Homework 2

Group BUAN635.501-1

2/22/2020

CLASS: "BUAN 6356"
GROUP MEMBERS: "Sai Raghavendra Sridhar(sxs180281), Shreya Tippannawar(sst190000), Smruti Viswanath Iyer(sxi180001), Piyush Dangwal(pxd142430),Shanshan Luo(sxl130330)"

Solutions :

```
if(!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(tidyverse, gplots, GGally, tinytex, data.table, reshape, knitr, leaps, pivottabler, fore
```

```
## Installing package into 'C:/Users/saira/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contri
##   cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/3.6/PACKAGES'
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\saira\AppData\Local\Temp\Rtmp6PjKeg\downloaded_packages
```

```
##
## tidyverse installed
```

```
## Warning in pacman::p_load(tidyverse, gplots, GGally, tinytex, data.table, : Failed to install/load:
## tidyverse
```

```
search()
```

```
##  [1] ".GlobalEnv"          "package:forecast"    "package:pivottabler"
##  [4] "package:leaps"       "package:knitr"       "package:reshape"
##  [7] "package:data.table"  "package:tinytex"     "package:GGally"
## [10] "package:ggplot2"     "package:gplots"      "package:pacman"
## [13] "package:stats"       "package:graphics"    "package:grDevices"
## [16] "package:utils"       "package:datasets"    "package:methods"
## [19] "Autoloads"           "package:base"
```

b. Read in the data from "Airfares":

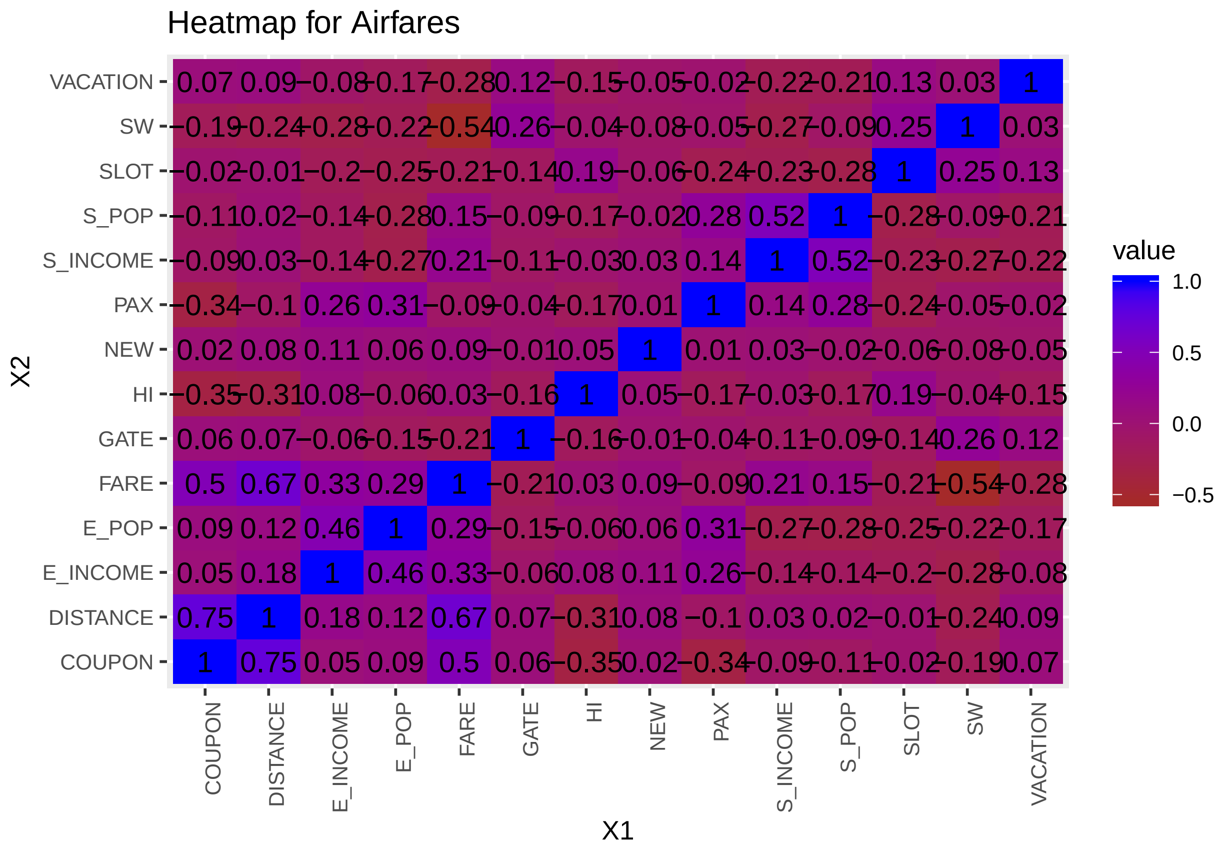
```
Airfares.dt <- read.csv("Airfares.csv")
Airfares.dt <- Airfares.dt[,-c(1:4)]
```

```
Airfares.dt$SW <- as.numeric(Airfares.dt$SW)
Airfares.dt$VACATION <- as.numeric(Airfares.dt$VACATION)
Airfares.dt$SLOT <- as.numeric(Airfares.dt$SLOT)
Airfares.dt$GATE <- as.numeric(Airfares.dt$GATE)


library(reshape)
correlation_matrix <- round(cor(Airfares.dt),2)
correlation_matrix[,14]
```

