CFRM 541 - Quantitative Risk Management - Yoshihiro Shirai
Homework 2

Due: February 9, 2025 - 11:59 pm

Question 1 (10 pts). Let $\boldsymbol{X}$ be an n-dimensional random vector with zero mean and unit variance and an equicorrelation matrix $\boldsymbol{\Sigma}$, i.e. all the off-diagonal elements are equal to a constant $\rho > 0$. This means that $\boldsymbol{\Sigma}$ may be written as $\boldsymbol{\Sigma} = \rho \boldsymbol{J}_n + (1-\rho)\boldsymbol{I}_n$, where $\boldsymbol{J}_n$ is an $n \times n$ matrix of 1's and $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. Note then that $\boldsymbol{\Sigma}$ has the form $\boldsymbol{\Sigma} = \boldsymbol{B\Omega B}^T + \boldsymbol{\Upsilon}$ with $\boldsymbol{B} = \sqrt{\rho}\boldsymbol{1}$, where $\boldsymbol{1}$ is a vector of 1's. Show that there is a 1-factor decomposition $\boldsymbol{X} = \boldsymbol{B}F + \boldsymbol{\varepsilon}$, where

$$\begin{aligned} F &= \frac{\sqrt{\rho}}{1+\rho(n-1)}\sum_{i=1}^n X_i + \sqrt{\frac{1-\rho}{1+\rho(n-1)}}Y \\ \varepsilon_i &= X_i - \sqrt{\rho}F \end{aligned}$$

and Y is any zero-mean, unit-variance random variable that is independent of $\boldsymbol{X}$.

Solution. Observe that $\boldsymbol{B}F$ is a vector whose components are all given by $\sqrt{\rho}F$, and so it is indeed the case that $\boldsymbol{X} = \boldsymbol{B}F + \boldsymbol{\varepsilon}$. Furthermore, by assumption, $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{0}$ and $\mathbb{E}[Y] = 0$, and so $\mathbb{E}[\boldsymbol{\varepsilon}] = \boldsymbol{0}$. It remains to show that $Cov(\varepsilon_i, \varepsilon_j) = 0$ for every i, j. In fact, we obtain

$$\begin{aligned} Cov(\varepsilon_i,\varepsilon_j) &= \mathbb{E}[X_iX_j - (X_i+X_j)\sqrt{\rho}F + \rho F^2] \\ &= \rho - \sqrt{\rho}\,\mathbb{E}[(X_i+X_j)F] + \rho\,\mathbb{E}[F^2] \\ &= \rho - \sqrt{\rho}\frac{\sqrt{\rho}}{1+\rho(n-1)}\sum_{k=1}^n \mathbb{E}[X_k(X_i+X_j) \\ &\quad + \frac{\rho^2 V(\sum_{k=1}^n X_k)}{(1+\rho(n-1))^2} + \frac{\rho(1-\rho)}{1+\rho(n-1)} \\ &= \rho - \frac{\rho}{1+\rho(n-1)}2(1+\rho(n-1)) \\ &\quad + \frac{\rho^2(n+\rho n(n-1)}{(1+\rho(n-1))^2} + \frac{\rho(1-\rho)}{1+\rho(n-1)} \\ &= -\rho + \frac{\rho(n\rho+1-\rho)}{1+\rho(n-1)} \\ &= 0, \end{aligned}$$

as requested.

Question 2 (10 pts). Let $\boldsymbol{A}$ be an n-dimensional equicorrelation matrix, i.e. all the off-diagonal elements are equal to a constant $\rho > 0$. This means that $\boldsymbol{A}$ may be written as $\boldsymbol{A} = \rho\boldsymbol{J}_n + (1-\rho)\boldsymbol{I}_n$, where $\boldsymbol{J}_n$ is an $n \times n$ matrix of 1's and $\boldsymbol{I}_n$ is the $n \times n$ identity matrix.

