Practical Session 7

Foundations Spatial Data Science

Today Goals & Aims

Practical goals

- Get familiar exploring textual data using NLTK (Natural Language Toolkit)
- Loading and tidying textual data
- Use of regular expressions
- Get familiar with Term Frequency and Inverse Document Frequency

Why are we doing this

- Textual Data complements the two other methods we have been exploring over the past weeks (Numeric Data and Spatial Data) to perform spatial data science.
- Textual Data contains a wealth of information that opens the opportunity to add some qualitative analysis in the usually predominant quantitative aspects of spatial data.

Today Goals & Aims

Term Calendar

			WORKSHOP		PRACTICAL Date	
	Weekly Topic		Lead	Date (Monday)	Groups 1,2,3 (Tuesday)	Groups 4,5,6 (Wednesday)
1	Getting Oriented	initiate	David, Nicolas	4 Oct	4 Oct	5 Oct
2	Foundations (Part 1)	initiate	Nicolas	11 Oct	11 Oct	12 Oct
3	Foundations (Part 2)	initiate	Nicolas	18 Oct	18 Oct	19 Oct
4	Objects & Classes	initiate	David	25 Oct	25 Oct	26 Oct
5	Numeric Data	engage	David	1 Nov	1 Nov	2 Nov
	Reading Week					
6	Spatial Data	engage	Nicolas	15 Nov	15 Nov	16 Nov
7	Textual Data	engage	Nicolas	22 Nov	22 Nov	23 Nov
8	Visualising Data	solve	David	29 Nov	29 Nov	30 Nov
9	Classifying Data	solve	David	6 Dec	6 Dec	7 Dec
10	Clustering Data	solve	Nicolas	13 Dec	13 Dec	14 Dec

Textual Data Analysis

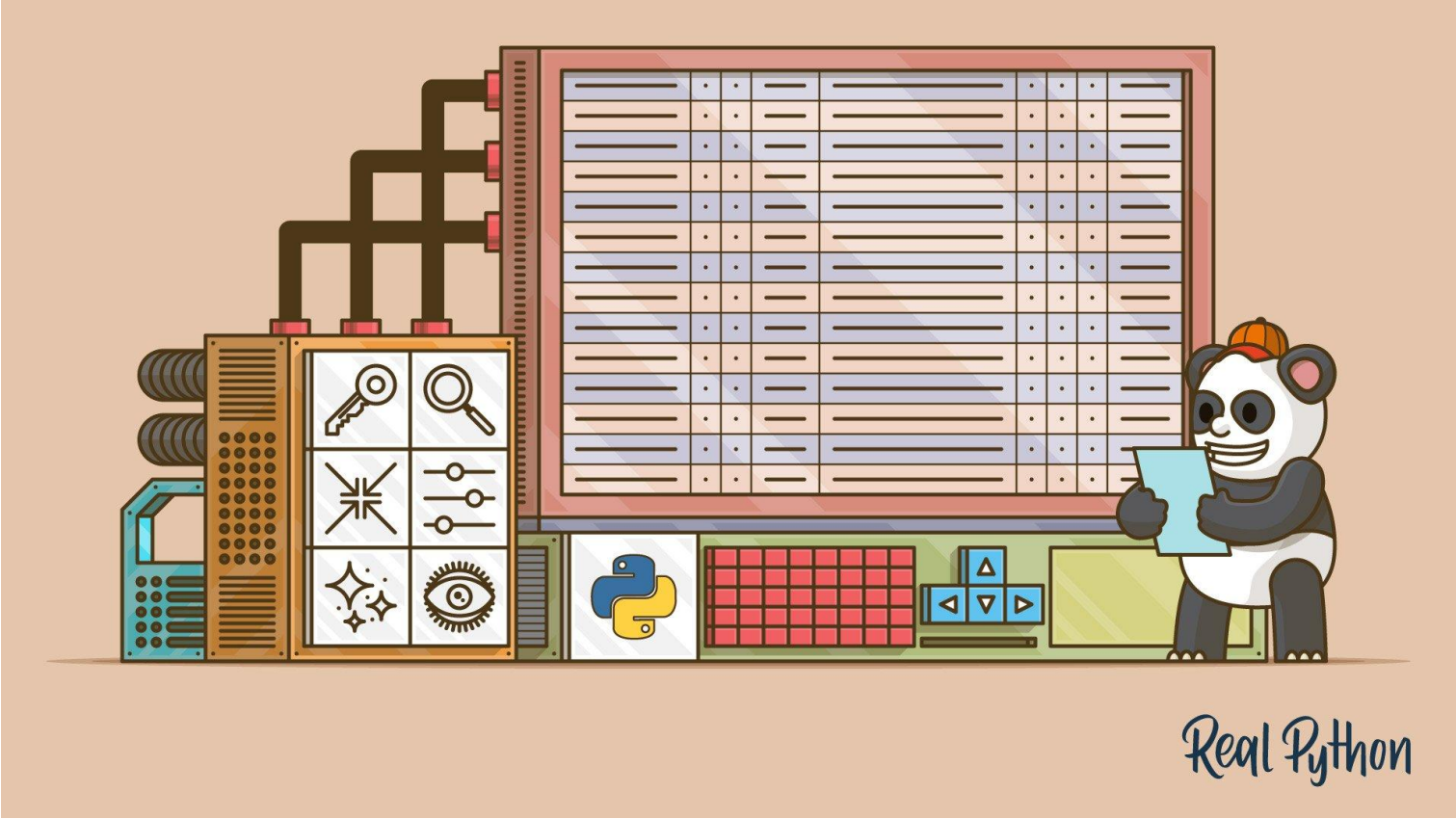

Image source: https://realpython.com/pandas-python-explore-dataset/

Regex - Keep in Mind!

