT county MaxTime

        if (PNR_Time<=@Mode60_MaxTime_UT@)

            ;print out link as NT leg

            PRINT PRINTO=1,

                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=60,  COST=',
PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

        endif


    else
```

```
            ;use general MaxTime
            if (PNR_Time<=@Mode60_MaxTime@)
              ;print out link as NT leg
              PRINT PRINTO=1,
                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=60,  COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
            endif
          endif  ;print drive access links if less than MaxTime

        ENDLOOP  ;print drive access links

    ENDLOOP  ;loop through TAZ

ENDRUN

;CREATE LRT DRIVE ACCESS LINKS ===================================================================================================

;calculate MODE70 drive access links
RUN PGM=MATRIX  MSG='Mode Choice 4: calculate light rail (mode 70) drive access links'
    FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_PNR_nodes.dbf', AUTOARRAY=ALLFIELDS
```

```
    FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf',
AUTOARRAY=ALLFIELDS

    FILEI DBI[3] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode7.dbf',
AUTOARRAY=ALLFIELDS


    FILEI MATI[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_Pk.mtx'


    FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode70.NTL'



    ;parameters

    ZONES = 1



    PRINT PRINTO=1, LIST=';;<<PT>>;; \n'



    ;define numeric arrays

    ARRAY STOPS_NODE     = 99999,


        Closest_PNR    = 1000,

        Closest_Time   = 1000,

        Closest_Dist   = 1000,

        Closest_XYDist = 1000
```

```
;define character arrays
ARRAY TYPE=C55, RailLine_Name  = 99999,
      TYPE=C55, UniqueRailLine = 500



;assign stop node array based on node number as index (reduce need to loop in script)
;and identify number of unique rail lines
LOOP recnum=1,DBI.3.NUMRECORDS
   idx              = DBA.3.N[recnum]
   STOPS_NODE[idx]    = recnum
   RailLine_Name[idx] = DBA.3.ROUTE[recnum]

   ;identify unique rail lines
   Unique_Name = DBA.3.ROUTE[recnum]
   inlist      = 0

   if (recnum=1)
      cnt_UniqueRail    = 1
      UniqueRailLine[1] = Unique_Name

      ;print unique lines to check file
      PRINT CSV=T, FORM=8.0,
FILE='@ParentDir@@ScenarioDir@Temp\4_ModeChoice\_check_unique_Lines_Mode7.csv',
         LIST=cnt_UniqueRail, UniqueRailLine[1]
```

```
    else
      LOOP chk_unq=1,cnt_UniqueRail
        if (Unique_Name=UniqueRailLine[chk_unq])
          ;route name already in list
          inlist = 1
          BREAK
        endif
      ENDLOOP

      ;add unique rail iine to array
      if (inlist<>1)
        cnt_UniqueRail              = cnt_UniqueRail + 1
        UniqueRailLine[cnt_UniqueRail] = Unique_Name

        ;print unique lines to check file
        PRINT CSV=T, FORM=8.0,
FILE='@ParentDir@@ScenarioDir@Temp\4_ModeChoice\_check_unique_Lines_Mode7.csv',
          LIST=cnt_UniqueRail, UniqueRailLine[cnt_UniqueRail]
      endif
    endif
  ENDLOOP



  ;calculate drive access links
  ;loop through TAZ
```

```
LOOP TAZ_Node=1,@Usedzones@

  procrec = ROUND(TAZ_Node / @UsedZones@ * 100)

  PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'


  TAZ_X      = DBA.2.X[TAZ_Node]

  TAZ_Y      = DBA.2.Y[TAZ_Node]

  TAZ_COUNTY = DBA.2.COUNTY[TAZ_Node]


  ;loop through unique rail lines

  LOOP lp_Rail=1,cnt_UniqueRail

    ;assign name of unique rail line for this loop iteration

    Check_RailLine = UniqueRailLine[lp_Rail]


    ;initialize PNR variables

    cnt_EligiblePNR = 0

    SET VAL=9999, VARS=Closest_Time    ;sets all elements in array to 9999 so 0 values
in array are sorted to back in the array


    ;populate closest PNR arrays

    LOOP PNR_recnum=1,DBI.1.NUMRECORDS

      ;assign PNR variables

      PNR_Node   = DBA.1.N[PNR_recnum]

      PNR_X      = DBA.1.X[PNR_recnum]

      PNR_Y      = DBA.1.Y[PNR_recnum]

      PNR_Code   = DBA.1.PNR[PNR_recnum]
```

```
PNR_TAZID  = DBA.1.TAZID[PNR_recnum]


;check for eligible PNR node

if (PNR_Code=@Mode70_EligiblePNR@ & STOPS_NODE[PNR_Node]>0 &
RailLine_Name[PNR_Node]=Check_RailLine)

  ;count eligible PNRs

  cnt_EligiblePNR = cnt_EligiblePNR + 1


  ;calculate distance from TAZ to PNR node

  xydist = SQRT( (TAZ_X-PNR_X)^2 + (TAZ_Y-PNR_Y)^2 ) / 1609.344    ;convert
meters to miles


  ;lookup AM time and distance skim values based on PNR's TAZID

  OverNetDist = MATVAL(1, 11, TAZ_Node, PNR_TAZID, 0)   ;MATVAL(file#,
matrix#, I, J, ReturnCode if error)

  IVT         = MATVAL(1, 5,  TAZ_Node, PNR_TAZID, 0)

  OVT          = MATVAL(1, 1,  TAZ_Node, PNR_TAZID, 0)


  ;calculate total time = in-vehicle time + out-of-vehicle time

  TotalTime   = IVT ;+ OVT


  ;check for zero values

  if (IVT=0 | OverNetDist=0)

    TotalTime   = 9999

    OverNetDist = 999

  endif
```

```
        ;assign initial array element
        Closest_PNR[PNR_recnum]    = PNR_Node
        Closest_Time[PNR_recnum]   = TotalTime
        Closest_Dist[PNR_recnum]   = OverNetDist
        Closest_XYDist[PNR_recnum] = xydist

      endif  ;check for eligble PNR node

    ENDLOOP  ;populate closest PNR arrays
```

;sort ascending based on shortest time, then shortest over-net distance, then shortest xy distance, then PNR node number

```
    SORT ARRAY='+Closest_Time','+Closest_Dist','+Closest_XYDist','+Closest_PNR'

    ;print drive access links
    LOOP Conct_num=1,@Mode70_NumCon@
      ;assign closest nth eligible PNR node and calculate avg over the network speed
      PNR_Node = Closest_PNR[Conct_num]
      PNR_Time = Closest_Time[Conct_num]
      PNR_Dist = Closest_Dist[Conct_num]

      OverNetSpeed = 60 * PNR_Dist / PNR_Time      ;in mph
```

```
;add dummy link in case no transit mode in scenario (need at least 1 to keep from crashing)

if (print_dummy_70=0)

   PRINT PRINTO=1, LIST='NT LEG=1-1,  MODE=70,  COST=2.40,  DIST=1.00, ONEWAY=F,  SPEED=25.0'

   print_dummy_70=1

endif



;print drive access links if less than MaxTime

if (TAZ_COUNTY=@SL_ID@)

   ;use SL county MaxTime

   if (PNR_Time<=@Mode70_MaxTime_SL@)

      ;print out link as NT leg

      PRINT PRINTO=1,

         LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=70, COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)

   endif


elseif (TAZ_COUNTY=@UT_ID@)

   ;use UT county MaxTime

   if (PNR_Time<=@Mode70_MaxTime_UT@)

      ;print out link as NT leg

      PRINT PRINTO=1,
```

```
                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=70,
COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=',
OverNetSpeed(6.2)

            endif



          else

            ;use general MaxTime

            if (PNR_Time<=@Mode70_MaxTime@ &&
PNR_Node<>@PNR_notfor_DAWE@)

              ;print out link as NT leg

              PRINT PRINTO=1,

                LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=70,
COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=',
OverNetSpeed(6.2)

            endif

          endif  ;print drive access links if less than MaxTime



        ENDLOOP  ;print drive access links



      ENDLOOP ;loop through unique rail lines



    ENDLOOP  ;loop through TAZ



ENDRUN
```

;CREATE CRT DRIVE ACCESS LINKS
===

;calculate MODE80 drive access links

RUN PGM=MATRIX MSG='Mode Choice 4: calculate commuter rail (mode 80) drive access links'

FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_PNR_nodes.dbf', AUTOARRAY=ALLFIELDS

FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf', AUTOARRAY=ALLFIELDS

FILEI DBI[3] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode8.dbf', AUTOARRAY=ALLFIELDS

FILEI MATI[1] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_Pk.mtx'

FILEO PRINTO[1] = '@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_drive_links_Mode80.NTL'

;parameters

ZONES = 1

PRINT PRINTO=1, LIST=';;<<PT>>;; \n'

