STATS 509 HOMEWORK 3 YUAN YIN

Roxanne

Question 1

(a)

```
## Loading required package: timeDate
## Loading required package: timeSeries
## [1] "the median of log returns is  0.000725615176728311"
## [1] "the mean of log returns is  0.000154944220313725"
## [1] "the variance of log returns is  0.000154469719505854"
## [1] "the skewness of log returns is  -0.271763622217888"
## [1] "the kurtosis of log returns is  11.3774755741033"
```

We can see that both median and mean are very close to zero, also the variance is very close to zero, which means that most data is around zero. The skewness is less than zero, which means it's left skewed. And the kurtosis is 11.377, which is far much larger than 0. This means that the real distribution of our data has much heavier tails than normal distribution.

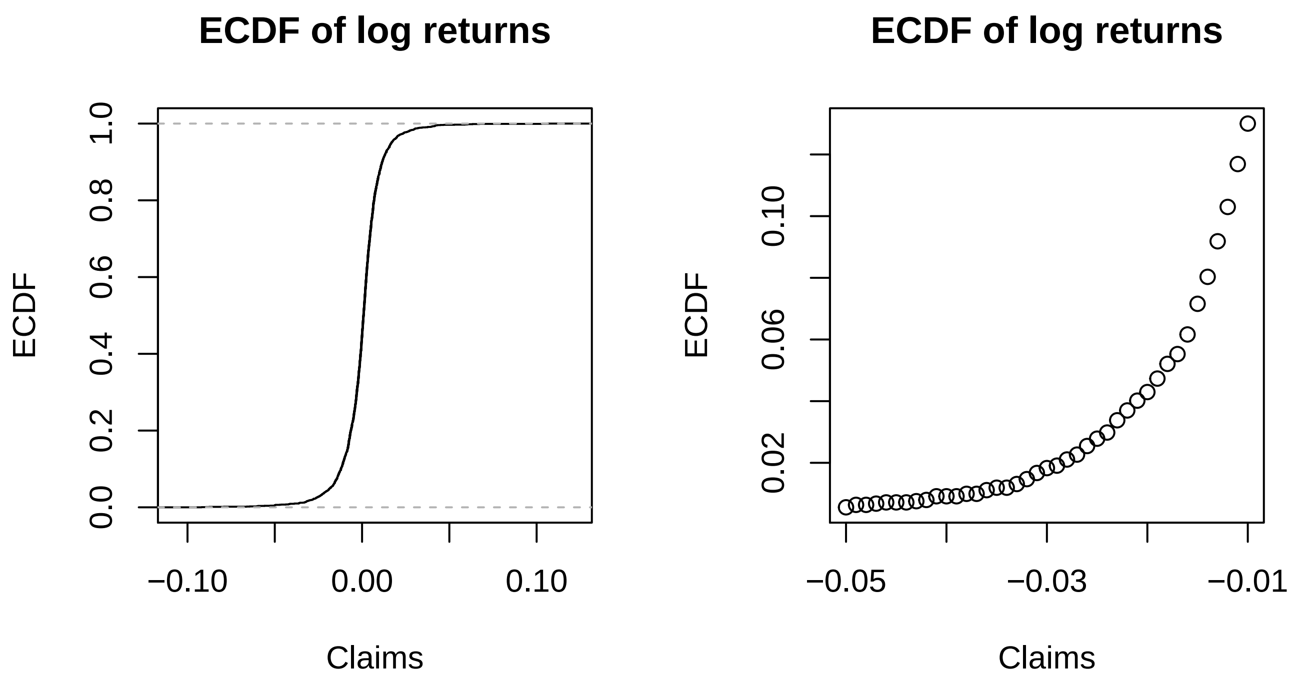
(b)

According the summary above, we want to focus more on the distribution of the tails of our data. A good way is to fit with GPD. As we only want to fit the lower tails, we need to filp over the data before we fit it.

