Before you turn this problem in, make sure everything runs as expected. First, restart the kernel (in the menubar, select Kernel$\rightarrow$Restart) and then run all cells (in the menubar, select Cell$\rightarrow$Run All).

Make sure you fill in any place that says `YOUR CODE HERE` or "YOUR ANSWER HERE", as well as your name and collaborators below:

```
NAME = "Riddhi Patel"
    COLLABORATORS = "Gautam Agarwal"
```

Midsemester Project : Twitter Data Analysis

Due Date : Thursday April 4th, 2024 by 11:59 PM

Completing this project

This is your mid-semester project. This is an individual project. Part 0 can only be completed if you have a paid Twitter account. We do not expect anyone to complete this part and it is TOTALLY optional. We left to show how to use API but it is NOT part of the project.

Project Purpose

The goal of this mid-semester project is to work with Twitter API to analyze tweets from a person, and in this case, Former President Donald Trump. @RealDonaldTrump tweets provide a great opportunity to understand how online media can be used to communicate over the traditional media. In fact, social media post are so influential, now the traditional media spends considerable amount of time discussing social media posts. Tweets from people like Donald Trump and Elon Musk have become so consequential, they can move the stock market on short term and get network TV to debate and discuss hours and hours about what Trump or Musk meant.

We hope this project will be fun as we can analyze range of emotions, hope, controversy, vagueness that are part of Trump tweets. We are interested in seeing what conclusions you can draw from former US Presidents tweets.

- DISCLAIMER: This project is not designed with any bias in mind. Note that we can pick any person (Hillary Clinton or Donald Trump or Elon Musk) or anyone else to do the same analysis. We hope your analysis is objective, independent of any political bias you may have. As Data Scientists, it is our responsiblity to do independent analysis of the data we try to understand. You should follow data and interpret insights w/o any bias.

Grading of the Project

You can test your project with the files provided. We may test the correctness of your code using different files. As a result, we will not provide sample outputs for this project. You will need to determine if the output received is reasonable. We are not looking for 100% compatibility with any one data set.

Set up

Let us get all the libaries initialized as necessary

```
# Run this cell to set up your notebook
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import zipfile
import json

# Ensure that Pandas shows at least 280 characters in columns, so we can see full
tweets
pd.set_option('max_colwidth', 280)

%matplotlib inline
plt.style.use('fivethirtyeight')
import seaborn as sns
sns.set()
sns.set_context("talk")
    import re
```

Downloading Tweets

Note: After Musk acquired twitter, the access to API is no longer free. A free API access can only do a few things. The discussion below assumes that one has access to a paid account.

It is important to download the most recent tweets (especially if you are working as a group). You cannot download the recent tweets by @realdonaldtrump as he was inactive for last two years. But you can download tweets from @elonmusk or @joebiden to see how things work. Those who are working by themselves are allowed to use the downloaded files in data folder w/o setting up access to any twitter API (which can sometime be bit complicated). Twitter provides the API Tweepy (http://www.tweepy.org/) that makes it easy to access twitter content that is publicly available. We will also provide example code as needed.

```
## Make sure you have set up tweepy if you are working locally.
# https://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python/
# After set up, the following should run:
    import tweepy
```

```
ERRORS --->
```

```
[0;31m---------------------------------------------------------------------------[0m
[0;31mModuleNotFoundError[0m                       Traceback (most recent call last)
Cell [0;32mIn[6], line 4[0m
```

```
[1;32m      1[0m [38;5;66;03m## Make sure you have set up tweepy if you are working
locally.[39;00m
[1;32m      2[0m [38;5;66;03m#
https://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python/[39;00m
[1;32m      3[0m [38;5;66;03m# After set up, the following should run:[39;00m
[0;32m----> 4[0m [38;5;28;01mimport[39;00m [38;5;21;01mtweepy[39;00m
```

```
[0;31mModuleNotFoundError[0m: No module named 'tweepy'
```

(NOT REQUIRED) PART 0: Accessing Twitter API (only for informational purposes)

This is optional, as this requires a paid twitter developer account.

In order to access Twitter API, you need to get keys by signing up as a Twitter developer. We will walk you through this process.

- if you are working by yourself on this project, you can skip PART 1, and complete the project using the data files provided in the data folder instead. We highly recommend that you do Part 1 as an individual (after completing the project with offline data). You will "learn" how to use Twitter API that might be useful for learning how to work with API's.

Task 0.1

Follow the instructions below to get your Twitter API keys. Read the instructions completely before starting.

1. Create a Twitter account. You can use an existing account if you have one; if you prefer to not do this assignment under your regular account, feel free to create a throw-away account.
2. Under account settings, add your phone number to the account.
3. Create a Twitter developer account by clicking the 'Apply' button on the top right of the page. Attach it to your Twitter account. You'll have to fill out a form describing what you want to do with the developer account. Explain that you are doing this for a class at Rutgers University and that you don't know exactly what you're building yet and just need the account to get started. These applications are approved by some sort of AI system, so it doesn't matter exactly what you write. Just don't enter a bunch of alweiofalwiuhflawiuehflawuihflaiwhfe type stuff or you might get rejected.
4. Once you're logged into your developer account, create an application for this assignment. You can call it whatever you want, and you can write any URL when it asks for a web site. You don't need to provide a callback URL.
5. On the page for that application, find your Consumer Key and Consumer Secret.
6. On the same page, create an Access Token. Record the resulting Access Token and Access Token Secret.
7. Edit the file keys.json and replace the placeholders with your keys.

WARNING (Please Read) !!!!

Protect your Twitter Keys

If someone has your authentication keys, they can access your Twitter account and post as you! So don't give them to anyone, and **don't write them down in this notebook**. The usual way to store sensitive information like this is to put it in a separate file and read it programmatically. That way, you can share the rest of your code without sharing your keys. That's why we're asking you to put your keys in `keys.json` for this assignment.

Avoid making too many API calls.

Twitter limits developers to a certain rate of requests for data. If you make too many requests in a short period of time, you'll have to wait awhile (around 15 minutes) before you can make more. So carefully follow the code examples you see and don't rerun cells without thinking. Instead, always save the data you've collected to a file. We've provided templates to help you do that.

Be careful about which functions you call!

This API can retweet tweets, follow and unfollow people, and modify your twitter settings. Be careful which functions you invoke! It is possible that you can accidentally re-tweet some tweets because you typed `retweet` instead of `retweet_count`.

Reading Keys.json

```
import json
key_file = 'keys.json'
# Loading your keys from keys.json (which you should have filled in question 1):
with open(key_file) as f:
    keys = json.load(f)
    # if you print or view the contents of keys be sure to delete the cell!
```

```
ERRORS --->

[0;31m---------------------------------------------------------------------------[0m
[0;31mFileNotFoundError[0m                         Traceback (most recent call last)
Cell [0;32mIn[6], line 4[0m
[1;32m      2[0m key_file [38;5;241m=[39m
[38;5;124m'[39m[38;5;124mkeys.json[39m[38;5;124m'[39m
[1;32m      3[0m [38;5;66;03m# Loading your keys from keys.json (which you should have
filled in question 1):[39;00m
[0;32m----> 4[0m [38;5;28;01mwith[39;00m
[38;5;28;43mopen[39;49m[43m([49m[43mkey_file[49m[43m)[49m [38;5;28;01mas[39;00m f:
[1;32m      5[0m     keys [38;5;241m=[39m json[38;5;241m.[39mload(f)
```

```
File
[0;32m/koko/system/anaconda3/envs/python310/lib/python3.10/site-packages/IPython/core/
interactiveshell.py:284[0m, in [0;36m_modified_open[0;34m(file, *args, **kwargs)[0m
[1;32m    277[0m [38;5;28;01mif[39;00m file [38;5;129;01min[39;00m {[38;5;241m0[39m,
[38;5;241m1[39m, [38;5;241m2[39m}:
[1;32m    278[0m     [38;5;28;01mraise[39;00m [38;5;167;01mValueError[39;00m(
[1;32m    279[0m         [38;5;124mf[39m[38;5;124m"[39m[38;5;124mIPython
won[39m[38;5;124m'[39m[38;5;124mt let you open
fd=[39m[38;5;132;01m{[39;00mfile[38;5;132;01m}[39;00m[38;5;124m by default
[39m[38;5;124m"[39m
[1;32m    280[0m         [38;5;124m"[39m[38;5;124mas it is likely to crash IPython. If
you know what you are doing, [39m[38;5;124m"[39m
[1;32m    281[0m         [38;5;124m"[39m[38;5;124myou can use
builtins[39m[38;5;124m'[39m[38;5;124m open.[39m[38;5;124m"[39m
[1;32m    282[0m     )
[0;32m--> 284[0m [38;5;28;01mreturn[39;00m
[43mio_open[49m[43m([49m[43mfile[49m[43m,[49m[43m
[49m[38;5;241;43m*[39;49m[43margs[49m[43m,[49m[43m
[49m[38;5;241;43m*[39;49m[38;5;241;43m*[39;49m[43mkwargs[49m[43m)[49m
```

```
    [0;31mFileNotFoundError[0m: [Errno 2] No such file or directory: 'keys.json'
```

Task 0.2 Testing Twitter Authentication

This following code should run w/o errors or warnings and display Rutgers University's twitter username

```
import tweepy
from tweepy import TweepyException
import logging

try:
    auth = tweepy.OAuthHandler(keys["consumer_key"], keys["consumer_secret"])
    redirect_url = auth.get_authorization_url()
    auth.set_access_token(keys["access_token"], keys["access_token_secret"])
    api = tweepy.API(auth)
    print("Rutgers username is:", api.get_user(screen_name="RutgersU").name)
except TweepyException as e:
    logging.warning("There was a Tweepy error. Double check your API keys and try
again.")
        logging.warning(e)
```

Getting more information from RutgersU

Find the following information about RutgersU. Show code and use a print statement to print the output.

```
# What is RutgersU screen name?

# What is the location RutgersU?
# What is a description for RutgersU?

# How many follow RutgersU?

# When was RutgersU account created?

    # Is RutgersU a verified account?
```

Task 0.3

Refactor and Extend Code

Re-factor the above twitter authentication code and extend the code into reusable snippets below.

```
def load_keys(path):
    """Loads your Twitter authentication keys from a file on disk.

    Args:
        path (str): The path to your key file.  The file should
          be in JSON format and look like this (but filled in):
            {
                "consumer_key": "<your Consumer Key here>",
                "consumer_secret":  "<your Consumer Secret here>",
                "access_token": "<your Access Token here>",
                "access_token_secret": "<your Access Token Secret here>"
            }

    Returns:
        dict: A dictionary mapping key names (like "consumer_key") to
          key values."""

    ### BEGIN ANSWER

    # your solution here

        ### END ANSWER


def download_recent_tweets_by_user(user_account_name, keys):
    """Downloads tweets by one Twitter user.

    Args:
        user_account_name (str): The name of the Twitter account
          whose tweets will be downloaded.
        keys (dict): A Python dictionary with Twitter authentication
          keys (strings), like this (but filled in):
            {
                "consumer_key": "<your Consumer Key here>",
                "consumer_secret":  "<your Consumer Secret here>",
                "access_token": "<your Access Token here>",
                "access_token_secret": "<your Access Token Secret here>"
            }

    Returns:
        list: A list of Dictonary objects, each representing one tweet."""
    import tweepy

   ### BEGIN ANSWER

    # your solution here

        ### END ANSWER


def load_tweets(path):
    """Loads tweets that have previously been saved.

    Calling load_tweets(path) after save_tweets(tweets, path)
    will produce the same list of tweets.

    Args:
        path (str): The place where the tweets were be saved.

    Returns:
        list: A list of Dictionary objects, each representing one tweet."""

    ### BEGIN ANSWER

    # your solution here

        ### END ANSWER


def get_tweets_with_cache(user_account_name, keys_path):
    """Get recent tweets from one user, loading from a disk cache if available.

    The first time you call this function, it will download tweets by
    a user.  Subsequent calls will not re-download the tweets; instead
    they'll load the tweets from a save file in your local filesystem.
    All this is done using the functions you defined in the previous cell.
    This has benefits and drawbacks that often appear when you cache data:

    +: Using this function will prevent extraneous usage of the Twitter API.
    +: You will get your data much faster after the first time it's called.
    -: If you really want to re-download the tweets (say, to get newer ones,
       or because you screwed up something in the previous cell and your
       tweets aren't what you wanted), you'll have to find the save file
       (which will look like <something>_recent_tweets.pkl) and delete it.

    Args:
        user_account_name (str): The Twitter handle of a user, without the @.
        keys_path (str): The path to a JSON keys file in your filesystem.
    """
    ### BEGIN ANSWER

    # your solution here

        ### END ANSWER
```

Task 0.4

If everything was implemented correctly you should be able to obtain roughly the last max number of tweets by @RutgersU. (This may take a few minutes)

```
# When you are done, run this cell to load latest @RutgersU 's tweets. This is to get
the latest tweets. Do not use the cached file
rutgers_tweets = download_recent_tweets_by_user("RutgersU", key_file)
    print("Number of tweets downloaded:", len(rutgers_tweets))
```

PART 1 - Working with Twitter Data

The json file in srv/shared folder contains some loaded tweets from @RutgersU and @realdonaldtrump. Run the folllowing code and read and understand and what it does. Groups must download the latest tweets from @RutgersU using tweepy (and call that). Individuals can use the given file.

Explore Rutgers tweets

```
from pathlib import Path
import json

ds_tweets_save_path = "/srv/shared/RutgersU_recent_tweets.json"   # file available
from /srv/shared

# Guarding against attempts to download the data multiple
# times:
if not Path(ds_tweets_save_path).is_file():
    # Getting as many recent tweets by @RutgersU as Twitter will let us have.
    # We use tweet_mode='extended' so that Twitter gives us full 280 character tweets.
    # This was a change introduced in September 2017.

    # The tweepy Cursor API actually returns "sophisticated" Status objects but we
    # will use the basic Python dictionaries stored in the _json field.
    example_tweets = [t._json for t in tweepy.Cursor(api.user_timeline,
screen_name="RutgersU",
                                                      tweet_mode='extended').items()]

    # Saving the tweets to a json file on disk for future analysis
    with open(ds_tweets_save_path, "w") as f:
        json.dump(example_tweets, f)

# Re-loading the json file:
with open(ds_tweets_save_path, "r") as f:
        example_tweets = json.load(f)
```

If things ran as expected, you should be able to look at the first tweet by running the code below. It probabably does not make sense to view all tweets in a notebook, as size of the tweets can freeze your browser (always a good idea to press ctrl-S to save the latest, in case you have to restart Jupyter)

```
# Looking at one tweet object, which has type Status:
from pprint import pprint # ...to get a more easily-readable view.
    pprint(example_tweets[0])
```

```
{'contributors': None,
'coordinates': None,
'created_at': 'Sat Nov 02 23:02:49 +0000 2019',
'display_text_range': [0, 140]
,
'entities': {'hashtags': [{'indices': [22, 31]
, 'text': 'internet'}],
'symbols': []
,
'urls': []
,
'user_mentions': [{'id': 955836661702774784,
'id_str': '955836661702774784',
'indices': [3, 13]
,
'name': 'Rutgers University-New Brunswick',
'screen_name': 'RutgersNB'},
{'id': 52517689,
'id_str': '52517689',
'indices': [42, 58]
,
'name': 'Rutgers School of Communication and '
'Information',
'screen_name': 'RutgersCommInfo'},
{'id': 392737670,
'id_str': '392737670',
'indices': [61, 72]
,
'name': 'Mary Chayko',
'screen_name': 'MaryChayko'}]
},
'favorite_count': 0,
'favorited': False,
'full_text': "RT @RutgersNB: As the #internet turns 50, @RutgersCommInfo's "
'@MaryChayko focuses on how we have used the innovation and what '
'it has meant f…',
'geo': None,
'id': 1190766238110236673,
'id_str': '1190766238110236673',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'place': None,
'retweet_count': 3,
'retweeted': False,
'retweeted_status': {'contributors': None,
'coordinates': None,
'created_at': 'Sat Nov 02 18:26:39 +0000 2019',
'display_text_range': [0, 191]
,
'entities': {'hashtags': [{'indices': [7, 16]
,
'text': 'internet'}]
,
'symbols': []
,
'urls': [{'display_url': 'nbcnews.com/think/opinion/…',
'expanded_url':
'https://www.nbcnews.com/think/opinion/what-50-years-spent-internet-worth-humanity-ncna1073656',
'indices': [168, 191]
,
'url': 'https://t.co/dmxjKYGmvU'}]
,
'user_mentions': [{'id': 52517689,
'id_str': '52517689',
'indices': [27, 43]
,
'name': 'Rutgers School '
'of '
'Communication '
'and Information',
'screen_name': 'RutgersCommInfo'},
{'id': 392737670,
'id_str': '392737670',
'indices': [46, 57]
,
'name': 'Mary Chayko',
'screen_name': 'MaryChayko'}]
},
'favorite_count': 6,
'favorited': False,
'full_text': 'As the #internet turns 50, '
"@RutgersCommInfo's @MaryChayko focuses on "
'how we have used the innovation and what '
'it has meant for the way we communicate '
'with each other.\n'
'\n'
```

```
'https://t.co/dmxjKYGmvU',
'geo': None,
'id': 1190696737868001280,
'id_str': '1190696737868001280',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'place': None,
'possibly_sensitive': False,
'retweet_count': 3,
'retweeted': False,
'source': '<a href="https://sproutsocial.com" '
'rel="nofollow">Sprout Social</a>',
'truncated': False,
'user': {'can_media_tag': True,
'contributors_enabled': False,
'created_at': 'Tue Jan 23 16:16:33 +0000 2018',
'default_profile': False,
'default_profile_image': False,
'description': 'The official Twitter account of '
'Rutgers University-New '
'Brunswick, the flagship home of '
'Rutgers, The State University '
'of New Jersey.',
'entities': {'description': {'urls': []
},
'url': {'urls': [{'display_url': 'newbrunswick.rutgers.edu',
'expanded_url': 'https://newbrunswick.rutgers.edu/',
'indices': [0,
23]
,
'url': 'https://t.co/Pll3p27ECO'}]
}},
'favourites_count': 1274,
'follow_request_sent': False,
'followed_by': False,
'followers_count': 2094,
'following': False,
'friends_count': 178,
'geo_enabled': True,
'has_extended_profile': False,
'id': 955836661702774784,
'id_str': '955836661702774784',
'is_translation_enabled': False,
'is_translator': False,
'lang': None,
'listed_count': 18,
'location': 'New Brunswick, NJ',
'name': 'Rutgers University-New Brunswick',
'notifications': False,
'profile_background_color': '000000',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_image_url_https':
'https://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_tile': False,
'profile_banner_url':
'https://pbs.twimg.com/profile_banners/955836661702774784/1564589696',
'profile_image_url':
'http://pbs.twimg.com/profile_images/1111272066609827848/9zSX0WEt_normal.png',
'profile_image_url_https':
'https://pbs.twimg.com/profile_images/1111272066609827848/9zSX0WEt_normal.png',
'profile_link_color': 'E81C4F',
'profile_sidebar_border_color': '000000',
'profile_sidebar_fill_color': '000000',
'profile_text_color': '000000',
'profile_use_background_image': False,
'protected': False,
'screen_name': 'RutgersNB',
'statuses_count': 1679,
'time_zone': None,
'translator_type': 'none',
'url': 'https://t.co/Pll3p27ECO',
'utc_offset': None,
'verified': False}},
'source': '<a href="http://twitter.com/download/iphone" '
'rel="nofollow">Twitter for iPhone</a>',
'truncated': False,
'user': {'can_media_tag': True,
'contributors_enabled': False,
'created_at': 'Wed Jan 21 02:57:47 +0000 2009',
'default_profile': False,
'default_profile_image': False,
'description': 'Rutgers, The State University of New Jersey, is a '
'leading public research university. Follow us for '
'all things Rutgers.',
'entities': {'description': {'urls': []
},
'url': {'urls': [{'display_url': 'rutgers.edu',
'expanded_url': 'http://www.rutgers.edu',
'indices': [0, 22]
,
'url': 'http://t.co/stAPJIzh8b'}]
}},
'favourites_count': 4590,
'follow_request_sent': False,
'followed_by': False,
'followers_count': 132798,
'following': False,
'friends_count': 595,
'geo_enabled': True,
'has_extended_profile': False,
'id': 19272796,
'id_str': '19272796',
'is_translation_enabled': False,
'is_translator': False,
'lang': None,
'listed_count': 838,
'location': 'New Jersey',
'name': 'Rutgers University',
'notifications': False,
'profile_background_color': 'C7141C',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme15/bg.png',
'profile_background_image_url_https':
'https://abs.twimg.com/images/themes/theme15/bg.png',
'profile_background_tile': False,
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/19272796/1494779773',
'profile_image_url':
'http://pbs.twimg.com/profile_images/809450270375772160/rWmyBIig_normal.jpg',
'profile_image_url_https':
'https://pbs.twimg.com/profile_images/809450270375772160/rWmyBIig_normal.jpg',
'profile_link_color': '0084B4',
'profile_sidebar_border_color': '000205',
'profile_sidebar_fill_color': 'C0DFEC',
'profile_text_color': '333333',
'profile_use_background_image': False,
'protected': False,
'screen_name': 'RutgersU',
'statuses_count': 16165,
'time_zone': None,
'translator_type': 'none',
'url': 'http://t.co/stAPJIzh8b',
'utc_offset': None,
    'verified': True}}
```

Task 1.1 - First 50 Rutgers Tweets

```
# print the first 50 rutgers tweets full_text (separated by a line)
counter = 1
for index,i in enumerate(example_tweets):
    print(counter)
    for key,value in i.items():
        if key == 'full_text':
            print(value)
    if index == 49:
        break
    counter+=1

        print('\n')
```

```
1
RT @RutgersNB: As the #internet turns 50, @RutgersCommInfo's @MaryChayko focuses on
how we have used the innovation and what it has meant f…


2
RT @RutgersNB: According to U.S. Rep. @FrankPallone, "continuous pharmaceutical
manufacturing is the future of medicine." Now, new legislat…
```

3
RT @RutgersLaw: The 34th Annual Mary Philbrook Public Interest Award Celebration honored Lloyd Freeman RLAW'07 (@Esquire1911), Partner and…

4
RT @prccrutgers: We will be holding our 4th Annual MLK Oratorical Competition in January 2020. You must write &amp; recite an original speech c…

5
Happy first day of November! Can you believe we are already more than halfway through the semester? 🍂

📷: @curtiswebsterr on Instagram https://t.co/8tOBOjvVzl

6
RT @RUFedRelations: @EnergyCommerce @FrankPallone @ContinuousMFG @RutgersU To learn more about @FrankPallone's new pharmaceutical manufactu…

7
Daniel Hayden went back to school to pursue a physics degree and found a new home in the @RUMakerspace where he was able to deepen his love of music by building his own guitar.

