数据收集：

1、WeRateDogs 推特档案。**twitter-archive-enhanced.csv 文件从 github 下载。**

2、推特图片预测，即根据神经网络，出现在每个推特中狗的品种（或其他物体、动物等）。**通过 Requests 库下载，然后保存成 image-predictions.tsv 文件。**

3、每条推特的数据，至少要包含转发数（retweet_count）和喜欢数（favorite_count），以及任何你觉得有趣的额外数据。**由于众所周知的原因，tweet_json.txt 文件从 github 下载。创建一个空列表，tweet，并附加字典。这个字典列表最终将被转换为 pandas DataFrame(这是 逐行构建 DataFrame 的最有效方式)。**

数据评估

质量问题:

在 WeRateDogs 推特档案里：

1、in_reply_to_status_id、in_reply_to_user_id、retweeted_status_id、retweeted_status_user_id、retweeted_status_timestamp 列的值大部分为空。

2、expanded_urls 的值不需要。

3、狗的“地位”数据大部分缺失。

4、有些用户的评分分子评级小于分母评级。

5、rating_denominator 的值不为 10。

6、name 数据部分缺失。

7、部分 rating_numerator 的数值过大。

在 image-predictions.tsv 里

1、有用户上传的不是狗的图片

整洁度问题：

1、相关 ID 的狗的种类应该放入 **twitter-archive-enhanced.csv 中。**

2、相关转发数 (retweet_count) 和喜欢数 (favorite_count) 应该放入 **twitter-archive-enhanced.csv 中。**

数据清洗

1、用 DataFrame.drop 删除 in_reply_to_status_id、in_reply_to_user_id、retweeted_status_user_id、retweeted_status_timestamp 列。

2、根据 retweeted_status_id 只保留值为空的行。

3、drop 掉 expanded_urls 行。

4、只保留分子大于分母的行。

5、统一分母为 10，且分子只取 10~14 的值。

6、drop 非狗列。

7、把识别出的狗的种类的行 merge 到 **twitter-archive-enhanced.csv 中。**

8、把相关转发数（retweet_count）和喜欢数（favorite_count）merge 到 **twitter-archive-enhanced.csv 中。**

9、把清理好的数据保存为 **twitter_archive_master.csv** 文件。

数据可视化

导入 pygal 库

把对狗狗的喜爱程度排序的结果由高到低做成交互式柱形图，点击柱形可跳转至狗狗的图片。

把对上传图片用户中拥有最多或最少数量的狗狗种类的结果做成柱形图。

结果导出为 **popupar_dogs.svg，unpopular_dogs，most_kind_dogs 和 minimal_kind_dogs 文件。**