Homework 2

Due on Friday, Feb 7

Instructions:

- Install `pdflatex`, `R`, and `RStudio` on your computer.
- Please edit the `HW2_First_Last.Rnw` file in `Rstudio` and compile with `knitr` instead of `Sweave`. Go to the menu `RStudio|Preferences...|Sweave` choose the `knitr` option, i.e., `Weave Rnw files using knitr?` You may have to install `knitr` and other necessary packages.
- Windows users: You may need to install TexLive.
- Replace "First" and "Last" in the file-name with your first and last names, respectively. Complete your assignment by modifying and/or adding necessary R-code in the text below.
- You should submit both the **data** and the *HW2_First_Last.Rnw* file in a zip-file in Canvas. The zip-file should be named "HW2_First_Last.zip" and it should contain a single folder named "First_Last" with all the necessary data files, the `HW2_First_Last.Rnw` and `HW2_First_Last.pdf` file, which was obtained from compiling `HW2_First_Last.Rnw` with `knitr` and LaTeX.

  NOTE: "First" is your first name and "Last" your last name.
- **IMPORTANT:** In addition to your .zip-file, you should submit a PDF file `HW2_First_Last.pdf` in Canvas. The PDF should be submitted **separately** form the zip-file so that the GSIs can annotate it and give you feedback.
- The GSI grader will unzip your file and compile it with Rstudio and knitr. If the file fails to compile due to errors other than missing packages, there will be an automatic 10% deduction to your score. Then, the GSI will proceed with grading your submitted PDF.

Problems:

1. Consider the mixture model

$$f(x; p, \mu, \sigma, c\sigma) = pf(x; \mu, \sigma) + (1-p)f(x; \mu, c\sigma),$$

   where $f(x; \mu, \sigma)$ is the probability density of the Normal distribution with zero mean μ and variance σ^2.

(a) Let $X \sim f(x; p, \mu, \sigma, c\sigma)$. Compute

$$\mathbb{E}|X - \mu| \quad \text{and} \quad \mathbb{E}[(X - \mu)^2]$$

in terms of the parameters p, c and σ.

Let Z be a standard normal variable, i.e., $Z \sim N(0, 1)$. The probability density function (PDF) of Z is:

$$f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

We want to compute the expected value of $|Z|$, which is given by:

$$E[|Z|] = \int_{-\infty}^{\infty} |z| f_Z(z)\, dz$$

$$E[|Z|] = 2 \int_0^{\infty} z \cdot \frac{1}{\sqrt{2\pi}} e^{-z^2/2}\, dz$$

$$\text{Let } u = \frac{z^2}{2} \quad \text{so that} \quad du = z\, dz$$

The limits of integration do not change, so the integral becomes:

$$E[|Z|] = 2 \int_0^{\infty} \frac{1}{\sqrt{2\pi}} e^{-u}\, du$$

The integral $\int_0^{\infty} e^{-u}\, du$ is a standard exponential integral and evaluates to 1. Thus:

$$E[|Z|] = 2 \cdot \frac{1}{\sqrt{2\pi}} \cdot 1 = \sqrt{\frac{2}{\pi}}$$

Now, if $X \sim N(\mu, \sigma^2)$, we can standardize X by converting it to a standard normal variable Z through the transformation:

$$Z = \frac{X - \mu}{\sigma}$$

Thus, we can rewrite $E[|X - \mu|]$ as:

$$E[|X-\mu|] = E[\sigma|Z|] = \sigma E[|Z|]$$

Assume that $X \sim p \cdot N(\mu, \sigma^2) + (1-p) \cdot N(\mu, (c\sigma)^2)$ is a mixture model, where $X_1 \sim N(\mu, \sigma^2)$ and $X_2 \sim N(\mu, (c\sigma)^2)$, with weights p and $1-p$ corresponding to the two normal distributions.

1. Computing $E[|X_1 - \mu|]$

For $X_1 \sim N(\mu, \sigma^2)$, standardizing it as $Z = \frac{X_1-\mu}{\sigma}$, we have $E[|Z|] = \sqrt{\frac{2}{\pi}}$. Therefore,

$$E[|X_1 - \mu|] = \sigma\sqrt{\frac{2}{\pi}}$$

2. Computing $E[|X_2 - \mu|]$

For $X_2 \sim N(\mu, (c\sigma)^2)$, we can standardize it as $Z_2 = \frac{X_2-\mu}{c\sigma}$, and similarly, the expectation is $E[|Z_2|] = \sqrt{\frac{2}{\pi}}$. Thus,

$$E[|X_2 - \mu|] = c\sigma\sqrt{\frac{2}{\pi}}$$

3. Expectation in the Mixture Model

Since X is a mixture of X_1 and X_2 weighted by p and $1-p$, the overall expectation $E[|X - \mu|]$ is:

$$E[|X - \mu|] = p \cdot E[|X_1 - \mu|] + (1-p) \cdot E[|X_2 - \mu|]$$

Substituting the results we computed:

$$E[|X - \mu|] = p \cdot \sigma\sqrt{\frac{2}{\pi}} + (1-p) \cdot c\sigma\sqrt{\frac{2}{\pi}}$$

Simplifying:

$$E[|X - \mu|] = \sigma\sqrt{\frac{2}{\pi}}\,(p + (1-p)c)$$

$$E[(X-\mu)^2] = \mathrm{Var}(X) = p \cdot \mathrm{Var}(X_1) + (1-p) \cdot \mathrm{Var}(X_2)$$

Where:

- $X_1 \sim N(\mu, \sigma^2)$ has variance σ^2,
- $X_2 \sim N(\mu, (c\sigma)^2)$ has variance $(c\sigma)^2$,

Thus, the formula simplifies to:

$$E[(X-\mu)^2] = p \cdot \sigma^2 + (1-p) \cdot (c\sigma)^2$$

This expands to:

$$E[(X-\mu)^2] = p \cdot \sigma^2 + (1-p) \cdot c^2\sigma^2$$

Factoring out σ^2:

$$E[(X-\mu)^2] = \sigma^2 \left(p + (1-p) \cdot c^2\right)$$

(b) The parameter $\sigma > 0$ is known. Given an iid sample $X_1, \ldots, X_n \sim f(x; p, \mu, \sigma, c\sigma)$, construct estimators for the unknown parameters $p,\ \mu,$ $c > 0$ using the (generalized) method of moments.

Hint: You can estimate μ via $\overline{X}_n$. Consider the statistics

$$\widehat{m}_1 := \frac{1}{n}\sum_{i=1}^{n} |X_i - \overline{X}_n| \quad \text{and} \quad \widehat{m}_2 := \frac{1}{n}\sum_{i=1}^{n} (X_i - \overline{X}_n)^2,$$

which estimate $\mathbb{E}|X-\mu|$ and $\mathbb{E}[(X-\mu)^2]$, respectively. Using the formulae derived in **(a)**, solve for p and c.

