Introduction to R

Lab 1

Dr. Cathy Poliak, cpoliak@uh.edu

University of Houston

Lab 1 Intsructions

- This lab is an introduction to some simple `R` commands.
- As we go through this lab you will answer some multiple choice questions that are in Canvas under Lab 1 in this week's module.
- The lab is due today by 2:00 pm.
- You should have `R` and `Rstudio` downloaded. If not follow the instructions on the syllabus. Open `Rstudio`.

R

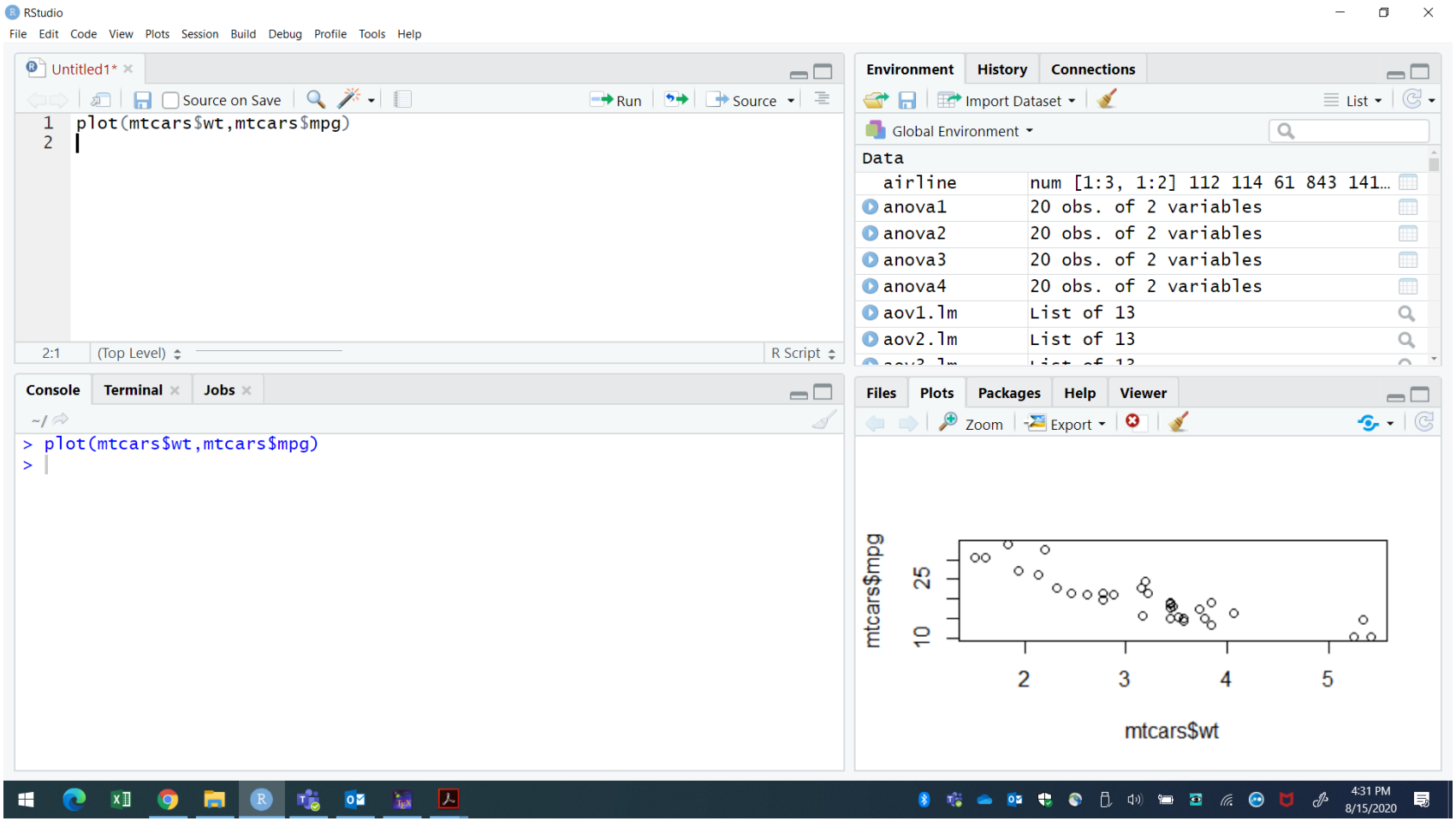
- R has become very popular over the past decade.
- It is an *open* source
- It is free
- Powerful enough to implement all of the methods discussed in this class
- Optional packages
- R is the language of choice for academic statisticians
- New approaches often become available in R years before they are implemented in commercial packages

