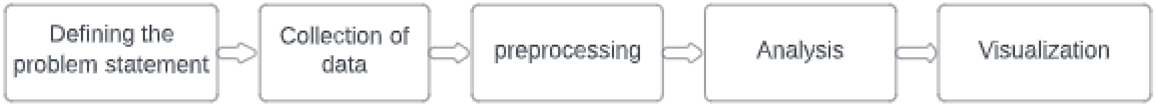
Assignment No. 2.3

Aim: Data visualization using R Programming (Consider different input files like csv, excel, JSON etc.)

Theory:

Data Analysis is a subset of data analytics, it is a process where the objective has to be made clear, collect the relevant data, preprocess the data, perform analysis (understand the data, explore insights), and then visualize it. The last step visualization is important to make people understand what’s happening in the firm.

Steps involved in data analysis:



The aim of this practical is to explore data visualization techniques in R using different types of input files such as CSV, Excel, and JSON. Data visualization plays a crucial role in understanding complex datasets by providing a graphical representation of data patterns, trends, and relationships.

Data Import and Manipulation:

Using the readxl and writexl libraries, Excel files are loaded into R, allowing for data extraction and manipulation. Specific cells and sheets are read to focus on relevant data, while rows and columns can be skipped for streamlined analysis. Similarly, CSV files are imported using read.csv, showcasing R's flexibility in handling different data formats. Data frames are created and combined to generate new datasets, which can be written back into CSV format for future use.

Visualization Techniques:

R provides powerful visualization libraries like ggplot2 and base plotting functions. Basic plots such as scatter plots are created to explore the relationships between variables, as seen in the iris dataset's Petal Length and Width. Enhancements like changing point shapes (pch), colors, and adding labels help in better data interpretation. The ggplot2 library offers advanced customization, enabling layered plotting with aesthetics (e.g., color, shape) mapped to variables like Species. This modular approach allows for clear, informative visualizations tailored to specific analytical needs.

By combining data handling and visualization, this practical highlights how R can transform raw data from multiple sources into actionable insights through clear, impactful visual representations.

Code and Output:

install.packages("readxl")

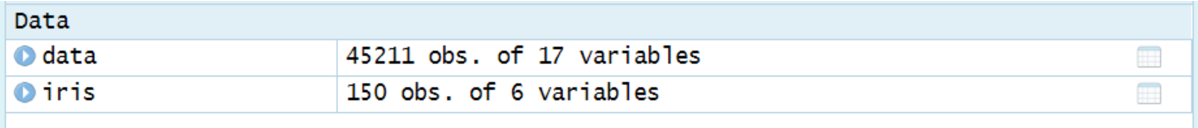
install.packages("writexl")

library(readxl)

library(writexl)

data <- read_excel("file_show (6).xlsx")

iris <- read_excel("file_show (6).xlsx", sheet = "iris")

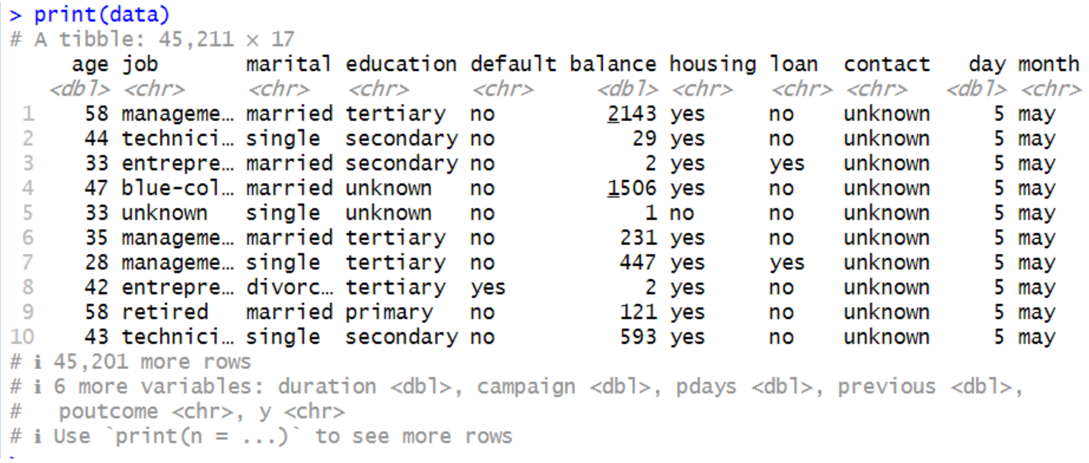


```
Data
data    45211 obs. of 17 variables
iris    150 obs. of 6 variables
```

bank_full <- read_xlsx("file_show (6).xlsx", sheet = 1)

