**Assignment 2:**

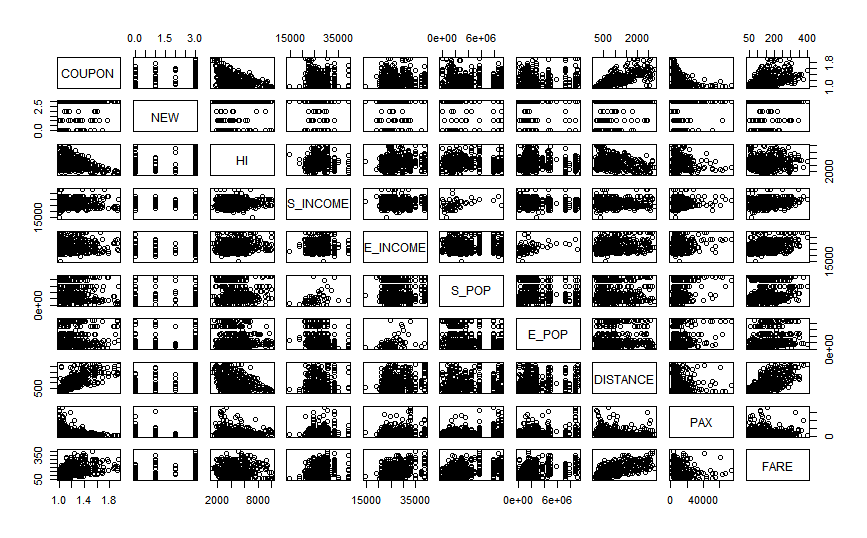
a. Explore the numerical predictors and response (FARE) by creating a correlation table and examining some scatterplots between FARE and those predictors. What seems to be the best single predictor of FARE?

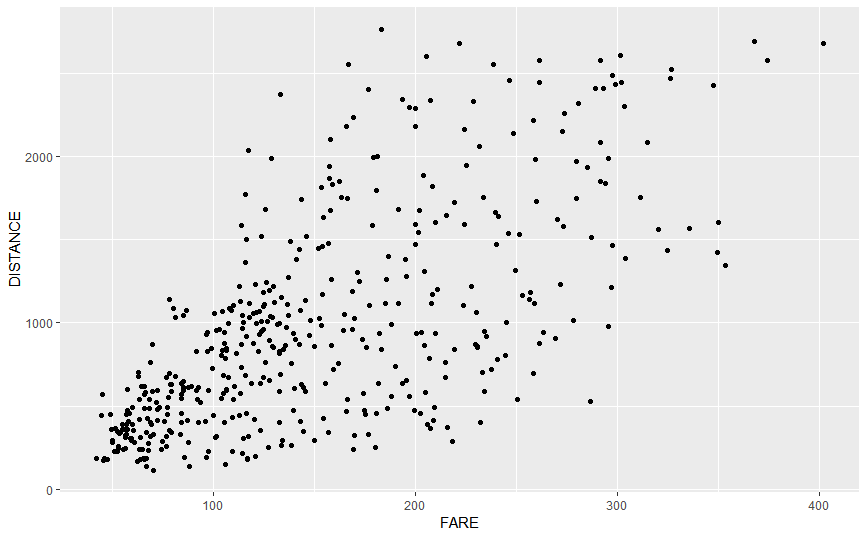
Correlation:

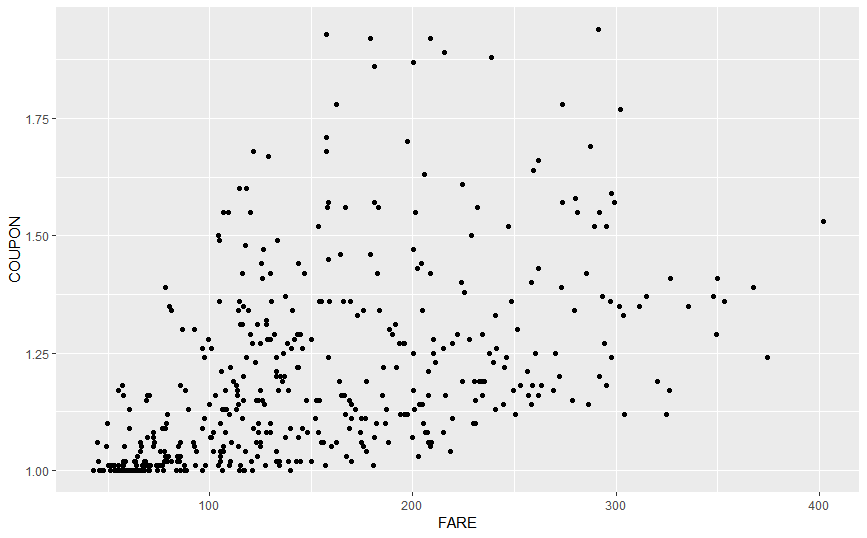
> cor(dataa$FARE,dataa)