```
;define numeric arrays
ARRAY STOPS_NODE     = 99999,


      Closest_PNR    = 1000,
      Closest_Time   = 1000,
      Closest_Dist   = 1000,
      Closest_XYDist = 1000


;define character arrays
ARRAY TYPE=C55, RailLine_Name  = 99999,
      TYPE=C55, UniqueRailLine = 500



;assign stop node array based on node number as index (reduce need to loop in script)
;and identify number of unique rail lines
LOOP recnum=1,DBI.3.NUMRECORDS
    idx               = DBA.3.N[recnum]
    STOPS_NODE[idx]    = recnum
    RailLine_Name[idx] = DBA.3.ROUTE[recnum]


    ;identify unique rail lines
    Unique_Name = DBA.3.ROUTE[recnum]
    inlist      = 0
```

```
if (recnum=1)
   cnt_UniqueRail    = 1
   UniqueRailLine[1] = Unique_Name

   ;print unique lines to check file
   PRINT CSV=T, FORM=8.0,
FILE='@ParentDir@@ScenarioDir@Temp\4_ModeChoice\_check_unique_Lines_Mode8.csv',
      LIST=cnt_UniqueRail, UniqueRailLine[1]

else
   LOOP chk_unq=1,cnt_UniqueRail
      if (Unique_Name=UniqueRailLine[chk_unq])
         ;route name already in list
         inlist = 1
         BREAK
      endif
   ENDLOOP

   ;add unique rail iine to array
   if (inlist<>1)
      cnt_UniqueRail                = cnt_UniqueRail + 1
      UniqueRailLine[cnt_UniqueRail] = Unique_Name

      ;print unique lines to check file
      PRINT CSV=T, FORM=8.0,
FILE='@ParentDir@@ScenarioDir@Temp\4_ModeChoice\_check_unique_Lines_Mode8.csv',
```

```
            LIST=cnt_UniqueRail, UniqueRailLine[cnt_UniqueRail]
        endif
    endif
ENDLOOP



;calculate drive access links
;loop through TAZ
LOOP TAZ_Node=1,@Usedzones@
    procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
    PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'

    TAZ_X      = DBA.2.X[TAZ_Node]
    TAZ_Y      = DBA.2.Y[TAZ_Node]
    TAZ_COUNTY = DBA.2.COUNTY[TAZ_Node]

    ;loop through unique rail lines
    LOOP lp_Rail=1,cnt_UniqueRail
        ;assign name of unique rail line for this loop iteration
        Check_RailLine = UniqueRailLine[lp_Rail]

        ;initialize PNR variables
        cnt_EligiblePNR = 0
        SET VAL=9999, VARS=Closest_Time    ;sets all elements in array to 9999 so 0 values
in array are sorted to back in the array
```

```
;populate closest PNR arrays

LOOP PNR_recnum=1,DBI.1.NUMRECORDS

    ;assign PNR variables

    PNR_Node   = DBA.1.N[PNR_recnum]

    PNR_X      = DBA.1.X[PNR_recnum]

    PNR_Y      = DBA.1.Y[PNR_recnum]

    PNR_Code   = DBA.1.PNR[PNR_recnum]

    PNR_TAZID  = DBA.1.TAZID[PNR_recnum]


    ;check for eligible PNR node

    if (PNR_Code=@Mode80_EligiblePNR@ & STOPS_NODE[PNR_Node]>0 &
RailLine_Name[PNR_Node]=Check_RailLine)

        ;count eligible PNRs

        cnt_EligiblePNR = cnt_EligiblePNR + 1


        ;calculate distance from TAZ to PNR node

        xydist = SQRT( (TAZ_X-PNR_X)^2 + (TAZ_Y-PNR_Y)^2 ) / 1609.344    ;convert
meters to miles


        ;lookup AM time and distance skim values based on PNR's TAZID

        OverNetDist = MATVAL(1, 11, TAZ_Node, PNR_TAZID, 0)   ;MATVAL(file#,
matrix#, I, J, ReturnCode if error)

        IVT         = MATVAL(1, 5,  TAZ_Node, PNR_TAZID, 0)

        OVT         = MATVAL(1, 1,  TAZ_Node, PNR_TAZID, 0)
```

```
            ;calculate total time = in-vehicle time + out-of-vehicle time
            TotalTime   = IVT ;+ OVT


            ;check for zero values
            if (IVT=0 | OverNetDist=0)
               TotalTime   = 9999
               OverNetDist = 999
            endif


            ;assign initial array element
            Closest_PNR[PNR_recnum]    = PNR_Node
            Closest_Time[PNR_recnum]   = TotalTime
            Closest_Dist[PNR_recnum]   = OverNetDist
            Closest_XYDist[PNR_recnum] = xydist


         endif  ;check for eligble PNR node


      ENDLOOP  ;populate closest PNR arrays
```

;sort ascending based on shortest time, then shortest over-net distance, then shortest xy distance, then PNR node number

```
      SORT ARRAY='+Closest_Time','+Closest_Dist','+Closest_XYDist','+Closest_PNR'


      ;print drive access links
      LOOP Conct_num=1,@Mode80_NumCon@
```

```
;assign closest nth eligible PNR node and calculate avg over the network speed
PNR_Node = Closest_PNR[Conct_num]
PNR_Time = Closest_Time[Conct_num]
PNR_Dist = Closest_Dist[Conct_num]

OverNetSpeed = 60 * PNR_Dist / PNR_Time      ;in mph



;add dummy link in case no transit mode in scenario (need at least 1 to keep from crashing)
if (print_dummy_80=0)
   PRINT PRINTO=1, LIST='NT LEG=1-1,  MODE=80,  COST=2.40,  DIST=1.00, ONEWAY=F,  SPEED=25.0'
   print_dummy_80=1
endif



;print drive access links if less than MaxTime
if (TAZ_COUNTY=@SL_ID@)
   ;use SL county MaxTime
   if (PNR_Time<=@Mode80_MaxTime_SL@)
      ;print out link as NT leg
      PRINT PRINTO=1,
         LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=80, COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=', OverNetSpeed(6.2)
   endif
```

```
            elseif (TAZ_COUNTY=@UT_ID@)
              ;use UT county MaxTime
              if (PNR_Time<=@Mode80_MaxTime_UT@)
                ;print out link as NT leg
                PRINT PRINTO=1,
                  LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=80,
COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=',
OverNetSpeed(6.2)
              endif


            else
              ;use general MaxTime
              if (PNR_Time<=@Mode80_MaxTime@)
                ;print out link as NT leg
                PRINT PRINTO=1,
                  LIST='NT LEG=', TAZ_Node(6.0),' -', PNR_Node(6.0), ',  MODE=80,
COST=', PNR_Time(6.2), ',  DIST=', PNR_Dist(6.2), ',  ONEWAY=T,  SPEED=',
OverNetSpeed(6.2)
              endif
            endif  ;print drive access links if less than MaxTime


          ENDLOOP  ;print drive access links


        ENDLOOP ;loop through unique rail lines


      ENDLOOP  ;loop through TAZ
```

```
ENDRUN



;CREATE MaaS DRIVE/Egress ACCESS LINKS
===========================================================================
==========================================


;calculate MODE30 drive access links

RUN PGM=MATRIX  MSG='Mode Choice 4: calculate MaaS (mode 20) drive/egress access
links'

   FILEI DBI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_HwyNodes.dbf',
AUTOARRAY=ALLFIELDS

   FILEI DBI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_StopsMode7.dbf',
AUTOARRAY=ALLFIELDS


   FILEO PRINTO[1] =
'@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_MaaS_links_Mode20.NTL'


   ;parameters

   ZONES = 1



   PRINT PRINTO=1, LIST=';;<<PT>>;; \n'



   ;loop through TAZ nodes
```

```
LOOP TAZ_Node=1,@UsedZones@
   procrec = ROUND(TAZ_Node / @UsedZones@ * 100)
   PRINT PRINTO=0 LIST='Process Completed: ', procrec(4.0), '%'

   ;calculate TAZ X & Y coordinates, TAZ area in sqare miles
   TAZ_X  = DBA.1.X[TAZ_Node]
   TAZ_Y  = DBA.1.Y[TAZ_Node]
   County = DBA.1.COUNTY[TAZ_Node]

