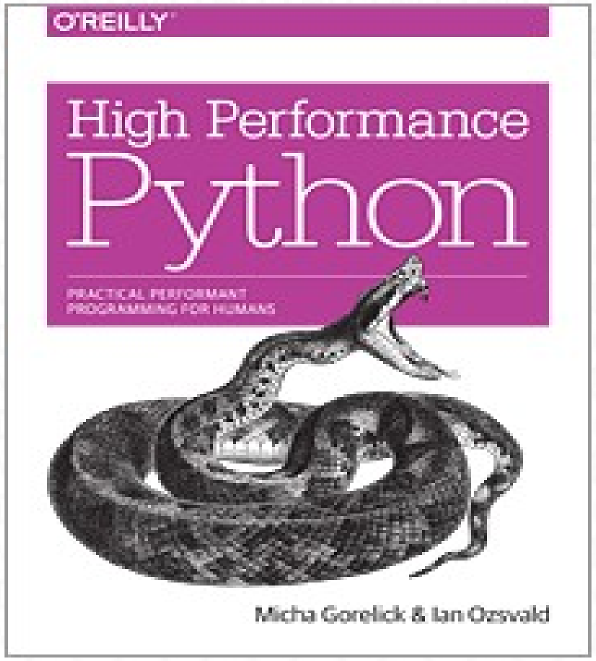
Cleaning Confused Collections of Characters @ PyDataParis 2015

Ian Ozsvald @IanOzsvald ModelInsight.io

Who am I?

- Past speaker+teacher at PyDatas, EuroPythons, PyCons, PyConUKs
- Co-org of PyDataLondon
- O'Reilly Author
- ModelInsight.io for NLP+ML

  IP creation in London
- “I clean data” #sigh


O'REILLY
High Performance
Python
PRACTICAL PERFORMANT PROGRAMMING FOR HUMANS
Micha Gorelick & Ian Ozsvald

Unstructured data->Value

- Increasing rate of growth and meant for human consumption
- Hard to:
  - Extract
  - Parse
  - Make machine-readable
- It is also very valuable...part of my consultancy - we're currently automating recruitment:
- Previously - eCommerce recomm., house price & logistics prediction
- Uses: search, visualisation, new ML features
- Most industrial problems messy, not “hard”, but time consuming!
- How can we make it easier for ourselves?

What can we extract?

- Plain text (languages? platforms? broken files?)
- HTML and XML (e.g. ePub)
- PDFs
- PDF tables
- Image containing text
- (Audio files with speech)

Extracting from HTML/XML

- Assuming you've scraped (e.g. scrapy)
- regular expressions (brittle)
- BeautifulSoup
- xpath via scrapy or lxml
  - s.xpath('.//span[@class="at_sl"]/text()')[0].extract()
- *You need unit tests*

