

act_report

January 10, 2023

0.1 Reporting: act_report by Abdulraqib Omotosho

This document communicates all the insights and displays the visualizations I produced from the data I wrangled.

0.1.1 Insights and Visualizations

- Most popular dog names: Tucker, Lucy and Cooper
- Most popular dog type: Pupper
- Dog type with the highest confidence interval: Doggo
- Most popular dog breeds: golden_retriever
- Most rated dog breeds: golden_retriever
- Highest Mean retweet counts of dog types: doggo at 4914.918033
- Highest Mean retweet counts of dog breeds: Bedlington_terrier at 6562.333333
- The average count of the most liked dog types: doggo at 15632.688525
- The average count of the most liked dog breeds: Bedlington_terrier at 20995.666667
- The most popular source of tweets: Twitter for iPhone (1584)
- The highest average retweet counts by on dog names: Stephan at 50687.00

Importing the libraries used in the process

```
[2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style='white')
import warnings
warnings.simplefilter(action='ignore')

df = pd.read_csv('twitter_archive_master.csv')
```

- Most popular dog names

```
[3]: df['names'].value_counts()
```

```
[3]: Tucker      9
Lucy            9
Cooper         9
Penny          8
Charlie        8
..
```

```

Remus      1
Mabel      1
Misty      1
Mosby      1
Christopher 1
Name: names, Length: 837, dtype: int64

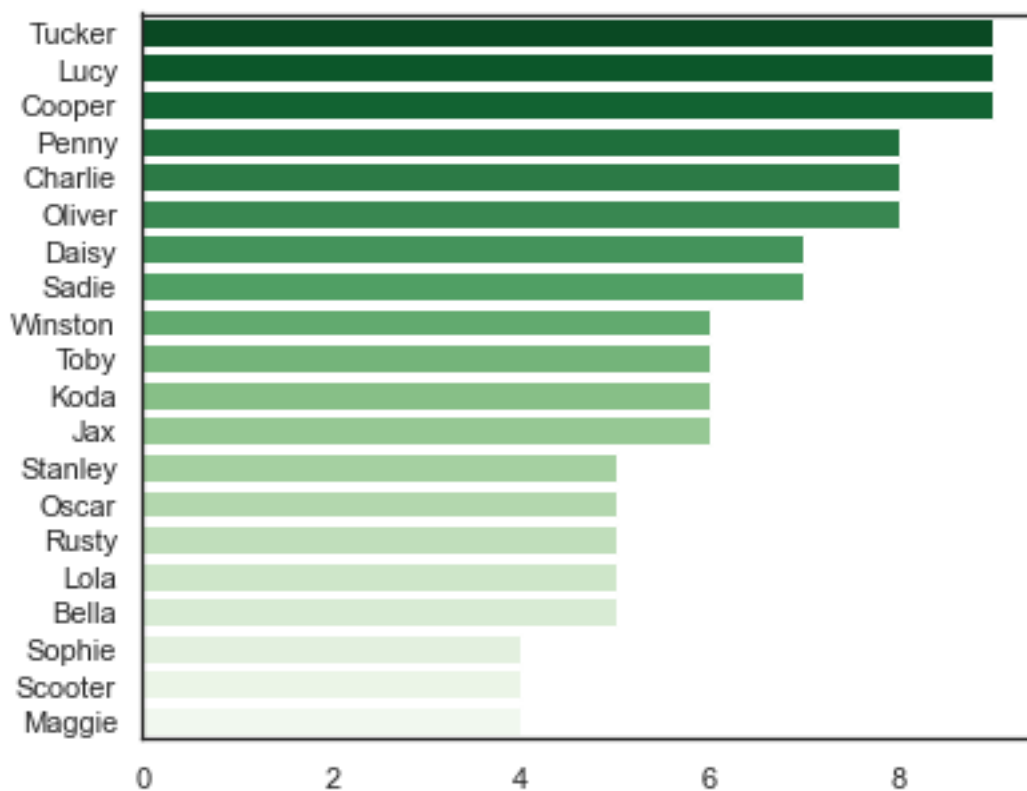
```

```

[4]: plt.figure(figsize=(6, 5))
     x = df['names'].value_counts().head(20)
     sns.barplot(x.values, x.index, palette='Greens_r')

```

[4]: <AxesSubplot:>



- What dog types are the most popular

```

[5]: df['types'].value_counts()

