Clustering Analysis on ETF Trading

Group5: Preeti Bekal, Wyatt Marciniak, Neel Shah

May 5, 2018

- OBJECTIVE: Create a procedure using Kmeans and Hierarchal Clustering Methodolgies to create optimal indicator sets for modeling ETF data.
- HOW: We will be using Linear Models and Decision Trees to create prediction “trading signals” and back test these strategies for profitability

Install/Load packages + Open connection to Functions Source File

- Install applicable libraries and load them into the working directtories. These libraries give us a very large range of funtionality from model creations and predictions to summaries and outputting methods..

1. Collect Data and add movement classifications

```
##        X          s.bio          s.biomove     gold.p
##  08-06  : 1   Min.   :-0.13957   down:32   Min.   :-0.208068
##  08-07  : 1   1st Qu.:-0.02839   up  :41   1st Qu.:-0.032645
##  08-08  : 1   Median : 0.01088             Median : 0.007202
##  08-09  : 1   Mean   : 0.01346             Mean   : 0.005489
##  08-10  : 1   3rd Qu.: 0.06692             3rd Qu.: 0.053194
##  08-11  : 1   Max.   : 0.18316             Max.   : 0.123567
##  (Other):67
##       oil                 usd                 vix
##  Min.   :-0.391169   Min.   :-0.0679549   Min.   :-0.385118
##  1st Qu.:-0.053450   1st Qu.:-0.0134864   1st Qu.:-0.162387
##  Median : 0.017892   Median : 0.0011452   Median :-0.033808
##  Mean   :-0.002505   Mean   : 0.0009421   Mean   :-0.005924
##  3rd Qu.: 0.055067   3rd Qu.: 0.0138189   3rd Qu.: 0.104700
##  Max.   : 0.275342   Max.   : 0.0834170   Max.   : 0.645797
##
##    brazil.us           canada.us             china.us
##  Min.   :-0.098916   Min.   :-0.0858318   Min.   :-0.0172482
##  1st Qu.:-0.020042   1st Qu.:-0.0144487   1st Qu.:-0.0035062
##  Median :-0.002334   Median :-0.0030441   Median :-0.0011258
##  Mean   : 0.004167   Mean   : 0.0009813   Mean   :-0.0015366
##  3rd Qu.: 0.026452   3rd Qu.: 0.0164781   3rd Qu.: 0.0002575
##  Max.   : 0.162045   Max.   : 0.1374164   Max.   : 0.0141375
##
##      jap.us            mexico.us           skorea.us
##  Min.   :-0.075052   Min.   :-0.067867   Min.   :-0.151203
##  1st Qu.:-0.019691   1st Qu.:-0.017831   1st Qu.:-0.020722
##  Median :-0.003629   Median :-0.002022   Median :-0.007669
##  Mean   :-0.000554   Mean   : 0.003118   Mean   :-0.000227
##  3rd Qu.: 0.011999   3rd Qu.: 0.016561   3rd Qu.: 0.017636
##  Max.   : 0.084392   Max.   : 0.146594   Max.   : 0.138996
##
##     swiss.us            us.euro             fedfunds
##  Min.   :-0.130846   Min.   :-0.104643   Min.   :-0.91115
##  1st Qu.:-0.018534   1st Qu.:-0.016722   1st Qu.:-0.11778
##  Median :-0.006795   Median : 0.001692   Median : 0.00000
##  Mean   :-0.002217   Mean   :-0.001754   Mean   :-0.04090
##  3rd Qu.: 0.016235   3rd Qu.: 0.020691   3rd Qu.: 0.07411
##  Max.   : 0.118363   Max.   : 0.092125   Max.   : 0.38299
##
##      umsent              homep              unemply
##  Min.   :-0.199249   Min.   :-2.284e-02   Min.   :-0.06156
##  1st Qu.:-0.026508   1st Qu.:-8.621e-03   1st Qu.:-0.01290
##  Median : 0.011881   Median :-5.573e-04   Median : 0.00000
##  Mean   : 0.004408   Mean   :-6.437e-05   Mean   : 0.00167
##  3rd Qu.: 0.046595   3rd Qu.: 8.328e-03   3rd Qu.: 0.01105
##  Max.   : 0.127624   Max.   : 2.017e-02   Max.   : 0.07095
##
##      bondhy              ust1               ust2
```

```
##  Min.   :-0.29462   Min.   :-0.78101   Min.   :-0.57808
##  1st Qu.:-0.09154   1st Qu.:-0.10763   1st Qu.:-0.14217
##  Median :-0.03167   Median :-0.04256   Median :-0.01227
##  Mean   :-0.01230   Mean   :-0.04144   Mean   :-0.02321
##  3rd Qu.: 0.05493   3rd Qu.: 0.05407   3rd Qu.: 0.08388
##  Max.   : 0.37963   Max.   : 0.34294   Max.   : 0.32047
##
##       ust3               ust5               ust7
##  Min.   :-0.58192   Min.   :-0.411980   Min.   :-0.400160
##  1st Qu.:-0.12063   1st Qu.:-0.086178   1st Qu.:-0.070690
##  Median :-0.02532   Median :-0.022223   Median :-0.007273
##  Mean   :-0.01500   Mean   :-0.008611   Mean   :-0.006265
##  3rd Qu.: 0.08456   3rd Qu.: 0.058108   3rd Qu.: 0.046520
##  Max.   : 0.39043   Max.   : 0.357415   Max.   : 0.275229
##
##      ust10               ust30             autosale
##  Min.   :-0.377530   Min.   :-0.33198   Min.   :-0.354156
##  1st Qu.:-0.055119   1st Qu.:-0.03133   1st Qu.:-0.038962
##  Median : 0.000000   Median : 0.00000   Median : 0.006093
##  Mean   :-0.005484   Mean   :-0.00406   Mean   :-0.002780
##  3rd Qu.: 0.040491   3rd Qu.: 0.02836   3rd Qu.: 0.040714
##  Max.   : 0.175657   Max.   : 0.13712   Max.   : 0.164333
##
```

- (1): We have imported our data sets from the Data Script we wrote prior to the analysis. This script collects market index and etf daily data as well as economic indicator data, exchange rate data and interest rate data for US. Treasuries (1 - 30 years). This data can then be aggregated on a daily, weekly, monthly or quarterly basis. After the script is read into the environment, we separate it into training and testing subsets. We will be predicting for 3 years (using roughly 9 years of data to test) from June 2014 - July 2017. We believe that this time frame will show a fair amount of standard deviation but also put the economy in an more “average” growth period of a bull mark.

For this project, we are focusing on monthly data returns so that we can better utilize market/economic instruments to predict their price movements. These ETFs are “baskets” that are designed to represent a sector of the market (in our case, but in general ETFs have a wide range of applications. For our purpose, we are using US Sector ETFs and interpreting them as proxies for actual “actu} act”

- We are limiting the predictors data sets to non-ETF sets because we want to try to capture market effects in the least biased way possible to create a more “pure” indicator strategy. Allowing the use of ETF data will create multi-colinearity effects in our models and the procedure will be less accurate. In practice, this additional analysis should be conducted and compared on predidiction accuracy as a baseline but we felt that objective of our projective did not reaire that adddional

2. Analzying Correlation and regression rankings (by abs() size)

```
## Subset Data for evaluating indicator to ETF correlations
etf.data <- train      # Remove Date (Useless data for correlation testing)
etf.data <- etf.data[5:length(etf.data)]  # Subset ETFs + Indicators (Drop Dates and Market Indica
tors)

# Analyze and create correlation subsets
obs.cor.all   <- analyze.ranks(etf.data, "All Return Correlations","subset",10, 0.00, top = NULL)
```

```
## [1] ------------------------------------------------------------------------
## [1] Ranking Correlations for All Return Correlations
## [1] (1) Building Correlation + Regression (R^2) Tables...
## [1] (2) Ranking Correlations and R^2 values + Creating Summary Data Frame...
## [1] (3) Subsetting abs(Correlations) >= 0 ...
```

```
obs.cor.sub   <- analyze.ranks(etf.data, "Sub Return correlations","subset",10, 0.30, top = NULL,
print = TRUE)
```

```
## [1] ------------------------------------------------------------------------
## [1] Ranking Correlations for Sub Return correlations
## [1] (1) Building Correlation + Regression (R^2) Tables...
## [1] (2) Ranking Correlations and R^2 values + Creating Summary Data Frame...
## [1] (3) Subsetting abs(Correlations) >= 0.3 ...
## [1] ------------------------------------------------------------------------
## [1] For s.bio : ( mexico.us, vix, skorea.us, usd, canada.us, us.euro, brazil.us, swiss.us )
## [1] For s.cd : ( skorea.us, mexico.us, canada.us, vix, usd, us.euro, brazil.us, autosale, oil,
swiss.us )
## [1] For s.cs : ( canada.us, vix, mexico.us, usd, us.euro, skorea.us, oil, brazil.us, swiss.us,
autosale )
## [1] For s.energy : ( canada.us, oil, mexico.us, usd, brazil.us, vix, us.euro, skorea.us, swiss.
us, autosale, ust10, ust7, ust30, ust2, ust3, ust5 )
## [1] For s.fin : ( mexico.us, skorea.us, canada.us, usd, vix, us.euro, brazil.us, oil, autosale,
swiss.us, unemply )
## [1] For s.hc : ( vix, mexico.us, skorea.us, usd, us.euro, canada.us, brazil.us, swiss.us, oil,
autosale )
## [1] For s.ind : ( mexico.us, skorea.us, vix, canada.us, usd, brazil.us, us.euro, oil, autosale,
swiss.us )
## [1] For s.mat : ( mexico.us, canada.us, vix, brazil.us, usd, skorea.us, us.euro, oil, autosale,
swiss.us )
## [1] For s.tech : ( mexico.us, vix, canada.us, skorea.us, usd, oil, brazil.us, us.euro, autosale
, swiss.us )
## [1] For s.util : ( vix, mexico.us, usd, canada.us, brazil.us, us.euro, oil, skorea.us )
```

```
save(obs.cor.all, obs.cor.sub, file = "FE582_DataCorrTables.RData")

# Exemple of Ranked objects
kable(obs.cor.all$df$s.fin, caption = "S.fin ~ Indicators with Correlation >= 0")
```

S.fin ~ Indicators with Correlation >= 0

data	corr	r2
mexico.us	-0.6910	0.4775
skorea.us	-0.6823	0.4655
canada.us	-0.6515	0.4244
usd	-0.6211	0.3858
vix	-0.5625	0.3164
us.euro	0.5598	0.3134
brazil.us	-0.5144	0.2646
oil	0.5063	0.2563
autosale	0.4914	0.2414
swiss.us	-0.4500	0.2025
unemply	-0.3081	0.0950
umsent	0.2969	0.0882
homep	0.2727	0.0743
ust2	0.2572	0.0662
ust3	0.2463	0.0606
ust7	0.2335	0.0545
ust5	0.2261	0.0511
ust10	0.1959	0.0384

data	corr	r2
fedfunds	0.1860	0.0346
jap.us	0.1727	0.0298
ust1	0.1589	0.0253
ust30	0.1336	0.0178
china.us	-0.1332	0.0177
bondhy	-0.0921	0.0085
gold.p	0.0058	0.0000

```
kable(obs.cor.sub$df$s.fin, caption = "S.fin ~ Indicators with Correlation >= 0.30")
```

S.fin ~ Indicators with Correlation >= 0.30

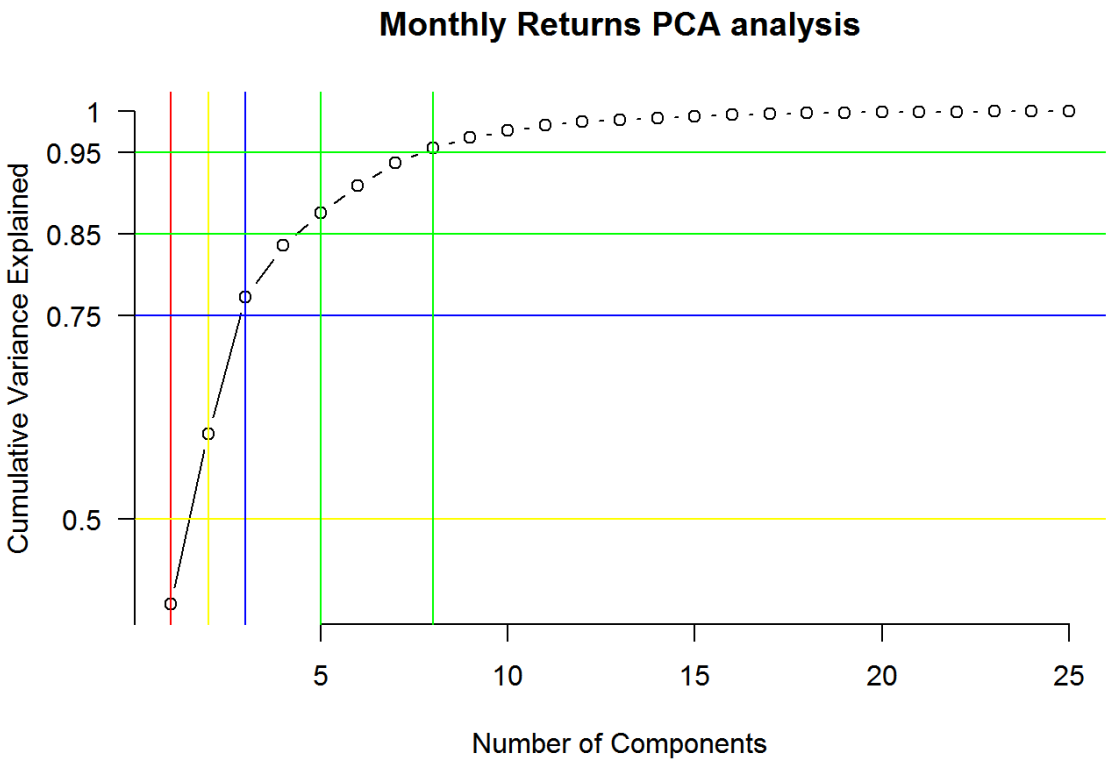
data	corr	r2
mexico.us	-0.6910	0.4775
skorea.us	-0.6823	0.4655
canada.us	-0.6515	0.4244
usd	-0.6211	0.3858
vix	-0.5625	0.3164
us.euro	0.5598	0.3134
brazil.us	-0.5144	0.2646
oil	0.5063	0.2563
autosale	0.4914	0.2414
swiss.us	-0.4500	0.2025
unemply	-0.3081	0.0950

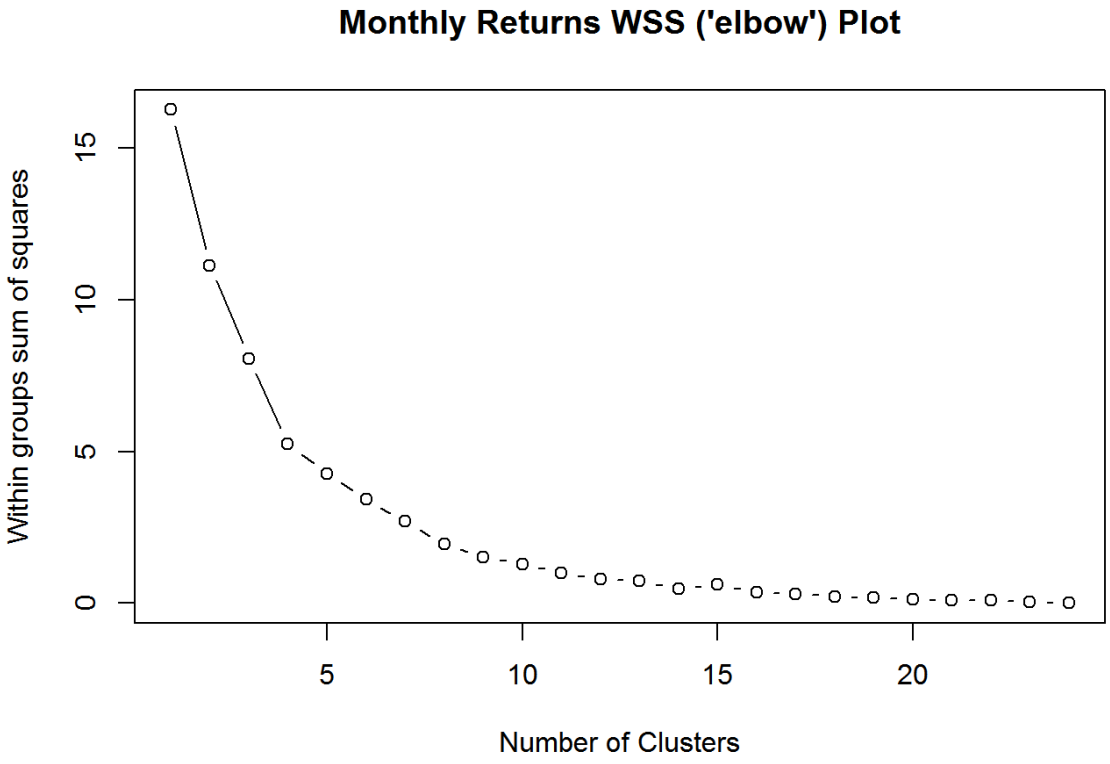
- (2): Once the data is prepared, we conduct an intial preparation step of subsetting the data sets by indicator correlations with the taregt ETF. To do this, we rank the correlations by their absolute values (to establish strength, not direction). We use the '10' to subset the data because it takes only the first 10 rows with all columns after the first 10 with repsect to the correlation and regression tables. This allows us to isolate only the ETF data (rows) against indicators (columns after 10). We then subset these orders by the correlation subsets of 0 and 0.3.. 0 correlation groups means all the data avaliable and is done to create a base to analyze the clustered models from. The 30% threshold is set because we want to only take indicators with at least a weak correlation to analyse. After this step, we now have the first truly filtered daily sets by relationship. Next, we will use cluster analysis to identify deeper layers of relationships and subset further to create the most optimized models.

