Homework2_BUAN6356503_Group10

("Chitresh Kumar","Chaitanya Narella","Disha Punjabi","Nidaa Tamkeen","Mai Han Tran")

R Markdown

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.

```
airfare.df <- read.csv("Airfares.csv")
air.df <- airfare.df[,-c(1,2,3,4)]
data_airfares <- air.df[-c(3,4,10,11)]
air.dt <- setDT(air.df)
```

Including Plots

```
cor.mat <- round(cor(air.dt[,!c("S_CODE","S_CITY","E_CODE","E_CITY","VACATION","SW","SLOT","GATE")]),2)
```

```
## Warning in `[.data.table`(air.dt, , !c("S_CODE", "S_CITY",
## "E_CODE", "E_CITY", : column(s) not removed because not found:
## S_CODE,S_CITY,E_CODE,E_CITY
```

```
cor.mat
```

```
##          COUPON   NEW    HI S_INCOME E_INCOME S_POP E_POP DISTANCE   PAX
## COUPON     1.00  0.02 -0.35    -0.09     0.05 -0.11  0.09     0.75 -0.34
## NEW        0.02  1.00  0.05     0.03     0.11 -0.02  0.06     0.08  0.01
## HI        -0.35  0.05  1.00    -0.03     0.08 -0.17 -0.06    -0.31 -0.17
## S_INCOME  -0.09  0.03 -0.03     1.00    -0.14  0.52 -0.27     0.03  0.14
## E_INCOME   0.05  0.11  0.08    -0.14     1.00 -0.14  0.46     0.18  0.26
## S_POP     -0.11 -0.02 -0.17     0.52    -0.14  1.00 -0.28     0.02  0.28
## E_POP      0.09  0.06 -0.06    -0.27     0.46 -0.28  1.00     0.12  0.31
## DISTANCE   0.75  0.08 -0.31     0.03     0.18  0.02  0.12     1.00 -0.10
## PAX       -0.34  0.01 -0.17     0.14     0.26  0.28  0.31    -0.10  1.00
## FARE       0.50  0.09  0.03     0.21     0.33  0.15  0.29     0.67 -0.09
##           FARE
## COUPON    0.50
## NEW       0.09
## HI        0.03
## S_INCOME  0.21
## E_INCOME  0.33
## S_POP     0.15
## E_POP     0.29
## DISTANCE  0.67
## PAX      -0.09
## FARE      1.00
```

