Final_677

Frank

4/30/2019

Statistics and the Law

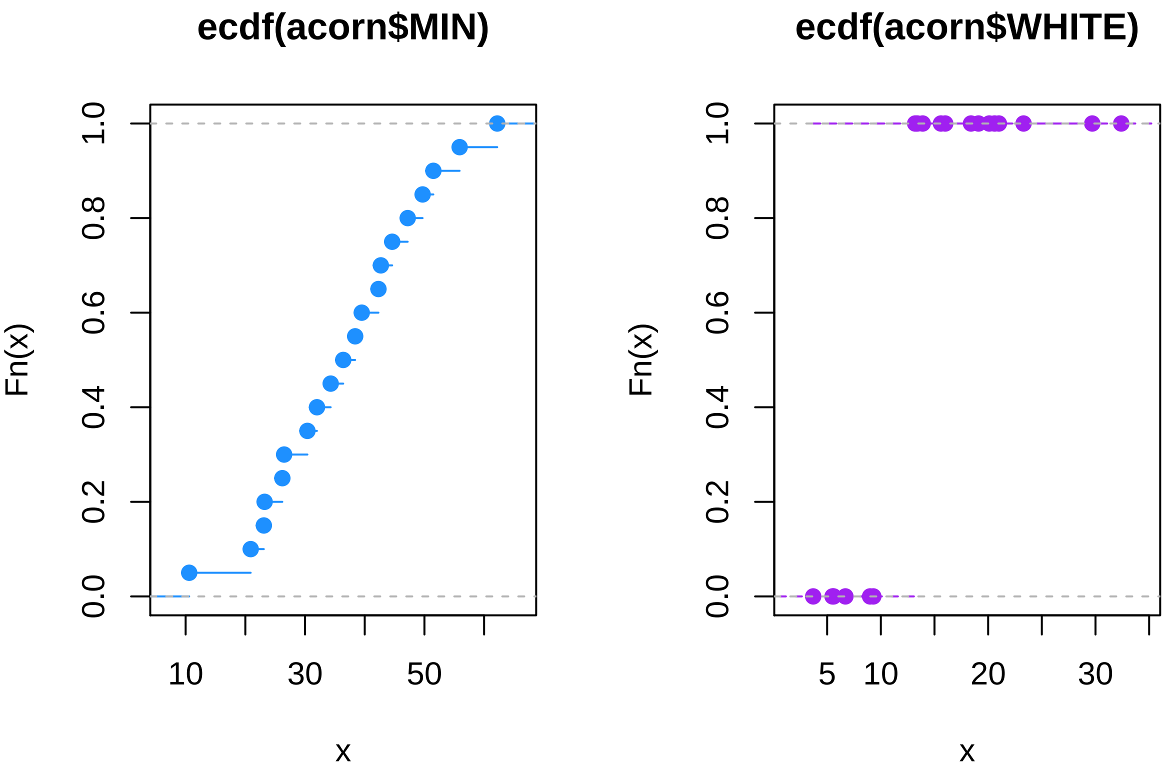
In this question, we need to know if there is any difference between the rates of mortgage application refusals of wihte applicants and minority applicants. That is to say, we need to find a way to test if the refusals of wihte applicants are stochastically smaller than the other sample. Therefore, I'm gonna use Kolmogorov-Smirnov Test.

```
#load
acorn <- read_csv("/Users/yifudong/Desktop/R and Python/acorn.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1], 'X2' [2], 'X3' [3],
## 'X4' [4], 'X5' [5]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_character(),
##   X4 = col_character(),
##   X5 = col_character()
## )
```

```
#clean data
colnames(acorn) <- acorn[1,]
acorn <- acorn[2:21,]


#white applicants versus minor applicants.
par(mfrow=c(1,2))
plot(ecdf(acorn$MIN), col="dodgerblue")
plot(ecdf(acorn$WHITE),  lty="dashed", col="purple")
```



```
#KSTEST
ks.test(jitter(as.numeric(acorn$MIN)), jitter(as.numeric(acorn$WHITE)), alternative = 'l')
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(as.numeric(acorn$MIN)) and jitter(as.numeric(acorn$WHITE))
## D^- = 0.75, p-value = 1.301e-05
## alternative hypothesis: the CDF of x lies below that of y
```

```
#POWER TEST
pwr.t.test(n = 20, 1.98,sig.level = 0.05, power = NULL, type = c("two.sample"))
```

```
##
##      Two-sample t test power calculation
##
##               n = 20
##               d = 1.98
##       sig.level = 0.05
##           power = 0.9999824
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Thus, we can conclude from test result that the discrimination exists between these two groups of people.

Comparing Suppliers

For this question, we need to figure out if there is a school producing ornithopters that statistically have higher quality than another schools. Therefore, we can divide it into two parts: “flies” and “looks good”.

Then for each school, we can create two lists of binary numbers for “flies” and “looks good” respectivley. If they have a flying art, I will add a “1” to their lists of “flies” and “looks good”. If they have a display art, I will add a “1” to the list of “looks good” and 0 to the list of “flies”. If they have a dead bird, I will add “0”s to their lists of “flies” and “looks good”.

