```r
if(!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse, reshape, reshape2, gplots, ggmap, cowplot, data.table, ggplot2, GGally, caret

search()
```

```r
Airfare.data =  fread("Airfares.csv")
#Airfare.data = Airfare.data[,-19]
str(Airfare.data)
```

```
## Classes 'data.table' and 'data.frame':   638 obs. of  18 variables:
##  $ S_CODE  : chr  "*" "*" "*" "ORD" ...
##  $ S_CITY  : chr  "Dallas/Fort Worth   TX" "Atlanta             GA" "Boston              MA" "Chicag
##  $ E_CODE  : chr  "*" "*" "*" "*" ...
##  $ E_CITY  : chr  "Amarillo            TX" "Baltimore/Wash Intl MD" "Baltimore/Wash Intl MD" "Baltim
##  $ COUPON  : num  1 1.06 1.06 1.06 1.06 1.01 1.28 1.15 1.33 1.6 ...
##  $ NEW     : int  3 3 3 3 3 3 3 3 3 2 ...
##  $ VACATION: chr  "No" "No" "No" "No" ...
##  $ SW      : chr  "Yes" "No" "No" "Yes" ...
##  $ HI      : num  5292 5419 9185 2657 2657 ...
##  $ S_INCOME: num  28637 26993 30124 29260 29260 ...
##  $ E_INCOME: num  21112 29838 29838 29838 29838 ...
##  $ S_POP   : int  3036732 3532657 5787293 7830332 7830332 2230955 3036732 1440377 3770125 1694803 ..
##  $ E_POP   : int  205711 7145897 7145897 7145897 7145897 7145897 7145897 7145897 7145897 7145897 ...
##  $ SLOT    : chr  "Free" "Free" "Free" "Controlled" ...
##  $ GATE    : chr  "Free" "Free" "Free" "Free" ...
##  $ DISTANCE: int  312 576 364 612 612 309 1220 921 1249 964 ...
##  $ PAX     : int  7864 8820 6452 25144 25144 13386 4625 5512 7811 4657 ...
##  $ FARE    : num  64.1 174.5 207.8 85.5 85.5 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
Airfare = Airfare.data[, -c(1,2,3,4)]      # Removing first 4 columns
summary(Airfare)
```

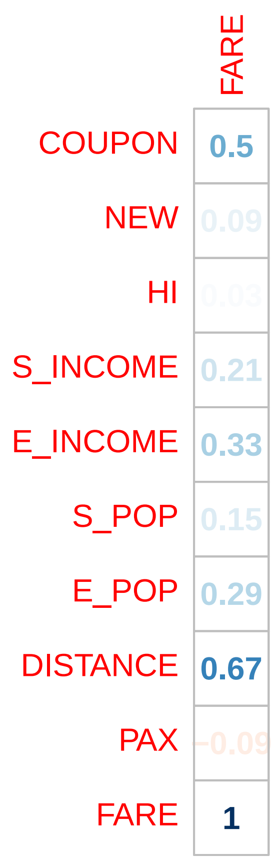
```
##      COUPON           NEW          VACATION              SW
##  Min.   :1.000   Min.   :0.000   Length:638         Length:638
##  1st Qu.:1.040   1st Qu.:3.000   Class :character   Class :character
##  Median :1.150   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1.202   Mean   :2.754
##  3rd Qu.:1.298   3rd Qu.:3.000
##  Max.   :1.940   Max.   :3.000
##        HI           S_INCOME        E_INCOME         S_POP
##  Min.   : 1230   Min.   :14600   Min.   :14600   Min.   :  29838
##  1st Qu.: 3090   1st Qu.:24706   1st Qu.:23903   1st Qu.:1862106
##  Median : 4208   Median :28637   Median :26409   Median :3532657
##  Mean   : 4442   Mean   :27760   Mean   :27664   Mean   :4557004
##  3rd Qu.: 5481   3rd Qu.:29694   3rd Qu.:31981   3rd Qu.:7830332
##  Max.   :10000   Max.   :38813   Max.   :38813   Max.   :9056076
##      E_POP             SLOT               GATE              DISTANCE
##  Min.   : 111745   Length:638         Length:638         Min.   : 114.0
##  1st Qu.:1228816   Class :character   Class :character   1st Qu.: 455.0
##  Median :2195215   Mode  :character   Mode  :character   Median : 850.0
##  Mean   :3194503                                         Mean   : 975.7
```

```
##  3rd Qu.:4549784                                         3rd Qu.:1306.2
##  Max.   :9056076                                         Max.   :2764.0
##       PAX             FARE
##  Min.   : 1504   Min.   : 42.47
##  1st Qu.: 5328   1st Qu.:106.29
##  Median : 7792   Median :144.60
##  Mean   :12782   Mean   :160.88
##  3rd Qu.:14090   3rd Qu.:209.35
##  Max.   :73892   Max.   :402.02
```

Question 1)

Create a correlation table and scatterplots between FARE and the predictors. What seems to be the best single predictor of FARE? Explain your answer.

