Wrangle Report

The three datasets viz. archive, image prediction, and API-generated reactions were assessed in two ways:

- Visual assessment
- Programmatic assessment

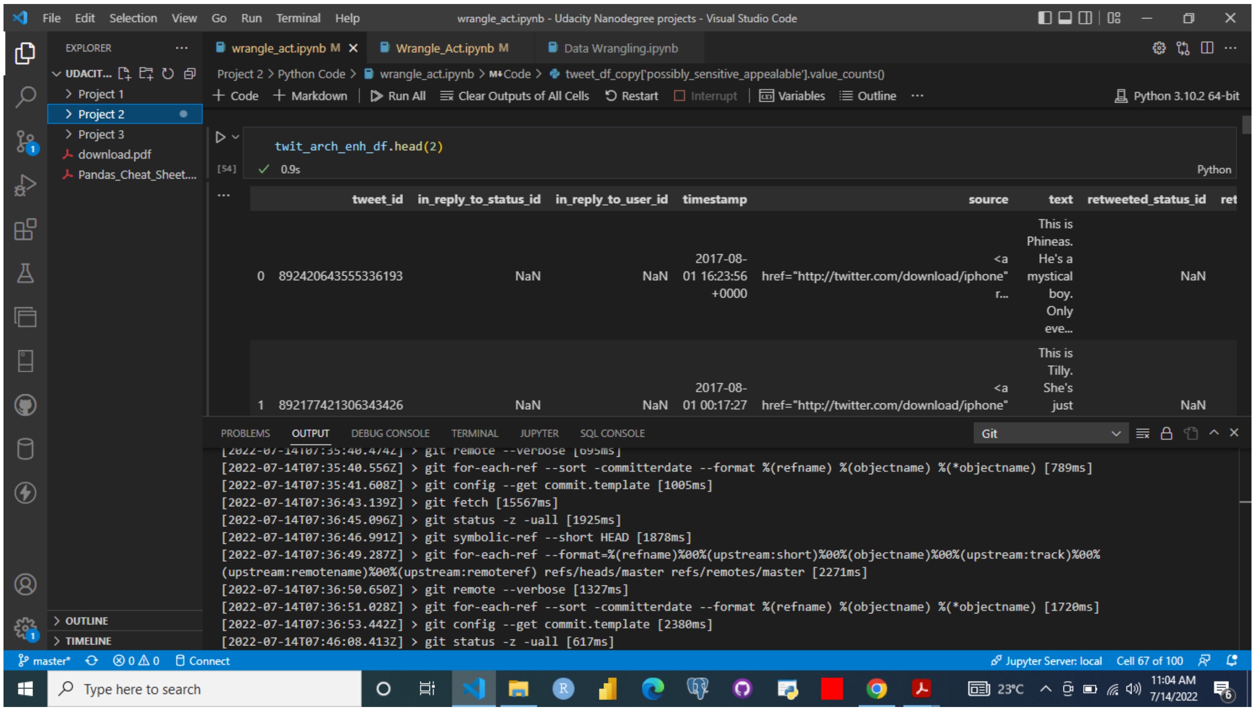
Visual Assessment

This involves the use of spreadsheet applications such as Google Sheet, Microsoft Excel, and some pandas functions in Python.

In order to visually access these datasets adequately, I tried to view and scroll through the data examining what might be missing, wrongly inputted, or inaccurate structures.

However, I recognize the limitations of using spreadsheets for assessment or examination as in many cases data might be so voluminous or be available in a format not so easily readable by humans. Hence, I utilized the pandas' Python package. Some of the pandas' data frame methods such as

1. Head: This shows the first five observations. The default shows five observations but this may be modified by parsing the count as an argument. I choose to set the argument to 2 to limit the space.

