Stock Data Assignment

Soumya Mukherjee

22 May 2020

Installing/loading required packages

```
# Packages used randtests, tactile, mvShapiroTest, factoextra, psych
# Data is stored in the file "~//Stock Data Assignment//stocks.txt" relative to the present
# working directory (can be obtained using getwd() command in R console)
# So if getwd() gives output "C:/Users/Soumya/Documents"
# Then full file path of the data is "C://Users//Soumya//Documents//Stock Data Assignment//stocks.txt"
# Please change the file location as required

# Please remove the Hash sign from the code lines with install.packages
# if the packages are not installed

#install.packages("randtests")
#install.packages("mvShapiroTest")
#install.packages("mvnormtest")
#install.packages("factoextra")
#install.packages("psych")
library(randtests)
library(tactile)
```

```
## Loading required package: lattice
```

```
library(mvShapiroTest)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

Importing the data and performing preliminary analysis

```
# There is one line of data for each week and the
# weekly gains are represented as
# x1 = ALLIED CHEMICAL
```

```r
# x2 = DUPONT
# x3 = UNION CARBIDE
# x4 = EXXON
# x5 = TEXACO

stocks = read.delim("Stock Data Assignment//stocks.txt",header=F)
colnames(stocks)=c("Allied Chemical","Du Pont","Union Carbide","Exxon","Texaco")

# The 5 variables
x1=stocks[,1]
x2=stocks[,2]
x3=stocks[,3]
x4=stocks[,4]
x5=stocks[,5]

# A glimpse of few rows of the data
stocks[c(1:4,100:103),]
```

```
##     Allied Chemical    Du Pont Union Carbide      Exxon     Texaco
## 1         0.0130338 -0.0078431    -0.0031889 -0.0447693  0.0052151
## 2         0.0084862  0.0166886    -0.0062100  0.0119560  0.0134890
## 3        -0.0179153 -0.0086393     0.0100360  0.0000000 -0.0061428
## 4         0.0215589 -0.0034858     0.0174353 -0.0285917 -0.0069534
## 100       0.0033626  0.0029016    -0.0030507 -0.0012193 -0.0097005
## 101       0.0170147  0.0095061     0.0181994 -0.0161758 -0.0075614
## 102       0.0103929 -0.0026612     0.0044290 -0.0024818 -0.0164502
## 103      -0.0127948 -0.0143678    -0.0187402 -0.0049759 -0.0163732
```

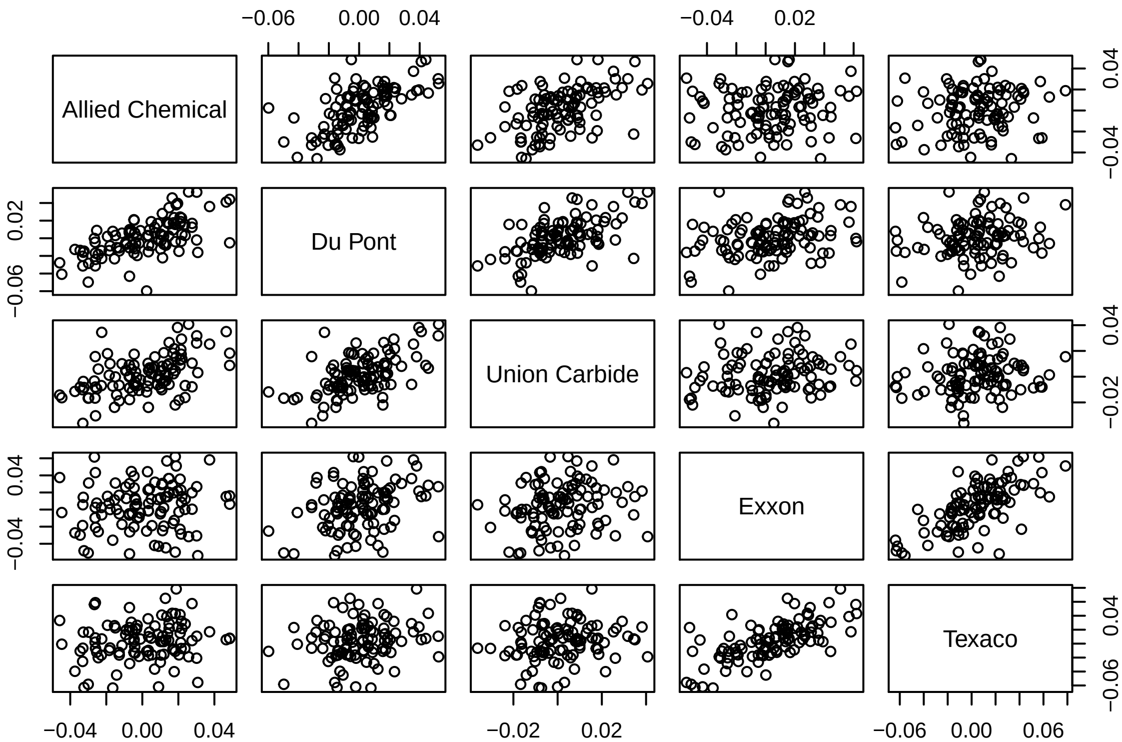
```r
# Computation of the sample mean, covariance and correaltion matrices
mean=apply(stocks,2,mean)
S=var(stocks)
zapsmall(S)
```

```
##                 Allied Chemical      Du Pont Union Carbide        Exxon
## Allied Chemical    0.0004332695 0.0002756679  0.0001590265 0.0000641193
## Du Pont            0.0002756679 0.0004387172  0.0001799737 0.0001814512
## Union Carbide      0.0001590265 0.0001799737  0.0002239722 0.0000734135
## Exxon              0.0000641193 0.0001814512  0.0000734135 0.0007224964
## Texaco             0.0000889662 0.0001232623  0.0000605461 0.0005082772
##                       Texaco
## Allied Chemical 0.0000889662
## Du Pont         0.0001232623
## Union Carbide   0.0000605461
## Exxon           0.0005082772
## Texaco          0.0007656742
```

```r
R=cor(stocks)
R
```

```
##                 Allied Chemical   Du Pont Union Carbide     Exxon    Texaco
## Allied Chemical       1.0000000 0.6322878     0.5104973 0.1146019 0.1544628
## Du Pont               0.6322878 1.0000000     0.5741424 0.3222921 0.2126747
## Union Carbide         0.5104973 0.5741424     1.0000000 0.1824992 0.1462067
## Exxon                 0.1146019 0.3222921     0.1824992 1.0000000 0.6833777
## Texaco                0.1544628 0.2126747     0.1462067 0.6833777 1.0000000
```

```r
# Drawing the pairwise scatterplots
pairs(stocks)
```



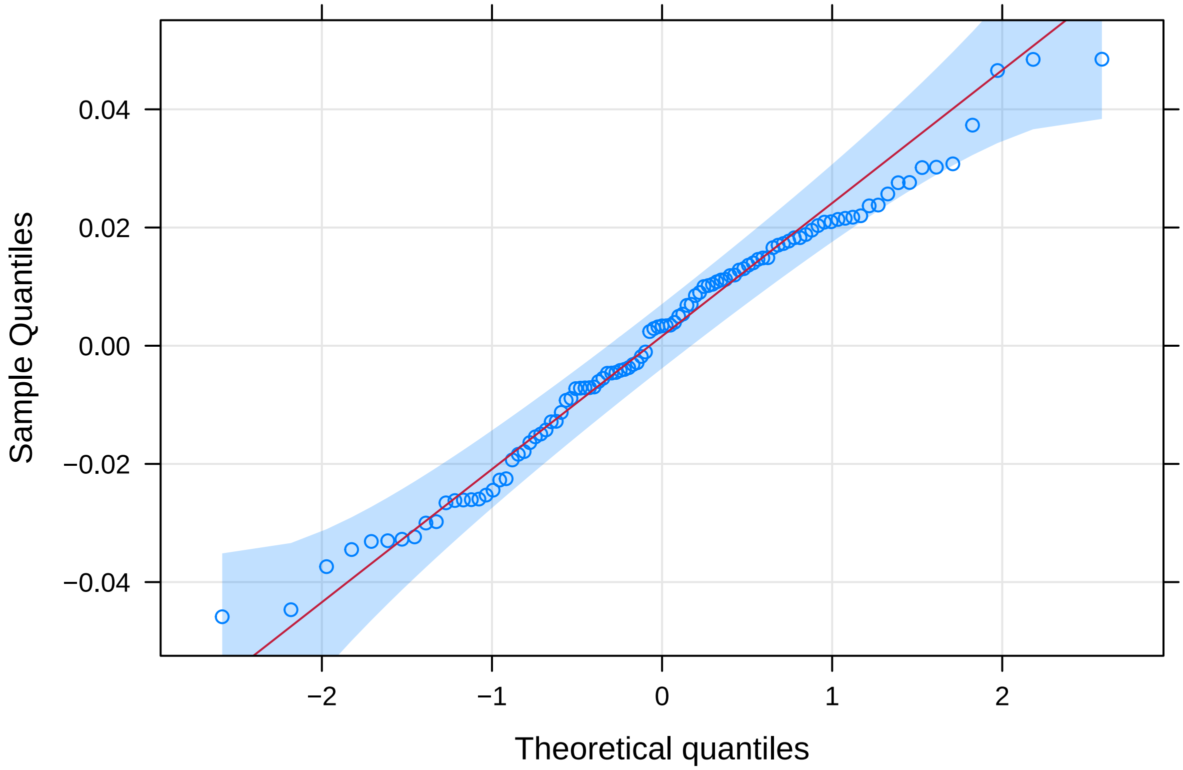

```r
# Performing the Wald-Wolfowitz runs test on each variable
apply(stocks,2,runs.test)
```

