Exercise_answers

Franky Zhang

5/11/2022

Exercise 4.25

```
# n = 5
approx = c()
for (i in 1:5) {
  approx[i] = (i - (1/3))/(5 + (1/3))
}
iter = 10000
U_5 = matrix(NA, ncol = 5, nrow = iter)
for (n in 1:iter){
  series = runif(5, min = 0, max = 1)
  U_5[n, 1] = sort(series)[1]
  U_5[n, 2] = sort(series)[2]
  U_5[n, 3] = sort(series)[3]
  U_5[n, 4] = sort(series)[4]
  U_5[n, 5] = sort(series)[5]
}

comparison1 = data.frame(rbind(approx, apply(U_5, 2, median)))
rownames(comparison1) = c("approximation", "calculated value")
colnames(comparison1) = c("i = 1", "i = 2", "i = 3", "i = 4", "i = 5")

# n = 10
U_10 = matrix(NA, ncol = 10, nrow = iter)
for (n in 1:iter){
  series = runif(10, min = 0, max = 1)
  U_10[n, 1] = sort(series)[1]
  U_10[n, 2] = sort(series)[2]
  U_10[n, 3] = sort(series)[3]
  U_10[n, 4] = sort(series)[4]
  U_10[n, 5] = sort(series)[5]
  U_10[n, 6] = sort(series)[6]
  U_10[n, 7] = sort(series)[7]
  U_10[n, 8] = sort(series)[8]
  U_10[n, 9] = sort(series)[9]
  U_10[n, 10] = sort(series)[10]
}
approx = c()
for (i in 1:10) {
  approx[i] = (i - (1/3))/(10 + (1/3))
```

```
}

comparison2 = data.frame(rbind(approx, apply(U_10, 2, median)))
rownames(comparison2) = c("approximation", "calculated value")
colnames(comparison2) = c("i = 1", "i = 2", "i = 3", "i = 4", "i = 5",
                          "i = 6", "i = 7", "i = 8", "i = 9", "i = 10")
```

```
comparison1
```

```
##                      i = 1     i = 2     i = 3     i = 4    i = 5
## approximation    0.1250000 0.3125000 0.5000000 0.6875000 0.875000
## calculated value 0.1314749 0.3125739 0.4980919 0.6813255 0.867813
```

```
comparison2
```

```
##                       i = 1     i = 2     i = 3     i = 4     i = 5     i = 6
## approximation    0.06451613 0.1612903 0.2580645 0.3548387 0.4516129 0.5483871
## calculated value 0.06807123 0.1606799 0.2571132 0.3540970 0.4497805 0.5449387
##                      i = 7     i = 8     i = 9    i = 10
## approximation    0.6451613 0.7419355 0.8387097 0.9354839
## calculated value 0.6439770 0.7416869 0.8373434 0.9332284
```

I've done the simulation for order statistic. From the results, the approximation equation is true.

Exercise 4.27

(a)

```
Jan_1940 = 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_1940 = 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)
```

```
print("summary statistic of Jan 1940:")
```

```
## [1] "summary statistic of Jan 1940:"
```

```
summary(Jan_1940)
```

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

```
cat("\n")
```

```
print("summary statistic of Jul 1940:")
```

