Data Analytics 344

Tutorial Test 1
Department of Engineering
Stellenbosch University
26 July 2023

Instructions

Welcome to the first tutorial test! This tutorial and future tutorials will be auto-graded using test cases. To ensure your submission can be graded, please follow these guidelines:

1. Adding your student number: In the first code block, please store your student number in the variable student_number. For instance if your student number is 1234, the code block should contain the code student_number <- 1234
2. Package Usage: Only use the packages specified in the setup code block. If you need additional packages, install them using the R Console.
3. Hands-Off Code Blocks: Do not modify code blocks that start with the comment #DO NOT EDIT. These blocks are for loading packages or illustrating programming concepts and should remain unchanged.
4. Completing Code Blocks: Your main task is to complete the incomplete code provided. There's no need to add or modify any of the code blocks or questions; simply fill in the necessary answers.
5. Error-Free Submission: Before submitting your final answer, make sure your R Markdown document runs without errors by pressing the knit button. Only variables declared in this document will be accessible for grading.
6. Case Sensitivity: Remember that variables in R are case-sensitive. Use the exact variable names specified in the questions (e.g., Q1, not q1) and do not overwrite your answers in subsequent code blocks.
7. Submission: Your final submission should consist of a single .Rmd file. Name your file as ???????.Rmd where the question marks should be replaced with your student number.

Please adhere to these guidelines, as submissions that do not follow them cannot be graded. Following the above principles will also help you to learn how to produce reproducible code.

Introduction

In this case study, we will work with Formula 1 (F1) data to showcase the process of creating a dataframe from different data types.

Our final dataframe will consist of the following columns:

```
Driver: The name of the driver
Lap: The lap number
Pitstop: Whether the driver stopped for a pitstop during the race
LapTime: The time taken to complete the lap
```

The dataframe will contain 20 drivers and 57 laps, resulting in a total of 20 * 57 = 1140 rows. You can assume that the data will be ordered by driver and lap. In other words, the first row will contain the lap time for the first driver for lap 1, the second row will contain the lap time for the second driver for lap 1.

Question 1: Creating a character vector

We will start by creating a character vector that contains the driver names. Please create a character vector that contains the names: verstappen, leclerc, perez, sainz, hamilton, russell, alonso, bottas, stroll, norris, ocon, albon, sargeant, hulkenberg, tsunoda, piastri, zhou, kevin_magnussen, gasly, and de_vries. Store your answer in the variable q1.

```
# Your answer here
q1 <- c("verstappen", "leclerc", "perez", "sainz", "hamilton", "russell",
        "alonso", "bottas", "stroll", "norris", "ocon", "albon", "sargeant",
        "hulkenberg", "tsunoda", "piastri", "zhou", "kevin_magnussen", "gasly", "de_vries")
q1
```

```
##  [1] "verstappen"      "leclerc"         "perez"           "sainz"
##  [5] "hamilton"        "russell"         "alonso"          "bottas"
##  [9] "stroll"          "norris"          "ocon"            "albon"
## [13] "sargeant"        "hulkenberg"      "tsunoda"         "piastri"
## [17] "zhou"            "kevin_magnussen" "gasly"           "de_vries"
```

Question 2: Creating an integer vector

For the column Lap, we want to add the lap number for each of the 57 laps. Create an integer vector that represents the lap numbers from 1 to 57 and store your answer in the variable q2.

```
# Your answer here
q2 <- c(1L:57L)
```

Question 3: Creating our first data frame

Now that we have created the drivers vector and the lap vectors, we will proceed to create a data frame. Create the data frame q3. The first column of the data frame should be called Driver and the second column of the data frame should be called Lap.

Hint: Your final dataframe should contain 1140 rows and be structured according to the previous instructions.

```
# Your answer here
q3 <- data.frame(Driver = rep(q1, 57), Lap = rep(q2, each = 20))
```

Question 4: Creating a logical vector for pitstops

Usually in a race, each driver will stop two or three times to change their tyres. To create our column Pitstop, we will generate a logical vector that contains 40 true values. Create a logical vector called q4, that contains 1140 values, where the first 40 values should be true.

```
# Your answer here
q4 <- c(rep(TRUE, 40), rep(FALSE, 1140-40))
```

Question 5: Shuffling a vector

Since we do not expect our drivers to stop during the first two laps, expect if their is a major crash, we will shuffle the values of the vector pitstop. Use the function `sample` to shuffle the values stored in the vector q4, Store your answer in the vector q5. Do not change any of the default arguments.

```
# Your answer here
q5 <- sample(q4)
```

Question 6: Adding a vector to a dataframe

Add the vector q5 to the dataframe stored in the variable q3 using the column heading Pitstop. Store your new dataframe in the variable q6.

```
# Your answer here
q6 <- data.frame(q3, Pitstop = q5)
head(q6)
```

```
##       Driver Lap Pitstop
## 1 verstappen   1   FALSE
## 2    leclerc   1   FALSE
## 3      perez   1   FALSE
## 4      sainz   1   FALSE
## 5   hamilton   1   FALSE
## 6    russell   1   FALSE
```

Question 7: Generating random laptimes

Use the run_if function to generate random lap_times between 80 and 120 for all 1140 laps seconds by setting the min argument to 80 and the maximum argument to 120. Create a new data frame called q7, adding your new column to the data frame created in q6. The new column should have the heading Laptime

```
# Your answer here
q7 <- data.frame(q6, Laptime = runif(1140, 80, 120))
view(q7)
```

Question 8: Determine the winner

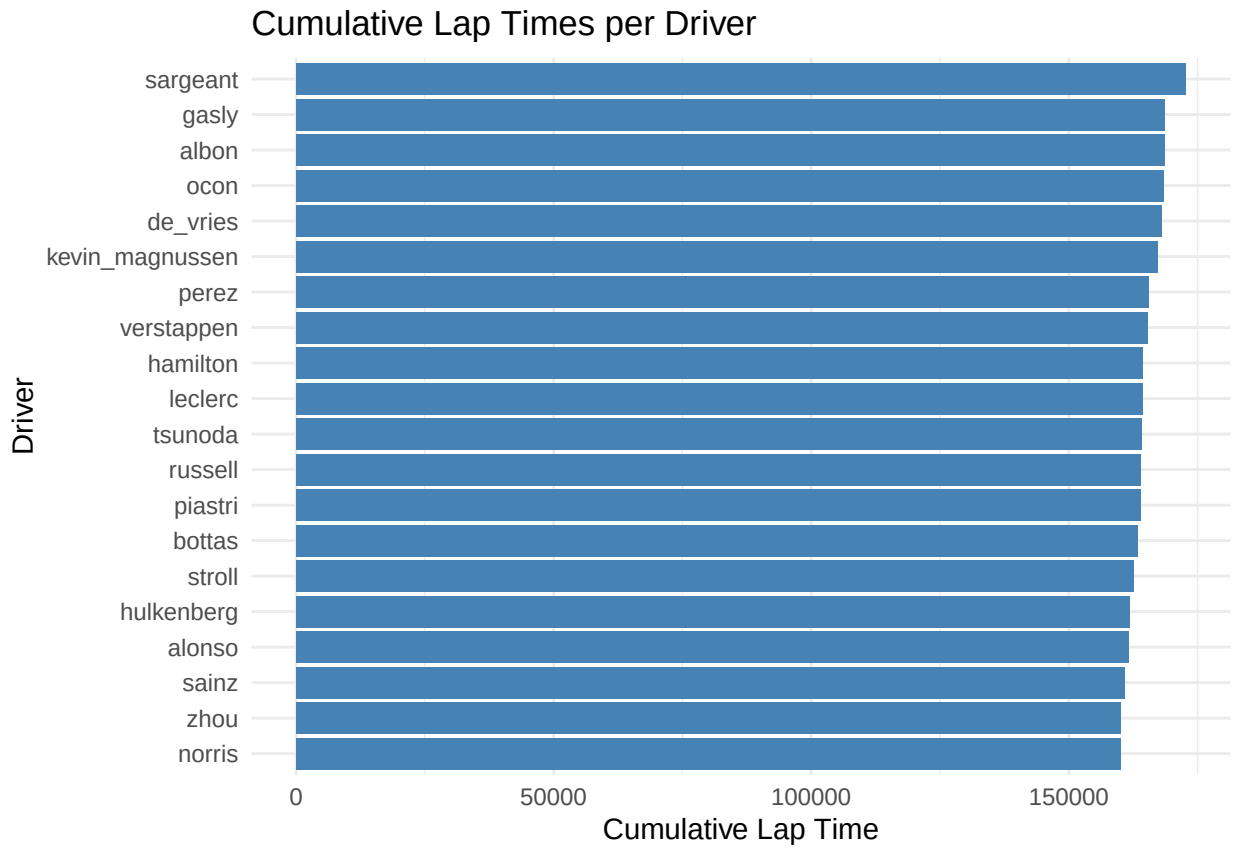
Question 8 will count 50% of the tutorial mark. Clear your global environment and run each code block once starting from the top, making sure that you filled in your student_number. If the data frame that you have created in q7 is correct, then the code block below will generate a graph.

```
# Do not edit
library(dplyr)

df <- q7 %>%
  group_by(Driver) %>%
  arrange(Lap) %>%
  mutate(CumulativeLapTime = cumsum(Laptime))

# Step 2: Create the ggplot horizontal bar plot
library(ggplot2)

ggplot(df, aes(x = CumulativeLapTime, y = reorder(Driver, CumulativeLapTime))) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Cumulative Lap Times per Driver",
       x = "Cumulative Lap Time",
       y = "Driver") +
  theme_minimal()
```



Using the above graph, determine who won the race i.e. the driver with the smallest cumelative lap times. Store the name of the winner in a character vector q8. Use the names as defined in Question 1. For example, if the winner is Max Verstappen, you answer should be q8 < c("verstappen")

```
# Your answer here
q8 <- c("norris")
```