IMT 573: Problem Set 5 - Statistical Theory

Miloni Desai

Due: Wednesday, November 06, 2019

Collaborators:

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Server.

1. Download the `problemset5.Rmd` file from Canvas or save a copy to your local directory on RStudio Server. Open `problemset5.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset5.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do not need four different visualizations of the same pattern.
4. Collaboration on problem sets is fun and useful, and we encourage it, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. Students are *strongly* encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run without errors, you can do so with the `eval=FALSE` option. (Note: I am also using the `include=FALSE` option here to not include this code in the PDF, but you need to remove this or change it to `TRUE` if you want to include the code chunk.)
7. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the knitted PDF file to `ps5_YourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```r
# Load standard libraries
library(tidyverse)
```

Problem 1: Overbooking Flights

You are hired by *Air Nowhere* to recommend the optimal overbooking rate. It is a small airline that uses a 100-seat plane to carry you from Seattle to, well, nowhere. The tickets cost $100 each, so a fully booked plane generates \$10,000 revenue. The sales team has found that the probability, that the passengers who have paid their fare actually show up is 98%, and individual show-ups can be considered independent. The additional costs, associated with finding an alternative solutions for passengers who are refused boarding are \$500 per person.

(a)Which distribution would you use to describe the actual number of show-ups for the flight? Hint: read OIS ch 3 about distributions. #The scenario outlined is an example of a binomial distribution scenario. The binomial distribution describes the probability of having exactly k successes in n independent Bernoulli trials (here the 100 passengers)with probability of a success p(here 0.98 which is the proability that a passenger who pays actually shows up)

(b)Assume the airline never overbooks. What is it's expected revenue? Expected revenue means expected income from the ticket sales, minus the expected costs related to alternative solutions. #Expect Revenue = Expected Ticket Sales Income - Expected cost of Alternate solutions #Expected Ticket Sales Income = 100 * 100 = \$10,000 (This is the regular sales, no overbooking) #P(passenger who buys a ticket showing up) = 0.98 #P(passenger who buys a ticket not showing up) = 0.02 #Expected cost of Alternate solutions = 2 * 500 = \$1000 #Expected Revenue = \$9000

(c)Now assume the airline sells 101 tickets for 100 seats. What is the probability that all 101 passengers will show up?

```r
dbinom(100, size=101, prob=0.98)
```

```
## [1] 0.2678915
```

#Thus we can see that the proability that all the 101 passengers show up is 0.267 or 26.7%

(d)What are the expected profits (= revenue − expected additional costs) in this case? Would you recommend overbooking over selling just the right number of tickets? #Expected profits in this case = ((101)* 100) - E(x=101,0.98)*500 #Expected Value is E(x) = np* = 101* 0.98 = 98.98 #Thus we have to make arrangements for only the one person who did not show up as opposed to two when the flight was not overbooked. #Thus expected profits = 10100- 1(500) = \$9600 #In this case it looks like it works out to be more profitable for the airline. So I would reccommend overbooking over selling just the right number of tickets.

(e)Now assume the airline sells 102 tickets. What is the probability that all 102 passengers show up?

```r
dbinom(100, size=102, prob=0.98)
```

```
## [1] 0.2732493
```

#Thus we can see that the proability that all the 102 passengers show up is 0.27 or 27%.

(f)What is the probability that 101 passengers – still one too many – will show up?

```r
pbinom(100,size = 101, prob = 0.98)
```

```
## [1] 0.8700328
```

#This gives us the proability of 101 or less than 101 passengers showing up.

(g)Would it be advisable to sell 102 tickets, i.e. is the expected revenue from selling 102 tickets larger than from selling 100 and 101 tickets? #Selling 102 tickets = 102* 100 = \$10200 #E(x = 102) = 102 * 0.98 = 99.96 which is almost 100 # Thus the expected revenue = 10200 - 2*500 = 9200. #Clearly we can see that revenue from selling tickets larger than 102 is more profitable than selling the exact number but is not as profitable as selling 101.

(h)What is the optimal number of seats to sell for the airline? How big are the expected profits? #Ideally a line grapg showing us a curve would help us decide the optimal number of seats for a 100 seater plane with passengers who buy a ticket have a proability of 0.98 to show up. #Based on the proability of overbooking and those passengers actually showing up, I think 101 passengers is the most optimal number of seats to sell and is profitable too. The profits are reasonable. They are 6.67 percent more than if no overbooking was done.

(i)What does it mean that the show-ups are independent? Why is it important? Hint: read about independence in OIS 2.1.6 (2017 version). #The show ups are indepent means that each trial has two outcomes, one that the passenger shows up and the other that he does not, and the independent here means that whether one passenger shows up or not is not dependent on another passenger. That the outcome of each passenger is not dependent on the show or no show of any other passenger. #This is important as it makes sure that no single outcome can influence the outcome of another event and thus proabilities can be calculated without introducing such measure which would be hard to account for mathemtically.

Note: some of the expressions may be hard to write analytically. Feel free to use computer for the calculations, just show the code and explain what you are doing.

Problem 2: The Normal Distribution

In this problem we will explore data and ask whether it is approximately normal. We will consider two different datasets, one on height and one of research paper citations.

(a) Let's start with the human height data.

(1)What level of measurement (nominal, ordered, difference, ratio) does this data on human height use? How should it be measured (e.g. continuous, discrete, positive...)?

```
library(HistData)
data("Galton")
df<-GaltonFamilies
#View(df)
```

#The level of measurement in this data is mostly ratio, when it comes to height of various participants as it as continuous variable and a measure of lenght. Gender is a nominal measure. #It will be measure as a continuous variable.

(2) Read the `fatherson.csv` dataset into R. It contains two columns, father's height and son's height, (in cm). Let's focus on father's height for a moment, (variable `fheight`). Provide a basic description of this variable, for example how many observations do we have? Do we have any missing data?

```
df2 <- read.csv(file = "fatherson.csv.bz2", sep="\t")
#View(df2)
dim(df2)
```