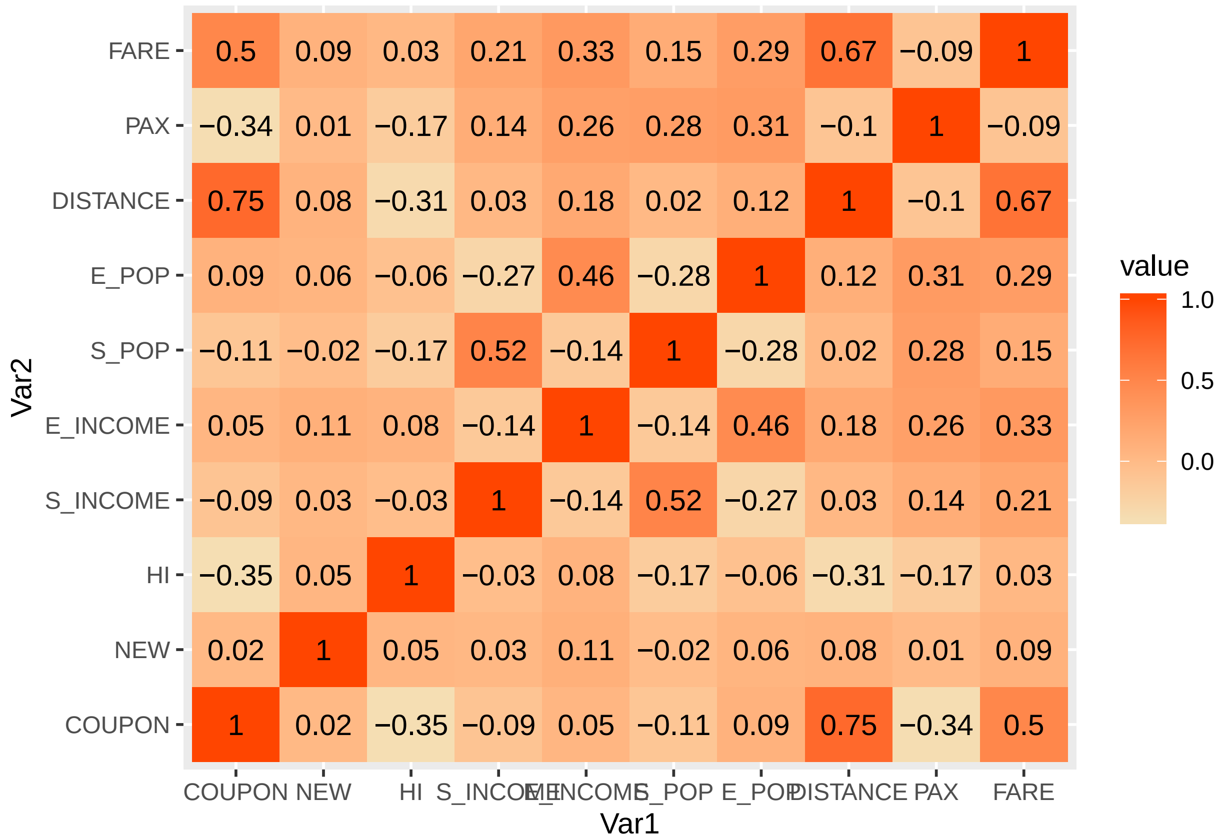
```
melted.cor.mat <- melt(cor.mat)
melted.cor.mat
```

```
##        Var1     Var2 value
## 1    COUPON   COUPON  1.00
## 2       NEW   COUPON  0.02
## 3        HI   COUPON -0.35
```

```
## 4   S_INCOME   COUPON -0.09
## 5   E_INCOME   COUPON  0.05
## 6      S_POP   COUPON -0.11
## 7      E_POP   COUPON  0.09
## 8   DISTANCE   COUPON  0.75
## 9        PAX   COUPON -0.34
## 10      FARE   COUPON  0.50
## 11    COUPON      NEW  0.02
## 12       NEW      NEW  1.00
## 13        HI      NEW  0.05
## 14  S_INCOME      NEW  0.03
## 15  E_INCOME      NEW  0.11
## 16     S_POP      NEW -0.02
## 17     E_POP      NEW  0.06
## 18  DISTANCE      NEW  0.08
## 19       PAX      NEW  0.01
## 20      FARE      NEW  0.09
## 21    COUPON       HI -0.35
## 22       NEW       HI  0.05
## 23        HI       HI  1.00
## 24  S_INCOME       HI -0.03
## 25  E_INCOME       HI  0.08
## 26     S_POP       HI -0.17
## 27     E_POP       HI -0.06
## 28  DISTANCE       HI -0.31
## 29       PAX       HI -0.17
## 30      FARE       HI  0.03
## 31    COUPON S_INCOME -0.09
## 32       NEW S_INCOME  0.03
## 33        HI S_INCOME -0.03
## 34  S_INCOME S_INCOME  1.00
## 35  E_INCOME S_INCOME -0.14
## 36     S_POP S_INCOME  0.52
## 37     E_POP S_INCOME -0.27
## 38  DISTANCE S_INCOME  0.03
## 39       PAX S_INCOME  0.14
## 40      FARE S_INCOME  0.21
## 41    COUPON E_INCOME  0.05
## 42       NEW E_INCOME  0.11
## 43        HI E_INCOME  0.08
## 44  S_INCOME E_INCOME -0.14
## 45  E_INCOME E_INCOME  1.00
## 46     S_POP E_INCOME -0.14
## 47     E_POP E_INCOME  0.46
## 48  DISTANCE E_INCOME  0.18
## 49       PAX E_INCOME  0.26
## 50      FARE E_INCOME  0.33
## 51    COUPON    S_POP -0.11
## 52       NEW    S_POP -0.02
## 53        HI    S_POP -0.17
## 54  S_INCOME    S_POP  0.52
## 55  E_INCOME    S_POP -0.14
## 56     S_POP    S_POP  1.00
## 57     E_POP    S_POP -0.28
```

```
## 58  DISTANCE    S_POP  0.02
## 59       PAX    S_POP  0.28
## 60      FARE    S_POP  0.15
## 61    COUPON    E_POP  0.09
## 62       NEW    E_POP  0.06
## 63        HI    E_POP -0.06
## 64  S_INCOME    E_POP -0.27
## 65  E_INCOME    E_POP  0.46
## 66     S_POP    E_POP -0.28
## 67     E_POP    E_POP  1.00
## 68  DISTANCE    E_POP  0.12
## 69       PAX    E_POP  0.31
## 70      FARE    E_POP  0.29
## 71    COUPON DISTANCE  0.75
## 72       NEW DISTANCE  0.08
## 73        HI DISTANCE -0.31
## 74  S_INCOME DISTANCE  0.03
## 75  E_INCOME DISTANCE  0.18
## 76     S_POP DISTANCE  0.02
## 77     E_POP DISTANCE  0.12
## 78  DISTANCE DISTANCE  1.00
## 79       PAX DISTANCE -0.10
## 80      FARE DISTANCE  0.67
## 81    COUPON      PAX -0.34
## 82       NEW      PAX  0.01
## 83        HI      PAX -0.17
## 84  S_INCOME      PAX  0.14
## 85  E_INCOME      PAX  0.26
## 86     S_POP      PAX  0.28
## 87     E_POP      PAX  0.31
## 88  DISTANCE      PAX -0.10
## 89       PAX      PAX  1.00
## 90      FARE      PAX -0.09
## 91    COUPON     FARE  0.50
## 92       NEW     FARE  0.09
## 93        HI     FARE  0.03
## 94  S_INCOME     FARE  0.21
## 95  E_INCOME     FARE  0.33
## 96     S_POP     FARE  0.15
## 97     E_POP     FARE  0.29
## 98  DISTANCE     FARE  0.67
## 99       PAX     FARE -0.09
## 100     FARE     FARE  1.00
```