```
## $`Allied Chemical`
##
##  Runs Test
##
## data:  newX[, i]
## statistic = -1.5921, runs = 44, n1 = 51, n2 = 51, n = 102, p-value =
## 0.1114
## alternative hypothesis: nonrandomness
##
##
## $`Du Pont`
##
##  Runs Test
##
## data:  newX[, i]
## statistic = -0.59705, runs = 49, n1 = 51, n2 = 51, n = 102, p-value =
## 0.5505
## alternative hypothesis: nonrandomness
##
##
## $`Union Carbide`
##
##  Runs Test
##
## data:  newX[, i]
```

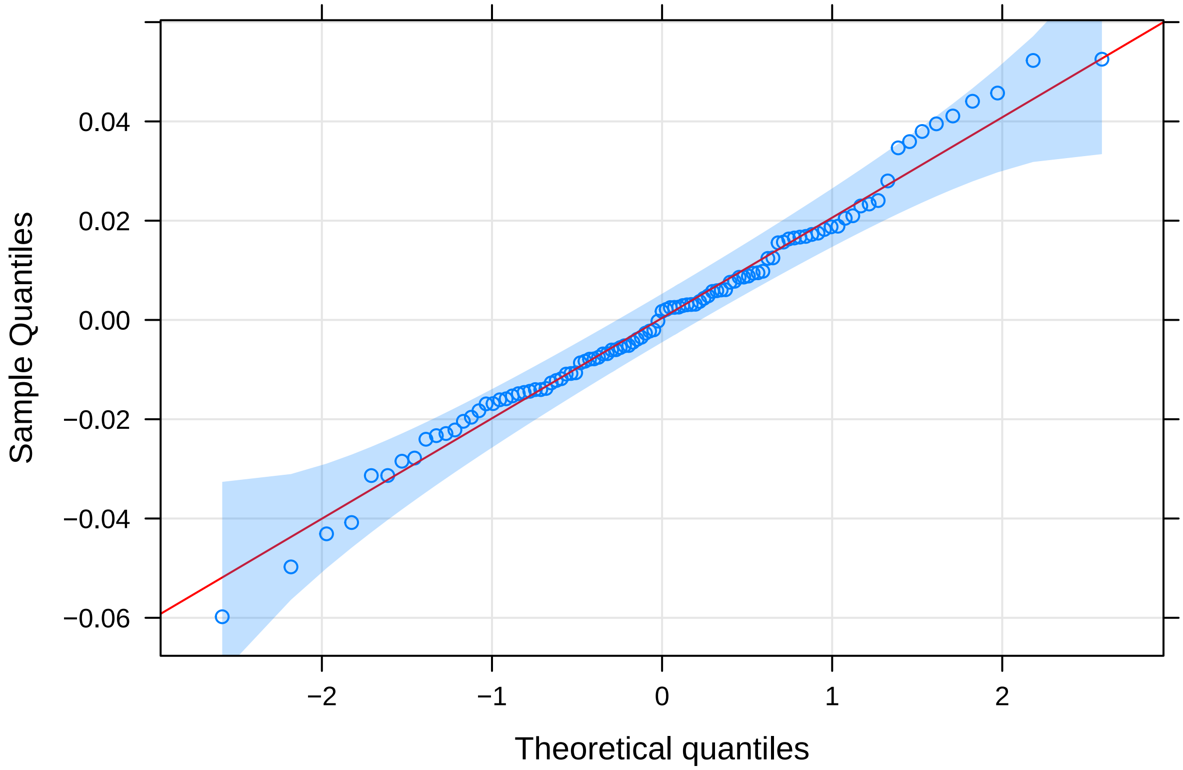
```
## statistic = 1.3931, runs = 59, n1 = 51, n2 = 51, n = 102, p-value =
## 0.1636
## alternative hypothesis: nonrandomness
##
##
## $Exxon
##
##  Runs Test
##
## data:  newX[, i]
## statistic = 1.3931, runs = 59, n1 = 51, n2 = 51, n = 102, p-value =
## 0.1636
## alternative hypothesis: nonrandomness
##
##
## $Texaco
##
##  Runs Test
##
## data:  newX[, i]
## statistic = 0.79607, runs = 56, n1 = 51, n2 = 51, n = 102, p-value =
## 0.426
## alternative hypothesis: nonrandomness
```

Checking multivariate normality of the data

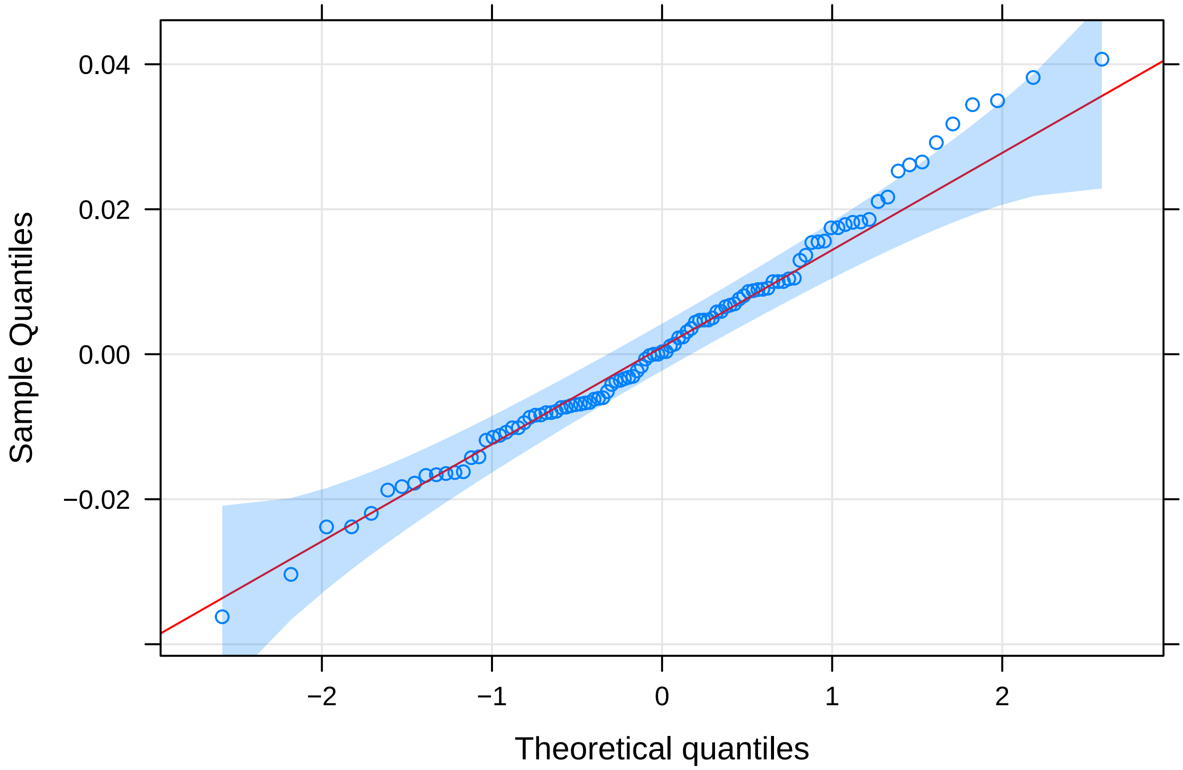
```r
## QQ Plots
qqmath(x1, distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(x1, grid = TRUE)
  panel.qqmathline(x1, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```



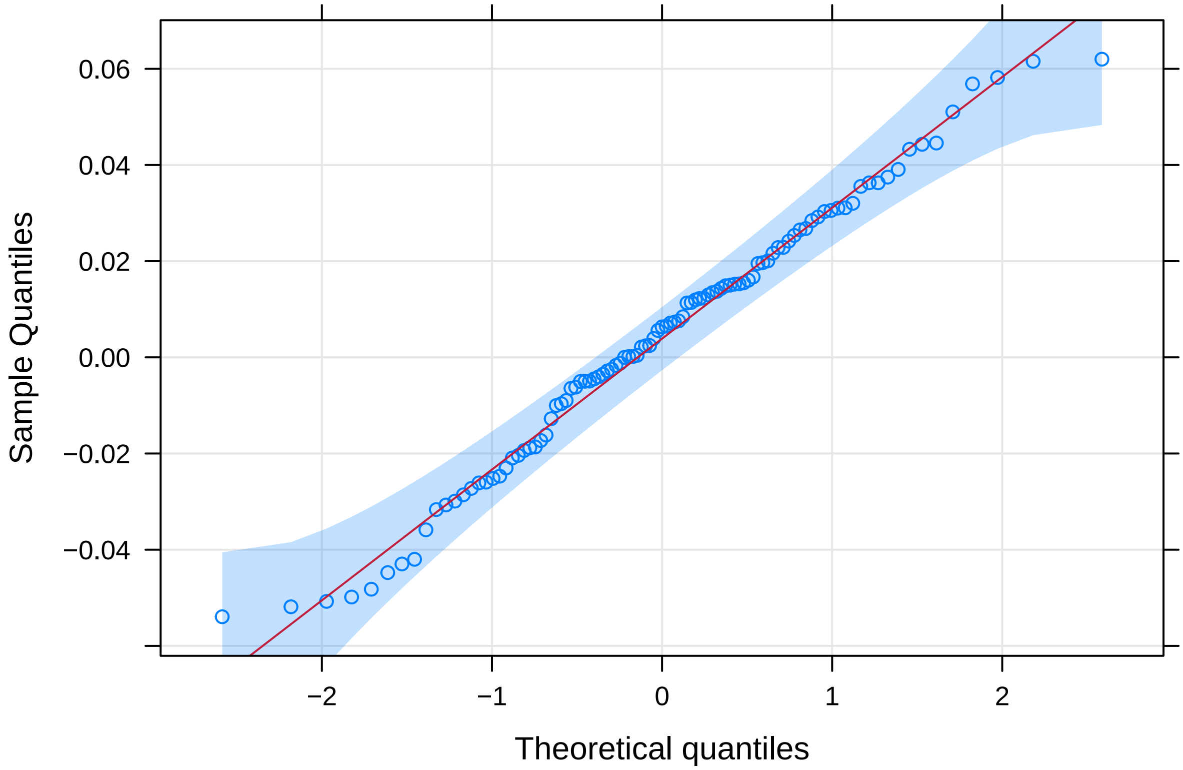

```r
qqmath(x2, distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(x2, grid = TRUE)
  panel.qqmathline(x2, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```



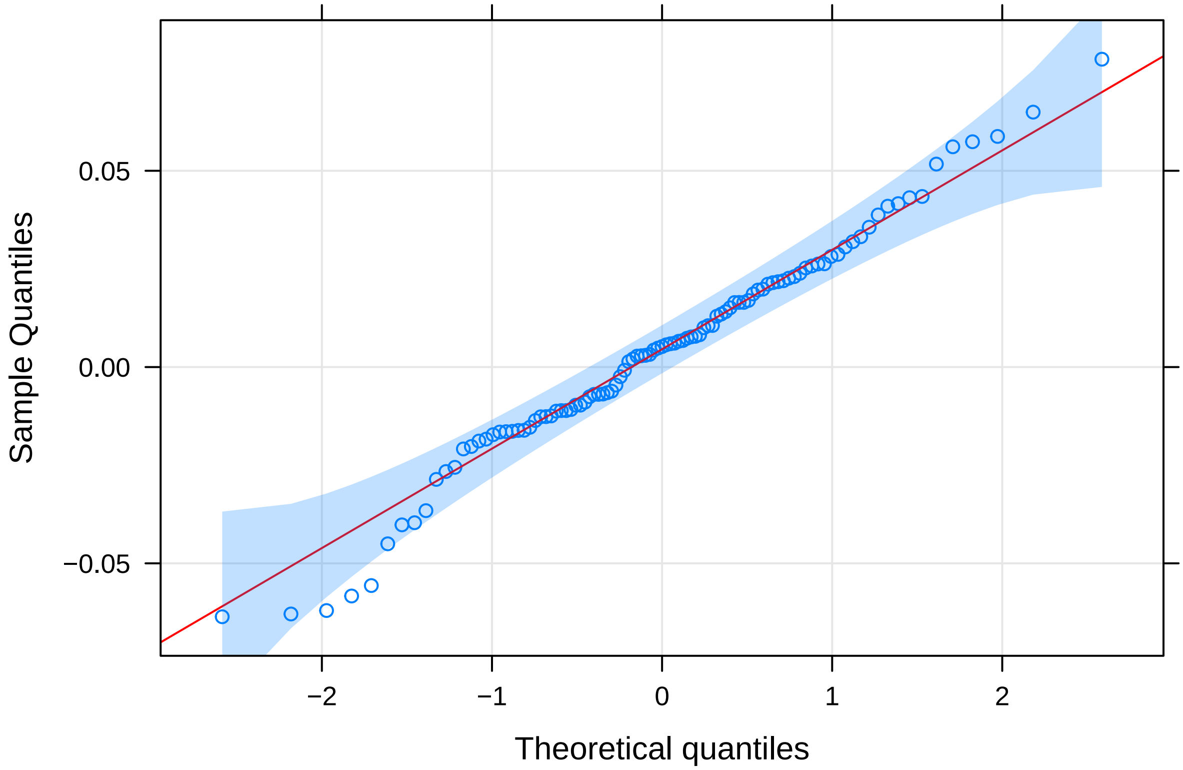

