MA677 Final Project

Handing Zhang

2022/5/12

4.25

```
f <- function(x, a=0, b=1) dunif(x, a,b) #pdf function
F <- function(x, a=0, b=1) punif(x, a,b, lower.tail=FALSE) #cdf function

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

#get expectation
E <- function(r,n) {
  (1/beta(r,n-r+1)) * integrate(integrand,-Inf,Inf, r, n)$value
}

# approx function
medianprrox<-function(k,n){
  m<-(k-1/3)/(n+1/3)
  return(m)
}


E(2.5,5)
```

```
## [1] 0.4166667
```

```
medianprrox(2.5,5)
```

```
## [1] 0.40625
```

```
E(5,10)
```

```
## [1] 0.4545455
```

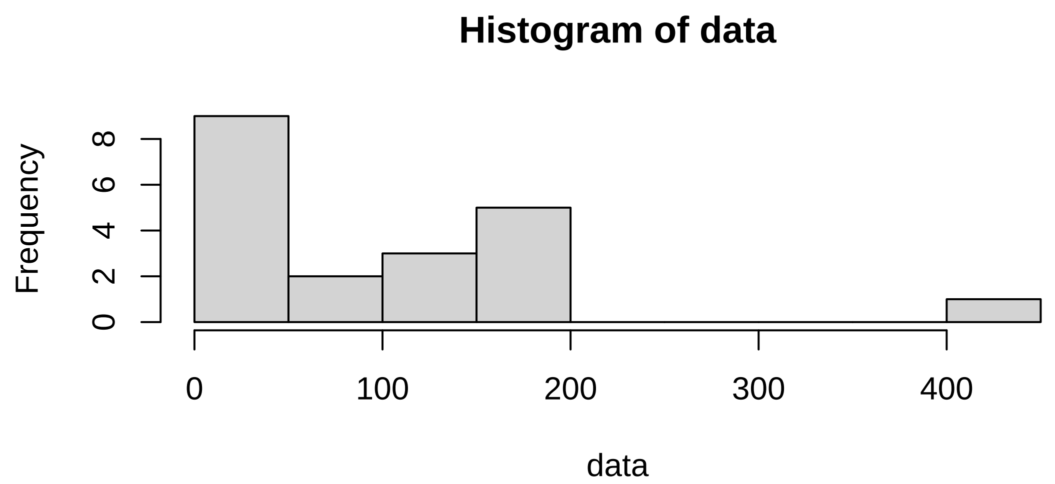
```
medianprrox(5,10)
```

```
## [1] 0.4516129
```

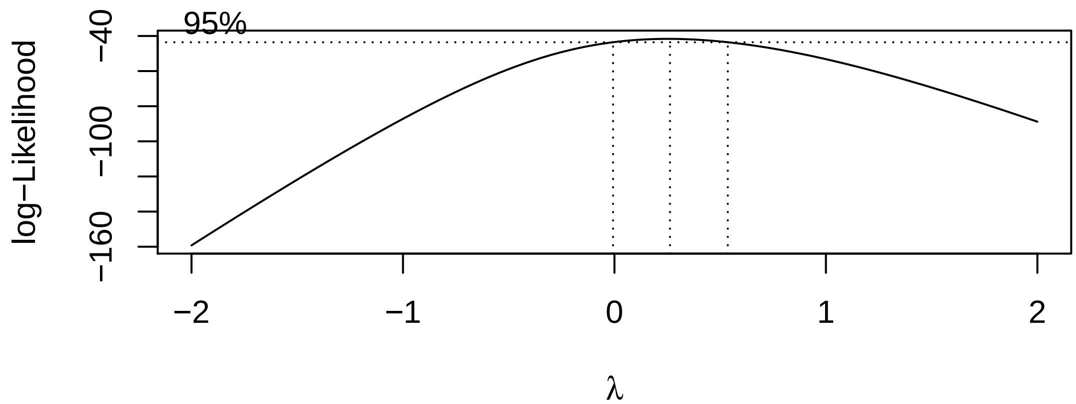
The result shows that they are similar.

