RWorksheet_Quillo#4a

Carl

2023-10-25

```r
dfHouseholdData <- 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
                              "Height" = c(66.0,68.0,64.5,65.0,70.0,64.0,70.0,71.0,72.0,64.0,74.5,67.0,7
                              "Gender" = c("F","F","F","F","M","F","F","F","M","F","M","F","M","M","M",


# 1. B.Create a subset by males and females with their corresponding shoe size and height.
# What its result? Show the R scripts.

subMaleShoeSize <- subset(dfHouseholdData, Gender == 'M')
subMaleShoeSize
```

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

```r
subFemaleShoeSize <- subset(dfHouseholdData, Gender == 'F')
subFemaleShoeSize
```

```
##    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
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 10       7.5   64.0      F
## 12       8.5   67.0      F
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 24       8.5   69.0      F
```

```
#1.C Find the mean of shoe size and height of the respondents. Write the R scripts and its
# result.

shoeSizeandHeight <- mean(dfHouseholdData$Shoe.size & dfHouseholdData$Height)
shoeSizeandHeight
```

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

```
#1.D Is there a relationship between shoe size and height? Why?

# if the height is small the shoe size it also small


# 2 Construct character vector months to a factor with factor() and assign the result to
# factor_months_vector. Print out factor_months_vector and assert that R prints out
# the factor levels below the actual values.

months_vector <- c("March","April","January","November","January","September","October","September","No
months_vector
```

```
##  [1] "March"     "April"     "January"   "November"  "January"   "September"
##  [7] "October"   "September" "November"  "August"    "January"   "November"
## [13] "November"  "February"  "May"       "August"    "July"      "December"
## [19] "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. | Inter-
# pret the results of both vectors. Are they both equally useful in this case?

sumofmonths <- summary(months_vector)
sumofmonths
```

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

```
sumoffacmonths <- summary(factor_months_vector)
sumoffacmonths
```

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

```
# 4

direction_vector <- c("East", "West", "West", "West", "West", "North", "North", "North")

direction_factor <- factor(direction_vector, levels = c("East", "West", "North"))
```

```r
print(direction_factor)
```

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

```r
#5

excelData<- read.csv("import_match.csv")



#6a Create an R Program that allows the User to randomly select numbers from 1 to 50.
# Then display the chosen number. If the number is beyond the range of the selected choice,
# it will have to display a string "The number selected is beyond the range of 1 to 50". If
#number 20 is inputted by the User, it will have to display "TRUE", otherwise display the input number.

inputnum <- readline(paste("Eneter a number from 1 to 50"))
```

```
## Eneter a number from 1 to 50
```

```r
if(inputnum > 50){
  paste("You Enetered ", inputnum)
}else{
  paste("The number selected is beyond the range of 1 to 50")
}
```

```
## [1] "The number selected is beyond the range of 1 to 50"
```

```r
# 7

priceofsnack <- as.numeric(readline(paste("Enter the price of the snacks: ")))
```

```
## Enter the price of the snacks:
```

```r
numofbills <- function(priceofsnack){

  minBills <- priceofsnack %/% 50
  paste("The minimum number of bills is:", minBills)

}



# 8a
dfstudents <- data.frame("Name" = c("Annie","Thea", "Steve","Hanna"),
                         "Grade 1" = c(85,65,75,95),
                         "Grade 2" = c(65,75,55,75),
                         "Grade 3" = c(85,90,80,100),
                         "Grade 4" = c(100,90,85,90))
dfstudents
```

```
##    Name Grade.1 Grade.2 Grade.3 Grade.4
## 1 Annie      85      65      85     100
## 2  Thea      65      75      90      90
## 3 Steve      75      55      80      85
## 4 Hanna      95      75     100      90
```

```r
#8b

```

```r
test_averages <- rowSums(dfstudents[, -1]) / ncol(dfstudents[, -1])
test_averages
```

```
## [1] 83.75 80.00 73.75 90.00
```

```r
high_scorers <- dfstudents[test_averages >= 90, ]
high_scorers <- high_scorers[ 0, c("name")]
high_scorers$average_grade <- test_averages[test_averages> 90]

cat("if none appears means that no student has an average math score over 90 points during the semester
```

```
## if none appears means that no student has an average math score over 90 points during the semester
```

```r
#8c

test_averages <- rowSums(dfstudents[, -1]) / ncol(dfstudents[, -1])
test_averages
```

```
## [1] 83.75 80.00 73.75 90.00
```

```r
difficult_tests <- which(test_averages < 80)


if (length(difficult_tests) > 0) {
  cat(paste("The grade", paste(difficult_tests, collapse = ", "), "test(s) were difficult.\n"))
} else {
  cat("No test was difficult.\n")
}
```

```
## The grade 3 test(s) were difficult.
```

```r
#8d

highest_grades <- numeric(nrow(dfstudents))

# Find and store the highest grade for each student without using max
for (i in 1:nrow(dfstudents)) {
  student_scores <- dfstudents[i, 2:5]
  highest_grade <- student_scores[1]
  for (score in student_scores) {
    if (score > highest_grade) {
      highest_grade <- score
    }
  }
  highest_grades[i] <- highest_grade
}

# Check which students have a highest grade exceeding 90 and print the output
high_scorers <- dfstudents$Name[highest_grades > 90]

if (length(high_scorers) > 0) {
  for (i in 1:length(high_scorers)) {
    student_name <- high_scorers[i]
    student_highest_score <- highest_grades[dfstudents$Name == student_name]
    cat(paste(student_name, "'s highest grade this semester is", student_highest_score, ".\n"))
  }
} else {
  cat("No student had a highest grade exceeding 90 points this semester.\n")
}
```

```
## Annie 's highest grade this semester is 100 .
## Hanna 's highest grade this semester is 100 .
```