Untitled

Jeron J. Aurelio

2025-11-26

```
###1.The table below shows the data about shoe size and height. Create a data frame.
shoe_data <- data.frame(
  ShoeSize = c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 7.0, 7.5, 7.5, 8.5, 10.5,
               13.0, 11.5, 8.5, 5.0, 10.0, 6.5, 7.5, 10.5, 8.0, 11.0, 9.0, 13.0),
  Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 70.0, 71.0, 72.0, 64.0, 64.0, 67.0, 71.0,
             77.0, 72.0, 59.0, 60.0, 72.0, 66.0, 64.0, 69.0, 67.0, 70.0, 69.0, 70.0),
  Gender = c('F', 'F', 'F', 'F', 'M', 'F', 'F', 'F', 'F', 'F', 'F', 'M',
             'M', 'M', 'F', 'F', 'M', 'F', 'F', 'M', 'F', 'M', 'M', 'M')
)
shoe_data
```

```
##    ShoeSize 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   70.0      F
## 7       9.5   71.0      F
## 8       7.0   72.0      F
## 9       7.5   64.0      F
## 10      7.5   64.0      F
## 11      8.5   67.0      F
## 12     10.5   71.0      M
## 13     13.0   77.0      M
## 14     11.5   72.0      M
## 15      8.5   59.0      F
## 16      5.0   60.0      F
## 17     10.0   72.0      M
## 18      6.5   66.0      F
## 19      7.5   64.0      F
## 20     10.5   69.0      M
## 21      8.0   67.0      F
## 22     11.0   70.0      M
## 23      9.0   69.0      M
## 24     13.0   70.0      M
```

```
### A. Describe data
summary(shoe_data)
```

```
##     ShoeSize          Height         Gender
##  Min.   : 5.000   Min.   :59.00   Length:24
##  1st Qu.: 7.500   1st Qu.:64.88   Class :character
##  Median : 8.500   Median :68.50   Mode  :character
##  Mean   : 8.917   Mean   :67.81
##  3rd Qu.:10.500   3rd Qu.:70.25
##  Max.   :13.000   Max.   :77.00
```

```
### B.Create a subset by males and females with their corresponding shoe size and height.What its resul
female_data <- subset(shoe_data, Gender == "F", select = c(ShoeSize, Height))
female_data
```

```
##    ShoeSize Height
## 1       6.5   66.0
## 2       9.0   68.0
## 3       8.5   64.5
## 4       8.5   65.0
## 6       7.0   70.0
## 7       9.5   71.0
## 8       7.0   72.0
## 9       7.5   64.0
## 10      7.5   64.0
## 11      8.5   67.0
## 15      8.5   59.0
## 16      5.0   60.0
## 18      6.5   66.0
## 19      7.5   64.0
## 21      8.0   67.0
```

```
male_data <- subset(shoe_data, Gender == "M", select = c(ShoeSize, Height))
male_data
```

```
##    ShoeSize Height
## 5      10.5     70
## 12     10.5     71
## 13     13.0     77
## 14     11.5     72
## 17     10.0     72
## 20     10.5     69
## 22     11.0     70
## 23      9.0     69
## 24     13.0     70
```

```
### C.Find the mean of shoe size and height of the respondents. Write the R scripts and its result.
mean_shoe <- mean(shoe_data$ShoeSize)
mean_height <- mean(shoe_data$Height)

mean_shoe
```

```
## [1] 8.916667
```

```
mean_height
```

```
## [1] 67.8125
```

```
# D
correlation <- cor(shoe_data$ShoeSize, shoe_data$Height)
correlation
```

```
## [1] 0.6723337
```

```
### 2.Construct character vector months to a factor with factor() and assign the result to factor_month
# Character vector of months (copied exactly)
```

```r
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
```

```r
### 3.Then check the summary() of the months_vector and factor_months_vector. | Inter-pret the results 
summary("months_vector")
```

```
##    Length     Class      Mode
##         1 character character
```

```r
summary("factor_months_vector")
```

```
##    Length     Class      Mode
##         1 character character
```

```r
### 4.Create a vector and factor for the table below.
direction <- c("East", "West", "North")
frequency <- c(1, 4, 3)

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

print(factor_data)
```

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

```r
table_data <- data.frame(Direction = factor_data, Frequency = frequency)
print(table_data)
```

```
##   Direction Frequency
## 1      East         1
## 2      West         4
## 3     North         3
```

```r
### 6.Full Search
if (interactive()) {
  mode <- tolower(trimws(readline("Enter mode ('r' for random, 'm' for manual): ")))
} else {
  mode <- "r"
}

if (mode == "r") {
  chosen <- sample(1:50, 1)
  cat("Randomly chosen number:", chosen, "\n")
} else if (mode == "m") {
  if (interactive()) {
    input <- readline("Enter an integer: ")
    chosen_num <- suppressWarnings(as.integer(input))
    if (is.na(chosen_num)) stop("Invalid input: please enter an integer.")
    chosen <- chosen_num
  } else {
    chosen <- 20
    cat("Default number selected for knitting:", chosen, "\n")
  }
} else {
  stop("Invalid mode. Use 'r' or 'm'.")
}
```

```
## Randomly chosen number: 11
```

```r
if (chosen < 1 || chosen > 50) {
  cat("The number selected is beyond the range of 1 to 50\n")
} else if (chosen == 20) {
  cat("TRUE\n")
} else {
  cat("Selected number:", chosen, "\n")
}
```

```
## Selected number: 11
```

```r
### 7.Change
min_bills <- function(price) {
  bills <- c(1000, 500, 200, 100, 50)
  count <- 0

  remaining <- price

  for (b in bills) {
    if (remaining >= b) {
      n <- remaining %/% b      # how many of that bill
      count <- count + n
      remaining <- remaining - n*b
    }
  }

  return(count)
}

price <- sample(seq(50, 5000, by = 50), 1)
cat("Price of snack:", price, "\n")
```

```
## Price of snack: 3200
```

```r
cat("Minimum bills needed:", min_bills(price), "\n")
```

```
## Minimum bills needed: 4
```

```r
### 8.The following is each student’s math score for one semester. Based on this, answer the following

### A. Create a dataframe from the above table. Write the R codes and its output.
```

```r
math_scores <- 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)
)

print(math_scores)
```

```
##     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. write R code and its output.

```r
average_scores <- (math_scores$Grade1 + math_scores$Grade2 + math_scores$Grade3 + math_scores$Grade4) /

average_scores
```

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

```r
high_achievers <- math_scores[average_scores > 90, ]

high_achievers
```

```
## [1] Name   Grade1 Grade2 Grade3 Grade4
## <0 rows> (or 0-length row.names)
```

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.

```r
average_scores <- (math_scores$Grade1 + math_scores$Grade2 + math_scores$Grade3 + math_scores$Grade4) /

average_scores
```

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

```r
low_scores <- math_scores[average_scores < 80, ]

low_scores
```

```
##     Name Grade1 Grade2 Grade3 Grade4
## 3 Steve     75     55     80     85
```

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

```r
for (i in 1:nrow(math_scores)) {

  grades <- c(math_scores$Grade1[i], math_scores$Grade2[i], math_scores$Grade3[i], math_scores$Grade4[i
  highest <- grades[1]
  for (g in grades) {
    if (g > highest) {
      highest <- g
    }
  }

  if (highest > 90) {
    cat(math_scores$Name[i], "'s highest grade this semester is", highest, "\n")
  }
}
```

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