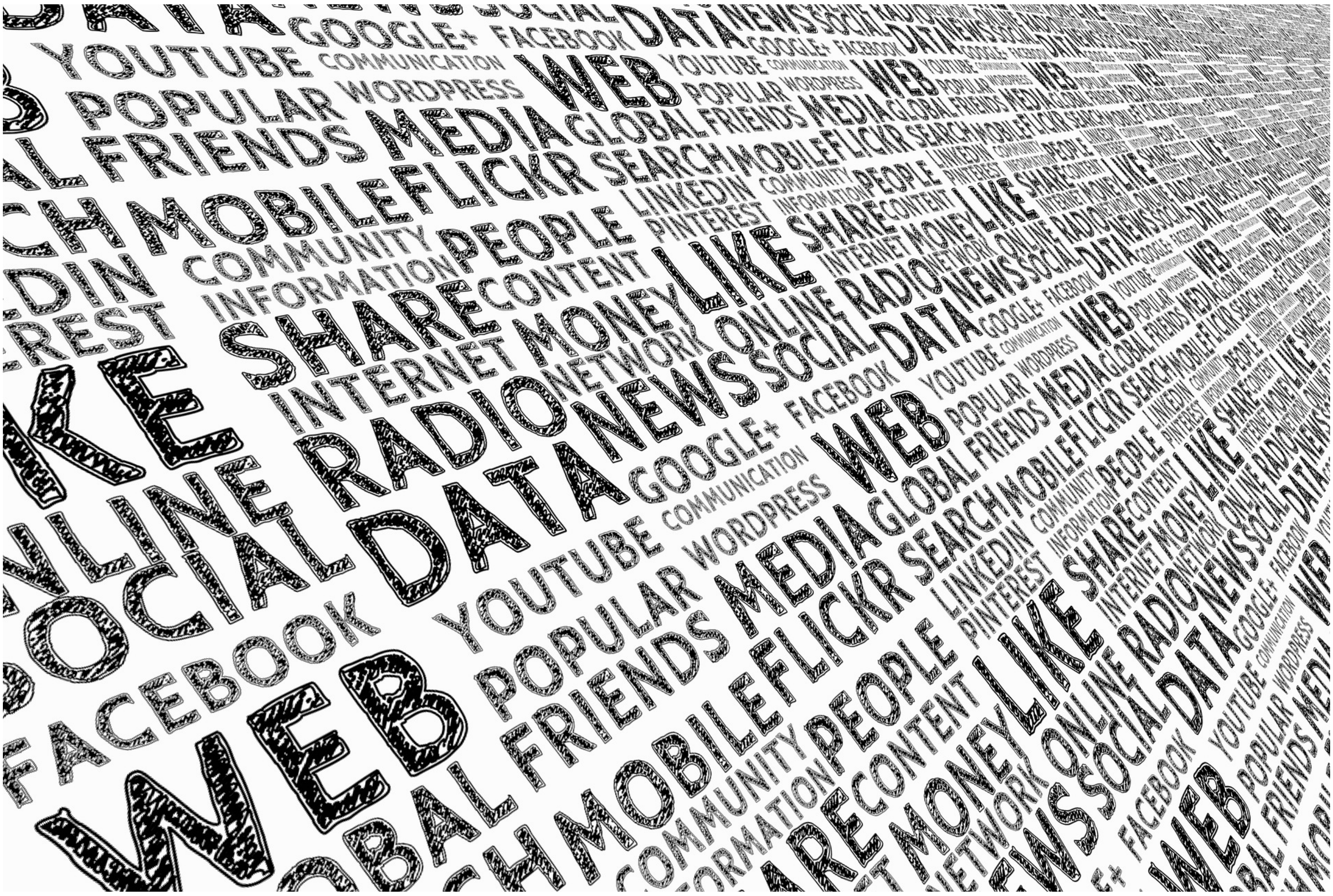


# UDACITY Data Analysis Nanodegree

## Project:- Wrangle & Analyze Data, Report

**Grant Patience, 28th August 2019**

# Table of Contents

# Introduction

This project focused on wrangling data from the **WeRateDogs Twitter (https://twitter.com/dog_rates)** account using Python, documented in a Jupyter Notebook (wrangle_act.ipynb). This Twitter account rates dogs with humorous commentary. The rating denominator is usually 10, however, the numerators are usually greater than 10. They're Good Dogs Brent wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for us to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017

---

# 1. Determine Objectives and Assess the Situation

For this project we will use the **CRISP-DM process (https://www.sv-europe.com/crisp-dm-methodology/)**. The first stage of the CRISP-DM process is to understand what you want to accomplish. The goal of this stage of the process is to uncover important factors that could influence the outcome of the project.

