title: "EDA_VII"
author: "David Hunte"
date: "July 13, 2018"
output:
  word_document: default
  pdf_document: default
  html_document: default

```
library(data.table)
library(TTR)
library(dplyr)
library(imputeTS)
library(zoo)
library(quantmod)
library(MeanShift)
library(ggfortify)
library(ggplot2)
library(h2o)
library(gapminder)
require(reshape2)

memory.limit()
memory.limit(size=90000)
```

#S&P500 Stock Data Import

```
AllStocksData <- read_csv("E:/AllStocksData.csv",
      col_types = cols(Close = col_number(),
      Date = col_date(format = "%m/%d/%Y"),
      High = col_number(), Low = col_number(),
      Volume = col_number()))


AllStocksData <- as.data.table(AllStocksData)
AllStocksData <- na.omit(AllStocksData)
AllStocksData
```

#Dataframe setup

```
df.stocks<-  data.frame(Date = AllStocksData$Date,
                        Open = AllStocksData$Open,
                        Close = AllStocksData$Close,
                        High = AllStocksData$High,
                        Low = AllStocksData$Low,
                        Volume = AllStocksData$Volume,
                        OpenInt = AllStocksData$OpenInt)
```

Feature #1: All monthly moving averages

```
df.stocks1 <- df.stocks %>% group_by(Months = months(as.Date(Date,"%a"))) %>%
            mutate(Open.Mvn.Avg = runMean(Open, 1),
                Close.Mvn..Avg= runMean(Close, 1),
                High.Mvn.Avg = runMean(High, 1),
                Low.Mvn.Avg = runMean(Low, 1),
                Volume.Mvn.Avg= runMean(Volume, 1))%>%
            dplyr::select(-Open, -High, -Low, -Volume, -OpenInt, -Date,
            -Close)
df.stocks1
```

#Feature 2: Monthly log returns

```
df.stocks2 <- df.stocks %>%
        group_by(Date = months(as.Date(Date,"%a")))%>%
        mutate("Monthly.Log.Returns" = (Close-Open)/Open-1)%>%
        dplyr::select(-OpenInt,-Volume, -High, -Low, -Open, -Close)
df.stocks2
```

#Feature 2: dataframe reshaped to wide columns for monthly log returns

```
df.stocks2.dcast <- dcast(df.stocks2, Close ~Date, value.var = "Monthly.Log.Returns")
df.stocks2.dcast[is.na(df.stocks2.dcast)]<- "0"
df.stocks2.dcast<- as.matrix(df.stocks2.dcast)
df.stocks2.dcast <- as.data.frame(df.stocks2.dcast)
df.stocks2.dcast
```

#Feature 3: Monthly log changes

```
df.stocks3 <- df.stocks %>%
        group_by(Date = months(as.Date(AllStocksData$Date, "%a")))%>%
        mutate("Monthly.log.change.Percent" = (Close-Open)/Open)%>%
        dplyr::select(-OpenInt, -Volume, -High, -Low, -Open, -Close)
df.stocks3
```

#Feature 3: dataframe reshaped to wide columns for monthly log changes

```
df.stocks3.dcast <- dcast(df.stocks3, Monthly.log.change.Percent ~Date, value.var =
"Monthly.log.change.Percent")
df.stocks3.dcast[is.na(df.stocks3.dcast)]<- "0"
df.stocks3.dcast<- as.matrix(df.stocks3.dcast)
df.stocks3.dcast <- as.data.frame(df.stocks3.dcast)
df.stocks3.dcast
```

Feature 4: Monthly Stock volume

```
Volume2 <- AllStocksData$Volume
df.stocks4 <- df.stocks %>%
        group_by(Date = months(as.Date(AllStocksData$Date,"%a")))%>%
        mutate("Monthly.Volume" = Volume/12) %>%
        dplyr::select(-OpenInt, -High, -Low, -Open, -Close, -Volume)
df.stocks4
```

#Feature 4: dataframe reshaped to wide columns for monthly volume changes

```
df.stocks4.dcast <- dcast(df.stocks4, Monthly.Volume ~Date, value.var = "Monthly.Volume")
df.stocks4.dcast[is.na(df.stocks4.dcast)]<- "0"
df.stocks4.dcast<- as.matrix(df.stocks4.dcast)
df.stocks4.dcast <- as.data.frame(df.stocks4.dcast)
df.stocks4.dcast
```

