Travel Behavior Inventory

2021 – 2022 Wave 2 Household Travel Survey

Workshop Sessions 3 & 4

METROPOLITAN COUNCIL

Team

Project Review Team

Jonathan Ehrlich Project Manager

Ashley Asmus Deputy Project Manager

Dennis Farmer Project Advisor

Sara Maaske Project Advisor

Eric Lind Project Advisor

Jim Henricksen Project Advisor

Consultant Team

Prime Consultant

Translation Services (DBE)

Public Outreach (DBE)

Address Sample Provider

Printer (DBE)

Call Center

Workshop Agenda

THURSDAY, JUNE 30

Session 3: Dataset Introduction

11:00 AM – 12:00 PM CT, 1 hour

- Review dataset structure and key info for data users
- Discuss privacy considerations for working with the data
- Discuss methods for running key metrics (including code examples)
- Review new analysis using smartphone data (including code examples)

15-minute Break

Session 4: Weighting Methodology

12:15 – 1:30 PM CT, 1 hour 15 minutes

- Review weighting methodology

Session 3: TBI Dataset Introduction

2021-2022 Dataset Contents

The dataset includes six different data tables, often referred to as their own "level" of data:

Data Table	Records	Contents
Household	7,952	One record for each complete household
Vehicle	11,792	One record for each household vehicle (if any)
Person	15,805	One record for each household member
Day	50,057	One record for each person-day during the assigned travel period
Trip	182,446	One record for each trip reported (if any)
Location*	3,363,764	Two or more records for each trip reported (if any)

**This dataset is excluded from public data downloads. To access this table, please contact jonathan.ehrlich@metc.state.mn.us*

Data Table Relationships

Records in each data table can be linked to records in other tables, for example:

- Each person belongs to one household; each trip belongs to one person
- One household may have many vehicles, persons, and trips; a household may have no vehicles and trips, but must have at least one person

Records are linked through ID numbers in each table:

HH ID	Person #	Person ID	Trip #	Trip ID
21000152	1	2100015201	1	210001520100**1**
21000153	1	2100015301	1	210001530100**1**
21000153	1	2100015301	2	210001530100**2**

Joining Data Tables

All data tables can be joined into a single database as needed.

Some unique IDs are a combination of two variables. In these cases, joining on only one of the variables will create duplicate records.

	HOUSEHOLD	PERSON	VEHICLE	DAY	TRIP	LOCATION
Household	--	hh_id	hh_id	hh_id	hh_id	hh_id
Person	hh_id	--	--	person_id	person_id	person_id
Vehicle	hh_id	--	vehicle_id	--	vehicle_id	--
Day	hh_id	person_id	--	--	day_id	--
Trip	hh_id	person_id	vehicle_id	day_id	--	trip_id
Location	hh_id	person_id	--	--	trip_id	--

Data Coding and Labeling

TIME AND LOCATION STANDARDS: All timestamps are in Central Time, regardless of where the trip took place.

MISSING VALUES IN THE DATA:

Note: Continuous variables are not coded with missing value codes and text values are left blank.

- **A value or response was not required or is missing due to participant non-response (coded as 995).**

  Example: Respondents who were not employed were not asked to provide their workplace type.

  Example: A smartphone respondent did not complete a trip survey and trip mode is missing.

- **A respondent indicated that they didn't know the answer and skipped that question (coded as 998).**

  Example: Some respondents didn't know how much they personally pay to park at work.

- **A respondent indicated that they prefer not to answer a question (coded as 999).**

  Example: Some respondents preferred not to provide household income.

NOTE: IN PUBLIC DATA DOWNLOADS PROVIDED BY MET COUNCIL, MISSING VALUES HAVE BEEN REPLACED WITH "NA" TO FACILITATE ANALYSIS; RAW VERSIONS ARE AVAILABLE ON REQUEST.

Review Codebook and Dataset Guide

Questions?

We recommend all data users begin with the dataset guide and review the "Overview" and "Using the Data" tabs first.

A Brief Introduction to the data.table package in R

Benefits of Data.Table

Much of the R code that we'll be sharing during the session today leverages the data.table package in R.

- data.table provides an enhancement to the base data.frame available in R.
- Improves speed and memory management specifically for larger and more complicated datasets.
- Has a succinct and easy to learn syntactical structure.
- For more information: https://cran.r-project.org/web/packages/data.table/vignettes/datatable-intro.html
- For a python version: https://datatable.readthedocs.io/en/latest/index.html

Data.table Example Code

```
library(data.table)

# example data table
hh
         hh_id segment income days_complete
   1: 21000152       2      3             1
   2: 21000174       2      1             7
   3: 21000333       3      3             1
   4: 21000392       2      5             1
   5: 21000502       2      3             1

# assignment
hh[, new_var := pi]

# general form dt[i, j, by]

# row selection (i)
hh[income == 1]
        hh_id segment income days_complete new_var
  1: 21000174       2      1             7   3.142
  2: 21001166       3      1             1   3.142
  3: 21001174       3      1             1   3.142
  4: 21001391       2      1             6   3.142
  5: 21001454       3      1             1   3.142
```

```
# column selection (j)
hh[, .(hh_id, segment)]
      hh_id segment
1: 21000152       2
2: 21000174       2
3: 21000333       3
4: 21000392       2
5: 21000502       2

# aggregation (by)
hh[, .N, keyby = .(income)]
   income   N
1:      1 442
2:      2 758
3:      3 768
4:      4 602
5:      5 864
6:      6 156
7:    999 422

# joining two tables together
# performs a left join joining the day table to the hh
# table
day[hh, on = .(hh_id)]
```

Working with the Data

Privacy Considerations

This survey dataset contains sensitive information including personally identifiable data (PII).

- RSG has removed any PII used solely for survey administration (e.g., email addresses and nicknames).
- Survey access codes have been replaced with numerical household IDs (hh_id)
- It is important to follow your organization's privacy policy as well as the study's privacy policy when working with these data
- It is important to remember that combinations of variables might constitute personally identifiable data (e.g., combining trip destination location with trip purpose might uniquely identify someone).

