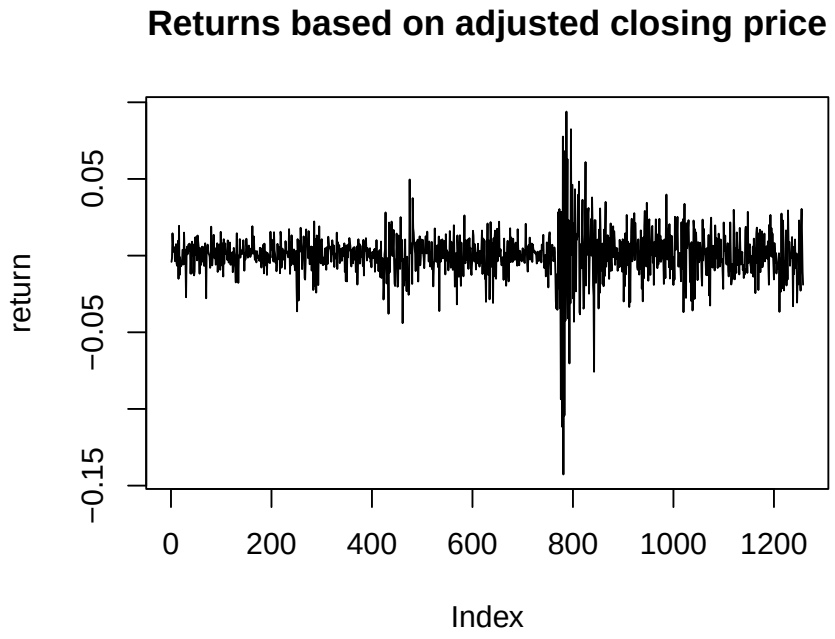
SOLUTIONS TO STATS 509 PROBLEM SET 4

1.

(a).

```
# 1a
price= read.csv('Rus2000_daily_Feb3_2017-Feb3_2022.csv', header = TRUE)
return = diff(price$Adj.Close) / price$Adj.Close[-length(price$Adj.Close)]
dev.new()
pdf('1a.pdf', width = 5, height = 4)
plot(return, type = 'l', main = 'Returns based on adjusted closing price',
     ylab = 'return')
dev.off()
```

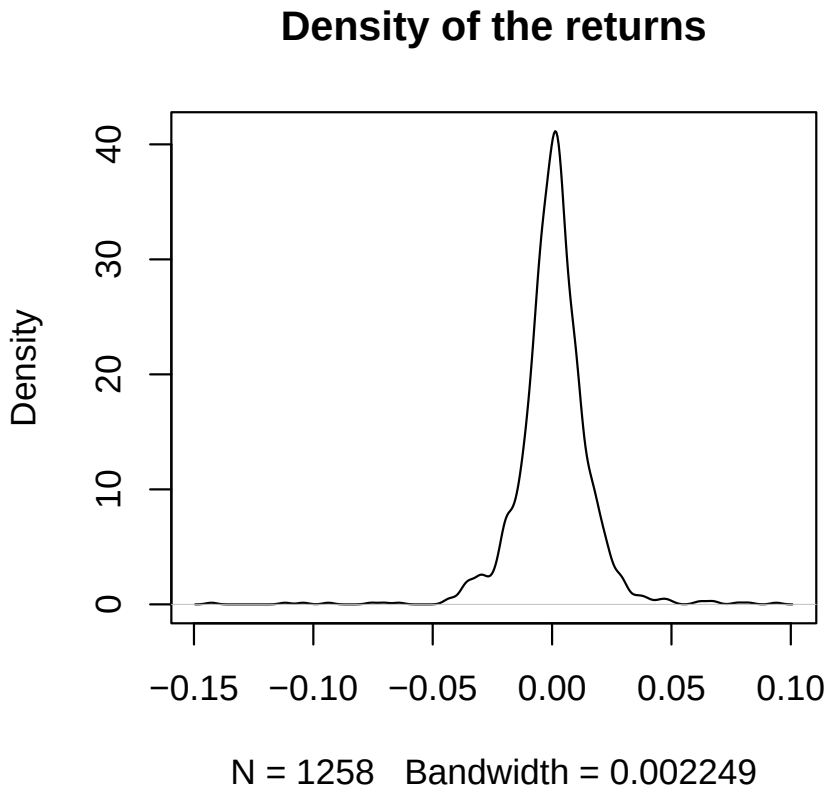


The returns appear to be somewhat stable, but there are regions with high variation/volatility (indices 850-1200) and regions with lower variation/volatility (indices 0-400), and there are very large fluctuations around index 800.

