Lab Week 11

Donald Elrod

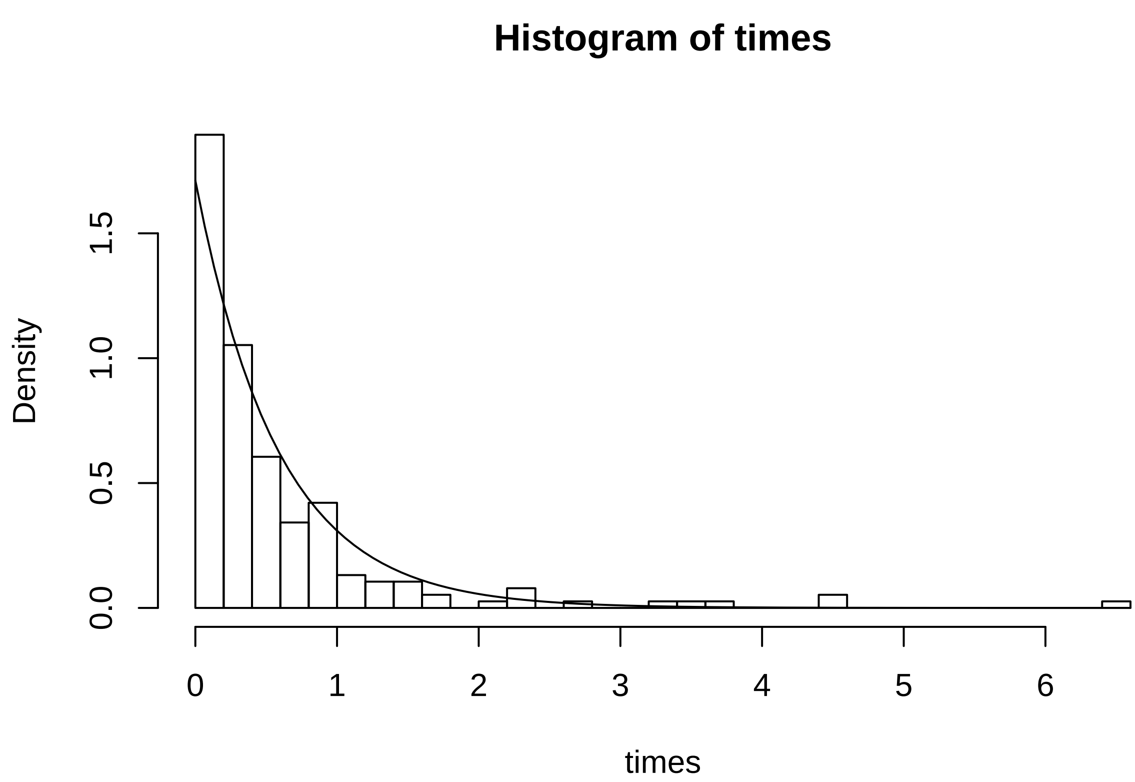
May 24, 2017

```
library(boot)
coal = boot::coal
```

