Homework Assignment 2 – Group No 8

Question 1: Use ToyotaCorolla dataset.
Question 1 Part a:

- Predicting price using Fuel type and HP as the predictor variables.
- Model 1 (Price ~ Fuel_type.Petrol+Fuel_type.Diesel+HP)

- Adjusted R Square: 0.1925
- Residual Standard Error: 3356
- F-statistic: 69.34
- RMSE: 3252.051

```
> summary(model1.regressor)

Call:
lm(formula = Price ~ ., data = model1.train.data)

Residuals:
   Min     1Q Median     3Q    Max 
 -6541  -2250   -550   1250  15765 

Coefficients:
                  Estimate Std. Error t value             Pr(>|t|)    
(Intercept)      -5335.956   1636.080  -3.261              0.00115 ** 
Fuel_Type.Petrol  2350.017   1275.508   1.842              0.06576 .  
Fuel_Type.Diesel  6794.849   1348.946   5.037          0.000000576 ***
HP                 131.691      9.394  14.019 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3356 on 857 degrees of freedom
Multiple R-squared:  0.1953,	Adjusted R-squared:  0.1925 
F-statistic: 69.34 on 3 and 857 DF,  p-value: < 0.00000000000000022
```

```
> head(model1.valid.res)
   model1.valid.data.Price model1.pred  residuals
1                    13500    13311.04   188.9617
2                    13750    13311.04   438.9617
4                    14950    13311.04  1638.9617
9                    21500    22298.64  -798.6390
10                   12950    10545.54  2404.4623
12                   19950    22298.64 -2348.6390
> 
```

```
> accuracy(model1.pred,model1.valid.data$Price)
                ME     RMSE      MAE       MPE     MAPE
Test set -254.8523 3252.051 2458.617 -9.703865 23.92214
> 
```

- Model 2 (Price ~ Fuel_type.Petrol+Fuel_type.CNG+HP)
- Adjusted R Square: 0.1925
- Residual Standard Error: 3356
- F-statistic: 69.34
- RMSE: 3252.051

```
> summary(model2.regressor)

Call:
lm(formula = Price ~ ., data = model2.train.data)

Residuals:
   Min     1Q Median     3Q    Max 
 -6541  -2250   -550   1250  15765 

Coefficients:
                 Estimate Std. Error t value             Pr(>|t|)    
(Intercept)      1458.893    824.026   1.770                0.077 .  
Fuel_Type.Petrol -4444.831    442.371 -10.048 < 0.0000000000000002 ***
Fuel_Type.CNG    -6794.849   1348.946  -5.037          0.000000576 ***
HP                131.691      9.394  14.019 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3356 on 857 degrees of freedom
Multiple R-squared:  0.1953,	Adjusted R-squared:  0.1925 
F-statistic: 69.34 on 3 and 857 DF,  p-value: < 0.00000000000000022
```

```
> head(model2.valid.res)
   model2.valid.data.Price model2.pred  residuals
1                    13500    13311.04   188.9617
2                    13750    13311.04   438.9617
4                    14950    13311.04  1638.9617
9                    21500    22298.64  -798.6390
10                   12950    10545.54  2404.4623
12                   19950    22298.64 -2348.6390
> 
```

```
> accuracy(model2.pred,model2.valid.data$Price)
                ME     RMSE      MAE       MPE     MAPE
Test set -254.8523 3252.051 2458.617 -9.703865 23.92214
> 
```

- Adjusted R Square, RSE, F-statistic, RMSE obtained are same with both the models.
- We converted Fuel type variable into dummy variable leading to three dummy categories: petrol, diesel and CNG. If we use a dummy variable as a predictor, R automatically drops one of the columns as its value is redundant (in case of three dummy variables, if we know two values, we automatically know the third one)
- In the problem, we manually dropped one dummy columns for each model, resulting in practically similar models with the same values as the predictor variables are same in both the models.

Question 1 Part b:

- Predictor variables: Age_08_04, KM, HP, Met_Color, Automatic, CC, Doors, Weight (Quarterly tax excluded from the predictor variable as its value is part of the Price being predicted)
- Regression Model (categorical variables used as continuous variables):

```
> str(model3.factor.dataset)
'data.frame':	1436 obs. of  9 variables:
 $ Price    : int  13500 13750 13950 14950 13750 12950 16900 18600 21500 12950 ...
 $ Age_08_04: int  23 23 24 26 30 32 27 30 27 23 ...
 $ KM       : int  46986 72937 41711 48000 38500 61000 94612 75889 19700 71138 ...
 $ HP       : int  90 90 90 90 90 90 90 90 192 69 ...
 $ Met_Color: int  1 1 1 0 0 0 1 1 0 0 ...
 $ Automatic: int  0 0 0 0 0 0 0 0 0 0 ...
 $ CC       : int  2000 2000 2000 2000 2000 2000 2000 2000 1800 1900 ...
 $ Doors    : int  3 3 3 3 3 3 3 3 3 3 ...
 $ Weight   : int  1165 1165 1165 1165 1170 1170 1245 1245 1185 1105 ...
>
```

```
> summary(model3.regressor)

Call:
lm(formula = Price ~ ., data = model3.train.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-10062.1   -751.1     -8.2    752.4   6244.0 

Coefficients:
               Estimate  Std. Error t value             Pr(>|t|)    
(Intercept) -4630.93345  1162.87091  -3.982            0.0000741 ***
Age_08_04    -123.63191     3.51432 -35.179 < 0.0000000000000002 ***
KM             -0.01818     0.00167 -10.886 < 0.0000000000000002 ***
HP             34.86139     3.54260   9.841 < 0.0000000000000002 ***
Met_Color     122.38874   101.38803   1.207               0.2277    
Automatic     457.24111   220.72730   2.072               0.0386 *  
CC             -0.02999     0.09461  -0.317               0.7514    
Doors         -38.52472    52.06367  -0.740               0.4595    
Weight         18.70885     1.02887  18.184 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1385 on 852 degrees of freedom
Multiple R-squared:  0.8637,	Adjusted R-squared:  0.8624 
F-statistic: 674.7 on 8 and 852 DF,  p-value: < 0.00000000000000022
```

```
>
> head(model3.valid.res)
  model3.valid.data.Price model3.pred  residuals
1                   13500    16551.49 -3051.4949
2                   13750    16079.70 -2329.6989
4                   14950    16039.78 -1089.7756
9                   21500    20366.68  1133.3187
10                  12950    14138.39 -1188.3947
12                  19950    20550.15  -600.1508
>
```

```
> model3.accuracy
                ME     RMSE      MAE        MPE     MAPE
Test set 40.39014 1302.479 1031.959 -0.5967672 10.30058
>
```