```

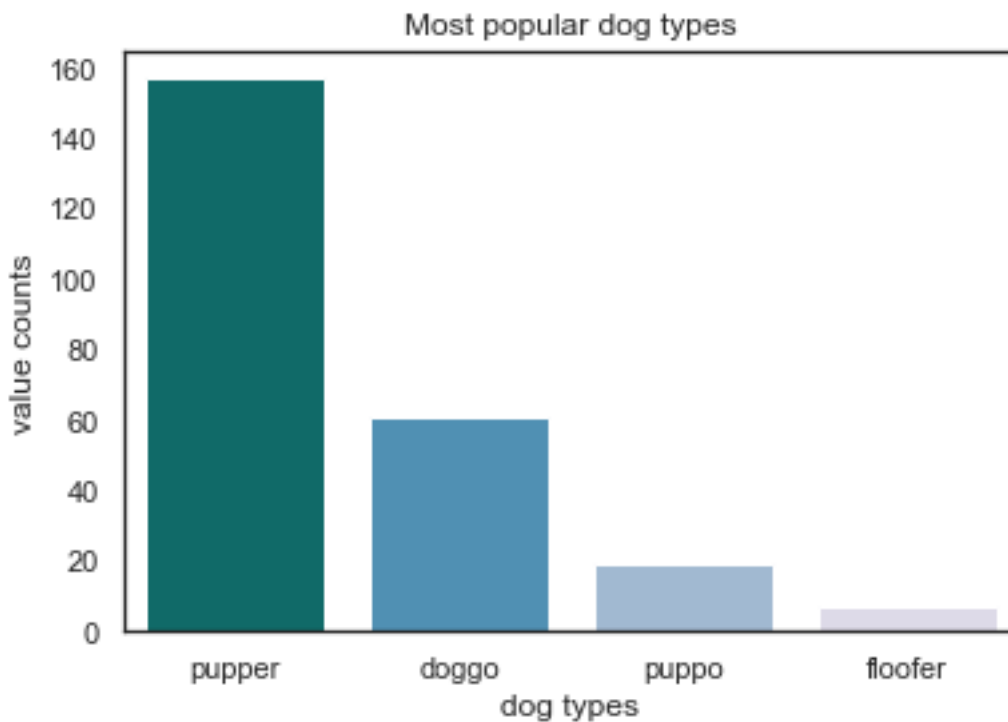
```

[5]: pupper      157
     doggo       61
     puppo       19
     floofer       7
     Name: types, dtype: int64

```

```
[6]: x = df['types'].value_counts()
sns.barplot(x.index, x.values, palette='PuBuGn_r').set(title='Most popular dog_
↳types', xlabel='dog types', ylabel='value counts')
```

```
[6]: [Text(0.5, 1.0, 'Most popular dog types'),
Text(0.5, 0, 'dog types'),
Text(0, 0.5, 'value counts')]
```



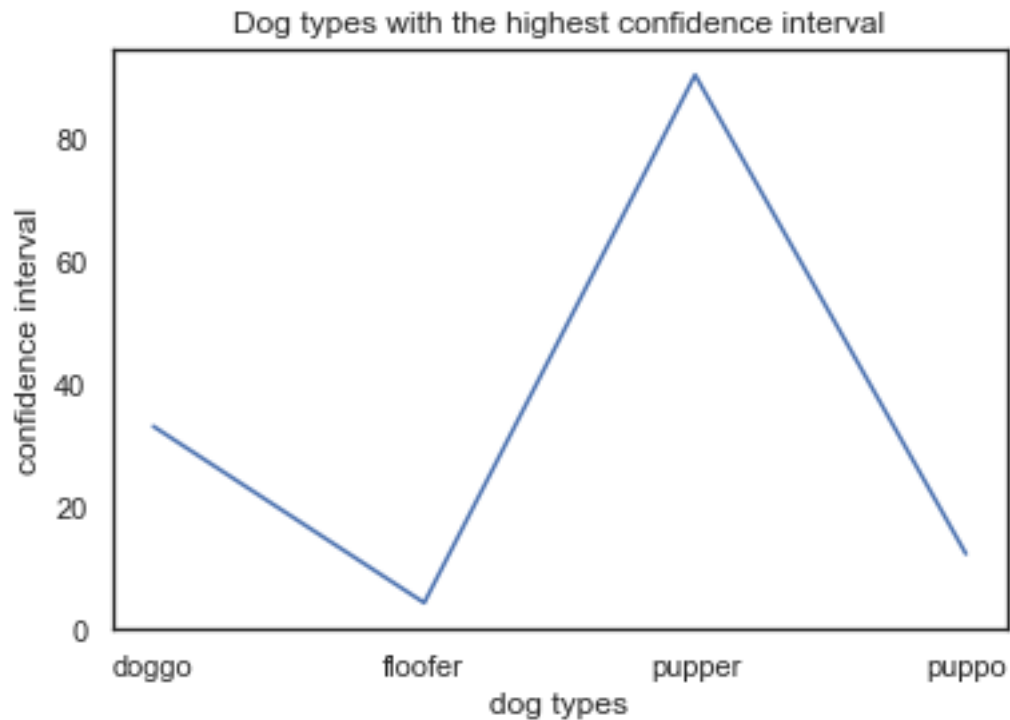
- Which dog has the highest confidence interval

```
[7]: df.groupby('types')['confidence'].sum()
```

```
[7]: types
doggo      33.137328
floofer     4.264427
pupper     90.599764
puppo      12.254248
Name: confidence, dtype: float64
```

```
[8]: x = df.groupby('types')['confidence'].sum()
sns.lineplot(x.index, x.values, ).set(title='Dog types with the highest_
↳confidence interval', xlabel='dog types', ylabel='confidence interval')
```

```
[8]: [Text(0.5, 1.0, 'Dog types with the highest confidence interval'),
      Text(0.5, 0, 'dog types'),
      Text(0, 0.5, 'confidence interval')]
```