Read his story: https://t.co/sQGqQsLBjZ

#RutgersImpact

8
RT @RutgersU_News: Discussing politics – especially with someone who doesn’t agree with you – can be challenging. But perhaps we can have a…

9
RT @RUFedRelations: Thank you to @EnergyCommerce Chairman @FrankPallone for introducing his new bill to create university centers of excell…

10
RT @sssrutgers: We are just ONE week away from our National F1rst Gen Day Celebration! Stay tuned for what TRIO (@ru_ububms @runbmcnair) ha…

11
RT @RUStudentHealth: #RUReadyForFluSeason? If you said no, here are some tips to protect yourself, before you wreck yourself. Still need to…

12

RT @RUAthletics: Join Us on #CFB150 Anniversary Day - Wednesday Nov. 6 -for 2 Special Events open to the public:

2 PM: Ringing of the 🔔@ O…

13
RT @MasonGross: "Art gives us the opportunity to step back and see things that are happening around us. Without it, I don’t think we’d be a…

14
Take away Dracula’s black satin cape, razor-sharp fangs and insatiable thirst for blood, and what do you have?

A fiercely fascinating human being.

Read about a Rutgers Winter Session course examining the real-life royal behind the vampire myth: https://t.co/3ipJNnW6gJ https://t.co/SwYGlHvFQf

15
RT @RUWSoccer: Congratulations to Amanda Visco on Second Team All-Big Ten honors! Visco anchored a back line that posted 11 shutouts, while…

16
Check out how @RU_nursing gets into the Halloween spirit with their annual Hospital of Horrors training scenarios.

#SpookySZN #RutgersImpact https://t.co/71t0bScJg5

17
RT @RUWSoccer: Congratulations to Amirah Ali on her fourth All-Big Ten honor and second straight First Team nod! Ali posted nine goals, fou…

18
RT @RUWSoccer: Congratulations to Amirah Ali, Taylor Aylmer, Meagan McClelland, Nneka Moneme, Chantelle Swaby, and Amanda Visco on earning…

19
@fred1313 @RUAthletics We love it! https://t.co/ha77PA2y7Q

20
RT @RutgersU_News: Sssscared of snakes? @RutgersSASN professor @vanessalobue talks about why snakes are a source of disgust and fear in thi…

21
RT @RutgersSPH: Meet our #APHA2019 social media ambassadors: @denise_mulbah, @LauraJeanBruce, and @calebloschiavo 👉 https://t.co/f1fVxjfR7w…

22

RT @RutgersWBB: 🎃👻😱 Nothing to fear... Halloween means #RHoops is back in just five days! https://t.co/SvU0VWSmco

23

RT @RutgersWGolf: Checking the wind, yardage, and candy selection 🎃👻

Happy Halloween from #RUWGolf! https://t.co/prE3XaxqHG

24

RT @RutgersU_News: Cycling is safer these days, but injuries are on the rise among older riders, according to a study by led by Corina Din-…

25

RT @RUHonorsCollNB: Professor and Faculty Fellow in Residence @MaryChayko has published an editorial on @NBCNewsThink on the importance of…

26

Happy Halloween, Rutgers ghouls, ghosts and goblins!

To celebrate the spookiest day of the year, read about some of our favorite Rutgers haunts: https://t.co/PNzxwVEsyD

#SpookySZN https://t.co/ifghNlJS0j

27

RT @Rutgers_IPO: Five new pictures have been added for our Portrait Project!

The Portrait Project is a way for IP&amp;O to show appreciation f…

28

Halloween is tomorrow but should you wear a costume to the office? @RU_SMLR's Jessica Methot weighs the tricks and treats of bringing #Halloween celebrations to work.

Read more: https://t.co/ySpKrZtjpi

29

Learn how the new Rutgers Optimizes Innovation Program will speed up the process of turning biomedical discoveries into market-ready solutions and train the next generation of innovators: https://t.co/iGyf9GBn6u

#RutgersResearch

30

RT @RFootball: https://t.co/I6R50vB9zL

#***TheBirthplace*** | @Cole_Murphy_10

31
The Rutgers Jewish Film Festival will celebrate its 20th year when it opens on November 3. Get to know Rutgers alumna Sharon Karmazin, who proposed the festival more than two decades ago, and learn about the events planned for this year: https://t.co/FXMZOmGX62

#RutgersPride

32
RT @Rutgers_IPO: Superheroes! @Rutgers_PD presents a donation to @RutgersCancer to help raise money for breast cancer awareness! https://t.…

33
RT @RutgersU_News: After launching an interdisciplinary master's of forensic science, the first of its kind in NJ, @Rutgers_Camden’s Kimber…

34
RT @RutgersMBB: .@CoachPikiell teamed up with @CoachGoodale &amp; @CoachTimRU to hand out 🆓 @adidasUS slides to some lucky students today at @t…

35
Believe it or not, there are over half a million living #RutgersAlumni. Where did you call home after your time here at Rutgers?

#RutgersPride https://t.co/s5UfejKAri

36
Read the larger than life story of George H. Large, a member of the 1869 @RFootball team, as remembered by his great-granddaughter Catherine Wetstein: https://t.co/VstKeYTmou

#RutgersPride #TheBirthplace

37
On Wednesday, October 30, the @Rutgers_Camden Honors College will be hosting a TED talk-style presentation exploring the cultural implications and roots of #Halloween.

Get a deeper understanding of this American tradition: https://t.co/TP8OglWjyr https://t.co/GvNtgKKaeK

38
RT @RULibraries: 🙀 #NationalCatDay https://t.co/4mHEhtzvSo

39

RT @RUdental: Who are the dentists and other oral healthcare providers of tomorrow? Our students. Read all about them in RSDM's annual repo…

40
RT @RutgersNSO: It’s that time of year again! We are currently recruiting Orientation Leaders for summer 2020. We will be tabling the next…

41
RT @Rutgers_Camden: Worried about all the sugar in that Halloween candy? Prof. @Char_Markey shares some perspective on holiday treats with…

42
Meet three @SHP_Rutgers students whose commitment to helping people in underserved communities get access to the health care they need earned them prestigious scholarships from the National Health Service Corps: https://t.co/XwM4fiQua7

#RutgersImpact

43
Did you see @RutgersNB vice chancellor for enrollment management Courtney McAnuff on @HodaAndJenna last week?

Find out what tips he had to share for navigating the college admissions process: https://t.co/qV2Uel7iBS

#RutgersPride

44
RT @MasonGross: What a night @NewBrunswickPAC Friday: @RU_Film screened "Good Time" starring Robert Pattinson and hosted co-directors Josh…

45
Read about the impact @RutgersGSE had on alumna Meghan Stratton, who was named the 2019 New Jersey History Teacher of the Year: https://t.co/IHeyFm9Xw9

#RutgersPride https://t.co/VckOxZddds

46
RT @MasonGross: Stop by the Mason Gross Galleries at Civic Square, 33 Livingston Ave., CAC, for our undergrad exhibit, "Visions For the Fut…

47
RT @RutgersNJAES: Rutgers Researcher Discusses the Health Benefits of Cranberries. In 1999 Nicholi Vorsa &amp; Amy Howell are the first to docu…

48

```
RT @RUCSRR: Join us for our lecture series "Islam and the Humanities" on 11/7 with
Professor Sinan Antoon. @RutgersLaw @Rutgers_Newark @Wak…


49
RT @RutgersSEBS: Cook Community Alumni Association announces George Hammell Cook
Distinguished Alumni Award Recipients #CookAlumni #Rutgers…


50
As we count down to Halloween, learn about a Rutgers class where students examine
horror films based on the works of Stephen King through the lens of psychiatry.

    https://t.co/LIHNgqzwwt
```

Task 1.2 - First 50 Trump Tweets

To be consistent we are going to use the same dataset no matter what you get from your twitter api. So from this point on, if you are working as a group or individually, be sure to use the data sets provided to you in the data folder. One of the files is 'TrumpTweets_1.json', the other one is 'TrumpTweets_2.json'. First load TrumpTweets_1.

```python
def load_tweets(path):
    """Loads tweets that have previously been saved.

    Calling load_tweets(path) after save_tweets(tweets, path)
    will produce the same list of tweets.

    Args:
        path (str): The place where the tweets will be saved.

    Returns:
        list: A list of Dictionary objects, each representing one tweet."""

    with open(path, "rb") as f:
        import json
            return json.load(f)


dest_path = "/srv/shared/TrumpTweets_1.json"
    trump_tweets = load_tweets(dest_path)


# print the first 10 Trump tweets full_text only
for index, i in enumerate(trump_tweets):
    for key,value in i.items():
        if key == 'text':
            print(value)
            print('\n')
    if index == 9:
            break
```

```
Will be leaving Florida for Washington (D.C.) today at 4:00 P.M. Much work to be done,
but it will be a great New Year!


Iran is failing at every level despite the terrible deal made with them by the Obama
Administration. The great Iranian people have been repressed for many years. They are
hungry for food &amp; for freedom. Along with human rights, the wealth of Iran is
being looted. TIME FOR CHANGE!


The United States has foolishly given Pakistan more than 33 billion dollars in aid
over the last 15 years, and they have given us nothing but lies &amp; deceit, thinking
of our leaders as fools. They give safe haven to the terrorists we hunt in
Afghanistan, with little help. No more!


HAPPY NEW YEAR! We are MAKING AMERICA GREAT AGAIN, and much faster than anyone thought
possible!


As our Country rapidly grows stronger and smarter, I want to wish all of my friends,
supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and
Healthy New Year. 2018 will be a great year for America!


Iran, the Number One State of Sponsored Terror with numerous violations of Human
Rights occurring on an hourly basis, has now closed down the Internet so that peaceful
demonstrators cannot communicate. Not good!


What a year it’s been, and we're just getting started. Together, we are MAKING AMERICA
GREAT AGAIN! Happy New Year!! https://t.co/qsMNyN1UJG


My deepest condolences to the victims of the terrible shooting in Douglas County
@DCSheriff, and their families. We love our police and law enforcement - God Bless
them all! #LESM


Why would smart voters want to put Democrats in Congress in 2018 Election when their
policies will totally kill the great wealth created during the months since the
Election. People are much better off now not to mention ISIS, VA, Judges, Strong
Border, 2nd A, Tax Cuts &amp; more?
```

```
If the Dems (Crooked Hillary) got elected, your stocks would be down 50% from
values on Election Day. Now they have a great future - and just beginning!
https://t.co/9TzSC8F8vY
```

Task 1.3 - Oldest Tweet

Find the number of the month of the oldest tweet.

```
# Find the number of the month of the oldest tweet (e.g. 1 for January)
trump_tweets = pd.DataFrame(trump_tweets)


### BEGIN
def oldest_tweet(df):
    #idea: the smallest date implies the oldest.
    df['created_at'] = pd.to_datetime(df['created_at'])
    min_date = df['created_at'].min().month

    return min_date
### END ANSWER

oldest_month = oldest_tweet(trump_tweets)
    print(oldest_month)
```

```
    1
```

PART 2 Twitter Source Analysis (group/individual)

Task 2.1 - Create and Merge

Create and Merge two dataframes created from TrumpTweets_1 and TrumpTweets_2. Call this new dataframe all_tweets. Please check to make sure files are compatible.

```
### YOUR ANSWER
tw1 = '/srv/shared/TrumpTweets_1.json'
trump1path = load_tweets(tw1)
tw2 = '/srv/shared/TrumpTweets_2.json'
trump2path = load_tweets(tw2)


TrumpTweets_1 = pd.DataFrame(trump1path)
#TrumpTweets_1
TrumpTweets_2 = pd.DataFrame(trump2path)

test = [TrumpTweets_1,TrumpTweets_2]
all_tweets = pd.concat(test)
######
all_tweets.head()
    #all_tweets.info()
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

source	id	text	created_at	retweet_count	in_reply_to_user_id_str	favorite_count	is_retweet	id_str	full_text	...	quoted_status_id	quoted_status_id_str	quoted_status_permalink	quoted_status	favorited	retweeted	possibly_sensitive	lang	extended_entities	retweeted_status

0	Twitter for iPhone	947824196909961216	Will be leaving Florida for Washington (D.C.) today at 4:00 P.M. Much work to be done, but it will be a great New Year!	Mon Jan 01 13:37:52 +0000 2018	8237	None	51473	False	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1	Twitter for iPhone	947810806430826496	Iran is failing at every level despite the terrible deal made with them by the Obama Administration. The great Iranian people have been repressed for many years.	Mon Jan 01 12:44:40 +0000 2018	14595	25073877	53557	False	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

They are hungry for food & for freedom. Along with human rights, the wealth of Iran is being looted. TIME FOR ...

2	Twitter for iPhone	947802588174577664	The United States has foolishly given Pakistan more than 33 billion dollars in aid over the last 15 years, and they have given us nothing but lies & deceit, thinking of	Mon Jan 01 12:12:00 +0000 2018	49566	None	138808	False	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

our leaders as fools. They give safe haven to the terrorists we hunt in Afghanistan, with little help. ...

3	Twitter for iPhone	947614110082043904	HAPPY NEW YEAR! We are MAKING AMERICA GREAT AGAIN, and much faster than anyone thought possible!	Sun Dec 31 23:43:04 +0000 2017	35164	None	154769	False	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

4	Twitter for iPhone	947592785519173637	As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News	Sun Dec 31 22:18:20 +0000 2017	39428	None	1576755	False	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Media, a Happy and Healthy New Year. 2018 will be a great year for America!

5 rows × 33 columns

```
               source                  id  \
0  Twitter for iPhone  947824196909961216
1  Twitter for iPhone  947810806430826496
2  Twitter for iPhone  947802588174577664
3  Twitter for iPhone  947614110082043904
4  Twitter for iPhone  947592785519173637


text  \
0
Will be leaving Florida for Washington (D.C.) today at 4:00 P.M. Much work to be done,
but it will be a great New Year!
1  Iran is failing at every level despite the terrible deal made with them by the
Obama Administration. The great Iranian people have been repressed for many years.
They are hungry for food &amp; for freedom. Along with human rights, the wealth of
Iran is being looted. TIME FOR ...
2  The United States has foolishly given Pakistan more than 33 billion dollars in aid
over the last 15 years, and they have given us nothing but lies &amp; deceit, thinking
of our leaders as fools. They give safe haven to the terrorists we hunt in
Afghanistan, with little help. ...
3
HAPPY NEW YEAR! We are MAKING AMERICA GREAT AGAIN, and much faster than anyone thought
possible!
```

```
4                                                      As our Country rapidly grows
stronger and smarter, I want to wish all of my friends, supporters, enemies, haters,
and even the very dishonest Fake News Media, a Happy and Healthy New Year. 2018 will
be a great year for America!

                       created_at  retweet_count in_reply_to_user_id_str  \
0  Mon Jan 01 13:37:52 +0000 2018           8237                    None
1  Mon Jan 01 12:44:40 +0000 2018          14595                25073877
2  Mon Jan 01 12:12:00 +0000 2018          49566                    None
3  Sun Dec 31 23:43:04 +0000 2017          35164                    None
4  Sun Dec 31 22:18:20 +0000 2017          39428                    None

   favorite_count is_retweet id_str full_text  ... quoted_status_id  \
0           51473      False    NaN       NaN  ...              NaN
1           53557      False    NaN       NaN  ...              NaN
2          138808      False    NaN       NaN  ...              NaN
3          154769      False    NaN       NaN  ...              NaN
4          157655      False    NaN       NaN  ...              NaN

  quoted_status_id_str quoted_status_permalink  quoted_status favorited  \
0                  NaN                     NaN            NaN       NaN
1                  NaN                     NaN            NaN       NaN
2                  NaN                     NaN            NaN       NaN
3                  NaN                     NaN            NaN       NaN
4                  NaN                     NaN            NaN       NaN

   retweeted possibly_sensitive lang extended_entities retweeted_status
0        NaN                NaN  NaN               NaN              NaN
1        NaN                NaN  NaN               NaN              NaN
2        NaN                NaN  NaN               NaN              NaN
3        NaN                NaN  NaN               NaN              NaN
4        NaN                NaN  NaN               NaN              NaN
```

```
[5 rows x 33 columns]
```

Task 2.2

Construct a DataFrame called `df_trump` containing all the tweets stored in `all_tweets`.

Important: There may/will be some overlap so be sure to eliminate duplicate tweets. If you do not eliminate the duplicates properly, your results might not be compatible with the test solution. Hint: the `id` of a tweet is always unique.

The index of the dataframe should be the ID of each tweet (looks something like `907698529606541312`). df_trump should have these columns:

- `time`: The time the tweet was created encoded as a datetime object. (Use `pd.to_datetime` to encode the timestamp.)
- `source`: The source device of the tweet.

- text: The text of the tweet.
- retweet_count: The retweet count of the tweet.
- favorite_count: The favorite count of the tweet.

Finally, the resulting dataframe should be sorted by date/time.

Warning: *Some tweets may store the text in the* text *field and other will use the* full_text *field.*

```
# Sort daaframe by date/time (earliet tweet first)

### BEGIN

# Construct the df_trump dataframe, get the columns needed and set the index to id:

#may have possibly the same id, so drop those duplicates.
all_tweets['id'] = all_tweets['id'].astype(int)
all_tweets = all_tweets.drop_duplicates(subset=['id'])

df_trump =
all_tweets[['source','text','full_text','retweet_count','favorite_count']].copy()
df_trump['time'] = pd.to_datetime(all_tweets['created_at'].copy())
df_trump.set_index(all_tweets['id'].copy(), inplace=True)



