Data Analysis of Abortion-Related Tweets

Author: Malika Top

Collaborators: Remi Fernandez, Kai Barreras, Leanne Shen, Tiasa Kim, Luka Pearson

Date: December 22, 2022

Purpose: In this notebook, we upload the CSV file of annotated abortion-related tweets into pandas and attempt to create multiple visualizations and run a statistical test to explore the effects of the Dobbs decision on Twitter user sentiment

Table of Content

Installations and Setup

In [6]:
```
%%capture
pip install textblob
```

In [7]:
```
%%capture
pip install wordcloud
```

In [8]:
```
%%capture
pip install sklearn
```

In [9]:
```
%%capture
pip install nltk
```

In [10]:
```
%%capture
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
import json
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

Initial Exploration

Read in csv file into a dataframe

In [11]:
```
tweets = pd.read_csv('annotated_tweets_all.csv')
tweets.head()
```

Out[11]:

	created_at	id	text	pro-life	pro-choice	neutral	directive	informative	emotional	political	ideological	anecdotal	an
0	2022-04-16 19:43:44+00:00	1.515416e+18	@PPegasus843 @CurrentNewsGlo @nypost I hear yo...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0.0	
1	2022-04-19 03:52:58+00:00	1.516264e+18	Can't be a pro-choice Christian\nOr at least a...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	
2	2022-04-23 05:16:12+00:00	1.517734e+18	@PLPercussionist Oh, I'm sorry - make it impos...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0	1.0	
3	2022-04-29 15:52:20+00:00	1.520068e+18	Feminist Coffee Hour livestream for abortion f...	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1	0.0	
4	2022-04-26 04:33:53+00:00	1.518811e+18	@rachelkayw Abortion is the woman's choice, ...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0	0.0	

As we can see, the dataframe of annotated tweets has 1200 rows where each row is a tweet and 14 columns that include when it was created, the tweet ID, the tweet's text, and binary markers for various categories regarding sentiment and stance.

Noticing that the 1s and 0s are decimals, I want to see the type of each column.

In [12]:
```
col_type = tweets.dtypes
print('Data type of each column of Dataframe :')
print(col_type)
```

```
Data type of each column of Dataframe :
created_at      object
id             float64
text            object
pro-life       float64
pro-choice     float64
neutral        float64
directive      float64
informative    float64
emotional      float64
political      float64
ideological     object
anecdotal      float64
ambiguous      float64
irrelevant     float64
dtype: object
```

Pandas operations are likely better to work when the type is object rather than float, so I will convert here.

In [13]:
```
#df[:, 3:9] = df[:, 3:9].astype(str)
#tweets[['pro-life', 'pro-choice', 'neutral', 'directive',
        #'informative', 'emotional', 'political','anecdotal', 'ambiguous', 'irrelevant']] = tweets[['pro-life', 'pro-c
        #'informative', 'emotional', 'political','anecdotal', 'ambiguous', 'irrelevant']].astype(num)

tweets['ideological'] = pd.to_numeric(tweets['ideological'], errors='coerce')
```

In [14]:
```
col_type = tweets.dtypes
print('Data type of each column of Dataframe :')
print(col_type)
```

```
Data type of each column of Dataframe :
created_at      object
id             float64
text            object
pro-life       float64
pro-choice     float64
neutral        float64
directive      float64
informative    float64
emotional      float64
political      float64
ideological    float64
anecdotal      float64
ambiguous      float64
irrelevant     float64
dtype: object
```

Split the data frame into two time-dependent subsets

Originally, I wanted to analyze the data on a pre- and post- Dobbs decision level. However, with the Politico leak that occurred on May 3rd, there was a comparable spike in tweets as well. I figured that it would not make sense to compare pre- and post- overturning because of the leak was a pivotal moment itself.

Therefore, I am splitting the data into two time frames: April 1-May 31, 2022 and June 1-July 30, 2022. By splitting right down the middle, the first time frame includes the leak and the second time frame includes the actual official overturning.

Sort by descending, chrononological order:

In [15]:
```
tweets.sort_values(by='created_at')
```

Out[15]:

	created_at	id	text	pro-life	pro-choice	neutral	directive	informative	emotional	political	ideological	anecdot
297	2022-04-01 00:15:40+00:00	1.509686e+18	pro choice? pro life? bro i'm procreating with...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
370	2022-04-01 02:48:05+00:00	1.509724e+18	5 fetuses found inside DC home of anti-abortio...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.
311	2022-04-01 04:42:39+00:00	1.509753e+18	It is my most fervent wish that every anti-abo...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.
258	2022-04-01 10:47:35+00:00	1.509845e+18	The party of Pro Life strikes again! https://t...	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.
347	2022-04-01 12:10:49+00:00	1.509866e+18	Vote Pro-Life in the NI Assembly Election on 5...	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	.
1002	2022-07-29 10:52:28+00:00	1.552970e+18	@SkyNews @adamboultonTABB @KayBurley The peopl...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.
1180	2022-07-29 15:16:51+00:00	1.553037e+18	AOC rips Justice Alito for 'alarming' mockery ...	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.
1016	2022-07-29 16:59:04+00:00	1.553063e+18	Biden nominates abortion rights lawyer to be f...	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.
1115	2022-07-29 19:17:51+00:00	1.553097e+18	@darlene1waters @UltraSunshine22 @RobynPyeatt7...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.
1091	2022-07-30 03:45:52+00:00	1.553225e+18	As if not wanting to kill babies is a bad thin...	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.

1200 rows × 14 columns

For simplicity's sake, create a column that is just the date, rather than including the time

In [16]:
```
tweets['created_at']= pd.to_datetime(tweets['created_at']) #convert to datetime object
tweets['just_date'] = tweets['created_at'].dt.date #create new datetime object just by day, not time.
```

In [17]:
```
tweets = tweets.sort_values(by='created_at')
```

Count how many tweets there were for each day

In [18]:
```
dates = tweets['just_date']
freq_by_day = dates.value_counts(sort=False)
```

In [19]:
```
tweets['just_date'] = pd.to_datetime(tweets['just_date'])
tweets = tweets.set_index(tweets['just_date'])
tweets = tweets.sort_index()
```

In [20]:
```
tweets.head()
```

Out[20]:

just_date	created_at	id	text	pro-life	pro-choice	neutral	directive	informative	emotional	political	ideological	anecdotal	a
2022-04-01	2022-04-01 00:15:40+00:00	1.509686e+18	pro choice? pro life? bro i'm procreating with...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2022-04-01	2022-04-01 02:48:05+00:00	1.509724e+18	5 fetuses found inside DC home of anti-abortio...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	
2022-04-01	2022-04-01 04:42:39+00:00	1.509753e+18	It is my most fervent wish that every anti-abo...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
2022-04-01	2022-04-01 10:47:35+00:00	1.509845e+18	The party of Pro Life strikes again! https://t...	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	
2022-04-01	2022-04-01 12:10:49+00:00	1.509866e+18	Vote Pro-Life in the NI Assembly Election on 5...	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	

Create the two separate data frames by splicing

In [21]:
```
leak_frame = tweets['2022-04-01':'2022-05-31']
official_frame = tweets['2022-06-01':]
```

In [300…
```
leak_frame_sum = leak_frame.sum()
```

