**Module 4: Additional Exercises with Answers**

Clear your R workspace using the following command

rm(list = ls())

We use the "airquality” dataset for this exercise. The dataset is part of the latest distribution of R 3.4.1 or can be accessed from the “datasets” package which you can install using install.packages() command). I have also uploaded a copy on the course portal, under datasets, just in case (most likely you already have it in your base). To check the availability of the dataset simply type *summary(airquality)* into the R-studio console (case sensitive). If you don’t get an error message, it means that you have the dataset.

The following is the description of the dataset:

**Format**

A data frame with 154 observations on 6 variables.

[,1] Ozone numeric Ozone (ppb)

[,2] Solar.R numeric Solar R (lang)

[,3] Wind numeric Wind (mph)

[,4] Temp numeric Temperature (degrees F)

[,5] Month numeric Month (1--12)

[,6] Day numeric Day of month (1--31)

**Details**

Daily readings of the following air quality values for May 1, 1973 to September 30, 1973.

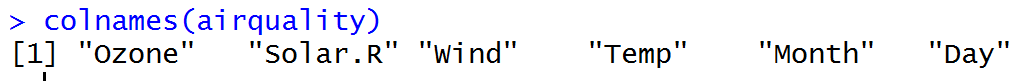
* **Ozone:** Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island
* **Solar.R:** Solar radiation in Langleys in the frequency band 4000–7700 Angstroms from 0800 to 1200 hours at Central Park
* **Wind:** Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport
* **Temp:** Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.

**Source**

The data were obtained from the New York State Department of Conservation (ozone data) and the National Weather Service (meteorological data).

Tasks:

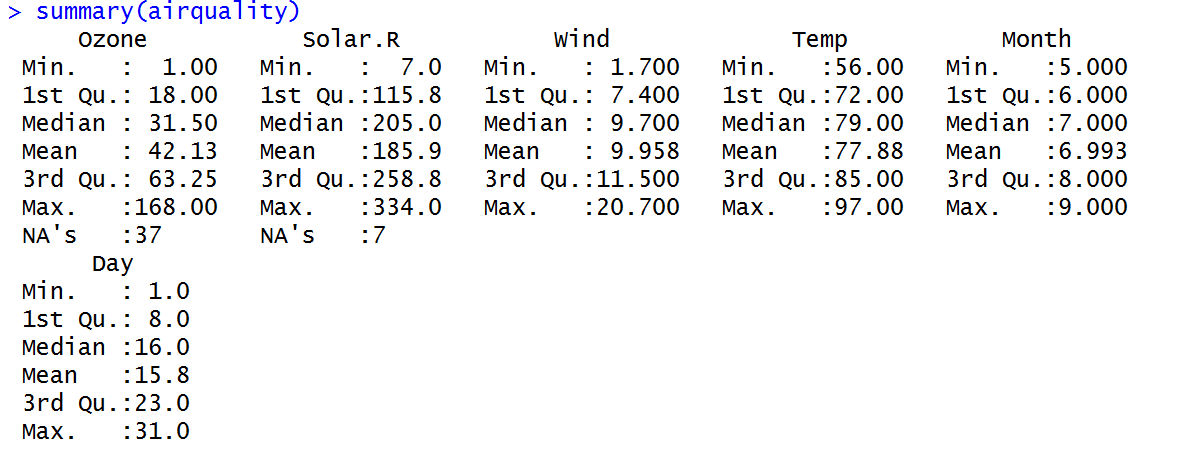
1. **What are the variable names in the “airquality” dataframe?**



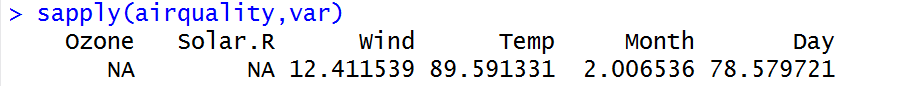
**2. How many observations (rows) are in the in the “airquality” dataframe?**



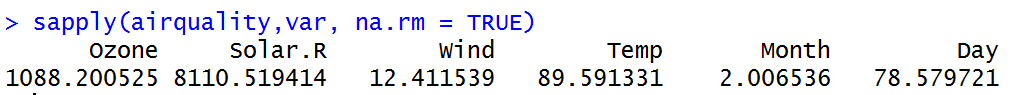
1. **What are the ranges and distribution of the variables in the “airquality” dataframe?**



1. **Calculate the variance of all variables in the “airquality” dataframe using the sapply()?**

****

**We are getting NA values for numerical calculations even if we have a single missing value. To avoid this, we use the rm.na=TRUE option:**

****

**This removes the NA (missing) values from the caluculations but doesn’t remove them permanently from the data frame.**

1. **Select the days where the Temperature was above 77 and store the records in a new dataframe called “High\_Temp”. Do this using the base R and “dplyr” package.**

With Base R:



With dplyr:

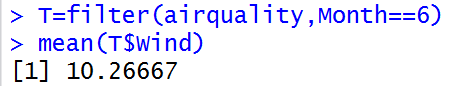
#note make sure that dplyr is installed otherwise install using

install.packages('dplyr')

#while you are connected to the internet.