# Merge the text and full_text to become one:
df_trump['text'] = df_trump['text'].combine_first(df_trump['full_text'])
df_trump = df_trump.drop('full_text', axis=1)

# Sort by date-time now
df_trump = df_trump.sort_values('time', ascending=True)
df_trump
    ### END ANSWER
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	source	text	retweet_count	favorite_count	time
id					
---	---	---	---	---	---

682723973449289728	Twitter for Android	I will be on @FoxNews live, with members of my family, at 11:50 P.M. We will ring in the New Year together! MAKE AMERICA GREAT AGAIN!	2108	6735	2016-01-01 00:44:14+00:00
682764544402440192	Twitter for iPhone	HAPPY NEW YEAR & THANK YOU! https://t.co/YO1Yi8QbZy https://t.co/uxUXWJ1Rbv	3460	8581	2016-01-01 03:25:27+00:00
682792967736848385	Twitter for iPhone	#HappyNewYear America! https://t.co/EeQb8PDrUe	3434	9143	2016-01-01 05:18:23+00:00
682805320217980929	Twitter for iPhone	Happy New Year from #MarALago! Thank you to my great family for all of their support. https://t.co/6UsqSiaaj7	1948	8258	2016-01-01 06:07:28+00:00
682805477168779264	Twitter for Android	"@jallenaip: Hillary said she was in a "Fog of War" as explanation for the lies about Benghazi. No fog allowed in WH. Vote Trump POTUS!"	2721	7490	2016-01-01 06:08:06+00:00

...	...	...	...	...	...
1052213711295930368	Twitter for iPhone	“Federal Judge throws out Stormy Danials lawsuit versus Trump. Trump is entitled to full legal fees.” @FoxNews Great, now I can go after Horseface and her 3rd rate lawyer in the Great State of Texas. She will confirm the letter she signed! She knows nothing about me, a total ...	14594	54635	2018-10-16 15:04:32+00:00
1052217314463100928	Twitter for iPhone	“Conflict between Glen Simpson’s testimony to another House Panel about his contact with Justice Department official Bruce Ohr. Ohr was used by Simpson and Steele as a Back Channel to get (FAKE) Dossier to FBI. Simpson pleading Fifth.” Catherine Herridge. Where is Jeff Sessions?	6271	20251	2018-10-16 15:18:51+00:00

1052219253384994816	Twitter for iPhone	Is it really possible that Bruce Ohr, whose wife Nellie was paid by Simpson and GPS Fusion for work done on the Fake Dossier, and who was used as a Pawn in this whole SCAM (WITCH HUNT), is still working for the Department of Justice????? Can this really be so?????	13103	41253	2018-10-16 15:26:33+00:00
1052232230972678145	Twitter for iPhone	RT @WhiteHouse: https://t.co/RNqLpOtS3O	4478	0	2018-10-16 16:18:08+00:00
1052233253040640001	Twitter for iPhone	REGISTER TO https://t.co/0pWiwCHGbh! #MAGAus https://t.co/ACTMe53TZU	5415	16565	2018-10-16 16:22:11+00:00

9478 rows × 5 columns

```
source  \
id
682723973449289728
Twitter for Android
682764544402440192
Twitter for iPhone
682792967736848385
Twitter for iPhone
682805320217980929
Twitter for iPhone
```

```
682805477168779264
Twitter for Android
...
...
1052213711295930368  <a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>
1052217314463100928  <a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>
1052219253384994816  <a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>
1052232230972678145  <a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>
1052233253040640001  <a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>
```

```
text  \
id
682723973449289728
I will be on @FoxNews live,  with members of my family, at 11:50 P.M. We will ring in
the New Year together! MAKE AMERICA GREAT AGAIN!
682764544402440192
HAPPY NEW YEAR &amp; THANK YOU! https://t.co/YO1Yi8QbZy https://t.co/uxUXWJ1Rbv
682792967736848385
#HappyNewYearAmerica! https://t.co/EeQb8PDrUe
682805320217980929
Happy New Year from #MarALago! Thank you to my great family for all of their support.
https://t.co/6UsqSiaaj7
682805477168779264
"@jallenaip: Hillary said she was in a "Fog of War" as explanation for the lies about
Benghazi. No fog allowed in WH. Vote Trump POTUS!"
...
...
1052213711295930368  “Federal Judge throws out Stormy Danials lawsuit versus Trump.
Trump is entitled to full legal fees.” @FoxNews Great, now I can go after Horseface
and her 3rd rate lawyer in the Great State of Texas. She will confirm the letter she
signed! She knows nothing about me, a total ...
1052217314463100928  “Conflict between Glen Simpson’s testimony to another House Panel
about his contact with Justice Department official Bruce Ohr. Ohr was used by Simpson
and Steele as a Back Channel to get (FAKE) Dossier to FBI. Simpson pleading Fifth.”
Catherine Herridge. Where is Jeff Sessions?
1052219253384994816                 Is it really possible that Bruce Ohr, whose wife
Nellie was paid by Simpson and GPS Fusion for work done on the Fake Dossier, and who
was used as a Pawn in this whole SCAM (WITCH HUNT), is still working for the
Department of Justice????? Can this really be so?????
1052232230972678145
RT @WhiteHouse: https://t.co/RNqLpOtS3O
1052233253040640001
REGISTER TO https://t.co/0pWiwCHGbh! #MAGAUS https://t.co/ACTMe53TZU
```

```
                     retweet_count  favorite_count                      time
id
682723973449289728            2108            6735 2016-01-01 00:44:14+00:00
682764544402440192            3460            8581 2016-01-01 03:25:27+00:00
682792967736848385            3434            9143 2016-01-01 05:18:23+00:00
```

```
682805320217980929              1948              8258 2016-01-01 06:07:28+00:00
682805477168779264              2721              7490 2016-01-01 06:08:06+00:00
...                              ...               ...                       ...
1052213711295930368            14594             54635 2018-10-16 15:04:32+00:00
1052217314463100928             6271             20251 2018-10-16 15:18:51+00:00
1052219253384994816            13103             41253 2018-10-16 15:26:33+00:00
1052232230972678145             4478                 0 2018-10-16 16:18:08+00:00
1052233253040640001             5415             16565 2018-10-16 16:22:11+00:00
```

```
[9478 rows x 5 columns]
```

In the following questions, we are going to find out the charateristics of Trump tweets and the devices used for the tweets.

First let's examine the source field:

```
df_trump['source'].unique()
```

```
array(['Twitter for Android', 'Twitter for iPhone', 'Twitter Web Client',
       'Mobile Web (M5)', 'Instagram', 'Twitter Ads', 'Twitter for iPad',
       'Media Studio', 'TweetDeck', 'Periscope',
       '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>',
       '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
       '<a href="https://studio.twitter.com" rel="nofollow">Media Studio</a>',
       '<a href="http://twitter.com/#!/download/ipad" rel="nofollow">Twitter for
iPad</a>'],
      dtype=object)
```

Task 2.3 - Remove HTML Tags

Remove the HTML tags from the source text field.

Hint: Use `df_trump['source'].str.replace` and your favorite regular expression.

```
import re
### BEGIN

df_trump['source'] = df_trump['source'].str.replace(r'^<a.+>(.+)</a>',r'\1',
regex=True)

df_trump['source'].unique() #double checking that the html tags are gone after using
regex pattern to replace
```

```
    ### END ANSWER
```

```
array(['Twitter for Android', 'Twitter for iPhone', 'Twitter Web Client',
       'Mobile Web (M5)', 'Instagram', 'Twitter Ads', 'Twitter for iPad',
           'Media Studio', 'TweetDeck', 'Periscope'], dtype=object)
```

Question. What is the most common device used for Trump tweets? Make a plot to find out the most common device types used

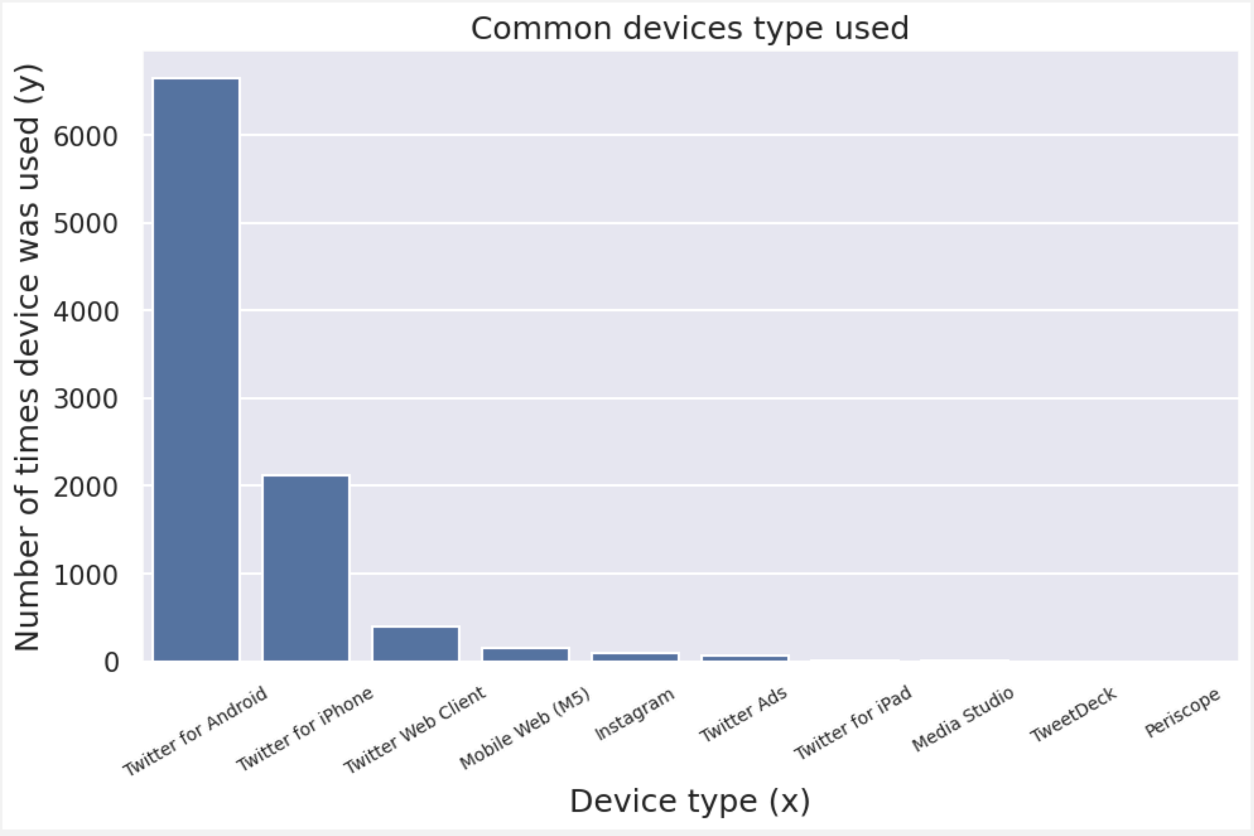
Sort the plot in decreasing order of the most common device type

```
x_val = df_trump['source'].unique()
y_val = df_trump['source'].value_counts()

    y_val
```

```
Twitter for iPhone     6649
Twitter for Android    2116
Twitter Web Client      395
Media Studio            157
Twitter Ads              96
Twitter for iPad         59
Instagram                 2
TweetDeck                 2
Mobile Web (M5)           1
Periscope                 1
    Name: source, dtype: int64
```

```
### BEGIN

import matplotlib.pyplot as plt

# Count the occurrences of each source (device type)
x_val = df_trump['source'].unique()
y_val = df_trump['source'].value_counts()

fig = plt.figure(figsize = (10, 6))

sns.barplot(y=y_val,x=x_val)

plt.xlabel("Device type (x)")
plt.ylabel("Number of times device was used (y)")
plt.title("Common devices type used")

plt.yticks(fontsize = 15)
plt.xticks(fontsize = 10, rotation=30)

plt.show()
```

```
### END ANSWER
```



```
<Figure size 1000x600 with 1 Axes>
```

Task 2.4 - Device Analysis

Is there a difference between his Tweet behavior across these devices? We will attempt to answer this question in our subsequent analysis.

First, we'll take a look at whether Trump's tweets from an Android come at different times than his tweets from an iPhone. Note that Twitter gives us his tweets in the UTC timezone (notice the `+0000` in the first few tweets)

Note - If your `time` column is not in datetime format, the following code will not work.

```
df_trump['time'][0:3]
```

```
id
682723973449289728   2016-01-01 00:44:14+00:00
682764544402440192   2016-01-01 03:25:27+00:00
682792967736848385   2016-01-01 05:18:23+00:00
    Name: time, dtype: datetime64[ns, UTC]
```

We'll convert the tweet times to US Eastern Time, the timezone of New York and Washington D.C., since those are the places we would expect the most tweet activity from Trump.

```
df_trump['est_time'] = (
    df_trump['time'] # Set initial timezone to UTC
                 .dt.tz_convert("EST") # Convert to Eastern Time
)
    df_trump.head()
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	source	text	retweet_count	favorite_count	time	est_time
id						
---	---	---	---	---	---	---
682723973449289728	Twitter for Android	I will be on @FoxNews live, with members of my family, at 11:50 P.M. We will ring in the New Year together! MAKE AMERICA GREAT AGAIN!	2108	6735	2016-01-01 00:44:14+00:00	2015-12-31 19:44:14-05:00

682764544402440192	Twitter for iPhone	HAPPY NEW YEAR & THANK YOU! https://t.co/YO1Yi8QbZy https://t.co/uxUXWJ1Rbv	3460	8581	2016-01-01 03:25:27+00:00	2015-12-31 22:25:27-05:00
682792967736848385	Twitter for iPhone	#HappyNewYearAmerica! https://t.co/EeQb8PDrUe	3434	9143	2016-01-01 05:18:23+00:00	2016-01-01 00:18:23-05:00
682805320217980929	Twitter for iPhone	Happy New Year from #MarALago! Thank you to my great family for all of their support. https://t.co/6UsqSiaaj7	1948	8258	2016-01-01 06:07:28+00:00	2016-01-01 01:07:28-05:00
682805477168779264	Twitter for Android	"@jallenaip: Hillary said she was in a "Fog of War" as explanation for the lies about Benghazi. No fog allowed in WH. Vote Trump POTUS!"	2721	7490	2016-01-01 06:08:06+00:00	2016-01-01 01:08:06-05:00

```
                                source  \
id
682723973449289728  Twitter for Android
682764544402440192   Twitter for iPhone
682792967736848385   Twitter for iPhone
682805320217980929   Twitter for iPhone
682805477168779264  Twitter for Android


text  \
id
682723973449289728    I will be on @FoxNews live,  with members of my family, at 11:50
P.M. We will ring in the New Year together! MAKE AMERICA GREAT AGAIN!
682764544402440192                                                             HAPPY NEW
YEAR &amp; THANK YOU! https://t.co/YO1Yi8QbZy https://t.co/uxUXWJ1Rbv
682792967736848385
#HappyNewYearAmerica! https://t.co/EeQb8PDrUe
682805320217980929                               Happy New Year from #MarALago! Thank
you to my great family for all of their support. https://t.co/6UsqSiaaj7
682805477168779264  "@jallenaip: Hillary said she was in a "Fog of War" as explanation
for the lies about Benghazi. No fog allowed in WH. Vote Trump POTUS!"

                    retweet_count  favorite_count                      time  \
id
682723973449289728           2108            6735 2016-01-01 00:44:14+00:00
682764544402440192           3460            8581 2016-01-01 03:25:27+00:00
682792967736848385           3434            9143 2016-01-01 05:18:23+00:00
682805320217980929           1948            8258 2016-01-01 06:07:28+00:00
682805477168779264           2721            7490 2016-01-01 06:08:06+00:00

                                    est_time
id
682723973449289728 2015-12-31 19:44:14-05:00
682764544402440192 2015-12-31 22:25:27-05:00
682792967736848385 2016-01-01 00:18:23-05:00
682805320217980929 2016-01-01 01:07:28-05:00
    682805477168779264 2016-01-01 01:08:06-05:00
```

What you need to do:

Add a column called `hour` to the `df_trump` table which contains the hour of the day as floating point number computed by:

$$
\text{hour} + \frac{\text{minute}}{60} + \frac{\text{second}}{60^2}
$$

```python
#Convert to string since its a date-time object:
df_trump['est_time'] = df_trump['est_time'].astype(str)

#df_trump['est_time'] -> for displaying to see where to split.

df_trump['hour'] = df_trump['est_time'].str.split(' ').str[1].str.split('-').str[0]
df_trump['hour'] = df_trump['hour'].str.split(':')
df_trump['hour'] = df_trump['hour'].apply(lambda x: [int(i) for i in x])
df_trump['hour'] = df_trump['hour'].apply(lambda x: x[0] + x[1]/60 + x[2]/3600)
df_trump['hour']

# a new column that contains the rounded hour
```

```
df_trump['roundhour']=round(df_trump['hour'])
    df_trump['roundhour']
```

```
id
682723973449289728     20.0
682764544402440192     22.0
682792967736848385      0.0
682805320217980929      1.0
682805477168779264      1.0
                       ...
1052213711295930368    10.0
1052217314463100928    10.0
1052219253384994816    10.0
1052232230972678145    11.0
1052233253040640001    11.0
    Name: roundhour, Length: 9478, dtype: float64
```

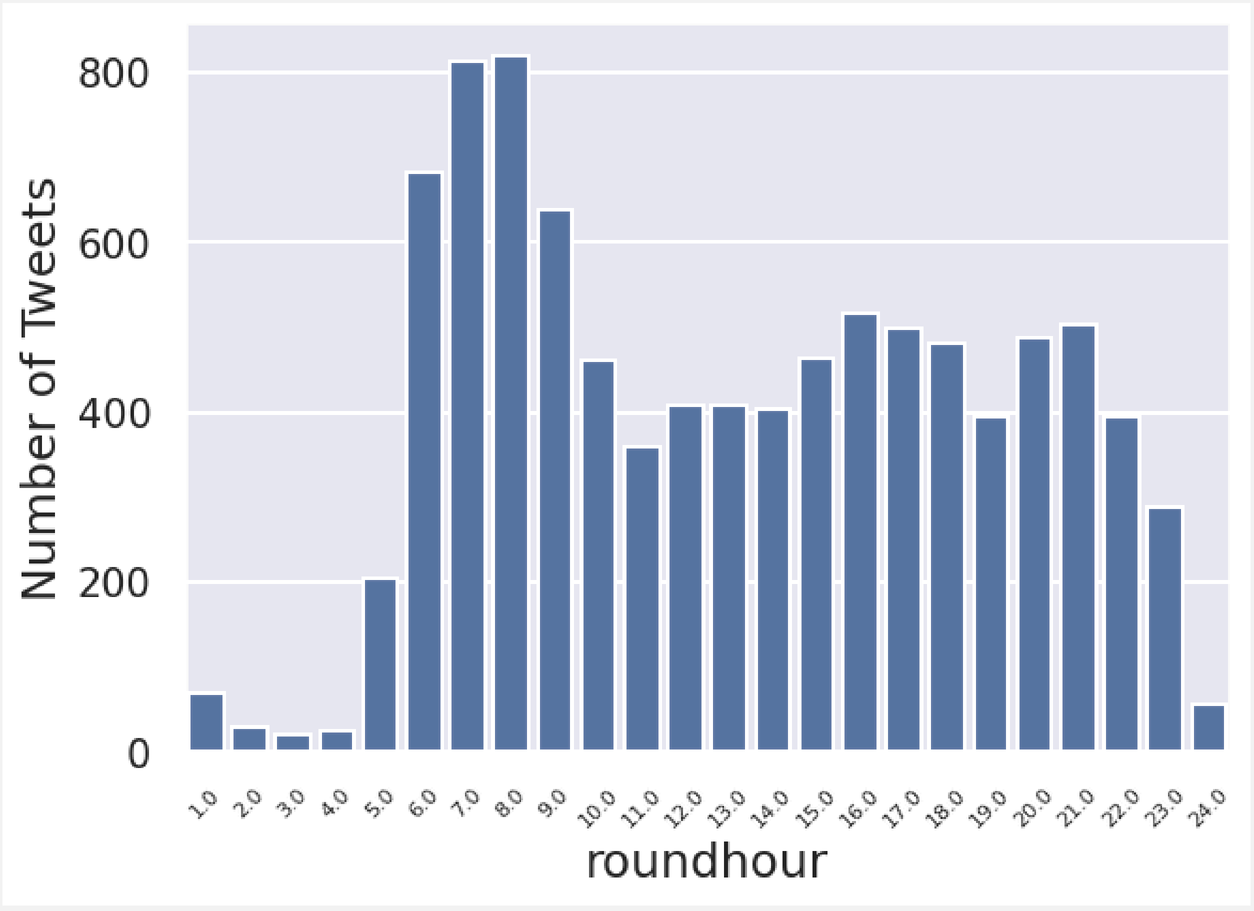
```
x = df_trump['roundhour'].value_counts()
    x
```

