ELON MUSK'S SENTIMENTS TWEETS

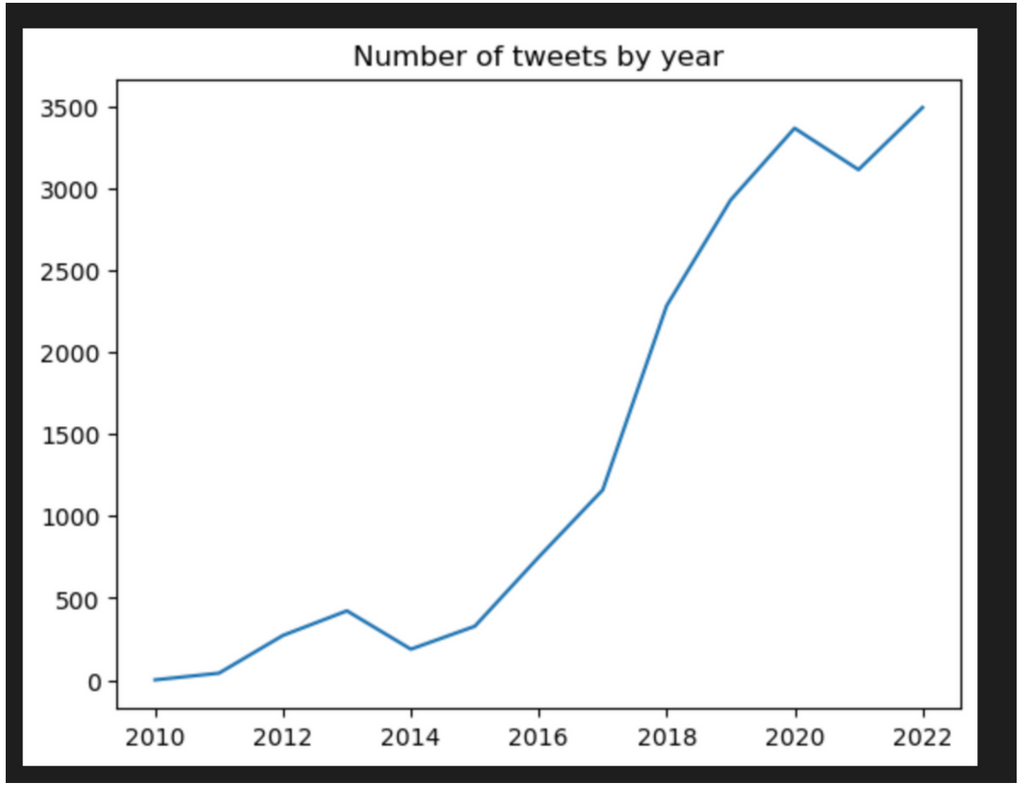
BY : VINCENT AND VINCENT

CODE FOR DATA CLEANING

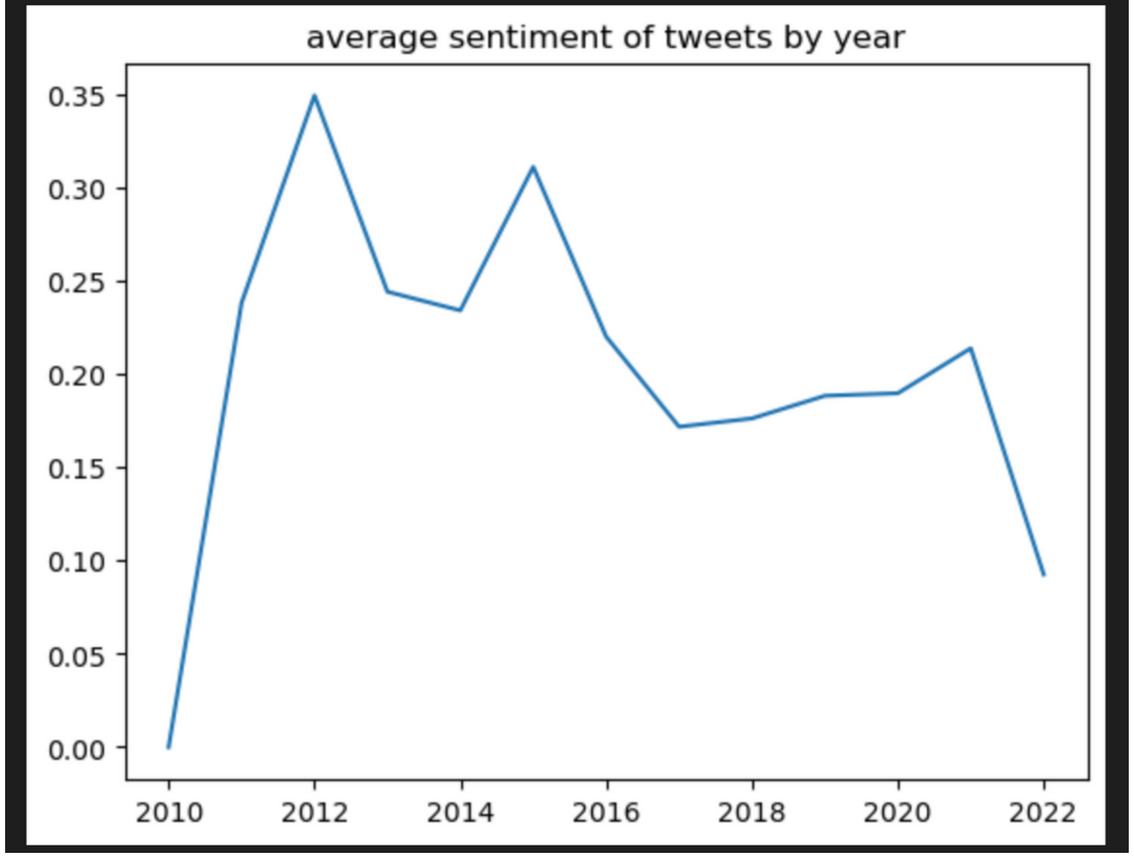
ELON MUSK SENTIMENTS TWEETS DATA

```
import pandas as pd
FILES = ['ElonTweets(Sentiment)_10-28-22.csv', 'ElonTweets(Sentiment).csv', 'ElonTweets(Sentiment)_11-9-22.csv']
df = pd.concat([pd.read_csv(f) for f in FILES], ignore_index = True)
assert (df.groupby('Tweet Id').agg({'sentiment' : set}).sentiment.apply(len) == 1).all()
df = df.drop_duplicates('Tweet Id')

df = df.drop(columns = [col for col in df.columns if 'Unnamed' in col])
def parse_sentiment(value):
    sentiment, proba = value[1:-1].split(',')
    return pd.Series([sentiment[1:-1], float(proba)])
df[['sentiment','sentiment_proba']] =df.sentiment.apply(parse_sentiment)
