Sookha Poori Assignment 1 (R & AA)

Section A

1. Vectors – Basics

(a) Create a numeric vector of 10 elements.
(b) Find its mean, median, and standard deviation.
(c) Replace all even numbers with NA.
(d) Remove NA values and display the final vector.

2. Vectors – Operations

(a) Create two vectors x and y of equal length.
(b) Perform element-wise addition, subtraction, and multiplication.
(c) Find positions where x > y.
(d) Display all common elements without using intersect().

3. Factors – Creation

(a) Create a factor for student grades: A, B, C, A, B, C, A.
(b) Display the levels.
(c) Change the order of levels from C < B < A.
(d) Add a new level "D" and assign it to one observation.

4. Factors – Summary

(a) Create a factor of 20 random colors from c("Red", "Blue", "Green").
(b) Count how many times each color appears using table().
(c) Convert the factor into a character vector.
(d) Sort it alphabetically.

5. Data Frames – Creation

(a) Create a dataframe with columns: Name, Age, Gender, Marks.
(b) Add a new column Result with “Pass” if Marks > 50, otherwise “Fail”.
(c) Display only students who passed.
(d) Find the mean Marks of male and female students separately.

6. Data Frames – Editing

(a) Create a dataframe of 5 rows and 3 columns.
(b) Add a column at the 2nd position.
(c) Reorder the columns alphabetically by name.
(d) Empty a column without deleting it.

7. Matrices – Creation

(a) Create a 3×3 matrix of numbers 1 to 9.
(b) Find the transpose, determinant, and inverse.
(c) Extract the diagonal elements.
(d) Multiply by another 3×3 matrix.

8. Matrices – Operations

(a) Create a 4×3 matrix filled by row.
(b) Find row sums and column means.
(c) Replace all elements > 8 with 0.
(d) Convert the matrix into a dataframe.

9. Control Statements – if else

(a) Write an R program that takes a number and checks if it is positive, negative, or zero.
(b) Modify it to also check if the number is even or odd.
(c) Display all results in one message using cat().

10. **Control Statements – Nested if**

(a) Write a nested if-statement to grade marks:
>=90: A, >=75: B, >=60: C, otherwise Fail.
(b) Take marks as user input using readline().
(c) Display output using formatted message.

11. Loops – for

(a) Print numbers from 1 to 10 using a for loop.
(b) Print only even numbers on the same line.
(c) Print their squares using cat() in the same line.

12. Loops – while

(a) Use a while loop to find the sum of numbers from 1 to 50.
(b) Use a while loop to count how many numbers between 1 and 100 are divisible by both 3 and 5.
(c) Print the final count.

13. Loops – repeat

(a) Create a repeat loop that keeps asking for a number until the user enters a negative value.
(b) Display how many positive numbers were entered.
(c) Break the loop when negative number appears.

14. Functions – Basics

(a) Write a user-defined function to calculate the area of a circle.
(b) Modify it to return both area and circumference.
(c) Call the function for radius = 5, 7, and 10.

15. Functions – with Loops

(a) Write a function that takes a vector as input and returns a vector of squares.
(b) Use a for loop inside.
(c) Test it on a vector of numbers 1:10.

16. Functions – Conditional Logic

(a) Write a function that takes marks as input and returns grade using ifelse().
(b) Apply it on a vector of marks.
(c) Display all grades in one line.

17. Plots – Scatter Plot

(a) Create two vectors x and y of equal length.
(b) Plot them using plot().
(c) Add title, axis labels, color, and change point type.
(d) Add a regression line using abline(lm(y ~ x)).

18. Plots – Bar Plot

(a) Create a vector of monthly sales.
(b) Plot it using barplot().
(c) Add color, title, and custom x-axis labels.
(d) Make it horizontal.

19. Plots – Histogram

(a) Generate 100 random numbers using rnorm().
(b) Plot a histogram with color blue.
(c) Overlay the density line in red.
(d) Add a title and label the x-axis.

20. Combined Concepts

(a) Create a dataframe with 2 columns Age and Height (20 rows).
(b) Plot a scatter plot between them.
(c) Use ifelse() to label observations as “Tall” or “Short” based on height > 160.
(d) Create a bar plot showing the count of “Tall” vs “Short”.

Section B

1. Vectors – Logical & Indexing

(a) Create a numeric vector of 20 random integers between 1 and 50.
(b) Replace all numbers divisible by both 3 and 5 with their square.
(c) Extract all prime numbers from the vector.
(d) Find sum of all elements greater than the mean.

2. Vectors – Custom Filtering

(a) Create a character vector of fruit names.
(b) Find all fruits whose names start with vowels.
(c) Replace any fruit name containing more than 5 letters with "LongFruit".
(d) Sort the final vector alphabetically.

3. Factors – Manipulation

(a) Create a factor of departments: "HR", "IT", "Finance", "Sales", "IT", "Finance", "HR".
(b) Convert factor to numeric codes.
(c) Merge "HR" and "IT" into a single level "Support".
(d) Drop unused levels after merging.

4. Factors – Analysis

(a) Generate a factor of 30 random grades (A, B, C, D).
(b) Find the most frequently occurring grade.
(c) Convert factor into an ordered factor from D < C < B < A.
(d) Plot a bar chart of grade distribution.

5. Data Frames – Conditional Filtering

(a) Create a dataframe with columns Name, Gender, Age, Marks.
(b) Select all female students above age 20 with marks > 70.
(c) Sort dataframe by marks (descending).
(d) Display only top 3 rows.

6. Data Frames – Advanced Operations

(a) Create a dataframe of 10 employees with Name, Department, Salary.
(b) Add a new column Tax = 10% of salary if salary > 50000, else 5%.
(c) Add a column NetSalary = Salary - Tax.
(d) Find average net salary by department using aggregate().

7. Matrices – Row/Column Operations

(a) Create a 4×4 matrix of numbers 1–16.
(b) Replace the diagonal with zeros.
(c) Swap first and last rows.
(d) Find row having the maximum sum.

8. Matrices – Logical Computations

(a) Create a 5×5 matrix with random integers (10–99).
(b) Replace all even numbers with their half.
(c) Find all elements greater than 40 and less than 60.
(d) Calculate sum of each column using apply().

9. Control Statements – Nested if

(a) Accept a numeric input temp (temperature).
(b) Display messages:
temp > 40 → "Too Hot",
temp < 10 → "Too Cold",
10–40 → "Pleasant Weather".
(c) Add condition to display "Rainy Day" if temperature between 20–25 and divisible by 5.

10. **Loops – For + Condition**

(a) Write a for loop to print all numbers from 1–50 that are perfect squares.
(b) Find their sum.
(c) Count how many are even.

11. Loops – Nested

(a) Use nested loops to print the following pattern:

```
*

* *

* * *

* * * *
```

(b) Modify the same loop to print numbers instead of stars.

12. While Loop – Data Validation

(a) Keep asking user to input marks until a valid value (0–100) is entered.
(b) Display "Valid marks entered!" when condition is met.
(c) Use break to exit loop.

13. Repeat Loop – Counting

(a) Generate random numbers (1–100) using repeat until you get a multiple of 13.
(b) Count how many iterations it took.
(c) Print the final number.

14. Functions – Multi-Return

(a) Write a function analyze_vector(x) that returns:
- sum, mean, median, sd, and max value.

(b) Test it on a random numeric vector.
(c) Display output as a named list.

15. Functions – String Handling

(a) Write a function that takes a character vector of names.
(b) Returns how many names start with each letter (A–Z).
(c) Hint: Use substr() and table().

16. Functions – Using Loops

(a) Create a function that takes a number n and prints multiplication table up to 10.
(b) Modify it to store results in a vector and return that vector.
(c) Call function for n = 7.

17. Plots – Histogram + Density

(a) Generate 200 random numbers from a normal distribution.
(b) Plot histogram with freq = FALSE.
(c) Add red density curve on same plot.
(d) Add title, axis labels, and legend.

18. Plots – Multiple Lines

(a) Create two vectors time (1–10) and speed1 = time*3, speed2 = time*2+5.
(b) Plot both lines on the same graph with different colors and line types.
(c) Add legend and grid lines.
(d) Save the plot as "speed_plot.png" using png() and dev.off().

19. Combined Concepts – Data Frame + Plot

(a) Create a dataframe of 15 students with Name, Gender, Marks1, Marks2, Marks3.
(b) Calculate total and average marks.
(c) Create a bar plot of average marks by gender.
(d) Highlight top 3 students in a different color.

20. Full Challenge – Integration

(a) Write a function analyze_data(df) that takes a dataframe with numeric columns only.
(b) Returns a dataframe with column names, means, medians, and standard deviations.
(c) Use loops (not apply functions).
(d) Test on a dataframe with 3 numeric columns.

Section C

1. mtcars Dataset

(a) Display the first and last 5 rows.
(b) Find how many cars have mpg > 25.
(c) Calculate mean horsepower (hp) by number of cylinders (cyl).
(d) Create a scatter plot of mpg vs hp with red points.

2. mtcars – Filtering & Sorting

(a) Extract cars with 6 cylinders and automatic transmission (am = 0).
(b) Find the car with maximum mileage (mpg).
(c) Sort the dataset by descending horsepower.
(d) Plot a bar chart of mpg for the top 5 most fuel-efficient cars.

3. iris Dataset

(a) Display the structure and summary of the dataset.
(b) Find the mean Sepal.Length for each Species.
(c) Plot a boxplot comparing Petal.Length across species.
(d) Create a scatter plot of Sepal.Length vs Petal.Length with color-coded species.

4. iris – Advanced Manipulation

(a) Add a new column Sepal.Ratio = Sepal.Length / Sepal.Width.
(b) Find species with the highest average Sepal.Ratio.
(c) Create a histogram of Sepal.Ratio for setosa.
(d) Overlay a density curve on the same histogram.

5. airquality Dataset

(a) Display all rows where Ozone value is missing.
(b) Replace missing Ozone values with the mean of the column.
(c) Calculate average temperature (Temp) for each month.
(d) Plot a line chart of Ozone levels across Day for month 7 (July).

6. airquality – Correlation Analysis

(a) Create a correlation matrix of numeric columns.
(b) Plot a scatter plot of Wind vs Ozone.
(c) Add regression line using abline(lm(Ozone ~ Wind)).
(d) Comment on the relationship.

7. ToothGrowth Dataset

(a) Display the structure and summary.
(b) Find mean tooth length (len) for each supplement type (supp).
(c) Create a boxplot comparing len by dose.
(d) Add a title and color customization to the plot.

8. ToothGrowth – Group Analysis

(a) Use aggregate() to find mean len by dose and supp.
(b) Plot the result using a grouped bar plot.
(c) Add legend and custom colors.
(d) Comment on which supplement seems more effective.

9. PlantGrowth Dataset

(a) Display mean and median weight by treatment group.
(b) Perform a t-test between ctrl and trt1.
(c) Create a bar plot of mean weight for each group.
(d) Add error bars using arrows().

10. CO2 Dataset

(a) View the structure and number of unique plants.
(b) Calculate mean uptake by Type and Treatment.
(c) Plot uptake vs conc using different colors for Type.
(d) Add a legend and title "CO2 Uptake by Plant Type".

11. ChickWeight Dataset

(a) Display first 10 rows.
(b) Find the average weight for each Time point.
(c) Plot weight progression of each chick using line plot.
(d) Highlight the chick with maximum final weight.

12. trees Dataset

(a) Calculate the correlation between Girth, Height, and Volume.
(b) Fit a linear model predicting Volume using Girth.
(c) Plot Volume vs Girth with regression line.
(d) Display equation on plot using text().

13. USArrests Dataset

(a) Find states with highest and lowest Murder rate.
(b) Calculate mean of each column.
(c) Plot a bar chart of UrbanPop for all states.
(d) Highlight top 5 states in a different color.

14. women Dataset

(a) Plot height vs weight using points and line (type = "b").
(b) Add main title, axis labels, and grid lines.
(c) Fit a linear regression line.
(d) Display correlation coefficient on the plot.

15. pressure Dataset

(a) Plot pressure vs temperature using a line plot.
(b) Add points using different color and shape.
(c) Fit a smooth curve using lines(lowess()).
(d) Save the plot as "pressure_plot.png".

16. faithful Dataset

(a) Display summary of eruptions and waiting time.
(b) Plot histogram of waiting time.
(c) Overlay density curve.
(d) Plot eruptions vs waiting time with regression line.

17. cars Dataset

(a) Display head of dataset.
(b) Plot speed vs distance using red points.
(c) Add regression line and equation on plot.
(d) Predict stopping distance for speed = 25 using lm().

18. mtcars – Using apply()

(a) Use apply() to find mean of all numeric columns.
(b) Use tapply() to find mean mpg by cyl.
(c) Use sapply() to check class of each column.
(d) Plot mean mpg by cyl using barplot.

19. iris – ggplot2 (optional if tidyverse used)

(a) Create scatter plot of Sepal.Length vs Petal.Length using ggplot().
(b) Color points by species.
(c) Add regression line for each species using geom_smooth().
(d) Add custom title and theme.

20. airquality – Complete Visualization

(a) Plot Ozone over time (by Day) for each Month.
(b) Use different colors for each month.
(c) Add legend, grid, and main title.
(d) Export the plot to a file "airquality_trend.jpg".

Section D

1. Central Tendency – Manual & Built-in

(a) Create a numeric vector of 20 values representing monthly expenses of families.
(b) Calculate mean, median, and mode (write your own function for mode).
(c) Compare all three measures and comment on the data's skewness.
(d) Plot a histogram and mark mean and median using vertical lines.

2. Dispersion – Range, Variance, and SD

(a) Create a vector of students' marks (20 observations).
(b) Calculate the range, variance, standard deviation, and interquartile range (IQR).
(c) Compute the **coefficient of variation (CV)** = (SD / Mean) × 100.
(d) Interpret the CV — is data more or less consistent?

3. Comparison Between Two Groups

(a) Create two vectors representing heights of male and female students.
(b) Compute mean, median, variance, and standard deviation for both.
(c) Compare which group shows more variation using CV.
(d) Plot both distributions on the same histogram (use different colors).

4. Using Built-in Dataset – mtcars

(a) Calculate the mean, median, and standard deviation of mpg, hp, and wt.
(b) Find the range and interquartile range of mpg.
(c) Compute coefficient of variation for all three variables.
(d) Create a boxplot comparing mpg and hp side by side and comment on spread.

5. Correlation & Covariance – iris Dataset

(a) Compute the correlation between all numeric columns using cor().
(b) Display covariance matrix using cov().
(c) Identify which pair of variables are most strongly correlated.
(d) Plot a scatterplot matrix using pairs() or GGally::ggpairs() (if ggplot is allowed).

6. Karl Pearson & Spearman Correlation

(a) Create two numeric vectors x and y (e.g., student marks and hours studied).
(b) Compute **Pearson correlation** and interpret strength & direction.
(c) Compute **Spearman rank correlation** and explain difference.
(d) Plot both variables and add regression line using abline(lm(y ~ x)).

7. Realistic Data Analysis

(a) Create a dataframe with 3 numeric columns: Income, Expenditure, Savings.
(b) Calculate mean, SD, and CV for all three.
(c) Check correlation among the variables.
(d) Create a scatter plot of Income vs Expenditure with regression line and add correlation coefficient on plot.

8. Group-wise Summary – ToothGrowth Dataset

(a) Find mean, median, and SD of len for each supplement type (supp).
(b) Compare variation across doses using aggregate().
(c) Plot boxplots of tooth length by supplement type.
(d) Add interpretation: which supplement shows more consistency?

9. Measures of Central Tendency on Grouped Data

(a) Create a frequency table for class intervals of marks:
0–10, 10–20, 20–30, 30–40, 40–50 with random frequencies.
(b) Calculate **approximate mean** using midpoints.
(c) Find **mode** using the grouped formula.
(d) Compare mean, median, and mode — what can you infer about skewness?

10. **Comprehensive Analysis – airquality Dataset**

(a) Handle missing values in Ozone and Solar.R (replace with mean).
(b) Compute descriptive statistics (mean, median, variance, sd, range, IQR) for all numeric variables.
(c) Find correlation between Ozone, Wind, and Temp.
(d) Plot a correlation heatmap using corrplot or scatterplot matrix.
(e) Write interpretation: which variable pair has strongest linear relationship?