20INMCAL204- Laboratory Report

Sanjay Krishnan J[1,1,1,*], Siju K.S[b,2], Jobin Jose[b,2]

[a] *Department of Computer Applications RB402*
[b] *Department of Mathematics AB304*

Abstract

Experiments listed in the Lab Manual are successfully executed in the R version 4.1.0. Details of the experiments with input & ouput are summerized in the form of a report. Experiments are arranged in the form of sections. This report is prepared using the R-package `rticles` (**?**).

Contents

---

*Corresponding author
Email addresses: `sanjay.inmca@saintgits.org` (Sanjay Krishnan J), `siju.swamy@saintgits.org` (Siju K.S), `jobin.jose@saintgits.org` (Jobin Jose)
[1]Student, 20INMCAL204.
[2]Course Faculty, 20INMCAL204.

1. Experiment 4: Statistical Summary and measure of normality of a dataset

1.1. Aim

1. To create the statistical summary of a data
2. To study normality of the data

1.2. Packages used and syntax of R methods

For statistical summary of a given dataset, the `rbase` package will be used. To calculate skewness and kurtosis of dataset, the `ACSWR` is used.

> *Note:* The functions `skewness` and `kurtosis` from the `e1071` package are more generic functions. Another resouse is `moments` package.

1.3. Algorithm

- Step 1: Load the dataset
- Step 2: Load necessary packages
- Step 3: Calculate statistical summaries
- Step 4: Calculate the `skewness` and `kurtosis` of the numerical data
- Step 5: Report the results

1.4. R code

```
#loading package
library(ACSWR)
#loading data
data(yb)
#view structure of data
str(yb)
```

```
## 'data.frame':    8 obs. of  2 variables:
##  $ Preparation_1: int  31 20 18 17 9 8 10 7
##  $ Preparation_2: int  18 17 14 11 10 7 5 6
```

```
# creating statistical summary

summary(yb)
```

```
##  Preparation_1   Preparation_2
##  Min.   : 7.00   Min.   : 5.00
##  1st Qu.: 8.75   1st Qu.: 6.75
##  Median :13.50   Median :10.50
##  Mean   :15.00   Mean   :11.00
##  3rd Qu.:18.50   3rd Qu.:14.75
##  Max.   :31.00   Max.   :18.00
```

```
range(yb$Preparation_1); range(yb$Preparation_2) # list out ranges of data
```

```
## [1]  7 31
```

```
## [1]  5 18
```

```r
#skewness and kurtosis of preparation_1
skewcoeff(yb$Preparation_1); kurtcoeff(yb$Preparation_1)
```

```
## [1] 0.8548652
```

```
## [1] 2.727591
```

```r
#skewness and kurtosis of preparation_2
skewcoeff(yb$Preparation_2); kurtcoeff(yb$Preparation_2)
```

```
## [1] 0.2256965
```

```
## [1] 1.6106
```

1.5. Results & discussions

A distribution is normal then `mean=median=mode` and the skewness is 0 and kurtosis is 2. In this experiment statistical summaries of two variables are created. From the skewness and kurtosis measures, both the variables are positively skewed and `preparation_1` is lepto-kurtic and `preparation_2` is meso-kurtic. Based on the statistical summary and skewness and kurtosis measures, both the variables are different from a normal distribution.

2. Experiment 5- Implementation of Bayes Theorem

2.1. Aim

1. To calculate Bayes posterior probability using Bayes theorem

2.2. Packages used and syntax of R methods

Bayes posterior probability can be directly calculated using mathematical method or using the package `LaplacesDemon`.

2.3. Algorithm

- Step 1: Load the package, prior probabilities and conditionals
- Step 2: Calculate the Bayes posterior probability using the formula-$P(B_j|A) = \dfrac{P(A|B_j)P(B_j)}{\sum_{j=1}^{m} P(A|B_j)P(B_j)}$
- Step 3: Calculate the same prior probability using `LaplaceDemon` package
- Step 4: Report the results

  Case: Classical Problem from Hoel, Port, and Stone (1971). Suppose there are three tables with two drawers each. The first table has a gold coin in each of the drawers, the second table has a gold coin in one drawer and a silver coin in the other drawer, while the third table has silver coins in both of the drawers. A table is selected at random and a drawer is opened which shows a gold coin.

  *Observation:*The problem is to compute the probability of the other drawer also showing a gold coin. The Bayes formula can be easily implemented in an R program.

2.4. R code

```r
#loading data
prob_GC <- c(1,1/2,0)
priorprob_GC <- c(1/3,1/3,1/3)
```

```r
#calculating postrior probability
post_GC <- prob_GC*priorprob_GC
post_GC/sum(post_GC)
```

```
## [1] 0.6666667 0.3333333 0.0000000
```

```r
# do the same using LaplacesDemon` package
library(LaplacesDemon)
BayesTheorem(prob_GC, priorprob_GC)
```

```
## [1] 0.6666667 0.3333333 0.0000000
## attr(,"class")
## [1] "bayestheorem"
```

2.5. Results & discussions

The Bayes theorem is used to calculate posterior probability of the Mathematical model of the given case. Also the result is verified using the `LaplacesDemon` package.

