########################

###NBA 2016-2017 DATA###

#######PROJECT 2########

########################

library(ggplot2)

library(plyr)

library(tseries)

library(readxl)

getwd()

NBA <- read\_excel("TIDYNBA.xlsx")

View(NBA)

head(NBA)

tail(NBA)

summary(NBA)

#VARIABLES NON-GAME STATS & HISTOGRAMS

AGE <- NBA$AGE

hist(AGE)

salary<- NBA$SALARY\_MILLIONS

hist(salary)

twitter\_count <- NBA$TWITTER\_FOLLOWER\_COUNT\_MILLIONS

hist(twitter\_count)

#VARIABLES GAME STATS & HISTOGRAMS

Win\_pct<- NBA$W\_PCT

hist(Win)

Offensive <- NBA$OFF\_RATING\_RANK

hist(Offensive)

FGA<-NBA$FGA

hist(FGA)

FGM<-NBA$FGM

hist(FGM)

PPG<-NBA$PTS

hist(PPG)

#RELATIONSHIP BETWEEN FOLLOWER COUNT & SALARY#

ggplot(NBA, aes(x = salary, y = twitter\_count)) +

geom\_point()

#RELATIONSHIP BETWEEN FOLLOWER COUNT & AGE#

ggplot(NBA, aes(x = AGE, y = twitter\_count)) +

geom\_point()

#RELATIONSHIP BETWEEN FOLLOWER COUNT & WIN PCT#

ggplot(NBA, aes(x = Win\_pct, y = twitter\_count)) +

geom\_point()

#######################################

########PARTITIONING THE DATA##########

#######################################

#70% of sample to be used for training#

p<-.7

#number of observations (rows) in the NBA data frame#

obs\_count<-dim(NBA)[1]

#number of observations to be selected for the training partition#

#floor() rounds down to the nearest integer#

training\_size <- floor(p \* obs\_count)

training\_size

#set the seed to make your partition reproducible

set.seed(1234)

#create a vector with the shuffled row numbers of the original dataset

train\_ind <- sample(obs\_count, size = training\_size)

Training <- NBA[train\_ind, ] #pulls random rows for training

Testing <- NBA[-train\_ind, ] #pulls random rows for testing

#CHECKING THE DIMENSIONS OF THE PARTITIONED DATA#

dim(Training)

dim(Testing)

#PLOTTING THE TRAINING AND TESTING PARTITIONS (Have to create variables first)

plot(twitter\_count ~ salary, NBA, xlim=c(0,30),ylim=c(0,30)) #PLOT ENTIRE DATASET

plot(twitter\_count ~ salary, Training, xlim=c(0,30),ylim=c(0,30), col ='blue') #PLOTS THE IN-SAMPLE TRAINING PARTITION

plot(twitter\_count ~ salary, Testing, xlim=c(0,30),ylim=c(0,30), col ='red', pch=3) #PLOTS THE OUT-OF-SAMPLE TESTING PARTITION

points(Training$SALARY\_MILLIONS, Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS, col='blue') #PLOTS THE OUT-OF-SAMPLE TESTING PARTITION

points(Testing$SALARY\_MILLIONS, Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS, col='red', pch=3) #PLOTS THE OUT-OF-SAMPLE TESTING PARTITION

##LINEAR REGRESSION twitter\_count = B0 + B1\*displ+u

M1<-lm(twitter\_count ~ salary, Training)

summary(M1) #SUMMARY OUTPUT OF THE MODEL OBJECT INCLUDING KEY VALUES TO PROVE SIGNIFICANCE

M1$coefficients #RETURNS BETA ESTIMATES

M1$residuals #RETURNS RESIDUALS WE CAN SEE HOW MUCH THE MODEL UNDERESTIMATED OR OVERESTIMATED

M1$fitted.values #RETURNS FITTED (PREDICTED) VALUES

#ARE THE RESIDUALS NORMAL?

hist(M1$residuals) #PLOT OF REDIDUALS

jarque.bera.test(M1$residuals) #TEST FOR NORMAL DISTRIBUTION

#GENERATING PREDICTIONS ON THE TRAINING DATA

PRED\_1\_IN <- predict(M1, Training) #generate predictions on the (in-sample) training data

View(PRED\_1\_IN)

View(M1$fitted.values) #these are the same as the fitted values

#GENERATING PREDICTIONS ON THE TEST DATA FOR BENCHMARKING

PRED\_1\_OUT <- predict(M1, Testing) #generate predictions on the (out-of-sample) testing data

#COMPUTING IN-SAMPLE AND OUT-OF-SAMPLE ROOT MEAN SQUARED ERROR

RMSE\_1\_IN<-sqrt(sum((PRED\_1\_IN-Training$SALARY\_MILLIONS)^2)/length(PRED\_1\_IN)) #computes in-sample error

RMSE\_1\_OUT<-sqrt(sum((PRED\_1\_OUT-Testing$SALARY\_MILLIONS)^2)/length(PRED\_1\_OUT)) #computes out-of-sample

