CE 264 Problem Set 4: Forecasting

Chenglong Li (3032129387)
Junzhe Shi (3033030938)
Franklin Zhao (3033030808)

6 Mar. 2018

Part 1 Forecasting

Task 1:

First let's estimate the provided model specification. Results are shown in Table 1 and 2.

Table 1: Estimation result (part 1)

Dep. Variable:	choice	**No. Observations:**	9,999
Model:	Multinomial Logit Model	**Df Residuals:**	9,984
Method:	MLE	**Df Model:**	15
Date:	Mon, 05 Mar 2018	**Pseudo R-squ.:**	0.372
Time:	23:43:55	**Pseudo R-bar-squ.:**	0.371
AIC:	16,071.616	**Log-Likelihood:**	-8,020.808
BIC:	16,179.770	**LL-Null:**	-12,766.793

Table 2: Estimation result (part 2)

Variables	**coef**	**Std err**	**z**	**P > |z|**	**[0.025**	**0.975]**
ASC SR	-2.1158	0.049	-43.217	0.000	-2.212	-2.020
ASC Walk	-2.5376	0.187	-13.546	0.000	-2.905	-2.170
ASC Bike	-3.4882	0.185	-18.852	0.000	-3.851	-3.126
ASC WT	1.5572	0.154	10.133	0.000	1.256	1.858
ASC DT	-0.8970	0.183	-4.893	0.000	-1.256	-0.538
In-Vehicle Travel Time, units:hrs (DA, SR, WT)	-1.9053	0.094	-20.306	0.000	-2.089	-1.721
Bike Time, units:hrs (Bike)	-4.7177	0.361	-13.058	0.000	-5.426	-4.010
Walk Time, units:hrs (Walk)	-1.1014	0.122	-9.012	0.000	-1.341	-0.862
In-Vehicle Travel Time, units:hrs, (DT)	-1.2238	0.121	-10.149	0.000	-1.460	-0.987
Walk Time, units:hrs, (WT)	-3.1670	0.227	-13.968	0.000	-3.611	-2.723
Walk Time, units:hrs, (DT)	-5.2712	0.384	-13.732	0.000	-6.024	-4.519
Waiting Time, units:hrs, (WT and DT)	-2.6341	0.225	-11.705	0.000	-3.075	-2.193
Cost: Under $2	-1.2832	0.064	-20.084	0.000	-1.408	-1.158
Cost: (2 - 7)$	-0.3359	0.019	-17.883	0.000	-0.373	-0.299
Cost: Above $7	-0.0781	0.010	-7.976	0.000	-0.097	-0.059

Question (a):

In this model, the specification takes into account the alternative specific constants (ASC), in-vehicle travel time, bike time, walk time, waiting time and cost. ASCs include shared ride, walk, bike, walk to transit, and drive to transit. The coefficient for in-vehicle travel time is alternative specific for drive-transit mode, while generic for drive alone, shared ride, and walk to transit mode. Bike time is only considered in bike mode. For walk time, the coefficient is alternative specific for walk, walk to transit, and drive to transit mode. For waiting time, the coefficient is generic for walk to transit and drive to transit mode. For cost, the coefficient is alternative specific, and divided into three parts based on two thresholds \$2 and \$7.

Question (b):

The estimation result makes sense since all these selected variables should have some impact on the mode choice based on our intuition. Apart from ASCs, the sign (i.e., positive or negative) of the coefficients also makes sense since we all notice that time and cost both have negative impact on the mode choice (we all prefer the mode with less time and cost). Also, the P-values are all close to 0, indicating that we should reject the null hypothesis and the selected variables are significant.

Question (c):

As we discussed in the previous question, the specification is resonable. Now let's take a look at the travel cost. Travel cost is divided into three parts: under \$2, \$2-7, and above \$7. The thresholds \$2 and \$7 would capture the sensitivity of different levels of cost. As we can see from the result, it is interesting that lower cost range has actually more impact on the mode choice. We can assume that the lower cost, the more sensitive a decision maker will be. Such variables may also reflect the impact of income, since those thresholds may also be considered as income level.

Question (d):

We have discussed about this. If we specified the model ourselves, we would take into account the access time and egree time since these variables also reflect the "time consuming" for the modes, which we think is important.

Task 2:

Question (a):

The forecasting results are shown in Table 3 and Figure 1.