```
/var/folders/hr/k09c3cq53p74mq0_6x6r633m0000gn/T/ipykernel_11472/485784601.py:1: FutureWarning: Dropping of nuisance c
olumns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeErro
r.  Select only valid columns before calling the reduction.
```

Because the size of the data frames are different, we should calculate the proportion of tweets for a given category instead.

In [301…
```
to_add_leak = {'just_date':'', 'created_at':'Total', 'id':'',
               'text':'', 'pro-life':196.0, 'pro-choice':295.0,
               'neutral':230.0, 'directive':41.0, 'informative':131.0,
               'emotional':191.0, 'political':245.0, 'ideological':228.0,
               'anecdotal':228.0, 'ambiguous':27.0, 'irrelevant':164.0}


to_add_leak_prop = {'just_date':'', 'created_at':'proportion', 'id':'',
               'text':'', 'pro-life':196.0/753.0, 'pro-choice':295.0/753.0,
               'neutral':230.0/753.0, 'directive':41.0/753.0, 'informative':131.0/753.0,
               'emotional':191.0/753.0, 'political':245.0/753.0, 'ideological':228.0/753.0,
               'anecdotal':228.0/753.0, 'ambiguous':27.0/753.0, 'irrelevant':164.0/753.0}


leak_frame_final = leak_frame.append(to_add_leak, ignore_index=True)
leak_frame_final = leak_frame.append(to_add_leak_prop, ignore_index=True)
leak_frame_final
```

```
/var/folders/hr/k09c3cq53p74mq0_6x6r633m0000gn/T/ipykernel_11472/3647639388.py:14: FutureWarning: The frame.append met
hod is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
/var/folders/hr/k09c3cq53p74mq0_6x6r633m0000gn/T/ipykernel_11472/3647639388.py:15: FutureWarning: The frame.append met
hod is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
```

Out[301]:

	created_at	id	text	pro-life	pro-choice	neutral	directive	informative	emotional	political
0	2022-04-01 00:15:40+00:00	1509685896488050688.0	pro choice? pro life? bro i'm procreating with...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	2022-04-01 02:48:05+00:00	1509724250281807872.0	5 fetuses found inside DC home of anti-abortio...	0.000000	0.000000	1.000000	0.000000	1.000000	0.000000	0.000000
2	2022-04-01 04:42:39+00:00	1509753080908288000.0	It is my most fervent wish that every anti-abo...	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000
3	2022-04-01 10:47:35+00:00	1509844920533544704.0	The party of Pro Life strikes again! https://t...	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000
4	2022-04-01 12:10:49+00:00	1509865867353415680.0	Vote Pro-Life in the NI Assembly Election on 5...	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000
...	...	...	...	...	...	...	...	...	...	...
749	2022-05-31 07:04:04+00:00	1531531944021721088.0	@Sandernista412 Remember the GOP GIVES NO bene...	0.000000	1.000000	0.000000	1.000000	0.000000	1.000000	1.000000
750	2022-05-31 15:52:08+00:00	1531664837104787456.0	Democrats are the pro-life party. We were tryi...	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000
751	2022-05-31 17:57:11+00:00	1531696306837667840.0	@Regularguysmith @AussieAlex27 @Opsimath57 @re...	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000
752	2022-05-31 23:00:04+00:00	1531772529936932864.0	@JudiciaryGOP Like Roe?	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
753	proportion			0.260292	0.391766	0.305445	0.054449	0.173971	0.253652	0.325365

754 rows × 15 columns

In [302…
```
official_frame
official_frame_sum = official_frame.sum()
to_add_official = {'just_date':'', 'created_at':'Total', 'id':'',
               'text':'', 'pro-life':87.0, 'pro-choice':160.0,
               'neutral':179.0, 'directive':59.0, 'informative':126.0,
               'emotional':172.0, 'political':148.0, 'ideological':96.0,
               'anecdotal':31.0, 'ambiguous':42.0, 'irrelevant':35.0}

to_add_official_prop = {'just_date':'', 'created_at':'Total', 'id':'',
               'text':'', 'pro-life':87.0/447, 'pro-choice':160.0/447,
               'neutral':179.0/447, 'directive':59.0/447, 'informative':126.0/447,
               'emotional':172.0/447, 'political':148.0/447, 'ideological':96.0/447,
               'anecdotal':31.0/447, 'ambiguous':42.0/447, 'irrelevant':35.0/447}

official_frame_final = official_frame.append(to_add_official, ignore_index=True)
official_frame_final = official_frame.append(to_add_official_prop, ignore_index=True)
```