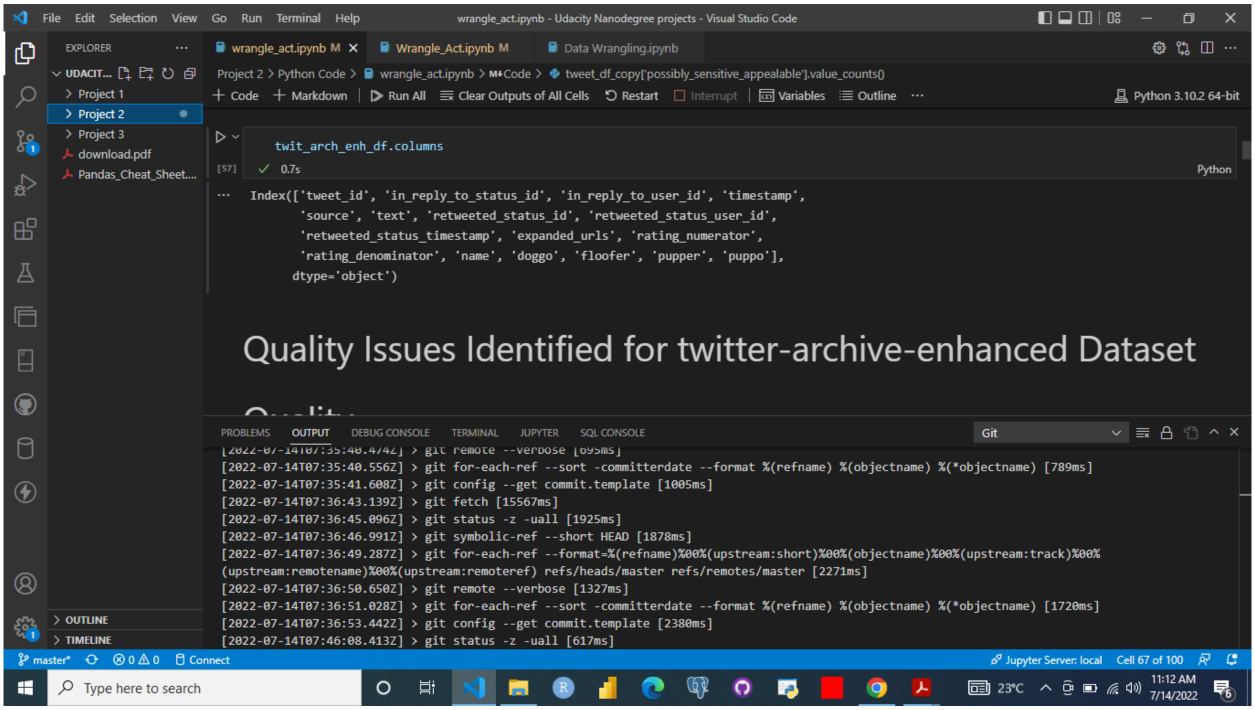

File Edit Selection View Go Run Terminal Help
wrangle_act.ipynb - Udacity Nanodegree projects - Visual Studio Code
EXPLORER
UDACIT...
Project 1
Project 2
Project 3
download.pdf
Pandas_Cheat_Sheet....
wrangle_act.ipynb M
Wrangle_Act.ipynb M
Data Wrangling.ipynb
Project 2 > Python Code > wrangle_act.ipynb > M↓Code > tweet_df_copy['possibly_sensitive_appealable'].value_counts()
+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline
Python 3.10.2 64-bit
[54]
0.9s
Python
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER SQL CONSOLE
Git
OUTLINE
TIMELINE
master*
⊗ 0 ⚠ 0
Connect
Jupyter Server: local
Cell 67 of 100
Type here to search
23°C
11:04 AM
7/14/2022

```python
twit_arch_enh_df.head(2)
```

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	ret
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27	<a href="http://twitter.com/download/iphone"	This is Tilly. She's just	NaN	

```
[2022-07-14T07:35:40.474Z] > git remote --verbose [695ms]
[2022-07-14T07:35:40.556Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [789ms]
[2022-07-14T07:35:41.608Z] > git config --get commit.template [1005ms]
[2022-07-14T07:36:43.139Z] > git fetch [15567ms]
[2022-07-14T07:36:45.096Z] > git status -z -uall [1925ms]
[2022-07-14T07:36:46.991Z] > git symbolic-ref --short HEAD [1878ms]
[2022-07-14T07:36:49.287Z] > git for-each-ref --format=%(refname)%00%(upstream:short)%00%(objectname)%00%(upstream:track)%00%
(upstream:remotename)%00%(upstream:remoteref) refs/heads/master refs/remotes/master [2271ms]
[2022-07-14T07:36:50.650Z] > git remote --verbose [1327ms]
[2022-07-14T07:36:51.028Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [1720ms]
[2022-07-14T07:36:53.442Z] > git config --get commit.template [2380ms]
[2022-07-14T07:46:08.413Z] > git status -z -uall [617ms]
```