(3.1) Conduct K-means Clusterng Analysis (Euclidean Distance)

```
set.seed(1234)
max.out = 8

### Prepare Data Set for modelling (transpose training data and remove date and market/etf returns
)
ret <- t(train[,-1]) ; colnames(ret) <- train[,1]
ret <- ret[14:nrow(ret),]

## Analyze Data to interpret possible clusterings
pc.wss.analysis(ret, nrow(ret)-1, title = "Monthly Returns")
```

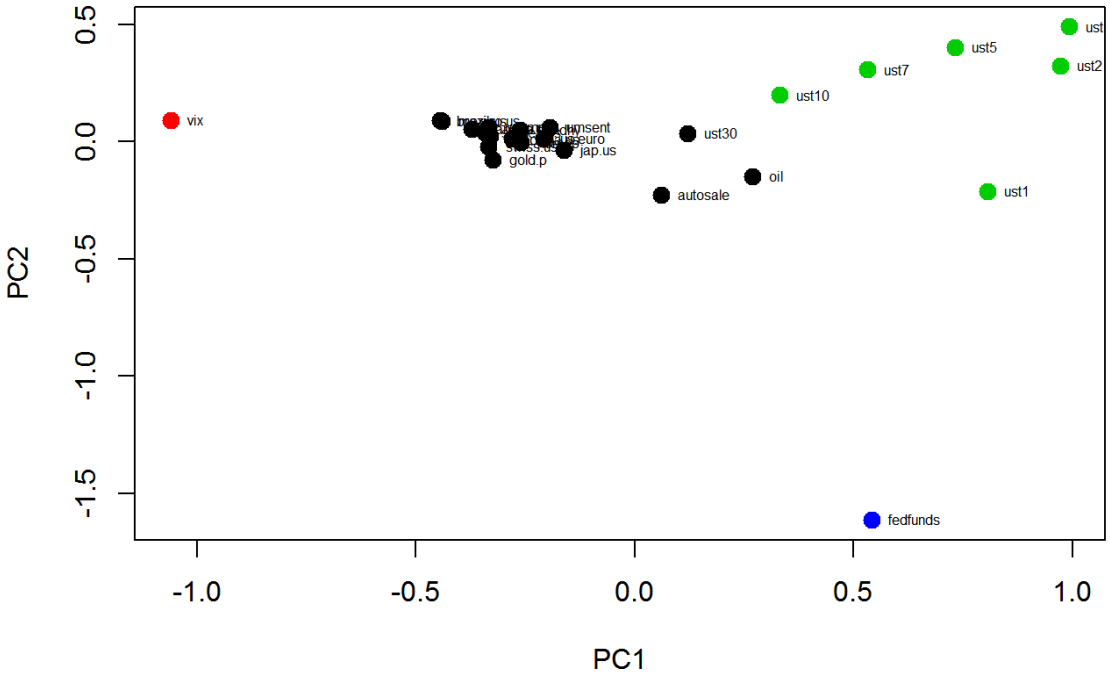
```
## [1] # Number of PCs needed to explain 25% of the variance: 1
## [1] # Number of PCs needed to explain 50% of the variance: 2
## [1] # Number of PCs needed to explain 75% of the variance: 3
## [1] # Number of PCs needed to explain 85% of the variance: 5
## [1] # Number of PCs needed to explain 95% of the variance: 8
## [1] ----------------------------------------------------------------------
```



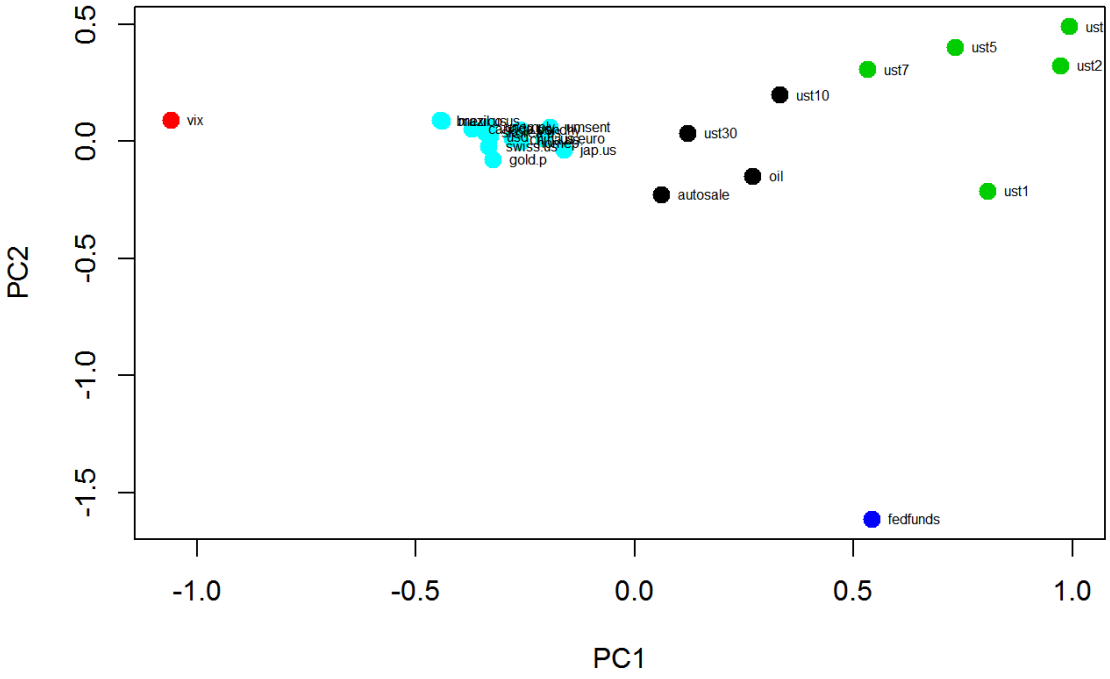



```
test.kmeans(ret, 4, 9, add.labels = TRUE)
```