(b).

```
> # 1b
> library(moments)
> median(return)
[1] 0.0008414084
> mean(return)
```

```
[1] 0.0004221921
> var(return)
[1] 0.0002433763
> skewness(return)
[1] -0.9856218
> kurtosis(return)
[1] 16.39863
> plot(density(return), main = 'Density of the returns',
+       ylab = 'Density')
```



The median, mean and variance are all close to 0 which indicates that the returns are colsely centered around 0. The skewness is negative so the data is slightly left skewed. The kurtosis of the data is 13.37, which means that the increased variance is caused by low-frequency extreme differences, and also that the data is much heavier-tailed than normal distribution and even the double exponential distribution. We can also see some evidence for this from the density estimation.

(c). We sue normal distribution to fit the data and get $R \sim \mathcal{N}(0.00084, 0.00024)$. So that the VaR is

$$\widetilde{VaR} = -\mu - \sigma\Phi^{-1}(q).$$

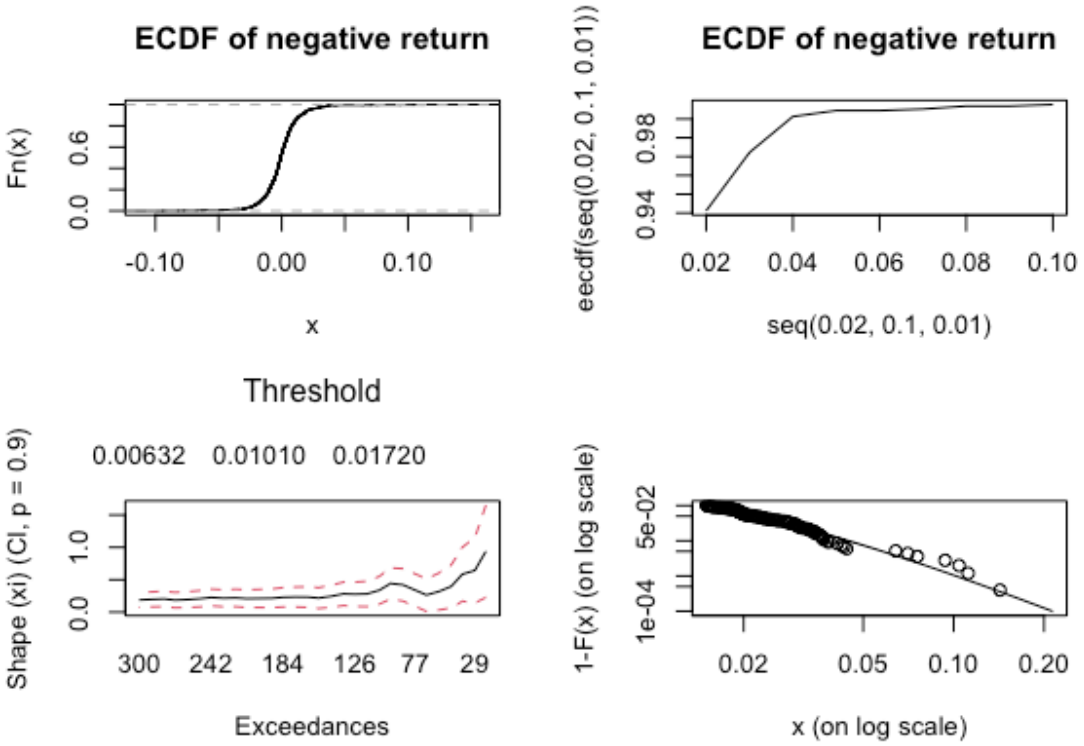
```
> # 1c
> VaR = 1000000 * (-mean(return) - sd(return)*qnorm(0.005))
```

```
> VaR
[1] 39762.09
```

So the VaR is 39762.09 dolloars.