```
## [1] 1078    2
```

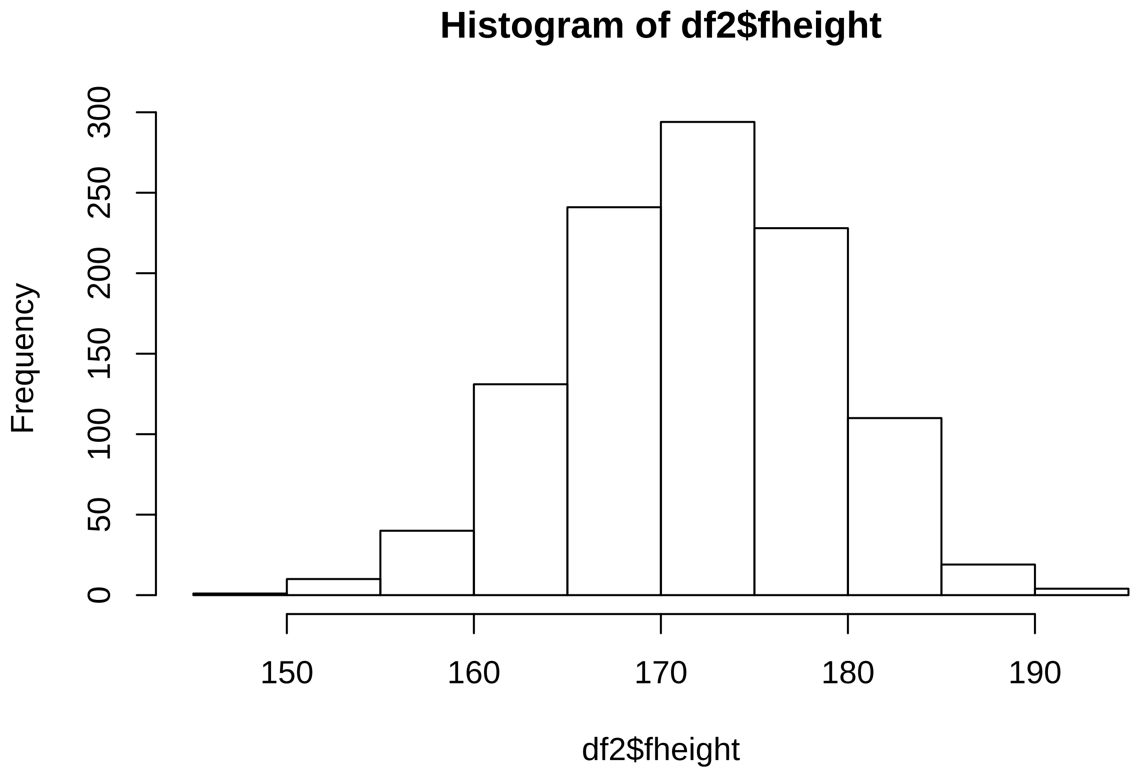
```
which(is.na(df2), arr.ind=TRUE)
```

```
##      row col
```

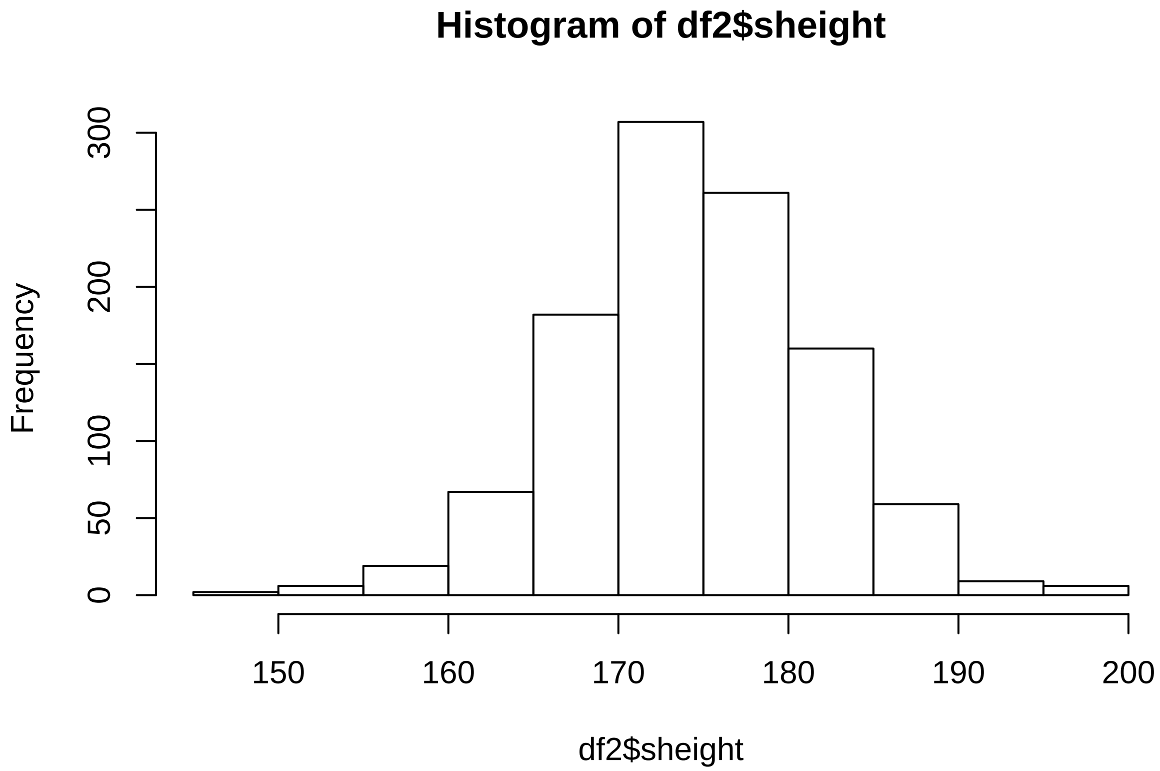
#This variable gives us the height of the father as a continuous variable and possibly in cm looking at the values. Each fathers height has the corresponding height of his son. There are 1078 observations and none of which seem to have missing data.

(3) Compute mean, median, mode, standard deviation and range of the heights. Discuss the relationship between these numbers. Is mean larger than median? Than mode? By how much (in relative terms)? How does standard deviation compare to mean?

```
hist(df2$fheight)
```



```
hist(df2$sheight)
```



#Just getting a better understanding of the distributions of heights

```
mean(df2$fheight)
```

```
## [1] 171.9252
```

```
mean(df2$sheight)
```

```
## [1] 174.4572
```

#The mean height for fathers is 171.92 and for sons is 174.4. We can see that it seems to be as though the next generation is taller than the previous by a small margin of 2.5 cms

```
median(df2$fheight)
```

```
## [1] 172.1
```

```
median(df2$sheight)
```

```
## [1] 174.3
```

#The median height for fathers is 172.1 and for sons is 174.3. The median is slightly more than the mean but very marginally.