2. Columns: This shows the names of the columns in the data frame.



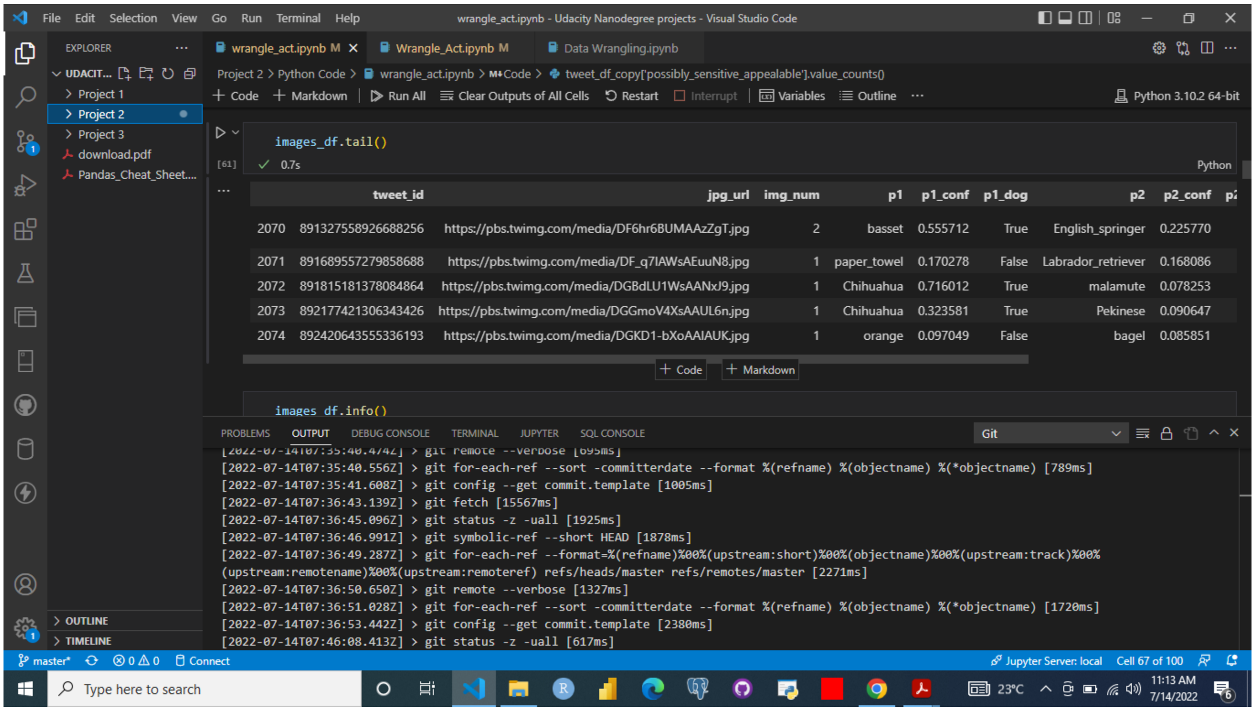
```
twit_arch_enh_df.columns
```

```
Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
       'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
       'retweeted_status_timestamp', 'expanded_urls', 'rating_numerator',
       'rating_denominator', 'name', 'doggo', 'floofer', 'pupper', 'puppo'],
      dtype='object')
```

Quality Issues Identified for twitter-archive-enhanced Dataset

```
[2022-07-14T07:35:40.474Z] > git remote --verbose [695ms]
[2022-07-14T07:35:40.556Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [789ms]
[2022-07-14T07:35:41.608Z] > git config --get commit.template [1005ms]
[2022-07-14T07:36:43.139Z] > git fetch [15567ms]
[2022-07-14T07:36:45.096Z] > git status -z -uall [1925ms]
[2022-07-14T07:36:46.991Z] > git symbolic-ref --short HEAD [1878ms]
[2022-07-14T07:36:49.287Z] > git for-each-ref --format=%(refname)%00%(upstream:short)%00%(objectname)%00%(upstream:track)%00%
(upstream:remotename)%00%(upstream:remoteref) refs/heads/master refs/remotes/master [2271ms]
[2022-07-14T07:36:50.650Z] > git remote --verbose [1327ms]
[2022-07-14T07:36:51.028Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [1720ms]
[2022-07-14T07:36:53.442Z] > git config --get commit.template [2380ms]
[2022-07-14T07:46:08.413Z] > git status -z -uall [617ms]
```

