Lecture15_APIs

April 17, 2024

1 Accessing data through APIs

1.1 APIs

- An API (Application Programming Interface) is an interface that sits on top of a computer based system and simplifies certain tasks, such as extracting subsets of data from a large repository or database.
- Real-time server-to-browser communication.
- An API is code that allows software programs to communicate.
- Web APIs allow you to access data available via an internet web interface.
- Often you can access data from web APIs using a URL that contains sets of parameters that specifies the type and particular subset of data that you are interested in.
- Web APIs are a way to strip away all the extraneous visual interface that you don't care about and get the data that you want.

1.2 JSON files

- Data from APIs are often JSONs (Java Script Object Notation)
- Structured Machine-readable files: Files that can be stored in a text format but are hierarchical and structured in some way that optimizes machine readability. JSON files are an example of structured machine-readable files.
- They are stored in human-readable text.
- They are 'lightweight' for storing and transferring data. This makes it very easy to work with quickly and productively. The specification is designed to minimise the number of requests and the amount of data that needs sending between client and server.
- Python is particularly good at reading in JSON files.
- It usually makes sense to load these in to Python as dictionaries.

1.3 Why do we use APIs?

Among other things, APIs allow us to: * Get information that would be time-consuming to get otherwise. * Get information that you can't get otherwise. * Automate analytical workflows that require continuously updated data. * Access data using a more direct interface. * There are many different types of web APIs. One of the most common types is a REST, or RESTful, API.

A RESTful API is a web API that uses URL arguments to specify what information you want returned through the API.

1.4 Using APIs

- API is actually a very simple tool that allows anyone to access information from a given website. You might require the use of certain headers but some APIs require just the URL.
- Data REQUEST: You try to access a URL in your browser.
- Data processing: A web server somewhere that uses URL to query a specified dataset.
- Data RESPONSE: That web server then sends you back some content.

1.5 Python libraries for accessing APIs

- As part of accessing the API content and getting the data into a .CSV file, we'll have to import a number of Python Libraries.
- requests library helps us get the content from the API by using the get() method. The json() method converts the API response to JSON format for easy handling.
- json library is needed so that we can work with the JSON content we get from the API.
- pandas library helps to create a DataFrame which we can format with proper headings and indexing, and then analyse.

2 Open Notify API example

- Now collect data from the Open Notify API. This example follows https://www.dataquest.io/blog/python-api-tutorial/.
- The Open Notify API gives access to data about the international space station. It's a great API for learning because it has a very simple design, and doesn't require authentication.
- The first endpoint we'll use is http://api.open-notify.org/astros.json, which returns data about astronauts currently in space.

Import the packages needed and request the URL:

[33]:
```
import numpy as np
import pandas as pd
import requests
import json
import os

response = requests.get("http://api.open-notify.org/astros.json")

print(response.status_code)
```

```
200
```

We received a '200' code which tells us our request was successful. The documentation tells us that the API response we'll get is in JSON format. Next, let's use the response.json() method to see the data we received back from the API:

[34]:
```
print(response.json())
```

```
{'message': 'success', 'people': [{'name': 'Jasmin Moghbeli', 'craft': 'ISS'},
{'name': 'Andreas Mogensen', 'craft': 'ISS'}, {'name': 'Satoshi Furukawa',
'craft': 'ISS'}, {'name': 'Konstantin Borisov', 'craft': 'ISS'}, {'name': 'Oleg
Kononenko', 'craft': 'ISS'}, {'name': 'Nikolai Chub', 'craft': 'ISS'}, {'name':
"Loral O'Hara", 'craft': 'ISS'}], 'number': 7}
```

2.1 Convert this to a pandas DataFrame

[35]:
```
iss_df = pd.DataFrame(response.json())

print(iss_df)
```

```
   message                                            people  number
0  success     {'name': 'Jasmin Moghbeli', 'craft': 'ISS'}       7
1  success    {'name': 'Andreas Mogensen', 'craft': 'ISS'}       7
2  success    {'name': 'Satoshi Furukawa', 'craft': 'ISS'}       7
3  success  {'name': 'Konstantin Borisov', 'craft': 'ISS'}       7
4  success      {'name': 'Oleg Kononenko', 'craft': 'ISS'}       7
5  success        {'name': 'Nikolai Chub', 'craft': 'ISS'}       7
6  success        {'name': 'Loral O'Hara', 'craft': 'ISS'}       7
```

This is not how we would like our data frame to look. The message and number columns are unnecessary. Remove them and convert the people column to a data frame.

[36]:
```
print(response.json()['people'])
```

```
[{'name': 'Jasmin Moghbeli', 'craft': 'ISS'}, {'name': 'Andreas Mogensen',
'craft': 'ISS'}, {'name': 'Satoshi Furukawa', 'craft': 'ISS'}, {'name':
'Konstantin Borisov', 'craft': 'ISS'}, {'name': 'Oleg Kononenko', 'craft':
'ISS'}, {'name': 'Nikolai Chub', 'craft': 'ISS'}, {'name': "Loral O'Hara",
'craft': 'ISS'}]
```

[37]:
```
iss_df = pd.DataFrame(response.json()['people'])

print(iss_df)
```

```
                name craft
0     Jasmin Moghbeli   ISS
1    Andreas Mogensen   ISS
2    Satoshi Furukawa   ISS
3  Konstantin Borisov   ISS
4      Oleg Kononenko   ISS
5        Nikolai Chub   ISS
6        Loral O'Hara   ISS
```

3 Colorado Population Projections Example

See what the dataset looks like at the bottom of this page: https://data.colorado.gov/Demographics/Population-Projections-in-Colorado/q5vp-adf3

At the top of the page, you will notice an API button. If you click on that, it gives you the URL for accessing the API: https://data.colorado.gov/resource/q5vp-adf3.json

We will use this URL in our code.

Import the packages needed and request the URL:

[38]:
```
import numpy as np
import pandas as pd
import requests
import json
import os

url = "https://data.colorado.gov/resource/q5vp-adf3.json"

requests.get(url)
```

[38]:
```
<Response [200]>
```

3.1 Understanding the code

- requests.get(url).json() outputs the data at the URL as a JSON to the console.
- We will save this as JSON_data: JSON_data = requests.get(url).json()
- Use json.dumps(JSON_data) to print the data to the console without indentation.
- Notice that the data looks like a Python dictionary!

[39]:
```
JSON_data = requests.get(url).json()

my_data = json.dumps(JSON_data)

# print(my_data)
```

- Recall that JSON_data = requests.get(url).json()
- To convert the JSON_data to a pandas DataFrame, simply use pd.DataFrame: pop_df = pd.DataFrame(JSON_data)

[40]:
```
pop_df = pd.DataFrame(JSON_data)

print(pop_df)
```

```
        id    county fipscode  year age malepopulation femalepopulation  \
0    60927   Douglas       35  2000  47           1495             1546
1    22233  Phillips       95  1993  28             23               26
2   228497   Prowers       99  2026   5             78               75
```

```
3      1461     Delta       29  1990   4            138              145
4    129987    Pitkin       97  2011  71             67               54
..      …       …       …    …  ..            …              …
995  141233     Kiowa       61  2013   5              9               11
996  272873  La Plata       67  2033  42            363              410
997  378638    Moffat       81  2049  60             77               89
998  124836      Yuma      125  2010  71             41               39
999  371016     Kiowa       61  2048  13              9                9

    totalpopulation  datatype
0              3041  Estimate
1                49  Estimate
2               153  Forecast
3               283  Estimate
4               122  Estimate
..              …       …
995              20  Estimate
996             773  Forecast
997             166  Forecast
998              80  Estimate
999              18  Forecast

[1000 rows x 9 columns]
```

4 RESTful APIs

- There are ways of doing more complex extractions using the API string. I will show these in the next few slides. These are RESTful APIs.
- REST (REpresentational State Transfer) is an architectural style, and an approach to communications that is often used in the development of Web services. The use of REST is often preferred over the more heavyweight SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use over the Internet
- However, unless there is a huge amount of data, we are more confident doing this filtering in Python.