   ;calculate max drive distance (min)
   if (County=@SL_ID@)
      maxdrive = @Mode20_MaxDist_SL@
   elseif (County=@UT_ID@)
      maxdrive = @Mode20_MaxDist_UT@
   else
      maxdrive = @Mode20_MaxDist@
   endif

   ;loop through stop nodes
   LOOP StopRec=1,DBI.2.NUMRECORDS
      ;calculate stop node number and X & Y coordinates
      Stop_Node = DBA.2.N[StopRec]
      Stop_X    = DBA.2.X[StopRec]
      Stop_Y    = DBA.2.Y[StopRec]
```

```
          ;calculate distance & from TAZ to stop node in miles

          xydist = SQRT( (TAZ_X-Stop_X)^2 + (TAZ_Y-Stop_Y)^2 ) / 1609.344    ;convert
meters to miles

          MaaSCost = xydist/25*60 + xydist*@AOC_MaaS@/@VOT_Auto_Per@

          ;print walk access support link if <= 1 max drive distance

          if (xydist<=maxdrive)

             ;print out link as NT leg

             PRINT PRINTO=1,

                LIST='NT LEG=', TAZ_Node(6.0),' -', Stop_Node(6.0), ',  MODE=20,  COST=',
MaaSCost(6.2), ',  DIST=', xydist(6.2), ',  ONEWAY=F,  SPEED=25.0'

          endif


       ENDLOOP  ;loop through stop records


    ENDLOOP  ;loop through TAZ centroids


ENDRUN
```

*(DEL 04_Create_drive_access_links.txt)

Appendix B: Script for Updating Transit Network Skims

```
;==============================================================================
==========================

;        Prepare NETWORK for transit skims (use Distrib loaded net)

;=============================================================================
===================


;add rail-bus transit network to distribution network & calculate congested bus speeds

RUN PGM=NETWORK   MSG='Mode Choice 5: add rail-bus net to Distrib net & calc congested bus speeds'

   FILEI NETI[1] = '@ParentDir@@ScenarioDir@3_Distribute\Distrib_Network__Summary.net'

   FILEI NETI[2] = '@ParentDir@@ScenarioDir@0_InputProcessing\c_BusRailLinks.net'


   FILEO NETO = '@ParentDir@@ScenarioDir@4_ModeChoice\_CongestedHwyNet_withBusRailLinks.net'

   FILEO LINKO = '@ParentDir@@ScenarioDir@4_ModeChoice\_CongestedHwyNet_withBusRailLinks.dbf'


   ;set NETWORK parameters

   ZONES = @UsedZones@

   MERGE RECORD=T
```

```
;calculate bus & BRT link speed

;  bus speed to auto speed ratios:

;  ratio_fway        = 0.95      ;bus speed to auto speed - freeways

;  ratio_ramp        = 0.75      ;bus speed to auto speed - freeway ramps

;  ratio_part        = 0.60      ;bus speed to auto speed - principal arterials & expressways

;  ratio_mart_urbcbd = 0.55      ;bus speed to auto speed - minor arterials, urban/cbd

;  ratio_mart_subrur = 0.65      ;bus speed to auto speed - minor arterials, suburban/rural

;  ratio_collector   = 0.60      ;bus speed to auto speed - collectors

;  minimum_bus_speed = 10.0      ;mph


;rail/bus links

if (li.2.FT>=50)

   ;assume speed on rail/bus link file

   SPEED_AM = li.2.SPEED

   SPEED_MD = li.2.SPEED


;freeways

elseif (li.1.FT=20-26,30-40)

   SPEED_AM = MAX(@minimum_bus_speed@, li.1.AM_SPD*@ratio_fway@)

   SPEED_MD = MAX(@minimum_bus_speed@, li.1.MD_SPD*@ratio_fway@)


;ramps

elseif (li.1.FT=28-29,41-42)
```

```
    SPEED_AM = MAX(@minimum_bus_speed@, li.1.AM_SPD*@ratio_ramp@)
    SPEED_MD = MAX(@minimum_bus_speed@, li.1.MD_SPD*@ratio_ramp@)

;minor arterials
elseif (li.1.FT=3)
    ;suburban
    if (li.1.AREATYPE=4-5)
        SPEED_AM = MAX(@minimum_bus_speed@, li.1.AM_SPD*@ratio_mart_urbcbd@)
        SPEED_MD = MAX(@minimum_bus_speed@, li.1.MD_SPD*@ratio_mart_urbcbd@)
    else
        SPEED_AM = MAX(@minimum_bus_speed@, li.1.AM_SPD*@ratio_mart_subrur@)
        SPEED_MD = MAX(@minimum_bus_speed@, li.1.MD_SPD*@ratio_mart_subrur@)
    endif

;collectors & locals
elseif (li.1.FT=1,4-8)
    SPEED_AM = MAX(@minimum_bus_speed@, li.1.AM_SPD*@ratio_collector@)
    SPEED_MD = MAX(@minimum_bus_speed@, li.1.MD_SPD*@ratio_collector@)

;principal arterials & expressways
else
    SPEED_AM = MAX(@minimum_bus_speed@, li.1.AM_SPD*@ratio_part@)
    SPEED_MD = MAX(@minimum_bus_speed@, li.1.MD_SPD*@ratio_part@)
```

```
    endif

ENDRUN

LOOP period = 1,2
    ;set name variable for output files
    if (period=1)
        prd       = 'Pk'
        TranSpeed = 'SPEED_AM'
    else
        prd       = 'Ok'
        TranSpeed = 'SPEED_MD'
    endif


;=============================================================================
====================
        ;                          MaaS to LRT ONLY MODE 7 skim

;=============================================================================
====================
        RUN PGM=PUBLIC TRANSPORT   MSG='Mode Choice 5: skim MaaS to LRT - @prd@'
            READ     FILE               =  '@ParentDir@1_Inputs\4_Transit\@Mlin@Readlines.block'
;read in transit line files
```

FILEI NETI = '@ParentDir@@ScenarioDir@4_ModeChoice_CongestedHwyNet_withBusRailLinks.net' ;highway network with Rail.link & Bus.link links

FILEI SYSTEMI = '@ParentDir@1_Inputs\4_Transit\@Mlin@PT_Parameter\GENERAL_System.PTS' ;system file

FILEI FAREI = '@ParentDir@1_Inputs\4_Transit\@Mlin@PT_Parameter\GENERAL_Fare.FAR' ;fare file

FILEI FACTORI[1] = '@ParentDir@1_Inputs\4_Transit\@Mlin@PT_Parameter\FAC_7_LRTonly_walk.FAC' ;drive-to-LRT factors file

FILEI FACTORI[2] = '@ParentDir@1_Inputs\4_Transit\@Mlin@PT_Parameter\FAC_7_LRTonly_drive.FAC' ;drive-to-LRT factors file

FILEI NTLEGI[1] = '@ParentDir@1_Inputs\4_Transit_General Hand-Coded Support Links\General_hand_coded_walk_links.NTL' ;GENERAL hand coded walk links

FILEI NTLEGI[2] = '@ParentDir@1_Inputs\4_Transit_General Hand-Coded Support Links\General_hand_coded_drive_links.NTL' ;GENERAL hand coded drive links

FILEI NTLEGI[3] = '@ParentDir@1_Inputs\4_Transit\@Mlin@Scenario_hand_coded_walk_links.NTL' ;SCENARIO hand coded walk links

FILEI NTLEGI[4] = '@ParentDir@1_Inputs\4_Transit\@Mlin@Scenario_hand_coded_drive_links.NTL' ;SCENARIO hand coded drive links

FILEI NTLEGI[5] = '@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_Xfer_Links.NTL' ;AUTO-GENERATED transfer links

FILEI NTLEGI[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_walk_links.NTL' ;AUTO-GENERATED walk links

FILEI NTLEGI[7] = '@ParentDir@@ScenarioDir@4_ModeChoice\NTL\auto_generated_MaaS_links_Mode20.NTL' ;AUTO-GENERATED MaaS links - LRT

FILEI MATI[1] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_@prd@.mtx' ;pk/ok auto skim

FILEI LOOKUPI = '@ParentDir@1_Inputs\1_TAZ\@TAZ_DBF@'

FILEO NETO = '@ParentDir@@ScenarioDir@4_ModeChoice\1b_EnumeratedRoutes\PTNETOUT_MaaS_LRT_skims_@prd@.NET' ;create PT network with walk access links

FILEO ROUTEO[1] = '@ParentDir@@ScenarioDir@4_ModeChoice\1b_EnumeratedRoutes\ROUTEO_MaaS_LRTonly_skims_@prd@.RET' ;enumerated route file for userclass 1 (need to activate SKIMIJ phase)

;FILEO ROUTEO[3] = '@ParentDir@@ScenarioDir@4_ModeChoice\1b_EnumeratedRoutes\ROUTEO_@D_LRT_skims@_@prd@.RET' ;enumerated route file for userclass 3 (need to activate SKIMIJ phase)

FILEO REPORTO = '@ParentDir@@ScenarioDir@4_ModeChoice\1c_Reports\PTREPORT_MaaS_LRT_skims_@prd@.RPT' ;PT report file

FILEO MATO[1] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\MaaS_LRTonly_skims_@prd@.mtx',

MO=1-23,

NAME=INITWAIT,

XFERWAIT,

DRIVETIME,

T4,

T5,

T6,

T7,

T8,

T9,

T456789,

WALKTIME,

XFARE,

DRIVEDIST,

D4,

D5,

D6,

