Principal Component Analysis and Self Organized Maps

Hassan Mir

2024-06-15

In this part, I employ Principal Component Analysis (PCA) and Self-Organized Map (SOM) algorithm to gain insights about 20 companies based on their expsoures to Fama French 3-factors, their alphas, volatility, adjusted R-squared, Sharpe Ratio, and Information Ratio.

Installing relevant packages.

```
warning = FALSE
library('kohonen') #SOM algorithm
```

```
## Warning: package 'kohonen' was built under R version 4.3.3
```

```
library('gmm') #data set 'Finance'
```

```
## Warning: package 'gmm' was built under R version 4.3.3
```

```
## Loading required package: sandwich
```

```
library('data.table') #data manipulation
```

```
## Warning: package 'data.table' was built under R version 4.3.2
```

```
library('ggplot2') #visualization
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library('frenchdata') #optional library
```

```
## Warning: package 'frenchdata' was built under R version 4.3.3
```

Creating a function that estimates model and some additional coefficients.

```
mylm<-function(y,x1,x2,x3)
{
  fit<-lm(y~x1+x2+x3,na.action='na.omit')
  IVOLhat<-summary(fit)$sigma
  adjR2<-summary(fit)$adj.r.squared
  m<-mean(y,na.rm=TRUE)
  s<-sd(y,na.rm=TRUE)
  SRhat<-m/s
  IRhat<-coef(fit)[1]/IVOLhat

  coefficients<-c(coef(fit),IVOLhat,adjR2,SRhat,IRhat)
  names(coefficients)<-c('alphahat','betahat','shat','hhat',
                         'sigmahat','adjR2','SRhat','IRhat')

  return(coefficients)
}
```

Using the Finance dataset to extract relevant variables including risk free rate, market return, and fama french factors. Then taking the differences and creating a dataframe for the resulting values.

```
data(Finance)
rstock<-Finance[,1:(ncol(Finance)-4)]
rf<-Finance[,'rf']
rm<-Finance[,'rm']
rmexcess<-rm-rf
smb<-Finance[,'smb']
hml<-Finance[,'hml']

rstockexcess<-sweep(x=rstock,MARGIN=1,STATS=rf,FUN="-")

EST<-apply(X=rstockexcess,MARGIN=2,FUN="mylm",x1=rmexcess,x2=smb,x3=hml)
EST<-t(EST)
X<-data.frame(EST)
X
```

```
##          alphahat   betahat        shat         hhat sigmahat      adjR2
## WMK  -0.014210460 0.5840843  0.36911382  0.511735796 1.358008 0.18523640
## UIS  -0.102665430 1.3988750  0.53399668  0.145826813 3.318372 0.18928825
## ORB  -0.021097135 1.0938957  0.72947598  0.007998101 3.454441 0.12450783
## MAT  -0.011163040 0.7526733  0.24607336  0.181042201 2.145538 0.13388211
## ABAX -0.024328233 0.8789724  1.04726575  0.588227848 4.468547 0.05252895
## T    -0.013939992 0.8260146 -0.53430829  0.302761692 1.576504 0.29505140
## EMR   0.004530843 0.9821885 -0.23923261  0.329103711 1.334034 0.41823207
## JCS  -0.007303786 0.3986744  0.17779106 -0.146393605 2.927464 0.02844975
## VOXX -0.089042429 1.5221496  1.68674327  0.455771776 3.647425 0.20574968
## ZOOM -0.116170587 0.6869271  0.63936729 -0.432720695 5.830268 0.02703197
## ROG   0.007446174 1.0472297  1.00092477  0.473072594 2.024605 0.26698611
## GGG   0.030862329 0.8654158  0.58254971  0.429910587 1.656164 0.25437910
## PC   -0.021085353 0.7644425  0.19917982  0.181481864 1.738315 0.19561619
## GCO  -0.025764752 1.1857563  0.97370253  0.532418981 3.343363 0.14077712
## EBF  -0.020935622 0.8211295  0.68184392  0.773311728 1.898858 0.19403897
## F    -0.073920555 1.2835707 -0.15107610  0.954673073 2.218184 0.29803587
## FNM  -0.122733423 1.2229485 -0.34080598  0.969911593 4.760113 0.08058095
## NHP   0.011309042 0.9495598  0.69759993  0.917312172 1.489851 0.33972671
## AA   -0.041765915 1.3217150 -0.09407038  0.755323149 1.932294 0.36847230
## TDW  -0.015788479 1.0459957  0.32450483  0.721491343 2.376836 0.18721783
##              SRhat        IRhat
## WMK   0.0033174561 -0.010464195
## UIS  -0.0209572487 -0.030938489
## ORB  -0.0007116718 -0.006107250
## MAT   0.0018833769 -0.005202910
## ABAX  0.0004499701 -0.005444328
## T     0.0019147315 -0.008842343
## EMR   0.0145806160  0.003396348
## JCS  -0.0011732275 -0.002494919
## VOXX -0.0131834043 -0.024412411
## ZOOM -0.0190067214 -0.019925429
## ROG   0.0144853833  0.003677841
## GGG   0.0277944915  0.018634827
## PC   -0.0028225406 -0.012129765
## GCO   0.0011786500 -0.007706239
## EBF   0.0034695652 -0.011025374
```

```
## F    -0.0138021589 -0.033324807
## FNM  -0.0173938483 -0.025783719
## NHP   0.0241465678  0.007590720
## AA   -0.0030334421 -0.021614678
## TDW   0.0054656144 -0.006642646
```