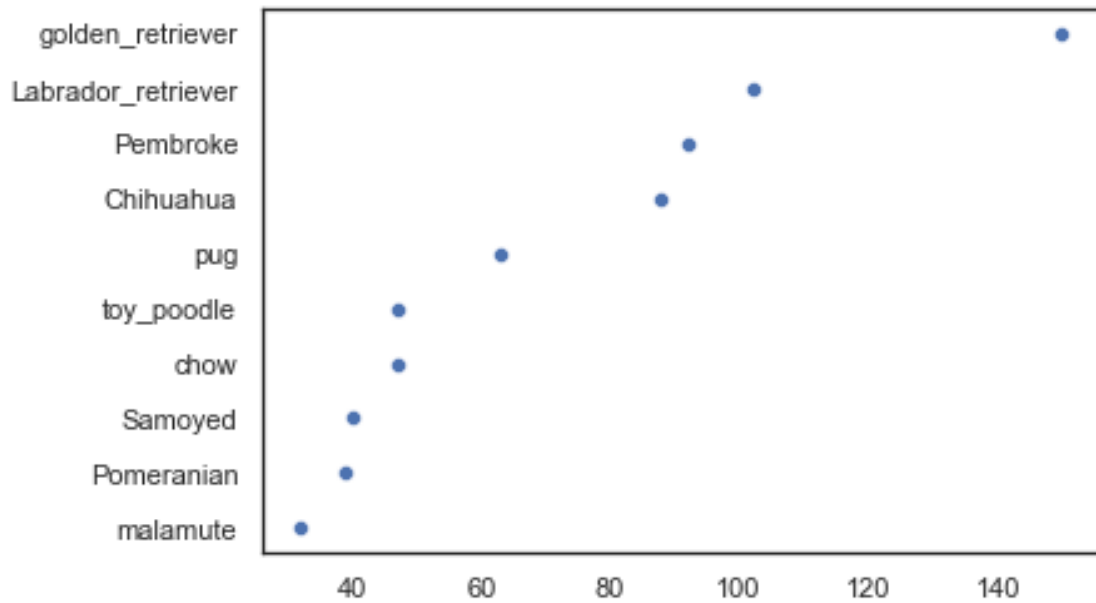
- Most popular dog breeds

```
[9]: df['dog_breed'].value_counts().head(10)
```

```
[9]: golden_retriever    150
      Labrador_retriever 102
      Pembroke           92
      Chihuahua          88
      pug                63
      toy_poodle         47
      chow               47
      Samoyed            40
      Pomeranian         39
      malamute           32
      Name: dog_breed, dtype: int64
```

```
[10]: x = df['dog_breed'].value_counts().head(10)
      sns.scatterplot(x.values, x.index)
```

```
[10]: <AxesSubplot:>
```