Rstudio

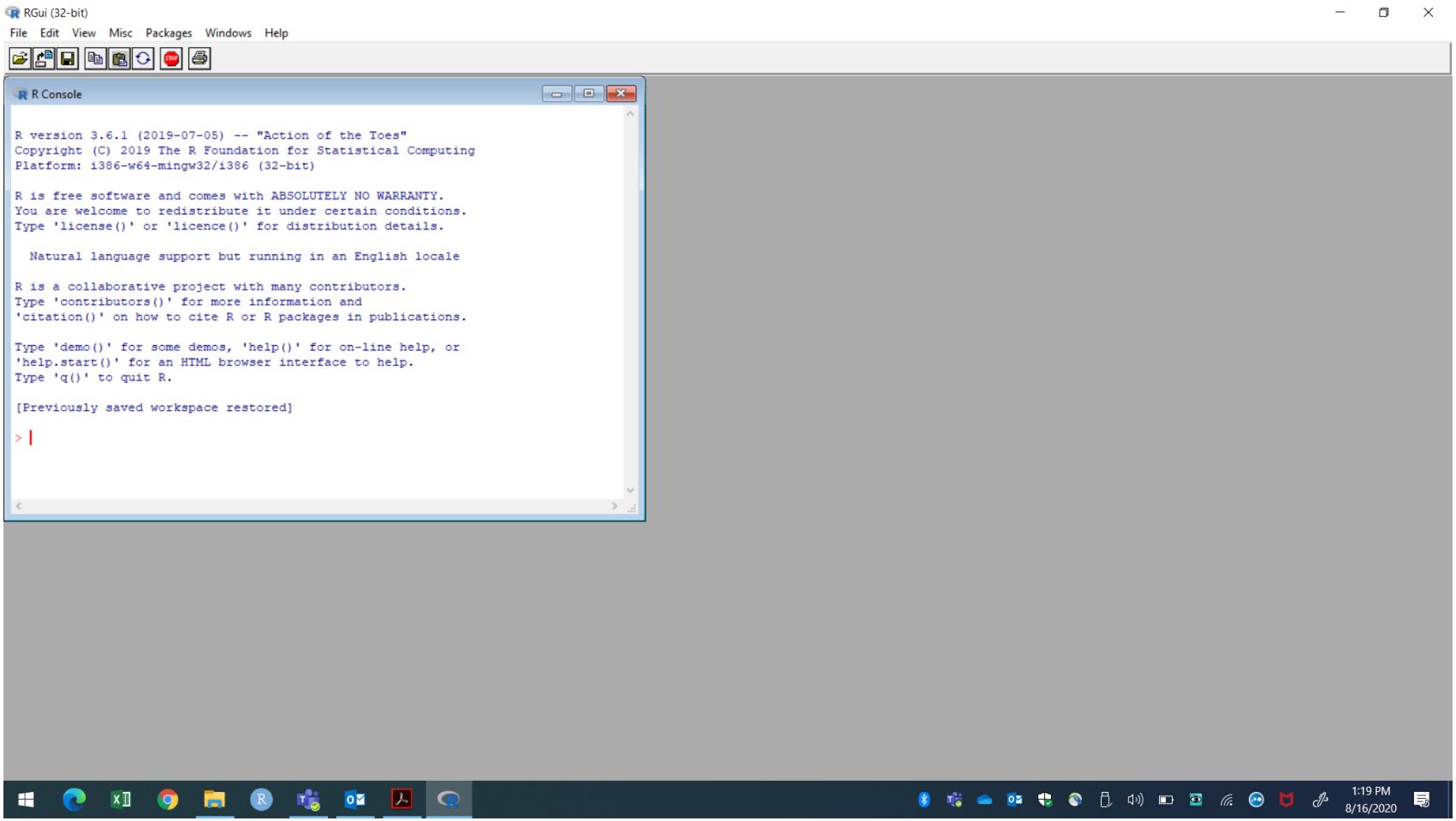
- In this class I will use `Rstudio`
- It is highly recommended that all users of `R` work in `Rstudio`
- `Rstudio` is an interface that provides both assistance for novices as well as productivity tools for experienced users.
- The `Rstudio` opens four windows:
  - One for editing code
  - A window for the console to execute R code
  - One track to the variables that are defined in the work space
  - The fourth to display graphical images

