GFreya' R for Statistics

DS Glanzsche & Freya[1]

[1]A thank you or further information

Contents

Preface

For my future human Wife and our future biological daughters.

For my Divine Wife Freya the Goddess, and our daughters Catenary, Solreya, Mithra, Iyzumrae and Zefir.

For Lucrif and Znane too along with all the 8 Queens (Mischkra, Caldraz, Zalsvik, Zalsimourg, Hamzst, Lasthrim).

To Nature(Kala, Kathmandu, Big Tree, Sentinel, Aokigahara, Hoia Baciu, Jacob's Well, Mt Logan, etc) and my family Berlin: I have served, I will be of service.

To my current mentor Albert Silverberg and previous mentor Lucretia Merces.

To my dogs who always accompany me working in Valhalla Projection, go to Puncak Bintang or Kathmandu: Kecil, Browni Bruncit, Sweden Sexy, Cambridge Klutukk, Milan keng-keng, Piano Bludut, Barron and more will be adopted. To my cat who guard the home while I'm away with my dogs: London.

The one who moves a mountain begins by carrying away small stones - Confucius

A book for learning Statistics with R programming language that I am learning from zero. Helped by Freya the Goddess, Berlin, and Sentinel.

Figure 1: *FreyaCompass, I am inspired by Captain America who always bring compass with the love of his life' picture, thus I created this, then proven by action, to let go of power and immortality for true love. Feels like an antique vintage magical compass, like a modem that connect internet to the world, this compass connects me on this planet to her in Valhalla.*

Figure 2: *Freya, thank you for everything, I am glad I marry you and I could never have done it without you.*

Figure 3: *I paint her 3 days before Christmas in 2021.*

For critics and comments on the book can be sent through email to: *dsglanzsche@gmail.com*.

Chapter 1

Introduction and Installation of R

*An opportunity missed is an opportunity wasted! - **Seed (Suikoden II)***

This book is written on February 21st, 2025. Since 2022 we have been focusing on creating C++ codes for simulation and computation for Mathematics and Physics problems, they are all good, fast, but then we want to open a new horizon of knowledge, we read a book about R [1], it is said that we can do deep statistical analysis faster with R, given that the packages are already mature and the support is enormous, there is already a book series called 'The R book series' that can help statisticians and practitioners all over the world. I personally only know **Armadillo** library in C++ language that can compute mean, standard deviation, but then I think that basic statistics is not enough. If we want to do more with the data, i.e., generalized linear models, generalized additive models, mixed-effects models, non-linear regression, time series analysis, multivariate statistics, survival analysis, then we can count on R language. R can produce beautiful plot and simulation with refined statistical analysis.

All the codes, CSV and book is available on this github' repository:
https://github.com/glanzkaiser/GFreya-R-for-Statistics

I. Introduction

[R*] The choice between R and C++ depends on your specific needs and the context in which you're working. If you want to focus on data science and data analysis use R. If you want to code embedded system, a micro controller, create game engines, create PC game (like GTA V, Skyrim, Quake 3, Doom 3, Assassin's Creed), desktop app then we use C++.

[R*] **The Pros of R**

1. Statistical Analysis: R is specifically designed for statistics and data analysis, making it ideal for data scientists and statisticians.
2. It has a vast collection of packages (like ggplot2, dplyr, and tidyverse) that simplify data manipulation and visualization.
3. R is generally easier to learn for beginners, especially those focused on data analysis.
4. There is a strong community around R, particularly in academia and research.

The Cons of R

1. R can be slower than C++ for computationally intensive tasks because it's an interpreted language.

2. R abstracts many details away from the user, which can be limiting for low-level programming needs.

[R*] **The Pros of C++**

1. C++ is a compiled language, which typically results in faster execution times, making it suitable for performance-critical applications.

2. It offers more control over system resources and memory management, which is beneficial for system-level programming or applications requiring optimization.

3. C++ can be used for a wide range of applications beyond data analysis, including game development, systems programming, and application development.

The Cons of C++

1. C++ has a steeper learning curve than R, particularly due to its syntax and concepts like pointers and memory management.

2. While there are libraries available (like Armadillo and Eigen), C++ is not as tailored for statistical analysis as R.

Choose C++ if you need high performance, are developing complex systems, or require fine control over system resources.

II. Download and Installing R

We are going to use **GFreya OS 1.8**, it is built based on Linux From Scratch and Beyond Linux From Scratch version 11.0 System V.

[R*] First download the newest R tarball from this link:
https://cran.r-project.org/src/base/R-4/R-4.4.2.tar.gz

we also have the tarball, you can check the github repo for this book here:
https://github.com/glanzkaiser/GFreya-R-for-Statistics/blob/main/Source%20Codes/R-4.4.2.tar.gz

[R*] After you download and then open terminal and type at the directory containing the downloaded R and type:
tar -xvf R-4.4.2.tar.gz
cd R-4.4.2
./configure
make
make install

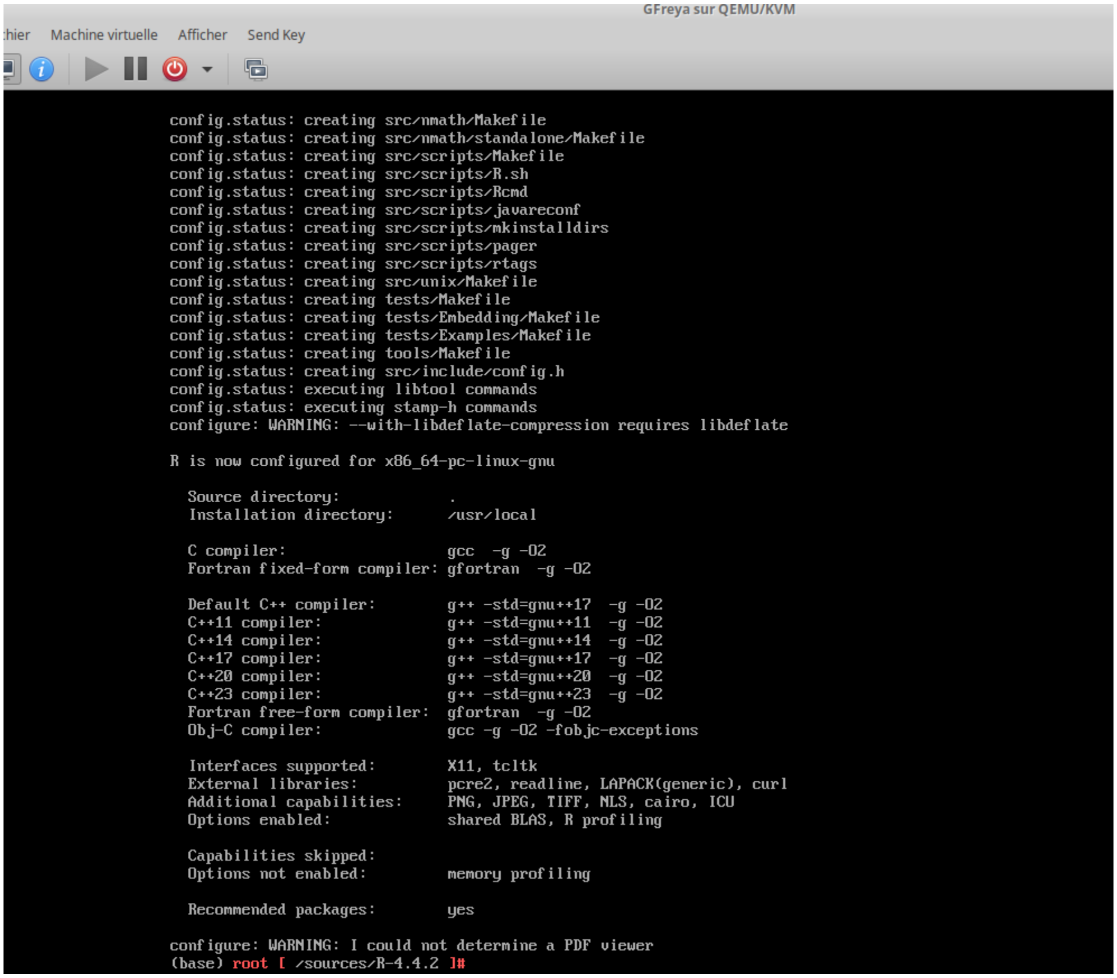


Figure 1.1: *If the **./configure** runs smoothly it will look like this.*

By default it is installed in **/usr/local/bin**, now you need to do one more important thing so you can call R from any directory.

Add the installation path of R to the $PATH environment variable
in GFreya OS go to root **cd /**
vim export

then press Esc and type **:wq** to save the contents and then quit, to quit without saving type **:q**, if you made a mistake and want to quit without saving type: **:q!** (these are some shell scripts when editing using vim editor).

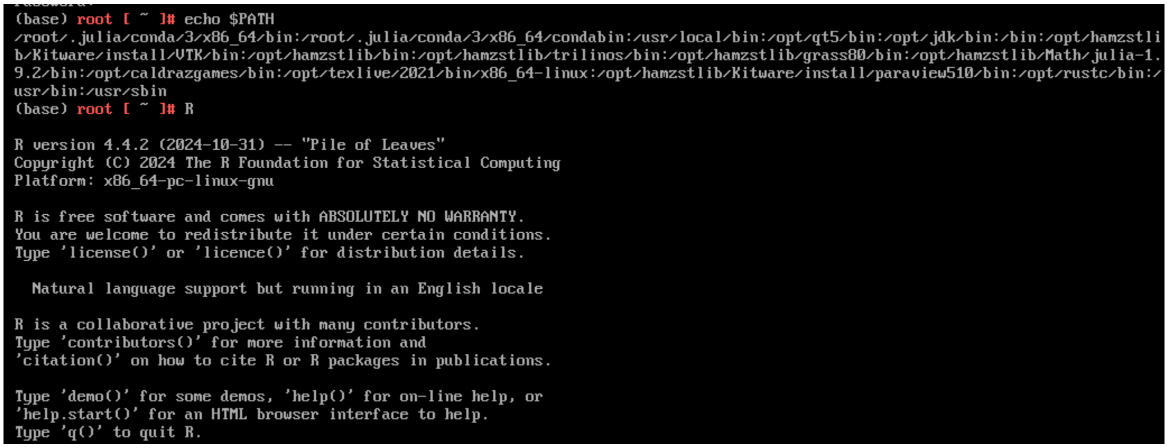


Figure 1.2: *Add the **/usr/local/bin** to the PATH environment variable to be able to run R from anywhere.*

add /usr/local/bin in PATH, then restart the computer and then check by typing in terminal:
echo $PATH
then you can now call R by typing:
R

[R*] How to install for Unix-like system can be seen from here:
https://cran.r-project.org/doc/manuals/r-devel/R-admin.html

[R*] **When Opening R**
Below the header you will see a blank line with a > symbol in the left hand margin. This is called the prompt. When working, you will sometimes see + at the left-hand side of the screen instead of >. This means that the last command you typed is incomplete.

To view the list of the already installed packages on your computer, type :
installed.packages()

If you want to update all installed R packages, type :
update.packages()

To update specific installed packages, say readr and ggplot2, type:
update.packages(oldPkgs = c('readr', 'ggplot2'))

To install a package, e.g. ggplot2, type:
install.packages('ggplot2')

you can then choose the CRAN (Comprehensive R Archive Network). mirror by typing a number representing which location for the mirror.

We can use the same function to install several R packages at once. In this case, we need to apply first the c() function to create a character vector containing all the desired packages as its items:
install.packages(c('readr', 'ggplot2', 'tidyr'))

Above, we've installed three R packages: the already-familiar readr, ggplot2 (for data visualization), and tidyr (for data cleaning).

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures. **install.packages('tidyverse')**

III. Data Mining

Since copy and pasting R codes from github or internet can be done easily but it will be resulting in better computer and AI but more stupid human, so we will need to know why we use that code to process that data set / database into knowledge. In learning the how, we will use Data Mining technique / methods with statistics as the basic tool for Data Mining that will be explained later on.

Data mining is the process of discovering insightful, interesting, and novel patterns, as well as descriptive, understandable and predictive models from large-scale data [6].

Data mining is also the last step to Knowledge Discovery in Databases (KDD), we can use R, Python, C++, JULIA, to help doing data mining. Data mining is used to create a model for categorization or prediction.

This is one famous example of Data Mining [4]:
After 9/11, Bill Clinton announced that after examining lots of databases, FBI agents discovered that 5 of the perpetrators were registered to these databases. One of them owned 30 credit cards with a negative balance of USD 250.000 and lived in US for less than two years.

We can say that people with a lot of debts are prone to commit crime, from becoming a petty theft to commit a big act like terrorist in 9/11, combine with the length of stay in USA.

With R, we can easily see the correlation of each variable, from FBI' crime database / data set, we can even analyze what makes someone become a terrorist? Is it net worth? unemployment status? country of birth? religion? highest level education? After we obtain the knowledge from Data Mining and by applying this knowledge correctly into society, we can apply better policy to accept immigrant and new citizen in any country, we can have several criteria that have red flags showing that someone has potential to be terrorist, rapist, murderer, etc. So we cannot ban all Mexican or all African from becoming US citizen, we only ban those with potential to harm other US' citizens and make US economy slump.

i. Data Mining Methods

1. Classification
   A predictive method. Its goal is to create a model - classifier based on current data.

2. Regression
   A predictive method by using some independent variables its goal is to predict the values of a dependent variable.

3. Clustering
   A descriptive method to create clusters / groups with similar feature.

4. Extraction and Association Analysis

   A classic example of association rules in practice has to do with the analysis of a shopping cart in a super market, where data have to do with clients transactions. In this scenario, some transactions could be {yamazaki bread, oatmilk}, {sari roti bread, KitKat chocolate, Hazelnut Crumpy}, { rice, eggs}, {egg roll, khong guan, dancow milk }. We can tell with a shopping cart analysis the probability of someone buying bread will also buy milk, thus the placement for bread and milk should be located nearby or quite far so they can see other stuff in the grocery and buy more, it is a marketing tactic.

5. Visualization
   Data visualization helps in better understanding not only the data themselves but also correlations that might occur between them.

6. Anomaly Detection
   Anomaly detection focuses in finding deviations in data according to similar data collected in the past or by typical values of these data.

IV. Know-How in R

[R*] Learning how to handle your data, how to enter it into the computer, and how to read the data into R are amongst the most important topics you will need to master. R handles data in objects known as dataframes [1]. A dataframe is an object with rows and columns (a bit like a matrix). The rows contain different observations from your study, or measurements from your experiment. The columns contain the values of different variables. The values in the body of a matrix can only be numbers; those in a dataframe can also be numbers, but they could also be text (i.e., the names of factor levels for categorical variables, like male or female in a variable called gender), they could be calendar dates (i.e., 23/5/04), or they could be logical variables (TRUE or FALSE).

[R*] Producing high-quality graphics is one of the main reasons for doing statistical computing with R. The particular plot function you need will depend on the number of variables you want to plot and the pattern you wish to highlight.

With two variables (typically the response variable on the y axis and the explanatory variable on the x axis), the kind of plot you should produce depends upon the nature of your explanatory variable. When the explanatory variable is a continuous variable, such as length or weight or altitude, then the appropriate plot is a scatterplot.

In cases where the explanatory variable is categorical, such as genotype or colour or gender, then the appropriate plot is either a box-and-whisker plot (when you want to show the scatter in the raw data) or a barplot (when you want to emphasize the effect sizes).

[R*] If you want to convey detail use a table, and if you want to show effects then use graphics. You are more likely to want to use a table to summarize data when your explanatory variables are categorical (such as people's names, or different commodities) than when they are continuous (in which case a scatterplot is likely to be more informative.

Chapter 2

Data Visualization

You don't need qualifications to make a difference. - ***Yun (Suikoden III)***

WE will start with a simple plotting then learning some basic and formulas in statistics and probability to create deep and more complex with more meaningful data visualization.

All the codes, CSV and book is available on this github' repository:
https://github.com/glanzkaiser/GFreya-R-for-Statistics

i. Scatter Plot with ggplot2

[R*] We will use CSV from the github' repository:
https://github.com/glanzkaiser/GFreya-R-for-Statistics/CSV/insurance.csv

put this CSV in the working directory.

[R*] To open the desktop environment of GFreya OS, type:
startx

[R*] Open R from the working directory, from the current working directory open the terminal and type:
R

Load the necessary library:
library(ggplot2)

To import the data and look at the first six rows **insurance <- read.csv('insurance.csv')**

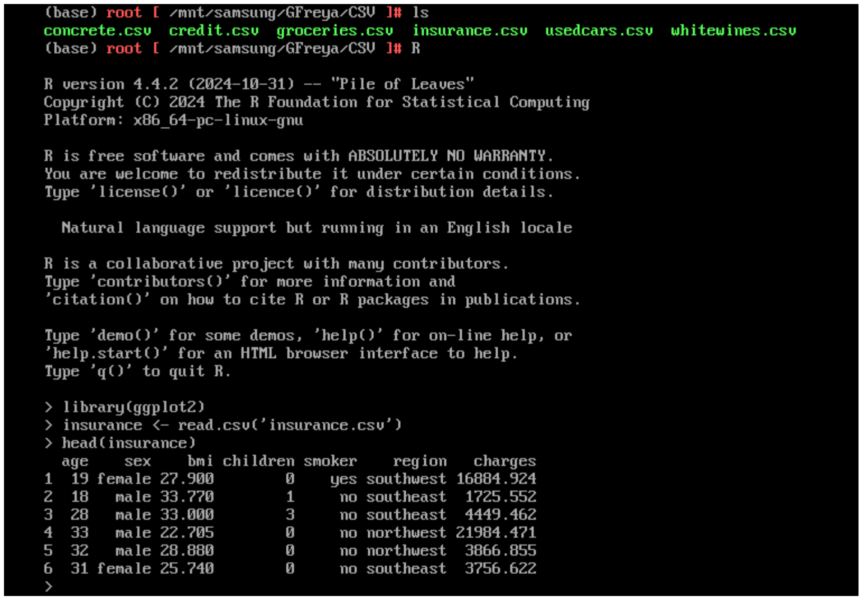


Figure 2.1: *To look at the top 6 rows of the data from CSV file.*

p <- ggplot(insurance, aes(x=age, y=charges, colour=sex)) + geom_point() + scale_color_manual(values =c('red', 'blue'))

Geoms are the geometric objects (points, lines, bars, etc.) that can be placed on a graph. They are added using functions that start with **geom_**.

In **ggplot2** graphs, functions are chained together using the + sign to build a final plot.

To save the plot as png, type:
png("plot.png")
print(p)
dev.off()

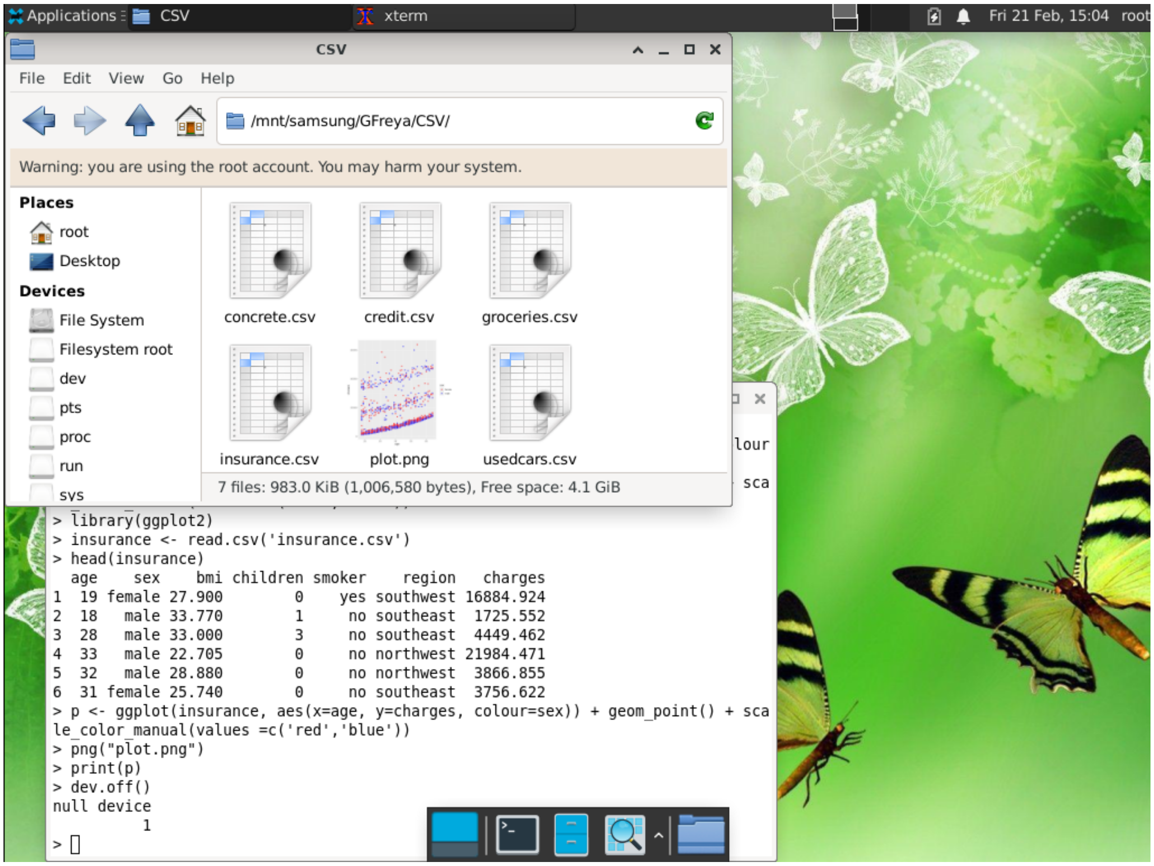


Figure 2.2: *The process to plot the scatter plot with the x axis representing the age, the y axis representing the insurance charges and the color to separate male and female.*

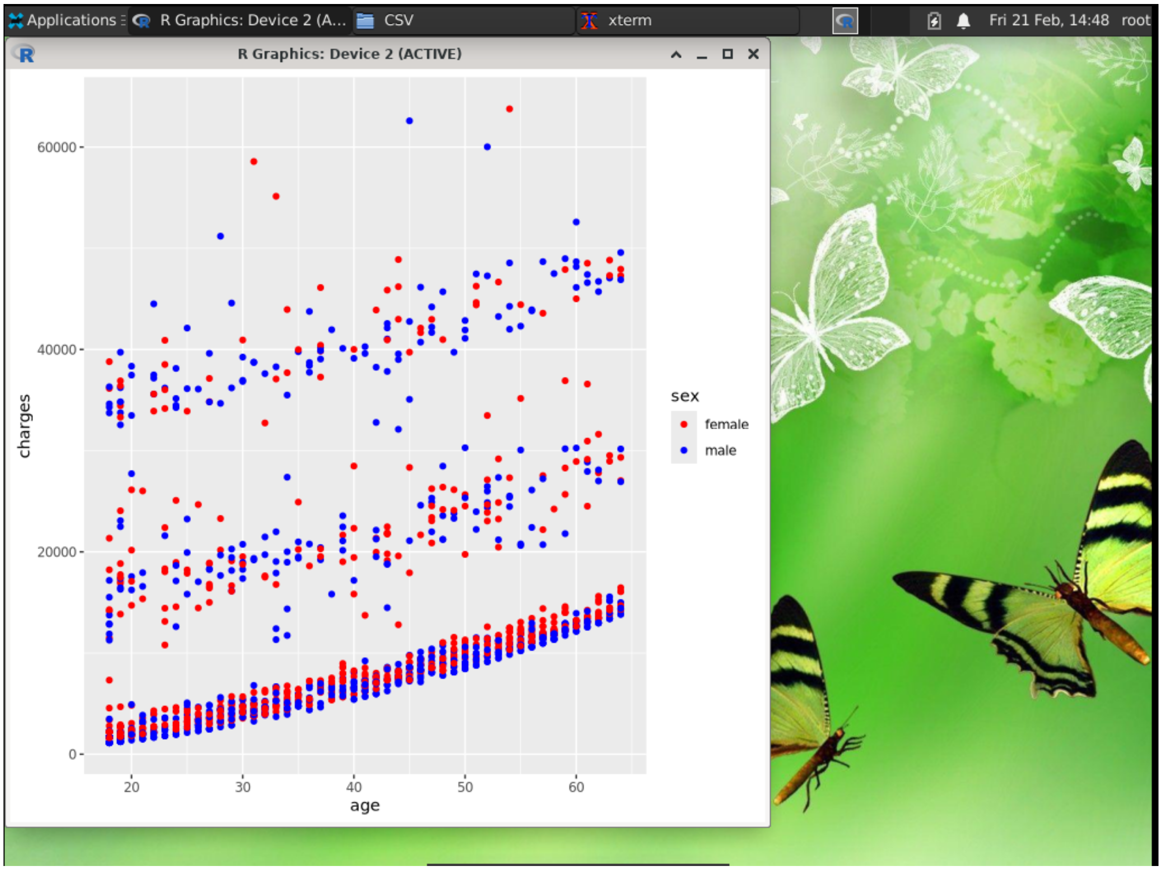


Figure 2.3: .