```
        D7,
        D8,
        D9,
        D456789,
        BOARDINGS,
        TRANSFERS,
        RAIL_XFERS

    ;define lookup function
    LOOKUP LOOKUPI=1,
      LIST=N,
      INTERPOLATE=F,
      NAME=FreeZones,
      LOOKUP[1]=TAZID, RESULT=@EcoEdPassZones@,
      LOOKUP[2]=TAZID, RESULT=@FreeFareZones@
    ;set parameters
    USERCLASSES = 1                ;1=walk-LRTonly 2=drive-LRTonly
    TRANTIME    = (li.DISTANCE / li.@TranSpeed@ * 60)    ;enclose expressions in
parentheses
    HDWAYPERIOD = @period@             ;headway to use in LINE files



    ;add walk access links
```

```
PHASE=DATAPREP
  GENERATE READNTLEGI=1                    ;GENERAL hand coded walk links
  GENERATE READNTLEGI=2                    ;GENERAL hand coded drive links
  GENERATE READNTLEGI=3                    ;SCENARIO hand coded walk links
  GENERATE READNTLEGI=4                    ;SCENARIO hand coded drive links
  GENERATE READNTLEGI=5                    ;AUTO-GENERATED transfer links
  GENERATE READNTLEGI=6                    ;AUTO-GENERATED walk links
  GENERATE READNTLEGI=7                   ;AUTO-GENERATED MaaS links - LRT
ENDPHASE



;perform transit skims
PHASE=SKIMIJ
   mw[01] = IWAITA(0)                        ;actual initial wait time
   mw[02] = XWAITA(0)                         ;actual transfer wait time
   mw[03] = TIMEA(0,20,30,40,50,60,70,80,90)      ;actual auto in-vehicle time (IVT)
   mw[04] = TIMEA(0,4)                       ;actual transit in-vehicle time (IVT) - local bus
   mw[05] = TIMEA(0,5)                       ;actual transit in-vehicle time (IVT) - BRT
   mw[06] = TIMEA(0,6)                      ;actual transit in-vehicle time (IVT) - express bus
   mw[07] = TIMEA(0,7)                       ;actual transit in-vehicle time (IVT) - LRT
   mw[08] = TIMEA(0,8)                       ;actual transit in-vehicle time (IVT) - CRT
   mw[09] = TIMEA(0,9)                         ;actual transit in-vehicle time (IVT) - enhanced
BRT
```

```
mw[10] = TIMEA(0,4,5,6,7,8,9)           ;actual transit in-vehicle time (IVT) - all transit modes
mw[11] = TIMEA(0,11,12,21,22)           ;actual walk access and walk transfer out-of-vehicle time (OVT)
mw[12] = FAREA(0,4,5,6,7,8,9)           ;actual transit mode fare
mw[13] = DIST(0,20,30,40,50,60,70,80,90)  ;auto in-vehicle distance
mw[14] = DIST(0,4)                      ;transit in-vehicle distance - local bus
mw[15] = DIST(0,5)                      ;transit in-vehicle distance - BRT
mw[16] = DIST(0,6)                      ;transit in-vehicle distance - express bus
mw[17] = DIST(0,7)                      ;transit in-vehicle distance - LRT
mw[18] = DIST(0,8)                      ;transit in-vehicle distance - CRT
mw[19] = DIST(0,9)                      ;transit in-vehicle distance - enhanced BRT
mw[20] = DIST(0,4,5,6,7,8,9)            ;transit in-vehicle distance - all transit modes
mw[21] = BRDINGS(0,4,5,6,7,8,9)         ;number of boardings
mw[22] = MAX(0, mw[21]-1)               ;number of transfers (transfers = boardings - 1)
mw[23] = 0                              ;rail-to-rail transfers (set to zero for now)
ENDPHASE


PHASE=MATO
;adjust local bus paths
JLOOP
```

;remove/exclude path if:

; - no LRT IVT

; - total auto + transit distnace > 1.5 * auto free-flow skim distance

; - drive access IVT>0 and auto free-flow skim distance between I & J is <= 3 miles

;if (mw[07][j]=0 | (mw[13][j]+mw[20][j]>mi.1.dist_GP[j]*1.5) | (mw[03]>0 & mi.1.dist_GP[j]<=3))

if (mw[07][j]=0)

mw[01] = 0 ;actual initial wait time

mw[02] = 0 ;actual transfer wait time

mw[03] = 0 ;actual auto in-vehicle time (IVT)

mw[04] = 0 ;actual transit in-vehicle time (IVT) - local bus

mw[05] = 0 ;actual transit in-vehicle time (IVT) - BRT

mw[06] = 0 ;actual transit in-vehicle time (IVT) - express bus

mw[07] = 0 ;actual transit in-vehicle time (IVT) - LRT

mw[08] = 0 ;actual transit in-vehicle time (IVT) - CRT

mw[09] = 0 ;actual transit in-vehicle time (IVT) - enhanced BRT

mw[10] = 0 ;actual transit in-vehicle time (IVT) - all transit modes

mw[11] = 0 ;actual walk access and walk transfer out-of-vehicle time (OVT)

mw[12] = 0 ;actual transit mode fare

mw[13] = 0 ;auto in-vehicle distance

mw[14] = 0 ;transit in-vehicle distance - local bus

mw[15] = 0 ;transit in-vehicle distance - BRT

mw[16] = 0 ;transit in-vehicle distance - express bus

```
            mw[17] = 0      ;transit in-vehicle distance - LRT
            mw[18] = 0      ;transit in-vehicle distance - CRT
            mw[19] = 0      ;transit in-vehicle distance - enhanced BRT
            mw[20] = 0      ;transit in-vehicle distance - all transit modes
            mw[21] = 0      ;number of boardings
            mw[22] = 0      ;number of transfers (transfers = boardings - 1)
            mw[23] = 0      ;rail-to-rail transfers (set to zero for now)
          endif


          ;make fare free for Ecopass, Edpass and free-fare zones
          if (FreeZones(1,j)>0 & @demographicyear@>=FreeZones(1,j))  mw[12] = 0
          if (FreeZones(2,i)>0 & FreeZones(2,j)>0 & FreeZones(2,i)<=@demographicyear@ &
FreeZones(2,j)<=@demographicyear@)  mw[12] = 0
          ;if (j=@BYUmain@) mw[12][j] = 0    ;BYU students pay $60/year for a pass
        ENDJLOOP
      ENDPHASE
    ENDRUN


ENDLOOP  ;period (pk/ok)
```

Appendix C: Script to Include MaaS to Transit for HBW and HBO Trips

```
;System
    ;file to halt the model run if model crashes


;print time stamp
RUN PGM=MATRIX
    ZONES = 1
    ScriptStartTime = currenttime()
    PRINT FILE='@ParentDir@@ScenarioDir@_Log\RunTime.txt',
        APPEND=T,
        LIST='\n',
             '\n     Calc HBW-HBO Trips by Mode     ', formatdatetime(ScriptStartTime, 40, 0, 'yyyy-
mm-dd,  hh:nn:ss')
ENDRUN