4.39

```
data<-c(0.4,1.0,1.9,3.0,5.5,8.1,12.1,25.6,50.0,56.0,70.0,115.0,115.0,119.5,154.5,157.0,175.0,179.0,180.
hist(data)
```



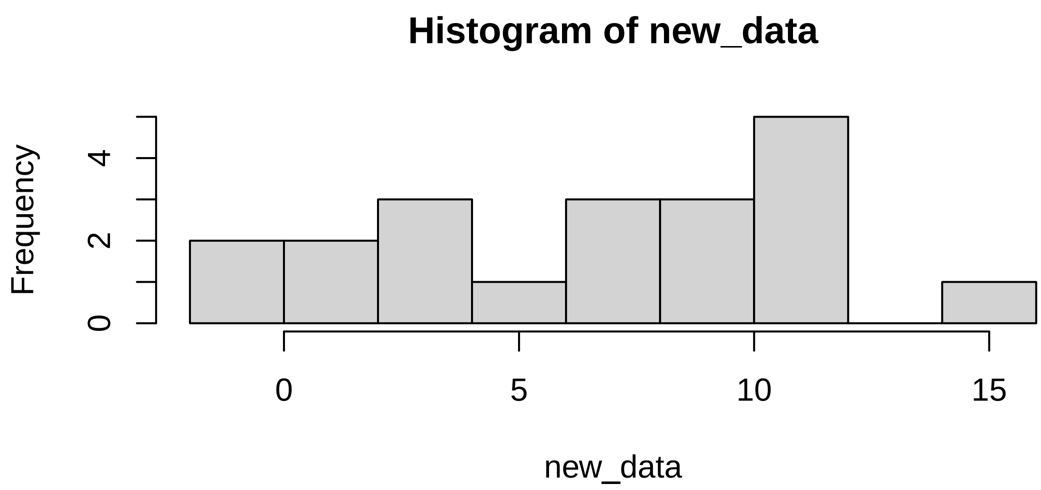

```
# Conduct boxcox transformation
b <- boxcox(lm(data ~ 1))
```




```
# Exact lambda
lambda <- b$x[which.max(b$y)]
lambda #lambda=0.2626263
```

```
## [1] 0.2626263
```

```
new_data <- (data ^ lambda - 1) / lambda
hist(new_data)
```



4.27

```r
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,
      0.2,0.1,0.1,0.1,0.42,0.85,1.6,0.1,0.25,0.1,0.2,0.1)
```

(a)

```r
summary(Jan)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1000  0.1875  0.4250  0.7196  0.9000  3.1700
```

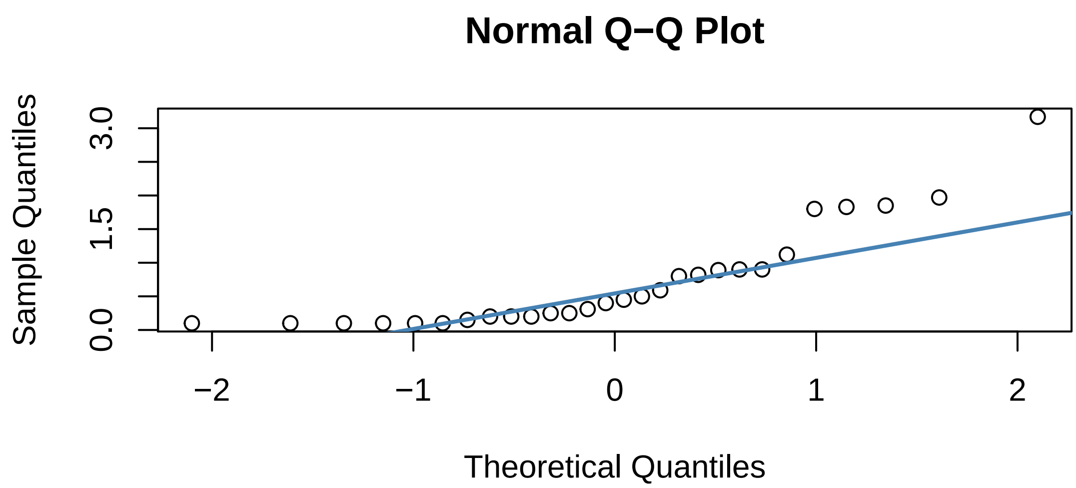
```r
summary(Jul)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1000  0.1000  0.2000  0.3931  0.4275  2.8000
```

