9_api_data

December 11, 2023

1 APIs: Exploring Guardian API Data

Guardian news data provides us a range of different types of variable that we can use to get an overall picture of our datatset, and perhaps even find a some interesting patterns along the way.

Below we look at a range of different options for examining Guardian Data. Whilst the text of the stories is obviously valuable data, we'll need more advanced text analysis methods for that. These methods allow us to get a good overall picture of our data and find general trends.

[ ]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[ ]:

```python
articles = pd.read_json('AI_articles.json')
articles.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2000 entries, 0 to 1999
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  2000 non-null   object
 1   type                2000 non-null   object
 2   sectionId           2000 non-null   object
 3   sectionName         2000 non-null   object
 4   webPublicationDate  2000 non-null   object
 5   webTitle            2000 non-null   object
 6   webUrl              2000 non-null   object
 7   apiUrl              2000 non-null   object
 8   fields              2000 non-null   object
 9   tags                2000 non-null   object
 10  isHosted            2000 non-null   bool
 11  pillarId            1980 non-null   object
 12  pillarName          1980 non-null   object
dtypes: bool(1), object(12)
memory usage: 205.1+ KB
```

1.1 Prepping the Data

First we get the data ready for analysis.

1.1.1 Transforming the date column

First we need to ensure that our data is clean and that the `webPublicationDate` is properly formatted as a `datetime`.

```
[ ]: articles.head()
```

```
[ ]:                                                   id         type   sectionId  \
     0  technology/2023/oct/31/educators-teachers-ai-l…      article  technology
     1  technology/ng-interactive/2023/oct/25/a-day-in…  interactive  technology
     2  technology/2023/oct/24/alphabet-q3-earnings-go…      article  technology
     3  stage/2023/sep/19/anthropology-review-hampstea…      article       stage
     4  film/2023/aug/20/tim-review-clunky-ai-paranoia…      article        film

       sectionName    webPublicationDate  \
     0  Technology  2023-10-31T10:00:39Z
     1  Technology  2023-10-25T13:38:11Z
     2  Technology  2023-10-24T22:07:37Z
     3       Stage  2023-09-19T12:02:55Z
     4        Film  2023-08-20T10:30:44Z

                                                 webTitle  \
     0  ‘Is this an appropriate use of AI or not?’: te…
     1                            A day in the life of AI
     2  Google Cloud revenue misses expectations despi…
     3  Anthropology review – clever AI missing-person…
     4           TIM review – clunky AI paranoia thriller

                                                   webUrl  \
     0  https://www.theguardian.com/technology/2023/oc…
     1  https://www.theguardian.com/technology/ng-inte…
     2  https://www.theguardian.com/technology/2023/oc…
     3  https://www.theguardian.com/stage/2023/sep/19/…
     4  https://www.theguardian.com/film/2023/aug/20/t…

                                                   apiUrl  \
     0  https://content.guardianapis.com/technology/20…
     1  https://content.guardianapis.com/technology/ng…
     2  https://content.guardianapis.com/technology/20…
     3  https://content.guardianapis.com/stage/2023/se…
     4  https://content.guardianapis.com/film/2023/aug…

                                                   fields  \
     0  {'byline': 'Johana Bhuiyan', 'body': '<p>In <a…
```

```
1  {'byline': 'Hannah Devlin Science Corresponden…
2  {'byline': 'Kari Paul', 'body': '<p>Google is …
3  {'byline': 'Mark Lawson', 'body': '<p>While sc…
4  {'byline': 'Wendy Ide', 'body': '<p>This styli…

                                               tags  isHosted     pillarId  \
0  [{'id': 'technology/technology', 'type': 'keyw…     False  pillar/news
1  [{'id': 'technology/artificialintelligenceai',…     False  pillar/news
2  [{'id': 'technology/alphabet', 'type': 'keywor…     False  pillar/news
3  [{'id': 'stage/stage', 'type': 'keyword', 'sec…     False  pillar/arts
4  [{'id': 'film/thriller', 'type': 'keyword', 's…     False  pillar/arts

  pillarName
0       News
1       News
2       News
3       Arts
4       Arts
```

```
[ ]: articles['webPublicationDate'] = pd.to_datetime(articles['webPublicationDate'])
```

1.1.2 Unpacking the Fields column

The content of the `fields` column is determined when we collect our API data, by what we passed to `show-fields` in our query parameters. However what is returned is a dictionary of information. Ideally we want to expand these dictionaries out and create additional columns for each field (`byline`, `body` and `wordcount`).

We'll mainly be using `wordcount` but the process will unpack all fields.

```
[ ]: articles.loc[0, 'fields']
```

```
[ ]: {'byline': 'Johana Bhuiyan',
      'body': '<p>In <a
     href="https://www.theguardian.com/technology/2023/oct/12/chatgpt-uses-writing-
     recipes-one-year">the year since OpenAI released ChatGPT</a>, high school
     teacher Vicki Davis has been rethinking every single assignment she gives her
     students. Davis, a computer science teacher at Sherwood Christian Academy in
     Georgia, was well-positioned to be an early adopter of the technology. She's
     also the IT director at the school and helped put together an AI policy in
     March: the school opted to allow the use of AI tools for specific projects so
     long as students discussed it with their teachers and cited the tool. In Davis's
     mind, there were good and bad uses of AI, and ignoring its growing popularity
     was not going to help students unlock the productive uses or understand its
     dangers.</p> <aside class="element element-rich-link element--thumbnail"> <p>
     <span>Related: </span><a
     href="https://www.theguardian.com/technology/2023/oct/26/ai-artificial-
     intelligence-investment-boom">Humanity at risk from AI 'race to the bottom',
```

says tech expert</a> </p> </aside> <p>"It’s actually changed how I design my projects because there are some times I want my students to use AI, and then there are times I don’t want them to," Davis said. "What am I trying to teach here? Is this an appropriate use of AI or not?"</p> <p>Like teachers across the US and UK, Davis, who also runs the education blog Cool Cat Teacher, spent the summer thinking through what the release of a technology could mean for her.</p> <p>Generative AI can produce images of the pope in a bomber jacket and answer nearly any math problem, so what could it do for students? Educators like her played with the tools and tried to understand how they work, what the utility could be - for teachers and students alike - and, perhaps most pressingly, how the software could be misused. Some took drastic measures, going so far as to abandon homework assignments as long as the technology was accessible.</p> <p>"It feels like we’re in some sort of lab experimenting with our kids because it’s changing so rapidly," Davis said. "If you had asked me about any of this last fall, I couldn’t have told you any of it because ChatGPT didn’t exist."</p> <p>In Davis’s senior level class, she prohibited the use of chatbots to code because until recently the College Board, which administers standardized tests like the SAT, didn’t permit AI assistance for programming. (This was <a href="https://apcentral.collegeboard.org/exam-administration-ordering-scores/administering-exams/preparing-for-exam-day/exam-security/artificial-intelligence-tools">recently changed</a> to allow for the use of generative AI as a supplemental tool.) But she has changed an annual project she assigns to incorporate AI into the process. Davis usually asks students to research current models of laptops and evaluate which would be the best fit based on where they want to go to college and what they want to study. Now she asks students to feed the research they have done on their computer options into ChatGPT and ask for a recommendation based on their chosen major and college. The students are then tasked with evaluating ChatGPT’s recommendation. Her goal is to show students how they can use their own knowledge and research on a topic to help them better supervise AI. </p> <aside class="element element-pullquote element--supporting"> <blockquote> <p>If you had asked me about any of this last fall, I couldn’t have told you any of it because ChatGPT didn’t exist</p> <footer> <cite>Vicki Davis</cite> </footer> </blockquote> </aside> <p>Teachers who spoke to the Guardian say their primary concern is helping students begin to use AI without enabling cheating. Looming over their futuristic lessons is a fear that an overreliance on these new tools could exacerbate the loss of learning many students suffered during the pandemic. Students had only returned to in-person instruction after two remote years when OpenAI launched ChatGPT, and many were still struggling with the huge hit to their ability to learn or engage in school at all.</p> <p>"There’s so much trauma, and AI can’t help me with that," said one Maryland high school teacher, Kevin Shindel.</p> <p> ***</p> <p>After a summer spent experimenting with AI, there’s little consensus among teachers on how to address its use in schools. Many educators in a nearly 370,000-person Facebook group called "ChatGPT for teachers" argue the widespread use of AI chatbots is inevitable and eagerly discuss the best ways to use these tools to make their jobs more efficient and help their students learn. Other teachers the Guardian spoke to suggested student use of the tools be banned until they learn more about the technology behind it.</p> <p>Still, others have focused largely on mitigating any AI-aided cheating; some have stopped assigning homework entirely, opting instead to have their students do supervised work in class. Some teachers have even required students to take handwritten exams or write the first drafts of essays by hand in class to ensure they are coming up with the ideas themselves.</p> <p>But all those the Guardian spoke to agree: regardless of where you land on its use, teachers everywhere are grappling with how to stay on top of constantly evolving generative AI tools.</p> <p>Shindel, a government teacher at a 3,300-student high school in Maryland, has been teaching his students about how AI impacts government and policy for 15 years, but he wasn't prepared for how quickly people would adopt ChatGPT. He spent the summer learning about and experimenting with various chatbots, and in July presented his findings to the school board in a 38-slide presentation titled <strong>"</strong>The promise and peril of ChatGPT in today's classroom".</p> <aside class="element element-rich-link element--thumbnail"> <p> <span>Related: </span><a href="https://www.theguardian.com/technology/ng-interactive/2023/oct/25/a-day-in-the-life-of-ai">A day in the life of AI</a> </p> </aside> <p>Shindel gave those in attendance ChatGPT-led activities to experiment with and posed questions about the ethics of its use ("What would a code of ethics for data usage and protection look like?"). Ultimately, he urged the school board to come up with a district-wide policy.</p> <p>"Teachers shouldn't be responsible for developing classroom policies alone," Shindel said. "There needs to be some kind of concerted, systemic effort."</p> <p>Shindel doesn't believe teachers and policymakers know enough about how chatbots collect student's personal information - or how to prevent cheating - to allow students to use it. He also worries the tools could exacerbate the lack of student engagement caused by remote learning. Students and teachers are still reeling from the impacts of the pandemic, Shindel said. A recent Harvard graduate school of education <a href="https://www.gse.harvard.edu/ideas/news/23/05/new-data-show-how-pandemic-affected-learning-across-whole-communities#:~:text=Living%20in%20a%20community%20where,of%20a%20year%20in%20reading.">study</a> concluded the average public school student between third and eighth grade was half a year behind in math and reading and that nearly all students failed to recover the learning lost after returning to in-person instruction. A 2021 <a href="https://www.gov.uk/government/publications/learning-during-the-pandemic/learning-during-the-pandemic-quantifying-lost-time--2">review of 10 studies</a> on pandemic learning loss published by the UK's Department for Education found that "disadvantaged primary school students were disproportionately behind expectations", with many students 50% further behind.</p> <p>"I have a couple classes that are almost completely silent. Students don't interact with each other or answer any questions,<strong>"</strong> Shindel said. </p> <p>Though they may be in the minority, other schools have made progress establishing AI policies. Little Falls high school in Minnesota decided to ban the use of AI tools entirely in an addendum to the school-wide cheating policy. Davis's class policy allows certain tools to be used but requires students to seek permission and review the links the AI cites as sources. Kimberly Van Orman, a University of Georgia philosophy

```
professor who is currently teaching a course on the ethics of AI, says she is
focusing on transparency. Van Orman requires her students to include the prompt
they entered into a chatbot and the response in any assignment they use it for
to ensure they don’t "use it in a way that takes the place of learning".</p>
<p>"If you’re trying to understand a concept from the book and you want to kind
of talk it over with ChatGPT, that would be fine," Van Orman said. "Consulting
it on your homework problem would not be fine."</p> <p>***</p> <p>Dozens of AI
apps targeting students have cropped up in the past few years. Photomath, for
instance, predates the current versions of ChatGPT and pitches itself as the No
1 app for math learning. Users can upload a picture of a math problem or
equation, and the app will give them the answer with explanations. But several
teachers said students began using it during the pandemic to cheat or, at the
very least, replace the "productive struggle" that results in learning.
Inevitably, students who relied on Photomath during the pandemic struggled when
they returned to the classroom, several teachers said. </p>  <aside
class="element element-pullquote element--supporting"> <blockquote> <p>There’s
so much trauma, and AI can’t help me with that</p> <footer> <cite>Kevin
Shindel</cite> </footer> </blockquote> </aside>  <p>But there are also tools
being built to refuse to just give students the answer. Khanmigo, an AI tutor
being piloted by educational non-profit Khan Academy, is trained instead to ask
questions that nudge students to better understand the material. When the
Guardian asked Khanmigo a basic programming question (implement a cache with
expirations in Javascript), the chatbot responded: "I can’t provide direct
answers or solutions to coding problems." When the Guardian was asked to solve
for z in the equation "3z = 15" and repeatedly responded with "I don’t know",
the AI tutor kept providing guidance on how to solve it until it finally
provided four multiple-choice options. Khanmigo was quicker to provide the right
answer when the Guardian responded with an incorrect answer twice. ChatGPT, on
the other hand, immediately provided the solution in both cases.</p> <p>Sal
Khan, the founder of Khan Academy, says the organization spent thousands of
hours training the system, which is powered by ChatGPT-4, to understand that
it’s not supposed to do people’s work for them. "We said stuff like: ‘You’re a
Socratic tutor, you are here to make the students actively learn, not just
passively,’" he said.</p> <p>Though it’s still in an experimental phase, these
training processes are what distinguishes an AI tutor from an AI cheating tool,
Khan argues.</p> <p>"This time next year, you’re going to have 50 [companies]
who say that they have an AI tutor," Khan said. "But probably 90% of them are
going to be somewhat shady and they’re just going slap a little bit of a layer
on top of ChatGPT-3.5. They’re going to be mainly cheating tools, and not good
ones."</p>',
 'wordcount': '1585'}
```