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

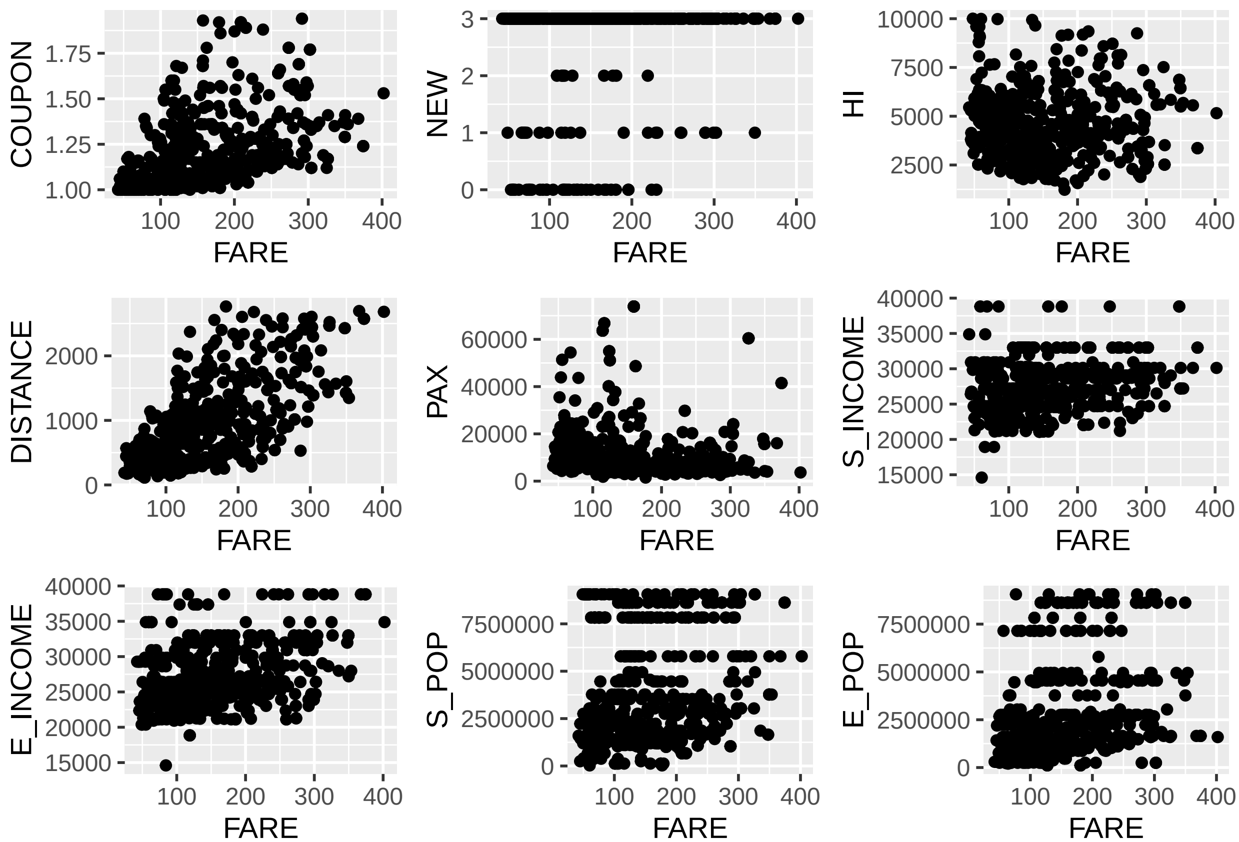
```
##   COUPON      NEW VACATION       SW       HI S_INCOME E_INCOME    S_POP
##     0.50     0.09    -0.28    -0.54     0.03     0.21     0.33     0.15
##    E_POP     SLOT     GATE DISTANCE      PAX     FARE
##     0.29    -0.21    -0.21     0.67    -0.09     1.00
```

```
melted_co_matrix <- melt(correlation_matrix)

  ggplot(melted_co_matrix,aes(x=X1,y=X2,fill = value))+
  scale_fill_gradient(low = "brown",high = "blue")+
  geom_tile()+
  geom_text(aes(x=X1,y=X2,label = value))+
  theme(text = element_text(size = 10), axis.text.x = element_text(angle = 90,hjust = 1))+
  ggtitle("Heatmap for Airfares")
```




```
library(ggplot2)
library(gridExtra)

x = ggplot(Airfares.dt)

coupon.plot <- x+
  geom_point(aes(x=FARE, y=COUPON))
new.plot <- x+
  geom_point(aes(x=FARE, y=NEW))
hi.plot <- x+
  geom_point(aes(x=FARE, y=HI))
distance.plot <- x+
  geom_point(aes(x=FARE, y=DISTANCE))
pax.plot <- x+
  geom_point(aes(x=FARE, y=PAX))
sincome.plot <- x+
  geom_point(aes(x=FARE, y=S_INCOME))
eincome.plot <- x+
  geom_point(aes(x=FARE, y=E_INCOME))
spop.plot <- x+
  geom_point(aes(x=FARE, y=S_POP))
epop.plot <- x+
  geom_point(aes(x=FARE, y=E_POP))
```

```
grid.arrange(coupon.plot,new.plot,hi.plot,distance.plot,
             pax.plot,sincome.plot,eincome.plot,spop.plot,epop.plot)
```



