Starting with `Rstudio` and What is Statistical Learning?

Section 2.1

Cathy Poliak, Ph.D.
cpoliak@central.uh.edu

Department of Mathematics
University of Houston

Outline

1. Starting with `R` and `Rstuio`
2. Statistical Learning
3. Statistical Approaches

R

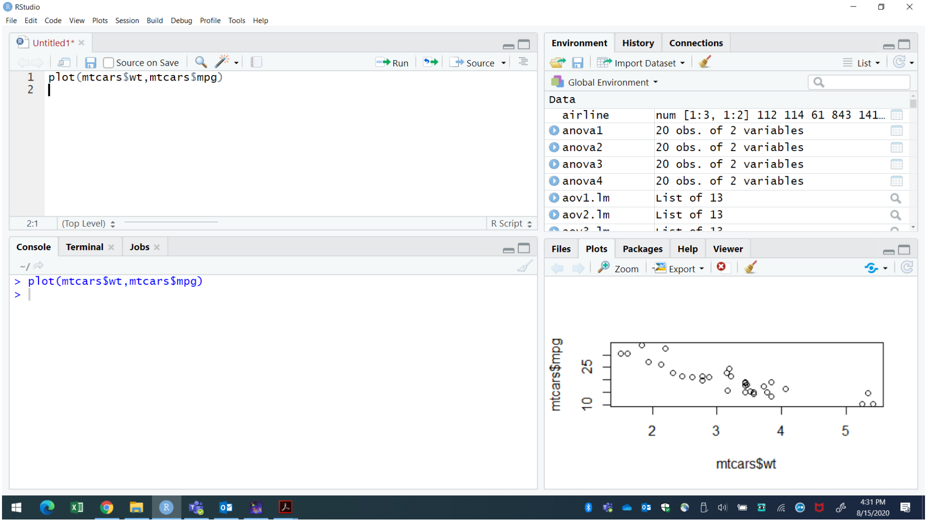
- R has become very popular over the past decade.
- It is an *open* source
- It is free
- Powerful enough to implement all of the methods discussed in this class
- Optional packages
- R is the language of choice for academic statisticians
- New approaches often become available in R years before they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

Source: Applied Multivariate Statistics with `R`

Rstudio Windows

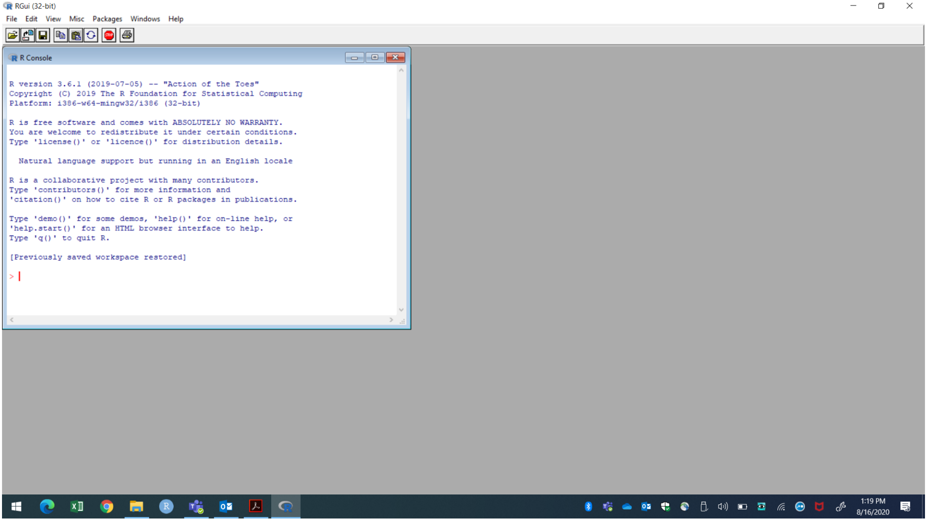


```r
plot(mtcars$wt,mtcars$mpg)
```

```
Data
  airline   num [1:3, 1:2] 112 114 61 843 141…
  anova1    20 obs. of 2 variables
  anova2    20 obs. of 2 variables
  anova3    20 obs. of 2 variables
  anova4    20 obs. of 2 variables
  aov1.lm   List of 13
  aov2.lm   List of 13
```