```
qqmath(x3, distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(x3, grid = TRUE)
  panel.qqmathline(x3, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```




```
qqmath(x4, distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(x4, grid = TRUE)
  panel.qqmathline(x4, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```




```r
qqmath(x5, distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(x5, grid = TRUE)
  panel.qqmathline(x5, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```




```r
#Shapiro-Wilks Test of Univariate Normality
shapiro.test(x1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x1
## W = 0.98333, p-value = 0.2222
```

```r
shapiro.test(x2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x2
## W = 0.98597, p-value = 0.3513
```

```r
shapiro.test(x3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x3
## W = 0.98631, p-value = 0.372
```

```r
shapiro.test(x4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x4
## W = 0.98675, p-value = 0.3994
```

```r
shapiro.test(x5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x5
## W = 0.98425, p-value = 0.2614
```

```r
#Test for Multivariate normality

#Multivariate Shapiro-Wilks Test for normality
mvShapiro.Test(as.matrix(stocks))
```

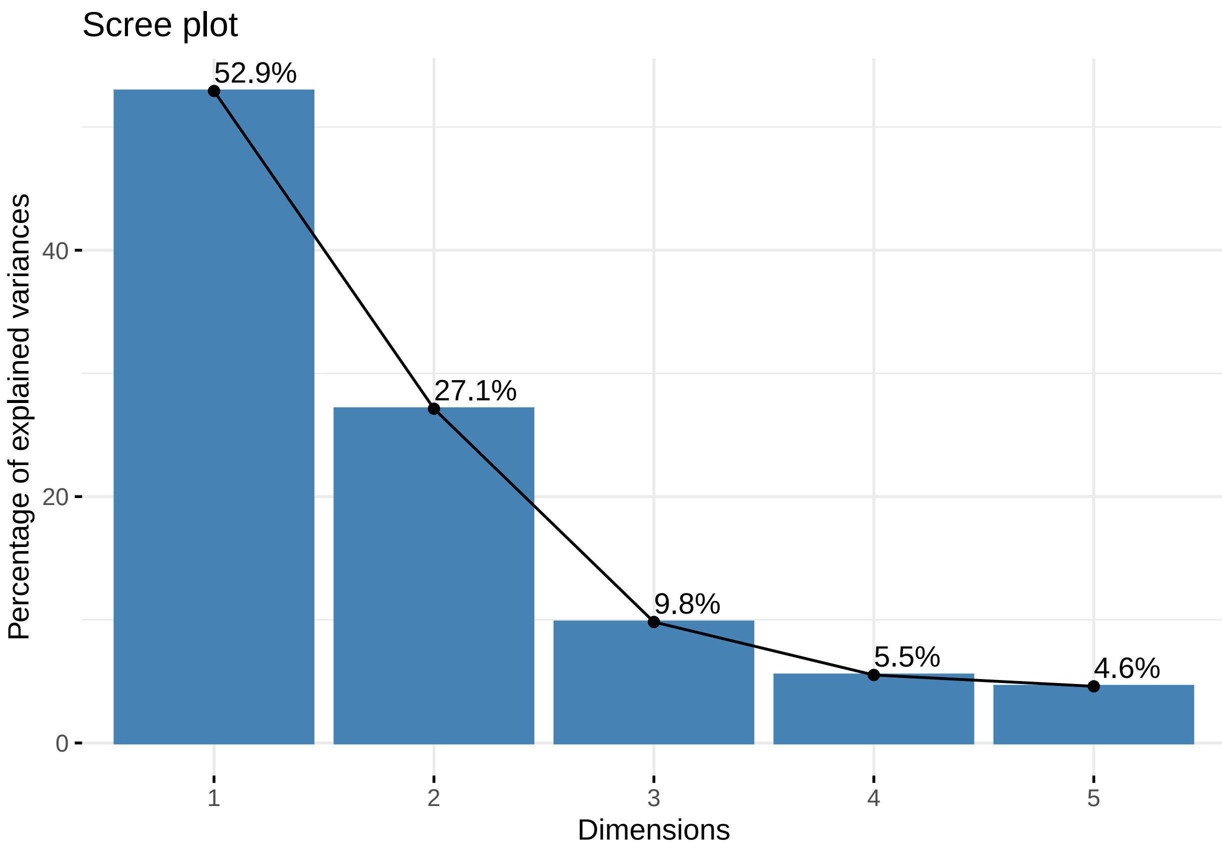
```
##
##  Generalized Shapiro-Wilk test for Multivariate Normality by
##  Villasenor-Alva and Gonzalez-Estrada
##
## data:  as.matrix(stocks)
## MVW = 0.98734, p-value = 0.5194
```

Principal Component method along with estimate of no. of common factors 'm'

```r
#Covariance matrix

#Principal Component Analysis
stocks_var_pca=prcomp(stocks,scale.=FALSE)

#Scree Plot
fviz_eig(stocks_var_pca, addlabels = TRUE)
```




```
#Variances of Principal components i.e. Eigenvalues
eigenvalues_stocks_var_pca=(stocks_var_pca$sdev)^2
eigenvalues_stocks_var_pca
```

```
## [1] 0.0013676780 0.0007011596 0.0002538024 0.0001426026 0.0001188868
```

```
#Eigenvectors used in determining the sample principal components
eigenvectors_stocks_var_pca=stocks_var_pca$rotation
eigenvectors_stocks_var_pca
```

```
##                       PC1        PC2         PC3        PC4         PC5
## Allied Chemical -0.2228228  0.6252260 -0.32611218  0.6627590 -0.11765952
## Du Pont         -0.3072900  0.5703900  0.24959014 -0.4140935  0.58860803
## Union Carbide   -0.1548103  0.3445049  0.03763929 -0.4970499 -0.78030428
## Exxon           -0.6389680 -0.2479475  0.64249741  0.3088689 -0.14845546
## Texaco          -0.6509044 -0.3218478 -0.64586064 -0.2163758  0.09371777
```

```
#Proportion and cumulative proportion of variation in the data explained by
#each principal component
prop_var_pca=eigenvalues_stocks_var_pca/sum(diag(S))
names(prop_var_pca)=c("1st PC","2nd PC","3rd PC","4th PC","5th PC")
cum_prop_var_pca=cumsum(prop_var_pca)
summary_var_pca=cbind("Eigenvalue"=eigenvalues_stocks_var_pca,"Proportion of variance"=prop_var_pca,"Cu
summary_var_pca
```

```
##          Eigenvalue Proportion of variance Cumulative Proportion of variance
## 1st PC 0.0013676780             0.52926066                         0.5292607
## 2nd PC 0.0007011596             0.27133298                         0.8005936
## 3rd PC 0.0002538024             0.09821584                         0.8988095
```

```
## 4th PC 0.0001426026             0.05518400                          0.9539935
## 5th PC 0.0001188868             0.04600652                          1.0000000
```

```r
#Number of common factors
mvar=2

#Estimate of L, Psi and communality
L_var_pca = eigenvectors_stocks_var_pca[,1:mvar] %*% diag(sqrt(eigenvalues_stocks_var_pca)[1:mvar])
Psi_var_pca=diag(diag(S-L_var_pca %*% t(L_var_pca)))
communality_var_pca = apply(L_var_pca,1,function(x)sum(x^2))
L_var_pca
```

```
##                         [,1]         [,2]
## Allied Chemical -0.008240463  0.016555621
## Du Pont         -0.011364238  0.015103596
## Union Carbide   -0.005725215  0.009122290
## Exxon           -0.023630398 -0.006565506
## Texaco          -0.024071832 -0.008522342
```

```r
zapsmall(Psi_var_pca)
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 9.127563e-05 0.000000e+00 0.0000000000 0.0000000000 0.0000000000
## [2,] 0.000000e+00 8.145269e-05 0.0000000000 0.0000000000 0.0000000000
## [3,] 0.000000e+00 0.000000e+00 0.0001079779 0.0000000000 0.0000000000
## [4,] 0.000000e+00 0.000000e+00 0.0000000000 0.0001209948 0.0000000000
## [5,] 0.000000e+00 0.000000e+00 0.0000000000 0.0000000000 0.0001135907
```

```r
communality_var_pca
```

```
## Allied Chemical         Du Pont   Union Carbide           Exxon          Texaco
##    0.0003419938    0.0003572645    0.0001159943    0.0006015016    0.0006520834
```

```r
#Residual matrix
res_var_pca=S-Psi_var_pca-L_var_pca %*% t(L_var_pca)
res_var_pca
```

```
##                 Allied Chemical       Du Pont Union Carbide         Exxon
## Allied Chemical    0.000000e+00 -6.802809e-05 -3.917707e-05 -2.191010e-05
## Du Pont           -6.802809e-05  0.000000e+00 -2.286839e-05  1.207249e-05
## Union Carbide     -3.917707e-05 -2.286839e-05  0.000000e+00 -1.983177e-06
## Exxon             -2.191010e-05  1.207249e-05 -1.983177e-06  0.000000e+00
## Texaco             3.169578e-05 -2.157775e-05  4.729779e-07 -1.165033e-04
##                       Texaco
## Allied Chemical  3.169578e-05
## Du Pont         -2.157775e-05
## Union Carbide    4.729779e-07
## Exxon           -1.165033e-04
## Texaco           0.000000e+00
```

