RWorksheet_Condag#4a

Angel Blase

2024-10-18

Data Frame

1. The table below shows the data about shoe size and height. Create a data frame.

```
data <- data.frame(
  Shoe_Size = c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 9.0, 13.0, 7.5, 10.5, 8.5, 12.0, 10.5, 13.0, 11.5, 8
  Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.5, 67.0, 71.0, 71.0, 77.0, 7
  Gender = c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F",
)
data
```

```
##    Shoe_Size Height Gender
## 1        6.5   66.0      F
## 2        9.0   68.0      F
## 3        8.5   64.5      F
## 4        8.5   65.0      F
## 5       10.5   70.0      M
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

a. Describe the data: The dataset contains information about individuals' shoe sizes, heights, and gender. It shows a range of shoe sizes from 5.0 to 13.0 and heights from 59 to 77 inches, with both male and female entries, allowing for analysis of potential relationships between these variables.

b. Create a subset by males and females with their corresponding shoe size and height.

```
male <- subset(data, Gender == "M", select = c(Shoe_Size, Height))
print("Male Data:")
```

```
## [1] "Male Data:"
```

```
male
```

```
##    Shoe_Size Height
## 5       10.5   70.0
## 9       13.0   72.0
## 11      10.5   74.5
## 13      12.0   71.0
## 14      10.5   71.0
## 15      13.0   77.0
## 16      11.5   72.0
## 19      10.0   72.0
## 22       8.5   67.0
## 23      10.5   73.0
## 25      10.5   72.0
## 26      11.0   70.0
## 27       9.0   69.0
## 28      13.0   70.0
```

```
female<- subset(data, Gender == "F", select = c(Shoe_Size, Height))
print("Female Data:")
```

```
## [1] "Female Data:"
```

```
female
```

```
##    Shoe_Size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 6        7.0   64.0
## 7        9.5   70.0
## 8        9.0   71.0
## 10       7.5   64.0
## 12       8.5   67.0
## 17       8.5   59.0
## 18       5.0   62.0
## 20       6.5   66.0
## 21       7.5   64.0
## 24       8.5   69.0
```

c. Find the mean of shoe size and height of the respondents.

```
shoeSize <- mean(data$Shoe_Size)
shoeSize
```

```
## [1] 9.410714
```

```
height <- mean(data$Height)
height
```

```
## [1] 68.57143
```

d. Is there a relationship between shoe size and height? Why?
   - Yes, people with taller heights are tend to have larger feet to support their body size which means that they have bigger shoe size compared to people who are not that tall.

Factors

2. Construct character vector months to a factor with factor() and assign the result to factor_months_vector.

```
months_vector <- 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")

factor_months_vector <- factor(months_vector)
factor_months_vector
```

```
##  [1] March     April     January   November  January   September October  
##  [8] September November  August    January   November  November  February 
## [15] May       August    July      December  August    August    September
## [22] November  February  April    
## 11 Levels: April August December February January July March May ... September
```

3. Then check the summary() of the months_vector and factor_months_vector.

```
summary(months_vector)
```

```
##    Length     Class      Mode 
##        24 character character 
```

```
summary(factor_months_vector)
```

```
##     April    August  December  February   January      July     March       May 
##         2         4         1         2         3         1         1         1 
##  November   October September 
##         5         1         3 
```

Interpret the results of both vectors. Are they both equally useful in this case?

```
- The months_vector summary only shows the total number of items (24), without
telling how often each month appears. The factor_months_vector summary lists
each month and how many times it shows up. This makes the factor vector more
useful for understanding the data, while the character vector is less helpful.
```

4. Create a vector and factor for the table below.

```
factor_data <- rep(c("East", "West", "North"), c(1, 4, 3))
new_order_data <- factor(factor_data,levels = c("East","West","North"))
new_order_data
```

```
## [1] East  West  West  West  West  North North North
## Levels: East West North
```

5. Enter the data below in Excel with file name = import_march.csv

a. Import the excel file into the Environment Pane using read.table() function.

```
stud_strategy<- read.table("import_march.csv", , header = TRUE, sep = ",", na.strings = "")
```

b. View the dataset.