4.1 Using REST API for Colorado Population Projections Example

[41]:
```
url = 'https://data.colorado.gov/resource/tv8u-hswn.json?
 ↪county=Boulder&$where=age between 20 and 40 and year between 2016 and␣
 ↪2025&$select=year,age,femalepopulation'

JSON_data = requests.get(url).json()

# print(JSON_data)
```

4.2 Create the dataset

[42]:
```
# Create the dataset

pop_df_filter = pd.DataFrame(JSON_data)

print(pop_df_filter)
```

```
     year age femalepopulation
0    2020  34             2069
1    2025  40             2053
2    2022  40             2069
3    2025  32             1987
4    2024  39             2054
..    …  ..              …
205  2023  38             2051
206  2024  40             1976
207  2016  30             2184
208  2025  36             2059
209  2018  23             2619

[210 rows x 3 columns]
```

4.3 Breaking down the API string

Notice that the colorado.data.gov API URL in the cell above starts with data.colorado.gov but then has various parameters attached to the end of the URL that specify the particular type of information that you are looking for.

The parameters in the url are: * The Data set itself: /tv8u-hswn.json * AGE: where=age between 20 and 40 * YEAR: year between 2016 and 2025 * COUNTY: county=Boulder * Columns to get: select=year,age,femalepopulation

5 Exercise

Do your own filtering of data from the Colorado population projection data using an API string.

6 Newspaper search example

- We will now look at the Chronicling America API. Details on how to use it are at the following link: https://chroniclingamerica.loc.gov/about/api/
- The base URL for the API is: https://chroniclingamerica.loc.gov/
- The Chronicling America API allows access to metadata and text for millions of scanned newspaper pages. In addition, unlike many other APIs, it also does not require an authentication process, allowing us to immediately explore the available data without signing up for an account.

- In our example, we will try to find data on when Castleblayney was mentioned in an American paper.
- From the about API page, we see that the URL for creating a request to the API is: http://chroniclingamerica.loc.gov/search/pages/results/
- We see that this contains all results.
- We want to search for Castleblayney in JSON format. To do this add ?andtext=castleblayney&format=json: https://chroniclingamerica.loc.gov/search/pages/results/?andtext=castleblayney&format=json
- If we request this URL, convert it to a JSON file, and try to convert it to a DataFrame, it is not in the format we would like.
- Notice that the items column appears to contain the dictionary we would like:

[43]:
```
url = "https://chroniclingamerica.loc.gov/search/pages/results/?
 ↪andtext=castleblayney&format=json"
#JSONContent = requests.get(url, headers={'content-type':'application/json'}).
 ↪json()

JSONContent = requests.get(url).json()


blayney_df = pd.DataFrame(JSONContent)

print(blayney_df)

print(blayney_df.iloc[0:2,0:4])
```

```
    totalItems  endIndex  startIndex  itemsPerPage  \
0           96        20           1            20
1           96        20           1            20
2           96        20           1            20
3           96        20           1            20
4           96        20           1            20
5           96        20           1            20
6           96        20           1            20
7           96        20           1            20
8           96        20           1            20
9           96        20           1            20
10          96        20           1            20
11          96        20           1            20
12          96        20           1            20
13          96        20           1            20
14          96        20           1            20
15          96        20           1            20
16          96        20           1            20
17          96        20           1            20
```

```
18          96        20           1            20
19          96        20           1            20

                                                items
0   {'sequence': 4, 'county': ['Hennepin', 'Ramsey…
1   {'sequence': 3, 'county': ['Jefferson'], 'edit…
2   {'sequence': 6, 'county': ['Hennepin', 'Ramsey…
3   {'sequence': 5, 'county': ['Honolulu'], 'editi…
4   {'sequence': 6, 'county': ['Denver', 'Salt Lak…
5   {'sequence': 7, 'county': ['Hennepin', 'Ramsey…
6   {'sequence': 6, 'county': ['Hennepin', 'Ramsey…
7   {'sequence': 6, 'county': ['Hennepin', 'Ramsey…
8   {'sequence': 4, 'county': ['Jefferson'], 'edit…
9   {'sequence': 5, 'county': ['Abbeville'], 'edit…
10  {'sequence': 3, 'county': ['Jefferson'], 'edit…
11  {'sequence': 4, 'county': ['Jefferson'], 'edit…
12  {'sequence': 4, 'county': ['Jefferson'], 'edit…
13  {'sequence': 4, 'county': ['Jefferson'], 'edit…
14  {'sequence': 3, 'county': ['Jefferson'], 'edit…
15  {'sequence': 4, 'county': ['Jefferson'], 'edit…
16  {'sequence': 2, 'county': ['Hennepin', 'Ramsey…
17  {'sequence': 2, 'county': ['Hennepin', 'Ramsey…
18  {'sequence': 7, 'county': ['Hennepin', 'Ramsey…
19  {'sequence': 6, 'county': ['Hennepin', 'Ramsey…
   totalItems  endIndex  startIndex  itemsPerPage
0          96        20           1            20
1          96        20           1            20
```

- If we apply pd.DataFrame to JSONContent[‘items’], we get the dataset we would like.
- There are some missing values and strangely formatted columns. These would have to be tidied up.
- This only gives the data for the first 20 results. How do you think we will access the data for the other results?

[44]:
```
blayney_df = pd.DataFrame(JSONContent['items'])

# blayney_df

print(blayney_df.head())

# print(blayney_df.iloc[0])
```

```
   sequence               county edition frequency  \
0         4   [Hennepin, Ramsey]    None    Weekly
1         3          [Jefferson]    None    Weekly
2         6   [Hennepin, Ramsey]    None    Weekly
3         5           [Honolulu]    None     Daily
4         6  [Denver, Salt Lake]    None    Weekly
```

```
                                      id  \
0  /lccn/sn90059959/1919-05-31/ed-1/seq-4/
1  /lccn/sn86069180/1906-06-16/ed-1/seq-3/
2  /lccn/sn90059959/1915-10-09/ed-1/seq-6/
3  /lccn/sn85047084/1900-11-10/ed-1/seq-5/
4  /lccn/sn93062856/1909-12-04/ed-1/seq-6/

                                          subject  \
0  [Hennepin County (Minn.)--Newspapers., Irish A…
1  [Irish Americans--Kentucky--Newspapers., Irish…
2  [Hennepin County (Minn.)--Newspapers., Irish A…
3  [Hawaii--Honolulu.--fast--(OCoLC)fst01204916, …
4  [Catholics--Newspapers., Catholics.--fast--(OC…

                       city      date  \
0  [Minneapolis, Saint Paul]  19190531
1              [Louisville]  19060616
2  [Minneapolis, Saint Paul]  19151009
3                [Honolulu]  19001110
4   [Denver, Salt Lake City]  19091204

                                    title  end_year  …   language  \
0               The Irish standard. [volume]      1920  …  [English]
1                   Kentucky Irish American.      1968  …  [English]
2               The Irish standard. [volume]      1920  …  [English]
3  The Pacific commercial advertiser. [volume]      1921  …  [English]
4                The Intermountain Catholic.      1920  …  [English]

                                       alt_title        lccn    country  \
0                                             []  sn90059959  Minnesota
1                                 [Irish American]  sn86069180   Kentucky
2                                             []  sn90059959  Minnesota
3  [Daily Pacific commercial advertiser, Sunday a…  sn85047084     Hawaii
4                 [Intermountain Colorado Catholic]  sn93062856       Utah

                                         ocr_eng                    batch  \
0  &\ns^v\n1\n,r\n5- ?w\n-w &**&*** .7*?fSF"«r»\n…       mnhi_castor_ver01
1  T r w\na\n1\nI 1TIIOKY 1I tiSH AMERIOr\nI\nITI…      kyu_liberace_ver01
2  1\npip\ni, t.\n7\n4\nNews\nfrom\nANTRIM.\nA £3…       mnhi_castor_ver01
3  f .\nn\nir.\nJKA\n.11\nr\ntn 1\nft.\nTHE PACIF…  hihouml_desdemona_ver03
4  c\nd ij 6 I THE INTERMOUNTAIN CATHOLIC DECEMBE…     uuml_kirilenko_ver01

                  title_normal  \
0               irish standard.
1       kentucky irish american.
2               irish standard.
3  pacific commercial advertiser.
4        intermountain catholic.
```

```
                                                  url  \
0  https://chroniclingamerica.loc.gov/lccn/sn9005…
1  https://chroniclingamerica.loc.gov/lccn/sn8606…
2  https://chroniclingamerica.loc.gov/lccn/sn9005…
3  https://chroniclingamerica.loc.gov/lccn/sn8504…
4  https://chroniclingamerica.loc.gov/lccn/sn9306…

                                              place page
0  [Minnesota--Hennepin--Minneapolis, Minnesota--…
1                  [Kentucky--Jefferson--Louisville]
2  [Minnesota--Hennepin--Minneapolis, Minnesota--…
3                       [Hawaii--Honolulu--Honolulu]    5
4  [Colorado--Denver--Denver, Utah--Salt Lake--Sa…    6

[5 rows x 28 columns]
```