Source: Applied Multivariate Statistics with `R`

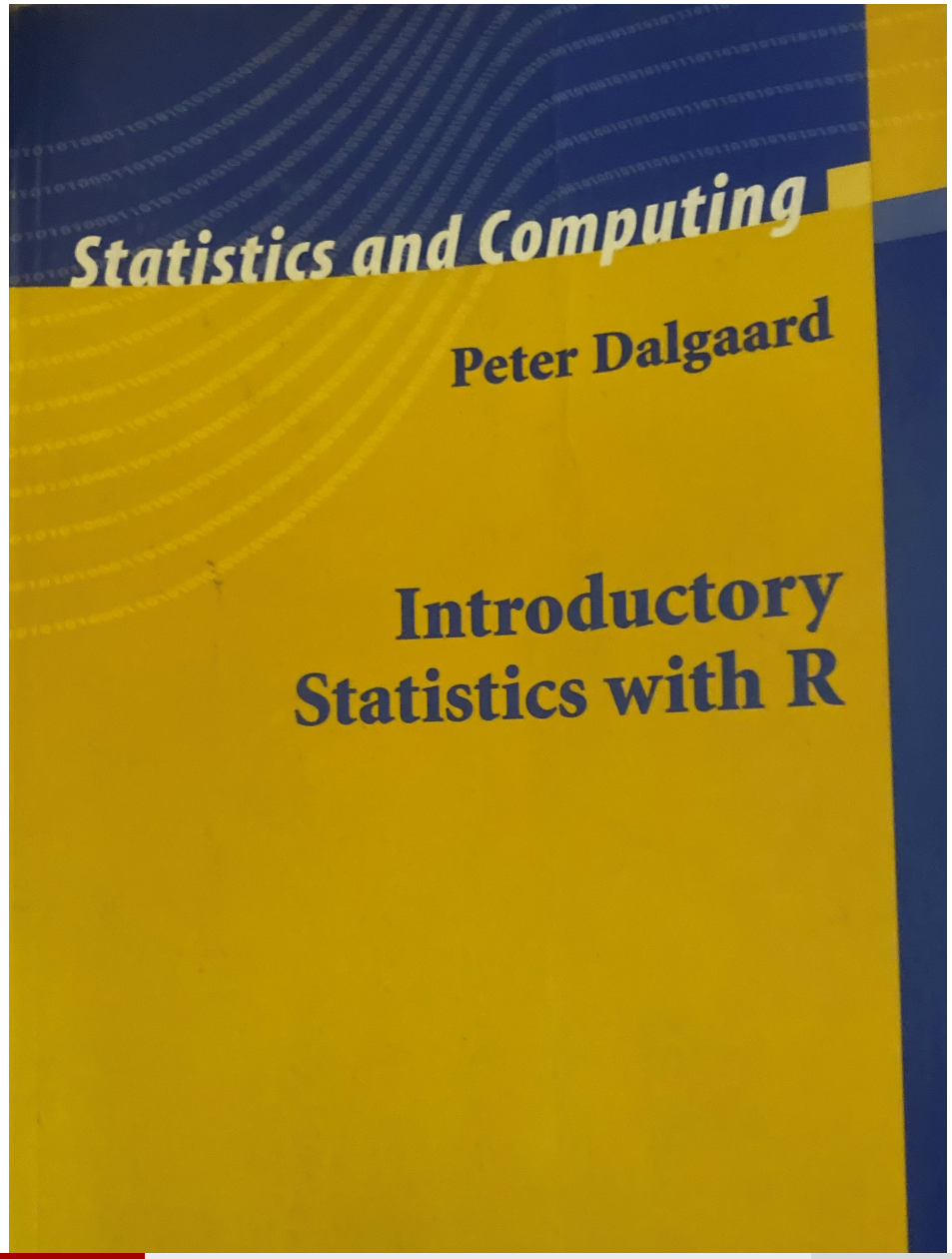
Rstudio Windows



R Window


RGui (32-bit)
File Edit View Misc Packages Windows Help
R Console
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Previously saved workspace restored]
1:19 PM
8/16/2020

