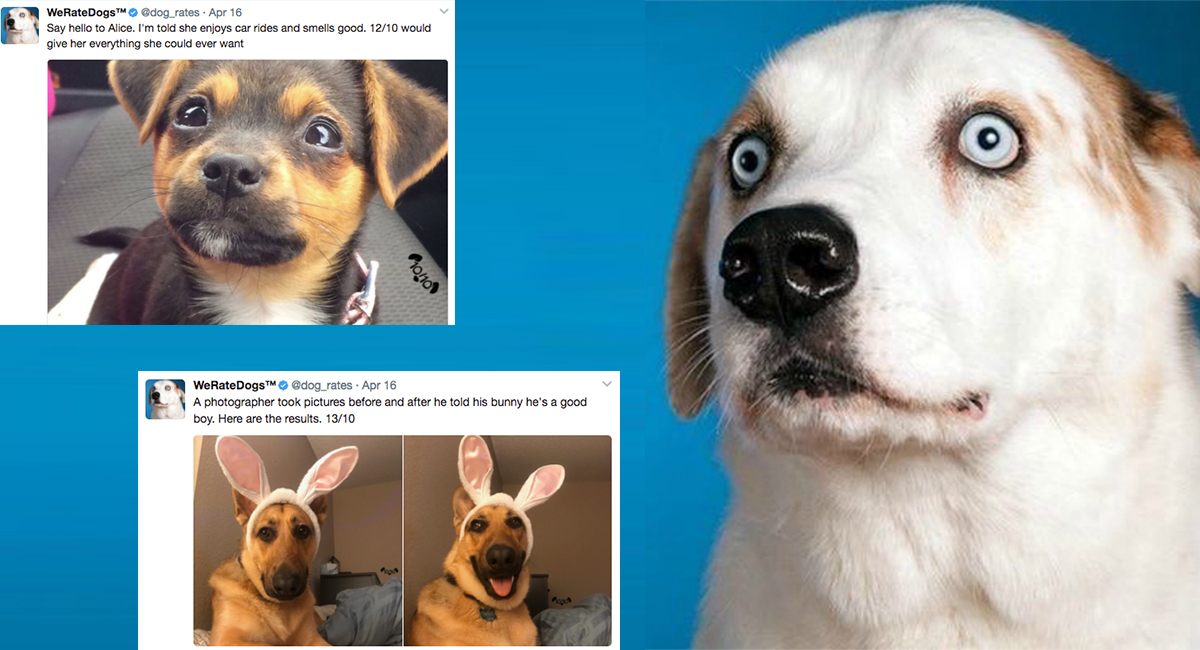
## 简介

现实世界的数据通常都不干净。使用 Python 以及 Python 的库，你可以收集各种来源、各种格式的数据，评估数据的质量和整洁度，然后进行清洗。这个过程叫做数据整理。你可以在 Jupyter Notebook 中记录并展示数据整理的过程，然后使用 Python (及其库) 和/或 SQL 进行分析和可视化。

你将要整理 (以及分析和可视化) 的数据集是推特用户 [**@dog\_rates**](https://twitter.com/dog_rates) 的档案, 推特昵称为 **[WeRateDogs](https://en.wikipedia.org/wiki/WeRateDogs" \t "_blank)**。WeRateDogs 是一个推特主，他以诙谐幽默的方式对人们的宠物狗评分。这些评分通常以 10 作为分母。但是分子则一般大于 10：11/10、12/10、13/10 等等。为什么会有这样的评分？因为 "[**They're good dogs Brent.**](http://knowyourmeme.com/memes/theyre-good-dogs-brent)" WeRateDogs 拥有四百多万关注者，曾受到国际媒体的报道。

WeRateDogs [**下载了他们的推特档案**](https://support.twitter.com/articles/20170160)，并通过电子邮件发送给优达学城，专门为本项目使用。这个档案是基本的推特数据（推特 ID、时间戳、推特文本等），包含了截止到 2017 年 4 月 1 日的 5000 多条推特。



* 图片来自 [**波士顿杂志**](http://www.bostonmagazine.com/arts-entertainment/blog/2017/04/18/dog-rates-mit/)\*

