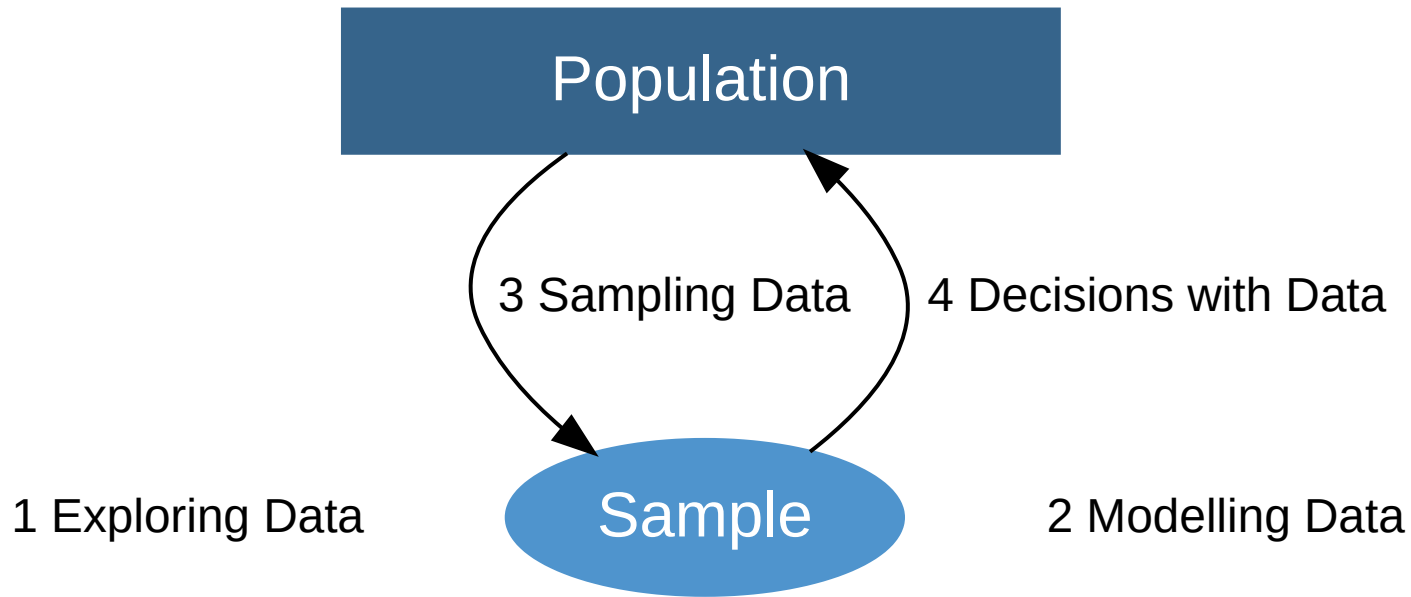


Data Wrangling

Exploring Data | Numerical Summaries

© University of Sydney DATA1001/1901

Unit Overview





Module1 Exploring Data

Design of Experiments

Where did the data come from & can we make reliable conclusions?

Data & Graphical Summaries

What type of data do we have & how can we visualise it?

Numerical Summaries

What are the main features of the data?



Data Wrangling

Data Stories | Sydney house prices

Data Wrangling

Sourcing Data

Scraping Data

Cleaning & Tidying Data

Reshaping Data

Summary

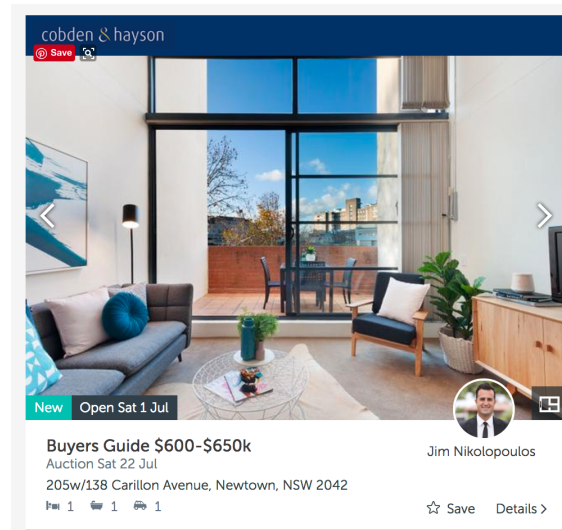
Data Stories

Sydney house prices

How to monitor Sydney House Prices?

Is the Sydney housing market a “bubble” about to burst? Where should I buy a property?

To answer these question, it's possible to scrape data from [domain](#) by manually typing in the data. But data entry can take a long time! Instead, there are ways to **automate** this process.





