Airfares Analysis

Homer Kay

3/26/2018

```
Airfares <- read.csv("~/Desktop/Data Mining/Airfares.csv")
str(Airfares)
```

```
## 'data.frame':    638 obs. of  18 variables:
##  $ S_CODE  : Factor w/ 8 levels "*","DCA","EWR",..: 1 1 1 8 7 1 1 1 1 1 ..
.
##  $ S_CITY  : Factor w/ 51 levels "Albuquerque         NM",..: 14 3 7 9 9 1
1 14 18 23 25 ...
##  $ E_CODE  : Factor w/ 8 levels "*","DCA","EWR",..: 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ E_CITY  : Factor w/ 68 levels "Amarillo            TX",..: 1 2 2 2 2 2
2 2 2 2 ...
##  $ 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: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ SW      : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 2 1 2 2 2 ...
##  $ 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 7
145897 7145897 7145897 ...
##  $ SLOT    : Factor w/ 2 levels "Controlled","Free": 2 2 2 1 2 2 2 2 2 2 .
..
##  $ GATE    : Factor w/ 2 levels "Constrained",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ 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 ...
```

```
## STR shows which varaibles are numeric, integer, factor, etc.
library(corrplot)
```

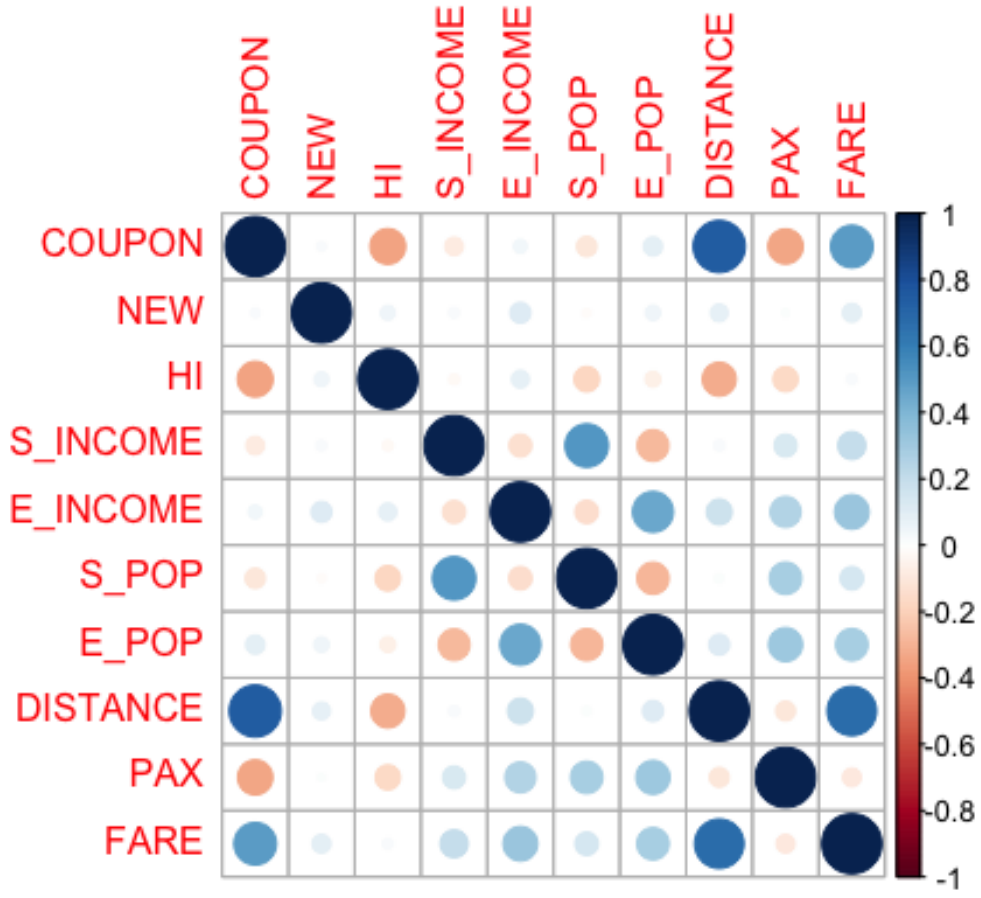
```
## corrplot 0.84 loaded
```

```
## Make correlation matrix with numeric predictors
CORR_MATRIX <- cor(Airfares[,c(5:6,9:13,16:18)])
CORR_MATRIX
```

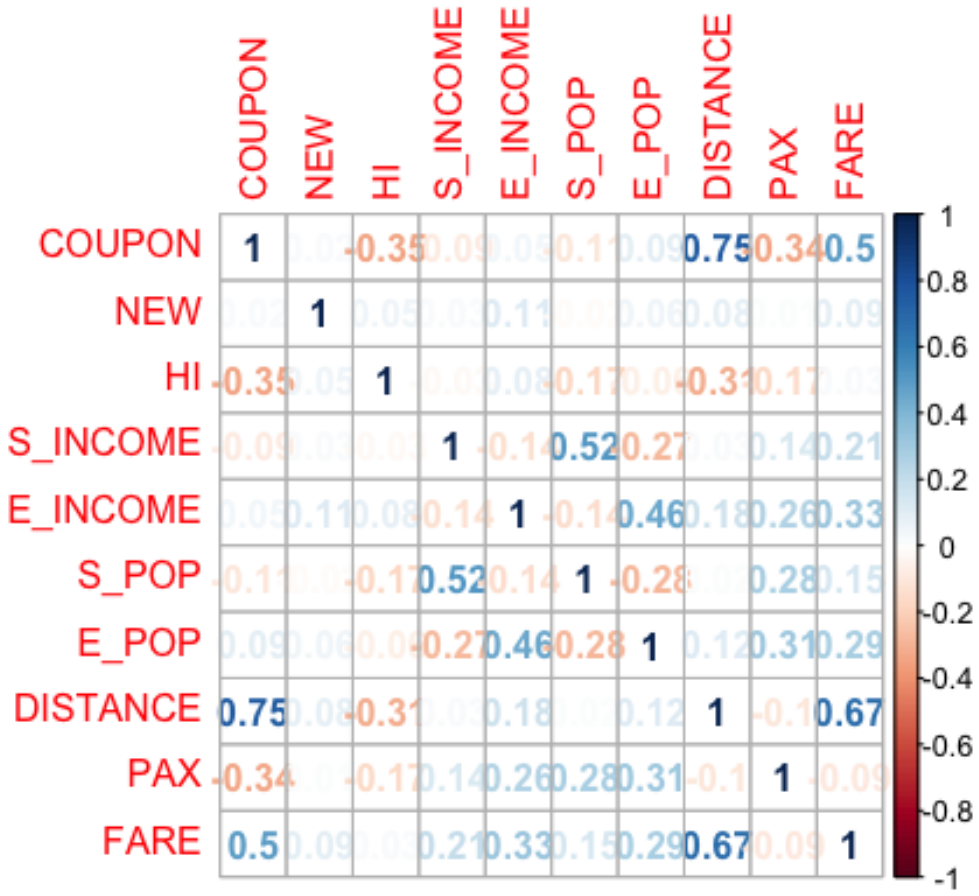
```
##                COUPON         NEW          HI    S_INCOME    E_INCOME
## COUPON     1.00000000  0.02022307 -0.34725207 -0.08840265  0.0468892
## NEW        0.02022307  1.00000000  0.05414685  0.02659673  0.1133766
## HI        -0.34725207  0.05414685  1.00000000 -0.02738221  0.0823926
```