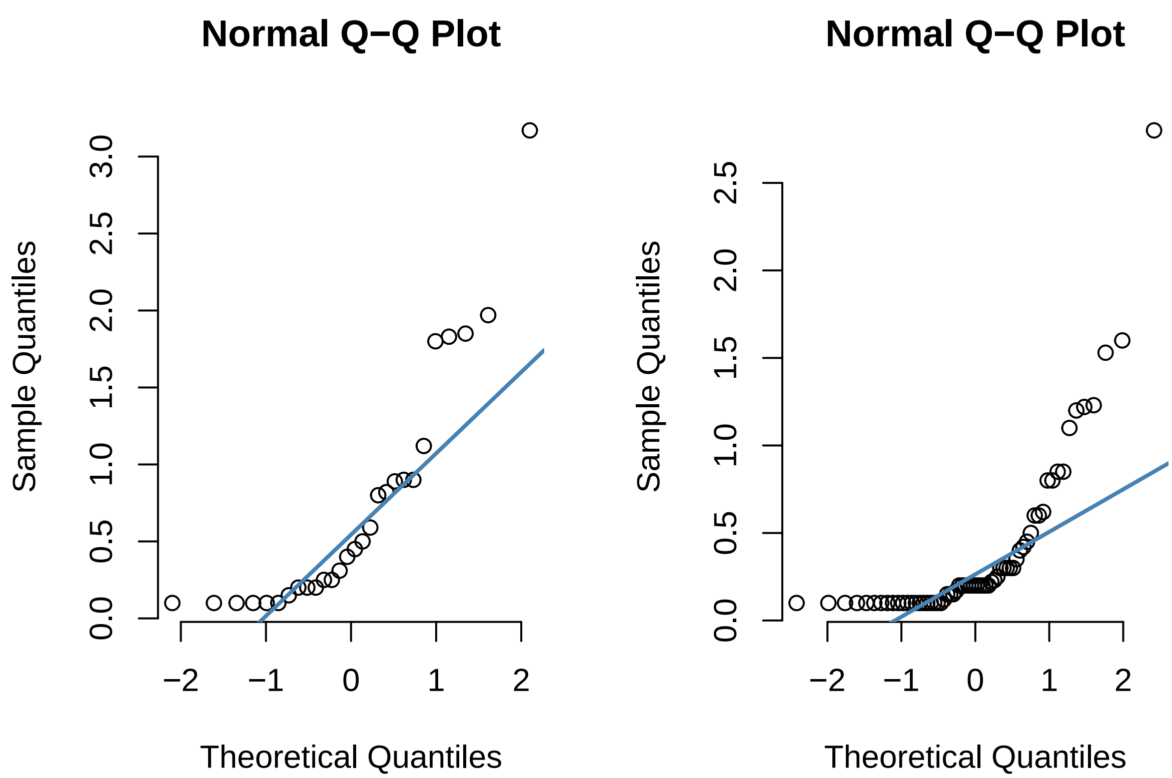
```
## [1] "summary statistic of Jul 1940:"
```

```
summary(Jul_1940)
```

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

(b)

```
par(mfrow = c(1, 2))
qqnorm(Jan_1940, pch = 1, frame = FALSE)
qqline(Jan_1940, col = "steelblue", lwd = 2)
qqnorm(Jul_1940, pch = 1, frame = FALSE)
qqline(Jul_1940, col = "steelblue", lwd = 2)
```



Since the observations are relatively far away from the normal distribution line, normal distribution is no longer be considered. Also, the observations are continuously distributed, consider about the gamma distribution.

(c)

```r
fit_Jan <- fitdist(Jan_1940, distr = "gamma", method = "mle")
fit_Jul <- fitdist(Jul_1940, distr = "gamma", method = "mle")
mean_Jan <- fit_Jan$estimate[1]/fit_Jan$estimate[2]
mean_Jul <- fit_Jul$estimate[1]/fit_Jul$estimate[2]
summary(fit_Jan)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## shape 1.056222  0.2497495
## rate  1.467650  0.4396202
## Loglikelihood:  -18.7616   AIC:  41.5232   BIC:  44.18761
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.7893943
## rate  0.7893943 1.0000000
```

```r
summary(fit_Jul)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## shape 1.196419  0.1891196
## rate  3.043403  0.5936302
## Loglikelihood:  -3.634886   AIC:  11.26977   BIC:  15.58754
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.8103948
## rate  0.8103948 1.0000000
```

Answer: For Jan 1940, the maximum log likelihood is -19.8, the estimated value of shape parameter is 1.06 with sd equal to 0.25 and the rate parameter is 1.47 with sd equal to 0.44. The mean parameter is 0.72.

For Jul 1940, the maximum log likelihood is -3.63, the estimated value of shape parameter is 1.196 with sd equal to 0.19 and the rate parameter is 3.04 with sd equal to 0.59. The mean parameter is 0.393.

(d)

```r
par(mfrow = c(1, 2))
library(EnvStats)
```

```
## Warning: package 'EnvStats' was built under R version 4.1.2
```

```
##
## Attaching package: 'EnvStats'
```