;============================================================================
================================
;PURPOSE:  Mode Choice, Segmented by auto ownership (HBW, HBO)
;============================================================================
================================
```

;copy ASC file to calibration folder (currently does not work when running with HailMary.bat)

;*COPY "coeffs\HBO_MC_constants_Ok_0.txt" "coeffs\calib_const\HBO_MC_constants_Ok_0.txt"

;*COPY "coeffs\HBO_MC_constants_Pk_0.txt" "coeffs\calib_const\HBO_MC_constants_Pk_0.txt"

;*COPY "coeffs\HBW_MC_constants_Ok_0.txt" "coeffs\calib_const\HBW_MC_constants_Ok_0.txt"

;*COPY "coeffs\HBW_MC_constants_Pk_0.txt" "coeffs\calib_const\HBW_MC_constants_Pk_0.txt"

;Cluster: distrubute MATRIX call onto processor 2

DistributeMULTISTEP PROCESSID=ClusterNodeID PROCESSNUM=2

```
  purpose = 1
  period  = 1
  purp    = 'HBW'
  prd     = 'Pk'

 RUN PGM=MATRIX   MSG='Mode Choice 11: identify cells with trips - @purp@ @prd@'
  FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Pk.mtx'
  FILEI MATI[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Pk.mtx'
```

```
FILEI MATI[3] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Pk.mtx'
FILEI MATI[4] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Ok.mtx'
FILEI MATI[5] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Ok.mtx'
FILEI MATI[6] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Ok.mtx'


FILEO MATO[1] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx', MO=1



ZONEMSG = @ZoneMsgRate@  ;reduces print messages in TPP DOS. (i.e. runs faster).


jloop
 if((mi.1.1[j] + mi.1.2[j] + mi.2.1[j] + mi.2.2[j] + mi.3.1[j] + mi.3.2[j] +
     mi.4.1[j] + mi.4.2[j] + mi.5.1[j] + mi.5.2[j] + mi.6.1[j] + mi.6.2[j]) > 0)
  MW[1][j]=1
 endif
endjloop
ENDRUN
```

```
loop n=1,30    ; calibration loop - Set iters to 1 if not calibrating, 50 if calibrating
               ; BREAK statement after MATRIX call - see below
  n_1 = n-1

  RUN PGM=MATRIX    MSG='Mode Choice 11: calculate trip mode choice - @purp@ @prd@ - iter @n@'
   zones=@Usedzones@
   maxmw=700    ;resets the maximimum number of working matrices from 200 to 700

   FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\access_to_transit_markets.mtx'
   FILEI MATI[2] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LCL_skims@_@prd@.mtx'
   FILEI MATI[3] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LCL_skims@_@prd@.mtx'
   ;******************* BRT didn't fit in the pattern, SEE BELOW
   FILEI MATI[4] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_EXP_skims@_@prd@.mtx'
   FILEI MATI[5] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_EXP_skims@_@prd@.mtx'
   FILEI MATI[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LRT_skims@_@prd@.mtx'
   FILEI MATI[7] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LRT_skims@_@prd@.mtx'
```

FILEI MATI[8] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_CRT_skims@_@prd@.mtx'

FILEI MATI[9] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_CRT_skims@_@prd@.mtx'

FILEI MATI[10] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_@prd@.mtx'

FILEI MATI[11] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_@prd@.mtx'

FILEI MATI[12] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_@prd@.mtx'

FILEI MATI[13] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_@prd@.mtx'

FILEI MATI[14] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_mode9_skims@_@prd@.mtx'

FILEI MATI[15] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_mode9_skims@_@prd@.mtx'

FILEI MATI[17] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx'

;****************** BRT didn't fit in the pattern

FILEI MATI[18] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_BRT_skims@_@prd@.mtx'

FILEI MATI[19] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_BRT_skims@_@prd@.mtx'

FILEI MATI[20] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\direct_walk_premium_@prd@.mtx'

;****************** Add MaaS LRT

FILEI MATI[21] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\MaaS_LRTonly_skims_@prd@.mtx'

FILEI ZDATI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\Urbanization.dbf'

FILEO MATO[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_0veh_@prd@_tmp.mtx', MO=11,12,15,16,17,19,20,25-26,96-97,27-32,112,113,516,517,519,520,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p,MaaSLRT

FILEO MATO[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_1veh_@prd@_tmp.mtx', MO=41,42,45,46,47,49,50,55-56,98-99,57-62,134,135,546,547,549,550,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p,MaaSLRT

FILEO MATO[3] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_2veh_@prd@_tmp.mtx', MO=71,72,75,76,77,79,80,85-86,100-101,87-92,139,140,576,577,579,580,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9, MaaS, MaaSalone, MaaSSR2, MaaSSR3p,MaaSLRT

FILEO MATO[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\2_DetailedTripMatrices\@purp@_trips_@prd@_auto_managedlanes.mtx', MO=63-70, 563-570,

name=alone_non,alone_toll,sr2_non,sr2_hov,sr2_toll,sr3_non,sr3_hov,sr3_toll,

MaaSalone_non,MaaSalone_toll,MaaSsr2_non,MaaSsr2_hov,MaaSsr2_toll,MaaSsr3_non,MaaSsr3_hov,MaaSsr3_toll

FILEO MATO[7] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_@prd@_tolltrips_income.mtx', MO=130-133,630-633,

name=alone_low, shared_low, alone_high, shared_high, maasalone_low, maasshared_low, maasalone_high, maasshared_high

ZONEMSG = 10 ;reduces print messages in TPP DOS. (i.e. runs faster).

;assign skims and mode choice data into working matrices

```
READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_working_matrices_maas.block'
;read mode choice model coefficients and nesting constants
READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_coefficients.txt'
READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\1Nesting_Constants.txt'

;read in mode choice alternative specific constants
if (@calib@=1)
  READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_const\@purp@_MC_constants_@prd@_@n_1@.txt'
else
  READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_constants_@prd@_0.txt'
endif

if (i=1)  _count_calib=0   ;counter to check how many constants have been calibrated

loop VEH=1,3,1   ;loop through vehicle ownership segments

 ;assign alternative specific constants for each vehicle ownership segment
```

```
READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_asc.block'

loop INC=1,2,1   ;loop through income segments

 loop ACCESS=1,3,1    ;loop through access-to-transit segments

 ;based on market segment (vehicles/income) - assign a trip table
 ;based on income - assign a cost coefficient
 READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_market_trips.block'

  jloop
    if (MW[2][j]>0)
     if (i == @dummyzones@ | j == @dummyzones@)
     else
      READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_utilities_maa
s.block'  ;calculate utilities, relative probabilities, up the nesting structure
      READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_trips_maas.b
lock'    ;calculate trips by segment, working down the nesting structure
     endif
     ;if (i=834 & j=449)
```

```
                ;READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_check_calculations.block'
                ;endif
               endif


             endjloop
            endloop   ;ACCESS
          endloop   ;INC
        endloop   ;VEH


        IF (I==@Usedzones@)
          READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_regional_shares_maas.block'   ;computes the probs of each mode (in absolute terms) - for output
          READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_shares_maas.block'      ;prints the regional shares
          READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_trips_maas.block'       ;prints the regional trips


          IF (@calib@=1)
```

```
            READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_modeshare_targets\calib_target
_@purp@_@prd@.txt'          ;read in target shares by mode, by segment
            READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_update_constants.block'
;adjust alternative specific bias constants
          ENDIF
        ENDIF ;if i==usedzones


        ENDRUN


        ;check calibration
        n_HBW_Pk = n
        IF (calib==0) BREAK  ;if not calibrating, don't loop through calibration routine
        IF (MATRIX._count_calib == num_calib_HBW_HBO) BREAK  ;if calibrated, break out of
calibration loop
    ENDLOOP ;n


;Cluster: end of group distributed to processor 2
EndDistributeMULTISTEP



;Cluster: distrubute MATRIX call onto processor 3
DistributeMULTISTEP PROCESSID=ClusterNodeID PROCESSNUM=3
```

```
purpose = 1
period  = 2
purp    = 'HBW'
prd     = 'Ok'

RUN PGM=MATRIX   MSG='Mode Choice 11: identify cells with trips - @purp@ @prd@'
FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Pk.mtx'
FILEI MATI[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Pk.mtx'
FILEI MATI[3] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Pk.mtx'
FILEI MATI[4] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Ok.mtx'
FILEI MATI[5] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Ok.mtx'
FILEI MATI[6] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Ok.mtx'

FILEO MATO[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx', MO=1


ZONEMSG = @ZoneMsgRate@  ;reduces print messages in TPP DOS. (i.e. runs faster).
```

```
  jloop
   if((mi.1.1[j] + mi.1.2[j] + mi.2.1[j] + mi.2.2[j] + mi.3.1[j] + mi.3.2[j] +
       mi.4.1[j] + mi.4.2[j] + mi.5.1[j] + mi.5.2[j] + mi.6.1[j] + mi.6.2[j]) > 0)
    MW[1][j]=1
   endif
  endjloop
 ENDRUN



  loop n=1,30    ; calibration loop - Set iters to 1 if not calibrating, 50 if calibrating
                 ; BREAK statement after MATRIX call - see below
     n_1 = n-1


