CodingChallenge6

AJ Neff

2025-03-27

# Question 1

The main point is simplify your code so that it’s easy to use repeatably without error.

# Question 2

A function must first be assigned a name and given the beginning of the input code such as the syntax below: function\_name <- function(variable) { body return(output) }. You then add the code in the body and a return for what you want the output to do.

# Question 3

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggplot2)  
Cities <- read.csv("Cities.csv")

# Question 4

# convert to radians  
dist\_bet <- 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

Cities\_subset <- subset(Cities, city %in% c("New York", "Auburn"), select = c(long, lat))  
print(Cities\_subset)

## long lat  
## 1 -73.9249 40.6943  
## 40 -85.4903 32.6087

nyc\_auburn <- dist\_bet(40.6943, -73.9249, 32.6087, -85.4903)  
print(nyc\_auburn)

## [1] 1367.854

# Question 6

dist\_all <- NULL  
for (i in 1:nrow(Cities)){  
 distance\_i <- dist\_bet(Cities$lat[i], Cities$long[i], 32.6087, -85.4903)  
 dist\_all <- rbind.data.frame(dist\_all, distance\_i)  
}  
  
print(dist\_all)

## X1367.85395084397  
## 1 1367.8540  
## 2 3051.8382  
## 3 1045.5213  
## 4 916.4138  
## 5 993.0298  
## 6 1056.0217  
## 7 1239.9732  
## 8 162.5121  
## 9 1036.9900  
## 10 1665.6985  
## 11 2476.2552  
## 12 1108.2288  
## 13 3507.9589  
## 14 3388.3656  
## 15 2951.3816  
## 16 1530.2000  
## 17 591.1181  
## 18 1363.2072  
## 19 1909.7897  
## 20 1380.1382  
## 21 2961.1199  
## 22 2752.8142  
## 23 1092.2595  
## 24 796.7541  
## 25 3479.5376  
## 26 1290.5492  
## 27 3301.9923  
## 28 1191.6657  
## 29 608.2035  
## 30 2504.6312  
## 31 3337.2781  
## 32 800.1452  
## 33 1001.0879  
## 34 732.5906  
## 35 1371.1633  
## 36 1091.8970  
## 37 1043.2727  
## 38 851.3423  
## 39 1382.3721  
## 40 0.0000