```
## S_INCOME -0.08840265  0.02659673 -0.02738221  1.00000000 -0.1388642
## E_INCOME  0.04688920  0.11337664  0.08239260 -0.13886420  1.0000000
## S_POP    -0.10776336 -0.01667212 -0.17249541  0.51718718 -0.1440586
## E_POP     0.09496994  0.05856818 -0.06245600 -0.27228027  0.4584181
## DISTANCE  0.74680521  0.08096520 -0.31237457  0.02815334  0.1765307
## PAX      -0.33697358  0.01049527 -0.16896078  0.13819710  0.2599611
## FARE      0.49653696  0.09172969  0.02519492  0.20913485  0.3260923
##                S_POP       E_POP    DISTANCE         PAX        FARE
## COUPON   -0.10776336  0.09496994  0.74680521 -0.33697358  0.49653696
## NEW      -0.01667212  0.05856818  0.08096520  0.01049527  0.09172969
## HI       -0.17249541 -0.06245600 -0.31237457 -0.16896078  0.02519492
## S_INCOME  0.51718718 -0.27228027  0.02815334  0.13819710  0.20913485
## E_INCOME -0.14405857  0.45841806  0.17653074  0.25996105  0.32609229
## S_POP     1.00000000 -0.28014283  0.01843667  0.28461056  0.14509708
## E_POP    -0.28014283  1.00000000  0.11563970  0.31469750  0.28504299
## DISTANCE  0.01843667  0.11563970  1.00000000 -0.10248160  0.67001599
## PAX       0.28461056  0.31469750 -0.10248160  1.00000000 -0.09070541
## FARE      0.14509708  0.28504299  0.67001599 -0.09070541  1.00000000
```

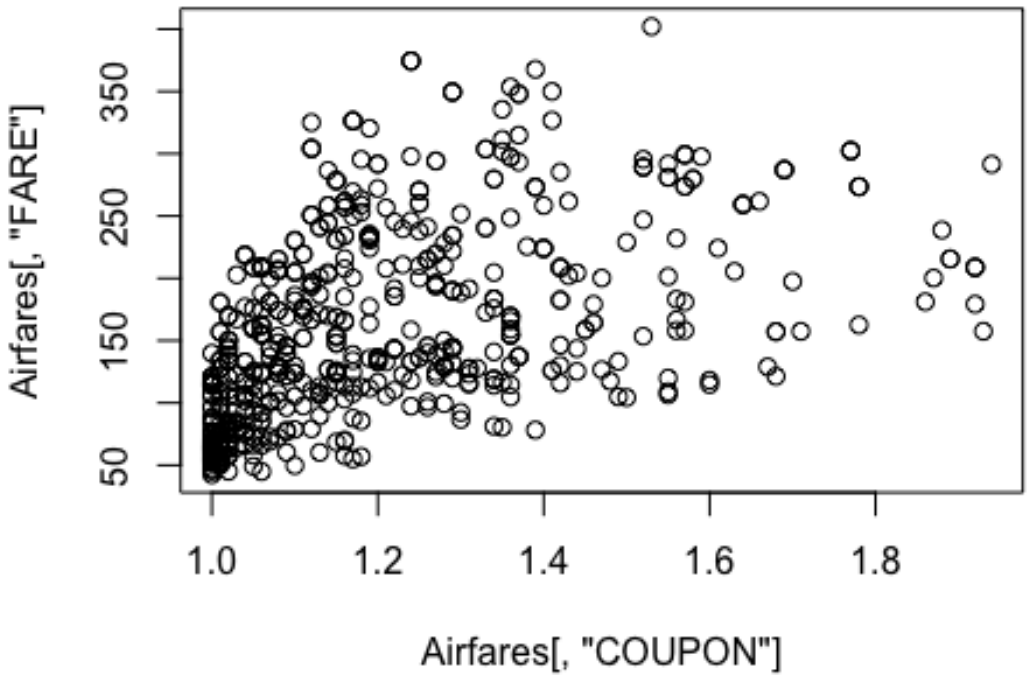
```
corrplot(CORR_MATRIX)
```



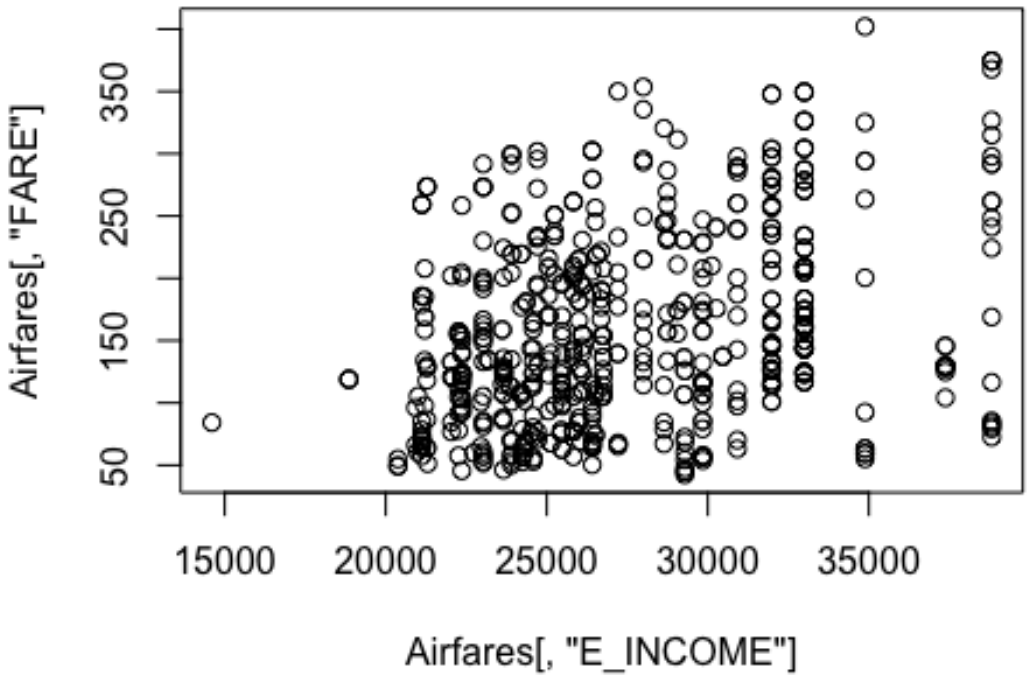