*Answer 1 - From the heatmap (correlation matrix) we can figure out that DISTANCE is the best predictor for FARE with high positive co-relation of 0.67. Also from the scatterplot we can that there is high positive correlation and strong positive relation which brings us to the conclusion that DISTANCE IS BEST PREDICTOR OF FARE.

####**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

```
percentage_sw = (nrow(subset(Airfares.dt, SW == 2))/ nrow(Airfares.dt))*100
percentage_sw_vector <- c(percentage_sw, (100-percentage_sw))
names(percentage_sw_vector) <- c("Yes","No")


percentage_vacation = (nrow(subset(Airfares.dt, VACATION == 2))/ nrow(Airfares.dt))*100
percentage_vacation_vector <- c(percentage_vacation, (100-percentage_vacation))
names(percentage_vacation_vector) <- c("Yes", "No")

percentage_slot = (nrow(subset(Airfares.dt, SLOT ==2))/ nrow(Airfares.dt))*100
percentage_slot_vector <- c(percentage_slot, (100-percentage_slot))
names(percentage_slot_vector) <- c("Free","Controlled")

percentage_gate = (nrow(subset(Airfares.dt, GATE ==2))/nrow(Airfares.dt))*100
percentage_gate_vector <- c(percentage_gate, (100-percentage_gate))
```

```r
names(percentage_gate_vector) <- c("Free", "Constrained")


perc.df <- data.frame(percentage_sw_vector, percentage_vacation_vector,percentage_slot_vector, percentag

perc.df
```

```
##     percentage_sw_vector percentage_vacation_vector percentage_slot_vector
## Yes             30.40752                   26.64577               71.47335
## No              69.59248                   73.35423               28.52665
##     percentage_gate_vector
## Yes               80.56426
## No                19.43574
```

```r
# Index : For percentage_sw_vec : Yes: SW serves the route
#       : For percentage_vac_vec :Yes: A vacation route
#       : For pecentage_slot_vec : Yes: end-point airport is free
#       : for pecentage_gate_vec : Yes: end point airport do not have gate constraints


category_analysis <- function(category_value) {
form <- as.formula(paste("Airfares.dt$FARE ~ Airfares.dt$", category_value))
print(aggregate(form, data <- Airfares.dt, FUN <- mean))
}
category_variables <- c("VACATION", "SW", "SLOT", "GATE")
for (var in category_variables){
category_analysis(var)
cat('\n')
}
```

```
##   Airfares.dt$VACATION Airfares.dt$FARE
## 1                    1         173.5525
## 2                    2         125.9809
##
##   Airfares.dt$SW Airfares.dt$FARE
## 1              1        188.18279
## 2              2         98.38227
##
##   Airfares.dt$SLOT Airfares.dt$FARE
## 1                1         186.0594
## 2                2         150.8257
##
##   Airfares.dt$GATE Airfares.dt$FARE
## 1                1          193.129
## 2                2          153.096
```

```r
#Index : VACATION : 1 = 'No',           2 = 'Yes'
#        SW       : 1 = 'No',           2 = 'Yes'
#        SLOT     : 1 = 'Controlled',   2 = 'Free'
#        GATE     : 1 = 'Constrained',  2 = 'Free'
```