NOTE: IN PUBLIC DATA DOWNLOADS PROVIDED BY MET COUNCIL, PII VARIABLES HAVE BEEN REMOVED.

Examples of Sensitive and PII

Variable Name(s)	Data Level	Data Type
sample_home_lat, sample_home_lon, home_lat, home_lon	Household	PII
income_detailed, income_broad	Household	Sensitive
ethnicity_afam, ethnicity_hapi, … ethnicity_white, race_asian_1, race_black_african_1, … race_hispanic_3	Person	Sensitive
ethnicity_other_specify, race_asian_other, race_black_african_other, … race_hispanic_other	Person	Text entry: May contain sensitive and PII
school_lat, school_lon	Person	PII
work_lat, work_lon	Person	PII
o_lat, o_lon, d_lat, d_lon	Trip	Sensitive/PII
d_purpose_other, mode_other_comment	Trip	Text entry: May contain sensitive and PII
make_model_other	Vehicle	Text entry: May contain sensitive and PII

NOTE: IN PUBLIC DATA DOWNLOADS PROVIDED BY MET COUNCIL, PII VARIABLES HAVE BEEN REMOVED.

Performing Analysis with Weights

Analyses designed to draw conclusions about travel behavior in the region (as opposed to just the survey respondents) should use weighted data.

When applied, the weights make the dataset representative of travel for residents within the study region for the time period studied – June 2021 to February 2022.

Just a reminder that the dataset does not include:

- Commercial vehicle travel
- Travel for persons residing in group quarters outside of the address-based sample frame (e.g., college dorms, institutional housing)
- Travel from non-residents (i.e., visitors to the region)
- Seasonal/holiday travel outside of the survey fielding period.

Applying Weights

Using weighted data generally involves summing the weights for the groups of interest. The sum of weights in each table represents the following groups:

Household: Represents the total number of households within the survey region.

Person: Represents the total number of persons within the survey region.

Vehicle: Represents the total number of personal vehicles of households in the survey region.

Day: Represents one day for all persons residing in the survey region. (This is because households with multiple travel days have their travel “averaged” to equal one day.)

Trip: Represents the total number of trips all persons residing in the survey region make on a typical day.

- Note: this differs from the number of trips made in the survey region on a typical day, given that some residents make trips outside the region.

Weighted Crosstabs or Descriptive Statistics

To calculate weighted crosstabs or descriptive statistics, sum the weights for that table.

Keep in mind the following when creating weighted statistics and summaries from HTS data:

1. Filter to the data relevant to your analysis.
   - Note that not all people are asked every question, so understanding the ‘missing value’ codes and ‘survey logic’ in the data dictionary are important.
2. Remember the survey design when using and interpreting weighted values.
   - For example, the TBI study included both one-day online and call center participation and seven-day smartphone participation. Therefore, it is best to avoid filtering by day of week since not all participants traveled on all days.

Using Weights for Statistical Analysis

Two common approaches for variance estimation using weights:

1. Replicate weights

- Generate multiple sets of weights
- Repeat analysis using each set of weights; “average” the results
- Drawback is that replicates increase dataset complexity and weighting process

2. Approximation using Taylor-series linearization

- Approximates the variance using a simpler to implement formula
- Simplifies the dataset and weight generation (one set of weights instead of many)
- Implemented in the survey R library or the samplics package in Python

Note: It is important to remember we are working with survey data and the sampling error is not the only source of error, but it is the only error that is easily measured.

Code

Weighted Means

```
library(survey)

# applies weights and stratification
hh_design = svydesign(
  ids = ~ hh_id,
  data = hh,
  weights = ~ hh_weight,
  strata = ~ sample_segment)

# weighted mean
hh[,
  .(weighted_mean = sum(hh_weight * num_hh_days) / sum(hh_weight))]

    weighted_mean
1: 2.1869394569598

# use linearization to approximate the standard error
svymean(~ num_hh_days, hh_design)

                     mean      SE
num_hh_days 2.18693945696 0.04889
```

Code

Weighted Proportions

```
hh[,
  .(weighted_N = sum(hh_weight),
    weighted_prop = sum(hh_weight) / hh[, sum(hh_weight)]),
  keyby = income_broad]
```

```
   income_broad      weighted_N     weighted_prop
1:            1 154324.32973056 0.106578252453326
2:            2 214063.60414492 0.147834919376995
3:            3 209972.26406368 0.145009390331696
4:            4 194244.57473886 0.134147657471460
5:            5 392618.73403185 0.271147256084706
6:            6 154824.81433196 0.106923893184702
7:          999 127942.48550794 0.088358631097114
```

```
svyciprop(
  ~ I(income_broad == 1),
  hh_design,
  method = 'mean')
```

```
                                          2.5%   97.5%
I(income_broad == 1) 0.10657825245 0.09369594736 0.11946
```

Generating Weighted Trip Rates

To calculate a weighted trip rate – the number of trips per day– data users must divide the number of weighted trips by the number of weighted travel days.

FOR EXAMPLE:

- If there are 300,000 weighted person-trips across 75,000 person-days, then the average person-trip rate is 4.0 per day.
- If there are 225,000 person-trips by car across 75,000 person-days, then the person-trip rate for car trips is 3.0.

Note: Data users should always calculate the number of weighted travel days using the day table rather than the trip table given that persons with zero-trip travel days do not have any records in the trip tables for those days.

