**Set-I**

1. .(i) Write a function called kelvin\_to\_celsius() that takes a temperature in Kelvin and returns that

temperature in Celsius (Hint: To convert from Kelvin to Celsius you subtract 273.15).

**Input :**

kelvin\_to\_celsius <- function(kelvin) {

celsius <- kelvin - 273.15

return(celsius)

}

(ii) Write suitable R code to compute the mean, median ,mode of the following values

c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

**Input :**

x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

mean\_x <- mean(x)

median\_x <- median(x)

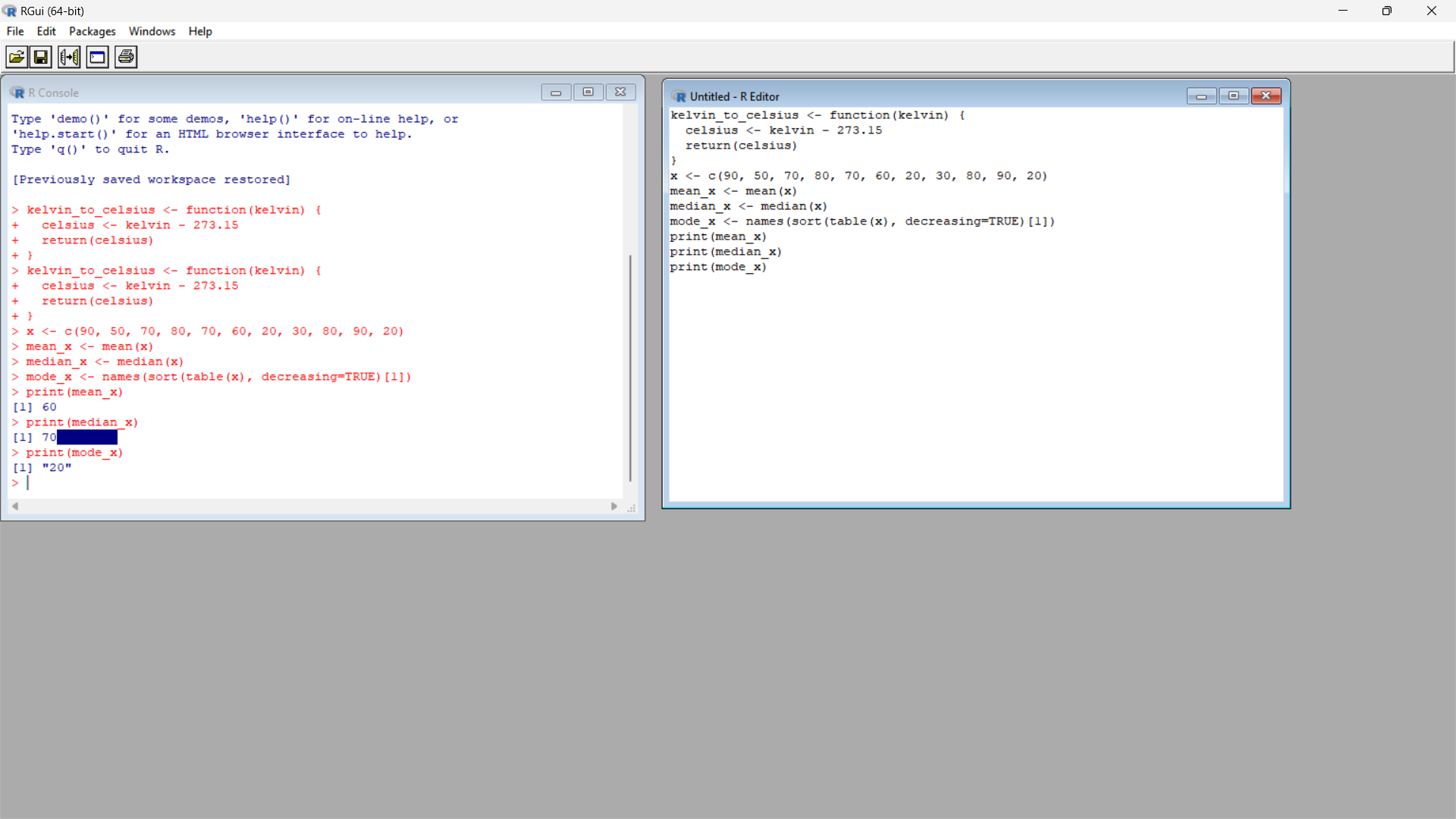
mode\_x <- names(sort(table(x), decreasing=TRUE)[1])

print(mean\_x)

print(median\_x)

print(mode\_x)

**Output :**



(iii) Write R code to find 2nd

highest and 3rd Lowest value of above problem.

**Input :**

x\_sorted <- sort(x)

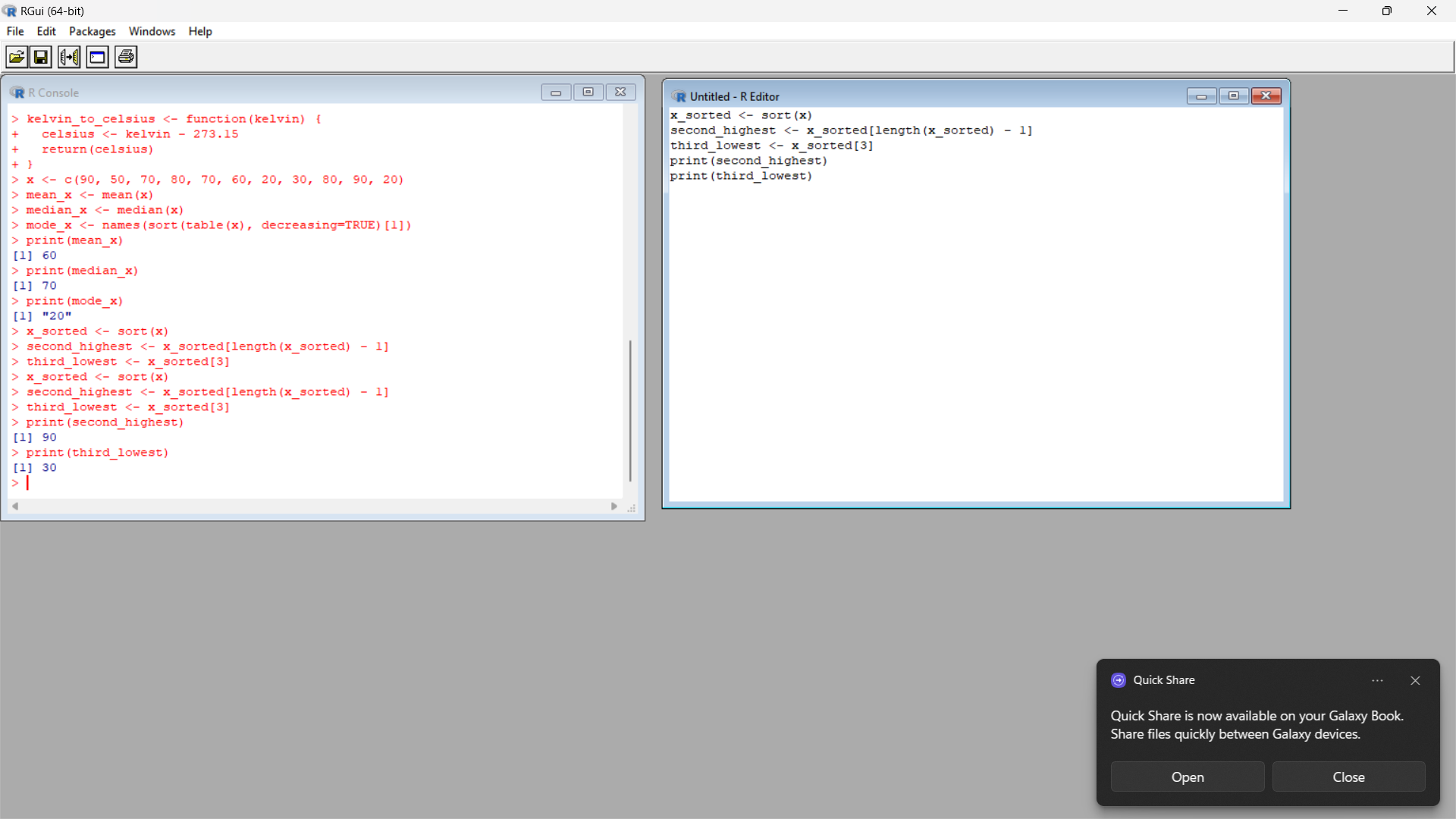
second\_highest <- x\_sorted[length(x\_sorted) - 1]

