Shivani Manojkumar Panchiwala (1001982478)

Kuldip Rameshbhai Savaliya (1001832000)

Meghaben Ghanshyambhai Patel (1002006777)

R Documentation on Healthcare Stroke Dataset

Introduction

R is a programming language and software environment for statistical analysis, graphics representation and reporting. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for wide variety of operating systems like Linux, Windows, and Mac. This programming language was named as **R**, which is based on the first letter of first name of the two R authors (Robert Gentleman and Ross Ihaka). It is used to clean, analyze, and graph the data. It is widely used by researchers from various disciplines to estimate and show results and by professors of statistics and research methods. A group of packages known as tidy-verse, which can be considered as "dialect of the R language", is popular in the R ecosystem. It strives to offer a cohesive collection of functions to deal with data science tasks, including data import, cleaning, transformation, and visualization techniques.

The motive of this report is to conduct analysis and study on the provided dataset using R programming language. R is largely used for statistical analysis and graphical representation in Data mining. We have used “healthcare_stroke_dataset.csv” for this report. In this dataset, there are various health parameters of a patient. Based on various parameters mentioned in the dataset, it can be used to predict the possibility of a patient to suffer a stroke. Aggregation, Filtering and Rank operations have been executed on the dataset to know certain trends. We have used “plotly” and “ggplot2” packages of R.

Content:

Contains the health information about People such as:

1. id - unique for everyone
2. date - The dataset contains data of all the people with different disorders
3. gender - male/female
4. age - age of the people
5. hypertension - binary digit (0,1) where 1 and 0 represent present and not present
6. heart_disease - binary digit (0,1) where 1 and 0 represent present and not present
7. ever_married - marital status
8. work_type - designation of different type of people.
9. Residence_type - urban/rural
10. avg_glucose_level - Numeric value which specifies the avg body glucose level.
11. bmi - Numeric value which indicates body mass index
12. smoking status - smokes/ formally smoked/never smoked

stroke - binary digit (0,1) where 1 and 0 represent present and not present

Retrieving the data

There are multiple options to import any dataset in R. We have used “read.csv()” function to fetch the data from our csv file.

Shivani Manojkumar Panchiwala (1001982478)

Kuldip Rameshbhai Savaliya (1001832000)

Meghaben Ghanshyambhai Patel (1002006777)

```
# Read the file
df<-read.csv("C:\\Users\\pshiv\\Downloads\\Assignment1 Sumer2022-1\\Assignment1 Sumer2022\\healthcare_stroke_dataset.csv", TRUE,
```

```
#return the first 5 rows of the dataset
head(df, 5)
```

id	date	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
9046	12/30/2020	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	8/18/2020	Female	61	0	0	Yes	Self-employed	Rural	202.21	NA	never smoked	1
31112	3/5/2020	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
60182	7/8/2020	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	6/5/2020	Female	79	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Glimpse Of Data

You need to install a package and then load it to be able to use it, packages give you access to more functions. Some packages we used and installed here are:

dplyr: is a package for making data manipulation easier. Packages in R are basically sets of additional functions that let you do more stuff in R. The functions we've been using, like str (), come built into R.

ggplot2: ggplot2 is a plotting package that makes it simple to create complex plots from data in a data frame. It provides a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties. Therefore, we only need minimal changes if the underlying data change or if we decide to change from a bar plot to a scatterplot. This helps in creating publication quality plots with minimal amounts of adjustments and tweaking.

Check For Missing Data

```
print('Display missing values in each column of dataframe:')
colSums(is.na(df))
print('Total missing values:')
sum(is.na(df))
```

```
[1] "Display missing values in each column of dataframe:"
```

id	0
date	0
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
bmi	201
smoking_status	0
stroke	0

```
[1] "Total missing values:"
201
```

Data Exploration

 # Task 1: Statistical Exploratory Data Analysis

```
#1-a Print the details of dataframe
print('Print the details of dataframe are:')
sprintf('No. of columns: %i', ncol(df))
sprintf('No. of rows: %i', nrow(df))
print('Column Names and their datatypes:')
sapply(df, class)
```

```
[1] "Print the details of dataframe are:"
'No. of columns: 13'
'No. of rows: 5110'
[1] "Column Names and their datatypes:"
```

id	'integer'
date	'factor'
gender	'factor'
age	'numeric'
hypertension	'integer'
heart_disease	'integer'
ever_married	'factor'
work_type	'factor'
Residence_type	'factor'
avg_glucose_level	'numeric'
bmi	'numeric'
smoking_status	'factor'
stroke	'integer'

