RWorkSheet_Ti-ad

2024-09-17

#1.Setup 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 point AgeDataP <- length(age) AgeDataP

#2.Find the reciprocal of the values for age ReciproAge <- 1 / age ReciproAge

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

#4.Sort the values for age SortDecrease <- sort(age, decreasing = TRUE) SortDecrease

SortIncrease <- sort(age, decreasing = FALSE) SortIncrease

#5.Find the minimum and maximum value for age Age_min <- min(age) Age_min

Age_max <- max(age) Age_max

#6.Set up a vector named data, consisting of 2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3, 2.5, 2.3, 2.4, and 2.7 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)

dataDataP <- length(data) dataDataP

#7.Generates a new vector for data where you double every value of the data DoubleData <- data * 2 DoubleData

#8.Generate a sequence for the following scenario: #8.1 Integers from 1 to 100. OneTOHundred <- seq(1:100) OneTOHundred

#8.2 Numbers from 20 to 60 TwentyTOSixty <- seq(from = 20, to = 60, by = 1) TwentyTOSixty

#8.3 Mean of numbers from 20 to 60 Mean20TO60 <- mean(TwentyTOSixty) Mean20TO60

#8.4 Sum of numbers from 51 to 91 Sum51To91 <- sum(seq(from = 51, to = 91, by =1)) Sum51To91

#8.5 Integers from 1 to 1,000
INTEGER1TO1000 <- seq(1:1000) INTEGER1TO1000

#a.How many data points from 8.1 to 8.4 length(c(OneTOHundred, TwentyTOSixty, Mean20TO60, Sum51To91))

#c.For 8.5 find only maximum data points until 10 maxDTpnts10 <- INTEGER1TO1000[1:10] maxDTpnts10 length(maxDTpnts10)

#9.Print a vector with the integers between 1 and 100 that are not divisible by 3, 5 and 7 using filter option divisibleVector <- Filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(1:100)) divisibleVector

#10.Generate a sequence backwards of the integers from 1 to 100. reverse1to100 <- sort(seq(1:100), decreasing = TRUE) reverse1to100

#11.List all the natural numbers below 25 that are multiples of 3 or 5. Find the sum of these multiples. multiples <- Filter(function(i) {any(i %% c(3,5) == 0)}, seq(24)) multiples SumOfMultiples <- sum(multiples) SumOfMultiples

#a.How many data points from 10 to 11? LENGTH10_11 <- length(c(reverse1to100, multiples, SumOfMultiples)) LENGTH10_11

#12. x <- {0 + x + 5 + }

#13.*Set up a vector named score, consisting of 72, 86, 92, 63, 88, 89, 91, 92, 75, 75 and 77. To access individual elements of an atomic vector, one generally uses the x[i] construction. Find x[2] and x[3]. score <- c(72, 86, 92, 63, 88, 89, 91, 92, 75, 75, 77) score2 <- score[2] score2 score3 <- score[3] score3

#14.*Create a vector a = c(1,2,NA,4,NA,6,7) 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 assignmentoperator 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)