```
8.0     819
7.0     813
6.0     683
9.0     638
16.0    517
21.0    504
17.0    498
20.0    488
18.0    482
15.0    463
10.0    461
13.0    409
12.0    407
14.0    404
22.0    395
19.0    395
11.0    360
23.0    289
5.0     205
1.0      70
24.0     57
0.0      46
2.0      29
4.0      26
3.0      20
    Name: roundhour, dtype: int64
```

Use the `roundhour` column and plot the number of tweets at every hour of the day.

Order the plot using the hour of the day (1 to 24). Use seaborn `countplot`

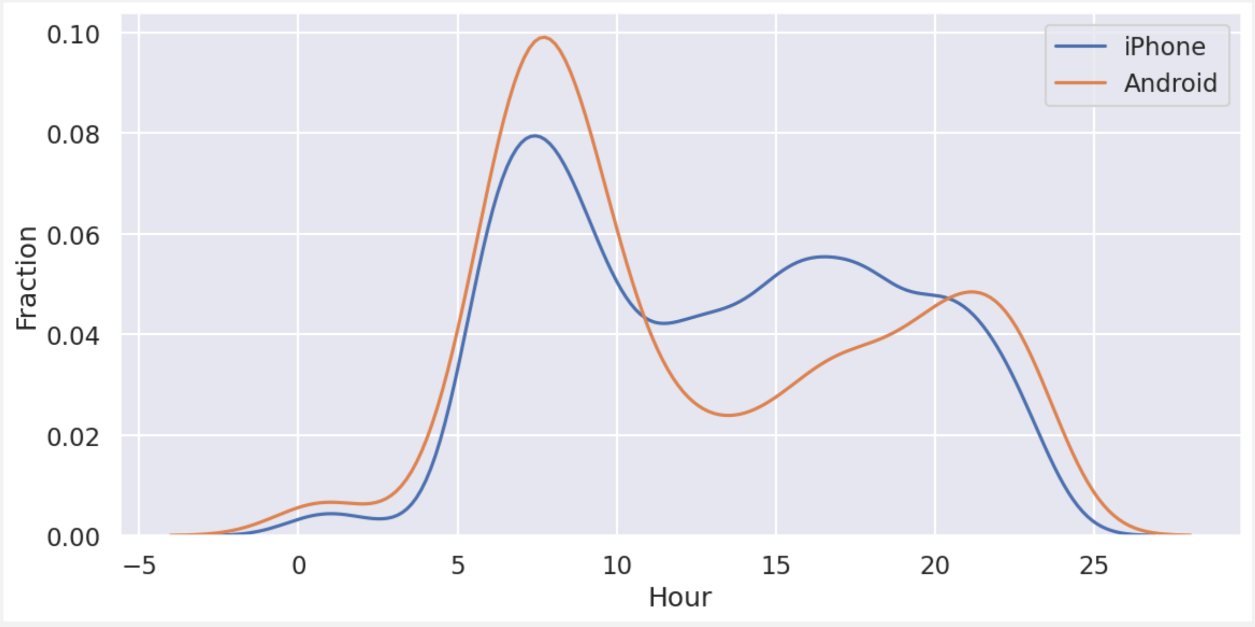
```
# make a bar plot here
### BEGIN ANSWER
import seaborn as sns
import matplotlib.pyplot as plt


sns.countplot(x = df_trump['roundhour'], data=df_trump)

plt.ylabel("Number of Tweets")
plt.yticks(fontsize = 15)
plt.xticks(fontsize = 8, rotation=45)
plt.xlim(0.5, 24.5)
# Show the plot
plt.show()

    ### END ANSWER
```



```
<Figure size 640x480 with 1 Axes>
```

Now, use this data along with the seaborn `distplot` function to examine the distribution over hours of the day in eastern time that trump tweets on each device for the 2 most commonly used devices. Your plot should look somewhat similar to the following.

```
### BEGIN ANSWER
import seaborn as sns
import matplotlib.pyplot as plt

iphone_tweets = df_trump[df_trump['source'] == 'Twitter for iPhone']['roundhour']
android_tweets = df_trump[df_trump['source'] == 'Twitter for Android']['roundhour']

plt.figure(figsize=(12,6))

sns.distplot(iphone_tweets, hist=False, kde=True, rug=False, label='iPhone')
sns.distplot(android_tweets, hist=False, kde=True, rug=False, label='Android')

plt.xlabel('Hour')
plt.ylabel('Fraction')
plt.legend()
plt.show()


    ### END ANSWER
```



```
<Figure size 1200x600 with 1 Axes>
```

Task 2.5

According to this Verge article, Donald Trump switched from an Android to an iPhone sometime in March 2017.

Create a figure identical to your figure from 3.4, except that you should show the results only from 2016. If you get stuck consider looking at the `year_fraction` function from the next problem.

Use this data along with the seaborn `distplot` function to examine the distribution over hours of the day in eastern time that trump tweets on each device for the 2 most commonly used devices. Your plot should look somewhat similar to the following.

During the campaign, it was theorized that Donald Trump's tweets from Android were written by him personally, and the tweets from iPhone were from his staff. Does your figure give support the theory?

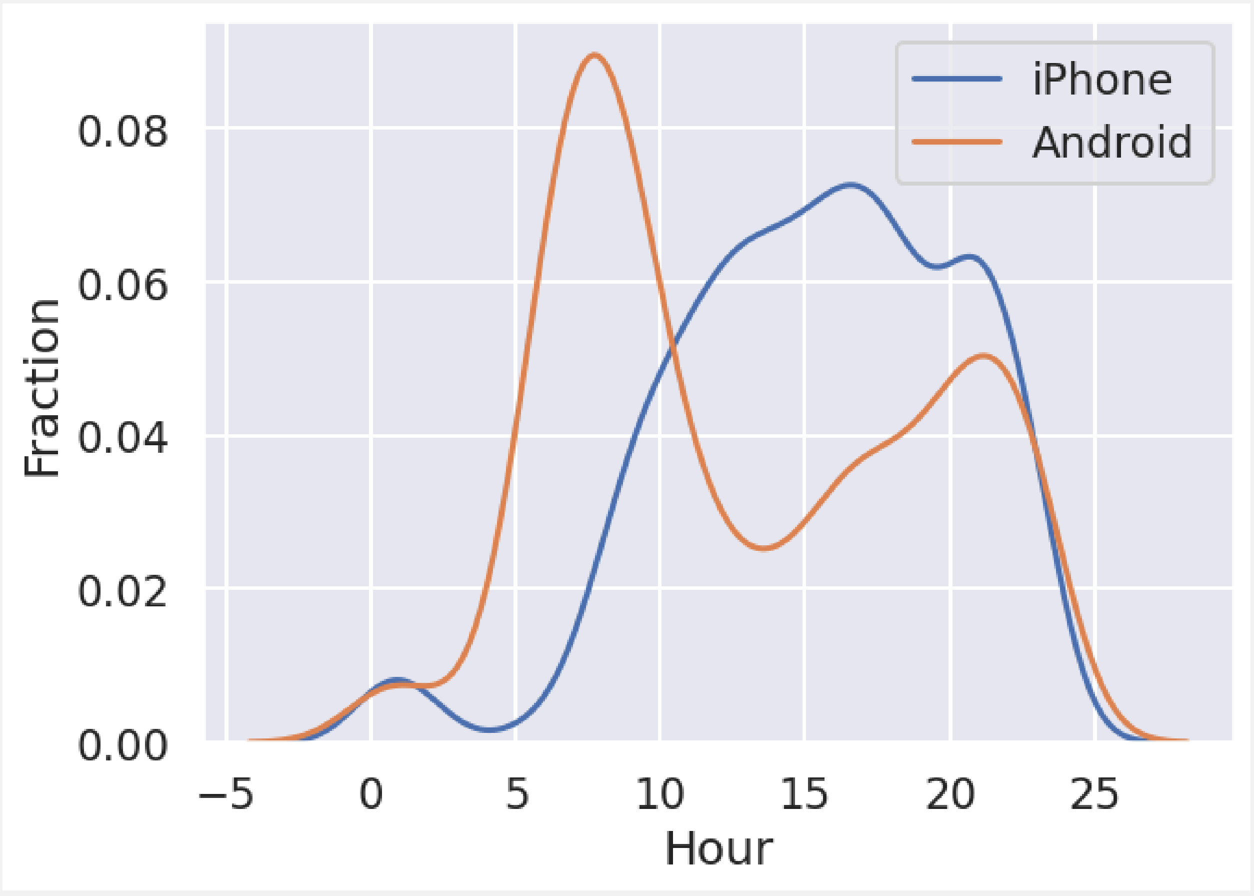
Your Response:

In 2016, the time allocation for the usage of the iphone centered in the afternoon, while his tweets from 2015 to present shows that he mostly tweets in the morning. It seems that the tweets from iphone in 2016 were from his staff, not himself.

\

```
### BEGIN ANSWER

#have a year column inside df_trump, this is so we can focus on a specific year which
is 2016:
df_trump['year'] = pd.to_datetime(df_trump['time']).dt.year


#same procedure as 2.4:
iphone_tweet_2016 = df_trump[(df_trump['source'] == 'Twitter for iPhone') &
(df_trump['year'] ==2016)]['roundhour']
```

```
android_tweets_2016 = df_trump[(df_trump['source'] == 'Twitter for Android') &
(df_trump['year'] ==2016)]['roundhour']


sns.distplot(iphone_tweet_2016, hist=False, kde=True, rug=False, label='iPhone')
sns.distplot(android_tweets_2016, hist=False, kde=True, rug=False, label='Android')


plt.xlabel('Hour')
plt.ylabel('Fraction')
plt.legend()
plt.show()

df_trump = df_trump.drop('year', axis=1) #then drop the column since we don't need it
anymore.

    ### END ANSWER
```



```
<Figure size 640x480 with 1 Axes>
```

Task 2.6

Edit this cell to answer the following questions.

- What time of the day the Android tweets were made by Trump himself? (eg: morning, late night etc):

Answer: According to the data, the android tweets were made by Trump himself in the early to late morning hours since the peak is near hours 5-10.

- What time of the day the Android tweets were made by paid staff?

Answer: The android tweets could be made by paid staff within the 0 to 1st hours and 20th-23rd hour, this is because from 0 to 1st its 12am-1am, indicating very late/odd hours. The same indication could be made for the 21st-23rd hour as those are night hours from 9pm-11pm. They could be posting during these hours due to timezone differences.

Note that these are speculations based on what you observe in the data set.

Task 2.7 Device Analysis

Let's now look at which device he has used over the entire time period of this dataset.

To examine the distribution of dates we will convert the date to a fractional year that can be plotted as a distribution.

(Code borrowed from https://stackoverflow.com/questions/6451655/python-how-to-convert-datetime-dates-to-decimal-years)

```
import datetime
def year_fraction(date):
    start = datetime.date(date.year, 1, 1).toordinal()
    year_length = datetime.date(date.year+1, 1, 1).toordinal() - start
    return date.year + float(date.toordinal() - start) / year_length
```

```
df_trump['year'] = df_trump['time'].apply(year_fraction) #should be df_trump
```

```
df_trump
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	source	text	retweet_count	favorite_count	time	est_time	hour	roundhour	year
id									
---	---	---	---	---	---	---	---	---	---
682723973449289728	Twitter for Android	I will be on @FoxNews live, with members of my family, at 11:50 P.M. We will ring in the New Year together! MAKE AMERICA GREAT AGAIN!	2108	6735	2016-01-01 00:44:14+00:00	2015-12-31 19:44:14-05:00	19.737222	20.0	2016.000000
682764544402440192	Twitter for iPhone	HAPPY NEW YEAR & THANK YOU! https://t.co/YO1Yi8QbZy https://t.co/uxUXWJ1Rbv	3460	8581	2016-01-01 03:25:27+00:00	2015-12-31 22:25:27-05:00	22.424167	22.0	2016.000000

682792967736848385	Twitter for iPhone	#HappyNewYearAmerica! https://t.co/EeQb8PDrUe	3434	9143	2016-01-01 05:18:23+00:00	2016-01-01 00:18:23-05:00	0.306389	0.0	2016.000000
682805320217980929	Twitter for iPhone	Happy New Year from #MarALago! Thank you to my great family for all of their support. https://t.co/6UsqSiaaj7	1948	8258	2016-01-01 06:07:28+00:00	2016-01-01 01:07:28-05:00	1.124444	1.0	2016.000000
682805477168779264	Twitter for Android	"@jallenaip: Hillary said she was in a "Fog of War" as explanation for the lies about Benghazi. No fog allowed in WH. Vote Trump POTUS!"	2721	7490	2016-01-01 06:08:06+00:00	2016-01-01 01:08:06-05:00	1.135000	1.0	2016.000000
...	...	...	...	...	...	...	...	...	...

1052213711295930368	Twitter for iPhone	“Federal Judge throws out Stormy Danials lawsuit versus Trump. Trump is entitled to full legal fees.” @FoxNews Great, now I can go after Horseface and her 3rd rate lawyer in the Great State of Texas. She will confirm the letter she signed! She knows nothing about me, a total ...	14594	54635	2018-10-16 15:04:32+00:00	2018-10-16 10:04:32-05:00	10.075556	10.0	2018.789041

1052217314463100928	Twitter for iPhone	“Conflict between Glen Simpson’s testimony to another House Panel about his contact with Justice Department official Bruce Ohr. Ohr was used by Simpson and Steele as a Back Channel to get (FAKE) Dossier to FBI. Simpson pleading Fifth.” Catherine Herridge. Where is Jeff Sessions?	6271	20251	2018-10-16 15:18:51+00:00	2018-10-16 10:18:51-05:00	10.314167	10.0	2018.789041

1052219253384994816	Twitter for iPhone	Is it really possible that Bruce Ohr, whose wife Nellie was paid by Simpson and GPS Fusion for work done on the Fake Dossier, and who was used as a Pawn in this whole SCAM (WITCH HUNT), is still working for the Department of Justice????? Can this really be so?????	13103	41253	2018-10-16 15:26:33+00:00	2018-10-16 10:26:33-05:00	10.442500	10.0	2018.789041
1052232230972678145	Twitter for iPhone	RT @WhiteHouse: https://t.co/RNqLpOtS3O	4478	0	2018-10-16 16:18:08+00:00	2018-10-16 11:18:08-05:00	11.302222	11.0	2018.789041

1052233253040640001	Twitter for iPhone	REGISTER TO https://t.co/0pWiwCHGbh! #MAGAus https://t.co/ACTMe53TZU	5415	16565	2018-10-16 16:22:11+00:00	2018-10-16 11:22:11-05:00	11.369722	11.0	2018.789041

9478 rows × 9 columns

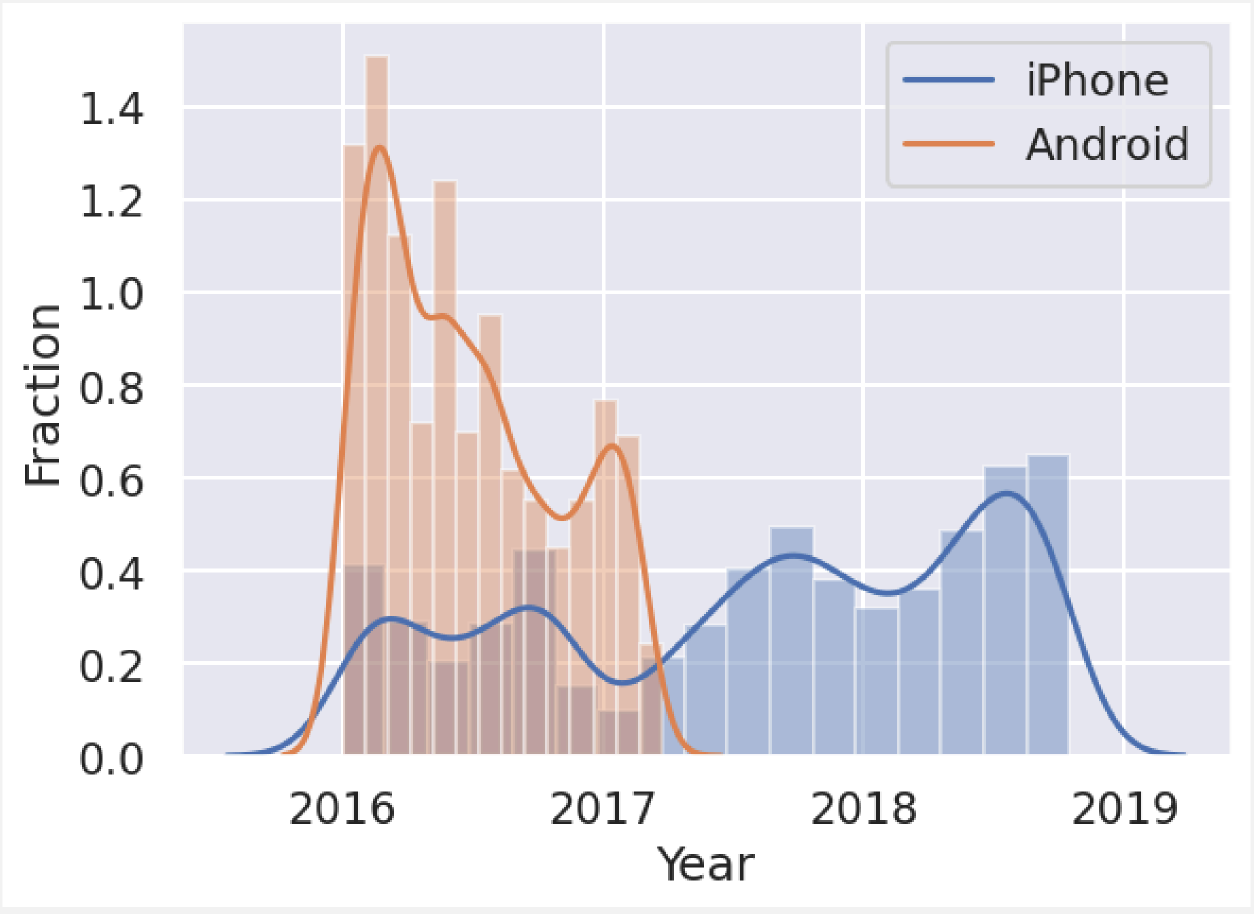
```
                                  source  \
id
682723973449289728   Twitter for Android
682764544402440192    Twitter for iPhone
682792967736848385    Twitter for iPhone
682805320217980929    Twitter for iPhone
682805477168779264   Twitter for Android
...                                  ...
1052213711295930368   Twitter for iPhone
1052217314463100928   Twitter for iPhone
1052219253384994816   Twitter for iPhone
1052232230972678145   Twitter for iPhone
1052233253040640001   Twitter for iPhone


text  \
id
682723973449289728
I will be on @FoxNews live,  with members of my family, at 11:50 P.M. We will ring in
the New Year together! MAKE AMERICA GREAT AGAIN!
682764544402440192
HAPPY NEW YEAR &amp; THANK YOU! https://t.co/YO1Yi8QbZy https://t.co/uxUXWJ1Rbv
682792967736848385
#HappyNewYearAmerica! https://t.co/EeQb8PDrUe
682805320217980929
Happy New Year from #MarALago! Thank you to my great family for all of their support.
https://t.co/6UsqSiaaj7
682805477168779264
"@jallenaip: Hillary said she was in a "Fog of War" as explanation for the lies about
Benghazi. No fog allowed in WH. Vote Trump POTUS!"
...
...
1052213711295930368  “Federal Judge throws out Stormy Danials lawsuit versus Trump.
Trump is entitled to full legal fees.” @FoxNews Great, now I can go after Horseface
and her 3rd rate lawyer in the Great State of Texas. She will confirm the letter she
signed! She knows nothing about me, a total ...
```

```
1052217314463100928  “Conflict between Glen Simpson’s testimony to another House Panel
about his contact with Justice Department official Bruce Ohr. Ohr was used by Simpson
and Steele as a Back Channel to get (FAKE) Dossier to FBI. Simpson pleading Fifth.”
Catherine Herridge. Where is Jeff Sessions?
1052219253384994816                Is it really possible that Bruce Ohr, whose wife
Nellie was paid by Simpson and GPS Fusion for work done on the Fake Dossier, and who
was used as a Pawn in this whole SCAM (WITCH HUNT), is still working for the
Department of Justice????? Can this really be so?????
1052232230972678145
RT @WhiteHouse: https://t.co/RNqLpOtS3O
1052233253040640001
REGISTER TO https://t.co/0pWiwCHGbh! #MAGAUS https://t.co/ACTMe53TZU

                     retweet_count  favorite_count                       time  \
id
682723973449289728            2108            6735 2016-01-01 00:44:14+00:00
682764544402440192            3460            8581 2016-01-01 03:25:27+00:00
682792967736848385            3434            9143 2016-01-01 05:18:23+00:00
682805320217980929            1948            8258 2016-01-01 06:07:28+00:00
682805477168779264            2721            7490 2016-01-01 06:08:06+00:00
...                            ...             ...                       ...
1052213711295930368          14594           54635 2018-10-16 15:04:32+00:00
1052217314463100928           6271           20251 2018-10-16 15:18:51+00:00
1052219253384994816          13103           41253 2018-10-16 15:26:33+00:00
1052232230972678145           4478               0 2018-10-16 16:18:08+00:00
1052233253040640001           5415           16565 2018-10-16 16:22:11+00:00

                                      est_time       hour  roundhour  \
id
682723973449289728   2015-12-31 19:44:14-05:00  19.737222       20.0
682764544402440192   2015-12-31 22:25:27-05:00  22.424167       22.0
682792967736848385   2016-01-01 00:18:23-05:00   0.306389        0.0
682805320217980929   2016-01-01 01:07:28-05:00   1.124444        1.0
682805477168779264   2016-01-01 01:08:06-05:00   1.135000        1.0
...                                        ...        ...        ...
1052213711295930368  2018-10-16 10:04:32-05:00  10.075556       10.0
1052217314463100928  2018-10-16 10:18:51-05:00  10.314167       10.0
1052219253384994816  2018-10-16 10:26:33-05:00  10.442500       10.0
1052232230972678145  2018-10-16 11:18:08-05:00  11.302222       11.0
1052233253040640001  2018-10-16 11:22:11-05:00  11.369722       11.0

                            year
id
682723973449289728   2016.000000
682764544402440192   2016.000000
682792967736848385   2016.000000
682805320217980929   2016.000000
682805477168779264   2016.000000
...                          ...
1052213711295930368  2018.789041
1052217314463100928  2018.789041
1052219253384994816  2018.789041
1052232230972678145  2018.789041
1052233253040640001  2018.789041
```

```
[9478 rows x 9 columns]
```