References

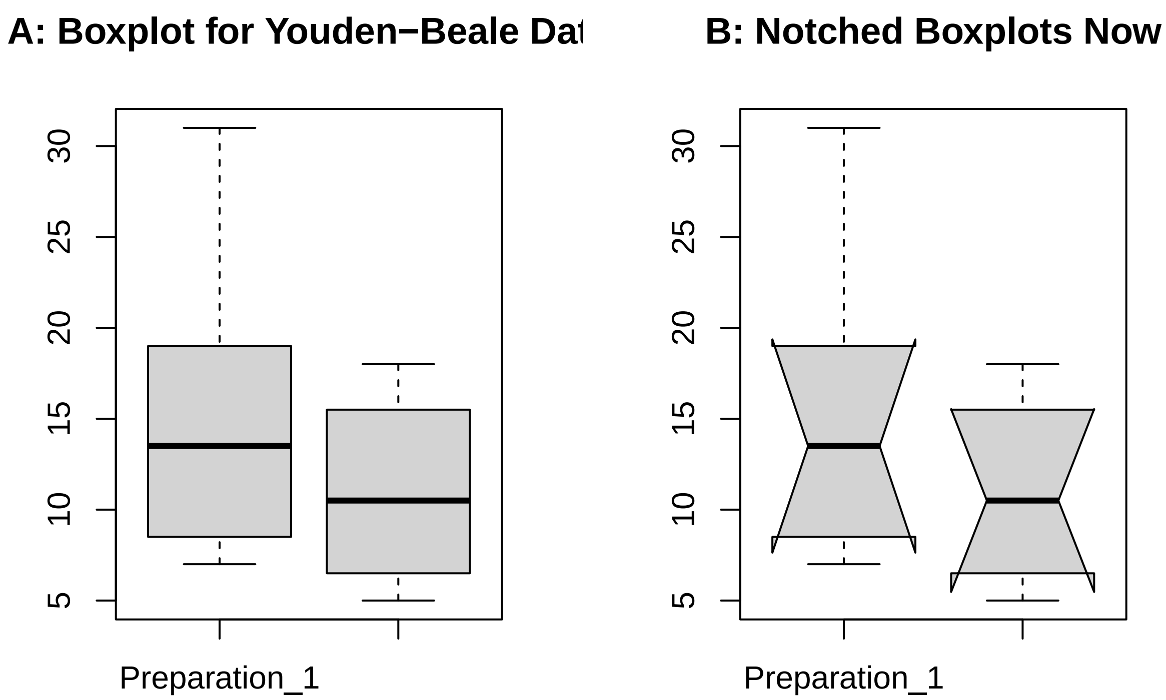
3. Experiment -2 implement various EDA techniques using R.

##Aim implement various EDA techniques using R.
##Algorithm
step1: Create R-code chunks for coding
step 2: Create a Boxplot using built -in dataset
step3: Create a Histogram using R
step 4:create a Scatter plot using R
step 5: Create a Running chart/time series plot using R
##R-code

3.0.1. Boxplot

The Youden-Beale Experiment. We have used this dataset in Chapter 2, Section 4, and in a few other places too. We need to compare here if the two virus extracts have a varying effect on the tobacco leaf or not. We have already read this dataset into R on more than one occasion. First, the boxplot is generated without the notches for yb data.frame using the boxplot function. The median for `Preparation_1` certainly appears higher than for `Preparation_2`, see Part A of Figure 3.1. Thus, we are tempted to check whether the medians for the two preparations are significantly different with the notched boxplot. Now, the boxplot is generated to produce the notches with the option notch=TRUE. Appropriate headers for a figure are specified with the title function. Most importantly, we have used a powerful graphical technique of R through par, which is useful in setting graphical parameters. Here, mfrow indicates that we need a multi-row figure with one row and two columns(Tattar 2015).

```
library(ACSWR)
data(yb)
par(mfrow=c(1,2))
boxplot(yb)
title("A: Boxplot for Youden-Beale Data")
boxplot(yb,notch=TRUE)
title("B: Notched Boxplots Now")
```




```
summary(yb$Preparation_1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00    8.75   13.50   15.00   18.50   31.00
```

```
summary(yb$Preparation_2)
```