Annualizing and converting to unit-less value for better interpretability.

```
sweep(x=X[,c('alphahat','sigmahat','SRhat','IRhat')],
      STATS=c(252*0.01,sqrt(252)*0.01,sqrt(252),sqrt(252)),MARGIN=2,FUN='*')
```

```
##         alphahat  sigmahat        SRhat       IRhat
## WMK  -0.03581036 0.2155771  0.052662982 -0.16611394
## UIS  -0.25871688 0.5267753 -0.332686009 -0.49113329
## ORB  -0.05316478 0.5483755 -0.011297439 -0.09694959
## MAT  -0.02813086 0.3405936  0.029897681 -0.08259363
## ABAX -0.06130715 0.7093598  0.007143054 -0.08642603
## T    -0.03512878 0.2502623  0.030395420 -0.14036784
## EMR   0.01141773 0.2117713  0.231460103  0.05391535
## JCS  -0.01840554 0.4647206 -0.018624409 -0.03960561
## VOXX -0.22438692 0.5790107 -0.209280055 -0.38753502
## ZOOM -0.29274988 0.9255263 -0.301722348 -0.31630639
## ROG   0.01876436 0.3213960  0.229948331  0.05838392
## GGG   0.07777307 0.2629079  0.441223873  0.29581871
## PC   -0.05313509 0.2759490 -0.044806443 -0.19255405
## GCO  -0.06492717 0.5307424  0.018710489 -0.12233275
## EBF  -0.05275777 0.3014344  0.055077640 -0.17502238
## F    -0.18627980 0.3521258 -0.219102481 -0.52901491
## FNM  -0.30928823 0.7556446 -0.276118781 -0.40930385
## NHP   0.02849879 0.2365065  0.383314880  0.12049894
## AA   -0.10525011 0.3067422 -0.048154400 -0.34312238
## TDW  -0.03978697 0.3773110  0.086763939 -0.10544874
```

```
pca_results <- prcomp(X, scale = TRUE)
print(summary(pca_results))
```

