

MA677 Final Project

Kosuke Sasaki

2022/05/11

#Planning For this final project, first I started reading and tried to understand the relevant chapters in “In All Likelihood”. However, as I did not have any math background, I expected it to be really hard to understand the contents by myself, and it was, so I also asked my classmates, especially Yuli, to help me understand them as well as to give some tips to solve the problems.

The results and the deliverable of my work are shown below. It was really fun to understand and get to know what I didn’t know and solve these problems, and I also appreciate my classmates’ kind help!

4.25: Approximation of the median using the distribution of median statistics

Based on the Example 4.8 of the textbook “In All Likelihood”, the approximation of median of U_i is accurate. I will compare the values from the approximation with the value of expectation obtained from the distribution of the order statistics as below. [1]

```
#Create pdf and cdf of uniform distribution(0,1)
a <- 0
b <- 1
pdf <- function(x) dunif(x, a,b)
cdf <- function(x) punif(x, a,b, lower.tail=FALSE)

#Calculate the distribution of the order statistics
integrand <- function(x,r,n) {x * (1 - cdf(x))^(r-1) * cdf(x)^(n-r) * pdf(x)}

#Calculate the expectation from the above distribution
Expect <- function(r,n) {
  (1/beta(r,n-r+1)) * integrate(integrand,-Inf,Inf, r, n)$value
}

#Create approximation function from the text
Approx <- function(k,n){
  m<-(k-1/3)/(n+1/3)
  return(m)
}

#Create table to compare approximation with expectation for n=5 and n=10 respectively
Approximation5<- vector()
Expectation5 <- vector()
Approximation10<- vector()
Expectation10 <- vector()

for (i in 1:5) {
  Approximation5 <- c(Approximation5, Approx(i,5))
  Expectation5 <- c(Expectation5,Expect(i,5))
}
```

```

table1 <- data.frame(Approximation5, Expectation5)

for (i in 1:10) {
  Approximation10 <- c(Approximation10, Approx(i,10))
  Expectation10 <- c(Expectation10, Expect(i,10))
}
table2 <- data.frame(Approximation10, Expectation10)

kable(table1, caption = "Comparison of Approximation with Expectation (n=5)") %>%
  kable_styling(position = "center") %>%
  kable_styling(latex_options = "HOLD_position")

```

Table 1: Comparison of Approximation with Expectation (n=5)

Approximation5	Expectation5
0.1250	0.1666667
0.3125	0.3333333
0.5000	0.5000000
0.6875	0.6666667
0.8750	0.8333333

```

kable(table2, caption = "Comparison of Approximation with Expectation (n=10)") %>%
  kable_styling(position = "center") %>%
  kable_styling(latex_options = "HOLD_position")

```

Table 2: Comparison of Approximation with Expectation (n=10)

Approximation10	Expectation10
0.0645161	0.0909091
0.1612903	0.1818182
0.2580645	0.2727273
0.3548387	0.3636364
0.4516129	0.4545455
0.5483871	0.5454546
0.6451613	0.6363636
0.7419355	0.7272727
0.8387097	0.8181818
0.9354839	0.9090909

As you can see above tables, the smaller the value of n and the more extreme the value of i, the more the approximation tends to depart from the expectation. However, as already mentioned in the textbook, this approximation remains accurate to some extent even when n is small, especially when i is closer to the middle value.

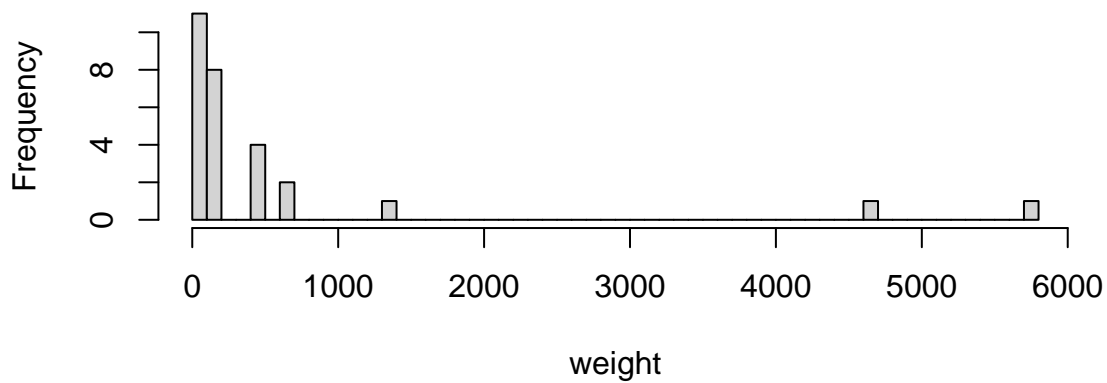
4.39:

First I entered the weight data and plot the histogram of the data as below.

```
weight<-c(0.4, 1.0, 1.9, 3.0, 5.5, 8.1, 12.1, 25.6, 50.0, 56.0, 70.0, 115.0, 115.0, 119.5, 154.5, 157.0)
```

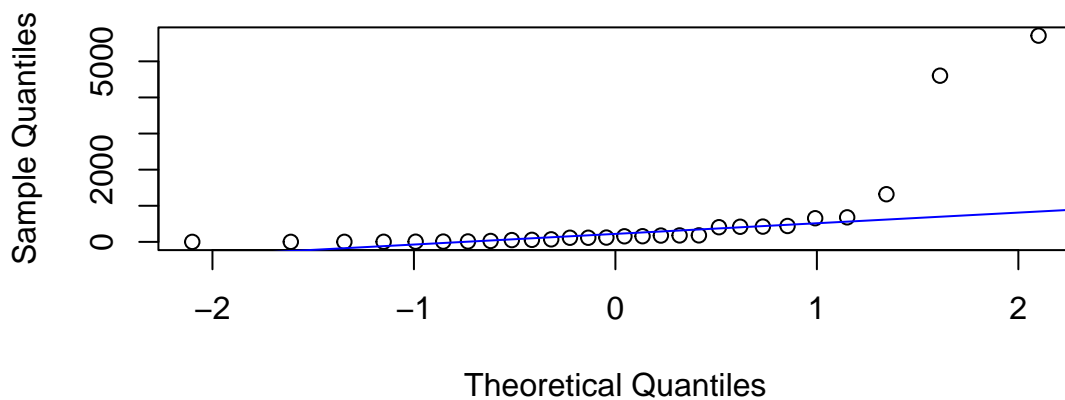
```
#Histogram of the data
hist(weight, breaks = 50)
```