[45]:
```
blayney_df.url[0]
```

[45]:
```
'https://chroniclingamerica.loc.gov/lccn/sn90059959/1919-05-31/ed-1/seq-4.json'
```

- Use the page argument to see results other than the first 20:

  https://chroniclingamerica.loc.gov/search/pages/results/?andtext=castleblayney&format=json&page=2

[46]:
```
url = "https://chroniclingamerica.loc.gov/search/pages/results/?
 ↪andtext=castleblayney&format=json&page=2"
JSONContent2 = requests.get(url).json()


blayney_df2 = pd.DataFrame(JSONContent2['items'])

print(blayney_df2)
```

```
    sequence               county edition            frequency  \
0          6   [Hennepin, Ramsey]    None               Weekly
1          6   [Hennepin, Ramsey]    None               Weekly
2          6   [Hennepin, Ramsey]    None               Weekly
3          6   [Hennepin, Ramsey]    None               Weekly
4          6   [Hennepin, Ramsey]    None               Weekly
5          6   [Hennepin, Ramsey]    None               Weekly
6          6   [Hennepin, Ramsey]    None               Weekly
7          7   [Hennepin, Ramsey]    None               Weekly
8          6   [Hennepin, Ramsey]    None               Weekly
9          6   [Hennepin, Ramsey]    None               Weekly
10         5   [Hennepin, Ramsey]    None               Weekly
11         4   [Hennepin, Ramsey]    None               Weekly
12         7   [Hennepin, Ramsey]    None               Weekly
13         4          [Jefferson]    None               Weekly
```

```
14         4          [Jefferson]    None               Weekly
15         3         [Silver Bow]    None  Daily (except Sun.)
16         6  [Denver, Salt Lake]    None               Weekly
17         6  [Denver, Salt Lake]    None               Weekly
18         7  [Denver, Salt Lake]    None               Weekly
19         3            [Orleans]    None               Weekly
```

```
                                        id  \
0   /lccn/sn90059959/1906-05-26/ed-1/seq-6/
1   /lccn/sn90059959/1914-10-31/ed-1/seq-6/
2   /lccn/sn90059959/1914-12-26/ed-1/seq-6/
3   /lccn/sn90059959/1917-08-25/ed-1/seq-6/
4   /lccn/sn90059959/1917-10-27/ed-1/seq-6/
5   /lccn/sn90059959/1919-02-08/ed-1/seq-6/
6   /lccn/sn90059959/1918-03-30/ed-1/seq-6/
7   /lccn/sn90059959/1919-03-08/ed-1/seq-7/
8   /lccn/sn90059959/1919-03-22/ed-1/seq-6/
9   /lccn/sn90059959/1919-04-19/ed-1/seq-6/
10  /lccn/sn90059959/1919-05-03/ed-1/seq-5/
11  /lccn/sn90059959/1919-10-11/ed-1/seq-4/
12  /lccn/sn90059959/1887-01-01/ed-1/seq-7/
13  /lccn/sn86069180/1913-09-13/ed-1/seq-4/
14  /lccn/sn86069180/1914-03-28/ed-1/seq-4/
15  /lccn/sn83045085/1919-06-30/ed-1/seq-3/
16  /lccn/sn93062856/1908-06-13/ed-1/seq-6/
17  /lccn/sn93062856/1910-12-17/ed-1/seq-6/
18  /lccn/sn93062856/1908-10-24/ed-1/seq-7/
19  /lccn/sn86086284/1868-04-05/ed-1/seq-3/
```

```
                                           subject  \
0   [Hennepin County (Minn.)--Newspapers., Irish A…
1   [Hennepin County (Minn.)--Newspapers., Irish A…
2   [Hennepin County (Minn.)--Newspapers., Irish A…
3   [Hennepin County (Minn.)--Newspapers., Irish A…
4   [Hennepin County (Minn.)--Newspapers., Irish A…
5   [Hennepin County (Minn.)--Newspapers., Irish A…
6   [Hennepin County (Minn.)--Newspapers., Irish A…
7   [Hennepin County (Minn.)--Newspapers., Irish A…
8   [Hennepin County (Minn.)--Newspapers., Irish A…
9   [Hennepin County (Minn.)--Newspapers., Irish A…
10  [Hennepin County (Minn.)--Newspapers., Irish A…
11  [Hennepin County (Minn.)--Newspapers., Irish A…
12  [Hennepin County (Minn.)--Newspapers., Irish A…
13  [Irish Americans--Kentucky--Newspapers., Irish…
14  [Irish Americans--Kentucky--Newspapers., Irish…
15  [Labor--Montana--Newspapers., Labor.--fast--(O…
16  [Catholics--Newspapers., Catholics.--fast--(OC…
17  [Catholics--Newspapers., Catholics.--fast--(OC…
```

```
18  [Catholics--Newspapers., Catholics.--fast--(OC…
19  [Catholics--Louisiana--New Orleans--Newspapers…

                       city      date  \
0   [Minneapolis, Saint Paul]  19060526
1   [Minneapolis, Saint Paul]  19141031
2   [Minneapolis, Saint Paul]  19141226
3   [Minneapolis, Saint Paul]  19170825
4   [Minneapolis, Saint Paul]  19171027
5   [Minneapolis, Saint Paul]  19190208
6   [Minneapolis, Saint Paul]  19180330
7   [Minneapolis, Saint Paul]  19190308
8   [Minneapolis, Saint Paul]  19190322
9   [Minneapolis, Saint Paul]  19190419
10  [Minneapolis, Saint Paul]  19190503
11  [Minneapolis, Saint Paul]  19191011
12  [Minneapolis, Saint Paul]  18870101
13              [Louisville]  19130913
14              [Louisville]  19140328
15                   [Butte]  19190630
16   [Denver, Salt Lake City]  19080613
17   [Denver, Salt Lake City]  19101217
18   [Denver, Salt Lake City]  19081024
19             [New Orleans]  18680405

                                          title  end_year  …  \
0                      The Irish standard. [volume]      1920  …
1                      The Irish standard. [volume]      1920  …
2                      The Irish standard. [volume]      1920  …
3                      The Irish standard. [volume]      1920  …
4                      The Irish standard. [volume]      1920  …
5                      The Irish standard. [volume]      1920  …
6                      The Irish standard. [volume]      1920  …
7                      The Irish standard. [volume]      1920  …
8                      The Irish standard. [volume]      1920  …
9                      The Irish standard. [volume]      1920  …
10                     The Irish standard. [volume]      1920  …
11                     The Irish standard. [volume]      1920  …
12                     The Irish standard. [volume]      1920  …
13                         Kentucky Irish American.      1968  …
14                         Kentucky Irish American.      1968  …
15                The Butte daily bulletin. [volume]      1921  …
16                       The Intermountain Catholic.      1920  …
17                       The Intermountain Catholic.      1920  …
18                       The Intermountain Catholic.      1920  …
19  The morning star and Catholic messenger. [volume]      1881  …

     language                           alt_title        lccn    country  \
```