- RMSE: 1302.479
- RMSE (obtained in class): 1228

- Regression Model (categorical variables, Met_Color, Automatic, Doors converted to factors)

```
> str(model3.factor.dataset)
'data.frame':	1436 obs. of  9 variables:
 $ Price    : int  13500 13750 13950 14950 13750 12950 16900 18600 21500 12950 ...
 $ Age_08_04: int  23 23 24 26 30 32 27 30 27 23 ...
 $ KM       : int  46986 72937 41711 48000 38500 61000 94612 75889 19700 71138 ...
 $ HP       : int  90 90 90 90 90 90 90 90 192 69 ...
 $ Met_Color: Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 2 1 1 ...
 $ Automatic: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ CC       : int  2000 2000 2000 2000 2000 2000 2000 2000 1800 1900 ...
 $ Doors    : Factor w/ 4 levels "2","3","4","5": 2 2 2 2 2 2 2 2 2 2 ...
 $ Weight   : int  1165 1165 1165 1165 1170 1170 1245 1245 1185 1105 ...
>
```

```
> summary(model3.regressor)

Call:
lm(formula = Price ~ ., data = model3.train.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-10062.8   -749.7     -8.9    754.5   6243.3 

Coefficients:
                Estimate   Std. Error t value            Pr(>|t|)    
(Intercept) -4332.641663  1806.807417  -2.398              0.0167 *  
Age_08_04    -123.633857     3.520377 -35.119 <0.0000000000000002 ***
KM             -0.018179     0.001677 -10.838 <0.0000000000000002 ***
HP             34.894001     3.553255   9.820 <0.0000000000000002 ***
Met_Color1    121.580717   101.699615   1.195              0.2322    
Automatic1    457.991480   221.780165   2.065              0.0392 *  
CC             -0.029734     0.094766  -0.314              0.7538    
Doors3       -418.056796  1390.725265  -0.301              0.7638    
Doors4       -458.314371  1397.202889  -0.328              0.7430    
Doors5       -493.190375  1391.861872  -0.354              0.7232    
Weight         18.708693     1.043752  17.924 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1387 on 850 degrees of freedom
Multiple R-squared:  0.8637,	Adjusted R-squared:  0.8621 
F-statistic: 538.6 on 10 and 850 DF,  p-value: < 0.00000000000000022
```

```
> head(model3.valid.res)
   model3.valid.data.Price model3.pred  residuals
1                    13500    16549.78 -3049.7809
2                    13750    16078.03 -2328.0270
4                    14950    16038.87 -1088.8655
9                    21500    20369.00  1131.0041
10                   12950    14136.83 -1186.8274
12                   19950    20552.51  -602.5139
> model3.accuracy=accuracy(model3.pred,model3.valid.data$Price)
> model3.accuracy
              ME     RMSE      MAE        MPE     MAPE
Test set 40.1255 1302.377 1031.429 -0.6001345 10.29391
> 
```

- RMSE: 1302.377
- When the categorical variables (Met_Color, Automatic and Doors) are used as continuous predictor variables, RMSE value is 1302.479 while when the categorical variables are converted to factors, RMSE value obtained is 1302.377, indicating a very small decrease (0.102). Hence, we can say that there is a slight improvement in the regression model when categorical variables are used as factors.

Question 2: Use Airfares dataset. You may ignore the first 4 variables.

Question 2 Part a:

- Numerical predictors: HI, S_INCOME, E_INCOME, S_POP, E_POP, COUPON, DISTANCE, PAX
- Calculating mean, standard deviation, median, length, minimum value, maximum value and number of missing values for the numerical predictors.

```
> data.frame(mean=sapply(airfares.numeric.dataset, mean),
+            sd=sapply(airfares.numeric.dataset, sd),
+            min=sapply(airfares.numeric.dataset, min),
+            max=sapply(airfares.numeric.dataset, max),
+            median=sapply(airfares.numeric.dataset, median),
+            length=sapply(airfares.numeric.dataset, length),
+            miss.val=sapply(airfares.numeric.dataset, function(x)
+              sum(length(which(is.na(x))))))
                 mean           sd       min        max      median length miss.val
FARE     1.608767e+02 7.602244e+01     42.47     402.02     144.600    638        0
HI       4.442141e+03 1.724267e+03   1230.48   10000.00    4208.185    638        0
S_INCOME 2.775986e+04 3.596208e+03  14600.00   38813.00   28637.000    638        0
E_INCOME 2.766373e+04 4.611325e+03  14600.00   38813.00   26409.000    638        0
S_POP    4.557004e+06 3.010985e+06  29838.00 9056076.00 3532657.000    638        0
E_POP    3.194503e+06 2.735604e+06 111745.00 9056076.00 2195215.000    638        0
COUPON   1.202335e+00 2.038207e-01      1.00       1.94       1.150    638        0
DISTANCE 9.756536e+02 6.462424e+02    114.00    2764.00     850.000    638        0
PAX      1.278221e+04 1.320223e+04   1504.00   73892.00    7792.000    638        0
```

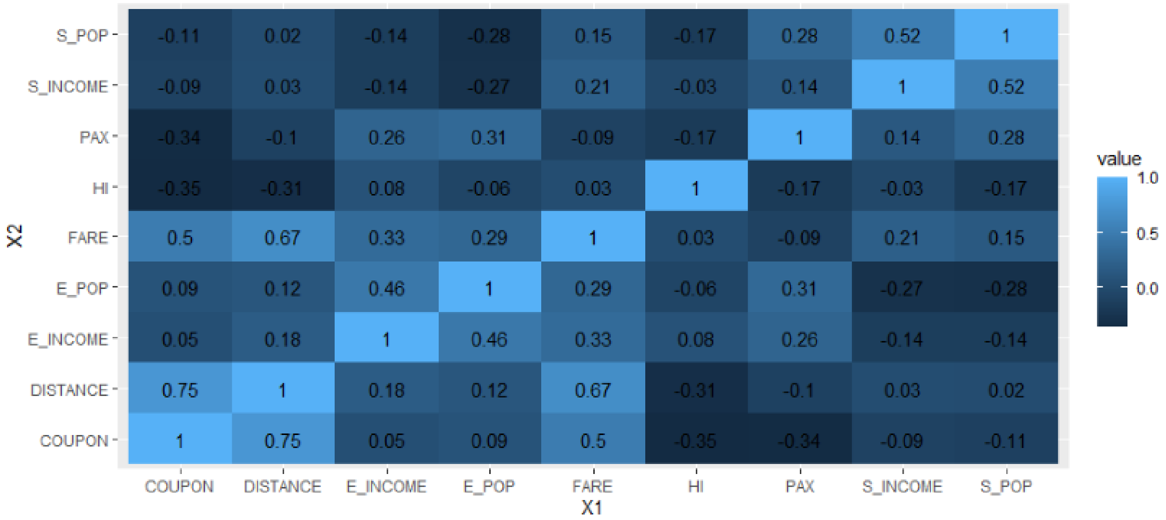
- Correlations between FARE and numerical predictors

```
> #Using cor() function to interpret the relationship between FARE and other numerical predictor variables
> round(cor(airfares.numeric.dataset),2)
          FARE    HI S_INCOME E_INCOME S_POP E_POP COUPON DISTANCE   PAX
FARE      1.00  0.03     0.21     0.33  0.15  0.29   0.50     0.67 -0.09
HI        0.03  1.00    -0.03     0.08 -0.17 -0.06  -0.35    -0.31 -0.17
S_INCOME  0.21 -0.03     1.00    -0.14  0.52 -0.27  -0.09     0.03  0.14
E_INCOME  0.33  0.08    -0.14     1.00 -0.14  0.46   0.05     0.18  0.26
S_POP     0.15 -0.17     0.52    -0.14  1.00 -0.28  -0.11     0.02  0.28
E_POP     0.29 -0.06    -0.27     0.46 -0.28  1.00   0.09     0.12  0.31
COUPON    0.50 -0.35    -0.09     0.05 -0.11  0.09   1.00     0.75 -0.34
DISTANCE  0.67 -0.31     0.03     0.18  0.02  0.12   0.75     1.00 -0.10
PAX      -0.09 -0.17     0.14     0.26  0.28  0.31  -0.34    -0.10  1.00
>
```

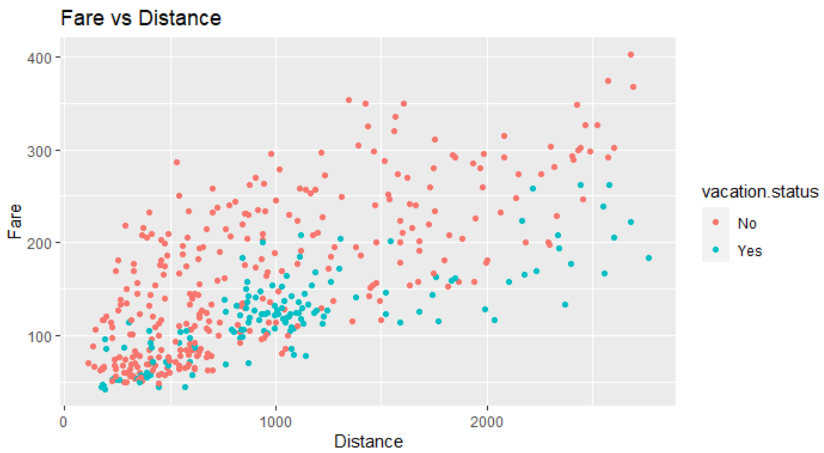
Interpretation from correlation values:

- FARE has a moderate positive correlation with DISTANCE, with the increase in distance the fare is going to increase.
- FARE has low negative correlation with PAX, with the increase in number of passengers on that route the fare is going to decrease.
- DISTANCE has a strong positive linear relationship with COUPON

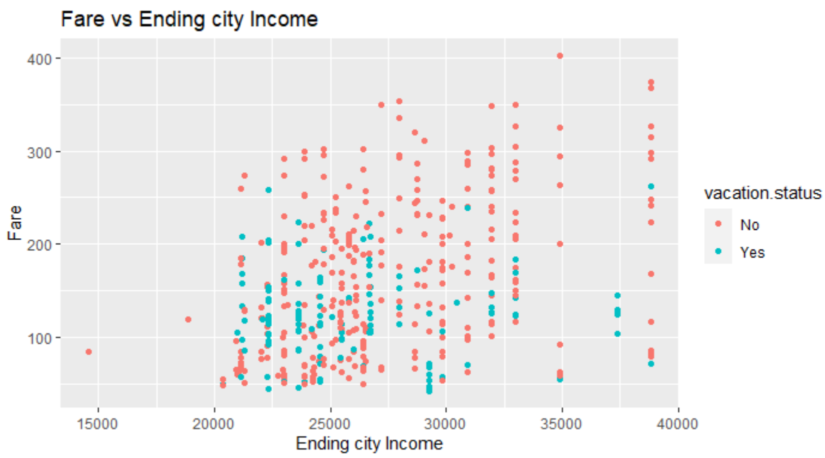
- Heatmap between FARE and other numerical predictor varibales



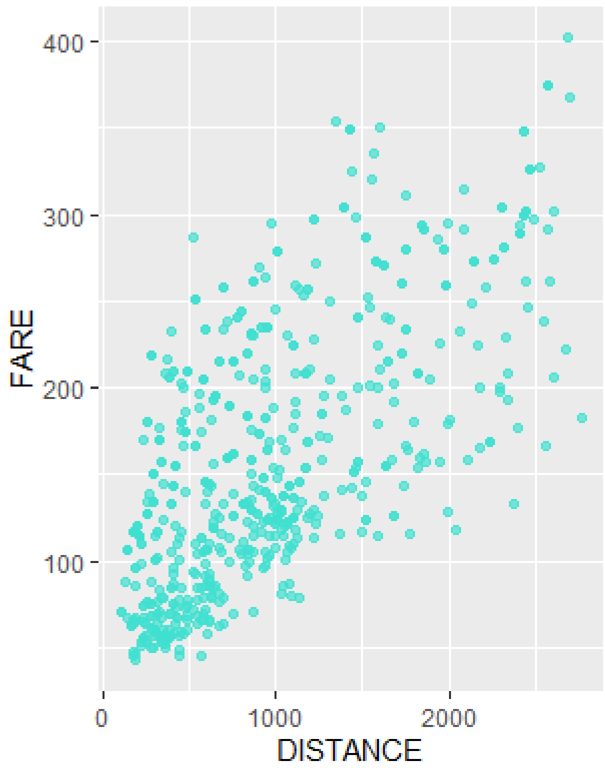
- Scatterplot: FARE vs DISTANCE coded by vacation



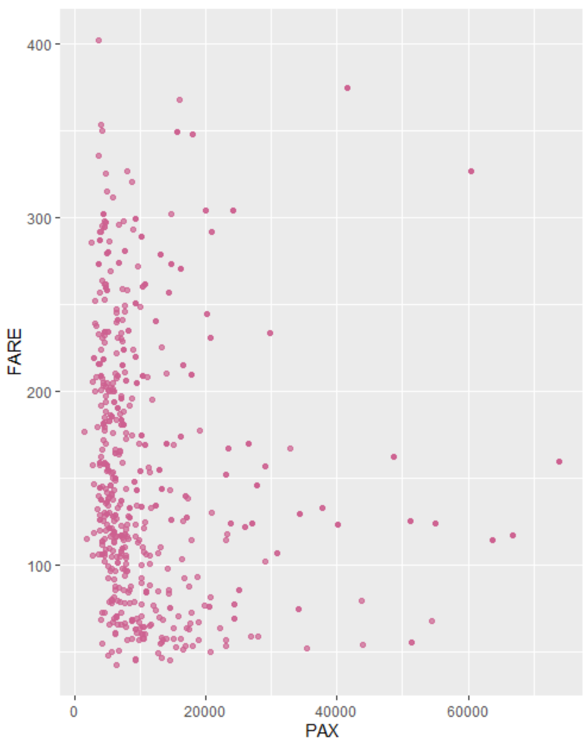
- Scatterplot: FARE vs E_INCOME coded by vacation



- Scatterplot: FARE vs DISTANCE (highest correlation value of 0.67)



- Scatterplot: FARE vs PAX (low negative correlation of -0.09)



Question 2 Part b:

- Categorical predictor variable VACATION and SW with FARE

```
> aggregate(airfares.dataset$FARE, by=list(VACATION=airfares.dataset$VACATION),
+                                         FUN=mean)
  VACATION        x
1       No 173.5525
2      Yes 125.9809
>
> #Analyzing the interaction between FARE and SW status(if SW serves the route)
> aggregate(airfares.dataset$FARE, by=list(SW=airfares.dataset$SW),
+           FUN=mean)
   SW         x
1  No 188.18279
2 Yes  98.38227
> 
```