RMSE\_1\_IN #IN-SAMPLE ERROR

RMSE\_1\_OUT #OUT-OF-SAMPLE ERROR

#PLOTTING THE MODEL IN 2D AGAINST BOTH DATA PARTITIONS

x\_grid <- seq(0,35,.1) #CREATES GRID OF X-AXIS VALUES

predictions <- predict(M1, list(salary=x\_grid))

plot(Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Training$SALARY\_MILLIONS, col='blue')

lines(x\_grid, predictions, col='green', lwd=3)

points(Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Testing$SALARY\_MILLIONS, col='red', pch=3)

salary2<-NBA$SALARY\_MILLIONS^2 #QUADRATIC TRANSFORMATION (2nd ORDER)

#BUILDING THE QUADRATIC MODEL FROM THE TRAINING DATA

M2 <- lm(twitter\_count ~ salary + salary2, Training)

summary(M2) #generates summary diagnostic output

M2$coefficients #RETURNS BETA ESTIMATES

M2$residuals #RETURNS RESIDUALS WE CAN SEE HOW MUCH THE MODEL UNDERESTIMATED OR OVERESTIMATED

M2$fitted.values #RETURNS FITTED (PREDICTED) VALUES

#ARE THE RESIDUALS NORMAL?

hist(M2$residuals) #PLOT OF REDIDUALS

jarque.bera.test(M2$residuals) #TEST FOR NORMAL DISTRIBUTION

#GENERATING PREDICTIONS ON THE TRAINING DATA

PRED\_2\_IN <- predict(M2, Training) #generate predictions on the (in-sample) training data

View(PRED\_2\_IN)

View(M2$fitted.values) #these are the same as the fitted values

#GENERATING PREDICTIONS ON THE TEST DATA FOR BENCHMARKING

PRED\_2\_OUT <- predict(M2, Testing) #generate predictions on the (out-of-sample) testing data

#COMPUTING IN-SAMPLE AND OUT-OF-SAMPLE ROOT MEAN SQUARED ERROR

RMSE\_2\_IN<-sqrt(sum((PRED\_2\_IN-Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS)^2)/length(PRED\_2\_IN)) #computes in-sample error

RMSE\_2\_OUT<-sqrt(sum((PRED\_2\_OUT-Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS)^2)/length(PRED\_2\_OUT)) #computes out-of-sample

RMSE\_2\_IN #IN-SAMPLE ERROR

RMSE\_2\_OUT #OUT-OF-SAMPLE ERROR

#PLOTTING THE MODEL IN 2D AGAINST BOTH DATA PARTITIONS

x\_grid <- seq(0,35,.1) #CREATES GRID OF X-AXIS VALUES

predictions <- predict(M2, list(salary=x\_grid, salary2=x\_grid^2))

plot(Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Training$SALARY\_MILLIONS, col='blue')

lines(x\_grid, predictions, col='green', lwd=3)

points(Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Testing$SALARY\_MILLIONS, col='red', pch=3)

#BUILDING A MULTIPLE REGRESSION MODEL FROM THE TRAINING DATA

M3<-lm(twitter\_count ~ PPG + Offensive, Training)

summary(M3) #generates summary diagnostic output

M3$coefficients #RETURNS BETA ESTIMATES

M3$residuals #RETURNS RESIDUALS WE CAN SEE HOW MUCH THE MODEL UNDERESTIMATED OR OVERESTIMATED

M3$fitted.values #RETURNS FITTED (PREDICTED) VALUES

#ARE THE RESIDUALS NORMAL?

hist(M3$residuals) #PLOT OF REDIDUALS

jarque.bera.test(M3$residuals) #TEST FOR NORMAL DISTRIBUTION

#GENERATING PREDICTIONS ON THE TRAINING DATA

PRED\_3\_IN <- predict(M3, Training) #generate predictions on the (in-sample) training data

View(PRED\_3\_IN)

View(M3$fitted.values) #these are the same as the fitted values

#GENERATING PREDICTIONS ON THE TEST DATA FOR BENCHMARKING

PRED\_3\_OUT <- predict(M3, Testing) #generate predictions on the (out-of-sample) testing data

#COMPUTING IN-SAMPLE AND OUT-OF-SAMPLE ROOT MEAN SQUARED ERROR

RMSE\_3\_IN<-sqrt(sum((PRED\_3\_IN-Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS)^2)/length(PRED\_3\_IN)) #computes in-sample error

RMSE\_3\_OUT<-sqrt(sum((PRED\_3\_OUT-Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS)^2)/length(PRED\_3\_OUT)) #computes out-of-sample

RMSE\_3\_IN #IN-SAMPLE ERROR

RMSE\_3\_OUT #OUT-OF-SAMPLE ERROR

#CANNOT PLOTTING THE MODEL IN 2D AGAINST BOTH DATA PARTITIONS BECAUSE OF MULTIPLE RIGHT HAND SIDE VARIABLES

x\_grid <- seq(0,35,.1) #CREATES GRID OF X-AXIS VALUES

predictions <- predict(M3, list(PPG=x\_grid, Offensive=x\_grid^2))

plot(Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Training$PTS, col='blue')

lines(x\_grid, predictions, col='green', lwd=3)

points(Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Testing$PTS, col='red', pch=3)