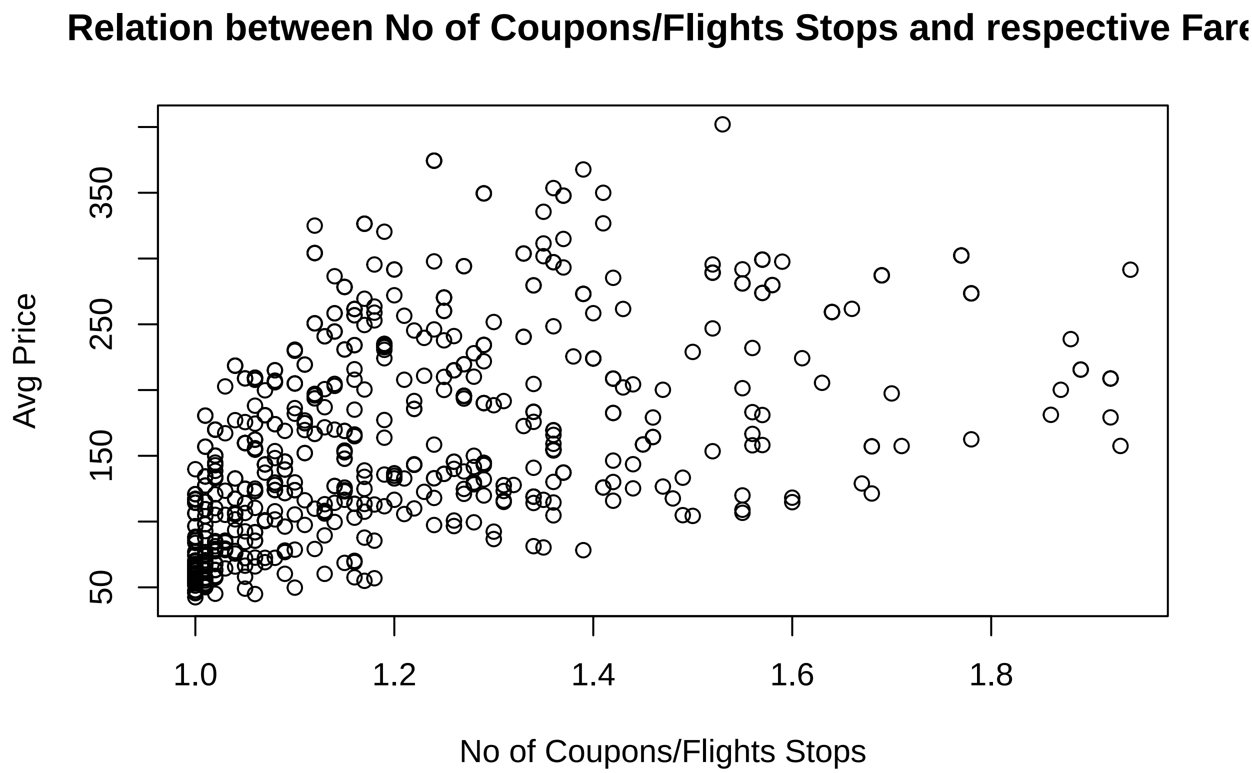
```r
Airfare.corr = select_if(Airfare, is.numeric) # selecting the one which are numeric
coorelation = corrplot(cor(Airfare.corr)[, 10 ,  drop = FALSE],  method = "number" , cl.pos='n')
```



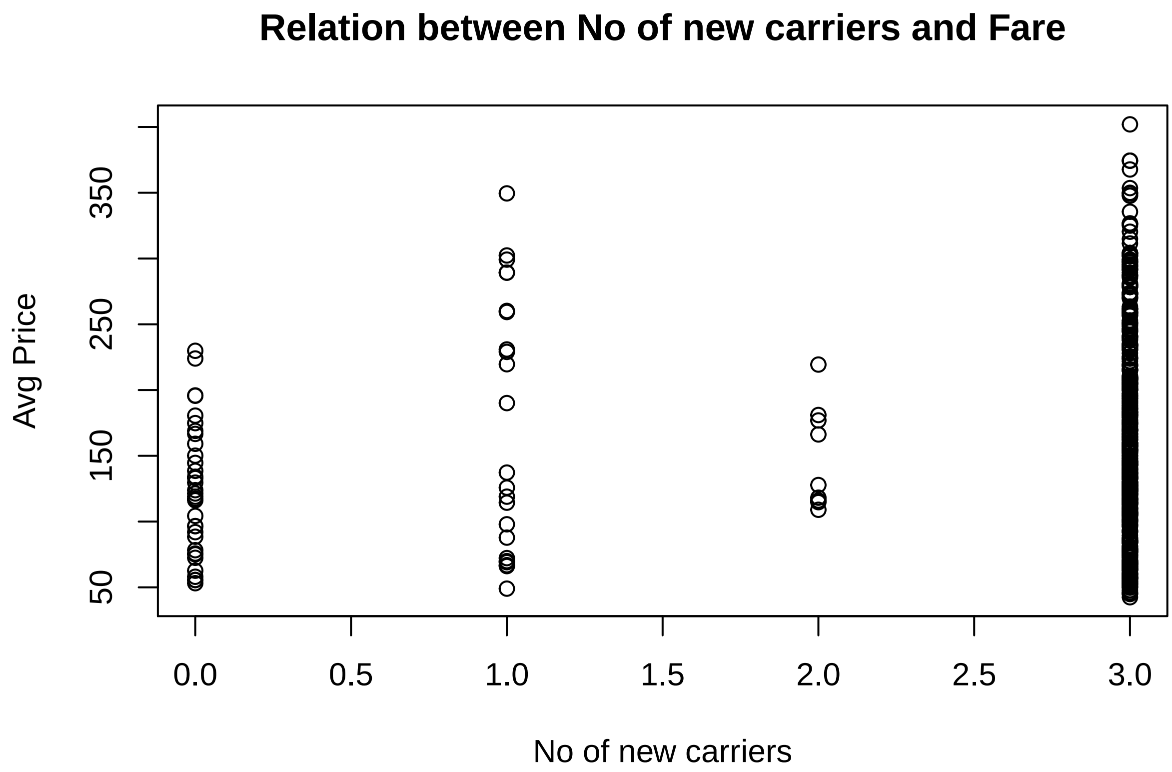
Answer 1)

From the above plot, it can be clearly seen that Distance is the best predictor of FARE with correlation value of 0.67. Since the correlation value is positive, it means that Distance and FARE are positively correlated, that is, with increase in Distance, FARE also increases. Below, we have created individual scatter plots to observe the behaviors of all the predictors with FARE.