```
y <- table(df2$fheight)
y
```

```
##
## 149.9 151.1 151.5 152.5 152.9 153.5 154.4 154.7   155 155.1 155.3 155.7
##     1     2     1     1     1     1     1     1     2     2     1     1
## 156.2 156.7 156.9   157 157.2 157.6 157.8 157.9 158.2 158.4 158.5 158.6
##     1     1     1     1     1     1     3     1     1     1     2     1
## 158.8 158.9 159.2 159.3 159.4 159.5 159.6 159.7 159.8 159.9   160 160.1
##     2     1     3     2     1     1     2     2     2     2     3     3
## 160.2 160.4 160.5 160.6 160.7 160.9 161.3 161.5 161.6 161.7 161.8 161.9
##     2     3     1     2     3     1     1     3     1     3     3     1
##   162 162.1 162.2 162.3 162.4 162.5 162.7 162.8 162.9 163.3 163.4 163.5
##     1     2     4     7     3     2     2     4     3     3     5     3
## 163.6 163.7 163.8 163.9   164 164.1 164.2 164.3 164.4 164.5 164.6 164.7
##     2     5     4     4     3     8     4     6     7     2     5     5
## 164.8 164.9   165 165.1 165.2 165.3 165.4 165.5 165.6 165.7 165.8 165.9
##     4     2     4     6     2     4     1     7     2     3     2     2
##   166 166.1 166.2 166.3 166.4 166.5 166.6 166.7 166.8 166.9   167 167.1
##     3     6     4     3     3    11     3     6     8     3     5     6
## 167.2 167.3 167.4 167.5 167.6 167.7 167.8 167.9   168 168.1 168.2 168.3
##     9     7     5     6     7    10     8     6     1     4     6     3
## 168.4 168.5 168.6 168.7 168.8 168.9   169 169.1 169.2 169.3 169.4 169.5
##     6     1     3     4     4     4     7     6     4     4     5     5
## 169.6 169.7 169.8 169.9   170 170.1 170.2 170.3 170.4 170.5 170.6 170.7
##     7     3     6     5     5     4     2     6     6     8     8     4
## 170.8 170.9   171 171.1 171.2 171.3 171.4 171.5 171.6 171.7 171.8 171.9
##     2     2    11     5    12     3     4     4    10     7     3     3
##   172 172.1 172.2 172.3 172.4 172.5 172.6 172.7 172.8 172.9   173 173.1
##     7     7     8     7     6     4    12     3     7     5     7     3
## 173.2 173.3 173.4 173.5 173.6 173.7 173.8 173.9   174 174.1 174.2 174.3
##     7     6    10     5     8     6     2     6     5     8     4     7
## 174.4 174.5 174.6 174.7 174.8 174.9   175 175.1 175.2 175.3 175.4 175.5
##     5     6     7     2    10     6     4     4     3     5    13     6
## 175.6 175.7 175.8 175.9   176 176.1 176.2 176.3 176.4 176.5 176.6 176.7
##     4     5     4     4     4    11     4     6     3     6     3     4
## 176.8 176.9   177 177.1 177.2 177.3 177.4 177.5 177.6 177.7 177.8 177.9
##     8     4     5     4     5     2     2     7     7     4     6     6
##   178 178.1 178.2 178.3 178.4 178.5 178.6 178.7 178.8 178.9   179 179.1
##     3     3     8     5     7     2     3     4     3     5     5     3
## 179.2 179.3 179.4 179.5 179.6 179.7 179.8 179.9   180 180.1 180.2 180.3
##     4     2     4     3     3     2     1     4     5     2     1     1
## 180.4 180.5 180.6 180.8 180.9   181 181.1 181.2 181.3 181.4 181.5 181.6
##     4     2     3     6     1     2     9     1     4     3     4     4
## 181.8 181.9   182 182.1 182.2 182.3 182.4 182.5 182.6 182.7 182.8 182.9
##     1     2     2     1     1     1     1     1     1     2     2     1
##   183 183.1 183.2 183.3 183.4 183.6 183.7 183.8 183.9   184 184.1 184.2
##     3     2     1     2     3     1     5     6     4     3     3     1
## 184.3 184.4 184.7 184.8 184.9 185.2 185.3 185.5 185.6 185.9 186.1 186.5
##     2     3     1     3     4     1     1     4     1     2     3     1
## 186.7 187.6 187.7   189 189.7   190 190.3 190.4 190.9 191.6
##     1     1     1     1     1     1     1     1     1     1
```

```
mode1 = names(y)[which(y==max(y))]
fmode = as.numeric(mode1)
fmode
```

```
## [1] 175.4
```

#The mode for the fathers age is 175.4 which is more than the mean and the median by almost 1.02 times.

```
z <- table(df2$sheight)
z
```

```
##
## 148.6 149.3 151.8 151.9 152.5 154.4 154.7   155 155.1 155.5 155.7 156.6
##     1     1     1     1     1     1     1     1     1     1     1     1
## 156.7 156.9 157.2 157.5   158 158.2 158.4 158.7 158.8 159.4 159.5 159.7
##     1     1     1     1     1     1     1     1     1     1     2     1
## 159.8   160 160.3 160.5 160.6 160.7 160.8 160.9 161.2 161.3 161.4 161.5
##     1     1     1     1     1     1     1     1     2     1     2     1
## 161.6 161.7 161.8 161.9   162 162.1 162.3 162.4 162.5 162.6 162.7 162.8
##     1     1     1     2     1     1     1     4     3     1     1     3
##   163 163.1 163.2 163.3 163.4 163.5 163.6 163.7 163.9 164.1 164.3 164.4
##     1     1     1     3     1     3     1     1     1     3     3     1
## 164.5 164.6 164.7 164.8 164.9   165 165.1 165.3 165.4 165.5 165.6 165.7
##     2     5     2     2     1     3     1     3     4     4     1     4
## 165.8 165.9   166 166.2 166.3 166.4 166.5 166.6 166.7 166.8 166.9   167
##     2     2     2     4     2     2     4     3     5     4     4     3
## 167.1 167.2 167.3 167.4 167.5 167.6 167.7 167.9   168 168.1 168.2 168.3
##     3     4     1     4     2     2     3     4     4     2     3     5
## 168.4 168.5 168.6 168.7 168.8 168.9   169 169.1 169.2 169.3 169.4 169.5
##     5    11     5     6     5     2     5     3     4     1     2     6
## 169.6 169.7 169.8 169.9   170 170.1 170.2 170.3 170.4 170.5 170.6 170.7
##     7     8     4     4    13     5     5     5     5     4    10     6
## 170.8 170.9   171 171.1 171.2 171.3 171.4 171.5 171.6 171.7 171.8 171.9
##     8     4     2     4     5     8     6     8     5     5     3     6
##   172 172.1 172.2 172.3 172.4 172.5 172.6 172.7 172.8 172.9   173 173.1
##     6     8     2     5     7     7     7     5     7    11     7     7
## 173.2 173.3 173.4 173.5 173.6 173.7 173.8 173.9   174 174.1 174.2 174.3
##    11     6     8     4     7     7     5     3     6     6    13    10
## 174.4 174.5 174.6 174.7 174.8 174.9   175 175.1 175.2 175.3 175.4 175.5
##     6     7     8     3     8     1     5     6     9     6     5     8
## 175.6 175.7 175.8 175.9   176 176.1 176.2 176.3 176.4 176.5 176.6 176.7
##     4     6     7     9     9     6    11     5     3     6     5     4
## 176.8 176.9   177 177.1 177.2 177.3 177.4 177.5 177.6 177.7 177.8 177.9
##     4     6     4     8     5     3     8     2     7     9     6     4
##   178 178.1 178.2 178.3 178.4 178.5 178.6 178.7 178.8 178.9   179 179.1
##     3     3     3     5     6     9     5     4     8     3     4     8
## 179.2 179.3 179.4 179.5 179.6 179.7 179.8 179.9   180 180.1 180.2 180.3
##     2     2     2     3     2     6     1     2     5     7     5     4
## 180.4 180.5 180.6 180.7 180.8 180.9   181 181.1 181.2 181.3 181.4 181.5
##     4     2     2     4     6     7     6     2     5     8     6     3
## 181.6 181.7 181.8 181.9   182 182.2 182.3 182.4 182.5 182.6 182.7 182.8
##     5     5     3     5     4     1     3     4     1     3     2     2
## 182.9   183 183.1 183.2 183.3 183.4 183.5 183.6 183.7 183.8 183.9   184
##     4     4     3     1     3     1     1     2     3     3     1     4
## 184.1 184.2 184.3 184.4 184.5 184.6 184.8 184.9   185 185.1 185.4 185.5
##     1     2     5     2     4     1     1     4     1     4     1     3
## 185.6 185.8 185.9   186 186.1 186.4 186.5 186.6 186.7   187 187.1 187.2
##     3     1     5     1     2     3     1     1     1     3     1     3
## 187.3 187.4 187.5 187.6 187.7 187.8 187.9   188 188.5 188.7 188.8 188.9
##     1     1     4     2     1     3     1     1     1     2     2     1
##   189 189.2 189.5 189.6 189.7 190.7 190.8 191.4 191.9   192 192.2 192.3
##     1     2     1     1     1     1     1     1     1     1     1     1
## 193.3 193.9 195.1   196 196.1 196.2 198.7   199
##     1     1     1     1     1     1     1     1
```