Use the `sns.distplot` to overlay the distributions of the 2 most frequently used web technologies over the years. Your final plot should be similar to:

```
### BEGIN ANSWER


iphone_tweet_years = df_trump[(df_trump['source'] == 'Twitter for iPhone')]['year']
android_tweets_years = df_trump[(df_trump['source'] == 'Twitter for Android')]['year']

#plt.figure(figsize=(15,15))

sns.distplot(iphone_tweet_years)

# Plot the Android data with a red color
sns.distplot(android_tweets_years)

plt.xlabel('Year')
plt.ylabel('Fraction')
plt.legend(['iPhone', 'Android'])
plt.show()

    ### END ANSWER
```



```
<Figure size 640x480 with 1 Axes>
```

PART 3 - Sentiment Analysis

It turns out that we can use the words in Trump's tweets to calculate a measure of the sentiment of the tweet. For example, the sentence "I love America!" has positive sentiment, whereas the sentence "I hate taxes!" has a negative sentiment. In addition, some words have stronger positive / negative sentiment than others: "I love America." is more positive than "I like America."

We will use the VADER (Valence Aware Dictionary and sEntiment Reasoner) lexicon to analyze the sentiment of Trump's tweets. VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media which is great for our usage.

The VADER lexicon gives the sentiment of individual words. Run the following cell to show the first few rows of the lexicon:

```
print(''.join(open("/srv/shared/vader_lexicon.txt").readlines()[:10]))
```

```
$:      -1.5    0.80623         [-1, -1, -1, -1, -3, -1, -3, -1, -2, -1]
%)      -0.4    1.0198 [-1, 0, -1, 0, 0, -2, -1, 2, -1, 0]
%-)     -1.5    1.43178         [-2, 0, -2, -2, -1, 2, -2, -3, -2, -3]
&-:     -0.4    1.42829         [-3, -1, 0, 0, -1, -1, -1, 2, -1, 2]
&:      -0.7    0.64031         [0, -1, -1, -1, 1, -1, -1, -1, -1, -1]
( '}{' )        1.6     0.66332         [1, 2, 2, 1, 1, 2, 2, 1, 3, 1]
(%      -0.9    0.9434 [0, 0, 1, -1, -1, -1, -2, -2, -1, -2]
('-:    2.2     1.16619         [4, 1, 4, 3, 1, 2, 3, 1, 2, 1]
(':     2.3     0.9     [1, 3, 3, 2, 2, 4, 2, 3, 1, 2]
    ((-:        2.1     0.53852         [2, 2, 2, 1, 2, 3, 2, 2, 3, 2]
```

Task 3.1

As you can see, the lexicon contains emojis too! The first column of the lexicon is the *token*, or the word itself. The second column is the *polarity* of the word, or how positive / negative it is.

Question How did they decide the polarities of these words? What are the other two columns in the lexicon? (See the link above.)

Read in the lexicon into a DataFrame called `df_sent`. The index of the DF should be the tokens in the lexicon. `df_sent` should have one column: `polarity`: The polarity of each token.

```
### BEGIN ANSWER
import pandas as pd
# your solution here
column_names = ['token', 'polarity', 'standard_deviation',
'raw_human_sentiment_ratings']
df_sent = pd.read_csv('/srv/shared/vader_lexicon.txt', names=column_names, sep='\t',
index_col='token')
df_sent = df_sent.drop(columns=['standard_deviation', 'raw_human_sentiment_ratings'])

df_sent.tail(10)
```

```
### END ANSWER
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	polarity		
token			
---	---		
	;-)	2.2	
	=	-0.4	
	^:	-1.1	
	o:	-0.9	
		-:	-2.3
}:	-2.1		
}:(	-2.0		
}:)	0.4		
}:-(	-2.1		
}:-)	0.3		

```
      polarity
token
|;-)        2.2
|=         -0.4
|^:        -1.1
|o:        -0.9
||-:       -2.3
}:         -2.1
}:(        -2.0
}:)         0.4
}:-(       -2.1
    }:-)        0.3
```

Task 3.2

Now, let's use this lexicon to calculate the overall sentiment for each of Trump's tweets. Here's the basic idea:

1. For each tweet, find the sentiment of each word.
2. Calculate the sentiment of each tweet by taking the sum of the sentiments of its words.

Be sure to lowercase the text in the tweets since the lexicon is also lowercase. Set the `text` column of the `df_trump` DF to be the lowercased text of each tweet.

```
### BEGIN ANSWER
```

```
df_trump['text'] = df_trump['text'].str.lower()
```

```
df_trump
```

```
    ### END ANSWER
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	source	text	retweet_count	favorite_count	time	est_time	hour	roundhour	year
id									
---	---	---	---	---	---	---	---	---	---
682723973449289728	Twitter for Android	i will be on @foxnews live, with members of my family, at 11:50 p.m. we will ring in the new year together! make america great again!	2108	6735	2016-01-01 00:44:14+00:00	2015-12-31 19:44:14-05:00	19.737222	20.0	2016.000000
682764544402440192	Twitter for iPhone	happy new year & thank you! https://t.co/yo1yi8qbzy https://t.co/uxuxwj1rbv	3460	8581	2016-01-01 03:25:27+00:00	2015-12-31 22:25:27-05:00	22.424167	22.0	2016.000000

682792967736848385	Twitter for iPhone	#happynewyearamerica! https://t.co/eeqb8pdrue	3434	9143	2016-01-01 05:18:23+00:00	2016-01-01 00:18:23-05:00	0.306389	0.0	2016.000000
682805320217980929	Twitter for iPhone	happy new year from #maralago! thank you to my great family for all of their support. https://t.co/6usqsiaj7	1948	8258	2016-01-01 06:07:28+00:00	2016-01-01 01:07:28-05:00	1.124444	1.0	2016.000000
682805477168779264	Twitter for Android	"@jallenaip: hillary said she was in a "fog of war" as explanation for the lies about benghazi. no fog allowed in wh. vote trump potus!"	2721	7490	2016-01-01 06:08:06+00:00	2016-01-01 01:08:06-05:00	1.135000	1.0	2016.000000
...	...	...	...	...	...	...	...	...	...

1052213711295930368	Twitter for iPhone	“federal judge throws out stormy danials lawsuit versus trump. trump is entitled to full legal fees.” @foxnews great, now i can go after horseface and her 3rd rate lawyer in the great state of texas. she will confirm the letter she signed! she knows nothing about me, a total ...	14594	54635	2018-10-16 15:04:32+00:00	2018-10-16 10:04:32-05:00	10.075556	10.0	2018.789041

1052217314463100928	Twitter for iPhone	“conflict between glen simpson’s testimony to another house panel about his contact with justice department official bruce ohr. ohr was used by simpson and steele as a back channel to get (fake) dossier to fbi. simpson pleading fifth.” catherine herridge. where is jeff sessions?	6271	20251	2018-10-16 15:18:51+00:00	2018-10-16 10:18:51-05:00	10.314167	10.0	2018.789041

1052219253384994816	Twitter for iPhone	is it really possible that bruce ohr, whose wife nellie was paid by simpson and gps fusion for work done on the fake dossier, and who was used as a pawn in this whole scam (witch hunt), is still working for the department of justice????? can this really be so?????	13103	41253	2018-10-16 15:26:33+00:00	2018-10-16 10:26:33-05:00	10.442500	10.0	2018.789041
1052232230972678145	Twitter for iPhone	rt @whitehouse: https://t.co/rnqlpots3o	4478	0	2018-10-16 16:18:08+00:00	2018-10-16 11:18:08-05:00	11.302222	11.0	2018.789041

1052233253040640001	Twitter for iPhone	register to https://t.co/0pwiwchgbh! #magaus https://t.co/actme53tzu	5415	16565	2018-10-16 16:22:11+00:00	2018-10-16 11:22:11-05:00	11.369722	11.0	2018.789041

9478 rows × 9 columns

```
                                  source  \
id
682723973449289728   Twitter for Android
682764544402440192    Twitter for iPhone
682792967736848385    Twitter for iPhone
682805320217980929    Twitter for iPhone
682805477168779264   Twitter for Android
...                                  ...
1052213711295930368   Twitter for iPhone
1052217314463100928   Twitter for iPhone
1052219253384994816   Twitter for iPhone
1052232230972678145   Twitter for iPhone
1052233253040640001   Twitter for iPhone


text  \
id
682723973449289728
i will be on @foxnews live,  with members of my family, at 11:50 p.m. we will ring in
the new year together! make america great again!
682764544402440192
happy new year &amp; thank you! https://t.co/yo1yi8qbzy https://t.co/uxuxwj1rbv
682792967736848385
#happynewyearamerica! https://t.co/eeqb8pdrue
682805320217980929
happy new year from #maralago! thank you to my great family for all of their support.
https://t.co/6usqsiaaj7
682805477168779264
"@jallenaip: hillary said she was in a "fog of war" as explanation for the lies about
benghazi. no fog allowed in wh. vote trump potus!"
...
...
1052213711295930368  “federal judge throws out stormy danials lawsuit versus trump.
trump is entitled to full legal fees.” @foxnews great, now i can go after horseface
and her 3rd rate lawyer in the great state of texas. she will confirm the letter she
signed! she knows nothing about me, a total ...
1052217314463100928  “conflict between glen simpson’s testimony to another house panel
about his contact with justice department official bruce ohr. ohr was used by simpson
```

```
and steele as a back channel to get (fake) dossier to fbi. simpson pleading fifth.”
catherine herridge. where is jeff sessions?
1052219253384994816                 is it really possible that bruce ohr, whose wife
nellie was paid by simpson and gps fusion for work done on the fake dossier, and who
was used as a pawn in this whole scam (witch hunt), is still working for the
department of justice????? can this really be so?????
1052232230972678145
rt @whitehouse: https://t.co/rnqlpots3o
1052233253040640001
register to https://t.co/0pwiwchgbh! #magaUS https://t.co/actme53tzu

                     retweet_count  favorite_count                      time  \
id
682723973449289728            2108            6735 2016-01-01 00:44:14+00:00
682764544402440192            3460            8581 2016-01-01 03:25:27+00:00
682792967736848385            3434            9143 2016-01-01 05:18:23+00:00
682805320217980929            1948            8258 2016-01-01 06:07:28+00:00
682805477168779264            2721            7490 2016-01-01 06:08:06+00:00
...                            ...             ...                       ...
1052213711295930368          14594           54635 2018-10-16 15:04:32+00:00
1052217314463100928           6271           20251 2018-10-16 15:18:51+00:00
1052219253384994816          13103           41253 2018-10-16 15:26:33+00:00
1052232230972678145           4478               0 2018-10-16 16:18:08+00:00
1052233253040640001           5415           16565 2018-10-16 16:22:11+00:00

                                      est_time       hour  roundhour  \
id
682723973449289728   2015-12-31 19:44:14-05:00  19.737222       20.0
682764544402440192   2015-12-31 22:25:27-05:00  22.424167       22.0
682792967736848385   2016-01-01 00:18:23-05:00   0.306389        0.0
682805320217980929   2016-01-01 01:07:28-05:00   1.124444        1.0
682805477168779264   2016-01-01 01:08:06-05:00   1.135000        1.0
...                                        ...        ...        ...
1052213711295930368  2018-10-16 10:04:32-05:00  10.075556       10.0
1052217314463100928  2018-10-16 10:18:51-05:00  10.314167       10.0
1052219253384994816  2018-10-16 10:26:33-05:00  10.442500       10.0
1052232230972678145  2018-10-16 11:18:08-05:00  11.302222       11.0
1052233253040640001  2018-10-16 11:22:11-05:00  11.369722       11.0

                            year
id
682723973449289728   2016.000000
682764544402440192   2016.000000
682792967736848385   2016.000000
682805320217980929   2016.000000
682805477168779264   2016.000000
...                          ...
1052213711295930368  2018.789041
1052217314463100928  2018.789041
1052219253384994816  2018.789041
1052232230972678145  2018.789041
1052233253040640001  2018.789041

    [9478 rows x 9 columns]
```

Task 3.3

Now, let's get rid of punctuation since it'll cause us to fail to match words. Create a new column called `no_punc` in the `df_trump` to be the lowercased text of each tweet with all punctuation replaced by a single space. We consider punctuation characters to be any character that isn't a Unicode word character or a whitespace character. You may want to consult the Python documentation on regexes for this problem.

Question Why don't we simply remove punctuation instead of replacing with a space? See if you can figure this out by looking at the tweet data.

Answer: It will ruin the sentences in the tweet by joining them together. This is why we need a space to act as a buffer between words to make sure they remain legible.

```
# Save your regex in punct_re
punct_re = r'[^\w\s\\n]'

### BEGIN ANSWER
df_trump['no_punc'] = df_trump['text'].str.replace(punct_re, ' ', regex=True)


    ### END ANSWER


assert isinstance(punct_re, str)
assert re.search(punct_re, 'this') is None
assert re.search(punct_re, 'this is ok') is None
assert re.search(punct_re, 'this is\nok') is None
assert re.search(punct_re, 'this is not ok.') is not None
assert re.search(punct_re, 'this#is#ok') is not None
assert re.search(punct_re, 'this^is ok') is not None
assert df_trump['no_punc'].loc[800329364986626048] == 'i watched parts of  nbcsnl
saturday night live last night  it is a totally one sided  biased show   nothing funny
at all  equal time for us '
    assert df_trump['text'].loc[884740553040175104] == 'working hard to get the
    olympics for the united states (l.a.). stay tuned!'
```

Task 3.4

Now, let's convert the tweets into what's called a *tidy format* to make the sentiments easier to calculate. Use the `no_punc` column of `df_trump` to create a table called `tidy_format`. The index of the table should be the IDs of the tweets, repeated once for every word in the tweet. It has two columns:

1. num: The location of the word in the tweet. For example, if the tweet was "i love america", then the location of the word "i" is 0, "love" is 1, and "america" is 2.
2. word: The individual words of each tweet.

The first few rows of our tidy_format table look like:

	num	word
894661651760377856	0	i
894661651760377856	1	think
894661651760377856	2	senator
894661651760377856	3	blumenthal
894661651760377856	4	should

You can double check that your tweet with ID 894661651760377856 has the same rows as ours. Our tests don't check whether your table looks exactly like ours.

As usual, try to avoid using any for loops. Our solution uses a chain of 5 methods on the 'trump' DF, albeit using some rather advanced Pandas hacking.

- Hint 1: Try looking at the expand argument to pandas' str.split.
- Hint 2: Try looking at the stack() method.
- Hint 3: Try looking at the level parameter of the reset_index method.

```
#make a copy of df_trump but drop the rest of the unused columns for this problem.
tidy_format = df_trump.copy()
tidy_format = tidy_format.drop(['source','retweet_count', 'favorite_count', 'time',
'est_time', 'hour', 'roundhour', 'text', 'year'], axis=1)

tidy_format = tidy_format['no_punc'].str.split(expand=True).stack()


tidy_format = tidy_format.reset_index(level=1)
tidy_format.columns = ['num', 'word']

tidy_format
    ### END ANSWER
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	num	word
id		
---	---	---
682723973449289728	0	i
682723973449289728	1	will
682723973449289728	2	be
682723973449289728	3	on
682723973449289728	4	foxnews
...	...	...
1052233253040640001	6	maga
1052233253040640001	7	https
1052233253040640001	8	t
1052233253040640001	9	co
1052233253040640001	10	actme53tzu

225001 rows × 2 columns

```
                     num        word
id
682723973449289728     0           i
682723973449289728     1        will
682723973449289728     2          be
682723973449289728     3          on
682723973449289728     4     foxnews
...                  ...         ...
1052233253040640001    6        maga
1052233253040640001    7       https
1052233253040640001    8           t
1052233253040640001    9          co
1052233253040640001   10  actme53tzu

[225001 rows x 2 columns]
```

```
assert tidy_format.loc[894661651760377856].shape == (27, 2)
    assert ' '.join(list(tidy_format.loc[894661651760377856]['word'])) == 'i think senator blumenthal should take a nice long vacation in vietnam where he lied about his service so he can at least say he was there'
```

Task 3.5

Now that we have this table in the tidy format, it becomes much easier to find the sentiment of each tweet: we can join the table with the lexicon table.

Add a `polarity` column to the `df_trump` table. The `polarity` column should contain the sum of the sentiment polarity of each word in the text of the tweet.

Hint you will need to merge the `tidy_format` and `df_sent` tables and group the final answer.

