Rworksheet_Juntanilla#4a.Rmd

2023-10-26

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.

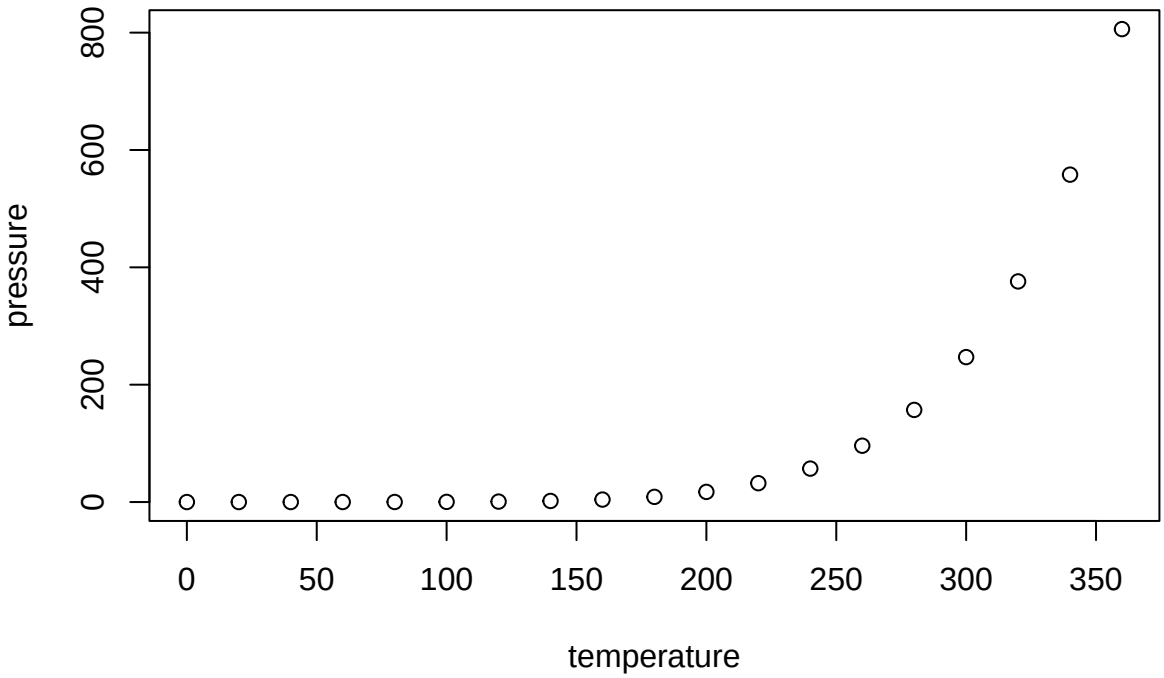
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.

```r
# code here:

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

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
Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.2, 67.0, 71.0, 71.0, 77.0, 72
Gender = c("F", "F", "F", "F", "M", "F","F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F", "M"
)

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   74.2      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
```

```r
# 1b.Create a subset by males and females with their corresponding shoe size and height.What its result
Household_subset_Male <- subset(Household_data,Gender == "M",select = c(Shoe_size,Height))
Household_subset_Male
```

```
##    Shoe_size Height
## 5       10.5   70.0
## 9       13.0   72.0
## 11      10.5   74.2
## 13      12.0   71.0
## 14      10.5   71.0
## 15      13.0   77.0
## 16      11.5   72.0
## 19      10.0   72.0
```

```
## 22       8.5   67.0
## 23      10.5   73.0
## 25      10.5   72.0
## 26      11.0   70.0
## 27       9.0   69.0
## 28      13.0   70.0
```

```
Household_subset_Female <- subset(Household_data,Gender == "F",select = c(Shoe_size,Height))
Household_subset_Female
```

```
##    Shoe_size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 6        7.0   64.0
## 7        9.5   70.0
## 8        9.0   71.0
## 10       7.5   64.0
## 12       8.5   67.0
## 17       8.5   59.0
## 18       5.0   62.0
## 20       6.5   66.0
## 21       7.5   64.0
## 24       8.5   69.0
```

```
# 1c.Find the mean of shoe size and height of the respondents. Write the R scripts and its result.
mean_shoe_size <- mean(Household_data$Shoe_size)
mean_shoe_size
```

```
## [1] 9.410714
```

```
mean_Height <- mean(Household_data$Height)
mean_Height
```

```
## [1] 68.56071
```

```
# 1d. Is there a relationship between shoe size and height? Why?
# Answer: Yes, because the more taller you are the bigger shoe size you will have.

# 2. Construct character vector months to a factor with factor() and assign the result to factor_months
Months <- c("March", "April", "January", "November", "January", "September", "October", "September", "N
Months
```

```
##  [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 <- factor(Months)
Factor_months
```

```
##  [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 of b
summary(Factor_months)
```

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

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

Direction_factor <- factor(Direction_vector,levels = c("East", "West", "North"))
print(Direction_factor)
```

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

```r
Frequency_factor <- factor(Frequency_vector,levels = c(1,4,3))
print(Frequency_factor)
```

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

```r
# 5a.Import the excel file into the Environment Pane using read.table() function.Write the code.
library(readr)
import_march <- read_csv("/cloud/project/import_march.csv")
```

```
## Rows: 6 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Students
## dbl (3): Strategy 1, Strategy 2, Strategy 3
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# 5b.View the dataset. Write the R scripts and its result.
head(import_march)
```

```
## # A tibble: 6 x 4
##   Students `Strategy 1` `Strategy 2` `Strategy 3`
##   <chr>           <dbl>        <dbl>        <dbl>
## 1 Male                8           10            8
## 2 <NA>                4            8            6
## 3 <NA>                0            6            4
## 4 Female             14            4           15
## 5 <NA>               10            2           12
## 6 <NA>                6            0            9
```

6.

random_num <- as.numeric(readline(prompt = "Enter a number from 1 to 50:"))

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

```r
# 7.
minimum_bill_price <- function(price) {

minimum_price<- price %% 50
paste("The minimum number of bills:", minimum_price)

}
 minimum_bill_price(280)
```

```
## [1] "The minimum number of bills: 30"
```

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

Grade_stud_data <- 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)
)
Grade_stud_data
```

```
##    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
```

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

Grade_stud_data$average <- (Grade_stud_data$Grade1 + Grade_stud_data$Grade2 + Grade_stud_data$Grade3 + 
mathgrade <- Grade_stud_data[Grade_stud_data$average > 90, ]


if(nrow(mathgrade) > 0 ) {
print(mathgrade$Name," 's average grade this semester is:", mathgrade)
} else {
print("No student got 90 Average on Math Subject")
}
```

```
## [1] "No student got 90 Average on Math Subject"
```

```r
# c. Without using the mean function, output as follows for the tests in which the average score was le
students_ave_score <- colMeans(Grade_stud_data[, -1])

if(students_ave_score[1] < 80){
cat("The 1st test was difficult.")
}else if(students_ave_score[2] < 80){
cat("The 2nd test was difficult.")
}else if(students_ave_score[3] < 80){
cat("The 3rd test was difficult.")
}else if(students_ave_score[4] < 80){
cat("The 4th test was difficult." )
}else {
cat("Students did not find the test difficult")
}
```

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

```r
# d.Without using the max function, output as follows for students whose highest score for a semester e
# Assuming Grade_stud_data data frame is already defined as you provided

ave_exceed_90 <- character(0)

for (i in 1:nrow(Grade_stud_data)) {
  student <- Grade_stud_data[i, ]
  student_name <- student$Name

  if (any(student[-1] > 90)) {
    ave_exceed_90 <- c(ave_exceed_90, student_name)
  }
}

if (length(ave_exceed_90) > 0) {
  cat("Students whose highest score exceeds 90 points:\n")
  cat(ave_exceed_90, "\n")
} else {
  cat("No students have a highest score exceeding 90 points.\n")
}
```

```
## Students whose highest score exceeds 90 points:
## Annie Hanna
```