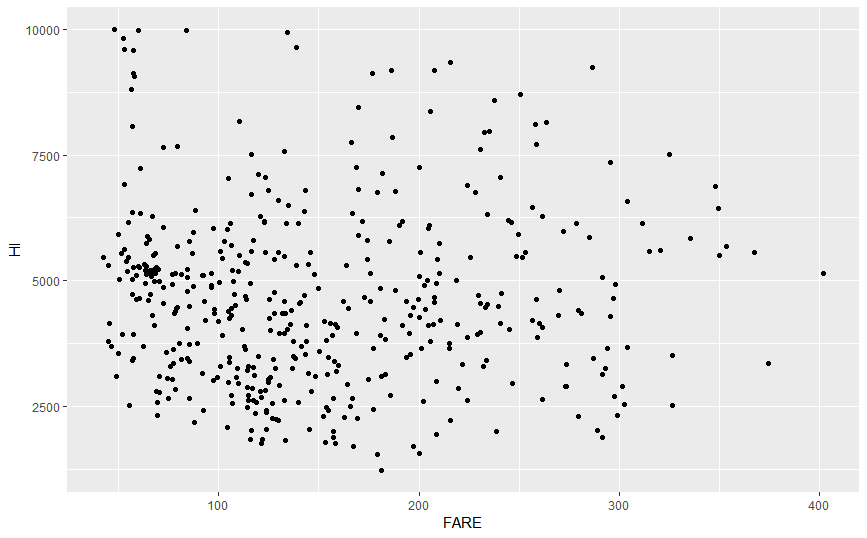


Scatter plot between FARE and predictors:









From the correlation table and scatter plots we can observe that **DISTANCE** seems to be the best single predictor for the FARE

b. Explore the categorical predictors excluding the first four (S\_CODE, S\_CITY, E\_CODE, E\_CITY) by computing the mean value of FARE according to each category. Which categorical predictor seems best for predicting FARE?

Computating the mean value of FARE according to each category:

> summaryVacation

VACATION FARE

1 No 173.5525

2 Yes 125.9809

> summarySW

SW FARE

1 No 188.18279

2 Yes 98.38227

> summarySLOT

SLOT FARE

1 Controlled 186.0594

2 Free 150.8257

> summaryGATE

GATE FARE

1 Constrained 193.129

2 Free 153.096

**SW** seems to be best categorical predictor for the FARE

c. Partition the data into training (60%) and validation (40%) sets (the random seed should be set at **value of 12345**). Build a simple model that involves only the two predictors identified in parts (a) and (b), and report the regression model results. After running the model, generate the plot of residual vs. predicted values and explain whether it violates any of the residual assumptions.

> summary(regressor)

Call:

lm(formula = FARE ~ DISTANCE + SW, data = training\_set)

Residuals:

Min 1Q Median 3Q Max

-140.191 -28.939 -3.925 28.178 128.526

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 112.427592 4.922178 22.84 <2e-16 \*\*\*

DISTANCE 0.067923 0.003804 17.85 <2e-16 \*\*\*

SWYes -63.109943 5.452649 -11.57 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.5 on 379 degrees of freedom

Multiple R-squared: 0.5887, Adjusted R-squared: 0.5865

F-statistic: 271.3 on 2 and 379 DF, p-value: < 2.2e-16

> regressor

Call:

lm(formula = FARE ~ DISTANCE + SW, data = training\_set)

Coefficients:

(Intercept) DISTANCE SWYes

112.42759 0.06792 -63.10994

> BIC(regressor)

[1] 4054.431

> fitsummary = summary(regressor)

