Stephen Williams

For my eda, I wanted to answer two connected questions. "Which days do we see a large increase in Russian cumulative equipment losses?" And "For tweets relating to those days, what are the most common words." For example, If I find that Mar 5 was particularly bad for the Russians, I will find the most common words in tweets created on Mar 5.

I choose this topic and questions because of my interest in the ongoing Ukraine-Russian war. Being a zoomer, this is the first time I am conscious of an actual war going on, and I would like to explore datasets related to this event. I also know that the war is not going as expected for Russia, a nuclear power country. I just wanted to visualize their losses in this costly war. Furthermore, I am very interested in doing NLP tasks such as finding common words and sentiment analysis on tweets.

FIRST QUESTION

To answer my first question, I will be using the Kaggle datasets "2022 Ukraine Russia War". More specifically, the "russia_losses_equipment.csv" dataset. It has accumulated equipment loss over time. Through this data set, I can pick dates with considerable cumulated equipment losses. This dataset is updated daily, but I will be only using dates from Feb 25 to Mar 18. One thing to note is that the sources that this dataset uses only account for equipment losses where there is photo evidence. This means the actual equipment losses are higher than the documented amount and are not entirely accurate. It is a fog of war, after all!

The dataset has 14 features. With most of the features being the name of an equipment

```
russia_bad.head()
```

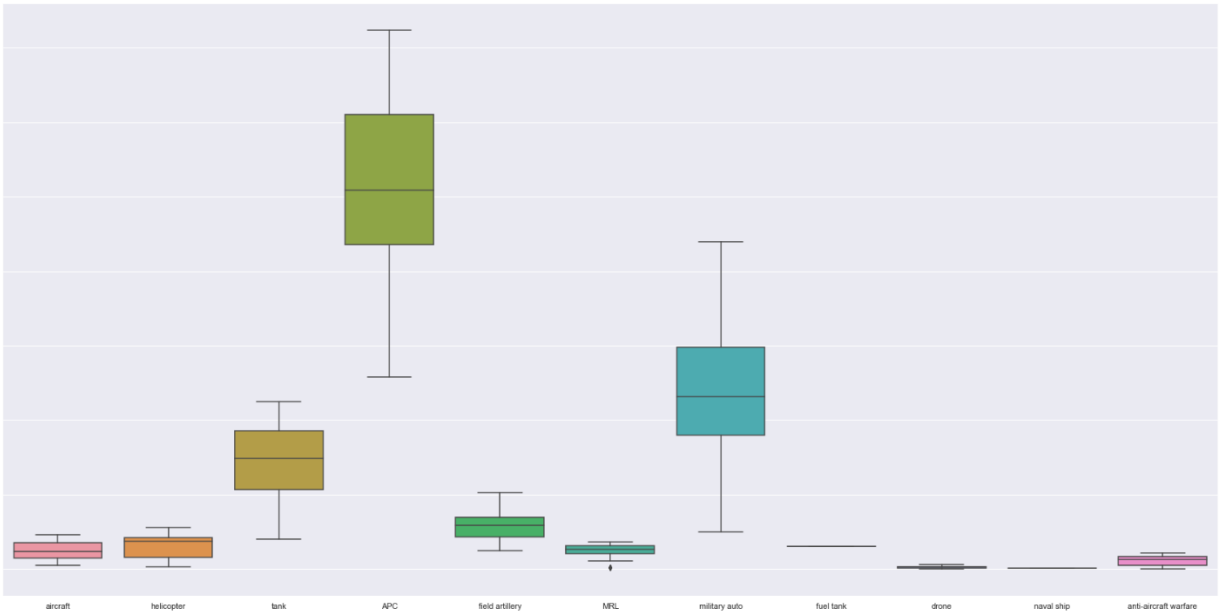
	date	day	aircraft	helicopter	tank	APC	field artillery	MRL	military auto	fuel tank	drone	naval ship	anti-aircraft warfare	special equipment
0	2/25/2022	2	10	7	80	516	49	4	100	60	0	2	0	NaN
1	2/26/2022	3	27	26	146	706	49	4	130	60	2	2	0	NaN
2	2/27/2022	4	27	26	150	706	50	4	130	60	2	2	0	NaN
3	2/28/2022	5	29	29	150	816	74	21	291	60	3	2	5	NaN
4	3/1/2022	6	29	29	198	846	77	24	305	60	3	2	7	NaN

I found null values in one column called "special equipment". 80% of this column were null values, and the non-null values were minimal. I decided to remove this column as it would not impact my question. I also removed the "day" column as the "date" column has more information on time.