We know the following equations for m_1 and m_2:

$$m_1 = \sigma\sqrt{\frac{2}{\pi}}\,(p + (1-p)c)$$

and

$$m_2 = \sigma^2\left(p + (1-p)c^2\right).$$

By dividing m_1 by σ and multiplying by $\sqrt{\frac{\pi}{2}}$, we get:

$$\frac{m_1}{\sigma}\sqrt{\frac{\pi}{2}} = p + (1-p)c = p + c - pc.$$

This leads to:

$$(1-c)p = \frac{m_1}{\sigma}\sqrt{\frac{\pi}{2}} - c.$$

Similarly, from the expression for m_2, we have:

$$p(1-c^2) = \frac{m_2}{\sigma^2} - c^2.$$

Thus, we obtain the system of equations:

$$p(1-c) = \frac{m_1}{\sigma}\sqrt{\frac{\pi}{2}} - c \quad (1)$$

and

$$p(1-c^2) = p(1-c)(1+c) = \frac{m_2}{\sigma^2} - c^2 \quad (2).$$

For convenience, let:

$$a = \frac{m_1}{\sigma}\sqrt{\frac{\pi}{2}}, \quad b = \frac{m_2}{\sigma^2}.$$

Now, dividing equation (2) by equation (1), we get:

$$1 + c = \frac{b - c^2}{a - c}.$$

Multiplying both sides by $a - c$, we have:

$$(1+c)(a-c) = b - c^2.$$

Expanding both sides:

$$a + ac - c - c^2 = b - c^2.$$

Simplifying:

$$c = \frac{b-a}{a-1}.$$

Substitute this into equation (1), we get:

$$p = \frac{a-c}{1-c} = \frac{a - \frac{b-a}{a-1}}{1 - \frac{b-a}{a-1}}.$$

Multiplying both the numerator and denominator by $a-1$, we obtain:

$$p = \frac{a^2 - b}{2a - b - 1}.$$

Substituting a and b back into the equations, we obtain:

$$\hat{c} = \frac{\frac{\hat{m_2}}{\sigma^2} - \frac{\hat{m_1}}{\sigma}\sqrt{\frac{\pi}{2}}}{\frac{\hat{m_1}}{\sigma}\sqrt{\frac{\pi}{2}} - 1}$$

and

$$\hat{p} = \frac{\left(\frac{\hat{m_1}}{\sigma}\right)^2\left(\frac{\pi}{2}\right) - \frac{\hat{m_2}}{\sigma^2}}{\frac{\hat{m_1}\sqrt{2\pi}}{\sigma} - \frac{\hat{m_2}}{\sigma^2} - 1}.$$

(c) Let $\mu = 0.1$, $p = 0.3$, $\sigma = 1$ and $c = 1.3$. Simulate $n = 500$ independent samples from the above mixture model and compute the point estimators from part **(b)**. Repeat this simulation $N = 1000$ times and produce a table with the empirical 95% confidence intervals for μ, p and c.

Hint: You can do so by modifying the following code, for example,

```
n=500
iter = 1000
mu = 0.1
sig = 1
m = c()
s = c()
c_array = c()
p_array = c()
c = 1.3
p = 0.3

for (i in c(1:iter)){
 U = runif(n)
 Z = rnorm(n)

 x = (mu + sig * Z) * (U < p) +
     (mu + c * sig * Z) * (U >= p)

 m = c(m, mean(x))

 m1 = sum(abs(x-mean(x)))/n
 m2 = sum((x-mean(x))**2)/n
```

```
 A = (m1/sig) * (sqrt(pi/2))
 B = (m2) / (sig**2)

 c_array = c(c_array, (B-A)/(A-1))
 p_array = c(p_array, (A**2 - B)/ (2*A - B -1))
}

q.m = quantile(m,probs = c(0.025,0.975))
q.p = quantile(p_array,probs = c(0.025,0.975))
q.c = quantile(c_array,probs = c(0.025,0.975))
X = matrix(c(q.m,q.p,q.c),nrow=3,ncol=2,byrow = T)
dimnames(X) = list(c("mu","p",'c'),c("2.5 percentile",
                                     "97.5 percentile"))

kable(X,digits = 4)
```

	2.5 percentile	97.5 percentile
mu	-0.0078	0.2044
p	-3.3279	0.8510
c	1.0024	1.6990

2. Consider the t-distributed random variable $T = Z/\sqrt{Y/\nu}$, where $Z \sim \mathcal{N}(0,1)$ and $Y \sim \text{Gamma}(\nu/2, 1/2)$, $\nu > 0$ are two independent random variables.

   (a) Assuming that $\nu > 4$, compute the kurtosis of T, that is

   $$\text{kurt}(T) = \mathbb{E}\left(\frac{(T-\mu)^4}{\sigma^4}\right),$$

   in terms of ν, where $\mu = \mathbb{E}(T)$ and $\sigma^2 = \text{Var}(T)$.

   We know that the kurtosis of T is given by:

   $$\text{kurt}(T) = \mathbb{E}\left[\frac{(T-\mu)^4}{\sigma^4}\right]$$

   Expanding the expression inside the expectation:

   $$= \mathbb{E}\left[\frac{T^4 - 4\mu T^3 + 6\mu^2 T^2 - 4\mu^3 T + \mu^4}{\sigma^4}\right]$$

Since T is symmetric, we have $\mathbb{E}[T] = \mu = 0$, simplifying the equation to:

$$\text{kurt}(T) = \mathbb{E}\left[\frac{T^4}{\sigma^4}\right] = \frac{1}{\sigma^4}\mathbb{E}[T^4] = \frac{1}{\sigma^4}\mathbb{E}[Z^4]\mathbb{E}\left[\frac{\nu^2}{Y^2}\right]$$

From the lecture slides, we know:

$$\sigma^2 = \mathbb{E}[T^2] = \frac{\nu}{\nu - 2}$$

Also, we have:

$$\mathbb{E}[Z^4] = 3, \quad \sigma^2 = \frac{\nu}{\nu - 2}$$

Thus, the kurtosis simplifies to:

$$\text{kurt}(T) = \frac{3}{\sigma^4}\nu^2\mathbb{E}[Y^{-2}]$$

From properties of the Gamma function:

$$\mathbb{E}[Y^{-2}] = \left(\frac{1}{2}\right)^2 \frac{\Gamma(\nu/2 - 2)}{\Gamma(\nu/2)}$$

which simplifies further:

$$= \frac{3\nu^2}{\sigma^4} \times \left(\frac{1}{4} \times \frac{1}{(\nu/2 - 2)(\nu/2 - 1)}\right)$$

$$= \frac{3\nu^2}{\sigma^4} \times \frac{1}{(\nu - 4)(\nu - 2)}$$

$$= 3\nu^2 \times \left(\frac{(\nu - 2)^2}{\nu^2}\right) \times \frac{1}{(\nu - 4)(\nu - 2)}$$

$$= \frac{3(\nu - 2)}{\nu - 4}$$

(b) Suppose that $T_1, \ldots, T_n$ are independent realizations form the above t-distribution model. Estimate ν if the sample kurtosis

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{T_i - \overline{T}_n}{s_n}\right)^4 = 9,$$

where $s_n^2 = n^{-1}\sum_{i=1}^{n}(T_i - \overline{T}_n)^2$.

By the Weak Law of Large Numbers, we know:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{T_i - \bar{T}_n}{s_n} \to \text{kurt}(T)$$

Therefore, we can estimate ν using:

$$\frac{3(\nu-2)}{\nu-4} = 9$$

Solving for ν:

$$3\nu - 6 = 9\nu - 36$$

$$-6 + 36 = 9\nu - 3\nu, \text{therefore } \nu = 5$$

Thus, the estimated value of ν is 5.

3. Let $p_1 = 0.2, p_2 = 0.5$ and $p_3 = 0.3$, consider the mixture density

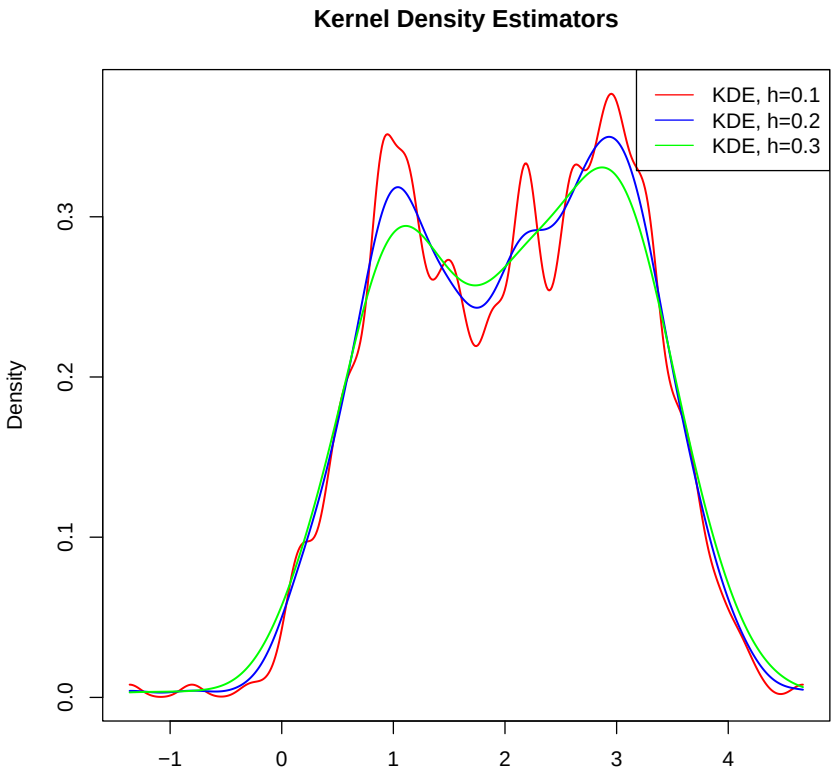
$$f(x) = p_1 f_1(x) + p_2 f_2(x) + p_3 f_3(x),$$

where f_i are $\mathcal{N}(\mu_i, \sigma_i^2),\ i = 1, 2, 3$ densities.

