WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Dr. Iman Ahmadi
Assistant Prof. of Marketing

Marketing & Strategy Analytics:

Introduction to R

Overview

- **Practical part (in our workshop):**
  - Introduction to R and RStudio
  - The Basics of the R Language: Using the Software

Marketing Analytics and Value Creation

How do Real-World Companies Deploy Marketing Analytics? — Netflix (I/II)

Netflix, an online streaming movie and TV service:

- has more than 220 million streaming subscribers (2022)
- in over 50 countries
- enjoying more than 100 million hours of TV shows and movies a day

The main question:

How do Real-World Companies Deploy Marketing Analytics? — Netflix (I/II)

Netflix, an online streaming movie and TV service:

- has more than 220 million streaming subscribers (2022)
- in over 50 countries
- enjoying more than 100 million hours of TV shows and movies a day

The main question:

What the customer will enjoy watching?

(link)

- In 2006: Netflix Prize, $1 million for the best algorithm for predicting how their customers would rate a movie based on their previous ratings

Discussion:
How Netflix Realizes what you Enjoy Watching Next?

How do Real-World Companies Deploy Marketing Analytics? — Netflix (II/II)

- Netflix moved towards positioning itself as a content creator, not just a distributor
- This strategy was firmly driven by their data – e.g., their subscribers have a appetite for content directed by David Fincher and starring Kevin Spacey

Discussion:
Where did the Value from Marketing Analytics Come from in the Netflix Example?

How do Real-World Companies Deploy Marketing Analytics? — Airbnb (I/II)

Airbnb, website that connects travelers with available accommodation around the world, launched in 2008:

- more than 5 million listings worldwide[1]
- in over 220 countries
- served more than 150 million registered users

The main question:

How to connect large volumes of guests with those who have accommodation to offer?

- Requires an understanding of:
  - hosts’ and guests’ preferences,
  - availability of the right sort of properties,
    - in right place
    - for the right price

[1] https://www.thezebra.com/resources/home/airbnb-statistics/#infographic

How do Real-World Companies Deploy Marketing Analytics? — Airbnb (II/II)

Data:

- Internal: Mixture of structured and unstructured formats:
  - image data from host photos,
  - location data,
  - accommodation features (number of rooms/beds, Wi-Fi, hot tub, etc.),
  - customer feedback and ratings,
  - transaction data, etc.

- External

What would be examples of external data for Airbnb?

Marketing Analytics not Just for Big Businesses!

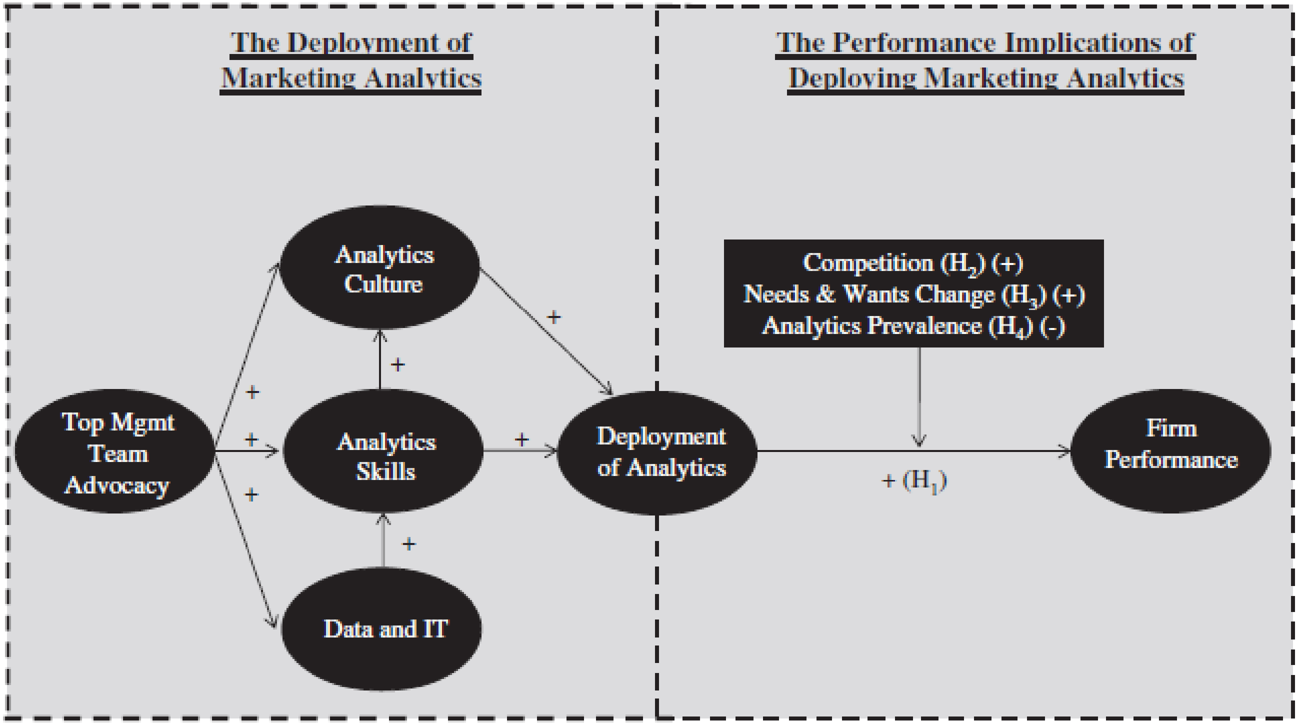
(link)