Statistical Thinking

- Who owns the data on the sale of properties?
- Do you think this data set should be able to be accessed and used by anyone and everyone?
- What are potential ethical issues?

Data Wrangling

Data Wrangling



Data wrangling

Data wrangling is whatever is needed to get the data ready for analysis.

- It is also called **data munging** or **data janitor work**.

The need for data wrangling

- According to [NY Times](#) in 2014, “Data scientists ... spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.”
- “The modern Wild West of data needs to be tamed somewhat so it can be recognized and exploited by a computer program.”
- Constant IT development underpinned by statistical research allow more [automation](#) of this process.

Steps in Data Wrangling

Data wrangling can involve:

- sourcing data
- scraping data
- cleaning and tidying data
- reshaping data
- splitting data
- combining data
- summarising data

 [Boehme: Data Wrangling in R](#)

Sourcing data

Sourcing data

- Often we source data from the web by searching specific terms using engines such as [Google](#), [Bing](#), [Yahoo](#), [Ask](#), [Baidu](#), [Yandex](#).
- For example, use one of the above search engines to search for “sydney house price data” and see what hits you get. What types of data are available?
- When you source data you need to carefully think about the integrity of the data:
 - Is it a reliable source or is it “fake data”? Which year is the data from?
 - Is it a primary data source or has it been manipulated or changed in any way?
 - Sometimes in public media articles, the original source of the data is not clear, even if a given interpretation of the data may get “requoted” many times. This means that the conclusions are unverifiable, and possibly incorrect or misleading.

Scraping data

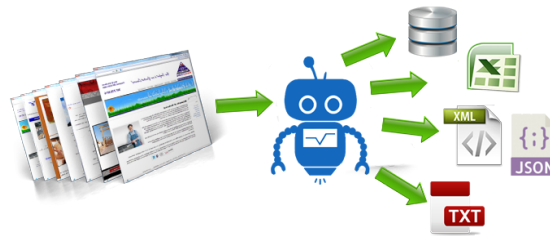
Scraping data



Web scraping

Web scraping is automatically taking data from a web site(s) and transforming it into a structured dataset that can be analysed. More generally, **data scraping** is extracting data from any source.

Lots of companies offer web scraping, like [Web Scraping Crawler](#).



Methods of Web Scraping

There are a number of [methods](#) of web scraping, including:

- **Screen scraping:** We extract data directly from the source code (html) of a website.
- **API:** We interact with an [Application Programming Interface](#) which shares data.

For example: [domain](#)

Data can also be scraped from non-web sources, such as downloadable documents (eg pdf reports).

But is data scraping legal?



Data wrangling

Is web scraping legal?



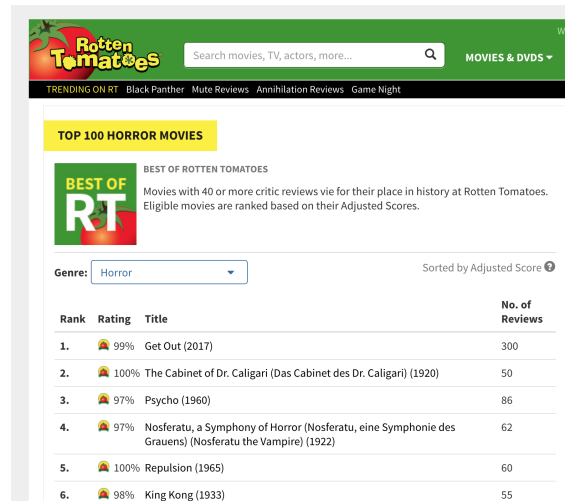
BenBernardBlog

Domain Terms & Conditions

Domain Conditions of Use

Method1: Screen scraping data

- We want to scrape the 100 top horror movies from Rotten Tomatoes https://www.rottentomatoes.com/top/bestofrt/top_100_horror_movies/ and displays..



TOP 100 HORROR MOVIES

BEST OF RT BEST OF ROTTEN TOMATOES
Movies with 40 or more critic reviews vie for their place in history at Rotten Tomatoes. Eligible movies are ranked based on their Adjusted Scores.