Table 3: Forecasting results (probabilities)

Toll($)	Drive alone	Shared ride	Walk	Bike	Walk transit	Drive transit
0	0.631724	0.224905	0.017962	0.014922	0.070864	0.039549
1	0.625117	0.220704	0.019581	0.015552	0.078338	0.040634
2	0.621775	0.218869	0.020270	0.015827	0.081754	0.041431
3	0.619725	0.217952	0.020531	0.015984	0.083658	0.042077
4	0.617890	0.217331	0.020741	0.016118	0.085213	0.042632
5	0.616337	0.216945	0.020907	0.016222	0.086398	0.043116
6	0.615252	0.216656	0.021029	0.016280	0.087142	0.043567
7	0.614468	0.216474	0.021077	0.016307	0.087597	0.044004
8	0.613772	0.216327	0.021099	0.016326	0.087968	0.044433
9	0.613098	0.216185	0.021120	0.016345	0.088324	0.044855
10	0.612445	0.216046	0.021139	0.016362	0.088664	0.045270

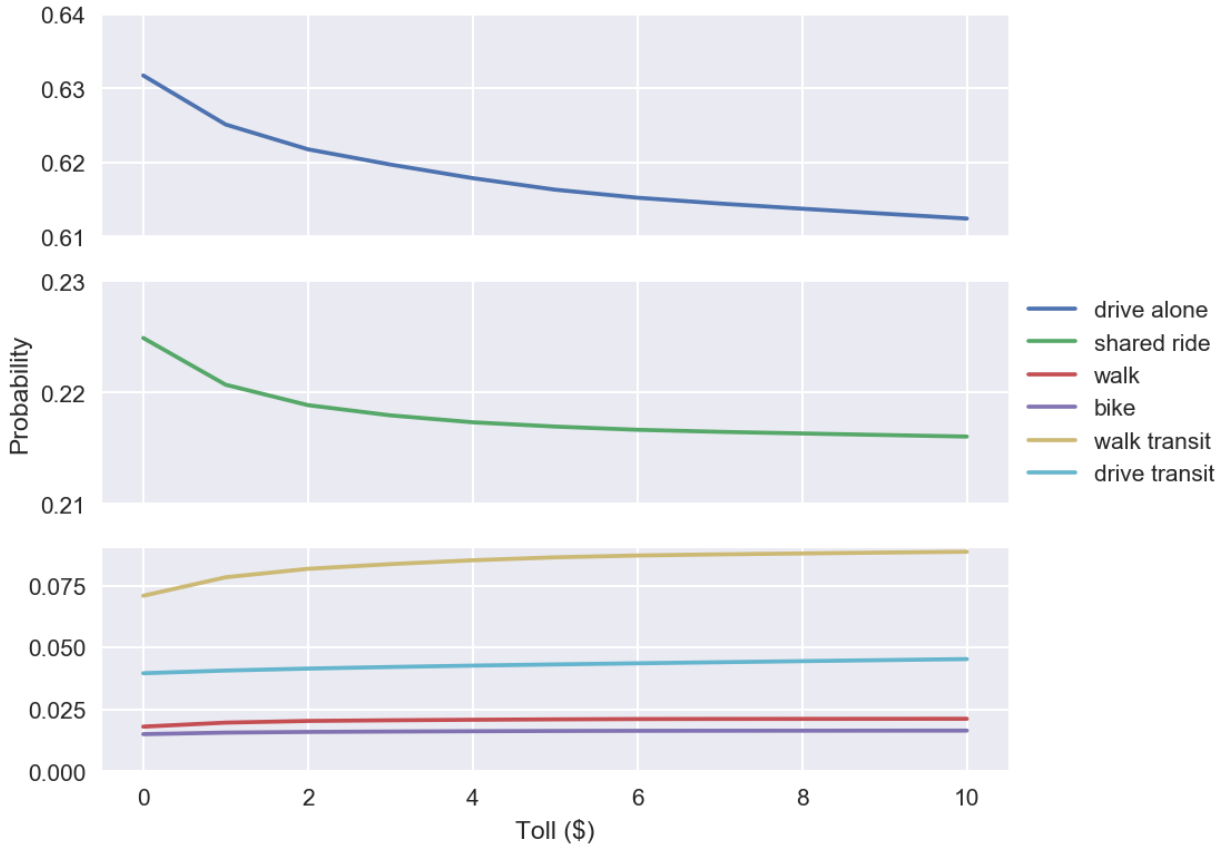

Fig. 1: Forecasting results visualization

Question (b):

- The influence of toll on drive to transit (DT) is ignored here. On the one hand, this part has very little influence, as people usually choose the public transportation station closest to their origin. Therefore it is very unlikely that they will have to pass a toll point to access a station. On the other hand, this part is hard to investigate, as we do not have very adequate information about the destination of persons driving route.
- For shared driving, only two persons share a ride, instead of three or four. This will be used in task 3, as a conservative consideration for estimating the reduction of CO_2 emission.
- The assumptions stated in the problem statement.

Question (c):

As we can see from Figure 1, as toll increases, the probability of drive alone and share ride decreases, since as the congestion status increases, people tend to switch driving to other modes. Hence, it is also resonable to notice that the probablities of choosing other modes are increasing, including drive-transit since such a mode could ameliorate the effect of congestion. However, even though driving modes are decreasing and other modes are increasing, the probabilities of driving modes are still much higher than others, at least in our toll domain. This is probably because the congestion level is not high enough to affect the mode choice of decision makers in this scenario.

Task 3:

Question (a):

The estimation results are shown in Table 4 and Figure 2.

Table 4: Estimation results

Toll ($)	**Emission (lbs)**	**Reduction (lbs)**	**Percentage (%)**
0	16770830	0	0.00
1	16701468	69362	0.41
2	16656403	114427	0.68
3	16619952	150878	0.90
4	16586270	184560	1.10
5	16556207	214623	1.28
6	16530295	240535	1.43
7	16506326	264504	1.58
8	16483230	287600	1.71
9	16460507	310323	1.85
10	16438269	332561	1.98

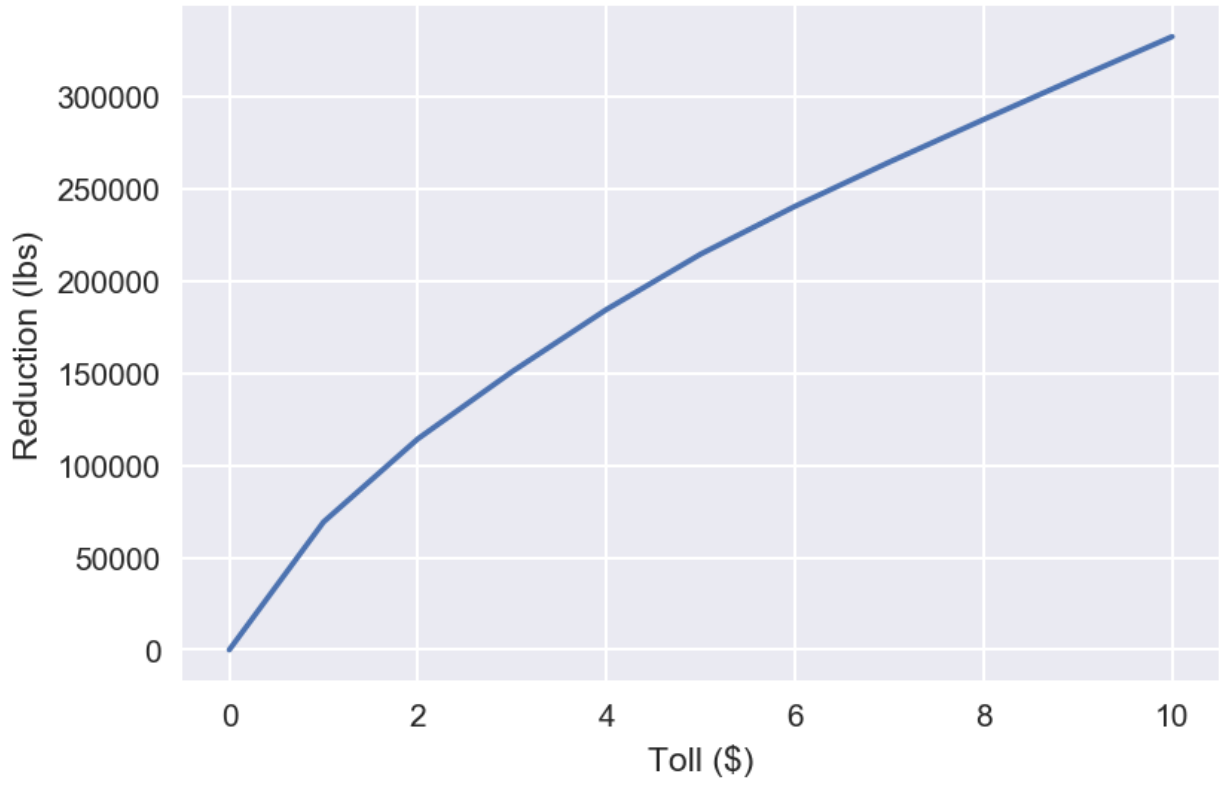

Fig. 2: Forecasting results visualization

Question (b):