Extracting text from PDFs & Word

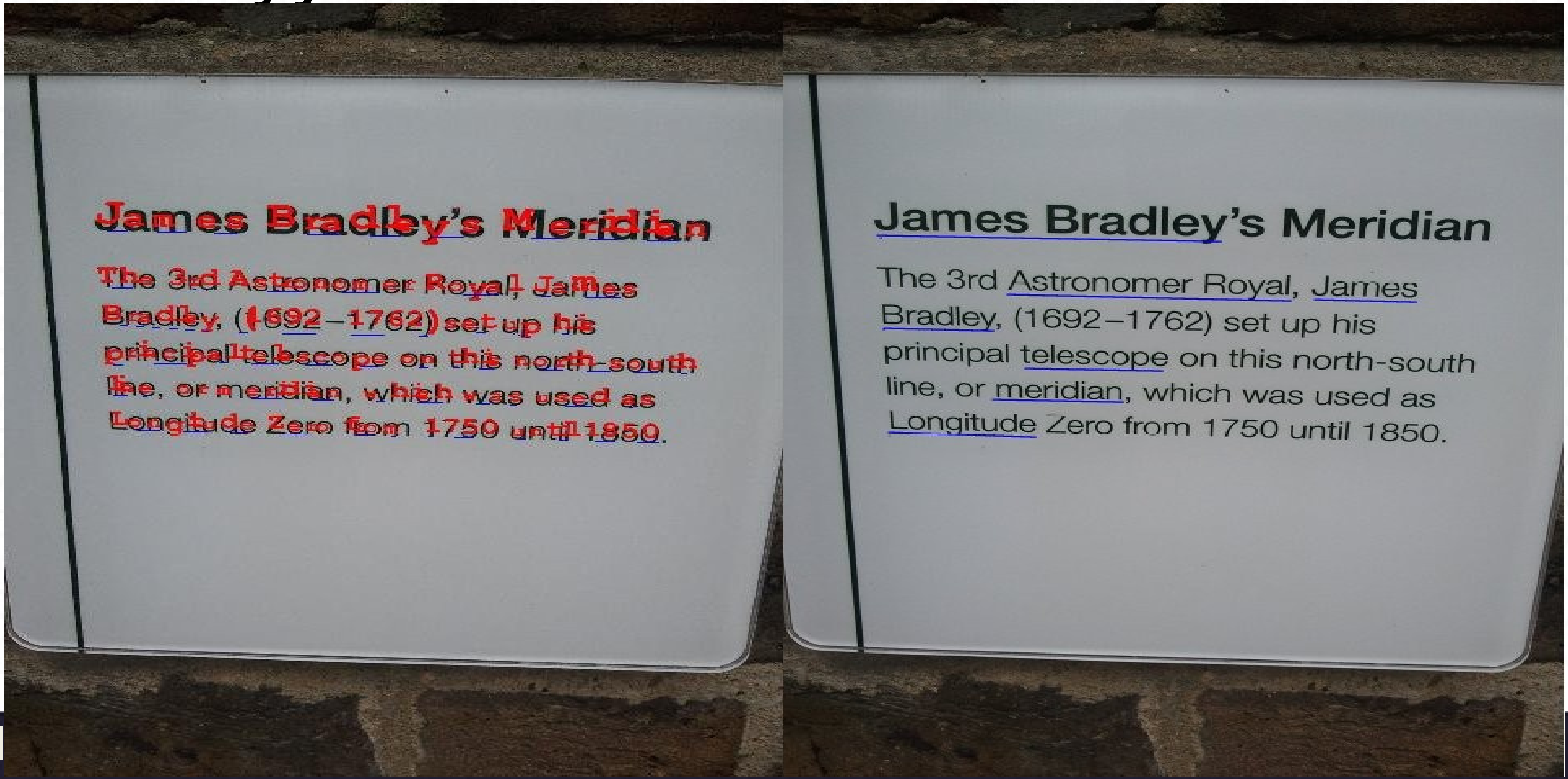
- pdftotext (Linux), pdfminer (Python 2 with 3 port, 'slow')
- Apache Tika (Java - via jnius?)
- http://textract.readthedocs.org/en/latest/ (Python)
- Difficulties
    - Formatting probably horrible
    - No semantic interpretation (e.g. CVs)
    - Keyword stuffing, images, out-of-order or multi-column text, tables #sigh
- “Content ExtRactor and MINEr” (CERMINE) for academic papers
- Commercial CV parsers (e.g. Sovren)

Extracting tables from PDFs

- ScraperWiki's https://pdftables.com/ (builds on pdfminer)
- http://tabula.technology/ (Ruby/Java OS, seems to require user intervention)
- messytables (Python, lesser known, autoguesses dtypes)

Extracting text from Images

- OCR e.g. tesseract (below)
- Abbyy's online commercial service

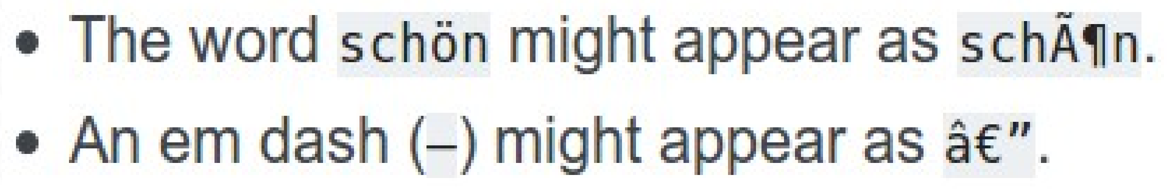


James Bradley's Meridian

The 3rd Astronomer Royal, James Bradley, (1692–1762) set up his principal telescope on this north-south line, or meridian, which was used as Longitude Zero from 1750 until 1850.

Fixing badly encoding text

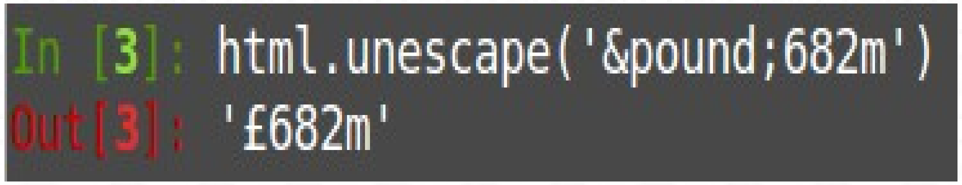
- http://ftfy.readthedocs.org/en/latest/



- The word `schön` might appear as `schÃ¶n`.
- An em dash (`—`) might appear as `â€”`.