(a) Set $\mu_i = i,\ i = 1, 2, 3$, $\sigma_2 = 1$ and $\sigma_1 = \sigma_3 = 0.5$. Simulate $n = 500$ points from this mixture distribution and estimate the density $f(x)$ using a kernel density estimator with bandwidth h, for three values of h. Plot on the same graph the resulting KDEs.

```
n = 500
mu_1 = 1
mu_2 = 2
mu_3 = 3
```

```
sigma_1 = sigma_3 = 0.5
sigma_2 = 1
p_1 = 0.2
p_2 = 0.5
p_3 = 0.3

U = runif(n)
Z = rnorm(n)

X = (mu_1 + sigma_1 * Z) * (U < p_1) +
    (mu_2 + sigma_2 * Z) * (U >= p_1) * (U < p_1 + p_2) +
    (mu_3 + sigma_3 * Z) * (U >= p_1 + p_2)

h = c(0.1, 0.2, 0.3)

K1 = c()
K2 = c()
K3 = c()
grid = seq(from = min(X), to = max(X), length.out = n)
for (i in c(1:n)) K1 = rbind(K1, dnorm(grid - X[i], sd = h[1]))
for (i in c(1:n)) K2 = rbind(K2, dnorm(grid - X[i], sd = h[2]))
for (i in c(1:n)) K3 = rbind(K3, dnorm(grid - X[i], sd = h[3]))

f.hat1 = colMeans(K1)
f.hat2 = colMeans(K2)
f.hat3 = colMeans(K3)

par(mfrow=c(1,1))
plot(grid, f.hat1, type = 'l', col='red',xlab = "",ylab = "Density",
     main = "Kernel Density Estimators", lwd=1.5)
lines(grid, f.hat2, col='blue', lwd=1.5)
lines(grid, f.hat3, col='green', lwd=1.5)
legend('topright', legend= paste0("KDE, h=",h),
       lwd=c(1,1,1), col=c('red','blue','green'))
```



(b) Knowing the true density f, write an R-function that takes as inputs the simulated data and the bandwidth h and computes exactly (up to numerical precision) the quantity

$$ISE(h) := \int_{\mathbb{R}} (\widehat{f}_h(x) - f(x))^2 dx.$$

Note: You will have to calculate integrals of the form

$$\int_{\mathbb{R}} K_h(x - X_i)^2 dx \quad \int_{\mathbb{R}} K_h(x - X_i) f(x) dx \quad \text{and} \quad \int_{\mathbb{R}} f(x)^2 dx$$

Using the fact that K is a Gaussian kernel and f is a mixture of Gaussians, these integrals can be computed exactly. You should first derive exact formulae and then use them to write an R function that computes ISE(h), where the data as well as the bandwidth parameter are

inputs to that function.

$$\text{ISE} = \int_{\mathbb{R}} \left(\hat{f}_h(x) - f(x)\right)^2 \, dx = \int_{\mathbb{R}} \hat{f}_h(x)^2 \, dx - 2\int_{\mathbb{R}} \hat{f}_h(x) f(x) \, dx + \int_{\mathbb{R}} f(x)^2 \, dx$$

where

$$\hat{f}_h(x) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} e^{\frac{-(x_i - x)^2}{2h^2}}$$

Part 1:

$$\int_{\mathbb{R}} \hat{f}_h(x)^2 = \frac{1}{2n^2h^2\pi} \int_{\mathbb{R}} \left(\sum_{i=1}^{n} \exp\left(\frac{-(x_i - x)^2}{2h^2}\right)\right)^2 \, dx$$

$$= \frac{1}{2n^2h^2\pi} \int_{\mathbb{R}} \sum_{i=1}^{n} \exp\left(\frac{-(x_i - x)^2}{2h^2}\right) \sum_{j=1}^{n} \exp\left(\frac{-(x_j - x)^2}{2h^2}\right) \, dx$$

$$= \frac{1}{2n^2h^2\pi} \sum_{i=1}^{n}\sum_{j=1}^{n} \int_{\mathbb{R}} \exp\left(\frac{-(x_i - x)^2}{2h^2}\right) \exp\left(\frac{-(x_j - x)^2}{2h^2}\right) \, dx$$

$$= \frac{1}{2n^2h^2\pi} \sum_{i=1}^{n}\sum_{j=1}^{n} \int_{\mathbb{R}} \exp\left(\frac{-(x - x_i)^2 - (x - x_j)^2}{2h^2}\right) \, dx$$

$$= \frac{1}{2n^2h^2\pi} \sum_{i=1}^{n}\sum_{j=1}^{n} \int_{\mathbb{R}} \left(-\frac{1}{2h^2}\left(2x^2 - 2x(x_i + x_j) + (x_i^2 + x_j^2)\right)\right) dx$$

Now we focus on

$$-\frac{1}{2h^2}\left(2x^2 - 2x(x_i + x_j) + (x_i^2 + x_j^2)\right)$$

$$= -\frac{x^2}{h^2} + \frac{x(x_i + x_j)}{h^2} - \frac{x_i^2}{2h^2} - \frac{x_j^2}{2h^2}$$

$$= -\frac{1}{h^2}\left[x^2 - x(x_i + x_j)\right] - \frac{x_i^2}{2h^2} - \frac{x_j^2}{2h^2}$$

$$= -\frac{1}{h^2}\left\{\left[x - \frac{x_i + x_j}{2}\right]^2 - \frac{(x_i + x_j)^2}{4}\right\} - \frac{x_i^2}{2h^2} - \frac{x_j^2}{2h^2}$$

$$= -\frac{1}{h^2}\left[x - \frac{x_i + x_j}{2}\right]^2 + \frac{(x_i + x_j)^2}{4h^2} - \frac{x_i^2}{2h^2} - \frac{x_j^2}{2h^2}$$

$$= -\frac{1}{h^2}\left[x - \frac{x_i + x_j}{2}\right]^2 - \frac{(x_i - x_j)^2}{4h^2}$$

Now, consider the following expression:

$$\frac{1}{2n^2h^2\pi}\sum_{i=1}^{n}\sum_{j=1}^{n}\int \exp\left(-\frac{(x - (x_i + x_j))^2}{h^2}\right)dx \cdot \exp\left(-\frac{(x_i - x_j)^2}{4h^2}\right)$$

Since the inner integral is a Gaussian integral, we use the following standard Gaussian result:

$$\int \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx = 1$$

For $2\sigma^2 = h^2$, the integral simplifies as: $\sqrt{\pi h^2}$

The original expression becomes:

$$\frac{\sqrt{\pi h^2}}{2n^2h^2\pi}\sum_{i=1}^{n}\sum_{j=1}^{n}\exp\left(-\frac{(x_i - x_j)^2}{4h^2}\right) = \frac{1}{2n^2h\sqrt{\pi}}\sum_{i=1}^{n}\sum_{j=1}^{n}e^{-\frac{(x_i - x_j)^2}{4h^2}}$$

