Programming Homework 4

Trever Yoder

Task 1: Conceptual Questions

Load Packages and Create a List of Questions

```
library(tidyverse)
library(httr)
library(jsonlite)
```

```
#Create a list with the requested questions
my_list1 <- list("1. What is the purpose of the lapply() function?
                 What is the equivalent purrr function?",
"2. Suppose we have a list called my_list.
Each element of the list is a numeric data frame (all columns
are numeric). We want use lapply() to run the code
cor(numeric_matrix, method = kendall)
on each element of the list. Write code to do this below!
(I'm really trying to ask you how you specify
method = kendall when calling lapply())",
"3. What are two advantages of using purrr functions
instead of the BaseR apply family?",
"4. What is a side-effect function?",
"5. Why can you name a variable sd in a function and
not cause any issues with the sd function?")

#This prints poorly, so it has been left out.
```

Question 1

`lapply()` is used to apply functions across many rows/columns and the "l" ensures R always outputs a list.

Question 2

```r
#create data frames
df1 <- data.frame(a = c(1:3), b = c(4:6))
df2 <- data.frame(x = c(7:9), y = c(10:12), z = c(13:15))

#create list
my_list <- list(df1, df2)

#write/show lapply function
lapply(my_list, cor, method = "kendall")
```

```
[[1]]
  a b
a 1 1
b 1 1

[[2]]
  x y z
x 1 1 1
y 1 1 1
z 1 1 1
```

Question 3

Two advantages of using `purr` functions instead of `BaseR` apply family are: 1. Greater consistency between functions. For example, you can predict the output type exclusively from the function name, which isn't always true for `BaseR` apply functions. 2. `Purr` also has some functions to fill in some gaps such as `imap()` where you can map simultaneously over `x` and its indices.

Question 4

A side-effect function does something beyond it's function return value. For example, write files to a disk. If we want the side effect of `hist` (which is the visual part) we can use the `walk()` function to only print the histogram.

Question 5

We can name a variable `sd` in a function and not cause any issues with the `sd` function because functions have their own temporary environment. So, once the function executes, all variables within that function are "gone" or in other words, not saved to the main environment.

Task 2: Writing R Functions

Question 1

Here I will create a function that calculates RMSE. There is an ellipses in the function to allow for additional arguments.

```
getRMSE <- function(response, predicted, ...){
  add <- list(...)
  remove <- isTRUE(add$na.rm)
  if (remove) {
    dropmissing <- !is.na(response) & !is.na(predicted)
    response <- response[dropmissing]
    predicted <- predicted[dropmissing]
  }
  sqrt(mean((response-predicted)^2))
}
```

Question 2

Let's run some code to create some response values and predictions.

```
set.seed(10)
n <- 100
x <- runif(n)
resp <- 3 + 10*x + rnorm(n)
pred <- predict(lm(resp ~ x), data.frame(x))

#now let's test our RMSE function using this data!
getRMSE(resp, pred)
```

```
[1] 0.9581677
```

```
#Add 2 missing vlaues to resp
resp[c(1,5)] <- NA_real_

#Test with and without specifying what R should do with the missing values
getRMSE(resp, pred)
```

```
[1] NA
```

```
getRMSE(resp, pred, na.rm = TRUE)
```

```
[1] 0.9646971
```

Question 3

Let's create a similar function, except it calculates the MAE instead of the RMSE.

```
getMAE <- function(response, predicted, ...){
  add <- list(...)
  remove <- isTRUE(add$na.rm)
  if (remove) {
    dropmissing <- !is.na(response) & !is.na(predicted)
    response <- response[dropmissing]
    predicted <- predicted[dropmissing]
  }
  mean(abs(response-predicted))
}
```

Question 4

Now let's create some data and test our MAE function using the data.

```
set.seed(10)
n <- 100
x <- runif(n)
resp <- 3 + 10*x + rnorm(n)
pred <- predict(lm(resp ~ x), data.frame(x))

#Let's test our MAE function
getMAE(resp, pred)
```

```
[1] 0.8155776
```

```
#Add 2 missing vlaues to resp
resp[c(1,5)] <- NA_real_

#Test with and without specifying what R should do with the missing values
getMAE(resp, pred)
```

```
[1] NA
```

```
getMAE(resp, pred, na.rm = TRUE)
```

```
[1] 0.8210863
```

Question 5

Now we want to create a wrapper function that can be used to get either both metrics or a single functioned called.

```
#Create the wrapper function
my_wrapper <- function(response, predicted, metrics = c("MAE", "RMSE"), ...) {
  if (!(is.vector(response) && is.atomic(response) && is.numeric(response))) {
    message("Error: 'response' must be a numeric atomic vecotr.")
    return(invisible(NULL))
  }
  if (!(is.vector(predicted) && is.atomic(predicted) && is.numeric(predicted))) {
    message("Error: 'predicted' must be a numeric atomic vecotr.")
    return(invisible(NULL))
  }

  #create empty list for results
  results <- list()

  #calculate metrics
  if ("MAE" %in% toupper(metrics)) {
    results$MAE <- getMAE(response, predicted, ...)
  }
  if ("RMSE" %in% toupper(metrics)) {
    results$RMSE <- getRMSE(response, predicted, ...)
  }
```

```
  return(results)
}
```

Question 6

Now let's test our wrapper function in a similar way that we tested our previous functions.

```
#create data
set.seed(10)
n <- 100
x <- runif(n)
resp <- 3 + 10*x + rnorm(n)
pred <- predict(lm(resp ~ x), data.frame(x))

#call individually then together
my_wrapper(resp, pred, metrics = "MAE")
```

```
$MAE
[1] 0.8155776
```

```
my_wrapper(resp, pred, metrics = "RMSE")
```

```
$RMSE
[1] 0.9581677
```

```
my_wrapper(resp, pred, metrics = c("MAE", "RMSE"))
```

```
$MAE
[1] 0.8155776

$RMSE
[1] 0.9581677
```

```
#test again but with 2 NA values in resp
resp[c(1,5)] <- NA_real_
my_wrapper(resp, pred, metrics = "MAE")
```

```
$MAE
[1] NA
```

```
my_wrapper(resp, pred, metrics = "RMSE", na.rm = TRUE) #tested excluding NA
```

```
$RMSE
[1] 0.9646971
```

```
my_wrapper(resp, pred, metrics = c("MAE", "RMSE"))
```

```
$MAE
[1] NA

$RMSE
[1] NA
```

```
#Test by passing a data frame created in Task 1 Question 2.
my_wrapper(df1, df1, metrics = c("MAE", "RMSE"))
```

```
Error: 'response' must be a numeric atomic vecotr.
```

```
my_wrapper(resp, df1, metrics = c("MAE", "RMSE"), na.rm = TRUE)
```

```
Error: 'predicted' must be a numeric atomic vecotr.
```

Task 3: Querying an API and a Tidy-Style Function

Question 1

Let's query the newsapi for articles on war.

```
URL_ids <- "https://newsapi.org/v2/everything?q=war&from=2025-06-05&sortBy=popularity&apiKey=
id_info <- GET(URL_ids)

#look at the structure
str(id_info, max.level = 1)
```

```
List of 10
 $ url        : chr "https://newsapi.org/v2/everything?q=war&from=2025-06-05&sortBy=popularit
 $ status_code: int 200
 $ headers    :List of 15
  ..- attr(*, "class")= chr [1:2] "insensitive" "list"
 $ all_headers:List of 1
 $ cookies    :'data.frame':    0 obs. of  7 variables:
 $ content    : raw [1:76820] 7b 22 73 74 ...
 $ date       : POSIXct[1:1], format: "2025-06-25 22:37:43"
 $ times      : Named num [1:6] 0 0.0808 0.1135 0.1579 0.4591 ...
  ..- attr(*, "names")= chr [1:6] "redirect" "namelookup" "connect" "pretransfer" ...
 $ request    :List of 7
  ..- attr(*, "class")= chr "request"
 $ handle     :Class 'curl_handle' <externalptr>
 - attr(*, "class")= chr "response"
```

Question 2

Now let's parse this data so we can actually browse some articles.

```
parsed_data <- fromJSON(rawToChar(id_info$content))
my_tibble <- as_tibble(parsed_data$articles)
my_tibble
```

```
# A tibble: 100 x 8
   source$id $name author title description url   urlToImage publishedAt content
   <chr>     <chr> <chr>  <chr> <chr>       <chr> <chr>      <chr>       <chr>
 1 wired     Wired Brian~ Elon~ "Donald Tr~ http~ "https://~ 2025-06-05~ "Elon ~
 2 wired     Wired Justi~ Taiw~ "Unmanned ~ http~ "https://~ 2025-06-23~ "But U~
 3 wired     Wired Matt ~ Iran~ "Iran is l~ http~ "https://~ 2025-06-18~ "Alima~
 4 wired     Wired Andre~ Trut~ "The socia~ http~ "https://~ 2025-06-22~ "Truth~
 5 wired     Wired Vitto~ Far-~ "The \"App~ http~ "https://~ 2025-06-17~ "A con~
 6 wired     Wired Jake ~ '28 ~ "The Briti~ http~ "https://~ 2025-06-20~ "In 20~
 7 wired     Wired Molly~ The ~ "\"The EPA~ http~ "https://~ 2025-06-11~ "The U~
 8 wired     Wired Boone~ Goog~ "A partner~ http~ "https://~ 2025-06-24~ "With ~
 9 wired     Wired Steve~ What~ "Meta CTO ~ http~ "https://~ 2025-06-20~ "When ~
10 wired     Wired Lily ~ Isra~ "Plus: Ukr~ http~ ""         2025-06-21~ "Amid ~
# i 90 more rows
```

If we want to see a simplified version with just the author and titles, we can do that!

```r
simple_tibble <- my_tibble %>%
    select(author, title)

simple_tibble
```

```
# A tibble: 100 x 2
   author                          title
   <chr>                           <chr>
 1 Brian Barrett                   Elon Musk Is Posting Through It
 2 Justin Ling                     Taiwan Is Rushing to Make Its Own Drones Befor~
 3 Matt Burgess                    Iran's Internet Blackout Adds New Dangers for ~
 4 Andrew Couts, Lily Hay Newman   Truth Social Crashes as Trump Live-Posts Bombi~
 5 Vittoria Elliott, Leah Feiger   Far-Right 'Appeal to Heaven' Flag Flown Above ~
 6 Jake Kleinman                   '28 Years Later' Director Danny Boyle Says Sho~
 7 Molly Taft                      The EPA Wants to Roll Back Emissions Controls ~
 8 Boone Ashworth                  Google Wants to Get Better at Spotting Wildfir~
 9 Steven Levy                     What Lt. Col. Boz and Big Tech's Enlisted Exec~
10 Lily Hay Newman                 Israel Says Iran Is Hacking Security Cameras f~
# i 90 more rows
```

Question 3

Let's write a function that allows the user to easily input any title/date/key. Let's make it only output a tibble with the authors and titles to make it fast and easy to browse the articles. Finally, let's test this by searching for gamestop articles.

```r
API_function <- function(title, date, key) {
  URL_ids <- paste0(
  "https://newsapi.org/v2/everything?",
  "q=", title,
  "&from=", date,
  "&sortBy=popularity",
  "&pageSize=100",
  "&apiKey=", key
)
  id_info <- GET(URL_ids)
  parsed_data <- fromJSON(rawToChar(id_info$content))
  my_tibble <- as_tibble(parsed_data$articles)

  #print title and authors only for fast and simple querying.
  results <- my_tibble %>%
```

```
    select(author, title)
  print(results)
}

#test function
API_function("gamestop", "2025-05-27", "93563897e7a24afaaac43bd50c70d0af")
```

```
# A tibble: 100 x 2
   author                    title
   <chr>                     <chr>
 1 David Imel                A night at New York's biggest Switch 2 launch eve~
 2 Brandon Widder            The Verge's guide to Amazon Prime Day 2025
 3 Ash Parrish               The Switch 2's promising start hides an uncertain~
 4 Kyle Barr                 Target and Walmart Cancel Switch 2 Orders but Hop~
 5 James Pero                Did the Switch 2 Just Accidentally Make Waiting i~
 6 James Pero                Someone Stole 2,810 Nintendo Switch 2's and I Thi~
 7 Luc Olinga                Your Bitcoin Might Soon Get You a Mortgage-
No, Re~
 8 Kyle Barr and Raymond Wong Nintendo Switch 2 Review: The Ultimate Handheld a~
 9 Nicholas Sutrich          Motorola Edge 2025 review: Not fast enough
10 Oscar Gonzalez            You Can't Screenshot Switch 2 Footage From Ninten~
# i 90 more rows
```