## Project Details

Fully assessing and cleaning the entire dataset would require exceptional effort so only a subset of its issues (eight quality issues and two tidiness issues at minimum) needed to be assessed and cleaned.

The tasks for this project were:

- Data wrangling, which consists of:
  - Gathering data
  - Assessing data
  - Cleaning data
- Storing, analyzing, and visualizing our wrangled data
- Reporting on 1) our data wrangling efforts and 2) our data analyses and visualizations

## Key Points

Key points to keep in mind when data wrangling for this project:

- We only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- Fully assessing and cleaning the entire dataset requires exceptional effort so only a subset of its issues (eight (8) quality issues and two (2) tidiness issues at minimum) need to be assessed and cleaned.
- Cleaning includes merging individual pieces of data according to the rules of tidy data.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.
- We do not need to gather the tweets beyond August 1st, 2017. We can, but note that we won't be able to gather the image predictions for these tweets since we don't have access to the algorithm used.

## 1.1 Outline of Steps

- We state what resources are available to us and in this section we discuss what it is we wish to achieve,
- We decide which Questions we want to ask of the data
- We will Gather the Data that we need
- Import the data into Python to perform some initial Understanding of the data to help us understand the data, and Assess Data Quality and perform any resolve any Data Cleansing.
- Perform Exploratory Data Analysis where we will research the answers to our questions
- Create visualisations to aid exploration and research
- Draw our Conclusion based on the data and communicate our findings

## 1.2 What are the desired outputs of the project?

- Accurate project submission:

> - Ensure you meet specifications for all items in the Project Rubric. Your project "meets specifications" only if it meets specifications for all of the criteria.
> - Ensure you have not included your API keys, secrets, and tokens in your project files.
> - If you completed your project in the Project Workspace, ensure the following files are present in your workspace, then click "Submit Project" in the bottom righthand corner of the Project Workspace page:
> - wrangle_act.ipynb: code for gathering, assessing, cleaning, analyzing, and visualizing data
> - wrangle_report.pdf or wrangle_report.html: documentation for data wrangling steps: gather, assess, and clean
> - act_report.pdf or act_report.html: documentation of analysis and insights into final data
> - twitter_archive_enhanced.csv: file as given
> - image_predictions.tsv: file downloaded programmatically
> - tweet_json.txt: file constructed via API
> - twitter_archive_master.csv: combined and cleaned data
> - any additional files (e.g. files for additional pieces of gathered data or a database file for your stored clean data)

- Meet the Criteria of the Udacity Rubric:

> **Code Functionality**
> - The student’s code is functional. All project code is contained in a Jupyter Notebook named wrangle_act.ipynb and runs without errors.
> - The student’s code is readable, i.e., uses good coding practices. The Jupyter Notebook has an intuitive, easy-to-follow logical structure. The code uses comments effectively and is interspersed with Jupyter Notebook Markdown cells. The steps of the data wrangling process (i.e. gather, assess, and clean) are clearly identified with comments or Markdown cells, as well.

> **Gathering Data**
> - The student is able to gather data from a variety of sources and file formats. Data is successfully gathered: -- From at least the three (3) different sources on the Project Details page. -- In at least the three (3) different file formats on the Project Details page. -- Each piece of data is imported into a separate pandas DataFrame at first.

**Assessing Data**

- The student is able to assess data visually and programmatically for quality and tidiness. Two types of assessment are used:

  > - Visual assessment: each piece of gathered data is displayed in the Jupyter Notebook for visual assessment purposes. Once displayed, data can additionally be assessed in an external application (e.g. Excel, text editor).
  > - Programmatic assessment: pandas' functions and/or methods are used to assess the data.

- The student is able to thoroughly assess a dataset. At least eight (8) data quality issues and two (2) tidiness issues are detected, and include the issues to clean to satisfy the Project Motivation. Each issue is documented in one to a few sentences each.

**Cleaning Data**

- The student uses the steps in the data cleaning process to guide their cleaning efforts. The define, code, and test steps of the cleaning process are clearly documented.
- The student is able to thoroughly clean a dataset programmatically. Copies of the original pieces of data are made prior to cleaning. All issues identified in the assess phase are successfully cleaned (if possible) using Python and pandas, and include the cleaning tasks required to satisfy the Project Motivation. A tidy master dataset (or datasets, if appropriate) with all pieces of gathered data is created.