Histogram of weight



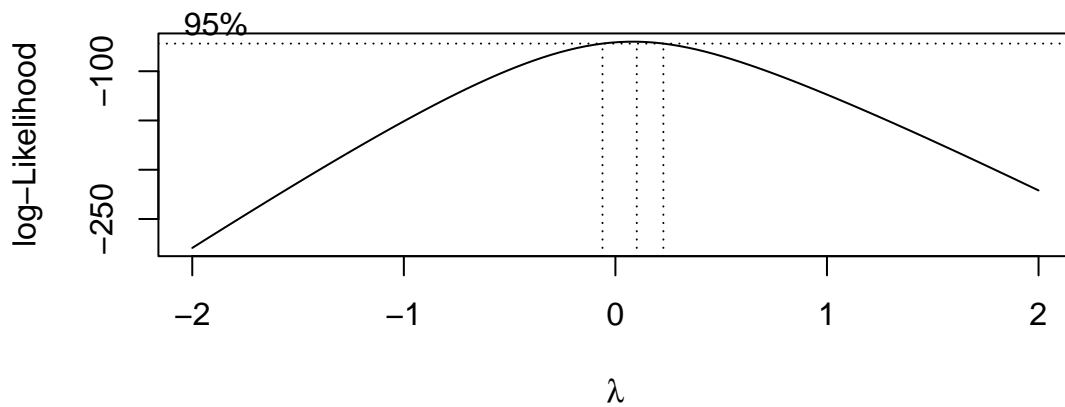
```
#QQplot of the data
qqnorm(weight)
qqline(weight, col = "blue")
```

Normal Q-Q Plot



As you can see, the distribution is skewed to the right, so I could expect Box-Cox transformation similar to log-transform would work well ($\lambda=0$). So first I looked at profile of likelihood of λ of Box-Cox transformation family, and chose the λ with maximum likelihood, which was expected to transform the distribution into approximate normal distribution. [2]

```
#Create the profile of likelihood of lambda of Box-Cox
b <- boxcox(lm(weight ~ 1))
```



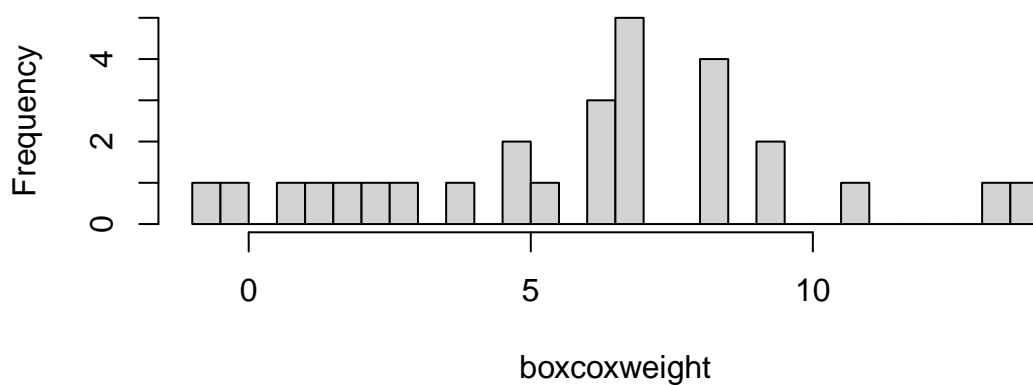
```
#Choose the lambda with maximum likelihood
lambda <- b$x[which.max(b$y)]
```

As we expected, the lambda with maximum likelihood is 0.1010101, which is around 0. Then I did Box-Cox transformation with the lambda value as below.

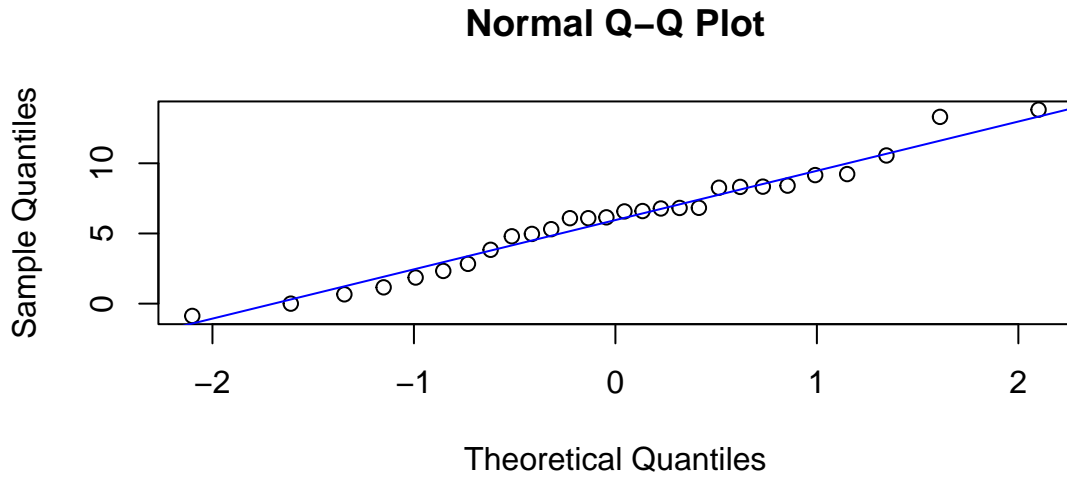
```
# Exact lambda
boxcoxweight <- (weight ^ lambda - 1) / lambda

# The histogram of data transformed and its QQplot
hist(boxcoxweight, breaks = 50)
```

Histogram of boxcoxweight



```
qqnorm(boxcoxweight)
qqline(boxcoxweight, col = "blue")
```



After the Box-Cox transformation, the histogram and QQplot of the data show approximate normal.