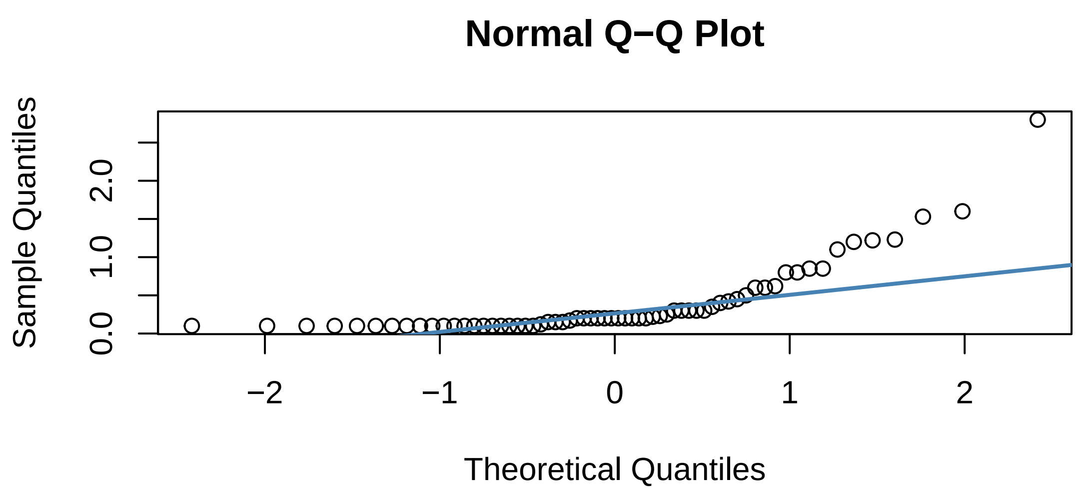
Jan’s 1st, Median, Mean 3rd Max are higher than the one in Jul. Also, Jan’s IQR is higher than the one in Jul.

(b)

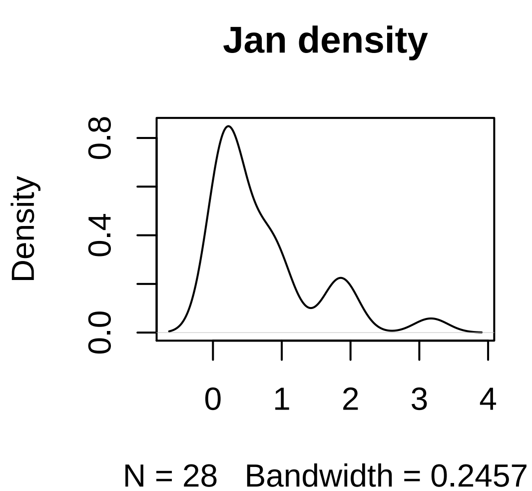
```r
qqnorm(Jan, pch = 1)
qqline(Jan, col = "steelblue", lwd = 2)
```

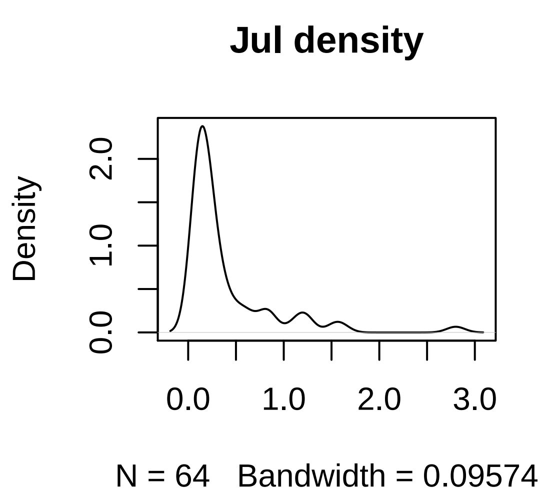



```r
qqnorm(Jul, pch = 1)
qqline(Jul, col = "steelblue", lwd = 2)
```




```r
par(mfrow = c(1, 2))
plot(density(Jan),main='Jan density')
plot(density(Jul),main='Jul density')
```





The qqplots show that the sample doesn't follow normal distribution.
From the density plot, these data looks like gamma distribution. Therefore, gamma distribution can be considered to fit the model.

(c)

There are many ways to solve the problem. I listed three methods here. The first one is to use fitdist:

```
Jan.fit1=fitdist(Jan,'gamma','mle')
Jan.fit1
```

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

```
Jul.fit1=fitdist(Jul,'gamma','mle')
Jul.fit1
```

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

The second method is to nlm:

```
data<-Jan
neg_likelihood<-function(param){
  alpha<-param[1]
  beta<-param[2]
  p<-dgamma(data,shape=alpha,scale=1/beta)
  re<--1*sum(log(p))
  return(re)
}
#neg_likelihood(c(0.5,1))


p <- array(c(0.4, 0.4), dim = c(2, 1))
ans_jan <- nlm(f = neg_likelihood,p,hessian=T)
ans_jan$estimate
```

```
## [1] 1.056259 1.467754
```

```
data<-Jul
ans_jul <- nlm(f = neg_likelihood,p,hessian=T)
ans_jul$estimate
```

```
## [1] 1.196403 3.043315
```

Here is the std

```r
sqrt(diag(solve(ans_jan$hessian))) #use hessian matrix to get std
```

```
## [1] 0.2498280 0.4397828
```

```r
sqrt(diag(solve(ans_jul$hessian)))
```

```
## [1] 0.1891739 0.5938105
```

For MLE, do some stransformation loglikelihood into MLE:

```r
exp(Jan.fit1$loglik)
```

```
## [1] 7.11117e-09
```

```r
exp(Jul.fit1$loglik)
```

```
## [1] 0.02638693
```

```r
exp(-ans_jan$minimum)
```

```
## [1] 7.11117e-09
```

```r
exp(-ans_jul$minimum)
```

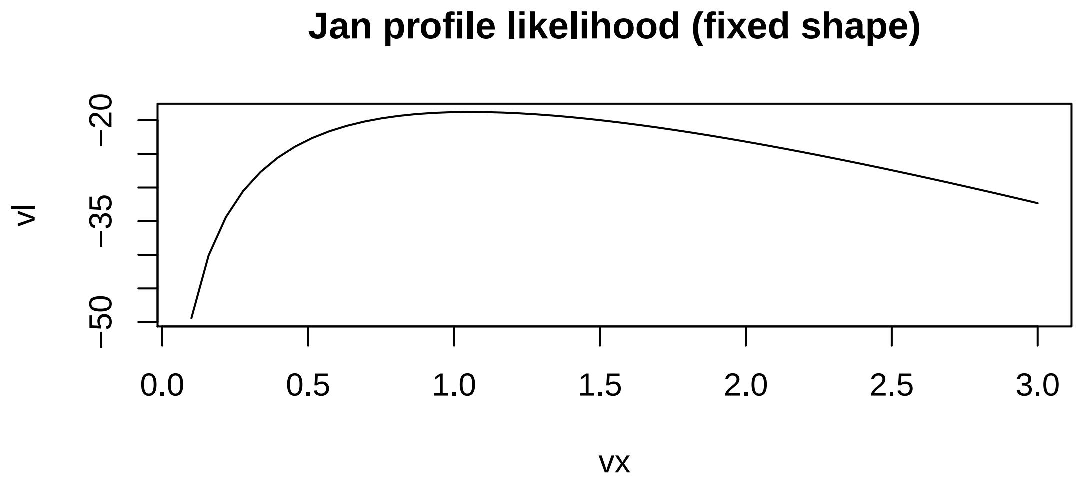
```
## [1] 0.02638693
```

From MLE, Jul's MLE is higher than the one of Jan. Jul's model is better than Jan's.
Parameter comparison: Jan's alpha is lower than Jul's alpha. Jan's beta is lower than Jul's beta.

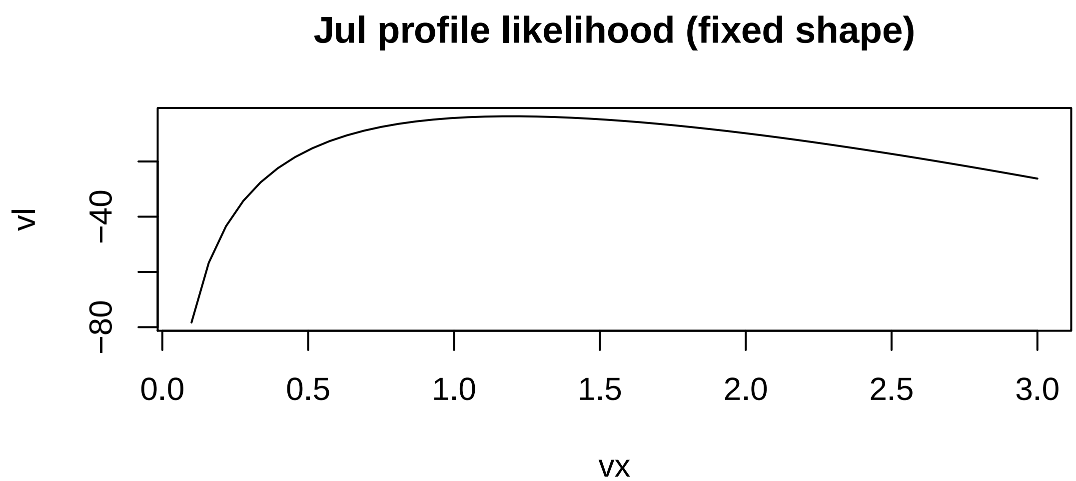
The third way is to use optim. I will use this method to conduct profile likelihood.

```r
#https://www.r-bloggers.com/2015/11/profile-likelihood/
# optim is similar to nlm
# Jan
x=Jan
prof_log_lik=function(a){
   b=(optim(1,function(z) -sum(log(dgamma(x,a,z)))))$par
   return(-sum(log(dgamma(x,a,b))))
 }

vx=seq(.1,3,length=50)
vl=-Vectorize(prof_log_lik)(vx)
plot(vx,vl,type="l",main='Jan profile likelihood (fixed shape)')
```



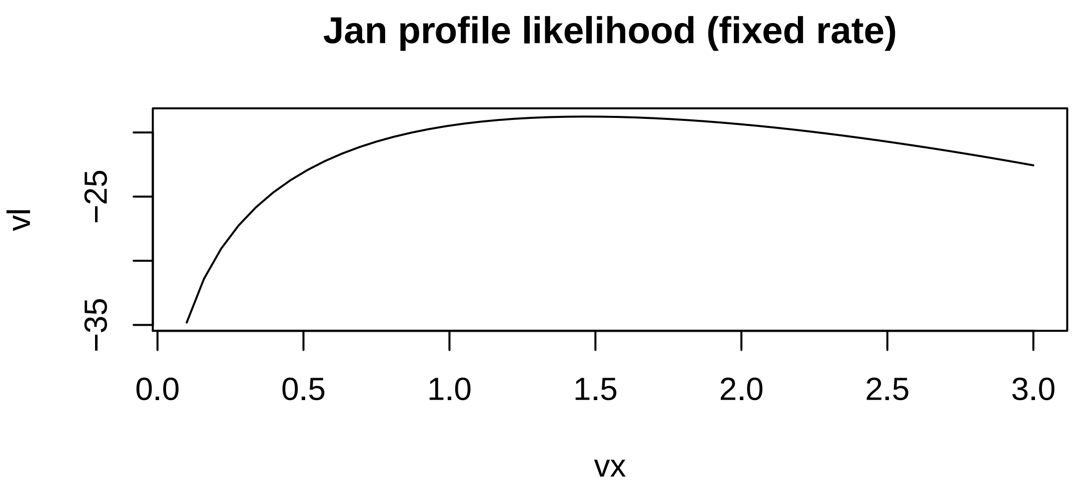

```
x=Jul
vx=seq(.1,3,length=50)
vl=-Vectorize(prof_log_lik)(vx)
plot(vx,vl,type="l",main='Jul profile likelihood (fixed shape)')
```



For fixed rate, we can use the same method to get the profile likelihood.

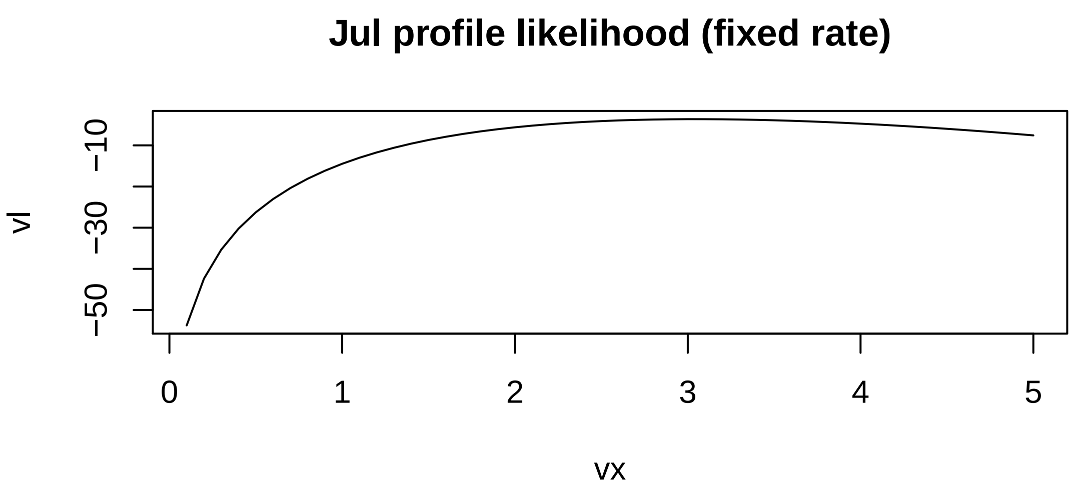
```
x=Jan
prof_log_lik=function(z){
   a=(optim(1,function(a) -sum(log(dgamma(x,a,z)))))$par
   return(-sum(log(dgamma(x,a,z))))
 }

vx=seq(.1,3,length=50)
vl=-Vectorize(prof_log_lik)(vx)
plot(vx,vl,type="l",main='Jan profile likelihood (fixed rate)')
```




```
x=Jul

vx=seq(.1,5,length=50)
vl=-Vectorize(prof_log_lik)(vx)
plot(vx,vl,type="l",main='Jul profile likelihood (fixed rate)')
```



(d)

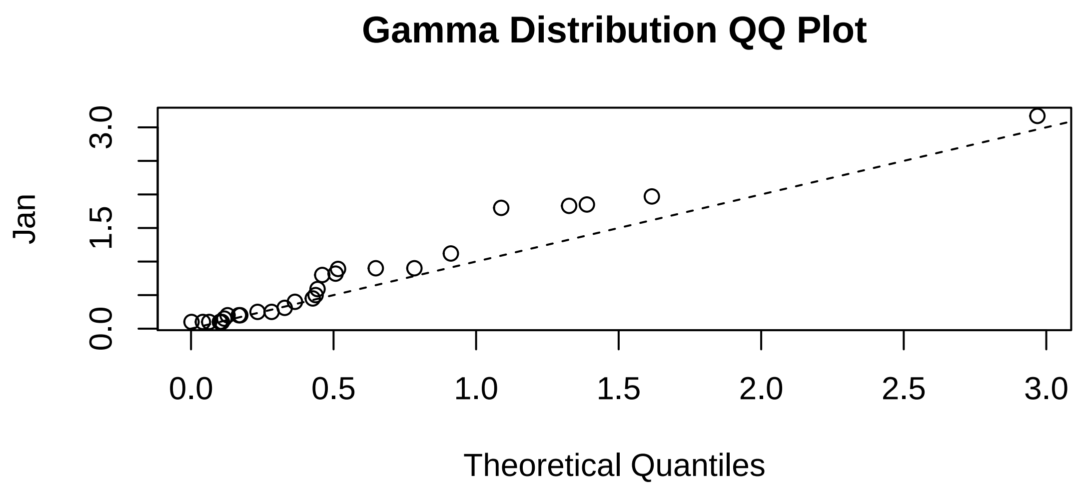
```
# library(qpToolkit)
# qqGamma(resid(Jan.fit))
# reference:qpToolkit
# https://github.com/qPharmetra/qpToolkit/blob/master/R/qqGamma.r
qqGamma <- function(x
                    , ylab = deparse(substitute(x))
                    , xlab = "Theoretical Quantiles"
                    , main = "Gamma Distribution QQ Plot",...)
{
    # Plot qq-plot for gamma distributed variable

    xx = x[!is.na(x)]
```

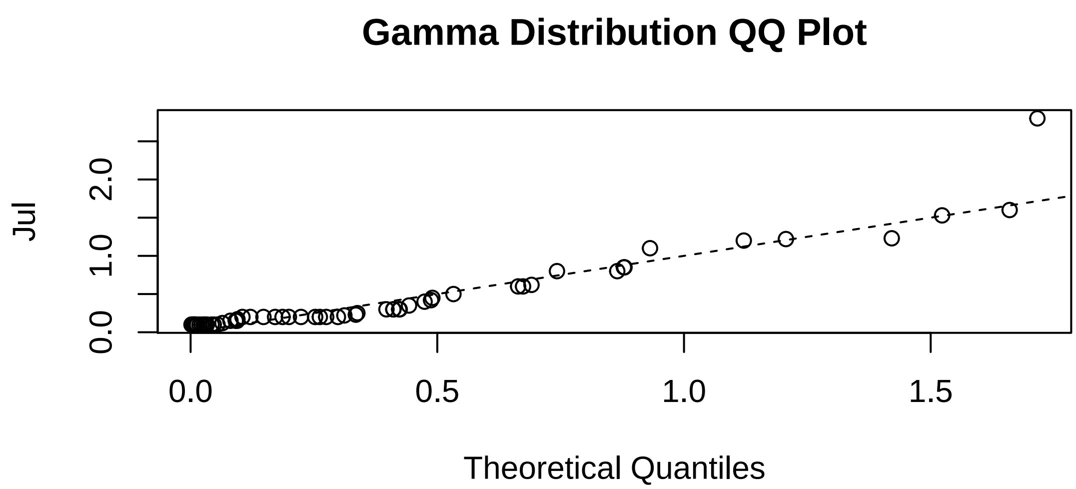
```
    aa = (mean(xx))^2 / var(xx)
    ss = var(xx) / mean(xx)
    test = rgamma(length(xx), shape = aa, scale = ss)

    qqplot(test, xx, xlab = xlab, ylab = ylab, main = main,...)
    abline(0,1, lty = 2)
}


qqGamma(Jan)
```




```
qqGamma(Jul)
```



It seems that Jul is better.

Illinois rain

Question 11

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.

```r
rain <- read.xlsx('Illinois_rain_1960-1964.xlsx')
rain_df <- read.xlsx('Illinois_rain_1960-1964.xlsx')

rain <- as.data.frame(rain)
```

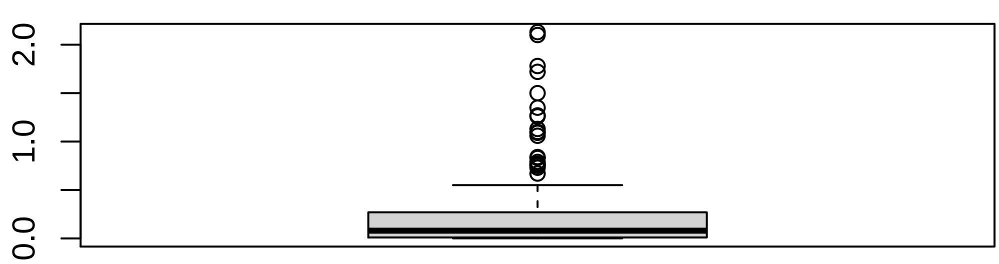
Wrangling.

```r
rain <- c(rain[,1], rain[,2], rain[,3], rain[,4]) # combine the four years of data
rain <- rain[!is.na(rain)] # drop na
rain
```

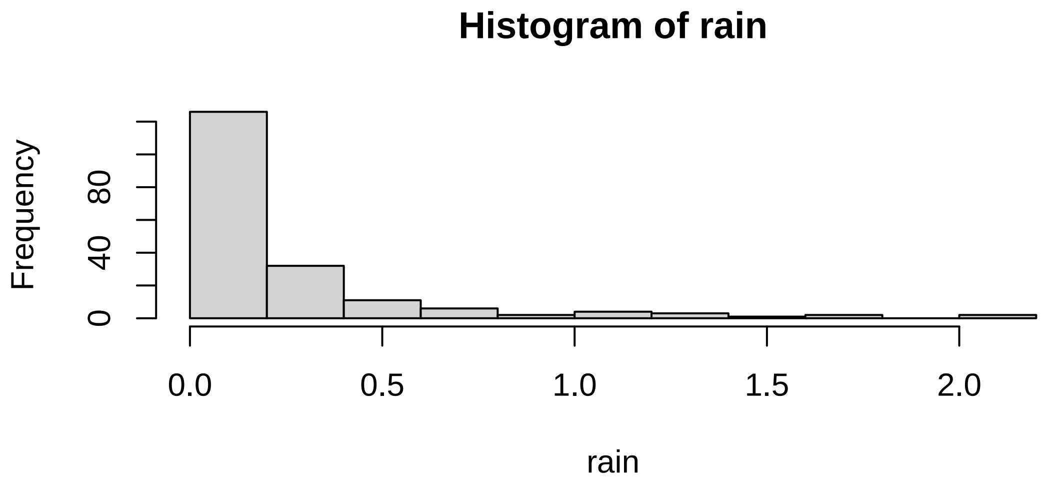
```
##   [1] 0.020 0.001 0.001 0.120 0.080 0.420 1.720 0.050 0.010 0.010 0.003 0.001
##  [13] 0.003 0.270 0.001 0.060 0.050 2.130 0.040 1.100 0.020 0.001 0.140 0.080
##  [25] 0.210 0.070 0.320 0.240 0.290 0.001 0.290 1.130 0.003 0.010 0.190 0.002
##  [37] 0.010 0.040 0.002 0.070 0.450 0.010 0.180 0.670 0.003 0.010 0.040 0.002
##  [49] 0.490 0.020 0.020 0.340 0.140 0.370 0.330 0.330 0.350 0.010 0.500 0.760
##  [61] 1.060 0.002 0.060 0.160 0.270 0.250 0.290 0.020 0.050 0.460 0.070 0.410
##  [73] 0.020 0.080 0.210 0.010 0.440 0.020 0.050 0.110 1.500 0.003 0.180 0.010
##  [85] 0.002 0.240 0.010 0.750 0.010 0.140 0.130 0.010 0.010 0.270 0.450 1.780
##  [97] 0.250 0.240 0.004 0.210 0.170 0.830 0.150 0.030 0.030 0.500 0.040 0.090
## [109] 0.040 0.060 0.060 0.120 0.003 0.003 0.400 0.020 0.510 0.003 0.020 0.020
## [121] 0.020 0.010 0.001 0.140 0.001 0.100 0.010 1.090 0.010 0.002 0.001 0.840
## [133] 0.030 0.350 0.070 0.001 0.002 0.002 0.200 0.060 0.140 0.010 0.020 0.020
## [145] 0.002 0.001 0.550 0.130 0.190 2.100 0.090 0.350 0.790 0.320 1.350 0.170
## [157] 0.020 0.002 0.010 0.250 0.230 0.170 0.010 0.020 0.001 0.010 0.020 0.110
## [169] 0.210 1.260 0.010 0.730 0.100 0.090 0.007 0.360 0.770 0.210 1.270 0.070
## [181] 0.080 0.160 0.260 0.010 0.230 0.080 0.020 0.010 0.290
```

Basic visualization.

