Homework2

Shraddha Hemant Kadam (sxk190069@utdallas.edu)

06/23/2020

Loading Packages

```
pacman::p_load(data.table, forecast, leaps, tidyverse, ggcorrplot, corrplot, MASS)
theme_set(theme_classic())
```

Using Airfares data set

```
air.df <- read.csv("Airfares.csv")

# Remove first four features
air.df <- air.df[-c(1:4)]
head(air.df)
```

```
##   COUPON NEW VACATION  SW      HI S_INCOME E_INCOME   S_POP   E_POP       SLOT
## 1   1.00   3       No Yes 5291.99    28637    21112 3036732  205711       Free
## 2   1.06   3       No  No 5419.16    26993    29838 3532657 7145897       Free
## 3   1.06   3       No  No 9185.28    30124    29838 5787293 7145897       Free
## 4   1.06   3       No Yes 2657.35    29260    29838 7830332 7145897 Controlled
## 5   1.06   3       No Yes 2657.35    29260    29838 7830332 7145897       Free
## 6   1.01   3       No Yes 3408.11    26046    29838 2230955 7145897       Free
##   GATE DISTANCE   PAX   FARE
## 1 Free      312  7864  64.11
## 2 Free      576  8820 174.47
## 3 Free      364  6452 207.76
## 4 Free      612 25144  85.47
## 5 Free      612 25144  85.47
## 6 Free      309 13386  56.76
```

Question 1 Correlation table and scatterplots:

```
#correlation table
numeric.air.df <- air.df[, -c(3,4,10,11)]
round(cor(numeric.air.df),3)
```

```
##          COUPON    NEW     HI S_INCOME E_INCOME  S_POP  E_POP DISTANCE    PAX
## COUPON    1.000  0.020 -0.347   -0.088    0.047 -0.108  0.095    0.747 -0.337
## NEW       0.020  1.000  0.054    0.027    0.113 -0.017  0.059    0.081  0.010
## HI       -0.347  0.054  1.000   -0.027    0.082 -0.172 -0.062   -0.312 -0.169
## S_INCOME -0.088  0.027 -0.027    1.000   -0.139  0.517 -0.272    0.028  0.138
## E_INCOME  0.047  0.113  0.082   -0.139    1.000 -0.144  0.458    0.177  0.260
```

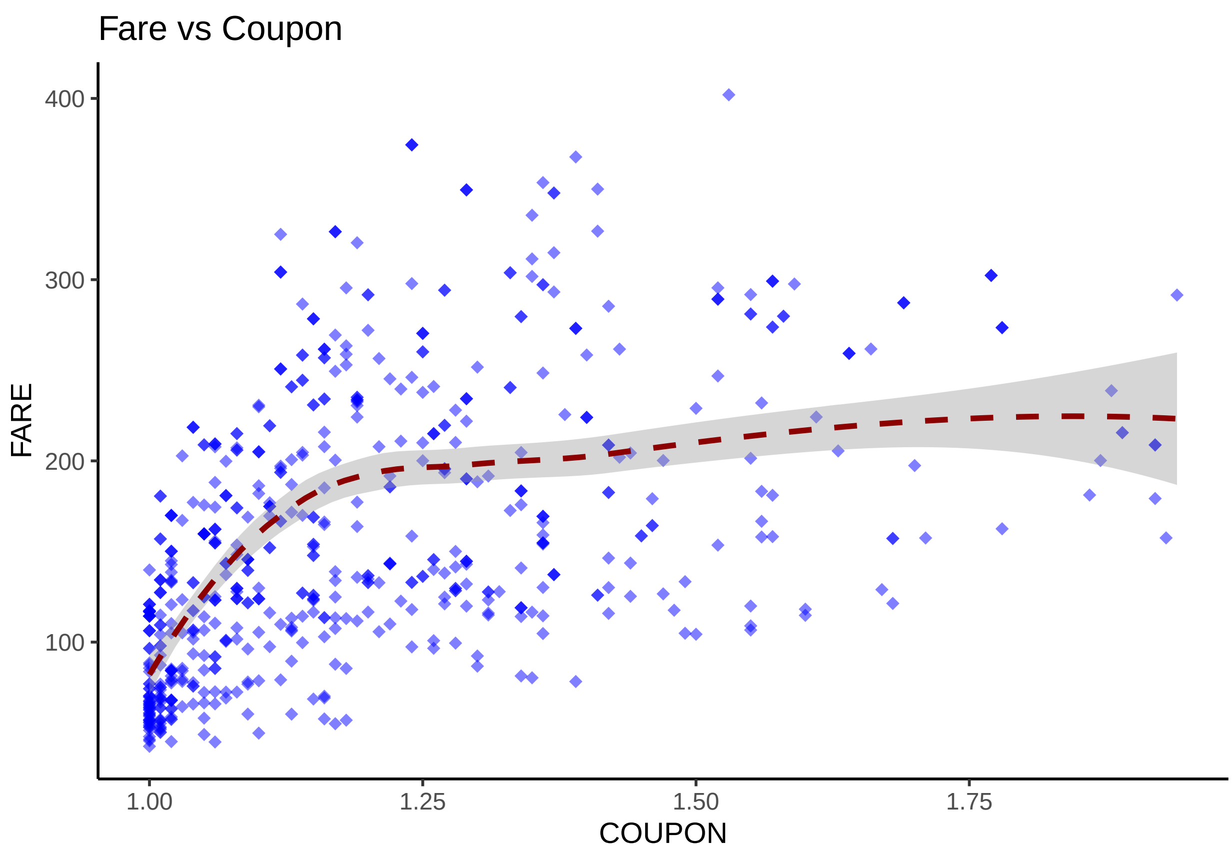
```
## S_POP    -0.108 -0.017 -0.172    0.517   -0.144  1.000 -0.280    0.018  0.285
## E_POP     0.095  0.059 -0.062   -0.272    0.458 -0.280  1.000    0.116  0.315
## DISTANCE  0.747  0.081 -0.312    0.028    0.177  0.018  0.116    1.000 -0.102
## PAX      -0.337  0.010 -0.169    0.138    0.260  0.285  0.315   -0.102  1.000
## FARE      0.497  0.092  0.025    0.209    0.326  0.145  0.285    0.670 -0.091
##            FARE
## COUPON    0.497
## NEW       0.092
## HI        0.025
## S_INCOME  0.209
## E_INCOME  0.326
## S_POP     0.145
## E_POP     0.285
## DISTANCE  0.670
## PAX      -0.091
## FARE      1.000
```

```r
head(numeric.air.df)
```

```
##   COUPON NEW      HI S_INCOME E_INCOME   S_POP   E_POP DISTANCE   PAX   FARE
## 1   1.00   3 5291.99    28637    21112 3036732  205711      312  7864  64.11
## 2   1.06   3 5419.16    26993    29838 3532657 7145897      576  8820 174.47
## 3   1.06   3 9185.28    30124    29838 5787293 7145897      364  6452 207.76
## 4   1.06   3 2657.35    29260    29838 7830332 7145897      612 25144  85.47
## 5   1.06   3 2657.35    29260    29838 7830332 7145897      612 25144  85.47
## 6   1.01   3 3408.11    26046    29838 2230955 7145897      309 13386  56.76
```

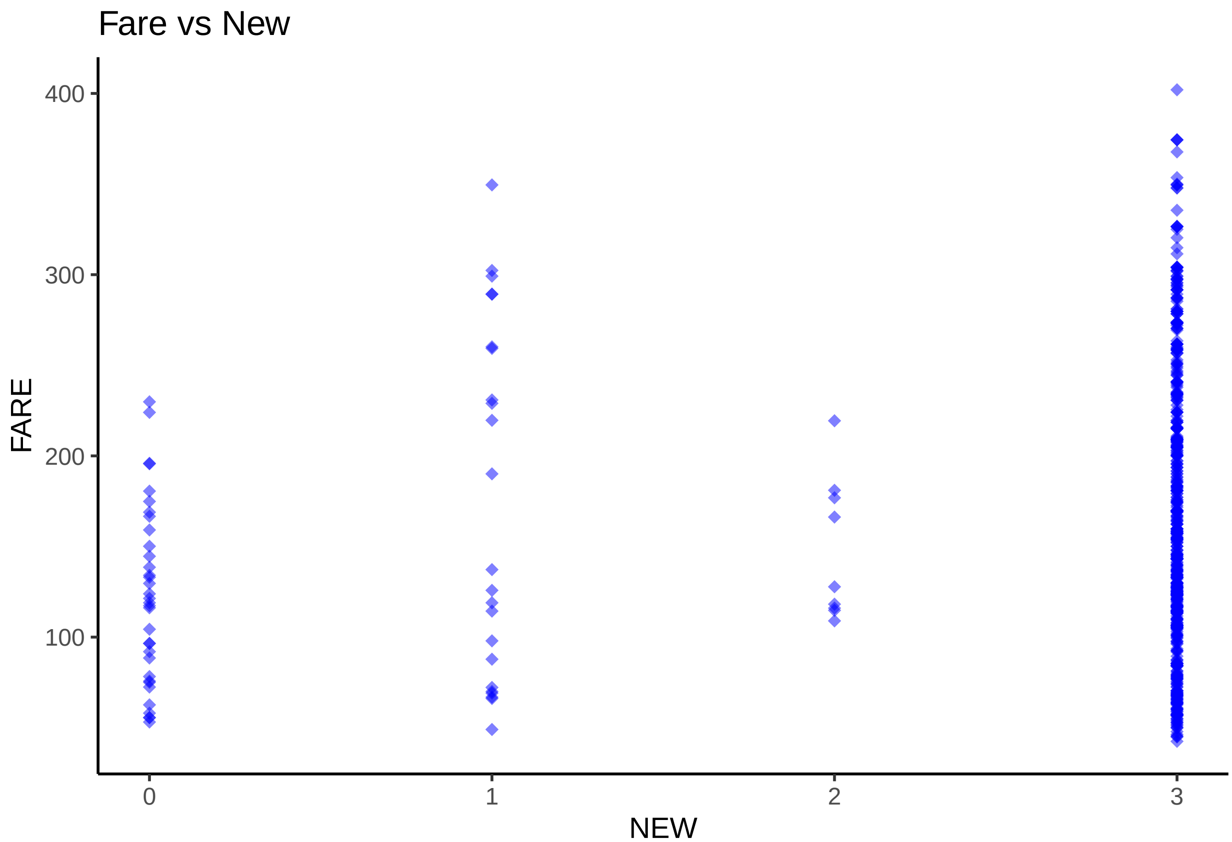
```r
#scatter plot
ggplot(air.df, aes(x = COUPON, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs Coupon")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



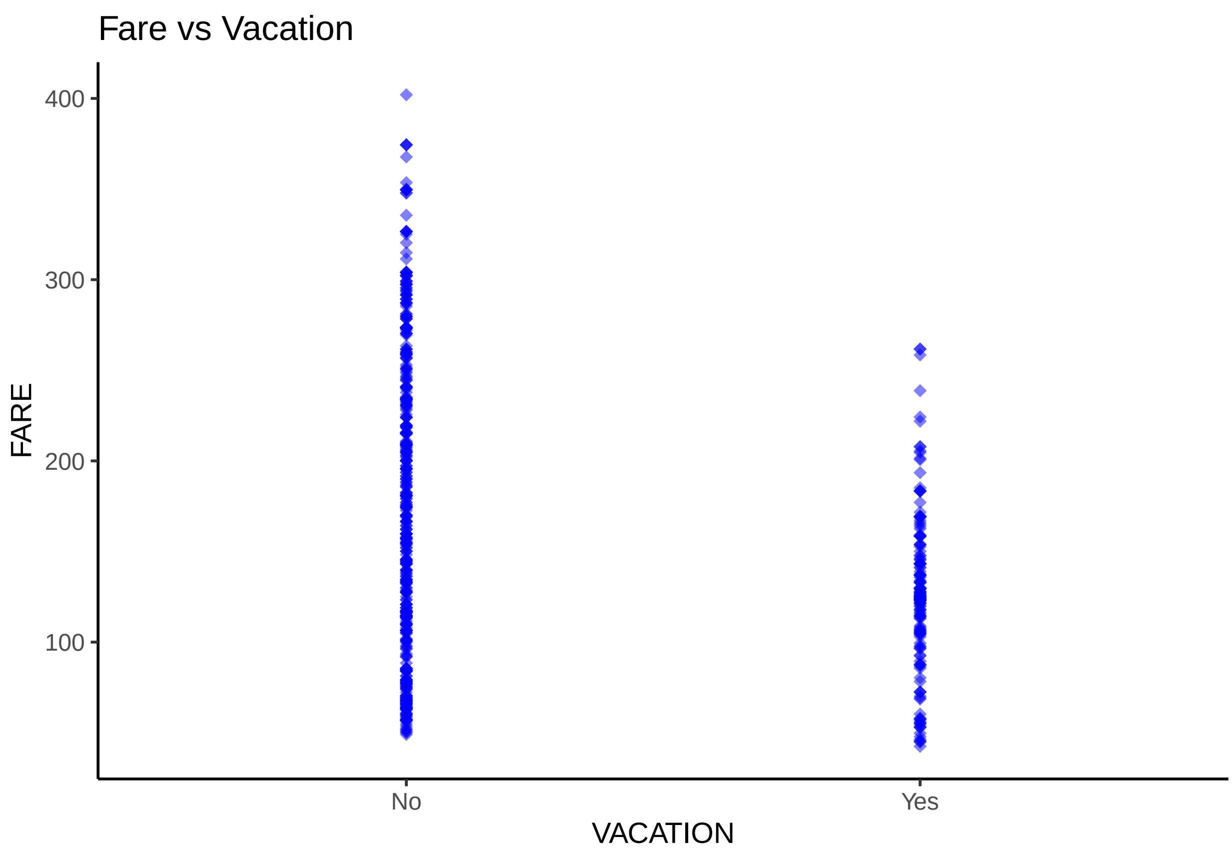

```r
ggplot(air.df, aes(x = NEW, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs New")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



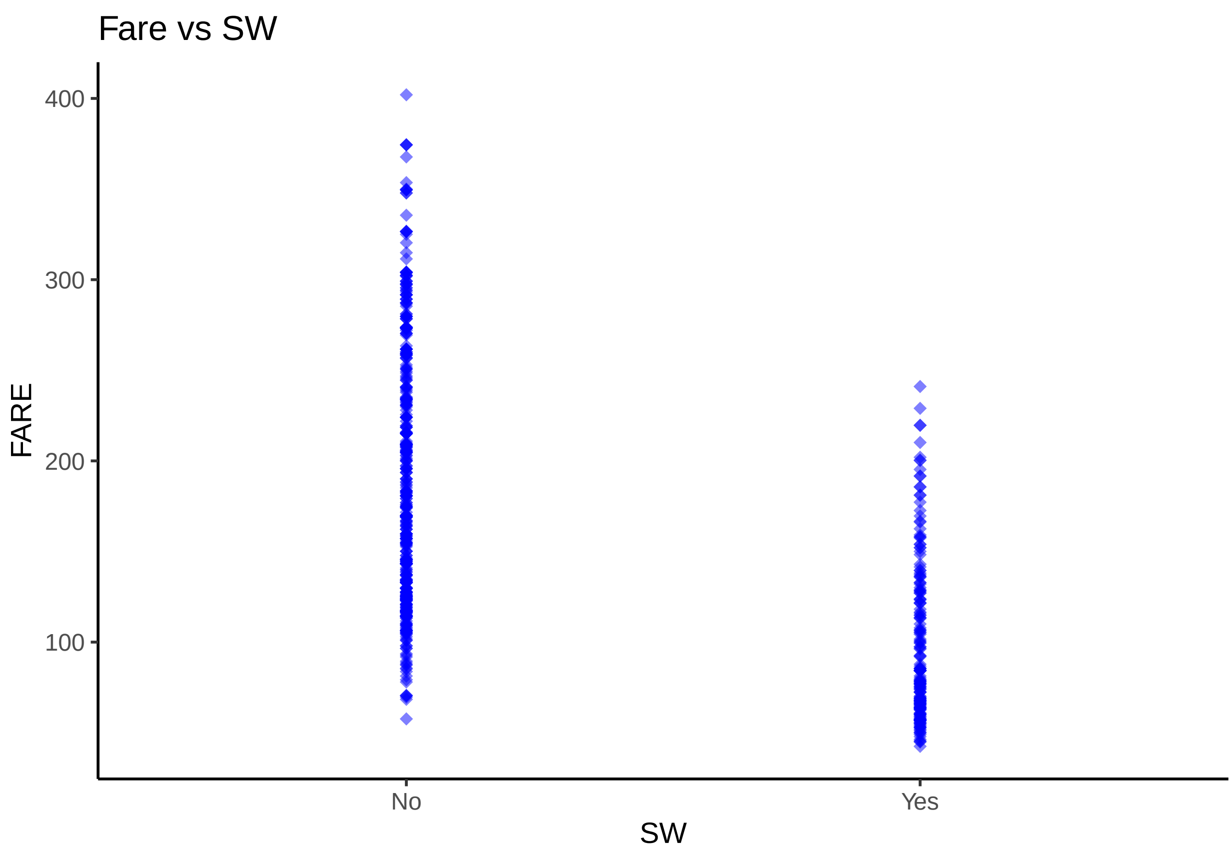

```
ggplot(air.df, aes(x = VACATION, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs Vacation")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



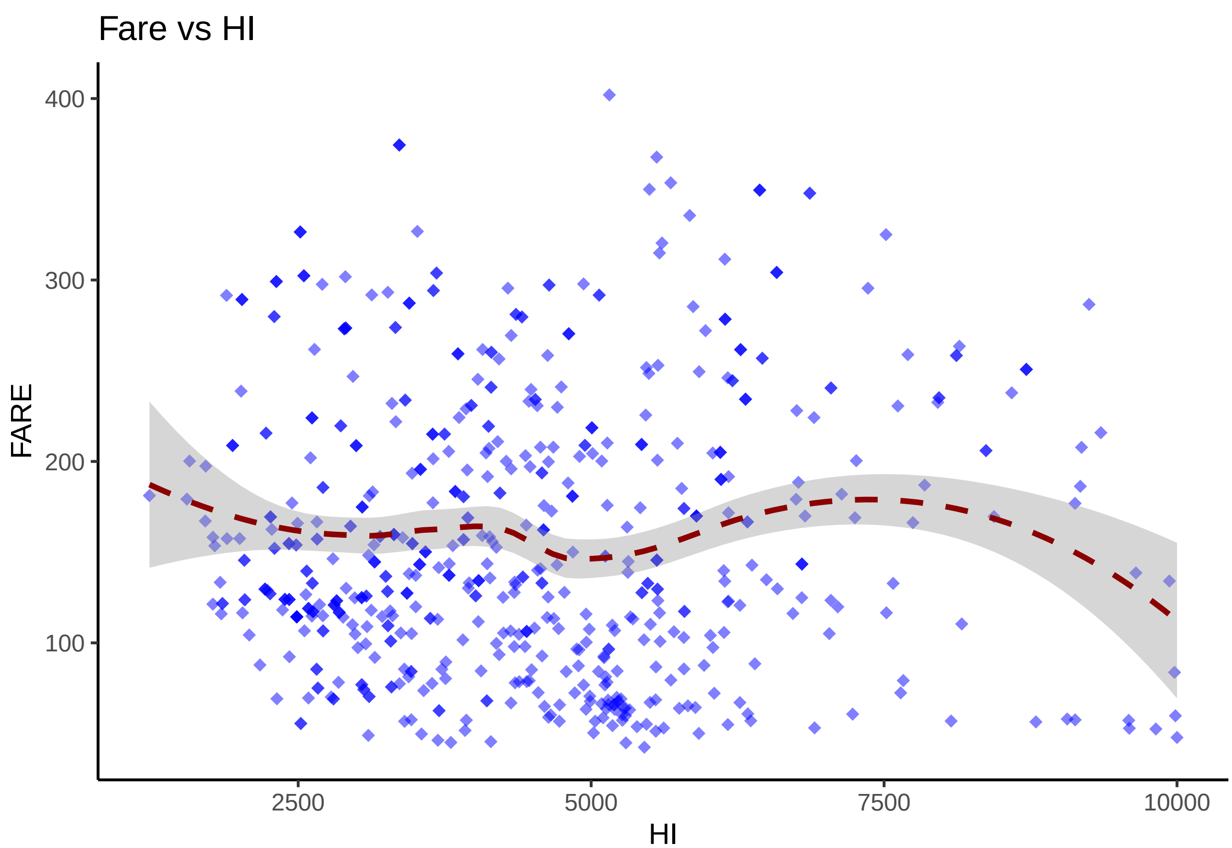

```r
ggplot(air.df, aes(x = SW, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs SW")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



