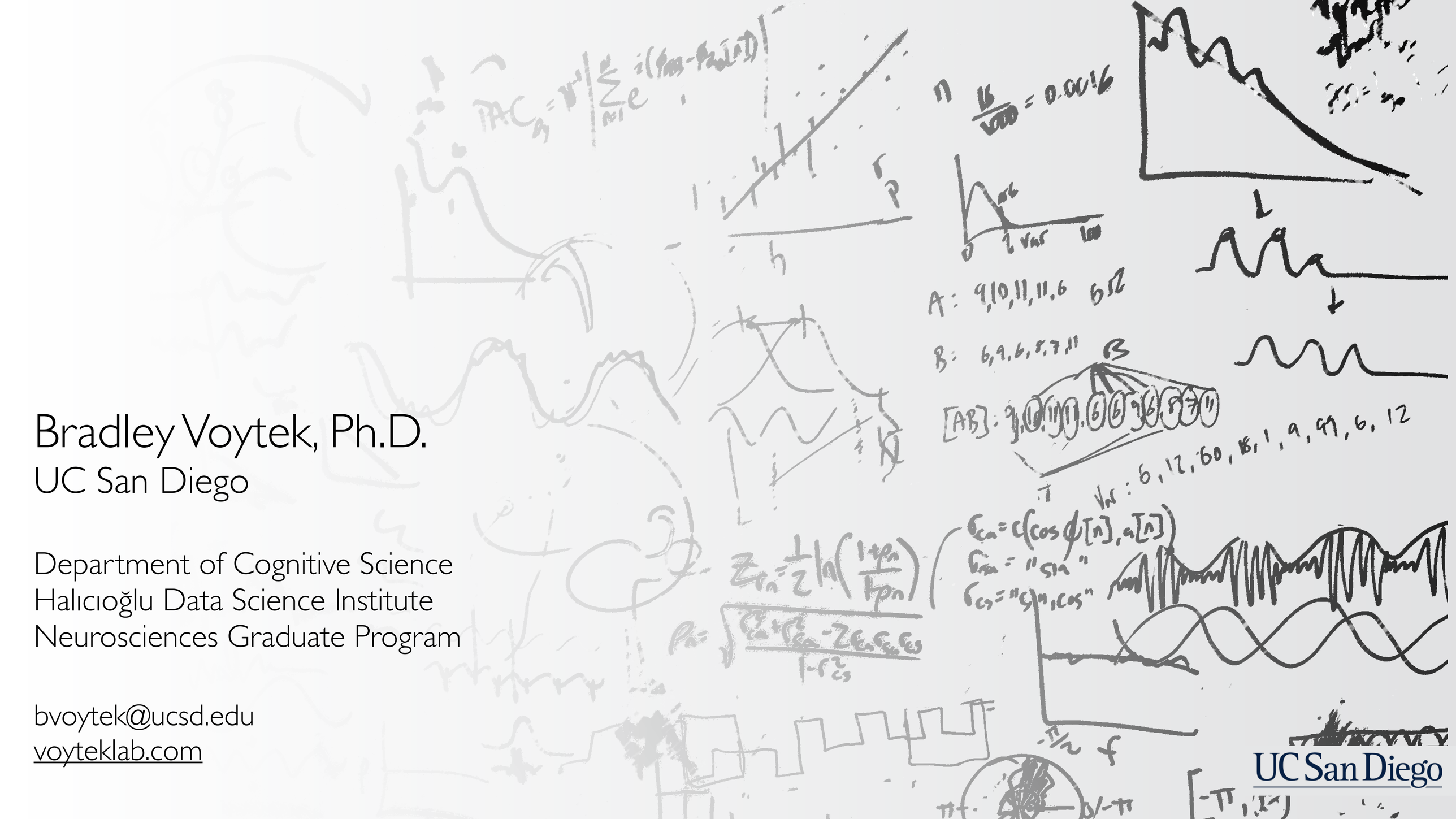
Bradley Voytek, Ph.D.

UC San Diego

Department of Cognitive Science
Halıcıoğlu Data Science Institute
Neurosciences Graduate Program

bvoytek@ucsd.edu
voyteklab.com



UC San Diego

COGS 9
Introduction to Data Science

Data Wrangling

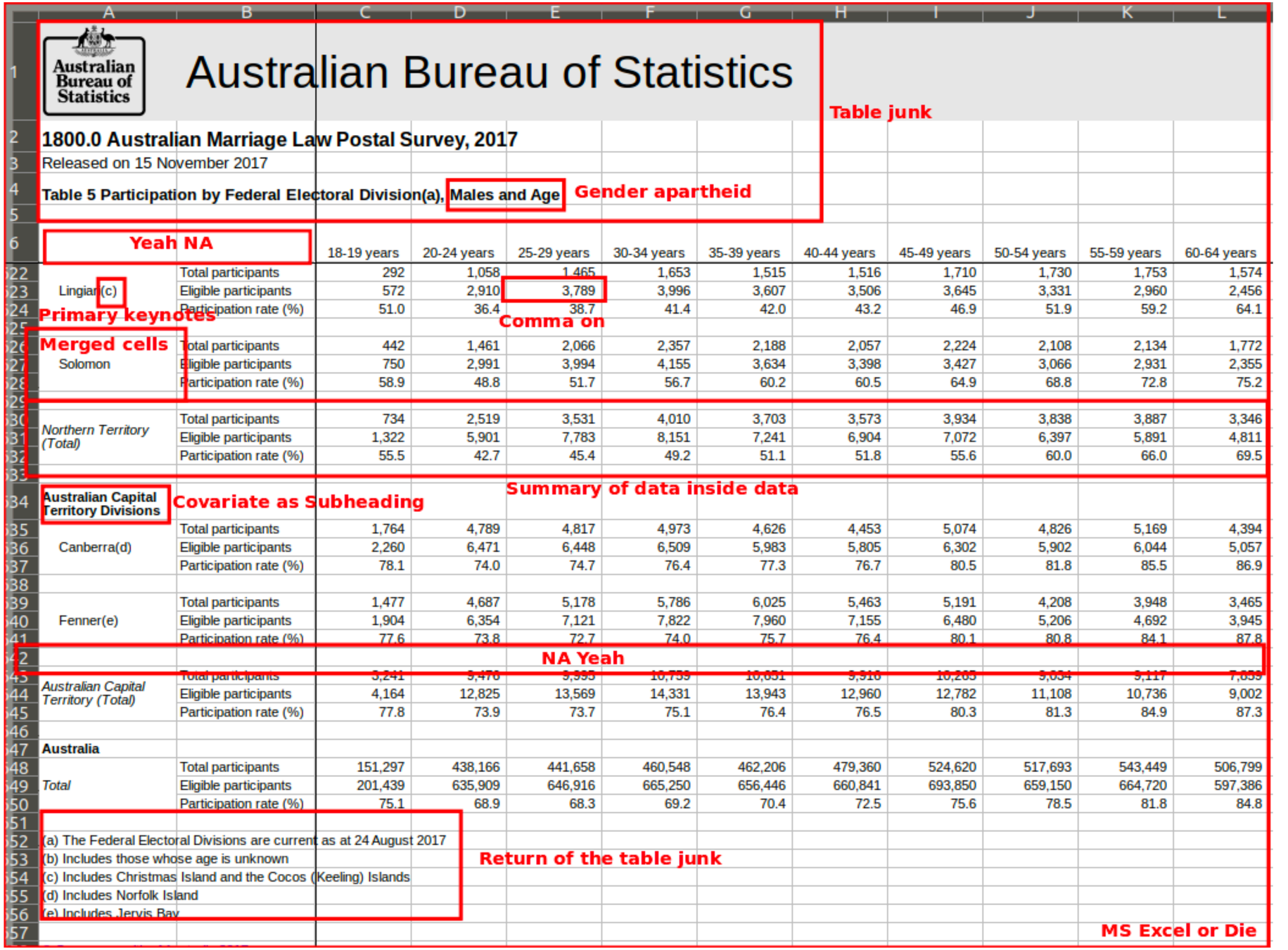
UC San Diego

Today's Learning Objective

- *Describe best practices for storing tabular data*
- *Explain the tidy data principles*
- *Demonstrate how to wrangle a dataset into the tidy data format*

"Good data scientists understand, in a deep way, that the heavy lifting of cleanup and preparation isn't something that gets in the way of solving the problem: *it is the problem*."

