STA355A2

Yichen Zhu, 1001421115

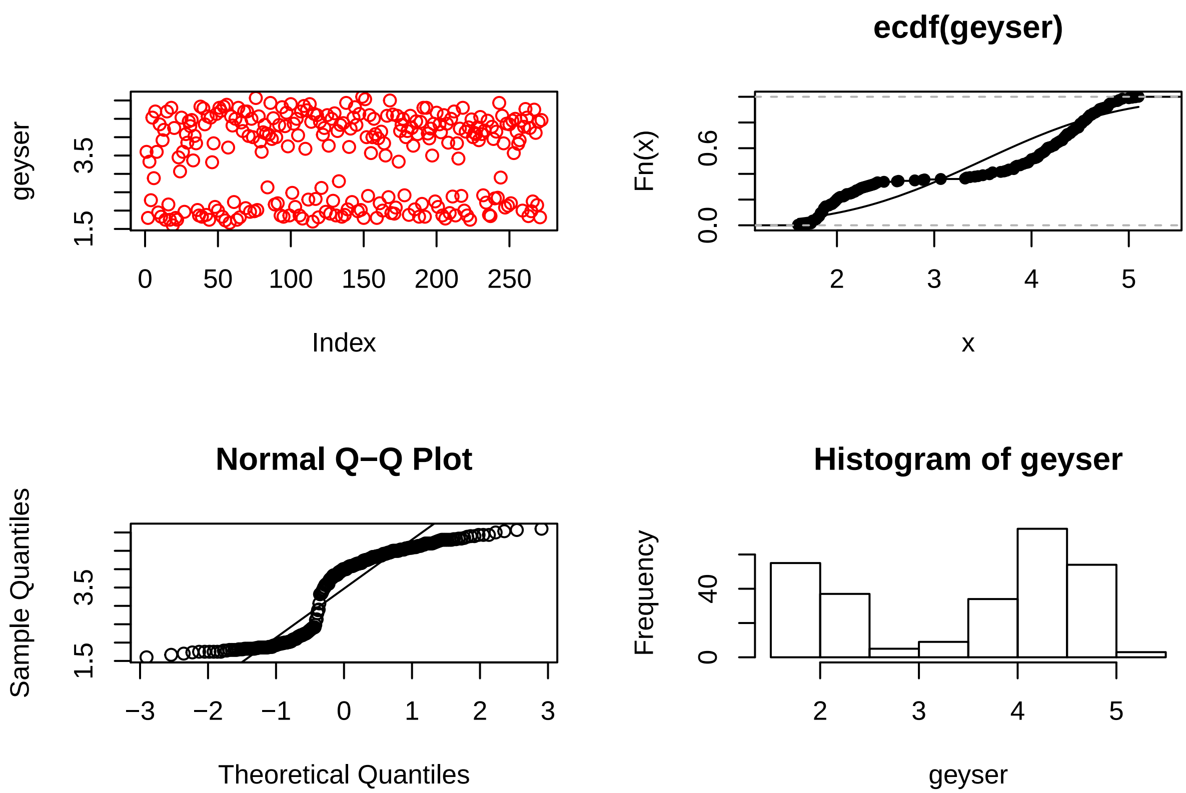
```
geyser = scan("/home/yiche/Desktop/github/UofT_course/STA355/geyser.txt");
length(geyser)
```

```
## [1] 272
```

```
sort_geyser = sort(geyser)
par(mfrow = c(2,2))
plot(geyser, col="red")
plot(ecdf(geyser))
lines(sort_geyser, pnorm(sort_geyser, mean(geyser), sqrt(var(geyser))))
x.norm = rnorm(1000, mean(geyser), var(geyser))
qqnorm(geyser); qqline(x.norm);
shapiro.test(geyser)
```