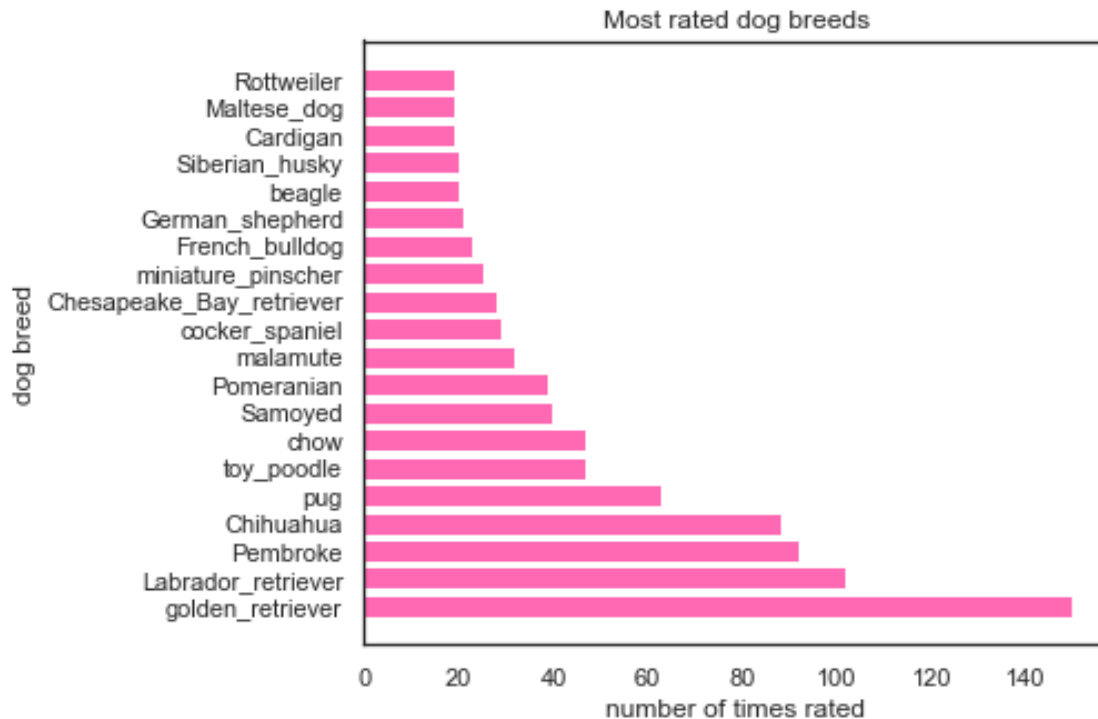
- Most rated dog breeds

```
[11]: df.groupby('dog_breed')['rating'].size().sort_values(ascending=False).head(20)
```

```
[11]: dog_breed
golden_retriever      150
Labrador_retriever    102
Pembroke              92
Chihuahua             88
pug                   63
toy_poodle            47
chow                  47
Samoyed               40
Pomeranian            39
malamute              32
cocker_spaniel        29
Chesapeake_Bay_retriever 28
miniature_pinscher    25
French_bulldog        23
German_shepherd       21
beagle                20
Siberian_husky        20
Cardigan              19
Maltese_dog           19
Rottweiler            19
Name: rating, dtype: int64
```

```
[12]: x = df.groupby('dog_breed')['rating'].size().sort_values(ascending=False).
      ↪head(20)
      plt.figure(figsize=(6, 5))
      plt.barh(x.index, x.values, color='hotpink')
      plt.xlabel('number of times rated')
      plt.ylabel('dog breed')
      plt.title('Most rated dog breeds')
```

```
[12]: Text(0.5, 1.0, 'Most rated dog breeds')
```



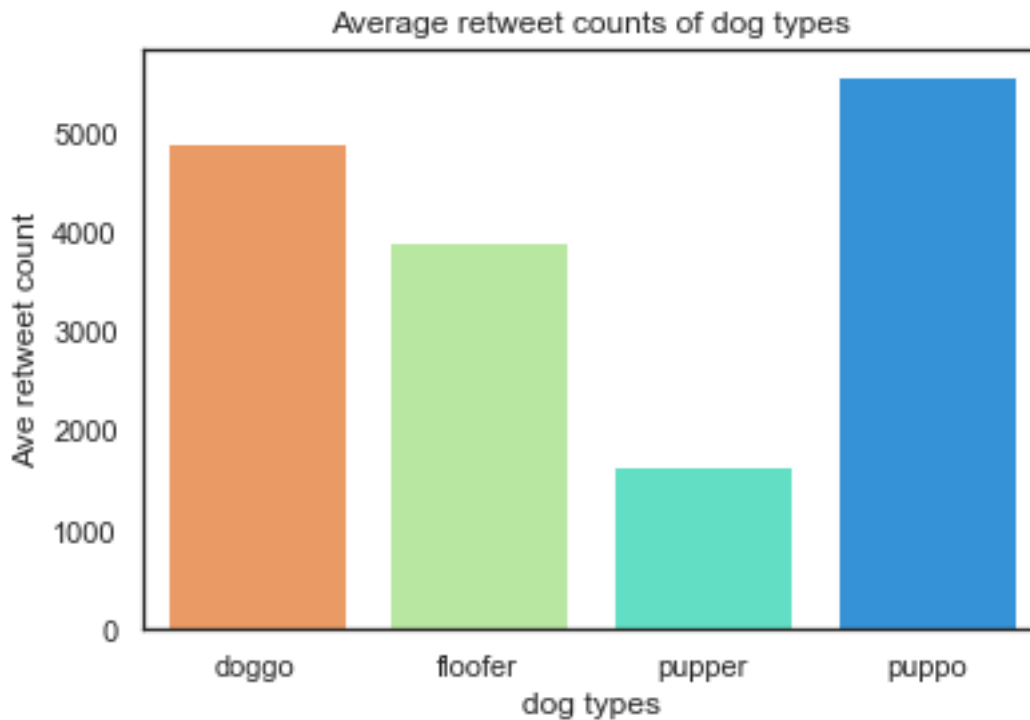
- Mean retweet counts of dog types

```
[13]: df.groupby('types')['retweet_count'].mean()
```

```
[13]: types
doggo      4914.918033
floofer    3906.285714
pupper     1639.165605
puppo      5577.842105
Name: retweet_count, dtype: float64
```

```
[14]: x = df.groupby('types')['retweet_count'].mean()
      sns.barplot(x.index, x.values, palette='rainbow_r').set(title='Average retweet_
      ↪counts of dog types', xlabel='dog types', ylabel='Ave retweet count')
```

```
[14]: [Text(0.5, 1.0, 'Average retweet counts of dog types'),
      Text(0.5, 0, 'dog types'),
      Text(0, 0.5, 'Ave retweet count')]
```