Dictionaries:

1. No need to loop over the entire data set to find one data point!
    a. Use the index() function → `myData['Key'].index('Value')`

1. Every values associated with a key will be of the same type → `type(myData['Key'][0])`

Today's Agenda

1. Exploratory Data Analysis (EDA) - Recap
2. Basic Text Data Pre-processing
3. Cleaning Text Data
4. Prepare Data for EDA

Exploratory Data Analysis - Textual Data

"Process of exploring data, generate insights, test hypotheses, check assumption and reveal underlying hidden pattern in the data."

Basic Text Data Pre-Processing

1. Remove null value `df.dropna(inplace=True)`
2. Remove unwanted data (drop all columns that we will not use for the analysis) `df = df[['col_A', 'col_B', 'col_L']]`
3. Remove value above, below specific threshold
4. Remove duplicate values -> `df.['column'].unique()`

Cleaning Text Data

1. Expand contractions [“can’t”:“cannot”, “don’t”:“do not”]
2. Transform to lowercase

```
df.[‘column_lower’] = df.[‘column’].apply(lambda x: x.lower())
```

3. Remove digits [0-9]

```
df.[‘column_nodigits’] = df.[‘column’].apply(lambda x: re.sub(‘\w*\d\d*’, ‘’, x)
```

4. Remove punctuations

```
df['col_nopunct']=df['col'].apply(lambda x: re.sub('[%s]' %
re.escape(string.punctuation), '', x))
```

Cleaning Text Data - Regex

Anchors - ^and $

^The	matches any string that **starts with The** -> **Try it!**
end$	matches a string that **ends with end**
^The end$	**exact string match** (starts and ends with **The end**)
roar	matches any string that **has the text roar in it**

Image Link: https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285

Cleaning Text Data - Regex

Character classes - \d \w \s

```
\d          matches a single character that is a digit -> Try it!

\w          matches a word character (alphanumeric character plus
underscore) -> Try it!

\s          matches a whitespace character (includes tabs and line
breaks)

.           matches any character -> Try it!
```

Cleaning Text Data - Use Cases

Image Link: https://www.analyticsvidhya.com/blog/2020/01/4-applications-of-regular-expressions-that-every-data-scientist-should-know-with-python-code/

Cleaning Text Data - Use Cases

1. **Data validation** - verify email address, password requirement
2. **Data scraping** - find all webpages with a specific word
3. **Data wrangling** - Transform “raw” data into another format
4. **String parsing** - Capture URL
5. **String replacement** - Replace all “**.**” with “**,**”
6. And many others - file renaming, syntax highlighting,

Preparing Text Data for EDA

There are different library to prepare text data. You will most probably come across NLTK, spaCy and Gensim.

1. **Stopwords Removal**
2. **Lemmatization**
3. **Document Term Matrix**

Preparing Text Data - Stopwords

- **Most common words** in any natural language (the, is, in, at)
- Do **not add value** to the meaning of a document.
- Before removing stopwords, you need to perform **Tokenization**

- Remove it for tasks like **text classification** (spam filtering, language classification), **auto-tag generation**
- Do NOT remove it for tasks such as machine translation, text summarization, question-answering problems

Stemming & Lemmatization

Stemming

- Cuts off the end or beginning of a word
- Rudimentary rule-based process of stripping the suffixes (ing, ly, es, s, …)

Lemmatization

- Normalisation of words. Reduce a words to its root form.
- Use dictionary and word structure and grammar relations to reduce the lemma or words.

Document Term Matrix - Bag-of-Words

Example:

- Review 1: This movie is very scary and long. -> [1 1 1 1 1 1 1 0 0 0 0]
- Review 2: This movie is not scary and is slow. -> [1 1 2 0 0 1 1 0 1 0 0]
- Review 3: This movie is spooky and good. -> [1 1 1 0 0 0 1 0 0 1 1]

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Document Term Matrix - Bag-of-Words

Problems:

- Length of vectors continuously increase if we have new sentences with new words as our vocabulary (list of all words) increases.
- Vectors contain many 0. This result in large sparse matrix.
- No information on the grammar of the sentences or on the ordering of the words in the text is retained.

Document Term Matrix - TF-IDF

TF-IDF - Term Frequency Inverse Document Frequency:

- Numerical statistic that aims to reflect **how important** a word is to a document in the collection of words.
- **Term Frequency:** Measure how frequently a term, t, appears in a document, d.
- **Inverse Document Frequency:** Measure how important a term is. In the number of documents, how many document have the term, t.

Document Term Matrix - TF-IDF

Example:

- Review 1: This movie is very scary and long.
- Review 2: This movie is not scary and is slow.
- Review 3: This movie is spooky and good.

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

Topic Modelling

[The Guardian Food Recipes — Topic Modeling | by Aude Vuilli]

Next Week

Move on to the last part of the module. Review methods for problem solving starting with data visualisation. Look at how to link data and spatial data as well as group, aggregate and visualise data.

Week 8 READINGS - Visualisation and Gentrification

- D'Ignazio, Catherine, and Lauren F. Klein. 2020. *Data Feminism*. MIT Press. "*Chapter 3: On rational, scientific, objective viewpoints from mythical, imaginary, impossible standpoints.*" https://bookbook.pubpub.org/data-feminism.
- Badger, E., Q. Bui, and R. Gebeloff. 2019. *"Neighborhood Is Mostly Black. The Home Buyers Are Mostly White. New York Times."* New York Times. https://www.nytimes.com/interactive/2019/04/27/upshot/diversity-housing-maps-raleigh-gentrification.html

Time to practice !