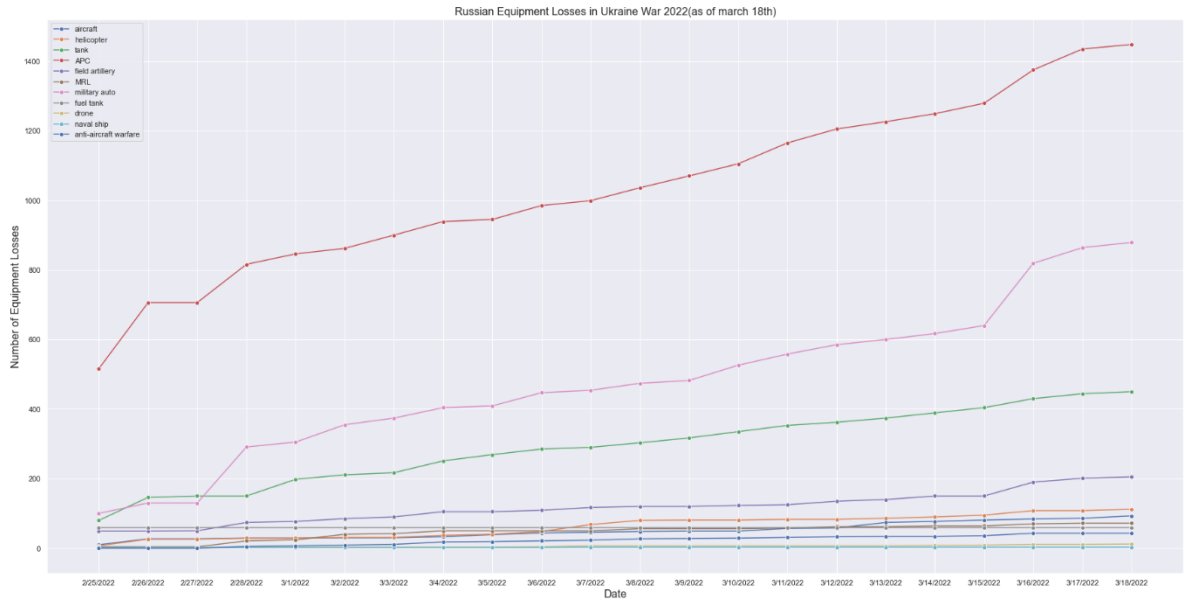
Looking at the boxplot of the remaining variables, I saw "APC" and "military auto" above the rest of the other equipment. They also had longer whiskers. This tells me that they have a higher range and variation. There are very few outliers and some variables like "fuel tank" were

flat. This meant that they had little to no variation. This implies that these pieces of equipment were not used daily.

I also did a correlation heatmap and found that all the variables have a strong positive correlation with each other. When one variable increases, the other's increase as well.



I created a time series plot with "date" as the x variable and the equipment names as the y variables. I observed a positive trend within the plot. However, visually, it was messy as there were overlapping lines. I couldn't choose precise dates where there is an observable leap in the cumulative losses for some or all of the equipment.

