Coding_challenge_6

Bibek

2025-03-26

#Question 1 1) Regarding reproducibility, what is the main point of writing your own functions and iterations? Ans »

Writing own fucntions and iteration makes the code more organized and easier to run it again. With the function, we can code a set of instruction that can be used over again and again without rewriting them. Additionally, when we combine this with iteration which are loop structures it ensures that each steps in the analysis is clear and consistent for all the inputs. So, both of this ensure that our code behaves in same way, making it simpler to understand and free form minor error that might arise from copying and pasting. All of these ensures proper reproducibility.

#Question 2 2) 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 »

a) For function

-First we give the name to function.Then we use "function ()" where we list variables that we want to pass i.e. inside the bracket.This is followed by the curly brackets where actual code is written for calculations or specific operation. Finally, we use "return()" to obtained specific output.

Below is the example for this. When we call my_function(1), it returns 3 as output.

```
##Example
my_function <- function(x) {
  result <- x + 2   ## This is where we write the code
  return(result)
}

my_function(1)
```

```
## [1] 3
```

b) for loop

-The for loop starts with "for (i in sequence range) {}, where"i" is the iterator that uses the values from the sequence range one by one. The sequence is the list or range of values that the loop goes through one by one. It can be numbers, characters, or logical operators and guide the loop for how many times and in what order to run.The main code for execution is written inside the curly brackets. This code is executed by each value in the sequence which is specified by the "i". Then, print() function helps to obtained the result of the code in the console windows. This loop continues to run until the last values in the provided sequence. Instead of printing directly in console, we can use an empty objject before the loop and the store each value for later use.

Below is the simple example for the for loop.(without null object) This will give the multiple of 2 till 20.

```r
for (i in 1:10){
  print(i*2)
}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
## [1] 14
## [1] 16
## [1] 18
## [1] 20
```

#Question 3
Read in the Cities.csv file from Canvas using a relative file path.

```r
Thecities <- read.csv("Sample_data/Cities.csv")
```

#Question 4

Making the function

```r
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
  # 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 Computing the distance between Auburn, AL and New York City

```r
# Get coordinates for Auburn
auburn <- subset(Thecities, city == "Auburn")
lat1 <- auburn$lat
lon1 <- auburn$long

# Get coordinates for New York City
```

```r
nyc <- subset(Thecities, city == "New York")
lat2 <- nyc$lat
lon2 <- nyc$long

# Calculate the distance
distance_to_nyc <- distance(lat1, lon1, lat2, lon2)
distance_to_nyc
```

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

#**Question 6** Calculate the distance between all other cities in the data.

```r
#without creating null object:
for (i in seq_along(Thecities$city)){
    if (Thecities$city[i] != "Auburn") {
    lat2 <- Thecities$lat[i]
    lon2 <- Thecities$long[i]
    the_distance <- distance(lat1, lon1, lat2, lon2)
    print(the_distance)
    }
}
```

```
## [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
```

Bonus Question:

```r
#With null object to stored the data:

result <- NULL
#for loop
for (i in seq_along(Thecities$city)){          # Loop each row in the df
    city_name <- Thecities$city[i]             # Extract city name
    if (city_name != "Auburn") {               # Skip Auburn itself
    lat2 <- Thecities$lat[i]                   # Extract the latitude and longitude
    lon2 <- Thecities$long[i]
    the_distance <- distance(lat1, lon1, lat2, lon2)  # Using above distance function to calculate dist
    print(the_distance)                      # Print the distance in the console

    # making new row to results as mentioned in the question
    new_row <- data.frame(City1 = Thecities$city[i] , City2 = "Auburn", Distance_km = the_distance)
    result <- rbind(result, new_row)

  }
}
```

```
## [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
```

```
result
```

```
##            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
## 7   Philadelphia Auburn   1239.9732
## 8        Atlanta Auburn    162.5121
## 9     Washington Auburn   1036.9900
## 10        Boston Auburn   1665.6985
## 11       Phoenix Auburn   2476.2552
## 12       Detroit Auburn   1108.2288
## 13       Seattle Auburn   3507.9589
## 14 San Francisco Auburn   3388.3656
## 15     San Diego Auburn   2951.3816
## 16   Minneapolis Auburn   1530.2000
## 17         Tampa Auburn    591.1181
## 18      Brooklyn Auburn   1363.2072
## 19        Denver Auburn   1909.7897
## 20        Queens Auburn   1380.1382
## 21     Riverside Auburn   2961.1199
## 22     Las Vegas Auburn   2752.8142
## 23     Baltimore Auburn   1092.2595
## 24     St. Louis Auburn    796.7541
## 25      Portland Auburn   3479.5376
## 26   San Antonio Auburn   1290.5492
## 27    Sacramento Auburn   3301.9923
## 28        Austin Auburn   1191.6657
## 29       Orlando Auburn    608.2035
## 30      San Juan Auburn   2504.6312
## 31      San Jose Auburn   3337.2781
## 32  Indianapolis Auburn    800.1452
## 33    Pittsburgh Auburn   1001.0879
## 34    Cincinnati Auburn    732.5906
```

```
## 35     Manhattan Auburn   1371.1633
## 36   Kansas City Auburn   1091.8970
## 37     Cleveland Auburn   1043.2727
## 38      Columbus Auburn    851.3423
## 39         Bronx Auburn   1382.3721
```

Without using if function; this will not check if the same cities are being compared with each other. So we will get Auburn to Auburn at zero distance apart.

```
result_distance2 <- NULL

for (i in seq_along(Thecities$city)) {
      lat1 <- Thecities$lat[Thecities$city=="Auburn"]
      lon1 <- Thecities$long[Thecities$city=="Auburn"]
      lat2 <- Thecities$lat[i]
      lon2 <- Thecities$long[i]

      the_distance <- distance(lat1, lon1, lat2, lon2)  # Using above distance function to calculate di
    print(the_distance)                      # Print the distance in the console

    # making new row to results as mentioned in the question
    new_row <- data.frame(City1 = Thecities$city[i], City2 = "Auburn", Distance_km = the_distance)
    result_distance2 <- rbind(result_distance2, new_row)

}
```

```
## [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
```

```
result_distance2
```

```
##            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
## 7   Philadelphia Auburn   1239.9732
## 8        Atlanta Auburn    162.5121
## 9     Washington Auburn   1036.9900
## 10        Boston Auburn   1665.6985
## 11       Phoenix Auburn   2476.2552
## 12       Detroit Auburn   1108.2288
## 13       Seattle Auburn   3507.9589
## 14 San Francisco Auburn   3388.3656
## 15     San Diego Auburn   2951.3816
## 16   Minneapolis Auburn   1530.2000
## 17         Tampa Auburn    591.1181
## 18      Brooklyn Auburn   1363.2072
## 19        Denver Auburn   1909.7897
## 20        Queens Auburn   1380.1382
## 21     Riverside Auburn   2961.1199
## 22     Las Vegas Auburn   2752.8142
## 23     Baltimore Auburn   1092.2595
## 24     St. Louis Auburn    796.7541
## 25      Portland Auburn   3479.5376
## 26   San Antonio Auburn   1290.5492
## 27    Sacramento Auburn   3301.9923
## 28        Austin Auburn   1191.6657
## 29       Orlando Auburn    608.2035
## 30      San Juan Auburn   2504.6312
## 31      San Jose Auburn   3337.2781
## 32  Indianapolis Auburn    800.1452
## 33    Pittsburgh Auburn   1001.0879
## 34    Cincinnati Auburn    732.5906
## 35     Manhattan Auburn   1371.1633
## 36   Kansas City Auburn   1091.8970
## 37     Cleveland Auburn   1043.2727
## 38      Columbus Auburn    851.3423
```

```
## 39          Bronx Auburn   1382.3721
## 40         Auburn Auburn      0.0000
```

#Question 7 Commit and push a gfm .md file to GitHub inside a directory called Coding Challenge 6.

Please click this: **Link to my GitHub**