Data Analyst's Toolbox: R and Python

Institute of Political Science, NSYSU, 2020-12-25

Kuo, Yao-Jen yaojenkuo@datainpoint.com (mailto:yaojenkuo@datainpoint.com) from DATAINPOINT (https://www.datainpoint.com)

TL; DR

Source: https://memes.tw/ (https://memes.tw/)

About me

Teaching practical data science online/offline, for individuals

- 如何成為資料分析師：從問題解決到行動方案，Hahow 好學校 (https://hahow.in/cr/dajourney)
- Visualization and modern data science, Adjunct Instructor, National Taiwan University
- Programming and business analytics, Adjunct Instructor, National Taiwan Normal University
- Python for data analysis, Instructor, Chunghwa Telecom Academy
- Python for data science, Machine learning from scratch, Senior Instructor, CSIE Training Program, National Taiwan University

Also for commercial banking clients

- 2020 DBS Training Program
- 2019 HNCB Training Program
- 2017 ESUN Training Program

Writing books

- 新手村逃脫！初心者的 Python 機器學習攻略 (https://www.books.com.tw/products/0010867390)
- 進擊的資料科學 (https://www.books.com.tw/products/0010827812)
- 輕鬆學習 R 語言 (https://www.books.com.tw/products/0010835361)

Writing blogs

- Medium (https://medium.com/@tonykuoyj)
- Substack (https://datainpoint.substack.com/about)
- 方格子 (https://vocus.cc/user/@yaojenkuo)

Before being a instructor

- Working experience
    - Senior Data Analyst, Coupang Shanghai
    - Analytical Consultant, SAS Taiwan
    - Management Associate, Chinatrust Banking Corporation Taiwan
    - Research Assistant, McKinsey & Company Taiwan
- Education
    - MBA, National Taiwan University
    - BA, National Taiwan University

Loves running with a marathon PR of 2:43:12 at 2019 Seoul Marathon

Source: https://giphy.com (https://giphy.com)

What is data analysis

The definition

> *We generate questions about a specific topic, we search for answers by exploring, transforming, and modelling data referring to our topic. And then use what we've learned to refine questions or generate new questions.*

Source: R for Data Science (https://r4ds.had.co.nz/)

Why data analysis

It is now an era of data-driven strategic thinking, and is probably never coming back.

The three means of persuasion that an orator must rely on

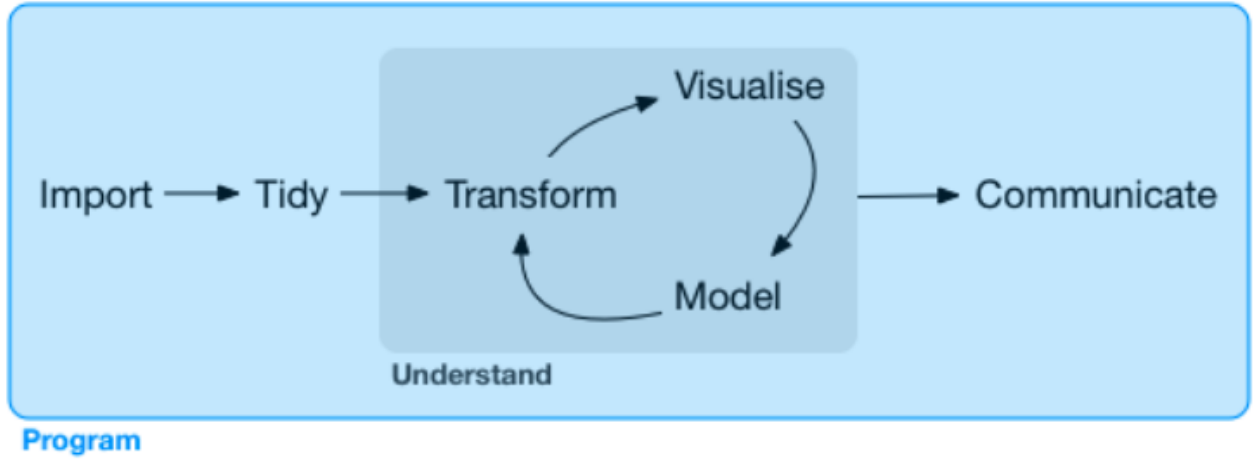
- Ethos
- Pathos
- Logos

Source: Aristotle, Rhetoric (https://en.wikipedia.org/wiki/Rhetoric)