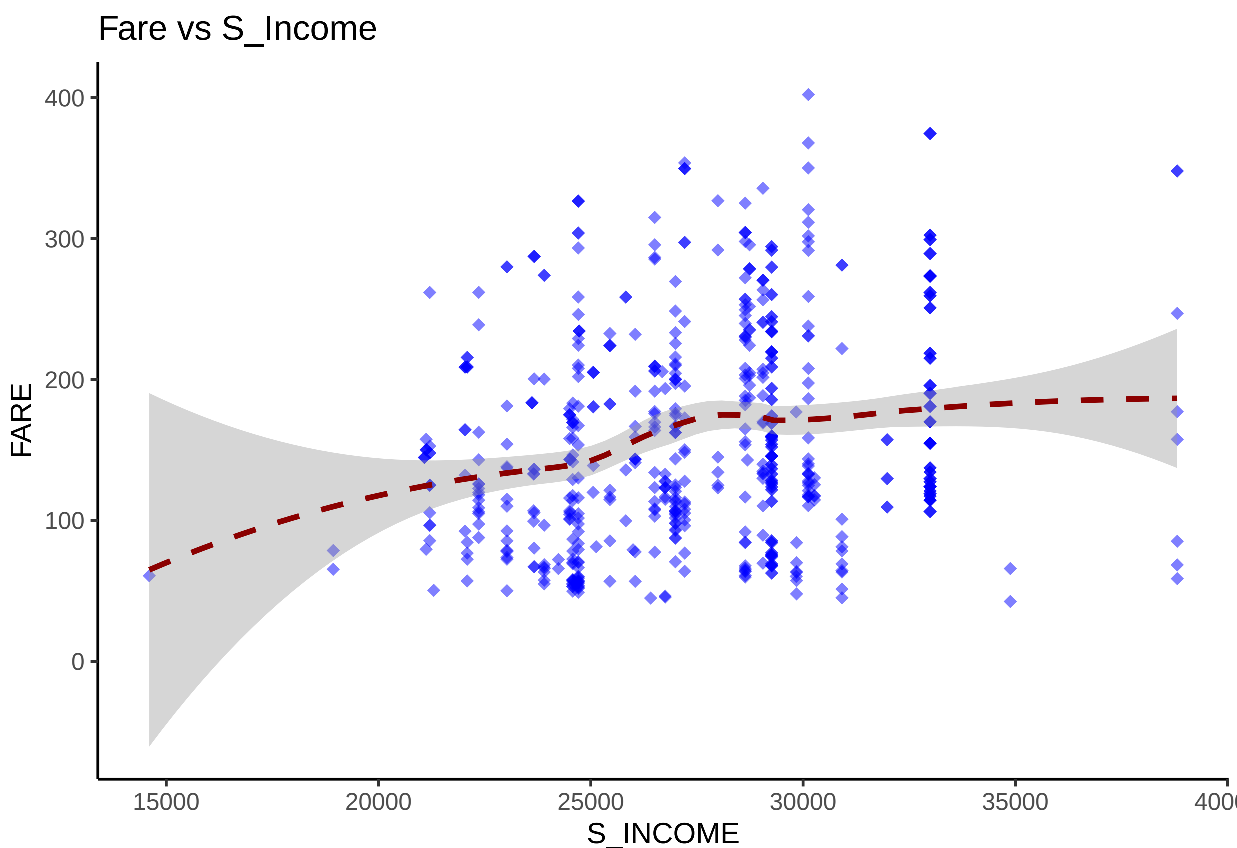

```r
ggplot(air.df, aes(x = HI, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs HI")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



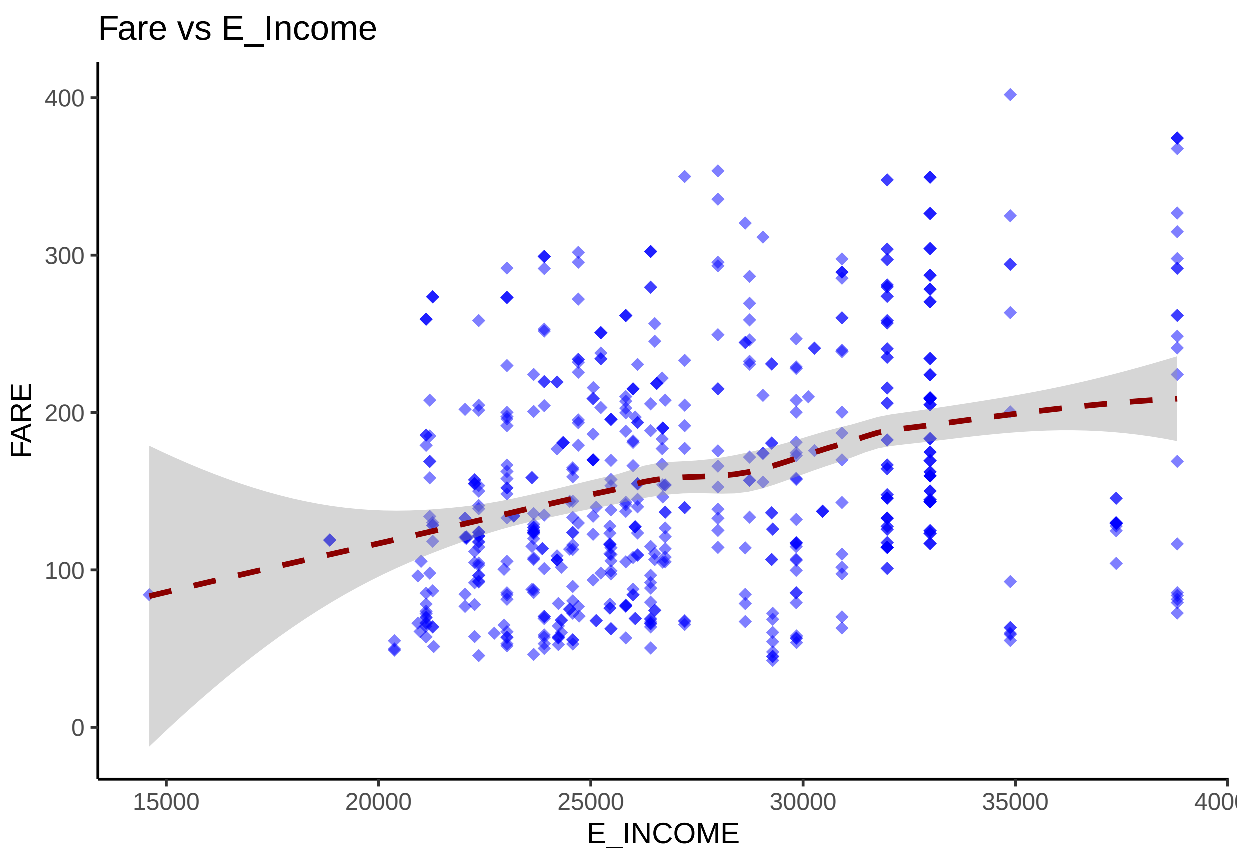

```r
ggplot(air.df, aes(x = S_INCOME, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs S_Income")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



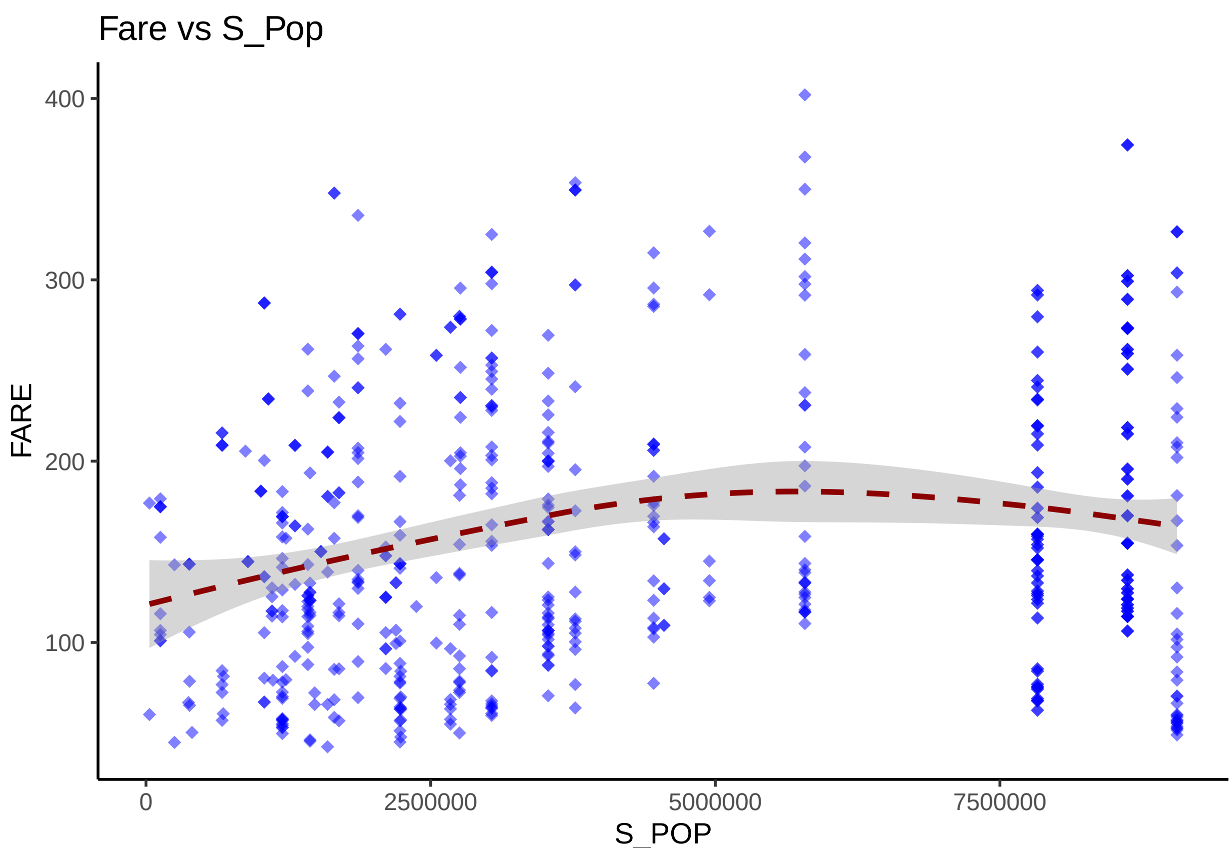

```r
ggplot(air.df, aes(x = E_INCOME, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs E_Income")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



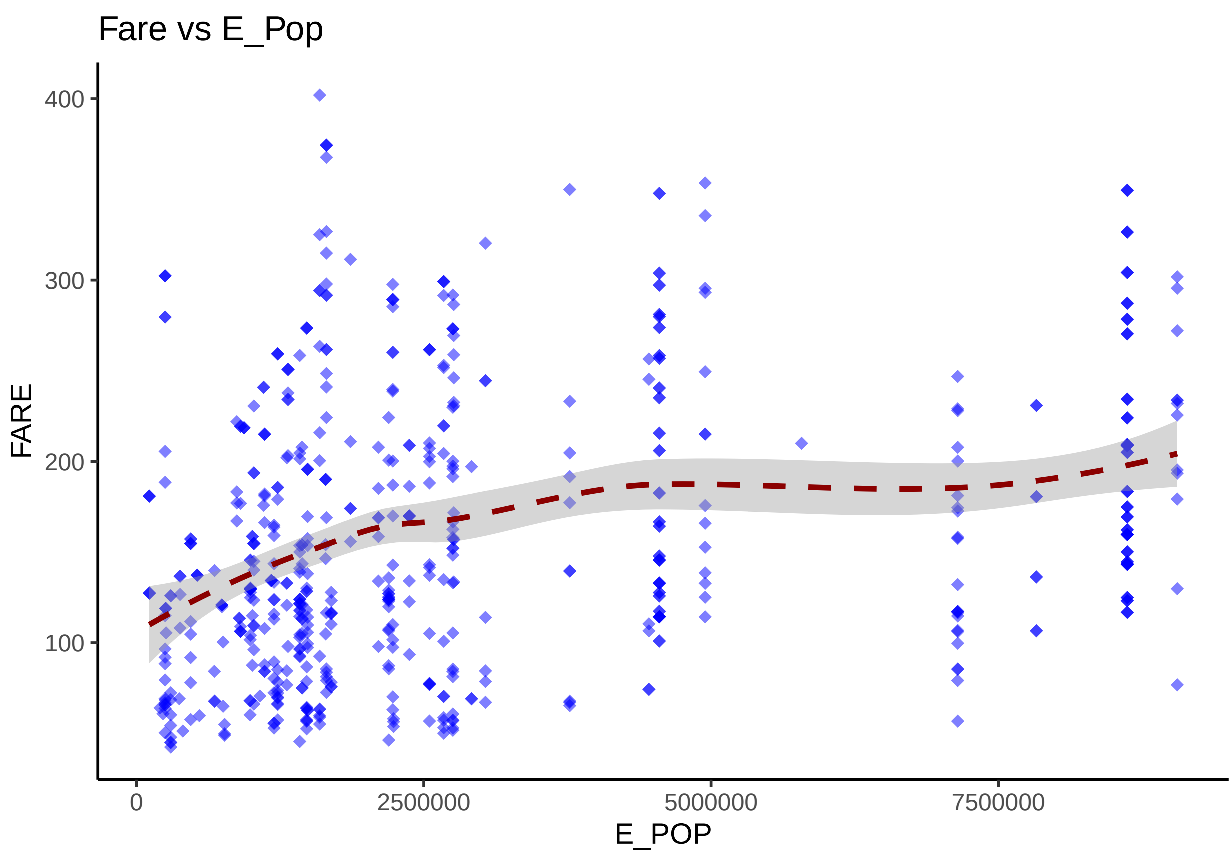