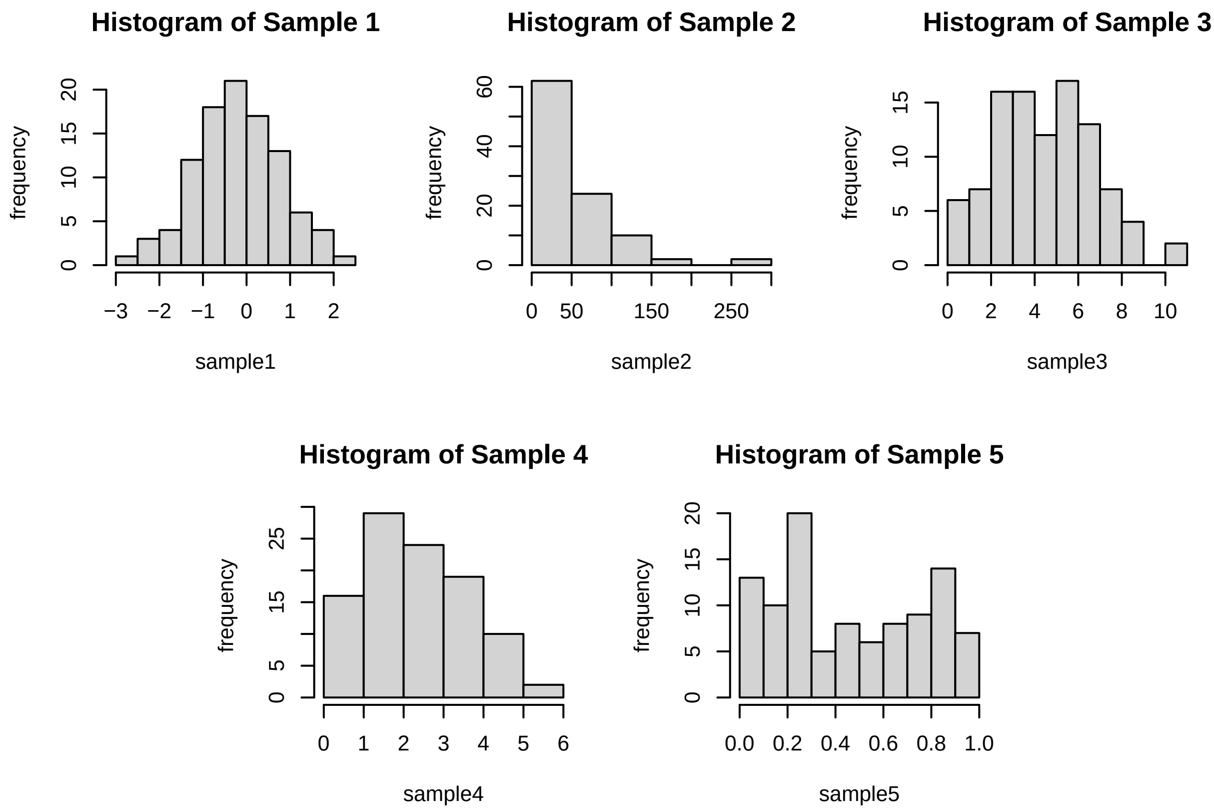
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00    6.75   10.50   11.00   14.75   18.00
```

3.0.2. Histogram:

The histogram was invented by the eminent statistician Karl Pearson and is one of the earliest types of graphical display. It goes without saying that its origin is earlier than EDA, at least the EDA envisioned by Tukey, and yet it is considered by many EDA experts to be a very useful graphical technique, and makes it to the list of one of the very useful practices of EDA. The basic idea is to plot a bar over an interval proportional to the frequency of the observations that lie in that interval. If the sample size is moderately good in some sense and the sample is a true representation of a population, the histogram reveals the shape of the true underlying uncertainty curve. Though histograms are plotted as two-dimensional, they are essentially one-dimensional plots in the sense that the shape of the uncertainty curve is revealed without even looking at the range of the x-axis. Furthermore, the Pareto chart, stem-and-leaf plot, and a few others may be shown as special cases of the histogram.We begin with a “cooked” dataset for understanding a range of uncertainty curves.

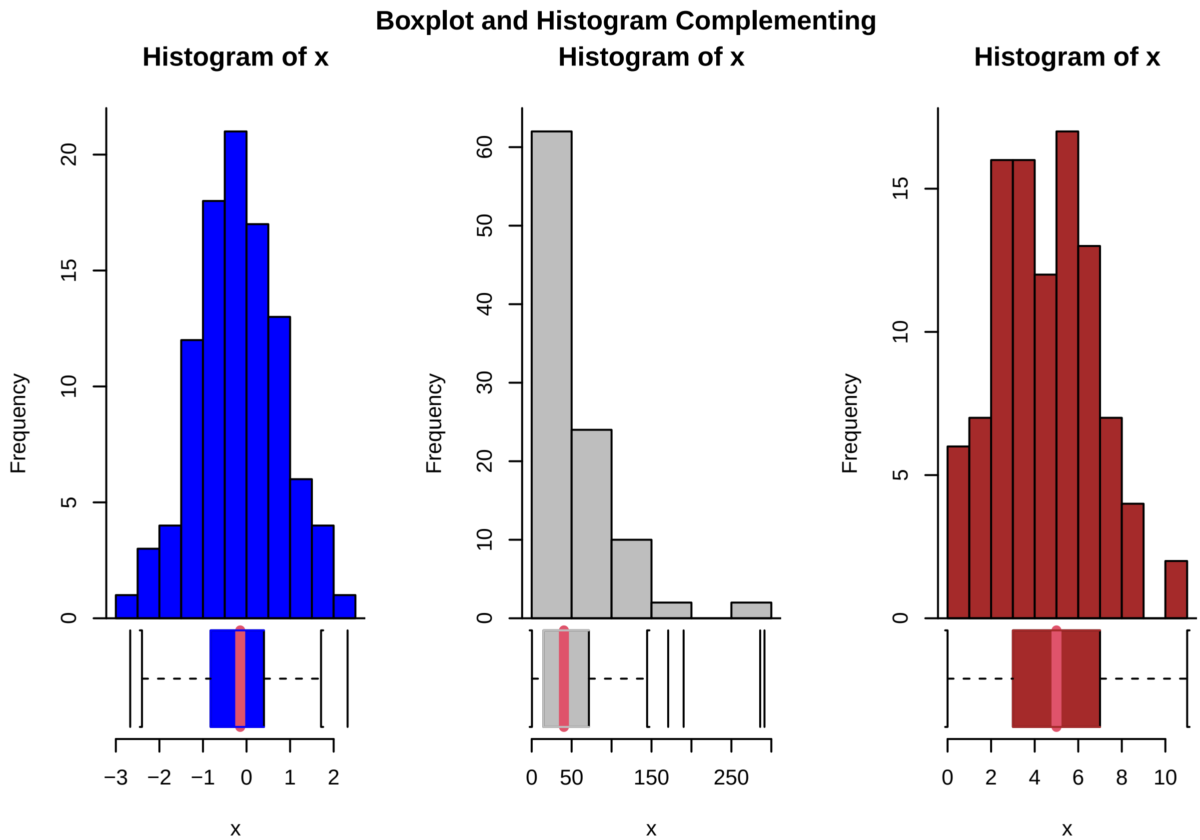
Understanding Histogram of Various Uncertainty Curves. In the dataset sample, we have data from five different probability distributions. Towards understanding the plausible distribution of the samples, we plot the histogram and see how useful it is.

```
data(sample)
layout(matrix(c(1,1,2,2,3,3,0,4,4,5,5,0), nrow=2, ncol=6, byrow=TRUE),respect=FALSE)
#matrix(c(1,1,2,2,3,3,0,4,4,5,5,0), nrow=2, ncol=6, byrow=TRUE)
hist(sample[,1],main="Histogram of Sample 1",xlab="sample1",  ylab="frequency")
hist(sample[,2],main="Histogram of Sample 2",xlab="sample2", ylab="frequency")
hist(sample[,3],main="Histogram of Sample 3",xlab="sample3", ylab="frequency")
hist(sample[,4],main="Histogram of Sample 4",xlab="sample4", ylab="frequency")
hist(sample[,5],main="Histogram of Sample 5",xlab="sample5", ylab="frequency")
```



Histogram extensions: Understanding Histogram of Various Uncertainty Curves. The short program for this problem is given below.