- HTML unescaping:

  (also ftfy)



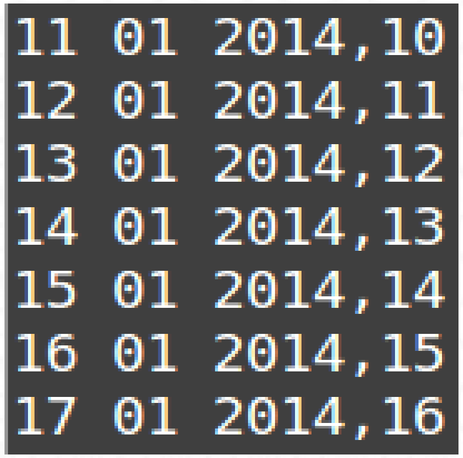
```
In [3]: html.unescape('&pound;682m')
Out[3]: '£682m'
```

- chromium-compact-language-detector will guess human language from 80+ options (so you can choose your own decoding options)

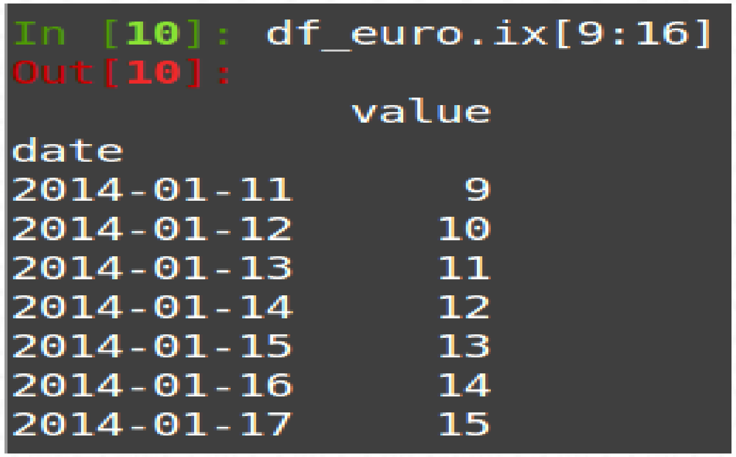
Interpreting dtypes

- Use pandas to get text table (e.g. from JSON/CSV)
- Dates are problematic unless you know their format (next slides), Labix dateutil helpful
- Categories (e.g. “male”/”female”) are easily spotted by eye
- [“33cm”, “22inches”, ...] could be easily converted
- *Could you write a module to suggest possible conversions on dataframe for the user (and notify if ambiguities are present e.g. 1/1 to 12/12...MM/DD or DD/MM)?*