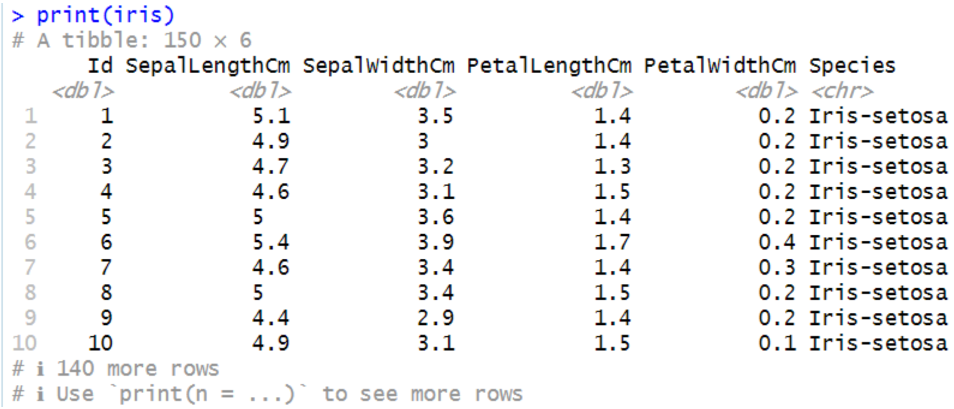
sdbank_full <- read_xlsx("file_show (6).xlsx", sheet = 1, skip=5)

print(data)



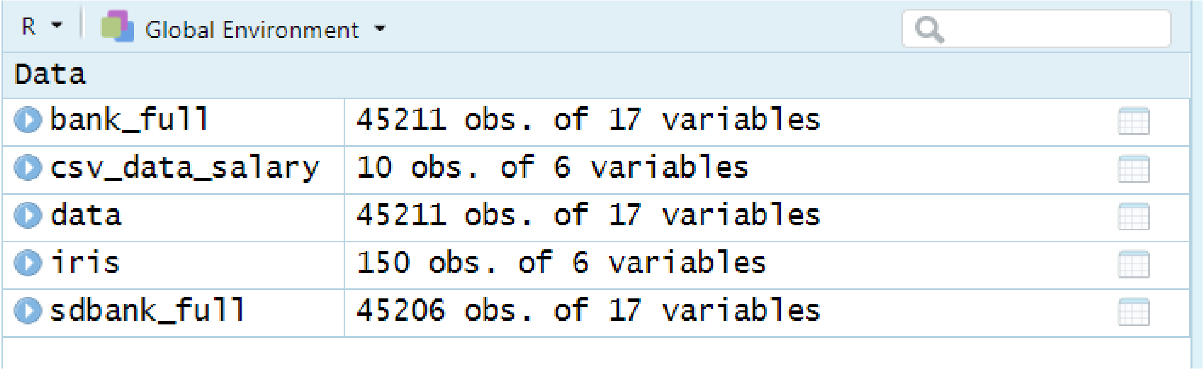
```
> print(data)
# A tibble: 45,211 × 17
     age job        marital education default balance housing loan  contact   day month
   <dbl> <chr>      <chr>   <chr>     <chr>     <dbl> <chr>   <chr> <chr>   <dbl> <chr>
 1    58 manageme… married tertiary  no         2143 yes     no    unknown     5 may
 2    44 technici… single  secondary no           29 yes     no    unknown     5 may
 3    33 entrepre… married secondary no            2 yes     yes   unknown     5 may
 4    47 blue-col… married unknown   no         1506 yes     no    unknown     5 may
 5    33 unknown   single  unknown   no            1 no      no    unknown     5 may
 6    35 manageme… married tertiary  no          231 yes     no    unknown     5 may
 7    28 manageme… single  tertiary  no          447 yes     yes   unknown     5 may
 8    42 entrepre… divorc… tertiary  yes           2 yes     no    unknown     5 may
 9    58 retired   married primary   no          121 yes     no    unknown     5 may
10    43 technici… single  secondary no          593 yes     no    unknown     5 may
# ℹ 45,201 more rows
# ℹ 6 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>, previous <dbl>,
#   poutcome <chr>, y <chr>
# ℹ Use `print(n = ...)` to see more rows
```

print(iris)



```
> print(iris)
# A tibble: 150 × 6
      Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
   <dbl>         <dbl>        <dbl>         <dbl>        <dbl> <chr>
 1     1           5.1          3.5           1.4          0.2 Iris-setosa
 2     2           4.9          3             1.4          0.2 Iris-setosa
 3     3           4.7          3.2           1.3          0.2 Iris-setosa
 4     4           4.6          3.1           1.5          0.2 Iris-setosa
 5     5           5            3.6           1.4          0.2 Iris-setosa
 6     6           5.4          3.9           1.7          0.4 Iris-setosa
 7     7           4.6          3.4           1.4          0.3 Iris-setosa
 8     8           5            3.4           1.5          0.2 Iris-setosa
 9     9           4.4          2.9           1.4          0.2 Iris-setosa
10    10           4.9          3.1           1.5          0.1 Iris-setosa
# ℹ 140 more rows
# ℹ Use `print(n = ...)` to see more rows
```

csv_data_salary <- read_xlsx("file_show (4).xlsx")

