In [4]:
```
import os
import time
import datetime
import pandas as pd
import tweepy as tw
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from nltk import FreqDist
from nltk.corpus import gutenberg
from nltk.corpus import stopwords
from nltk.sentiment import SentimentIntensityAnalyzer
from wordcloud import WordCloud

import creds
# create creds.py and place the creds in this format
# ACCESS_TOKEN = ""
#ACCESS_TOKEN_SECRET = ""
#CONSUMER_KEY = ""
#CONSUMER_SECRET = ""


pd.options.plotting.backend = "plotly"
```

Analyzing Elon Musk's Twitter

In [5]:
```
# auth with twitter
auth = tw.OAuthHandler(creds.CONSUMER_KEY, creds.CONSUMER_SECRET)
auth.set_access_token(creds.ACCESS_TOKEN, creds.ACCESS_TOKEN_SECRET)
api = tw.API(auth, wait_on_rate_limit=True)
```

In [6]:
```
max_tweets = 5000
account_name = "elonmusk"

# Collect tweets
results = api.user_timeline(screen_name = account_name, count = max_twee
# storing the data after fetching from the api
data = [[tweet.created_at , tweet.full_text] for tweet in results]
text_only = [tweet.full_text for tweet in results]
# creating pandas dataframe to stode the data
tweet_df = pd.DataFrame(data)

tweet_df.columns = ['date', 'text']
```

In [7]:
```
# to clean the tweets and get meaningful infomation
stop_words = stopwords.words('english')
stop_words = set(stop_words)  # removing any duplicates
stop_words = list(stop_words)  # converting back to the list
stop_words.append("This")
stop_words.append("I")
```

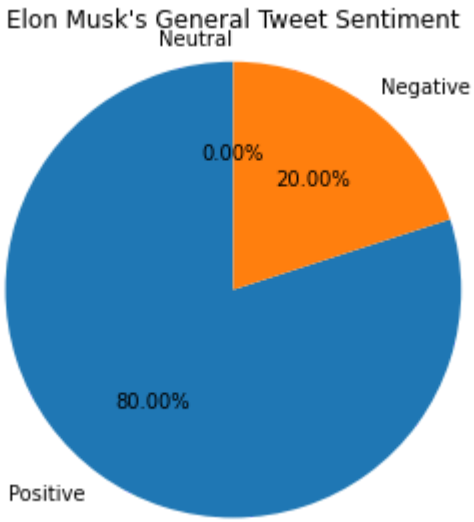
In [8]:
```
# for removing unnecessary punctuations
punctuations = [".", "!", "?", ",", ";", ":", "-", "[", "]", "{", "}", '
                "<", ">", "`", "^", "_","—", "|", "#", "$", "%", "+", "=
                " "]
```

In [9]:
```
# cleaning the data using stopwords and punctuations
clean_words = []
for tweet in text_only:
    for item in tweet.split():
        if item not in stop_words and item not in punctuations and item|
            item = item.replace(",","")
            item = item.replace(".","")
            item = item.replace("!","")
            clean_words.append(item)
```

In [10]:
```
# this if for sentiments analysis
positive = 0
negative = 0
neutral = 0
# initializing the object
sia = SentimentIntensityAnalyzer()

# calculating score and adding to score
for tweet in text_only:
    val = sia.polarity_scores(tweet)
    val.pop('compound', None)
    great = max(val, key=val.get)
    if great =='pos':
        positive +=1
    elif great == 'neg':
        negative +=1
    else:
        neutral +=0

# creating the pie chart
labels=['Positive', 'Negative','Neutral']
sizes=[positive,negative,neutral]
colors=['yellowgreen',"gold","red"]
chart=plt.pie(sizes,labels=labels, startangle=90, autopct='%.2f%%')
plt.title("Elon Musk's General Tweet Sentiment")
plt.axis("equal")
plt.tight_layout()
plt.show()
```



In [11]:

```python
#convert list to string and generate the word cloud
unique_string=(" ").join(clean_words)
wordcloud = WordCloud(width = 1000, height = 500).generate(unique_string
plt.figure(figsize=(15,8))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
plt.close()
```

In [12]:

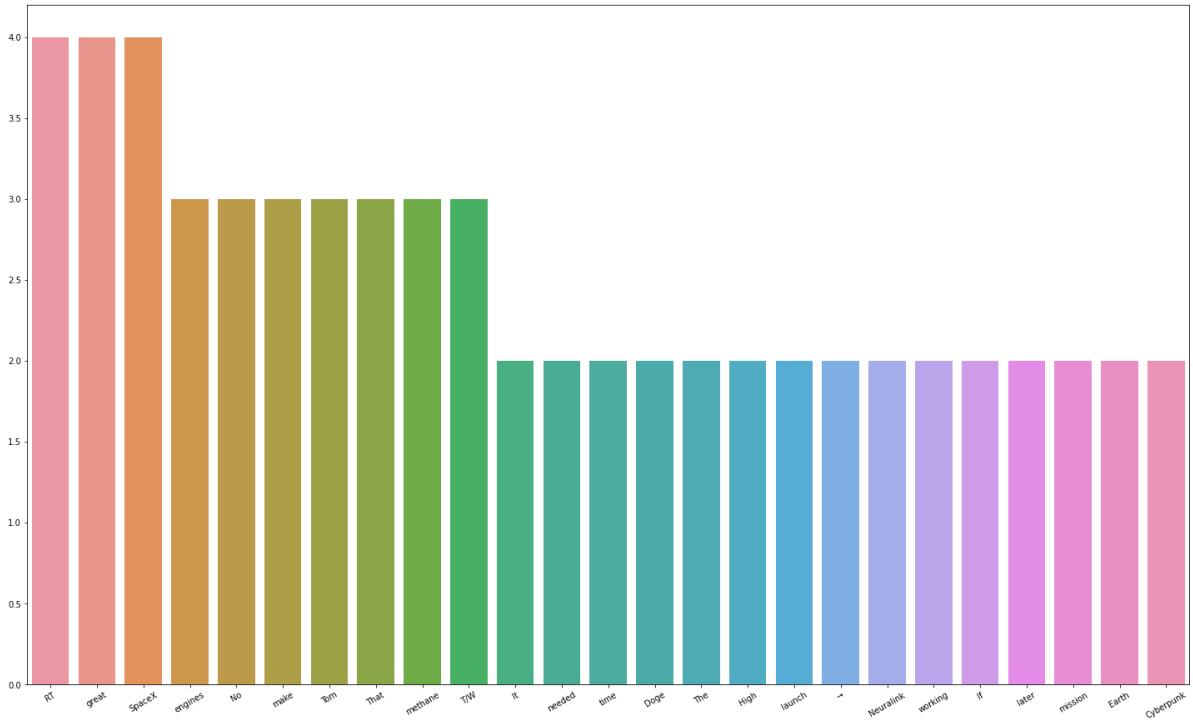
```python
# get word frequencies from the filtered list of words from cleaned_word
punct_freq = FreqDist(clean_words)

#15 Most Common Words in Macbeth (no stopwords and no punctuation):
all_fdist = punct_freq.most_common(25)
```

In [13]:
```
all_fdist = pd.Series(dict(all_fdist))

# Setting figure, ax into variables
fig, ax = plt.subplots(figsize=(25, 15))

# Seaborn plotting using Pandas attributes + xtick rotation for ease of
all_plot = sns.barplot(x=all_fdist.index, y=all_fdist.values, ax=ax)
plt.xticks(rotation=30)
plt.title("")
plt.show()
```



It is intresting that "Doge" is quite high on the list, Let's look into the correlation between dogecoin and Elon Musk

In [14]:
```
# getting data for analysis
TICKR="DOGE-USD"
PERIOD_1=int(time.mktime(datetime.datetime(2021,1,1,23,59).timetuple()))
PERIOD_2=int(time.mktime(datetime.datetime(2021,3,16,23,59).timetuple()))
INTERVAL="1d"

url_finance = f'https://query1.finance.yahoo.com/v7/finance/download/{TI
df = pd.read_csv(url_finance)
copy_df = df
```