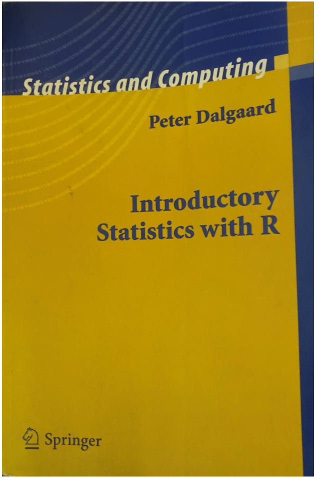
```
> plot(mtcars$wt,mtcars$mpg)
>
```

R Window



```
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]

> 
```

Book Recommendation about R



Aspects of R

- R Script - to start a new session, my recommendation is to from "File" → "R script". This opens up a blank window that allows you to edit code.
- Open Rstudio and select file → New File → 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.
  - x = 2
  - x
  - x + x

Vectorized Arithmetic

- R can handle data vectors as single objects.
- 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)
  - The "c(. . .)" is used to define vectors.
- The brackets [] are an indication of the index of the numbers. For example type in and Run
  - weight[3]
  - rnorm(15)
- We can use different functions to get information from the data vector. We type in that function then in parenthesis type in the assigned object we want to use. For example, type in the following and then "Run".
  - mean(weight)
  - sd(weight)

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

Answer the following questions

1. Type in the script ?mtcars and click Run. What year was this data was extracted from?

   a) 2019
   b) 2000
   c) 1984
   [✓ d) 1974]

2. 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

Lab Questions

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

3. How many rows appear?
    - a) 32
    - b) 16
    - [✓ c) 6]
    - d) 2

4. How many cylinders are in the Hornet Sportabout?
    - a) 4
    - b) 6
    - [✓ c) 8]
    - d) This car is not on the list.

Lab Questions

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

5. In the script type in plot(wt,mpg), click Run. Describe this plot
   - a) Positive, linear
   - b) Negative, linear
   - c) No relationship
   - [✓ d) I got an error]

6. 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 got 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.

Lab Questions

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 should be categorical (factors).

mtcars$

7. Type in the script window plot(cyl,mpg) click Run, what plot do you see?
   a) Bar plot
   [✓ b) Boxplot]
   c) Scatterplot
   d) Histogram
   e) I got an error

8. 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) A histogram
   d) I got an error

Lab Questions

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

9. 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, type in
  - 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()
```

Goal of Statistical Learning

The main goal of statistical learning theory is to provide a framework for studying the problem of inference, that is of gaining knowledge, making predictions, making decisions or constructing models from a set of data. This is studied in a statistical framework, that is there are assumptions of statistical nature about the underlying phenomena (in the way the data is generated). [1]

[1] *Introduction to Statistical Learning Theory, O. Bousquet, S. Boucheron, and G. Lugosi*

Example

The goal is to predict the *stock_index_price* (the dependent variable) of a fictitious economy based on two independent/input variables:

- *Interest_Rate*
- *Unemployment_Rate*

The data is in the *stock_price.csv* data set in BlackBoard. This is from `https://datatofish.com/multiple-linear-regression-in-r/`

Questions We Want To Answer

1. Is there a relationship between *stock index price* and *interest rate*?
2. How strong is the relationship between *stock index price* and *interest rate*?
3. Is the relationship linear?
4. How accurately can we predict the *stock index price*?
5. Do both *interest rate* and *unemployment rate* contribute to the *stock index price*?

General Approach

- *Stock index price* is the response or output. We refer to the response usually as Y.

- *Interest rate* is an input or predictor, we will name it X_1.

- Also, *Unemployment rate* is an input, we will name it X_2.

- Let $X = (X_1, X_2, \ldots, X_p)$ be p different predictors (independent) variables.

- For this example we will have an input vector as

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

- We assume there is some sort of relationship between X and Y, which can be written in the general form thus our model is

$$Y = f(X) + \epsilon$$

- Where ϵ captures the measurement errors and other discrepancies.

- Statistical leaning refers to a set of approaches for estimating f.

Why Estimate $f(X)$?

- If we have a good $f(X)$ we can make predictions of Y at new points where $X = x$.
- We can understand which variables (components) of $X = (X_1, X_2, \ldots, X_p)$ are important in explaining Y and which ones are irrelevant.
- Depending on the complexity of f, we may be able to understand how each variable X_j of X affects Y.
- Adapted from https://hastie.su.domains/ISLR2/Slides/Ch2_Statistical_Learning.pdf

How Do We Estimate f?

- The goal is to apply a statistical learning method to the training data in order to estimate the unknown function of f.

- Using a model-based approach, called **parametric**, with assumptions about the model.
  1. We make an assumption about the function form or shape of f.
  2. We need a procedure that uses the training data to fit or train the model.

- No assumptions about the model is called a **non-parametric** method.
  - Non-parametric method seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.
  - **Advantage**: they have the potential to accurately fit a wider range of possible shapes for f.
  - **Disadvantage**: a very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for f.

Parametric Method

Parametric methods involve a two-step model-based approach.

1. We make an assumption about the functional form, or shape, of f. Then determine a model.
2. After a model has been selected, we need a procedure that uses the *training* data to fit or train the model.
   - The training data are observations used to train or teach our method how to estimate f.
   - Let x_{ij} represent the value of the jth predictor for observation i, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$.
   - Let y_i be the response variable for the ith observation.
   - Then the training data consist of $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$.

Training, test, and validation sets

- The model is initially fit on a **training data set**, that is a set of observations used to fit the parameters.
- Successively, the fitted model is used to predict the responses for the observations in a second data set called the **validation data set.**
- Finally, the **test data set** is a data set used to provide an unbiased evaluation of a final model fit on the training data set

Confusingly the terms test data set and validation data set are sometimes used with swapped meaning. As a result it has become commonplace to refer to the set used in iterative training as the test/validation set and the set that is used for hyper parameter tuning as the **holdout set**.

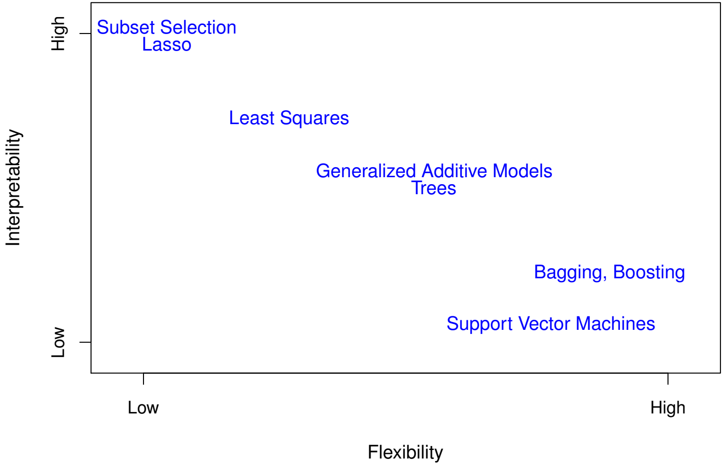
Flexibility vs. Interpretation

Why choose to use a more restrictive method instead of a very flexible approach?

- If we are mainly interested in inference, then restrictive models are much more interpret-able. For example, when inference is the goal, the linear model may be a good choice since it will be quite easy to understand the relationship between Y and $X_1, X_2, \ldots, X_p$.

- Very flexible approaches, such as the splines and the boosting methods can lead to such complicated estimates of f that it is difficult to understand how any individual predictor is associated with the response.

Flexibility vs. Interpretation



Two Analysis

- Given response and predictor - supervised learning
  - Regression - quantitative variables
  - Classification -categorical variable
- Given only factors without a response variable - unsupervised learning
  - Clustering

Prediction $Y = f(x) + \underset{\sigma^2}{\underline{\varepsilon}}$

The accuracy of $\hat{Y}$ as a prediction for Y depends on two quantities:

- **Reducible error**
  - $\hat{f}$ is not a perfect estimate for f, and this inaccuracy will introduce some error.
  - We can potentially improve the accuracy of $\hat{f}$ by using the most appropriate statistical learning technique to estimate f.
- **Irreducible error**
  - However, even if it were possible to form a perfect estimate for f, so that our estimated response took the form $\hat{Y} = f(X)$, our prediction would still have some error in it!
  - This is because Y is also a function of ϵ, which, by definition, cannot be predicted using X.
  - Therefore, variability associated with ϵ also affects the accuracy of our predictions.
  - No matter how well we estimate f, we cannot reduce the error introduced by ϵ.

Inference Questions

If we are interested in inference we are asking we are interested in answering the the following questions:

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?
- Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

Types of Problems

- **Regression** problem is when we are the response is a *continuous* or *quantitative* output value.
- **Classification** problem is when the response is a *categorical* or *qualitative* output.

Regression or Classification?

In the following examples do would we use Regression methods or Classification methods? Also are we most interested in prediction or inference? What is the sample size (n) and the number of variables (p)?

1. We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect the CEO salary.

Y = CEO Salary quant ⇒ regression → Inference

$n = 500 \quad p = 4$

2. We are considering launching a new product and wish to know whether it will be a success or failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price and 10 other variables.

Y = success/failure ⇒ classification

Prediction

$n = 20 \quad p = 14$

Lab Questions

10. What type of statistical learning problem is the Stock index price example?
    - [✓ a) Supervised regression]
    - b) Supervised classification
    - c) Unsupervised