It is a lot easier to persuade via ethos or pathos, but it takes time

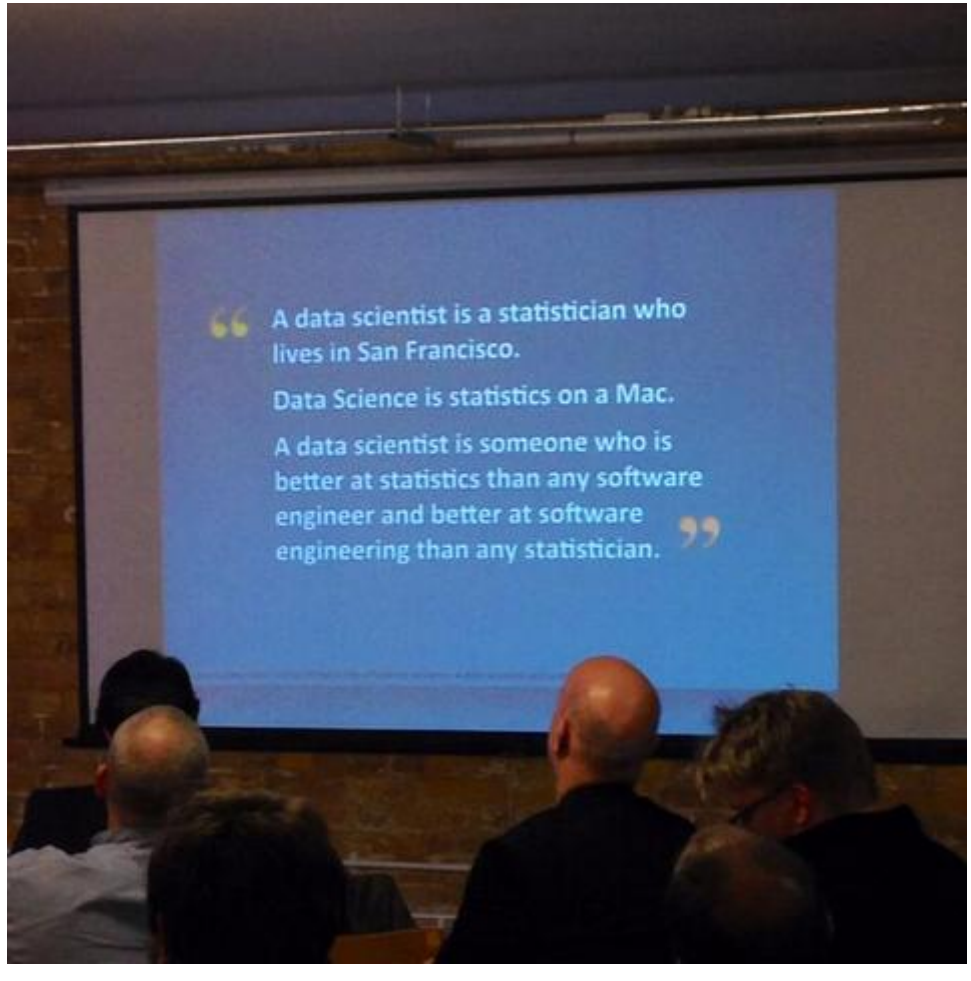
However, logos can be easily acquired once it is a fact and can be proven. Hence, data analysis is often the express way to logos.

Modern data analysis can be illustrated as the flow of data



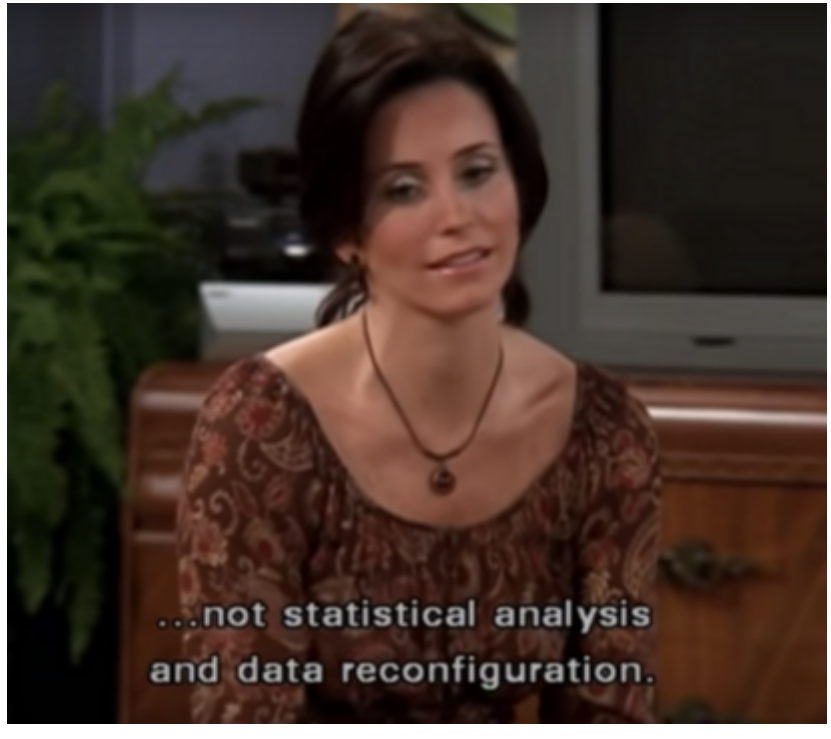
Source: [R for Data Science (https://r4ds.had.co.nz/)]