4.27: The amount of rainfall in Valencia

(a)

```
#Set the dataset
Jan<-c(0.15, 0.25, 0.10, 0.20, 1.85, 1.97, 0.80, 0.20, 0.10, 0.50, 0.82, 0.40,
      1.80, 0.20, 1.12, 1.83, 0.45, 3.17, 0.89, 0.31, 0.59, 0.10, 0.10, 0.90,
      0.10, 0.25, 0.10, 0.90)
Jul<-c(0.30, 0.22, 0.10, 0.12, 0.20, 0.10, 0.10, 0.10, 0.10, 0.10, 0.10, 0.17,
      0.20, 2.80, 0.85, 0.10, 0.10, 1.23, 0.45, 0.30, 0.20, 1.20, 0.10, 0.15,
      0.10, 0.20, 0.10, 0.20, 0.35, 0.62, 0.20, 1.22, 0.30, 0.80, 0.15, 1.53,
      0.10, 0.20, 0.30, 0.40, 0.23, 0.20, 0.10, 0.10, 0.60, 0.20, 0.50, 0.15,
      0.60, 0.30, 0.80, 1.10, 0.20, 0.10, 0.10, 0.10, 0.42, 0.85, 1.60, 0.10,
      0.25, 0.10, 0.20, 0.10)

#Get summary statistics
kable(describeBy(Jan),caption = "Summary statistics of rainfall in January") %>%
  kable_styling(latex_options="scale_down") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 3: Summary statistics of rainfall in January

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	28	0.7196429	0.7650586	0.425	0.6170833	0.481845	0.1	3.17	3.07	1.471453	1.627648	0.1445825

```
kable(describeBy(Jul),caption = "Summary statistics of rainfall in July") %>%
  kable_styling(latex_options="scale_down") %>%
  kable_styling(latex_options = "HOLD_position")
```

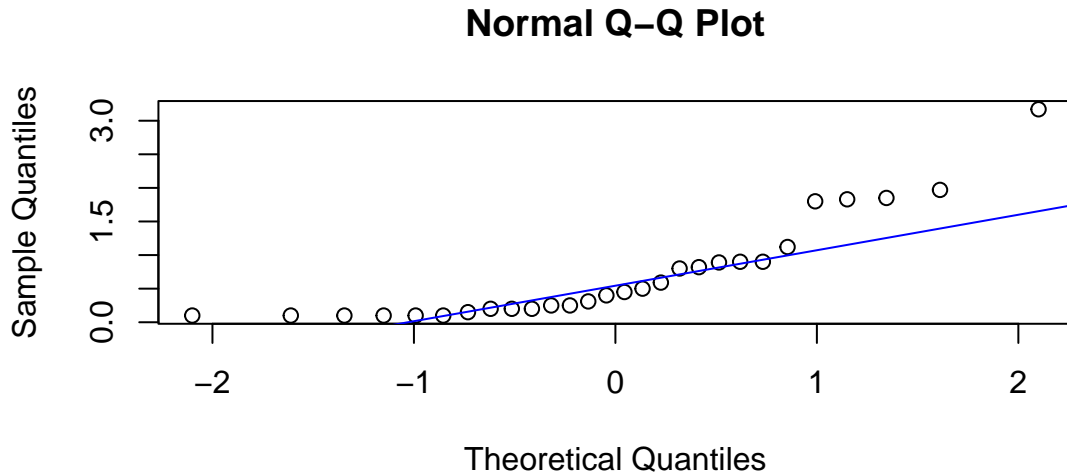
Table 4: Summary statistics of rainfall in July

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	64	0.393125	0.4816733	0.2	0.2880769	0.14826	0.1	2.8	2.7	2.63659	8.414384	0.0602092

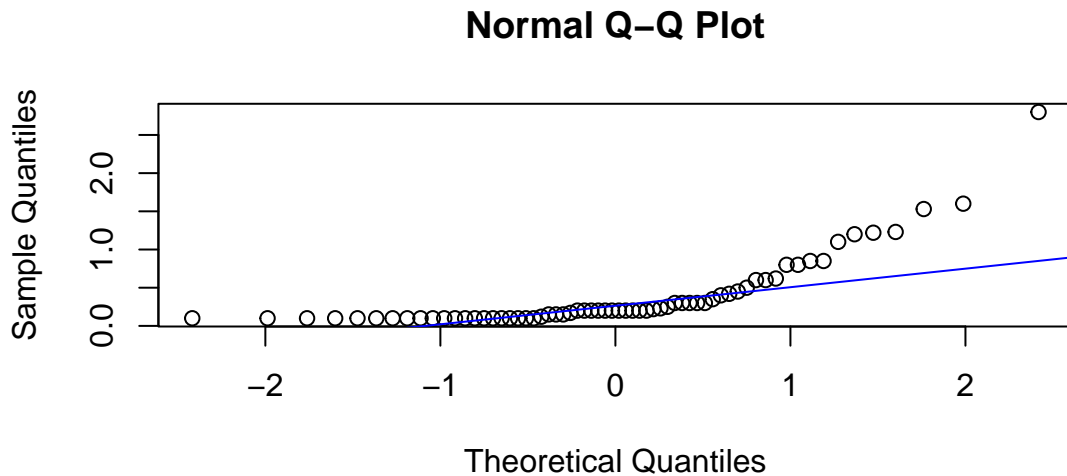
Based on the summary statistics above, we can see that the amount of rainfall in July is less than that in January in terms of mean and maximum. We can also see that we have more data in July so that the standard deviation is less in July than in January.

(b)

