CFRM 541 Homework 3

Greg Damico

October 28, 2016

Problem #1

```r
library(mpo)
```

```
## Loading required package: boot

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

## Loading required package: mvtnorm

## Loading required package: rrcov

## Loading required package: robustbase

##
## Attaching package: 'robustbase'

## The following object is masked from 'package:boot':
##
##     salinity

## Scalable Robust Estimators with High Breakdown Point (version 1.4-3)

## Loading required package: robust
```

```
## Loading required package: fit.models

## Loading required package: MASS

## Loading required package: PortfolioAnalytics

## Loading required package: foreach

## Loading required package: PerformanceAnalytics

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

## Loading required package: corpcor

## Loading required package: shiny

##
## Attaching package: 'mpo'

## The following object is masked from 'package:shiny':
##
##     runExample

## The following objects are masked from 'package:PerformanceAnalytics':
##
##     SFM.beta, StdDev.annualized
```

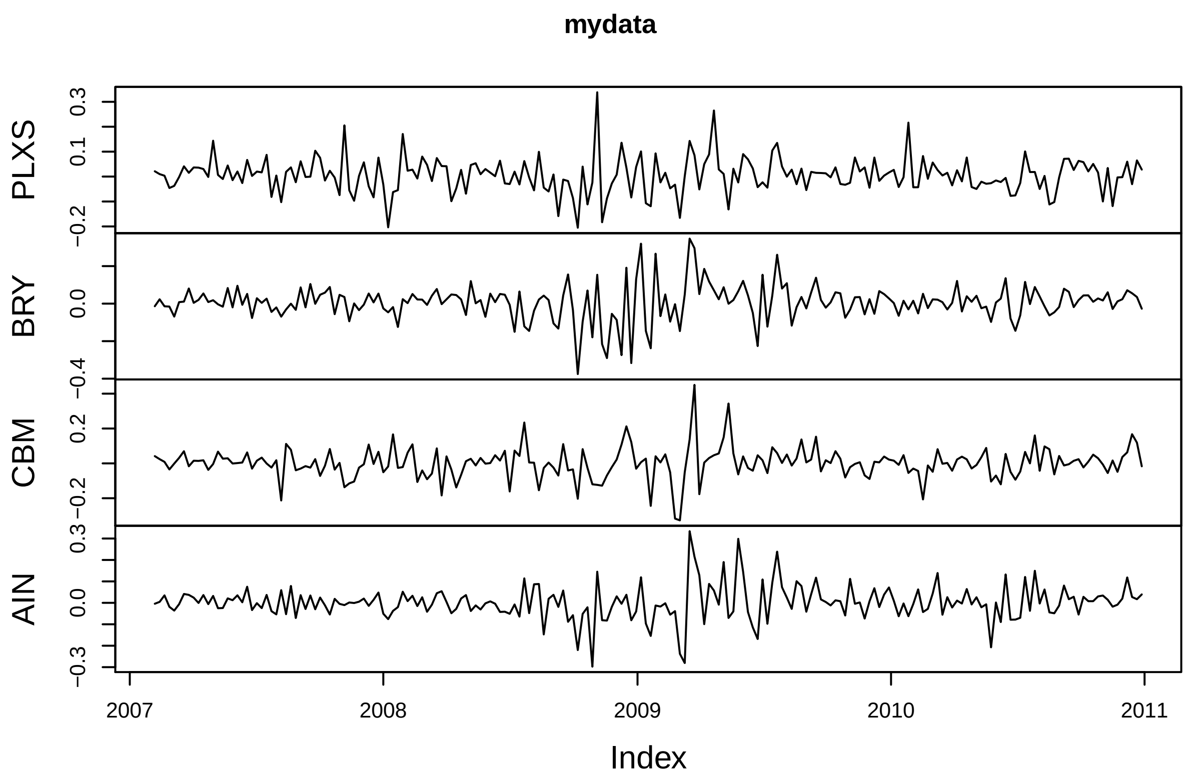
```r
library(sn)
```

```
## Loading required package: stats4

##
## Attaching package: 'sn'

## The following object is masked from 'package:stats':
##
##     sd
```

```r
library(quantmod)
```