Part 2:

$$\int f(x)^2\,dx = \int\left[\sum_{j=1}^{3} p_j\left(\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp\left(\frac{(x-\mu_j)^2}{2\sigma_j^2}\right)\right)\right]^2 dx$$

$$= \sum_{i=1}^{3}\sum_{j=1}^{3} \int \left(\frac{p_i p_j}{2\pi\sigma_i\sigma_j} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2} - \frac{(x-\mu_j)^2}{2\sigma_j^2} \right) \right) dx$$

$$= \sum_{i=1}^{3}\sum_{j=1}^{3} \int \left(\frac{p_i p_j}{2\pi\sigma_i\sigma_j} \exp\left(-\frac{x^2 - 2x\mu_i + \mu_i^2}{2\sigma_i^2} - \frac{x^2 - 2x\mu_j + \mu_j^2}{2\sigma_j^2} \right) \right) dx$$

$$= \sum_{i=1}^{3}\sum_{j=1}^{3} \left(\frac{p_i p_j}{2\pi\sigma_i\sigma_j} \right) \int \exp\left(-\frac{x^2}{2}\left(\frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2} \right) - x\left(\frac{\mu_i}{\sigma_i^2} + \frac{\mu_j}{\sigma_j^2} \right) - \left(\frac{\mu_i^2}{\sigma_i^2} + \frac{\mu_j^2}{\sigma_j^2} \right) \right) dx$$

Let

$$\sigma_{ij}^2 = \frac{\sigma_i^2\sigma_j^2}{\sigma_i^2 + \sigma_j^2}, \quad \text{then} \quad \frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2} = \frac{1}{\sigma_{ij}^2}$$

and

$$\mu_{ij} = \frac{\mu_i\sigma_j^2 + \mu_j\sigma_i^2}{\sigma_i^2 + \sigma_j^2}, \quad \text{then} \quad \frac{\mu_i}{\sigma_i^2} + \frac{\mu_j}{\sigma_j^2} = \frac{1}{\mu_{ij}}.$$

We define the following convenient variables:

$$A = \frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2}, \quad B = \frac{\mu_i}{\sigma_i^2} + \frac{\mu_j}{\sigma_j^2}, \quad C = -\frac{\mu_i^2}{\sigma_i^2} - \frac{\mu_j^2}{\sigma_j^2}$$

Now, let's simplify the expression:

$$-\frac{x^2}{2}A - Bx = -\frac{A}{2}\left(x^2 - \frac{2B}{A}x \right) = -\frac{A}{2}\left[x - \frac{B}{A} \right]^2 + \frac{B^2}{2A}$$

Now, we define $\frac{B}{A}$ as:

$$\frac{B}{A} = \frac{\sigma_i^2\sigma_j^2}{\mu_i\sigma_j^2 + \mu_j\sigma_i^2} = \mu_{ij}$$

Thus, the expression becomes:

$$-\frac{1}{2}\sigma_{ij}^2\left(x-\mu_{ij}\right)^2+\frac{B^2}{2A}$$

$$B^2=\frac{\mu_i^2}{\sigma_i^4}+\frac{2\mu_i\mu_j}{\sigma_i^2\sigma_j^2}+\frac{\mu_j^2}{\sigma_j^4}$$

$$\frac{B^2}{2A}=\frac{(\frac{\mu_i^2}{\sigma_i^4}+\frac{2\mu_i\mu_j}{\sigma_i^2\sigma_j^2}+\frac{\mu_j^2}{\sigma_j^4})*(\sigma_i^2\sigma_j^2)}{2\left(\frac{1}{\sigma_i^2}+\frac{1}{\sigma_j^2}\right)*(\sigma_i^2\sigma_j^2)}$$

$$\frac{B^2}{2A}=\frac{\mu_i^2\frac{\sigma_j^2}{\sigma_i^2}+2\mu_i\mu_j+\mu_j^2\frac{\sigma_i^2}{\sigma_j^2}}{2(\sigma_i^2+\sigma_j^2)}$$

$$\frac{B^2}{2A}=\frac{(\mu_i\sigma_j^2+\mu_j\sigma_i^2)^2}{2(\sigma_i^2\sigma_j^2)(\sigma_i^2+\sigma_j^2)}$$

$$=\frac{(\mu_i\sigma_j^2+\mu_j\sigma_i^2)^2}{2(\sigma_i^2+\sigma_j^2)\sigma_i^2\sigma_j^2}$$

Now, considering μ_{ij}^2, it becomes:

$$=\frac{\mu_{ij}(\sigma_i^2+\sigma_j^2)}{2\sigma_i^2\sigma_j^2}=\frac{1}{2}\cdot\frac{\left(\frac{\mu_i\sigma_j^2+\mu_j\sigma_i^2}{\sigma_i^2+\sigma_j^2}\right)^2}{\frac{\sigma_i^2\sigma_j^2}{\sigma_i^2+\sigma_j^2}}$$

$$=\frac{1}{2}\left(\frac{(\mu_i\sigma_j^2+\mu_j\sigma_i^2)^2}{\sigma_i^2\sigma_j^2(\sigma_i^2+\sigma_j^2)}\right)$$

Now consider

$$C=-\frac{\mu_i^2}{\sigma_i^2}-\frac{\mu_j^2}{\sigma_j^2}=-\frac{\mu_i^2\sigma_j^2+\mu_j^2\sigma_i^2}{2\sigma_i^2\sigma_j^2}$$

Then,

$$\frac{B^2}{2A} + C = -\left[\frac{\mu_i^2\sigma_j^2 + \mu_j^2\sigma_i^2}{2\sigma_i^2\sigma_j^2} - \frac{(\mu_i\sigma_j^2 + \mu_j\sigma_i^2)^2}{2\sigma_i^2\sigma_j^2(\sigma_i^2+\sigma_j^2)}\right]$$

$$= -\frac{1}{2\sigma_i^2\sigma_j^2(\sigma_i^2+\sigma_j^2)}\left[\mu_i^2\sigma_i^2\sigma_j^2 + \mu_i^2\sigma_j^4 + \mu_j^2\sigma_i^4 + \mu_i^2\sigma_i^2\sigma_j^2 - \mu_i^2\sigma_j^4 - 2\mu_i\mu_j\sigma_i^2\sigma_j^2 - \mu_j^2\sigma_i^4\right]$$

$$= -\frac{1}{2\sigma_i^2\sigma_j^2(\sigma_i^2+\sigma_j^2)}\left[\mu_i^2\sigma_i^2\sigma_j^2 + \mu_j^2\sigma_i^2\sigma_j^2 - 2\mu_i\mu_j\sigma_i^2\sigma_j^2\right]$$

$$= -\frac{1}{2\sigma_i^2\sigma_j^2(\sigma_i^2+\sigma_j^2)}\left(\sigma_i^2\sigma_j^2\right)\left[\mu_i^2 - 2\mu_i\mu_j + \mu_j^2\right]$$

$$= -\frac{(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}$$

Therefore, $\int f(x)^2 = \sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{p_ip_j}{2\pi\sigma_i\sigma_j}\right)\int \exp\left[-\frac{(x-\mu_{ij})^2}{2\sigma_{ij}} - \frac{(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}\right]dx$

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{p_ip_j}{2\pi\sigma_i\sigma_j}\right)e^{\frac{-(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}}\cdot\sqrt{2\pi\sigma_{ij}^2}$$

$$= \sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{p_ip_j}{\sqrt{2\pi(\sigma_i^2+\sigma_j^2)}}\right)e^{\frac{-(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}}$$

Part 3:

$$2\int f_{\hat{h}}(x)f(x)\,dx = 2\int\left[\frac{1}{nh\sqrt{2\pi}}\sum_{i=1}^{n}\exp\left(-\frac{(x-x_i)^2}{2h^2}\right)\right]\left[\sum_{j=1}^{3}\frac{p_j}{\sqrt{2\pi\sigma_j^2}}\exp\left(-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right)\right]dx$$

$$= 2\sum_{i=1}^{n}\sum_{j=1}^{3}\frac{p_j}{2nh\pi\sigma_j}\int e^{-\frac{(x-x_i)^2}{2h^2}}e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}\,dx$$