Generating Weighted Trip Rates
Code Example

Weighted Trip Rates

```
trip_counts = trip[,
  .(weighted_num_trips = sum(trip_weight)),
  .(day_id)]

# join weighted trip count to day table
day[
  trip_counts,
  weighted_num_trips := i.weighted_num_trips,
  on = .(day_id)]

# set weighted_num_trips to 0 where no trips occured
day[num_trips == 0, weighted_num_trips := 0]

day[
  day_weight > 0,
  .(weighted_num_trips = sum(weighted_num_trips),
    weighted_num_days = sum(day_weight),
    weighted_trip_rate = sum(weighted_num_trips) /
    sum(day_weight))]
```

```
   weighted_num_trips weighted_num_days weighted_trip_rate
1:    12037646.899014   3365089.1543126    3.5772148513767
```

Error Margin for a Trip Rate

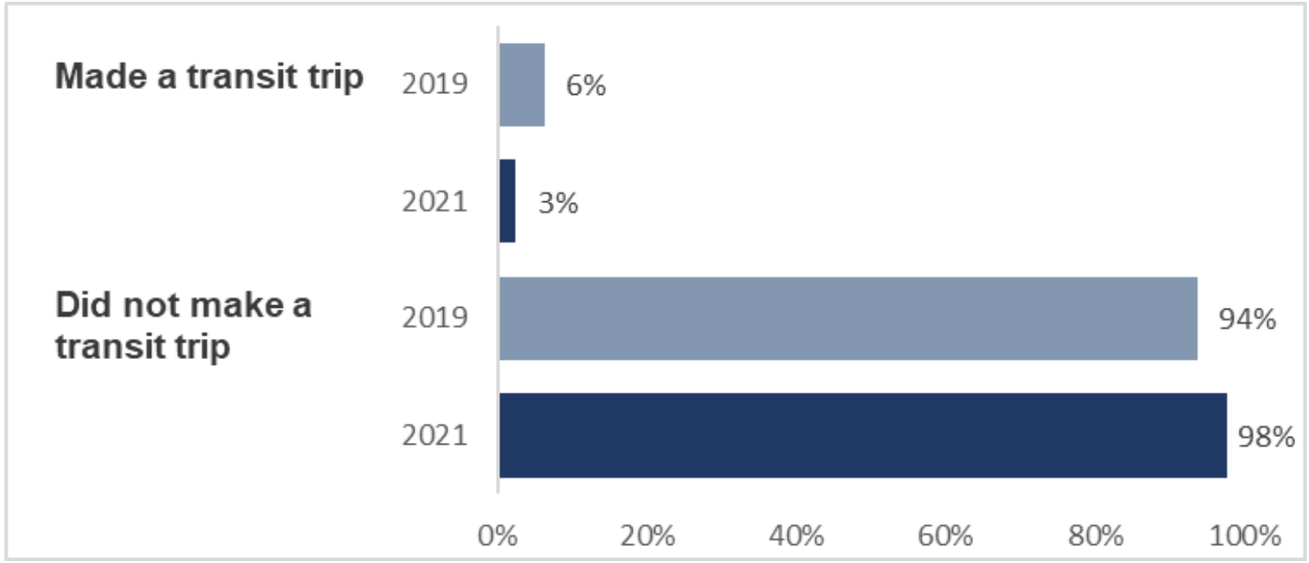
```
day_design = svydesign(
  ids = ~ day_id,
  data = day,
  weights = ~ day_weight,
  strata = ~ sample_segment)

# important to remember to filter day table to only
# weighted days to avoid division by 0
svymean(~ I(weighted_num_trips / day_weight), day_design)
```

```
                                                mean      SE
I(weighted_num_trips/day_weight) 3.57721485138 0.05109
```

Analyzing behavioral changes over time

- While efforts were made to minimize changes between the 2018-2019 and 2021-2022 datasets, there are differences.
- The respective dataset guides and codebooks are invaluable for data users.
- Here's an example of analyzing behavioral changes over time, evaluating the percentage of people who used transit on an average weekday
- There is a substantial drop in transit usage due to the COVID 19 pandemic



Analyzing behavioral changes over time

Code Example

Transit Usage 2021

```
person_2021[,
  made_transit_trip := fcase(
    # made a transit trip during a weekday
    person_id %in% trip_2021[
      mode_type == 13 &
      trip_weight > 0, person_id], 1,
    default = 0)]

transit_trips_2021 = person_2021[,
  .(.N,
    wtd_N = round(sum(person_weight), 0)),
  made_transit_trip]

transit_trips_2021[, prop_N := round(N / sum(N), 3)]
transit_trips_2021[,
  wtd_prop_N := round(wtd_N / sum(wtd_N), 3)]

transit_trips_2021
```

```
> transit_trips_2021
   made_transit_trip     N   wtd_N prop_N wtd_prop_N
1:                 0 15064 3490549  0.953      0.975
2:                 1   741   90903  0.047      0.025
```

Transit Usage 2019

```
person_2019[,
  made_transit_trip := fcase(
    # made a transit trip during a weekday
    person_id %in% trip_2019[
      mode_type %in% c(1, 3, 4) &
      trip_weight > 0, person_id], 1,
    default = 0)]

transit_trips_2019 = person_2019[,
  .(.N,
    wtd_N = round(sum(person_weight), 0)),
  made_transit_trip]

transit_trips_2019[, prop_N := round(N / sum(N), 3)]
transit_trips_2019[,
  wtd_prop_N := round(wtd_N / sum(wtd_N), 3)]

transit_trips_2019
```

```
> transit_trips_2019
   made_transit_trip     N   wtd_N prop_N wtd_prop_N
1:                 0 14688 3806185  0.909      0.937
2:                 1  1464  254259  0.091      0.063
```

New Analysis Using Multiday Smartphone Data

- **Person Miles Traveled: rMove App vs. Web**
- **Day-to-Day Variability Within People**

Person Miles Traveled

- HTS are useful ways to collect information about Person Miles Traveled (PMT).
- Similar to vehicle miles (VMT) in concept, PMT can help data users understand which types of people are traveling different distances (and for what reasons).