The funny definitions



Source: https://twitter.com/cdixon/status/428914681911070720/photo/1 (https://twitter.com/cdixon/status/428914681911070720/photo/1)

Source: https://www.warnerbros.com/tv/friends/ (https://www.warnerbros.com/tv/friends/)



Source: https://www.warnerbros.com/tv/friends/ (https://www.warnerbros.com/tv/friends/)

The serious definition

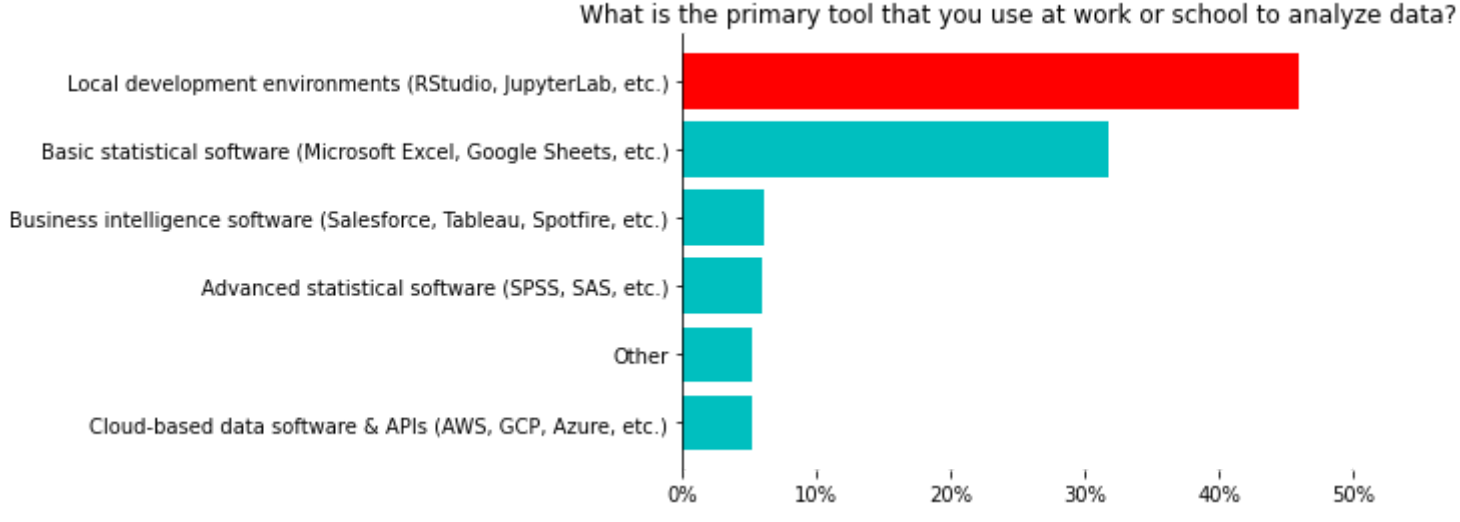
Modern data analysis invovles applications and tools like importing, tidying, transformation, visualization, modeling, and communication. Surrounding all these is programming.

Use programming language to analyze data

Let's review a question from [2020 Kaggle ML & DS Survey (https://www.kaggle.com/c/kaggle-survey-2020)](https://www.kaggle.com/c/kaggle-survey-2020):

> *What is the primary tool that you use at work or school to analyze data?*

In [3]:
```
plot_ans_38(ans_38)
```



It seems inevitable to write codes in modern data analysis

Source: https://giphy.com/ (https://giphy.com/)

Simply put, we can choose any programming language as long as it is capable of

- Importing data
- Tidying data
- Transforming data
- Visualizing data
- Modeling data
- Communicating data

Well, actually a lot of programming languages are capable of doing these

- Python
- R
- Julia
- Scala
- Matlab
- SAS
- ...etc.