Innovation with Marketing Analytics!

- Lexus has revealed an ad that has been entirely scripted by artificial intelligence (AI)

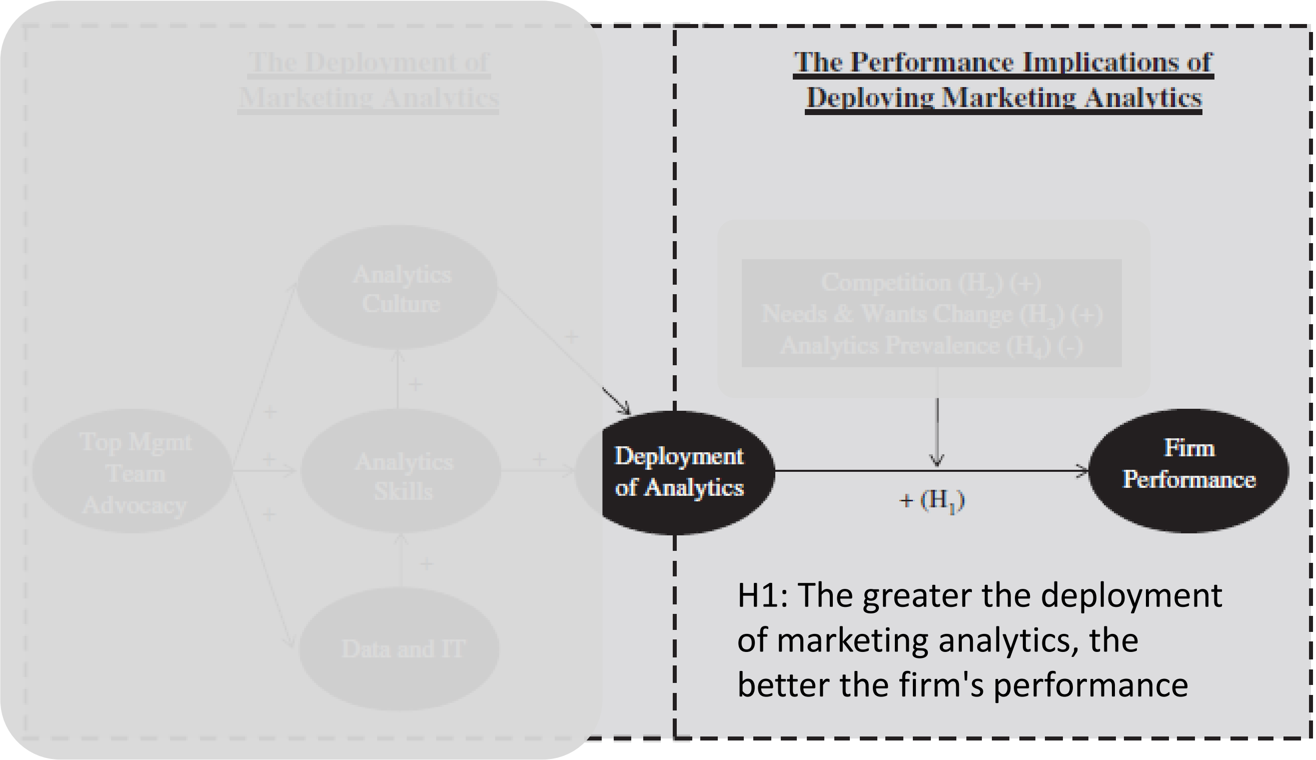
(link)

Marketing Analytics and Value Creation: The Marketing Analytics Chain of Effects


The Deployment of Marketing Analytics
The Performance Implications of Deploying Marketing Analytics
Analytics Culture
Top Mgmt Team Advocacy
Analytics Skills
Data and IT
Deployment of Analytics
Competition (H_2) (+)
Needs & Wants Change (H_3) (+)
Analytics Prevalence (H_4) (-)
Firm Performance
+
+
+
+
+
+
+
+ (H_1)

Germann, F., Lilien, G.L. and Rangaswamy, A. (2013) ‘Performance Implications of Deploying Marketing Analytics’, *International Journal of Research in Marketing*, 30(2), pp. 114–128.