## 我们需要什么软件？

使用这里提供的 Jupyter Notebook workspace，在优达学城课堂的 **项目 workspace** 页面可以完成整个项目。

如果想在优达学城课堂以外操作，你需要满足以下软件要求：

* 你需要在电脑上用 Jupyter Notebook 操作。请再次访问纳米学位课程中的 Anaconda 和 Jupyter Notebook 教程，作为安装指南。
* 需要安装下列软件包（即 Python 库）。你可以通过 conda 或 pip 安装这些库。请再次访问纳米学位课程中的 Anaconda 和 Jupyter Notebook 教程，作为文件包的安装指南。
  + pandas
  + numpy
  + requests
  + tweepy
  + json
* 你需要创建包含图片的书面文档，并且把这些文档导出为 PDF 文件。这个任务可以在 Jupyter Notebook 中进行，但是最好使用文字处理软件，如免费软件 [**Google Docs**](https://www.google.com/docs/about/) 或 Microsoft Word。
* 文本编辑器，如免费软件 [**Sublime**](https://www.sublimetext.com/)，这可能会用到，但不是必须的。

**项目动机**

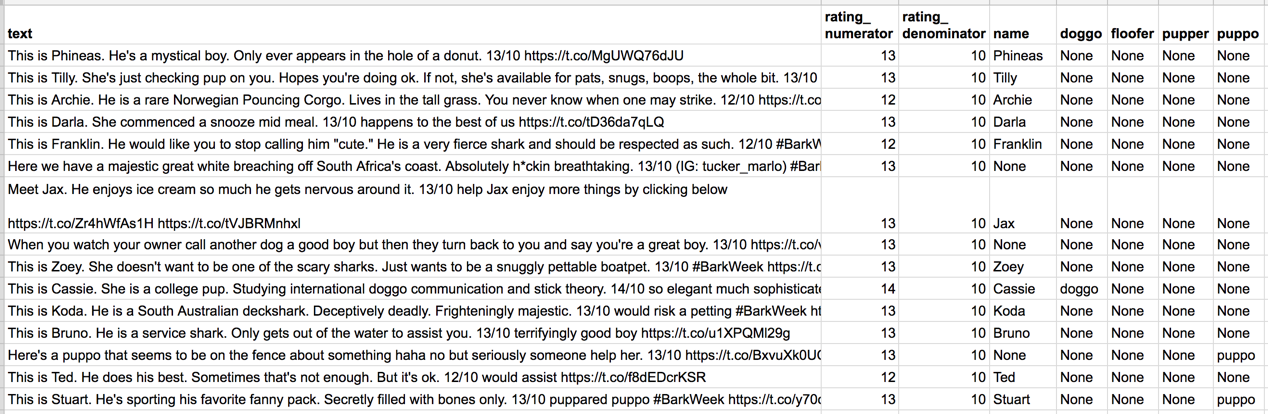
**背景**

你的目标：清洗 WeRateDogs 推特数据，创建有趣且可靠的分析和可视化。这份推特档案很棒，但是只包含基本的推特信息。要达到 "Wow!" 的效果，在分析和可视化前，还需要收集额外的数据、然后进行评估和清洗。

**数据**

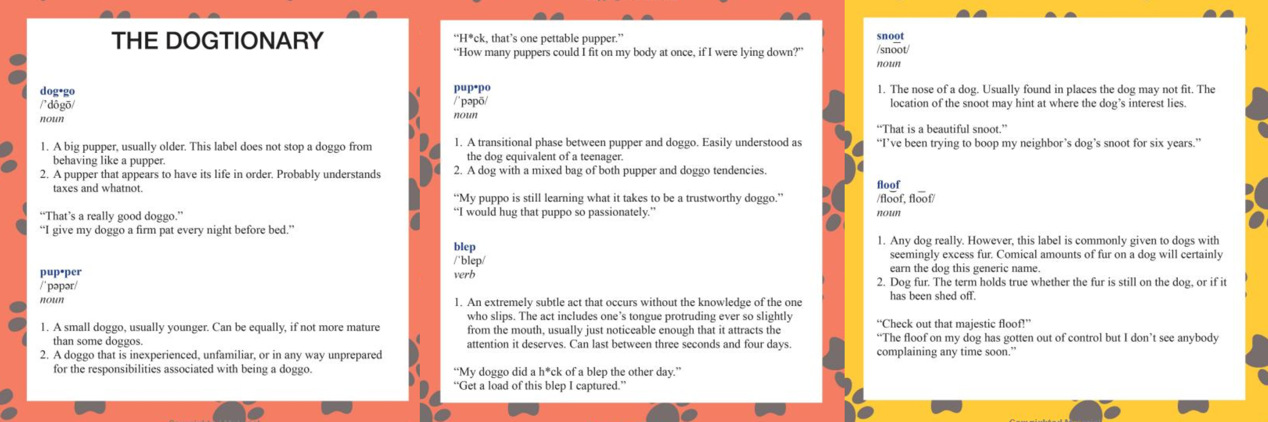
**完善推特档案**

WeRateDogs 的推特档案包括 5000 多条推特的基本信息，但并不包括所有内容。不过档案中有一列包含每个推特的文本，我用这一列数据提取了评分、狗的名字和“地位”（即 doggo、floofer、pupper 和 puppo）——这使数据得以“完善”。在这 5000 多条中，我只筛选出了 2356 条包含评分的推特数据。