Feature 5: Monthly high's and low's

```
df.stocks5 <- df.stocks %>%
        group_by(Date = months(as.Date(AllStocksData$Date,"%a")))%>%
        mutate('Monthly.High' = High, 'Monthly.Low' = Low)%>%
        dplyr::select(-OpenInt, -Volume, -Open, -Close, -OpenInt, -High,
        -Low)
df.stocks5
```

#Feature 5: dataframe reshaped to wide columns for monthly high's

```
df.stocks5.dcast1 <- dcast(df.stocks5, Monthly.High ~ Date)
df.stocks5.dcast1[is.na(df.stocks5.dcast1)]<- "0"
df.stocks5.dcast1<- as.matrix(df.stocks5.dcast1)
df.stocks5.dcast1 <- as.data.frame(df.stocks5.dcast1)
df.stocks5.dcast1
```

#Feature 5: dataframe reshaped to wide columns for monthly Low's

```
df.stocks5.dcast2 <- dcast(df.stocks5, Monthly.Low~ Date)
df.stocks5.dcast2[is.na(df.stocks5.dcast2)]<- "0"
df.stocks5.dcast2<- as.matrix(df.stocks5.dcast2)
df.stocks5.dcast2 <- as.data.frame(df.stocks5.dcast2)
df.stocks5.dcast2
```

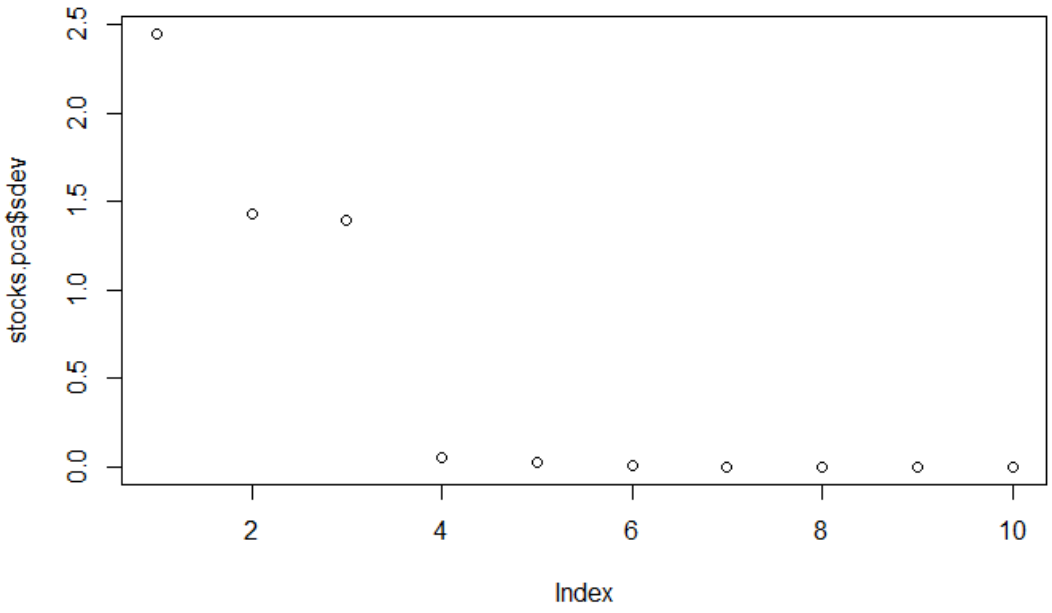
#PCA analysis: Dimension reduction

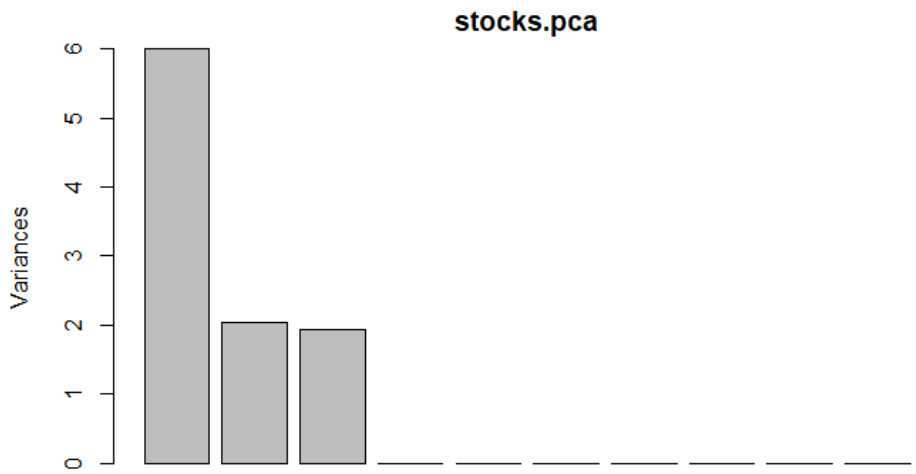
```
df.stocks.pca <- cbind(df.stocks5[, 2:3],df.stocks4[,2], df.stocks3[, 2],
            df.stocks2[,2], df.stocks1[,2:6])
df.stocks.pca

## 75% of the sample size
smp_size <- floor(0.002 * nrow(df.stocks.pca))
smp_size2 <- floor(0.75 * nrow(df.stocks.pca))

## set the seed to make your partition reproducible
set.seed(123)
test_ind <- sample(seq_len(nrow(df.stocks.pca)), size = smp_size)
train_ind <- sample(seq_len(nrow(df.stocks.pca)), size = smp_size2)


test <- df.stocks.pca[test_ind, ]
```

```
train <- df.stocks.pca[train_ind, ]


dim(train)
dim(test)
stocks.pca <- prcomp(test,
             center = TRUE,
             scale. = TRUE)
print(stocks.pca)

summary(stocks.pca)
```