```r
stud_strategy
```

```
##   Students Strategy.1 Strategy.2 Strategy.3
## 1     Male          8         10          8
## 2     Male          4          8          6
## 3     Male          0          6          4
## 4   Female         14          4         15
## 5   Female         10          2         12
## 6   Female          6          0          9
```

Using Conditional Statements (IF-ELSE)

6. Full Search

```r
input <- as.numeric(readline(prompt = "Select a number between 1 and 50: "))

if (input < 1 || input > 50) {
  print("The number selected is beyond the range of 1 to 50")
} else if (input == 20) {
  print("TRUE")
} else {
  print(paste("The number you selected is:", input))
}
```

7. Change

a. Write a function that prints the minimum number of bills that must be paid, given the price of the snack.

```r
min_bills <- function(price) {
  bills <- c(1000, 500, 200, 100, 50)
  bill_count <- 0

  for (bill in bills) {
    num_bills <- price %/% bill
    price <- price %% bill
    bill_count <- bill_count + num_bills
  }

  return(bill_count)
}

price <- as.numeric(readline(prompt = "Enter the price of the snack (divisible by 50): "))

if (price %% 50 == 0) {
  print(paste("Minimum number of bills needed:", min_bills(price)))
} else {
  print("Price must be divisible by 50.")
}
```

8. The following is each student's math score for one semester.

a. Create a dataframe from the above table.

```r
math_score <- data.frame(
  Name = c("Annie", "Thea", "Steve", "Hanna"),
  Grade1 = c(85, 65, 75, 95),
  Grade2 = c(65, 75, 55, 75),
  Grade3 = c(85, 90, 80, 100),
  Grade4 = c(100, 90, 85, 90))
math_score
```

```
##    Name Grade1 Grade2 Grade3 Grade4
## 1 Annie     85     65     85    100
## 2  Thea     65     75     90     90
## 3 Steve     75     55     80     85
## 4 Hanna     95     75    100     90
```

}b. Without using the rowMean function, output the average score of students whose average math score over 90 points during the semester.

```
for (i in 1:nrow(math_score)) {
  avg_grade <- (math_score$Grade1[i] + math_score$Grade2[i] + math_score$Grade3[i] + math_score$Grade4[
  if (avg_grade > 90) {
    print(paste(math_score$Name[i], "'s average grade this semester is", avg_grade))
  }
}
```

c. Without using the mean function, output as follows for the tests in which the average score was less than 80 out of 4 tests.

```
avg_test1 <- sum(math_score$Grade1) / nrow(math_score)
avg_test2 <- sum(math_score$Grade2) / nrow(math_score)
avg_test3 <- sum(math_score$Grade3) / nrow(math_score)
avg_test4 <- sum(math_score$Grade4) / nrow(math_score)

if (avg_test1 < 80) {
  print("The 1st test was difficult.")
}
if (avg_test2 < 80) {
  print("The 2nd test was difficult.")
}
```

```
## [1] "The 2nd test was difficult."
```

```
if (avg_test3 < 80) {
  print("The 3rd test was difficult.")
}
if (avg_test4 < 80) {
  print("The 4th test was difficult.")
}
```

d. Without using the max function, output as follows for students whose highest score for a semester exceeds 90 points.

```
for (i in 1:nrow(math_score)) {
  student_name <- math_score$Name[i]
  highest_score <- math_score$Grade1[i]

  if (math_score$Grade2[i] > highest_score) {
    highest_score <- math_score$Grade2[i]
  }
  if (math_score$Grade3[i] > highest_score) {
    highest_score <- math_score$Grade3[i]
  }
  if (math_score$Grade4[i] > highest_score) {
    highest_score <- math_score$Grade4[i]
  }

  if (highest_score > 90) {
    print(paste(student_name, "'s highest grade this semester is ", highest_score, ".", sep=""))
  } else {
    print(paste(student_name, "'s highest score does not exceed 90."))
  }
}
```

```
## [1] "Annie's highest grade this semester is 100."
## [1] "Thea 's highest score does not exceed 90."
## [1] "Steve 's highest score does not exceed 90."
## [1] "Hanna's highest grade this semester is 100."
```