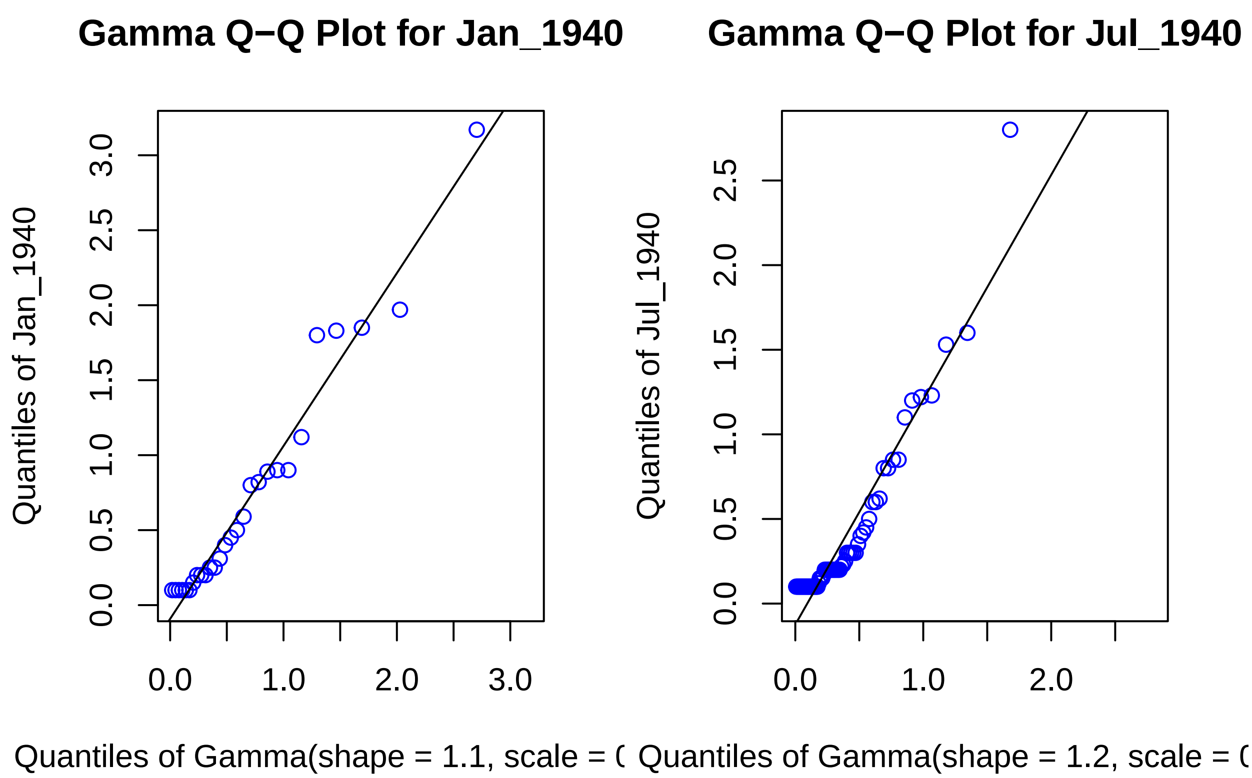
```
## The following object is masked from 'package:car':
##
##     qqPlot
```

```
## The following object is masked from 'package:MASS':
##
##     boxcox

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm

## The following object is masked from 'package:base':
##
##     print.default
```

```r
qqPlot(Jan_1940, dist = "gamma",
      estimate.params = TRUE, digits = 2, points.col = "blue",
      add.line = TRUE)
qqPlot(Jul_1940, dist = "gamma",
      estimate.params = TRUE, digits = 2, points.col = "blue",
      add.line = TRUE)
```




```r
detach("package:EnvStats", unload = TRUE)
```