Source: DJ Patil, Former head of Data Science (LinkedIn) and US Chief Data Scientist under Obama



Australian Bureau of Statistics

Australian Bureau of Statistics

Table junk

1800.0 Australian Marriage Law Postal Survey, 2017

Released on 15 November 2017

Table 5 Participation by Federal Electoral Division(a), Males and Age Gender apartheid

Yeah NA

		18-19 years	20-24 years	25-29 years	30-34 years	35-39 years	40-44 years	45-49 years	50-54 years	55-59 years	60-64 years
Lingian(c)	Total participants	292	1,058	1,465	1,653	1,515	1,516	1,710	1,730	1,753	1,574
	Eligible participants	572	2,910	3,789	3,996	3,607	3,506	3,645	3,331	2,960	2,456
	Participation rate (%)	51.0	36.4	38.7	41.4	42.0	43.2	46.9	51.9	59.2	64.1
Solomon	Total participants	442	1,461	2,066	2,357	2,188	2,057	2,224	2,108	2,134	1,772
	Eligible participants	750	2,991	3,994	4,155	3,634	3,398	3,427	3,066	2,931	2,355
	Participation rate (%)	58.9	48.8	51.7	56.7	60.2	60.5	64.9	68.8	72.8	75.2
Northern Territory (Total)	Total participants	734	2,519	3,531	4,010	3,703	3,573	3,934	3,838	3,887	3,346
	Eligible participants	1,322	5,901	7,783	8,151	7,241	6,904	7,072	6,397	5,891	4,811
	Participation rate (%)	55.5	42.7	45.4	49.2	51.1	51.8	55.6	60.0	66.0	69.5
Australian Capital Territory Divisions											
Canberra(d)	Total participants	1,764	4,789	4,817	4,973	4,626	4,453	5,074	4,826	5,169	4,394
	Eligible participants	2,260	6,471	6,448	6,509	5,983	5,805	6,302	5,902	6,044	5,057
	Participation rate (%)	78.1	74.0	74.7	76.4	77.3	76.7	80.5	81.8	85.5	86.9
Fenner(e)	Total participants	1,477	4,687	5,178	5,786	6,025	5,463	5,191	4,208	3,948	3,465
	Eligible participants	1,904	6,354	7,121	7,822	7,960	7,155	6,480	5,206	4,692	3,945
	Participation rate (%)	77.6	73.8	72.7	74.0	75.7	76.4	80.1	80.8	84.1	87.8
Australian Capital Territory (Total)	Total participants	3,241	9,476	9,993	10,759	10,651	9,910	10,265	9,034	9,117	7,859
	Eligible participants	4,164	12,825	13,569	14,331	13,943	12,960	12,782	11,108	10,736	9,002
	Participation rate (%)	77.8	73.9	73.7	75.1	76.4	76.5	80.3	81.3	84.9	87.3
Australia											
Total	Total participants	151,297	438,166	441,658	460,548	462,206	479,360	524,620	517,693	543,449	506,799
	Eligible participants	201,439	635,909	646,916	665,250	656,446	660,841	693,850	659,150	664,720	597,386
	Participation rate (%)	75.1	68.9	68.3	69.2	70.4	72.5	75.6	78.5	81.8	84.8

Primary keynotes

Merged cells

Comma on

Summary of data inside data

Covariate as Subheading

NA Yeah

(a) The Federal Electoral Divisions are current as at 24 August 2017
(b) Includes those whose age is unknown
(c) Includes Christmas Island and the Cocos (Keeling) Islands
(d) Includes Norfolk Island
(e) Includes Jervis Bay

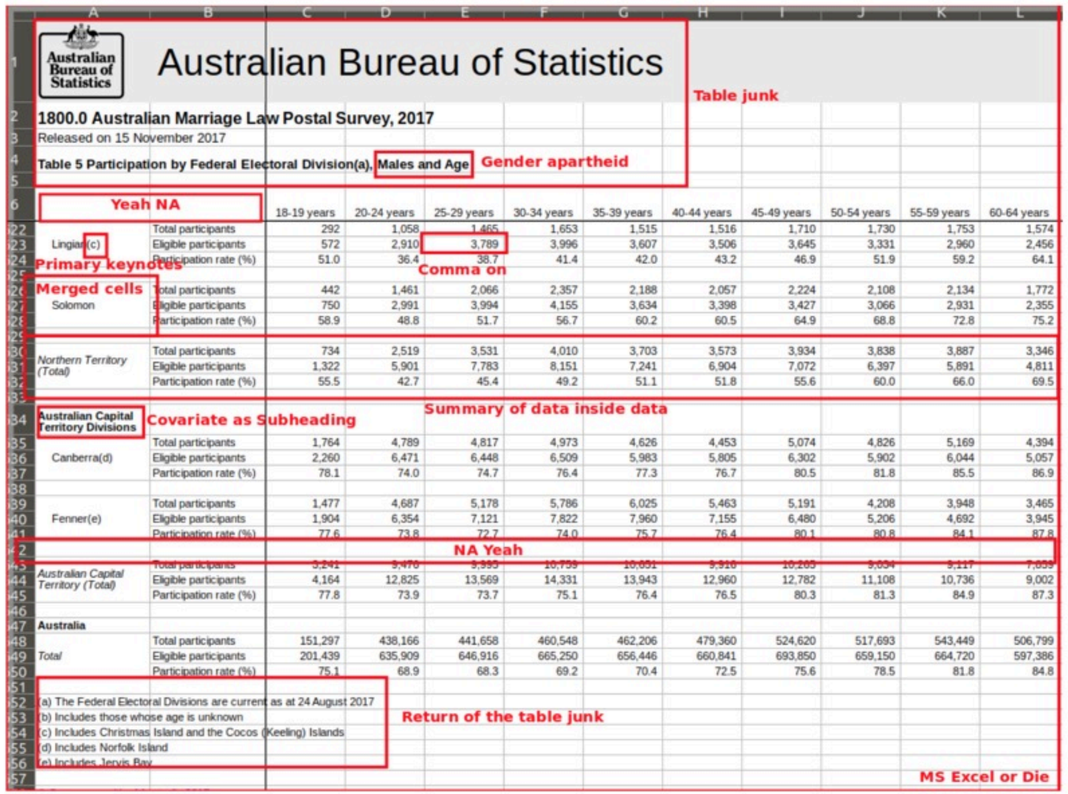
Return of the table junk

MS Excel or Die

Source: https://medium.com/@miles.mcbain/tidying-the-australian-same-sex-marriage-postal-survey-data-with-r-5d35cea07962

Tidy data

untidy data



Australian Bureau of Statistics

1800.0 Australian Marriage Law Postal Survey, 2017

Released on 15 November 2017

Table 5 Participation by Federal Electoral Division(a), Males and Age

Table junk

Gender apartheid

Yeah NA

Primary keynotes

Comma on

Merged cells

Covariate as Subheading

Summary of data inside data

NA Yeah

Return of the table junk

MS Excel or Die

		18-19 years	20-24 years	25-29 years	30-34 years	35-39 years	40-44 years	45-49 years	50-54 years	55-59 years	60-64 years
Lingiari(c)	Total participants	292	1,058	1,465	1,653	1,515	1,516	1,710	1,730	1,753	1,574
	Eligible participants	572	2,910	3,789	3,996	3,607	3,506	3,645	3,331	2,960	2,456
	Participation rate (%)	51.0	36.4	38.7	41.4	42.0	43.2	46.9	51.9	59.2	64.1
Solomon	Total participants	442	1,461	2,066	2,357	2,188	2,057	2,224	2,108	2,134	1,772
	Eligible participants	750	2,991	3,994	4,155	3,634	3,398	3,427	3,066	2,931	2,355
	Participation rate (%)	58.9	48.8	51.7	56.7	60.2	60.5	64.9	68.8	72.8	75.2
Northern Territory (Total)	Total participants	734	2,519	3,531	4,010	3,703	3,573	3,934	3,838	3,887	3,346
	Eligible participants	1,322	5,901	7,783	8,151	7,241	6,904	7,072	6,397	5,891	4,811
	Participation rate (%)	55.5	42.7	45.4	49.2	51.1	51.8	55.6	60.0	66.0	69.5
Australian Capital Territory Divisions											
Canberra(d)	Total participants	1,764	4,789	4,817	4,973	4,626	4,453	5,074	4,826	5,169	4,394
	Eligible participants	2,260	6,471	6,448	6,509	5,983	5,805	6,302	5,902	6,044	5,057
	Participation rate (%)	78.1	74.0	74.7	76.4	77.3	76.7	80.5	81.8	85.5	86.9
Fenner(e)	Total participants	1,477	4,687	5,178	5,786	6,025	5,463	5,191	4,208	3,948	3,465
	Eligible participants	1,904	6,354	7,121	7,822	7,960	7,155	6,480	5,206	4,692	3,945
	Participation rate (%)	77.6	73.8	72.7	74.0	75.7	76.4	80.1	80.8	84.1	87.8
Australian Capital Territory (Total)	Total participants	5,241	9,470	9,995	10,759	10,051	9,916	10,265	9,054	9,117	7,659
	Eligible participants	4,164	12,825	13,569	14,331	13,943	12,960	12,782	11,108	10,736	9,002
	Participation rate (%)	77.8	73.9	73.7	75.1	76.4	76.5	80.3	81.3	84.9	87.3
Australia											
Total	Total participants	151,297	438,166	441,658	460,548	462,206	479,360	524,620	517,693	543,449	506,799
	Eligible participants	201,439	635,909	646,916	665,250	656,446	660,841	693,850	659,150	664,720	597,386
	Participation rate (%)	75.1	68.9	68.3	69.2	70.4	72.5	75.6	78.5	81.8	84.8

(a) The Federal Electoral Divisions are current as at 24 August 2017
(b) Includes those whose age is unknown
(c) Includes Christmas Island and the Cocos (Keeling) Islands
(d) Includes Norfolk Island
(e) Includes Jervis Bay

data wrangling

tidy data

	area	gender	age	State	Area (sq km)	Eligible participants	Participation rate (%)	Total participants	Total Paticipants
2	Adelaide	Female	18-19 years	SA	76	1341	83.5	1120	1120
3	Adelaide	Female	20-24 years	SA	76	4620	81.2	3750	3750
4	Adelaide	Female	25-29 years	SA	76	4897	81.8	4004	4004
5	Adelaide	Female	30-34 years	SA	76	4784	79.8	3820	3820
6	Adelaide	Female	35-39 years	SA	76	4319	79	3411	3411
7	Adelaide	Female	40-44 years	SA	76	4310	80.6	3472	3472
8	Adelaide	Female	45-49 years	SA	76	4579	81.4	3728	3728
9	Adelaide	Female	50-54 years	SA	76	4475	84.7	3791	3791
10	Adelaide	Female	55-59 years	SA	76	4622	87.3	4033	4033
11	Adelaide	Female	60-64 years	SA	76	4342	89.3	3879	3879
12	Adelaide	Female	65-69 years	SA	76	3970	90.7	3602	3602
13	Adelaide	Female	70-74 years	SA	76	3009	90.3	2716	2716
14	Adelaide	Female	75-79 years	SA	76	2156	88.5	1908	1908
15	Adelaide	Female	80-84 years	SA	76	1673	85.1	1423	1423

Source: https://medium.com/@miles.mcbain/tidying-the-australian-same-sex-marriage-postal-survey-data-with-r-5d35cea07962

UC San Diego

DS

Data Science

is the scientific process of extracting value from data

Some typical job duties:

- Collecting, "cleaning" and organizing data sets
- Building data models
- Asking and answering questions with large scale data analysis
- Creating data visualizations and presenting findings to stakeholders

Data scientists have to be *entirely* comfortable working with tabular data.

Source: https://www.rasmussen.edu/degrees/technology/blog/computer-science-vs-data-science/

Data vocabulary

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Data vocabulary

There are 7 different variables (features) in this spreadsheet. Variables are stored in columns.

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Data vocabulary

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

For each variable, we see there are 4 different observations. Observations are stored in rows.

Data vocabulary

Demographic Survey Data

	A	B	C	D	E	F	G
1	ID	Last Name	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Two different types of data

Doctor's Office Measurements Data

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	Smith	Jane	65	180	0.60	163
3	4587	Nayef	Mohammed	75	215	1.46	150
4	1727	Doe	Janice	62	124	0.72	177
5	6879	Jordan	Alex	77	160	1.23	205

Tidy data

1. Each variable you measure should be in a single column

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Tidy data

2. Every observation of a variable should be in a different row

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Tidy data

3. There should be one table for each type of data

Demographic Survey Data

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Doctor's Office Measurements Data

	A	D	E	F	G
1	ID	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	65	180	0.60	163
3	4587	75	215	1.46	150
4	1727	62	124	0.72	177
5	6879	77	160	1.23	205

Tidy data

4. If you have multiple tables, they should include a column in each *with the same column label* that allows them to be joined or merged

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

	A	D	E	F	G
1	ID	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	65	180	0.60	163
3	4587	75	215	1.46	150
4	1727	62	124	0.72	177
5	6879	77	160	1.23	205

Tidy data

Tidy data == rectangular data

A

	A	B	C	D	E
1	id	sex	glucose	insulin	triglyc
2	101	Male	134.1	0.60	273.4
3	102	Female	120.0	1.18	243.6
4	103	Male	124.8	1.23	297.6
5	104	Male	83.1	1.16	142.4
6	105	Male	105.2	0.73	215.7

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

Tidy data benefits

1. consistent data structure
2. foster tool development
3. require only a small set of tools to be learned
4. allow for datasets to be combined

Good spreadsheets

Rules for Tidy Spreadsheets

1. Be consistent
2. Choose good names for things
3. Write dates as YYYY-MM-DD
4. No empty cells
5. Put just one thing in a cell
6. Don't use font color or highlighting as data
7. Save the data as plain text files (i.e. CSV)

Good spreadsheets

1. Be Consistent!

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Keep exactly the same variable names across spreadsheets.

In these data, sex is always specified as "female" or "male." Pick a way to code your variables and stick to it.

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

UC San Diego

Good spreadsheets

2. Choose good names for things

	Do this...	Not This!
Avoid Extra Spaces	'male'	'male '
Use underscores not spaces	doctor_visit_1	Doctor Visit 1
Choose meaningful names	doctor_visit	“F1”

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

Good spreadsheets

3. Write dates as YYYY-MM-DD

	Do this...	Not This!
Use 'ISO 8601' standard	2018-02-27	2/27 *or* 2_27_2018 *or* Feb 27

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

Good spreadsheets

4. No empty cells

A

	A	B	C
1	id	date	glucose
2	101	2015-06-14	149.3
3	102		95.3
4	103	2015-06-18	97.5
5	104		117.0
6	105		108.0
7	106	2015-06-20	149.0
8	107		169.4

B

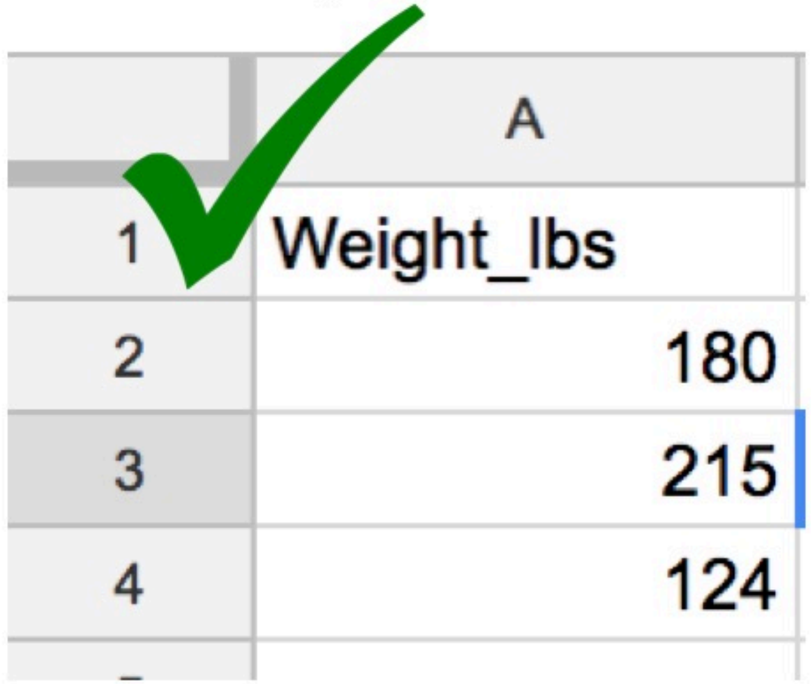
	A	B	C	D	E	F	G	H	I
1		1 min				5 min			
2	strain	normal		mutant		normal		mutant	
3	A	147	139	166	179	334	354	451	474
4	B	246	240	178	172	514	611	412	447

