

COVID-19 VISUALIZATION AND PREDICTION

DATASET USED:

"C:\Users\nivet\Downloads\corona_tested_individuals_ver_006.english.csv"

Import the dataset:

```
>library(readr)
```

```
>my_data1<-read.csv("C:/Users/nivet/Downloads/corona_tested_individuals_ver_006.english.csv")
```

Now,

Exploratory data analysis:

```
>str(my_data1)
```

This gives the no. of rows and no. of columns in the dataset and the column names .

'data.frame': 278848 obs. of 10 variables:

\$ test_date : chr "30-04-2020" "30-04-2020" "30-04-2020" "30-04-2020" ...

\$ cough : chr "0" "1" "0" "1" ...

\$ fever : chr "0" "0" "1" "0" ...

\$ sore_throat : chr "0" "0" "0" "0" ...

\$ shortness_of_breath: chr "0" "0" "0" "0" ...

\$ head_ache : chr "0" "0" "0" "0" ...

\$ age_60_and_above : chr "None" "None" "None" "None" ...

\$ gender : chr "female" "female" "male" "female" ...

\$ test_indication : chr "Other" "Other" "Other" "Other" ...

\$ corona_result : chr "negative" "negative" "negative" "negative" ...

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

	test_date	cough	fever	sore_throat	shortness_of_breath	head_ache	age_60_and_above	gender	test_indication	corona_result
1	30-04-2020	0	0	0	0	0	None	female	Other	negative
2	30-04-2020	1	0	0	0	0	None	female	Other	negative
3	30-04-2020	0	1	0	0	0	None	male	Other	negative
4	30-04-2020	1	0	0	0	0	None	female	Other	negative
5	30-04-2020	1	0	0	0	0	None	male	Other	negative
6	30-04-2020	1	0	0	0	0	None	female	Other	negative
7	30-04-2020	1	1	0	0	0	None	male	Abroad	negative
8	30-04-2020	0	0	0	0	0	None	female	Other	negative
9	30-04-2020	0	0	0	0	0	None	male	Other	negative
10	30-04-2020	0	0	0	0	0	None	male	Contact with confirmed	negative
11	30-04-2020	1	1	0	0	0	None	female	Other	negative
12	30-04-2020	0	0	0	0	0	None	female	Other	negative
13	30-04-2020	0	0	0	0	0	None	female	Other	negative
14	30-04-2020	0	0	0	0	0	None	female	Other	negative
15	30-04-2020	0	0	0	0	0	None	male	Other	negative
16	30-04-2020	1	0	0	0	0	None	male	Other	negative

Showing 1 to 16 of 278,848 entries, 10 total columns

Console Terminal Jobs

```
R 4.1.1 ~\RData  
[Workspace loaded from ~\RData]  
> library(readr)  
> my_data1<-read_csv("C:/Users/nivet/Downloads/corona_tested_individuals_ver_006.english.csv")  
> View(my_data1)  
> str(my_data1)  
'data.frame': 278848 obs. of 10 variables:  
 $ test_date      : chr "30-04-2020" "30-04-2020" "30-04-2020" "30-04-2020" ...  
 $ cough          : chr "0" "1" "0" "1" ...  
 $ fever          : chr "0" "0" "1" "0" ...  
 $ sore_throat    : chr "0" "0" "0" "0" ...  
 $ shortness_of_breath: chr "0" "0" "0" "0" ...  
 $ head_ache      : chr "0" "0" "0" "0" ...  
 $ age_60_and_above: chr "None" "None" "None" "None" ...  
 $ gender         : chr "female" "female" "male" "female" ...  
 $ test_indication : chr "Other" "Other" "Other" "Other" ...  
 $ corona_result  : chr "negative" "negative" "negative" "negative" ...  
> View(my_data1)  
>
```

Environment History Connections Tutorial

Global Environment

- iris2 List of 13
- italy 694 obs. of 7 variab...
- km List of 9
- Linearre... 1 obs. of 4 variables
- lm.iris2 List of 13
- LungCapd... 725 obs. of 6 variab...
- LungCapd... 725 obs. of 1 variab...
- marketing 200 obs. of 4 variab...
- model List of 30
- model1 List of 12
- my_data1 278848 obs. of 10 va...

Files Plots Packages Help Viewer

Zoom Export

ENG IN 12:10 11-01-2022

► We can see that our dataset contains 278848 rows of 10 columns.

► Now,
>View(my_data1)

► REMOVING UNWANTED COLUMNS:

► `my_data1<-subset(my_data1,select=-age_60_and_above)#my_data1$age_60_and_above<-NULL`

► `my_data1<-subset(my_data1,select=-gender)`

► `my_data1<-subset(my_data1,select=-test_indication)`

► Now,

`>str(my_data1)`

'data.frame': 278848 obs. of 7 variables:

\$ test_date : chr "30-04-2020" "30-04-2020" "30-04-2020" "30-04-2020" ...

\$ cough : chr "0" "1" "0" "1" ...

\$ fever : chr "0" "0" "1" "0" ...

\$ sore_throat : chr "0" "0" "0" "0" ...

\$ shortness_of_breath: chr "0" "0" "0" "0" ...

\$ head_ache : chr "0" "0" "0" "0" ...

\$ corona_result : chr "negative" "negative" "negative" "negative" ...

We can see that, the columns "age_60_and_above", "gender", and "test_indication" are removed.

- ▶ We can see this:
- ▶ >View(my_data1)

The screenshot displays the RStudio interface. The main editor window shows a data frame with 7 columns: test_date, cough, fever, sore_throat, shortness_of_breath, head_ache, and corona_result. The first 16 rows are visible, showing dates from 30-04-2020 and binary values for the symptoms, with 'corona_result' being 'negative'. The console window at the bottom shows the following R code and output:

```
R 4.1.1: >
Warning message:
package 'dplyr' was built under R version 4.1.2
> #removing unwanted columns
> my_data1<-subset(my_data1,select=age_60_and_above)#my_data1$age_60_and_above<-NULL
> my_data1<-subset(my_data1,select=gender)
> my_data1<-subset(my_data1,select=test_indication)
> str(my_data1)
'data.frame': 278848 obs. of 7 variables:
 $ test_date      : chr  "30-04-2020" "30-04-2020" "30-04-2020" ...
 $ cough          : chr  "0" "1" "0" "1" ...
 $ fever         : chr  "0" "0" "1" "0" ...
 $ sore_throat    : chr  "0" "0" "0" "0" ...
 $ shortness_of_breath: chr  "0" "0" "0" "0" ...
 $ head_ache      : chr  "0" "0" "0" "0" ...
 $ corona_result  : chr  "negative" "negative" "negative" "negative" ...
> View(my_data1)
>
```

On the right side, the Environment pane lists several objects: iris2 (List of 13), Italy (694 obs. of 7 variab...), km (List of 9), Linearre (1 obs. of 4 variables), lm.iris2 (List of 13), LungCapd (725 obs. of 6 variab...), LungCapd (725 obs. of 1 variab...), marketing (200 obs. of 4 variab...), model (List of 30), model1 (List of 12), and my_data1 (278848 obs. of 7 var...).

Click to add text

>summary(my_data1)#This gives the column names and their length and their type.

- ▶ test_date cough fever sore_throat shortness_of_breath head_ache corona_result
- ▶ Length:278848 Length:278848 Length:278848 Length:278848 Length:278848 Length:278848 Length:278848
- ▶ Class :character Class :character Class :character Class :character Class :character Class :character Class :character
- ▶ Mode :character Mode :character Mode :character Mode :character Mode :character Mode :character Mode :character
- ▶ We can see that all columns are of character type. So,let us convert this into numeric type to perform statistical analysis using ifelse function.

```
>my_data1$fever<-ifelse(my_data1$fever=="1",1,0)
```

```
>my_data1$cough<-ifelse(my_data1$cough=="1",1,0)
```

```
>my_data1$sore_throat<-ifelse(my_data1$sore_throat=="1",1,0)
```

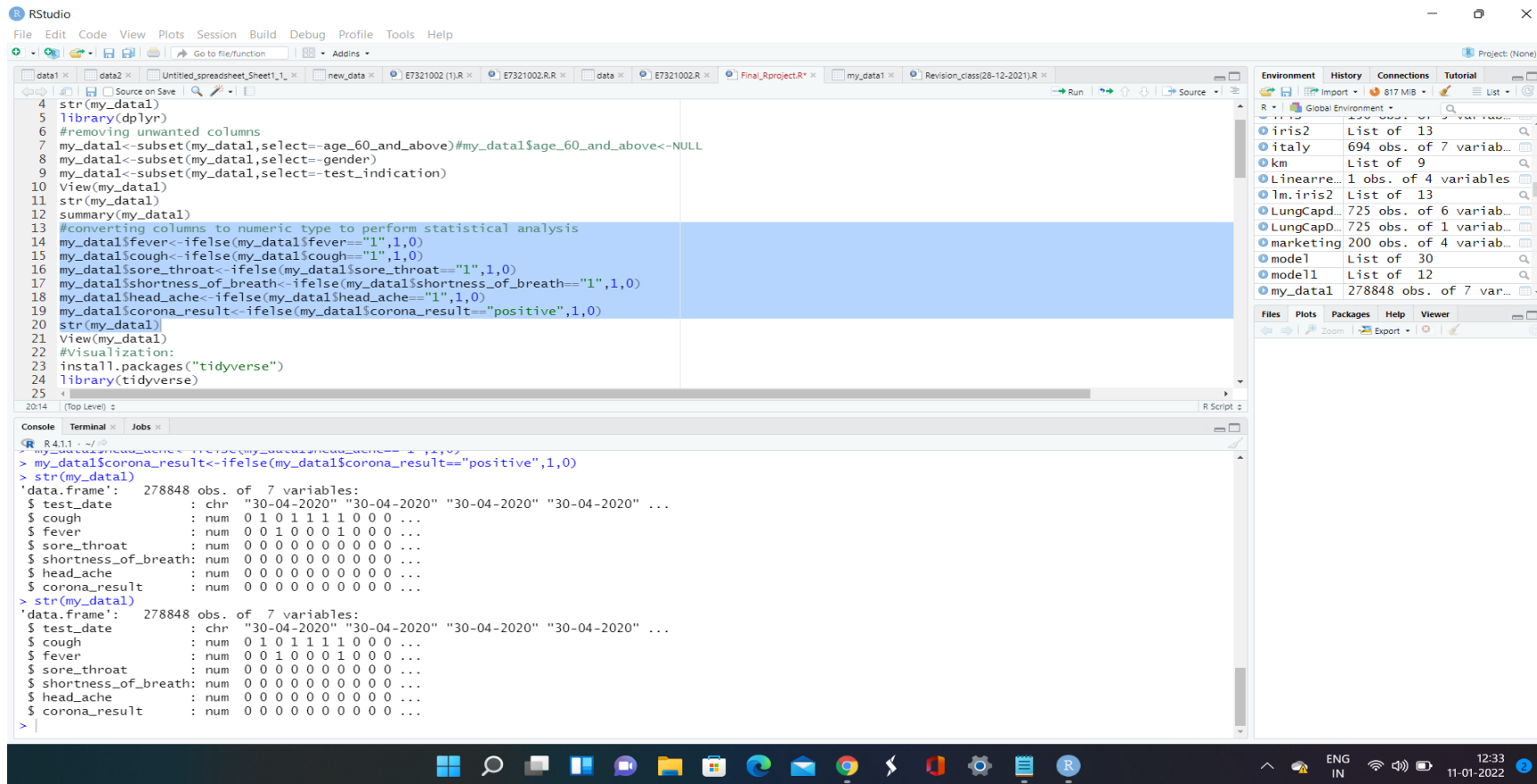
```
>my_data1$shortness_of_breath<-ifelse(my_data1$shortness_of_breath=="1",1,0)
```

```
>my_data1$head_ache<-ifelse(my_data1$head_ache=="1",1,0)
```

```
>my_data1$corona_result<-ifelse(my_data1$corona_result=="positive",1,0)
```

By doing this, the null values also get replaced by 0.

- ▶ Now,
- ▶ `>str(my_data1)`
- ▶ We can see that except "test_date" column, all other columns have been converted to numeric type.