     RUN PGM=MATRIX     MSG='Mode Choice 11: calculate trip mode choice - @purp@ @prd@ - iter @n@'
       zones=@Usedzones@
       maxmw=700    ;resets the maximimum number of working matrices from 200 to 700


       FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\access_to_transit_markets.mtx'
       FILEI MATI[2] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LCL_skims@_@prd@.mtx'
```

FILEI MATI[3] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LCL_skims@_@prd@.mtx'

;****************** BRT didn't fit in the pattern, SEE BELOW

FILEI MATI[4] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_EXP_skims@_@prd@.mtx'

FILEI MATI[5] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_EXP_skims@_@prd@.mtx'

FILEI MATI[6] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LRT_skims@_@prd@.mtx'

FILEI MATI[7] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LRT_skims@_@prd@.mtx'

FILEI MATI[8] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_CRT_skims@_@prd@.mtx'

FILEI MATI[9] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_CRT_skims@_@prd@.mtx'

FILEI MATI[10] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_@prd@.mtx'

FILEI MATI[11] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_@prd@.mtx'

FILEI MATI[12] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_@prd@.mtx'

FILEI MATI[13] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_@prd@.mtx'

```
FILEI MATI[14] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_mode9_skims@_@prd@.mtx'
FILEI MATI[15] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_mode9_skims@_@prd@.mtx'

FILEI MATI[17] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx'

;******************* BRT didn't fit in the pattern
FILEI MATI[18] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_BRT_skims@_@prd@.mtx'
FILEI MATI[19] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_BRT_skims@_@prd@.mtx'
FILEI MATI[20] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\direct_walk_premium_@prd@.mtx'

FILEI ZDATI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\Urbanization.dbf'

FILEO MATO[1] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_0veh_@prd@_tmp.mtx',
MO=11,12,15,16,17,19,20,25-26,96-97,27-32,112,113,516,517,519,520,
                    name=motor,nonmotor,transit,auto,DA,SR2,SR3p,
```

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9, MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_1veh_@prd@_tmp.mtx', MO=41,42,45,46,47,49,50,55-56,98-99,57-62,134,135,546,547,549,550,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[3] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_2veh_@prd@_tmp.mtx', MO=71,72,75,76,77,79,80,85-86,100-101,87-92,139,140,576,577,579,580,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9, MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\2_DetailedTripMatrices\@purp@_trips_@prd@_auto_managedlanes.mtx', MO=63-70, 563-570,

name=alone_non,alone_toll,sr2_non,sr2_hov,sr2_toll,sr3_non,sr3_hov,sr3_toll,

MaaSalone_non,MaaSalone_toll,MaaSsr2_non,MaaSsr2_hov,MaaSsr2_toll,MaaSsr3_non,MaaSsr3_hov,MaaSsr3_toll

FILEO MATO[7] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_@prd@_tolltrips_income.mtx', MO=130-133,630-633,

```
name=alone_low, shared_low, alone_high, shared_high,maasalone_low, maasshared_low, maasalone_high, maasshared_high

    ZONEMSG = 10  ;reduces print messages in TPP DOS. (i.e. runs faster).

    ;assign skims and mode choice data into working matrices
    READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_working_matrices.block'

    ;read mode choice model coefficients and nesting constants
    READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_coefficients.txt'
    READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\1Nesting_Constants.txt'

    ;read in mode choice alternative specific constants
    if (@calib@=1)
      READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_const\@purp@_MC_constants_@prd@_@n_1@.txt'
    else
      READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_constants_@prd@_0.txt'
    endif
```

```
        if (i=1)  _count_calib=0   ;counter to check how many constants have been calibrated

        loop VEH=1,3,1   ;loop through vehicle ownership segments

          ;assign alternative specific constants for each vehicle ownership segment

          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_asc.block'


          loop INC=1,2,1   ;loop through income segments

            loop ACCESS=1,3,1    ;loop through access-to-transit segments

            ;based on market segment (vehicles/income) - assign a trip table

            ;based on income - assign a cost coefficient

            READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_market_trips.block'


              jloop

                if (MW[2][j]>0)

                 if (i == @dummyzones@ | j == @dummyzones@)

                 else

                  READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_utilities_maa
s.block'  ;calculate utilities, relative probabilities, up the nesting structure

                  READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_trips_maas.b
lock'    ;calculate trips by segment, working down the nesting structure

                 endif
```

```
;if (i=834 & j=449)
; READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_check_calculations.block'
;endif
endif

endjloop
endloop   ;ACCESS
endloop   ;INC
endloop   ;VEH

IF (I==@Usedzones@)
READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_regional_shares_maas.block'   ;computes the probs of each mode (in absolute terms) - for output
READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_shares_maas.block'      ;prints the regional shares
READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_trips_maas.block'       ;prints the regional trips

IF (@calib@=1)
```

```
            READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_modeshare_targets\calib_target
_@purp@_@prd@.txt'          ;read in target shares by mode, by segment

            READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_update_constants.block'
;adjust alternative specific bias constants

          ENDIF

        ENDIF ;if i==usedzones

       ENDRUN


       ;check calibration

       n_HBW_Ok = n

       IF (calib==0) BREAK  ;if not calibrating, don't loop through calibration routine

       IF (MATRIX._count_calib == num_calib_HBW_HBO) BREAK  ;if calibrated, break out of
calibration loop

   ENDLOOP ;n


;Cluster: end of group distributed to processor 3

EndDistributeMULTISTEP




;Cluster: distrubute MATRIX call onto processor 4

DistributeMULTISTEP PROCESSID=ClusterNodeID PROCESSNUM=4

   purpose  = 2
```

```
  period   = 1
  purp     = 'HBO'
  prd      = 'Pk'

RUN PGM=MATRIX   MSG='Mode Choice 11: identify cells with trips - @purp@ @prd@'
  FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Pk.mtx'
  FILEI MATI[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Pk.mtx'
  FILEI MATI[3] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Pk.mtx'
  FILEI MATI[4] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Ok.mtx'
  FILEI MATI[5] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Ok.mtx'
  FILEI MATI[6] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Ok.mtx'

  FILEO MATO[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx', MO=1

  ZONEMSG = @ZoneMsgRate@  ;reduces print messages in TPP DOS. (i.e. runs faster).

  jloop
   if((mi.1.1[j] + mi.1.2[j] + mi.2.1[j] + mi.2.2[j] + mi.3.1[j] + mi.3.2[j] +
```

```
      mi.4.1[j] + mi.4.2[j] + mi.5.1[j] + mi.5.2[j] + mi.6.1[j] + mi.6.2[j]) > 0)
     MW[1][j]=1
    endif
   endjloop
  ENDRUN


   loop n=1,30    ; calibration loop - Set iters to 1 if not calibrating, 50 if calibrating
                  ; BREAK statement after MATRIX call - see below
     n_1 = n-1


     RUN PGM=MATRIX    MSG='Mode Choice 11: calculate trip mode choice - @purp@ @prd@ - iter @n@'
      zones=@Usedzones@
      maxmw=700    ;resets the maximimum number of working matrices from 200 to 700


      FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\access_to_transit_markets.mtx'
      FILEI MATI[2] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LCL_skims@_@prd@.mtx'
      FILEI MATI[3] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LCL_skims@_@prd@.mtx'
      ;******************* BRT didn't fit in the pattern, SEE BELOW
      FILEI MATI[4] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_EXP_skims@_@prd@.mtx'
```

```
FILEI MATI[5] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_EXP_skims@_@prd@.mtx'
FILEI MATI[6] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LRT_skims@_@prd@.mtx'
FILEI MATI[7] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LRT_skims@_@prd@.mtx'
FILEI MATI[8] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_CRT_skims@_@prd@.mtx'
FILEI MATI[9] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_CRT_skims@_@prd@.mtx'
FILEI MATI[10] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_@prd@.mtx'

FILEI MATI[11] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_@prd@.mtx'
FILEI MATI[12] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_@prd@.mtx'
FILEI MATI[13] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_@prd@.mtx'