#CREATING NATURAL LOG OD SALARY

ln\_salary<- log(NBA$SALARY\_MILLIONS)

#BUILDING THE LOGARITHMIC MODEL FROM THE TRAINING DATA

M4 <- lm(twitter\_count ~ ln\_salary, Training)

summary(M4) #generates summary diagnostic output

M4$coefficients #RETURNS BETA ESTIMATES

M4$residuals #RETURNS RESIDUALS WE CAN SEE HOW MUCH THE MODEL UNDERESTIMATED OR OVERESTIMATED

M4$fitted.values #RETURNS FITTED (PREDICTED) VALUES

#ARE THE RESIDUALS NORMAL?

hist(M4$residuals) #PLOT OF REDIDUALS

jarque.bera.test(M4$residuals) #TEST FOR NORMAL DISTRIBUTION

#GENERATING PREDICTIONS ON THE TRAINING DATA

PRED\_4\_IN <- predict(M4, Training) #generate predictions on the (in-sample) training data

View(PRED\_4\_IN)

View(M4$fitted.values) #these are the same as the fitted values

#GENERATING PREDICTIONS ON THE TEST DATA FOR BENCHMARKING

PRED\_4\_OUT <- predict(M4, Testing) #generate predictions on the (out-of-sample) testing data

#COMPUTING IN-SAMPLE AND OUT-OF-SAMPLE ROOT MEAN SQUARED ERROR

RMSE\_4\_IN<-sqrt(sum((PRED\_4\_IN-Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS)^2)/length(PRED\_4\_IN)) #computes in-sample error

RMSE\_4\_OUT<-sqrt(sum((PRED\_4\_OUT-Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS)^2)/length(PRED\_4\_OUT)) #computes out-of-sample

RMSE\_4\_IN #IN-SAMPLE ERROR

RMSE\_4\_OUT #OUT-OF-SAMPLE ERROR

#PLOTTING THE MODEL IN 2D AGAINST BOTH DATA PARTITIONS

x\_grid <- seq(0,35,.1) #CREATES GRID OF X-AXIS VALUES

predictions <- predict(M4, list(ln\_salary=log(x\_grid)))

plot(Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Training$SALARY\_MILLIONS, col='blue')

lines(x\_grid, predictions, col='green', lwd=3)

points(Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Testing$SALARY\_MILLIONS, col='red', pch=3)

######################################

###########MODEL COMPARISON###########

######################################

#COMPARISON OF IN-SAMPLE MODEL PERFORMANCE BY RMSE

RMSE\_1\_IN #MODEL WITH ONLY LINEAR TERM

RMSE\_2\_IN #MODEL WITH LINEAR AND QUADRATIC TERM

RMSE\_3\_IN #MODEL WITH TWO LINEAR TERMS

RMSE\_4\_IN #LOGARITHMIC MODEL

(RMSE\_1\_IN, RMSE\_2\_IN, RMSE\_3\_IN, RMSE\_4\_IN)

#COMPARISON OF OUT-OF-SAMPLE MODEL PERFORMANCE BY RMSE

RMSE\_1\_OUT #MODEL WITH ONLY LINEAR TERM

RMSE\_2\_OUT #MODEL WITH LINEAR AND QUADRATIC TERM

RMSE\_3\_OUT #MODEL WITH TWO LINEAR TERMS

RMSE\_4\_OUT #LOGARITHMIC MODEL

(RMSE\_1\_OUT, RMSE\_2\_OUT, RMSE\_3\_OUT, RMSE\_4\_OUT)

########################################################

###PLOTTING THE REGRESSION MODELS AGAINST ONE ANOTHER###

########################################################

x\_grid <- seq(0,35,.1) #CREATES GRID OF X-AXIS VALUES

plot(Training$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Training$SALARY\_MILLIONS, col='blue')

predictions\_1 <- predict(M1, list(salary=x\_grid))

predictions\_2 <- predict(M2, list(salary=x\_grid, salary2=x\_grid^2))

predictions\_3 <- predict(M3, list(PPG=x\_grid, Offensive=x\_grid))

predictions\_4 <- predict(M4, list(ln\_salary=log(x\_grid)))

lines(x\_grid, predictions\_1, col='blue', lwd=3) #PLOTS M1

lines(x\_grid, predictions\_2, col='green', lwd=3) #PLOTS M2

lines(x\_grid, predictions\_3, col='purple', lwd=3) #PLOTS M3

lines(x\_grid, predictions\_4, col='orange', lwd=3) #PLOTS M4

points(Testing$TWITTER\_FOLLOWER\_COUNT\_MILLIONS ~ Testing$SALARY\_MILLIONS, col='red', pch=3)