The screenshot shows the RStudio interface. The script editor contains the following code:

```
4 str(my_data1)
5 library(dplyr)
6 #removing unwanted columns
7 my_data1<-subset(my_data1,select=-age_60_and_above)#my_data1$age_60_and_above<-NULL
8 my_data1<-subset(my_data1,select=-gender)
9 my_data1<-subset(my_data1,select=-test_indication)
10 View(my_data1)
11 str(my_data1)
12 summary(my_data1)
13 #converting columns to numeric type to perform statistical analysis
14 my_data1$fever<-ifelse(my_data1$fever=="1",1,0)
15 my_data1$cough<-ifelse(my_data1$cough=="1",1,0)
16 my_data1$sore_throat<-ifelse(my_data1$sore_throat=="1",1,0)
17 my_data1$shortness_of_breath<-ifelse(my_data1$shortness_of_breath=="1",1,0)
18 my_data1$head_ache<-ifelse(my_data1$head_ache=="1",1,0)
19 my_data1$corona_result<-ifelse(my_data1$corona_result=="positive",1,0)
20 str(my_data1)
21 View(my_data1)
22 #Visualization:
23 install.packages("tidyverse")
24 library(tidyverse)
25
```

The console output shows the result of `str(my_data1)` after the conversions:

```
> str(my_data1)
'data.frame': 278848 obs. of 7 variables:
 $ test_date      : chr  "30-04-2020" "30-04-2020" "30-04-2020" "30-04-2020" ...
 $ cough          : num  0 1 0 1 1 1 1 0 0 0 ...
 $ fever          : num  0 0 1 0 0 0 0 1 0 0 ...
 $ sore_throat    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ shortness_of_breath: num  0 0 0 0 0 0 0 0 0 0 ...
 $ head_ache      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ corona_result  : num  0 0 0 0 0 0 0 0 0 0 ...

> str(my_data1)
'data.frame': 278848 obs. of 7 variables:
 $ test_date      : chr  "30-04-2020" "30-04-2020" "30-04-2020" "30-04-2020" ...
 $ cough          : num  0 1 0 1 1 1 1 0 0 0 ...
 $ fever          : num  0 0 1 0 0 0 0 1 0 0 ...
 $ sore_throat    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ shortness_of_breath: num  0 0 0 0 0 0 0 0 0 0 ...
 $ head_ache      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ corona_result  : num  0 0 0 0 0 0 0 0 0 0 ...
```

The Environment pane on the right shows the following objects:

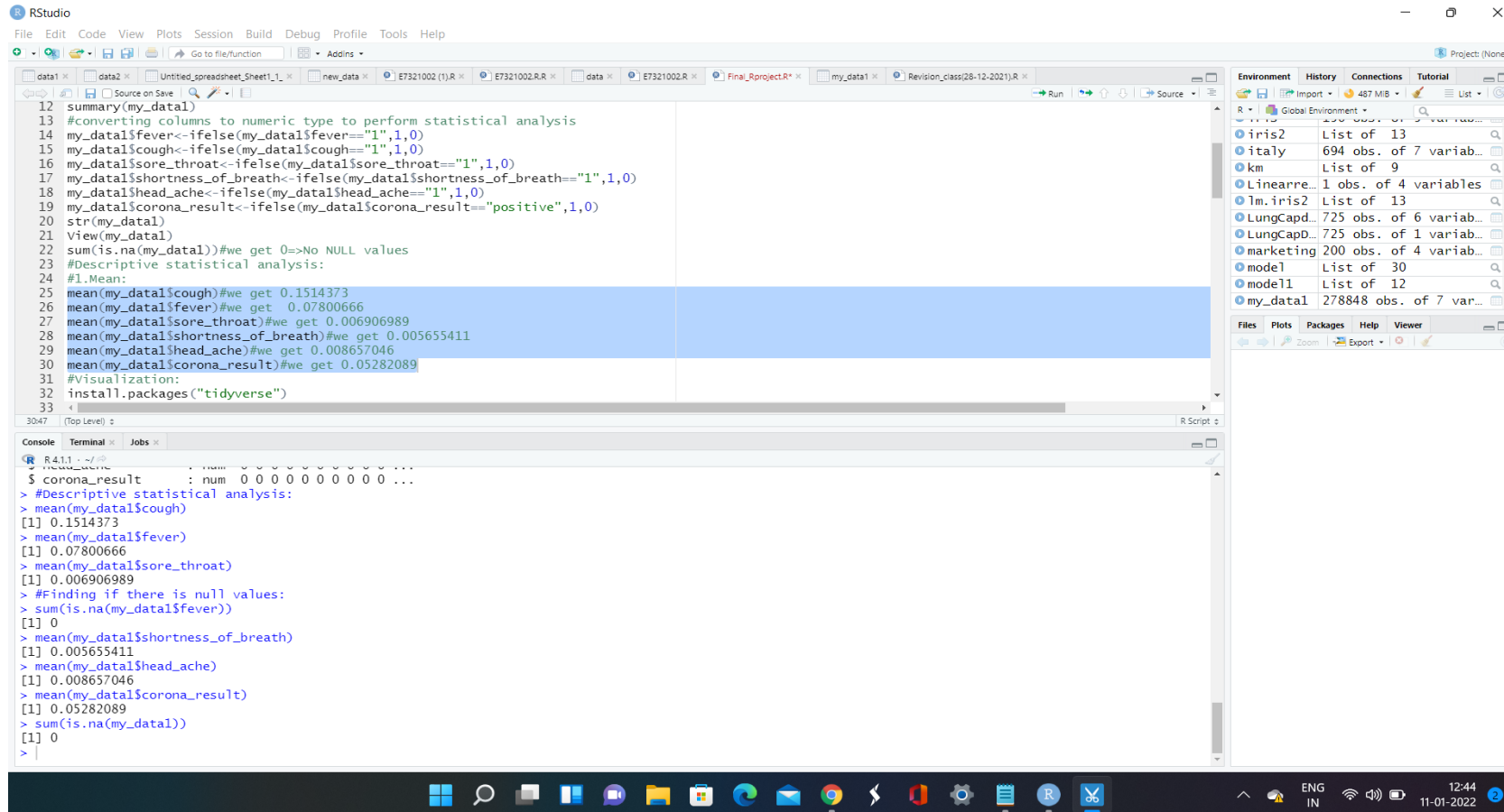
- Global Environment
- iris2: List of 13
- italy: 694 obs. of 7 variab...
- km: List of 9
- Linearre...: 1 obs. of 4 variables
- lm.iris2: List of 13
- LungCapd...: 725 obs. of 6 variab...
- LungCapD...: 725 obs. of 1 variab...
- marketing: 200 obs. of 4 variab...
- model: List of 30
- model1: List of 12
- my_data1: 278848 obs. of 7 var...

Checking if there is Null values:

```
>sum(is.na(my_data1))#we get 0=>No NULL values
```

DESCRIPTIVE STATISTICAL ANALYSIS:

1.Mean:



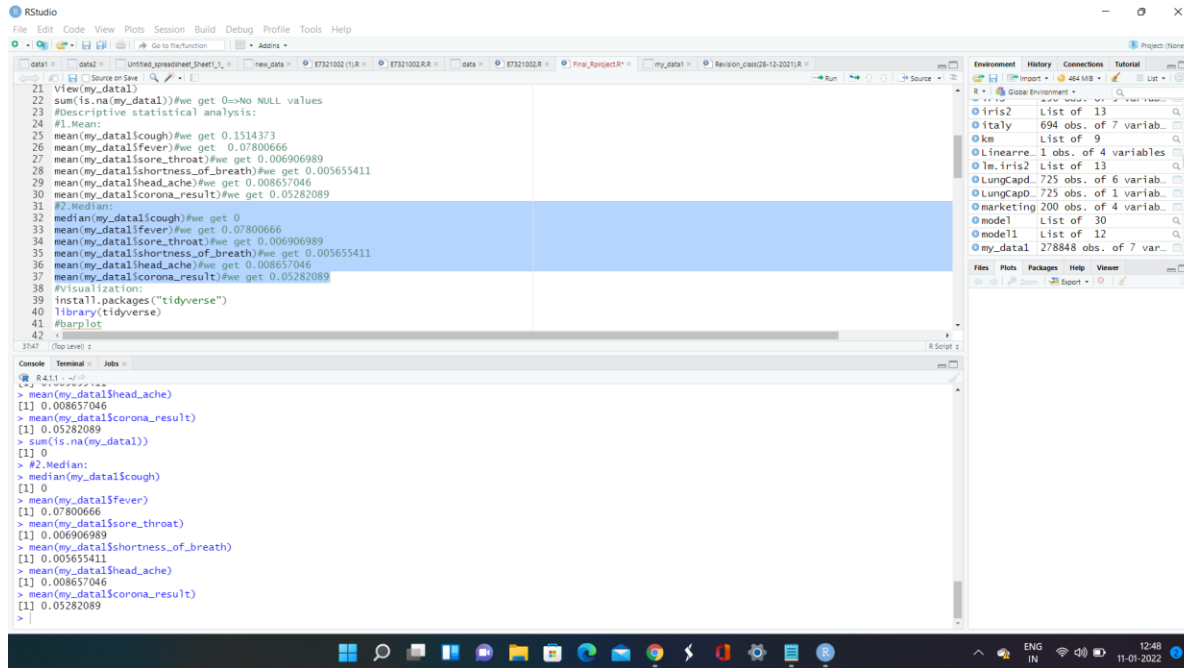
The screenshot displays the RStudio environment with a script editor on the left and a console on the bottom. The script editor contains R code for data manipulation and statistical analysis. The console shows the output of the executed code, including the structure of the data frame and the results of various mean calculations and null value checks.

```
12 summary(my_data1)
13 #converting columns to numeric type to perform statistical analysis
14 my_data1$fever<-ifelse(my_data1$fever=="1",1,0)
15 my_data1$cough<-ifelse(my_data1$cough=="1",1,0)
16 my_data1$sore_throat<-ifelse(my_data1$sore_throat=="1",1,0)
17 my_data1$shortness_of_breath<-ifelse(my_data1$shortness_of_breath=="1",1,0)
18 my_data1$head_ache<-ifelse(my_data1$head_ache=="1",1,0)
19 my_data1$corona_result<-ifelse(my_data1$corona_result=="positive",1,0)
20 str(my_data1)
21 View(my_data1)
22 sum(is.na(my_data1))#we get 0=>No NULL values
23 #Descriptive statistical analysis:
24 #1.Mean:
25 mean(my_data1$cough)#we get 0.1514373
26 mean(my_data1$fever)#we get 0.07800666
27 mean(my_data1$sore_throat)#we get 0.006906989
28 mean(my_data1$shortness_of_breath)#we get 0.005655411
29 mean(my_data1$head_ache)#we get 0.008657046
30 mean(my_data1$corona_result)#we get 0.05282089
31 #Visualization:
32 install.packages("tidyverse")
33
```

Console Output:

```
R 4.1.1 ~ ~/...
> corona_result
: num 0 0 0 0 0 0 0 0 0 0 ...
> #Descriptive statistical analysis:
> mean(my_data1$cough)
[1] 0.1514373
> mean(my_data1$fever)
[1] 0.07800666
> mean(my_data1$sore_throat)
[1] 0.006906989
> #Finding if there is null values:
> sum(is.na(my_data1$fever))
[1] 0
> mean(my_data1$shortness_of_breath)
[1] 0.005655411
> mean(my_data1$head_ache)
[1] 0.008657046
> mean(my_data1$corona_result)
[1] 0.05282089
> sum(is.na(my_data1))
[1] 0
>
```


► 2.MEDIAN:



The screenshot shows the RStudio interface. The script editor contains the following code:

```
21 View(my_data1)
22 sum(is.na(my_data1))#we get 0=>No NULL values
23 #Descriptive statistical analysis:
24 #1.Mean:
25 mean(my_data1$cough)#we get 0.1514373
26 mean(my_data1$fever)#we get 0.07800666
27 mean(my_data1$sore_throat)#we get 0.006906989
28 mean(my_data1$shortness_of_breath)#we get 0.005655411
29 mean(my_data1$head_ache)#we get 0.008657046
30 mean(my_data1$corona_result)#we get 0.05282089
31 #2.Median:
32 median(my_data1$cough)#we get 0
33 median(my_data1$fever)#we get 0.07800666
34 median(my_data1$sore_throat)#we get 0.006906989
35 median(my_data1$shortness_of_breath)#we get 0.005655411
36 median(my_data1$head_ache)#we get 0.008657046
37 median(my_data1$corona_result)#we get 0.05282089
38 #Visualization:
39 install.packages("tidyverse")
40 library(tidyverse)
41 #barplot
42
```

The console output shows the results of the commands:

```
R411 - RStudio
> mean(my_data1$head_ache)
[1] 0.008657046
> mean(my_data1$corona_result)
[1] 0.05282089
> sum(is.na(my_data1))
[1] 0
> #2.Median:
> median(my_data1$cough)
[1] 0
> mean(my_data1$fever)
[1] 0.07800666
> mean(my_data1$sore_throat)
[1] 0.006906989
> mean(my_data1$shortness_of_breath)
[1] 0.005655411
> mean(my_data1$head_ache)
[1] 0.008657046
> mean(my_data1$corona_result)
[1] 0.05282089
```

3. MODE:

```
>table(as.vector(my_data1$cough))
```

```
0      1
236620 42228
```

Interpretation:

We can see that 0 occurs most times=>0 is the mode for this column.
Similarly, we can find the mode for other columns.