```
/var/folders/hr/k09c3cq53p74mq0_6x6r633m0000gn/T/ipykernel_11472/1647611844.py:2: FutureWarning: Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeErro
r.  Select only valid columns before calling the reduction.
/var/folders/hr/k09c3cq53p74mq0_6x6r633m0000gn/T/ipykernel_11472/1647611844.py:15: FutureWarning: The frame.append met
hod is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
/var/folders/hr/k09c3cq53p74mq0_6x6r633m0000gn/T/ipykernel_11472/1647611844.py:16: FutureWarning: The frame.append met
hod is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
```

In [303…
```
official_frame_final.shape
```

Out[303]:
```
(448, 15)
```

```
##########################################
```

3. Basic Tweet Preprocessing

In [116…
```
pip install tweet-preprocessor
```

```
Collecting tweet-preprocessor
  Downloading tweet_preprocessor-0.6.0-py3-none-any.whl (27 kB)
Installing collected packages: tweet-preprocessor
Successfully installed tweet-preprocessor-0.6.0
Note: you may need to restart the kernel to use updated packages.
```

a) Removing extraneous characters

The first step in preprocessing the tweets is to remove URLs, usernames, hashtags, punctuation, special characters, and numbers. In the same function, we normalize characters and turn them all lower case.

In [148…
```
'''
Removes URLS, usernames, and hashtags, punctuation,
special characters, numbers, and normalizes characters,
and lower-cases text
'''
def remove(text):
    text = re.sub(r'https?://[^ ]+', '', text)
    text= re.sub(r'@[^ ]+', '', text)
    text = re.sub(r'#', '', text)
    text = re.sub(r'([A-Za-z])\1{2,}', r'\1', text)
    text = re.sub(r' 0 ', 'zero', text)
    text = re.sub(r'[^A-Za-z ]', '', text)
    text = text.lower()
    return text
```

In [304…
```
leak_frame_final['text'] = leak_frame_final['text'].apply(remove)
```

In [305…
```
official_frame_final['text'] = official_frame_final['text'].apply(remove)
```

b) Tokenization, Lemmatization, Removal of Stop Words

Next, we want to tokenize, lemmatize, and remove stop words.

In [336…
```
%%capture
import nltk
from nltk import word_tokenize, FreqDist
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
nltk.download
nltk.download('wordnet')
nltk.download('stopwords')
from nltk.tokenize import TweetTokenizer
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/malitop105/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/malitop105/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [156…
```
'''
Lemmatize and tokenize text
'''
lemmatizer = nltk.stem.WordNetLemmatizer()
w_tokenizer = TweetTokenizer()
def lemmatize_text(text):
 return [(lemmatizer.lemmatize(w)) for w in \
                                    w_tokenizer.tokenize((text))]
```

In [306…
```
leak_frame_final['text'] = leak_frame_final['text'].apply(lemmatize_text)
```

In [307…
```
official_frame_final['text'] = official_frame_final['text'].apply(lemmatize_text)
```

Now remove stop words

In [308…
```
stop_words = set(stopwords.words('english'))
leak_frame_final['text'] = leak_frame_final['text'].apply(lambda x: [item for item in x if item not in stop_words])
official_frame_final['text'] = official_frame_final['text'].apply(lambda x: [item for item in x if item not in stop_wc
```

In [311…
```
from nltk.probability import FreqDist
fdist = FreqDist()
for tweet in leak_frame_final['text']:
    for word in tweet:
        fdist[word] += 1
fdist
```