- Mean retweet counts of dog breeds

```
[15]: df.groupby('dog_breed')['retweet_count'].mean().sort_values(ascending=False).
      ↪ head(20)
```

```
[15]: dog_breed
Bedlington_terrier      6562.333333
Afghan_hound            4697.333333
standard_poodle        4400.090909
English_springer        4318.100000
whippet                 4258.000000
Saluki                  4050.500000
flat-coated_retriever   3637.500000
Samoyed                 3527.250000
Bouvier_des_Flandres    3522.000000
Mexican_hairless        3355.571429
Tibetan_mastiff         3290.000000
Lakeland_terrier        3269.555556
black-and-tan_coonhound 3251.500000
Great_Pyrenees          3193.357143
```

```

cocker_spaniel          3170.344828
Leonberg               3025.000000
Cardigan               2971.000000
Labrador_retriever     2854.509804
golden_retriever       2805.900000
Irish_water_spaniel    2672.000000
Name: retweet_count, dtype: float64

```

```

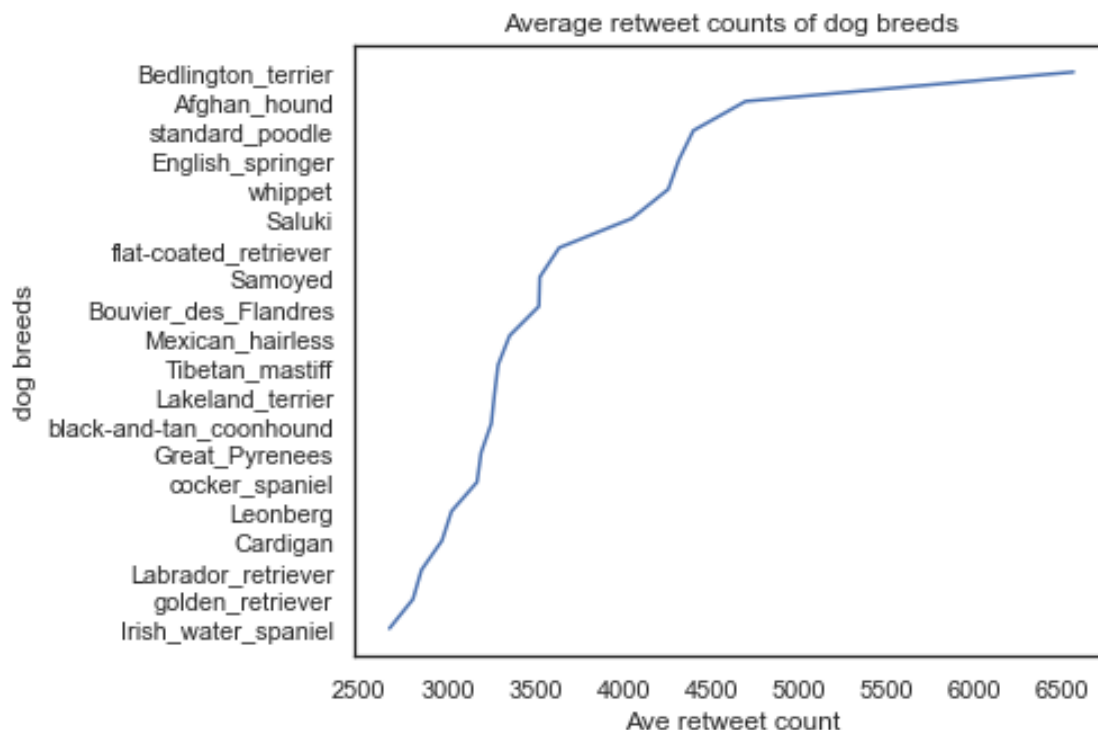
[16]: plt.figure(figsize=(6, 5))
      x = df.groupby('dog_breed')['retweet_count'].mean().
      ↪sort_values(ascending=False).head(20)
      sns.lineplot(x.values, x.index, palette='rainbow_r').set(title='Average retweet_
      ↪counts of dog breeds', ylabel='dog breeds', xlabel='Ave retweet count')

```

```

[16]: [Text(0.5, 1.0, 'Average retweet counts of dog breeds'),
      Text(0, 0.5, 'dog breeds'),
      Text(0.5, 0, 'Ave retweet count')]