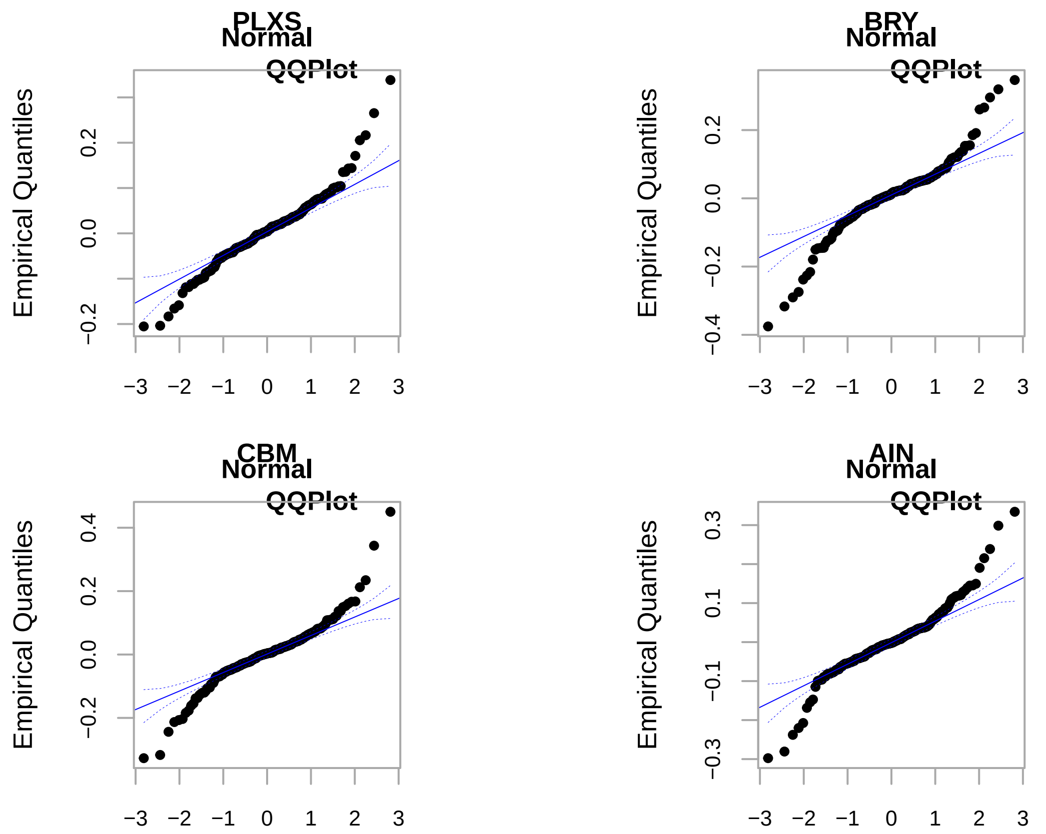
```
## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(TTR)
library(tseries)
library(nor1mix)

# There seems to be something funny going on here. I'm trying to call
# "chart.QQPlot.norMix" from the mpo package, but when I ran the code below I would get an
# error telling me that the compiler could not find the "chart.QQPlot.norMixEM" function.
# A student on Piazza helpfully pointed out that this other function was in the nor1mix
# package. Hence the call above to load the nor1mix package. But it still seems to me that
# what I really want is not the EM version. Why the code below has been being taken to call
# this other version remains mysterious to me.

retSW4 <- smallcapW["2007-01-31/"]
tickers <- c("PLXS", "BRY", "CBM", "AIN")
mydata <- retSW4[, tickers]
plot.zoo(mydata)
```




```r
# It does seem as though these returns are fairly non-normal.  The value of the returns
# seems generally unpredictable.

nrow = 2
ncol = 2
par(mfrow = c(nrow, ncol))
par(pty = "s")
par(mar = c(3.1, 4.1, 2.1, 1.1))
n = 4
for (i in 1:n) {
  chart.QQPlot(mydata[, i], xlab = "Normal Quantiles", main = c(tickers[i], "Normal
```

```
              QQPlot"), envelope = 0.95, pch = 20, line = c("quartiles"), lwd = 0.5)
}
```




```
# The normal qqplot leaves a lot to be desired here.  There are many outliers outside even
# the 0.95 (~ 2 sd) envelope.

nrow = 2
ncol = 2
par(mfrow = c(nrow, ncol))
par(pty = "s")
par(mar = c(3.1, 4.1, 2.1, 1.1))
n = 4
for (i in 1:n) {
  main = c(tickers[i], "NorMix-2 QQPlot")
  chart.QQPlot.norMix(mydata[, i], xlab = "Two-Component NorMix Quantiles", main = main,
                      na.rm = TRUE, envelope = 0.95, pch = ".", ylab = "", line =
        c("quartiles"), para = list(m = 2), distribution = "mixnormal", lwd = 0.5)
}
```

```
## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)
```

```
## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##           mu  sigma     w
## [1,] 0.00442 0.0476 0.766
```

```
## [2,] 0.00880 0.1207 0.234
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"
```