```
### BEGIN ANSWER


# Merge df1_reset and df2_reset
combined_df = tidy_format.merge(df_sent, how='left', left_on='word', right_index=True)

sentiment = combined_df.groupby(combined_df.index)['polarity'].sum()

df_trump['polarity'] = sentiment


### END ANSWER
    df_trump
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	source	**text**	**retweet_count**	**favorite_count**	**time**	**est_time**	**hour**	**roundhour**	**year**	**no_punc**	**polarity**
id											
---	---	---	---	---	---	---	---	---	---	---	---

682723973449289728	Twitter for Android	i will be on @foxnews live, with members of my family, at 11:50 p.m. we will ring in the new year together! make america great again!	2108	6735	2016-01-01 00:44:14+00:00	2015-12-31 19:44:14-05:00	19.737222	20.0	2016.000000	i will be on foxnews live with members of my family at 11 50 p m we will ring in the new year together make america great again	3.1
682764544402440192	Twitter for iPhone	happy new year & thank you! https://t.co/yo1yi8qbzy https://t.co/uxuxwj1rbv	3460	8581	2016-01-01 03:25:27+00:00	2015-12-31 22:25:27-05:00	22.424167	22.0	2016.000000	happy new year amp thank you https t co yo1yi8qbzy https t co uxuxwj1rbv	4.2

682792967736848385	Twitter for iPhone	#happynewyearamerica! https://t.co/eeqb8pdrue	3434	9143	2016-01-01 05:18:23+00:00	2016-01-01 00:18:23-05:00	0.306389	0.0	2016.000000	happynewyearamerica https t co eeqb8pdrue	0.0
682805320217980929	Twitter for iPhone	happy new year from #maralago! thank you to my great family for all of their support. https://t.co/6usqsiaaj7	1948	8258	2016-01-01 06:07:28+00:00	2016-01-01 01:07:28-05:00	1.124444	1.0	2016.000000	happy new year from maralago thank you to my great family for all of their support https t co 6usqsiaaj7	9.0

682805477168779264	Twitter for Android	"@jallenaip: hillary said she was in a "fog of war" as explanation for the lies about benghazi. no fog allowed in wh. vote trump potus!"	2721	7490	2016-01-01 06:08:06+00:00	2016-01-01 01:08:06-05:00	1.135000	1.0	2016.000000	jallenaip hillary said she was in a fog of war as explanation for the lies about benghazi no fog allowed in wh vote trump potus	-5.9
...	...	...	...	...	...	...	...	...	...	...	...

1052213711295930368	Twitter for iPhone	“federal judge throws out stormy danials lawsuit versus trump. trump is entitled to full legal fees.” @foxnews great, now i can go after horseface and her 3rd rate lawyer in the great state of texas. she will confirm the letter she signed! she knows nothing about me, a total ...	14594	54635	2018-10-16 15:04:32+00:00	2018-10-16 10:04:32-05:00	10.075556	10.0	2018.789041	federal judge throws out stormy danials lawsuit versus trump trump is entitled to full legal fees foxnews great now i can go after horseface and her 3rd rate lawyer in the great state of texas she will confirm the letter she signed she knows nothing about me a total ...	6.9

1052217314463100928	Twitter for iPhone	“conflict between glen simpson’s testimony to another house panel about his contact with justice department official bruce ohr. ohr was used by simpson and steele as a back channel to get (fake) dossier to fbi. simpson pleading fifth.” catherine herridge. where is jeff sessions?	6271	20251	2018-10-16 15:18:51+00:00	2018-10-16 10:18:51-05:00	10.314167	10.0	2018.789041	conflict between glen simpson s testimony to another house panel about his contact with justice department official bruce ohr ohr was used by simpson and steele as a back channel to get fake dossier to fbi simpson pleading fifth catherine herridge where is jeff sessions	-1.0

1052219253384994816	Twitter for iPhone	is it really possible that bruce ohr, whose wife nellie was paid by simpson and gps fusion for work done on the fake dossier, and who was used as a pawn in this whole scam (witch hunt), is still working for the department of justice????? can this really be so?????	13103	41253	2018-10-16 15:26:33+00:00	2018-10-16 10:26:33-05:00	10.442500	10.0	2018.789041	is it really possible that bruce ohr whose wife nellie was paid by simpson and gps fusion for work done on the fake dossier and who was used as a pawn in this whole scam witch hunt is still working for the department of justice can this really be so	-3.9

1052232230972678145	Twitter for iPhone	rt @whitehouse: https://t.co/rnqlpots3o	4478	0	2018-10-16 16:18:08+00:00	2018-10-16 11:18:08-05:00	11.302222	11.0	2018.789041	rt whitehouse https t co rnqlpots3o	0.0
1052233253040640001	Twitter for iPhone	register to https://t.co/0pwiwchgbh! #maga us https://t.co/actme53tzu	5415	16565	2018-10-16 16:22:11+00:00	2018-10-16 11:22:11-05:00	11.369722	11.0	2018.789041	register to https t co 0pwiwchgbh maga https t co actme53tzu	0.0

9478 rows × 11 columns

```
                                    source  \
id
682723973449289728     Twitter for Android
682764544402440192      Twitter for iPhone
682792967736848385      Twitter for iPhone
682805320217980929      Twitter for iPhone
682805477168779264     Twitter for Android
...                                    ...
1052213711295930368     Twitter for iPhone
1052217314463100928     Twitter for iPhone
1052219253384994816     Twitter for iPhone
1052232230972678145     Twitter for iPhone
1052233253040640001     Twitter for iPhone


text  \
id
682723973449289728
i will be on @foxnews live,  with members of my family, at 11:50 p.m. we will ring in
the new year together! make america great again!
682764544402440192
happy new year &amp; thank you! https://t.co/yo1yi8qbzy https://t.co/uxuxwj1rbv
682792967736848385
#happynewyearamerica! https://t.co/eeqb8pdrue
682805320217980929
happy new year from #maralago! thank you to my great family for all of their support.
https://t.co/6usqsiaaj7
```

```
682805477168779264
"@jallenaip: hillary said she was in a "fog of war" as explanation for the lies about
benghazi. no fog allowed in wh. vote trump potus!"
...
...
1052213711295930368  “federal judge throws out stormy danials lawsuit versus trump.
trump is entitled to full legal fees.” @foxnews great, now i can go after horseface
and her 3rd rate lawyer in the great state of texas. she will confirm the letter she
signed! she knows nothing about me, a total ...
1052217314463100928  “conflict between glen simpson’s testimony to another house panel
about his contact with justice department official bruce ohr. ohr was used by simpson
and steele as a back channel to get (fake) dossier to fbi. simpson pleading fifth.”
catherine herridge. where is jeff sessions?
1052219253384994816                 is it really possible that bruce ohr, whose wife
nellie was paid by simpson and gps fusion for work done on the fake dossier, and who
was used as a pawn in this whole scam (witch hunt), is still working for the
department of justice????? can this really be so?????
1052232230972678145
rt @whitehouse: https://t.co/rnqlpots3o
1052233253040640001
register to https://t.co/0pwiwchgbh! #magaUS https://t.co/actme53tzu

                     retweet_count  favorite_count                      time  \
id
682723973449289728            2108            6735 2016-01-01 00:44:14+00:00
682764544402440192            3460            8581 2016-01-01 03:25:27+00:00
682792967736848385            3434            9143 2016-01-01 05:18:23+00:00
682805320217980929            1948            8258 2016-01-01 06:07:28+00:00
682805477168779264            2721            7490 2016-01-01 06:08:06+00:00
...                            ...             ...                       ...
1052213711295930368          14594           54635 2018-10-16 15:04:32+00:00
1052217314463100928           6271           20251 2018-10-16 15:18:51+00:00
1052219253384994816          13103           41253 2018-10-16 15:26:33+00:00
1052232230972678145           4478               0 2018-10-16 16:18:08+00:00
1052233253040640001           5415           16565 2018-10-16 16:22:11+00:00

                                      est_time       hour  roundhour  \
id
682723973449289728   2015-12-31 19:44:14-05:00  19.737222       20.0
682764544402440192   2015-12-31 22:25:27-05:00  22.424167       22.0
682792967736848385   2016-01-01 00:18:23-05:00   0.306389        0.0
682805320217980929   2016-01-01 01:07:28-05:00   1.124444        1.0
682805477168779264   2016-01-01 01:08:06-05:00   1.135000        1.0
...                                        ...        ...        ...
1052213711295930368  2018-10-16 10:04:32-05:00  10.075556       10.0
1052217314463100928  2018-10-16 10:18:51-05:00  10.314167       10.0
1052219253384994816  2018-10-16 10:26:33-05:00  10.442500       10.0
1052232230972678145  2018-10-16 11:18:08-05:00  11.302222       11.0
1052233253040640001  2018-10-16 11:22:11-05:00  11.369722       11.0

                            year  \
id
682723973449289728   2016.000000
682764544402440192   2016.000000
682792967736848385   2016.000000
```

```
682805320217980929   2016.000000
682805477168779264   2016.000000
...                          ...
1052213711295930368  2018.789041
1052217314463100928  2018.789041
1052219253384994816  2018.789041
1052232230972678145  2018.789041
1052233253040640001  2018.789041
```

```
no_punc  \
id
682723973449289728
i will be on  foxnews live   with members of my family  at 11 50 p m  we will ring in
the new year together  make america great again
682764544402440192
happy new year  amp  thank you  https   t co yo1yi8qbzy https   t co uxuxwj1rbv
682792967736848385
happynewyearamerica  https   t co eeqb8pdrue
682805320217980929
happy new year from  maralago  thank you to my great family for all of their support
https   t co 6usqsiaaj7
682805477168779264
jallenaip  hillary said she was in a  fog of war  as explanation for the lies about
benghazi  no fog allowed in wh  vote trump potus
...
...
1052213711295930368   federal judge throws out stormy danials lawsuit versus trump
trump is entitled to full legal fees    foxnews great  now i can go after horseface
and her 3rd rate lawyer in the great state of texas  she will confirm the letter she
signed  she knows nothing about me  a total ...
1052217314463100928   conflict between glen simpson s testimony to another house panel
about his contact with justice department official bruce ohr  ohr was used by simpson
and steele as a back channel to get  fake  dossier to fbi  simpson pleading fifth
catherine herridge  where is jeff sessions
1052219253384994816                 is it really possible that bruce ohr  whose wife
nellie was paid by simpson and gps fusion for work done on the fake dossier  and who
was used as a pawn in this whole scam  witch hunt   is still working for the
department of justice      can this really be so
1052232230972678145
rt  whitehouse  https   t co rnqlpots3o
1052233253040640001
register to https   t co 0pwiwchgbh   maga   https   t co actme53tzu
```

```
                     polarity
id
682723973449289728        3.1
682764544402440192        4.2
682792967736848385        0.0
682805320217980929        9.0
682805477168779264       -5.9
...                       ...
1052213711295930368       6.9
1052217314463100928      -1.0
1052219253384994816      -3.9
```

```
1052232230972678145        0.0
1052233253040640001        0.0

    [9478 rows x 11 columns]
```

```
assert np.allclose(df_trump.loc[744701872456536064, 'polarity'], 8.4)
assert np.allclose(df_trump.loc[745304731346702336, 'polarity'], 2.5)
assert np.allclose(df_trump.loc[744519497764184064, 'polarity'], 1.7)
assert np.allclose(df_trump.loc[894661651760377856, 'polarity'], 0.2)
assert np.allclose(df_trump.loc[894620077634592769, 'polarity'], 5.4)
# If you fail this test, you dropped tweets with 0 polarity
    #assert np.allclose(df_trump.loc[744355251365511169, 'polarity'], 0.0)
```

Task 3.6

Now we have a measure of the sentiment of each of his tweets! You can read over the VADER readme to understand a more robust sentiment analysis.
Now, write the code to see the 20 most positive and most 20 negative tweets from Trump in your dataset:

Find the most negative and most positive tweets made by Trump

```
print('Most negative tweets:')

### BEGIN ANSWER
min_20_polarity = df_trump.nsmallest(20, 'polarity')[['text','polarity']]
print(min_20_polarity)

    ### END ANSWER
```

```
Most negative tweets:
text  \
id
1031590431379865600   it is outrageous that poisonous synthetic heroin fentanyl comes
pouring into the u.s. postal system from china. we can, and must, end this now! the
senate should pass the stop act – and firmly stop this poison from killing our
children and destroying our country. no more delay!
1029731513573822464            the rigged russian witch hunt goes on and on as the
“originators and founders” of this scam continue to be fired and demoted for their
corrupt and illegal activity. all credibility is gone from this terrible hoax, and
much more will be lost as it proceeds. no collusion!
984763579210633216   james comey is a proven leaker &amp; liar. virtually everyone in
washington thought he should be fired for the terrible job he did-until he was, in
fact, fired. he leaked classified information, for which he should be prosecuted. he
lied to congress under oath. he is a weak a...
1027585937163931648  this is an illegally brought rigged witch hunt run by people who
are totally corrupt and/or conflicted. it was started and paid for by crooked hillary
and the democrats. phony dossier, fisa disgrace and so many lying and dishonest people
already fired. 17 angry dems? stay tuned!
1031508193107763200                       where’s the collusion? they made up a phony
crime called collusion, and when there was no collusion they say there was obstruction
```

```
(of a phony crime that never existed). if you fight back or say anything bad about the
rigged witch hunt, they scream obstruction!
1022808452677160960  ....,the only collusion with russia was with the democrats, so
now they are looking at my tweets (along with 53 million other people) - the rigged
witch hunt continues! how stupid and unfair to our country....and so the fake news
doesn’t waste my time with dumb questions, no,...
1031137499995930624  ....and have demanded transparency so that this rigged and
disgusting witch hunt can come to a close. so many lives have been ruined over nothing
- mccarthyism at its worst! yet mueller &amp; his gang of dems refuse to look at the
real crimes on the other side - media is even...
934080974773776384                                                 horrible and
cowardly terrorist attack on innocent and defenseless worshipers in egypt. the world
cannot tolerate terrorism, we must defeat them militarily and discredit the extremist
ideology that forms the basis of their existence!
1023653191974625280    there is no collusion! the robert mueller rigged witch hunt,
headed now by 17 (increased from 13, including an obama white house lawyer) angry
democrats, was started by a fraudulent dossier, paid for by crooked hillary and the
dnc. therefore, the witch hunt is an illegal scam!
977585879651966980                                 our thoughts and prayers are with
the victims of the horrible attack in france yesterday, and we grieve the nation’s
loss. we also condemn the violent actions of the attacker and anyone who would provide
him support. we are with you @emmanuelmacron!
1035120511259500544  what’s going on at @cnn is happening, to different degrees, at
other networks - with @nbcnews being the worst. the good news is that andy lack(y) is
about to be fired(?) for incompetence, and much worse. when lester holt got caught
fudging my tape on russia, they were hurt ba...
803423203620245504
"@fiiibuster: @jeffzeleny pathetic - you have no sufficient evidence that donald trump
did not suffer from voter fraud, shame! bad reporter.
854283110191685634
democrat jon ossoff would be a disaster in congress. very weak on crime and illegal
immigration, bad for jobs and wants higher taxes. say no
984877999718895616
doj just issued the mccabe report - which is a total disaster. he lied! lied! lied!
mccabe was totally controlled by comey - mccabe is comey!! no collusion, all made up
by this den of thieves and lowlifes!
1041330897948160002             the illegal mueller witch hunt continues in search of
a crime. there was never collusion with russia, except by the clinton campaign, so the
17 angry democrats are looking at anything they can find. very unfair and bad for the
country. also, not allowed under the law!
1008709364939677697
it is the democrats fault for being weak and ineffective with boarder security and
crime. tell them to start thinking about the people devastated by crime coming from
illegal immigration. change the laws!
925931294705545216
nyc terrorist was happy as he asked to hang isis flag in his hospital room. he killed
8 people, badly injured 12. should get death penalty!
1038103589904777218     under our horrible immigration laws, the government is
frequently blocked from deporting criminal aliens with violent felony convictions.
house gop just passed a bill to increase our ability to deport violent felons (crazy
dems opposed). need to get this bill to my desk fast!
748906952713875456
yet another terrorist attack today in israel -- a father, shot at by a palestinian
terrorist, was killed while:\nhttps://t.co/cv1hzkvbit
```

```
980065427375128576           governor jerry “moonbeam” brown pardoned 5 criminal
illegal aliens whose crimes include (1) kidnapping and robbery (2) badly beating wife
and threatening a crime with intent to terrorize (3) dealing drugs. is this really
what the great people of california want? @foxnews

polarity
id
1031590431379865600     -20.3
1029731513573822464     -16.6
984763579210633216      -16.0
1027585937163931648     -15.2
1031508193107763200     -15.0
1022808452677160960     -14.9
1031137499995930624     -14.8
934080974773776384      -14.6
1023653191974625280     -14.4
977585879651966980      -14.3
1035120511259500544     -14.0
803423203620245504      -13.8
854283110191685634      -13.8
984877999718895616      -13.6
1041330897948160002     -13.5
1008709364939677697     -13.3
925931294705545216      -13.2
1038103589904777218     -13.1
748906952713875456      -13.0
    980065427375128576      -13.0
```

```
print('Most positive tweets:')

### BEGIN ANSWER

max_20_polarity = df_trump.nlargest(20, 'polarity')[['text','polarity']]
print(max_20_polarity)

    ### END ANSWER
```

```
Most positive tweets:
text  \
id
983143317889323008
congratulations to patrick reed on his great and courageous masters win! when patrick
had his amazing win at doral 5 years ago, people saw his great talent, and a bright
future ahead. now he is the masters champion!
1007974129474121728   my supporters are the smartest, strongest, most hard working and
most loyal that we have seen in our countries history. it is a beautiful thing to
watch as we win elections and gather support from all over the country. as we get
stronger, so does our country. best numbers ever!
1016638035281219584  thank you to all of my great supporters, really big progress
being made. other countries wanting to fix crazy trade deals. economy is roaring.
supreme court pick getting great reviews. new poll says trump, at over 90%, is the
most popular republican in history of the party. wow!
```

1014287566386888709 thank you, @wvgovernor jim justice, for that warm introduction. tonight, it was my great honor to attend the “greenbrier classic – salute to service dinner” in west virginia! god bless our veterans. god bless america - and happy independence day to all! https://t.co/v35qvcn8m6

994176238846664706 the republican party had a great night. tremendous voter energy and excitement, and all candidates are those who have a great chance of winning in november. the economy is sooo strong, and with nancy pelosi wanting to end the big tax cuts and raise taxes, why wouldn’t we win?

819541997325316096 thank you to linda bean of l.l.bean for your great support and courage. people will support you even more now. buy l.l.bean. @lbperfectmaine

1018541464416997376 congratulations to france, who played extraordinary soccer, on winning the 2018 world cup. additionally, congratulations to president putin and russia for putting on a truly great world cup tournament -- one of the best ever!

939560154269405184 it was my great honor to celebrate the opening of two extraordinary museums-the mississippi state history museum &amp; the mississippi civil rights museum. we pay solemn tribute to our heroes of the past &amp; dedicate ourselves to building a future of freedom, equality, just...

1029717928130818048 congratulations to bryan steil on a wonderful win last night. you will be replacing a great guy in paul ryan, and your win in november will make the entire state of wisconsin very proud. you have my complete and total endorsement!

1016663151935291393 on behalf of the united states, congratulations to the thai navy seals and all on the successful rescue of the 12 boys and their coach from the treacherous cave in thailand. such a beautiful moment - all freed, great job!

815449933453127681 rt @ivankatrump: 2016 has been one of the most eventful and exciting years of my life. i wish you peace, joy, love and laughter. happy new…

996723907867676673 congratulations to lou barletta of pennsylvania. he will be a great senator and will represent his people well - like they haven’t been represented in many years. lou is a friend of mine and a special guy, he will very much help make america great again!

950866561153331202 today, it was my great honor to sign a new executive order to ensure veterans have the resources they need as they transition back to civilian life. we must ensure that our heroes are given the care and support they so richly deserve! https://t.co/0mdp9ddias https://t.co/lp2a...

1029712857724792832 congratulations to leah vukmir of wisconsin on your great win last night. you beat a very tough and good competitor and will make a fantastic senator after winning in november against someone who has done very little. you have my complete and total endorsement!

956298399946563585 it was my great honor to welcome mayor’s from across america to the wh. my administration will always support local government - and listen to the leaders who know their communities best. together, we will usher in a bold new era of peace and prosperity! https://t.co/dmyectnk...

1022248383531163648 this week, my administration is hosting the first-ever #irfministerial. the u.s. will continue to promote #religiousfreedom around the world. nations that support religious freedom are far more free, prosperous &amp; peaceful. great job, @vp, @secpompeo, @irf_ambassador &amp;...