(d).

```
> # 1d
> library(evir)
> eecdf = ecdf(-return)
> par(mfrow=c(2,2))
> plot(eecdf,main='ECDF of negative return')
> plot(seq(0.02,0.1,0.01),eecdf(seq(0.02,0.1,0.01)),
+      type='l',main='ECDF of negative return')
> shape_out = shape(-return, models=30, start=300, end=20,
+                   ci=0.9, reverse=T, auto.scale=T)
> gpd_out = gpd(-return, threshold=0.015, method=c('ml'),
+               information=c("observed"))
> length(which(-return>0.01))
[1] 206
> gpd_out$par.ests
       xi       beta
0.29026972 0.00885722
> tail_out = tailplot(gpd_out)
> scale = as.numeric(gpd_out$par.ests[2])
> xi = as.numeric(gpd_out$par.ests[1])
> m = 0.01
```



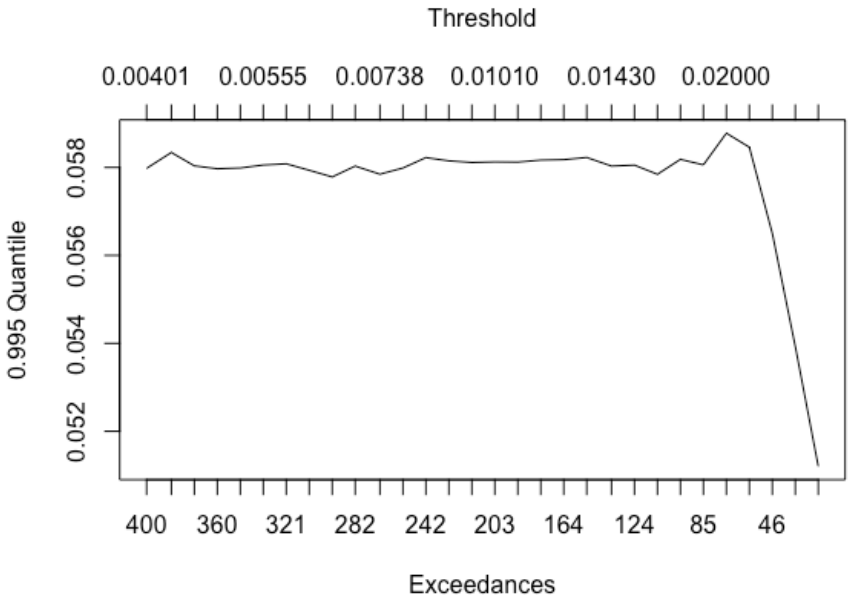
The threshold choices between 0.007 to 0.019 has relatively stable shape parameters. For this reason, we can take the threshold as 0.015 and we will get corresponding estimated parameters ξ, σ. As for the tail plot, GPD seems a good fit for the right tail of negative returns.

(e).

```
> # 1e
> qt = 1-.005/(1-eecdf(0.015))
> xi = gpd_out$par.ests[1]
> scale = gpd_out$par.ests[2]
> VaRt = qgpd(qt,xi,0.015,scale)
> VaRt*1000000
   beta
57985.9
```

(f).

```
> # 1f
> quant(-return, p = 0.995, models = 30, start = 400, end = 20,
+       reverse =TRUE, ci = FALSE, auto.scale = TRUE, labels = TRUE)
```



The threshold choices in this case are quite stable between 0.00401 and 0.02.

(g).

```
# 1g
> # for ES in (c)
> q = 0.005
> z = qnorm(1-q)
> ESt_norm = -mean(return)+(sd(return)/q)*(exp(-z^2/2)/sqrt(2*pi))
> 1e6 * ESt_norm
[1] 44693.72
> # for ES in (e)
> xi = gpd_out$par.ests[1]
> scale = gpd_out$par.ests[2]
> 1e6 * (VaRt + (scale + xi * (VaRt -0.015)) / (1 - xi) )
    beta
88046.23
```

2.

(a). Now

$$E(X) = 0$$

and

$$E(XY) = E\left(X^3\right) = 0$$

since X has symmetric distribibution about 0 , or can show it analytically. Thus

$$\mathrm{Cov}(X, Y) = E(XY) - E(X)E(Y) = 0 - 0 \cdot E(Y) = 0$$

But for X, Y to be independent, would imply that for any functions g, h,

$$E(g(X)h(Y)) = E(g(X))E(h(Y))$$

But if we take

$$g(X) = X^2$$

and

$$h(Y) = Y$$

then since $E(X) = 0$, have

$$E(Y) = E\left(X^2\right) = \text{Var}(X) = \frac{2a^2}{12} = \frac{a^2}{6}$$

But

$$E\left(YX^2\right) = E\left(X^4\right) = \frac{1}{2a}\int_{-a}^{a} x^4 dx = \frac{1}{2a}\frac{2a^5}{5} = \frac{a^4}{5}$$

and this is not equal to

$$E(Y) \cdot E\left(X^2\right) = \frac{a^4}{36}$$

so cannot be independent.

