ddata

4/18/2021

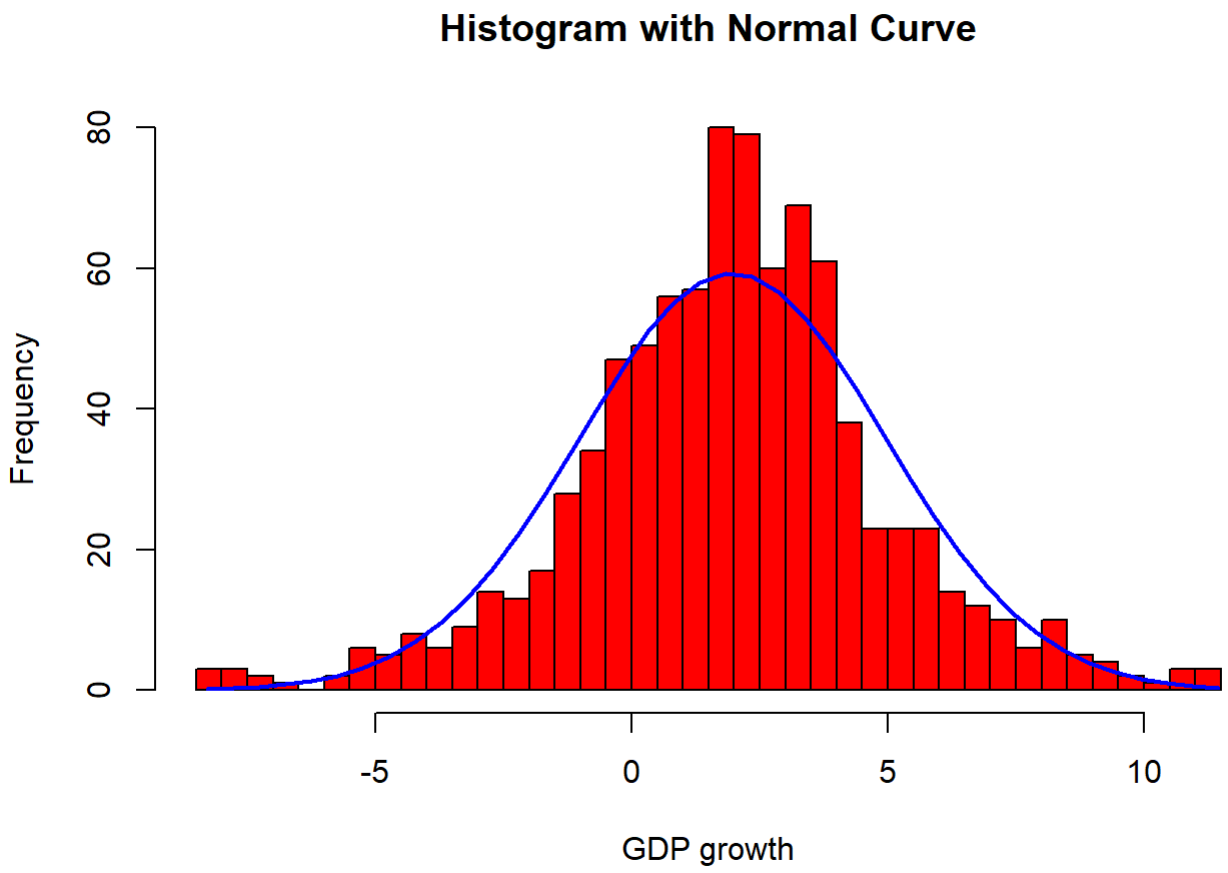
First I will read in my dataset, and choose some variables to run a linear model on.

```
setwd("C:/Users/sheri/Desktop")
df<-read.csv("data.csv")


library(tidyverse)
library(randomForest)
library(nnet)
library(NeuralNetTools)
library(e1071)

la_and_c = df %>%
 select(countryname, year,
         output_gap = IMF_NGAP_NPGDP,
         unemployment= IMF_LUR,
        savings= IMF_NGSD_NGDP,
         real_gdp_growth = WB_ny_adj_nnty_pc_kd_zg)

newdata  = na.omit(la_and_c)
summary(newdata)
```

```
##  countryname             year        output_gap        unemployment
##  Length:906         Min.   :1980   Min.   :-15.8100   Min.   : 1.256
##  Class :character   1st Qu.:1990   1st Qu.: -1.7352   1st Qu.: 4.710
##  Mode  :character   Median :2000   Median : -0.4370   Median : 6.883
##                     Mean   :2000   Mean   : -0.3726   Mean   : 7.534
##                     3rd Qu.:2009   3rd Qu.:  0.9825   3rd Qu.: 9.090
##                     Max.   :2017   Max.   : 11.9180   Max.   :27.475
##     savings       real_gdp_growth
##  Min.   : 5.138   Min.   :-27.3230
##  1st Qu.:19.884   1st Qu.:  0.2364
##  Median :22.901   Median :  2.0124
##  Mean   :23.346   Mean   :  1.9386
##  3rd Qu.:26.420   3rd Qu.:  3.6564
##  Max.   :41.765   Max.   : 17.4898
```

```
cleandata <- subset(newdata,!(newdata$real_gdp_growth > quantile(newdata$real_gdp_growth, probs=
c(.01, .99))[2] | newdata$real_gdp_growth < quantile(newdata$real_gdp_growth, probs=c(.01, .99))
[1]) )
```

```
# Histogram + Normal Curve
x <- cleandata$real_gdp_growth
h<-hist(x, breaks=40, col="red", xlab="GDP growth",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```



#Now that I chose the predictors I will split my data into a test and training data set, refit the model using *only* the train set. Then make predictions for the test set. Then I will compute the residuals (predictions vs actual values).

```
set.seed(7)

n <- dim(cleandata )[1]
train_ind <- runif(n) < 0.75
df_train <- cleandata [ train_ind, ]
df_test <- cleandata [ !train_ind, ]


lm_training<- lm(real_gdp_growth ~  unemployment+savings+output_gap, data = df_train)
summary(lm_training)
```

```
##
## Call:
## lm(formula = real_gdp_growth ~ unemployment + savings + output_gap,
##     data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5325  -1.5736   0.0363   1.7317   8.8146
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.84093    0.67737  -4.194 3.12e-05 ***
## unemployment  0.13275    0.03115   4.261 2.33e-05 ***
## savings       0.16773    0.02277   7.365 5.38e-13 ***
## output_gap    0.20497    0.04511   4.543 6.60e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.811 on 651 degrees of freedom
## Multiple R-squared:  0.1084, Adjusted R-squared:  0.1043
## F-statistic: 26.39 on 3 and 651 DF,  p-value: 4.083e-16
```

The p value is larger than the alpha value of 0.05 for engineers, electricity, secureservers, unemployment, and transportation. They are not great predictors of GDP. I dropped them from the linear model. The linear model shows a growth of 1.68. I will try a random forest.

```
ypred <- predict(lm_training, df_test)
actual_vs_pred <- data.frame(cbind(actuals=df_test$real_gdp, predicteds=ypred))

summary(actual_vs_pred)
```

```
##     actuals           predicteds
##  Min.   :-8.28331   Min.   :-0.9477
##  1st Qu.:-0.01503   1st Qu.: 1.1866
##  Median : 1.62877   Median : 1.8102
##  Mean   : 1.69806   Mean   : 1.8503
##  3rd Qu.: 3.23044   3rd Qu.: 2.4130
##  Max.   :10.57584   Max.   : 4.4035
```

```
Canada = df %>%
 select(countryname, year,
         output_gap = IMF_NGAP_NPGDP,
         unemployment= IMF_LUR,
        savings= IMF_NGSD_NGDP,
         real_gdp_growth = WB_ny_adj_nnty_pc_kd_zg) %>%

  filter(countryname == "Canada", year == "1999")
Canada  = na.omit(Canada)

CanadaGrowth = predict(lm_training, newdata = Canada)

CanadaGrowth
```

```
##       1
## 1.68439
```

```
library(randomForest)

rf01 = randomForest(formula = real_gdp_growth ~  unemployment+savings+output_gap, data = df_trai
n,ntree = 100, type = "classification")


rf01_predicted= predict(rf01, df_test)
rf01_vs_pred <- data.frame(cbind(actuals=df_test$real_gdp, predicteds=rf01_predicted))

summary(rf01_vs_pred)
```

```
##     actuals           predicteds
##  Min.   :-8.28331   Min.   :-4.051
##  1st Qu.:-0.01503   1st Qu.: 1.144
##  Median : 1.62877   Median : 1.857
##  Mean   : 1.69806   Mean   : 1.854
##  3rd Qu.: 3.23044   3rd Qu.: 2.597
##  Max.   :10.57584   Max.   : 5.403
```

```
Canada = df %>%
 select(countryname, year,
         output_gap = IMF_NGAP_NPGDP,
         unemployment= IMF_LUR,
        savings= IMF_NGSD_NGDP,
         real_gdp_growth = WB_ny_adj_nnty_pc_kd_zg) %>%

  filter(countryname == "Canada", year == "1999")
Canada  = na.omit(Canada)

CanadaGrowth = predict(rf01, newdata = Canada)

CanadaGrowth
```

```
##        1
## 3.188479
```

The value for GDP growth is 3.5 The random forest has a value of 3.18 for growth and the linear model is 1.68439. I will try a decision tree using the c.5 decision tree model.

```
library(C50)
df_train$real_gdp_growth <- factor(df_train$real_gdp_growth)
C5 <- C5.0(real_gdp_growth ~  unemployment+savings+output_gap, data = df_train)

C5_predicted= predict(C5, df_test)
c5_vs_pred <- data.frame(cbind(actuals=df_test$real_gdp, predicteds=C5_predicted))

summary(c5_vs_pred)
```

```
##     actuals           predicteds
##  Min.   :-8.28331   Min.   :  5.0
##  1st Qu.:-0.01503   1st Qu.: 59.5
##  Median : 1.62877   Median :174.0
##  Mean   : 1.69806   Mean   :180.0
##  3rd Qu.: 3.23044   3rd Qu.:263.5
##  Max.   :10.57584   Max.   :579.0
```

```
Canada = df %>%
 select(countryname, year,
         output_gap = IMF_NGAP_NPGDP,
         unemployment= IMF_LUR,
        savings= IMF_NGSD_NGDP,
         real_gdp_growth = WB_ny_adj_nnty_pc_kd_zg) %>%

  filter(countryname == "Canada", year == "1999")
Canada  = na.omit(Canada)

CanadaGrowth = predict(C5, newdata = Canada)

print (CanadaGrowth)
```

```
## [1] 0.846212355
## 655 Levels: -8.280972472 -8.157278401 -7.769200962 -7.338225981 ... 11.4509236
```

The c5 tree predicted a growth of 0.8. The random forest model has the best results for GDP growth.