

Homework04.rmd

Peyton Hall

02/08/2024

```
# Suppose you own stocks and watch the stock market every day, write a function in R named  
#  
# using if-else conditional statements:  
#  
# return  
#  
# return  
#  
#  
# Include 4 arguments in your function: today  
# Note: to get the average of the past 3 days, use (day1+day2+day3)/3  
# Find the predicted decision if today  
# $10.39, $15.98, and $13.96.
```

```
stock_advice <- function(today, day1, day2, day3) {  
  average_past_3_days <- (day1 + day2 + day3) / 3  
  
  if (today < average_past_3_days) {  
    return("buy")  
  } else if (today > average_past_3_days) {  
    return("sell")  
  } else {  
    return("watch")  
  }  
}
```

```
today_value <- 12.11  
day1_value <- 10.39  
day2_value <- 15.98  
day3_value <- 13.96
```

```
decision <- stock_advice(today_value, day1_value, day2_value, day3_value)
print(decision)
```

Let x be a vector of (3,6,12,17,19,33,24,21,10,8,5). Use a for loop and a conditional statement to do the following:

If the x current element value is less than 15, then update that element with its current value-1.
(e.g. if the current x value is 3, then update it with 2). If the x current element value is larger than

or equal to 15, then no change will be made, and keep the current value.

```
x <- c(3, 6, 12, 17, 19, 33, 24, 21, 10, 8, 5)
```

```
for (i in 1:length(x)) {
  if (x[i] < 15) {
    x[i] <- x[i] - 1
  }
}
```

```
print(x)
```

Enter the data from the following table with three variables to create a

data frame

Name Education Salary

Jack Master 20000

Jerry Master 20000

Emma Master 20000

Andy Bachelor 30000

Jayde Bachelor 10000

Lynn PhD 70000

Liam PhD 70000

```
df <- data.frame(
  Name = c("Jack", "Jerry", "Emma", "Andy", "Jayde", "Lynn", "Liam"),
  Education = c("Master", "Master", "Master", "Bachelor", "Bachelor", "PhD", "PhD"),
  Salary = c(20000, 20000, 20000, 30000, 10000, 70000, 70000)
)
```

```
print(df)
```

```
# Use a loop and conditional statement to do the following:
# a. if the education level is Bachelor, increase the salary by 1000
# b. if the education level is Master, increase the salary by 2000
# c. if the education level is Ph.D. increase the salary by 3000
```

```
# Loop through each row of the data frame
for (i in 1:nrow(df)) {
  # Check the education level of each row
  if (df$Education[i] == "Bachelor") {
    # Increase salary by 1000 if Bachelor
    df$Salary[i] <- df$Salary[i] + 1000
  } else if (df$Education[i] == "Master") {
    # Increase salary by 2000 if Master
    df$Salary[i] <- df$Salary[i] + 2000
  } else if (df$Education[i] == "PhD") {
    # Increase salary by 3000 if PhD
    df$Salary[i] <- df$Salary[i] + 3000
  }
}
```

```
# Print the updated data frame
print(df)
```

```
# Suppose that you have a dataset with 8 individuals recording the blood
# pressure at baseline and
# a follow-up. Your goal is to show the correct treatment for these 8
# individuals. The rule is: if the
# baseline>follow-up, then the patient is recovered, and the treatment
# should show
# needed
# treatment needed
# if baseline=follow-up, then the treatment is
# vector of baseline and
# a vector of follow-up with the data in the following table. Use a loop and
# conditional statement
# to show the appropriate treatment. Please do NOT create any function.
# Patient 1 2 3 4 5 6 7 8
```

```

# Baseline 100 110 109 99 103 101 125 130
# Follow-up 105 100 102 105 100 97 108 120

# Create vectors for baseline and follow-up data
baseline <- c(100, 110, 109, 99, 103, 101, 125, 130)
follow_up <- c(105, 100, 102, 105, 100, 97, 108, 120)

# Initialize an empty vector to store treatment information
treatment <- vector()

# Loop through each individual's data
for (i in 1:length(baseline)) {
  # Check the condition and assign treatment accordingly
  if (baseline[i] > follow_up[i]) {
    treatment[i] <- "no treatment needed"
  } else if (baseline[i] < follow_up[i]) {
    treatment[i] <- "more treatment needed"
  } else {
    treatment[i] <- "to be determined"
  }
}

# Print the treatment vector
print(treatment)

# Write a double for-loop function to calculate the sum of all numbers in a
# matrix. The code on
# our lecture slide 20 is a good reference for this question. Suppose the
# matrix is
#
# (
# 36 51 40
# 39 78 62
# 22 37 52
# )
# Run the function with matrix A.

calculate_matrix_sum <- function(matrix) {

```

```
# Initialize a variable to store the sum
total_sum <- 0

# Loop through rows
for (i in 1:nrow(matrix)) {
  # Loop through columns
  for (j in 1:ncol(matrix)) {
    # Add each element to the total sum
    total_sum <- total_sum + matrix[i, j]
  }
}

# Return the total sum
return(total_sum)
}

# Define the matrix A
A <- matrix(c(36, 51, 40, 39, 78, 62, 22, 37, 52), nrow = 3, byrow = TRUE)

# Call the function with matrix A
result <- calculate_matrix_sum(A)
print(result)
```