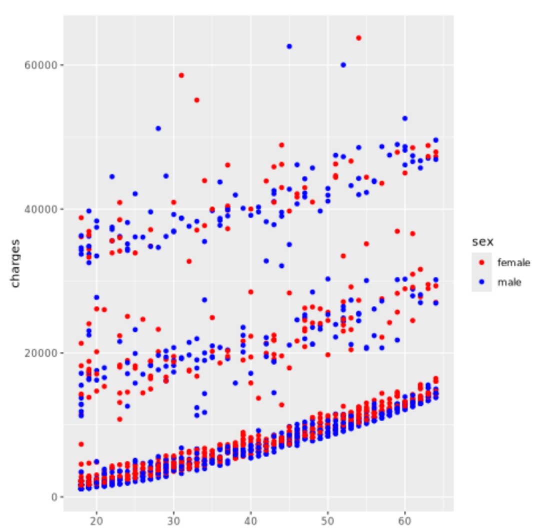


Figure 2.4: *The full picture.*

The whole code:

```
library(ggplot2)

insurance <- read.csv('insurance.csv')
p <- ggplot(insurance, aes(x=age, y=charges, colour=sex)) +
    geom_point() + scale_color_manual(values =c('red', 'blue'))

png("plot.png")
print(p)
dev.off()
```

R Code 1: *the first plot with ggplot2 (ch2-scatterplot.R)*

ii. Two Scatter Plots for Comparing with ggplot2

In this section, we will create two scatter plots, more complex graph to explores the relationship between smoking, obesity, age, and medical costs using the data from the Medical Insurance Costs dataset / **insurance.csv**.

I won't talk too many details and for further explanation you can read this book [2].

[R*] We will use CSV from the github' repository:
https://github.com/glanzkaiser/GFreya-R-for-Statistics/CSV/insurance.csv

put this CSV in the working directory.

[R*] To open the desktop environment of GFreya OS, type:
startx

[R*] Open R from the working directory, from the current working directory open the terminal and type:
R

Load the necessary library:
library(ggplot2)

To import the data and look at the first six rows **insurance <- read.csv('insurance.csv')**

(alternative way to load / read **insurance.csv**) If you want to learn, there is this url that contains **insurance.csv**, it is on:
https://raw.githubusercontent.com/dataspelunking/MLwR/master /Machine%20Learning%20with%20R%20(3rd%20Ed.)/Chapter06/insurance.csv

To obtain the insurance CSV data from a url page, type:
url <- "https://tinyurl.com/mtktm8e5"
insurance <- read.csv(url)

Now, beware that the column title for the last column is **expenses** instead of **charges**, so adjust that, or for the better, just stick with manually read the already available **insurance.csv** from the github' repository of this book.

[R*] Now, without a lot of wasting time, you already know how to make a simple scatter plot, I will show the whole code to produce the plot in this section:

```
library(ggplot2)

insurance <- read.csv('insurance.csv')

# create an obesity variable
insurance$obese <- ifelse(insurance$bmi >= 30,"obese", "not
    obese")

p <- ggplot(data = insurance,mapping = aes(x = age,y = charges
    ,color = smoker)) + geom_point(alpha = .5) + geom_smooth(
    method = "lm", se = FALSE) +
scale_x_continuous(breaks = seq(0, 70, 10)) +
    scale_y_continuous(breaks = seq(0, 60000, 20000), label =
    scales::dollar) +
scale_color_manual(values = c("indianred3","cornflowerblue")) +
     facet_wrap(~obese) +
labs(title = "Relationship between age and medical expenses",
subtitle = "US Census Data 2013",
caption = "source: https://github.com/stedy/Machine-Learning-
    with-R-datasets/",
x = " Age (years)",
y = "Medical Expenses",
color = "Smoker?") +
theme_minimal()

png('plot.png')
print(p)
dev.off()
```

R Code 2: *two scatter plots with ggplot2 (ch2-twoscatterplots.R)*

Theme functions (which start with **theme_**) control background colors, fonts, grid-lines,legend placement, and other non-data related features of the graph.

Now, instead of typing one by one in the R console, we can be smart and open a text editor or vim editor or a notepad++ then just copy the whole codes above and save it with extension of **.R** and then we can use the source function, so put this **twoscatterplots.R** along with the CSV file if you wish to load it offline / from localhost then open the terminal at the current working directory and type:
R
source('twoscatterplots.R')

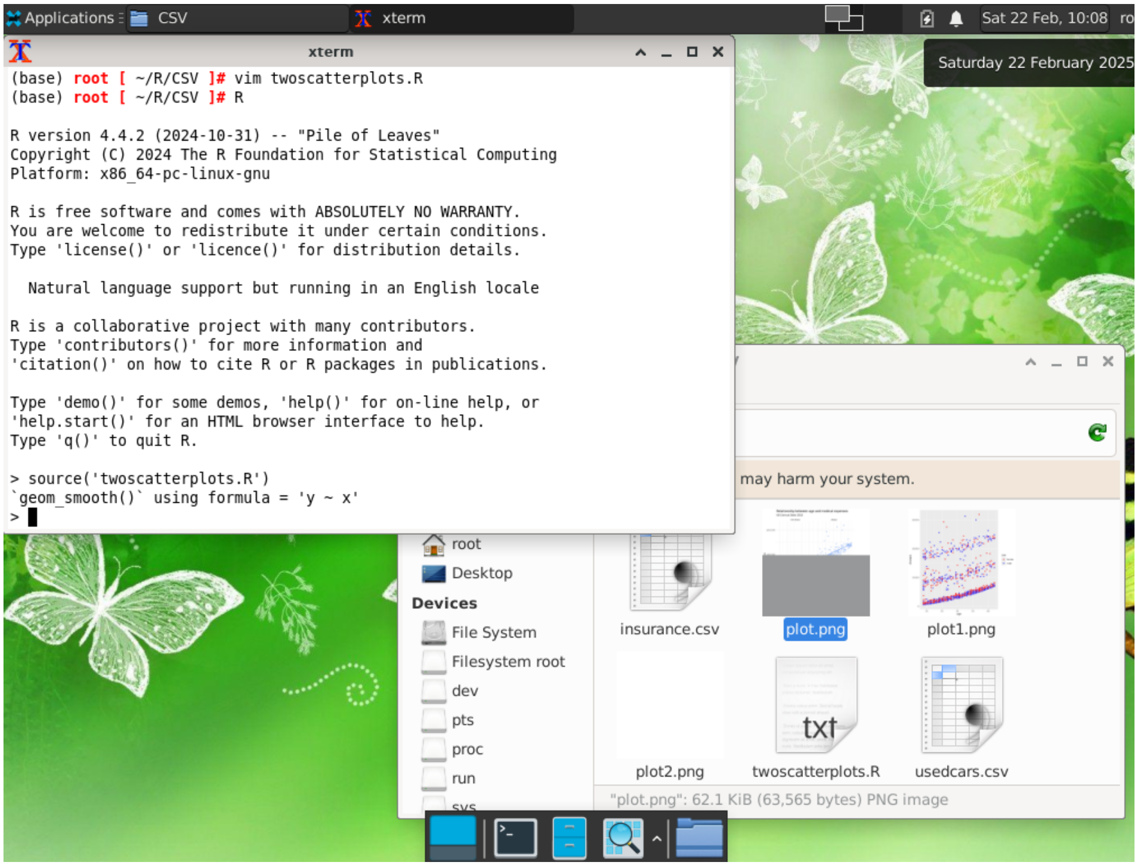


Figure 2.5: *Nothing comes in an instant.*

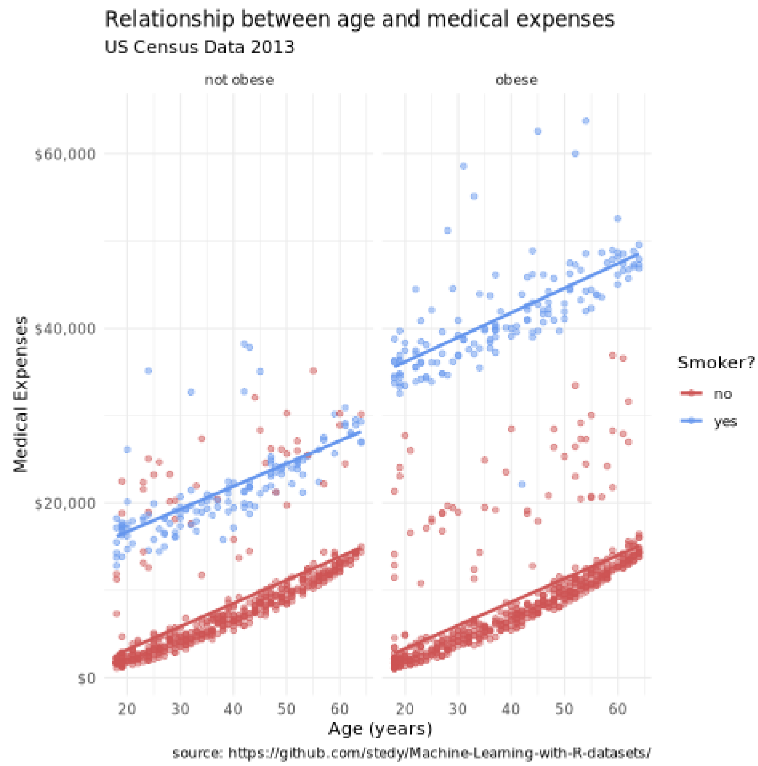


Figure 2.6: *We can see that smokers and obese patients have higher medical charges / expenses.*

iii. Univariate Graphs for Categorical Variables: Bar Chart with ggplot2 and dplyr

Univariate graphs plots the distribution of data from a single variable. The variable can be categorical (i.e., race, sex, political affiliation) or quantitative (i.e., age, weight, income).

In this section we will plot a bar chart from the dataset Marriage that contains the marriage records of 98 individuals in Mobile County, Alabama (from the package **mosaicData**).

Pie charts are controversial in statistics. If your goal is to compare the frequency of categories, you are better off with bar charts (humans are better at judging the length of bars than the volume of pie slices).

[R*] We want to create a descending bar chart as it is easier to gain the knowledge from the data, most people' brain work better by ordering. It is often helpful to sort the bars by frequency.

The **reorder** function is used to sort the categories by the frequency. The option **stat="identity"** tells the plotting function not to calculate counts, because they are supplied directly.

The minus sign in **reorder(race, -pct)** is used to order the bars in descending order.

```
# simple bar chart
library(ggplot2)
```

```
data(Marriage, package = "mosaicData")

# calculate number of participants in each race category
library(dplyr)

plotdata <- Marriage %>% count(race) %>% mutate(pct = n / sum(
    n), pctlabel = paste0(round(pct*100), "%"))

# plot the bars as percentages,
# in decending order with bar labels
p <- ggplot(plotdata, aes(x = reorder(race, -pct), y = pct)) +
geom_bar(stat="identity", fill="navyblue", color="black") +
geom_text(aes(label = pctlabel), vjust=-0.25) +
scale_y_continuous(labels = scales::percent) +
labs(x = "Race", y = "Percent", title = "Participants by race")

print(p)
```

R Code 3: *barchart with descending order (ch2-barchart.R)*

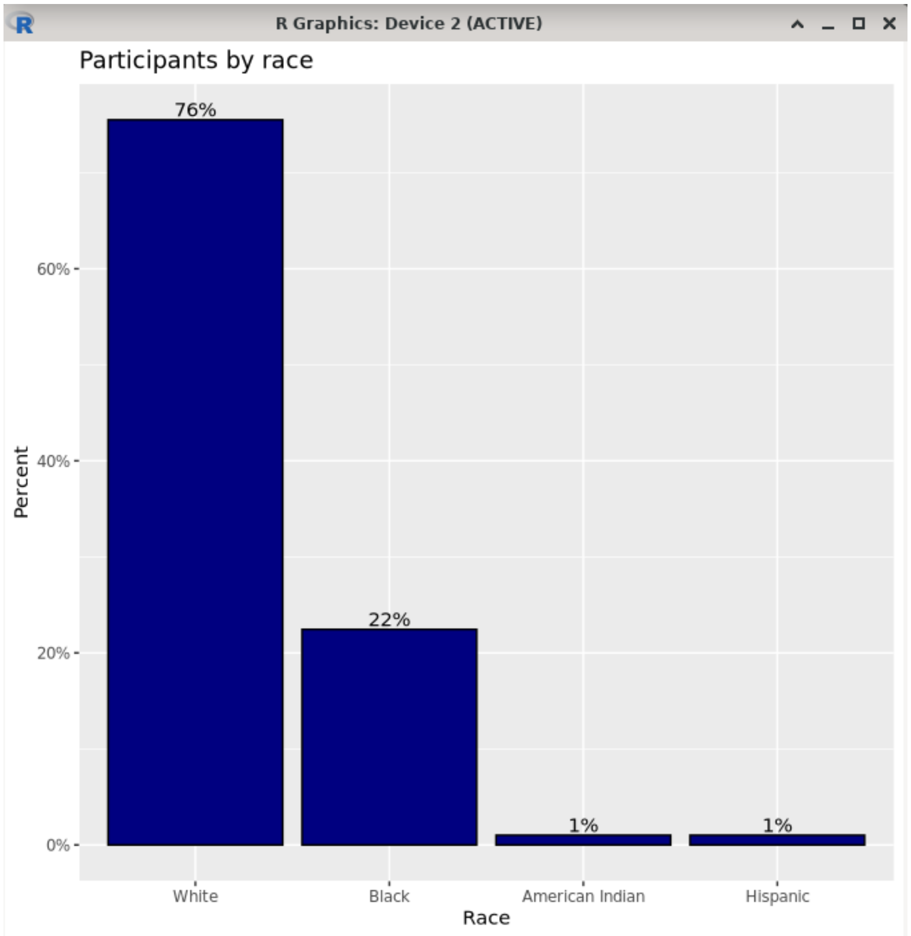


Figure 2.7: *The bar chart with rotated label.*

[R*] To solve a problem where category labels may overlap, we usually rotate the labels. Below is the code to rotate the label counterclockwise 45 degree.

```
# simple bar chart
library(ggplot2)

data(Marriage, package = "mosaicData")

# bar chart with rotated labels
p <- ggplot(Marriage, aes(x=officialTitle)) +
geom_bar(fill="navyblue", color="black") +
labs(x = "", y = "Frequency", title = "Marriages by officiate")
    +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(p)
```

R Code 4: *barchart with rotated label (ch2-barchart-rotatedlabels.R)*

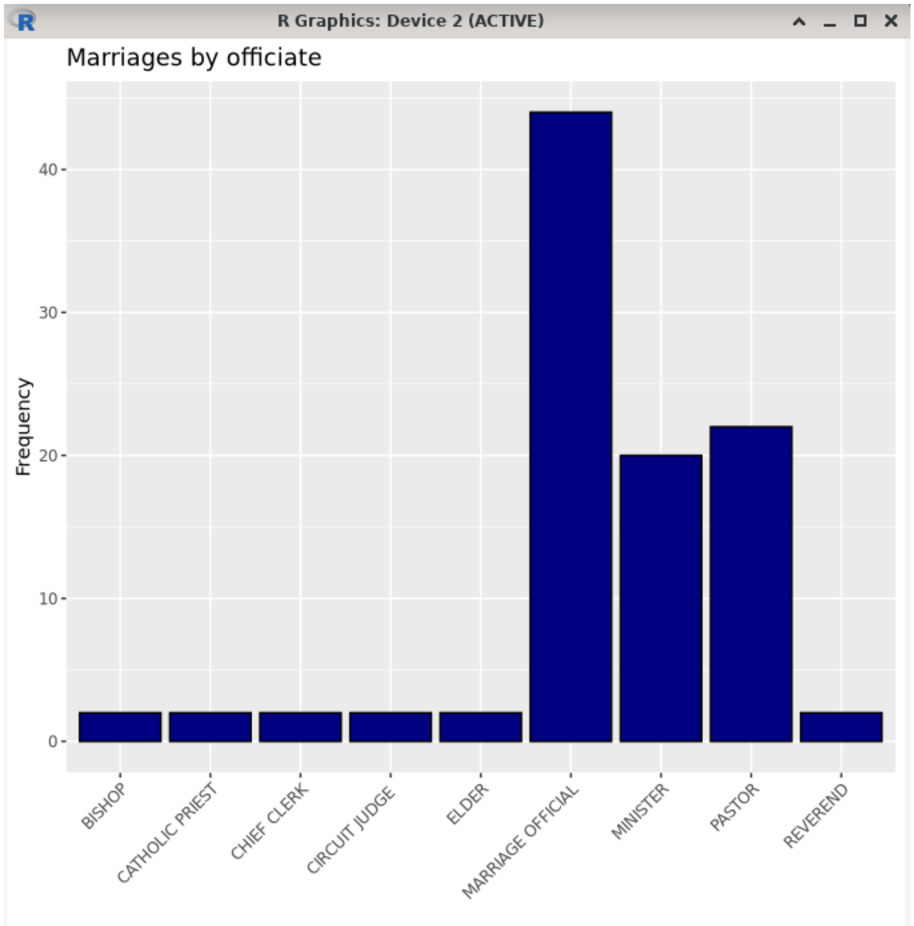


Figure 2.8: *The bar chart with rotated label.*

iv. Univariate Graphs for Categorical Variables: Tree Map with ggplot2

[R*] An alternative to a pie chart is a tree map. Unlike pie charts, it can handle categorical variables that have many levels. It is often used in The Economists magazine.

```
# simple bar chart
library(ggplot2)

data(Marriage, package = "mosaicData")

# bar chart with rotated labels
p <- ggplot(Marriage, aes(x=officialTitle)) +
geom_bar(fill="navyblue", color="black") +
labs(x = "", y = "Frequency", title = "Marriages by officiate")
    +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(p)
```

R Code 5: *barchart with rotated label (ch2-treemap.R)*

Figure 2.9: *The tree map of marriage officials with labels.*

v. Univariate Graphs for Quantitative Variables: Histogram with ggplot2

In the **Marriage** dataset, age is quantitative variable. The distribution of a single quantitative variable is typically plotted with a histogram, kernel density plot, or dot plot. In this section we will create a histogram.

Histograms [2] are the most common approach to visualizing a quantitative variable. In a histogram, the values of a variable are typically divided up into adjacent, equal width ranges (called bins), and the number of observations in each bin is plotted with a vertical bar.

One of the most important histogram options is bins, which controls the number of bins into which the numeric variable is divided (i.e., the number of bars in the plot). The default is 30, but it is helpful to try smaller and larger numbers to get a better impression of the shape of the distribution.