```
library(sfsmisc)
par(mfrow=c(1,3))
histBxp(sample$Sample_1,col="blue",boxcol="blue",xlab="x")
histBxp(sample$Sample_2,col="grey",boxcol="grey",xlab="x")
histBxp(sample$Sample_3,col="brown",boxcol="brown",xlab="x")
title("Boxplot and Histogram Complementing",outer=TRUE,line=-1)
```



3.0.3. Parto chart theory :

Pareto Chart The Pareto chart has been designed to address the implicit questions answered by the Pareto law. The common understanding of the Pareto law is that "majority resources" is consumed by a "minority user." The most common of the percentages is the 80–20 rule, implying that 80% of the effects come from 20% of the causes. The Pareto law is also known as the law of vital few, or the 80–20 rule. The Pareto chart gives very smart answers by completely answering how much is owned by how many. Montgomery (2005), page 148, has listed the Pareto chart as one of the seven major tools of Statistical Process Control.

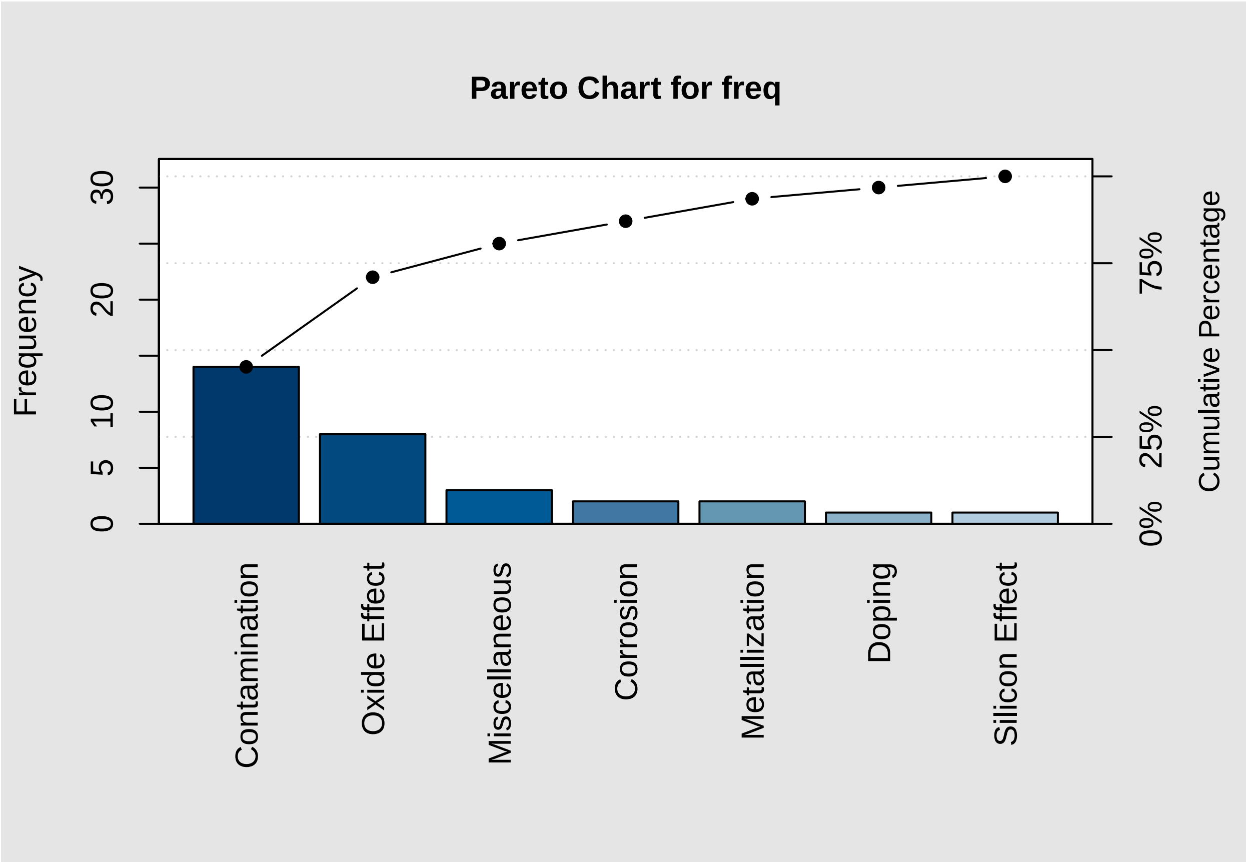
A Pareto chart is a bar graph. The lengths of the bars represent frequency or cost (time or money), and are arranged with longest bars on the left and the shortest to the right. In this way the chart visually depicts which situations are more significant. This cause analysis tool is considered one of the seven basic quality tools

```
library(qcc)
```

```
## Package 'qcc' version 2.7
```

```
## Type 'citation("qcc")' for citing this R package in publications.
```

```r
freq <- c(14,2,1,2,3,8,1)
names(freq) <- c("Contamination","Corrosion","Doping", "Metallization", "Miscellaneous", "Oxide Effect
pareto.chart(freq)
```




```
##
## Pareto chart analysis for freq
##                   Frequency  Cum.Freq. Percentage Cum.Percent.
##   Contamination   14.000000  14.000000  45.161290    45.161290
##   Oxide Effect     8.000000  22.000000  25.806452    70.967742
##   Miscellaneous    3.000000  25.000000   9.677419    80.645161
##   Corrosion        2.000000  27.000000   6.451613    87.096774
##   Metallization    2.000000  29.000000   6.451613    93.548387
##   Doping           1.000000  30.000000   3.225806    96.774194
##   Silicon Effect   1.000000  31.000000   3.225806   100.000000
```

##Result & Discussions various visualization techniques for data analysis are implemented in R. #Experiment 3 ## Aim
To administer baseline statistical analysis on a dataset and report descriptive analysis summary.

3.1. Algorithm

- Step-1: Load the data and required R-packages for data analysis

- Step-2: Apply basic statistic functions
- Step-3: Create appropriate visualizations
- Step-4: Report the findings based on descriptive analysis

3.2. R-code

Loading data