The equation of reduction with respect to toll ($i) is:

$$Reduction_i = Emission_0 - Emission_i \tag{1}$$

where $Emission_0$ means no toll. This equation is applicable for both total and individual reduction.

The CO_2 emission of an individual with respect to toll ($i) is:

$$Emission_i = 0.916 \times (Prob_{da,i} + \frac{Prob_{sr,i}}{2}) \times distance \tag{2}$$

where 2 means persons share a ride. Note that the assumption that drive to transit mode is still ignored here.

For example, for the first observation:

$$\begin{aligned} Emission_0 &= 0.916 \times \left(Prob_{da,i} + \tfrac{Prob_{sr,i}}{2}\right) \times distance \\ &= 0.916 \times \left(0.700 + \tfrac{0.300}{2}\right) \times 11.2 = 8.72 \end{aligned} \tag{3}$$

$$\begin{aligned} Emission_1 &= 0.916 \times \left(Prob_{da,i} + \frac{Prob_{sr,i}}{2} \right) \times distance \\ &= 0.916 \times \left(0.700 + \frac{0.300}{2} \right) \times 11.2 = 8.72 \end{aligned} \tag{4}$$

which is equal, indicating there is no reduction for observation 1. It is easy to understand, as the toll does not have any influence on this observation.

To calculate the total emission, we need to calculate the weighted summation:

$$TotalEmission_i = \sum_{observations} Weight \times Emission_i \tag{5}$$

The results has been shown in part (a).

Task 4:

Question (a):

We think the answer is yes. Travel time would probably be less as a result of the congestion charge since in that case, people tend to shift modes include driving to other modes. Hence, there will be fewer people driving, which makes travel time less since there will be less congestion.

Question (b):

The mode shares of drive alone and share ride are underestimated and the mode shares of other 4 modes are overestimated. CO_2 emission forecasts are overestimated. The reasons are as follows:

For modes include driving, the influence of congestion fee on travel time was not considered. Hence, travel time for computing these 2 mode shares should be more than their actual values. Similarly, we can conclude the opposite case in other 4 modes. For CO_2 emission forecasts, congestion fee would cause less emission since fewer people would drive.

Task 5:

Code (MNL-Forecasting.ipynb) is attached in **Appendix**.

Part 2 Research Project

Chenglong's answer:

Our group's research project is people's decision on divorcement.

1. Key attributes and characteristics

Attributes:
1. availability of different sources (supermarkets, parks)
2. society's acceptance of divorcement
3. safety of surrounding environment

Characteristics:
1. people's age
2. people's psychological loneliness state.
3. people's income level
4. people's need for help (in the life)

2.

stated preferences (SP): We could create some extreme conditions for people to decide. Could have some data about peoples characteristics that are hard to observe (like psychological conditions). We could create lots of conditions for people to consider, therefore increasing the data available. However, data may be a little incongruent with actual behavior. People may be easy to respond divorce, but when the condition is met, people may be careful about divorcement.

Revealed preferences (RP): It is the real behavior of people, so no stated error as above. Moreover, other factors that are not initially considered in our investigations are reflected. However, factors like people's psychological loneliness state is hard to observe. Examples of some extreme conditions are hard to find in real world. Conditions may appear that couples live together, but they have divorced.

3.

We go to some regions and record the attributes of environments. We could also consult local officials for data about the neighborhood, like the availability of supermarket and recreation facilities. Then we randomly choose residents and ask them about their characteristics as well as marital status. Note that we should select regions that can reflect all kinds of environments, including cities, suburban areas and rural areas.

4. SP survey

1. What is your age?
2. Are you feeling lonely right now?
3. How much money do your earn every year? (Several choices)
4. Do you have any special needs that require someone elses (like couple) help?
5. Suppose your relationship with your wife/husband is bad (you don't talk to each other for several days). Suppose you do not have a car and supermarkets are quite far away from you (you need to take 2 hour's bus to the nearest supermarket), also there is not library, sports center and park that are within walking distance (2 miles), but the surrounding safety condition of your house is not bad, will you choose divorce?