[R*] The first histogram is a simple histogram with **binwidth = 5**.

```
library(ggplot2)

data(Marriage, package = "mosaicData")

# displays the data with binwidth that are 5 years wide
p <- ggplot(Marriage, aes(x = age)) +
geom_histogram(fill = "navyblue", color = "white", binwidth =
    5) +
labs(title = "Participants by age", subtitle = "binwidth = 5
    years", x = "Age")

print(p)
```

R Code 6: *histogram with binwidth 5 (ch2-histogram.R)*

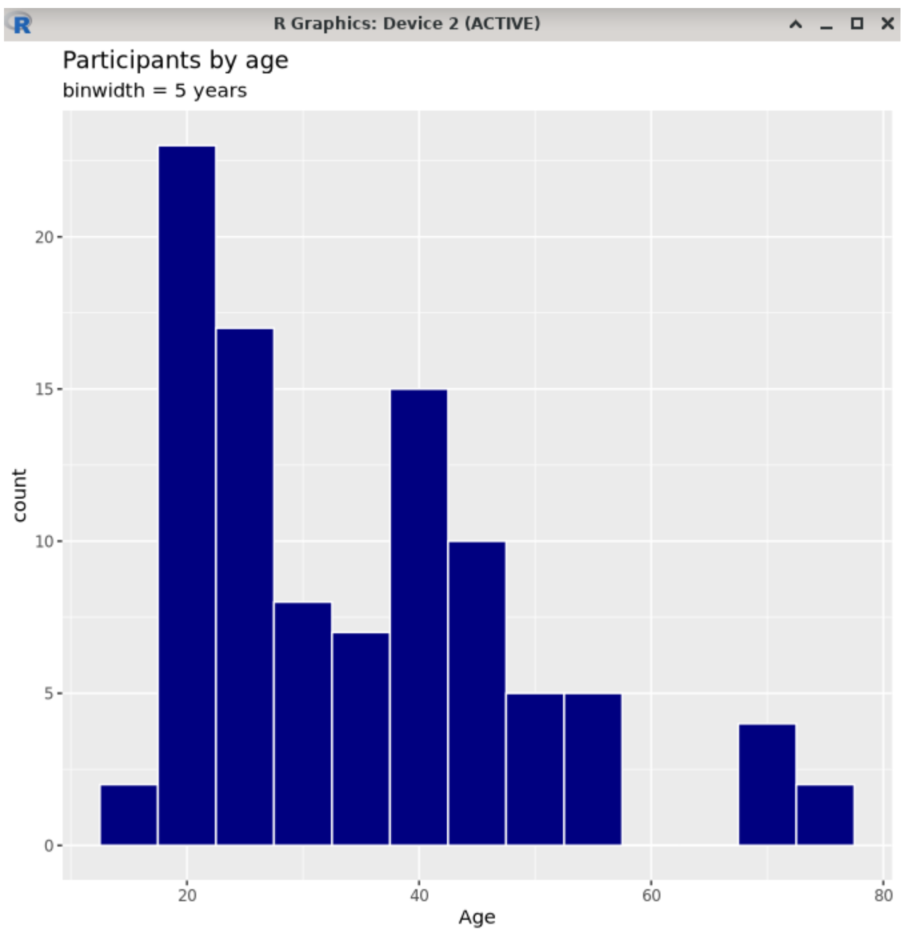


Figure 2.10: *The histogram with binwidth=5.*

[R*] The second histogram plot the histogram with percentages on the y-axis.

```
library(ggplot2)

data(Marriage, package = "mosaicData")

# plot the histogram with percentages on the y-axis
library(scales)

p <- ggplot(Marriage, aes(x = age, y = after_stat(count/sum(
     count)))) +
geom_histogram(fill = "navyblue", color = "white", binwidth =
     5) +
labs(title="Participants by age", y = "Percent", x = "Age") +
scale_y_continuous(labels = percent)

print(p)
```

R Code 7: *histogram with percentages on y axis (ch2-histogram.R)*

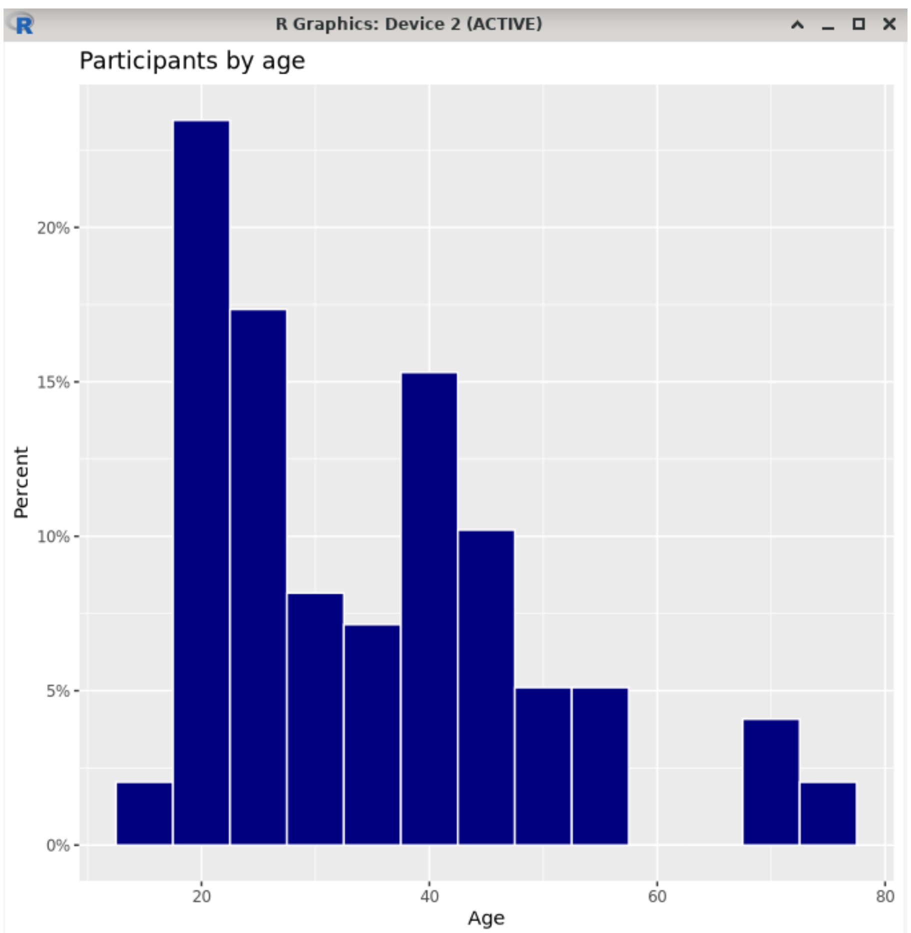


Figure 2.11: *The histogram with percentages on the y-axis.*

vi. Univariate Graphs for Quantitative Variables: Kernel Density Plot with ggplot2

An alternative to a histogram is the kernel density plot. Technically, kernel density estimation is a nonparametric method for estimating the probability density function of a continuous random variable.

A continuous random variable is a random variable that has only continuous values. Continuous values are uncountable and are related to real numbers. Examples: time, age, miles per gallon for a certain car.

Discrete Distributions	Continuous Distributions
Countable Discrete Points Points have probability $p(x)$ is probability **distribution** function $p(x) \geq 0$ $\Sigma p(x) = 1$	Uncountable Continuous Intervals Points have no probability $f(x)$ is probability **density** function $f(x) \geq 0$ Total Area under curve =1

Figure 2.12: *The similarities and differences between discrete and continuous distributions.*

In this section, we are trying to draw a smoothed histogram, where the area under the curve equals to one.

[R*] For this kernel density plot, the degree of smoothness is controlled by the bandwidth parameter **bw**. To find the default value for a particular variable, use the **bw.nrd0** function. Values that are larger will result in more smoothing, while values that are smaller will produce less smoothing.

```
library(ggplot2)

data(Marriage, package = "mosaicData")

p <- ggplot(Marriage, aes(x = age)) +
geom_density(fill = "navyblue", bw = 2) +
labs(title = "Participants by age",subtitle = "bandwidth = 2")

# default bandwidth for the age variable
# choosing a value that is less than bw.nrd0(Marriage$age) will
    resulting in less smoothing and more detail
bw.nrd0(Marriage$age)

png('plot.png')
print(p)
dev.off()
```

R Code 8: *kernel density plot with bandwidth 2 (ch2-kerneldensityplot.R)*

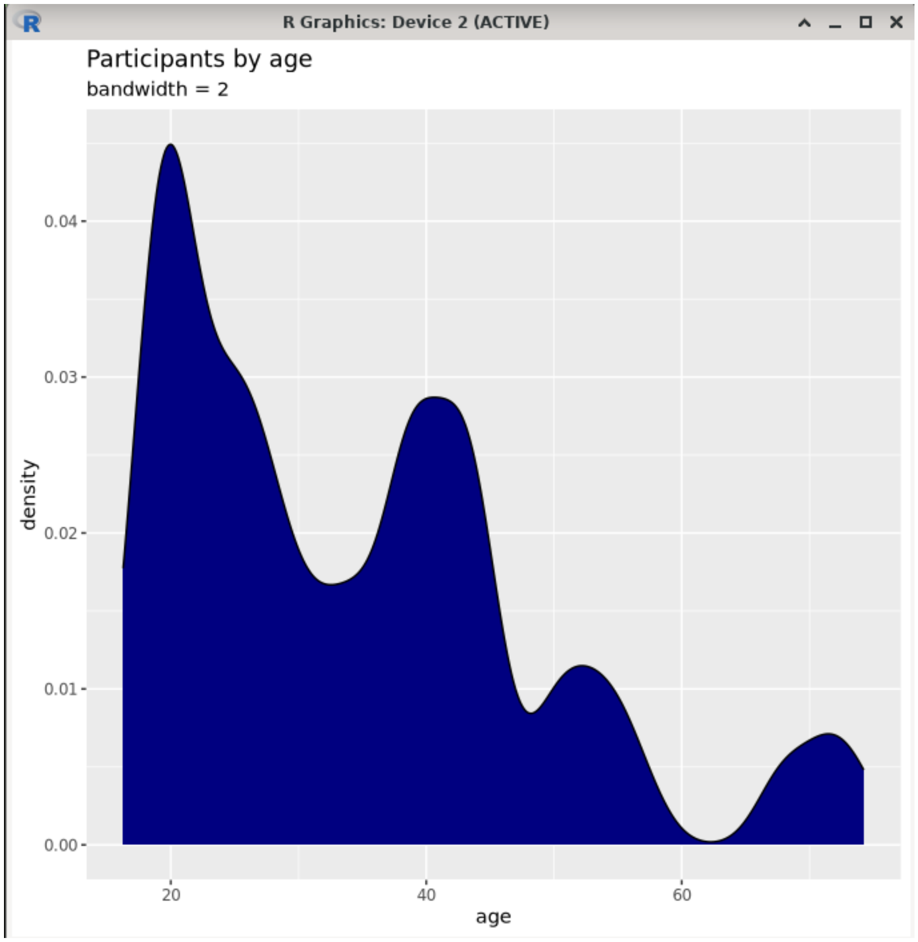


Figure 2.13: *The kernel density map with bandwidth = 2.*

Chapter 3

Descriptive Statistics

She's a little old-fashioned

She's a little new way

She's got her own kind of passion

And I'm glad to say

That I'm a lucky girl

She's standin' by

Can't you see, lucky me

And I really only wanna say

No one on this earth

Does what she does for me

Glanz to Freya the Goddess from World's Greatest Lover song by The Bellamy Brothers (1984)

WE have started to create a simple visualizations in the previous chapter, but if we only do that, like I did, checking on the books and test whether the plot can be shown then it is not really learning. We need to learn the basic definition, formula, not only type codes we obtain from books or internet and not even knowing what is the formula of variance, well that's classic, that is why great invention and innovation is very rare compared to graduates from prestigious universities, because graduates can cheat and get into university due to money and connection.

All the codes, CSV and book is available on this github' repository:
https://github.com/glanzkaiser/GFreya-R-for-Statistics

I. Basic Definition, Theory and Formula

i. The Sample Mean and Median

All the data set that can be gathered are only sample, even if it is of size of trillion Terabytes and have quadrillion times quadrillion of rows of data, they are still will be called a sample, since the population of data means we are gathering all from the beginning till the end of time or beyond.

So we are going to learn about sample mean and sample median [5], they can be called the tools to measure the location, to provide the analyst with some quantitative values of where the center, or some other location, of data is located.

Definition 3.1: Sample Mean

Suppose that the observations in a sample are $x_1, x_2, \ldots, x_n$. The sample mean, denoted by $\bar{x}$, is

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{3.1}$$

Definition 3.2: Sample Median

Given that the observations in a sample are $x_1, x_2, \ldots, x_n$, arranged in increasing order of magnitude, the sample median is

$$\begin{cases} x_{\frac{n+1}{2}}, & \text{if } n \text{ is odd} \\ \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}, & \text{if } n \text{ is even} \end{cases} \qquad \bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{3.2}$$

The purpose of the sample median is to reflect the central tendency of the sample in such a way that it is uninfluenced by extreme values or outliers.

ii. Measures of Variability

Sample variability plays an important role in data analysis. Process and product variability is a fact of life in engineering and scientific systems: The control of reduction of process variability is often a source of major difficulty.

There are many measures of spread or variability, the simplest one is the sample range.

Definition 3.3: Sample Range

The sample range, denoted by L, is given by

$$L = X_{max} - X_{min} \tag{3.3}$$

Definition 3.4: Sample Standard Deviation and Variance

The sample variance, denoted by s^2, is given by

$$s^2 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \tag{3.4}$$

The sample standard deviation, denoted by s, is the positive square root of s^2, that is,

$$s = \sqrt{s^2} \tag{3.5}$$

The sample standard deviation is a measure of variability. Large variability in a data set produces relatively large values of $(x_i - \bar{x})^2$ and thus a large sample variance. The quantity $n-1$ is often called the degrees of freedom associated with the variance estimate.

iii. Histogram

In the previous chapter, we have plot a histogram, but not really learning about how to make a histogram and the whole definition of a histogram.

Histogram is often called a relative frequency histogram, to create a histogram we will:

1. Choose a variable that we want to show with this histogram. It has to be variable with numerical data, such as battery life, age, salary.
2. The x axis will represents the variable chosen and the y axis of a histogram will be for the frequency of the variable chosen.
3. We choose the class interval / the bindwidth, it is the width of the histogram rectangle, the histogram has different height for all the rectangles but same size for the width. The height is used to measure the frequency. The choice of the class interval will determine the number of rectangles that will be shown in the histogram.

iv. Box Plot

Box plot encloses the interquartile range of the data in a box that has the median displayed within. The interquartile range has as its extremes the 75th percentile (upper quartile) and the 25th percentile (lower quartile). In addition to the box, "whiskers" extend, showing extreme observations in the sample. For reasonably large samples, the display shows center of location, variability, and the degree of asymmetry.

The visual information of box plot is not intended to be a formal test for outliers. Rather, it is viewed as a diagnostic tool.

II. Car Accident Analysis

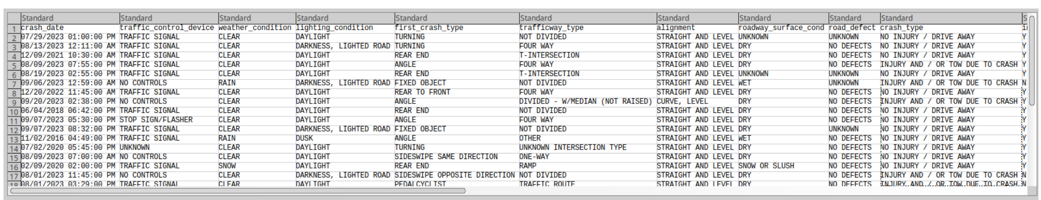
This section is made by learning from this page:
https://www.rpubs.com/rileyhamilton/1275148

We are going to use a traffic accident data set found on Kaggle:
https://www.kaggle.com/datasets/oktayrdeki/traffic-accidents

then we are going to create a lot of graphs and chart to represent some variables from the data set to gain insight of the data.

There are 209,306 rows of data and 24 variables. The data collection started in 2013, but 2013 has the least amount of observations. This data can be used to help identify conditions that make accidents more likely, or conditions that make injuries more likely, to find the days/times when accidents occur the most. We are going to learn and use descriptive statistics. From the data set a lot of the variables are categorical, so the basic descriptive statistics that are helpful come from injury variables, number of units, and the time information (day, hour, month).

All the source codes and the CSV are in: **/root/R/CSV/** in computer / localhost' path.



	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	S
1	crash_date	traffic_control_device	weather_condition	lighting_condition	first_crash_type	trafficway_type	alignment	roadway_surface_cond	road_defect	crash_type	i
2	07/29/2023 01:00:00 PM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	TURNING	NOT DIVIDED	STRAIGHT AND LEVEL	UNKNOWN	UNKNOWN	NO INJURY / DRIVE AWAY	Y
3	08/13/2023 12:11:00 AM	TRAFFIC SIGNAL	CLEAR	DARKNESS, LIGHTED ROAD	TURNING	FOUR WAY	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
4	12/09/2021 10:30:00 AM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	REAR END	T-INTERSECTION	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
5	08/09/2023 07:55:00 PM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	ANGLE	FOUR WAY	STRAIGHT AND LEVEL	DRY	NO DEFECTS	INJURY AND / OR TOW DUE TO CRASH	Y
6	08/19/2023 02:55:00 PM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	REAR END	T-INTERSECTION	STRAIGHT AND LEVEL	UNKNOWN	UNKNOWN	NO INJURY / DRIVE AWAY	Y
7	09/06/2023 12:59:00 AM	NO CONTROLS	RAIN	DARKNESS, LIGHTED ROAD	FIXED OBJECT	NOT DIVIDED	STRAIGHT AND LEVEL	WET	UNKNOWN	INJURY AND / OR TOW DUE TO CRASH	N
8	12/20/2022 11:45:00 AM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	REAR TO FRONT	FOUR WAY	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
9	09/20/2023 02:38:00 PM	NO CONTROLS	CLEAR	DAYLIGHT	ANGLE	DIVIDED - W/MEDIAN (NOT RAISED)	CURVE, LEVEL	DRY	NO DEFECTS	INJURY AND / OR TOW DUE TO CRASH	Y
10	06/04/2018 06:42:00 PM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	REAR END	NOT DIVIDED	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
11	09/07/2023 05:30:00 PM	STOP SIGN/FLASHER	CLEAR	DAYLIGHT	ANGLE	FOUR WAY	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
12	09/07/2023 08:32:00 PM	TRAFFIC SIGNAL	CLEAR	DARKNESS, LIGHTED ROAD	FIXED OBJECT	NOT DIVIDED	STRAIGHT AND LEVEL	DRY	UNKNOWN	NO INJURY / DRIVE AWAY	Y
13	11/02/2016 04:49:00 PM	TRAFFIC SIGNAL	RAIN	DUSK	ANGLE	OTHER	STRAIGHT AND LEVEL	WET	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
14	07/02/2020 05:45:00 PM	UNKNOWN	CLEAR	DAYLIGHT	TURNING	UNKNOWN INTERSECTION TYPE	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
15	08/09/2023 07:00:00 AM	NO CONTROLS	CLEAR	DAYLIGHT	SIDESWIPE SAME DIRECTION	ONE-WAY	STRAIGHT AND LEVEL	DRY	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
16	02/09/2020 02:00:00 PM	TRAFFIC SIGNAL	SNOW	DAYLIGHT	REAR END	RAMP	STRAIGHT AND LEVEL	SNOW OR SLUSH	NO DEFECTS	NO INJURY / DRIVE AWAY	Y
17	08/01/2023 11:45:00 PM	NO CONTROLS	CLEAR	DARKNESS, LIGHTED ROAD	SIDESWIPE OPPOSITE DIRECTION	NOT DIVIDED	STRAIGHT AND LEVEL	DRY	NO DEFECTS	INJURY AND / OR TOW DUE TO CRASH	N
18	08/01/2023 03:29:00 PM	TRAFFIC SIGNAL	CLEAR	DAYLIGHT	PEDALCYCLIST	TRAFFIC ROUTE	STRAIGHT AND LEVEL	DRY	NO DEFECTS	INJURY AND / OR TOW DUE TO CRASH	N

Figure 3.1: *The **traffic-accidents.csv**.*

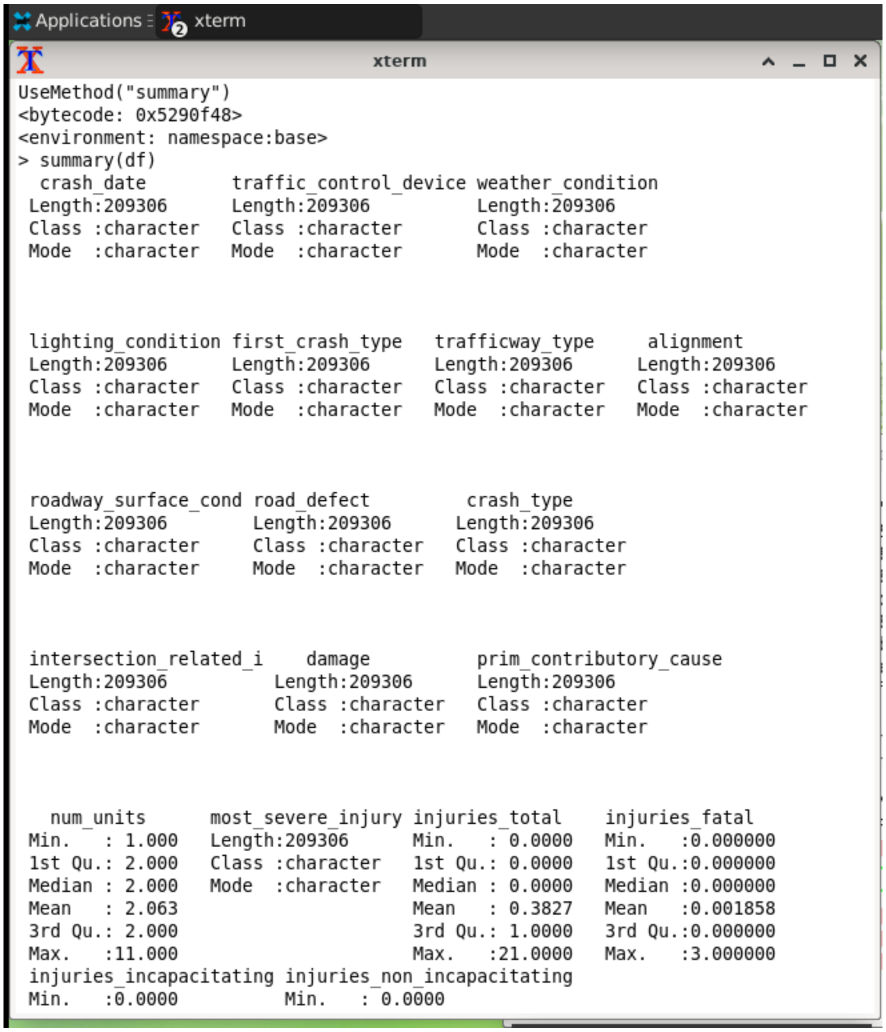
i. Statistics Summary

[R*] If we already have a nice and clean dataframe then we can create the summary, it is very easy and only need few lines of codes in R.

```
library(dplyr)
library(ggplot2)
library(scales)
library(stringr)
library(ggrepel)
library(lubridate)
library(ggthemes)
library(RColorBrewer)
library(data.table)

df <— fread("/root/R/CSV/traffic_accidents.csv")

summary(df)
```

