

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 14, 2022

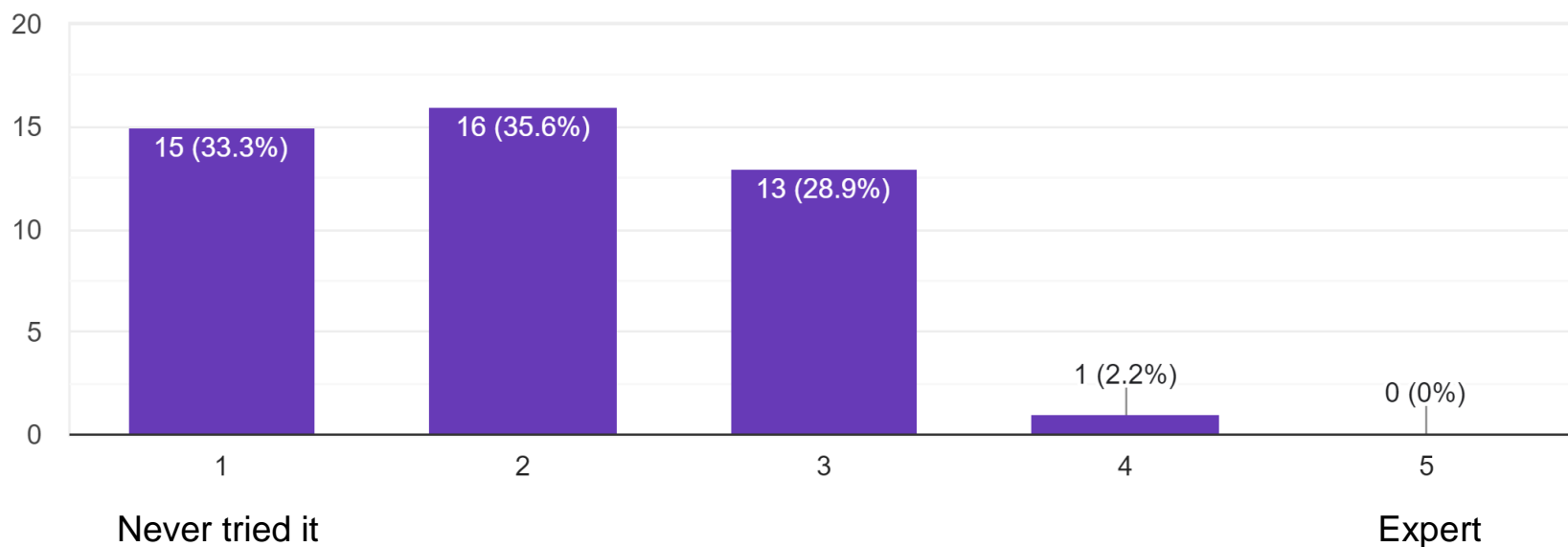
Today

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

About You: Software experience

R