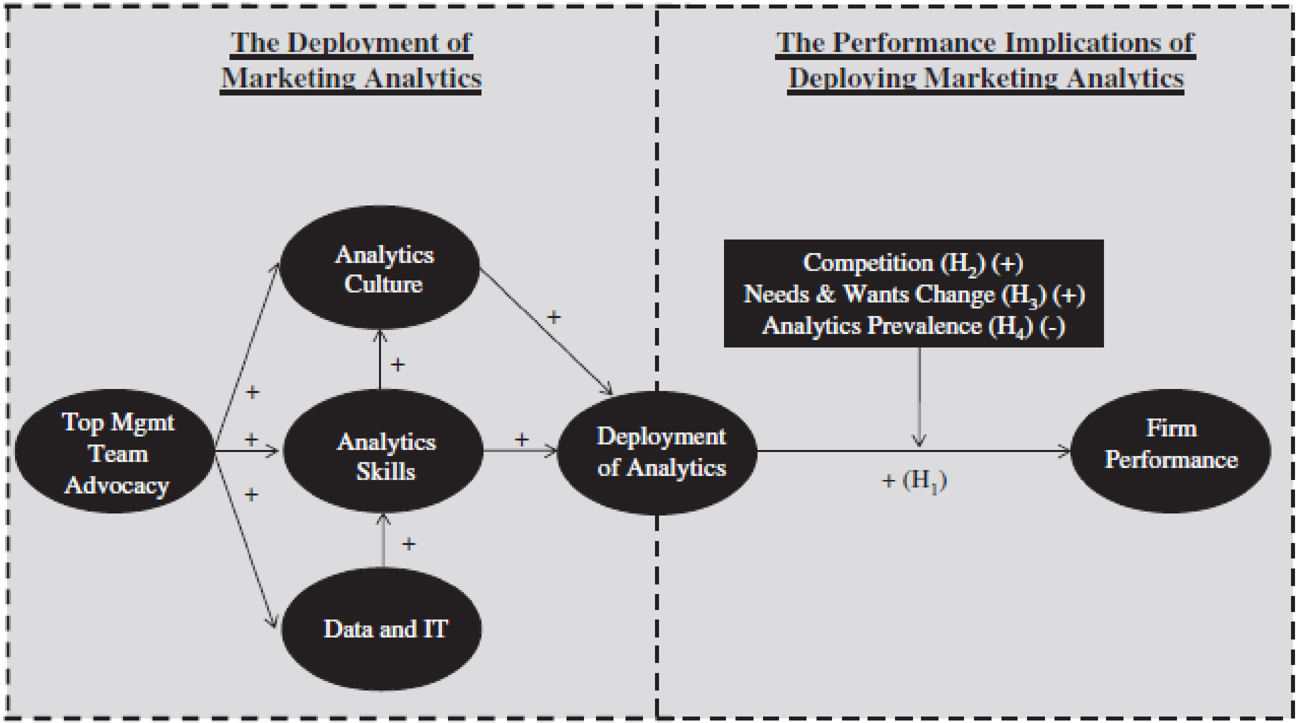
Marketing Analytics and Value Creation: The Marketing Analytics Chain of Effects


The Deployment of Marketing Analytics
Analytics Culture
Top Mgmt Team Advocacy
Analytics Skills
Data and IT
+
+
+
+
+
+
+
The Performance Implications of Deploying Marketing Analytics
Competition (H_2) (+)
Needs & Wants Change (H_3) (+)
Analytics Prevalence (H_4) (-)
Deployment of Analytics
Firm Performance
+ (H_1)

H1: The greater the deployment of marketing analytics, the better the firm's performance

Germann, F., Lilien, G.L. and Rangaswamy, A. (2013) ‘Performance Implications of Deploying Marketing Analytics’, *International Journal of Research in Marketing*, 30(2), pp. 114–128.

Marketing Analytics and Value Creation: The Marketing Analytics Chain of Effects


The Deployment of Marketing Analytics
The Performance Implications of Deploying Marketing Analytics
Analytics Culture
Top Mgmt Team Advocacy
Analytics Skills
Data and IT
Deployment of Analytics
Competition (H2) (+)
Needs & Wants Change (H3) (+)
Analytics Prevalence (H4) (-)
Firm Performance
+ (H1)
+

Germann, F., Lilien, G.L. and Rangaswamy, A. (2013) 'Performance Implications of Deploying Marketing Analytics', *International Journal of Research in Marketing*, 30(2), pp. 114–128.