Book Reccomendation about R



Aspects of R

- R Script - to start a new session, my recommendation is to from "File" $\rightarrow$ "R script". This opens up a blank window that allows you to edit code.
- Open `Rstudio` and select file $\rightarrow$ New File $\rightarrow$ R script
- An overgrown calculator - in `R` we can enter an arithmetic expression and receive a result. Type in the following in the R script, press "Enter" after each line.
  - $2 + 2$
  - exp(-2)
  - sqrt(224)
- Notice that you did not get an answer when you typed these lines. These lines have to be inputted into the console. `R` works as you enter a line with a command and press "Enter", then the program gives you a result from that input in the console. The ">" at the beginning of a line in the Console is a prompt to tell you `R` is ready for an input. To input the lines from the script you can have the cursor at the line you wish to answer and click on "Run" or press "Ctrl + Enter".
- Assignments - we can use "=" or "$<-$" to store results. For example type in the following in the R script then "Run" these lines.

Functions

- `R` uses functions to perform operations.
- To run a function say called `funcname` we type `funcname(input1, input2)`
- The inputs (or arguments) `input1` and `input2` tell `R` how to run the function.
- A function can have any number of inputs.
- For example to find a cumulative probability of a value based on the normal distribution is the function. The following information comes from the help file by typing in `?pnorm`.

pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

Lab Question 1

Using the `pnorm` function in `Rstudio`, determine the probability of getting at least a value of 34 from a Normal distribution with mean, $\mu = 30$ and standard deviation $\sigma = 2$.

a. 0.97725
b. 0.02275
c. 0.22750
d. 0.02700

$X \sim N(\mu = 30, \sigma = 2)$

$P(X \geq 34)$

1-pnorm(34, 30, 2)

or pnorm(34, 30, 2, lower.tail=F)

Creating a Vector

- To create a vector of numbers, we use the function "c(...)" (for *concatente* or "combine").
- A data vector is an array of numbers. The following vector variable can be created called 'weight'.

```
weight = c(60,72,57,90,95,72)
```

<-

Output

When we type `weight` in the script and run we will get back the vector

```
weight
```

```
[1] 60 72 57 90 95 72
```

The brackets [ ] are an indication of the index of the numbers. For example the following will give you the third value in the vector called "weight."

```
weight[3]
```

```
[1] 57
```

The parentheses ( ) are for the functions. For example run:

```
rnorm(15)
```

This gives you 15 values based on the standard normal distribution.

Lab Question 2

Create two vectors named *x* and *y*. The vector *x* should have the values: 1, 6, and 2. The vector *y* should have the values: 1, 4, and 3. After creating these two vectors type and run the following:

```
x + y
```

What is your result?

a. 2, 10, 5
b. 17
c. 1, 6, 2
d. 1, 4, 3

```
x = c(1,6,2)
y = c(1,4,3)
x + y
```

Other Functions

- `length()` checks the length (number of elements) in a vector.
- `ls()` function allows us look at a list of all of the objects that we have saved so far.
- `rm()` function can be used to delete any objects hat we do not want.
- `matrix()` function can be used to create a matrix of numbers. Type and run the following

```
(dog = matrix(c(1,2,3,4),nrow = 2, ncol = 2))
```

Parentheses ( ) around a line allows us to print that line.

Lab Questions

3. Type the following in the script and run. Do you get the same mean value?
   a. Yes
   b. No

```
x = rnorm(50)
mean(x)
```

```
[1] -0.2726718
```

4. Type the following in the script and run. Do you get the same mean value?
   a. Yes
   b. No

```
set.seed(1303)
x = rnorm(50)
mean(x)
```

```
[1] 0.1695526
```

Other Functions to Create Vectors

- The `rep(,)` function takes two arguments and is used to make multiple copies of the first argument.

```
c(1,rep(2,3),4)
```

```
[1] 1 2 2 2 4
```

```
rep(c(1,12),4)
```

```
[1]  1 12  1 12  1 12  1 12
```

```
rep(c(3,5),each =4)
```

```
[1] 3 3 3 3 5 5 5 5
```

- The `seq()` function can be used to create a sequence of numbers.

```
seq(1,10)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
1:10
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
head(seq(-3,3,by = 0.01))
```

```
[1] -3.00 -2.99 -2.98 -2.97 -2.96 -2.95
```

Indexing Data

Recall that the brackets [ ] are used for indexes. For example we will create a matrix and want to determine the element in the 2nd row 3rd column we would do

```
(A = matrix(1:16, 4, 4))
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