**Storing and Acting on Wrangled Data**

- The student is able to store a gathered, assessed, and cleaned dataset. Students will save their gathered, assessed, and cleaned master dataset(s) to a CSV file or a SQLite database.
- The student is able to act on their wrangled data to produce insights (e.g. analyses, visualizations, and/or models). The master dataset is analyzed using pandas or SQL in the Jupyter Notebook and at least three (3) separate insights are produced. At least one (1) labeled visualization is produced in the Jupyter Notebook using Python’s plotting libraries or in Tableau. Students must make it clear in their wrangling work that they assessed and cleaned (if necessary) the data upon which the analyses and visualizations are based.

**Report**

- The student is able to reflect upon and describe their data wrangling efforts. The student’s wrangling efforts are briefly described. This document (wrangle_report.pdf or wrangle_report.html) is concise and approximately 300-600 words in length.
- The student is able to describe some insights found in their wrangled dataset. The three (3) or more insights the student found are communicated. At least one (1) visualization is included. This document (act_report.pdf or act_report.html) is at least 250 words in length.

> **Project Files**
> - Are all required files included in the student's submission?
> - The following files (with identical filenames) are included:
>
> > - wrangle_act.ipynb
> > - wrangle_report.pdf or wrangle_report.html
> > - act_report.pdf or act_report.html
> > - All dataset files are included, including the stored master dataset(s), with filenames and extensions as specified on the Project Submission page.

## 1.3 What Resources are Available?

- UDACITY Rubric (https://review.udacity.com/#!/rubrics/1136/view) for guidance on project submission
- Dataset supplied and gathered (Details in Section Data Description )
- Twitter API on Twitter's Developer Portal (https://developer.twitter.com/en/docs/basics/developer-portal/overview)
- Jupyter Python Notebook

## 1.4 What Questions Are We Trying To Answer?

> - **Q1. What Correlations can we find in the data? e.g. Favourite / Retweet**
> - **Q2. Which are the more popular; doggos, puppers, fullfers or poppos?**
> - **Q3. Which are the more popular dog breeds**

# 3. Exploratory Data Analysis

In [2]:

```
# Import necessary libraries for initial data understanding, visualisations and exploratory
import numpy as np
import pandas as pd
import requests
import tweepy
import json
import time
import re

#For Visuals
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('darkgrid')
```

In [5]:

```
# reads the data from the file - denotes as CSV, it has no header row, sets column headers
df_twitter = pd.read_csv('./Data/twitter_archive_master.csv')
```

Now let's take our first look at the data

In [7]:

```
df_twitter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1978 entries, 0 to 1977
Data columns (total 27 columns):
tweet_id                 1978 non-null int64
in_reply_to_status_id    21 non-null float64
in_reply_to_user_id      21 non-null float64
timestamp                1978 non-null object
source                   1978 non-null object
text                     1978 non-null object
expanded_urls            1978 non-null object
rating_numerator         1978 non-null int64
rating_denominator       1978 non-null int64
name                     1342 non-null object
dog_class                305 non-null object
favorites                1978 non-null int64
retweets                 1978 non-null int64
user_followers           0 non-null float64
user_favourites          0 non-null float64
date_time                1978 non-null object
jpg_url                  1978 non-null object
img_num                  1978 non-null int64
Breed_Probability1       1978 non-null object
Breed_Confidence1        1978 non-null float64
Dog_Flag_1               1978 non-null bool
Breed_Probability2       1978 non-null object
Breed_Confidence2        1978 non-null float64
Dog_Flag_2               1978 non-null bool
Breed_Probability3       1978 non-null object
Breed_Confidence3        1978 non-null float64
Dog_Flag_3               1978 non-null bool
dtypes: bool(3), float64(7), int64(6), object(11)
memory usage: 376.7+ KB
```

In [8]:

```
df_twitter.head(3)
```

Out[8]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | source | |
|---|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56+00:00 | Twitter for iPhone | T<br>Phi<br>H<br>my<br>boy. |
| **1** | 892177421306343426 | NaN | NaN | 2017-08-01 00:17:27+00:00 | Twitter for iPhone | T<br>Tilly.<br>che<br>pı<br>y |
| **2** | 891815181378084864 | NaN | NaN | 2017-07-31 00:18:03+00:00 | Twitter for iPhone | T<br>Archi<br>is a<br>Norw<br>Pour |