Question 1

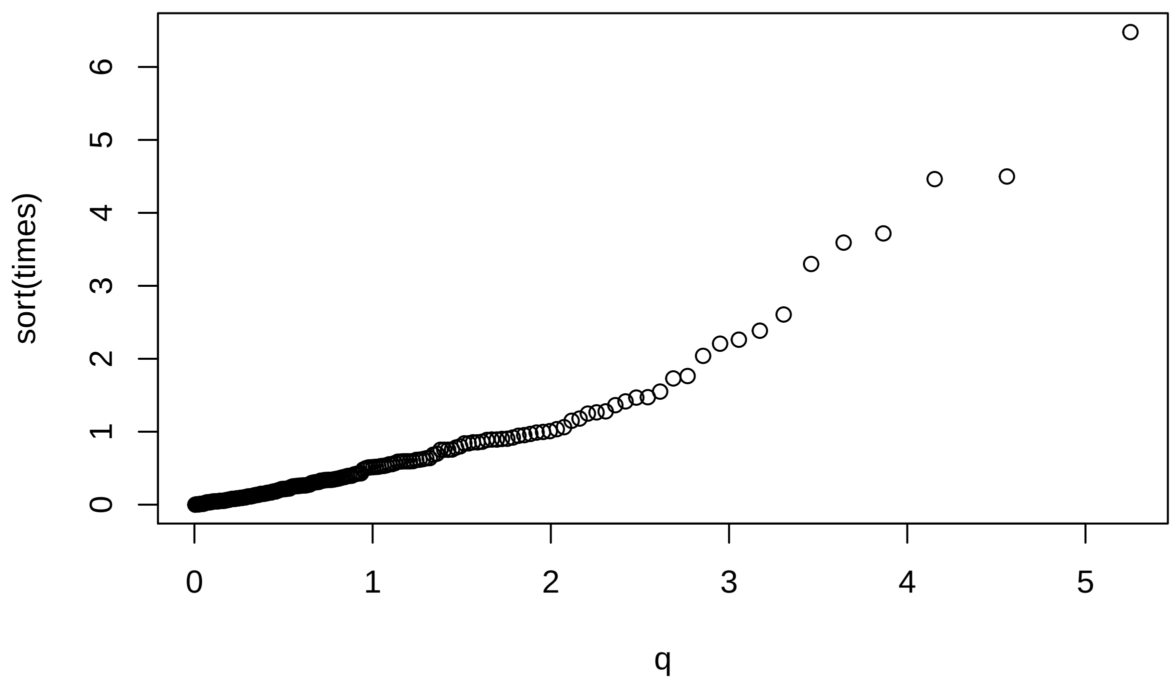
*some code copied from lecture

```
d = coal$date
times = diff(d)
timelen = length(times)

lam.hat = 1 / mean(times)

hist(times, pr=T, n=30)

curve(dexp(x, rate=lam.hat), add=T)
```



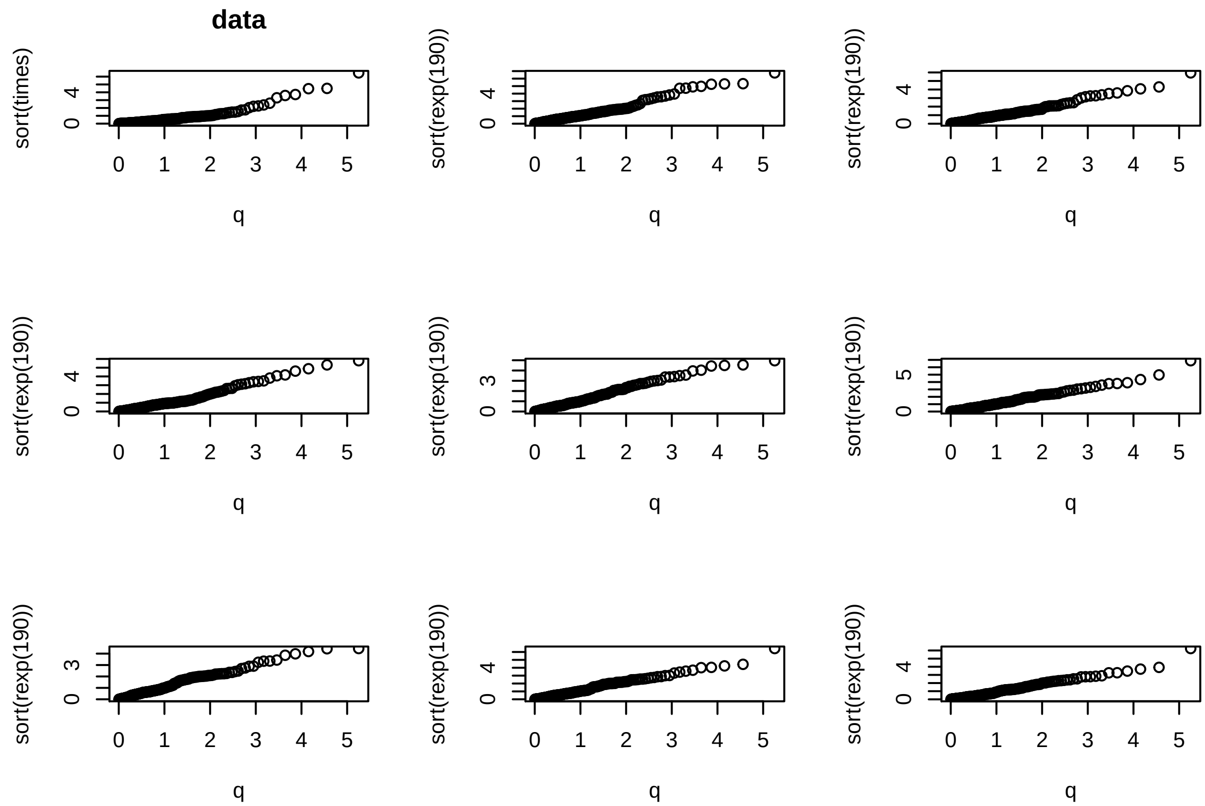
The plot, both the histogram and the line plot, seem to have an exponential trend, with the trendline looking more exponential than the histogram (the histogram has a few outlying blocks)

```
q=qexp((1:190)/191)
plot(q, sort(times))
```