FILEI MATI[14] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_mode9_skims@_@prd@.mtx'
FILEI MATI[15] =
'@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_mode9_skims@_@prd@.mtx'
```

FILEI MATI[17] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx'

;******************* BRT didn't fit in the pattern

FILEI MATI[18] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_BRT_skims@_@prd@.mtx'

FILEI MATI[19] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_BRT_skims@_@prd@.mtx'

FILEI MATI[20] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\direct_walk_premium_@prd@.mtx'

FILEI ZDATI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\Urbanization.dbf'

FILEO MATO[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_0veh_@prd@_tmp.mtx', MO=11,12,15,16,17,19,20,25-26,96-97,27-32,112,113,516,517,519,520,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9, MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_1veh_@prd@_tmp.mtx', MO=41,42,45,46,47,49,50,55-56,98-99,57-62,134,135, 546,547,549,550,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[3] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_2veh_@prd@_tmp.mtx', MO=71,72,75,76,77,79,80,85-86,100-101,87-92,139,140,576,577,579,580,

name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\2_DetailedTripMatrices\@purp@_trips_@prd@_auto_managedlanes.mtx', MO=63-70, 563-570,

name=alone_non,alone_toll,sr2_non,sr2_hov,sr2_toll,sr3_non,sr3_hov,sr3_toll,

MaaSalone_non,MaaSalone_toll,MaaSsr2_non,MaaSsr2_hov,MaaSsr2_toll,MaaSsr3_non,MaaSsr3_hov,MaaSsr3_toll

FILEO MATO[7] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_@prd@_tolltrips_income.mtx', MO=130-133,630-633,

name=alone_low, shared_low, alone_high, shared_high, maasalone_low, maasshared_low, maasalone_high, maasshared_high

ZONEMSG = 10 ;reduces print messages in TPP DOS. (i.e. runs faster).

;assign skims and mode choice data into working matrices

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_working_matrices.block'

```
;read mode choice model coefficients and nesting constants

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_coefficients.txt'

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\1Nesting_Constants.txt'


;read in mode choice alternative specific constants

if (@calib@=1)

  READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_const\@purp@_MC_constants_@prd@_@n_1@.txt'

else

  READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_constants_@prd@_0.txt'

endif


if (i=1)  _count_calib=0   ;counter to check how many constants have been calibrated

loop VEH=1,3,1   ;loop through vehicle ownership segments

 ;assign alternative specific constants for each vehicle ownership segment

 READ FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_asc.block'

 loop INC=1,2,1   ;loop through income segments

  loop ACCESS=1,3,1    ;loop through access-to-transit segments
```

```
;based on market segment (vehicles/income) - assign a trip table

;based on income - assign a cost coefficient

READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_market_trips.block'

 jloop

   if (MW[2][j]>0)

    if (i == @dummyzones@ | j == @dummyzones@)

    else

     READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_utilities_maa
s.block'  ;calculate utilities, relative probabilities, up the nesting structure

     READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_trips_maas.b
lock'    ;calculate trips by segment, working down the nesting structure

    endif

    ;if (i=834 & j=449)

    ;                                                                          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_check_calculations.blo
ck'

    ;endif

   endif

 endjloop

endloop   ;ACCESS

endloop   ;INC

endloop   ;VEH
```

```
IF (I==@Usedzones@)
  READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_regional_shares_maas.
block'   ;computes the probs of each mode (in absolute terms) - for output
  READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_shares_maas.blo
ck'      ;prints the regional shares
  READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_trips_maas.block
'       ;prints the regional trips


  IF (@calib@=1)
    READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_modeshare_targets\calib_target
_@purp@_@prd@.txt'          ;read in target shares by mode, by segment
    READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_update_constants.block'
;adjust alternative specific bias constants
  ENDIF
ENDIF ;if i==usedzones
ENDRUN


;check calibration
n_HBO_Pk = n
```

```
        IF (calib==0) BREAK  ;if not calibrating, don't loop through calibration routine
        IF (MATRIX._count_calib == num_calib_HBW_HBO) BREAK  ;if calibrated, break out of
calibration loop
   ENDLOOP ;n


;Cluster: end of group distributed to processor 4
EndDistributeMULTISTEP



;Cluster: keep processing on processor 1 (Main)


   purpose  = 2
   period   = 2
   purp     = 'HBO'
   prd      = 'Ok'

  RUN PGM=MATRIX   MSG='Mode Choice 11: identify cells with trips - @purp@ @prd@'
   FILEI                                        MATI[1]                                        =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Pk.mtx'
   FILEI                                        MATI[2]                                        =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Pk.mtx'
   FILEI                                        MATI[3]                                        =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Pk.mtx'
```

```
FILEI MATI[4] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_Ok.mtx'
FILEI MATI[5] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_Ok.mtx'
FILEI MATI[6] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_Ok.mtx'
FILEO MATO[1] =
'@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx', MO=1


ZONEMSG = @ZoneMsgRate@  ;reduces print messages in TPP DOS. (i.e. runs faster).


jloop
 if((mi.1.1[j] + mi.1.2[j] + mi.2.1[j] + mi.2.2[j] + mi.3.1[j] + mi.3.2[j] +
     mi.4.1[j] + mi.4.2[j] + mi.5.1[j] + mi.5.2[j] + mi.6.1[j] + mi.6.2[j]) > 0)
  MW[1][j]=1
 endif
endjloop
ENDRUN


loop n=1,30    ; calibration loop - Set iters to 1 if not calibrating, 50 if calibrating
               ; BREAK statement after MATRIX call - see below
   n_1 = n-1
```

RUN PGM=MATRIX MSG='Mode Choice 11: calculate trip mode choice - @purp@ @prd@ - iter @n@'

zones=@Usedzones@

maxmw=700 ;resets the maximimum number of working matrices from 200 to 700

FILEI MATI[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\access_to_transit_markets.mtx'

FILEI MATI[2] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LCL_skims@_@prd@.mtx'

FILEI MATI[3] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LCL_skims@_@prd@.mtx'

;****************** BRT didn't fit in the pattern, SEE BELOW

FILEI MATI[4] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_EXP_skims@_@prd@.mtx'

FILEI MATI[5] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_EXP_skims@_@prd@.mtx'

FILEI MATI[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_LRT_skims@_@prd@.mtx'

FILEI MATI[7] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_LRT_skims@_@prd@.mtx'

FILEI MATI[8] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_CRT_skims@_@prd@.mtx'

FILEI MATI[9] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_CRT_skims@_@prd@.mtx'

FILEI MATI[10] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\skm_auto_@prd@.mtx'

FILEI MATI[11] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_0veh_@prd@.mtx'

FILEI MATI[12] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_1veh_@prd@.mtx'

FILEI MATI[13] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_2veh_@prd@.mtx'

FILEI MATI[14] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_mode9_skims@_@prd@.mtx'

FILEI MATI[15] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_mode9_skims@_@prd@.mtx'

FILEI MATI[17] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\pa_@purp@_trips.mtx'

;******************* BRT didn't fit in the pattern

FILEI MATI[18] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@W_BRT_skims@_@prd@.mtx'

FILEI MATI[19] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\@D_BRT_skims@_@prd@.mtx'

FILEI MATI[20] = '@ParentDir@@ScenarioDir@4_ModeChoice\1a_Skims\direct_walk_premium_@prd@.mtx'

FILEI ZDATI[1] = '@ParentDir@@ScenarioDir@0_InputProcessing\Urbanization.dbf'

```
FILEO MATO[1] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_0veh_@prd@_tmp.mtx', MO=11,12,15,16,17,19,20,25-26,96-97,27-32,112,113,516,517,519,520,

            name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

            wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9, MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[2] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_1veh_@prd@_tmp.mtx', MO=41,42,45,46,47,49,50,55-56,98-99,57-62,134,135,546,547,549,550,

            name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

            wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p

FILEO MATO[3] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_2veh_@prd@_tmp.mtx', MO=71,72,75,76,77,79,80,85-86,100-101,87-92,139,140,576,577,579,580,

            name=motor,nonmotor,transit,auto,DA,SR2,SR3p,

            wLCL, dLCL, wBRT, dBRT, wEXP, dEXP, wLRT, dLRT, wCRT, dCRT, wmode9, dmode9,MaaS, MaaSalone, MaaSSR2, MaaSSR3p