How to choose among so many alternatives?

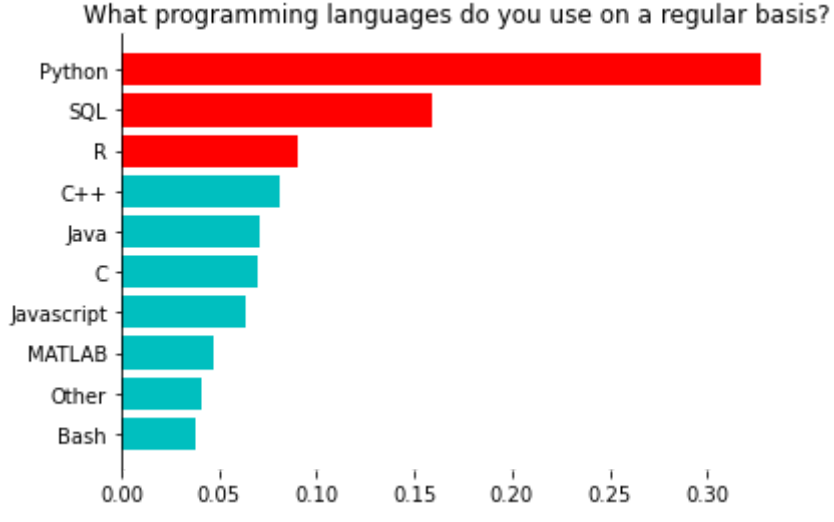
- The philosophy of "Eating a water mellon".
- The full support of scientific computing.
- Our objectivity.

The philosophy of ”Eating a water mellon”

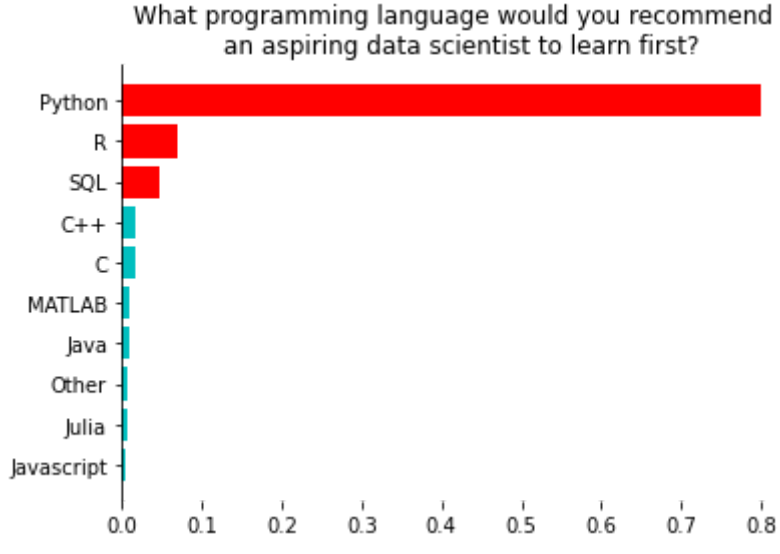
Let's review another 2 questions from [2020 Kaggle ML & DS Survey (https://www.kaggle.com/c/kaggle-survey-2020)](https://www.kaggle.com/c/kaggle-survey-2020):

- What programming languages do you use on a regular basis?
- What programming language would you recommend an aspiring data scientist to learn first?

In [4]:
```
plot_ans(ans_7, "What programming languages do you use on a regular basis?")
```



In [5]:
```
plot_ans(ans_8, "What programming language would you recommend \n an aspiring data scien
tist to learn first?")
```



R and Python in Stack Overflow Trends

https://insights.stackoverflow.com/trends?tags=python%2Cr%2Cjulia%2Cscala%2Cmatlab%2Csas (https://insights.stackoverflow.com/trends?tags=python%2Cr%2Cjulia%2Cscala%2Cmatlab%2Csas)

The full support of scientific computing

- Does the language support vectorization?
- Does the language support various data format?
- Does the language support visualization?

Both R and Python support vectorization

- R uses built-in `vector` and `matrix`.
- Python uses a third-party `ndarray`.