Shivani Manojkumar Panchiwala (1001982478)

Kuldip Rameshbhai Savaliya (1001832000)

Meghaben Ghanshyambhai Patel (1002006777)

#1-a Print the details of dataframe

```
str(df)
```

```
'data.frame':   5110 obs. of  13 variables:
 $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
 $ date             : Factor w/ 366 levels "1/1/2020","1/10/2020",..: 116 315 179 304 270 245 299 4 238 259 ...
 $ gender           : Factor w/ 3 levels "Female","Male",..: 2 1 2 1 1 2 2 1 1 1 ...
 $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
 $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
 $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
 $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 2 ...
 $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5 4 4 4 4 4 ...
 $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2 ...
 $ avg_glucose_level: num  229 202 106 171 174 ...
 $ bmi              : num  36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2 ...
 $ smoking_status   : Factor w/ 4 levels "-","formerly smoked",..: 2 3 3 4 3 2 3 3 1 1 ...
 $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

#1-b Find the number of rows and columns in dataset

```
rows<-nrow(df)
cat("Number of rows:", rows)
col<-ncol(df)
cat("\nNumber of colums:", col)
```

```
Number of rows: 5110
Number of colums: 13
```

#1-c Print descriptive detail of a column in dataset

```
summary(df)
```

```
       id               date         gender          age        
 Min.   :   67   6/9/2020  :  27   Female:2994   Min.   : 0.08  
 1st Qu.:17741   5/9/2020  :  26   Male  :2115   1st Qu.:25.00  
 Median :36932   5/3/2020  :  25   Other :   1   Median :45.00  
 Mean   :36518   12/21/2020:  23                 Mean   :43.23  
 3rd Qu.:54682   4/7/2020  :  23                 3rd Qu.:61.00  
 Max.   :72940   8/14/2020 :  23                 Max.   :82.00  
                 (Other)   :4963                                
  hypertension     heart_disease     ever_married         work_type    
 Min.   :0.00000   Min.   :0.00000   No :1757     children     : 687  
 1st Qu.:0.00000   1st Qu.:0.00000   Yes:3353     Govt_job     : 657  
 Median :0.00000   Median :0.00000                Never_worked :  22  
 Mean   :0.09746   Mean   :0.05401                Private      :2925  
 3rd Qu.:0.00000   3rd Qu.:0.00000                Self-employed: 819  
 Max.   :1.00000   Max.   :1.00000                                    

 Residence_type avg_glucose_level      bmi                smoking_status 
 Rural:2514     Min.   : 55.12    Min.   :10.30   -              :1544  
 Urban:2596     1st Qu.: 77.25    1st Qu.:23.50   formerly smoked: 885  
                Median : 91.89    Median :28.10   never smoked   :1892  
                Mean   :106.15    Mean   :28.89   smokes         : 789  
                3rd Qu.:114.09    3rd Qu.:33.10                         
                Max.   :271.74    Max.   :97.60                         
                                  NA's   :201                           
     stroke        
 Min.   :0.00000   
```

```
1st Qu.:0.00000
Median :0.00000
Mean   :0.04873
3rd Qu.:0.00000
Max.   :1.00000
```

#1-d Find all the count of unique values for a 'avg_glucose_level' column in dataset

```
#1-d Find all the count of unique values for a 'avg_glucose_level' column in dataset
cat("No. of unique values of avg_glucose_level are:", length(unique(df[["avg_glucose_level"]])))
unique(df$avg_glucose_level)
#https://www.r-bloggers.com/2021/12/how-to-find-unique-values-in-r/#:~:text=Use%20the%20unique()%20
```

```
No. of unique values of avg_glucose_level are: 3979
```

```
228.69  202.21  105.92  171.23  174.12  186.21  70.09  94.39  76.15  58.57  80.43  120.46  104.51  219.84  2
221.29  89.22  217.08  193.94  233.29  228.7  208.3  102.87  104.12  100.98  189.84  195.23  211.78  212.08
84.03  219.72  74.63  92.62  60.91  78.03  71.22  144.9  90.9  213.03  243.58  109.78  107.26  99.33  58.09
124.13  197.54  196.71  59.32  237.75  194.99  180.93  185.17  74.9  61.94  93.72  104.72  113.01  221.58  1
```

#1-d Find all percentage of 'Residence_type' for all the values