```
#flies
area51 <- c(rep(0,35),rep(1,89))
bdv <- c(rep(0,20), rep(1,62))
giffen <- c(rep(0,51),rep(1,119))


par(mfrow=c(2,2))
plot(ecdf(area51), col="red")
plot(ecdf(giffen), lty="dashed", col="green")
plot(ecdf(bdv),  col="dodgerblue")



ks.test(jitter(giffen),jitter(bdv),alternative="l")
```

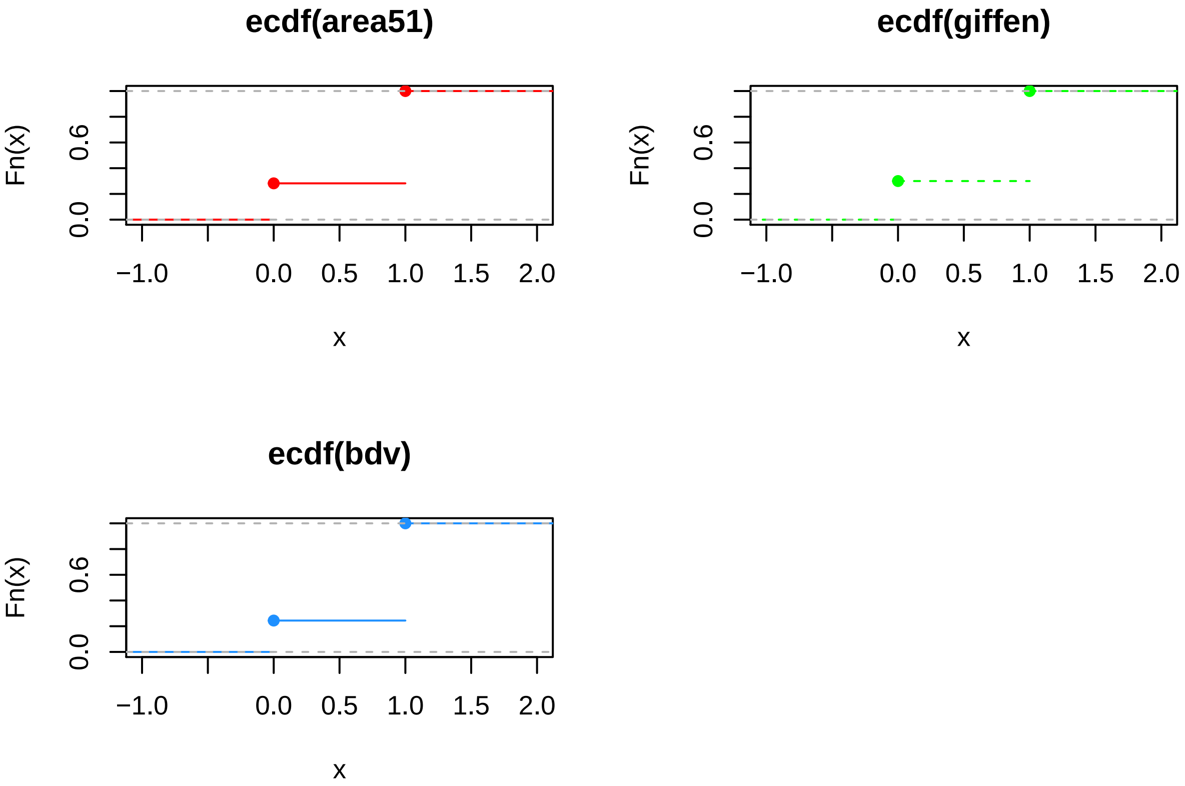
```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(giffen) and jitter(bdv)
## D^- = 0.027834, p-value = 0.9179
## alternative hypothesis: the CDF of x lies below that of y
```

```
ks.test(jitter(bdv),jitter(area51),alternative="l")
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(bdv) and jitter(area51)
## D^- = 0.11231, p-value = 0.2879
## alternative hypothesis: the CDF of x lies below that of y
```

```
ks.test(jitter(area51),jitter(giffen),alternative="greater")
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(area51) and jitter(giffen)
## D^+ = 0.076565, p-value = 0.4314
## alternative hypothesis: the CDF of x lies above that of y
```



From the result above, we know that in terms of "Flies", they produce the same quality. Now let's look at the "Looks good":