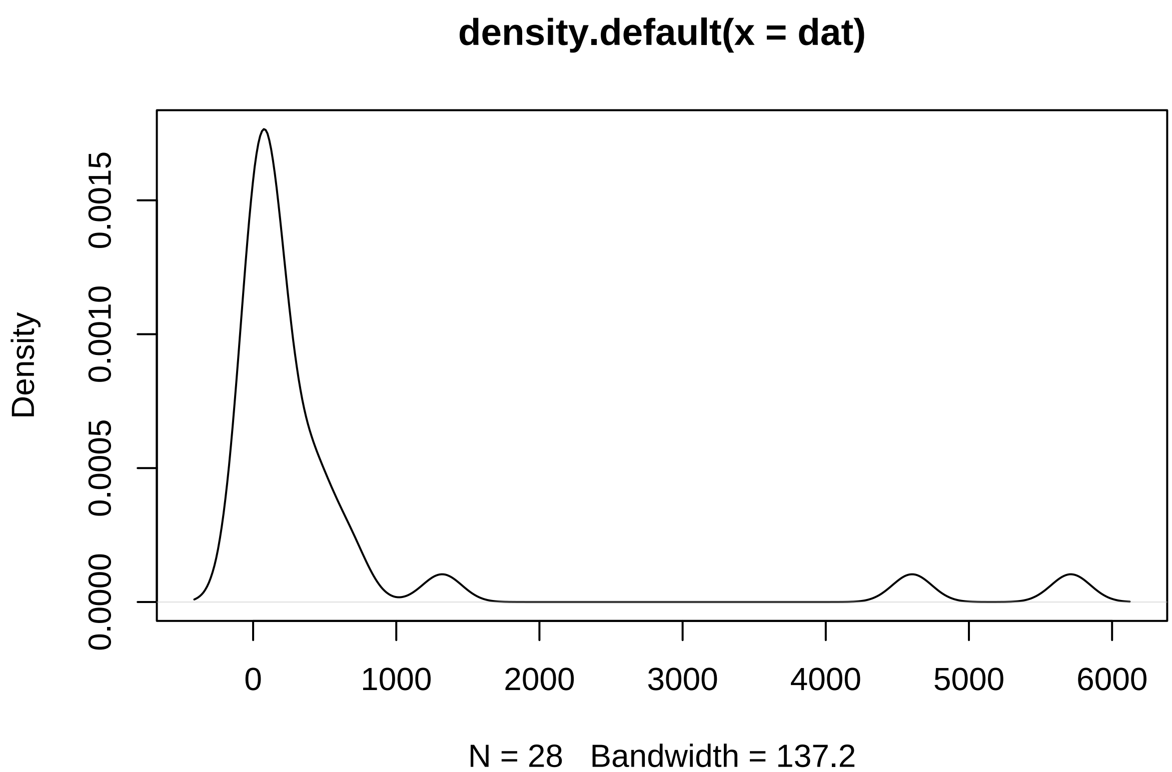
Answer:

The observations from Jan 1940 and Jul 1940 fit really will with estimated gamma distribution. Thus the adequency of gamma model is good.

Exercise 4.39

step 1

```
dat = c(0.4, 1.0, 1.9, 3.0, 5.5, 8.1, 12.1, 25.6,
 115.0,      119.5,   154.5,  157.0,   175.0,
 419.0,      423.0,   440.0,  655.0,   680.0,
  50.0,       56.0,    70.0,  115.0,
 179.0,      180.0,   406.0,
1320.0,     4603.0,  5712.0)
plot(density(dat))
```



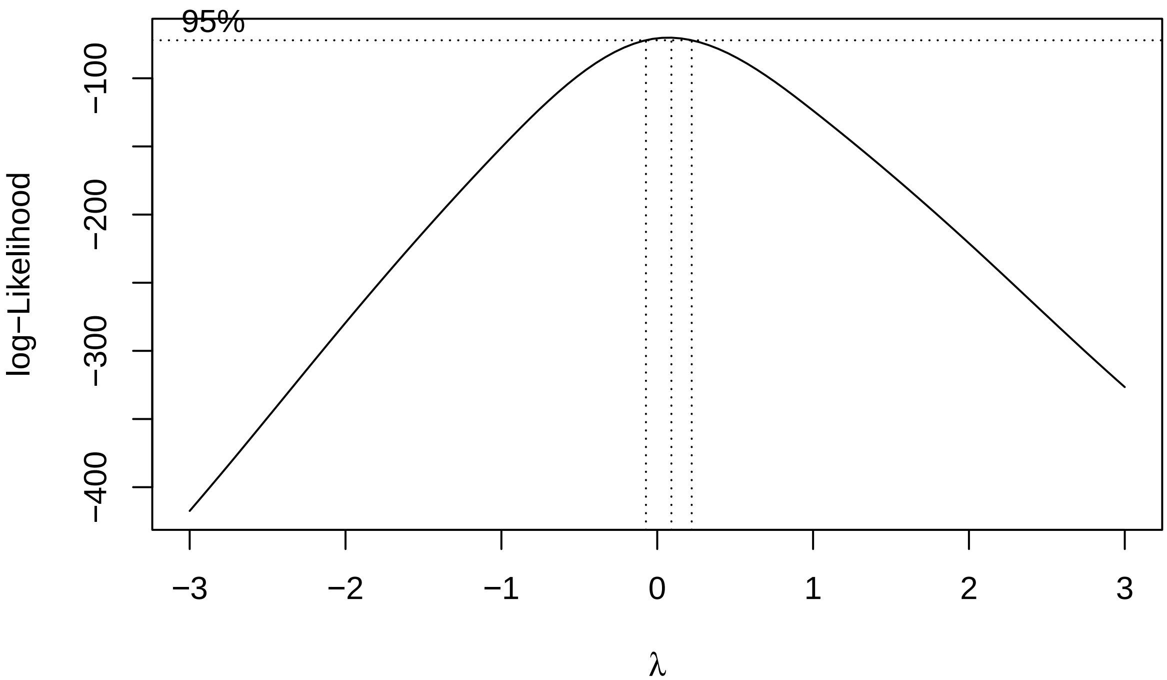

```
shapiro.test(dat)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dat
## W = 0.45173, p-value = 3.763e-09
```