0	[English]	[]	sn90059959	Minnesota	
1	[English]	[]	sn90059959	Minnesota	
2	[English]	[]	sn90059959	Minnesota	
3	[English]	[]	sn90059959	Minnesota	
4	[English]	[]	sn90059959	Minnesota	
5	[English]	[]	sn90059959	Minnesota	
6	[English]	[]	sn90059959	Minnesota	
7	[English]	[]	sn90059959	Minnesota	
8	[English]	[]	sn90059959	Minnesota	
9	[English]	[]	sn90059959	Minnesota	
10	[English]	[]	sn90059959	Minnesota	
11	[English]	[]	sn90059959	Minnesota	
12	[English]	[]	sn90059959	Minnesota	
13	[English]	[Irish American]	sn86069180	Kentucky	
14	[English]	[Irish American]	sn86069180	Kentucky	
15	[English]	[]	sn83045085	Montana	
16	[English]	[Intermountain Colorado Catholic]	sn93062856	Utah	
17	[English]	[Intermountain Colorado Catholic]	sn93062856	Utah	
18	[English]	[Intermountain Colorado Catholic]	sn93062856	Utah	
19	[English]	[Catholic messenger, Morning star]	sn86086284	Louisiana	

	ocr_eng	batch	\
0	Sf\nSKy\nI\nI\nfr\n[4\nSs\nafc&t*\nNEWS FROM I…	mnhi_betelgeuse_ver01	
1	Antrim.\nJ. V. Brennan, Belfast, is the archi\…	mnhi_castor_ver01	
2	•H Antrim.\nThe late John Rogers, Belfast,\nCh…	mnhi_castor_ver01	
3	J8f\nSSL\n%v"\n3t\n1\nMS\nW-\n«ri-\n6\n11\nf^l…	mnhi_castor_ver01	
4	i\y\niV\n•M\n4\nUj\n,!\nMl\nI\nII\nti\nv-i\nSI…	mnhi_castor_ver01	
5	fesiSS\n£$ &•<\nI\nt-\n$••\nNews\nS\nARMAGH.\n…	mnhi_castor_ver01	
6	6\nNews\nCONNAUGHT.\nSubject to the defendants…	mnhi_castor_ver01	
7	,:%T\\nv-'\n'V'\nert. r:\nrfc\nSSiil\nll,'t»\n…	mnhi_castor_ver01	
8	1MM\nSlit-:'\nfa" Bench In an application on b…	mnhi_castor_ver01	
9	i'"V.\n®5£®ra?35fc I I\n^$Sa|\n^tv-. •..-.\nW …	mnhi_castor_ver01	
10	•j,. .V.^j\n1 1\n0\ny.fcfTj--'-. V'\n"^1\n,v\n…	mnhi_castor_ver01	
11	-•.\niK\nIvv\nI\n,,\nI?\nI\n/iv\nIm\nIS1\nNews…	mnhi_castor_ver01	
12	,1'\npi\na\nSHAMROCKS FROM ERIN.\nWritten for …	mnhi_antares_ver03	
13	W.Ni UCKY IRIII iVIVIIKICVIV.\nDo You Jfealize…	kyu_albatross_ver01	
14	m"z\nWhat the Club Plan Is\nAnd What It Means …	kyu_albatross_ver01	
15	The Great Drought\n(By United Press.)\nWashing…	mthi_lynx_ver01	
16	L\nI 6 t INTERMOUNTAIN AND COLORADO CATHOLIC J…	uuml_kirilenko_ver01	
17	6 h THE INTERMOUNTAIN CATHOLIC DEC 17 1910\n+ …	uuml_kirilenko_ver01	
18	< J1 tt\nr\nI j 6 THE IHTERMOUNTAiH Arm COLORA…	uuml_kirilenko_ver01	
19	mroammmeo stan aw tazawcw summmwas\n]xw orrLAr…	lu_Cypher_ver01	

	title_normal	\
0	irish standard.	
1	irish standard.	
2	irish standard.	
3	irish standard.	

```
4                        irish standard.
5                        irish standard.
6                        irish standard.
7                        irish standard.
8                        irish standard.
9                        irish standard.
10                       irish standard.
11                       irish standard.
12                       irish standard.
13              kentucky irish american.
14              kentucky irish american.
15                 butte daily bulletin.
16               intermountain catholic.
17               intermountain catholic.
18               intermountain catholic.
19  morning star and catholic messenger.

                                                    url  \
0   https://chroniclingamerica.loc.gov/lccn/sn9005…
1   https://chroniclingamerica.loc.gov/lccn/sn9005…
2   https://chroniclingamerica.loc.gov/lccn/sn9005…
3   https://chroniclingamerica.loc.gov/lccn/sn9005…
4   https://chroniclingamerica.loc.gov/lccn/sn9005…
5   https://chroniclingamerica.loc.gov/lccn/sn9005…
6   https://chroniclingamerica.loc.gov/lccn/sn9005…
7   https://chroniclingamerica.loc.gov/lccn/sn9005…
8   https://chroniclingamerica.loc.gov/lccn/sn9005…
9   https://chroniclingamerica.loc.gov/lccn/sn9005…
10  https://chroniclingamerica.loc.gov/lccn/sn9005…
11  https://chroniclingamerica.loc.gov/lccn/sn9005…
12  https://chroniclingamerica.loc.gov/lccn/sn9005…
13  https://chroniclingamerica.loc.gov/lccn/sn8606…
14  https://chroniclingamerica.loc.gov/lccn/sn8606…
15  https://chroniclingamerica.loc.gov/lccn/sn8304…
16  https://chroniclingamerica.loc.gov/lccn/sn9306…
17  https://chroniclingamerica.loc.gov/lccn/sn9306…
18  https://chroniclingamerica.loc.gov/lccn/sn9306…
19  https://chroniclingamerica.loc.gov/lccn/sn8608…

                                                  place page
0   [Minnesota--Hennepin--Minneapolis, Minnesota--…
1   [Minnesota--Hennepin--Minneapolis, Minnesota--…
2   [Minnesota--Hennepin--Minneapolis, Minnesota--…
3   [Minnesota--Hennepin--Minneapolis, Minnesota--…
4   [Minnesota--Hennepin--Minneapolis, Minnesota--…
5   [Minnesota--Hennepin--Minneapolis, Minnesota--…
6   [Minnesota--Hennepin--Minneapolis, Minnesota--…
7   [Minnesota--Hennepin--Minneapolis, Minnesota--…
```

```
8   [Minnesota--Hennepin--Minneapolis, Minnesota--…
9   [Minnesota--Hennepin--Minneapolis, Minnesota--…
10  [Minnesota--Hennepin--Minneapolis, Minnesota--…
11  [Minnesota--Hennepin--Minneapolis, Minnesota--…
12  [Minnesota--Hennepin--Minneapolis, Minnesota--…
13                 [Kentucky--Jefferson--Louisville]
14                 [Kentucky--Jefferson--Louisville]
15                      [Montana--Silver Bow--Butte]    3
16  [Colorado--Denver--Denver, Utah--Salt Lake--Sa…    6
17  [Colorado--Denver--Denver, Utah--Salt Lake--Sa…    6
18  [Colorado--Denver--Denver, Utah--Salt Lake--Sa…    6
19                 [Louisiana--Orleans--New Orleans]    3

[20 rows x 28 columns]
```

6.1 Putting all of the pages together into one data frame

- We will use a for loop to loop through the page numbers, and add the data to a DataFrame each time.
- First, we must create an empty DataFrame: blayney_df_all = pd.DataFrame()
- We will specify a range in our for loop of range(1, 6) because we want to go from pages 1 to 5 inclusive.
- We will add the number on as a string at the end of the URL to specify the page number.