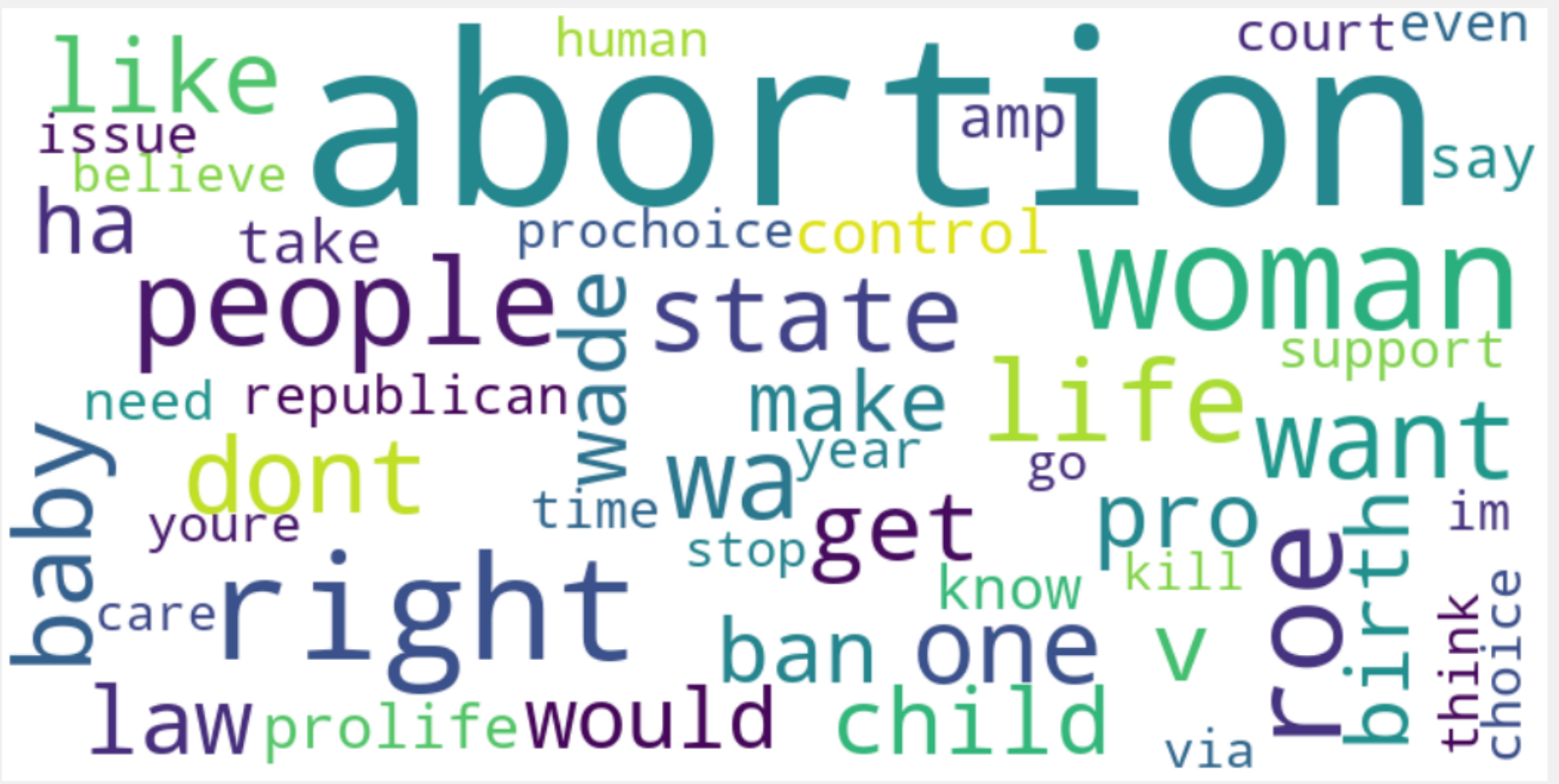
Out[311]:
```
FreqDist({'abortion': 558, 'woman': 132, 'right': 125, 'roe': 96, 'people': 85, 'wa': 78, 'life': 76, 'want': 73, 'ba
by': 69, 'state': 69, ...})
```

4. Making Visualizations

a) Word cloud for April-May 2022 (Politico Leak Period)

In [312…
```
from wordcloud import WordCloud

#stopwords = stop_words.extend(new_stop)
wc_leak = WordCloud(width=800, height=400, max_words=50, background_color="white", stopwords = stopwords).generate_fro
plt.figure(figsize=(12,10))
plt.imshow(wc_leak, interpolation="bilinear")
plt.axis("off")
plt.show()
```