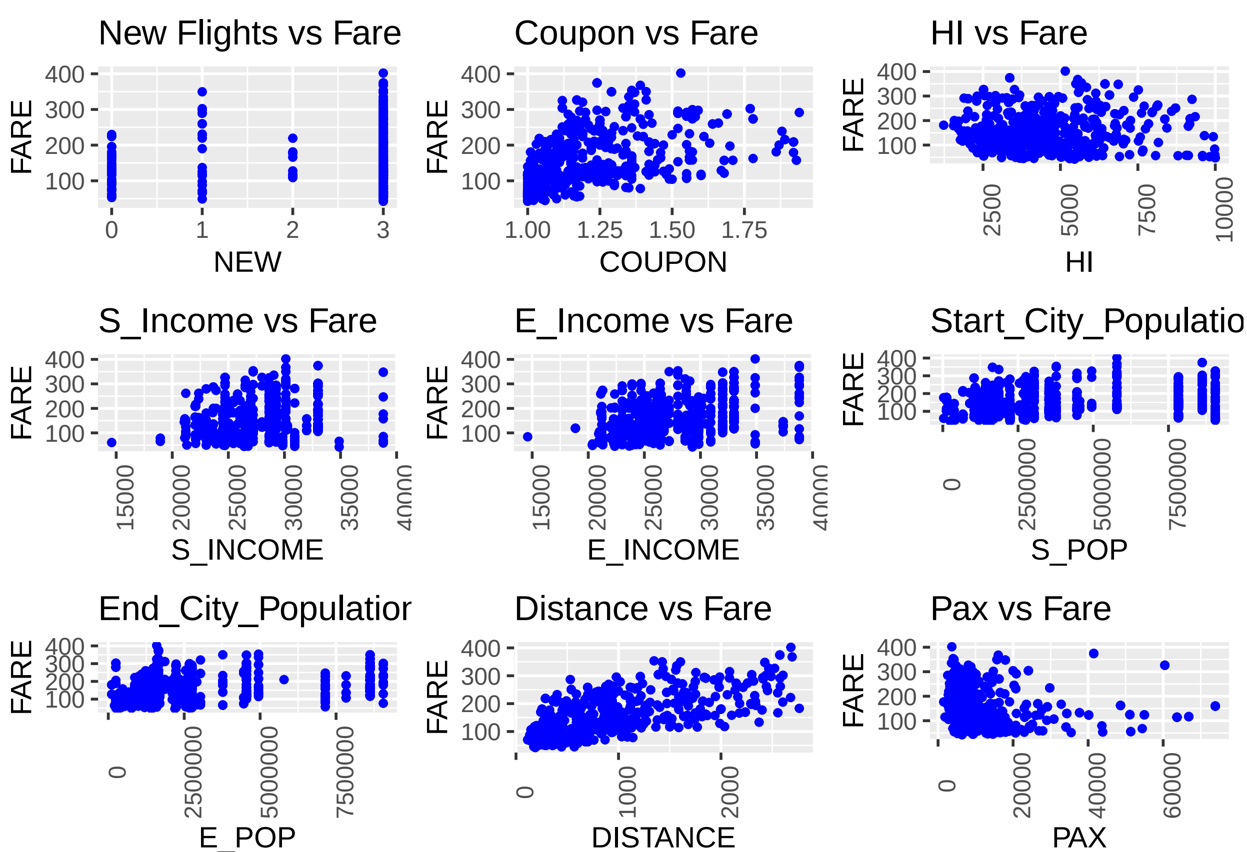
```r
ggplot(melted.cor.mat, aes(x = Var1, y = Var2, fill = value)) +
  scale_fill_gradient(low="wheat", high="orangered") +
  geom_tile() +
  geom_text(aes(x = Var1, y = Var2, label = value))
```




```
Coupon and distance have strong positive correlation .As distance increases coupon increases.
Fare and distance has positive correlation .As distance increases the fare will increase
Fare and coupon has positive correlation .As coupon increases the fare will increase
```

```r
plot_1 <- ggplot(data_airfares ) +
geom_point(aes(x= NEW, y = FARE ), size = 1,colour="blue") + ggtitle("New Flights vs Fare")

plot_2 <- ggplot(data_airfares )+
geom_point(aes(x= COUPON, y = FARE ), size = 1,colour="blue") + ggtitle("Coupon vs Fare")


plot_3 <- ggplot(data_airfares )+
geom_point(aes(x= HI, y = FARE ), size = 1,colour="blue")+ ggtitle("HI vs Fare")+
theme(axis.text.x = element_text(angle = 90))

plot_4 <- ggplot(data_airfares )+
geom_point(aes(x= S_INCOME, y = FARE ), size = 1,colour="blue")+ ggtitle("S_Income vs Fare")+
theme(axis.text.x = element_text(angle = 90))

plot_5 <- ggplot(data_airfares )+
geom_point(aes(x= E_INCOME, y = FARE ), size = 1,colour="blue")+ ggtitle("E_Income vs Fare")+
theme(axis.text.x = element_text(angle = 90))

plot_6 <- ggplot(data_airfares )+
geom_point(aes(x= S_POP, y = FARE ), size = 1,colour="blue")+ ggtitle("Start_City_Population vs Fare")+
theme(axis.text.x = element_text(angle = 90))
```

```r
plot_7 <- ggplot(data_airfares )+
geom_point(aes(x= E_POP, y = FARE ), size = 1,colour="blue")+ ggtitle("End_City_Population vs Fare")+
theme(axis.text.x = element_text(angle = 90))

plot_8 <- ggplot(data_airfares )+
geom_point(aes(x= DISTANCE, y = FARE ), size = 1,colour="blue")+ggtitle("Distance vs Fare")+
theme(axis.text.x = element_text(angle = 90))

plot_9 <- ggplot(data_airfares )+
geom_point(aes(x= PAX, y = FARE ), size = 1,colour="blue")+ ggtitle("Pax vs Fare")+
theme(axis.text.x = element_text(angle = 90))

grid.arrange(plot_1, plot_2, plot_3, plot_4, plot_5, plot_6, plot_7, plot_8,
 plot_9, nrow = 3)
```