```
A[2,3]
```

```
[1] 10
```

Lab Questions

5. Create the matrix A as in the previous slide in `R` and type in `A[1, ]` this prints out
   a. The first row of matrix A.
   b. The first column of matrix A.
   c. The number 1.
   d. All but the first row of matrix A.
6. Type in `A[-1, ]` this prints out
   a. The first row of matrix A.
   b. The first column of matrix A.
   c. The number 1.
   d. All but the first row of matrix A.

The Plot Function

- The `plot(x,y)` function produces a scatterplot of the numbers in x versus the numbers in y.
- There are many other options that can passed in the `plot()` function.
- Type in `?plot` for more information.
- Type and run the following in `R`.
- There are other functions that we will use in this class and also a package called `ggplot2` to create plots.

```
x = rnorm(100)
y = rnorm(100)
plot(x,y)
```

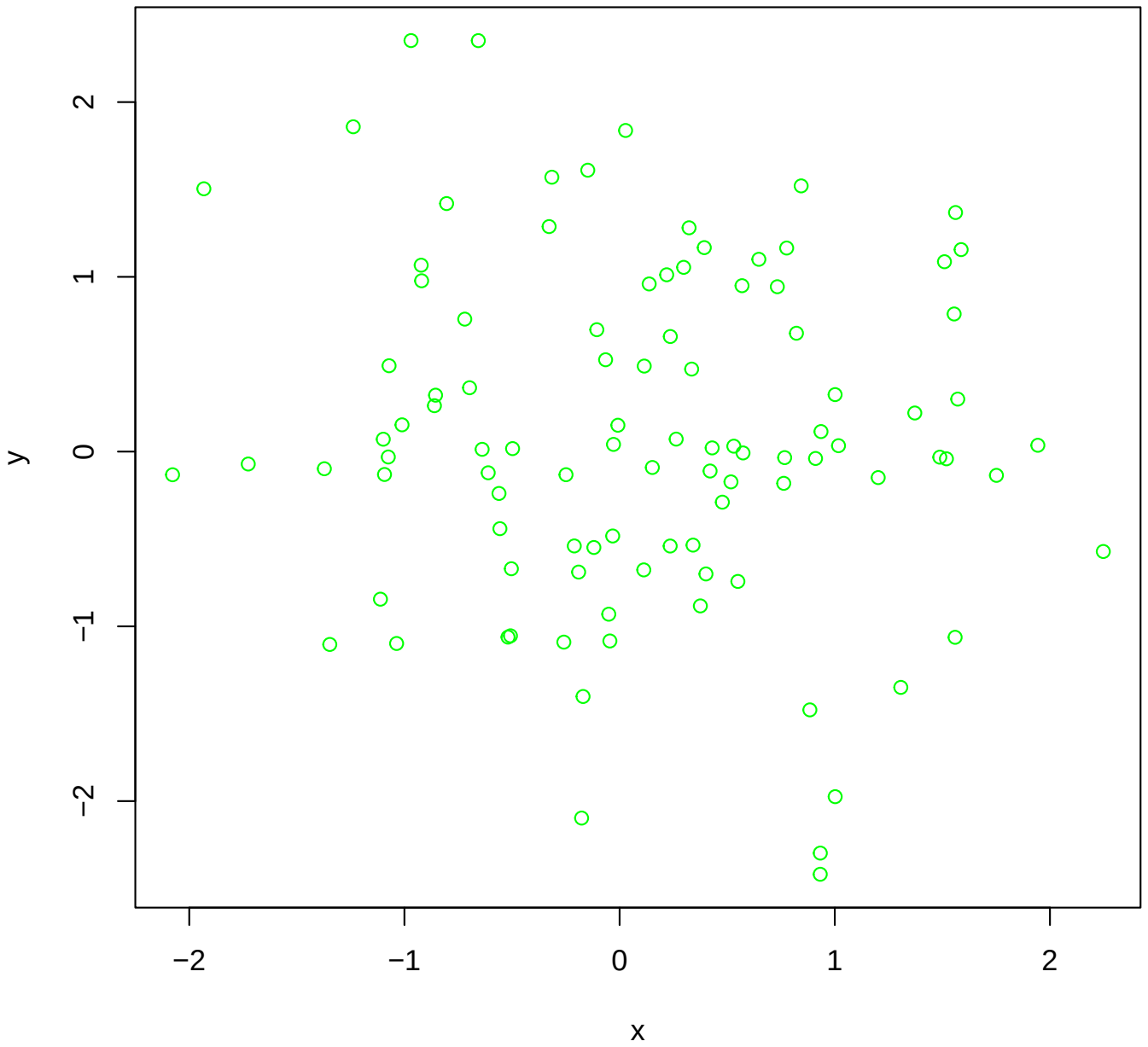
```
plot(x,y, xlab = "this is the x-axis",
     ylab = "this is the y-axis",
     main = "Plot of X vs Y")
```

Saving a Plot

- We can save a plot depending on the file type that we would like to create.
- `pdf()` will create a pdf file
- `jpeg()` will create a jpeg file
- We can also simply copy the plot window and past it into an appropriate file.

For example the following below will save the scatterplot as a PDF file in the directory that you have set. The function `dev.off( )` indicates to `R` that we are done creating the plot.

