|  |
| --- |
| # R program to illustrate  # the use of min() function    # Creating a matrix  arr = array(2:13, dim = c(2, 3, 2))  print(arr)    # Using min() function  min(arr)  max(arr) |

|  |
| --- |
| # create a list of numbers  ls <- list(11, 9, 33, 65, 46)  # min value in list  min(ls)  #Error in min(ls): invalid 'type' (list) of argument  #You can see that we get an error. This is because we cannot apply the min() function in R directly to a list.  #Now let’s use the above syntax,i.e  #first convert the list to a vector using unlist() and then apply the min() function.  ls <- list(11, 9, 33, 65, 46)  # min value in list  min(unlist(ls))  # create a list with NA values  ls <- list(1, 3, NA, 5, NA, 4, NA)  # max value in the list  max(unlist(ls), na.rm=TRUE)  # create three lists  ls1 <- list("milk", "eggs")  ls2 <- list("butter", "cheese", "jam")  ls3 <- list("bread")  # combine lists  ls <- c(ls1, ls2, ls3)  # display the resulting list  print(ls) |

|  |
| --- |
| # Create two 2x3 matrices.  matrix1 <- matrix(c(10, 20, 30, 40, 50, 60), nrow = 2)  print(matrix1)  matrix2 <- matrix(c(5, -2, 4, 9, 3, 6), nrow = 2)  print(matrix2)  # Add the matrices.  result <- matrix1 + matrix2  cat("Result of addition","\n")  print(result)  # Subtract the matrices  result <- matrix1 - matrix2  cat("Result of subtraction","\n")  print(result) |

|  |
| --- |
| # Create a vector.  x <- c(12,7,3,4.2,18,2,54,-21,8,-5)  # Find Mean.  result.mean <- mean(x)  print(result.mean)  # trim  result.mean <- mean(x,trim = 0.3)  print(result.mean)  #Applying NA Option  #If there are missing values, then the mean function returns NA.  #To drop the missing values from the calculation use na.rm = TRUE. which means remove the NA values.  # Create a vector.  x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)  # Find mean.  result.mean <- mean(x)  print(result.mean)  # Find mean dropping NA values.  result.mean <- mean(x,na.rm = TRUE)  print(result.mean)  median(x, na.rm = FALSE)  # Create the vector.  x <- c(12,7,3,4.2,18,2,54,-21,8,-5)  # Find the median.  median.result <- median(x)  print(median.result)  # Create the function.  getmode <- function(v) {  uniqv <- unique(v)  uniqv[which.max(tabulate(match(v, uniqv)))]  }  # Create the vector with numbers.  v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)  # Calculate the mode using the user function.  result <- getmode(v)  print(result)  # Create the vector with characters.  charv <- c("o","it","the","it","it")  # Calculate the mode using the user function.  result <- getmode(charv)  print(result) |

|  |
| --- |
| # VECTOR WITH VALUES (NON ZERO VALUES) CONSIDERED AS "TRUE"  x <- c(TRUE,FALSE,0,6)  y <- c(FALSE,TRUE,FALSE,TRUE)  !x  x&y  x|y |

|  |
| --- |
| fahrenheit\_to\_celsius <- function(temp\_F)  {  # Converts Fahrenheit to Celsius  temp\_C <- (temp\_F - 32) \* 5 / 9  return(temp\_C)  }  # > fahrenheit\_to\_celsius(35.5)  # [1] 1.944444 |

|  |
| --- |
| year = as.integer(readline(prompt="Enter a year: "))  if((year %% 4) == 0) {  if((year %% 100) == 0) {  if((year %% 400) == 0) {  print(paste(year,"is a leap year"))  } else {  print(paste(year,"is not a leap year"))  }  } else {  print(paste(year,"is a leap year"))  }  } else { print(paste(year,"is not a leap year"))  } |