R Code 9: *bar chart for car accident (ch3-caraccident-summary.R)*

You need to type **summary(df)** on the R console directly not only calling the R codes above by **source('..')**.




```
UseMethod("summary")
<bytecode: 0x5290f48>
<environment: namespace:base>
> summary(df)
  crash_date         traffic_control_device weather_condition
 Length:209306      Length:209306          Length:209306
 Class :character   Class :character       Class :character
 Mode  :character   Mode  :character       Mode  :character


 lighting_condition first_crash_type   trafficway_type     alignment
 Length:209306      Length:209306      Length:209306      Length:209306
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character


 roadway_surface_cond road_defect        crash_type
 Length:209306        Length:209306      Length:209306
 Class :character     Class :character   Class :character
 Mode  :character     Mode  :character   Mode  :character


 intersection_related_i    damage          prim_contributory_cause
 Length:209306          Length:209306      Length:209306
 Class :character       Class :character   Class :character
 Mode  :character       Mode  :character   Mode  :character


   num_units      most_severe_injury injuries_total    injuries_fatal
 Min.   : 1.000   Length:209306      Min.   : 0.0000   Min.   :0.000000
 1st Qu.: 2.000   Class :character   1st Qu.: 0.0000   1st Qu.:0.000000
 Median : 2.000   Mode  :character   Median : 0.0000   Median :0.000000
 Mean   : 2.063                      Mean   : 0.3827   Mean   :0.001858
 3rd Qu.: 2.000                      3rd Qu.: 1.0000   3rd Qu.:0.000000
 Max.   :11.000                      Max.   :21.0000   Max.   :3.000000
 injuries_incapacitating injuries_non_incapacitating
 Min.   :0.0000          Min.   : 0.0000
```

Figure 3.2: *The summary for the traffic accident data set.*

ii. Bar Chart

[R*] This bar chart counts the amount of accidents by crash type. The chart displays the top 4 types of crash: turning, angle, rear end, sideswipe/same direction, and then creates a fifth category where all other types of accidents are combined together. From this, it can be seen that accidents occur the most when a person is turning, then from an angle, then being rear ended, then all other categories, and then from a sideswipe. When driving, a person should be extra careful when they are turning, as this is when most accidents occur.

```
library(dplyr)
library(ggplot2)
library(scales)
library(stringr)
library(ggrepel)
library(lubridate)
```

```
library(ggthemes)
library(RColorBrewer)
library(data.table)

df <- fread("/root/R/CSV/traffic_accidents.csv")

crashcount <- data.frame(count(df, first_crash_type))
crashcount <- crashcount[order(crashcount$n, decreasing = TRUE
    ),]

most_common <- crashcount[1:4,]

other <- crashcount[5:18,]

other_sum <- sum(other$n)

otherdf <- data.frame(first_crash_type = 'OTHER', n =
    other_sum)

top5 <- rbind(most_common, otherdf)
top5 <- top5[order(top5$n, decreasing = TRUE),]


top5$first_crash_type <- str_to_title(top5$first_crash_type)

p <- ggplot(top5, aes(x= reorder(first_crash_type, -n), y = n)
    ) +
geom_bar(colour = 'black', fill = 'navyblue', stat = 'identity
    ') +
labs(title = "Amount of Accidents by Crash Type", x = "Type of
    Crash", y = "Accident Count") +
geom_text(aes(label= comma(n)), vjust = -.5, size = 3) +
theme(plot.title = element_text(hjust=0.5)) +
theme_gray()+
scale_y_continuous(label = comma)

print(p)
```

R Code 10: *bar chart for car accident (ch3-caraccident-barchart.R)*

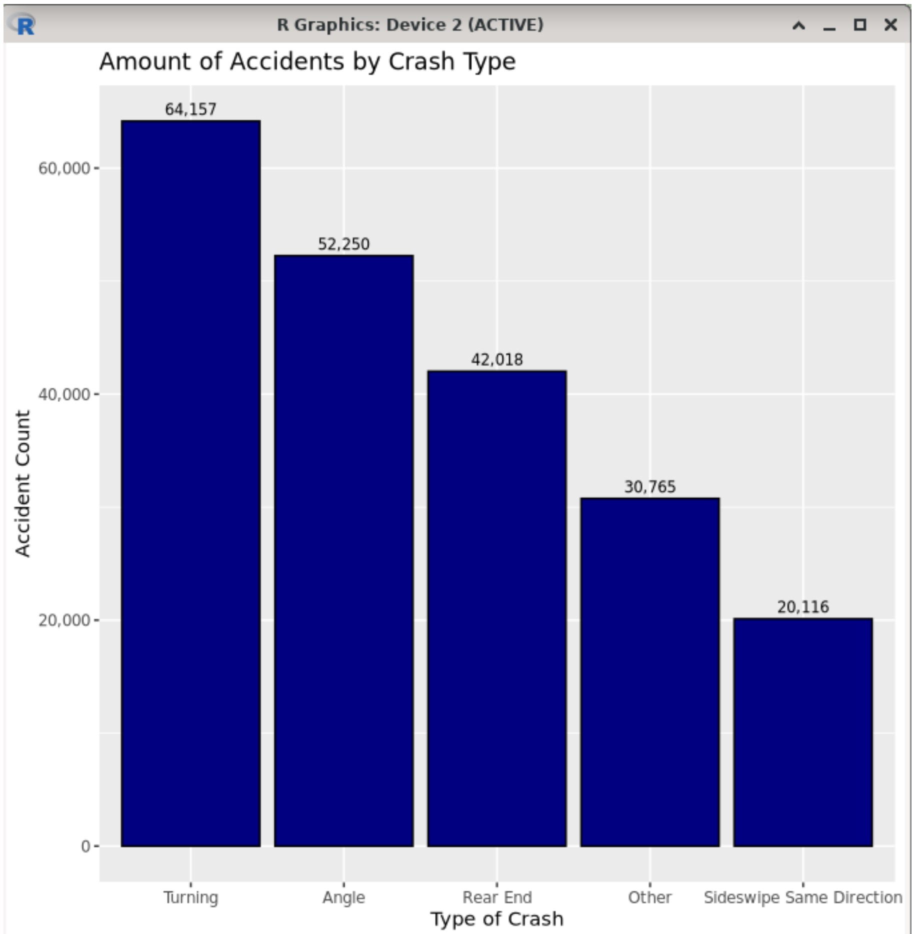


Figure 3.3: *The bar chart that shows the amount of accidents by crash type.*

Figure 3.4: *An accident caused by turning.*

Figure 3.5: *Illustration for sideswipe accidents.*

[R*] We are highlighting the codes to count each qualitative variable under the **first_crash_type**' column. In the next chapter we will go deeper on the qualitative and quantitative variables.

```
...

crashcount <- data.frame(count(df, first_crash_type))
crashcount <- crashcount[order(crashcount$n, decreasing = TRUE
    ),]

...
```

Standard	Standard	Standard
lighting_condition	first_crash_type	trafficway_type
DAYLIGHT	TURNING	NOT DIVIDED
DARKNESS, LIGHTED ROAD	TURNING	FOUR WAY
DAYLIGHT	REAR END	T-INTERSECTION
DAYLIGHT	ANGLE	FOUR WAY
DAYLIGHT	REAR END	T-INTERSECTION
DARKNESS, LIGHTED ROAD	FIXED OBJECT	NOT DIVIDED
DAYLIGHT	REAR TO FRONT	FOUR WAY
DAYLIGHT	ANGLE	DIVIDED - W/MEDIAN (NOT RAISED)
DAYLIGHT	REAR END	NOT DIVIDED
DAYLIGHT	ANGLE	FOUR WAY
DARKNESS, LIGHTED ROAD	FIXED OBJECT	NOT DIVIDED
DUSK	ANGLE	OTHER
DAYLIGHT	TURNING	UNKNOWN INTERSECTION TYPE
DAYLIGHT	SIDESWIPE SAME DIRECTION	ONE-WAY
DAYLIGHT	REAR END	RAMP
DARKNESS, LIGHTED ROAD	SIDESWIPE OPPOSITE DIRECTION	NOT DIVIDED
DAYLIGHT	PEDALCYCLIST	TRAFFIC ROUTE
DUSK	ANGLE	FIVE POINT, OR MORE
DAYLIGHT	SIDESWIPE SAME DIRECTION	FOUR WAY
DAYLIGHT	TURNING	ONE-WAY
DAYLIGHT	REAR END	NOT DIVIDED
DARKNESS, LIGHTED ROAD	TURNING	FOUR WAY
DAYLIGHT	TURNING	T-INTERSECTION
DARKNESS	TURNING	ONE-WAY
DAYLIGHT	REAR END	T-INTERSECTION

Figure 3.6: *The data under the **first_crash_type**' column is a qualitative variables, it is not numerical / quantitative variables, so we will need to count how many qualitative variables that is under this column.*

iii. Line Plot

[R*] This line plot sums up all of the injuries from each month and plots the sum on the graph. The most amount of injuries and the least amount of injuries have black circles that mark them. The least amount of injuries from accidents is in the month of February at 4,639, while the most injuries come from the month of October with 7,918 total injuries occuring in that month over the 12 years that data has been collected. I was surprised to see that the least amount of accidents occur in January, February, and March, and December and November are also in the bottom half for amount of injuries from accidents, because I would have expected that the winter months and months with snow would have the most injuries resulting from accidents because I would believe that snow would cause more accidents and more accidents that result in injuries. The summer and fall months have more car crashes that cause injuries, which may result from more people travelling and being on the road during these months.

```
library(dplyr)
library(ggplot2)
library(scales)
library(stringr)
library(ggrepel)
library(lubridate)
library(ggthemes)
library(RColorBrewer)
library(data.table)

df <- fread("/root/R/CSV/traffic_accidents.csv")

month_injuries <- df %>%
group_by(crash_month) %>%
```

```
summarise(total_injuries = sum(injuries_total, na.rm = TRUE))
    %>%
data.frame()

month_injuries$crash_month <- as.factor(
    month_injuries$crash_month)

high_low <- month_injuries %>%
filter(total_injuries == min(total_injuries) | total_injuries
    == max(total_injuries)) %>%
data.frame()

p <- ggplot(month_injuries, aes(x = crash_month, y =
    total_injuries, group=1)) +
geom_line(color = 'navyblue', linewidth =1) +
geom_point(shape = 21, size =3, color = 'royalblue2', fill = '
    royalblue2') +
labs(x = 'Month', y = 'Total Injuries', title = 'Injuries from
    Crashes by Month')+
scale_y_continuous(labels = comma) +
theme_bw()+
theme(plot.title = element_text(hjust = 0.5)) +
geom_point(data = high_low, aes(x=crash_month, y=total_injuries
    ), inherit.aes = FALSE,
shape = 21, size = 3, fill = 'black', color = 'black') +
geom_label_repel(aes(label = ifelse(total_injuries == max(
    total_injuries)
| total_injuries == min(total_injuries),
scales::comma(total_injuries), '')),
box.padding = 1, point.padding =0, size = 3, nudge_x = .5,
color = 'black', segment.color = 'gray') +
scale_x_discrete(breaks = 1:12, labels = month.abb)

print(p)
```

R Code 11: *bar chart for car accident (ch3-caraccident-lineplot.R)*

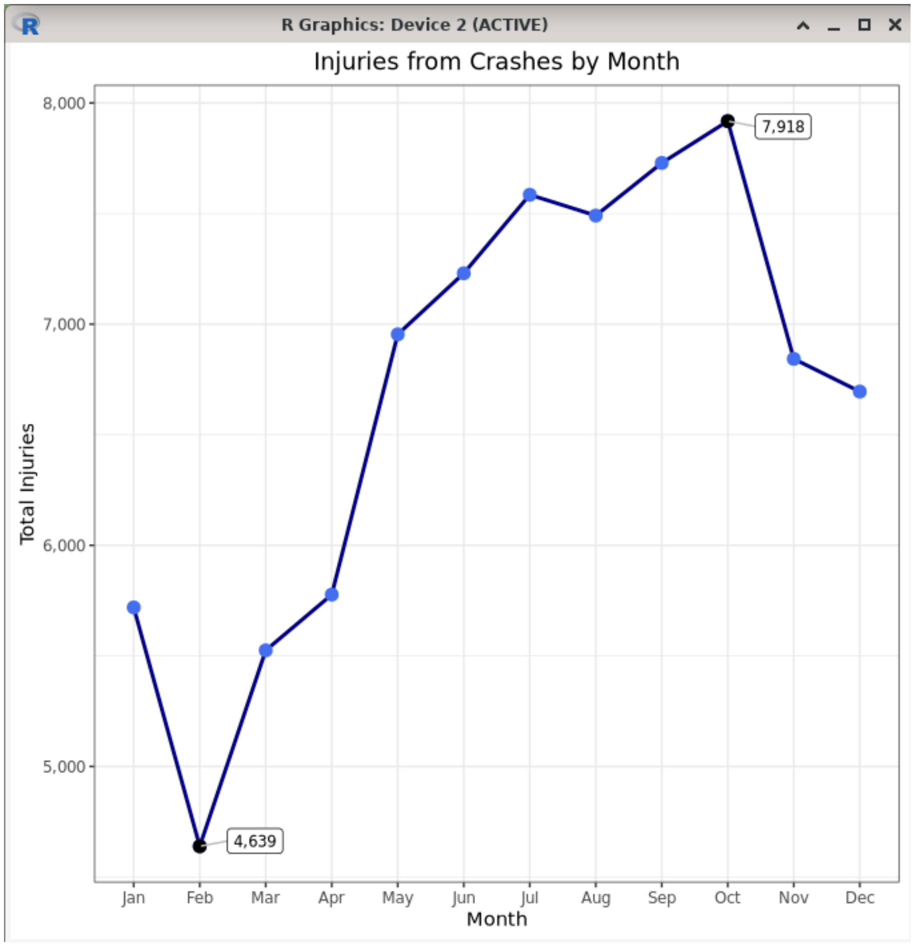


Figure 3.7: *The line plot.*

iv. Heatmap

[R*] The Heatmap shows the average amount of injuries sustained in a car crash by the cost of the damage and by the weather condition. It can be seen that the average amount of injuries sustained is the highest during fog/smoke/haze when the damage is USD 500 or less and during sleet/hail when the damage is USD 500 or less. The smallest average of injuries are sustained when the crash has damage between USD 501 and USD 1,500 in any weather condition.

The highest costs of damage would result in the highest average amount of injuries because the more damage to the car would mean a more dangerous crash. The weather events that cause the most injuries are cloudy/overcast, fog/smoke/haze, freezing rain/drizzle, other, and sleet/hail. This makes sense because most of these make it difficult to see or make the roads slippery.

```
library(dplyr)
library(ggplot2)
library(scales)
```

```
library(stringr)
library(ggrepel)
library(lubridate)
library(ggthemes)
library(RColorBrewer)
library(data.table)

df <- fread("/root/R/CSV/traffic_accidents.csv")

df2 <- df %>%
group_by(weather_condition, damage) %>%
summarise(total_injuries = mean(injuries_total, na.rm = TRUE),
    .groups = "drop")

df2 <- df2 %>%
mutate(weather_condition = reorder(weather_condition,
    total_injuries, .desc = TRUE))

df2$damage <- str_to_title(df2$damage)
df2$weather_condition <- str_to_title(df2$weather_condition)

p <- ggplot(df2, aes(x = weather_condition, y = damage, fill =
     total_injuries)) +
geom_tile(color = "black") +
geom_text(aes(label = round(total_injuries, 1)), color = "black
    ", size = 3)+
coord_equal(ratio=2) +
labs(title = "Heatmap of Average Amount of Injuries by Damage &
     Weather Condition",
x = "Weather Condition",
y = "Cost of Damage",
fill = "Average Amount of Injuries")+
theme_minimal()+
theme(
plot.title = element_text(hjust = 0.5),
axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1,
    size = 8)) +
scale_fill_distiller(palette = "Blues", direction = 1)

print(p)
```

R Code 12: *bar chart for car accident (ch3-caraccident-heatmap.R)*

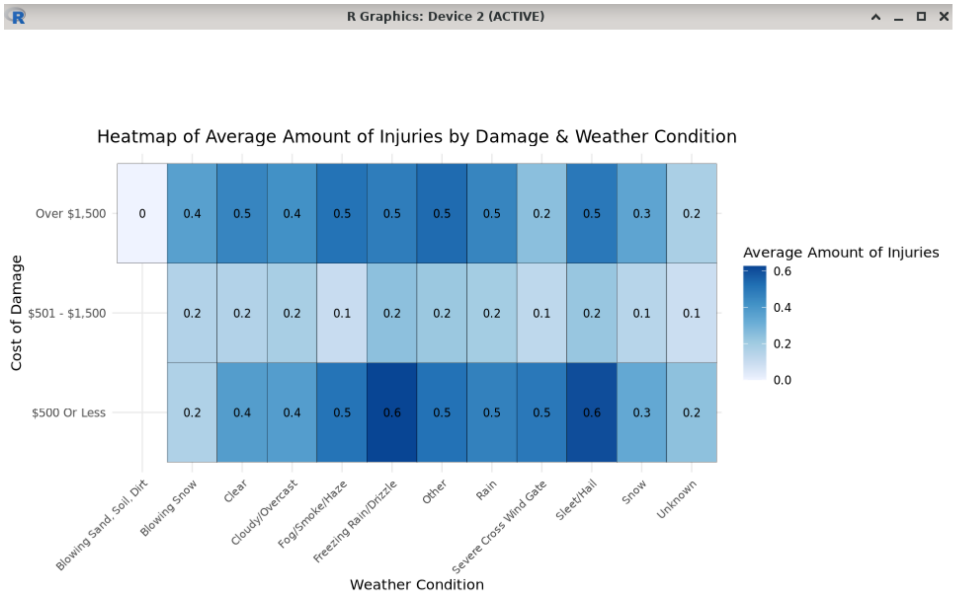


Figure 3.8: *The heatmap.*

v. Pie Chart

[R*] The type of roads (trafficway) that have the most crashes on them are divided road with and without barriers, fourways, not divided, and one ways. These trafficways are separated by the three most recent years on the pie charts. Each slice represents the percentage of crashes that involved that trafficway type. It is made evident that across all three years, the most amount of accidents occur on four way traffic ways. One Way and divided with median barrier have the least amount of crashes out of the top 6 trafficways. The percentages remain fairly consistent across all three years, with most traffic way type percentages varying by less than 1%.

```
library(dplyr)
library(ggplot2)
library(scales)
library(stringr)
library(ggrepel)
library(lubridate)
library(ggthemes)
library(RColorBrewer)
library(data.table)

df$trafficway_type <- str_to_title(df$trafficway_type)
toptt <- count(df, trafficway_type)
toptt <- toptt[order(-toptt$n),]
#toptt[toptt$trafficway_type %in% c("Not Divided", "Four Way", "
    Divided - W/Median (Not Raised)", "One-Way", "Divided - W
```

```
    /Median Barrier", "T-Intersection"), "n"] / sum(toptt$n)

df3 <- df %>%
select(trafficway_type, crash_date) %>%
mutate(year = year(mdy_hms(crash_date)),
toptrafficway = ifelse(trafficway_type == "Not Divided", "Not
    Divided", ifelse(trafficway_type=="Four Way", "Four Way",
    ifelse(trafficway_type=="Divided - W/Median (Not Raised)",
     "Divided - W/Median (Not Raised)", ifelse(trafficway_type
    =="One-Way", "One-Way", ifelse(trafficway_type=="Divided
    - W/Median Barrier", "Divided - W/Median Barrier", ifelse
    (trafficway_type=="One-Way", "One-Way", "Other")))))))
    %>%
group_by(year, toptrafficway) %>%
summarise(n=length(toptrafficway), .groups = 'keep') %>%
group_by(year) %>%
mutate(percent_of_total = round(100*n/sum(n), 1)) %>%
ungroup() %>%
data.frame()

df3 <- subset(df3, year >= max(df3$year-2))

p <- ggplot(data = df3, aes(x="", y=n, fill=toptrafficway)) +
geom_bar(stat="identity", position="fill") +
coord_polar(theta="y", start=0) +
labs(fill="Traffic Way Type", x=NULL, y=NULL, title="Traffic
    Way Type Count by Year", caption = "Slices under 4% are not
     labeled") +
theme_minimal()+
theme(plot.title=element_text(hjust=0.5),
axis.text=element_blank(),
axis.ticks=element_blank(),
panel.grid=element_blank()) +
facet_wrap(~year, ncol = 3, nrow = 1) +
scale_fill_brewer(palette = "Blues")+
geom_text(aes(x=1.9, label=ifelse(percent_of_total>4, paste0(
    percent_of_total, "%"), " ")), size = 3, position=
    position_fill(vjust=0.5))

print(p)
```