Principal Component Analysis

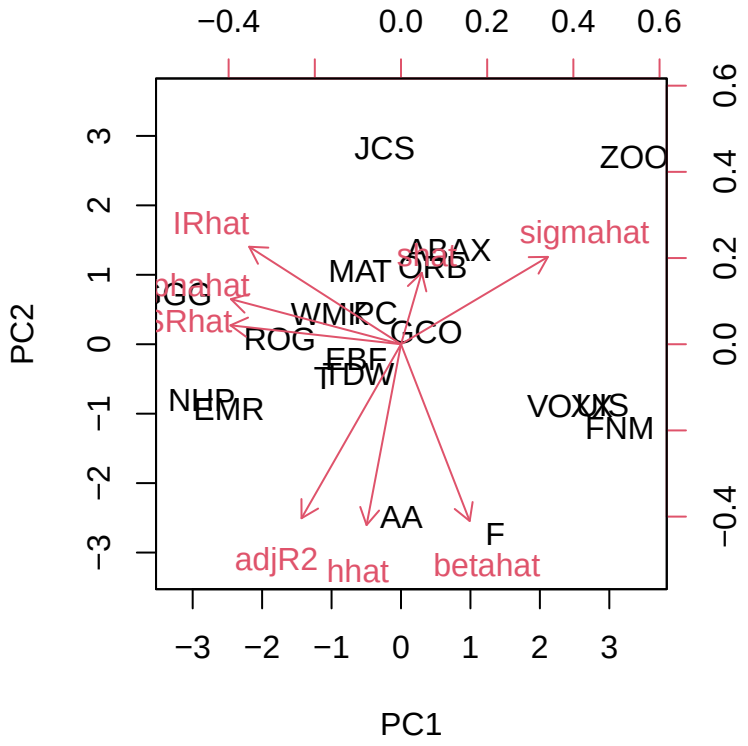
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5    PC6     PC7
## Standard deviation     1.9355 1.4323 1.0979 0.74263 0.54412 0.3382 0.18026
## Proportion of Variance 0.4683 0.2564 0.1507 0.06894 0.03701 0.0143 0.00406
## Cumulative Proportion  0.4683 0.7247 0.8754 0.94432 0.98133 0.9956 0.99969
##                           PC8
## Standard deviation     0.04952
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

```
pca_loadings <- pca_results$rotation
round(pca_loadings,4)
```

```
##              PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
## alphahat -0.4933  0.1309 -0.0605  0.0077  0.1261  0.5043  0.6823 -0.0090
## betahat   0.1993 -0.5127 -0.4059 -0.2668 -0.3573  0.5585 -0.1365  0.0571
## shat      0.0602  0.2074 -0.8248 -0.1763  0.4316 -0.2277  0.0203 -0.0591
## hhat     -0.0999 -0.5245 -0.2021  0.7888  0.1214 -0.1090  0.0577 -0.1482
```

```
## sigmahat  0.4261  0.2533 -0.1927  0.2096 -0.5953 -0.2215  0.5146  0.0701
## adjR2    -0.2885 -0.5047  0.0480 -0.4665 -0.1390 -0.5425  0.3223 -0.1569
## SRhat    -0.4946  0.0549 -0.1948  0.0984 -0.2628 -0.1633 -0.2073  0.7523
## IRhat    -0.4404  0.2828 -0.1819  0.0656 -0.4612  0.0060 -0.3167 -0.6129
```

```
biplot(pca_results, scale=0)
```

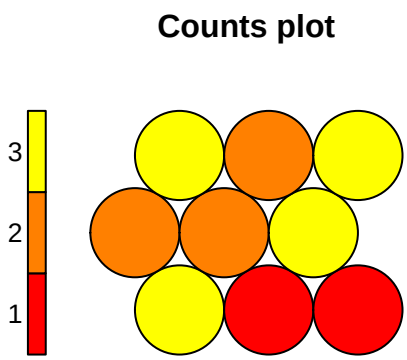


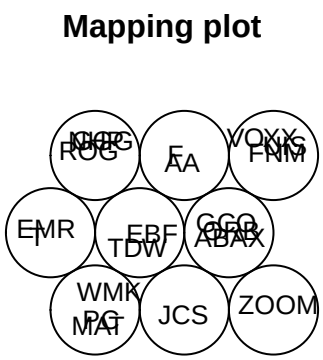
Self Organizing Maps (SOM) Using unsupervised SOMs to visually inspect common variables.