```r
mode2 = names(z)[which(z==max(z))]
mode2
```

```
## [1] "170"   "174.2"
```

```r
smode = as.numeric(mode2)
smode
```

```
## [1] 170.0 174.2
```

#The sons height has two means 170 and 174.2. While one is lower than the mean and median the other is almost equal to the mean and median.

```r
sd(df2$fheight)
```

```
## [1] 6.972346
```

```r
sd(df2$sheight)
```

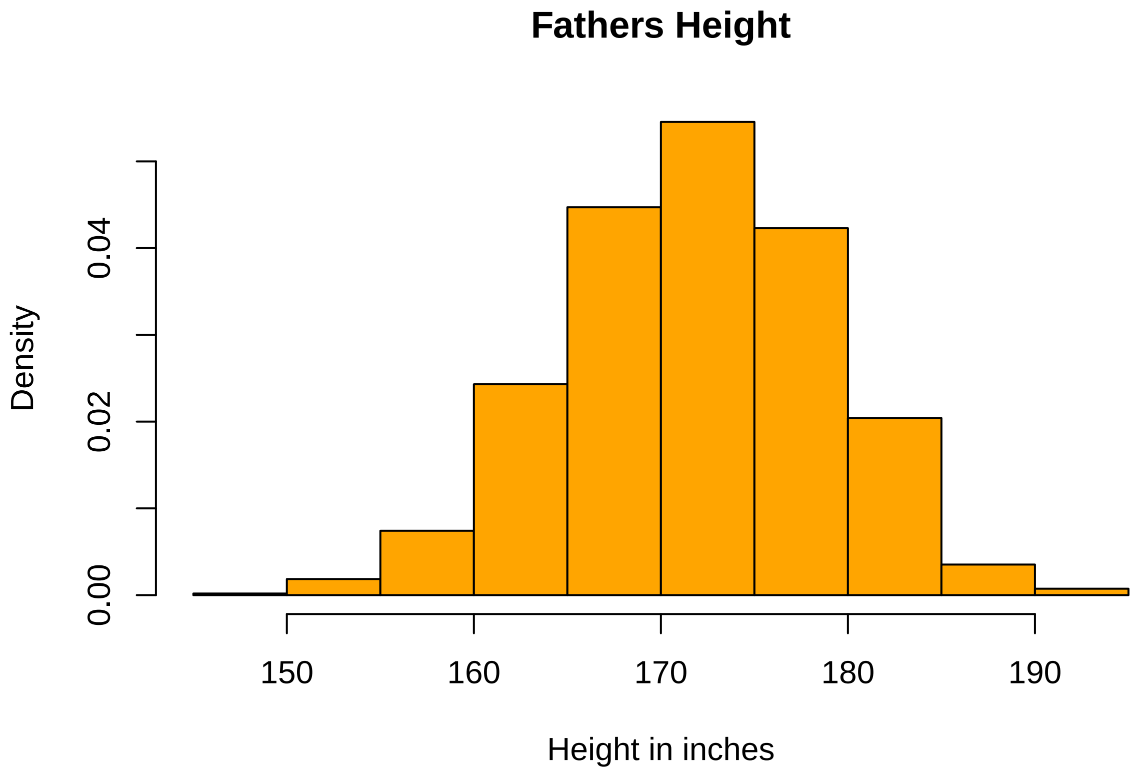
```
## [1] 7.150713
```

#The std deviationf for the fathers height is 6.97 which is almost 7 and that for the sons height is 7.15. As the value of std deviation for both is fairly large it is possible that most of the values of height are away from the mean height in the distribution.

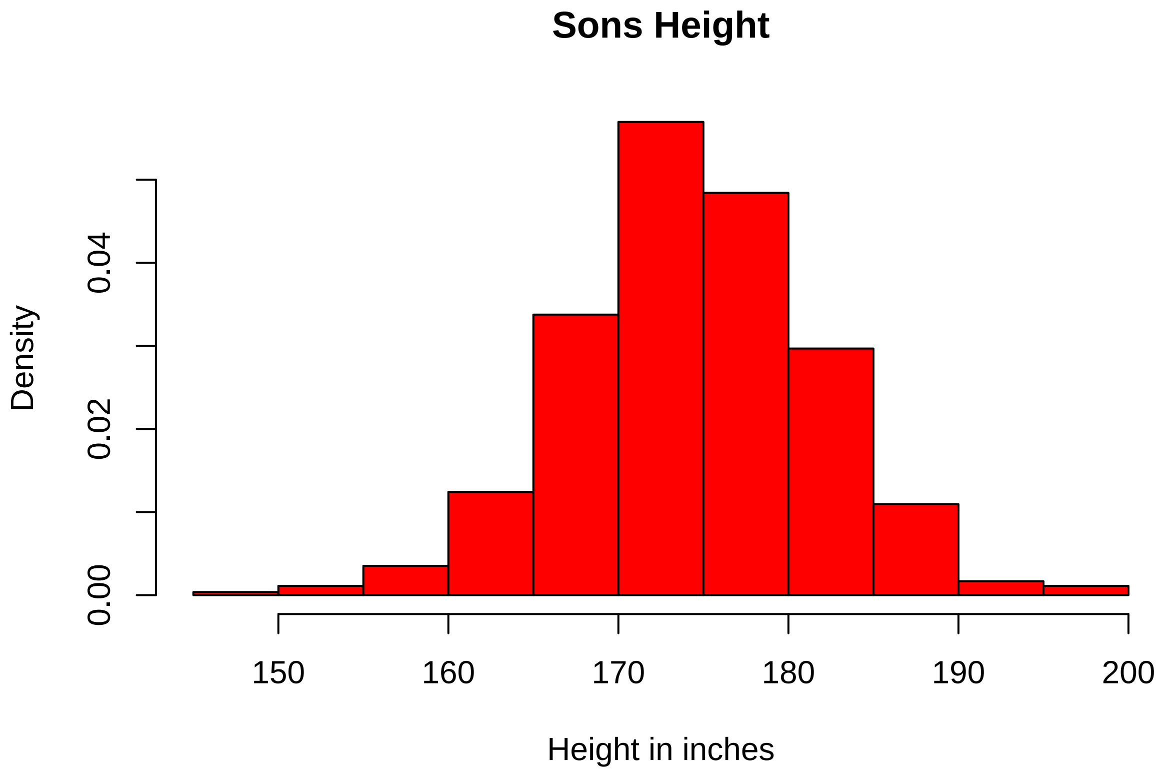
Hint: there is no built-in method to computing sample modes in R. Several packages provide a way to do it, for example try `modeest::mlv` (installed on the server). However, as height is a continuous variable, there are many ways to compute it. Take a look at the corresponding documentation. You may experiment with a few options and pick one, for instance the *naive* method or write your own!

(4) Plot a histogram of the data. Add to this histogram: (1) a plot of normal distribution with the same mean and standard deviation as the data, (2) the sample mean, median, and mode. You can use vertical lines of different colors to do this. What do you find? Are the histogram and the density plot similar?

