Methods Camp Assignment 2 Answers

Day 2

2025 Methods Camp Instructors

Warm-Up

Before jumping in to the assignment, let's review what we learned about translating math notation into R code. Implement the following functions in R:

Function 1

$$f(x) = \begin{cases} x^2 - 1 & \text{if } x < 0 \\ |x - 3| & \text{if } x \geq 0 \end{cases}$$

Test your function by plotting it over the range -20 to 20:

Function 2

For the following, don't use the built-in `max()` function but instead implement your own!

$$\max_{x \in D} f(x)$$

Test your function by finding the maximum of the following random vector. Check your answer using `max()`.

```
set.seed(123)
D = runif(n=100, min=-100, max=100)
```

Main Assignment

For today's assignment we will be using more of the `mtcars` data set to have you create and save multiple data visualizations in a for loop.

By the end of this task, your code will use a loop to create 5 different plots and save them automatically as image files.

Unlike yesterday's assignment, which had many smaller discrete data exploration tasks, today's assignment is going to be one larger task.

For this assignment, we are asking that you try only working together among yourselves for the first 30 minutes and consult AI for debugging at the latter 30 minutes if needed!

> ⚠️ Warning
>
> To show us your plots on your gradescope submission, please input your image files into your rendered quarto PDF using `![]()` format. Please see this link here for more details.

The Task

We will write a for loop that creates scatter plots comparing `mpg` against 5 different variables `wt`, `hp`, `disp`, `drat` and `qsec`. But first, we have to construct our dataset.

Our data has been split into two dataframes: `mtcars_names` and `mtcars_values`. The first contains the car names and IDs, while the second contains the values for each variable.

Load the dataframes and merge them into a single dataframe called `mtcars`. The `car_id` column in the `mtcars_names` is a combination of the `car_number` and `year` columns from `mtcars_values`. Use `paste0()` and `str_pad()` to create the `car_id` column in `mtcars_values` before merging.

Now, we have to convert our `mtcars` dataframe into a wide format so that we can easily plot the variables against `mpg`. Use the `pivot_wider()` function from the `tidyverse` package to reshape the data so each variable (`mpg`,`wt`, `hp`, `disp`, `drat`, `qsec`) is in its own column.

Now, we're ready to create our plots!

Your code will:

1. Use a for loop to iterate through the 5 variables
2. Create a scatter plot for each variable against mpg
3. Add appropriate titles and axis labels
4. Save each plot as a png file with a descriptive filename

5. Include a different color for each plot

Requirements for Each Plot

- **X-axis:** The variable from your list (wt, hp, disp, drat, or qsec)
- **Y-axis:** mpg (miles per gallon)
- **Title:** Something descriptive like "MPG vs Weight"
- **Axis labels:** Use the descriptive labels from var_labels
- **Points:** Use the color specified in the colors vector
- **File name:** Save as "mpg_vs_[variable_name].png" (e.g., "mpg_vs_wt.png")

A rough template is provided below:

```
# Define the variables you want to plot against mpg

# Define nice labels for your variables

# Define colors for each plot

# Your for loop goes here

  # Create your ggplot here
  # Save your plot here
```

> 💡 **Hint**
>
> If you're struggling with how reference a variable indirectly in a loop, see the "indirection" section of this dplyr page here!

Add your finished images below using the `![]()` format.

Challenge Extension (if you have time!)

Create box plots that compare average `mpg`, `hp`, and `wt` across different categories (`cyl_group` and `transmission`) using nested loops.

Your code will:

1. Create a nested loop that has:

- an outerloop that goes through each categorical variable (`cyl_group`, `transmission`)
- an inner loop that creates a boxplot for each performance variable

2. Saves each plot with a descriptive file name.

Requirements for Each Plot

- **Boxplot:** Use `geom_boxplot()` to show the distribution
- **Title:** Include a clear title like "Miles per Gallon by Number of Cylinders"
- **File name:** Save as "[category]_[variable_name].png" (e.g., "transmission_wt.png")

A rough template is provided below:

```
# Your challenge: Use a loop to calculate and save summary stats

  # Calculate mean, median, and count for each group
  # Save as a CSV file
  # Print a summary message

# Nested plotting loop

    # Create your boxplot here

    # Create filename and save
```

Add your finished images below using the `![]()` format.