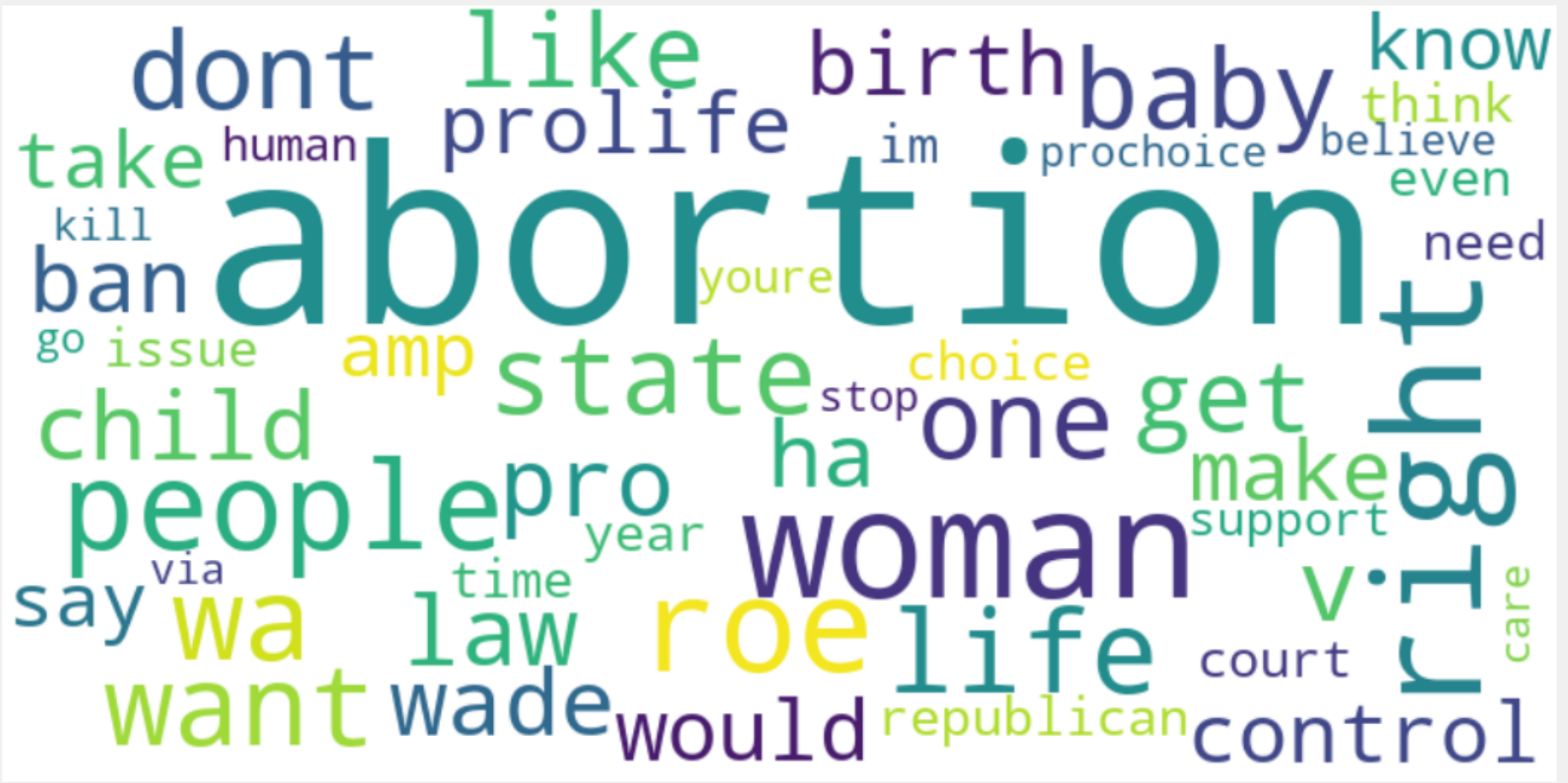


In [313…
```
fdist2 = FreqDist()
for tweet in official_frame_final['text']:
    for word in tweet:
        fdist2[word] += 1
fdist2
```

Out[313]:
```
FreqDist({'abortion': 320, 'woman': 92, 'roe': 92, 'right': 86, 'state': 62, 'v': 60, 'wade': 58, 'people': 52, 'don
t': 49, 'life': 48, ...})
```

b) Word cloud for June-July 2022 (Official Overturning Period)

In [314…
```
wc_official = WordCloud(width=800, height=400, max_words=50, background_color="white").generate_from_frequencies(fdist
plt.figure(figsize=(12,10))
plt.imshow(wc_official, interpolation="bilinear")
plt.axis("off")
plt.show()
```



In [317…
```
leak_frame_final['text'] = leak_frame_final['text'].astype(str)
```

In [318…
```
leak_frame_final['text']
```

Out[318]:
```
0      ['pro', 'choice', 'pro', 'life', 'bro', 'im', ...
1      ['fetus', 'found', 'inside', 'dc', 'home', 'an...
2      ['fervent', 'wish', 'every', 'antiabortion', '...
3                      ['party', 'pro', 'life', 'strike']
4      ['vote', 'prolife', 'ni', 'assembly', 'electio...
                             ...
749    ['remember', 'gop', 'give', 'benefit', 'grandp...
750    ['democrat', 'prolife', 'party', 'trying', 'sa...
751    ['um', 'root', 'society', 'problem', 'cant', '...
752                                    ['like', 'roe']
753                                                []
Name: text, Length: 754, dtype: object
```

In [27]:
```
import string
import re
import nltk
string.punctuation
```

Out[27]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [79]:
```
#defining the function to remove punctuation
def remove_punctuation(text):
    punctuationfree="".join([i for i in text if i not in string.punctuation])
    return punctuationfree

# remove punctuation, lowercase

leak_frame_final['clean_text']= leak_frame_final['text'].apply(lambda x:remove_punctuation(x))
leak_frame_final['clean_text']= leak_frame_final['clean_text'].apply(lambda x: x.lower())
```

In [88]:
```
import textblob
from textblob import TextBlob
```

In [198…
```
def listToString(s):

    # initialize an empty string
    str1 = " "

    # return string
    return (str1.join(s))
```

In [204…
```
leak_frame_final['str_text'] = leak_frame_final['text'].apply(listToString)

corpus_leak = leak_frame_final['str_text']

corpus_df = pd.DataFrame(corpus_leak)
corpus_df.columns = ['tweet']
#corpus_df
corpus_df['polarity'] = corpus_df['tweet'].apply(lambda x: TextBlob(x).polarity)
corpus_df['subjective'] = corpus_df['tweet'].apply(lambda x: TextBlob(x).subjectivity)

corpus_df
```

Out[204]:

	tweet	polarity	subjective
0	pro choice pro life bro im procreating ur mom	0.000000	0.000000
1	fetus found inside dc home antiabortion activist	0.000000	0.000000
2	fervent wish every antiabortion forced birth s...	-0.312500	0.562500
3	party pro life strike	0.000000	0.000000
4	vote prolife ni assembly election th may via	0.000000	0.000000
...	...	...	...
749	remember gop give benefit grandparent mother d...	-0.283333	0.483333
750	democrat prolife party trying save life yetunb...	-0.071429	0.214286
751	um root society problem cant mind business ins...	0.000000	0.000000
752	like roe	0.000000	0.000000
753		0.000000	0.000000

754 rows × 3 columns

In [337…
```
%%capture
pip install plotnine
```

In [232…
```
from pandas.api.types import CategoricalDtype
from plotnine import ggplot, aes, geom_line
from plotnine import *
from plotnine.data import mpg
%matplotlib inline
```

