RWorksheet_Delgado#4a.Rmd

Clyde Marcelo Delgado

2024-10-14

Question 1: Shoe size and height data frame

Create data frame with shoe size and height for males and females

library(openxlsx)

shoe_size_height_data <- read.xlsx(file.path("C:", "Rworksheets", "worksheet#4", "shoe_size_height_data.xlsx"))

df <- shoe_size_height_data

a. Describe the data

summary(df)

b. Subset by males and females

df_male <- subset(df, gender == "M") df_female <- subset(df, gender == "F")

Print male and female subsets

df_male df_female

c. Find the mean of shoe size and height

mean_shoe_size <- mean(df$shoe_size)$mean_height < -mean(df$height)

Print the means

mean_shoe_size mean_height

d. Check relationship between shoe size and height (Correlation)

correlation <- cor(df$shoe_size, df$height)

Print the correlation value

correlation

A positive correlation suggests a relationship between shoe size and height.

Question 2: Create a factor for months

months <- c("March", "April", "January", "November", "January", "September", "October", "September", "November", "August", "January", "November", "November", "February", "May", "August", "July", "December", "August", "August", "September", "November", "February", "April")

Create factor for months

factor_months <- factor(months)

Print factor levels

factor_months

3. Summarize months vector and factor_months

summary(months) summary(factor_months)

The summary of the factor gives the count of each month, which is more useful than just summarizing the months vector.

Question 4: Create a vector and factor for direction frequencies

direction <- c("East", "West", "North", "West", "West", "West", "North", "East", "North") frequency <- c(1, 4, 3)

Factor with a specified order

factor_direction <- factor(direction, levels = c("East", "West", "North"))

Print new ordered factor

factor_direction

Question 5: Import CSV file into R and view dataset

a. Import the file import_march.csv

dataset <- read.table(file.path("C:", "Rworksheets", "worksheet#4", "import_march.csv"), header = TRUE, sep = ",", stringsAsFactors = FALSE)

b. View the dataset

head(dataset)

Question 6: Exhaustive search

exhaustive_search <- function(num) { if(num < 1 | num > 50) { return("The number selected is beyond the range of 1 to 50") } else if(num == 20) { return(TRUE) } else { return(num) } }

exhaustive_search(25) exhaustive_search(20)

Question 7: Change - minimum number of bills

min_bills <- function(price) { bills <- c(1000, 500, 200, 100, 50) count <- 0 for (bill in bills) { count <- count + floor(price / bill) price <- price %% bill } return(count) }

min_bills(1350)

Question 8: Student math scores

a. Create dataframe from student grades

students <- c("Annie", "Thea", "Steve", "Hanna") grade1 <- c(85, 65, 85, 100) grade2 <- c(65, 75, 90, 90) grade3 <- c(75, 55, 80, 85) grade4 <- c(95, 75, 100, 90) df_grades <- data.frame(students, grade1, grade2, grade3, grade4)

b. Calculate average score of students with avg score > 90

for (i in 1:nrow(df_grades)) { avg_score <- sum(df_grades[i, 2:5]) / 4 if (avg_score > 90) { print(paste(df_grades$students[i], "average grade this semester is", avg_score)) } }

c. Output tests with average score < 80

test_avg <- colMeans(df_grades[, 2:5]) for (i in 1:length(test_avg)) { if (test_avg[i] < 80) { print(paste("Test", i, "was difficult.")) } }

d. Output highest score exceeding 90 for each student

for (i in 1:nrow(df_grades)) { max_score <- max(df_grades[i, 2:5]) if (max_score > 90) { print(paste(df_grades$students[i], "highest grade this semester is", max_score)) } }