- Pivot table for VACATION+SW and FARE

```
> cast(vac.sw.mlt, VACATION ~ SW, subset=variable=="FARE", margins=c("grand_row", "grand_col"), mean)
  VACATION       No       Yes    (all)
1       No 204.3866 100.57101 173.5525
2      Yes 141.8257  92.85073 125.9809
3    (all) 188.1828  98.38227 160.8767
> 
```

- For category No for VACATION and SW, average FARE is 204.3866. The average FARE price is high (204.3866) when carrier is non-SW airline compared with average FARE (100.57101) when carrier is SW airline on a non a vacation route
- For category YES for VACATION and SW, average FARE decreases to 92.85073. On a VACATION route or a non-vacation route, SW airline serving that route has a low average FARE (98.38227) compared to other airlines serving the same route (188.1828)
- For categorical variable VACATION and SW, the overall average FARE is 160.8767

- Categorical predictor variable SLOT and GATE with FARE

```
> aggregate(airfares.dataset$FARE, by=list(SLOT=airfares.dataset$SLOT),
+           FUN=mean)
        SLOT        x
1 Controlled 186.0594
2       Free 150.8257
>
> #Analyzing the interaction between FARE and GATE status
> aggregate(airfares.dataset$FARE, by=list(GATE=airfares.dataset$GATE),
+           FUN=mean)
         GATE       x
1 Constrained 193.129
2        Free 153.096
```

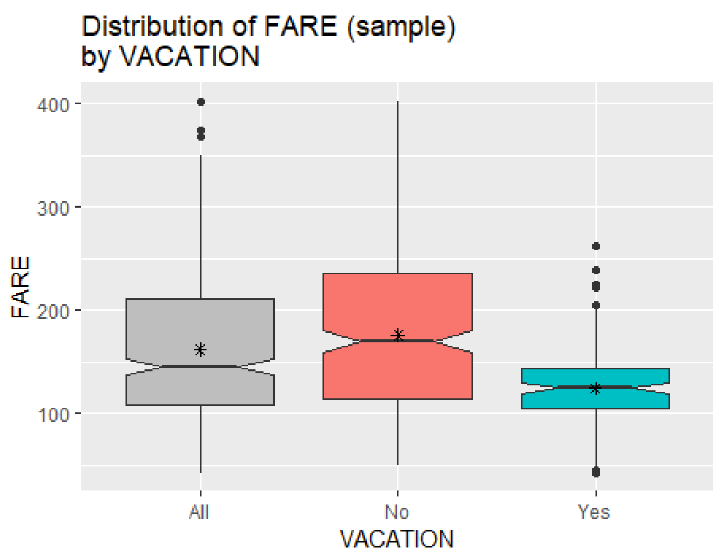
- Pivot table for SLOT+GATE and FARE

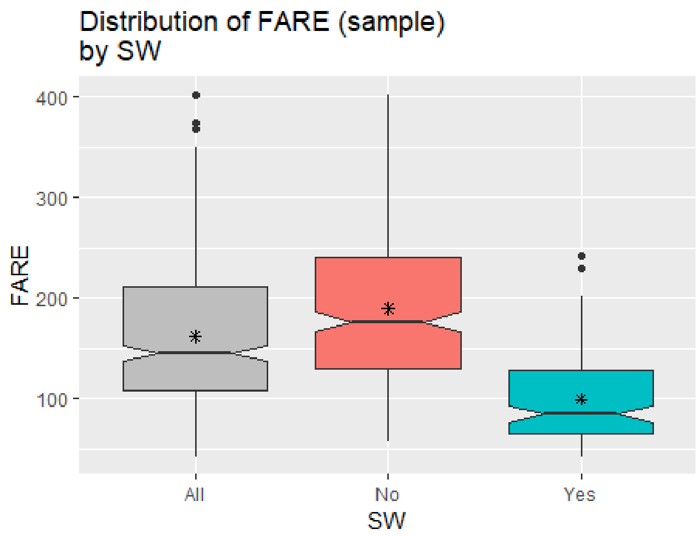
```
> cast(slot.gate.mlt, SLOT ~ GATE, subset=variable=="FARE",
+      margins=c("grand_row", "grand_col"), mean)
        SLOT Constrained     Free    (all)
1 Controlled    199.8232 184.4550 186.0594
2       Free    191.9177 138.5332 150.8257
3      (all)    193.1290 153.0960 160.8767
>
```

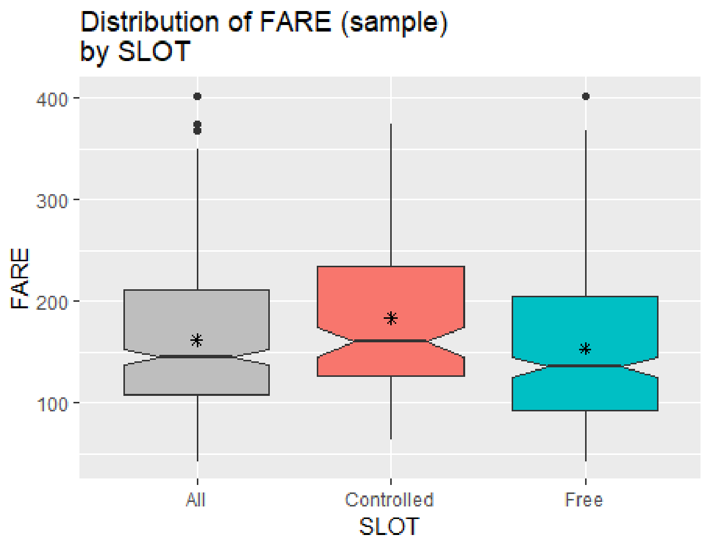
- For category Controlled for SLOT and Constrained for GATE, average FARE is 199.8232
- For category Free for SLOT and GATE, average FARE decreases to 138.5332
- For categorical variable SLOT and GATE, the overall average FARE is 160.8767 which is similar to when we used categorical variable VACATION and SW