> fitsummary$r.squared

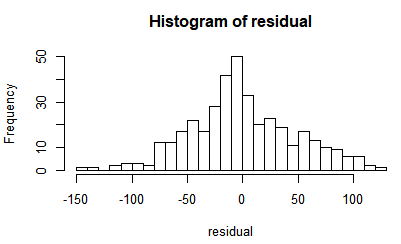
[1] 0.5887181

> fitsummary$adj.r.squared

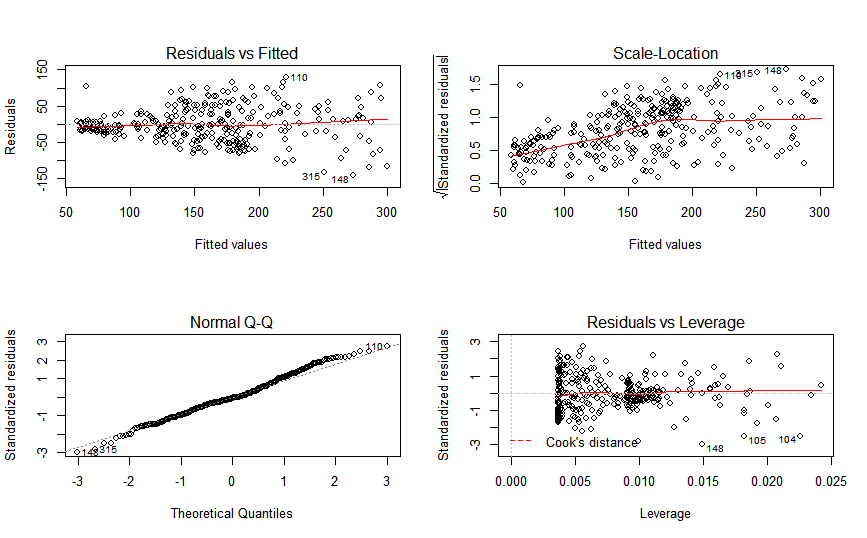
[1] 0.5865477

**BIC of the model is 4054.431**

**R squared values of the model is 0.5887**



Residual Plot:



From the above plot we can observe that residual vs fitted displays funnel pattern, which violates **constant variance** assumption.

MSE in validation set:

> mean((validation\_set[,"FARE"] - PredBase$fit)^2)

[1] 2361.868

**MSE in validation set is 2361.868**

d. Assume that all variables are available to you now. Use backward selection method to build a regression model. You can ignore the first four predictors in generating the candidate models.

> coefficients(backward)

(Intercept) VACATIONYes SWYes HI S\_INCOME E\_INCOME S\_POP E\_POP

2.214768e+01 -3.826410e+01 -4.016140e+01 8.464879e-03 1.340116e-03 7.150514e-04 2.606848e-06 4.685540e-06

SLOTFree GATEFree DISTANCE PAX

-1.345358e+01 -2.129136e+01 7.701913e-02 -7.814457e-04

> summary(backward)

Call:

lm(formula = FARE ~ VACATION + SW + HI + S\_INCOME + E\_INCOME +

S\_POP + E\_POP + SLOT + GATE + DISTANCE + PAX, data = training\_set)

Residuals:

Min 1Q Median 3Q Max

-102.023 -21.742 -1.107 19.377 103.178

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.215e+01 2.588e+01 0.856 0.39263

VACATIONYes -3.826e+01 4.547e+00 -8.416 8.65e-16 \*\*\*

SWYes -4.016e+01 4.685e+00 -8.572 2.80e-16 \*\*\*

HI 8.465e-03 1.187e-03 7.131 5.26e-12 \*\*\*

S\_INCOME 1.340e-03 6.324e-04 2.119 0.03474 \*

E\_INCOME 7.151e-04 4.759e-04 1.502 0.13385

S\_POP 2.607e-06 8.168e-07 3.192 0.00154 \*\*

E\_POP 4.686e-06 9.680e-07 4.840 1.91e-06 \*\*\*

SLOTFree -1.345e+01 4.738e+00 -2.839 0.00477 \*\*

GATEFree -2.129e+01 4.922e+00 -4.325 1.96e-05 \*\*\*

DISTANCE 7.702e-02 3.135e-03 24.565 < 2e-16 \*\*\*

PAX -7.814e-04 1.653e-04 -4.727 3.25e-06 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.55 on 370 degrees of freedom

Multiple R-squared: 0.7876, Adjusted R-squared: 0.7812

F-statistic: 124.7 on 11 and 370 DF, p-value: < 2.2e-16

> BIC(backward)

[1] 3855.601

> fitsummaryback = summary(backward)

> fitsummaryback$r.squared

[1] 0.7875503

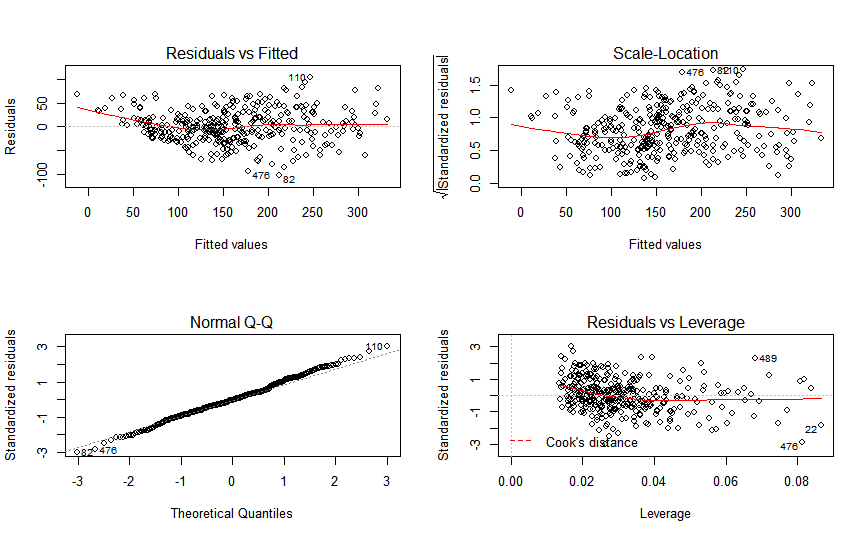
> fitsummaryback$adj.r.squared

[1] 0.7812342

**BIC of the model is 3855.601**

**R squared values of the model is 0.7875**

Residual Plot:



It violates constant variance violation as well.