```
df<-read.csv("https://raw.githubusercontent.com/sijuswamy/StatLab/main/Dataset_1.csv",header = TRUE)
df$Gender=as.factor(df$Gender)

head(df)
```

```
##          Student_Name Gender X20IMCAT201 X20IMCAT203 X20IMCAT205 X20IMCAT207
## 1 ABEL MATHEW ABRAHAM      M         1.5         1.5          18           1
## 2         ABEN B JOHN      M         8.5         6.0          26           5
## 3           ABIN SAJI      M         0.0         0.0          21           0
## 4       ADWAITH SANIL      M         8.0        13.0          18           4
## 5         AKSHAY BABU      M         8.0         4.0          14          10
## 6         ALEN T BINU      M        29.5        24.0          43          19
##   X20IMCAT209
## 1          16
## 2          23
## 3           0
## 4          17
## 5           5
## 6          35
```

```
str(df)
```

```
## 'data.frame':    56 obs. of  7 variables:
##  $ Student_Name: chr  "ABEL MATHEW ABRAHAM" "ABEN B JOHN" "ABIN SAJI" "ADWAITH SANIL" ...
##  $ Gender      : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 1 1 2 ...
##  $ X20IMCAT201 : num  1.5 8.5 0 8 8 29.5 16.5 21 25.5 30 ...
##  $ X20IMCAT203 : num  1.5 6 0 13 4 24 8.5 15 11.5 32.5 ...
##  $ X20IMCAT205 : num  18 26 21 18 14 43 25 22 25 32 ...
##  $ X20IMCAT207 : num  1 5 0 4 10 19 10 11 1 12 ...
##  $ X20IMCAT209 : num  16 23 0 17 5 35 12 17 30 37 ...
```

Finding Column sums

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:sfsmisc':
##
##     last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

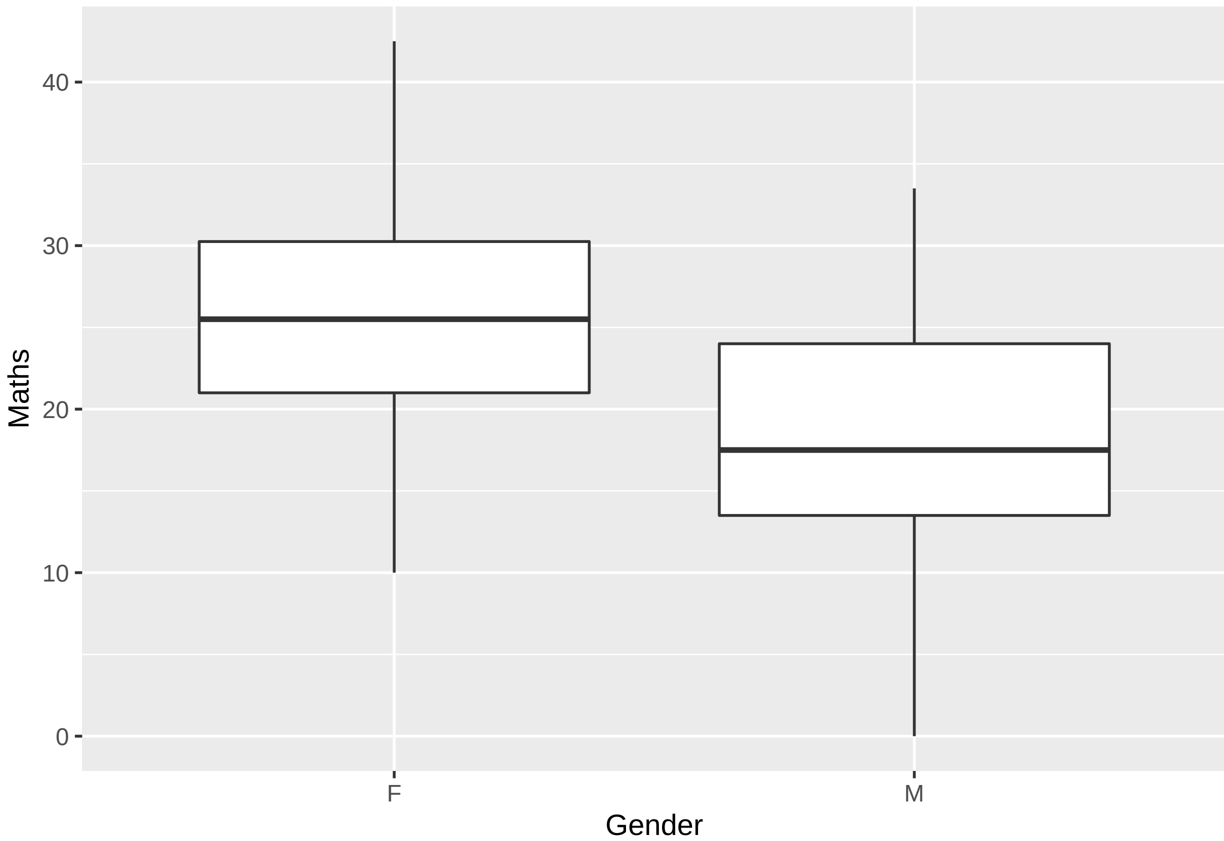
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df1=select(df,-Student_Name,-Gender)
Sub_total=colSums(df1)
Sub_average=colMeans(df1)
```

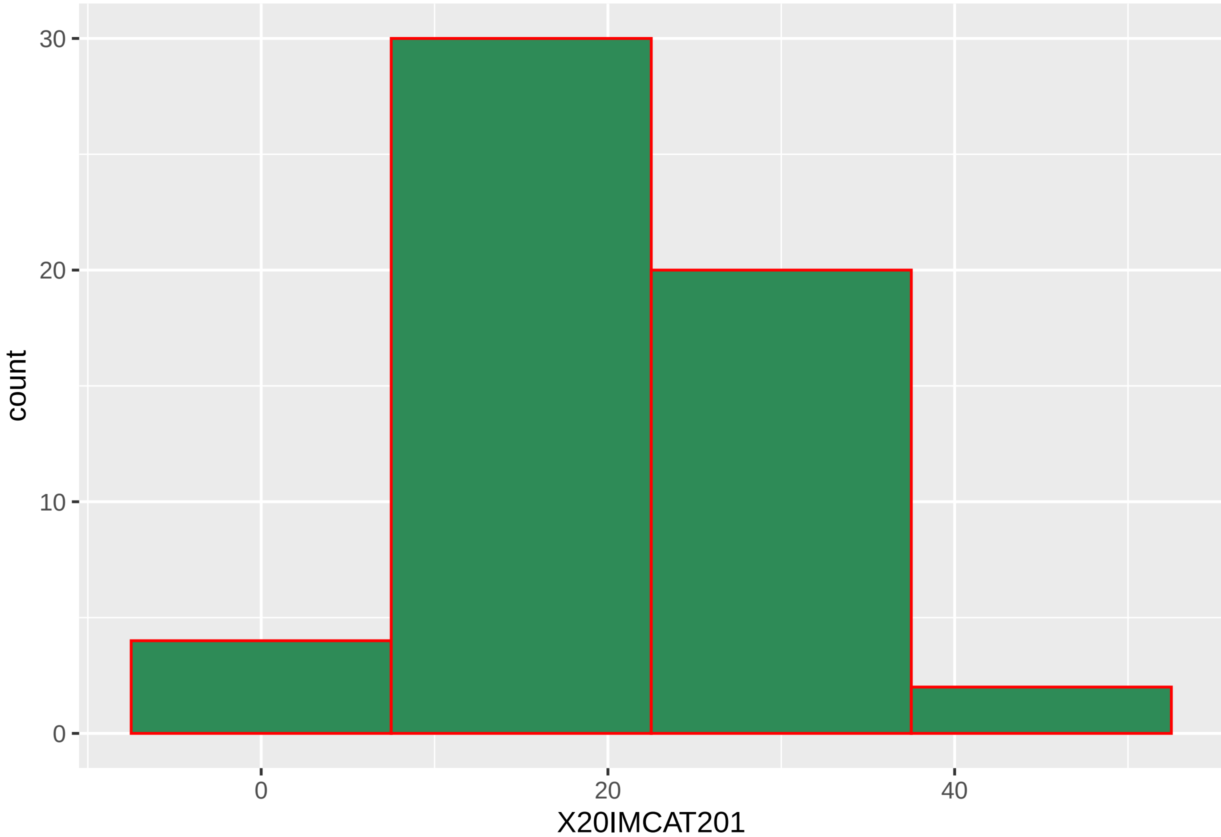
```
round(Sub_average,2)
```

```
## X20IMCAT201 X20IMCAT203 X20IMCAT205 X20IMCAT207 X20IMCAT209
##       20.54       12.17       26.68       11.29       24.68
```

```
library(ggplot2)
crop=ggplot(data=df, mapping=aes(x=Gender, y=X20IMCAT201))+geom_boxplot()+labs(x ="Gender", y = "Maths
crop
```

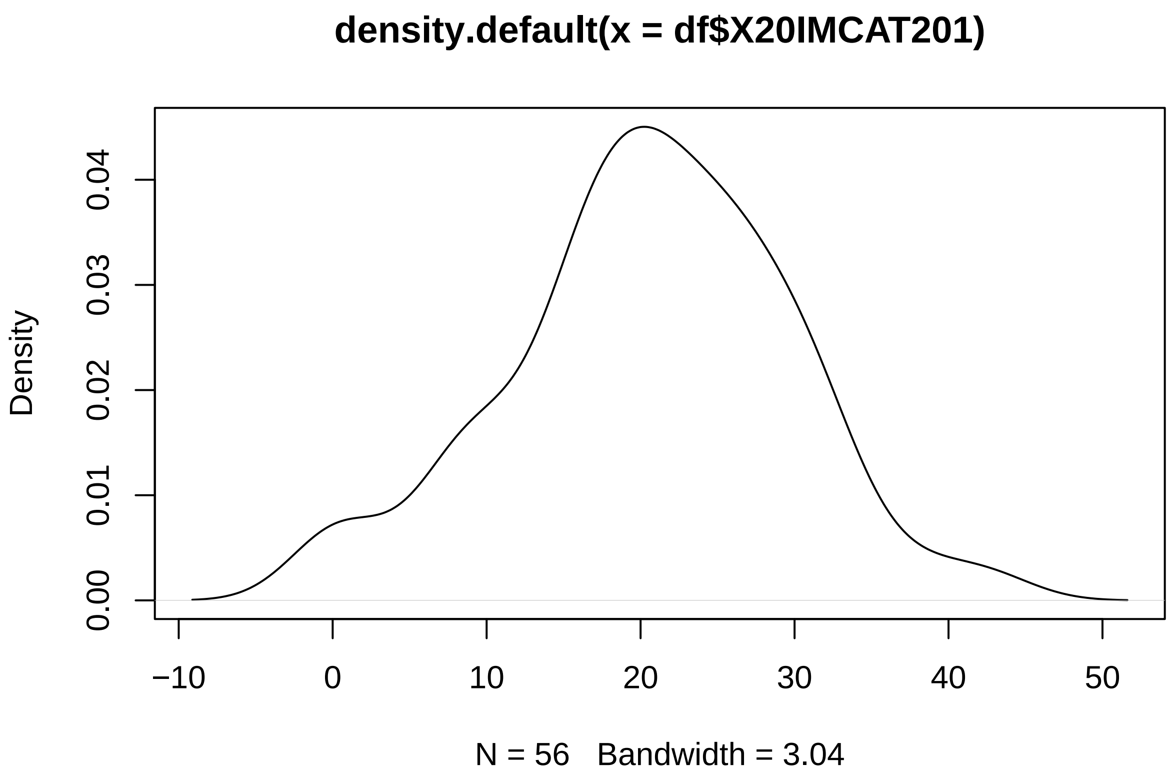


```r
ggplot(data = df, aes(x = X20IMCAT201,fill = df$Gender)) + geom_histogram(binwidth = 15, fill = "seagr
  theme(legend.position = "top")#+facet_grid(~Gender)
```