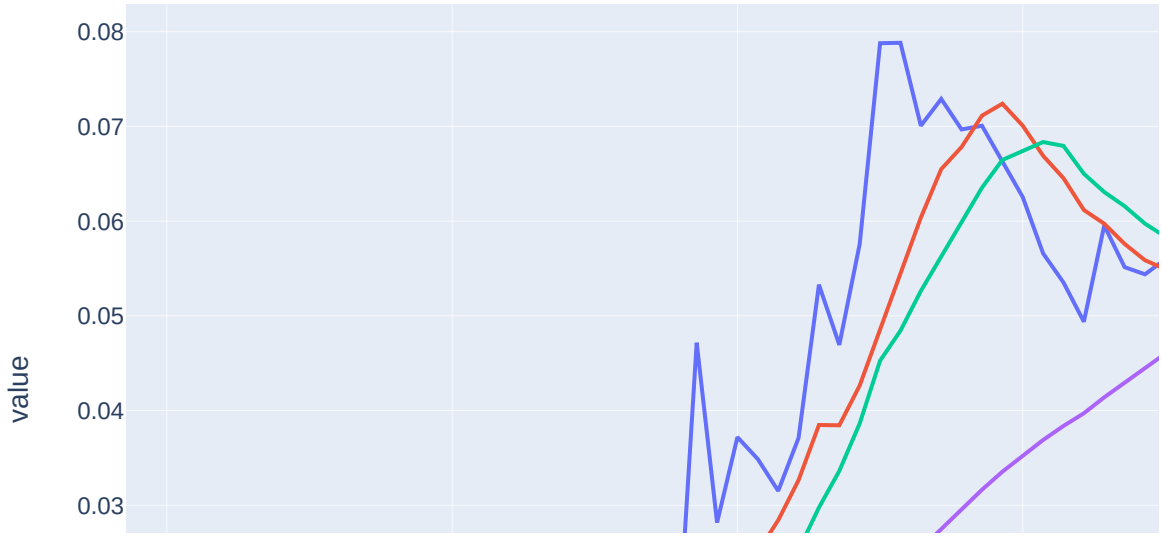
In [15]:
```
df.head()
```

Out[15]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-01-01	0.004681	0.005685	0.004615	0.005685	0.005685	2.289615e+08
1	2021-01-02	0.005686	0.013698	0.005584	0.010615	0.010615	3.421563e+09
2	2021-01-03	0.010602	0.013867	0.009409	0.009771	0.009771	2.707004e+09
3	2021-01-04	0.009785	0.011421	0.007878	0.009767	0.009767	1.372399e+09
4	2021-01-05	0.009767	0.010219	0.008972	0.009920	0.009920	6.872561e+08

