STAT509-001-HW5-Xinye Xu

Q2

(a) Carry out a preliminary data analysis, including skewness, kurtosis, correlational anal- ysis and scatter diagrams, and provide a summary discussion of your findings.

NYB is negative skewness (-0.3481627), ALTR (0.2738863) and APH (0.4139656) are positive skewed. APH has the largest Kurtosis (4.0401221), which means more higher peak than normal, then 3.6611891 for ALTR and NYB is 1.3195310. The largest correlation between variables happen between ALTR and APH, which is 0.3810, although it seems not obvious in the scatterplot.

```r
library(moments)
library(Ecdat)
```

```
## Loading required package: Ecfun

##
## Attaching package: 'Ecfun'

## The following object is masked from 'package:base':
##
##     sign

##
## Attaching package: 'Ecdat'

## The following object is masked from 'package:datasets':
##
##     Orange
```

```r
library(fGarch)
```

```
## Loading required package: timeDate

## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018i.
## 1.0/zoneinfo/America/Detroit'

##
## Attaching package: 'timeDate'

## The following objects are masked from 'package:moments':
##
##     kurtosis, skewness

## Loading required package: timeSeries

## Loading required package: fBasics

##

## Rmetrics Package fBasics

## Analysing Markets and calculating Basic Statistics

## Copyright (C) 2005-2014 Rmetrics Association Zurich

## Educational Software for Financial Engineering and Computational Science

## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.

## https://www.rmetrics.org --- Mail to: info@rmetrics.org
```

```r
library(copula)
```

```
## Warning: package 'copula' was built under R version 3.4.4
```

```r
library(fCopulae)
```

```
## Loading required package: fMultivar
```

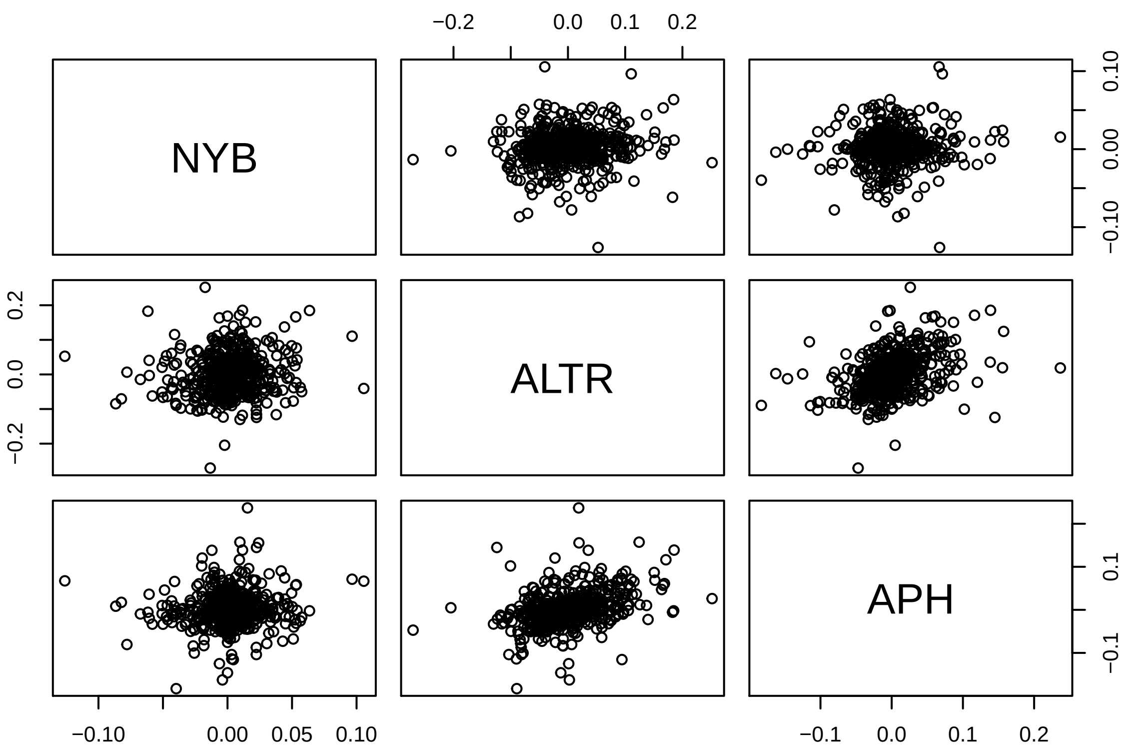
```r
Data = read.csv("Data/midcapD.csv", header=TRUE)
Mid = Data[,c(5,6,7)]

nyb = Mid$NYB
altr = Mid$ALTR
aph = Mid$APH
apply(Mid, 2, function (x) c(skewness(x),kurtosis(x)) )
```

```
##             NYB      ALTR       APH
## [1,] -0.3481627 0.2738863 0.4139656
## [2,]  3.6611891 1.3195310 4.0401221
```

```r
signif( cor(Mid), digits=3 )
```

```
##         NYB  ALTR    APH
## NYB  1.0000 0.116 0.0755
## ALTR 0.1160 1.000 0.3810
## APH  0.0755 0.381 1.0000
```

```r
pairs(Mid)
```



(b) Carry out a fitting of a multivariate normal distribution to the returns and carry out diagnostic plots – univariate QQ plots for each.

Calculate the sample mean and sample covariance matrix. From the QQ plots, distributions of NYB return and APH return have significantly heavier tails than normal. ALTR seems to be more linear in the line, meaning it is much close to normal distribution.

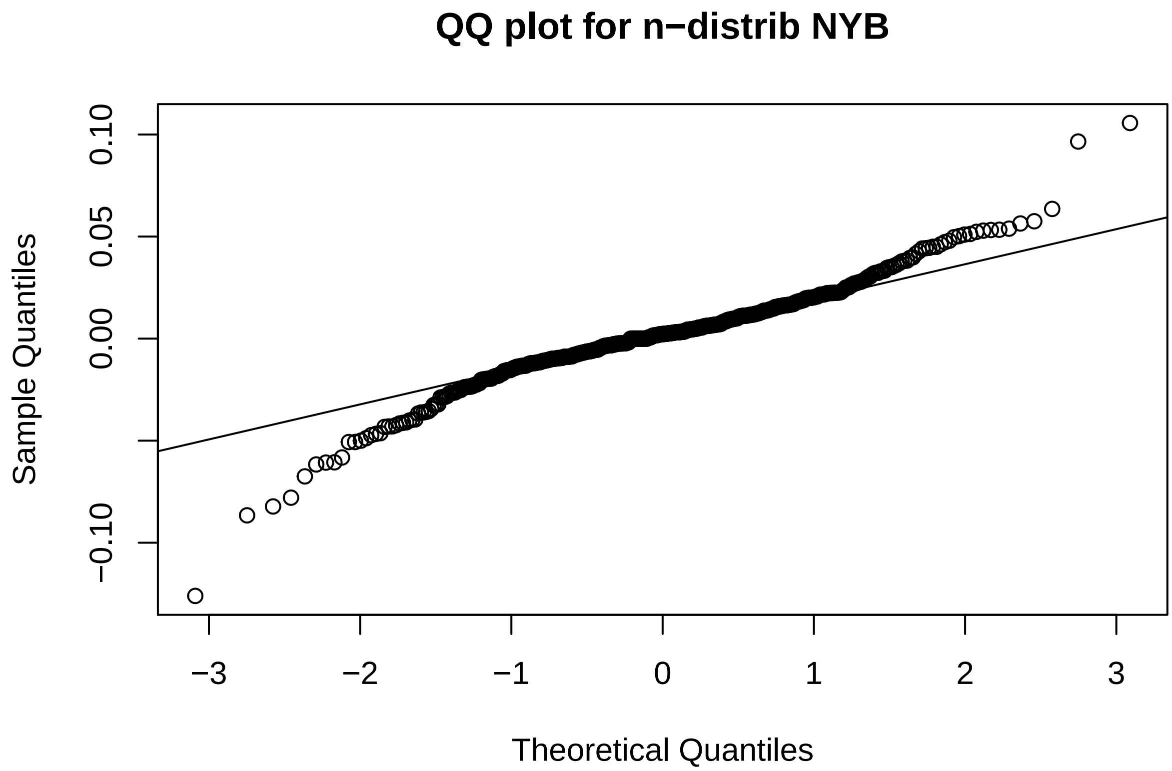
```
library(timeSeries)
library(mnormt)
signif(colMeans(Mid), digits=3)
```

```
##     NYB    ALTR     APH
## 0.00169 0.00149 0.00161
```