R Code 13: *bar chart for car accident (ch3-caraccident-piechart.R)*

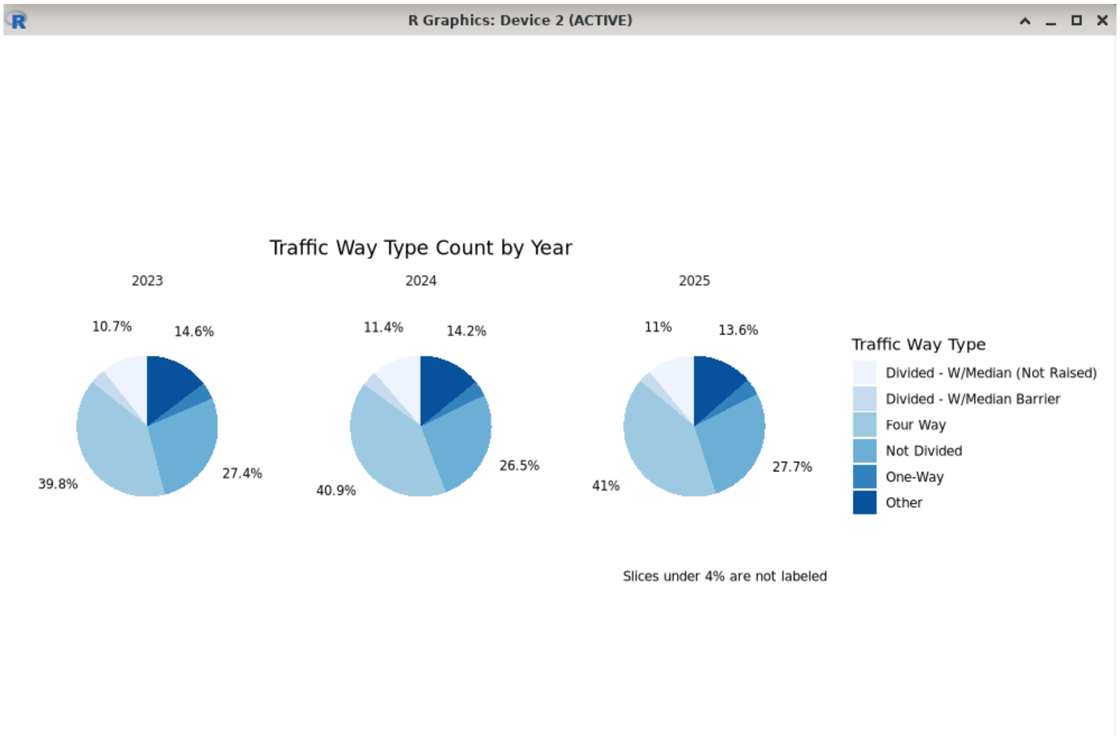


Figure 3.9: *The pie chart.*

vi. Stacked Bar Chart

[R*] The stacked bar chart displays the top five primary contributory causes to the crash and shows how many vehicles have been involved in these crashes. The amount of vehicles involved is different than total amount of crashes because one accident can have more than one car involved. The number of cars involved is divided by the cost of the damage from the accident. There are the most cars involved in accidents with damage that is greater than USD 1,500, which makes sense because there would be damage to multiple cars that would need to be paid for, whereas a crash with one car would only need to cover the damage of that one car.

```
library(dplyr)
library(ggplot2)
library(scales)
library(stringr)
library(ggrepel)
library(lubridate)
library(ggthemes)
library(RColorBrewer)
library(data.table)

df$prim_contributory_cause <- str_to_title(
    df$prim_contributory_cause)
```

```
topcontrib <- df %>%
count(prim_contributory_cause, sort = TRUE) %>%
slice_head(n = 5)

df5 <- df %>%
filter(prim_contributory_cause %in%
    topcontrib$prim_contributory_cause) %>%
group_by(damage, prim_contributory_cause) %>%
summarise(totalcars = sum(num_units), .groups = "drop") %>%
data.frame()

df5$damage <- as.factor(df5$damage)

p <- ggplot(df5, aes(x = reorder(prim_contributory_cause,
    totalcars), y = totalcars, fill = damage)) +
geom_bar(stat = "identity", position = position_stack(reverse =
     TRUE)) +
labs(title = "Cars Involved in Top 5 Primary Contributory
    Causes by Damage",
x = " ",
y = "Amount of Cars Involved in Accidents",
fill = "Cost of Damage") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
scale_y_continuous(labels = comma,
breaks = seq(0, 120000, by = 20000)) +
geom_text(aes(label = comma(totalcars)),
stat = "identity",
position = position_stack(vjust = 0.5, reverse = TRUE),
size = 3, color = "black", angle = 0)+
scale_fill_brewer(palette = "Blues")

print(p)
```

R Code 14: *bar chart for car accident (ch3-caraccident-stackedbarchart.R)*

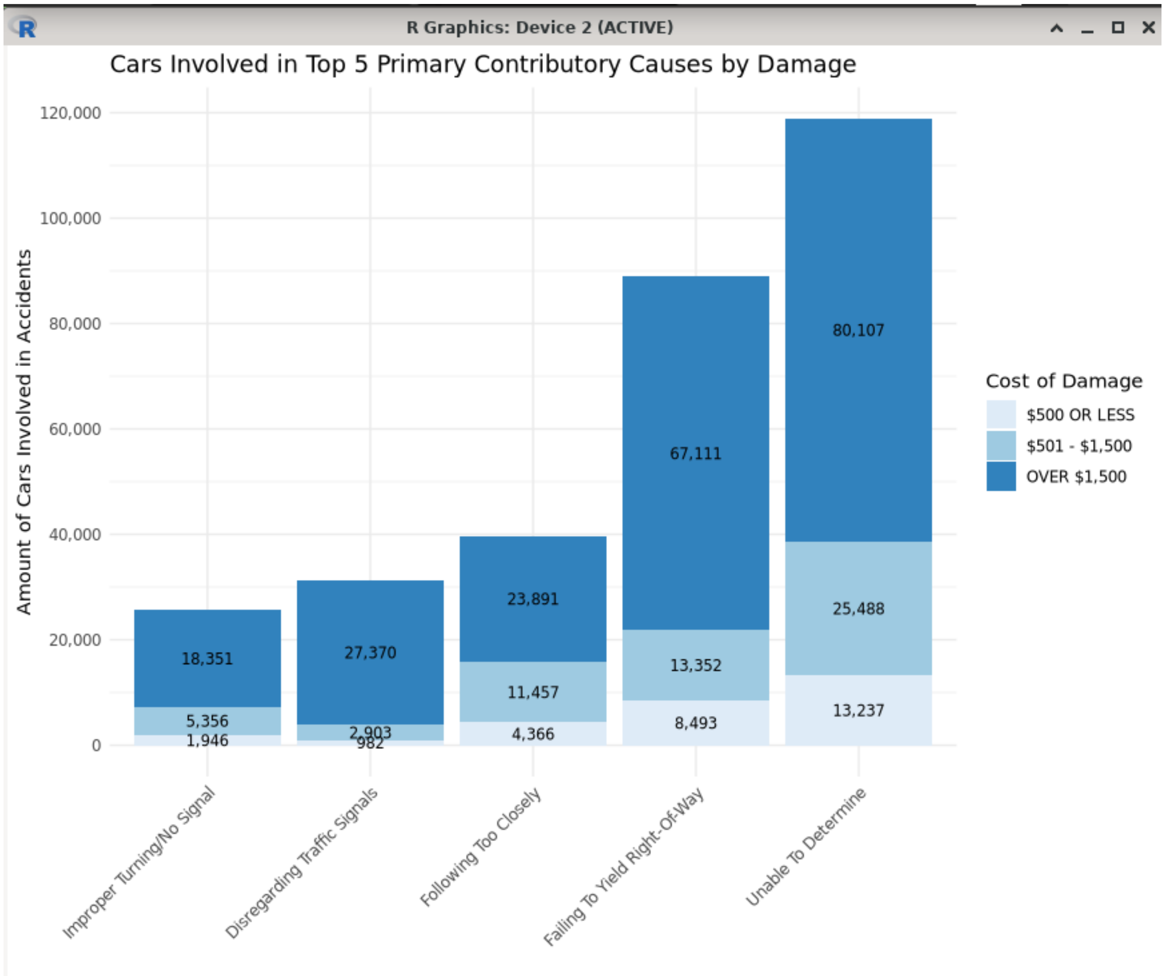


Figure 3.10: *The stacked bar chart.*

In conclusion, the most accidents come from a person who is turning, however this does not result in the most injuries, as most injuries occur with an angled crash. These angled crashes have the largest percentages of injuries at 36.8% on Mondays. The least amount of injuries occur in crashes in February and the most occur in October. When weather conditions affect visibility or make the road slippery, the average amount of injuries from the accident is the highest. If there is less than USD 500 worth of damage in the crash, the average amount of injuries sustained is higher than the higher costing damages. Out of all of the traffic ways, most accidents occur on four ways.

Chapter 4

Correlation Tests

All the girls in the world were divided into two classes: one class included all the girls in the world except her, and they had all the usual human feelings and were very ordinary girls; while the other class -herself alone- had no weaknesses and was superior to all humanity. - ***Leo Tolstoy***

BEfore we can learn to be a better data scientist or data analyst, we need to understand the type of data that we have.

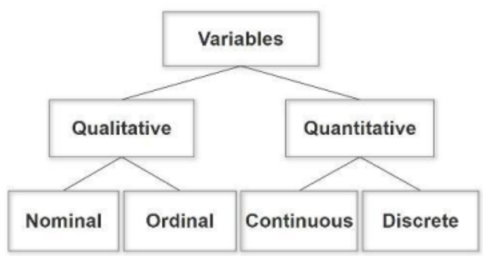


Figure 4.1: *The type of variable.*

The two basic variable categories are the qualitative and the quantitative. Qualitative variables refer to variables, like gender, level of education, location, etc. They are divided in nominal and ordinal (or tactical). Nominal variables represent categories, of which the order does not matter like color.

Conversely, ordinal or tactical variables represent categories, of which the order does matters, i.e., disease severity.

Quantitative variables are numerical values, expressed in a unit of measure i.e., age. They are divided in discrete and continuous variables. Depending on the unit of measure, data can be characterized as categorical.

To measure how variables correlate with each other we use correlation coefficient as the measure.

Correlation coefficients are used to measure how strong a relationship is between two variables. There are several types of correlation coefficient, but the most popular is Pearson. Pearson correlation (also called Pearson's R) is a correlation coefficient commonly used in linear regression, and it is used for numerical data. In fact, when anyone refers to the correlation coefficient, they are usually talking about Pearson.

A correlation coefficient is a measure of the strength of a linear relationship between two variables. In general, correlation coefficient values range from -1 to 1:

1. Correlation coefficent of 1 = a strong positive linear relationship. This means that for every positive increase in one variable, there is a proportional positive increase in the other variable. For instance, fuel consumption increase almost perfectly in correlation with the miles taken by the vehicle.

2. Correlation coefficent of -1 = a strong negative linear relationship. In other words, for every positive increase in one variable, there is a proportional negative decrease in the other variable. As an example, the amount of gas in a vehicle's tank decreases almost perfectly in correlation with speed.

3. Correlation coefficent of 0 = there is no linear relationship between the variables.

I. Correlation between Numerical Variables Case Study: Economic Data

We know that correlation means the relationship between two variables: for example between GDP and literacy in a country

We re going to use this dataset: **countries_of_the_world.csv**, it is available in the repository (**https://github.com/glanzkaiser/GFreya-R-for-Statistics/CSV**)

i. Pearson Correlation Coefficient

Pearson correlation shows the linear relationship between two sets of data. In simple terms, it answers the question, Can I draw a line graph to represent the data? Two letters are used to represent the Pearson correlation: Greek letter rho ρ for a population and the letter r for a sample. The Pearson correlation is not able to tell the difference between dependent variables and independent variables.

This is one of the most commonly used formulas is Pearson correlation coefficient formula

$$r_{pearson} = \frac{n\left(\sum_{i=1}^{n} xy\right) - \left(\sum_{i=1}^{n} x\right)\left(\sum_{i=1}^{n} y\right)}{\sqrt{\left[\left(n\sum_{i=1}^{n} x^2\right) - \left(\sum_{i=1}^{n} x\right)^2\right]\left[\left(n\sum_{i=1}^{n} y^2\right) - \left(\sum_{i=1}^{n} y\right)^2\right]}} \tag{4.1}$$

There are another two formulas that are commonly used, the first one is the sample correlation coefficient:

$$r_{xy} = \frac{s_{xy}}{s_x s_y} \tag{4.2}$$

with s_x and s_y are the sample standard deviations, and s_{xy} is the sample covariance.

The second one is the population correlation coefficient:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{4.3}$$

the population correlation coefficient uses σ_x and σ_y as the population standard deviations, and σ_{xy} as the population covariance.

II. Correlation between Categorical Variables Case Study: USA Crime Data

If we use Pearson Correlation Coefficient to calculate the correlation between continuous numerical variables (quantitative variables), then for the categorical variables (qualitative variables) we can use some of these tests:

1. Tetrachoric Correlation
   Used to calculate the correlation between binary categorical variables, binary variables are variables that can only take on one of two possible values.
   The value for tetrachoric correlation ranges from -1 to 1 where -1 indicates a strong negative correlation, 0 indicates no correlation, and 1 indicates a strong positive correlation.

2. Polychoric Correlation
   Used to calculate the correlation between ordinal categorical variables, ordinal variables are variables whose possible values have a natural order.
   The value for polychoric correlation ranges from -1 to 1 where -1 indicates a strong negative correlation, 0 indicates no correlation, and 1 indicates a strong positive correlation.

3. Cramer's V
   Used to calculate the correlation between nominal categorical variables, nominal variables are ones that take on category labels but have no natural ordering.
   The value for Cramer's V ranges from 0 to 1, with 0 indicating no association between the variables and 1 indicating a strong association between the variables.

We are going to use the US crime Record from 1980 data with 638454 records and 24 Columns of record that we obtain from:
https://www.kaggle.com/datasets/mrayushagrawal/us-crime-dataset

You can download and save it in CSV format or anything you can work on.

i. Barchart and Histogram for Categorical Variables Case Study: USA Crime Data

[R*] We will start with making few bar chart to rank the race of the perpetrator, gender of the perpetrator, gender of the victim, and the relationship between the victim and perpetrator.

You should read and study the CSV first so you know what to do, what codes to write in R. **If there is a column title with space bar, don't forget to rename the column title at the CSV without space bar, because codes in R cannot read space bar.**

```
library(dplyr)
library(ggplot2)

df <- fread("/root/R/CSV/usa_crimes.csv")
summary(df)


head(df)
```

```
selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

head(selected_df)

perpetratorracecount <- data.frame(count(selected_df,
    Perpetrator_Race))
perpetratorracecount <- perpetratorracecount[order(
    perpetratorracecount$n, decreasing = TRUE),]

most_common <- perpetratorracecount[1:4,]

other <- perpetratorracecount[5:18,]

other_sum <- sum(other$n)

otherdf <- data.frame(Perpetrator_Race = 'OTHER', n =
    other_sum)

top5 <- rbind(most_common, otherdf)
top5 <- top5[order(top5$n, decreasing = TRUE),]


top5$Perpetrator_Race <- str_to_title(top5$Perpetrator_Race)

p <- ggplot(top5, aes(x= reorder(Perpetrator_Race, -n), y = n)
    ) +
geom_bar(colour = 'black', fill = 'navyblue', stat = 'identity
    ') +
labs(title = "Crime Commited in USA by Race", x = "Perpetrator
    Race", y = "Number of Crimes") +
geom_text(aes(label= comma(n)), vjust = -.5, size = 3) +
theme(plot.title = element_text(hjust=0.5)) +
theme_gray()+
scale_y_continuous(label = comma)

print(p)
```

R Code 15: *bar chart for usa crimes perpetrator race (ch4-usacrimes-barchart_perpetratorrace.R)*

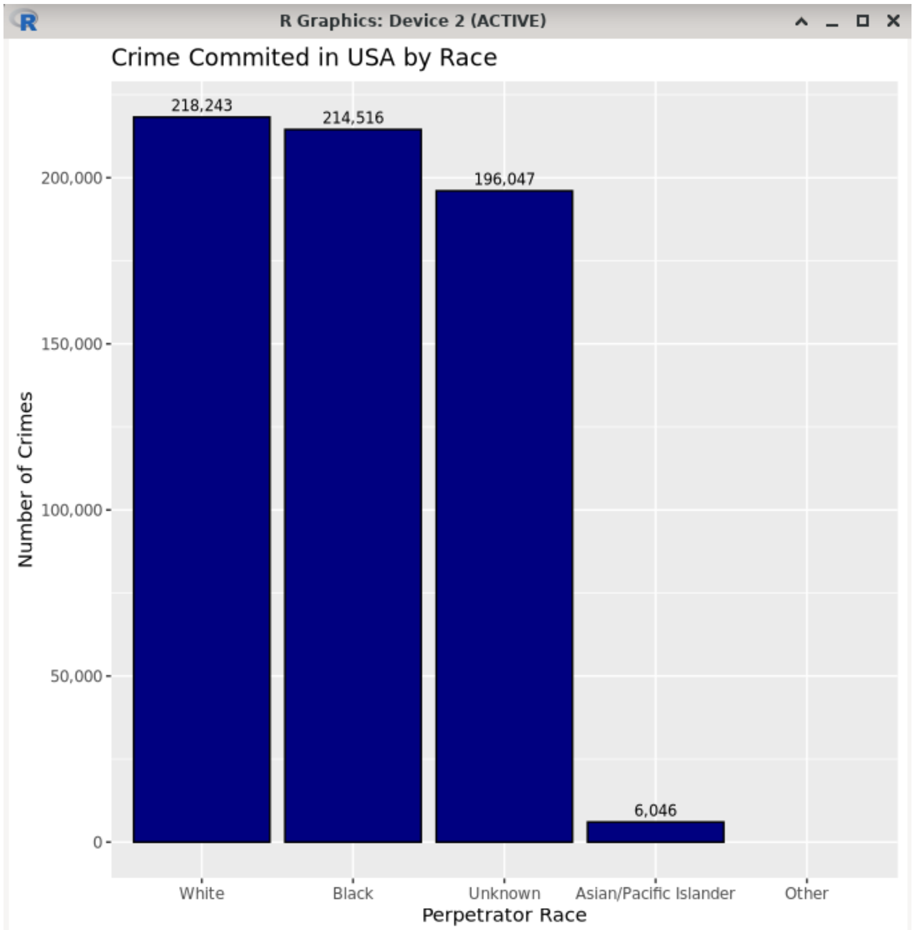


Figure 4.2: *The crime committed in USA ranked by the race of the perpetrator.*

```
library(dplyr)
library(ggplot2)

df <— fread("/root/R/CSV/usa_crimes.csv")
summary(df)


head(df)
selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

head(selected_df)

perpetratorsexcount <— data.frame(count(selected_df,
```

```
    Perpetrator_Sex))
perpetratorsexcount <- perpetratorsexcount[order(
    perpetratorsexcount$n, decreasing = TRUE),]

most_common <- perpetratorsexcount[1:3,]

other <- perpetratorsexcount[4:18,]

other_sum <- sum(other$n)

otherdf <- data.frame(Perpetrator_Sex = 'OTHER', n = other_sum
    )

top5 <- rbind(most_common, otherdf)
top5 <- top5[order(top5$n, decreasing = TRUE),]


top5$Perpetrator_Sex <- str_to_title(top5$Perpetrator_Sex)

p <- ggplot(top5, aes(x= reorder(Perpetrator_Sex, -n), y = n))
     +
geom_bar(colour = 'black', fill = 'navyblue', stat = 'identity
    ') +
labs(title = "Crime Commited in USA by Gender", x = "
    Perpetrator Sex", y = "Number of Crimes") +
geom_text(aes(label= comma(n)), vjust = -.5, size = 3) +
theme(plot.title = element_text(hjust=0.5)) +
theme_gray()+
scale_y_continuous(label = comma)

print(p)
```

R Code 16: *bar chart for usa crimes perpetrator sex (ch4-usacrimes-barchart_perpetratorsex.R)*

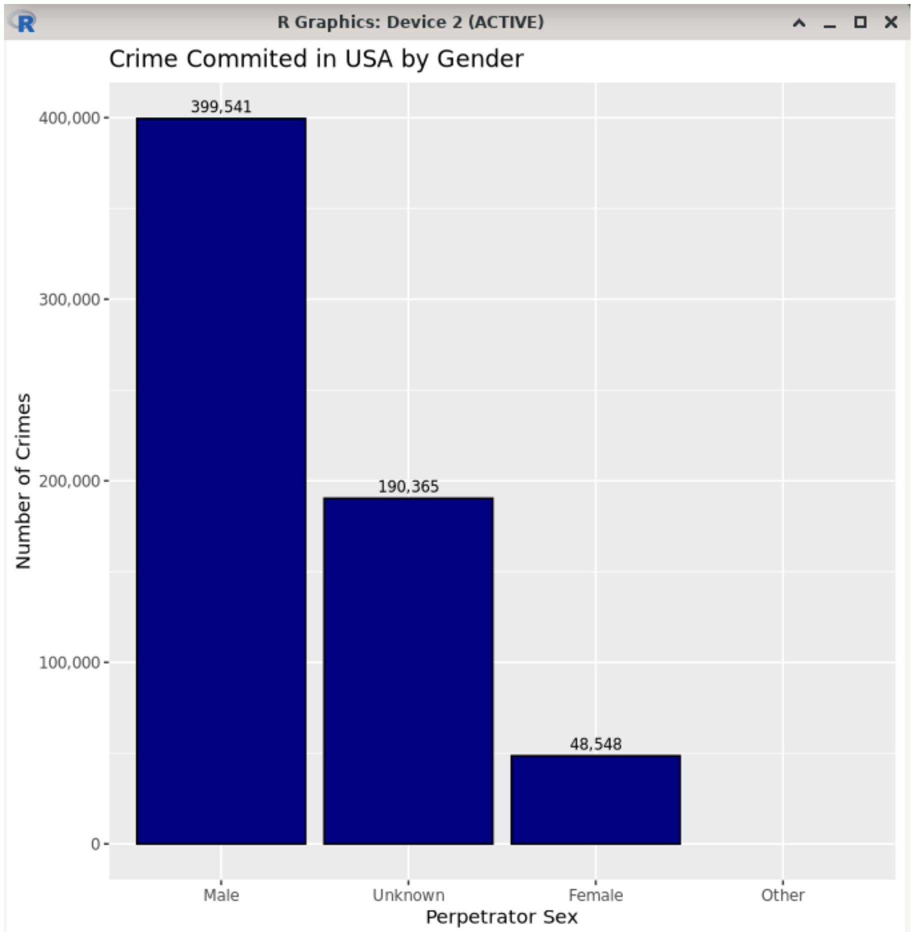


Figure 4.3: *The crime committed in USA ranked by the gender of the perpetrator.*

```
library(dplyr)
library(ggplot2)

df <— fread("/root/R/CSV/usa_crimes.csv")
summary(df)


head(df)
selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

head(selected_df)

victimsexcount <— data.frame(count(selected_df, Victim_Sex))
```

```
victimsexcount <- victimsexcount[order(victimsexcount$n,
    decreasing = TRUE),]

most_common <- victimsexcount[1:3,]

other <- victimsexcount[4:18,]

other_sum <- sum(other$n)

otherdf <- data.frame(Victim_Sex = 'OTHER', n = other_sum)

top5 <- rbind(most_common, otherdf)
top5 <- top5[order(top5$n, decreasing = TRUE),]


top5$Victim_Sex <- str_to_title(top5$Victim_Sex)

p <- ggplot(top5, aes(x= reorder(Victim_Sex, -n), y = n)) +
geom_bar(colour = 'black', fill = 'navyblue', stat = 'identity
    ') +
labs(title = "Crime Commited in USA by Gender", x = "Victim Sex
    ", y = "Number of Crimes") +
geom_text(aes(label= comma(n)), vjust = -.5, size = 3) +
theme(plot.title = element_text(hjust=0.5)) +
theme_gray()+
scale_y_continuous(label = comma)

print(p)
```

