Week 8 assignment: NLP on social media data

Take our social media we collected last week and:

- extract the sentiment scores of the titles of the posts
  - you can use the keyword method, Python packages, or other methods to extract the sentiment scores
- plot a histogram of the sentiment scores
- look at descriptive statistics (mean, median, standard deviation) of the sentiment scores
- examine the text for some of the highest and lowest sentiment scores
- write a short analysis of the results and our process, as well as propose one idea for something we could use this data for

Optional advanced challenges:

- Compare different sentiment analysis methods (e.g. textblob and VADER). Does one seem to work better than another?
- Get the sentiments of the comments for each post. We can do a lot with this, such as:
  - look at the average sentiment for each post and compare it with the sentiment of the title and/or text
  - look at the distribution of sentiments for each post and find the posts with the widest range of sentiments (controversial posts)
- Examine the subjectivity of our data (e.g. using textblob)
- Use topic modeling on the posts
  - you can also add in the comments to the topic model
- Look at the most frequent words for positive and negative sentiment posts

Note: There is no assignment solution file for this week.

Import Social Media Data

In [1]:
```
#import sqlite3
#import pandas as pd

con = sqlite3.connect('../Week7/data/co_reddit.sqlite')
df = pd.read_sql_query('SELECT * from posts;', con)
con.close()
df
```

Out[1]:

	title	link	author
0	Wind power has gone from just an idea to one o...	/r/Colorado/comments/1dp2ozn/wind_power_has_go...	thecoloradosun
1	Last light turns dunes purple in Great Sand Du...	/r/Colorado/comments/1dp3y93/last_light_turns_...	_raidboss
2	Stunning sunset last night. Longmont, CO.	/r/Colorado/comments/1doxkkg/stunning_sunset_l...	razzledazzle125
3	Mysterious monolith appears in Northern Colorado	/r/Colorado/comments/1dpc62f/mysterious_monoli...	Knightbear49
4	From a hike in Woodland Park	/r/Colorado/comments/1dp66i5/from_a_hike_in_wo...	invincible789
...	...	...	...
637	Calling all CO musicians. What's been your exp...	/r/Colorado/comments/1b88yav/calling_all_co_mu...	J8R9L
638	Colorado grandmother awarded $3.76M after bung...	/r/Colorado/comments/1b7kc2k/colorado_grandmot...	nbcnews
639	State lawmakers introduce bill to reintroduce ...	/r/Colorado/comments/1b7ailg/state_lawmakers_i...	ButterscotchEmpty535
640	Congressional Candidate Now Supports a Nationa...	/r/Colorado/comments/1b7guy3/congressional_can...	Odd_Cranberry_8059
641	Red Rocks Through the	/r/Colorado/comments/1b7f2ix/red_rocks_through...	vegandread

	title	link	author
	Ruins on Mt. Falcon		

642 rows × 6 columns

Get Sentiment Score via the keyword method

In [2]:
```
sentiment_df = pd.read_csv('AFINN-en-165.txt', sep='\t', names=['word', 'score'], i
```

In [3]:
```
sentiment_df
```

Out[3]:

word	score
abandon	-2
abandoned	-2
abandons	-2
abducted	-2
abduction	-2
...	...
yucky	-2
yummy	3
zealot	-2
zealots	-2
zealous	2

3382 rows × 1 columns

In [4]:
```
sentiment_dict = sentiment_df.to_dict()['score']
```

Average sentiment for the title of each post

In [5]:
```
#import numpy as np

title_sentiments = []
for title in df['title']:
    words = title.lower().split()
    this_titles_sentiments = []
    for w in words:
        if w in sentiment_dict.keys():
            this_titles_sentiments.append(sentiment_dict[w])
```

```
        else:
            this_titles_sentiments.append(0)


    title_sentiments.append(np.mean(this_titles_sentiments))
```

