RWorksheet#1

Tamonan

2024-09-18

#1. Set up a vector named age

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, 42, 53, 41, 51, 35, 24, 33, 41)

#a. How many data points?

num_data_points <- length(age)

#b. Write the R code and its output.

num_data_points

#2. Find the reciprocal of the values for age.

reciprocal_age <- 1 / age

#Write the R code and its output.

reciprocal_age

#3. Assign also new_age <- c(age, 0, age).

new_age <- c(age, 0, age)

#What happen to the new_age?

new_age

#4. Sort the values for age.

sorted_age <- sort(age)

#Write the R code and its output.

sorted_age

#5. Find the minimum and maximum value for age.

min_age <- min(age) max_age <- max(age)

#Write the R code and its output.

min_age
max_age

#6. Set up a vector named data

data <- c(2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3, 2.5, 2.3, 2.4, 2.7)

#a. How many data points?

num_data_points <- length(data)

#b. Write the R code and its output.

num_data_points

#7. Generates a new vector for data where you double every value of the data.

double_data <- data * 2

#What happen to the data?

double_data

#8. Generate a sequence #8.1 Integers from 1 to 100.

seq_1_to_100 <- seq(1, 100)

#8.2 Numbers from 20 to 60

seq_20_to_60 <- seq(20, 60)

#*8.3 Mean of numbers from 20 to 60

mean_20_to_60 <- mean(seq_20_to_60)

#*8.4 Sum of numbers from 51 to 91

seq_51_to_91 <- seq(51, 91) sum_51_to_91 <- sum(seq_51_to_91)

#*8.5 Integers from 1 to 1,000

max_first_10 <- max(seq(1, 1000)[1:10]) print(max_first_10)

#a. How many data points from 8.1 to 8.4?

seq_1_to_100 seq_20_to_60 mean_20_to_60 sum_51_to_91

#c. For 8.5 find only maximum data points until 10.

seq_1_to_1000_first_10 <- 1:10 print(seq_1_to_1000_first_10)

#9. *Print a vector with the integers between 1 and 100 that are not divisible by 3, 5 and 7 using filter option.

not_div_3_5_7 <- Filter(function(i) all(i %% c(3, 5, 7) != 0), seq(100))

#Write the R code and its output.

not_div_3_5_7

#10. Generate a sequence backwards of the integers from 1 to 100.

seq_backwards <- seq(100, 1)

#Write the R code and its output.

seq_backwards

#11. List all the natural numbers below 25 that are multiples of 3 or 5.

multiples_3_5 <- Filter(function(x) x %% 3 == 0 | x %% 5 == 0, 1:24) multiples_3_5

#Find the sum of these multiples.

sum_multiples_3_5 <- sum(multiples_3_5) sum_multiples_3_5

#12. Statements can be grouped together using braces '{' and '}'. A group of statements is sometimes called a block. Single statements are evaluated when a new line is typed at the end of the syntactically complete statement. Blocks are not evaluated until a new line is entered after the closing brace.

x <- 0 x <- {x + 5} x # Output: 5 (after adding 5 to the initial value of x)

#13. *Set up a vector named score

score <- c(72, 86, 92, 63, 88, 89, 91, 92, 75, 75, 77)

#Find x[2] and x[3]. Write the R code and its output.

score[2]
score[3]

#14. *Create a vector

a <- c(1, 2, NA, 4, NA, 6, 7)

#a. Change the NA to 999 using the codes print(a,na.print="-999").

print(a, na.print="-999")

#15. A special type of function calls can appear on the left hand side of the assignment operator as in > class(x) <- "foo".

name = readline(prompt="Input your name:") age = readline(prompt="Input your age:") print(paste("My name is",name, "and I am",age ,"years old.")) print(R.version.string)

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.

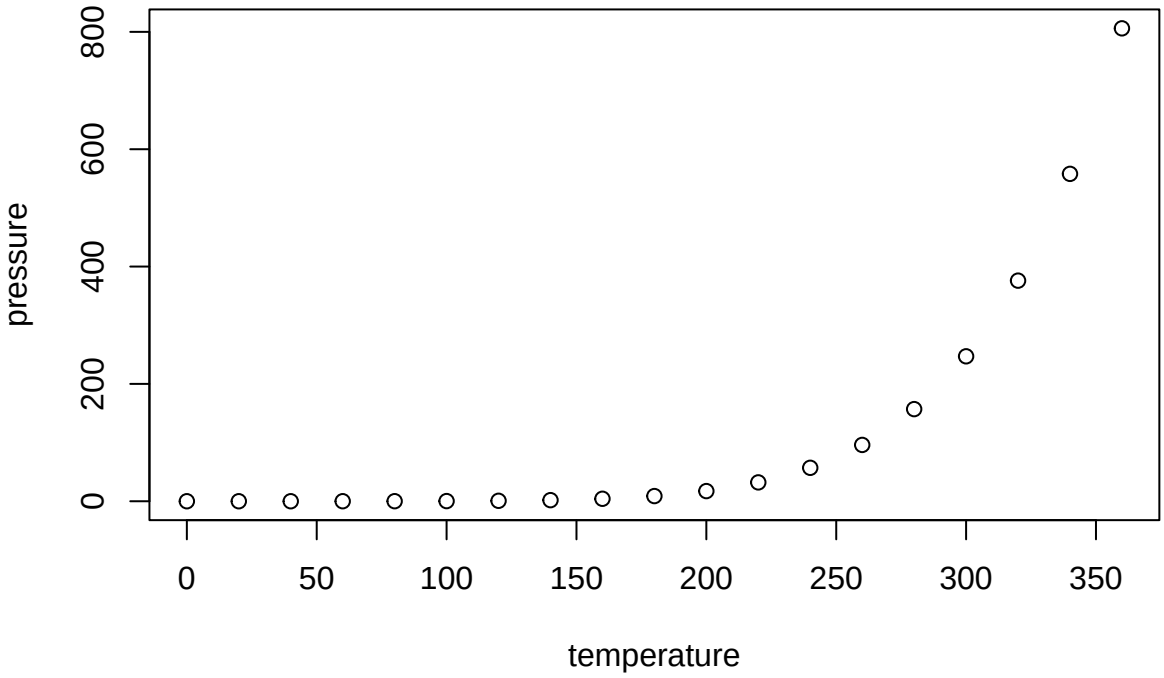
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.