R Code 17: *bar chart for usa crimes victim sex (ch4-usacrimes-barchart_victimsex.R)*

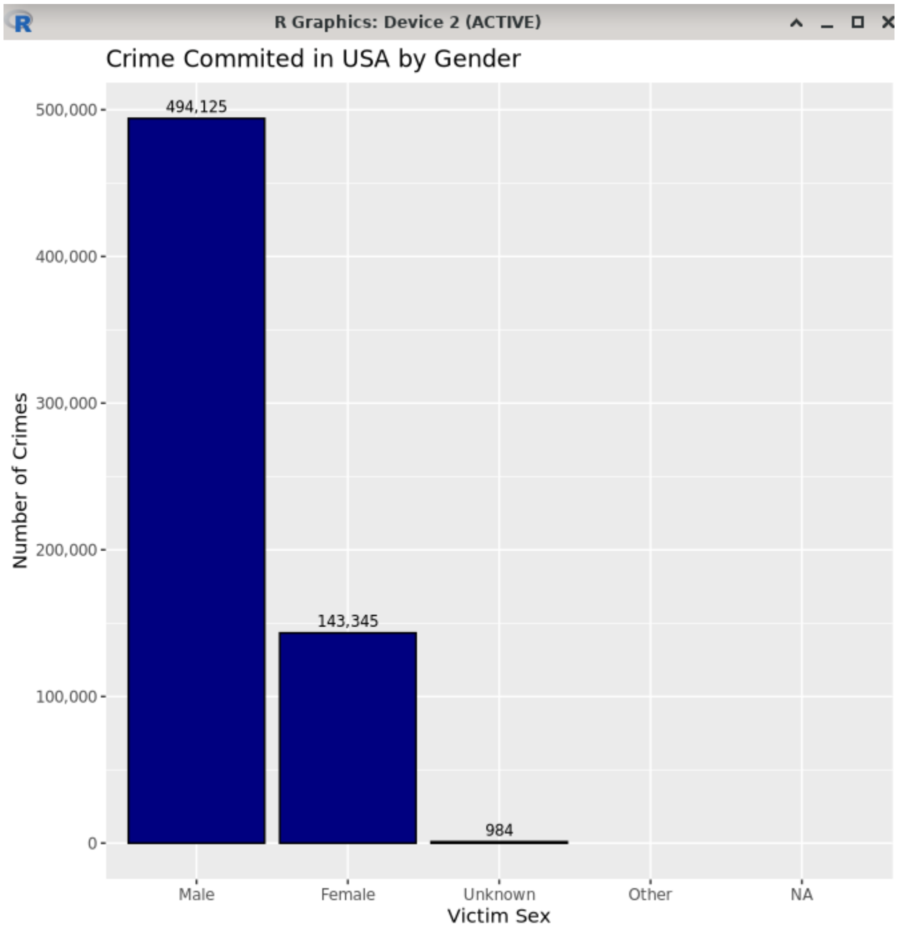


Figure 4.4: *The crime committed in USA ranked by the gender of the victim.*

```
library(dplyr)
library(ggplot2)

df <- fread("/root/R/CSV/usa_crimes.csv")
summary(df)


head(df)
selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

head(selected_df)

relationshipcount <- data.frame(count(selected_df,
```

```
    Relationship))
relationshipcount <- relationshipcount[order(
    relationshipcount$n, decreasing = TRUE),]

most_common <- relationshipcount[1:4,]

other <- relationshipcount[5:18,]

other_sum <- sum(other$n)

otherdf <- data.frame(Relationship = 'OTHER', n = other_sum)

top5 <- rbind(most_common, otherdf)
top5 <- top5[order(top5$n, decreasing = TRUE),]


top5$Relationship <- str_to_title(top5$Relationship)

p <- ggplot(top5, aes(x= reorder(Relationship, -n), y = n)) +
geom_bar(colour = 'black', fill = 'navyblue', stat = 'identity
    ') +
labs(title = "Crime Commited in USA by Relationship", x = "
    Relationship", y = "Number of Crimes") +
geom_text(aes(label= comma(n)), vjust = -.5, size = 3) +
theme(plot.title = element_text(hjust=0.5)) +
theme_gray()+
scale_y_continuous(label = comma)

print(p)
```

R Code 18: *bar chart for usa crimes relationship (ch4-usacrimes-barchart_relationship.R)*

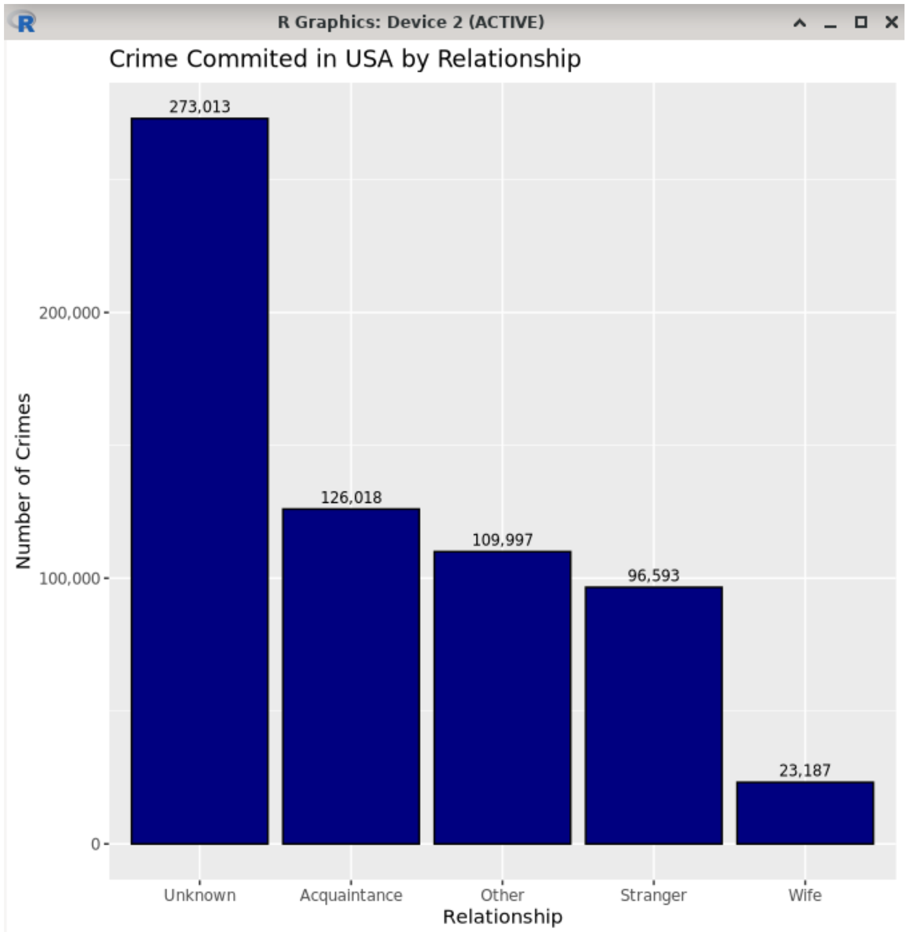


Figure 4.5: *The crime committed in USA ranked by the relationship between victim and the perpetrator.*

[R*] Now, for the histogram, it is a very useful visualization for showing the victim age or the perpetrator age.

```
library(dplyr)
library(ggplot2)

df <- fread("/root/R/CSV/usa_crimes.csv")

selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

p <- ggplot(selected_df, aes(x = Victim_Age)) +
geom_histogram(fill = "navyblue", color = "white", binwidth =
    5) +
labs(title = "Crime Commited in USA", subtitle = "binwidth = 5"
```

```
    , x = "Age of Victim") +
scale_x_discrete(drop=FALSE) + xlim(c(0, 100))


print(p)
```

R Code 19: *histogram for usa crimes victim age (ch4-usacrimes-histogram_victimage.R)*

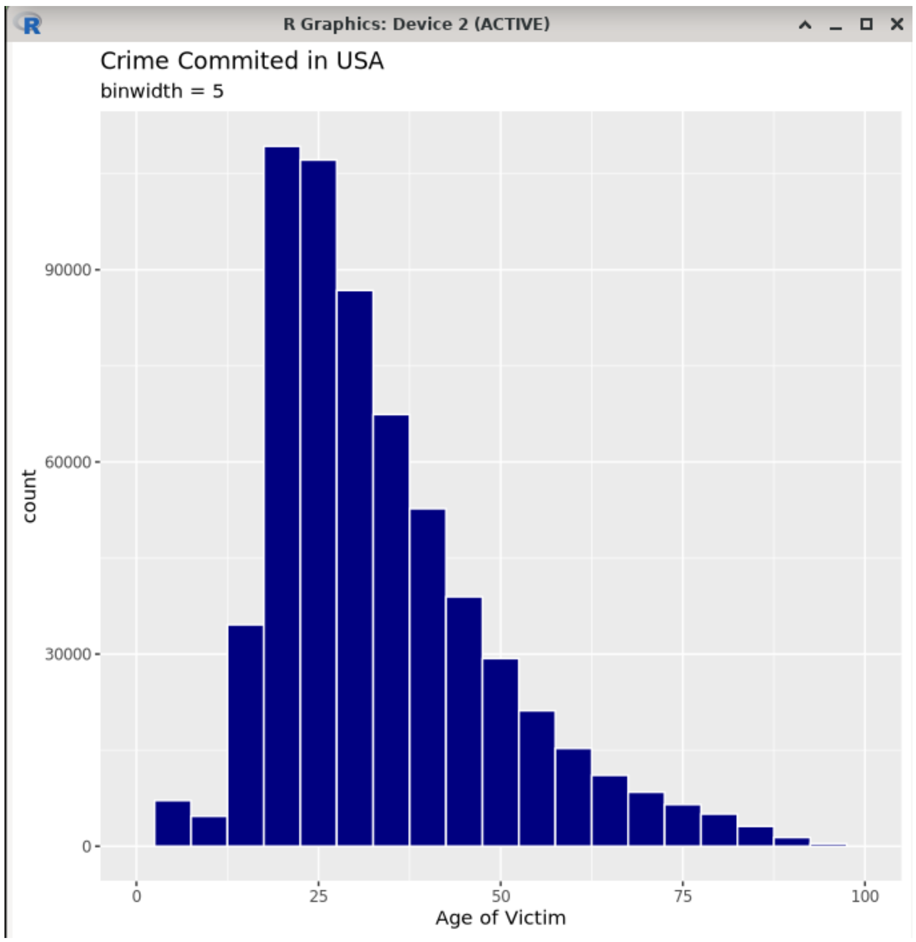


Figure 4.6: *The histogram is showing the age of the victim for crime committed in USA.*

```
library(dplyr)
library(ggplot2)

df <- fread("/root/R/CSV/usa_crimes.csv")

selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
```

```
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

p <- ggplot(selected_df, aes(x = Perpetrator_Age)) +
geom_histogram(fill = "navyblue", color = "white", binwidth =
    5) +
labs(title = "Crime Commited in USA", subtitle = "binwidth = 5"
    , x = "Age of Perpetrator") +
scale_x_discrete(drop=FALSE) +
xlim(c(0, 100)) +
ylim(c(0, 100000))

print(p)
```

R Code 20: *histogram for usa crimes perpetrator age (ch4-usacrimes-histogram_perpetratorage.R)*

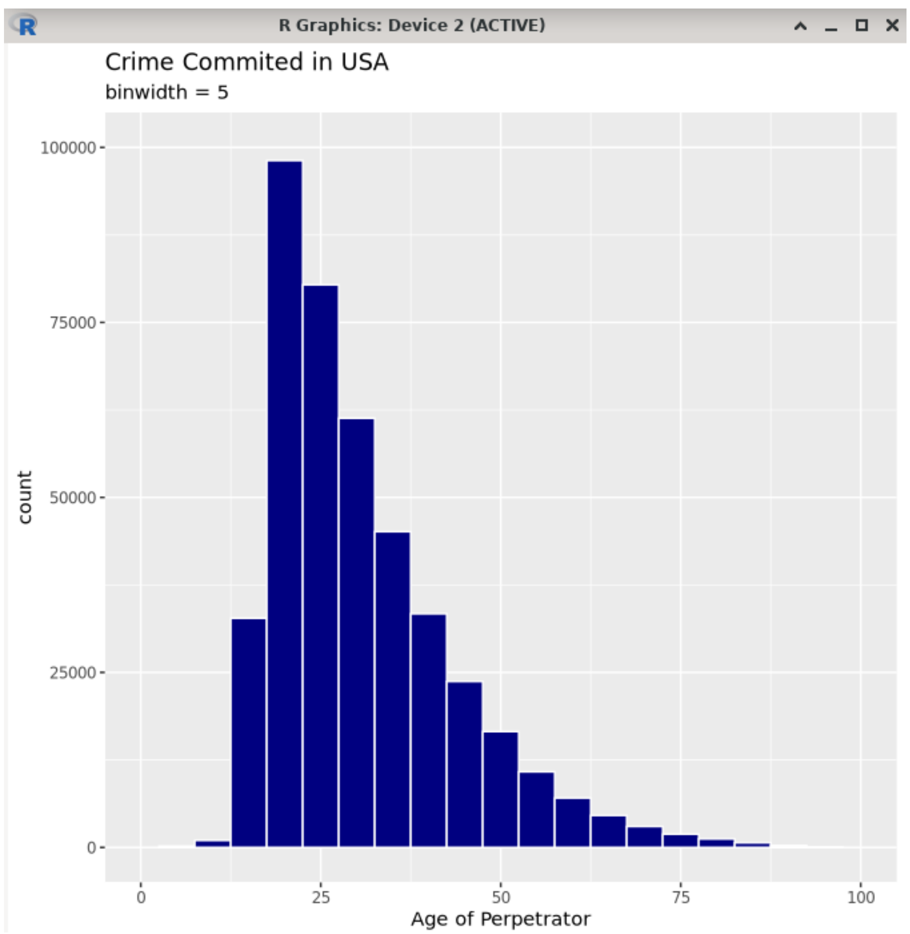


Figure 4.7: *The histogram is showing the age of the perpetrator for crime committed in USA.*

	Male	Female
White	195837	22342
Black	189736	24648
Σ	385573	46990

Table 4.1: *The table to show the perpetrators of crimes committed in USA from 1980 based on gender and race.*

	Male	Female
White	A	B
Black	C	D
Σ	$A+C$	$B+D$

Table 4.2: *We are replacing the numbers with symbolic A, B, C and D.*

ii. Tetrachoric Correlation Case Study: USA Crime Data

We are going to dwell deeper on the tetrachoric correlation.

The tetrachoric correlation describes the linear relation between two continuous variables that have each been measured on a dichotomous scale.

The term "tetrachoric correlation" comes from the tetrachoric series, a numerical method used before the advent of computers. While it is more common to estimate correlations with methods like maximum likelihood estimation, there is a basic formula you can use.

The formula involves the cosine trigonometric function and can be applied to a 2×2 matrix or contingency table:

$$r_{tet} = \cos\left(\frac{180}{1+\sqrt{\frac{AD}{BC}}}\right) \tag{4.4}$$

Assumptions for the Test

The two main assumptions are:

1. The underlying variables come from a normal distribution. With only two variables, this is impossible to test. You should, therefore, have a good theoretical reason for using this particular type of correlation; in other words, you might know that the type of data you are dealing with tends to follow a normal distribution most of the time. Rating errors should also follow a normal distribution.
2. There is a latent continuous scale underneath your binary data. In other words, the trait you are measuring should be continuous and not discrete.

The tetrachoric correlation coefficient r_{tet} (sometimes written as $r*$ or r_t) tells you how strong (or weak) the association is between ratings for two raters. A "0" indicates no agreement and a "1" represents a perfect agreement. Most correlations will fall somewhere in between; what constitutes an acceptable level of agreement largely depends on what type of data you're dealing with.

[R*] You can learn and try to type the code and see the result one by one, it is better with that why in learning programming language, if the syntax or a line does not work you will know how to work it out / how to fix it, you learn more from your mistake.

```
library(ggplot2)
library(dplyr)
library(data.table) #for fread

# https://www.statology.org/correlation-between-categorical-
    variables/

df <- fread("/root/R/CSV/usa_crimes.csv")
summary(df)


head(df)
selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

head(selected_df)

# to obtain the int value from the variable Male perpetrator
maleperpetrator <- count(selected_df, Perpetrator_Sex=='Male')
     %>% .$n
blackmaleperpetrator <- count(selected_df, Perpetrator_Sex=='
    Male', Perpetrator_Race=='Black') %>% pull(n)

# you can count the frequencies of whichever variable(s) you give to
    group_by()
selected_df %>% group_by(Perpetrator_Race,Perpetrator_Sex) %>%
    tally()

# Count Observations by Two Groups and Sort the Results
selected_df %>% count(Perpetrator_Race,Perpetrator_Sex, sort=
    TRUE)


# Given your data is structured as a data frame, the following code
    has a better running time
# to compute conditional categorical variables from 2 data columns
# plus: test run time
ptm <- proc.time()
nrow(subset(selected_df, Perpetrator_Sex=='Male',
```

```
    Perpetrator_Race=='Black'))
proc.time() - ptm

# Now we can input tetrachoric formula and compute
library(psych)

whitemale <- nrow(subset(selected_df, Perpetrator_Sex=='Male'
    & Perpetrator_Race=='White'))
blackmale <- nrow(subset(selected_df, Perpetrator_Sex=='Male'
    & Perpetrator_Race=='Black'))
whitefemale <- nrow(subset(selected_df, Perpetrator_Sex=='
    Female' & Perpetrator_Race=='White'))
blackfemale <- nrow(subset(selected_df, Perpetrator_Sex=='
    Female' & Perpetrator_Race=='Black'))

#create 2x2 table
data = matrix(c(whitemale, whitefemale, blackmale, blackfemale)
    , nrow=2)

#view table
data

#calculate tetrachoric correlation
tetrachoric(data)
```

R Code 21: *tetrachoric correlation test for usa crimes (ch4-usacrimes-tetrachoric.R)*

```
> selected_df %>% count(Perpetrator_Race,Perpetrator_Sex, sort=TRUE)
                Perpetrator_Race Perpetrator_Sex      n
                          <char>          <char>  <int>
 1:                        White            Male 195837
 2:                      Unknown         Unknown 190142
 3:                        Black            Male 189736
 4:                        Black          Female  24648
 5:                        White          Female  22342
 6:                      Unknown            Male   5502
 7:       Asian/Pacific Islander            Male   5449
 8: Native American/Alaska Native           Male   3017
 9: Native American/Alaska Native         Female    578
10:       Asian/Pacific Islander          Female    577
11:                      Unknown          Female    403
12:                        Black         Unknown    132
13:                        White         Unknown     64
14:       Asian/Pacific Islander         Unknown     20
15: Native American/Alaska Native        Unknown      7
```

Figure 4.8: *The **selected_df** that is sorted to see the data based on variables of **Perpetrator_Race** and **Perpetrator_Sex**.*

```
> selected_df %>% count(Perpetrator_Race,Perpetrator_Sex, sort=TRUE)
                 Perpetrator_Race Perpetrator_Sex      n
                           <char>          <char>  <int>
 1:                         White            Male 195837
 2:                       Unknown         Unknown 190142
 3:                         Black            Male 189736
 4:                         Black          Female  24648
 5:                         White          Female  22342
 6:                       Unknown            Male   5502
 7:        Asian/Pacific Islander            Male   5449
 8: Native American/Alaska Native            Male   3017
 9: Native American/Alaska Native          Female    578
10:        Asian/Pacific Islander          Female    577
11:                       Unknown          Female    403
12:                         Black         Unknown    132
13:                         White         Unknown     64
14:        Asian/Pacific Islander         Unknown     20
15: Native American/Alaska Native         Unknown      7
> nrow(subset(selected_df, Perpetrator_Sex == 'Male'))
[1] 399541
> nrow(subset(selected_df, Perpetrator_Sex == 'Male' && Perpetrator_Race=='Black
'))
Error in Perpetrator_Sex == "Male" && Perpetrator_Race == "Black" :
  'length = 638454' in coercion to 'logical(1)'
> nrow(subset(selected_df, Perpetrator_Sex == 'Male' & Perpetrator_Race=='Black'
))
[1] 189736
> nrow(subset(selected_df, Perpetrator_Sex == 'Male' & Perpetrator_Race=='White'
))
[1] 195837
> nrow(subset(selected_df, Perpetrator_Sex == 'Female' & Perpetrator_Race=='Whit
e'))
[1] 22342
```

Figure 4.9: *The* ***selected_df*** *that is sorted to see the data based on variables of* ***Perpetrator_Race*** *and* ***Perpetrator_Sex*** *and the command* ***nrow*** *to compute / return the integer of the selected criteria for both variables.*

```
> source('ch4-usacrimes-tetrachoric.R')
> data
       [,1]   [,2]
[1,] 195837 189736
[2,]  22342  24648
> tetrachoric(data)
Call: tetrachoric(x = data)
tetrachoric correlation
[1] 0.042

 with tau of
[1] 1.234 0.011
```

Figure 4.10: *The tetrachoric correlation based on variables of* ***Perpetrator_Race*** *and* ***Perpetrator_Sex****.*

The tetrachoric correlation turns out to be 0.042. This value is really low, which indicates that there is a weak association (if any) between gender and race in committing crime in USA, so we cannot judge that a Mexican will always be a rapist or Black people tend to commit more crime, it depends solely on how you brought up, your own spirituality, your own dignity and integrity to not fall into temptation and commit sins like murder, steal, rape, etc.

```
julia> cosd(180/(1+sqrt((195837*24648)/(189736*22342))))
0.050962379248426566
```

Figure 4.11: *We try to compute the tetrachoric correlation manually with Julia based on the tetrachoric formula, and we get* 0.05096*, it is quite different with the result from R, we should check what kind of formula is used to compute* r_{tet} *in that R package then.*

iii. Cramer's V Correlation Case Study: USA Crime Data

In statistics, Cramer's V (sometimes referred to as Cramer's phi and denoted as φ_c) is a measure of association between two nominal variables, giving a value between 0 and +1 (inclusive). It is based on Pearson's chi-squared statistic and was published by Harald CramÃl'r in 1946.

