Homework-4-Functions-APIs

Task 1: Conceptual Questions

Question 1

lapply() applies a function to each element of a vector or list. The equivalent purrr function is map().

Question 2

```
# lapply with anonymous function
lapply(my_list, FUN = function(df) {
                    cor(df, method = "kendall")
                  })
```

Question 3

purrr functions give a shorthand way to make anonymous functions. purrr gives a bit cleaner / more consistent way to apply functions to objects, especially when using tidyverse (it includes lots of additional helper functions).

Question 4

A side-effect function modifies something outside the function, but doesn't necessarily automatically return the data as well.

Question 5

When you call a function, it creates temporary function environments. R uses lexical scoping, so when you create a new variable like 'sd' in a function, it doesn't overwrite the sd function.

Task 2: Writing R Functions

Question 1

```
# calculates and returns RMSE
getRMSE <- function(resps, preds, ...) {
  squared_errors <- (resps - preds)^2
  return(sqrt(mean(squared_errors, ...)))
}
```

Question 2

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

# call function
getRMSE(resp, pred)
```

```
[1] 0.9581677
```

```
# set 2 values to NA
resp[[1]] <- NA_real_
resp[[2]] <- NA_real_
# redo predictions based on NA vals being included
pred <- predict(lm(resp ~ x), data.frame(x))

# call function with and without specifying whether to disregard NA vals
getRMSE(resp, pred)
```

```
[1] NA
```

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

```
[1] 0.9661358
```

Question 3

```
# calculates and returns MAE
getMAE <- function(resps, preds, ...) {
  absolute_errors <- abs(resps - preds)
  return(mean(absolute_errors, ...))
}
```

Question 4

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

# call function
getMAE(resp, pred)
```

```
[1] 0.8155776
```

```
# set 2 values to NA
resp[[1]] <- NA_real_
resp[[2]] <- NA_real_

# redo predictions based on NA vals being included
pred <- predict(lm(resp ~ x), data.frame(x))

# call function with and without specifying whether to disregard NA vals
getMAE(resp, pred)
```

```
[1] NA
```

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

```
[1] 0.8233219
```

Question 5

```
# wrapper function with default metrics param and ...
getMetrics <- function(resps, preds, metrics = c("RMSE", "MAE"), ...) {
  if (!(is.vector(resps) && is.numeric(resps) && is.vector(preds)
        && is.numeric(preds))) {

    # warning message and return
    message("Both responses and predictions should be numeric vectors")
    return(invisible(NULL))
  }

  results <- list()

  # if RMSE in metrics param calc RMSE
  if ("RMSE" %in% metrics) {
    results$RMSE <- getRMSE(resps, preds, ...)
  }
  # if MAE in metrics param calc MAE
  if ("MAE" %in% metrics) {
    results$MAE <- getMAE(resps, preds, ...)
  }

  return(results)
}
```

Question 6

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

# call function
getMetrics(resp, pred, metrics = "RMSE")
```

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

```r
getMetrics(resp, pred, metrics = "MAE")
```

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

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

```
$RMSE
[1] 0.9581677

$MAE
[1] 0.8155776
```

```r
# set 2 values to NA
resp[[1]] <- NA_real_
resp[[2]] <- NA_real_

# redo predictions based on NA vals being included
pred <- predict(lm(resp ~ x), data.frame(x))

# call function using all combos of metrics param and disregard NA vals
getMetrics(resp, pred, metrics = "RMSE", na.rm = TRUE)
```

```
$RMSE
[1] 0.9661358
```

```r
getMetrics(resp, pred, metrics = "MAE", na.rm = TRUE)
```

```
$MAE
[1] 0.8233219
```

```r
getMetrics(resp, pred, metrics = c("RMSE", "MAE"), na.rm = TRUE)
```

```
$RMSE
[1] 0.9661358

$MAE
[1] 0.8233219
```

```
# call function with incorrect data
getMetrics(data.frame(y = c(1)), data.frame(y_pred = c(2)))
```

```
Both responses and predictions should be numeric vectors
```

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

Question 1

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

api_key <- "86074e5006bb450db86061bc67de258c"

# GET call with q, from, and apiKey query params
resp <- GET("https://newsapi.org/v2/everything", query = list(
  q = "Physics",
  from = "2025-05-24",
  apiKey = api_key
))
```

Question 2

```
# parse raw content to JSON and then to list
parsed <- fromJSON(rawToChar(resp$content))

# get articles column and convert to tibble
as_tibble(parsed$articles)
```

```
# 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 Rhett~ Are ~ According ~ http~ https://m~ 2025-06-13~ "Touch~
 2 <NA>      Gizm~ Passa~ Shar~ It sort of~ http~ https://g~ 2025-06-04~ "Scien~
 3 the-verge The ~ Justi~ Goog~ Google is ~ http~ https://p~ 2025-06-12~ "Googl~
 4 <NA>      Vent~ Dean ~ Late~ Latent Tec~ http~ https://v~ 2025-06-05~ "Laten~
 5 <NA>      Hack~ John ~ A DI~ The Franck~ http~ https://h~ 2025-06-17~ "The F~
```

```
 6 <NA>      Gizm~ Isaac~ Simu~ The lightl~ http~ https://g~ 2025-06-04~ "When ~
 7 <NA>      Piku~ <NA>   Verl~ Physics si~ http~ https://p~ 2025-06-20~ "Physi~
 8 <NA>      Hack~ Lewin~ Port~ Portal 2 i~ http~ https://h~ 2025-06-01~ "Porta~
 9 <NA>      Hack~ Bryan~ Open~ The Portal~ http~ https://h~ 2025-06-14~ "The P~
10 wired     Wired Jake ~ '28 ~ The Britis~ http~ https://m~ 2025-06-20~ "In 20~
# i 90 more rows
```

Question 3

```r
# function that queries API for articles with subject, from, and api_key params
queryAPI <- function(subject, from, api_key) {
  resp <- GET("https://newsapi.org/v2/everything", query = list(
    q = subject,
    from = from,
    apiKey = api_key
  ))
  parsed <- fromJSON(rawToChar(resp$content))
  return(as_tibble(parsed$articles))
}

# call function
queryAPI("gamestop", "2025-05-30", api_key = api_key)
```

```
# A tibble: 98 x 8
   source$id $name author title description url   urlToImage publishedAt content
   <chr>     <chr> <chr>  <chr> <chr>       <chr> <chr>      <chr>       <chr>
 1 the-verge The ~ David~ A ni~ I'm standi~ http~ https://p~ 2025-06-05~ "Body ~
 2 the-verge The ~ Brand~ The ~ Amazon's m~ http~ https://p~ 2025-06-20~ "Amazo~
 3 <NA>      Gizm~ Kyle ~ Targ~ Check to m~ http~ https://g~ 2025-06-03~ "The S~
 4 <NA>      Slas~ msmash Game~ GameStop i~ http~ https://a~ 2025-06-13~ "Cohen~
 5 <NA>      Gizm~ James~ Did ~ Maybe orde~ http~ https://g~ 2025-06-05~ "When ~
 6 <NA>      Hipe~ Gabri~ Desa~ El gran dí~ http~ https://i~ 2025-06-05~ "El gr~
 7 <NA>      Kota~ Ethan~ Stat~ Imagine yo~ http~ https://i~ 2025-06-05~ "Imagi~
 8 <NA>      Gizm~ James~ Some~ There's on~ http~ https://g~ 2025-06-18~ "The S~
 9 <NA>      Gizm~ Kyle ~ Scre~ Even if Ga~ http~ https://g~ 2025-06-05~ "Gamer~
10 <NA>      Kota~ Zack ~ PSA:~ The Ninten~ http~ https://i~ 2025-06-02~ "The N~
# i 88 more rows
```