Lydia Yu
912328745
15.780 HW3

Problem 1

1)

```
# PROBLEM 1
commute = read.csv('commuteFall2020.csv')

# 1) data preprocessing
# create columns for total travel time -- bus.wait + bus.travel; car.travel
commute$time_CAR = commute$Car.Travel
commute$time_BUS = commute$Bus.Travel + commute$Bus.Wait
# create columns for cost -- BusFare; ParkingCost
commute$cost_CAR = commute$ParkingCost
commute$cost_BUS = commute$BusFare
# drop unnecessary columns from df
commute = subset(commute, select = -c(Car.Travel, Bus.Travel, Bus.Wait, ParkingCost, BusFare))
```

2)

```
# 2) fit MNL model
com = mlogit.data(commute, choice = "Choice", shape = "wide",  varying = 3:6, sep= "_")
head(com)
model  = mlogit(Choice ~ time + cost | HHIncome -1, data = com)
summary(model)
```

```
Coefficients :
               Estimate  Std. Error z-value  Pr(>|z|)
time        -5.6938e-02  7.4811e-03 -7.6110 2.709e-14 ***
cost        -3.7658e-01  5.4133e-02 -6.9565 3.489e-12 ***
HHIncome:CAR  1.3865e-05  6.0188e-06  2.3037   0.02124 *
---
```

3) Probability of choosing car for group L: **0.577**
Probability of choosing car for group H: **0.366**

```
> as.numeric(share_car_L)
[1] 0.5769579
```

```
> as.numeric(share_car_H)
[1] 0.3655602
```

```
# 3) probability that person from group H and L will take a car
car_time_L = 25
car_cost_L = 3
bus_time_L = 40 + 5
bus_cost_L = 0.8
income_L = 40000

car_time_H = 40
car_cost_H = 8
bus_time_H = 60 + 10
bus_cost_H = 2
income_H = 80000

time_coef = coef(model)[1]
cost_coef = coef(model)[2]
income_coef = coef(model)[3]

# get values of car/bus for group L
VC_L = exp(time_coef*car_time_L + cost_coef*car_cost_L + income_coef*income_L)
VB_L = exp(time_coef*bus_time_L + cost_coef*bus_cost_L + income_coef*income_L)
# get probability of choosing car for group L
share_car_L = VC_L / (VC_L + VB_L)
as.numeric(share_car_L)

# get values of car/bus for group H
VC_H = exp(time_coef*car_time_H + cost_coef*car_cost_H + income_coef*income_H)
VB_H = exp(time_coef*bus_time_H + cost_coef*bus_cost_H + income_coef*income_H)
# get probability of choosing car for group L
share_car_H = VC_H / (VC_H + VB_H)
as.numeric(share_car_H)
```

4) Original probabilities of choosing bus (before new policies):

```
> as.numeric(share_bus_L)
[1] 0.4230421
> as.numeric(share_bus_H)
[1] 0.6344398
```

a. Reducing bus fair by $0.50:

```
> as.numeric(share_bus_L)
[1] 0.4695355
> as.numeric(share_bus_H)
[1] 0.6769083
```

L: 0.47
H: 0.68

```
# a) reducing bus fare by 50 cents for both L and H
bus_cost_L = 0.8 - 0.5
bus_cost_H = 2 - 0.5
VC_L = exp(time_coef*car_time_L + cost_coef*car_cost_L + income_coef*income_L)
VB_L = exp(time_coef*bus_time_L + cost_coef*bus_cost_L + income_coef*income_L)
VC_H = exp(time_coef*car_time_H + cost_coef*car_cost_H + income_coef*income_H)
VB_H = exp(time_coef*bus_time_H + cost_coef*bus_cost_H + income_coef*income_H)
share_bus_L = VB_L / (VC_L + VB_L)
share_bus_H = VB_H / (VC_H + VB_H)
as.numeric(share_bus_L)
as.numeric(share_bus_H)
```

b. cutting bus wait times in half:

```
> as.numeric(share_bus_L)
[1] 0.4581105
> as.numeric(share_bus_H)
[1] 0.6976227
```

