

RWorksheet_Nava#4a

Samantha Nicole Nava

2024-10-14

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

```
library(openxlsx)
Household_Data <- read.xlsx("household_data2.xlsx")
Household_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. The data shows the gender, height and shoe size.
- b. Male subset and Female subset

```
male_data <- subset(Household_Data, Gender == "M")
female_data <- subset(Household_Data, Gender == "F")
male_data
```

##	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
```

```
female_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
## 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
```

c.

```
mean_shoesize <- mean(Household_Data$`Shoe Size`)
```

```
## Warning in mean.default(Household_Data$`Shoe Size`): argument is not numeric or
## logical: returning NA
```

```
mean_shoesize
```

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

```
mean_height <- mean(Household_Data$Height)
```

```
mean_height
```

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

d. Is there a relationship between shoe size and height? Why? Yes. There is a relationship between shoe size and height. I noticed that those who have taller height have bigger shoe size.

FACTORS 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", "November", "August",
                  "January", "November", "November", "February", "May", "August",
                  "July", "December", "August", "August", "September", "November",
```

```
"February", "April")
```

```
factor_months_vector <- factor(months_vector)
```

3. Then check the `summary()` of the `months_vector` and `factor_months_vector`. | Interpret the results of both vectors. Are they both equally useful in this case?

```
summary_months_vector <- summary(months_vector)
summary_factor_months_vector <- summary(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
```

In this case, the factor vector is helpful. It has straightforward, organized information, whereas the character vector requires processing in order to extract similar information from it.

4.

```
direction <- c("East", "West", "North")
frequency <- c(1, 4, 3)
```

```
factor_data <- factor(direction, levels = c("East", "West", "North"))
```

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

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

5.

```
library(openxlsx)
data <- read.table("student.csv", header = TRUE, sep = ",", stringsAsFactors = TRUE)
```

```
print(head(data))
```

```
##      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
```

6. 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.

```
num <- readline(prompt="Enter a number from 1-50")
```

```
## Enter a number from 1-50
```

```
num
```

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

```
if (num>50){  
  print("The number selected is beyond the range of 1 to 50")  
} else if (num==20){  
  print("TRUE")  
} else {  
  num  
}
```

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

7. Change

a.

```
minBills <- function(price){  
  
  bills <- c(1000, 500, 200 , 100 ,50)  
  
  billCount <- 0  
  
  billCount <- billCount + price %/% bills[1]  
  price <- price %% bills[1]  
  
  billCount <- billCount + price %/% bills[2]  
  price <- price %% bills[2]  
  
  billCount <- billCount + price %/% bills[3]  
  price <- price %% bills[3]  
  
  billCount <- billCount + price %/% bills[4]  
  price <- price %% bills[4]  
  
  billCount <- billCount + price %/% bills[5]  
  price <- price %% bills[5]  
  
  cat("Minimum number of bills needed:", billCount, "\n")  
}  
  
minBills(8000)
```

```
## Minimum number of bills needed: 8
```

8.

a.

```
studentsScores <- 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)
```

```
)
```

```
(studentsScores)
```

```
##      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.

```
total_scores <- studentsScores$Grade1 + studentsScores$Grade2 + studentsScores$Grade3 + studentsScores$Grade4
```

```
average_scores <- total_scores / 4
```

```
for (i in 1:nrow(studentsScores)) {
  if (average_scores[i] > 90) {
    cat(studentsScores$Name[i], "'s average grade this semester is", round(average_scores[i], 2), "(above 90)")
  } else if (average_scores[i] == 90) {
    cat(studentsScores$Name[i], "'s average grade this semester is", round(average_scores[i], 2), "(exactly 90)")
  }
}
```

```
## Hanna 's average grade this semester is 90 (exactly 90).
```

c.

```
num_students <- nrow(studentsScores)
```

```
total_scores <- c(0, 0, 0, 0)
```

```
for (i in 1:num_students) {
  total_scores[1] <- total_scores[1] + studentsScores$Grade1[i]
  total_scores[2] <- total_scores[2] + studentsScores$Grade2[i]
  total_scores[3] <- total_scores[3] + studentsScores$Grade3[i]
  total_scores[4] <- total_scores[4] + studentsScores$Grade4[i]
}
```

```
for (j in 1:length(total_scores)) {
  average_score <- total_scores[j] / num_students
  if (average_score < 80) {
    cat("Test", j, "was difficult.\n")
  }
}
```

```
## Test 2 was difficult.
```

d.

```
highest_scores <- apply(studentsScores[2:5], 1, function(x) {
  highest_score <- x[1]
  for (score in x) {
    if (score > highest_score) {
```

```

        highest_score <- score
    }
}
return(highest_score)
})

for (i in 1:nrow(studentsScores)) {
  if (highest_scores[i] > 90) {
    cat(studentsScores$Name[i], "'s highest grade this semester is", highest_scores[i], ".\n")
  }
}

```

```

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

```