```
#Create QQplots  
qqnorm(Jan)  
qqline(Jan, col = "blue")
```

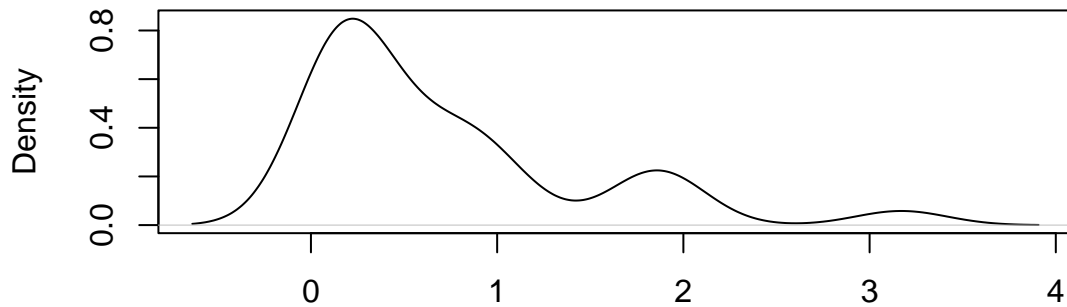


```
qqnorm(Jul)  
qqline(Jul, col = "blue")
```



```
#Create Density plots  
plot(density(Jan),main="Density plot in January")
```

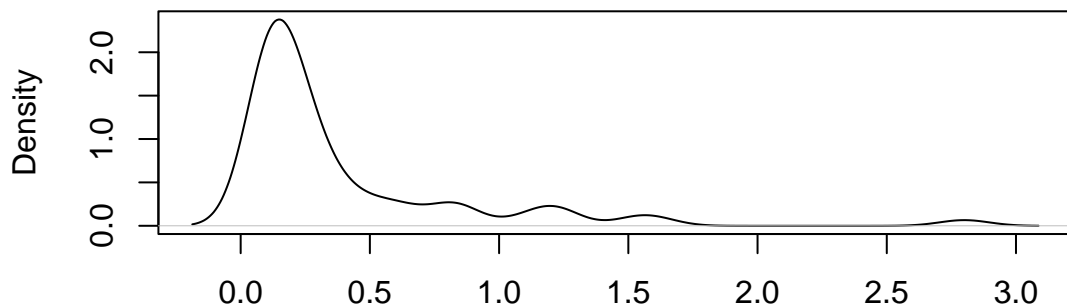
Density plot in January



N = 28 Bandwidth = 0.2457

```
plot(density(Jul),main="Density plot in July")
```

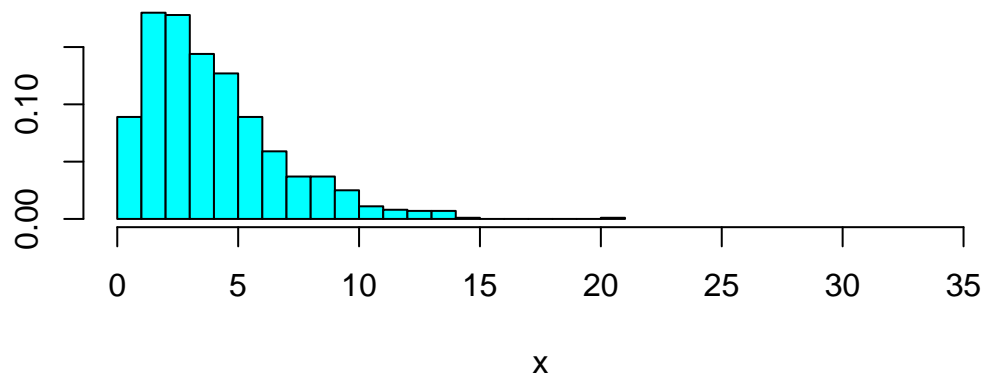
Density plot in July



N = 64 Bandwidth = 0.09574

Based on the QQplots and density plots, both data sets of rainfall in January and July seem to follow gamma distribution. Just to compare the distribution, I draw the gamma distribution with $K=2$ and $\theta=0.5$ as below, which looks very similar to distributions of two data sets.[4]

```
set.seed(2022)
x <- rgamma(1000, 2, 0.5)
truehist(x, xlim = c(0, 35))
```

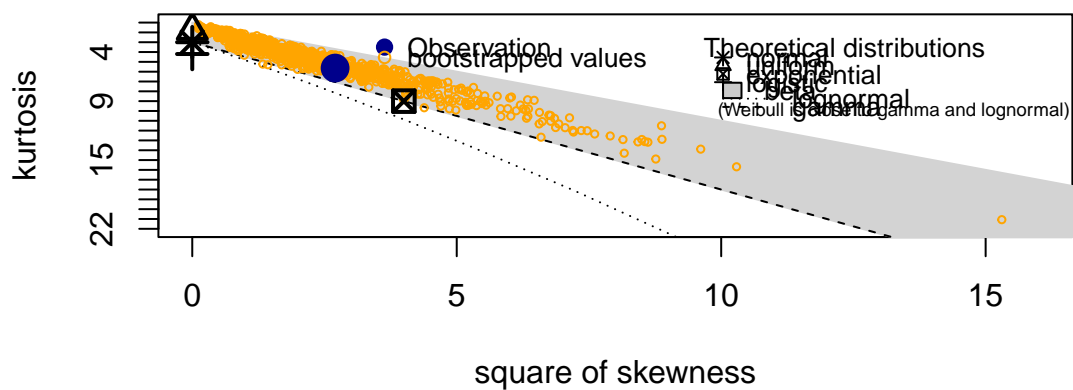


```
rm(x)
```

I also draw skewness-kurtosis plots and plot bootstrap samples as well as the true observations. Then we can see that both of the observations in January and July are very close to gamma distribution.[3]