Person Miles Traveled: By Mode

	Path Distance (Median)	Straight Line Distance (Median)	Median Ratio
Walk	0.4	0.3	1.3
Bicycle	1.9	1.3	1.4
Bike-share	1.8	1.3	1.3
Scooter-Share	2.1	0.5	2.1
Taxi	9	5.7	1.4
Smartphone-app ridehailing service	4.8	3.4	1.3
Other	2.4	0.9	1.4
Vehicle	4	2.8	1.3
Carshare	3.1	2.3	1.3
School bus	2.4	1.9	1.4
Shuttle	3.7	2.4	1.5
Ferry	3.7	1.3	1.4
Transit	2.6	2	1.2
Long distance passenger mode	688.8	680.5	1

- Lots of variation in trip distance across mode types
- Path distance is consistently higher across all modes except for long-distance modes

Person Miles Traveled: By Purpose

	Path Distance (Median)	Straight Line Distance (Median)	Median Ratio
Home	3.1	2.2	1.3
Work	4.9	3.5	1.3
Work-related	4	2.7	1.3
School	2.5	1.7	1.3
School-related	0.7	0.5	1.3
Escort	3.8	2.7	1.3
Shopping	2.4	1.7	1.3
Meal	2.4	1.7	1.3
Social/Recreation	1.7	1	1.3
Errand	3.4	2.5	1.3
Change mode	1.4	1.1	1.2
Overnight	4.8	3.4	1.3

- Lots of variation across trip purposes
- The ratio of path to straight line distance is consistent

Person Miles Traveled: By Employment Status

	Path Distance (Median)	Straight Line Distance (Median)	Median Ratio
Employed full-time (paid)	2.9	2.0	1.3
Employed part-time (paid)	2.7	1.9	1.3
Self-employed	2.8	2.0	1.3
Not employed and not looking for work	2.8	1.9	1.3
Unemployed and looking for work	2.2	1.5	1.3
Unpaid volunteer or intern	3.0	2.1	1.3
Furloughed	2.9	1.9	1.3

- Employed participants traveled the farthest
- Unemployed participants traveled the least distance
- Difference between path and straight-line distance is consistent

Person Miles Traveled: By Day of Week

	Path Distance (Median)	Straight Line Distance (Median)	Median Ratio
Monday	2.8	2.0	1.3
Tuesday	2.7	1.9	1.3
Wednesday	2.8	2.0	1.3
Thursday	2.8	2.0	1.3
Friday	2.7	1.9	1.3
Saturday	2.8	2.0	1.3
Sunday	2.8	2.0	1.3

- Very little variation across weekdays

Person Miles Traveled

Code Example

```
# calculate straight line distance in KMs
trip[,
  strt_distance := get_distance_meters(
    c(o_lon, o_lat), c(d_lon, d_lat)) * 0.000621371]

# get mean and median path/straight distance for each
# mode_type, purpose, and employment status
dt_mode = trip[,
  .(mean_path = mean(distance),
    mean_strt = mean(strt_distance),
    median_path = median(distance),
    median_strt = median(strt_distance),
    mean_ratio = mean(distance / strt_distance),
    median_ratio = median(distance / strt_distance)),
  keyby = mode_type]
```

```
# function for calculating straight-line distance
# using the haversine formula
get_distance_meters = function(
  location_1,
  location_2,
  radius = 6378137) {

  location_1 = matrix(location_1, ncol = 2)
  location_2 = matrix(location_2, ncol = 2)
  lon_1 = location_1[, 1] * pi/180
  lon_2 = location_2[, 1] * pi/180
  lat_1 = location_1[, 2] * pi/180
  lat_2 = location_2[, 2] * pi/180
  dLat = lat_2 - lat_1
  dLon = lon_2 - lon_1
  a = sin(dLat / 2) ^ 2 +
    cos(lat_1) * cos(lat_2) * sin(dLon / 2) ^ 2
  a = pmin(a, 1)
  dist = 2 * atan2(sqrt(a), sqrt(1 - a)) * radius
  return(dist)
}
```

Day-to-day Variability within People

- While multi-day data collection increases the volume of data collected per household, it also helps capture a wider variety of behaviors.
- To exemplify these behaviors, data users can randomly select one or more days per each person and analyze the differences.
- For each person in the dataset, we can sample 1, 2, 3, ..., 7 days and then look at the additional information that sampling more days provides along a number of key metrics.
- For this analysis, we filtered to households who completed all 7 days of their travel period.

Day-to-day Variability within People

- The table below shows variation within rMove respondents on one to seven randomly selected days.
- Increasing the number of days sampled marginally increases the number of non-walk, non-vehicle trip modes included in the dataset
- Increasing the number of days sampled significantly increases the number of trip purposes other than home, school, or work included in the dataset

	1 DAY	2 DAYS	3 DAYS	4 DAYS	5 DAYS	6 DAYS	7 DAYS
Average # of distinct modes	1.3	1.5	1.6	1.7	1.8	1.8	1.9
Average # of walk/car modes	1.2	1.3	1.4	1.5	1.5	1.5	1.6
Average # of other modes	0.1	0.2	0.2	0.2	0.3	0.3	0.3
Average # of distinct purposes	3.2	4.1	4.8	5.2	5.6	5.9	6.1
Average # of habitual purposes	1.3	1.4	1.5	1.6	1.6	1.6	1.7
Average # of other purposes	1.9	2.7	3.3	3.7	4	4.3	4.5

Day-to-day Variability within People

- Additional days of data have little impact on trip rate and the likelihood that no travel would be reported on that day
- Total daily distance and total trip duration drops over the course of 7 days

	1 DAY	2 DAYS	3 DAYS	4 DAYS	5 DAYS	6 DAYS	7 DAYS
Trip rate	5.2	4.7	4.5	4.4	4.4	4.4	4.4
Percent of days with no travel	4.6%	4.7%	4.6%	4.6%	4.6%	4.6%	4.6%
Mean trip distance	8.1	8.8	8.7	8.5	8.5	8.4	8.4
Mean total daily distance	40.6	38.8	37	35.8	35.9	35.8	35.7
Mean trip duration	22.2	20.7	20.1	19.6	19.9	19.7	19.8
Mean total trip duration	101.2	88.7	85.4	82.6	82.6	82.1	81.9