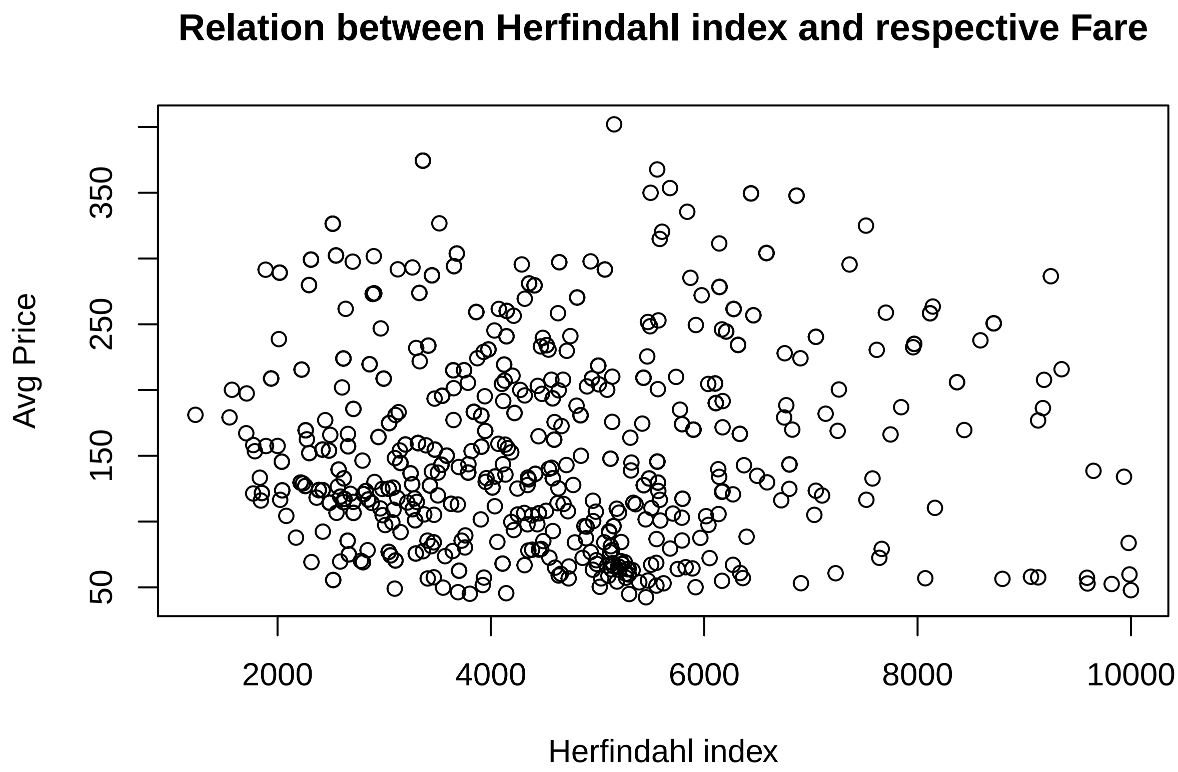
```
plot(x = Airfare$COUPON, y = Airfare$FARE,type = "p", main = "Relation between No of Coupons/Flights Sto
```



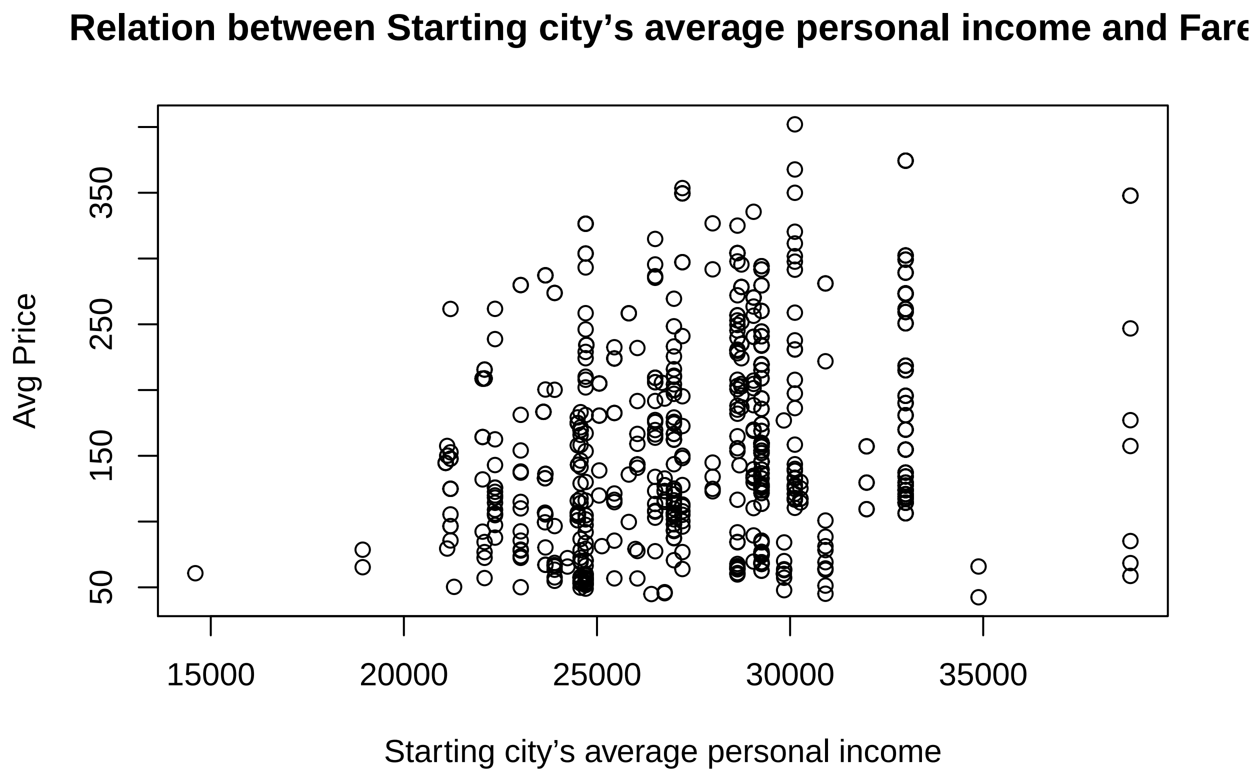

```
plot(x = Airfare$NEW, y = Airfare$FARE,type = "p", main = "Relation between No of new carriers and Fare
```