```
descdist(Jan, boot = 1000)
```

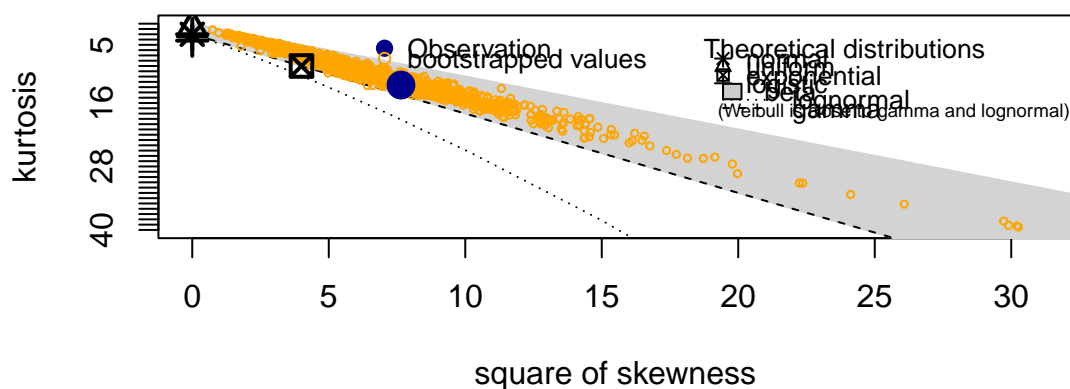
Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.1   max: 3.17
## median: 0.425
## mean: 0.7196429
## estimated sd: 0.7650586
## estimated skewness: 1.643332
## estimated kurtosis: 5.630496
```

```
descdist(Jul, boot = 1000)
```


Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.1 max: 2.8
## median: 0.2
## mean: 0.393125
## estimated sd: 0.4816733
## estimated skewness: 2.764842
## estimated kurtosis: 12.60617
```

(c)

First I fit a gamma model to the data from each month as below.[3]

```
#Fit the gamma distribution to the data
fitg1 <- fitdist(Jan, "gamma", method = "mle")
fitg1
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## shape 1.056222 0.2497495
## rate 1.467650 0.4396202
```

```
fitg2 <- fitdist(Jul, "gamma", method = "mle")
fitg2
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## shape 1.196419 0.1891196
## rate 3.043403 0.5936302
```

MLE and its standard error for shape parameter in January are 1.0562223 and 0.2497495 respectively. MLE and its standard error for rate parameter in January are 1.4676504 and 0.4396202 respectively. MLE and its standard error for shape parameter in July are 1.1964187 and 0.1891196 respectively. MLE and its standard error for rate parameter in January are 3.0434027 and 0.5936302 respectively.

Then I drew profile likelihood with contour based on the results of fitting gamma model to each data set as

below.[4]

```
#Draw contour in January
x <- Jan
NxPts <- 100
NyPts <- 100

xGridPts <- seq(from = 0, to = 2, length = NxPts)
yGridPts <- seq(from = 0.4, to = 2.4, length = NyPts)

xypairs <- expand.grid(xGridPts, yGridPts)
# coordinates of all gridpoints, rows of an N x 2 matrix
# N = NxPts * NyPts
LogLikFun <- function(shape, rate) sum(dgamma(x, shape = shape, rate = rate, log = TRUE))

LogLikOnGrid <- mapply(LogLikFun, xypairs[, 1], xypairs[, 2]) # another long vector

## There is an explicit "for loop" going on inside of "mapply".
## There ought to be a "vectorised" way to do this.

dim(LogLikOnGrid) <- c(NxPts, NyPts) # convert to matrix

## I'll subtract the maximum from the log likelihood function.
## We are only interested in *differences*, not in its absolute
## value. Now I know the maximum of the surface is at height z = 0.
## Also I reshape as a matrix.

MaxLogLik <- max(LogLikOnGrid)
LogLikOnGrid <- matrix(LogLikOnGrid, NxPts, NyPts) - max(LogLikOnGrid)

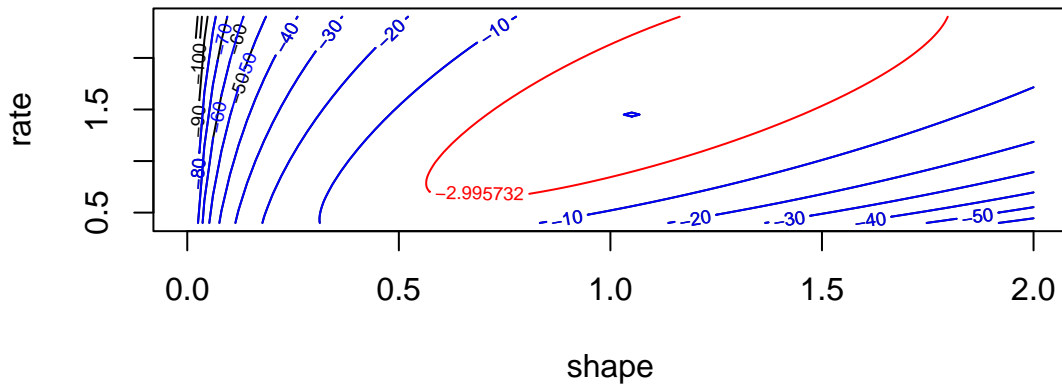
#Draw contour of profile likelihood

contour(x = xGridPts, y = yGridPts, z = LogLikOnGrid)

contour(x = xGridPts, y = yGridPts, z = LogLikOnGrid,
        levels = c((0:-8)*10), col = "blue", add = TRUE)