935493619204620288 melania, our great and very hard working first lady, who truly loves what she is doing, always thought that “if you run, you will win.” she would tell everyone that, “no doubt, he will win.” i also felt i would win (or i would not have run) - and country is doing great!

```
1026495277824401408  kris kobach, a strong and early supporter of mine, is running for
governor of the great state of kansas.  he is a fantastic guy who loves his state and
our country - he will be a great governor and has my full &amp; total endorsement!
strong on crime, border &amp; military. v...
962784821389996033   rep. lou barletta, a great republican from pennsylvania who was
one of my very earliest supporters, will make a fantastic senator. he is strong &amp;
smart, loves pennsylvania &amp; loves our country! voted for tax cuts, unlike bob
casey, who listened to tax hikers pelosi and...
973716838889660416                     it was my great honor to deliver a message at
the marine corps air station miramar to our great u.s. military, straight from the
heart of the american people: we support you, we thank you, we love you - and we will
always have your back! https://t.co/oct1nh3don

polarity
id
983143317889323008       26.5
1007974129474121728      20.7
1016638035281219584      18.9
1014287566386888709      18.6
994176238846664706       18.5
819541997325316096       18.2
1018541464416997376      17.8
939560154269405184       17.7
1029717928130818048      17.7
1016663151935291393      17.5
815449933453127681       17.3
996723907867676673       17.3
950866561153331202       16.6
1029712857724792832      16.5
956298399946563585       16.3
1022248383531163648      16.2
935493619204620288       16.1
1026495277824401408      16.0
962784821389996033       15.8
    973716838889660416       15.7
```

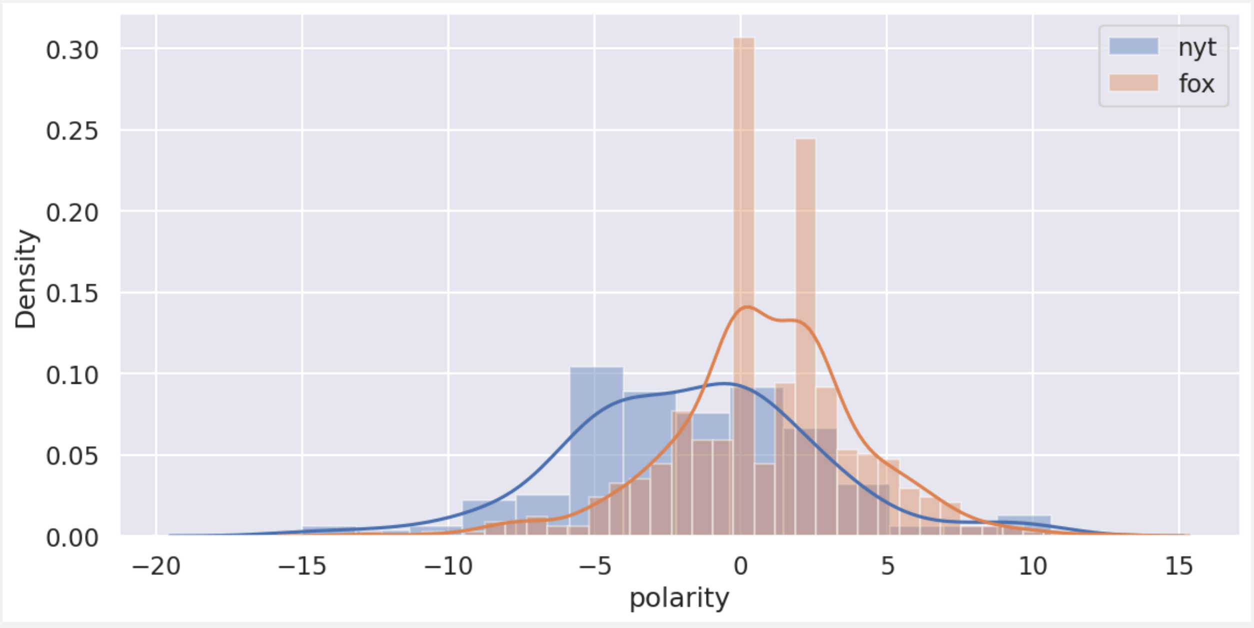
Task 3.7

Plot the distribution of tweet sentiments broken down by whether the text of the tweet contains `nyt` or `fox`. Then in the box below comment on what we observe?

```
### BEGIN ANSWER
```

```
# your solution here

plt.figure(figsize=(12,6))
nyt = df_trump[df_trump['text'].str.contains('nyt')]['polarity']
fox = df_trump[df_trump['text'].str.contains('fox')]['polarity']

sns.distplot(nyt, label='nyt')
sns.distplot(fox, label='fox')


plt.legend()
plt.show()
```

```
### END ANSWER
```



```
<Figure size 1200x600 with 1 Axes>
```

Comment on what you observe:

Based on the distplot graph, it shows that Trump's tweets that involve the New York Times often lean towards very negative sentiments (implying as a way to target and offend the New York Times) with very little to no positive sentiment at all. Meanwhile Trump's tweets involving Fox News has more variation with sentiment, where majority of them do have positive sentiment but can sometimes have negative sentiment depending on the context of the tweet but very minimal.

PART 4 - Principal Component Analysis (PCA) and Twitter

A look at the top words used and the sentiments expressed in Trump tweets indicates that, some words are used with others almost all the time. A notable example is the slogan like Make America Great Again. As such, it may be beneficial to look at groups of words rather than individual words. For that, we will look at an approach applying a Principal Component Analysis.

The PCA

The Principal Component Analysis, or PCA, is a tool generally used to identify patterns and to reduce the number of variables you have to consider in your analysis. For example, if you have data with 200 columns, it may be that a significant amount of the variance in your data can be explained by just 100 principal components. In the PCA, the first component is chosen in such a way that has the largest variance, subsequent components are orthogonal and continue covering as much variance as possible. In this way, the PCA samples as much of the variability in the data set with the first few components. Mathematically, each component is a linear combination of all the input parameters times coefficients specific for that component. These coefficients, or loading factors, are constrained such that the sum of the squares of them are equal to 1. As such, the loading factors serve as weights describing how strongly certain parameters contribute to the specific principal component. Parameters with large values of positive or negative loading factors are correlated with each other, which can serve to identify trends in your data.

Task 4.1 Cleaning up the Data

Using NLTK (Natural Language Toolkit) package for language processing and other python libraries, parse the json file to deal with inflected words, such as plurals, and removed stop words like common English words (the, and, it, etc) and certain political terms (the candidates names, for example). You can start with the top 50 words, but full analysis may require large number of words.
Create a document-frequecy (df) matrix with 5000 rows and 50 columns where each column is a particular word (feature) and each row is a tweet (observation). The values of the matrix is how often the word appears. Apply the techniques we learned to reduce the weight of most common words (if necessary). Since this is a sparse matrix, you can use the sparse martix libraries to make things a bit more efficient (we can also use a regular numpy arrays to store these things since the dimensions are not too large). See demo notes books and lecture slides for some sparse matrix methods.

Print the first 10 rows of the df to show the matrix you created

Start with the `tidy_format` dataframe

```
### BEGIN ANSWER
## code to plot the first 10 rows of the matrix
import nltk
import nltk.corpus
```

```
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

# create a dataframe called tmp to store all words appear in the tweets

tmp = pd.DataFrame(tidy_format['word'], columns=['word'])


# remove stopwords and drop the empty ones that originally contained stopwords.

stopwords = nltk.corpus.stopwords.words('english')
stopwords.extend(['rt', 'https', 'co', 'trump2016', 'realdonaldtrump', 'u', 'amp',
'trump', 'hillary', 'clinton'])
tmp['word'] = tmp['word'].apply(lambda x: ' '.join([word for word in
nltk.word_tokenize(x) if word not in stopwords]))

tmp = tmp[tmp['word'] != '']

# deal with plurals

lemmatizer = nltk.WordNetLemmatizer()

tmp['word'] = tmp['word'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for
word in nltk.word_tokenize(x)]))


# Remove numbers

tmp['word'] = tmp['word'].str.replace(r'\d+', '', regex=True)
tmp = tmp[tmp['word'] != '']

# Remove words with only 1 or 2 length

tmp = tmp[(tmp['word'].str.len() != 2) & (tmp['word'].str.len() != 1)]

#construct the document matrix:

words = tmp.value_counts(ascending=False)
top50_words = words.reset_index()['word'][:50].to_list()
top50_words

frequency = {}

doc = pd.DataFrame(columns=top50_words)
X = tmp[tmp['word'].isin(top50_words)].index.unique()[:5000] #locate 5000 unique id's
that have words that belong to the top50 words


for i in X:

    for word in top50_words: #make a dict filled with 0's as a start based on top 50
words.
        frequency[word] = 0
    lst = list(tmp.loc[i, 'word'])

    for word in lst:
        if word in top50_words:
            frequency[word] += 1

    doc.loc[i] = list(frequency.values())
```

```
doc

#referenced recitation

    ### END ANSWER
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	great	thank	people	america	president	country	job	big	time	make	...	election	obama	deal	even	join	love	first	trade	campaign	poll
682723973449289728	1	0	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
682764544402440192	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

682805320217980929	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
682805477168779264	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
683037464504745985	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
921871707564101632	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
921888248707846144	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0

921889525881819136	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
922067676708638721	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
922070659093671936	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5000 rows × 50 columns

```
                    great  thank  people  america  president  country  job  \
682723973449289728      1      0       0        1          0        0    0
682764544402440192      0      1       0        0          0        0    0
682805320217980929      1      1       0        0          0        0    0
682805477168779264      0      0       0        0          0        0    0
683037464504745985      0      0       0        0          0        0    0
...                   ...    ...     ...      ...        ...      ...  ...
921871707564101632      0      0       0        0          0        0    1
921888248707846144      0      0       0        0          0        0    0
921889525881819136      0      0       0        0          0        0    0
922067676708638721      0      0       0        0          0        0    0
922070659093671936      0      0       0        0          0        0    0

                    big  time  make  ...  election  obama  deal  even  join  \
682723973449289728    0     0     1  ...         0      0     0     0     0
682764544402440192    0     0     0  ...         0      0     0     0     0
682805320217980929    0     0     0  ...         0      0     0     0     0
682805477168779264    0     0     0  ...         0      0     0     0     0
683037464504745985    0     0     0  ...         0      0     0     0     0
```

```
...                   ...   ...   ...  ...        ...    ...   ...   ...   ...
921871707564101632      0     0     0  ...          0      0     0     0     0
921888248707846144      0     0     0  ...          0      0     0     0     0
921889525881819136      0     0     0  ...          0      0     0     0     0
922067676708638721      0     0     0  ...          0      0     0     0     0
922070659093671936      0     0     0  ...          0      0     0     0     0

                    love  first  trade  campaign  poll
682723973449289728     0      0      0         0     0
682764544402440192     0      0      0         0     0
682805320217980929     0      0      0         0     0
682805477168779264     0      0      0         0     0
683037464504745985     0      0      0         0     0
...                  ...    ...    ...       ...   ...
921871707564101632     0      0      0         0     0
921888248707846144     0      1      0         0     0
921889525881819136     0      0      0         0     0
922067676708638721     0      0      0         0     0
922070659093671936     0      0      0         0     0
```

```
[5000 rows x 50 columns]
```

Task 4.2 Find the PCA's

Write the code to find the first 50 PCA's for the document-frequency matrix. Pass the document-term-matrix to scikit-learn's (https://scikit-learn.org/stable/modules/decomposition.html#decompositions) PCA method to obtain the components and loading factors.

```
!pip install -U scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting scikit-learn
Downloading
scikit_learn-1.4.1.post1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(12.1 MB)
[2K     [90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━[0m
[32m12.1/12.1 MB[0m [31m54.2 MB/s[0m eta [36m0:00:00[0m00:01[0m00:01[0m
[?25hRequirement already satisfied: numpy<2.0,>=1.19.5 in
/koko/system/anaconda3/envs/python310/lib/python3.10/site-packages (from scikit-learn)
(1.23.5)
Requirement already satisfied: scipy>=1.6.0 in
/koko/system/anaconda3/envs/python310/lib/python3.10/site-packages (from scikit-learn)
(1.10.1)
Requirement already satisfied: joblib>=1.2.0 in
/koko/system/anaconda3/envs/python310/lib/python3.10/site-packages (from scikit-learn)
(1.2.0)
Collecting threadpoolctl>=2.0.0 (from scikit-learn)
Downloading threadpoolctl-3.4.0-py3-none-any.whl (17 kB)
Installing collected packages: threadpoolctl, scikit-learn
    Successfully installed scikit-learn-1.4.1.post1 threadpoolctl-3.4.0
```

```
### BEGIN ANSWER
from sklearn.decomposition import PCA

pca = PCA(n_components=50)
df_pca = pca.fit(doc)