The exponential qq-plot is roughly linear for the first part, however it seems to become less linear and more random towards the tail of the plot

```
par(mfrow=c(3,3))
plot(q, sort(times), main="data")
for (i in 1:8) plot(q, sort(rexp(190)))
```



most of the above plots look pretty similar to the original plot, with some looking more linear and some deviating more than than the original, as well as earlier.

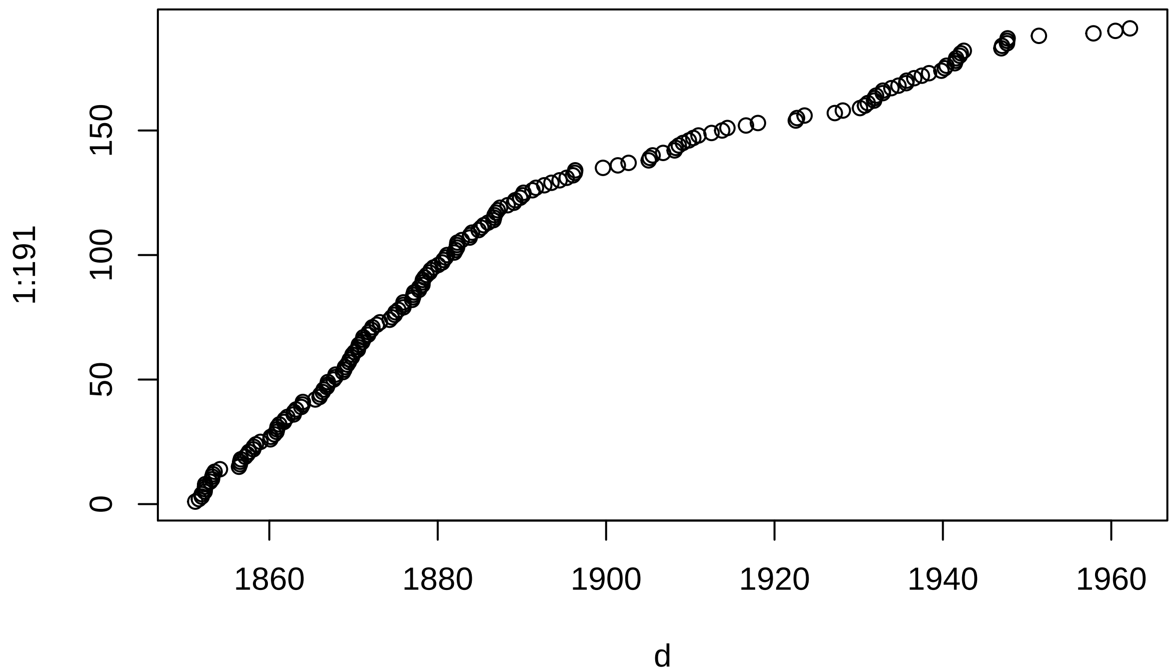
Question 2

It seems that the full dataset is less exponential than the first 100 values in the dataset, as the graphs are much less linear and the histogram is also not consistantly exponential unlike the one in the lecture (while that one was not consistantly exponential either, it was more so than this plot)

Because the qq exponential plot of the data was not very linear, as well as having a psuedo-exponential histogram, it can be seen that the exponential distribution is not a good fit here for the data