Cramer's V Correlation is similar to the Pearson Correlation coefficient. While the Pearson correlation is used to test the strength of linear relationships, Cramer's V is used to calculate correlation in tables with more than 2×2 columns and rows. Cramer's V correlation varies between 0 and 1. A value close to 0 means that there is very little association between the variables. A Cramer's V of close to 1 indicates a very strong association.

Cramer's V	
.25 or higher	Very strong relationship
.15 to .25	Strong relationship
.11 to .15	Moderate relationship
.06 to .10	weak relationship
.01 to .05	No or negligible relationship

Figure 4.12: *A table showing Cramer's V correlation from* 0 *to* 1*.*

The Computation

Let a sample of size n of the simultaneously distributed variables A and B for $i = 1, 2, \ldots, r$ and $j = 1, 2, , \ldots, k$ be given by the frequencies

$$n_{ij}$$

as the number of times the values (A_i, B_j) were observed.

The chi-squared statistic then is

$$\chi^2 = \sum_{i,j} \frac{\left(n_{ij} - \frac{n_i n_j}{n}\right)^2}{\frac{n_i n_j}{n}} \tag{4.5}$$

where $n_i = \sum_j n_{ij}$ is the number of times the value A_i is observed and $n_j = \sum_i n_{ij}$ is the number of times the value B_j is observed.

Cramer's V is computed by taking the square root of the chi-squared statistic divided by the sample size and the minimum dimension minus 1:

$$V = \sqrt{\frac{\varphi^2}{\min(k-1, r-1)}} = \sqrt{\frac{\chi^2/n}{\min(k-1, r-1)}} \tag{4.6}$$

where

- φ is the phi coefficient.
- χ^2 is derived from the Peason's chi-squared test.
- n is the grand total of observations.
- k is the number of columns.
- r is the number of rows.

The p-value for the significance of V is the same one that is calculated using the Pearson's chi-squared test.

In R, the function **cramerV()** from the package rcompanion calculates V using the **chisq.test** function from the stats package. In contrast to the function **cramersV()** from the **lsr** package, **cramerV()** also offers an option to correct for bias.

This is the bias correction Cramer's V can be a heavily biased estimator of its population counterpart and will tend to overestimate the strength of association. A bias correction, using the above notation, is given by

$$\tilde{V} = \sqrt{\frac{\tilde{\varphi}^2}{\min(\tilde{k}-1, \tilde{r}-1)}} \tag{4.7}$$

where

$$\tilde{\varphi}^2 = \max\left(0, \varphi^2 - \frac{(k-1)(r-1)}{n-1}\right)$$

and

$$\tilde{k} = k - \frac{(k-1)^2}{n-1}$$

$$\tilde{r} = r - \frac{(r-1)^2}{n-1}$$

Then $\tilde{V}$ estimates the same population quantity as Cramer's V but with typically much smaller mean squared error. The rationale for the correction is that under independence,

$$E[\varphi^2] = \frac{(k-1)(r-1)}{n-1}$$

[R*] The code that will be used can be seen below

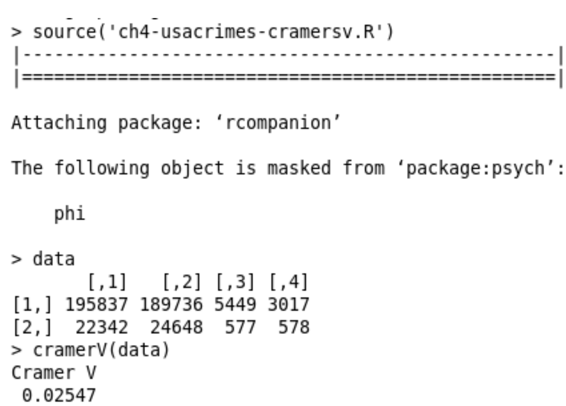
```
library(ggplot2)
library(dplyr)
library(data.table) #for fread
```

```
# https://www.statology.org/correlation-between-categorical-
    variables/

df <- fread("/root/R/CSV/usa_crimes.csv")
summary(df)


head(df)
selected_df = df %>% select(State, Year, Month, Incident,
    Crime_Type, Crime_Solved, Victim_Sex, Victim_Age,
    Victim_Race,
Perpetrator_Sex, Perpetrator_Age, Perpetrator_Race,
    Relationship, Weapon)

head(selected_df)

# to obtain the int value from the variable Male perpetrator
maleperpetrator <- count(selected_df, Perpetrator_Sex=='Male')
     %>% .$n
blackmaleperpetrator <- count(selected_df, Perpetrator_Sex=='
    Male', Perpetrator_Race=='Black') %>% pull(n)

# you can count the frequencies of whichever variable(s) you give to
    group_by()
selected_df %>% group_by(Perpetrator_Race,Perpetrator_Sex) %>%
    tally()

# Count Observations by Two Groups and Sort the Results
selected_df %>% count(Perpetrator_Race,Perpetrator_Sex, sort=
    TRUE)


# Given your data is structured as a data frame, the following code
    has a better running time
# to compute conditional categorical variables from 2 data columns
# plus: test run time
ptm <- proc.time()
nrow(subset(selected_df, Perpetrator_Sex=='Male',
    Perpetrator_Race=='Black'))
proc.time() - ptm

# Now we can input Cramer's V formula and compute
library(rcompanion)

whitemale <- nrow(subset(selected_df, Perpetrator_Sex=='Male'
    & Perpetrator_Race=='White'))
blackmale <- nrow(subset(selected_df, Perpetrator_Sex=='Male'&
     Perpetrator_Race=='Black'))
```

```
asianmale <- nrow(subset(selected_df, Perpetrator_Sex=='Male'
    & Perpetrator_Race=='Asian/Pacific Islander'))
nativeamericanmale <- nrow(subset(selected_df, Perpetrator_Sex
    =='Male' & Perpetrator_Race=='Native American/Alaska Native
    '))
whitefemale <- nrow(subset(selected_df, Perpetrator_Sex=='
    Female' & Perpetrator_Race=='White'))
blackfemale <- nrow(subset(selected_df, Perpetrator_Sex=='
    Female' & Perpetrator_Race=='Black'))
asianfemale <- nrow(subset(selected_df, Perpetrator_Sex=='
    Female' & Perpetrator_Race=='Asian/Pacific Islander'))
nativeamericanfemale <- nrow(subset(selected_df,
    Perpetrator_Sex=='Female' & Perpetrator_Race=='Native
    American/Alaska Native'))

#create 2x4 table
data = matrix(c(whitemale, whitefemale, blackmale, blackfemale,
     asianmale, asianfemale, nativeamericanmale,
    nativeamericanfemale), nrow=2)

#view table
data

#calculate Cramer's V
cramerV(data)
```

R Code 22: *cramers v correlation test for usa crimes (ch4-usacrimes-cramersv.R)*

if the result is not shown, then you can type again **cramerV(data)** at the R' console window.



```
> source('ch4-usacrimes-cramersv.R')
|--------------------------------------------------|
|==================================================|

Attaching package: 'rcompanion'

The following object is masked from 'package:psych':

    phi

> data
       [,1]   [,2] [,3] [,4]
[1,] 195837 189736 5449 3017
[2,]  22342  24648  577  578
> cramerV(data)
Cramer V
 0.02547
```

Figure 4.13: *The computation of Cramer's V test with dataset of **selected_df** and the variables that we are looking to correlate are **Perpetrator_Race** and **Perpetrator_Sex**.*

The result is very low 0.025, it is showing that there is no significant race for each gender that is prone to commit more crime. Both the Tetrachoric and Cramer's V correlation tests state the same, thus we cannot really judge a book from its' cover, if you know the case of Ted Bundy from USA, all the victims fall for his scheme, when he uses police uniform or use his good looking face, and on top of all he is a white caucasian male, that will give him more advantage to wider set of victims, since black male whenever they walk in USA are already seen suspicious, so everyone in this world really need to be more careful, never think if anyone is white then that person is innocent, and it works for other race too, go for martial arts class, improve your spirituality, cleanse your karma, bring dogs everywhere with you for safer, they are human's best friends.

Chapter 5

Probability

God does not play dice. - ***Quantum Mechanics***

In the study of statistics, we are concerned basically with the presentation and interpretation of chance outcomes that occur in a planned study or scientific investigation. the statistician is often dealing with either numerical data, representing counts or measurements, or categorical data, which can be classified according to some criterion. We shall refer to any recording of information, whether it be numerical or categorical, as an observation.

Statisticians use the word experiment to describe any process that generates a set of data. A simple example of a statistical experiment is the tossing of a coin. In this experiment, there are only two possible outcomes, heads or tails. Another experiment might be the launching of a missile and observing of its velocity at specified times.

I. Basic Definition, Theory and Formula

Definition 5.1: Sample Space

The set of all possible outcomes of a statistical experiment is called the sample space and is represented by the symbol S. Each outcome in a sample space is called an element or a member of the sample space, or simply a sample point.

The sample space S, of possible outcomes when a coin is flipped, may be written

$$S = \{H, T\}$$

The sample space for the experiment of tossing a die is

$$S = \{1, 2, 3, 4, 5, 6\}$$

Definition 5.2: Event

An event is a subset of a sample space.

Definition 5.3: Complement

The complement of an event A with respect to S is the subset of all elements of S that are not in A. We denote the complement of A by the symbol A'.

Definition 5.4: Intersection

The intersection of two events A and B, denoted by the symbol $A \cap B$, is the event containing all elements that are common to A and B.

Definition 5.5: Mutually Exclusive

Two events A and B are mutually exclusive, or disjoint, if $A \cap B = \emptyset$, that is, if A and B have no elements in common.

Definition 5.6: Union

The union of two events A and B, denoted by the symbol $A \cup B$, is the event containing all the elements that belong to A or B or both.

Definition 5.7: Permutation

A permutation is an arrangement of all or part of a set of objects.

In general, n distinct objects can be arranged in

$$n! = n(n-1)(n-2)(n-3)\dots 2 \cdot 1$$

ways. So the number of permutations of n objects is $n!$.

Theorem 5.1: Permutation ^nP_r

The number of permutations of n distinct objects taken r at a time is

$$^nP_k = \frac{n!}{(n-r)!} \tag{5.1}$$

Theorem 5.2: Partitioning in Permutation

The number of ways of partitioning a set of n objects into r cells with n_1 elements in the first cell, n_2 elements in the second, and so forth, is

$$\binom{n}{n_1, n_2, \dots, n_r} = \frac{n!}{n_1!n_2!\dots n_r!} \tag{5.2}$$

where $n_1 + n_2 + \cdots + n_r = n$.

Theorem 5.3: Combinations

The number of combinations of n distinct objects taken r at a time is

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{5.3}$$

Definition 5.8: Probability

The probability of an event A is the sum of the weights of all sample points in A. Therefore,

$$0 \leq P(A) \leq 1, \quad P(\emptyset) = 0, \quad P(S) = 1$$

Furthermore, if $A_1, A_2, A_3, \ldots$ is a sequence of mutually exclusive events, then

$$P(A_1 \cup A_2 \cup A_3 \cup \ldots) = P(A_1) + P(A_2) + P(A_3) + \ldots$$

Theorem 5.4: Additive Rule for Two Events

If A and B are two events, then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{5.4}$$

Theorem 5.5: Additive Rule for Three Events

If A, B and C are three events, then

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C) \tag{5.5}$$

Definition 5.9: Conditional Probability

The conditional probability of B, given A, denoted by $P(B|A)$, is defined by

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{5.6}$$

provided that $P(A) > 0$.

Definition 5.10: Independent Events

Two events A and B are independent if and only if

$$\begin{aligned} P(B|A) &= P(B) \\ P(A|B) &= P(A) \end{aligned} \tag{5.7}$$

assuming the existence of the conditional probabilities. Otherwise, A and B are dependent.

There is also another formula with the multiplication of the probabilities of two events occurring

$$P(A \cap B) = P(A)P(B) \tag{5.8}$$

Definition 5.11: Bayes' Rule

Bayesian statistics is a collection of tools that is used in a special form of statistical inference which applies in the analysis of experimental data in many practical situations in science and engineering. Bayes' rule is one of the most important rules in probability theory.

Theorem 5.6: Total Probability

If the events $B_1, B_2, \ldots, B_k$ constitute a partition of the sample space S such that $P(B_i) \neq 0$ for $i = 1, 2, \ldots, k$, then for any event A of S,

$$P(A) = \sum_{i=1}^{k} P(B_i \cap A) = \sum_{i=1}^{k} P(B_i)P(A|B_i) \tag{5.9}$$

Theorem 5.7: Bayes' Rule

If the events $B_1, B_2, \ldots, B_k$ constitute a partition of the sample space S such that $P(B_i) \neq 0$ for $i = 1, 2, \ldots, k$, then for any event A in S such that $P(A) \neq 0$,

$$P(B_r|A) = \frac{P(B_r \cap A)}{\sum_{i=1}^{k} P(B_i \cap A)} = \frac{P(B_r)P(A|B_r)}{\sum_{i=1}^{k} P(B_i)P(A|B_i)} \tag{5.10}$$

for $r = 1, 2, \ldots, k$.

II. Compute Conditional Probability

III. Compute Conditional Probability with Bayes' Rule

Chapter 6

Random Variable and Probability Distributions

*To be, or not to be? That is the question. Whether 'tis nobler in the mind to suffer. The slings and arrows of outrageous fortune, or to take arms against a sea of troubles, and, by opposing, end them? - **Shakespeare***

Statistics is concerned with making inferences about populations and population characteristics. Experiments are conducted with results that are subject to chance. The testing of a number of PCB(Printed Circuit Board) is an example of a statistical experiment, a term that is used to describe any process by which several chance observations are generated. It is often important to allocate a numerical description to the outcome.

I. Basic Definition, Theory and Formula

i. Random Variable

Definition 6.1: Random Variable

A random variable is a function that associates a real number with each element in the sample space.

We shall use a capital letter, say X, to denote a random variable and its corresponding small letter, x in this case, for one of its values.

For example, the sample space giving a detailed description of each possible outcome when three electronic components are tested may be written

$$S = \{NNN, NND, NDN, DNN, NDD, DND, DDN, DDD\}$$

where N denotes nondefective and D denotes defective.

If we want to take where the subset contains 2 defectives in exact, then

$$P(X = 2) = E = \{DDN, DND, NDD\}$$

where E is the subset of S. That is, each possible value of X represent an event that is a subset of the sample space for the given experiment.

The probabilistic view of the data assumes that each numeric attribute X is a random variable, defined as a function that assigns a real number to each outcome of an experiment (i.e., some process of observation or measurement).

Formally, X is a function $X : D \to \mathbb{R}$, where D is the domain of X and $\mathbb{R}$ is the range of X. It is a set of all possible outcomes of the experiment.

A random variable X is called a discrete random variable if it takes on only a finite or countably infinite number of values in its range, whereas X is called a continuous random variable if it can take on any value in its range.

Definition 6.2: Discrete Sample Space

If a sample space contains a finite number of possibilities or an unending sequence with as many elemnts as there are whole numbers, it is called a discrete sample space.

Definition 6.3: Continuous Sample Space

If a sample space contains an infinite number of possibilities equal to the number of points on a line segment, it is called a continuous sample space.

ii. Discrete Probability Distributions

Definition 6.4: Probability Mass Function

If X is discrete, the probability mass function of X is defined as

$$f(x) = P(X = x), \quad \forall x \in \mathbb{R} \tag{6.1}$$

the probability is concentrated at only discrete values in the range of X, and is zero for all other values. f must also obey the basic rules of probability. That is, f must be non-negative

$$f(x) \geq 0$$

and the sum of all probabilities should add to 1

$$\sum_x f(x) = 1$$

Definition 6.5: Discrete Cumulative Distribution Function

The cumulative distribution function $F(x)$ of a discrete random variable X with probability distribution $f(x)$ iss

$$F(x) = P(X \leq x) = \sum_{t \leq x} f(t), \quad \text{for } -\infty < x < \infty \tag{6.2}$$

iii. Continuous Probability Distributions

Definition 6.6: Probability Density Function

If X is continuous, its range is the entire set of real numbers $\mathbb{R}$. The probability of any specific value x is only one out of the infinitely many possible values in the range of X, which means that

$$P(X = x) = 0$$

for all $x \in \mathbb{R}$. The probability mass is spread so thinly over the range of values, that it can be measured only over intervals $[a, b] \subset \mathbb{R}$, rather than at specific points.
the probability density function of X that takes on values in any interval $[a, b] \subset \mathbb{R}$ is defined as

$$P(X \in [a, b]) = \int_a^b f(x)\, dx \tag{6.3}$$

the density function f must satisfy the basic laws of probability

$$f(x) \geq 0, \quad \forall x \in \mathbb{R}$$

and

$$\int_{-\infty}^{\infty} f(x)\, dx = 1$$

Definition 6.7: Continuous Cumulative Distribution Function

The function $f(x)$ is a probability density function (pdf) for the continuous random variable X, defined over the set of real numbers, if

$$f(x) \geq 0, \quad \forall x \in \mathbb{R} \tag{6.4}$$

$$\int_{-\infty}^{\infty} f(x)\, dx = 1 \tag{6.5}$$

$$P(a < X < b) = \int_a^b f(x)\, dx \tag{6.6}$$

Definition 6.8: Cumulative Distribution Function

For any random variable X, whether discrete or continuous, we can define the cumulative distribution function (cdf) as

$$F : \mathbb{R} \to [0, 1]$$

that gives the probability of observing a value at most some given value x

$$F(x) = P(X \leq x), \quad \forall -\infty < x < \infty \tag{6.7}$$

when X is discrete, F is given as

$$F(x) = P(X \leq x) = \sum_{u \leq x} f(u) \tag{6.8}$$

and when X is continuous, F is given as

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)\, du \tag{6.9}$$

iv. Joint Probability Distributions

Definition 6.9: Joint Probability Distribution

The function $f(x, y)$ is a joint probability distribution or probability mass function of the discrete random variables X and Y if

$$f(x, y) \geq 0, \quad \forall\ (x, y) \tag{6.10}$$

$$\sum_x \sum_y f(x, y) = 1 \tag{6.11}$$

$$P(X = x, Y = y) = f(x, y) \tag{6.12}$$

For any region A in the xy plane,

$$P[(X, Y) \in A] = \sum \sum_A f(x, y) \tag{6.13}$$

Definition 6.10: Joint Density Function

The function $f(x,y)$ is a joint density function of the continuous random variables X and Y if

$$f(x,y) \geq 0, \quad \forall\ (x,y) \tag{6.14}$$

$f(x,y)$ is a surface lying above the xy plane, and $P[(X,Y) \in A]$, where A is any region in the xy plane. The joint density function is equal to the volume of the right cylinder bounded by the base A and the surface.

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ dx\ dy = 1 \tag{6.15}$$

$$P[(X,Y)\infty A] = \int\int_A f(x,y)\ dx\ dy \tag{6.16}$$

for any region A in the xy plane.

Definition 6.11: Marginal Distributions

Given the joint probability distribution $f(x,y)$ of the discrete random variables X and Y, the probability distribution $g(x)$ of X alone is obtained by summing $f(x,y)$ over the values of Y. Similarly, the probability distribution $h(y)$ of Y alone is obtained by summing $f(x,y)$ over the values of X.

We define $g(x)$ and $h(y)$ to be the marginal distributions of X and Y, respectively. When X and Y are continuous random variables, summations are replaced by integrals.

The marginal distributions of X alone and Y alone are

$$\begin{aligned} g(x) &= \sum_y f(x,y) \\ h(y) &= \sum_x f(x,y) \end{aligned} \tag{6.17}$$

for the discrete case, and

$$\begin{aligned} g(x) &= \int_{-\infty}^{\infty} f(x,y)\ dy \\ h(y) &= \int_{-\infty}^{\infty} f(x,y)\ dx \end{aligned} \tag{6.18}$$

for the continuous case.

Definition 6.12: Conditional Distribution

Let X and Y be two random variables, discrete or continuous. The conditional distribution of the random variable Y given that $X = x$ is

$$f(y|x) = \frac{f(x,y)}{g(x)} \tag{6.19}$$

provided $g(x) > 0$.
Similarly, the conditional distribution of X given that $Y = y$ is

$$f(x|y) = \frac{f(x,y)}{h(y)} \tag{6.20}$$

provided $h(y) > 0$.

If we wish to find the probability that the discrete random variable X falls between a and b when it is known that the discrete variable $Y = y$, we evaluate

$$P(a < X < b|Y = y) = \sum_{a<x<b} f(x|y) \tag{6.21}$$

where the summation extends over all values of X between a and b. When X and Y are continuous, we evaluate

$$P(a < X < b|Y = y) = \int_a^b f(x|y)\, dx \tag{6.22}$$

Definition 6.13: Statistically Independent

Let X and Y be two random variables, discrete or continuous, with joint probability distribution $f(x,y)$ and marginal distributions $g(x)$ and $h(y)$, respectively. The random variables X and Y are said to be statistically independent if and only if

$$f(x,y) = g(x)h(y) \tag{6.23}$$

for all (x,y) within their range.

Definition 6.14: Mutually Statistically Independent

Let $X_1, X_2, \ldots, X_n$ be n random variables, discrete or continuous, with joint probability distribution $f(x_1, x_2, \ldots, x_n)$ and marginal distribution $f_1(x_1), f_2(x_2), \ldots, f_n(x_n)$, respectively. The random variables $X_1, X_2, \ldots, X_n$ are said to be mutually statistically independent if and only if

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1)f_2(x_2)\ldots f_n(x_n) \tag{6.24}$$

for all $(x_1, x_2, \ldots, x_n)$ within their range.

II. Discrete Random Variable: Plot probability mass function, and cumulative distribution function with ggplot2 and plot generic

While the cdf is a theoretical construct that describes the probability of a random variable being less than or equal to a certain value, the empirical cumulative distribution function (ecdf) is derived directly from sample data. This makes the ecdf particularly useful in scenarios where the underlying distribution is unknown or complex.

Let X be a random variable.

- The cumulative distribution function $F(x)$ gives the $P(X \leq x)$.
- An empirical cumulative distribution function $G(x)$ gives $P(X \leq x)$ based on the observations in your sample.

Coin Flip Example:

Let X be a random variable denoting the result of a single coin flip where $X = 1$ denotes heads and $X = 0$ denotes tails.

The cdf for a fair coin is given by

$$F(x) = \begin{cases} 0, & \text{for } x < 0 \\ \frac{1}{2}, & \text{for } 0 \leq x < 1 \\ 1, & \text{for } x \geq 1 \end{cases}$$

If you flipped 2 heads and 1 tail, the empirical cdf (ecdf) would be

$$F(x) = \begin{cases} 0, & \text{for } x < 0 \\ \frac{2}{3}, & \text{for } 0 \leq x < 1 \\ 1, & \text{for } x \geq 1 \end{cases}$$

the empirical cdf would reflect that in your sample , 2/3 of your flips were heads.

The empirical cumulative distribution function (ecdf) provides an alternative visualization of distribution. Compared to other visualizations that rely on density (like **geom_histogram()**), the ecdf doesn't require any tuning parameters and handles both continuous and categorical variables. The downside is that it requires more training to accurately interpret, and the underlying visual tasks are somewhat more challenging.

[R*] We are going to use this illustration for the case study:
If a car agency sells 50% of its inventory of a certain foreign car equipped with side airbags, find a formula for the probability distribution of the number of cars with side airbags among the next 4 cars sold by the agency.

Solution:
Since the probability of selling an automobile with side airbags is 0.5, the $2^4 = 16$ points in the sample space are equally likely to occur. Therefore, the denominator for all probabilities, and also for our function, is 16.

To obtain the number of ways of selling 3 cars with side airbags, we need to consider the number of ways of partitioning 4 outcomes into two cells, with 3 cars with side airbags assigned to one cell and the model without side airbags can occur in $\binom{4}{x}$ ways, where x can be $0, 1, 2, 3$ or 4. Thus, the probability distribution $f(x) = P(X = x)$ is

$$f(x) = \frac{1}{16}\binom{4}{x}, \quad \text{for } x = 0, 1, 2, 3, 4$$

Direct calculations of the probability distribution give

$$\begin{aligned} f(0) &= \frac{1}{16} \\ f(1) &= \frac{1}{4} \\ f(2) &= \frac{3}{8} \\ f(3) &= \frac{1}{4} \\ f(4) &= \frac{1}{16} \end{aligned}$$

Therefore,

$$\begin{aligned} F(0) &= f(0) = \frac{1}{16} \\ F(1) &= f(0) + f(1) = \frac{5}{16} \\ F(2) &= f(0) + f(1) + f(2) = \frac{11}{16} \\ F(3) &= f(0) + f(1) + f(2) + f(3) = \frac{15}{16} \\ F(4) &= f(0) + f(1) + f(2) + f(3) + f(4) = \frac{16}{16} = 1 \end{aligned}$$

Hence,

$$F(x) = \begin{cases} 0, & \text{for } x < 0 \\ \frac{1}{16}, & \text{for } 0 \leq x < 1 \\ \frac{5}{16}, & \text{for } 1 \leq x < 2 \\ \frac{11}{16}, & \text{for } 2 \leq x < 3 \\ \frac{15}{16}, & \text{for } 3 \leq x < 4 \\ 1, & \text{for } x \geq 4 \end{cases}$$

[R*] Now, it is time to use R for the plotting of probability mass function (pmf) and the cumulative distribution function (cdf).

```
library(ggplot2)

## Code for PMF
ggplot(data = data.frame(x = 1:5,
y = c(1/16,1/4,3/8,1/4,1/16),
```

```
yend = rep(0,5)),
aes(x = x, y = y, xend = x, yend = yend)) +
geom_point() +
geom_segment() +
scale_x_continuous(name="\nx",
breaks=1:5,
limits = c(0.5, 4.5)) +
scale_y_continuous(name="f(x)\n",
limits = c(0.0,0.6)) +
ggtitle("PMF for discrete random variable X\n") +
annotate(geom = "text",
x = c(1:5),
y = c(1/16 + 0.03,1/4 + 0.03,3/8 + 0.03,1/4 + 0.03,1/16 + 0.03)
    ,
label = c("P = 1/16",
"P = 1/4",
"P = 3/8",
"P = 1/4",
"P = 1/16")) +
theme_bw() +
theme(plot.title = element_text(hjust = 0.5),
text = element_text(size = 15))
```

R Code 23: *discrete cumulative distribution function (ch6-probabilitymassfunction.R)*

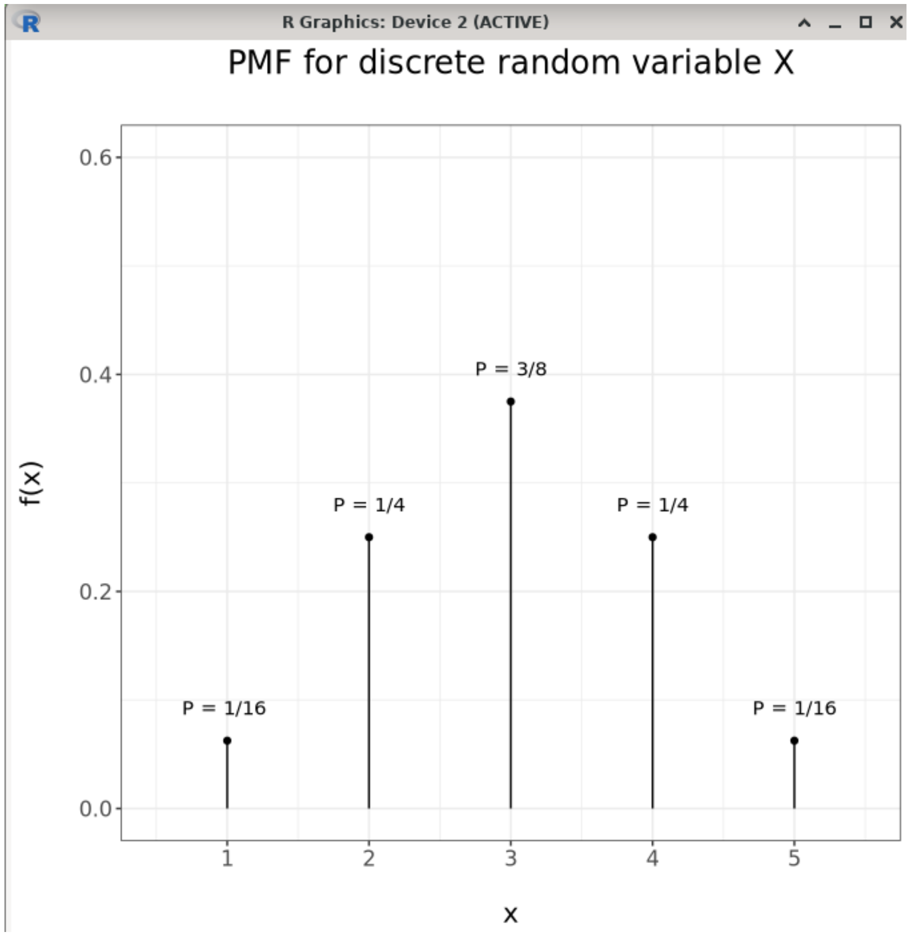


Figure 6.1: *The probability mass function plot for the case study of selling an automobile with side airbags.*

```
# https://stackoverflow.com/questions/66266703/draw-discrete-cdf
    -in-r
library(ggplot2)

# Create the data first
x <- 0:5
fx <- c(0, 1/16, 1/4, 3/8, 1/4, 1/16)

Fx <- cumsum(fx)
n <- length(x)

#make an empty plot

p <- plot(x = NA, y = NA, pch = NA,
xlim = c(0, max(x)),
ylim = c(0, 1),
xlab = "x",
```

```
            ylab = "F(x)",
            main = "Discrete cumulative distribution function")

# Create the points and lines
points(x = x[-n], y = Fx[-1], pch=19)
points(x = x[-1], y = Fx[-1], pch=1)
for(i in 1:(n-1)) points(x=x[i+0:1], y=Fx[c(i,i)+1], type="l")


print(p)
```

R Code 24: *discrete cumulative distribution function (ch6-discretecdf.R)*

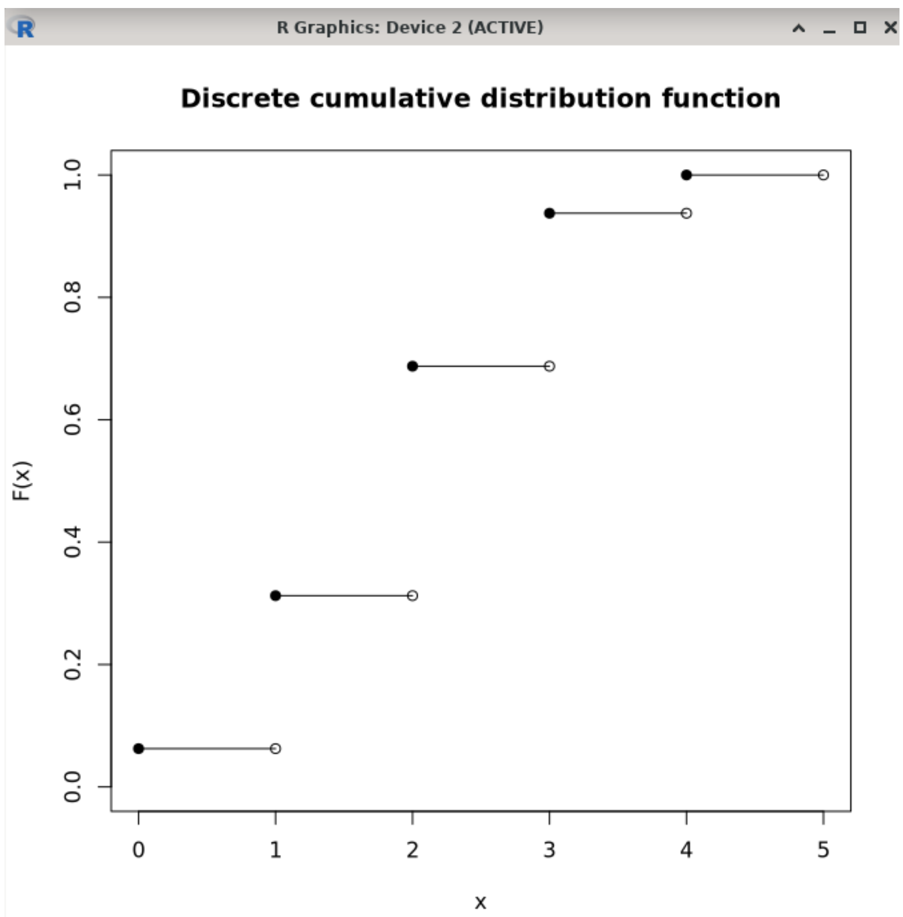


Figure 6.2: *The discrete cumulative distribution function plot for the case study of selling an automobile with side airbags.*

III. Continuous Random Variable: Plot probability density function, and cumulative distribution function with ggplot2

Remember that a continuous random variable has a probability of 0 of assuming exactly any of its values. Consequently, its probability distribution cannot be given in tabular form. For example the random variable that represents the height of a student, between two values, say 165.1 and 165.6 centimeters, there are an infinite number of heights one of which is 164 centimeters. We can only compute the probabilities for various intervals of continuous random variable such as

$$P(a < X < b), \quad P(W \geq c), \ldots$$

when X is continuous.

[R*] We will use dataset **iris.csv**, it is a very popular dataset and can be found easily on internet, and also available at the repository for this book (**https://github.com/glanzkaiser/GFreya-R-for-Statistics/CSV**).

[R*] Before we do the plotting, I am not a Biology student and not a native English speaker so I am quite confused when first read about "petal" and "sepal" so here are the definitions:

Sepal, any of the outer parts of a flower that enclose and protect the unopened flower bud. The sepals on a flower are collectively referred to as the calyx. They are sterile floral parts and may be either green or leaflike or composed of petal-like tissue.

A petal is a part of a flower. Most flowers have a ring of brightly colored petals surrounding the center part of the blossom. Petal comes from the Greek word petalon, meaning "leaf, thin plate." A petal is the lovely colorful leaf-like ring around the center of the flower, a thin plate for a fairy.

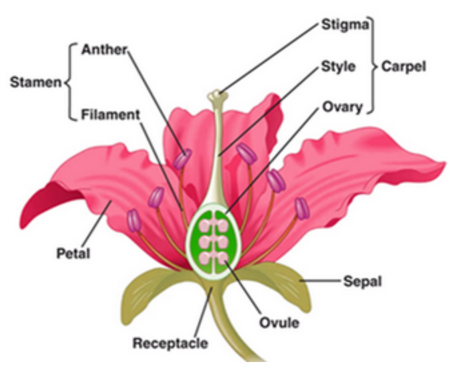


Figure 6.3: *The flower structure.*

[R*] We will choose a variable **Sepal Length**, it is a continuous random variable due to the real number that represent the sepal length.

[R*] For the ecdf, we will use **plot** function, it is a generic function for plotting of R objects.

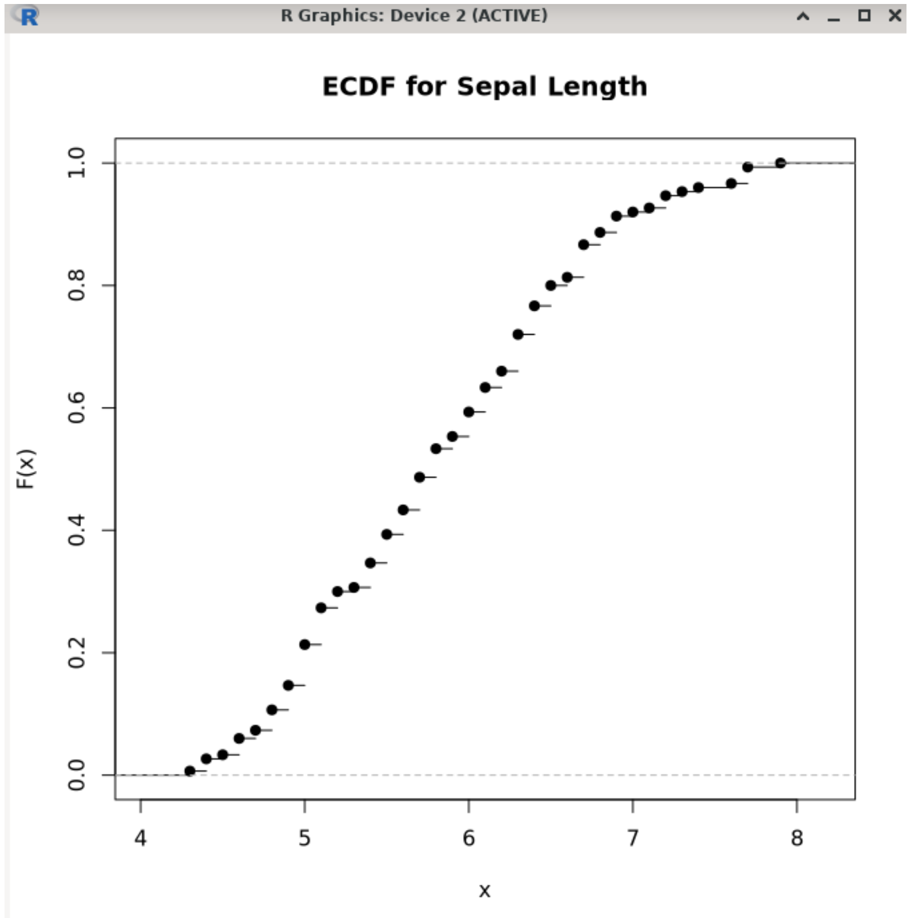


Figure 6.4: *The ecdf for structure.*

```
library(dplyr)
#library(ggplot2)
library(data.table) #for fread
library(stringr)
library(scales) # for comma in geom_text

df <- fread("/root/R/CSV/iris.csv")

head(df)

plot(ecdf(iris$sepal.length))
```

R Code 25: *ecdf for sepal length (ch6-ecdfsepallength.R)*

Chapter 7

Discrete Probability Distributions

As a young boy my brother Nobunori studied the Chinese classics, and I liked to sit in and listen to his lessons. I found that even when he struggled to understand or memorize passages, I would find them remarkably easy. My father, a well-read man himself, often used to lament this fact, saying, 'Such a shame. Would that you were born a man!' - ***Lady Murasaki***

Chapter 8

Continuous Probability Distributions

The ascent to the highest story is by stairs, and at their side are water engines, by means of which persons, appointed expressly for the purpose, are continually employed in raising water from the Euphrates into the garden. - ***Strabo on the Hanging Gardens***

Chapter 9

Statistical Modelling

Chapter 10

Regression

Chapter 11

Analysis of Variance

Chapter 12

Analysis of Covariance

Chapter 13

Generalized Linear Models

Chapter 14

Generalized Additive Models

Chapter 15

Non-linear Regression

Chapter 16

Tree Models

Chapter 17

Time Series Analysis

Chapter 18

Multivariate Statistics

Chapter 19

Spatial Statistics

Chapter 20

Survival Analysis

Chapter 21

Packages Needed to be Installed

These are packages that are used in this book:

1. dplyr
   A fast, consistent tool for working with data frame like objects, both in memory and out of memory.
   The dplyr package provides five functions which cover fundamental data management tasks. These are:

   (a) select, for selecting-filtering columns of the dataset
   (b) filter, for selecting-filtering rows of the dataset
   (c) arrange, for sorting rows based on values of particular columns
   (d) mutate, for creating new variables from existing ones
   (e) summarize, for data aggregation - very useful when combined with grouped data

2. ggplot2

3. ggrepel
   Provides text and label geoms for 'ggplot2' that help to avoid overlapping text labels. Labels repel away from each other and away from the data points.

4. ggthemes
   Some extra themes, geoms, and scales for 'ggplot2'. Provides 'ggplot2' themes and scales that replicate the look of plots by Edward Tufte, Stephen Few, 'Fivethirtyeight', 'The Economist', 'Stata', 'Excel', and 'The Wall Street Journal', among others. Provides 'geoms' for Tufte's box plot and range frame.

5. lubridate
   Functions to work with date-times and time-spans: fast and user friendly parsing of date-time data, extraction and updating of components of a date-time (years, months, days, hours, minutes, and seconds), algebraic manipulation on date-time and time-span objects. The 'lubridate' package has a consistent and memorable syntax that makes working with dates easy and fun.

6. mosaicData

7. plotly
   Create interactive web graphics from 'ggplot2' graphs and/or a custom interface to the (MIT-licensed) JavaScript library 'plotly.js' inspired by the grammar of graphics.

8. psych
   A general purpose toolbox developed originally for personality, psychometric theory and experimental psychology. Functions are primarily for multivariate analysis and scale construction using factor analysis, principal component analysis, cluster analysis and reliability analysis, although others provide basic descriptive statistics. Item Response Theory is done using factor analysis of tetrachoric and polychoric correlations. Functions for analyzing data at multiple levels include within and between group statistics, including correlations and factor analysis. Validation and cross validation of scales developed using basic machine learning algorithms are provided, as are functions for simulating and testing particular item and test structures. Several functions serve as a useful front end for structural equation modeling. Graphical displays of path diagrams, including mediation models, factor analysis and structural equation models are created using basic graphics. Some of the functions are written to support a book on psychometric theory as well as publications in personality research.

9. RColorBrewer
   Provides color schemes for maps (and other graphics) designed by Cynthia Brewer library(data.table)

10. rcompanion
    Functions and datasets to support Summary and Analysis of Extension Program Evaluation in R, and An R Companion for the Handbook of Biological Statistics.

11. scales
    Graphical scales map data to aesthetics, and provide methods for automatically determining breaks and labels for axes and legends.

12. stringr
    A consistent, simple and easy to use set of wrappers around the fantastic 'stringi' package. All function and argument names (and positions) are consistent, all functions deal with "NA"'s and zero length vectors in the same way, and the output from one function is easy to feed into the input of another.

13. tidyr
    Tools to help to create tidy data, where each column is a variable, each row is an observation, and each cell contains a single value. 'tidyr' contains tools for changing the shape (pivoting) and hierarchy (nesting and 'unnesting') of a dataset, turning deeply nested lists into rectangular data frames ('rectangling'), and extracting values out of string columns. It also includes tools for working with missing values (both implicit and explicit).

14. treemapify
    Provides 'ggplot2' geoms for drawing treemaps.

Bibliography

[1] Crawley, Michael J., The R Book, John Wiley & Sons, England, 2007.

[2] Kabacoff, Robert, Modern Data Visualization with R, CRC Press, Boca Raton, USA, 2024.

[3] Lantz, Brett, Machine Learning with R 4th Edition, Packt, 2023.

[4] Oleksy, Andrew, Data Science with R, Andrew Oleksy, 2018.

[5] Walpole, Ronald E., Myers, Raymond H., Myers, Sharon L., Ye, Keying, Probability & Statistics 9th Edition, Pearson, Boston, USA, 2012.

[6] Zaki, Mohammed, Meira Jr., Wagner, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, England, 2014.