1.Write a R program to take input from the  
user (name and age) and display the values. Also print the version of R  
installation.

Code:

name <- readline(prompt = "Enter your name: ")

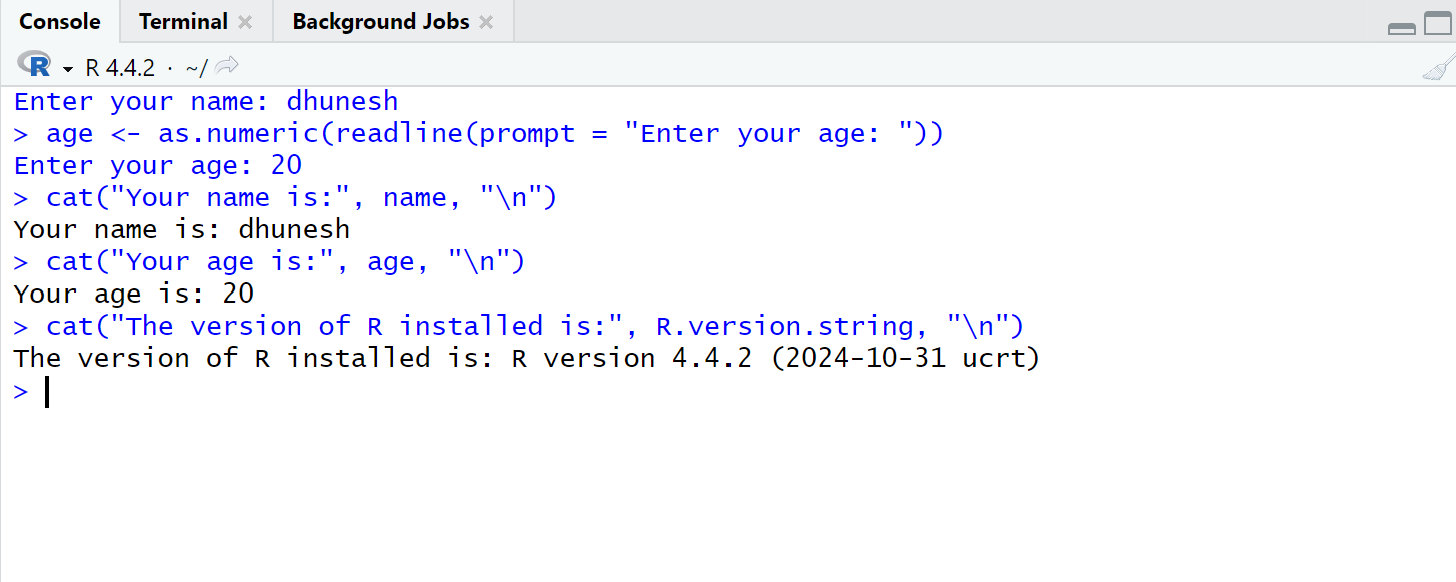
age <- as.numeric(readline(prompt = "Enter your age: "))

cat("Your name is:", name, "\n")

cat("Your age is:", age, "\n")

cat("The version of R installed is:", R.version.string, "\n")

output:



2.  Write a R program to get the details of the  
objects in memory.

Code:

name <- "Python"

num1 <- 8

num2 <- 1.5

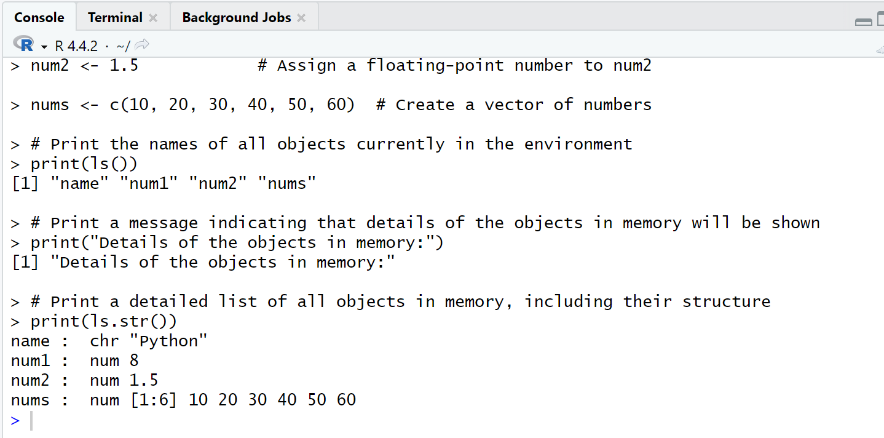
nums <- c(10, 20, 30, 40, 50, 60)

print(ls())

print("Details of the objects in memory:")

print(ls.str())

output:



3.Write a R program to create a sequence of  
numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of  
numbers from 51 to 91

Code:

sequence\_20\_to\_50 <- seq(20, 50)

print("Sequence from 20 to 50:")

print(sequence\_20\_to\_50)

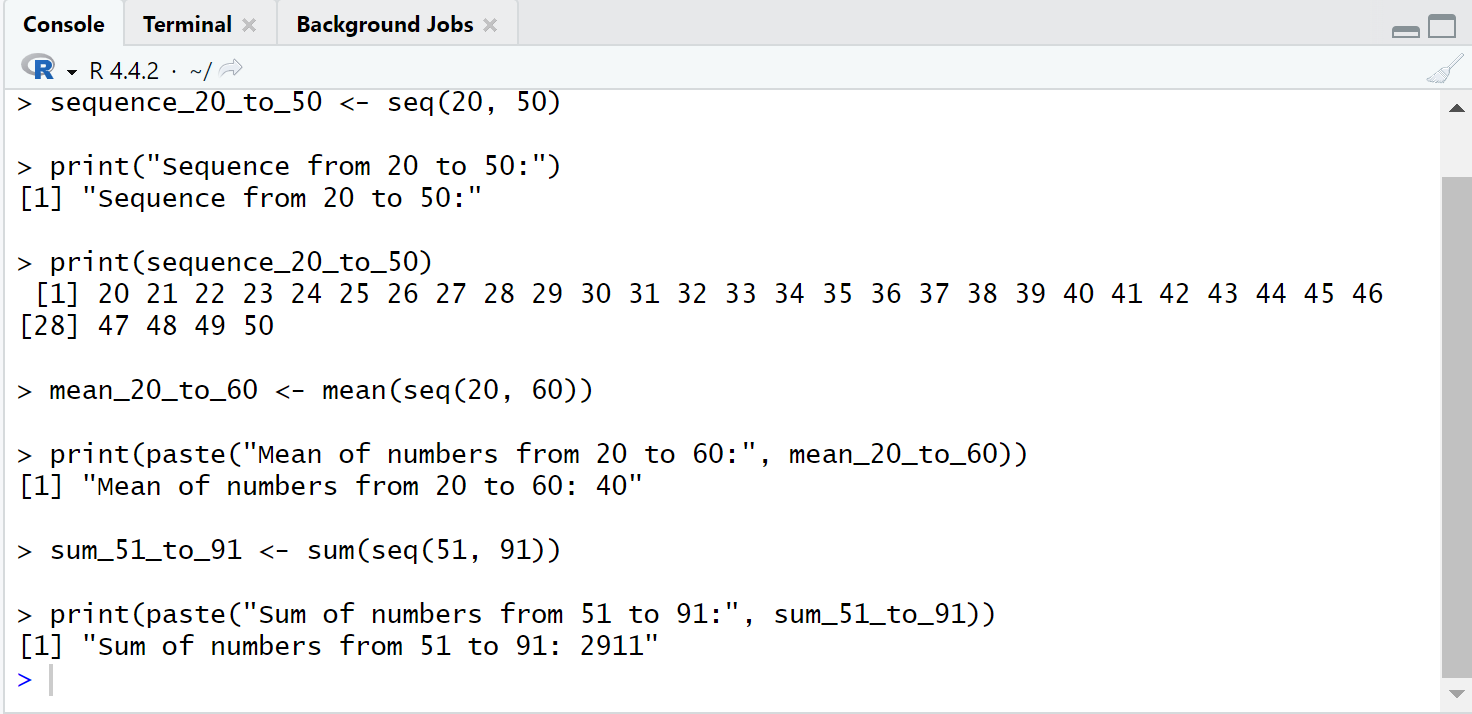
mean\_20\_to\_60 <- mean(seq(20, 60))

print(paste("Mean of numbers from 20 to 60:", mean\_20\_to\_60))

sum\_51\_to\_91 <- sum(seq(51, 91))

print(paste("Sum of numbers from 51 to 91:", sum\_51\_to\_91))̥

output:



4. Write a R program to create a vector which  
contains 10 random integer values between -50 and +50.

Code:

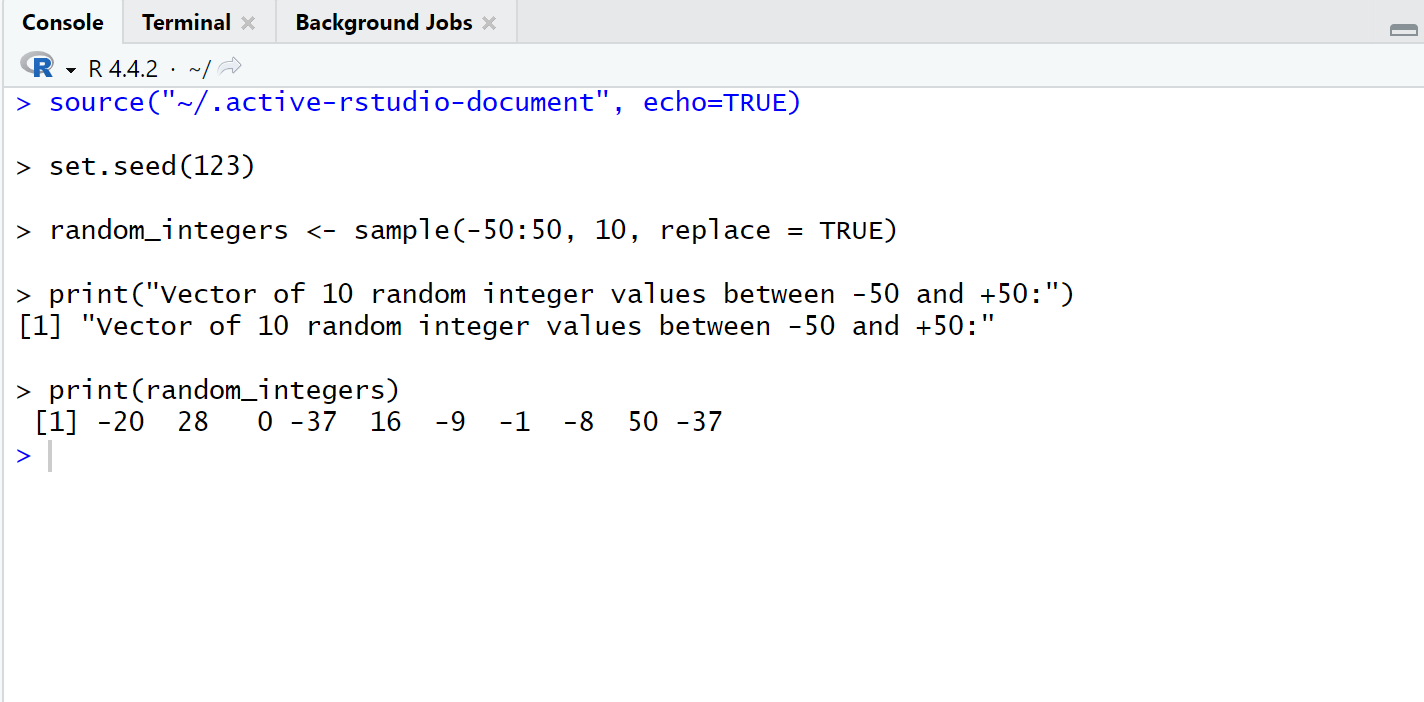
set.seed(123)

random\_integers <- sample(-50:50, 10, replace = TRUE)

print("Vector of 10 random integer values between -50 and +50:")

print(random\_integers)̥

output:



5. Write a R program to get the first 10  
Fibonacci numbers.

Code:

fibonacci\_numbers <- function(n) {

fib\_sequence <- numeric(n)

fib\_sequence[1] <- 0

fib\_sequence[2] <- 1

for (i in 3:n) {

fib\_sequence[i] <- fib\_sequence[i - 1] + fib\_sequence[i - 2]

}

return(fib\_sequence)

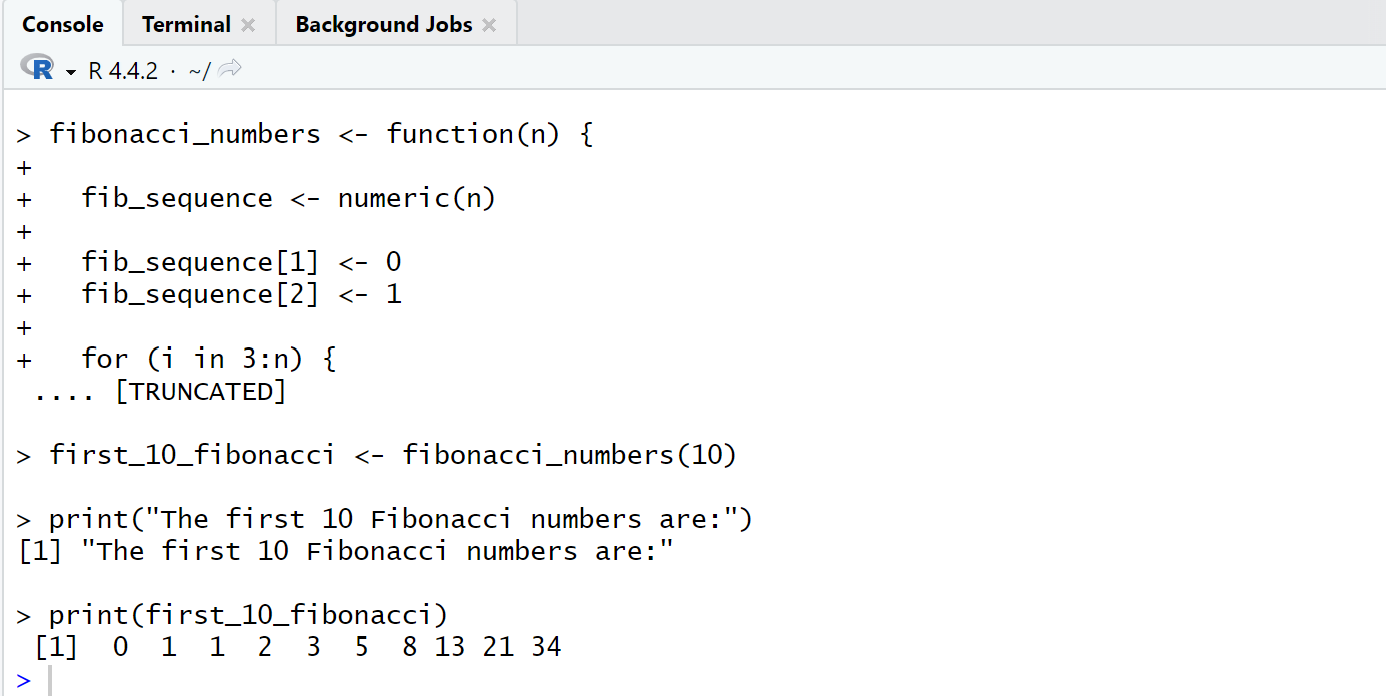
}

first\_10\_fibonacci <- fibonacci\_numbers(10)

print("The first 10 Fibonacci numbers are:")

print(first\_10\_fibonacci)̥

output:



6. Write a R program to get all prime numbers  
up to a given number (based on the sieve of Eratosthenes)

Code:

sieve\_of\_eratosthenes <- function(n) {

is\_prime <- rep(TRUE, n + 1)

is\_prime[1] <- FALSE

for (i in 2:sqrt(n)) {

if (is\_prime[i]) {

for (j in seq(i^2, n, by = i)) {

is\_prime[j] <- FALSE

}

}

}

return(which(is\_prime))

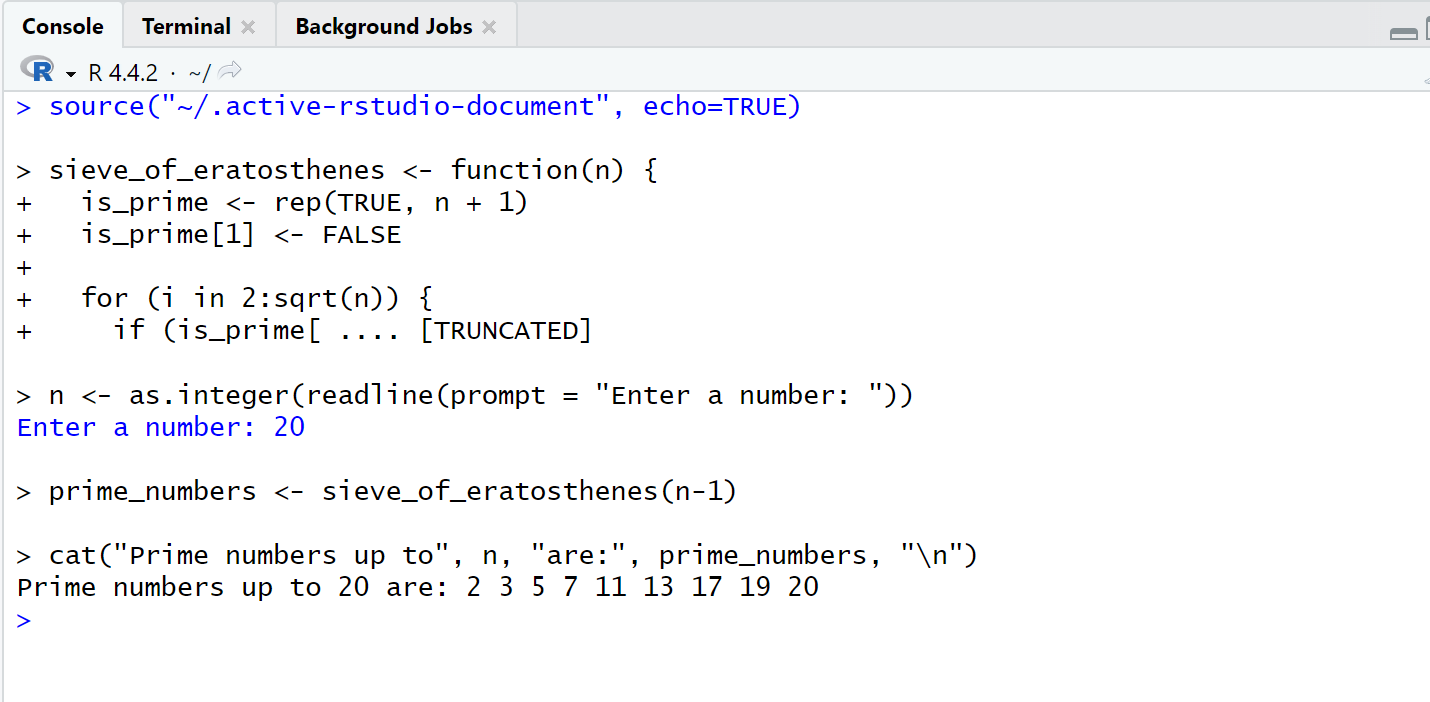
}

n <- as.integer(readline(prompt = "Enter a number: "))

prime\_numbers <- sieve\_of\_eratosthenes(n-1)

cat("Prime numbers up to", n, "are:", prime\_numbers, "\n")̥

output:



7. Write a R program to print the numbers from  
1 to 100 and print "Fizz" for multiples of 3, print "Buzz"  
for multiples of 5, and print "FizzBuzz" for multiples of both.

Code:

for (i in 1:100) {

if (i %% 3 == 0 && i %% 5 == 0) {

print("FizzBuzz")

}

else if (i %% 3 == 0) {

print("Fizz")

}

else if (i %% 5 == 0) {

print("Buzz")

}

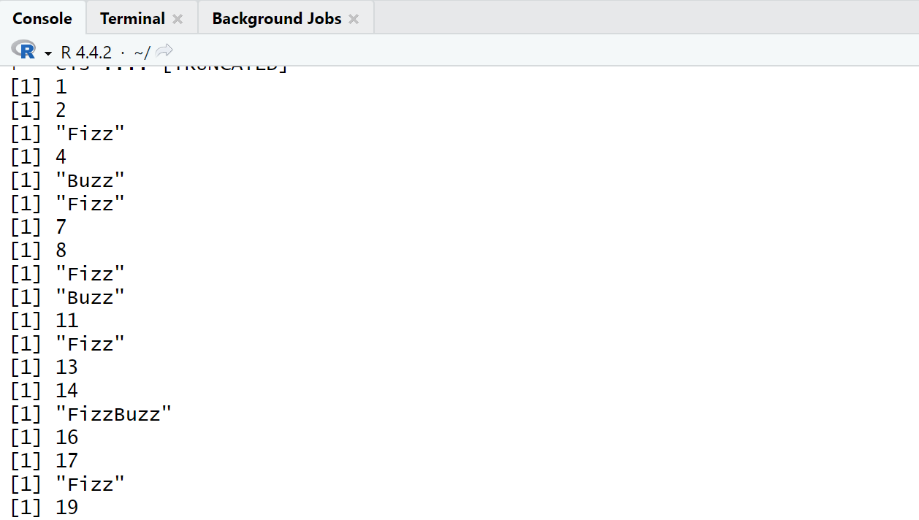
else {

print(i)

}

}

Output:



8. Write a R program to extract first 10 English  
letters in lower case and last 10 letters in upper case and extract letters  
between 22nd to 24th letters in upper case.

Code:

letters\_lowercase <- letters[1:10]

letters\_uppercase <- toupper(letters[17:26])

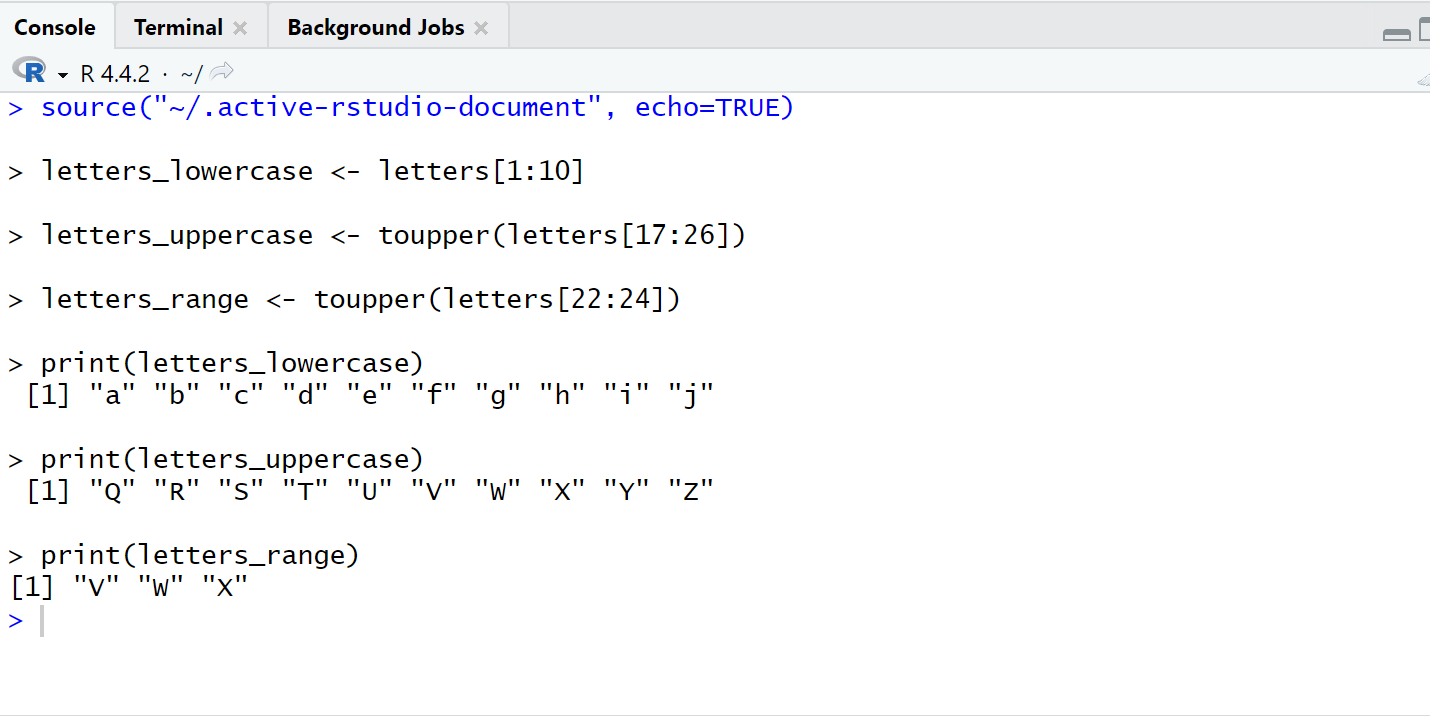
letters\_range <- toupper(letters[22:24])

print(letters\_lowercase)

print(letters\_uppercase)

print(letters\_range)̥

output:



9. Write a R program to find the factors of a  
given number

Code:

find\_factors <- function(num) {

factors <- c()

for (i in 1:num) {

if (num %% i == 0) {

factors <- c(factors, i)

}

}

return(factors)

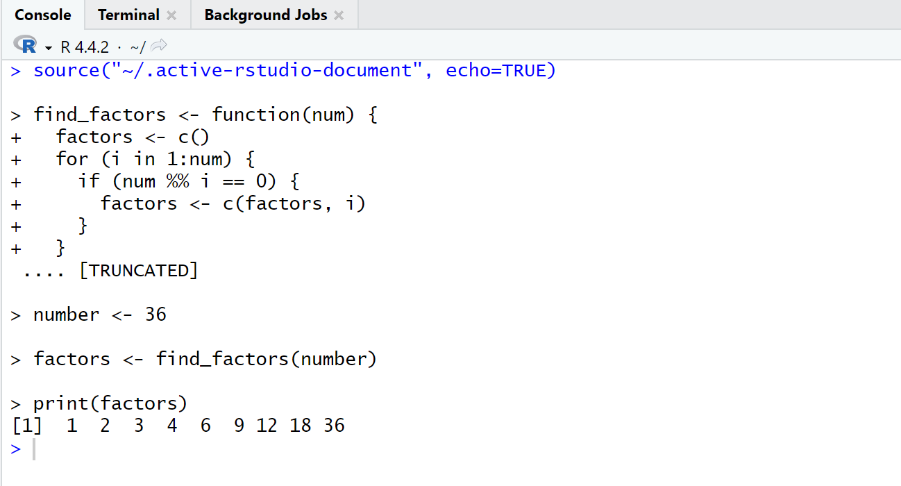
}

number <- 36

factors <- find\_factors(number)

print(factors)̥

output:



10.  Write a R program to find the maximum and the  
minimum value of a given vector

Code:

find\_max\_min <- function(vec) {

max\_value <- max(vec)

min\_value <- min(vec)

return(list(Max = max\_value, Min = min\_value))

}

vector <- c(12, 45, 7, 89, 23, 56)

result <- find\_max\_min(vector)

print(result)̥

output:

