Analysing and Visualising data using python.

Introduction

Python is one of the most commonly used tools for data analysis and it has some very good libraries providing powerful methods for use in any stage of the analysis including analysis and visualisation.

Today, we will use the pandas library to analyse an already clean twitter archive dataset that contains dog reviews from the WeRateDogs twitter account.

We are interested in answering the following questions:

1. *What are the top 10 breeds with the most likes and retweets?*
2. *What dog ratings have the most likes and retweets?*
3. *Might there be correlation between the dog rating and counts of likes and retweets?*
4. *What is the trend of likes and retweets over time?*

To start us off, we first import the libraries we need, which are pandas, matplotlib and seaborn to provide a bit of formating.

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_theme()
```

We then read the file we want to analyse

```
Enhanced_twitter_archive = pd.read_csv(‘twitter_archive_master.csv’)
enhanced_twitter_archive.head(3)
```

We can check for the top 10 most likes and retweets by breed using the **groupby** pandas method and plot the results using the **plot** still from pandas.[1]

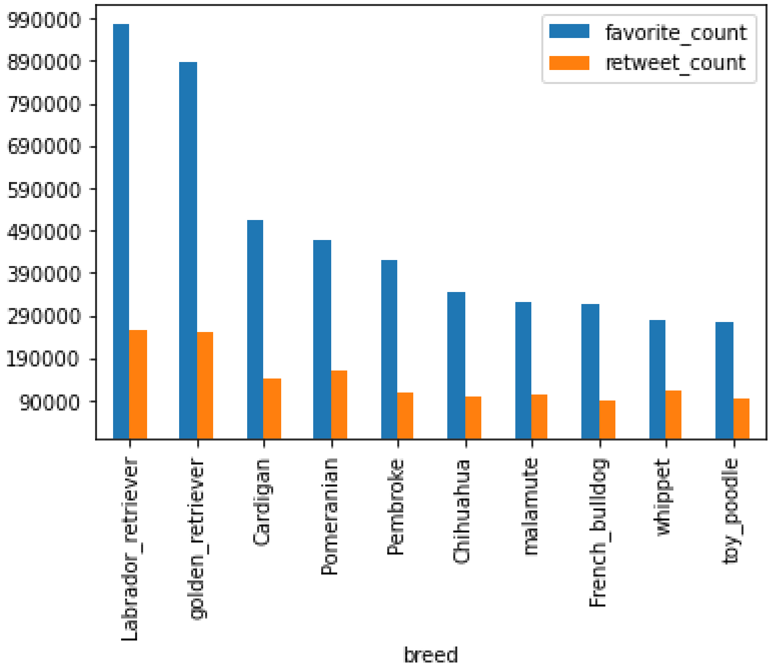
```
df = enhanced_twitter_archive.copy()# precaution in case I mess up and have to start again.

favourite_breeds = df.groupby('breed')[['favorite_count','retweet_count']].sum().sort_values(['favorite_count','retweet_count'], ascending=[False, False])

```

[1] Pandas plotting uses matplotlib at its core and shares some apis with core matplotlib.