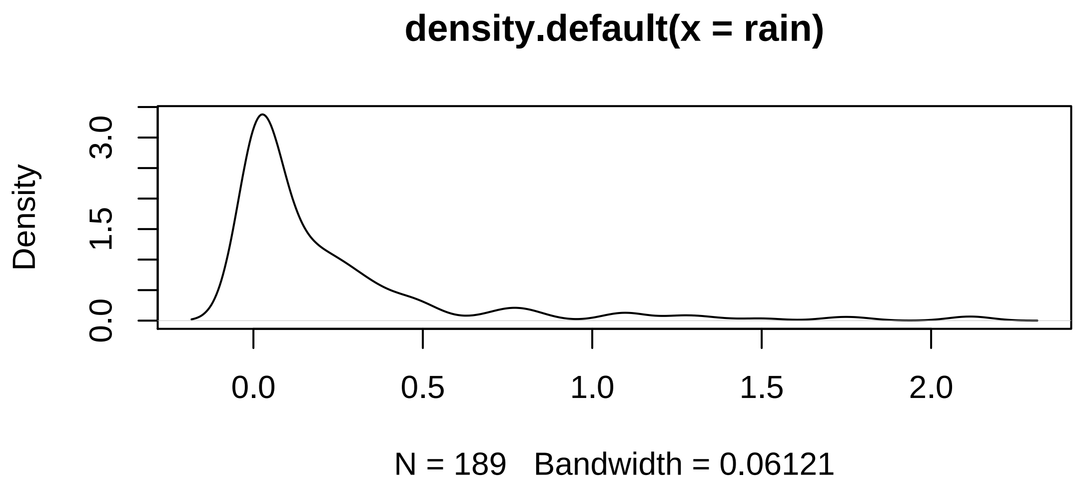
```r
boxplot(rain)
```




```r
hist(rain)
```




```r
density(rain) %>% plot()
```



Use the whole dataset to conduct fitdist using MLE method.

```r
fit1<-fitdist(unlist(rain) %>%  na.omit() %>% c(),'gamma',method='mle') #MLE estimation
```

```r
summary(bootdist(fit1)) #boot get confidence interval
```

Table 1: MLE fit of Rain

	Median	2.5%	97.5%
shape	0.4447568	0.3867703	0.521934
rate	1.9825172	1.5781062	2.567854

Q2

Using this distribution, identify wet years and dry years. Are the wet years wet because there were more storms, because individual storms produced more rain, or for both of these reasons?

```r
rain_mean=fit1$estimate[1]/fit1$estimate[2] #get mean for whole dataset


mean(rain)
```

```
## [1] 0.2318889
```

```r
re=apply(rain_df,2,mean,na.rm =TRUE) # get mean for each year


out<-c(re,rain_mean %>% as.numeric() %>% round(4))
names(out)[6]='mean'
#out

num_storm<-c(nrow(rain_df)-apply(is.na(rain_df),2,sum),'/')

knitr::kable(rbind(out,num_storm)) # show the result
```

	1960	1961	1962	1963	1964	mean
out	0.220291666666667	0.2749375	0.18475	0.262432432432432	0.187105263157895	0.2319
num_storm	48	48	56	37	38	/

Comparing the mean, we can see that 1962, 1964 are dryer years, 1961 and 1963 are wetter years. 1960 is the normal year. We can also conclude that more storms don't necessarily result in wet year and more rain in individual storm don't necessarily result in wet year.

Q3

To what extent do you believe the results of your analysis are generalizable? What do you think the next steps would be after the analysis? An article by Floyd Huff, one of the authors of the 1967 report is included.

I think there is not enough data for our analysis to be generalizable and a good way to begin next step is to collect more data.

Reference:

https://stackoverflow.com/questions/24211595/order-statistics-in-r?msclkid=fd6683dac56711ecbfcea9bd8a172395

https://stackoverflow.com/questions/59435824/nlm-with-multiple-variables-in-r

https://stats.stackexchange.com/questions/81542/standard-error-of-mle#:~:text=How%20are%20you%20obtaining%20the%
Hessian%29%29%29.

https://github.com/MA615-Yuli/MA677_final/blob/main/final.Rmd