This exercise sheet is a supplemental material to the lecture Financial Econometrics: Risk Management\(^1\) at the LMU Munich. We focus our attention on quantifying central market risk measures such as Value-at-Risk (VaR) and Expected Shortfall (ES). These are trained on different everyday problems as follows:

- we express VaR and ES in closed form in the cases of normally and Student’s t location-scale distributed returns with known or unknown parameters;
- we obtain nonparametric empirical estimations of VaR and ES based on historical data;
- we assess the uncertainty of our estimations by means of parametric and nonparametric bootstrapping;
- we use time-series filtering in a dynamic model-based approach for VaR and ES prediction;
- we build up a multivariate model-based setup for risk assessment of portfolios.

\(^1\)http://www.finmetrics.statistik.uni-muenchen.de/studium_lehre/sommersemester-2013/riskman_13/index.html, the following exercises are based on David Ruppert Statistics and Data Analysis for Financial Engineering. Springer, 2011.
1. **Problem**

Suppose that the yearly return on a stock is normally distributed with mean 0.03 and standard deviation 0.15. An investor purchases 20,000 € worth of this stock.

   a) What are the VaR_{0.95} and VaR_{0.99} with T equal to one year?

   b) What are the ES_{0.95} and ES_{0.99} with T equal to one year?

   c) Plot the VaR_{\alpha} and ES_{\alpha} for values of \alpha ranging between 0.9 and 0.999.

*Hint:* The density, cumulative probability and the quantile functions of the normal distribution are computed by `dnorm`, `pnorm` and `qnorm` respectively.

**Solution**

We know that the returns are normally distributed \( y \sim N(0.03, 0.15^2) \). Hence, the loss function follows the same distribution "mirrored" at the x-axis, namely \( L \sim N(-0.03, 0.15^2) \). Therefore, the respective risk measures take the form

\[
\text{VaR}_\alpha = \mu_L + \sigma \Phi^{-1}(\alpha) \quad \text{and} \quad \text{ES}_\alpha = \mu_L + \sigma \frac{\phi(\Phi^{-1}(\alpha))}{1 - \alpha}.
\]

```r
## a) VaR
alpha <- c(0.95, 0.99)
qnorm(alpha, mean=-0.03, sd=0.15) * 20000
## [1] 4335 6379

## or equivalently
(-0.03 + 0.15 * qnorm(alpha)) * 20000
## [1] 4335 6379

## b) ES
(-0.03 + 0.15 * dnorm(qnorm(alpha))/(1 - alpha)) * 20000
## [1] 5588 7396

## c) plot both
x <- seq(0.9,0.999, length=100)
yVaR <- (-0.03 + 0.15 * qnorm(x)) * 20000
yES <- (-0.03 + 0.15 * dnorm(qnorm(x))/(1 - x)) * 20000
plot(x, yVaR, type="l", ylim=range(yVaR, yES), xlab=expression(alpha), ylab="")
lines(x, yES, lty=2, col=2)
legend("topleft", legend=c("VaR","ES"),col=1:2, lty=1:2)
```
2. Problem

Now re-calculate Problem 1 after assuming a Student’s \( t \) location-scale distribution for the returns. The mean \( \mu = 0.03 \) remains equivalent to the previous exercise. In addition, we also have a scale parameter \( \lambda = 0.116 \) and \( \nu = 5 \) degrees of freedom, i.e., \( y \sim t(\mu = 0.03, \lambda = 0.116, \nu = 5) \). Can you repeat Problem 1, a) and b)? Comment on your results.

**Hint**: The Student’s \( t \) location-scale distribution is provided by the \texttt{sn} package through the functions \texttt{dst}, \texttt{pst}, \texttt{qst}, and \texttt{rst}. For example, the density of a \( t(\mu = 0.03, \lambda = 0.116, \nu = 5) \) distribution at point \( x \) is evaluated by entering \texttt{dst(x, location=0.03, scale=0.116, df=5)} in the R console.

**Hint 2**: The expected shortfall is

\[
\text{ES}_\alpha = \mu_L + \lambda \cdot \frac{f_\nu(F_\nu^{-1}(\alpha))}{1 - \alpha} \cdot \frac{\nu + [F_\nu^{-1}(\alpha)]^2}{\nu - 1}
\]

where \( f_\nu(\cdot) \) and \( F_\nu(\cdot) \) are the density and the cumulative density of the standard \( t \)-distribution with \( \nu \) degrees of freedom. It is readily seen that letting \( \nu \to \infty \) results in the formula for the \( \text{ES}_\alpha \) of the normal distribution.

**Solution**

Note that the scale parameter \( \lambda \) is carefully chosen so that the standard deviation is \( \sigma \approx 0.15 \). This is due to the following relation: suppose that a random variable \( X \) follows the standard \( t \)-distribution with \( \nu \) degrees of freedom, i.e., \( X \sim t(\nu) \). It is known that \( \sqrt{\nu} \sim t(\nu) \). Therefore, \( Y = \lambda X + \mu \) follows the \( t \) location-scale distribution\(^2\) with

\[
\sqrt{Y} = \lambda^2 \sqrt{X} = \lambda^2 \frac{\nu}{\nu - 2}.
\]

We will use \( Y \sim t(\mu, \lambda, \nu) \) to denote this distribution. With our example, we have

\[
\sqrt{Y} = 0.116^2 \cdot \frac{5}{3} \approx 0.0225 \quad \Leftrightarrow \quad \text{sd}(Y) = \sqrt{0.0225} = 0.015.
\]

Therefore, the returns are \( y \sim t(\mu, \lambda, \nu) \) distributed. Hence, the loss \( L \) follows the \( t(-\mu, \lambda, \nu) \) distribution.

\(^2\)See also [http://grollchristian.wordpress.com/2013/04/30/students-t-location-scale/](http://grollchristian.wordpress.com/2013/04/30/students-t-location-scale/) for an elaborate report about the Student’s \( t \) location-scale distribution.
library("sn")

## a) VaR
alpha <- c(0.95, 0.99)
mu <- 0.03
lambda <- 0.116
nu <- 5
q <- qst(alpha, location=-mu, scale=lambda, df=nu)
q * 20000

## [1] 4075 7207

## b) ES
f <- dt(qt(alpha, df=nu), df=nu)
(-mu + lambda * f / (1-alpha) * (nu + qt(alpha, df=nu)^2) / (nu-1) ) * 20000

## [1] 6105 9730

Obviously, assuming a fat-tailed distribution leads to greater losses in the extreme cases. For example, in the normal case we got VaR_{0.99} = 6379, while now VaR_{0.99} increased to 7207. The difference becomes greater with an increasing \( \alpha \).

3. Problem

Suppose an investor wants to take a position in an S&P 500 index fund with a 20,000 € investment. Index funds are portfolios designed to match the movements of major stock indices, in this case the leading U.S. indicator Standard & Poor's 500 (S&P 500). Therefore, her investment is doing as well as the S&P 500.\(^3\)

a) We are interested in the period ending in April, 2013. Get the 1,000 closing levels of S&P 500 prior to that date and plot them. For that purpose, you can use the `get.hist.quote()` function from the `tseries` package. The S&P 500 index is obtained by setting `instrument = "^gspc"` and `quote = "Close"` in `get.hist.quote()`.

b) Compute the returns in that period\(^5\) and depict the corresponding histogram using 50 breakpoints, i.e., `breaks=50`. Normalize it to get the empirical density and indicate the quantile corresponding to the empirical VaR\(_{0.95}\).

c) Compute the empirical VaR\(_{0.95}\) and ES\(_{0.95}\). In some textbooks you find the empirical VaR and ES under the description VaR and ES based on historical simulation or nonparametric VaR and ES but the meaning is the same.

d) Find 95% bootstrap confidence intervals for the estimates obtained in c). We will consider two possible solutions: an "intuitive" one using a `for`-loop and an alternative utilizing the `boot` function from the `boot` package. Try to carry out at least one of them by using 10,000 bootstrap replications. Advanced users are encouraged to try both solutions.

e) Plot the return times series from b) and indicate the quantile resulting in VaR\(_{95}\). Do you find this estimation reasonable?

Solution

---

\(^3\)Assuming a very narrow (ignorable) tracking error.

\(^4\)Note that one is usually interested in the Adjusted Close price, i.e., `quote = "AdjClose"` reflecting splits and dividend payments. This is, however, not an issue for market indices.

\(^5\)When downloading data with `get.hist.quote()` you get a `zoo` object. Therefore, you also need to load the `zoo` package to ensure that everything works properly.
a) We have the final date 2013-04-30 at hand and need to get the 1,000 closing quotes of S&P 500 prior to that date. Keep in mind that the date 2010-08-04 defines a time window with exactly 1,000 days in between. Nevertheless, this does not correspond to 1,000 observations of S&P 500 due to weekends and holidays when the markets are closed.

```r
## a)
## find the starting date
sp.end <- as.Date("2013-04-30")
(sp.start <- sp.end - 1000)
## [1] "2010-08-04"

## get SP500
library("tseries")
x <- get.hist.quote(start=sp.start, end = sp.end, instrument = "^gspc",
                    provider = "yahoo", quote="Close", retclass="ts")
nrow(x) # not 1000?
## [1] 689

x <- get.hist.quote(start="1990-01-02", end = sp.end, instrument = "^gspc",
                    provider = "yahoo", quote="Close")
x <- tail(x, 1000)
nrow(x) # 1000
## [1] 1000

plot(x, xlab="S&P 500", ylab="level")
ggrid()
```

![Graph of S&P 500 level from 2010 to 2013](image.png)
## b)
returns
```r
library("zoo")
sp500.ret.zoo <- diff(log(x))  # for this to work you need the zoo library
sp500.ret <- as.numeric(sp500.ret.zoo)
hist(sp500.ret, breaks=50, freq=FALSE)
alpha <- 0.95
q <- quantile(sp500.ret, probs=1-alpha) # (!) mirrored losses
names(q) <- NULL
abline(v=q, col=2)
```

### Histogram of sp500.ret

![Histogram of sp500.ret](image)

## c)
### Empirical VaR (Historical Simulation)
```r
q * (-20000)
```

```
[1] 379.7
```

### Empirical ES (Historical Simulation)
```r
mean(sp500.ret[sp500.ret < q]) * (-20000)
```

```
[1] 554.7
```

## d) bootstrap CI
```r
set.seed(1234)
B <- 10000
n <- length(sp500.ret)
res <- matrix(0, nrow=B, ncol=2, dimnames=list(NULL, c("VaR","ES")))
for (i in 1:B){
x <- sample(sp500.ret, replace=TRUE)
q <- quantile(x, probs=1-alpha)
res[i,"VaR"] <- -q * 20000
res[i,"ES"] <- -mean(x[x < q]) * 20000
} 
CI <- apply(res, 2, function(x) quantile(x, probs = c(0.025,0.975)))
t(CI)
```

Nikolay Robinzonov 6 June 2013
This estimation is obviously not the best choice. We use the quantile of the marginal distribution which, for example, can be roughly approximated by the histogram in b). Therefore, we ignore potential changes in the distribution over time. Entering dynamics in this context is to assume a different distribution (histogram) at each time point $t$. Obviously, we can not produce any meaningful histogram with a single observation.
One popular solution is to use the available information $\mathcal{F}_{t-1}$ up to time $t$ and to predict the future distribution conditioned on that information. The standard GARCH model (Problem 5) is doing precisely this with a special emphasize on the conditional variance.

4. Problem

Use the same data set as in Problem 3 and the same initial investment.

a) First we assume that the returns are i.i.d. and normally distributed. Estimate the parameters of the distribution by using the `fitdistr()` function from the `MASS` package.

b) Now assume that the returns are i.i.d. and $t$ distributed (more precisely, the Student’s $t$ location-scale distribution is meant). Estimate the parameters of this distribution as well.

c) Calculate the $\text{VaR}_{0.99}$ and the $\text{ES}_{0.99}$ for a) and b).

d) Find 95% bootstrap confidence intervals for the estimates obtained in b). Use 50 bootstrap replications.\(^6\)

e) Inspect the fitted distributions in a) and b) by means of Quantile-Quantile Plots, i.e., use the `qqplot()` function. Do you think the assumptions are reasonable?

Solution

```r
## a)
library(MASS)
library(sn)
## Normal distribution
(fit.norm <- fitdistr(sp500.ret, "normal"))
## mean sd
## 0.0005642 0.0113566
## (0.0003593) (0.0002541)
mu.norm <- fit.norm$estimate["mean"]
sd.norm <- fit.norm$estimate["sd"]

## b)
## Student's t location-scale distribution
## Warning: Estimating the degrees-of-freedom can lead to an infinite likelihood
## see Fernandez & Steel (1999, Biometrika) for further details
(fit.t <- fitdistr(sp500.ret, "t"))
## m s df
## 0.0009847 0.0075106 3.0784145
## (0.0002875) (0.0003366) (0.3778855)
mu.t <- fit.t$estimate["m"]
lambda <- fit.t$estimate["s"]
nu <- fit.t$estimate["df"]

## c)
alpha <- 0.99
## VaR
qnorm(alpha, mean=-mu.norm, sd=sd.norm) * 20000 # Normal distr.
## [1] 517.1
qst(alpha, location=-mu.t, scale=lambda, df=nu) * 20000 # t-distr.
```

\(^6\)We use 20 bootstrap samples for demonstration purposes only. Otherwise, you should increase this number and wait.
## ES

\((-\mu_{\text{norm}} + \text{sd}_{\text{norm}} \times \text{dnorm}(\text{qnorm}(\alpha))/(1 - \alpha)) \times 20000 \) # Normal distr.

## [1] 648.7

\(f \gets \text{dt}(\text{qt}(\alpha, \text{df}=\nu), \text{df}=\nu)
\quad (-\mu_{\text{t}} + \lambda \times f / (1-\alpha) \times (\nu + \text{qt}(\alpha, \text{df}=\nu)^2) / (\nu-1) ) \times 20000 \) # t-distr.

## [1] 594.1

## d) bootstrap

### i)

```r
set.seed(1234)
B <- 50
res <- matrix(0, nrow=B, ncol=2, dimnames=list(NULL, c("VaR", "ES")))
for (i in 1:B){
x <- sample(sp500.ret, replace=TRUE)
fit.t <- fitdistr(x, "t")
mu.t <- fit.t$estimate["m"]
lambda <- fit.t$estimate["s"]
nu <- fit.t$estimate["df"]

# VaR
VaR <- qst(alpha, location=-mu.t, scale=lambda, df=nu) * 20000 # t-distr.
# ES
f <- dt(qt(alpha, df=nu), df=nu)
ES <- (-mu.t + lambda * f / (1-alpha) * (nu + qt(alpha, df=nu)^2) / (nu-1) ) * 20000 # t-distr.

res[i,"VaR"] <- VaR
res[i,"ES"] <- ES
}
CI <- apply(res, 2, function(x) quantile(x, probs = c(0.025,0.975)))
t(CI)
```

### ii) alternative solution with "boot"

```r
myfun <- function(x, i){
    fit.t <- fitdistr(x[i], "t")
    mu.t <- fit.t$estimate["m"]
    lambda <- fit.t$estimate["s"]
    nu <- fit.t$estimate["df"]
    # VaR
    VaR <- qst(alpha, location=-mu.t, scale=lambda, df=nu) * 20000 # t-distr.
    # ES
    f <- dt(qt(alpha, df=nu), df=nu)
    ES <- (-mu.t + lambda * f / (1-alpha) * (nu + qt(alpha, df=nu)^2) / (nu-1) ) * 20000 # t-distr.
    return(c(VaR, ES))
}
res <- boot(sp500.ret, statistic = myfun, R = 50, stype="i")
boot.ci(res, conf=0.95, type="perc", index=1) # index=2 for ES
```

### BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

### Based on 50 bootstrap replicates

### CALL:

```r
boot.ci(boot.out = res, conf = 0.95, type = "perc", index = 1)
```
The normal distribution seems to be rather poor. Especially in the tails, the fit is very bad off. Assuming a more flexible distribution such as the Student’s $t$ location-scale distribution improves the goodness-of-fit in the tails but there is still room for improvement. In the next example, we will relax the i.i.d. assumption.

5. Problem

In this example we will carry out the VaR$_{\alpha}$ and ES$_{\alpha}$ in a more realistic setup by filtering potential ARMA/GARCH effects. Therefore, we do not assume i.i.d. data anymore. The data set and the initial investment remain as in the previous exercises.

a) Is there an evidence for dependencies in the conditional mean and in the conditional variance of the data? Use the `acf` function on the (squared) returns for this purpose. Use the Ljung-Box multiple test statistic to test for auto-correlations in the mean and in the variance? Filter the data with the ARMA(p,q) model by using the `auto.arima` function from the `forecast` package. By "filtering" we mean that we fit the following model

$$y_t = \mu + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

where $\epsilon_t \sim N(0, \sigma^2)$. We continue our work with the residuals, say $r_t = \hat{\epsilon}_t$, afterwards. What do we achieve by this filtering? Plot the filtered return auto-correlation and the squared return auto-correlation.
b) Now we want to adjust the model for volatility clustering. Therefore, we want to remove the so called GARCH effects found in the conditional variance. Fit a GARCH(2,1) model on top of the ARMA-filtered series, i.e., on the $r_i$’s from a), with an underlying $t$ location-scale error distribution. This means that the innovations (the errors) $\epsilon_t$ are $t(\mu, \lambda, \nu)$ distributed and we have

$$
\begin{align*}
    r_t &= \sigma_t \cdot \epsilon_t \\
    \sigma_t^2 &= \beta_0 + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \beta_1 \sigma_{t-1}^2 \\
    \alpha_1 + \alpha_2 + \beta_1 &< 1.
\end{align*}
$$

(5)

Use the garchFit function from the fGarch package for this purpose. Standardize the time series and plot them along the original data. This is the “filtering” part in this case. Plot the auto-correlation function of the ARMA/GARCH filtered series. Which effects are more relevant for the downside risk in your opinion: the dependencies in the conditional mean or the dependencies in the conditional variance?

c) Make a prediction of the VaR$_{0.95}$ and ES$_{0.95}$ for the next period $T+1$. Use forecast.Arima for the mean and predict for the standard deviation in period $T+1$. The scale parameter $\lambda$ is not directly estimated by the garchFit function and you will have to derive it.

d) Plot the VaR$_{0.95}$ based on the conditional standard deviation for the whole period. Thus, we will get the conditional (time-dependent) VaR$_{0.95,t}$. Compare it graphically to the parametric VaR$_{0.95}$ based on the marginal distribution and to the empirical VaR$_{0.95}$.

Solution

a) To examine the dependencies in the conditional mean and variance we consider the sample auto-correlation functions for both the returns and the squared returns. These can be regarded as indicators for the existence of dependencies in the mean and in the variance respectively.

```
## a)
layout(t(1:2))
acf(sp500.ret)
acf(sp500.ret^2)
```
There is a slight evidence for dependencies in the conditional mean and a much stronger one in the conditional variance. This can be tested using the Ljung-Box multiple test statistic

\[ Q(m) = T(T + 2) \sum_{l=1}^{m} \frac{\hat{\rho}_l^2}{T-l} \]  

(6)

where the null hypothesis \( H_0 : \rho_1 = \cdots = \rho_m = 0 \) assumes no correlation among the first \( m \) lags. Under \( H_0 \), the test statistic \( Q(m) \) is \( \chi^2(m) \)-distributed. This test is implemented in the `Box.test(..., type="Ljung-Box")` function.

```r
Box.test(sp500.ret, lag=1, type="Ljung-Box")
##
## Box-Ljung test
##
## data: sp500.ret
## X-squared = 5.315, df = 1, p-value = 0.02114

Box.test(sp500.ret^2, lag=1, type="Ljung-Box")
##
## Box-Ljung test
##
## data: sp500.ret^2
## X-squared = 34.45, df = 1, p-value = 4.373e-09
```

We should take these effects into account. First, we remove the mean-signal (the dependencies in the conditional mean) by fitting an ARMA(p,q) model to the data.

```r
library(forecast)
fit.arma <- auto.arima(coredata(sp500.ret)) # ARMA(3,2)
sp500.ret.arma <- resid(fit.arma)
acf(sp500.ret.arma, na.action=na.pass)
acf(sp500.ret.arma^2, na.action=na.pass)
```

```r
Box.test(sp500.ret.arma, lag=1, type="Ljung-Box")
```
## Box-Ljung test

```r
data: sp500.ret.arma
X-squared = 0, df = 1, p-value = 0.9995
```

Box.test(sp500.ret.arma^2, lag=1, type="Ljung-Box")

```r
X-squared = 13.03, df = 1, p-value = 0.0003066
```

We managed to reduce the auto-correlation in the series which translates into weaker dependence in the conditional mean. However, the auto-correlation in the squared returns remains high which is an evidence for dependence in the conditional variance. Next, we will account for this.

b) Now we will consider the time-varying nature of the conditional standard deviation by fitting a GARCH(2,1) model to the data with $t$ distributed innovations.

```r
## b) GARCH(2,1)
library(fGarch)
fit.garch <- garchFit(~garch(2,1), sp500.ret.arma, cond.dist="std", trace=FALSE)
fit.armagarch <- garchFit(~arma(3,2) + garch(2,1), sp500.ret, cond.dist="std", trace=FALSE)
csd <- fit.garch@sigma.t  # conditional standard deviation
sp500.ret.arma.garch <- sp500.ret.arma/csd  # GARCH filter
par(mfrow=c(2,1))
plot(fit.garch, which=3)
plot(sp500.ret.arma.garch, type="l", col="steelblue", main="Standardized series")
```

![Series with 2 Conditional SD Superimposed](image)

![Standardized series](image)

We clearly see volatility clusters in the upper panel. This means that $\mathbb{V}(y_t | \mathcal{F}_{t-1}) = \sigma_{t-1}^2$ is heteroskedastic. In other words, the variance conditioned on the information set $\mathcal{F}_{t-1} = y_{t-1}$...
is time-dependent. In our notation, this is indicated by the index \( t \) in \( \sigma_t \). After “filtering” the GARCH effects, i.e., standardizing them as in

\[
\text{sp500.ret.arma.garch} \leftarrow \text{sp500.ret.arma/csd}
\]

we observe that the auto-correlation vanishes in the (filtered) squared returns as well (previous figure, bottom panel).

Note that the \texttt{fGarch} package is quite flexible and we could have filtered both effects in one call, e.g.,

\[
\text{garchFit(arma(3,2) + garch(2,1), sp500.ret, cond.dist="std")} \quad \# \text{ARMA(3,2)-GARCH(2,1)}
\]

In addition, this alternative solution would also be more accurate (why?) but we separated the estimation process in two steps for training purposes.

\[
\text{par(mfrow=c(1,2))}
\]
\[
\text{acf(sp500.ret.arma.garch)}
\]
\[
\text{acf(sp500.ret.arma.garch^2)}
\]

Downside risk measures like VaR and ES are much more sensitive about changes in the volatility compared to changes in the mean. Therefore, GARCH effects in the underlying time series deteriorate their estimations heavily.
c) We need to extract the parameter predictions first.

```r
## c)
alpha <- 0.95
tau <- forecast.Arima(fit.arma)$mean[1]
nu <- coef(fit.garch)["shape"]
sd <- predict(fit.garch, n.ahead=1)["standardDeviation"]
```

Since we do not have a direct estimation of the scale parameter \( \lambda \) we will estimate it. This parameter is needed to compute the quantile through the `qst` function.\(^7\) Rearranging Equation (3) we have

\[
\lambda_t = \sigma_t \sqrt{\frac{\nu - 2}{\nu}}. \tag{7}
\]

Note that we assume the scale parameter \( \nu \) being time invariant.

```r
lambda <- sd * sqrt((nu-2)/nu) # Equation (7)
```

Note that we assume the scale parameter \( \nu \) being time invariant.

```r
## VaR
qst(alpha, location=-mu, scale=lambda, df=nu) # pkg: sn
```

```r
## [1] 0.01233
```

```r
(q <- qstd(alpha, mean=-mu, sd=sd, nu=nu)) # pkg: fGarch
```

```r
## [1] 0.01233
```

```r
(VaR <- 20000 * q)
```

```r
## [1] 246.6
```

```r
## ES
f <- dt(qt(alpha, df=nu), df=nu)
(-mu + lambda * f / (1-alpha) * (nu + qt(alpha, df=nu)^2) / (nu-1) ) * 20000
```

```r
## [1] 349.8
```

\[d\] First we visualize the marginal (parametric), the empirical, and the conditional VaR\(_{0.95}\). The latter is a dynamic function of time, the first and the second one are static throughout the whole time window. The marginal distribution is estimated by the `fitdistr` from the `MASS` package (see Exercise 4 b), the empirical VaR\(_{0.95}\) is based on the \( 1 - \alpha \) empirical quantile of the returns (see Exercise 3 c). For the conditional VaR you will need the one-step ahead predictions for the whole period. Therefore, the conditional mean is extracted by the function `fitted.Arima()` from the `forecast` package, while the conditional standard deviation is obtained through the `@sigma.t` slot from the `fGarch` object.

```r
## d)
## parametric estimation (marginal SD)
fit.marg <- fitdistr(sp500.ret, "t")
mu.marg <- fit.marg$estimate["m"]
nu.marg <- fit.marg$estimate["s"]
lambda.marg <- fit.marg$estimate["df"]
```

\[\text{Alternatively}, \text{ the `qstd` function from the `fGarch` package makes use of Equation (7) automatically.}\]
Considering the dynamics in the conditional standard deviation gives a much more reasonable estimation of the underlying volatility than any static alternative. VaR\(_\alpha\) and ES\(_\alpha\) are greatly affected by the standard deviation, therefore, taking the GARCH-effects into account is mandatory.

6. Problem

Load the file `Rweek.csv` which contains weekly (Thursday to Thursday) percentage net returns of the S&P 500, FTSE, and DAX indices over the period from January 1984 to August 2005, a sample of 1127 observations. The returns are denominated in terms of dollars. We denote the return vector at time \(t\) by \(r_t = (r_{1t}, r_{2t}, r_{3t})^T\), where \(r_{1t}\), \(r_{2t}\), and \(r_{3t}\) are the time \(t\) returns of the S&P 500, the FTSE, and the DAX, respectively. Assume an equally weighted portfolio with weights \(x = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T\).
a) Estimate the mean return of the portfolio defined by

\[ \mu_p = \hat{x}^\top \hat{\mu} \]

where \( \hat{\mu} \) denotes the sample mean.

b) Fit a GARCH(1,1) model with normally distributed innovations to each of the three assets. Filter the GARCH effects, i.e., standardize the returns.

c) Estimate the portfolio variance of the standardized series by

\[ \sigma_p^2 = \hat{x}^\top \hat{\Sigma} \hat{x} \]

where \( \hat{\Sigma} \) denotes the empirical covariance matrix. Make a one step ahead prediction of the conditional variances of all three series fitted in b). Thus we get \( \hat{\sigma}^2_{1,T+1}, \hat{\sigma}^2_{2,T+1} \) and \( \hat{\sigma}^2_{3,T+1} \), respectively. Plug these into the main diagonal of \( \hat{\Sigma} \) in order to get a dynamic estimation \( \hat{\Sigma}_{T+1} \) and, therefore

\[ \sigma_p^2_{T+1} = \hat{x}^\top \hat{\Sigma}_{T+1} \hat{x}. \]

Compute the portfolio's VaR\(_{0.95}\) for the next period \( T + 1 \).

Solution

```r
## a)
library("fGarch")
#rweek <- read.table("Rweek.csv", header=TRUE, sep="", dec=".")
rweek <- read.table("http://robinzoni.userweb.mwn.de/riskman/data/Rweek.csv", header=TRUE, sep="", dec=".")
rweek <- rweek[,,-1]
rweek <- rweek/100 # net returns (not percentage anymore)
x <- rep(1/3, 3) # weights
mu_p <- colMeans(rweek) %*% x # portfolio return
mu_p
## [,1]
## [1,] 0.002192

## b)
## GARCH(1,1)
# if("rweek" %in% search()) detach(rweek)
attach(rweek)
sp500.garch <- garchFit(~garch(1,1), sp500, cond.dist="norm", trace=FALSE)
ftse.garch <- garchFit(~garch(1,1), ftse, cond.dist="norm", trace=FALSE)
dax.garch <- garchFit(~garch(1,1), dax, cond.dist="norm", trace=FALSE)

# Standardize
sp500.stand <- sp500 / sp500.garch@sigma.t
ftse.stand <- ftse / ftse.garch@sigma.t
dax.stand <- dax / dax.garch@sigma.t

# c)
Cor <- cor(cbind(sp500.stand, ftse.stand, dax.stand))
(Cov <- cov(cbind(sp500.stand, ftse.stand, dax.stand)))

# sp500.stand ftse.stand dax.stand
# sp500.stand 1.0013 0.5295 0.4610
# ftse.stand 0.5295 1.0067 0.5974
# dax.stand 0.4610 0.5974 0.9995
```

Be aware for changes in the correlations in \( \Sigma_{T+1} \). Adjust them accordingly.
We make use of the empirical covariance matrix $\hat{\Sigma}$ with elements $\hat{\sigma}_{ij}$ and the respective correlation matrix $\hat{\rho}$ whose elements are $\hat{\rho}_{ij} = \frac{\hat{\sigma}_{ij}}{\hat{\sigma}_i \hat{\sigma}_j}$, where $\hat{\sigma}_i^2 = \hat{\sigma}_{ii}$. Once we get the GARCH estimates $\hat{\sigma}_{1,T+1}^2, \hat{\sigma}_{2,T+1}^2, \hat{\sigma}_{3,T+1}^2$, we plug them into the main diagonal of $\hat{\Sigma}$ and also adjust the correlations through

$$\sigma_{i,j,T+1} = \hat{\rho}_{ij} \hat{\sigma}_{i,T+1} \hat{\sigma}_{j,T+1}$$

or in matrix notation

$$\hat{\Sigma}_{T+1} = \begin{bmatrix} \hat{\sigma}_{1,T+1} & \hat{\sigma}_{2,T+1} & \hat{\sigma}_{3,T+1} \\ \hat{\sigma}_{2,T+1} & \hat{\sigma}_{2,T+1} & \hat{\sigma}_{3,T+1} \\ \hat{\sigma}_{3,T+1} & \hat{\sigma}_{3,T+1} & \hat{\sigma}_{3,T+1} \end{bmatrix} \hat{\rho} \begin{bmatrix} \hat{\sigma}_{1,T+1} \\ \hat{\sigma}_{2,T+1} \\ \hat{\sigma}_{3,T+1} \end{bmatrix}.$$ 

$sigma_p <- x %*% Cov %*% x$

$sig1 <- (predict(sp500.garch, n.ahead = 1)[["standardDeviation"]])$

$sig2 <- (predict(ftse.garch, n.ahead = 1)[["standardDeviation"]])$

$sig3 <- (predict(dax.garch, n.ahead = 1)[["standardDeviation"]])$

$D <- diag(c(sig1, sig2, sig3))$

$(Cov_t <- D %*% Cor %*% D)$

$sd_p <- sqrt(x %*% Cov_t %*% x) # portfolio sd$

# VaR
alpha <- 0.95
$qnorm(alpha, mean = -mu_p, sd = sd_p) * 20000$

# [1] 438.9