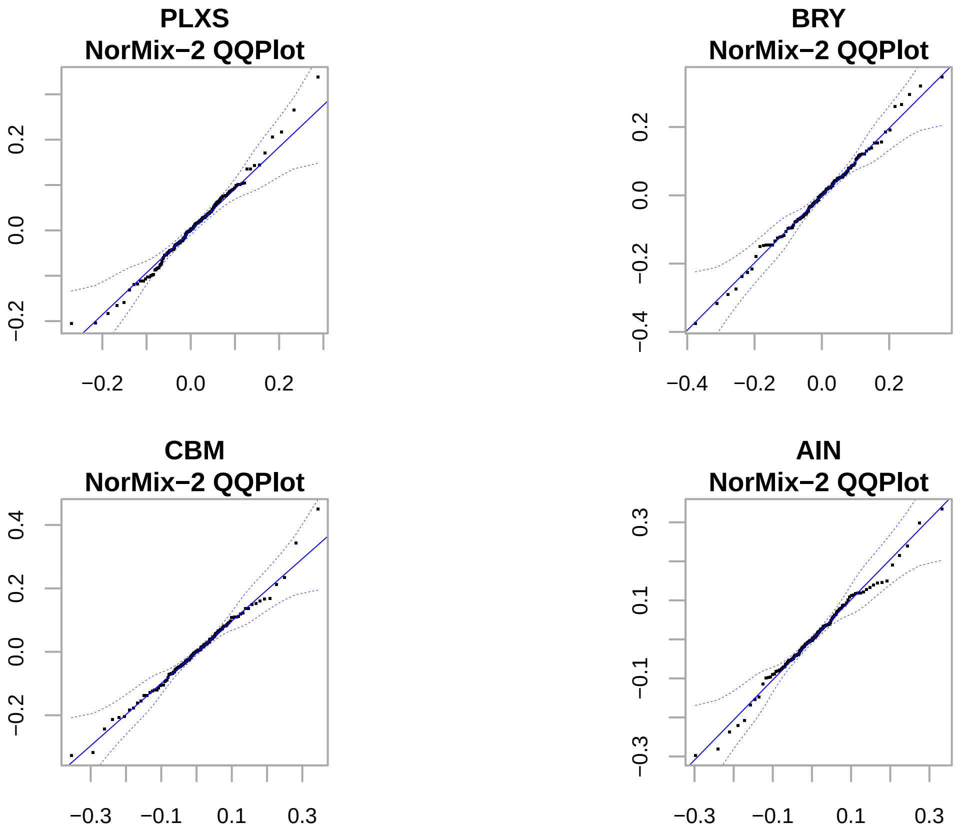
```
## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)
```

```
## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)
```

```
## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##            mu  sigma     w
## [1,] -0.00948 0.1495 0.344
## [2,]  0.01514 0.0445 0.656
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"
```

```
## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##            mu  sigma     w
## [1,]  0.00464 0.0444 0.679
## [2,] -0.00533 0.1441 0.321
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"
```

```
## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##           mu  sigma     w
## [1,] -0.0038 0.0415 0.709
## [2,]  0.0172 0.1317 0.291
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"
```

