Homework 4

Task 1: Conceptual Questions

Question 1: What is the purpose of the lapply() function? What is the equivalent purrr function?

The purpose of the lapply() function is to apply a function to each element in a list in R. The equivalent purrr function is map().

Question 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())

```
# lapply(my_list, cor, method = "kendall")
```

Question 3: What are two advantages of using purrr functions instead of the BaseR apply family?

1. Greater consistency between functions when using the purrr package
2. Almost every purrr function is type stable, meaning you are easily able to predict the type of data output you will receive from the function name.

Question 4: What is a side-effect function?

A side-effect function does not actually change the data, instead the goal is to just produce something. Some examples include; plot(), print(), and write.csv().

Question 5: Why can you name a variable sd in a function and not cause any issues with the sd function?

Variable names within functions create a temperorary environment that does not overwrite other functions.

Task 2: Writing R Functions

Question 1: Write a basic function (call it getRMSE()) that takes in a vector of responses and a vector of predictions and outputs the RMSE.

```r
getRMSE <- function(vector_resp, vector_pred){

  diff_sq <- (vector_resp - vector_pred)^2

  MSE <- mean(diff_sq, na.rm = TRUE)

  RMSE <- sqrt(MSE)

  return(RMSE)

}
```

Question 2: Testing getRMSE()

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

# No NA

getRMSE(resp, pred)
```

```
[1] 0.9581677
```

```r
# With NA

resp_missing <- resp
resp_missing[c(13, 20)] <- NA_real_

getRMSE(resp_missing, pred)
```

```
[1] 0.9536828
```

Question 3: Write a function called getMAE() that follows the specifications of the getRMSE() function.

```r
getMAE <- function(vector_resp, vector_pred){

  abs_diff <- abs(vector_resp - vector_pred)

  MAE <- mean(abs_diff, na.rm = TRUE)

  return(MAE)
}
```

Question 4: Testing getMAE()

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

#No NA

getMAE(resp, pred)
```

```
[1] 0.8155776
```

```r
# With NA

resp_missing <- resp
resp_missing[c(13, 20)] <- NA_real_

getMAE(resp_missing, pred)
```

```
[1] 0.8098271
```

Question 5: Create a wrapper function

```r
wrapper <- function(vector_resp, vector_pred, metrics = c("RMSE", "MAE"))
                    {

  if((!is.vector(vector_resp)) | (!is.atomic(vector_resp))|
     ((!is.numeric(vector_resp))) | (!is.vector(vector_pred))
     | (!is.atomic(vector_pred))| (!is.numeric(vector_pred))){
    return("Both inputs must be numeric atomic vectors.")
  } else{

  result <- list()

  if("RMSE" %in% metrics){
    result$RMSE <- getRMSE(vector_resp, vector_pred)}

  if("MAE" %in% metrics){
    result$MAE <- getMAE(vector_resp, vector_pred)}

 return(result)}

}
```

Question 6: Test wrapper function

```r
set.seed(10)
n <- 100
x <- runif(n)
resp <- 3 + 10*x + rnorm(n)
```

```r
pred <- predict(lm(resp ~ x), data.frame(x))

# No NA (Individual)

wrapper(resp, pred, metric = "RMSE")
```

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

```r
wrapper(resp, pred, metric = "MAE")
```

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

```r
# No NA (Both Metrics)

wrapper(resp, pred)
```

```
$RMSE
[1] 0.9581677

$MAE
[1] 0.8155776
```

```r
# With NA

resp_missing <- resp
resp_missing[c(13, 20)] <- NA_real_

wrapper(resp_missing, pred, metric = "RMSE")
```

```
$RMSE
[1] 0.9536828
```

```r
wrapper(resp_missing, pred, metric = "MAE")
```

```
$MAE
[1] 0.8098271
```

```
wrapper(resp_missing, pred)
```

```
$RMSE
[1] 0.9536828

$MAE
[1] 0.8098271
```

```
# Incorrect data

resp_incorrect <- data.frame(1:100)

wrapper(resp_incorrect, pred)
```

```
[1] "Both inputs must be numeric atomic vectors."
```

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

Question 1:Use GET() from the httr package to return information about a topic that you are interested in that has been in the news lately (store the result as an R object).

The API I chose relates to recent published articles on climate change.

```
climate_change_api <- httr::GET("https://newsapi.org/v2/everything?q=climate%20change&languag
```

Question 2: Parse what is returned and find your way to the data frame that has the actual article information in it (check content). Note the first column should be a list column!

```
climate_parsed <- fromJSON(rawToChar(climate_change_api$content))

climate_articles <- as_tibble(climate_parsed)
```

Question 3: Now write a quick function that allows the user to easily query this API. The inputs to the function should be the title/subject to search for (string), a time period to search from (string - you'll search from that time until the present), and an API key.

```r
api_function <- function(query, from_date, api_key){
  base_url <- "https://newsapi.org/v2/everything"
  url <- paste0(base_url,
                "?q=",
                query,
                "&from=",
                from_date,
                "&apikey=",
                ... = api_key)

  raw_data <- httr::GET(url)

  parsed_data <- fromJSON(rawToChar(raw_data$content))

  data <- as_tibble(parsed_data)

  return(data)
}
```

Testing function on GameStop data

```r
api_key <- "9aacf935958947a9aced2053f23ea00c"
api_function(
  query = "gamestop",
  from_date = "2025-06-19",
  api_key = api_key
)
```

```
# A tibble: 16 x 3
   status totalResults articles$source$id $author      $title $description $url
   <chr>         <int> <lgl>              <chr>        <chr>  <chr>        <chr>
 1 ok               16 NA                 "Adamya Sha~ Someo~ "In a high--~ http~
 2 ok               16 NA                 "HDblog.it"  Furto~ "Probabilme~ http~
 3 ok               16 NA                 "Kamil Świt~ Ukrad~ "To tak dzi~ http~
 4 ok               16 NA                 "Nico Schol~ Switc~ "Ein LKW vo~ http~
 5 ok               16 NA                 "Bublik1"       ~ "      ~ http~
 6 ok               16 NA                 "Davide Leo~ Il re~ "I negozi g~ http~
 7 ok                16 NA                 "      ~    ~ "      ~ http~
 8 ok                16 NA                 "Cointelegr~ Here’~ "While US d~ http~
 9 ok                16 NA                 "Maya Ganda~ The 6~ "Some of th~ http~
10 ok                16 NA                 "MarketBeat~ Virtu~ "Meta Platf~ http~
11 ok                16 NA                 "HashFly"    HashF~ "In 2025, H~ http~
12 ok                16 NA                 "      ~   Nin~ "      ~ http~
13 ok                16 NA                 "Sarwak"        ~ "      ~ http~
14 ok                16 NA                 "Dimitry Ha~ News:~ "Der Launch~ http~
15 ok                16 NA                 "Francesco ~ Xbox ~ "Le inserzi~ http~
16 ok                16 NA                 "Francesco ~ Xbox ~ "Xbox abban~ http~
# i 4 more variables: articles$source$name <chr>, articles$urlToImage <chr>,
#   $publishedAt <chr>, $content <chr>
```