This project was broken down into the following parts:

1. Gather and import data
2. Inspect data
3. Remediate issues in tidiness and quality.
4. Create a master dataframe
5. Evaluate the dataframe for insights

Gather and import data

The data sources were the following:

- WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)
- tweet image prediction (image_predictions.tsv)
- Tweets Data

Notes: each one of these data sources required a different method to import, WeRateDogs was a straight CSV to dataframe import of a CSV file, tweet image prediction required the requests method to access a link to import a tab delimited file into a dataframe, while the tweets data was ideally gathered using the Twitter API (due to issues with my Twitter developer account I elected to use the provided file) once the Twitter data was gathered then a Json file was created which was converted to a dataframe. These requirements demonstrated 3 different methods of accessing data (4 if you count building the Json file)

Inspect data

I programmatically inspected data using the head() and info() methods in Python, but also exported the dataframes to CSV files so I could inspect them in Excel. Below were my findings:

1. Tidiness issues¶
    2. For twitter-archive: Dog category values are spread across the doggo, floofer, pupper, and puppo columns

    1. For twitter-archive: Retweet/In Reply rows could potentially refer to already existing tweet rows thus raising the prospect of multiple rows with same original tweet ID

3. Quality issues

    1. For twitter-archive numerator ratings: some values lie outside the 1-10 range thus skewing the data
    2. For twitter-archive denominator ratings: some values lie outside the 1-10 range thus skewing the data. in one case the value is zero thus making any calculated value using this data problematic.
    3. For twitter-archive: In some cases, there may be multiple values for a row in doggo, floofer, pupper, and puppo columns, this will be determined once the information in those 4 columns is transformed from wide to tall.

4. For twitter-archive: In isolation the numerator and denominator columns lack value, a new calculated column needs to created that provides a rating value, making it easier to perform analysis.
5. For image predications, some images in the p1 column (highest confidence) have a false value.
6. For image predications: p2, p2_conf, p2_dog, p3, p3_conf, p3_dogare unnecessary as we should default to the value with the highest confidence/
7. For Tweets: some values in tweet_id are not integers (this will be the value used to join dataframes
8. Some column headings are unclear (Twitter Archive: source, image predictions: p1 and will be renamed in the interest of clarity.

Notes: of these issues I found the Dog category values are spread across the doggo, floofer, pupper, and puppo columns, and for the Tweets dataframe that some values in tweet_id are not integers (this will be the value used to join dataframes, the most troubling, the wide presentation of the dog categories causing me to ask why anyone would do that and the absence of all integer in the tweet_id column I wondered how that could happen (was the breakdown a function of the method the data was gathered?)

Remediate issues in tidiness and quality

Fairly straight forward, the only method to mention was using the Pandas melt method to go from wide to tall with the dog categories data.
Notes: I don't see the point of the p2, p2_conf, p2_dog, p3, p3_conf, p3_dog columns in the image predictions file as it seems obvious(to me) that p1&p1_conf was all that was required.

Create a master dataframe

Straight forward, tweet_ID was used to join the 3 dataframes, anyone with experience in SQL would have immediately grasped the concept.

Evaluate the dataframe for insights

I used the pyplot library to visualize three insights:

4. Top 5 most Common Dog Breeds (value_counts() based)
5. Highest rated category (pivot_table() based)
6. Most Common Dog types (value_counts() based)
7.

Notes: I chose the Highest rated category in a effort to understand how pivoting dataframes worked in Pandas