third\_lowest <- x\_sorted[3]

print(second\_highest)

print(third\_lowest)

**Output :**



2. Explore the airquality dataset. It contains daily air quality measurements from New York during a period

of five months:

• Ozone: mean ozone concentration (ppb),

• Solar.R: solar radiation (Langley),

• Wind: average wind speed (mph),

• Temp: maximum daily temperature in degrees Fahrenheit,

• Month: numeric month (May=5, June=6, and so on),

• Day: numeric day of the month (1-31).

1. Compute the mean temperature(don’t use build in function)

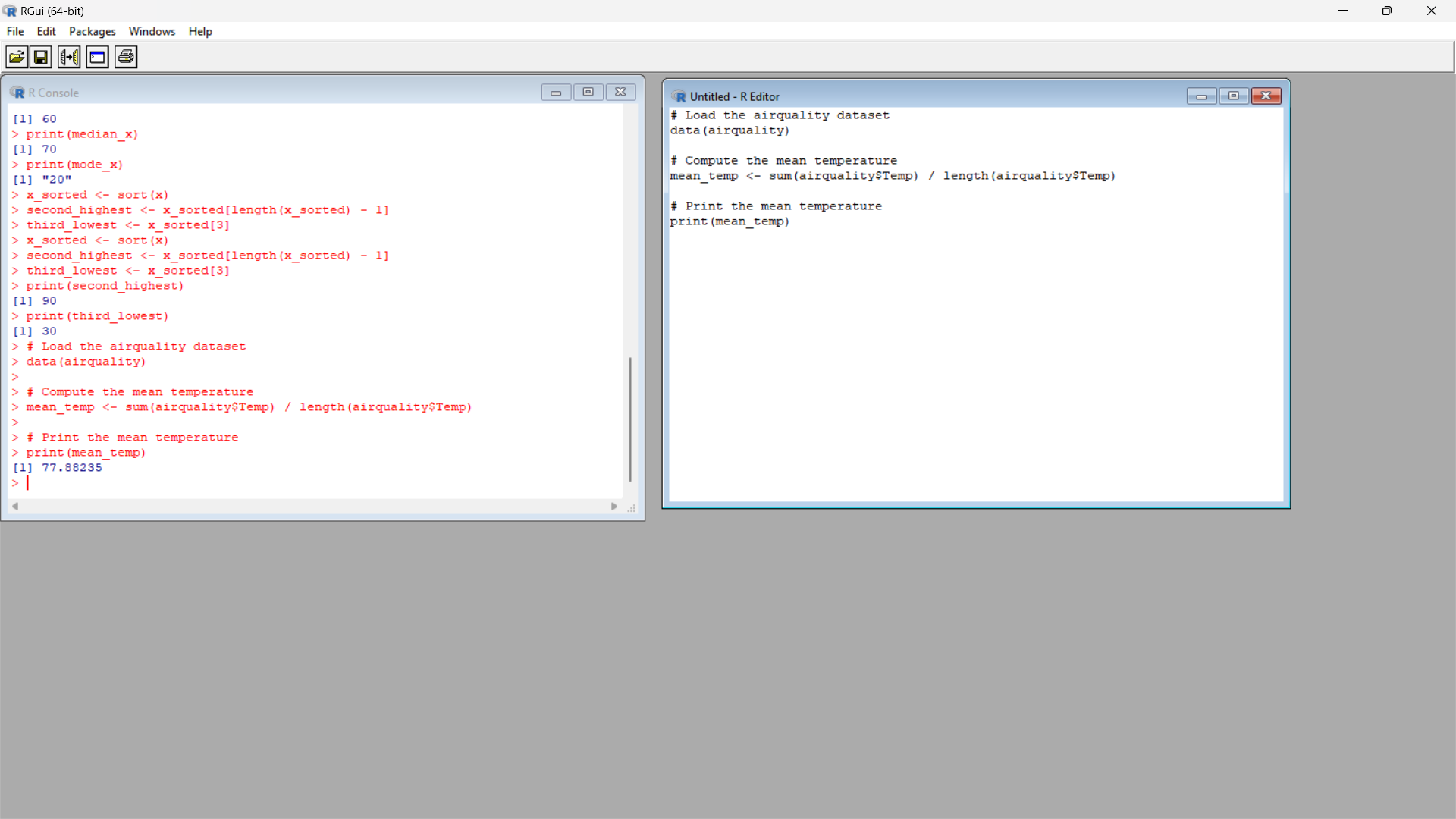
**Input:**

data(airquality)

mean\_temp <- sum(airquality$Temp) / length(airquality$Temp)

print(mean\_temp)

**Output:**



1. Extract the first five rows from airquality.

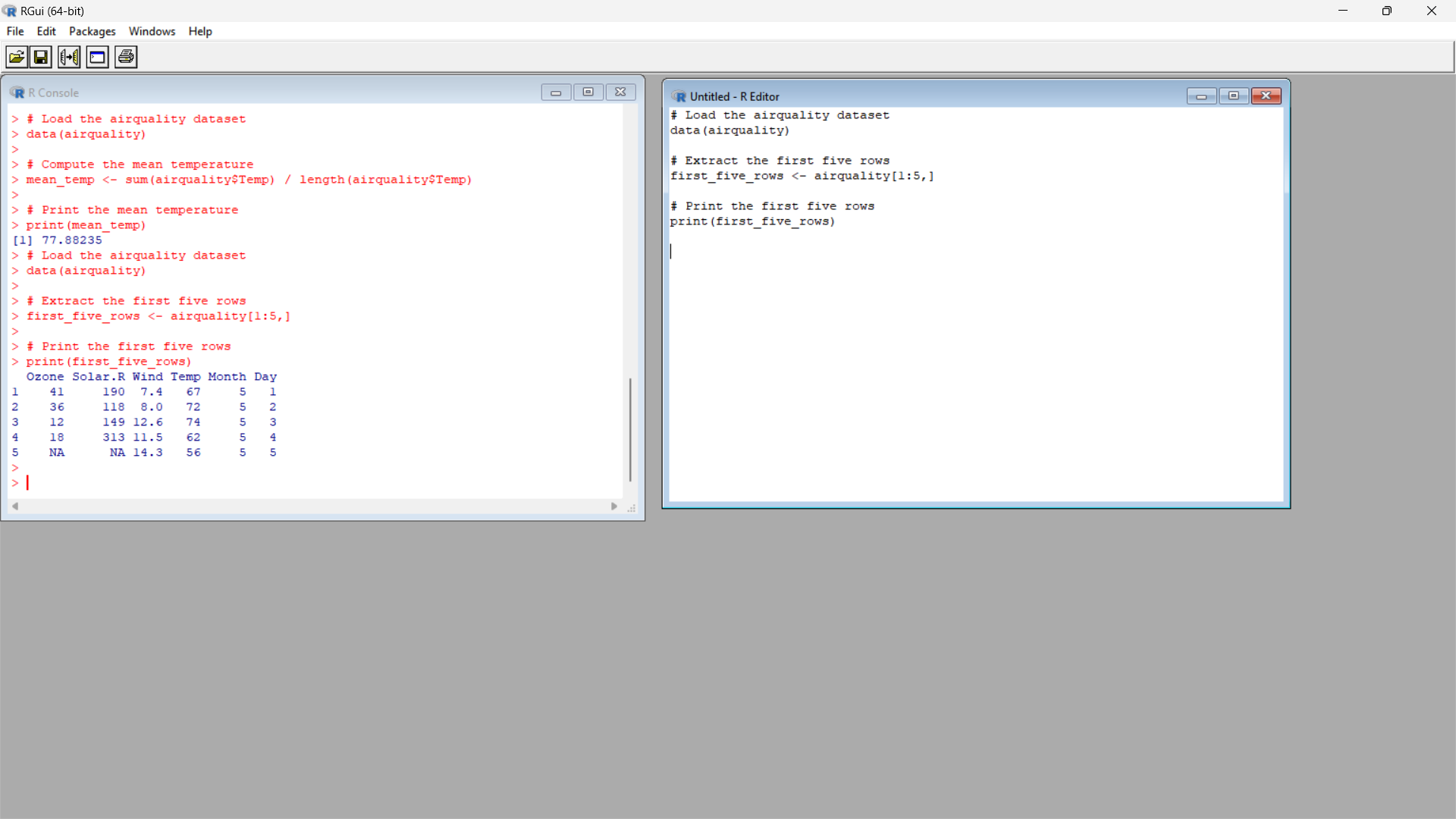
**Input :**

data(airquality)

first\_five\_rows <- airquality[1:5,]

print(first\_five\_rows)

**Output :**



1. Extract all columns from airquality except Temp and Wind

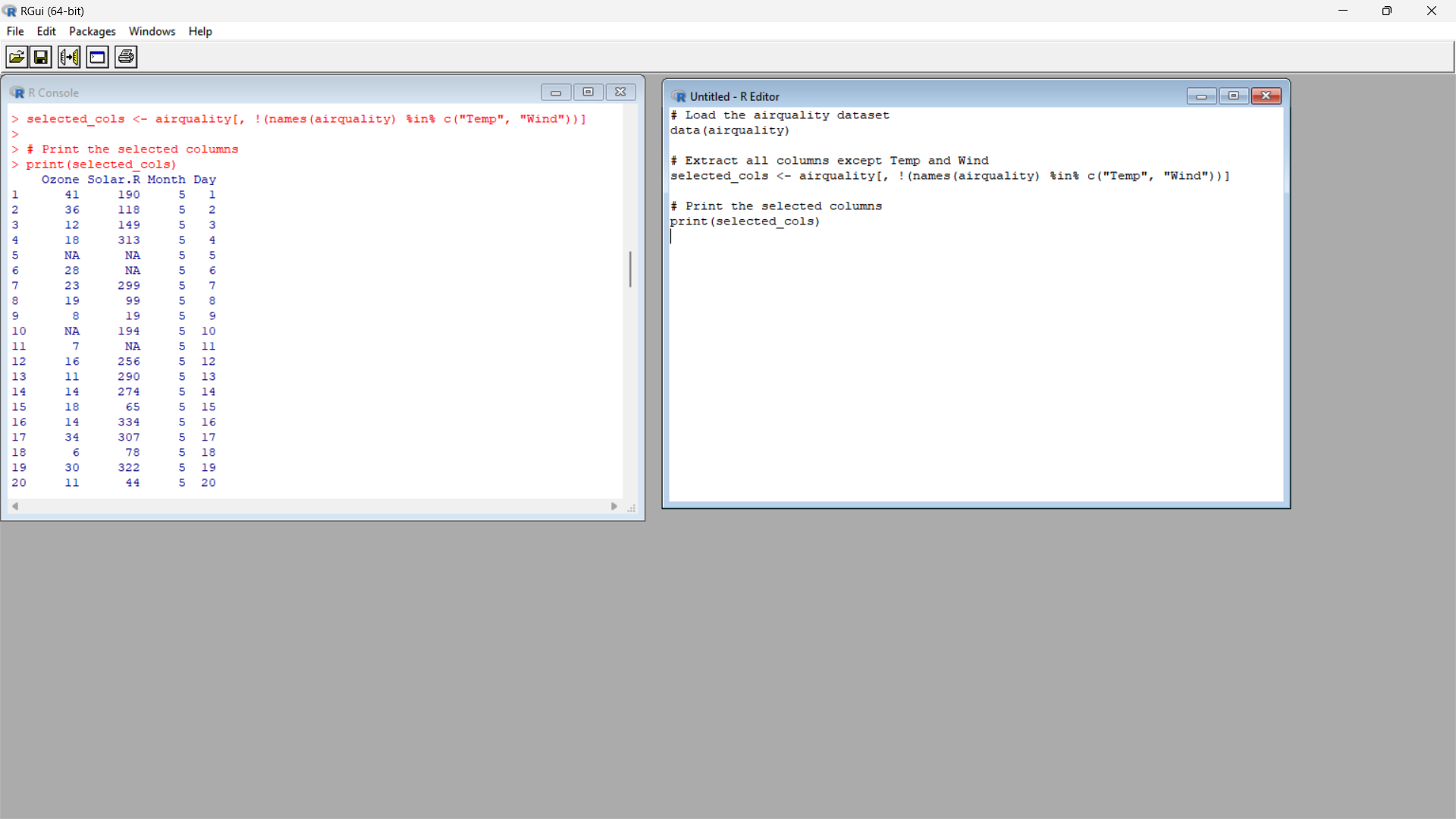
**Input :**

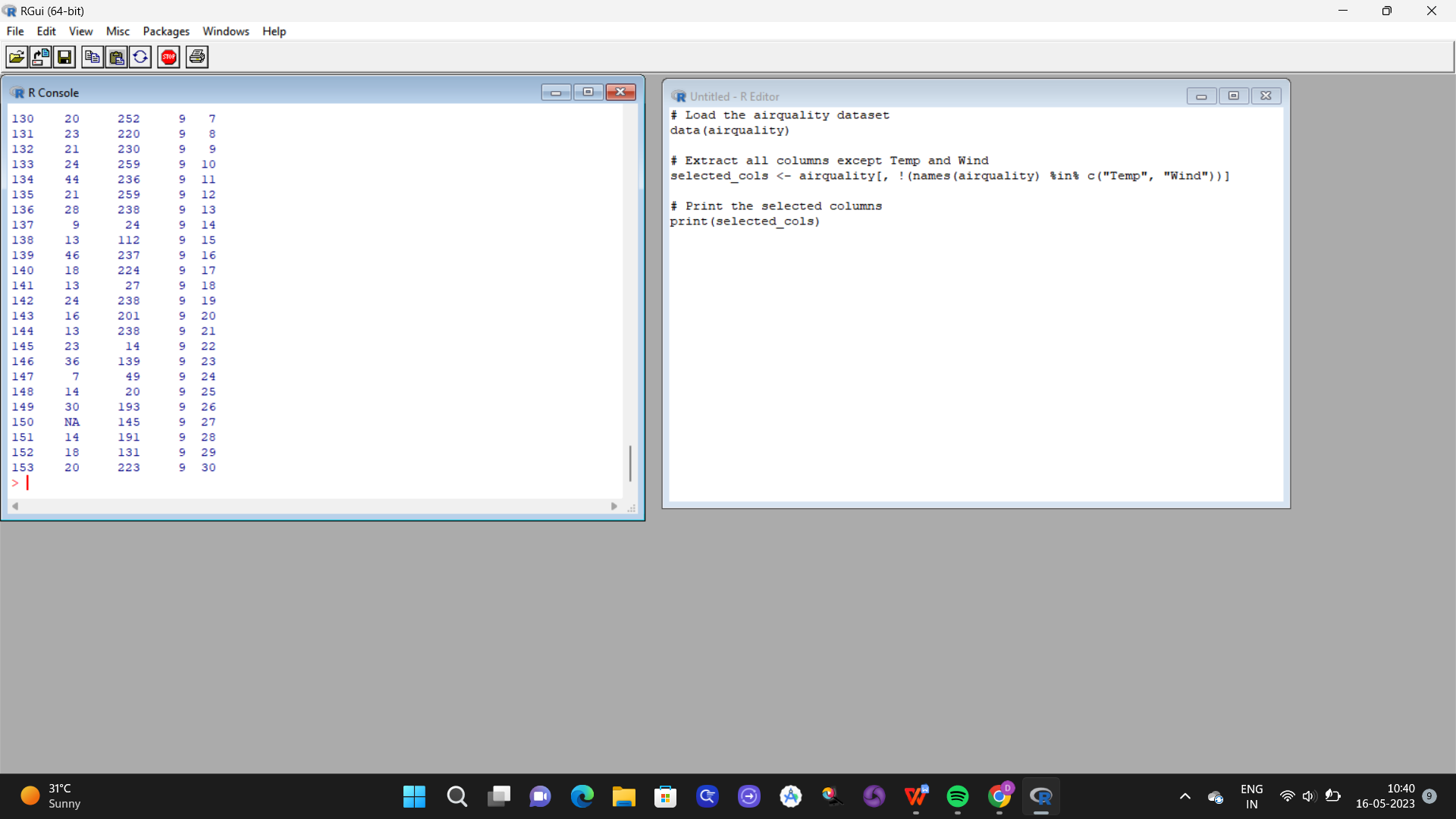
data(airquality)

selected\_cols <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]

print(selected\_cols)

**Output:**





1. Which was the coldest day during the period?

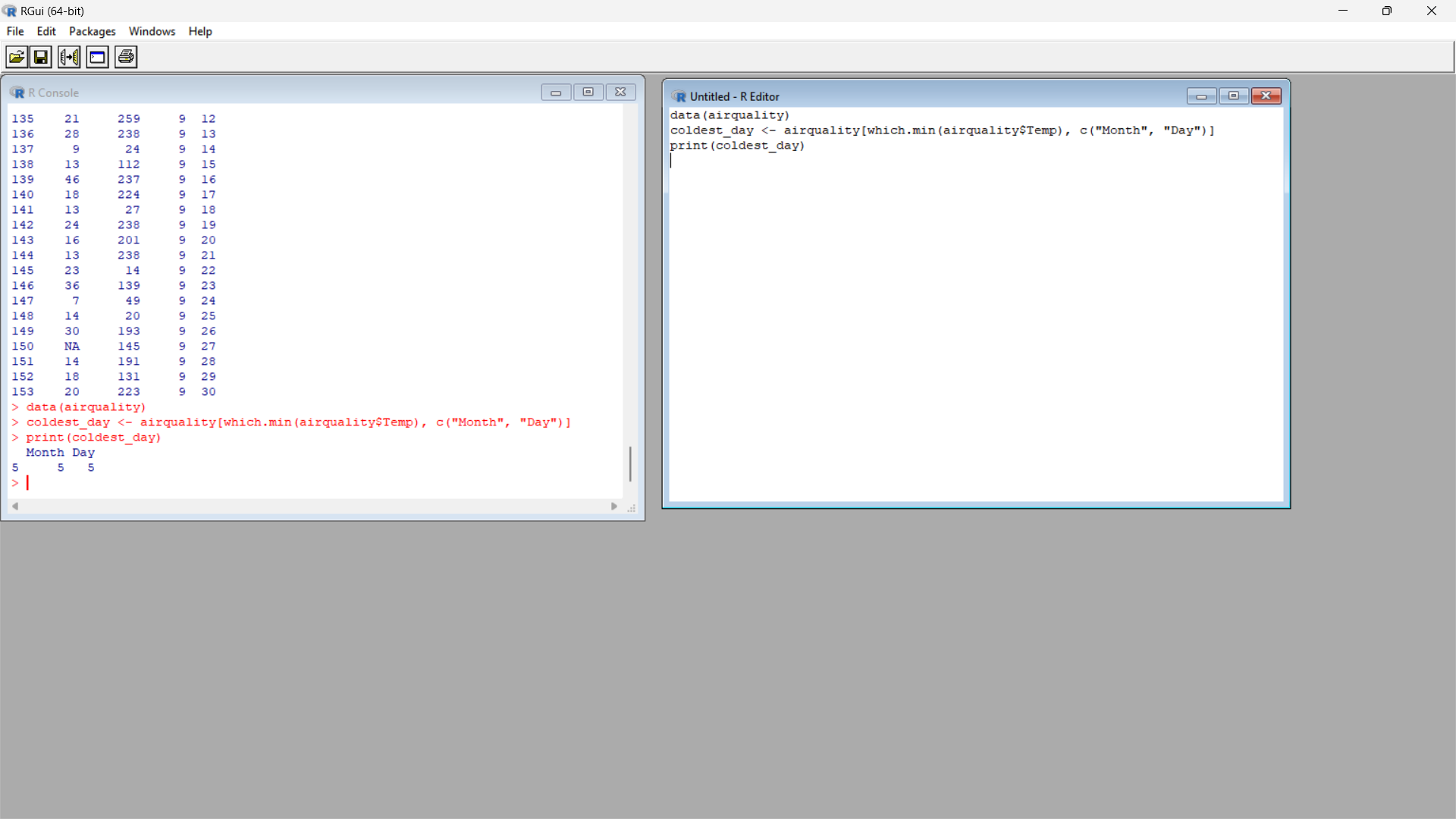
**Input :**

data(airquality)

coldest\_day <- airquality[which.min(airquality$Temp), c("Month", "Day")]

print(coldest\_day)

**Output:**



1. How many days was the wind speed greater than 17 mph?

**Input :**

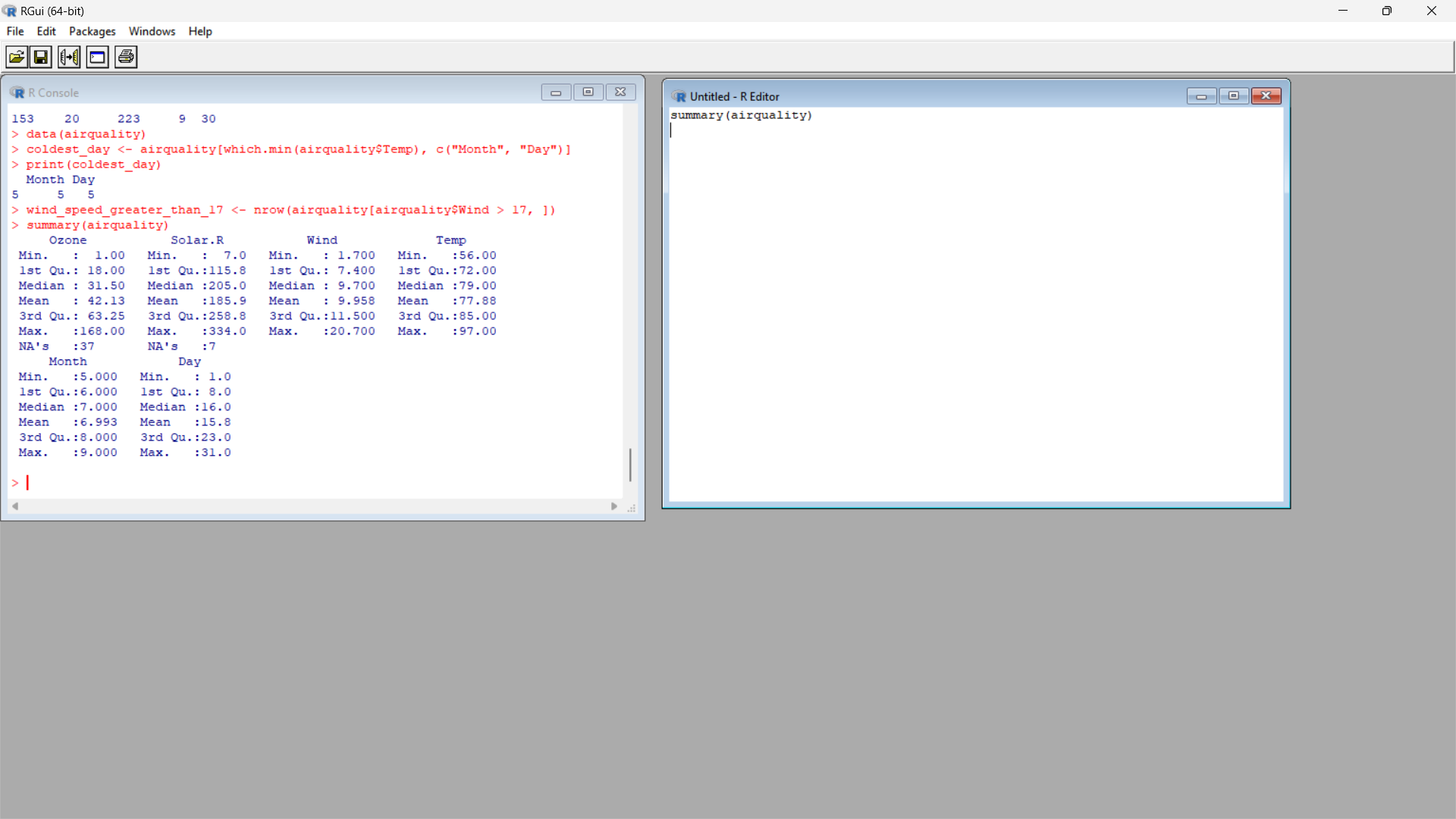
wind\_speed\_greater\_than\_17 <- nrow(airquality[airquality$Wind > 17, ])

1. (i) Get the Summary Statistics of air quality dataset.

**Input :**

summary(airquality)

**Output:**



(ii)Melt airquality data set and display as a long – format data?

**Input :**

library(reshape2)

airquality\_long <- melt(airquality, id.vars = c("Month", "Day"), variable.name = "Variable", value.name = "Value")

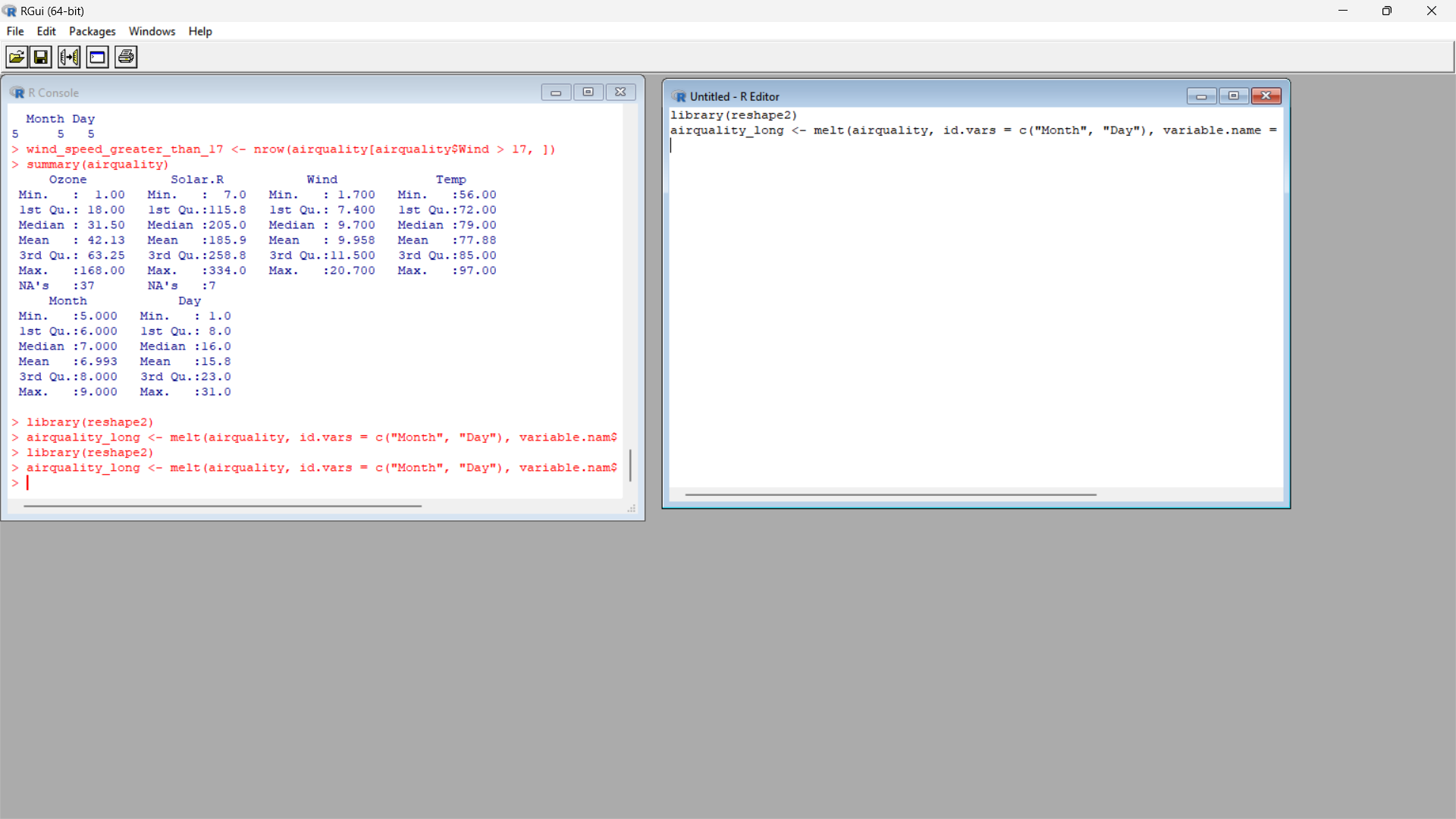
(iii)Melt airquality data and specify month and day to be “ID variables”?

**Input :**

library(reshape2)

airquality\_long <- melt(airquality, id.vars = c("Month", "Day"), variable.name = "Variable", value.name = "Value")

**Output:**



(iv)Cast the molten airquality data set with respect to month and date features

**Input :**

library(reshape2)

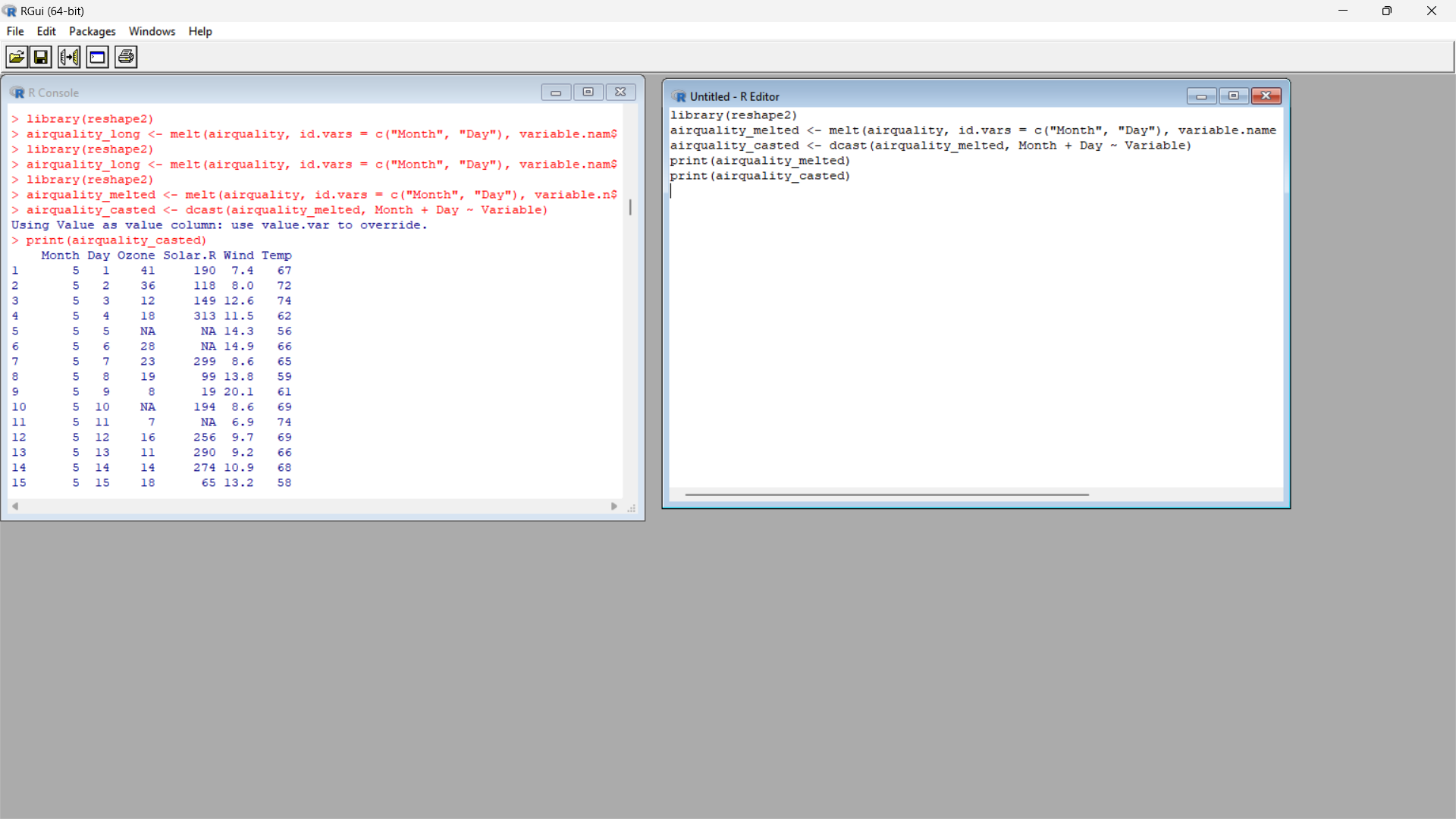
airquality\_melted <- melt(airquality, id.vars = c("Month", "Day"), variable.name = "Variable", value.name = "Value")

airquality\_casted <- dcast(airquality\_melted, Month + Day ~ Variable)

print(airquality\_melted)

print(airquality\_casted)

**Ouput:**



(v) Use cast function appropriately and compute the average of Ozone, Solar.R , Wind and

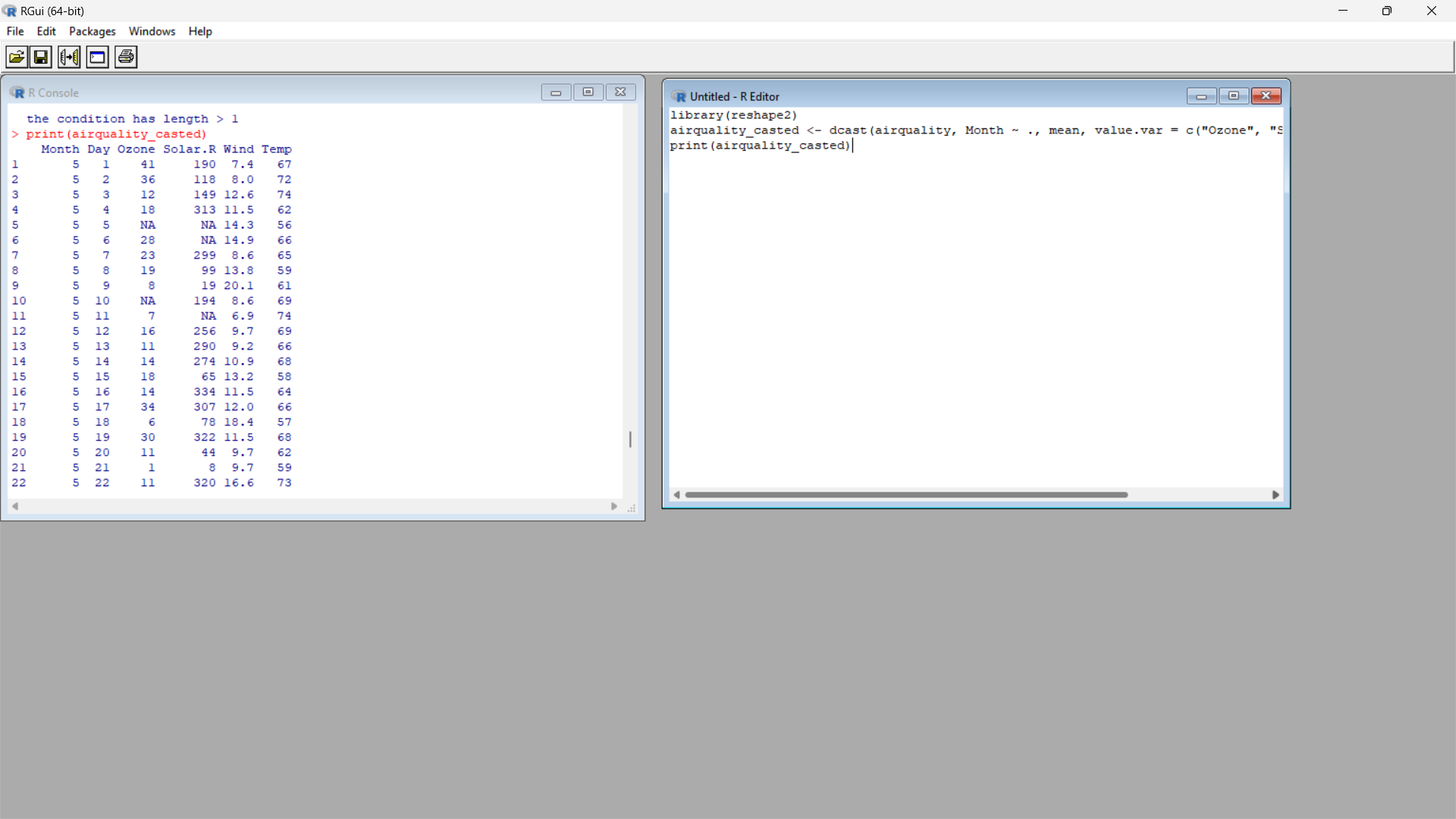
temperature per month?

**Input :**

library(reshape2)

airquality\_casted <- dcast(airquality, Month ~ ., mean, value.var = c("Ozone", "Solar.R", "Wind", "Temp"))

**Output:**



4.(i) Find any missing values(na) in features and drop the missing values if its less than 10% else

replace that with mean of that feature.

**Input :**

missing\_values <- colSums(is.na(airquality))

missing\_percentage <- missing\_values / nrow(airquality) \* 100

less\_than\_10\_percent <- missing\_percentage < 10

airquality\_dropped <- airquality[, less\_than\_10\_percent]

airquality\_filled <- airquality[, !less\_than\_10\_percent]

for (col in names(airquality\_filled)) {

if (sum(is.na(airquality\_filled[[col]])) > 0) {

mean\_value <- mean(airquality\_filled[[col]], na.rm = TRUE)