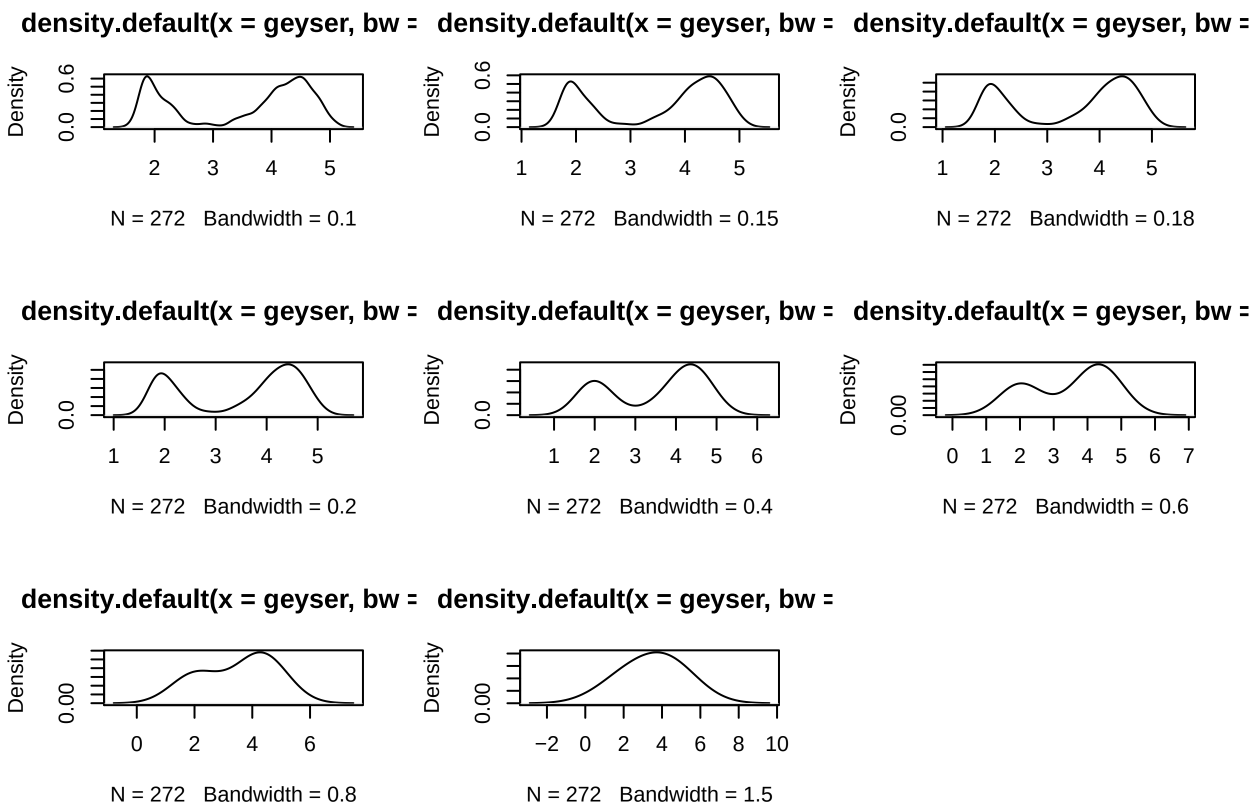
```
##
##  Shapiro-Wilk normality test
##
## data:  geyser
## W = 0.84592, p-value = 9.036e-16
```

```
hist(geyser)
```




```
bw = c(0.1, 0.15, 0.18, 0.2, 0.4, 0.6, 0.8, 1.5)
plot_density = function(bw) {
  for (i in bw){
```

```
    plot(density(geyser, bw = i))
  }
}
par(mfrow = c(3,3))
plot_density(bw)
```



```r
library("exptest")
library("mixtools")
```

```
## mixtools package, version 1.1.0, Released 2017-03-10
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
```

```r
counter1 = 1; exp1 = 0
counter2 = 1; exp2 = 0
for(i in geyser) {
  u = runif(1, 2.5,3.5)
  if(i < 3) {
    exp1[counter1] = i
    counter1 = counter1 + 1
  } else {
    exp2[counter2] = i
    counter2 = counter2 + 1
  }
}
theta1= (counter1-1)/length(geyser)
theta2 = (counter2-1)/length(geyser)
theta1; theta2
```

```
## [1] 0.3566176
```

```
## [1] 0.6433824
```

```r
exp1.mean = round(mean(exp1),3); exp2.mean = round(mean(exp2),3);
exp1.mean; exp2.mean
```

```
## [1] 2.038
```

```
## [1] 4.291
```

```r
std1 = 0; std2 = 0
for (i in 1:counter1 - 1) {
  std1[i] = (exp1[i] - exp1.mean)^2
}
for (i in 1:counter2 - 1) {
  std2[i] = (exp2[i] - exp2.mean)^2
}
var1 = round(sum(std1)/(counter1-1),3); var1
```

```
## [1] 0.07
```

```r
var2 = round(sum(std2)/(counter2-1),3); var2
```

```
## [1] 0.168
```

From the histgram we can see there are two peak and one pit in the graph, we use the pit as edge of two normal distribution, for any number less than 3 it is belongs to first normal distribution, for any number larger than 3 it is belongs to second normal distribution.
We calculate the mean and variance for the first and second distribution respectively and we obtains

$$\mu_1 = 2.04, \mu_2 = 4.29, \sigma_1 = 0.07, \sigma_2 = 0.168$$

. And we estimate theta as the portion of whole data belongs to first normal distribution and second normal distribution, we get

$$\theta_1 = 0.36, \theta_2 = 0.64$$

We can see the mean and theta are approximately matches to the frequency graph for geyser data.
In order to check the accurcy of our approximation, we use EM algorithm the obtains the result.

```
#use em to verify.
library("mixtools")
myEM = normalmixEM(geyser, mu = c(exp1.mean, exp2.mean),lambda = c(theta1, theta2), sigma = c(1, 1))
```

```
## number of iterations= 21
```

```
myEM$mu; myEM$lambda; (myEM$sigma)^2
```

```
## [1] 2.018609 4.273344
```

```
## [1] 0.3484049 0.6515951
```

```
## [1] 0.05551813 0.19102334
```

We can see the approximation use EM are quite close to our approximation, the difference might cause by insufficient sample. Therefore our approximation are reasonable.

```r
library("stats4")
income = scan("/home/yiche/Desktop/github/UofT_course/STA355/incomes.txt");
n = length(income);n
```

```
## [1] 200
```

```r
pnorm(1.96, 0, 1)
```

```
## [1] 0.9750021
```

```r
head(income)
```

```
## [1] 25572 67106 12365 14006 12692 28511
```

```r
income.log = log(income)
```

```r
pietra.exp.test(income, nrepl = 5000)
```

```
##
##  Test for exponentiality based on the Pietra statistic
##
## data:  income
## Pn = 0.34196, p-value = 0.134
```

```r
x = income
theta = NULL
for (i in 1:200) {
  xi = x[-i]
  theta = c(theta, log(mean(xi)))
}
jaceknife_se = sqrt(199*sum((theta - mean(theta))^2)/200);
```

d)
We knoe that the normal distribution is:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Loglikelihood of normal distribution is:

$$-\frac{n}{2}loog(2\pi) - \frac{n}{2}log(\sigma^2) - \frac{\sum(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{\partial}{\partial\sigma^2} = -\frac{n}{2\sigma^2} + \frac{\sum(x_i-\mu)^2}{2\sigma^4}$$

Use MLE:

$$-\frac{n}{2\sigma^2} + \frac{\sum(x_i-\mu)^2}{2\sigma^4} = 0$$

$$\sigma^2 = \frac{\sum(x_i-\mu)^2}{n}$$

$$\frac{\partial}{\partial(\sigma^2)^2} = \frac{n}{2\sigma^4} - \frac{\sum(x_i-\mu)^2}{2\sigma^6}$$

After subsistution of mle variance we obtain:

$$\frac{\partial}{\partial(\sigma^2)^2} = -\frac{n}{2\sigma^4}$$

We can get estimated std err using fisher information:

$$se(\sigma^2) = \frac{\partial}{\partial\sigma^4} = (\frac{n}{2\sigma^4})^{-\frac{1}{2}}$$

```
D(expression(2*pnorm(sigma/2)-1),"sigma")
```

```
## 2 * (dnorm(sigma/2) * (1/2))
```

$$se(P(\sigma^2)) = |P'(\sigma^2)|se(\sigma^2)$$

Where

$$P'(\sigma^2) = dnorm(\frac{\sigma^2}{2})$$

Then we have

$$se(P(\sigma^2)) = |dnorm(\frac{\sigma^2}{2})| * (\frac{n}{2\sigma^4})^{-\frac{1}{2}}$$

We use income data to calculate std error:

```
mean_logincome = mean(income.log); mean(income.log)
```

```
## [1] 10.44447
```

```
mle_var = sum((income.log - mean_logincome)^2)/length(income.log); mle_var
```

```
## [1] 0.7492891
```

```
se_P = abs(dnorm(mle_var/2)*(length(income.log)/(2*(mle_var)^2))^(-0.5)); se_P
```

```
## [1] 0.02786641
```