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

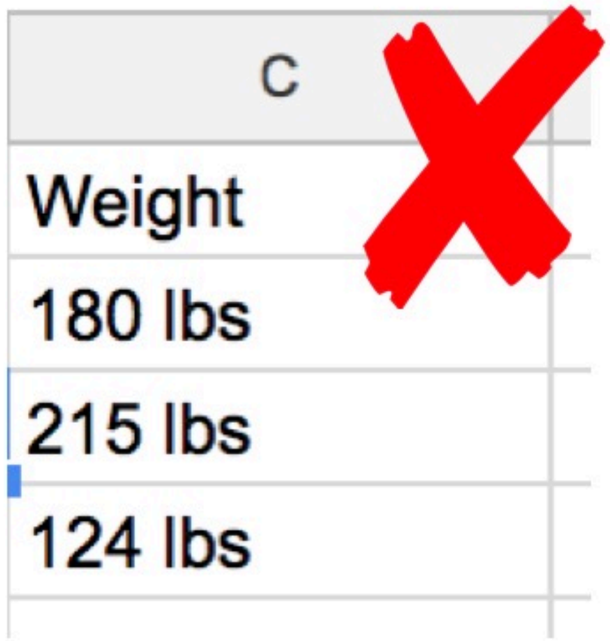
UC San Diego

Good spreadsheets

5. Put just one thing in a cell



	A
1	Weight_lbs
2	180
3	215
4	124



C
Weight
180 lbs
215 lbs
124 lbs

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

UC San Diego

Good spreadsheets

6. Don't use font color or highlighting as data

A

	A	B	C
1	id	date	glucose
2	101	2015-06-14	149.3
3	102	2015-06-14	95.3
4	103	2015-06-18	97.5
5	104	2015-06-18	1.1
6	105	2015-06-18	108.0
7	106	2015-06-20	149.0
8	107	2015-06-20	169.4

B

	A	B	C	D
1	id	date	glucose	outlier
2	101	2015-06-14	149.3	FALSE
3	102	2015-06-14	95.3	FALSE
4	103	2015-06-18	97.5	FALSE
5	104	2015-06-18	1.1	TRUE
6	105	2015-06-18	108.0	FALSE
7	106	2015-06-20	149.0	FALSE
8	107	2015-06-20	169.4	FALSE

Source: Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ*

When..	Be sure to...	So Do this...	Avoid this...	Why?
Naming variables (aka assigning column headers)	Use meaningful variable names	`AgeAtDiagnosis`	`ADx`	`ADx` is an unclear and uninformative abbreviation
Naming variables	Avoid spacing in column headers	`AgeAtDiagnosis`	`Age At Diagnosis`	Spacing in variable names makes the analyst's life more difficult
Naming variables	Use consistent capitalization	`AgeAtDiagnosis`	Using both `AgeAtDiagnosis` and `ageatdiagnosis`	Using consistent column names across tables/spreadsheets simplifies any merging the statistician may have to do.
Naming variables	Avoid using separators, but if it's necessary, use an underscore (`_`)	`IGF1` (or `IGF_1`)	`IGF.1`, `IGF-1`, `IGF/1`, `IGF,1`	Separators (commas, periods, hyphens, slashes, spaces etc.) often have different meanings in coding languages than they do in text. Avoiding them avoids error.
Coding variables	Avoid unnecessary spaces	'male'	'male '	That extra space after 'male ' makes it different from 'male' without a space.
Coding variables	Be consistent!	'male'	'Male',`male', and 'M',	In the eyes of the statistician, 'Male',`male', and 'M' could be incorrectly perceived as three different values.
Coding variables	Be careful of spelling errors	'male'	'maale'	That extra 'a' makes these two different categories.
Coding date and time	Use ISO 8601 coding	'YYYY-MM-DD'	'MM/DD/YY` and `Month Day, Year`	Consistency simplifies the analyst's life, and YYYY-MM-DD will not be misconstrued if opened in Excel.
Coding missing data	Not leave any cells blank and use a consistent value	'NA'	'0', '-9', red-highlighted blank cells, '. ', ' ', ...	Each cell should be filled with a consistent value. Pick a way to denote missingness (ideally 'NA') and stick with it. Avoid using numbers or punctuation to denote missing data.
Entering data	Stick to text and numbers	Convey all information with direct text/numerical entry	Using cell highlighting or font color to convey information	Your analyst may not use the same platform for analysis as you used for data entry, so avoiding font color and cell highlighting will minimize issues.
Generating an Excel file	Save the data in an appropriate format	Use one worksheet per table and save as CSV or text files	Multiple worksheets	Statisticians require this format to import your data onto other platforms.
Entering Data	Avoid entering unnecessary lines of text at the start	Start your first row with variable names	Adding lines of text	This violates the rules of tidy data and makes processing more difficult. Include this information in the "Code book" instead.
Opening files in Excel	Know and avoid its pitfalls	Consistently include one value per cell and be careful of date and time data.	Using macros, splitting cells, and merging cells	These formats are not amenable to data analysis on other platforms.

Source: Ellis SE, Leek JT. (2017) How to share data for collaboration. *PeerJ*

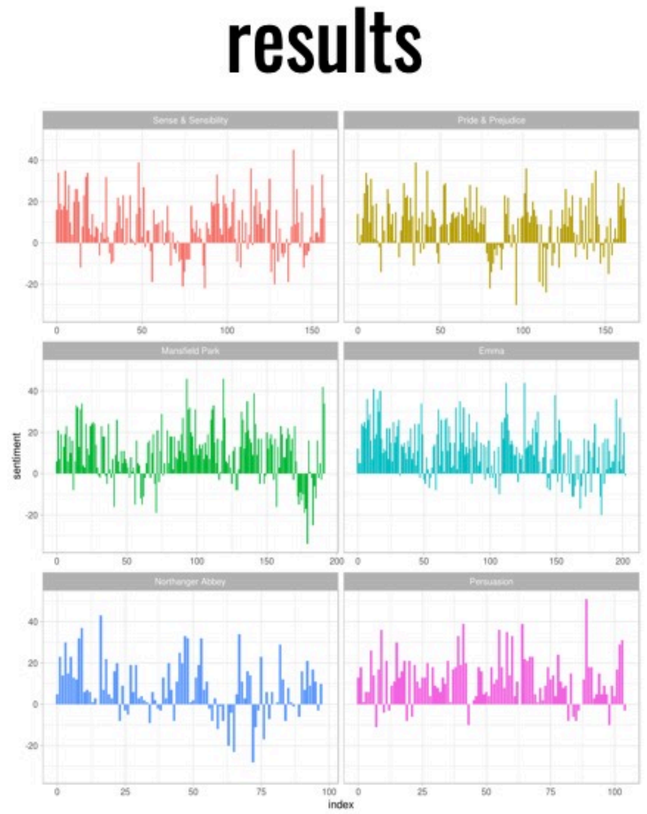
Common problems with messy datasets

1. Column headers are values but should be variable names.
2. A single column has multiple variables.
3. Variables have been entered in both rows and columns.
4. Multiple "types" of data are in the same spreadsheet.
5. A single observation is stored across multiple spreadsheets.