```
pdf("Figure.pdf")
plot(x,y,col = "green")
dev.off( )
```


y
2
1
0
−1
−2
−2
−1
0
1
2
x

Example Data Frame Within R

- Suppose we want to predict miles per gallon (mpg) for automobiles based on certain values.
  - *cyl* Number of cylinders
  - *disp* Displacement (cu.in.)
  - *hp* Gross horsepower
  - *wt* Weight (1000 lbs)
  - *am* Transmission (0 = automatic, 1 = manual)
- This data set is in R called *mtcars*.

Lab Questions

7. Type in the script ?mtcars and click Run. What year was this data was extracted from?
   a. 2019
   b. 2000
   c. 1984
   d. 1974
8. Type in the script dim(mtcars) then click Run the first number is the number of observations (rows) the second number is the number of variables (columns). How many observations?
   a. 11
   b. 32
   c. 352
   d. 43

Type in the script head(mtcars) then click Run.

9. How many rows appear?
   a. 32
   b. 16
   c. 6
   d. 2
10. How many cylinders are in the Hornet Sportabout?
   a. 4
   b. 6
   c. 8
   d. This car is not on the list.

Lets compare the Weight of a car (*wt*) with the *mpg* by a plot.

11. In the script type in plot(wt,mpg), click Run. Describe this plot
    a. Positive, linear
    b. Negative, linear
    c. No relationship
    d. I get an error
12. In the script type in plot(mtcars$wt,mtcars$mpg), click Run. Describe this plot
    a. Positive, linear
    b. Negative, linear
    c. No relationship
    d. I get an error

To refer to a variable, we must type the data set and the variable name joined with a $ symbol. Alternatively, we can use the attach() function in order to tell R to make the variables in this data frame available by name. In the script window type in attach(mtcars) click Run.

In the script window type in cyl = as.factor(mtcars$cyl) and click Run.

Since the number of cylinders is numeric, R recognizes these values as continuous or quantitative. However, these could be categorical (factors).