```
#1-d Find all percentage of 'Residence_type' for all the values
proportions <- table(df$Residence_type)/length(df$Residence_type)
percentage <- proportions*100
print(percentage)
#https://stackoverflow.com/questions/42379751/how-do-i-find-the-pe
```

```
   Rural    Urban
49.19765 50.80235
```

Task 2: Aggregation & Filtering & Rank

#Task 2-a: Find out the gender with largest number of records

```
print('Task 2-a:')
library(dplyr)
freq <- max(table(df$gender))
category <- tail(names(sort(table(df$gender))),1)
sprintf('The gender with the largest no. of records is "%s" with %i records:', category, freq)
#https://stackoverflow.com/questions/12187187/how-to-retrieve-the-most-repeated-value-in-a-column
```

```
[1] "Task 2-a:"
```

```
Warning message:
"package 'dplyr' was built under R version 3.6.3"
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

'The gender with the largest no. of records is "Female" with 2994 records:'

Shivani Manojkumar Panchiwala (1001982478)
Kuldip Rameshbhai Savaliya (1001832000)
Meghaben Ghanshyambhai Patel (1002006777)

#Task 2-b: Find out the total number of Residence_type "Urban" who are Male

```
#Task 2-b: Find out the total number of Residence_type "Urban" who are Male
print('Task 2-b:')
pns <- nrow(df[df$Residence_type == "Urban" & df$gender == "Male",])
sprintf('The total number of Residence_type "Urban" who are Male: %i', pns)
```

```
[1] "Task 2-b:"

'The total number of Residence_type "Urban" who are Male: 1067'
```

 # Group by function for dataframe in R using pipe operator

#2-c 1 question #Find the top 10 ages with highest av_glucose_level

```
print('Task 2-c 1:')
task2_c <- df %>% arrange(-avg_glucose_level)
print('Top 10 ages with highest avg_glucose_level:')
task2_c <- task2_c[1:10,]
task2_c %>% summarize(age, avg_glucose_level)
```

```
[1] "Task 2-c 1:"
[1] "Top 10 ages with highest avg_glucose_level:"
```

age	avg_glucose_level
68	271.74
49	267.76
76	267.61
76	267.60
60	266.59
67	263.56
71	263.32
62	261.67
67	260.85
80	259.63

#2-d 2nd question top 10 ages with more number of strokes

```
print('Task 2-d:')
#task2_d %>% group_by(age) %>% summarize(stroke)
#task2_d <- task2_d[1:10,]
task2_d <- df %>% arrange(-stroke)
print('Top 10 ages with more number of strokes:',)
task2_d <- task2_d[1:10,]
task2_d %>% summarize(age, stroke)
```

```
[1] "Task 2-d:"
[1] "Top 10 ages with more number of strokes:"
```