```



- The average count of the most liked dog types

```

[17]: df.groupby('types')['favorite_count'].mean()

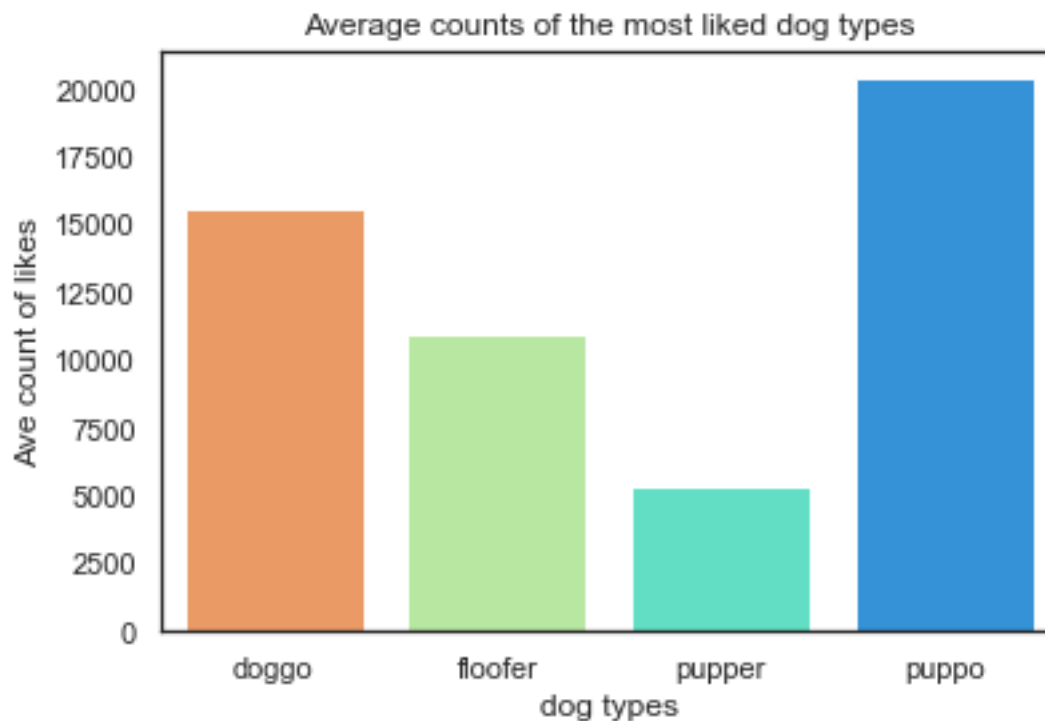
```



```
[17]: types
      doggo      15632.688525
      floofer   10933.714286
      pupper    5354.012739
      puppo     20398.263158
      Name: favorite_count, dtype: float64
```

```
[18]: x = df.groupby('types')['favorite_count'].mean()
      sns.barplot(x.index, x.values, palette='rainbow_r').set(title='Average counts_
      ↪of the most liked dog types', xlabel='dog types', ylabel='Ave count of_
      ↪likes')
```

```
[18]: [Text(0.5, 1.0, 'Average counts of the most liked dog types'),
      Text(0.5, 0, 'dog types'),
      Text(0, 0.5, 'Ave count of likes')]
```



- The average count of the most liked dog breeds

```
[19]: df.groupby('dog_breed')['favorite_count'].mean().sort_values(ascending=False).
      ↪head(20)
```

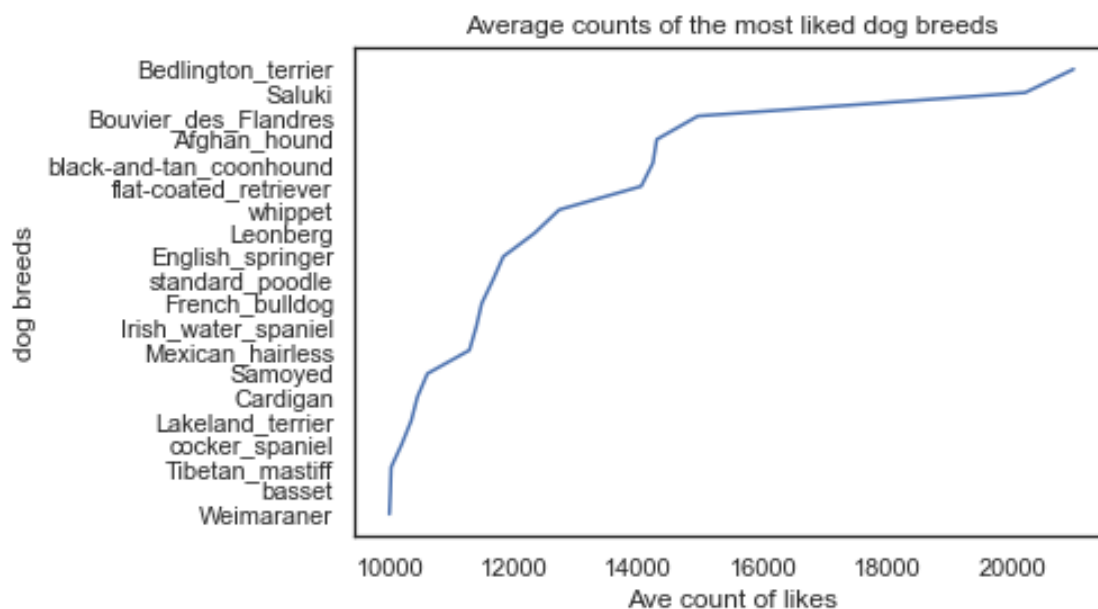
```
[19]: dog_breed
      Bedlington_terrier    20995.666667
      Saluki                20216.500000
```

