Week4GroupActivities.rmd

Peyton Hall

2024-02-01

```
# Conditional statement
#  Activity
#  Suppose you hire a consulting firm to do some of your work and the original price is to pay $40 per
# hour. But you pay differently to different types of clients:
#    If it is for public clients, you pay 98% of the total price
#  If it is for private clients, you pay 95% of the total price
#  If it is for overseas clients, you pay 100% of the total price.
#  Name your function as price_consulting with two arguments: hours and client
#  Return the final calculated price
#  Call the function with hours=30, client ="public"

price_consulting <- function(hours, client) {
  # Original price per hour
  hourly_rate <- 40

  # Calculate total price
  total_price <- hours * hourly_rate

  # Apply conditional pricing based on client type
  if (client == "public") {
    final_price <- total_price * 0.98
  } else if (client == "private") {
    final_price <- total_price * 0.95
  } else if (client == "overseas") {
    final_price <- total_price  # 100% for overseas clients
  } else {
    # Handle unknown client types (optional)
    print("Unknown client type. Using default pricing.")
    final_price <- total_price
  }

  return(final_price)
}

# Call the function
hours <- 30
client_type <- "public"
result <- price_consulting(hours, client_type)
print(paste("Total price: $", result))
```

```
# a) What do you expect from the following loop without running the code?
# b) What is wrong with the syntax without running the code?

x=NULL
for(i in 1:5) {
x=x^2
print(x)
}

# a) I would expect a value to be returned, like 0 being printed 5 times.
# b) The problem with the syntax is the usage
#    of a mathematical process with a NULL value.
```

```
# Given the following code:
fib<-c(0,1)
for( i in 2:10){
New<-fib[i]+fib[i-1]
fib<-c(fib,New)
}
fib

# a) What are the outputs without running it?
# b) What is Fib[1],Fib[2], Fib[3],....?

# a) The console shows the following:
# > fib<-c(0,1)
# > for( i in 2:10){
# + New<-fib[i]+fib[i-1]
# + fib<-c(fib,New)
# + }
# > fib
#  [1]  0  1  1  2  3  5  8 13 21 34 55
# >
# a) The Environment shows the Values and Functions.
# b) Fib[1],Fib[2], Fib[3],.... etceters, represent
# the first 10 elements of the Fibonacci sequence.
```

```
# Use a for loop to find the sum of the first 100 squares
sum = 0
for (x in 1:100) {
  square <- x^2
  sum <- sum + square
}
sum
```

```
largest <- NULL
Predscore <- function(x){
for (i in 1:nrow(x)){
largest[i] <- x[i,1]
for (j in 1:ncol(x)){
if (x[i,j] > largest [i]){
largest [i] =x[i,j]
}
}
}
return(largest)
}
A<-matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=TRUE)
Predscore(A)

# What will this function return without running the code?
# (Note: Here X is a matrix.)

# This vector represents the largest element in each row of the matrix A.
```

```
# Example
# The data Brain volume on d2l describes the brain volume and weight for 22
# monkeys. Among them, 11 are females (sex=2) and 11 are males (sex=1). The
# year variable recorded the year
# of data collection. The variables of volume and weight recorded the brain
# volume and corresponding weight.
#  Use a for loop and conditional statement to do the following:
#  The researchers recorded the wrong weight.
#  If it is the year 1984, the actual weight needs to be increased by 10%.
#  If it is the other years, the actual weight needs to be increased by 5%.
# Note: length() is to find the length of a vector.

# Sample data
data <- data.frame(
  Sex = c(2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
  Year = c(1982, 1982, 1983, 1984, 1984, 1982, 1983, 1983, 1984, 1983, 1982, 1982, 1983, 1984, 1984, 1982, 1983,
1983, 1984, 1983, 1983, 1984),
  Volumn = c(1005, 963, 1035, 1027, 1281, 1272, 1051, 1079, 1034, 1070, 1173, 1079, 1067, 1104, 1347, 1439, 1029,
1100, 1204, 1160),
  Weight = c(57.607, 58.968, 64.184, 58.514, 63.958, 61.69, 133.358, 107.503, 62.143, 83.009, 61.236, 61.236, 83.
916, 79.38, 97.524, 99.792, 81.648, 88.452, 79.38, 72.576)
)

# For loop and conditional statements to adjust weights
for (i in 1:nrow(data)) {
  if (data$Year[i] == 1984) {
    data$Weight[i] <- data$Weight[i] * 1.1  # Increase by 10%
  } else {
    data$Weight[i] <- data$Weight[i] * 1.05  # Increase by 5%
  }
}

# Print the adjusted data
print(data)
```

```
# Researchers developed a new drug to treat migraine. They would like to
# compare the new
# drug with the current standard drug (Excedrin). The researchers recruited
# 10 patients and
# assigned them Excedrin. After a few months, when these 10 patients had
# migraine, they
# were assigned with the new drug. They recorded the time (in hours) to
# relief migraine from
# the 20 patients after they took their medications. The data is listed below

# Patients 1 2 3 4 5 6 7 8 9 10
# Excedrin 3.5 5.7 2.5 2 1.5 1 4 3 1 2
# New drug 3 2 3 1 0.5 2 1 1 1 0.5


# Create two vectors: one for Excedrin data; and one for new drug data.
# Use a loop of 1:10 and conditional statements to generate the output showing # whether the
# new drug is better than Excedrin for the 10 pairs of patients.
# Do you think the new drug works better?

# Excedrin and new drug data
excedrin_data <- c(3.5, 5.7, 2.5, 2, 1.5, 1, 4, 3, 1, 2)
new_drug_data <- c(3, 2, 3, 1, 0.5, 2, 1, 1, 1, 0.5)
ex<-0
nd<-0

# Loop and conditional statements to compare the two drugs
for (i in 1:10) {
  if (excedrin_data[patient] < new_drug_data[patient]) {
    ex <- ex + 1
  }else if (excedrin_data[patient]>new_drug_data[patient]) {
    nd <- nd + 1
  }
}
ifelse(ex < nd, "The New Drug Data works better then Excedrin for these 10
       patients.", "Excedrin works better than the new drug for these 10
       patients")
```