# Week 1 Assignment Write-Up

By: Zach Adair

Regis University

MSDS 692: Exploratory Data Analysis

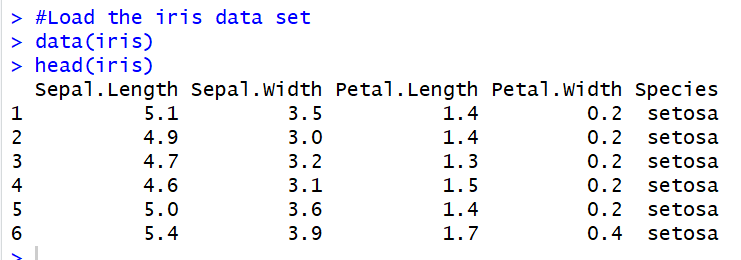
9/5/2019

## Introduction

The first assignment of MSDS 692 is designed to learn about some basic R commands that are critical for a quality EDA. The first data set will be with the Iris data set that is preloaded into RStudio. The second part I will input a dataset with the heights and weights of women measured.

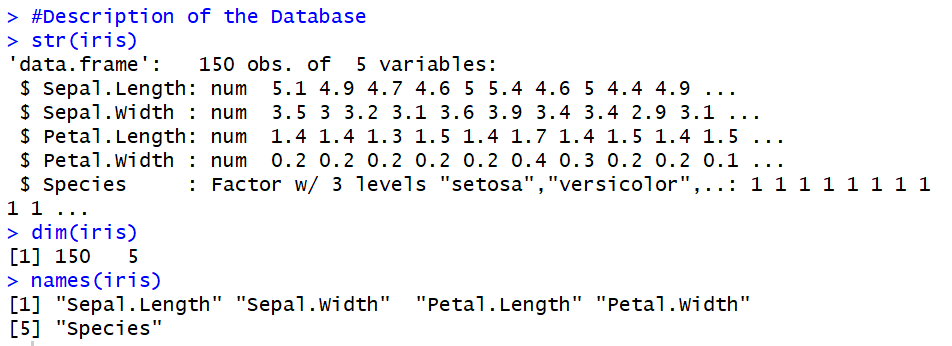
## Iris Data Set

Start by loading the data set and take a quick look at the data set using the head function.



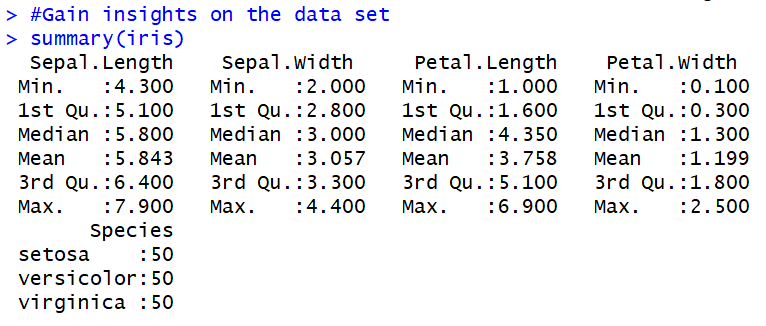
We see that the dataset is loaded from R and we can look at some of the data thanks to the head function.

Now let’s check out the description of the data set.



The three commands used here were str, dim, and names. Str gives some information about the data frame of iris. We know from this function that there are 150 observations with 5 different attributes and from those attributes or columns we see Sepal.Length, Sepal.Width, Petal.Width, Petal.Length and Species. With all but Species being numeric values and Species being a 3 level factor with those levels being virginica, setosa and versicolor. The dim command lets us know that the data set has 150 rows and 5 columns. The names command lets us know the names of our different attributes which I listed earlier.

Now I will gain insights on the dataset by using the summary command.