Both R and Python support various data format

- R uses
    - built-in named `list` to support key-value storage
    - built-in `data.frame` to support tabular data
- Python uses
    - built-in `dict` to support key-value storage
    - third-party `DataFrame` to support tabular data

Both R and Python support visualization

- R uses
    - built-in basic plotting system to support static plotting
    - third-party `ggplot2` to support high-end static plotting
    - third-party `shiny` to support dynamic plotting
- Python uses
    - third-party `matplotlib` to support static plotting
    - third-party `seaborn` to support high-end static plotting
    - third-party `plotly` to support dynamic plotting

Last but not least, it depends on our objectivity

- Specific or general-purposed?
- Functional or object-oriented?
- ...etc.

We will generate our own objectivity once we start coding

Source: https://giphy.com (https://giphy.com)

Let’s write some codes to analyze data

Bringing up a topic

> 大選開票看哪個里最準？「章魚里」神預測告訴你。每次到了選舉，總是會有幾個里開票與大選結果相似，因此被各界視為重點關注的開票區域。

Source: https://www.cw.com.tw/article/5093012 (https://www.cw.com.tw/article/5093012)

We can generate some questions regarding this topic

- How to define「章魚里」?
- Can we find out「章魚里」based on 2020 presidential data?
- Can we find the similarity of our own village?

How to define 「章魚里」？

Basically, after a few literature search, you may find the definition of 「章魚里」 is quite ambigious. So we are using a much fancier metric: **cosine similarity**.

What is cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is defined to equal the cosine of the angle between them, which is also the same as the inner product of the same vectors normalized to both have length 1.

$$a = (a_1, a_2, a_3)$$
$$b = (b_1, b_2, b_3)$$
$$cos\theta = \frac{\sum_i (a_i \times b_i)}{\sqrt{\sum_i a_i^2} \times \sqrt{\sum_i b_i^2}}$$
$$= \frac{a \cdot b}{\|a\| \times \|b\|}$$

Source: https://en.wikipedia.org/wiki/Cosine_similarity (https://en.wikipedia.org/wiki/Cosine_similarity)

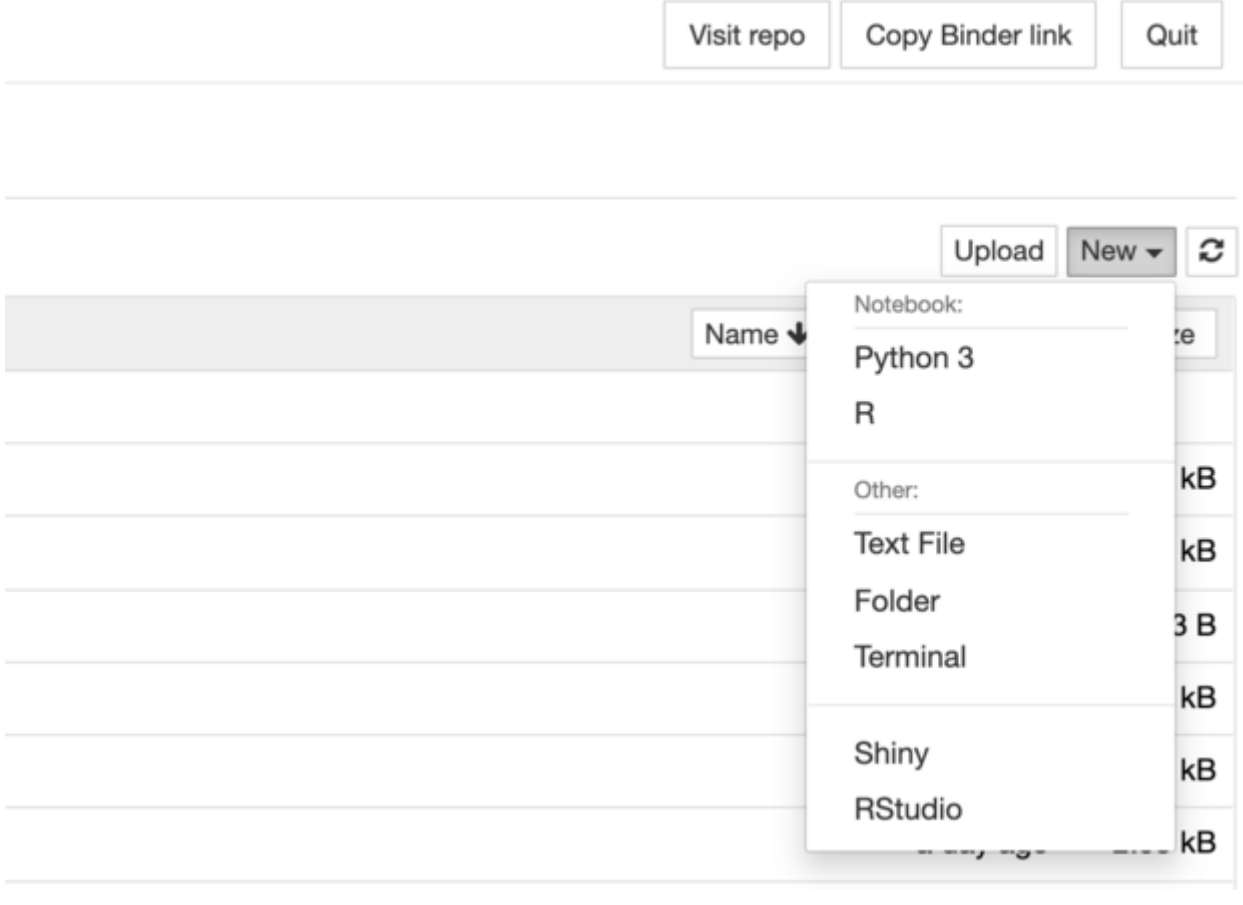
Can we find out 「章魚里」 based on 2020 presidential data？