airquality\_filled[[col]][is.na(airquality\_filled[[col]])] <- mean\_value

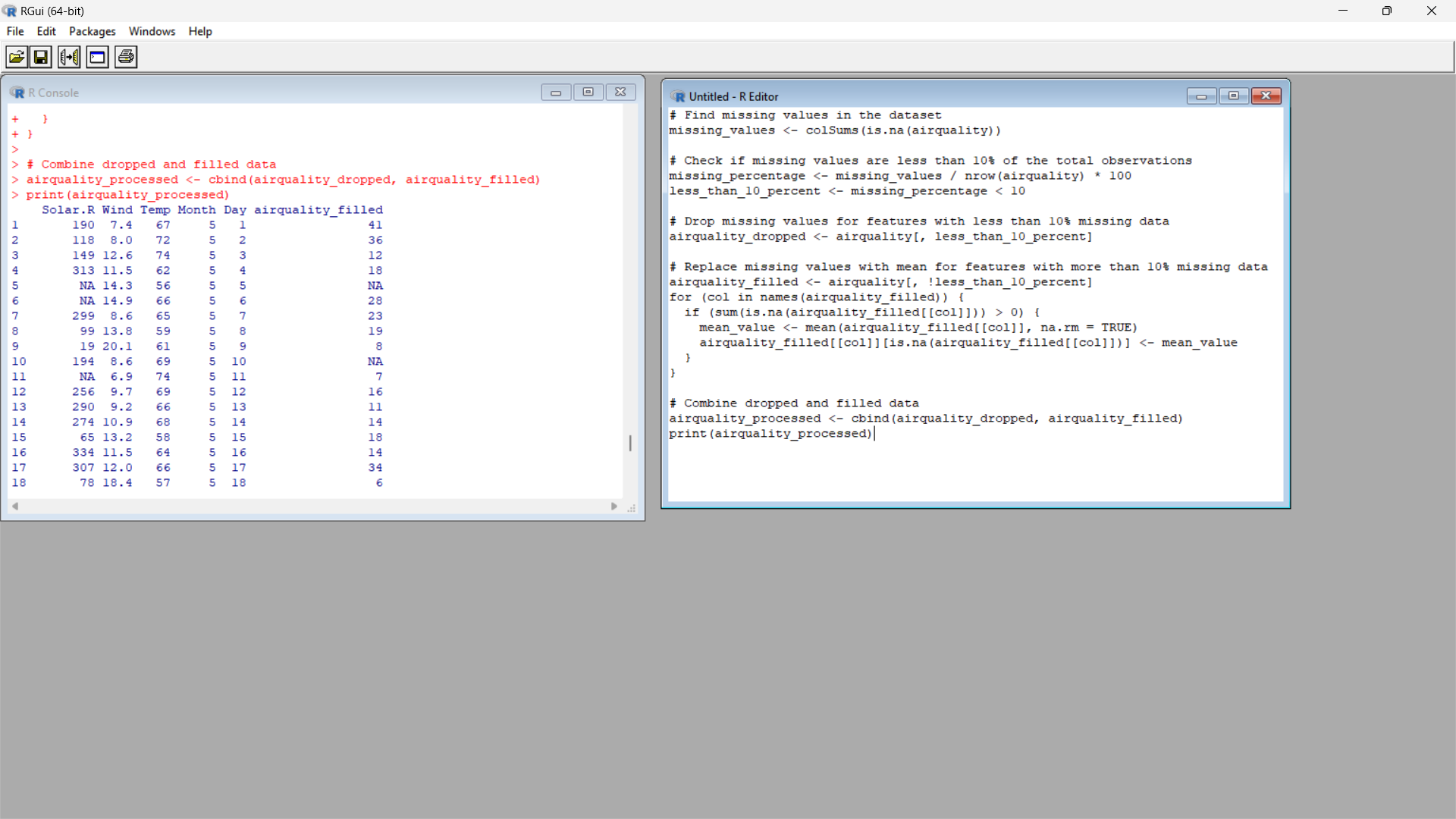
}

}

airquality\_processed <- cbind(airquality\_dropped, airquality\_filled)

print(airquality\_processed)

**Output:**



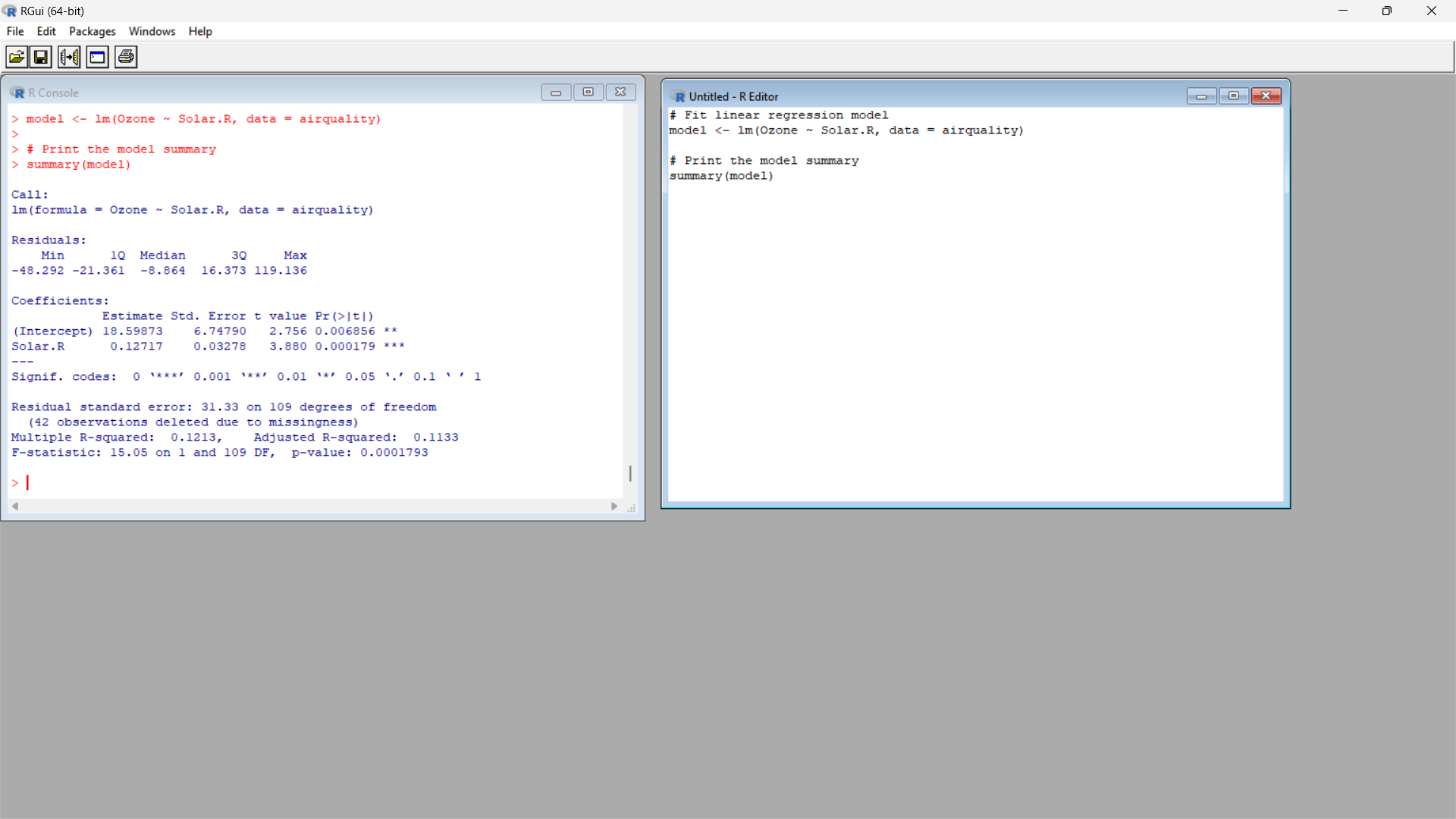
(ii) Apply a linear regression algorithm using Least Squares Method on “Ozone” and “Solar.R”

**Input :**

model <- lm(Ozone ~ Solar.R, data = airquality)

summary(model)

**Output:**



(iii)Plot Scatter plot between Ozone and Solar and add regression line created by above model

**Input :**

plot(airquality$Solar.R, airquality$Ozone, xlab = "Solar.R", ylab = "Ozone", main = "Scatter Plot")

abline(model, col = "red")

**Output;**