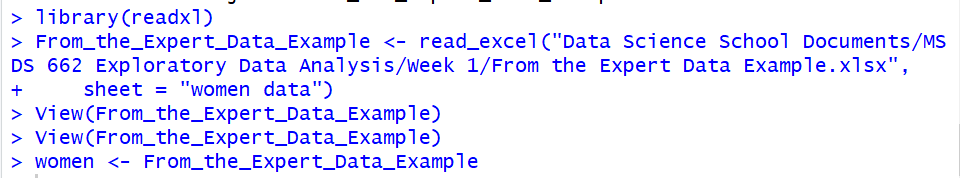


The summary command gives insight on each of the columns of the data set, for the numeric data sets we will get 6 stats: minimum, 1st quartile, median, mean, 3rd quartile, and maximum. In this situation we know the factorial is of 3 degrees and we get to see the number of records for each is 50.

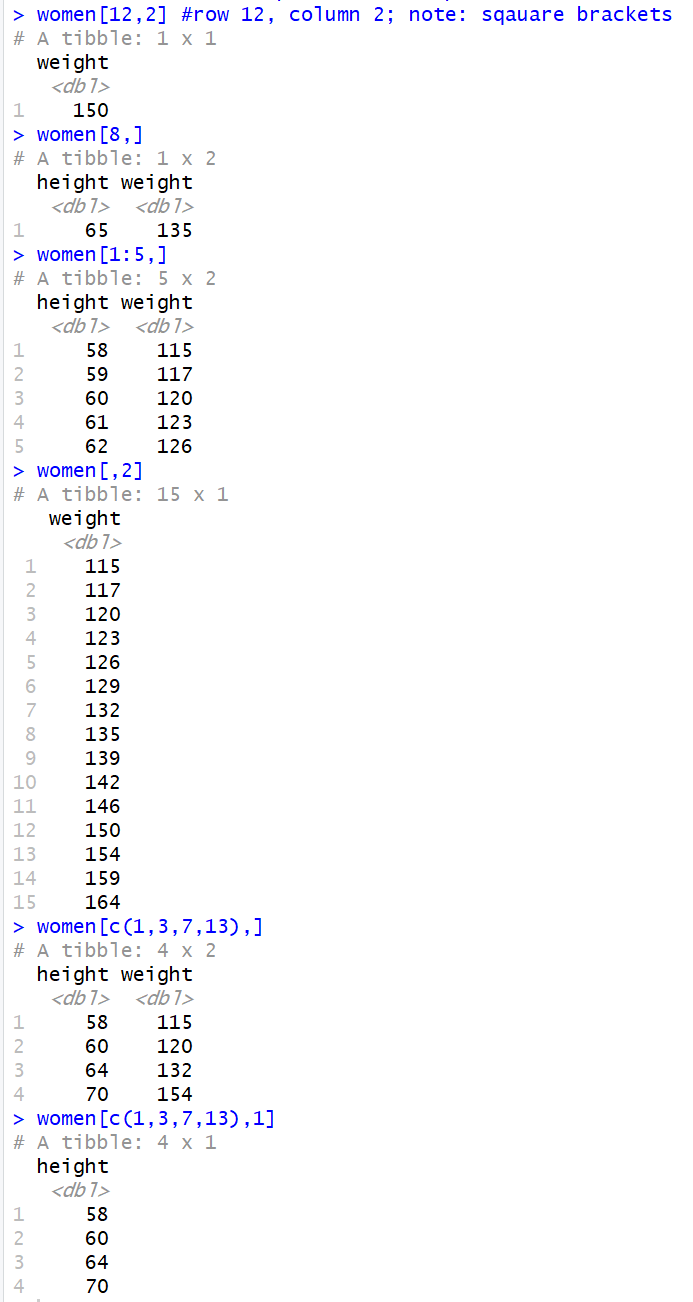
These commands are great for finding some initial information about data sets, but let’s do some more commands on a different data set.

## Women Measurements

To load this data, I imported it from an excel datasheet.



Now one way to look at the values with the data besides the commands we were using before with the Iris data set is to look at it by either the attribute or the specific value within the data frame. These commands will show how this is done in RStudio.



