# DSI Data Science: Introduction to R Assessment 2

## Introduction and Goals

During the duration of the course, we had the opportunity to explore the RStudio environment, where we could navigate through its features. We also had the chance to download the tidyverse package (Wickham, H. et al, 2019) and delve into its functions, which enabled us to manipulate, wrangle, and visualize data effectively. Moreover, we gained knowledge about crafting personalized functions, as well as exploring tools and techniques for functional programming and data simulation. Additionally, we touched upon significant topics such as reproducibility, bias, diversity, inclusion, ethical considerations, equity concepts, data security, and best coding practices. The purpose of this assessment is to ensure that learners acquire the following skills:

- Proficiently manipulate tabular data using *dplyr*: A Grammar of Data Manipulation.
- Apply manipulation and wrangling techniques to reshape data effectively.
- Create meaningful visualizations suitable for different types of data using *ggplot2*: A Grammar of Graphics.
- Develop custom functions capable of analyzing input data and generating output.
- Understand the importance of equity, diversity, and inclusion (EDI) practices in the field of data sciences.

**The questions in the assessment carry a total of 30 marks, while an additional 3 marks are allocated for formatting (see Submission Instructions on page 4).**

## Tasks

Please respond to all the questions sequentially. You have the option to utilize a PDF editor, the provided R markdown template, or any other platform/software of your preference to generate the PDF document containing both the questions and answers. Alternatively, you can open your R script using a text editor and export it as a PDF document. **When submitting your answers, make sure to include both the question number and the question itself, along with your responses.** This requirement is in place to ensure that no questions are skipped and that all sub-questions within each main question are addressed. When copying and pasting the questions, you may disregard any formatting such as italics.

For questions 1 and 2, we will use the **built-in dataset *mtcars* from R base package datasets**.To access this dataset, type the following in your script:

```
data(mtcars)
?mtcars #to view data format and understand column names
```

1. [10 marks Total]

   (a) [5 marks] Please create a custom function called "mpgFilter" using both tidyverse functions and other base R functions if needed. This function should have two arguments: "dataset" and "minMpg". Within the function, utilize dplyr::filter to filter the user-provided dataset based on the specified **minimum** mpg (miles/gallon) in the "minMpg" argument. Finally, return the resulting output to the user within the function. Please include all the necessary code.

```
mpgFilter <- function(dataset, minMpg){
  output <- filter(dataset, dataset$mpg >= minMpg)
  return(output)
}

## 1 marks for line 1 ; function name and correct arguments
## 3 marks for line 2; correct use of filter function, and correct arguments
## 1 1 mark for line 2; correct return
```

(b) [2 mark] Run the mpgFilter function you wrote using the built-in dataset mtcars. Set the dataset argument to mtcars and minMpg argument to 20. Save the resulting output from function mpgFilter into an object called output_mpgFilter. How many rows and columns are in the resulting data frame, i.e., the dimensions of output_mpgFilter? Please include all the necessary code.

```
output_mpgFilter <- mpgFilter(mtcars, 20)
dim(output_mpgFilter)
```

```
## [1] 14 11
```

```
## 1 mark for correctly saving output
## 1 marks for correct row and column
```

c) [3 marks] Using the "output_mpgFilter" object obtained previously, mutate it by adding a new column utilizing a dplyr function. This new column should display the ratio of mpg (miles/gallon) to hp (gross horsepower) from the mpg dataset. You can achieve this by dividing the mpg by hp. Name for the new column "Ratio.MPGHP". Please include all the necessary code.

```
output_mpgFilter %>%
  mutate(Ratio.MPGHP = mpg / hp)
```

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb Ratio.MPGHP
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4   0.1909091
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4   0.1909091
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1   0.2451613
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1   0.1945455
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2   0.3935484
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2   0.2400000
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1   0.4909091
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2   0.5846154
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1   0.5215385
## Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1   0.2216495
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1   0.4136364
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2   0.2857143
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2   0.2690265
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2   0.1963303
```

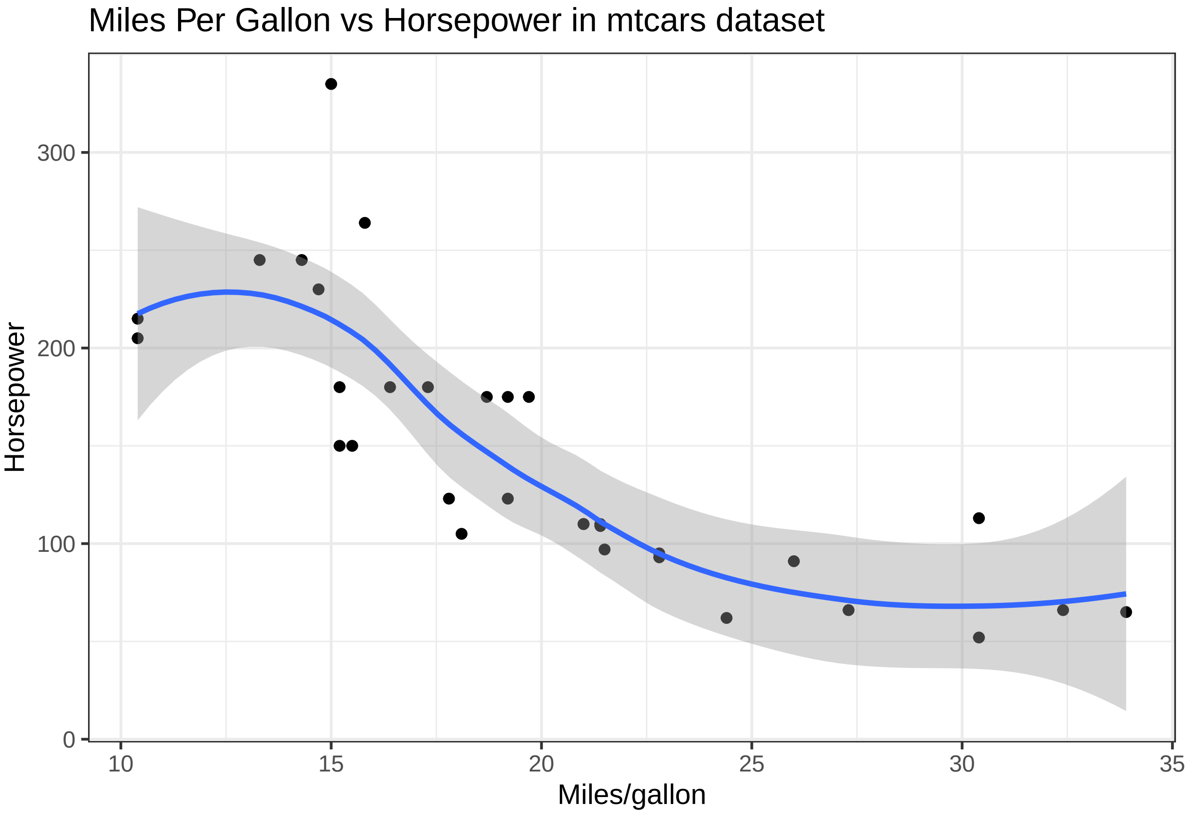
```
## 1 mark for using mutate
## 1 mark for labeling the new column correctly
## 1 marks for correct output
```

2. [10 marks] Generate a plot for the mtcars dataset using the ggplot2::ggplot function. Customize the plot by incorporating layers from the ggplot2 package. This customization should include an informative title, clear labels, axes, color and/or size, and an overall appealing appearance. Please include the code for generating the plot (8 marks). Finally, provide the created plot along with a detailed description, consisting of 3 to 4 sentences, explaining what the plot represents (2 marks).

   Example

```
mtcars %>%
  ggplot(aes(x = mpg,
             y = hp),
             size = 3,
             col = "blue") +
  geom_point() +
  geom_smooth() +
  labs(x = "Miles/gallon",
       y = "Horsepower",
       title = "Miles Per Gallon vs Horsepower in mtcars dataset") +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
##full marks for title and labeled axes, color and/or size, overall appearance
```

3. [10 marks] To ensure equity, diversity, and inclusion (EDI) practices in your data science work, what steps can you take? Provide examples. Additionally, what challenges might you encounter or circumstances that could be beyond your control? Please elaborate on this topic using approximately half to three-quarters of the page.

   *#Full marks given to anything that is coherent and well thought out.*

## Grading Scheme

The mark assigned for each question is indicated with the question. Use that to guide the answers. There will be additional marks assigned for submitting the Assessment in correct format.

## Submission Instructions

[1 mark] Remember: Submissions should only be in PDF format. When emailing the instructor, name PDF using format: LASTNAME_FirstInitial_A2.PDF.

E.g., GALLUCCI_J_A2.PDF.

[2 marks] In your submission to the instructor, you must provide both the question number AND the question,in addition to your answers. Answer all the questions, in order.

## How to Use R Markdown To Create A PDF With Answers

R Markdown is a formatting system that enables the creation of reproducible and dynamic reports using R. These reports allow for the inclusion of R code and its corresponding results in various formats such as slideshows, PDFs, HTML files, Word documents, and more. If you opt to use R Markdown to generate PDF files with solutions, it will require some time to familiarize yourself with R Markdown's syntax and capabilities. To simplify this process, a template named '**Assessment2_template.Rmd**' has been provided, which you can utilize. To locate the provided R Markdown template, please navigate to the Assessments folder on GitHub: https://github.com/UofT-DSI/04-intro_r/tree/main/Assessments

Open this template file in RStudio

In order to generate PDF documents from R Markdown, two essential components are required:

- The "rmarkdown" R package
- The LaTeX distribution.

Note: There are various LaTeX options available, such as MiKTeX, MacTeX, TeX Live, and TinyTeX. You have the freedom to choose any of these LaTeX distributions that suits your needs. In this instance, I will demonstrate the use of TinyTeX. To install TinyTeX, you can utilize the "tinytex" R package.

```r
# install the rmarkdown package

#install.packages("rmarkdown")

#library("rmarkdown")

# install the tinytex package

#install.packages("tinytex")

#library("tinytex")

# to install TinyTeX

#tinytex::install_tinytex()
```

The provided template already includes all the question numbers and the corresponding questions. You will only need to insert your answers. Before you begin entering your answers, please follow these steps:

1. Open the file 'Assessment2_template.Rmd' in RStudio.
2. Locate the 'Knit' icon in RStudio.
3. Click on the 'Knit' icon and select 'Knit to PDF' to generate a PDF output.

You are recommended to add small chunks of code at a time and 'Knit' the document. For more information including basics, see https://rmarkdown.rstudio.com/lesson-1.html or seek help during tutorial early.