*从每个推特文本中提取的数据*

我用编程方式从 text 推特原文中提取了评分、地位和名字等，但是并没有做好。评分并不都是正确的，狗的名字和地位也有不正确的 (狗的地位 stage 参见下面更多相关信息)。**如果你想用评分、地位和名字进行分析和可视化，需要评估和清洗这些列。**



*这个狗狗字典解释了狗狗的不同地位 (stage): doggo, pupper, puppo, 和 floof(er) (摘自 Amazon 上的*[*#WeRateDogs book*](https://www.amazon.com/WeRateDogs-Most-Hilarious-Adorable-Youve/dp/1510717145)*)*

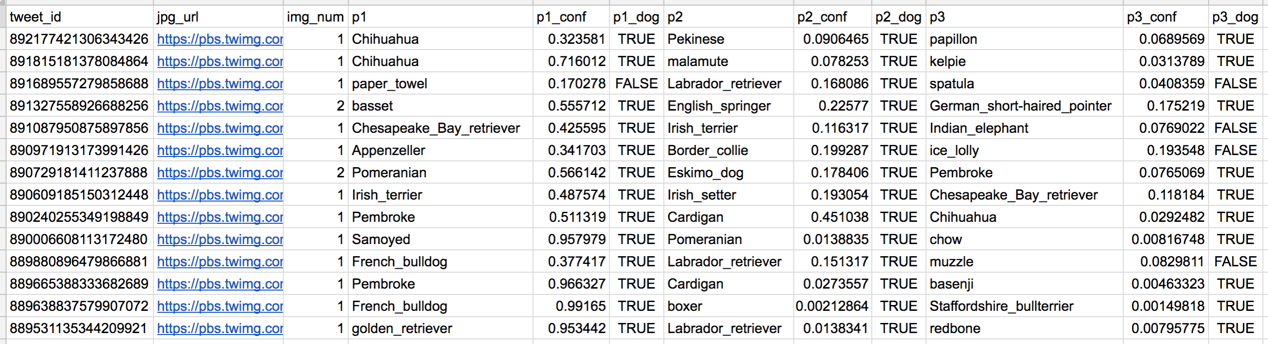
**通过推特 API 获取附加数据**

回到基础的推特档案：转发数（retweet count）和喜爱数（favorite count）是两个遗漏的列。幸运的是，从推特 API 中，任何人都可以收集到这些数据。其实，"任何人" 只是能获取最多 3000 条的最近推特数据。但是因为你拥有 WeRateDogs 推特档案和其中的推特 ID，你可以收集到这其中所有的 5000 多条推特。你将查询推特 API 来收集这些有价值的数据。

*特别提示：* 如果你无法访问 Twitter 的话，我们将直接给你提供返回的 Twitter 数据。你可以 [右键点击这里选择另存为](https://raw.githubusercontent.com/udacity/new-dand-advanced-china/master/%E6%95%B0%E6%8D%AE%E6%B8%85%E6%B4%97/WeRateDogs%E9%A1%B9%E7%9B%AE/tweet_json.txt) 下载。该文件为 txt 格式，每一行为一条独立的 twitter 信息，格式为 JSON。该文件比较大，下载可能需要几分钟。

**图像预测文件**

还有一件更酷的事情：我通过一个可以对狗狗种类进行分类的[神经网络](https://www.youtube.com/watch?v=2-Ol7ZB0MmU" \t "_blank)，运行这份推特档案中的所有图像。获取的结果：一份图像预测结果表格，其中包含了预测结果的前三名，推特 ID，图像 url 以及最可信的预测结果对应的图像编号（由于推特最多包含 4 个图片，所以编号为 1 到 4）。



*推特图像预测数据*

以该表格中的最后一行数据来理解各列数据：

* tweet\_id 是推特链接的最后一部分，位于 "status/" 后面 → <https://twitter.com/dog_rates/status/889531135344209921>
* jpg\_url 是预测的图像资源链接
* img\_num 最可信的预测结果对应的图像编号 → 1 推特中的第一张图片
* p1 是算法对推特中图片的一号预测 → 金毛犬
* p1\_conf 是算法的一号预测的可信度 → 95%
* p1\_dog 是一号预测该图片是否属于“狗”（有可能是其他物种，比如熊、马等） → True 真
* p2 是算法对推特中图片预测的第二种可能性 → 拉布拉多犬
* p2\_conf 是算法的二号预测的可信度 → 1%
* p2\_dog 是二号预测该图片是否属于“狗” → True 真
* 以此类推...