Based on Part 2, we know the product of two Gaussians is also Gaussian. The expression becomes:

$$= 2\sum_{i=1}^{n}\sum_{j=1}^{3}\frac{p_j}{2nh\pi\sigma_j}\sqrt{2\pi\frac{h^2\sigma_j^2}{h^2+\sigma_j^2}}e^{-\frac{(x_i-\mu_j)^2}{2(h^2+\sigma_j^2)}}$$

$$2\sum_{i=1}^{n}\sum_{j=1}^{3}\frac{p_j}{\sqrt{2\pi}nh}\sqrt{\frac{h^2}{h^2+\sigma_j^2}}e^{-\frac{(x-\mu_j)^2}{2(h^2+\sigma_j^2)}}$$

$$= 2\sum_{i=1}^{n}\sum_{j=1}^{3}\frac{p_j}{n\sqrt{2\pi(h^2+\sigma_j^2)}}e^{-\frac{(x-\mu_j)^2}{2(h^2+\sigma_j^2)}}$$

Tehrefore, ISE =

$$\frac{1}{2n^2h\sqrt{\pi}}\sum_{i=1}^{n}\sum_{j=1}^{n}e^{-\frac{(x_i-x_j)^2}{4h^2}}$$

$$-2\sum_{i=1}^{n}\sum_{j=1}^{3}\frac{p_j}{n\sqrt{2\pi(h^2+\sigma_j^2)}}e^{-\frac{(x-\mu_j)^2}{2(h^2+\sigma_j^2)}}$$

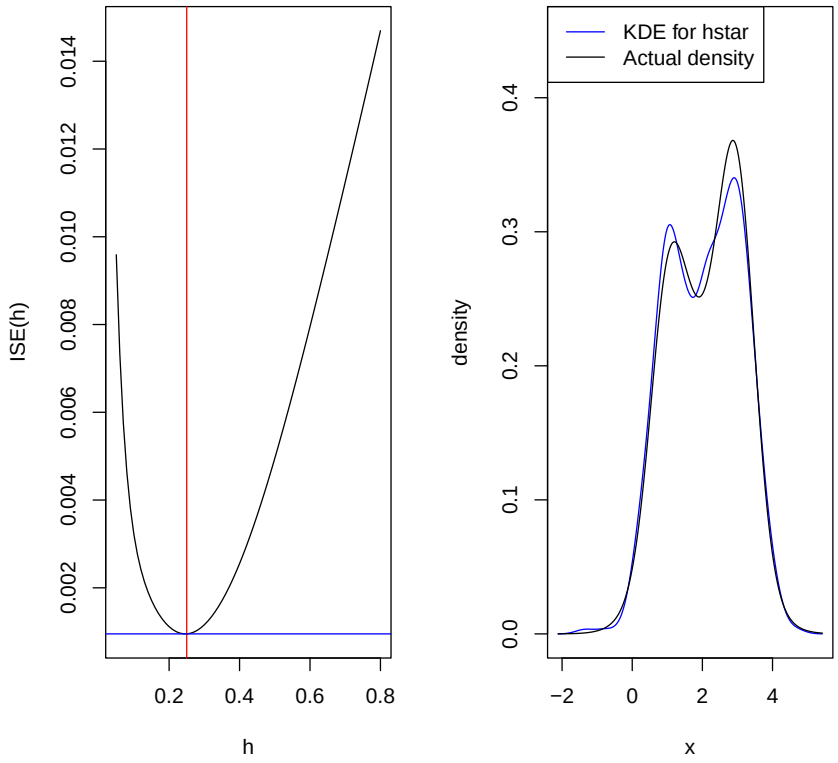
$$+\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{p_ip_j}{\sqrt{2\pi(\sigma_i^2+\sigma_j^2)}}\right)e^{-\frac{(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}}$$

```
p = c(p_1, p_2, p_3)
mu = c(mu_1, mu_2, mu_3)
sig = c(sigma_1, sigma_2, sigma_3)

ISE <- function(h, data=X){
  n = length(X)
  A = B = C = 0
  for(i in 1:n){
    for(j in 1:n){
      A = A + 1/(2* n**2 * h * sqrt(pi)) * exp(-(X[i]-X[j])**2 / (4 * h**2))
    }
  }

  for(i in 1:n){
    for(j in 1:3){
      B = B - 2 * p[j] / (n * sqrt(2 * pi) * sqrt(h**2 + sig[j]**2))*
```

```
          exp(-(X[i] - mu[j])**2 / (2 * (sig[j]**2 + h**2)) )
    }
  }

  for(i in 1:3){
    for(j in 1:3){
      C = C + p[i] * p[j] / (sqrt(2 * pi) * sqrt(sig[i]**2 + sig[j]**2)) *
            exp(-(mu[i] - mu[j])**2 / (2 * (sig[i]**2 + sig[j]**2)))
    }
  }

  return(A+B+C)
}
```

(c) Plot the quantity $ISE(h)$ as a function of h and determine visually the best value of $h = h^*$. Plot the resulting KDE $\widehat{f}_{h^*}(x)$ as a function of x.

```
ISE_list = c()
h_val = (5:80)/100

for (h in h_val){
  ISE_list = c(ISE_list, ISE(h))
}

par(mfrow=c(1,2))
plot(h_val, ISE_list, type='l', xlab='h', ylab='ISE(h)')
h_star = h_val[which.min(ISE_list)]
abline(v = h_star, col='red')
abline(h = min(ISE_list), col='blue')

density = density(X, bw=h_star, kernel = 'gaussian')
plot(density$x, density$y, type='l', col='blue', ylim=c(0,0.45),
     xlab='x', ylab='density')
lines(density$x, (p_1*dnorm(density$x, mean=mu_1, sd=sigma_1)
                + p_2*dnorm(density$x, mean=mu_2, sd=sigma_2)
                + p_3*dnorm(density$x, mean=mu_3, sd=sigma_3)))
legend("topleft", legend = c("KDE for hstar","Actual density"),
       col = c("blue","black"), lwd = c(1,1))
```




```
cat("h* = ", h_star)
```

```
## h* =  0.25
```

4. Load the data set `sp500_full.cvs` found in the `data` folder on Canvas.

   (a) Using maximum likelihood, fit the skewed t-distribution discussed in class to the daily returns (variable `sprtrn`) of the sp500 time series. Namely, consider the range of years 1962, 1963, ..., 2023. For the samples of daily returns corresponding to each year y in this range, obtain $\hat{\theta}_{MLE}(y),\ y = 1962, \ldots, 2023$. Do so by using the `optim` function in R in two differnt ways (1) using the method "BFGS" and (2) via "Nelder-Mead". Compare the results and comment on the stability of each of the optimization methods.