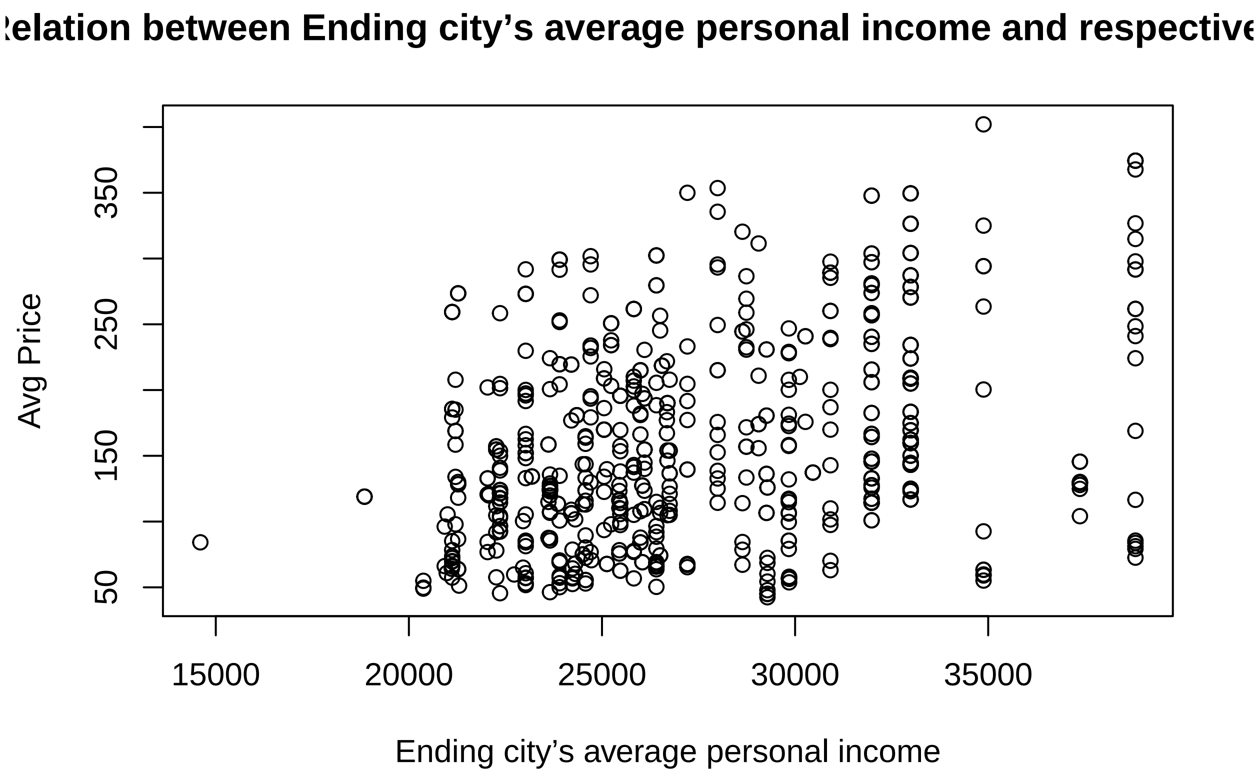

```
plot(x = Airfare$HI, y = Airfare$FARE,type = "p", main = "Relation between Herfindahl index and respecti
```



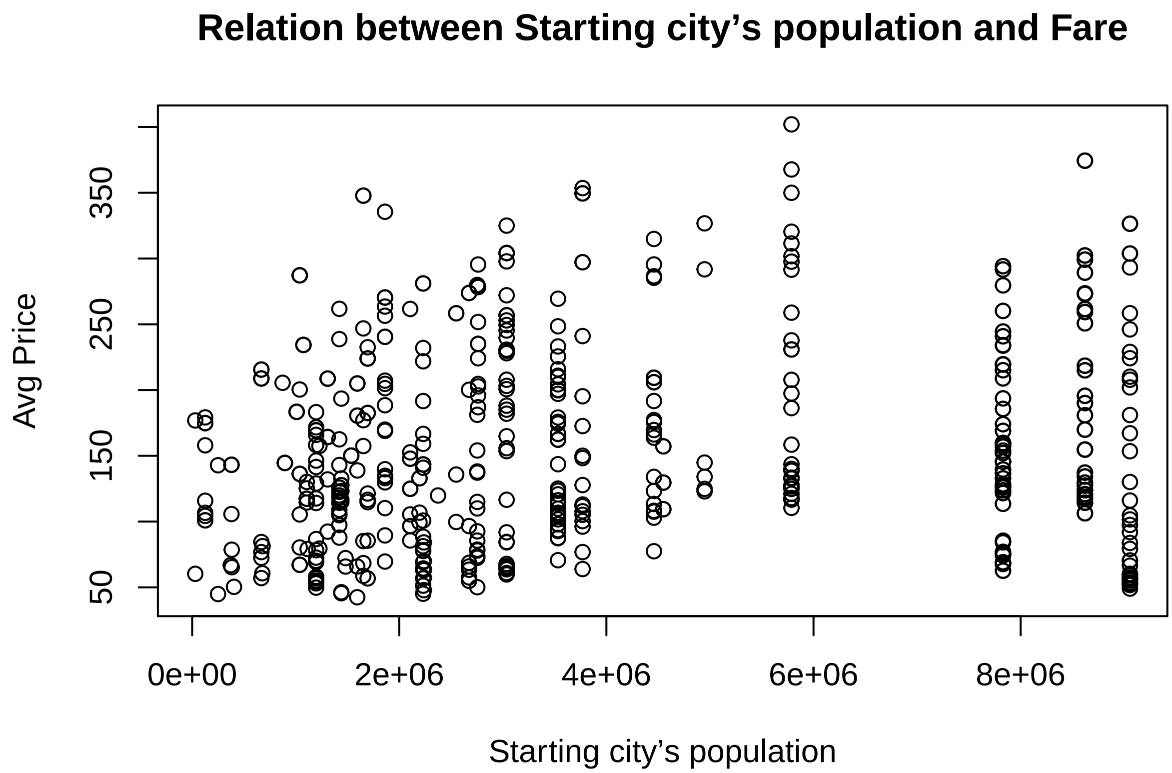

```
plot(x = Airfare$S_INCOME, y = Airfare$FARE,type = "p", main = "Relation between Starting city's averag
```



