R Fundamentals

Instructions

This project is designed to build a foundational understanding of R programming through practical, hands-on exercises. The objective is to help learners become familiar with how data is created, stored, and manipulated in R; the essential first step in any data analysis workflow. Through a sequence of structured problems, you will practice creating vectors, performing basic mathematical and statistical operations, using indexing and subsetting to access data, and working with built-in R functions. The project gradually introduces file handling and descriptive summaries, enabling you to connect coding concepts with real-world data exploration.

Section 1 – Vector Creation and Basic Operations (Problems 1–8)

1. Write lines of code to compute all of the following. Include the answers in your written report.

```
123 * 453
5^2 * 40
TRUE & FALSE
TRUE | FALSE
75 %% 10
75 / 10
```

2. Create a vector using the **c** function with the values 17, 12, -33, 5 and assign it to a variable called **first_vector**.

```
[1]  17  12 -33   5
```

3. Create a vector using the **c** function with the values 5, 10, 15, 20, 25, 30, 35 and assign it to a variable called **counting_by_fives.**

4. Create a vector using the range operator (the colon), that contains the numbers from 20 down to 1 . Store the result in a variable called **second_vector**.

```
 [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
```

5. Create a vector using the range operator that contains the number from 5 to 15. Store the result in a variable called **counting_vector**

```
 [1]  5  6  7  8  9 10 11 12 13 14 15
```

6. Create a vector with the values (96, 100, 85, 92, 81, 72). Store the result in a variable called **grades**

```
[1]  96 100  85  92  81  72
```

7. Add the number 3 to the vector **grades**. Store the result in a variable called **bonus_points_added**.

```
[1]  99 103  88  95  84  75
```

8. Create a vector with the values 1 – 100 and store it in a variable called one_to_one_hundred. Do not type out all 100 numbers.

Section 2 – Descriptive Statistics and Basic Functions (Problems 9–14)

9. Write each of the following lines of code. Add a one-sentence comment above each line explaining what is computed. Include your comments in the written report.

```
second_vector + 20
second_vector * 20
second_vector >= 20
second_vector != 20  # != means "not equal"
```

10. Using the built in **sum** function, compute the sum of **one_to_one_hundred**. Store the result in a variable called **total**.

```
[1] 5050
```

11. Using the built in **mean** function, compute the average of **one_to_one_hundred**. Store the result in a variable called **average_value**

```
[1] 50.5
```

12. Using the built in **median** function, compute the average of **one_to_one_hundred**. Store the result in a variable called **median_value**

```
[1] 50.5
```

13. Using the built in **max** function, compute the max of **one_to_one_hundred**. Store the result in a variable called **max_value**

```
[1] 100
```

14. Using the built in **min** function, compute the min of **one_to_one_hundred**. Store the result in a variable called **min_value**

```
[1] 1
```

Section 3 – Indexing and Subsetting (Problems 15–18)

15. Using brackets, extract the first value from **second_vector** and store it in a variable called **first_value**

```
[1] 20
```

16. Using brackets, extract the first, second and third values from **second_vector**. Store the result in a variable called **first_three_values.**

```
[1] 20 19 18
```

17. Using brackets, extract the 1st, 5th, 10th, and 11th elements of **second_vector**. Store the resulting vector in a variable called **vector_from_brackets**.

```
[1] 20 16 11 10
```

18. Use the brackets to extract elements from **first_vector** using the following vector **c(FALSE, TRUE, FALSE, TRUE)**. Store the result in a variable called **vector_from_boolean_brackets**. Explain in a comment what happens. Include the answer in your written report.

```
[1] 12  5
```

Section 4 – Code Interpretation and Conditional Filtering (Problems 19–23)

19. Examine the following piece of code and write a one sentence comment explaining what is happening. Include the answer in your written report.

```
second_vector >= 10
```

20. Examine the following piece of code and write a one sentence comment explaining what is happening and assuming **one_to_one_hundred**was computed in the previous problem. Include the answers in your written report.

```
one_to_one_hundred[one_to_one_hundred >= 20]
```

21. Using the same approach as in the previous question, create a new vector from the **grades** vector with only values larger than 85. Store the result in a variable called **lowest_grades_removed**.

```
[1]  96 100  92
```

22. Use the **grades** vector to create a new vector with the 3rd and 4th elements of **grades** removed. Store the result in a variable called **middle_grades_removed**. Try utilizing a vector of negative indexes to complete this task.

```
[1]  96 100  81  72
```

23. Use bracket notation to remove the 5th and 10th elements of **second_vector**. Store the result in a variable called **fifth_vector**.

```
 [1] 20 19 18 17 15 14 13 12 10  9  8  7  6  5  4  3  2  1
```

Section 5 – Working with Random Vectors and Mathematical Functions (24–30)

24. Write the following code. This creates a variable called random_vector that will be utilized in problems 25 - 30.

```
set.seed(5)
random_vector <- runif(n=10, min = 0, max = 1000)
```

25. Use the **sum** function to compute the total of **random_vector**. Store the result in a variable called **sum_vector**

```
[1] 5295.264
```

26. Use the **cumsum** function to compute the cumulative sum of **random_vector**. Store the result in a variable called called **cumsum_vector**

```
 [1]  200.2145  885.4330 1802.3088 2086.7083 2191.3584 2892.4159
3420.3759
 [8] 4228.3111 5184.8112 5295.2642
```

27. Use the **mean** function to compute the mean of **random_vector**. Store the result in a variable called **mean_vector**

```
[1] 529.5264
```

28. Use the **sd** function to compute the standard deviation of **random_vector**. Store the result in a variable called **sd_vector**

```
[1] 331.3606
```

29. Use the **round** function to round the values of **random_vector** Store the result in a variable called **round_vector**

```
 [1] 200 685 917 284 105 701 528 808 957 110
```

30. Use the **sort** function to sort the values of **random_vector.** Store the result in a variable called **sort_vector**

Section 6 – Data Import and Exploration (Problems 31–33)

31. Download the datafile **ds_salaries.csv** from Canvas. Save it on your computer in the same folder (directory) where your .R file for this project is located.

32. Use the function **read.csv** to read the **ds_salaries.csv** file. Store the result of the read into a variable called **first_dataframe.**

33. Use the **summary** function with **first_dataframe** to produce summary statistics based on each column of the data frame.