Evidently from the scatterplot and data, Distance is the best single predictor of Fare. They both are highly correlated as compared to the other predictors. The scatterplot reflects strong positive correlation between Distance and 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.

```r
#View the pivot table of Vacation
Vacation <- air.df %>%
dplyr::select(VACATION,FARE) %>%
group_by(VACATION) %>%
```

```
summarise(Count = length(VACATION),Total = nrow(air.df),
Percent = (length(VACATION)/nrow(air.df)) *100 ,
AvgFare = mean(FARE))

Vacation
```

```
## # A tibble: 2 x 5
##   VACATION Count Total Percent AvgFare
##   <fct>    <int> <int>   <dbl>   <dbl>
## 1 No         468   638    73.4    174.
## 2 Yes        170   638    26.6    126.
```

```
#View the pivot table of SouthWest
Southwest <- air.df %>%
dplyr::select(SW,FARE) %>%
group_by(SW) %>%
summarise(Count = length(SW),Total = nrow(air.df),
Percent = (length(SW)/nrow(air.df))* 100,
AvgFare = mean(FARE))
Southwest
```

```
## # A tibble: 2 x 5
##   SW    Count Total Percent AvgFare
##   <fct> <int> <int>   <dbl>   <dbl>
## 1 No      444   638    69.6   188.
## 2 Yes     194   638    30.4    98.4
```

```
#View the pivot table of Gate
Gate <- air.df %>%
dplyr::select(GATE,FARE) %>%
group_by(GATE) %>%
summarise(Count = length(GATE),Total = nrow(air.df),
Percent = (length(GATE)/nrow(air.df))*100,
AvgFare = mean(FARE))
Gate
```

```
## # A tibble: 2 x 5
##   GATE        Count Total Percent AvgFare
##   <fct>       <int> <int>   <dbl>   <dbl>
## 1 Constrained   124   638    19.4    193.
## 2 Free          514   638    80.6    153.
```

```
Slot <- air.df %>%
dplyr::select(SLOT,FARE) %>%
group_by(SLOT) %>%
summarise(Count = length(SLOT),Total = nrow(air.df),
Percent = (length(SLOT)/nrow(air.df))*100,
AvgFare = mean(FARE))
Slot
```

```
## # A tibble: 2 x 5
##   SLOT       Count Total Percent AvgFare
##   <fct>      <int> <int>   <dbl>   <dbl>
## 1 Controlled   182   638    28.5    186.
## 2 Free         456   638    71.5    151.
```

From the above scenario, Southwest Airline is a highly impacting categorical predictor. It strikingly a

Question 3) Create data partition by assigning 80% of the records to the training dataset. Use rounding

Linear Regression Model

```r
a <- nrow(air.df)
b <- a*0.80;
round(b,digits=0)
```

```
## [1] 510
```

```r
set.seed(42)
train.index <- sample(c(1:510), 128)
train.df <- air.df[-train.index, ]
valid.df <- air.df[+train.index, ]

air.lm <- lm(FARE~ ., data = train.df)

options(scipen = 999)
summary(air.lm)
```

```
##
## Call:
## lm(formula = FARE ~ ., data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.085 -21.491  -0.404  19.888 129.369
##
## Coefficients:
##                  Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)  30.3026738819 29.0660922651   1.043              0.29767
## COUPON        8.9253306190 12.7880756505   0.698              0.48554
## NEW          -3.3140517563  2.0290805650  -1.633              0.10305
## VACATIONYes -37.5720456501  3.8217563451  -9.831 < 0.0000000000000002 ***
## SWYes       -44.8363251714  4.0199772905 -11.153 < 0.0000000000000002 ***
## HI            0.0086378591  0.0010827115   7.978   0.0000000000000104 ***
## S_INCOME      0.0008438433  0.0005475265   1.541              0.12391
## E_INCOME      0.0012928227  0.0004146025   3.118              0.00193 **
## S_POP         0.0000028516  0.0000007080   4.028   0.0000650919218052 ***
## E_POP         0.0000036134  0.0000008188   4.413   0.0000125212716976 ***
## SLOTFree    -17.0097926786  4.1334603128  -4.115   0.0000453066018437 ***
## GATEFree    -22.5297451287  4.2853179691  -5.257   0.0000002175134087 ***
## DISTANCE      0.0723565081  0.0037586354  19.251 < 0.0000000000000002 ***
## PAX          -0.0007641725  0.0001531347  -4.990   0.0000008361426766 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.81 on 496 degrees of freedom
## Multiple R-squared:  0.8001, Adjusted R-squared:  0.7949
## F-statistic: 152.7 on 13 and 496 DF,  p-value: < 0.00000000000000022
```

```r
class(air.lm)
```

```
## [1] "lm"
```

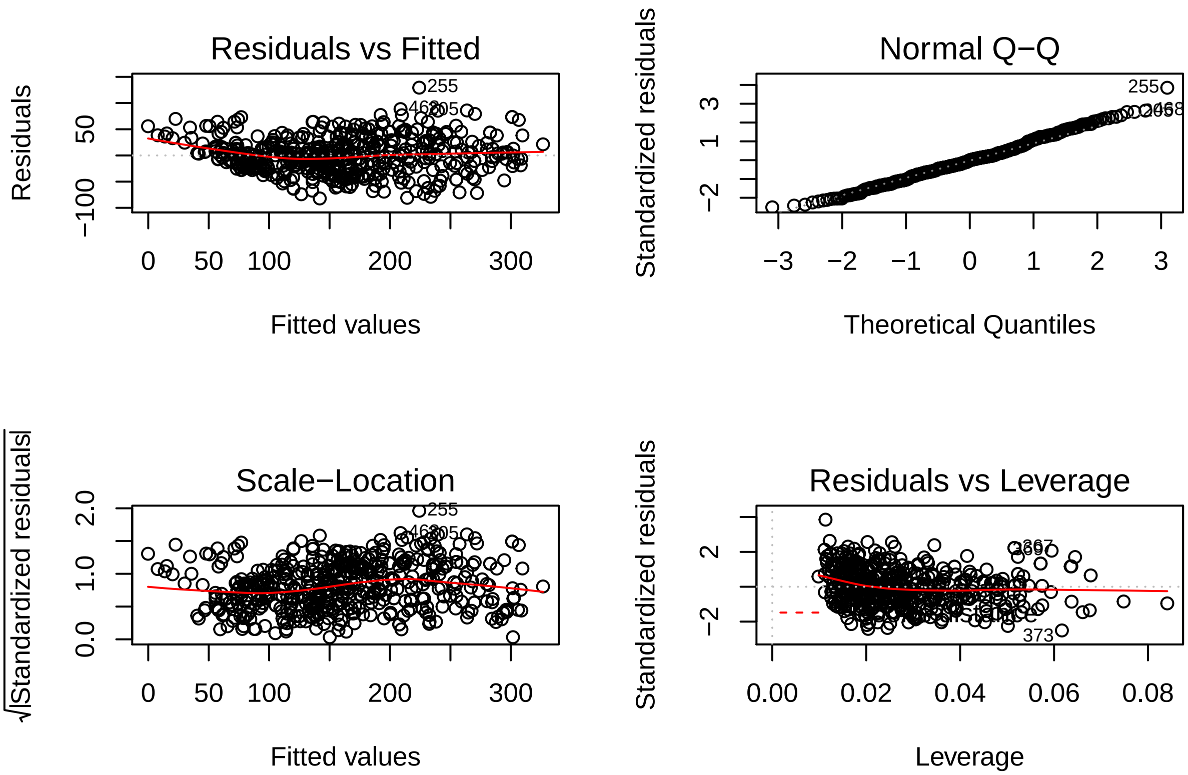
```r
methods(class=class(air.lm))
```

```
##  [1] add1           alias          anova          augment       
##  [5] case.names     coerce         confint        cooks.distance
##  [9] cull_for_do    deviance       dfbeta         dfbetas       
## [13] drop1          dummy.coef     effects        extractAIC    
## [17] family         forecast       formula        fortify       
## [21] getResponse    glance         hatvalues      influence     
## [25] initialize     kappa          labels         logLik        
## [29] makeFun        merge          model.frame    model.matrix  
## [33] mplot          msummary       nobs           plot          
## [37] predict        print          proj           qqnorm        
## [41] qr             residuals      rstandard      rstudent      
## [45] sample         show           simulate       slotsFromS3   
## [49] summary        tidy           TukeyHSD       variable.names
## [53] varImp         vars           vcov           waldtest      
## see '?methods' for accessing help and source code
```

```r
confint(air.lm)
```

```
##                        2.5 %          97.5 %
## (Intercept) -26.805171657365  87.410519421112
## COUPON      -16.200146941924  34.050808179856
## NEW          -7.300704603651   0.672601091079
## VACATIONYes -45.080873085706 -30.063218214473
## SWYes       -52.734608851306 -36.938041491465
## HI            0.006510592729   0.010765125448
## S_INCOME     -0.000231913978   0.001919600620
## E_INCOME      0.000478229024   0.002107416448
## S_POP         0.000001460647   0.000004242605
## E_POP         0.000002004628   0.000005222232
## SLOTFree    -25.131043066494  -8.888542290607
## GATEFree    -30.949359104861 -14.110131152628
## DISTANCE      0.064971698154   0.079741318135
## PAX          -0.001065045245  -0.000463299747
```

```r
par(mfrow = c(2,2))
plot(air.lm)
```




```
par(mfrow = c(10,10))

air.lm.pred <-predict(air.lm,valid.df)

some.residuals <- valid.df$FARE[1:128] - air.lm.pred[1:128]


data.frame("Predicted" = air.lm.pred[1:128], "Actual" = valid.df$FARE[1:128],
           "Residual" = some.residuals)
```

```
##      Predicted Actual    Residual
## 1  167.9497032 180.56  12.6102968
## 2  150.2509467  79.23 -71.0209467
## 3  103.4818121 123.97  20.4881879
## 4  178.3416767 115.84 -62.5016767
## 5  182.3795666 244.50  62.1204334
## 6  109.2825493 116.78   7.4974507
## 7  180.7403145 143.62 -37.1203145
## 8  132.5792134 105.73 -26.8492134
## 9  211.2717703 142.83 -68.4417703
## 10  50.7589133  97.36  46.6010867
## 11  74.3250745 121.67  47.3449255
## 12  99.8450554 106.77   6.9249446
## 13 159.9740035 215.06  55.0859965
## 14 271.4543588 273.12   1.6656412
## 15 140.3842626 121.09 -19.2942626
```

```
## 16  236.5568387 233.78   -2.7768387
## 17  244.4831179 349.97  105.4868821
## 18  189.1182080 157.50  -31.6182080
## 19  212.0397415 125.90  -86.1397415
## 20  169.6504087 169.90    0.2495913
## 21  186.6602014 169.90  -16.7602014
## 22  114.8859321  96.58  -18.3059321
## 23   77.9985919  96.18   18.1814081
## 24  323.0367555 402.02   78.9832445
## 25  139.4395955 124.92  -14.5195955
## 26  157.8413929 218.54   60.6986071
## 27  223.5919385 297.83   74.2380615
## 28   73.2373558  67.10   -6.1373558
## 29  104.8836266  85.47  -19.4136266
## 30  134.5129994 120.70  -13.8129994
## 31   36.0851128  42.47    6.3848872
## 32  137.3410688 133.04   -4.3010688
## 33  110.5936903  81.32  -29.2736903
## 34  173.4538906 143.20  -30.2538906
## 35   71.7066223 121.35   49.6433777
## 36  113.5408632  67.77  -45.7708632
## 37  250.9979555 294.18   43.1820445
## 38   -0.1908839  55.57   55.7608839
## 39   96.5310705  67.77  -28.7610705
## 40  139.8120969 123.44  -16.3720969
## 41  153.6602495 154.73    1.0697505
## 42  191.8095784  83.74 -108.0695784
## 43  175.2657336 116.18  -59.0857336
## 44  155.6836462 150.13   -5.5536462
## 45  150.0532211 125.09  -24.9632211
## 46  104.2990415  92.57  -11.7290415
## 47  196.6507172 116.52  -80.1307172
## 48   43.9340856  72.58   28.6459144
## 49  169.3559208 133.50  -35.8559208
## 50  121.8934192  85.47  -36.4234192
## 51  251.6641546 287.23   35.5658454
## 52  135.7533244 185.65   49.8966756
## 53  218.2971224 195.91  -22.3871224
## 54   69.6799669  84.53   14.8500331
## 55  152.7631171 185.65   32.8868829
## 56  253.4248722 304.18   50.7551278
## 57  294.0497325 326.76   32.7102675
## 58   70.3580057  65.31   -5.0480057
## 59  177.6819800 144.60  -33.0819800
## 60  123.4870887  88.46  -35.0270887
## 61  225.7405758 157.45  -68.2905758
## 62  198.7484349 134.09  -64.6584349
## 63  197.3498868 207.76   10.4101132
## 64  252.6541924 279.61   26.9558076
## 65  148.1402970 154.73    6.5897030
## 66  167.9339382 143.20  -24.7339382
## 67  276.9743113 273.12   -3.8543113
## 68  152.4811999 146.36   -6.1211999
## 69   83.4877966  75.71   -7.7777966
```

```
## 70  227.3331244 250.73   23.3968756
## 71  190.9218155 241.04   50.1181845
## 72  151.3611278 118.95  -32.4111278
## 73   64.6251493 118.17   53.5448507
## 74  168.1822390 240.88   72.6977610
## 75   83.6709331  63.39  -20.2809331
## 76  107.1281431  70.41  -36.7181431
## 77  237.9277943 335.55   97.6222057
## 78   91.6885136 112.99   21.3014864
## 79  186.2484069 210.90   24.6515931
## 80  269.1678750 349.53   80.3621250
## 81  131.7462544  97.96  -33.7862544
## 82  171.1542440 162.28   -8.8742440
## 83  153.8834330 117.23  -36.6534330
## 84  271.4543588 273.12    1.6656412
## 85  260.3699397 208.79  -51.5799397
## 86  259.2558001 270.36   11.1041999
## 87  185.2424132 138.56  -46.6824132
## 88  133.7337911 109.78  -23.9537911
## 89   65.8877449  63.92   -1.9677449
## 90  212.2802385 110.42 -101.8602385
## 91  106.1949676 109.44    3.2450324
## 92  169.6504087 169.90    0.2495913
## 93  156.1732136 193.67   37.4967864
## 94  134.8706131 127.38   -7.4906131
## 95  155.8622501  81.28  -74.5822501
## 96  256.4974690 291.66   35.1625310
## 97   66.7676211  66.14   -0.6276211
## 98   76.4506969  52.92  -23.5306969
## 99   75.0000505  57.62  -17.3800505
## 100 166.2961748 180.85   14.5538252
## 101 321.9867492 367.72   45.7332508
## 102 197.1350484 168.96  -28.1750484
## 103  93.7140691 153.95   60.2359309
## 104 207.0961904 223.99   16.8938096
## 105 164.3966298 168.92    4.5233702
## 106  71.2649710  69.10   -2.1649710
## 107 212.0525210 200.09  -11.9625210
## 108 209.4845188 208.71   -0.7745188
## 109 179.9868215 215.01   35.0231785
## 110 253.4248722 304.18   50.7551278
## 111 228.2068562 154.74  -73.4668562
## 112 253.0305838 278.39   25.3594162
## 113 125.9547285 114.28  -11.6747285
## 114 156.9158558 233.16   76.2441442
## 115 236.7849290 205.51  -31.2749290
## 116 253.0216875 197.42  -55.6016875
## 117 172.6179357 195.64   23.0220643
## 118 118.8082827 139.56   20.7517173
## 119 235.8892587 320.37   84.4807413
## 120  69.5324233  50.10  -19.4324233
## 121  68.0493894  64.11   -3.9393894
## 122 134.8512926 123.27  -11.5812926
## 123 101.9027471 107.86    5.9572529
```

```
## 124   9.4058839  45.55   36.1441161
## 125  78.7837520  58.68  -20.1037520
## 126 210.3233318 250.73   40.4066682
## 127  79.5829286  65.80  -13.7829286
## 128 140.3905655 127.38  -13.0105655
```

```r
accuracy(air.lm.pred,valid.df$FARE)
```

```
##                     ME     RMSE      MAE       MPE     MAPE
## Test set -0.01073946 41.75928 32.76367 -6.259192 24.28634
```

From the probibility values we can figure out Vacation,Herfindahl index,SouthWest Airline serving the route ,Destination Population Income,Starting city's population,End, city's population,SlotFree,GateFree,Distance,Number of passengers are significant variables for linear regression modelling

From Residual VS Fitted Values We see the values are evenly distributed around the line so our linear assumption is valid``From Normal Q-Q We can see that only one outlier is not following the normal curve and every other values are following normal line so our assumption of Normal Distribution is valid tooFrom Scale Location Plot we see the values are evenly distributed above and below the residual line therefore we are satisfying homoskadacityFrom Residuals Vs Leverage plot we can see that 373 is an influential observation and there would be some change in the model"'

Question4)Using leaps package, run stepwise regression to reduce the number of predictors.Discuss the results from this model **Stepwise Regression**

```r
search.stepwise <- regsubsets(FARE ~ ., data = train.df, nbest = 1, nvmax = dim(train.df)[2],
                     method = "seqrep")
sum <- summary(search.stepwise)
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   TRUE  TRUE        TRUE  TRUE  TRUE     TRUE     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  FALSE FALSE     TRUE     TRUE     TRUE  TRUE
## 8  FALSE FALSE     TRUE     TRUE     TRUE  TRUE
## 9   TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
## 10  TRUE  TRUE     TRUE    FALSE    FALSE FALSE
## 11  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
```

```
## 12  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
## 13  TRUE  TRUE     TRUE     TRUE     TRUE  TRUE
```

```r
sum$rsq
```

```
##  [1] 0.4475868 0.6115635 0.7226510 0.7509498 0.7619641 0.7798545 0.7833041
##  [8] 0.7878218 0.7945906 0.6416637 0.7990185 0.7999065 0.8001028
```

```r
sum$adjr2
```

```
##  [1] 0.4464994 0.6100312 0.7210067 0.7489772 0.7596027 0.7772286 0.7802825
##  [8] 0.7844337 0.7908932 0.6344826 0.7945792 0.7950752 0.7948635
```

```r
sum$cp
```

```
##  [1] 864.68903 459.81773 186.17909 117.96199  92.63245  50.24146  43.68206
##  [8]  34.47261  19.67727 401.13097  12.69036  12.48712  14.00000
```

From squared R we are getting the highest value when we are considering all 13 variables.We need to consider 12 variables because mallow cp is lowest for 12 variable model and highest adjusted r squared for 12 variable 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.exhaustive <- regsubsets(FARE ~ ., data = train.df, nbest = 1, nvmax = dim(train.df)[2],
                     method = "exhaustive")
sum <- summary(search.exhaustive)


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  FALSE FALSE     TRUE     TRUE     TRUE  TRUE
## 8  FALSE FALSE     TRUE     TRUE     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
```

```r
sum$rsq
```

```
##  [1] 0.4475868 0.6115635 0.7226510 0.7509498 0.7619641 0.7798545 0.7833041
##  [8] 0.7878218 0.7945906 0.7979499 0.7990185 0.7999065 0.8001028
```

```r
sum$adjr2
```

```
##  [1] 0.4464994 0.6100312 0.7210067 0.7489772 0.7596027 0.7772286 0.7802825
##  [8] 0.7844337 0.7908932 0.7939008 0.7945792 0.7950752 0.7948635
```

```r
sum$cp
```

```
##  [1] 864.68903 459.81773 186.17909 117.96199  92.63245  50.24146  43.68206
##  [8]  34.47261  19.67727  13.34198  12.69036  12.48712  14.00000
```

From squared R we are getting the highest value when we are considering all 13 variables.We need to consider 12 variables because mallow cp is lowest for 12 variable model and highest adjusted r squared for 12 variable model

```r
# 6. Compare the predictive accuracy of both models-stepwise regression and exhaustive search-using mea

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

```
## Start:  AIC=3604.91
## 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       557 567472 3603.4
## <none>                  566915 3604.9
## - S_INCOME  1      2715 569630 3605.3
## - NEW       1      3049 569964 3605.6
## - E_INCOME  1     11113 578028 3612.8
## - S_POP     1     18544 585459 3619.3
## - SLOT      1     19356 586270 3620.0
## - E_POP     1     22258 589173 3622.6
## - PAX       1     28462 595377 3627.9
## - GATE      1     31592 598507 3630.6
## - HI        1     72748 639663 3664.5
## - VACATION  1    110469 677384 3693.7
## - SW        1    142183 709098 3717.0
## - DISTANCE  1    423576 990490 3887.5
##
## Step:  AIC=3603.41
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                   567472 3603.4
## - S_INCOME  1      2518  569990 3603.7
## - NEW       1      3121  570592 3604.2
## + COUPON    1       557  566915 3604.9
## - E_INCOME  1     10873  578344 3611.1
```

```
## - S_POP     1     18199  585671 3617.5
## - SLOT      1     20223  587695 3619.3
## - E_POP     1     22659  590131 3621.4
## - GATE      1     31718  599190 3629.2
## - PAX       1     37597  605068 3634.1
## - HI        1     75019  642491 3664.7
## - VACATION  1    112334  679806 3693.5
## - SW        1    145969  713441 3718.2
## - DISTANCE  1    854008 1421479 4069.7
```

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

```
##                    ME     RMSE      MAE       MPE     MAPE
## Test set -0.07438682 41.69746 32.73148 -6.269217 24.31014
```

```r
air.lm.exhaustive <- lm(FARE ~ NEW+VACATION+SW+HI+S_INCOME+E_INCOME+S_POP+E_POP+GATE+SLOT+DISTANCE+PAX,
exhaustive.pred<-predict(air.lm.exhaustive,valid.df)
accuracy(exhaustive.pred,valid.df$FARE)
```

```
##                    ME     RMSE      MAE       MPE     MAPE
## Test set -0.07438682 41.69746 32.73148 -6.269217 24.31014
```

The RMSE value for Exhaustive Model is 31.03422 and the RMSE value for Stepwise Regression Model is 30.8338.Lesser RMSE value, the better the fit.Hence, we concclude by saying the Stepwise Regression Model is a slightly better fit than the Exhaustive Search model; although both models are similar since the RMSE values are comparable. #Keeping in mind the number of variables and the values of RMSE, Stepwise Regression Model is more attractive.