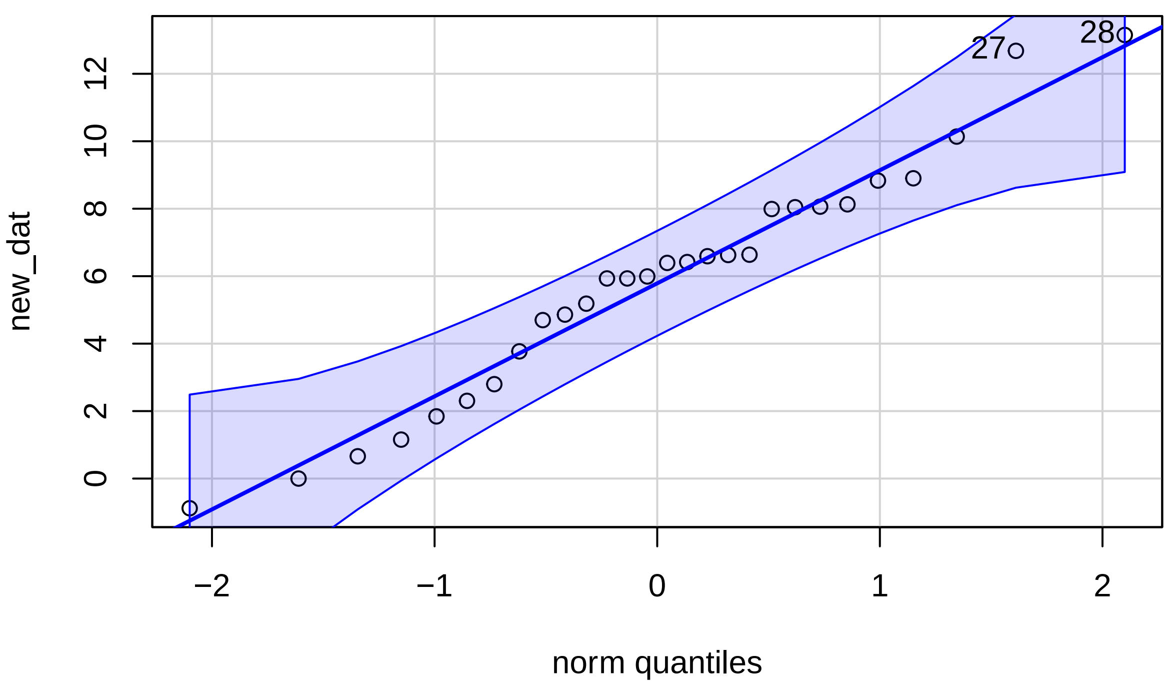
Do Shaprio test to raw data, the p-value is 3.763e-09, which is far below the significance level. The result rejects the null hypothesis of normality, thus it is necessary to conduct Box Cox transformation.

step 2

```
full_model = lm(dat~1)
library(MASS)
bc = boxcox(full_model, lambda = seq(-3, 3))
```




```
lambda = bc$x[which(bc$y == max(bc$y))]
# the best transformation lambda = 0.09090909
new_dat = ((dat^lambda-1)/lambda)
qqPlot(new_dat, dist = "norm")
```



```
## [1] 28 27
```

Answer:

The best transformation lambda = 0.09090909, and after transformation, all observation points are in the normal distribution confidence interval.

step 3

```
# Conduct Shaprio test again
shapiro.test(new_dat)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  new_dat
## W = 0.9724, p-value = 0.6462
```

The p-value of Shaprio test = 0.6462, which is far higher than the significance level, thus the test passes.