age	stroke
67	1
61	1
80	1
49	1
79	1
81	1
74	1
69	1
59	1
78	1

##TASK 3: VISUALIZATION

#task 3-a

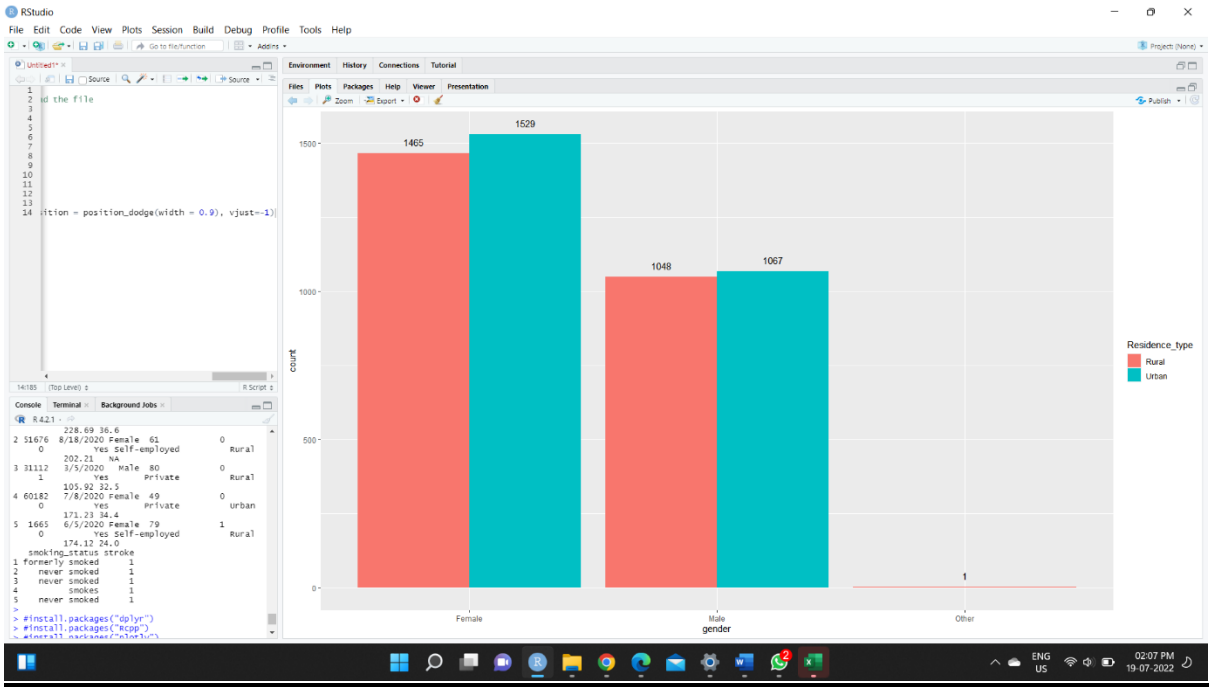
#Create barplot showing gender with count with residence type

```
print('Task 3-a:')
library(ggplot)
print('Gender count with residence type')
ggplot(data = df, aes(x=gender, fill = Residence_type)) + geom_bar(position = 'dodge') + geom_text(stat='count', aes(label=..cou
#https://intellipaat.com/community/16343/how-to-put-labels-over-geombar-for-each-bar-in-r-with-ggplot2
```

Shivani Manojkumar Panchiwala (1001982478)

Kuldip Rameshbhai Savaliya (1001832000)

Meghaben Ghanshyambhai Patel (1002006777)



#task 3-b

#Display pie chart for the smoking status data

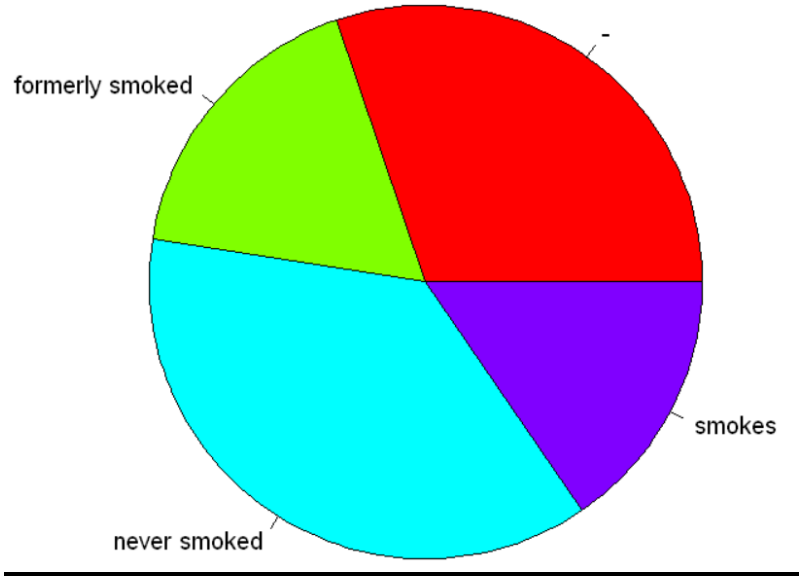
```
print('Task 3-b')
print('pie chart for the smoking status data:')
task_3_b<- data.frame(df %>% group_by(smoking_status) %>% summarise(number_of_rows = n(
task_3_b
task_3_b<- top_n(task_3_b,10)
pie(task_3_b$number_of_rows, labels = paste(task_3_b$smoking_status, sep = " "),
    col = rainbow(length(task_3_b$smoking_status)),
    main = "smoking_status")
```

```
[1] "Task 3-b"
[1] "pie chart for the smoking status data:"
```

smoking_status	number_of_rows
-	1544
formerly smoked	885
never smoked	1892
smokes	789

```
Selecting by number_of_rows
```

Shivani Manojkumar Panchiwala (1001982478)
Kuldip Rameshbhai Savaliya (1001832000)
Meghaben Ghanshyambhai Patel (1002006777)