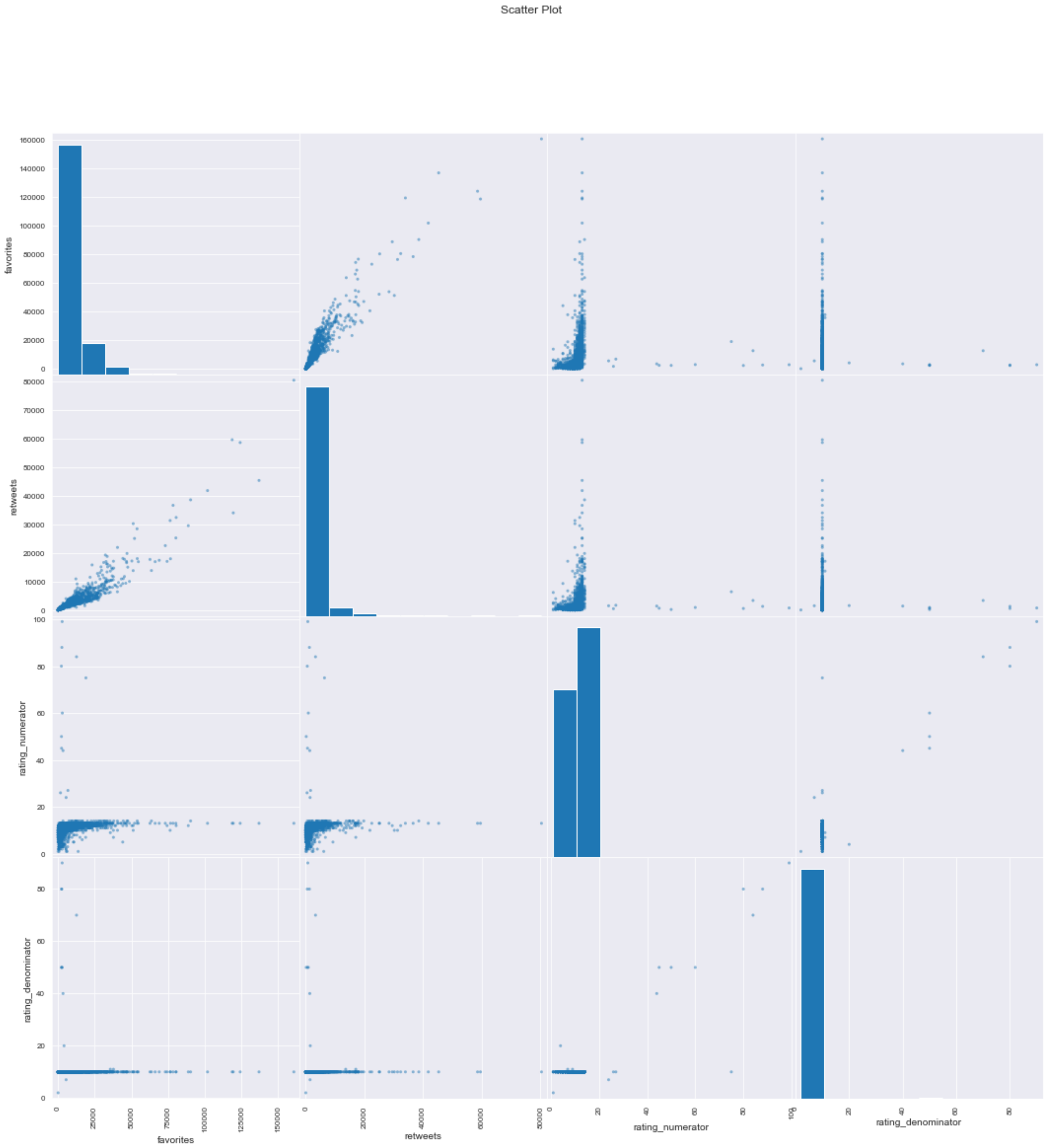
3 rows × 27 columns

## Q1. What Correlations can we find in the data? e.g. Favourite / Retweet

First, let's do some simple correlation charts - can we find any interesting correlations? I suspect Favourite & Retweet will be correlated since these are both ways to show your appreciation for a tweet on Twitter.

