CodingChallenege6

Vaibhav B. Shelar

2025-03-27

### Q1

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

# Ans: The main point of writing your own functions and iterations is to make the code more readable, reusable, and reproducible.

### Q2

# 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.

# Ans: To write a function in R, we use the function keyword followed by the function name and its arguments. The body of the function is enclosed in curly braces {} and contains the code that performs the desired task.

# Examples

# 1)

C\_to\_F <- function(c\_temp) {  
 fahrenheit <- (c\_temp \* 9/5) + 32 # Conversion formula  
 return(fahrenheit) # Return the result  
}  
  
# Example usage:  
C\_to\_F(25)

## [1] 77

for(i in -25:50){  
 result <- C\_to\_F(i)  
 print(result)  
}

## [1] -13  
## [1] -11.2  
## [1] -9.4  
## [1] -7.6  
## [1] -5.8  
## [1] -4  
## [1] -2.2  
## [1] -0.4  
## [1] 1.4  
## [1] 3.2  
## [1] 5  
## [1] 6.8  
## [1] 8.6  
## [1] 10.4  
## [1] 12.2  
## [1] 14  
## [1] 15.8  
## [1] 17.6  
## [1] 19.4  
## [1] 21.2  
## [1] 23  
## [1] 24.8  
## [1] 26.6  
## [1] 28.4  
## [1] 30.2  
## [1] 32  
## [1] 33.8  
## [1] 35.6  
## [1] 37.4  
## [1] 39.2  
## [1] 41  
## [1] 42.8  
## [1] 44.6  
## [1] 46.4  
## [1] 48.2  
## [1] 50  
## [1] 51.8  
## [1] 53.6  
## [1] 55.4  
## [1] 57.2  
## [1] 59  
## [1] 60.8  
## [1] 62.6  
## [1] 64.4  
## [1] 66.2  
## [1] 68  
## [1] 69.8  
## [1] 71.6  
## [1] 73.4  
## [1] 75.2  
## [1] 77  
## [1] 78.8  
## [1] 80.6  
## [1] 82.4  
## [1] 84.2  
## [1] 86  
## [1] 87.8  
## [1] 89.6  
## [1] 91.4  
## [1] 93.2  
## [1] 95  
## [1] 96.8  
## [1] 98.6  
## [1] 100.4  
## [1] 102.2  
## [1] 104  
## [1] 105.8  
## [1] 107.6  
## [1] 109.4  
## [1] 111.2  
## [1] 113  
## [1] 114.8  
## [1] 116.6  
## [1] 118.4  
## [1] 120.2  
## [1] 122

### Q3

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

# Read in the dataset  
cities <- read.csv("Cities.csv")

### Q4

# Function to calculate the distance between two pairs of coordinates  
haversine\_distance <- function(lat1, lon1, lat2, lon2) {  
 rad.lat1 <- lat1 \* pi/180 # convert to radians  
 rad.lon1 <- lon1 \* pi/180  
 rad.lat2 <- lat2 \* pi/180  
 rad.lon2 <- lon2 \* pi/180  
 delta\_lat <- rad.lat2 - rad.lat1 # Haversine formula  
 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 # Earth's radius in kilometers  
 distance\_km <- (earth\_radius \* c)/1000 # Calculate the distance  
 return(distance\_km)  
}

### Q5

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

# a. Subset/filter the Cities.csv data to include only the latitude and longitude values you need and input as input to your function.

# a

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Subset the data for Auburn, AL and New York City  
nyc<- cities %>% filter(city == "New York")   
auburn<- cities %>% filter(city == "Auburn")  
  
# Extracting just latitude and longitude values  
lat1<- nyc$lat  
lon1<- nyc$long  
  
lat2<- auburn$lat  
lon2<- auburn$long  
  
# b   
  
haversine\_distance(lat1, lon1, lat2, lon2)

## [1] 1367.854

### Q6

# Now, use your function within a for loop to calculate the distance between all other cities in the data

# loops to calculate distance from auburn to each city  
for(i in 1:nrow(cities)){  
 if(cities$city[[1]] != "Auburn"){  
 lat1<- cities$lat[i]  
 lon1<- cities$long[i]  
   
 dist<- haversine\_distance(lat1, lon1, lat2, lon2)  
 print(dist)  
 }  
}

## [1] 1367.854  
## [1] 3051.838  
## [1] 1045.521  
## [1] 916.4138  
## [1] 993.0298  
## [1] 1056.022  
## [1] 1239.973  
## [1] 162.5121  
## [1] 1036.99  
## [1] 1665.699  
## [1] 2476.255  
## [1] 1108.229  
## [1] 3507.959  
## [1] 3388.366  
## [1] 2951.382  
## [1] 1530.2  
## [1] 591.1181  
## [1] 1363.207  
## [1] 1909.79  
## [1] 1380.138  
## [1] 2961.12  
## [1] 2752.814  
## [1] 1092.259  
## [1] 796.7541  
## [1] 3479.538  
## [1] 1290.549  
## [1] 3301.992  
## [1] 1191.666  
## [1] 608.2035  
## [1] 2504.631  
## [1] 3337.278  
## [1] 800.1452  
## [1] 1001.088  
## [1] 732.5906  
## [1] 1371.163  
## [1] 1091.897  
## [1] 1043.273  
## [1] 851.3423  
## [1] 1382.372  
## [1] 0

## Bonus point code

# creating a empty data frame to store results  
citydistance\_df<- data.frame(city1 = character(),  
 city2 = character(),  
 distance\_km = numeric())  
# Loop through each city  
for(i in 1:nrow(cities)){  
 if(cities$city[i] != "Auburn"){  
 lat1<- cities$lat[i]  
 lon1<- cities$long[i]  
# calculating the distance  
 dist<- haversine\_distance(lat1, lon1, lat2, lon2)  
   
# Adding results as new row  
 new\_row<- data.frame(city1 = cities$city[i],  
 city2 = "Auburn",  
 distance\_km = dist)  
   
 citydistance\_df<- rbind(citydistance\_df, new\_row)  
 }  
   
}  
  
# first six rows of the data frame  
head(citydistance\_df)

## city1 city2 distance\_km  
## 1 New York Auburn 1367.8540  
## 2 Los Angeles Auburn 3051.8382  
## 3 Chicago Auburn 1045.5213  
## 4 Miami Auburn 916.4138  
## 5 Houston Auburn 993.0298  
## 6 Dallas Auburn 1056.0217

### Q7 Adding a link to my github

(Github\_Link)[<https://github.com/vbs0303/Class-Reports.git>]