Date examples



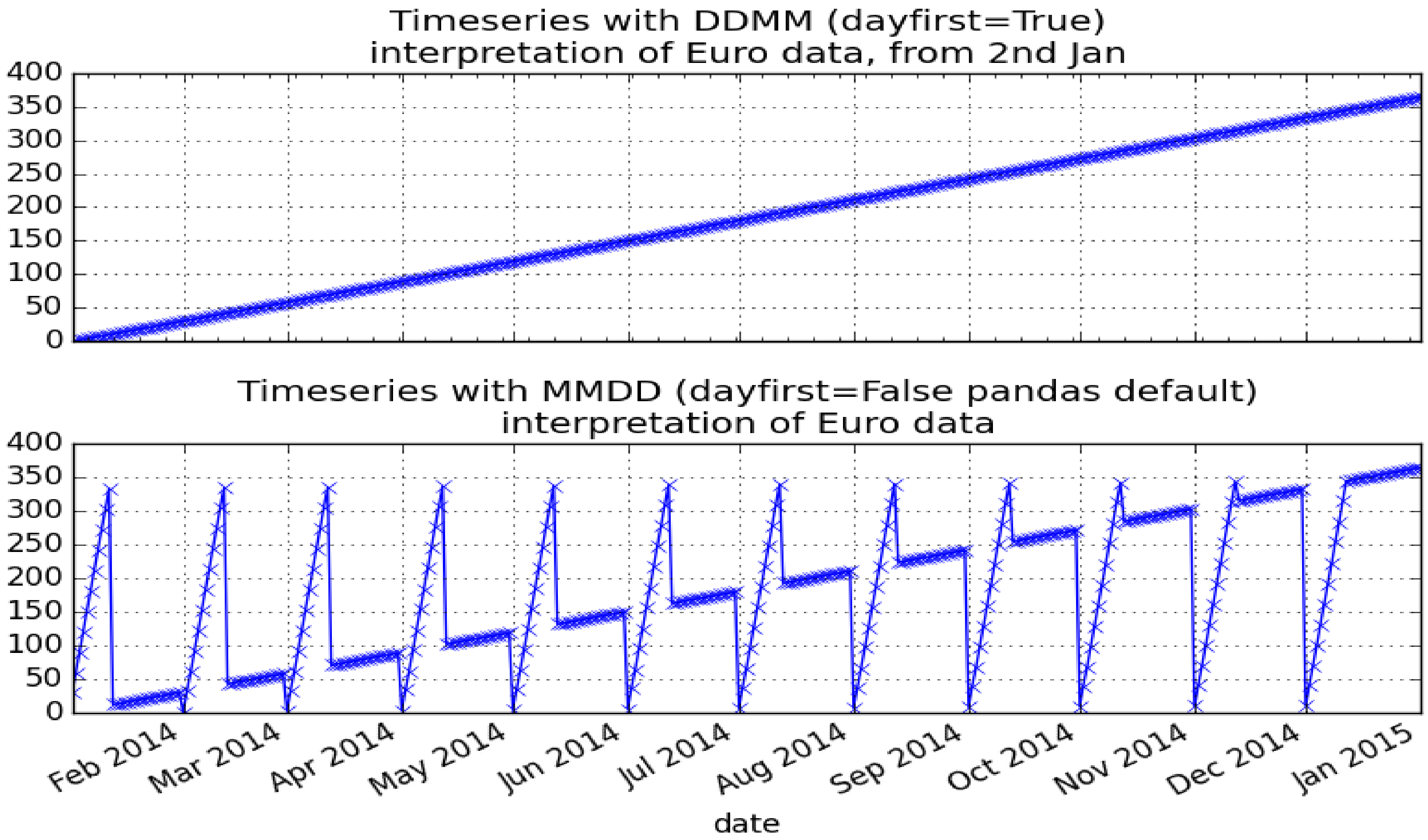
```
11 01 2014,10
12 01 2014,11
13 01 2014,12
14 01 2014,13
15 01 2014,14
16 01 2014,15
17 01 2014,16
```



```
In [10]: df_euro.ix[9:16]
Out[10]:
            value
date
2014-01-11      9
2014-01-12     10
2014-01-13     11
2014-01-14     12
2014-01-15     13
2014-01-16     14
2014-01-17     15
```

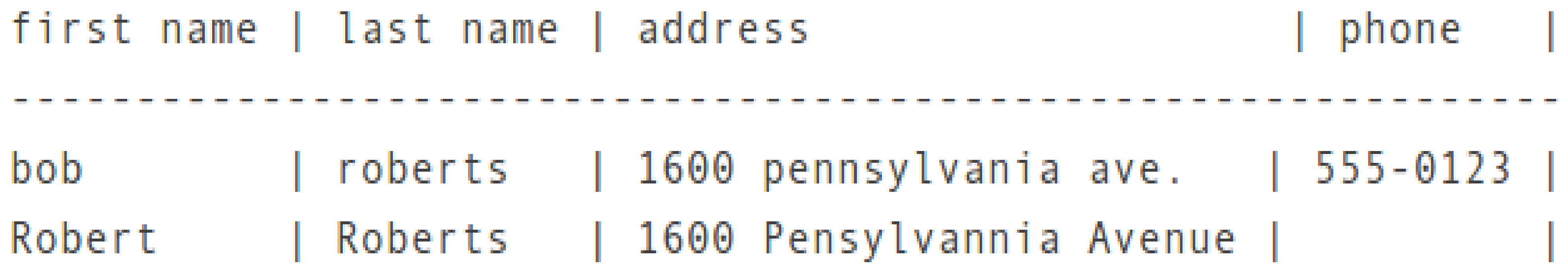
- The default is for US-style (MMDD), not Euro-style (DDMM)
- pd.from_csv(parse_dates=[*cols*], **dayfirst=False**)