Genre: Sorted by Adjusted Score

Rank	Rating	Title	No. of Reviews
1.	99%	Get Out (2017)	300
2.	100%	The Cabinet of Dr. Caligari (Das Cabinet des Dr. Caligari) (1920)	50
3.	97%	Psycho (1960)	86
4.	97%	Nosferatu, a Symphony of Horror (Nosferatu, eine Symphonie des Grauens) (Nosferatu the Vampire) (1922)	62
5.	100%	Repulsion (1965)	60
6.	98%	King Kong (1933)	55

Parsing

1st, directly read the html table into R, which is referred to as **parsing** in web scraping jargon.

```
## Method 1
require(rvest)
require(xml2)
require(pander)
url <- "https://www.rottentomatoes.com/top/bestofrt/top_100_horror_movies"
tableA <- read_html(url)
table1A <- html_table(tableA, fill=TRUE)
```

Convert into “nice” table

2nd, convert the table into a “nice” format that is readable by R, for example using `pander`.

```
table2A <- as.data.frame(table1A[[3]])  
pander(table2A[1:3,], split.cells = 5)
```


Rank	RatingTomatometer	Title	No. of Reviews
1	93%	Us (2019)	527
2	100%	The Cabinet of Dr. Caligari (Das Cabinet des Dr. Caligari) (1920)	54
3	98%	Get Out (2017)	382

Method2: Scraping from a PDF

- We can download reports, data files or forms directly from the internet.
- Some of these documents may be easily read by R such as Excel, text, or comma separated files.
- Others may be more challenging, such as PDF files.
- The following example downloads a PDF of Saturday property auctions in Sydney, then calculates the average 3 bedroom house price per suburb and plots this on a bar chart.

Sydney Auction Results Domain

Saturday 24th February 2018

	Property Snapshot	<small>These auction results are compiled by Home Price Guide ® (Phone: 1800 817 816) based on results collected on or before Saturday 24th February 2018. © Australian Property Monitors Ltd</small>
	Number Listed Auctions:	964
	Number Reported Auctions:	548
	Sold:	455
	Withdrawn:	103
	% Cleared:	70 %
	Total Sales:	\$438,911,000
	Median:	\$1,275,000

For APM Auction Clearance Rate methodology please [click here](#)
For a more comprehensive list of results covering all reported sales by postcode for the past 12 or 24 months, visit [www.homepriceguide.com.au](#)
Agencies: Find out how to report results at [www.apmpropertydata.com.au/report/default.aspx](#)

Saturday's Auctions

Suburb	Address	Type	Price	Result	Agent
Abbotsford	18/3 Harbourview Cr	3 br u	\$2,120,000	S	Warwick Williams RE
Alexandria	202/141-143 McEvoy St	1 br u	\$690,000	S	BresicWhitney
Annandale	6 View St	2 br h	\$1,100,000	S	RW Petersham

Cleaning and tidying data

Cleaning and tidying data

- One of the biggest challenges once you have downloaded data is to get it into a format which is useful! The following video shows some of the challenges as well as some really tidy solutions.

 [Data Wrangling with R Studio and R](#)

Reshaping data

Reshaping data

- We may also want to reshape or group the data to help with our analysis or understanding of the data.
- Two packages that are really useful for this are `dplyr` and `tidyr`, which are part of `tidyverse`.
- A great cheatsheet can be found [here](#).

“Neat” datasets

- The data sets we have looked at so far are “pretty” datasets. Why?
- They all generally conform to the three principles of “tidy” data:
 - Each variable forms a column.
 - Each subject/observation forms a row.
 - Each type of observational unit forms a table.

See [tidy data vignette](#) for more information and some great examples!

Summary

Summary

Data wrangling is often needed to get data ready for analysis. This process can be time-consuming and messy, like the data it is cleaning!

Key Words

data wrangling, data munging, data janitor work, sourcing, scraping, cleaning, tidying, reshaping, splitting, combining, summarising, web scraping, data scraping, screen scraping, API, pipes

Further Thinking

 [R Cheatsheet](#)

 [R Tips](#)

Session Info

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] dplyr_0.8.4  pdftools_2.3 pander_0.6.3 rvest_0.3.5  xml2_1.2.2
## [6] knitr_1.28
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.4      compiler_3.6.2  pillar_1.4.3    RColorBrewer_1.1-2
## [5] tools_3.6.2     digest_0.6.25   jsonlite_1.6.1   evaluate_0.14
## [9] tibble_2.1.3    pkgconfig_2.0.3  rlang_0.4.5      cli_2.0.1
## [13] rstudioapi_0.10  curl_4.3         yaml_2.2.1       xfun_0.12
## [17] DiagrammeR_1.0.5 httr_1.4.1       stringr_1.4.0    htmlwidgets_1.5.1
## [21] vctrs_0.2.2     askpass_1.1      tidyselect_1.0.0  glue_1.4.0
## [25] qpdf_1.1         R6_2.4.1         fansi_0.4.1      rmarkdown_2.1
## [29] purrr_0.3.3     selectr_0.4-2    magrittr_1.5      htmltools_0.4.0
## [33] assertthat_0.2.1 utf8_1.1.4       stringi_1.4.6     visNetwork_2.0.9
## [37] crayon_1.3.4
```