

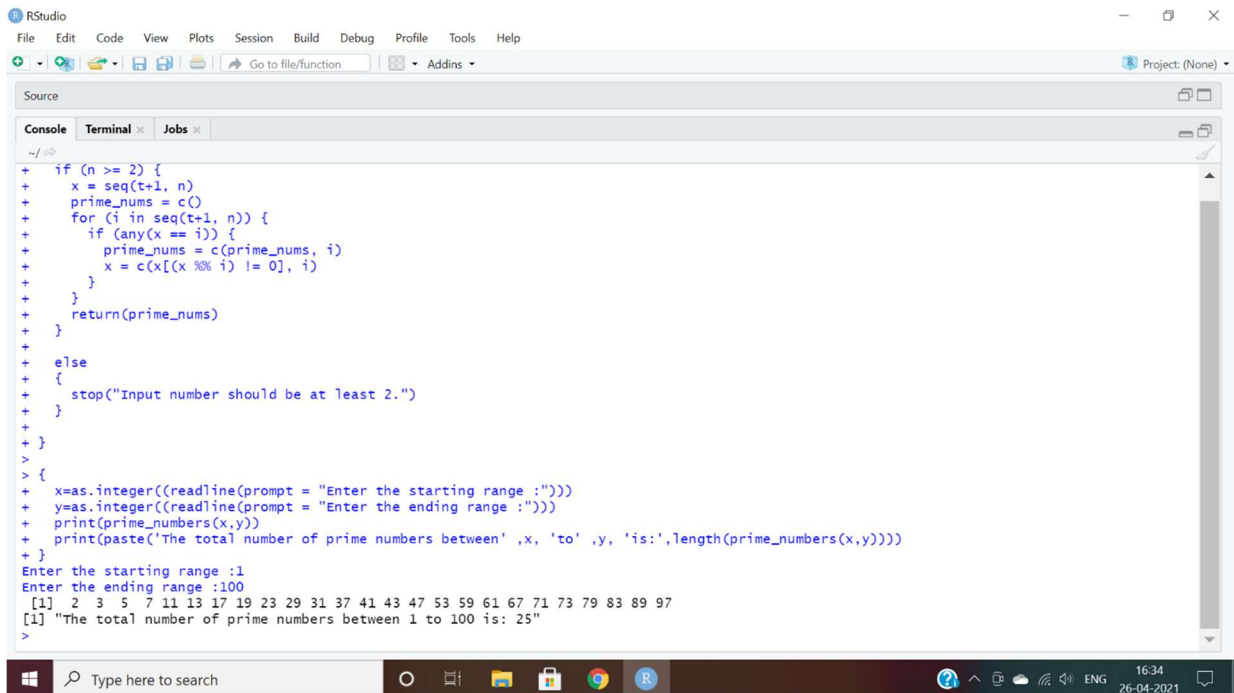
Lab Test #1 (Based on R)

Q1. Write a program in R to find prime number within a range. Input number for starting range: 1 Input number for ending range: 100 The prime numbers between 1 and 100 are: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 The total number of prime numbers between 1 to 100 is: 25

Answer:

```
prime_numbers <- function(t,n) {  
  if (n >= 2) {  
    x = seq(t+1, n)  
    prime_nums = c()  
    for (i in seq(t+1, n)) {  
      if (any(x == i)) {  
        prime_nums = c(prime_nums, i)  
        x = c(x[(x %% i) != 0], i)  
      }  
    }  
    return(prime_nums)  
  }  
  else  
  {  
    stop("Input number should be at least 2.")  
  }  
}  
  
{  
  x=as.integer((readline(prompt = "Enter the starting range :")))  
  y=as.integer((readline(prompt = "Enter the ending range :")))  
  print(prime_numbers(x,y))  
  
  print(paste('The total number of prime numbers between' ,x, 'to' ,y,  
'is:',length(prime_numbers(x,y))))  
}
```