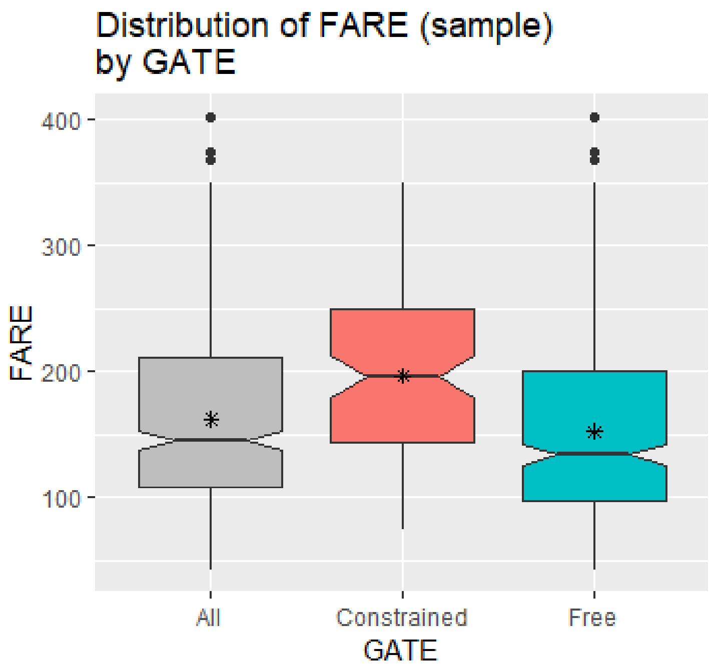
.

- Boxplots of FARE with individual categorical variables:




Distribution of FARE (sample)
by SW
FARE
400
300
200
100
All
No
Yes
SW


Distribution of FARE (sample)
by SLOT
FARE
400
300
200
100
All
Controlled
Free
SLOT


Distribution of FARE (sample)
by GATE
FARE
400
300
200
100
All
Constrained
Free
GATE

Question 2 Part c:

- Partitioning the dataset into training data and validation data

```
> dim(airfaremodel.train.data)
[1] 382  14
> dim(airfaremodel.valid.data)
[1] 256  14
>
```

- Exhaustive search algorithm on the training dataset

```
> print(models.list.summary$which)
   (Intercept) COUPON  NEW1  NEW2  NEW3 VACATIONYes SWYes    HI S_INCOME E_INCOME S_POP E_POP SLOTFree GATEFree
1         TRUE  FALSE FALSE FALSE FALSE       FALSE FALSE FALSE    FALSE    FALSE FALSE FALSE    FALSE    FALSE
2         TRUE  FALSE FALSE FALSE FALSE       FALSE  TRUE FALSE    FALSE    FALSE FALSE FALSE    FALSE    FALSE
3         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE FALSE    FALSE    FALSE FALSE FALSE    FALSE    FALSE
4         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE FALSE    FALSE    FALSE
5         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE FALSE    FALSE     TRUE
6         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE FALSE     TRUE     TRUE
7         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE FALSE FALSE     TRUE     TRUE
8         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE  TRUE    FALSE    FALSE
9         TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE  TRUE    FALSE     TRUE
10        TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE    FALSE     TRUE
11        TRUE  FALSE FALSE FALSE FALSE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE     TRUE
12        TRUE   TRUE FALSE FALSE FALSE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE     TRUE
13        TRUE   TRUE FALSE FALSE  TRUE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE     TRUE
14        TRUE   TRUE  TRUE FALSE  TRUE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE     TRUE
   DISTANCE   PAX
1      TRUE FALSE
2      TRUE FALSE
3      TRUE FALSE
4      TRUE FALSE
5      TRUE FALSE
6      TRUE FALSE
7      TRUE FALSE
8      TRUE  TRUE
9      TRUE  TRUE
10     TRUE  TRUE
11     TRUE  TRUE
12     TRUE  TRUE
13     TRUE  TRUE
14     TRUE  TRUE
> # showing Ad.Rsquare values for each model
> print(models.list.summary$adjr2)
 [1] 0.4495375 0.6154761 0.7097192 0.7487321 0.7607563 0.7761915 0.7797985 0.7888046 0.7965958 0.8020897 0.8052561
[12] 0.8051470 0.8046648 0.8042696
```

- Adjusted R square: 1 < 2 < 3 < 4< 5 < 6 < 7 < 8 < **9 < 10 < 11** > 12 > 13 > 14