`.json_normalize` happens to do this kind of job for us, but it will create an entirely new dataframe from the results.

[ ]:
```
articles_field_data = pd.json_normalize(articles['fields'])
articles_field_data.head()
```

```
[ ]:                                              byline  \
     0                                     Johana Bhuiyan
     1  Hannah Devlin Science Correspondent, Rich Cous…
     2                                          Kari Paul
     3                                        Mark Lawson
     4                                          Wendy Ide

                                                  body wordcount
     0  <p>In <a href="https://www.theguardian.com/tec…      1585
     1  <figure class="element element-atom element--i…      1741
     2  <p>Google is doing well, but not well enough f…       554
     3  <p>While screenwriters strike, partly over the…       410
     4  <p>This stylishly icy-looking thriller sounds …        86
```

Having produced our dataframe of field data we just need to merge the `articles` dataframe and the new one together, matching up the indexes. When merging dataframes, left literally refers to the dataframe on the left of the operation, and right to the one most towards the right.

`left.merge(right)`

```
[ ]: articles = articles.merge(articles_field_data, left_index=True,␣
      ↪right_index=True)
     articles.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2000 entries, 0 to 1999
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  2000 non-null   object
 1   type                2000 non-null   object
 2   sectionId           2000 non-null   object
 3   sectionName         2000 non-null   object
 4   webPublicationDate  2000 non-null   datetime64[ns, UTC]
 5   webTitle            2000 non-null   object
 6   webUrl              2000 non-null   object
 7   apiUrl              2000 non-null   object
 8   fields              2000 non-null   object
 9   tags                2000 non-null   object
 10  isHosted            2000 non-null   bool
 11  pillarId            1980 non-null   object
 12  pillarName          1980 non-null   object
 13  byline              1944 non-null   object
 14  body                2000 non-null   object
 15  wordcount           2000 non-null   object
dtypes: bool(1), datetime64[ns, UTC](1), object(14)
memory usage: 316.5+ KB
```

1.1.3 Converting wordcount to numeric

Wordcount has been stored as a string. We can rectify that by using `.to_numeric`

```
[ ]: articles['wordcount'] = pd.to_numeric(articles['wordcount'])
```

1.2 Data Counts over time

A key question of a dataset about the news, would when this news took place. Equally, we may also be interested in trends over time. Depending on your query, it may be interesting to see if there were changing publication rates related to your topic of interest.

A simple `.describe` on the date column will tell us a little about the spread of the dates.

```
[ ]: articles['webPublicationDate'].describe(datetime_is_numeric=True)
```

```
[ ]: count                                    2000
     mean     2023-03-31 20:56:41.787000064+00:00
     min                2020-08-21 14:20:31+00:00
     25%      2023-01-31 11:58:43.750000128+00:00
     50%                2023-05-25 13:13:57+00:00
     75%      2023-08-15 11:19:21.750000128+00:00
     max                2023-11-08 15:05:42+00:00
     Name: webPublicationDate, dtype: object
```

If we want to see trends, we can group our rows by publication period such as by `Day`, `Month` or `Year`. To do this we make a special time grouping object, and then group our data using it. We count the number of articles in each group and then plot them as a line plot.

```
[ ]: time_grouper = pd.Grouper(key='webPublicationDate', freq='M')
     count_over_time = articles[['webPublicationDate','id']].groupby(time_grouper).
      ↪count().reset_index()
     count_over_time.head()
```

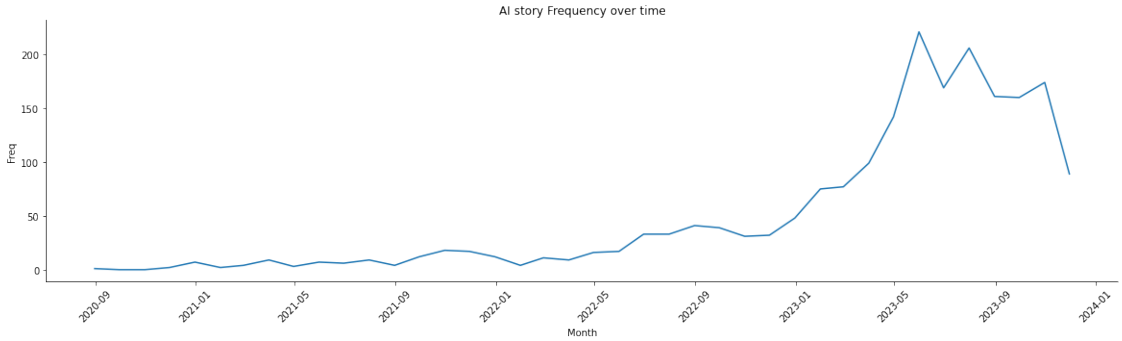
```
[ ]:          webPublicationDate  id
     0 2020-08-31 00:00:00+00:00   1
     1 2020-09-30 00:00:00+00:00   0
     2 2020-10-31 00:00:00+00:00   0
     3 2020-11-30 00:00:00+00:00   2
     4 2020-12-31 00:00:00+00:00   7
```

Time series are a little trickier to plot and Seaborn doesn't have a built in covnenience method for it. However we can use the `sns.lineplot` method to manually create one, and make some adjustments to size and label positioning manually.

```
[ ]: plt.figure(figsize=(20,5))
     plot = sns.lineplot(data=count_over_time, x='webPublicationDate', y='id')

     plot.tick_params(axis='x', labelrotation=45)
     plot.set(title='AI story Frequency over time', xlabel='Month', ylabel='Freq')
```

```
sns.despine()
plt.show()
```



1.3 Optional: Filtering by a Date Range

Using our timeseries plot we might decide to filter our data so we only work with a specific range period.

```
[ ]: date_filter = articles['webPublicationDate'] >= 'January 2022'
     articles = articles[date_filter]
```

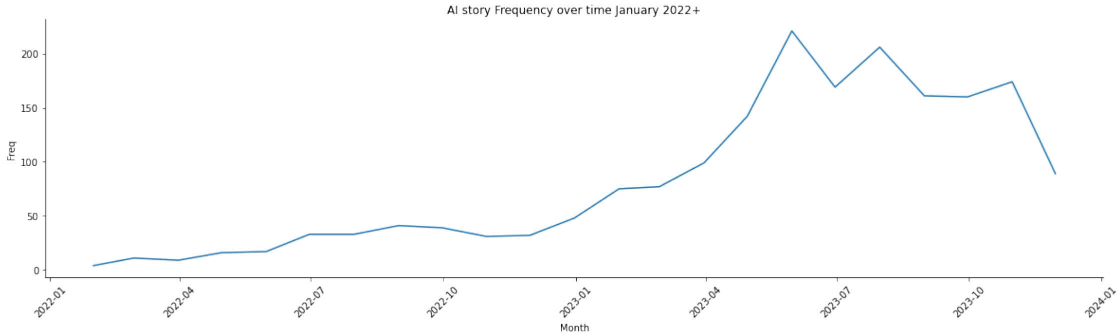
```
[ ]: articles['webPublicationDate'].describe()
```

```
<ipython-input-13-a6523f342fc1>:1: FutureWarning: Treating datetime data as
categorical rather than numeric in `.describe` is deprecated and will be removed
in a future version of pandas. Specify `datetime_is_numeric=True` to silence
this warning and adopt the future behavior now.
  articles['webPublicationDate'].describe()
```

```
[ ]: count                          1887
     unique                         1863
     top       2023-03-26 09:00:14+00:00
     freq                              3
     first     2022-01-09 09:00:17+00:00
     last      2023-11-08 15:05:42+00:00
     Name: webPublicationDate, dtype: object
```

```
[ ]: time_grouper = pd.Grouper(key='webPublicationDate', freq='M')
     count_over_time = articles[['webPublicationDate','id']].groupby(time_grouper).
       ↪count().reset_index()

     plt.figure(figsize=(20,5))
     plot = sns.lineplot(data=count_over_time, x='webPublicationDate', y='id')
     plot.tick_params(axis='x', labelrotation=45)
     plot.set(title='AI story Frequency over time January 2022+', xlabel='Month',␣
       ↪ylabel='Freq')
     sns.despine()
     plt.show()
```



1.4 Appropriate Pillars?

The Guardian has a number of major sections they refer to as Pillars. We can examine the distribution of our articles across these major categories.