FILEO MATO[6] = '@ParentDir@@ScenarioDir@4_ModeChoice\2_DetailedTripMatrices\@purp@_trips_@prd@_auto_managedlanes.mtx', MO=63-70, 563-570,


name=alone_non,alone_toll,sr2_non,sr2_hov,sr2_toll,sr3_non,sr3_hov,sr3_toll,

MaaSalone_non,MaaSalone_toll,MaaSsr2_non,MaaSsr2_hov,MaaSsr2_toll,MaaSsr3_non,MaaSsr3_hov,MaaSsr3_toll
```

FILEO MATO[7] = '@ParentDir@@ScenarioDir@Temp\4_ModeChoice\@purp@_trips_@prd@_tolltrips_income.mtx', MO=130-133,630-633,

name=alone_low, shared_low, alone_high, shared_high,maasalone_low, maasshared_low, maasalone_high, maasshared_high

ZONEMSG = 10 ;reduces print messages in TPP DOS. (i.e. runs faster).

;assign skims and mode choice data into working matrices

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_working_matrices.block'

;read mode choice model coefficients and nesting constants

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_coefficients.txt'

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\1Nesting_Constants.txt'

;read in mode choice alternative specific constants

if (@calib@=1)

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_const\@purp@_MC_constants_@prd@_@n_1@.txt'

else

READ FILE = '@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\@purp@_MC_constants_@prd@_0.txt'

endif

```
if (i=1)  _count_calib=0   ;counter to check how many constants have been calibrated

loop VEH=1,3,1   ;loop through vehicle ownership segments

  ;assign alternative specific constants for each vehicle ownership segment

  READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_asc.block'

  loop INC=1,2,1   ;loop through income segments

    loop ACCESS=1,3,1    ;loop through access-to-transit segments

    ;based on market segment (vehicles/income) - assign a trip table

    ;based on income - assign a cost coefficient

    READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_market_trips.block'

      jloop

        if (MW[2][j]>0)

         if (i == @dummyzones@ | j == @dummyzones@)

         else

          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_utilities_maa
s.block'  ;calculate utilities, relative probabilities, up the nesting structure

          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_calculate_trips_maas.b
lock'    ;calculate trips by segment, working down the nesting structure

         endif

         ;if (i=834 & j=449)
```

```
                    ;                                                       READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_check_calculations.block'

                    ;endif

                  endif

              endjloop

            endloop   ;ACCESS

          endloop   ;INC

        endloop   ;VEH


        IF (I==@Usedzones@)

          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_regional_shares_maas.block'   ;computes the probs of each mode (in absolute terms) - for output

          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_shares_maas.block'      ;prints the regional shares

          READ
FILE='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_print_trips_maas.block'       ;prints the regional trips


          IF (@calib@=1)

            READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\coeffs\calib_modeshare_targets\calib_target_@purp@_@prd@.txt'          ;read in target shares by mode, by segment
```

READ
file='@ParentDir@2_ModelScripts\4_ModeChoice\block\HBW_HBO_update_constants.block'
;adjust alternative specific bias constants

ENDIF

ENDIF ;if i==usedzones

ENDRUN

;check calibration

n_HBO_Ok = n

IF (calib==0) BREAK ;if not calibrating, don't loop through calibration routine

IF (MATRIX._count_calib == num_calib_HBW_HBO) BREAK ;if calibrated, break out of calibration loop

ENDLOOP ;n

;Cluster: bring together all distributed steps before continuing

WAIT4FILES, FILES="ClusterNodeID2.Script.End", FILES="ClusterNodeID3.Script.End", FILES="ClusterNodeID4.Script.End", CheckReturnCode=T

*(DEL 11_MC_HBW_HBO.txt)

REFERENCES

Alemi, F., Circella, G., Handy, S. and Mokhtarian, P., 2018. What influences travelers to use Uber? Exploring the factors affecting the adoption of on-demand ride services in California. Travel Behaviour and Society, 13, pp.88-104.

Azevedo, C.L., Seshadri, R., Gao, S., Atasoy, B., Akkinepally, A.P., Christofa, E., Zhao, F., Trancik, J. and Ben-Akiva, M., 2018. Tripod: sustainable travel incentives with prediction, optimization, and personalization. In Transportation Research Board 97th Annual Meeting.

Blodgett, M., Khani, A., Negoescu, D. and Benjaafar, S., 2017. Public/Private Partnerships in Transit: Case Studies and Analysis.

Conway, M., Salon, D. and King, D., 2018. Trends in taxi use and the advent of ridehailing, 1995–2017: Evidence from the US National Household Travel Survey. Urban Science, 2(3), p.79.

Costain, C., Ardron, C., & Habib, K. N. (2012). Synopsis of users' behaviour of a carsharing program: A case study in Toronto. *Transportation Research Part A: Policy and Practice, 46*(3), 421-434.

Firnkorn, J., & Müller, M. (2011). What will be the environmental effects of new free-floating car-sharing systems? The case of car2go in Ulm. *Ecological Economics, 70*(8), 1519-1528.

Haghighi, N., Chamberlin, R., Fayyaz, K. and Liu, C., 2019. Impact of Shared Autonomous Vehicles (SAVs) on Vehicle Miles Traveled (VMT) in Utah (UDOT Research Report No. UT-19.10).

Hall, J.D., Palsson, C. and Price, J., 2018. Is Uber a substitute or complement for public transit? Journal of Urban Economics, 108, pp.36-50.

Hampshire, R., Simek, C., Fabusuyi, T., Di, X. and Chen, X., 2017. Measuring the Impact of an Unanticipated Disruption of Uber/Lyft in Austin, TX. Lyft in Austin, TX.

Xie, Y., Danaf, M., Azevedo, C.L., Akkinepally, A.P., Atasoy, B., Jeong, K., Seshadri, R. and Ben-Akiva, M., 2019. Behavioral modeling of on-demand mobility services: general framework and application to sustainable travel incentives. *Transportation*, pp.1-23.

Clewlow, R.R., 2016. Carsharing and sustainable travel behavior: Results from the San Francisco Bay Area. Transport Policy, 51, pp.158-164.

Clewlow, R. R., & Mishra, G. S. (2017). Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the United States. Retrieved from https://itspubs.ucdavis.edu/wp-content/themes/ucdavis/pubs/download_pdf.php?id=2752.

Dias, F.F., Lavieri, P.S., Garikapati, V.M., Astroza, S., Pendyala, R.M. and Bhat, C.R., 2017. A behavioral choice model of the use of car-sharing and ride-sourcing services. Transportation, 44(6), pp.1307-1323.

Dias, F.F., Lavieri, P.S., Kim, T., Bhat, C.R. and Pendyala, R.M., 2019. Fusing multiple sources of data to understand ride-hailing use. Transportation Research Record, 2673(6), pp.214-224.

Henao, A., 2017. Impacts of Ridesourcing-Lyft and Uber-on Transportation Including VMT, Mode Replacement, Parking, and Travel Behavior. University of Colorado at Denver.

Kooti, F., Grbovic, M., Aiello, L.M., Djuric, N., Radosavljevic, V. and Lerman, K., 2017, April. Analyzing Uber's ride-sharing economy. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 574-582). International World Wide Web Conferences Steering Committee.

Le Vine, S., Adamou, O., & Polak, J. (2014). Predicting new forms of activity/mobility patterns enabled by shared-mobility services through a needs-based stated-response method: Case study of grocery shopping. Transport Policy, 32, 60-68.

Monthly number of Uber's active users worldwide. Statista. Retrieved December 2, 2019.

Partnering with transit systems. Uber. Retrieved from https://www.uber.com/us/en/community/supporting-cities/transit/ (Accessed on December 2019)

Rayle, L., Dai, D., Chan, N., Cervero, R. and Shaheen, S., 2016. Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco. Transport Policy, 45, pp.168-178.

Sadowsky, N. and Nelson, E., 2017. The impact of ride-hailing services on public transportation use: A discontinuity regression analysis.

Schaller, B.: Second Chances, Regulation and Deregulation of Taxi and For-Hire Ride Services. TR News 43–48 (2018)

Smith, A., 2015. Using Open Data and Uber Data to Understand How Ridesharing Complements Public Transportation, Washington, D.C.

Statista. Monthly number of Uber's active users worldwide from 2016 to 2019. Retrieved from https://www.statista.com/statistics/833743/us-users-ride-sharing-services/ (Accessed on December 2019)

UTA on demand by Via. Retrieved from https://www.rideuta.com/Services/UTA-on-Demand-by-Via/ (Accessed on January 2020)

Young, M., Allen, J. and Farber, S., 2019. Measuring when Uber behaves as a substitute or complement to transit: An examination of travel-time differences in Toronto.

Yu, H. and Peng, Z.R., 2019. Exploring the spatial variation of ridesourcing demand and its relationship to built environment and socioeconomic factors with the geographically weighted Poisson regression. Journal of Transport Geography, 75, pp.147-163.