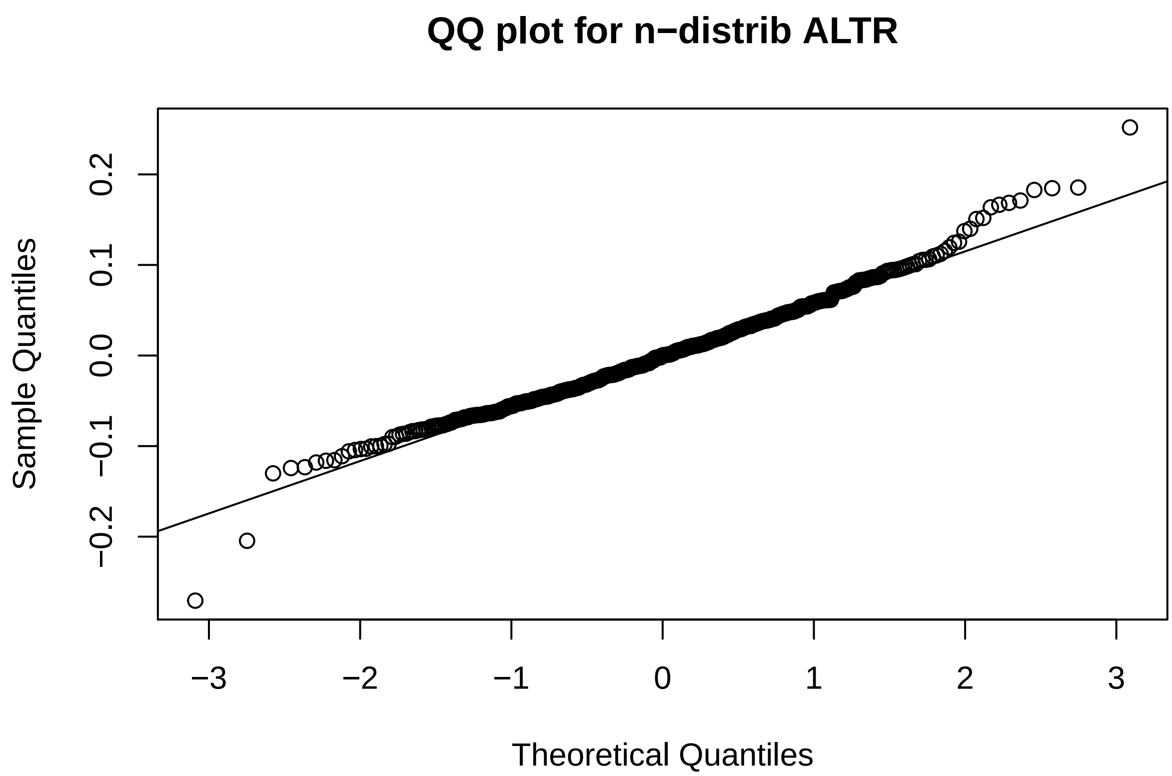
```
signif(cov(Mid), digits=3)
```

```
##           NYB     ALTR      APH
## NYB  5.41e-04 0.000162 7.39e-05
## ALTR 1.62e-04 0.003620 9.67e-04
## APH  7.39e-05 0.000967 1.77e-03
```

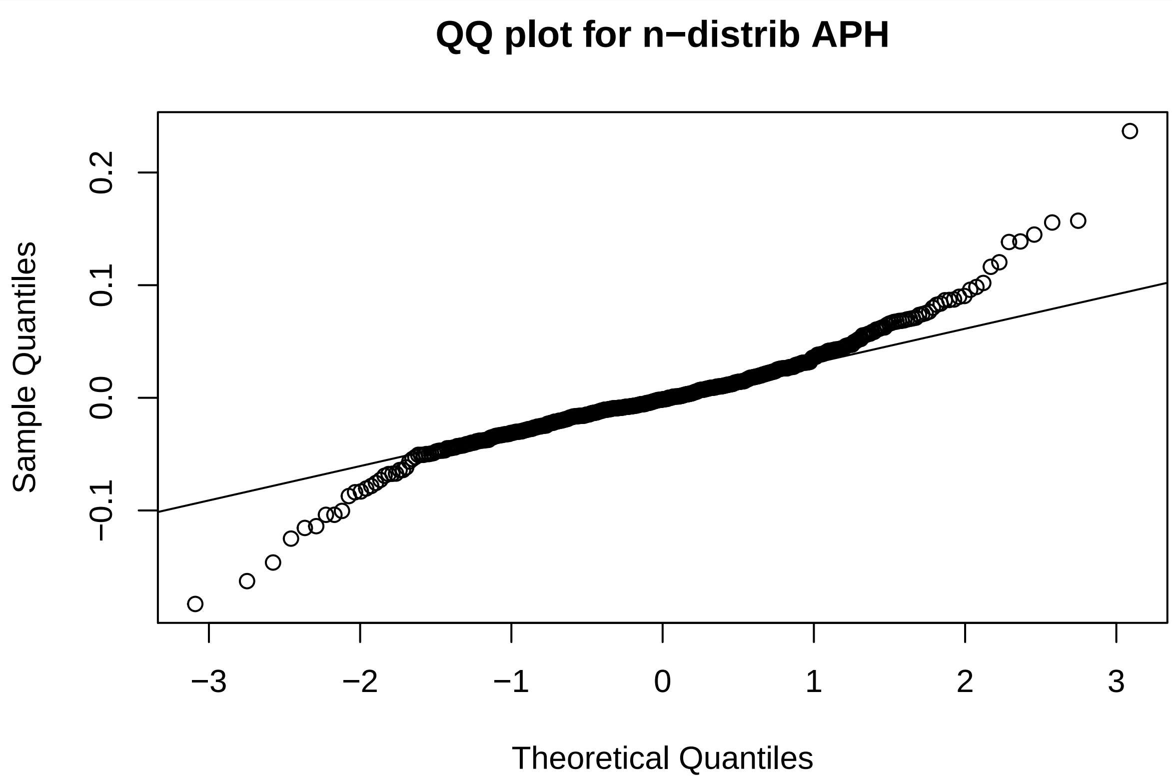
```
qqnorm(nyb, main="QQ plot for n-distrib NYB")
qqline(nyb)
```




```
qqnorm(altr, main="QQ plot for n-distrib ALTR")
qqline(altr)
```




```r
qqnorm(aph, main="QQ plot for n-distrib APH")
qqline(aph)
```



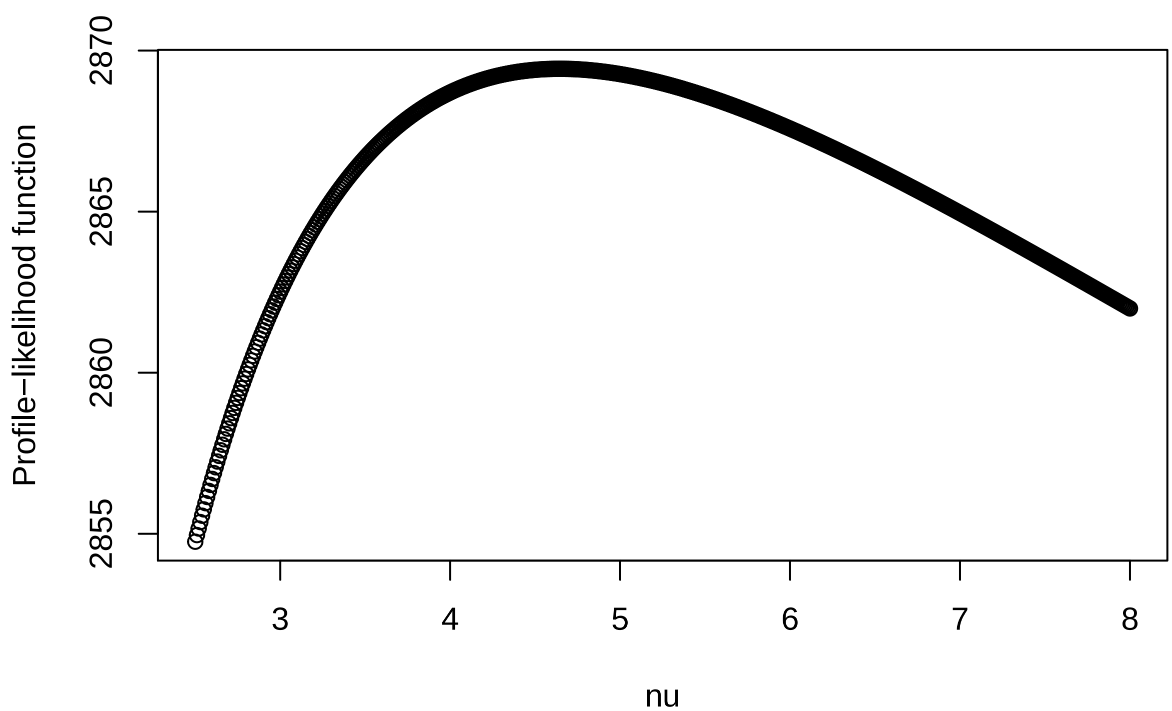
(c) Same as (b), but now use a multivariate t distribution – also derive a confidence interval for the degrees of freedom via the method of profile likelihood.

From the first plot, we can notice that MLE happens around 4.5. We also derive the confidence interval for the degree is [3.665, 6.025] Notice from the plots, NYB and APH have heavier tails than T distribution (nu=4.64). ALTR has a lighter tail than T distribution, which follows the above conclusion that it is much closed to normal.

```
# Chapter 6, slide page 41
library(MASS)
```

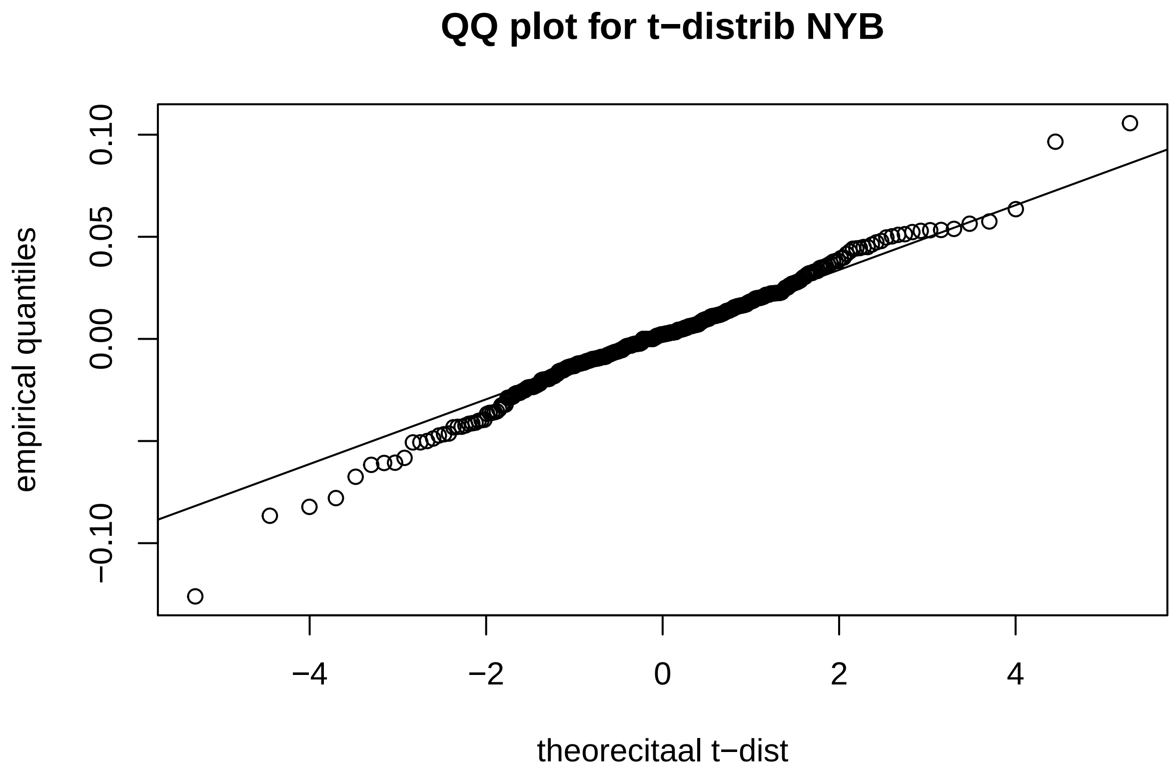
```
## Warning: package 'MASS' was built under R version 3.4.4

##
## Attaching package: 'MASS'

## The following object is masked from 'package:Ecdat':
##
##     SP500
```

```
df = seq(2.5, 8, 0.01)
n = length(df)
loglik_max = rep(0,n)
for (i in 1:n){
  fit = cov.trob(Mid, nu=df[i])
  mu = as.vector(fit$center)
  sigma = matrix(fit$cov, nrow=3)
  loglik_max[i] = sum(log(dmt(Mid, mean=fit$center, S=fit$cov, df=df[i])))}
plot(df, loglik_max, xlab='nu', ylab='Profile-likelihood function')
```

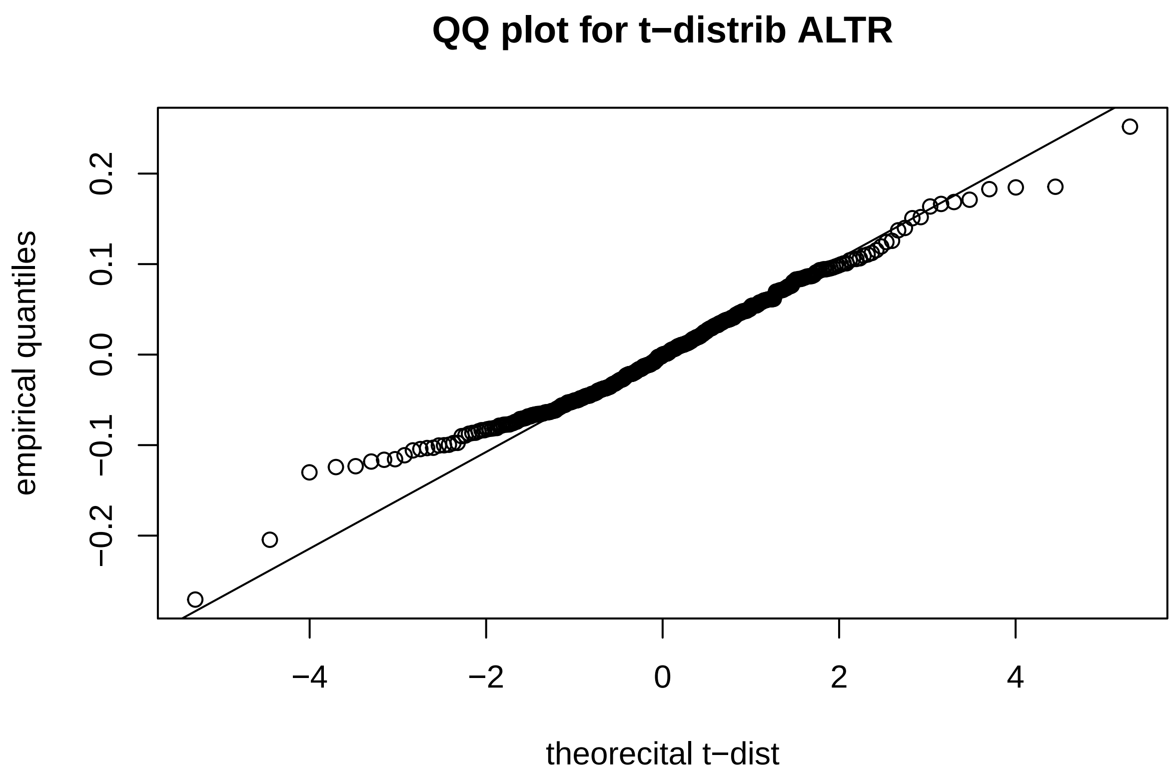



```
nuest = df[which.max(loglik_max)]
fitfinal = cov.trob(Mid, nu=nuest)
muest = fitfinal$center
lambdaest = fitfinal$cov

N = length(Mid[,1])
quantv = (1/(N+1))*seq(1,N,1)
```

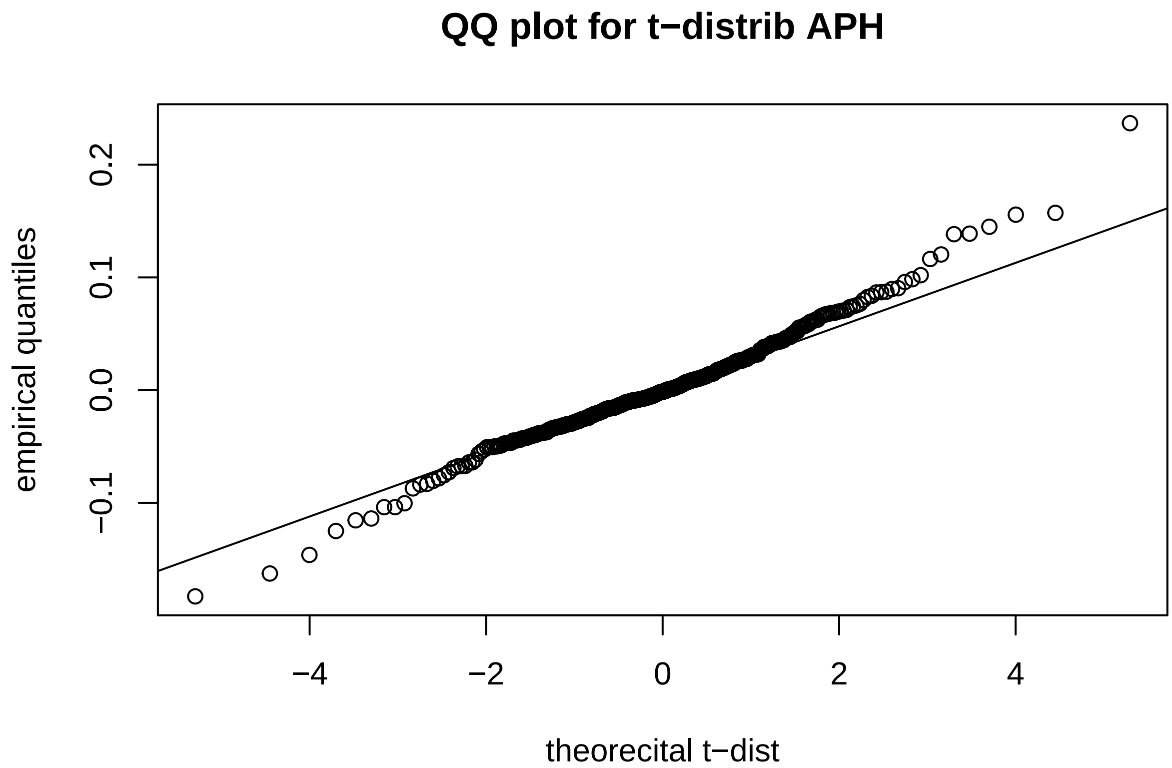
```
qqplot(qt(quantv, nuest), sort(Mid[,1]), main='QQ plot for t-distrib NYB', xlab='theorecitaal t-dist', y
abline(lm(quantile(Mid[,1], c(.25,.75))~qt(c(.25,.75), nuest )))
```




```
qqplot(qt(quantv, nuest), sort(Mid[,2]), main='QQ plot for t-distrib ALTR', xlab='theorecital t-dist', y
abline(lm(quantile(Mid[,2], c(.25,.75))~qt(c(.25,.75), nuest )))
```



```r
qqplot(qt(quantv, nuest), sort(Mid[,3]), main='QQ plot for t-distrib APH', xlab='theorecital t-dist', yl
abline(lm(quantile(Mid[,3], c(.25,.75))~qt(c(.25,.75), nuest )))
```




```r
ka_value = (1/2)*qchisq(0.95, 1)
CI = signif(max(loglik_max) - ka_value ,digits = 6)
CI
```

```
## [1] 2867.51
```

```r
CI_ind = which(CI == signif(loglik_max,digits = 6))
CI_ind #  117 354
```

```
## [1] 117 354
```

```r
0.5 * (df[117] + df[118])
```

```
## [1] 3.665
```

```r
0.5 * (df[353] + df[354])
```

```
## [1] 6.025
```

(d) Based on results in (b) and (c), which model do you prefer and why. Compare the two models of multivariate normal vs. multivariate t using the AIC criteria.

AIC for multivariate normal vs. multivariate t is -5553.348 -5718.868 respectively,

```r
sampmean = signif(colMeans(Mid),digits = 3)
sampcov = signif(cov(Mid),digits = 4)
logliken_max = sum(log(dmnorm(Mid,mean=sampmean,sampcov)))
logliket_max = loglik_max[which.max(loglik_max)]
AIC_mnorm = -2*logliken_max + 2*(3+6)
AIC_mt = -2*logliket_max + 2*(3+6+1)
```

```r
c(AIC_mnorm,AIC_mt)
```

```
## [1] -5553.348 -5718.868
```

Q3

(a) Based on the estimated multivariate model in 2-(b), derive the optimal portfolio between the 3 stocks that minimizes volatility.

optimal portfolio for these 3 stocks are weighted as 0.77042794, 0.03385205 and 0.19572001, and then the minimal vol = 0.02089274.

```r
COV = cov(Mid)
e = c(1,1,1)
portfolio = ( ginv(COV) %*% e ) / as.double(t(e) %*% ginv(COV) %*% e )
portfolio
```

```
##             [,1]
## [1,] 0.77042794
## [2,] 0.03385205
## [3,] 0.19572001
```

```r
sqrt(t(portfolio) %*% COV %*% portfolio)
```

```
##             [,1]
## [1,] 0.02089274
```

(b) Based on the estimated multivariate model in 2-(b), derive the optimal portfolio be- tween the 3 stocks that minimizes relative VaR at q = .002.

Best weights are 0.77 0.03 0.20 for each stock, and then the minimal VaR = 0.0584757 * Price

```r
weigh = matrix(0, 101*101,3)
count = 1
for (i in 0:100){
  for (j in 0:100){
    weigh[count, 1] = i/100
    weigh[count, 2] = j/100
    weigh[count, 3] = 1 - i/100 - j/100
    count = count + 1
  }
}

# min VaR - multi-Normal
n = length(weigh[,1])
VaR_v = rep(0, n)
MU = as.vector(colMeans(Mid))
COV = as.matrix(cov(Mid))
for (i in 1:n){
  w_i = weigh[i,]
  P_mean = t(weigh[i,]) %*% MU
  P_sd = sqrt( t(weigh[i,]) %*% COV %*% weigh[i,] )
  VaR_v[i] = -qnorm(.002,P_mean,P_sd)
}

weigh[which.min(VaR_v),]
```

```
## [1] 0.77 0.03 0.20
```

```r
VaR_v[which.min(VaR_v)]
```

```
## [1] 0.05847047
```

(c) Based on the estimated multivariate model in 2-(c), derive the optimal portfolio between the 3 stocks that minimizes volatility.

Optimal portfolio for these 3 stocks are weighted as 0.765709430, 0.004300497 and 0.229990073, and then the minimal vol = 0.01564688

```r
fit = cov.trob(Mid, nu=4.64)
COV = fit$cov
e = c(1,1,1)
portfolio_n = ( ginv(COV) %*% e ) / as.double(t(e) %*% ginv(COV) %*% e )
portfolio_n
```

```
##             [,1]
## [1,] 0.765709430
## [2,] 0.004300497
## [3,] 0.229990073
```

```r
sqrt(t(portfolio_n) %*% COV %*% portfolio_n)
```

```
##            [,1]
## [1,] 0.01564688
```

(d) Based on the estimated multivariate model in 2-(c), derive the optimal portfolio between the 3 stocks that minimizes relative VaR at q = .002.

Best weights are 0.77 0.00 0.23 for each stock, and then the minimal VaR = 0.08121188 * Price

```r
fit = cov.trob(Mid, nu=4.64)
MU = as.vector(fit$center)
COV = as.matrix(fit$cov)

for (i in 1:n){
  w_i = weigh[i,]
  P_mean = t(weigh[i,]) %*% MU
  P_scale = sqrt( t(weigh[i,]) %*% COV %*% weigh[i,] )
  VaR_v[i] = -( P_scale*qt(.002, 4.64) + P_mean )
}

weigh[which.min(VaR_v),]
```

```
## [1] 0.77 0.00 0.23
```

```r
VaR_v[which.min(VaR_v)]
```

```
## [1] 0.08121188
```