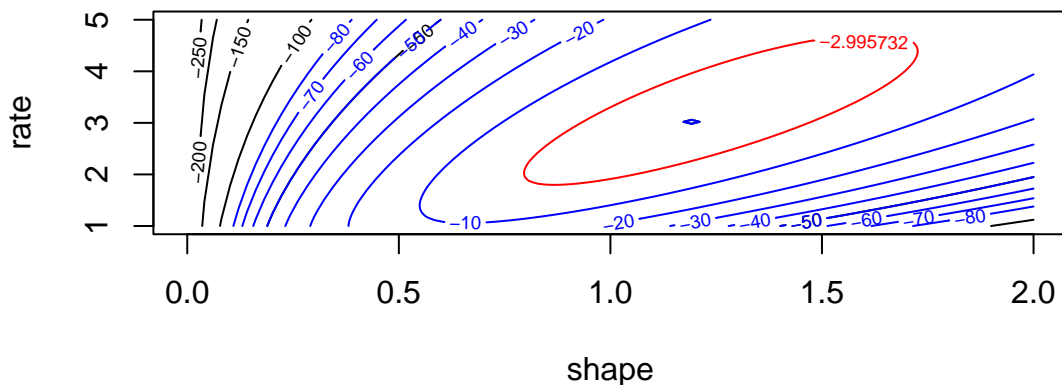
contour(x = xGridPts, y = yGridPts, z = LogLikOnGrid,
        levels = - qchisq(0.95, 2)/2, col = "red", add = TRUE)
title("Contour of profile likelihood (January)", xlab = "shape", ylab = "rate", font = 4)
```

Contour of profile likelihood (January)



```
#Draw contour in July (repeat the same process above)
x <- Jul
xGridPts <- seq(from = 0, to = 2, length = NxPts)
yGridPts <- seq(from = 1, to = 5, length = NyPts)
xypairs <- expand.grid(xGridPts, yGridPts)
LogLikFun <- function(shape, rate) sum(dgamma(x, shape = shape, rate = rate, log = TRUE))
LogLikOnGrid <- mapply(LogLikFun, xypairs[, 1], xypairs[, 2]) # another long vector
dim(LogLikOnGrid) <- c(NxPts, NyPts)
MaxLogLik <- max(LogLikOnGrid)
LogLikOnGrid <- matrix(LogLikOnGrid, NxPts, NyPts) - max(LogLikOnGrid)
contour(x = xGridPts, y = yGridPts, z = LogLikOnGrid)
contour(x = xGridPts, y = yGridPts, z = LogLikOnGrid,
        levels = c((0:-8)*10), col = "blue", add = TRUE)
contour(x = xGridPts, y = yGridPts, z = LogLikOnGrid,
        levels = - qchisq(0.95, 2)/2, col = "red", add = TRUE)
title("Contour of profile likelihood (July)", xlab = "shape", ylab = "rate", font = 4)
```

Contour of profile likelihood (July)



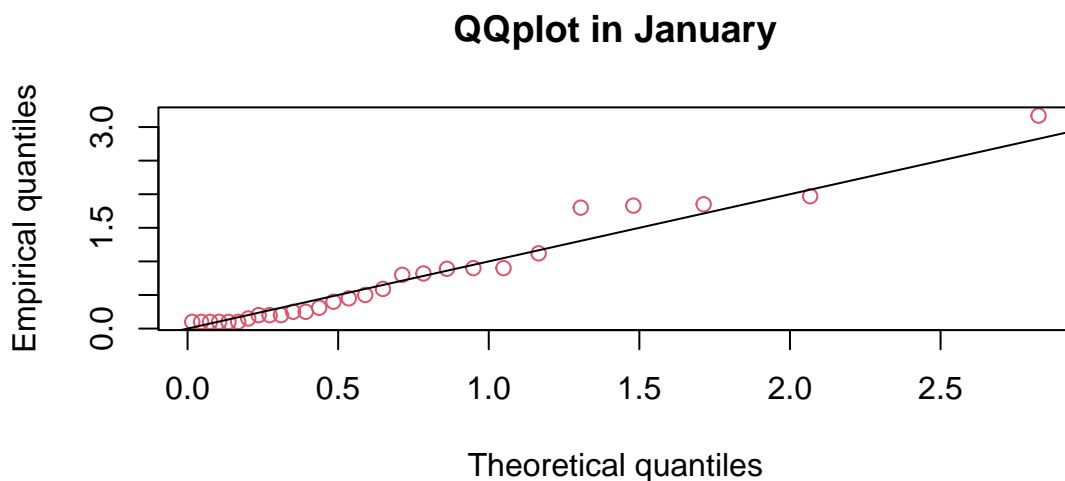
```
# Expectation and variance are calculated by parameters as follows:
# E(x)=shape/rate, Var(x)=shape/(rate^2)
Ex1 <- as.numeric(fitg1$estimate[1]/fitg1$estimate[2])
Var1 <- as.numeric(fitg1$estimate[1]/(fitg1$estimate[2]^2))
Ex2 <- as.numeric(fitg2$estimate[1]/fitg2$estimate[2])
Var2 <- as.numeric(fitg2$estimate[1]/(fitg2$estimate[2]^2))
```

Finally, I compared the parameters from the both data sets. Basically the parameters just decide the shape of gamma distribution, and what's important is we can calculate its expectation (rainfall) and variance based on the parameters.[5] Based on the parameters, the expectation and its variance of January was 0.7196689 and 0.4903544, and those of July was 0.3931188 and 0.1291708. These are quite similar to the mean and variance of the observations. So we can believe this gamma model fit well. We can also see that the expectation of rainfall in January is higher than that in July based on the results.

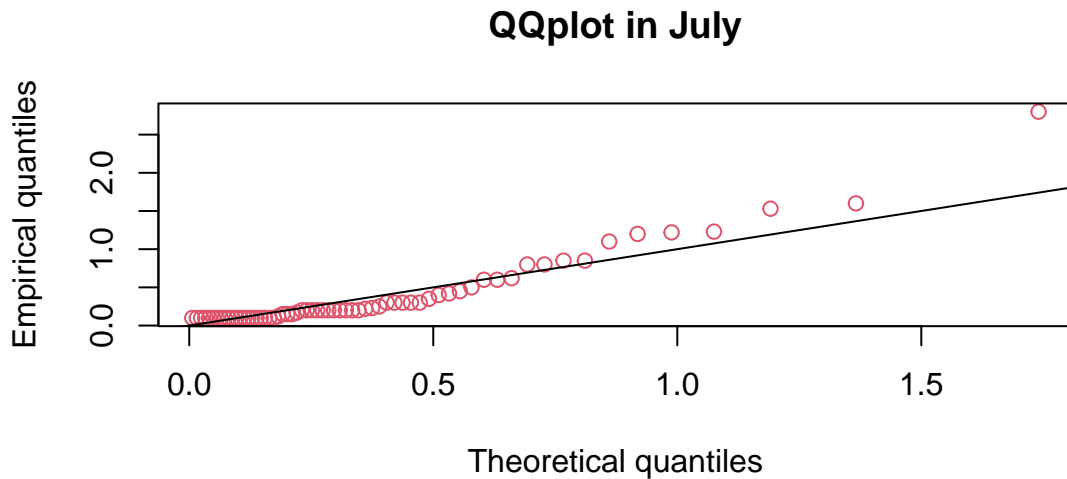
(d)

Lastly, I drew QQplots to check if gamma model is adequate.