1. Model 1 (9 variables: VACATION, SW, HI, E_INCOME, S_POP, E_POP, DISTANCE, PAX, GATE)

```
Coefficients:
                 Estimate     Std. Error t value            Pr(>|t|)    
(Intercept)   2.0334000962  16.1079822515   0.126             0.89961    
VACATIONYes -37.1549557339   4.9183549328  -7.554    0.000000000000330 ***
SWYes       -48.9925012628   4.6181487210 -10.609 < 0.0000000000000002 ***
HI            0.0093319621   0.0012080072   7.725    0.000000000000105 ***
E_INCOME      0.0022222578   0.0004687706   4.741    0.000003039391331 ***
S_POP         0.0000055468   0.0000007947   6.980    0.000000000013663 ***
E_POP         0.0000041253   0.0000009333   4.420    0.000012962253684 ***
DISTANCE      0.0741066495   0.0030973270  23.926 < 0.0000000000000002 ***
PAX          -0.0010392565   0.0001753623  -5.926    0.000000007066983 ***
GATEFree    -18.8947251296   4.8325088742  -3.910             0.00011 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 35.7 on 372 degrees of freedom
Multiple R-squared:  0.8014,    Adjusted R-squared:  0.7966 
F-statistic: 166.8 on 9 and 372 DF,  p-value: < 0.00000000000000022

   airfaremodel.valid.data.FARE airfaremodel.pred  residuals
4                         85.47         117.38624 -31.916241
6                         56.76          83.09909 -26.339093
7                        228.00         244.40534 -16.405339
8                        116.54         115.40068   1.139321
9                        172.63         178.79599  -6.165986
10                       114.76         130.35410 -15.594096
                ME     RMSE      MAE       MPE     MAPE
Test set -3.837492 37.95085 29.21489 -6.296362 24.2167
                             Model ends
```

- Model 1 RMSE using valid data: 37.95085

2. Model 2 (10 variables: VACATION, SW, HI, E_INCOME, S_POP, E_POP, DISTANCE, PAX, GATE, S_INCOME)

```
Coefficients:
                  Estimate     Std. Error t value            Pr(>|t|)    
(Intercept) -60.6435472442  24.4804215857  -2.477            0.013686 *  
VACATIONYes -33.7611927728   4.9551708250  -6.813    0.000000000038662 ***
SWYes       -43.3689678509   4.8521435773  -8.938 < 0.0000000000000002 ***
HI            0.0091977880   0.0011922483   7.715    0.000000000000113 ***
E_INCOME      0.0024041108   0.0004655430   5.164    0.000000395007314 ***
S_POP         0.0000047950   0.0000008151   5.883    0.000000009024049 ***
E_POP         0.0000049682   0.0000009541   5.207    0.000000318102615 ***
DISTANCE      0.0735714085   0.0030593483  24.048 < 0.0000000000000002 ***
PAX          -0.0011631440   0.0001768513  -6.577    0.000000000163501 ***
GATEFree    -18.3941583746   4.7691197531  -3.857            0.000135 ***
S_INCOME      0.0020941653   0.0006222469   3.365            0.000844 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 35.22 on 371 degrees of freedom
Multiple R-squared:  0.8073,	Adjusted R-squared:  0.8021 
F-statistic: 155.4 on 10 and 371 DF,  p-value: < 0.00000000000000022

   airfaremodel.valid.data.FARE airfaremodel.pred residuals
4                         85.47         123.87277 -38.40277
6                         56.76          88.58242 -31.82242
7                        228.00         249.23415 -21.23415
8                        116.54         126.70654 -10.16654
9                        172.63         185.58120 -12.95120
10                       114.76         135.82933 -21.06933
                ME     RMSE      MAE       MPE     MAPE
Test set -4.012774 38.14996 29.54923 -6.486008 24.48618
```

- Model 2 RMSE using valid data: 38.14996

3. Model 3 (11 variables: VACATION, SW, HI, E_INCOME, S_POP, E_POP, DISTANCE, PAX, GATE, S_INCOME, SLOT)

```
Coefficients:
                 Estimate     Std. Error t value            Pr(>|t|)
(Intercept) -33.4966292925  26.3533937251  -1.271             0.20451
VACATIONYes -34.3350683659   4.9201337810  -6.978 0.000000000013857896 ***
SWYes       -41.4733728683   4.8659644013  -8.523 0.000000000000000399 ***
HI            0.0097087603   0.0011982666   8.102 0.000000000000007932 ***
E_INCOME      0.0022328686   0.0004662970   4.789 0.000002434450315483 ***
S_POP         0.0000041386   0.0000008456   4.894 0.000001476856966589 ***
E_POP         0.0000042189   0.0000009877   4.272 0.000024716165838751 ***
DISTANCE      0.0753095733   0.0031047542  24.256 < 0.0000000000000002 ***
PAX          -0.0011197662   0.0001761919  -6.355 0.000000000611854341 ***
GATEFree    -22.6016837311   4.9897995646  -4.530 0.000007984533428892 ***
S_INCOME      0.0017514156   0.0006306365   2.777             0.00576 **
SLOTFree    -12.9105872030   4.8685844144  -2.652             0.00835 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.94 on 370 degrees of freedom
Multiple R-squared:  0.8109,    Adjusted R-squared:  0.8053
F-statistic: 144.2 on 11 and 370 DF,  p-value: < 0.00000000000000022

   airfaremodel.valid.data.FARE airfaremodel.pred  residuals
4                         85.47         126.58696 -41.116956
6                         56.76          82.51034 -25.750341
7                        228.00         242.76278 -14.762782
8                        116.54         122.17161  -5.631613
9                        172.63         180.13237  -7.502367
10                       114.76         130.66911 -15.909114
                ME     RMSE      MAE       MPE     MAPE
Test set -3.604363 37.43276 29.13853 -5.886503 24.09033
```

- Model 3 RMSE using valid data: 37.43276
- RMSE: Model 3 < Model 1 < Model 2
- We choose model 3 with predictor variables VACATION, SW, HI, E_INCOME, S_POP, E_POP, DISTANCE, PAX, GATE, S_INCOME, SLOT for predicting FARE as the best model based on valid data accuracy metrics, having the lowest RMSE value.
- Considering the fact that we should keep the models parsimonious, and also increase in number of predictor variables increases the probability of missing values and the cost of a data collection, we can choose model 1 with 9 predictor variables as the best model for prediction. Model 1 and model 3 has nearly similar RMSE values as well.

- Criterias to choose/drop predictor variables:
    a. To keep the model parsimonious, bias-variance trade-off
    b. Multicollinearity
    c. More predictor variables, higher chances of missing values, cost of data collection, increase in the number of measurements for a new subject to use the model

- Backward search algorithm on the training data set gives the same model as model 3, RMSE value is the same as that of model 3

```
Coefficients:
                 Estimate     Std. Error t value            Pr(>|t|)
(Intercept) -33.4966292925  26.3533937251  -1.271             0.20451
VACATIONYes -34.3350683659   4.9201337810  -6.978 0.000000000013857896 ***
SWYes       -41.4733728683   4.8659644013  -8.523 0.000000000000000399 ***
HI            0.0097087603   0.0011982666   8.102 0.000000000000007932 ***
S_INCOME      0.0017514156   0.0006306365   2.777             0.00576 **
E_INCOME      0.0022328686   0.0004662970   4.789 0.000002434450315483 ***
S_POP         0.0000041386   0.0000008456   4.894 0.000001476856966589 ***
E_POP         0.0000042189   0.0000009877   4.272 0.000024716165838751 ***
SLOTFree    -12.9105872030   4.8685844144  -2.652             0.00835 **
GATEFree    -22.6016837311   4.9897995646  -4.530 0.000007984533428892 ***
DISTANCE      0.0753095733   0.0031047542  24.256 < 0.0000000000000002 ***
PAX          -0.0011197662   0.0001761919  -6.355 0.000000000611854341 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.94 on 370 degrees of freedom
Multiple R-squared:  0.8109,	Adjusted R-squared:  0.8053
F-statistic: 144.2 on 11 and 370 DF,  p-value: < 0.00000000000000022
```

```
   airfaremodel.valid.data.FARE airfaremodel.lm.predbkwd  residuals
4                         85.47                126.58696 -41.116956
6                         56.76                 82.51034 -25.750341
7                        228.00                242.76278 -14.762782
8                        116.54                122.17161  -5.631613
9                        172.63                180.13237  -7.502367
10                       114.76                130.66911 -15.909114
> #Calculating accuracy parameters based on above prediction
> accuracy(airfaremodel.lm.predbkwd, airfaremodel.valid.data$FARE)
                ME     RMSE      MAE       MPE     MAPE
Test set -3.604363 37.43276 29.13853 -5.886503 24.09033
```

- Forward search algorithm gives the same model as model 3, RMSE value is the same as that of model 3

```
Coefficients:
                  Estimate       Std. Error t value                Pr(>|t|)
(Intercept) -33.4966292925   26.3533937251  -1.271                 0.20451
DISTANCE      0.0753095733    0.0031047542  24.256 < 0.0000000000000002 ***
SWYes       -41.4733728683    4.8659644013  -8.523 0.000000000000000399 ***
VACATIONYes -34.3350683659    4.9201337810  -6.978 0.000000000013857896 ***
HI            0.0097087603    0.0011982666   8.102 0.000000000000007932 ***
GATEFree    -22.6016837311    4.9897995646  -4.530 0.000007984533428892 ***
SLOTFree    -12.9105872030    4.8685844144  -2.652                0.00835 **
E_INCOME      0.0022328686    0.0004662970   4.789 0.000002434450315483 ***
PAX          -0.0011197662    0.0001761919  -6.355 0.000000000611854341 ***
S_POP         0.0000041386    0.0000008456   4.894 0.000001476856966589 ***
E_POP         0.0000042189    0.0000009877   4.272 0.000024716165838752 ***
S_INCOME      0.0017514156    0.0006306365   2.777                0.00576 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.94 on 370 degrees of freedom
Multiple R-squared:  0.8109,    Adjusted R-squared:  0.8053
F-statistic: 144.2 on 11 and 370 DF,  p-value: < 0.00000000000000022
```

```
   airfaremodel.valid.data.FARE airfaremodel.lm.predfwd  residuals
4                         85.47               126.58696 -41.116956
6                         56.76                82.51034 -25.750341
7                        228.00               242.76278 -14.762782
8                        116.54               122.17161  -5.631613
9                        172.63               180.13237  -7.502367
10                       114.76               130.66911 -15.909114
>
> #Calculating accuracy parameters based on above prediction
> accuracy(airfaremodel.lm.predfwd, airfaremodel.valid.data$FARE)
                ME     RMSE      MAE       MPE     MAPE
Test set -3.604363 37.43276 29.13853 -5.886503 24.09033
```

Question 2 Part d:

- Based on the predictor variables obtained in the model, we can conclude that when Southwest airlines is operating on a route the airfare is less as compared to the route on which Southwest is not functional.
- Model 3 has MAPE of 24.09%, that means it has a reasonable prediction value. Other airline services can therefore use the predictor variables present in the model to optimally predict prices and keep them comparable to southwest airlines to equally attract flyers.