As you can see from the first command, we can pull out a specific value by designating a specific row and column. We can also do an entire row by picking the row and having a comma before the ending square bracket. We can grab a run of rows by using a colon to determine which rows you want to grab and then a comma before the square bracket again to get all the columns for that data. A user can also grab an entire column by putting the column before the number of column they want to isolate. A user can concatenate rows together by using the c function within the square bracket, picking the rows they want and then putting a comma just before the closing square bracket. Finally a user can also concatenate values from a specific column. By designating the column number you want after designating the rows to concatenate the R command will return only values from that specific row.

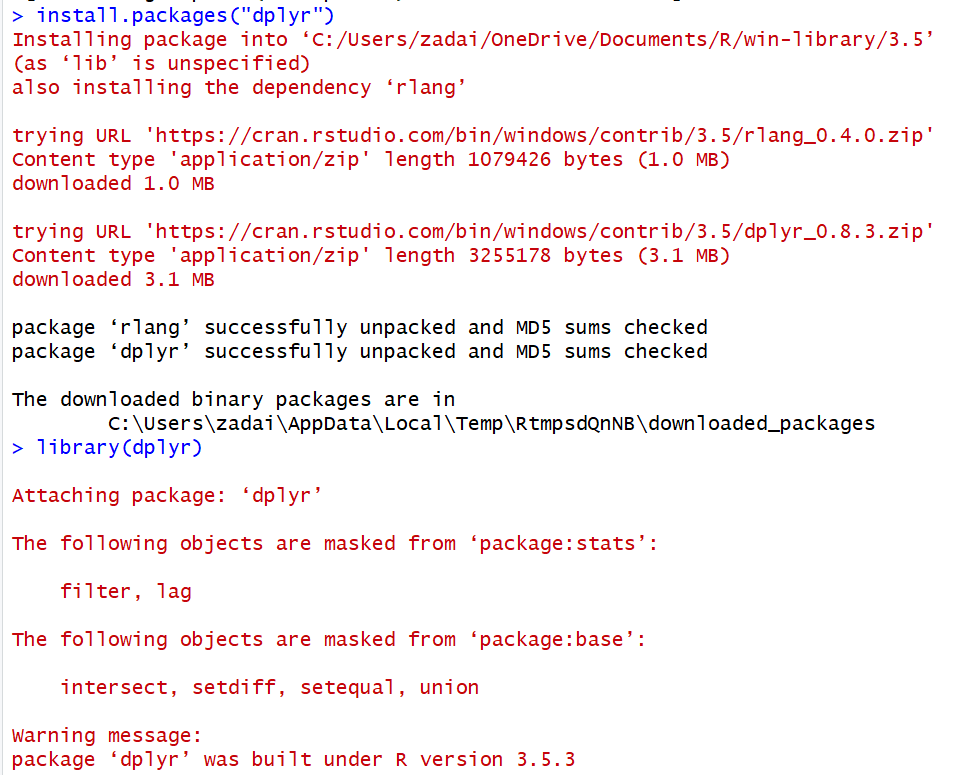
These kinds of commands can be great for finding out more about different individual attributes or entities within a data set, being able to isolate data in this way can help a user figure out even more about a data set by breaking it down in a multitude of ways than they would be able to with only being able to analysis the data set as a whole.

## Chicago Pollution Data Exploration

The Chicago pollution data set is a set that has information that measures the city pollution numbers daily starting on January 1st, 1987. An EDA analysis using the RStudio library dplyr will help with the understanding of this data.

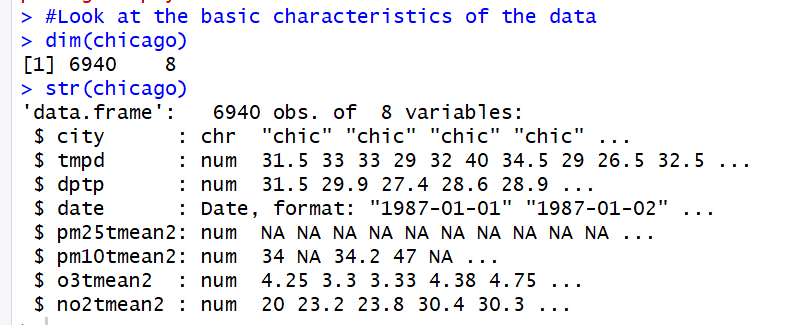
## Load the Data

Load the library and the Chicago data.

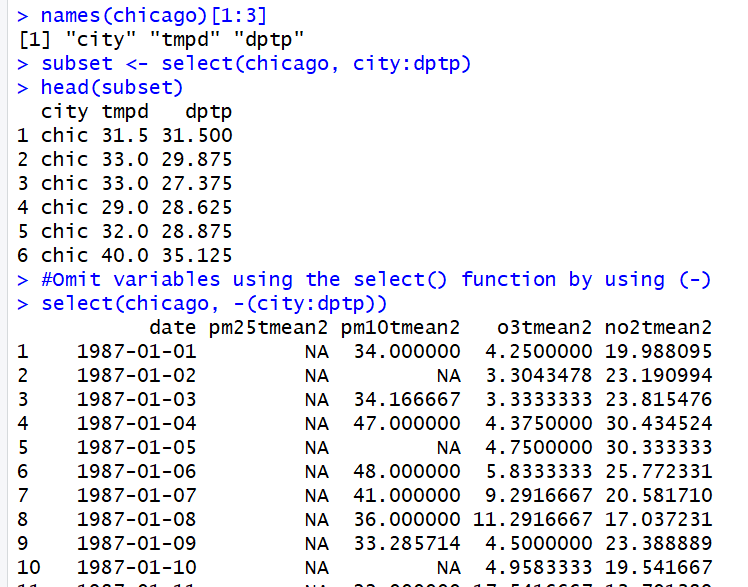


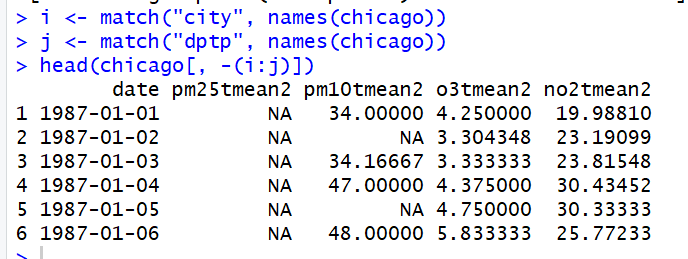
## Explore the Data

Now that the data and the dplyr package are in the fold we can really start to explore the data.