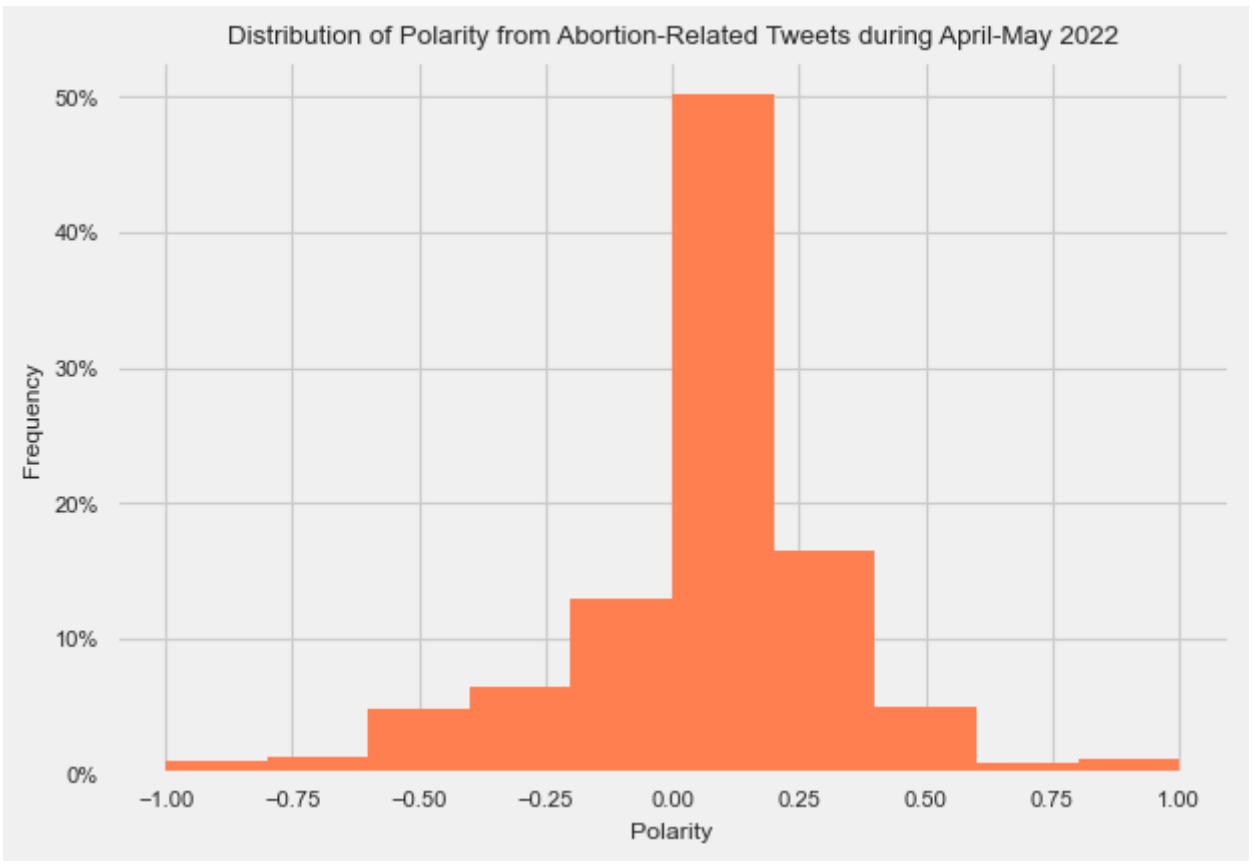
In [262…
```
from matplotlib.ticker import PercentFormatter
```

In [244…
```
plt.style.use('seaborn-paper')
```

In [263…
```
len(corpus_df['polarity'])
```

Out[263]:
```
754
```

In [276…
```
plt.hist(corpus_df['polarity'], color='coral', weights=np.ones(len(corpus_df['polarity'])) / len(corpus_df['polarity']
plt.gca().yaxis.set_major_formatter(PercentFormatter(1))
plt.xlabel("Polarity")
plt.ylabel("Frequency")
plt.title("Distribution of Polarity from Abortion-Related Tweets during April-May 2022")
plt.show()
```



In [287…
```
official_frame_final['str_text'] = official_frame_final['text'].apply(listToString)

corpus_off = official_frame_final['str_text']

corpus_df_off = pd.DataFrame(corpus_off)
corpus_df_off.columns = ['tweet']
#corpus_df
corpus_df_off['polarity'] = corpus_df_off['tweet'].apply(lambda x: TextBlob(x).polarity)
corpus_df_off['subjective'] = corpus_df_off['tweet'].apply(lambda x: TextBlob(x).subjectivity)

corpus_df_off
```