```
qqcomp(fitg1, addlegend=FALSE, main="QQplot in January")
```



```
qqcomp(fitg2, addlegend=FALSE, main="QQplot in July")
```



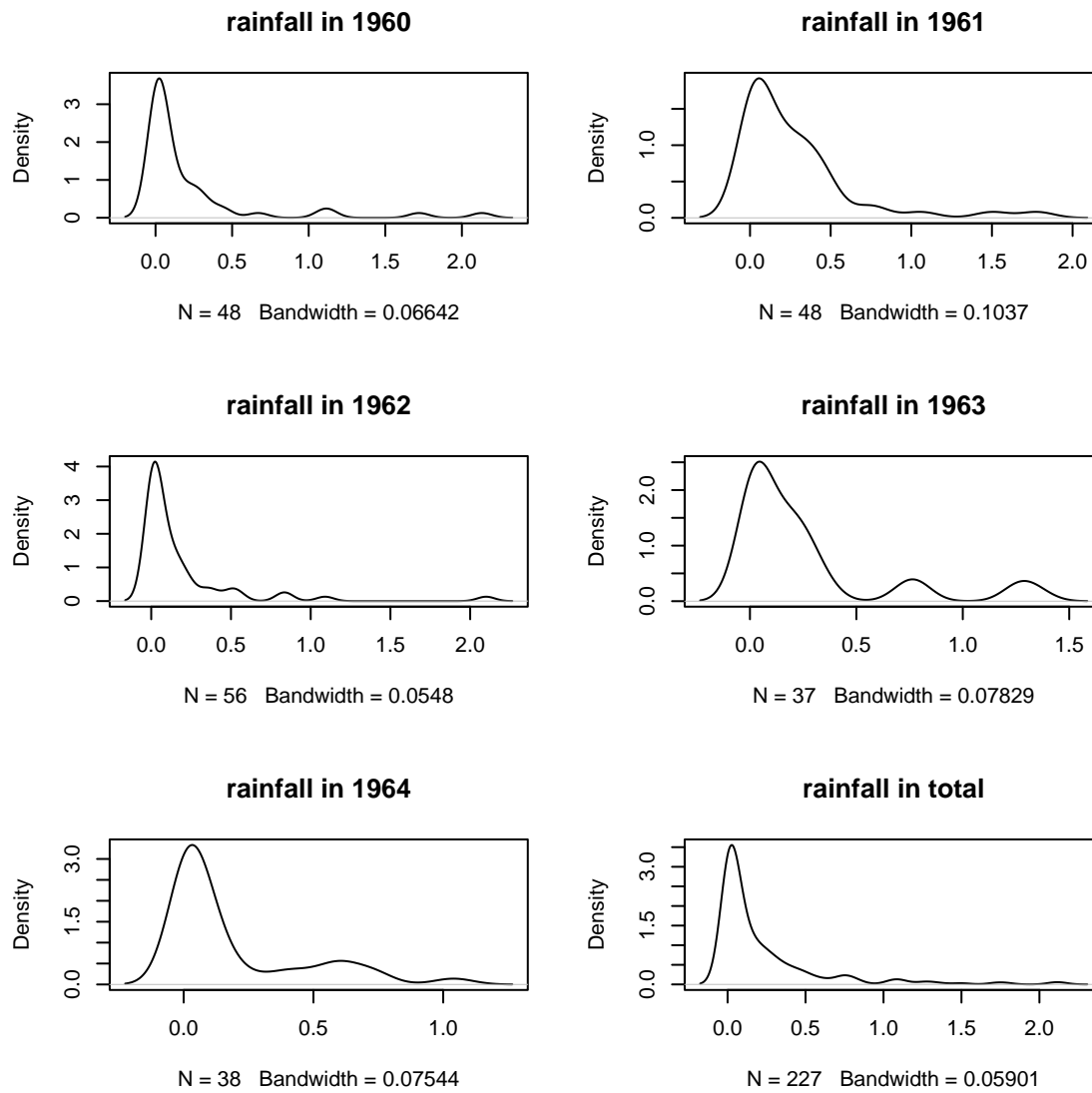
As you can see above, both QQ-plots show adequate fit on the theoretical quantile line, which suggests that gamma models fit well.

The rainfall in Illinois

Identify the distribution of rainfall and Estimate the parameters of the distribution

Use the data to identify the distribution of rainfall produced by the storms in southern Illinois. Estimate the parameters of the distribution using MLE. Prepare a discussion of your estimation, including how confident you are about your identification of the distribution and the accuracy of your parameter estimates. First, I drew density plots for each year as well as the one in total as below.

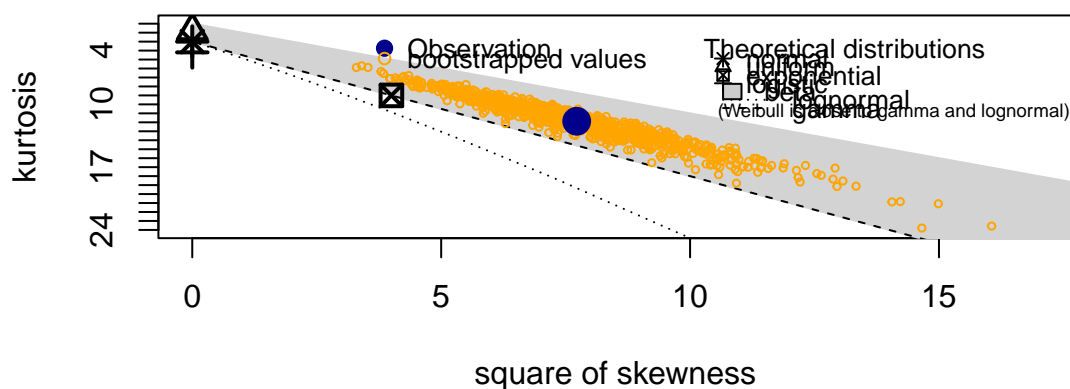
```
rain=read.xlsx('Illinois_rain_1960-1964.xlsx')
par(mfrow = c(3, 2))
plot(density(na.omit(rain$`1960`)),main="rainfall in 1960")
plot(density(na.omit(rain$`1961`)),main="rainfall in 1961")
plot(density(na.omit(rain$`1962`)),main="rainfall in 1962")
plot(density(na.omit(rain$`1963`)),main="rainfall in 1963")
plot(density(na.omit(rain$`1964`)),main="rainfall in 1964")
plot(density(na.omit(unlist(rain))),main="rainfall in total")
```



Then, I drew kurtosis-skewness plots for the total dataset to see which distribution should be fitted.