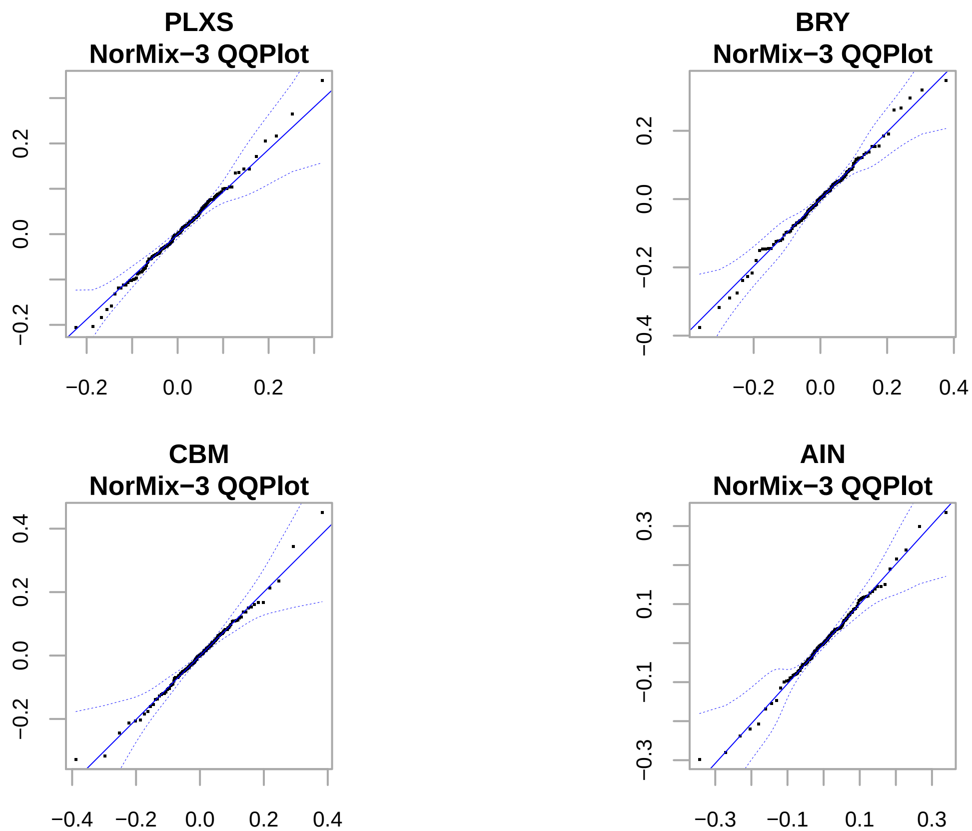



```
# This normal-mixture plot seems to do much better.  Now most of the values are inside the
# envelope.

nrow = 2
ncol = 2
par(mfrow = c(nrow, ncol))
par(pty = "s")
par(mar = c(3.1, 4.1, 2.1, 1.1))
n = 4
for (i in 1:n) {
  main = c(tickers[i], "NorMix-3 QQPlot")
  chart.QQPlot.norMix(mydata[, i], xlab = "Three-Component NorMix Quantiles", main = main,
                      na.rm = TRUE, envelope = 0.95, pch = ".", ylab = "", line =
        c("quartiles"), para = list(m = 3), distribution = "mixnormal", lwd = 0.5)
}
```

```
## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##            mu  sigma     w
## [1,] -0.0229 0.0742 0.337
## [2,]  0.0116 0.0407 0.557
## [3,]  0.0633 0.1277 0.106
## attr(,"class")
## [1] "fitEM" "nMfit"
```

```
## [1] "using fitted model as theoretical distribution"

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##           mu  sigma     w
## [1,] -0.0577 0.1290 0.167
## [2,]  0.0159 0.0448 0.663
## [3,]  0.0341 0.1557 0.170
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##             mu  sigma     w
## [1,] -0.000863 0.0880 0.387
## [2,]  0.004349 0.0371 0.490
## [3,] -0.002841 0.1874 0.124
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## [1] "fitted model:"
## 'Normal Mixture' object   ``norMixEM(x, m = para$m)[1:para$m,]''
##           mu  sigma     w
```

```
## [1,]  0.0462 0.0587 0.212
## [2,] -0.0115 0.0381 0.614
## [3,] -0.0025 0.1554 0.174
## attr(,"class")
## [1] "fitEM" "nMfit"
## [1] "using fitted model as theoretical distribution"

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)

## Warning in norMixEM(x, para$m, trace = 0): EM did not converge in 100
## iterations (with 'tol'=1.49012e-08)
```




```
# I don't see a tremendous improvement here.
```

Problem #2

Consider a fully-invested portfolio with single period return $r_P = w_1 r_1 + w_2 r_2$ where returns r_1, r_2 have means μ_1, μ_2, volatilities σ_1, σ_2, and correlation coefficient ρ. Let μ_P and σ_P be the portfolio mean return and volatility.

a. For the case $\rho = 1$ derive the straight-line relationship between μ_P and σ_P.

Now:

$$
\begin{aligned}
\mu_P &= E[r_P] && (1)\\
&= E[w_1 r_1 + w_2 r_2] && (2)\\
&= E[w_1 r_1] + E[w_2 r_2] && (3)\\
&= w_1\mu_1 + w_2\mu_2. && (4)
\end{aligned}
$$

Also:

$$
\begin{aligned}
\sigma_P^2 &= E[(r_P - \mu_P)^2] && (5)\\
&= E[(w_1(r_1 - \mu_1) + w_2(r_2 - \mu_2))^2] && (6)\\
&= w^T \Sigma w && (7)\\
& && (8)
\end{aligned}
$$

for

$$
\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}
$$

Thus $\sigma_P^2 = w_1^2\sigma_1^2 + 2w_1w_2\rho\sigma_1\sigma_2 + w_2^2\sigma_2^2$.

Since the portfolio is fully invested, we know that $w_1 + w_2 = 1$.

For $\rho = 1$, $\sigma_P^2 = (w_1\sigma_1 + w_2\sigma_2)^2$.

Thus $\mu_P = w_1\mu_1 + w_2\mu_2 = w_1\mu_1 + (1 - w_1)\mu_2$ and $\sigma_P = w_1\sigma_1 + w_2\sigma_2 = w_1\sigma_1 + (1 - w_1)\sigma_2$.

Solving for w_1 from the σ- equation, we have:

$$
\begin{aligned}
w_1(\sigma_1 - \sigma_2) &= \sigma_P - \sigma_2 && (9)\\
w_1 &= \frac{\sigma_P - \sigma_2}{\sigma_1 - \sigma_2}. && (10)
\end{aligned}
$$

Plugging this into the μ- equation, we have:

$$
\begin{aligned}
\mu_P &= \frac{\sigma_P - \sigma_2}{\sigma_1 - \sigma_2}\mu_1 + \frac{\sigma_1 - \sigma_2 - \sigma_P + \sigma_2}{\sigma_1 - \sigma_2}\mu_2 && (11)\\
&= \frac{\mu_1 - \mu_2}{\sigma_1 - \sigma_2}\sigma_P + \frac{\mu_2\sigma_1 - \mu_1\sigma_2}{\sigma_1 - \sigma_2}. && (12)
\end{aligned}
$$

Thus there is a linear relationship between μ_P and σ_P.

b. For the case $\rho = -1$ derive the two straight-line relationships between μ_P and σ_P.

For $\rho = -1$, $\sigma_P^2 = (w_1\sigma_1 - w_2\sigma_2)^2$.

Thus:

$$
\sigma_P = \begin{cases} w_1\sigma_1 - w_2\sigma_2, & w_1\sigma_1 \geq w_2\sigma_2 \\ w_2\sigma_2 - w_1\sigma_1, & w_1\sigma_1 < w_2\sigma_2 \end{cases}
$$

Suppose $w_1\sigma_1 \geq w_2\sigma_2$. Then $\sigma_P = w_1\sigma_1 - (1 - w_1)\sigma_2$.

Hence:

$$w_1(\sigma_1 + \sigma_2) = \sigma_P + \sigma_2 \tag{13}$$
$$w_1 = \frac{\sigma_P + \sigma_2}{\sigma_1 + \sigma_2}. \tag{14}$$

Thus:

$$\mu_P = \frac{\sigma_P + \sigma_2}{\sigma_1 + \sigma_2}\mu_1 + \frac{\sigma_1 + \sigma_2 - \sigma_P - \sigma_2}{\sigma_1 + \sigma_2}\mu_2 \tag{15}$$
$$= \frac{\mu_1 + \mu_2}{\sigma_1 + \sigma_2}\sigma_P + \frac{\mu_2\sigma_1 + \mu_1\sigma_2}{\sigma_1 + \sigma_2}. \tag{16}$$

Otherwise: $\sigma_P = -w_1\sigma_1 + (1 - w_1)\sigma_2$.