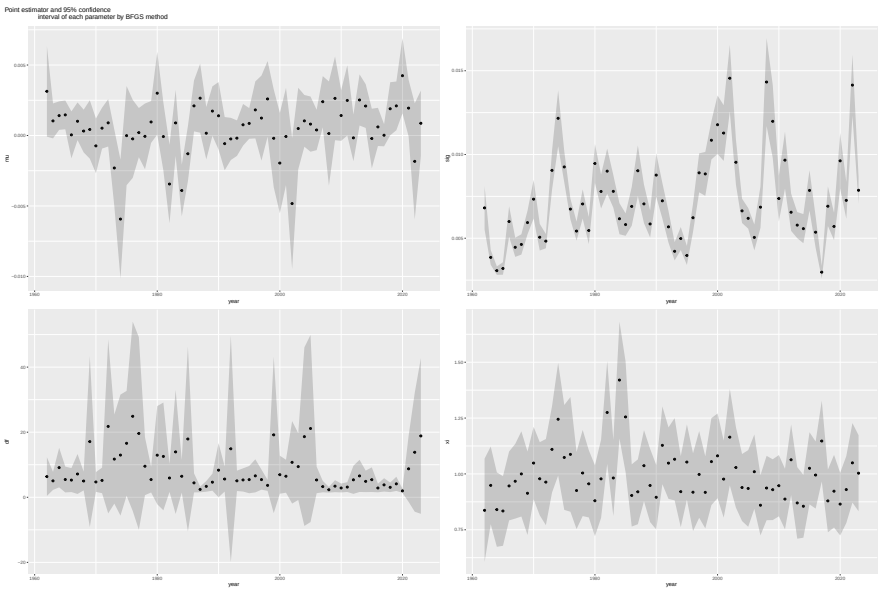
Finally, using the Hessian estimate from the optimization routine, for each of the 4 parameters μ, σ, ν, and ξ, produce a plot of the point estimate along with point-wise 95% confidence intervals as a function of time (years).

```
ginv <- function(A){ # Computes the Moore-Penrose
  out = svd(A)       # generalized inverse.
  d = out$d; d[d>0] = 1/d[d>0]; d[d<0]=0;
  return(out$v %*% diag(d) %*% t(out$u) )
}

data = read.csv("sp500_full.csv",header=TRUE)

data$caldt <- as.Date(as.character(data$caldt), format = "%Y%m%d")

# Filter dates between 1962-01-01 and 2023-12-31
data <- subset(data,
  caldt >= as.Date("1962-01-01") & caldt <= as.Date("2023-12-31"))

# Remove NA value
data <- data %>% filter(!is.na(sprtrn))

# Extract daily returns and years
data$year <- format(data$caldt, "%Y")

# Defining the skew t density
dskew.t = function(x, xi=1, df=5) {
  (dt(x * xi^(-sign(x)), df=df) * 2 * xi / (1 + xi^2))}

nlog_lik_skew_t = function(theta) {
  mu = theta[1]
  sigma = theta[2]
  df = theta[3]
  xi = theta[4]
  x = year_data$sprtrn

  -sum(log(dskew.t((x - mu) / sigma, df = df, xi = xi)) - log(sigma))
}
```

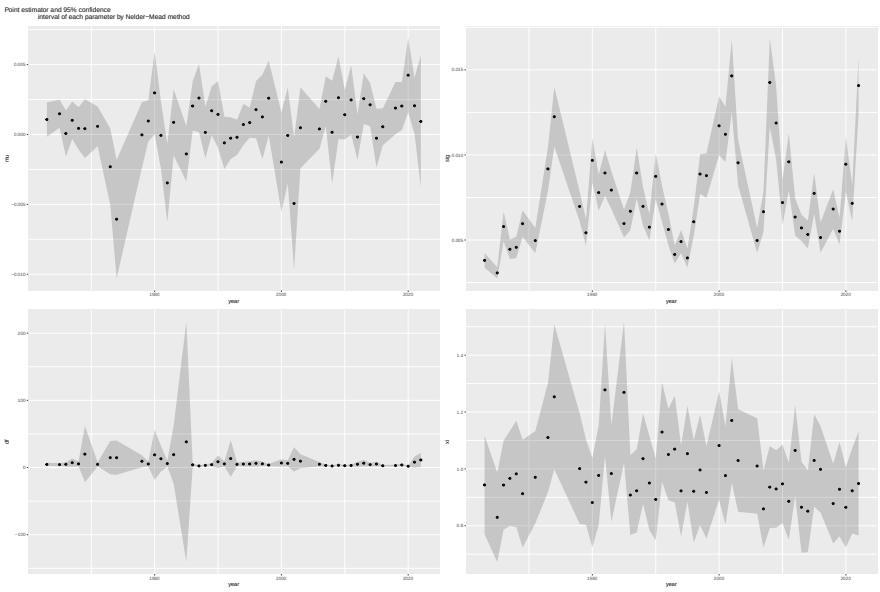
```
years = seq(1962,2023)
result_bfgs = c()
result_nm = c()

for (year in years){
  year_data = data[data$year == year, ]

  start = c(mean(year_data$sprtrn), sd(year_data$sprtrn), 5, 1)
  fit_bfgs = suppressWarnings(optim(start, nlog_lik_skew_t,
                                    hessian=T, method="BFGS"))
  fit_nm = suppressWarnings(optim(start, nlog_lik_skew_t,
                                  hessian=T, method="Nelder-Mead"))

  if (fit_bfgs$convergence == 0){
   Sig_bfgs = ginv(fit_bfgs$hessian)
   se_bfgs = sqrt(diag(Sig_bfgs))
   result_bfgs = rbind(result_bfgs, c(year, fit_bfgs$par,
                                  fit_bfgs$par -1.96*se_bfgs,
                                  fit_bfgs$par +1.96*se_bfgs,
                                  se_bfgs))
  }

  if (fit_nm$convergence == 0){

   Sig_nm = ginv(fit_nm$hessian)
   se_nm = sqrt(diag(Sig_nm))
   result_nm = rbind(result_nm,c(year, fit_nm$par,
                                fit_nm$par -1.96*se_nm,
                                fit_nm$par +1.96*se_nm,
                                se_nm))
  }
}

col_names = c("year", "mu", "sig", "df", "xi",
              "mu_lb", "sig_lb", "df_lb", "xi_lb",
              "mu_ub", "sig_ub", "df_ub", "xi_ub",
              "mu_se", "sig_se", "df_se", "xi_se")

colnames(result_bfgs) = col_names
```

```
colnames(result_nm) = col_names

rs_etm = rbind(cbind(result_bfgs[, "mu"], "mu", "BFGS"),
               cbind(result_bfgs[, "sig"], "sig", "BFGS"),
               cbind(result_bfgs[, "df"], "df", "BFGS"),
               cbind(result_bfgs[, "xi"], "xi", "BFGS"),

               cbind(result_nm[, "mu"], "mu", "NM"),
               cbind(result_nm[, "sig"], "sig", "NM"),
               cbind(result_nm[, "df"], "df", "NM"),
               cbind(result_nm[, "xi"], "xi", "NM"))

colnames(rs_etm) = c("value", "parameter", "method")
rs_etm = data.frame(rs_etm)
rs_etm$value = as.numeric(rs_etm$value)
```

```
p1_bfgs = ggplot(result_bfgs, aes(x = year, y = mu)) +
  geom_point() +
  geom_ribbon(aes(ymin = mu_lb, ymax = mu_ub), alpha = 0.2)

p2_bfgs = ggplot(result_bfgs, aes(x = year, y = sig)) +
  geom_point() +
  geom_ribbon(aes(ymin = sig_lb, ymax = sig_ub), alpha = 0.2)

p3_bfgs = ggplot(result_bfgs, aes(x = year, y = df)) +
  geom_point() +
  geom_ribbon(aes(ymin = df_lb, ymax = df_ub), alpha = 0.2)

p4_bfgs = ggplot(result_bfgs, aes(x = year, y = xi)) +
  geom_point() +
  geom_ribbon(aes(ymin = xi_lb, ymax = xi_ub), alpha = 0.2)

p1_bfgs + p2_bfgs + p3_bfgs + p4_bfgs +
  plot_annotation("Point estimator and 95% confidence
                   interval of each parameter by BFGS method")
```




```
p1_nm = ggplot(result_nm, aes(x = year, y = mu)) +
  geom_point() +
  geom_ribbon(aes(ymin = mu_lb, ymax = mu_ub), alpha = 0.2)

p2_nm = ggplot(result_nm, aes(x = year, y = sig)) +
  geom_point() +
  geom_ribbon(aes(ymin = sig_lb, ymax = sig_ub), alpha = 0.2)

p3_nm = ggplot(result_nm, aes(x = year, y = df)) +
  geom_point() +
  geom_ribbon(aes(ymin = df_lb, ymax = df_ub), alpha = 0.2)

p4_nm = ggplot(result_nm, aes(x = year, y = xi)) +
  geom_point() +
  geom_ribbon(aes(ymin = xi_lb, ymax = xi_ub), alpha = 0.2)

p1_nm + p2_nm + p3_nm + p4_nm +
  plot_annotation("Point estimator and 95% confidence
                   interval of each parameter by Nelder-Mead method")
```