```r
hist(df2$fheight, main = "Fathers Height", xlab = "Height in inches", prob = T,
    col = "orange")
```



```r
hist(df2$sheight, main = "Sons Height", xlab = "Height in inches", prob = T,
    col = "red")
```

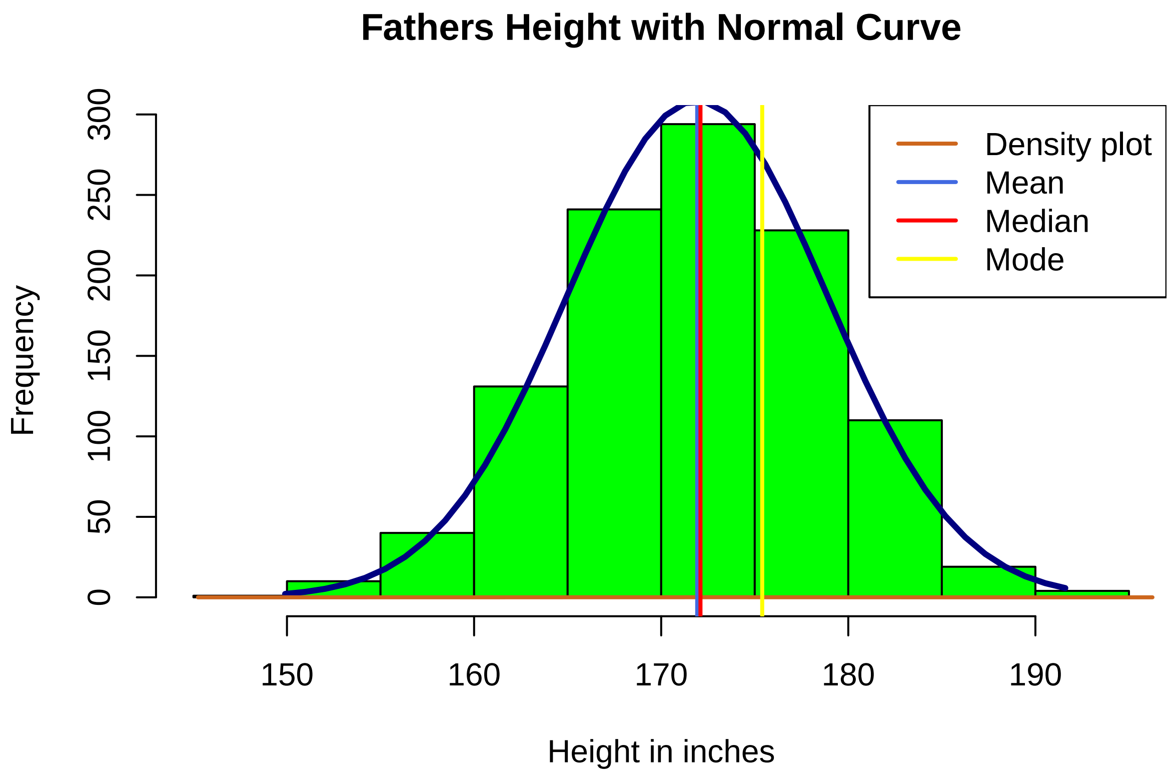


```
library(modeest)
```

```
## Registered S3 method overwritten by 'rmutil':
##   method         from
##   print.response httr
```

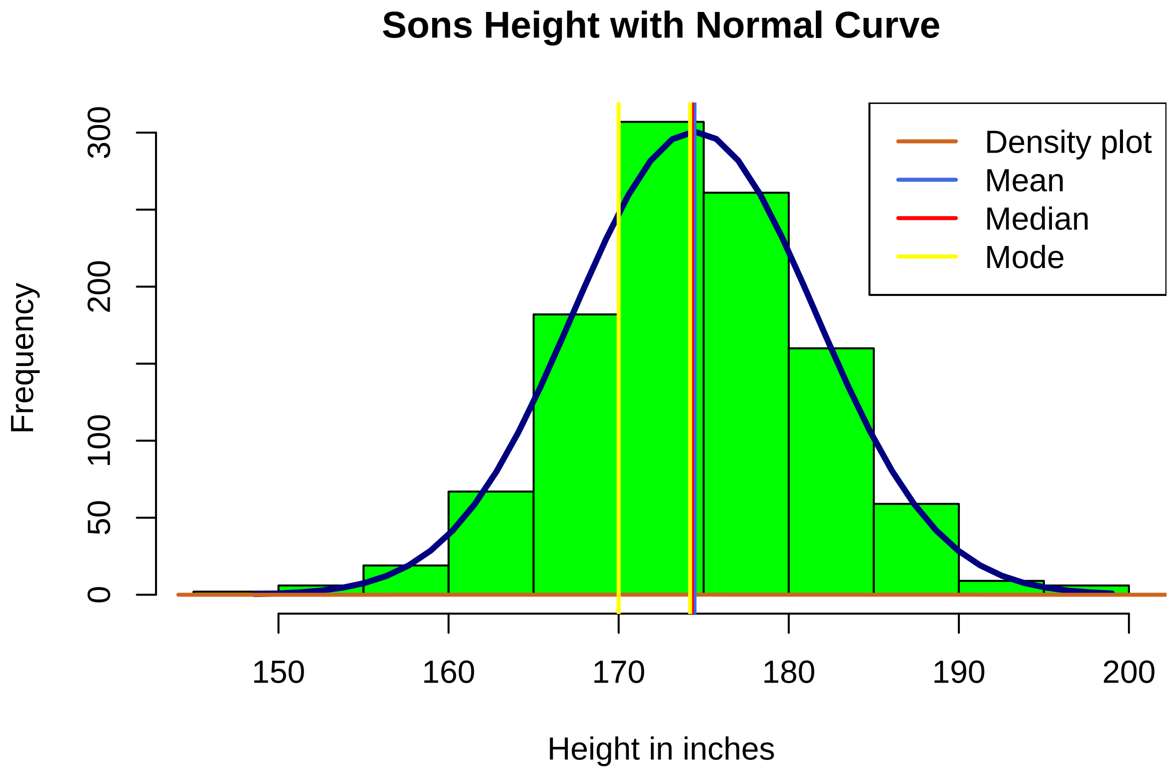
```
x <- df2$fheight
h <- hist(x, breaks = 10, col = "green", xlab = "Height in inches", main = "Fathers Height with Normal C
xfit <- seq(min(x), max(x), length = 40)
yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
yfit <- yfit * diff(h$mids[1:2]) * length(x)
lines(xfit, yfit, col = "navy", lwd = 3)
lines(density(df2$fheight), # density plot
 lwd = 2, # thickness of line
 col = "chocolate3")
abline(v = mean(df2$fheight),
 col = "royalblue",
 lwd = 2)
abline(v = median(df2$fheight),
 col = "red",
 lwd = 2)
abline(v = mfv(df2$fheight),
 col = "yellow",
 lwd = 2)

legend(x = "topright", # location of legend within plot area
 c("Density plot", "Mean", "Median","Mode"),
 col = c("chocolate3", "royalblue", "red","yellow"),
 lwd = c(2, 2, 2))
```