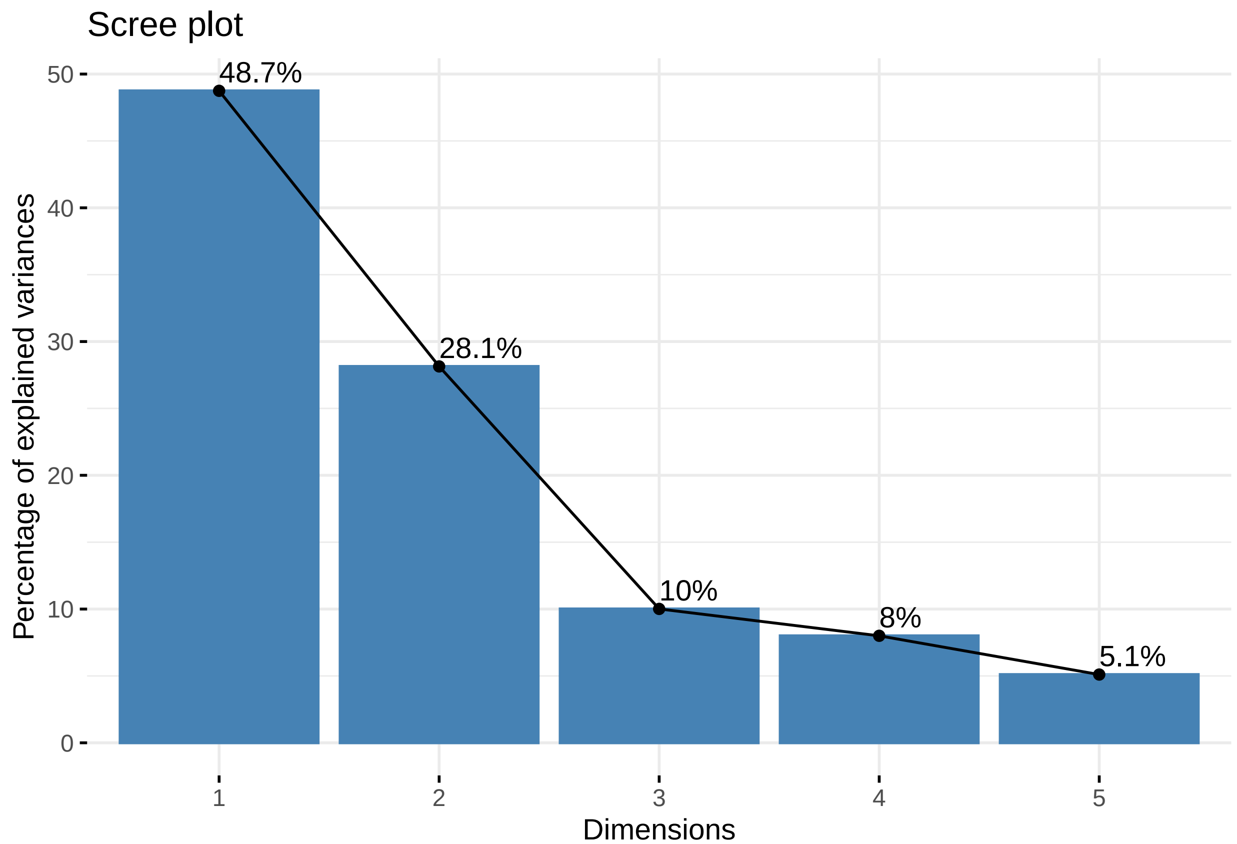
```r
sum(res_var_pca^2)
```

```
## [1] 4.471763e-08
```

```r
#Correlation matrix

#Principal Component Analysis
stocks_cor_pca=prcomp(stocks,scale.=TRUE)
```

```r
#Scree Plot
fviz_eig(stocks_cor_pca, addlabels = TRUE)
```




```r
#Variances of Principal components i.e. Eigenvalues
eigenvalues_stocks_cor_pca=(stocks_cor_pca$sdev)^2
eigenvalues_stocks_cor_pca
```

```
## [1] 2.4372731 1.4070127 0.5005127 0.4000316 0.2551699
```

```r
#Eigenvectors used in determining the sample principal components
eigenvectors_stocks_cor_pca=stocks_cor_pca$rotation
eigenvectors_stocks_cor_pca
```

```
##                        PC1        PC2         PC3        PC4         PC5
## Allied Chemical -0.4690832  0.3680070 -0.60431522  0.3630228  0.38412160
## Du Pont         -0.5324055  0.2364624 -0.13610618 -0.6292079 -0.49618794
## Union Carbide   -0.4651633  0.3151795  0.77182810  0.2889658  0.07116948
## Exxon           -0.3873459 -0.5850373  0.09336192 -0.3812515  0.59466408
## Texaco          -0.3606821 -0.6058463 -0.10882629  0.4934145 -0.49755167
```

```r
#Proportion and cumulative proportion of variation in the data explained by
#each principal component
prop_cor_pca=eigenvalues_stocks_cor_pca/5
names(prop_cor_pca)=c("1st PC","2nd PC","3rd PC","4th PC","5th PC")
cum_prop_cor_pca=cumsum(prop_cor_pca)
summary_cor_pca=cbind("Eigenvalue"=eigenvalues_stocks_cor_pca,"Proportion of variance"=prop_cor_pca,"Cu
summary_cor_pca
```

```
##        Eigenvalue Proportion of variance Cumulative Proportion of variance
## 1st PC  2.4372731             0.48745462                         0.4874546
## 2nd PC  1.4070127             0.28140253                         0.7688572
## 3rd PC  0.5005127             0.10010255                         0.8689597
## 4th PC  0.4000316             0.08000632                         0.9489660
## 5th PC  0.2551699             0.05103398                         1.0000000
```

```
#Number of common factors
mcor=3

#Estimate of L, Psi and communality
L_cor_pca = eigenvectors_stocks_cor_pca[,1:mcor] %*% diag(sqrt(eigenvalues_stocks_cor_pca)[1:mcor])
Psi_cor_pca=diag(diag(R-L_cor_pca %*% t(L_cor_pca)))
communality_cor_pca = apply(L_cor_pca,1,function(x)sum(x^2))
L_cor_pca
```

```
##                        [,1]       [,2]        [,3]
## Allied Chemical -0.7323218  0.4365209 -0.42753444
## Du Pont         -0.8311791  0.2804859 -0.09629094
## Union Carbide   -0.7262022  0.3738582  0.54604465
## Exxon           -0.6047155 -0.6939569  0.06605069
## Texaco          -0.5630885 -0.7186401 -0.07699125
```

```
zapsmall(Psi_cor_pca)
```

```
##            [,1]     [,2]      [,3]      [,4]      [,5]
## [1,] 0.09036854 0.000000 0.0000000 0.0000000 0.0000000
## [2,] 0.00000000 0.221197 0.0000000 0.0000000 0.0000000
## [3,] 0.00000000 0.000000 0.0346956 0.0000000 0.0000000
## [4,] 0.00000000 0.000000 0.0000000 0.1483802 0.0000000
## [5,] 0.00000000 0.000000 0.0000000 0.0000000 0.1605601
```

```
communality_cor_pca
```

```
## Allied Chemical         Du Pont   Union Carbide           Exxon          Texaco
##       0.9096315       0.7788030       0.9653044       0.8516198       0.8394399
```

```
#Residual matrix
res_cor_pca=R-Psi_cor_pca-L_cor_pca %*% t(L_cor_pca)
res_cor_pca
```

```
##                 Allied Chemical     Du Pont Union Carbide        Exxon
## Allied Chemical     0.000000000 -0.14000842    0.04893955  0.002921187
## Du Pont            -0.140008423  0.00000000   -0.08174450  0.020670425
## Union Carbide       0.048939551 -0.08174450    0.00000000 -0.033271662
## Exxon               0.002921187  0.02067042   -0.03327166  0.000000000
## Texaco              0.022885789 -0.06119782    0.04800079 -0.150750629
##                      Texaco
## Allied Chemical  0.02288579
## Du Pont         -0.06119782
## Union Carbide    0.04800079
## Exxon           -0.15075063
## Texaco           0.00000000
```

```
sum(res_cor_pca^2)
```

```
## [1] 0.1190423
```

Iterative Principal Components Method

```r
#Covariance matrix
iter_var = fa(stocks,nfactors = 2,rotate = 'none',fm='pa',covar = TRUE)

# Estimate of L, Psi and communality
summary_var_iter <- data.frame(dimnames(S)[[1]],iter_var$loadings[,], iter_var$communality, iter_var$uni
colnames(summary_var_iter)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uniquen
rownames(summary_var_iter)=NULL
print(summary_var_iter)
```

```
##          Variable Factor 1 loadings Factor 2 loadings  Communality   Uniqueness
## 1 Allied Chemical       0.008812028       0.012278722 0.0002284188 0.0002048506
## 2          Du Pont       0.012049919       0.011312633 0.0002731762 0.0001655410
## 3   Union Carbide       0.006536749       0.007934450 0.0001056846 0.0001182876
## 4            Exxon       0.019842768      -0.007255823 0.0004463824 0.0002761140
## 5           Texaco       0.019231993      -0.007924641 0.0004326695 0.0003330047
```

```r
L_var_iter=as.matrix(summary_var_iter[,c(2,3)])
communality_var_iter=as.vector(summary_var_iter[,4])
Psi_var_iter=diag(communality_var_iter)
L_var_iter
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]       0.008812028       0.012278722
## [2,]       0.012049919       0.011312633
## [3,]       0.006536749       0.007934450
## [4,]       0.019842768      -0.007255823
## [5,]       0.019231993      -0.007924641
```

```r
zapsmall(Psi_var_iter)
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 0.0002284188 0.0000000000 0.0000000000 0.0000000000 0.0000000000
## [2,] 0.0000000000 0.0002731762 0.0000000000 0.0000000000 0.0000000000
## [3,] 0.0000000000 0.0000000000 0.0001056846 0.0000000000 0.0000000000
## [4,] 0.0000000000 0.0000000000 0.0000000000 0.0004463824 0.0000000000
## [5,] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0004326695
```

```r
communality_var_iter
```

```
## [1] 0.0002284188 0.0002731762 0.0001056846 0.0004463824 0.0004326695
```

```r
#Residual matrix
res_var_iter=S-Psi_var_iter-L_var_iter %*% t(L_var_iter)
res_var_iter
```

```
##                 Allied Chemical       Du Pont Union Carbide         Exxon
## Allied Chemical   -2.356822e-05  3.057901e-05  3.999612e-06 -2.164349e-05
## Du Pont            3.057901e-05 -1.076352e-04  1.144688e-05  2.442992e-05
## Union Carbide      3.999612e-06  1.144688e-05  1.260304e-05  1.277254e-06
## Exxon             -2.164349e-05  2.442992e-05  1.277254e-06 -1.702684e-04
## Texaco             1.679777e-05 -1.883313e-05 -2.290924e-06  6.916144e-05
##                        Texaco
## Allied Chemical  1.679777e-05
## Du Pont         -1.883313e-05
## Union Carbide   -2.290924e-06
```

```
## Exxon             6.916144e-05
## Texaco           -9.966479e-05
```

```r
sum(res_var_iter^2)
```

```
## [1] 6.637283e-08
```

```r
#Correlation matrix
iter_cor = fa(stocks,nfactors = 2,rotate = 'none',fm='pa')
```

```
## maximum iteration exceeded
```

```r
# Estimate of L, Psi and communality
summary_cor_iter <- data.frame(dimnames(R)[[1]],iter_cor$loadings[,], iter_cor$communality, iter_cor$uni
colnames(summary_cor_iter)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uniquen
rownames(summary_cor_iter)=NULL
print(summary_cor_iter)
```

```
##          Variable Factor 1 loadings Factor 2 loadings Communality  Uniqueness
## 1 Allied Chemical         0.6254584        -0.4293152   0.5755098 0.424490154
## 2         Du Pont         0.7766148        -0.3417133   0.7198985 0.280101509
## 3   Union Carbide         0.5909665        -0.3320045   0.4594683 0.540531664
## 4           Exxon         0.7035483         0.7087661   0.9973296 0.002670401
## 5          Texaco         0.5087849         0.4549090   0.4658042 0.534195813
```

```r
L_cor_iter=as.matrix(summary_cor_iter[,c(2,3)])
communality_cor_iter=as.vector(summary_cor_iter[,4])
Psi_cor_iter=diag(as.vector(summary_cor_iter[,5]))
L_cor_iter
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]         0.6254584        -0.4293152
## [2,]         0.7766148        -0.3417133
## [3,]         0.5909665        -0.3320045
## [4,]         0.7035483         0.7087661
## [5,]         0.5087849         0.4549090
```

```r
zapsmall(Psi_cor_iter)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.4244902 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.2801015 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.5405317 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.0026704 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.5341958
```

```r
communality_cor_iter
```

```
## [1] 0.5755098 0.7198985 0.4594683 0.9973296 0.4658042
```

```r
#Residual matrix
res_cor_iter=R-Psi_cor_iter-L_cor_iter %*% t(L_cor_iter)
res_cor_iter
```

```
##                 Allied Chemical       Du Pont Union Carbide        Exxon
## Allied Chemical    0.0000000000 -0.0001551642 -1.662220e-03 -0.021154225
## Du Pont           -0.0001551642  0.0000000000  1.738787e-03  0.018100937
## Union Carbide     -0.0016622198  0.0017387867 -5.551115e-17  0.002039266
## Exxon             -0.0211542254  0.0181009368  2.039266e-03  0.000000000
```

```
## Texaco             0.0315383223 -0.0270066728 -3.436274e-03  0.002998902
##                         Texaco
## Allied Chemical  0.031538322
## Du Pont         -0.027006673
## Union Carbide   -0.003436274
## Exxon            0.002998902
## Texaco           0.000000000
```

```
sum(res_cor_iter^2)
```

```
## [1] 0.005059883
```