Hence:

$$w_1(\sigma_1 + \sigma_2) = -\sigma_P + \sigma_2 \tag{17}$$
$$w_1 = \frac{\sigma_2 - \sigma_P}{\sigma_1 + \sigma_2}. \tag{18}$$

Thus:

$$\mu_P = \frac{\sigma_2 - \sigma_P}{\sigma_1 + \sigma_2}\mu_1 + \frac{\sigma_1 + \sigma_2 - \sigma_2 + \sigma_P}{\sigma_1 + \sigma_2}\mu_2 \tag{19}$$
$$= \frac{\mu_2 - \mu_1}{\sigma_1 + \sigma_2}\sigma_P + \frac{\mu_2\sigma_1 + \mu_1\sigma_2}{\sigma_1 + \sigma_2}. \tag{20}$$

Thus there is a linear relationship between μ_P and σ_P, since:

$$\mu_P = \begin{cases} \frac{\mu_1+\mu_2}{\sigma_1+\sigma_2}\sigma_P + \frac{\mu_2\sigma_1+\mu_1\sigma_2}{\sigma_1+\sigma_2}, & w_1\sigma_1 \geq w_2\sigma_2 \\ \frac{\mu_2-\mu_1}{\sigma_1+\sigma_2}\sigma_P + \frac{\mu_2\sigma_1+\mu_1\sigma_2}{\sigma_1+\sigma_2}, & w_1\sigma_1 < w_2\sigma_2 \end{cases}$$

Probem #3

Consider the quadratic utility optimal weights $\mathbf{w} = \lambda^{-\mathbf{1}}\mathbf{\Sigma}^{-\mathbf{1}}\mu$ for the case of two assets with returns means, variances, and correlation μ_1, μ_2, σ_1^2, σ_2^2, and ρ respectively. First derive the expression for the two-by-two inverse covariance matrix Σ^{-1} in terms of μ_1, μ_2, σ_1^2, σ_2^2, and ρ. Then derive the expression for the derivative of w_1 with respect to μ_1 and explain how small changes in μ_1 affect w_1 when ρ is close to 1. Also derive an expression for the derivative of w_2 with respect to μ_1 and explain how changes in μ_1 affect changes in w_2.

Now:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

Thus:

$$\Sigma^{-1} = \frac{1}{\sigma_1^2\sigma_2^2 - \rho\sigma_1^2\sigma_2^2}\begin{bmatrix}\sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2\end{bmatrix} \tag{21}$$

$$= \begin{bmatrix}\frac{1}{\sigma_1^2(1-\rho^2)} & \frac{-\rho}{\sigma_1\sigma_2(1-\rho^2)} \\ \frac{-\rho}{\sigma_1\sigma_2(1-\rho^2)} & \frac{1}{\sigma_2^2(1-\rho^2)}\end{bmatrix} \tag{22}$$

For quadratic utility, we have:

$U_q = w^T\mu - \frac{1}{2}\lambda w^T\Sigma w$. Hence $\mu - \lambda\Sigma w = 0$ and $w = \lambda^{-1}\Sigma^{-1}\mu$.

Thus:

$$\begin{bmatrix}w_1 \\ w_2\end{bmatrix} = \lambda^{-1}\begin{bmatrix}\frac{1}{\sigma_1^2(1-\rho^2)} & \frac{-\rho}{\sigma_1\sigma_2(1-\rho^2)} \\ \frac{-\rho}{\sigma_1\sigma_2(1-\rho^2)} & \frac{1}{\sigma_2^2(1-\rho^2)}\end{bmatrix}\begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix} \tag{23}$$

$$w_1 = \frac{\mu_1}{\lambda\sigma_1^2(1-\rho^2)} - \frac{\mu_2\rho}{\lambda\sigma_1\sigma_2(1-\rho^2)}. \tag{24}$$

Thus:

$\frac{\partial w_1}{\partial \mu_1} = \frac{1}{\lambda\sigma_1^2(1-\rho^2)}$.

Thus when $\rho \approx 1$, i.e. for a highly correlated pair of assets, $\frac{\partial w_1}{\partial \mu_1}$ is huge and so w_1 is incredibly sensitive to small changes in μ_1.