OUTPUT



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

Source
Console Terminal Jobs

~/
+ if (n >= 2) {
+   x = seq(t+1, n)
+   prime_nums = c()
+   for (i in seq(t+1, n)) {
+     if (any(x == i)) {
+       prime_nums = c(prime_nums, i)
+       x = x[(x %% i) != 0], i)
+     }
+   }
+   return(prime_nums)
+ }
+ else
+ {
+   stop("Input number should be at least 2.")
+ }
+ }
+ }
>
> {
+ x=as.integer(readline(prompt = "Enter the starting range :"))
+ y=as.integer(readline(prompt = "Enter the ending range :"))
+ print(prime_numbers(x,y))
+ print(paste("The total number of prime numbers between ' ,x, 'to' ,y, 'is:',length(prime_numbers(x,y)))")
+ }
Enter the starting range :1
Enter the ending range :100
[1] 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
[1] "The total number of prime numbers between 1 to 100 is: 25"
```

Q2. Create the 10*10 identity matrix and add it to another matrix of same dimension. Print the result of multiplication, subtraction in R.

Answer:

```
x<-diag(10)
```

```
x
```

```
y<-matrix(1:100, nrow = 10,byrow=TRUE)
```

```
y
```

```
add<-x+y
```

```
print("Addition of two matrices")
```

```
add
```

```
multiply<-x*y
```

```
print("Multiplication of two matrices")
```

multiply

sub<-y-x

print("Subtraction of two matrices")

sub

OUTPUT

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

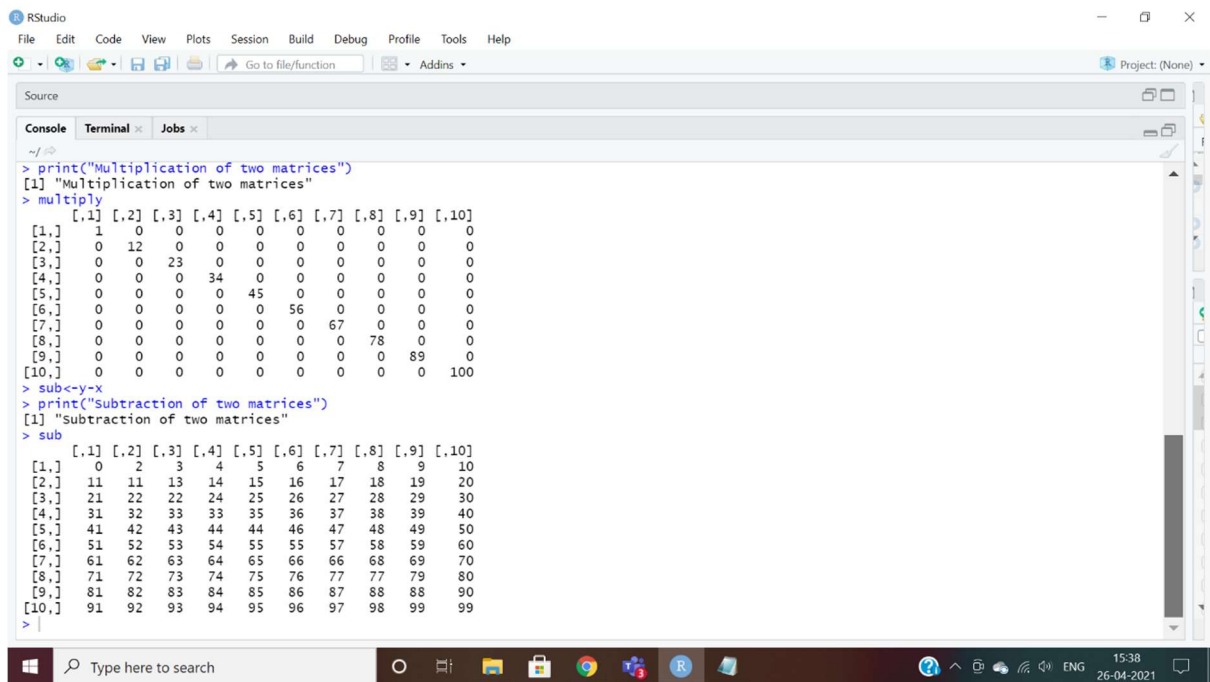
Source
Console Terminal Jobs

~/
> x<-diag(10)
> x
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  1    0    0    0    0    0    0    0    0    0
[2,]  0    1    0    0    0    0    0    0    0    0
[3,]  0    0    1    0    0    0    0    0    0    0
[4,]  0    0    0    1    0    0    0    0    0    0
[5,]  0    0    0    0    1    0    0    0    0    0
[6,]  0    0    0    0    0    1    0    0    0    0
[7,]  0    0    0    0    0    0    1    0    0    0
[8,]  0    0    0    0    0    0    0    1    0    0
[9,]  0    0    0    0    0    0    0    0    1    0
[10,] 0    0    0    0    0    0    0    0    0    1
> y<-matrix(1:100, nrow = 10,byrow=TRUE)
> y
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  1    2    3    4    5    6    7    8    9   10
[2,] 11   12   13   14   15   16   17   18   19   20
[3,] 21   22   23   24   25   26   27   28   29   30
[4,] 31   32   33   34   35   36   37   38   39   40
[5,] 41   42   43   44   45   46   47   48   49   50
[6,] 51   52   53   54   55   56   57   58   59   60
[7,] 61   62   63   64   65   66   67   68   69   70
[8,] 71   72   73   74   75   76   77   78   79   80
[9,] 81   82   83   84   85   86   87   88   89   90
[10,] 91   92   93   94   95   96   97   98   99   100
> add<-x+y
> print("Addition of two matrices")
[1] "Addition of two matrices"
> add
```

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

Source
Console Terminal Jobs

~/
> print("Addition of two matrices")
[1] "Addition of two matrices"
> add
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  2    2    3    4    5    6    7    8    9   10
[2,] 11   13   13   14   15   16   17   18   19   20
[3,] 21   22   24   24   25   26   27   28   29   30
[4,] 31   32   33   35   35   36   37   38   39   40
[5,] 41   42   43   44   46   46   47   48   49   50
[6,] 51   52   53   54   55   57   57   58   59   60
[7,] 61   62   63   64   65   66   68   68   69   70
[8,] 71   72   73   74   75   76   77   79   79   80
[9,] 81   82   83   84   85   86   87   88   90   90
[10,] 91   92   93   94   95   96   97   98   99   101
> multiply<-x*y
> print("Multiplication of two matrices")
[1] "Multiplication of two matrices"
> multiply
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  1    0    0    0    0    0    0    0    0    0
[2,]  0   12    0    0    0    0    0    0    0    0
[3,]  0    0   23    0    0    0    0    0    0    0
[4,]  0    0    0   34    0    0    0    0    0    0
[5,]  0    0    0    0   45    0    0    0    0    0
[6,]  0    0    0    0    0   56    0    0    0    0
[7,]  0    0    0    0    0    0   67    0    0    0
[8,]  0    0    0    0    0    0    0   78    0    0
[9,]  0    0    0    0    0    0    0    0   89    0
[10,] 0    0    0    0    0    0    0    0    0   100
> sub<-y-x
```

The screenshot shows the RStudio interface with the Console pane active. It displays the output of two R commands. The first command, 'multiply', prints a 10x10 matrix of zeros. The second command, 'sub', prints a 10x10 matrix of integers from 0 to 99. The RStudio window includes a menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help), a toolbar, and a source editor area.

```
> print("Multiplication of two matrices")
[1] "Multiplication of two matrices"
> multiply
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  0  0  0  0  0  0  0  0  0  0
[2,]  0 12  0  0  0  0  0  0  0  0
[3,]  0  0 23  0  0  0  0  0  0  0
[4,]  0  0  0 34  0  0  0  0  0  0
[5,]  0  0  0  0 45  0  0  0  0  0
[6,]  0  0  0  0  0 56  0  0  0  0
[7,]  0  0  0  0  0  0 67  0  0  0
[8,]  0  0  0  0  0  0  0 78  0  0
[9,]  0  0  0  0  0  0  0  0 89  0
[10,] 0  0  0  0  0  0  0  0  0 100
> sub<-y-x
> print("Subtraction of two matrices")
[1] "Subtraction of two matrices"
> sub
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  0  2  3  4  5  6  7  8  9 10
[2,] 11 11 13 14 15 16 17 18 19 20
[3,] 21 22 22 24 25 26 27 28 29 30
[4,] 31 32 33 33 35 36 37 38 39 40
[5,] 41 42 43 44 44 46 47 48 49 50
[6,] 51 52 53 54 55 55 57 58 59 60
[7,] 61 62 63 64 65 66 66 68 69 70
[8,] 71 72 73 74 75 76 77 77 79 80
[9,] 81 82 83 84 85 86 87 88 88 90
[10,] 91 92 93 94 95 96 97 98 99 99
```

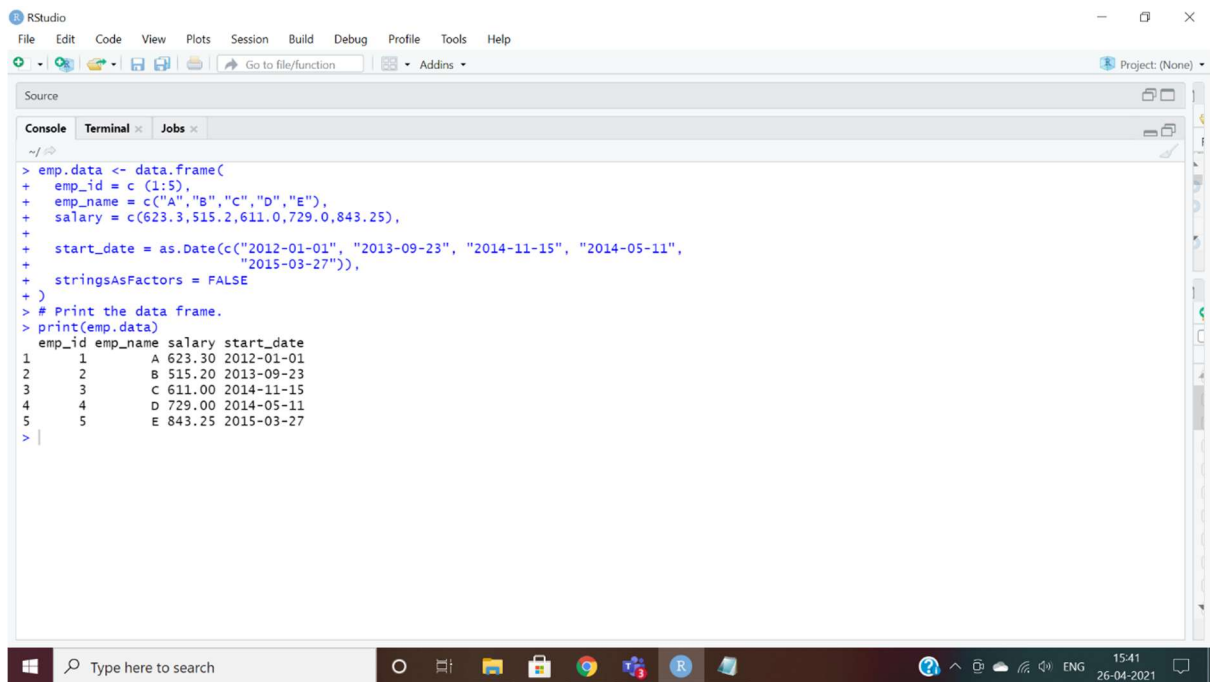
Q3. Write R script to create data frame “Emp” with fields of emp_id, emp_name, salary, and start_date: emp_id = c(1:5) emp_name = c("A","B","C","D","E") salary = c(623.3,515.2,611.0,729.0,843.25) start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05- 11", "2015-03-27"))),

i)Print the structure of data frame.

Answer:

```
emp.data <- data.frame(
  emp_id = c (1:5),
  emp_name = c("A","B","C","D","E"),
  salary = c(623.3,515.2,611.0,729.0,843.25),
  start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
    "2015-03-27")),
  stringsAsFactors = FALSE
)
# Print the data frame.
print(emp.data)
```

OUTPUT



```
# RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

Source
Console Terminal Jobs

~/f
> emp.data <- data.frame(
+   emp_id = c(1:5),
+   emp_name = c("A","B","C","D","E"),
+   salary = c(623.3,515.2,611.0,729.0,843.25),
+   start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
+   "2015-03-27")),
+   stringsAsFactors = FALSE
+ )
> # Print the data frame.
> print(emp.data)
  emp_id emp_name salary start_date
1      1      A 623.30 2012-01-01
2      2      B 515.20 2013-09-23
3      3      C 611.00 2014-11-15
4      4      D 729.00 2014-05-11
5      5      E 843.25 2015-03-27
> |
```

ii) Extract emp_name and salary columns.

Answer:

```
emp.data <- data.frame(

  emp_id = c(1:5),

  emp_name = c("A","B","C","D","E"),

  salary = c(623.3,515.2,611.0,729.0,843.25),

  start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
                          "2015-03-27")),

  stringsAsFactors = FALSE

)

# Extract Specific columns.

result <- data.frame(emp.data$emp_name,emp.data$salary)

print(result)
```

OUTPUT

```
> emp.data <- data.frame(  
+   emp_id = c(1:5),  
+   emp_name = c("A", "B", "C", "D", "E"),  
+   salary = c(623.3, 515.2, 611.0, 729.0, 843.25),  
+  
+   start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",  
+   "2015-03-27")),  
+   stringsAsFactors = FALSE  
+ )  
> # Extract Specific columns.  
> result <- data.frame(emp.data$emp_name, emp.data$salary)  
> print(result)  
emp.data.emp_name emp.data.salary  
1                A           623.30  
2                B           515.20  
3                C           611.00  
4                D           729.00  
5                E           843.25  
> |
```

Q4. In the following real-world data (airquality), we have considered few observations (Obs) and attributes.

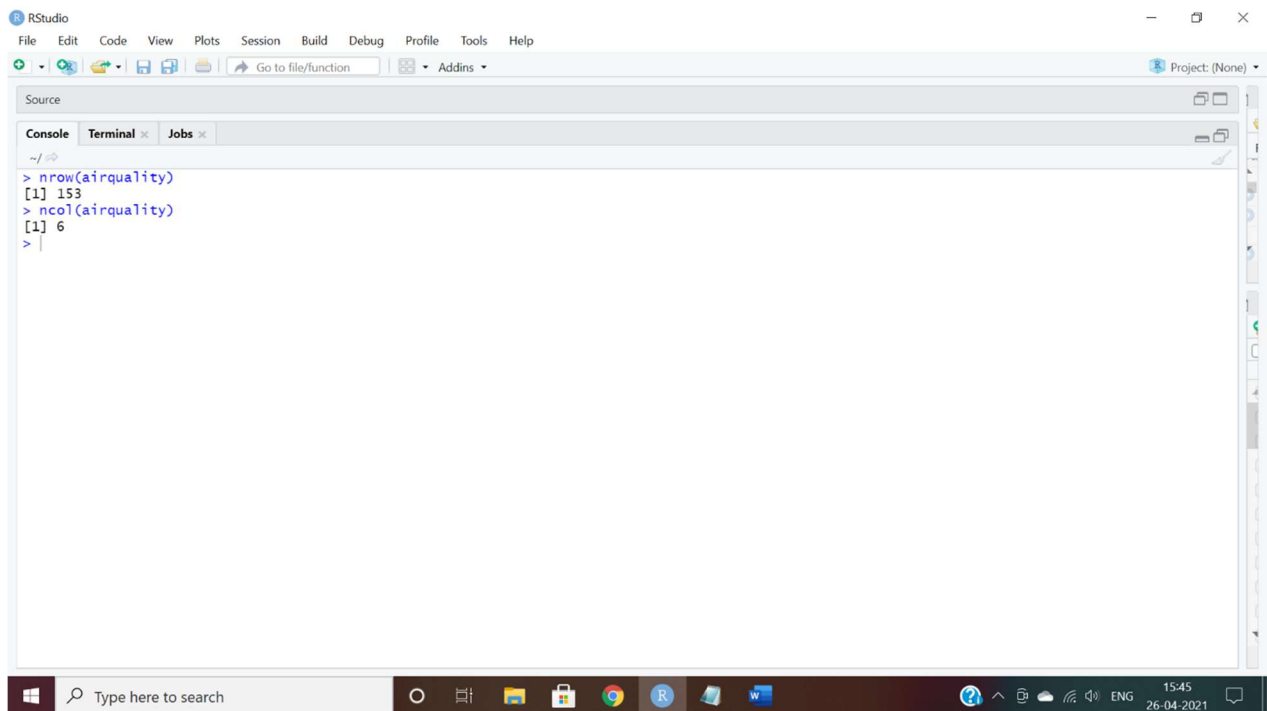
i) Write R syntax to print number of columns and rows in the data

Answer:

nrow(airquality)

ncol(airquality)

OUTPUT



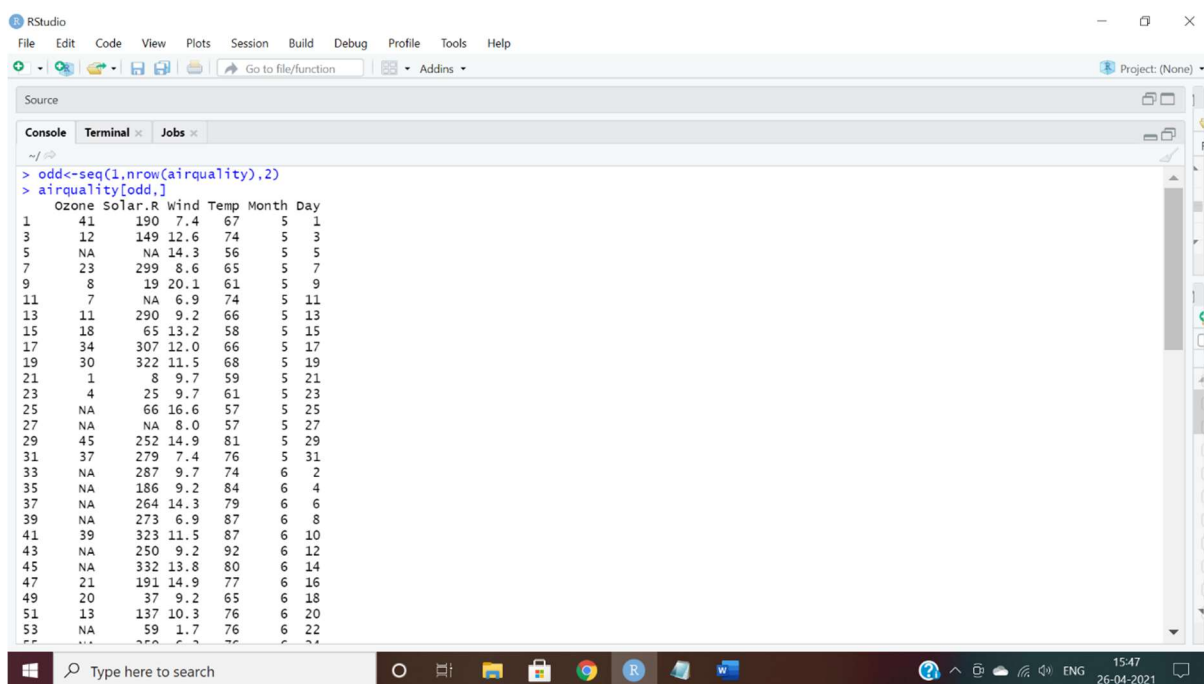
ii) Write R syntax to print all odd indexed rows for given data.

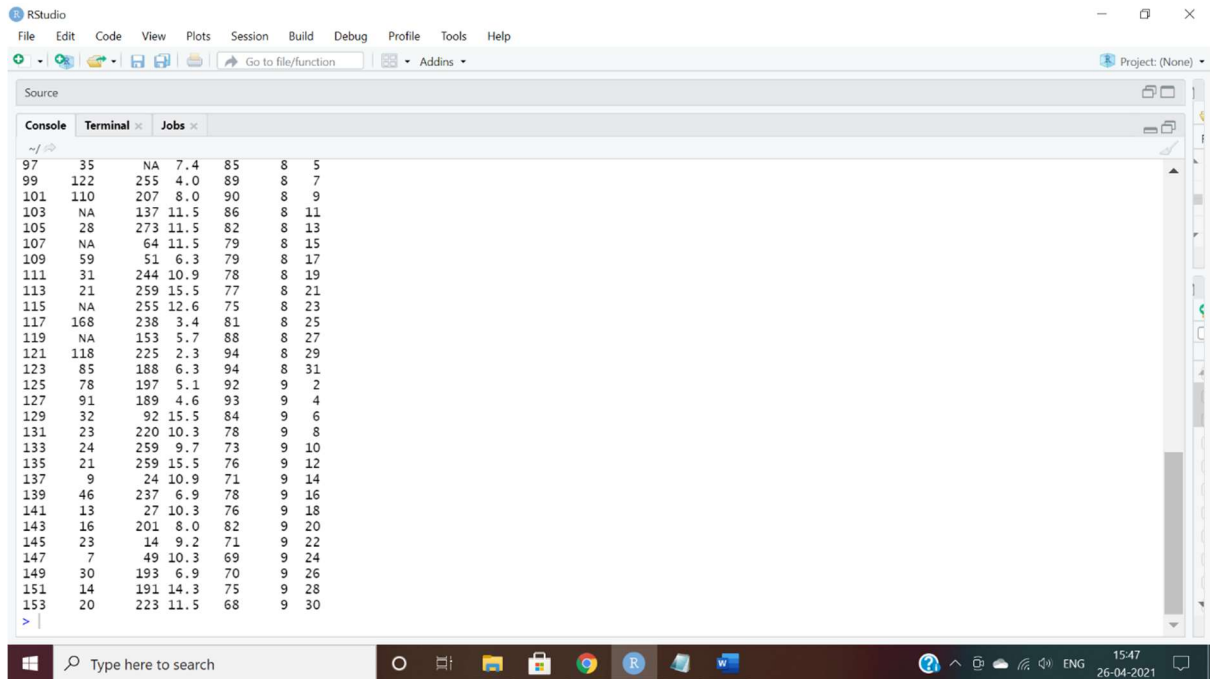
Answer:

```
odd<-seq(1,nrow(airquality),2)
```

```
airquality[odd,]
```

OUTPUT



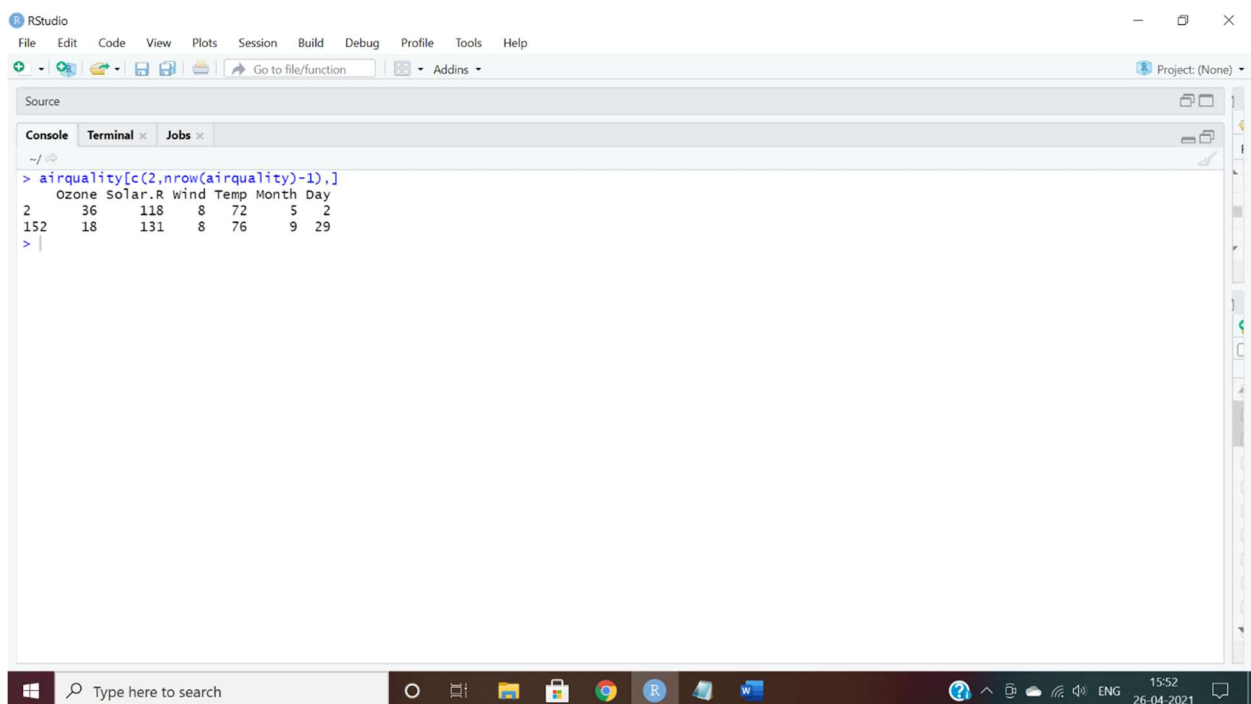


iii) Write R syntax to print second row and second last row for given data.

Answer:

`airquality[c(2,nrow(airquality)-1),]`

OUTPUT



iv) Write R syntax to print even and odd index column for given data.

Answer:

even

```
even<- seq(2,ncol(airquality),2)
```

```
airquality[,even]
```

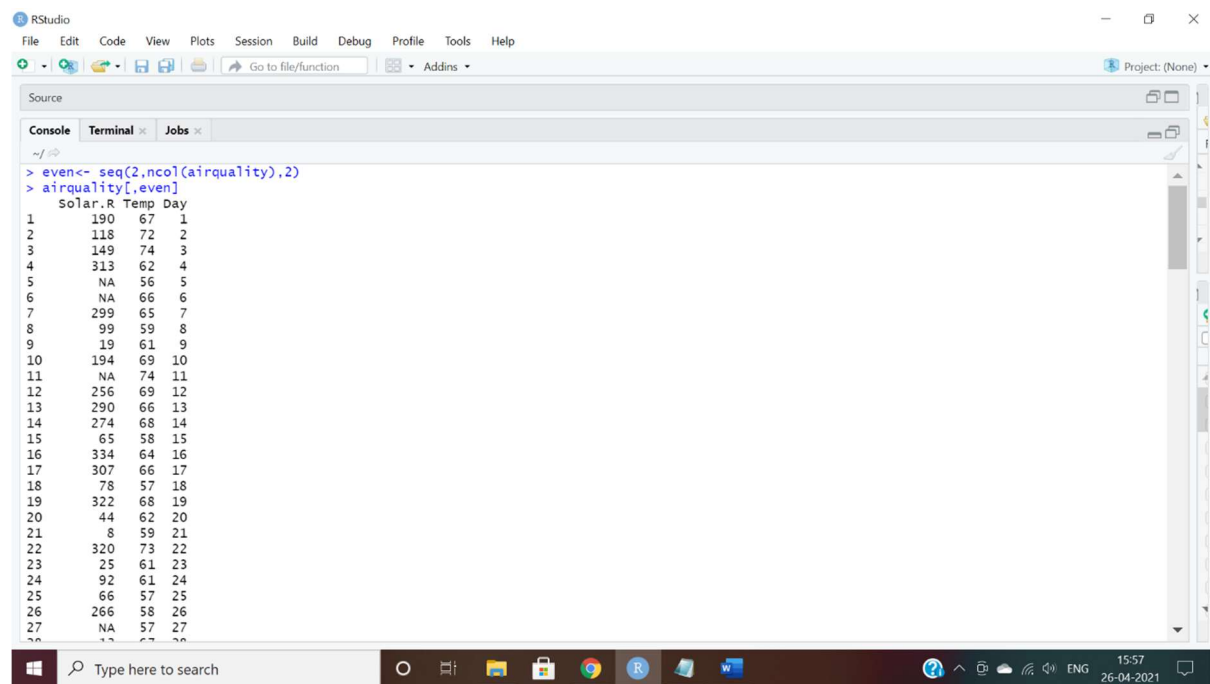
odd

```
odd<- seq(1,ncol(airquality),2)
```

```
airquality[,odd]
```

OUTPUT

EVEN Columns



The screenshot shows the RStudio interface with the console pane active. The code entered is: `even<- seq(2,ncol(airquality),2)` followed by `> airquality[,even]`. The output displays the first 27 rows of the 'airquality' dataset, specifically columns 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, and 26. The columns are labeled 'Solar.R', 'R', 'Temp', and 'Day' respectively. The data is as follows:

	Solar.R	R	Temp	Day
1	190	67	1	
2	118	72	2	
3	149	74	3	
4	313	62	4	
5	NA	56	5	
6	NA	66	6	
7	299	65	7	
8	99	59	8	
9	19	61	9	
10	194	69	10	
11	NA	74	11	
12	256	69	12	
13	290	66	13	
14	274	68	14	
15	65	58	15	
16	334	64	16	
17	307	66	17	
18	78	57	18	
19	322	68	19	
20	44	62	20	
21	8	59	21	
22	320	73	22	
23	25	61	23	
24	92	61	24	
25	66	57	25	
26	266	58	26	
27	NA	57	27	

ODD Columns

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins Project: (None)

Source
Console Terminal Jobs
~/
> odd<- seq(1,ncol(airquality),2)
> airquality[,odd]
  Ozone wind Month
1    41  7.4     5
2    36  8.0     5
3    12 12.6     5
4    18 11.5     5
5     NA 14.3     5
6    28 14.9     5
7    23  8.6     5
8    19 13.8     5
9     8 20.1     5
10   NA  8.6     5
11    7  6.9     5
12   16  9.7     5
13   11  9.2     5
14   14 10.9     5
15   18 13.2     5
16   14 11.5     5
17   34 12.0     5
18    6 18.4     5
19   30 11.5     5
20   11  9.7     5
21    1  9.7     5
22   11 16.6     5
23    4  9.7     5
24   32 12.0     5
25   NA 16.6     5
26   NA 14.9     5
27   NA  8.0     5
```

Q5. Consider the data set “airquality” (mentioned in Q. 4) in R. · Write a command to draw a boxplot at the column as ozone.

Answer:

```
boxplot(airquality$Ozone)
```

OUTPUT

