Module 1 Report

R Practice

ALY 6000
Professor: Maria Ayala
Analyst: Syed Faizan
Date: 1/12/2024

CONTENTS

Module 1 Report

Introduction

Module 1 of ALY 6000 introduces the RStudio environment in which we solve a list of prescribed problems in R. The problems are designed to enable us to develop competency with statements, variable assignments, different expressions, vectors and their manipulation, matrices, functionality of R, and the installation and loading of packages in R.

Historical Overview

R is a programming language and environment for statistical computing and graphics developed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, in the early 1990s. It is a successor of the S language and has gained tremendous popularity within the data science community as it is open-source and is equipped with extensive libraries for statistical modeling and visualization.

Problems and their Solutions

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

123 * 453
Answer 55719
5^2 * 40
Answer 1000
TRUE & FALSE
Answer FALSE
TRUE | FALSE
Answer TRUE
75 %% 10
Answer 5
75 / 10
Answer 7.5

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

(17 12 -33 5)

3. Create a vector using the **c** function with the values 5, 10, 15, 20, 2 5, 30, 35 and assign it to a variable called **counting_by_fives.**

(5, 10, 15, 20, 25, 30, 35)

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**.

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**

```
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**

```
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**.

```
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.

```
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 5
1 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 [73] 7
3 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96
97 98 99 100
```

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
```

Here 20 is added to each number of the numeric class vector named second_vector

```
second_vector * 20
```

Here 20 is multiplied to each number of the numeric class vector named second_vector

second_vector >= 20

Here we are checking if the value of a variable in the vector second_vector is greater than or equal to 20, and a logic value which is either TRUE or FALSE is returned based on the comparison.

second_vector != 20

Here we are checking if the value of a variable in the vector second_vector is equal to 20 or not, and a logic value which is TRUE is returned if the value is not equal to 20 and FALSE is returned if the value is equal to 20.

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

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**

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**

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**

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

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

20

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

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**.

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.

12 5

In the above problem the new vector vector_from_boolean_brackets contains only the elements of first_vector where the corresponding element is TRUE. That is, it selects the elements in first_vector at positions where the logic value is TRUE i.e. positions 2 and 4) giving the values 12 and 5. This is a way to selectively extract elements from a vector based on a logical condition.

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
```

This code checks which elements in second_vector are greater than or equal to 10.

```
second_vector >= 10
```

```
  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE
```

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]
```

This code extracts elements from one_to_one_hundred that are greater than or equal to 20.

```
20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  3
9  40  41 42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58
59  60  61  62  63 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78
79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  9
8  99 100
```

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**.

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.

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**.

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

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**

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**

200.2145 885.4330 1802.3088 2086.7083 2191.3584 2892.4159 3420.37 59 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**

529.5264

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

331.3606

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

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**

105 110 200 284 528

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.

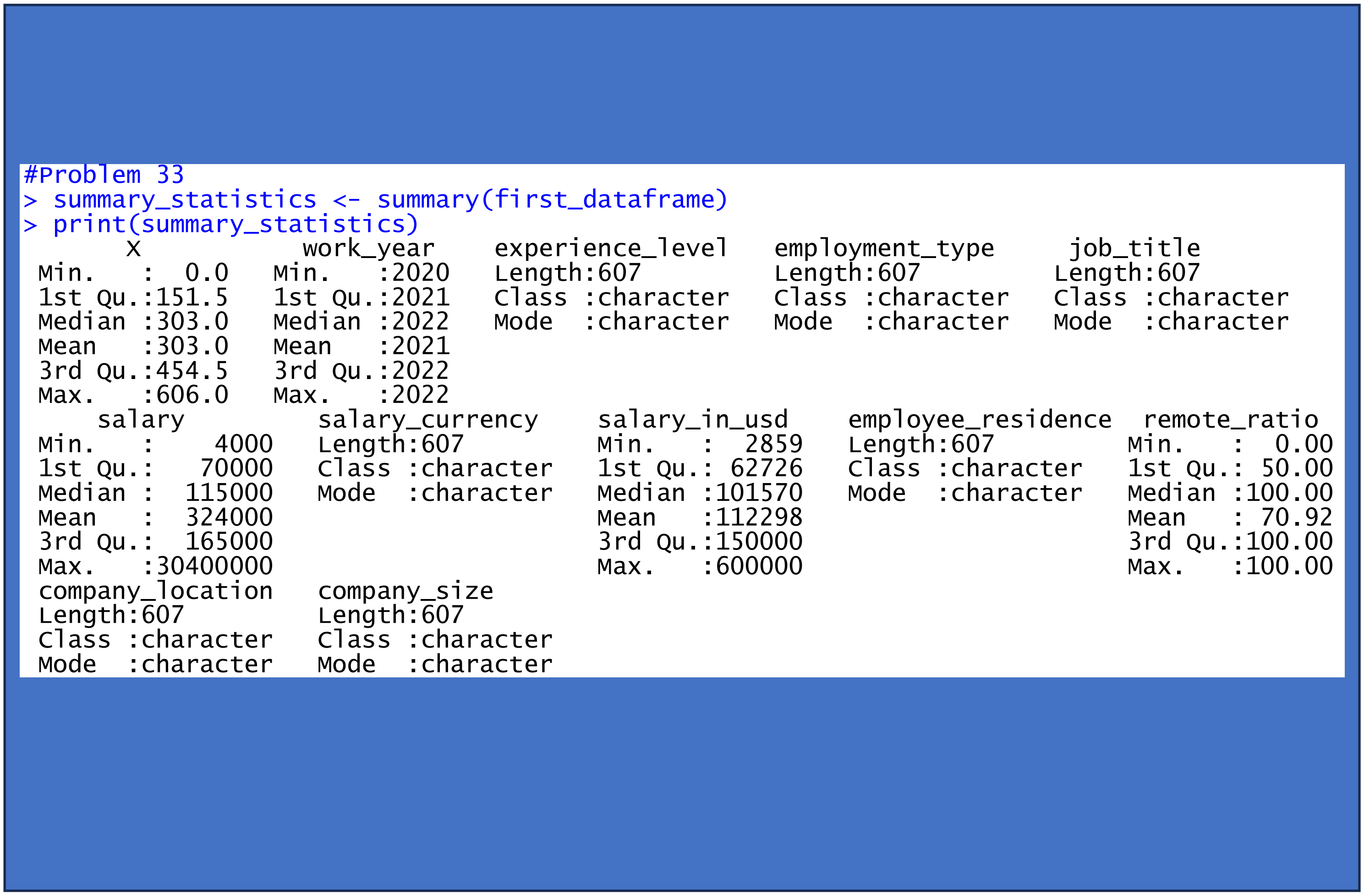
Demonstrated in the R Source file

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.**

Also demonstrated in the R Source file

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

Summary included below-



```
#Problem 33
> summary_statistics <- summary(first_dataframe)
> print(summary_statistics)
       X           work_year    experience_level   employment_type     job_title
 Min.   :  0.0   Min.   :2020   Length:607         Length:607         Length:607
 1st Qu.:151.5   1st Qu.:2021   Class :character   Class :character   Class :character
 Median :303.0   Median :2022   Mode  :character   Mode  :character   Mode  :character
 Mean   :303.0   Mean   :2021
 3rd Qu.:454.5   3rd Qu.:2022
 Max.   :606.0   Max.   :2022
     salary        salary_currency    salary_in_usd    employee_residence  remote_ratio
 Min.   :    4000   Length:607         Min.   :  2859   Length:607         Min.   :  0.00
 1st Qu.:   70000   Class :character   1st Qu.: 62726   Class :character   1st Qu.: 50.00
 Median :  115000   Mode  :character   Median :101570   Mode  :character   Median :100.00
 Mean   :  324000                      Mean   :112298                      Mean   : 70.92
 3rd Qu.:  165000                      3rd Qu.:150000                      3rd Qu.:100.00
 Max.   :30400000                      Max.   :600000                      Max.   :100.00
 company_location   company_size
 Length:607         Length:607
 Class :character   Class :character
 Mode  :character   Mode  :character
```

Conclusion

This report lists the various answers for the problems prescribed to us in the ALY 6000 Introduction to Analytics Module 1 as a Practice in the R studio environment with the R language.

Bibliography

Bluman, A. (2018). *Elementary statistics: A step by step approach* (10th ed.). McGraw Hill.ISBN 13: 978-1-259-755330

Kabacoff, R. I. (2022). *R in action: Data analysis and graphics with R and tidyverse* (3rd ed.). Manning Publications. ISBN 978-1-617-29605-5

Ayala, Professor Maria. (2024, January). Module 1 Assignment — Project 1- R Practice. Canvas. Retrieved

13th January, 2024 from https://northeastern.instructure.com/courses/165041/assignments/2099213