https://www.overleaf.com/project/6209c9b9b7162aea3a1574e1

SOLUTIONS TO STATS 509 PROBLEM SET 5

1.

(a).

$$\mathbb{E}(0.6R_1 + 0.4R_2) = 0.6\mathbb{E}(R_1) + 0.4\mathbb{E}(R_2) = 0.034$$

$$\begin{aligned}&\mathrm{Var}(0.6R_1 + 0.4R_2)\\ =&0.6^2\mathrm{Var}(R_1) + 0.4^2\mathrm{Var}(R_2) + 2 \cdot 0.6 \cdot 0.4\mathrm{Cov}(R_1, R_2)\\ =&0.6^2\mathrm{Var}(R_1) + 0.4^2\mathrm{Var}(R_2) + 2 \cdot 0.6 \cdot 0.4\mathrm{Corr}(R_1, R_2)\sqrt{\mathrm{Var}(R_1)\mathrm{Var}(R_2)}\\ =&0.001728\end{aligned}$$

(b). Let $R_w = w \cdot R_1 + (1-w) \cdot R_2$,

$$\begin{aligned}\mathrm{Var}\left(R_w\right) &= w^2 \cdot \mathrm{Var}\left(R_1\right) + (1-w)^2 \cdot \mathrm{Var}\left(R_2\right) + 2w(1-w)COV\left(R_1, R_2\right)\\ &= w^2 \cdot 0.0016 + (1-w)^2 \cdot 0.0036 + 2 \cdot 0.5 \cdot 0.04 \cdot 0.06 \cdot \left(w - 2w^2\right)\end{aligned}$$

Note that it's convex in 2. Let $\frac{d\,\mathrm{Var}(R_w)}{dw} = 0$, then

$$w = \frac{\sigma_2^2 - \rho\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2} = \frac{6}{7}$$

When w equals to 6/7, the variance of the portfolio, i.e., the risk is minimized.

(c). For $\omega = 0, .01, .02, \ldots, 1.0$, compute:

$$\begin{aligned}E\left[R_\omega\right] &= \omega(.03) + (1-\omega)(.04)\\ \mathrm{Var}\left[R_\omega\right] &= \omega^2(.04)^2 + (1-\omega)^2(.06)^2 + 2\omega(1-\omega)(0.0012)\end{aligned}$$

Then compute VaR_ω as the qth quantile of normal distribution with mean $E[R_w]$ and variance $\mathrm{Var}[R_w]$ and pick ω that minimizes VaR_ω.

```
> # 1c
> VaR.norm <- function(w, q=0.005, eR1=0.03, eR2=0.04,
+                          vR1=0.04^2, vR2=0.06^2, cR1R2=0.5){
+   e <- w*eR1 + (1-w)*eR2
+   v <- w^2*vR1 + (1-w)^2*vR2 + 2*w*(1-w)*cR1R2*sqrt(vR1)*sqrt(vR2)
+   ret <- -qnorm(q, mean=e, sd=sqrt(v))
+   return(ret)
+ }
> w_vec <- seq(from=0, to=1, by=0.01)
> (which.min(sapply(w_vec,FUN=VaR.norm)) - 1)/100
[1] 0.8
```

So $w = 0.8$ minimizes VaR at $q = .005$.

(d). For R_1 and R_2 follow multi-variate t-distribution,

$$\begin{aligned}E\left[R_\omega\right] &= \mu_1 = \omega(.03) + (1-\omega)(.04)\\ \mathrm{Var}[R_w] &= \lambda_w^2\frac{v}{v-2} = \omega^2(.04)^2 + (1-\omega)^2(.06)^2 + 2\omega(1-\omega)(0.0012)\end{aligned}$$

Given $v = 6$,

$$\lambda_\omega^2 = \frac{2}{3} \times \left[\omega^2(.04)^2 + (1-\omega)^2(.06)^2 + 2\omega(1-\omega)(.04)(.06)(0.0012)\right]$$

Do the same computing as in (c) and we have

```
> # 1d
> VaR.t <- function(w, q=0.005, eR1=0.03, eR2=0.04,
+                     vR1=0.04^2, vR2=0.06^2, cR1R2=0.5){
+   e <- w*eR1 + (1-w)*eR2
+   v <- 2 * (w^2*vR1 + (1-w)^2*vR2 + 2*w*(1-w)*cR1R2*sqrt(vR1)*sqrt(vR2))/3
+   ret <- -(qt(q, df=6)*sqrt(v) + e)
+   return(ret)
+ }
> w_vec <- seq(from=0, to=1, by=0.01)
> (which.min(sapply(w_vec,FUN=VaR.t)) - 1)/100
[1] 0.81
```

$w = 0.81$ is the minimizer now.

2.

(a). Note that

$$F_X(x) = \mathbb{P}(X < x) = \mathbb{P}(e^X < e^x) = \mathbb{P}(Y < e^x) = F_Y(e^x) = F_Y(y),$$

then

$$\begin{aligned} C(u_1, u_2) &= \mathbb{P}\left(F_X(X) \le u_1, F_Y(Y) \le u_2\right) \\ &= \mathbb{P}\left(F_X(X) \le u_1, F_X(X) \le u_2\right) \\ &= \mathbb{P}(F_X(X) \le \min(u_1, u_2)) = \min(u_1, u_2). \end{aligned}$$

(b). Note that

$$F_Y(y) = P((-X)^3 \le y) = P(X \ge -y^{1/3}) = 1 - F_X(-y^{1/3})$$

and thus

$$F_Y(Y) = 1 - F_X(-((-X)^3)^{1/3}) = 1 - F_X(X)$$

So,

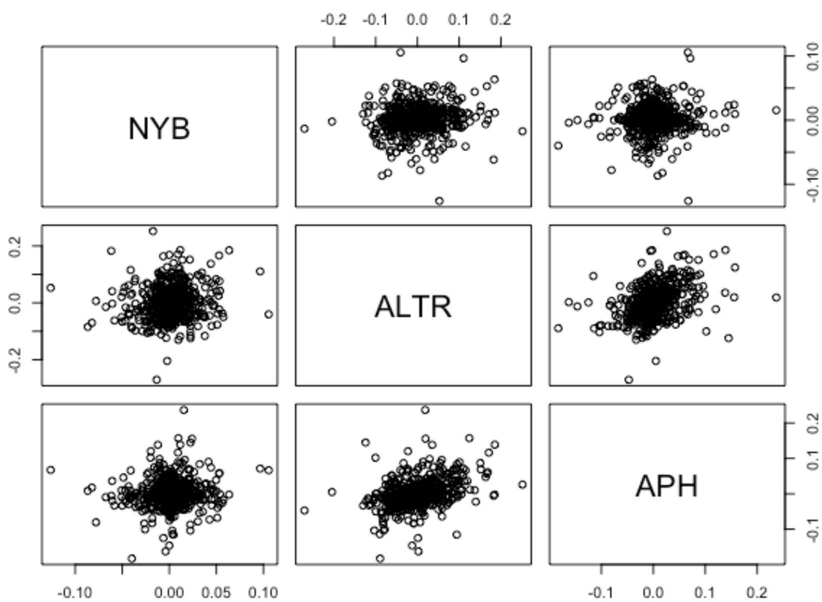
$$\begin{aligned} C(u_1, u_2) &= \mathbb{P}\left(F_X(X) \le u_1, F_Y(Y) \le u_2\right) \quad = \mathbb{P}\left(F_X(X) \le u_1, 1 - F_X(X) \le u_2\right) \\ &= \mathbb{P}\left(1 - u_2 \le F_X(X) \le u_1\right) \\ &= \max(u_1 + u_2 - 1, 0). \end{aligned}$$

3 (With pure codes).

(a).

```
> # 3a
> # load data
> Data = read.csv("midcapD.csv",header=TRUE)
> Mid_Returns = Data[,c(5,6,7)]
> # skewness and kurtosis
> library(moments)
> skewness(Mid_Returns)
       NYB       ALTR        APH
```

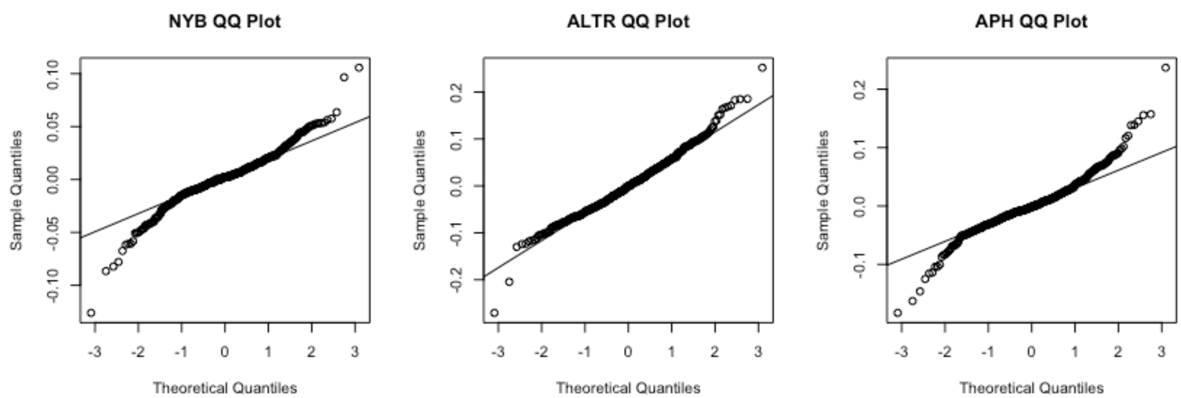
```
-0.3492098  0.2747100  0.4152106
> kurtosis(Mid_Returns)
     NYB      ALTR      APH
6.687914 4.336861 7.068367
> # correlation and scatter
> round(cor(Mid_Returns),digits=3)
       NYB  ALTR   APH
NYB  1.000 0.116 0.075
ALTR 0.116 1.000 0.381
APH  0.075 0.381 1.000
> round(cor(Mid_Returns,method="spearman"),digits=3)
       NYB  ALTR   APH
NYB  1.000 0.127 0.038
ALTR 0.127 1.000 0.423
APH  0.038 0.423 1.000
> pairs(Mid_Returns)
```



(b).

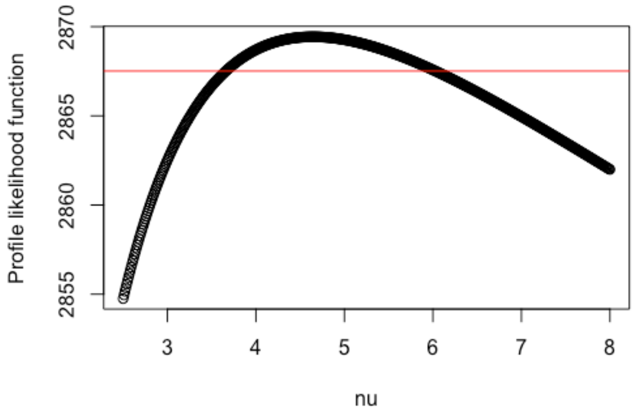
```
> # 3b
> # estimated mu
> mu <- sapply(Mid_Returns, FUN=mean)
> mu
        NYB        ALTR         APH
0.001686861 0.001488180 0.001612351
> # estimated Sigma
> Sigma <- cov(Mid_Returns)
> round(Sigma,5)
```

```
        NYB     ALTR      APH
NYB  0.00054 0.00016 0.00007
ALTR 0.00016 0.00362 0.00097
APH  0.00007 0.00097 0.00177
> # plots
> par(mfrow=c(1,3))
> qqnorm(Mid_Returns$NYB,main="NYB QQ Plot")
> qqline(Mid_Returns$NYB)
> qqnorm(Mid_Returns$ALTR,main="ALTR QQ Plot")
> qqline(Mid_Returns$ALTR)
> qqnorm(Mid_Returns$APH,main="APH QQ Plot")
> qqline(Mid_Returns$APH)
```

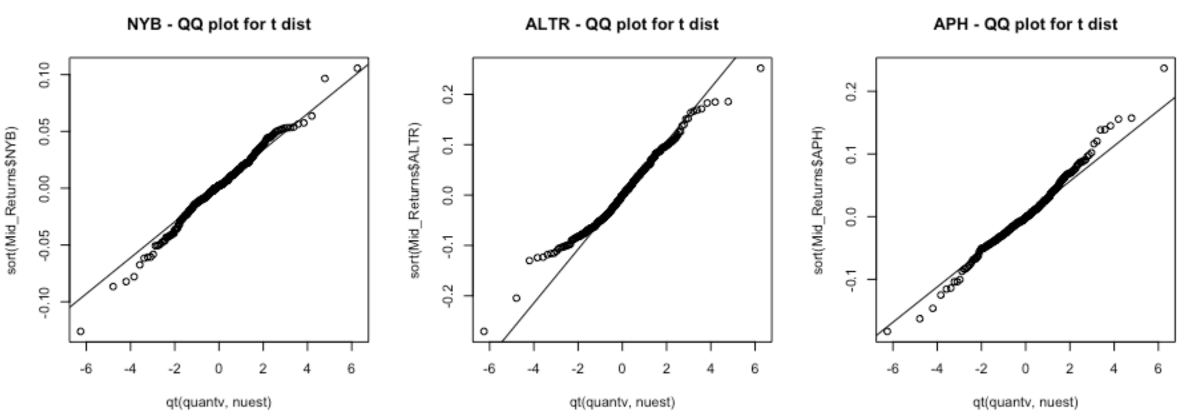


(c).

```
> # 3c
> library(mnormt)
> library(MASS)
> #find MLE
> df = seq(2.5,8,.01)
> n = length(df)
> loglik_max = rep(0,n)
> for (i in 1:n){
+   fit = cov.trob(Mid_Returns,nu=df[i])
+   m = as.vector(fit$center)
+   sigma = matrix(fit$cov,nrow=3)
+   loglik_max[i] = sum(log(dmt(Mid_Returns,mean=m,S=sigma,df=df[i])))
+ }
> threshold <- max(loglik_max) - qchisq(0.95,df=1) /2
> par(mfrow=c(1,1))
> plot(df,loglik_max,xlab= "nu" ,ylab="Profile likelihood function")
> abline(h=threshold, col="red")
```




```
> # nu hat
> df[which.max(loglik_max)]
[1] 4.64
> # nu 95% CI
> c(min(df[loglik_max>threshold]), max(df[loglik_max>threshold]))
[1] 3.67 6.023
> # t-plots
> N <- length(Mid_Returns$NYB)
> nuest <- df[which.max(loglik_max)]
> quantv <- (1/N)*seq(0.5,N-0.5,1)
> par(mfrow=c(1,3))
> qqplot(qt(quantv,nuest), sort(Mid_Returns$NYB), main="NYB - QQ plot for t dist")
> abline(lm(quantile(Mid_Returns$NYB,c(.25,.75))~qt(c(.25,.75),nuest)))
> qqplot(qt(quantv,nuest), sort(Mid_Returns$ALTR), main="ALTR - QQ plot for t dist")
> abline(lm(quantile(Mid_Returns$ALTR,c(.25,.75))~qt(c(.25,.75),nuest)))
> qqplot(qt(quantv,nuest), sort(Mid_Returns$APH), main="APH - QQ plot for t dist")
> abline(lm(quantile(Mid_Returns$APH,c(.25,.75))~qt(c(.25,.75),nuest)))
```



(d).

```
> #AIC of t-dist model
> 2*10 - 2*max(loglik_max)
[1] -5718.868
> #AIC of normal model
> 2*9 - 2*sum(mvtnorm::dmvnorm(Mid_Returns,mean=mu,sigma=Sigma, log=TRUE))
[1] -5553.348
```

The t distributed model has lower AIC, thus preferable based on the AIC criterion.

3. (A detailed solution with discussions).

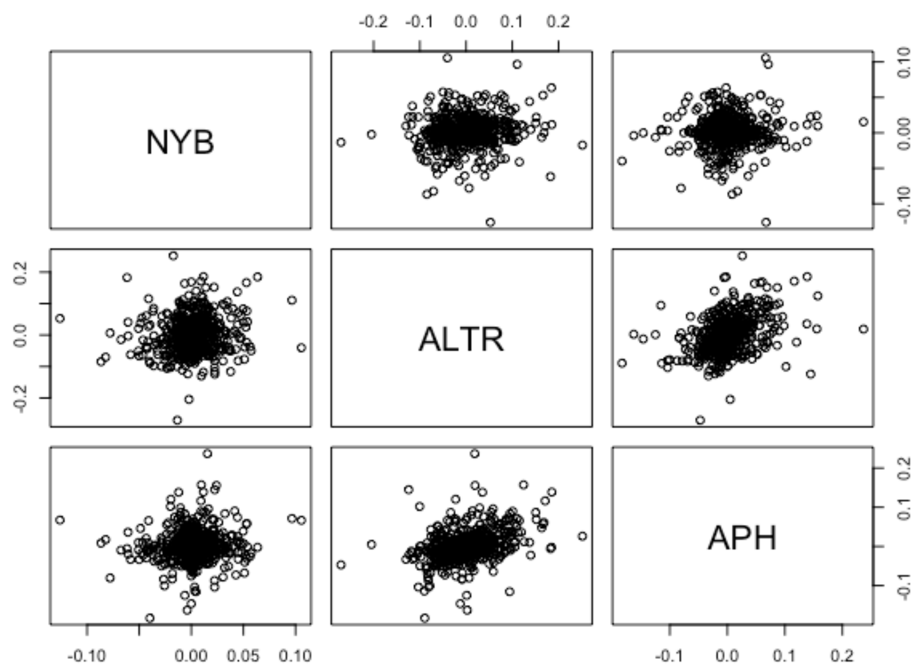
(a). Below is the table of the of the skewness and kurtosis, and below that is the pearson correlation matrix and the spearman correlation matrix. The skewness values are not too large, ranging from -.35 for NYB to .42 for APH. The kurtosis values show that it appears that NYB and APH have relatively heavier tails with 3.66 and 4.04, respectively, and ALTR has significantly less heavy tails, having a kurtosis of 1.32.

Stocks	Skewness	Kurtosis
NYB	-3.48	3.66
ALTR	.27	1.32
APH	.41	4.04

The correlation matrices, both Pearson and Spearman are given below. We see that all of the correlations are positive, but they are fairly small with the exception of ALTR and APH having a Spearman correlation of .42.

$$\text{Pearson Correlation} = \begin{bmatrix} 1.000 & 0.116 & 0.076 \\ 0.116 & 1.000 & 0.381 \\ 0.075 & 0.381 & 1.000 \end{bmatrix}, \quad \text{Spearman Correlation} = \begin{bmatrix} 1.000 & 0.127 & 0.038 \\ 0.127 & 1.000 & 0.423 \\ 0.381 & 0.423 & 1.000 \end{bmatrix}$$

Finally below are the scatter diagrams, and we see the the non-correlatedness of the pairs of the stock NYB with the other two, and we see the slight correlation of .ALTR with APH. Also, see the outliers in the scatter diagram indicating heavier tails and supporting the the multivariate t will be a better fit than the multivariate normal.



R-code

```
> Data = read.csv("..\\Data\\midcapD.csv",header=TRUE)
> Mid_Returns = Data[,c(5,6,7)]
> library(fExtremes)
> skewness(Mid_Returns)
       NYB        ALTR        APH
-0.3481627  0.2738863  0.4139656
> kurtosis(Mid_Returns)
     NYB     ALTR      APH
3.661189 1.319531 4.040122
> pairs(Mid_Returns)

> cor(Mid_Returns)
           NYB       ALTR       APH
NYB  1.0000000 0.1155701 0.0754963
ALTR 0.1155701 1.0000000 0.3814662
APH  0.0754963 0.3814662 1.0000000

> cor(Mid_Returns,method='spearman')
            NYB       ALTR        APH
NYB  1.00000000 0.1267679 0.03818014
ALTR 0.12676793 1.0000000 0.42325910
APH  0.03818014 0.4232591 1.00000000
```

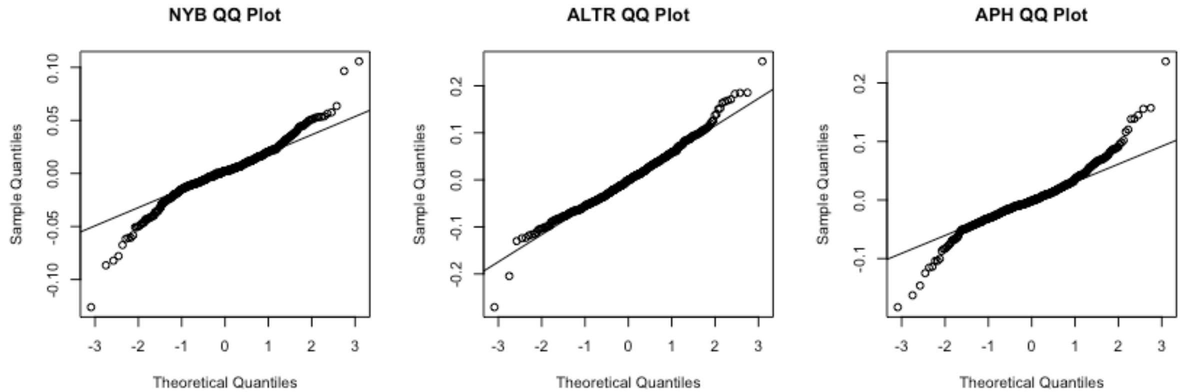
(b). The MLE estimates are simply the sample mean and sample covariance matrix, and are calculated as

$$\hat{\mu} = [0.0017 \;\; 0.0015 \;\; 0.0016]$$

and

$$\hat{\Sigma} = \begin{bmatrix} .0005406 & 0.000161 & .000739 \\ .0001617 & 0.003623 & .000966 \\ .0000739 & 0.000966 & .001772 \end{bmatrix}$$

The QQ-plots are shown below – they clearly show the heavy-tailedness of the NYB and APH returns, and less heavy-tailed ness of ALTR relative to the normal distribution. So this just confirms the kurtosis values in part **(a)**.



R-code

```
> mu_est = colMeans(Mid_Returns)
> Sigma_est = cov(Mid_Returns)
> mu_est
        NYB        ALTR         APH
0.001686861 0.001488180 0.001612351
> Sigma_est
              NYB         ALTR          APH
NYB  5.406949e-04 0.0001617717 7.390143e-05
ALTR 1.617717e-04 0.0036237794 9.666918e-04
APH  7.390143e-05 0.0009666918 1.772157e-03
>
> qqnorm(Mid_Returns[,1],main='NYB Normal QQ Plot')
> qqline(Mid_Returns[,1])
>
> qqnorm(Mid_Returns[,2],main='ALTR Normal QQ Plot')
> qqline(Mid_Returns[,2])
>
>
> qqnorm(Mid_Returns[,3],main='APH Normal QQ Plot')
> qqline(Mid_Returns[,3])
```

(c). Carried out a profile-likelihood estimation of the multivariate t-distribution, with the result of the final estimates being

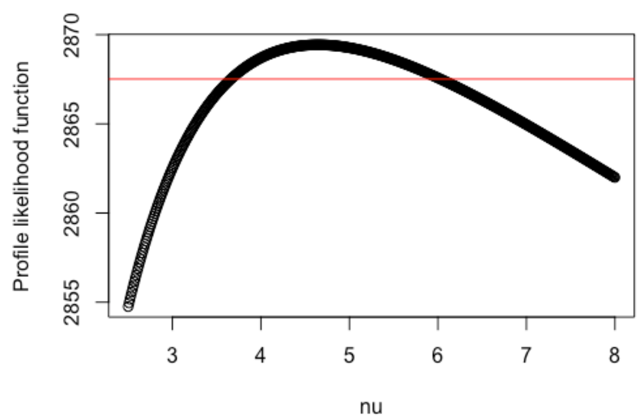
$$\hat{\nu} = 4.64$$

$$\hat{\mu} = [0.00218 \quad -0.00045 \quad -0.00024]$$

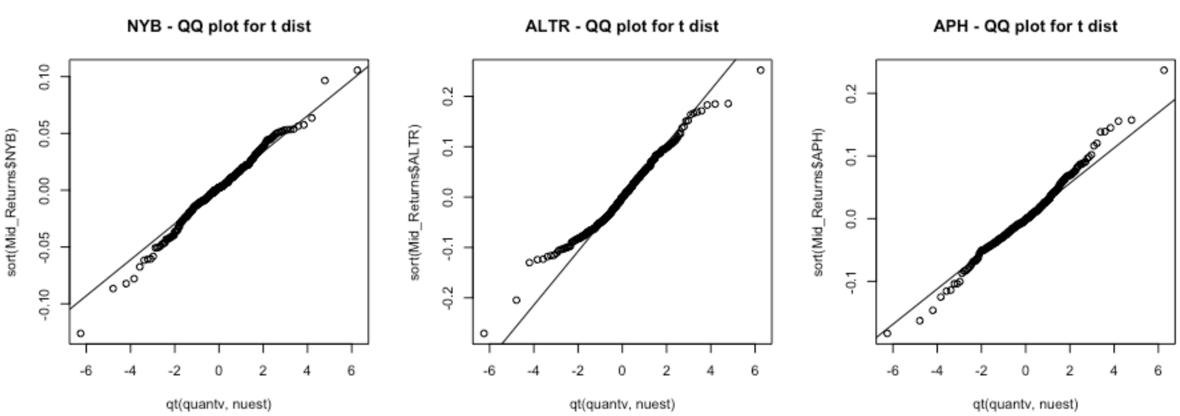
$$\hat{\Lambda} = \begin{bmatrix} .000312 & 0.000111 & .000243 \\ .000110 & 0.002450 & .000653 \\ .000024 & 0.000653 & .000972 \end{bmatrix}$$

The profile likelihood function is shown below, and utilizing the confidence interval method as presented in class, we get the 95% confidence interval for the degrees of freedom is [3.67, 6.02]. Below are the univariate QQ plots for the t-distribution with the estimated degrees of freedom. are shown below. As can be seen, it seems that true tails of the NYB and APH, especially are heavier tailed than the theoretical t-distribution models estimated from the multivariate t-distribution, and the tails of the ALTR are lighter tailed than the estimated t-distribution. This is not surprising based on the computed kurtosis values. In addition, we seem to see some assymetry of the empirical data, as the tails of the NYB appear to heavier on the lower (left) tail as compared to the upper (right) tail.

Finally, we did compute the AIC criteria. For multivariate t-distribution there a $p = 10$ paramters (3 mean parameters, 6 scale matrix parameters, and 1 degree of freedom) as compared to $p = 9$ for the multivariate normal. Utilizing the log-likelihood function being the sum of the log-likelihoods, compute the AIC criteria of $-2 * log - likelihood + 2 * p$ and found a value of -5718.78 for the multivariate t-distribution (with a total of 10 parameters) and -5558.99 for the multivariate normal distribution (with a total of 9 parameters). To compute the multivariate normal, just utilized the dmt function in R with a very large number of degrees of freedom (500) and utilizing the covariance matrix as the scale matrix. So it is clear that by this criteria, as well by the QQ-plots, that the the multivariate t-distribution is the preferred model over the multivariate normal distribution.



Profile likelihood function



R-code

```
> library(mnormt) # needed for dmt
> library(MASS) # needed for cov.trob
> df = seq(2.5,8,.01)
> n = length(df)
> loglik_max = rep(0,n)
> for(i in 1:n){
+   fit = cov.trob(Mid_Returns,nu=df[i])
+   mu = as.vector(fit$center)
+   Lambda = matrix(fit$cov,nrow=3)
+   loglik_max[i] = sum(log(dmt(Mid_Returns,mean=fit$center,S=Lambda,df=df[i])))
+ }
> plot(df,loglik_max,xlab='nu',ylab='Profile-likelihood function')
>
> nu_est = df[which.max(loglik_max)]
> nu_est
[1] 4.64
> t = .5*qchisq(.95,1)
> nu_lconf = min(df[loglik_max>max(loglik_max)-t])
> nu_uconf = max(df[loglik_max>max(loglik_max)-t])
> show(c(nu_lconf,nu_uconf))
[1] 3.67 6.02
> fitfinal = cov.trob(Mid_Returns,nu=nu_est)
> fitfinal$center
          NYB          ALTR           APH
 0.0021833203 -0.0004536351 -0.0002412612
> fitfinal$cov
              NYB         ALTR          APH
NYB  3.118324e-04 0.0001098744 2.425963e-05
ALTR 1.098744e-04 0.0024496798 6.528902e-04
APH  2.425963e-05 0.0006528902 9.715263e-04
>
> N = length(Mid_Returns[,1])
> quantv = (1/(N+1))*c(1:N)
> qqplot(sort(Mid_Returns[,1]),qt(quantv,nu_est),main='NYB - QQ plot for t-dist')
> abline(lm(qt(c(.25,.75),nu_est)~quantile(Mid_Returns[,1],c(.25,.75))))
>
> qqplot(sort(Mid_Returns[,2]),qt(quantv,nu_est),main='ALTR - QQ plot for t-dist')
> abline(lm(qt(c(.25,.75),nu_est)~quantile(Mid_Returns[,2],c(.25,.75))))
>
> qqplot(sort(Mid_Returns[,3]),qt(quantv,nu_est),main='APH - QQ plot for t-dist')
> abline(lm(qt(c(.25,.75),nu_est)~quantile(Mid_Returns[,3],c(.25,.75))))

> AIC_t = -2*sum(log(dmt(Mid_Returns,mean=fit$center,S=fitfinal$cov,df=nu_est)))+2*(3+6+1)
> AIC_norm = -2*sum(log(dmt(Mid_Returns,mean=mu_est,S=Sigma_est,df=500)))+2*(3+6)
> show(c(AIC_t,AIC_norm))
[1] -5718.777 -5558.994
```