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
newrow <- list(COUPON = 1.202, NEW = 3, VACATION = "No", SW = "No", HI = 4442.141, S_INCOME = 28760, E_
new <- rbind(air.df, newrow)
newrow.df <- new[nrow(new),]
test_mat = model.matrix(FARE ~ ., data = newrow.df)
coefs = coef(search.exhaustive, id = 12)
prednew = test_mat[, names(coefs)] %*% coefs
prednew
```

```
##          [,1]
## [1,] 249.5502
```

Question 8) Predict the reduction in average fare on the route in question if Southwest decides to cover this route

```r
newrow2 <- list(COUPON = 1.202, NEW = 3, VACATION = "No", SW = "Yes", HI = 4442.141, S_INCOME = 28760, 
new2 <- rbind(new, newrow2)
newrow2.df <- new2[nrow(new2),]
test_mat2 = model.matrix(FARE ~ ., data = newrow2.df)
coefs = coef(search.exhaustive, id = 12)
prednew2 = test_mat2[, names(coefs)] %*% coefs
prednew2
```

```
##          [,1]
## [1,] 204.4021
```

There is drop in the value of 45 in the fare if southwest airlines starts operating

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

```
search <- regsubsets(FARE ~ ., data = train.df, nbest = 1, nvmax = dim(train.df)[2],
                     method = "backward")
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    FALSE
## 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  FALSE FALSE     TRUE     TRUE     TRUE  TRUE
## 8  FALSE  TRUE     TRUE     TRUE     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$rsq
```

```
##  [1] 0.4475868 0.6115635 0.7226510 0.7509498 0.7619641 0.7798545 0.7833041
##  [8] 0.7877417 0.7945906 0.7979499 0.7990185 0.7999065 0.8001028
```

```
sum$adjr2
```

```
##  [1] 0.4464994 0.6100312 0.7210067 0.7489772 0.7596027 0.7772286 0.7802825
##  [8] 0.7843524 0.7908932 0.7939008 0.7945792 0.7950752 0.7948635
```

```
sum$cp
```

```
##  [1] 864.68903 459.81773 186.17909 117.96199  92.63245  50.24146  43.68206
##  [8]  34.67119  19.67727  13.34198  12.69036  12.48712  14.00000
```

The R squared value for the 13 varibles is the highest. Subsequently the adjusted R squared value for the 12 varibles is highest and also the Mallows Cp value also reflects the same for the 12 variables which is the lowest.The backward model has removed the COUPON variable.

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)
```

```
##
## Attaching package: 'MASS'

## The following objects are masked from 'package:fma':
##
##     cement, housing, petrol

## The following object is masked from 'package:dplyr':
##
##     select
```

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

```
## Start:  AIC=3604.91
## 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       557 567472 3603.4
## <none>                  566915 3604.9
## - S_INCOME  1      2715 569630 3605.3
## - NEW       1      3049 569964 3605.6
## - E_INCOME  1     11113 578028 3612.8
## - S_POP     1     18544 585459 3619.3
## - SLOT      1     19356 586270 3620.0
## - E_POP     1     22258 589173 3622.6
## - PAX       1     28462 595377 3627.9
## - GATE      1     31592 598507 3630.6
## - HI        1     72748 639663 3664.5
## - VACATION  1    110469 677384 3693.7
## - SW        1    142183 709098 3717.0
## - DISTANCE  1    423576 990490 3887.5
##
## Step:  AIC=3603.41
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##             Df Sum of Sq    RSS    AIC
## <none>                   567472 3603.4
## - S_INCOME  1      2518  569990 3603.7
## - NEW       1      3121  570592 3604.2
## - E_INCOME  1     10873  578344 3611.1
## - S_POP     1     18199  585671 3617.5
## - SLOT      1     20223  587695 3619.3
## - E_POP     1     22659  590131 3621.4
## - GATE      1     31718  599190 3629.2
## - PAX       1     37597  605068 3634.1
## - HI        1     75019  642491 3664.7
```

```
## - VACATION  1    112334  679806 3693.5
## - SW        1    145969  713441 3718.2
## - DISTANCE  1    854008 1421479 4069.7
```

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

```
##
## Call:
## lm(formula = FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -81.22 -21.77  -0.88  19.48 129.84
##
## Coefficients:
##                  Estimate    Std. Error t value            Pr(>|t|)
## (Intercept)  42.7003222223 22.9948188200   1.857             0.06391 .
## NEW          -3.3516419387  2.0273187916  -1.653             0.09891 .
## VACATIONYes -37.7764676699  3.8085497925  -9.919 < 0.0000000000000002 ***
## SWYes       -45.1480636881  3.9930247809 -11.307 < 0.0000000000000002 ***
## HI            0.0084285988  0.0010398328   8.106  0.00000000000000411 ***
## S_INCOME      0.0008094035  0.0005450170   1.485             0.13815
## E_INCOME      0.0012767735  0.0004137507   3.086             0.00214 **
## S_POP         0.0000028187  0.0000007060   3.992  0.00007528449392384 ***
## E_POP         0.0000036415  0.0000008174   4.455  0.00001038081267020 ***
## SLOTFree    -17.2991299205  4.1104965534  -4.209  0.00003050736120982 ***
## GATEFree    -22.5723574082  4.2826715623  -5.271  0.00000020305116676 ***
## DISTANCE      0.0741717467  0.0027120735  27.349 < 0.0000000000000002 ***
## PAX          -0.0008064926  0.0001405460  -5.738  0.00000001664529731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.79 on 497 degrees of freedom
## Multiple R-squared:  0.7999, Adjusted R-squared:  0.7951
## F-statistic: 165.6 on 12 and 497 DF,  p-value: < 0.00000000000000022
```

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

```
##                   ME     RMSE      MAE       MPE     MAPE
## Test set -0.07438682 41.69746 32.73148 -6.269217 24.31014
```

The StepAIC model is based on the Akaike information Criteria .According to this model the Variblewith the lowest AIC values are removed and subsequently the results are produced .Here the lowest AIC value is for the COUNPON variable and is therefore removed .Considering the AIC value for all the 13 varibles is 3604.91 when the variable COUNPON has been removed the AIC value is 3603.41.