[47]:
```
blayney_df_all = pd.DataFrame()

for i in range(1,6):
    JSONContent = requests.get("https://chroniclingamerica.loc.gov/search/pages/
 ↪results/?andtext=castleblayney&format=json&page="+str(i),␣
 ↪headers={'content-type':'application/json'}).json()
    blayney_df_page = pd.DataFrame(JSONContent['items'])
    blayney_df_all = pd.concat([blayney_df_all, blayney_df_page])

blayney_df_all
```

[47]:
```
    sequence              county edition frequency  \
0          4   [Hennepin, Ramsey]    None    Weekly
1          3          [Jefferson]    None    Weekly
2          6   [Hennepin, Ramsey]    None    Weekly
3          5           [Honolulu]    None     Daily
4          6  [Denver, Salt Lake]    None    Weekly
..       …                    …     …       …
11         9           [New York]    None     Daily
12         3               [None]    None     Daily
13         3          [Jefferson]    None    Weekly
14         4         [Shiawassee]    None    Weekly
```

```
15         2                [Lake]     None    Weekly

                                            id  \
0     /lccn/sn90059959/1919-05-31/ed-1/seq-4/
1     /lccn/sn86069180/1906-06-16/ed-1/seq-3/
2     /lccn/sn90059959/1915-10-09/ed-1/seq-6/
3     /lccn/sn85047084/1900-11-10/ed-1/seq-5/
4     /lccn/sn93062856/1909-12-04/ed-1/seq-6/
..                                          …
11    /lccn/sn83030313/1872-05-25/ed-1/seq-9/
12    /lccn/sn83045462/1886-08-05/ed-1/seq-3/
13    /lccn/sn86069180/1907-08-10/ed-1/seq-3/
14  /lccn/sn2003060105/1881-11-11/ed-1/seq-4/
15    /lccn/sn85026421/1881-11-10/ed-1/seq-2/

                                               subject  \
0   [Hennepin County (Minn.)--Newspapers., Irish A…
1   [Irish Americans--Kentucky--Newspapers., Irish…
2   [Hennepin County (Minn.)--Newspapers., Irish A…
3   [Hawaii--Honolulu.--fast--(OCoLC)fst01204916, …
4   [Catholics--Newspapers., Catholics.--fast--(OC…
..                                                …
11  [New York (N.Y.)--Newspapers., New York (State…
12  [Washington (D.C.)--fast--(OCoLC)fst01204505, …
13  [Irish Americans--Kentucky--Newspapers., Irish…
14  [Michigan--Owosso.--fast--(OCoLC)fst01231829, …
15  [Baldwin (Mich.)--Newspapers., Chase (Mich.)--…

                         city      date  \
0   [Minneapolis, Saint Paul]  19190531
1                [Louisville]  19060616
2   [Minneapolis, Saint Paul]  19151009
3                  [Honolulu]  19001110
4    [Denver, Salt Lake City]  19091204
..                        …       …
11                 [New York]  18720525
12               [Washington]  18860805
13               [Louisville]  19070810
14                   [Owosso]  18811111
15                  [Baldwin]  18811110

                                        title  end_year  …   language  \
0                 The Irish standard. [volume]      1920  …  [English]
1                     Kentucky Irish American.      1968  …  [English]
2                 The Irish standard. [volume]      1920  …  [English]
3   The Pacific commercial advertiser. [volume]      1921  …  [English]
4                  The Intermountain Catholic.      1920  …  [English]
```

```
..                                        …       …  …        …
11               The New York herald. [volume]      1920  …  [English]
12                     Evening star. [volume]      1972  …  [English]
13                  Kentucky Irish American.       1968  …  [English]
14                        The Owosso times.       1882  …  [English]
15               The Lake County star. [volume]      9999  …  [English]

                                          alt_title          lccn  \
0                                                []    sn90059959
1                                  [Irish American]    sn86069180
2                                                []    sn90059959
3   [Daily Pacific commercial advertiser, Sunday a…    sn85047084
4                   [Intermountain Colorado Catholic]    sn93062856
..                                              …           …
11                           [New York daily herald]    sn83030313
12                               [Star, Sunday star]    sn83045462
13                                  [Irish American]    sn86069180
14                                               []  sn2003060105
15                                               []    sn85026421

                country                                                ocr_eng  \
0             Minnesota  &\ns^v\n1\n,r\n5- ?w\n-w &**&*** .7*?fSF"«r»\n…
1              Kentucky  T r w\na\n1\nI 1TIIOKY 1I tiSH AMERIOr\nI\nITI…
2             Minnesota  1\npip\ni, t.\n7\n4\nNews\nfrom\nANTRIM.\nA £3…
3                Hawaii  f .\nn\nir.\nJKA\n.11\nr\ntn 1\nft.\nTHE PACIF…
4                  Utah  c\nd ij 6 I THE INTERMOUNTAIN CATHOLIC DECEMBE…
..                  …                                                   …
11            New York  FINANCIAL AND COMMERCIAL.\nGOLD DULL.\nA Vesuv…
12  District of Columbia  FOR SALE?HOUSES.\nDMtSAlX-.NK E 7 ROOM BRICK; …
13             Kentucky  I < NTUOKY RISH Y RIO\nc < iJI I\nJ BE SURE TO…
14             Michigan  XEWS OF THE H LtoK.\nMICHIGAN.\nMarshall P. Wi…
15             Michigan  LAKE MY STAR.\nC. K. BlDCUm, PsUlsaer,\nBALDWI…

                     batch                   title_normal  \
0         mnhi_castor_ver01                 irish standard.
1        kyu_liberace_ver01        kentucky irish american.
2         mnhi_castor_ver01                 irish standard.
3   hihouml_desdemona_ver03  pacific commercial advertiser.
4      uuml_kirilenko_ver01         intermountain catholic.
..                      …                              …
11      dlc_houseleek_ver01                new york herald.
12   dlc_newfoundland_ver04                   evening star.
13       kyu_liberace_ver01        kentucky irish american.
14    mimtptc_detroit_ver02                   owosso times.
15     mimtptc_beulah_ver01                lake county star.

                                            url  \
```

```
0   https://chroniclingamerica.loc.gov/lccn/sn9005…
1   https://chroniclingamerica.loc.gov/lccn/sn8606…
2   https://chroniclingamerica.loc.gov/lccn/sn9005…
3   https://chroniclingamerica.loc.gov/lccn/sn8504…
4   https://chroniclingamerica.loc.gov/lccn/sn9306…
..                                                …
11  https://chroniclingamerica.loc.gov/lccn/sn8303…
12  https://chroniclingamerica.loc.gov/lccn/sn8304…
13  https://chroniclingamerica.loc.gov/lccn/sn8606…
14  https://chroniclingamerica.loc.gov/lccn/sn2003…
15  https://chroniclingamerica.loc.gov/lccn/sn8502…

                                                 place page
0   [Minnesota--Hennepin--Minneapolis, Minnesota--…
1                    [Kentucky--Jefferson--Louisville]
2   [Minnesota--Hennepin--Minneapolis, Minnesota--…
3                         [Hawaii--Honolulu--Honolulu]    5
4   [Colorado--Denver--Denver, Utah--Salt Lake--Sa…    6
..                                                  …  …
11                      [New York--New York--New York]    9
12                  [District of Columbia--Washington]
13                   [Kentucky--Jefferson--Louisville]
14                      [Michigan--Shiawassee--Owosso]
15                           [Michigan--Lake--Baldwin]

[96 rows x 28 columns]
```

7 Accessing an API with authorisation: YouTube

https://developers.google.com/youtube/v3

https://developers.google.com/youtube/v3/getting-started

Create a project at the credentials page (link on this page: https://developers.google.com/youtube/registering_an_application)

Click on Create Credentials and request an API key. My key is AIzaSyDokWbkaupuwHNDWs9i6wQFEzuc-gJ4mv8

Enable the YouTube Data API v3.