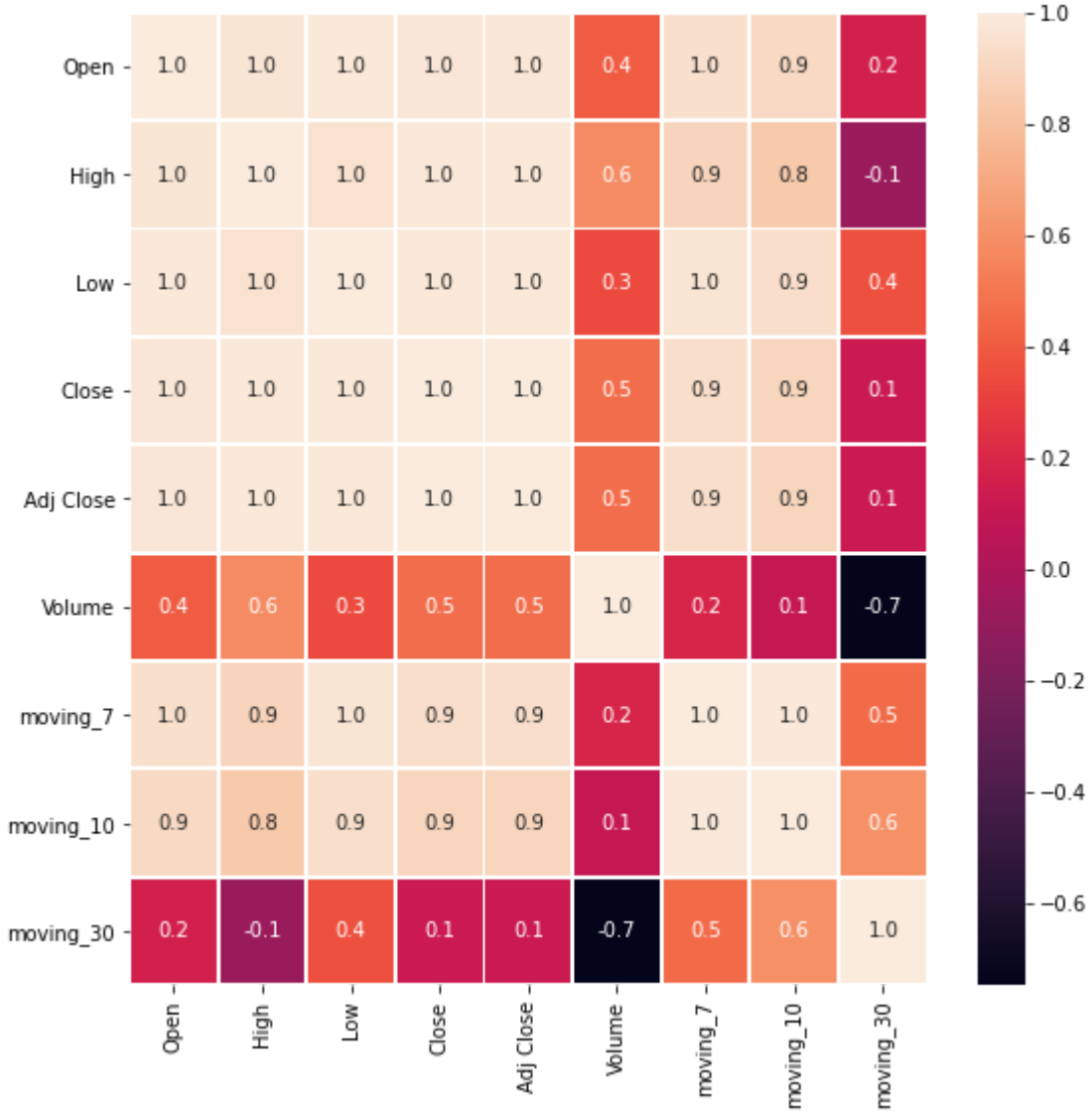
In [16]:
```
# calculating the moving average of the data to get general
# idea of the data
df["moving_7"] = df['Close'].rolling(window=7).mean()
df["moving_10"] = df['Close'].rolling(window=10).mean()
df["moving_30"] = df['Close'].rolling(window=30).mean()
```

In [17]:
```
# plotting the line graph
fig = px.line(df, x='Date', y=['Close','moving_7', 'moving_10','moving_3
fig.show()
```



In [18]:
```
df.Date = pd.to_datetime(df.Date)
df = df.set_index('Date')
```

In [19]:
```
# Displays the correlation between all the variables
df.corr()
f,ax = plt.subplots(figsize=(9, 9))
sns.heatmap(df.corr(), annot=True, linewidths=.9, fmt= '.1f',ax=ax)
plt.show()
```



In [20]:
```
# getting the tweets from elon and filtering only dogecoin related twee
doge_tweets = []
for tweet in tw.Cursor(api.user_timeline, screen_name='@elonmusk', exclu
    if "doge" in tweet.text.lower() or "moon" in tweet.text.lower() or '
        doge_tweets.append([str(tweet.created_at)[0:10],
        copy_df[copy_df['Date'] == str(tweet.created_at)[0:10]].iloc[0]|
        tweet.text])
```

In [21]:
```
# converting data into dataframe
doge_tweets = pd.DataFrame(doge_tweets)
doge_tweets.columns = ['date', 'doge_price', 'tweet_text']
```

In [22]:
```
# plotting
fig = px.area(copy_df, x='Date', y="Close", template='plotly')
fig.add_trace(go.Scatter(mode="markers", x=doge_tweets["date"],
                         y=doge_tweets["doge_price"], name="Elon tweets
                        text=doge_tweets['tweet_text'],
                        textposition='bottom center',
                        textfont=dict(color='#23C552'),))
fig.show()
```