

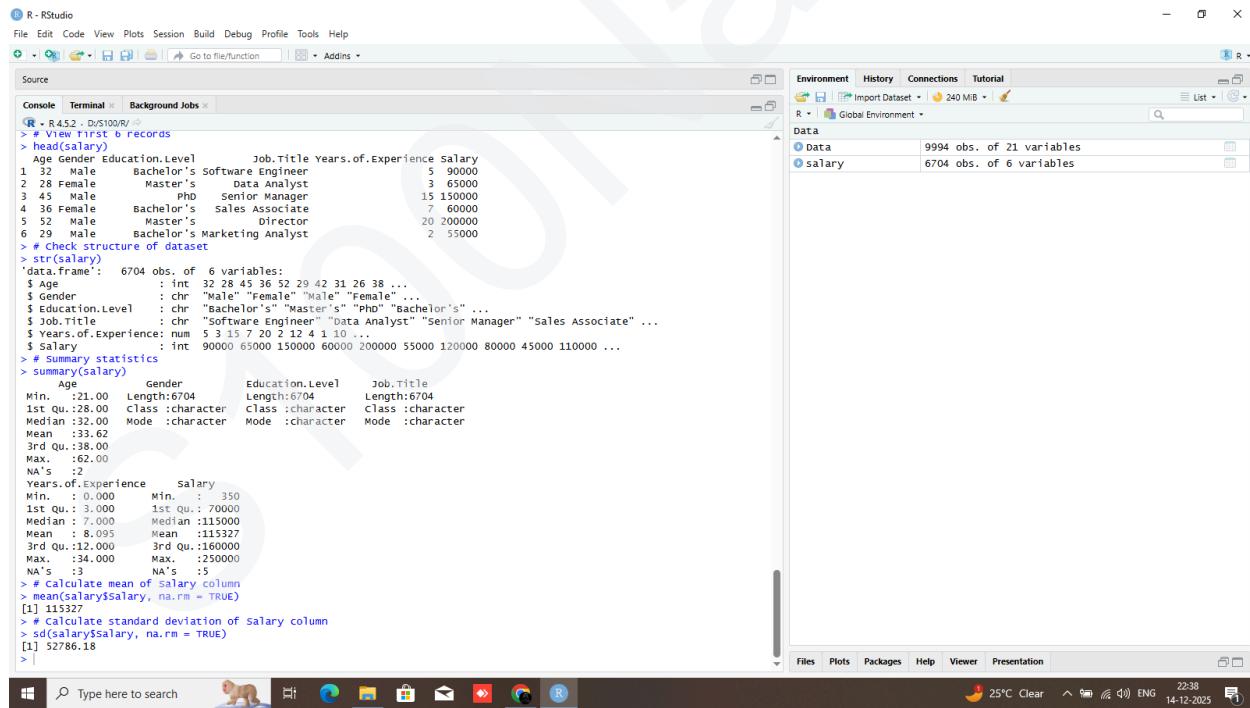
MVLU COLLEGE

Practical no :- 01 (Mod 2)

Aim :- Generating descriptive statistics using summary() or describe() ®

Code :-

```
install.packages(c("psych", "dplyr", "tidyverse"))
library(psych)
library(dplyr)
library(tidyverse)
# Load the dataset
salary <- read.csv("C:/Users/Arvind/Downloads/Salary_Data.csv")
# View first 6 records
head(salary)
# Check structure of dataset
str(salary)
# Summary statistics
summary(salary)
# Calculate mean of Salary column
mean(salary$Salary, na.rm = TRUE)
# Calculate standard deviation of Salary column
sd(salary$Salary, na.rm = TRUE)
```



MVLU COLLEGE

Practical N :- 02 (Mod 2)

Aim :- Generating frequency tables using table() or count() (R).

Code :-

```
library(psych)
library(dplyr)
library(tidyverse)
```

```
# Load dataset
```

```
jobs <- read.csv("C:/Users/Arvind/Downloads/Data_Science_Jobs_in_India.csv")
colnames(jobs)
```

```
# View structure
```

```
str(jobs)
```

```
job_table <- table(jobs$job_title)
```

```
job_table
```

```
# Proportion table
```

```
prop.table(job_table)
```

MVLU COLLEGE

The image displays two side-by-side screenshots of RStudio environments, showing the analysis of a dataset named 'Data_Science_Jobs_in_India.csv'. Both screenshots show the same code execution and output.