Definitely, we all have access to the [Central Election Commission (https://db.cec.gov.tw/)](https://db.cec.gov.tw/).

We've downloaded these spreadsheets and prepared an in-browser environment for you.

(https://mybinder.org/v2/gh/yaojenkuo/talks/HEAD)

We can access to a Python notebook, R notebook, or RStudio in browser, no strings attached.



Besides a few kernels to execute, we also attached some data.

- Reading a CSV file.
- Reading a Excel spreadsheet.

In [6]:
```
# reading data via Python's pandas library
csv_df = pd.read_csv('presidential_2020.csv')
excel_df = pd.read_excel('presidential-2020/總統-A05-4-候選人得票數一覽表-各投開票所(南投縣).xls', skiprows=[0, 1, 3, 4])
```

Our CSV file is an integrated file after manipulations

In [7]:
```
csv_df.head()
```

Out[7]:

	county	town	village	office	number	candidate	votes
0	宜蘭縣	宜蘭市	民族里	1	1	宋楚瑜/余湘	37
1	宜蘭縣	宜蘭市	民族里	2	1	宋楚瑜/余湘	31
2	宜蘭縣	宜蘭市	建軍里	3	1	宋楚瑜/余湘	19
3	宜蘭縣	宜蘭市	建軍里	4	1	宋楚瑜/余湘	29
4	宜蘭縣	宜蘭市	泰山里	5	1	宋楚瑜/余湘	25

Our Excel spreadsheets are the original files downloaded from Central Election Commission (https://db.cec.gov.tw/)

In [8]:
```
excel_df.head()
```

Out[8]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	(1)\n宋楚瑜\n余湘	(2)\n韓國瑜\n張善政	(3)\n蔡英文\n賴清德	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11
0	總　計	NaN	NaN	13,315	133,791	152,046	299,152	3,555	302,707	13	302,720	110,765
1	南投市	NaN	NaN	3,077	26,690	30,910	60,677	693	61,370	3	61,373	20,480
2	NaN	龍泉里	1.0	26	241	391	658	8	666	0	666	228
3	NaN	康壽里	2.0	30	216	266	512	4	516	0	516	128
4	NaN	康壽里	3.0	25	239	306	570	8	578	0	578	154

We can also try importing via the RStudio interface.

```
library(readxl)

csv_df = read.csv('presidential_2020.csv')
excel_df = read_excel('presidential-2020/總統-A05-4-候選人得票數一覽表-各投開票所(南投縣).xls')
head(csv_df)
head(excel_df)
```

We write codes to integrate these spreadsheets into a CSV file

In [9]:
```
from presidential import Presidential

presidential = Presidential('presidential-2020')
presidential_df = presidential.adjust_presidential_df()
presidential_df.to_csv('presidential_2020.csv', index=False)
```

```
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(宜蘭縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(彰化縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(金門縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(桃園市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(苗栗縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(臺南市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(雲林縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(南投縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(高雄市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(臺北市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(新北市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(花蓮縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(新竹市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(新竹縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(基隆市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(連江縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(嘉義縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(嘉義市).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(屏東縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(澎湖縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(臺東縣).xls
Tidying 總統-A05-4-候選人得票數一覽表-各投開票所(臺中市).xls
```