Install the googleapiclient (I have commented it out to avoid installing multiple times):

[48]:
```
# pip install --upgrade google-api-python-client
```

[49]:
```
#import necessary libraries
from googleapiclient.discovery import build
import pandas as pd
import seaborn as sns
```

[50]:
```
api_key = 'AIzaSyDokWbkaupuwHNDWs9i6wQFEzuc-gJ4mv8'

youtube  = build('youtube' , 'v3' , developerKey = api_key)
```

[ ]:
```

```

We will need to find the channel_ids of the channels we want to find details on.

See details here on how to find channel ids: https://www.youtube.com/watch?v=qPKmPaNaCmE

1. Go to the YouTube channel homepage.
2. Right-click anywhere on the page and click 'View page source'.
3. Use Ctrl-F to find 'channel_id=' on the page. The channel_id will appear after this text.
4. Copy the channel_id into the Python code as seen below.

[51]:
```
channel_ids = ['UCNAf1k0yIjyGu3k9BwAg3lg', # Sky Sports Premier League
               'UCWw6scNyopJ0yjMu1SyOEyw', # Talksport
               'UCjXIw1GlwaY1IzpW_jN9iCQ', # The Overlap
              ]
```

Let's run some code to see what data we can get on the channels.

[52]:
```
request = youtube.channels().list(
                part = "snippet,contentDetails,statistics",
                id=','.join(channel_ids))
response = request.execute()

response
```

[52]:
```
{'kind': 'youtube#channelListResponse',
 'etag': 'BxNXj3kZTZa4br_CT78dd4TD7hs',
 'pageInfo': {'totalResults': 3, 'resultsPerPage': 5},
 'items': [{'kind': 'youtube#channel',
   'etag': 'IUuWW1k69L7t5GofOrK53N6VN5Q',
   'id': 'UCjXIw1GlwaY1IzpW_jN9iCQ',
   'snippet': {'title': 'The Overlap',
    'description': 'Welcome to The Overlap, the multi award winning channel,
with some of the biggest names in sport.\n\nJoin us for Fan Debates, Exclusive
Interviews, and Stick to Football  with Gary Neville, Roy Keane, Jamie
Carragher, Jill Scott, and Ian Wright. Subscribe now so you never miss a show.
\n',
    'customUrl': '@theoverlap',
    'publishedAt': '2021-04-19T09:36:23.371102Z',
    'thumbnails': {'default': {'url': 'https://yt3.ggpht.com/uOn9g5DLZQYlQ8qKwEs
vv9KN_dUStV6YCv_D82ILcob1gwJt8YVFO5bUJC27vyOUloopV0g7=s88-c-k-c0x00ffffff-no-
rj',
      'width': 88,
      'height': 88},
     'medium': {'url': 'https://yt3.ggpht.com/uOn9g5DLZQYlQ8qKwEsvv9KN_dUStV6YCv
```

```
_D82ILcob1gwJt8YVFO5bUJC27vyOUloopV0g7=s240-c-k-c0x00ffffff-no-rj',
      'width': 240,
      'height': 240},
     'high': {'url': 'https://yt3.ggpht.com/uOn9g5DLZQYlQ8qKwEsvv9KN_dUStV6YCv_D
82ILcob1gwJt8YVFO5bUJC27vyOUloopV0g7=s800-c-k-c0x00ffffff-no-rj',
      'width': 800,
      'height': 800}},
    'localized': {'title': 'The Overlap',
     'description': 'Welcome to The Overlap, the multi award winning channel,
with some of the biggest names in sport.\n\nJoin us for Fan Debates, Exclusive
Interviews, and Stick to Football  with Gary Neville, Roy Keane, Jamie
Carragher, Jill Scott, and Ian Wright. Subscribe now so you never miss a show.
\n'},
    'country': 'GB'},
   'contentDetails': {'relatedPlaylists': {'likes': '',
     'uploads': 'UUjXIw1GlwaY1IzpW_jN9iCQ'}},
   'statistics': {'viewCount': '276092864',
    'subscriberCount': '1030000',
    'hiddenSubscriberCount': False,
    'videoCount': '548'}},
  {'kind': 'youtube#channel',
   'etag': '7dpXaqyYB_D41OARsRzytCD3Wlo',
   'id': 'UCNAf1k0yIjyGu3k9BwAg3lg',
   'snippet': {'title': 'Sky Sports Premier League',
    'description': "Sky Sports Premier League is the home of Sky Sports' Premier
League videos on YouTube featuring highlights from every game of the season, as
well as post match interviews, exclusive player access and top level
analysis!\n\nMake sure you subscribe and turn on notifications so you don't miss
a single upload!",
    'customUrl': '@skysportspremierleague',
    'publishedAt': '2015-07-06T11:11:54Z',
    'thumbnails': {'default': {'url': 'https://yt3.ggpht.com/so_lrPh6XZG8QdMd5wM
OFhLWAM8SfudqKoJhIe0vKUyk3LsQtNdCPvDtDtLSoTnZ4ZnrVLVm6Q=s88-c-k-c0x00ffffff-no-
rj',
      'width': 88,
      'height': 88},
     'medium': {'url': 'https://yt3.ggpht.com/so_lrPh6XZG8QdMd5wMOFhLWAM8SfudqKo
JhIe0vKUyk3LsQtNdCPvDtDtLSoTnZ4ZnrVLVm6Q=s240-c-k-c0x00ffffff-no-rj',
      'width': 240,
      'height': 240},
     'high': {'url': 'https://yt3.ggpht.com/so_lrPh6XZG8QdMd5wMOFhLWAM8SfudqKoJh
Ie0vKUyk3LsQtNdCPvDtDtLSoTnZ4ZnrVLVm6Q=s800-c-k-c0x00ffffff-no-rj',
      'width': 800,
      'height': 800}},
    'localized': {'title': 'Sky Sports Premier League',
     'description': "Sky Sports Premier League is the home of Sky Sports'
Premier League videos on YouTube featuring highlights from every game of the
```