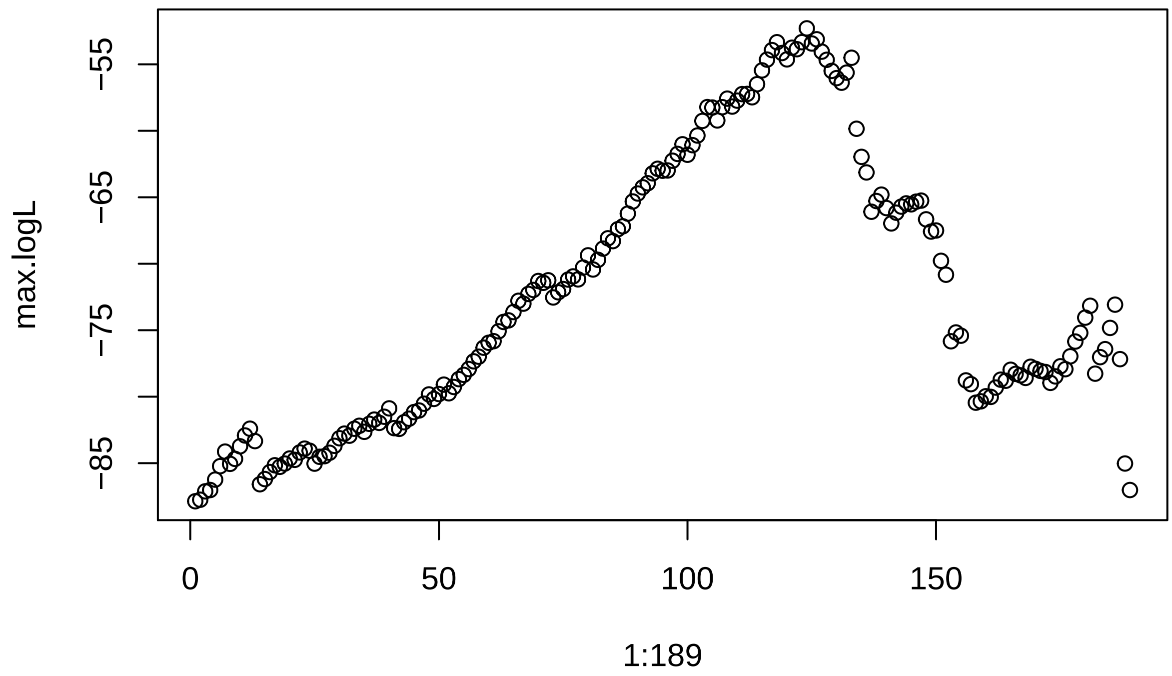
Question 3

```
plot(d, 1:191)
```



Question 4

```
max.logL=0
for(i in 1:189){
  y1=times[1:i]
  y2=times[(i+1):190]
  m=i
  n=190-i
  max.logL[i]=m*log(1/mean(y1))-m +n*log(1/mean(y2)) - n
}
plot(1:189,max.logL)
```



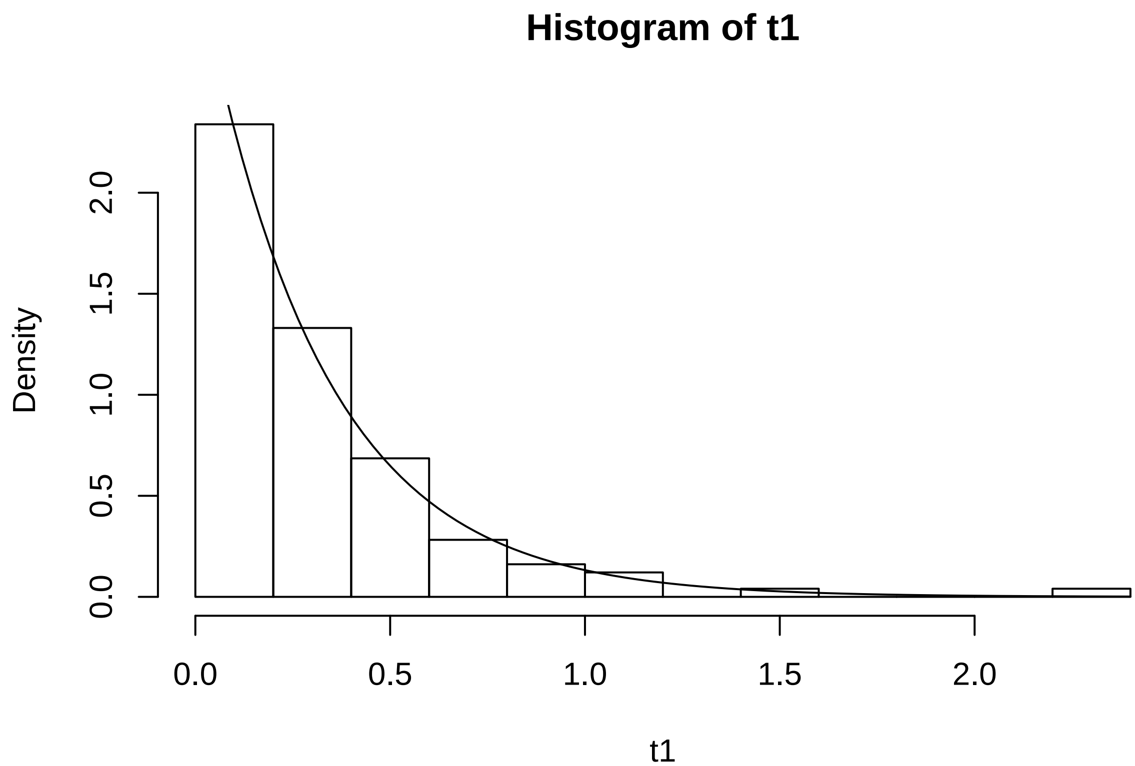

```
i.hat=(1:189)[max.logL==max(max.logL)]
i.hat
```

```
## [1] 124
```

