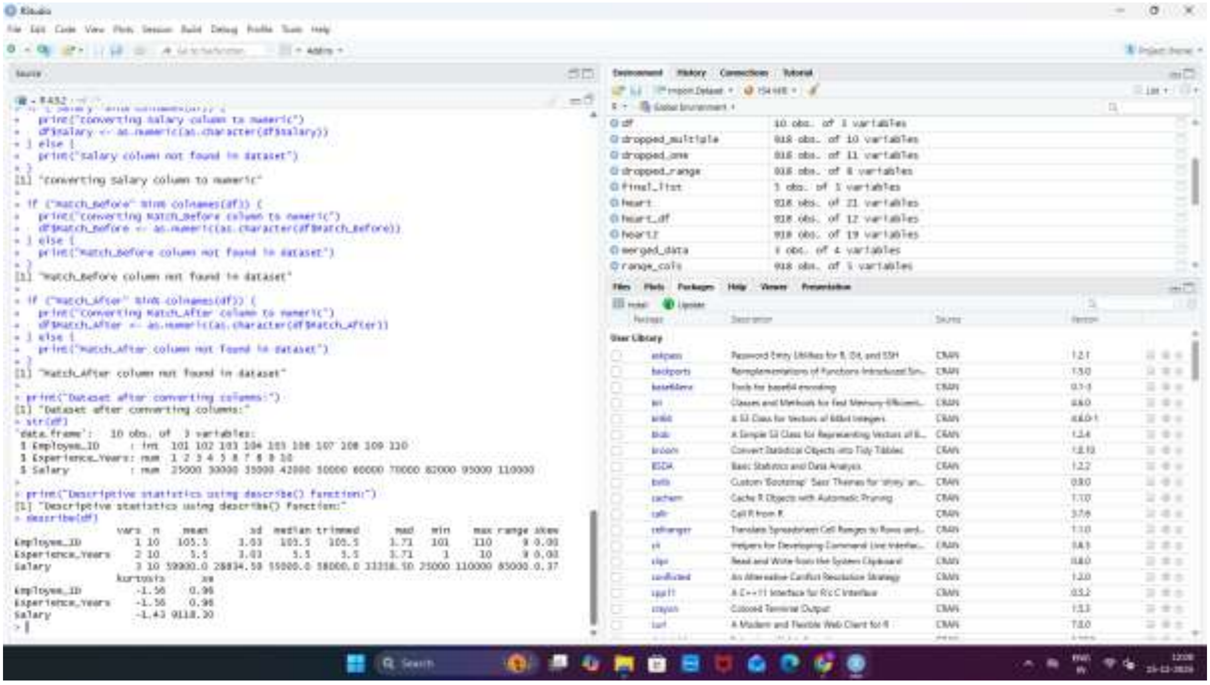
SETH L.U.J. AND SIR M.V COLLEGE

PRAC M2:1-6

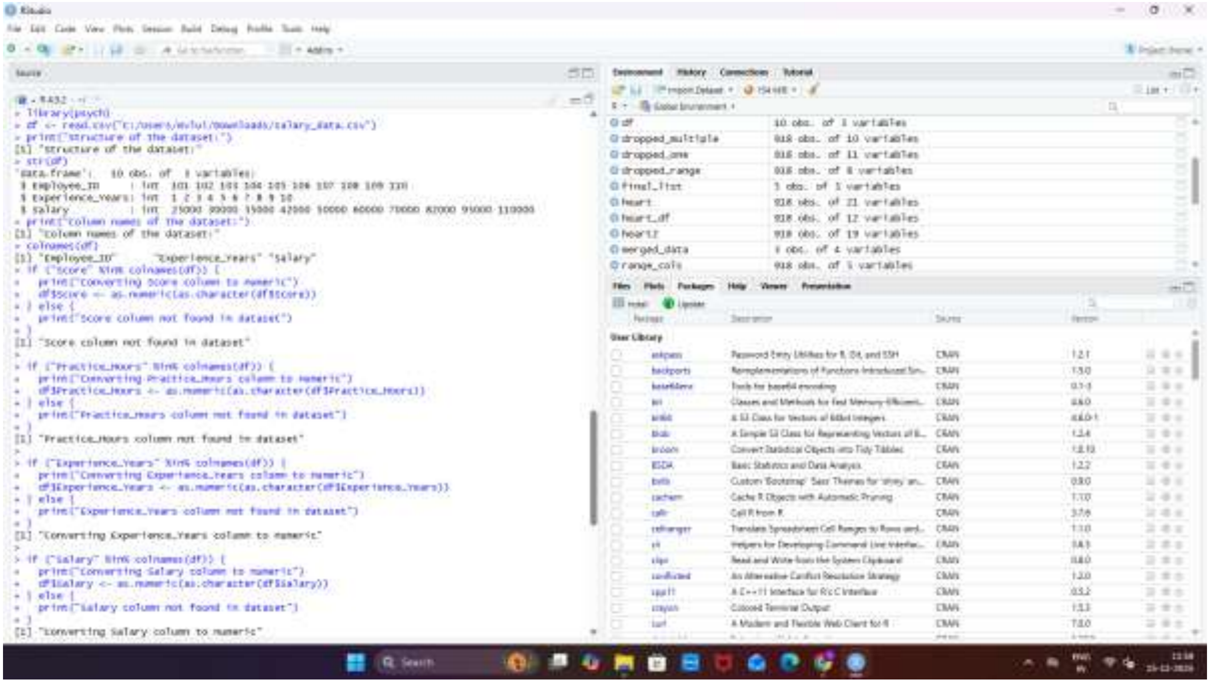
AIM:-

1]Generating Descriptive statistics using summary()or describe() (R)



```
+   print("Converting Salary column to numeric")
+   df$Salary <- as.numeric(as.character(df$Salary))
+ } else {
+   print("Salary column not found in dataset")
+ }
[1] "Converting Salary column to numeric"
>
> if ("Match_Before" %in% colnames(df)) {
+   print("Converting Match_Before column to numeric")
+   df$Match_Before <- as.numeric(as.character(df$Match_Before))
+ } else {
+   print("Match_Before column not found in dataset")
+ }
[1] "Match_Before column not found in dataset"
>
> if ("Match_After" %in% colnames(df)) {
+   print("Converting Match_After column to numeric")
+   df$Match_After <- as.numeric(as.character(df$Match_After))
+ } else {
+   print("Match_After column not found in dataset")
+ }
[1] "Match_After column not found in dataset"
>
> print("Dataset after converting columns:")
[1] "Dataset after converting columns:"
> str(df)
'data.frame':   10 obs. of  3 variables:
 $ Employee_ID     : int  101 102 103 104 105 106 107 108 109 110
 $ Experience_Years: num  1 2 3 4 5 6 7 8 9 10
 $ Salary          : num  25000 30000 35000 42000 50000 60000 70000 82000 95000 110000
>
> print("Descriptive statistics using describe() function:")
[1] "Descriptive statistics using describe() function:"
> describe(df)
                 vars  n    mean       sd  median trimmed      mad   min    max range skew
Employee_ID         1 10   105.5     3.03   105.5   105.5     3.71   101    110     9 0.00
Experience_Years    2 10     5.5     3.03     5.5     5.5     3.71     1     10     9 0.00
Salary              3 10 59900.0 28834.50 55000.0 58000.0 33358.50 25000 110000 85000 0.37
                 kurtosis      se
Employee_ID         -1.56    0.96
Experience_Years    -1.56    0.96
Salary              -1.43 9118.30
> |
```

```
df                 10 obs. of 3 variables
dropped_multiple   918 obs. of 10 variables
dropped_one        918 obs. of 11 variables
dropped_range      918 obs. of 8 variables
final_list         5 obs. of 1 variables
heart              918 obs. of 21 variables
heart_df           918 obs. of 12 variables
heart2             918 obs. of 19 variables
merged_data        4 obs. of 4 variables
range_cols         918 obs. of 5 variables
```

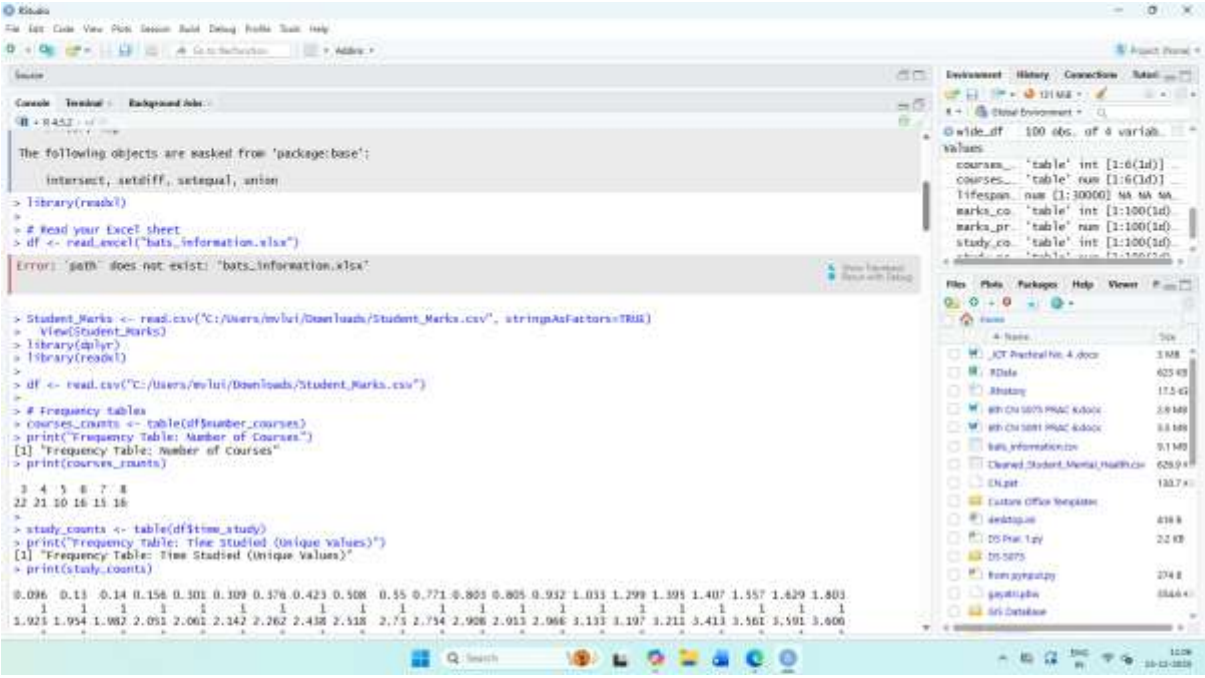


```
> library(psych)
> df <- read.csv("C:/Users/mvlul/Downloads/salary_data.csv")
> print("Structure of the dataset:")
[1] "Structure of the dataset:"
> str(df)
'data.frame':   10 obs. of  3 variables:
 $ Employee_ID     : int  101 102 103 104 105 106 107 108 109 110
 $ Experience_Years: int  1 2 3 4 5 6 7 8 9 10
 $ Salary          : int  25000 30000 35000 42000 50000 60000 70000 82000 95000 110000
> print("Column names of the dataset:")
[1] "Column names of the dataset:"
> colnames(df)
[1] "Employee_ID"      "Experience_Years" "Salary"
> if ("Score" %in% colnames(df)) {
+   print("Converting Score column to numeric")
+   df$Score <- as.numeric(as.character(df$Score))
+ } else {
+   print("Score column not found in dataset")
+ }
[1] "Score column not found in dataset"
>
> if ("Practice_Hours" %in% colnames(df)) {
+   print("Converting Practice_Hours column to numeric")
+   df$Practice_Hours <- as.numeric(as.character(df$Practice_Hours))
+ } else {
+   print("Practice_Hours column not found in dataset")
+ }
[1] "Practice_Hours column not found in dataset"
>
> if ("Experience_Years" %in% colnames(df)) {
+   print("Converting Experience_Years column to numeric")
+   df$Experience_Years <- as.numeric(as.character(df$Experience_Years))
+ } else {
+   print("Experience_Years column not found in dataset")
+ }
[1] "Converting Experience_Years column to numeric"
>
> if ("Salary" %in% colnames(df)) {
+   print("Converting Salary column to numeric")
+   df$Salary <- as.numeric(as.character(df$Salary))
+ } else {
+   print("Salary column not found in dataset")
+ }
[1] "Converting Salary column to numeric"
```

```
df                 10 obs. of 3 variables
dropped_multiple   918 obs. of 10 variables
dropped_one        918 obs. of 11 variables
dropped_range      918 obs. of 8 variables
final_list         5 obs. of 1 variables
heart              918 obs. of 21 variables
heart_df           918 obs. of 12 variables
heart2             918 obs. of 19 variables
merged_data        4 obs. of 4 variables
range_cols         918 obs. of 5 variables
```

Pratik Mali
S091

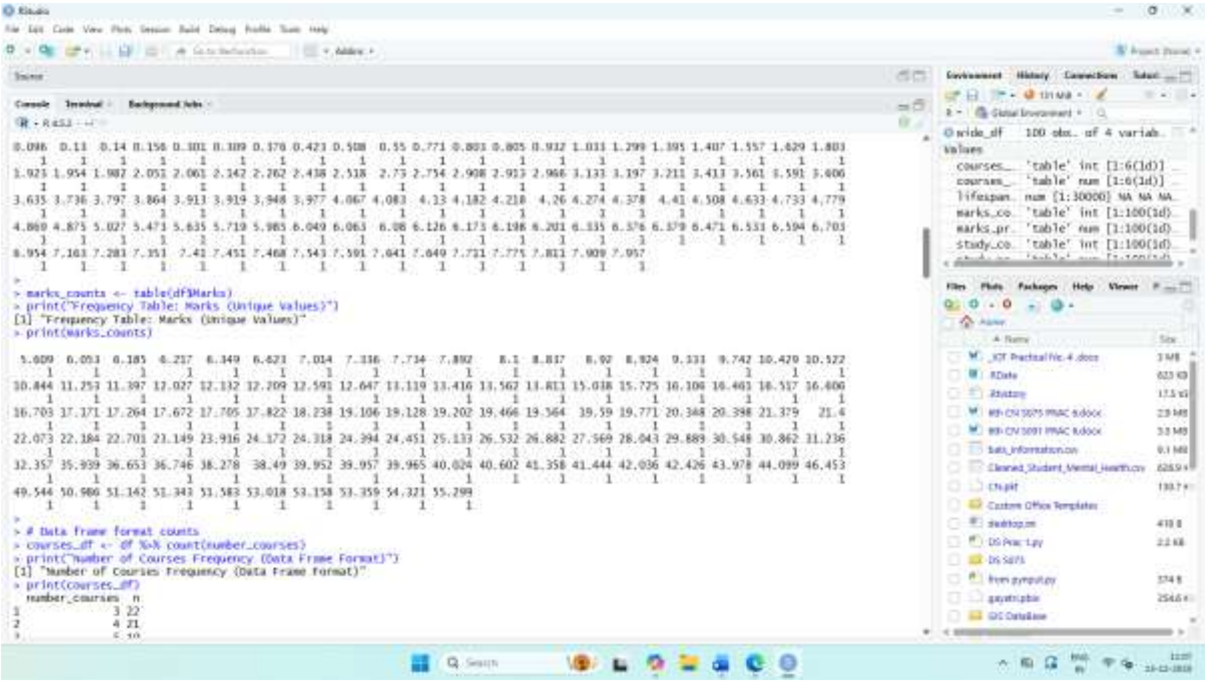
2]Generating Frequency tables using table() or count() or describe() (R)