#Plots of PCA components

```
plot(stocks.pca$sdev) #PC1-3 is more significant to go forward with

plot(stocks.pca)# confirm PC1-3
```





#Meanshift model data manipulations

```
set.seed(123)
dim(stocks.pca$x)
# mean and sd
stocks.data.sd<-scale(stocks.pca$x[,1:3])
stocks.data <-t(stocks.data.sd)
dim(stocks.data)


# MeanShift clustering (Option 1)
Stocks.Clusters<- msClustering( stocks.data, h=1.0 )
print(Stocks.Clusters)
```

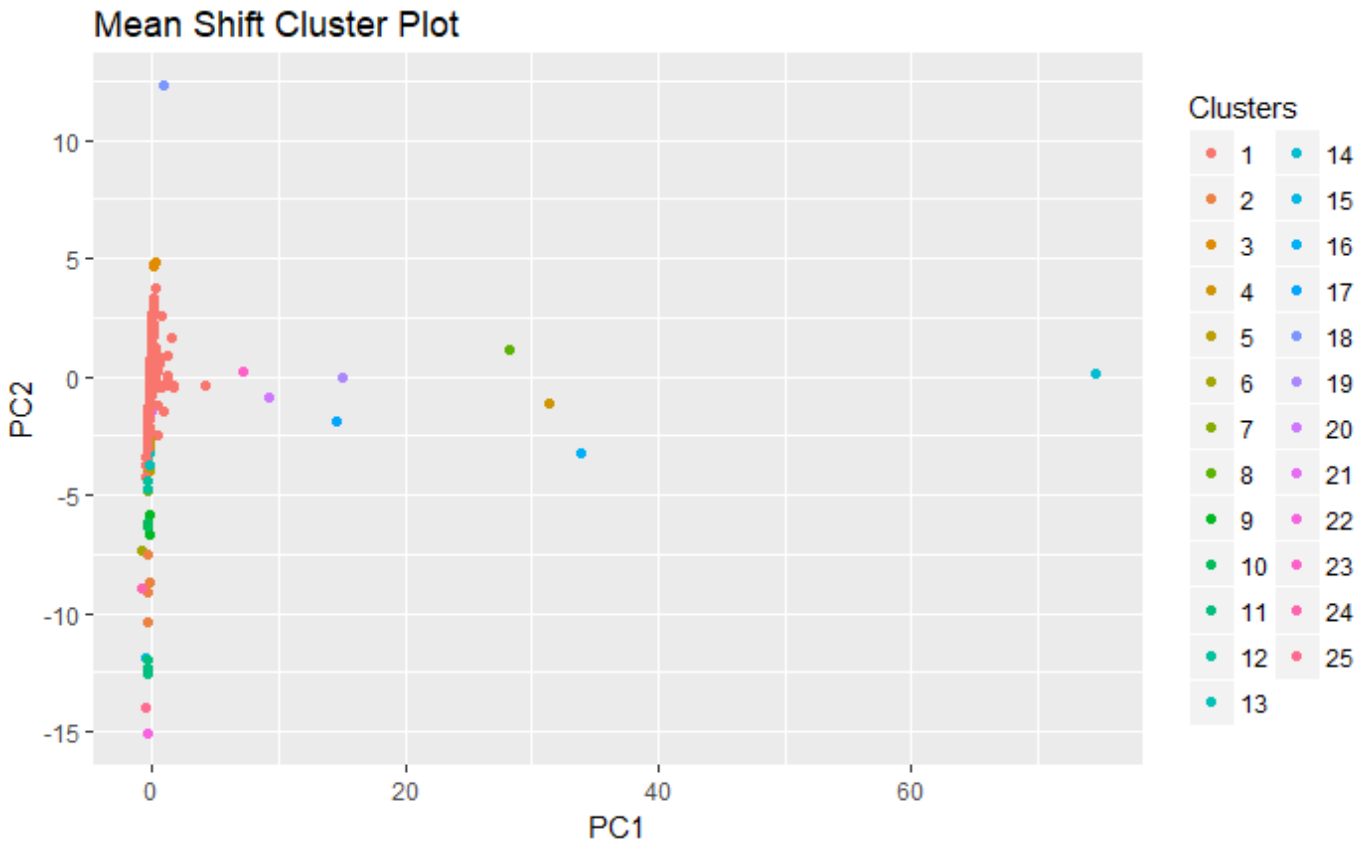
#Meanshift plots

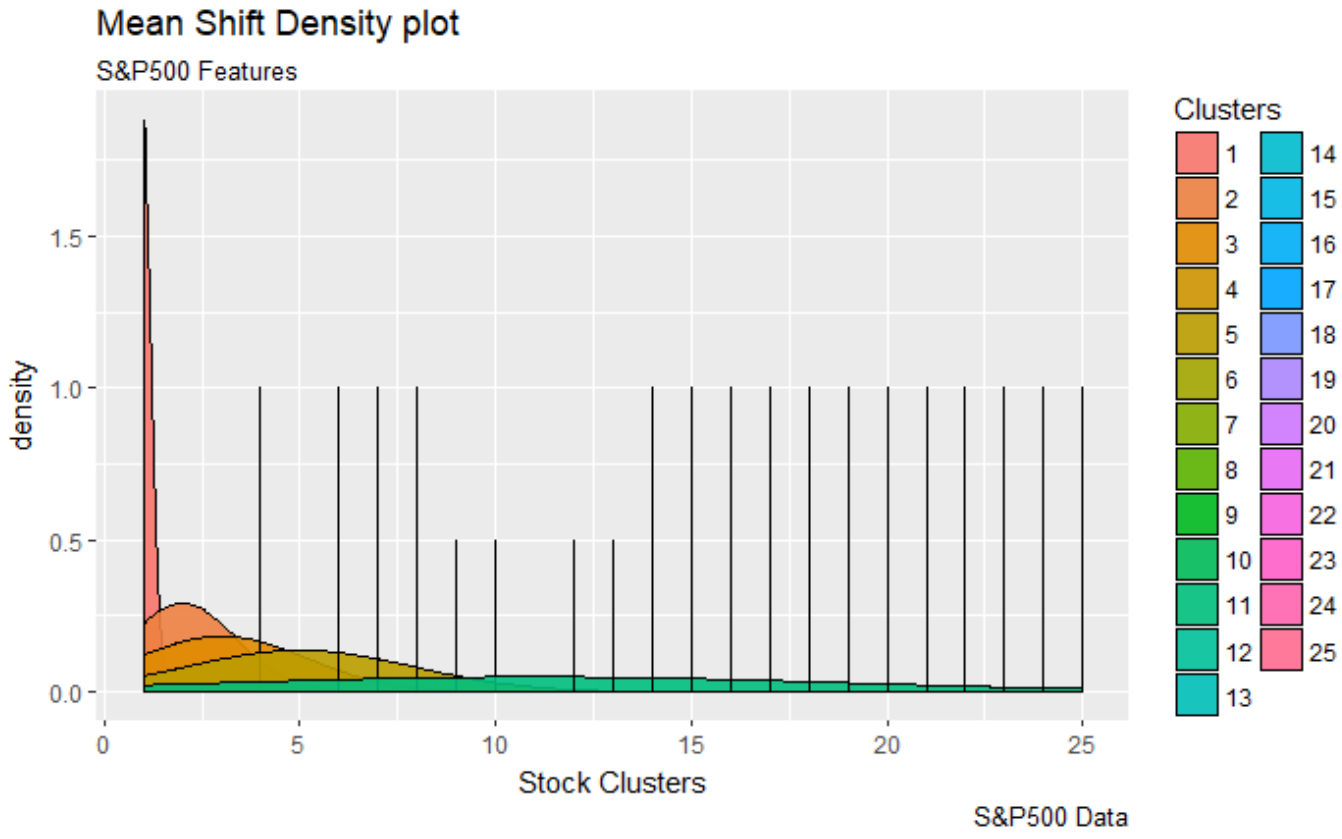
#Scatter Plot 1

```
ggplot(stocks.pca$x, aes(x= PC1, y= PC2, col= as.factor(Stocks.Clusters$labels)))+
  geom_point()+
  labs(title="Mean Shift Cluster Plot")+
  scale_color_discrete(name="Clusters")
```