13. Type in the script window plot(cyl,mtcars$mpg) click Run, what plot do you see?
    a. Bar plot
    b. Boxplot
    c. Scatterplot
    d. Histogram
    e. I get an error
14. Type in the script window pairs(~mpg+disp+hp+wt,mtcars) click Run, what do you see in the graph window?
    a. Several scatterplots
    b. One scatterplot
    c. Histogram
    d. I get an error

In the script window type summary(mtcars$mpg).

15. What is the mean of `mpg`?
    a. 15.43
    b. 19.20
    c. 20.09
    d. 22.80

If you want to save this script you can select File → save as . . . then save this were you want. It will save it with .R (`R` file).

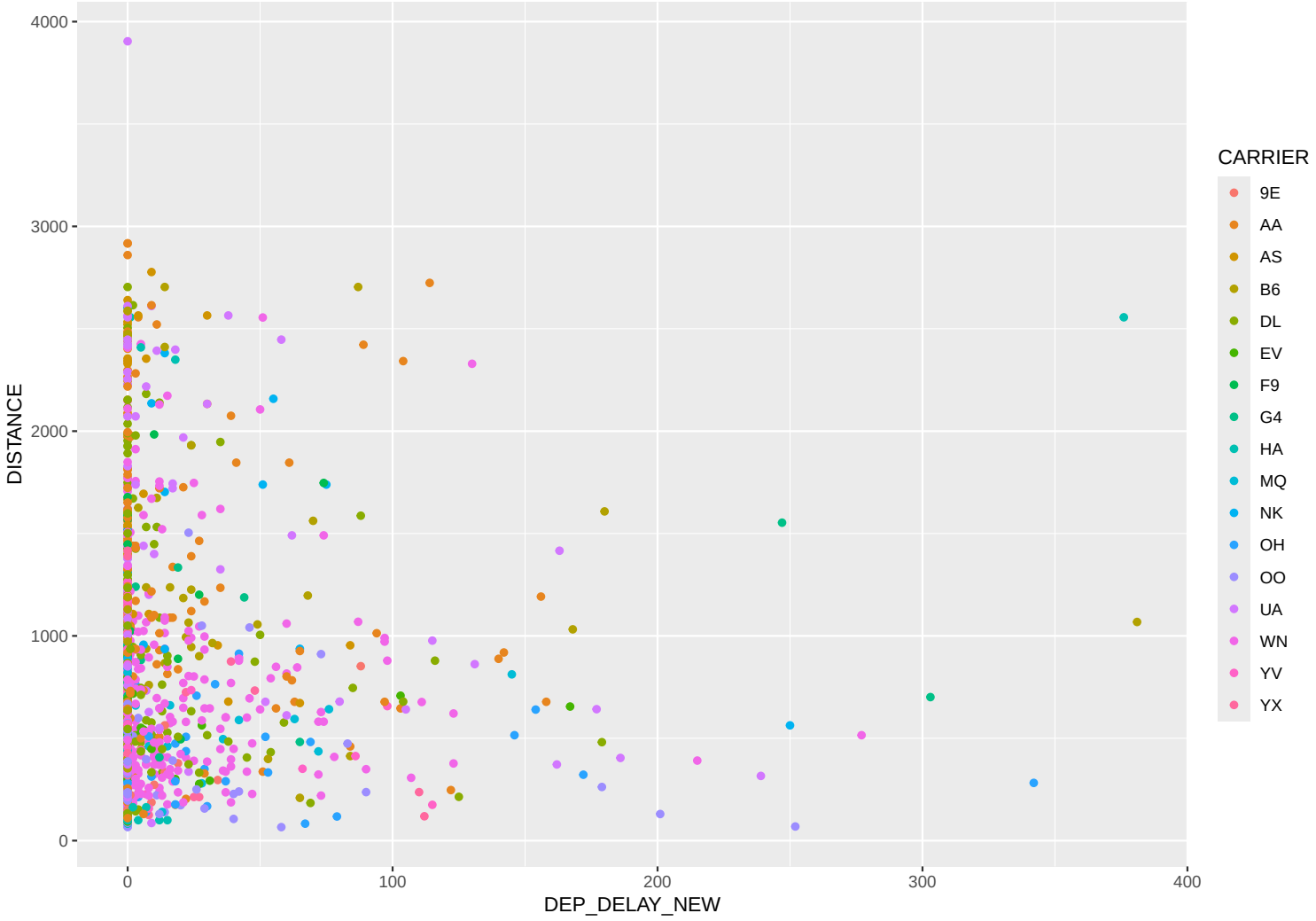
To Import a Data Set

- We will import the ontime.csv file. Use this link and save this file in any location.
- To import a data frame, in the Global Environment window select Import Dataset, then select From Text (base) then select `ontime.csv` to be imported.
- Here are some things we can do for data exploratory for this data frame.
  - summary(ontime)
  - ontime$CARRIER = as.factor(ontime$CARRIER)
  - summary(ontime$CARRIER)

Packages

- Since R is an open source, there have been several people that have built packages to use some other functions. We will explore the ggplot2 package. Grammar of Graphics plot
- To install a package type in the function install.packages. You only have to install once but every time you start R you are required to call that package if you want to use it by the function library(). For example,
  - install.packages("ggplot2")
  - library(ggplot2)
- Then we can create nicer plots. Type in the following.

```
ggplot(ontime, aes(x = DEP_DELAY_NEW, y = DISTANCE,color = CARRIER))+
     geom_point()
```


4000
3000
2000
1000
0
DISTANCE
0
100
200
300
400
DEP_DELAY_NEW
CARRIER
9E
AA
AS
B6
DL
EV
F9
G4
HA
MQ
NK
OH
OO
UA
WN
YV
YX