Bouvier_des_Flandres	14951.000000
Afghan_hound	14289.000000
black-and-tan_coonhound	14228.500000
flat-coated_retriever	14039.375000
whippet	12720.900000
Leonberg	12324.333333
English_springer	11820.700000
standard_poodle	11654.727273
French_bulldog	11473.826087
Irish_water_spaniel	11386.000000
Mexican_hairless	11275.714286
Samoyed	10599.975000
Cardigan	10438.736842
Lakeland_terrier	10341.277778
cocker_spaniel	10189.275862
Tibetan_mastiff	10016.500000
basset	10004.562500
Weimaraner	9988.500000

Name: favorite_count, dtype: float64

```
[20]: x = df.groupby('dog_breed')['favorite_count'].mean().
      ↪sort_values(ascending=False).head(20)
      sns.lineplot(x.values, x.index, palette='rainbow_r').set(title='Average counts_
      ↪of the most liked dog breeds', ylabel='dog breeds', xlabel='Ave count of_
      ↪likes')
```

```
[20]: [Text(0.5, 1.0, 'Average counts of the most liked dog breeds'),
      Text(0, 0.5, 'dog breeds'),
      Text(0.5, 0, 'Ave count of likes')]
```



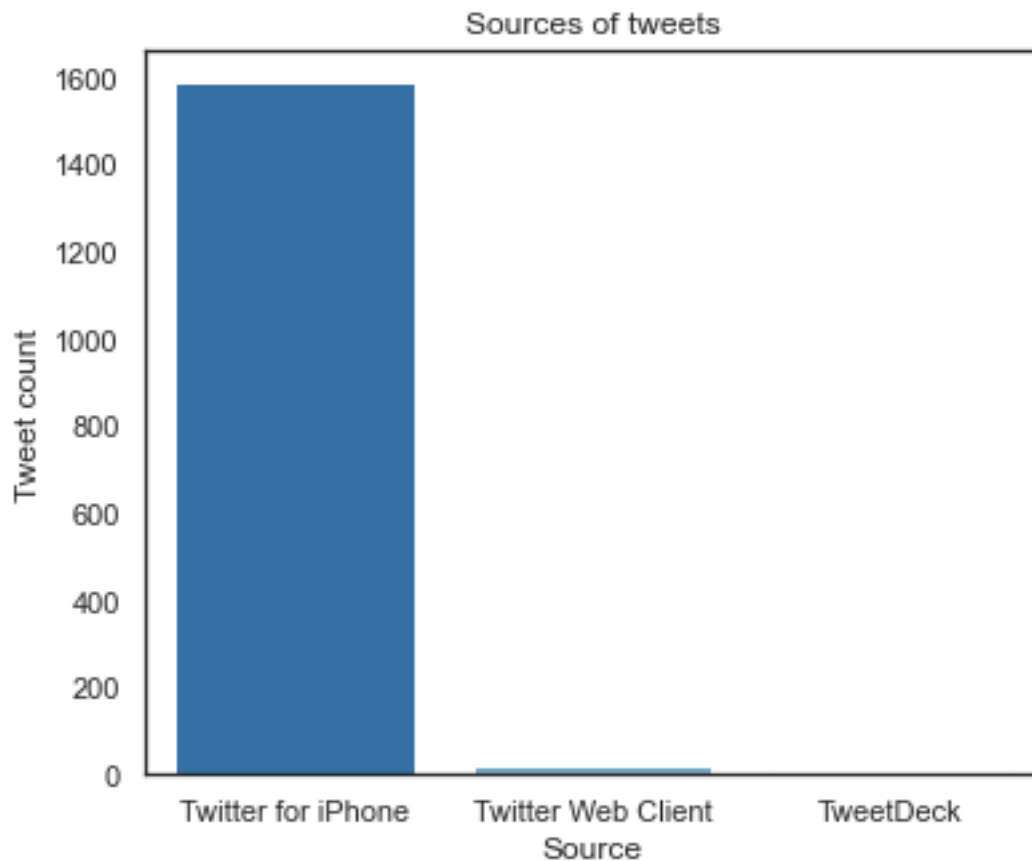
- The most popular source of tweets

```
[21]: df.source.value_counts()
```

```
[21]: Twitter for iPhone    1584
      Twitter Web Client    23
      TweetDeck            8
      Name: source, dtype: int64
```

```
[22]: plt.figure(figsize=(6, 5))
      x = df.source.value_counts()
      sns.barplot(x.index, x.values, palette='Blues_r').set(title='Sources of
      ↪tweets', xlabel='Source', ylabel='Tweet count')
```

```
[22]: [Text(0.5, 1.0, 'Sources of tweets'),
      Text(0.5, 0, 'Source'),
      Text(0, 0.5, 'Tweet count')]
```