Code Execution:

```
R - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source Terminal Background Jobs
> library(tidyverse)
> # Load dataset
> jobs <- read.csv("C:/Users/Arvind/Downloads/Data_Science_Jobs_in_India.csv")
> View(jobs)
> # view structure
> str(jobs)
'data.frame': 1602 obs. of 8 variables:
 $ X : int 0 1 2 3 4 5 6 7 8 9 ...
 $ company_name : chr "TCS" "Accenture" "IBM" "Cognizant" ...
 $ job_title : chr "Data scientist" "Data scientist" "Data Scientist" "Data scientist" ...
 $ min_experience : int 2 2 2 2 2 2 2 2 2 ...
 $ avg_salary : chr "7.8L" "12.4L" "13.4L" "9.8L" ...
 $ min_salary : chr "4.5L" "5.8L" "5.3L" "5.0L" ...
 $ max_salary : chr "16.0L" "23.0L" "25.0L" "18.0L" ...
 $ num_of_salaries: int 841 501 394 318 300 228 218 166 163 ...
> # Frequency table for job locations
> table(jobs$location)
<table>
> # Proportions
> prop.table(table(jobs$location))
num[0]
> colnames(jobs)
[1] "X"           "company_name"   "job_title"      "min_experience"
[5] "avg_salary"  "min_salary"     "max_salary"    "num_of_salaries"
> job_table <- table(jobs$job_title)
> # Load dataset
> jobs <- read.csv("C:/Users/Arvind/Downloads/Data_Science_Jobs_in_India.csv")
> # display frequency table
> job_table
Business Analyst          Data Analyst          Data Architect
188                         187                      50
Data Engineer               Data Scientist        Machine Learning Engineer
188                         188                      59
Senior Business Analyst    Senior Data Analyst Senior Data Engineer
187                         187                     183
Senior Data Scientist
185
> # proportion table
> prop.table(job_table)
Business Analyst          Data Analyst          Data Architect
0.11735331                 0.11672909                 0.03121099
Data Engineer               Data Scientist        Machine Learning Engineer
0.11735331                 0.11735331                 0.03682896
Senior Business Analyst    Senior Data Analyst Senior Data Engineer
0.11672909                 0.11672909                 0.11423221
Senior Data Scientist
0.11548061
```

Environment View:

The Environment pane shows the following objects:

- Data: 9994 obs. of 21 variables
- jobs: 1602 obs. of 8 variables
- salary: 6704 obs. of 6 variables
- Values:
 - company_table: 'table' int [1:642(1d)] 2 1 1 5 1 3 1 10 2 2 ...
 - job_table: 'table' int [1:10(1d)] 188 187 50 188 188 59 187...

MVLU COLLEGE

Practical No :- 03 (Mod 2)

Aim :- Creating cross-tabulations and two-way tables using table() (R).

CODE :-

```
library(psych)
library(dplyr)
library(tidyverse)
```

```
phones <- read.csv("C:/Users/Arvind/Downloads/Smartphones_cleaned_dataset.csv")
names(phones)
```

```
# Convert has_5g to factor (recommended)
phones$has_5g <- as.factor(phones$has_5g)
```

Cross-tabulation

```
brand_5g_table <- table(phones$brand_name, phones$has_5g)
```