#**Answer 2 - SW is the best categorical predictor as there is significant drop in average when it is being included

####**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

```
set.seed(42)
rows  <- sample(nrow(Airfares.dt))
Airfares.dt <- Airfares.dt[rows, ]


split <- round(nrow(Airfares.dt) * 0.8)
train.df <- Airfares.dt[1:split, ]
test.df <- Airfares.dt[(split+1):nrow(Airfares.dt),]
```

#**Answer 3 - Rouding off 80% of training data and rest 20% data is done

####**Question 4 Using leaps package, run stepwise regression to reduce the number of predictors. Discuss the results from this model.

```
Airfares.lm <- lm(FARE ~ ., data= train.df)
options(scipen =999)
summary(Airfares.lm)
```

```
##
## Call:
## lm(formula = FARE ~ ., data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.282 -23.384  -2.476  22.156 106.501
##
## Coefficients:
##                   Estimate     Std. Error t value             Pr(>|t|)
## (Intercept) 128.2690818640  36.3202234544   3.532             0.000452 ***
## COUPON       11.6744988371  13.6949175687   0.852             0.394365
## NEW          -2.2468005921   2.0827213457  -1.079             0.281210
## VACATION    -37.8385127965   3.9788129464  -9.510 < 0.0000000000000002 ***
## SW          -38.9566477546   4.2526101838  -9.161 < 0.0000000000000002 ***
## HI            0.0085414832   0.0010936608   7.810   0.0000000000000343 ***
## S_INCOME      0.0006160967   0.0005709965   1.079             0.281119
## E_INCOME      0.0015472928   0.0004141497   3.736             0.000209 ***
## S_POP         0.0000040087   0.0000007411   5.409   0.0000000987167149 ***
## E_POP         0.0000039572   0.0000008329   4.751   0.0000026562530825 ***
## SLOT        -16.4322948237   4.3647846605  -3.765             0.000187 ***
## GATE        -21.1634823059   4.4093579183  -4.800   0.0000021065804690 ***
## DISTANCE      0.0715673994   0.0039223121  18.246 < 0.0000000000000002 ***
## PAX          -0.0007340587   0.0001662490  -4.415   0.0000123830100844 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.41 on 496 degrees of freedom
## Multiple R-squared:  0.7817, Adjusted R-squared:  0.7759
## F-statistic: 136.6 on 13 and 496 DF,  p-value: < 0.00000000000000022
```

```r
Airfares.lm.stepwise <- step(Airfares.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(Airfares.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) 159.4301883561  20.7284827817   7.691   0.0000000000000782 ***
## VACATION    -38.7574569132   3.8500841929 -10.067 < 0.0000000000000002 ***
## SW          -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 ***
## SLOT        -16.8515659965   4.3045728245  -3.915             0.000103 ***
## GATE        -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: < 0.0000000000000022
```

#**Answer 4 -Using the stepwise regression, the number of variables has been reduced to 10 from 13. We can see that AIC has been decreasing in the subsequent steps and least observed value is 3649.22 when COUPON, NEW, S_INCOME are removed from the model.

####**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.

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

# show models

sum$which
```

```
##    (Intercept) COUPON   NEW VACATION    SW    HI S_INCOME E_INCOME S_POP E_POP
## 1         TRUE  FALSE FALSE    FALSE FALSE FALSE    FALSE    FALSE FALSE FALSE
## 2         TRUE  FALSE FALSE    FALSE  TRUE FALSE    FALSE    FALSE FALSE FALSE
## 3         TRUE  FALSE FALSE     TRUE  TRUE FALSE    FALSE    FALSE FALSE FALSE
## 4         TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE    FALSE FALSE FALSE
## 5         TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE    FALSE FALSE FALSE
## 6         TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE    FALSE FALSE FALSE
## 7         TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE    FALSE  TRUE  TRUE
## 8         TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE     TRUE  TRUE  TRUE
## 9         TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE    FALSE  TRUE  TRUE
## 10        TRUE  FALSE FALSE     TRUE  TRUE  TRUE    FALSE     TRUE  TRUE  TRUE
## 11        TRUE  FALSE  TRUE     TRUE  TRUE  TRUE    FALSE     TRUE  TRUE  TRUE
## 12        TRUE  FALSE  TRUE     TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE
## 13        TRUE   TRUE  TRUE     TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE
##     SLOT  GATE DISTANCE   PAX
## 1  FALSE FALSE     TRUE FALSE
## 2  FALSE FALSE     TRUE FALSE
## 3  FALSE FALSE     TRUE FALSE
## 4  FALSE FALSE     TRUE FALSE
## 5   TRUE FALSE     TRUE FALSE
## 6   TRUE  TRUE     TRUE FALSE
## 7  FALSE FALSE     TRUE  TRUE
## 8  FALSE FALSE     TRUE  TRUE
## 9   TRUE  TRUE     TRUE  TRUE
## 10  TRUE  TRUE     TRUE  TRUE
## 11  TRUE  TRUE     TRUE  TRUE
## 12  TRUE  TRUE     TRUE  TRUE
## 13  TRUE  TRUE     TRUE  TRUE
```

