

Applied Computational Statistics for Analytics!

Welcome!

The lecture will begin in a few moments

Sonja Petrović

Created for ITMD/ITMS/STAT 514

SonjaPetrovicStats.com/teaching/514sp21

Spring 2021.

First lectures & expectations

- You will learn a little bit about me
- I will learn a little bit about you
- We will go over some of the course logistics
 - The course syllabus website has **a lot more** information than I can cover in a few minutes.
 - What we will go over is **an overview**.
 - The rest is your “homework” to read and discover.

First lectures & expectations

- You will learn a little bit about me
- I will learn a little bit about you
- We will go over some of the course logistics
 - The course syllabus website has **a lot more** information than I can cover in a few minutes.
 - What we will go over is **an overview**.
 - The rest is your “homework” to read and discover.

Let's get started!



Roadmap

- Lectures: **Zoom** [as you know, since you are here!]
 - Be prepared to open a browser or your smart phone or other device to **AhaSlides.com/STATITM n** ← note link changes weekly to $n =$ current week number.
- Everything else: **Campuswire**
 - links to lecture video (speaker&screen view only)
 - PDFs of notes written during lecture
 - any slides shared
 - any handouts for reading
 - HW links for download & submission. These will take you to **Google Classroom**. You need to add this course on Google classroom.
 - Questions, information, discussion, peer Q&A, etc.

The tech:

- We will be using R code and Python code.
- You will learn to type Markdown documents.
- This will be required submission format for all work.

Lecture time:

The majority of the lecture was done via AhaSlides and iPad screen share in the live lecture. Current students will receive the video recording.

- Statistical reasoning with data
- Role of probability, role of chance
- A short overview of what R/Rstudio look like (next slides)

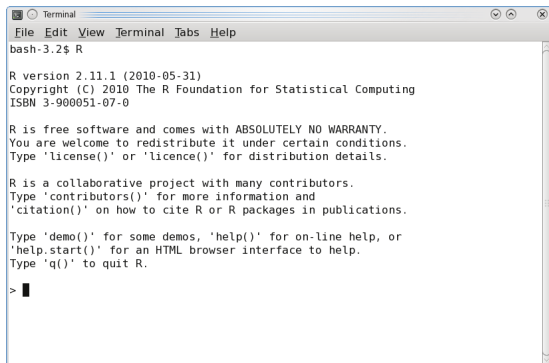
Why R?

- Free (open-source)
- Programming language (not point-and-click)
- Excellent graphics
- Offers broadest range of statistical tools
- Easy to generate reproducible reports
- Easy to integrate with other tools

The R Console

Basic interaction with R is through typing in the **console**

This is the **terminal** or **command-line** interface



```
Terminal
File Edit View Terminal Tabs Help
bash-3.2$ R
R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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.

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.

> █
```

The R Console & RStudio & Python (together)

- You type in commands, R gives back answers (or errors)
- Menus and other graphical interfaces are extras built on top of the console
- We will use **RStudio** in this class
- ① Download R: <https://cran.r-project.org/mirrors.html>
- ② Then download RStudio: <http://www.rstudio.com/>
- ③ Install Python (if you don't already have it):
<https://www.anaconda.com/>

RStudio is an IDE for R

RStudio has 4 main windows ('panes'):

- Source
- Console
- Workspace/History
- Files/Plots/Packages/Help

RStudio is an IDE for R

RStudio has 4 main windows (aka 'panes'):

- Source
- Console
- Workspace/History
- Files/Plots/Packages/Help

The screenshot displays the RStudio IDE interface with three main panes:

- Source Pane:** Contains R code for data analysis and plotting:

```
1 library(ggplot2)
2 source("plots/FormatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12           data=diamonds, color=clarity,
13           xlab="carat", ylab="price",
14           main="Diamond Pricing")
15
```
- Console Pane:** Shows the output of the executed code:

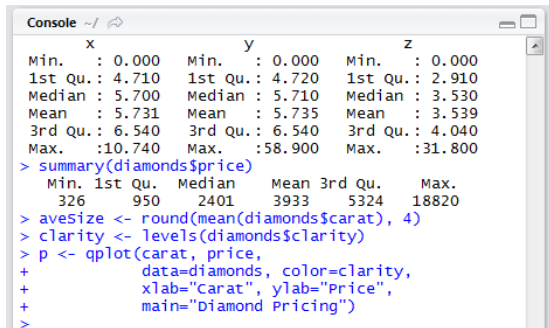
```
15:1 [Top Level] >
  x      y      z
Min.   : 0.000 Min.   : 0.000 Min.   : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean   : 5.731 Mean   : 5.735 Mean   : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max.   :10.740 Max.   :18.900 Max.   :131.800
> summary(diamonds$price)
  Min. 1st Qu. Median Mean 3rd Qu.  Max.
  326   950   2401  3913  5324 18620
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
```
- Plots Pane:** Displays a scatter plot titled "Diamond Pricing". The x-axis is labeled "carat" and the y-axis is labeled "Price". The plot shows a positive correlation between carat weight and price. Points are colored by clarity, with a legend on the right showing categories: I1, SI2, SI1, VS2, VS1, VVS2, VVS1, and IF.

RStudio: Panes overview

- 1 **Source** pane: create a file that you can save and run later
- 2 **Console** pane: type or paste in commands to get output from R
- 3 **Workspace/History** pane: see a list of variables or previous commands
- 4 **Files/Plots/Packages/Help** pane: see plots, help pages, and other items in this window.

Console pane

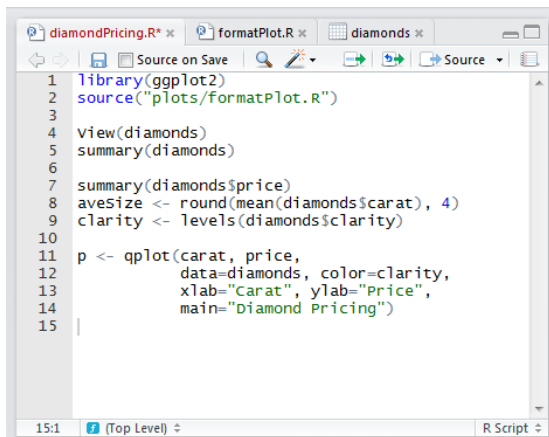
- Use the **Console** pane to type or paste commands to get output from R
- To look up the help file for a function or data set, type `?function` into the Console
 - E.g., try typing in `?mean`
- Use the tab key to auto-complete function and object names



```
Console ~/ | ↻  
  
      x              y              z  
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
Median : 5.700   Median : 5.710   Median : 3.530  
Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
Max.   :10.740   Max.   :58.900   Max.   :31.800  
> summary(diamonds$price)  
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     
   326   950    2401   3933   5324   18820  
> averse <- round(mean(diamonds$carat), 4)  
> clarity <- levels(diamonds$clarity)  
> p <- qplot(carat, price,  
+           data=diamonds, color=clarity,  
+           xlab="Carat", ylab="Price",  
+           main="Diamond Pricing")  
>
```

Source pane

- Use the **Source** pane to create and edit R and Rmd files
- The menu bar of this pane contains handy shortcuts for sending code to the **Console** for evaluation



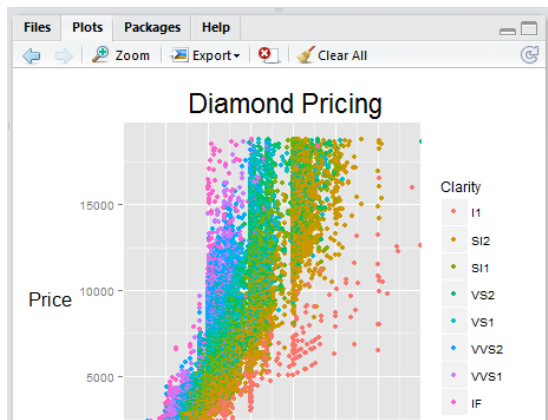
The screenshot shows the R Source pane with the following code:

```
1 library(ggplot2)
2 source("plots/FormatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12            data=diamonds, color=clarity,
13            xlab="Carat", ylab="Price",
14            main="Diamond Pricing")
15 |
```

The interface includes a menu bar with options like "Source on Save", "Source", and "Source" (with a dropdown arrow). The status bar at the bottom shows "15:1" and "R Script".

Files/Plots/Packages/Help pane

- By default, any figures you produce in R will be displayed in the **Plots** tab
 - Menu bar allows you to Zoom, Export, and Navigate back to older plots
- When you request a help file (e.g., ?mean), the documentation will appear in the **Help** tab



RStudio: Source and Console panes

Source window: Create a file here, so that you can save and run it later (or turn in as homework)

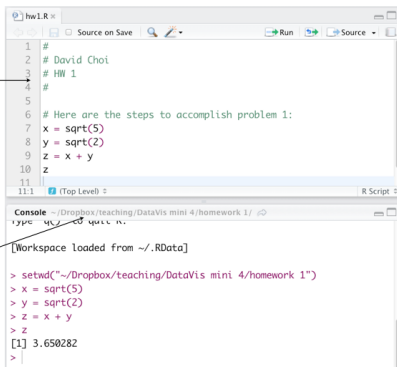
Any non-command line should start with a #

Console: Type or paste commands in here to get results from R

If you are loading a data file, you will need to be in the correct directory

> denotes that R is waiting for a command

+ denotes that R is waiting for you to finish the previous command (not shown here)



```
hw1.R x
Source on Save Run Source
1 #
2 # David Choi
3 # HW 1
4 #
5
6 # Here are the steps to accomplish problem 1:
7 x = sqrt(5)
8 y = sqrt(2)
9 z = x + y
10 z
11 |
11:1 (Top Level) ± R Script

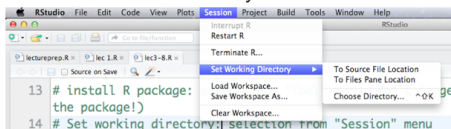
Console ~/Dropbox/teaching/DataVis mini 4/homework 1/
type ctrl+q ctrl+n
[Workspace loaded from ~/.RData]

> setwd("~/Dropbox/teaching/DataVis mini 4/homework 1")
> x = sqrt(5)
> y = sqrt(2)
> z = x + y
> z
[1] 3.650282
> |
```