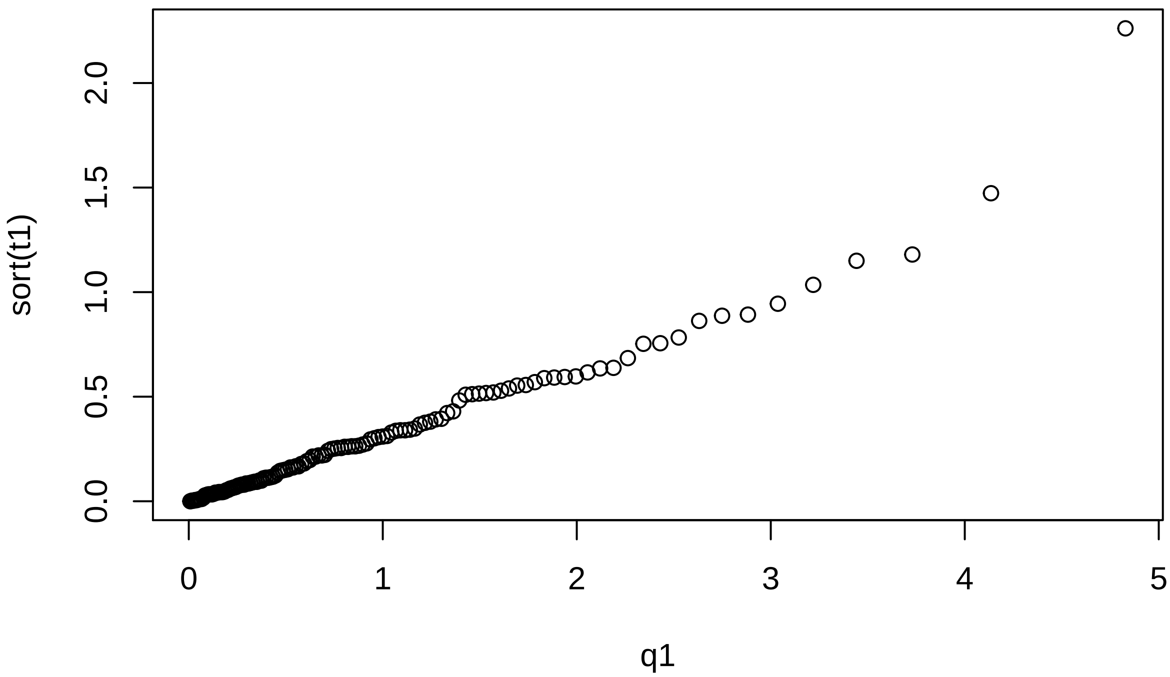
Question 5

T1

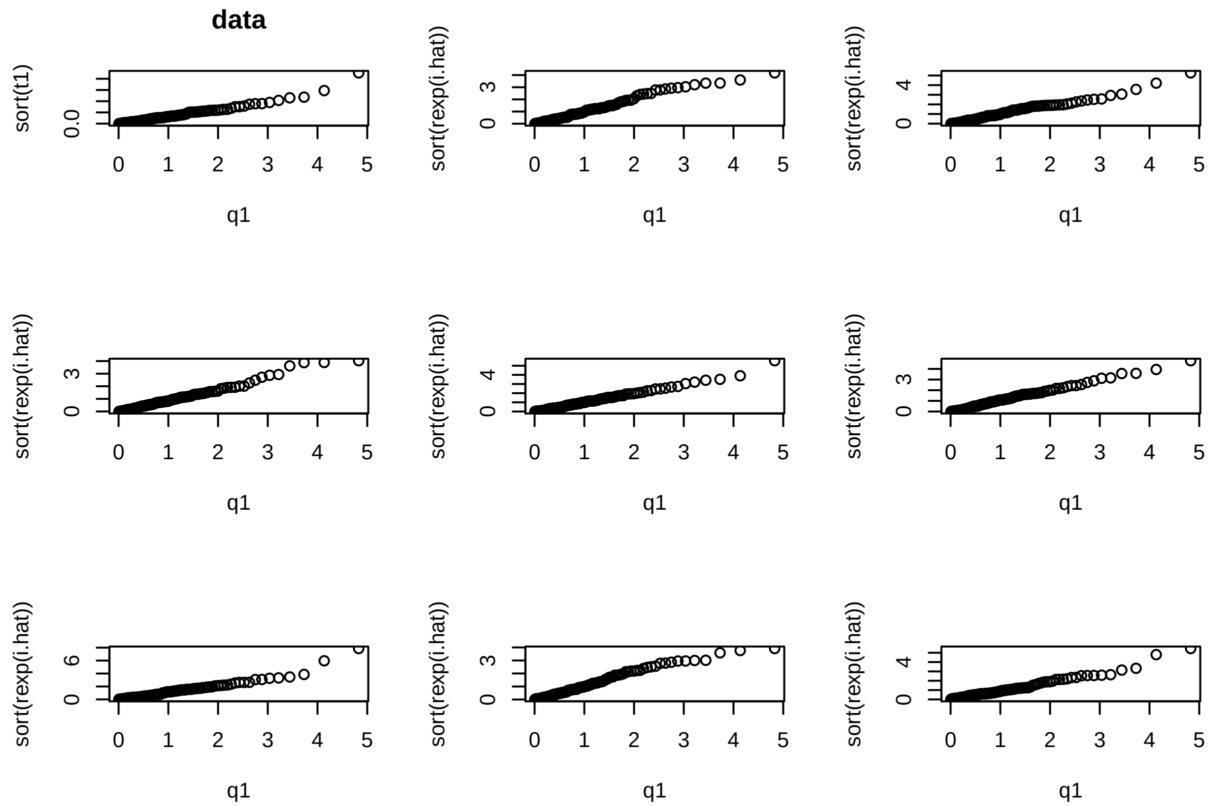
```
t1=times[1:i.hat]
t1len = length(t1)

lam.hat1 = 1 / mean(t1)

hist(t1, pr=T, n=15)

curve(dexp(x, rate=lam.hat1), add=T)