3. Tail: This shows a specified number of observations. The default shows five observations but the same may be modified by parsing in the count as an argument.



```
images_df.tail()
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf
2070	891327558926688256	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	2	basset	0.555712	True	English_springer	0.225770
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	1	paper_towel	0.170278	False	Labrador_retriever	0.168086
2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	1	Chihuahua	0.716012	True	malamute	0.078253
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	1	Chihuahua	0.323581	True	Pekinese	0.090647
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1	orange	0.097049	False	bagel	0.085851

```
images_df.info()
```

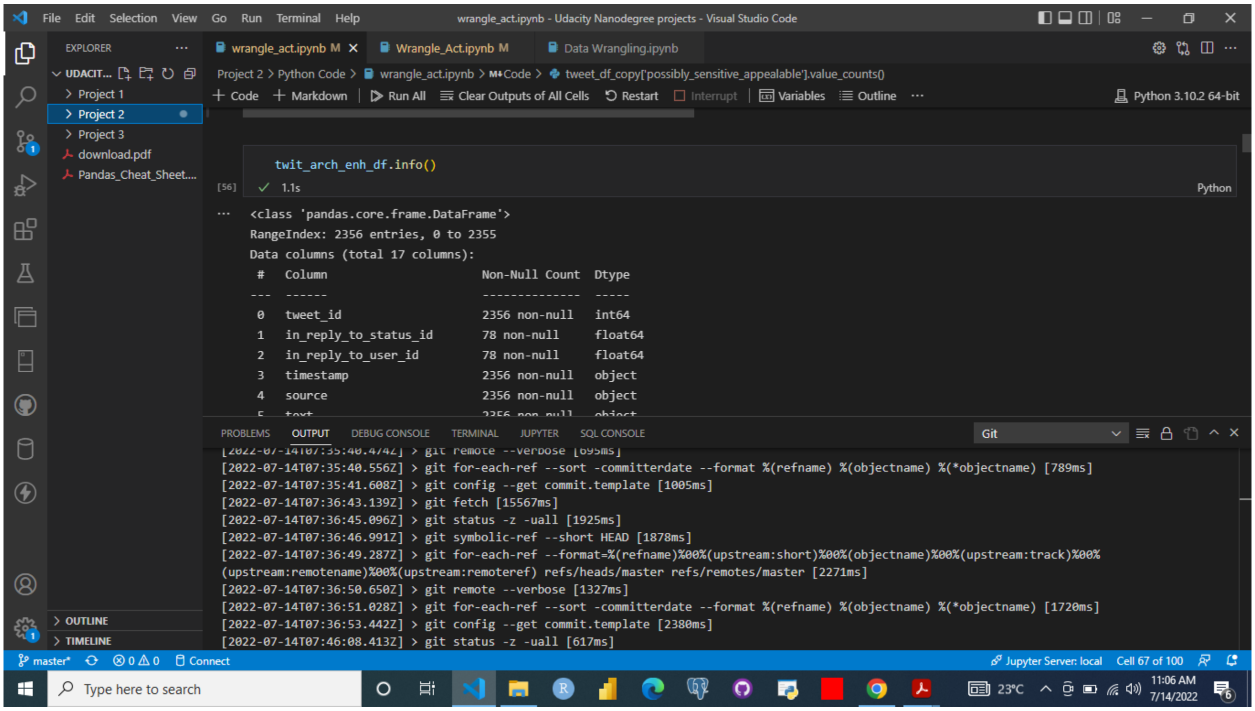
```
[2022-07-14T07:35:40.474Z] > git remote --verbose [695ms]
[2022-07-14T07:35:40.556Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [789ms]
[2022-07-14T07:35:41.608Z] > git config --get commit.template [1005ms]
[2022-07-14T07:36:43.139Z] > git fetch [15567ms]
[2022-07-14T07:36:45.096Z] > git status -z -uall [1925ms]
[2022-07-14T07:36:46.991Z] > git symbolic-ref --short HEAD [1878ms]