```
z <- df2$sheight
h <- hist(z, breaks = 10, col = "green", xlab = "Height in inches", main = "Sons Height with Normal Cur
xfit <- seq(min(z), max(z), length = 40)
yfit <- dnorm(xfit, mean = mean(z), sd = sd(z))
yfit <- yfit * diff(h$mids[1:2]) * length(x)
lines(xfit, yfit, col = "navy", lwd = 3)
lines(density(df2$sheight), # density plot
 lwd = 2, # thickness of line
 col = "chocolate3")
abline(v = mean(df2$sheight),
 col = "royalblue",
 lwd = 2)
abline(v = median(df2$sheight),
 col = "red",
 lwd = 2)
abline(v = mfv(df2$sheight),
 col = "yellow",
 lwd = 2)

legend(x = "topright", # location of legend within plot area
 c("Density plot", "Mean", "Median","Mode"),
 col = c("chocolate3", "royalblue", "red","yellow"),
 lwd = c(2, 2, 2))
```



#From the histograms and density plots we can see how similar they are and that in both the cases the concentration of values shown by the peaks matches.

(b) Next, let's take a look at the number of citations of research papers.

(1) What kind of measure is this? What kind of valid values would you expect to see (continuous, discrete, positive, . . .)

```
df3 <- read.csv(file = "/home/imt573/Data/mag-in-citations.csv.bz2", sep=",")
#View(df3)
```

#The paper ids would be nominal measure while the citations would be ratios. We would expect to see discrete values.

(2)Read the `mag-in-citations.csv` data. This is Microsoft Academic Graph for citations of research papers, and it contains two columns: paper id and number of citations. We only care about citations here. Provide basic descriptives of this variable: how many observations do we have? Do we have any missing observations?

```
dim(df3)
```

```
## [1] 388258      2
```

```
which(is.na(df3), arr.ind=TRUE)
```

```
##      row col
```

#The paper id column tells us how we can identify a given academic paper and the citations tell us the number of citations used in that research paper. #There are 388258 observations in this data and no missing values have been found in either of the columns

(3) Compute mean, median, mode, standard deviation and range of the heights. Discuss the relationship between these numbers. Is mean larger than median? Than mode? By how much (in relative terms)? How does standard deviation compare to mean?