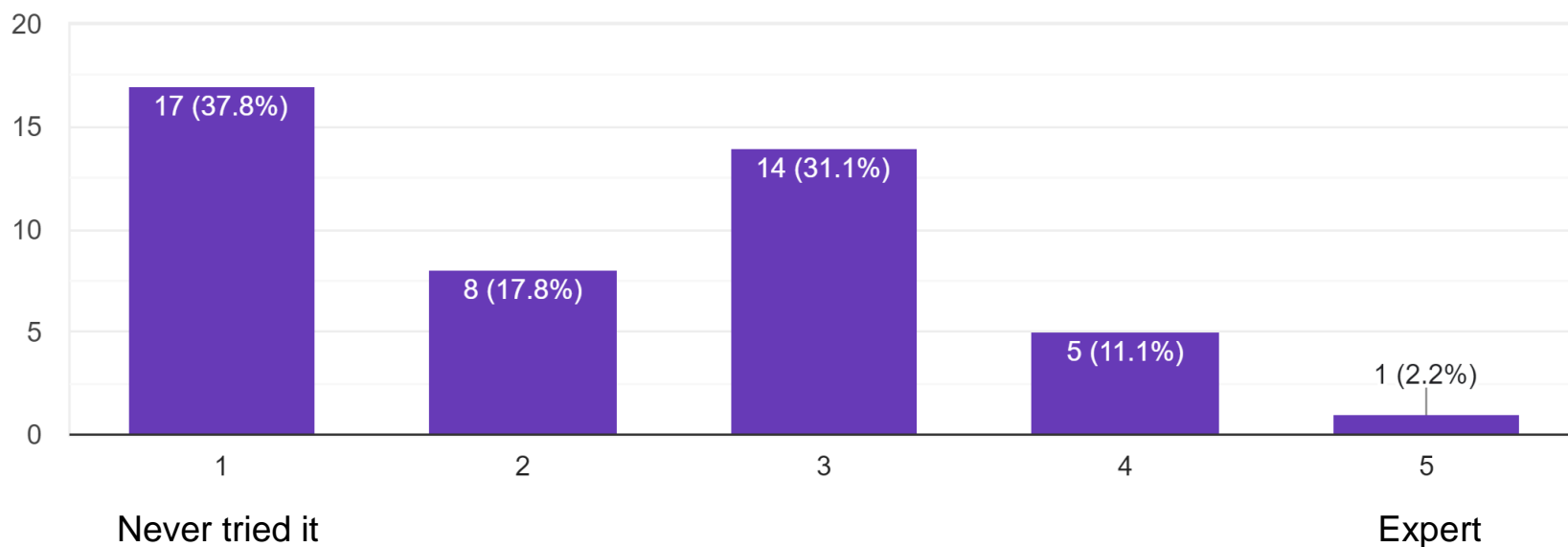
45 responses



About You: Software experience

Stata

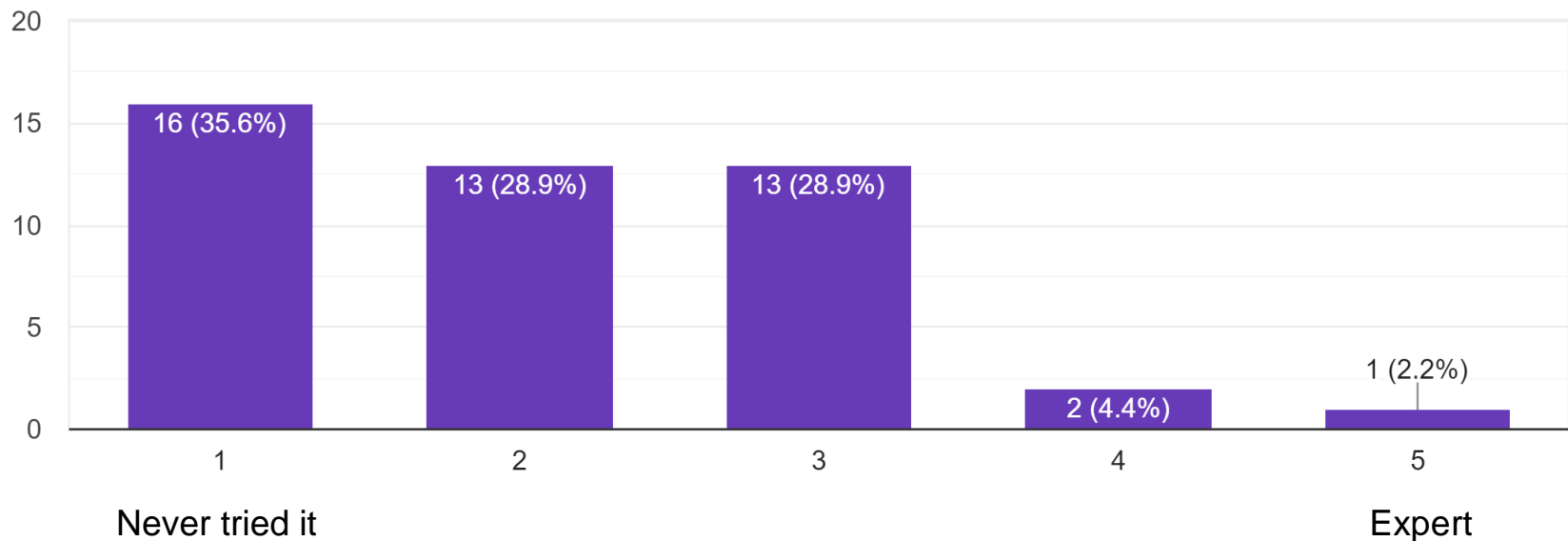
45 responses



About You: Software experience

Python

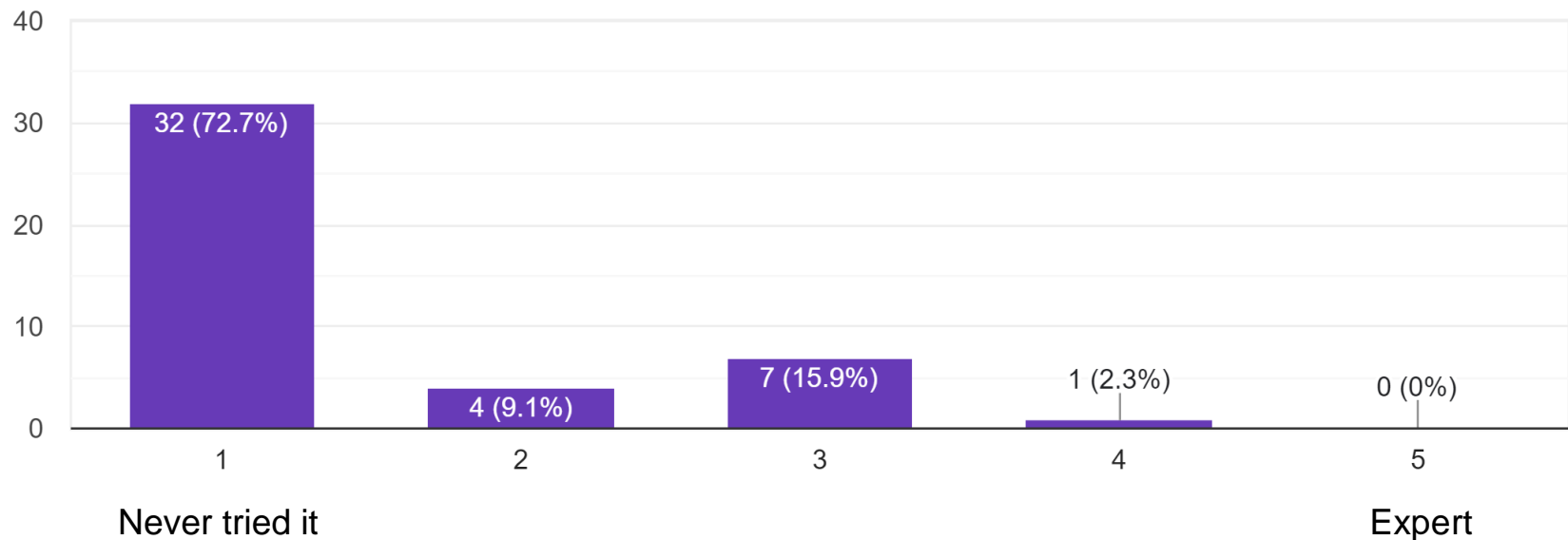
45 responses



About You: Software experience

LaTeX

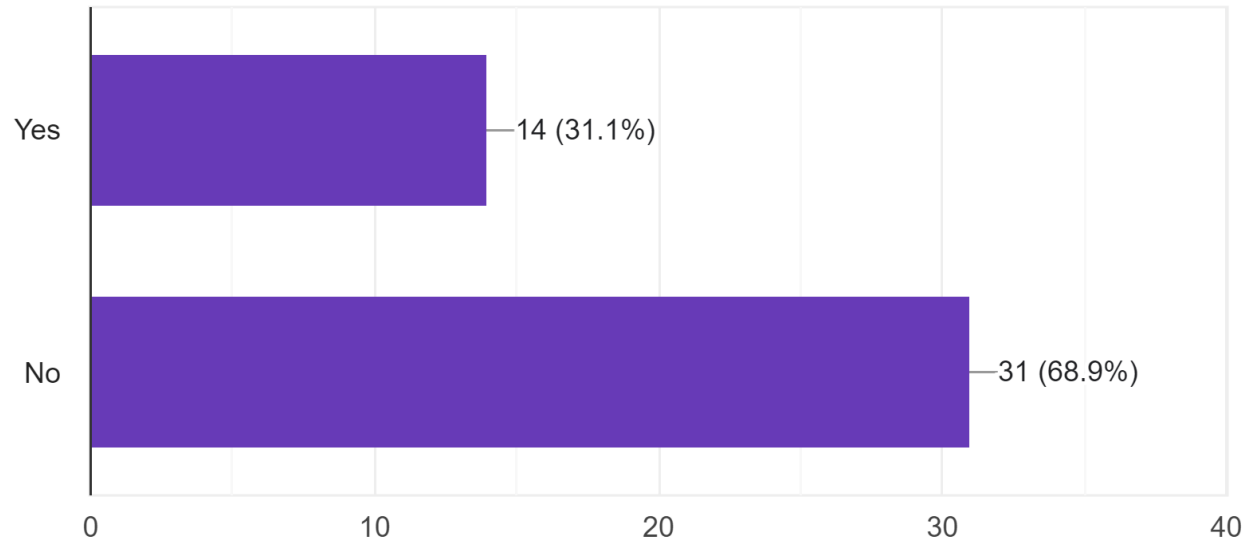
44 responses



About You

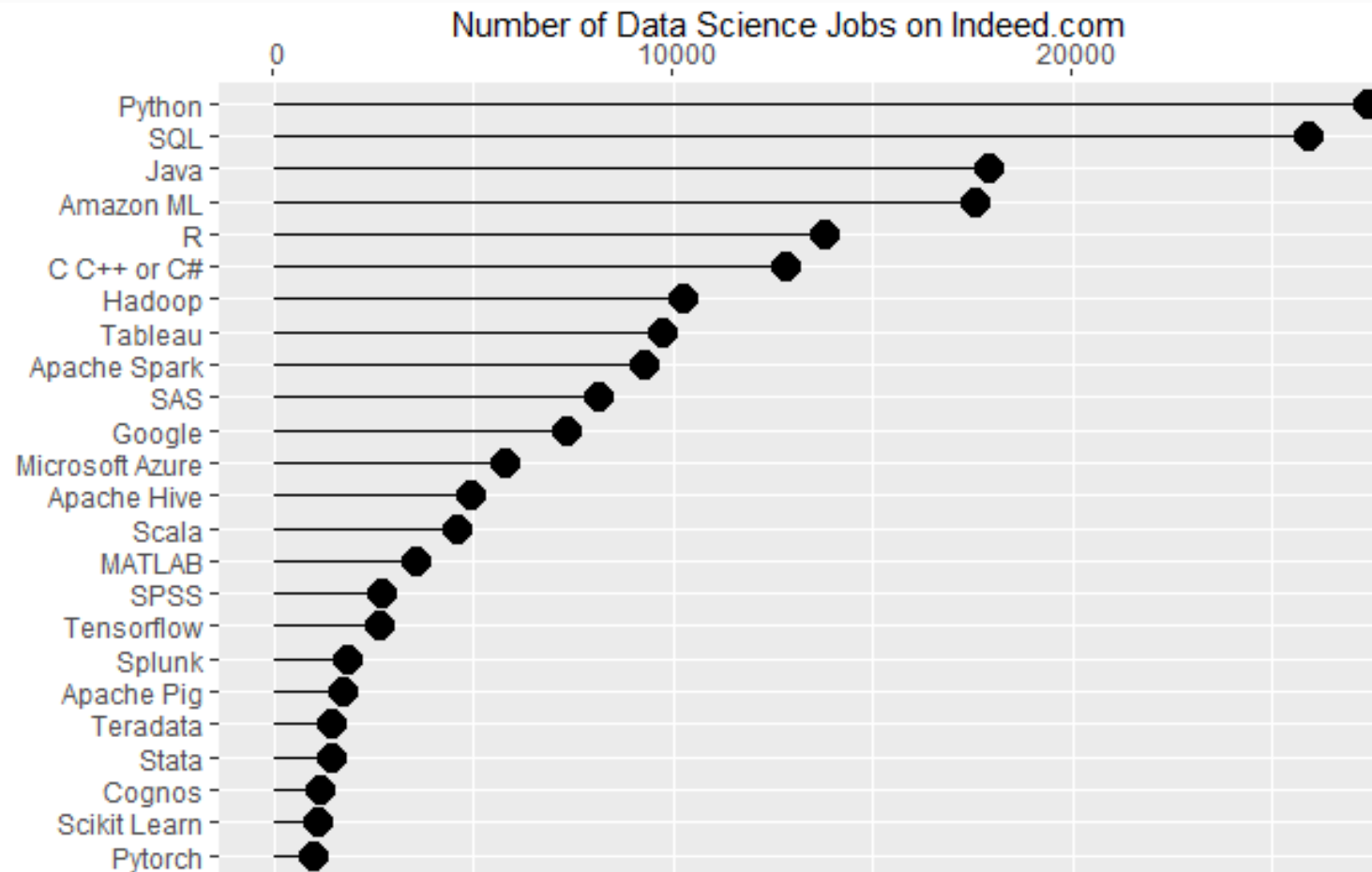
This is my first econometrics course

45 responses



Intro to R

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



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

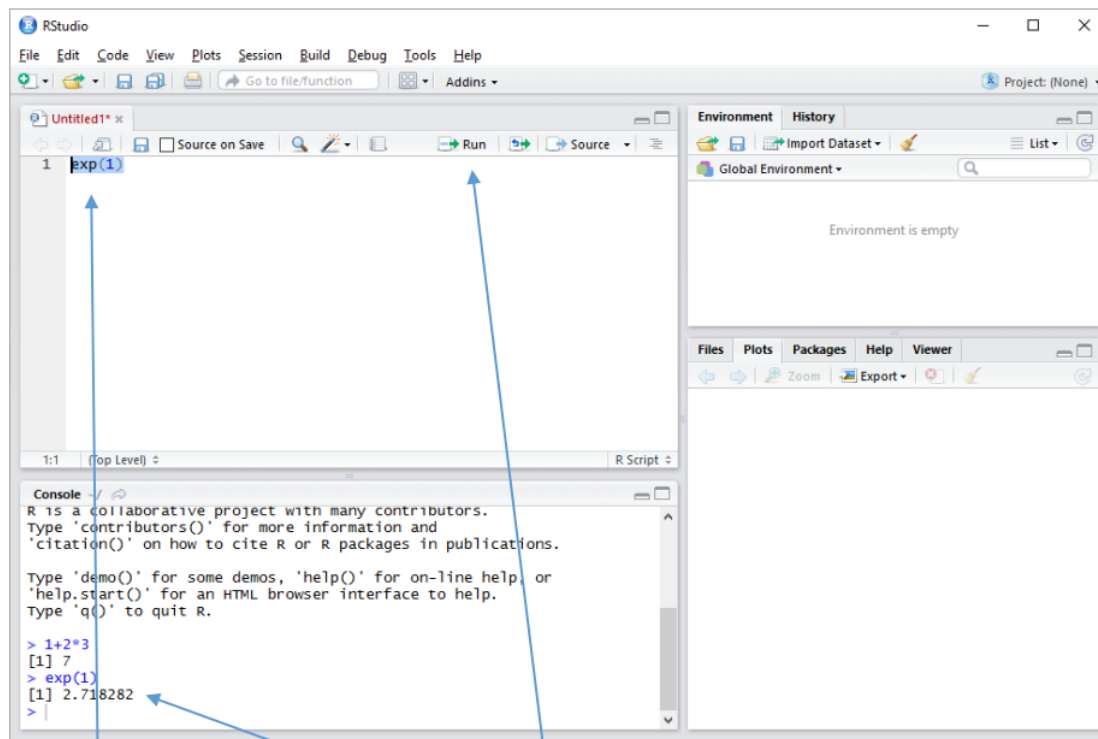
Learning R: Pros

- 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

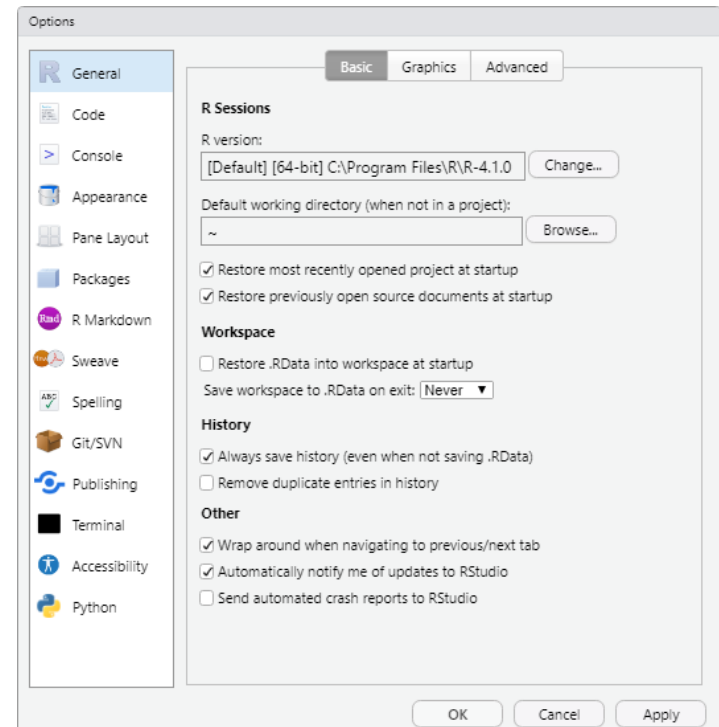
RStudio = An IDE (Integrated Development Environment) for the R programming language



1. Type your command here. Highlight it.

2. Click the 'Run' button, or press
CTRL+Enter on your keyboard.

3. Output displayed here.



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

Dipping our toes in

[RStudio live session]

Download the R script here:

https://www.wedu.github.io/NYU_F22/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

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

Vector Functions

sort(x)	rev(x)
Return x sorted.	Return x reversed.
table(x)	unique(x)
See counts of values.	See unique values.

Selecting Vector Elements

By Position

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

By Value

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

Named Vectors

x['apple']	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
df <- read.table('file.txt')	write.table(df, 'file.txt')	Read and write a delimited text file.
df <- read.csv('file.csv')	write.csv(df, 'file.csv')	Read and write a comma separated value file. This is a special case of read.table/write.table.
load('file.Rdata')	save(df, file = 'file.Rdata')	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: <http://github.com/rstudio/cheatsheets/raw/master/base-r.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.

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

Maths Functions

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

Variable Assignment

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

The Environment

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

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

Matrices

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

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

Lists

```
l <- list(x = 1:5, y = c('a', 'b'))
```

A list is a collection of elements which can be of different types.

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

Strings

Also see the **stringr** package.

```
paste(x, y, sep = ' ')
```

Join multiple vectors together.

```
paste(x, collapse = ' ')
```

Join elements of a vector together.

```
grep(pattern, x)
```

Find regular expression matches in x.

```
gsub(pattern, replace, x)
```

Replace matches in x with a string.

```
toupper(x)
```

Convert to uppercase.

```
tolower(x)
```

Convert to lowercase.

```
nchar(x)
```

Number of characters in a string.

Factors

```
factor(x)
```

Turn a vector into a factor. Can set the levels of the factor and the order.

```
cut(x, breaks = 4)
```

Turn a numeric vector into a factor by 'cutting' into sections.

Statistics

```
lm(y ~ x, data=df)
```

Linear model.

```
glm(y ~ x, data=df)
```

Generalised linear model.

```
summary
```

Get more detailed information out a model.

```
t.test(x, y)
```

Perform a t-test for difference between means.

```
prop.test
```

Test for a difference between proportions.

```
pairwise.t.test
```

Perform a t-test for paired data.

```
aov
```

Analysis of variance.

Distributions

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

Plotting

Also see the **ggplot2** package.

```
plot(x)
```

Values of x in order.

```
plot(x, y)
```

Values of x against y.

```
hist(x)
```

Histogram of x.

Dates

See the **lubridate** package.

Source: <http://github.com/rstudio/cheatsheets/raw/master/base-r.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*, 2nd ed by Florian Heiss
<http://www.urfie.net/>

Data Structures, Packages, Plots, Read/Write Files

[RStudio live session]

https://www.wedu.github.io/NYU_F22/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://www.wedu.github.io/NYU_F22/Lab02/r_intro_sampling_probability_bias_v_consistency_v_efficiency

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^{^2^}/subscript_{~2~}

~~~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://github.com/rstudio/cheatsheets/raw/master/rmarkdown-2.0.pdf>

# Next Steps

# Next steps

- Next week: statistics review
- Homework 1 due Sep 25
- Group signups due Sep 28. (I'll post signup sheet next week)

| Project Requirement | Date Due       |
|---------------------|----------------|
| Group Signup        | Sep 28         |
| Problem Statement   | Oct 12         |
| Model Description   | Nov 2          |
| Presentation        | Nov 30 – Dec 7 |
| <b>Final Report</b> | <b>Dec 19</b>  |