CodingChallenge6

Kylie Blake

2025-03-27

Contents

Question 1

1. 2 pts. Regarding reproducibility, what is the main point of writing your own functions and iterations?

- The main point of writing your own functions and iterations in ragards to reproducibility is that it reduces error from copy and pasting the same function or entering the same values multiple times. A custom function also improves consistency in your script and allows easier modifications to code if edits are needed.

Question 2

2. 2 pts. 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.

a. Writing a function:

- To write a function, you first need to give the function a name, i.e. NAME <- function(). This will store your function in the Environment.
- Second, you need to give the function an input name(s) or argument(s), these arguments will be executed within the body of a function. Ex. function(velocity, time)
- Lastly, the "tasks" of the function sit inside curly brackets, and are known as the body of the function. Ex. function(velocity, time){ distance <- (velocity * time) return(distance) }
- To see the calculated distance in the console, you would tell the function to return() that variable.

b. Writing a for loop:

In simplest terms, a for loop executes a task for every itemp in a sequence, and continues running that task until the last value in the sequence is reached.

- The first component of the loop is setting defining the sequence. The sequence may be numeric or characters. The syntax is always the same for(variable in vector). The variable can be named whatever name you prefer. For example "for(i in 1:10)" is saying for variable i in the sequence 1 through 10, or set variable i equal to 1:10.
- Secondly, you need to give the for loop a task to do, which will sit in the body of the for loop. Again the body is represented with curly brackets. ex. for (i in 1:10) {print(i+2)}. This is saying take values 1 through 10 and add two to each value in the sequence.
- For loops can become more complex by saving outputted values into a dataframe. In this situation, you would define a dataframe prior to executing the for loop with no values. ex. Table <- NULL. You would then define a dataframe for values within the body of the for loop.

Question 3

3. 2 pts. Read in the Cities.csv file from Canvas using a relative file path.

```
Cities <- read.csv("CodingChallenge6/Cities.csv")
```

Question 4

4. 6 pts. 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.

```
Function <- function(lat1, lon1, lat2,lon2){
# 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

return(distance_km) }
```

Question 5

5. 5 pts. 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.

b. The output of your function should be 1367.854 km

```
#Part a
#Subset out nyc data and Auburn data
nyc <- subset(Cities, city == "New York")
auburn <- subset(Cities, city == "Auburn")

lat1 <- nyc$lat
lon1 <- nyc$long
lat2 <- auburn$lat
lon2 <- auburn$long

#Part b - Calculate distance from NYC to Auburn
distance_km <- Function(lat1, lon1, lat2, lon2)
print(distance_km)
```

```
## [1] 1367.854
```

Question 6

6. 6 pts. Now, use your function within a for loop to calculate the distance between all other cities in the data. The output of the first 9 iterations is shown below.

```
#list city names
city_name <- Cities$city
city_name
```

```
##  [1] "New York"      "Los Angeles"   "Chicago"       "Miami"
##  [5] "Houston"       "Dallas"        "Philadelphia"  "Atlanta"
##  [9] "Washington"    "Boston"        "Phoenix"       "Detroit"
## [13] "Seattle"       "San Francisco" "San Diego"     "Minneapolis"
```

```
## [17] "Tampa"         "Brooklyn"      "Denver"        "Queens"
## [21] "Riverside"     "Las Vegas"     "Baltimore"     "St. Louis"
## [25] "Portland"      "San Antonio"   "Sacramento"    "Austin"
## [29] "Orlando"       "San Juan"      "San Jose"      "Indianapolis"
## [33] "Pittsburgh"    "Cincinnati"    "Manhattan"     "Kansas City"
## [37] "Cleveland"     "Columbus"      "Bronx"         "Auburn"
```

```
for(i in seq_along(city_name)){
all_cities <- subset(Cities, city == city_name[i]) #Go through all city names for i
  lat1 <- all_cities$lat
  lon1 <- all_cities$long
  lat2 <- auburn$lat
  lon2 <- auburn$long

distance_km <- Function(lat1, lon1, lat2, lon2)
print(distance_km)}
```

```
## [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

Bonus point if you can have the output of each iteration append a new row to a dataframe, generating a new column of data. In other words, the loop should create a dataframe with three columns called city1, city2, and distance_km, as shown below. The first six rows of the dataframe are shown below.

- I could not figure this out! If we could break it down in class, I would appreciate that!

Question 7

7. 2 pts. Commit and push a gfm .md file to GitHub inside a directory called Coding Challenge 6. Provide me a link to your github written as a clickable link in your .pdf or .docx

Click here to my GitHub