MMDD (default) vs DDMM



Merging two data sources

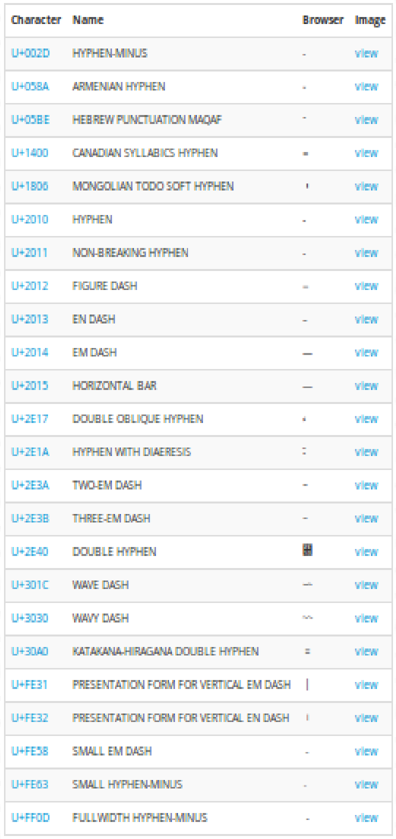
- pd.merge(df1, df2) # exact keys, SQL-like
- fuzzywuzzy/metaphone for approximate string matching
- DataMade's dedupe.readthedocs.org to identify duplicates (or OpenRefine)



```
first name | last name | address                  | phone    |
-----------------------------------------------------------------
bob        | roberts   | 1600 pennsylvania ave.   | 555-0123 |
Robert     | Roberts   | 1600 Pensylvannia Avenue |          |
```

Manual Normalisation

- Eyeball the problem, solve by hand
- *Lots of unit-tests!*
- lower() # “Accenture”->”accenture”
- strip() # “   this and ”->”this and”
- replace(<pattern>,””) # “BigCo Ltd”->”BigCo”
- unidecode # “áéîöũ”->”aeiou”
- normalise unicode (e.g. >50 dash variants!)
- NLTK stemming & WordNet ISA relt.



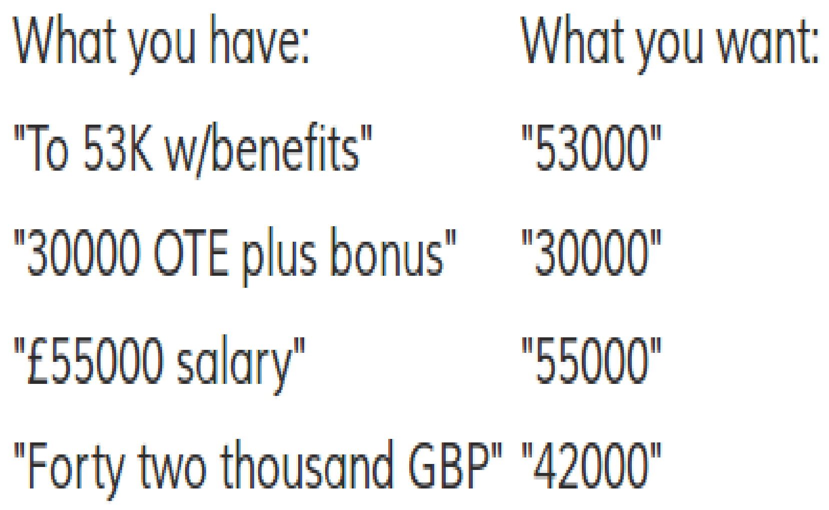
Character	Name	Browser	Image
U+002D	HYPHEN-MINUS	-	view
U+058A	ARMENIAN HYPHEN	֊	view
U+05BE	HEBREW PUNCTUATION MAQAF	־	view
U+1400	CANADIAN SYLLABICS HYPHEN	᐀	view
U+1806	MONGOLIAN TODO SOFT HYPHEN	᠆	view
U+2010	HYPHEN	‐	view
U+2011	NON-BREAKING HYPHEN	‑	view
U+2012	FIGURE DASH	‒	view
U+2013	EN DASH	–	view
U+2014	EM DASH	—	view
U+2015	HORIZONTAL BAR	―	view
U+2E17	DOUBLE OBLIQUE HYPHEN	⸗	view
U+2E1A	HYPHEN WITH DIAERESIS	⸚	view
U+2E3A	TWO-EM DASH	⸺	view
U+2E3B	THREE-EM DASH	⸻	view
U+2E40	DOUBLE HYPHEN	⹀	view
U+301C	WAVE DASH	〜	view
U+3030	WAVY DASH	〰	view
U+30A0	KATAKANA-HIRAGANA DOUBLE HYPHEN	゠	view
U+FE31	PRESENTATION FORM FOR VERTICAL EM DASH	︱	view
U+FE32	PRESENTATION FORM FOR VERTICAL EN DASH	︲	view
U+FE58	SMALL EM DASH	﹘	view
U+FE63	SMALL HYPHEN-MINUS	﹣	view
U+FF0D	FULLWIDTH HYPHEN-MINUS	－	view