#Task4 finding an interesting pattern

 # atleast two visualization with explanation

#Task4 finding an interesting pattern

 # atleast two visualization with explanation

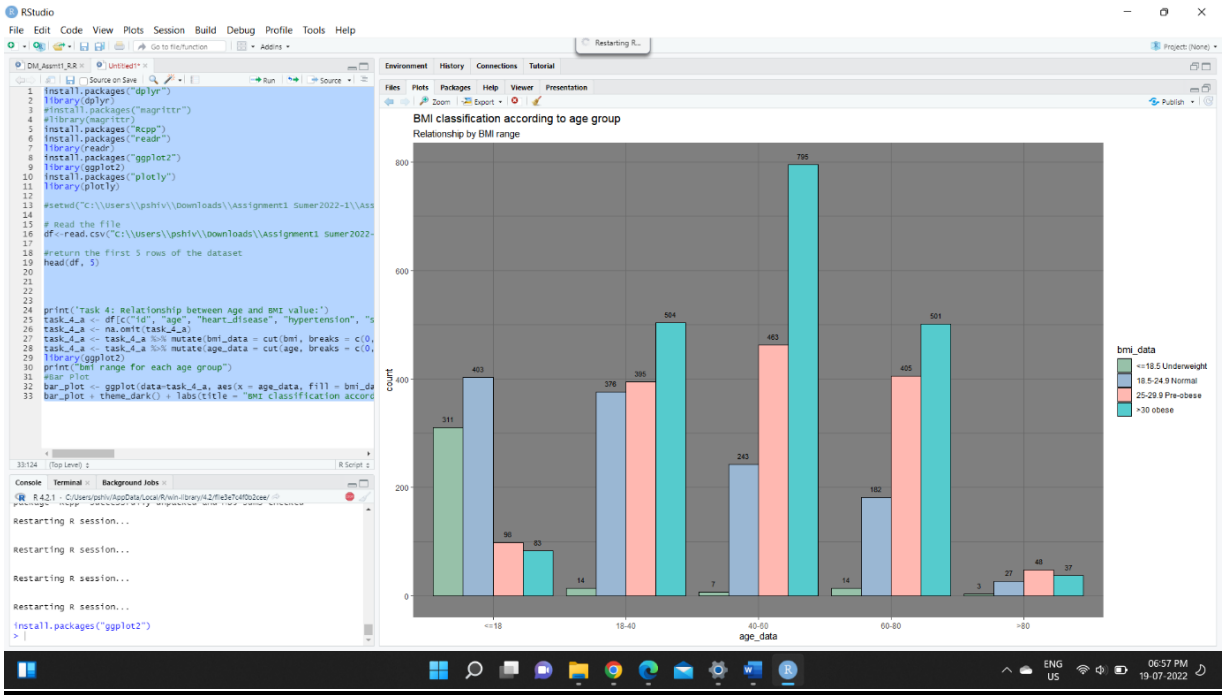
From the bar graph given below, we can see that there is a linear mild uphill relationship between "BMI" and "Age". Moreover, the children are majority in the area of "underweight" and "normal weight", while the adults (> 18) are majorly in the area of "pre-obese" and "obese". And from the scatter graph we can see that the minority of people are under weight and they are children.

```
# atleast two visualization with explanation
print('Task 4: Relationship between Age and BMI value:')
task_4_a <- df[c("id", "age", "heart_disease", "hypertension", "smoking_status", "bmi")]
task_4_a <- na.omit(task_4_a)
task_4_a <- task_4_a %>% mutate(bmi_data = cut(bmi, breaks = c(0,18.5,24.9,29.9, Inf), labels = c("<=18.5 Underweight", "18.5-24.
task_4_a <- task_4_a %>% mutate(age_data = cut(age, breaks = c(0,18,40,60,80, Inf), labels = c("<=18", "18-40", "40-60", "60-80",
library(ggplot2)
print("bmi range for each age group")
bar_plot <- ggplot(data=task_4_a, aes(x = age_data, fill = bmi_data, width = 1)) + geom_bar(position = 'dodge', color = "black")
bar_plot + theme_light() + labs(title = "BMI classification according to age group", subtitle = "Relationship by BMI range")
```

Shivani Manojkumar Panchiwala (1001982478)

Kuldip Rameshbhai Savaliya (1001832000)