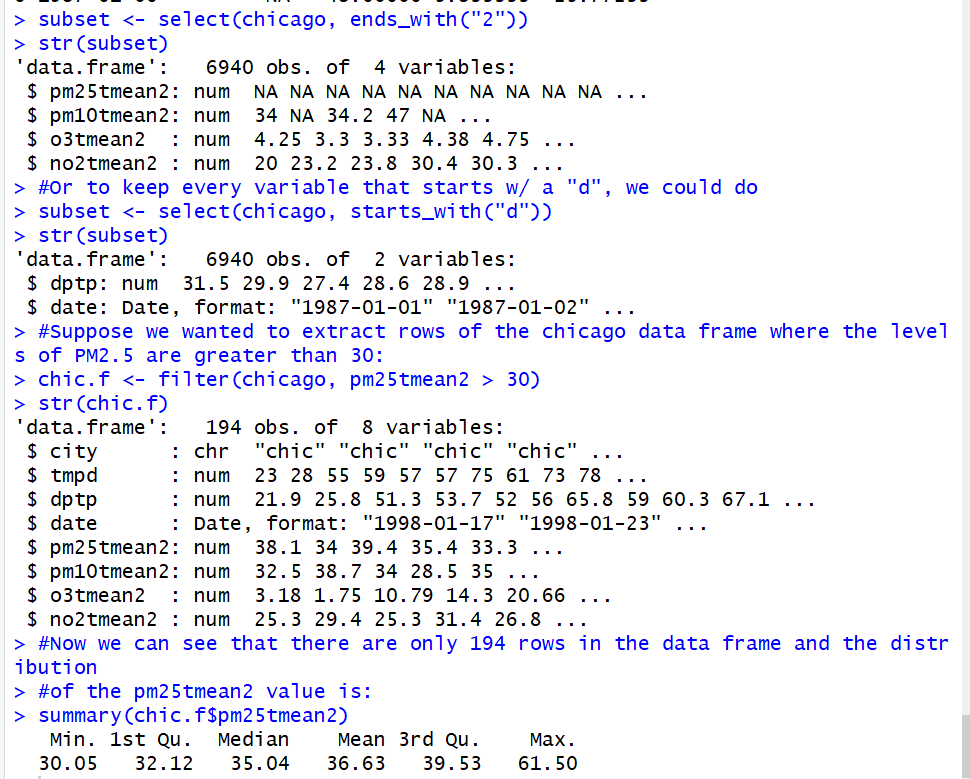


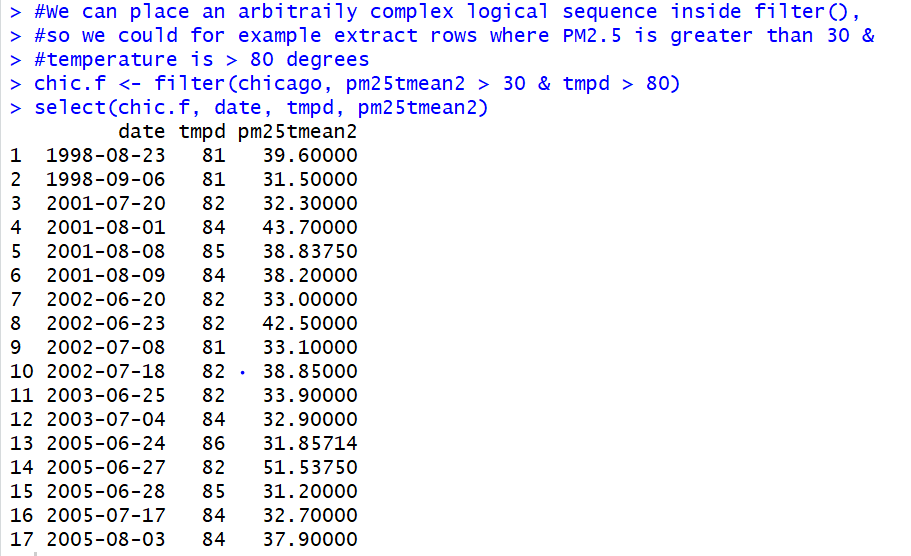
Next, we will look to create a subset of the data using the columns city, tmpd and dptp then use the select() command.





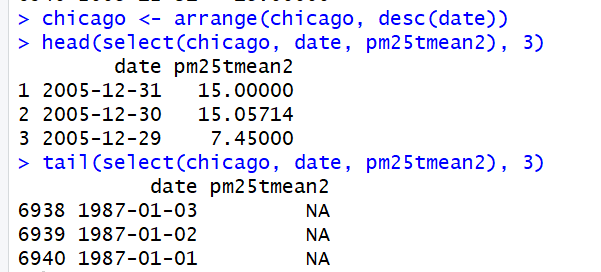
Next, let’s see what happens using the filter() command.



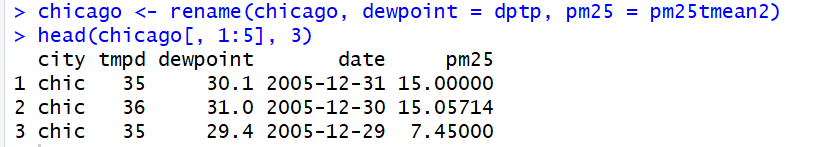


Now let’s see what the arrange() function will do for this Chicago pollution dataset.

