HW4: Hodge Podge Em

Mike Keating

```r
library(tidyverse)
```

Task 1: Conceptual Questions

Q1: What is the purpose of the lapply() function? What is the equivalent purrr function?

The purpose of lapply() is to apply a function to each element of a list.

The equivalent purr function is map().

Q2: 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()

```r
df1 <- data.frame(col1 = c(1 ,3, 4), col2 =c(2, 5, 5))
df2 <- df1 * 2

my_list <- list(df1, df2)

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

```
[[1]]
          col1      col2
col1 1.0000000 0.8164966
col2 0.8164966 1.0000000

[[2]]
```

```
          col1      col2
col1 1.0000000 0.8164966
col2 0.8164966 1.0000000
```

Q3: What are 2 advantages of using purr functions instead of BaseR apply family?

The purr functions allow for the use of helper/anonymous functions which will help us write more compact code.

The purr functions (map) have options like map_dbl() and map_int() can help us ensure our output is of a specific type.

Q4: What is a side-effect function?

A side-effect function is a function that performs an action without necessarily returning a value. In other words, something else happens instead of returning a value.

Q5: Why can you name a variable sd in a function and and not cause any issues with the sd function?

The variables created within the function are local/accessible only to the function. They are not initialized in the greater environment, so they do not conflict with the sd function.

Task 2: Writing R Functions

Q1: getRMSE

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(resp, pred, ...){
  args <- list(...)
  # Show a warning if NA is present
  if (anyNA(resp)) {
    if ("na.rm" %in% names(args)){
      if (isFALSE(args$na.rm)){
        warning("NA found in response vector. Consider setting na.rm = TRUE")
        }
      } else {
        warning("NA found in response vector. Consider setting na.rm = TRUE")
        }
  }

  # Get squared errors
  squared_errors <- (resp - pred)^2

  # Get mean squared errors, mse
  mse <- mean(squared_errors, ...)

  # Finally, get rmse
  rmse <- sqrt(mse)

  return(rmse)


}
```

Q2: Test getRMSE

Create 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 RMSE function

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

```
[1] 0.9581677
```

Repeat after replacing two of the response values with missing values.

```r
# To make things simple let's just replace the first two values in
# the response values with NA
resp_NA <- replace(resp, 1:2, NA_real_)
```

```r
# Calling without remove na arument
getRMSE(resp_NA, pred)
```

```
Warning in getRMSE(resp_NA, pred): NA found in response vector. Consider
setting na.rm = TRUE
```

```
[1] NA
```

```r
# Calling with our optional argument
getRMSE(resp_NA, pred, na.rm = TRUE)
```

```
[1] 0.9661699
```

Q3: getMAE()

```r
getMAE <- function(resp, pred, ...){
   args <- list(...)
  # Show a warning if NA is present
  if (anyNA(resp)) {
    if ("na.rm" %in% names(args)){
      if (isFALSE(args$na.rm)){
        warning("NA found in response vector. Consider setting na.rm = TRUE")
        }
      } else {
        warning("NA found in response vector. Consider setting na.rm = TRUE")
        }
  }
  # Get absolute error
   ae <- abs(resp - pred)
   mae <- mean(ae, ...)

   return(mae)

}
```

Q4: Test getMAE()

Create 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 MAE function:

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

```
[1] 0.8155776
```

Repeat after replacing two of the response values with missing values.

```
# To make things simple let's just replace the first two values in
# the response values with NA
resp_NA <- replace(resp, 1:2, NA_real_)


# Calling without remove na arument
getMAE(resp_NA, pred)
```

```
Warning in getMAE(resp_NA, pred): NA found in response vector. Consider setting
na.rm = TRUE
```

```
[1] NA
```

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

```
[1] 0.8241201
```

Q5: Wrapper for Both Metrics

```r
getMetrics <- function(resp, pred, metrics = c("rmse", "mae"), ...){
  if (!is.numeric(resp) | !is.numeric(pred)){
    stop("ERROR: Responses or predictions not a numeric vector")
  }

  results = list()



  if ("rmse" %in% metrics){
    results$rmse <- getRMSE(resp, pred, ...)

  }
  if ("mae" %in% metrics){
    results$mae <- getMAE(resp, pred, ...)
  }

  if (length(results) == 0){
    stop("ERROR: Unknown metrics supplied")
  }

  return(results)

}
```

Q6: Test New Metric Function

Call function for RMSE:

```r
getMetrics(resp, pred, c("rmse"))
```

```
$rmse
[1] 0.9581677
```

Call function for MAE:

```r
getMetrics(resp, pred, c("mae"))
```

```
$mae
[1] 0.8155776
```

Calling function for both metrics:

```r
getMetrics(resp, pred, c("rmse","mae"))
```

```
$rmse
[1] 0.9581677

$mae
[1] 0.8155776
```

Repeat with NA values:

```r
# Recall we already replaced with NA in our resp_NA object
getMetrics(resp_NA, pred, c("rmse", "mae"))
```

```
Warning in getRMSE(resp, pred, ...): NA found in response vector. Consider
setting na.rm = TRUE
```

```
Warning in getMAE(resp, pred, ...): NA found in response vector. Consider
setting na.rm = TRUE
```

```
$rmse
[1] NA

$mae
[1] NA
```

```r
# Recall we already replaced with NA in our resp_NA object
getMetrics(resp_NA, pred, c("rmse", "mae"), na.rm = TRUE)
```

```
$rmse
[1] 0.9661699

$mae
[1] 0.8241201
```

Task 3

Q1: 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). Note: We can only look 30 days into the past with a free account.

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

```
api_key ="0ccf25ee550542dcb67870ea981d4e3f"
subject <- "magic the gathering"
# We need to replace spaces with &, as well as add and extra one at the end
subject <- paste0(str_replace_all(subject, " ", "&"),"&")

url <- paste0("https://newsapi.org/v2/everything?q=", subject,
              "apiKey=", api_key)
response <- GET(url = url)
```

Q2: Parse What is Returned

```
# We were given a hint that the data is in the contents field, so we will parse that
parsed_response <- fromJSON((rawToChar(response$content)))
str(parsed_response)
```

```
List of 3
 $ status      : chr "ok"
 $ totalResults: int 11695
 $ articles    :'data.frame':   100 obs. of  8 variables:
  ..$ source     :'data.frame': 100 obs. of  2 variables:
  .. ..$ id  : chr [1:100] "the-verge" NA NA NA ...
  .. ..$ name: chr [1:100] "The Verge" "Gizmodo.com" "Gizmodo.com" "Gizmodo.com" ...
  ..$ author     : chr [1:100] "Ash Parrish" "Germain Lussier" "Justin Carter" "Gizmodo Deals
  ..$ title      : chr [1:100] "Final Fantasy fans, now is the time to get into Magic: The Ga
  ..$ description: chr [1:100] "The Final Fantasy Magic: The Gathering set is here, and there
  ..$ url        : chr [1:100] "https://www.theverge.com/games/690509/how-to-play-final-fanta
  ..$ urlToImage : chr [1:100] "https://platform.theverge.com/wp-content/uploads/sites/2/2025
  ..$ publishedAt: chr [1:100] "2025-06-20T19:35:42Z" "2025-06-03T21:30:00Z" "2025-06-21T18:5
  ..$ content    : chr [1:100] "Heres a guide to get started playing one of the best Magic se
```

Looks like our article info is stored in the articles df:

```r
articles <-as_tibble(parsed_response$articles)
str(articles)
```

```
tibble [100 x 8] (S3: tbl_df/tbl/data.frame)
 $ source     :'data.frame':    100 obs. of  2 variables:
  ..$ id  : chr [1:100] "the-verge" NA NA NA ...
  ..$ name: chr [1:100] "The Verge" "Gizmodo.com" "Gizmodo.com" "Gizmodo.com" ...
 $ author     : chr [1:100] "Ash Parrish" "Germain Lussier" "Justin Carter" "Gizmodo Deals"
 $ title      : chr [1:100] "Final Fantasy fans, now is the time to get into Magic: The Gath
 $ description: chr [1:100] "The Final Fantasy Magic: The Gathering set is here, and there's
 $ url        : chr [1:100] "https://www.theverge.com/games/690509/how-to-play-final-fantasy
 $ urlToImage : chr [1:100] "https://platform.theverge.com/wp-content/uploads/sites/2/2025/0
 $ publishedAt: chr [1:100] "2025-06-20T19:35:42Z" "2025-06-03T21:30:00Z" "2025-06-21T18:50:
 $ content    : chr [1:100] "Heres a guide to get started playing one of the best Magic sets
```

Q3: Quick API Function

Write a function to allow the user to easily query API.

```r
searchNews <- function(subject, from_date = Sys.Date(), api_key){
  # API only goes back 30 days, so let's check and adjust
  todays_date <- Sys.Date()
  thirty_days_ago <- todays_date - 30
  if (from_date < thirty_days_ago){
    print("from_date is too far in the past, searching instead from 30 days ago")
    from_date <- thirty_days_ago
  }

  # Clean up subject and from_date
  subject <- paste0(str_replace_all(subject, " ", "&"),"&")
  from_date <- paste0(from_date, "&")
  # Build url
  url <- paste0("https://newsapi.org/v2/everything?q=", subject,
                "from=", from_date,
                "apiKey=", api_key)

  # Query api for response
  response <- GET(url = url)
  parsed_response <- fromJSON((rawToChar(response$content)))
  articles <-as_tibble(parsed_response$articles)


  return(articles)
}
```

```r
searchNews("gamestop", "2025-05-19", api_key)
```

```
[1] "from_date is too far in the past, searching instead from 30 days ago"
```

```
# A tibble: 99 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 business~ Busi~ fdemo~ Game~ GameStop a~ http~ https://i~ 2025-05-28~ "GameS~
 4 <NA>      Gizm~ Kyle ~ Targ~ Check to m~ http~ https://g~ 2025-06-03~ "The S~
 5 <NA>      Slas~ msmash Game~ GameStop i~ http~ https://a~ 2025-06-13~ "Cohen~
 6 <NA>      Yaho~ Brad ~ Bitc~ The 2025 a~ http~ https://s~ 2025-05-28~ "Bitco~
 7 <NA>      Gizm~ James~ Did ~ Maybe orde~ http~ https://g~ 2025-06-05~ "When ~
 8 <NA>      Hipe~ Gabri~ Desa~ El gran dí~ http~ https://i~ 2025-06-05~ "El gr~
 9 <NA>      Kota~ Ethan~ Stat~ Imagine yo~ http~ https://i~ 2025-06-05~ "Imagi~
10 <NA>      Gizm~ James~ Some~ There's on~ http~ https://g~ 2025-06-18~ "The S~
# i 89 more rows
```