```
## Output plot with numeric values.
corrplot(CORR_MATRIX, method ="number")
```



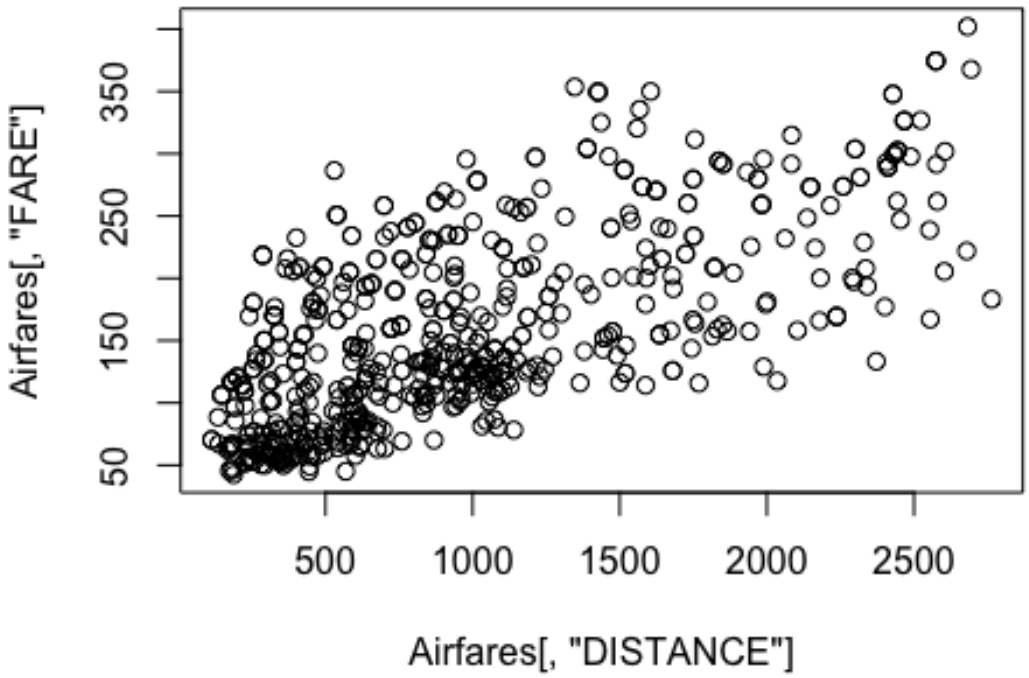

```
plot_COUPON<-plot(Airfares[,"COUPON"],Airfares[,"FARE"])
```




```
plot_E_INCOME<-plot(Airfares[,"E_INCOME"],Airfares[,"FARE"])
```




```
plot_DISTANCE<-plot(Airfares[,"DISTANCE"],Airfares[,"FARE"])
```




```
## Categorical Predictors, excluding 1st 4: Columns (7:8,14:15)
library(reshape)
summary(Airfares$FARE)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   42.47  106.29  144.60  160.88  209.35  402.02

## Mean = 160.88
PVT1 <- aggregate(Airfares$FARE, by=list(Airfares$VACATION), FUN=mean, na.rm=
TRUE)
PVT1

##   Group.1        x
## 1      No 173.5525
## 2     Yes 125.9809

PVT2 <- aggregate(Airfares$FARE, by=list(Airfares$SW), FUN=mean, na.rm=TRUE)
PVT2

##   Group.1         x
## 1      No 188.18279
## 2     Yes  98.38227
```

```r
PVT3 <- aggregate(Airfares$FARE, by=list(Airfares$SLOT), FUN=mean, na.rm=TRUE
)
PVT3
```

```
##      Group.1        x
## 1 Controlled 186.0594
## 2       Free 150.8257
```

```r
PVT4 <- aggregate(Airfares$FARE, by=list(Airfares$GATE), FUN=mean, na.rm=TRUE
)
PVT4
```

```
##       Group.1       x
## 1 Constrained 193.129
## 2        Free 153.096
```

```r
summary(Airfares$VACATION)
```

```
##  No Yes
## 468 170
```

```r
summary(Airfares$SW)
```

```
##  No Yes
## 444 194
```

```r
summary(Airfares$SLOT)
```

```
## Controlled       Free
##        182        456
```

```r
summary(Airfares$GATE)
```

```
## Constrained        Free
##         124         514
```

```r
# Making Dummy Variables
VAC_DUM <- ifelse(Airfares$VACATION=="Yes",1,0)
SW_DUM <- ifelse(Airfares$SW=="Yes",1,0)
SLOT_DUM <- ifelse(Airfares$SLOT=="Controlled",1,0)
GATE_DUM <-  ifelse(Airfares$GATE=="Constrained",1,0)
# Combining Dataset
DATASET2 <- cbind(Airfares, VAC_DUM)
DATASET2 <- cbind(DATASET2, SW_DUM)
DATASET2 <- cbind(DATASET2, SLOT_DUM)
DATASET2 <- cbind(DATASET2, GATE_DUM)
# Partition
set.seed(123)
# Get rid of extra variables
DATASET2.index <- DATASET2[order(runif(638)), ]#randomized the observations
train <- DATASET2.index[1:510, ] #create training set
```

```r
valid <- DATASET2.index[511:638, ] #create validation set
dim(train)
```

```
## [1] 510  22
```

```r
dim(valid)
```

```
## [1] 128  22
```

```r
## Get rid of 1-4 variables and categorical variables (now replaced by dummy
variables)
train_reduce <- train[,c(-1,-2,-3,-4,-7,-8,-14,-15)]
options(scipen=999) #no Sci Notation
train.lm <- lm(FARE~., data=train_reduce)
##Stepwise Regression
stepreg.train <- step(train.lm, direction="both")
```

```
## Start:  AIC=3664.75
## FARE ~ COUPON + NEW + HI + S_INCOME + E_INCOME + S_POP + E_POP +
##     DISTANCE + PAX + VAC_DUM + SW_DUM + SLOT_DUM + GATE_DUM
##
##             Df Sum of Sq     RSS    AIC
## - NEW        1       884  638373 3663.5
## - COUPON     1      1932  639421 3664.3
## <none>                    637489 3664.7
## - S_INCOME   1      6578  644067 3668.0
## - E_INCOME   1     15848  653337 3675.3
## - SLOT_DUM   1     23982  661471 3681.6
## - S_POP      1     25134  662624 3682.5
## - GATE_DUM   1     27297  664786 3684.1
## - E_POP      1     34800  672290 3689.9
## - PAX        1     35707  673196 3690.5
## - HI         1     77260  714749 3721.1
## - VAC_DUM    1    102176  739665 3738.6
## - SW_DUM     1    108075  745564 3742.6
## - DISTANCE   1    371510 1008999 3896.9
##
## Step:  AIC=3663.46
## FARE ~ COUPON + HI + S_INCOME + E_INCOME + S_POP + E_POP + DISTANCE +
##     PAX + VAC_DUM + SW_DUM + SLOT_DUM + GATE_DUM
##
##             Df Sum of Sq     RSS    AIC
## - COUPON     1      2101  640474 3663.1
## <none>                    638373 3663.5
## + NEW        1       884  637489 3664.7
## - S_INCOME   1      6680  645053 3666.8
## - E_INCOME   1     15667  654040 3673.8
## - SLOT_DUM   1     23531  661904 3679.9
## - S_POP      1     25576  663950 3681.5
## - GATE_DUM   1     27309  665682 3682.8
## - E_POP      1     34891  673265 3688.6
```

```
## - PAX       1     35492  673865 3689.1
## - HI        1     76566  714939 3719.2
## - VAC_DUM   1    101710  740083 3736.9
## - SW_DUM    1    107347  745720 3740.7
## - DISTANCE  1    370825 1009198 3895.0
##
## Step:  AIC=3663.13
## FARE ~ HI + S_INCOME + E_INCOME + S_POP + E_POP + DISTANCE +
##     PAX + VAC_DUM + SW_DUM + SLOT_DUM + GATE_DUM
##
##            Df Sum of Sq     RSS    AIC
## <none>                   640474 3663.1
## + COUPON    1      2101  638373 3663.5
## + NEW       1      1053  639421 3664.3
## - S_INCOME  1      6376  646850 3666.2
## - E_INCOME  1     14505  654979 3672.6
## - SLOT_DUM  1     24628  665102 3680.4
## - S_POP     1     24659  665133 3680.4
## - GATE_DUM  1     27670  668144 3682.7
## - E_POP     1     36239  676713 3689.2
## - PAX       1     49358  689832 3699.0
## - HI        1     75443  715917 3717.9
## - VAC_DUM   1    102879  743353 3737.1
## - SW_DUM    1    110602  751076 3742.4
## - DISTANCE  1    800294 1440768 4074.6
```

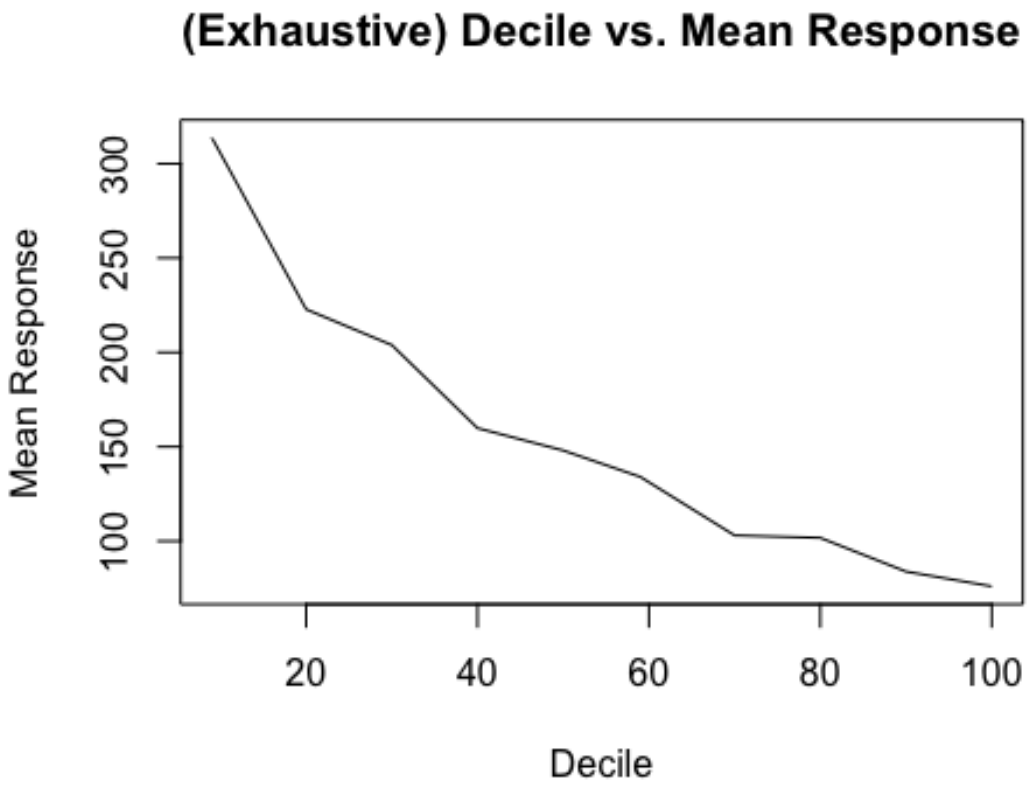
```
summary(stepreg.train)
```

```
##
## Call:
## lm(formula = FARE ~ HI + S_INCOME + E_INCOME + S_POP + E_POP +
##     DISTANCE + PAX + VAC_DUM + SW_DUM + SLOT_DUM + GATE_DUM,
##     data = train_reduce)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -106.350  -22.172   -1.482   22.171  128.439
##
## Coefficients:
##                   Estimate     Std. Error t value             Pr(>|t|)
## (Intercept) -29.4323056333  22.8407590327  -1.289             0.198140
## HI            0.0082820859   0.0010813486   7.659   0.0000000000000983 ***
## S_INCOME      0.0013623841   0.0006118609   2.227             0.026419 *
## E_INCOME      0.0014188566   0.0004224950   3.358             0.000844 ***
## S_POP         0.0000032738   0.0000007476   4.379   0.0000145560565069 ***
## E_POP         0.0000045268   0.0000008528   5.308   0.0000001669713270 ***
## DISTANCE      0.0740793682   0.0029696727  24.945 < 0.0000000000000002 ***
## PAX          -0.0009450523   0.0001525509  -6.195   0.0000000012218637 ***
## VAC_DUM     -36.9188959113   4.1278215282  -8.944 < 0.0000000000000002 ***
## SW_DUM      -39.9364354733   4.3064879160  -9.274 < 0.0000000000000002 ***
```

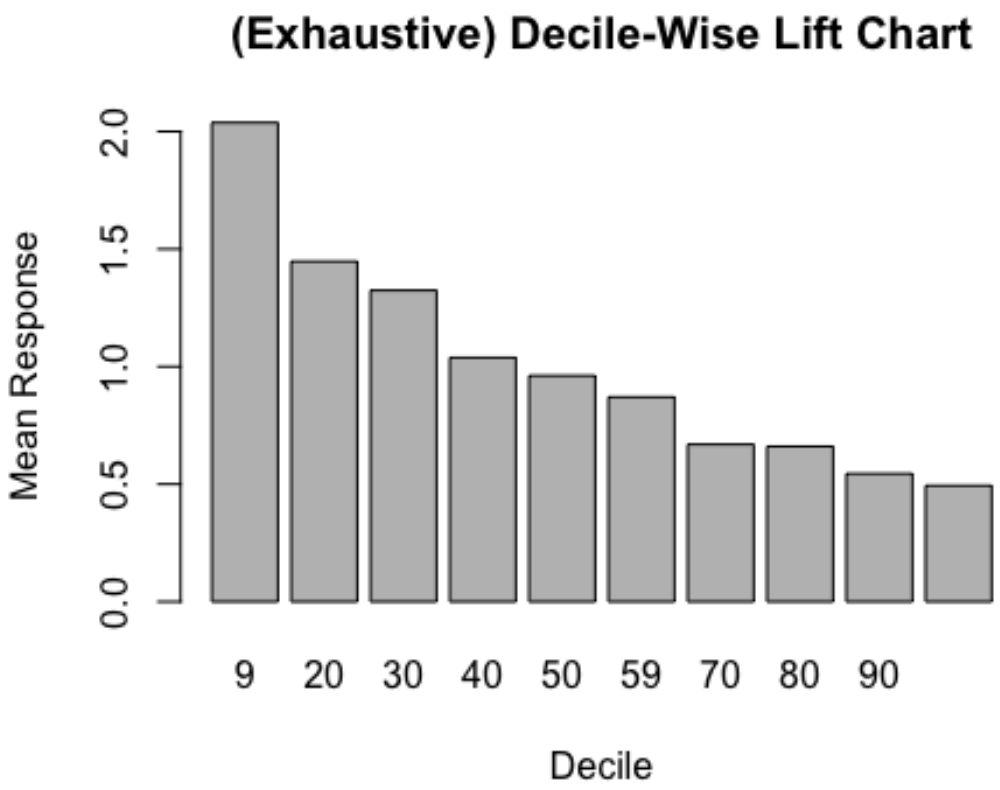
```
## SLOT_DUM     18.8581797839   4.3094329357   4.376   0.0000147331131613 ***
## GATE_DUM     20.1757334826   4.3497285177   4.638   0.0000044936764467 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.86 on 498 degrees of freedom
## Multiple R-squared:  0.7801, Adjusted R-squared:  0.7753
## F-statistic: 160.6 on 11 and 498 DF,  p-value: < 0.00000000000000022

##Exhaustive Search
library(leaps)
exh.search <- regsubsets(FARE~., data = train_reduce, nbest=1, nvmax=dim(trai
n_reduce)[2], method="exhaustive")
sum <- summary(exh.search)
sum$which

##    (Intercept) COUPON   NEW    HI S_INCOME E_INCOME S_POP E_POP DISTANCE
## 1         TRUE  FALSE FALSE FALSE    FALSE    FALSE FALSE FALSE     TRUE
## 2         TRUE  FALSE FALSE FALSE    FALSE    FALSE FALSE FALSE     TRUE
## 3         TRUE  FALSE FALSE FALSE    FALSE    FALSE FALSE FALSE     TRUE
## 4         TRUE  FALSE FALSE  TRUE    FALSE    FALSE FALSE FALSE     TRUE
## 5         TRUE  FALSE FALSE  TRUE    FALSE    FALSE FALSE FALSE     TRUE
## 6         TRUE  FALSE FALSE  TRUE    FALSE    FALSE FALSE FALSE     TRUE
## 7         TRUE  FALSE FALSE  TRUE    FALSE    FALSE FALSE FALSE     TRUE
## 8         TRUE  FALSE FALSE  TRUE    FALSE    FALSE FALSE  TRUE     TRUE
## 9         TRUE  FALSE FALSE  TRUE    FALSE    FALSE  TRUE  TRUE     TRUE
## 10        TRUE  FALSE FALSE  TRUE    FALSE     TRUE  TRUE  TRUE     TRUE
## 11        TRUE  FALSE FALSE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE
## 12        TRUE   TRUE FALSE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE
## 13        TRUE   TRUE  TRUE  TRUE     TRUE     TRUE  TRUE  TRUE     TRUE
##      PAX VAC_DUM SW_DUM SLOT_DUM GATE_DUM
## 1  FALSE   FALSE  FALSE    FALSE    FALSE
## 2  FALSE   FALSE   TRUE    FALSE    FALSE
## 3  FALSE    TRUE   TRUE    FALSE    FALSE
## 4  FALSE    TRUE   TRUE    FALSE    FALSE
## 5  FALSE    TRUE   TRUE     TRUE    FALSE
## 6  FALSE    TRUE   TRUE     TRUE     TRUE
## 7   TRUE    TRUE   TRUE     TRUE     TRUE
## 8   TRUE    TRUE   TRUE     TRUE     TRUE
## 9   TRUE    TRUE   TRUE     TRUE     TRUE
## 10  TRUE    TRUE   TRUE     TRUE     TRUE
## 11  TRUE    TRUE   TRUE     TRUE     TRUE
## 12  TRUE    TRUE   TRUE     TRUE     TRUE
## 13  TRUE    TRUE   TRUE     TRUE     TRUE

sum$adjr2

##  [1] 0.4189186 0.5748006 0.6907367 0.7194972 0.7370115 0.7524999 0.7555822
##  [8] 0.7607220 0.7696601 0.7735081 0.7752904 0.7755767 0.7754357
```

```
exh.lm.exhaust <- lm(FARE~COUPON+HI+S_INCOME+E_INCOME+S_POP+E_POP+DISTANCE+PA
X+VAC_DUM+SW_DUM+SLOT_DUM+GATE_DUM,data = train_reduce)
summary(exh.lm.exhaust)

##
## Call:
## lm(formula = FARE ~ COUPON + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + DISTANCE + PAX + VAC_DUM + SW_DUM + SLOT_DUM + GATE_DUM,
##     data = train_reduce)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -108.420  -22.240   -1.577   21.714  127.413
##
## Coefficients:
##                    Estimate     Std. Error t value             Pr(>|t|)
## (Intercept) -53.1355430971  29.4040524076  -1.807             0.071354 .
## COUPON       17.9958242985  14.0723658423   1.279             0.201562
## HI            0.0086762964   0.0011237668   7.721   0.0000000000000641 ***
## S_INCOME      0.0013956853   0.0006120253   2.280             0.023004 *
## E_INCOME      0.0014859229   0.0004254703   3.492             0.000521 ***
## S_POP         0.0000033428   0.0000007491   4.462   0.0000100389185633 ***
## E_POP         0.0000044522   0.0000008542   5.212   0.0000002744779180 ***
## DISTANCE      0.0703837770   0.0041423551  16.991 < 0.0000000000000002 ***
## PAX          -0.0008655144   0.0001646529  -5.257   0.0000002182746190 ***
## VAC_DUM     -36.7316236727   4.1277894479  -8.899 < 0.0000000000000002 ***
## SW_DUM      -39.4796200881   4.3185430419  -9.142 < 0.0000000000000002 ***
## SLOT_DUM     18.4773519150   4.3169704661   4.280   0.0000224152220346 ***
## GATE_DUM     20.0489617301   4.3480867176   4.611   0.0000051033018057 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.84 on 497 degrees of freedom
## Multiple R-squared:  0.7809, Adjusted R-squared:  0.7756
## F-statistic: 147.6 on 12 and 497 DF,  p-value: < 0.00000000000000022

##Compare Predictive Accuracy
## Stepwise
valid_reduce <- valid[,c(-1,-2,-3,-4,-7,-8,-14,-15)]
library(forecast)
pred_v_stepreg <-predict(stepreg.train,valid_reduce)
accuracy(pred_v_stepreg, valid_reduce$FARE)

##                 ME     RMSE      MAE       MPE     MAPE
## Test set -2.228121 34.14603 27.15491 -6.289403 21.20073

##Compare Predictive Accuracy
## Exhaustive
pred_v_exh <-predict(exh.lm.exhaust,valid_reduce)
accuracy(pred_v_exh, valid_reduce$FARE)
```

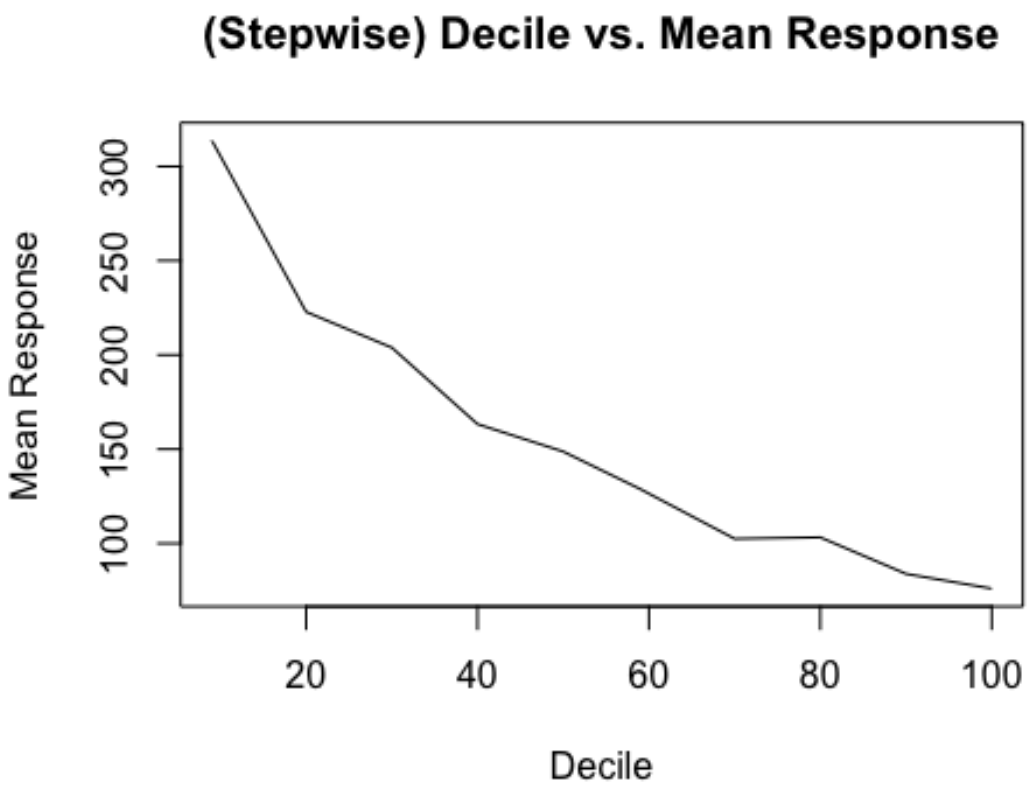
```
##                   ME     RMSE      MAE       MPE     MAPE
## Test set -2.265262 34.55807 27.56386 -6.240263 21.40841

##Lift Charts - Exh
library(gains)
gain.num <- gains(valid_reduce$FARE, pred_v_exh, groups=10)
plot(gain.num$depth, gain.num$mean.resp, xlab= "Decile", ylab = "Mean Respons
e", main = "(Exhaustive) Decile vs. Mean Response",type="l")
```



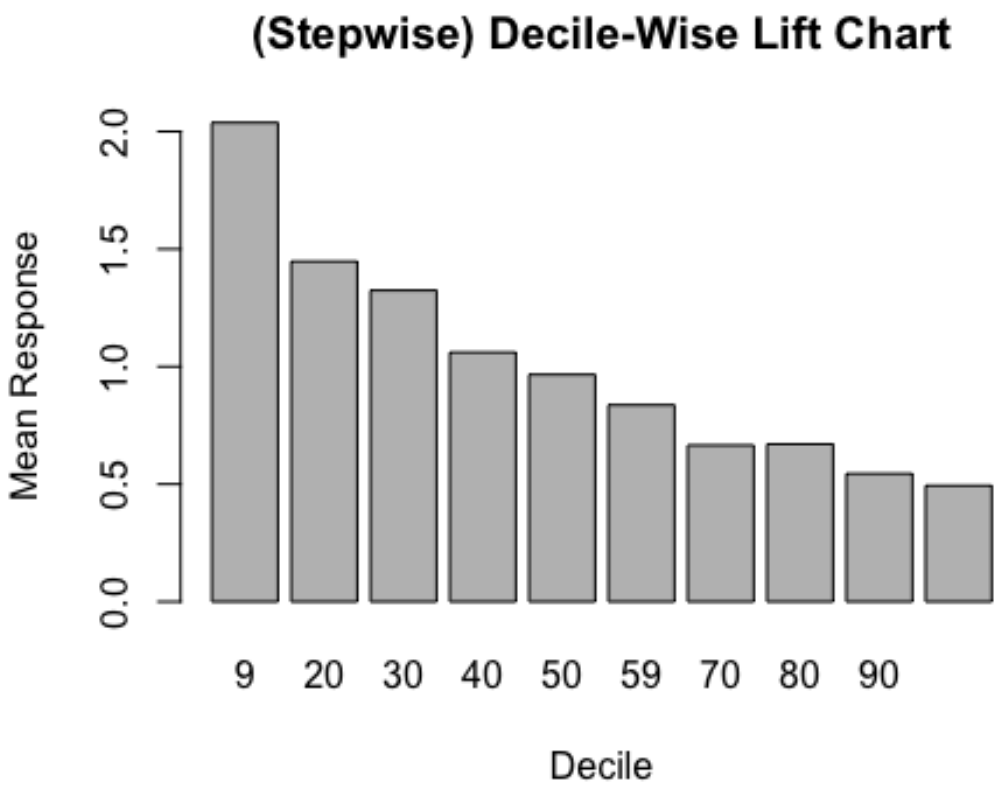

```
##Barplot - Exh
barplot(gain.num$mean.resp/mean(valid_reduce$FARE), names.arg=gain.num$depth,
xlab = "Decile",ylab="Mean Response",main="(Exhaustive) Decile-Wise Lift Char
t")
```




```
##Lift Charts - Step
gain.num2 <- gains(valid_reduce$FARE, pred_v_stepreg, groups=10)
plot(gain.num2$depth, gain.num2$mean.resp, xlab= "Decile", ylab = "Mean Respo
nse", main = "(Stepwise) Decile vs. Mean Response",type="l")
```




```
##Barplot - Step
barplot(gain.num2$mean.resp/mean(valid_reduce$FARE), names.arg=gain.num2$dept
h, xlab = "Decile",ylab="Mean Response",main="(Stepwise) Decile-Wise Lift Cha
rt")
```




```
##Prediction
COUPON<- 1.202
NEW <- 3
HI <- 4442.141
S_INCOME<-28760
E_INCOME<-27664
S_POP<-4557004
E_POP<-3195503
DISTANCE<-1976
PAX<-12782
FARE<-0
VAC_DUM<-0
SW_DUM<-0
SLOT_DUM<-0
GATE_DUM<-0

PV_VALUE <- data.frame(COUPON,NEW,HI,S_INCOME,E_INCOME,S_POP,E_POP,DISTANCE,P
AX,FARE,VAC_DUM,SW_DUM,SLOT_DUM,GATE_DUM)
pred_v_exh2 <-predict(exh.lm.exhaust,PV_VALUE)
pred_v_exh2

##        1
## 245.7588
```

```r
##Prediction
COUPON<- 1.202
NEW <- 3
HI <- 4442.141
S_INCOME<-28760
E_INCOME<-27664
S_POP<-4557004
E_POP<-3195503
DISTANCE<-1976
PAX<-12782
FARE<-0
VAC_DUM<-0
SW_DUM<-1
SLOT_DUM<-0
GATE_DUM<-0

PV_VALUE <- data.frame(COUPON,NEW,HI,S_INCOME,E_INCOME,S_POP,E_POP,DISTANCE,P
AX,FARE,VAC_DUM,SW_DUM,SLOT_DUM,GATE_DUM)
pred_v_exh2 <-predict(exh.lm.exhaust,PV_VALUE)
pred_v_exh2
```

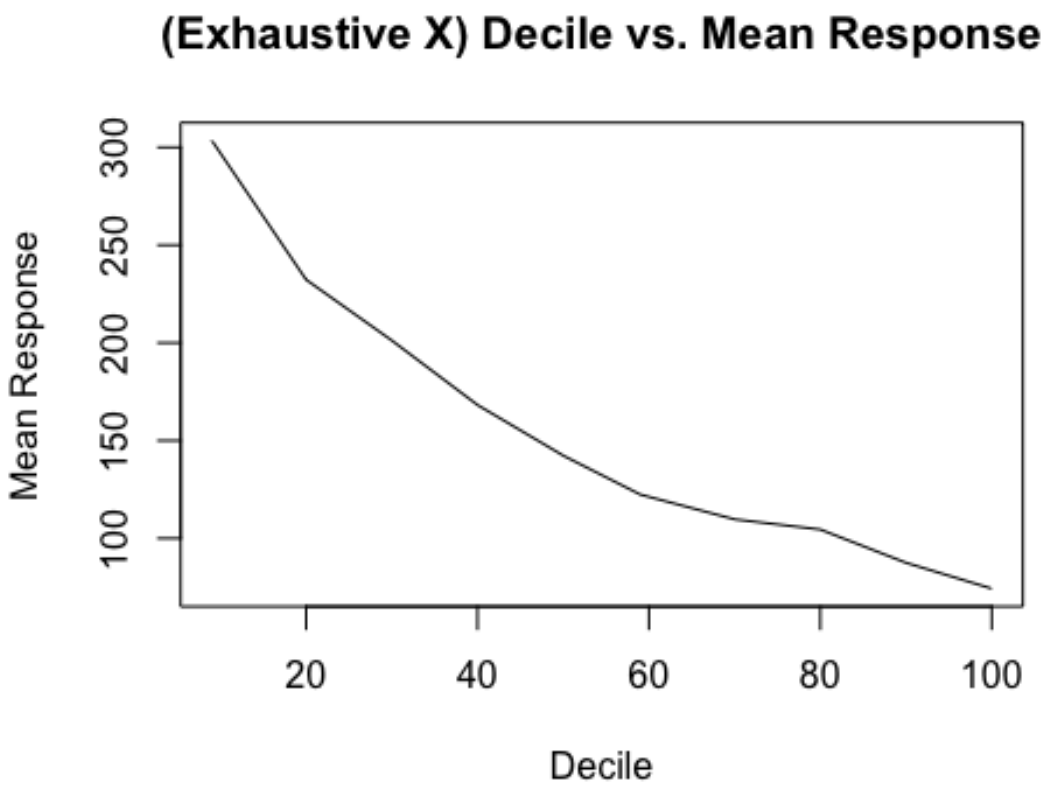
```
##        1
## 206.2791
```

```r
##Exhaustive Search
exh.search.reduce <- regsubsets(FARE~COUPON+VAC_DUM+SW_DUM+HI+S_INCOME+E_INCO
ME+S_POP+E_POP+DISTANCE, data = train_reduce, nbest=1, nvmax=dim(train_reduce
)[2], method="exhaustive")
sum <- summary(exh.search.reduce)
sum$which
```

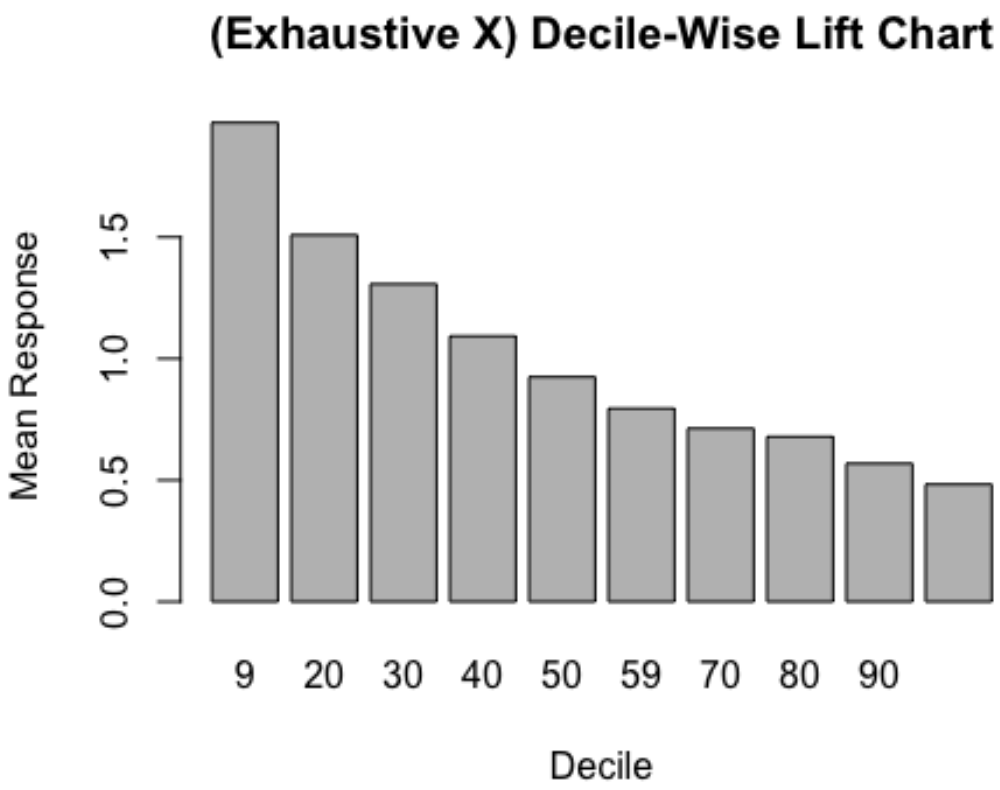
```
##   (Intercept) COUPON VAC_DUM SW_DUM    HI S_INCOME E_INCOME S_POP E_POP
## 1        TRUE  FALSE   FALSE  FALSE FALSE    FALSE    FALSE FALSE FALSE
## 2        TRUE  FALSE   FALSE   TRUE FALSE    FALSE    FALSE FALSE FALSE
## 3        TRUE  FALSE    TRUE   TRUE FALSE    FALSE    FALSE FALSE FALSE
## 4        TRUE  FALSE    TRUE   TRUE  TRUE    FALSE    FALSE FALSE FALSE
## 5        TRUE  FALSE    TRUE   TRUE  TRUE    FALSE    FALSE FALSE  TRUE
## 6        TRUE  FALSE    TRUE   TRUE  TRUE    FALSE    FALSE  TRUE  TRUE
## 7        TRUE   TRUE    TRUE   TRUE  TRUE    FALSE    FALSE  TRUE  TRUE
## 8        TRUE   TRUE    TRUE   TRUE  TRUE    FALSE     TRUE  TRUE  TRUE
## 9        TRUE   TRUE    TRUE   TRUE  TRUE     TRUE     TRUE  TRUE  TRUE
##   DISTANCE
## 1     TRUE
## 2     TRUE
## 3     TRUE
## 4     TRUE
## 5     TRUE
## 6     TRUE
## 7     TRUE
```

```
## 8     TRUE
## 9     TRUE

sum$adjr2

## [1] 0.4189186 0.5748006 0.6907367 0.7194972 0.7269040 0.7397251 0.7440842
## [8] 0.7469646 0.7493620

exh.lm.exhaust.2 <- lm(FARE~COUPON+VAC_DUM+SW_DUM+HI+S_INCOME+E_INCOME+S_POP+