Marketing Analytics and Value Creation: Performance Implication Considerations

- The deployment of marketing analytics can help to **better match** products and services to customer needs and wants (competitive advantage)

- Implication: Marketing Analytics can help to improve firm performance

- When should we expect the positive effects of marketing analytics to be biggest?
  - The stronger the competition
  - The more rapidly consumer preferences change
  - The fewer competitors also use marketing analytics techniques

Marketing Analytics and Value Creation: Antecedents of Deployment Considerations

- The analytics culture of the firm is important
  - Is there support from the top management teams?
  - Critical for the transfer from insight generation to implementation

"In God we trust; all others must bring data."
(W. Edwards Deming)

- Access to data and IT resources

- **Level of marketing analytics skills**

The Basics of the R Language

What is R?

- “R is a language and environment for statistical computing and graphics” http://www.r-project.org/about.html

- R is:
  - A programming language (not a ‘statistics program’)
  - An environment for writing code and handling data
  - User extensible through add-ons and custom code
  - Open source and freely available
  - Where new statistics and analytics innovation most often happen first

What is R Not?

- R is not:
  - intended for point-and-click usage
  - designed to hide complexity from the user
  - necessarily a fast or easy way to do something that is new to you

- Like learning a non-computer human language, **R takes time, practice, patience, and application to real problems**

Why R (and RStudio)?

- If it is (somewhat) hard to learn, why is it worth it?
  - R is optimized for working with data and statistics
  - The programming language is relatively simple and flexible
  - 1,000s of people have contributed additions to R
  - Once you have mastered something in R, it is easy to automate it and become more and more productive
  - And, finally: R skills are in very high demand!

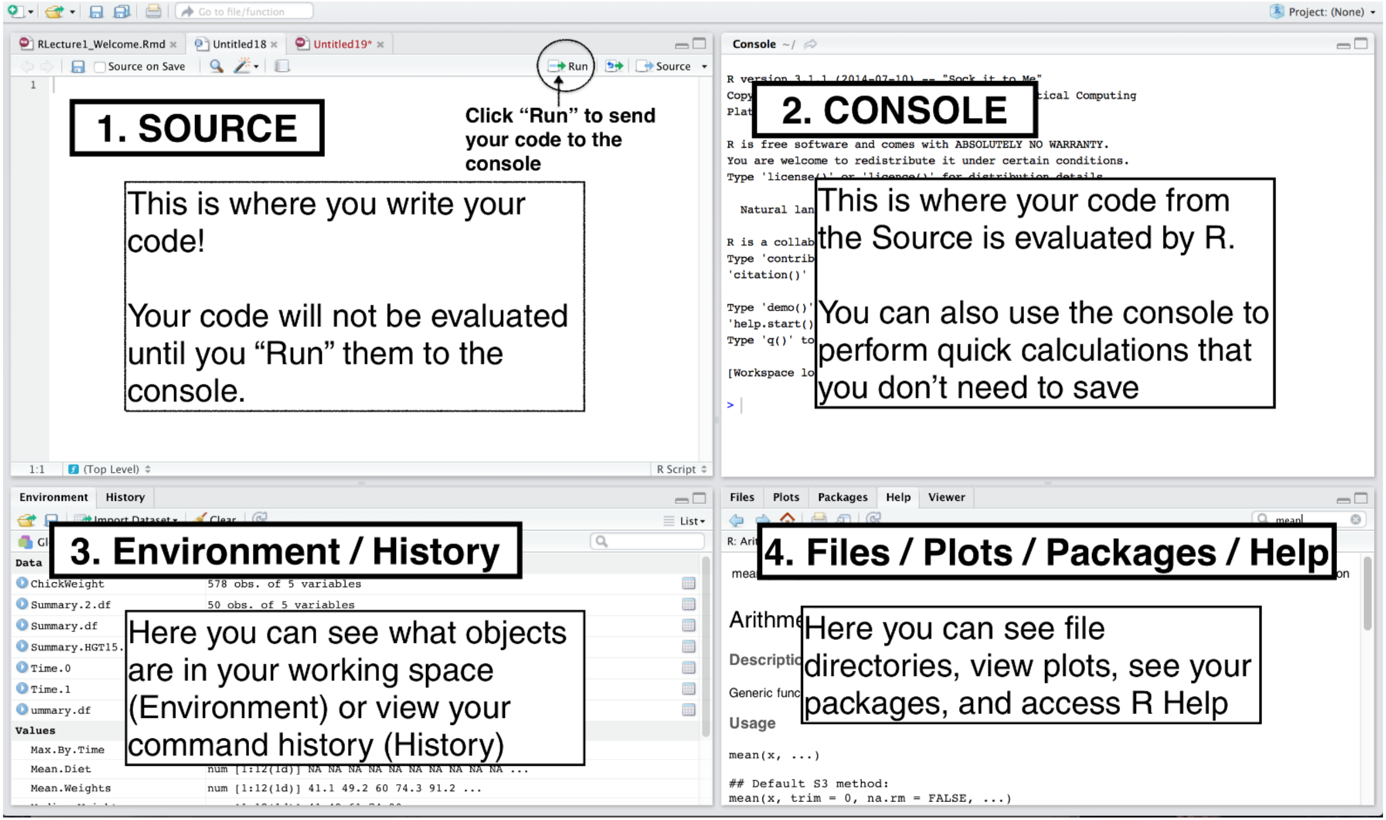
- RStudio in turn will make it much easier to work with R

The Basics of the R Language: Using the Software

Content of the First Practical Session

- Getting started
- The basic object types in R
  - these include variables and data sets (data frames)
- Where to get help with R
- How to summarize objects in R
- Sequencing and indexing
- Missing and interesting values
- Working with hypothetical store data
- Read and write CSVs
- Exercises

The Four Panes of RStudio



1. SOURCE

Click “Run” to send your code to the console

This is where you write your code!

Your code will not be evaluated until you “Run” them to the console.

2. CONSOLE

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copy                                         tical Computing
Plat

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 lan

R is a collab
Type 'contrib
'citation()'

Type 'demo()'
'help.start()
Type 'q()' to

[Workspace lo

>
```

This is where your code from the Source is evaluated by R.

You can also use the console to perform quick calculations that you don’t need to save

3. Environment / History

```
Data
ChickWeight          578 obs. of 5 variables
Summary.2.df         50 obs. of 5 variables
Summary.df
Summary.HGT15.
Time.0
Time.1
ummary.df
Values
  Max.By.Time
  Mean.Diet          num [1:12(1d)] NA NA NA NA NA NA NA NA NA NA ...
  Mean.Weights       num [1:12(1d)] 41.1 49.2 60 74.3 91.2 ...
```

Here you can see what objects are in your working space (Environment) or view your command history (History)

4. Files / Plots / Packages / Help

Arithme

Descripti

Generic func

Usage

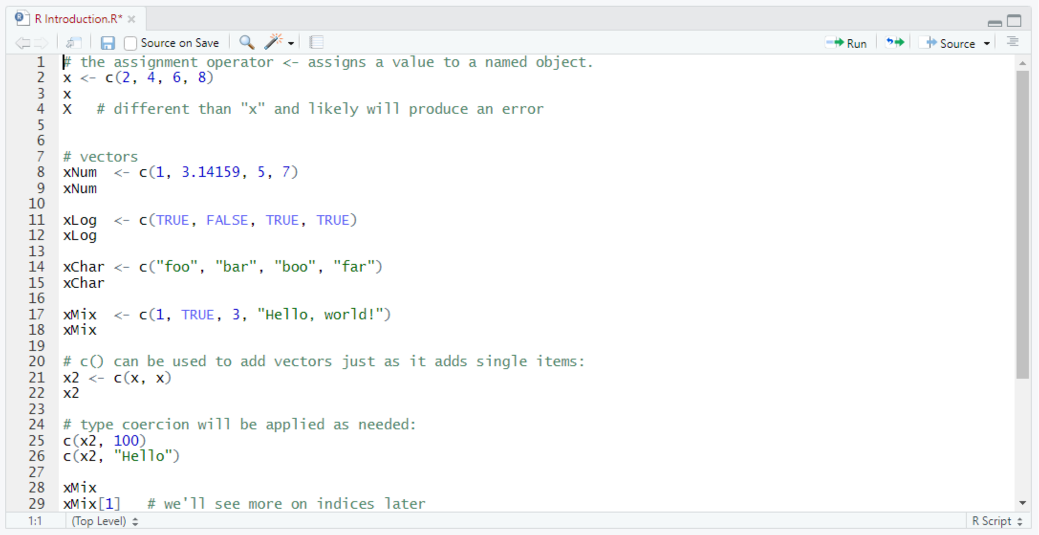
```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Here you can see file directories, view plots, see your packages, and access R Help

Phillips, N. D. (2017). Yarrr! The pirate’s guide to R. Chapter 2.

1. Source - Your Notepad for Code (I/II)

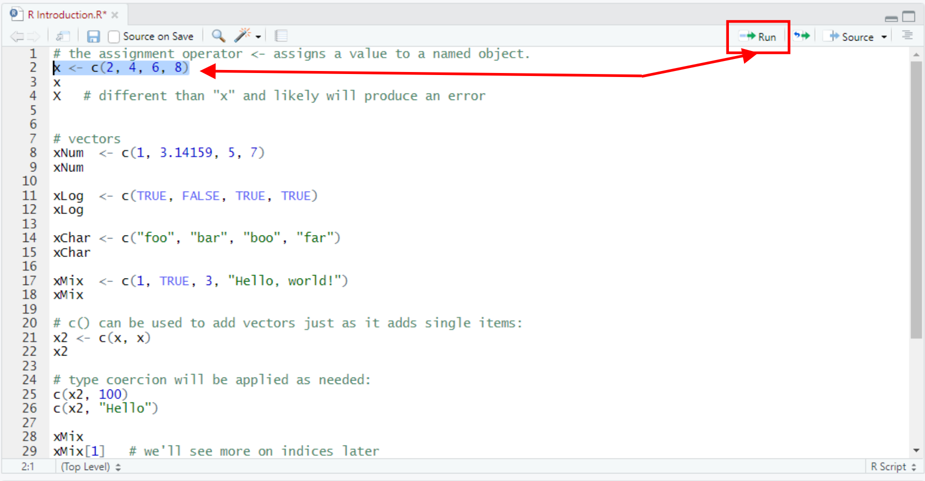
- The source pane is where you create and edit R Scripts - your collections of code
- When typing code in a script in the Source panel, R won’t evaluate the code as you type
- To have R actually evaluate your code, you need to first ‘send’ the code to the Console



```r
# the assignment operator <- assigns a value to a named object.
x <- c(2, 4, 6, 8)
x
X   # different than "x" and likely will produce an error


# vectors
xNum  <- c(1, 3.14159, 5, 7)
xNum

xLog  <- c(TRUE, FALSE, TRUE, TRUE)
xLog

xChar <- c("foo", "bar", "boo", "far")
xChar

xMix  <- c(1, TRUE, 3, "Hello, world!")
xMix

# c() can be used to add vectors just as it adds single items:
x2 <- c(x, x)
x2

# type coercion will be applied as needed:
c(x2, 100)
c(x2, "Hello")

xMix
xMix[1]   # we'll see more on indices later
```

1. Source - Your Notepad for Code (II/II)

- There are many ways to send your code from the Source to the console
  - The slowest way is to copy and paste
  - A faster way is to highlight the code you wish to evaluate and clicking on the “Run” button on the top right of the Source

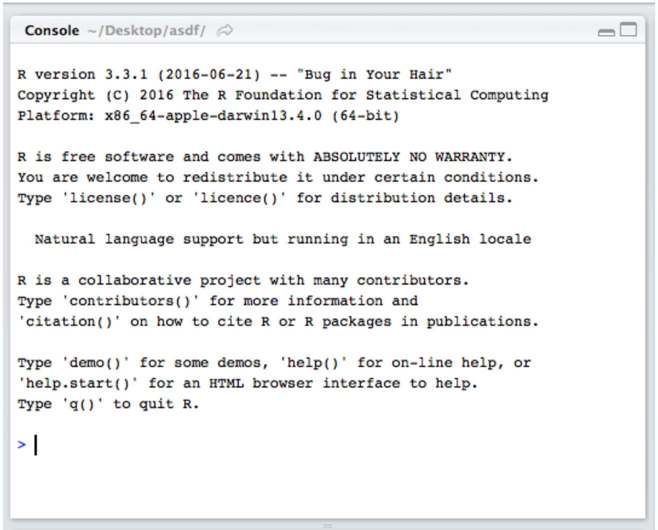


```r
# the assignment operator <- assigns a value to a named object.
x <- c(2, 4, 6, 8)
x
X   # different than "x" and likely will produce an error


# vectors
xNum  <- c(1, 3.14159, 5, 7)
xNum

xLog  <- c(TRUE, FALSE, TRUE, TRUE)
xLog

xChar <- c("foo", "bar", "boo", "far")
xChar

xMix  <- c(1, TRUE, 3, "Hello, world!")
xMix

# c() can be used to add vectors just as it adds single items:
x2 <- c(x, x)
x2

# type coercion will be applied as needed:
c(x2, 100)
c(x2, "Hello")

xMix
xMix[1]    # we'll see more on indices later
```

Tip: You can use the hot-key “Command + Return” on Mac, or “Control + Enter” on PC to send all highlighted code to the console