#Density Plot 2

```
ggplot(stocks.pca$x, aes(Stocks.Clusters$labels))+
  geom_density(aes(fill=as.factor(Stocks.Clusters$labels)), alpha=0.9) +
  labs(title="Mean Shift Density plot",
  subtitle="S&P500 Features",
  caption="S&P500 Data",
  x="Stock Clusters",
  fill="Clusters")
```


Mean Shift Cluster Plot
PC2
PC1
10
5
0
-5
-10
-15
0
20
40
60
Clusters
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25


Mean Shift Density plot
S&P500 Features
density
Stock Clusters
S&P500 Data
1.5
1.0
0.5
0.0
0
5
10
15
20
25
Clusters
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

#Autoencoder Mode

```
h2o.init()
```

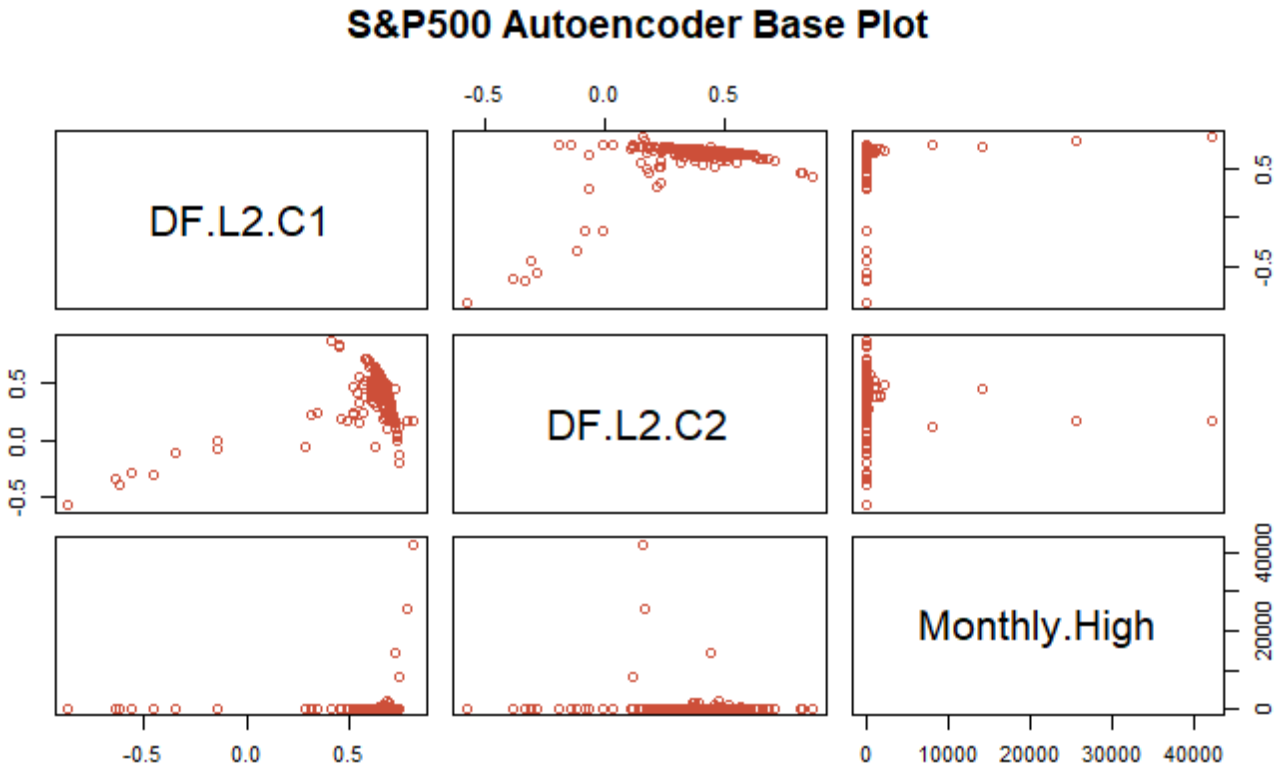
#Autoencoder model: S&P data set data manipulations