assert ((df.location == "Twitter HQ")| (df.location.isna())).all()

sentiment_value_dict = {
    'neutral' : 0,
    'positive' : 1,
    'negative' : -1,
}
def sentiment_to_value(sentiment):
    return sentiment_value_dict[sentiment]
assert all( value in sentiment_value_dict.keys() for value in df.sentiment.unique())
df['sentiment_value'] = df.sentiment.apply(sentiment_to_value)


df['Date'] = pd.to_datetime(df['Date'])
df['number_of_tweets'] = 1

def mention_count(mention_list):
    if mention_list == '_':
        return 0
    else:
        return 1 + mention_list.count(',')
df['number_of_mentions'] = df['mentions'].apply(mention_count)
TWEET_METRICS = ['reply count', 'retweet count', 'like count']
```

```
plt.plot(tweets_per_year.index, tweets_per_year['number_of_tweets'])
plt.title("Number of tweets by year")
plt.show()
```

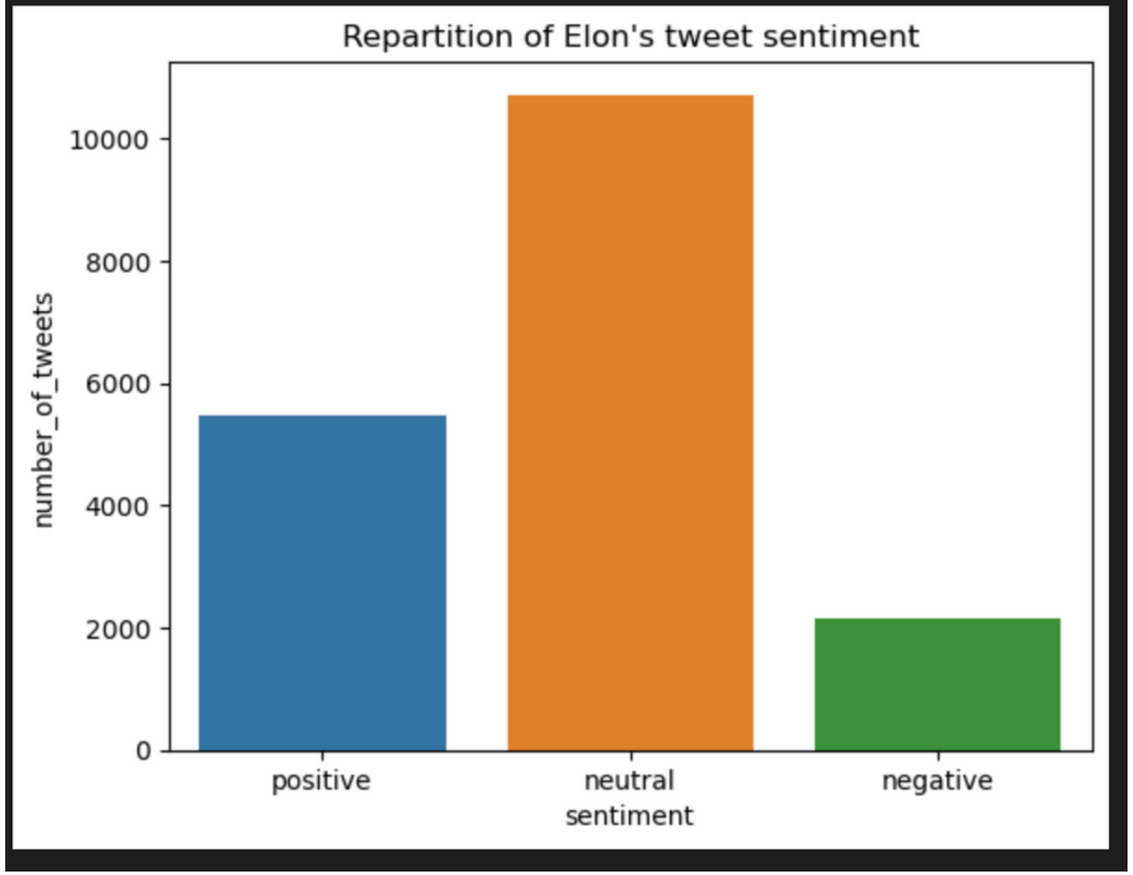


```python
from matplotlib import pyplot as plt

tweets_per_year = df.groupby(df['Date'].dt.year).agg({'sentiment_value' : 'mean', 'number_of_tweets' : 'sum', 'reply count' : 'm

plt.plot(tweets_per_year.index, tweets_per_year['sentiment_value'])
plt.title("average sentiment of tweets by year")
plt.show()
```

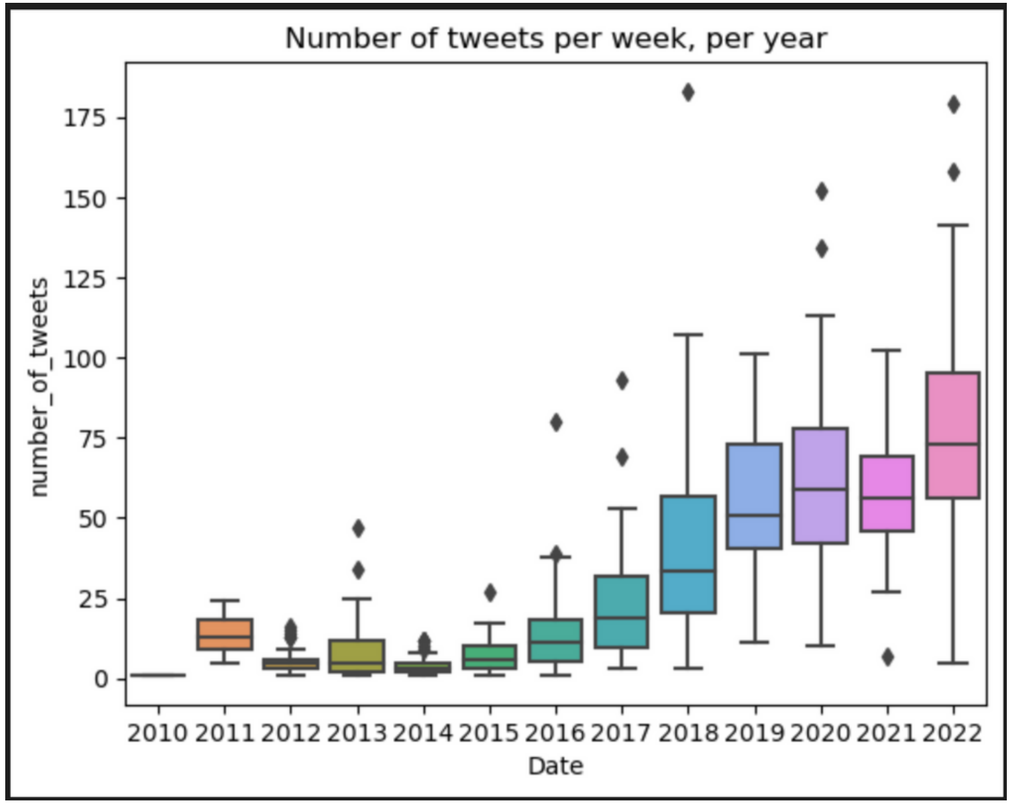



```python
sns.barplot(data = df, x = 'sentiment', y = 'number_of_tweets', estimator=sum)
plt.title("Repartition of Elon's tweet sentiment")
plt.show()
```

✓ 2.2s



```python
import seaborn as sns

df_by_week = df.groupby([df.Date.dt.year, df.Date.dt.isocalendar().week]).agg({col : 'sum' for col in[ 'number_of_tweets'] + TWEET_METR
sns.boxplot(data = df_by_week, x = "Date", y = "number_of_tweets")
plt.title("Number of tweets per week, per year")
plt.show()

for col in TWEET_METRICS:
    sns.boxplot(data = df_by_week, x = "Date", y = col)
    plt.yscale('log')
    plt.title(f"Number of {col} per week, per year")
    plt.show()
```



```python
for col_to_show in TWEET_METRICS:
    QUANTILE_TO_KEEP = 0.90
    max_for_graph = df[col_to_show].quantile(QUANTILE_TO_KEEP)
    df_for_graph = df[df[col_to_show] < max_for_graph].copy()
    sns.displot(data = df_for_graph, x = col_to_show, hue = 'sentiment', stat = 'probability', common_norm = False,   element = "step",
    plt.title(f"Distribution of '{col_to_show}' by tweet sentiment")
    plt.show()
    # We want to show what kind of tweet triggers the most reaction.
    # There doesn't seem to be a significative difference
```