Rcode.R

Alice Li

Mon Jul 17 15:14:38 2017

```
rm(list=ls())
#setwd("C:/Users/Tong/Desktop/UWaterloo/2017 Summer/ACTSC445/A4")
library("tseries")
```

```
## Warning: package 'tseries' was built under R version 3.4.1
```

```
library("Kendall")
```

```
## Warning: package 'Kendall' was built under R version 3.4.1
```

```
library("QRM")
```

```
## Loading required package: gsl
## Loading required package: Matrix
## Loading required package: mvtnorm
## Loading required package: numDeriv
## Loading required package: timeSeries
## Loading required package: timeDate
##
## Attaching package: 'QRM'
## The following object is masked from 'package:Kendall':
##
##     Kendall
## The following object is masked from 'package:base':
##
##     lbeta
```

```
library("evd")
library("copula")
```

```
## Warning: package 'copula' was built under R version 3.4.1
##
## Attaching package: 'copula'
## The following objects are masked from 'package:gsl':
##
##     psi, sinc
```

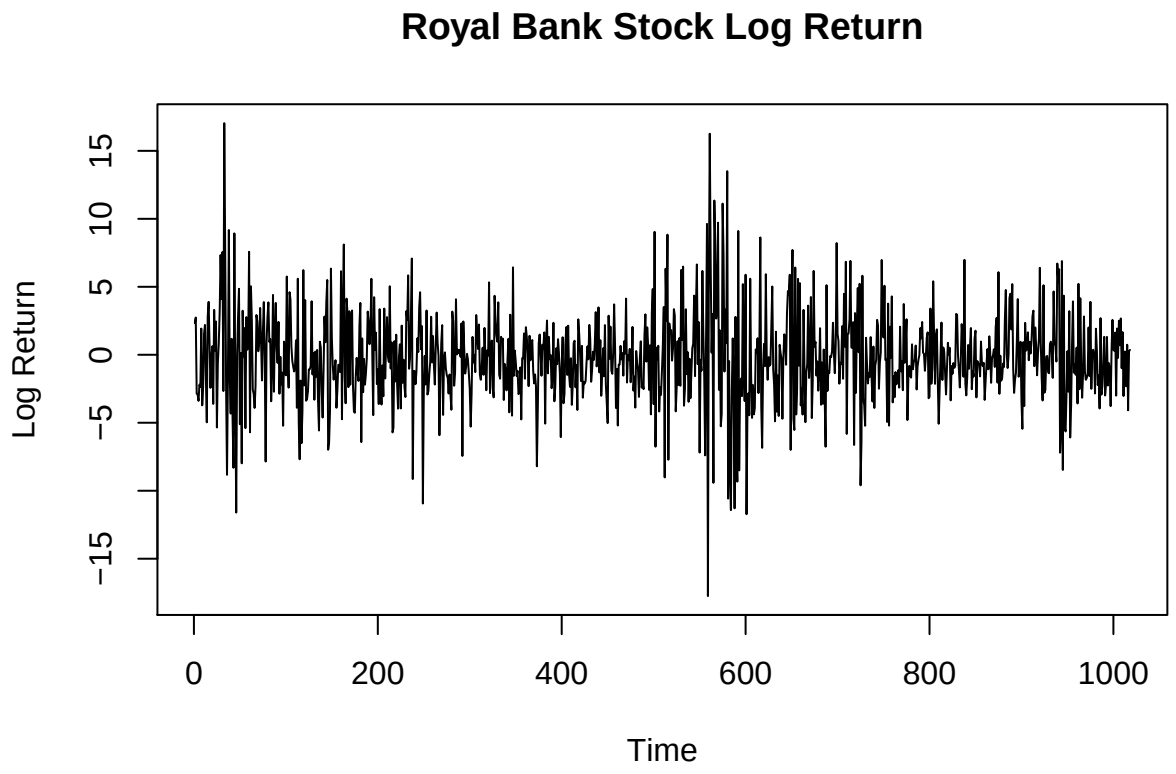
```
#1)
RY<-read.csv("RY.csv")
MS<-read.csv("MS.csv")
a_array <- rep(0,nrow(RY)-1)
b_array <- rep(0,nrow(MS)-1)

for (i in 1:nrow(RY)-1){
  a_array[i] = -100*log(RY$Adj.Close[i+1]/RY$Adj.Close[i])
```

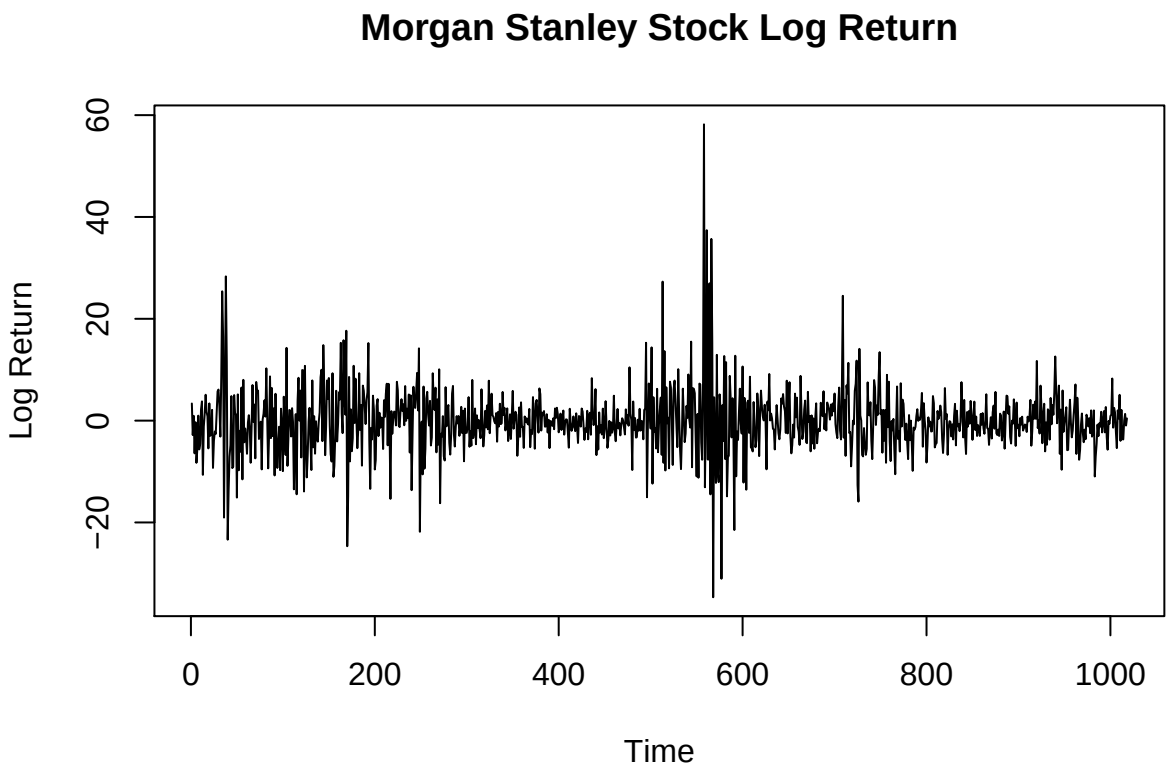
```
}

for (i in 1:nrow(MS)-1){
  b_array[i] = -100*log(MS$Adj.Close[i+1]/MS$Adj.Close[i])
}

plot.ts(a_array,ylab="Log Return", main="Royal Bank Stock Log Return")
```




```
plot.ts(b_array,ylab="Log Return", main="Morgan Stanley Stock Log Return")
```



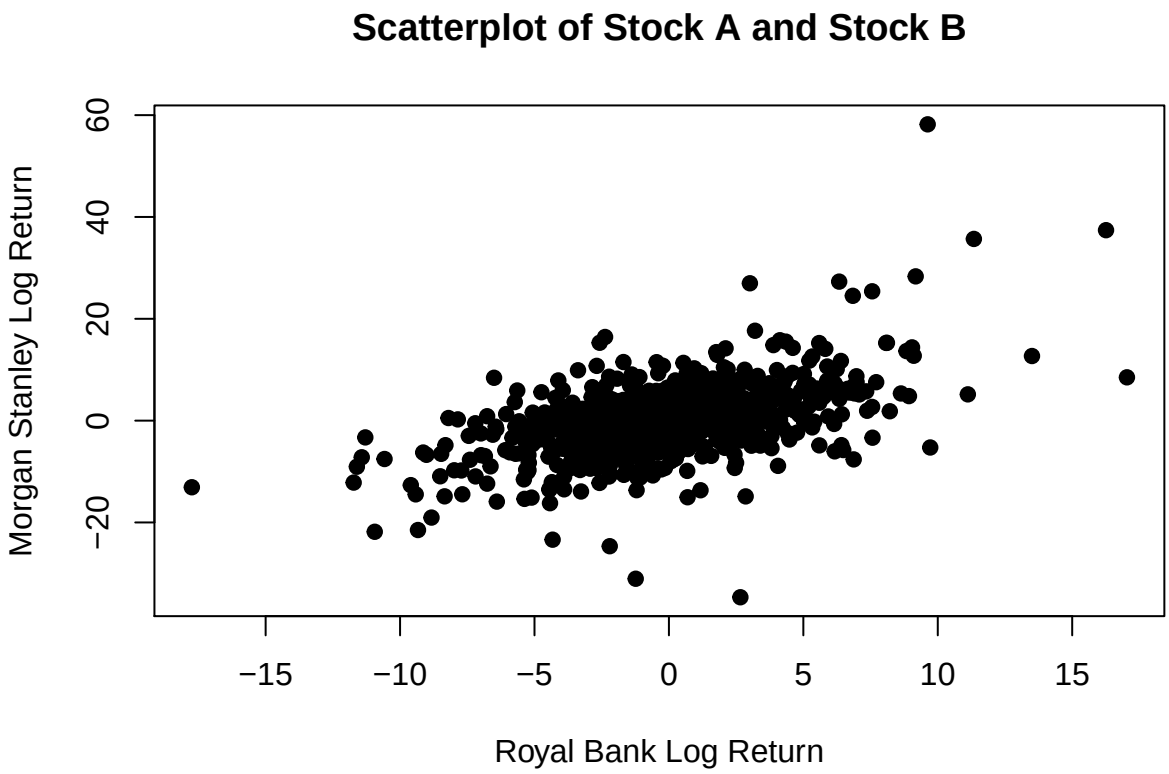

```
#2)
#install package tseries
jarque.bera.test(a_array)
```

```
##
##  Jarque Bera Test
##
## data:  a_array
## X-squared = 272.82, df = 2, p-value < 2.2e-16
```

```
jarque.bera.test(b_array)
```

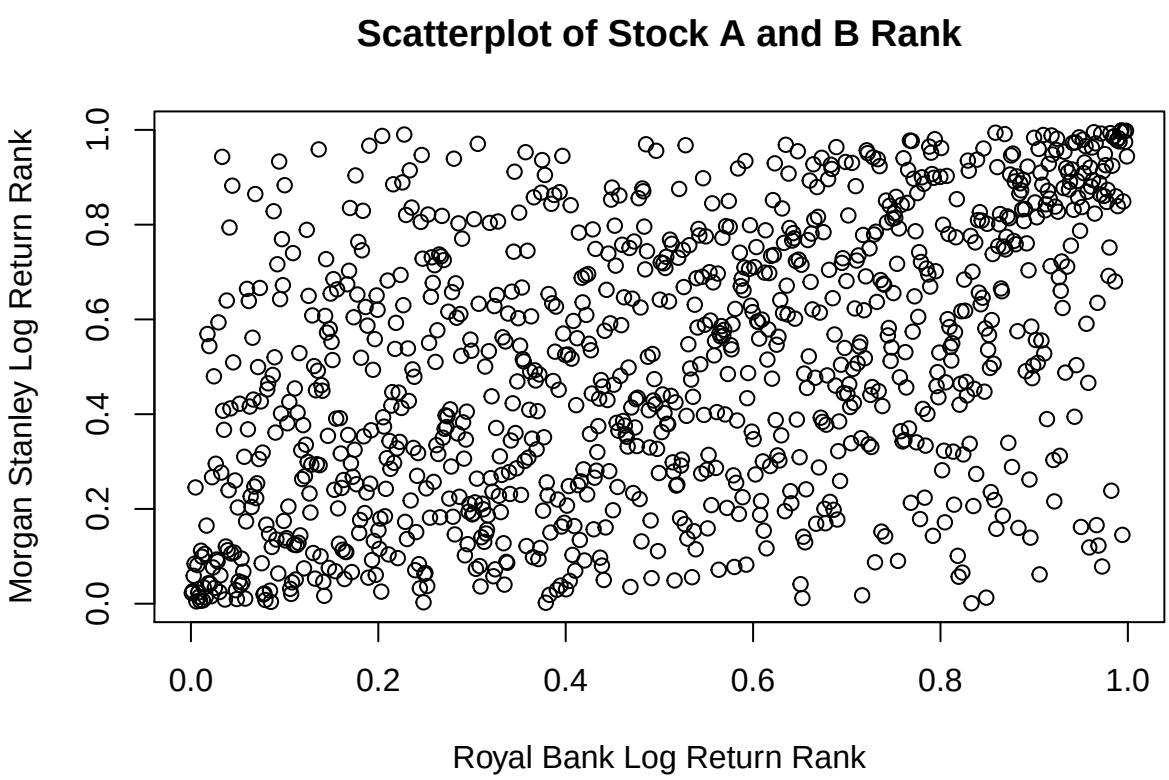
```
##
##  Jarque Bera Test
##
## data:  b_array
## X-squared = 5548.8, df = 2, p-value < 2.2e-16
```

```
#3)
plot(a_array, b_array, main="Scatterplot of Stock A and Stock B",
     xlab="Royal Bank Log Return", ylab="Morgan Stanley Log Return", pch=19)
```




```r
#4)
ranka_array<-rank(a_array, ties.method = 'random')
rankb_array<-rank(b_array, ties.method = 'random')
n<-length(a_array)
ra<-ranka_array/(n+1)
rb<-rankb_array/(n+1)

plot(ra, rb, main="Scatterplot of Stock A and B Rank",
     xlab="Royal Bank Log Return Rank", ylab="Morgan Stanley Log Return Rank")
```




```
#5)
#a)
Kendall(ra,rb)
```

```
## [1] 0.3301227
```

```
rab<-cbind(ra,rb)
#Upper right conner of top 70%
UR70<-rab[ra>0.7 & rb>0.7,]
Kendall(UR70[,1],UR70[,2])
```

```
## [1] 0.2422764
```

```
#Upper right conner of top 80%
UR80<-rab[ra>0.8 & rb>0.8,]
Kendall(UR80[,1],UR80[,2])
```

```
## [1] 0.2359596
```

```
#Upper right conner of top 95%
UR95<-rab[ra>0.95 & rb>0.95,]
Kendall(UR95[,1],UR95[,2])
```

```
## [1] 0.3235294
```

```
#Upper right conner of top 99%
UR99<-rab[ra>0.99 & rb>0.99,]
Kendall(UR99[,1],UR99[,2])
```

```
## [1] 0.3333333
```

```r
#Upper right conner of top 99.5%
UR995<-rab[ra>0.995 & rb>0.995,]
Kendall(UR995[,1],UR995[,2])
```

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

```r
#Lower left conner of top 30%
LL30<-rab[ra<0.3 & rb<0.3,]
Kendall(LL30[,1],LL30[,2])
```

```
## [1] 0.2393999
```

```r
#Lower left conner of top 20%
LL20<-rab[ra<0.2 & rb<0.2,]
Kendall(LL20[,1],LL20[,2])
```

```
## [1] 0.2671172
```

```r
#Lower left conner of top 5%
LL5<-rab[ra<0.05 & rb<0.05,]
Kendall(LL5[,1],LL5[,2])
```

```
## [1] 0.1029412
```

```r
#Lower left conner of top 1%
LL1<-rab[ra<0.01 & rb<0.01,]
Kendall(LL1[,1],LL1[,2])
```

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

```r
#Lower left conner of top 0.5%
LL05<-rab[ra<0.005 & rb<0.005,]
Kendall(LL05[,1],LL05[,2])
```

```
## [1] NA
```

```r
#b)
UTD95 <- length(rab[ra>0.95 & rb > 0.95])/length(rab[rb > 0.95])
UTD99 <- length(rab[ra>0.99 & rb > 0.99])/length(rab[rb > 0.99])
UTD995 <-length(rab[ra>0.995 & rb > 0.995,])/length(rab[rb > 0.995])
UTD95
```

```
## [1] 0.34
```

```r
UTD99
```

```
## [1] 0.4
```

```r
UTD995
```

```
## [1] 0.4
```

```r
LTD5 <- length(rab[ra<0.05 & rb < 0.05])/length(rab[rb < 0.05])
LTD1 <- length(rab[ra<0.01 & rb < 0.01])/length(rab[rb < 0.01])
LTD05 <- length(rab[ra<0.005 & rb < 0.005])/length(rab[rb < 0.005])
LTD5
```

```
## [1] 0.34
```

```r
LTD1
```

```
## [1] 0.2
```

```
LTD05
```

```
## [1] 0
```

```
UTD95a <- length(rab[ra>0.95 & rb > 0.95])/length(rab[ra > 0.95])
UTD99a <- length(rab[ra>0.99 & rb > 0.99])/length(rab[ra > 0.99])
UTD995a <-length(rab[ra>0.995 & rb > 0.995,])/length(rab[ra > 0.995])
UTD95a
```

```
## [1] 0.34
```

```
UTD99a
```

```
## [1] 0.4
```

```
UTD995a
```

```
## [1] 0.4
```

```
LTD5a <- length(rab[ra<0.05 & rb < 0.05])/length(rab[ra < 0.05])
LTD1a <- length(rab[ra<0.01 & rb < 0.01])/length(rab[ra < 0.01])
LTD05a <- length(rab[ra<0.005 & rb < 0.005])/length(rab[ra < 0.005])
LTD5a
```

```
## [1] 0.34
```

```
LTD1a
```

```
## [1] 0.2
```

```
LTD05a
```

```
## [1] 0
```

```
#c)

#6
#Gumbel 1
gumfit <- fit.AC(rab, name = "gumbel")
#Gaussian 1
gausfit <- fit.gausscopula(rab)
#student t 2
tfit <- fit.tcopula(rab)
```

```
## Warning in FUN(newX[, i], ...): NaNs produced
```

```
## Warning in FUN(newX[, i], ...): NaNs produced
```

```
## Warning in log(pi * df): NaNs produced
```

```
## Warning in nlminb(theta, negloglik1, data = Udata, ...): NA/NaN function
## evaluation
```

```
## Warning in FUN(newX[, i], ...): NaNs produced
```

```
## Warning in FUN(newX[, i], ...): NaNs produced
```

```
## Warning in log(pi * df): NaNs produced
```

```
## Warning in nlminb(theta, negloglik1, data = Udata, ...): NA/NaN function
## evaluation
```

```
gumfit
```

```
## $ll.max
## [1] 159.5913
##
## $theta
## [1] 1.4944
##
## $se
## [1] 0.0367197
##
## $converged
## [1] TRUE
##
## $fit
## $fit$par
## [1] 1.4944
##
## $fit$objective
## [1] -159.5913
##
## $fit$convergence
## [1] 0
##
## $fit$iterations
## [1] 6
##
## $fit$evaluations
## function gradient
##        8        8
##
## $fit$message
## [1] "relative convergence (4)"
```

```
gausfit
```

```
## $P
##           [,1]      [,2]
## [1,] 1.0000000 0.5087614
## [2,] 0.5087614 1.0000000
##
## $converged
## [1] TRUE
##
## $ll.max
## [1] 149.9969
##
## $fit
## $fit$par
## [1] 0.5909594
##
## $fit$objective
## [1] -149.9969
##
## $fit$convergence
## [1] 0
##
```

```
## $fit$iterations
## [1] 5
##
## $fit$evaluations
## function gradient
##        7        8
##
## $fit$message
## [1] "relative convergence (4)"
```

```
tfit
```

```
## $P
##          [,1]     [,2]
## [1,] 1.000000 0.179839
## [2,] 0.179839 1.000000
##
## $nu
## [1] 0.4572832
##
## $converged
## [1] TRUE
##
## $ll.max
## [1] 1092.219
##
## $fit
## $fit$par
## [1] 0.4572832 0.1828197
##
## $fit$objective
## [1] -1092.219
##
## $fit$convergence
## [1] 0
##
## $fit$iterations
## [1] 19
##
## $fit$evaluations
## function gradient
##       30       45
##
## $fit$message
## [1] "relative convergence (4)"
```

```
gumllmax<-gumfit$ll.max
gausllmax<-gausfit$ll.max
tllmax<-tfit$ll.max

gumllmax
```

```
## [1] 159.5913
```

```
gausllmax
```

```
## [1] 149.9969
```

```
tllmax
```

```
## [1] 1092.219
```

```
gumAIC<- -2*gumllmax+2
gausAIC<- -2*gausllmax+2
tAIC<- -2*tllmax+4
```