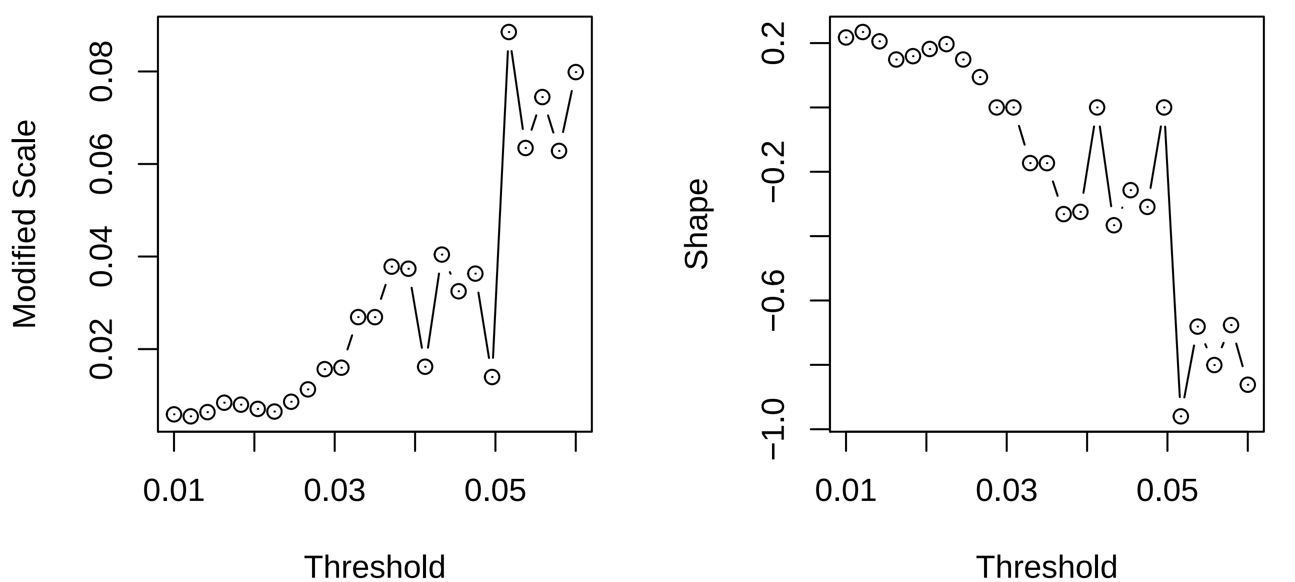
```
library(POT)
library(evir)
```

```
##
## Attaching package: 'evir'
## The following objects are masked from 'package:POT':
##
##     dgpd, pgpd, qgpd, rgpd
```

```
eecdf = ecdf(SP100_dl_lreturn)
par(mfrow = c(1,2))
plot(eecdf, main = 'ECDF of log returns', xlab = 'Claims', ylab = 'ECDF')
uv = seq(from = -0.05, to = -0.01, by = 0.001)
plot(uv, eecdf(uv), main = 'ECDF of log returns', xlab = 'Claims', ylab = 'ECDF')
```



