ST558ProgrammingHW4

Holly Probasco

```r
library(tidyverse)
```

Task 1: Conceptual Questions

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 over a list. This is more efficient than using a loop. The equivalent purrr function is map(). ### 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(X = my_list, FUN = function(x) {cor(x, method = "kendall") }) ### 3. What are two advantages of using purrr functions instead of the BaseR apply family?
- One advantage is consistency. Each function has the same arguments, unlike the apply() functions.
- Another is purrr has more options for shorthand anonymous functions, which allows for more simplification. ### 4. What is a side-effect function?
- A side effect function is a function that does not change the data, but just produces something, like print() or plot() ### 5. Why can you name a variable sd in a function and not cause any issues with the sd function?
- Because variables in functions don't overwrite things in the global environment since they are just in the temporary function environment

Task 2: Writing R Functions

1. Write a basic function that takes in a vector of responses and a vector of predictions and outputs the RMSE.

```r
getRMSE <- function(response_vec, prediction_vec,...) {
 mse <- mean((response_vec - prediction_vec)^2, ...)
 rmse <- sqrt(mse)
 return(rmse)
}
```

2. Run the following code to create some response values and predictions.

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

Test your RMSE function using this data.

```r
getRMSE(resp,pred)
```

```
[1] 0.9581677
```

Repeat after replacing two of the response values with missing values (NA_real_). – Test your RMSE function with and without specifying the behavior to deal with missing values.

```r
set.seed(10)
x_resp <- runif(98)
resp <- c(3 + 10*x_resp + rnorm(98), NA_real_, NA_real_)
```

With specifying

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

```
[1] 0.9858791
```

As we can see, removing the NAs is possible because of the ..., and we get basically the same RMSE as before.

Without specifying

```r
getRMSE(resp,pred)
```

```
[1] NA
```

When removing the NA values are not specified, we get an NA due to the mean function

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

```r
getMAE <- function(response_vec, prediction_vec, ...) {
  mae <- mean(abs(response_vec-prediction_vec), ...)

  return(mae)
}
```

4. Run the following code to create some response values and predictions.

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

Test your MAE function using this data.

```r
getMAE(resp,pred)
```

```
[1] 0.8155776
```

Repeat after replacing two of the response values with missing values (NA_real_). – Test your MAE function with and without specifying the behavior to deal with missing values.

```r
set.seed(10)
x_resp <- runif(98)
resp <- c(3 + 10*x_resp + rnorm(98), NA_real_, NA_real_)
```

With specifying

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

```
[1] 0.834818
```

As we can see, removing the NAs is possible because of the …, and we get a very similar MAE as before.

Without specifying

```r
getMAE(resp,pred)
```

```
[1] NA
```

When removing the NA values are not specified, we get an NA due to the mean function

5. Let's create a wrapper function that can be used to get either or both metrics returned with a single function call.

Do not rewrite your above two functions, call them inside the wrapper function (we would call the getRMSE() and getMAE() functions helper functions). When returning your values, give them appropriate names. • The function should check that two numeric (atomic) vectors have been passed (consider is.vector(), is.atomic(), and is.numeric()). If not, a message should print and the function should exit. • The function should return both metrics by default and include names. The behavior should be able to be changed using a character string of metrics to find.

```r
wrapper <- function(response_vec, prediction_vec, method = "Both", ...) {
  if(!(is.vector(response_vec) && is.atomic(response_vec) && is.numeric(response_vec))) {
    print("First input is not a numeric atomic vector") }
else if(!(is.vector(prediction_vec) && is.atomic(prediction_vec) &&
is.numeric(prediction_vec))) {
    print("Second input is not a numeric atomic vector") }
else {
      if(method == "RMSE") {
        result <- getRMSE(response_vec,prediction_vec,na.rm=T)
        return(cat("RMSE", result, sep = " "))
      } else if(method == "MAE") {
        result <- getMAE(response_vec,prediction_vec,na.rm=T)
        return(cat("MAE", result, sep = " "))
      }
      names <- c("RMSE","MAE")
      RMSE <- getRMSE(response_vec,prediction_vec,na.rm=T)
      MAE <- getMAE(response_vec,prediction_vec,na.rm=T)
      results <- c(RMSE,MAE)
      results_matrix <- matrix(c(names,results),ncol = 2)
      return(results_matrix)
      }
}
```

6. Run the following 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))
```

Test your new function using this data. Call it once asking for each metric individually and once specifying both metrics

Specifying RMSE

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

```
RMSE 0.9581677
```

Specifying MAE

```
wrapper(resp,pred,method = "MAE")
```

```
MAE 0.8155776
```

Both metrics (is the default)

```
wrapper(resp, pred, method = "Both")
```

```
     [,1]   [,2]
[1,] "RMSE" "0.958167655151933"
[2,] "MAE"  "0.815577593682669"
```

- Repeat with replacing two of the response values with missing values (NA_real_).

```
set.seed(10)
x_resp <- runif(98)
resp <- c(3 + 10*x_resp + rnorm(98), NA_real_, NA_real_)
```

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

```
     [,1]   [,2]
[1,] "RMSE" "0.985879109538499"
[2,] "MAE"  "0.834817972288852"
```

- Finally, test your function by passing it incorrect data (i.e. a data frame or something else instead of vectors)

```
testing = data.frame(resp)
wrapper(testing,pred)
```

```
[1] "First input is not a numeric atomic vector"
```

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

This is the API key I got: 0c084c9512084f4b8643b04a090a68e0 ### 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

```
news_url =
"https://newsapi.org/v2/everything?q=pokemon&from=2025-06-22&apiKey=0c084c9512084f4b8643b04a
info <- httr::GET(url=news_url)
str(info, max.level = 1)
```

```
List of 10
 $ url        : chr "https://newsapi.org/v2/everything?q=pokemon&from=2025-06-22&apiKey=0c084
 $ 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 [1:8167] 7b 22 73 74 ...
 $ date       : POSIXct[1:1], format: "2025-06-23 23:33:13"
 $ times      : Named num [1:6] 0 0.00327 0.02783 0.06964 0.19746 ...
  ..- 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"
```

2. Parse what is returned and find your way to the data frame that has the actual article information in it

```r
parsed = jsonlite::fromJSON(rawToChar(info$content))
poke_data = as.tibble(parsed$articles)
```

```
Warning: `as.tibble()` was deprecated in tibble 2.0.0.
i Please use `as_tibble()` instead.
i The signature and semantics have changed, see `?as_tibble`.
```

```r
poke_data
```

```
# A tibble: 9 x 8
  source$id $name  author title description url   urlToImage publishedAt content
  <lgl>     <chr>  <chr>  <chr> <chr>       <chr> <chr>      <chr>       <chr>
1 NA        Comic~ Marc ~ Poke~ Pokemon Go~ http~ https://c~ 2025-06-22~ "Pokem~
2 NA        Comic~ Marc ~ Poke~ Back in Fe~ http~ https://c~ 2025-06-22~ "Back ~
3 NA        Yahoo~ <NA>   Poké~    Niant~ http~ <NA>       2025-06-22~ "If yo~
4 NA        Comic~ Charl~ Ever~ A new week~ http~ https://c~ 2025-06-22~ "A new~
5 NA        Secur~ Pierl~ SECU~ Security A~ http~ https://s~ 2025-06-22~ "SECUR~
6 NA        Secur~ Pierl~ Secu~ A new roun~ http~ https://s~ 2025-06-22~ "SECUR~
7 NA        Inter~ Joann~ Odzi~ Uniqlo pon~ http~ https://i~ 2025-06-22~ "T-shi~
8 NA        Bleed~ Rich ~ The ~ The Cancel~ http~ https://b~ 2025-06-22~ "Poste~
9 NA        Secur~ Pierl~ Qili~ Qilin rans~ http~ https://s~ 2025-06-22~ "SECUR~
```

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
query_function <- function(subject, time_from, key) {
  news_url = paste0("https://newsapi.org/v2/everything?q=",subject,"&from=",
time_from,"&apiKey=",key)
info <- httr::GET(url=news_url)
parsed = jsonlite::fromJSON(rawToChar(info$content))
queried_data = as.tibble(parsed$articles)
return(queried_data)
}
```

```r
query_function("celtics", "2025-06-22", "0c084c9512084f4b8643b04a090a68e0")
```

```
# A tibble: 27 x 8
   source$id $name author title description url   urlToImage publishedAt content
   <chr>     <chr> <chr>  <chr> <chr>       <chr> <chr>      <chr>       <chr>
 1 espn      ESPN  "NBA ~ Thre~ The Celtic~ http~ https://a~ 2025-06-22~ "Jun 2~
 2 espn      ESPN  "NBA ~ Game~ The Pacers~ http~ https://a~ 2025-06-22~ "Jun 2~
 3 <NA>      Nbcs~ "Sanj~ Pace~ Indiana Pa~ http~ https://m~ 2025-06-22~ "India~
 4 <NA>      Nbcs~ "Sanj~ Pace~ Indiana Pa~ http~ https://m~ 2025-06-22~ "India~
 5 <NA>      Nbcs~ "Sanj~ Pace~ Indiana Pa~ http~ https://m~ 2025-06-22~ "India~
 6 <NA>      Spor~ "Anth~ No M~ For the fi~ http~ https://w~ 2025-06-22~ "For t~
 7 newsweek  News~ "Rica~ Suns~ The Phoeni~ http~ https://d~ 2025-06-22~ "Ricar~
 8 <NA>      Mund~ "Auto~ El s~ Que Shaqui~ http~ https://w~ 2025-06-22~ "Que S~
 9 <NA>      Spor~ "Lev ~ Kevi~ The Phoeni~ http~ https://w~ 2025-06-22~ "The P~
10 <NA>      Forb~ "Shan~ Hous~ The Kevin ~ http~ https://i~ 2025-06-22~ "PHOEN~
# i 17 more rows
```