```r
area51 <- c(rep(0,12),rep(1,89+23))
bdv <- c(rep(0,8), rep(1,62+12))
giffen <- c(rep(0,21),rep(1,119+30))


par(mfrow=c(2,2))
plot(ecdf(area51), col="red")
plot(ecdf(giffen), lty="dashed", col="green")
plot(ecdf(bdv),  col="dodgerblue")


ks.test(jitter(giffen),jitter(bdv),alternative="l")
```

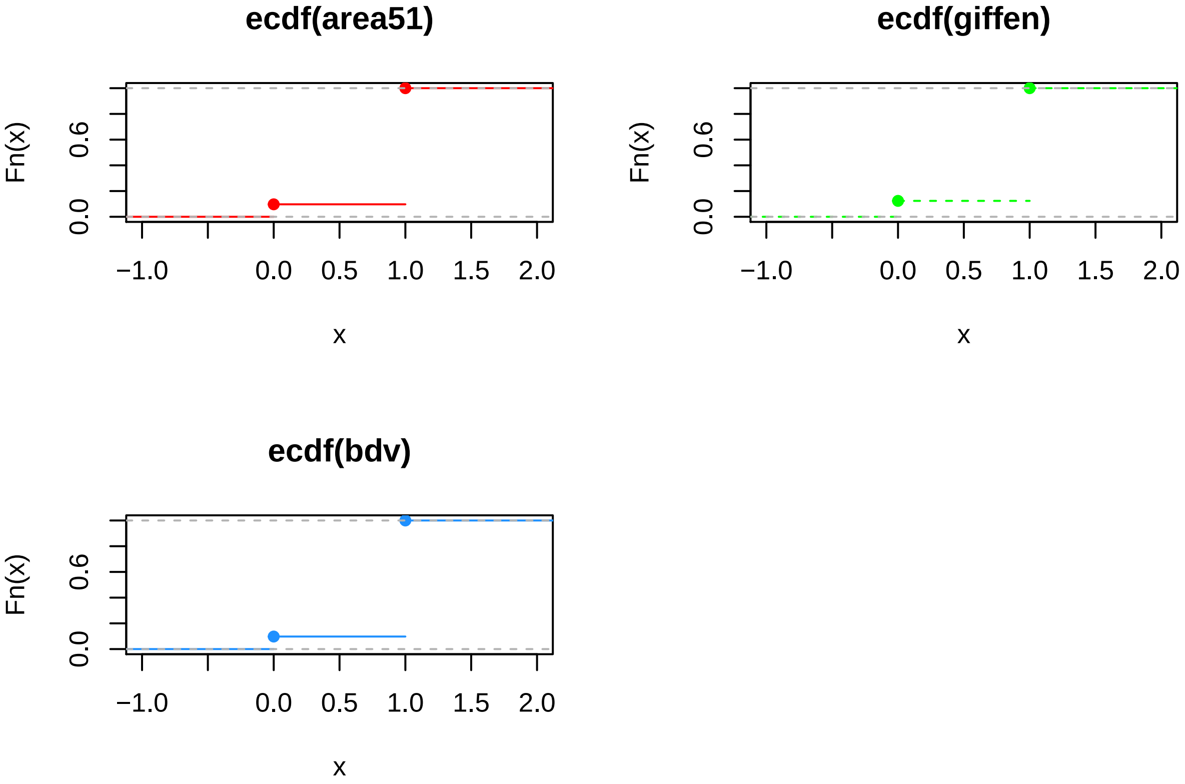
```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(giffen) and jitter(bdv)
## D^- = 0.013199, p-value = 0.9809
## alternative hypothesis: the CDF of x lies below that of y
```

```r
ks.test(jitter(bdv),jitter(area51),alternative="l")
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(bdv) and jitter(area51)
## D^- = 0.062746, p-value = 0.678
## alternative hypothesis: the CDF of x lies below that of y
```

```r
ks.test(jitter(area51),jitter(giffen),alternative="greater")
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  jitter(area51) and jitter(giffen)
## D^+ = 0.082732, p-value = 0.3747
## alternative hypothesis: the CDF of x lies above that of y
```



Therefore, now we can concluded that these three schools all produce ornithopters with same quality.

How deadly are sharks?

```r
library(readr)
sharkattack <- read_csv("/Users/yifudong/Desktop/R and Python/sharkattack.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Date = col_character(),
##   Country = col_character(),
##   `Country code` = col_character(),
##   Type = col_character(),
##   Continent = col_character(),
##   Hemisphere = col_character(),
##   Activity = col_character(),
##   Fatal = col_character()
## )
```

```
sharkattack_us <- sharkattack%>%filter(sharkattack$`Country code`=='US')
sharkattack_au <- sharkattack%>%filter(sharkattack$`Country code`=='AU')

#remove "unknown"
sharkattack_us <- sharkattack_us%>%filter(sharkattack_us$Fatal!="UNKNOWN")
sharkattack_au <- sharkattack_au%>%filter(sharkattack_au$Fatal!="UNKNOWN")

#transfer from "character" to "numeric"
sharkattack_us$Fatalnumeric <- ifelse(sharkattack_us$Fatal == "Y", 1, 0)
sharkattack_au$Fatalnumeric <- ifelse(sharkattack_au$Fatal == "Y", 1, 0)

table(sharkattack_au$Fatalnumeric)[2]
```

```
##   1
## 318
```

```
#power analysis

##effect size
effectsize <- ES.h(mean(sharkattack_au$Fatalnumeric), mean(sharkattack_us$Fatalnumeric))

library(pwr)
pwr.2p2n.test(h=effectsize,n1 = dim(sharkattack_au)[1], n2 = dim(sharkattack_us)[1], sig.level = 0.05, a
```

