ALY_6000 Project 1 Report

Junchen Yi

College Professional Studies in Analytics, Northeastern University

ALY_6000: Introduction to Analytics

Zhi He

Submission Date: 25/09/2023

Introduction:

In this project, I was asked how to create R scripts using R Studio, learned the basics of the R language, and gained experience in analyzing and visualizing data by writing code.

Key findings:

Basic Knowledge: I initially acquired a grasp of the R language's fundamental syntax and structure. I gained an understanding of key concepts like variables, functions, and vectors, among others.

Data Visualization: While learning, I observed that the R programming language bears similarities to Python, which I had previously studied. Both languages excel in rapid data processing and analysis, offering robust visualization tools like Matplotlib for Python and ggplot2 for R. For this week's assignment, I employed the 'hist' function and the 'ggplot2' package to create two bar graphs.

Reading External Data: Similar to Python, R enables the effortless reading of external files, such as CSV and Excel files. In this particular task, I read a CSV file sourced from the internet and conducted an analysis of its rows and columns.

Coursework Conclusion

Question1:

> 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

Question 14:

second_vector + 20 # It means each element in second_vectors is added by 20.

second_vector * 20 # It means each element in second_vectors is times by 20.

second_vector >= 20 # Determine whether each element in second_vector is greater than or equal to 20.

second_vector != 20 # Determines whether each element in the array is not equal to 20. If it is not equal to 20, the result is true, otherwise it is false.

Question 23:

logical_indices <- c(FALSE, TRUE, FALSE, TRUE)

# Extract elements from 'first_vector' using the logical indices.
vector_from_boolean_brackets <- first_vector[logical_indices]
vector_from_boolean_brackets
logical_indices is a logical vector that specifies which elements to extract from first_vector. The TRUE value represents the element to be extracted, while the FALSE value represents the element to be excluded. Therefore, when assigning the variable vector_from_boolean_brackets, since the values of 12 and 5 correspond to TRUE in logical_indices, the final output is 12 and 5.

Question 24:
second_vector >= 20
Checks each element in second_vector, returning TRUE if the element is greater than 20 and FALSE if it is less than 20.

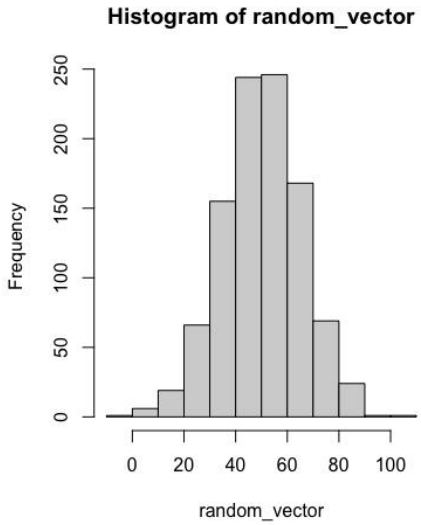
Question 25:
ages_vector <- seq(from = 10, to = 30, by = 2)
# Create a sequence of numbers representing age, starting from 10 and increasing by 2 with each element.

Question 26:
ages_vector [ages_vector >= 20]
# Indicates that ages_vector filters a number of eligible elements on the basis of the original, each of which is greater than or equal to 20.

Question 30:
Answer: 200.2145 685.2186 916.8758 284.3995 104.6501 701.0575 527.9600 807.9352 956.5001 110.4530
set.seed () is used to set a random number seed, a specific seed can produce a specific random sequence, the main purpose of this function is to make the simulation repeatable. runif() function generates a vector of 10 random numbers from a uniform distribution between 0 and 1000. Because the seed is set to 5, if we run this code again with the same seed, we will get the same set of random numbers.

Question 37:
#This vector contains 1000 random numbers generated from a normal distribution with a mean of 50 and a standard deviation of 15. Since we set the seed to 5, if we run this code again with the same seed, we will get the same set of random numbers.

Question 38:
Explanation:
X-axis: The X-axis of the histogram represents the range of values in a random vector and is divided into different intervals.
Y-axis: The Y-axis represents the frequency or count of values falling between each bin. It represents how many values in the random vector fall into each interval.



Question 42:

```
head(first_dataframe)
# Running head(first_dataframe) will display the first 6 rows of the first_dataframe (by default) and the value of each of its columns.

head(first_dataframe, n = 7)
# n = 7 means this function used to view the first 7 rows of the data frame "first_dataframe".

names(first_dataframe)
# The function names(first_dataframe) function is used to retrieve the names of the columns, which are variables of the data frame.

smaller_dataframe <- select(first_dataframe, job_title, salary_in_usd)
# Create a new data frame called smaller_dataframe by selecting specific three columns from the original data frame first_dataframe.

smaller_dataframe
# Display contents of the new dataFrame.

better_smaller_dataframe <- arrange(smaller_dataframe,
                                    desc(salary_in_usd))
better_smaller_dataframe
# The new dataframe will contain the same columns as the smaller_dataframe, and the rows will be sorted in decreasing order according to the salary_in_usd column, the rows with the highest salaries at the top.

better_smaller_dataframe <- filter(smaller_dataframe, salary_in_usd > 80000)
better_smaller_dataframe
# The better_smaller_dataframe filters the rows in the smaller_dataframe by keeping only those rows with values greater than 80000 in the salary_in_usd column.
```

```
better_smaller_dataframe <-
   mutate(smaller_dataframe, salary_in_euros = salary_in_usd * .94)
better_smaller_dataframe
```

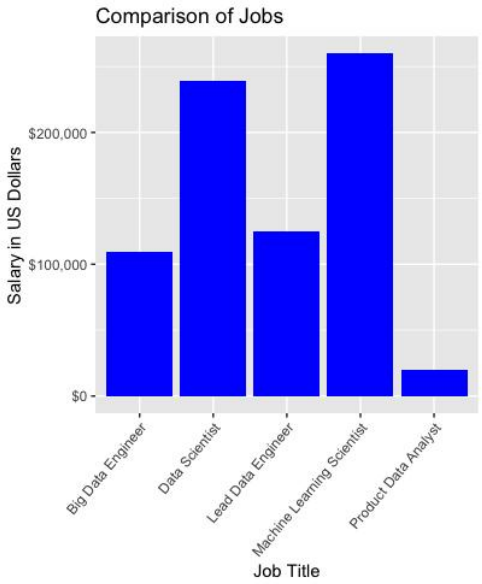
The mutate() function is used to create a new dataframe and convert the value of the "salary_in_usd" column to the "salary_in_euros" column, and multiply each value in the column by 0.94.

```
better_smaller_dataframe <- slice(smaller_dataframe, 1, 1, 2, 3, 4, 10, 1)
better_smaller_dataframe
```

create a new data frame named better_smaller_dataframe by selecting specific rows from the original data frame.

```
ggplot(better_smaller_dataframe) +
   geom_col(mapping = aes(x = job_title, y = salary_in_usd), fill =
                    "blue") +
   xlab("Job Title") +
   ylab("Salary in US Dollars") +
   labs(title = "Comparison of Jobs ") +
   scale_y_continuous(labels = scales::dollar) +
   theme(axis.text.x = element_text(angle = 50, hjust = 1))
```

#The code generates a bar chart with the x-axis representing the job title and the y-axis representing the dollar salary.The bars are filled in blue. In addition, it rotates the labels by 50 degrees and aligns them horizontally to the right.



Works Cited

[1] R get started. (n.d.). https://www.w3schools.com/r/r_get_started.asp

[2] YouTube. (2019, June 6). R programming tutorial - LEARN THE BASICS of statistical computing. YouTube. https://www.youtube.com/watch?v=_V8eKsto3Ug