```
favourite_breeds.head(10).plot( y = ['favorite_count','retweet_count'],kind = 'bar',
yticks = range(90000, 1000000,100000))
```

Which yields the following output:

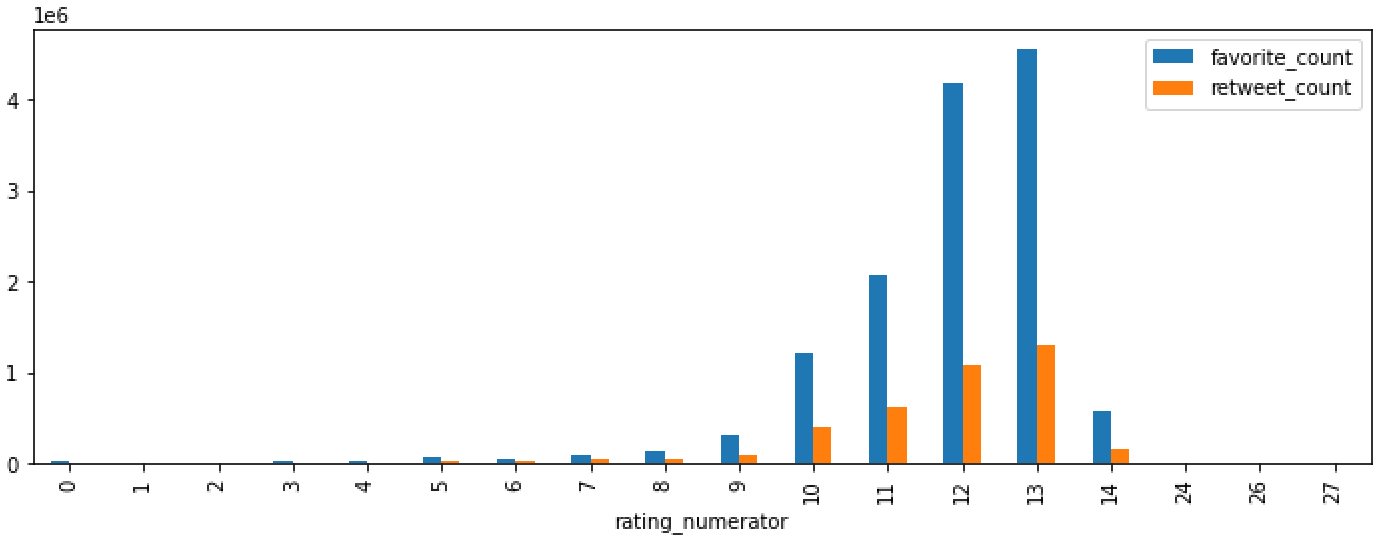
```
<AxesSubplot:xlabel='breed'>
```



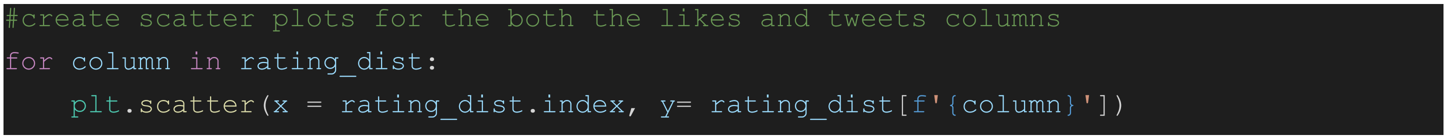
Next, still using groupby we can check for likes and retweets for different ratings:

```
#group by rating_numerator and check for sum of likes and retweets
rating_dist = df.groupby('rating_numerator')[['favorite_count',
'retweet_count']].sum()
rating_dist.plot(kind = 'bar', figsize=(12,4))
```

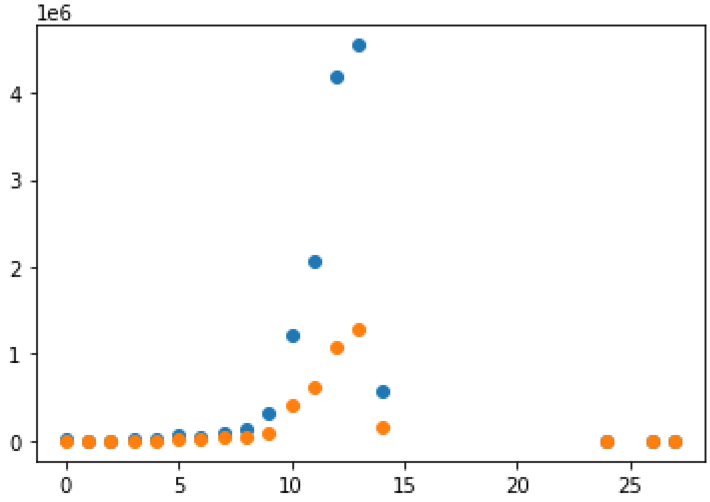
And we get:



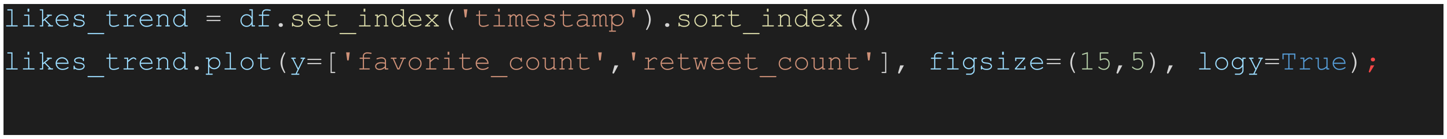
We then create a scatter plot using matplotlib which we imported as plt.



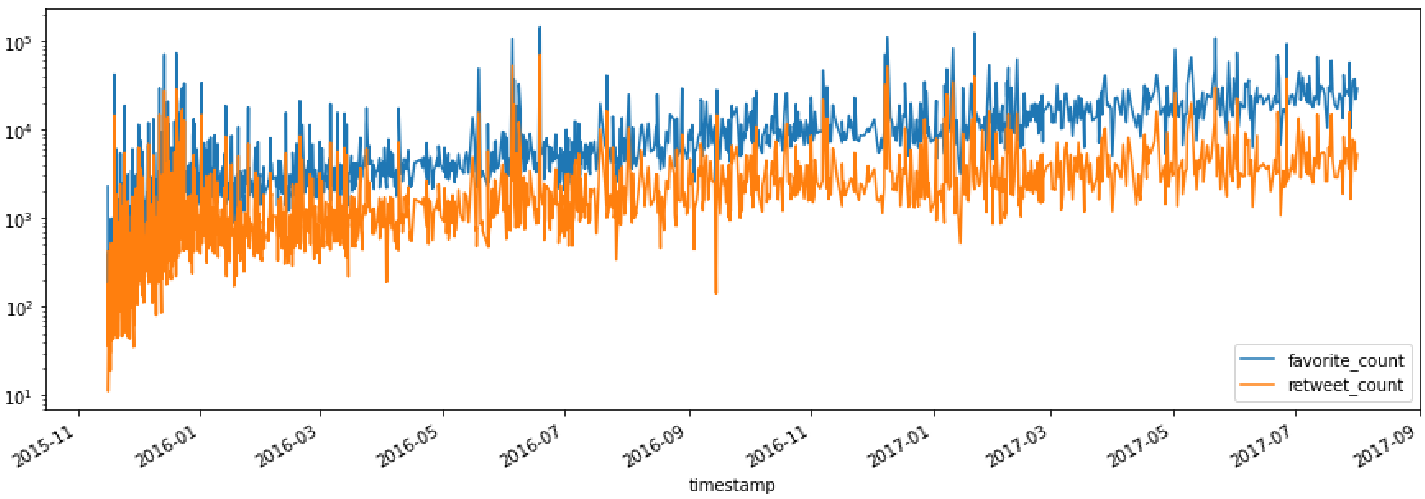
```
#create scatter plots for the both the likes and tweets columns
for column in rating_dist:
    plt.scatter(x = rating_dist.index, y= rating_dist[f'{column}'])
```



Lastly we use the plot function in pandas to check for the change in likes and retweets over time as follows:



```
likes_trend = df.set_index('timestamp').sort_index()
likes_trend.plot(y=['favorite_count','retweet_count'], figsize=(15,5), logy=True);
```



From our analysis above we found out that:

Based on the sums of likes and retweets counts for each breed in our sample, the **Labrador** and **golden** retrievers take the day.[2]

From our sample, there seems to be correlation between the **rating** and both the **likes** and **retweet** counts

We can see that there are a few spikes here and there, but the tweets and retweets are rising albeit slowly over time.

These analysis is not conclusive and the sample size is not adequate enough, but the methods used can be applied to whatever data set you have to make aggregations and plot visualisations.

[2] We have to note however that we have not looked at the distribution of tweets, where we may probably find that a few tweets contribute to most likes and retweets.