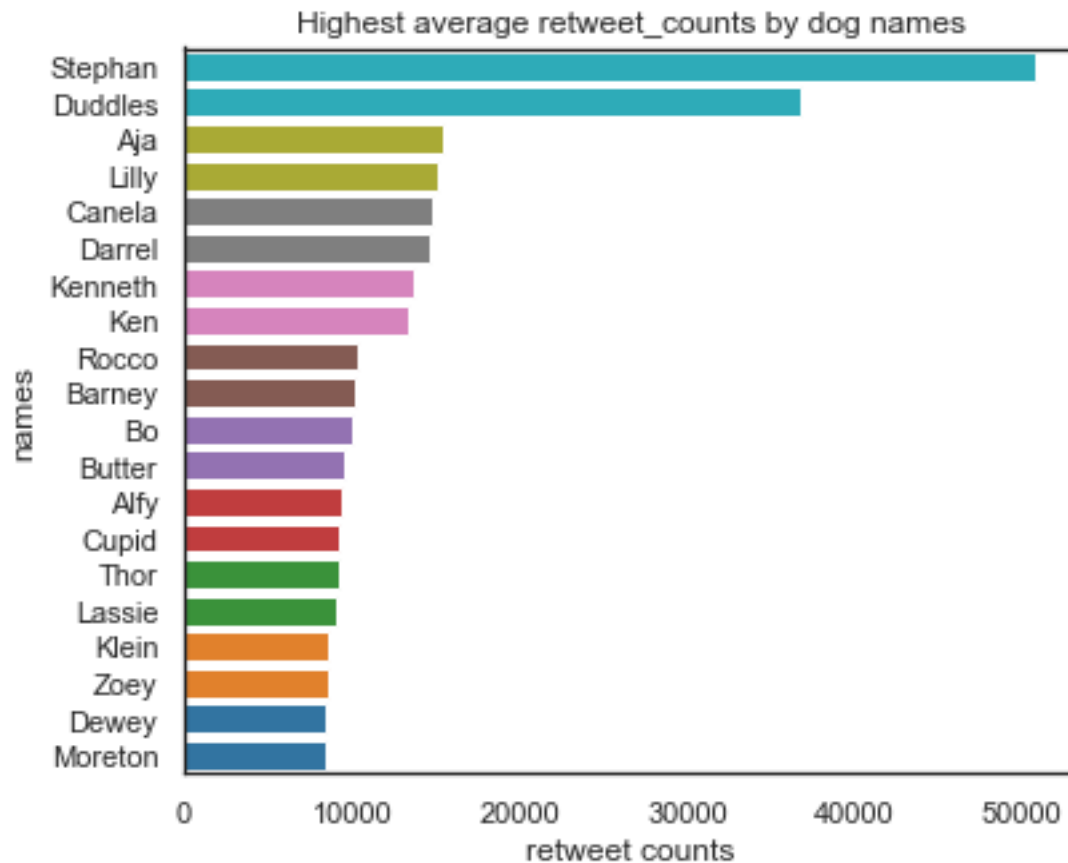
- The highest average retweet counts by dog names

```
[23]: df.groupby('names')['retweet_count'].mean().sort_values(ascending=False).
      ↪head(20)
```

```
[23]: names
Stephan      50687.00
Duddles      36714.00
Aja          15516.00
Lilly        15158.00
Canela       14759.00
Darrel        14632.00
Kenneth       13661.00
Ken           13347.00
Rocco         10382.00
Barney        10202.00
Bo            10094.25
Butter        9492.00
Alfy          9414.00
Cupid         9177.00
Thor          9175.00
Lassie        9085.00
Klein         8658.00
Zoey          8594.00
Dewey         8494.00
Moreton       8441.00
Name: retweet_count, dtype: float64
```

```
[24]: plt.figure(figsize=(6,5))
x = df.groupby('names')['retweet_count'].mean().sort_values(ascending=False).
      ↪head(20)
sns.barplot(x.values, x.index, palette='tab10_r').set(title='Highest average_
      ↪retweet_counts by dog names', xlabel='retweet counts', ylabel='names')
```

```
[24]: [Text(0.5, 1.0, 'Highest average retweet_counts by dog names'),
      Text(0.5, 0, 'retweet counts'),
      Text(0, 0.5, 'names')]
```



- Word Cloud of the tweets

```
[25]: plt.figure(figsize=(10,10))
from wordcloud import WordCloud
text = " ".join(cat for cat in df.text)
wordcloud = WordCloud(collocations=False, background_color='white').
    generate(text)

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```