

# Week 2

# Introduction to R

Dragos Ailoae  
dragos@nyu.edu

Applied Statistics and Econometrics 1  
ECON GA-1101  
Lab Sections 002 and 005

New York University  
September 13, 2022

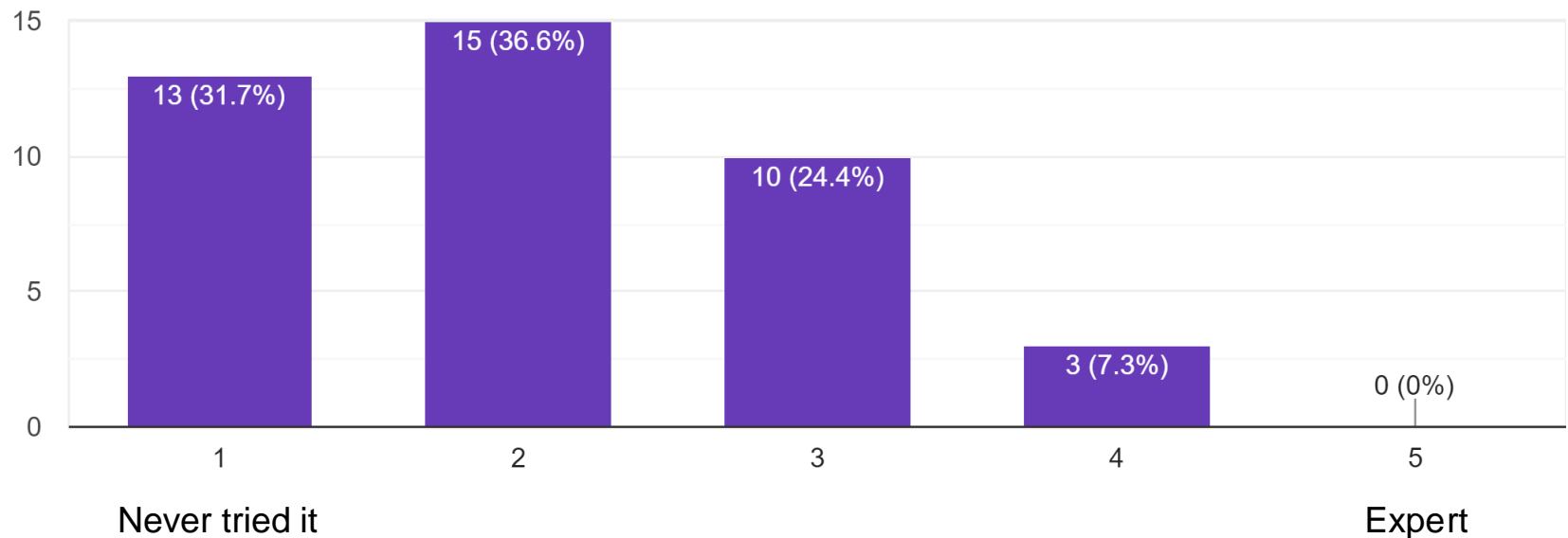
# Today

1. About You: Survey Results
2. Intro to R
3. Next Steps

# About You: Software experience

R

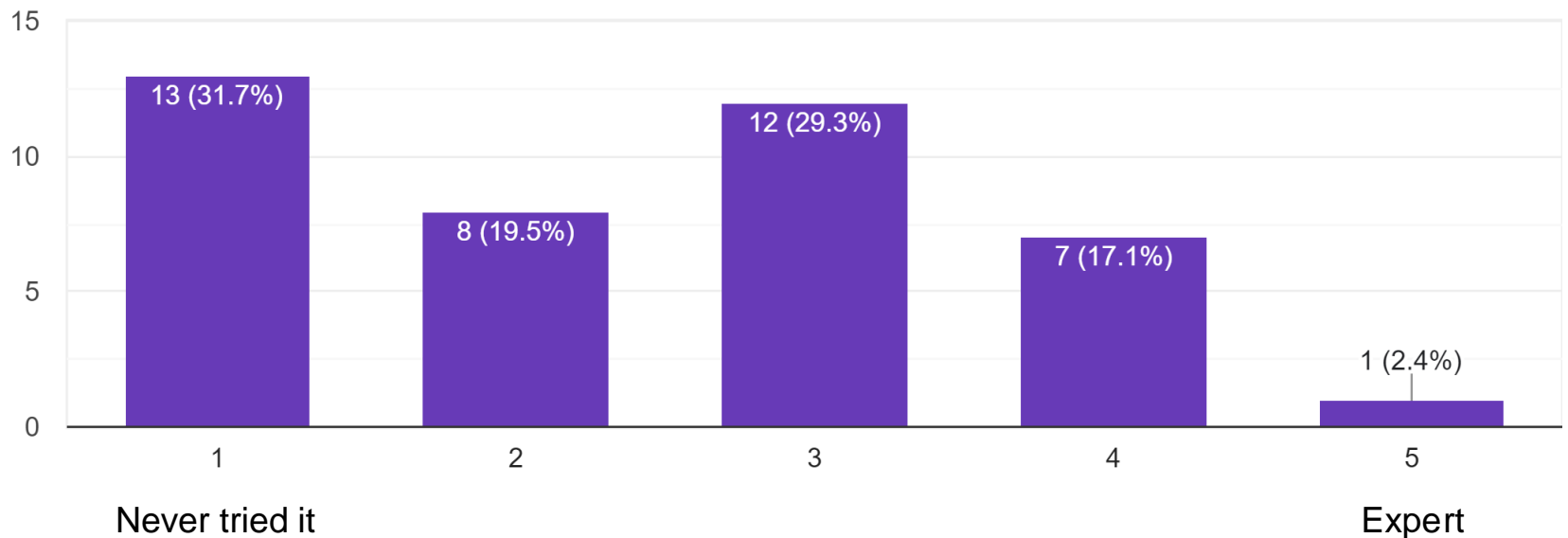
41 responses



# About You: Software experience

## Stata

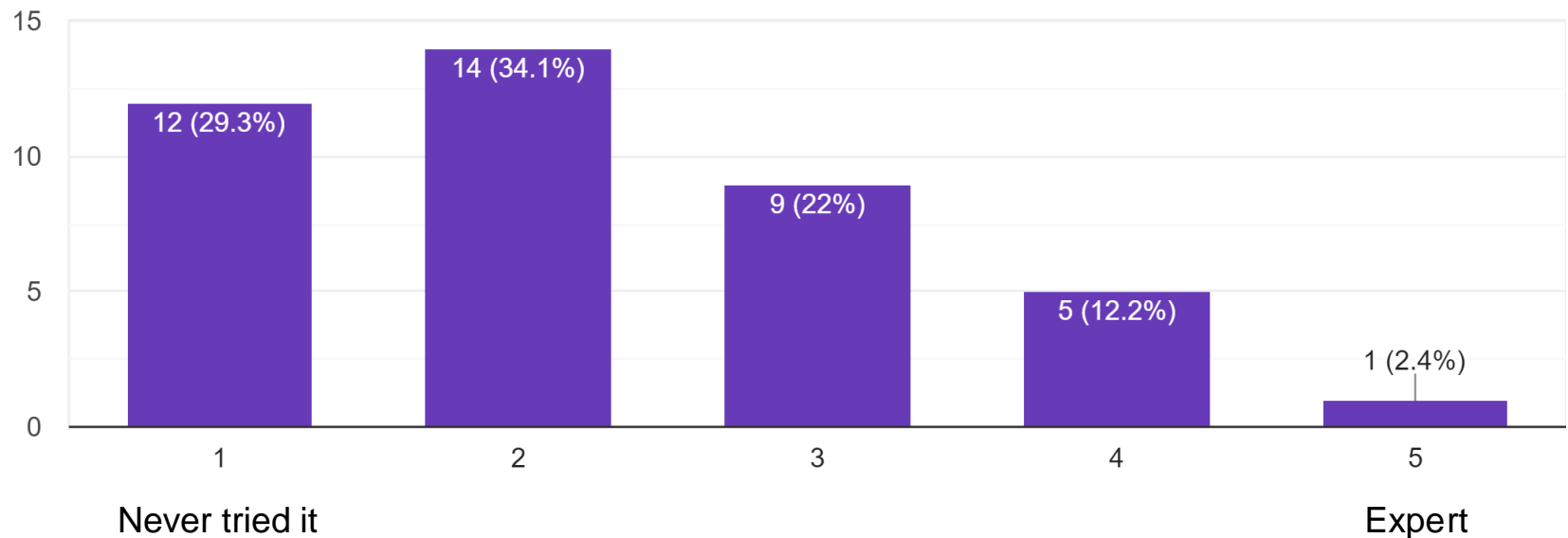
41 responses



# About You: Software experience

Python

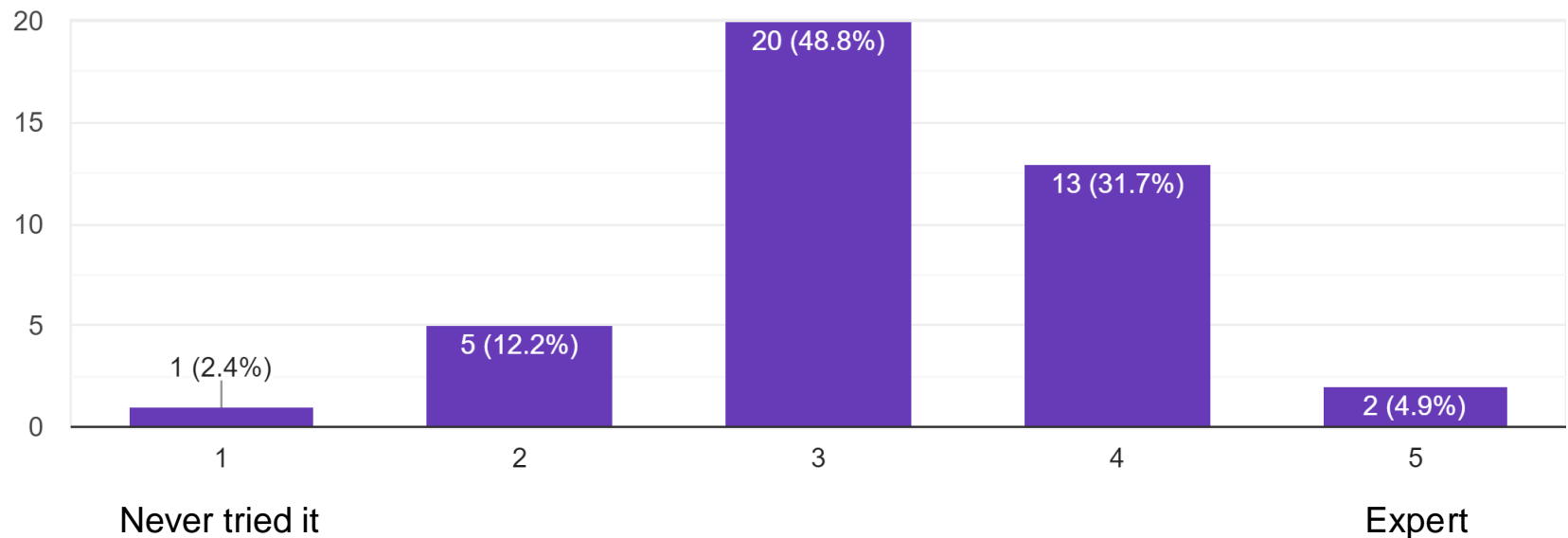
41 responses



# About You: Software experience

Excel

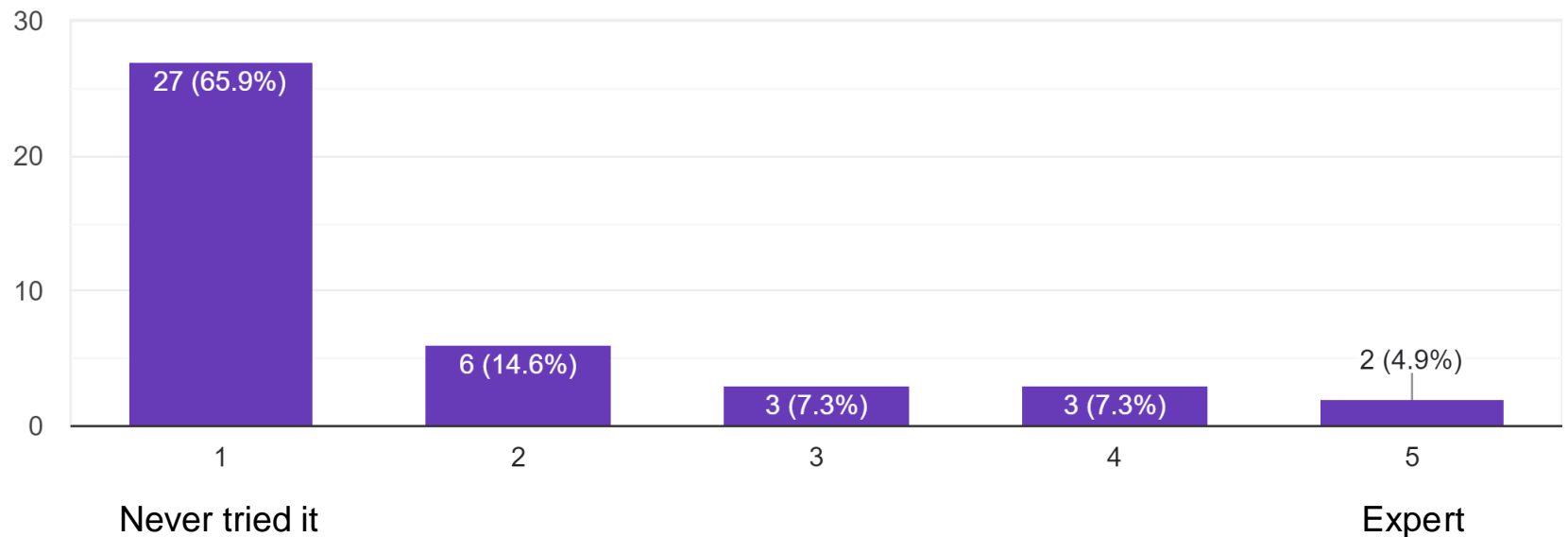
41 responses



# About You: Software experience

LaTeX

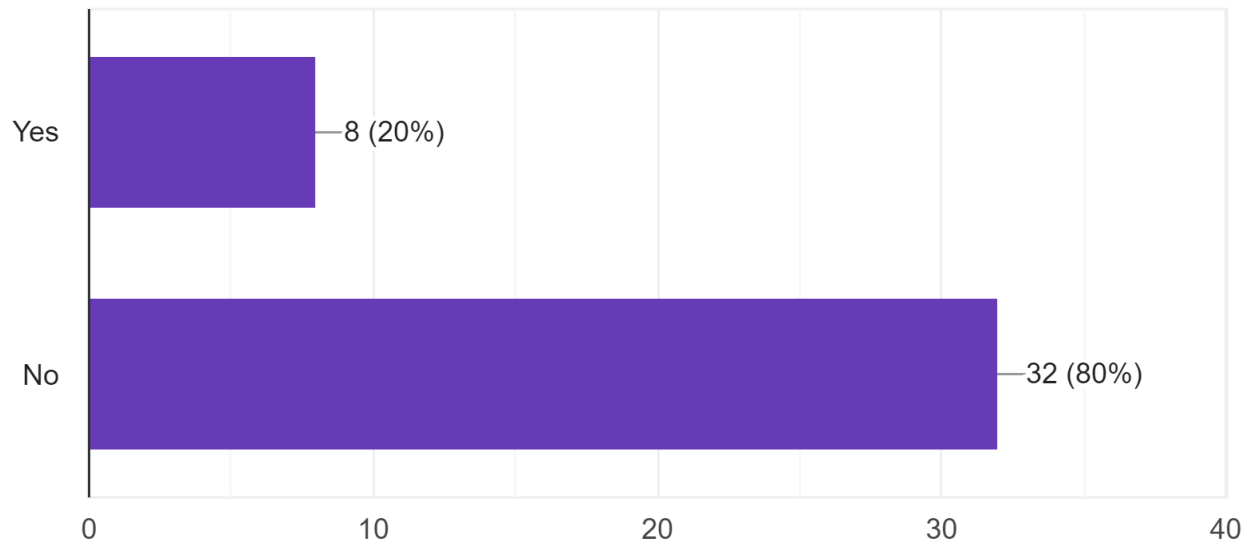
41 responses



# About You

This is my first econometrics course

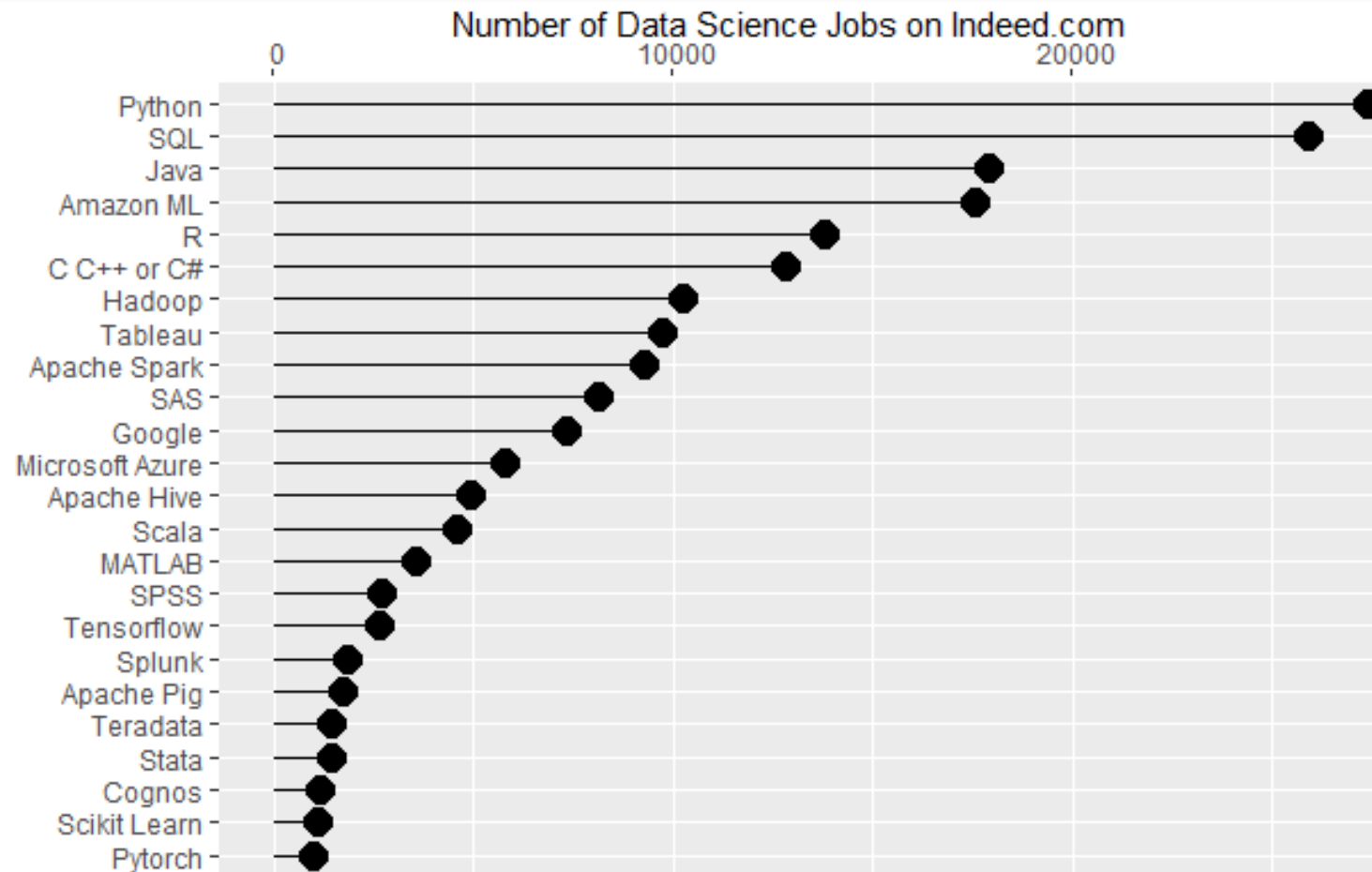
40 responses





# Intro to R

# Number of data science jobs for the more popular software (2019)



Source: <http://r4stats.com/articles/popularity/>

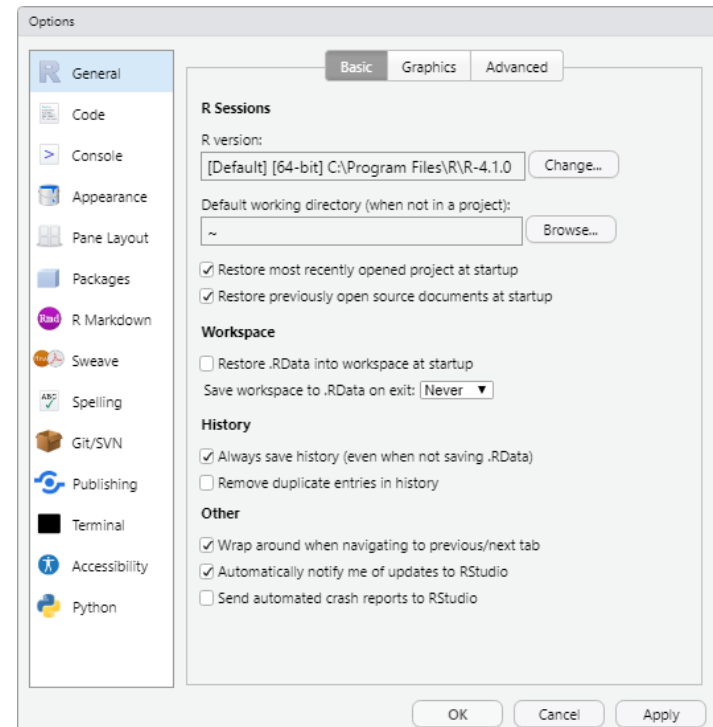
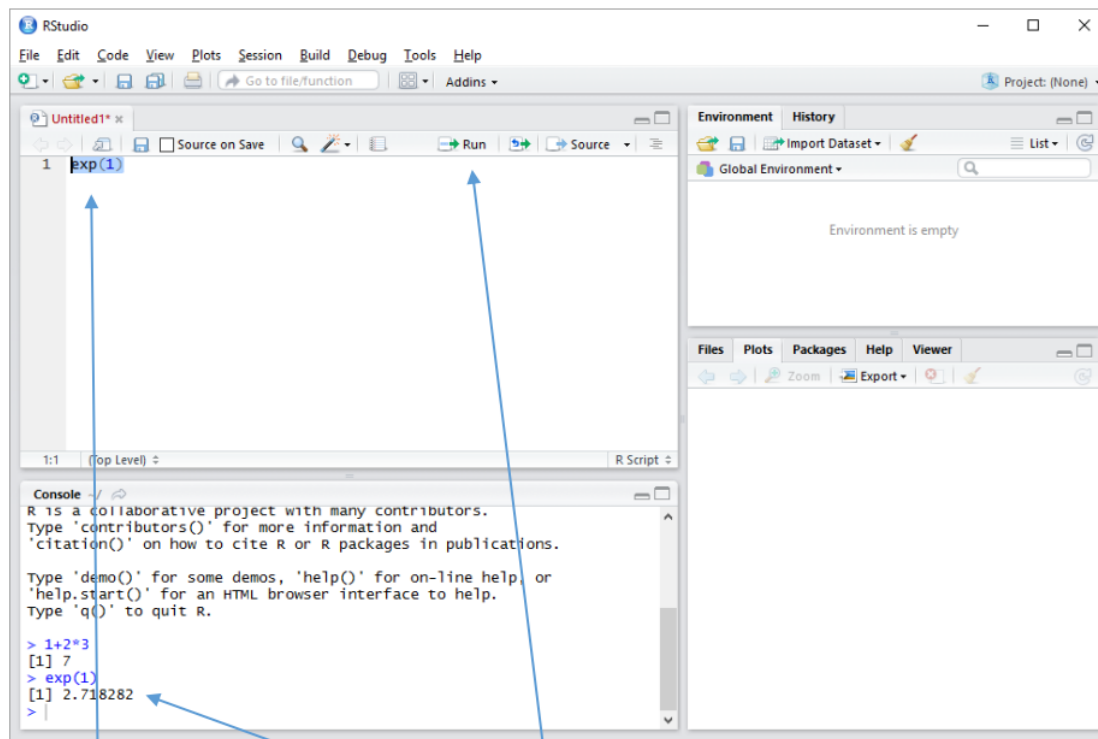
# Learning R: Pros

- Jobs: Excellent outside options for econ graduates going into non-academic positions
- Free (included by default on most supercomputer platforms)
- Easy transition to Python (Stata approach quite different from other languages)
- A huge array of packages and ancillary features. GIS, machine learning, databases, version control, websites and apps
- Produce better documents (R Markdown)

# Learning R: Cons

- Unfamiliarity: Learning a new language can be tough. Go easy on yourself and R
- Tyranny of choice: R invariably has multiple ways to do the same thing. Example: Assignment with `<-` and/or `=`
  - Just pick whatever method/package looks easiest to you. And don't be afraid to mix and match

# RStudio = An IDE (Integrated Development Environment) for the R programming language



# Useful RStudio shortcuts

## 2 RUN CODE

### Search command history

Navigate command history

Move cursor to start of line

Move cursor to end of line

Change working directory

### Interrupt current command

### Clear console

Quit Session (desktop only)

### Restart R Session

### Run current line/selection

## Windows/Linux

**Ctrl+↑**

**↑/↓**

Home

End

Ctrl+Shift+H

**Esc**

**Ctrl+L**

Ctrl+Q

**Ctrl+Shift+F10**

**Ctrl+Enter**

## Mac

**Cmd+↑**

**↑/↓**

Cmd+←

Cmd+→

Ctrl+Shift+H

**Esc**

**Ctrl+L**

Cmd+Q

**Cmd+Shift+F10**

**Cmd+Enter**

Source: [http://www.jeroenclaes.be/statistics\\_for\\_linguistics/cheatsheets/rstudio.pdf](http://www.jeroenclaes.be/statistics_for_linguistics/cheatsheets/rstudio.pdf)

# Dipping our toes in

[RStudio live session]

Download the R script here:

[https://www747.github.io/NYU\\_F23/Lab02/r\\_intro\\_basics\\_script.R](https://www747.github.io/NYU_F23/Lab02/r_intro_basics_script.R)

# R In Two Pages: Page 1/2

## Base R Cheat Sheet

### Getting Help

#### Accessing the help files

**?mean**

Get help of a particular function.

**help.search('weighted mean')**

Search the help files for a word or phrase.

**help(package = 'dplyr')**

Find help for a package.

#### More about an object

**str(iris)**

Get a summary of an object's structure.

**class(iris)**

Find the class an object belongs to.

### Using Packages

**install.packages('dplyr')**

Download and install a package from CRAN.

**library(dplyr)**

Load the package into the session, making all its functions available to use.

**dplyr::select**

Use a particular function from a package.

**data(iris)**

Load a built-in dataset into the environment.

### Working Directory

**getwd()**

Find the current working directory (where inputs are found and outputs are sent).

**setwd('C:/file/path')**

Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

### Vectors

#### Creating Vectors

<code>c(2, 4, 6)</code>	<code>2 4 6</code>	Join elements into a vector
<code>2:6</code>	<code>2 3 4 5 6</code>	An integer sequence
<code>seq(2, 3, by=0.5)</code>	<code>2.0 2.5 3.0</code>	A complex sequence
<code>rep(1:2, times=3)</code>	<code>1 2 1 2 1 2</code>	Repeat a vector
<code>rep(1:2, each=3)</code>	<code>1 1 1 2 2 2</code>	Repeat elements of a vector

#### Vector Functions

<b>sort(x)</b>	<b>rev(x)</b>
Return x sorted.	Return x reversed.
<b>table(x)</b>	<b>unique(x)</b>
See counts of values.	See unique values.

#### Selecting Vector Elements

##### By Position

<code>x[4]</code>	The fourth element.
<code>x[-4]</code>	All but the fourth.
<code>x[2:4]</code>	Elements two to four.
<code>x[-(2:4)]</code>	All elements except two to four.
<code>x[c(1, 5)]</code>	Elements one and five.

##### By Value

<code>x[x == 10]</code>	Elements which are equal to 10.
<code>x[x &lt; 0]</code>	All elements less than zero.
<code>x[x %in% c(1, 2, 5)]</code>	Elements in the set 1, 2, 5.

##### Named Vectors

<code>x['apple']</code>	Element with name 'apple'.
-------------------------	----------------------------

### Programming

#### For Loop

```
for (variable in sequence){
  Do something
}
```

##### Example

```
for (i in 1:4){
  j <- i + 10
  print(j)
}
```

#### If Statements

```
if (condition){
  Do something
} else {
  Do something different
}
```

##### Example

```
if (i > 3){
  print('Yes')
} else {
  print('No')
}
```

#### While Loop

```
while (condition){
  Do something
}
```

##### Example

```
while (i < 5){
  print(i)
  i <- i + 1
}
```

#### Functions

```
function_name <- function(var){
  Do something
  return(new_variable)
}
```

##### Example

```
square <- function(x){
  squared <- x*x
  return(squared)
}
```

### Reading and Writing Data

Also see the **readr** package.

Input	Output	Description
<code>df &lt;- read.table('file.txt')</code>	<code>write.table(df, 'file.txt')</code>	Read and write a delimited text file.
<code>df &lt;- read.csv('file.csv')</code>	<code>write.csv(df, 'file.csv')</code>	Read and write a comma separated value file. This is a special case of read.table/write.table.
<code>load('file.Rdata')</code>	<code>save(df, file = 'file.Rdata')</code>	Read and write an R data file, a file type special for R.

Conditions	a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
	a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null

Source: <https://iqss.github.io/dss-workshops/R/Rintro/base-r-cheat-sheet.pdf>



# R In Two Pages: Page 2/2




## Types

Converting between common data types in R. Can always go from a higher value in the table to a lower value.

<b>as.logical</b>	TRUE, FALSE, TRUE	Boolean values (TRUE or FALSE).
<b>as.numeric</b>	1, 0, 1	Integers or floating point numbers.
<b>as.character</b>	'1', '0', '1'	Character strings. Generally preferred to factors.
<b>as.factor</b>	'1', '0', '1', levels: '1', '0'	Character strings with preset levels. Needed for some statistical models.

## Matrices

`m <- matrix(x, nrow = 3, ncol = 3)`  
Create a matrix from x.

 <code>m[2, ]</code> - Select a row	<code>t(m)</code> Transpose
 <code>m[, 1]</code> - Select a column	<code>m %*% n</code> Matrix Multiplication
 <code>m[2, 3]</code> - Select an element	<code>solve(m, n)</code> Find x in: $m \cdot x = n$

## Strings

Also see the **stringr** package.

<code>paste(x, y, sep = ' ')</code>	Join multiple vectors together.
<code>paste(x, collapse = ' ')</code>	Join elements of a vector together.
<code>grep(pattern, x)</code>	Find regular expression matches in x.
<code>gsub(pattern, replace, x)</code>	Replace matches in x with a string.
<code>toupper(x)</code>	Convert to uppercase.
<code>tolower(x)</code>	Convert to lowercase.
<code>nchar(x)</code>	Number of characters in a string.

## Factors

<code>factor(x)</code>	Turn a vector into a factor. Can set the levels of the factor and the order.
<code>cut(x, breaks = 4)</code>	Turn a numeric vector into a factor by 'cutting' into sections.

## Statistics

<code>lm(y ~ x, data=df)</code> Linear model.	<code>t.test(x, y)</code> Perform a t-test for difference between means.	<code>prop.test</code> Test for a difference between proportions.
<code>glm(y ~ x, data=df)</code> Generalised linear model.	<code>pairwise.t.test</code> Perform a t-test for paired data.	<code>aov</code> Analysis of variance.
<code>summary</code> Get more detailed information out a model.		

## Maths Functions

<code>log(x)</code>	Natural log.	<code>sum(x)</code>	Sum.
<code>exp(x)</code>	Exponential.	<code>mean(x)</code>	Mean.
<code>max(x)</code>	Largest element.	<code>median(x)</code>	Median.
<code>min(x)</code>	Smallest element.	<code>quantile(x)</code>	Percentage quantiles.
<code>round(x, n)</code>	Round to n decimal places.	<code>rank(x)</code>	Rank of elements.
<code>signif(x, n)</code>	Round to n significant figures.	<code>var(x)</code>	The variance.
<code>cor(x, y)</code>	Correlation.	<code>sd(x)</code>	The standard deviation.

## Lists

`l <- list(x = 1:5, y = c('a', 'b'))`  
A list is a collection of elements which can be of different types.

<code>l[[2]]</code> Second element of l.	<code>l[1]</code> New list with only the first element.	<code>l\$x</code> Element named x.	<code>l['y']</code> New list with only element named y.
---	--	---------------------------------------	--



## Data Frames

Also see the **dplyr** package.

`df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))`  
A special case of a list where all elements are the same length.

x	y
1	a
2	b
3	c




**List subsetting**

`df$x`  `df[[2]]` 

**Understanding a data frame**

`View(df)` See the full data frame.  
`head(df)` See the first 6 rows.

**Matrix subsetting**

`df[, 2]`   
`df[2, ]`   
`df[2, 2]` 

`nrow(df)` Number of rows.  
`ncol(df)` Number of columns.  
`dim(df)` Number of columns and rows.

`cbind` - Bind columns.  
`rbind` - Bind rows.

## Variable Assignment

```
> a <- 'apple'
> a
[1] 'apple'
```

## The Environment

<code>ls()</code>	List all variables in the environment.
<code>rm(x)</code>	Remove x from the environment.
<code>rm(list = ls())</code>	Remove all variables from the environment.

You can use the environment panel in RStudio to browse variables in your environment.

## Distributions

	Random Variates	Density Function	Cumulative Distribution	Quantile
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>	<code>qnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>	<code>qpois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>	<code>qbinom</code>
Uniform	<code>runif</code>	<code>dunif</code>	<code>punif</code>	<code>qunif</code>

## Plotting

Also see the **ggplot2** package.

<code>plot(x)</code> Values of x in order.	<code>plot(x, y)</code> Values of x against y.	<code>hist(x)</code> Histogram of x.
---	---	---

## Dates

See the **lubridate** package.

Source: <https://iqss.github.io/dss-workshops/R/Rintro/base-r-cheat-sheet.pdf>

# Other resources

- Basic R in 26 pages: <https://www.zeileis.org/teaching/AER/Ch-Basics-handout.pdf>
- If you prefer a structured book: *Using R for Introductory Econometrics*, 2<sup>nd</sup> ed by Florian Heiss  
<http://www.urfie.net/>

# Data Structures, Packages, Plots, Read/Write Files

[RStudio live session]

[https://www747.github.io/NYU\\_F23/Lab02/r\\_intro\\_data\\_structures\\_plots\\_packages\\_read\\_write.html](https://www747.github.io/NYU_F23/Lab02/r_intro_data_structures_plots_packages_read_write.html)

# Sampling and Probability

[RStudio live session]

- Properties of the sample mean
  - Estimator bias, consistency, efficiency

[https://www747.github.io/NYU\\_F23/Lab02/r\\_intro\\_sampling\\_probability\\_bias\\_v\\_consistency\\_v\\_efficiency.html](https://www747.github.io/NYU_F23/Lab02/r_intro_sampling_probability_bias_v_consistency_v_efficiency.html)

# R Markdown (.Rmd files)

- Rmd files · Develop your code and ideas side-by-side in a single document. Run code as individual chunks or as an entire document. Dynamic Documents
- Knit together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS Powerpoint. Reproducible Research
- Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.

The syntax on the left renders as the output on the right.

Plain text.

End a line with two spaces to start a new paragraph.

Also end with a backslash\ to make a new line.

*\*italics\** and **\*\*bold\*\***

superscript<sup>2</sup>/subscript<sub>2</sub>

~~~strikethrough~~~

escaped: `* _ \`

endash: --, emdash: ---

# Header 1

## Header 2

...

##### Header 6

- unordered list

- item 2

- item 2a (indent 1 tab)

- item 2b

1. ordered list

2. item 2

- item 2a (indent 1 tab)

- item 2b

<link url>

[This is a link.](link url)

[This is another link][id].

Plain text.

End a line with two spaces to start a new paragraph.

Also end with a backslash\ to make a new line.

*italics* and **bold**

superscript<sup>2</sup>/subscript<sub>2</sub>

~~strikethrough~~

escaped: `* _ \`

endash: –, emdash: —

**Header 1**  
**Header 2**

...  
**Header 6**

• unordered list

• item 2

• item 2a (indent 1 tab)

• item 2b

1. ordered list

2. item 2

• item 2a (indent 1 tab)

• item 2b

<http://www.rstudio.com/>

This is a link.

This is another link.

Source: <https://evoldyn.gitlab.io/evomics-2018/ref-sheets/rmarkdown-cheatsheet-2.0.pdf>

# Next Steps

# Next steps

- Next week: statistics review
- Homework 1 will be posted tomorrow due Sep 26
- Group signups due Sep 27. (I'll post a signup sheet next week)

| Project Requirement | Date Due      |
|---------------------|---------------|
| Group Signup        | Sep 27        |
| Problem Statement   | Oct 11        |
| Model Description   | Nov 1         |
| Presentation        | Dec 6         |
| <b>Final Report</b> | <b>Dec 18</b> |