Maximum likelihood method

```
#Covariance matrix
ml_var = fa(stocks,nfactors = 2,rotate = 'none',fm='ml',covar = TRUE)

# Estimate of L, Psi and communality
summary_var_ml <- data.frame(dimnames(S)[[1]],ml_var$loadings[,], ml_var$communality, ml_var$uniquenesse
colnames(summary_var_ml)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uniquene
rownames(summary_var_ml)=NULL
print(summary_var_ml)
```

```
##          Variable Factor 1 loadings Factor 2 loadings Communality   Uniqueness
## 1 Allied Chemical             1e-15             1e-15       2e-30 0.0004332695
## 2         Du Pont             1e-15             1e-15       2e-30 0.0004387172
## 3   Union Carbide             1e-15             1e-15       2e-30 0.0002239722
## 4           Exxon             1e-15             1e-15       2e-30 0.0007224964
## 5          Texaco             1e-15             1e-15       2e-30 0.0007656742
```

```
L_var_ml=as.matrix(summary_var_ml[,c(2,3)])
communality_var_ml=as.vector(summary_var_ml[,4])
Psi_var_ml=diag(communality_var_ml)
L_var_ml
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]             1e-15             1e-15
## [2,]             1e-15             1e-15
## [3,]             1e-15             1e-15
## [4,]             1e-15             1e-15
## [5,]             1e-15             1e-15
```

```
zapsmall(Psi_var_ml)
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] 2e-30 0e+00 0e+00 0e+00 0e+00
## [2,] 0e+00 2e-30 0e+00 0e+00 0e+00
## [3,] 0e+00 0e+00 2e-30 0e+00 0e+00
## [4,] 0e+00 0e+00 0e+00 2e-30 0e+00
## [5,] 0e+00 0e+00 0e+00 0e+00 2e-30
```

```
communality_var_ml
```

```
## [1] 2e-30 2e-30 2e-30 2e-30 2e-30
```

```r
#Residual matrix
res_var_ml=S-Psi_var_ml-L_var_ml %*% t(L_var_ml)
res_var_ml
```

```
##                  Allied Chemical      Du Pont Union Carbide        Exxon
## Allied Chemical     4.332695e-04 0.0002756679  1.590265e-04 6.411929e-05
## Du Pont             2.756679e-04 0.0004387172  1.799737e-04 1.814512e-04
## Union Carbide       1.590265e-04 0.0001799737  2.239722e-04 7.341348e-05
## Exxon               6.411929e-05 0.0001814512  7.341348e-05 7.224964e-04
## Texaco              8.896616e-05 0.0001232623  6.054612e-05 5.082772e-04
##                       Texaco
## Allied Chemical 8.896616e-05
## Du Pont         1.232623e-04
## Union Carbide   6.054612e-05
## Exxon           5.082772e-04
## Texaco          7.656742e-04
```

```r
sum(res_var_ml^2)
```

```
## [1] 2.461053e-06
```

```r
#Correlation matrix
ml_cor = fa(stocks,nfactors = 2,rotate = 'none',fm='ml')

# Estimate of L, Psi and communality
summary_cor_ml <- data.frame(dimnames(R)[[1]],ml_cor$loadings[,], ml_cor$communality, ml_cor$uniquenesse
colnames(summary_cor_ml)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uniquenes
rownames(summary_cor_ml)=NULL
print(summary_cor_ml)
```

```
##          Variable Factor 1 loadings Factor 2 loadings Communality Uniqueness
## 1 Allied Chemical         0.1205972       0.754267066   0.5834625 0.41653750
## 2         Du Pont         0.3284924       0.785749656   0.7253098 0.27469024
## 3   Union Carbide         0.1876017       0.650216951   0.4579765 0.54202352
## 4           Exxon         0.9974724      -0.007103504   0.9950016 0.00499844
## 5          Texaco         0.6851746       0.026317440   0.4701568 0.52984315
```

```r
L_cor_ml=as.matrix(summary_cor_ml[,c(2,3)])
communality_cor_ml=as.vector(summary_cor_ml[,4])
Psi_cor_ml=diag(as.vector(summary_cor_ml[,5]))
L_cor_ml
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]         0.1205972       0.754267066
## [2,]         0.3284924       0.785749656
## [3,]         0.1876017       0.650216951
## [4,]         0.9974724      -0.007103504
## [5,]         0.6851746       0.026317440
```

```r
zapsmall(Psi_cor_ml)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.4165375 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.2746902 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.5420235 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.0049984 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.5298432
```

```
communality_cor_ml
```

```
## [1] 0.5834625 0.7253098 0.4579765 0.9950016 0.4701568
```

```
#Residual matrix
res_cor_ml=R-Psi_cor_ml-L_cor_ml %*% t(L_cor_ml)
res_cor_ml
```

```
##                  Allied Chemical       Du Pont Union Carbide         Exxon
## Allied Chemical     0.000000e+00  7.481355e-06 -2.564147e-03 -3.325559e-04
## Du Pont             7.481355e-06  0.000000e+00  1.608946e-03  2.116216e-04
## Union Carbide      -2.564147e-03  1.608946e-03  0.000000e+00 -9.518880e-06
## Exxon              -3.325559e-04  2.116216e-04 -9.518880e-06  0.000000e+00
## Texaco              5.198222e-02 -3.307885e-02  5.547216e-04  1.218872e-04
##                        Texaco
## Allied Chemical  0.0519822233
## Du Pont         -0.0330788458
## Union Carbide    0.0005547216
## Exxon            0.0001218872
## Texaco           0.0000000000
```

```
sum(res_cor_ml^2)
```

```
## [1] 0.007612006
```

```
# Bartlett's large sample test of goodness of fit

bartlett.gof.test = function(data,R,m,L,Psi)
{
  n=nrow(data)
  p=ncol(data)
  n_prime=n-1-(2*p+4*m+5)/6
  bartlett.statistic=n_prime*log(det(L%*% t(L)+Psi)/det(R))
  df=((p-m)^2-(p+m))/2
  crit_val=qchisq(0.95,df)
  pval=1-pchisq(bartlett.statistic,df)

  if(bartlett.statistic>crit_val)
  {
    print("Bartlett's large sample test of goodness of fit")
    print(paste("Value of the test statistic is ",bartlett.statistic))
    print(paste("The degree of freedom is ",df))
    print(paste("The p-value is ",pval))
    print(paste("The critical value is",crit_val))
    print("The null hypothesis is rejected at 5% level of significance")
  }
  else
  {
    print("Bartlett's large sample test of goodness of fit")
    print(paste("Value of the test statistic is ",bartlett.statistic))
    print(paste("The degree of freedom is ",df))
    print(paste("The p-value is ",pval))
    print(paste("The critical value is",crit_val))
    print("We fail to reject the null hypothesis at 5% level of significance")
  }
}
```

```r
bartlett.gof.test(stocks,R,m=2,L_cor_ml,Psi_cor_ml)
```

```
## [1] "Bartlett's large sample test of goodness of fit"
## [1] "Value of the test statistic is  2.00446333744723"
## [1] "The degree of freedom is  1"
## [1] "The p-value is  0.156836790051934"
## [1] "The critical value is 3.84145882069412"
## [1] "We fail to reject the null hypothesis at 5% level of significance"
```