Day-to-Day Variability Analysis

Code Example

```
# Randomly sample the days of the week-----------------
day[, sample := sample(.N, .N), by = 'person_id']

result_list = list()
for (n_day in 1:7) {

  trip_sample = trip1[sample <= n_day]

  # person day summary
  person_summary = trip_sample[,
    .(n_trips = sum(!is.na(distance)) / n_day,
      n_mode = uniqueN(mode_type[!is.na(mode_type) & mode_type != 995]),
      n_common_mode = uniqueN(mode_type[!is.na(mode_type) & mode_type %in% c(1, 8)]),
      n_uncommon_mode = uniqueN(mode_type[!is.na(mode_type) & !mode_type %in% c(1, 8, 995)]),
      n_purpose = uniqueN(d_purpose_category[!is.na(d_purpose_category) & !d_purpose_category %in% c(-1, 995)])),
    by = .(person_id)]

  day_means = person_summary[,
    .(n_mode = mean(n_mode),
      n_walk_car_mode = mean(n_common_mode),
      n_other_mode = mean(n_uncommon_mode),
      n_purpose = mean(n_purpose),
      trip_rate = mean(n_trips))]
}

means_dt = rbindlist(result_list, idcol = 'n_day')}
```

Session 4: Weighting Methodology

Weighting Overview

Following data collection, survey data can be weighted to match the population across key sociodemographic dimensions.

- Weighting ensures the survey data aligns with census totals across key demographics.
- Weighting can help correct biases that may be present in the sample of people who completed the survey (e.g., from surveying too many persons of a particular profile or from a particular area).
- The weighting process results in a new variable that reflects how many households (or persons, days, or trips) that survey record represents in the region.

    For example, a household weight of 250 implies the survey record represents 250 similar households in the region.

Weighting Process Overview

1. **Initial Expansion:** Calculating an ‘initial weight’ based on the probability of selection – essentially reversing the sample plan, providing higher initial weights to areas where less sampling occurred.
2. **Reweighting to account for non-response bias**: Performing a fitting routine to match several key household and person dimensions to ensure the weighted data accurately represent the entire survey region (and reduce sampling biases).
3. **Creating day-level weights to account for multi-day survey data:** Adjusting the day-level and trip-level data to account for the fact that smartphone respondents provided multi-day travel diaries, while online and call center respondents provided a single-day travel diary (this is the “multi-day adjustment”).
4. **Adjusting for non-response bias in day-pattern and trip rates:** Adjusting the trip-level weights by data collection method (smartphone, online, call center) to account for reporting biases. These adjustments help make the day and trip-level data more consistent and increase the accuracy of trip rates across survey participation methods.

Differences in weighting methods between Wave 1 and Wave 2

Unlike the Wave 1 2018-2019 survey sample, the Wave 2 2021-2022 sample includes a substantial set of data collected via supplemental non-probability methods in addition to the traditional addressed-based sampling (ABS) approach.

- Outreach through community-based organizations
- Metro Transit's Transit Assistance Program (TAP) email and text lists.

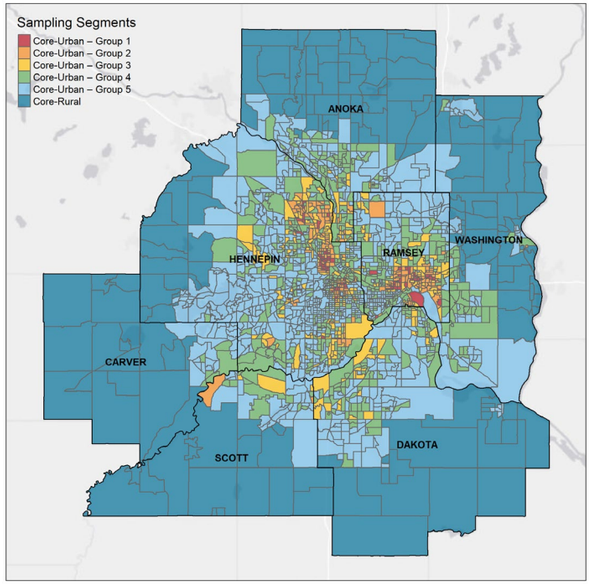
The goal was to use supplementary sampling methods to increase the proportion of hard-to-survey households in the final dataset.

The inclusion of non-probability sample primarily impacts the calculation of the initial expansion factors. The primary decision point is how we will allocate the population to the ABS and the supplemental sample and how that is implemented.

Initial Expansion

Calculating an ‘initial weight’ based on the probability of selection. This essentially ‘reverses’ the sample plan, providing higher initial weights to areas where less sampling occurred.

- Consider the days that will be weighted – just weekdays or weekdays and weekends.
- Selection of respondents for weighting: households that had at least one complete and valid travel day are included in weighting.
- To calculate the initial expansion factors for each sample segment, the ratio of population household counts to sampled households is calculated.
- One important consideration for this study is how to incorporate the supplemental sample



Initial Expansion

Typically, the initial expansion weights follow a very simple formula:

IW(s) = H / R(s)

Initial Expansion Weights for 2019 TBI

SAMPLE STRATA (S)	HOUSEHOLDS (H)	COMPLETES (R)	INITIAL WEIGHT (IW = H/R)
Core-Rural	114,240	557	205.10
Core-Urban	879,673	4,530	194.19
Rural Ring	239,753	1,215	197.33
Hard-to-Survey	198,713	1,446	137.42
Total	1,432,379	7,748	

Initial Expansion

To incorporate the supplemental sample, we adjust the Initial Weight formula:

$IW_{abs}(s) = H * (1- C(s)) / R_{abs}(s)$

$IW_{sup}(s) = H * C(s) / R_{sup}(s)$

The introduction of C effectively apportions the population used for the ABS and supplemental sample

2021 TBI

SAMPLE STRATA (S)	HOUSEHOLDS (H)	COMPLETES (R_{ABS})	COMPLETES (R_{SUP})	C(S)	INITIAL WEIGHT (IW_{ABS})	INITIAL WEIGHT (IW_{SUP})
Core-Urban 1	33,659	544	47	0.2	49.50	143.23
Core-Urban 2	74,362	755	111	0.2	78.79	133.99
Core-Urban 3	133,237	948	159	0.1	126.49	83.80
Core-Urban 4	357,150	2,278	242	0.1	141.10	147.58
Core-Urban 5	472,450	1,462	204	0.025	315.07	579.90
Core-Rural	116,565	387	9	0.025	293.67	323.79
Ring Bgs	257,959	806	0	0	320.05	NA
Total	1,445,382	7,180	772			