[2022-07-14T07:36:49.287Z] > git for-each-ref --format=%(refname)%00%(upstream:short)%00%(objectname)%00%(upstream:track)%00%
(upstream:remotename)%00%(upstream:remoteref) refs/heads/master refs/remotes/master [2271ms]
[2022-07-14T07:36:50.650Z] > git remote --verbose [1327ms]
[2022-07-14T07:36:51.028Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [1720ms]
[2022-07-14T07:36:53.442Z] > git config --get commit.template [2380ms]
[2022-07-14T07:46:08.413Z] > git status -z -uall [617ms]
```

Programmatic Assessment

Programmatic assessment involves the use of code to evaluate the possible causes of dirty or untidy datasets.

The following are the pandas' data frame methods utilized:

1. Info: The info method provides a series of vital information, such as the dimension, columns list, datatypes, data types, the total number of rows, and columns, total nonnull entries, memory usage, etc.

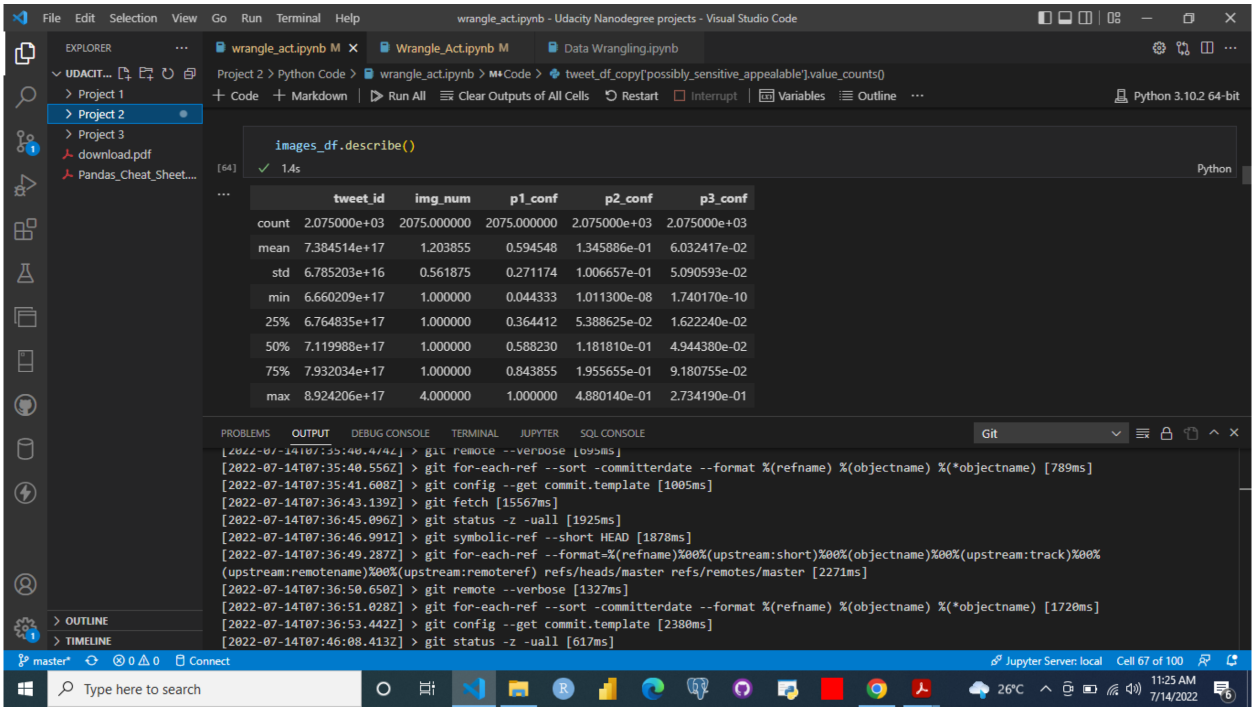

wrangle_act.ipynb - Udacity Nanodegree projects - Visual Studio Code
wrangle_act.ipynb M
Wrangle_Act.ipynb M
Data Wrangling.ipynb
Project 2 > Python Code > wrangle_act.ipynb > Code > tweet_df_copy['possibly_sensitive_appealable'].value_counts()
Python 3.10.2 64-bit
Jupyter Server: local
Cell 67 of 100