2. Console: R's Heart

- Console is the heart of R, where R actually evaluates code



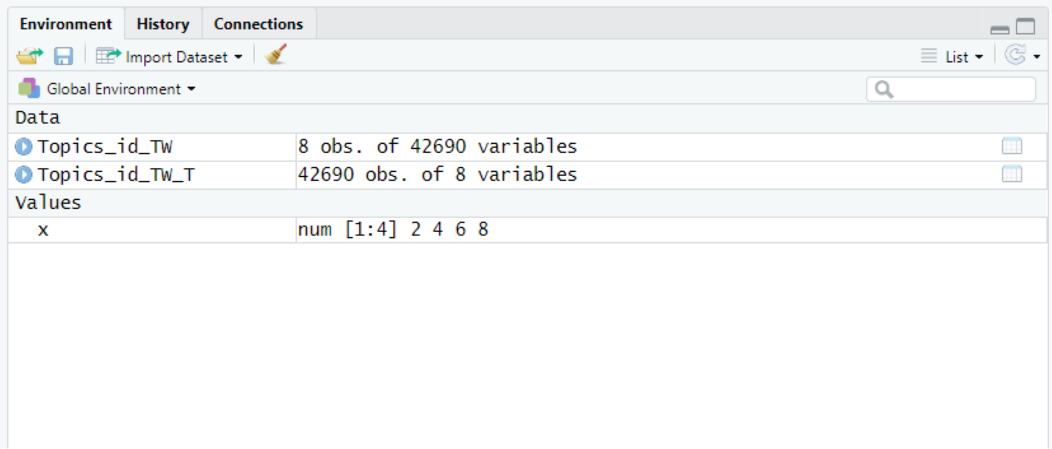
```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

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.

> |
```

- Try to write most of your code in a document in the Source:
  - if you type code into the console, **it won't be saved**
  - if you make a mistake in typing code into the console, you'd have to re-type everything all over again

3. Environment / History

- Environment tab of this panel shows
  - the names of all the data objects (like vectors, matrices, and data frames) that you've defined in your current R session
  - information like the number of observations and rows in data objects
  - a few clickable actions (e.g., Import Dataset)

- The History tab shows a history of all the code previously evaluated in the Console



```
Data
Topics_id_TW      8 obs. of 42690 variables
Topics_id_TW_T    42690 obs. of 8 variables
Values
  x               num [1:4] 2 4 6 8
```

4. Files / Plots / Packages / Help

- **Files**: gives access to the file directory on your hard drive. You can use it to set your working directory (“More” -> “Set As Working Directory”)

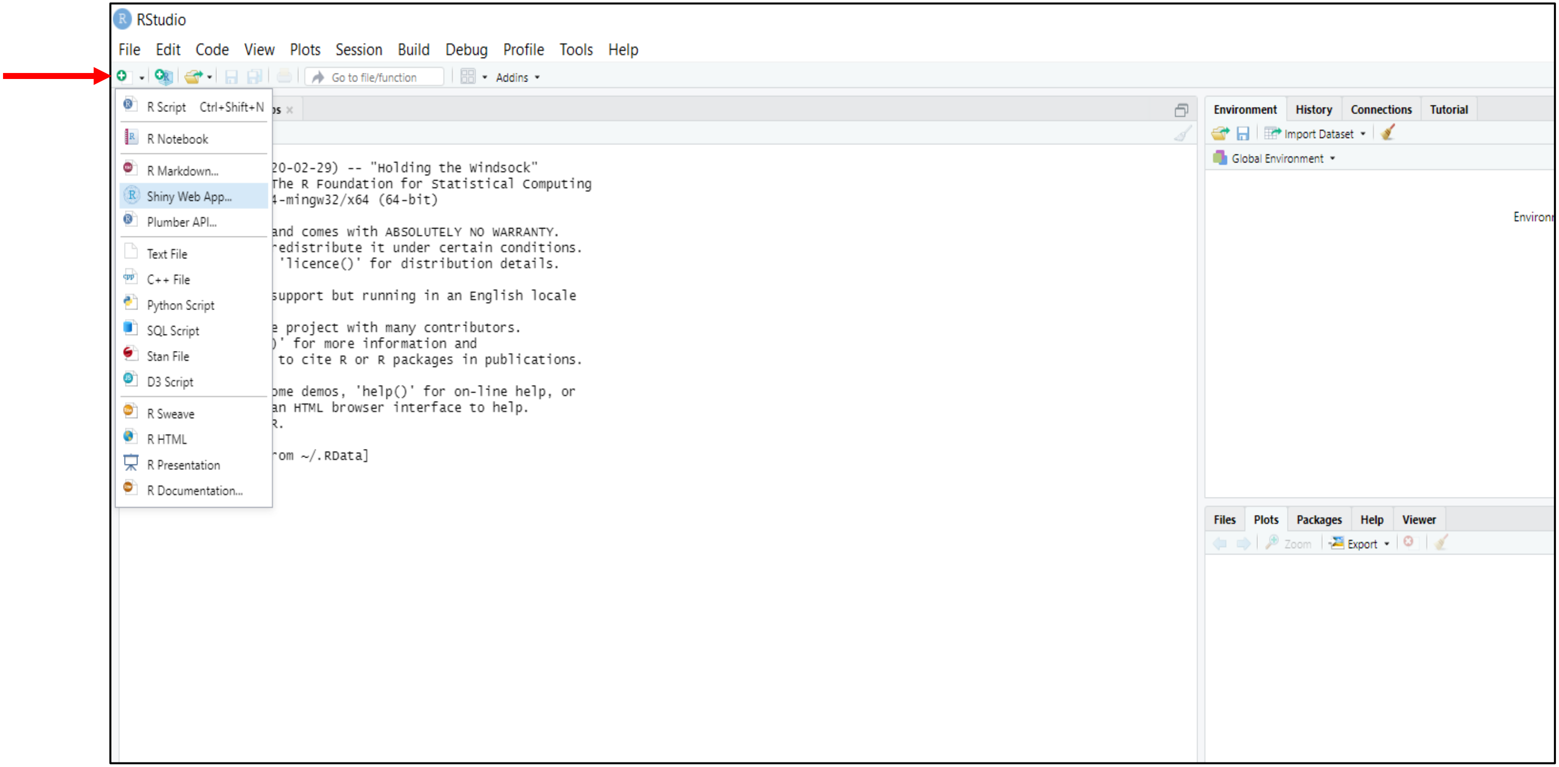
- **Plots**: shows all your plots

- **Packages**: a list of all the R packages installed and indicates whether or not they are currently loaded (indicated by: ✓)

- **Help**: menu for R functions.

Basic Objects

Open a New ‘R Script’



```
20-02-29) -- "Holding the Windsock"
The R Foundation for Statistical Computing
4-mingw32/x64 (64-bit)

and comes with ABSOLUTELY NO WARRANTY.
redistribute it under certain conditions.
 'licence()' for distribution details.

support but running in an English locale

e project with many contributors.
)' for more information and
 to cite R or R packages in publications.

ome demos, 'help()' for on-line help, or
an HTML browser interface to help.
R.

rom ~/.RData]
```

Basic Objects

(“R Introduction" R code)

The assignment operator <- assigns a value to a named object.

Source

```
x <- c(2, 4, 6, 8)
x
```

```
[1] 2 4 6 8
```

Console

Object names are case sensitive. Instead of 'x', 'X' produces an error:

```
X
```

```
Error: object 'X' not found
```

Tip: A handy short-cut for the assignment operator in RStudio is “Alt” + “-”

Chapman, C. and Feit, E.M. (2015) R for Marketing Research and Analytics. Springer. Chapter 2.

Vectors

We've just seen how to create a vector: the c() function concatenates individual items into a vector.

```
xNum <- c(1, 3.14159, 5, 7)
xNum
```

```
[1] 1.00000 3.14159 5.00000 7.00000
```

```
xLog <- c(TRUE, FALSE, TRUE, TRUE)
xLog
```

```
[1] TRUE FALSE TRUE TRUE
```

```
xChar <- c("foo", "bar", "boo", "far")
xChar
```

```
[1] "foo" "bar" "boo" "far"
```

Vectors: Type Coercion

A vector can only hold a single type of value (number, text, etc). Values are coerced to the most general type.

```
xMix <- c(1, TRUE, 3, "Hello, world!")
xMix
```

```
[1] "1" "TRUE" "3" "Hello, world!"
```

More about Vectors

c() can be used to add vectors just as it adds single items:

```
x2 <- c(x, x)
x2
```

```
[1] 2 4 6 8 2 4 6 8
```

Type coercion will be applied as needed:

```
c(x2, 100)
```

```
[1] 2 4 6 8 2 4 6 8 100
```

```
c(x2, "Hello")
```

```
[1] "2"  "4"  "6"   "8"  "2"  "4"  "6"  "8"  "Hello"
```

Indexing and Forcing Coercion

```
xMix
```

```
[1] "1"    "TRUE"   "3"       "Hello, world!"
```

```
xMix[1]                 # we'll see more on indices later
```

```
[1] "1"
```

```
as.numeric(xMix[1])           # forces it to "numeric"
```

```
[1] 1
```

```
as.numeric(xMix[1]) + 1.5
```

```
[1] 2.5
```

Help!

There are many ways to get help for R:

Command/Source	**Note**
R: ?*someword*	to get help on *someword* that R knows
R: ??*someword*	to search all R help files for the word in text
R: ? or ??“some string”	search for a string, character, etc. that doesn't work as a word
R: vignette()	list all the vignettes available
R: vignette(“zoo”)	open the vignette named (for package) “zoo”
Web: CRAN task view	Suggested packages by area such as Econometrics, Clustering, etc. https://cran.r-project.org/web/views/
Web: R help list	Monitored by volunteers with many R experts and authors
Web: Google	Understands “R” in many contexts
Web: Stack Overflow	Often great contributions, http://stackoverflow.com/questions/tagged/r

Thank You!

Iman.Ahmadi@wbs.ac.uk
Room No.: 3.207