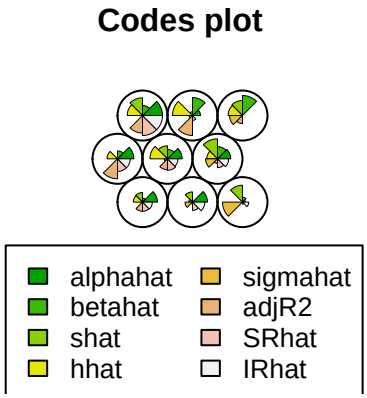
```
xdim<-ydim<-3
Xst<-scale(as.matrix(X))#convert X to a matrix and standardize the columns of X

mysom <- som(X=Xst, grid = somgrid(xdim=xdim, ydim=ydim, 'hexagonal'))

par(mfrow=c(2,2),mar=c(2, 4, 1, 1),cex=0.8)
plot(mysom,type='counts')
plot(mysom,type='mapping',labels=rownames(X))
plot(mysom,type='codes')
```







Extracting relevant nodes and centroids for each stock.

```
info<-data.frame(ticker=rownames(X),node=mysom$unit.classif)
info
```

```
##    ticker node
## 1     WMK    1
## 2     UIS    9
## 3     ORB    6
## 4     MAT    1
## 5    ABAX    6
## 6       T    4
## 7     EMR    4
## 8     JCS    2
## 9    VOXX    9
## 10   ZOOM    3
## 11    ROG    7
## 12    GGG    7
## 13     PC    1
## 14    GCO    6
## 15    EBF    5
## 16      F    8
## 17    FNM    9
## 18    NHP    7
## 19     AA    8
## 20    TDW    5
```

```r
table(x=info$node)
```

```
## x
## 1 2 3 4 5 6 7 8 9
## 3 1 1 2 2 3 3 2 3
```

```r
mysom$codes
```

```
## [[1]]
##      alphahat     betahat        shat       hhat   sigmahat       adjR2
## V1  0.4008721 -0.93740870 -0.29388775 -0.4148888 -0.7126300 -0.24526069
## V2  0.4182431 -1.49883676 -0.22410674 -1.1642135  0.1915622 -1.20599538
## V3 -1.2350603 -0.80464620  0.50327122 -1.6800166  2.0331034 -1.41857769
## V4  0.6056278 -0.35732579 -1.33635166 -0.2396122 -0.9106607  1.14331559
## V5  0.3704263 -0.15921165  0.08145582  0.7427249 -0.4389867 -0.02068377
## V6  0.1833441  0.24525350  0.84435446 -0.2047074  0.8458508 -0.82891788
## V7  1.1039710 -0.05986914  0.59320021  0.4271646 -0.7434531  0.82381744
## V8 -0.5356395  1.07781845 -0.92531996  1.0780626 -0.4299136  1.13437401
## V9 -1.6297899  1.37504768  0.30330658  0.3022795  0.9918710 -0.38332335
##          SRhat       IRhat
## V1  0.01558959  0.04909615
## V2 -0.11561168  0.38276032
## V3 -1.02534237 -0.42826551
## V4  0.50391569  0.44454336
## V5  0.34044435  0.13491506
## V6 -0.03663810  0.23296916
## V7  1.57159730  1.43573896
## V8 -0.61029619 -1.22964871
## V9 -1.31260862 -1.28750184
```

```r
df.centroids<-data.frame(mysom$codes)
df.centroids$node<-paste0('Node',1:nrow(df.centroids))
df.centroids$node<-factor(df.centroids$node,labels=paste0('Node',1:nrow(df.centroids)))
dt<-data.table(df.centroids)
dt
```

```
##      alphahat     betahat        shat       hhat   sigmahat       adjR2
## 1:  0.4008721 -0.93740870 -0.29388775 -0.4148888 -0.7126300 -0.24526069
## 2:  0.4182431 -1.49883676 -0.22410674 -1.1642135  0.1915622 -1.20599538
## 3: -1.2350603 -0.80464620  0.50327122 -1.6800166  2.0331034 -1.41857769
## 4:  0.6056278 -0.35732579 -1.33635166 -0.2396122 -0.9106607  1.14331559
## 5:  0.3704263 -0.15921165  0.08145582  0.7427249 -0.4389867 -0.02068377
## 6:  0.1833441  0.24525350  0.84435446 -0.2047074  0.8458508 -0.82891788
## 7:  1.1039710 -0.05986914  0.59320021  0.4271646 -0.7434531  0.82381744
## 8: -0.5356395  1.07781845 -0.92531996  1.0780626 -0.4299136  1.13437401
## 9: -1.6297899  1.37504768  0.30330658  0.3022795  0.9918710 -0.38332335
##          SRhat       IRhat  node
## 1:  0.01558959  0.04909615 Node1
## 2: -0.11561168  0.38276032 Node2
## 3: -1.02534237 -0.42826551 Node3
## 4:  0.50391569  0.44454336 Node4
## 5:  0.34044435  0.13491506 Node5
## 6: -0.03663810  0.23296916 Node6
## 7:  1.57159730  1.43573896 Node7
## 8: -0.61029619 -1.22964871 Node8
```

```
## 9: -1.31260862 -1.28750184 Node9
```