df_pca

    ### END ANSWER
```

```
#sk-container-id-1 { /* Definition of color scheme common for light and dark mode */ --sklearn-color-text:
black; --sklearn-color-line: gray; /* Definition of color scheme for unfitted estimators */
--sklearn-color-unfitted-level-0: #fff5e6; --sklearn-color-unfitted-level-1: #f6e4d2;
--sklearn-color-unfitted-level-2: #ffe0b3; --sklearn-color-unfitted-level-3: chocolate; /* Definition of color
scheme for fitted estimators */ --sklearn-color-fitted-level-0: #f0f8ff; --sklearn-color-fitted-level-1: #d4ebff;
--sklearn-color-fitted-level-2: #b3dbfd; --sklearn-color-fitted-level-3: cornflowerblue; /* Specific color for light
theme */ --sklearn-color-text-on-default-background: var(--sg-text-color, var(--theme-code-foreground,
var(--jp-content-font-color1, black))); --sklearn-color-background: var(--sg-background-color,
var(--theme-background, var(--jp-layout-color0, white))); --sklearn-color-border-box: var(--sg-text-color,
var(--theme-code-foreground, var(--jp-content-font-color1, black))); --sklearn-color-icon: #696969; @media
(prefers-color-scheme: dark) { /* Redefinition of color scheme for dark theme */
--sklearn-color-text-on-default-background: var(--sg-text-color, var(--theme-code-foreground,
var(--jp-content-font-color1, white))); --sklearn-color-background: var(--sg-background-color,
var(--theme-background, var(--jp-layout-color0, #111))); --sklearn-color-border-box: var(--sg-text-color,
var(--theme-code-foreground, var(--jp-content-font-color1, white))); --sklearn-color-icon: #878787; } }
#sk-container-id-1 { color: var(--sklearn-color-text); } #sk-container-id-1 pre { padding: 0; } #sk-container-id-1
input.sk-hidden--visually { border: 0; clip: rect(1px 1px 1px 1px); clip: rect(1px, 1px, 1px, 1px); height: 1px;
margin: -1px; overflow: hidden; padding: 0; position: absolute; width: 1px; } #sk-container-id-1
div.sk-dashed-wrapped { border: 1px dashed var(--sklearn-color-line); margin: 0 0.4em 0.5em 0.4em;
box-sizing: border-box; padding-bottom: 0.4em; background-color: var(--sklearn-color-background); }
#sk-container-id-1 div.sk-container { /* jupyter's `normalize.less` sets `[hidden] { display: none; }` but
bootstrap.min.css set `[hidden] { display: none !important; }` so we also need the `!important` here to be able
to override the default hidden behavior on the sphinx rendered scikit-learn.org. See:
https://github.com/scikit-learn/scikit-learn/issues/21755 */ display: inline-block !important; position: relative; }
#sk-container-id-1 div.sk-text-repr-fallback { display: none; } div.sk-parallel-item, div.sk-serial, div.sk-item { /*
draw centered vertical line to link estimators */ background-image:
linear-gradient(var(--sklearn-color-text-on-default-background),
var(--sklearn-color-text-on-default-background)); background-size: 2px 100%; background-repeat: no-repeat;
background-position: center center; } /* Parallel-specific style estimator block */ #sk-container-id-1
div.sk-parallel-item::after { content: ""; width: 100%; border-bottom: 2px solid
var(--sklearn-color-text-on-default-background); flex-grow: 1; } #sk-container-id-1 div.sk-parallel { display: flex;
align-items: stretch; justify-content: center; background-color: var(--sklearn-color-background); position:
relative; } #sk-container-id-1 div.sk-parallel-item { display: flex; flex-direction: column; } #sk-container-id-1
div.sk-parallel-item:first-child::after { align-self: flex-end; width: 50%; } #sk-container-id-1
div.sk-parallel-item:last-child::after { align-self: flex-start; width: 50%; } #sk-container-id-1
div.sk-parallel-item:only-child::after { width: 0; } /* Serial-specific style estimator block */ #sk-container-id-1
div.sk-serial { display: flex; flex-direction: column; align-items: center; background-color:
var(--sklearn-color-background); padding-right: 1em; padding-left: 1em; } /* Toggleable style: style used for
estimator/Pipeline/ColumnTransformer box that is clickable and can be expanded/collapsed. - Pipeline and
ColumnTransformer use this feature and define the default style - Estimators will overwrite some part of the
style using the `sk-estimator` class */ /* Pipeline and ColumnTransformer style (default) */ #sk-container-id-1
div.sk-toggleable { /* Default theme specific background. It is overwritten whether we have a specific estimator
or a Pipeline/ColumnTransformer */ background-color: var(--sklearn-color-background); } /* Toggleable label */
#sk-container-id-1 label.sk-toggleable__label { cursor: pointer; display: block; width: 100%; margin-bottom: 0;
padding: 0.5em; box-sizing: border-box; text-align: center; } #sk-container-id-1
label.sk-toggleable__label-arrow:before { /* Arrow on the left of the label */ content: "▸"; float: left;
margin-right: 0.25em; color: var(--sklearn-color-icon); } #sk-container-id-1
label.sk-toggleable__label-arrow:hover:before { color: var(--sklearn-color-text); } /* Toggleable content -
dropdown */ #sk-container-id-1 div.sk-toggleable__content { max-height: 0; max-width: 0; overflow: hidden;
text-align: left; /* unfitted */ background-color: var(--sklearn-color-unfitted-level-0); } #sk-container-id-1
div.sk-toggleable__content.fitted { /* fitted */ background-color: var(--sklearn-color-fitted-level-0); }
#sk-container-id-1 div.sk-toggleable__content pre { margin: 0.2em; border-radius: 0.25em; color:
var(--sklearn-color-text); /* unfitted */ background-color: var(--sklearn-color-unfitted-level-0); }
#sk-container-id-1 div.sk-toggleable__content.fitted pre { /* unfitted */ background-color:
var(--sklearn-color-fitted-level-0); } #sk-container-id-1
input.sk-toggleable__control:checked~div.sk-toggleable__content { /* Expand drop-down */ max-height:
200px; max-width: 100%; overflow: auto; } #sk-container-id-1
input.sk-toggleable__control:checked~label.sk-toggleable__label-arrow:before { content: "▾"; } /*
Pipeline/ColumnTransformer-specific style */ #sk-container-id-1 div.sk-label
input.sk-toggleable__control:checked~label.sk-toggleable__label { color: var(--sklearn-color-text);
background-color: var(--sklearn-color-unfitted-level-2); } #sk-container-id-1 div.sk-label.fitted
input.sk-toggleable__control:checked~label.sk-toggleable__label { background-color:
var(--sklearn-color-fitted-level-2); } /* Estimator-specific style */ /* Colorize estimator box */ #sk-container-id-1
div.sk-estimator input.sk-toggleable__control:checked~label.sk-toggleable__label { /* unfitted */
background-color: var(--sklearn-color-unfitted-level-2); } #sk-container-id-1 div.sk-estimator.fitted
input.sk-toggleable__control:checked~label.sk-toggleable__label { /* fitted */ background-color:
var(--sklearn-color-fitted-level-2); } #sk-container-id-1 div.sk-label label.sk-toggleable__label,
#sk-container-id-1 div.sk-label label { /* The background is the default theme color */ color:
var(--sklearn-color-text-on-default-background); } /* On hover, darken the color of the background */
#sk-container-id-1 div.sk-label:hover label.sk-toggleable__label { color: var(--sklearn-color-text);
background-color: var(--sklearn-color-unfitted-level-2); } /* Label box, darken color on hover, fitted */
#sk-container-id-1 div.sk-label.fitted:hover label.sk-toggleable__label.fitted { color: var(--sklearn-color-text);
background-color: var(--sklearn-color-fitted-level-2); } /* Estimator label */ #sk-container-id-1 div.sk-label label
{ font-family: monospace; font-weight: bold; display: inline-block; line-height: 1.2em; } #sk-container-id-1
div.sk-label-container { text-align: center; } /* Estimator-specific */ #sk-container-id-1 div.sk-estimator {
font-family: monospace; border: 1px dotted var(--sklearn-color-border-box); border-radius: 0.25em; box-sizing:
border-box; margin-bottom: 0.5em; /* unfitted */ background-color: var(--sklearn-color-unfitted-level-0); }
#sk-container-id-1 div.sk-estimator.fitted { /* fitted */ background-color: var(--sklearn-color-fitted-level-0); } /* on
hover */ #sk-container-id-1 div.sk-estimator:hover { /* unfitted */ background-color:
var(--sklearn-color-unfitted-level-2); } #sk-container-id-1 div.sk-estimator.fitted:hover { /* fitted */
background-color: var(--sklearn-color-fitted-level-2); } /* Specification for estimator info (e.g. "i" and "?") */ /*
Common style for "i" and "?" */ .sk-estimator-doc-link, a:link.sk-estimator-doc-link,
a:visited.sk-estimator-doc-link { float: right; font-size: smaller; line-height: 1em; font-family: monospace;
background-color: var(--sklearn-color-background); border-radius: 1em; height: 1em; width: 1em;
text-decoration: none !important; margin-left: 1ex; /* unfitted */ border: var(--sklearn-color-unfitted-level-1) 1pt
solid; color: var(--sklearn-color-unfitted-level-1); } .sk-estimator-doc-link.fitted,
a:link.sk-estimator-doc-link.fitted, a:visited.sk-estimator-doc-link.fitted { /* fitted */ border:
var(--sklearn-color-fitted-level-1) 1pt solid; color: var(--sklearn-color-fitted-level-1); } /* On hover */
div.sk-estimator:hover .sk-estimator-doc-link:hover, .sk-estimator-doc-link:hover, div.sk-label-container:hover
.sk-estimator-doc-link:hover, .sk-estimator-doc-link:hover { /* unfitted */ background-color:
var(--sklearn-color-unfitted-level-3); color: var(--sklearn-color-background); text-decoration: none; }
div.sk-estimator.fitted:hover .sk-estimator-doc-link.fitted:hover, .sk-estimator-doc-link.fitted:hover,
div.sk-label-container:hover .sk-estimator-doc-link.fitted:hover, .sk-estimator-doc-link.fitted:hover { /* fitted */
background-color: var(--sklearn-color-fitted-level-3); color: var(--sklearn-color-background); text-decoration:
none; } /* Span, style for the box shown on hovering the info icon */ .sk-estimator-doc-link span { display:
none; z-index: 9999; position: relative; font-weight: normal; right: .2ex; padding: .5ex; margin: .5ex; width:
min-content; min-width: 20ex; max-width: 50ex; color: var(--sklearn-color-text); box-shadow: 2pt 2pt 4pt #999;
/* unfitted */ background: var(--sklearn-color-unfitted-level-0); border: .5pt solid
var(--sklearn-color-unfitted-level-3); } .sk-estimator-doc-link.fitted span { /* fitted */ background:
var(--sklearn-color-fitted-level-0); border: var(--sklearn-color-fitted-level-3); } .sk-estimator-doc-link:hover span
{ display: block; } /* "?"-specific style due to the `` HTML tag */ #sk-container-id-1 a.estimator_doc_link { float:
right; font-size: 1rem; line-height: 1em; font-family: monospace; background-color:
var(--sklearn-color-background); border-radius: 1rem; height: 1rem; width: 1rem; text-decoration: none; /*
unfitted */ color: var(--sklearn-color-unfitted-level-1); border: var(--sklearn-color-unfitted-level-1) 1pt solid; }
#sk-container-id-1 a.estimator_doc_link.fitted { /* fitted */ border: var(--sklearn-color-fitted-level-1) 1pt solid;
color: var(--sklearn-color-fitted-level-1); } /* On hover */ #sk-container-id-1 a.estimator_doc_link:hover { /*
unfitted */ background-color: var(--sklearn-color-unfitted-level-3); color: var(--sklearn-color-background);
text-decoration: none; } #sk-container-id-1 a.estimator_doc_link.fitted:hover { /* fitted */ background-color:
var(--sklearn-color-fitted-level-3); }
```

PCA(n_components=50)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

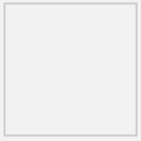
PCA?Documentation for PCAiFitted

PCA(n_components=50)

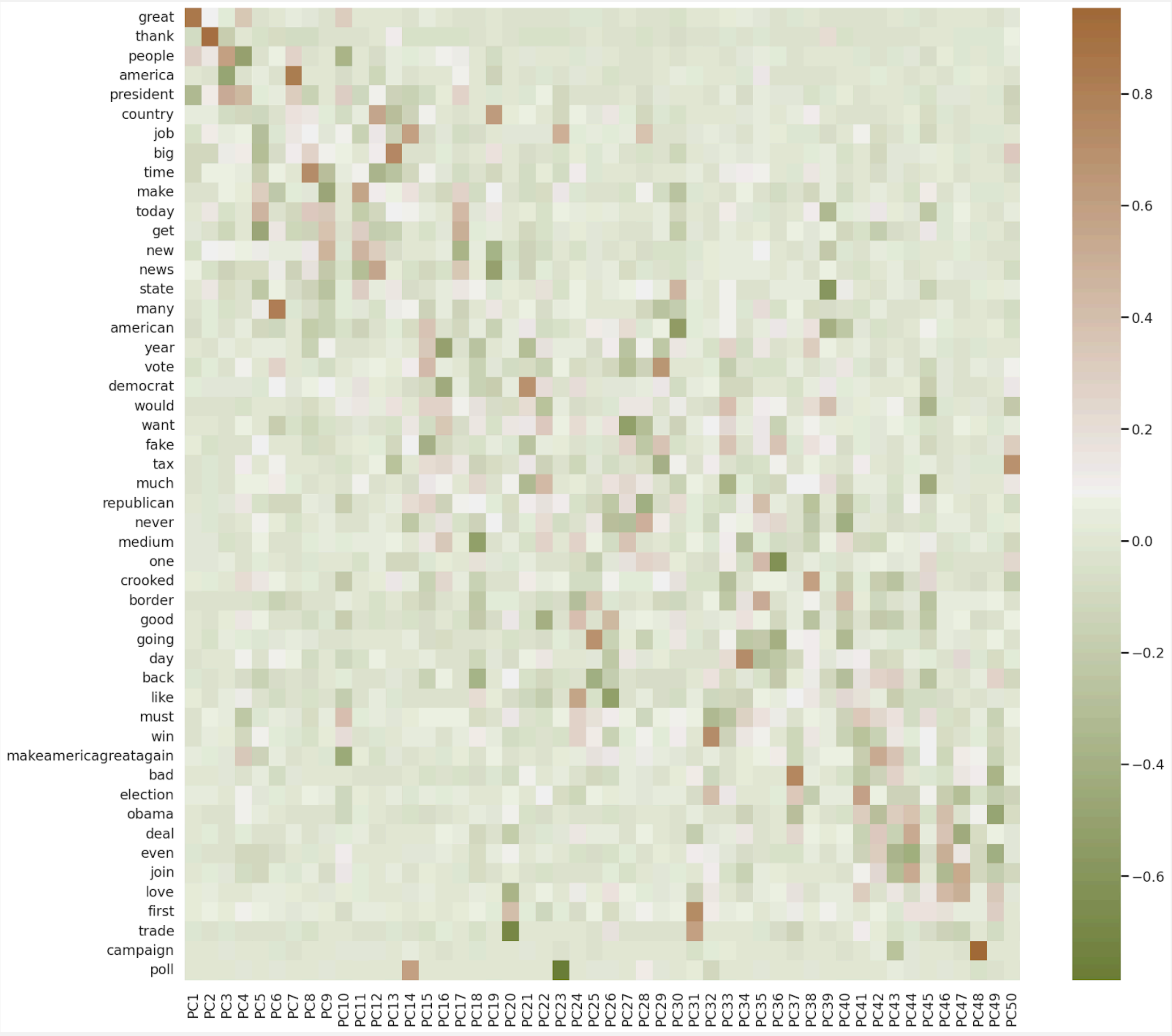
```
PCA(n_components=50)
```

Task 4.3 Examine the PCA

We can examine the PCA results to look at the heatmap. Make a grid plot which shows the various principal component along the x-axis and the individual words along the y-axes. Each grid box should be color-coded based on the sign of the loading factor and how large the square of that value is. Looking at it vertically, you can see which words constitute your principal components. Looking at it horizontally, you can see how individual terms are shared between components. Your answer will look closer to this.



```
### BEGIN ANSWER

pcalabel = []
for i in range(1,51):
    pcalabel.append('PC'+str(i))


plt.figure(figsize=(20,20))


cmap = sns.diverging_palette(100, 400,as_cmap=True)
sns.heatmap(pca.components_, cmap=cmap, yticklabels = top50_words, xticklabels =
pcalabel)
plt.show()

#referenced recitation

    ### END ANSWER
```



```
<Figure size 2000x2000 with 2 Axes>
```

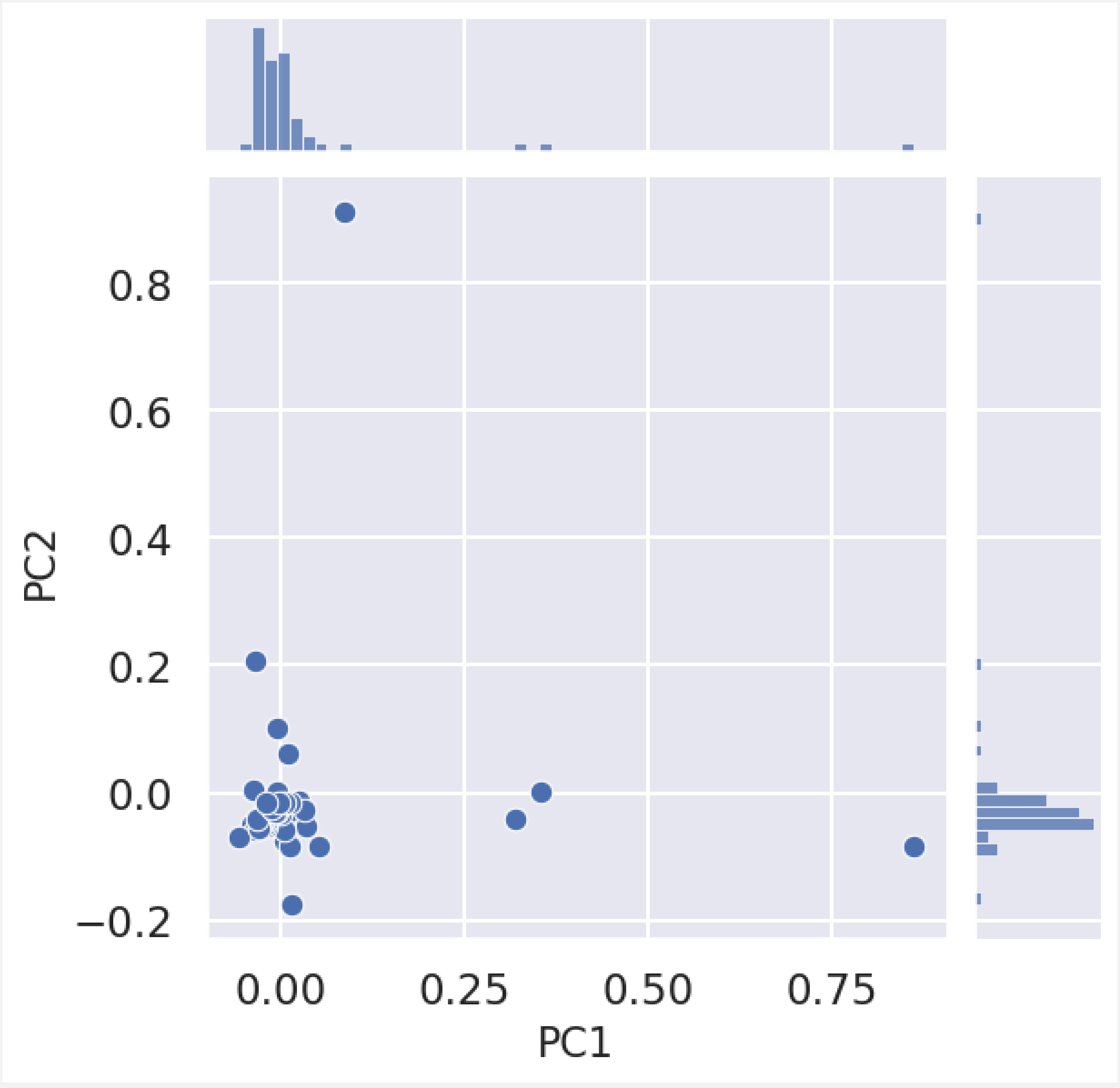
Task 4.4 PCA Compare

We can determine how many words and how many components are needed to do a good visualization. Plot PC1 and PC2 in a 2D plot. The results should be similar to following scatter plot

This is a scatter plot of the values of the components, but with arrows indicating some of the prominent terms as indicated by their loading factors. The values of the loading factors are used to determine the length and direction of these arrows and as such they serve as a way of expressing direction. That is, tweets which use these terms will be moved along the length of those arrows. Shown are the most important parameters.

```
### BEGIN ANSWER
pca_indivs = pca.components_

pca1 = pca_indivs[0]
pca2 = pca_indivs[1]

fig = sns.jointplot(x=pca1, y=pca2)
fig.set_axis_labels('PC1', 'PC2', fontsize=16)

plt.show()
```

```
### END ANSWER
```



```
<Figure size 600x600 with 3 Axes>
```

PART 5 - Twitter Engagement

In this problem, we'll explore which words led to a greater average number of retweets. For example, at the time of this writing, Donald Trump has two tweets that contain the word 'oakland' (tweets 932570628451954688 and 1016609920031117312) with 36757 and 10286 retweets respectively, for an average of 23,521.5.

Your `top_20` table should have this format:

	retweet_count
word	
---	---
jong	40675.666667
try	33937.800000
kim	32849.595745
un	32741.731707
maybe	30473.192308

Task 5.1

Find the top 20 most retweeted words. Include only words that appear in at least 25 tweets. As usual, try to do this without any for loops. You can string together ~5-7 pandas commands and get everything done on one line.

```
#top_20 = df_trump.groupby(['text','retweet_count'])#.filter(lambda x: len(x) >= 25)
#top_20 = top_20['no_punc'].str.split(expand=True).stack()

m = df_trump['retweet_count']


### BEGIN ANSWER
merged_df = tmp.merge(m, left_index=True, right_index=True)

top_20 = merged_df.groupby('word').filter(lambda x: len(x) >= 25)
top_20 = top_20.groupby('word')['retweet_count'].mean()
top_20_final = top_20.to_frame(name='retweet_count')

top_20_final = top_20_final.sort_values(by='retweet_count', ascending=False).head(20)
top_20_final
    ### END ANSWER
```

.dataframe tbody tr th:only-of-type { vertical-align: middle; } .dataframe tbody tr th { vertical-align: top; } .dataframe thead th { text-align: right; }

	retweet_count
word	

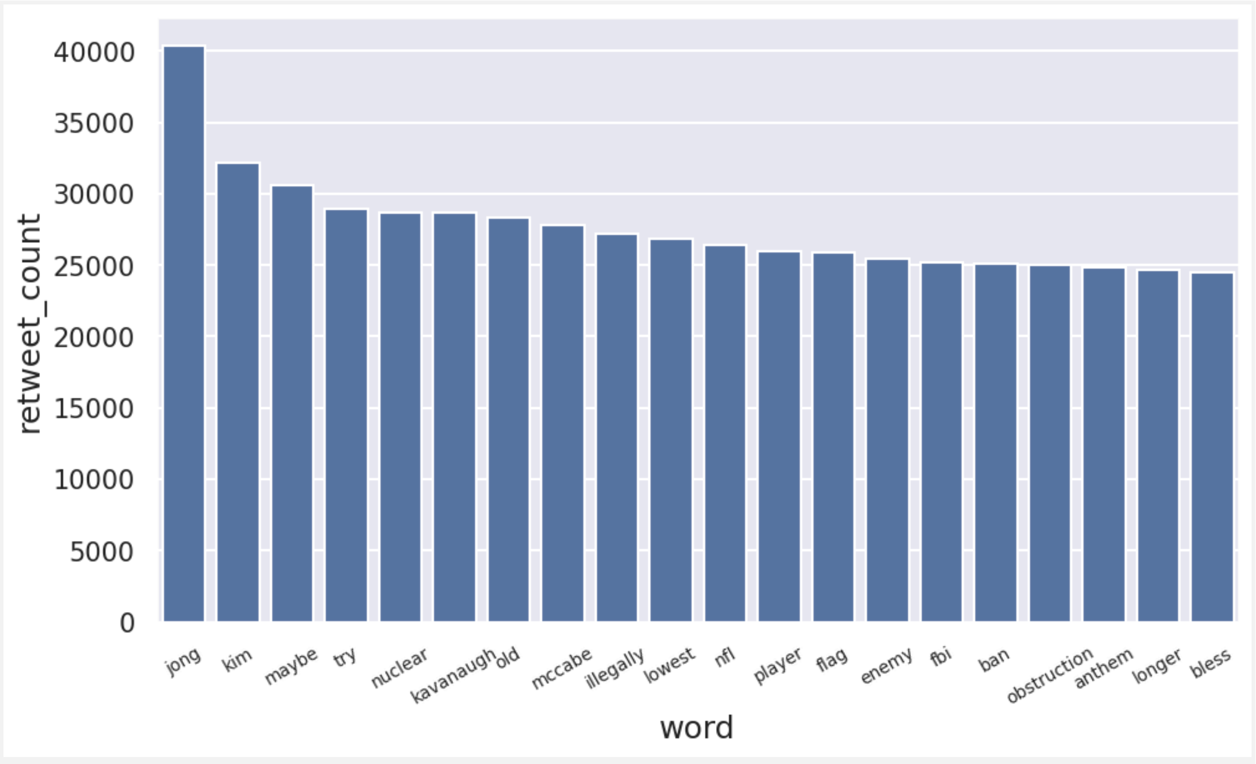
---	---
jong	40408.666667
kim	32148.081633
maybe	30622.000000
try	28923.437500
nuclear	28703.000000
kavanaugh	28651.962963
old	28340.848485
mccabe	27836.870968
illegally	27161.343750
lowest	26819.543478
nfl	26396.540541
player	26019.333333
flag	25897.406250
enemy	25430.428571
fbi	25187.850746
ban	25085.000000
obstruction	25012.594595
anthem	24870.210526
longer	24699.481481
bless	24495.515152

```
            retweet_count
word
jong         40408.666667
kim          32148.081633
maybe        30622.000000
try          28923.437500
nuclear      28703.000000
kavanaugh    28651.962963
old          28340.848485
mccabe       27836.870968
illegally    27161.343750
lowest       26819.543478
nfl          26396.540541
player       26019.333333
flag         25897.406250
enemy        25430.428571
fbi          25187.850746
ban          25085.000000
obstruction  25012.594595
anthem       24870.210526
longer       24699.481481
    bless        24495.515152
```

Task 5.2

Plot a bar chart of your results:

```
### BEGIN ANSWER
fig = plt.figure(figsize = (10, 6))
sns.barplot(x = 'word', y = 'retweet_count', data = top_20_final)


plt.xlabel('word')
plt.ylabel('retweet_count')
plt.yticks(fontsize = 15)
plt.xticks(fontsize = 10, rotation=30)

plt.show()
    ### END ANSWER
```


retweet_count
jong
kim
maybe
try
nuclear
kavanaugh
old
mccabe
illegally
lowest
nfl
player
flag
enemy
fbi
ban
obstruction
anthem
longer
bless
word