(b). Now the cdf of X is given as

$$F_X(x) = \frac{x+a}{2a} \quad -a \leq x \leq a$$

and

$$\begin{aligned} F_Y(y) &= P\left(X^2 \leq y\right) \\ &= P(-\sqrt{y} \leq X \leq \sqrt{y}) \\ &= \frac{2\sqrt{y}}{2a} = \frac{\sqrt{y}}{a} \end{aligned}$$

Thus

$$F_Y(Y) = F_Y\left(X^2\right) = \frac{\sqrt{X^2}}{a} = \frac{|X|}{a}$$

Hence the Spearman correlation is given as

$$\begin{aligned} \rho_S(X,Y) &= \rho\left(F_X(X), F_Y(Y)\right) \\ &= \rho\left(\frac{X+a}{2a}, \frac{|X|}{a}\right) \\ &= \rho(X, |X|) \end{aligned}$$

where the last line is due to the fact that correlation is unchanged by additive constants and scalings. But since $E(X) = 0$

$$\begin{aligned} \text{Cov}(X, |X|) &= E(X|X|) \\ &= \int_{-a}^{a} x|x|dx \\ &= \int_{0}^{a} x|x|dx + \int_{-a}^{0} x|x|dx \\ &= \int_{0}^{a} x|x|dx - \int_{0}^{a} x|x|dx \\ &= 0 \end{aligned}$$

by Change of variables

(c). Now want to show that $a = 0$, and will do this by showing that there is a vector v such that the $X = \mathbf{v}^\top \mathbf{Y}$ would have a negative variance, which is not possible. But recall from class,

$$\text{Var}(X) = \mathbf{v}^\top \text{Cov}(\mathbf{Y})\mathbf{v}$$

and so just need to find $\mathbf{v}$ such that above is negative, and that would imply that a cannot be 0 . But doing a eigen analysis in R shows that the vector of

$$\mathbf{v} = \begin{bmatrix} .5000 \\ -.7071 \\ .5000 \end{bmatrix}$$

satisfies that

$$\mathbf{v}^\top \text{Cov}(\mathbf{Y})\mathbf{v} = -.273$$

```
> # 2c
> covt = matrix(c(1.0,.9,0 ,.9 ,1.0 ,.9 ,0 ,.9 ,1.0),nrow=3,ncol=3)
> covt
     [,1] [,2] [,3]
[1,]  1.0  0.9  0.0
[2,]  0.9  1.0  0.9
[3,]  0.0  0.9  1.0
> eig_out = eigen(covt)
> eig_out
eigen() decomposition
$values
[1]  2.2727922  1.0000000 -0.2727922

$vectors
          [,1]          [,2]       [,3]
[1,] 0.5000000 -7.071068e-01  0.5000000
[2,] 0.7071068 -8.722736e-16 -0.7071068
[3,] 0.5000000  7.071068e-01  0.5000000

> v = eig_out$vectors[,3]
> vart = t(v) %*% covt %*%v
> vart
           [,1]
[1,] -0.2727922
```

0.1. *(d).* Now want to determine the minimum a. So use R to compute eigenvalues of the matrix for a ranging from 0 to 1, in increments of .001. Then choose the minimum eigen-value and check when this gets above 0. Can show using R that it is approximately .62. The R-code and result are shown below.

```
> # 2d
> a = seq(from = 0, to = 1, by = .001)
> eig_min = 0*a
> for (j in c(1:1001)){
```

```
+   Covt = matrix(c(1.0,.9, a[j] ,.9 ,1.0 ,.9 ,a[j] ,.9 ,1.0),nrow=3,ncol=3)
+   eig_out = eigen(Covt)
+   eig_min[j] = min(eig_out$values)
+   }
> amin = min(a[eig_min>0])
> amin
[1] 0.62
```