1. Show that

$$\boldsymbol{e}_1 = (1,...,1)^T, \ \boldsymbol{e}_i = (1,...,0,-1,0,...,0)^T$$

for $i = 2,...,n$ constitute a system of n independent eigenvectors (here, the i-th component of $\boldsymbol{e}_i$ is -1, whereas all the others are zero except for the first one, which is equal to 1).

2. What are the eigenvalues of $\boldsymbol{A}$? How would you interpret these eigenvalues for moderate to large values of ρ? What is the minimum value of ρ for which $\boldsymbol{A}$ remains a correlation matrix?

Solution.

1. A direct computation yields:

$$\boldsymbol{A}\boldsymbol{e}_1 = \begin{bmatrix} 1+\rho(n-1) \\ \vdots \\ 1+\rho(n-1) \end{bmatrix} = (1+\rho(n-1))\boldsymbol{e}_1, \ \boldsymbol{A}\boldsymbol{e}_i = \begin{bmatrix} 1-\rho \\ 0 \\ \vdots \\ 0 \\ \rho-1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = (1-\rho)\boldsymbol{e}_i, \ i = 2, ..., n.$$

2. The eigenvalues are $1+\rho(n-1)$ and $1-\rho$. When ρ is large, the factor $\boldsymbol{B}F$ in the previous exercise is approximately equal to the portfolio with weighgts $1\sigma_j$, since their covariance is exactly $1+\rho(n-1)$. No diversification is possible in this case. When ρ is moderate, the opposite becomes true. If $\boldsymbol{A}$ is a correlation matrix, then it must be positive semidefinite, which implies

$$1+\rho(n-1) \geq 0 \Leftrightarrow \rho \geq -\frac{1}{n-1}$$

3. Not all elements of $\boldsymbol{R}^*$ are necessarily between -1 and 1. On the other hand, step (d) guarantees that this is the case for $\overline{\boldsymbol{R}}$. Furthermore, $\overline{\boldsymbol{R}}$ is necessarily symmetric and positive definite, as its eigenvalues are positive (as long as $\lambda_1 > \varepsilon$) and both $\boldsymbol{R}^*$ and $\boldsymbol{D}$ are symmetric.

Question 3 (10 pts). Let $\boldsymbol{A}$ be again an $n \times n$ equicorrelation matrix, with all off diagonal elements equal to a constant ρ. As shown in the previous problem, such a matrix contains n independent eigenvectors. In practice, one often wishes to perturb or bump a given correlation matrix $\boldsymbol{R}$. In doing so, it is important to ensure that $\boldsymbol{R}$ remains a correlation matrix. There are several ways to perform such a stress. One is as follows.

1. Compute the spectral decomposition of $\boldsymbol{R}$, and obtain $\boldsymbol{R} = \boldsymbol{\Gamma\Delta\Gamma}^T$. Assume that the diagonal elements of Δ are in decreasing order, say $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$.
2. Set $\Delta = \Delta^*$, where Δ^* is the same as Δ, except that $\lambda_1^* = \lambda_1 + \varepsilon$, for some real number ε.
3. Set $\boldsymbol{R}^* = \boldsymbol{\Gamma\Delta}^*\boldsymbol{\Gamma}^T$. Explain why $\boldsymbol{R}^*$ may not be a correlation matrix.
4. Set $\overline{\boldsymbol{R}} = \boldsymbol{D}\boldsymbol{R}^*\boldsymbol{D}^{-1}$, where $\boldsymbol{D}$ is a diagonal matrix with $D_{i,i} = \sqrt{\boldsymbol{R}^*_{i,i}}$.

Investigate the success of this technique using the Dow Jones data that is available in the fBasics package in R. The data can be accessed as follows in R:

```
library("fBasics")
data("DowJones30")
x = DowJones30[,2:31]
```

(Note that the data contains approximately 10 years worth of data so feel free to use a shorter time window if you prefer.) As part of your investigation, you should check if a positive value of ε results in an increase in the average correlation. Does it increase every pairwise correlation? What happens if ε is negative? In the light of the previous problem are you surprised by this? Note that this technique may be used more generally for creating correlation matrices from matrices that look like correlation matrices but are not, for example, positive semidefinite.