In [10]:
```
presidential_df.head()
```

Out[10]:

	county	town	village	office	number	candidate	votes
0	宜蘭縣	宜蘭市	民族里	1	1	宋楚瑜/余湘	37
1	宜蘭縣	宜蘭市	民族里	2	1	宋楚瑜/余湘	31
2	宜蘭縣	宜蘭市	建軍里	3	1	宋楚瑜/余湘	19
3	宜蘭縣	宜蘭市	建軍里	4	1	宋楚瑜/余湘	29
4	宜蘭縣	宜蘭市	泰山里	5	1	宋楚瑜/余湘	25

In [11]:
```
presidential_df.tail()
```

Out[11]:

	county	town	village	office	number	candidate	votes
51673	臺中市	和平區	梨山里	1845	3	蔡英文/賴清德	132
51674	臺中市	和平區	梨山里	1846	3	蔡英文/賴清德	107
51675	臺中市	和平區	梨山里	1847	3	蔡英文/賴清德	40
51676	臺中市	和平區	平等里	1848	3	蔡英文/賴清德	24
51677	臺中市	和平區	平等里	1849	3	蔡英文/賴清德	102

Check if the summations are right with Python

In [12]:
```
ttl_votes = presidential_df['votes'].sum()
ttl_votes_by_candidates = presidential_df.groupby('number')['votes'].sum()
ttl_votes_by_candidates
```

Out[12]:
```
number
1     608590
2    5522119
3    8170231
Name: votes, dtype: int64
```

Check if the summations are right with R

```
library(dplyr)

csv_df %>%
    group_by(number) %>%
    summarise(ttl_votes = sum(votes))
```

National percentage is our target vector to be compared

In [13]:
```
national_percentage = ttl_votes_by_candidates / ttl_votes
national_percentage
```

Out[13]:
```
number
1    0.042556
2    0.386137
3    0.571307
Name: votes, dtype: float64
```

Total votes for each village

In [14]:
```
combined_key = presidential_df['county'].str.cat(presidential_df['town']).str.cat(presid
ential_df['village'])
presidential_df = presidential_df.assign(combined_key=combined_key)
ttl_votes_by_combined_key = presidential_df.groupby(['combined_key'])['votes'].sum()
ttl_votes_by_combined_key
```

Out[14]:
```
combined_key
南投縣中寮鄉中寮村      443
南投縣中寮鄉內城村      297
南投縣中寮鄉八仙村      535
南投縣中寮鄉和興村      422
南投縣中寮鄉崁頂村      304
                  ...
高雄市鼓山區鼓岩里      847
高雄市鼓山區鼓峰里     1425
高雄市鼓山區龍井里      906
高雄市鼓山區龍子里    11410
高雄市鼓山區龍水里    16333
Name: votes, Length: 7737, dtype: int64
```

Votes percentage by each candidate and village

In [15]:
```
ttl_votes_by_combined_key_candidates = presidential_df.groupby(['combined_key', 'number'
])['votes'].sum()
soong = ttl_votes_by_combined_key_candidates[:, '1'] / ttl_votes_by_combined_key
han = ttl_votes_by_combined_key_candidates[:, '2'] / ttl_votes_by_combined_key
tsai = ttl_votes_by_combined_key_candidates[:, '3'] / ttl_votes_by_combined_key
votes_obtained = pd.concat([soong, han, tsai], axis=1)
votes_obtained.columns = ['soong', 'han', 'tsai']
```

In [16]:
```
votes_obtained
```

Out[16]:

combined_key	soong	han	tsai
南投縣中寮鄉中寮村	0.040632	0.489842	0.469526
南投縣中寮鄉內城村	0.057239	0.474747	0.468013
南投縣中寮鄉八仙村	0.039252	0.435514	0.525234
南投縣中寮鄉和興村	0.021327	0.500000	0.478673
南投縣中寮鄉崁頂村	0.052632	0.381579	0.565789
...	...	...	...
高雄市鼓山區鼓岩里	0.014168	0.309327	0.676505
高雄市鼓山區鼓峰里	0.032982	0.473684	0.493333
高雄市鼓山區龍井里	0.023179	0.367550	0.609272
高雄市鼓山區龍子里	0.032340	0.381420	0.586240
高雄市鼓山區龍水里	0.037654	0.398947	0.563399