Rule lists

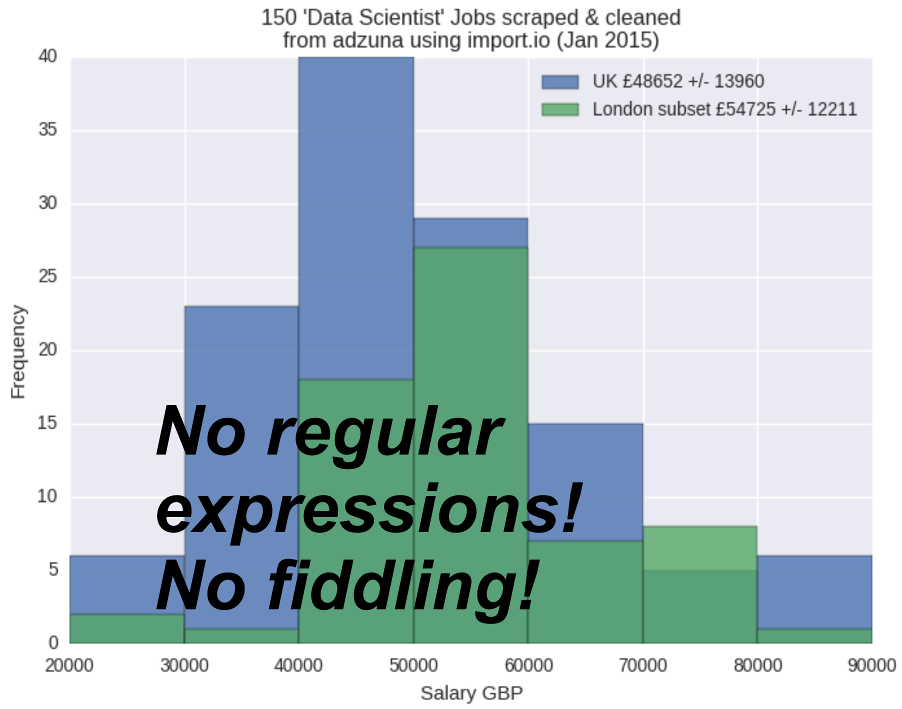
- Don't forget the old and simple approaches
- Big lists of exact-match rules
- Easy to put into a SQL DB for quick matches!
- A set/dict of strings is super-quick in Python (or use e.g. MarissaTrie)

Automated Normalisation?

- My annotate.io
- Why not make the machine do this for us?



What you have:	What you want:
"To 53K w/benefits"	"53000"
"30000 OTE plus bonus"	"30000"
"£55000 salary"	"55000"
"Forty two thousand GBP"	"42000"



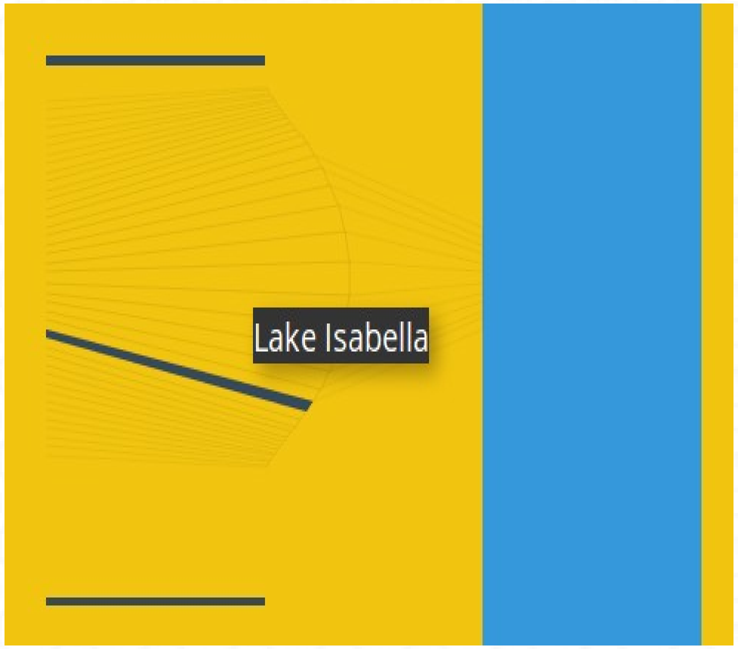
No regular expressions!
No fiddling!