Initial Expansion

To select C(s):

- **We acknowledge that the selection of C is arbitrary and based on professional judgement**
- **Probability sampling/ABS is the gold standard for sampling**
- **Setting C at the strata level allows for more targeted control**
  - **We want to set C higher when the ABS sample does poorly representing the population**
  - **We want to set C lower when the ABS sample does better at representing the population**
- **Do not want C to be larger than 0.20 for any sample strata and no bigger than 0.10 across the entire dataset**

Percent of HHs less than $50k

SAMPLE STRATA (S)	ABS	SUPP	ACS
Core-Urban 1	59%	83%	68%
Core-Urban 2	46%	76%	51%
Core-Urban 3	41%	68%	42%
Core-Urban 4	30%	72%	32%
Core-Urban 5	23%	53%	23%

Percent of Non-White HHs

SAMPLE STRATA (S)	ABS	SUPP	ACS
Core-Urban 1	36%	30%	81%
Core-Urban 2	26%	49%	65%
Core-Urban 3	17%	27%	44%
Core-Urban 4	12%	28%	25%
Core-Urban 5	8%	16%	9%

C
0.2
0.2
0.1
0.1
0.025

Reweighting for Non-response bias

Performing a fitting routine to match several key household and person dimensions to ensure the weighted data accurately represent the entire survey region (and reduce sampling biases).

For the purpose of the fitting routine:

- Geographic zones are created to increase sample sizes in the weighting geography to have a sufficient number of individuals with certain characteristics to improve the weighting fit.

Household and person-level characteristics are then chosen as targets for weighting the survey data to the target data.

- Household characteristics: household size, income, number of workers, number of household vehicles, age of head of household, presence of children
- Person characteristics: gender, age, employment status, student status, race, ethnicity, education attainment
- 2019 1-year PUMS data are used to define target controls

Missing values are imputed where necessary for income gender, race, and ethnicity.

Reweighting for Non-response bias

We used an entropy maximization-based list balancing routine to fit the final set of household and person-level weights

The entropy maximization (EM) algorithm is an alternative to the more common iterative proportional fit or iterative proportional updating algorithms

- We used the EM algorithm as implemented in PopulationSim
- https://activitysim.github.io/populationsim/

At its core, the EM algorithm uses the 'relative entropy' as the objective function to be maximized

- The base entropy is defined using the initial weights
- It uses the relative entropy function to identify weights that introduce the least amount of new information
- The EM algorithm maintains the distribution of initial weights while matching marginal controls

Reweighting for Non-response bias

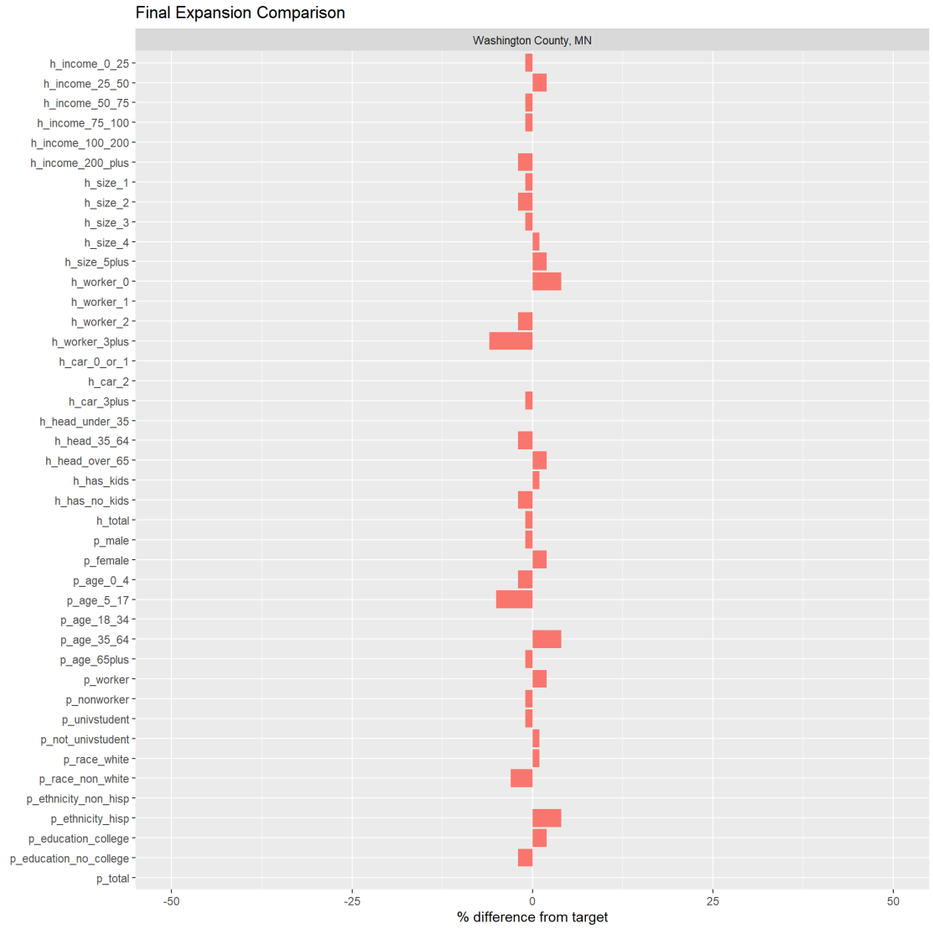
The weighting process used the following geographies:

- **Anoka County, MN** (PUMAs: 01101, 01102, 01103)
- **Washington County, MN** (PUMAs: 01201, 01202)
- **Ramsey County, MN – St. Paul (**PUMAs: 01303, 01304)
- **Ramsey County, MN – Other** (PUMAs: 01301, 01302)
- **Hennepin County, MN – Minneapolis** (PUMAs: 01405, 01406, 01407)
- **Hennepin, County MN – Other** (PUMAs: 01401, 01402, 01403, 01404, 01408, 01409, 01410)
- **Dakota County, MN** (PUMAs: 01501, 01502, 01503)
- **Scott & Carver Counties, MN** (PUMAs: 01600, 01700)
- **Chisago & Isanti Counties, MN** (PUMAs: 00600)
- **Sherburne, Wright, McLeod, & Sibley Counties, MN** (PUMAs: 01000, 01800, 01900)
- **Goodhue, Rice, & Le Sueur Counties, MN** (PUMAs: 02300)
- **Pierce, Polk, & St. Croix Counties, WI** (PUMAs: 00700, 55101, 55102)

Reweighting for Non-response bias

We compare the fitted weights against the targets.

- The targets are based on survey data so a perfect match is not necessary or desired
- Constraints are applied to the weights to prevent them from varying too much from the initial expansion factors
- The goal is to reduce variance introduced during weighting



Multi-day, Day pattern, and Trip Rate Adjustments

Previous surveys have shown that trip rates from smartphone survey data are 15-20% higher than those from online or call center survey data.

There are three main reasons for this:

1. Smartphone-owning households have different socio-demographic characteristics than non-smartphone households and tend to make more trips.
2. Online and call center-based data have twice as many “stay at home” days with no reported trips
3. Even on days with one or more reported trips, there are more trips per day reported on average in the smartphone-based data than in the online- or call center-based data.

A two-stage approach is typically applied to account for these differences:

1. Weights are created at the person-day level to account for biases in 1) day pattern types and 2) participants with multiple travel days.
2. Weights are created at the trip-level to account for biases in trip rate.

Day-pattern Adjustment

Adjustment of the day-level weights to account for the fact that those who participated online or through the call center had a higher proportion of no travel or stay at home days.

Most activity-based models include a model component to predict the day-pattern type:

- Stay at home
- Make mandatory (work or school) trips (and possibly other trips)
- Make non-mandatory trips only

"Stay at home" cases have no trip records in the data, so the correction cannot be made by factoring weights at the trip level. Adjustments to the day weights correct for biases that distort the day-pattern type.

RSG typically estimates a multinomial choice model and applies it at the person day-level to calculate the probabilities of each of the three day-pattern alternatives with biases removed.

Example day pattern Choice model

ALTERNATIVE	DESCRIPTION	ESTIMATE	T-STAT
Makes mandatory trips	Intercept	-2.408	-18.3
	Online diary data	**-0.107**	**-2.0**
	Call center diary data	**-0.599**	**-4.3**
	No vehicles in household	-0.019	-0.2
	Income 25k-50k	0.182	1.7
	Income 50k-75k	0.091	0.8
	Income 75k-100k	-0.201	-1.8
	Income 100k-150k	-0.082	-0.8
	Income > 150k	-0.311	-2.8
	Prefer not to answer income	-0.283	-2.5
	AGE < 35 YEARS	0.265	2.5
	Age between 35-65 years	0.610	8.1
	Employed full/part/self	3.241	36.7
	Full or part-time student	0.396	5.1
	Online diary data x Age	**-0.251**	**-2.6**
	Call center diary data x Age	**-0.131**	**-0.3**
Makes non-mandatory trips only	Intercept	1.190	13.6
	Online diary data	**-0.383**	**-7.7**
	Call center diary data	**-0.544**	**-6.3**
	No vehicles in household	-0.531	-6.0
	Income 25k-50k	-0.119	-1.4
	Income 50k-75k	0.109	1.2
	Income 75k-100k	0.124	1.4
	Income 100k-150k	-0.030	-0.4
	Income > 150k	0.140	1.5
	Prefer not to answer income	-0.105	-1.2
	AGE < 35 YEARS	-0.332	-3.7
	Age between 35-65 years	-0.077	-1.5
	Employed full/part/self	0.096	2.0
	Full or part-time student	-0.081	-1.1
	Online diary data x Age	**-0.405**	**-4.3**
	Call center diary data x Age	**-1.585**	**-2.9**

The ‘stay at home’ alternative was used as the base in this model so all coefficients are relative to this.

Bias capturing variables

To get unbiased estimates of day pattern usage, these bias variables are set to 0.

Day pattern Model applied

	WITH BIAS			BIAS REMOVED		
HOUSEHOLD GROUP TYPE	NO-TRAVEL DAYS	MANDATORY TRIP DAYS	NON-MANDATORY TRIP DAYS	NO-TRAVEL DAYS	MANDATORY TRIP DAYS	NON-MANDATORY TRIP DAYS
Online Diary	**22%**	37%	42%	**16%**	34%	50%
Call Center Diary	**35%**	12%	53%	**23%**	14%	63%
rMove Diary	16%	37%	47%	16%	37%	47%

Before and after

Code

Estimation Code

```
model = multinom(
  day_trip_class ~
    online_data +
    call_center_data +
    zero_vehicle +
    income_aggregate +
    age_under_35 + age_35_65 +
    employed +
    is_student +
    age_under_35 * online_data +
    age_under_35 * call_center_data,
  data = fit_dt,
  weights = fit_dt[, estimation_weight],
  model = TRUE,
  Hess = TRUE)
```

Application Code

```
model_coefs = coef(model)

model_coefs[, 'online_data'] = 0
model_coefs[, 'online_data:age_under_35'] = 0
model_coefs[, 'call_center_data '] = 0
model_coefs[, 'call_center_data :age_under_35'] = 0

model_rhs = ~ online_data + call_center_data +
    zero_vehicle +
    income_aggregate +
    age_under_35 + age_35_65 +
    income_aggregate +
    employed +
    is_student +
    age_under_35 * online_data + age_under_35 * call_center_data

V_none = 0
V_mandatory = model.matrix(model_rhs, fit_dt) %*% t(t(model_coefs['mandatory’, ]))
V_non_mand = model.matrix(model_rhs, fit_dt) %*% t(t(model_coefs['non-mandatory’, ]))
V = cbind(V_none = V_none, V_mand = V_mandatory, V_non_mand = V_non_mand)
colnames(V) = c('none', 'mandatory', 'non-mandatory’)

P = exp(V)
P = P/rowSums(P)
```

Multi-day Adjustment

Adjustment of the day-level weights to account for the fact that smartphone respondents participated for multiple travel days while online and call center respondents participated for a single day.