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

    intersect, setdiff, setequal, union

> library(readxl)
>
> # Read your Excel sheet
> df <- read_excel("bats_information.xlsx")
Error: `path` does not exist: 'bats_information.xlsx'

> Student_Marks <- read.csv("C:/Users/mvlui/Downloads/Student_Marks.csv", stringsAsFactors=TRUE)
>   View(Student_Marks)
> library(dplyr)
> library(readxl)
>
> df <- read.csv("C:/Users/mvlui/Downloads/Student_Marks.csv")
>
> # Frequency tables
> courses_counts <- table(df$number_courses)
> print("Frequency Table: Number of Courses")
[1] "Frequency Table: Number of Courses"
> print(courses_counts)

 3  4  5  6  7  8 
22 21 10 16 15 16 
>
> study_counts <- table(df$time_study)
> print("Frequency Table: Time Studied (Unique Values)")
[1] "Frequency Table: Time Studied (Unique Values)"
> print(study_counts)

0.096  0.11  0.14 0.156 0.301 0.309 0.376 0.423 0.508  0.55 0.771 0.803 0.805 0.932 1.033 1.299 1.395 1.407 1.557 1.629 1.803 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
1.923 1.954 1.982 2.051 2.061 2.142 2.262 2.438 2.518  2.73 2.754 2.908 2.913 2.966 3.133 3.197 3.211 3.413 3.561 3.591 3.606 
```

```
wide_df     100 obs. of 4 variab...
Values
  courses_...  'table' int [1:6(1d)] ...
  courses_...  'table' num [1:6(1d)] ...
  lifespan...  num [1:30000] NA NA NA...
  marks_co...  'table' int [1:100(1d)...
  marks_pr...  'table' num [1:100(1d)...
  study_co...  'table' int [1:100(1d)...
```

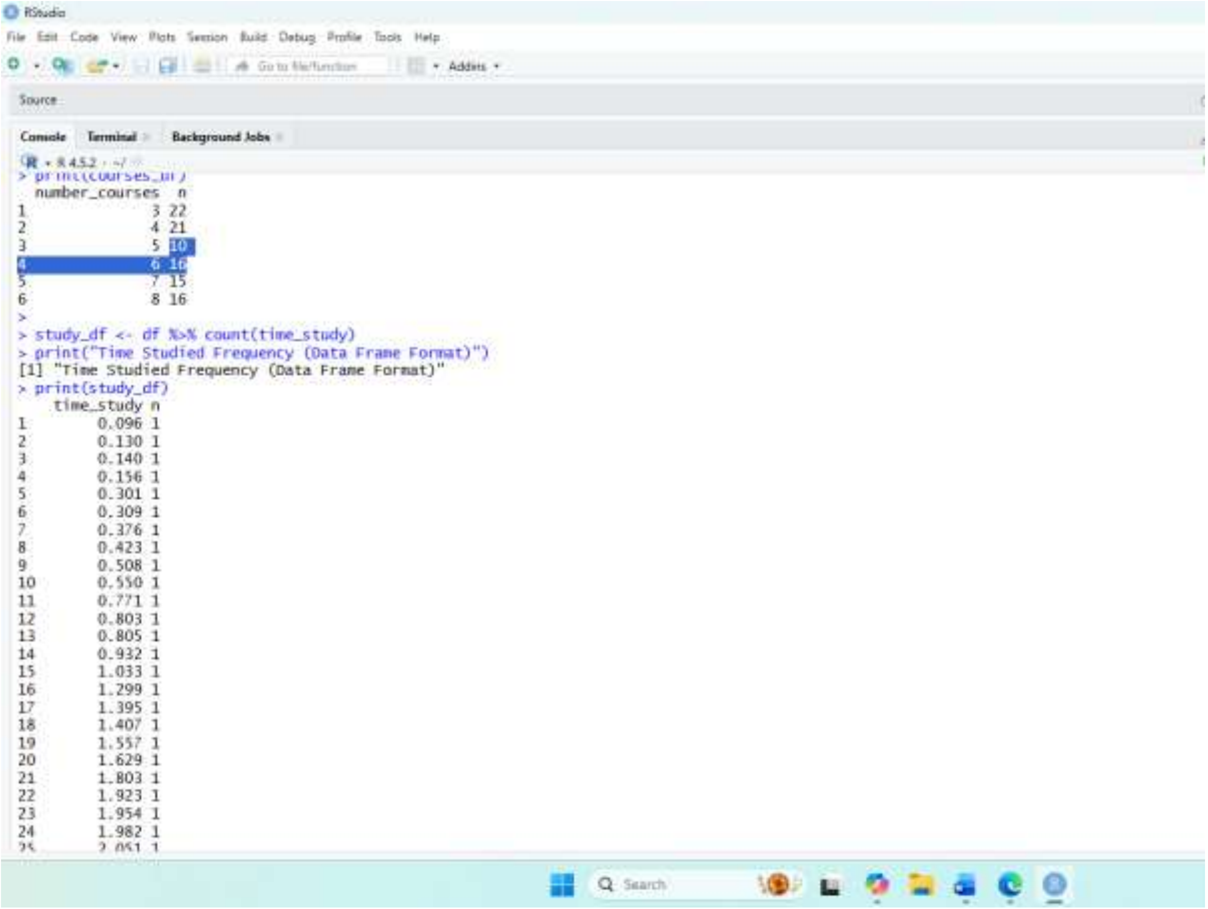
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Source
Console Terminal Background Jobs
R 4.5.2
Environment History Connections Tutorial
Global Environment
Files Plots Packages Help Viewer
Home
Name Size
_ICT Practical No. 4.docx 3 MB
RData 625 KB
.Rhistory 17.5 KB
MSc CN 5075 PRAC 6.docx 3.6 MB
MSc CN 5091 PRAC 6.docx 3.5 MB
bats_information.csv 0.1 MB
Cleaned_Student_Mental_Health.csv 629.9 KB
CN.pkt 133.7 KB
Custom Office Templates
desktop.ini 416 B
DS Prac 1.py 2.2 KB
DS 5075
from pympul.py 274 B
gayatri.pbix 3544.6 KB
GIS Database
Search



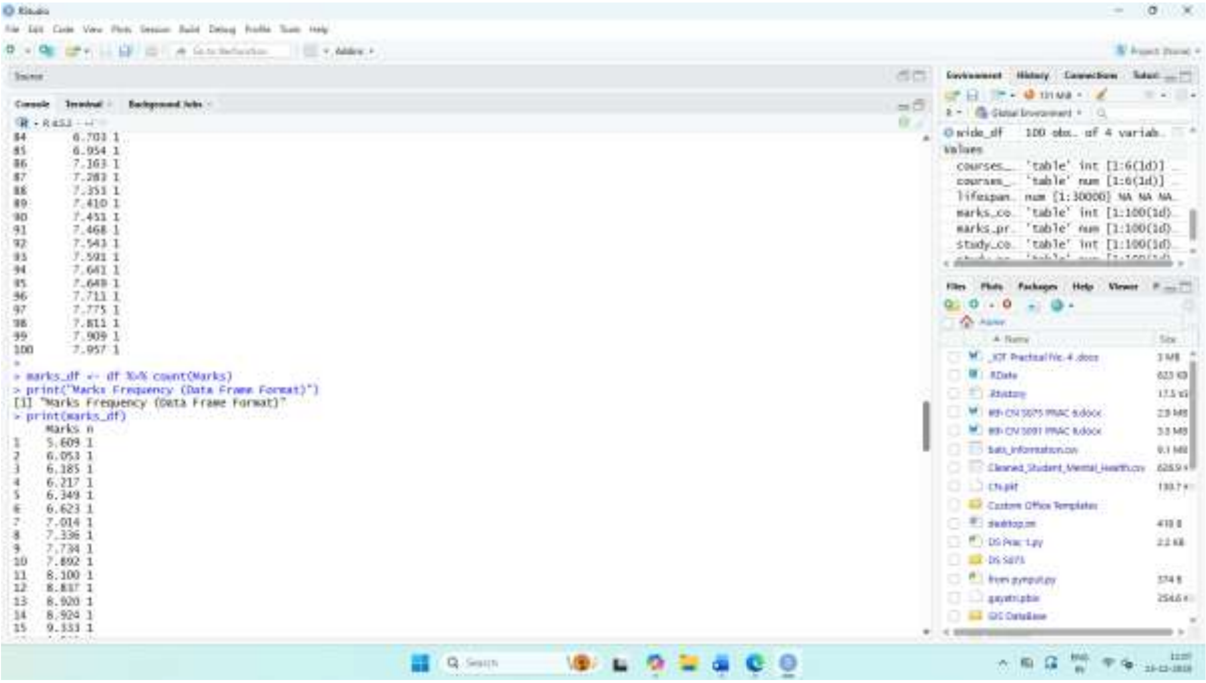
```
0.096  0.11  0.14 0.156 0.301 0.309 0.376 0.423 0.508  0.55 0.771 0.803 0.805 0.932 1.033 1.299 1.395 1.407 1.557 1.629 1.803 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
1.923 1.954 1.982 2.051 2.061 2.142 2.262 2.438 2.518  2.73 2.754 2.908 2.913 2.966 3.133 3.197 3.211 3.413 3.561 3.591 3.606 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
3.635 3.736 3.797 3.864 3.913 3.919 3.948 3.977 4.067 4.083  4.13 4.182 4.218  4.26 4.274 4.378  4.41 4.508 4.633 4.733 4.779 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
4.869 4.875 5.027 5.471 5.635 5.719 5.985 6.049 6.063  6.08 6.126 6.173 6.198 6.201 6.335 6.376 6.379 6.471 6.531 6.594 6.703 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
6.954 7.163 7.283 7.351  7.41 7.451 7.468 7.543 7.591 7.641 7.649 7.711 7.775 7.811 7.909 7.957 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
>
> marks_counts <- table(df$Marks)
> print("Frequency Table: Marks (Unique Values)")
[1] "Frequency Table: Marks (Unique Values)"
> print(marks_counts)

 5.609  6.053  6.185  6.217  6.349  6.623  7.014  7.336  7.734  7.892    8.1  8.817   8.92  8.924  9.333  9.742 10.429 10.522 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
10.844 11.253 11.397 12.027 12.132 12.209 12.591 12.647 13.119 13.416 13.562 13.811 15.038 15.725 16.106 16.461 16.517 16.606 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
16.703 17.171 17.264 17.672 17.705 17.822 18.238 19.106 19.128 19.202 19.466 19.564  19.59 19.771 20.348 20.398 21.379   21.4 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
22.073 22.184 22.701 23.149 23.916 24.172 24.318 24.394 24.451 25.133 26.532 26.882 27.569 28.043 29.889 30.548 30.862 31.236 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
32.357 35.939 36.653 36.746 38.278  38.49 39.952 39.957 39.965 40.024 40.602 41.358 41.444 42.036 42.426 43.978 44.099 46.453 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
49.544 50.986 51.142 51.343 51.583 53.018 53.158 53.359 54.321 55.299 
     1      1      1      1      1      1      1      1      1      1 
>
> # Data frame format counts
> courses_df <- df %>% count(number_courses)
> print("Number of Courses Frequency (Data Frame Format)")
[1] "Number of Courses Frequency (Data Frame Format)"
> print(courses_df)
  number_courses  n
1              3 22
2              4 21
3              5 10
```

```
wide_df     100 obs. of 4 variab...
Values
  courses_...  'table' int [1:6(1d)] ...
  courses_...  'table' num [1:6(1d)] ...
  lifespan...  num [1:30000] NA NA NA...
  marks_co...  'table' int [1:100(1d)...
  marks_pr...  'table' num [1:100(1d)...
  study_co...  'table' int [1:100(1d)...
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Source
Console Terminal Background Jobs
R 4.5.2
Environment History Connections Tutorial
Global Environment
Files Plots Packages Help Viewer
Home
Name Size
_ICT Practical No. 4.docx 3 MB
RData 623 KB
.Rhistory 17.5 KB
MSc CN 5075 PRAC 6.docx 2.9 MB
MSc CN 5091 PRAC 6.docx 3.5 MB
bats_information.csv 0.1 MB
Cleaned_Student_Mental_Health.csv 629.9 KB
CN.pkt 133.7 KB
Custom Office Templates
desktop.ini 416 B
DS Prac 1.py 2.2 KB
DS 5075
from pympul.py 374 B
gayatri.pbix 2544.6 KB
GIS Database
Search

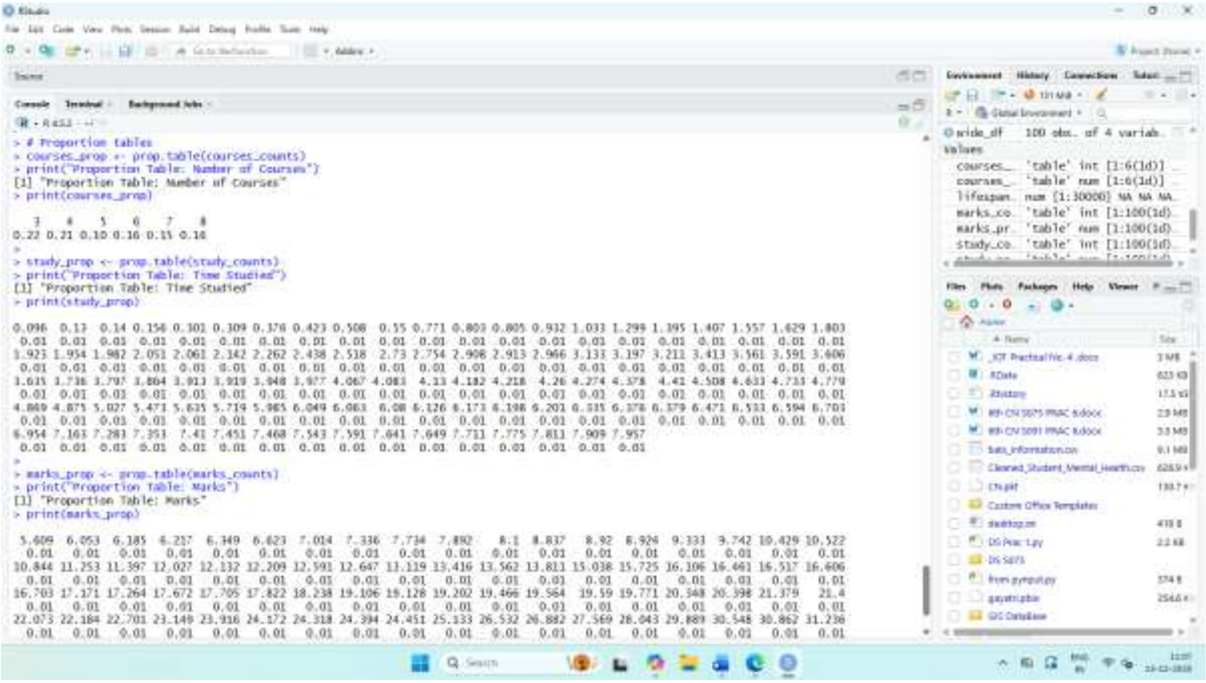


```
> print(courses_df)
  number_courses  n
1              3 22
2              4 21
3              5 10
4              6 16
5              7 15
6              8 16
>
> study_df <- df %>% count(time_study)
> print("Time Studied Frequency (Data Frame Format)")
[1] "Time Studied Frequency (Data Frame Format)"
> print(study_df)
    time_study n
1        0.096 1
2        0.130 1
3        0.140 1
4        0.156 1
5        0.301 1
6        0.309 1
7        0.376 1
8        0.423 1
9        0.508 1
10       0.550 1
11       0.771 1
12       0.803 1
13       0.805 1
14       0.932 1
15       1.033 1
16       1.299 1
17       1.395 1
18       1.407 1
19       1.557 1
20       1.629 1
21       1.803 1
22       1.923 1
23       1.954 1
24       1.982 1
25       2.051 1
```



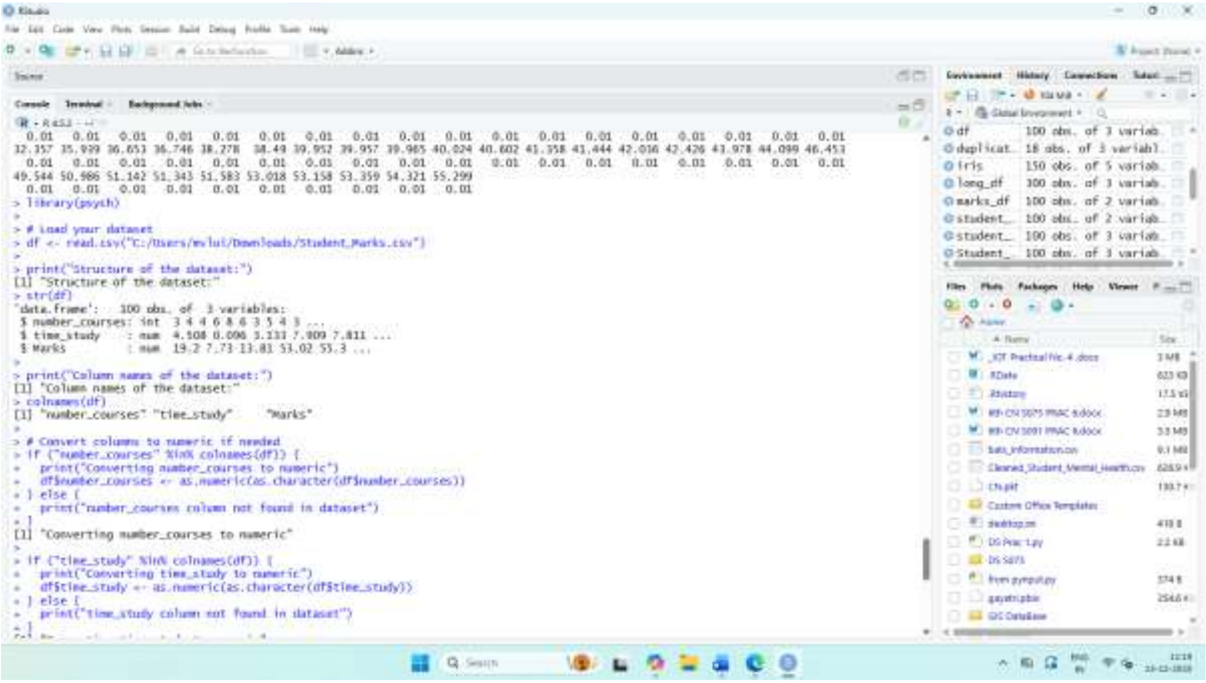
```
84         6.703 1
85         6.954 1
86         7.163 1
87         7.283 1
88         7.351 1
89         7.410 1
90         7.451 1
91         7.468 1
92         7.543 1
93         7.591 1
94         7.641 1
95         7.649 1
96         7.711 1
97         7.775 1
98         7.811 1
99         7.909 1
100        7.957 1
>
> marks_df <- df %>% count(Marks)
> print("Marks Frequency (Data Frame Format)")
[1] "Marks Frequency (Data Frame Format)"
> print(marks_df)
    Marks n
1   5.609 1
2   6.053 1
3   6.185 1
4   6.217 1
5   6.349 1
6   6.623 1
7   7.014 1
8   7.336 1
9   7.734 1
10  7.892 1
11  8.100 1
12  8.837 1
13  8.920 1
14  8.924 1
15  9.333 1
```

```
wide_df     100 obs. of 4 variab…
Values
  courses_…   'table' int [1:6(1d)] …
  courses_…   'table' num [1:6(1d)] …
  lifespan…   num [1:30000] NA NA NA…
  marks_co…   'table' int [1:100(1d)…
  marks_pr…   'table' num [1:100(1d)…
  study_co…   'table' int [1:100(1d)…
```

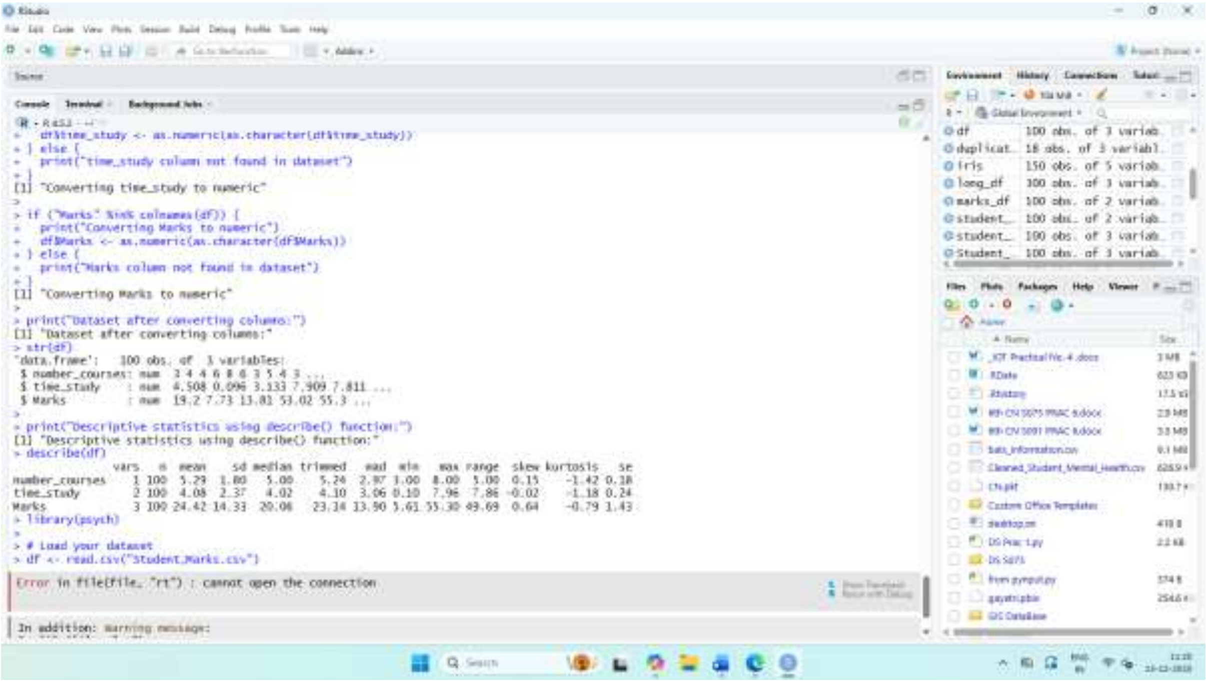


```
> # Proportion tables
> courses_prop <- prop.table(courses_counts)
> print("Proportion Table: Number of Courses")
[1] "Proportion Table: Number of Courses"
> print(courses_prop)

   3    4    5    6    7    8 
0.22 0.21 0.10 0.16 0.15 0.16 
>
> study_prop <- prop.table(study_counts)
> print("Proportion Table: Time Studied")
[1] "Proportion Table: Time Studied"
> print(study_prop)

0.096  0.13  0.14 0.156 0.301 0.309 0.376 0.423 0.508  0.55 0.771 0.803 0.805 0.932 1.033 1.299 1.395 1.407 1.557 1.629 1.803 
 0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01 
1.923 1.954 1.982 2.051 2.061 2.142 2.262 2.438 2.518  2.73 2.754 2.908 2.913 2.966 3.133 3.197 3.211 3.413 3.561 3.591 3.606 
 0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01 
3.635 3.736 3.797 3.864 3.913 3.919 3.948 3.977 4.067 4.083  4.13 4.182 4.218  4.26 4.274 4.378  4.41 4.508 4.633 4.733 4.779 
 0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01 
4.869 4.875 5.027 5.471 5.635 5.719 5.985 6.049 6.063  6.08 6.126 6.173 6.198 6.201 6.335 6.376 6.379 6.471 6.533 6.594 6.703 
 0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01 
6.954 7.163 7.283 7.353  7.41 7.451 7.468 7.543 7.591 7.641 7.649 7.711 7.775 7.811 7.909 7.957 
 0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01  0.01 
>
> marks_prop <- prop.table(marks_counts)
> print("Proportion Table: Marks")
[1] "Proportion Table: Marks"
> print(marks_prop)

 5.609  6.053  6.185  6.217  6.349  6.623  7.014  7.336  7.734  7.892    8.1  8.837   8.92  8.924  9.333  9.742 10.429 10.522 
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
10.844 11.253 11.397 12.027 12.132 12.209 12.591 12.647 13.119 13.416 13.562 13.811 15.038 15.725 16.106 16.461 16.517 16.606 
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
16.703 17.171 17.264 17.672 17.705 17.822 18.238 19.106 19.128 19.202 19.466 19.564  19.59 19.771 20.348 20.398 21.379   21.4 
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
22.073 22.184 22.701 23.149 23.916 24.172 24.318 24.394 24.451 25.133 26.532 26.882 27.569 28.043 29.889 30.548 30.862 31.236 
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
```

3] Creating cross-tabulations and two-way tables using table() (R)

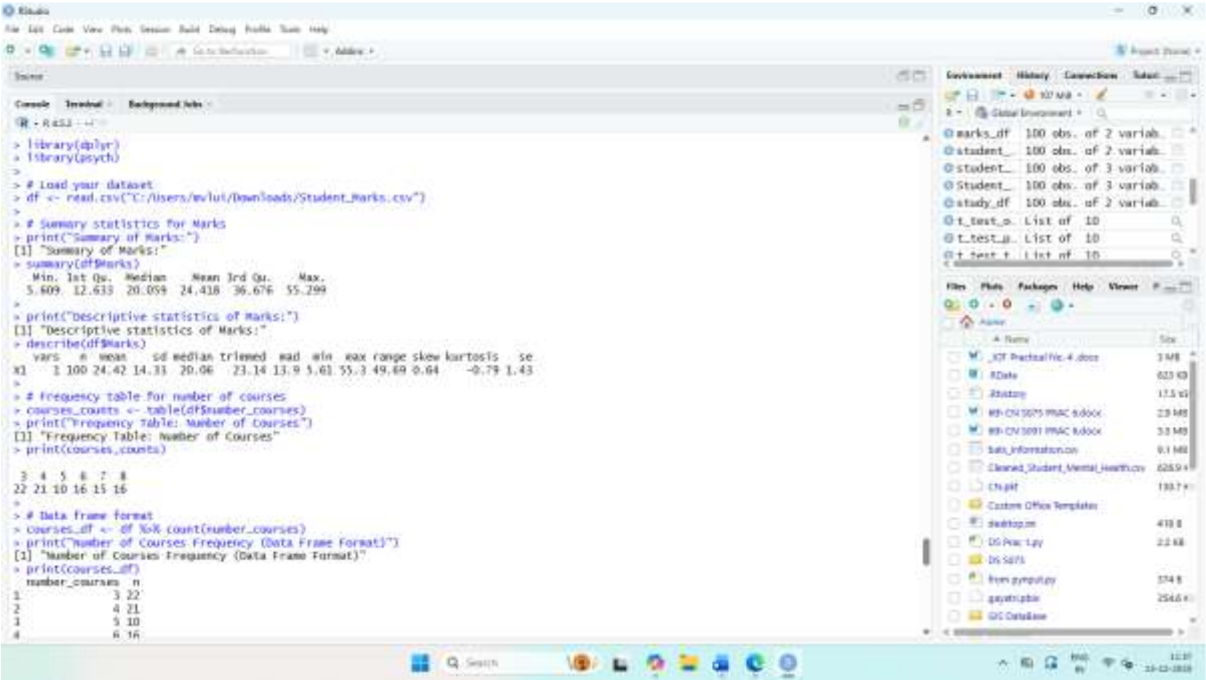


```
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
32.357 35.939 36.653 36.746 38.278  38.49 39.952 39.957 39.985 40.024 40.602 41.358 41.444 42.016 42.426 43.978 44.099 46.453 
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
49.544 50.986 51.142 51.343 51.583 53.018 53.158 53.359 54.321 55.299 
  0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01 
> library(psych)
>
> # Load your dataset
> df <- read.csv("C:/Users/mvlui/Downloads/Student_Marks.csv")
>
> print("Structure of the dataset:")
[1] "Structure of the dataset:"
> str(df)
'data.frame':	100 obs. of  3 variables:
 $ number_courses: int  3 4 4 6 8 6 3 5 4 3 ...
 $ time_study    : num  4.508 0.096 3.133 7.909 7.811 ...
 $ Marks         : num  19.2 7.73 13.81 53.02 55.3 ...
>
> print("Column names of the dataset:")
[1] "Column names of the dataset:"
> colnames(df)
[1] "number_courses" "time_study"     "Marks"
>
> # Convert columns to numeric if needed
> if ("number_courses" %in% colnames(df)) {
+   print("Converting number_courses to numeric")
+   df$number_courses <- as.numeric(as.character(df$number_courses))
+ } else {
+   print("number_courses column not found in dataset")
+ }
[1] "Converting number_courses to numeric"
>
> if ("time_study" %in% colnames(df)) {
+   print("Converting time_study to numeric")
+   df$time_study <- as.numeric(as.character(df$time_study))
+ } else {
+   print("time_study column not found in dataset")
+ }
```