从推特中可以看到，对该图像的一号预测（p1）是准确的：



*名为 Stuart 的金毛犬*

所以这既有趣，也很不错。但是所有附加数据都需要收集、评估和清洗。所以才需要你。

**关键要点**

清洗这个项目的数据时要牢记几个要点：

* 我们只需要含有图片的原始评级 (不包括转发)。尽管数据集中有 5000 多条数据，但是并不是所有都是狗狗评分，并且其中有一些是转发。
* 完整地评估和清理整个数据集将需要大量时间，实践和展示数据处理技巧没有必要将这个数据集全部清理。因此，本项目的要求只是评估和清理此数据集中的至少 8 个质量问题和至少 2 个整洁度问题。
* 根据 [整洁数据 tidy data](https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html) 的规则要求，本项目的数据清理应该包括将三个数据片段进行合并。
* 如果分子评级超过分母评级，不需要进行清洗。这个 [特殊评分系统](http://knowyourmeme.com/memes/theyre-good-dogs-brent) 是 WeRateDogs 人气度较高的主要原因。（同样，也不需要删除分子小于分母的数据）
* 不必收集 2017 年 8 月 1 日之后的数据，你可以收集到这些推特的基本信息，但是你不能收集到这些推特对应的图像预测数据，因为你没有图像预测算法的使用权限。

## 项目细节

你在这个项目中的任务如下：

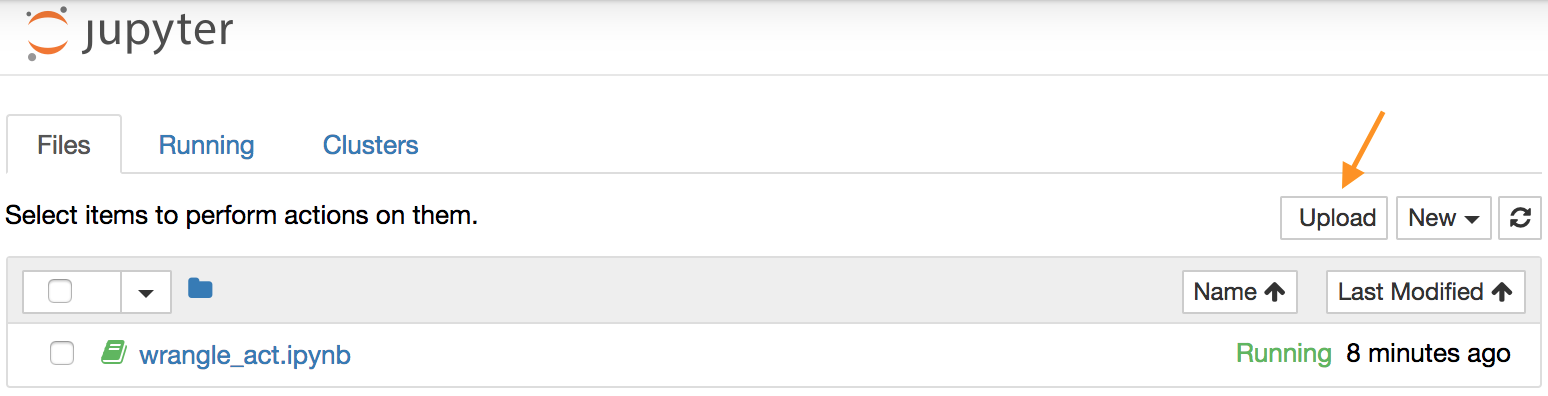
* 数据整理，其中包括：
  + 收集数据
  + 评估数据
  + 清洗数据
* 对清洗过的数据进行储存、分析和可视化
* 书面报告 1) 你的数据整理工作 和 2) 你的数据分析和可视化