A. yes;
B. no;
C. don't know, maybe

Junzhe's answer:

The research focuses on determining whether the driving cost will influence the ridership of bike.

1. Key attributes and characteristics

Key attributes: parking condition, distance, weather, and geography.
Key characteristics: age, gender, and health condition.

2.

Although RP reveals choice behaviors in actual conditions and cognitively congruent with actual behavior, it is hard to be collected in this study. For example, because only its choice is available, the health conditions of decision makers which are not easily collected by RP. Besides, RP is too expensive for this study because it is difficult to obtain multiple responses from an individual. SP is the good choice for the study, although some market and personal constraints may not be considered. SP bases on hypothetical scenarios which let repetitive questioning to be easily implemented. Furthermore, these is no measurement errors in the SP.

3.

In order to collect RP data, I will make a survey which lasts a year to include different weather conditions. The survey will be nationwide. People live in different geographies will take the survey. The characteristics of volunteers will be recorded by the survey, and the attributes of the choosing condition will be measured manually. All the alternatives of peoples choices, even including staying at home for extreme weathers, will be recorded.

4. SP survey

Questions of peoples characteristics:
Age: 0-20, 20-40, 40-60, >60
Gender: female, male
Health conditions: lot of energy, normal, tired, ill

Questions of attributes:
The hypothetical situations of different combinations of attributes will be asked in the survey.

The survey form is shown in Table 5

Table 5: Junzhe's survey form

Time for finding a parking carport	Distance	If the weather is good	If there is a hill	Choose bike or others?
0 - 5 mins	0- 2 miles	Y	Y	
5 - 10 mins	2 - 4 miles	Y	Y	
>10 mins	4 - 6 miles	Y	Y	
0 - 5 mins	0- 2 miles	N	N	
5 - 10 mins	2 - 4 miles	N	N	
>10 mins	4 - 6 miles	N	N	

Franklin's answer:

Again, my answer for Part 2 in this problem set is based on the same research question in my previous problem sets: Whether driving costs at Berkeley would influence a Berkeley's graduate student's choice on driving, taking public transit or riding a bike to campus everyday (very similar to Junzhe's question).

1. Key attributes and characteristics

Key attributes: parking fee, fuel cost, bike cost, public transit cost, maintenance cost, travel time, traffic condition, and hills.

Key characteristics: level of patience, financial status, and physical condition.

2.

For SP data, the best thing we may notice is that data can be significantly augmented. For example, in the survey, we could ask people like "which mode would you choose given condition A/B/C", which is very easy to answer since people do not have to remember something they did or chose. Hence, data augmentation is easy in this case. However, since such data is largely based on people's "imagination", they might be less convincing than RP data. Also, such data can be easily affected by the Hawthorne Effect.

For RP data, it is kind like the "opposite" of SP data. While such data is more precise and valuable, it is harder to collect since people should have the experience and they are going to remember it.

3.

To collect RP data, we need "facts". Instead of asking people their mode choices, we can observe and count the number of each chosen mode at the nearest bus stops, parking lots, and the entrance of the campus in the morning. We might also need to collect the traffic condition data from the Internet during certain periods, as well as the geography data (i.e., hills). Most importantly, the price data. Based on these conditions, we should be able to estimate our model.

4. SP survey

1. Do you like driving a car?
A. Yes B. No C. Can't drive or don't possess a car

2. Do you like riding a bike?
A. Yes B. No C. Can't ride or don't possess a bike

3. Do you like taking a public transit?
A. Yes B. No

4. Please rate your level of patience (0-5, "5" is the most patient).
A. 0 B. 1 C. 2 D. 3 E. 4 F. 5

5. Please rate your financial status (0-5, "5" has the best financial status).
A. 0 B. 1 C. 2 D. 3 E. 4 F. 5

6. Please rate your physical condition (0-5, "5" has the best physical condition).
A. 0 B. 1 C. 2 D. 3 E. 4 F. 5

Part 3 Supplemental Problems

A)

Model 1 is better than Model 2 and 3 since it has better ρ^2 and better likelihood compared to Model 2 and 3. Then, we perform a likelihood ratio test:

$$-2 \times (LModel_1 - LModel_4) = -19.6 < \chi^2_{1,0.05} \tag{6}$$

So we conclude that Model 4 does not improve Model 1 significantly. Thus, we prefer Model 1.

B)

$$\varepsilon = \varepsilon_{2n} - \varepsilon_{1n} \tag{7}$$

$$f(\varepsilon) = \begin{cases} 0 & V_{1n} - V_{2n} \leq -1 \\ \varepsilon + 1 & -1 < V_{1n} - V_{2n} < 0 \\ -\varepsilon + 1 & 0 < V_{1n} - V_{2n} < 1 \\ 0 & V_{1n} - V_{2n} \geq 1 \end{cases} \tag{8}$$

$$P_n(1) = Pr(V_{1n} - V_{2n} \geq \varepsilon_{2n} - \varepsilon_{1n}) = \int_{-1}^{-0.5} \varepsilon + 1 \; d\varepsilon = 0.125 \tag{9}$$

Appendix

```
# PS4 - CE264
# GSI: Mustapha Harb, Mengqiao Yu, Andrew Campbell
# Authors: Chenglong Li, Junzhe Shi, and Franklin Zhao

# importing the requried libraries
from collections import OrderedDict    # For recording the model specification

import pandas as pd                    # For file input/output
import numpy as np                     # For vectorized math operations

import pylogit as pl                   # For MNL model estimation and
                                       # conversion from wide to long format

# reading the data file
data_wide  = pd.read_csv("data01.csv",sep=",")

# converting the data from wide to long format

# Create the list of individual specific variables
ind_variables = data_wide.columns.tolist()[:2] + ["weights"]

# Specify the variables that vary across individuals and some or all alternatives
# The keys are the column names that will be used in the long format dataframe.
# The values are dictionaries whose key-value pairs are the alternative id and
# the column name of the corresponding column that encodes that variable for
# the given alternative. Examples below.
alt_varying_variables = {u'travel_time': dict([(1, 'tt_da'),
                                               (2, 'tt_sr'),
                                               (3, 'tt_walk'),
                                               (4, 'tt_bike'),
                                               (5, 'tt_wt'),
                                               (6, 'tt_dt')]),
                         u'distance_car': dict([(1, 'dist_car'),
                                                (2, 'dist_car')]),
                         u'travel_cost': dict([(1, 'cost_da'),
                                               (2, 'cost_sr'),
                                               (5, 'cost_wt'),
                                               (6, 'cost_dt')]),
                         u'access_time': dict([(5, 'accTime_wt'),
                                               (6, 'accTime_dt')]),
                         u'egress_time': dict([(5, 'egrTime_wt'),
                                               (6, 'egrTime_dt')]),
                         u'initial_wait': dict([(5, 'iWait_wt'),
                                                (6, 'iWait_dt')]),
                         u'transfer_wait': dict([(5, 'xWait_wt'),
                                                 (6, 'xWait_dt')]),
```

```
                           u'access_distance_dt': dict([(6, "accDist_dt")])}

# Specify the availability variables
# Note that the keys of the dictionary are the alternative id's.
# The values are the columns denoting the availability for the
# given mode in the dataset.


availability_variables = {1: 'avail_da',
                          2: 'avail_sr',
                          3: 'avail_walk',
                          4: 'avail_bike',
                          5: 'avail_wt',
                          6: 'avail_dt'}

##########
# Determine the columns for: alternative ids, the observation ids and the choice
##########
# The 'custom_alt_id' is the name of a column to be created in the long-format data
# It will identify the alternative associated with each row.
custom_alt_id = "mode_id"

# Create a custom id column that ignores the fact that this is a
# panel/repeated-observations dataset. Note the +1 ensures the id's start at one.
obs_id_column = "obsID"

# Create a variable recording the choice column
choice_column = "choice"

# Perform the conversion to long-format
data_long = pl.convert_wide_to_long(data_wide,
                                    ind_variables,
                                    alt_varying_variables,
                                    availability_variables,
                                    obs_id_column,