```
##
##      difference of proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.4137712
##              n1 = 1197
##              n2 = 2012
##       sig.level = 0.05
##           power = 1
##     alternative = greater
##
## NOTE: different sample sizes
```

```
prop.test(x=c(318,217), n=c(1197,2012),alternative = "greater")
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  c(318, 217) out of c(1197, 2012)
## X-squared = 133.41, df = 1, p-value < 2.2e-16
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.1332633 1.0000000
## sample estimates:
##    prop 1    prop 2
## 0.2656642 0.1078529
```

First, I used power analysis function “pwr.2p2n.test” which is sepecifically used for 2 proportions with 2 different sample size. Power analysis allows us to determine the sample size required to detect an effect of a given size with a given degree of confidence. Then I used 2proportion test to check if the two proportions are different sigfinicantly. We can see from the results that the P-value is less than 2.2e-16, on the other hand, the sample estimates of prop1(shark attack proportion in Australia) is 0.2656, while the prop2(shark attack proportion in US) is 0.107 which is much less. Therefore, we can conclude that the prportions of shark attacks in Australia is significantly greater than the prportions of shark attacks in the US.

Power analysis

The arcsine transformation is calculated as the arcsine of the square root of the proportion. When doing power analysis, if we directly use the differences of two proportions to be the ES index, we might find that the power to detect (0.65-0.45) is 0.48, while the power to detect (0.25-0.05) is 0.82. Therefore, directly using difference will not provide a scale of euqal unit of detectability.

On the other hand, the arcsine transformation is calculated as two times the arcsine of the square root of the proportion, which will make the scale stop at pi/2. This transformation is essentially linear over the range of 0.3–0.7, but with more curvature near the ends. Therefore, the equal differences between the results of arcsine transformation are equally detectable. In other words, using arcsine transformation can help the detectability be independent from whether the proportions fall around the middle or on the side of their possibility range.

Estimartors

Use the Method of Moments and MLE to find estimators as described in these three cases.

Exponential

X1, · · · , Xn are independent draws from an exponential distribution, exp(λ). Find the MLE of λ.

Expotential distribution:

$$f(x_i; \lambda) = \lambda e^{-\lambda x}$$

Therefore,

MLE of λ:

$$L(\lambda; x_1, x_2, ..., x_n) = f(x_1)f(x_2)...f(x_n) = \lambda^n e^{-\lambda \sum_{i=0}^{n} x_i}$$

Therefore,

$$l(\lambda; x1, ..., xn) = nlog(\lambda) - \lambda \sum_{i=0}^{n} x_i \Rightarrow \Delta l / \Delta\lambda = n/\lambda - \sum_{i=0}^{n} x_i = 0$$

,

Thus,

$$n/\lambda = \sum_{i=0}^{n} x_i \Rightarrow \hat{\lambda} = 1/\bar{x}$$

A new distribution

Method of Moment:

$$E[x] = \int_0^1 x((1-\theta) + 2x\theta)dx = (1-\theta)\int_0^1 xdx + \int_0^1 2\theta x^2 dx = (1-\theta)\frac{1}{2}x^2 \Big|_0^1 + 2\theta\frac{1}{3}x^3 \Big|_0^1 = \frac{1}{2} - \frac{1}{2}\theta + \frac{2}{3}\theta$$

Therefore,

$$E[x] = \frac{1}{2} + \frac{1}{6}\theta \Rightarrow \bar{x} = \frac{1}{2} + \frac{1}{6}\theta \Rightarrow \hat{\theta} = 6\bar{x} - 3$$

MLE:

λ:

$$L(\lambda; x_1, x_2, ..., x_n) = f(x_1)f(x_2)...f(x_n) = ((1-\theta) + 2\theta x_1)....((1-\theta) + 2\theta x_n)$$

Thus,

$$l(\lambda; x1, ..., xn) = log(((1-\theta) + 2\theta x_1)) + .. + log(((1-\theta) + 2\theta x_n)) \Rightarrow \Delta l/\Delta\theta = \sum_{i=1}^{n} \frac{2x_i - 1}{1 - \theta + 2\theta x_i} = 0$$

Rain in Southern Illinois

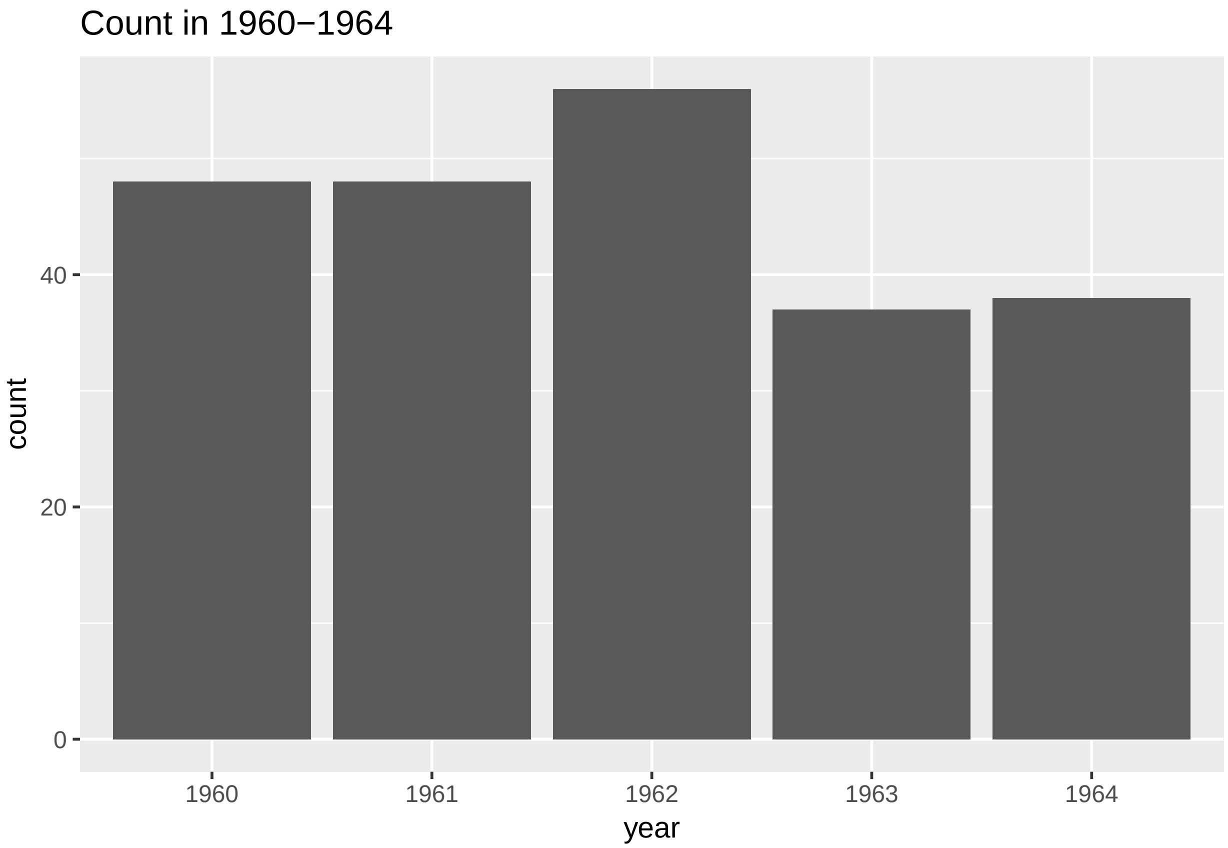
First, we need to construct a dataframe containing all the information:

```r
#import
data60 <- read.table("/Users/yifudong//Desktop/R and Python/ill-60.txt", quote='\'', comment.char="")
data60 <-as.numeric(as.array(data60 [,1]))
data61 <- read.table("/Users/yifudong//Desktop/R and Python/ill-61.txt", quote="\"", comment.char="")
data61<-as.numeric(as.array(data61[,1]))
data62 <- read.table("/Users/yifudong//Desktop/R and Python/ill-62.txt", quote="\"", comment.char="")
data62<-as.numeric(as.array(data62[,1]))
data63<- read.table("/Users/yifudong//Desktop/R and Python/ill-63.txt", quote="\"", comment.char="")
data63<-as.numeric(as.array(data63[,1]))
data64 <- read.table("/Users/yifudong//Desktop/R and Python/ill-64.txt", quote="\"", comment.char="")
data64<-as.numeric(as.array(data64[,1]))

#dataframe
data1960 <- data.frame(data60,rep(1960,48))
colnames(data1960) <- c("value","year")
data1961 <- data.frame(data61,rep(1961,48))
colnames(data1961) <- c("value","year")
data1962 <- data.frame(data62,rep(1962,56))
colnames(data1962) <- c("value","year")
data1963 <- data.frame(data63,rep(1963,37))
colnames(data1963) <- c("value","year")
data1964 <- data.frame(data64,rep(1964,38))
colnames(data1964) <- c("value","year")



rain <- rbind(data1960,data1961,data1962,data1963,data1964)
```

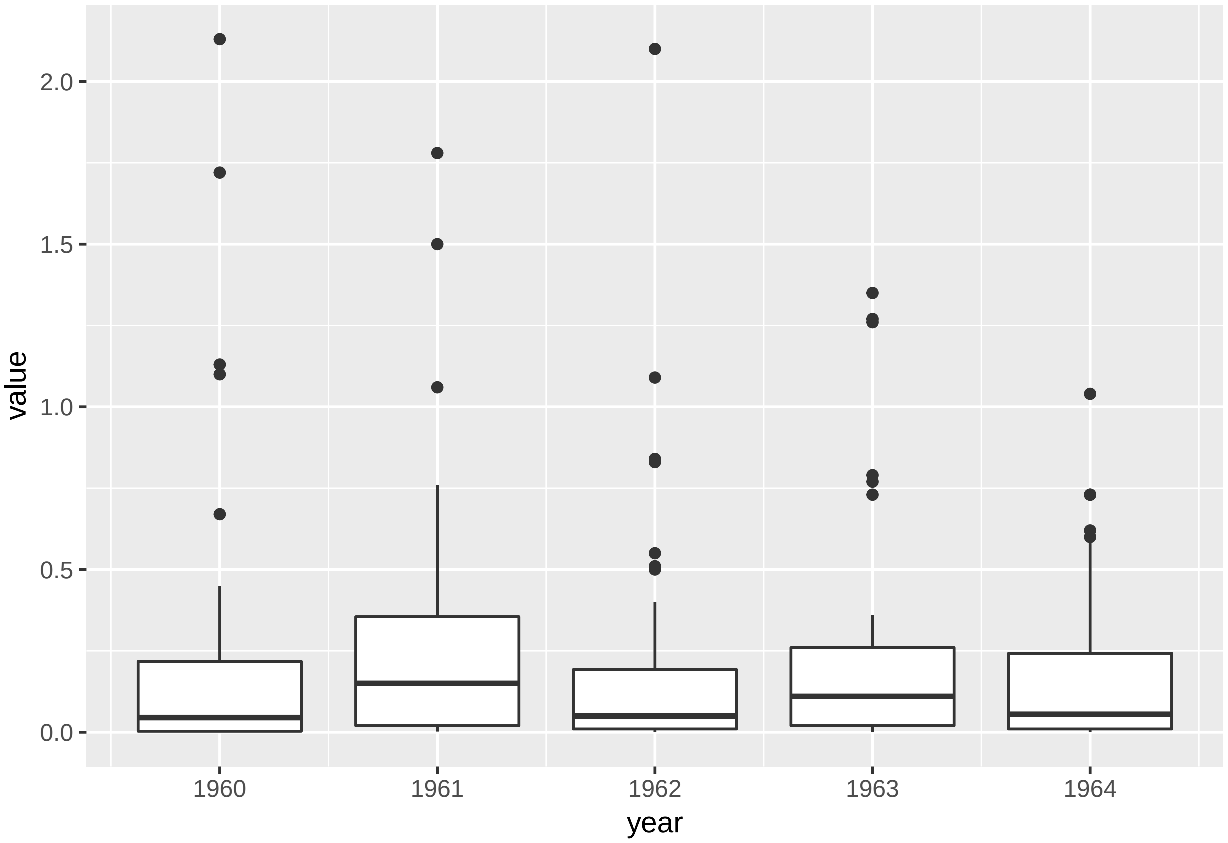
Now we've got a dataframe named "rain" containing all the value and year information, now let's explore the data:

```r
#count
t <- data.frame(table(rain$year))
ggplot(data = t,mapping = aes(x=t$Var1, y=t$Freq)) +
  geom_histogram(stat='identity',bins = 30) + labs(title = "Count in 1960-1964")+xlab("year")+ylab("cou
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

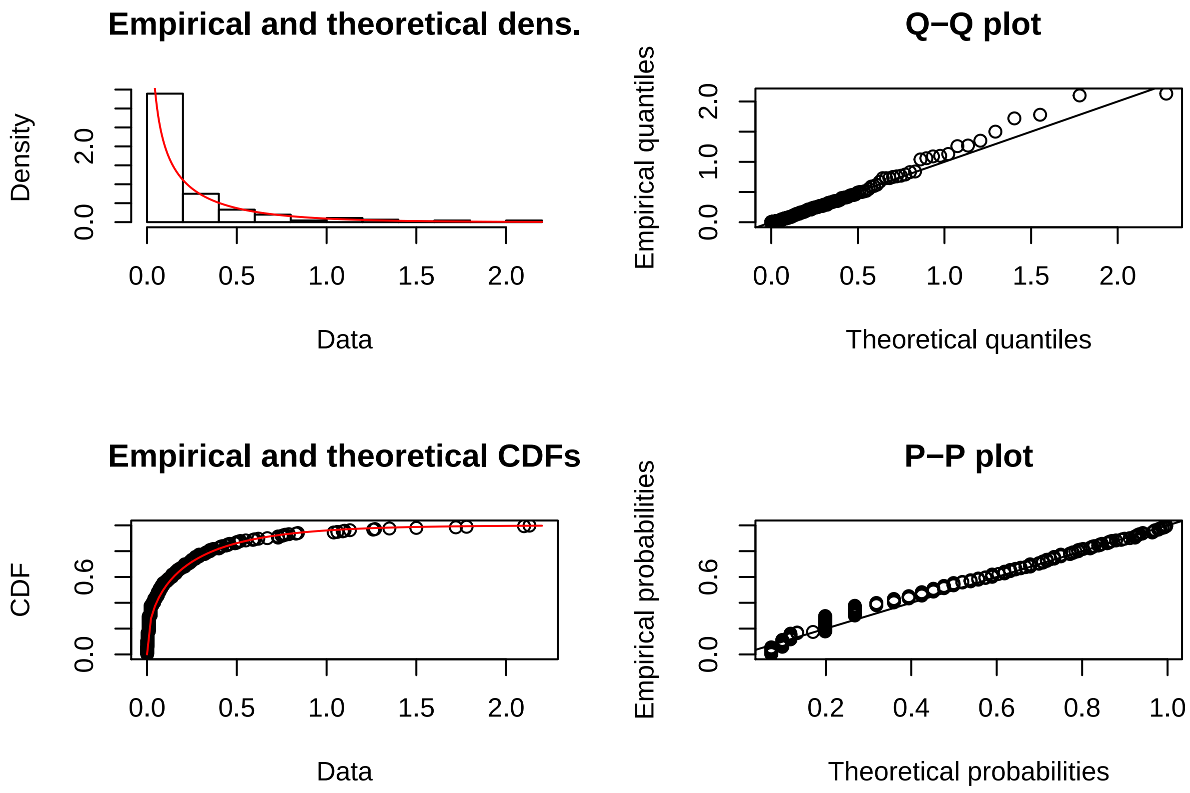


```
#boxplot
ggplot(data=rain, mapping = aes(x=rain$year,y=rain$value,group=rain$year))+
  geom_boxplot()+ylab("value")+xlab("year")
```

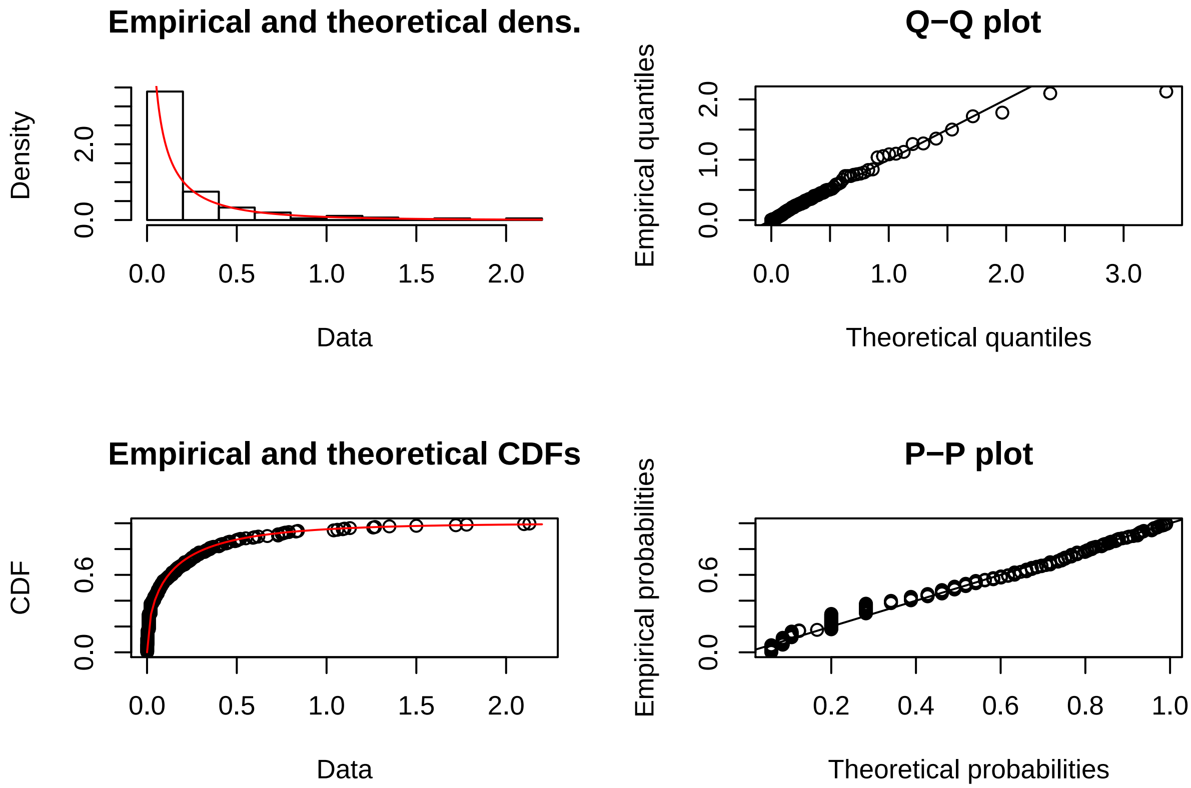


From the plots above, we can know that most storms happens in 1962, but in terms of rainfall, 1961 and 1963 are the wetter years whereas 1961 and 1963 are not the two years that storms happen most frequently. Instead, 1963 is the year with smallest frequency count. Therefore, I think storms will not produced more rain. The relation is not that strong.

```
library(fitdistrplus)
#fitdistrplus
raing  <- fitdist(rain$value, "gamma")
plot(raing)
```



```r
rainw  <- fitdist(rain$value, "weibull")
plot(rainw)
```



From the plots above, I would say gamma distribution looks to fit well under the CDF plot and density plot. But gamma distribution doesn't have such a good performance when looking at QQ plot. Instead, the "weibull" seems to be not worse and even better than gamma distribution in terms of QQ plot. Thus, I think they are right, but they may also want to have a look at the weibull distribution. All in all, I think gamma distribution is good for our modelling and these two distributions are similar.

As for the MoM and MLE:

```r
#mom
rainmom <- fitdist(rain$value, "gamma", method = "mme")
t1 <- bootdist(rainmom)
summary(t1)
```

```
## Parametric bootstrap medians and 95% percentile CI
##          Median      2.5%     97.5%
## shape 0.3940533 0.2691985 0.5280178
## rate  1.7754374 1.1320316 2.5067025
```

```r
#mle
rainmle <- fitdist(rain$value, "gamma", method = "mle")
t2 <- bootdist(rainmle)
summary(t2)
```

```
## Parametric bootstrap medians and 95% percentile CI
##          Median      2.5%     97.5%
## shape 0.4426246 0.3851147 0.5134107
## rate  1.9781197 1.5512504 2.6046387
```

From the result avoe, I would prefer the MLE, because the CI of MLE is more accurate than MoM.

Analysis of decision theory article

Since we have: $P(\delta) = Beta(c,d) = \frac{1}{B(c,d)}\delta^{c-1}(1-\delta)^{d-1}$, which is our prior distribution, we can calcualte out our posterior distribution: $P(\delta|x_1,...,x_n) = P(x_1,...,x_n|\delta)P(\delta)$.

Posterior:

$$P(x_1,...,x_n|\delta) = \prod_{i=1}^{N}\delta^{x_i}(1-\delta)^{x_i}$$

, let $(\beta_s, s \in S) = (0,1)$, then we have:

$$P(x_1,...,x_n|\delta) = \prod_{i=1}^{N}\delta^{x_i}(1-\delta)^{x_i} = \delta^n(1-\delta)^{N-n}$$

So now we can calculate the posterior:

$$P(\delta|x_1,...,x_n) = P(x_1,...,x_n|\delta)P(\delta) = (\frac{1}{B(c,d)}\delta^{c-1}(1-\delta)^{d-1}) * (\delta^n(1-\delta)^{N-n}) = \frac{1}{C}\delta^{c+n-1}(1-\delta)^{N-n+d-1}$$

Thus, we can conclude from the formula that for the posterior beta distribution: $\alpha = c+n, \beta = d+N-n$.

Therefore,

$$E(X|x1...xn) = \frac{\alpha}{\alpha+\beta} = \frac{c+n}{c+d+N}.....(1)$$

.

Since the initial Bayes ruls is:

$$\delta(n) = 0, for E(X|x1..xn) < \alpha$$

$$\delta(n) = \lambda for E(X|x1..xn) = \alpha, where \quad 0 < \lambda < 1$$
$$\delta(n) = 1 for E(X|x1...xn) > \alpha........(2)$$

Thus, we derive equations (10a),(10b),(10c) from equations(1) and (2) as shown above.

Reproduce: #cite: Here I consulted with Mira about how to reproduce

```
theta_n <- function(n, n0, lambda) {
  if (n < n0) {
    theta_n <- 0
  }
  else if (n == n0) {
    theta_n <- lambda
  }
  else {
    theta_n <- 1
  }
  return(theta_n)
}
f <- function(i, beta, N) {
  return(factorial(N) * ((factorial(i) *
                          factorial(N - 1))^(-1)) *
           (beta^i) * ((1 - beta)^(N - i)))
}
# Calculate the walfare of rule theta
E_theta_n <- function(N, beta, n0, lambda) {
  E_theta_n <- 0
  for (i in 1:N) {
    E_theta_n <- E_theta_n + theta_n(i, n0, lambda) * f(i, beta, N)
  }
  return(E_theta_n)
}
# Calculate the regret of rule thata in state s
regretrule <- NULL
regret_rule <- function(n, N, n0, lambda, alpha) {
  beta_s <- c(rep(1, n), rep(0, N - n))
  beta <- n / N
  for (i in 1:N) {
    if (beta_s[i] >= alpha) {
      regretrule[i] <- (beta_s[i] - alpha) *
        (1 - E_theta_n(N, beta, n0, lambda)) # Please cite Mira
    }
    else {
      regretrule[i] <- (alpha - beta_s[i]) *
        (E_theta_n(N, beta, n0, lambda))
    }
  }
  return(regretrule)
}
# Calculate the Minimax-regret rule
max_regret_rule <- NULL
minimax_regret_rule <- NULL
minimax_regret_rule <- function(N, n0, lambda, alpha) {
  for (n in 1:N) {
    maxregret_rule <- max(regret_rule(n, N, n0, lambda, alpha))
    max_regret_rule[n] <- maxregret_rule
  }
  minimax_regret_rule <- min(max_regret_rule)
  return(minimax_regret_rule)
}
```