### 对项目数据进行收集

在名字为 wrangle\_act.ipynb 的 Jupyter Notebook 中收集下面所述的三份数据：

1. WeRateDogs 的推特档案。这个数据文件是直接提供给你的，所以你可以将其当作是“资料来源：手头文件”来收集（参见课程 2：收集数据）。点击此链接下载：[twitter\_archive\_enhanced.csv](https://raw.githubusercontent.com/udacity/new-dand-advanced-china/master/%E6%95%B0%E6%8D%AE%E6%B8%85%E6%B4%97/WeRateDogs%E9%A1%B9%E7%9B%AE/twitter-archive-enhanced.csv" \t "_blank)
2. 推特图像的预测数据，即根据神经网络，对出现在每个推特中狗的品种（或其他物体、动物等）进行预测的结果。这个文件你需要使用 Python 的 [Requests](http://docs.python-requests.org/en/master/) 库和以下提供的 URL 来进行编程下载。下载用的 URL：[https://static-documents.s3.cn-north-1.amazonaws.com.cn/nd002/image-predictions.tsv](https://static-documents.s3.cn-north-1.amazonaws.com.cn/nd002/image-predictions.tsv" \t "_blank)) 。如果你想要回顾这部分的知识，可以复习【课程 2：收集数据——12. 资料来源：从互联网下载文件】。
3. 每条推特的额外附加数据，至少要包含转发数（retweet\_count）和喜欢数（favorite\_count）这两列，还可以收集任何你觉得有趣的列（注意：如果你的分析中不涉及到其他列则不需要收集）。使用 WeRateDogs 推特档案中的推特 ID，使用 Python [Tweepy 库](http://www.tweepy.org/" \t "_blank)查询 API 中每个推特的 JSON 数据，把所有 JSON 数据存储到一个名为 tweet\_json.txt 的文件中。每个推特的 JSON 数据应当写入单独一行。然后将这个 .txt 文件逐行读入一个 pandas DataFrame 中，至少包含tweet ID、retweet\_count 和 favorite\_count字段。**注：不要在项目提交中包含你的推特 API 密钥和访问令牌（可以用 \* 号代替）**。

如果你决定在项目 workspace 中完成项目，注意你可以点击操作面板右上角的 "Upload"（上传） 按钮在 Jupyter Notebook workspace上传文件。



Jupyter 记事本指示板上，箭头指向 " 上传 " 按钮

### 对项目数据进行评估

收集上述三个数据集之后，使用目测评估和编程评估的方式，对数据进行质量和清洁度的评估。在你的 wrangle\_act.ipynb Jupyter Notebook 中记录评估过程和结果，最终列出至少 **8 个质量问题** 和 **2 个清洁度问题**。要符合项目规范，必须对项目动机中的要求进行评估（参见上一页课程的 **关键要点** 标题）。

### 对项目数据进行清洗

对你在评估时列出的每个问题进行清洗。在 wrangle\_act.ipynb 展示清洗的过程。结果应该为一个优质干净整洁的主数据集（pandas DataFrame 类型） （如果都是以推特 ID 为观察对象的一些特征列，则清理最终只能有一个主数据集，如果有其他观察对象及其对应的特征字段，可以创建其他的数据集，同样需要清理）。同样地，必须符合项目动机的要点要求。

### 对项目数据进行存储、分析和可视化

将清理后的数据集存储到 CSV 文件中，命名为 twitter\_archive\_master.csv。如果有其他观察对象的数据集存在，需要多个表格，那么要给这些文件合理命名。另外，你也可以把清洗后的数据存储在 SQLite 数据库中（如果这样存储的话，该数据库文件也需要提交）。

在 wrangle\_act.ipynb Jupyter Notebook 中对清洗后的数据进行分析和可视化。必须生成至少 **3 个见解和 1 个可视化**。

### 项目汇报

创建一个 **300-600 字的书面报告**，命名为 wrangle\_report.pdf，在该报告中简要描述你的数据整理过程。这份报告可以看作是一份内部文档，供你的团队成员查看交流。

创建一个 **250 字以上的书面报告**，命名为 act\_report.pdf，在该报告中，你可以与读者交流观点，展示你使用整理过的数据生成的可视化图表。这份报告可以看作是一份外部文档，如博客帖子或杂志文章。

这两个报告都可以使用 Jupyter Notebook 中的 [Markdown 功能](http://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html)进行创建，然后以 PDF 格式下载（见下图）。不过你可能更倾向于使用文字处理软件，如 Google Docs 或 Microsoft Word，因为这两份报告 **并不包含代码**，使用文字处理软件更加便捷。