Solution. This problem is implemented in R. You are free to use anything you want, but as the dataset is from R you have to at least download it using R. The result show that the average correlation increases (decreases) as $\epsilon > 0$ ($\epsilon < 0$), but not each pairwise correlation increases (decreases). If this was the case, the matrix $\overline{\boldsymbol{R}}$ could not be a correlation matrix. Below is the code and output for this questions.

```
> ## Load libraries and data
> library("fBasics")
> data("DowJones30")
> x <- DowJones30[1:600, 2:31]
>
> # (a) Compute spectral decomposition of the correlation matrix
> c <- cor(x)
> r <- eigen(c)
> gam <- r$vectors
> del <- r$values #c = gam del gam^-1, Note: del is row vector of eigenvalues
>
> ## (b) Perturb first eigenvalue
> eps <- 10
> del[1] <- del[1] + eps
> del[1]
[1] 28.58542
>
> ## (c) Define matrix with perturbed eigenvalue
> # matrix multiplication is optimized given form of del
> cstar <- t(sweep(t(gam), 1, del, "*"))
> cstar <- cstar %*% t(gam)
> # Check that eigenvalues of matrix cstar are indeed given by del
> rstar <- eigen(cstar)
> gamstar <- rstar$vectors
> delstar <- rstar$values
> all.equal(delstar, del)
[1] TRUE
>
> ## (d) Construct correlation matrix cc from rstar
> d <- diag(cstar)
> d <- 1 / d
> cc <- sweep(cstar, 1, d, "*")
> cc <- t(sweep(t(cstar), 1, d, "*"))
>
> ## Comparing cc and c
> # Average correlation increases (decreases) as eps>0 (eps<0)...
> mean(cc) > mean(c)
[1] TRUE
> # ...but not each pairwise correlation increases (decreases)
> comp <- cc > c
> sum(comp)
[1] 710
> # In fact the larger |eps|, the smaller (larger) is sum(comp)
```

Question 4 (10 pts). Download the spreadsheet USRiskFreeRatesWeekly.xls from Canvas. The spreadsheet consists of weekly default-free spot interest rates for the following maturities: 1, 3 and 6 months, and 1, 2, 3, 5, 7, 10 and 20 years. The data is available between August 2001 and February 2010. This question can be done in any of Matlab, R etc.

1. Write a piece of code that performs a Principal Components Analysis on the interest rate data in the USRiskFreeRatesWeekly.xls spreadsheet. Your code should be general enough to perform the PCA on the data in a given time window. In addition to the factor loadings, your code should (i) check that the spectral decomposition is correct, i.e. that $\mathbf{\Sigma} = \mathbf{\Gamma}\mathbf{\Delta}\mathbf{\Gamma}^T$, and (ii) compute the cumulative percentage of the variability that is explained by the principal components (remember to perform your analysis on the changes in interest rates).

2. Run your code for the following time windows: 2007, 2008 and 2009. What do you notice?

Solution. See below the R output.