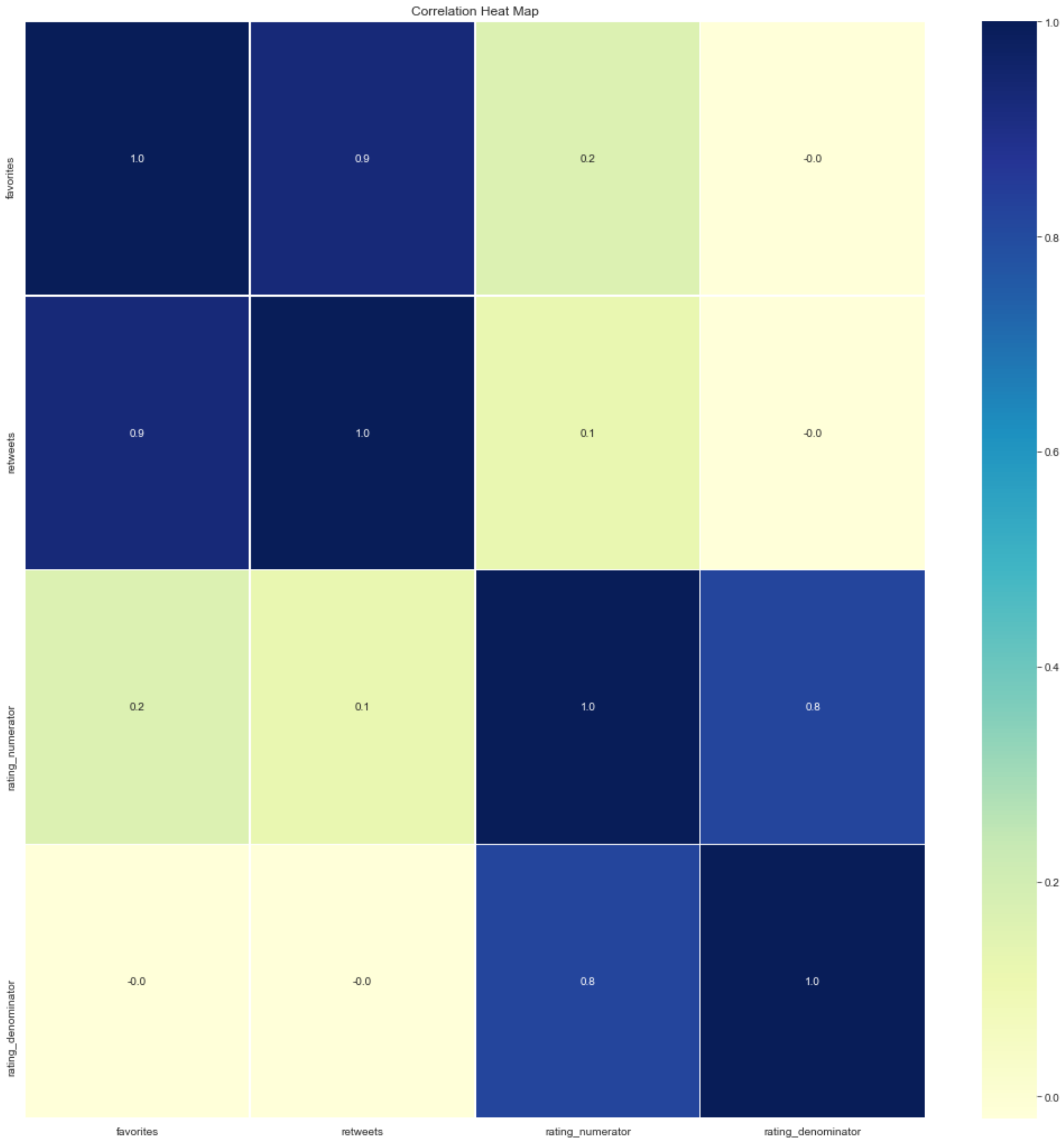
In [9]:

```
pd.plotting.scatter_matrix(df_twitter[['favorites', 'retweets', 'rating_numerator', 'rating

plt.suptitle('Scatter Plot')
plt.show()
```



In [10]:

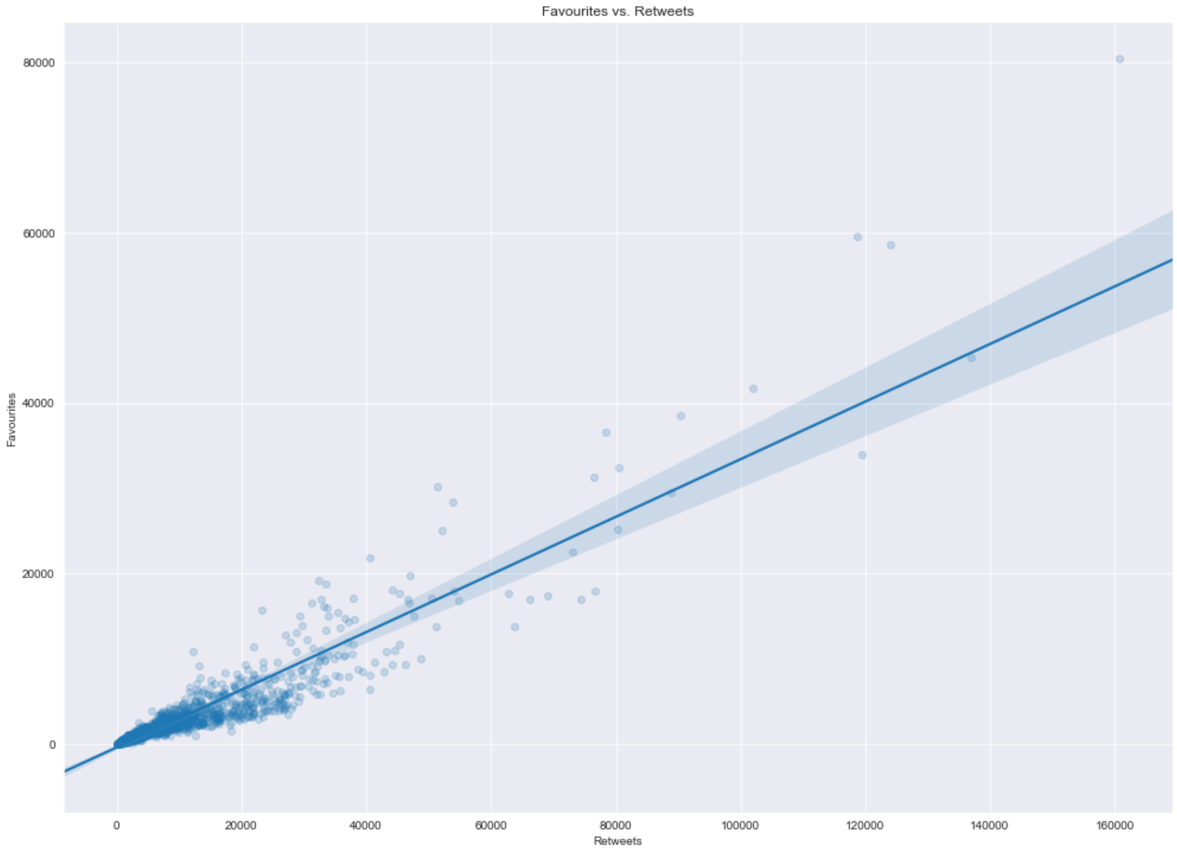
```
f, ax= plt.subplots(figsize=(18,18))
sns.heatmap(df_twitter[['favorites', 'retweets', 'rating_numerator', 'rating_denominator',

plt.title('Correlation Heat Map')
plt.show()
```



```
In [11]:
```

```
# Plot scatterplot of retweet vs favorite count
sns.lmplot(x="favorites",
           y="retweets",
           data=df_twitter,
           size = 10,
           aspect=1.4,
           scatter_kws={'alpha':1/5})
plt.title('Favourites vs. Retweets')
plt.xlabel('Retweets')
plt.ylabel('Favourites');
```

```
D:\Program_Files\lib\site-packages\seaborn\regression.py:546: UserWarning: T
he `size` paramter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



- As we assumed, There is a strong linear relationship between Favourites and Retweets.
- The regression coefficient for this relationship is (r= 0.797)
- From the points we plotted, we cannot find any other correlations.
- In future, we could try and categorize the source and dog_stage to investigate correlations there with popularity of the Tweet.

## Q2. Which are the more popular; doggos, puppers, fullfers or poppos?

We performed some data wrangling on the tweet_archive dataset to integrate 4 different "Class" of doggos down into one column which would be easier to use (dog_class = (doggo, pupper, fluffer, puppo))

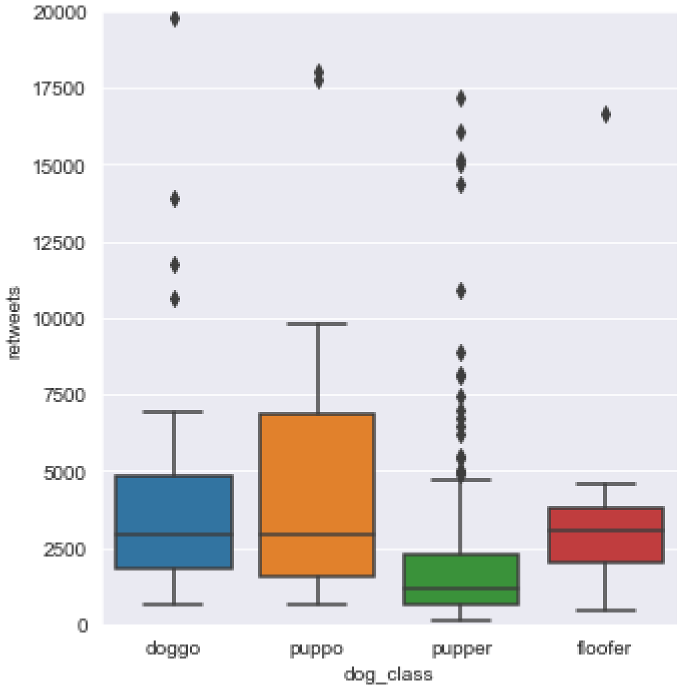
Using this newly cleaned data, this makes it much easier to analyse and visualise the contents, Lets use this to our advantage to determine the top dog breeds If we use the dog_class column, can we ascertain which category of dog is more popular?

In [12]:

```
ax = sns.catplot(x="dog_class",y="retweets",kind='box',data=df_twitter)
ax.set(ylim=(0, 20000))
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x21bb56fb6d8>
```

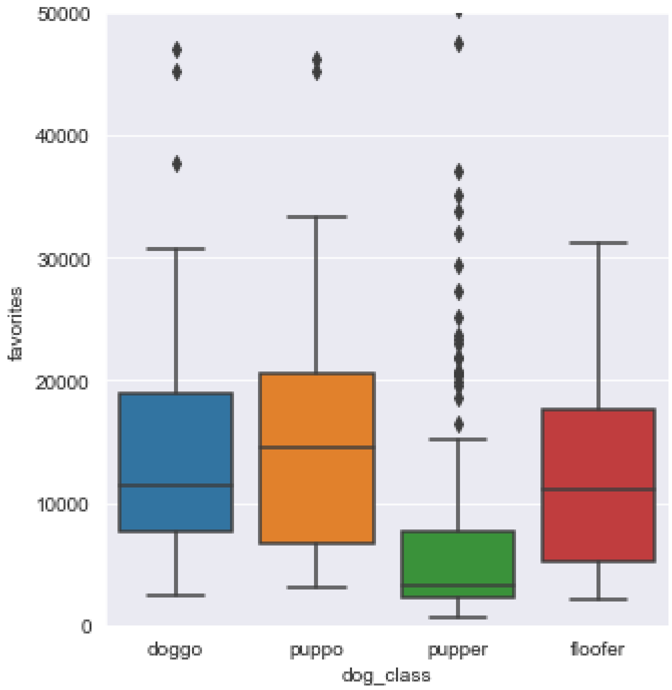


In [13]:

```
ax = sns.catplot(x="dog_class",y="favorites",kind='box',data=df_twitter)
ax.set(ylim=(0, 50000))
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x21bb58e4128>
```



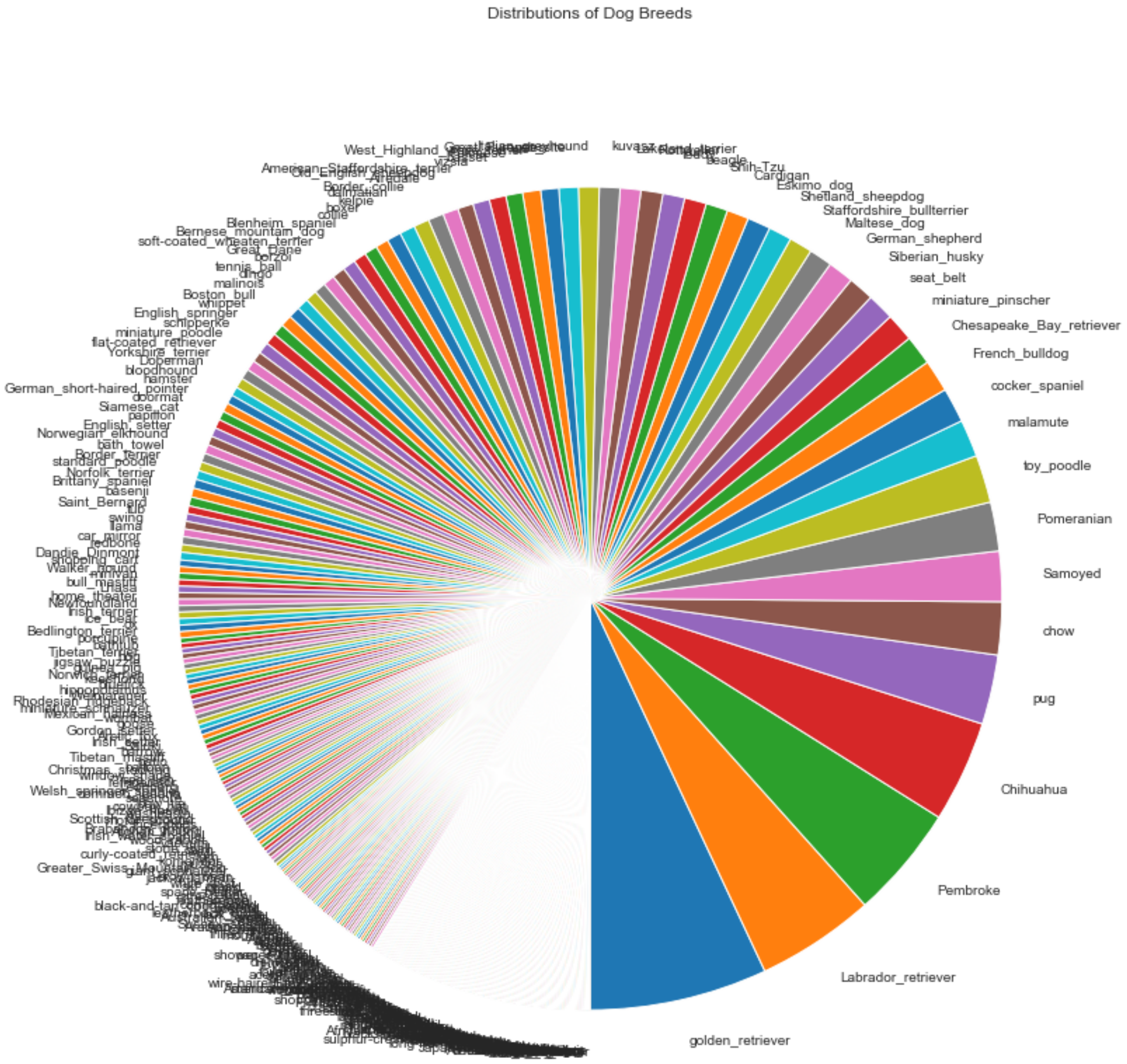
- Interesting, as we look at Retweets and Favourites, Puppos are by far the more popular
- From the points we plotted, we can see that Puppers have a lot of outliers.

## Q3. Which are the more popular dog breeds?

By integrating the image_prediction data into our dataset, we have three columns denoting the probability chance of the image being of a particular breed. This is some really interesting data to use, lets use it to see if we can determine the popularity of certain breeds of dogs

In [14]:

```
# Piechart
df = df_twitter[pd.notnull(df_twitter.Breed_Probability1)]
plt.rcParams['figure.figsize']=(15,15)

import matplotlib.pyplot as plt
df.Breed_Probability1.value_counts(sort=True).plot.pie(startangle=270, pctdistance=0.8, rac
plt.title('Distributions of Dog Breeds')
plt.ylabel('')
plt.show();
```



- We can see the most common types of dog here are Golden Retrievers and Labrator Retrievers, this seems sensible since these dog types are very common. Other dog breeds rounding out the top 5 are Chihuahuas, Pugs and Pembrokes.
- In future, we could try and narrow the dataset to only the twop 10 dog breeds to declutter the visual
- We could also limit the probability to ensure it meets a minimum probability level
- Some incorrect values like Seat-Belt, hamster, bath towel still exist in the data which we could clean given more time in future
- We only used the 1st prediction column, we may have been able to use all 3 to determine the overall probability or popularity of dog breeds

# 4. Observations and Conclusion

- During our analysis, we ound that there is a strong linear relationship between the number of Favourites and the number of Retweets of a given Tweet. The regression coefficient for this relationship is (r= 0.797)
- We did anticipate this relationship already since there is a fair chance that if a user enjoys a twee they have the choice option to Favourite or Retweet it - both are a measure of the users enjoyment of the tweet.
- We have also found through visualisation and data wrnagling that the pupper "dog class" is the most popular, with on average, more Retweets and more Favourites per tweet than the other 3 categories Doggo, Fluffer and Puppo.
- Golden retriever, Labrador Retriever, Pembroke, Chihuahua and Pugs are the top 5 common dog breeds in the data. After integrating the image_predictions data, we can see that incorrect values like Seat-Belt still exist in the data which need to be removed.

## Limitations and Assumptions

- In future, we could try and categorize the source and dog_stage to investigate correlations there with popularity of the Tweet.
- There are some outlier numbers in the denominator and numerator columns, we could have cleaned these with hindsight
- There are too many different dog types to represent easily in a visualisation. Given more time, we could trim the types down to the top 10, or we could also limit the probability to ensure it meets a minimum probability level
- Some incorrect values remain in the image predition calssifications, like Seat-Belt, hamster, bath towel still exist in the data which we could clean given more time in future
- We only used the 1st prediction column, we may have been able to use all 3 to determine the overall probability or popularity of dog breeds

---

# References

- Title Image (https://pixabay.com/illustrations/social-media-media-board-networking-1989152/)
- Seaborn Line Plot (https://seaborn.pydata.org/generated/seaborn.lineplot.html)
- seaborn Boxplot (https://towardsdatascience.com/data-visualization-using-seaborn-fc24db95a850)

In [ ]: