RWorksheet_Gallo#4a

Alfe Kevin P. Gallo

2023-10-25

```
#1

Household_Data <- data.frame(
  Shoe_size = c(6.5,9.0,8.5,8.5,10.5,7.0,9.5,9.0,13.0,7.5,10.5,8.5,12.0,10.5,13.0,11.5,8.5,5.0,10.0,6.5
  Height = c(66.0,68.0,64.0,65.0,70.0,64.0,70.0,71.0,72.0,64.0,74.0,67.0,71.0,71.0,77.0,72.0,59.0,62.0,
  Gender = c("F","F","F","F","M","F","F","F","M","F","M","F","M","M","M","M","F","F","M","F","F","M","M
)
Household_Data
```

```
##    Shoe_size Height Gender
## 1        6.5     66      F
## 2        9.0     68      F
## 3        8.5     64      F
## 4        8.5     65      F
## 5       10.5     70      M
## 6        7.0     64      F
## 7        9.5     70      F
## 8        9.0     71      F
## 9       13.0     72      M
## 10       7.5     64      F
## 11      10.5     74      M
## 12       8.5     67      F
## 13      12.0     71      M
## 14      10.5     71      M
## 15      13.0     77      M
## 16      11.5     72      M
## 17       8.5     59      F
## 18       5.0     62      F
## 19      10.0     72      M
## 20       6.5     66      F
## 21       7.5     64      F
## 22       8.5     67      M
## 23      10.5     73      M
## 24       8.5     69      F
## 25      10.5     72      M
## 26      11.0     70      M
## 27       9.0     69      M
## 28      13.0     70      M
```

```
write.csv(Household_Data, file = "Shoe_Sizes")
#1a Describe the data
#The males shoe size and height are bigger/higher than the females.
```

```r
#1b Create a subset by males and females with their corresponding shoe size and height. What its result

male_shoeH <- subset(Household_Data, Gender == "M")

female_shoeH <- subset(Household_Data, Gender == "F")

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

meanshoeSH <- mean(Household_Data$Shoe_size&Household_Data$Height)
meanshoeSH
```

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

```r
#1d Is there a relationship between shoe size and height? Why?

#Yes. The higher the height, the bigger the shoe size

#FACTORS
#2
Monthsss <- c("March","April","January","November","January","September","October","September","Novembe
factor_months_vector <- factor(Monthsss)
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
```

```r
#3. Then check the summary() of the months_vector and factor_months_vector
summary(Monthsss)
```

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

```r
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
```

```r
#The result discplayed how mony of the same months are there in the vector and also displayed its data

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

Direction <- c("East","West","North")
Frequency <- c(1,4,3)

factorDirect <- factor(Direction)
factorFreq <- factor(Frequency)

new_order_data <- factor(factorDirect,levels = c("East","West","North"))
print(new_order_data)
```

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

```r
new_order_data2 <- factor(factorFreq,levels = c(1,4,3))
print(new_order_data2)
```

```
## [1] 1 4 3
## Levels: 1 4 3
```

```r
#5. Enter the data below in Excel with file name = import_march.csv
#5a.
ExcelData <- read.csv("import_march.csv")
ExcelData
```

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

```r
#5b.
ExcelData
```

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

```r
#6a

inputNum <- readline(prompt="Enter number from 1 to 50:
                     ")
```

```
## Enter number from 1 to 50:
##
```

```r
if(inputNum>50){
  print("The number is beyond the range of 1 to 50")
}else{
  print("True")
}
```

```
## [1] "True"
```

```r
#7 Change

calculateMinBills <- function() {

  bills <- c(1000, 500, 200, 100, 50)


  amount <- as.numeric(readline("Enter the price of the snack (a multiple of 50 pesos): "))

  if (is.na(amount) || amount %% 50 != 0) {
    cat("Invalid input. Price must be a multiple of 50 pesos.\n")
    return()
```

```
  }

  snackprice <- amount


  bill1000 <- 0
  bill500 <- 0
  bill200 <- 0
  bill100 <- 0
  bill50 <- 0


  if (snackprice >= 1000) {
    bill1000 <- snackprice %/% 1000
    snackprice <- snackprice %% 1000
  }
  if (snackprice >= 500) {
    bill500 <- snackprice %/% 500
    snackprice <- snackprice %% 500
  }
  if (snackprice >= 200) {
    bill200 <- snackprice %/% 200
    snackprice <- snackprice %% 200
  }
  if (snackprice >= 100) {
    bill100 <- snackprice %/% 100
    snackprice <- snackprice %% 100
  }
  if (snackprice >= 50) {
    bill50 <- snackprice %/% 50
  }


  cat("Price:", amount, "pesos\n")
  cat("Minimum number of bills needed:\n")
  cat("1000 pesos:", bill1000, "bills\n")
  cat("500 pesos:", bill500, "bills\n")
  cat("200 pesos:", bill200, "bills\n")
  cat("100 pesos:", bill100, "bills\n")
  cat("50 pesos:", bill50, "bills\n")
}

calculateMinBills()
```

```
## Enter the price of the snack (a multiple of 50 pesos):
## Invalid input. Price must be a multiple of 50 pesos.
```

```
## NULL
```

```
#8a Create a data frame
GradesMath <- 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)
)
GradesMath
```

```
##    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
```

```
#8b.Without using the rowMean function, output the average score of students whose average math score o

GradesMath$Average <- (GradesMath$Grade1 + GradesMath$Grade2 + GradesMath$Grade3 + GradesMath$Grade4) /

HighGrades <-  GradesMath[GradesMath$Average > 90, ]

if(nrow(HighGrades)>0){
  print(HighGrades$Name,"'s average grade this semester is:",HighGrades)
}else{
  print("there is no student that got 90 average grades")
}
```

```
## [1] "there is no student that got 90 average grades"
```

```
#8c Without using the mean function, output as follows for the tests in which the average score was les

 AverageScores <- colMeans(GradesMath[, -1])

if (AverageScores[1] < 80) {
    print("The 1st test was difficult.\n")
}else if (AverageScores[2] < 80) {
    print("The 2nd test was difficult.\n")
}else if (AverageScores[3] < 80) {
    print("The 3rd test was difficult.\n")
}else if (AverageScores[4] < 80) {
    print("The 4th test was difficult.\n")
}else{
  print("No test that students find it difficult")
}
```

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

```
 #8d Without using the max function, output as follows for students whose highest score for a semester

 #Annie Scores

if (GradesMath[1,2] > GradesMath[1,3] && GradesMath[1,2] > GradesMath[1,4] && GradesMath[1,2] > GradesMa
  AnnieScores <-GradesMath[1,2]
} else if (GradesMath[1,3] > GradesMath[1,4] && GradesMath[1,3] > GradesMath[1,5]) {
  AnnieScores <- GradesMath[1,3]
} else if (GradesMath[1,4] > GradesMath[1,5] && GradesMath[1,2] > GradesMath[1,5]) {
  AnnieScores <- GradesMath[1,4]
} else {
  AnnieScores <- GradesMath[1,5]
}
```

```
# Thea Scores
if (GradesMath[2,2] > GradesMath[2,3] && GradesMath[2,2] >GradesMath[2,4] && GradesMath[2,2] > GradesMa
  TheaScores <- GradesMath[2,2]
} else if (GradesMath[2,3] > GradesMath[2,4] &&GradesMath[2,3] > GradesMath[2,5]) {
  theaScores <- mathgrades[2,3]
} else if (GradesMath[2,4] > GradesMath[2,5] && GradesMath[2,2] > GradesMath[2,5]) {
  TheaScores <- GradesMath[2,4]
} else {
  TheaScores <-GradesMath[2,5]
}

# Steve Scores
if (GradesMath[3,2] > GradesMath[3,3] &&GradesMath[3,2] > GradesMath[3,4] && GradesMath[3,2] >GradesMat
  SteveScores <- GradesMath[3,2]
} else if (GradesMath[3,3] > GradesMath[3,4] && GradesMath[3,3] > GradesMath[3,5]) {
 SteveScores <- GradesMath[2,3]
} else if (GradesMath[3,4] > GradesMath[3,5] && GradesMath[3,2] > GradesMath[3,5]) {
  SteveScores <- GradesMath[3,4]
} else {
  SteveScores <- GradesMath[3,5]
}

# Hanna Scores
if (GradesMath[4,2] > GradesMath[4,3] && GradesMath[4,2] > GradesMath[4,4] && GradesMath[4,2] > GradesMa
  HannaScores <- GradesMath[4,2]
} else if (GradesMath[4,3] > GradesMath[4,4] && GradesMath[4,3] > GradesMath[4,5]) {
  HannaScores <- mathgrades[2,3]
} else if (GradesMath[4,4] > GradesMath[4,5] && GradesMath[4,2] > GradesMath[4,5]) {
  HannaScores <- GradesMath[4,4]
} else {
  HannaScores <- GradesMath[4,5]
}

GradesMath$HighestGrades <- c(AnnieScores, TheaScores, SteveScores, HannaScores)


NinetyHighest <- GradesMath[GradesMath$HighestGrades > 90,]
NinetyHighest
```

```
##    Name Grade1 Grade2 Grade3 Grade4 Average HighestGrades
## 1 Annie     85     65     85    100   83.75           100
## 4 Hanna     95     75    100     90   90.00           100
```

```
if (nrow(NinetyHighest) > 0) {
  paste(NinetyHighest$Name, "'s highest grade this semester is", NinetyHighest$HighestGrade)
} else {
  paste("No students have an average math score over 90.")
}
```

```
## [1] "Annie 's highest grade this semester is 100"
## [2] "Hanna 's highest grade this semester is 100"
```