222

```r
plot(density(df$X20IMCAT201))
```




```r
median(df$X20IMCAT201)
```

```
## [1] 21
```

```r
library(DescTools)
```

```
##
## Attaching package: 'DescTools'
```

```
## The following object is masked from 'package:LaplacesDemon':
##
##     Mode
```

```r
Mode(df$X20IMCAT201)
```

```
## [1] 21
## attr(,"freq")
## [1] 4
```

User defined funtion

```
calcmode <- function(a) {
vector <- unique(a)
vector[which.max(tabulate(match(a, vector)))]
}
```

```
calcmode(df$X20IMCAT201)
```

```
## [1] 21
```

```
sd(df$X20IMCAT201)
```

```
## [1] 9.053014
```

3.3. Results and discussions

#experiment 4

3.4. Aim

1. To create the statistical summary of a data
2. To study normality of the data

3.5. Packages used and syntax of R methods

For statistical summary of a given dataset, the `rbase` package will be used. To calculate skewness and kurtosis of dataset, the `ACSWR` is used.

> *Note:* The functions `skewness` and `kurtosis` from the `e1071` package are more generic functions. Another resouse is `moments` package.

3.6. Algorithm

- Step 1: Load the dataset
- Step 2: Load necessary packages
- Step 3: Calculate statistical summaries
- Step 4: Calculate the `skewness` and `kurtosis` of the numerical data
- Step 5: Report the results

3.7. R code

```
#loading package
library(ACSWR)
#loading data
data(yb)
#view structure of data
str(yb)
```

```
## 'data.frame':    8 obs. of  2 variables:
##  $ Preparation_1: int  31 20 18 17 9 8 10 7
##  $ Preparation_2: int  18 17 14 11 10 7 5 6
```

```r
# creating statistical summary

summary(yb)
```

```
##  Preparation_1    Preparation_2
##  Min.   : 7.00   Min.   : 5.00
##  1st Qu.: 8.75   1st Qu.: 6.75
##  Median :13.50   Median :10.50
##  Mean   :15.00   Mean   :11.00
##  3rd Qu.:18.50   3rd Qu.:14.75
##  Max.   :31.00   Max.   :18.00
```

```r
range(yb$Preparation_1); range(yb$Preparation_2) # list out ranges of data
```

```
## [1]  7 31
```

```
## [1]  5 18
```

```r
#skewness and kurtosis of preparation_1
skewcoeff(yb$Preparation_1); kurtcoeff(yb$Preparation_1)
```

```
## [1] 0.8548652
```

```
## [1] 2.727591
```

```r
#skewness and kurtosis of preparation_2
skewcoeff(yb$Preparation_2); kurtcoeff(yb$Preparation_2)
```

```
## [1] 0.2256965
```

```
## [1] 1.6106
```

3.8. Results & discussions

A distribution is normal then `mean=median=mode` and the skewness is 0 and kurtosis is 2. In this experiment statistical summaries of two variables are created. From the skewness and kurtosis measures, both the variables are positively skewed and `preparation_1` is lepto-kurtic and `preparation_2` is meso-kurtic. Based on the statistical summary and skewness and kurtosis measures, both the variables are different from a normal distribution. #Experiment 5 ## Aim

1. To calculate Bayes posterior probability using Bayes theorem

3.9. Packages used and syntax of R methods

Bayes posterior probability can be directly calculated using mathematical method or using the package `LaplacesDemon`.

3.10. Algorithm

- Step 1: Load the package, prior probabilities and conditionals
- Step 2: Calculate the Bayes posterior probability using the formula-$P(B_j|A) = \dfrac{P(A|B_j)P(B_j)}{\sum\limits_{j=1}^{m} P(A|B_j)P(B_j)}$
- Step 3: Calculate the same prior probability using `LaplaceDemon` package
- Step 4: Report the results

  Case: Classical Problem from Hoel, Port, and Stone (1971). Suppose there are three tables with two drawers each. The first table has a gold coin in each of the drawers, the second table has a gold coin in one drawer and a silver coin in the other drawer, while the third table has silver coins in both of the drawers. A table is selected at random and a drawer is opened which shows a gold coin.

  *Observation:*The problem is to compute the probability of the other drawer also showing a gold coin. The Bayes formula can be easily implemented in an R program.

3.11. R code

```
#loading data
prob_GC <- c(1,1/2,0)
priorprob_GC <- c(1/3,1/3,1/3)
```

```
#calculating postrior probability
post_GC <- prob_GC*priorprob_GC
post_GC/sum(post_GC)
```

```
## [1] 0.6666667 0.3333333 0.0000000
```

```
# do the same using LaplacesDemon` package
library(LaplacesDemon)
BayesTheorem(prob_GC, priorprob_GC)
```

```
## [1] 0.6666667 0.3333333 0.0000000
## attr(,"class")
## [1] "bayestheorem"
```

3.12. Results & discussions

The Bayes theorem is used to calculate posterior probability of the Mathematical model of the given case. Also the reslut is verified using the `LaplacesDemon` package. #Experiment 6 ## Aim

1. To calculate probability mass density, probability distribution and quantiles using binomial distribution

3.13. Packages used and syntax of R methods

Functions from `stat` package (which is loaded by default).
The probability mass at a point x can be evaluated using the syntax:

`dbinom(x=x,size=n,p=p)`.