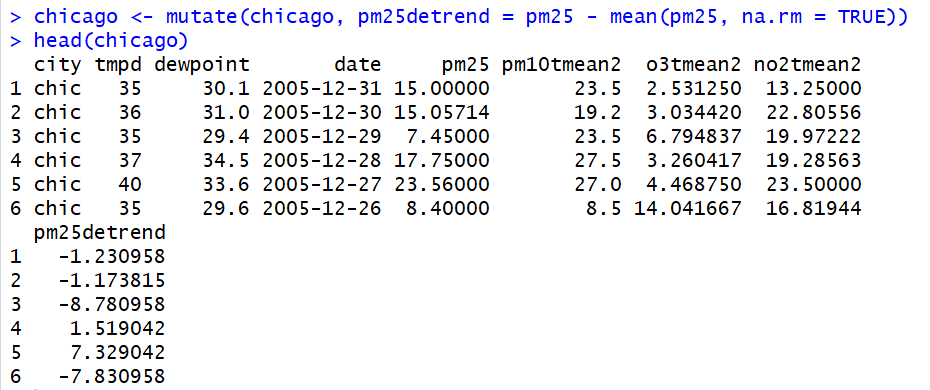
## 

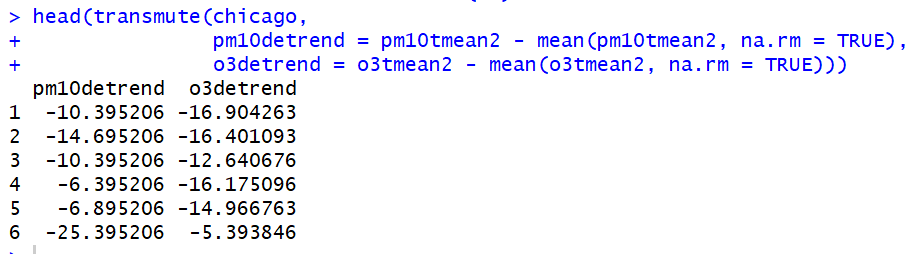


The rename() function will be the next function from the dplyr library that will be used on this data.