```
ggplot(air.df, aes(x = S_POP, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs S_Pop")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



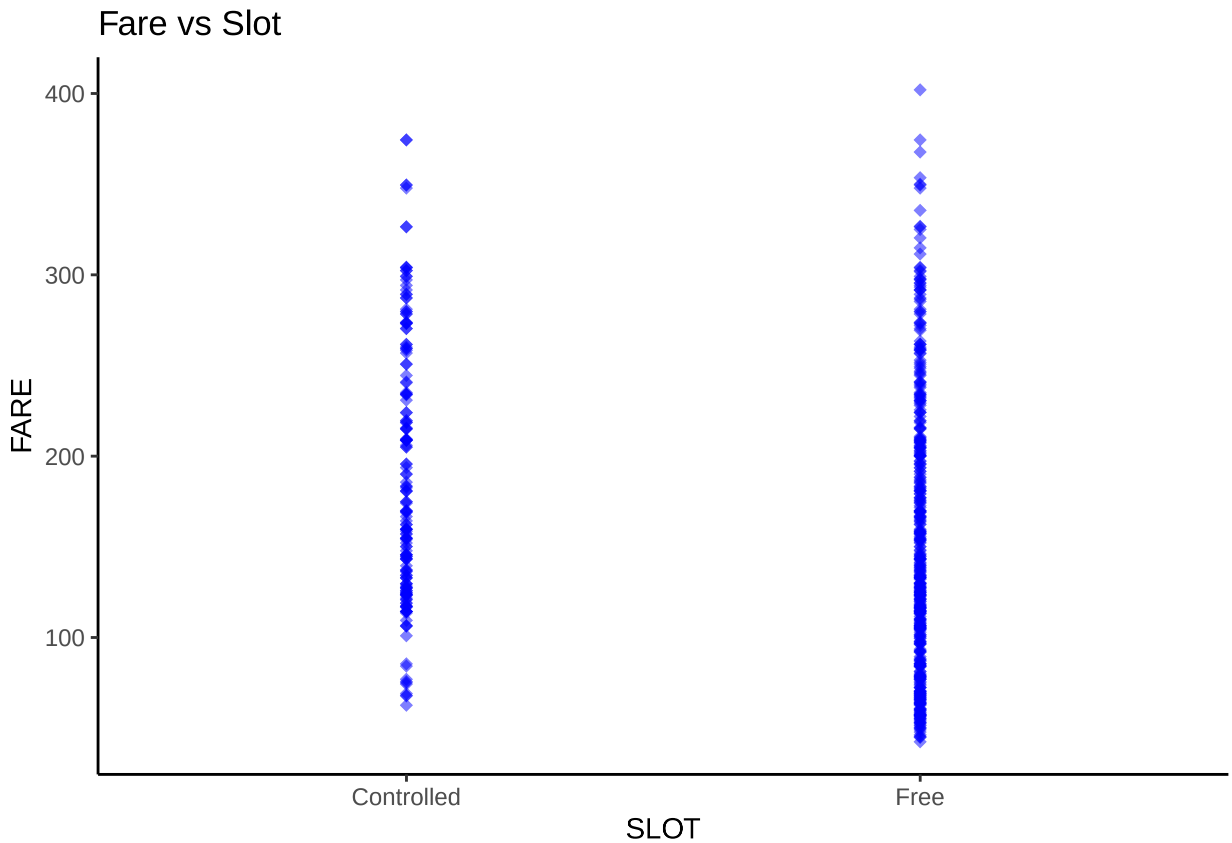

```r
ggplot(air.df, aes(x = E_POP, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs E_Pop")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



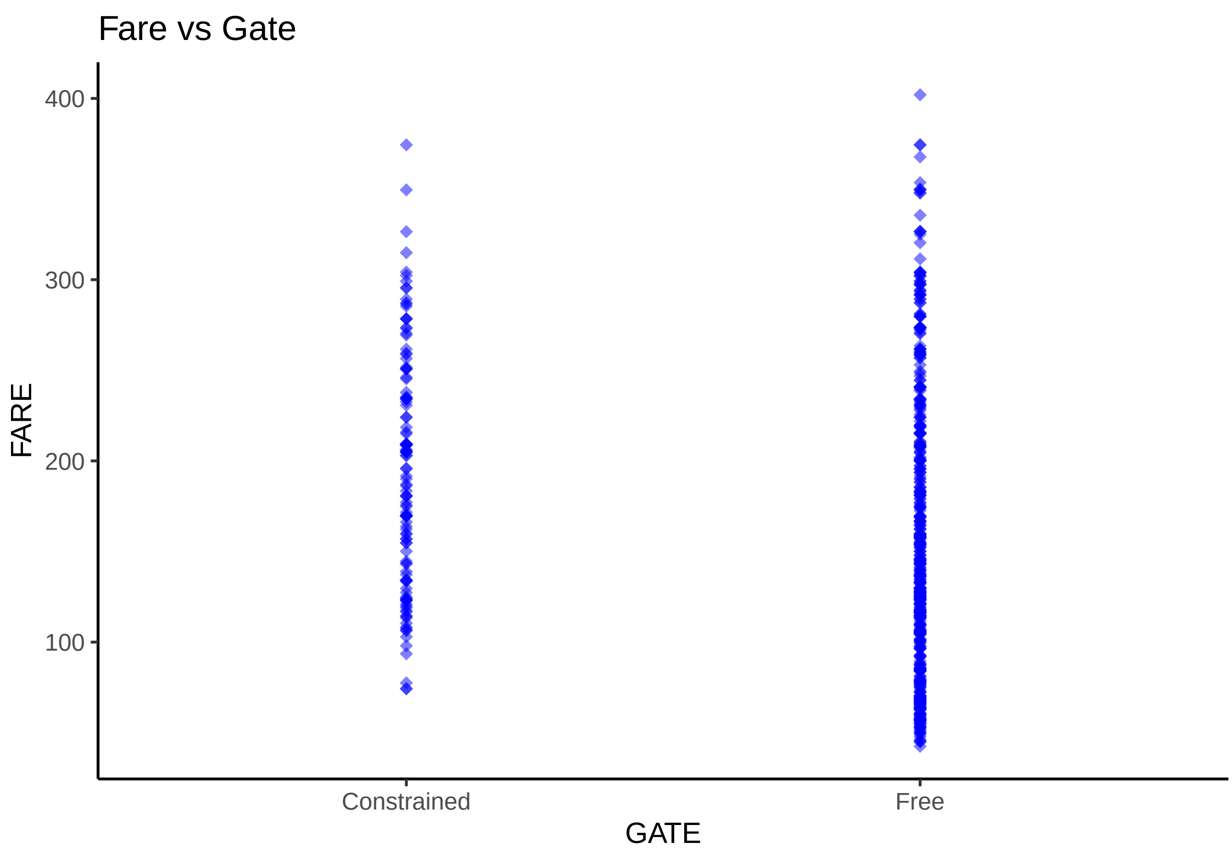

```r
ggplot(air.df, aes(x = SLOT, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs Slot")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