1. **What is the average wind speed for days in June? Do this in an step by step fashion (i.e. store the filter results in a temporary intermediate dataframe, T, first)**

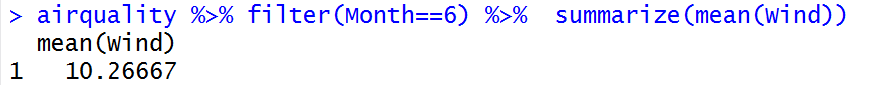
****

1. **Repeat the above question without using an intermediate temporary variable.**

with filter function from dplyr

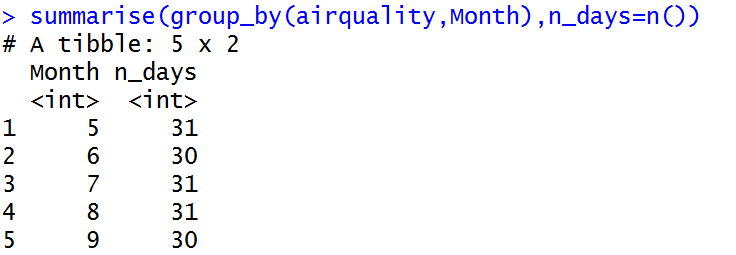
****

#Even more elegant solution with piping would be

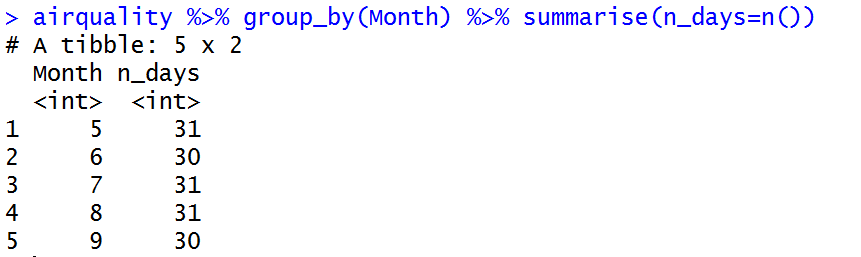


1. **How many days in each month is reported in the dataset?**

Best is to use group\_by function, and summarise with count(i.e. n()) function.

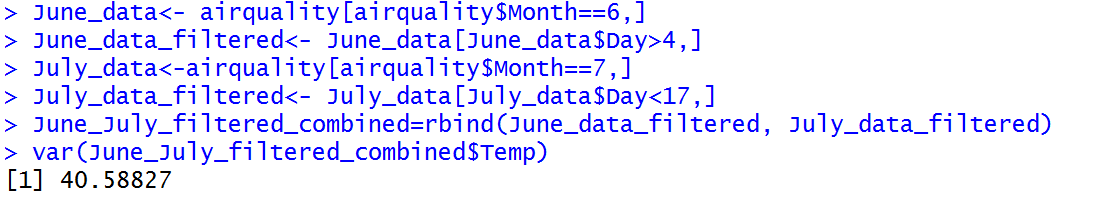
****

Or using piping

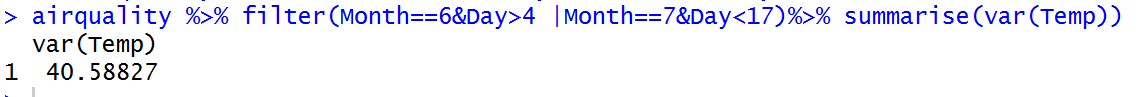


1. **What is the variance of the temperature for days between 5th June and 16th July?**

Solution 1: Many intermediate dataframes

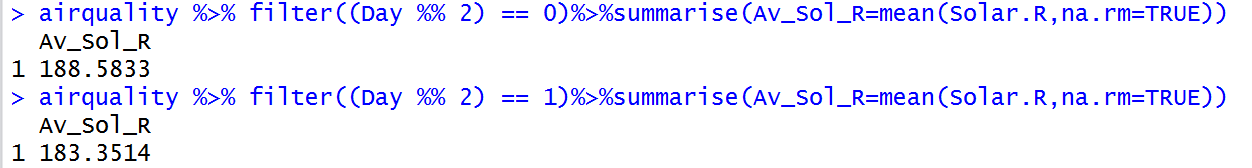


Solution 2: More elegant and efficient

****

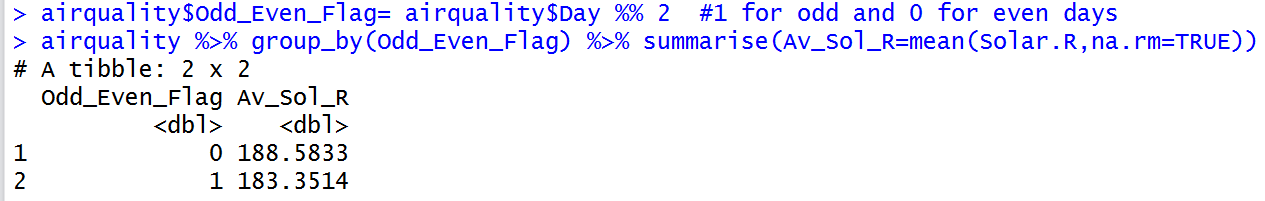
1. **Can you see a big difference between the average Solar Radiation on even days and odd days?**

In R, **x %% y** is themodulus (x mod y). Example: 5%%2 is 1. For even days the modulus will be zero and for odd days it will be 1. We can use the filter function to select the even days and calculate the mean of Solar.R for those records and do the same for odd days.

****

1. **Solve question 10 by first creating a new variable called “Odd\_Even\_Flag” in the dataframe and then use the Group\_by and summerise functions to calculate the mean of the Solar Radiation for each group.**

First, we create the new variable in the dataframe and then we use group\_by()and summerise () functions.

****