Display table

```
brand_5g_table
```

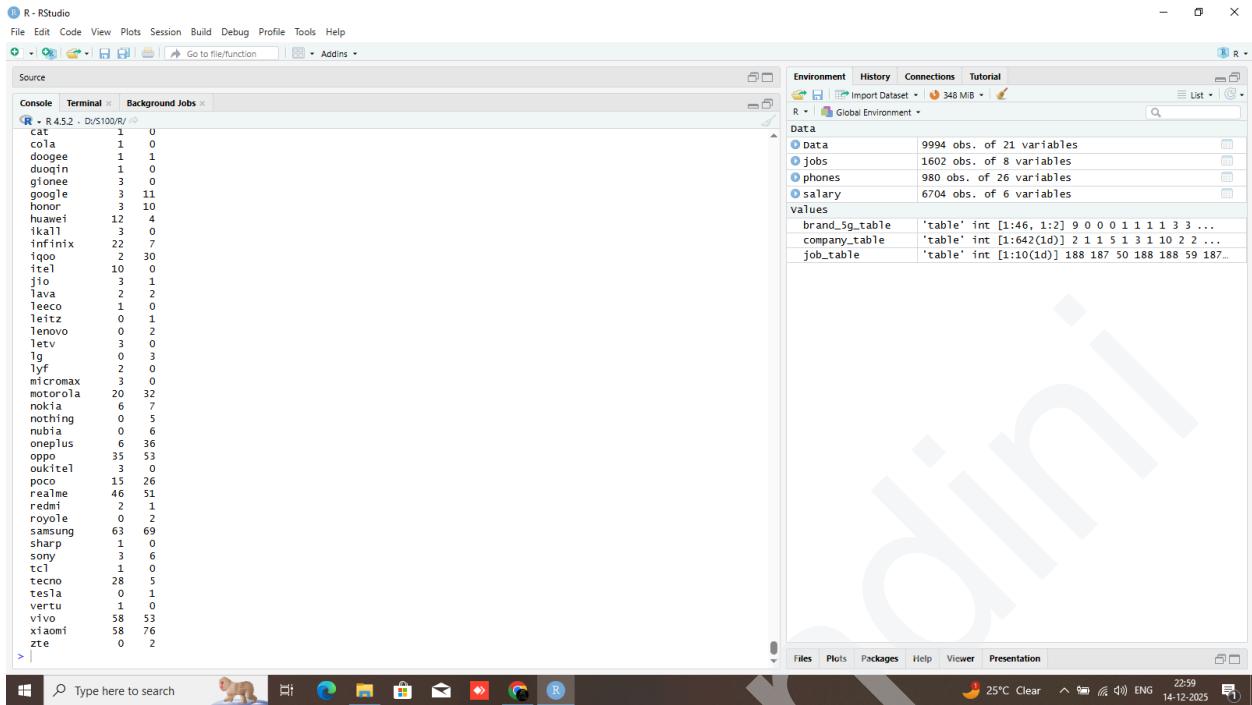
The screenshot shows the RStudio interface with the following details:

- Console:** Displays the R session history with the following code and output:

```
> library(psych)
> library(dplyr)
> library(tidyverse)
> phones <- read.csv("C:/Users/Arvind/Downloads/Smartphones_cleaned_dataset.csv")
> # view columns
> names(phones)
[1] "brand_name"      "model"
[3] "price"           "rating"
[5] "has_5g"          "has_nfc"
[7] "has_ir_beacon"   "processor_brand"
[9] "num_cores"        "processor_speed"
[11] "battery_capacity" "fast_charging_available"
[13] "fast_charging"    "ram_capacity"
[15] "internal_memory"  "screen_size"
[17] "refresh_rate"     "num_rear_cameras"
[19] "num_front_cameras" "os"
[21] "primary_camera_rear" "primary_camera_front"
[23] "extended_memory_available" "extended_upto"
[25] "resolution_width"    "resolution_height"
> # Cross-tabulation vs 5G support
> table(phones$brand_name, phones$has_5g)
<table>
<table> extent 0 x 0 >
> # Convert has_5g to factor (recommended)
> phones$has_5g <- as.factor(phones$has_5g)
> # cross-tabulation
> brand_5g_table <- table(phones$brand_name, phones$has_5g)
> # Display table
> brand_5g_table
```
- Data View:** Shows the dataset structure:

	brand_5g_table	company_table	job_table
brand_5g_table	'table' int [1:46, 1:2]	9 0 0 0 1 1 1 1 3 3 ...	
company_table	'table' int [1:642(1d)]	2 1 1 5 1 3 1 10 2 2 ...	
job_table	'table' int [1:10(1d)]	188 187 50 188 188 59 187...	
- Environment View:** Shows the global environment variables.
- Plots:** Shows a small icon of a monkey, likely a placeholder for a plot.
- System Tray:** Shows the Windows taskbar with icons for File Explorer, Mail, and other applications, along with system status like battery level, temperature (25°C), and date (14-12-2025).

MVLU COLLEGE



Practical No :- 04 (Mod 2)

Aim :- Performing one-sample t-tests using t.test() (R).

Code :-

```
library(psych)
library(dplyr)
library(tidyverse)
```

```
# Load dataset
```

```
lang <- read.csv("C:/Users/Arvind/Downloads/programming language trend over time.csv")
names(lang)
```

```
# View data
```

```
head(lang)
```

```
lang_clean <- na.omit(lang[, c("Python", "Java")])
```

```
# Compare popularity of Python vs Java using independent t-test
t_test_result <- t.test(lang_clean$Python, lang_clean$Java)
```

```
# Display result
```

```
t_test_result
```

MVLU COLLEGE

R - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Console Terminal Background Jobs

R - R 4.2.2 - D:\S100\Nandini\

```
> lang <- read.csv("C:/users/Arvind/Downloads/programming language trend over time.csv")
> names(lang)
[1] "week"   "Python"  "Java"    "C.."
> # view data
> head(lang)
#> # week Python Java C..
#> 1 4/21/2019 55 55 18
#> 2 4/28/2019 52 50 16
#> 3 5/5/2019 56 56 17
#> 4 5/12/2019 56 61 18
#> 5 5/19/2019 57 56 17
#> 6 5/26/2019 55 57 17
> # Compare popularity of Python vs Java
> t.test(lang$Python, lang$Java)
```

Welch Two Sample t-test

```
data: lang$Python and lang$Java
t = 20.757, df = 380.41, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
18.20915 22.0986
sample estimates:
mean of x mean of y
68.96565 48.85115
```

```
> lang_clean <- na.omit(lang[, c("Python", "Java")])
> # compare popularity of Python vs Java using independent t-test
> t_test_result <- t.test(lang_clean$Python, lang_clean$Java)
> # display result
> t_test_result
```

Welch Two Sample t-test

```
data: lang_clean$Python and lang_clean$Java
t = 20.757, df = 380.41, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
18.20915 22.0986
sample estimates:
mean of x mean of y
68.96565 48.85115
```

Environment History Connections Tutorial

Import Dataset 346 MB

Data

- Data 9994 obs. of 21 variables
- jobs 1602 obs. of 8 variables
- lang 262 obs. of 4 variables
- lang_clean 262 obs. of 2 variables
- phones 980 obs. of 26 variables
- salary 6704 obs. of 6 variables
- t_test_result List of 10

values

- brand_5g_table 'table' int [1:46, 1:2] 9 0 0 0 1 1 1 1 3 3 ...
- company_table 'table' int [1:642(1d)] 2 1 1 5 1 3 1 10 2 2 ...
- job_table 'table' int [1:10(1d)] 188 187 50 188 188 59 187 ...

Files Plots Packages Help Viewer Presentation

25°C Clear 23:05 14-12-2025

MVLU COLLEGE

Practical No :- 05 (Mod 2)

Aim : Performing independent two-sample t-tests using t.test() with grouping (R).

Code :-

```
library(psych)
library(dplyr)
library(tidyverse)
```

```
gender_marks <- data.frame(
  Gender = c("Male","Male","Male","Female","Female","Female"),
  Marks = c(65, 70, 68, 75, 78, 80)
)
gender_marks

# Independent t-test
t.test(Marks ~ Gender, data = gender_marks)
```

The screenshot shows the RStudio interface with the following details:

- Console:** Displays the R session history. It includes the creation of the `gender_marks` data frame, the execution of the `t.test` command, and the output of the Welch Two Sample t-test.
- Environment:** Shows the global environment with objects like `Data`, `gender_marks`, `jobs`, `lang`, `lang_clean`, `phones`, `salary`, and `t_test_result`.
- Plots:** No plots are visible in this screenshot.
- Packages:** No packages are visible in this screenshot.
- Help:** No help pages are visible in this screenshot.
- Viewer:** No viewer content is visible in this screenshot.
- Presentation:** No presentation content is visible in this screenshot.

MVLU COLLEGE

Practical No :- 06 (Mod 2)

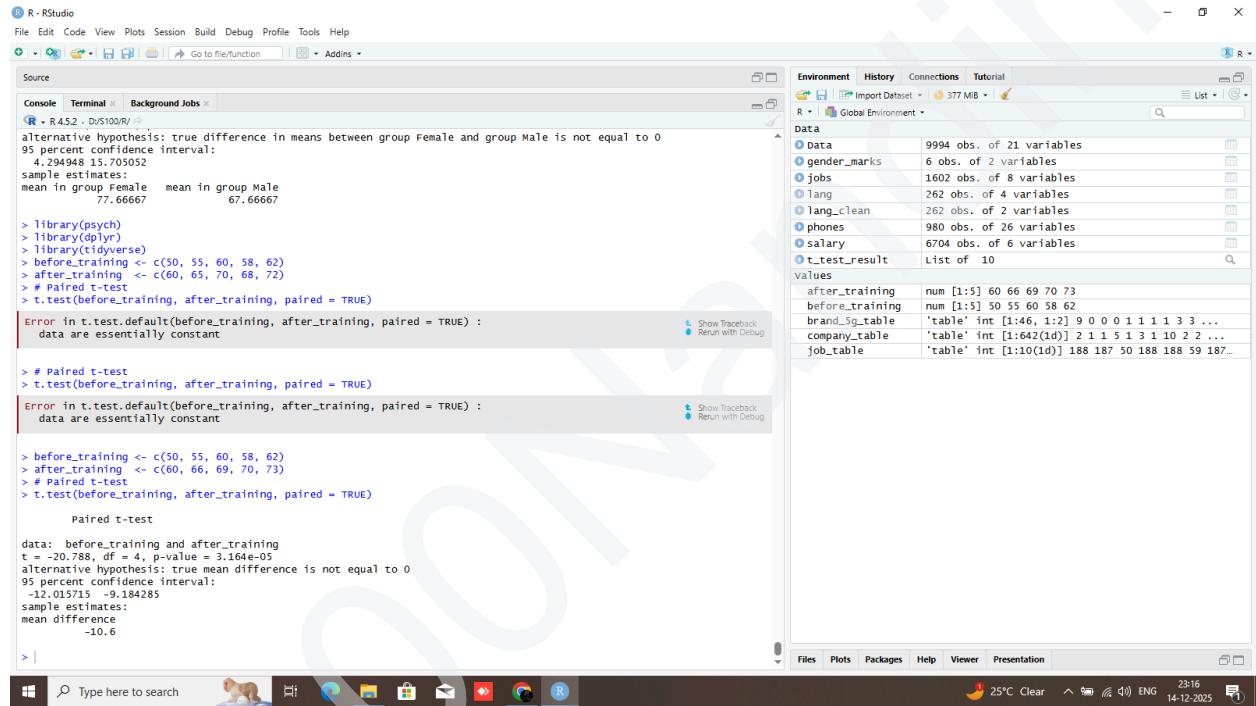
Aim : Performing paired t-tests using t.test(paired=TRUE) (R).

```
library(psych)
library(dplyr)
library(tidyverse)
```

```
before_training <- c(50, 55, 60, 58, 62)
after_training <- c(60, 66, 69, 70, 73)
```

Paired t-test

```
t.test(before_training, after_training, paired = TRUE)
```



The screenshot shows the RStudio interface with the following details:

- Console:** Displays the R session history. It shows the loading of psych, dplyr, and tidyverse packages, followed by defining vectors for before_training and after_training. It then attempts to run a paired t-test, which fails due to data being essentially constant.
- Data View:** Shows the environment with various objects:
 - Data: 9994 obs. of 21 variables
 - gender_marks: 6 obs. of 2 variables
 - jobs: 1602 obs. of 8 variables
 - lang: 262 obs. of 4 variables
 - lang_clean: 262 obs. of 2 variables
 - phones: 980 obs. of 26 variables
 - salary: 6704 obs. of 6 variables
 - t_test_result: List of 10
- Values View:** Shows the values for after_training and before_training.
- Environment View:** Shows the global environment with objects like brand_Sg_table, company_table, and job_table.
- Bottom Status Bar:** Shows system information including temperature (25°C), battery level (Clear), and date/time (14-12-2025).