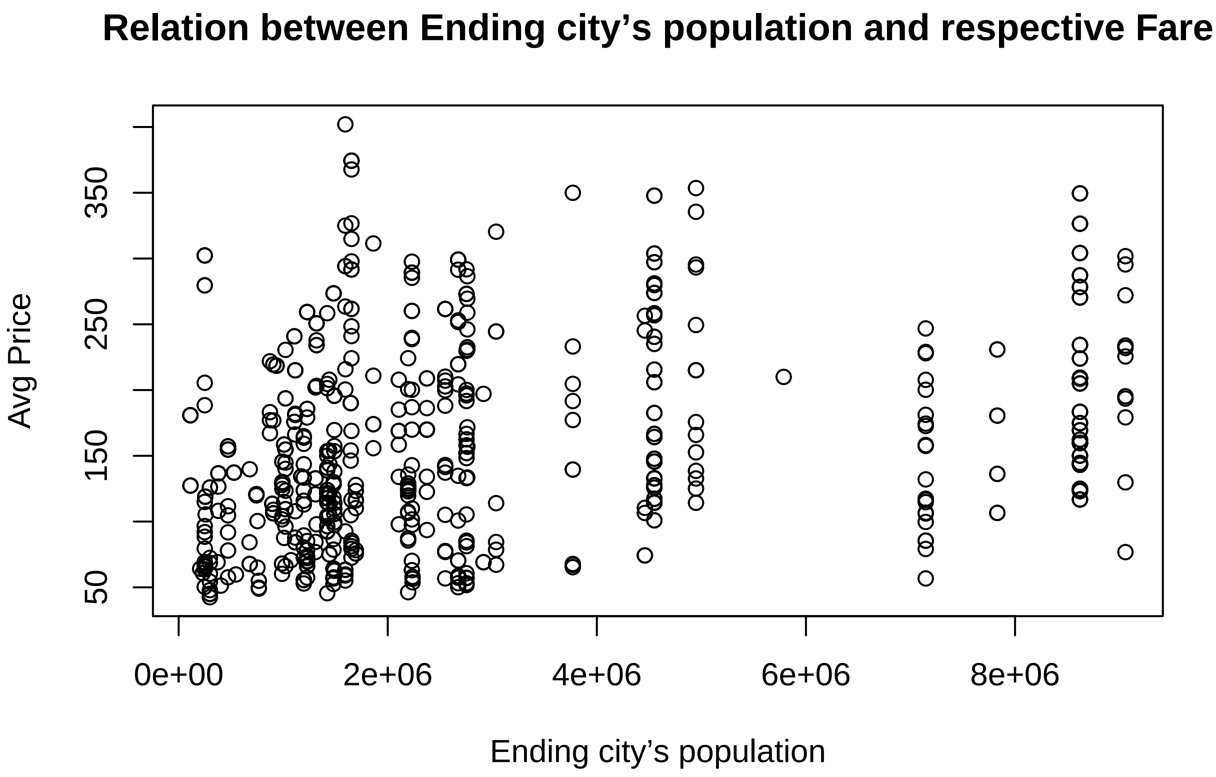

```
plot(x = Airfare$E_INCOME, y = Airfare$FARE,type = "p", main = "Relation between Ending city's average p
```



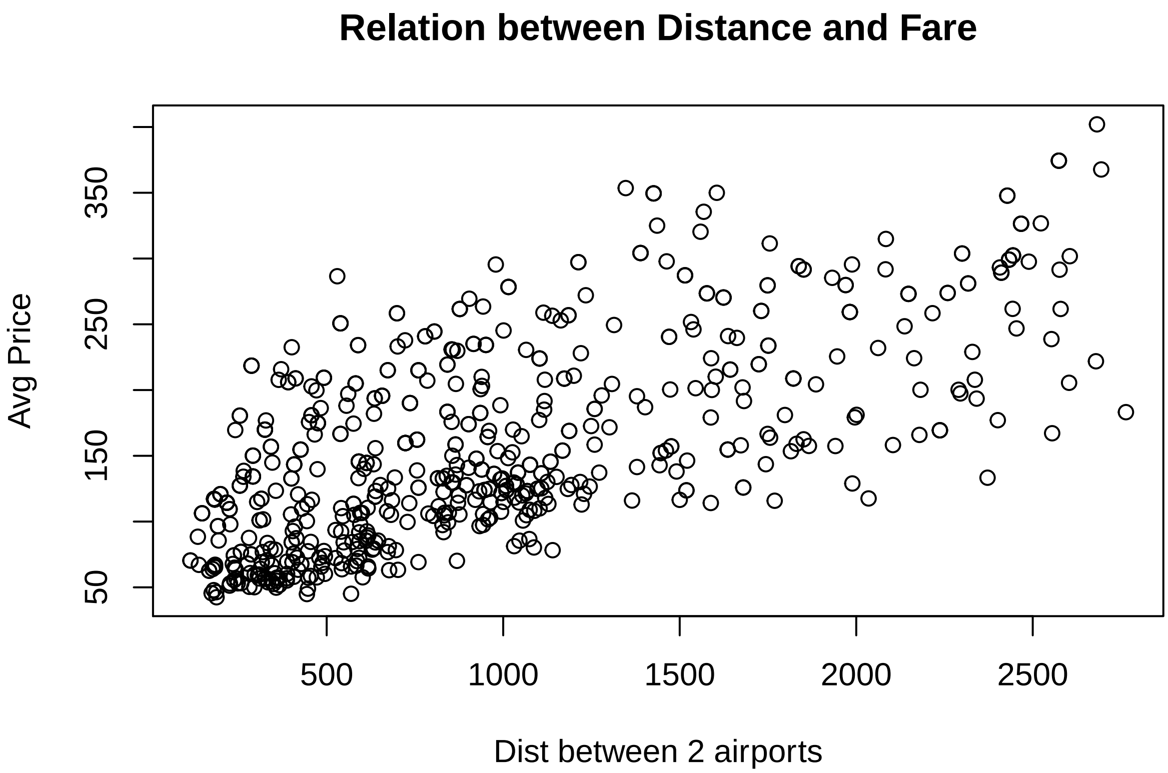

```
plot(x = Airfare$S_POP, y = Airfare$FARE,type = "p", main = "Relation between Starting city's populatio
```