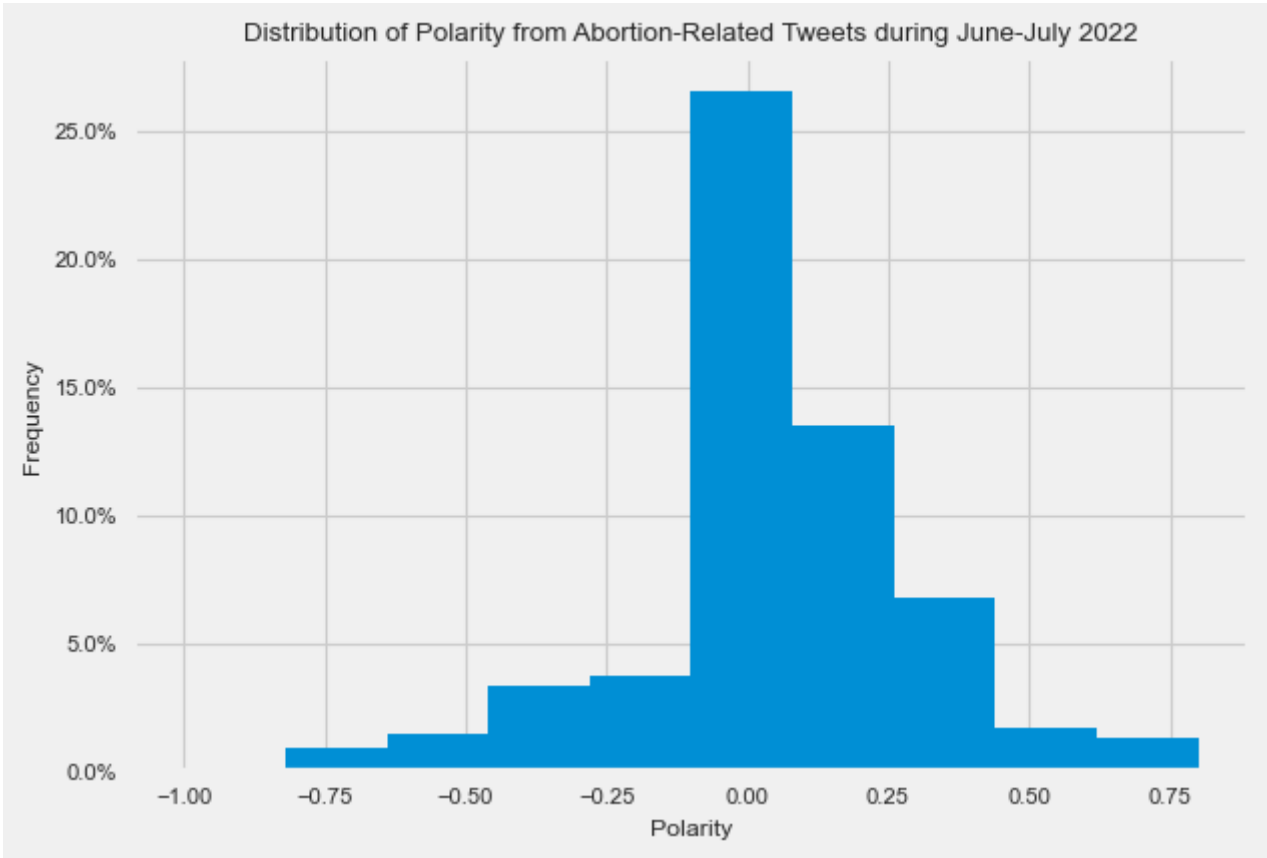
Out[287]:

	tweet	polarity	subjective
0	abortion ban dont work abortion becomes danger...	-0.095238	0.403571
1	disturbing libs dont care abortion mean reduci...	-0.190625	0.484375
2	must secure abortion right	0.342857	0.567857
3	hey marcos rich baby killed year thank local d...	0.058333	0.250000
4	abortion gun biggest case biggest culture war ...	0.000000	0.000000
...	...	...	...
443	aoc rip justice alito alarming mockery figure ...	-0.100000	0.600000
444	biden nominates abortion right lawyer federal ...	0.285714	0.535714
445	federal vaxx policy eliminates choice new fede...	-0.049784	0.508658
446	wanting kill baby bad thing	-0.700000	0.666667
447		0.000000	0.000000

448 rows × 3 columns

In [288…
```
plt.style.use('seaborn-paper')

plt.hist(corpus_df_off['polarity'], weights=np.ones(len(corpus_df_off['polarity'])) / len(corpus_df['polarity']))
plt.gca().yaxis.set_major_formatter(PercentFormatter(1))
plt.xlabel("Polarity")
plt.ylabel("Frequency")
plt.title("Distribution of Polarity from Abortion-Related Tweets during June-July 2022")
plt.show()
```



5. Statistical Tests

I want to run a series of multiple paired t-tests that compare pre- and post- Dobbs category frequencies and see if the difference is significant or not

In [320…
```
leak_df_test = leak_frame_final.iloc[:-1, :]
leak_df_test.tail()
```

Out[320]:

	created_at	id	text	pro-life	pro-choice	neutral	directive	informative	emotional	political	ideological	a
0	2022-04-01 00:15:40+00:00	1509685896488050688.0	['pro', 'choice', 'pro', 'life', 'bro', 'im', ...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	2022-04-01 02:48:05+00:00	1509724250281807872.0	['fetus', 'found', 'inside', 'dc', 'home', 'an...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	
2	2022-04-01 04:42:39+00:00	1509753080908288000.0	['fervent', 'wish', 'every', 'antiabortion', '...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	
3	2022-04-01 10:47:35+00:00	1509844920533544704.0	['party', 'pro', 'life', 'strike']	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	
4	2022-04-01 12:10:49+00:00	1509865867353415680.0	['vote', 'prolife', 'ni', 'assembly', 'electio...	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	
…	…	…	…	…	…	…	…	…	…	…	…	…
748	2022-05-30 19:12:01+00:00	1531352751925518336.0	['always', 'bring', 'abortion', 'conservative'...	1.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	
749	2022-05-31 07:04:04+00:00	1531531944021721088.0	['remember', 'gop', 'give', 'benefit', 'grandp...	0.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	
750	2022-05-31 15:52:08+00:00	1531664837104787456.0	['democrat', 'prolife', 'party', 'trying', 'sa...	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	
751	2022-05-31 17:57:11+00:00	1531696306837667840.0	['um', 'root', 'society', 'problem', 'cant', '...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	
752	2022-05-31 23:00:04+00:00	1531772529936932864.0	['like', 'roe']	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	