Similarly:

$w_2 = \frac{-\rho\mu_1}{\lambda\sigma_1\sigma_2(1-\rho^2)} + \frac{\mu_2}{\lambda\sigma_2^2(1-\rho^2)}$,

and so $\frac{\partial w_2}{\partial \mu_1} = \frac{-\rho}{\lambda\sigma_1\sigma_2(1-\rho^2)}$.

Again, when $\rho \approx 1$, $|\frac{\partial w_2}{\partial \mu_1}|$ is huge and so w_2 is incredibly sensitive to small changes in μ_1. In this case the differential is negative.

Problem #4

Show that the covariance of a global minimum variance (GMV) portfolio with that of any other frontier portfolio is constant, i.e. does not depend on any portfolio mean returns.

To express the covariance of a GMV portfolio with any other, we write:

$$cov(r_{GMV}, r_P) = E[(r_{GMV} - \mu_{GMV})(r_P - \mu_P)] \tag{25}$$

$$= E[w_{GMV}^T(r-\mu)^2 w_P] \tag{26}$$

$$= w_{GMV}^T\Sigma w_P. \tag{27}$$

Now:

$$w_{GMV}^T = \frac{1}{1^T\Sigma^{-1}1}1^T(\Sigma^{-1})^T \tag{28}$$

$$= \frac{1}{1^T\Sigma^{-1}1}1^T\Sigma^{-1}. \tag{29}$$

Thus:

$$cov(r_{GMV}, r_P) = \frac{1^T\Sigma^{-1}\Sigma w_P}{1^T\Sigma^{-1}1} \quad (30)$$
$$= \frac{1^T w_P}{1^T\Sigma^{-1}1}. \quad (31)$$

Thus the covariance is independent of μ_P.

Problem #5

Complete the steps to derive the expression (3.77) for the QU optimal portfolio weights w_{QU} with risky assets only.

Expression (3.77) tells us that $w_q = w_{GMV} + \gamma_{RA}^{-1}\Sigma^{-1}(\mu - \mu_{GMV} \cdot 1)$. Let us start with an expression for quadratic utility.

$U_q = w^T\mu - \frac{1}{2}\gamma_{RA}w^T\Sigma w + \lambda(1 - w^T 1)$.

Thus, taking derivatives, $\mu - \gamma_{RA}\Sigma w - \lambda \cdot 1 = 0$.

So $w = \gamma_{RA}^{-1}\Sigma^{-1}(\mu - \lambda \cdot 1)$.

But now $1^T w = 1$, so $1 = \gamma_{RA}^{-1}1^T\Sigma^{-1}(\mu - \lambda \cdot 1)$.

Thus $\lambda = -\frac{1-\gamma_{RA}^{-1}1^T\Sigma^{-1}\mu}{\gamma_{RA}^{-1}1^T\Sigma^{-1}1} = \frac{\gamma_{RA}^{-1}1^T\Sigma^{-1}\mu-1}{\gamma_{RA}^{-1}1^T\Sigma^{-1}1}$.

Plugging this equation for λ into the above expression for w, we have:

$$w = \gamma_{RA}^{-1}\Sigma^{-1}(\mu - \frac{\gamma_{RA}^{-1}1^T\Sigma^{-1}\mu - 1}{\gamma_{RA}^{-1}1^T\Sigma^{-1}1} \cdot 1) \quad (32)$$
$$= \frac{1}{\gamma}\Sigma^{-1}(\mu - \frac{1^T\Sigma^{-1}\mu - \gamma}{1^T\Sigma^{-1}1} \cdot 1) \quad (33)$$
$$= \frac{1}{\gamma}\Sigma^{-1}(\mu - \frac{1^T\Sigma^{-1}\mu}{1^T\Sigma^{-1}1} \cdot 1 + \frac{\gamma}{1^T\Sigma^{-1}1} \cdot 1) \quad (34)$$
$$= \frac{1}{\gamma}\Sigma^{-1}(\mu - \mu_{GMV} \cdot 1) + \frac{\gamma\Sigma^{-1}}{\gamma 1^T\Sigma^{-1}1} \cdot 1 \quad (35)$$
$$= w_{GMV} + \frac{1}{\gamma}\Sigma^{-1}(\mu - \mu_{GMV} \cdot 1). \quad (36)$$