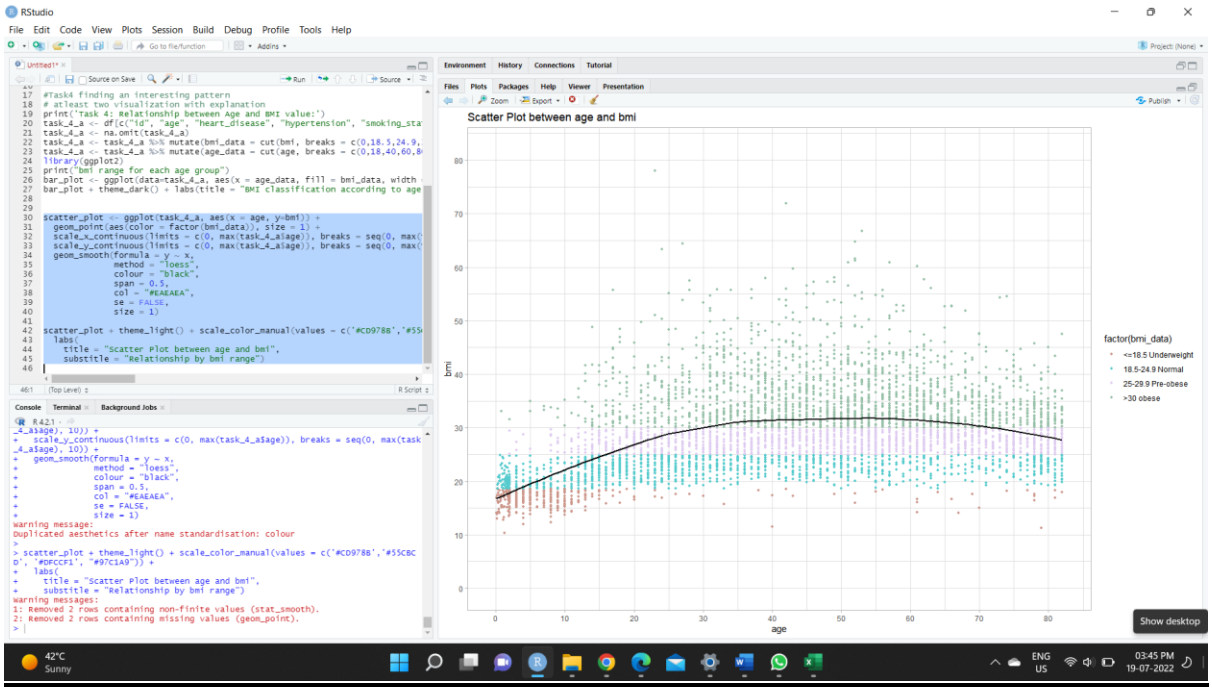
Meghaben Ghanshyambhai Patel (1002006777)



Scatter Plot Visualization

```
scatter_plot <- ggplot(task_4_a, aes(x = age, y=bmi)) +
  geom_point(aes(color = factor(bmi_data)), size = 1) +
  scale_x_continuous(limits = c(0, max(task_4_a$age)), breaks = seq(0, max(task_4_a$age), 10)) +
  scale_y_continuous(limits = c(0, max(task_4_a$age)), breaks = seq(0, max(task_4_a$age), 10)) +
  geom_smooth(formula = y ~ x,
              method = "loess",
              colour = "black",
              span = 0.5,
              col = "#EAEAEA",
              se = FALSE,
              size = 1)

scatter_plot + theme_light() + scale_color_manual(values = c('#CD978B','#55CBCD', '#DFCCF1', "#97C1A9")) +
  labs(
    title = "Scatter Plot between age and bmi",
    substitle = "Relationship by bmi range")
#https://www.guru99.com/r-scatter-plot-ggplot2.html
```



Here, we can see that there is a linear mild uphill relationship between “Age” and “Bmi”.

Also, the children are majority in the area of “Underweight” and “Normal Weight” while the adults are in majority in the area of “Pre-obese” and “Obese”.

New Strategies

I used Pipe operator in some tasks. The pipe operator, written as %>%, has been a longstanding feature of the magrittr package for R. It takes the output of one function and passes it into another function as an argument. This allows us to link a sequence of analysis steps.

Other Two Member’s Observation

Observed By Kuldip Savaliya (1001832000):

She did very well. She Properly explained us about R and implementation. Even she used some new strategies in some tasks. In this part, she implemented all the task which was assigned and created reported for that and mentioned description for everything.

Observed By Meghaben Patel (1002006777):

R implemented by Shivani Panchiwala. She performed very well. She gave her 100% in this assignment. She worked on every task and solved each and every query while she implemented this task. Even she explains us and consider our opinion also.