Melting data from wide to long.

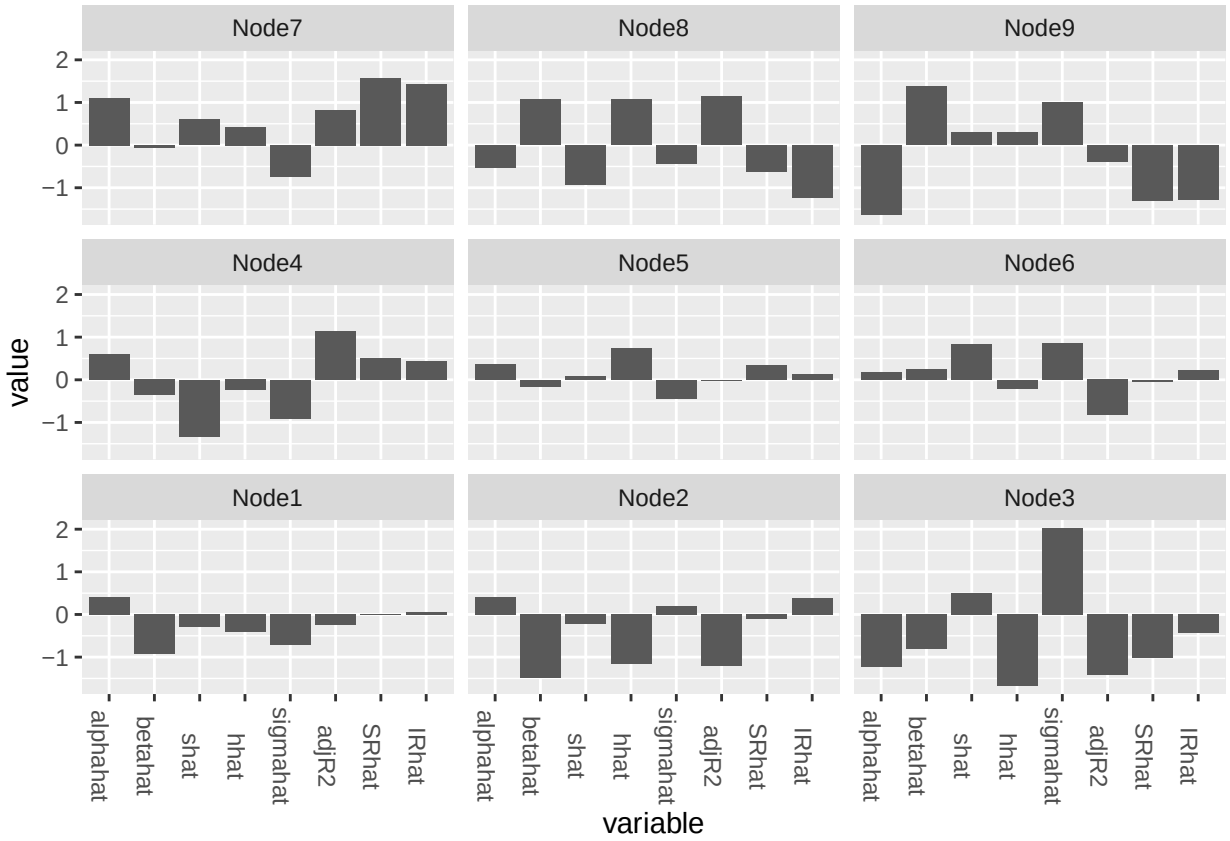
```
dt.molten<-melt.data.table(data=dt, id.vars='node')
dt.molten
```

```
##      node variable       value
##  1: Node1 alphahat  0.40087212
##  2: Node2 alphahat  0.41824308
##  3: Node3 alphahat -1.23506032
##  4: Node4 alphahat  0.60562778
##  5: Node5 alphahat  0.37042630
##  6: Node6 alphahat  0.18334414
##  7: Node7 alphahat  1.10397100
##  8: Node8 alphahat -0.53563945
##  9: Node9 alphahat -1.62978990
## 10: Node1  betahat -0.93740870
## 11: Node2  betahat -1.49883676
## 12: Node3  betahat -0.80464620
## 13: Node4  betahat -0.35732579
## 14: Node5  betahat -0.15921165
## 15: Node6  betahat  0.24525350
## 16: Node7  betahat -0.05986914
## 17: Node8  betahat  1.07781845
## 18: Node9  betahat  1.37504768
## 19: Node1     shat -0.29388775
## 20: Node2     shat -0.22410674
## 21: Node3     shat  0.50327122
## 22: Node4     shat -1.33635166
## 23: Node5     shat  0.08145582
## 24: Node6     shat  0.84435446
## 25: Node7     shat  0.59320021
## 26: Node8     shat -0.92531996
## 27: Node9     shat  0.30330658
## 28: Node1     hhat -0.41488883
## 29: Node2     hhat -1.16421347
## 30: Node3     hhat -1.68001664
## 31: Node4     hhat -0.23961215
## 32: Node5     hhat  0.74272489
## 33: Node6     hhat -0.20470742
## 34: Node7     hhat  0.42716465
## 35: Node8     hhat  1.07806264
## 36: Node9     hhat  0.30227949
## 37: Node1 sigmahat -0.71263001
## 38: Node2 sigmahat  0.19156219
## 39: Node3 sigmahat  2.03310341
## 40: Node4 sigmahat -0.91066071
## 41: Node5 sigmahat -0.43898672
## 42: Node6 sigmahat  0.84585080
## 43: Node7 sigmahat -0.74345311
## 44: Node8 sigmahat -0.42991365
## 45: Node9 sigmahat  0.99187104
## 46: Node1    adjR2 -0.24526069
## 47: Node2    adjR2 -1.20599538
```

```
## 48: Node3    adjR2 -1.41857769
## 49: Node4    adjR2  1.14331559
## 50: Node5    adjR2 -0.02068377
## 51: Node6    adjR2 -0.82891788
## 52: Node7    adjR2  0.82381744
## 53: Node8    adjR2  1.13437401
## 54: Node9    adjR2 -0.38332335
## 55: Node1    SRhat  0.01558959
## 56: Node2    SRhat -0.11561168
## 57: Node3    SRhat -1.02534237
## 58: Node4    SRhat  0.50391569
## 59: Node5    SRhat  0.34044435
## 60: Node6    SRhat -0.03663810
## 61: Node7    SRhat  1.57159730
## 62: Node8    SRhat -0.61029619
## 63: Node9    SRhat -1.31260862
## 64: Node1    IRhat  0.04909615
## 65: Node2    IRhat  0.38276032
## 66: Node3    IRhat -0.42826551
## 67: Node4    IRhat  0.44454336
## 68: Node5    IRhat  0.13491506
## 69: Node6    IRhat  0.23296916
## 70: Node7    IRhat  1.43573896
## 71: Node8    IRhat -1.22964871
## 72: Node9    IRhat -1.28750184
##      node variable       value
```

Plotting the centroids on a bar graph.

```
g<-ggplot(data=dt.molten, aes(x=variable,y=value))
g<-g+stat_summary(fun='mean', geom='bar')+facet_wrap(~node,nrow=xdim,as.table=FALSE)
g<-g+theme(axis.text.x=element_text(angle=-90))
print(g)
```


Node7
Node8
Node9
Node4
Node5
Node6
Node1
Node2
Node3
value
2
1
0
−1
alphahat
betahat
shat
hhat
sigmahat
adjR2
SRhat
IRhat
variable