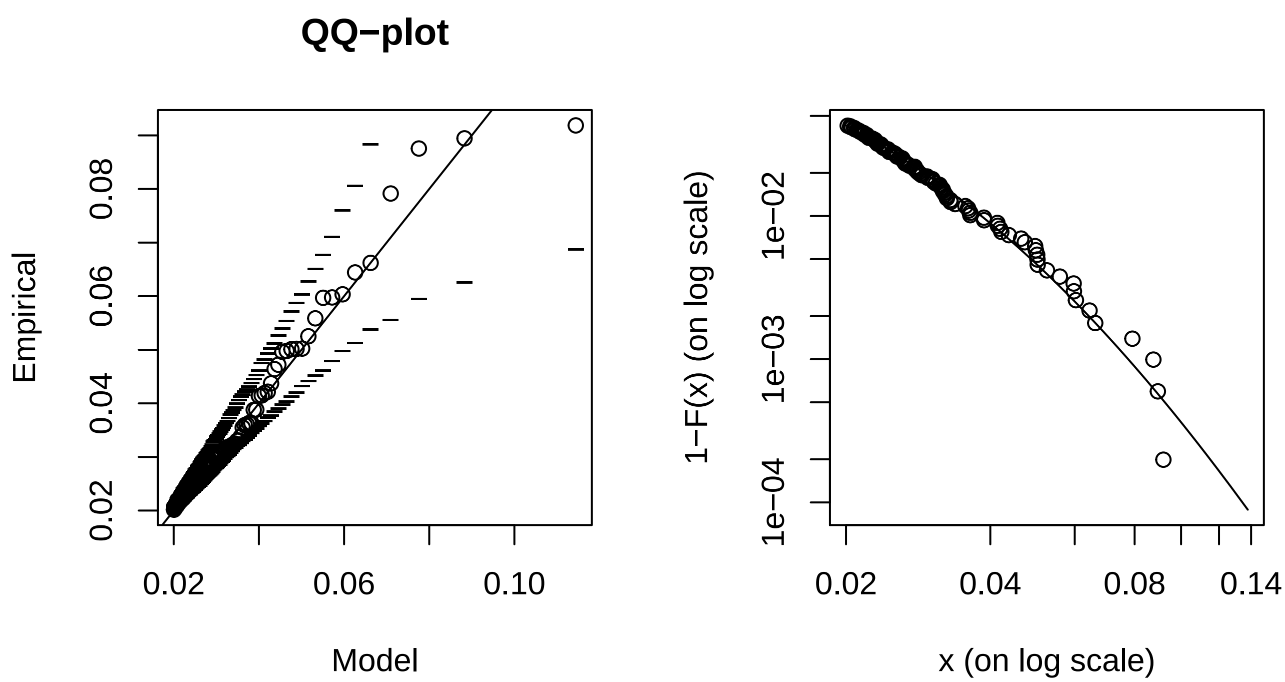

```r
par(mfrow = c(1,2))
tcplot(-SP100_dl_lreturn, c(0.01,0.06), nt = 25, conf = 0)
```




```r
gpd_fit = fitgpd(-SP100_dl_lreturn, 0.02)
par(mfrow = c(1,2))
qq(gpd_fit)
gpd_est = gpd(-SP100_dl_lreturn, thresh = 0.02, method = c("ml"), information = c("observed")
gpd_est$par.ests
```

```
##         xi        beta
## 0.15997766 0.01116821
```

```
tp = tailplot(gpd_est)
```



First we look at the empircal cdf of our data, it shows that around the tail, the cdf plot increases very slowly, which indicades that it has very heavy tails.

Then we use GPD distribution to fit our tail data, first we choose the appropriate threshold from "candidate threshold", as we can see that around interval of [0.01, 0.03], the shape looks pretty stable, thus we choose threshold = 0.02.

After fit our data with GPD, we plot the QQ-plot of our distribution with 95% confidence interval, it shows that they are quite consistent with the line. Also we plot the tails of our estimation. Again, most data is around the line we obtain.

(c)

```
norm_log_return = qnorm(0.005, mean = mean(SP100_dl_lreturn),
                        sd = sd(SP100_dl_lreturn))
norm_return = exp(norm_log_return) - 1
norm_VaR_tilde = - norm_return
```

The $V\tilde{a}R$ corresponding to normal distribution is 0.03135682 at the level = 0.005.

