RWORKSHEET_GENTAPAO#2

BEA JULIETTE L. GENTAPAO

2025-10-01

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.

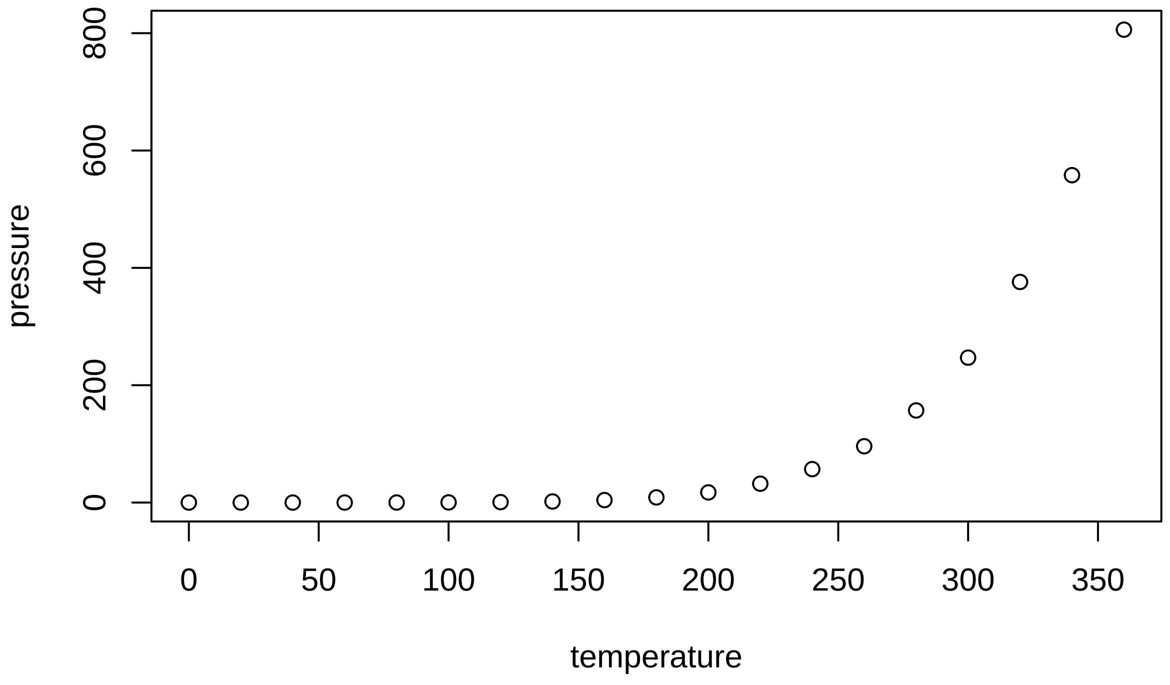
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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. Create a vector using : operator

```r
# 1a. Sequence from -5 to 5
seq_neg5_to_5 <- -5:5
seq_neg5_to_5
```

```
##  [1] -5 -4 -3 -2 -1  0  1  2  3  4  5
```

Output: -5 -4 -3 -2 -1 0 1 2 3 4 5

Description: The : operator creates a sequence of integers from -5 to 5 with increments of 1. It produces 11 elements total.

```r
# 1b. x <- 1:7. What will be the value of x?
x <- 1:7
x
```

```
## [1] 1 2 3 4 5 6 7
```

Output: 1 2 3 4 5 6 7

Description: x contains integers from 1 to 7

2. Create a vector using seq() function

```r
# 2a. seq(1, 3, by=0.2)
seq_result <- seq(1, 3, by=0.2)
seq_result
```

```
##  [1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0
```

Output: 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

Description: The seq() function creates a sequence from 1 to 3 with a step size of 0.2. It generates 11 elements, incrementing by 0.2 each time.

3. Factory census of workers

```r
# Workers' ages
workers_age <- c(34, 28, 22, 36, 27, 18, 52, 39, 42, 29, 35, 31, 27,
                 22, 37, 34, 19, 20, 57, 49, 50, 37, 46, 25, 17, 37,
                 43, 53, 41, 51, 35, 24, 33, 41, 53, 40, 18, 44, 38,
                 41, 48, 27, 39, 19, 30, 61, 54, 58, 26, 18)
```

```r
# 3a. Access 3rd element
workers_age[3]
```

```
## [1] 22
```

Output: 22

```r
# 3b. Access 2nd and 4th element
workers_age[c(2, 4)]
```

```
## [1] 28 36
```

Output: 28 36

```
# 3c. Access all but the 1st element
workers_age[-1]
```

```
##  [1] 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50 37 46 25 17 37
## [26] 43 53 41 51 35 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18
```

Output: 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50 37 46 25 17 37 43 53 41 51 35 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18

Description: This returns all elements except the first one (34 is excluded)

4. Create a named vector

```
x <- c("first"=3, "second"=0, "third"=9)
names(x)
```

```
## [1] "first"  "second" "third"
```

Output: "first" "second" "third"

```
# 4a. Print results and access x[c("first", "third")]
print(x)
```

```
##  first second  third
##      3      0      9
```

```
x[c("first", "third")]
```

```
## first third
##     3     9
```

Output:

```
first second  third
    3      0      9

first third
    3     9
```

Description: This accesses the elements named "first" and "third" from the named vector, returning their values (3 and 9).

5. Create a sequence x from -3:2

```
x <- -3:2
x
```

```
## [1] -3 -2 -1  0  1  2
```

Output: -3 -2 -1 0 1 2

```
# 5a. Modify 2nd element and change it to 0
x[2] <- 0
x
```

```
## [1] -3  0 -1  0  1  2
```

Output: -3 0 -1 0 1 2

Description: The second element (originally -2) is changed to 0. The vector now has -3 as first element, 0 as second element, and continues with -1, 0, 1, 2.

6. Diesel fuel purchased by Mr. Cruz

```
# 6a. Create a data frame
month <- c("Jan", "Feb", "March", "Apr", "May", "June")
price_per_liter <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
purchase_quantity <- c(25, 30, 40, 50, 10, 45)

fuel_data <- data.frame(
  Month = month,
  Price_per_liter_PhP = price_per_liter,
  Purchase_quantity_Liters = purchase_quantity
)

print(fuel_data)
```

```
##   Month Price_per_liter_PhP Purchase_quantity_Liters
## 1   Jan               52.50                       25
## 2   Feb               57.25                       30
## 3 March               60.00                       40
## 4   Apr               65.00                       50
## 5   May               74.25                       10
## 6  June               54.00                       45
```

```
# 6b. Average fuel expenditure
average_expenditure <- weighted.mean(price_per_liter, purchase_quantity)
average_expenditure
```

```
## [1] 59.2625
```

Output: 59.2625

Description: This represents the weighted average price per liter based on purchase quantity

7. Built-in dataset: rivers

```
# 7a. Create a vector data with 7 elements
data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers),
          var(rivers), sd(rivers), min(rivers), max(rivers))
```

```
# 7b. Results
data
```

```
## [1]    141.0000  83357.0000    591.1844    425.0000 243908.4086    493.8708
## [7]    135.0000   3710.0000
```

Output: 141.0000 83357.0000 591.1844 425.0000 243908.4086 493.8708 135.0000 3710.0000

```
# 7c. Code and outputs
print("Length of rivers:")
```

```
## [1] "Length of rivers:"
```

```r
length(rivers)
```

```
## [1] 141
```

```r
print("Sum of rivers:")
```

```
## [1] "Sum of rivers:"
```

```r
sum(rivers)
```

```
## [1] 83357
```

```r
print("Mean of rivers:")
```

```
## [1] "Mean of rivers:"
```

```r
mean(rivers)
```

```
## [1] 591.1844
```

```r
print("Median of rivers:")
```

```
## [1] "Median of rivers:"
```

```r
median(rivers)
```

```
## [1] 425
```

```r
print("Variance of rivers:")
```

```
## [1] "Variance of rivers:"
```

```r
var(rivers)
```

```
## [1] 243908.4
```

```r
print("Standard deviation of rivers:")
```

```
## [1] "Standard deviation of rivers:"
```

```r
sd(rivers)
```

```
## [1] 493.8708
```

```r
print("Minimum of rivers:")
```

```
## [1] "Minimum of rivers:"
```

```r
min(rivers)
```

```
## [1] 135
```

```r
print("Maximum of rivers:")
```

```
## [1] "Maximum of rivers:"
```

```r
max(rivers)
```

```
## [1] 3710
```

8. Forbes 25 Most Powerful Celebrities

```r
# 8a. Create vectors according to the table
power_ranking <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
                   16, 17, 18, 19, 20, 21, 22, 23, 24, 25)

celebrity_name <- c("Tom Cruise", "Rolling Stones", "Oprah Winfrey",
                    "U2", "Tiger Woods", "Steven Spielberg",
                    "Howard Stern", "50 Cent", "Cast of Sopranos",
                    "Dan Brown", "Bruce Springsteen", "Donald Trump",
                    "Muhammad Ali", "Paul McCartney", "George Lucas",
                    "Elton John", "David Letterman", "Phil Mickelson",
                    "J.K. Rowling", "Bradd Pitt", "Peter Jackson",
                    "Dr. Phil McGraw", "Jay Lenon", "Celine Dion",
                    "Kobe Bryant")

pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40,
         233, 34, 40, 47, 75, 25, 39, 45, 32, 40, 31)

# Create a data frame for better organization
forbes_data <- data.frame(
  PowerRanking = power_ranking,
  CelebrityName = celebrity_name,
  Pay = pay
)

print(forbes_data)
```

```
##    PowerRanking     CelebrityName Pay
## 1             1        Tom Cruise  67
## 2             2    Rolling Stones  90
## 3             3     Oprah Winfrey 225
## 4             4                U2 110
## 5             5       Tiger Woods  90
## 6             6  Steven Spielberg 332
## 7             7      Howard Stern 302
## 8             8           50 Cent  41
## 9             9  Cast of Sopranos  52
## 10           10         Dan Brown  88
## 11           11 Bruce Springsteen  55
## 12           12      Donald Trump  44
## 13           13      Muhammad Ali  55
## 14           14    Paul McCartney  40
## 15           15      George Lucas 233
## 16           16        Elton John  34
## 17           17   David Letterman  40
## 18           18    Phil Mickelson  47
## 19           19      J.K. Rowling  75
## 20           20        Bradd Pitt  25
## 21           21     Peter Jackson  39
## 22           22   Dr. Phil McGraw  45
## 23           23         Jay Lenon  32
## 24           24       Celine Dion  40
## 25           25       Kobe Bryant  31
```

```r
# 8b. Modify power ranking and pay of J.K. Rowling
# J.K. Rowling is at index 19
forbes_data[19, "PowerRanking"] <- 15
```

```r
forbes_data[19, "Pay"] <- 90

print("Modified Forbes Data:")
```

```
## [1] "Modified Forbes Data:"
```

```r
print(forbes_data)
```

```
##    PowerRanking     CelebrityName Pay
## 1             1        Tom Cruise  67
## 2             2    Rolling Stones  90
## 3             3     Oprah Winfrey 225
## 4             4                U2 110
## 5             5       Tiger Woods  90
## 6             6  Steven Spielberg 332
## 7             7      Howard Stern 302
## 8             8           50 Cent  41
## 9             9  Cast of Sopranos  52
## 10           10         Dan Brown  88
## 11           11 Bruce Springsteen  55
## 12           12      Donald Trump  44
## 13           13      Muhammad Ali  55
## 14           14    Paul McCartney  40
## 15           15      George Lucas 233
## 16           16        Elton John  34
## 17           17   David Letterman  40
## 18           18    Phil Mickelson  47
## 19           15      J.K. Rowling  90
## 20           20        Bradd Pitt  25
## 21           21     Peter Jackson  39
## 22           22   Dr. Phil McGraw  45
## 23           23         Jay Lenon  32
## 24           24       Celine Dion  40
## 25           25       Kobe Bryant  31
```

```r
# Verify the change
print("J.K. Rowling's updated information:")
```

```
## [1] "J.K. Rowling's updated information:"
```

```r
print(forbes_data[19, ])
```

```
##    PowerRanking CelebrityName Pay
## 19           15  J.K. Rowling  90
```

```r
# 8c. Interpretation of the data:
cat("\n=== INTERPRETATION ===\n")
```

```
##
## === INTERPRETATION ===
```

```r
cat("The Forbes 25 Most Powerful Celebrities ranking shows:\n")
```

```
## The Forbes 25 Most Powerful Celebrities ranking shows:
```

```r
cat("- Power rankings range from 1 to 25, with 1 being most powerful\n")
```

```
## - Power rankings range from 1 to 25, with 1 being most powerful
```

```r
cat("- Annual pay varies significantly, from $25M to $332M\n")
```

```
## - Annual pay varies significantly, from $25M to $332M
```

```r
cat("- Steven Spielberg has the highest pay at $332M\n")
```

```
## - Steven Spielberg has the highest pay at $332M
```

```r
cat("- Oprah Winfrey ranks 3rd in power with $225M pay\n")
```

```
## - Oprah Winfrey ranks 3rd in power with $225M pay
```

```r
cat("- After modification, J.K. Rowling's power ranking changed from 19 to 15\n")
```

```
## - After modification, J.K. Rowling's power ranking changed from 19 to 15
```

```r
cat("- Her pay increased from $75M to $90M\n")
```

```
## - Her pay increased from $75M to $90M
```

```r
cat("- The data represents celebrity influence and earnings in the entertainment industry\n")
```

```
## - The data represents celebrity influence and earnings in the entertainment industry
```