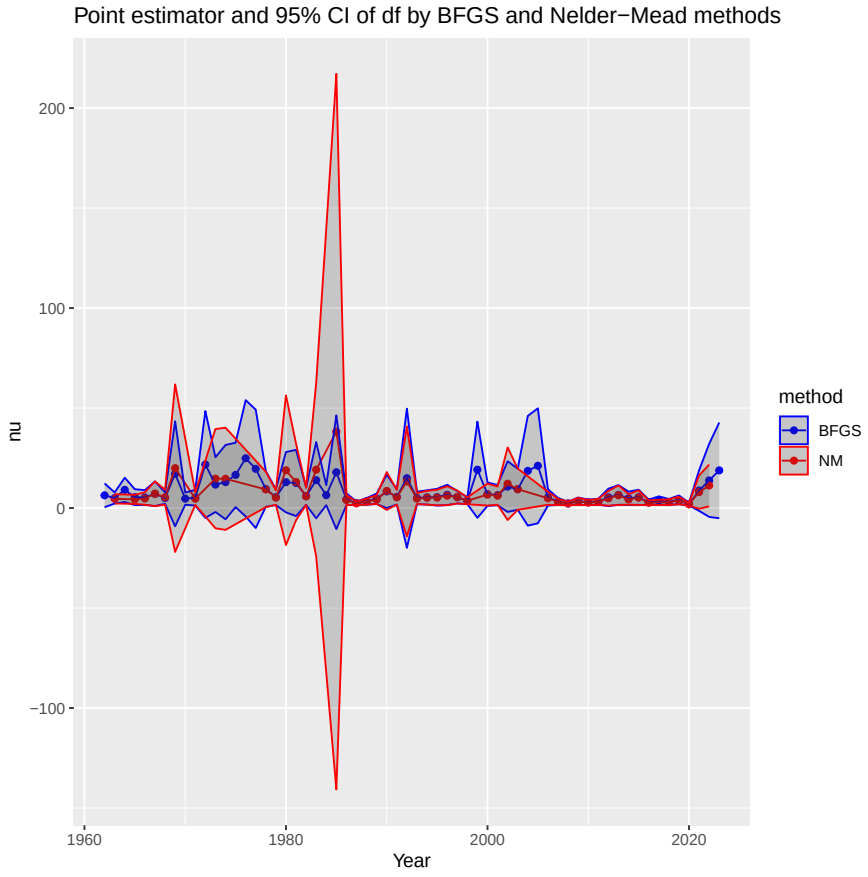


The BFGS method appears to be the more stable and reliable approach for estimating the skewed t-distribution parameters. While Nelder-Mead can still provide reasonable estimates, its instability in df and wider confidence intervals suggest that it may not be the best choice for this specific optimization task.

(b) Plot the estimated degrees of freedom parameter $\hat{\nu}(y)$ as a function of time y (in years) from part (a). Which years y seem to have relatively heavier tailed returns? Are there any years for which the estimated model has infinite variance? Explain why and plot the time series of the daily returns for one of these years (if any).

```
combined_df = data.frame(
  year = c(result_bfgs[, 'year'], result_nm[, 'year']),
  df = c(result_bfgs[, 'df'], result_nm[, 'df']),
  df_lb = c(result_bfgs[, 'df_lb'], result_nm[, 'df_lb']),
  df_ub = c(result_bfgs[, 'df_ub'], result_nm[, 'df_ub']),
  method = rep(c("BFGS", "NM"), c(nrow(result_bfgs), nrow(result_nm)))
)
```

```r
# Plot both datasets in the same plot, with different colors for each method
ggplot(combined_df, aes(x = year, y = df, color = method)) +
  geom_point() +
  geom_line() +
  geom_ribbon(aes(ymin = df_lb, ymax = df_ub), alpha = 0.2) +
  scale_color_manual(values = c("BFGS" = "blue", "NM" = "red")) +
  labs(x = "Year", y = 'nu',
       title = "Point estimator and 95% CI of df by BFGS and Nelder-Mead methods
```




```r
df_heavy <- combined_df %>%
  filter(2 <= df & df < 3)

df_infinite <- combined_df %>%
  filter(df <= 2)
```

Table 1: Heavier Tailed Distribution (df ¡ 3)

year	method	df
1987	BFGS	2.421986
2008	BFGS	2.318087
2010	BFGS	2.856913
2016	BFGS	2.854006
1987	NM	2.387973
2008	NM	2.302516
2010	NM	2.740828
2016	NM	2.656323
2018	NM	2.980023

Table 2: Years with Infinite Variance (df ¡= 2)

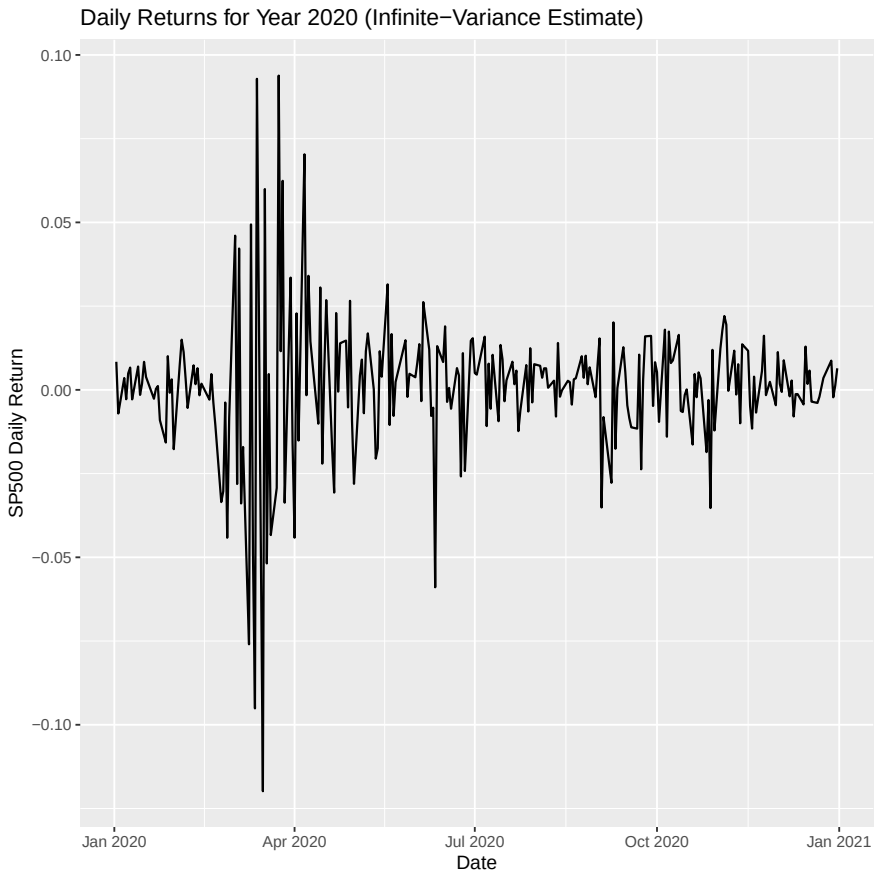
year	method	df
2020	BFGS	1.982608
2020	NM	1.913596

```
kable(
  df_heavy %>% select(year, method, df),
  caption = "Heavier Tailed Distribution (df < 3)",
  label="1")
```

```
kable(
  df_infinite %>% select(year, method, df),
  caption = "Years with Infinite Variance (df <= 2)",
  label="2")
```

```
infinite_year = df_infinite[df_infinite$df == min(df_infinite$df), ]$year
infinite_year_data = data[data$year == infinite_year, ]

# For my local system(My default language is Chines, which will yield error)
Sys.setlocale("LC_TIME", "English")

## [1] "English_United States.1252"
```

```
ggplot(infinite_year_data, aes(x = caldt, y = sprtrn)) +
    geom_line(linewidth = 0.6) +
    labs(
      title = "Daily Returns for Year 2020 (Infinite-Variance Estimate)",
      x = "Date",
      y = "SP500 Daily Return"
    )
```



From Table 1, 1987, 2008, 2010, and 2016 show heavy-tailed distributions (df ¡ 3), aligning with major financial crises: Black Monday (1987), Global Financial Crisis (2008), Flash Crash (2010), and geopolitical instability (2016)

From Table 2, 2020 has an estimated df near 2 (1.98 BFGS,

1.91 NM), suggesting infinite variance, consistent with extreme market volatility during the COVID-19 pandemic

(c) Consider the confidence intervals for the skewness parameter estimates obtained in part (a). Identify during which years y (if any) the skewness parameter ξ is likely to be significantly different from 1. Ignoring multiple testing issues, for each year y in the range, test the corresponding (two-sided) hypothesis at a level of 5%.