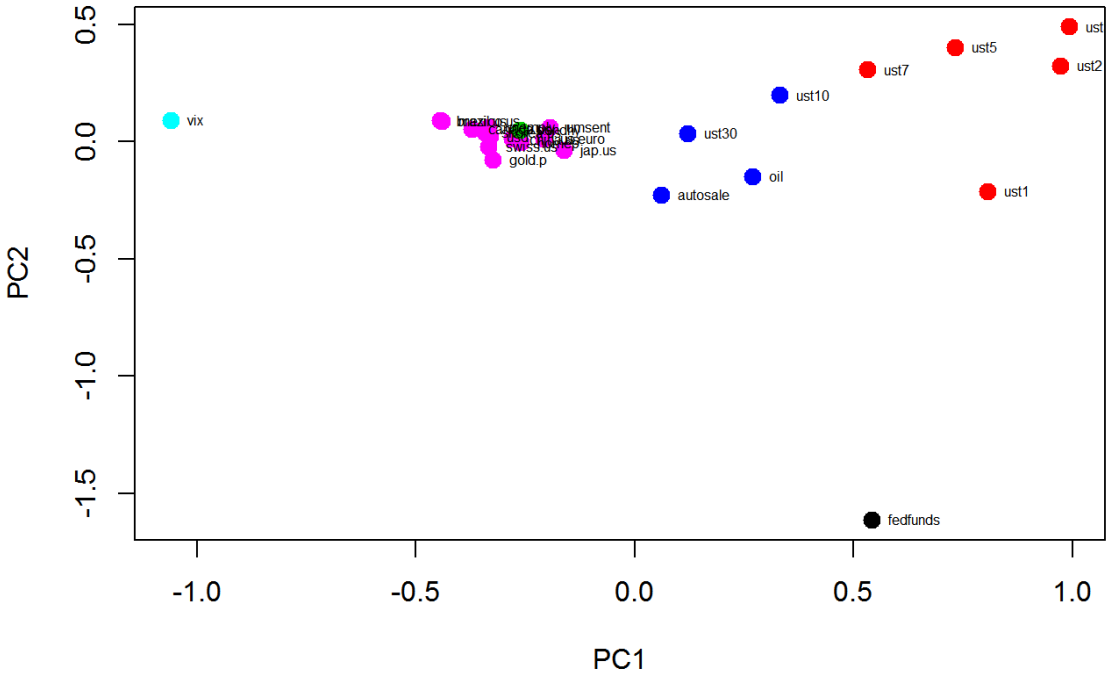
Kmean Clustering (Distance) with k = 4



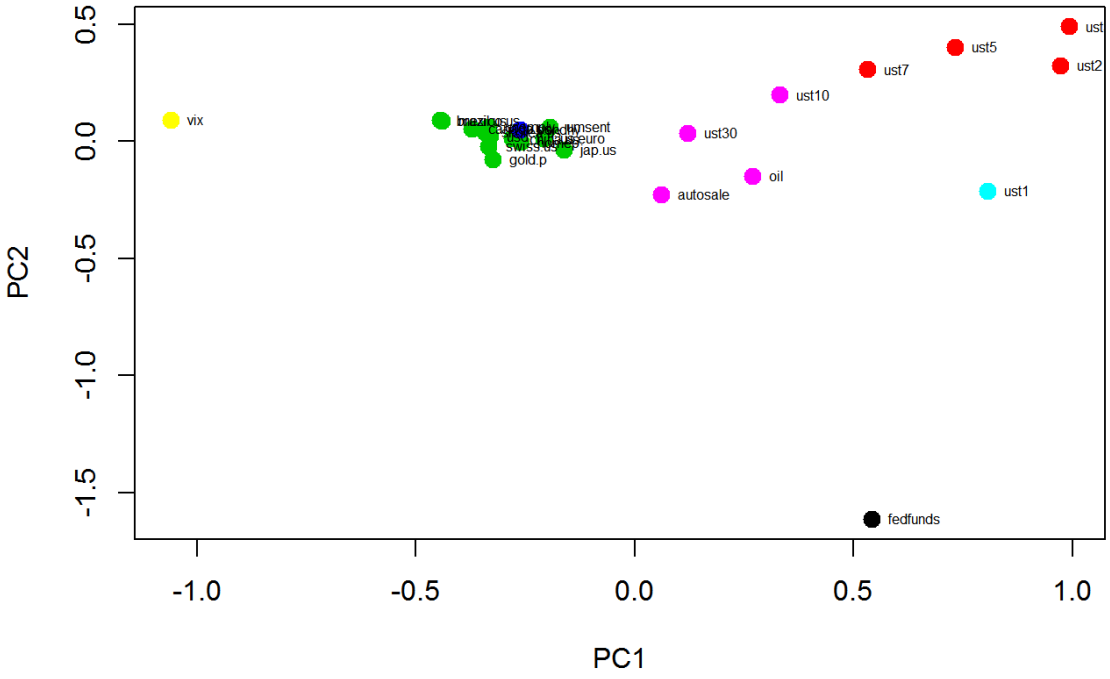
Kmean Clustering (Distance) with k = 5

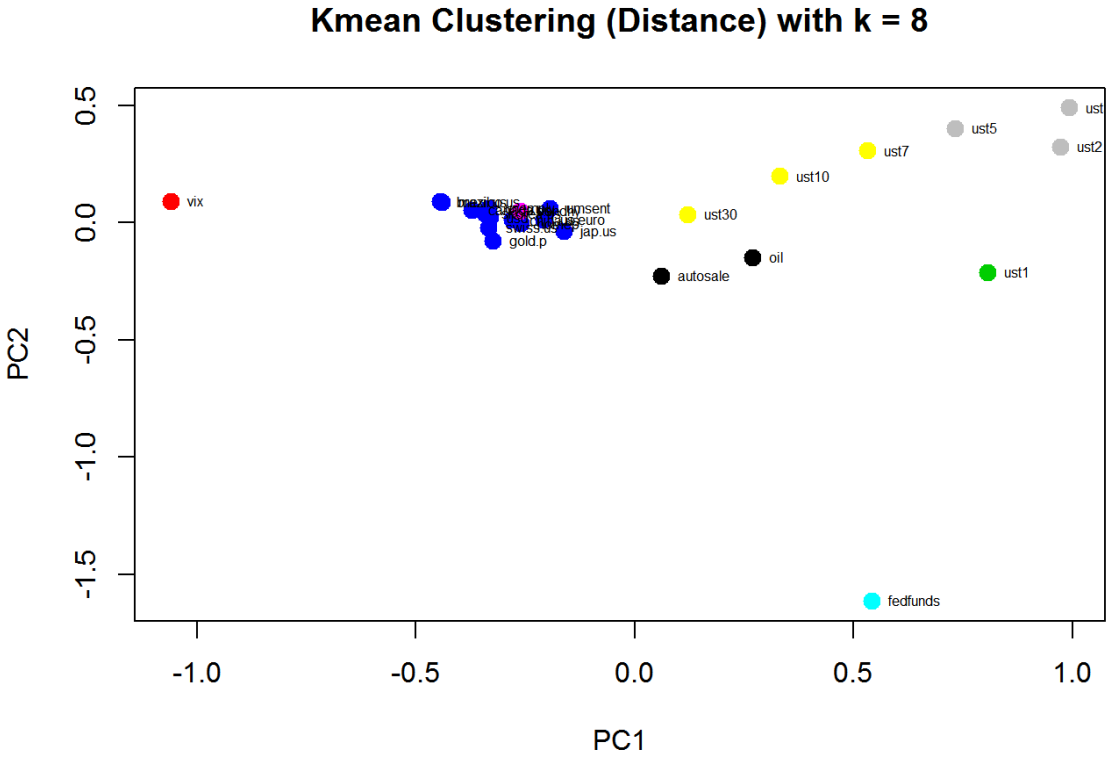


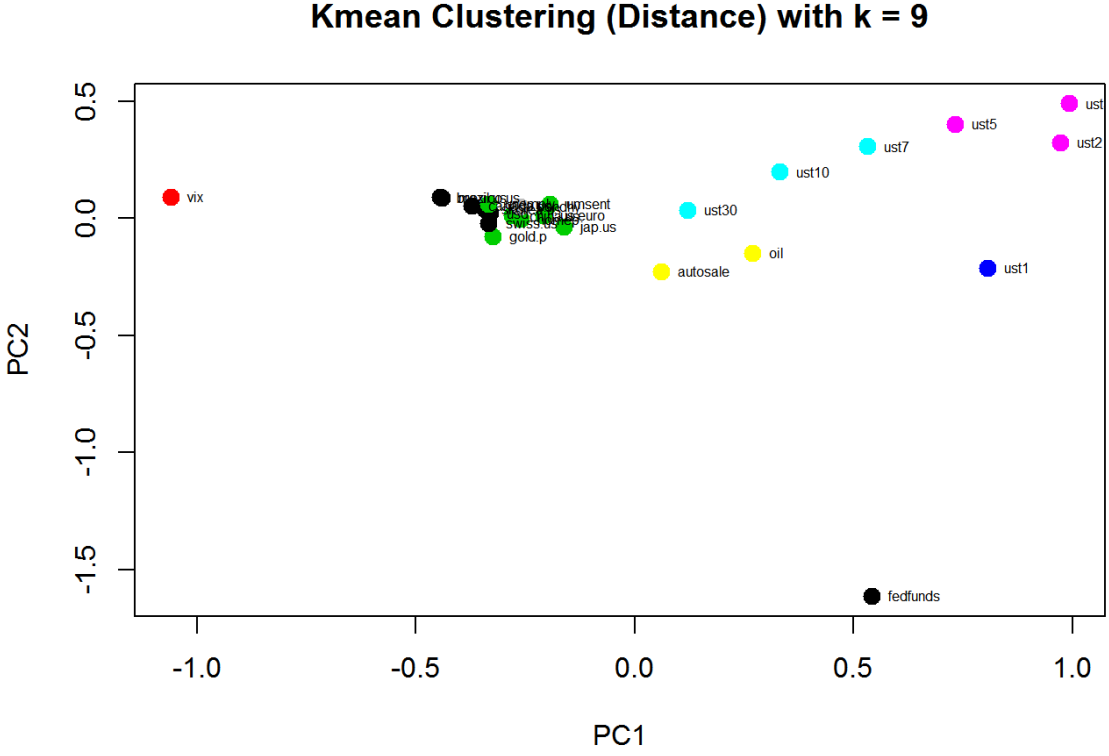
Kmean Clustering (Distance) with k = 6



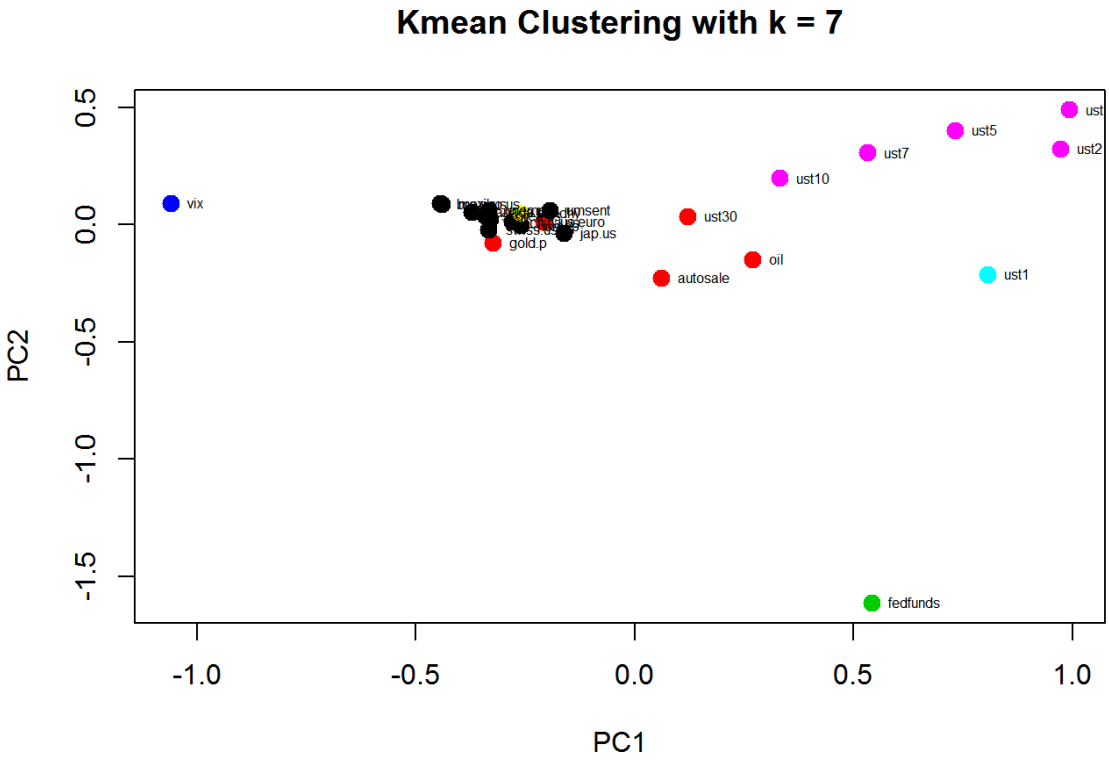
Kmean Clustering (Distance) with k = 7








```
# Selected Model (store cluster groups with their tags)
k = 7 ; k.out <- run.clustering(ret, k, "kmeans", max.elem.out = max.out, add.labels = TRUE)
```




```
## [1] Cluster 1 contents (up to 8) :usd brazil.us canada.us china.us jap.us mexico.us skorea.us s
wiss.us
## [1] Cluster 2 contents (up to 8) :gold.p oil us.euro ust30 autosale
## [1] Cluster 3 contents (up to 8) :fedfunds
## [1] Cluster 4 contents (up to 8) :vix
## [1] Cluster 5 contents (up to 8) :ust1
## [1] Cluster 6 contents (up to 8) :ust2 ust3 ust5 ust7 ust10
## [1] Cluster 7 contents (up to 8) :bondhy
```

```
cluster.k.d <- k.out[[2]]
```

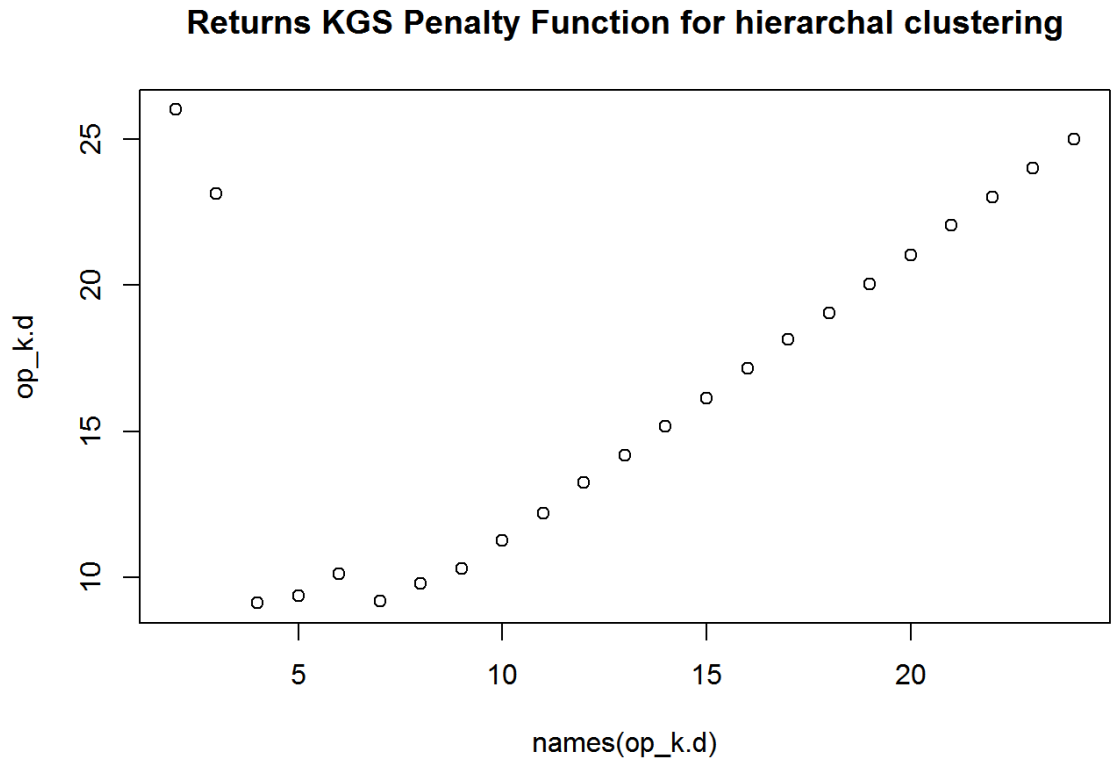
- (3.1): K-means clustering consists of cluster data based on some type of “distance measure” between them (over their value used). The most common methology to use is Euclidean Distance, which is the “linear” distance between points in the clusters from other clusters. The goal here is to identify sime number K clusters the data should fit into and then test those clusters to idenify a truly best K value (number of clusters) the data fits into. We use the WSS plot to see best ordes by minimizing the Within Sum of Squares deviance and use our with our defined PCA plot and output to see how many principal compenents we need to model x% of accuracy. We can see 3-5 capture 75-95% of the variance so a 2d plot may suffice. We see from the WSS plot the best k’s to test are from 4 to 8 (where changes in WSS become minimal by adding more k’s). After testing (shown above) we select 7 clusters for this model. The plot looks a little off but that is due to the number of P.C. we need. The contents, however, seem to make sense based on our data sets.

We will run 3 more clustering/ordering models to extract different behavior classification types (by clusters) and use the corresponding cluster tags (groups) to filter our correlation filtered data to attempt to create optimized models.

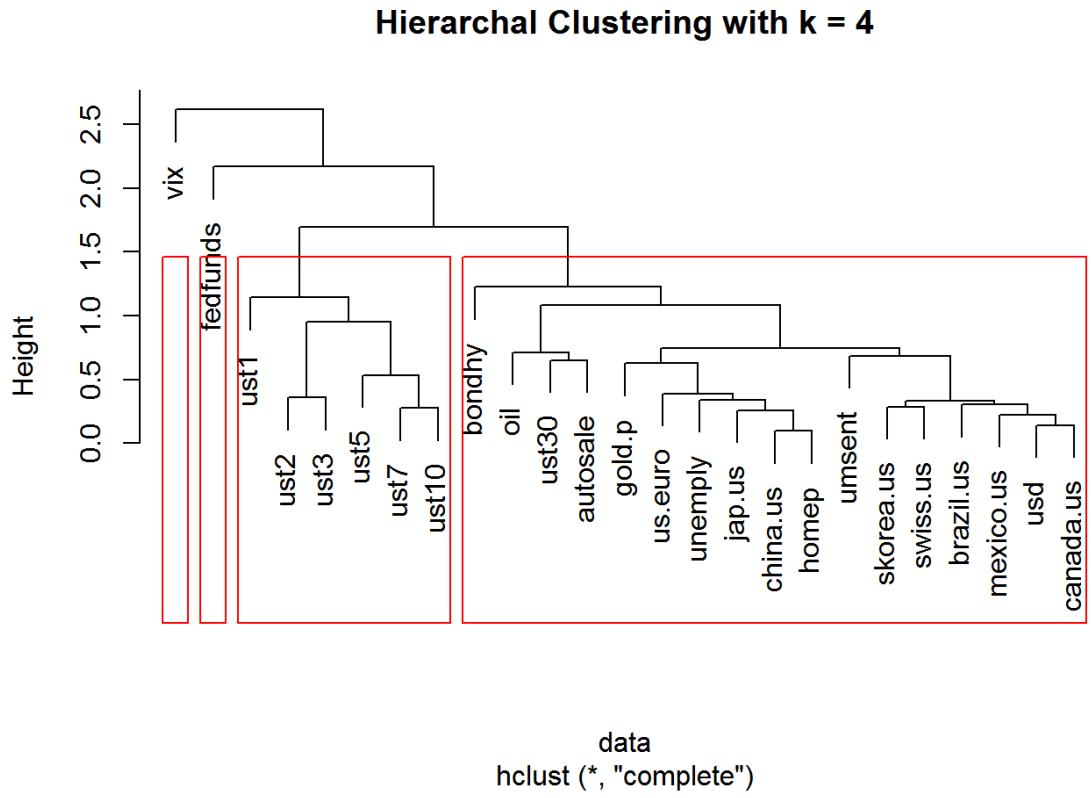
- Just like for correlation filtering, we use only indicator data sets here.

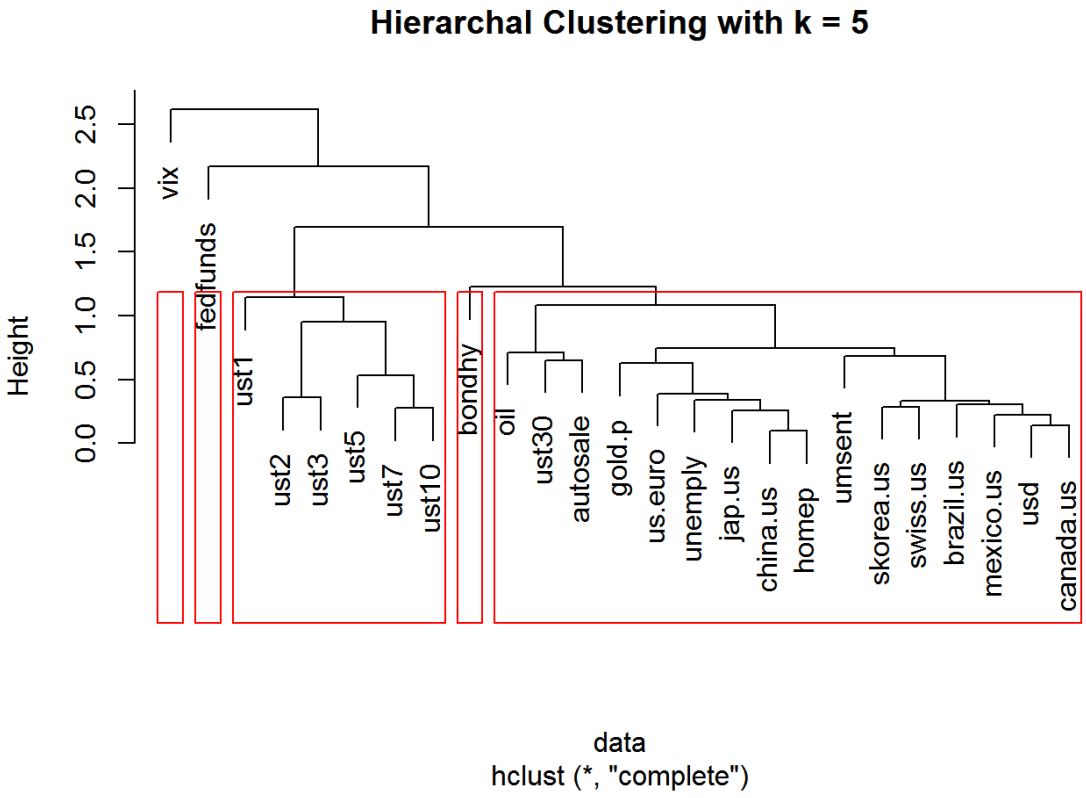
(3.2) Conduct Hierarchal Clustering (Euclidean Distance)

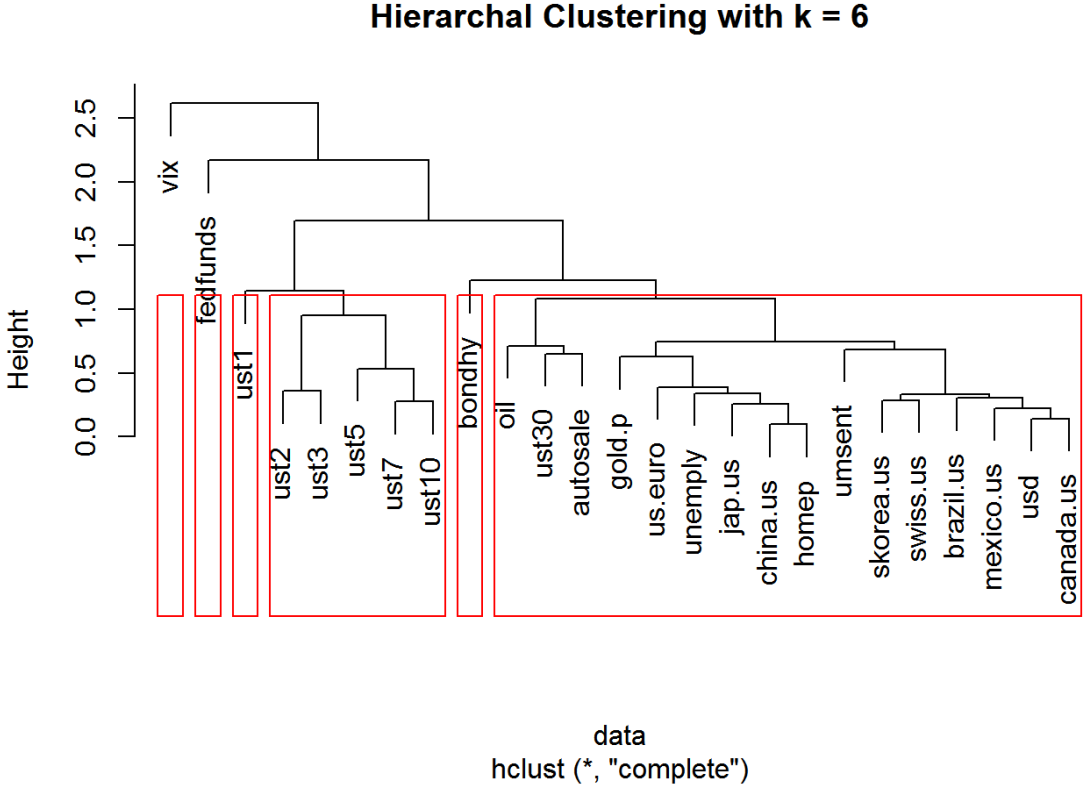
```
set.seed(1234)

### Prepare Data Set for modelling (distance matrix and intial model for order plot analysis)
d  <- dist(ret)
hc <- hclust(d, method = "complete")

## Analyze Data to interpret possible clusterings
op_k.d <- kgs(hc, d, maxclust = nrow(ret)-1) # Min = optimal clusters (k)
plot(names(op_k.d), op_k.d, main = "Returns KGS Penalty Function for hierarchal clustering")
```

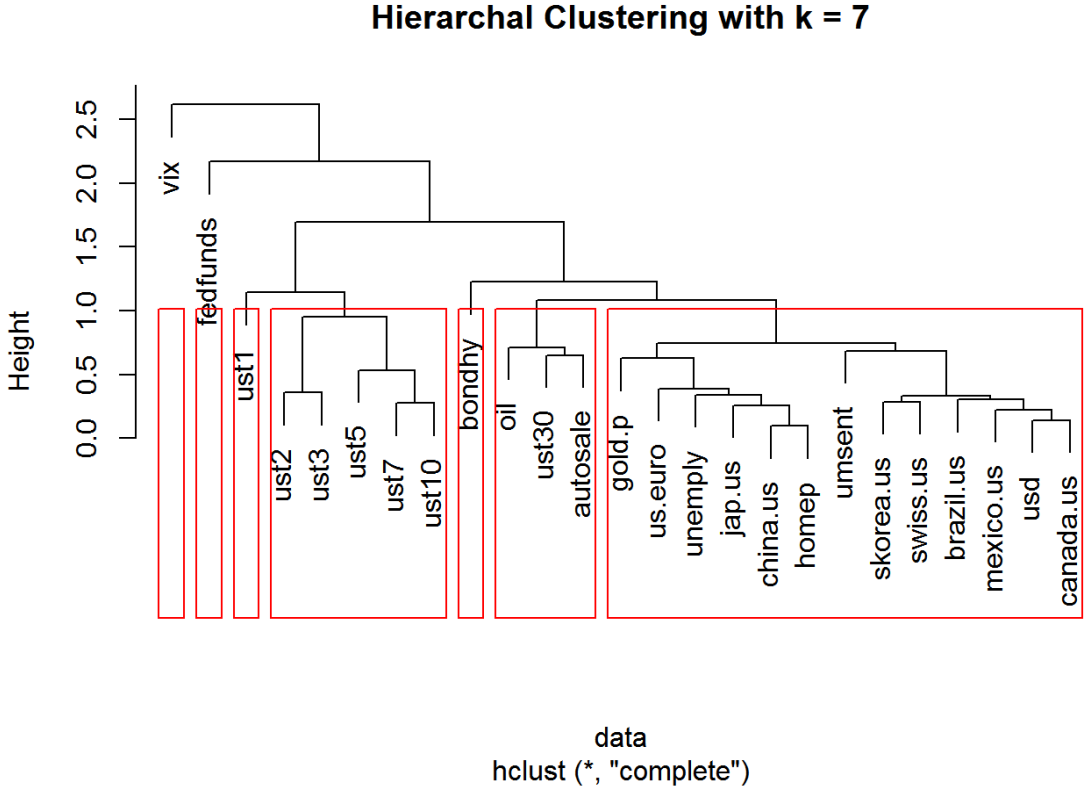

Returns KGS Penalty Function for hierarchal clustering
op_k.d
10
15
20
25
5
10
15
20
names(op_k.d)

