Coding Challenge #5 - For Loop

SA Donohoo

2025-03-27

Table of Contents

[Github Link 1](#_Toc193933556)

[Question 1 - Reproducibility 1](#_Toc193933557)

[Question 2 - Explain For Loops 1](#_Toc193933558)

[Question 3 - Adding Data 1](#_Toc193933559)

[Question 4 - Generate a For Loop 2](#_Toc193933560)

[Question 5 - A Single Distance 2](#_Toc193933561)

[Question 6 - For Loop 3](#_Toc193933562)

# Github Link

Sam Donohoo - PLPA 6820 - <https://github.com/sad0046/PLPA6820_SP25>

### Question 1 - Reproducibility

Regarding reproducibility, what is the main point of writing your own functions and iterations? **Answer**:

### Question 2 - Explain For Loops

In your own words, describe how to write a function and a for loop in R and how they work. Give me specifics like syntax, where to write code, and how the results are returned. **Answer**

### Question 3 - Adding Data

Read in the Cities.csv file from Canvas using a relative file path.

# Load in Packages  
library(ggplot2)  
library(drc)   
library(tidyverse)  
library(dplyr)  
  
# Read in Data  
  
All.Cities <- read.csv("../Raw\_Data\_for\_Class\_Exercises/Cities.csv", header = TRUE, na.strings = "NA")

### Question 4 - Generate a For Loop

Write a function to calculate the distance between two pairs of coordinates based on the Haversine formula (see below). The input into the function should be lat1, lon1, lat2, and lon2. The function should return the object distance\_km. All the code below needs to go into the function.

#### Haversine formula

# Convert to radians  
rad.lat1 <- lat1 \* pi/180  
rad.lon1 <- lon1 \* pi/180  
rad.lat2 <- lat2 \* pi/180  
rad.lon2 <- lon2 \* pi/180  
  
# Haversine formula  
delta\_lat <- rad.lat2 - rad.lat1  
delta\_lon <- rad.lon2 - rad.lon1  
a <- sin(delta\_lat / 2)^2 + cos(rad.lat1) \* cos(rad.lat2) \* sin(delta\_lon / 2)^2  
c <- 2 \* asin(sqrt(a))   
  
# Earth's radius in kilometers  
earth\_radius <- 6378137  
  
# Calculate the distance  
distance\_km <- (earth\_radius \* c)/1000

#### Answer - Function

distance <- function(lat1,lon1,lat2,lon2) {  
 rad.lat1 <- lat1 \* pi/180  
 rad.lon1 <- lon1 \* pi/180  
 rad.lat2 <- lat2 \* pi/180  
 rad.lon2 <- lon2 \* pi/180  
 delta\_lat <- rad.lat2 - rad.lat1  
 delta\_lon <- rad.lon2 - rad.lon1  
 a <- sin(delta\_lat / 2)^2 + cos(rad.lat1) \* cos(rad.lat2) \* sin(delta\_lon / 2)^2  
 c <- 2 \* asin(sqrt(a))  
 earth\_radius <- 6378137  
 distance\_km <- (earth\_radius \* c)/1000  
  
}

### Question 5 - A Single Distance

Using your function, compute the distance between Auburn, AL and New York City

* Subset/filter the Cities.csv data to include only the latitude and longitude values you need and input as input to your function.
* The output of your function should be 1367.854 km

# Subset and Preset New York  
NewYork.lat <- All.Cities$lat[All.Cities$city == "New York"]  
NewYork.lon <- All.Cities$long[All.Cities$city == "New York"]  
  
# Subset and Preset Auburn Coordinates  
Auburn.lat <- All.Cities$lat[All.Cities$city == "Auburn"]  
Auburn.lon <- All.Cities$long[All.Cities$city == "Auburn"]  
  
# Generate Empty Dataframe  
Cities.Distance.NewYork <- NULL # create a null object  
  
# Run Function  
Distance.NewYork <- distance(NewYork.lat,NewYork.lon,Auburn.lat,Auburn.lon)  
  
# Combine Empty and Output Dataframe  
Cities.Distance.NewYork <- rbind.data.frame(Cities.Distance.NewYork, Distance.NewYork)  
  
# Edit Column Name  
colnames(Cities.Distance.NewYork) <- "km\_distance"  
  
# Print Values  
print(Cities.Distance.NewYork)

## km\_distance  
## 1 1367.854

### Question 6 - For Loop

Now, use your function within a for loop to calculate the distance between all other Cities.nm in the data.

# Select All Cities That Are Not Auburn  
Citites <- All.Cities[All.Cities$city != "Auburn",]  
  
# Create a vector of City Names  
Citites.names <- unique(Citites$city)  
  
# Generate Empty Dataframe  
Cities.Distance.All <- NULL # create a null object  
  
# Create for Loop that runs through all Cities and Calculates Distance to Auburn  
for (i in seq\_along(Citites.names)) {  
 Cities.Distance.Single <- distance(Citites$lat[i],  
 Citites$long[i],  
 Auburn.lat,  
 Auburn.lon)  
 Cities.Distance.All <- rbind.data.frame(Cities.Distance.All, Cities.Distance.Single)  
}  
  
# Create a Dataframe with Similar Observation Length as All.Citites  
Citites.Two <- rep("Auburn",39)  
  
# Combine Three Dataframes  
Cities.Distance.All <- cbind(Citites.names,City.Two.Name,Cities.Distance.All)  
  
# Edit Column Names  
colnames(Cities.Distance.All) <- c("City1", "City2", "km\_distance")  
  
# Print Values  
print(Cities.Distance.All)