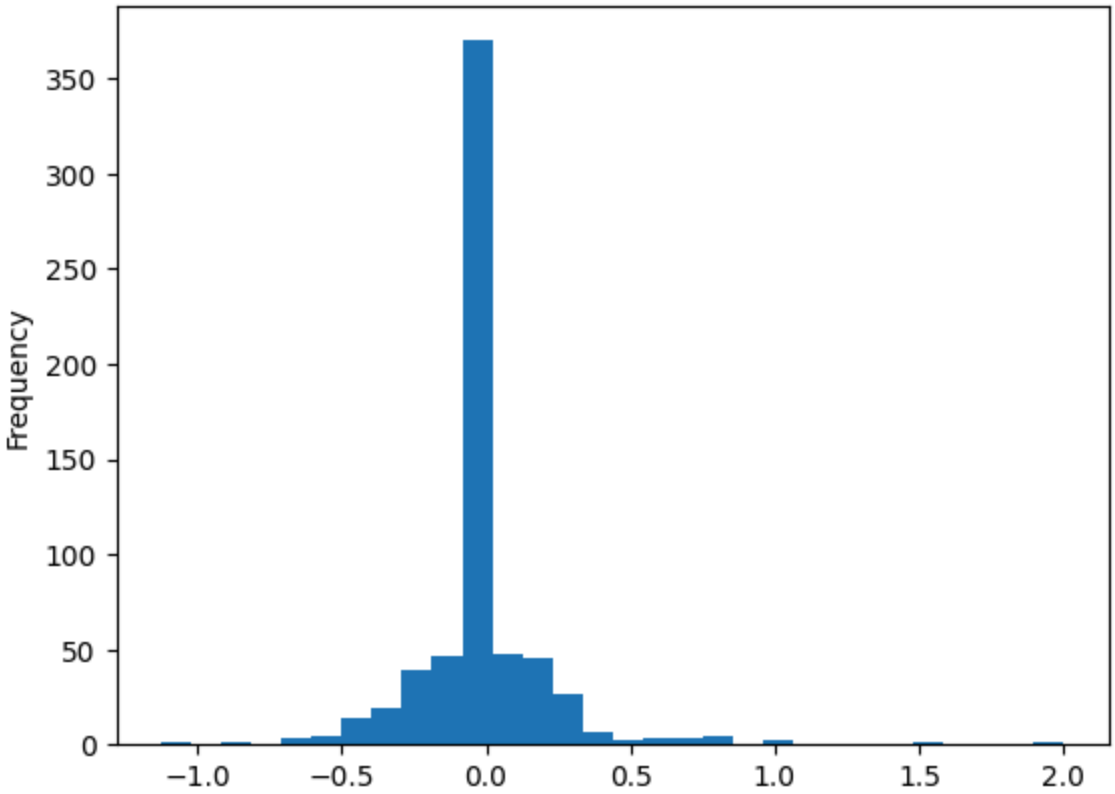
In [6]:
```
df['keyword_sentiment'] = title_sentiments
```

Histogram of the sentiment scores

In [7]:
```
#import matplotlib.pyplot as plt
```

In [8]:
```
df['keyword_sentiment'].plot.hist(bins=30)
```

Out[8]: <Axes: ylabel='Frequency'>



Look at descriptive statistics

In [10]:
```
df['keyword_sentiment'].mean()
```

Out[10]: -0.0014717687980526823

In [14]:
```
df['keyword_sentiment'].median()
```

Out[14]: 0.0

In [15]:
```
df['keyword_sentiment'].std()
```

Out[15]: 0.2236292582958655

Highest and lowest sentiment scores

In [11]:
```
df.sort_values(by='keyword_sentiment')[['title', 'keyword_sentiment']]
```

Out[11]:

	title	keyword_sentiment
432	Serial rapist with nationwide trail of victims...	-1.125000
368	Autopsy: Suzanne Morphew died by homicide	-0.833333
55	The shooter who killed 5 at a Colorado LGBTQ+ ...	-0.705882
93	Twin lakes fire	-0.666667
434	Colorado funeral home owners accused of storin...	-0.666667
...	...	...
586	Having fun in the snow	0.800000
401	Happy earth day!	1.000000
114	Peaceful morning	1.000000
206	Beautiful Boulder	1.500000
584	Evergreen	2.000000

642 rows × 2 columns

Lowest

In [12]:
```
df.sort_values(by='keyword_sentiment')['title'].to_list()[:10]
```

Out[12]:
```
['Serial rapist with nationwide trail of victims sentenced',
 'Autopsy: Suzanne Morphew died by homicide',
 'The shooter who killed 5 at a Colorado LGBTQ+ club pleads guilty to 50 federal h
ate crimes',
 'Twin lakes fire',
 'Colorado funeral home owners accused of storing 190 decaying bodies are charged
with Covid fraud',
 'Colorado man who sought revenge for a stolen phone pleads guilty to fire that ki
lled a Senegalese family of 5',
 'Bad Faith: The Narrowgate Cult',
 'Pueblo West man accused of threatening to “kill” young victim if they reported s
exual assaults',
 "A Colorado family's struggle with young woman's mental illness faces frightening
reality",
 'How bad is Colorado’s road rage compared to other states?']
```

Highest

In [13]:
```
df.sort_values(by='keyword_sentiment', ascending=False)['title'].to_list()[:10]
```

Out[13]:
```
['Evergreen',
 'Beautiful Boulder',
 'Happy earth day!',
 'Peaceful morning',
 'Biden wins Colorado Democratic primary',
 'Having fun in the snow',
 'DU hockey wins national championship',
 'Good Morning, Colorado Springs...',
 'Salida was beautiful tonight.',
 'Alamosa wins "Best Small Town Cultural Scene" award (USA Today)']
```

Summary

We started by importing social media data from a SQLite database into a Pandas DataFrame. Using the AFINN lexicon, we calculated sentiment scores for post titles based on their words. The average sentiment score was around -0.001, indicating a neutral sentiment overall. The median score was 0, showing a balance of positive and negative sentiments. The standard deviation was 0.224, meaning the scores didn't vary too much. We found that the most negative title was "Serial rapist with nationwide trail of victims sentenced" with a score of -1.125, while the most positive title was "Evergreen" with a score of 2.0.

We could use this data to track how community sentiment changes over time. By linking sentiment scores with dates, we can see how events like natural disasters or political news affect public mood. This could help us understand what types of content people respond to positively or negatively, helping content creators and social media managers engage better with their audience.