NY Times API Project

Introduction/Summary

Here is my presentation of my NY Times API Project. Within this documentation, I will be showing hints and details of specific parts of my code. My end goal is to show and support my few but many questions in regard to the NY Times API site and the search for articles written on diets.

My project is intended to answer the following questions: How many articles about diet were written within a 6-year span every day on every New Year's Day (related to New Year's Resolution). Within this project, I was able to see a dramatic change in my findings & was capable of seeing new ones as well. Though my project aim was specifically referencing the annual New Year's Resolution, much of my findings can be referenced in a small portion of the data usage with the pandemic of COVID-19. Now, I will be demonstrating the building blocks of this project.

"The Building Blocks"

Upon approaching this project, I knew there were several packages and technologies I did want to include. Other technologies I included were list methods such as append(), pprint modules, many parameters, and JSON-- those are some, just to name a few. Some packages I did enjoy included creating a CSV file/Excel sheet and a line plot (matplotlib). I felt those two factors greatly impacted this project and allowed users (and myself) to see the change in a visual way. Beyond the numbers, I do believe the line plot allows for the comprehension of the decrease due to its visual component. Much of the code I documented below is what I call "the essentials". These parts of the code are what truly made this project come together. These were the building blocks that allowed for this idea to carry through-- beginning to end!

Block 1

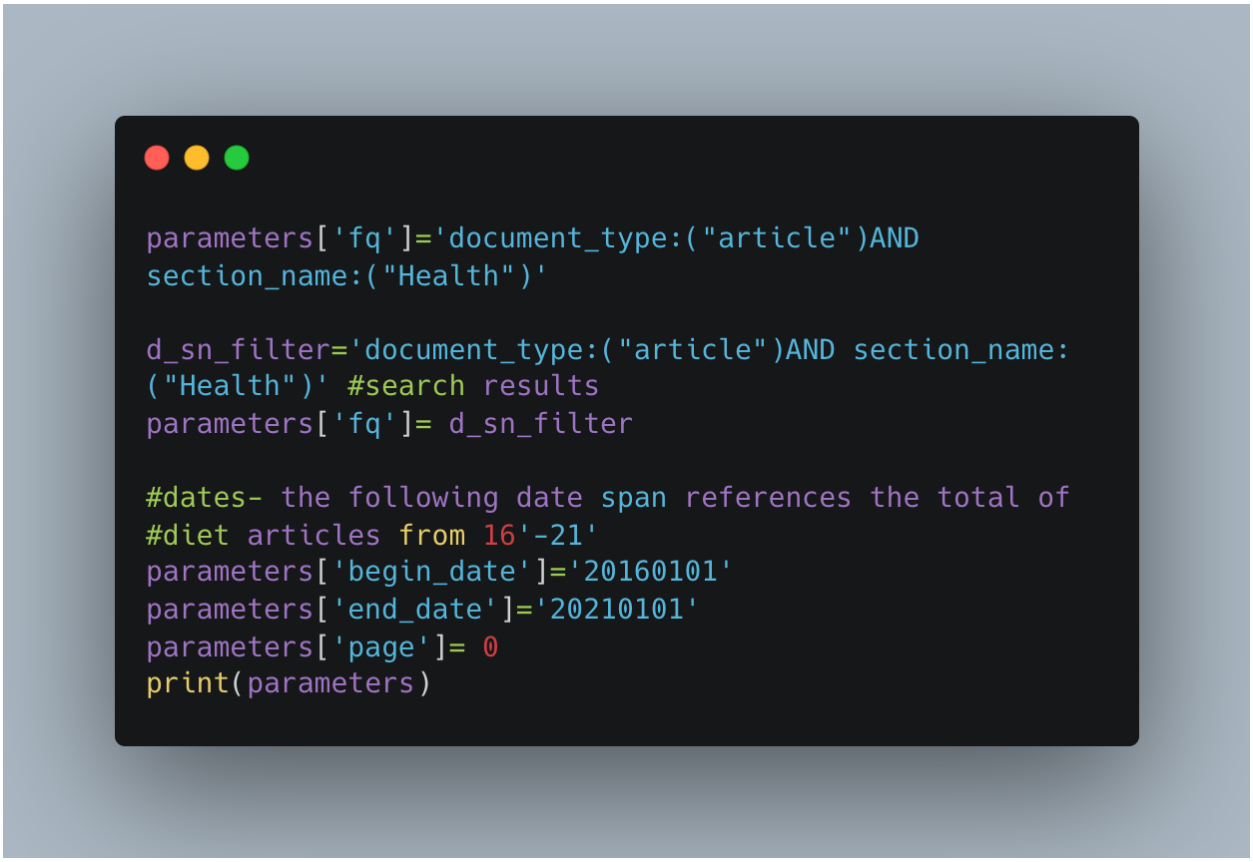
Below are the first results of the hits for the search for "diet." One can see any average number of the results. In my original code, the average number of articles shown on a diet was an average of nine thousand. While creating this project, I could see the number of articles change every few days or weeks. I did verify the results of

the NYTime site and the output of my code. Therefore, I noted the actual code of the average number amount that should be present.

```
parameters={'q':'diet', 'api-key': 'WU8iszA33rPw8ZIlhA4S9WqJ1Evgy9uS'}
response=requests.get(base_url,params=parameters)
content=response.json()
#how content - expanded
print(response.json())
pprint.pprint(content)
content=response.json()
print(content['response']['meta']['hits'])
```

Block 2

I could simplify my original topic using the code below based on the NYT API site and its filters. With my keyword on the prior code being “diet”, I decided to key my search to be based on articles on a diet in the health section. To make it more concentrated, I took a time period of 5 years to gain a reference for potential changes. From there, I could generate parameters capable of giving me a workable and decent amount of data information. Though the gap is five years, I was concrete to make the start and end date by Jan 1st of each year. Not only in homage to the theme/idea of New Year’s Resolution, however, its purpose was to also get precise data with it being the first DAY of the year versus a random date.



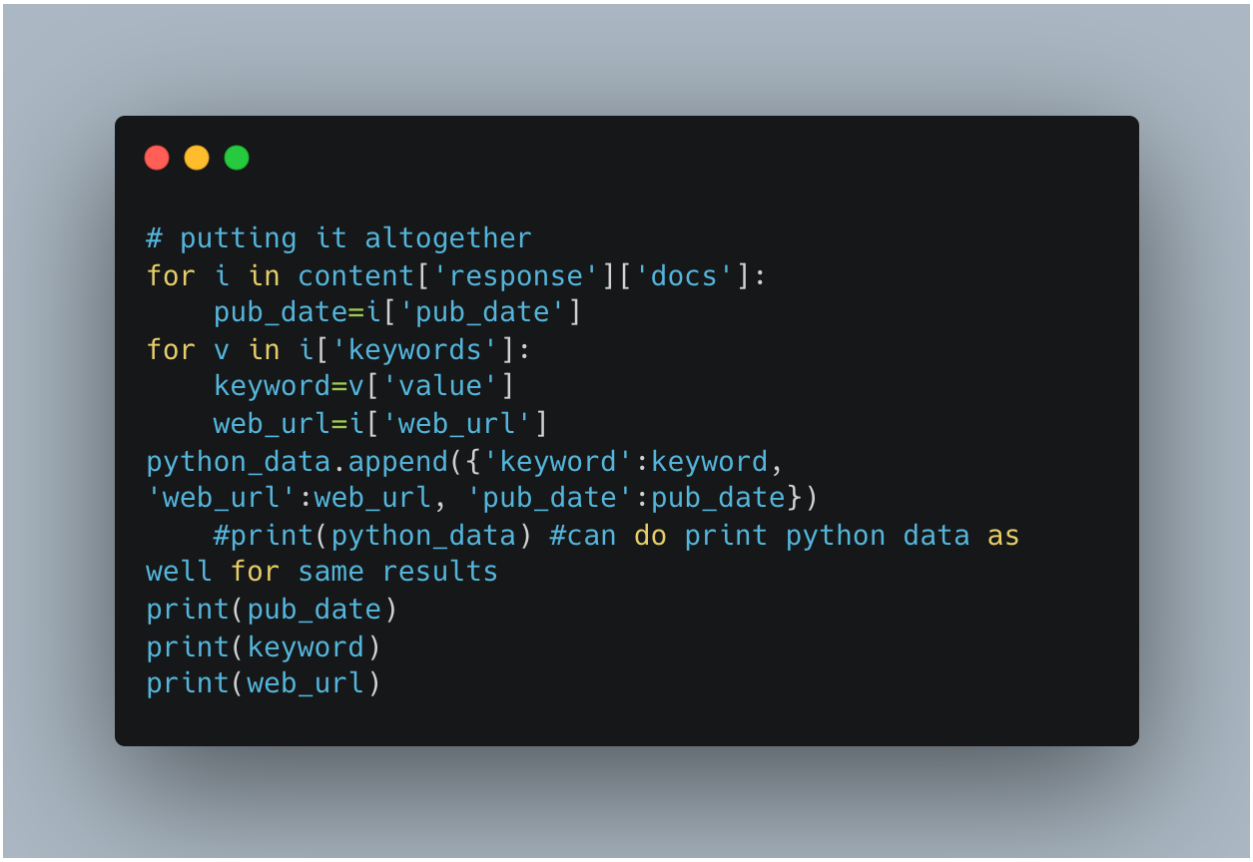
```
parameters['fq']='document_type:("article")AND
section_name:("Health")'

d_sn_filter='document_type:("article")AND section_name:
("Health")' #search results
parameters['fq']= d_sn_filter

#dates- the following date span references the total of
#diet articles from 16'-21'
parameters['begin_date']='20160101'
parameters['end_date']='20210101'
parameters['page']= 0
print(parameters)
```

Block 3

This code below essentially “taps” in the large content given from the results and provides specific content which is in the output. Those specific keywords or content prevalent to this project that I was able to see where the publishing date, keywords/values, and its URLs. By getting this information, I can see information associated with my original search and topic. As a result, much of the output left me curious about this project’s future progressions and how those content (specific keywords) can be used further.



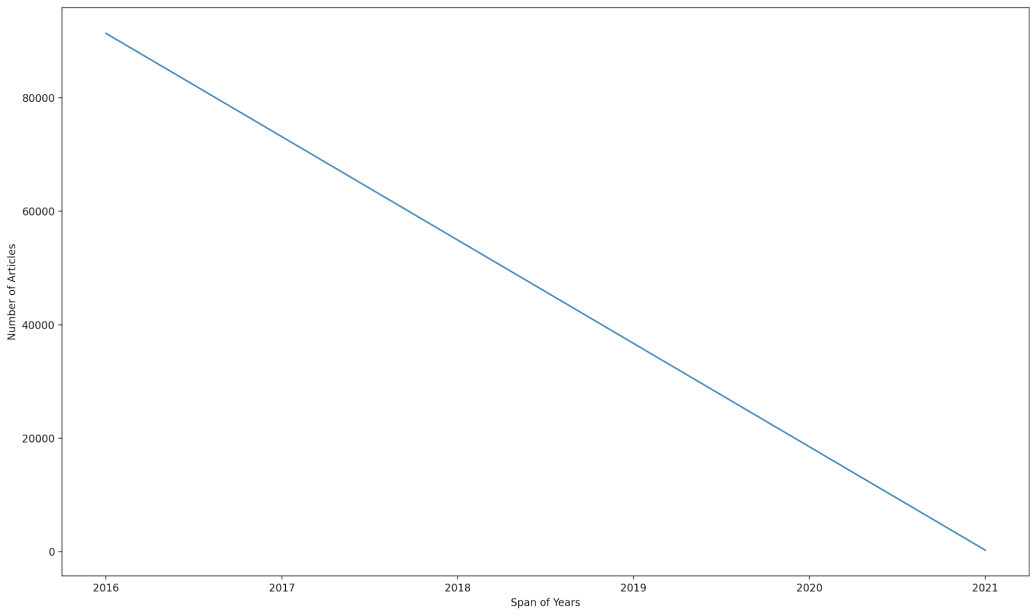
```
# putting it altogether
for i in content['response']['docs']:
    pub_date=i['pub_date']
for v in i['keywords']:
    keyword=v['value']
    web_url=i['web_url']
python_data.append({'keyword':keyword,
'web_url':web_url, 'pub_date':pub_date})
    #print(python_data) #can do print python data as
well for same results
print(pub_date)
print(keyword)
print(web_url)
```

Block 4

Below is the line plot or matplotlib code that gives the visual line plot of the start and end (or decrease) of articles with the keyword diet within the “articles' ' and “health” filters.

```
#LOAD --
#2016 data start date to 2021 end date - results will
show in a line graph using matplotlib
plt.plot([2016,2021], [91372,292])
plt.ylabel('Number of Articles')
plt.xlabel('Span of Years')
plt.show()
```

Python Mon Dec 20 3:19 PM

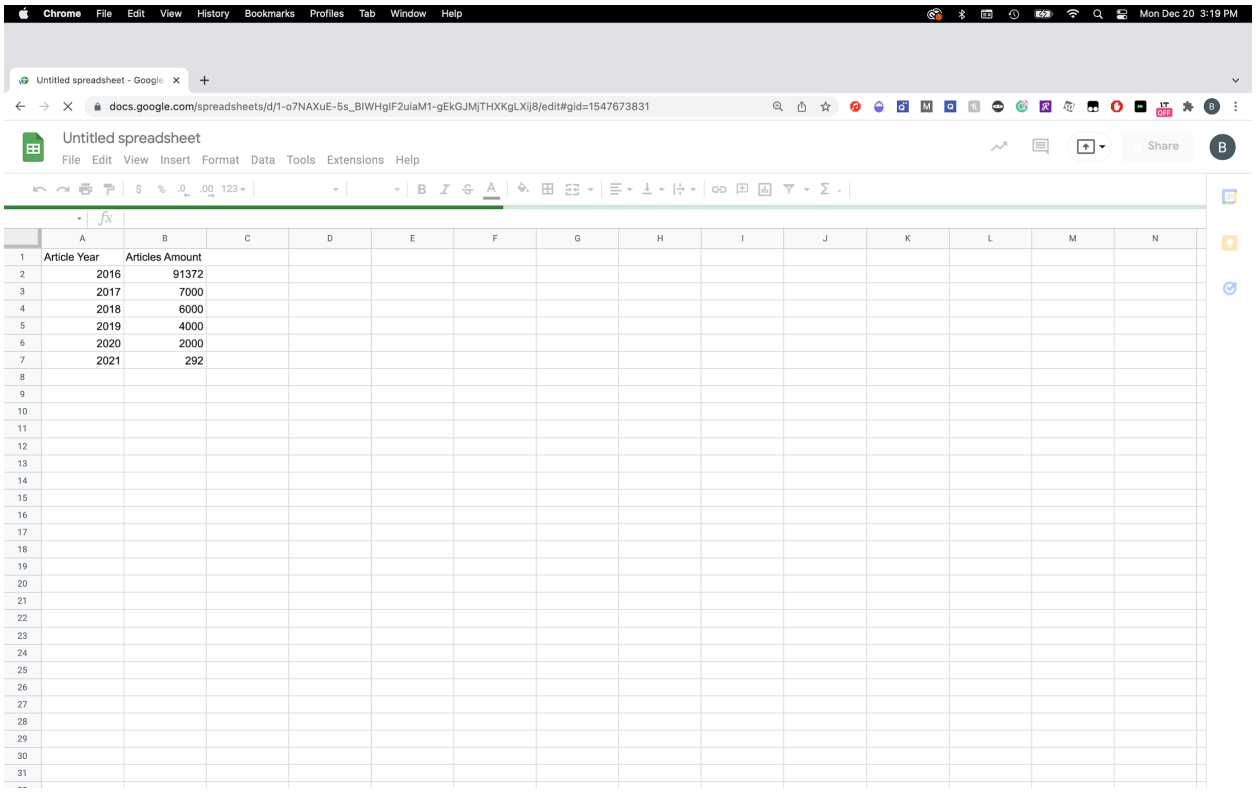


x=2016.538 y=3.991e+04

Block 5

Below is the line plot data that is displayed within a CSV chart. I attached the visual below that essentially explains the code.

```
nyt_data_info=[
{"Article Year":"2016","Articles Amount":91372}, #9.26
in decimial the following numbers in between will be
rounded
{"Article Year":"2017","Articles Amount":7000}, #7.17
{"Article Year":"2018","Articles Amount":6000}, #5.73
{"Article Year":"2019","Articles Amount":4000}, #3.79
{"Article Year":"2020","Articles Amount":2000},#1.71
{"Article Year":"2021","Articles Amount":292} ] #5.e +
03
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Article Year	Articles Amount												
2	2016	91372												
3	2017	7000												
4	2018	6000												
5	2019	4000												
6	2020	2000												
7	2021	292												

Conclusion

To conclude, these are the findings of my NYTimes API Report. Much of the data I discovered helped support the answer to my main question. As described above, those codes listed really became the building blocks to support my main question regarding finding further information about the reduction/increase of articles written on a diet.

When it comes to the term “diet,” is a constant trend that develops, changes, and is constantly discussed in everyday society. While building this project, I was able to see a massive change of that. Since the data shown is to see if the number of articles changed each year and within a span of 6 years, I made sure the start and end dates were Jan 1st. It is to reference the theme of the New Year Resolution and the common theme for a diet to be a part of that. Usually, people care about diet, health, weight loss, and well-being in the new year.

Therefore, if I were to make a follow-up on this project, I would go beyond the term diet. For example, my future plans would be to find the results or hits of keywords associated with diet. In addition, I would discover the result of those new searched keywords and make a CSV and line plot comparing both or multiple terms. I hope to compare the results and understand both prevalence and importance in terms of topics for articles and society as a whole (in addition to living in a pandemic).

Some packages I would include, in addition to what I have included in the original, would be Textblob, so I may be able to do textual analysis. With this specific package, I would be able to find frequencies with certain words. And with the topic of diet, I know there are commonly specified words used in association beyond my NYTimes results. By further progressing with those options, I know there will be a success with this future follow-up project.