UC San Diego

Source: http://vita.had.co.nz/papers/tidy-data.pdf

Tidying unstructured data



Source: https://www.tidytextmining.com

UC San Diego

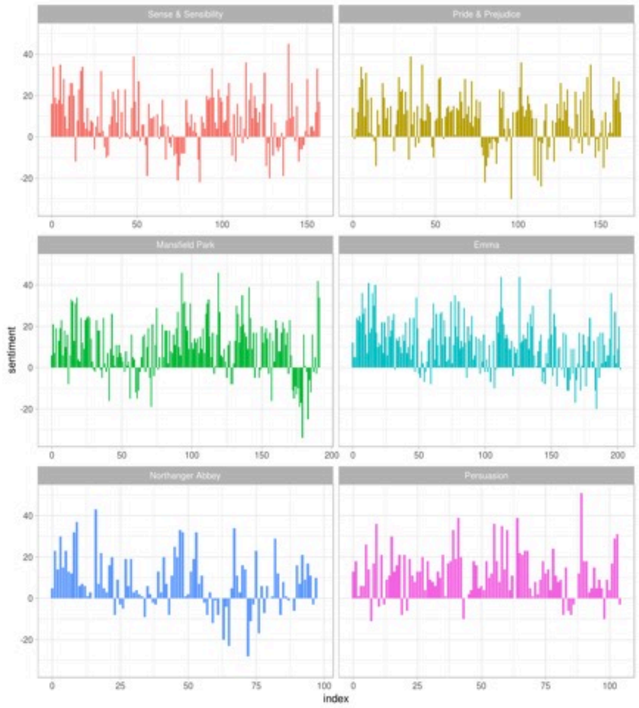
Tidying unstructured data

text

tidy dataset

Word	Novel	Frequency
good	Emma	359
young	Emma	192
friend	Emma	166

results



Source: https://www.tidytextmining.com

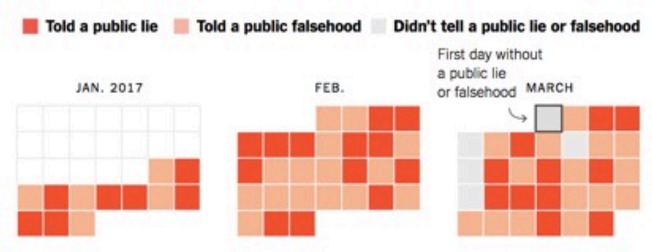
Tidying unstructured data

website

tidy dataset

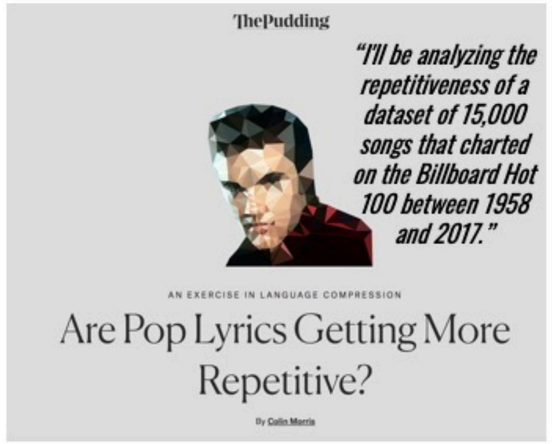
	date	lie	explanation	url
0	Jan. 21, 2017	I wasn't a fan of Iraq. I didn't want to go in...	He was for an invasion before he was against it.	https://www.buzzfeed.com/andrewkaczynski/in-20...
1	Jan. 21, 2017	A reporter for Time magazine — and I have been...	Trump was on the cover 11 times and Nixon appe...	http://nation.time.com/2013/11/06/10-things-yo...
2	Jan. 23, 2017	Between 3 million and 5 million illegal votes ...	There's no evidence of illegal voting.	https://www.nytimes.com/2017/01/23/us/politics...
3	Jan. 25, 2017	Now, the audience was the biggest ever. But th...	Official aerial photos show Obama's 2009 inaug...	https://www.nytimes.com/2017/01/21/us/politics...
4	Jan. 25, 2017	Take a look at the Pew reports (which show vot...	The report never mentioned voter fraud.	https://www.nytimes.com/2017/01/24/us/politics...

results



Tidying unstructured data

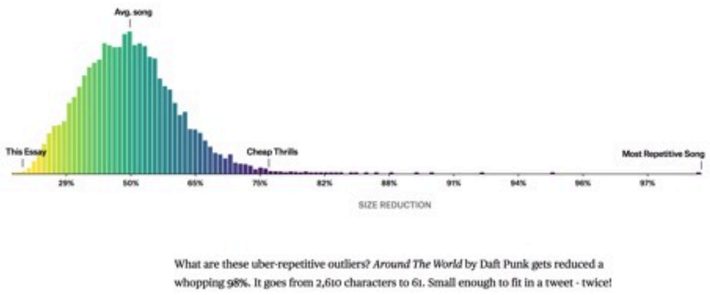
text (lyrics)



tidy dataset

song	Artist	Released	Reduction
Cheap Thrills	Sia	2016	76
Around The World	Daft Punk	1997	98
Everybody Dies	J. Cole	2018	27

results



Source: https://pudding.cool/2017/05/song-repetition/

Tidying unstructured data

The Most Repetitive Songs

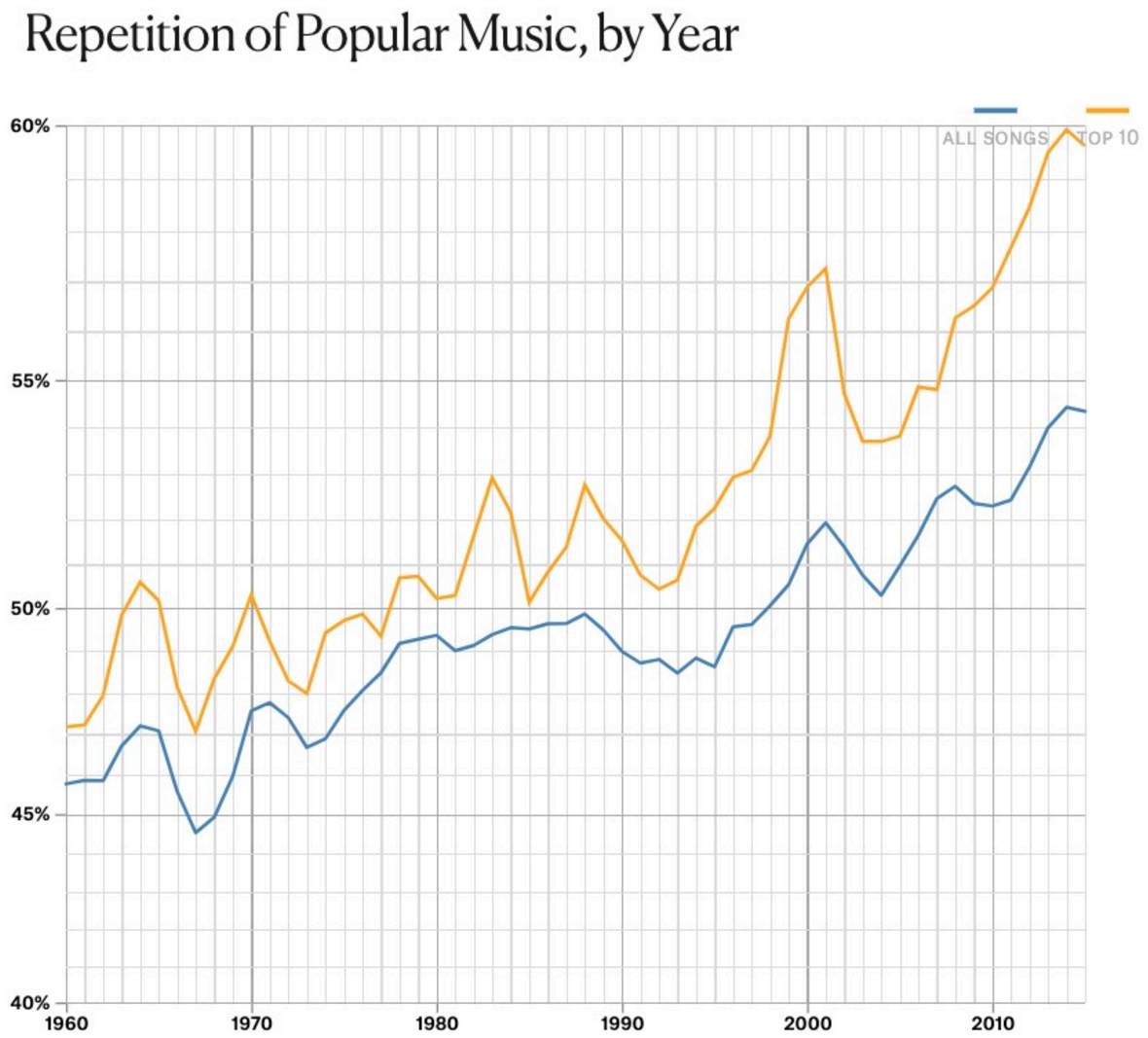
Of 15,000 songs from the Billboard Hot 100

All Decades | '10s | '00s | '90s | '80s | '70s | '60s

	Track	Size Reduction
1.	**Around The World** - Daft Punk, 1997	98%
2.	**The Rockafeller Skank** - Fatboy Slim, 1998	95%
3.	**Chain Of Fools (Part 1)** - Jimmy Smith, 1968	92%
4.	**Get Low** - Dillon Francis & DJ Snake, 2015	90%
5.	**Barbra Streisand** - Duck Sauce, 2011	89%
6.	**Let's All Chant** - The Michael Zager Band, 1978	88%
7.	**Keep It Comin' Love** - KC And The Sunshine Band, 1977	87%
8.	**Turn Down For What** - DJ Snake & Lil Jon, 2014	87%
9.	**Who'd She Coo?** - Ohio Players, 1976	86%
10.	**Watch Me** - Silento, 2015	86%

Source: https://pudding.cool/2017/05/song-repetition/

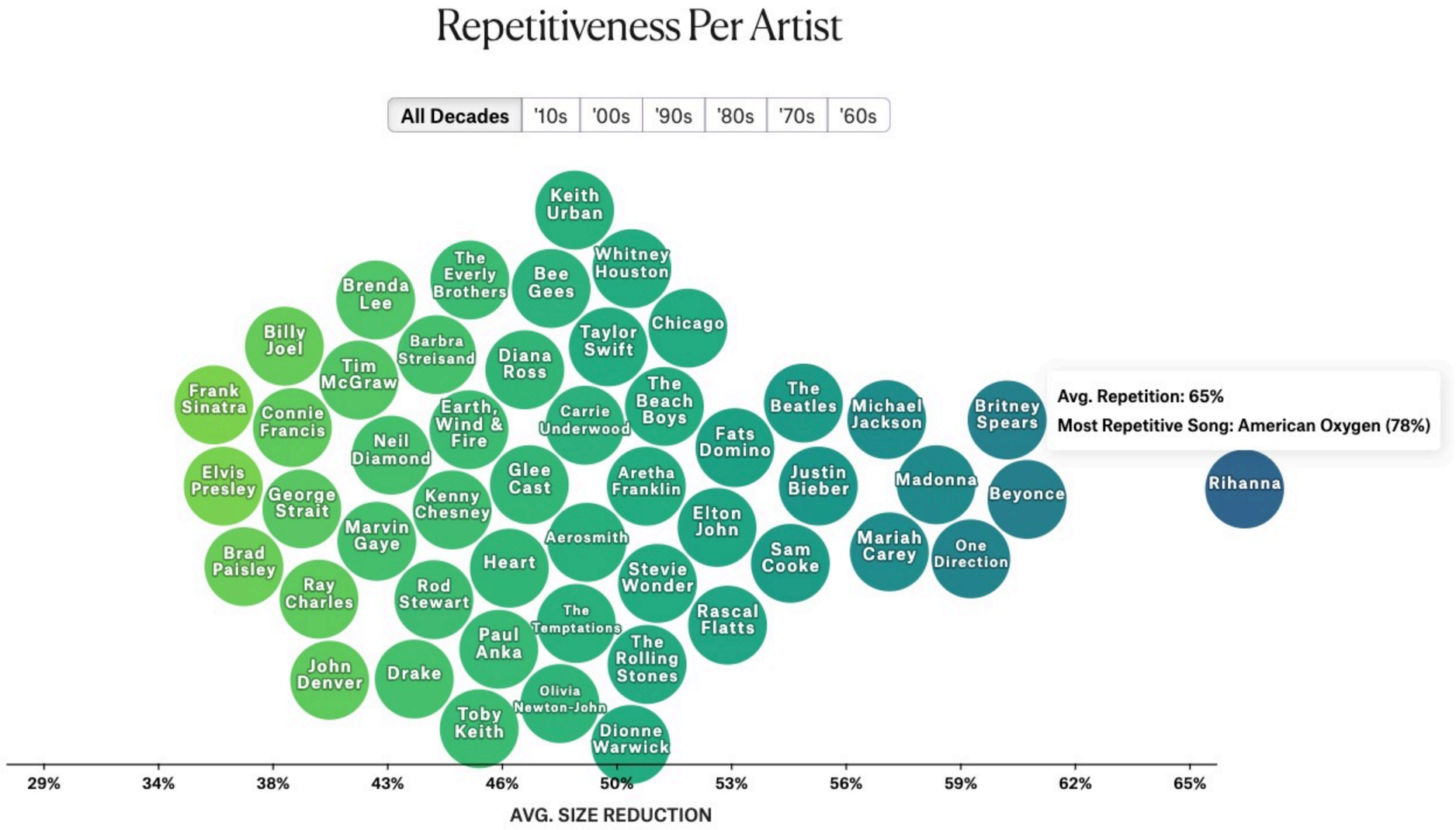
Tidying unstructured data



Source: https://pudding.cool/2017/05/song-repetition/

UC San Diego

Tidying unstructured data



Source: https://pudding.cool/2017/05/song-repetition/

UC San Diego

What about *actually* working with that tabular data?

Common data wrangling tasks (and verbs)

- subset dataset
  - filter : filter rows
  - select : select columns
- change order
  - arrange : reorder rows
  - (reorder) : reorder columns
- add a column : mutate
- summarize data
  - group_by : group by other variables
  - summarize : reduce multiple values down to a single value

filter rows

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

id	last	first	sex	height
1004	Smith	Jane	F	65
1727	Doe	Janice	F	60

```
filter(sex == 'F')
```

arrange rows

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

```
arrange(height)
```

arrange rows

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

id	last	first	sex	height
6879	Jordan	Alex	M	55
1727	Doe	Janice	F	60
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72

```
arrange(height)
```

select & reorder columns

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

```
select(id, first, sex)
```

select & reorder columns

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

id	first	sex
1004	Jane	F
4587	Mohammed	M
1727	Janice	F
6879	Alex	M

```
select(id, first, sex)
```

select & reorder columns

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

id	first	sex
1004	Jane	F
4587	Mohammed	M
1727	Janice	F
6879	Alex	M

```
select(id, first, sex)
```

You can change up the order of the variables to reorder columns

mutate to create a new column

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

```
mutate(patient_dr = 'Grey' if id < 2000 else 'Shepherd')
```

mutate to create a new column

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

id	last	first	sex	height	patient_dr
1004	Smith	Jane	F	65	Grey
4587	Nayef	Mohammed	M	72	Shepherd
1727	Doe	Janice	F	60	Grey
6879	Jordan	Alex	M	55	Shepherd

```
mutate(patient_dr = 'Grey' if id < 2000 else 'Shepherd')
```

group by & summarize

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

```
summarize('height', mean)
```

group by & summarize

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

height
63

```
summarize('height', mean)
```

group by & summarize

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

```
group by(sex)
summarize('height', mean)
```

group by & summarize

id	last	first	sex	height
1004	Smith	Jane	F	65
4587	Nayef	Mohammed	M	72
1727	Doe	Janice	F	60
6879	Jordan	Alex	M	55

sex	height
F	62.5
M	63.5

```
group by(sex)
summarize('height', mean)
```

pandas!

Pandas

Preprocessing our data

Much of what data scientists do involves cleaning and preprocessing data:

- Handling missing or invalid values
- Extracting usable information from messy strings
- Transforming/normalizing variables and variable names
- Filtering redundant or bad data
- Merging with other datasets
- Etc...

Source: Tal Yarkoni

Pandas

Pandas data structures

- Provides functionality similar to data frames in R
- Two main data structures: Series and DataFrames
- A Series is a 1-dimensional numpy array with axis labels

Source: Tal Yarkoni

Pandas

```
# Initialize a Series from a numpy array and index labels
a = np.arange(3, 8)
b = pd.Series(a, index=['apple', 'banana', 'orange', 'pear', 'grapes'])

# Let's take a look...
print(b)
```

```
apple      3
banana     4
orange     5
pear       6
grapes     7
dtype: int64
```

Source: Tal Yarkoni

Pandas

```
# Unlike numpy arrays, we can now refer to elements by label.
# The syntax is similar to dictionary indexing. You can also
# treat labels like attributes (e.g., b.pear), but this runs
# the risk of collisions and should be avoided.
print(b['pear'])

# We can always retrieve the underlying numpy array with .values
print(b.values)

# Many numpy operations work as expected, including slicing
print(b[2:4])

# Each column in our loaded dataset is a Series
print(data['Breed'][:5])
```

```
6
[3 4 5 6 7]
orange    5
pear      6
dtype: int64
0    Labrador Retriever Mix
1    Domestic Shorthair Mix
2    Domestic Shorthair Mix
3    Domestic Shorthair Mix
4                Bulldog Mix
Name: Breed, dtype: object
```

Source: Tal Yarkoni

UC San Diego

Pandas

The pandas DataFrame

- The workhorse of data analysis in pandas
- A container of multiple aligned Series
- Heterogeneous: a DF's Series can have different dtypes

Source: Tal Yarkoni

Pandas

Indexing pandas DataFrames

- pandas DFs spport flexible indexing by labels and/or indices
  - A common gotcha: R-style indexing won't work
  - Be explicit about whether you're using integer or label indexing

Source: Tal Yarkoni

Pandas

Python: 0-indexing
R: 1-indexing

Indexing pandas DataFrames

- pandas DFs spport flexible indexing by labels and/or indices
  - A common gotcha: R-style indexing won't work
  - Be explicit about whether you're using integer or label indexing

Source: Tal Yarkoni

Pandas

```
# This won't work!
data[0, 'Animal Type']

# # but .ix supports mixed integer and label based access
data.ix[0, 'Animal Type']

# # Returns the entire column
data['Animal Type']

# # Position-based selection; returns all of rows 2 - 5
data.iloc[2:5]

# # Returns rows 2 - 5, columns 2 and 7
data.iloc[2:5, [2, 7]]

# # Label-based indexing; equivalent to data['Animal Type']
# # in this case
data.loc[:, 'Animal Type']
```

Source: Tal Yarkoni

Pandas

Slide Type Fragment

```
data.describe()
```

	Animal ID	Name	DateTime	MonthYear	Outcome Type	Outcome Subtype	Animal Type	Sex upon Outcome	Age upon Outcome	Breed	Color
count	43870	30614	43870	43870	43861	21197	43870	43869	43836	43870	43870
unique	40612	9939	36235	36235	8	18	5	5	45	1792	433
top	A694501	Bella	08/11/2015 12:00:00 AM	08/11/2015 12:00:00 AM	Adoption	Partner	Dog	Neutered Male	1 year	Domestic Shorthair Mix	Black/White
freq	8	207	25	25	17342	11652	24964	15645	7478	13039	4602

Source: Tal Yarkoni

Pandas

Importing data

- Before we do anything else, we need to get our data into a usable form
- Most commonly, data will come from a flat file
- But sometimes we need to retrieve data from other sources
- We'll do both

Source: Tal Yarkoni

Not Pandas

> **Reading data in with the standard library**
>
> There are many ways to read in data in Python using the standard library. Here's a simple example, where we read in the data line-by-line and split each line into its own list.

UC San Diego

Source: Tal Yarkoni

Not Pandas

```
filename = '../data/Austin_Animal_Center_Outcomes.csv'
data = []  # Initialize an empty list to store the data

# Loop over rows in the file, split each one into a list
# of values, and add the result to the data list.
for line in open(filename).readlines():
    line = line.strip().split(',')
    data.append(line)

print("Found {} rows.".format(len(data)))

# Print the 1000th row to see what it looks like
data[1000]
```

```
Found 43871 rows.
```

```
['A664984',
 'Buddy',
 '10/18/2013 06:46:00 PM',
 '10/18/2013 06:46:00 PM',
 'Adoption',
 '',
 'Dog',
 'Neutered Male',
 '1 year',
 'Pit Bull Mix',
 'Blue']
```

Source: Tal Yarkoni

Pandas

Slide Type Skip

The problem with approaches like the one above is that the data lack a tabular format, making it very hard to operate over rows or columns. We're much better off using the *pandas* package to hold our data in a pandas DataFrame (DF)--a data structure that wraps around numpy arrays and is expressly designed to support a range of powerful operations over data. Reading a dataset into a pandas DF is very easy with the workhorse read_csv() or read_table() methods. These methods take a large number of optional arguments that make it easy to read in almost any kind of orderly data represented in a text file.

Source: Tal Yarkoni

Pandas

Slide Type: Sub-Slide

Reading data, the pandas way

Slide Type: Fragment

```
# Note that we're reading the file directly from GitHub.
# pandas accepts URLs in addition to local files.

# url = 'http://raw.githubusercontent.com/tyarkoni/SSI2016/master/data/Austin_Animal_Center_O
# If you're working from the cloned course GitHub repo, comment the line above and uncomment
# the line below for faster loading.
url = '../data/Austin_Animal_Center_Outcomes.csv'

# The workhorse data-reading method in pandas.
# It accepts a LOT of optional arguments--
# see http://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html
data = pd.read_csv(url)

# calling head() on a DataFrame shows the top N rows.
data.head(5)
```

Source: Tal Yarkoni

Pandas

Other formats

Pandas has built-in support for reading from or to other common formats/sources:

- Generic delimited text -- read_table()
- Excel -- read_excel()
- JSON -- read_json()
- SQL -- read_sql()
- Stata -- read_stata()
- SAS (XPORT or SAS7BDAT) -- read_sas()
- etc...

Source: Tal Yarkoni

Types of data

Structured & semi-structured

Spreadsheets (CSVs, .xlsx)
JSON & XML
relational databases (SQL)

Unstructured

everything else: video, audio, images, websites, apps, text, etc.

CSV

Each column separated by a comma

```
10/29/19 9:33,170,165,15,100000,100
10/29/19 9:34,183,160,10,10000000,90
10/29/19 9:34,182,164,16,123456,727
10/29/19 9:34,162,165,15,10000,5
10/29/19 9:34,174,165,30,200000,15
10/29/19 9:34,168,172,21,1000,200
10/29/19 9:34,159,175,50,500,100
10/29/19 9:34,170.18,173.6,10,34600000,73
10/29/19 9:34,190,160,30,6,60
10/29/19 9:34,170,170,25,30,15
```

Each row is separated by a new line

JSON

JSON: key-value pairs

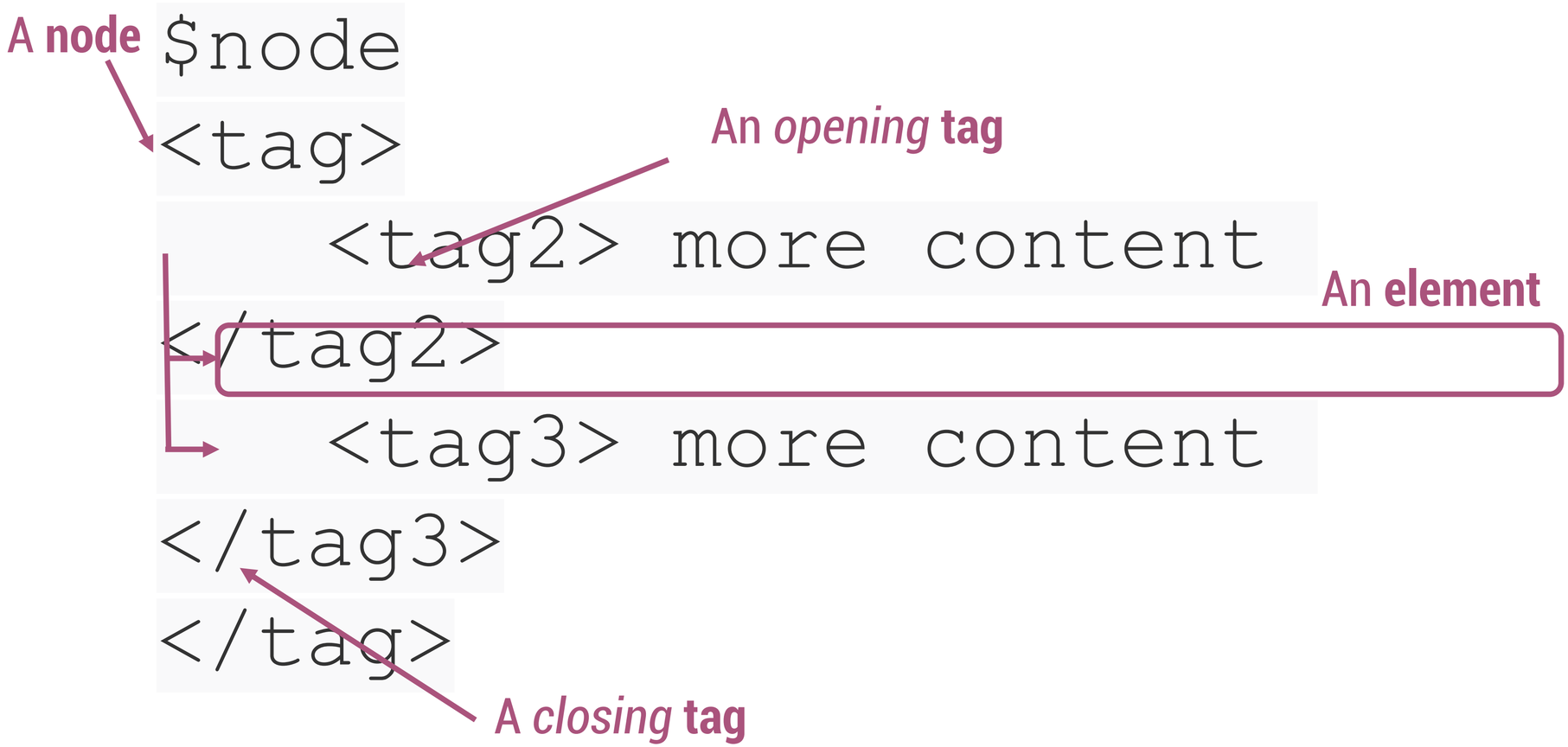
nested/hierarchical data

JSON

These are all nested within `attributes`

These are all nested within `"Good For"`

XML


A node
$node
<tag>
An opening tag
<tag2> more content
An element
</tag2>
<tag3> more content
</tag3>
</tag>
A closing tag

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<customers>
    <customer>
        <customer_id>1</customer_id>
        <first_name>John</first_name>
        <last_name>Doe</last_name>
        <email>john.doe@example.com</email>
    </customer>
    <customer>
        <customer_id>2</customer_id>
        <first_name>Sam</first_name>
        <last_name>Smith</last_name>
        <email>sam.smith@example.com</email>
    </customer>
    <customer>
        <customer_id>3</customer_id>
        <first_name>Jane</first_name>
        <last_name>Doe</last_name>
        <email>jane.doe@example.com</email>
    </customer>
</customers>
```

Data science questions

"If I had four hours to chop down a tree, I'd spend the first two hours sharpening the axe."

Data science questions

- Be specific
- Be answerable with data
- Specify what's being measured

Real data crunching

Test No. 8

33. Accrued depreciation is a term used in the real estate appraisal field. Which of the following methods would accrued depreciation have its greatest effect on?

a. Market data approach
b. Cost approach
c. Capitalization of net income approach
d. Sales comparison approach

Ans.(b):

34. What transfers less than an entire leasehold, with the original lessee being primarily liable for the rental agreement?

a. Sublease
b. Sandwich lease
c. Residential lease
d. Assignment

Ans.(a):

- PDF of 10 tests, 200 questions each

UC San Diego

Real data crunching

Test No. 8

33. Accrued depreciation is a term used in the real estate appraisal field. Which of the following methods would accrued depreciation have its greatest effect on?

a. Market data approach
b. Cost approach
c. Capitalization of net income approach
d. Sales comparison approach

Ans.(b):

34. What transfers less than an entire leasehold, with the original lessee being primarily liable for the rental agreement?

a. Sublease
b. Sandwich lease
c. Residential lease
d. Assignment

Ans.(a):

- PDF of 10 tests, 200 questions each
- How many unique questions?

UC San Diego

Real data crunching

Test No. 8

33. Accrued depreciation is a term used in the real estate appraisal field. Which of the following methods would accrued depreciation have its greatest effect on?

a. Market data approach
b. Cost approach
c. Capitalization of net income approach
d. Sales comparison approach

Ans.(b):

34. What transfers less than an entire leasehold, with the original lessee being primarily liable for the rental agreement?

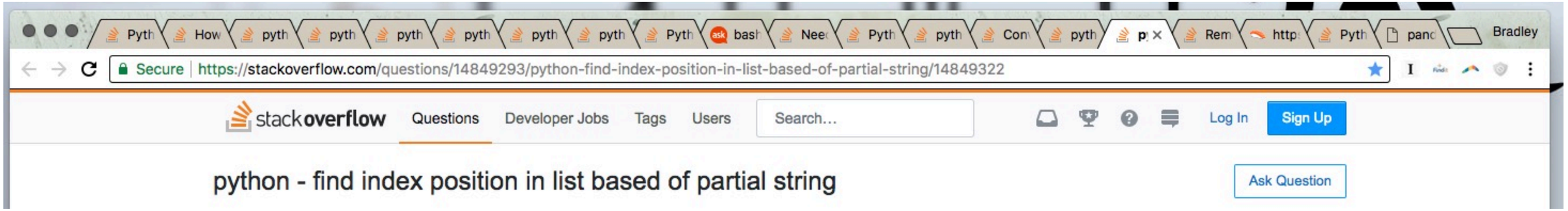
a. Sublease
b. Sandwich lease
c. Residential lease
d. Assignment

Ans.(a):

- PDF of 10 tests, 200 questions each
- How many unique questions?
- Which ones are duplicated, and how often?

UC San Diego

Real data crunching



Real data crunching

```
In [2]: filename = 'exam.pdf'
        pdf = PyPDF2.PdfFileReader(open(filename, 'rb'))
        total_pages = pdf.getNumPages()
        total_pages

Out[2]: 500
```

Real data crunching

```
In [3]: text_data = ''
        for i in range(0, total_pages):
            pageObj = pdf.getPage(i)
            text_data = text_data + pageObj.extractText()
        text_data
```

```
Out[3]: '     Test No. 1, page \n1       1. Any \ncomplaint as to the violat\nion of the United States Civil Rights\n Act of\
        n 1968 should be filed within how many days of its occurrence?\n   a. 180 days\n  b. 365 days\n  c. 30 days\n  d. 90
        days\n  Ans.(b): \nAn aggrieved person may file a\n complaint directly to a U.S. District\n Court within one year of
        the al\nleged discriminatory practice, whether or not a\n verified complaint has been filed with the Secretary of HUD
        .\n    2. A fee \nsimple e\nstate\n is mo\nst likely to be a:\n   a. life\n estate.\n  b. leasehold.\n  c. less\n-tha
```

- How do I parse this into "questions"?

Real data crunching

```
In [4]: new_data = text_data.replace('\n','')
        new_data = new_data.replace('->',' ')
        new_data

Out[4]: '     Test No. 1, page 1       1. Any complaint as to the violation of the United States Civil Rights Act of 1968 sho
        uld be filed within how many days of its occurrence?   a. 180 days  b. 365 days  c. 30 days  d. 90 days  Ans.(b): An
        aggrieved person may file a complaint directly to a U.S. District Court within one year of the alleged discriminatory
        practice, whether or not a verified complaint has been filed with the Secretary of HUD.    2. A fee simple estate is
        most likely to be a:   a. life estate.  b. leasehold.  c. less-than-freehold estate.  d. maximum interest obtainable.
```

- How do I parse this into "questions"?
- Clean it up first…

Real data crunching

In [5]:
```
new_data = text_data.replace('\n','')
new_data = new_data.replace('->',' ')

questions = 200
tests = 10
total_questions = questions * tests

sentence = []

i = 1
j = 0

while len(sentence) < total_questions:
    my_regex = '\ \ ' + str(i) + '\.\ (.*?)a\.\ '
    result = re.search(my_regex, new_data)
    result.group()

    sentence.append(new_data[(result.span()[0]+len(str(i))+4):(result.span()[1]-len(str(i))-2)])
    sentence[j] = sentence[j].strip()

    new_data = new_data[(result.span()[1]-len(str(i))-4):]

    i += 1
    if i>questions:
        i = 1
    j += 1
```

- Extract sentences

Real data crunching

```
In [8]: len(sentence), sentence

Out[8]: (2000,
         ['Any complaint as to the violation of the United States Civil Rights Act of 1968 should be filed within how many da
        ys of its occurrence?',
          'A fee simple estate is most likely to be a:',
          'Mr. Jones, an owner of a packaging firm, purchased a new machine in 2008 and paid $5,500. It was estimated at the
        time of the purchase to have a total economic life of 10 years and a salvage value of $550. Using the straight line m
        ethod of depreciation, the book value at the end of 7 years would be:',
```

- How do I parse this into "questions"?
- Clean it up first…
- Did it work?

Real data crunching

In [9]:
```
import pandas as pd
import numpy as np

df = pd.DataFrame({'id': np.arange(len(sentence))+1, 'questions': sentence})

df['similarity'] = ''
df['indices'] = ''

similarity = []
indices = []


for i in range(0, len(sentence)):
    similarity.append([])
    indices.append([])

    for j in range(0, len(sentence)):
        l_val = Levenshtein.ratio(sentence[i], sentence[j])

        if i != j:
            if l_val >= 0.75:
                similarity[i].append(l_val)
                indices[i].append(j)

    df.at[i, 'similarity'] = similarity[i]
    df.at[i, 'indices'] = indices[i]
```

- Calculate similarity

Real data crunching

```
In [53]: i = 0
         similarity[i], indices[i], sentence[i], sentence[indices[i][0]], sentence[indices[i][1]], sentence[indices[i][2]]

Out[53]: ([1.0, 1.0, 1.0],
          [847, 1304, 1748],
          'Any complaint as to the violation of the United States Civil Rights Act of 1968 should be filed within how many day
         s of its occurrence?',
          'Any complaint as to the violation of the United States Civil Rights Act of 1968 should be filed within how many day
         s of its occurrence?',
          'Any complaint as to the violation of the United States Civil Rights Act of 1968 should be filed within how many day
         s of its occurrence?',
          'Any complaint as to the violation of the United States Civil Rights Act of 1968 should be filed within how many day
         s of its occurrence?')
```

- How do I parse this into "questions"?
- Clean it up first…
- Did it work?
- What is a "unique" question?

Real data crunching

```
In [15]: i = 3
similarity[i], indices[i], sentence[i], sentence[indices[i][0]], sentence[indices[i][1]], sentence[indices[i][2]]

Out[15]: ([0.8872727272727273, 0.9390681003584229, 1.0],
 [643, 1190, 1893],
 'Fred pays $16,240 for a home. If it costs 10% to sell his home before he could sell it at a profit, how much would
it have to appreciate?',
 'Mr. Robert pays $15,240 for a lot. It costs 12% to sell his lot. Before he could sell it at a profit how much would
it have to appreciate?',
 'Mr. Bill pays $152,400 for a home. If it costs 12% to sell his home before he could sell it at a profit, how much w
ould it have to appreciate?',
 'Fred pays $16,240 for a home. If it costs 10% to sell his home before he could sell it at a profit, how much would
it have to appreciate?')
```

- How do I parse this into "questions"?
- Clean it up first…
- Did it work?
- What is a "unique" question?