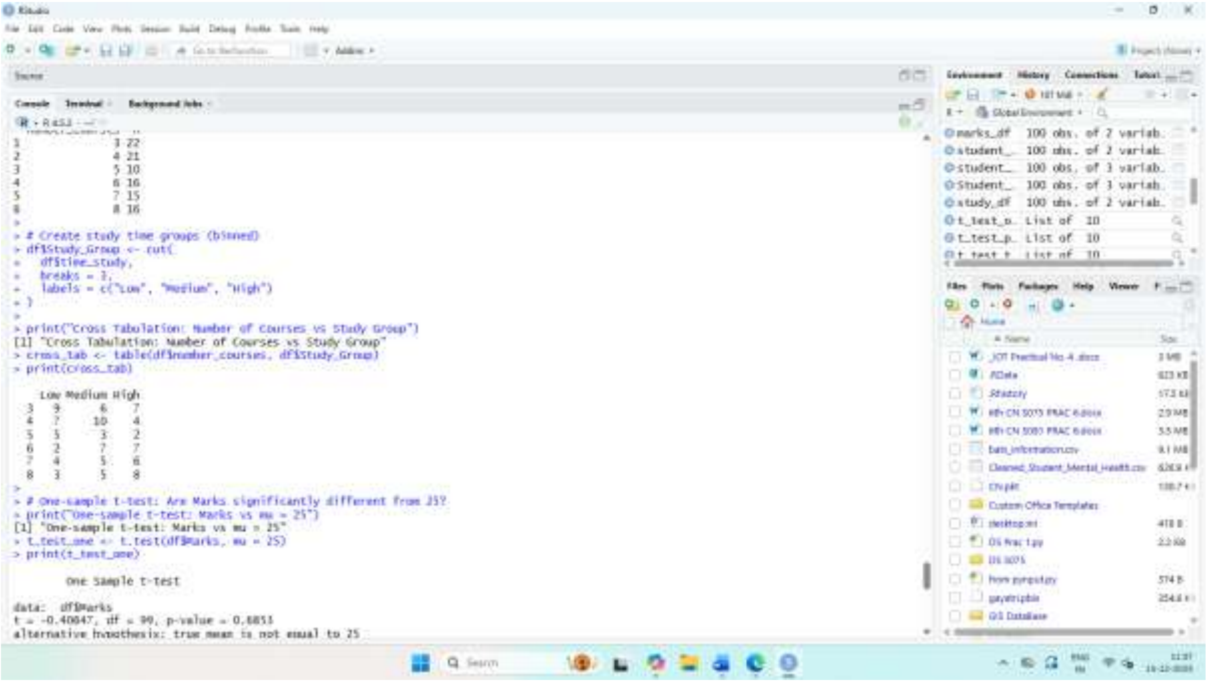


```
+   df$time_study <- as.numeric(as.character(df$time_study))
+ } else {
+   print("time_study column not found in dataset")
+ }
[1] "Converting time_study to numeric"
>
> if ("Marks" %in% colnames(df)) {
+   print("Converting Marks to numeric")
+   df$Marks <- as.numeric(as.character(df$Marks))
+ } else {
+   print("Marks column not found in dataset")
+ }
[1] "Converting Marks to numeric"
>
> print("Dataset after converting columns:")
[1] "Dataset after converting columns:"
> str(df)
'data.frame':   100 obs. of  3 variables:
 $ number_courses: num  3 4 4 6 8 6 3 5 4 3 ...
 $ time_study    : num  4.508 0.096 3.133 7.909 7.811 ...
 $ Marks         : num  19.2 7.73 13.81 53.02 55.3 ...
>
> print("Descriptive statistics using describe() function:")
[1] "Descriptive statistics using describe() function:"
> describe(df)
               vars   n  mean    sd median trimmed   mad  min   max range  skew kurtosis   se
number_courses    1 100  5.29  1.80   5.00    5.24  2.97 3.00  8.00  5.00  0.15    -1.42 0.18
time_study        2 100  4.08  2.37   4.02    4.10  3.06 0.10  7.96  7.86 -0.02    -1.18 0.24
Marks             3 100 24.42 14.33  20.06   23.14 13.90 5.61 55.30 49.69  0.64    -0.79 1.43
> library(psych)
>
> # Load your dataset
> df <- read.csv("Student_Marks.csv")
Error in file(file, "rt") : cannot open the connection

In addition: Warning message:
```

4]Performing one-sample t-tsts using t.test() (R)

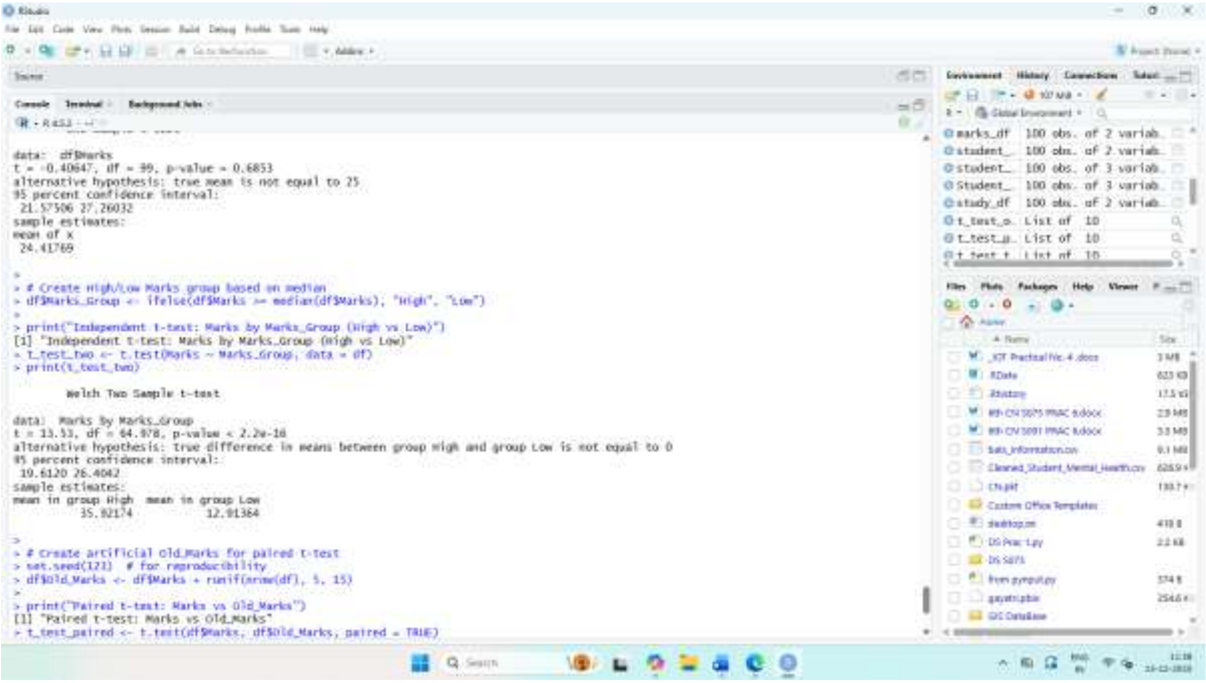


```
> library(dplyr)
> library(psych)
>
> # Load your dataset
> df <- read.csv("C:/Users/mvlui/Downloads/Student_Marks.csv")
>
> # Summary statistics for Marks
> print("Summary of Marks:")
[1] "Summary of Marks:"
> summary(df$Marks)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.609  12.633  20.059  24.418  36.676  55.299
>
> print("Descriptive statistics of Marks:")
[1] "Descriptive statistics of Marks:"
> describe(df$Marks)
   vars   n  mean    sd median trimmed  mad  min  max range skew kurtosis   se
X1    1 100 24.42 14.33  20.06   23.14 13.9 5.61 55.3 49.69 0.64    -0.79 1.43
>
> # Frequency table for number of courses
> courses_counts <- table(df$number_courses)
> print("Frequency Table: Number of Courses")
[1] "Frequency Table: Number of Courses"
> print(courses_counts)

 3  4  5  6  7  8
22 21 10 16 15 16
>
> # Data frame format
> courses_df <- df %>% count(number_courses)
> print("Number of Courses Frequency (Data Frame Format)")
[1] "Number of Courses Frequency (Data Frame Format)"
> print(courses_df)
  number_courses  n
1              3 22
2              4 21
3              5 10
4              6 16
```



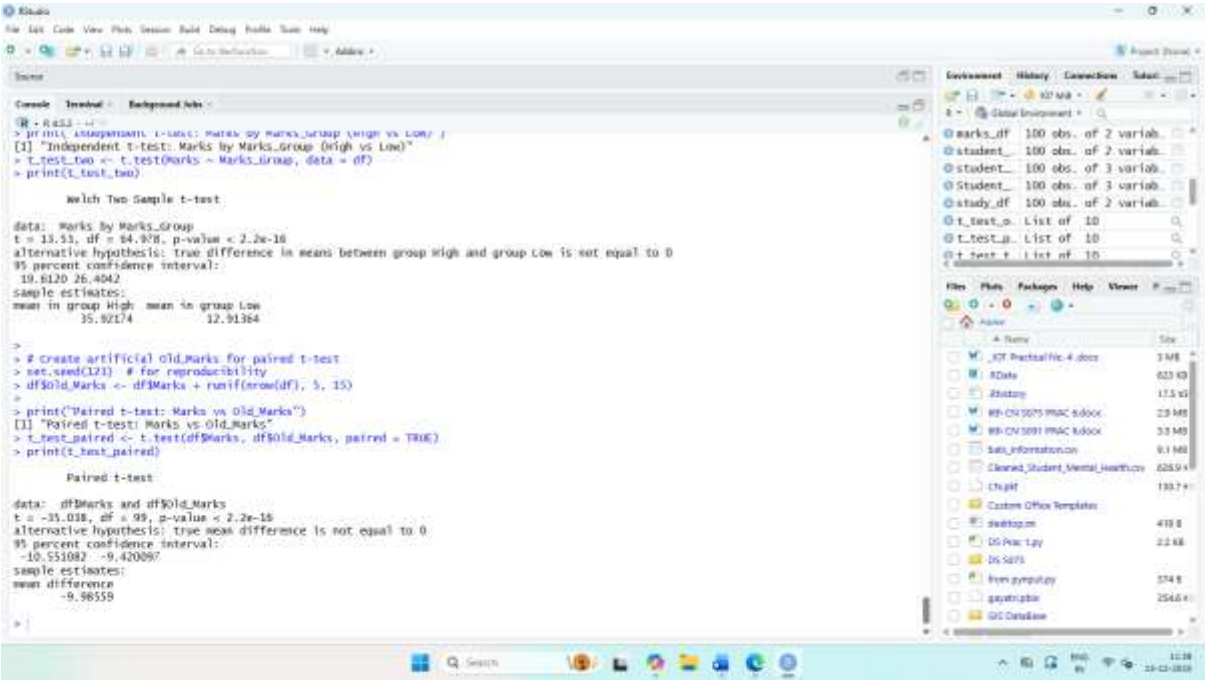
```
1                  3 22
2                  4 21
3                  5 10
4                  6 16
5                  7 15
6                  8 16
>
> # Create study time groups (binned)
> df$Study_Group <- cut(
+   df$time_study,
+   breaks = 3,
+   labels = c("Low", "Medium", "High")
+ )
>
> print("Cross Tabulation: Number of Courses vs Study Group")
[1] "Cross Tabulation: Number of Courses vs Study Group"
> cross_tab <- table(df$number_courses, df$Study_Group)
> print(cross_tab)

    Low Medium High
  3   9      6    7
  4   7     10    4
  5   5      3    2
  6   2      7    7
  7   4      5    6
  8   3      5    8
>
> # One-sample t-test: Are Marks significantly different from 25?
> print("One-sample t-test: Marks vs mu = 25")
[1] "One-sample t-test: Marks vs mu = 25"
> t_test_one <- t.test(df$Marks, mu = 25)
> print(t_test_one)

        One Sample t-test

data:  df$Marks
t = -0.40647, df = 99, p-value = 0.6853
alternative hypothesis: true mean is not equal to 25
```

marks_df 100 obs. of 2 variab.
student_ 100 obs. of 2 variab.
student_ 100 obs. of 3 variab.
Student_ 100 obs. of 3 variab.
study_df 100 obs. of 2 variab.
t_test_o. List of 10
t_test_p. List of 10
t_test_t. List of 10



```
data:  df$Marks
t = -0.40647, df = 99, p-value = 0.6853
alternative hypothesis: true mean is not equal to 25
95 percent confidence interval:
 21.57506 27.26032
sample estimates:
mean of x
 24.41769

>
> # Create High/Low Marks group based on median
> df$Marks_Group <- ifelse(df$Marks >= median(df$Marks), "High", "Low")
>
> print("Independent t-test: Marks by Marks_Group (High vs Low)")
[1] "Independent t-test: Marks by Marks_Group (High vs Low)"
> t_test_two <- t.test(Marks ~ Marks_Group, data = df)
> print(t_test_two)

        Welch Two Sample t-test

data:  Marks by Marks_Group
t = 13.53, df = 64.978, p-value < 2.2e-16
alternative hypothesis: true difference in means between group High and group Low is not equal to 0
95 percent confidence interval:
 19.6120 26.4042
sample estimates:
mean in group High  mean in group Low
          35.92174           12.91364

>
> # Create artificial Old_Marks for paired t-test
> set.seed(123)  # for reproducibility
> df$Old_Marks <- df$Marks + runif(nrow(df), 5, 15)
>
> print("Paired t-test: Marks vs Old_Marks")
[1] "Paired t-test: Marks vs Old_Marks"
> t_test_paired <- t.test(df$Marks, df$Old_Marks, paired = TRUE)
```