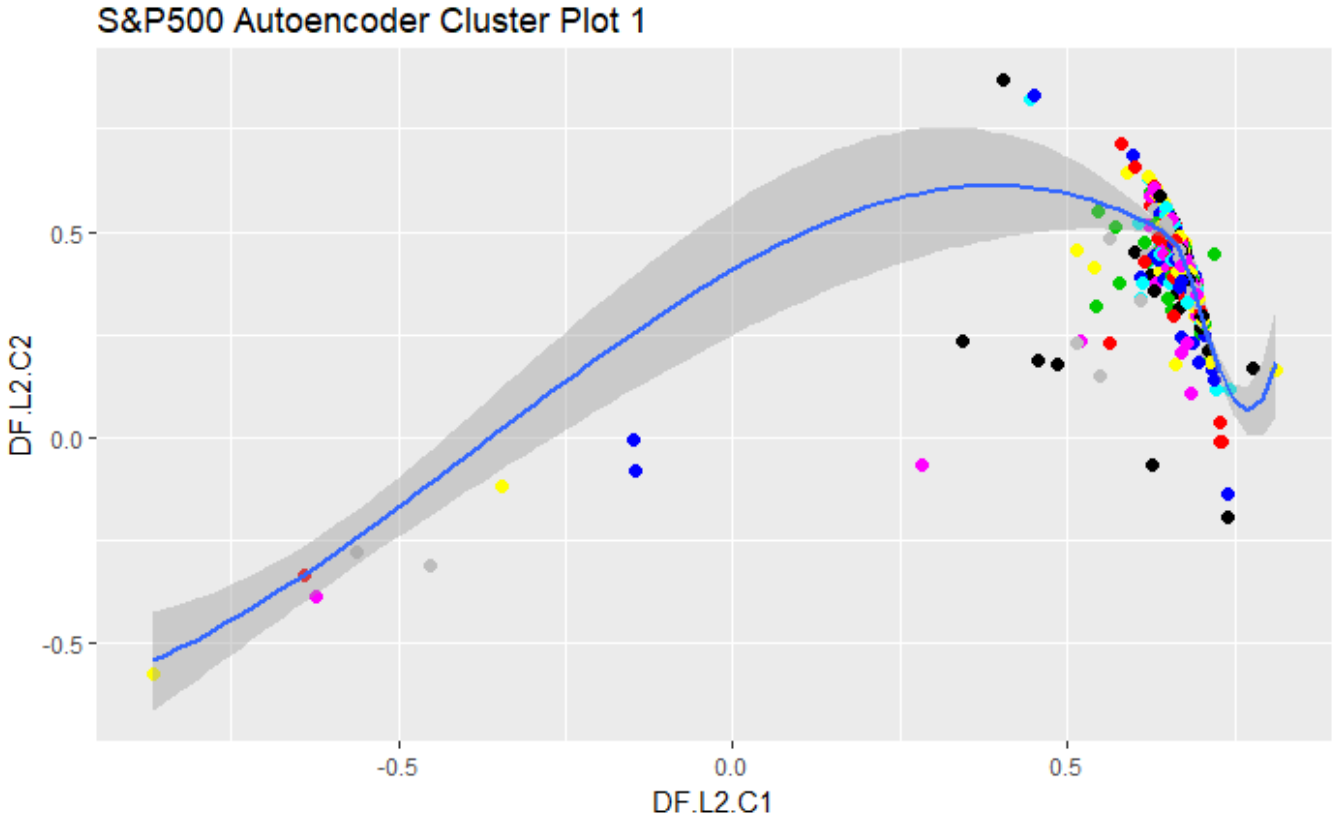
```
colnames(df.stocks.pca)

# convert data to H2OFrame
stock.col <- setdiff(colnames(df.stocks.pca), c('Monthly.High'))

df.stocks.pca<- as.h2o(df.stocks.pca)
head(df.stocks.pca)

df.stocks.pca2_train_test <- h2o.splitFrame(df.stocks.pca,
                  ratios = c(0.001, 0.001),
                  seed = 42)


train  <- df.stocks.pca2_train_test[[1]]
test1  <- df.stocks.pca2_train_test[[2]]
test2  <- df.stocks.pca2_train_test[[3]]
dim(train)
dim(test1)
dim(test2)


auto.en.model <- h2o.deeplearning(x = stock.col,
                    training_frame = train,
                    model_id = "model",
                    autoencoder = TRUE,
                    hidden = c(5, 2, 5),
                    epochs = 100,
                    activation = "Tanh")


test1.col <- h2o.deepfeatures(auto.en.model, test1, layer = 2) %>%
        as.data.frame()

Monthly.High <- as.vector(test1[,'Monthly.High'])

test1.col  <- cbind(test1.col, Monthly.High)

test1.col <- na.omit(test1.col)

head(test1.col)
```