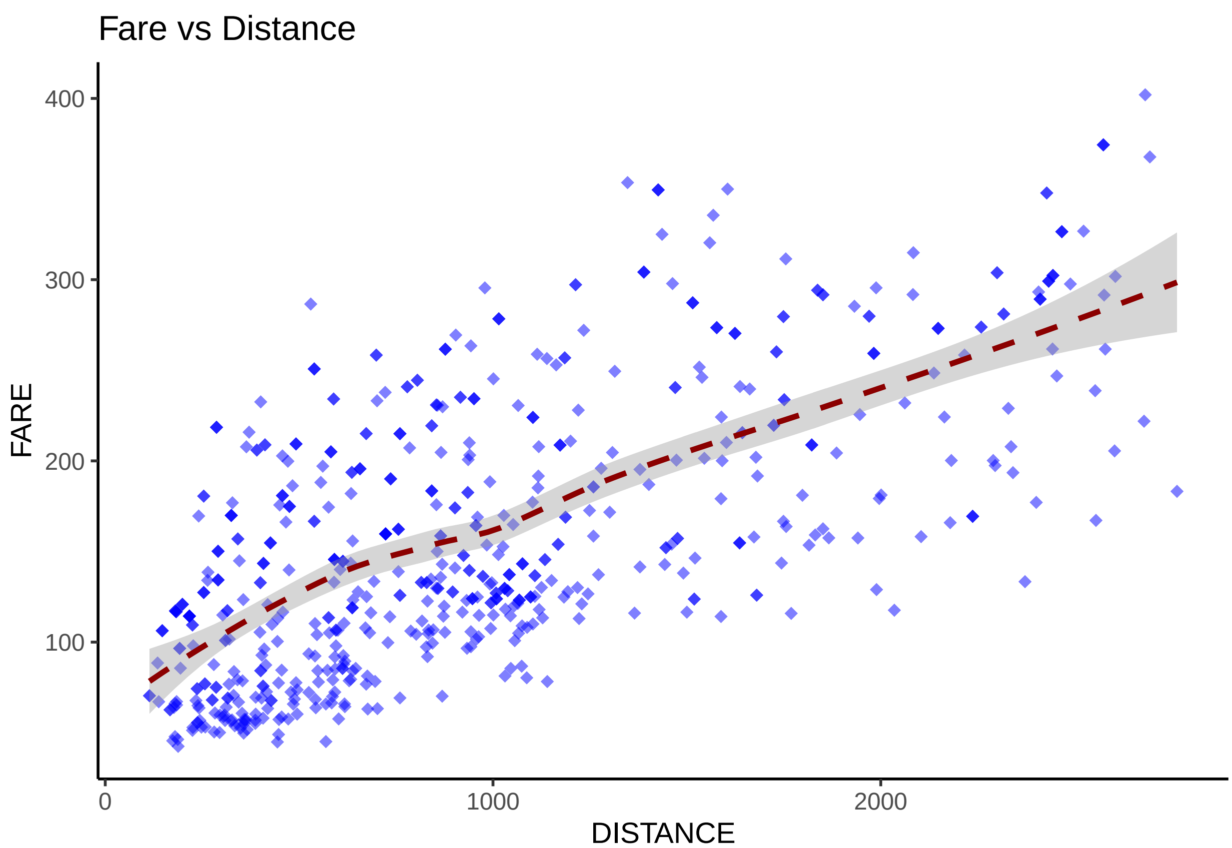

```
ggplot(air.df, aes(x = GATE, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs Gate")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



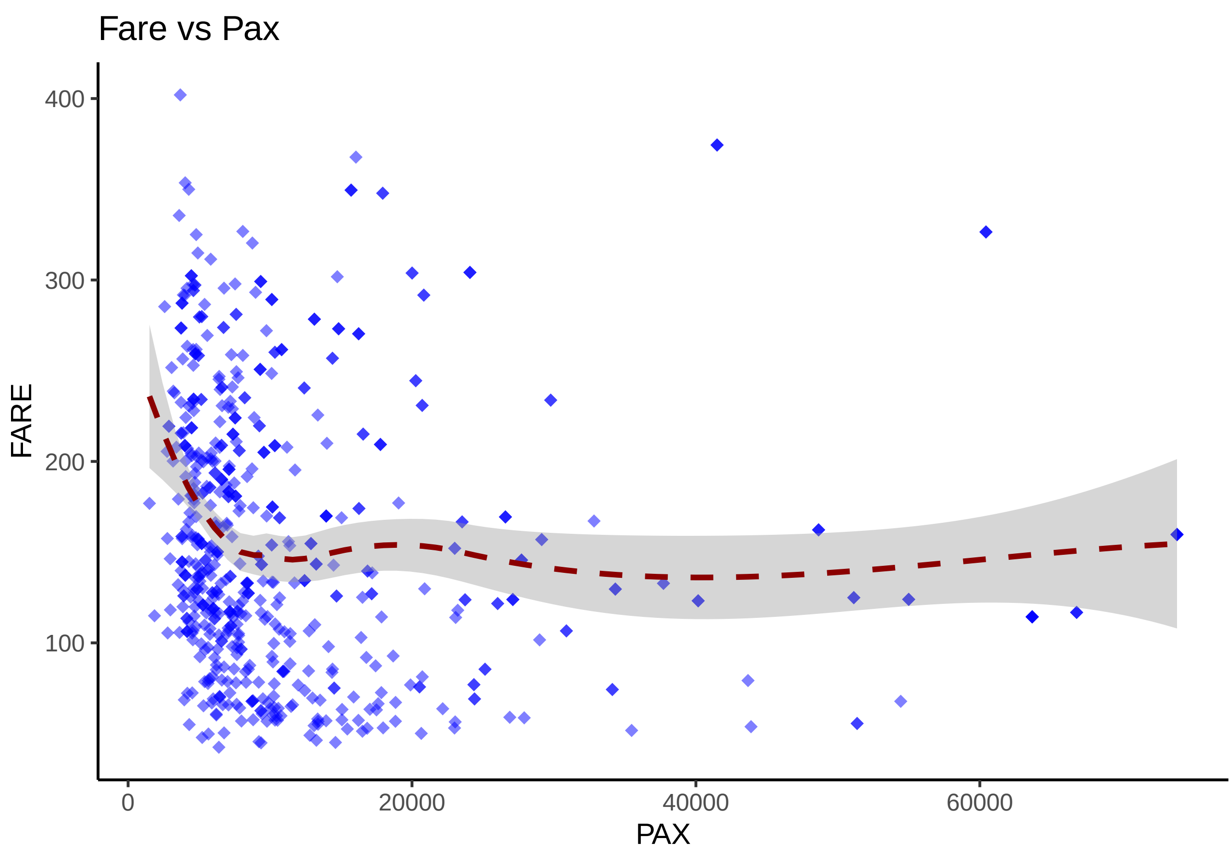

```
ggplot(air.df, aes(x = DISTANCE, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs Distance")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```r
ggplot(air.df, aes(x = PAX, y = FARE)) +
  geom_point(size= 2, shape= 18, color = "blue", alpha = 0.5) +
  ggtitle("Fare vs Pax")+
  geom_smooth(linetype="dashed",
              color="darkred")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Distance seems to be the best single predictor of FARE because it shows the most positive correlation of 0.670. That means 67% of the variation in FARE can be explained by change in Distance predictor.

Question2 Pivot tables for categorical predictors:

```
VACATION_f <- prop.table(table(air.df$VACATION))
SW_f <- prop.table(table(air.df$SW))
SLOT_f <- prop.table(table(air.df$SLOT))
GATE_f <- prop.table(table(air.df$GATE))

VACATION_AVG_Fare <- air.df %>% group_by(VACATION) %>% summarise(mean(FARE))
SW_AVG_Fare <- air.df %>% group_by(SW) %>% summarise(mean(FARE))
SLOT_AVG_Fare <- air.df %>% group_by(SLOT) %>% summarise(mean(FARE))
GATE_AVG_Fare <- air.df %>% group_by(GATE) %>% summarise(mean(FARE))

VACATION_PIVOT <- cbind(VACATION_f, VACATION_AVG_Fare)
VACATION_PIVOT['Freq'] <- VACATION_PIVOT['Freq']*100
VACATION_PIVOT$Var1<- NULL
VACATION_PIVOT <- VACATION_PIVOT[c("VACATION","Freq","mean(FARE)")]
names(VACATION_PIVOT)[2] <- "Percentage"
names(VACATION_PIVOT)[3] <- "AVERAGE FARE"
VACATION_PIVOT
```

```
##   VACATION Percentage AVERAGE FARE
## 1       No   73.35423     173.5525
## 2      Yes   26.64577     125.9809
```

```r
SW_PIVOT <- cbind(SW_f, SW_AVG_Fare)
SW_PIVOT['Freq'] <- SW_PIVOT['Freq']*100
SW_PIVOT$Var1<- NULL
SW_PIVOT <- SW_PIVOT[c("SW","Freq","mean(FARE)")]
names(SW_PIVOT)[2] <- "Percentage"
names(SW_PIVOT)[3] <- "AVERAGE FARE"
SW_PIVOT
```

```
##    SW Percentage AVERAGE FARE
## 1  No   69.59248    188.18279
## 2 Yes   30.40752     98.38227
```

```r
SLOT_PIVOT <- cbind(SLOT_f, SLOT_AVG_Fare)
SLOT_PIVOT['Freq'] <- SLOT_PIVOT['Freq']*100
SLOT_PIVOT$Var1<- NULL
SLOT_PIVOT <- SLOT_PIVOT[c("SLOT","Freq","mean(FARE)")]
names(SLOT_PIVOT)[2] <- "Percentage"
names(SLOT_PIVOT)[3] <- "AVERAGE FARE"
SLOT_PIVOT
```

```
##         SLOT Percentage AVERAGE FARE
## 1 Controlled   28.52665     186.0594
## 2       Free   71.47335     150.8257
```

```r
GATE_PIVOT <- cbind(GATE_f, GATE_AVG_Fare)
GATE_PIVOT['Freq'] <- GATE_PIVOT['Freq']*100
GATE_PIVOT$Var1<- NULL
GATE_PIVOT <- GATE_PIVOT[c("GATE","Freq","mean(FARE)")]
names(GATE_PIVOT)[2] <- "Percentage"
names(GATE_PIVOT)[3] <- "AVERAGE FARE"
GATE_PIVOT
```

```
##          GATE Percentage AVERAGE FARE
## 1 Constrained   19.43574      193.129
## 2        Free   80.56426      153.096
```

From the above pivot tables of categorical variables, we can see that the difference in the FARE with and without SW on the routes is highest with the percentage of 69.59% and 30.40% respectively and as compared to other categorical variables SW can impact the fare prices significantly. Hence, SW is the best predictor of FARE.

Question3 Data Partition:

```r
set.seed(42)
rows <- sample(nrow(air.df))
air.df <- air.df[rows, ]

#rows to split on
split <- round(nrow(air.df) * (0.8))
train.df <- air.df[1:split, ]
test.df <- air.df[(split+1):nrow(air.df), ]
test.df
```

```
##      COUPON NEW VACATION  SW      HI S_INCOME E_INCOME   S_POP   E_POP
## 77     1.35   3        No  No 6140.91    30124    29055 5787293 1862106
## 7      1.28   3        No  No 6754.48    28637    29838 3036732 7145897
## 264    1.29   3        No  No 6438.27    27211    32991 3770125 8621121
## 454    1.78   3        No  No 2905.97    32991    21276 8621121 1481709
## 631    1.26   3       Yes  No 2040.42    29260    37375 7830332  991717
## 490    1.24   3        No  No 3362.86    32991    38813 8621121 1653017
## 101    1.34   3        No  No 5137.41    26993    30268 3532657 1106780
## 460    1.27   3        No Yes 2863.20    29260    23903 7830332 2673620
## 596    1.36   3        No  No 4641.26    27211    31981 3770125 4549784
## 375    1.34   3        No  No 4409.42    29260    26409 7830332  249561
## 523    1.52   3        No  No 2019.98    32991    30916 8621121 2230831
## 576    1.12   3        No  No 6334.03    26993    31981 3532657 4549784
## 90     1.15   3       Yes Yes 2482.76    29260    26752 7830332 1440377
## 295    1.00   3        No Yes 5391.08    24706    29846 9056076 2237227
## 399    1.03   3        No  No 7046.11    26993    26101 3532657 1021830
## 324    1.06   3        No  No 4248.47    26993    27994 3532657 4948339
## 286    1.15   3        No  No 6143.20    28739    32991 2761118 8621121
## 577    1.12   3        No  No 6334.03    26993    31981 3532657 4549784
## 50     1.01   0        No  No 3910.81    25059    29260 1595139 7830332
## 39     1.14   3        No  No 4439.86    28637    25237 3036732 1318892
## 34     1.34   1        No  No 2587.89    32991    18851 8621121  254153
## 585    1.07   3        No  No 3289.86    24502    31981  125722 4549784
## 320    1.30   3       Yes Yes 2424.61    22038    22360 1308499 1421287
## 176    1.02   3        No Yes 4109.87    29260    24307 7830332  989164
## 140    1.02   3        No  No 5505.79    29055    25450 1862106 1694803
## 437    1.05   3        No Yes 5089.75    24706    21121 9056076 1228816
## 367    1.02   3        No  No 5898.74    32991    25054 8621121 2374260
## 271    1.40   3        No  No 2617.87    25450    32991 1694803 8621121
## 551    1.01   3       Yes  No 4891.84    26993    23654 3532657 2195215
## 621    1.37   3        No  No 6865.77    38813    31981 1653017 4549784
## 422    1.04   3        No  No 5006.45    32991    26553 8621121  936107
## 448    1.28   3        No Yes 3262.15    29260    21276 7830332 1481709
## 279    1.29   3        No  No 6317.55    24725    32991 1074558 8621121
## 332    1.36   3        No  No 5679.25    27211    27994 3770125 4948339
## 380    1.77   3        No  No 2548.46    32991    26409 8621121  249561
## 416    1.06   3       Yes Yes 5296.51    26409    29284  249561  298680
## 242    1.11   0        No  No 3046.45    24502    32991  125722 8621121
## 69     1.02   3        No Yes 5222.30    22089    28637  668159 3036732
## 281    1.05   3       Yes  No 3042.09    21207    32991 2105604 8621121
## 143    1.39   0       Yes Yes 2844.24    24575    25450 1197234 1694803
## 408    1.01   3        No  No 3266.44    31981    26101 4549784 1021830
## 607    1.46   3        No  No 2946.23    22038    31981 1308499 4549784
## 354    1.01   3        No Yes 3923.94    24706    23025 9056076 2753373
## 625    1.14   3        No  No 8117.12    25824    31981 2549844 4549784
## 65     1.13   3        No  No 5356.51    26993    24502 3532657 1442203
## 575    1.89   3        No  No 2225.74    22089    31981  668159 4549784
## 308    1.08   3       Yes  No 3819.52    28637    22360 3036732 1421287
## 230    1.05   3        No  No 3316.90    29260    32991 7830332 8621121
## 534    1.08   3        No  No 4128.60    29055    25824 1862106 2549844
## 222    1.06   3        No  No 4593.38    26993    32991 3532657 8621121
## 64     1.68   3        No  No 2661.53    31981    22263 4549784  472254
## 21     1.87   3        No Yes 1572.93    23903    29838 2673620 7145897
## 339    1.25   3        No  No 4275.35    26993    23025 3532657 2753373
```

```
## 592   1.31   3       Yes  No 5433.17    26752    31981 1440377 4549784
## 539   1.28   3        No Yes 5138.01    24706    25824 9056076 2549844
## 183   1.16   3       Yes  No 5772.86    28637    21207 3036732 2105604
## 29    1.01   3        No  No 4040.09    32991    23184 8621121 1173217
## 571   1.17   3        No Yes 6167.00    23903    20375 2673620  766956
## 537   1.00   3        No Yes 5034.18    25450    25824 1694803 2549844
## 4     1.06   3        No Yes 2657.35    29260    29838 7830332 7145897
## 232   1.05   3        No  No 3316.90    29260    32991 7830332 8621121
## 459   1.27   1        No Yes 2863.20    29260    23903 7830332 2673620
## 223   1.06   3        No  No 4593.38    26993    32991 3532657 8621121
## 381   1.77   3        No  No 2548.46    32991    26409 8621121  249561
## 9     1.33   3        No Yes 4662.44    27211    29838 3770125 7145897
## 200   1.13   3       Yes  No 6172.12    24575    28739 1197234 2761118
## 92    1.16   3       Yes  No 4677.03    28637    26752 3036732 1440377
## 137   1.04   3        No Yes 3296.05    29260    25450 7830332 1694803
## 584   1.07   3        No  No 3289.86    24502    31981  125722 4549784
## 434   1.22   3        No Yes 2711.42    29260    21121 7830332 1228816
## 291   1.02   3        No  No 3585.86    21125    32991 1536012 8621121
## 44    1.04   3        No  No 2712.37    26993    29260 3532657 7830332
## 160   1.00   3        No Yes 5293.05    28637    22726 3036732  547633
## 71    1.00   3        No Yes 5502.33    23665    28637 1038660 3036732
## 204   1.10   3        No  No 7138.34    28637    25995 3036732 1115048
## 511   1.47   3        No  No 5090.58    26993    30916 3532657 2230831
## 456   1.60   2       Yes Yes 2366.36    22360    21276 1421287 1481709
## 227   1.00   3        No  No 2850.33    30124    32991 5787293 8621121
## 478   1.20   3        No  No 5068.53    29260    38813 7830332 1653017
## 562   1.00   3       Yes  No 5791.78    21207    23654 2105604 2195215
## 175   1.02   3        No Yes 4109.87    29260    24307 7830332  989164
## 485   1.02   3        No  No 7664.03    24706    38813 9056076 1653017
## 36    1.25   3        No  No 8589.17    30124    25237 5787293 1318892
## 563   1.10   3       Yes  No 2422.98    32991    23654 8621121 2195215
## 256   1.06   3       Yes  No 2828.16    26752    32991 1440377 8621121
## 72    1.19   3        No  No 5605.06    30124    28637 5787293 3036732
## 452   1.42   3        No Yes 2909.15    24706    21276 9056076 1481709
## 469   1.57   1        No  No 2313.60    32991    23903 8621121 2673620
## 177   1.00   3        No Yes 6337.20    28637    20980 3036732  231325
## 586   1.16   3        No  No 6460.84    28637    31981 3036732 4549784
## 13    1.12   3        No Yes 4471.62    25995    29838 1115048 7145897
## 458   1.94   3        No  No 1888.30    30124    23903 5787293 2673620
## 266   1.34   3       Yes  No 3840.28    23614    32991 1008768 8621121
## 48    1.15   3        No  No 3977.23    30124    29260 5787293 7830332
## 450   1.00   3        No Yes 5751.82    27211    21276 3770125 1481709
## 373   1.01   1        No Yes 4315.92    23901    26409  372606  249561
## 118   1.12   3        No  No 5180.13    26993    25475 3532657 1489247
## 293   1.00   3        No Yes 8795.73    24706    29846 9056076 2237227
## 318   1.00   0       Yes  No 5149.70    21207    22360 2105604 1421287
## 598   1.42   3        No  No 4221.56    25450    31981 1694803 4549784
## 630   1.26   3       Yes  No 2040.42    29260    37375 7830332  991717
## 290   1.02   0        No  No 3585.86    21125    32991 1536012 8621121
## 334   1.37   3        No  No 3264.94    24706    27994 9056076 4948339
## 529   1.07   3        No  No 4636.00    26993    25824 3532657 2549844
## 302   1.24   3       Yes  No 3123.35    30124    22360 5787293 1421287
## 536   1.03   3        No Yes 3467.70    27211    25824 3770125 2549844
## 637   1.28   3       Yes  No 5566.43    31981    37375 4549784  991717
```

```
## 5     1.06   3       No Yes 2657.35    29260    29838 7830332 7145897
## 519   1.16   3      Yes Yes 2781.55    24575    30916 1197234 2230831
## 415   1.07   3      Yes Yes 4860.36    23025    29284 2753373  298680
## 464   1.00   3       No  No 3105.31    24706    23903 9056076 2673620
## 533   1.06   3       No  No 4803.13    28637    25824 3036732 2549844
## 502   1.01   3      Yes Yes 5472.43    24575    34880 1197234 1594251
## 341   1.70   3       No  No 1710.90    30124    23025 5787293 2753373
## 600   1.33   3       No  No 3680.60    24706    31981 9056076 4549784
## 570   1.05   1       No Yes 3098.74    24706    20375 9056076  766956
## 379   1.06   0       No Yes 3153.68    24706    26409 9056076  249561
## 221   1.06   3       No  No 4593.38    26993    32991 3532657 8621121
## 6     1.01   3       No Yes 3408.11    26046    29838 2230955 7145897
## 549   1.55   3      Yes  No 3503.11    22360    22069 1421287  743633
## 66    1.22   3       No  No 3789.64    30124    24502 5787293 1442203
## 476   1.00   3       No  No 9978.49    24706    38813 9056076 1653017
## 209   1.26   3       No  No 3647.27    32991    25995 8621121 1115048
## 342   1.00   3       No Yes 5266.72    24706    23025 9056076 2753373
## 363   1.04   3       No  No 4215.01    26993    25054 3532657 2374260
## 70    1.05   3       No  No 4624.90    26993    28637 3532657 3036732
## 154   1.16   3      Yes  No 4446.51    28637    24575 3036732 1197234
## 155   1.13   3      Yes  No 3760.10    29055    24575 1862106 1197234
##           SLOT        GATE DISTANCE   PAX   FARE
## 77        Free        Free     1755  5820 311.46
## 7         Free        Free     1220  4625 228.00
## 264 Controlled        Free     1426 15711 349.53
## 454 Controlled        Free     1577  3732 273.53
## 631       Free        Free     1134  5449 145.53
## 490       Free Constrained     2574 41492 374.40
## 101       Free        Free      854  5806 175.81
## 460       Free        Free     1724  9252 219.63
## 596       Free        Free     1213  4708 297.20
## 375       Free        Free     1749  5025 279.61
## 523 Controlled        Free     2411 10125 289.25
## 576       Free        Free      539 23531 166.67
## 90  Controlled        Free     1168 10117 153.95
## 295       Free        Free      334 43884  53.80
## 399       Free        Free      356  9307 123.44
## 324       Free        Free      674 16512 125.09
## 286       Free Constrained     1015 13123 278.39
## 577 Controlled        Free      539 23531 166.67
## 50        Free Constrained      254  7069 180.56
## 39        Free Constrained      940  4493 203.17
## 34        Free Constrained      637  6003 118.95
## 585 Controlled        Free      310  6583 100.95
## 320       Free        Free      541  5057  92.35
## 176       Free        Free      276  8793  68.06
## 140       Free        Free      541  7679 110.25
## 437       Free        Free      584 17617  66.46
## 367 Controlled Constrained      325 13957 169.90
## 271       Free Constrained     1103  7543 223.99
## 551       Free        Free      414 17437  87.35
## 621 Controlled        Free     2428 17938 347.82
## 422       Free Constrained      287  4472 218.54
## 448       Free        Free     1038  6233 128.36
```

```
## 279 Controlled        Free      951  4614 234.31
## 332       Free        Free     1347  4023 353.56
## 380 Controlled        Free     2444  4455 302.33
## 416       Free        Free      444  9368  44.89
## 242 Controlled        Free      475 10168 174.87
## 69        Free        Free      573 10941  84.46
## 281 Controlled        Free     1097 51122 124.92
## 143       Free        Free     1140  8309  78.24
## 408       Free        Free      225  7241 109.44
## 607 Controlled        Free      956  6208 164.30
## 354       Free        Free      366 35471  51.73
## 625 Controlled        Free      699  4957 258.37
## 65        Free        Free      445  6075 113.20
## 575 Controlled        Free     1643  3740 215.57
## 308       Free        Free      984 11392 153.58
## 230 Controlled        Free      723 73892 159.71
## 534       Free        Free      785  4186 207.17
## 222 Controlled        Free      756 48642 162.28
## 64  Controlled        Free     1476  4945 157.20
## 21        Free        Free     2290  3170 200.20
## 339       Free        Free     1591  5944 200.09
## 592       Free        Free      896  5935 127.67
## 539       Free        Free     1602  6165 210.16
## 183       Free        Free     1116  4613 185.11
## 29  Controlled        Free      291 12432 134.30
## 571       Free        Free      358  4307  54.96
## 537       Free        Free      244  9784  56.80
## 4   Controlled        Free      612 25144  85.47
## 232 Controlled Constrained      723 73892 159.71
## 459 Controlled        Free     1724  9252 219.63
## 223       Free Constrained      756 48642 162.28
## 381 Controlled        Free     2444  4455 302.33
## 9         Free        Free     1249  7811 172.63
## 200       Free Constrained     1301  4353 171.67
## 92        Free        Free     1118  3402 207.84
## 137 Controlled        Free      407 20529  75.71
## 584       Free        Free      310  6583 100.95
## 434       Free        Free     1259  5763 185.65
## 291 Controlled        Free      291  6295 150.13
## 44        Free        Free      595 30877 106.60
## 160       Free        Free      308 10451  59.77
## 71        Free        Free      184 18843  67.17
## 204       Free        Free      634  4632 181.99
## 511       Free        Free     2182  6124 200.20
## 456       Free        Free     1032  2978 118.17
## 227 Controlled        Free      183 66820 116.78
## 478       Free        Free     1851 20831 291.66
## 562       Free        Free      194  7464  85.62
## 175 Controlled        Free      276  8793  68.06
## 485       Free        Free      341 43671  79.23
## 36        Free Constrained      722  3263 237.80
## 563 Controlled        Free     1009 27103 123.89
## 256       Free Constrained     1068 40159 123.18
## 72        Free        Free     1559  8756 320.37
```

```
## 452       Free        Free     1218  4620 130.09
## 469       Free Constrained     2433  9343 299.17
## 177       Free        Free      283  9446  60.87
## 586       Free        Free     1185 14398 256.86
## 13        Free        Free      587  5654  79.17
## 458       Free        Free     2576  3987 291.51
## 266 Controlled        Free      842  7098 183.43
## 48        Free        Free      854 20718 230.87
## 450       Free        Free      177 10581  63.92
## 373       Free        Free      344  5899  66.88
## 118       Free        Free      430  5378 109.78
## 293       Free        Free      325 23041  56.43
## 318       Free        Free      192  7967  96.53
## 598       Free        Free      935  5252 182.56
## 630 Controlled        Free     1134  5449 145.53
## 290 Controlled        Free      291  6295 150.13
## 334       Free        Free     2407  8981 293.21
## 529       Free        Free      471  5303 199.80
## 302       Free        Free     1119 23222 117.97
## 536       Free        Free      682  7785 105.13
## 637       Free        Free      858  4877 129.62
## 5         Free        Free      612 25144  85.47
## 519       Free        Free      869 15887  70.16
## 415       Free        Free      592  4517  72.42
## 464       Free        Free      114  6446  70.41
## 533       Free        Free      556  7478 188.11
## 502       Free        Free      387 13378  55.16
## 341       Free        Free     2295  7130 197.42
## 600       Free        Free     2300 20007 303.82
## 570       Free        Free      447 12808  49.02
## 379       Free        Free      831 16784  91.97
## 221 Controlled        Free      756 48642 162.28
## 6         Free        Free      309 13386  56.76
## 549       Free        Free     1054  3861 119.90
## 66        Free        Free      633  4758 143.59
## 476       Free        Free      332 14363  83.74
## 209 Controlled        Free      760  7387 215.01
## 342       Free        Free      363 10529  57.33
## 363       Free Constrained      525  7664  93.55
## 70        Free        Free      734 23075 113.99
## 154       Free        Free     1052  6986 164.88
## 155       Free        Free      618 10206  89.47
```

Question4 Stepwise Regression:

```
air.lm<- lm(FARE ~ ., data= train.df)
air.lm.stepwise <- step(air.lm,direction="both")
```

```
## Start:  AIC=3652.06
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
##              Df Sum of Sq     RSS    AIC
```

```
## - COUPON    1       911  622732 3650.8
## - NEW       1      1459  623280 3651.3
## - S_INCOME  1      1460  623281 3651.3
## <none>                   621821 3652.1
## - E_INCOME  1     17499  639320 3664.2
## - SLOT      1     17769  639590 3664.4
## - PAX       1     24441  646263 3669.7
## - E_POP     1     28296  650118 3672.8
## - GATE      1     28881  650702 3673.2
## - S_POP     1     36680  658501 3679.3
## - HI        1     76469  698290 3709.2
## - SW        1    105205  727026 3729.8
## - VACATION  1    113382  735204 3735.5
## - DISTANCE  1    417379 1039200 3912.0
##
## Step:  AIC=3650.81
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - S_INCOME  1      1261  623994 3649.8
## - NEW       1      1678  624410 3650.2
## <none>                   622732 3650.8
## + COUPON    1       911  621821 3652.1
## - E_INCOME  1     17126  639859 3662.6
## - SLOT      1     18407  641139 3663.7
## - GATE      1     29285  652018 3672.2
## - E_POP     1     29484  652217 3672.4
## - PAX       1     34128  656860 3676.0
## - S_POP     1     36089  658821 3677.5
## - HI        1     78594  701326 3709.4
## - SW        1    107735  730468 3730.2
## - VACATION  1    114276  737009 3734.7
## - DISTANCE  1    824468 1447200 4078.9
##
## Step:  AIC=3649.84
## FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - NEW       1      1697  625690 3649.2
## <none>                   623994 3649.8
## + S_INCOME  1      1261  622732 3650.8
## + COUPON    1       713  623281 3651.3
## - E_INCOME  1     16167  640161 3660.9
## - SLOT      1     20012  644006 3663.9
## - E_POP     1     28559  652552 3670.7
## - GATE      1     29766  653759 3671.6
## - PAX       1     32869  656863 3674.0
## - S_POP     1     41722  665715 3680.8
## - HI        1     79501  703495 3709.0
## - SW        1    126837  750831 3742.2
## - VACATION  1    128080  752073 3743.1
## - DISTANCE  1    826967 1450960 4078.2
```

```
##
## Step:  AIC=3649.22
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                   625690 3649.2
## + NEW       1      1697  623994 3649.8
## + S_INCOME  1      1280  624410 3650.2
## + COUPON    1       907  624783 3650.5
## - E_INCOME  1     15649  641339 3659.8
## - SLOT      1     19217  644907 3662.6
## - E_POP     1     28766  654456 3670.1
## - GATE      1     29165  654856 3670.5
## - PAX       1     32706  658396 3673.2
## - S_POP     1     42648  668338 3680.9
## - HI        1     78891  704581 3707.8
## - SW        1    126577  752267 3741.2
## - VACATION  1    127066  752756 3741.5
## - DISTANCE  1    825966 1451656 4076.4
```

```
summary(air.lm.stepwise)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.148 -22.077  -2.028  21.491 107.744
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.208e+01  1.476e+01   2.851 0.004534 **
## VACATIONYes -3.876e+01  3.850e+00 -10.067  < 2e-16 ***
## SWYes       -4.053e+01  4.034e+00 -10.047  < 2e-16 ***
## HI           8.268e-03  1.042e-03   7.932 1.43e-14 ***
## E_INCOME     1.445e-03  4.089e-04   3.533 0.000450 ***
## S_POP        4.185e-06  7.176e-07   5.832 9.85e-09 ***
## E_POP        3.779e-06  7.890e-07   4.790 2.21e-06 ***
## SLOTFree    -1.685e+01  4.305e+00  -3.915 0.000103 ***
## GATEFree    -2.122e+01  4.399e+00  -4.823 1.88e-06 ***
## DISTANCE     7.367e-02  2.870e-03  25.666  < 2e-16 ***
## PAX         -7.619e-04  1.492e-04  -5.107 4.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.41 on 499 degrees of freedom
## Multiple R-squared:  0.7803, Adjusted R-squared:  0.7759
## F-statistic: 177.2 on 10 and 499 DF,  p-value: < 2.2e-16
```

```r
air.lm.stepwise.pred<-  predict(air.lm.stepwise, test.df)
```

The model above has dropped three variables based on the decreasing AIC values which are COUPON, S_INCOME, and NEW respectively which finalizes the minimum AIC value to 3649.22. The p value is much less than 0.05 and the adjusted R square value is 0.7759 which states that this model can explain 77.59% of changes in the FARE.

Question5 Exhaustive Search:

```r
search <- regsubsets(FARE ~ ., data = train.df, nbest = 1, nvmax = dim(train.df)[2],
                     method = "exhaustive")
sum <- summary(search)
sum$which
```

```
##    (Intercept) COUPON   NEW VACATIONYes SWYes    HI S_INCOME E_INCOME S_POP
## 1         TRUE  FALSE FALSE       FALSE FALSE FALSE    FALSE    FALSE FALSE
## 2         TRUE  FALSE FALSE       FALSE  TRUE FALSE    FALSE    FALSE FALSE
## 3         TRUE  FALSE FALSE        TRUE  TRUE FALSE    FALSE    FALSE FALSE
## 4         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE
## 5         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE
## 6         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE
## 7         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE  TRUE
## 8         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE
## 9         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE  TRUE
## 10        TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE
## 11        TRUE  FALSE  TRUE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE
## 12        TRUE  FALSE  TRUE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE
## 13        TRUE   TRUE  TRUE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE
##    E_POP SLOTFree GATEFree DISTANCE   PAX
## 1  FALSE    FALSE    FALSE     TRUE FALSE
## 2  FALSE    FALSE    FALSE     TRUE FALSE
## 3  FALSE    FALSE    FALSE     TRUE FALSE
## 4  FALSE    FALSE    FALSE     TRUE FALSE
## 5  FALSE     TRUE    FALSE     TRUE FALSE
## 6  FALSE     TRUE     TRUE     TRUE FALSE
## 7   TRUE    FALSE    FALSE     TRUE  TRUE
## 8   TRUE    FALSE    FALSE     TRUE  TRUE
## 9   TRUE     TRUE     TRUE     TRUE  TRUE
## 10  TRUE     TRUE     TRUE     TRUE  TRUE
## 11  TRUE     TRUE     TRUE     TRUE  TRUE
## 12  TRUE     TRUE     TRUE     TRUE  TRUE
## 13  TRUE     TRUE     TRUE     TRUE  TRUE
```

```r
sum$rsq
```

```
##  [1] 0.4168069 0.5793894 0.6966218 0.7232479 0.7366555 0.7565835 0.7607777
##  [8] 0.7674947 0.7748171 0.7803115 0.7809073 0.7813501 0.7816700
```

```r
sum$adjr2
```

```
##  [1] 0.4156589 0.5777302 0.6948231 0.7210558 0.7340429 0.7536799 0.7574419
##  [8] 0.7637820 0.7707638 0.7759090 0.7760679 0.7760708 0.7759476
```

```r
sum$cp
```

```
##  [1] 818.89220 451.53899 187.21153 128.72255 100.26346  56.99127  49.46286
##  [8]  36.20326  21.56831  11.08605  11.73270  12.72670  14.00000
```

From the above Exhaustive search, we need to select the set of variables having highest adjusted R square value. The highest adjusted R square value in the above model is 0.77607 which is the 12th row and has 12 variables. But, for accuracy we will consider the CP values as the values look close to each other. From CP value results, we can see that cp<= p+1 is satisfies by 11th row and so we consider all the variables except COUPON and S_INCOME. By comparing Exhaustive search model with the stepwise regression model we see that adjusted R squared value in stepwise model was 0.7759 whereas in exhaustive model is 0.77607. Also the stepwise model dropped three variables whereas the exhaustive model dropped two variables.

Question6 Comparing Predictive Accuracy:

```r
accuracy(air.lm.stepwise.pred, test.df$FARE) # Stepwise Accuracy
```

```
##               ME    RMSE      MAE       MPE     MAPE
## Test set 3.06081 36.8617 27.70568 -5.938062 21.62142
```

```r
air.exhaustive.lm <- lm(FARE~ COUPON+NEW+VACATION+SW+HI+E_INCOME+S_POP+E_POP+SLOT+GATE+
                       DISTANCE+PAX, data = train.df )
predict.exhaustive.lm <- predict(air.exhaustive.lm, test.df)
accuracy(predict.exhaustive.lm, test.df$FARE) # Exhaustive Accuracy
```

```
##                ME     RMSE      MAE       MPE     MAPE
## Test set 3.065742 36.97323 27.59975 -5.852725 21.47488
```

By comparing the predictive accuracies of both models using RMSE measure from above results, we can see that stepwise RMSE is 36.8617 and that of exhaustie is 36.97323 which helps us concluding that stepwise regression is better as it has low RMSE value.Also, the number of variable left in stepwise regression were 10 (excluding FARE) and that in exhaustive were 11 (excluding FARE).

Question7 Predict Average Fare on a route:

```r
test2.df <- data.frame(COUPON = 1.202, NEW = 3, VACATION = 'No', SW = 'No',
                       HI = 4442.141, S_INCOME = 28760, E_INCOME = 27664, S_POP = 4557004,
                       E_POP = 3195503, SLOT = 'Free', GATE = 'Free',
                       PAX = 12782, DISTANCE = 1976)
predict.exhaustive.lm2 <- predict(air.exhaustive.lm, test2.df)
predict.exhaustive.lm2
```

```
##        1
## 245.2815
```

From the above results of exhaustive search model, the average fare on a route with given characteristics is $245.2815.

Question8 Predict Reduction in average fare:

```
test3.df <- data.frame(COUPON = 1.202, NEW = 3, VACATION = 'No',
                        SW = 'Yes', HI = 4442.141, S_INCOME = 28760,
                        E_INCOME = 27664, S_POP = 4557004, E_POP = 3195503,
                        SLOT = 'Free', GATE = 'Free', PAX = 12782, DISTANCE = 1976)
predict.exhaustive.lm3 <- predict(air.exhaustive.lm, test3.df)
predict.exhaustive.lm3
```

```
##        1
## 204.8958
```

```
predict.exhaustive.lm2-predict.exhaustive.lm3
```

```
##        1
## 40.38569
```

If southwest decides to cover the route, using the exhaustive search model, the average fare turns out to be \$204.8958 with a reduction of \$40.38569.

Question9 Backward Selection Regression:

```
air.lm.bselect <- step(air.lm, direction = "backward")
```

```
## Start:  AIC=3652.06
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
##             Df Sum of Sq     RSS    AIC
## - COUPON     1       911  622732 3650.8
## - NEW        1      1459  623280 3651.3
## - S_INCOME   1      1460  623281 3651.3
## <none>                    621821 3652.1
## - E_INCOME   1     17499  639320 3664.2
## - SLOT       1     17769  639590 3664.4
## - PAX        1     24441  646263 3669.7
## - E_POP      1     28296  650118 3672.8
## - GATE       1     28881  650702 3673.2
## - S_POP      1     36680  658501 3679.3
## - HI         1     76469  698290 3709.2
## - SW         1    105205  727026 3729.8
## - VACATION   1    113382  735204 3735.5
## - DISTANCE   1    417379 1039200 3912.0
##
## Step:  AIC=3650.81
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##             Df Sum of Sq     RSS    AIC
## - S_INCOME   1      1261  623994 3649.8
## - NEW        1      1678  624410 3650.2
## <none>                    622732 3650.8
## - E_INCOME   1     17126  639859 3662.6
## - SLOT       1     18407  641139 3663.7
```

```
## - GATE      1     29285  652018 3672.2
## - E_POP     1     29484  652217 3672.4
## - PAX       1     34128  656860 3676.0
## - S_POP     1     36089  658821 3677.5
## - HI        1     78594  701326 3709.4
## - SW        1    107735  730468 3730.2
## - VACATION  1    114276  737009 3734.7
## - DISTANCE  1    824468 1447200 4078.9
##
## Step:  AIC=3649.84
## FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - NEW       1      1697  625690 3649.2
## <none>                   623994 3649.8
## - E_INCOME  1     16167  640161 3660.9
## - SLOT      1     20012  644006 3663.9
## - E_POP     1     28559  652552 3670.7
## - GATE      1     29766  653759 3671.6
## - PAX       1     32869  656863 3674.0
## - S_POP     1     41722  665715 3680.8
## - HI        1     79501  703495 3709.0
## - SW        1    126837  750831 3742.2
## - VACATION  1    128080  752073 3743.1
## - DISTANCE  1    826967 1450960 4078.2
##
## Step:  AIC=3649.22
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                   625690 3649.2
## - E_INCOME  1     15649  641339 3659.8
## - SLOT      1     19217  644907 3662.6
## - E_POP     1     28766  654456 3670.1
## - GATE      1     29165  654856 3670.5
## - PAX       1     32706  658396 3673.2
## - S_POP     1     42648  668338 3680.9
## - HI        1     78891  704581 3707.8
## - SW        1    126577  752267 3741.2
## - VACATION  1    127066  752756 3741.5
## - DISTANCE  1    825966 1451656 4076.4
```

```r
summary(air.lm.bselect)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.148 -22.077  -2.028  21.491 107.744
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.208e+01  1.476e+01   2.851 0.004534 **
## VACATIONYes -3.876e+01  3.850e+00 -10.067  < 2e-16 ***
## SWYes       -4.053e+01  4.034e+00 -10.047  < 2e-16 ***
## HI           8.268e-03  1.042e-03   7.932 1.43e-14 ***
## E_INCOME     1.445e-03  4.089e-04   3.533 0.000450 ***
## S_POP        4.185e-06  7.176e-07   5.832 9.85e-09 ***
## E_POP        3.779e-06  7.890e-07   4.790 2.21e-06 ***
## SLOTFree    -1.685e+01  4.305e+00  -3.915 0.000103 ***
## GATEFree    -2.122e+01  4.399e+00  -4.823 1.88e-06 ***
## DISTANCE     7.367e-02  2.870e-03  25.666  < 2e-16 ***
## PAX         -7.619e-04  1.492e-04  -5.107 4.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.41 on 499 degrees of freedom
## Multiple R-squared:  0.7803, Adjusted R-squared:  0.7759
## F-statistic: 177.2 on 10 and 499 DF,  p-value: < 2.2e-16
```

```
air.lm.bselect.pred <- predict(air.lm.bselect, test.df)
accuracy(air.lm.bselect.pred, test.df$FARE)
```

```
##                ME    RMSE      MAE       MPE     MAPE
## Test set 3.06081 36.8617 27.70568 -5.938062 21.62142
```

As we can see in the above results, the backward selection model dropped three variable, COUPON, S_INCOME, and NEW. The RMSE value using backward selection model is 36.8617 which is same as that of stepwise regression because both the models dropprd same set of variables. Although the RMSE value of backward selection model is less than that of the exhaustive search model.

Question10 Backward Selection model using stepAIC():

```
air.lm.step2 <- stepAIC(air.lm, direction = "backward")
```

```
## Start:  AIC=3652.06
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq    RSS    AIC
## - COUPON    1       911 622732 3650.8
## - NEW       1      1459 623280 3651.3
## - S_INCOME  1      1460 623281 3651.3
## <none>                  621821 3652.1
## - E_INCOME  1     17499 639320 3664.2
## - SLOT      1     17769 639590 3664.4
## - PAX       1     24441 646263 3669.7
## - E_POP     1     28296 650118 3672.8
## - GATE      1     28881 650702 3673.2
## - S_POP     1     36680 658501 3679.3
## - HI        1     76469 698290 3709.2
## - SW        1    105205 727026 3729.8
```

```
## - VACATION  1    113382  735204 3735.5
## - DISTANCE  1    417379 1039200 3912.0
##
## Step:  AIC=3650.81
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - S_INCOME  1      1261  623994 3649.8
## - NEW       1      1678  624410 3650.2
## <none>                   622732 3650.8
## - E_INCOME  1     17126  639859 3662.6
## - SLOT      1     18407  641139 3663.7
## - GATE      1     29285  652018 3672.2
## - E_POP     1     29484  652217 3672.4
## - PAX       1     34128  656860 3676.0
## - S_POP     1     36089  658821 3677.5
## - HI        1     78594  701326 3709.4
## - SW        1    107735  730468 3730.2
## - VACATION  1    114276  737009 3734.7
## - DISTANCE  1    824468 1447200 4078.9
##
## Step:  AIC=3649.84
## FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - NEW       1      1697  625690 3649.2
## <none>                   623994 3649.8
## - E_INCOME  1     16167  640161 3660.9
## - SLOT      1     20012  644006 3663.9
## - E_POP     1     28559  652552 3670.7
## - GATE      1     29766  653759 3671.6
## - PAX       1     32869  656863 3674.0
## - S_POP     1     41722  665715 3680.8
## - HI        1     79501  703495 3709.0
## - SW        1    126837  750831 3742.2
## - VACATION  1    128080  752073 3743.1
## - DISTANCE  1    826967 1450960 4078.2
##
## Step:  AIC=3649.22
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                   625690 3649.2
## - E_INCOME  1     15649  641339 3659.8
## - SLOT      1     19217  644907 3662.6
## - E_POP     1     28766  654456 3670.1
## - GATE      1     29165  654856 3670.5
## - PAX       1     32706  658396 3673.2
## - S_POP     1     42648  668338 3680.9
## - HI        1     78891  704581 3707.8
## - SW        1    126577  752267 3741.2
```

```
## - VACATION  1    127066  752756 3741.5
## - DISTANCE  1    825966 1451656 4076.4
```

```r
summary(air.lm.step2)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.148 -22.077  -2.028  21.491 107.744
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.208e+01  1.476e+01   2.851 0.004534 **
## VACATIONYes -3.876e+01  3.850e+00 -10.067  < 2e-16 ***
## SWYes       -4.053e+01  4.034e+00 -10.047  < 2e-16 ***
## HI           8.268e-03  1.042e-03   7.932 1.43e-14 ***
## E_INCOME     1.445e-03  4.089e-04   3.533 0.000450 ***
## S_POP        4.185e-06  7.176e-07   5.832 9.85e-09 ***
## E_POP        3.779e-06  7.890e-07   4.790 2.21e-06 ***
## SLOTFree    -1.685e+01  4.305e+00  -3.915 0.000103 ***
## GATEFree    -2.122e+01  4.399e+00  -4.823 1.88e-06 ***
## DISTANCE     7.367e-02  2.870e-03  25.666  < 2e-16 ***
## PAX         -7.619e-04  1.492e-04  -5.107 4.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.41 on 499 degrees of freedom
## Multiple R-squared:  0.7803, Adjusted R-squared:  0.7759
## F-statistic: 177.2 on 10 and 499 DF,  p-value: < 2.2e-16
```

```r
air.lm.step2.pred <- predict(air.lm.step2, test.df)
accuracy(air.lm.step2.pred, test.df$FARE)
```

```
##               ME    RMSE      MAE       MPE     MAPE
## Test set 3.06081 36.8617 27.70568 -5.938062 21.62142
```

The AIC value decreases as we drop each variable till we get the best-fit model. The AIC values in the Backward and StepAIC methods are same as 3649.22 (minimum). The values came out to be same because both the models have dropped the same set of variables which are COUPON, NEW, and S_Income. Hence, we did not see any effect of using StepAIC function in this case.