```
twit_arch_enh_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   tweet_id                   2356 non-null   int64
 1   in_reply_to_status_id      78 non-null     float64
 2   in_reply_to_user_id        78 non-null     float64
 3   timestamp                  2356 non-null   object
 4   source                     2356 non-null   object
 5   text                       2356 non-null   object
```

```
[2022-07-14T07:35:40.474Z] > git remote --verbose [695ms]
[2022-07-14T07:35:40.556Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [789ms]
[2022-07-14T07:35:41.608Z] > git config --get commit.template [1005ms]
[2022-07-14T07:36:43.139Z] > git fetch [15567ms]
[2022-07-14T07:36:45.096Z] > git status -z -uall [1925ms]
[2022-07-14T07:36:46.991Z] > git symbolic-ref --short HEAD [1878ms]
[2022-07-14T07:36:49.287Z] > git for-each-ref --format=%(refname)%00%(upstream:short)%00%(objectname)%00%(upstream:track)%00%
(upstream:remotename)%00%(upstream:remoteref) refs/heads/master refs/remotes/master [2271ms]
[2022-07-14T07:36:50.650Z] > git remote --verbose [1327ms]
[2022-07-14T07:36:51.028Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [1720ms]
[2022-07-14T07:36:53.442Z] > git config --get commit.template [2380ms]
[2022-07-14T07:46:08.413Z] > git status -z -uall [617ms]
```

2. Describe: This gives a quick summary of statistics such as mean, min, max, quartiles, etc.


wrangle_act.ipynb - Udacity Nanodegree projects - Visual Studio Code
wrangle_act.ipynb M
Wrangle_Act.ipynb M
Data Wrangling.ipynb
Project 2 > Python Code > wrangle_act.ipynb > Code > tweet_df_copy['possibly_sensitive_appealable'].value_counts()
Python 3.10.2 64-bit
Jupyter Server: local
Cell 67 of 100

```
images_df.describe()
```

	tweet_id	img_num	p1_conf	p2_conf	p3_conf
count	2.075000e+03	2075.000000	2075.000000	2.075000e+03	2.075000e+03
mean	7.384514e+17	1.203855	0.594548	1.345886e-01	6.032417e-02
std	6.785203e+16	0.561875	0.271174	1.006657e-01	5.090593e-02
min	6.660209e+17	1.000000	0.044333	1.011300e-08	1.740170e-10
25%	6.764835e+17	1.000000	0.364412	5.388625e-02	1.622240e-02
50%	7.119988e+17	1.000000	0.588230	1.181810e-01	4.944380e-02
75%	7.932034e+17	1.000000	0.843855	1.955655e-01	9.180755e-02
max	8.924206e+17	4.000000	1.000000	4.880140e-01	2.734190e-01

```
[2022-07-14T07:35:40.474Z] > git remote --verbose [695ms]
[2022-07-14T07:35:40.556Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [789ms]
[2022-07-14T07:35:41.608Z] > git config --get commit.template [1005ms]
[2022-07-14T07:36:43.139Z] > git fetch [15567ms]
[2022-07-14T07:36:45.096Z] > git status -z -uall [1925ms]
[2022-07-14T07:36:46.991Z] > git symbolic-ref --short HEAD [1878ms]
[2022-07-14T07:36:49.287Z] > git for-each-ref --format=%(refname)%00%(upstream:short)%00%(objectname)%00%(upstream:track)%00%
(upstream:remotename)%00%(upstream:remoteref) refs/heads/master refs/remotes/master [2271ms]
[2022-07-14T07:36:50.650Z] > git remote --verbose [1327ms]
[2022-07-14T07:36:51.028Z] > git for-each-ref --sort -committerdate --format %(refname) %(objectname) %(*objectname) [1720ms]
[2022-07-14T07:36:53.442Z] > git config --get commit.template [2380ms]
[2022-07-14T07:46:08.413Z] > git status -z -uall [617ms]
```

3. Shape: To get the dimension of the data frame in terms of the number of rows and columns.

Other programmatic assessment functions used include:

4. IsNull: This checks null values in the dataset. For this dataset, we had a lot of them.
5. Duplicated: Although we did not have cases of duplicated tweet_id, this function checks for duplicates.
6. Unique: This finds unique values and combined with the sum will have a reasonable insight into the data
7. Count
8. Sum
9. Isna
10. Value_counts