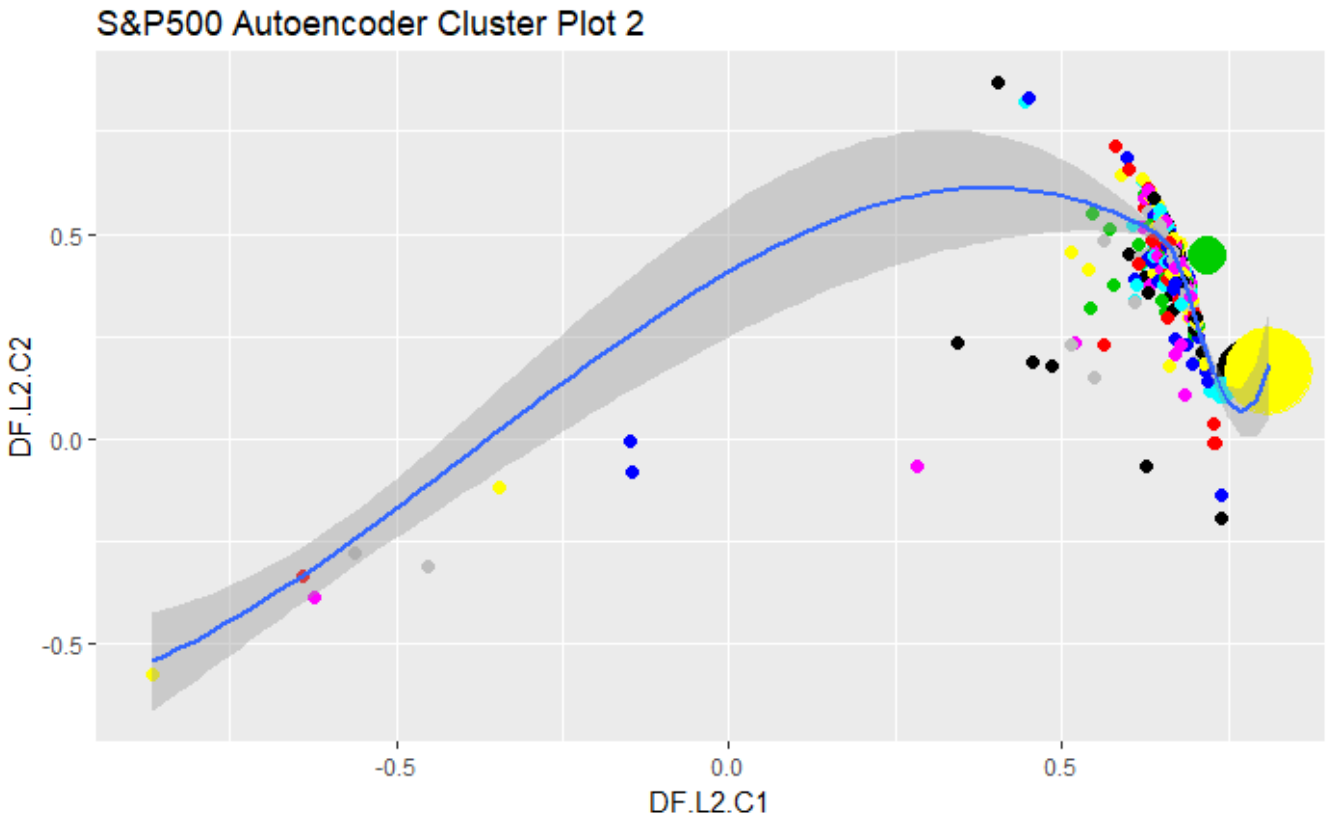
```
test1.col$Monthly.High <-as.character(test1.col$Monthly.High)

dim(test1.col)


#write_csv(test1.col,"test1.col.csv")
```