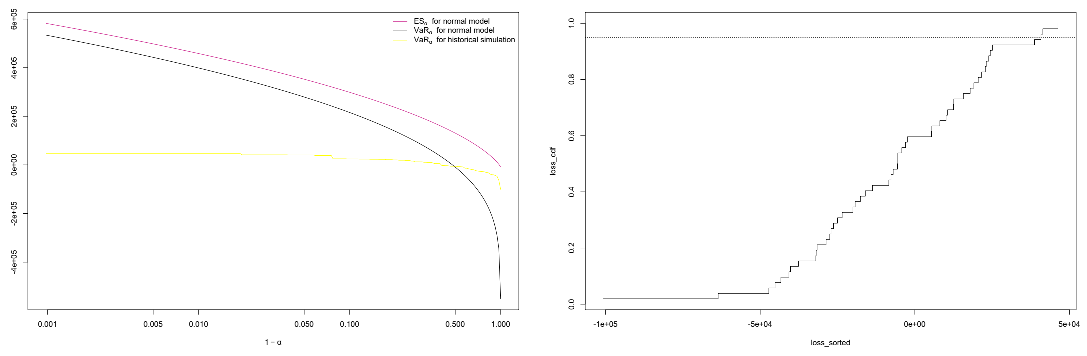

Figure 1: Plots for question 5.

```
> ## Load libraries and data
> library(data.table, qrmtools)
> filename <- here("QRM", "H2", "USRiskFreeRatesWeekly.csv")
> data <- fread(file = filename, header = TRUE)
> dates <- as.Date(as.matrix(data[, 1]))
> t0 <- as.Date("2007-01-01") #initial date
> t1 <- as.Date("2007-12-31") #final date
> t0 <- which(abs(dates - t0) == min(abs(dates - t0))) # find row
> t1 <- which(abs(dates - t1) == min(abs(dates - t1)))
> r <- as.matrix(data[t0:t1, 2:ncol(data)])
> n <- nrow(r)
> ## PCA
> delta_r <- r[2:n, ] - r[1:n - 1, ] #PCA performed on changes
> set.seed(1) #set seed for reproducibility
> res <- princomp(delta_r, scale = FALSE)
Warning message:
In princomp.default(delta_r, scale = FALSE) :
 extra argument 'scale' will be disregarded
> del <- res$sdev^2
> gam <- res$loadings
> # Explained variability
> varexp <- cumsum(del) / sum(del)
> varexp
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
0.6281811 0.9433498 0.9811911 0.9937966 0.9970498 0.9987996 0.9994246 0.9997054
   Comp.9   Comp.10
0.9998590 1.0000000
```

Question 5 (10 pts). Now suppose you have a portfolio of 1-year, 10-year and 20-year zero-coupon default-free bonds, each with one million dollars face value.

1. Compute the current value of your portfolio using the spot rate curve as of January 29th 2010 in the USRiskFreeRatesWeekly.xls spreadsheet.

2. Use the results of your PCA analysis in the previous question for 2009 to estimate the 95% and 99% VaR and CVaR of your portfolio. You should use just the first three principal components and assume that they have a multi-variate normal distribution with a diagonal variance-covariance matrix whose entries are given by the first three eigenvalues.

3. Use a historical Monnte-Carlo to estimate the 95% and 99% VaR and CVaR of your portfolio using the rate data from all three years. Compare your answers with part (b).

Solution.

See figure 1 and the code output below.

```
> ## Load libraries and data
> library(data.table) # recall to run install.packages("data.table")
> library(qrmtools) # recall to run install.packages("qrmtools")
> library(here) # recall to run install.packages("here")
> filename <- here("QRM", "H2", "USRiskFreeRatesWeekly.csv")
> data <- fread(file = filename, header = TRUE)
> dates <- as.Date(as.matrix(data[, 1]))
>
> ## Question 1
> t <- as.Date("2010-01-29")
> t <- which(abs(dates - t) == min(abs(dates - t)))
> r <- as.matrix(data[t, 2:ncol(data)])
> r1 <- r[, "WGS1YR"]
> r10 <- r[, "WGS10YR"]
> r20 <- r[, "WGS20YR"]
> r_pi <- c(1 - r1 / 100, 1 - r10 * 10 / 100, 1 - r20 * 20 / 100)
> f <- 1000000
> pi <- f * (exp(-r1 / 100) + exp(-r10 * 10 / 100) + exp(-r20 * 20 / 100))
> pi_approx <- f * (r_pi %*% c(1, 1, 1))
> pi
 WGS1YR
2103535
> pi_approx
       [,1]
[1,] 1746900
>
> ## Question 2
> # PCA
> library("matrixStats")
> t0 <- as.Date("2009-01-01") #initial date
> t1 <- as.Date("2009-12-31") #terminal date
> t0 <- which(abs(dates - t0) == min(abs(dates - t0))) # find row
> t1 <- which(abs(dates - t1) == min(abs(dates - t1)))
> r <- as.matrix(data[t0:t1, 2:ncol(data)])
> n <- nrow(r)
> dr <- r[2:n, ] - r[1:n - 1, ] #PCA performed on changes
> mu_dr <- colMeans(dr)
> sig_dr <- sqrt(colVars(dr))
> y <- (dr - mu_dr) / sig_dr #normalized vectors of changes
> set.seed(1) #set seed
> res <- princomp(y, scale = FALSE)
Warning message:
In princomp.default(y, scale = FALSE) :
 extra argument 'scale' will be disregarded
> del <- res$sdev
> gam <- res$loadings
> del3 <- del[1:3]
> gam3 <- gam[, 1:3]
> gam3 <- gam3[c("WGS1YR", "WGS10YR", "WGS20YR"), ]
> # VaR
> mu_dr_pi <- c(mu_dr["WGS1YR"] / 100,
+               mu_dr["WGS10YR"] * 10 / 100, mu_dr["WGS20YR"] * 20 / 100)
> mu_pi <- -f * sum(mu_dr_pi)
> sig_dr_pi <- diag(c(sig_dr["WGS1YR"], sig_dr["WGS10YR"], sig_dr["WGS20YR"]))
> sig_pc <- (del[1:3])^2 # standard deviation of principal components
> sig_pi_1 <- sweep(t(gam3), 1, sig_dr_pi, "*")
Warning message:
In sweep(t(gam3), 1, sig_dr_pi, "*") :
  STATS is longer than the extent of 'dim(x)[MARGIN]'
> sig_pi <- t(sweep(t(sig_pi_1), 1, sig_pc, "*"))
> sig_pi <- sqrt(f^2 * sum(sig_pi %*% t(sig_pi_1)))
> alpha <- 1 - 2^seq(log(1 - 0.001, base = 2), -10, length.out = 256)
> dg3 <- t(sweep(t(gam3), 1, del3, "*"))
> dgd3 <- dg3 %*% t(gam3)
> var_n <- mu_pi + VaR_t(alpha, scale = sig_pi, df = Inf)
> es_n <- mu_pi + ES_t(alpha, scale = sig_pi, df = Inf)
>
> ## Question 3
> # Historical Simulation
> r <- as.matrix(data[t0:t1, 2:ncol(data)])
> r <- r[, c("WGS1YR", "WGS10YR", "WGS20YR")]
> dr <- (r[2:n, ] - r[1:n - 1, ])
> loss <- -f * rowSums(dr * c(1 / 100, 10 / 100, 20 / 100))
> loss_sorted <- sort(loss)
> loss_cdf <- seq_along(loss_sorted) / length(loss_sorted)
> # Plot
> # For multiple plots in vscode, activate httpgd in Files > Pref > Settings >$
> plot(loss_sorted, loss_cdf, type = "s")
> abline(h = 0.95, lty = 3)
> var_hs <- loss_sorted[which(loss_cdf >= alpha[1])[1]]
> for (a in alpha[2:length(alpha)]) {
+   var_hs <- c(var_hs, loss_sorted[which(loss_cdf >= a)[1]])
+ }
>
> #Plot
> plot(1 - alpha, es_n, type = "l", ylim = range(var_n, es_n, var_hs),
+     log = "x", col = "maroon3", xlab = expression(1 - alpha), ylab = "")
> lines(1 - alpha, var_n, col = "black")
> lines(1 - alpha, var_hs, col = "yellow")
> legend("topright", bty = "n", lty = rep(1, 3),
+       col = c("maroon3", "black", "yellow"),
+       legend = c(expression(ES[alpha] ~ ~ "for normal model"),
+                  expression(VaR[alpha] ~ ~ "for normal model"),
+                  expression(VaR[alpha] ~ ~ "for historical simulation")))
```