```
mean(df3$citations)
```

```
## [1] 15.61223
```

```
median(df3$citations)
```

```
## [1] 3
```

```
sd(df3$citations)
```

```
## [1] 78.39079
```

```
m <- table(df3$citations)
mode = names(m)[which(m==max(m))]
mode
```

```
## [1] "0"
```

```
library(modeest)
mfv(df3$citations)
```

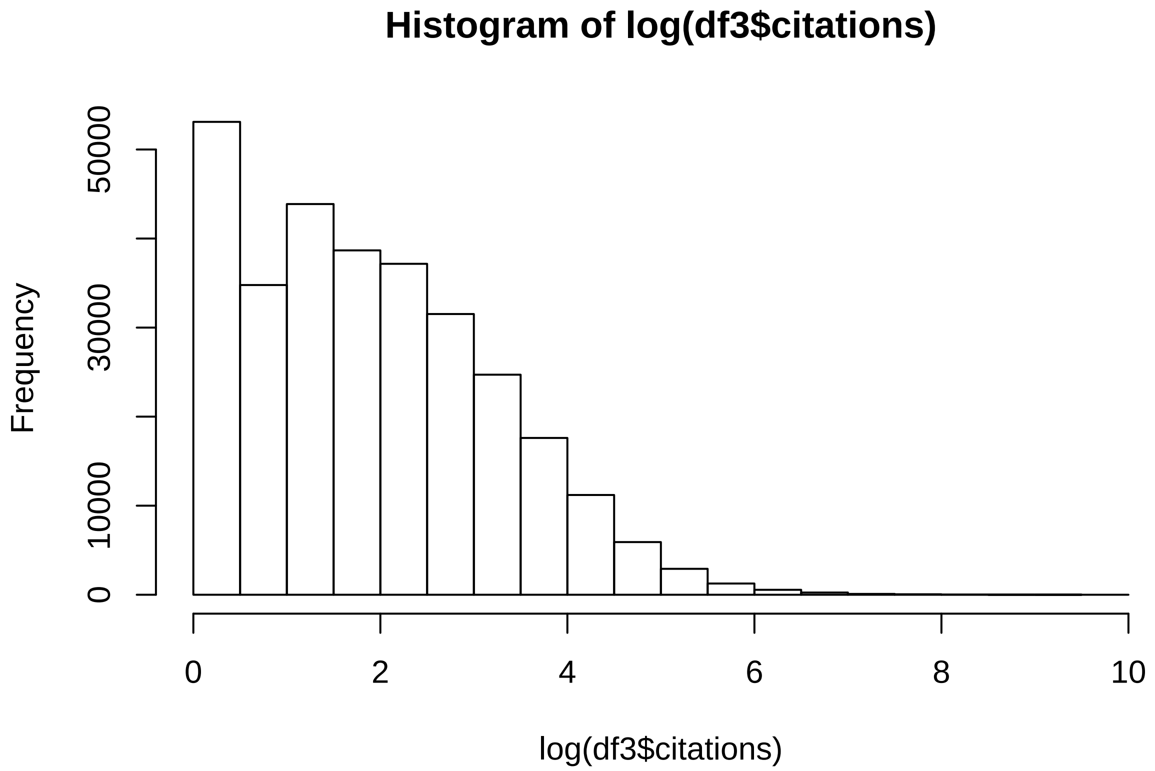
```
## [1] 0
```

#Thus we can see that the mean and median values are extremely far apart. The mean is almost more than 5 times the median value of citations. This can be explained by the extremely high std deviation of 78.4 which means that the data is widely spread and most values are away from the mean. The mode for this distribition is 0.

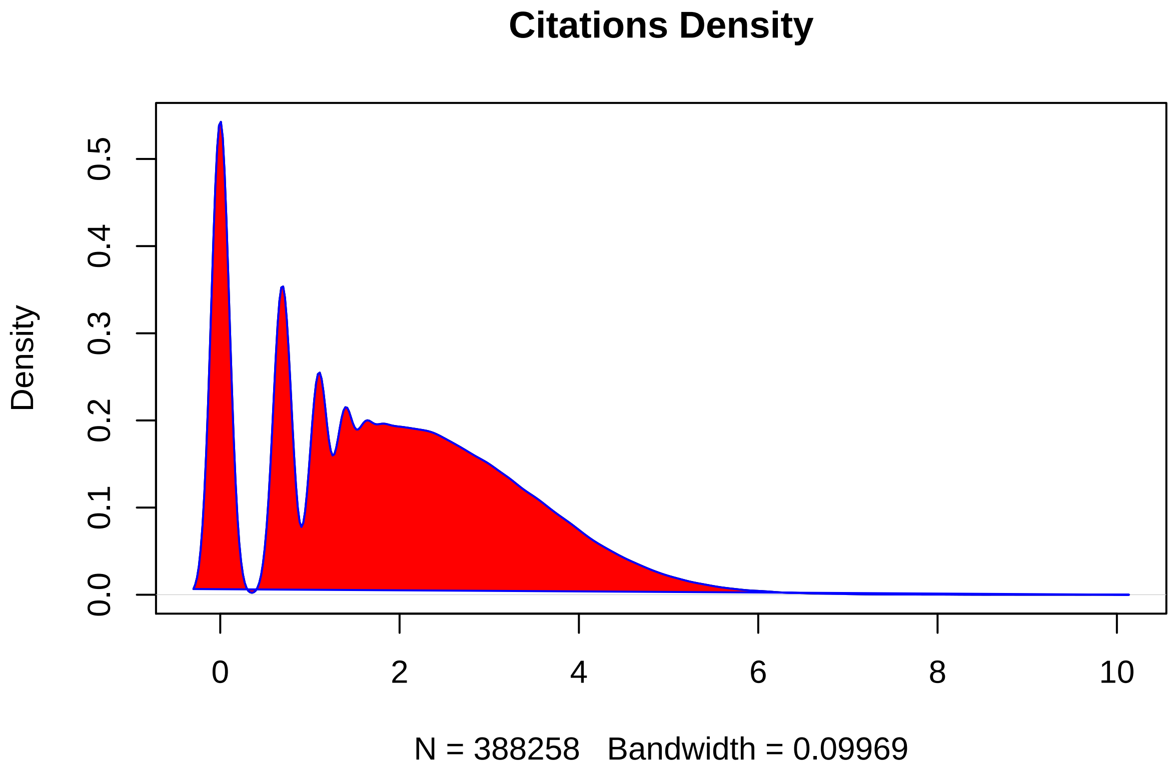
Hint: here you do not want to use any smoothing as we are measuring discrete counts. Use the plain "most frequent value", `method="mfv"` if using the `modeest` package.

(4) Plot a histogram of the data. Add to this histogram: (1) a plot of normal distribution with the same mean and standard deviation as the data, (2) the sample mean, median, and mode. You can use vertical lines of different colors to do this. What do you find? Are the histogram and the density plot similar?

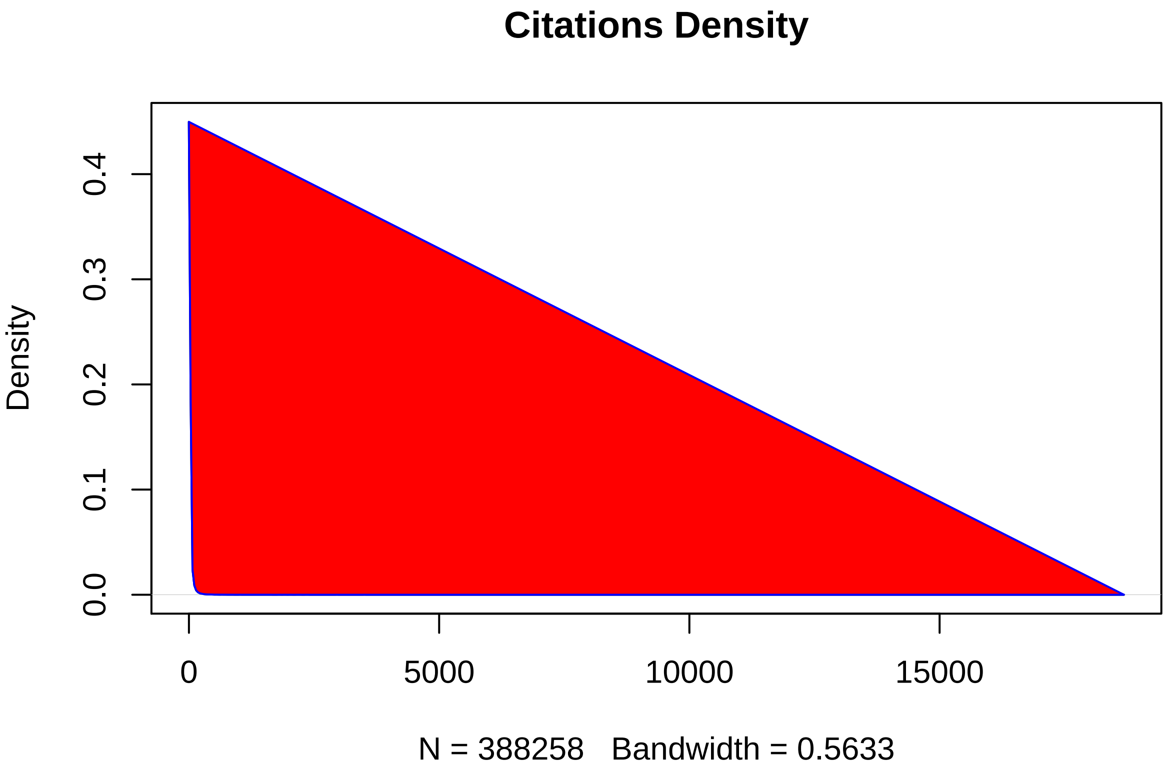
```
hist(log(df3$citations))
```