season, as well as post match interviews, exclusive player access and top level analysis!\n\nMake sure you subscribe and turn on notifications so you don't miss a single upload!"},
    'country': 'GB'},
   'contentDetails': {'relatedPlaylists': {'likes': '',
     'uploads': 'UUNAf1k0yIjyGu3k9BwAg3lg'}},
   'statistics': {'viewCount': '4043576395',
    'subscriberCount': '4920000',
    'hiddenSubscriberCount': False,
    'videoCount': '10946'}},
  {'kind': 'youtube#channel',
   'etag': 'kkX9hS2On5lNelzwK29vmVwXqFU',
   'id': 'UCWw6scNyopJ0yjMu1SyOEyw',
   'snippet': {'title': 'talkSPORT',
    'description': "talkSPORT is the world's biggest sports radio station. The talkSPORT network, incorporating talkSPORT and sister station talkSPORT 2, has more exclusive English football than any other UK broadcaster. talkSPORT is available on 1089/1053AM, on digital radio, on mobile, through the talkSPORT App, via your TV, and online at talkSPORT.com. talkSPORT 2 is available on digital radio, on mobile, through the talkSPORT App and online at talkSPORT.com. \n\nSubscribe here: https://youtube.com/c/talkSPORT\n talkSPORT's Website: /https://talksport.com/\n talkSPORT's Twitter: https://twitter.com/talkSPORT\n talkSPORT's Instagram: https://www.instagram.com/talksport/?hl=en\n talkSPORT's Facebook: https://www.facebook.com/talkSPORT/\n talkSPORT's Tik Tok: https://www.tiktok.com/@talksport?\n",
    'customUrl': '@talksport',
    'publishedAt': '2009-05-01T11:38:01Z',
    'thumbnails': {'default': {'url': 'https://yt3.ggpht.com/brUeyy-sv-UNAGeBSdTNf-TvPEUSdwedFPucTRDx33AJXx_NPFAvqg0SyefautLq9FUzYW8b=s88-c-k-c0x00ffffff-no-rj',
      'width': 88,
      'height': 88},
     'medium': {'url': 'https://yt3.ggpht.com/brUeyy-sv-UNAGeBSdTNf-TvPEUSdwedFPucTRDx33AJXx_NPFAvqg0SyefautLq9FUzYW8b=s240-c-k-c0x00ffffff-no-rj',
      'width': 240,
      'height': 240},
     'high': {'url': 'https://yt3.ggpht.com/brUeyy-sv-UNAGeBSdTNf-TvPEUSdwedFPucTRDx33AJXx_NPFAvqg0SyefautLq9FUzYW8b=s800-c-k-c0x00ffffff-no-rj',
      'width': 800,
      'height': 800}},
    'localized': {'title': 'talkSPORT',
     'description': "talkSPORT is the world's biggest sports radio station. The talkSPORT network, incorporating talkSPORT and sister station talkSPORT 2, has more exclusive English football than any other UK broadcaster. talkSPORT is available on 1089/1053AM, on digital radio, on mobile, through the talkSPORT App, via your TV, and online at talkSPORT.com. talkSPORT 2 is available on digital radio, on mobile, through the talkSPORT App and online at talkSPORT.com.

```
\n\nSubscribe here: https://youtube.com/c/talkSPORT\n talkSPORT's Website:
/https://talksport.com/\n talkSPORT's Twitter: https://twitter.com/talkSPORT\n
talkSPORT's Instagram: https://www.instagram.com/talksport/?hl=en\n talkSPORT's
Facebook: https://www.facebook.com/talkSPORT/\n talkSPORT's Tik Tok:
https://www.tiktok.com/@talksport?\n"},
    'country': 'GB'},
   'contentDetails': {'relatedPlaylists': {'likes': '',
     'uploads': 'UUWw6scNyopJ0yjMu1SyOEyw'}},
   'statistics': {'viewCount': '1095475777',
    'subscriberCount': '1470000',
    'hiddenSubscriberCount': False,
    'videoCount': '18828'}}]}
```

Define a function called channel_stats to get channel statistics:

[53]:
```
def channel_stats(youtube, channel_ids):
    all_data = []
    request = youtube.channels().list(
                part = "snippet,contentDetails,statistics",
                id=','.join(channel_ids))
    response = request.execute()

    for i in range(len(response['items'])):
        data = {
            'channel_name': response['items'][i]['snippet']['title'],
            'num_Subscribers':␣
 ↪response['items'][i]['statistics']['subscriberCount'],
            'num_views': response['items'][i]['statistics']['viewCount'],
            'num_vids': response['items'][i]['statistics']['videoCount'],
            'playlist_ID':
 ↪response['items'][i]['contentDetails']['relatedPlaylists']['uploads']
    }
        all_data.append(data)

    return all_data
```

[54]:
```
channel_stats(youtube, channel_ids)
```

[54]:
```
[{'channel_name': 'Sky Sports Premier League',
  'num_Subscribers': '4920000',
  'num_views': '4043576395',
  'num_vids': '10946',
  'playlist_ID': 'UUNAf1k0yIjyGu3k9BwAg3lg'},
 {'channel_name': 'The Overlap',
  'num_Subscribers': '1030000',
  'num_views': '276092864',
  'num_vids': '548',
```

```
 'playlist_ID': 'UUjXIw1GlwaY1IzpW_jN9iCQ'},
{'channel_name': 'talkSPORT',
 'num_Subscribers': '1470000',
 'num_views': '1095475777',
 'num_vids': '18828',
 'playlist_ID': 'UUWw6scNyopJ0yjMu1SyOEyw'}]
```

[55]:
```
channel =  channel_stats(youtube, channel_ids)
```

[56]:
```
channelStats = pd.DataFrame(channel) #convert to pandas df
```

[57]:
```
channelStats
```

[57]:
```
                channel_name num_Subscribers   num_views num_vids  \
0                  talkSPORT         1470000  1095475777    18828
1                The Overlap         1030000   276092864      548
2  Sky Sports Premier League         4920000  4043576395    10946

                playlist_ID
0  UUWw6scNyopJ0yjMu1SyOEyw
1  UUjXIw1GlwaY1IzpW_jN9iCQ
2  UUNAf1k0yIjyGu3k9BwAg3lg
```

Now we will try to get details on videos from the playlist_id from the channel 'The Overlap'. First, we extract the playlist_id from the channelStats df:

[58]:
```
channel_name = 'The Overlap'
playlist_id = channelStats.loc[channelStats['channel_name'] == channel_name,␣
 ↪'playlist_ID'].iloc[0]
playlist_id
```

[58]:
```
'UUjXIw1GlwaY1IzpW_jN9iCQ'
```

Next, create a function to extract 50 video IDs from the playlist.

[59]:
```
def get_vid_id(youtube, playlist_id):

    request =  youtube.playlistItems().list(  #playlistItem() is gotten from␣
 ↪the YT developer to list playlist items.
                part = "contentDetails",
                playlistId = playlist_id,
                maxResults = 50 ) # to increase the results per page from the␣
 ↪default 5 to the max 50
    response = request.execute()

    video_ids = []
    for i in range(len(response['items'])):
        video_ids.append(response['items'][i]['contentDetails']['videoId'])
```

```
    return video_ids
```

Run this function on our playlist_id:

[60]:
```
video_ids = get_vid_id(youtube, playlist_id)
video_ids
```

[60]:
```
['pQmGqnjm8m0',
 'lc7zKpWQ-lY',
 '1dh1qU1ReeU',
 'git7lQ3tIsg',
 'YkFLno4fPAA',
 'Qui9YgzETFg',
 'HCYdjZzCBEw',
 'rucwiWiwKtw',
 '7bk_8vem1ro',
 'AhnM2GInRIM',
 'sRRU1VZYu_g',
 'NPSjXLBM-5g',
 '49wGzUm1tTU',
 'uG6XIp9F7HI',
 'pd_nzfN9XGg',
 'vPhXVPKpwSw',
 '_aW3vyraA5o',
 'JEyGTIQgHZ4',
 '2a7F3Vdlx9s',
 'lQ3scPF2MrE',
 'yIGQwr2m7bE',
 'IagNyGquMkA',
 'h3OnfcAvAgk',
 'hUGhvF-5QLI',
 'Sl1-IKjO0wI',
 '1qPeTzJd8K4',
 'uCUoAgncax0',
 '7pxk3WpU_YQ',
 'UeYMjJwdaiY',
 '_e7vOIiWYO8',
 'SgHI-9cO4XA',
 '3Yz1X2YqqJs',
 'TucTEtYA9R0',
 '3_kUsOagzLk',
 'm_7vaqFmPCY',
 'pIcKhKtSSpk',
 'p9UvWw3mxM4',
 'c2OK4n-ESYo',
 '7OisAoFjwJ8',
```

```
 'VQ8r9GD5dxI',
 '1TRU8sky0qo',
 '6t9nl9Ozct4',
 'pOa7AZO7Cwk',
 '72tzySlXHJo',
 '76vKLMWSFxA',
 '-ci6q7I-wQc',
 'mGt1bo3t8rY',
 'y68vgr_rVrg',
 'r5u4_m4Rk9c',
 'vlRZ2db8GIw']
```