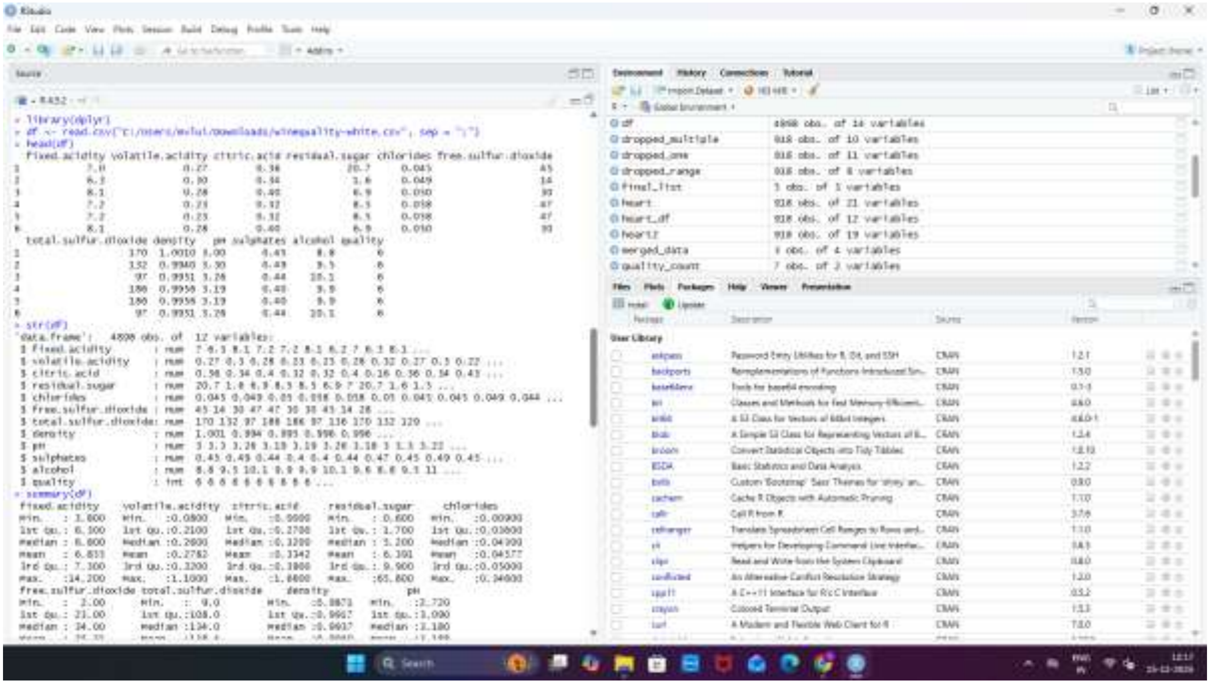
marks_df 100 obs. of 2 variab.
student_ 100 obs. of 2 variab.
student_ 100 obs. of 3 variab.
Student_ 100 obs. of 3 variab.
study_df 100 obs. of 2 variab.
t_test_o. List of 10
t_test_p. List of 10
t_test_t. List of 10



```
> print("Independent t-test: Marks by Marks_Group (High vs Low)")
[1] "Independent t-test: Marks by Marks_Group (High vs Low)"
> t_test_two <- t.test(Marks ~ Marks_Group, data = df)
> print(t_test_two)

        Welch Two Sample t-test

data:  Marks by Marks_Group
t = 13.53, df = 84.978, p-value < 2.2e-16
alternative hypothesis: true difference in means between group High and group Low is not equal to 0
95 percent confidence interval:
 19.6120 26.4042
sample estimates:
mean in group High  mean in group Low
          35.92174           12.91364


>
> # Create artificial Old_Marks for paired t-test
> set.seed(123)  # for reproducibility
> df$Old_Marks <- df$Marks + runif(nrow(df), 5, 15)
>
> print("Paired t-test: Marks vs Old_Marks")
[1] "Paired t-test: Marks vs Old_Marks"
> t_test_paired <- t.test(df$Marks, df$Old_Marks, paired = TRUE)
> print(t_test_paired)

        Paired t-test

data:  df$Marks and df$Old_Marks
t = -35.038, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -10.551082  -9.420097
sample estimates:
mean difference
       -9.98559

>
```

```
marks_df    100 obs. of 2 variab...
student_... 100 obs. of 2 variab...
student_... 100 obs. of 3 variab...
Student_... 100 obs. of 3 variab...
study_df    100 obs. of 2 variab...
t_test_o... List of 10
t_test_p... List of 10
t_test_t... List of 10
```

5]Performing independent two-sample t-tests using t-tests() with grouping (R).



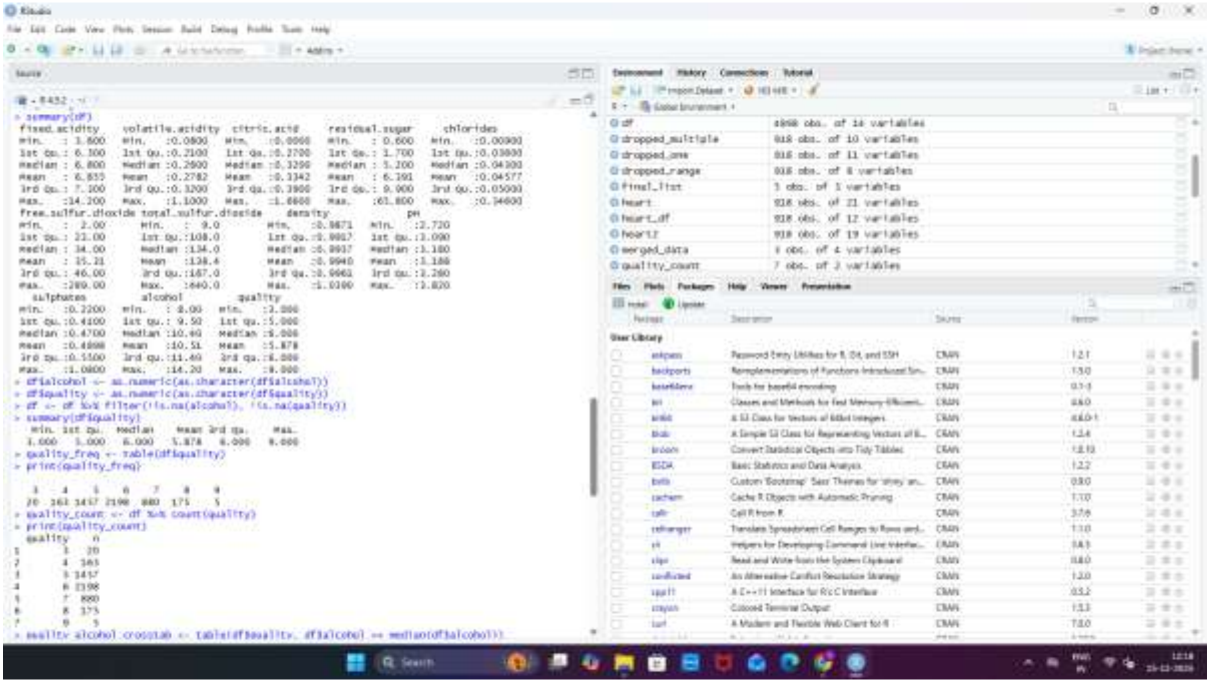
```
> library(dplyr)
> df <- read.csv("C:/Users/mulul/Downloads/winequality-white.csv", sep = ";")
> head(df)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
1           7.0             0.27        0.36           20.7     0.045                  45
2           6.3             0.30        0.34            1.6     0.049                  14
3           8.1             0.28        0.40            6.9     0.050                  30
4           7.2             0.23        0.32            8.5     0.058                  47
5           7.2             0.23        0.32            8.5     0.058                  47
6           8.1             0.28        0.40            6.9     0.050                  30
  total.sulfur.dioxide density   pH sulphates alcohol quality
1                  170  1.0010 3.00      0.45     8.8       6
2                  132  0.9940 3.30      0.49     9.5       6
3                   97  0.9951 3.26      0.44    10.1       6
4                  186  0.9956 3.19      0.40     9.9       6
5                  186  0.9956 3.19      0.40     9.9       6
6                   97  0.9951 3.26      0.44    10.1       6
> str(df)
'data.frame':   4898 obs. of  12 variables:
 $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
 $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
 $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
 $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
 $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
 $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
 $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
 $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
 $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
 $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
 $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
 $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
> summary(df)
 fixed.acidity    volatile.acidity  citric.acid     residual.sugar     chlorides
 Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600   Min.   :0.00900
 1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700   1st Qu.:0.03600
 Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200   Median :0.04300
 Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391   Mean   :0.04577
 3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900   3rd Qu.:0.05000
 Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800   Max.   :0.34600
 free.sulfur.dioxide total.sulfur.dioxide    density             pH
 Min.   :  2.00      Min.   :  9.0        Min.   :0.9871   Min.   :2.720
 1st Qu.: 23.00      1st Qu.:108.0        1st Qu.:0.9917   1st Qu.:3.090
 Median : 34.00      Median :134.0        Median :0.9937   Median :3.180
 Mean   : 35.31      Mean   :138.4        Mean   :0.9940   Mean   :3.188
```

```
df                4898 obs. of 14 variables
dropped_multiple  918 obs. of 10 variables
dropped_one       918 obs. of 11 variables
dropped_range     918 obs. of 8 variables
final_list        1 obs. of 1 variables
heart             918 obs. of 21 variables
heart_df          918 obs. of 12 variables
heart2            918 obs. of 19 variables
merged_data       1 obs. of 4 variables
quality_count     7 obs. of 2 variables
```

Package	Description	Source	Version
askpass	Password Entry Utilities for R, Git, and SSH	CRAN	1.2.1
backports	Reimplementations of Functions Introduced Sin...	CRAN	1.5.0
base64enc	Tools for base64 encoding	CRAN	0.1-3
bit	Classes and Methods for Fast Memory-Efficient...	CRAN	4.6.0
bit64	A S3 Class for Vectors of 64bit Integers	CRAN	4.6.0-1
blob	A Simple S3 Class for Representing Vectors of B...	CRAN	1.2.4
broom	Convert Statistical Objects into Tidy Tibbles	CRAN	1.0.10
BSDA	Basic Statistics and Data Analysis	CRAN	1.2.2
bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' an...	CRAN	0.9.0
cachem	Cache R Objects with Automatic Pruning	CRAN	1.1.0
callr	Call R from R	CRAN	3.7.6
cellranger	Translate Spreadsheet Cell Ranges to Rows and...	CRAN	1.1.0
cli	Helpers for Developing Command Line Interfac...	CRAN	3.6.5
clipr	Read and Write from the System Clipboard	CRAN	0.8.0
conflicted	An Alternative Conflict Resolution Strategy	CRAN	1.2.0
cpp11	A C++11 Interface for R's C Interface	CRAN	0.5.2
crayon	Colored Terminal Output	CRAN	1.5.3
curl	A Modern and Flexible Web Client for R	CRAN	7.0.0

SETH L.U.J. AND SIR M.V COLLEGE

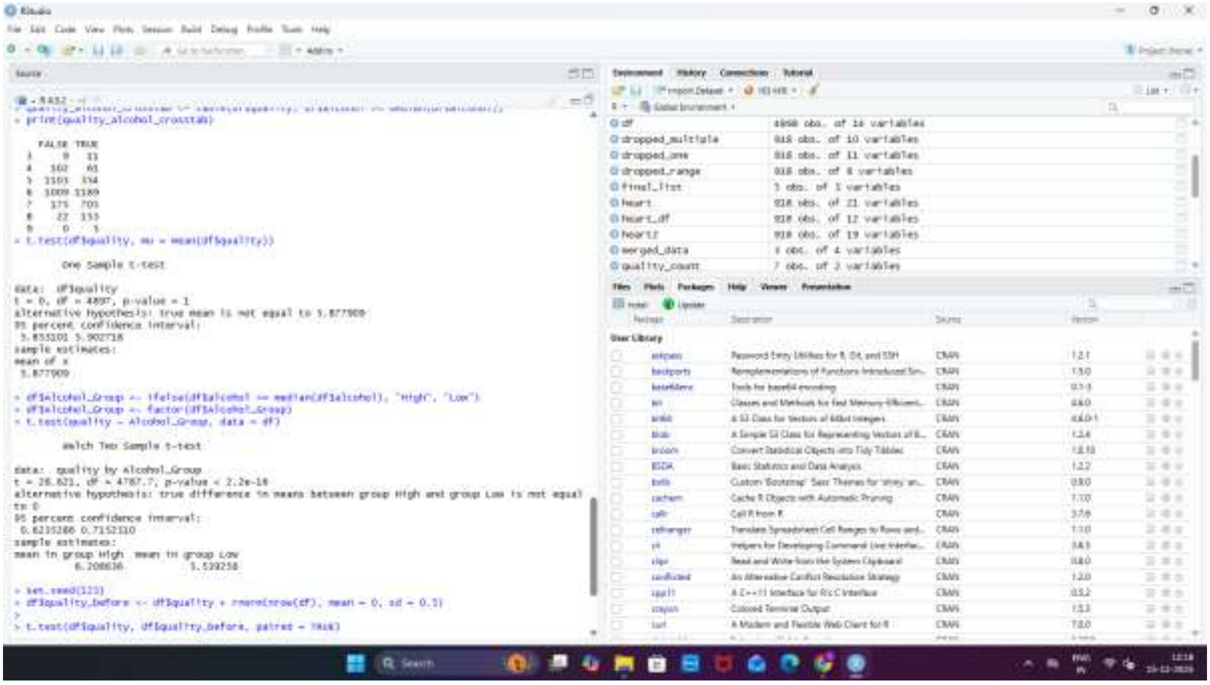
PRAC M2:1-6



```
> summary(df)
 fixed.acidity    volatile.acidity  citric.acid      residual.sugar     chlorides      
 Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600   Min.   :0.00900  
 1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700   1st Qu.:0.03600  
 Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200   Median :0.04300  
 Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391   Mean   :0.04577  
 3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900   3rd Qu.:0.05000  
 Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800   Max.   :0.34600  
 free.sulfur.dioxide total.sulfur.dioxide    density             pH       
 Min.   :  2.00      Min.   :  9.0        Min.   :0.9871   Min.   :2.720  
 1st Qu.: 23.00      1st Qu.:108.0        1st Qu.:0.9917   1st Qu.:3.090  
 Median : 34.00      Median :134.0        Median :0.9937   Median :3.180  
 Mean   : 35.31      Mean   :138.4        Mean   :0.9940   Mean   :3.188  
 3rd Qu.: 46.00      3rd Qu.:167.0        3rd Qu.:0.9961   3rd Qu.:3.280  
 Max.   :289.00      Max.   :440.0        Max.   :1.0390   Max.   :3.820  
   sulphates         alcohol         quality     
 Min.   :0.2200   Min.   : 8.00   Min.   :3.000  
 1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000  
 Median :0.4700   Median :10.40   Median :6.000  
 Mean   :0.4898   Mean   :10.51   Mean   :5.878  
 3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000  
 Max.   :1.0800   Max.   :14.20   Max.   :9.000  
> df$alcohol <- as.numeric(as.character(df$alcohol))
> df$quality <- as.numeric(as.character(df$quality))
> df <- df %>% filter(!is.na(alcohol), !is.na(quality))
> summary(df$quality)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  3.000   5.000   6.000   5.878   6.000   9.000 
> quality_freq <- table(df$quality)
> print(quality_freq)

   3    4    5    6    7    8    9 
  20  163 1457 2198  880  175    5 
> quality_count <- df %>% count(quality)
> print(quality_count)
  quality    n
1       3   20
2       4  163
3       5 1457
4       6 2198
5       7  880
6       8  175
7       9    5
> quality_alcohol_crosstab <- table(df$quality, df$alcohol >= median(df$alcohol))
```

df	4898 obs. of 14 variables
dropped_multiple	918 obs. of 10 variables
dropped_one	918 obs. of 11 variables
dropped_range	918 obs. of 8 variables
final_list	5 obs. of 1 variables
heart	918 obs. of 21 variables
heart_df	918 obs. of 12 variables
heart2	918 obs. of 19 variables
merged_data	4 obs. of 4 variables
quality_count	7 obs. of 2 variables

	Package	Description	Source	Version
User Library				
□	askpass	Password Entry Utilities for R, Git, and SSH	CRAN	1.2.1
□	backports	Reimplementations of Functions Introduced Sin...	CRAN	1.5.0
□	base64enc	Tools for base64 encoding	CRAN	0.1-3
□	bit	Classes and Methods for Fast Memory-Efficient...	CRAN	4.6.0
□	bit64	A S3 Class for Vectors of 64bit Integers	CRAN	4.6.0-1
□	blob	A Simple S3 Class for Representing Vectors of B...	CRAN	1.2.4
□	broom	Convert Statistical Objects into Tidy Tibbles	CRAN	1.0.10
□	BSDA	Basic Statistics and Data Analysis	CRAN	1.2.2
□	bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' an...	CRAN	0.9.0
□	cachem	Cache R Objects with Automatic Pruning	CRAN	1.1.0
□	callr	Call R from R	CRAN	3.7.6
□	cellranger	Translate Spreadsheet Cell Ranges to Rows and...	CRAN	1.1.0
□	cli	Helpers for Developing Command Line Interfac...	CRAN	3.6.5
□	clipr	Read and Write from the System Clipboard	CRAN	0.8.0
□	conflicted	An Alternative Conflict Resolution Strategy	CRAN	1.2.0
□	cpp11	A C++11 Interface for R's C Interface	CRAN	0.5.2
□	crayon	Colored Terminal Output	CRAN	1.5.3
□	curl	A Modern and Flexible Web Client for R	CRAN	7.0.0

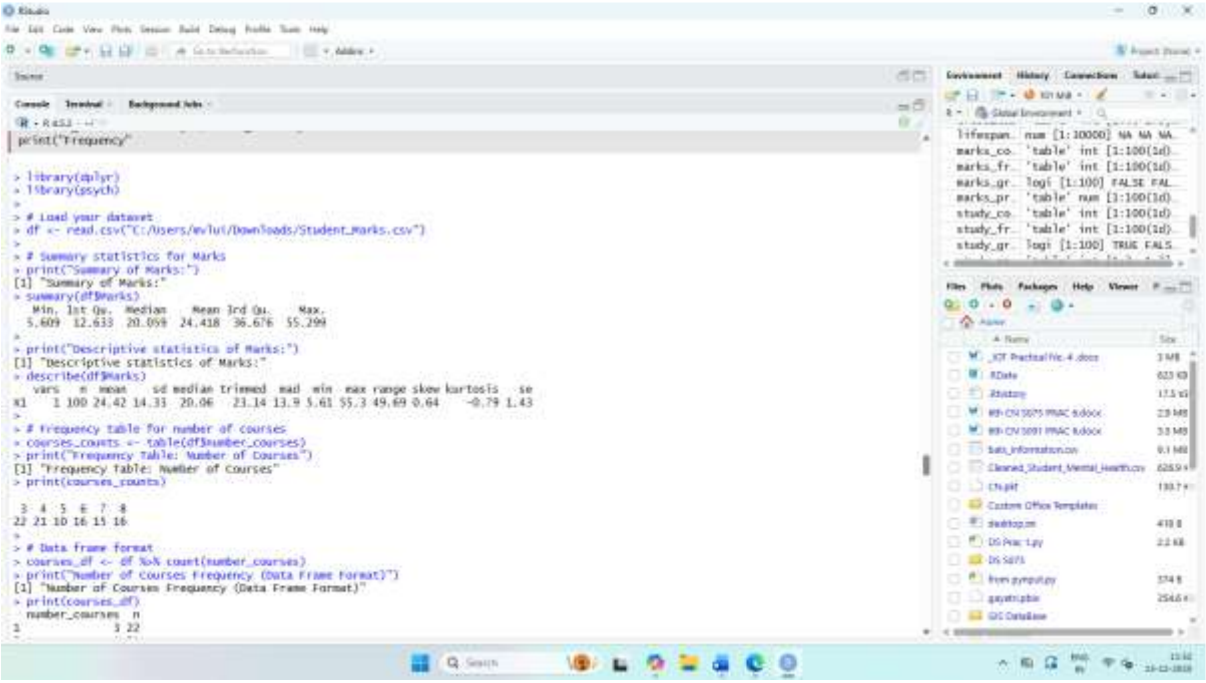


```
> print(quality_alcohol_crosstab)
   
    FALSE TRUE
  3     9   11
  4   102   61
  5  1103  354
  6  1009 1189
  7   175  705
  8    22  153
  9     0    5
> t.test(df$quality, mu = mean(df$quality))

	One Sample t-test

data:  df$quality
t = 0, df = 4897, p-value = 1
alternative hypothesis: true mean is not equal to 5.877909
95 percent confidence interval:
 5.853101 5.902718
sample estimates:
mean of x 
 5.877909 

> df$Alcohol_Group <- ifelse(df$alcohol >= median(df$alcohol), "High", "Low")
> df$Alcohol_Group <- factor(df$Alcohol_Group)
> t.test(quality ~ Alcohol_Group, data = df)

	Welch Two Sample t-test

data:  quality by Alcohol_Group
t = 28.621, df = 4787.7, p-value < 2.2e-16
alternative hypothesis: true difference in means between group High and group Low is not equal
 to 0
95 percent confidence interval:
 0.6235286 0.7152310
sample estimates:
mean in group High  mean in group Low 
          6.208636           5.539258 

> set.seed(123)
> df$quality_before <- df$quality + rnorm(nrow(df), mean = 0, sd = 0.5)
> 
> t.test(df$quality, df$quality_before, paired = TRUE)
```

df	4898 obs. of 14 variables
dropped_multiple	918 obs. of 10 variables
dropped_one	918 obs. of 11 variables
dropped_range	918 obs. of 8 variables
final_list	5 obs. of 1 variables
heart	918 obs. of 21 variables
heart_df	918 obs. of 12 variables
heart2	918 obs. of 19 variables
merged_data	4 obs. of 4 variables
quality_count	7 obs. of 2 variables

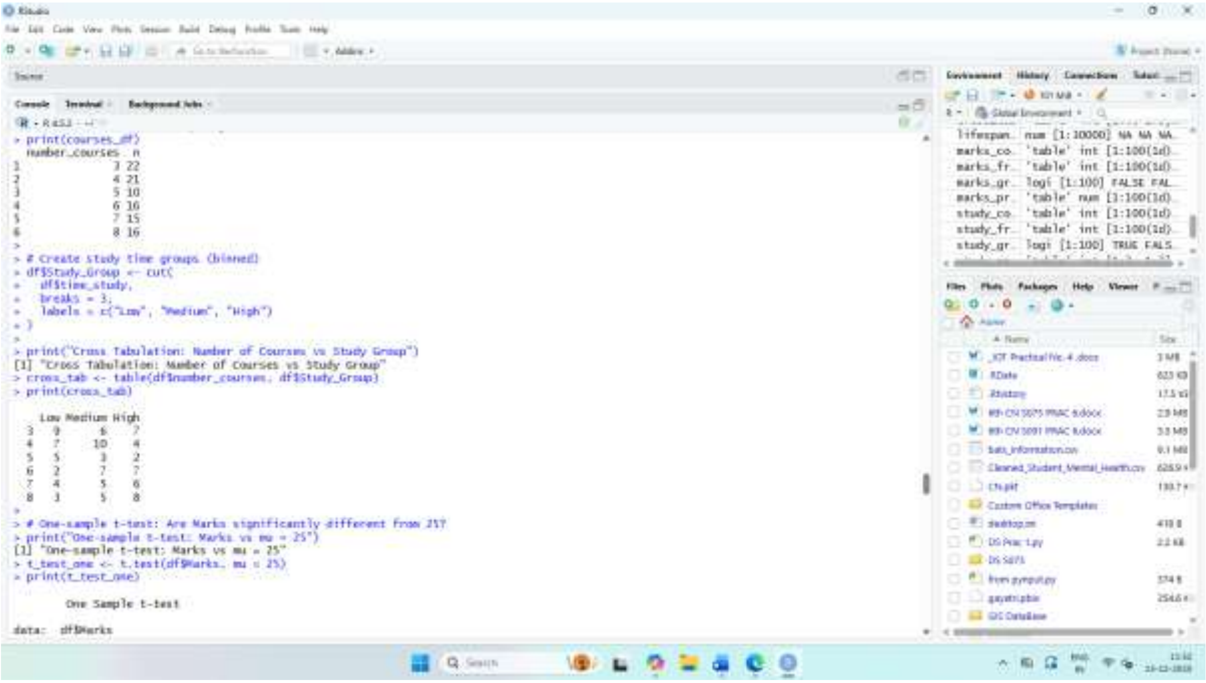
Pratik Mali
S091

6]Performing paired t-tests using t.test(paired=TRUE) (R).



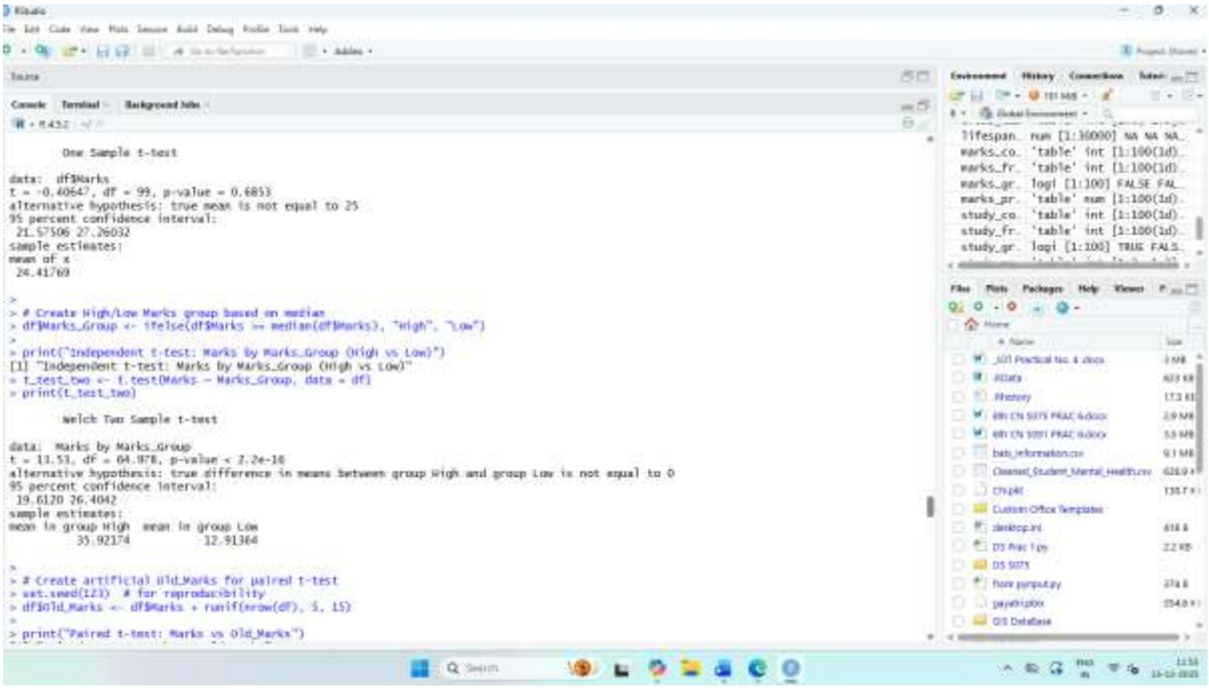
```
print("Frequency"

> library(dplyr)
> library(psych)
>
> # Load your dataset
> df <- read.csv("C:/Users/mvlui/Downloads/Student_Marks.csv")
>
> # Summary statistics for Marks
> print("Summary of Marks:")
[1] "Summary of Marks:"
> summary(df$Marks)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  5.609  12.633  20.059  24.418  36.676  55.299 
>
> print("Descriptive statistics of Marks:")
[1] "Descriptive statistics of Marks:"
> describe(df$Marks)
   vars   n  mean    sd median trimmed  mad  min  max range skew kurtosis   se
X1    1 100 24.42 14.33  20.06   23.14 13.9 5.61 55.3 49.69 0.64    -0.79 1.43
>
> # Frequency table for number of courses
> courses_counts <- table(df$number_courses)
> print("Frequency Table: Number of Courses")
[1] "Frequency Table: Number of Courses"
> print(courses_counts)

 3  4  5  6  7  8 
22 21 10 16 15 16 
>
> # Data frame format
> courses_df <- df %>% count(number_courses)
> print("Number of Courses Frequency (Data Frame Format)")
[1] "Number of Courses Frequency (Data Frame Format)"
> print(courses_df)
  number_courses  n
1              3 22
```

```
lifespan…  num [1:10000] NA NA NA…
marks_co…  'table' int [1:100(1d)…
marks_fr…  'table' int [1:100(1d)…
marks_gr…  logi [1:100] FALSE FAL…
marks_pr…  'table' num [1:100(1d)…
study_co…  'table' int [1:100(1d)…
study_fr…  'table' int [1:100(1d)…
study_gr…  logi [1:100] TRUE FALS…
```