Application of Varimax Rotation

```r
# Iterative PC method

#Covariance matrix
iter_var_rot = fa(stocks,nfactors = 2,rotate = 'varimax',fm='pa',covar = TRUE)

# Estimate of L, Psi and communality
summary_var_iter_rot <- data.frame(dimnames(S)[[1]],iter_var_rot$loadings[,], iter_var_rot$communality,
colnames(summary_var_iter_rot)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uni
rownames(summary_var_iter_rot)=NULL
print(summary_var_iter_rot)
```

```
##          Variable Factor 1 loadings Factor 2 loadings  Communality   Uniqueness
## 1 Allied Chemical       0.001429288       0.015045796 0.0002284188 0.0002048506
## 2         Du Pont       0.004713135       0.015841798 0.0002731762 0.0001655410
## 3   Union Carbide       0.001651426       0.010146791 0.0001056846 0.0001182876
## 4           Exxon       0.020796887       0.003724498 0.0004463824 0.0002761140
## 5          Texaco       0.020606044       0.002839093 0.0004326695 0.0003330047
```

```r
L_var_iter_rot=as.matrix(summary_var_iter_rot[,c(2,3)])
communality_var_iter_rot=as.vector(summary_var_iter_rot[,4])
Psi_var_iter_rot=diag(communality_var_iter_rot)
L_var_iter_rot
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]       0.001429288       0.015045796
## [2,]       0.004713135       0.015841798
## [3,]       0.001651426       0.010146791
## [4,]       0.020796887       0.003724498
## [5,]       0.020606044       0.002839093
```

```r
zapsmall(Psi_var_iter_rot)
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 0.0002284188 0.0000000000 0.0000000000 0.0000000000 0.0000000000
## [2,] 0.0000000000 0.0002731762 0.0000000000 0.0000000000 0.0000000000
## [3,] 0.0000000000 0.0000000000 0.0001056846 0.0000000000 0.0000000000
## [4,] 0.0000000000 0.0000000000 0.0000000000 0.0004463824 0.0000000000
## [5,] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0004326695
```

```r
communality_var_iter_rot
```

```
## [1] 0.0002284188 0.0002731762 0.0001056846 0.0004463824 0.0004326695
```

```r
#Residual matrix
res_var_iter_rot=S-Psi_var_iter_rot-L_var_iter_rot %*% t(L_var_iter_rot)
res_var_iter_rot
```

```
##                  Allied Chemical       Du Pont Union Carbide         Exxon
## Allied Chemical    -2.356822e-05  3.057901e-05  3.999612e-06 -2.164349e-05
## Du Pont             3.057901e-05 -1.076352e-04  1.144688e-05  2.442992e-05
## Union Carbide       3.999612e-06  1.144688e-05  1.260304e-05  1.277254e-06
## Exxon              -2.164349e-05  2.442992e-05  1.277254e-06 -1.702684e-04
## Texaco              1.679777e-05 -1.883313e-05 -2.290924e-06  6.916144e-05
##                         Texaco
## Allied Chemical  1.679777e-05
## Du Pont         -1.883313e-05
## Union Carbide   -2.290924e-06
## Exxon            6.916144e-05
## Texaco          -9.966479e-05
```

```r
sum(res_var_iter_rot^2)
```

```
## [1] 6.637283e-08
```

```r
#Correlation matrix
iter_cor_rot = fa(stocks,nfactors = 2,rotate = 'varimax',fm='pa')
```

```
## maximum iteration exceeded
```

```r
# Estimate of L, Psi and communality
summary_cor_iter_rot <- data.frame(dimnames(R)[[1]],iter_cor_rot$loadings[,], iter_cor_rot$communality,
colnames(summary_cor_iter_rot)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Un
rownames(summary_cor_iter_rot)=NULL
print(summary_cor_iter_rot)
```

```
##          Variable Factor 1 loadings Factor 2 loadings Communality  Uniqueness
## 1 Allied Chemical         0.7567758        0.05291676   0.5755098 0.424490154
## 2         Du Pont         0.8206362        0.21553342   0.7198985 0.280101509
## 3   Union Carbide         0.6692376        0.10765378   0.4594683 0.540531664
## 4           Exxon         0.1099817        0.99258935   0.9973296 0.002670401
## 5          Texaco         0.1153959        0.67267227   0.4658042 0.534195813
```

```r
L_cor_iter_rot=as.matrix(summary_cor_iter_rot[,c(2,3)])
communality_cor_iter_rot=as.vector(summary_cor_iter_rot[,4])
Psi_cor_iter_rot=diag(as.vector(summary_cor_iter_rot[,5]))
L_cor_iter_rot
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]         0.7567758        0.05291676
## [2,]         0.8206362        0.21553342
## [3,]         0.6692376        0.10765378
## [4,]         0.1099817        0.99258935
## [5,]         0.1153959        0.67267227
```

```r
zapsmall(Psi_cor_iter_rot)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.4244902 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.2801015 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.5405317 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.0026704 0.0000000
```

```
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.5341958
```

```
communality_cor_iter_rot
```

```
## [1] 0.5755098 0.7198985 0.4594683 0.9973296 0.4658042
```

```
# Maximum Likelihood method

#Covariance matrix
ml_var_rot = fa(stocks,nfactors = 2,rotate = 'varimax',fm='ml',covar = TRUE)

# Estimate of L, Psi and communality
summary_var_ml_rot <- data.frame(dimnames(S)[[1]],ml_var_rot$loadings[,], ml_var_rot$communality, ml_va
colnames(summary_var_ml_rot)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uniq
rownames(summary_var_ml_rot)=NULL
print(summary_var_ml_rot)
```

```
##          Variable Factor 1 loadings Factor 2 loadings Communality   Uniqueness
## 1 Allied Chemical             1e-15             1e-15       2e-30 0.0004332695
## 2          Du Pont            1e-15             1e-15       2e-30 0.0004387172
## 3   Union Carbide             1e-15             1e-15       2e-30 0.0002239722
## 4            Exxon            1e-15             1e-15       2e-30 0.0007224964
## 5           Texaco            1e-15             1e-15       2e-30 0.0007656742
```

```
L_var_ml_rot=as.matrix(summary_var_ml_rot[,c(2,3)])
communality_var_ml_rot=as.vector(summary_var_ml_rot[,4])
Psi_var_ml_rot=diag(communality_var_ml_rot)
L_var_ml_rot
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]             1e-15             1e-15
## [2,]             1e-15             1e-15
## [3,]             1e-15             1e-15
## [4,]             1e-15             1e-15
## [5,]             1e-15             1e-15
```

```
zapsmall(Psi_var_ml_rot)
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] 2e-30 0e+00 0e+00 0e+00 0e+00
## [2,] 0e+00 2e-30 0e+00 0e+00 0e+00
## [3,] 0e+00 0e+00 2e-30 0e+00 0e+00
## [4,] 0e+00 0e+00 0e+00 2e-30 0e+00
## [5,] 0e+00 0e+00 0e+00 0e+00 2e-30
```

```
communality_var_ml_rot
```

```
## [1] 2e-30 2e-30 2e-30 2e-30 2e-30
```

```
#Correlation matrix
ml_cor_rot = fa(stocks,nfactors = 2,rotate = 'varimax',fm='ml')

# Estimate of L, Psi and communality
summary_cor_ml_rot <- data.frame(dimnames(R)[[1]],ml_cor_rot$loadings[,], ml_cor_rot$communality, ml_co
colnames(summary_cor_ml_rot)=c("Variable","Factor 1 loadings","Factor 2 loadings", "Communality", "Uniq
rownames(summary_cor_ml_rot)=NULL
print(summary_cor_ml_rot)
```

```
##          Variable Factor 1 loadings Factor 2 loadings Communality Uniqueness
```

```
## 1 Allied Chemical          0.7632891         0.02919252   0.5834625 0.41653750
## 2          Du Pont          0.8194973         0.23180592   0.7253098 0.27469024
## 3   Union Carbide          0.6680336         0.10820147   0.4579765 0.54202352
## 4            Exxon          0.1126721         0.99111380   0.9950016 0.00499844
## 5           Texaco          0.1083670         0.67706236   0.4701568 0.52984315
```

```
L_cor_ml_rot=as.matrix(summary_cor_ml_rot[,c(2,3)])
communality_cor_ml_rot=as.vector(summary_cor_ml_rot[,4])
Psi_cor_ml_rot=diag(as.vector(summary_cor_ml_rot[,5]))
L_cor_ml_rot
```

```
##      Factor 1 loadings Factor 2 loadings
## [1,]         0.7632891        0.02919252
## [2,]         0.8194973        0.23180592
## [3,]         0.6680336        0.10820147
## [4,]         0.1126721        0.99111380
## [5,]         0.1083670        0.67706236
```

```
zapsmall(Psi_cor_ml_rot)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.4165375 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.2746902 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.5420235 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.0049984 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.5298432
```

```
communality_cor_ml_rot
```

```
## [1] 0.5834625 0.7253098 0.4579765 0.9950016 0.4701568
```

Factor scores

```
#Function to calculate the wls and regression factor scores
factor_scores=function(data,L,Psi,covar=TRUE)
{
  if(covar==FALSE)
  {
    data=scale(data)
  }
  mean=apply(data,2,mean)
  centered_data=apply(data,1,function(x){x-mean})
  mat_wls=solve(t(L)%*%solve(Psi)%*%L)%*%t(L)%*%solve(Psi)
  mat_reg=t(L)%*%solve((L%*%t(L)+Psi))
  fa_scores_wls=t(mat_wls%*%centered_data)
  fa_scores_reg=t(mat_reg%*%centered_data)
  colname=rep("name",ncol(L))
  for(i in 1:ncol(L))
  {
    colname[i]=paste("Factor",i,"scores")
  }
  colnames(fa_scores_wls)=colnames(fa_scores_reg)=colname
  return(list(wls_scores=fa_scores_wls,reg_scores=fa_scores_reg))
}
```

```r
# Maximum Likelihood

# Correlation matrix

ml_cor_scores=factor_scores(stocks,L_cor_ml_rot,Psi_cor_ml_rot,covar=FALSE)
ml_cor_scores$wls_scores
```

```
##       Factor 1 scores Factor 2 scores
##  [1,]      0.25089981    -1.853673042
##  [2,]      0.44303403     0.247877355
##  [3,]     -0.48078787    -0.098356467
##  [4,]      0.79921288    -1.314979302
##  [5,]     -0.09859627     0.958815103
##  [6,]     -0.47081661     0.412852687
##  [7,]      0.95854949     0.881740654
##  [8,]      1.03632774    -0.035575475
##  [9,]     -1.07061954    -1.148517333
## [10,]      0.55016746    -0.977893217
## [11,]     -0.38455824    -0.448688158
## [12,]      0.95063323     1.103463503
## [13,]     -0.49050339     0.896700129
## [14,]     -2.32247346     1.521578564
## [15,]     -0.79322946    -1.036081617
## [16,]     -0.87303616     0.219041912
## [17,]     -2.02544305     1.243895126
## [18,]     -0.51699822    -0.323871578
## [19,]      0.45745408    -0.932793920
## [20,]      1.08613936     1.055726321
## [21,]      0.18913546    -0.094482226
## [22,]      0.66432280     0.336543848
## [23,]     -0.46301149     1.066608504
## [24,]     -0.41688545    -0.216355945
## [25,]     -0.40874267    -1.028943670
## [26,]     -1.34076342     0.540994418
## [27,]     -1.25243768     0.083560452
## [28,]      0.77018452    -2.104793431
## [29,]      0.66554572     0.346105450
## [30,]     -1.37643268    -0.696750595
## [31,]      1.16998809     0.569221031
## [32,]      2.30781951     0.185515112
## [33,]      1.20700717    -0.273910020
## [34,]      0.13030521     0.694130194
## [35,]      0.19356622    -0.323632119
## [36,]     -0.31326862     0.314485302
## [37,]     -1.86859542    -0.887069095
## [38,]      1.09224044     0.477355806
## [39,]     -0.43080029     0.385946770
## [40,]     -1.07083965     0.056225211
## [41,]     -2.35448131     0.323577936
## [42,]      2.54040791     0.563076333
## [43,]      2.09787440     0.424140549
## [44,]      0.62928957     0.726075921
## [45,]     -2.44515146    -0.004550540
## [46,]      0.33201720     1.012212838
```

```
##  [47,]    0.29980150  -0.380462411
##  [48,]    0.22314334  -0.823730115
##  [49,]    0.60411285  -0.216066605
##  [50,]    1.61710680   0.738001821
##  [51,]   -0.59629491   0.058649838
##  [52,]    0.31186044  -1.168899211
##  [53,]   -1.56631686   0.454379665
##  [54,]   -0.49825291   0.187168416
##  [55,]    0.27457346   0.723672841
##  [56,]    1.43012096   1.611394509
##  [57,]   -0.54220424  -0.253644255
##  [58,]   -1.51555952   1.651308390
##  [59,]   -0.03369121   0.421812255
##  [60,]    0.23651585   1.176968565
##  [61,]   -1.18682266  -1.042728867
##  [62,]   -0.74421921  -0.436733423
##  [63,]   -2.11191860  -1.822716745
##  [64,]   -0.13723483   0.449555202
##  [65,]    0.72741871   0.492196161
##  [66,]   -0.01496851  -1.503895081
##  [67,]    0.62007545   0.525273849
##  [68,]    1.49495118  -0.808425801
##  [69,]   -0.17563151   0.135601632
##  [70,]   -0.94571116  -1.859561511
##  [71,]    2.65863710   0.114295837
##  [72,]   -0.87536173   1.092101646
##  [73,]    0.21406640  -0.415236885
##  [74,]   -0.10220596   0.107244668
##  [75,]   -0.16275259   2.191072187
##  [76,]   -0.92607962  -0.050973944
##  [77,]   -0.78219916   1.396201850
##  [78,]    0.09450311  -0.342285629
##  [79,]    1.18039308  -1.441289308
##  [80,]   -1.43312796  -1.924088034
##  [81,]   -0.88097216   0.569638973
##  [82,]   -0.35982339   1.541657916
##  [83,]   -1.06697921   2.284051693
##  [84,]    0.90942334  -1.833138064
##  [85,]   -1.68733601  -0.743414998
##  [86,]    0.53573851   1.920684900
##  [87,]    1.22810060  -1.142728783
##  [88,]    0.58822623   0.243360476
##  [89,]   -0.92201823  -0.405073317
##  [90,]   -0.16057556   0.369146165
##  [91,]    0.35967503  -1.814062354
##  [92,]    0.06609171  -0.867911221
##  [93,]    0.50086003  -2.241041972
##  [94,]    1.82709700   1.817507433
##  [95,]    0.31722787   0.274770831
##  [96,]    3.04438565  -1.684716195
##  [97,]   -0.10869072   1.533091945
##  [98,]    1.57089692  -0.008510014
##  [99,]   -0.87311079  -0.120339207
## [100,]    0.06316359  -0.207170742
```

```
## [101,]      1.03125728    -0.876430071
## [102,]      0.17139025    -0.269180788
## [103,]     -1.04440486    -0.222903385
```

```
ml_cor_scores$reg_scores
```

```
##        Factor 1 scores Factor 2 scores
##   [1,]      0.16535936     -1.83427933
##   [2,]      0.36753194      0.25550902
##   [3,]     -0.39519061     -0.10792826
##   [4,]      0.62520427     -1.28789328
##   [5,]     -0.06003872      0.94945268
##   [6,]     -0.37607047      0.39963093
##   [7,]      0.80260418      0.89564164
##   [8,]      0.84651294     -0.01307238
##   [9,]     -0.89995355     -1.16280636
##  [10,]      0.42882295     -0.95869823
##  [11,]     -0.32403192     -0.45354744
##  [12,]      0.80088905      1.11551878
##  [13,]     -0.38178505      0.87939890
##  [14,]     -1.86615231      1.46024974
##  [15,]     -0.67075392     -1.04526913
##  [16,]     -0.70907351      0.19865550
##  [17,]     -1.62926516      1.19103786
##  [18,]     -0.42963369     -0.33251574
##  [19,]      0.35399035     -0.91592895
##  [20,]      0.91065136      1.07104967
##  [21,]      0.15260542     -0.08971023
##  [22,]      0.55035455      0.34825304
##  [23,]     -0.35566299      1.04861289
##  [24,]     -0.34547721     -0.22366485
##  [25,]     -0.35625378     -1.02993605
##  [26,]     -1.08456844      0.50813958
##  [27,]     -1.02216963      0.05605807
##  [28,]      0.58452609     -2.07236051
##  [29,]      0.55155952      0.35776861
##  [30,]     -1.14028633     -0.72101540
##  [31,]      0.96876616      0.59002052
##  [32,]      1.89079695      0.23362673
##  [33,]      0.98094285     -0.24594362
##  [34,]      0.12142678      0.69167939
##  [35,]      0.15131154     -0.31703307
##  [36,]     -0.24937354      0.30538720
##  [37,]     -1.54674979     -0.92045466
##  [38,]      0.90323063      0.49718158
##  [39,]     -0.34393137      0.37378692
##  [40,]     -0.87428600      0.03282559
##  [41,]     -1.91802401      0.27061730
##  [42,]      2.08905603      0.61332435
##  [43,]      1.72427063      0.46594422
##  [44,]      0.53006956      0.73408927
##  [45,]     -1.99919364     -0.05697638
##  [46,]      0.29316596      1.01168554
##  [47,]      0.23694763     -0.37115456
##  [48,]      0.16476357     -0.81271658
```

```
##  [49,]   0.48927222  -0.20147238
##  [50,]   1.33794063   0.76711845
##  [51,]  -0.48625780   0.04541315
##  [52,]   0.22989097  -1.15337336
##  [53,]  -1.27083371   0.41734029
##  [54,]  -0.40334372   0.17506382
##  [55,]   0.24001079   0.72409399
##  [56,]   1.20380405   1.62989798
##  [57,]  -0.44873480  -0.26336001
##  [58,]  -1.20365589   1.60631120
##  [59,]  -0.01849521   0.41790121
##  [60,]   0.21862120   1.17314702
##  [61,]  -0.99268871  -1.06031060
##  [62,]  -0.61782544  -0.44939951
##  [63,]  -1.76575900  -1.85425080
##  [64,]  -0.10255472   0.44321296
##  [65,]   0.60527971   0.50408259
##  [66,]  -0.04450366  -1.49284926
##  [67,]   0.51822816   0.53460724
##  [68,]   1.20489089  -0.77024156
##  [69,]  -0.14068272   0.13080857
##  [70,]  -0.81308683  -1.86579634
##  [71,]   2.17608883   0.17047247
##  [72,]  -0.69224351   1.06506645
##  [73,]   0.16610662  -0.40750563
##  [74,]  -0.08126018   0.10424127
##  [75,]  -0.08605352   2.17101947
##  [76,]  -0.75823367  -0.07045752
##  [77,]  -0.60955166   1.36886694
##  [78,]   0.06991976  -0.33767097
##  [79,]   0.93413793  -1.40507044
##  [80,]  -1.21297128  -1.94029258
##  [81,]  -0.70803979   0.54643236
##  [82,]  -0.26110684   1.52228559
##  [83,]  -0.82333223   2.24389621
##  [84,]   0.70419266  -1.79977107
##  [85,]  -1.39547446  -0.77399746
##  [86,]   0.47921460   1.91766185
##  [87,]   0.97954798  -1.10774299
##  [88,]   0.48614062   0.25414135
##  [89,]  -0.76251031  -0.42179335
##  [90,]  -0.12336270   0.36291091
##  [91,]   0.25514116  -1.79263428
##  [92,]   0.03541411  -0.85993326
##  [93,]   0.36140980  -2.21335754
##  [94,]   1.53278406   1.84297007
##  [95,]   0.26525289   0.27950008
##  [96,]   2.45286997  -1.60666584
##  [97,]  -0.05597069   1.51917236
##  [98,]   1.28414430   0.02525759
##  [99,]  -0.71641587  -0.13816206
## [100,]   0.04719620  -0.20424971
## [101,]   0.82432709  -0.84768029
## [102,]   0.13434926  -0.26346908
```

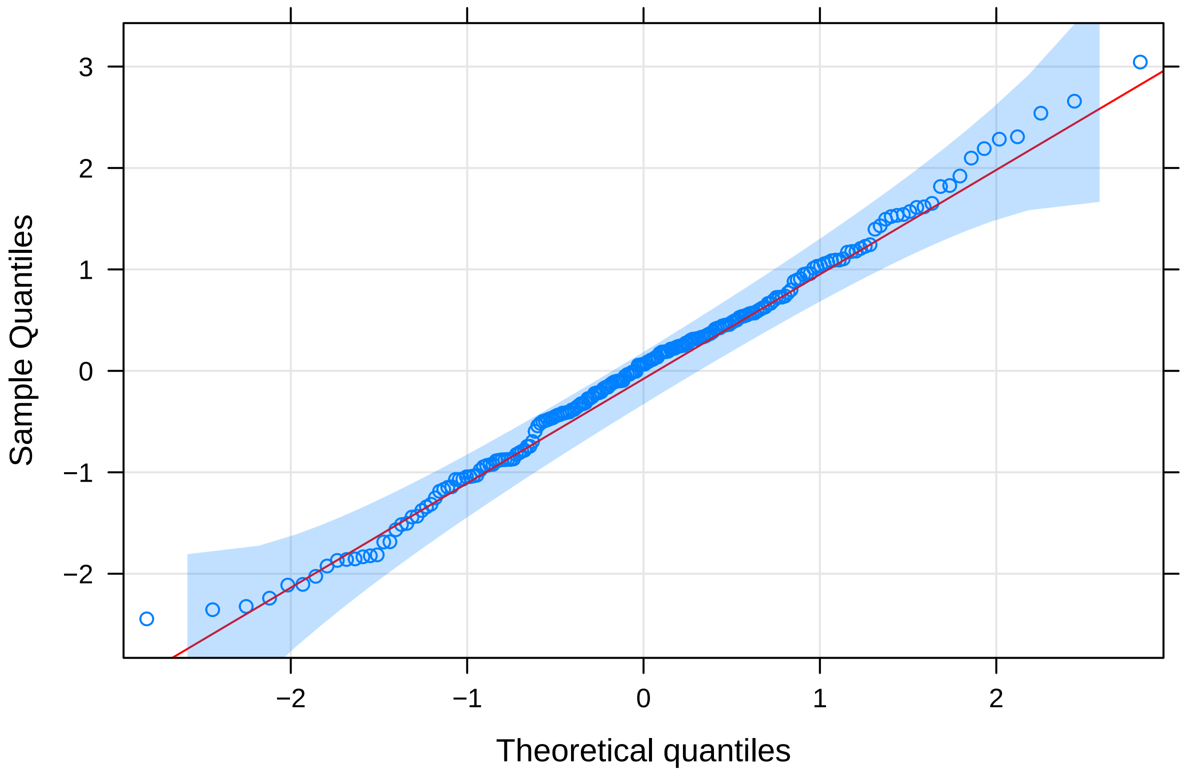
```
## [103,]    -0.85866221    -0.24362610
```

```r
# Checking normality and pairwise independence assumptions

# Weighted Least Squares Method

# QQPlot

# Factor 1 scores
qqmath(ml_cor_scores$wls_scores[,1], distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(ml_cor_scores$wls_scores, grid = TRUE)
  panel.qqmathline(ml_cor_scores$wls_scores, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```




```r
# Univariate normality testing using shapiro-wilks test
shapiro.test(ml_cor_scores$wls_scores[,1])
```

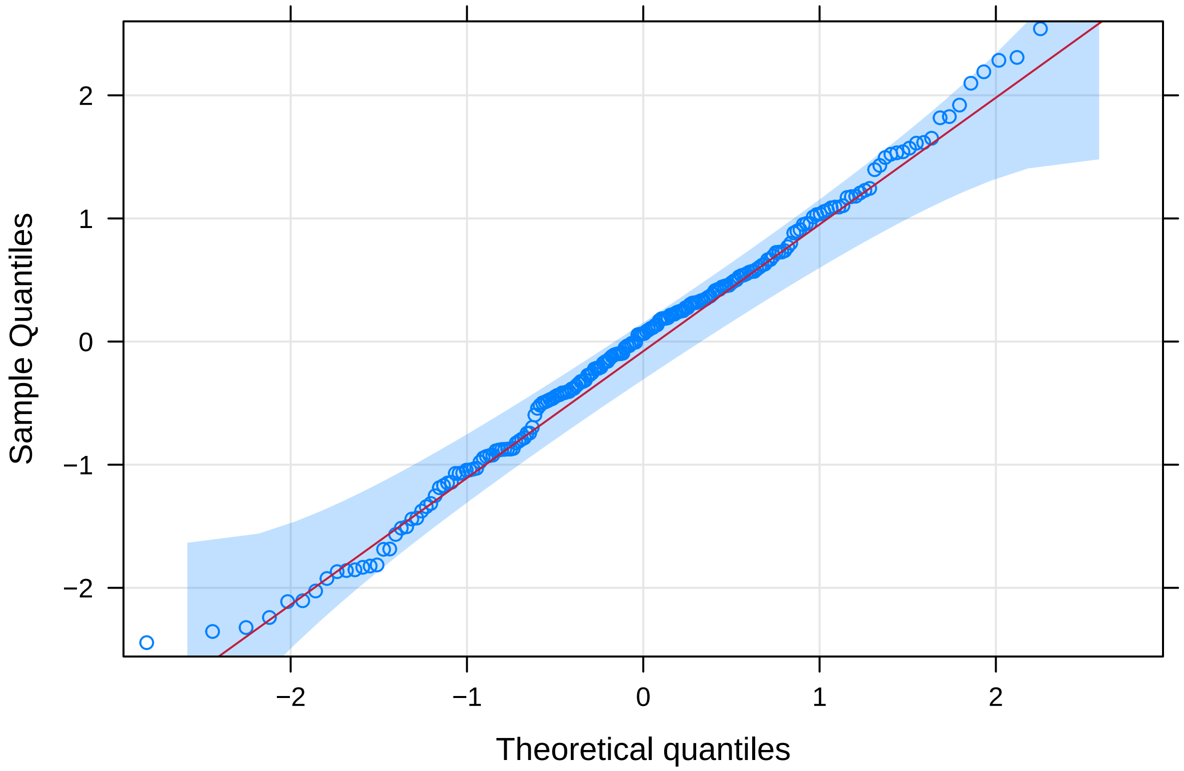
```
##
##  Shapiro-Wilk normality test
##
## data:  ml_cor_scores$wls_scores[, 1]
## W = 0.99163, p-value = 0.7779
```

```r
shapiro.test(ml_cor_scores$wls_scores[,1])$p.value
```

```
## [1] 0.7779096
```

```r
# Factor 2 scores
qqmath(ml_cor_scores$wls_scores[,2], distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(ml_cor_scores$wls_scores, grid = TRUE)
```

```r
  panel.qqmathline(ml_cor_scores$wls_scores, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```




```r
# Univariate normality testing using shapiro-wilks test
shapiro.test(ml_cor_scores$wls_scores[,2])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ml_cor_scores$wls_scores[, 2]
## W = 0.98594, p-value = 0.3497
```

```r
shapiro.test(ml_cor_scores$wls_scores[,2])$p.value
```

```
## [1] 0.3497038
```

```r
# Multivariate normality testing using multivaraite Shapiro-Wilks test
mvShapiro.Test(ml_cor_scores$wls_scores)
```

```
##
##  Generalized Shapiro-Wilk test for Multivariate Normality by
##  Villasenor-Alva and Gonzalez-Estrada
##
## data:  ml_cor_scores$wls_scores
## MVW = 0.98886, p-value = 0.6281
```

```r
mvShapiro.Test(ml_cor_scores$wls_scores)$p.value
```

```
## [1] 0.6281231
```

```r
# Checking pairwise independence of factor scores
cor.test(ml_cor_scores$wls_scores[,1], ml_cor_scores$wls_scores[,2], method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  ml_cor_scores$wls_scores[, 1] and ml_cor_scores$wls_scores[, 2]
## t = -0.23944, df = 101, p-value = 0.8112
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2163458  0.1704921
## sample estimates:
##         cor
## -0.02381842
```

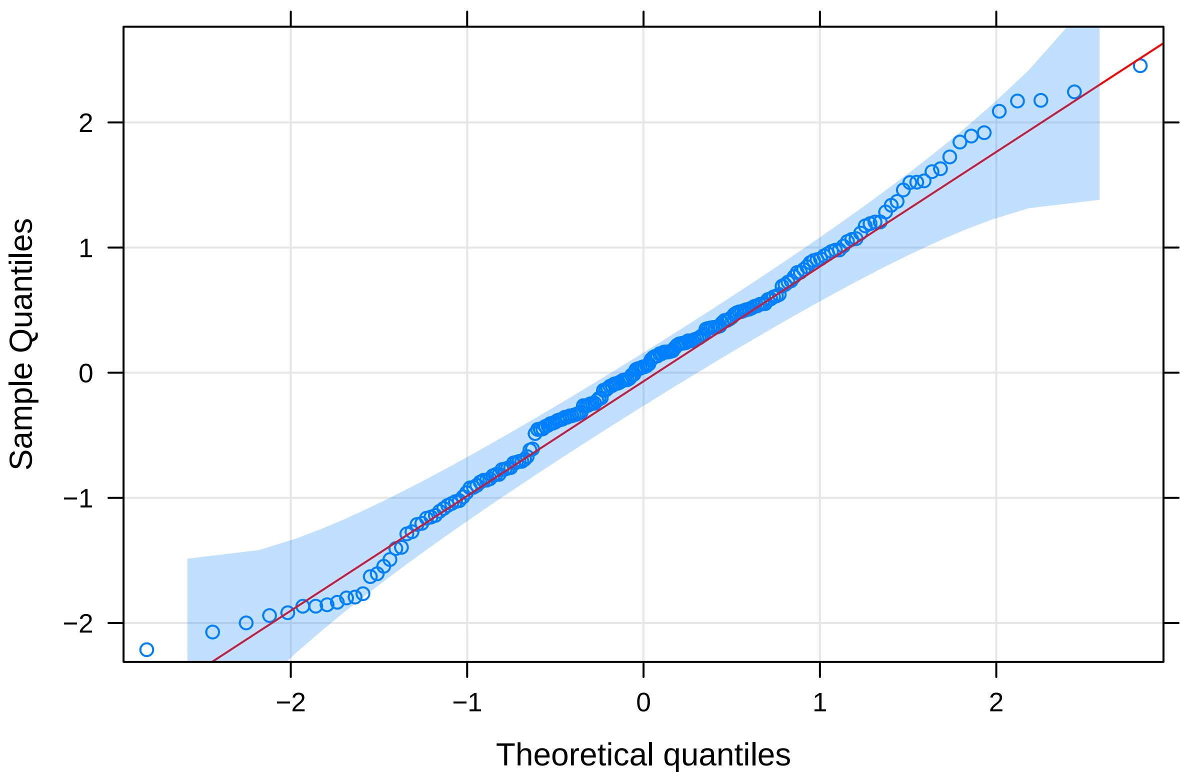
```r
cor.test(ml_cor_scores$wls_scores[,1], ml_cor_scores$wls_scores[,2], method = "pearson")$p.value
```

```
## [1] 0.8112496
```

```r
# Regression Method

# QQPlot

# Factor 1 scores
qqmath(ml_cor_scores$reg_scores[,1], distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(ml_cor_scores$reg_scores, grid = TRUE)
  panel.qqmathline(ml_cor_scores$reg_scores, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```



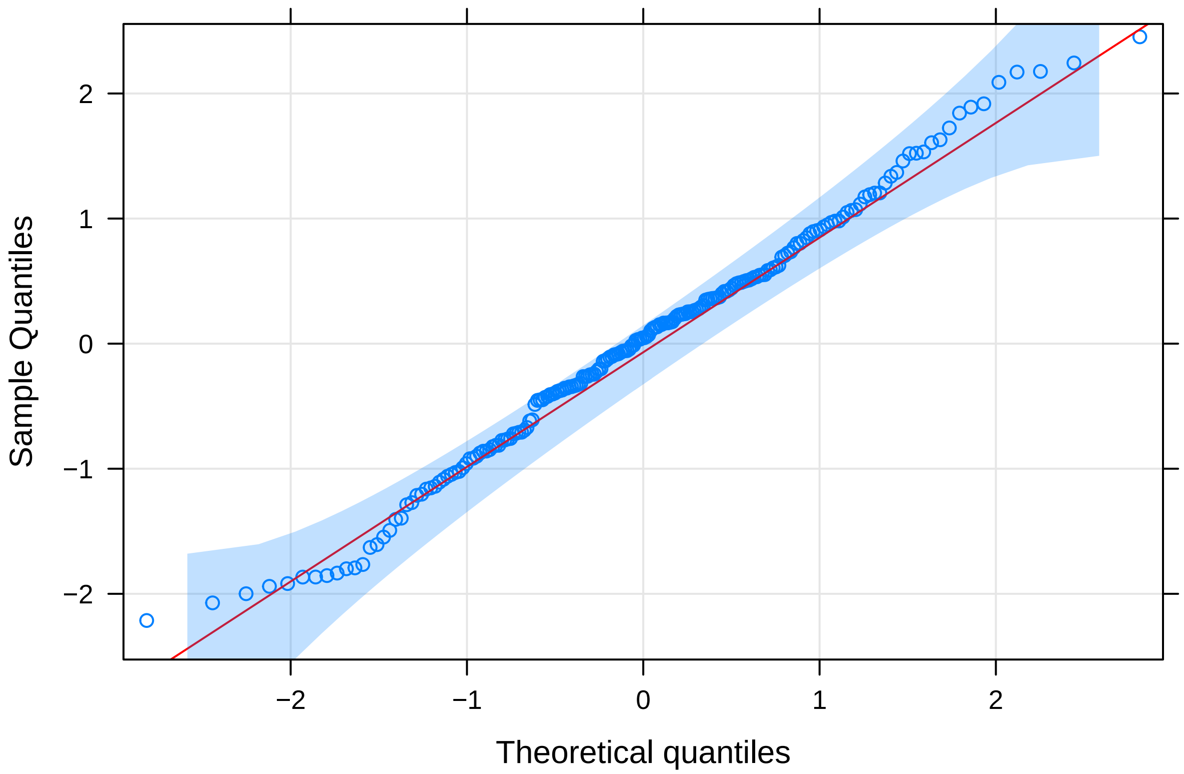
```r
# Univariate normality testing using shapiro-wilks test
shapiro.test(ml_cor_scores$reg_scores[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ml_cor_scores$reg_scores[, 1]
## W = 0.99118, p-value = 0.7423
```

```r
shapiro.test(ml_cor_scores$reg_scores[,1])$p.value
```

```
## [1] 0.7423269
```

```r
# Factor 2 scores
qqmath(ml_cor_scores$reg_scores[,2], distribution = qnorm, panel = function(x, ...) {
  panel.qqmath(ml_cor_scores$reg_scores, grid = TRUE)
  panel.qqmathline(ml_cor_scores$reg_scores, col = "red")
  panel.qqmathci(x, y = x, ci = 0.95)},
  xlab="Theoretical quantiles",
  ylab="Sample Quantiles")
```




```r
# Univariate normality testing using shapiro-wilks test
shapiro.test(ml_cor_scores$reg_scores[,2])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ml_cor_scores$reg_scores[, 2]
## W = 0.98649, p-value = 0.3831
```

```r
shapiro.test(ml_cor_scores$reg_scores[,2])$p.value
```

```
## [1] 0.3830935
```

```r
# Multivariate normality testing using multivaraite Shapiro-Wilks test
mvShapiro.Test(ml_cor_scores$reg_scores)
```

```
##
##  Generalized Shapiro-Wilk test for Multivariate Normality by
##  Villasenor-Alva and Gonzalez-Estrada
##
## data:  ml_cor_scores$reg_scores
## MVW = 0.98886, p-value = 0.6281
```

```r
mvShapiro.Test(ml_cor_scores$reg_scores)$p.value
```

```
## [1] 0.6281231
```

```r
# Checking pairwise independence of factor scores
cor.test(ml_cor_scores$reg_scores[,1], ml_cor_scores$reg_scores[,2], method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  ml_cor_scores$reg_scores[, 1] and ml_cor_scores$reg_scores[, 2]
## t = 0.23943, df = 101, p-value = 0.8113
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1704926  0.2163453
## sample estimates:
##        cor
## 0.02381788
```

```r
cor.test(ml_cor_scores$reg_scores[,1], ml_cor_scores$reg_scores[,2], method = "pearson")$p.value
```

```
## [1] 0.8112538
```