Now we need to get the video details from the video IDs:

[61]:
```
def get_vid_details(youtube, video_ids):
    combvidsstats = []

    for video_id in video_ids:
        request = youtube.videos().list(
            part='snippet,statistics',
            id=video_id
        )
        response = request.execute()

        for video in response['items']:
            video_stats = {
                'Title': video['snippet']['title'],
                'Publish_date': video['snippet']['publishedAt'],
                'num_views': video['statistics']['viewCount'],
                'num_likes': video['statistics']['likeCount'],
                'num_comm': video['statistics']['commentCount']
            }
            combvidsstats.append(video_stats)

    return combvidsstats
```

[62]:
```
video_details = get_vid_details(youtube, video_ids)
```

Convert to a data frame.

[63]:
```
video_details_df = pd.DataFrame(video_details)
```

[64]:
```
video_details_df
```

[64]:

	Title	Publish_date	\
0	Sadio Mane Cristiano Ronaldo	2024-04-16T16:00:40Z	
1	Out Now: Trent Alexander-Arnold: "I owe everyt…	2024-04-16T07:00:15Z	
2	Trent Alexander-Arnold: "I Owe Everything To K…	2024-04-16T07:00:01Z	

```
3          "He's above twitter…He's on LinkedIn!"    2024-04-15T07:00:18Z
4   Serge Pizzorno on Kasabian Kickstarting Leices…  2024-04-14T23:00:19Z
5   "If you don't have a quarterback you don't hav…  2024-04-14T16:00:18Z
6   Is the NFL set up better than the Premier Leag…  2024-04-14T07:00:07Z
7                           The Breakdown: Cole Palmer  2024-04-13T11:00:24Z
8      "He's raring to go at everybody, I love it!"    2024-04-13T07:00:04Z
9                  Stick to Football Parachute Jump?    2024-04-12T16:00:06Z
10  Why Cole Palmer Is One Of The Best Players In …  2024-04-12T07:00:22Z
11  Could Harry Kane get to the level of an NFL Ki…  2024-04-12T07:00:17Z
12                       Stick to Football: JJ Watt    2024-04-11T07:30:01Z
13  JJ Watt: NFL Superstar to Premier League Club …  2024-04-11T07:00:33Z
14  Eric Dier on tonight's Champions League clash …  2024-04-09T07:00:22Z
15                              Eric Dier: Exclusive    2024-04-08T19:12:06Z
16           Eric Dier: "I Should Be Part Of England"  2024-04-08T19:00:10Z
17                Man City & Real Madrid combined XI    2024-04-08T16:29:52Z
18  "At half-time Stuart Pearce was shouting… Ri…  2024-04-08T07:00:03Z
19  Vinnie Jones On Music In Dressing Rooms, Elton…  2024-04-07T07:00:30Z
20  Rio Ferdinand predicts Man United vs Liverpool…  2024-04-07T07:00:27Z
21  Gary Neville & Roy Keanes nuggets in the showe…  2024-04-06T07:00:09Z
22      Why Mac Allister Is The Signing Of The Season  2024-04-05T07:40:08Z
23   "I've never been so embarrassed on the pitch!"    2024-04-05T07:00:32Z
24                     Stick to Football: Rio Ferdinand  2024-04-04T07:00:37Z
25  Rio: Rashford Worry, Sir Alex & Liverpool Riva…  2024-04-04T07:00:02Z
26  Who would make your Chelsea v Man United combi…  2024-04-03T16:00:45Z
27  Is it failure if Arsenal don't win a trophy th…  2024-04-03T16:00:06Z
28              Aston Villa vs Man City combined XI?    2024-04-02T17:15:02Z
29  What's Bigger Premier League or Champions Leag…  2024-04-02T07:00:15Z
30                                      Gary vs Les!    2024-04-01T07:00:12Z
31                       CITY vs ARSENAL PREDICTIONS!    2024-03-31T07:00:02Z
32                    Man City v Arsenal combined XI?    2024-03-30T08:00:13Z
33                         Mikel Arteta's Big Game Plan  2024-03-30T08:00:03Z
34  Gary Neville On Playing With The Charlatans, D…  2024-03-29T12:00:50Z
35       "This could be Arsenal's defining moment!"    2024-03-29T08:00:33Z
36  "You Have to Buy Me Something!" | Super 6 Pred…  2024-03-29T08:00:26Z
37  Man City v Arsenal & Kobbie Mainoo for England…  2024-03-28T08:00:17Z
38  "It's still got a single camera!"  Roy's phon…  2024-03-27T12:00:07Z
39  "What if Kane had gone to Man City instead of …  2024-03-27T08:00:25Z
40  How Messi and Ronaldo caused the demise of the…  2024-03-26T17:00:03Z
41       Who's better - Alan Shearer or Harry Kane?    2024-03-26T12:00:06Z
42      Stick to Football: Les Ferdinand  - OUT NOW   2024-03-26T08:05:00Z
43  Les Ferdinand: The Demise of English Strikers …  2024-03-26T08:00:06Z
44  What would have happened to Phil Neville if Be…  2024-03-26T08:00:04Z
45     Rodri is so influential in this Man City team!  2024-03-25T08:00:00Z
46  "Let's not make out the England squad are best…  2024-03-24T08:00:23Z
47  Who else saw Ian Wright doing the FA Cup draw …  2024-03-23T08:00:34Z
48  What England's Midfield Should Be To Win EURO …  2024-03-23T08:00:33Z
49  How much would Roy Keane be worth in todays ma…  2024-03-22T08:00:29Z
```

	num_views	num_likes	num_comm
0	34506	991	68
1	29822	613	53
2	461812	12381	933
3	55323	1031	22
4	10389	180	42
5	28304	528	17
6	32748	574	61
7	9947	192	31
8	37146	534	14
9	45673	675	13
10	17398	409	120
11	80945	1294	35
12	61759	1184	26
13	903156	15597	1510
14	48168	790	46
15	29805	559	29
16	328915	5198	470
17	44610	531	100
18	144226	2450	36
19	150240	2086	161
20	95201	2030	54
21	104099	2768	45
22	61771	1319	258
23	188389	3555	96
24	111923	1861	38
25	1942849	23871	2792
26	36751	523	96
27	32138	428	46
28	41400	438	77
29	775204	9792	1383
30	107968	2097	35
31	26668	364	24
32	56506	749	149
33	54264	1025	87
34	190301	2390	240
35	59362	941	42
36	192868	2845	192
37	903823	10000	1891
38	198002	2568	148
39	166123	3040	121
40	279077	4895	133
41	129791	2360	137
42	62991	1063	13
43	1242145	16045	1733
44	362922	8223	71

45	49504	965	35
46	246039	4495	147
47	124584	2004	41
48	95767	1359	438
49	256433	4419	201

8 Exercises

1. Find an API online, and create a DataFrame in Python from the API data.
2. Use the YouTube API to get details on videos from a particular YouTube channel.

[ ]: