RWorksheet_Paclibar#4a

Jhon Albert B. Paclibar

2024-10-14

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

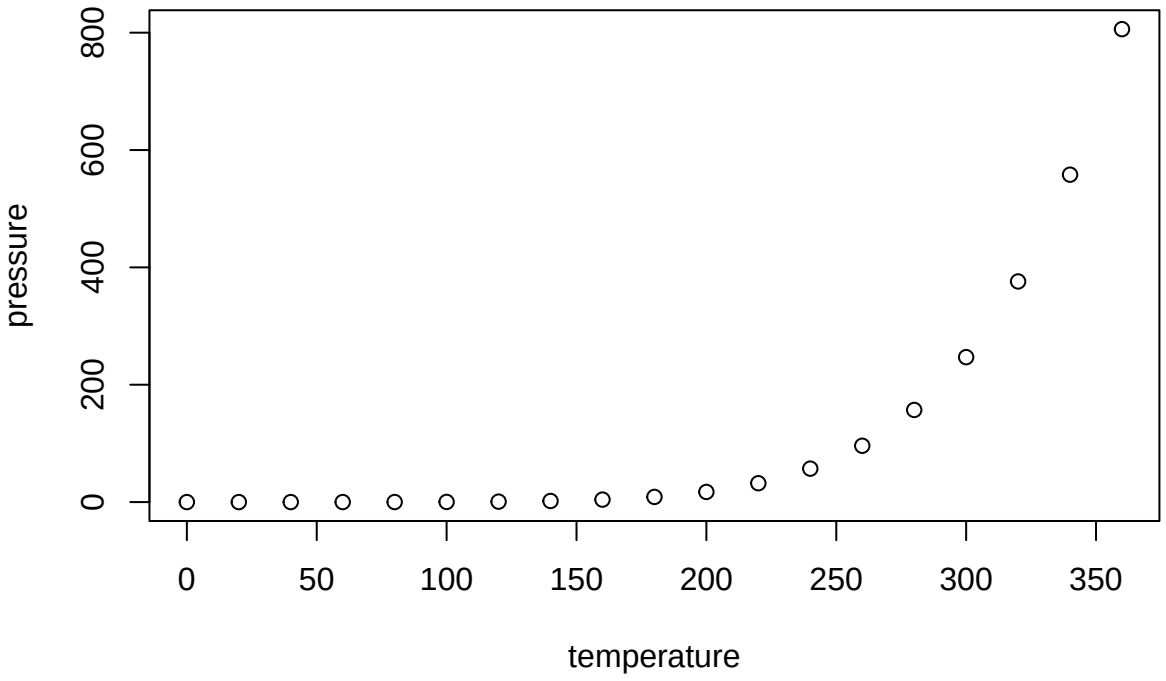
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

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

```
Shoe_size <- c(6.7,9.0,8.5,8.5,10.5,7.0,9.5,9.0,13.0,7.5,10.5,8.5,12.0,10.5, 13.0,11.5,8.5,5.0,10.0,6.5
Height <- c(66.0,68.0,64.5,65.0,70.0,64.0,70.0,71.0,72.0,64.0,74.5,67.0,71.0,71.0,77.0,72.0,59.0,62.0,7
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"
ShoeInfo <- data.frame(Shoe_size,Height,Gender)
print(ShoeInfo)
```

```
##    Shoe_size Height Gender
## 1        6.7   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 data shows the shoe size, height, and the gender of the people.

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

```
msd <- subset(ShoeInfo , Gender == "M")
msd
```

```
##    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
```

```
fsd <- subset(ShoeInfo , Gender == "F")
fsd
```

```
##    Shoe_size Height Gender
## 1        6.7   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. Find the mean of shoe size and height of the respondents. Write the R scripts and its result.

```
means <- mean(ShoeInfo$Shoe_size)
means
```

```
## [1] 9.417857
```

```
meanh <- mean(ShoeInfo$Height)
meanh
```

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

#d. Is there a relationship between shoe size and height? Why? #Yes, Because the taller you are the bigger the size of your feet.

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

```
months <- c("March","April","January","November","January",
"September","October","September","November","August",
"January","November","November","February","May","August","July","December","August","August","Septembe
"April")
factor_months_vector <- factor(months)
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
```

#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?

```
summary(months)
```

```
##    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
```

#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"))
factor_data
```

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

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

```
import_march <- read.csv("import_march.csv")
import_march
```

```
##   Students Strategy1 Strategy2 Strategy3
## 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
```

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

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

```
##   Students Strategy1 Strategy2 Strategy3
## 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
```

#b. View the dataset. Write the R scripts and its result.

```
import_march
```

```
##   Students Strategy1 Strategy2 Strategy3
## 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
```

#6. Full Search #Exhaustive search is a methodology for finding an answer by exploring all possible cases When trying to find a desired number in a set of given numbers, the method of finding the corresponding number by checking all elements in the set one by one can be called an exhaustive search. Implement an exhaustive search function that meets the input/output conditions below. #a. 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.

```
NumSel <- readline(prompt="Enter number: ")
```

```
## Enter number:
```

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

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

#7. Change #At ISATU University's traditional cafeteria, snacks can only be purchased with bills. Along-standing rule at the concession stand is that snacks must be purchased with as few coins as possible. There are three types of bills: 50 pesos, 100 pesos, 200 pesos, 500 pesos,1000 pesos. a. Write a function that prints the minimum number of bills that must be paid, given the price of the snack. Input: Price of snack (a random number divisible by 50) Output: Minimum number of bills needed to purchase a snack.

```
snackPrice<-readline(prompt = "Enter Amount: ")
```

```
## Enter Amount:
```

```
if (snackPrice == 50){
  print("The minimum bill is : 100")
}else if(snackPrice == 100){
  print("The minimum bill is : 100")
}else if(snackPrice == 200){
  print("The minimum bill is : 200")
}else if(snackPrice == 500){
  print("The minimum bill is : 500")
}else if(snackPrice == 1000){
  print("The minimum bill is : 1000")
}else{
  print("The number is not divisible by 50")
}
```

```
## [1] "The number is not divisible by 50"
```

#8. The following is each student's math score for one semester. Based on this, answer the following questions. #a. Create a dataframe from the above table. Write the R codes and its output.

```
Name<- c("Annie", "Thea", "Steve", "Anna")
Grade1<- c(85, 65, 75, 95)
```

```r
Grade2 <- c(65, 75, 55, 75)
Grade3<-c(85, 90, 80, 100)
Grade4<- c (100, 90, 85, 90)

Students <- data.frame(Name, Grade1, Grade2, Grade3, Grade4)
Students
```

```
##    Name Grade1 Grade2 Grade3 Grade4
## 1 Annie     85     65     85    100
## 2  Thea     65     75     90     90
## 3 Steve     75     55     80     85
## 4  Anna     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
for (i in 1:nrow(Students)) {
  avg_score <- (Students[i, "Grade1"] + Students[i, "Grade2"] + Students[i, "Grade3"] + Students[i, "Gra
  if (avg_score > 90) {
    cat(sprintf("%s's average grade this semester is %.2f. ", Students[i, "Name"], avg_score))
  }
}
```

#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
test_averages <- colMeans(Students[2:5])

# Check for tests with average less than 80
for (j in 1:length(test_averages)) {
  if (test_averages[j] < 80) {
    cat(sprintf("The %dnd test was difficult.\n", j))
  }
}
```

```
## The 2nd test was difficult.
```

```r
for (i in 1:nrow(Students)) {
  highest_score <- Students[i, 2:5][1]
  for (j in 2:4) {
    if (Students[i, j + 1] > highest_score) {
      highest_score <- Students[i, j + 1]
    }
  }
  if (highest_score > 90) {
    cat(sprintf("%s's highest grade this semester is %d.\n", Students$Name[i], highest_score))
  }
}
```

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