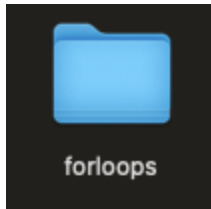


Problem 1:

create a simple R package called forloops that reproduces some Base R functions using For-loops.

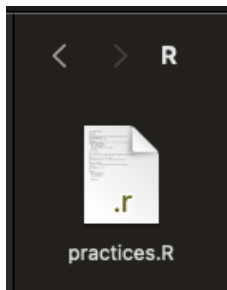
## Part a:

- Create a folder called “forloop”, set the working directory to this folder



## Part b:

- In the R sub-directory create a R script called practice



## Part c:

- Build a function inside of R script practice called col\_means() that will take as input a data frame and return a vector of column means. We will not use the colMeans() function from {base} package

```
col_means <- function(df) {  
  means <- numeric(ncol(df))  
  for (i in seq_along(df)) {  
    means[i] <- mean(df[[i]], na.rm = TRUE)  
  }  
  return(means)  
}
```

## Part d:

- Build a function inside of R script practice called `count_na()` that will use a for-loop to count how many NA's there are in a vector.

```
count_na <- function(vec) {  
  count <- 0  
  for (i in seq_along(vec)) {  
    if (is.na(vec[i])) {  
      count <- count + 1  
    }  
  }  
  return(count)  
}
```

## Part e:

- Create documentation for each functions that you have in your R script. Then load the package and type `?col_means` and then `?count_na`.

count\_na.Rd - Find in Topic

**Description**  
This function counts the number of NA values in a given vector using a for-loop.

**Usage**  

```
count_na(vec)
```

**Arguments**  

`vec` A numeric or character vector.

**Details**  
After inputting a vector, it will return the count of NA values.

**Value**  
An integer representing the count of NA values in the vector.

**Author(s)**  
Andrew Hwang

**Examples**  
Run examples  

```
vec <- c(1, 2, NA, 4, NA, 6)  
count_na(vec)
```

[Package *packages1* version 0.0.0.9000 [Index](#)]

The screenshot shows the RStudio Help viewer interface. At the top, there are tabs for Files, Plots, Packages, Help, Viewer, and Presentation. Below the tabs is a search bar and a navigation bar. The main content area displays the documentation for the `col_means` function. The documentation includes sections for Usage, Arguments, Details, Value, Author(s), and Examples. The Usage section shows the function signature `col_means(df)`. The Arguments section lists `df`. The Details section explains that the function outputs the mean of each column after inputting a dataframe. The Value section states that the function returns the mean of a column. The Author(s) section lists Andrew Hwang. The Examples section provides a code snippet for creating a dataframe and using the `col_means` function. At the bottom, there is a link to the package index.

**Usage**

```
col_means(df)
```

**Arguments**

`df`

**Details**

After inputting a dataframe it will output the mean of each column

**Value**

The Mean of a column

**Author(s)**

Andrew Hwang

**Examples**

[Run examples](#)

```
## Data_Frame <- data.frame (column1 = c(100, 150, 120), c  
## col_means(Data_Frame)
```

[Package *packages1* version 0.0.0.9000 [Index](#)]

```
library(packages1)  
?col_means  
  
?count_na
```

## Part f:

- In your R studio move to Terminal and type `ls`, it will show the list of all files and folders that you have in your working directory forloops.

The screenshot shows the RStudio Terminal window. The prompt is `andrews-mbp-2:forloops andrewhwang$`. The user has entered `ls`, and the output is displayed as a table with columns: DESCRIPTION, LICENSE, LICENSE.md, NAMESPACE, R, and man. The user has entered `ls` again, and the prompt is `andrews-mbp-2:forloops andrewhwang$`.

```
andrews-mbp-2:forloops andrewhwang$ ls  
DESCRIPTION    LICENSE        LICENSE.md     NAMESPACE     R              man  
andrews-mbp-2:forloops andrewhwang$ ls
```