[ ]:

```
pillar_counts = articles['pillarName'].value_counts()
pillar_counts
```

[ ]:

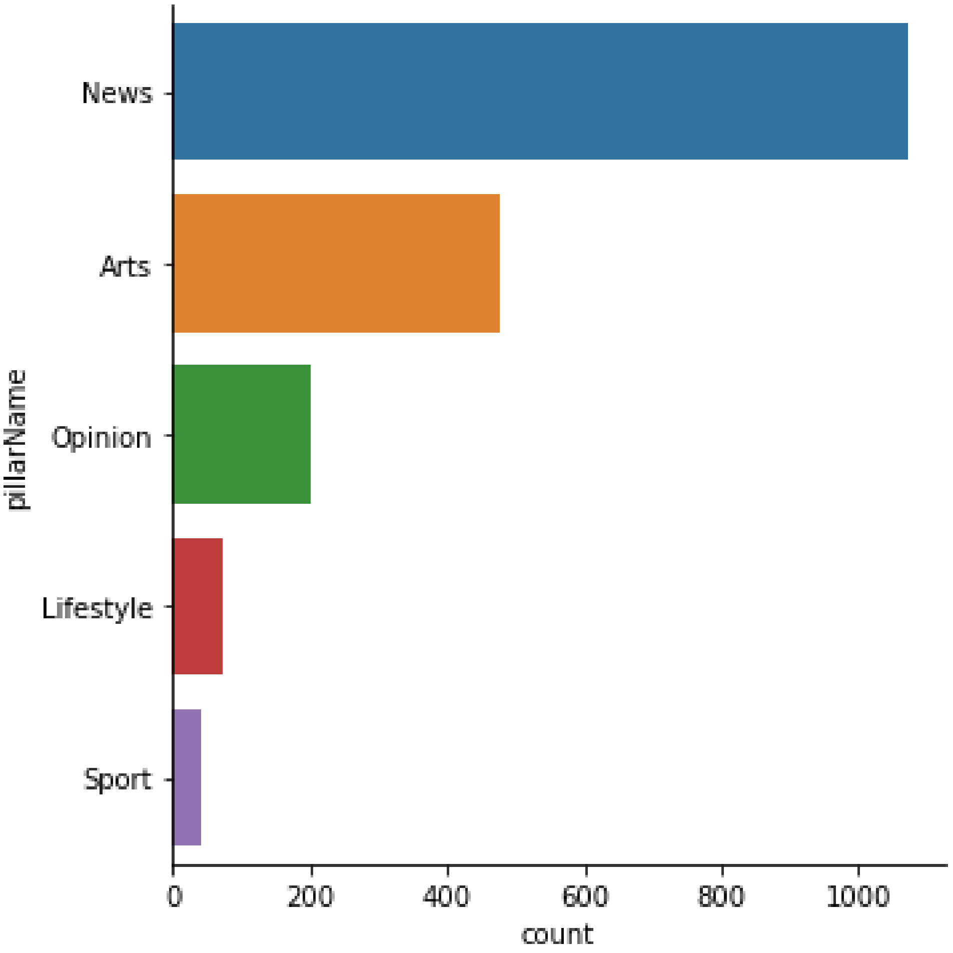
```
News         1072
Arts          479
Opinion       201
Lifestyle      73
Sport          42
Name: pillarName, dtype: int64
```

[ ]:

```
sns.catplot(data=articles, y='pillarName', kind='count', order=pillar_counts.
  ↪index)
```

[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x1207b3250>
```



1.5 Optional: Filtering by Pillar

Depending on your search query and the type of question you have, it may be worth filtering out material in unsuitable pillars, or focusing on just one.

[ ]:

```
chosen_pillars = ['News', 'Opinion']
pillar_filter = articles['pillarName'].isin(chosen_pillars)
articles = articles[pillar_filter]
```

After filtering we can re-run our counts to check the filtering was applied, and produce a new visualisation of we need it.

[ ]:

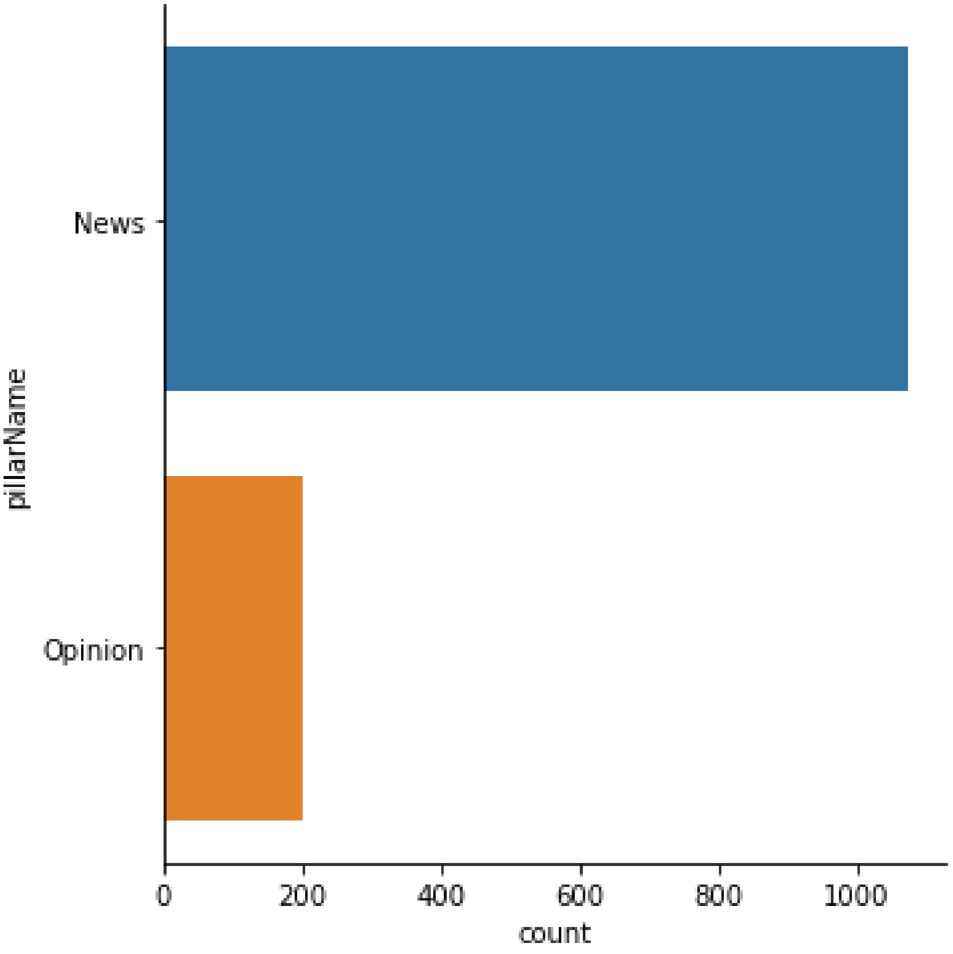
```
new_pillar_counts = articles['pillarName'].value_counts()
new_pillar_counts
```

[ ]:

```
News       1072
Opinion     201
Name: pillarName, dtype: int64
```

```
[ ]: sns.catplot(data=articles, y='pillarName', kind='count',␣
      ↪order=new_pillar_counts.index)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x11bd564f0>
```



1.6 Sections

Sections are the next form of categorisation. Sections give us a better sense of the overall topic of the stories.

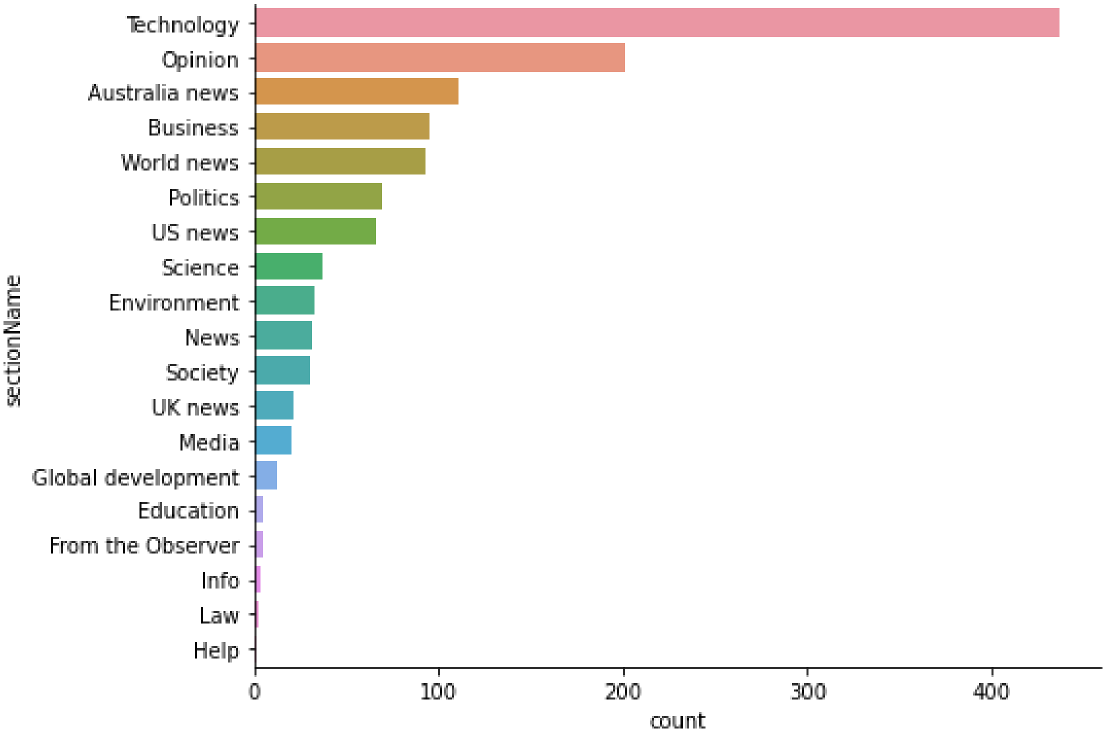
```
[ ]: section_counts = articles['sectionName'].value_counts()
     section_counts
```

```
[ ]: Technology          437
     Opinion             201
     Australia news      111
     Business             95
```

```
World news             93
Politics               69
US news                66
Science                37
Environment            33
News                   31
Society                30
UK news                21
Media                  20
Global development     12
Education               5
From the Observer       5
Info                    4
Law                     2
Help                    1
Name: sectionName, dtype: int64
```

```
[ ]: sns.catplot(data=articles, y='sectionName', kind='count', aspect=1.5,␣
      ↪order=section_counts.index)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x11be677c0>
```



Depending on how many sections are involved we may decide to keep only those above a certain threshold of presence in our dataset. This could be a top 10 or 20, or you could base it on some sort of summary metric of the counts such as categories above the mean or median count.

```
[ ]: section_counts.describe()
```

```
[ ]: count     19.000000
     mean      67.000000
     std      102.893148
     min        1.000000
     25%        8.500000
     50%       31.000000
     75%       81.000000
     max      437.000000
     Name: sectionName, dtype: float64
```

```
[ ]: above_avg_sections = section_counts[section_counts > section_counts.median()].
       ↪index
     above_avg_sections
```

```
[ ]: Index(['Technology', 'Opinion', 'Australia news', 'Business', 'World news',
            'Politics', 'US news', 'Science', 'Environment'],
           dtype='object')
```

We'll just go with a top 10

```
[ ]: top_sections = section_counts.index[:10]
     top_sections
```

```
[ ]: Index(['Technology', 'Opinion', 'Australia news', 'Business', 'World news',
            'Politics', 'US news', 'Science', 'Environment', 'News'],
           dtype='object')
```

```
[ ]: articles = articles[articles['sectionName'].isin(top_sections)]
     articles.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1173 entries, 0 to 1997
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  1173 non-null   object
 1   type                1173 non-null   object
 2   sectionId           1173 non-null   object
 3   sectionName         1173 non-null   object
 4   webPublicationDate  1173 non-null   datetime64[ns, UTC]
 5   webTitle            1173 non-null   object
 6   webUrl              1173 non-null   object
 7   apiUrl              1173 non-null   object
 8   fields              1173 non-null   object
 9   tags                1173 non-null   object
 10  isHosted            1173 non-null   bool
 11  pillarId            1173 non-null   object
 12  pillarName          1173 non-null   object
 13  byline              1138 non-null   object
 14  body                1173 non-null   object
 15  wordcount           1173 non-null   int64
dtypes: bool(1), datetime64[ns, UTC](1), int64(1), object(13)
memory usage: 147.8+ KB
```

1.6.1 Examining Section Content

We may find interesting sections in our dataset and wonder why they're there. We can iterate through the titles and URLs of the section we're interested in to get a better sense of why they've been included.

Below is a simple section filter but you could make it more complicated, such as limiting to after a time period, or in combination with a pillar classification for example.

N.B Below we use `.head` to limit the number of results for demonstration purposes, but during analysis there is no reason you cannot remove it and to view all the results.

[ ]:

```python
SECTION_OF_INTEREST = 'Australia news' # Just change this to switch sections


selected_data = articles[articles['sectionName'] == SECTION_OF_INTEREST].head(5)
for index, row in selected_data.iterrows():
    print(row['webTitle'])
    print(row['webUrl'])
    print('****')
```

```
Australian federal police using AI to analyse data obtained under surveillance
warrants
https://www.theguardian.com/australia-news/2023/sep/22/australian-federal-
police-afp-using-ai-analyse-surveillance-warrants-data
****
Morning Mail: ‘secretive’ Israel defence exports, bombshell testimony in Trump
trial, AI risks debated
https://www.theguardian.com/australia-news/2023/oct/25/morning-mail-secret-
israel-defence-exports-bombshell-testimony-in-trump-trial-ai-risks-debated
****
AI could ‘turbo-charge fraud’ and be monopolised by tech companies, Andrew Leigh
warns
https://www.theguardian.com/australia-news/2023/sep/20/ai-artificial-
intelligence-warnings-dangers-andrew-leigh-mckell-institute
****
Morning Mail: authors fear huge AI copyright ‘theft’, scorching weekend for
east, Michael Gambon dies
https://www.theguardian.com/australia-news/2023/sep/29/morning-mail-authors-
fear-huge-ai-copyright-theft-scorching-weekend-for-east-michael-gambon-dies
```

```
****
Democracies face ‘truth decay’ as AI blurs fact and fiction, warns head of
Australia’s military
https://www.theguardian.com/australia-news/2023/sep/14/democracies-face-truth-
decay-as-ai-blurs-fact-and-fiction-warns-head-of-australias-military
****
```

1.7 Tags

Tags are the last categorisation and they give us even more nuance in exactly what each story is about. However they are a little trickier to deal with because each story can have more than one tag associated with it. This presents us more of a challenge but also an opportunity for analysis too.

If we look at the first item in our `tags` column, we can see that the value is actually a quite complex object. A `list` and then each item in the `list` is a `dictionary`. A lot of information is provided but for our purposes we just want the tag string, which is held under the `webTitle` key.

```
[ ]: articles['tags'].iloc[0]
```

```
[ ]: [{'id': 'technology/technology',
       'type': 'keyword',
       'sectionId': 'technology',
       'sectionName': 'Technology',
       'webTitle': 'Technology',
       'webUrl': 'https://www.theguardian.com/technology/technology',
       'apiUrl': 'https://content.guardianapis.com/technology/technology',
       'references': []},
      {'id': 'technology/artificialintelligenceai',
       'type': 'keyword',
       'sectionId': 'technology',
       'sectionName': 'Technology',
       'webTitle': 'Artificial intelligence (AI)',
       'webUrl': 'https://www.theguardian.com/technology/artificialintelligenceai',
       'apiUrl':
     'https://content.guardianapis.com/technology/artificialintelligenceai',
       'references': []},
      {'id': 'education/education',
       'type': 'keyword',
       'sectionId': 'education',
       'sectionName': 'Education',
       'webTitle': 'Education',
       'webUrl': 'https://www.theguardian.com/education/education',
       'apiUrl': 'https://content.guardianapis.com/education/education',
       'references': []}]
```

As each story could have multiple tags we’re going to create a version of the `articles` dataframe where each row represents a single tag, and other story information like title, wordcount etc are duplicated. Pandas will keep track of which rows all refer to the same story using the index.

```
[ ]: tag_per_line = articles.explode('tags')
     tag_per_line.head(10)
```

```
[ ]:                                                  id         type   sectionId  \
     0  technology/2023/oct/31/educators-teachers-ai-l…      article  technology
     0  technology/2023/oct/31/educators-teachers-ai-l…      article  technology
     0  technology/2023/oct/31/educators-teachers-ai-l…      article  technology
     1  technology/ng-interactive/2023/oct/25/a-day-in…  interactive  technology
     1  technology/ng-interactive/2023/oct/25/a-day-in…  interactive  technology
     1  technology/ng-interactive/2023/oct/25/a-day-in…  interactive  technology
     1  technology/ng-interactive/2023/oct/25/a-day-in…  interactive  technology
     2  technology/2023/oct/24/alphabet-q3-earnings-go…      article  technology
     2  technology/2023/oct/24/alphabet-q3-earnings-go…      article  technology
     2  technology/2023/oct/24/alphabet-q3-earnings-go…      article  technology

       sectionName        webPublicationDate  \
     0  Technology 2023-10-31 10:00:39+00:00
     0  Technology 2023-10-31 10:00:39+00:00
     0  Technology 2023-10-31 10:00:39+00:00
     1  Technology 2023-10-25 13:38:11+00:00
     1  Technology 2023-10-25 13:38:11+00:00
     1  Technology 2023-10-25 13:38:11+00:00
     1  Technology 2023-10-25 13:38:11+00:00
     2  Technology 2023-10-24 22:07:37+00:00
     2  Technology 2023-10-24 22:07:37+00:00
     2  Technology 2023-10-24 22:07:37+00:00

                                                  webTitle  \
     0  ‘Is this an appropriate use of AI or not?’: te…
     0  ‘Is this an appropriate use of AI or not?’: te…
     0  ‘Is this an appropriate use of AI or not?’: te…
     1                          A day in the life of AI
     1                          A day in the life of AI
     1                          A day in the life of AI
     1                          A day in the life of AI
     2  Google Cloud revenue misses expectations despi…
     2  Google Cloud revenue misses expectations despi…
     2  Google Cloud revenue misses expectations despi…

                                                    webUrl  \
     0  https://www.theguardian.com/technology/2023/oc…
     0  https://www.theguardian.com/technology/2023/oc…
     0  https://www.theguardian.com/technology/2023/oc…
     1  https://www.theguardian.com/technology/ng-inte…
     1  https://www.theguardian.com/technology/ng-inte…
     1  https://www.theguardian.com/technology/ng-inte…
     1  https://www.theguardian.com/technology/ng-inte…
```

```
2  https://www.theguardian.com/technology/2023/oc…
2  https://www.theguardian.com/technology/2023/oc…
2  https://www.theguardian.com/technology/2023/oc…

                                              apiUrl  \
0  https://content.guardianapis.com/technology/20…
0  https://content.guardianapis.com/technology/20…
0  https://content.guardianapis.com/technology/20…
1  https://content.guardianapis.com/technology/ng…
1  https://content.guardianapis.com/technology/ng…
1  https://content.guardianapis.com/technology/ng…
1  https://content.guardianapis.com/technology/ng…
2  https://content.guardianapis.com/technology/20…
2  https://content.guardianapis.com/technology/20…
2  https://content.guardianapis.com/technology/20…

                                              fields  \
0  {'byline': 'Johana Bhuiyan', 'body': '<p>In <a…
0  {'byline': 'Johana Bhuiyan', 'body': '<p>In <a…
0  {'byline': 'Johana Bhuiyan', 'body': '<p>In <a…
1  {'byline': 'Hannah Devlin Science Corresponden…
1  {'byline': 'Hannah Devlin Science Corresponden…
1  {'byline': 'Hannah Devlin Science Corresponden…
1  {'byline': 'Hannah Devlin Science Corresponden…
2  {'byline': 'Kari Paul', 'body': '<p>Google is …
2  {'byline': 'Kari Paul', 'body': '<p>Google is …
2  {'byline': 'Kari Paul', 'body': '<p>Google is …

                                                tags  isHosted     pillarId  \
0  {'id': 'technology/technology', 'type': 'keywo…     False  pillar/news
0  {'id': 'technology/artificialintelligenceai', …     False  pillar/news
0  {'id': 'education/education', 'type': 'keyword…     False  pillar/news
1  {'id': 'technology/artificialintelligenceai', …     False  pillar/news
1  {'id': 'technology/computing', 'type': 'keywor…     False  pillar/news
1  {'id': 'technology/technology', 'type': 'keywo…     False  pillar/news
1  {'id': 'uk/uk', 'type': 'keyword', 'sectionId'…     False  pillar/news
2  {'id': 'technology/alphabet', 'type': 'keyword…     False  pillar/news
2  {'id': 'technology/google', 'type': 'keyword',…     False  pillar/news
2  {'id': 'technology/technology', 'type': 'keywo…     False  pillar/news

  pillarName                                            byline  \
0       News                                    Johana Bhuiyan
0       News                                    Johana Bhuiyan
0       News                                    Johana Bhuiyan
1       News  Hannah Devlin Science Correspondent, Rich Cous…
1       News  Hannah Devlin Science Correspondent, Rich Cous…
1       News  Hannah Devlin Science Correspondent, Rich Cous…
```

```
1       News  Hannah Devlin Science Correspondent, Rich Cous…
2       News                                        Kari Paul
2       News                                        Kari Paul
2       News                                        Kari Paul

                                                body  wordcount
0  <p>In <a href="https://www.theguardian.com/tec…       1585
0  <p>In <a href="https://www.theguardian.com/tec…       1585
0  <p>In <a href="https://www.theguardian.com/tec…       1585
1  <figure class="element element-atom element--i…       1741
1  <figure class="element element-atom element--i…       1741
1  <figure class="element element-atom element--i…       1741
1  <figure class="element element-atom element--i…       1741
2  <p>Google is doing well, but not well enough f…        554
2  <p>Google is doing well, but not well enough f…        554
2  <p>Google is doing well, but not well enough f…        554
```

If we check the length of the original `articles` dataframe against the new `tag_per_line` we can see that we have many more rows, one row per tag used in a story. We can also see the index values for our new dataframe are duplicated. This is because what was a single row, row `0` for example, is now three rows because the story had three tags. What was row `1` is now four rows, because the story at row `1` had four tags. These index values help us keep track of what rows 'go together' to make a single story, which we'll need later.

[ ]:
```python
len(tag_per_line)
```

[ ]:
```
7666
```

[ ]:
```python
len(articles)
```

[ ]:
```
1173
```

Now our `tags` column is similar in structure to our `fields` column, which we unpacked earlier using `.json_normalize`. We can do the same again to generate a seperate dataframe of `tag_data`. We also use a new method `.set_index()` to replace the default index that gets generated when `.json_normalize()` makes the new dataframe, with the index from `tag_per_line` which is keeping track of which rows go with which story.

[ ]:
```python
tag_data = pd.json_normalize(tag_per_line['tags'])
tag_data = tag_data.set_index(tag_per_line.index)
```

As we have index values keeping track of which row goes with which story, we could use those values to refer back to our original `articles` dataframe when we need additional information. However to simplify later tasks like printing out titles and urls, lets just copy the columns from `tag_per_line` into this new `tag_data` dataframe. The index lookup approach would be more space efficient but this isn't am issue with data this size.

[ ]:
```python
tag_data['wordcount'] = tag_per_line['wordcount']
tag_data['article_title'] = tag_per_line['webTitle']
```

```
tag_data['article_url'] = tag_per_line['webUrl']
tag_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7666 entries, 0 to 1997
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  7637 non-null   object
 1   type                7637 non-null   object
 2   sectionId           7614 non-null   object
 3   sectionName         7614 non-null   object
 4   webTitle            7637 non-null   object
 5   webUrl              7637 non-null   object
 6   apiUrl              7637 non-null   object
 7   references          7637 non-null   object
 8   description         391 non-null    object
 9   activeSponsorships  2 non-null      object
 10  wordcount           7666 non-null   int64
 11  article_title       7666 non-null   object
 12  article_url         7666 non-null   object
dtypes: int64(1), object(12)
memory usage: 838.5+ KB
```

We have quite a lot of columns in this dataset we've made, and probably only need a few. Let's just overwrite `tag_data` with a view of it that only includes the columns we need just to keep things simpler.

```
[ ]: tag_data = tag_data[['webTitle','article_title','article_url','wordcount']]
     tag_data.head()
```

```
[ ]:                         webTitle  \
     0                     Technology
     0  Artificial intelligence (AI)
     0                      Education
     1  Artificial intelligence (AI)
     1                      Computing

                                            article_title  \
     0  'Is this an appropriate use of AI or not?': te…
     0  'Is this an appropriate use of AI or not?': te…
     0  'Is this an appropriate use of AI or not?': te…
     1                              A day in the life of AI
     1                              A day in the life of AI

                                              article_url  wordcount
     0  https://www.theguardian.com/technology/2023/oc…       1585
     0  https://www.theguardian.com/technology/2023/oc…       1585
```

```
0  https://www.theguardian.com/technology/2023/oc…       1585
1  https://www.theguardian.com/technology/ng-inte…       1741
1  https://www.theguardian.com/technology/ng-inte…       1741
```

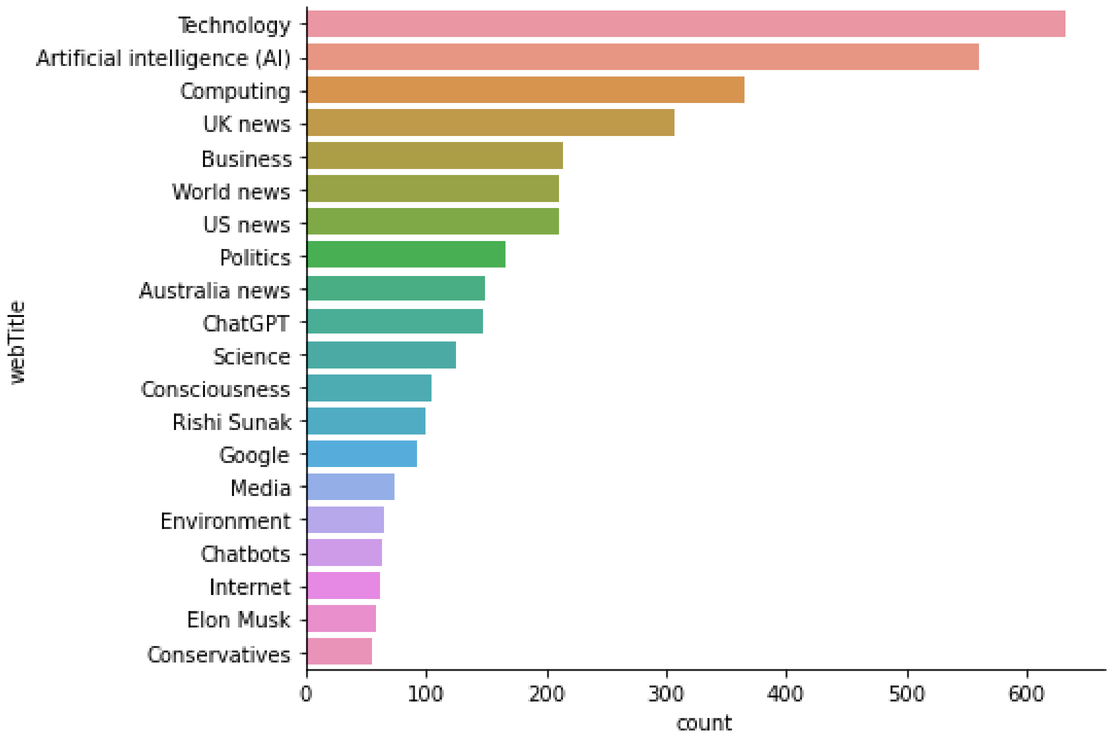
We can now check the count frequency of the different tags to get an overall picture like we did with sections.

```
[ ]: tag_counts = tag_data['webTitle'].value_counts().head(20)
     top_tags = tag_counts.index
     tag_counts
```

```
[ ]: Technology                        633
     Artificial intelligence (AI)      560
     Computing                         366
     UK news                           307
     Business                          214
     World news                        211
     US news                           211
     Politics                          166
     Australia news                    150
     ChatGPT                           147
     Science                           125
     Consciousness                     105
     Rishi Sunak                       100
     Google                             92
     Media                              73
     Environment                        65
     Chatbots                           63
     Internet                           62
     Elon Musk                          58
     Conservatives                      55
     Name: webTitle, dtype: int64
```

```
[ ]: sns.catplot(data=tag_data, y='webTitle', kind='count', aspect=1.5,␣
       ↪order=top_tags)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x120a202e0>
```



1.8 Titles by Tag

Like before with sections, we can examine what stories are associated with each tag. The column names will be different but the mechanics are the same. N.B Below we use `.head` to limit the number of results for demonstration purposes, but during analysis there is no reason you cannot remove it and to view all the results.

[ ]:

```
TAG_OF_INTEREST = 'Elon Musk' # Just change this to switch tags


selected_data = tag_data[tag_data['webTitle'] == TAG_OF_INTEREST].head()

for index, row in selected_data.iterrows():
    print(row['article_title'])
    print(row['article_url'])
    print('****')
```

```
Five takeaways from UK's AI safety summit at Bletchley Park
https://www.theguardian.com/technology/2023/nov/02/five-takeaways-uk-ai-safety-
summit-bletchley-park-rishi-sunak
****
Balancing the risks and rewards of AI will be key | Letters
https://www.theguardian.com/technology/2023/nov/06/balancing-the-risks-and-
rewards-of-ai-will-be-key
```

```
****
'Bletchley made me more optimistic': how experts reacted to AI summit
https://www.theguardian.com/technology/2023/nov/03/bletchley-made-me-more-
optimistic-how-experts-reacted-to-ai-summit
****
Sunak plays eager chatshow host as Musk discusses AI and politics
https://www.theguardian.com/politics/2023/nov/02/sunak-plays-eager-chatshow-
host-as-musk-discusses-ai-and-politics
****
Elon Musk unveils Grok, an AI chatbot with a 'rebellious streak'
https://www.theguardian.com/technology/2023/nov/05/elon-musk-unveils-grok-an-ai-
chatbot-with-a-rebellious-streak
****
```

We will use this data more later when examining wordcounts, and looking at tag correlation.

1.9 Word Counts

1.9.1 By Section

We defined `top_sections` earlier when we checked which sections had the highest number of stories. Here we'll use `.groupby` to get per section and per tag wordcounts. Word count is a good proxy for how much time was dedicated to a particular topic.

Total word count tells us the overall time dedicated to the topic related to each section or topic, whilst taking an average tells us how much space was given per story.

[ ]:

```
section_wordcounts = articles.groupby('sectionName').agg(
    avg_wordcount=('wordcount','mean'),
    total_wordcount=('wordcount','sum')
).sort_values('total_wordcount', ascending=False).loc[top_sections]

section_wordcounts
```

[ ]:

```
                avg_wordcount  total_wordcount
Technology         854.544622           373436
Opinion            905.955224           182097
Australia news    2039.099099           226340
Business          1412.684211           134205
World news        1616.053763           150293
Politics          2216.072464           152909
US news           1786.863636           117933
Science            917.540541            33949
Environment        874.878788            28871
News               267.451613             8291
```

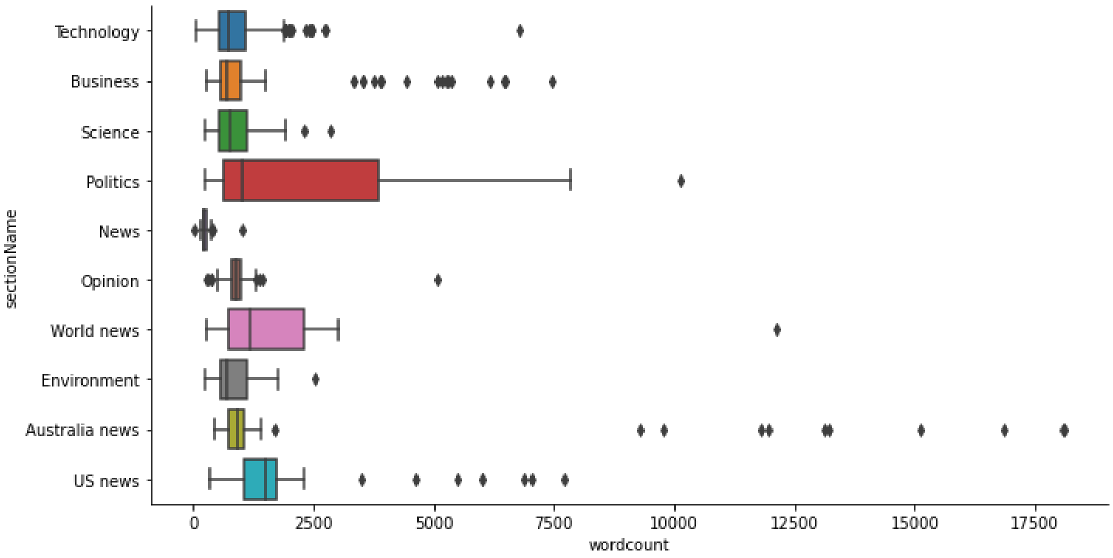
We can use box plots to see the distribution of these word counts. Remember we already filtered the `articles` data so it only included stories in top sections, however we include the filtering here to clarify that it is necessary before visualisation to reduce visual clutter.

[ ]:

```
to_plot = articles[articles['sectionName'].isin(top_sections)]
sns.catplot(data=to_plot, y='sectionName', x='wordcount', kind='box', aspect=2)
```

[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x120defd30>
```



1.9.2 By Tag

For a similar summary, but by tag, we do the same, but we use the `tag_data` dataframe, and the `webTitle` column.

[ ]:

```
tag_data.groupby('webTitle').agg(
    avg_wordcount=('wordcount','mean'),
    total_wordcount=('wordcount','sum')
).sort_values('total_wordcount', ascending=False).loc[top_tags]
```

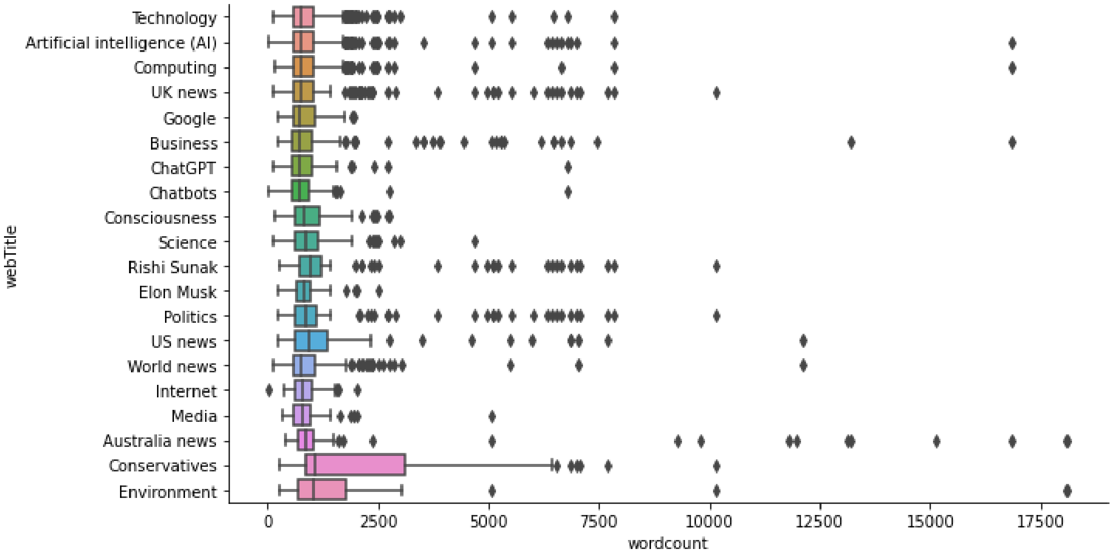
[ ]:

```
                              avg_wordcount  total_wordcount
Technology                       908.581359           575132
Artificial intelligence (AI)     977.994643           547677
Computing                        949.975410           347691
UK news                         1153.117264           354007
Business                        1263.654206           270422
World news                      1004.488152           211947
US news                         1245.033175           262702
Politics                        1471.198795           244219
Australia news                  1749.940000           262491
ChatGPT                          849.571429           124887
Science                         1017.304000           127163
Consciousness                    962.542857           101067
Rishi Sunak                     1849.910000           184991
```

```
Google                          865.043478          79584
Media                           900.136986          65710
Environment                    1925.907692         125184
Chatbots                        890.222222          56084
Internet                        857.306452          53153
Elon Musk                       900.931034          52254
Conservatives                  2261.545455         124385
```

```
[ ]: to_plot = tag_data[tag_data['webTitle'].isin(top_tags)]
     sns.catplot(data=to_plot, y='webTitle', x='wordcount', kind='box', aspect=2)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x120c8ceb0>
```



1.10 Tag Correlation

One analysis technique that is available to us is to examine the correlation of tags. What tags tend to co-occur in single stories, could this give us a sense of the themes or intersection of different topics?

Here we'll create a matrix of tag counts. In the first stage we use `.get_dummies` to reshape our column of tag names so that each possible tag is given its own column, and a value of 1 is entered if that tag is present in the row, otherwise 0.

(This may be a little confusing now but we're heading somewhere!)

```
[ ]: tag_matrix = pd.get_dummies(tag_data['webTitle'])
     tag_matrix
```

[ ]:

```
      3D  A-levels  AT&T  Abortion  Academics  Accountancy  Acting  Activism  \
0      0         0     0         0          0            0       0         0
0      0         0     0         0          0            0       0         0
0      0         0     0         0          0            0       0         0
1      0         0     0         0          0            0       0         0
1      0         0     0         0          0            0       0         0
…     ..         …     …         …          …            …       …         …
1997   0         0     0         0          0            0       0         0
1997   0         0     0         0          0            0       0         0
1997   0         0     0         0          0            0       0         0
1997   0         0     0         0          0            0       0         0
1997   0         0     0         0          0            0       0         0

      Adam Bandt  Adobe  …    YouTube  Young people  Youth unemployment  \
0              0      0  …          0             0                   0
0              0      0  …          0             0                   0
0              0      0  …          0             0                   0
1              0      0  …          0             0                   0
1              0      0  …          0             0                   0
…              …      …  …          …             …                   …
1997           0      0  …          0             0                   0
1997           0      0  …          0             0                   0
1997           0      0  …          0             0                   0
1997           0      0  …          0             0                   0
1997           0      0  …          0             0                   0

      Yuval Noah Harari  Yvette Cooper  Zambia  Zoology  iOS  iPad  iPhone
0                     0              0       0        0    0     0       0
0                     0              0       0        0    0     0       0
0                     0              0       0        0    0     0       0
1                     0              0       0        0    0     0       0
1                     0              0       0        0    0     0       0
…                     …              …       …        …    …     …       …
1997                  0              0       0        0    0     0       0
1997                  0              0       0        0    0     0       0
1997                  0              0       0        0    0     0       0
1997                  0              0       0        0    0     0       0
1997                  0              0       0        0    0     0       0

[7666 rows x 1032 columns]
```

Next we take our `tag_matrix`, use our list of `top_tags` to ensure only columns representing our selected top tags remain. We do this to aid visualisation later.

[ ]:

```
tag_matrix = tag_matrix[top_tags].copy()
tag_matrix.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 7666 entries, 0 to 1997
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Technology                7666 non-null   uint8
 1   Artificial intelligence (AI)  7666 non-null   uint8
 2   Computing                 7666 non-null   uint8
 3   UK news                   7666 non-null   uint8
 4   Business                  7666 non-null   uint8
 5   World news                7666 non-null   uint8
 6   US news                   7666 non-null   uint8
 7   Politics                  7666 non-null   uint8
 8   Australia news            7666 non-null   uint8
 9   ChatGPT                   7666 non-null   uint8
 10  Science                   7666 non-null   uint8
 11  Consciousness             7666 non-null   uint8
 12  Rishi Sunak               7666 non-null   uint8
 13  Google                    7666 non-null   uint8
 14  Media                     7666 non-null   uint8
 15  Environment               7666 non-null   uint8
 16  Chatbots                  7666 non-null   uint8
 17  Internet                  7666 non-null   uint8
 18  Elon Musk                 7666 non-null   uint8
 19  Conservatives             7666 non-null   uint8
dtypes: uint8(20)
memory usage: 209.6 KB
```

At the moment our `tag_matrix` is one row per tag per story, meaning that for every row *only one* of those columns will have a number 1 in it to represent a tag is associated with that story. In order to understand if certain tags correlate, if they go together, we need to simplify so that one row represents a story, and each column shows either a 0 or a 1 depending on whether the tag is present in that story.

As stories can only use each tag once if we took all the rows for one single story, and for each column added the row values together, the result would be one row where 1 indicates if the tag is there or not because if it's not, we'd simply be adding together 0 for each row, resulting in 0. Using `groupby` we can grab each set of rows representing a single story, `.sum()` together the values in each column and then get one row back which provides this representation.

Let's demo this with a simplified example…

```
[ ]: toy_matrix = pd.read_csv('toy_matrix.csv')
     toy_matrix
```

```
[ ]:   story  tag1  tag2  tag3
     0     A     1     0     0
     1     A     0     1     0
     2     A     0     0     1
     3     B     1     0     0
```

```
4     B      0      0      1
```

The `story` column just represents the id or title of the story, and then we have a column for each of three different tags. You'll see that each row only has one `1` in it because it is one row per tag. What we want is one row per story, so just two rows, one for story A, one for story B that puts the values spread across multiple rows into just one row.

You could probably do this in your head because really all we're saying is, for each story's subset of rows, if there is a `1` anywhere in the column, then the value is `1`, otherwise it's `0`. As we know a single story can only use a tag once, we can simplify this slightly complicated logic as just "grab all the rows for a story and for each column, add the values together".

[ ]:
```
toy_matrix.groupby('story').sum()
```

[ ]:
```
        tag1   tag2   tag3
story
A          1      1      1
B          1      0      1
```

We can do the same with our actual `tag_matrix`. As we want to group on the index of the dataframe rather than a column we don't have a column name to pass `.groupby()` as usual. However we can tell it to group by "level 0". Pandas refers to indexes as levels and on a regular dataframe with just a single index, there is only one level, level 0.

[ ]:
```
tag_matrix = tag_matrix.groupby(level=0).sum()
tag_matrix
```

[ ]:
```
      Technology  Artificial intelligence (AI)  Computing  UK news  Business  \
0              1                             1          0        0         0
1              1                             1          1        1         0
2              1                             0          0        0         1
5              1                             1          0        0         0
6              1                             1          0        0         0
…          …                           …        …      …       …
1991           0                             0          0        0         0
1992           0                             0          0        0         0
1994           0                             0          0        0         0
1996           0                             0          0        0         0
1997           0                             0          0        1         1

      World news  US news  Politics  Australia news  ChatGPT  Science  \
0              0        0         0               0        0        0
1              0        0         0               0        0        0
2              0        0         0               0        0        0
5              0        0         0               0        0        0
6              0        0         0               0        1        0
…          …      …       …             …      …      …
1991           1        0         0               0        0        0
1992           1        0         0               0        0        0
```

```
1994           0        1         0               0        0        0
1996           0        0         0               0        0        0
1997           0        0         1               0        0        0

      Consciousness  Rishi Sunak  Google  Media  Environment  Chatbots  \
0                 0            0       0      0            0         0
1                 0            0       0      0            0         0
2                 0            0       1      0            0         0
5                 0            0       0      0            0         0
6                 0            0       0      0            0         1
…             …          …     …    …          …       …
1991              0            0       0      0            0         0
1992              0            0       0      0            0         0
1994              0            0       0      0            0         0
1996              0            0       0      0            0         0
1997              0            0       0      0            0         0

      Internet  Elon Musk  Conservatives
0            0          0              0
1            0          0              0
2            0          0              0
5            0          0              0
6            0          0              0
…        …        …            …
1991         0          0              0
1992         0          0              0
1994         0          0              0
1996         0          0              0
1997         0          0              0

[1173 rows x 20 columns]
```

Finally we can get our correlation scores using `.corr`. This reshapes the data into a square, where both the rows and the columns represent tags, and the values represent the correleation between the two tags.

- 0 Represents no correlation
- 1 Represents the highest positive correlation, i.e. every story with tag `a` also includes tag `b`.
- A negative value indicates negative correlation, i.e. the presence of tag `a` means that the presence of tag `b` is less likely.

The 'diagonal' of the matrix will always equal 1 as the presence of tag `a` will always be correlated with the presence of tag `a`.

[ ]:

```
correlations = tag_matrix.corr()
correlations
```

```
[ ]:                              Technology  Artificial intelligence (AI)  \
     Technology                     1.000000                      0.574604
     Artificial intelligence (AI)   0.574604                      1.000000
     Computing                      0.533412                      0.605133
     UK news                        0.016848                      0.044402
     Business                       0.019890                     -0.101796
     World news                     0.098574                      0.050066
     US news                        0.000604                     -0.052139
     Politics                      -0.105899                      0.013467
     Australia news                -0.266054                     -0.161559
     ChatGPT                        0.266949                      0.349627
     Science                        0.167821                      0.188183
     Consciousness                  0.253656                      0.328054
     Rishi Sunak                   -0.061030                      0.056591
     Google                         0.205829                      0.171906
     Media                          0.103413                     -0.006011
     Environment                   -0.157571                     -0.119598
     Chatbots                       0.182106                      0.226543
     Internet                       0.157034                     -0.019828
     Elon Musk                      0.163318                      0.041806
     Conservatives                 -0.191595                     -0.074742

                                  Computing   UK news  Business  World news  \
     Technology                    0.533412  0.016848  0.019890    0.098574
     Artificial intelligence (AI)  0.605133  0.044402 -0.101796    0.050066
     Computing                     1.000000  0.034365 -0.055770    0.101387
     UK news                       0.034365  1.000000  0.099823    0.008971
     Business                     -0.055770  0.099823  1.000000    0.008604
     World news                    0.101387  0.008971  0.008604    1.000000
     US news                      -0.085447 -0.066770  0.077179    0.150511
     Politics                      0.001080  0.531680 -0.045873   -0.043685
     Australia news               -0.180739 -0.210569 -0.081274   -0.139455
     ChatGPT                       0.295293 -0.067208 -0.031944    0.043962
     Science                       0.224587  0.020464 -0.139635    0.060710
     Consciousness                 0.414030  0.003527 -0.108836    0.039749
     Rishi Sunak                   0.018436  0.380777 -0.041216   -0.007853
     Google                        0.159426 -0.051062  0.099730    0.020235
     Media                        -0.036388 -0.040989  0.051625   -0.019582
     Environment                  -0.106816 -0.076394 -0.046611    0.022389
     Chatbots                      0.206844 -0.038613 -0.014542    0.036109
     Internet                      0.038279 -0.001966  0.075426   -0.001514
     Elon Musk                     0.050099 -0.019501 -0.026135   -0.014672
     Conservatives                -0.105850  0.280779 -0.083416   -0.082877

                                   US news  Politics  Australia news   ChatGPT  \
     Technology                   0.000604 -0.105899       -0.266054  0.266949
     Artificial intelligence (AI) -0.052139  0.013467       -0.161559  0.349627
```

```
Computing                    -0.085447  0.001080       -0.180739  0.295293
UK news                      -0.066770  0.531680       -0.210569 -0.067208
Business                      0.077179 -0.045873       -0.081274 -0.031944
World news                    0.150511 -0.043685       -0.139455  0.043962
US news                       1.000000 -0.094629       -0.166041  0.003738
Politics                     -0.094629  1.000000       -0.155470 -0.109358
Australia news               -0.166041 -0.155470        1.000000 -0.067837
ChatGPT                       0.003738 -0.109358       -0.067837  1.000000
Science                      -0.081887 -0.013275       -0.114677  0.093756
Consciousness                -0.069098 -0.015930       -0.120065  0.224059
Rishi Sunak                  -0.039646  0.603007       -0.116898 -0.106333
Google                        0.036747 -0.100251       -0.064233  0.186482
Media                         0.017169 -0.043847       -0.077510 -0.001581
Environment                  -0.084329 -0.034195        0.018835 -0.069170
Chatbots                     -0.032810 -0.075029       -0.091226  0.321008
Internet                     -0.041194 -0.074050       -0.044820  0.048682
Elon Musk                     0.046756  0.008935       -0.075559 -0.003190
Conservatives                -0.072377  0.523148       -0.084931 -0.083955

                               Science  Consciousness  Rishi Sunak    Google  \
Technology                    0.167821       0.253656    -0.061030  0.205829
Artificial intelligence (AI)  0.188183       0.328054     0.056591  0.171906
Computing                     0.224587       0.414030     0.018436  0.159426
UK news                       0.020464       0.003527     0.380777 -0.051062
Business                     -0.139635      -0.108836    -0.041216  0.099730
World news                    0.060710       0.039749    -0.007853  0.020235
US news                      -0.081887      -0.069098    -0.039646  0.036747
Politics                     -0.013275      -0.015930     0.603007 -0.100251
Australia news               -0.114677      -0.120065    -0.116898 -0.064233
ChatGPT                       0.093756       0.224059    -0.106333  0.186482
Science                       1.000000       0.410658    -0.006437  0.012183
Consciousness                 0.410658       1.000000     0.021907  0.108460
Rishi Sunak                  -0.006437       0.021907     1.000000 -0.089060
Google                        0.012183       0.108460    -0.089060  1.000000
Media                        -0.065512      -0.031330    -0.078644  0.003603
Environment                   0.060726      -0.062892    -0.033911 -0.042939
Chatbots                      0.039921       0.150483    -0.072729  0.141484
Internet                     -0.044147      -0.034032    -0.058473  0.101151
Elon Musk                    -0.002283       0.080003     0.113432 -0.022657
Conservatives                -0.050016      -0.055420     0.495486 -0.064706

                                 Media  Environment  Chatbots  Internet  \
Technology                    0.103413    -0.157571  0.182106  0.157034
Artificial intelligence (AI) -0.006011    -0.119598  0.226543 -0.019828
Computing                    -0.036388    -0.106816  0.206844  0.038279
UK news                      -0.040989    -0.076394 -0.038613 -0.001966
Business                      0.051625    -0.046611 -0.014542  0.075426
```

```
World news                     -0.019582     0.022389  0.036109 -0.001514
US news                         0.017169    -0.084329 -0.032810 -0.041194
Politics                       -0.043847    -0.034195 -0.075029 -0.074050
Australia news                 -0.077510     0.018835 -0.091226 -0.044820
ChatGPT                        -0.001581    -0.069170  0.321008  0.048682
Science                        -0.065512     0.060726  0.039921 -0.044147
Consciousness                  -0.031330    -0.062892  0.150483 -0.034032
Rishi Sunak                    -0.078644    -0.033911 -0.072729 -0.058473
Google                          0.003603    -0.042939  0.141484  0.101151
Media                           1.000000     0.014728 -0.030066  0.175724
Environment                     0.014728     1.000000 -0.057703 -0.057217
Chatbots                       -0.030066    -0.057703  1.000000  0.214137
Internet                        0.175724    -0.057217  0.214137  1.000000
Elon Musk                       0.201686    -0.038053 -0.002007  0.121871
Conservatives                  -0.023752    -0.018468 -0.052841 -0.034372

                                Elon Musk  Conservatives
Technology                       0.163318      -0.191595
Artificial intelligence (AI)     0.041806      -0.074742
Computing                        0.050099      -0.105850
UK news                         -0.019501       0.280779
Business                        -0.026135      -0.083416
World news                      -0.014672      -0.082877
US news                          0.046756      -0.072377
Politics                         0.008935       0.523148
Australia news                  -0.075559      -0.084931
ChatGPT                         -0.003190      -0.083955
Science                         -0.002283      -0.050016
Consciousness                    0.080003      -0.055420
Rishi Sunak                      0.113432       0.495486
Google                          -0.022657      -0.064706
Media                            0.201686      -0.023752
Environment                     -0.038053      -0.018468
Chatbots                        -0.002007      -0.052841
Internet                         0.121871      -0.034372
Elon Musk                        1.000000      -0.013384
Conservatives                   -0.013384       1.000000
```

We can check the correlations for a specific tag by accessing its column…

```
[ ]: correlations['ChatGPT'].sort_values(ascending=False)
```

```
[ ]: ChatGPT                         1.000000
     Artificial intelligence (AI)    0.349627
     Chatbots                        0.321008
     Computing                       0.295293
     Technology                      0.266949
     Consciousness                   0.224059
```

```
Google                         0.186482
Science                        0.093756
Internet                       0.048682
World news                     0.043962
US news                        0.003738
Media                         -0.001581
Elon Musk                     -0.003190
Business                      -0.031944
UK news                       -0.067208
Australia news                -0.067837
Environment                   -0.069170
Conservatives                 -0.083955
Rishi Sunak                   -0.106333
Politics                      -0.109358
Name: ChatGPT, dtype: float64
```

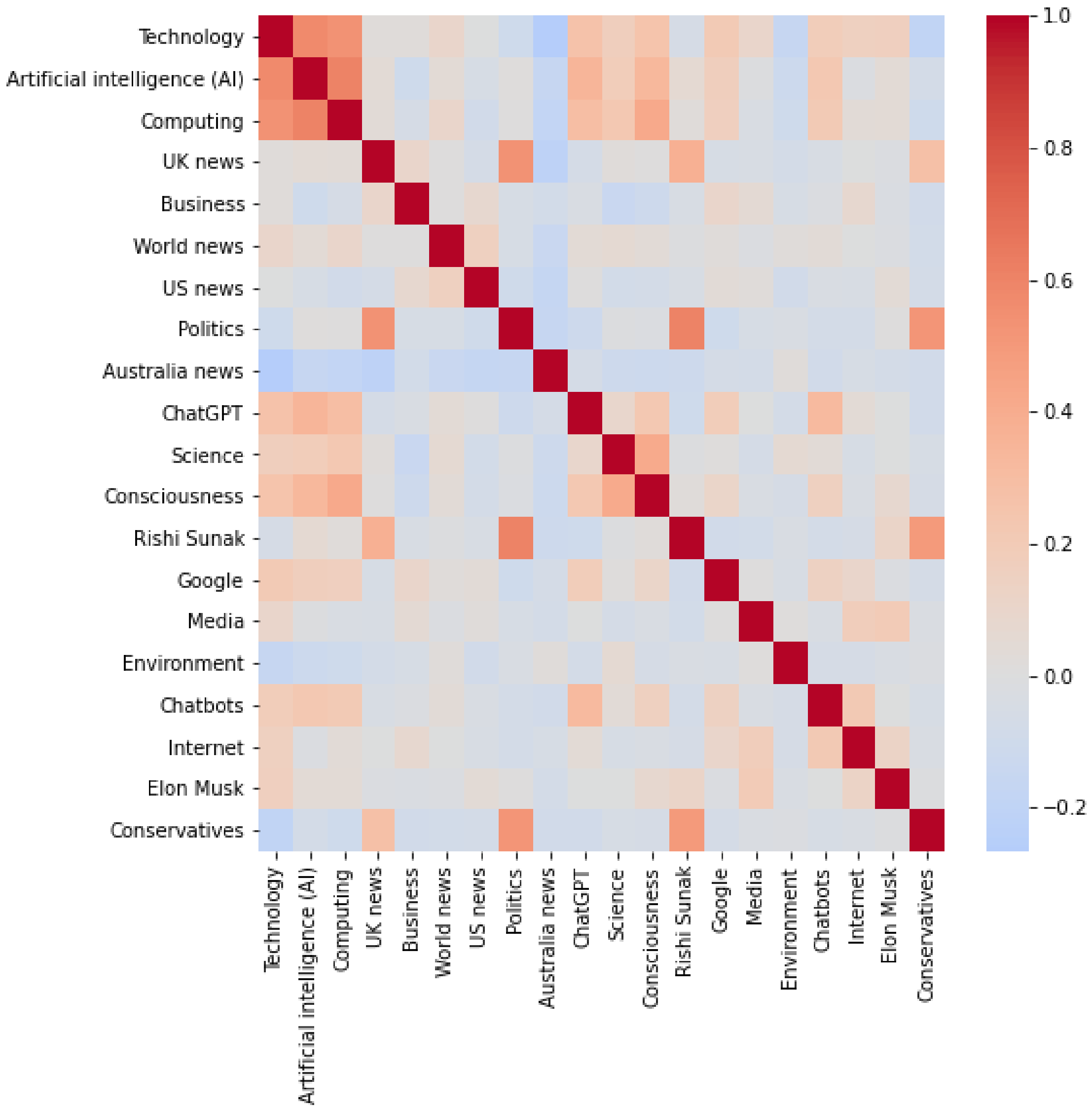
1.10.1 Tag Heatmap

We can also visualise these correlations using a heatmap. Using the `coolwarm` color scheme means colours run from deep blue to deep red. We set the `center` of the scale to 0 so that above zero, positive correlation, is a shade of red whilst below zero, negative correlation, is a shade of blue.

```
[ ]: plt.figure(figsize=(8,8))

     sns.heatmap(correlations, cmap='coolwarm', center=0)
```

```
[ ]: <AxesSubplot:>
```



1.10.2 Advanced: Identifying multi-tag titles

What if you wanted to understand WHY two tags correlate. Perhaps ones that are unexpected. You will need to identify which stories have both tags using our `tag_matrix`, and then use the index values to look up the correct rows in the `articles`. We can then iterate over them and view title and url like before.

[ ]:
```
TAG_1 = 'ChatGPT'
TAG_2 = 'Consciousness'
tag_filter = (tag_matrix[TAG_1] == 1) & (tag_matrix[TAG_2] == 1)

selected_story_index = tag_matrix[tag_filter].index
selected_story_index
```

```
[ ]: Int64Index([  74,  102,  171,  181,  206,  207,  212,  225,  270,  299,  317,
                  342,  352,  424,  446,  460,  477,  478,  512,  524,  530,  563,
                  578,  670,  759,  768,  788,  793,  796,  804,  908,  945,  967,
                  975,  981, 1049, 1077, 1187],
                dtype='int64')
```

```
[ ]: selected_data = articles.loc[selected_story_index].head()

     for index, row in selected_data.iterrows():
         print(row['webTitle'])
         print(row['webUrl'])
         print('****')
```

```
AI doomsday warnings a distraction from the danger it already poses, warns
expert
https://www.theguardian.com/technology/2023/oct/29/ai-doomsday-warnings-a-
distraction-from-the-danger-it-already-poses-warns-expert
****
AI watch: from deepfakes to a rock star humanoid
https://www.theguardian.com/technology/2023/jul/07/ai-watch-deepfakes-humanoid-
robot-artificial-intelligence
****
Instead of banning AI, schools should use it to enhance learning | Letters
https://www.theguardian.com/technology/2023/jul/09/instead-of-banning-ai-
schools-should-use-it-to-enhance-learning
****
The professor's great fear about AI? That it becomes the boss from hell
https://www.theguardian.com/technology/2023/aug/25/ai-artificial-intelligence-
michael-wooldridge-christmas-royal-institution-lectures
****
The existential threat from AI – and from humans misusing it | Letters
https://www.theguardian.com/technology/2023/jun/02/the-existential-threat-from-
ai-and-from-humans-misusing-it
****
```

1.11 Summary

There will be many other ways in which this kind of data can be explored, depending on the kind of question you might have. However the above techniques give us a good overview of the data including the time period covered, the top topics, the type of content that has been collected (news, sport, opinion etc.) and allows us to get a sense of some correlations of the topics.

1.12 Exercises

Explore your own data set from the Guardian API. Use the techniques above to get a better sense of what you've collected.