E_POP+DISTANCE,data = train_reduce)
summary(exh.lm.exhaust.2)

##
## Call:
## lm(formula = FARE ~ COUPON + VAC_DUM + SW_DUM + HI + S_INCOME +
##     E_INCOME + S_POP + E_POP + DISTANCE, data = train_reduce)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -126.271  -25.698   -2.209   26.474  117.971
##
## Coefficients:
##                    Estimate     Std. Error t value             Pr(>|t|)
## (Intercept) -89.2920423343  30.5275183412  -2.925             0.003601 **
## COUPON       51.8503579040  13.7120212649   3.781             0.000175 ***
## VAC_DUM     -40.0462320876   4.2757808182  -9.366 < 0.0000000000000002 ***
## SW_DUM      -47.1447463615   4.4140627275 -10.681 < 0.0000000000000002 ***
## HI            0.0104379241   0.0011159793   9.353 < 0.0000000000000002 ***
## S_INCOME      0.0015214370   0.0006321645   2.407             0.016459 *
## E_INCOME      0.0012218071   0.0004341256   2.814             0.005079 **
## S_POP         0.0000033304   0.0000007113   4.682          0.000003665 ***
## E_POP         0.0000040887   0.0000007722   5.295          0.000000179 ***
## DISTANCE      0.0643602571   0.0042740018  15.059 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.87 on 500 degrees of freedom
## Multiple R-squared:  0.7538, Adjusted R-squared:  0.7494
## F-statistic: 170.1 on 9 and 500 DF,  p-value: < 0.00000000000000022

##Prediction
COUPON<- 1.202
NEW <- 3
HI <- 4442.141
S_INCOME<-28760
E_INCOME<-27664
S_POP<-4557004
E_POP<-3195503
DISTANCE<-1976
PAX<-12782
FARE<-0
```

```
VAC_DUM<-0
SW_DUM<-0
SLOT_DUM<-0
GATE_DUM<-0

PV_VALUE.2 <- data.frame(COUPON,NEW,HI,S_INCOME,E_INCOME,S_POP,E_POP,DISTANCE
,PAX,FARE,VAC_DUM,SW_DUM,SLOT_DUM,GATE_DUM)
pred_v_exh3 <-predict(exh.lm.exhaust.2,PV_VALUE.2)
pred_v_exh3

##        1
## 252.3735

##Compare Predictive Accuracy
## Exhaustive
pred_v_exh4 <-predict(exh.lm.exhaust.2,valid_reduce)
accuracy(pred_v_exh4, valid_reduce$FARE)

##                 ME     RMSE      MAE       MPE     MAPE
## Test set -2.987695 35.76052 28.48067 -7.434022 22.42035

##Lift Charts - Exh
library(gains)
gain.num <- gains(valid_reduce$FARE, pred_v_exh4, groups=10)
plot(gain.num$depth, gain.num$mean.resp, xlab= "Decile", ylab = "Mean Respons
e", main = "(Exhaustive X) Decile vs. Mean Response",type="l")
```




```
##Barplot - Exh
barplot(gain.num$mean.resp/mean(valid_reduce$FARE), names.arg=gain.num$depth,
xlab = "Decile",ylab="Mean Response",main="(Exhaustive X) Decile-Wise Lift Ch
art")
```


(Exhaustive X) Decile-Wise Lift Chart
Mean Response
0.0
0.5
1.0
1.5
9
20
30
40
50
59
70
80
90
Decile