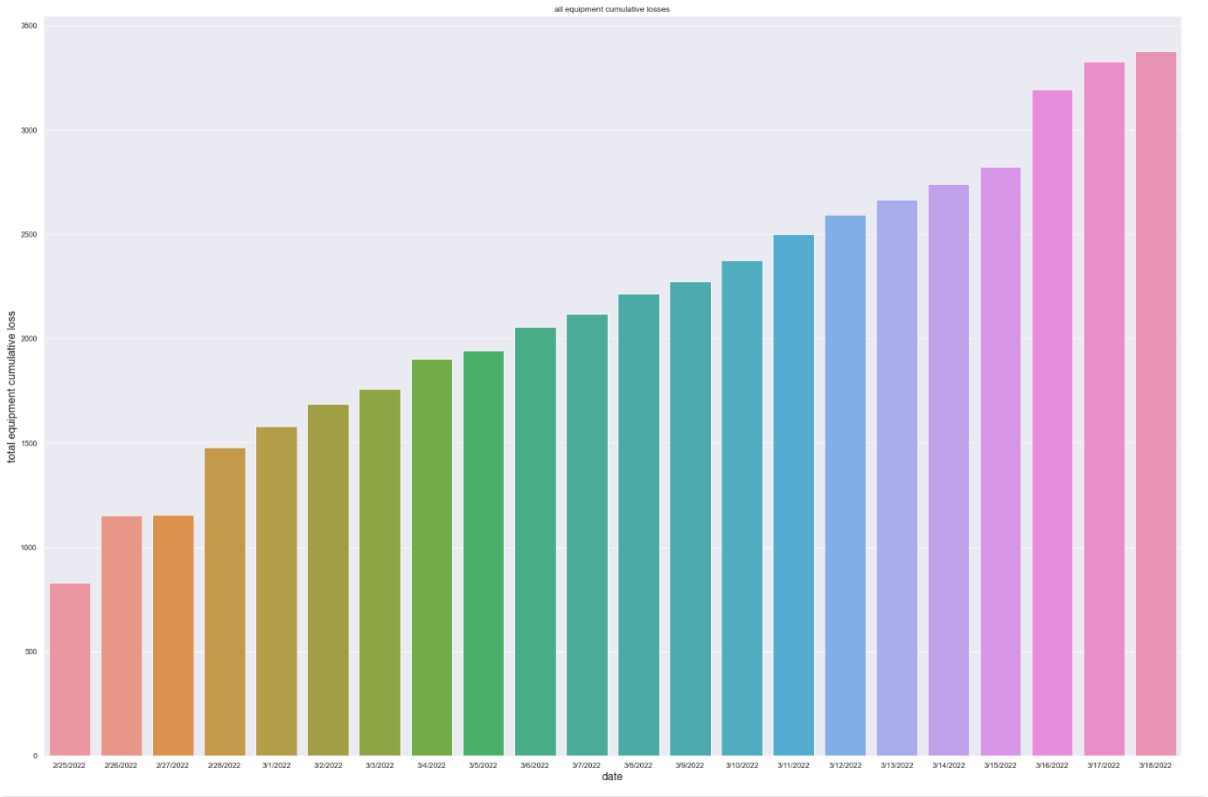


I then summed all the losses across all the equipment and grouped them by date to simplify things. This required me to create a new data frame.

```
#creating dataframe
sum_of_losses = pd.DataFrame()
#adding original dataframe column "date" to new dataframe
sum_of_losses["date"] = russia_bad["date"]
#summing all rows by date
sum_of_losses["total equipment cumulative loss"] = russia_bad[equipment_names].agg(sum,axis=1)

sum_of_losses.reset_index()
```

	index	date	total equipment cumulative loss
0	0	2/25/2022	828
1	1	2/26/2022	1152
2	2	2/27/2022	1157
3	3	2/28/2022	1480
4	4	3/1/2022	1580
5	5	3/2/2022	1688

I used a barplot to display this data frame, and it was much more legible. There is a clear positive trend in the cumulative equipment loss over time. Looking at the barplot, there is no change from Feb 26 to Feb 27. Which could mean there is no fighting going on on those dates. But this may not be accurate since the details are under a fog of war. There are three date intervals where I see a considerable increase/leap in the cumulative loss. These are Feb 25 to Feb 26, Feb 27 to Feb 28, and Mar 15 to Mar 16. I will only be using the end dates of two intervals for simplicity's sake. Selected dates are Feb 28 and Mar 16. Next, I will explore the tweets created on those dates. Doing the barplot analysis and time series plots, I believe equipment "APC" and "military auto" are primarily contributing to the observable large leaps in equipment cumulative losses



SECOND QUESTION

To answer my second/follow-up question, I will be choosing two datasets from “Russia vs Ukraine Tweets Dataset(Daily Updated)” found on Kaggle. These two datasets are “UkraineCombinedTweetsDeduped_MAR16.csv.gzip” and “UkraineCombinedTweetsDeduped_FEB28_part1.csv.gzip”. The dates are picked from the previous analysis. From these datasets, I will be able to analyze the common words used by Twitter users regarding the Ukraine / Russia war for the selected dates.