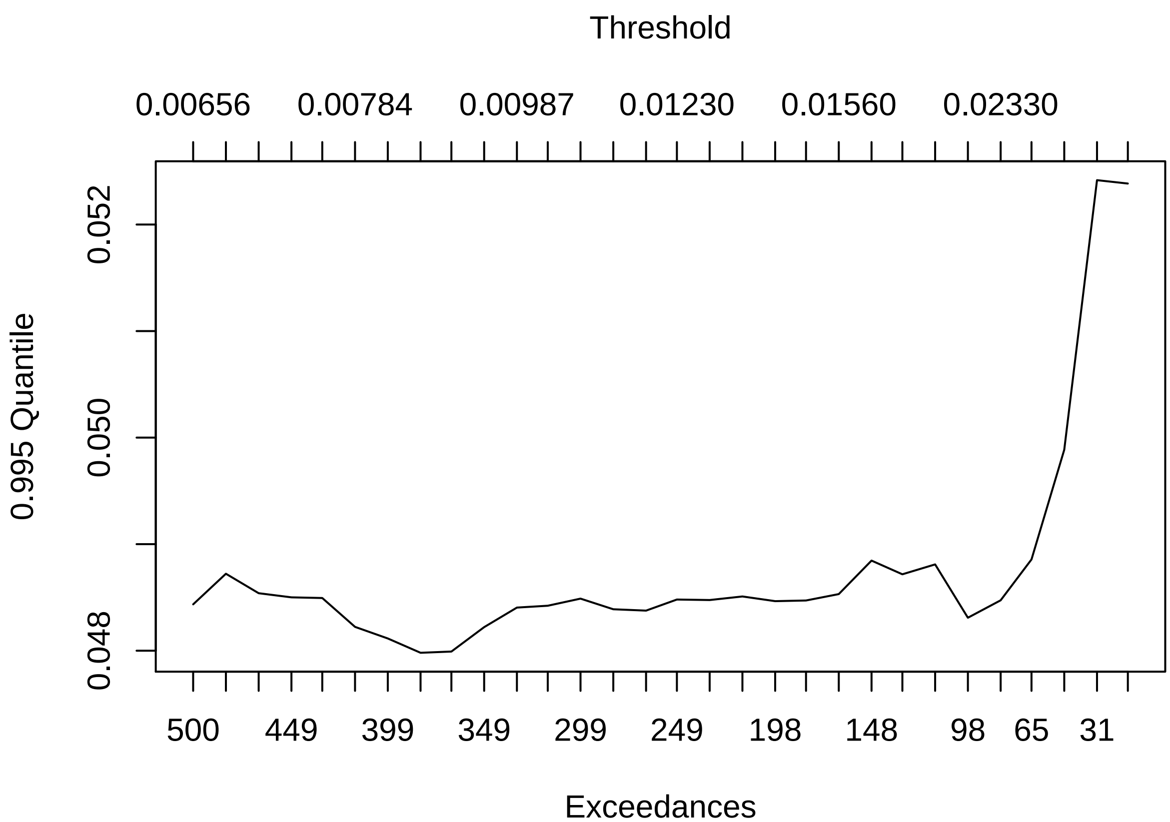
(d)

```
alphat = 1-0.005/eecdf(-0.02)
scale = gpd_est$par.ests[2]
xi = gpd_est$par.ests[1]
m = 0.02
gpd_VaR_tilde = 1 - exp(-evir::qgpd(alphat, xi, m, scale))
```

The $\tilde{VaR}$ corresponding to GPD distribution is 0.04749966, which is larger than relative VaR we compute corresponding to normal distribution. This means that normal distribution under estimate the value of risk of reality.

(e)

```
a = quant(-SP100_dl_lreturn, p = 0.995, models = 30,
          reverse = TRUE, ci = FALSE, auto.scale = TRUE, labels = TRUE)
```



From the plot above we can find that when threshold is in [0.00656,0.02130], the changes of value at 0.995 quantile aren't very large. We can say that VaR is quite stable when threshold is in this interval.