```
d <- density(log(df3$citations))
plot(d, main="Citations Density")
polygon(d, col="red", border="blue")
```

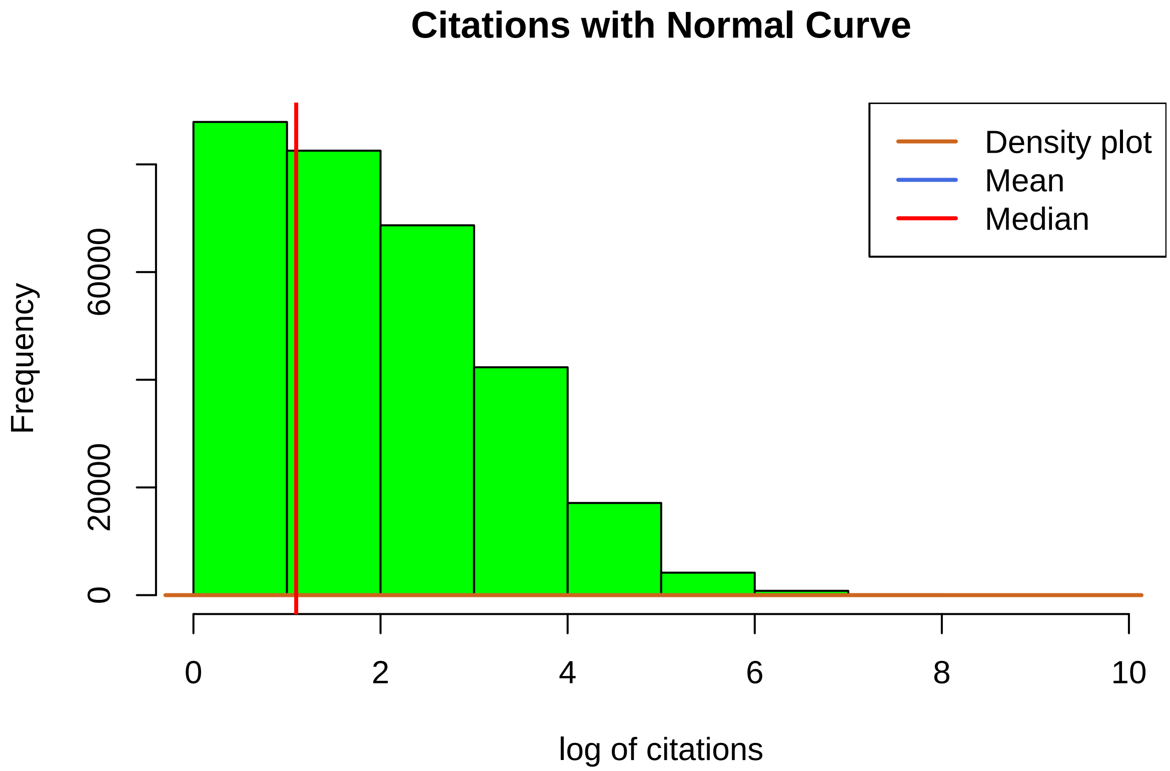


```
d <- density(df3$citations)
plot(d, main="Citations Density")
polygon(d, col="red", border="blue")
```



```
z <- log(df3$citations)
h <- hist(z, breaks = 10, col = "green", xlab = "log of citations", main = "Citations with Normal Curve
xfit <- seq(min(df3$citations), max(df3$citations), length = 40)
yfit <- dnorm(xfit, mean = mean(z), sd = sd(z))
yfit <- yfit * diff(h$mids[1:2]) * length(x)
lines(xfit, yfit, col = "navy", lwd = 3)
lines(density(z), # density plot
 lwd = 2, # thickness of line
 col = "chocolate3")
abline(v = mean(z),
 col = "royalblue",
 lwd = 2)
abline(v = median(z),
 col = "red",
 lwd = 2)

legend(x = "topright", # location of legend within plot area
 c("Density plot", "Mean", "Median"),
 col = c("chocolate3", "royalblue", "red"),
 lwd = c(2, 2, 2))
```

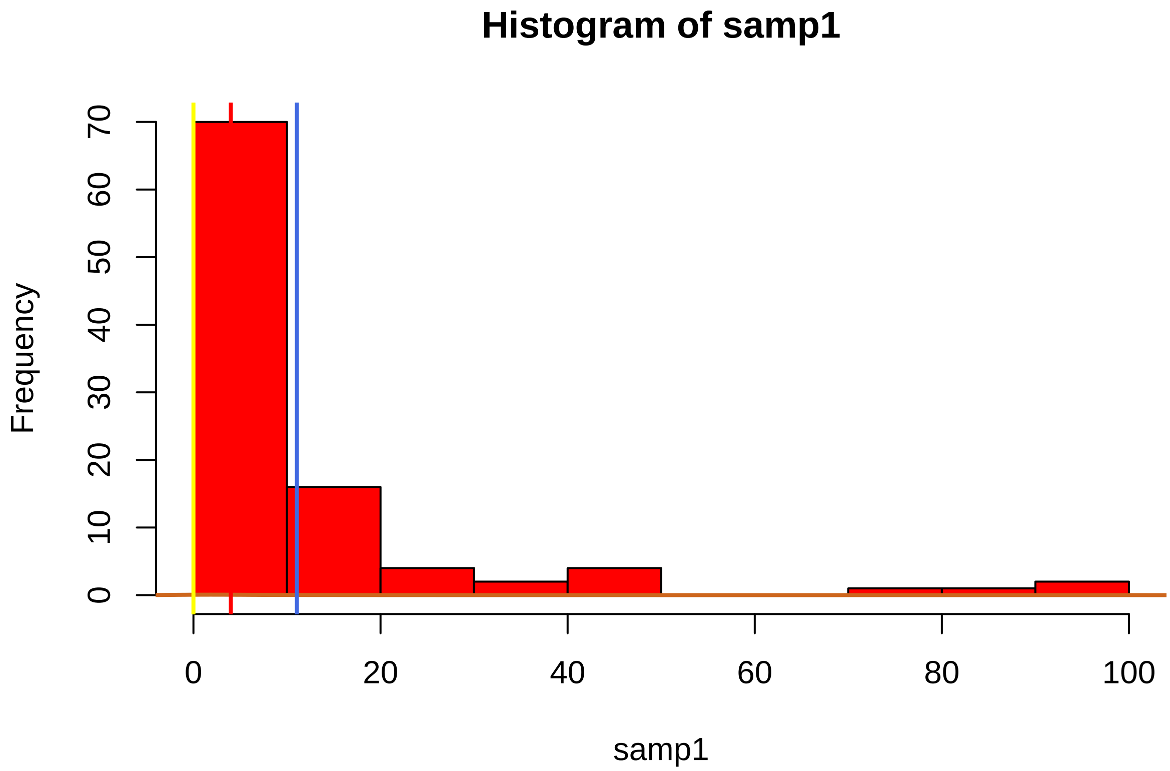


#The histogram and density plots are not very similar as they don't fit. The data is not distributed normally in this case and seems to have a wider std deviation of values aka more spread.

```
samp1 <- sample(df3$citations, 100)
summary(samp1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    1.00    4.00   11.06   13.00   98.00
```

```
hist(samp1, col = "red")
lines(density(samp1), # density plot
 lwd = 2, # thickness of line
 col = "chocolate3")
abline(v = mean(samp1),
 col = "royalblue",
 lwd = 2)
abline(v = median(samp1),
 col = "red",
 lwd = 2)
abline(v = mfv(samp1),
 col = "yellow",
 lwd = 2)
```



Hint: You might try experimenting with log-log scale for the histogram.

(c) Comment on your finding from part (a) and part (b). Be sure to compare the two cases.

#From part a we can see a continuous value distribution while in part b we see a discrete value distribution.We also see that in part a mean, median and mode values are closer and hence the density curve and histogram fit and the distribution is closer to a normal distribution. In part b despite using log and sampling the data we can see that normal distribution curve fails to spread and as there are alot of values of one kind, the distribution is skewed towards that frequency and is thus not at all normally distributed. It has a wide std deviation and is cannot fit a curve through it.

Extra Credit: Guessing on Multiple Choice Tests

In the exam, there is a multiple-choice question with four (mutually exclusive) options. In average, 80% of the students know the answer, but event those who know, still answer it wrong in 10% of time because of the exam stress.

1. If a student get's the answer right, what is the probability that she actually knows the material? Hint: read OIS 2.2, in particular 2.2.7 (2017 version).