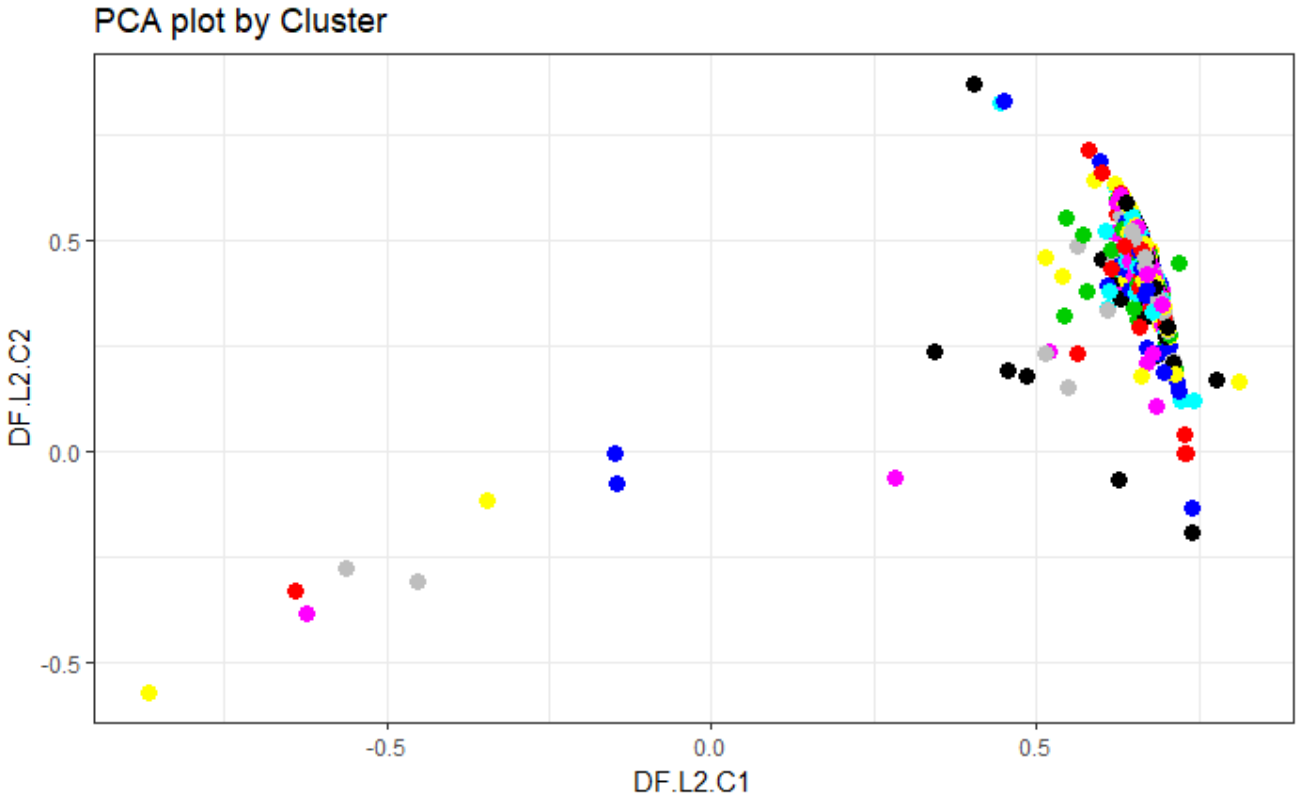
#Autoencoder model plots

```
tes1.col <- read_csv("test1.col.csv")


#Base plot
plot(test1.col, col= 'Tomato3', main = "S&P500 Autoencoder Base Plot")

#ggplot Cluster Plot 1
ggplot(test1.col, aes(x=DF.L2.C1,y=DF.L2.C2)) +
  geom_point( col= Monthly.High, size=2)+
  labs(title="S&P500 Autoencoder Cluster Plot 1")+ geom_smooth()

#ggplot Cluster Plot 2
ggplot(test1.col,
  mapping = aes(DF.L2.C1, DF.L2.C2)) +
  geom_point( col = Monthly.High,size=Monthly.High/3000+2) +
  geom_smooth() +
  labs(title="S&P500 Autoencoder Cluster Plot 2")


#ggplot Cluster Plot 3
ggplot(test1.col, aes(x= DF.L2.C1, y= DF.L2.C2))+
  geom_point(col= Monthly.High, size = 3)+
  labs(title="PCA plot by Cluster")+
  scale_color_discrete(name="S&P500 Autoencoder Cluster Plot 3")+theme_bw()
```


S&P500 Autoencoder Base Plot
DF.L2.C1
DF.L2.C2
Monthly.High
-0.5
0.0
0.5
0
10000
20000
30000
40000


S&P500 Autoencoder Cluster Plot 1
DF.L2.C2
DF.L2.C1
0.5
0.0
-0.5


S&P500 Autoencoder Cluster Plot 2
DF.L2.C2
0.5
0.0
-0.5
-0.5
0.0
0.5
DF.L2.C1


PCA plot by Cluster
DF.L2.C2
0.5
0.0
-0.5
-0.5
0.0
0.5
DF.L2.C1