L: 0.48
H: 0.70

```
# b) bus waiting times cut in half
# reset bus cost
bus_cost_L = 0.8
bus_cost_H = 2
# change bus time
bus_time_L = 40 + 5/2
bus_time_H = 60 + 10/2
VC_L = exp(time_coef*car_time_L + cost_coef*car_cost_L + income_coef*income_L)
VB_L = exp(time_coef*bus_time_L + cost_coef*bus_cost_L + income_coef*income_L)
VC_H = exp(time_coef*car_time_H + cost_coef*car_cost_H + income_coef*income_H)
VB_H = exp(time_coef*bus_time_H + cost_coef*bus_cost_H + income_coef*income_H)
share_bus_L = VB_L / (VC_L + VB_L)
share_bus_H = VB_H / (VC_H + VB_H)
as.numeric(share_bus_L)
as.numeric(share_bus_H)
```

c. doubling parking costs:

```
> as.numeric(share_bus_L)
[1] 0.694116
> as.numeric(share_bus_H)
[1] 0.972453
```

L: 0.69
H: 0.97

```
# c) doubling parking costs
# reset bus time
bus_time_L = 40 + 5
bus_time_H = 60 + 10
# change car cost
car_cost_L = 3 * 2
car_cost_H = 8 * 2
VC_L = exp(time_coef*car_time_L + cost_coef*car_cost_L + income_coef*income_L)
VB_L = exp(time_coef*bus_time_L + cost_coef*bus_cost_L + income_coef*income_L)
VC_H = exp(time_coef*car_time_H + cost_coef*car_cost_H + income_coef*income_H)
VB_H = exp(time_coef*bus_time_H + cost_coef*bus_cost_H + income_coef*income_H)
share_bus_L = VB_L / (VC_L + VB_L)
share_bus_H = VB_H / (VC_H + VB_H)
as.numeric(share_bus_L)
as.numeric(share_bus_H)
```

Doubling parking costs was the most effective in increasing bus ridership.

Problem 2

1) S1: iPhone 100 -- R(S1) = **428.57**

```
> r1 = r_i100 * w_i100 / (1 + w_i100)
> r1
[1] 428.5714
```

```
# PROBLEM 2

# 1) get revenues for each possible RO assortment
w_gs99 = 2
r_gs99 = 400
w_gs100 = 2.2
r_gs100 = 499
w_i99 = 2.1
r_i99 = 500
w_i100 = 2.5
r_i100 = 600
# RO set 1: iphone 100
r1 = r_i100 * w_i100 / (1 + w_i100)
r1
```

S2: iPhone 100, iPhone 99 -- R(S2) = **455.36**

```
> r2
[1] 455.3571
```

```
# RO set 2: iphone 100, iphone 99
r2 = (r_i100 * w_i100 + r_i99 * w_i99) / (1 + w_i100 + w_i99)
r2
```

S3: iPhone 100, iPhone 99, Galaxy S100 -- R(S3) = **467.67**

```
> r3
[1] 467.6667
```

```
# RO set 3: iphone 100, iphone 99, galaxy s100
r3 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100) /
       (1 + w_i100 + w_i99 + w_gs100)
r3
```

S4: iPhone 100, iPhone 99, Galaxy S100, Galaxy S99 -- R(S4) = **453.86**

```
> r4
[1] 453.8571
```

```
# RO set 4: iphone 100, iphone 99, galaxy s100, galaxy s99
r4 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100 + r_gs99 * w_gs99) /
  (1 + w_i100 + w_i99 + w_gs100 + w_gs99)
r4
```

The assortment that maximizes revenue is S3: iPhone 100, iPhone 99, Galaxy S100

2) The new revenue is **415.77**. This is lower than the expected revenue from the first part because now there is uncertainty with the type of customers that might show up -- since the assortment contains 2 iPhones and 1 Galaxy, if for example, only Galaxy customers show up, we could be losing out on revenue because they will very likely only choose to buy the Galaxy S100.

```
# 2) calculate expected revenue from optimal assortment with NEVER customers
w_gs99 = 0.5 * (0.1 + 2)
r_gs99 = 400
w_gs100 = 0.5 * (0.1 + 2.2)
r_gs100 = 499
w_i99 = 0.5 * (2.1 + 0.01)
r_i99 = 500
w_i100 = 0.5 * (2.5 + 0.01)
r_i100 = 600

r3 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100) /
  (1 + w_i100 + w_i99 + w_gs100)
r3
```

```
> r3
[1] 415.7735
```