(f)

```r
set.seed(2015)
r = evir::rgpd(1000000, xi, m, scale) # generate random numbers for 1000000 times
port = exp(-r)
s = 0
n = 0
port = -(port - 1)
for (i in port)
{
  if (i > gpd_VaR_tilde)
  {
    s = s + i
    n = n + 1
  }
}
gpd_expection = s/n
gpd_expection
```

```
## [1] 0.06506349
```

Using Monte Carlo simulation, we compute the expected shortfall is 0.06506349 multiplied by our asset $E = 0.065 * portfolio$.

Question 2

(a)

$E(0.6R_1 + 0.4R_2) = 0.6E(R_1) + 0.4E(R_2) = 0.024$
$\omega = 0.4$

$$
\begin{aligned}
Var(0.6R_1 + 0.4R_2) =& (0.6)^2 Var(R_1) + (0.4)^2 Var(R_2) + 2 * \omega(1-\omega)\sigma(R_1)\sigma(R_2) * corr(R_1, R_2) \\
=& 0.00058 + 0.000288 \\
=& 0.000868
\end{aligned}
$$

(b)

First we set the partial equation:

$$
\begin{aligned}
\frac{\partial Var}{\partial \omega} =& \frac{\partial(\omega^2 Var(R_1) + (1-\omega)^2 Var(R_2) + 2\omega(1-\omega)corr(R_1, R_2)\sigma(R_1)\sigma(R_2))}{\partial \omega} \\
=& 2 * (0.003)^2\omega - 2 * (0.004)^2(1-\omega) + 0.0012(1-2\omega) \\
=& 0
\end{aligned}
$$

Then compute this equation and we get $\omega = \frac{10}{13}$

(c)

```
library(MASS)
library(VaRES)
```

```
##
## Attaching package: 'VaRES'

## The following objects are masked from 'package:evir':
##
##     dgev, pgev
```

```
set.seed(12345678)
w = seq(from = 0, to = 1, by = 0.01)
expection = rep(0,length(w))
norm_VaR = rep(0,length(w))
for (i in 1:length(w))
{
  norm_mu = w[i]*0.02 + (1-w[i])*0.03
  norm_sig = sqrt(w[i]^2*0.03^2 + (1-w[i])^2*0.04^2 + w[i]*(1-w[i])*0.03*0.04)
  expection[i] = esnormal(0.005,norm_mu,norm_sig)
  norm_rv = qnorm(0.005, norm_mu, norm_sig)
  norm_VaR[i] = - norm_rv*10^6
}
```

```
## [1] 0.69

## [1] 60658.26

## [1] 51507.06
```

Thus, the ω that minimizes the expected shortfall is 0.69, where the expected shortfall is 60658.26, and it's useful to compute ω is because when we can two stocks with correlation, we can find a method to manage our portfolio to minize the value at risk. The associated VaR with this portfolio is $VaR = 51507.06$.

(d)

```
set.seed(2015)
nu = 6
n1 = 1000000
w = seq(from = 0, to = 1, by = 0.01)
expection = rep(0,length(w))
t_VaR = rep(0,length(w))
for (i in 1:length(w))
{
  t_mu = w[i]*0.02 + (1-w[i])*0.03
  t_sig = sqrt(w[i]^2*0.03^2 + (1-w[i])^2*0.04^2 + w[i]*(1-w[i])*0.03*0.04)
  lambda = t_sig*sqrt(2/3)
  std_qt_rv = qt(0.005, nu)
  qt_rv = t_mu + lambda*std_qt_rv
```

```r
  t_VaR[i] = -qt_rv*10^6
  r = rt(n1, nu)
  port = -10^6 * (t_mu + lambda*r)
  s = 0
  n = 0
  for (j in port)
  {
    if (j > t_VaR[i])
    {
      s = s + j
      n = n + 1
    }
  }
  expection[i] = s/n
}
```

```
## [1] 0.72
```

```
## [1] 86922.21
```

```
## [1] 64615.67
```

Thus when $\omega = 0.72$, the minimum expected short fall is 86922.21, and the associated VaR is 64615.67.