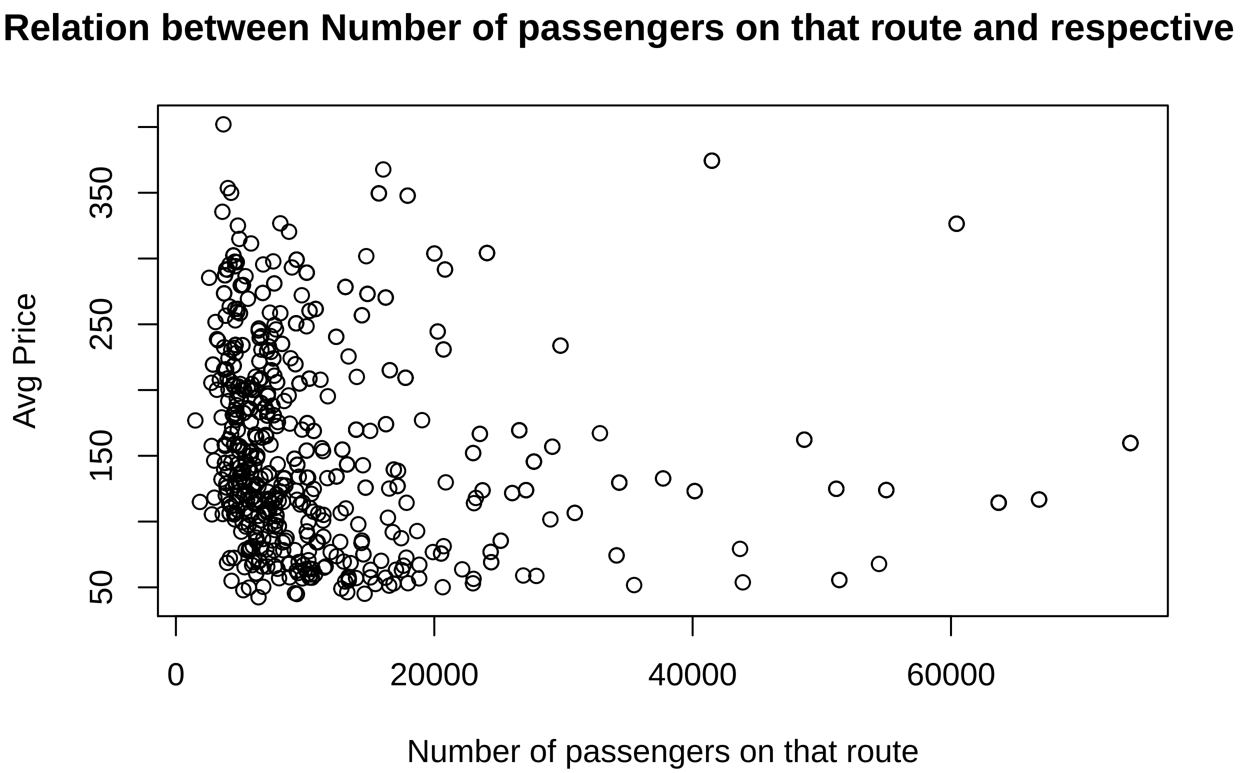

```
plot(x = Airfare$E_POP, y = Airfare$FARE,type = "p", main = "Relation between Ending city's population a
```




```
plot(x = Airfare$DISTANCE, y = Airfare$FARE,type = "p", main = "Relation between Distance and Fare", xla
```




```
plot(x = Airfare$PAX, y = Airfare$FARE,type = "p", main = "Relation between Number of passengers on tha
```



Question 2)

Explore the categorical predictors by computing the percentage of flights in each category. Create a pivot table with the average fare in each category. Which categorical predictor seems best for predicting FARE? Explain your answer.

```
# PivotTable
air <- Airfare
Vacation_Pivot <- air %>%
        dplyr::select(VACATION,FARE) %>%
        group_by(VACATION) %>%
        summarise(VCount = length(VACATION),VTotal = nrow(air), VPercent = percent(length(VACATION)/nro

Vacation_Pivot
```

```
## # A tibble: 2 x 5
##   VACATION VCount VTotal VPercent AvgFare
##   <chr>     <int>  <int> <chr>      <dbl>
## 1 No          468    638 73.4%       174.
## 2 Yes         170    638 26.6%       126.
```

```
SW_Pivot <- air %>%
       dplyr:: select(SW,FARE) %>%
```

```r
        group_by(SW) %>%
        summarise(WCount = length(SW),WTotal = nrow(air), WPercent = percent(length(SW)/nrow(air)), Avg

SW_Pivot
```

```
## # A tibble: 2 x 5
##   SW    WCount WTotal WPercent AvgFare
##   <chr>  <int>  <int> <chr>      <dbl>
## 1 No       444    638 69.6%      188.
## 2 Yes      194    638 30.4%       98.4
```

```r
Gate_Pivot <- air %>%
        dplyr::select(GATE,FARE) %>%
        group_by(GATE) %>%
        summarise(GCount = length(GATE),GTotal = nrow(air), GPercent = percent(length(GATE)/nrow(air)),

Gate_Pivot
```

```
## # A tibble: 2 x 5
##   GATE        GCount GTotal GPercent AvgFare
##   <chr>        <int>  <int> <chr>      <dbl>
## 1 Constrained    124    638 19.4%       193.
## 2 Free           514    638 80.6%       153.
```

```r
Slot_Pivot <- air %>%
        dplyr::select(SLOT,FARE) %>%
        group_by(SLOT) %>%
        summarise(SCount = length(SLOT),STotal = nrow(air), SPercent = percent(length(SLOT)/nrow(air)),

Slot_Pivot
```

```
## # A tibble: 2 x 5
##   SLOT       SCount STotal SPercent AvgFare
##   <chr>       <int>  <int> <chr>      <dbl>
## 1 Controlled    182    638 28.5%       186.
## 2 Free          456    638 71.5%       151.
```

Answer 2)

As seen above, there are 4 categorical predictors - VACATION, SW, GATE and SLOT. SW is a low-cost entrant and the average FARE is lowest for SW (98.38227), where it is serving the routes(YES). therefore, SW seems to be the most significant categorical predictor for calculating average FARE.

Question 3)

Create data partition by assigning 80% of the records to the training dataset. Use rounding if 80% of the index generates a fraction. Also, set the seed at 42.

```r
  # converting dummy variables

nrows<-NROW(Airfare)
Sample_size <-nrows*.8

set.seed(42)
train.index <- sample(c(1:638), Sample_size)
Airfare.training <- Airfare[train.index, ]
Airfare.test <- Airfare[-train.index, ]
summary(Airfare.training)
```

```
##      COUPON           NEW          VACATION              SW
##  Min.   :1.000   Min.   :0.000   Length:510         Length:510
##  1st Qu.:1.040   1st Qu.:3.000   Class :character   Class :character
##  Median :1.150   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1.204   Mean   :2.749
##  3rd Qu.:1.300   3rd Qu.:3.000
##  Max.   :1.930   Max.   :3.000
##        HI           S_INCOME        E_INCOME         S_POP
##  Min.   : 1230   Min.   :14600   Min.   :14600   Min.   :  29838
##  1st Qu.: 3091   1st Qu.:24706   1st Qu.:23903   1st Qu.:1862106
##  Median : 4197   Median :28637   Median :26409   Median :3532657
##  Mean   : 4468   Mean   :27854   Mean   :27601   Mean   :4532113
##  3rd Qu.: 5537   3rd Qu.:29846   3rd Qu.:31981   3rd Qu.:7830332
##  Max.   :10000   Max.   :38813   Max.   :38813   Max.   :9056076
##      E_POP              SLOT               GATE              DISTANCE
##  Min.   : 111745   Length:510         Length:510         Min.   : 114.0
##  1st Qu.:1228816   Class :character   Class :character   1st Qu.: 457.2
##  Median :2195215   Mode  :character   Mode  :character   Median : 865.0
##  Mean   :3161555                                         Mean   : 989.8
##  3rd Qu.:4549784                                         3rd Qu.:1389.0
##  Max.   :9056076                                         Max.   :2764.0
##       PAX             FARE
##  Min.   : 1504   Min.   : 42.47
##  1st Qu.: 5284   1st Qu.:107.60
##  Median : 7792   Median :143.44
##  Mean   :12584   Mean   :161.34
##  3rd Qu.:13957   3rd Qu.:209.35
##  Max.   :73892   Max.   :402.02
```

```r
# 14 cols
# COUPON NEW VACATION SW HI S_INCOME E_INCOME
# S_POP E_POP SLOT GATE DISTANCE PAX FARE
```

Question 4)

Using leaps package, run stepwise regression to reduce the number of predictors. Discuss the results from this model.

```r
modelLr = lm(FARE ~., data = Airfare.training)
options(scipen = 999)
modelLr.stepwise <- step(modelLr, 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
```

```r
summary(modelLr.stepwise)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = Airfare.training)
##
## 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)  42.0764345686  14.7566725244   2.851             0.004534 **
## VACATIONYes -38.7574569132   3.8500841929 -10.067 < 0.0000000000000002 ***
## SWYes       -40.5282166043   4.0337560764 -10.047 < 0.0000000000000002 ***
## HI            0.0082681499   0.0010423739   7.932   0.0000000000000143 ***
## E_INCOME      0.0014446281   0.0004089281   3.533             0.000450 ***
## S_POP         0.0000041850   0.0000007176   5.832   0.0000000098509604 ***
## E_POP         0.0000037791   0.0000007890   4.790   0.0000022053722984 ***
## SLOTFree    -16.8515659965   4.3045728245  -3.915             0.000103 ***
## GATEFree    -21.2165142735   4.3991611435  -4.823   0.0000018824635124 ***
## DISTANCE      0.0736714582   0.0028704349  25.666 < 0.0000000000000002 ***
```

```
## PAX          -0.0007619280   0.0001491869  -5.107   0.0000004660838631 ***
## ---
## 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: < 0.00000000000000022
```

```
modelLr.stepwise.pred <- predict(modelLr.stepwise , Airfare.test)
AccuracySR<-accuracy(modelLr.stepwise.pred, Airfare.test$FARE)
```

Answer 4)

We can see in the output that there are 4 models created and at every step, R has reduced one column. Finally, the last model where 3 columns- COUPON, S_INCOME, NEW (respectively) have been reduced, gives the lowest AIC and explains 77.59% of the data which is decent enough. All the colums in the model are significant, bt looking at the p-values and astericks.

Question 5)

Repeat the process in (4) using exhaustive search instead of stepwise regression. Compare the resulting best model to the one you obtained in (4) in terms of the predictors included in the final model.

```
search <- regsubsets(FARE ~ ., data = Airfare.training, nbest = 1 , nvmax = dim(Airfare.training)[2],  m
sum <- summary(search)

# show models
sum$which
```

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

```
## 6  FALSE FALSE     TRUE     TRUE     TRUE FALSE
## 7   TRUE  TRUE    FALSE    FALSE     TRUE  TRUE
## 8   TRUE  TRUE    FALSE    FALSE     TRUE  TRUE
## 9   TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
## 10  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
## 11  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
## 12  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
## 13  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
```

```
# show metrics
sum$adjr2 # the 12th model is best
```

```
##  [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
```

```
sum$cp    # the 10th model is best
```

```
##  [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
```

```
sum$rsq   ### adj Rsq is maximum 0.7760708 in 12th model, but according to Mallow's cp, we get the best
```

```
##  [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
```

```
coefficient <- coef(search,12)
exhaustive.lm.model <- lm(FARE~NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP + E_POP + SLOT + G
options(scipen = 999)
exhaustive.lm.model.pred <- predict(exhaustive.lm.model,Airfare.test)
AccuracyES <- accuracy(exhaustive.lm.model.pred, Airfare.test$FARE)
```

Answer 5)

By comparing the model selected by exhaustive search and the model in 4th question above, we see that they are very similar. Exactly same 12 columns have been included/ excluded in the model. Rsquare: by considering values of all the models, we see that the 12th model has the highest Rsquare and adjusted Rsqaure:0.77607.

#cp: we see that the 10th model has the best cp. since the difference with 11th model 11.08605-11 is lowest in the 10th model, we select the 10th model.

Question 6)

Compare the predictive accuracy of both models—stepwise regression and exhaustive search—using measures such as RMSE.

```
MACHINE_LEARNING_MODELS = c("Stepwise Regression","Exhaustive Search")
ERROR = rbind(AccuracySR,AccuracyES)

df = cbind(MACHINE_LEARNING_MODELS,ERROR)
df
```

```
##          MACHINE_LEARNING_MODELS ME                 RMSE
## Test set "Stepwise Regression"   "3.06081020839502" "36.8617049624065"
## Test set "Exhaustive Search"     "3.44349560217131" "36.4118412058548"
##          MAE                MPE                 MAPE
## Test set "27.7056762277249" "-5.93806225686087" "21.6214206121135"
## Test set "27.2649304928755" "-5.39424849356953" "21.1002899245629"
```

Answer 6)

By looking at the RMSE values of both methods, we see that the difference is not very high. since the RMSE value for exhaustive search method model is lower, we can say that it has a slightly better fit than the other.

Question 7)

Using the exhaustive search model, predict the average fare on a route with the following characteristics: COUPON = 1.202, NEW = 3, VACATION = No, SW = No, HI = 4442.141, S_INCOME = $28,760, E_INCOME = $27,664, S_POP = 4,557,004, E_POP = 3,195,503, SLOT = Free, GATE = Free, PAX = 12,782, DISTANCE = 1976 miles.

AND

Question 8)

Predict the reduction in average fare on the route in question (7.), if Southwest decides to cover this route [using the exhaustive search model above].

```r
Exhaustive_pred_value_SW0 <- modelLr$coefficients["VACATIONYes"]*0+
                             modelLr$coefficients["SWYes"]*0+
                             modelLr$coefficients["HI"]*4442.141 +
                             modelLr$coefficients["E_INCOME"]*27664 +
                             modelLr$coefficients["S_POP"]*4557004 +
                             modelLr$coefficients["E_POP"]*3195503 +
                             modelLr$coefficients["DISTANCE"]*1976 +
                             modelLr$coefficients["PAX"]*12782 +
                             modelLr$coefficients["(Intercept)"]
print("Exhaustive_pred_value_SW0")
```

```
## [1] "Exhaustive_pred_value_SW0"
```

```r
print(Exhaustive_pred_value_SW0)
```

```
## VACATIONYes
##    257.5722
```

```r
# 257.5722

Exhaustive_pred_value_SW1 <- modelLr$coefficients["VACATIONYes"]*0+
                             modelLr$coefficients["SWYes"]*1+
                             modelLr$coefficients["HI"]*4442.141 +
```

```r
                              modelLr$coefficients["E_INCOME"]*27664 +
                              modelLr$coefficients["S_POP"]*4557004 +
                              modelLr$coefficients["E_POP"]*3195503 +
                              modelLr$coefficients["DISTANCE"]*1976 +
                              modelLr$coefficients["PAX"]*12782 +
                              modelLr$coefficients["(Intercept)"]
print("Exhaustive_pred_value_SW1")
```

```
## [1] "Exhaustive_pred_value_SW1"
```

```r
print(Exhaustive_pred_value_SW1)
```

```
## VACATIONYes
##    218.6155
```

```r
# 218.6155
avg_reduction_fare <- Exhaustive_pred_value_SW0-Exhaustive_pred_value_SW1
print("AVERAGE REDUCTION FARE")
```

```
## [1] "AVERAGE REDUCTION FARE"
```

```r
print(avg_reduction_fare)
```

```
## VACATIONYes
##    38.95665
```

```r
#38.95665
```

Answer 7 and 8)

We see that there is a reduction in Fare of $38.95 when Southwest airline is not serving versus when it is serving the route.

Question 9)

Using leaps package, run backward selection regression to reduce the number of predictors. Discuss the results from this model.

```r
modelLr = lm(FARE ~., data = Airfare.training)
bsearch <- step(modelLr, 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(bsearch)  # Which variables were dropped?
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = Airfare.training)
##
## 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)  42.0764345686  14.7566725244   2.851             0.004534 **
## VACATIONYes -38.7574569132   3.8500841929 -10.067 < 0.0000000000000002 ***
## SWYes       -40.5282166043   4.0337560764 -10.047 < 0.0000000000000002 ***
## HI            0.0082681499   0.0010423739   7.932   0.0000000000000143 ***
## E_INCOME      0.0014446281   0.0004089281   3.533             0.000450 ***
## S_POP         0.0000041850   0.0000007176   5.832   0.0000000098509604 ***
## E_POP         0.0000037791   0.0000007890   4.790   0.0000022053722984 ***
## SLOTFree    -16.8515659965   4.3045728245  -3.915             0.000103 ***
## GATEFree    -21.2165142735   4.3991611435  -4.823   0.0000018824635124 ***
## DISTANCE      0.0736714582   0.0028704349  25.666 < 0.0000000000000002 ***
## PAX          -0.0007619280   0.0001491869  -5.107   0.0000004660838631 ***
## ---
## 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: < 0.00000000000000022
```

```r
#coupon, new, s_income

# this backward selection model, with the lowest AIC, also dropped the same 3 variables - coupon, new, 
```

Answer 9)

Looking at the results, we can say the following:

p-value is quite low for all the columns, which means all the columns are significant enough.

By looking at the Ajusted R-squared, we see that the model explains 77.59% of the data.

By looking at the coefficients, we see that there is a negative linear relation between FARE and predictors- VACATIONYes, SWYes,SLOTFree, GATEFre and PAX. Rest of the predictors have a positive linear relation with FARE.

Question 10)

Now run a backward selection model using stepAIC() function. Discuss the results from this model, including the role of AIC in this model.

```r
#MASS package
modelLr = lm(FARE ~., data = Airfare.training)
b_stepaic_search <- stepAIC(modelLr, 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(b_stepaic_search)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = Airfare.training)
##
## 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)  42.0764345686  14.7566725244   2.851             0.004534 **
## VACATIONYes -38.7574569132   3.8500841929 -10.067 < 0.0000000000000002 ***
## SWYes       -40.5282166043   4.0337560764 -10.047 < 0.0000000000000002 ***
## HI            0.0082681499   0.0010423739   7.932   0.0000000000000143 ***
## E_INCOME      0.0014446281   0.0004089281   3.533             0.000450 ***
## S_POP         0.0000041850   0.0000007176   5.832   0.0000000098509604 ***
## E_POP         0.0000037791   0.0000007890   4.790   0.0000022053722984 ***
## SLOTFree    -16.8515659965   4.3045728245  -3.915             0.000103 ***
## GATEFree    -21.2165142735   4.3991611435  -4.823   0.0000018824635124 ***
## DISTANCE      0.0736714582   0.0028704349  25.666 < 0.0000000000000002 ***
## PAX          -0.0007619280   0.0001491869  -5.107   0.0000004660838631 ***
## ---
## 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: < 0.00000000000000022
```

```
b_stepaic_search$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
## Final Model:
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##
##         Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                               496   621821.3 3652.059
## 2   - COUPON  1  911.0487       497   622732.4 3650.805
## 3 - S_INCOME  1 1261.1907       498   623993.5 3649.837
## 4      - NEW  1 1696.6579       499   625690.2 3649.222
```

Answer 10)

If we compare the model above, bsearch, and this model, we see that both are exactly same with same value for AIC. We see that with every step, one insignificant variable is getting eliminated thus lowering the value of AIC. We finally compare the values of AIC and choose the model with the lowest AIC for the best fit.