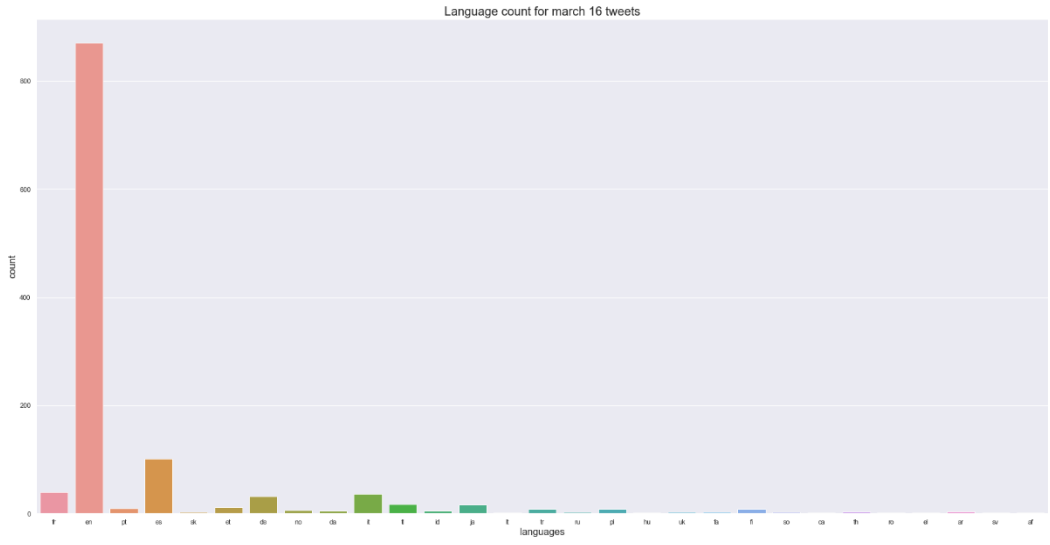
There are 18 features in these two data sets. Each has tens of thousands of observations. I will only be using the “text” column as it is the only one relevant to my question.

```
[20]: march16 = pd.read_csv("UkraineCombinedTweetsDeduped_MAR16.csv.gzip",compression = 'gzip')
      feb28 = pd.read_csv("UkraineCombinedTweetsDeduped_FEB28_part1.csv.gzip",compression = 'gzip')

      C:\Users\Stephen Williams\AppData\Local\Temp\ipykernel_6060\551459047.py:2: DtypeWarning: Columns (15) h
        feb28 = pd.read_csv("UkraineCombinedTweetsDeduped_FEB28_part1.csv.gzip",compression = 'gzip')

[21]: march16.columns

[21]: Index(['Unnamed: 0', 'userid', 'username', 'acctdesc', 'location', 'following',
             'followers', 'totaltweets', 'usercreatedts', 'tweetid',
             'tweetcreatedts', 'retweetcount', 'text', 'hashtags', 'language',
             'coordinates', 'favorite_count', 'extractedts'],
            dtype='object')
```

However, Before I find the common words, I would like to know the proportion of languages used in these tweets. I understand that English is not universal and Twitter has many foreign users. So I created a function to create a language dictionary that shows the language count for the tweets. It uses the langdetect module

```
]: #using langdetect to count the languages used in the tweets and store them in a dictionary
   from langdetect import detect
   from langdetect import DetectorFactory
   DetectorFactory.seed = 0

   def add_to_language_dictionary(string_input,dic):

       try:
           dic[detect(string_input)] += 1
       except KeyError:
           dic[detect(string_input)] = 1
   #testing
   test = {}
   add_to_language_dictionary("i was walking my dog yesterday but i forgot i dont have a dog and i need help",test)
   test
```

I then plotted the language dictionary for each data set. English is the most commonly used language for these tweets. The second language is Spanish. I will be only looking at English tweets from here on out as they represent most of the tweets. And it’s a bit easier to analyze English words.



SAMPLE SIZE

Interestingly enough, when I used sample sizes over 1200 for this language dictionary, I met an error of no feature found within the text. The langdect module is reaching this error, but I am unclear how to solve it. There are no null values for either tweet dataset's "text" columns. So, I will be using only a sample size of 1200 tweets from each dataset.

In addition to only analyzing English words, I cleaned stop words and punctuations from the tweet dataset. It works within a dictionary function that stores the word frequency used in the tweet datasets. I tried my best to manually clean as many stop words as possible, but there be hundreds to maybe thousands of stop words. Inputting them all in an array would be very tedious. Cleaning is hard

Next, I will be cleaning each dataset. Removing stopwords and punctuation

```
]: #creating function to handle the cleaning using regex and stopwords. It is incoporated in the dictionary function
import re
p = re.compile("[.,@;:!\"]")


def add_to_tweet_dictionary(string_input,dic):
    no_punctuation = re.sub(p,"",string_input)
    clean_string = no_punctuation.lower().split()
    stopwords = ["the","a","to","of","in","and","is","with","for","this","from","that","are","we","by","i","she","on"]
    for word in clean_string:
        if word in stopwords:
            continue
        try:
            dic[word] += 1
        except KeyError:
            dic[word] = 1
#just testing
test = {}
string = "i hate sundays and mondays. Mondays blues @yourdad"
add_to_tweet_dictionary(string, test)
test
```