Machine Learn the rules?

- What if we extend dedupe's idea?
- Can we give examples of e.g. company names that are similar and generalise a set of rules for unseen data?
- Could we train given a small set of data and re-train when errors occur on previously unseen data?

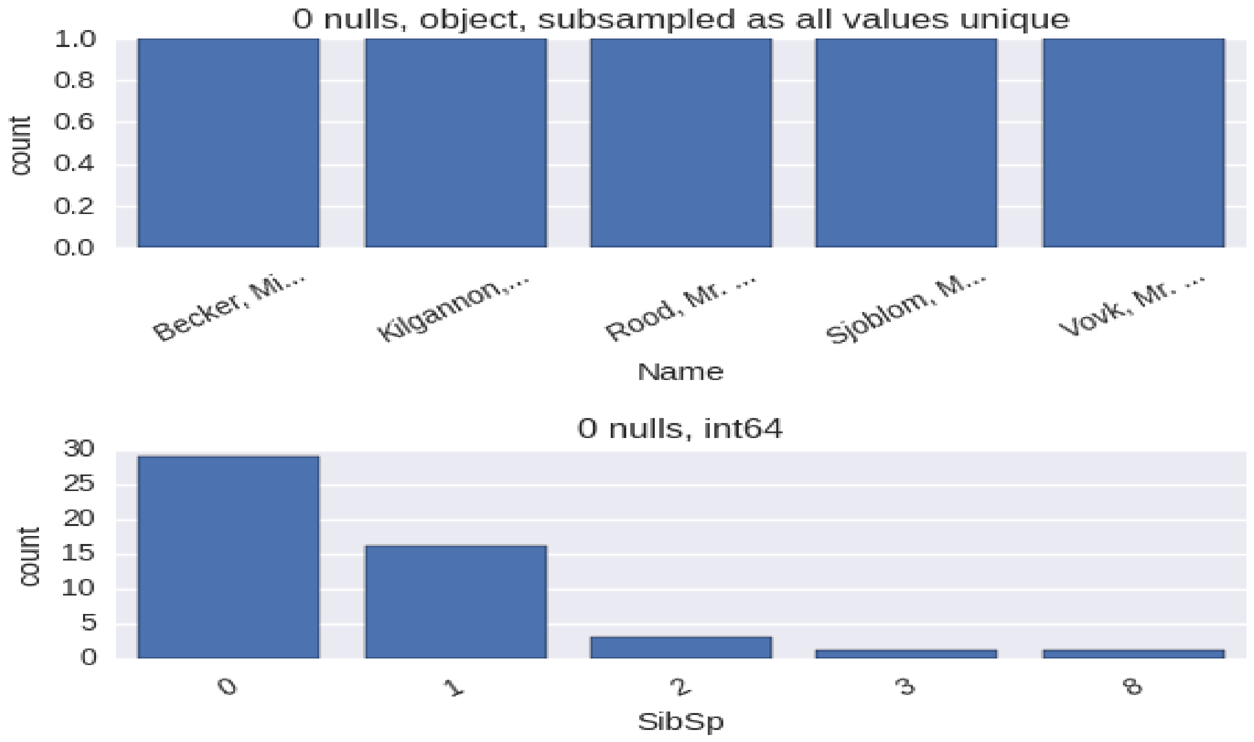
Visualising new data sources

- setosa.io/csv-fingerprint/
- SeaBorn (or Bokeh?)
- Do you have good tools?



