## Contents

1 GREAT FOR 3  
2 BENEFITS 5  
3 TYPICAL WORKFLOW 7  
4 GETTING STARTED 9  
  4.1 DESKTOP 9  
  4.2 SERVER/CLOUD 9  
5 FEATURES 11  
6 DATAFRAME COLUMNS 13  
7 FUNCTIONS 15  
  7.1 DATA COLLECTION 15  
  7.2 DATA PROCESSING 16  
  7.3 REPORTING 16  
  7.4 PLOTS 16  
  7.5 PERFORMANCE 17  
  7.6 UPGRADE 17  
  7.7 BUILT ON 17
SOMECODE is a research platform for serious observation and analysis of Twitter data. SOMECODE brings together 9 years of unbroken continuity in developing social media research tools. Previous tools and processes developed by the contributor team are in daily use by many FORTUNE100 companies and major advertising agencies. SOMECODE is the solution we always wanted to build, but due to the kinds of restraints commercial entities have, never got to.

```bash
pip install somecode
```

All you need to have is Python 2.7 and the somecode installation will take care of all the dependencies.
Somecode is great for researching a variety of topics, for example:

- Public figures
- Brands and organizations
- Special events (e.g. election)
- Emerging or ongoing crisis
- Ideas (e.g. radicalization)
- Websites and communities

Somecode comes with built-in scoring system and various signals that are specifically targeted at identifying bot, spam and troll accounts in Twitter, to help researchers better understand malicious techniques used in computer-aided propaganda.
CHAPTER 2

BENEFITS

Somecode takes you from an idea to understanding with one command in 20 seconds and allows any researcher to start with serious social media research in minutes from being a total novice.

- Equal or better capabilities vs. best industrial solution
- Up to 10 million tweets per day with single API key
- Optimized for minimizing out-of-scope time use (configuration, data wrangling, etc.)
- Supports both streaming and rest API endpoints for both status (tweet) and user objects
- Provides an ideal environment for academic research and publication

SOMECODE is built by researchers for researchers and is very easy to extend / customize to suit specific needs. For most research scopes SOMECODE will work “out of the box”. It has very few dependencies (below) and takes minutes to deploy for your first research project.
CHAPTER 3

TYPICAL WORKFLOW

Depending on the need of the researcher at the time of use, a typical scenario involves no more than 2 or 3 function calls and depending on the size of the sample and the function used for collecting the data, no more than 1 minute. Such a scenario may involve:

1. Use one of the ‘data collector’ functions to get the data you need
2. Use one or more of the ‘report’ functions to visualize the data
3. Based on the findings, use one of the ‘data collectors’ to get drill-down data
4. Export report and dataset for reference / later use
CHAPTER 4

GETTING STARTED

The easiest case is:

```
pip install somecode
```

The hardest case is to first run this in shell:

```
sudo apt-get update -y; sudo apt-get install python-pip -y; sudo pip install --upgrade pip; sudo pip install somecode; sudo apt-get update; sudo apt-get install python-dev -y; sudo pip install entropy;
```

And then run this in Jupyter / python shell:

```
import nltk; nltk.download('vader_lexicon'); nltk.download('vader_lexicon');
```

It will take less than a minute and because we’re only using common packages, the installation should be painless in any case.

4.1 DESKTOP

SOMECODE runs on any regular low-end laptop with a common web browser. Tested on Linux and Mac OS X. Jupyter is highly recommended.

4.2 SERVER/CLOUD

SOMECODE runs on Amazon Nano (or similar) for $5 per month. Tested on Ubuntu 14.04LTS

For both options:

```
pip install somecode
```
SOMECODE is very easy to customize / extend if you would feel the need to do it. Even if you are a beginner python learner.

100% fun, 0% mindless wrangling.
Collecting, processing and analyzing Twitter data comes with many caveats and obstacles, a factor that has kept most researchers oblivious to the potential Twitter data has. Many of Somecode’s features are related with making all of that completely disappear.

Some of the things we’ve figured out for you include:

- System performance
- Twitter API rate-limit management
- JSON parsing
- Signal selection
- Plot configuration

The ‘data collector’ functions all return the same format pandas DataFrame, which means that you can use any plots, models, etc. to go where you want to go with the data.
All of the ‘data collection’ methods (‘search’, ‘stream’, ‘timeline’, ‘flatfile’) return a pandas dataframe with direct signals from Twitter, together with SOMECODE scores and other inferred metrics.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>ABOUT</th>
<th>DTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>days_since_creation</td>
<td>user</td>
<td>int64</td>
</tr>
<tr>
<td>influence_score</td>
<td>scores</td>
<td>int64</td>
</tr>
<tr>
<td>reach_score</td>
<td>scores</td>
<td>int64</td>
</tr>
<tr>
<td>quality_score</td>
<td>scores</td>
<td>int64</td>
</tr>
<tr>
<td>retweet_count</td>
<td>tweet</td>
<td>int64</td>
</tr>
<tr>
<td>text</td>
<td>tweet</td>
<td>object</td>
</tr>
<tr>
<td>user_tweets</td>
<td>user</td>
<td>int64</td>
</tr>
<tr>
<td>user_favourites</td>
<td>user</td>
<td>int64</td>
</tr>
<tr>
<td>user_followers</td>
<td>user</td>
<td>int64</td>
</tr>
<tr>
<td>user_following</td>
<td>user</td>
<td>int64</td>
</tr>
<tr>
<td>user_listed</td>
<td>user</td>
<td>int64</td>
</tr>
<tr>
<td>handle</td>
<td>user</td>
<td>object</td>
</tr>
<tr>
<td>created_at</td>
<td>user</td>
<td>datetime64</td>
</tr>
<tr>
<td>default_profile</td>
<td>user</td>
<td>bool</td>
</tr>
<tr>
<td>egg_account</td>
<td>user</td>
<td>bool</td>
</tr>
<tr>
<td>description</td>
<td>user</td>
<td>object</td>
</tr>
<tr>
<td>location</td>
<td>user</td>
<td>object</td>
</tr>
<tr>
<td>timezone</td>
<td>user</td>
<td>object</td>
</tr>
<tr>
<td>compound</td>
<td>sentiment</td>
<td>float</td>
</tr>
<tr>
<td>neu</td>
<td>sentiment</td>
<td>any</td>
</tr>
<tr>
<td>neg</td>
<td>sentiment</td>
<td>any</td>
</tr>
<tr>
<td>pos</td>
<td>sentiment</td>
<td>any</td>
</tr>
</tbody>
</table>
There are four categories of functions in SOMECODE:

- Data Collection
- Data Processing
- Reporting
- Export

### 7.1 DATA COLLECTION

There are four ways to get data into SOMECODE.

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>REQUIRED PARAMETERS</th>
<th>DATA SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>search()</td>
<td>keyword</td>
<td>max 3200 tweets</td>
</tr>
<tr>
<td>stream()</td>
<td>keyword or userid</td>
<td>up to 100 per second</td>
</tr>
<tr>
<td>timeline()</td>
<td>handle</td>
<td>up to 3200 tweets</td>
</tr>
<tr>
<td>flatfile()</td>
<td>filename</td>
<td>any size</td>
</tr>
</tbody>
</table>

To get 1000 tweets for keyword “election”:

```python
some.search("election",1000)
```

To pen a stream with keyword “election”:

```python
some.stream("election")
```

To get maximum number of tweets from @realdonaldtrump timeline:

```python
some.timeline("realdonaldtrump")
```

To get tweets from a file:
7.2 DATA PROCESSING

While it is possible to call any of the 20 or so modules included in Somecode as standalone functions, the ‘data processing’ modules are a little different in the sense that they are not called directly with the exception of keywords() which makes sense also to call directly.

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>REQUIRED PARAMETERS</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_frame()</td>
<td>Somecode dataframe</td>
<td>max 3200 tweets</td>
</tr>
<tr>
<td>data_prep()</td>
<td>Somecode dataframe</td>
<td>Just for flatfile()</td>
</tr>
<tr>
<td>keywords()</td>
<td>Any series with text</td>
<td>basic keyword stats</td>
</tr>
</tbody>
</table>

To compute entropy and other signals for textual data:

```python
some.keywords(df)
```

Various additional semantic analysis is possible as part of freq_plot() and cooc_plot() reporting function.

7.3 REPORTING

There are two kinds of reporting capabilities; plots and tables. The tables come from the pretty.py library and plots are heavily customized Seaborn and Matplotlib plots.

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>KIND OF REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>age_plot()</td>
<td>Bubble chart</td>
</tr>
<tr>
<td>bars()</td>
<td>Bar chart</td>
</tr>
<tr>
<td>cooc_plot()</td>
<td>Bubble chart</td>
</tr>
<tr>
<td>freq_plot()</td>
<td>Side-by-side bar</td>
</tr>
<tr>
<td>hist_plot()</td>
<td>Histogram</td>
</tr>
<tr>
<td>neg_plot()</td>
<td>Bubble chart</td>
</tr>
<tr>
<td>neg2_plot()</td>
<td>Bar chart</td>
</tr>
<tr>
<td>retweet_plot()</td>
<td>Bubble chart</td>
</tr>
</tbody>
</table>

For the Pretty descriptive tables:

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>KIND OF REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>pretty.header()</td>
<td>Produces pretty header</td>
</tr>
<tr>
<td>pretty.table()</td>
<td>Produces pretty table</td>
</tr>
<tr>
<td>pretty.data()</td>
<td>Prepares data for table</td>
</tr>
<tr>
<td>pretty.toggle()</td>
<td>Hide code cells</td>
</tr>
<tr>
<td>pretty.warnings()</td>
<td>Turns of warnings</td>
</tr>
</tbody>
</table>

7.4 PLOTS

When you are on Jupyter, on the first line you must declare:

```python
%matplotlib inline
```

Otherwise you will not see the plots.
7.5 PERFORMANCE

During the 2016 election, SOMECODE topical, sentiment, scoring and other computations have been tested in up to 200,000 tweets per hour volume using a single $50 per month server (8gb RAM) where the computations required for every 10 minute cycle were generally completed in 20 seconds.

7.6 UPGRADE

At anytime, you can update the current version of SOMECODE:

```
sudo pip install somecode --upgrade
```

7.7 BUILT ON

Frankly speaking, SOMECODE would not be possible without all the amazing technology solutions it’s based on. What SOMECODE does, is put a few key technologies together, with “business logic” that came from working on over a thousand social media research projects since 2005. Somecode uses pandas, numpy, seaborn and matplotlib libraries heavily.

Other than that, dependent on the system, you should have minimal dependencies to worry about. Also if you’re not using it already, I highly recommend Jupyter (http://jupyter.org/). It helps make programming much more about fun, and less about frustration.