```
descdist(as.numeric(na.omit(unlist(rain))), boot = 1000)
```

Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.001    max: 2.13
## median: 0.07
## mean: 0.2243921
## estimated sd: 0.3658212
## estimated skewness: 2.778925
## estimated kurtosis: 11.87935
```

Based on the plot above, I thought gamma plot could be fitted to this data sets again, and I fitted the gamma model to the data set as below.

```
x <- as.numeric(na.omit(unlist(rain)))
fit <- fitdist(x, "gamma", method="mle")
fit
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## shape 0.4408386 0.0337663
## rate 1.9648409 0.2474440
```

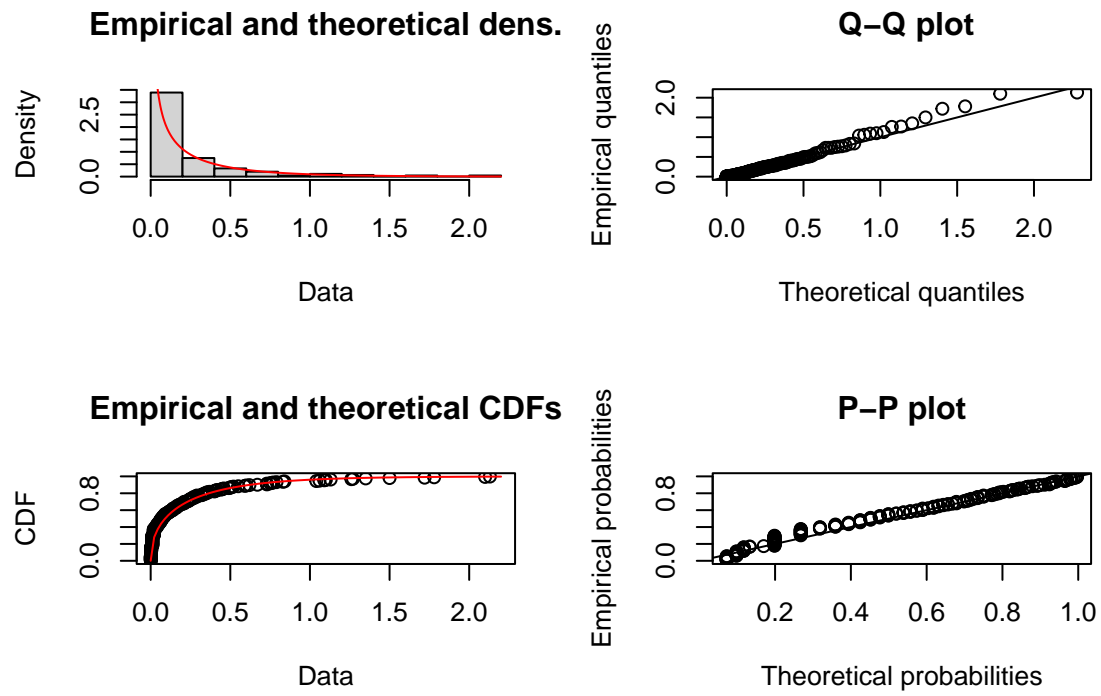
```
set.seed(2022)
summary(bootdist(fit)) #boot get confidence interval
```

Table 5: Parametric bootstrap median and 95% CIs for the MLE parameters

	Median	2.5%	97.5%
shape	0.4453178	0.3807186	0.5163997
rate	1.9989008	1.5697775	2.5787418

The median and 95% CIs for MLEs based on bootstrap samples are shown above. We can also see how the gamma model fit the data by plotting empirical and theoretical density, QQ plot, CDFs, and P-P plot as below. Based on these plot as well as 95% CIs

```
plot(fit)
```



Reference

1. Order statistics in R? <https://stackoverflow.com/questions/24211595/order-statistics-in-r?msclkid=fd6683dac56711ecbfcea9bd8a172395>
2. Box Cox transformation in R <https://r-coder.com/box-cox-transformation-r/>
3. fitdistrplus: An R Package for Fitting Distributions <https://cran.r-project.org/web/packages/fitdistrplus/vignettes/paper2JSS.pdf>
4. Contour, R pubs <https://rpubs.com/gill1109/contour>
5. Gamma Distribution Explained | What is Gamma Distribution? <https://www.mygreatlearning.com/blog/gamma-distribution/#:~:text=The%20PDF%20of%20the%20Gamma%20Distribution&text=Shape%20parameter%20%>