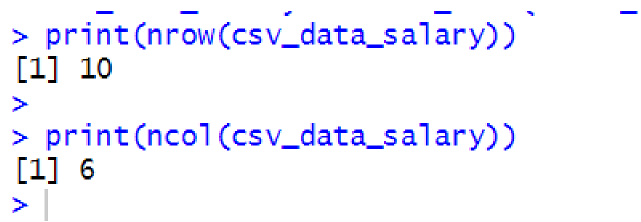


Data

bank_full	45211 obs. of 17 variables
csv_data_salary	10 obs. of 6 variables
data	45211 obs. of 17 variables
iris	150 obs. of 6 variables
sdbank_full	45206 obs. of 17 variables

print(nrow(csv_data_salary))

print(ncol(csv_data_salary))



```
> print(nrow(csv_data_salary))
[1] 10
>
> print(ncol(csv_data_salary))
[1] 6
>
```

result <- csv_data_salary[csv_data_salary$salary > 60000, c("name", "salary")]

#create dfs -> then combine into 1 single df-> write this on a csv file

Country <- c("China", "India", "US", "Indonesia", "Pakistan")

Population_1_july_2018 <- c("1,427,647,786", "1,352, 642,280",

"327,096,265", "267,670,543", "212, 228,286")

Population_1_july_2019 <- c("1,433,783,686", "1,366,417,754",

"329,064,917", "270,625,568", "216, 565,318")

change_in_percents <- c("+0.43%", "+1.02%", "+0.60%", "+1.10%",

"+2.04%")

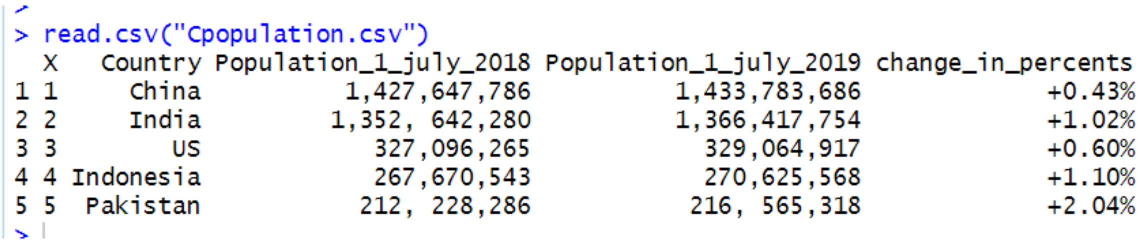
SDF <- data.frame(Country, Population_1_july_2018, Population_1_july_2019, change_in_percents)

write.csv(SDF, "Cpopulation.csv")

	Country	Population_1_july_2018	Population_1_july_2019	change_in_percents
1	China	1,42,76,47,786	1,43,37,83,686	0.43%
2	India	1,352, 642,280	1,36,64,17,754	1.02%
3	US	32,70,96,265	32,90,64,917	0.60%
4	Indonesia	26,76,70,543	27,06,25,568	1.10%
5	Pakistan	212, 228,286	216, 565,318	2.04%

library(readr)

read.csv("Cpopulation.csv")



```
> read.csv("Cpopulation.csv")
  X   Country Population_1_july_2018 Population_1_july_2019 change_in_percents
1 1     China          1,427,647,786          1,433,783,686             +0.43%
2 2     India         1,352, 642,280          1,366,417,754             +1.02%
3 3        US            327,096,265            329,064,917             +0.60%
4 4 Indonesia            267,670,543            270,625,568             +1.10%
5 5  Pakistan           212, 228,286           216, 565,318             +2.04%
> 
```

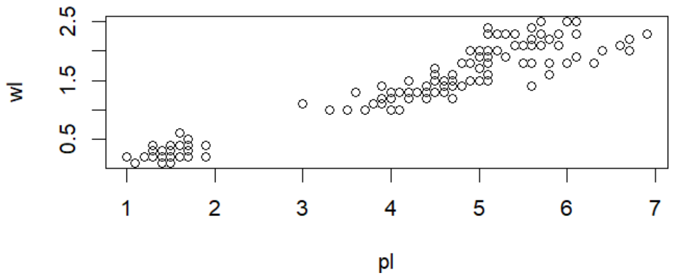
#PLOT

View(iris)