```




```
q1=qexp((1:i.hat)/(i.hat+1))
plot(q1, sort(t1))
```

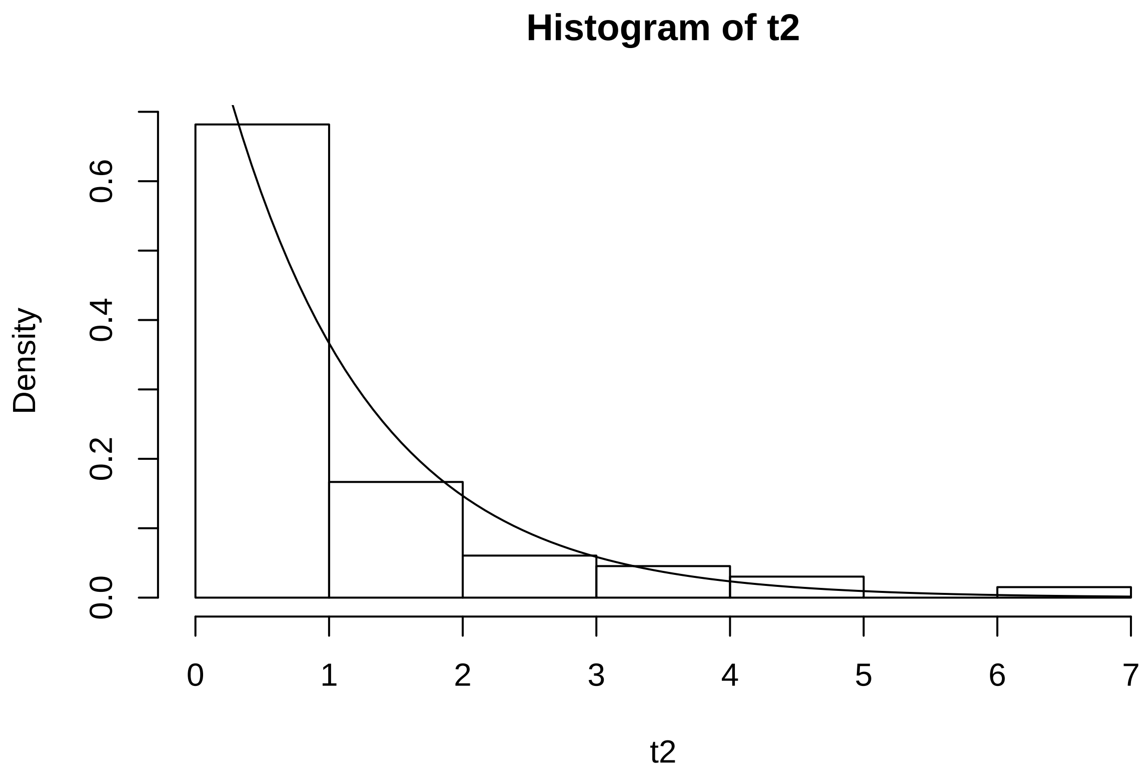



```
par(mfrow=c(3,3))
plot(q1, sort(t1), main="data")
for (i in 1:8) plot(q1, sort(rexp(i.hat)))
```