The probability distribution $P(X \leq x)$ is calculated using the `pbinom()` function. Syntax is:

`pbinom(x,size=n,p=p)`

The quantile for probability p can be evaluated using the `quantile()` function. Syntax is:

`qbinom(prob,size,p=p)`

3.14. Algorithm

- Step 1: Assign the inputs for required distribution
- Step 2: Calculate the required probabilities
- Step 3: Report the results

Case: Find the mass function of a binomial distribution with $n = 20, p = 0.4$. Also draw the graphs of the mass function and cumulative distribution function.

3.15. R code

```
# create input parameters
n=20
p=0.4
x=0:20
```

3.15.1. Prbability distribution

```
#calculating probability mass distribution and cumulative distribution
pmval=dbinom(x,size=n,prob=p)
pmval
```

```
##  [1] 3.656158e-05 4.874878e-04 3.087423e-03 1.234969e-02 3.499079e-02
##  [6] 7.464702e-02 1.244117e-01 1.658823e-01 1.797058e-01 1.597385e-01
## [11] 1.171416e-01 7.099488e-02 3.549744e-02 1.456305e-02 4.854351e-03
## [16] 1.294494e-03 2.696862e-04 4.230371e-05 4.700412e-06 3.298535e-07
## [21] 1.099512e-08
```

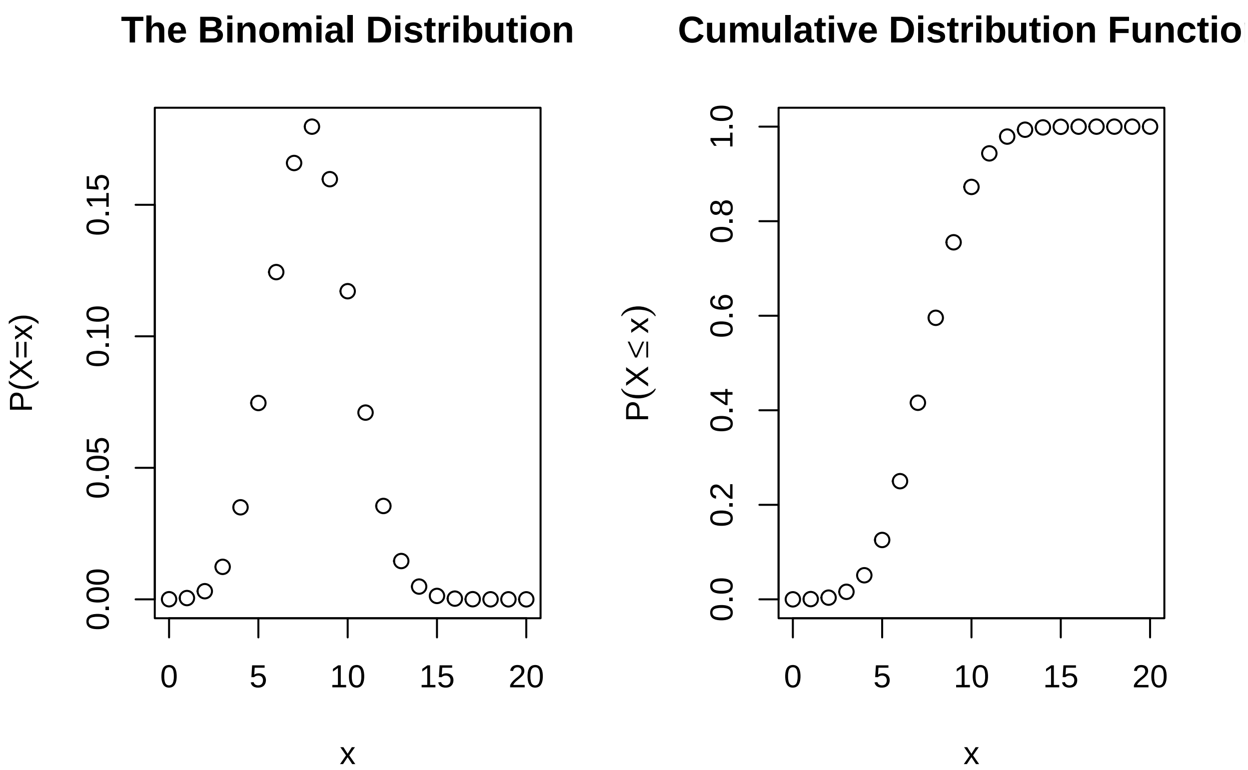
3.16. Cumulative probability distribution

```
#calculating cumulative density
cdval=pbinom(x,size=n,prob=p)
cdval
```

```
##  [1] 3.656158e-05 5.240494e-04 3.611472e-03 1.596116e-02 5.095195e-02
##  [6] 1.255990e-01 2.500107e-01 4.158929e-01 5.955987e-01 7.553372e-01
## [11] 8.724788e-01 9.434736e-01 9.789711e-01 9.935341e-01 9.983885e-01
## [16] 9.996830e-01 9.999527e-01 9.999950e-01 9.999997e-01 1.000000e+00
## [21] 1.000000e+00
```

3.16.1. Plotting the `pmf` and `cdf`

```
par(mfrow=c(1,2))
plot(x,pmval,xlab="x",ylab="P(X=x)", main="The Binomial Distribution")
plot(x,cdval,xlab="x",ylab=expression(P(X<=x)),main="Cumulative Distribution Function")
```



3.17. Results & discussions

The `pmf` and `cf` of the Binomial distribution for given input parameters are evaluated and create respective plots.

References