                                    choice_column,
                                    new_alt_id_name=custom_alt_id)
# Look at the resulting long-format dataframe
data_long.head(37921).T
#len(data_long)

##########
# Create scaled variables so the estimated coefficients are of similar magnitudes
##########
# Scale the travel time column by 60 to convert raw units (minutes) to hours
data_long["travel_time_hrs"] = data_long["travel_time"] / 60.0

# Scale the access by 60 to convert raw units (minutes) to hours
```

```
data_long["access_time_hrs"] = data_long["access_time"] / 60.0

# for drive to transit let us combine travel time and access time
data_long["travel_time_access_time_hrs"] = data_long["travel_time_hrs"] + data_long["access_time_hrs"]

#Scale the egress time by 60
data_long["egress_time_hrs"] = data_long["egress_time"] / 60.0

# combining access and egress time which we want to use for the walk to transit alternative
data_long["acess_egress_hrs"] = data_long["access_time_hrs"] + data_long["egress_time_hrs"]

# scaling the initial wait by 60
data_long["initial_wait_hrs"] = data_long["initial_wait"] / 60.0

# scaling the transfer wait by 60
data_long["transfer_wait_hrs"]  = data_long["transfer_wait"] / 60.0

# combining transfer wait and initial wait to be used for walk to transit and bike to transit
data_long["initial_transfer_wait_hrs"] = data_long["initial_wait_hrs"] + data_long["transfer_wait_hrs"]

# Consider the toll effect
toll = 10
data_long["travel_cost_toll"] = data_long["travel_cost"] + ((data_long["originTAZ"] <= 42) | (data_long["destTAZ"] <= 42)).astype(int) * \
                                ((data_long["mode_id"] == 1) | (data_long["mode_id"] == 2)).astype(int) * toll

# creating non-linear transformations for the cost variable
cutOff1 = 2
cutOff2 = 7

# ex: 1 become: cat1: 1; cat 2: 0; cat3: 0
# ex: 3 become: cat1: 2; cat 2: 1; cat3: 0
# ex: 7 become: cat1: 2; cat 2: 5; cat3: 0
# ex: 9 become: cat1: 2; cat 2: 5; cat3: 2
data_long["cost_cat_one"] = (data_long["travel_cost"] <= cutOff1)*data_long["travel_cost"] + \
                            (data_long["travel_cost"] > cutOff1)*cutOff1

data_long["cost_cat_two"] = (data_long["travel_cost"] > cutOff1)*(data_long["travel_cost"] <= cutOff2)*(data_long["travel_cost"] - cutOff1)\
                          + (data_long["travel_cost"] > cutOff2)* (cutOff2 - cutOff1)
```

```
133
134 data_long["cost_cat_three"] = (data_long["travel_cost"] > cutOff2)*(data_long["
    travel_cost"] - cutOff2)
135
136
137
138 # specifying the utility equations
139
140 # NOTE: - Specification and variable names must be ordered dictionaries.
141 #       - Keys should be variables within the long format dataframe.
142 #         The sole exception to this is the "intercept" key.
143 #       - For the specification dictionary, the values should be lists
144 #         of integers or or lists of lists of integers. Within a list,
145 #         or within the inner-most list, the integers should be the
146 #         alternative ID's of the alternative whose utility specification
147 #         the explanatory variable is entering. Lists of lists denote
148 #         alternatives that will share a common coefficient for the variable
149 #         in question.
150
151 basic_specification = OrderedDict()
152 basic_names = OrderedDict()
153
154
155 basic_specification["intercept"] = [ 2, 3, 4, 5, 6]
156 basic_names["intercept"] = ['ASC SR',
157                             'ASC Walk', 'ASC Bike', 'ASC WT', 'ASC DT']
158
159 basic_specification["travel_time_hrs"] = [[1, 2, 5], 4, 3]
160 basic_names["travel_time_hrs"] = ['In-Vehicle Travel Time, units:hrs (DA, SR, WT)',
161                                   'Bike Time, units:hrs (Bike)',
162                                   'Walk Time, units:hrs (Walk)']
163
164 basic_specification["travel_time_access_time_hrs"] = [6]
165 basic_names["travel_time_access_time_hrs"] = ["In-Vehicle Travel Time, units:hrs, (
    DT)"]
166
167 basic_specification["acess_egress_hrs"] = [5]
168 basic_names["acess_egress_hrs"] = ["Walk Time, units:hrs, (WT)"]
169
170 basic_specification["egress_time_hrs"] = [6]
171 basic_names["egress_time_hrs"] = ["Walk Time, units:hrs, (DT)"]