- A day weight is equal to the person weight divided by the number of complete travel days.
- The resulting day-weight when applied will result in households with multiple travel days having their travel “averaged” to equal one day allowing for analyses at the “average” day-level.

Trip-rate Adjustment

Adjusting the trip-level weights by data collection method to account for reporting biases. These adjustments help make the data more consistent across survey modes and support more accurate trip rates.

Adjusting the weights for day-pattern biases reduces the discrepancy in trip rates between methods but does not eliminate it altogether.

The difference in trip rates tends to be higher for:

- Non-mandatory trips than for mandatory trips, as respondents are less likely to omit their work and school trips in diary-based data.
- Non-home-based trips, since diary respondents often tend to omit intermediate stops on multi-stop tours.

For each trip type, a Poisson regression model is typically estimated where the dependent variable is the number of trips of that type in the person day. This model is then applied back to the data with biases removed providing a trip rate adjustment factor that is incorporated in the trip weight.

Trip rate adjustment models

Home-based Work

DESCRIPTION	ESTIMATE	T-STATISTIC
Intercept	-3.139	-33.9
Online diary data	**-0.593**	**-24.8**
Call center diary data	**-0.838**	**-7.7**
Under age 25	0.492	7.1
Age 25 to 45	0.328	5.7
Age 45 to 65	0.439	7.4
Employed full-time	2.433	31.8
Employed part-time	2.437	30.6
Self-employed	1.741	15.8
Has bachelor's degree	-0.342	-12.1
Has masters/PhD	-0.296	-9.6
Is student	-0.001	-0.0
Work location varies	0.113	2.8
Works 2+ Jobs	0.103	2.7
Lives in single family home	0.135	4.6

Home-based Other

DESCRIPTION	ESTIMATE	T-STATISTIC
Intercept	0.439	13.8
Online diary data	**-0.811**	**-64.4**
Call center diary data	**-0.803**	**-22.5**
Under age 25	-0.163	-4.8
Age 25 to 45	0.066	3.2
Age 45 to 65	0.042	2.0
Employed full-time	-0.187	-10.3
Employed part-time	0.012	0.5
Self-employed	-0.017	-0.5
Has bachelor's degree	0.337	22.3
Has masters/PhD	0.455	28.7
Is student	0.066	2.6
Work location varies	0.050	2.0
Works 2+ Jobs	-0.135	-5.5
Lives in single family home	0.086	5.9

Bias capturing variables

Bias is significant for HBW trips. Online respondents are better at remembering these types of trips than call center respondents.

Bias is significant for HBO trips. Online respondents are less good at remembering these types of trips.

Trip rate adjustments applied

DIARY METHOD	HOME-BASED WORK	HOME-BASED OTHER	NON-HOME-BASED WORK	NON-HOME-BASED OTHER
Smartphone participant	1.00	1.00	1.00	1.00
Online diary	1.81	2.00	1.03	1.03
Call center diary	2.00	2.00	1.13	1.00

Trip rate adjustment factors

The trip weights for online home-based work trips are multiplied by 1.81

Code

Estimation Code

```
formula_rhs =
  ~ online_data +
    call_center_data +
    age_under_25 +
    age_25_45 +
    age_45_65 +
    employed_ft +
    employed_pt +
    employed_self +
    bachelors +
    graduate_degree +
    is_student +
    work_loc_varies +
    two_plus_jobs +
    sf_home

#----------------------------
# home based work model
#----------------------------
model_hbw = glm(
  update(formula_rhs, num_trips_hbw ~ .),
  data = fit_dt,
  weights = fit_dt[, estimation_weight],
  family = "poisson")
```

Application Code

```
fit_dt[,
  num_trips_hbw_pred1 := predict(model_hbw, newdata = fit_dt, type = 'response')]

model_hbw_bias_removed = model_hbw

model_hbw_bias_removed$coefficients['online_data'] = 0
model_hbw_bias_removed$coefficients['call_center_data'] = 0

fit_dt[,
  num_trips_hbw_pred2 := predict(
    model_hbw_bias_removed,
    newdata = fit_dt,
    type = "response")]

fit_dt[, hbw_trip_rate_factor := 1]

fit_dt[
  survey_mode %in% c(1, 4, 7),  # subsetting to online respondents
  hbw_trip_rate_factor_online := num_trips_hbw_pred2 / num_trips_hbw_pred1]

fit_dt[
  survey_mode %in% c(2, 5, 8),  # subsetting to call center respondents
  hbw_trip_rate_factor_call_center := num_trips_hbw_pred2 / num_trips_hbw_pred1]
```

FINAL WEIGHTS AND RECOMMENDED USE

THE FINAL WEIGHTS PROVIDED WITH THE DATASET ARE:

hh_weight: The resulting weights from expansion to the target data. Should be used for household-level and vehicle-level analyses.

person_weight: Equivalent to hh_weight. Should be used for person-level analyses.

day_weight: The same as person_weight but divided by the number of complete days of data for each household. Should be used for household-day and person-day analyses – represents an average day.

trip_weight: Same as day_weight except when adjustments factors are applied from the trip correction process. Should be used for trip-related analysis.

Contacts

www.rsginc.com

JOANN LYNCH
Project Manager
Joann.Lynch@rsginc.com

ILONA REGAN
Lead Analyst
ilona.regan@rsginc.com

JEFF DUMONT
Senior Data Scientist
Jeff.Dumont@rsginc.com

MICHELLE LEE
Principal Investigator
Michelle.Lee@rsginc.com