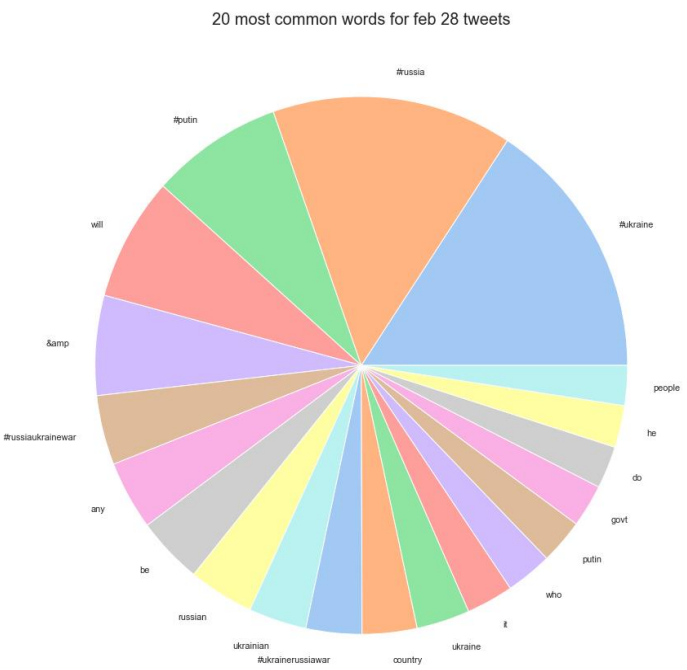
```
]: {'hate': 1, 'sundays': 1, 'mondays': 2, 'blues': 1, 'yourdad': 1}
```

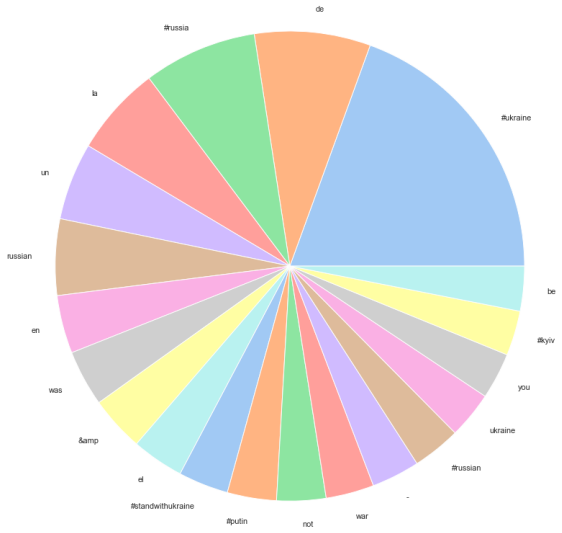
I then use that dictionary to create a separate dictionary to store the word frequency of the top 20 common words for the two datasets. I choose the top 20 words to look at because 20 is a nice number to round up to.

```
#creating the word dictionary for march 16 tweets
march_16_words = {}
count = 0
for tweet in march16_tweets:
    filtering out the non_english tweets
    if detect(tweet) != 'en':
        #pass
    add_to_tweet_dictionary(str(tweet),march_16_words)
    count += 1
    if count == 1200:
        break

# and then making a dictionary for top 20 words
top_20_march16_words ={}
count = 0
for w in sorted(march_16_words, key=march_16_words.get, reverse=True):
    top_20_march16_words[w] = march_16_words[w]
    count += 1
    if count > 19:
        break

top_20_march16_words
```

I then plotted each of the top 20 words for the two datasets.

The most common word for both tweet dates is “#ukraine”. “russia” is the next common word. Unfortunately, there are stop words shown in these plots. Oddly enough, I see some Spanish words like “la” and “de”. My method of cleaning stop words and non-english words has failed. This tells me that the lang detect module is not good enough at filtering out non- English tweets. Or that some tweets that are mostly English have few Spanish words. I tried to filter by words and not by tweets using lang detect but I received poor results. It finds difficulty in classifying short or obscure sentences and it completely fumbles at classifying words. Tweets were relatively easier for it to classify into languages.





Conclusion

- I found three date intervals where I saw a big jump of Russian equipment cumulative losses.
- I chose two end dates from these intervals and proceeded to analysis on tweets related to those dates. The tweets about the ongoing Ukraine/Russia war
- I displayed the top 20 words for these two tweet datasets.

- However, my method of cleaning was ineffective as there were still non-English and stop words in the plots.

Future work

What can be done is to improve the cleaning process of the tweets. Regarding the non-English words, I was too self-observed to realize that there was a column called “language”. I could have created a different filter that would use this column rather than the unreliable lang detect module. There seems to be a nltk module for the stopwords that I could use to import stop words. This library is better to use than me creating a library of thousands of stop words.

Another focus that I would like to do is sentiment analysis on the tweets. While finding the common words could convey a message about public opinion, it isn’t good enough. I would be interested in creating a model to collect and classify data on how many people support Russia or Ukraine. And more samples would be nice