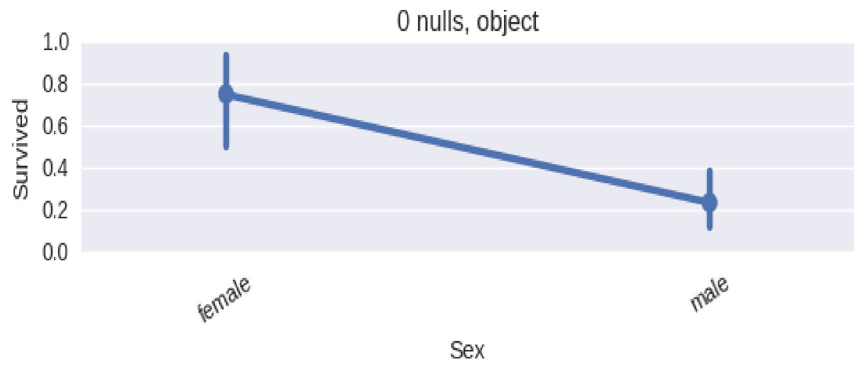
Visualising new data sources

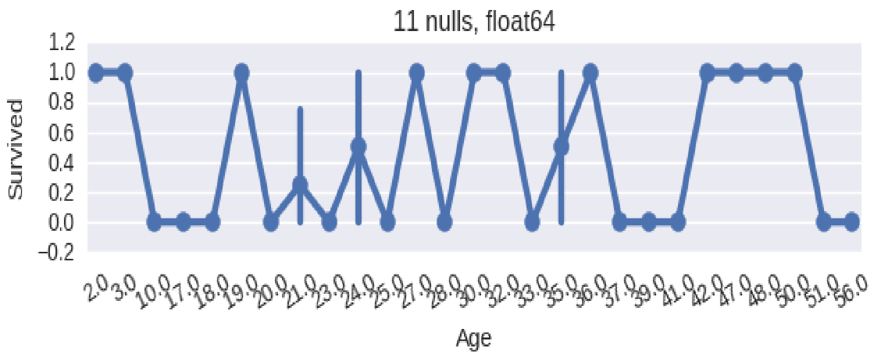
- github /ianozsvald/dataframe_visualiser

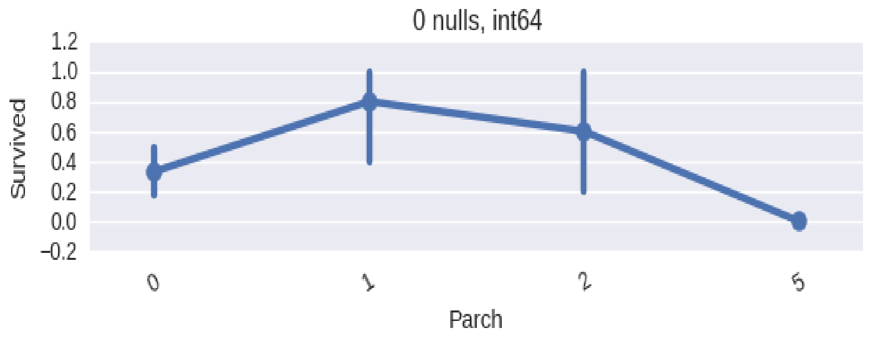


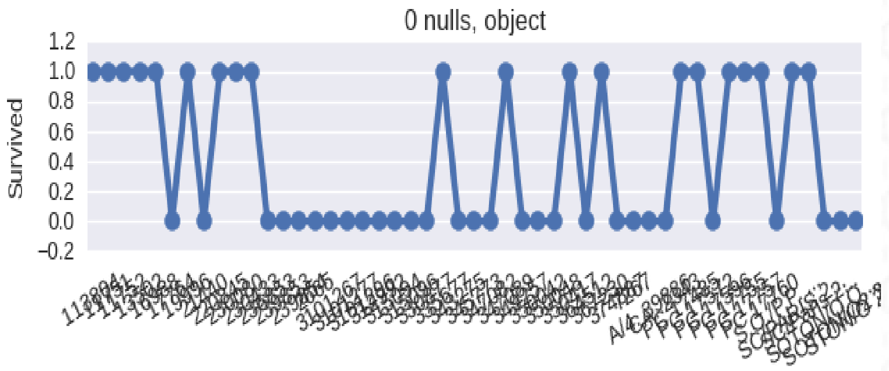
Visualising new data sources

- github /ianozsvald/dataframe_visualiser









Automate feature extraction?

- Can we extract features from e.g. Id columns (products/rooms/categories)?
- We could identify categorical labels and suggest Boolean column equivalents
- We could remove some of the leg-work...you avoid missing possibilities, junior data scientists get “free help”
- What tools do you know of and use?

Getting started

- Cleaning is more R&D than engineering
- FACT: Your data has missing items + it lies
- Visualise it
- Set realistic milestones, break into steps, have lots of tests
- Have a gold standard for measuring progress
- Aim for a high-quality output

Tips

- Lots of lesser-known good tools out there!
- APIs like Alchemy + DBPedia for Entity Recognition and Sentiment Analysis
- Python 3.4+ makes Unicode easier
- USE: pandas, StackOverflow

Closing...

- Give me feedback on annotate.io
- *Give me your dirty-data horror stories*
- http://ianozsvald.com/
- PyDataLondon monthly meetup
- PyDataLondon conference late June(?)
- *Do you have data science deployment stories for my keynote at PyConSweden? What's "hardest" in data science for your team?*