Technical Report for 2.4.1 Mode/Destination Choice for Work Tour to Unusual Location

Siyu LI

June 2016

Model Description

This model will predict both mode and destination for work tours to unusual location. Basically Mode/Destination Choice model for work tour to unusual location, shopping tour, and other tour have the same specification and variables.

Choice Set

There are 9 modes and 1092 MTZs. The choice set thus has `9*1092=9828` alternatives.

- `V_1` to `V_1092`: the mode is Public bus, destination is from 1092 MTZs.
- `V_1093` to `V_2184`: the mode is MRT/LRT, destination is from 1092 MTZs.
- `V_2185` to `V_3276`: the mode is Private bus, destination is from 1092 MTZs.
- `V_3277` to `V_4368`: the mode is Drive alone, destination is from 1092 MTZs.
- `V_4369` to `V_5460`: the mode is Shared ride 2, destination is from 1092 MTZs.
- `V_5461` to `V_6552`: the mode is Shared ride 3+, destination is from 1092 MTZs.
- `V_6553` to `V_7644`: the mode is Motorcycle, destination is from 1092 MTZs.
- `V_7645` to `V_8736`: the mode is Walk, destination is from 1092 MTZs.
- `V_8737` to `V_9828`: the mode is Taxi, destination is from 1092 MTZs.

Model Structure

The mode/destination Model is a MNL model.

```
V_counter=0
#bus
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_bus +

    cost_over_income_bus_%s * (1-missing_income) * beta_cost_bus_mrt_1 +
    cost_bus_%s * missing_income * beta_cost_bus_mrt_2 +
    tt_bus_%s * beta_tt_bus_mrt +
    beta_central_bus_mrt * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s) +
    (walk_distance_first_%s + walk_distance_second_%s) * beta_distance_bus_mrt +
    beta_female_bus * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j))

#mrt
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_mrt +

    cost_over_income_mrt_%s * (1-missing_income) * beta_cost_bus_mrt_1 +
    cost_mrt_%s*missing_income*beta_cost_bus_mrt_2 +
    tt_mrt_%s * beta_tt_bus_mrt +
    beta_central_bus_mrt * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_bus_mrt +
    beta_female_mrt * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j))

#private_bus
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_private_bus +

    cost_over_income_private_bus_%s * (1-missing_income) * beta_cost_private_bus_1 +
    cost_private_bus_%s*missing_income*beta_cost_private_bus_2 +
    tt_private_bus_%s * beta_tt_bus_mrt +
    beta_central_private_bus * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_private_bus +
    beta_female_private_bus * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j))

#drive1
for j in range(1,1093):
    V_counter=V_counter+1
```

```
    exec("V_%s = beta_cons_drive1+

    cost_over_income_drive1_%s * (1-missing_income) * beta_cost_drive1_1 +
    cost_drive1_%s*missing_income*beta_cost_drive1_2 +
    tt_drive1_%s * beta_tt_drive1+beta_central_drive1 * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_drive1 +
    beta_zero_drive1 * zero_car +
    beta_oneplus_drive1 * one_plus_car +
    beta_twoplus_drive1 * two_plus_car +
    beta_threeplus_drive1 * three_plus_car +
    beta_female_drive1 * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j,j))

#share2
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_share2 +

    cost_over_income_share2_%s * (1-missing_income) * beta_cost_share2_1 +
    cost_share2_%s*missing_income*beta_cost_share2_2 +
    tt_share2_%s * beta_tt_drive1 +
    beta_central_share2 * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_share2 +
    beta_zero_share2 * zero_car +
    beta_oneplus_share2 * one_plus_car +
    beta_twoplus_share2 * two_plus_car +
    beta_threeplus_share2 * three_plus_car +
    beta_female_share2 * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j,j))

#share3
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_share3 +

    cost_over_income_share3_%s * (1-missing_income) * beta_cost_share3_1 +
    cost_share3_%s*missing_income*beta_cost_share3_2 +
    tt_share3_%s * beta_tt_drive1 +
    beta_central_share3 * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_share3 +
    beta_zero_share3 * zero_car +
    beta_oneplus_share3 * one_plus_car +
```

```
    beta_twoplus_share3 * two_plus_car +
    beta_threeplus_share3 * three_plus_car +
    beta_female_share3 * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j))

#motor
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_motor +

    cost_over_income_motor_%s * (1-missing_income) * beta_cost_motor_1 +
    cost_motor_%s*missing_income*beta_cost_motor_2 +
    tt_motor_%s * beta_tt_drive1 +
    beta_central_motor * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_motor +
    beta_zero_motor * zero_motor +
    beta_oneplus_motor * one_plus_motor +
    beta_twoplus_motor * two_plus_motor +
    beta_threeplus_motor * three_plus_motor +
    beta_female_motor * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j,j))


#walk
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_walk+

    tt_walk_%s*beta_tt_walk +
    beta_central_walk * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1) +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_walk +
    beta_female_walk * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#taxi
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_taxi +

    cost_over_income_taxi_%s *(1-missing_income)* beta_cost_taxi_1 +
    cost_taxi_%s*missing_income*beta_cost_taxi_2+
    tt_taxi_%s * beta_tt_taxi +
    beta_central_taxi * central_dummy_%s +
    beta_log*
    log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1)+
```

```
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_taxi +
    beta_female_taxi * Female_dummy" % (V_counter,j,j,j,j,j,j,j,j,j))


#Estimated values for all betas
#Notice: the betas that not estimated are fixed to zero.

beta_cost_bus_mrt_1= Beta('bus/mrt cost over income',0,-10,10,0) = -7.05
beta_cost_private_bus_1 =Beta('private bus cost over income',0,-10,10,0) = -6.30
beta_cost_drive1_1= Beta('drive1 cost over income',0,-10,10,0) = -5.55
beta_cost_share2_1= Beta('shared ride 2 cost over income',0,-10,10,0) = -5.87
beta_cost_share3_1= Beta('shared ride 3 cost over income',0,-10,10,0) = -4.82
beta_cost_motor_1 = Beta('motor cost over income',0,-10,10,0) = -0.867
beta_cost_taxi_1 = Beta('taxi cost over income',0,-10,10,0) = -1.46

beta_cost_bus_mrt_2= Beta('bus/mrt cost (missing income)',0,-10,10,1) = 0
beta_cost_private_bus_2 =Beta('private bus cost (missing income)',0,-10,10,1) = 0
beta_cost_drive1_2= Beta('drive1 cost (missing income)',0,-10,10,1) = 0
beta_cost_share2_2= Beta('shared ride 2 cost (missing income)',0,-10,10,1) = 0
beta_cost_share3_2= Beta('shared ride 3 cost (missing income)',0,-10,10,1) = 0
beta_cost_motor_2 = Beta('motor cost (missing income)',0,-10,10,1) = 0
beta_cost_taxi_2 = Beta('taxi cost (missing income)',0,-10,10,1) = 0


beta_tt_bus_mrt = Beta('bus/mrt travel time',0,-10,10,0) = -0.952
beta_tt_private_bus= Beta('private bus travel time',0,-10,10,1) = 0
beta_tt_drive1 = Beta('drive1 travel time',0,-10,10,0) = -0.448
beta_tt_share2= Beta('shared ride 2 travel time',0,-10,10,1) = 0
beta_tt_share3= Beta('shared ride 3 travel time',0,-10,10,1) = 0
beta_tt_motor=Beta('motor travel time',0,-10,10,1) = 0
beta_tt_walk=Beta('walk travel time',0,-10,10,0) = -2.76
beta_tt_taxi= Beta('taxi travel time',0,-10,10,0) = -1.98

beta_log = Beta('beta for log term',0,-10,10,0) = 0.796
beta_area = Beta('beta for area',0,-10,10,0) = 5.49
beta_population = Beta('beta for population',0,-10,10,0) = -6.25

beta_central_bus_mrt = Beta('central dummy beta for bus/mrt',0,-10,10,0) = 0.327
beta_central_private_bus=Beta('central dummy beta for private bus',0,-10,10,0) = -1.26
beta_central_drive1=Beta('central dummy beta for drive1',0,-10,10,1) = 0
beta_central_share2=Beta('central dummy beta for shared ride 2',0,-10,10,0) = 0.477
beta_central_share3=Beta('central dummy beta for shared ride 3',0,-10,10,0) = 0.397
beta_central_motor=Beta('central dummy beta for motor',0,-10,10,0) = 0.488
beta_central_walk=Beta('central dummy beta for walk',0,-10,10,0) = -1.38
beta_central_taxi=Beta('central dummy beta for taxi',0,-10,10,0) = 0.899
```

```
beta_distance_bus_mrt = Beta('distance beta for bus/mrt',0,-10,10,0) = -0.00342
beta_distance_private_bus=Beta('distance beta for private bus',0,-10,10,0) = -0.00552
beta_distance_drive1=Beta('distance beta for drive1',0,-10,10,1) = 0
beta_distance_share2=Beta('distance beta for shared ride 2',0,-10,10,0) = -0.0178
beta_distance_share3=Beta('distance beta for shared ride 3',0,-10,10,0) = -0.0284
beta_distance_motor=Beta('distance beta for motor',0,-10,10,0) = -0.0230
beta_distance_walk=Beta('distance beta for walk',0,-10,10,1) = 0
beta_distance_taxi=Beta('distance beta for taxi',0,-10,10,0) = -0.00230


beta_cons_bus = Beta('constant for bus',0,-10,10,0) = -1.89
beta_cons_mrt = Beta('constant for mrt',0,-10,10,0) = -1.78
beta_cons_private_bus = Beta('constant for private bus',0,-10,10,0) = -2.71
beta_cons_drive1 = Beta('constant for drive1',0,-10,10,1) = 0
beta_cons_share2 = Beta('constant for shared ride 2',0,-10,10,0) = -4.77
beta_cons_share3 = Beta('constant for shared ride 3',0,-10,10,0) = -5.66
beta_cons_motor = Beta('constant for motor',0,-20,20,0) = -3.71
beta_cons_walk = Beta('constant for walk',0,-10,10,0) = -1.69
beta_cons_taxi = Beta('constant for taxi',0,-10,10,0) = -5.45

beta_zero_drive1=Beta('zero cars in drive1',0,-10,10,1) = 0
beta_oneplus_drive1=Beta('one plus cars in drive1',0,-10,10,1) = 0
beta_twoplus_drive1=Beta('two plus cars in drive1',0,-10,10,0) = 2.11
beta_threeplus_drive1=Beta('three plus cars in drive1',0,-10,30,1) = 0

beta_zero_share2=Beta('zero cars in share2',0,-10,10,1) = 0
beta_oneplus_share2=Beta('one plus cars in share2',0,-10,10,0) = 2.57
beta_twoplus_share2=Beta('two plus cars in share2',0,-10,10,0) = 1.80
beta_threeplus_share2=Beta('three plus cars in share2',0,-10,10,1) = 0

beta_zero_share3=Beta('zero cars in share3 plus',0,-10,10,1) = 0
beta_oneplus_share3=Beta('one plus cars in share3 plus',0,-10,10,0) =1.80
beta_twoplus_share3=Beta('two plus cars in share3 plus',0,-10,10,1) = 0
beta_threeplus_share3=Beta('three plus cars in share3 plus',0,-30,10,1) = 0


beta_zero_motor=Beta('zero motors in motor',0,-10,10,1) = 0
beta_oneplus_motor=Beta('one plus motors in motor',0,-10,20,1) = 0
beta_twoplus_motor=Beta('two plus motors in motor',0,-10,10,0) = 5.07
beta_threeplus_motor=Beta('three plus motors in motor',0,-10,10,1) = 0

beta_female_bus=Beta('female dummy in bus',0,-10,10,0) = 2.00
beta_female_mrt=Beta('female dummy in mrt',0,-10,10,0) = 1.73
beta_female_private_bus=Beta('female dummy in privatebus',0,-10,10,0) = 1.33
beta_female_drive1=Beta('female dummy in drive1',0,-10,10,1) = 0
beta_female_share2=Beta('female dummy in share2',0,-10,10,0) = 1.59
```

```
beta_female_share3=Beta('female dummy in share3 plus',0,-10,10,0) = 0.840
beta_female_motor=Beta('female dummy in motor',0,-10,10,0) = -2.48
beta_female_taxi=Beta('female dummy in taxi',0,-10,10,0) = 2.59
beta_female_walk=Beta('female dummy in walk',0,-10,10,0) = 3.29
```

Variables

Basically the variables used in mode/destination choice model is the same with variables used in mode choice model: travel time, travel cost, zonal information, personal and household characteristics. The difference is that for mode/destination choice, the destination of a tour is not determined. Thus we need to provide travel time, travel cost, zonal information for each of the 1092 zones.

In the specification, we include size variables. Mode/destination model for different purposes use different size variables to describe zonal characteristics.

1. For work tour to unusual location:

```
log(employment_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1)
```

2. For shopping tour:

```
log(shop_%s+exp(beta_area)*area_%s+exp(beta_population)*population_%s+1)
```

3. For other tour:

```
log(exp(beta_area)*area_%s+exp(beta_population)*population_%s+1)
```

Previous to generating related variables, data sets (`am`,`pm`,`op` and `zone_employment`) can be stored in dicts for fast querying.

```
# From gen.py
import numpy as np
am=np.genfromtxt(file_dir+'AMcosts.dat',names=True)
pm=np.genfromtxt(file_dir+'PMcosts.dat',names=True)
op=np.genfromtxt(file_dir+'OPcosts.dat',names=True)
zone_employ=np.genfromtxt(file_dir+'zone_employment.txt',names=True)
AM={};PM={}; OP={}; ZONE={}

for row in am:
    AM[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['AM2dis'], \
```

```
    'car_ivt' : row['AM2Tim']/60,\
    'pub_ivt':  row['AM2ivt']/60, \
    'pub_out':  (row['AM2aux']+row['AM2wtt'])/60, \
    'pub_wtt': row['AM2wtt']/60,\
    'pub_walkt': row['AM2aux']/60,\
    'car_cost_erp': row['AM2ERP']/100, \
    'avg_transfer': row['AM2trf'],\
    'pub_cost': row['AM2cos']/100}

for row in pm:
    PM[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['PM2dis'], \
    'car_ivt':   row['PM2Tim']/60,\
    'pub_ivt':  row['PM2ivt']/60, \
    'pub_out':  (row['PM2aux']+row['PM2wtt'])/60, \
    'pub_wtt': row['PM2wtt']/60,\
    'pub_walkt': row['PM2aux']/60,\
    'car_cost_erp': row['PM2ERP']/100, \
    'avg_transfer': row['PM2trf'],\
    'pub_cost': row['PM2cos']/100}

for row in op:
    OP[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['OPdis'], \
    'car_ivt':   row['OPTim']/60,\
    'pub_ivt':  row['OPivt']/60, \
    'pub_out':  (row['OPaux']+row['OPwtt'])/60, \
    'pub_wtt': row['OPwtt']/60,\
    'pub_walkt': row['OPaux']/60,\
    'car_cost_erp': row['OPERP']/100, \
    'avg_transfer': row['OPtrf'],\
    'pub_cost': row['OPcos']/100}

for row in zone_employ:
    ZONE[int(row['zone_ID'])]=\
    {'zone_id':int(row['zone_ID']),\
    'zone_code':int(row['zone_code']),\
    'employment':row['employment'],\
    'central_dummy': row['central_dummy'],\
    'parking_rate': row['parking_rate'],\
    'population':row['population'],\
    'area':row['area']}
```

Noted that in the original skims `AMCosts.dat`,`PMCosts.dat`,`OPCosts.dat`, costs are measured in cent and time is measured in minutes. For our variable, we need to first convert them to dollor and hour when generating the dicts (divided by

100 and 60 respectively)

Cost Related Variables

Variables in this section are related to travel cost. Noted that if `origin==destination`, all costs are zero (except parking cost)

```
for destination in 1092zones:

    #public transporation cost for first half tour from origin to destination.
    #0 if origin==destination
    cost_public_first = AM[(origin,destination)]['pub_cost']
    #public transporation cost for second half tour from destination to origin.
    #0 if origin==destination
    cost_public_second = PM[(destination,origin)]['pub_cost']

    cost_public_%s = cost_public_first + cost_public_second # % destination


    #cost of car ERP for first half tour from origin to destination
    #0 if origin==destination
    cost_car_ERP_first = AM[(origin,destination)]['car_cost_erp']

    #cost of car ERP for second half tour from destination to origin
    #0 if origin==destination
    cost_car_ERP_second = PM[(destination,origin)]['car_cost_erp']

    #cost of car operational cost for first half tour from origin to destination
    #0 if origin==destination
    cost_car_OP_first = AM[(origin,destination)]['distance']*0.147

    #cost of car operational cost for second half tour from destination to origin
    #0 if origin==destination
    cost_car_OP_second = PM[(destination,origin)]['distance']*0.147

    #cost of car parking cost at destination zone (assume an eight hour duration)
    cost_car_parking = 8 * ZONE[destination]['parking_rate']

    cost_car_ERP_%s = cost_car_ERP_first + cost_car_ERP_second # % destination
    cost_car_OP_%s = cost_car_OP_first + cost_car_OP_second # % destination


    walk_distance1_%s = AM[(origin,destination)]['distance'] # 0 if origin == destination
    walk_distance2_%s = PM[(destination,origin)]['distance'] # 0 if origin == destination
```

```
    central_dummy_%s=ZONE[destination]['central_dummy'] # % destination

for i in range(1,1093):
    exec("cost_bus_%s = cost_public_%s" % (i,i))
    exec("cost_mrt_%s = cost_public_%s" % (i,i))
    exec("cost_private_bus_%s = cost_public_%s" % (i,i))

    exec("cost_drive1_%s=
    cost_car_ERP_%s+cost_car_OP_%s+cost_car_parking_%s" % (i,i,i,i))

    exec("cost_share2_%s=
    (cost_car_ERP_%s+cost_car_OP_%s+cost_car_parking_%s)/2.0" % (i,i,i,i))

    exec("cost_share3_%s=
    (cost_car_ERP_%s+cost_car_OP_%s+cost_car_parking_%s)/3.0" % (i,i,i,i))

    exec("cost_motor_%s=
    0.5*cost_car_ERP_%s+0.5*cost_car_OP_%s+0.65*cost_car_parking_%s" % (i,i,i,i))

    exec("cost_taxi_%s=
    6.8+cost_car_ERP_%s+6*central_dummy_%s+
    ((walk_distance_first_%s*(walk_distance_first_%s>10)-
    10*(walk_distance_first_%s>10))/0.35+
    (walk_distance_first_%s*(walk_distance_first_%s<=10)+10*
    (walk_distance_first_%s>10))/0.4)*0.22+
    ((walk_distance_second_%s*(walk_distance_second_%s>10)-
    10*(walk_distance_second_%s>10))/0.35+
    (walk_distance_second_%s*(walk_distance_second_%s<=10)+
    10*(walk_distance_second_%s>10))/0.4)*0.22" % (i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i))

    # the income_mid can be acquired from income class txt file for each income_id
    exec("cost_over_income_bus_%s=30*cost_bus_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_mrt_%s=30*cost_mrt_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_private_bus_%s=30*cost_private_bus_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_drive1_%s=30*cost_drive1_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_share2_%s=30*cost_share2_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_share3_%s=30*cost_share3_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_motor_%s=30*cost_motor_%s/(0.5+Income_mid)" % (i,i))
    exec("cost_over_income_taxi_%s=30*cost_taxi_%s/(0.5+Income_mid)" % (i,i))
```

Travel Time Related Variables

This section introduces variables related to travel time.

```
for destinatin in 1092zones:

    # public transportation in vehicle time for first half tour. 0 if `origin==destination`
    tt_public_ivt_first = AM[(origin,destination)]['pub_ivt']
    # public transportation in vehicle time for second half tour. 0 if `origin==destination
    tt_public_ivt_second = PM[(destination,origin)]['pub_ivt']

    # public transportation out of vehicle time for first half tour. 0 if `origin==destinat
    tt_public_out_first = AM[(origin,destination)]['pub_wtt']
    # public transportation out of vehicle time for second half tour. 0 if `origin==destina
    tt_public_out_second = PM[(destination,origin)]['pub_wtt']


    # car in vehicle time for first half tour. 0 if `origin==destination`
    tt_ivt_car_first = AM[(origin,destination)]['car_ivt']
    # car in vehicle time for second falf tour. 0 if `origin==destination`
    tt_ivt_car_second = PM[(destination,origin)]['car_ivt']

    tt_bus_ivt_%s = tt_public_ivt_first + tt_public_ivt_second  # % destination
    tt_bus_out_%s = tt_public_out_first + tt_public_out_second  # % destination

    tt_ivt_car_%s = tt_ivt_car_first + tt_ivt_car_second # %s destination


for i in range(1,1093):
    exec("tt_bus_%s = tt_public_ivt_%s +tt_public_out_%s" % (i,i,i))
    exec("tt_mrt_%s = tt_public_ivt_%s +tt_public_out_%s" % (i,i,i))
    exec("tt_private_bus_%s = tt_car_ivt_%s" % (i,i))
    exec("tt_drive1_%s = tt_car_ivt_%s +1.0/6" % (i,i))
    exec("tt_share2_%s = tt_car_ivt_%s + 1.0/6" % (i,i))
    exec("tt_share3_%s = tt_car_ivt_%s + 1.0/6" % (i,i))
    exec("tt_motor_%s = tt_car_ivt_%s+ 1.0/6" % (i,i))
    exec("tt_walk_%s = (walk_distance_first_%s+walk_distance_second_%s)/5" % (i,i,i))
    exec("tt_taxi_%s = tt_car_ivt_%s +1.0/6" % (i,i))
```

Other Variables

```
employment_%s = ZONE[%s]['employment'] # % destination
population_%s = ZONE[%s]['population'] # % destination
area_%s = ZONE[%s]['area'] # % destination
shop_%s = ZONE[%s]['shop'] # % destination

# The following dummy variables are generated based on
```

```
# household car, motorcycle ownership.

zero_car = 1*(car_own_normal==0)
one_plus_car = 1*(car_own_normal>=1)
two_plus_car = 1*(car_own_normal>=2)
three_plus_car = 1*(car_own_normal>=3)

zero_motor = 1*(motor_own==0)
one_plus_motor = 1*(motor_own>=1)
two_plus_motor = 1*(motor_own>=2)
three_plus_motor = 1*(motor_own>=3)
```

Availability of Alternatives

The availibility of a mode-destination pair is affected by both the mode and destination. If a mode is not available for an agent, all mode-destination pairs have the mode will not be available. If a destination is not available for an agent(e.g., `destination==origin`), all mode-destination pairs have the destination will not be available. Besides, we have cases where both mode and destination need to consider to determine its availability.

1. if the `destination == origin`, the destination is not available.
2. public bus, private bus and MRT/LRT are only available `if AM[(origin,destination)]['pub_ivt']>0 and PM[(destination,origin)]['pub_ivt']>0:`
3. shared2, shared3+, taxi and motorcycle are available to all.
4. Walk is only avaiable `if (AM[(origin,destination)]['distance']<=5 and PM[(destination,origin)]['distance']<=5)`
5. drive alone is available when for the agent, `has_driving_license * one_plus_car == True`