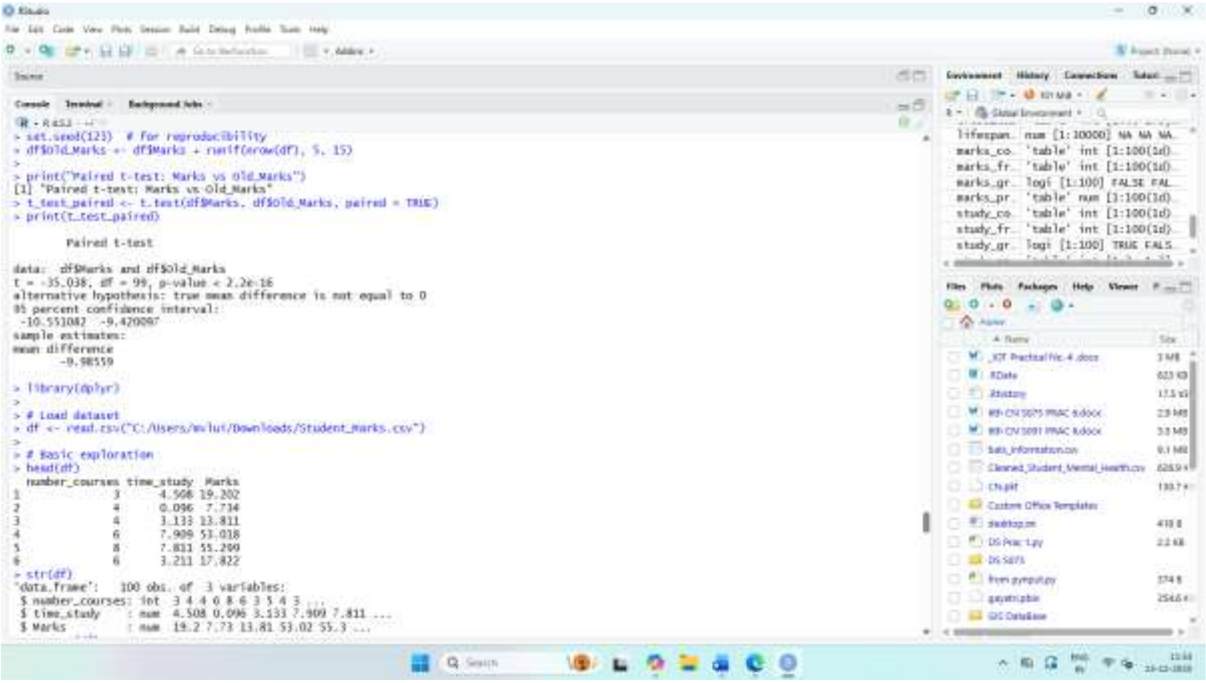
```
> print(courses_df)
  number_courses  n
1              3 22
2              4 21
3              5 10
4              6 16
5              7 15
6              8 16
>
> # Create study time groups (binned)
> df$Study_Group <- cut(
+   df$time_study,
+   breaks = 3,
+   labels = c("Low", "Medium", "High")
+ )
>
> print("Cross Tabulation: Number of Courses vs Study Group")
[1] "Cross Tabulation: Number of Courses vs Study Group"
> cross_tab <- table(df$number_courses, df$Study_Group)
> print(cross_tab)

    Low Medium High
  3   9      6    7
  4   7     10    4
  5   5      3    2
  6   2      7    7
  7   4      5    6
  8   3      5    8
>
> # One-sample t-test: Are Marks significantly different from 25?
> print("One-sample t-test: Marks vs mu = 25")
[1] "One-sample t-test: Marks vs mu = 25"
> t_test_one <- t.test(df$Marks, mu = 25)
> print(t_test_one)

        One Sample t-test

data:  df$Marks
```

```
lifespan_  num [1:10000] NA NA NA_
marks_co_  'table' int [1:100(1d)_
marks_fr_  'table' int [1:100(1d)_
marks_gr_  logi [1:100] FALSE FAL_
marks_pr_  'table' num [1:100(1d)_
study_co_  'table' int [1:100(1d)_
study_fr_  'table' int [1:100(1d)_
study_gr_  logi [1:100] TRUE FALS_
```

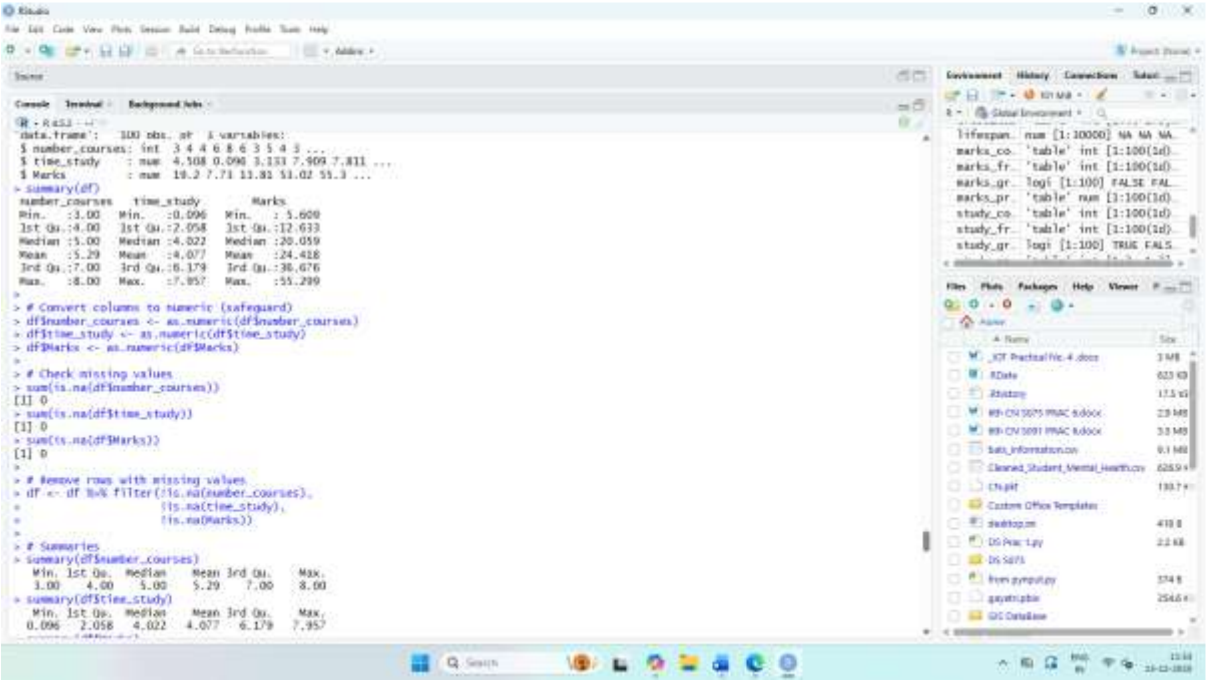


```
        One Sample t-test

data:  df$Marks
t = -0.40647, df = 99, p-value = 0.6853
alternative hypothesis: true mean is not equal to 25
95 percent confidence interval:
 21.57506 27.26032
sample estimates:
mean of x
 24.41769

>
> # Create High/Low Marks group based on median
> df$Marks_Group <- ifelse(df$Marks >= median(df$Marks), "High", "Low")
>
> print("Independent t-test: Marks by Marks_Group (High vs Low)")
[1] "Independent t-test: Marks by Marks_Group (High vs Low)"
> t_test_two <- t.test(Marks ~ Marks_Group, data = df)
> print(t_test_two)

        Welch Two Sample t-test

data:  Marks by Marks_Group
t = 13.53, df = 64.978, p-value < 2.2e-16
alternative hypothesis: true difference in means between group High and group Low is not equal to 0
95 percent confidence interval:
 19.6120 26.4042
sample estimates:
mean in group High  mean in group Low
          35.92174           12.91364

>
> # Create artificial Old_Marks for paired t-test
> set.seed(123)  # for reproducibility
> df$Old_Marks <- df$Marks + runif(nrow(df), 5, 15)
>
> print("Paired t-test: Marks vs Old_Marks")
```

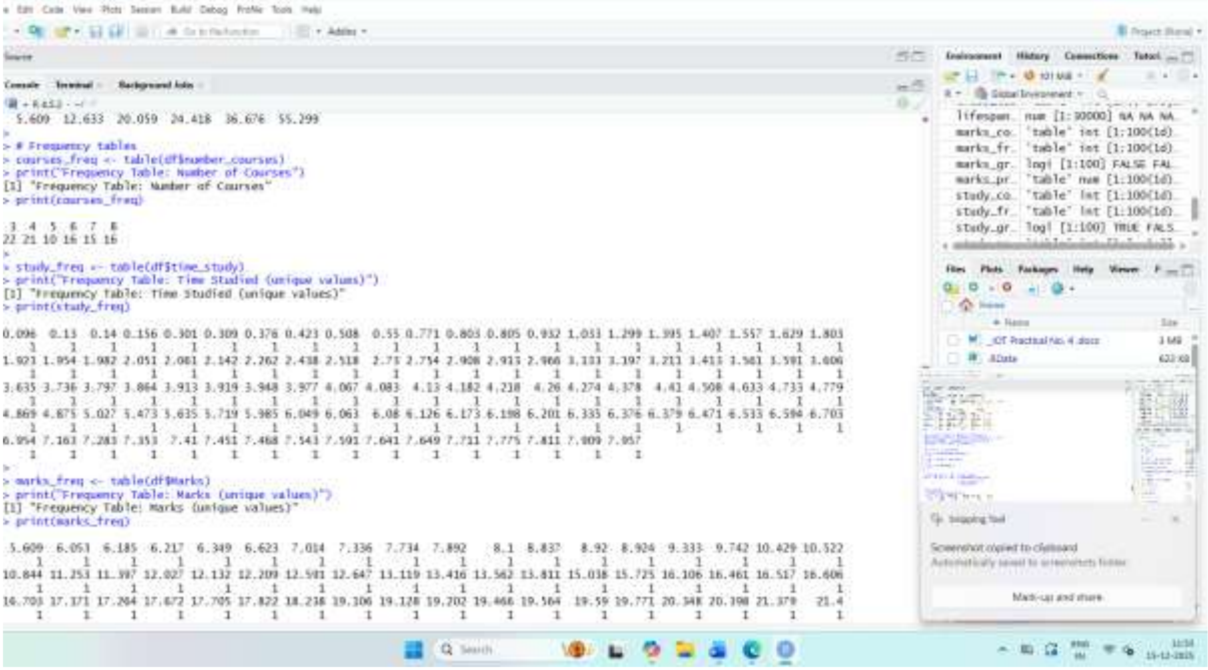
```
lifespan_  num [1:10000] NA NA NA_
marks_co_  'table' int [1:100(1d)_
marks_fr_  'table' int [1:100(1d)_
marks_gr_  logi [1:100] FALSE FAL_
marks_pr_  'table' num [1:100(1d)_
study_co_  'table' int [1:100(1d)_
study_fr_  'table' int [1:100(1d)_
study_gr_  logi [1:100] TRUE FALS_
```



```
> set.seed(123)  # For reproducibility
> df$Old_Marks <- df$Marks + runif(nrow(df), 5, 15)
>
> print("Paired t-test: Marks vs Old_Marks")
[1] "Paired t-test: Marks vs Old_Marks"
> t_test_paired <- t.test(df$Marks, df$Old_Marks, paired = TRUE)
> print(t_test_paired)

        Paired t-test

data:  df$Marks and df$Old_Marks
t = -35.038, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -10.551082  -9.420097
sample estimates:
mean difference
       -9.98559

> library(dplyr)
>
> # Load dataset
> df <- read.csv("C:/Users/mvlut/Downloads/Student_Marks.csv")
>
> # Basic exploration
> head(df)
  number_courses time_study  Marks
1              3      4.508 19.202
2              4      0.096  7.734
3              4      3.133 13.811
4              6      7.909 53.018
5              8      7.811 55.299
6              6      3.211 17.822
> str(df)
'data.frame':	100 obs. of  3 variables:
 $ number_courses: int  3 4 4 6 8 6 3 5 4 3 ...
 $ time_study    : num  4.508 0.096 3.133 7.909 7.811 ...
 $ Marks         : num  19.2 7.73 13.81 53.02 55.3 ...
```