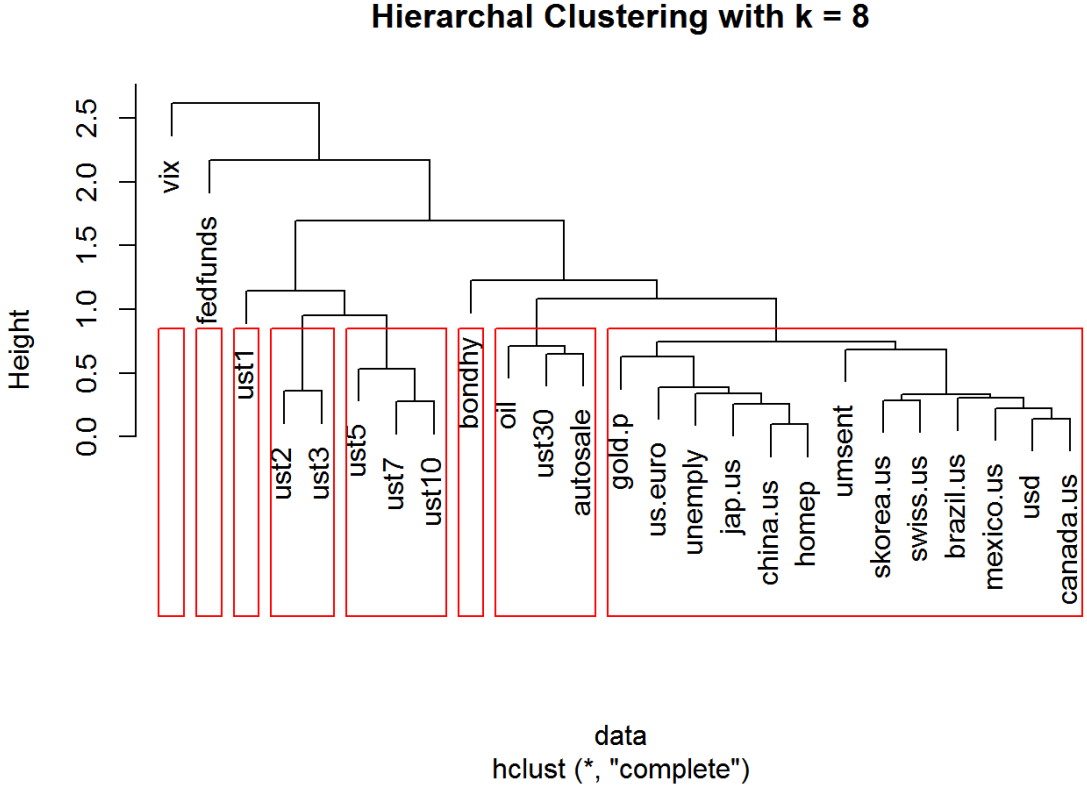
```
test.hc(d, 4, 8)
```


Hierarchal Clustering with k = 4
Height
0.0
0.5
1.0
1.5
2.0
2.5
vix
fedfunds
ust1
ust2
ust3
ust5
ust7
ust10
bondhy
oil
ust30
autosale
gold.p
us.euro
unemply
jap.us
china.us
homep
umsent
skorea.us
swiss.us
brazil.us
mexico.us
usd
canada.us
data
hclust (*, "complete")


Hierarchal Clustering with k = 5
Height
0.0
0.5
1.0
1.5
2.0
2.5
vix
fedfunds
ust1
ust2
ust3
ust5
ust7
ust10
bondhy
oil
ust30
autosale
gold.p
us.euro
unemply
jap.us
china.us
homep
umsent
skorea.us
swiss.us
brazil.us
mexico.us
usd
canada.us
data
hclust (*, "complete")


Hierarchal Clustering with k = 6
Height
0.0
0.5
1.0
1.5
2.0
2.5
vix
fedfunds
ust1
ust2
ust3
ust5
ust7
ust10
bondhy
oil
ust30
autosale
gold.p
us.euro
unemply
jap.us
china.us
homep
umsent
skorea.us
swiss.us
brazil.us
mexico.us
usd
canada.us
data
hclust (*, "complete")






```
# Selected Model (store ordered groups with their tags)
k = 8 ; k.out <- run.clustering(d, k, "hc", max.elem.out = max.out)
```

```
## [1] Cluster 1 contents (up to 8) :gold.p usd brazil.us canada.us china.us jap.us mexico.us skor
ea.us
## [1] Cluster 2 contents (up to 8) :oil ust30 autosale
## [1] Cluster 3 contents (up to 8) :vix
## [1] Cluster 4 contents (up to 8) :fedfunds
## [1] Cluster 5 contents (up to 8) :bondhy
## [1] Cluster 6 contents (up to 8) :ust1
## [1] Cluster 7 contents (up to 8) :ust2 ust3
## [1] Cluster 8 contents (up to 8) :ust5 ust7 ust10
```

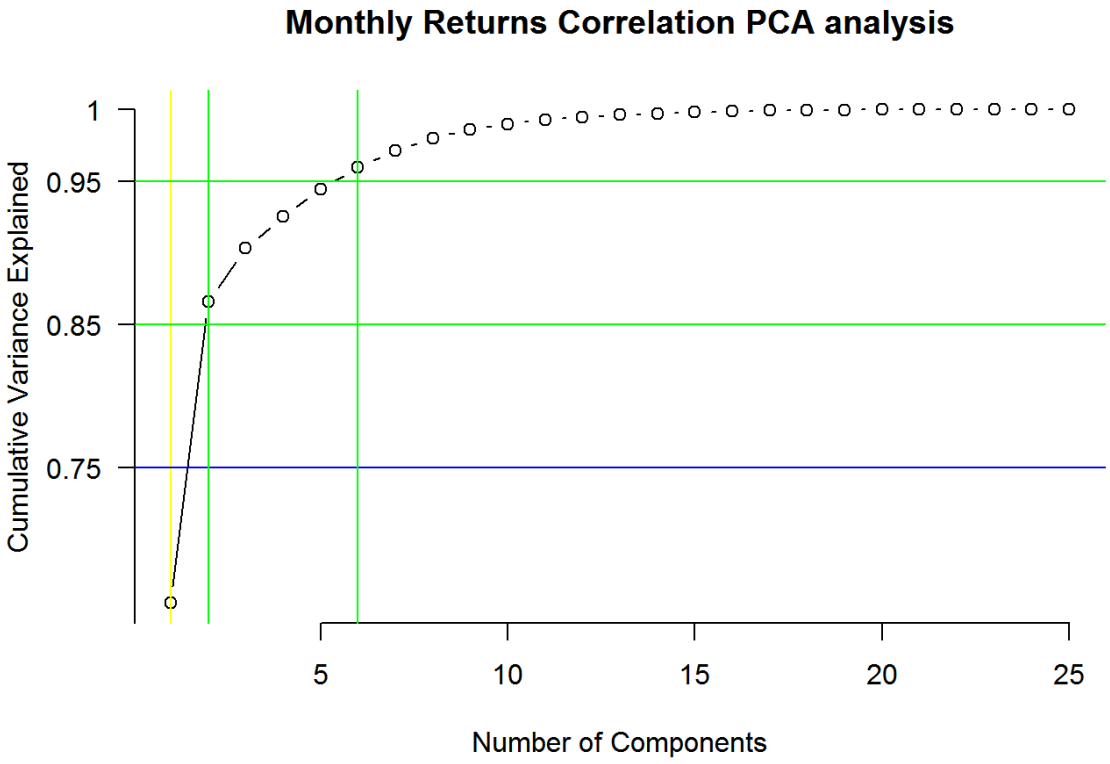
```
cluster.h.d <- k.out[[2]]
```

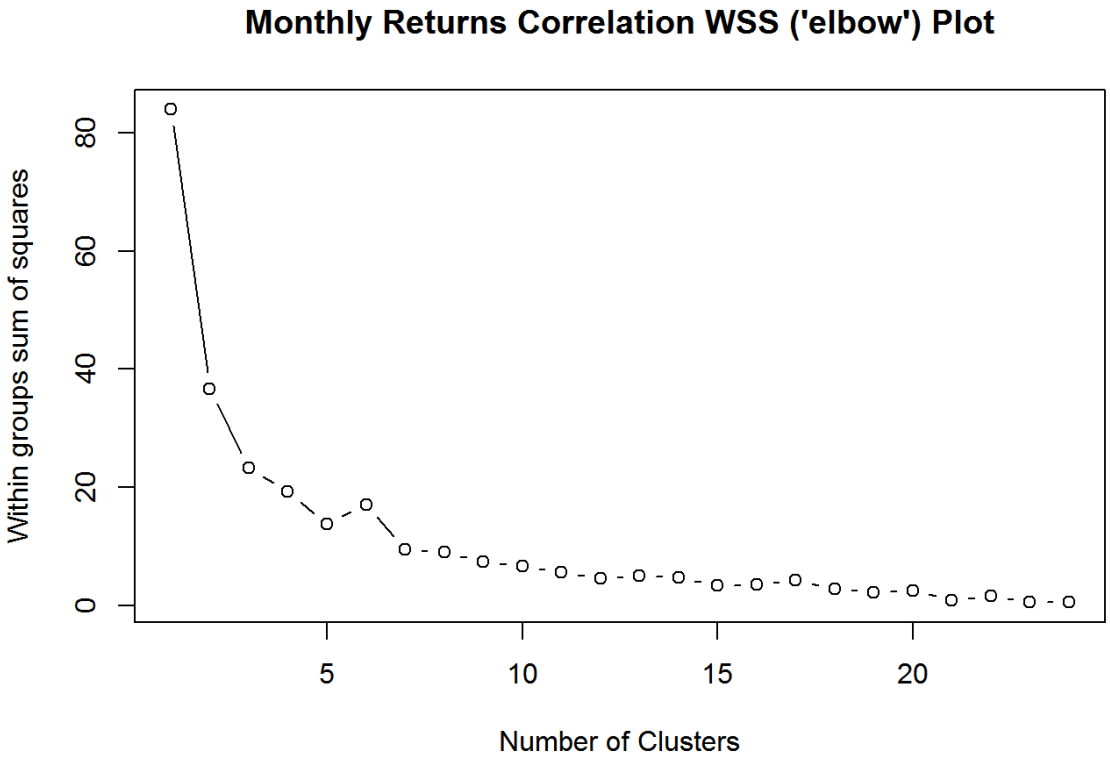
- (3.2): Hierarchal Clustering analalysis works differntly from K-means in that it looks to see at when (height of tree to root) the data types were grouped. The branches and their meeting poitns show distance and grouping points and the nodes (leaves) are the variables being analyzed. We can use the KGS (Kelley-Gardner-Sutcliffe penalty function) to identify on the constructed tree where the “pruning” should take place. These “groups” or “pruned leaves” are identified minimized penalties (min of function). We test prunings of 4 to 8 and after analysis, 8 groups were ideal. This analysis is very similair to Kmeans testting, but the tree visualization is much easier to interpret regardless of PCA reuslts. The contents of the groups make sense after the testing as we will use these paramaters for this analysis technique.

(3.3) Conduct K-means Clusterng Analysis (Corr-dist)

```
set.seed(1234)

### Prepare Data Set for modelling (transpose training data and create a correlation matrix)
ret.c <- data.frame(t(ret))
ret.c <- cor.agg(ret.c)

## Analyze Data to interpret possible clusterings
pc.wss.analysis(ret.c, nrow(ret.c)-1, title = "Monthly Returns Correlation")
```

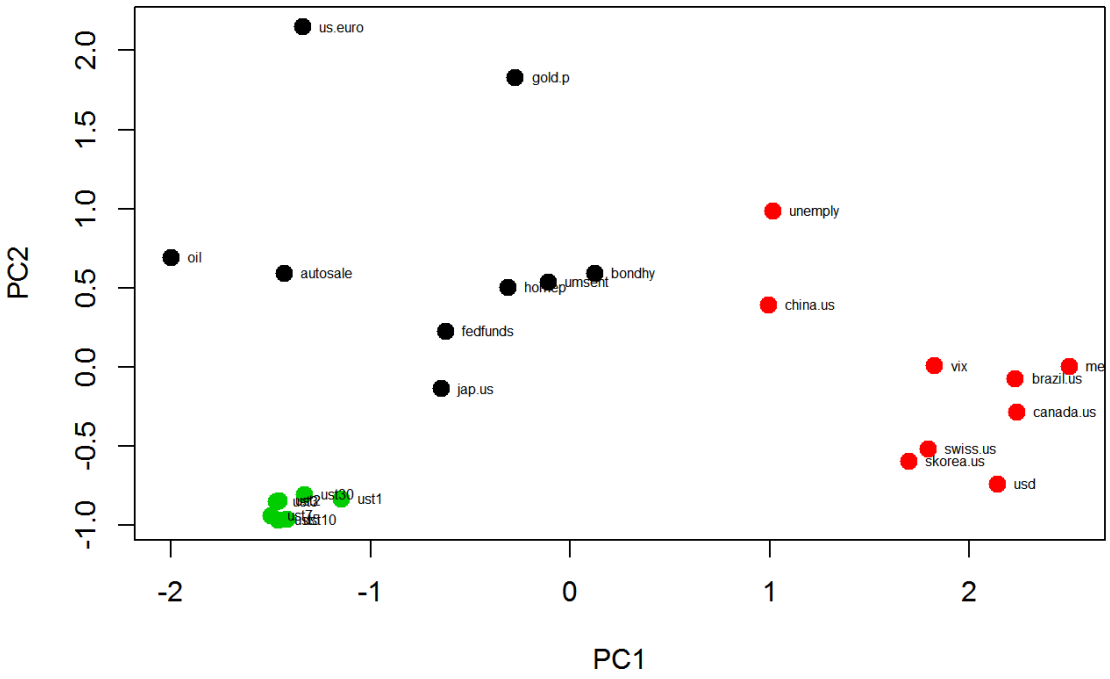
```
## [1] # Number of PCs needed to explain 25% of the variance: 1
## [1] # Number of PCs needed to explain 50% of the variance: 1
## [1] # Number of PCs needed to explain 75% of the variance: 2
## [1] # Number of PCs needed to explain 85% of the variance: 2
## [1] # Number of PCs needed to explain 95% of the variance: 6
## [1] ----------------------------------------------------------------
```



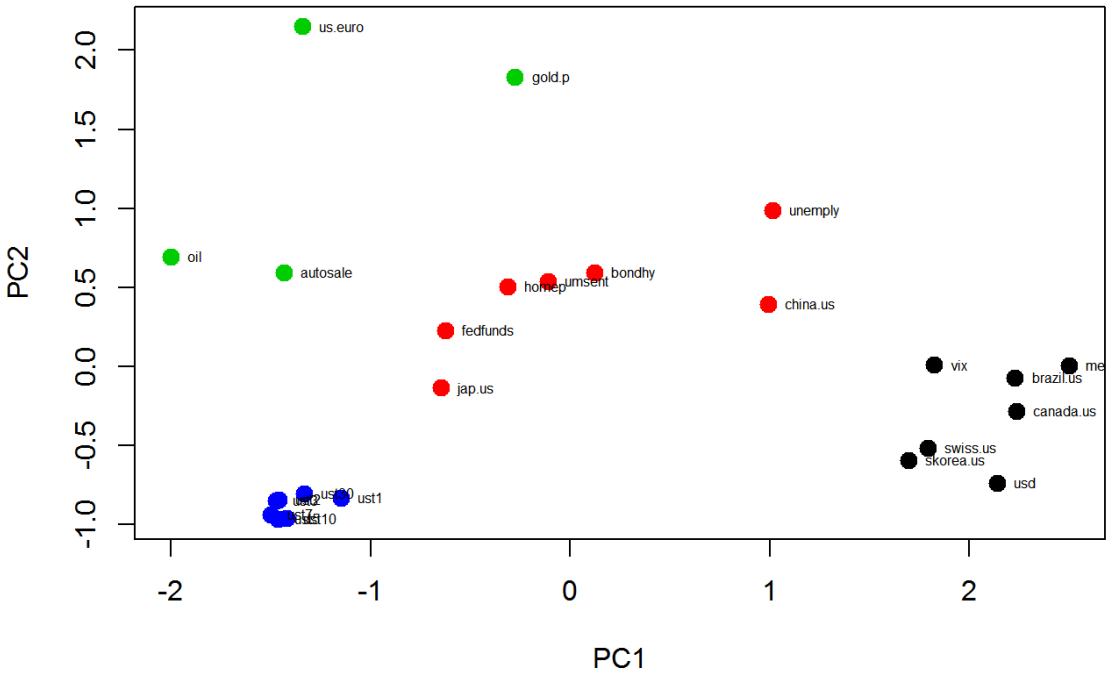



```
test.kmeans(ret.c, 3, 10, add.labels = TRUE)
```

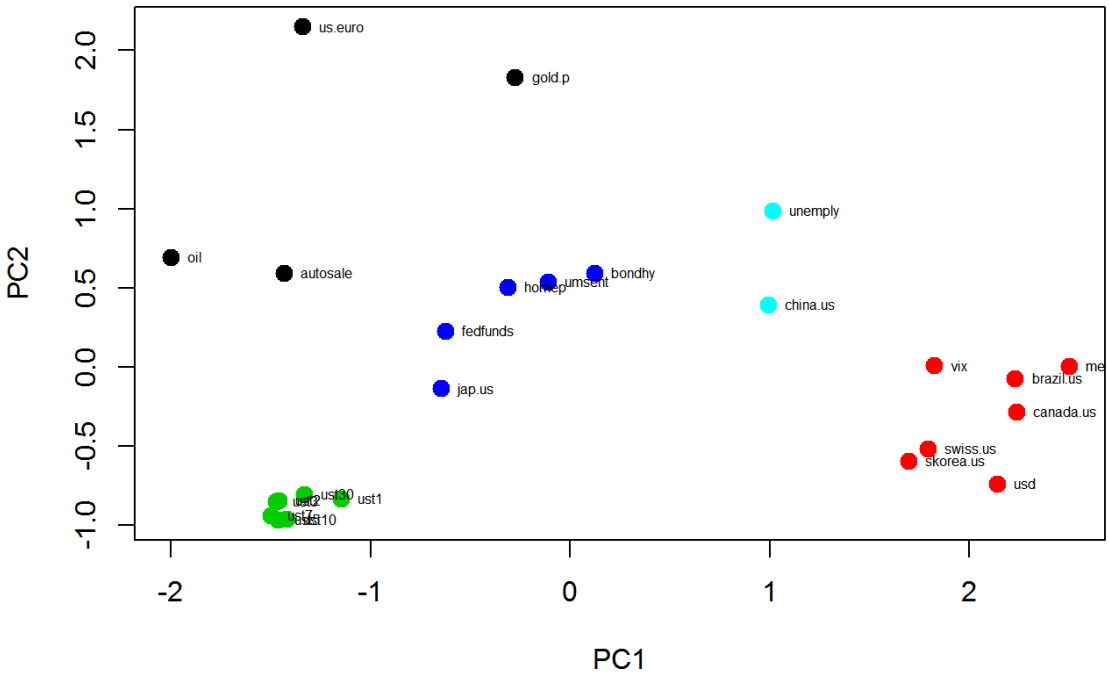
Kmean Clustering (Distance) with k = 3



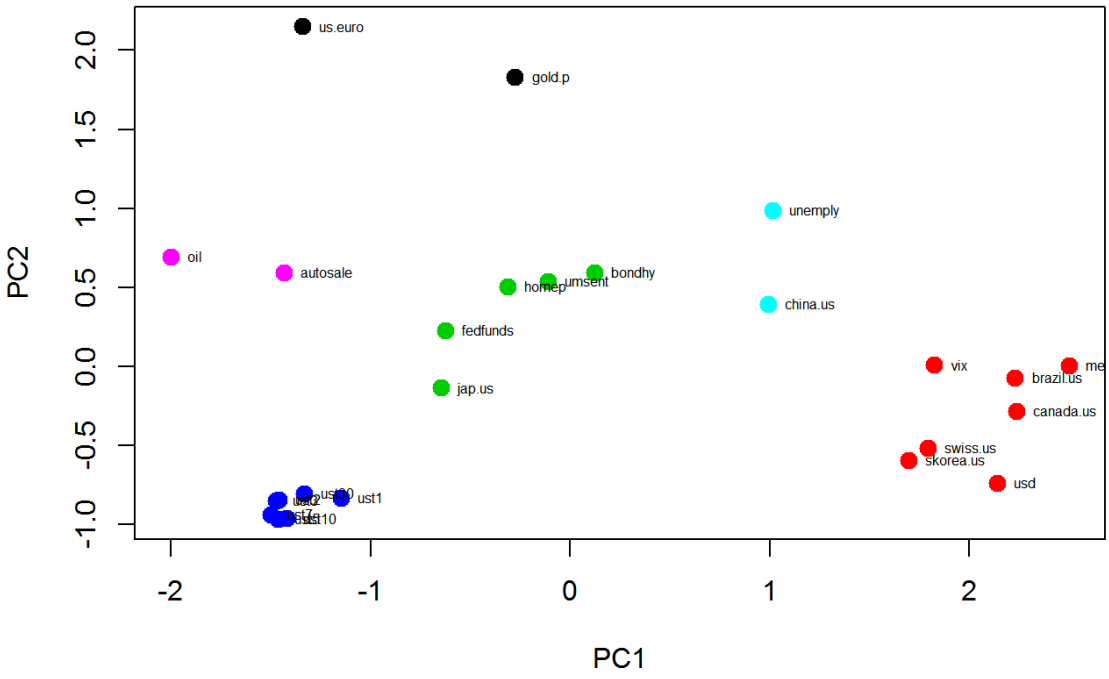
Kmean Clustering (Distance) with k = 4



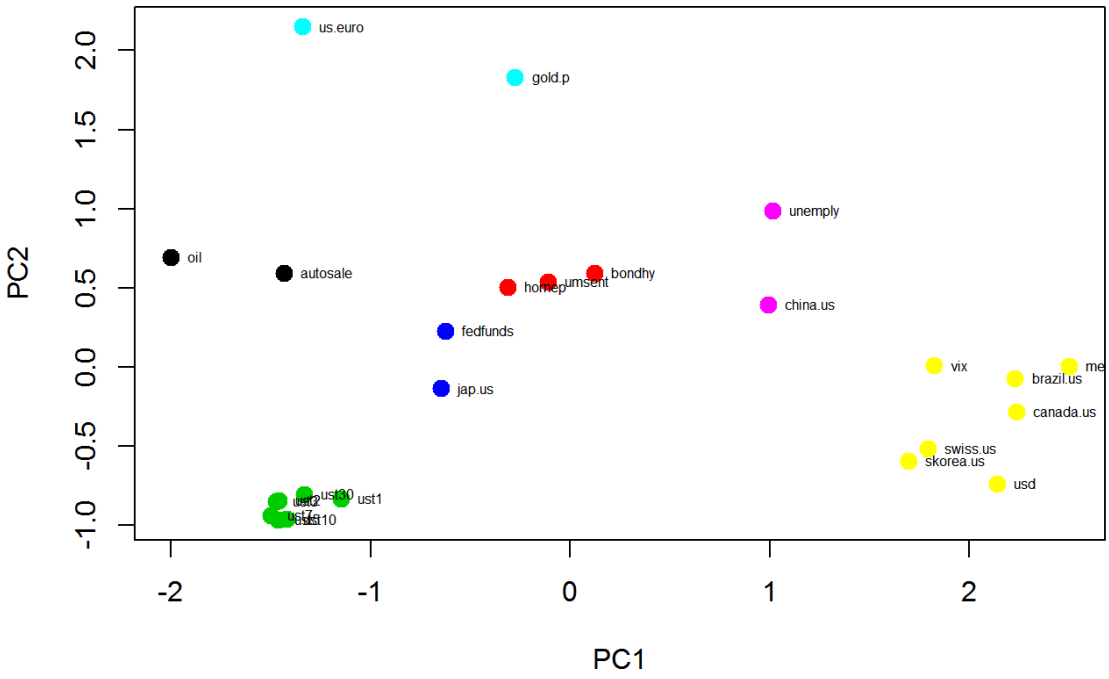
Kmean Clustering (Distance) with k = 5



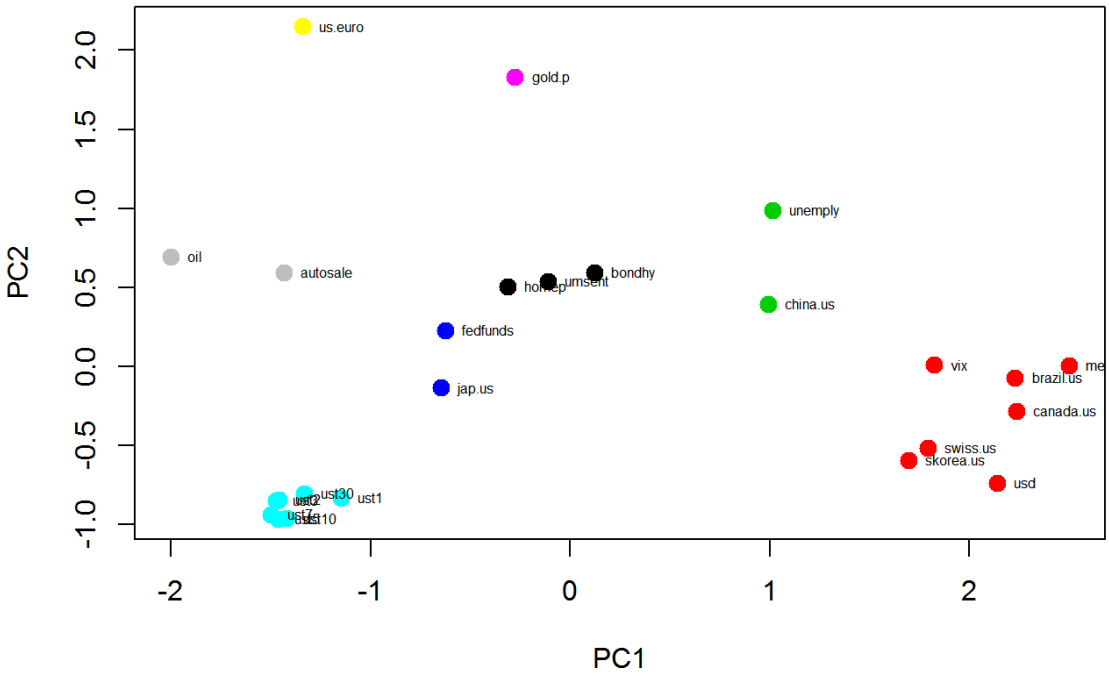
Kmean Clustering (Distance) with k = 6

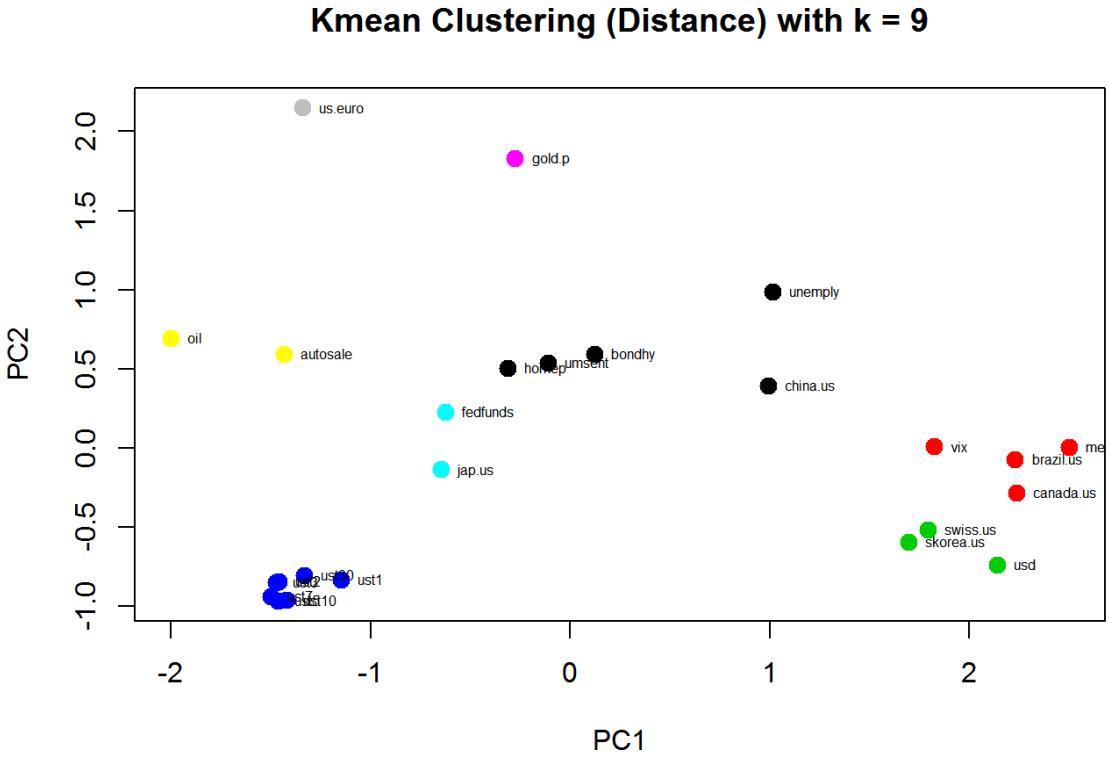


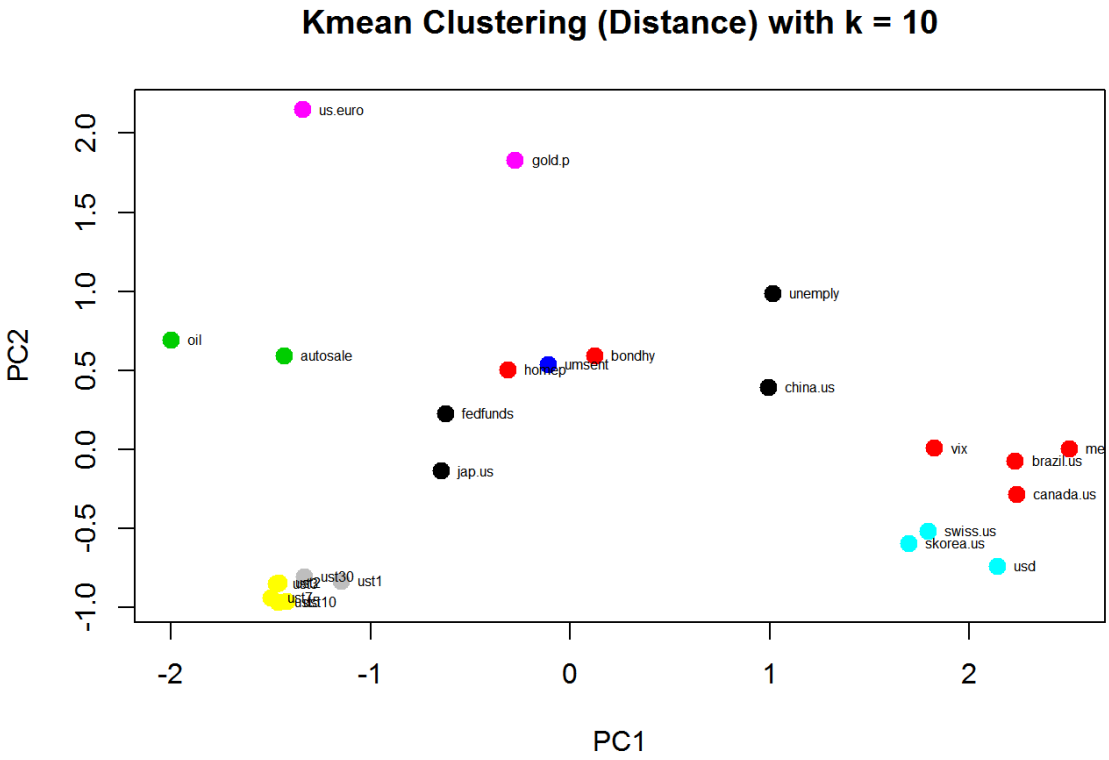
Kmean Clustering (Distance) with k = 7



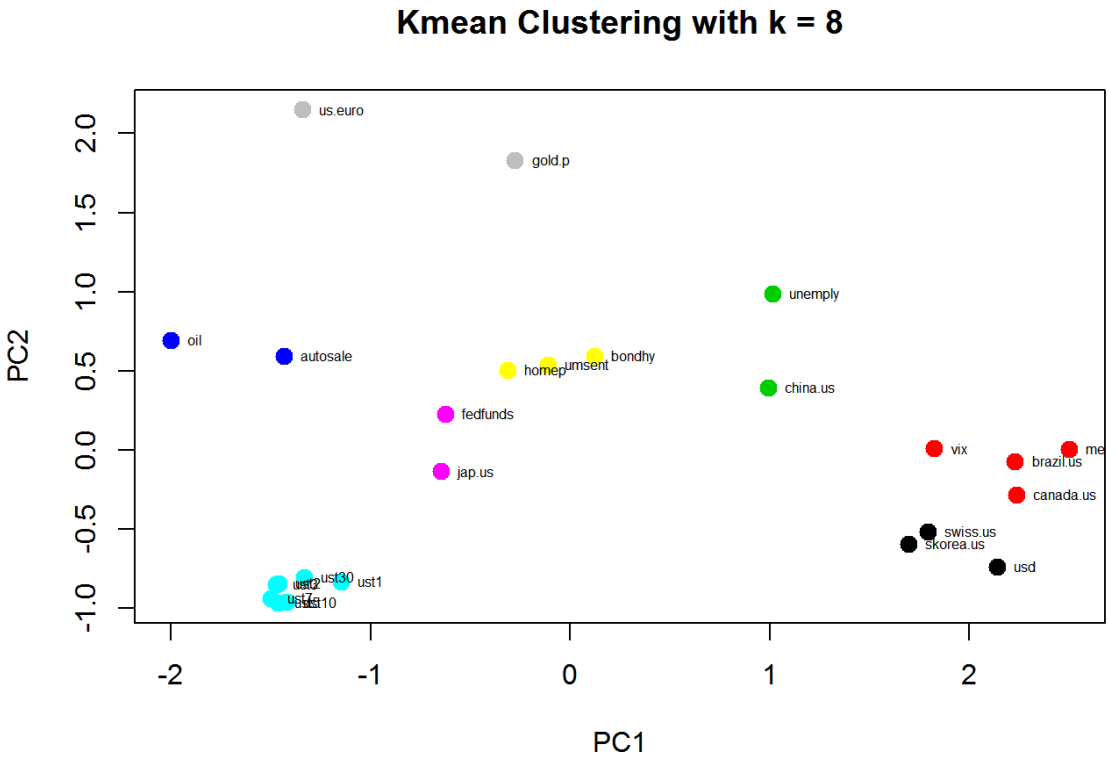
Kmean Clustering (Distance) with k = 8








```
# Selected Model (store cluster groups with their tags)
k = 8 ; k.out <- run.clustering(ret.c, k, "kmeans", max.elem.out = max.out, add.labels = TRUE)
```



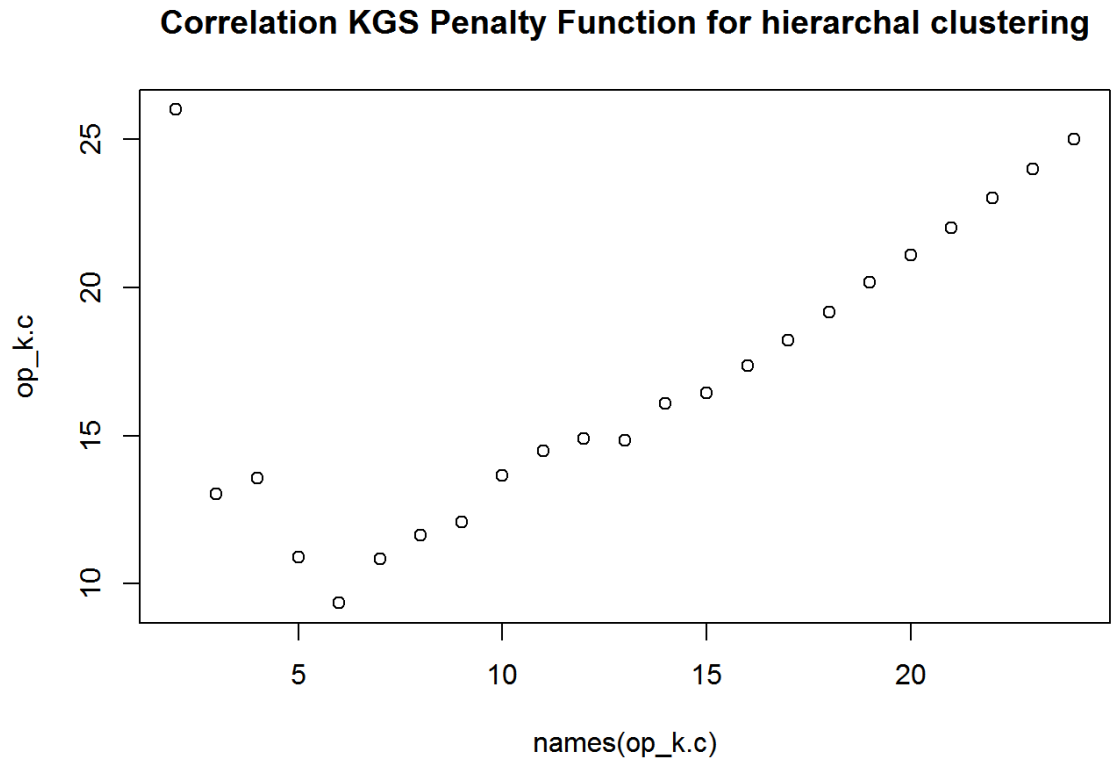

```
## [1] Cluster 1 contents (up to 8) :usd skorea.us swiss.us
## [1] Cluster 2 contents (up to 8) :vix brazil.us canada.us mexico.us
## [1] Cluster 3 contents (up to 8) :china.us unemply
## [1] Cluster 4 contents (up to 8) :oil autosale
## [1] Cluster 5 contents (up to 8) :ust1 ust2 ust3 ust5 ust7 ust10 ust30
## [1] Cluster 6 contents (up to 8) :jap.us fedfunds
## [1] Cluster 7 contents (up to 8) :umsent homep bondhy
## [1] Cluster 8 contents (up to 8) :gold.p us.euro
```

```
cluster.k.c <- k.out[[2]]
```

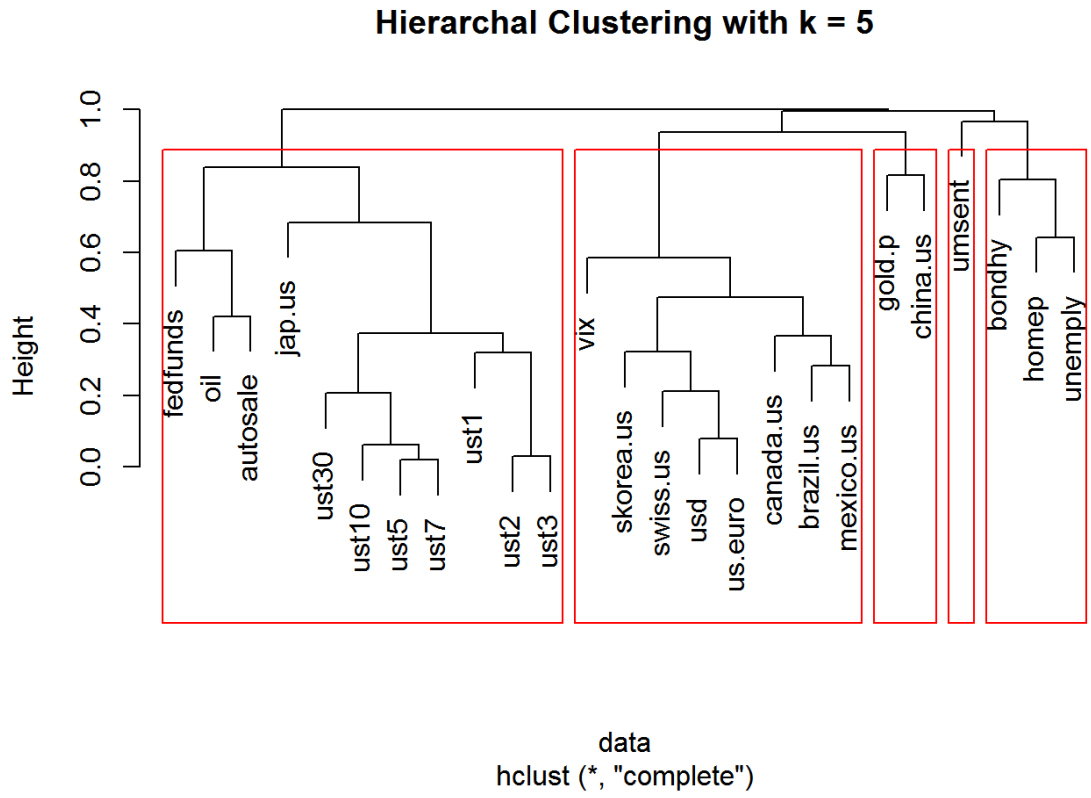
- (3.3): The next 2 tests are using the same kmeans and hierarcahcal clustering models as before but now we are analyzing the behaviors of the indicators by the distance between correlation coefficients of the data. The goal is to analyse a wider spectrum of possibilites for predictions and back testing. The PCA analysis looks better here and we test 3 - 10 clusters though the WSS plot did not offer us much clairty. Lower level clusters seemed to look good but the visualization analysis shows us that there are many variables spread out and more clusters will be need. Here is an example of how testng many models and creating a more automated procedure is a viable next step as these analysis techniues are open to interpretation. After much debate, we choose 8 clusters. The content analysis has changed but the grouoings are not bad and make sense under different scenarios of market conditions.

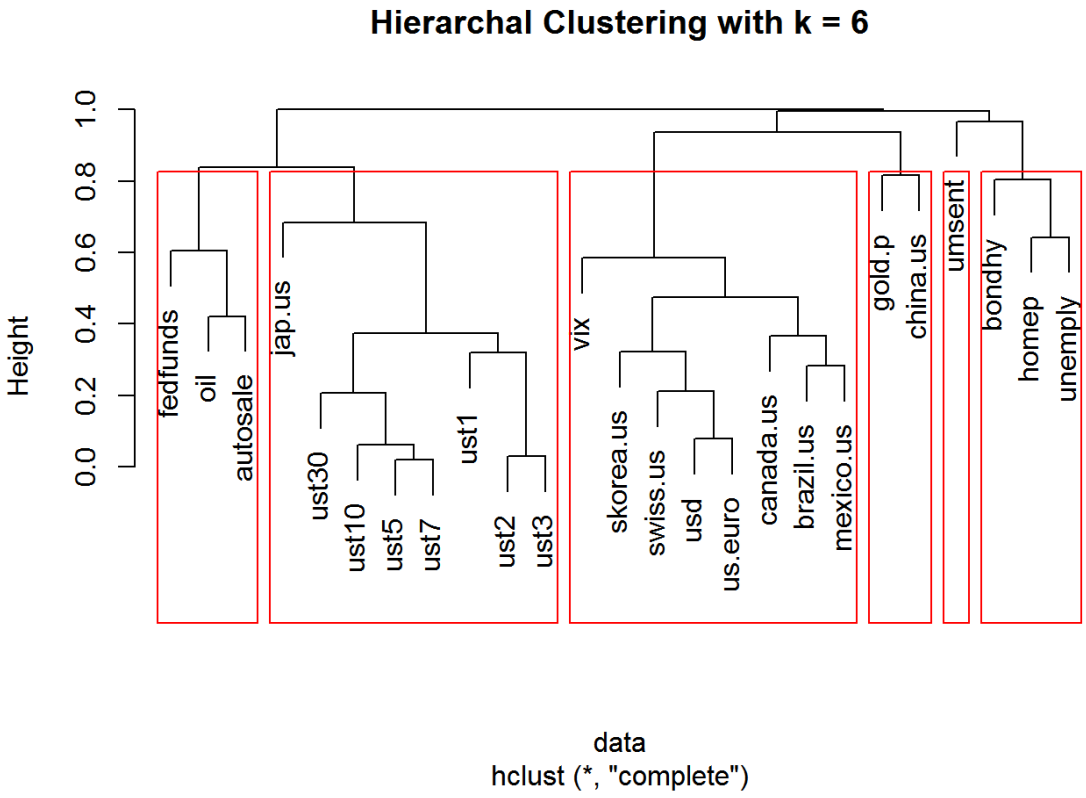
(3.4) Conduct Hierarchal Clustering (Corr. dist))

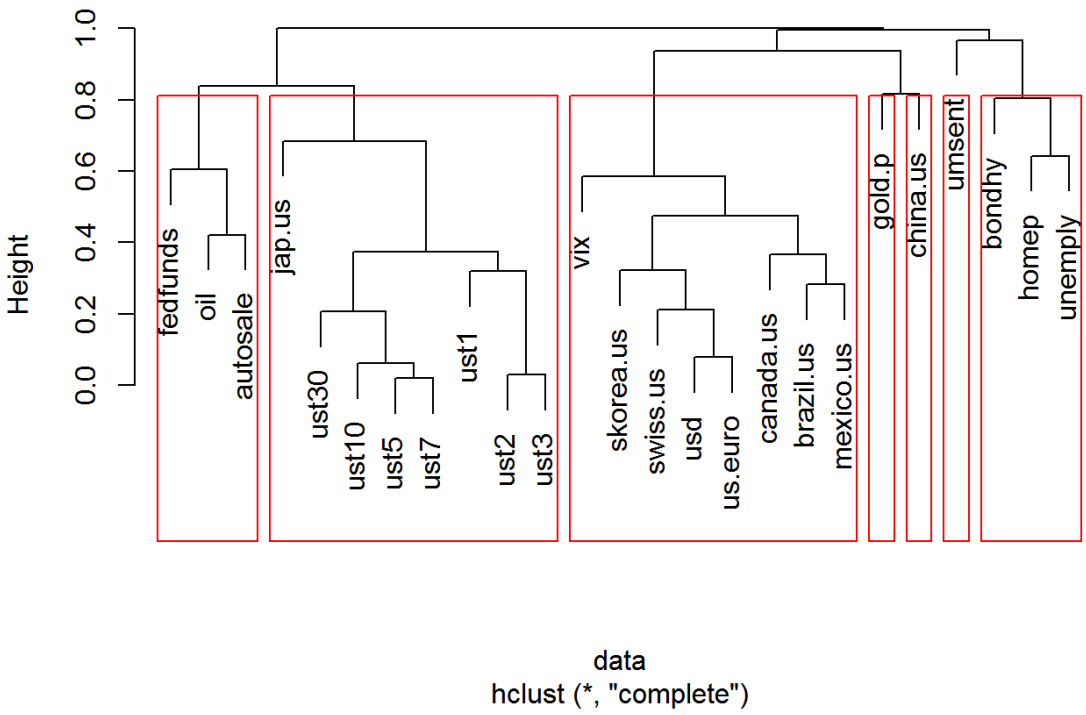
```
set.seed(1234)

### Prepare Data Set for modelling (Create a disimilarity matrix of the variable correlations)
diss = 1 - abs(ret.c) # /corr/ becasue some correlations are negative
dist = as.dist(diss)

hc   = hclust(dist, method = "complete")

## Analyze Data to interpret possible clusterings
op_k.c <- kgs(hc, dist, maxclust = nrow(ret.c)-1) # Min = optimal clusters (k)
plot(names(op_k.c), op_k.c, main = "Correlation KGS Penalty Function for hierarchal clustering")
```

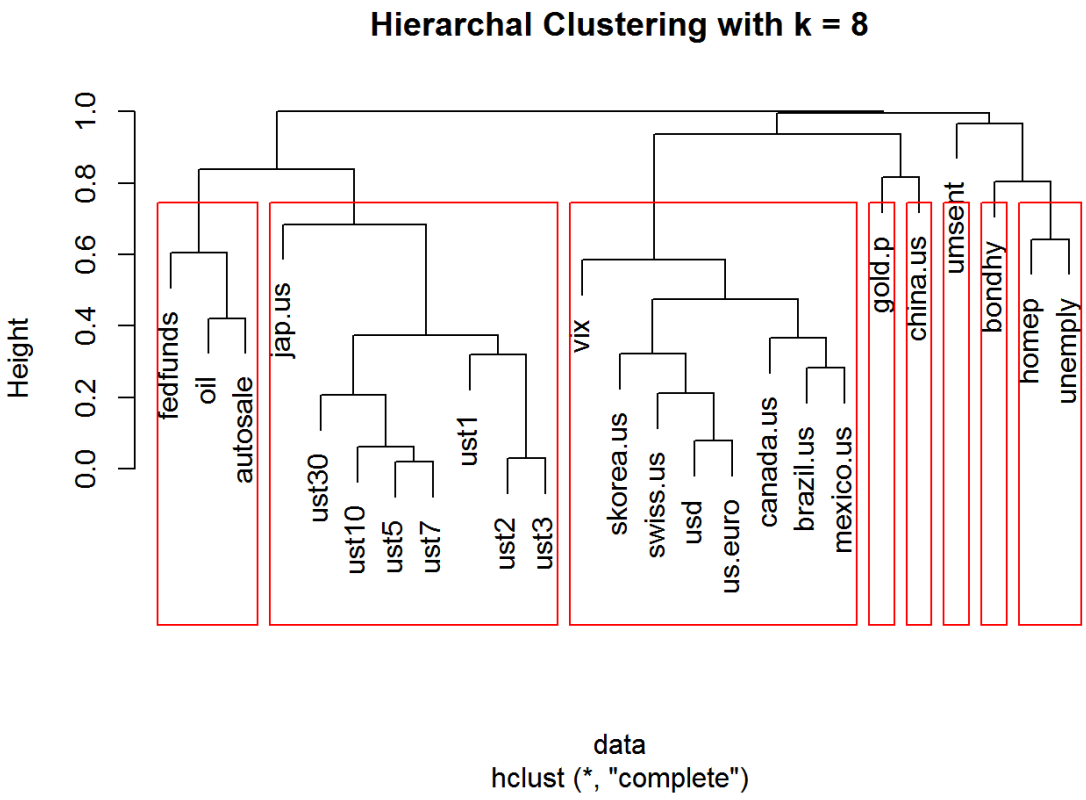


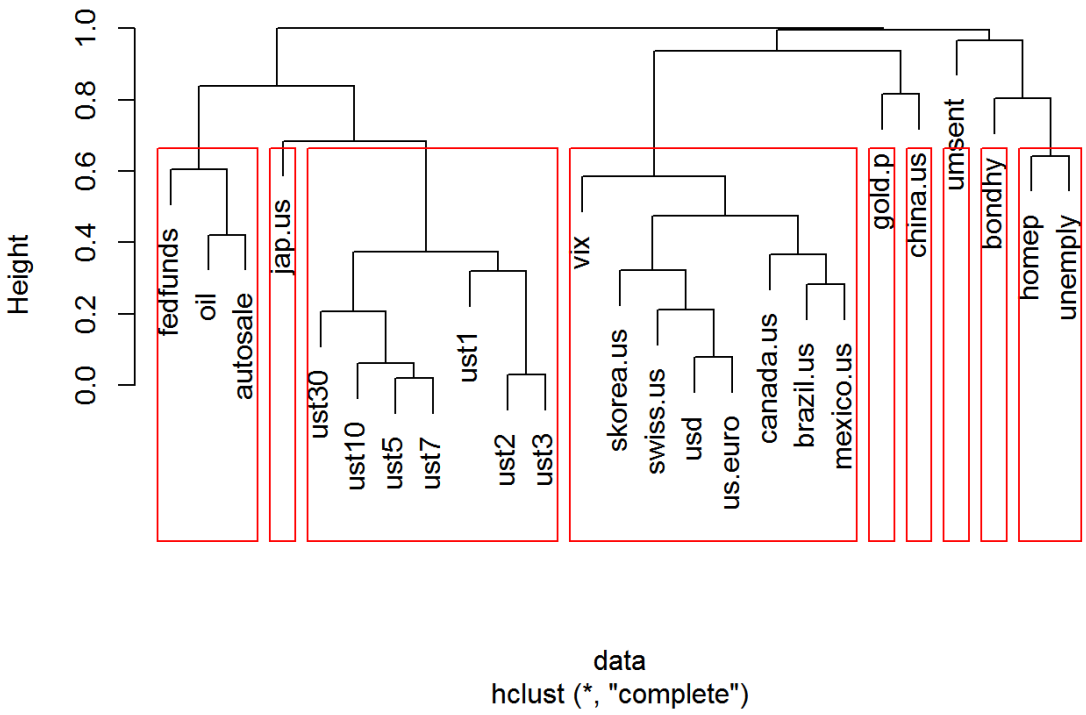

```
test.hc(dist, 5, 10)
```

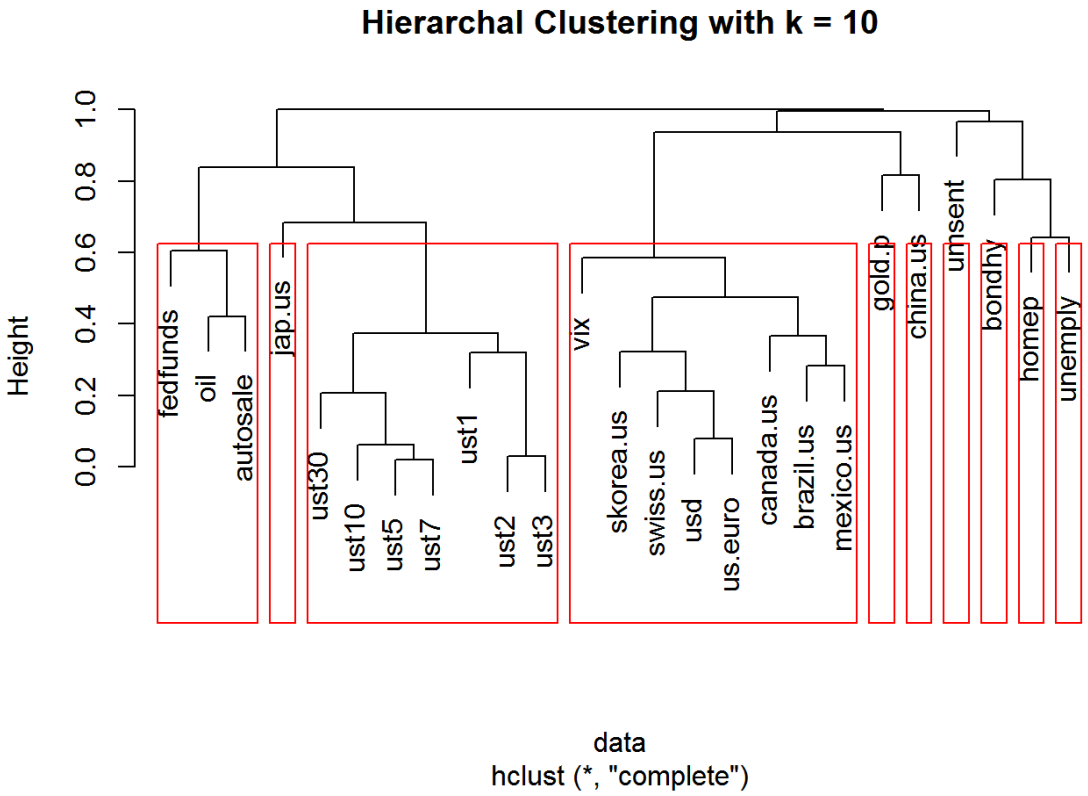



Hierarchal Clustering with k = 6
Height
0.0
0.2
0.4
0.6
0.8
1.0
fedfunds
oil
autosale
jap.us
ust30
ust10
ust5
ust7
ust1
ust2
ust3
vix
skorea.us
swiss.us
usd
us.euro
canada.us
brazil.us
mexico.us
gold.p
china.us
umsent
bondhy
homep
unemply
data
hclust (*, "complete")


Hierarchal Clustering with k = 7
Height
0.0
0.2
0.4
0.6
0.8
1.0
fedfunds
oil
autosale
jap.us
ust30
ust10
ust5
ust7
ust1
ust2
ust3
vix
skorea.us
swiss.us
usd
us.euro
canada.us
brazil.us
mexico.us
gold.p
china.us
umsent
bondhy
homep
unemply
data
hclust (*, "complete")


Hierarchal Clustering with k = 8
Height
0.0
0.2
0.4
0.6
0.8
1.0
fedfunds
oil
autosale
jap.us
ust30
ust10
ust5
ust7
ust1
ust2
ust3
vix
skorea.us
swiss.us
usd
us.euro
canada.us
brazil.us
mexico.us
gold.p
china.us
umsent
bondhy
homep
unemply
data
hclust (*, "complete")


Hierarchal Clustering with k = 9
Height
0.0
0.2
0.4
0.6
0.8
1.0
fedfunds
oil
autosale
jap.us
ust30
ust10
ust5
ust7
ust1
ust2
ust3
vix
skorea.us
swiss.us
usd
us.euro
canada.us
brazil.us
mexico.us
gold.p
china.us
umsent
bondhy
homep
unemply
data
hclust (*, "complete")




```
# Selected Model (store ordered groups with their tags)
k = 10 ; k.out <- run.clustering(dist, k, "hc", max.elem.out = max.out)
```

```
## [1] Cluster 1 contents (up to 8) :gold.p
## [1] Cluster 2 contents (up to 8) :oil fedfunds autosale
## [1] Cluster 3 contents (up to 8) :usd vix brazil.us canada.us mexico.us skorea.us swiss.us us.e
uro
## [1] Cluster 4 contents (up to 8) :china.us
## [1] Cluster 5 contents (up to 8) :jap.us
## [1] Cluster 6 contents (up to 8) :umsent
## [1] Cluster 7 contents (up to 8) :homep
## [1] Cluster 8 contents (up to 8) :unemply
## [1] Cluster 9 contents (up to 8) :bondhy
## [1] Cluster 10 contents (up to 8) :ust1 ust2 ust3 ust5 ust7 ust10 ust30
```

```
cluster.h.c <- k.out[[2]]

save(cluster.k.d, cluster.k.c, cluster.h.d, cluster.h.c, file = "FE582_ProjectClusters.RData")
```

- (3.4): The Hierarchal Clustering analysis for correlated data showed us good results. As explained before, we pick the best clusters and prune the tree to identify the the optimal analysis. After selecting 5 - 10 prunes, we selected 10 as the best model. There are many free floating compoenents here, as well clear groupings of FX rates and USt rates. This analysis is good and we now move on to putting these subsets to the test.

4. Create Final Data Subsets for each ETF based on the 4 classification methods (4x10 models)

```
# Create Clustered Data Sets across cluster groupings and within cluster correlation testing
clustered.data.all <- create.subs(obs.cor.all, cluster.k.d, cluster.h.d, cluster.k.c, cluster.h.c,
etf.class.train)
```

```
## [1] ------------------------------------------------------------------------
## [1] Step 1: Data Adjustments for input paramaters...
## [1] Step 2: Attaching Cluster Tags...
## [1] Step 3: Filtering parameter lists by best parameter and attempt colinearity removal by clus
ter...
## [1] Step 4: Creating Final Data Output list (List with data sets)...
```

```
clustered.data.sub <- create.subs(obs.cor.sub, cluster.k.d, cluster.h.d, cluster.k.c, cluster.h.c,
etf.class.train)
```

```
## [1] ------------------------------------------------------------------------
## [1] Step 1: Data Adjustments for input paramaters...
## [1] Step 2: Attaching Cluster Tags...
## [1] Step 3: Filtering parameter lists by best parameter and attempt colinearity removal by clus
ter...
## [1] Step 4: Creating Final Data Output list (List with data sets)...
```

```
print(noquote(paste("Data filtered per Cluster Technique per ETF i.e. (",paste(names(clustered.dat
a.sub$s.energy$kd),collapse=", "),")",sep="")))
```

```
## [1] Data filtered per Cluster Technique per ETF i.e. (s.energy, s.energymove, canada.us, oil, v
ix, ust10)
```

```
save(clustered.data.all, clustered.data.sub, file = "FE582_ClusteredDataSets.RData")
```

- (4): After the 4 clustering analyses are complete, the data needs to be filtered a second time and aggregated into a large data structure where for each ETF we store the relevant ETF value, classification factor and indicator data sets. The filtering is conducted by taking each cluster group for the current ETF and cluster technique used and selecting the most highly correlated elements from each seperate cluster. Then, if there are remianing memebrs in the same cluster, they are evaluated for their correlation with the chosen elem. If their correlation < 0.5, we keep these elems as well (le thn 50% correlation is the maximum threshold that is reasonabl allowed in Collinarity and redundancy testing that we have reserahced. These are done grouped by the cluster tags (clusters defined in analysis) so that we can build models across the ETFs awith repsect to the subsets dictated by the clusters.
- All data analysis is complete for model consruction. We will now build, predict and backtest our models to see how our hypothsis holds.

(5.1) Build Multi-Variate Linear Regression Models

```
# Build the Linear Models
models.lm.list.all <- build.lm(clustered.data.all, etf.class.train, sum = FALSE)
models.lm.list.sub <- build.lm(clustered.data.sub, etf.class.train, sum = FALSE)

# Glimpse at models
summary(models.lm.list.sub$s.hc$hd)
```

```
##
## Call:
## lm(formula = s.hc ~ vix + mexico.us, data = temp2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11742 -0.01176  0.00269  0.01505  0.04740
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.011184   0.003221   3.472  0.00089 ***
## vix         -0.102522   0.020100  -5.101 2.77e-06 ***
## mexico.us   -0.441283   0.111082  -3.973  0.00017 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02728 on 70 degrees of freedom
## Multiple R-squared:  0.6098, Adjusted R-squared:  0.5987
## F-statistic:  54.7 on 2 and 70 DF,  p-value: 4.953e-15
```

```
summary(models.lm.list.all$s.hc$hd)
```

```
##
## Call:
## lm(formula = s.hc ~ vix + mexico.us + ust1, data = temp2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09667 -0.01076  0.00308  0.01489  0.05123
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.009515   0.003211   2.963  0.00417 **
## vix         -0.096304   0.019700  -4.889 6.36e-06 ***
## mexico.us   -0.508177   0.111702  -4.549 2.24e-05 ***
## ust1        -0.046211   0.020121  -2.297  0.02468 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02648 on 69 degrees of freedom
## Multiple R-squared:  0.6375, Adjusted R-squared:  0.6218
## F-statistic: 40.45 on 3 and 69 DF,  p-value: 3.367e-15
```

```
save(models.lm.list.all, models.lm.list.sub, file = "FE582_LinearModels.RData")
```

- (5.1): The Multivariate Linear models are constructed in an automated function. The linear models are built for each cluster technique (x5) used for each ETF (x10) target acorss both correlation types (x2) and 2 model techniques (Linear Modelling and Descion trees) (x2). Overall, that means that 200 models are made and tested for each prediction model type. Above, we can see the model summaries for the S.hc ETF using data clustered by hierarchal clusterings. The model made with the non-subsets correlation seems to have a better model fit but the significne of the terms is waning in the model made with no intial correlation sgubsetting. The models are stored for predictions.

(5.2) Predict using the Linear Models

```
## Generate the predictions on the testing sets with directional movement classifications
pred.out <- pred.lm(models.lm.list.all, etf.class.test)
models.lm.all.pred <- pred.out[[1]]
models.lm.all.comp <- pred.out[[2]]

pred.out <- pred.lm(models.lm.list.sub, etf.class.test)
models.lm.sub.pred <- pred.out[[1]]
models.lm.sub.comp <- pred.out[[2]]

save(models.lm.all.pred,models.lm.all.comp,models.lm.sub.pred,models.lm.sub.comp, file = "FE582_LM
Predictions.RData")

# Glimpse at Model Accuracies for predicting on the data sets
kable(data.frame(models.lm.all.comp,"Mean"=rowMeans(models.lm.all.comp)), caption = "LR models, co
rrelation > 0.00")
```

LR models, correlation > 0.00

	s.bio	s.cd	s.cs	s.energy	s.fin	s.hc	s.ind	s.mat	s.tech	s.util	Mean
kd	58.33333	69.44444	61.11111	83.33333	58.33333	63.88889	69.44444	83.33333	75.00000	61.11111	68.33333
hd	58.33333	75.00000	63.88889	83.33333	58.33333	63.88889	69.44444	80.55556	75.00000	63.88889	69.16667
kc	52.77778	58.33333	47.22222	86.11111	63.88889	58.33333	69.44444	77.77778	63.88889	55.55556	63.33333
hc	50.00000	55.55556	36.11111	86.11111	58.33333	66.66667	55.55556	72.22222	52.77778	63.88889	59.72222
fulldata	69.44444	69.44444	58.33333	80.55556	63.88889	63.88889	69.44444	77.77778	80.55556	55.55556	68.88889

```
kable(data.frame(models.lm.sub.comp,"Mean"=rowMeans(models.lm.sub.comp)), caption = "LR models, co
rrelation >= 0.30")
```

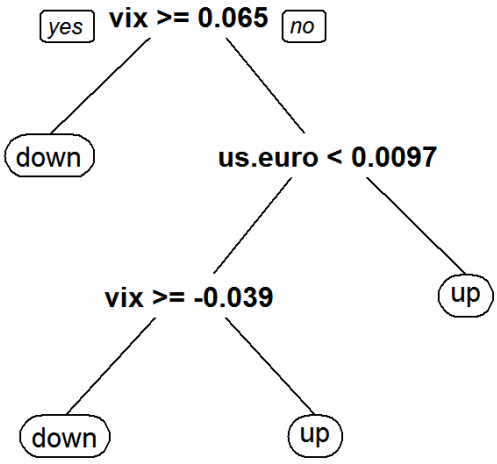
LR models, correlation >= 0.30

	s.bio	s.cd	s.cs	s.energy	s.fin	s.hc	s.ind	s.mat	s.tech	s.util	Mean
kd	58.33333	66.66667	61.11111	83.33333	58.33333	58.33333	69.44444	77.77778	77.77778	58.33333	66.94444
hd	58.33333	75.00000	61.11111	83.33333	58.33333	58.33333	69.44444	83.33333	77.77778	58.33333	68.33333
kc	52.77778	55.55556	44.44444	86.11111	61.11111	66.66667	66.66667	77.77778	61.11111	52.77778	62.50000
hc	50.00000	61.11111	47.22222	86.11111	52.77778	66.66667	75.00000	77.77778	55.55556	61.11111	63.33333
fulldata	69.44444	69.44444	58.33333	80.55556	63.88889	63.88889	69.44444	77.77778	80.55556	55.55556	68.88889

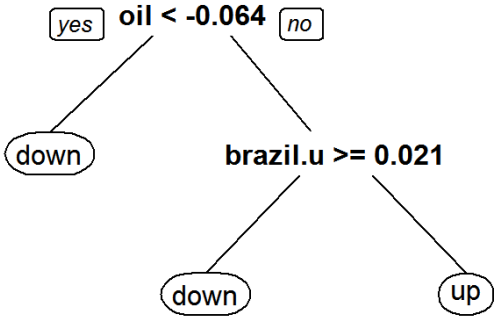
- (5.2): Here we simply predict the classification results and compare them t the actual reusult of the market. This gets us an accuracy Matrix that allows us to take a look at the performance of the clusters against each other and the clustered groups againstg the “FULL DATA” testing study. From this intial analysis, we conlcude that subsetting the data by correlation >= 0.30 had little effect on the overall accuracry of the clustering techniques. This can also be said for comparing the full model testing accuracy to the the clustered tests. The RowMeans() column to the right of the matrices shows the agreggate results are faily close. When we look at the individual accuracies, we see that the Energy and Materials sectors showed the most correct prediciton clalcuations wuth Tech not too far behind. For the Healthcare setctor, the accuracy for using all data with a Hierarchal Clusterng Model was 55%, but the subset equivalent accuracy was 75%. That seems to be an outlier here. For the linear models, it would seem that subsetting the data or using clustering models tend to do worse than the models using more intial data. This makes sense when we think about how stepwise functions operate as well as the numerous tests needed to verify best orders. With that said, we need to analyze the market performance in the back testing stage.

(6.1) Construct Decision Trees

```
# Build the Decision Trees
models.dt.list.all <- build.dt(clustered.data.all, etf.class.train, FALSE, FALSE)
models.dt.list.sub <- build.dt(clustered.data.sub, etf.class.train, FALSE, FALSE)

# Inspect Trees
prp(models.dt.list.sub$s.hc$kc)
```




```
prp(models.dt.list.sub$s.energy$fulldata)
```



```
models.dt.list.sub$s.cd$kd$variable.importance
```

```
##       vix   us.euro skorea.us
## 11.787462  3.929154  1.746291
```

```
save(models.dt.list.all,models.dt.list.sub, file = "FE582_DecisionTreeModels.RData")
```

- (6.1): Creating the decision tree using rpart() was a very familiar procedure to building the linear models. The trees are built on the the correlation and cluster subset datsets. The majority of the trees are very small, only 2 or 3 leaves maximum. We have shown some small examples if the trees and the variable importance results from the a tree built using Kmean Distance with corr. subset data. The treescuse factor tests to evaluate splitting points for data based on 'yes'/'no' in terms of a conndtional (numeruical or classification) decision. These models are good ones to use becuase they can evaluate factors and data to create paths to follow. In the real world, tree models can be very useful to keep growing as the values or the conditions chnage. The models hvave been stored for predictions.

(6.2) Predict using the Descion Trees

```
## Generate the predictions on the testing sets with directional movement classifications
pred.out <- pred.dt(models.dt.list.all, etf.class.test)
models.dt.all.pred <- pred.out[[1]]
models.dt.all.comp <- pred.out[[2]]

pred.out <- pred.dt(models.dt.list.sub, etf.class.test)
models.dt.sub.pred <- pred.out[[1]]
models.dt.sub.comp <- pred.out[[2]]

save(models.dt.all.pred,models.dt.all.comp,models.dt.sub.pred,models.dt.sub.comp, file = "FE582_DT
Predictions.RData")

# Glimpse at Model Accuracies for predicting on the data sets
kable(data.frame(models.dt.all.comp,"Mean"=rowMeans(models.dt.all.comp)), caption = "DT models, co
rrelation > 0.00")
```

DT models, correlation > 0.00

	s.bio	s.cd	s.cs	s.energy	s.fin	s.hc	s.ind	s.mat	s.tech	s.util	Mean
kd	63.88889	75.00000	66.66667	86.11111	52.77778	61.11111	69.44444	66.66667	80.55556	63.88889	68.61111
hd	63.88889	72.22222	66.66667	86.11111	72.22222	61.11111	69.44444	66.66667	80.55556	63.88889	70.27778
kc	47.22222	58.33333	55.55556	86.11111	52.77778	55.55556	63.88889	58.33333	58.33333	52.77778	58.88889
hc	63.88889	50.00000	55.55556	86.11111	66.66667	52.77778	63.88889	75.00000	58.33333	63.88889	63.61111
fulldata	58.33333	47.22222	66.66667	72.22222	61.11111	52.77778	69.44444	66.66667	61.11111	58.33333	61.38889

```
kable(data.frame(models.dt.sub.comp,"Mean"=rowMeans(models.dt.sub.comp)), caption = "DT models, co
rrelation >= 0.30")
```

DT models, correlation >= 0.30

	s.bio	s.cd	s.cs	s.energy	s.fin	s.hc	s.ind	s.mat	s.tech	s.util	Mean
kd	66.66667	77.77778	66.66667	86.11111	52.77778	61.11111	69.44444	66.66667	80.55556	69.44444	69.72222
hd	63.88889	77.77778	66.66667	86.11111	72.22222	61.11111	69.44444	66.66667	80.55556	63.88889	70.83333
kc	47.22222	61.11111	50.00000	86.11111	52.77778	63.88889	75.00000	58.33333	58.33333	63.88889	61.66667
hc	44.44444	61.11111	44.44444	86.11111	47.22222	61.11111	75.00000	75.00000	58.33333	63.88889	61.66667
fulldata	58.33333	47.22222	66.66667	72.22222	61.11111	52.77778	69.44444	66.66667	61.11111	58.33333	61.38889

- (6.2): The Decision Tree Predicitions immediately stand out because the averages of the rows show us thatg the Kmeans and Hierarchal clusterting techniques using Euclidean Distance (not correlation) seemd to clearly outporm by looking at the averages of the models, but the best ETFs to model were Energy, Consumer Discrtionary and Tech for these models. The Full Data models in these cases seemed to far worse than the Full Data models in the Linear models case. These accuracy analyses have shown us thst the decision trees may be better performing models and models built with 1 of the 4 actual modelling techniques tend to do better as well than

7. Conduct Backtesting on testing data sets

```
### Set Capital for test (split evenly among ETFs in back testing)
capital = 10000000

## Run back tests on the "base" portfolios (Holding the market and holding (long) a basket of all
10 etfs)
sim.djia  <- sim.hold(djia, "vector", "DJIA Index", capital, print.sum = TRUE)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: DJIA Index
## [1] Back Test over 36 periods yielded a 24.76 % Return
```

```
sim.sp500 <- sim.hold(sp500, "vector", "SP500 Index", capital, print.sum = TRUE)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: SP500 Index
## [1] Back Test over 36 periods yielded a 21.74 % Return
```

```
sim.etfs  <- sim.hold(etf.class.test, "list", "Holding Long all (10) Sector ETFs",
                      list.loc = 2, add.cum = TRUE, capital, print.sum = TRUE, print.all = TRUE)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: Holding Long all (10) Sector ETFs
## [1] Back Test (s.bio) over 36 periods yielded a 26.4 % Return
## [1] Back Test (s.cd) over 36 periods yielded a 37.54 % Return
## [1] Back Test (s.cs) over 36 periods yielded a 30.92 % Return
## [1] Back Test (s.energy) over 36 periods yielded a -33.28 % Return
## [1] Back Test (s.fin) over 36 periods yielded a 74.94 % Return
## [1] Back Test (s.hc) over 36 periods yielded a 32.8 % Return
## [1] Back Test (s.ind) over 36 periods yielded a 31.26 % Return
## [1] Back Test (s.mat) over 36 periods yielded a 10.62 % Return
## [1] Back Test (s.tech) over 36 periods yielded a 46.57 % Return
## [1] Back Test (s.util) over 36 periods yielded a 26.57 % Return
## [1] Cumulative Results over 36 periods yielded a 28.43 % Return
```

```
# Run back tests on Linear Models and Decision Trees
sim.lm.all <- pred.sim(models.lm.all.pred, etf.class.test, "ETF Linear Models (abs(Cor) >= 0)", ca
pital)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: ETF Linear Models (abs(Cor) >= 0)
## [1] --------------------------------------------------------------------------
## [1] Cumulative Results using (kd) yielded a 84.15 %
## [1] Cumulative Results using (hd) yielded a 86.42 %
## [1] Cumulative Results using (kc) yielded a 68.52 %
## [1] Cumulative Results using (hc) yielded a 62.75 %
## [1] Cumulative Results using (fulldata) yielded a 112.55 %
```

```
sim.lm.sub <- pred.sim(models.lm.sub.pred, etf.class.test, "ETF Linear Models (abs(Cor) >= 0.30)",
capital)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: ETF Linear Models (abs(Cor) >= 0.30)
## [1] --------------------------------------------------------------------------
## [1] Cumulative Results using (kd) yielded a 78.41 %
## [1] Cumulative Results using (hd) yielded a 81.17 %
## [1] Cumulative Results using (kc) yielded a 67.35 %
## [1] Cumulative Results using (hc) yielded a 66.7 %
## [1] Cumulative Results using (fulldata) yielded a 112.55 %
```

```
sim.dt.all <- pred.sim(models.dt.all.pred, etf.class.test, "ETF Decision Tree Models (abs(Cor) >= 0)", capital)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: ETF Decision Tree Models (abs(Cor) >= 0)
## [1] --------------------------------------------------------------------------
## [1] Cumulative Results using (kd) yielded a 98.2 %
## [1] Cumulative Results using (hd) yielded a 111.29 %
## [1] Cumulative Results using (kc) yielded a 45.75 %
## [1] Cumulative Results using (hc) yielded a 72.23 %
## [1] Cumulative Results using (fulldata) yielded a 51.71 %
```

```
sim.dt.sub <- pred.sim(models.dt.sub.pred, etf.class.test, "ETF Decision Tree Models (abs(Cor) >= 0.30)", capital)
```

```
## [1] --------------------------------------------------------------------------
## [1] Running back test for: ETF Decision Tree Models (abs(Cor) >= 0.30)
## [1] --------------------------------------------------------------------------
## [1] Cumulative Results using (kd) yielded a 112.26 %
## [1] Cumulative Results using (hd) yielded a 111.8 %
## [1] Cumulative Results using (kc) yielded a 54.82 %
## [1] Cumulative Results using (hc) yielded a 56.08 %
## [1] Cumulative Results using (fulldata) yielded a 51.71 %
```

```
save(sim.djia,sim.sp500,sim.etfs, file = "FE582_BacktestsBaseLine.RData")
save(sim.lm.all,sim.lm.sub,sim.dt.all,sim.dt.sub, file = "FE582_BacktestsModels.RData")
```

- All model output has been aggregated into a very large data structure with predicitons, accuracies and confuision matrices, etc… We will use some of this to better compare the performance of our models as well move from accuracy to return percentages, which is the truly determinat factor of model success. Before we do that, we can see above that based on our modelling techniques, over the testing period of 3 years (36 months), all clustering and correlation subset variant models outperformed market portfolios. The backtests for cumulative returns were conducted by Model Type and Cluster Type, meaning that 20 out of 20 portfolios constructed outperformred the Dow Jones Industrial Avverage, SP500 and the long portfolio for the 10 ETFS.

8. More Indepth Model Comparsion

```
source("FE582_Project_Functions.R")

capital = 10000000

model.p <- list(lm.all=models.lm.all.pred,dt.all=models.dt.all.pred,lm.sub=models.lm.sub.pred,dt.sub=models.dt.sub.pred)
model.c <- list(lm.all=models.lm.all.comp,dt.all=models.dt.all.comp,lm.sub=models.lm.sub.comp,dt.sub=models.dt.sub.comp)
model.s <- list(lm.all=sim.lm.all,        dt.all=sim.dt.all,        lm.sub=sim.lm.sub,        dt.sub=sim.dt.sub)

full.comp.data <- cum.dat(model.p, model.c, model.s, etf.class.test, capital, print = FALSE)
```

```
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
## [1] --------------------------------------------------------------------------------
```

```
save(full.comp.data, file = "FE582_ComparableModelResults.RData")

lma.comp <- full.comp.data$indcomp$lm.all$cumbycluster
lms.comp <- full.comp.data$indcomp$lm.sub$cumbycluster

dta.comp <- full.comp.data$indcomp$dt.all$cumbycluster
dts.comp <- full.comp.data$indcomp$dt.sub$cumbycluster
```

8. Analyze Linear Models and Descion Trees against each other and the clustered models

```
### Compare Performance of Cumulative returns

kable(full.comp.data$cumperf$performance, caption = "Cluster By ETF MODEL Comparision")
```

Cluster By ETF MODEL Comparision

	lm.all	dt.all	lm.sub	dt.sub
kd	84.15	98.2	78.41	112.26
hd	86.42	111.29	81.17	111.8
kc	68.52	45.75	67.35	54.82
hc	62.75	72.23	66.7	56.08
fulldata	112.55	51.71	112.55	51.71

```
## Performance:

# Linear Models
kable(lma.comp$performance,  caption = "performance  Table - LM (Corr >= 0)")
```

performance Table - LM (Corr >= 0)

	Accuracy	Return (%)	Return ($)
kd	68.89	112.55	21255000
hd	69.17	86.42	18642000
kc	68.33	84.15	18415000
hc	63.33	68.52	16852000
fulldata	59.72	62.75	16275000

```
kable(lms.comp$performance,  caption = "performance n Table - LM (Corr >= 0.30)")
```

performance n Table - LM (Corr >= 0.30)

	Accuracy	Return (%)	Return ($)
kd	68.89	112.55	21255000

	Accuracy	Return (%)	Return ($)
hd	66.39	81.07	16270000
kc	66.94	78.41	17841000
hc	62.50	67.35	16735000
fulldata	63.33	66.70	16670000

```
# Decision Trees
kable(dta.comp$performance, caption = "performance  Table - DT (Corr >= 0)")
```

performance Table - DT (Corr >= 0)

	Accuracy	Return (%)	Return ($)
kd	70.28	111.29	21129000
hd	68.61	98.20	19820000
kc	63.61	72.23	17223000
hc	61.39	51.71	15171000
fulldata	58.89	45.75	14575000

```
kable(dts.comp$performance, caption = "performance  Table - DT (Corr >= 0.30)")
```

performance Table - DT (Corr >= 0.30)

	Accuracy	Return (%)	Return ($)
kd	69.72	112.26	21226000
hd	70.83	111.80	21180000
kc	61.67	56.08	15608000
hc	61.67	54.82	15482000
fulldata	61.39	51.71	15171000

```
## Performance:

# Linear Models
kable(lma.comp$confusion,   caption = "Confusion Table - LM (Corr >= 0)")
```

Confusion Table - LM (Corr >= 0)

	Down	Up	Error-Down	Error-Up
Actual	156.00	204.00	0.00	0.00
kd	103.00	143.00	33.97	29.90
hd	107.00	142.00	31.41	30.39
kc	96.00	132.00	38.46	35.29
hc	88.00	127.00	43.59	37.75
fulldata	108.00	140.00	30.77	31.37
% total	0.43	0.57	0.00	0.00
Mean	-0.03	0.04	32.94	35.64
Std.	0.04	0.03	3.42	5.37

```
kable(lms.comp$confusion,   caption = "Confusion Table - LM (Corr >= 0.30)")
```

Confusion Table - LM (Corr >= 0.30)

	Down	Up	Error-Down	Error-Up
Actual	156.00	204.00	0.00	0.00
kd	105.00	136.00	32.69	33.33
hd	105.00	141.00	32.69	30.88
kc	97.00	128.00	37.82	37.25
hc	98.00	130.00	37.18	36.27
fulldata	108.00	140.00	30.77	31.37
% total	0.43	0.57	0.00	0.00
Mean	-0.03	0.04	33.82	34.23
Std.	0.04	0.03	2.86	3.09

```
# Decision Trees
kable(dta.comp$confusion,   caption = "Confusion Table - DT (Corr >= 0)")
```

Confusion Table - DT (Corr >= 0)

	Down	Up	Error-Down	Error-Up
Actual	156.00	204.00	0.00	0.00
kd	105.00	142.00	32.69	30.39
hd	107.00	146.00	31.41	28.43
kc	79.00	133.00	49.36	34.80
hc	84.00	145.00	46.15	28.92
fulldata	84.00	137.00	46.15	32.84
% total	0.43	0.57	0.00	0.00
Mean	-0.03	0.04	31.08	41.15
Std.	0.04	0.03	2.70	8.42

```
kable(dts.comp$confusion,   caption = "Confusion Table - DT (Corr >= 0.30)")
```

Confusion Table - DT (Corr >= 0.30)

	Down	Up	Error-Down	Error-Up
Actual	156.00	204.00	0.00	0.00
kd	107.00	144.00	31.41	29.41
hd	112.00	143.00	28.21	29.90
kc	90.00	132.00	42.31	35.29
hc	93.00	129.00	40.38	36.76
fulldata	84.00	137.00	46.15	32.84
% total	0.43	0.57	0.00	0.00

	Down	Up	Error-Down	Error-Up
Mean	-0.03	0.04	32.84	37.69
Std.	0.04	0.03	3.23	7.57

```
### Compare Accuracies of each model
kable(lma.comp$acctable,    caption = "Accuracy - LM (Corr >= 0)")
```

Accuracy - LM (Corr >= 0)

	s.bio	s.cd	s.cs	s.energy	s.fin	s.hc	s.ind	s.mat	s.tech	s.util
kd	58.33	69.44	61.11	83.33	58.33	63.89	69.44	83.33	75.00	61.11
hd	58.33	75.00	63.89	83.33	58.33	63.89	69.44	80.56	75.00	63.89
kc	52.78	58.33	47.22	86.11	63.89	58.33	69.44	77.78	63.89	55.56
hc	50.00	55.56	36.11	86.11	58.33	66.67	55.56	72.22	52.78	63.89
fulldata	69.44	69.44	58.33	80.56	63.89	63.89	69.44	77.78	80.56	55.56

```
kable(lms.comp$acctable,    caption = "Accuracy - LM (Corr >= 0)")
```

Accuracy - LM (Corr >= 0)

	s.bio	s.cd	s.cs	s.energy	s.fin	s.hc	s.ind	s.mat	s.tech	s.util
kd	58.33	66.67	61.11	83.33	58.33	58.33	69.44	77.78	77.78	58.33
hd	58.33	75.00	61.11	83.33	58.33	58.33	69.44	83.33	77.78	58.33
kc	52.78	55.56	44.44	86.11	61.11	66.67	66.67	77.78	61.11	52.78
hc	50.00	61.11	47.22	86.11	52.78	66.67	75.00	77.78	55.56	61.11
fulldata	69.44	69.44	58.33	80.56	63.89	63.89	69.44	77.78	80.56	55.56

- Following the results of the backtest, we have made a cumulative return matrix and out put it with kable() and the confusion matrices for the correponding model types uesd. A few important conclusions here are that up movements make uo ~ 60% of price movements. The majority of the models tested showed (based on their confusion table) that they had troube predicting these moves. The worst affected were the Decision Tree models. The average error rates accross the models and clusters was 31% to 39%. This is important becuase as well look at the models accuracies vs. their returns we can see that accuracy and returns are obviously correlated but the confusion matrices reveal how this is poition dependant. Therefore, these market risks and model risks need to be addressed to improve the models. In conclusion, however, our results show that over a 36 month period, we were able have all of our models outperform the market. The cluserting analysis showed promise as a data filtering procedure and Decision Tree models also show potential to help advance this strategy into a more dynamic asset allocation tool.