MSE in validation set:

> mean((validation\_set[,"FARE"] - PredBaseback$fit)^2)

[1] 1386.321

**MSE in validation set is 1386.321**

e. Compare the performance of models developed in parts (c) and (d) with in terms of the R-squared value, BIC value, and MSE measure. Which model is the better? Please explain briefly.

For the model developed in part c

* R-squared value is 0.58
* BIC Value is 4054
* MSE measure is 2361

For the model developed in part d

* R-squared value is 0.78
* BIC Value is 3855
* MSE measure is 1386.31

By observing the above values of both the models we can conclude that backward (part d) model is better as BIC value and MSE value of backward model is less than the values of part c model. And R-squared value is 0.78 whereas for part c it is 0.58. Model with R squared value closer to one is the better fit model.

**APPENDIX**

**R code:**

# Set working directory to working file location

setwd("C:/Users/avala/OneDrive/Desktop/SPRING19/OR568/Assignments/Assignment2")

# Read the csv file

dataset <- read.csv("AirfaresData.csv")

summary(dataset)

dataa <- dataset[, c(5,6,9:13,16:18)]

datab <- dataset[,c(7,8,14,15,18)]

df <- dataset[,c(5:18)]

summary(dataa)

summary(datab)

summary(datad)

#EXPLORING NUMERICAL PREDICTORS

#Correlation table

cor(dataa$FARE,dataa)

?aggregate

#visualizing predictors and FARE through scatter ploy

ggplot(dataa,aes(x = FARE,y = DISTANCE)) +

geom\_point()

ggplot(dataa,aes(x = FARE,y = COUPON)) +

geom\_point()

ggplot(dataa,aes(x = FARE,y = HI)) +

geom\_point()

pairs(dataa)

#DISTANCE seems to be best single predictor for FARE

#EXPLORING CATEGORICAL PREDICTORS

#mean value of FARE for each predictor

summaryVacation <-aggregate(FARE~VACATION, data = datab, FUN = mean) #47.5716

summarySW <-aggregate(FARE~SW, data = datab, FUN = mean) #89.80052

summarySLOT <-aggregate(FARE~SLOT, data = datab, FUN = mean) #35.2337

summaryGATE <-aggregate(FARE~GATE, data = datab, FUN = mean) #40.033

summaryVacation

summarySW

summarySLOT

summaryGATE

#SW seems best categorical predictor for predicting FARE as it has the largest difference in mean FARE values

#Splitting the data into training and Validation set

set.seed(12345)

library(caTools)

split = sample.split(df$FARE, SplitRatio = 0.6)

training\_set = subset(df, split == TRUE)

validation\_set = subset(df, split == FALSE)

#Regression model with first two predictors

regressor = lm(formula = FARE ~ DISTANCE + SW,

data = training\_set)

regressor

summary(regressor)

AIC(regressor)

BIC(regressor)

fitsummary = summary(regressor)

fitsummary$r.squared

fitsummary$adj.r.squared

residual<-residuals(regressor)

hist(residual,breaks=20)

layout(matrix(c(1,2,3,4),2,2)) # optional 4 graphs/page

plot(regressor)

# Computing the MSE on validation dataset based on model fit with training data

# We use the "predict" function to compute the predicted value on validation set

PredBase<-predict(regressor, validation\_set, se.fit=TRUE)

PredBase

mean((validation\_set[,"FARE"] - PredBase$fit)^2)

#Regression model with backward selection

backregressor = lm(formula = FARE ~ .,

data = training\_set)

backward<-step(backregressor, direction='backward')

coefficients(backward)

summary(backward)

AIC(backward)

BIC(backward)

fitsummaryback = summary(backward)

fitsummaryback$r.squared

fitsummaryback$adj.r.squared

residualback<-residuals(backward)

hist(residualback,breaks=20)

layout(matrix(c(1,2,3,4),2,2))

plot(backward)

# Computing the MSE on validation dataset based on model fit with training data

# We use the "predict" function to compute the predicted value on validation set

PredBaseback<-predict(backward, validation\_set, se.fit=TRUE)

PredBaseback

mean((validation\_set[,"FARE"] - PredBaseback$fit)^2)