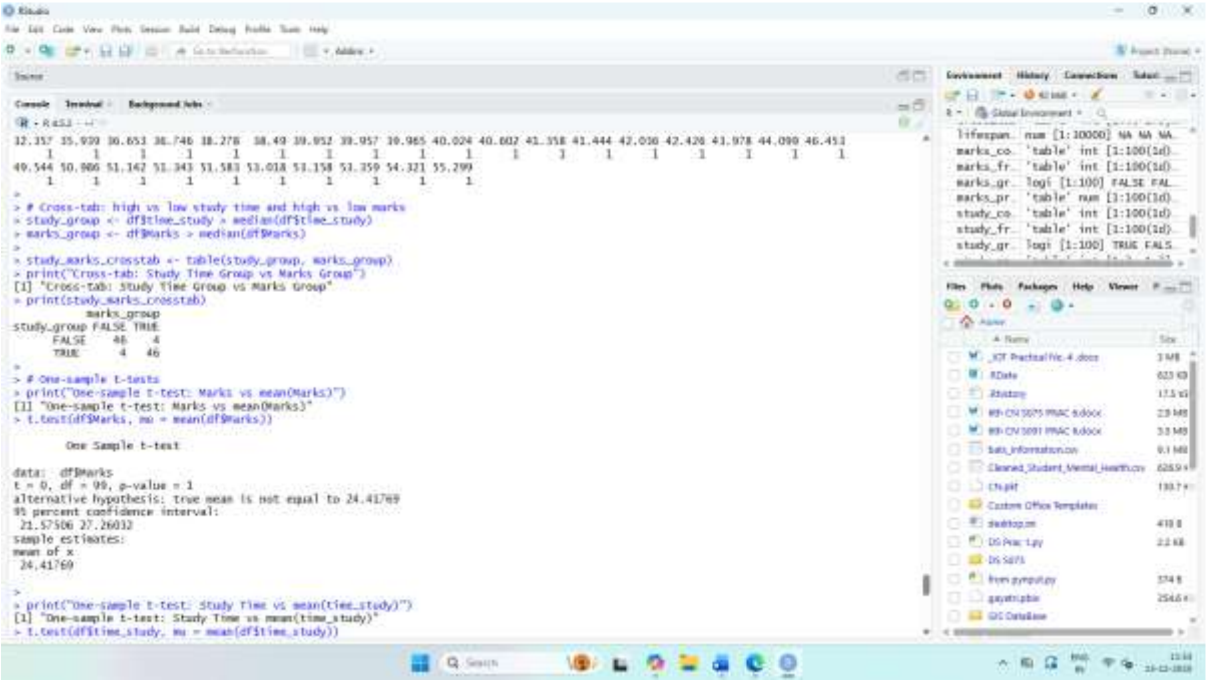
```
'data.frame':	100 obs. of  3 variables:
 $ number_courses: int  3 4 4 6 8 6 3 5 4 3 ...
 $ time_study    : num  4.508 0.096 3.133 7.909 7.811 ...
 $ Marks         : num  19.2 7.73 13.81 53.02 55.3 ...
> summary(df)
 number_courses   time_study        Marks
 Min.   :3.00   Min.   :0.096   Min.   : 5.609
 1st Qu.:4.00   1st Qu.:2.058   1st Qu.:12.633
 Median :5.00   Median :4.022   Median :20.059
 Mean   :5.29   Mean   :4.077   Mean   :24.418
 3rd Qu.:7.00   3rd Qu.:6.179   3rd Qu.:36.676
 Max.   :8.00   Max.   :7.957   Max.   :55.299
>
> # Convert columns to numeric (safeguard)
> df$number_courses <- as.numeric(df$number_courses)
> df$time_study <- as.numeric(df$time_study)
> df$Marks <- as.numeric(df$Marks)
>
> # Check missing values
> sum(is.na(df$number_courses))
[1] 0
> sum(is.na(df$time_study))
[1] 0
> sum(is.na(df$Marks))
[1] 0
>
> # Remove rows with missing values
> df <- df %>% filter(!is.na(number_courses),
+                     !is.na(time_study),
+                     !is.na(Marks))
>
> # Summaries
> summary(df$number_courses)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    4.00    5.00    5.29    7.00    8.00
> summary(df$time_study)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.096   2.058   4.022   4.077   6.179   7.957
```



```
 5.609  12.633  20.059  24.418  36.676  55.299 
> 
> # Frequency tables
> courses_freq <- table(df$number_courses)
> print("Frequency Table: Number of Courses")
[1] "Frequency Table: Number of Courses"
> print(courses_freq)

 3  4  5  6  7  8 
22 21 10 16 15 16 
> 
> study_freq <- table(df$time_study)
> print("Frequency Table: Time Studied (unique values)")
[1] "Frequency Table: Time Studied (unique values)"
> print(study_freq)

0.096  0.13  0.14 0.156 0.301 0.309 0.376 0.423 0.508  0.55 0.771 0.803 0.805 0.932 1.033 1.299 1.395 1.407 1.557 1.629 1.803 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
1.921 1.954 1.982 2.051 2.061 2.142 2.262 2.438 2.518  2.73 2.754 2.908 2.913 2.966 3.133 3.197 3.213 3.413 3.581 3.591 3.606 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
3.635 3.736 3.797 3.864 3.913 3.919 3.948 3.977 4.067 4.083  4.13 4.182 4.218  4.26 4.274 4.378  4.41 4.508 4.633 4.733 4.779 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
4.869 4.875 5.027 5.473 5.635 5.719 5.985 6.049 6.063  6.08 6.126 6.173 6.198 6.201 6.335 6.376 6.379 6.471 6.533 6.594 6.703 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
6.954 7.163 7.283 7.353  7.41 7.451 7.468 7.543 7.591 7.641 7.649 7.711 7.775 7.811 7.909 7.957 
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1 
> 
> marks_freq <- table(df$Marks)
> print("Frequency Table: Marks (unique values)")
[1] "Frequency Table: Marks (unique values)"
> print(marks_freq)

 5.609  6.053  6.185  6.217  6.349  6.623  7.014  7.336  7.734  7.892    8.1  8.837   8.92  8.924  9.333  9.742 10.429 10.522 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
10.844 11.253 11.397 12.027 12.132 12.209 12.591 12.647 13.119 13.416 13.562 13.811 15.038 15.725 16.106 16.461 16.517 16.606 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
16.703 17.171 17.264 17.672 17.705 17.822 18.238 19.106 19.128 19.202 19.466 19.564  19.59 19.771 20.348 20.398 21.379   21.4 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
```

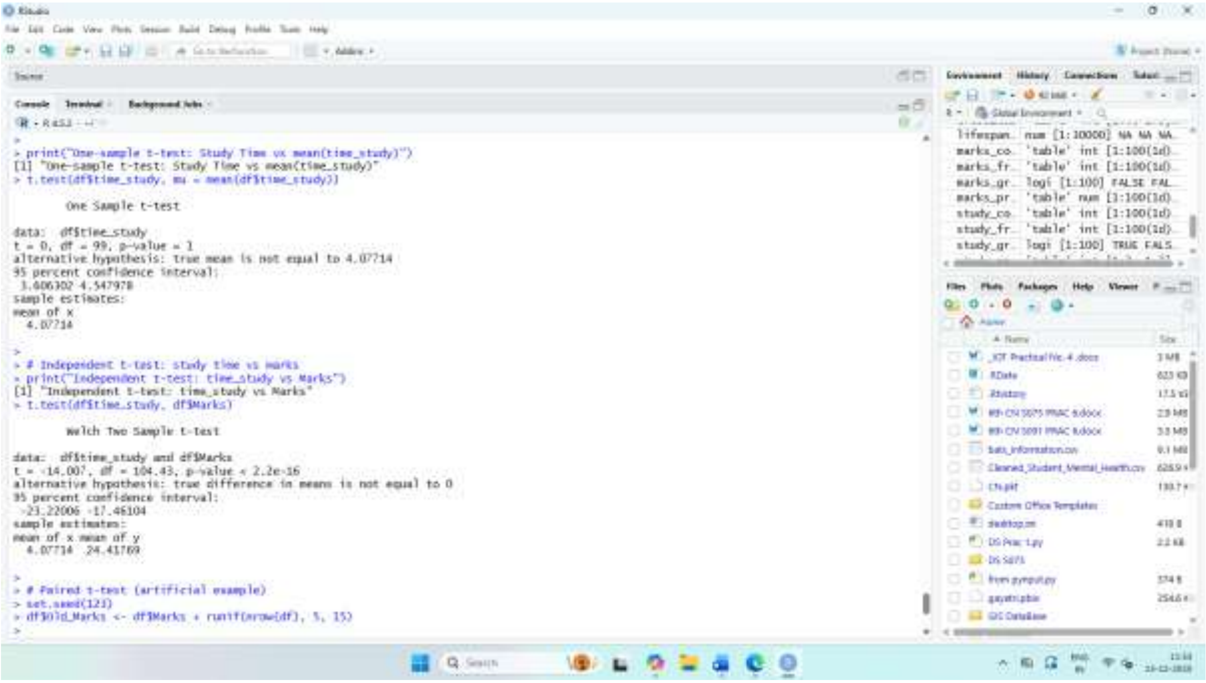
```
lifespan… num [1:10000] NA NA NA…
marks_co… 'table' int [1:100(1d)…
marks_fr… 'table' int [1:100(1d)…
marks_gr… logi [1:100] FALSE FAL…
marks_pr… 'table' num [1:100(1d)…
study_co… 'table' int [1:100(1d)…
study_fr… 'table' int [1:100(1d)…
study_gr… logi [1:100] TRUE FALS…
```

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder
Mark-up and share



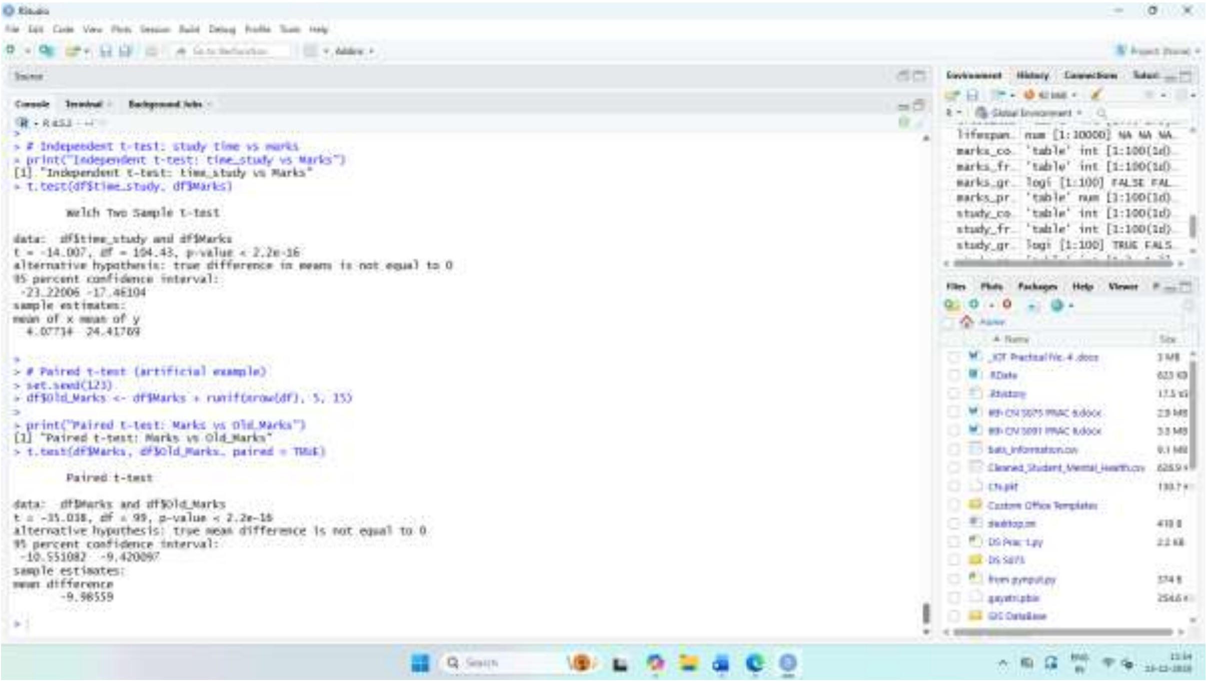
```
32.357 35.939 36.653 36.746 38.278  38.49 39.052 39.957 39.965 40.024 40.602 41.358 41.444 42.036 42.426 43.978 44.090 46.453 
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1 
49.544 50.986 51.142 51.343 51.583 53.018 53.158 53.359 54.321 55.299 
     1      1      1      1      1      1      1      1      1      1 
> 
> # Cross-tab: high vs low study time and high vs low marks
> study_group <- df$time_study > median(df$time_study)
> marks_group <- df$Marks > median(df$Marks)
> 
> study_marks_crosstab <- table(study_group, marks_group)
> print("Cross-tab: Study Time Group vs Marks Group")
[1] "Cross-tab: Study Time Group vs Marks Group"
> print(study_marks_crosstab)
           marks_group
study_group FALSE TRUE
      FALSE    46    4
      TRUE      4   46
> 
> # One-sample t-tests
> print("One-sample t-test: Marks vs mean(Marks)")
[1] "One-sample t-test: Marks vs mean(Marks)"
> t.test(df$Marks, mu = mean(df$Marks))

	One Sample t-test

data:  df$Marks
t = 0, df = 99, p-value = 1
alternative hypothesis: true mean is not equal to 24.41769
95 percent confidence interval:
 21.57506 27.26032
sample estimates:
mean of x 
 24.41769 

> 
> print("One-sample t-test: Study Time vs mean(time_study)")
[1] "One-sample t-test: Study Time vs mean(time_study)"
> t.test(df$time_study, mu = mean(df$time_study))
```

```
lifespan… num [1:10000] NA NA NA…
marks_co… 'table' int [1:100(1d)…
marks_fr… 'table' int [1:100(1d)…
marks_gr… logi [1:100] FALSE FAL…
marks_pr… 'table' num [1:100(1d)…
study_co… 'table' int [1:100(1d)…
study_fr… 'table' int [1:100(1d)…
study_gr… logi [1:100] TRUE FALS…
```



```
>
> print("One-sample t-test: Study Time vs mean(time_study)")
[1] "One-sample t-test: Study Time vs mean(time_study)"
> t.test(df$time_study, mu = mean(df$time_study))

        One Sample t-test

data:  df$time_study
t = 0, df = 99, p-value = 1
alternative hypothesis: true mean is not equal to 4.07714
95 percent confidence interval:
 3.606302 4.547978
sample estimates:
mean of x
  4.07714

>
> # Independent t-test: study time vs marks
> print("Independent t-test: time_study vs Marks")
[1] "Independent t-test: time_study vs Marks"
> t.test(df$time_study, df$Marks)

        Welch Two Sample t-test

data:  df$time_study and df$Marks
t = -14.007, df = 104.43, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -23.22006 -17.46104
sample estimates:
mean of x mean of y
  4.07714  24.41769

>
> # Paired t-test (artificial example)
> set.seed(123)
> df$Old_Marks <- df$Marks + runif(nrow(df), 5, 15)
>
```

```
lifespan.. num [1:10000] NA NA NA..
marks_co.. 'table' int [1:100(1d)..
marks_fr.. 'table' int [1:100(1d)..
marks_gr.. logi [1:100] FALSE FAL..
marks_pr.. 'table' num [1:100(1d)..
study_co.. 'table' int [1:100(1d)..
study_fr.. 'table' int [1:100(1d)..
study_gr.. logi [1:100] TRUE FALS..
```



```
>
> # Independent t-test: study time vs marks
> print("Independent t-test: time_study vs Marks")
[1] "Independent t-test: time_study vs Marks"
> t.test(df$time_study, df$Marks)

        Welch Two Sample t-test

data:  df$time_study and df$Marks
t = -14.007, df = 104.43, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -23.22006 -17.46104
sample estimates:
mean of x mean of y
  4.07714  24.41769

>
> # Paired t-test (artificial example)
> set.seed(123)
> df$Old_Marks <- df$Marks + runif(nrow(df), 5, 15)
>
> print("Paired t-test: Marks vs Old_Marks")
[1] "Paired t-test: Marks vs Old_Marks"
> t.test(df$Marks, df$Old_Marks, paired = TRUE)

        Paired t-test

data:  df$Marks and df$Old_Marks
t = -35.038, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -10.551082  -9.420097
sample estimates:
mean difference
       -9.98559

>
```

```
lifespan.. num [1:10000] NA NA NA..
marks_co.. 'table' int [1:100(1d)..
marks_fr.. 'table' int [1:100(1d)..
marks_gr.. logi [1:100] FALSE FAL..
marks_pr.. 'table' num [1:100(1d)..
study_co.. 'table' int [1:100(1d)..
study_fr.. 'table' int [1:100(1d)..
study_gr.. logi [1:100] TRUE FALS..
```