3) a. **Never Galaxy Customers:**

```
# NEVER Galaxy customers:
w_gs99 = 0.1
r_gs99 = 400
w_gs100 = 0.1
r_gs100 = 499
w_i99 = 2.1
r_i99 = 500
w_i100 = 2.5
r_i100 = 600
```

S1: iPhone 100 -- R(S1) = **428.57**

```
> r1 = r_i100 * w_i100 / (1 + w_i100)
> r1
[1] 428.5714
```

```
# RO set 1: iphone 100
r1 = r_i100 * w_i100 / (1 + w_i100)
r1
```

S2: iPhone 100, iPhone 99 -- R(S2) = **455.36**

```
> r2
[1] 455.3571
```

```
# RO set 2: iphone 100, iphone 99
r2 = (r_i100 * w_i100 + r_i99 * w_i99) / (1 + w_i100 + w_i99)
r2
```

S3: iPhone 100, iPhone 99, Galaxy S100 -- R(S3) = **456.12**

```
> r3
[1] 456.1228
```

```
# RO set 3: iphone 100, iphone 99, galaxy s100
r3 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100) /
  (1 + w_i100 + w_i99 + w_gs100)
r3
```

S4: iPhone 100, iPhone 99, Galaxy S100, Galaxy S99 -- R(S4) = **455.16**

```
> r4
[1] 455.1552
```

```
# RO set 4: iphone 100, iphone 99, galaxy s100, galaxy s99
r4 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100 + r_gs99 * w_gs99) /
  (1 + w_i100 + w_i99 + w_gs100 + w_gs99)
r4
```

For NEVER Galaxy customers, the optimal assortment is still **S3: iPhone 100, iPhone 99, Galaxy S100** at a revenue of 456.12.

NEVER iPhone Customers:

```
# NEVER iPhone customers:
w_gs99 = 2
r_gs99 = 400
w_gs100 = 2.2
r_gs100 = 499
w_i99 = 0.01
r_i99 = 500
w_i100 = 0.01
r_i100 = 600
```

S1: iPhone 100 -- R(S1) = **5.94**

```
> r1
[1] 5.940594
```

```
# RO set 1: iphone 100
r1 = r_i100 * w_i100 / (1 + w_i100)
r1
```

S2: iPhone 100, iPhone 99 -- R(S2) = **10.78**

```
> r2
[1] 10.78431
```

```
# RO set 2: iphone 100, iphone 99
r2 = (r_i100 * w_i100 + r_i99 * w_i99) / (1 + w_i100 + w_i99)
r2
```

S3: iPhone 100, iPhone 99, Galaxy S100 -- R(S3) = **344.35**

```
> r3
[1] 344.3478
```

```
# RO set 3: iphone 100, iphone 99, galaxy s100
r3 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100) /
  (1 + w_i100 + w_i99 + w_gs100)
r3
```

S4: iPhone 100, iPhone 99, Galaxy S100, Galaxy S99 -- R(S4) = **365.67**

```
> r4
[1] 365.6705
```

```
# RO set 4: iphone 100, iphone 99, galaxy s100, galaxy s99
r4 = (r_i100 * w_i100 + r_i99 * w_i99 + r_gs100 * w_gs100 + r_gs99 * w_gs99) /
  (1 + w_i100 + w_i99 + w_gs100 + w_gs99)
r4
```

For NEVER iPhone customers, the optimal assortment is **S4: iPhone 100, iPhone 99, Galaxy S100, Galaxy S99** at a revenue of 365.67.

b. The expected revenue when showing each type their optimal assortment is (365.67 + 456.12) / 2 = **410.90**