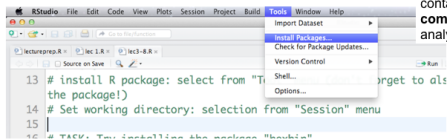
RStudio: Toolbar

Two helpful menu items in Rstudio

- Set the current directory:



- Install a package



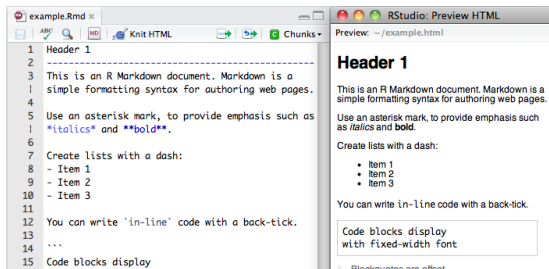
Packages are extensions to R, containing **new commands** for analysis or graphics

R Markdown

- R Markdown allows the user to integrate R code into a report
- When data changes or code changes, so does the report
- No more need to copy-and-paste graphics, tables, or numbers
- Creates **reproducible** reports
 - Anyone who has your R Markdown (.Rmd) file and input data can re-run your analysis and get the exact same results (tables, figures, summaries)
- Can output report in HTML (default), Microsoft Word, or PDF

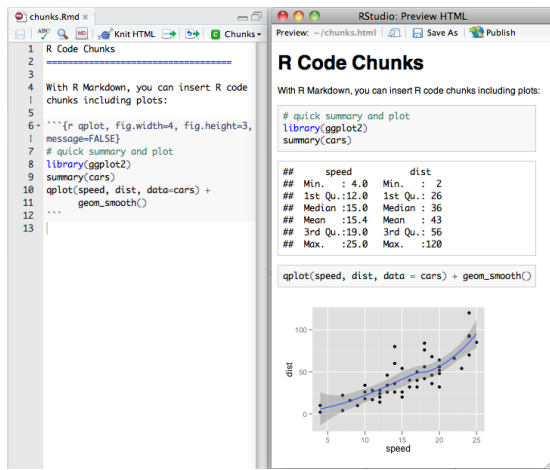
R Markdown

- This example shows an **R Markdown** (.Rmd) file opened in the Source pane of RStudio.
- To turn an Rmd file into a report, click the **Knit HTML** button in the Source pane menu bar
- The results will appear in a **Preview window**, as shown on the right
- You can knit into html (default), MS Word, and pdf format
- These lecture slides are also created in RStudio (using beamer_presentation as the output format, which is not what you typically will use)



R Markdown

- To integrate R output into your report, you need to use R code chunks
- All of the code that appears in between the “triple back-ticks” gets executed when you Knit



The image shows a side-by-side comparison of R code in a source editor and its rendered HTML output in a preview window.

Source Editor (chunks.Rmd):

```
1 R Code Chunks
2 =====
3
4 With R Markdown, you can insert R code
5 chunks including plots:
6
7 ```{r qplot, fig.width=4, fig.height=3,
8 message=FALSE}
9 # quick summary and plot
10 library(ggplot2)
11 summary(cars)
12 # quick summary and plot
13 aplot(speed, dist, data=cars) +
14   geom_smooth()
15 ```
```

Preview HTML:

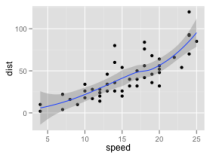
R Code Chunks

With R Markdown, you can insert R code chunks including plots:

```
# quick summary and plot
library(ggplot2)
summary(cars)
```

##	speed	dist
##	Min. : 4.0	Min. : 2
##	1st Qu.:12.0	1st Qu.: 26
##	Median :15.0	Median : 36
##	Mean :15.4	Mean : 43
##	3rd Qu.:19.0	3rd Qu.: 56
##	Max. :25.0	Max. :120

```
qplot(speed, dist, data = cars) + geom_smooth()
```



The plot displays a scatter of data points with a blue smoothed regression line and a grey shaded confidence interval, showing a positive correlation between speed and distance.

More information and instructions

Where's Python?

For an instructional 8-minute video on how to run Python inside RStudio, check the Campuswire link.

Check Campuswire for additional handouts, notes, etc.!

License

These slides are licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).

This document is created for ITMD/ITMS/STAT 514, Spring 2021, at Illinois Tech. Part of it is sourced from materials created by Prof. Alexandra Chouldechova from CMU distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.