7737 rows × 3 columns

Calculate cosine similarity

In [17]:
```
a = national_percentage.values
a_norm = np.linalg.norm(a)
cos_similarities = []
for i in range(votes_obtained.shape[0]):
    b = votes_obtained.iloc[i, :].values
    b_norm = np.linalg.norm(b)
    ab = np.dot(a, b)
    cos_similarity = np.dot(a, b) / (a_norm*b_norm)
    cos_similarities.append(cos_similarity)
votes_obtained = votes_obtained.assign(cosine_similarity=cos_similarities)
votes_obtained = votes_obtained.reset_index()
```

In [18]:
```
votes_obtained.head()
```

Out[18]:

	combined_key	soong	han	tsai	cosine_similarity
0	南投縣中寮鄉中寮村	0.040632	0.489842	0.469526	0.977648
1	南投縣中寮鄉內城村	0.057239	0.474747	0.468013	0.980246
2	南投縣中寮鄉八仙村	0.039252	0.435514	0.525234	0.995217
3	南投縣中寮鄉和興村	0.021327	0.500000	0.478673	0.977015
4	南投縣中寮鄉崁頂村	0.052632	0.381579	0.565789	0.999882

Sort by cosine similarity with descending order to find 「章魚里」

In [19]:
```
votes_obtained.sort_values(['cosine_similarity', 'combined_key'], ascending=[False, True
]).reset_index(drop=True).head(10)
```

Out[19]:

	combined_key	soong	han	tsai	cosine_similarity
0	嘉義縣番路鄉內甕村	0.042553	0.386018	0.571429	1.000000
1	臺南市東區關聖里	0.042450	0.386295	0.571255	1.000000
2	臺南市中西區南門里	0.043410	0.385460	0.571130	0.999999
3	新北市汐止區保長里	0.042833	0.386847	0.570320	0.999999
4	新北市金山區五湖里	0.043765	0.385632	0.570603	0.999998
5	臺北市南港區東新里	0.042036	0.385298	0.572666	0.999997
6	臺北市內湖區西湖里	0.041285	0.386008	0.572707	0.999997
7	新北市中和區清穗里	0.041865	0.385347	0.572788	0.999997
8	臺南市北區重興里	0.042837	0.384831	0.572331	0.999997
9	新北市板橋區景星里	0.042515	0.387607	0.569878	0.999996

Can we find the similarity of our own village?

Definitely.

In [20]:
```
def find_my_village(my_village, df):
    df = df.sort_values(['cosine_similarity', 'combined_key'], ascending=[False, True]).
reset_index(drop=True)
    my_village_df = df[df['combined_key'] == my_village]
    return my_village_df
```

In [21]:

```
my_village = '高雄市鼓山區桃源里'
my_village_df = find_my_village(my_village, votes_obtained)
my_village_similarity = my_village_df['cosine_similarity'].values[0]
my_village_rank = my_village_df.index[0]
n_rows = votes_obtained.shape[0]
print("{}的餘弦相似度為{:.4f}, 排名{}/{}".format(my_village, my_village_similarity, my_vil
lage_rank, n_rows))
my_village_df
```

高雄市鼓山區桃源里的餘弦相似度為0.9985, 排名1714/7737

Out[21]:

	combined_key	soong	han	tsai	cosine_similarity
1714	高雄市鼓山區桃源里	0.023419	0.370023	0.606557	0.998506

Feeling movivated?

Start with the most practical one: Python

- Procedural programming with Python
- Object-oriented programming with Python
- Using Python libraries

Start with the most practical one: R

- Procedural programming with R
- Functional programming with R
- Using R libraries

Resources I've used when learning Python

- Introducing Python (https://www.amazon.com/Introducing-Python-Modern-Computing-Packages/dp/1449359361)
- A Whirlwind Tour of Python (https://jakevdp.github.io/WhirlwindTourOfPython/index.html)
- Python Data Science Handbook (https://jakevdp.github.io/PythonDataScienceHandbook/)

Resources I've used when learning R

- The Art of R Programming (https://www.amazon.com/Art-Programming-Statistical-Software-Design/dp/1593273843)
- Advanced R (https://adv-r.hadley.nz/)
- R for Data Science (https://r4ds.had.co.nz/)
- Data Science Specialization (https://www.coursera.org/specializations/jhu-data-science)
- Statistics with R Specialization (https://www.coursera.org/specializations/statistics)

Learning resources from me

- 數據交點 (https://www.datainpoint.com)
- Substack (https://datainpoint.substack.com/about)

Phew, that is a lot to catch up...

You do not have to finish every course or book from end to end.

Source: https://giphy.com/ (https://giphy.com/)