

# Working With Text

Sam Lau

Data 100 Spring 2018 Lecture 10

# Who is this guy?

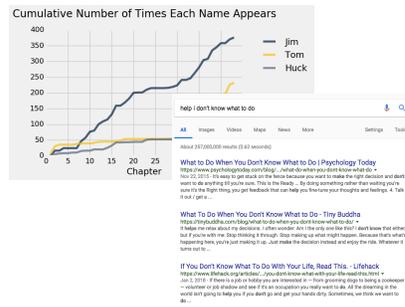
5th Year MS with Josh Hug  
CS and Education

Data 8 for its first 3 offerings

Data 100 for its first 2  
Now helping to write  
textbook for Data 100



# Why teach text?



Donald J. Trump @realDonaldTrump · 11h  
The journey to #MAGA began @CPAC 2011 and the opportunity to reconnect with friends and supporters is something I look forward to every year. See you at #MAGAC2018

Donald J. Trump @realDonaldTrump · 19h  
Thank you to Sue Kruczek, who lost her wonderful and talented son Nick to the Covid scourge, for your kind words written @realDonaldTrump. We are fighting this terrible epidemic hard - Nick will not have died in vain!

Donald J. Trump @realDonaldTrump · 19h  
This will be a big week for infrastructure. After so stupidly spending \$7 trillion in the Middle East, it is now time to start investing in OUR Country.

Donald J. Trump @realDonaldTrump · Feb 11  
Just spoke to @JohnKasich to express condolences and prayers to all for the horrible shooting of two great police officers from @WesternJAILCO. This is a true tragedy!

There's a world of data in text.

Text is hard to work with

Text is easy to work with

"There is no worse way to screw up data than to let a single human type it in, without validation.

I once acquired the complete dog licensing database for Cook County, Illinois...

...this database contained at least 250 spellings of Chihuahua."

– Quartz guide to bad data

<https://github.com/Quartz/bad-data-guide>

20% of the tools work for  
80% of text.

## In this lecture

### Simple

Python string methods

Looping

### Complex

Regular expressions

pandas string functions

## Use Case 1: Cleaning Text

New Tool: Python String Methods

```
County | State
De Witt County | IL
Lac qui Parle County | MN
Lewis and Clark County | MT
St John the Baptist Parish | LA
```

### join

```
County | Population
DeWitt | 16798
Lac Qui Parle | 8067
Lewis & Clark | 55716
St. John the Baptist | 43044
```

(demo)

## Python String Methods

Slicing `str[::-7]`

Replacements `str.replace('&', 'and')`

Deletions `str.replace(' ', '')`

Transformations `str.lower()`

Splitting `str.split('/')`

<https://docs.python.org/3/library/stdtypes.html#string-methods>

## Use Case 2: Extracting Fields

New Tool: Regular Expressions

## Extracting Fields

```
169.237.46.168 - - [26/Jan/2014:10:47:58 -0800] "GET /stat141/Winter04/ HTTP/1.1"
193.205.203.3 - - [2/Feb/2005:17:23:6 -0800] "GET /stat141/Notes/dim.html HTTP/1.0"
169.237.46.240 - "" [3/Feb/2006:10:18:37 -0800] "GET /stat141/homework/ HTTP/1.1"
```

↑ ↑  
Date Time

(demo)

# Regular Expressions

Take a single instance of a string:

26/Jan/2014

Use regex to generalize the pattern:

(.)/(.)/(. . .)      Use parentheses to specify fields to extract.

(.+)/(.+)/(.+)

(\d+)/([a-zA-Z]+)/(\d+)

## Meta Characters

	Description	Example	Match	No Match
.	Any character except \n	...	abc	ab abcd
[ ]	Any character inside brackets	[cb.]ar	bar .ar	jar
*	≥0 or more of last symbol	[pb]*ark	bbpark ark	dark
+	≥1 last symbol	[pb]+ark	bbpark bark	dark ark
?	Last symbol optional	s?he	she he	the
	Before or after	we us	we us	e s

## Meta Characters

	Description	Example	Match	No Match
[ ]	Any character inside brackets	[A-Z.]ar	Bar .ar	jar
[^ ]	Any character not inside brackets	[^a-z]ar	Bar :ar	car
\	Escapes next character	\[hi\]	[hi]	hi
^	Beginning of line	^ark	ark two	dark
\$	End of line	ark\$	noahs ark	arkk

## Shorthand Characters

Bracket Form	Shorthand	Description
[a-zA-Z0-9_]	\w	Alphanumeric character
[^\w]	\W	Not an alphanumeric char
[0-9]	\d	Digit
[^\d]	\D	Not a digit
[\t\n\f\r\p{Z}]	\s	Whitespace
[^\s]	\S	Not whitespace

## Survey Time

### Question 1

Today, I was honored to be joined by Republicans and Democrats from both the House and Senate, as well as members of my Cabinet - to discuss the urgent need to rebuild and restore America's depleted infrastructure. <http://45.wh.gov/UDL9yE&nbsp;pic.twitter.com/BVBRDvHfCC>

### Question 2

[ftp://file\\_server.com:21/top\\_secret/life\\_changing\\_plans.pdf](ftp://file_server.com:21/top_secret/life_changing_plans.pdf)  
<https://regexone.com/lesson/introduction#section>  
[file://localhost:4040/zip\\_file](file://localhost:4040/zip_file)  
<https://s3cur3-server.com:9999/>  
<market://search/angry%20birds>

## String vs. Regex

str	re
	re.findall(pat, st)
str.replace(old, new)	re.sub(pat, repl, st)
str.split(sep)	re.split(pat, st)
'ab' in str	re.search(pat, st)

<https://docs.python.org/3/library/re.html>

(demo)

# Use Case 3: Deriving Features

New Tool: pandas String Methods

(demo)

## String vs. Regex vs. Pandas

str	re	pandas
	re.findall	vio.str.findall
str.replace	re.sub	vio.str.replace
str.split	re.split	vio.str.split
'ab' in str	re.search	vio.str.contains
len(str)		vio.str.len
str[1:4]		vio.str[1:4]

<https://pandas.pydata.org/pandas-docs/stable/text.html>

## Top 20 Violations

- unclean or degraded floors walls or ceilings
- moderate risk food holding temperature
- inadequate and inaccessible handwashing facilities
- unapproved or unmaintained equipment or utensils
- inadequately cleaned or sanitized food contact surfaces
- wiping cloths not clean or properly stored or inadequate sanitizer
- improper food storage
- foods not protected from contamination
- moderate risk vermin infestation
- high risk food holding temperature
- unclean nonfood contact surfaces
- food safety certificate or food handler card not available
- unclean or unsanitary food contact surfaces
- inadequate food safety knowledge or lack of certified food safety manager
- improper storage of equipment utensils or linens
- low risk vermin infestation
- permit license or inspection report not posted
- improper cooling methods
- unclean hands or improper use of gloves
- improper or defective plumbing

## What Kinds of Features?

- Cleanliness 'clean|sanit'
- High risk 'high risk'
- Vermin 'vermin'
- Surfaces 'wall|ceiling|floor|surface'
- Humans 'hand|glove|hair|nail'
- Permits and Certification 'permit|certif'

(demo)

## In this lecture

### Simple

Python string methods

Looping

### Complex

Regular expressions

pandas string functions

## In this lecture