```r
# show metrics
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
```

##**Answer 5: In this adjusted R-sq has highest value for 12th susbset combination and Cp has the optimal value of 11.086. We use 10 variable reduction combination as we tend to reduce the number of variables. Hence we use Cp to finalize the subset.The combination shows COUPON, NEW, S_INCOME will not be considered for the model.The same number of models are eliminated both here and also in the stepwise model. Hence both model corresponds similarly.

####**Question 6 Compare the predictive accuracy of both models—stepwise regression and exhaustive search—using measures such as RMSE.

```r
##Stepwise

Airfares.lm.stepwise.predict <- predict(Airfares.lm.stepwise, test.df)
accuracy(Airfares.lm.stepwise.predict, test.df$FARE)
```

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

```r
#Accuracy Exhaustive

ex.lm <- lm(FARE ~VACATION+ SW+ HI+ E_INCOME+ S_POP+ E_POP+ SLOT+ GATE+ DISTANCE+ PAX,
            data = train.df[])
fares.lm.exhaustive.predict <- predict(ex.lm, test.df[,-c(1,2,6)])

accuracy(fares.lm.exhaustive.predict, test.df$FARE)
```

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

#**Answer 6: As both model tend to use same variables they produce same type of error. Hence based on the accuracy, the RMSE value are same for both the models.

####**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.

```r
without_sw <- predict(ex.lm, data.frame(VACATION = 1, SW =
1, HI = 4442.141, E_INCOME = 27664, S_POP =
4557004, E_POP = 3195503, SLOT = 2, GATE = 2, PAX = 12782,
DISTANCE = 1976))

without_sw
```

```
##       1
## 247.684
```

#**Answer 7: The answer(fare) was found out to be 247.684 upon prediction

####**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
with_sw <- predict(ex.lm, data.frame(VACATION = 1, SW =
2, HI = 4442.141, E_INCOME = 27664, S_POP =
4557004, E_POP = 3195503, SLOT = 2, GATE = 2, PAX = 12782,
DISTANCE = 1976))


avg_fare <- c(without_sw,with_sw, (without_sw-with_sw))
names(avg_fare) <-c("W/O SW","With SW", "FARE Difference")
avg_fare
```

```
##          W/O SW         With SW FARE Difference
##       247.68398       207.15577        40.52822
```

#**Answer 8: The answer was found out to be 207.155 with Southwest and the difference was found to be around 40.52

####**Question 9 Using leaps package, run backward selection regression to reduce the number of predictors. Discuss the results from this model.

```r
airfare.backward.lm <- step(Airfares.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(airfare.backward.lm)
```

```
##
## 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) 159.4301883561  20.7284827817   7.691   0.0000000000000782 ***
## VACATION    -38.7574569132   3.8500841929 -10.067 < 0.0000000000000002 ***
## SW          -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 ***
## SLOT        -16.8515659965   4.3045728245  -3.915             0.000103 ***
## GATE        -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
```

#**Answer 9 - On running backward regression we found out that the least acheived AIC was 3649.22 when we remove the variables COUPON, S_Income, NEW that is variables are now 10 from 13. The F-statistic was found out to be 177.2 which has significantly less p-value, predicts model holds good.

####**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
library(MASS)
airfares.lm.bselect <- stepAIC(Airfares.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(airfares.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) 159.4301883561  20.7284827817   7.691   0.0000000000000782 ***
## VACATION    -38.7574569132   3.8500841929 -10.067 < 0.0000000000000002 ***
## SW          -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 ***
## SLOT        -16.8515659965   4.3045728245  -3.915             0.000103 ***
## GATE        -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
```

#**Answer 10 - In this STEPAIC model, we remove variables based on the contributions to AIC. In the first iteration, second iteration, third iteration the variables COUPON, S_INCOME, NEW were removed respectively based on the AIC. Here we have contribution through S_INCOME, The 4th iteration seems to have the lowest AIC and hence we stopped there. The Optimal model gets created then.