753 rows × 15 columns

In [327…
```
leak_df_test.isnull().sum()
```

Out[327]:
```
created_at      0
id              0
text            0
pro-life        2
pro-choice      2
neutral         2
directive       4
informative     6
emotional       3
political       5
ideological     6
anecdotal       5
ambiguous       4
irrelevant      5
just_date       0
dtype: int64
```

In [321…
```
official_df_test = official_frame_final.iloc[:-1, :]
official_df_test
```

Out[321]:

	created_at	id	text	pro-life	pro-choice	neutral	directive	informative	emotional	political	ideological	anec
0	2022-06-01 11:17:42+00:00	1531958162185523200.0	[abortion, ban, dont, work, abortion, becomes,...	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	
1	2022-06-01 18:23:06+00:00	1532065213885083648.0	[disturbing, libs, dont, care, abortion, mean,...	1.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	
2	2022-06-01 19:09:46+00:00	1532076958334345216.0	[must, secure, abortion, right]	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	
3	2022-06-02 03:07:39+00:00	1532197221461946368.0	[hey, marcos, rich, baby, killed, year, thank,...	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	
4	2022-06-02 10:20:02+00:00	1532306035070861312.0	[abortion, gun, biggest, case, biggest, cultur...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...
442	2022-07-29 10:52:28+00:00	1552970306238947328.0	[people, told, truth, wrong, prohibit, abortio...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	
443	2022-07-29 15:16:51+00:00	1553036842416480256.0	[aoc, rip, justice, alito, alarming, mockery, ...	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	
444	2022-07-29 16:59:04+00:00	1553062563218264064.0	[biden, nominates, abortion, right, lawyer, fe...	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	
445	2022-07-29 19:17:51+00:00	1553097488998969088.0	[federal, vaxx, policy, eliminates, choice, ne...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	
446	2022-07-30 03:45:52+00:00	1553225335629647872.0	[wanting, kill, baby, bad, thing]	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	

447 rows × 15 columns

In [326…
```
official_df_test.isnull().sum()
```

Out[326]:
```
created_at      0
id              0
text            0
pro-life        3
pro-choice      4
neutral         3
directive       3
informative     3
emotional       3
political       3
ideological     4
anecdotal       3
ambiguous       3
irrelevant      5
just_date       0
dtype: int64
```

Comparing pro-life and pro-choice for the month before and of the Politico leak vs the month of and after the Dobbs decision

Pair 1: Pro-life (Apr-May) vs Pro-life (Jun-Jul)

Null hypothesis: There is no difference between the frequency of pro-life tweets between April-May and June-July of 2022.

In [323…
```
from scipy import stats
```

In [324…
```
stats.ttest_rel(leak_df_test['pro-life'], official_df_test['pro-life'])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [324], in <cell line: 1>()
----> 1 stats.ttest_rel(leak_df_test['pro-life'], official_df_test['pro-life'])

File /Applications/miniconda3/lib/python3.9/site-packages/scipy/stats/_stats_py.py:6895, in ttest_rel(a, b, axis, nan_
policy, alternative)
   6893 nb = _get_len(b, axis, "second argument")
   6894 if na != nb:
-> 6895     raise ValueError('unequal length arrays')
   6897 if na == 0:
   6898     return _ttest_nans(a, b, axis, Ttest_relResult)

ValueError: unequal length arrays
```

I wanted to originally conduct a hypothesis test but there were issues with missing data and the fact that the sample sizes are not equal. April-May has 753 tweets and June-July has 447.

I then also attempted to run a chi-squared test to see if pro-life vs pro-choice are independent from emotional tweets

In [1]:
```
from scipy.stats import chi2_contingency
```

Pro-life association with emotional tweets during Politico leak

In [28]:
```
contingency = pd.crosstab(leak_frame['pro-life'], leak_frame['emotional'])
contingency
c, p, dof, expected = chi2_contingency(contingency)
p
```

Out[28]:
```
0.029054233036183
```

Pro-choice association with emotional tweets during Politico leak

In [29]:
```
contingency2 = pd.crosstab(leak_frame['pro-choice'], leak_frame['emotional'])
contingency2
c2, p2, dof2, expected2 = chi2_contingency(contingency2)
p2
```

Out[29]:
```
1.8892566601710388e-09
```

Pro-life association with emotional tweets during Dobbs period

In [27]:
```
contingency3 = pd.crosstab(official_frame['pro-life'], official_frame['emotional'])
contingency3
c3, p3, dof3, expected3 = chi2_contingency(contingency3)
p3
```

Out[27]:
```
0.001232875725321124
```

Pro-choice association with emotional tweets during Dobbs period

In [31]:
```
contingency4 = pd.crosstab(official_frame['pro-choice'], official_frame['emotional'])
contingency4
c4, p4, dof4, expected4 = chi2_contingency(contingency4)
p4
```

Out[31]:
```
3.0022898798889277e-07
```

The p-values for the Chi-squared tests comparing pro-choice and emotion are much, much lower than the p-values for comparing pro-life with emotions. We reject the null that there is no relationship between these categorical variables. These data suggest there is a statistically significant association between abortion stance and whether or not their tweet expresses emotion.