4.VARIANCE:

```
>var(my_data1$sore_throat)
0.006859307
```

Similarly, we can find the variance of other columns

5.STANDARD DEVIATION:

```
>sqrt(var(my_data1$corona_result))
0.2236762
```

Similarly, we can find the standard deviation for other columns

6. RANGE:

range() gives min and max value of that column

```
>range(my_data1$shortness_of_breath)#we get min=0 and max=1
>range(my_data1$fever)#we get min=0 and max=1
```

Similarly we can find the range of other columns

TO USE skewness() and kurtosis() method, install moments package:

```
>install.packages("moments")
>library(moments)
```

7. SKEWNESS:

```
> skewness(my_data1$head_ache)
```

10.60762

We get 10.60762=>positively skewed

8. KURTOSIS:

```
>kurtosis(my_data1$cough)
```

4.781854

We get 4.781854=>Leptokurtic

DATA VISUALIZATION:

Install the packages "tidyverse" and "ggfortify"

```
>install.packages("tidyverse")
```

```
>library(tidyverse)
```

```
>install.packages("ggfortify")
```

```
>library(ggplot2)
```

BAR PLOT:

Bar plot w.r.t "corona_result" column

```
ggplot(my_data1,aes(x=corona_result))+
```

```
  geom_bar()+
```

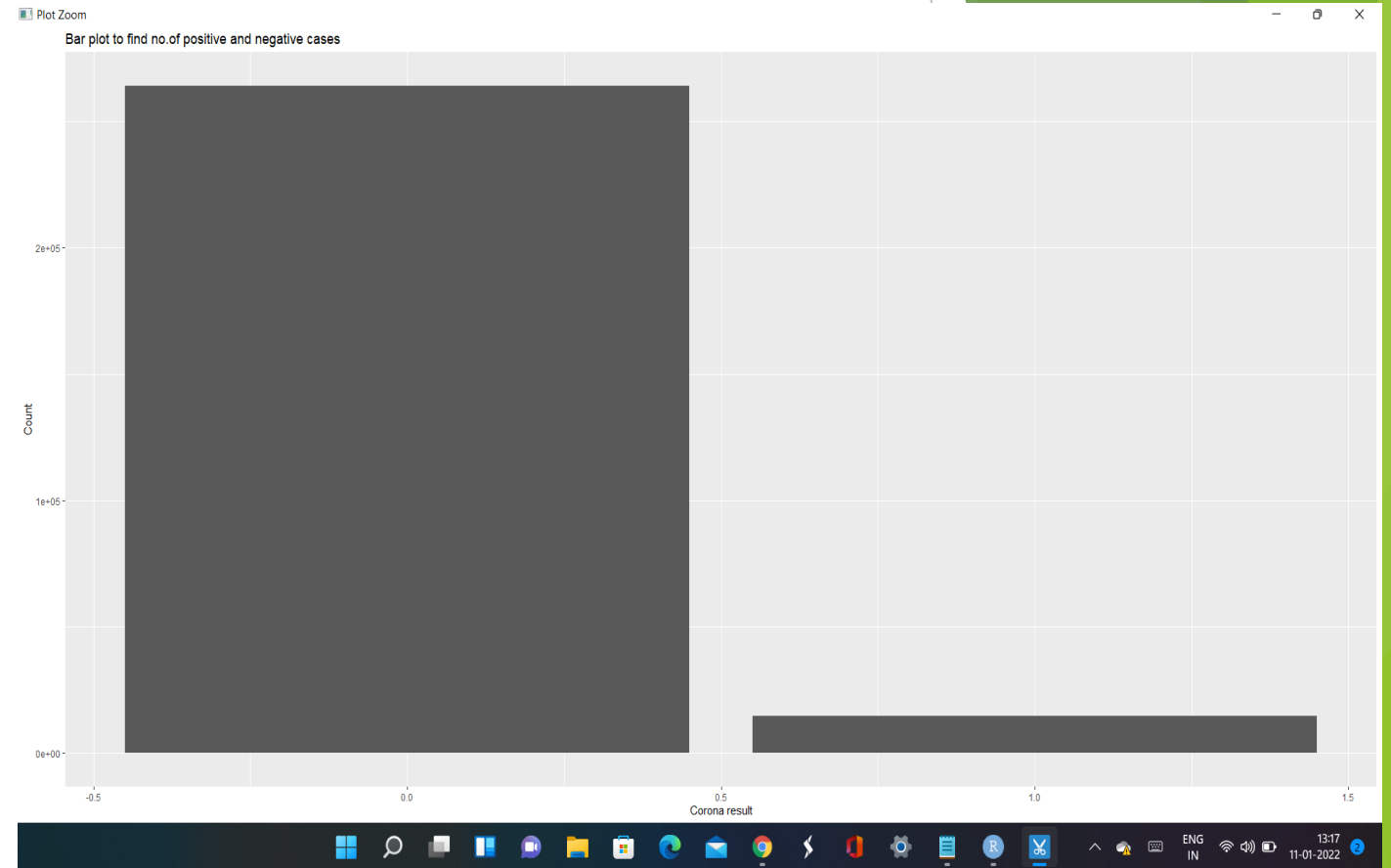
```
  xlab("Corona result")+
```

```
  ylab("Count")+
```

```
  ggtitle("Bar plot to find no.of positive  
and negative cases")
```

INTERPRETATION:

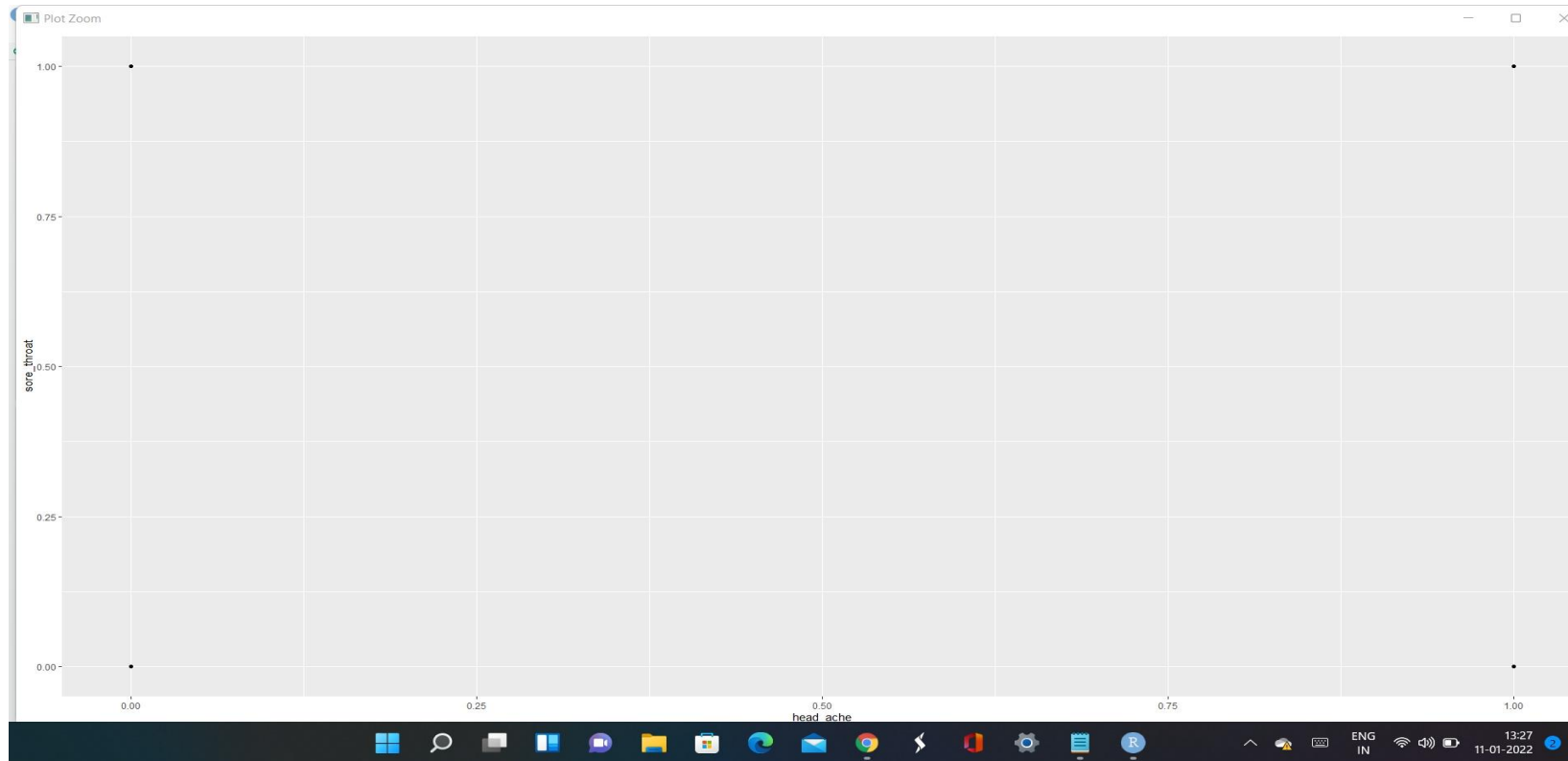
From the graph, we can see that, in the dataset, only few have been tested positive



SCATTER PLOT:

```
>ggplot(data=my_data1,aes(x=head_ache,y=sore_throat))+  
  geom_point()
```

Since,all our values are either 0 or 1 in our dataset,scatter plots are not useful.

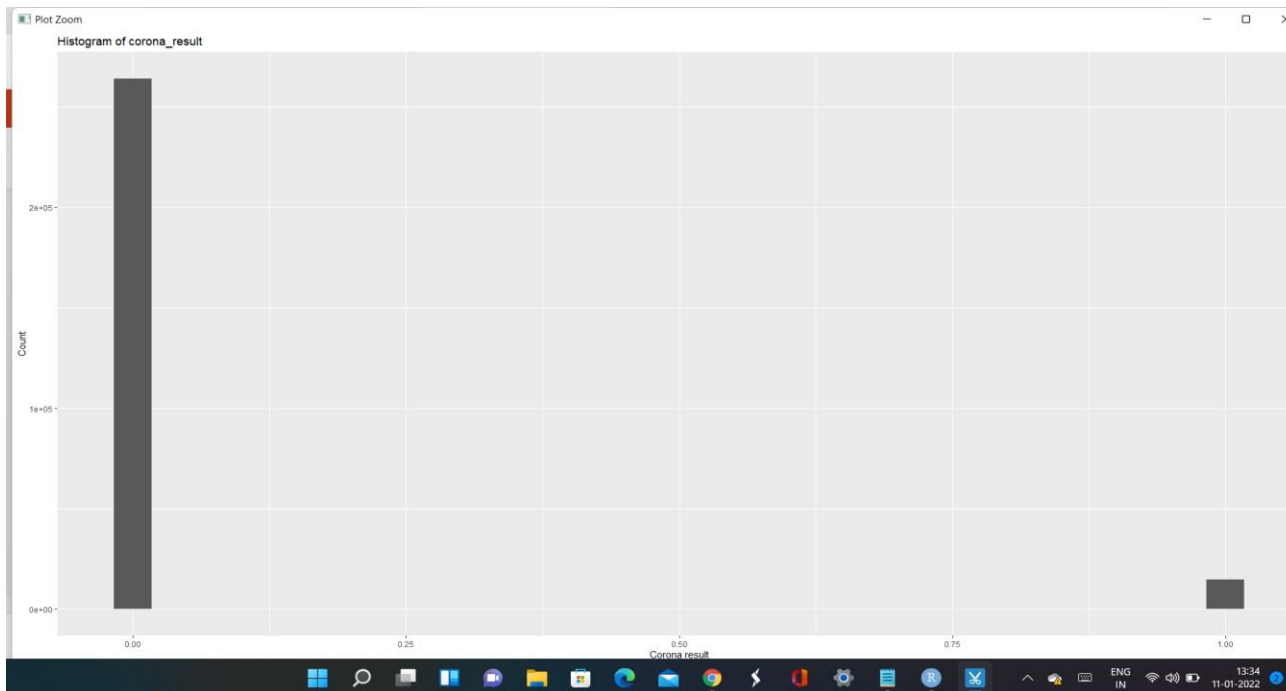


► HISTOGRAM:

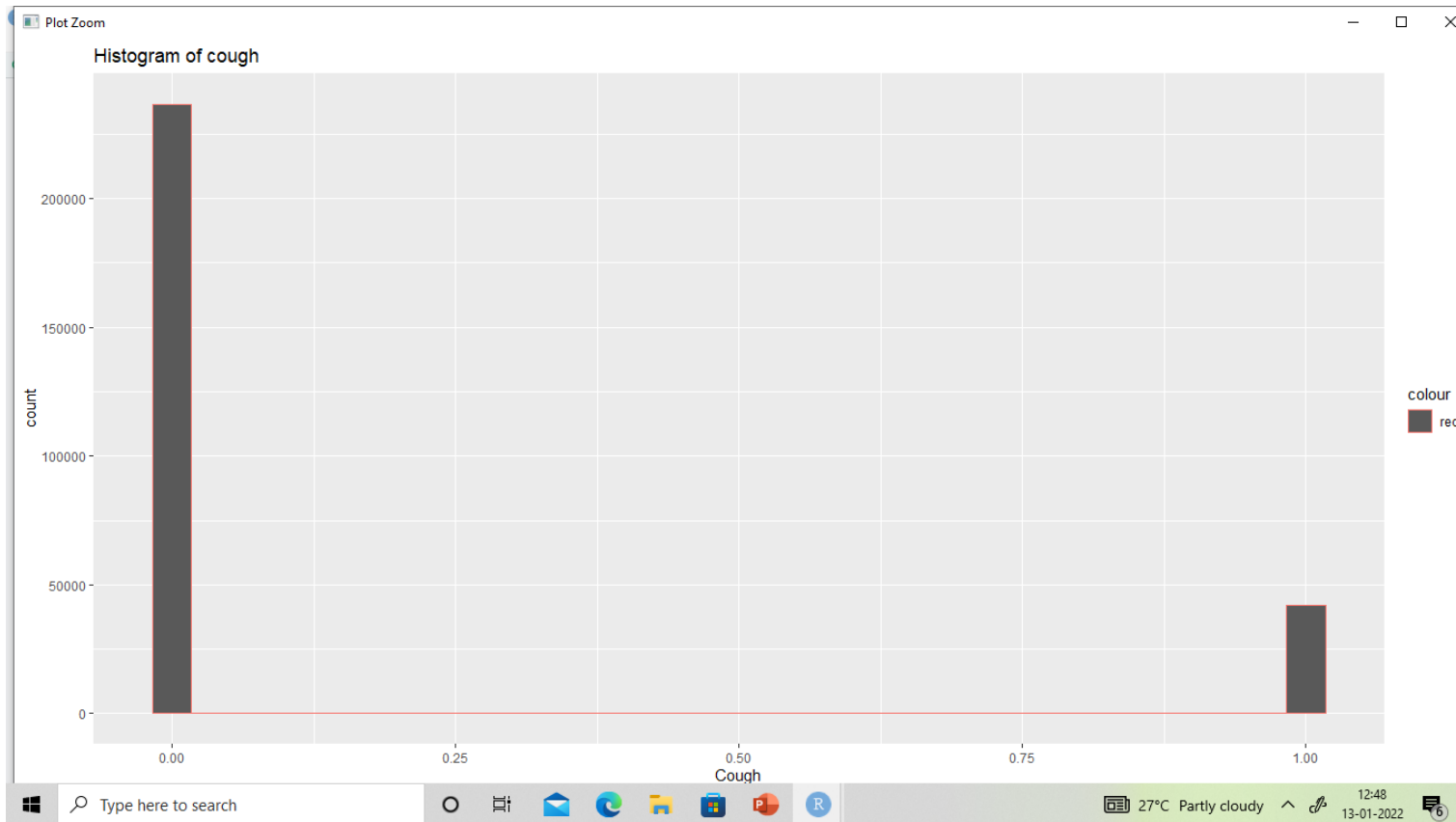
► (For variable corona result)

```
>ggplot(my_data1,aes(x=corona_result))+  
  geom_histogram(aes(fill=head_ache))+  
  xlab("Corona result")+  
  ylab("Count")+  
  ggtitle("Histogram of corona_result")
```

We can see that ,most of the people's corona_Result is 0=>negative.