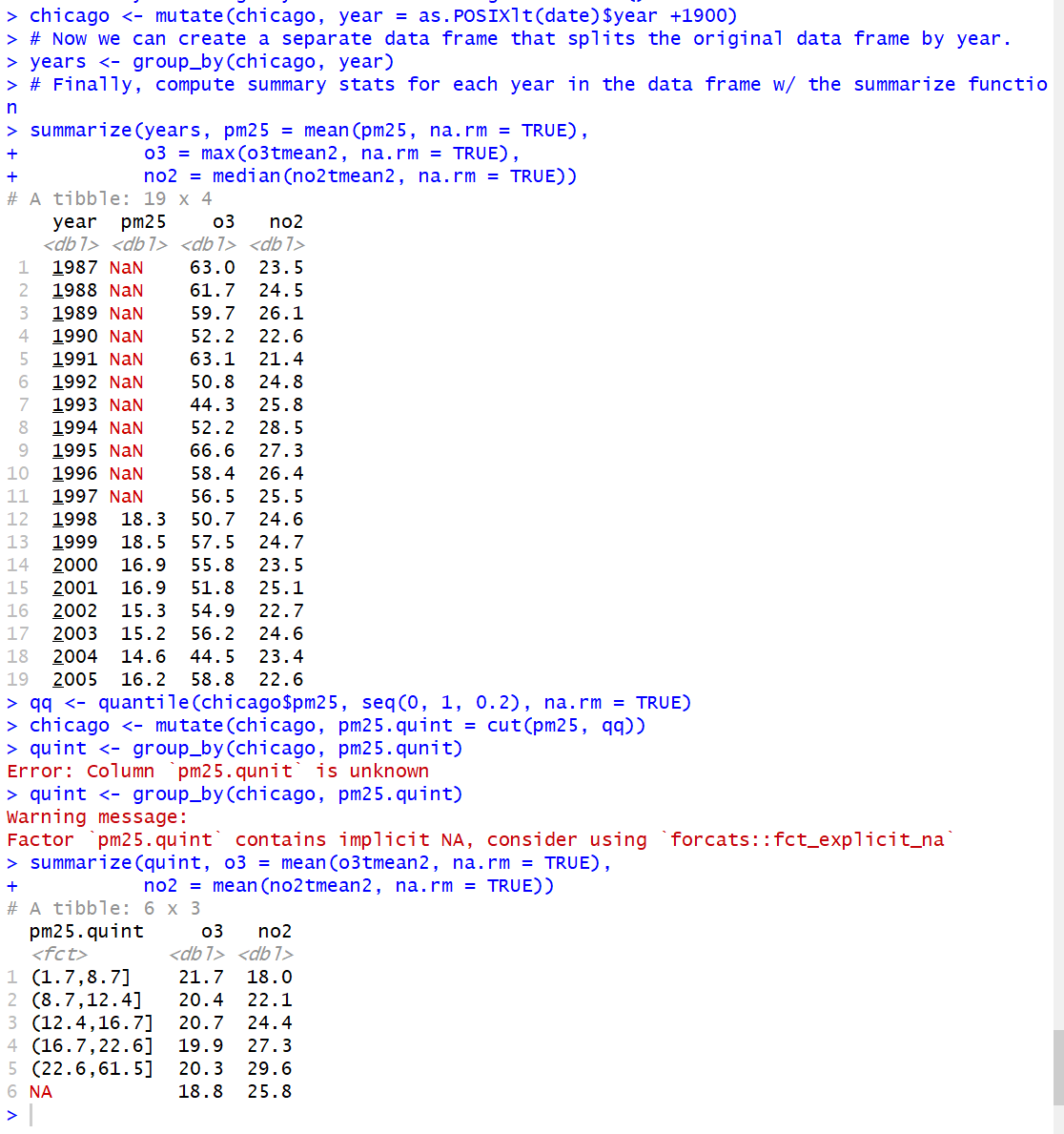


Next, let’s use the mutate() function from the dplyr library and checkout what it does to the data.



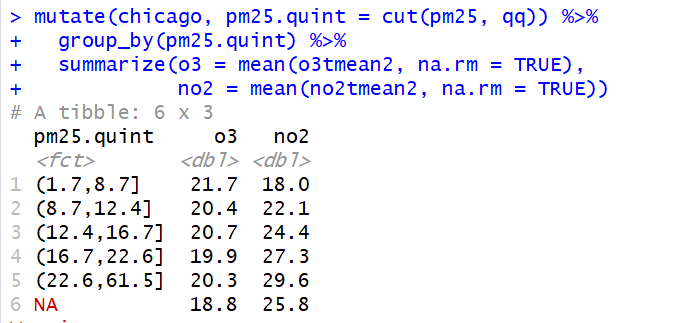


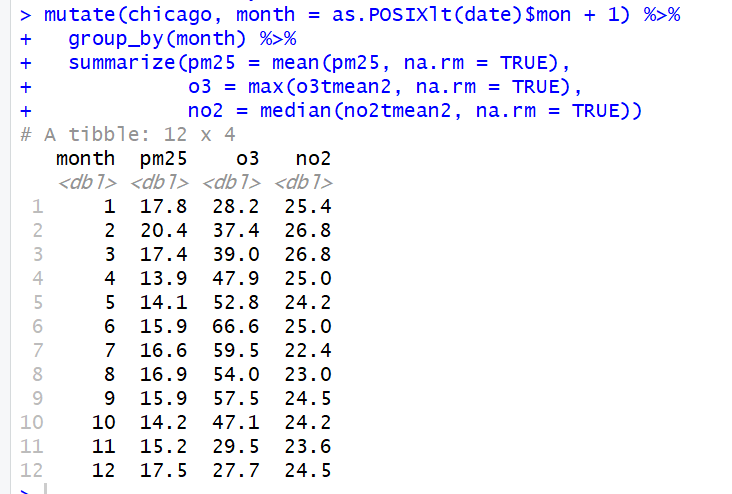
Let’s look into the group\_by() next.



Now we are seeing the data grouped together, in this example, by year.

The last function from the dplyr package is the pipeline operator. This is great for stringing together multiple dplyr functions within a sequence of operations.





## Reflection

This assignment was a good warm up to some of the basic commands necessary to perform a proper EDA on a data set. These fundamental RStudio commands are the basis around what will be accomplished within this class as well as in projects using EDA principles in the future. The dplyr library helps in that regard by providing analysts with commands they can use to further dissect and sort the data so that it is easier to explore and pull out meaningful information from.