pl <- iris$PetalLengthCm

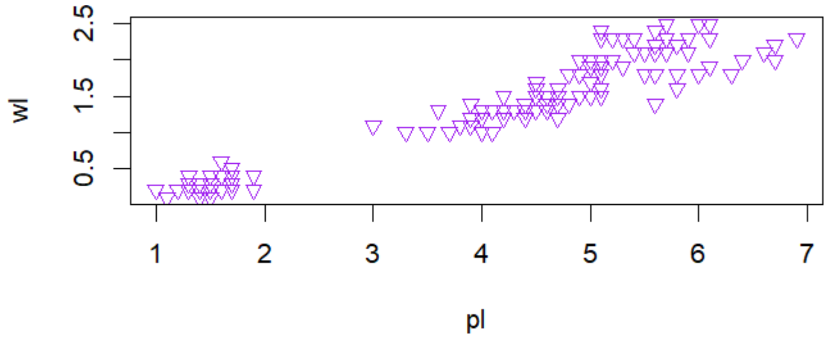
wl <- iris$PetalWidthCm

plot(pl, wl)

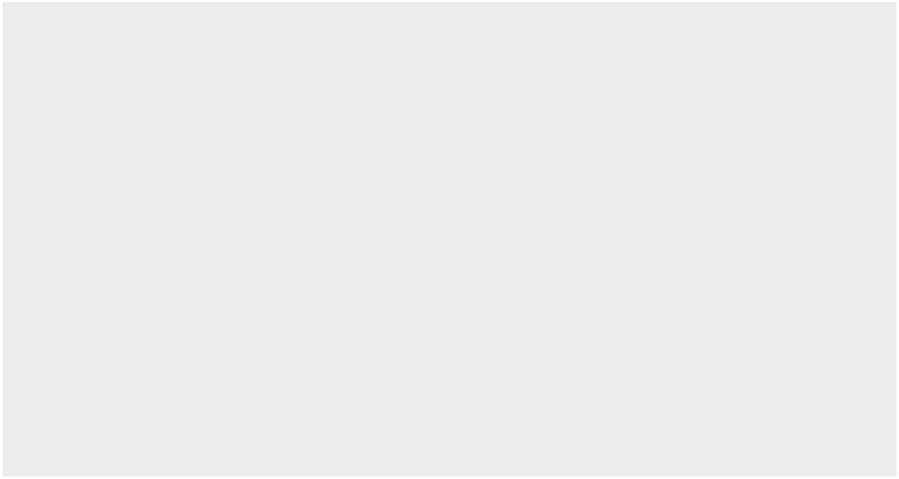


```
plot(pl, wl, pch=25)
```

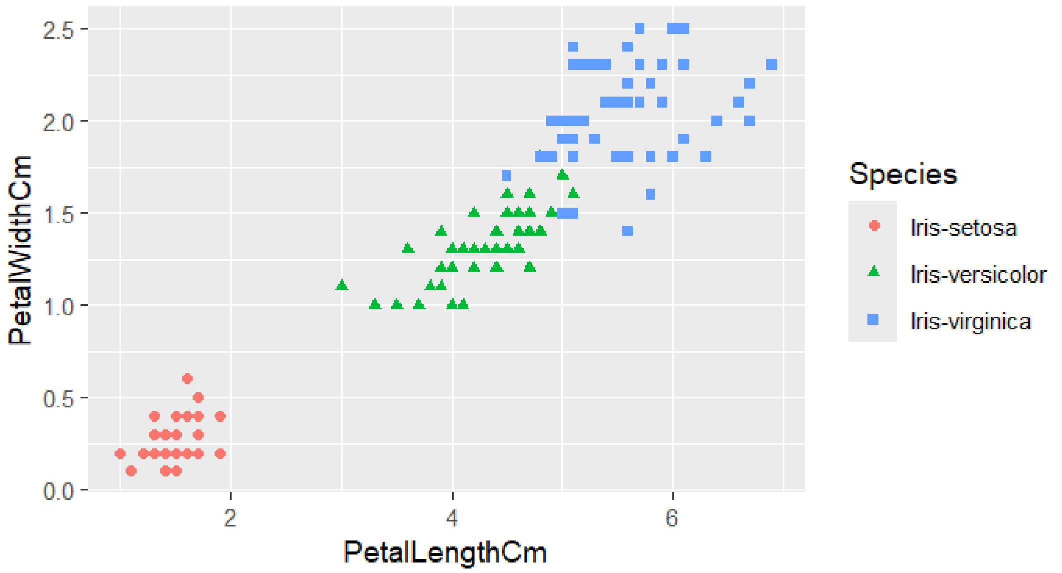
```
plot(pl, wl, pch=25, col="purple")
```



```
library(ggplot2)
ggplot(data=iris) #canvas creation
```



```
ggplot(data=iris) + aes(x=PetalLengthCm, y = PetalWidthCm) + geom_point(aes(color = Species, shape = Species))
```



Conclusion:

Data visualization using base R and ggplot2 effectively showcased relationships in datasets, improving data interpretation through clear graphical representation.