- ▶ HISTOGRAM : (For variable cough)
- ▶ `ggplot(data=my_data1,aes(x=cough))+`
- ▶ `geom_histogram(aes(col="red"))+`
- ▶ `xlab("Cough")+`
- ▶ `ylab("count")+`
- ▶ `ggtitle("Histogram of cough")`



STATISTICAL ANALYSIS :

For hypothesis testing , we need these packages: ("tidyverse", "ggpubr", "rstatix")

Installing the packages:

```
install.packages("tidyverse")
```

```
library(tidyverse)
```

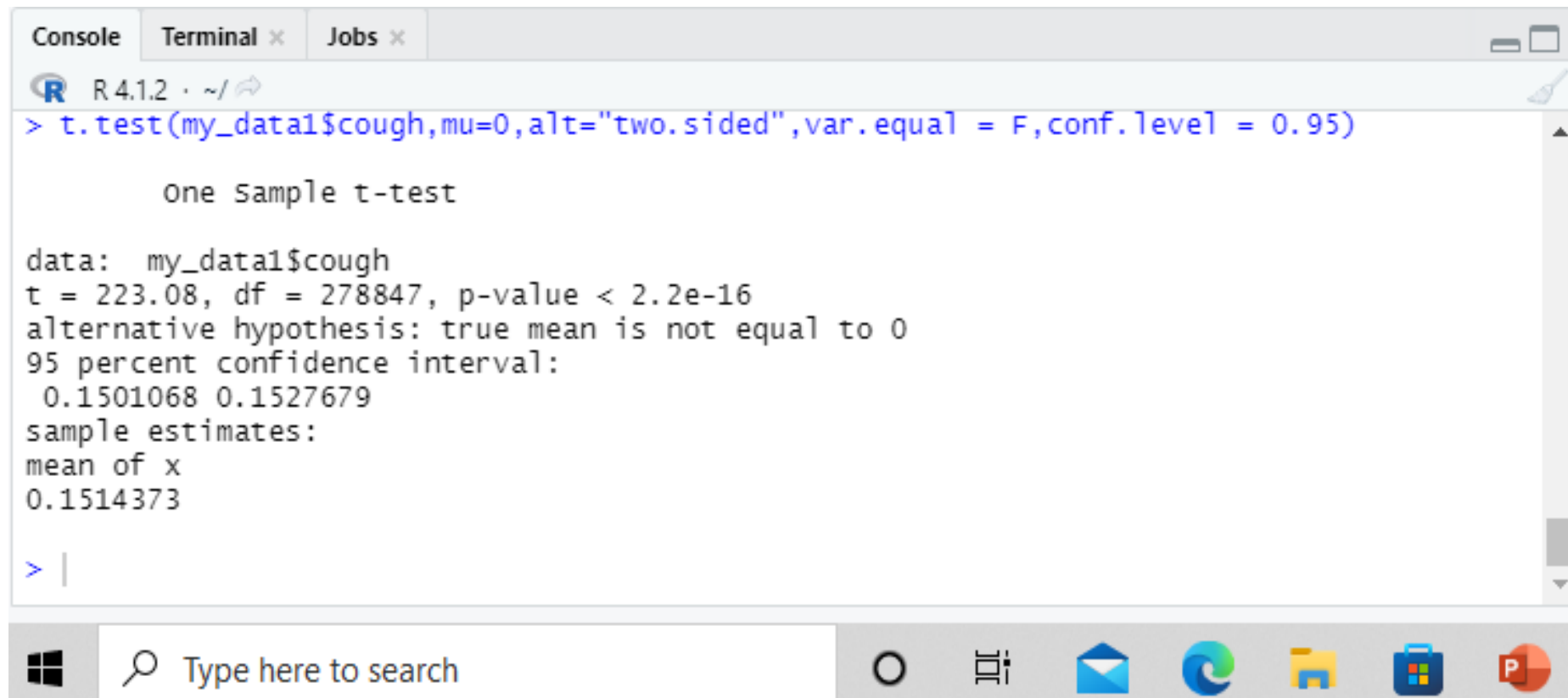
```
install.packages("ggpubr")
```

```
library(ggpubr)
```

```
install.packages("rstatix")
```

```
library(rstatix)
```


- ▶ 1.T-TEST
- ▶ Single-t-test
- ▶ i) H_0 : mean of cough column is 0
- ▶ `t.test(my_data1$cough,mu=0,alt="two.sided",var.equal = F,conf.level = 0.95)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get $p < 2.2e-16 < 0.05 \Rightarrow$ reject $H_0 \Rightarrow$ mean of cough column is not 0



```
Console Terminal x Jobs x
R 4.1.2 · ~/
> t.test(my_data1$cough,mu=0,alt="two.sided",var.equal = F,conf.level = 0.95)

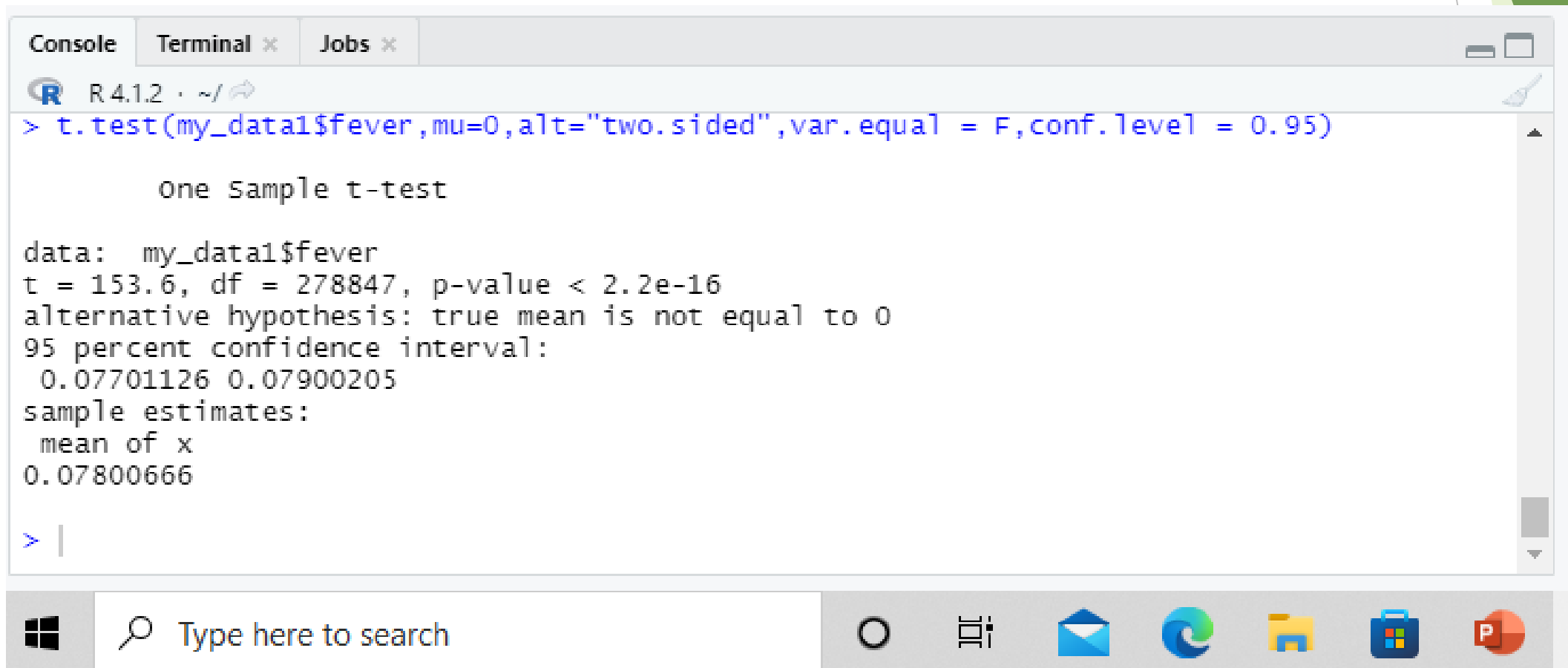
One sample t-test

data:  my_data1$cough
t = 223.08, df = 278847, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.1501068 0.1527679
sample estimates:
mean of x
0.1514373

> |
```

The screenshot shows an R console window with the following elements: a title bar with 'Console', 'Terminal x', and 'Jobs x'; a status bar indicating 'R 4.1.2 · ~/'; a command prompt where the `t.test` function is executed; and the resulting output of the t-test, including the test statistic, degrees of freedom, p-value, alternative hypothesis, confidence interval, and sample mean. The Windows taskbar is visible at the bottom with the search bar and several application icons.

- ▶ ii) H_0 : mean of fever column is 0
- ▶ `t.test(my_data1$fever, mu=0, alt="two.sided", var.equal = F, conf.level = 0.95)`
- ▶ **INTERPRETATION:**
- ▶ From the analysis , we get $p < 2.2e-16 < 0.05 \Rightarrow H_0$ is rejected \Rightarrow mean of fever column is not 0



```
Console Terminal x Jobs x
R 4.1.2 · ~/
> t.test(my_data1$fever, mu=0, alt="two.sided", var.equal = F, conf.level = 0.95)

      One Sample t-test

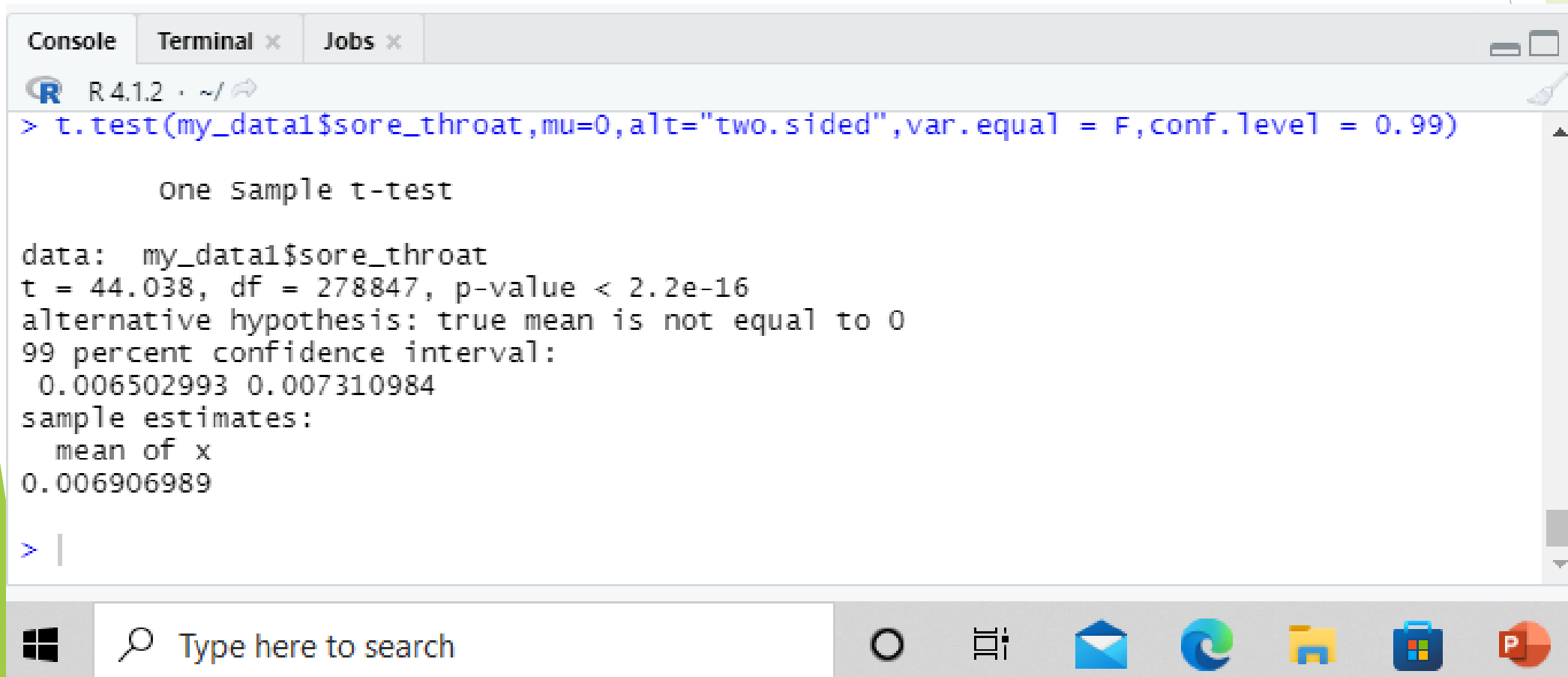
data:  my_data1$fever
t = 153.6, df = 278847, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.07701126 0.07900205
sample estimates:
mean of x
0.07800666

> |
```

The screenshot shows an R console window with the following elements:

- Tab Bar:** Contains 'Console', 'Terminal x', and 'Jobs x' tabs.
- Header:** Displays 'R 4.1.2 · ~/'. There is a search icon on the right.
- Code Input:** The command `t.test(my_data1$fever, mu=0, alt="two.sided", var.equal = F, conf.level = 0.95)` is entered.
- Output:** The console displays the results of a one-sample t-test, including the test statistic (t = 153.6), degrees of freedom (df = 278847), p-value ($< 2.2e-16$), alternative hypothesis, 95% confidence interval, and sample estimates.
- Taskbar:** At the bottom, there is a Windows taskbar with a search bar and several application icons (File Explorer, Edge, etc.).

- ▶ iii) H_0 : mean of sore throat column is 0
- ▶ `t.test(my_data1$sore_throat, mu=0, alt="two.sided", var.equal = F, conf.level = 0.99)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get $p < 2.2e-16 < 0.05 \Rightarrow H_0$ is rejected \Rightarrow mean of sore throat column is not 0



```
Console Terminal x Jobs x
R 4.1.2 · ~/
> t.test(my_data1$sore_throat, mu=0, alt="two.sided", var.equal = F, conf.level = 0.99)

One sample t-test

data:  my_data1$sore_throat
t = 44.038, df = 278847, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
99 percent confidence interval:
 0.006502993 0.007310984
sample estimates:
 mean of x
0.006906989

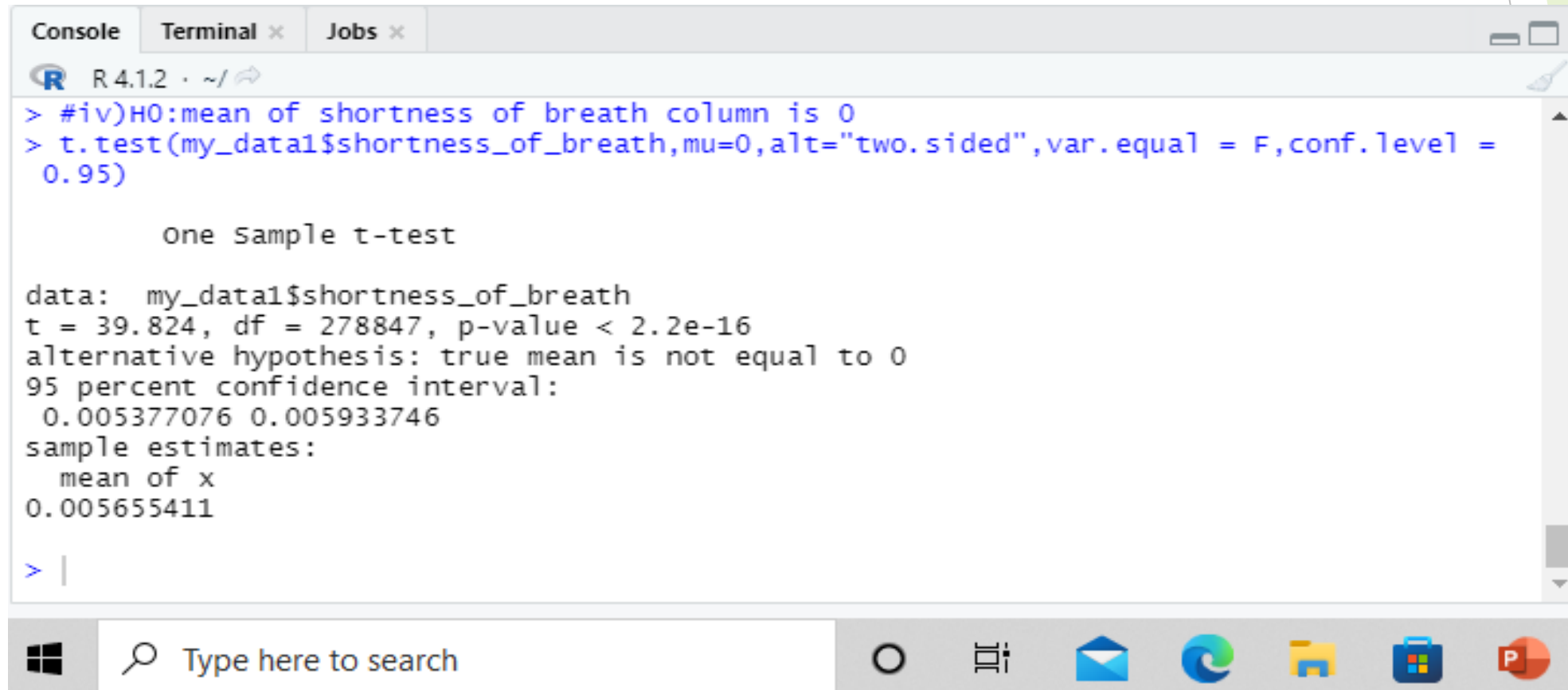
> |
```

The screenshot shows an R console window with the following elements:

- Tab bar: Console, Terminal x, Jobs x
- Header: R 4.1.2 · ~/
- Input: `> t.test(my_data1$sore_throat, mu=0, alt="two.sided", var.equal = F, conf.level = 0.99)`
- Output:
 - One sample t-test
 - data: my_data1\$sore_throat
 - t = 44.038, df = 278847, p-value < 2.2e-16
 - alternative hypothesis: true mean is not equal to 0
 - 99 percent confidence interval:
0.006502993 0.007310984
 - sample estimates:
mean of x
0.006906989
- Cursor: `> |`

At the bottom of the image, a Windows taskbar is visible with the Start button, a search bar containing "Type here to search", and several application icons including File Explorer, Microsoft Edge, and PowerPoint.

- ▶ iv) H_0 : mean of shortness of breath column is 0
- ▶ `t.test(my_data1$shortness_of_breath, mu=0, alt="two.sided", var.equal = F, conf.level = 0.95)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get $p < 2.2e-16 < 0.05 \Rightarrow H_0$ is rejected \Rightarrow mean of shortness of breath column is not 0



```
Console Terminal x Jobs x
R 4.1.2 · ~/
> #iv)H0:mean of shortness of breath column is 0
> t.test(my_data1$shortness_of_breath,mu=0,alt="two.sided",var.equal = F,conf.level = 0.95)

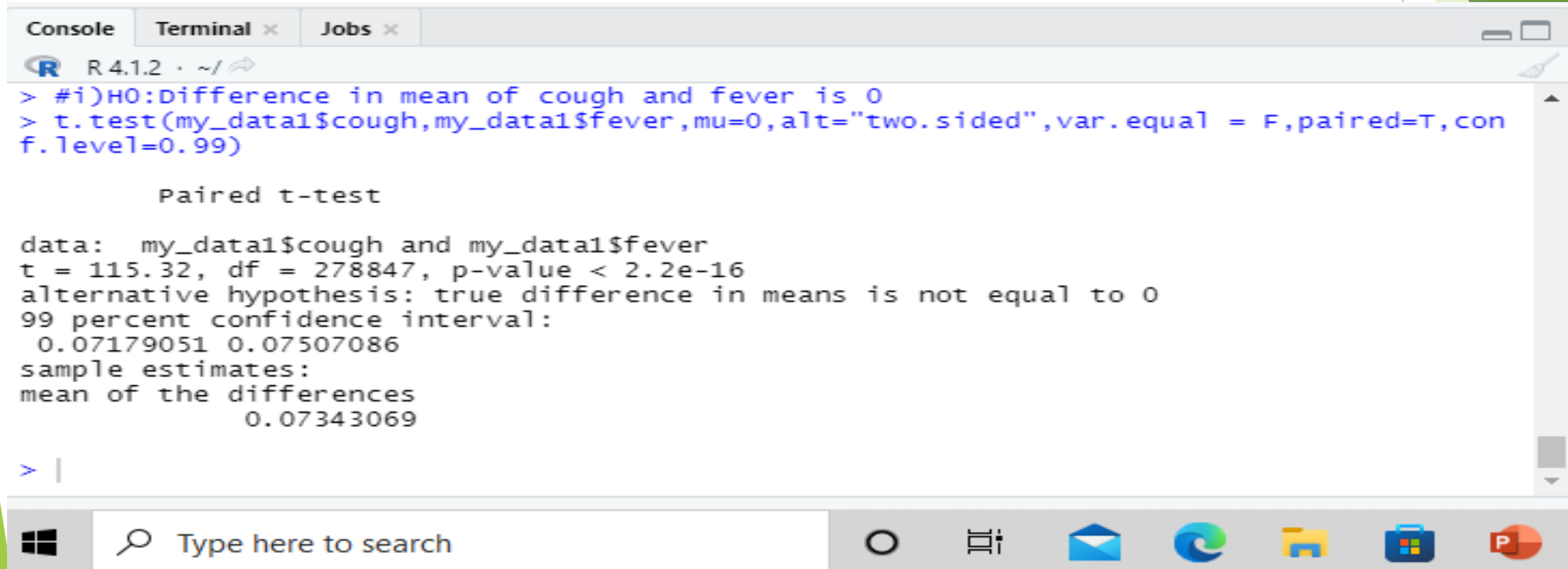
      one sample t-test

data:  my_data1$shortness_of_breath
t = 39.824, df = 278847, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.005377076 0.005933746
sample estimates:
mean of x
0.005655411

> |
```

The screenshot shows an R console window with the following elements: a title bar with 'Console', 'Terminal x', and 'Jobs x'; a status bar indicating 'R 4.1.2 · ~/'; and a command prompt where a t-test was performed. The output displays the test statistics, p-value, confidence interval, and sample mean. The Windows taskbar is visible at the bottom with icons for search, task view, mail, edge, file explorer, and other applications.

- ▶ PAIRED t-test
- ▶ i)H0:Difference in mean of cough and fever is 0
- ▶ `t.test(my_data1$cough,my_data1$fever,mu=0,alt="two.sided",var.equal = F,paired=T,conf.level=0.99)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get $t = 115.32$, $df = 278847$, $p\text{-value} < 2.2e-16$ which is less than 0.05.
- ▶ So H0 rejected ,alternative hypothesis: true difference in means is not equal to 0



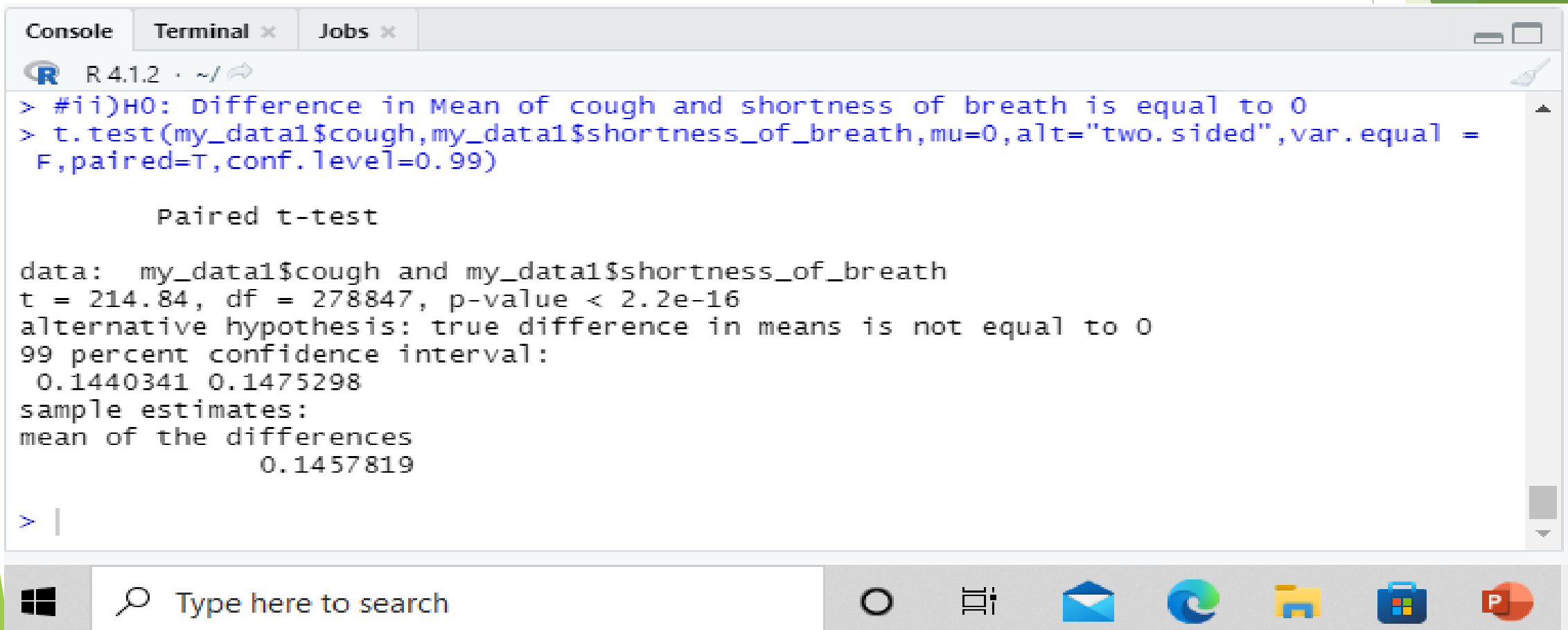
```
Console Terminal x Jobs x
R 4.1.2 · ~/
> #i)H0:Difference in mean of cough and fever is 0
> t.test(my_data1$cough,my_data1$fever,mu=0,alt="two.sided",var.equal = F,paired=T,conf.level=0.99)

Paired t-test

data: my_data1$cough and my_data1$fever
t = 115.32, df = 278847, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 0.07179051 0.07507086
sample estimates:
mean of the differences
      0.07343069

> |
```

- ▶ ii)H0: Difference in Mean of cough and shortness of breath is equal to 0
- ▶ `t.test(my_data1$cough,my_data1$shortness_of_breath,mu=0,alt="two.sided",var.equal = F,paired=T,conf.level=0.99)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get = 214.84, df = 278847, p-value < 2.2e-16 which is less than 0.05.
- ▶ So H0 rejected ,alternative hypothesis: true difference in means is not equal to 0



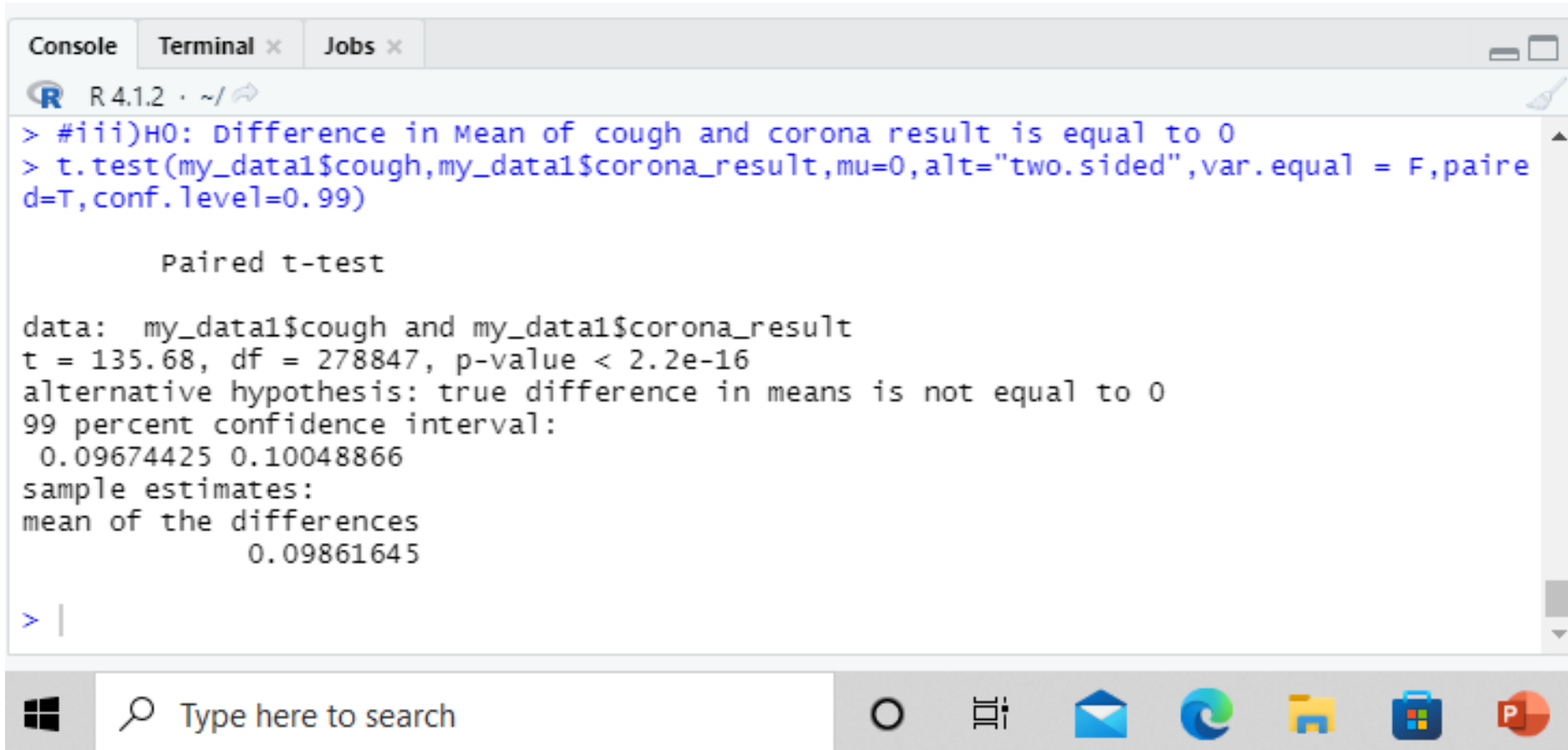
```
Console Terminal x Jobs x
R 4.1.2 · ~/
> #ii)H0: Difference in Mean of cough and shortness of breath is equal to 0
> t.test(my_data1$cough,my_data1$shortness_of_breath,mu=0,alt="two.sided",var.equal =
  F,paired=T,conf.level=0.99)

      Paired t-test

data:  my_data1$cough and my_data1$shortness_of_breath
t = 214.84, df = 278847, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 0.1440341 0.1475298
sample estimates:
mean of the differences
          0.1457819

> |
```

- ▶ iii)H0: Difference in Mean of cough and corona result is equal to 0
- ▶ `t.test(my_data1$cough,my_data1$corona_result,mu=0,alt="two.sided",var.equal = F,paired=T,conf.level=0.99)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get $t = 135.68$, $df = 278847$, $p\text{-value} < 2.2e-16$ which is less than 0.05.
- ▶ So H0 rejected ,alternative hypothesis: true difference in means is not equal to 0



```
Console Terminal x Jobs x
R 4.1.2 · ~/
> #iii)H0: Difference in Mean of cough and corona result is equal to 0
> t.test(my_data1$cough,my_data1$corona_result,mu=0,alt="two.sided",var.equal = F,paired=T,conf.level=0.99)

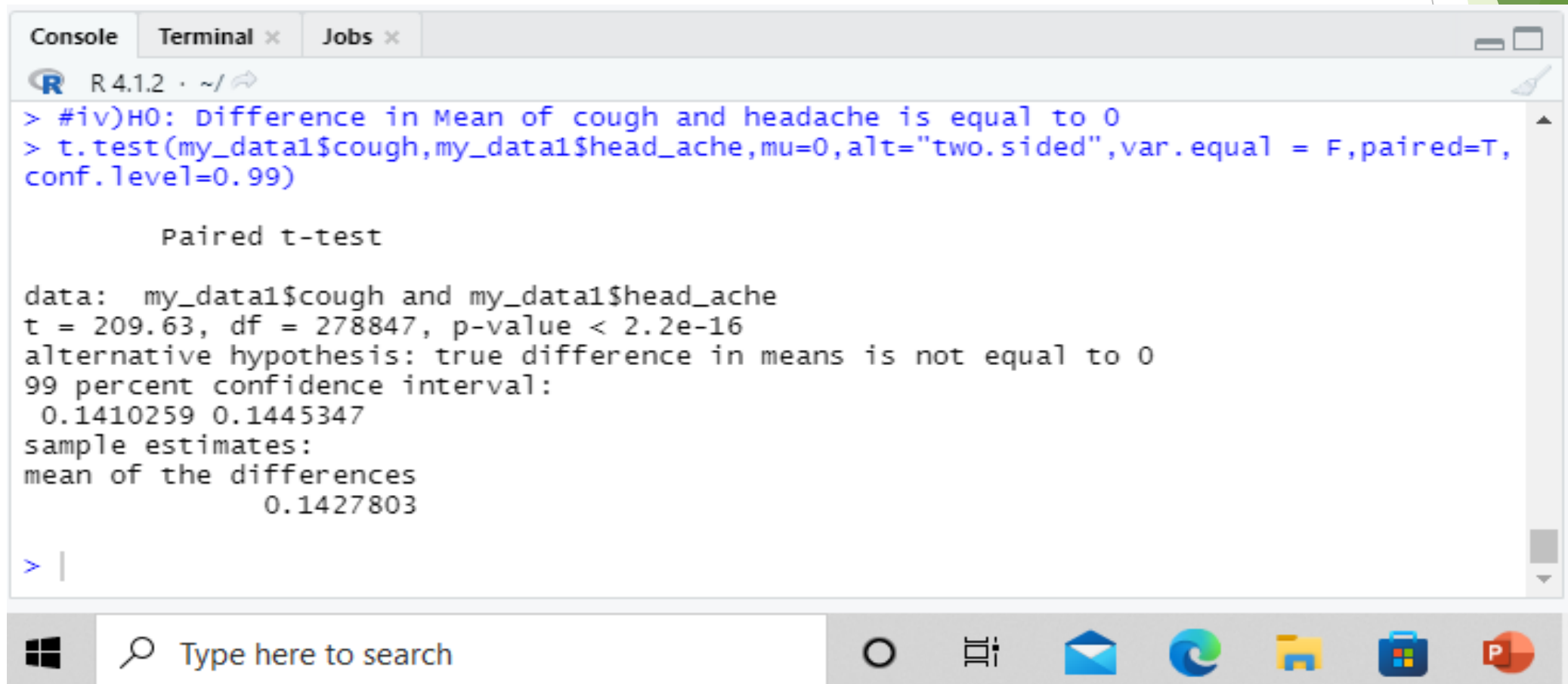
      Paired t-test

data:  my_data1$cough and my_data1$corona_result
t = 135.68, df = 278847, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 0.09674425 0.10048866
sample estimates:
mean of the differences
      0.09861645

> |
```

The screenshot shows an R console window with the following elements: a tab bar at the top with 'Console', 'Terminal', and 'Jobs'; a title bar indicating 'R 4.1.2 · ~/'; and a command prompt where a paired t-test was performed. The output displays the test statistics, p-value, confidence interval, and sample mean difference. At the bottom of the image, a Windows taskbar is visible with the search bar and several application icons.

- ▶ iv)H0: Difference in Mean of cough and headache is equal to 0
- ▶ `t.test(my_data1$cough,my_data1$head_ache,mu=0,alt="two.sided",var.equal = F,paired=T,conf.level=0.99)`
- ▶ INTERPRETATION:
- ▶ From the analysis ,we get $t = 209.63$, $df = 278847$, $p\text{-value} < 2.2e-16$ which is less than 0.05.
- ▶ So H0 rejected ,alternative hypothesis: true difference in means is not equal to 0



```
Console Terminal x Jobs x
R 4.1.2 · ~/
> #iv)H0: Difference in Mean of cough and headache is equal to 0
> t.test(my_data1$cough,my_data1$head_ache,mu=0,alt="two.sided",var.equal = F,paired=T,
conf.level=0.99)

      Paired t-test

data:  my_data1$cough and my_data1$head_ache
t = 209.63, df = 278847, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 0.1410259 0.1445347
sample estimates:
mean of the differences
          0.1427803

> |
```


► 2.CORRELATION:

► For performing correlation we need these packages:(“devtools”,”ggpubr”)

► Installing the packages:

► `install.packages("devtools")`

► `library(devtools)`

► `install.packages("ggpubr")`

► `library(ggpubr)`

► 1.Testing correlation for fever and cough using pearson Method

► H0:there is no correlation btw fever and cough

► `cor(my_data1$fever,my_data1$cough,method="pearson")`

► INTERPRETATION:

► we get $\rho=0.454386$ which is not equal to 0.so H0 rejected=>there is correlation bw fever and cough

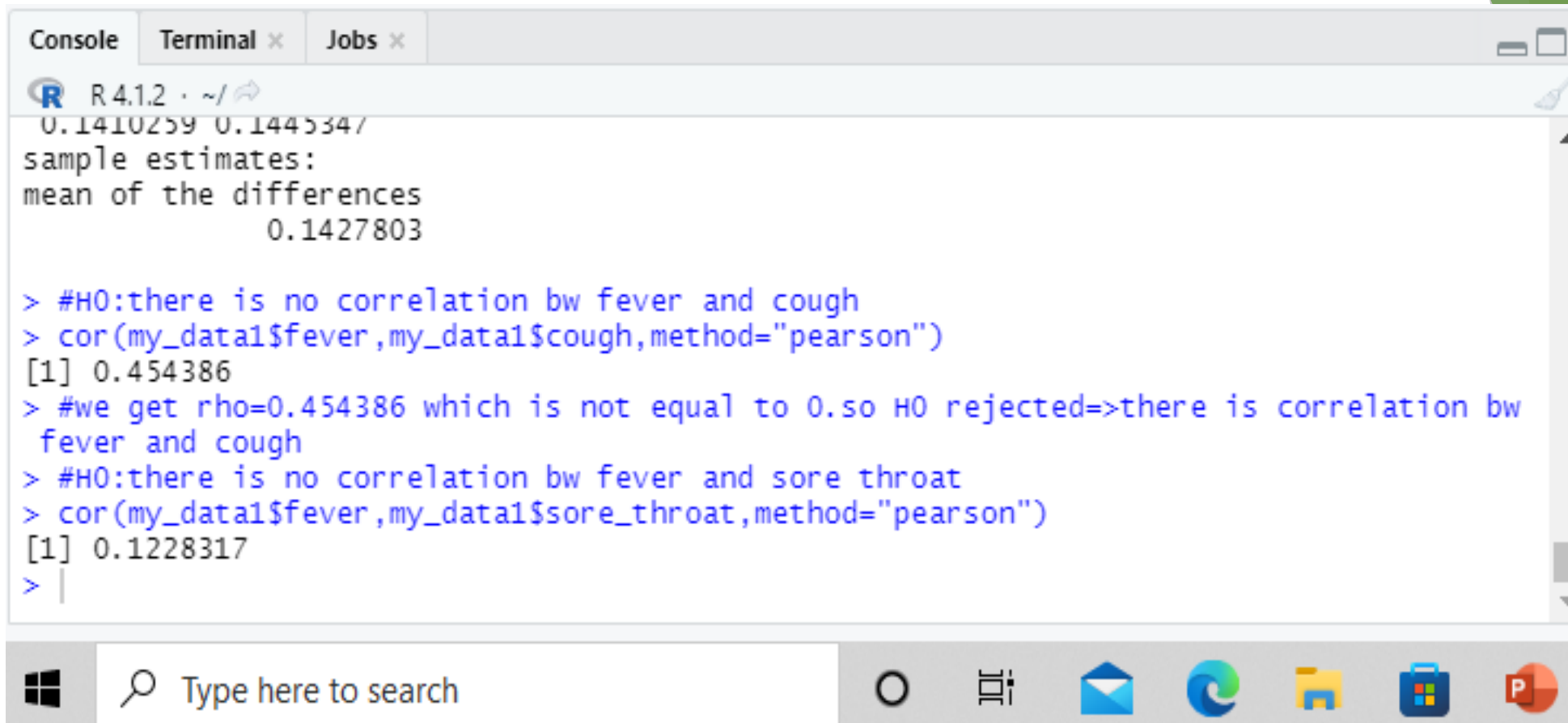
► 2.Testing correlation for fever and sore throat using pearson Method

► H0:there is no correlation bw fever and sore throat

► `cor(my_data1$fever,my_data1$sore_throat,method="pearson")`

► INTERPRETATION:

► From the analysis ,we get $\rho=0.1228317$ which is not equal to 0.so H0 rejected=>there is correlation btw fever and sore_throat



```
R 4.1.2 · ~/
0.1410259 0.1445347
sample estimates:
mean of the differences
      0.1427803

> #H0:there is no correlation bw fever and cough
> cor(my_data1$fever,my_data1$cough,method="pearson")
[1] 0.454386
> #we get rho=0.454386 which is not equal to 0.so H0 rejected=>there is correlation bw
  fever and cough
> #H0:there is no correlation bw fever and sore throat
> cor(my_data1$fever,my_data1$sore_throat,method="pearson")
[1] 0.1228317
> |
```

Performing correl Method Using agricolae package:

```
install.packages("agricolae")
```

```
library(agricolae)
```

Correlation using correl Method:

H0:there is no correlation btw fever and headache

- ▶ `correl(my_data1$fever,my_data1$head_ache,method="pearson")`
- ▶ **INTERPRETATION:**
- ▶ From the analysis ,we get $\rho=0.1688408$ which is not equal to 0.so H0 rejected=>there is correlation btw fever and headache
- ▶ **Prforming cor.test Using stats package:**
- ▶ `install.packages("stats")`
- ▶ `library(stats)`
- ▶ H0:there is no correlation bw fever and corona result
- ▶ `cor.test(my_data1$fever,my_data1$corona_result,method="pearson",conf.level = 0.95)`
- ▶ **INTERPRETATION:**
- ▶ From the analysis ,we get $\text{cor}=0.2636491$ which is not equal to 0.so H0 rejected=>There is correlation between fever and corona_result

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Final_Rproject.R x Final_Rproject[329].R* x C x cor x my_data1 x corona_tested_dataset x

Source on Save Run Source

```
136 #Using agricolae package:
137 install.packages("agricolae")
138 library(agricolae)
139 #H0:there is no correlation bw fever and headache
140 correl(my_data1$fever,my_data1$head_ache,method="pearson")
141 #we get rho=0.1688408 which is not equal to 0.so H0 rejected=>there is correlatio
142 #Using stats package:
143 install.packages("stats")
144
```

147:1 (Top Level) R Script

Console Terminal x Jobs x

R 4.1.2 ~/

```
> correl(my_data1$fever,my_data1$head_ache,method="pearson")
$stat
[1] 90.45651

$rho
[1] 0.1688408

$spvalue
[1] 0

> #H0:there is no correlation bw fever and corona result
> cor.test(my_data1$fever,my_data1$corona_result,method="pearson",conf.level = 0.95)

Pearson's product-moment correlation

data: my_data1$fever and my_data1$corona_result
t = 144.33, df = 278846, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2601921 0.2670994
sample estimates:
      cor
0.2636491

> |
```

Type here to search

2

▶ 3.ANOVA

▶ one-way anova

▶ i) H_0 :head_ache has no impact on corona_result

▶ `aov1<-aov(my_data1$corona_result~my_data1$head_ache)`

▶ `summary(aov1)`

▶ INTERPRETATION:

▶ From the analysis ,we get $p < 2e-16 < 0.05 \Rightarrow H_0$ rejected \Rightarrow head_ache has impact on corona_result

▶ ii) H_0 :cough has no impact on corona_Result

▶ `aov2<-aov(my_data1$corona_result~my_data1$cough)`

▶ `summary(aov2)`

▶ INTERPRETATION:

▶ From the analysis , we get $p < 2e-16 < 0.05 \Rightarrow H_0$ rejected \Rightarrow cough has impact on corona_result

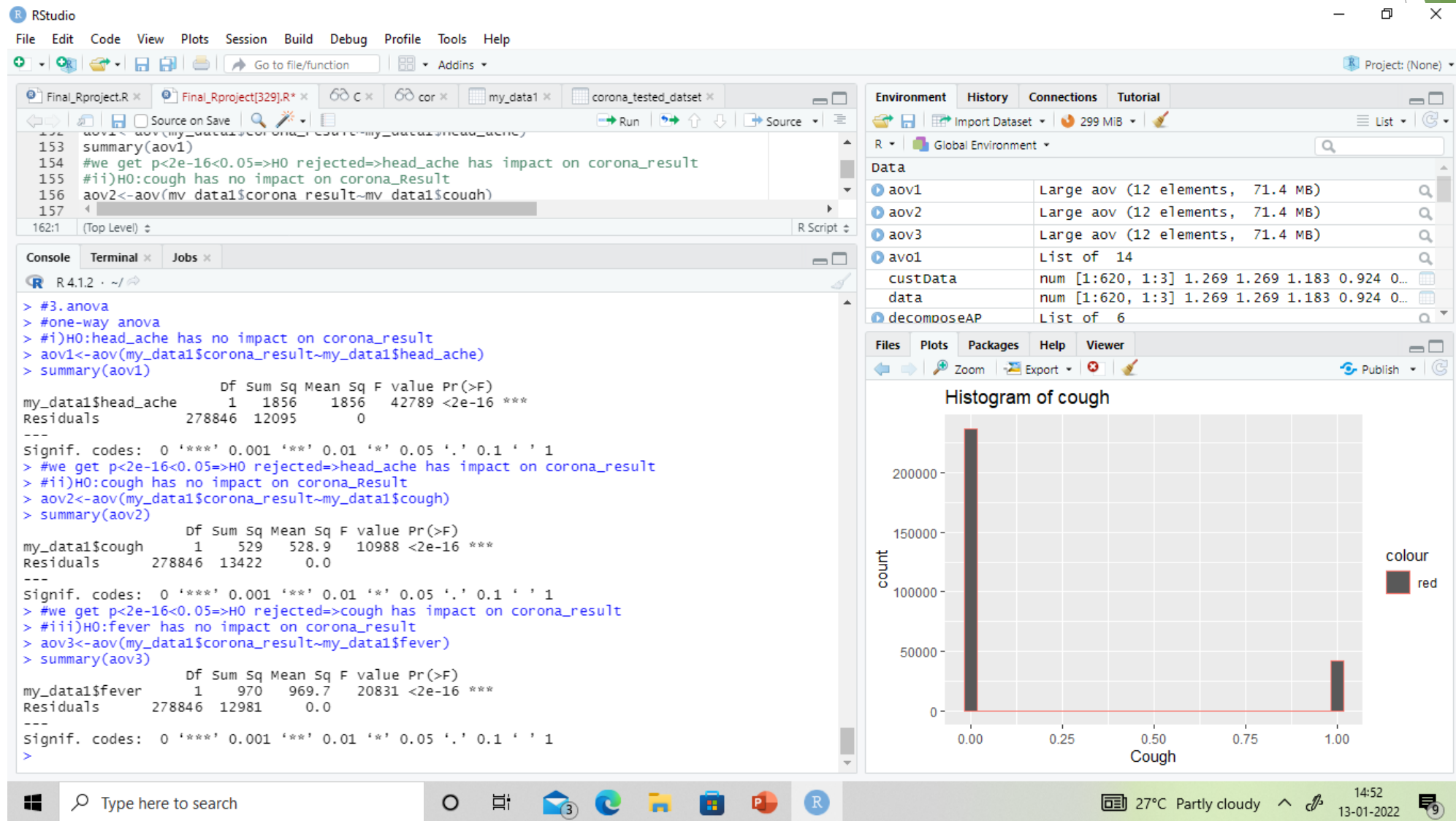
▶ iii) H_0 :fever has no impact on corona_result

▶ `aov3<-aov(my_data1$corona_result~my_data1$fever)`

▶ `summary(aov3)`

▶ INTERPRETATION:

▶ From the analysis , we get $p < 2e-16 < 0.05 \Rightarrow H_0$ rejected \Rightarrow fever has impact on corona_result



- ▶ 2 way anova

- ▶ i) H_0 : fever and sore_throat has no impact on cough

- ▶ `str(my_data1)`

- ▶ `aov4<-aov(cough~fever+sore_throat,data=my_data1)`

- ▶ `aov4`

- ▶ `summary(aov4)`

- ▶ INTERPRETATION:

- ▶ From the analysis , we get $p < 2e-16 < 0.05$, so H_0 rejected \Rightarrow fever and sore_throat has impact on cough

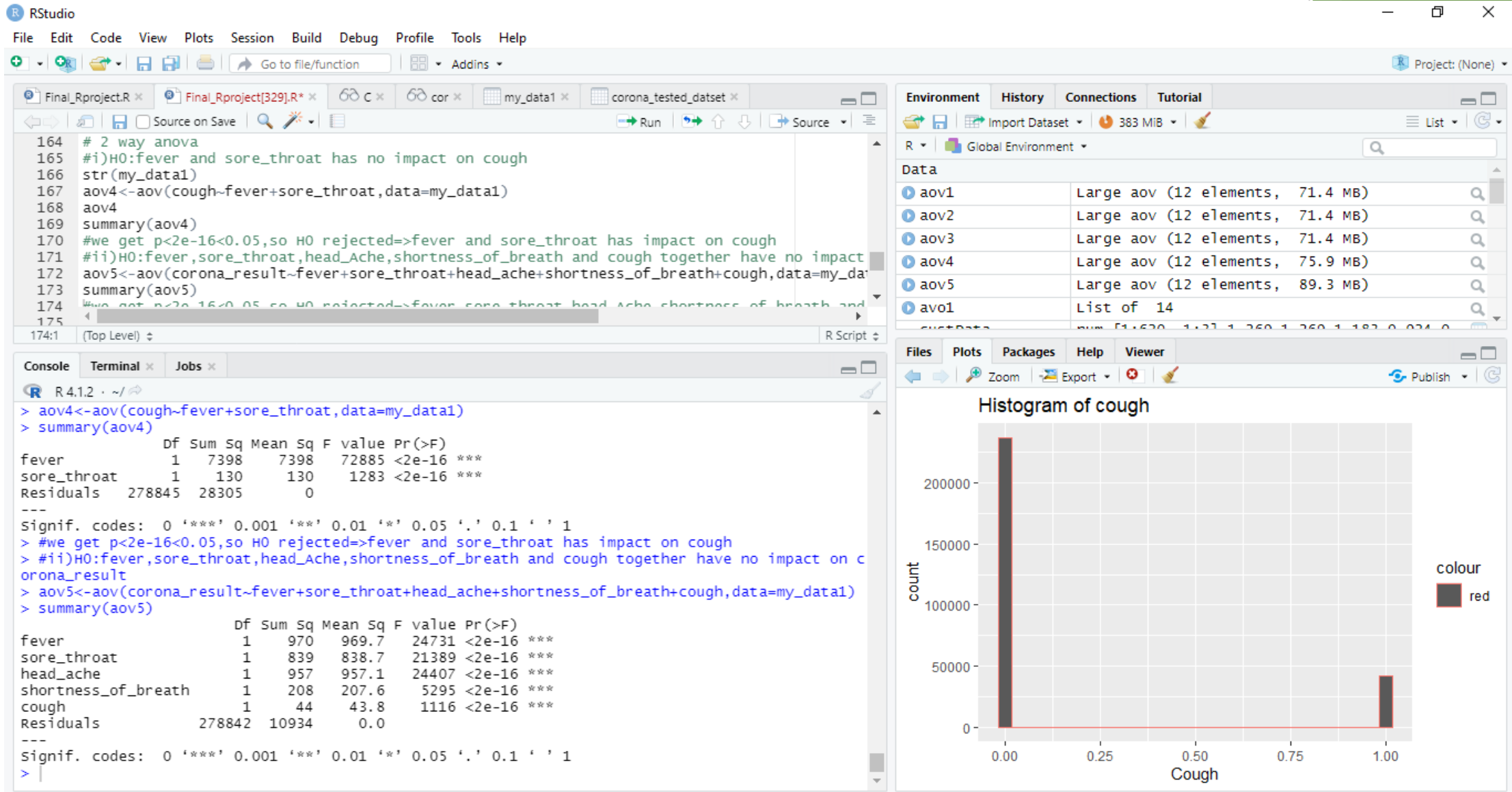
- ▶ ii) H_0 : fever, sore_throat, head_Ache, shortness_of_breath and cough together have no impact on corona_result

- ▶ `aov5<-`
`aov(corona_result~fever+sore_throat+head_ache+shortness_of_breath+cough,data=my_data1)`

- ▶ `summary(aov5)`

- ▶ INTERPRETATION:

- ▶ From the analysis , we get $p < 2e-16 < 0.05$, so H_0 rejected \Rightarrow fever, sore_throat, head_Ache, shortness_of_breath and cough together have impact on corona_result



Since our dataset contains class label, we perform classification.

LOGISTIC REGRESSION:

Install caTools package:

```
>install.packages("caTools")
```

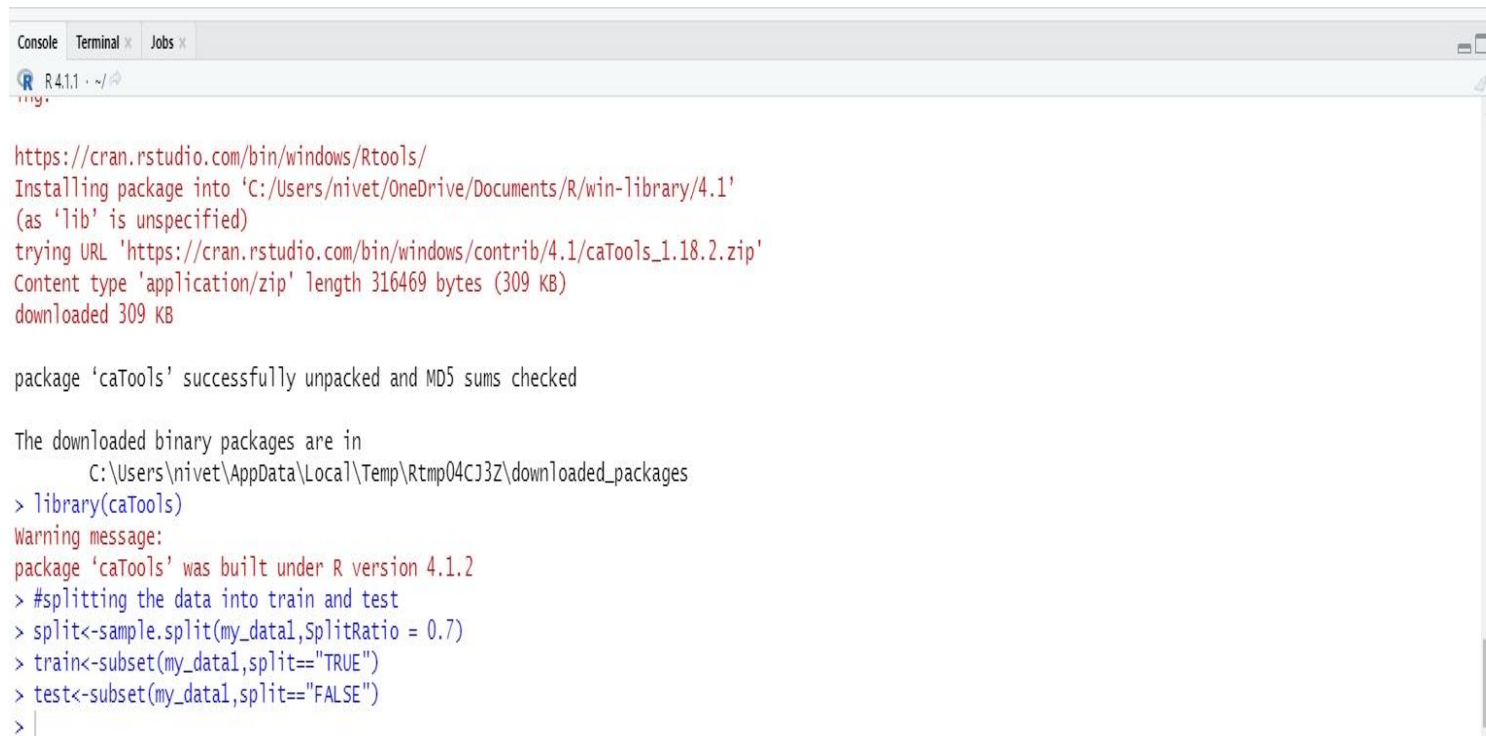
```
>library(caTools)
```

Split the dataset into train and test:

```
>split<-sample.split(my_data1,SplitRatio = 0.7)
```

```
>train<-subset(my_data1,split=="TRUE")
```

```
>test<-subset(my_data1,split=="FALSE")
```



```
R 4.1.1 ~/\nhttps://cran.rstudio.com/bin/windows/Rtools/\nInstalling package into 'C:/Users/nivet/OneDrive/Documents/R/win-library/4.1'\n(as 'lib' is unspecified)\ntrying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/caTools_1.18.2.zip'\nContent type 'application/zip' length 316469 bytes (309 KB)\ndownloaded 309 KB\n\npackage 'caTools' successfully unpacked and MD5 sums checked\n\nThe downloaded binary packages are in\n  C:/Users/nivet/AppData/Local/Temp/Rtmp04CJ3Z/downloaded_packages\n> library(caTools)\nWarning message:\npackage 'caTools' was built under R version 4.1.2\n> #splitting the data into train and test\n> split<-sample.split(my_data1,SplitRatio = 0.7)\n> train<-subset(my_data1,split=="TRUE")\n> test<-subset(my_data1,split=="FALSE")\n> |
```

We build the logistic regression model w.r.t the dependent variable corona_result and independent variables-cough+fever+sore_throat+shortness_of_breath+head_ache.

i.e corona_result will depend upon cough,fever,sore_throat,shortness_of_breath and head_Ache.

Use data to be entire data and give family attribute to be "binomial"=>gives 1 out of 2 results only.

```
>model<-
```

```
glm(formula=corona_result~cough+fever+sore_throat+shortness_of_breath+head_ache,data=my_data1,family="binomial")
```

Use predict() method to predict the value based on the model.

```
>res<-predict(model,data=my_data1,type="response")
```

```
>res[1:5]#gives predicted result for first 5 records
```

```
>pred <- ifelse(res > 0.5, 1, 0)#gives 1 if res>0.5 else gives 0
```

```
>pred #give:
```

```
R4.1.1 ~ ~/
> #Logistic regression model:
> model<-glm(formula=corona_result~cough+fever+sore_throat+shortness_of_breath+head_ache,data=my_data1,family="binomial")
> res<-predict(model,data=my_data1,type="response")
> res[1:5]#gives predicted result for first 5 records
1 2 3 4 5
0.02723615 0.05534408 0.10903624 0.05534408 0.05534408
> pred <- ifelse(res > 0.5, 1, 0)
> pred
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

FINDING CONFUSION MATRIX:

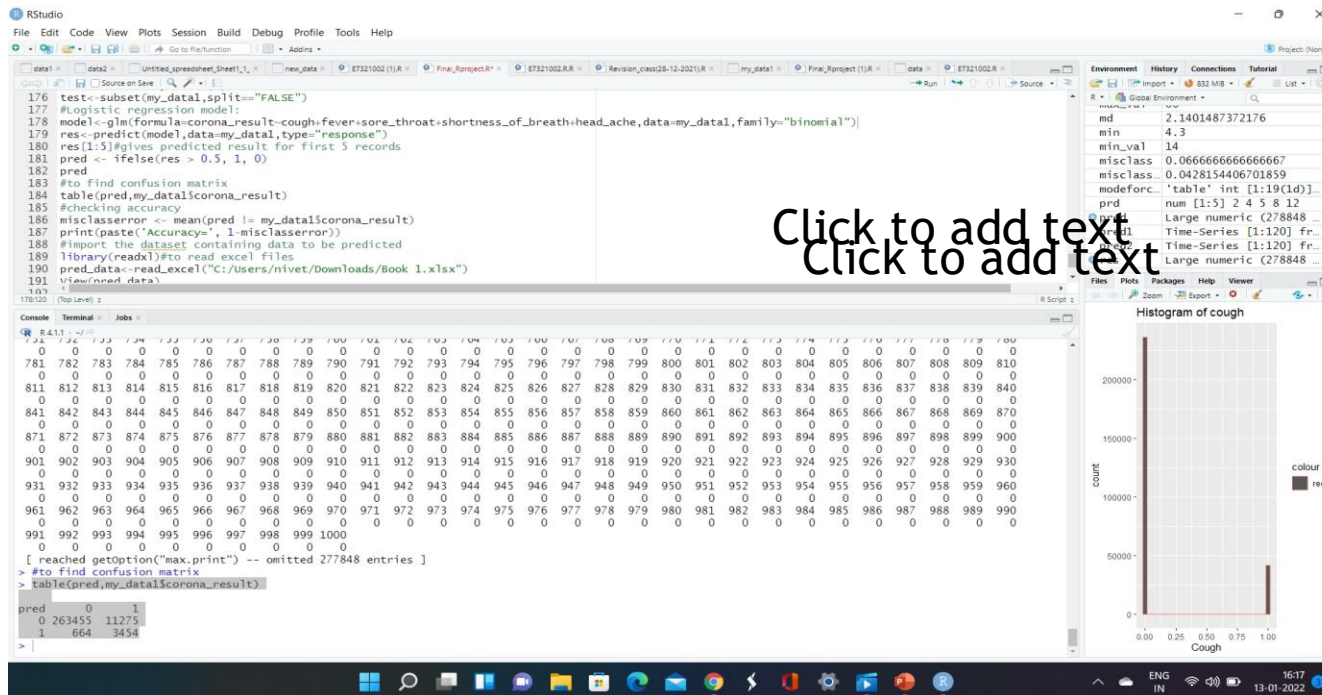
Confusion matrix tells you how much values are predicted correctly.

Method used:

`table(Predicted_Value,Actual_Value)`

Code:

`>table(pred,my_data1$corona_result)`



INTERPRETATION:

We can see that 263455 values are correctly predicted as 0 to be 0 and 3454 values are predicted 1 to be 1 correctly.

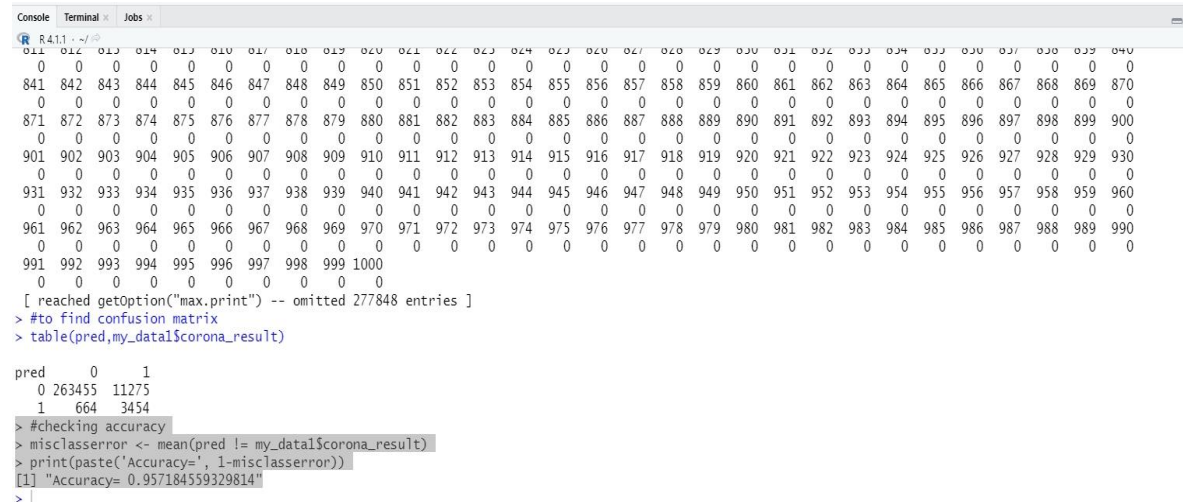
CHECKING ACCURACY PF PREDICTION:

First find misclasserror, which is got by taking mean of predicted values which are not equal to the actual value.

```
> misclasserror <- mean(pred != my_data1$corona_result)
```

Accuracy is got by subtracting misclasserror from 1.

```
> print(paste('Accuracy=', 1-misclasserror))
```



```
Console Terminal Jobs
R 4.1.1 ~ /
011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
991 992 993 994 995 996 997 998 999 1000
0 0 0 0 0 0 0 0 0 0
[ reached getOption("max.print") -- omitted 277848 entries ]
> #to find confusion matrix
> table(pred,my_data1$corona_result)

pred      0      1
0 263455 11275
1    664   3454
> #checking accuracy
> misclasserror <- mean(pred != my_data1$corona_result)
> print(paste('Accuracy=', 1-misclasserror))
[1] "Accuracy= 0.957184559329814"
> |
```

We get accuracy to be 0.957184559329814=>95.7% data are predicted correctly=>Model works very well.

Import the dataset containing the new data. The new dataset should not have the dependent variable column.

[illegible]

USE predict() method to predict for new data:

Inside predict() method, assign the name of new dataset to "newdata" attribute.

CODE:

```
>predict(model,newdata=pred_data,type="response")
```

OUTPUT:

0.9965705

INTERPRETATION:

We get 0.9965705 which is approximately equal to 1=>The person has been tested covid positive.

```
Console Terminal Jobs
R 4.1.1 ~ /
[1] Reached getOption('max.print') -- omitted 277646 entries ]
> #to find confusion matrix
> table(pred,my_data$corona_result)

pred      0      1
0 263455 11275
1    664   3454
> #checking accuracy
> misclasserror <- mean(pred != my_data$corona_result)
> print(paste('Accuracy=', 1-misclasserror))
[1] "Accuracy= 0.957184559329814"
> #import the dataset containing data to be predicted
> library(readxl)#to read excel files
> pred_data<-read_excel("C:/Users/nivet/Downloads/Book 1.xlsx")
> View(pred_data)
> str(pred_data)
tibble [1 x 5] (S3: tbl_df/tbl/data.frame)
 $ cough      : num 0
 $ fever      : num 1
 $ sore_throat : num 1
 $ shortness_of_breath: num 0
 $ head_ache  : num 1
> predict(model,newdata=pred_data,type="response")#gives predicted value=0.9965705 for new data
1
0.9965705
>
```

TEAM MEMBERS:

NIVETHA C- E7121015

NIVETHA K- E7321002

MAHA LAKSHMI P- E7321003

BRINDA G- E7321004

KUSHITHA G- E7321007



► **THANK YOU**