Plot the kernel density estimator for the daily returns for one of these "skewed" years and on the same plot display (in different line-style/color) the density of the estimated skewed t-model.

```
# Assuming result_bfgs and result_nm are matrices, convert them to data frames
result_bfgs <- as.data.frame(result_bfgs)
result_nm <- as.data.frame(result_nm)

# Add the 'method' column to each dataset to distinguish BFGS and NM
result_bfgs <- result_bfgs %>% mutate(method = "BFGS")
result_nm <- result_nm %>% mutate(method = "NM")

# Combine both datasets
combined_results <- bind_rows(result_bfgs, result_nm)

# Calculate z_value, p_value, and significance (sig)
combined_results <- combined_results %>%
  mutate(
    z_value = (xi - 1) / xi_se,  # z-value for hypothesis test
    p_value = 2 * pnorm(abs(z_value), lower.tail = FALSE),  # Two-sided p-value
    significant = (p_value < 0.05)  # Significance check (5% threshold)
  )

# Filter significant years where xi is likely not equal to 1
signif_years <- combined_results %>%
  filter(significant == TRUE) %>%
  select(year, method, xi, xi_se, z_value, p_value, sig)  # Only necessary colum

# Display the filtered table with significant years
kable(
```

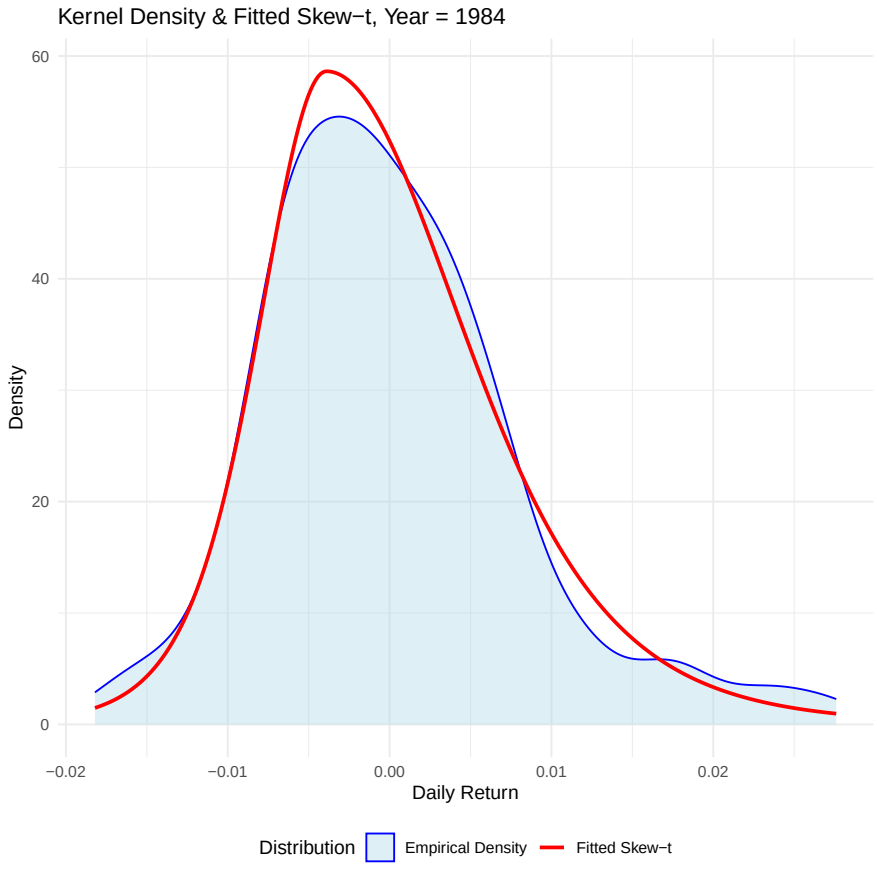
Table 3: Significant Skewed t-Distribution (xi != 1)

year	method	xi	xi_se	z_value	p_value	sig
1965	BFGS	0.8340890	0.0794831	-2.087376	0.0368542	0.0031957
1982	BFGS	1.2750694	0.1175220	2.340577	0.0192540	0.0090054
1984	BFGS	1.4201214	0.1334509	3.148134	0.0016432	0.0061639
1985	BFGS	1.2551719	0.1262844	2.020612	0.0433199	0.0058155
2007	BFGS	0.8597953	0.0688229	-2.037182	0.0416318	0.0068549
2014	BFGS	0.8552985	0.0717976	-2.015408	0.0438619	0.0055727
1965	NM	0.8295318	0.0802183	-2.125054	0.0335821	0.0030667
1982	NM	1.2778352	0.1187347	2.339966	0.0192855	0.0089388
1985	NM	1.2691888	0.1268660	2.121837	0.0338515	0.0059659
2007	NM	0.8593626	0.0693884	-2.026815	0.0426813	0.0066626
2014	NM	0.8512956	0.0733796	-2.026510	0.0427126	0.0053251

```
  signif_years,
  caption = "Significant Skewed t-Distribution (xi != 1)"
)
```

```
skew_year = signif_years[signif_years$xi == max(signif_years$xi), ]$year

skew_year_data = data[data$year == skew_year, ]


params_skew_year <- combined_results %>%
  filter(year == skew_year)

mu_hat    <- params_skew_year$mu
sigma_hat <- params_skew_year$sig
nu_hat    <- params_skew_year$df
xi_hat    <- params_skew_year$xi

# Define the PDF of the skewed t-distribution
dskew_t_full <- function(x, mu, sigma, nu, xi) {
  z <- (x - mu)/sigma
  val_std <- dskew.t(z, xi = xi, df = nu)
  val_full <- val_std / sigma
```

```
  return(val_full)
}

# Create a sequence of x values for plotting the skewed t-distribution
x_grid <- seq(min(skew_year_data$sprtrn),
              max(skew_year_data$sprtrn),
              length.out = 300)

# Compute the PDF of the skewed t-distribution
pdf_skew_t <- dskew_t_full(x_grid, mu_hat, sigma_hat, nu_hat, xi_hat)

# Create a data frame for the fitted skew-t PDF
df_plot <- data.frame(x = x_grid, fitted_pdf = pdf_skew_t)


# Plot with legend
ggplot(skew_year_data, aes(x = sprtrn)) +
  # Kernel density for the daily returns
  geom_density(aes(y = after_stat(density), color = "Empirical Density"),
               fill = "lightblue", alpha = 0.4) +
  # Fitted skew-t distribution
  geom_line(data = df_plot,
            aes(x = x, y = fitted_pdf, color = "Fitted Skew-t"),
            linewidth = 1) +
  # Customize colors and add legend
  scale_color_manual(values = c("Empirical Density" = "blue",
                                "Fitted Skew-t" = "red")) +
  labs(
    title = paste0("Kernel Density & Fitted Skew-t, Year = ", skew_year),
    x = "Daily Return",
    y = "Density",
    color = "Distribution"  # Legend title
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")  # Place legend at the bottom
```


Kernel Density & Fitted Skew−t, Year = 1984
60
40
20
0
Density
−0.02
−0.01
0.00
0.01
0.02
Daily Return
Distribution
Empirical Density
Fitted Skew−t