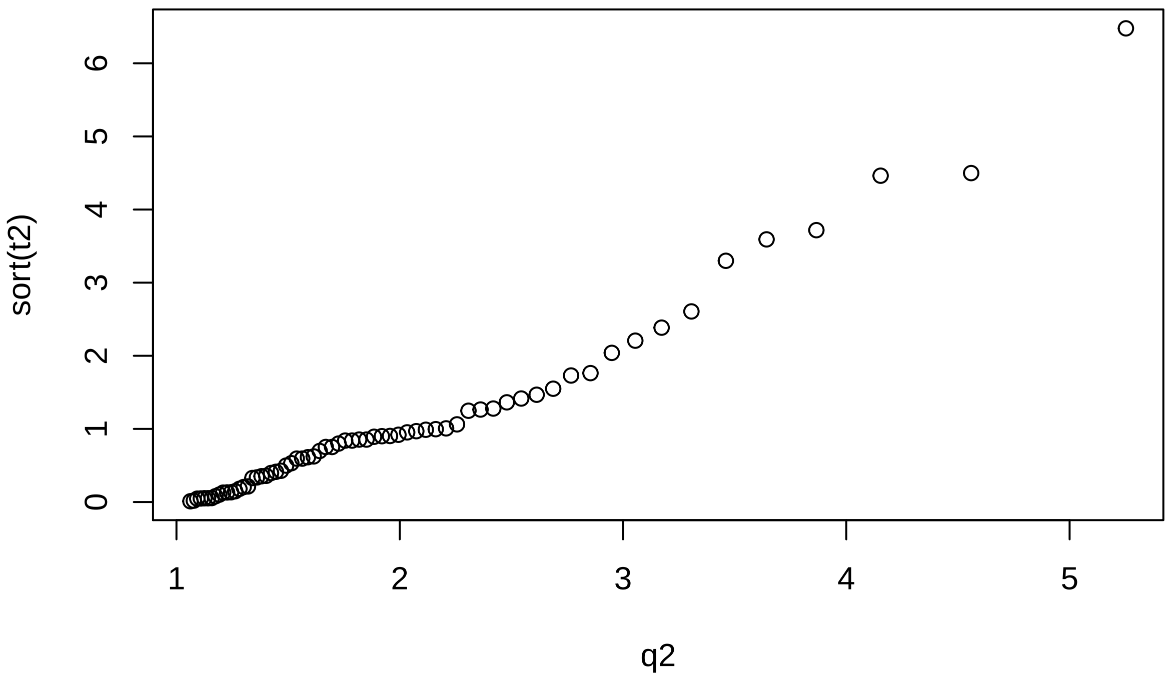


##T2

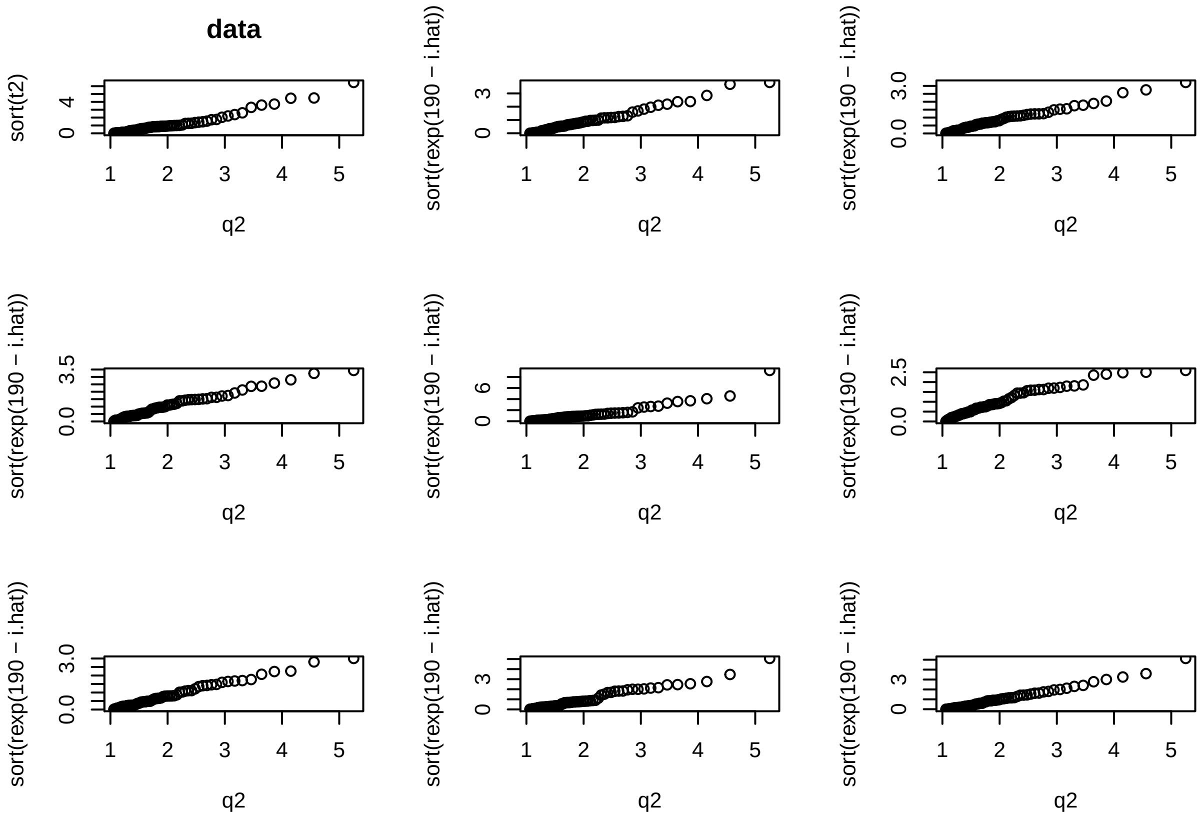
```r
t2=times[(i.hat+1):190]

lam.hat2 = 1 / mean(t2)

hist(t2, pr=T, n=8)

curve(dexp(x, rate=lam.hat2), add=T)
```




```
q2=qexp(((i.hat+1):190)/(191))
plot(q2, sort(t2))
```




```
par(mfrow=c(3,3))
plot(q2, sort(t2), main="data")
for (i in 1:8) plot(q2, sort(rexp(190-i.hat)))
```



Question 6

It does appear to provide a better model: however, I believe that the first part [t1, 1 to i.hat] fits the exponential model better than the second half of the data [t2, i.hat+1 to 190], as t1 doesn't vary as much in the points as t2 does. The histograms for each also look much more exponential than the dataset as a whole, and the variations in the datasets are still closer to linear than the entire dataset together.

Question 7

```
I.hat = i.hat
diff.sim = 0
for (j in 1:1000) {

  lam1 = 1/mean(times[1:I.hat])
  lam2 = 1/mean(times[(I.hat+1):190])

  samp1 = rexp(I.hat, rate=lam1)
  samp2 = rexp(190-I.hat, rate=lam2)
  times = c(samp1, samp2)


  max.logL=0
```

```r
  for(i in 1:189){
    y1=times[1:i]
    y2=times[(i+1):190]
    m=i
    n=190-i
    max.logL[i]=m*log(1/mean(y1))-m +n*log(1/mean(y2)) - n
  }
  i.hat2=(1:189)[max.logL==max(max.logL)]

  y1.sim = times[1:i.hat2]
  y2.sim = times[(i.hat2+1):190]
  diff.sim[j] = 1/mean(y1.sim) - 1/mean(y2.sim)
}

summary(diff.sim)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -120900.00   -3880.00    -136.40   -3469.00      60.23  194200.00
```