172
173 basic_specification["initial_transfer_wait_hrs"] = [[5, 6]]
174 basic_names["initial_transfer_wait_hrs"] = ["Waiting Time, units:hrs, (WT and DT)"]
175
176
177 basic_specification["cost_cat_one"] = [[1, 2, 5,6]]
178 basic_names["cost_cat_one"] = ['Cost: Under $2']
```

```
179
180 basic_specification["cost_cat_two"] = [[1, 2, 5,6]]
181 basic_names["cost_cat_two"] = ['Cost: (2 - 7)$']
182
183 basic_specification["cost_cat_three"] = [[1, 2, 5,6]]
184 basic_names["cost_cat_three"] = ['Cost: Above $7']
185
186 # taking a sample of 10,000 observation from the BATS 2000 dataset
187 new_data = data_long.loc[data_long[obs_id_column].isin(range(10000))].copy()
188
189 # Estimate the multinomial logit model (MNL)
190 data_mnl = pl.create_choice_model(data=new_data,
191                                   alt_id_col=custom_alt_id,
192                                   obs_id_col=obs_id_column,
193                                   choice_col=choice_column,
194                                   specification=basic_specification,
195                                   model_type="MNL",
196                                   names=basic_names)
197
198 # Specify the initial values and method for the optimization.
199 data_mnl.fit_mle(np.zeros(15))
200
201
202 # Look at the estimation results
203 bestmodel = data_mnl.get_statsmodels_summary()
204 bestmodel
205
206 # prediction - sample enumeration
207 # array of probabilities for each available alternative for all individuals in the
    new_data file that
208 # was used for estimation
209 #toll = 10
210 #data_long["toll_dummy"] = (data_long["originTAZ"] <= 42) | (data_long["destTAZ"] <=
     42)
211 #data_long["travel_cost_toll"] = data_long["travel_cost"]
212
213 #new_data["travel_cost"] = new_data["travel_cost"] + ((new_data["originTAZ"] <= 42)
    | (new_data["destTAZ"] <= 42)).astype(int) * \
214 #                                ((new_data["mode_id"] == 1) | (new_data["mode_id"]
    == 2)).astype(int) * toll
215 new_data["cost_cat_one"] = (new_data["travel_cost_toll"] <= cutOff1)*new_data["
    travel_cost_toll"] + \
216                            (new_data["travel_cost_toll"] > cutOff1)*cutOff1
217 new_data["cost_cat_two"] = (new_data["travel_cost_toll"] > cutOff1)*(new_data["
    travel_cost_toll"] <= cutOff2)*(new_data["travel_cost_toll"] - cutOff1)\
218                          + (new_data["travel_cost_toll"] > cutOff2)* (cutOff2 -
    cutOff1)
219
```

```
new_data["cost_cat_three"] = (new_data["travel_cost_toll"] > cutOff2)*(new_data["travel_cost_toll"] - cutOff2)

prediction_array = data_mnl.predict(new_data)
#pd.set_option('display.max_columns', 100)
#new_data.head(100).T
emission = []
emission = np.empty(9999)
for obs in range(1, 10000):
    tot_emission = (new_data["obsID"] == obs).astype(int) * (new_data["mode_id"] == 1).astype(int) * new_data["weights"] * \
    new_data["distance_car"] * prediction_array
    tot_emission = tot_emission + (new_data["obsID"] == obs).astype(int) * (new_data["mode_id"] == 2).astype(int) * new_data["weights"] * \
    new_data["distance_car"] * prediction_array /2
    tot_emission = 0.916 * tot_emission
    emission[obs - 1] = np.sum(tot_emission)

emission_tot = np.sum(emission)

# check out the probabilities
prediction_array[0:20]
emission_tot

# market shares
# the following script performs sample enumeration for all alternatives by catering for individual weights

total_weights = np.sum(data_wide.loc[:9999, 'weights'])
weights = new_data["weights"] / total_weights

alternative_names = ["drive", "shared ride", "walk", "bike",
                     "walk transit", "drive transit"]
num_alternatives = data_long["mode_id"].unique().size
market_shares = np.empty(num_alternatives)
for i in range(1, 7):
    filter_condition = (new_data["mode_id"] == i).values
    num_obs_in_condition = filter_condition.sum()
    current_weights = weights[filter_condition]
    assert current_weights.size == num_obs_in_condition
    current_alternative_share = prediction_array[filter_condition].dot(current_weights)
    market_shares[i - 1] = current_alternative_share

pd.Series(market_shares, index=alternative_names)
```