*特别提示：* 如果你无法访问 Twitter，我们将直接给你提供返回的 Twitter 数据。你可以 [右键点击这里选择另存为](https://raw.githubusercontent.com/udacity/new-dand-advanced-china/master/%E6%95%B0%E6%8D%AE%E6%B8%85%E6%B4%97/WeRateDogs%E9%A1%B9%E7%9B%AE/tweet_json.txt) 下载。该文件为 txt 格式，每一行为一条独立的 twitter 信息，格式为 JSON 。该文件比较大，下载可能需要几分钟。

**如何查询推特数据**

在这个项目中，你将会使用 [Tweepy](http://www.tweepy.org/" \t "_blank) 来查询推特 API，以获取 WeRateDogs 推特档案以外的附加数据。附加数据包括转发用户和喜欢用户。

有一些 API 是完全公开的，比如第 2 节课中的 MediaWiki (通过 [wptools](https://github.com/siznax/wptools/wiki" \t "_blank) library 获取)，而其他 API 则需要验证。推特 API 要求用户得到授权之后才能使用。这意味着在你运行 API 查询代码前，你需要配置好自己的推特应用。在此之前，你必须登录推特账户。[这份指南](https://www.slickremix.com/docs/how-to-get-api-keys-and-tokens-for-twitter/" \t "_blank)描述了配置过程。\* 一旦完成配置，一下代码将创建一个 API 对象，用于收集推特数据。该代码由 tweepy 文档中的 [Getting started](https://media.readthedocs.org/pdf/tweepy/latest/tweepy.pdf) 部分提供。

**import** tweepy

consumer\_key = 'YOUR CONSUMER KEY'

consumer\_secret = 'YOUR CONSUMER SECRET'

access\_token = 'YOUR ACCESS TOKEN'

access\_secret = 'YOUR ACCESS SECRET'

auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)

auth.set\_access\_token(access\_token, access\_secret)

api = tweepy.API(auth)

\**注释：如果你因为手机验证问题，无法安装应用，可以给*[*david.venturi@udacity.com*](mailto:david.venturi@udacity.com)*发邮件。*

推特数据以 JSON 格式存储在 Twitter 上。这个 [StackOverflow 答案](https://stackoverflow.com/questions/28384588/twitter-api-get-tweets-with-specific-id" \t "_blank)已经做出了清晰描述，教你如何利用 Tweepy 通过推特 ID 获取推特 JSON 数据。注意，在调用 get\_status 后， 请将参数 tweet\_mode 设置为 'extended'，即 api.get\_status(tweet\_id, tweet\_mode='extended')，这将会非常有用。

你还需要注意的是，档案中有少数 ID 对应的推特信息也许已经被删除。[Try-except 块](https://wiki.python.org/moin/HandlingExceptions" \t "_blank)也许会对你很有帮助。

**不要在提交项目时包含你的 API 密钥和访问令牌**

**不要**在提交时包含 API 密钥和访问令牌。这是 API 和公共代码的标准惯例。

**推特速率限制**

推特 API 具有速率限制。速率限制用于控制服务器发出或接收流量的速率。根据[推特速率限制信息](https://developer.twitter.com/en/docs/basics/rate-limiting" \t "_blank)：

*速率限制为 15 分钟的时间间隔。*

为了在 WeRateDogs 推特档案中查询所有推特 ID，至少需要 20-30 分钟的运行时间。为了使数据更加整洁，你可以在查询后输出每个推特 ID，并[使用定时器编码](https://stackoverflow.com/questions/7370801/measure-time-elapsed-in-python" \t "_blank)。你也可以在 [tweepy.api class](http://docs.tweepy.org/en/v3.2.0/api.html" \l "API" \t "_blank) 中将参数 wait\_on\_rate\_limit 和 wait\_on\_rate\_limit\_notify 设置为 True。

**编写和读取推特 JSON**

查询每个推特 ID 后，你将要编写 JSON 数据，并按要求将其存入 tweet\_json.txt 文件中，每个推特 JSON 数据占一行。接着你可以逐行读取这个文件，来创建一个 pandas DataFrame 以便在之后进行数据评估和清洗。你还可以阅读这篇来自 Stack Abuse 的文章，名为[使用 Python 在文件中读取和编写 JSON（Reading and Writing JSON to a File in Python）](http://stackabuse.com/reading-and-writing-json-to-a-file-in-python/) ,这对你的深入学习非常有用。



*推特标识*