

Final Report of Internship Program 2021  
on  
“Real-Time Insights from Social Media Data”

**MEDTOUREASY**



March 27, 2021

## Acknowledgement

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of this project. I express my sincere gratitude to MedTourEasy family for providing me an opportunity to undergo this internship programme. I am thankful to training coordinator for providing to me constant inspiration with his presence and also guidance for my future endeavors, during the course of training. Lastly, I would like to thank the almighty and my parents for their moral support and my co-trainees with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.

## Contents

|  |    |
|--|----|
| Acknowledgement.....                                 | 2  |
| Abstract.....  | 5  |
| 1. Introduction .....                                | 6  |
| 1.1 About the company .....                          | 6  |
| 1.2 About the project.....                           | 6  |
| 1.3 Project Objectives.....                          | 6  |
| 2. What we are doing here? .....                     | 7  |
| 2.1 steps involved in social media analytics .....   | 7  |
| 2.1.1 Data identification.....                       | 7  |
| 2.1.2 Data analysis.....                             | 8  |
| 2.1.3 Information interpretation .....               | 8  |
| 3 Redefining the whole project.....                  | 9  |
| 3.1 Define problem.....                              | 9  |
| 3.2 Locate Datasource .....                          | 9  |
| 3.3 Search Datasource and create the Data model..... | 9  |
| 3.4 Run analytics and get insights.....              | 9  |
| 4 DataSource and Data.....                           | 9  |
| 4.1 Why analyze Twitter Data? .....                  | 9  |
| 4.2 Collecting data through Twitter API .....        | 9  |
| 4.3 Streaming API .....                              | 10 |
| 4.4 Tweepy: Module for Accessing Twitter API .....   | 10 |
| 5 Languages and Software and packages used .....     | 10 |
| 5.1 Python .....                                     | 10 |
| 5.2 Jupyter Notebook (IDE) .....                     | 10 |
| 5.3 Packages .....                                   | 11 |
| 5.3.1 Numpy.....                                     | 11 |
| 5.3.2 Pandas .....                                   | 11 |
| 5.3.3 Matplotlib .....                               | 12 |
| 5.3.3 Seaborn .....                                  | 12 |
| 6 Information and data we have before starting ..... | 13 |
| 6.1 Tasks list .....                                 | 13 |
| 6.2 Datasets .....                                   | 13 |
| 6.3 Format of provided data(.json) .....             | 14 |
| 7 Project.....                                       | 14 |
| 7.1 Task 1: Local and global thought patterns.....   | 14 |

|   |    |
|---|----|
| 7.2 Task 2: Prettifying the output .....  | 17 |
| 7.3 Task 3: Finding common trends .....   | 22 |
| 7.4 Exploring the hot trend.....          | 24 |
| 7.5 Digging deeper.....                   | 28 |
| 7.6 Frequency analysis .....              | 29 |
| 7.7 Activity around the trend.....        | 31 |
| 7.8 A table that speaks a 1000 words..... | 32 |
| 7.9 Analyzing used languages .....        | 34 |

## Abstract

Social media has become an integral part of our daily life. We can't think about life without social media. Social media made it easy for us to connect to people but it can also help us to understand them. Understanding people what they are saying? what is the notion of the public towards a particular thing or person? and many more things can be done with the help of social media. It can be done by using the right things.

With the help of this internship, I came across many fabulous things in the field of data science. From this project, I am showcasing some of them. This project is "Real-Time Insights from Social Media Data".

In this project, I used some basic tools of social media analytics. Social media analytics means the process of gathering and analyzing data from social networks such as Facebook, Instagram, LinkedIn and Twitter. It is commonly used by marketers to track online conversations about products and companies.

As the name of the project clearly states that Real-time. It means the data we have is the latest to our knowledge. For this project, we used a language named python along with IDLE named jupyter Notebooks.

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable it uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical construction than other languages.

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

The data for this project comes from Twitter. Some part of the Data of tweets on Twitter is available. Only 1% of Whole Twitter data is readily available for research and analytics purpose. The rest of the data is not available for public access. This size might seem small to you but when you see it in the bigger picture it is lots of data. As estimated Twitter have daily more than 15 billion tweets 1% of that is still lots of data.

Our main task is to find trends that are on boom worldwide as well as in the US, find the most preferred language while tweeting worldwide and visually confirm it by plotting histogram for languages.

## 1. Introduction

### 1.1 About the company

MedTourEasy, a global healthcare company, provides you with the informational resources needed to evaluate your global options. It helps you find the right healthcare solution based on specific health needs, affordable care while meeting the quality standards that you expect to have in healthcare. MedTourEasy improves access to healthcare for people everywhere. It is an easy-to-use platform and service that helps patients to get medical second opinions and to schedule affordable, high-quality medical treatment abroad.

### 1.2 About the project

Fear of missing out, curiosity, self-esteem, speed: it's like social media has changed our basic human needs; these baits are keeping us hooked and engaged. And Twitter is a master at this game. Elon Musk's tweets keep Wall Street on its toes; Trump's tweets have the potential of starting wars — Twitter has this huge influence on the world because of the type of its users. Data from Twitter-storms is available in near real-time. This means we can learn about the big waves of thoughts and moods around the world as they arise. So of course, we are not going to miss the chance to analyze this treasure trove.

In this project, you will use pre-downloaded datasets to understand the nuts and bolts of Twitter Data. In particular, you will do a thorough analysis of a hot-trend.

This project provides the opportunity to apply the skills covered in DataCamp's Analyzing Social Media Data in Python course.

### 1.3 Project Objectives

This project has two parts first is to find trends which are going on worldwide and in the US respectively. Second use the data set of #welovetheearth and find retweet data of this particular hashtag.

This whole project is divided into multiple tasks each task have its importance and it also shows the workflow of the whole project. Those are given below are brief points that define the project.

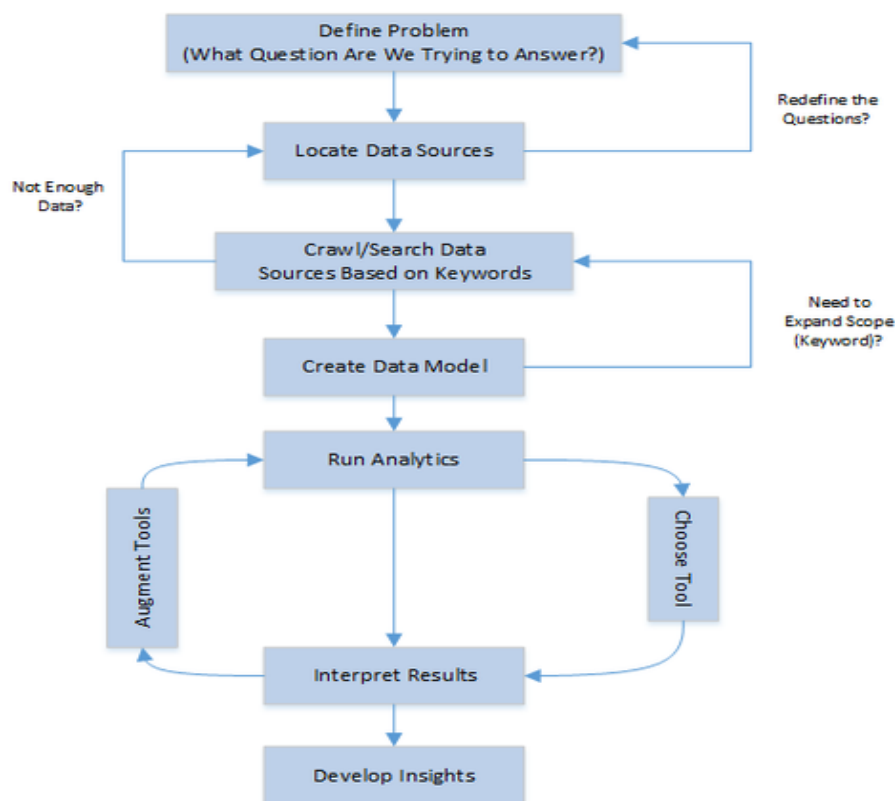
- Local and global thought patterns
- Prettifying the output
- Finding common trends
- Exploring the hot trend
- Digging deeper
- Frequency analysis
- Activity around the trend
- A table that speaks a 1000 words
- Analyzing used languages
- Final thoughts

## 2. What we are doing here?

The answer to that question is simple and to the point, that is "Social Media Analytics". This term seems big if we explain it in simple word "the art and science of extracting valuable hidden insights from vast amounts of semi-structured and unstructured social media data to enable informed and insightful decision making."

### 2.1 steps involved in social media analytics

There are three main steps in analyzing social media: **data identification, data analysis, and information interpretation**. To maximize the value derived at every point during the process, analysts may define a question to be answered. The important questions for data analysis are: "Who? What? Where? When? Why? and How?" These questions help in determining the proper data sources to evaluate, which can affect the type of analysis that can be performed.



#### 2.1.1 Data identification

Data identification is the process of identifying the subsets of available data to focus on for analysis. Raw data is useful once it is interpreted. After data has been analyzed, it can begin to convey a message. Any data that conveys a meaningful message becomes information. On a high level, unprocessed data takes the following forms to translate into an exact message: noisy data;

relevant and irrelevant data, filtered data; only relevant data, information; data that conveys a vague message, knowledge; data that conveys a precise message, wisdom; data that convey an exact message and reason behind it. To derive wisdom from unprocessed data, we need to start processing it, refine the dataset by including data that we want to focus on and organize data to identify information. In the context of social media analytics, data identification means "what" content is of interest.

### 2.1.2 Data analysis

Data analysis is the set of activities that assist in transforming raw data into insight, which in turn leads to a new base of knowledge and business value. In other words, data analysis is the phase that takes filtered data as input and transforms that into the information of value to the analysts. Many different types of analysis can be performed with social media data, including analysis of posts, sentiment, sentiment drivers, geography, demographics, etc. The data analysis step begins once we know what problem we want to solve and know that we have sufficient data that is enough to generate a meaningful result. How can we know if we have enough evidence to warrant a conclusion? The answer to this question is: we don't know. We can't know this unless we start analyzing the data. While analyzing if we found the data isn't sufficient, reiterate the first phase and modify the question. If the data is believed to be sufficient for analysis, we need to build a data model.

Developing a data model is a process or method that we use to organize data elements and standardize how the individual data elements relate to each other. This step is important because we want to run a computer program over the data; we need a way to tell the computer which words or themes are important and if certain words relate to the topic we are exploring.

### 2.1.3 Information interpretation

The insights derived from the analysis can be as varied as the original question that was posed in step one of the analysis. At this stage, as the non-technical business users are the receivers of the information, the form of presenting the data becomes important. How could the data make sense efficiently so it could be used in good decision making? Visualization (graphics) of the information is the answer to this question.

The best visualizations are ones that expose something new about the underlying patterns and relationships that contain the data. Exposure of the patterns and understating them play a key role in the decision-making process.

Now as we have seen the flow of any social media analytics project now we can define our project in these terms.



### 3 Redefining the whole project

Here we are defining the project based on the flow chart given earlier. It shows all the steps from getting data to making visualization.

#### 3.1 Define problem

What are the trends taking place on Twitter in the US and worldwide and find if there are some trends which are common in both of the lists of trends

Furthermore, we want to find dig deeper into one of the trends named **"WeLoveTheEarth"** what is it about. And then we make a visualization of the language in which the tweets are present.

It helps us to know the most used language for this particular trend is

#### 3.2 Locate Datasource

As for this project, we are talking about tweets hence our data also comes from a reputed social media Twitter.

#### 3.3 Search Datasource and create the Data model

The Data source may give us unrefined unreliable data so we have to refine it in a way that we can get the data we need in a separate file such as US\_trends.csv, WW\_trends etc.

#### 3.4 Run analytics and get insights

Load required data to program and perform some operations on data to get required metrics and required visualization from it

### 4 DataSource and Data

#### 4.1 Why analyze Twitter Data?

- Twitter is one of the largest social networking sites in operation right now while Twitter is far from the comprehensive record of the public conversation it can provide insights into popular trends and important cultural and political movements
- Can be used to measure public opinion on important political or social topics
- Used to analyze political polarization and spread of protest movements
- One can collect only 1% of the total data for free from Twitter
- A tweet contains:-
  - Text
  - User Profile information
  - Geolocation
  - Retweets and Quoted Tweets

#### 4.2 Collecting data through Twitter API

- API: Application Programming Interface
- It is the method of accessing data from a business or government organization
- Twitter have multiple API which can be used for different purposes
  - Search API:- allows access to tweets from the past week
  - Ads API:- Focuses on Twitter Ads
  - Streaming API:- gives real-time data

### 4.3 Streaming API

- This API is used to collect data for this project
- Allows us to collect a sample of tweets in real-time based on keywords, User ID, location
- The connection stays open until you close it
- It has 2 endpoints
  - Sample
    - Gives a random 1% sample of all of the Twitter data
  - Filter
    - You can request a few hundred keywords, few thousand usernames and 25 location ranges

### 4.4 Tweepy: Module for Accessing Twitter API

- Tweepy abstracts away much of the work we need to set up a stable Twitter streaming API Connection
- While we are going to do this we have to set our Twitter account and API keys for authentication
- It provides us with a JSON file
- JSON is a special format that is both human-readable and is easily transferred between machines.
- The JSON file is a lot like python and composed of a combination of dictionaries and lists.

## 5 Languages and Software and packages used

### 5.1 Python

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable it uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is interpreted. Python is processed at run time by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is interactive. You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented. Python supports an object-oriented style or technique of programming that encapsulates code within objects.
- Python is a beginner language. Python is a great language for a beginner-level programmer and supports the development of a wide range of applications

### 5.2 Jupyter Notebook (IDE)

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially refer to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list

of input/output cells which can contain code, text, mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to several open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command-line interface in a shell. To simplify the visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

## 5.3 Packages

### 5.3.1 Numpy

NumPy is a Python package. It stands for 'Numerical Python '. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created the NumPy package by incorporating the features of Numarray into the Numeric package. There are many contributors to this open-source project.

#### 5.3.1.1 Operations using NumPy

Using NumPy, a developer can perform the following operations –

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

#### 5.3.1.2 Why Use NumPy?

In Python, we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

### 5.3.2 Pandas

Pandas is an open-source Python library providing high-performance data manipulation and analysis tools using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Before Pandas, Python was majorly used for data munging and preparation. It had very little contribution to data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, statistics, analytics, etc.

#### 5.3.2.1 Key Features of Pandas

- Fast and efficient DataFrame object with the default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Label-based slicing, indexing, and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High-performance merging and joining of data.
- Time Series functionality.

#### 5.3.3 Matplotlib

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPython or Tkinter. It can be used in Python and IPython shells, Jupyter notebook, and web application servers also.

Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with NumPy can be considered as the open-source equivalent of MATLAB.

Matplotlib was originally written by John D. Hunter in 2003. The current stable version is 2.2.0 released in January 2018.

#### 5.3.3 Seaborn

In the world of Analytics, the best way to get insights is by visualizing the data. Data can be visualized by representing it as plots that are easy to understand, explore, and grasp. Such data helps in drawing the attention of key elements.

To analyze a set of data using Python, we make use of Matplotlib, a widely implemented 2D plotting library. Likewise, Seaborn is a visualization library in Python. It is built on top of Matplotlib.

### 5.3.3.2 Important Features of Seaborn

Seaborn is built on top of Python's core visualization library Matplotlib. It is meant to serve as a compliment and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in –

- Built-in themes for styling matplotlib graphics
- Visualizing univariate and bivariate data
- Fitting in and visualizing linear regression models
- Plotting statistical time-series data
- Seaborn works well with NumPy and Pandas data structures
- It comes with built-in themes for styling Matplotlib graphics

## 6 Information and data we have before starting




Now it's time for the real action. Total of 10 tasks assigned.

### 6.1 Tasks list

We are provided with datasets on which analysis is to be performed along with a jupyter notebook containing some initial steps and also provided with precise instruction to do the tasks. First, let go through the tasks we have to perform on the dataset in the jupyter notebook

1. Local and global thought patterns
2. Prettifying the output
3. Finding common trends
4. Exploring the hot trend
5. Digging deeper
6. Frequency analysis
7. Activity around the trend
8. A table that speaks a 1000 words
9. Analyzing used languages

### 6.2 Datasets

| Name  | Date modified       | Type      | Size     |
|---|---------------------|-----------|----------|
|  USTrends.json       | 21-03-2021 11:13 PM | JSON File | 8 KB     |
|  WeLoveTheEarth.json | 21-03-2021 11:13 PM | JSON File | 3,174 KB |
|  WWTrends.json       | 21-03-2021 11:13 PM | JSON File | 9 KB     |

We are provided with three files

#### 1. USTrends.json:-

contains information about trends in which are going on in the United States.

2. WWTrends.json:-  
contains information about trends which are going on WorldWide.
3. WeLoveTheEarth.json:-  
contains information about a particular “WeLoveTheEarth” trend.

### 6.3 Format of provided data(.json)

JSON (JavaScript Object Notation) is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value). It is a very common data format, with a diverse range of applications, such as serving as a replacement for XML in AJAX systems.

JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is application/json . JSON filenames use the extension .json.

Douglas Crockford originally specified the JSON format in the early 2000s. After RFC 4627 had been available as its "informational" specification since 2006, JSON was first standardized in 2013, as ECMA-404. RFC 8259, published in 2017, is the current version of the Internet Standard STD 90, and it remains consistent with ECMA-404. That same year, JSON was also standardized as ISO/IEC 21778:2017. The ECMA and ISO standards describe only the allowed syntax, whereas the RFC covers some security and interoperability considerations.

## 7 Project

### 7.1 Task 1: Local and global thought patterns

While we might not be Twitter fans, we have to admit that it has a huge influence on the world (who doesn't know about Trump's tweets). Twitter data is not only gold in terms of insights, but ***Twitter-storms are available for analysis in near real-time***. This means we can learn about the big waves of thoughts and moods around the world as they arise. Twitter provides both global and local trends. Let's load and inspect data for topics that were hot worldwide (WW) and in the United States (US) at the moment of query

Initial code given

```
In [0]: # Loading json module
# ... YOUR CODE FOR TASK 1 ...

# Load WW_trends and US_trends data into the the given variables respectively
WW_trends = ...
US_trends = ...

# Inspecting data by printing out WW_trends and US_trends variables
# ... YOUR CODE FOR TASK 1 ...
```

Steps performed

- Loaded JSON package
- Opened the JSON file using the open() method with 'datasets/WWTrends.json' as an input parameter

- Called the read() method on the opened file to read its content
- Passed the read JSON string to the json.loads() method as input parameter for decoding it
- store the decoded output in WW\_trends

## Final code

```
In [1]: # Loading json module
# ... YOUR CODE FOR TASK 1 ...
import json

WW_trends_json= open("datasets/WWTrends.json").read()
US_trends_json= open("datasets/USTrends.json").read()

# Load WW_trends and US_trends data into the the given variables respectively
WW_trends = json.loads(WW_trends_json)
US_trends = json.loads(US_trends_json)

# Inspecting data by printing out WW_trends and US_trends variables
print(WW_trends)
print(US_trends)
# ... YOUR CODE FOR TASK 1 ...
```

## Method Used

- open()  
The open() function opens a file and returns it as a file object.
- read()  
The read() method returns the specified number of bytes from the file. Default is -1 which means the whole file.
- loads()  
Deserializes ( str, bytes or bytearray instance containing a JSON document) to a Python object using this conversion table.

## Output

```
{'trends': [{'name': '#BeratKandili', 'url': 'http://twitter.com/search?q=#BeratKandili', 'promoted_content': None, 'query': '#BeratKandili', 'tweet_volume': 46373}, {'name': '#GoodFriday', 'url': 'http://twitter.com/search?q=#GoodFriday', 'promoted_content': None, 'query': '#GoodFriday', 'tweet_volume': 81891}, {'name': '#WeLoveTheEarth', 'url': 'http://twitter.com/search?q=#WeLoveTheEarth', 'promoted_content': None, 'query': '#WeLoveTheEarth', 'tweet_volume': 159698}, {'name': '#195TLdenTTVerilir', 'url': 'http://twitter.com/search?q=#195TLdenTTVerilir', 'promoted_content': None, 'query': '#195TLdenTTVerilir', 'tweet_volume': None}, {'name': '#AFLNorthDons', 'url': 'http://twitter.com/search?q=#AFLNorthDons', 'promoted_content': None, 'query': '#AFLNorthDons', 'tweet_volume': None}, {'name': 'Shiv Sena', 'url': 'http://twitter.com/search?q=Shiv+Sena', 'promoted_content': None, 'query': 'Shiv+Sena', 'tweet_volume': None}, {'name': 'Lyra McKee', 'url': 'http://twitter.com/search?q=Lyra+McKee', 'promoted_content': None, 'query': 'Lyra+McKee', 'tweet_volume': 17606}, {'name': 'PriyankaChaturvedi', 'url': 'http://twitter.com/search?q=Priyanka+Chaturvedi', 'promoted_content': None, 'query': 'Priyanka+Chaturvedi', 'tweet_volume': 22342}, {'name': 'Derry', 'url': 'http://twitter.com/search?q=Derry', 'promoted_content': None, 'query': 'Derry', 'tweet_volume': 28234}, {'name': '池袋の事故', 'url': 'http://twitter.com/search?q=池袋の事故', 'promoted_content': None, 'query': '池袋の事故', 'tweet_volume': 34381}, {'name': 'プリウス', 'url': 'http://twitter.com/search?q=プリウス', 'promoted_content': None, 'query': 'プリウス', 'tweet_volume': 22944}, {'name': 'Hemant Karkare', 'url': 'http://twitter.com/search?q=Hemant+Karkare', 'promoted_content': None, 'query': 'Hemant+Karkare', 'tweet_volume': 24067}, {'name': '高齢者', 'url': 'http://twitter.com/search?q=高齢者', 'promoted_content': None, 'query': '高齢者', 'tweet_volume': 28382}, {'name': '브이알', 'url': 'http://twitter.com/search?q=브이알', 'promoted_content': None, 'query': '브이알', 'tweet_volume': 15490}, {'name': '刀ステ', 'url': 'http://twitter.com/search?q=刀ステ', 'promoted_content': None, 'query': '刀ステ', 'tweet_volume': None}, {'name': '免許返納', 'url': 'http://twitter.com/search?q=免許返納', 'promoted_content': None, 'query': '免許返納', 'tweet_volume': None}, {'name': 'Berat Kandilimiz', 'url': 'http://twitter.com/search?q=Berat+Kandilimiz', 'promoted_content': None, 'query': 'Berat+Kandilimiz', 'tweet_volume': 10901}, {'name': 'örgütdeğil arkadaşgrubu', 'url': 'http://twitter.com/search?q=örgütdeğil+arkadaşgrubu', 'promoted_content': None, 'query': 'örgütdeğil+arkadaşgrubu', 'tweet_volume': None}, {'name': 'グレア', 'url': 'http://twitter.com/search?q=グレア', 'promoted_content': None, 'query': 'グレア', 'tweet_volume': 23485}, {'name': '東京・池袋衝突事故', 'url': 'http://twitter.com/search?q=東京・池袋衝突事故', 'promoted_content': None, 'query': '東京・池袋衝突事故', 'tweet_volume': None}, {'name': '重体の女性と女兒', 'url': 'http://twitter.com/search?q=重体の女性と女兒', 'promoted_content': None, 'query': '重体の女性と女兒', 'tweet_volume': None}]}]
```



'http://twitter.com/search?q=重体の女性と女兒','promoted\_content': None, 'query': '重体の女性と女兒', 'tweet\_volume': None},  
 {'name': 'Lil Dicky', 'url': 'http://twitter.com/search?q=Lil+Dicky', 'promoted\_content': None, 'query': 'Lil+Dicky', 'tweet\_volume': 42461},  
 {'name': '歩行者', 'url': 'http://twitter.com/search?q=歩行者', 'promoted\_content': None, 'query': '歩行者', 'tweet\_volume': 25405},  
 {'name': 'Derrick White', 'url': 'http://twitter.com/search?q=Derrick+White', 'promoted\_content': None, 'query': 'Derrick+White', 'tweet\_volume': 27104},  
 {'name': '十二国記', 'url': 'http://twitter.com/search?q=十二国記', 'promoted\_content': None, 'query': '十二国記', 'tweet\_volume': 46803},  
 {'name': '#KpuJanganCurang', 'url': 'http://twitter.com/search?q=#KpuJanganCurang', 'promoted\_content': None, 'query': '#KpuJanganCurang', 'tweet\_volume': 75384},  
 {'name': '#HayırlıCumalar', 'url': 'http://twitter.com/search?q=#HayırlıCumalar', 'promoted\_content': None, 'query': '#HayırlıCumalar', 'tweet\_volume': 19848},  
 {'name': '#HayırlıKandiller', 'url': 'http://twitter.com/search?q=#HayırlıKandiller', 'promoted\_content': None, 'query': '#HayırlıKandiller', 'tweet\_volume': None},  
 {'name': '#HanumanJayanti', 'url': 'http://twitter.com/search?q=#HanumanJayanti', 'promoted\_content': None, 'query': '#HanumanJayanti', 'tweet\_volume': 83138},  
 {'name': '#IndonesianElectionHeroes', 'url': 'http://twitter.com/search?q=#IndonesianElectionHeroes', 'promoted\_content': None, 'query': '#IndonesianElectionHeroes', 'tweet\_volume': 19664},  
 {'name': '#يوم\_الجمعة', 'url': 'http://twitter.com/search?q=#يوم\_الجمعة', 'promoted\_content': None, 'query': '#يوم\_الجمعة', 'tweet\_volume': 80799},  
 {'name': '#NRLBulldogsSouths', 'url': 'http://twitter.com/search?q=#NRLBulldogsSouths', 'promoted\_content': None, 'query': '#NRLBulldogsSouths', 'tweet\_volume': None},  
 {'name': '#NikahUmurBerapa', 'url': 'http://twitter.com/search?q=#NikahUmurBerapa', 'promoted\_content': None, 'query': '#NikahUmurBerapa', 'tweet\_volume': None},  
 {'name': '#DragRace', 'url': 'http://twitter.com/search?q=#DragRace', 'promoted\_content': None, 'query': '#DragRace', 'tweet\_volume': 37166},  
 {'name': '#ViernesSanto', 'url': 'http://twitter.com/search?q=#ViernesSanto', 'promoted\_content': None, 'query': '#ViernesSanto', 'tweet\_volume': None},  
 {'name': '#HardikPatel', 'url': 'http://twitter.com/search?q=#HardikPatel', 'promoted\_content': None, 'query': '#HardikPatel', 'tweet\_volume': None},  
 {'name': '#BLACKPINKxCorden', 'url': 'http://twitter.com/search?q=#BLACKPINKxCorden', 'promoted\_content': None, 'query': '#BLACKPINKxCorden', 'tweet\_volume': 253605},  
 {'name': '#Ontas', 'url': 'http://twitter.com/search?q=#Ontas', 'promoted\_content': None, 'query': '#Ontas', 'tweet\_volume': 27924},  
 {'name': '#ConCalmaRemix', 'url': 'http://twitter.com/search?q=#ConCalmaRemix', 'promoted\_content': None, 'query': '#ConCalmaRemix', 'tweet\_volume': 37846},  
 {'name': '#ProtestoEdiyorum', 'url': 'http://twitter.com/search?q=#ProtestoEdiyorum', 'promoted\_content': None, 'query': '#ProtestoEdiyorum', 'tweet\_volume': None},  
 {'name': '#DinahJane1', 'url': 'http://twitter.com/search?q=#DinahJane1', 'promoted\_content': None, 'query': '#DinahJane1', 'tweet\_volume': 23757},  
 {'name': '#ShivSena', 'url': 'http://twitter.com/search?q=#ShivSena', 'promoted\_content': None, 'query': '#ShivSena', 'tweet\_volume': None},  
 {'name': '#DuyguAsena', 'url': 'http://twitter.com/search?q=#DuyguAsena', 'promoted\_content': None, 'query': '#DuyguAsena', 'tweet\_volume': None},  
 {'name': '#TheJudasInMyLife', 'url': 'http://twitter.com/search?q=#TheJudasInMyLife', 'promoted\_content': None, 'query': '#TheJudasInMyLife', 'tweet\_volume': None},  
 {'name': '#Jersey', 'url': 'http://twitter.com/search?q=#Jersey', 'promoted\_content': None, 'query': '#Jersey', 'tweet\_volume': 20509},  
 {'name': '#اغلاق\_BBM', 'url': 'http://twitter.com/search?q=#اغلاق\_BBM', 'promoted\_content': None, 'query': '#اغلاق\_BBM', 'tweet\_volume': 17055},  
 {'name': '#19aprile', 'url': 'http://twitter.com/search?q=#19aprile', 'promoted\_content': None, 'query': '#19aprile', 'tweet\_volume': None},  
 {'name': '#CHlvLIO', 'url': 'http://twitter.com/search?q=#CHlvLIO', 'promoted\_content': None, 'query': '#CHlvLIO', 'tweet\_volume': None},  
 {'name': '#Karfreitag', 'url': 'http://twitter.com/search?q=#Karfreitag', 'promoted\_content': None, 'query': '#Karfreitag', 'tweet\_volume': None},  
 {'name': '#JunquerasACN', 'url': 'http://twitter.com/search?q=#JunquerasACN', 'promoted\_content': None, 'query': '#JunquerasACN', 'tweet\_volume': None},  
 {'name': 'asACN', 'url': 'http://twitter.com/search?q=asACN', 'promoted\_content': None, 'query': 'asACN', 'tweet\_volume': None},  
 {'name': 'Worldwide', 'url': 'http://twitter.com/search?q=Worldwide', 'promoted\_content': None, 'query': 'Worldwide', 'tweet\_volume': None},  
 {'name': 'weoid', 'url': 'http://twitter.com/search?q=weoid', 'promoted\_content': None, 'query': 'weoid', 'tweet\_volume': None},  
 {'name': 'trends', 'url': 'http://twitter.com/search?q=trends', 'promoted\_content': None, 'query': 'trends', 'tweet\_volume': None},  
 {'name': '#WeLoveTheEarth', 'url': 'http://twitter.com/search?q=#WeLoveTheEarth', 'promoted\_content': None, 'query': '#WeLoveTheEarth', 'tweet\_volume': 159698},  
 {'name': '#DragRace', 'url': 'http://twitter.com/search?q=#DragRace', 'promoted\_content': None, 'query': '#DragRace', 'tweet\_volume': 37166},  
 {'name': 'Lil Dicky', 'url': 'http://twitter.com/search?q=Lil+Dicky', 'promoted\_content': None, 'query': 'Lil+Dicky', 'tweet\_volume': 42461},  
 {'name': 'Derrick White', 'url': 'http://twitter.com/search?q=Derrick+White', 'promoted\_content': None, 'query': 'Derrick+White', 'tweet\_volume': 27104},  
 {'name': '#CUZILOVEYOU', 'url': 'http://twitter.com/search?q=#CUZILOVEYOU', 'promoted\_content': None, 'query': '#CUZILOVEYOU', 'tweet\_volume': None},  
 {'name': 'Kevin Durant', 'url': 'http://twitter.com/search?q=Kevin+Durant', 'promoted\_content': None, 'query': 'Kevin+Durant', 'tweet\_volume': 21870},  
 {'name': '#StarTrekDiscovery', 'url': 'http://twitter.com/search?q=#StarTrekDiscovery', 'promoted\_content': None, 'query': '#StarTrekDiscovery', 'tweet\_volume': None},  
 {'name': '#GSWvsLAC', 'url': 'http://twitter.com/search?q=#GSWvsLAC', 'promoted\_content': None, 'query': '#GSWvsLAC', 'tweet\_volume': None},  
 {'name': 'Oshie', 'url': 'http://twitter.com/search?q=Oshie', 'promoted\_content': None, 'query': 'Oshie', 'tweet\_volume': None},  
 {'name': 'Seth Abramson', 'url': 'http://twitter.com/search?q=Seth+Abramson', 'promoted\_content': None, 'query': 'Seth+Abramson', 'tweet\_volume': None},  
 {'name': 'Lyra McKee', 'url': 'http://twitter.com/search?q=Lyra+McKee', 'promoted\_content': None, 'query': 'Lyra+McKee', 'tweet\_volume': None},  
 {'name': 'Silky', 'url': 'http://twitter.com/search?q=Silky', 'promoted\_content': None, 'query': 'Silky', 'tweet\_volume': 12881},  
 {'name': 'Kazuo Koike', 'url': 'http://twitter.com/search?q=Kazuo+Koike', 'promoted\_content': None, 'query': 'Kazuo+Koike', 'tweet\_volume': None},  
 {'name': 'Game 6', 'url': 'http://twitter.com/search?q=Game+6', 'promoted\_content': None, 'query': 'Game+6', 'tweet\_volume': None},  
 {'name': 'Yvie', 'url': 'http://twitter.com/search?q=Yvie', 'promoted\_content': None, 'query': 'Yvie', 'tweet\_volume': 10680},  
 {'name': 'Gallant', 'url': 'http://twitter.com/search?q=Gallant', 'promoted\_content': None, 'query': 'Gallant', 'tweet\_volume': None},  
 {'name': 'Lone Wolf and Cub', 'url': 'http://twitter.com/search?q=Lone+Wolf+and+Cub', 'promoted\_content': None, 'query': 'Lone+Wolf+and+Cub', 'tweet\_volume': None},  
 {'name': 'George Conway', 'url': 'http://twitter.com/search?q=George+Conway', 'promoted\_content': None, 'query': 'George+Conway', 'tweet\_volume': 27458},  
 {'name': 'David Fletcher', 'url': 'http://twitter.com/search?q=David+Fletcher', 'promoted\_content': None, 'query': 'David+Fletcher', 'tweet\_volume': None},  
 {'name': 'Derry', 'url': 'http://twitter.com/search?q=Derry', 'promoted\_content': None, 'query': 'Derry', 'tweet\_volume': 28234},  
 {'name': 'Mike Anderson', 'url': 'http://twitter.com/search?q=Mike+Anderson', 'promoted\_content': None, 'query': 'Mike+Anderson', 'tweet\_volume': None},  
 {'name': 'Shy Glizzy', 'url': 'http://twitter.com/search?q=Shy+Glizzy', 'promoted\_content': None, 'query': 'Shy+Glizzy', 'tweet\_volume': None},  
 {'name': 'Tomas Hertl', 'url': 'http://twitter.com/search?q=Tomas+Hertl', 'promoted\_content': None, 'query': 'Tomas+Hertl', 'tweet\_volume': None},  
 {'name': 'Servais', 'url': 'http://twitter.com/search?q=Servais', 'promoted\_content': None, 'query': 'Servais', 'tweet\_volume': None},  
 {'name': 'WE LOVE THE EARTH', 'url': 'http://twitter.com/search?q=WE+LOVE+THE+EARTH', 'promoted\_content': None, 'query': 'WE+LOVE+THE+EARTH', 'tweet\_volume': 338417},  
 {'name': '#DinahJane1', 'url': 'http://twitter.com/search?q=#DinahJane1', 'promoted\_content': None, 'query': '#DinahJane1', 'tweet\_volume': 23757},  
 {'name': '#WhatStopsYouFromGoingHome', 'url': 'http://twitter.com/search?q=#WhatStopsYouFromGoingHome', 'promoted\_content': None, 'query': '#WhatStopsYouFromGoingHome', 'tweet\_volume': None},  
 {'name': '#MakeAMovieSensual', 'url': 'http://twitter.com/search?q=#MakeAMovieSensual', 'promoted\_content': None, 'query': '#MakeAMovieSensual', 'tweet\_volume': None},  
 {'name': '#DontChangeOutNow', 'url': 'http://twitter.com/search?q=#DontChangeOutNow', 'promoted\_content': None, 'query': '#DontChangeOutNow', 'tweet\_volume': None}



```
{
  'query': '#DontChangeOutNow',
  'tweet_volume': None,
  {'name': '#BLACKPINKxCorden',
   'url': 'http://twitter.com/search?q=#BLACKPINKxCorden',
   'promoted_content': None,
   'query': '#BLACKPINKxCorden',
   'tweet_volume': 253605},
  {'name': '#WorldofWarcraftMains',
   'url': 'http://twitter.com/search?q=#WorldofWarcraftMains',
   'promoted_content': None,
   'query': '#WorldofWarcraftMains',
   'tweet_volume': None},
  {'name': '#MyDrunkUncleSays',
   'url': 'http://twitter.com/search?q=#MyDrunkUncleSays',
   'promoted_content': None,
   'query': '#MyDrunkUncleSays',
   'tweet_volume': None},
  {'name': '#WGAMIX',
   'url': 'http://twitter.com/search?q=#WGAMIX',
   'promoted_content': None,
   'query': '#WGAMIX',
   'tweet_volume': None},
  {'name': '#Earth',
   'url': 'http://twitter.com/search?q=#Earth',
   'promoted_content': None,
   'query': '#Earth',
   'tweet_volume': 13655},
  {'name': '#TheLegendOfVoxMachina',
   'url': 'http://twitter.com/search?q=#TheLegendOfVoxMachina',
   'promoted_content': None,
   'query': '#TheLegendOfVoxMachina',
   'tweet_volume': None},
  {'name': '#AFLNorthDons',
   'url': 'http://twitter.com/search?q=#AFLNorthDons',
   'promoted_content': None,
   'query': '#AFLNorthDons',
   'tweet_volume': None},
  {'name': '#FridayFeeling',
   'url': 'http://twitter.com/search?q=#FridayFeeling',
   'promoted_content': None,
   'query': '#FridayFeeling',
   'tweet_volume': 19510},
  {'name': '#MyInnerDemonSaid',
   'url': 'http://twitter.com/search?q=#MyInnerDemonSaid',
   'promoted_content': None,
   'query': '#MyInnerDemonSaid',
   'tweet_volume': None},
  {'name': '#rupaulsdragrace',
   'url': 'http://twitter.com/search?q=#rupaulsdragrace',
   'promoted_content': None,
   'query': '#rupaulsdragrace',
   'tweet_volume': None},
  {'name': '#ConCalmaRemix',
   'url': 'http://twitter.com/search?q=#ConCalmaRemix',
   'promoted_content': None,
   'query': '#ConCalmaRemix',
   'tweet_volume': 37846},
  {'name': '#TimeToImpeach',
   'url': 'http://twitter.com/search?q=#TimeToImpeach',
   'promoted_content': None,
   'query': '#TimeToImpeach',
   'tweet_volume': 21732},
  {'name': '#NRLBulldogsSouths',
   'url': 'http://twitter.com/search?q=#NRLBulldogsSouths',
   'promoted_content': None,
   'query': '#NRLBulldogsSouths',
   'tweet_volume': None},
  {'name': '#CriticalRoleSpoilers',
   'url': 'http://twitter.com/search?q=#CriticalRoleSpoilers',
   'promoted_content': None,
   'query': '#CriticalRoleSpoilers',
   'tweet_volume': None},
  {'name': '#GossipShouldBe',
   'url': 'http://twitter.com/search?q=#GossipShouldBe',
   'promoted_content': None,
   'query': '#GossipShouldBe',
   'tweet_volume': None},
  {'name': '#LilDicky',
   'url': 'http://twitter.com/search?q=#LilDicky',
   'promoted_content': None,
   'query': '#LilDicky',
   'tweet_volume': None},
  {'name': '#RPDR',
   'url': 'http://twitter.com/search?q=#RPDR',
   'promoted_content': None,
   'query': '#RPDR',
   'tweet_volume': None},
  {'name': '#WeirdDateStories',
   'url': 'http://twitter.com/search?q=#WeirdDateStories',
   'promoted_content': None,
   'query': '#WeirdDateStories',
   'tweet_volume': None},
  {'name': '#HustleAndSoul',
   'url': 'http://twitter.com/search?q=#HustleAndSoul',
   'promoted_content': None,
   'query': '#HustleAndSoul',
   'tweet_volume': None},
  {'name': '#fridaymotivation',
   'url': 'http://twitter.com/search?q=#fridaymotivation',
   'promoted_content': None,
   'query': '#fridaymotivation',
   'tweet_volume': None}],
  'as_of': '2019-04-19T08:43:43Z',
  'created_at': '2019-04-19T08:39:15Z',
  'locations': [{'name': 'United States', 'woeid': 23424977}]}

```

## 7.2 Task 2: Prettifying the output

Our data was hard to read! Luckily, we can resort to *json.dumps()* method to have it formatted as a pretty JSON string.

Initial code given

```
In [0]: # Pretty-printing the results. First WW and then US trends.

print("WW trends:")
# ... YOUR CODE FOR TASK 2 ...

print("\n", "US trends:")
# ... YOUR CODE FOR TASK 2 ...
```

### Steps performed

- Passed the WW\_trends object to the json.dumps() method, with an additional input parameter indent set to 1. printed the output
- Passed the US\_trends object to the json.dumps() method, with an additional input parameter indent set to 1. printed the output

### Method Used

- dumps()
  - formats data as a JSON string. If you pass 'indent' to the method (a positive integer), then all the elements in the JSON array are printed with that indent level. This makes it easy to read the results

### Final Code

```
In [2]: # Pretty-printing the results. First WW and then US trends.
```

```
print("WW trends:")
# ... YOUR CODE FOR TASK 2 ...
print(json.dumps(WW_trends,indent=1))

print("\n", "US trends:")
# ... YOUR CODE FOR TASK 2 ...
print(json.dumps(US_trends,indent=1))
```

## Output

```
[
  {
    "trends": [
      {
        "name": "#BeratKandili",
        "url": "http://twitter.com/search?q=#BeratKandili",
        "promoted_content": null,
        "query": "#BeratKandili",
        "tweet_volume": 46373
      },
      {
        "name": "#GoodFriday",
        "url": "http://twitter.com/search?q=#GoodFriday",
        "promoted_content": null,
        "query": "#GoodFriday",
        "tweet_volume": 81891
      },
      {
        "name": "#WeLoveTheEarth",
        "url": "http://twitter.com/search?q=#WeLoveTheEarth",
        "promoted_content": null,
        "query": "#WeLoveTheEarth",
        "tweet_volume": 159698
      },
      {
        "name": "#195TLdenTTVerilir",
        "url": "http://twitter.com/search?q=#195TLdenTTVerilir",
        "promoted_content": null,
        "query": "#195TLdenTTVerilir",
        "tweet_volume": null
      },
      {
        "name": "#AFLNorthDons",
        "url": "http://twitter.com/search?q=#AFLNorthDons",
        "promoted_content": null,
        "query": "#AFLNorthDons",
        "tweet_volume": null
      },
      {
        "name": "Shiv Sena",
        "url": "http://twitter.com/search?q=Shiv+Sena",
        "promoted_content": null,
        "query": "Shiv+Sena",
        "tweet_volume": null
      },
      {
        "name": "Lyra McKee",
        "url": "http://twitter.com/search?q=Lyra+McKee",
        "promoted_content": null,
        "query": "Lyra+McKee",
        "tweet_volume": 17606
      },
      {
        "name": "Priyanka Chaturvedi",
        "url": "http://twitter.com/search?q=Priyanka+Chaturvedi",
        "promoted_content": null,
        "query": "Priyanka+Chaturvedi",
        "tweet_volume": 22342
      },
      {
        "name": "Derry",
        "url": "http://twitter.com/search?q=Derry",
        "promoted_content": null,
        "query": "Derry",
        "tweet_volume": 28234
      },
      {
        "name": "\u06c6\u0888b\u0306e\u04e8b\u06545",
        "url": "http://twitter.com/search?q=\u030d7\u030ea\u030a6\u030b9",
        "promoted_content": null,
        "query": "\u030d7\u030ea\u030a6\u030b9",
        "tweet_volume": 34381
      },
      {
        "name": "\u030d7\u030ea\u030a6\u030b9",
        "url": "http://twitter.com/search?q=\u030d7\u030ea\u030a6\u030b9",
        "promoted_content": null,
        "query": "\u030d7\u030ea\u030a6\u030b9",
        "tweet_volume": 22944
      },
      {
        "name": "Hemant Karkare",
        "url": "http://twitter.com/search?q=Hemant+Karkare",
        "promoted_content": null,
        "query": "Hemant+Karkare",
        "tweet_volume": 24067
      },
      {
        "name": "\u09ad8\u09f62\u08005",
        "url": "http://twitter.com/search?q=\u09ad8\u09f62\u08005",
        "promoted_content": null,
        "query": "\u09ad8\u09f62\u08005",
        "tweet_volume": 28382
      },
      {
        "name": "\u06c6\u0888b\u0306e\u04e8b\u06545",
        "url": "http://twitter.com/search?q=\u06c6\u0888b\u0306e\u04e8b\u06545",
        "promoted_content": null,
        "query": "\u06c6\u0888b\u0306e\u04e8b\u06545",
        "tweet_volume": 15490
      },
      {
        "name": "\u06c6\u0888b\u0306e\u04e8b\u06545",
        "url": "http://twitter.com/search?q=\u06c6\u0888b\u0306e\u04e8b\u06545",
        "promoted_content": null,
        "query": "\u06c6\u0888b\u0306e\u04e8b\u06545",
        "tweet_volume": null
      },
      {
        "name": "Berat Kandilimiz",
        "url": "http://twitter.com/search?q=Berat+Kandilimiz",
        "promoted_content": null,
        "query": "Berat+Kandilimiz",
        "tweet_volume": 10901
      },
      {
        "name": "\u00f6rg\u00fctde\u011fil+arkada\u015fgrubu",
        "url": "http://twitter.com/search?q=\u00f6rg\u00fctde\u011fil+arkada\u015fgrubu",
        "promoted_content": null,
        "query": "\u00f6rg\u00fctde\u011fil+arkada\u015fgrubu",
        "tweet_volume": null
      },
      {
        "name": "\u030b0\u030ec\u030a2",
        "url": "http://twitter.com/search?q=\u030b0\u030ec\u030a2",
        "promoted_content": null,
        "query": "\u030b0\u030ec\u030a2",
        "tweet_volume": 23485
      }
    ]
  }
]
```

```

"name":
"\u6771\u4eac\u30fb\u6c60\u88b\u885d\u7a81\u4e8b\u6545",
"url": "http://twitter.com/search?q=
東京・池袋衝突事故",
"promoted_content": null,
"query": "東京・池袋衝突事故",
"tweet_volume": null
},
{
"name":
"\u91cd\u4f53\u306e\u5973\u6027\u3068\u5973\u5b69",
"url": "http://twitter.com/search?q=
重体の女性と女児",
"promoted_content": null,
"query": "重体の女性と女児",
"tweet_volume": null
},
{
"name": "Lil Dicky",
"url":
"http://twitter.com/search?q=Lil+Dicky",
"promoted_content": null,
"query": "Lil+Dicky",
"tweet_volume": 42461
},
{
"name": "\u6b69\u884c\u8005",
"url": "http://twitter.com/search?q=
歩行者",
"promoted_content": null,
"query": "歩行者",
"tweet_volume": 25405
},
{
"name": "Derrick White",
"url":
"http://twitter.com/search?q=Derrick+White",
"promoted_content": null,
"query": "Derrick+White",
"tweet_volume": 27104
},
{
"name":
"\u5341\u4e8c\u56fd\u8a18",
"url": "http://twitter.com/search?q=
十二国記",
"promoted_content": null,
"query": "十二国記",
"tweet_volume": 46803
},
{
"name": "#KpuJanganCurang",
"url":
"http://twitter.com/search?q=#KpuJ
anganCurang",
"promoted_content": null,
"query": "#KpuJanganCurang",
"tweet_volume": 75384
},
{
"name":
"#Hay\u0131rl\u0131Cumalar",
"url":
"http://twitter.com/search?q=#HayırlıCumalar",
"promoted_content": null,
"query": "#HayırlıCumalar",
"tweet_volume": 19848
},
{
"name":
"#Hay\u0131rl\u0131Kandiller",
"url":
"http://twitter.com/search?q=#HayırlıKandiller",
"promoted_content": null,
"query": "#HayırlıKandiller",
"tweet_volume": null
},
{
"name": "#HanumanJayanti",
"url":
"http://twitter.com/search?q=#HanumanJayanti",
"promoted_content": null,
"query": "#HanumanJayanti",
"tweet_volume": 83138
},
{
"name": "#IndonesianElectionHeroes",
"url":
"http://twitter.com/search?q=#IndonesianElectionHeroes",
"promoted_content": null,
"query": "#IndonesianElectionHeroes",
"tweet_volume": 19664
},
{
"name":
"#\u064a\u0648\u0645\u0627\u0644\u062c\u0645\u0639",
"url": "http://twitter.com/search?q=#
يوم_الجمعة",
"promoted_content": null,
"query": "#يوم_الجمعة",
"tweet_volume": 80799
},
{
"name": "#NRLBulldogsSouths",
"url":
"http://twitter.com/search?q=#NRLBulldogsSouths",
"promoted_content": null,
"query": "#NRLBulldogsSouths",
"tweet_volume": null
},
{
"name": "#NikahUmurBerapa",
"url":
"http://twitter.com/search?q=#NikahUmurBerapa",
"promoted_content": null,
"query": "#NikahUmurBerapa",
"tweet_volume": null
},
{
"name": "#DragRace",
"url":
"http://twitter.com/search?q=#DragRace",
"promoted_content": null,
"query": "#DragRace",
"tweet_volume": 37166
},
{
"name": "#ViernesSanto",
"url":
"http://twitter.com/search?q=#ViernesSanto",
"promoted_content": null,
"query": "#ViernesSanto",
"tweet_volume": null
},
},
{
"name": "#HardikPatel",
"url":
"http://twitter.com/search?q=#HardikPatel",
"promoted_content": null,
"query": "#HardikPatel",
"tweet_volume": null
},
{
"name": "#BLACKPINKxCorden",
"url":
"http://twitter.com/search?q=#BLACKPINKxCorden",
"promoted_content": null,
"query": "#BLACKPINKxCorden",
"tweet_volume": 253605
},
{
"name": "#Ontas",
"url":
"http://twitter.com/search?q=#Ontas",
"promoted_content": null,
"query": "#Ontas",
"tweet_volume": 27924
},
{
"name": "#ConCalmaRemix",
"url":
"http://twitter.com/search?q=#ConCalmaRemix",
"promoted_content": null,
"query": "#ConCalmaRemix",
"tweet_volume": 37846
},
{
"name": "#ProtestoEdiyorum",
"url":
"http://twitter.com/search?q=#ProtestoEdiyorum",
"promoted_content": null,
"query": "#ProtestoEdiyorum",
"tweet_volume": null
},
{
"name": "#DinahJane1",
"url":
"http://twitter.com/search?q=#DinahJane1",
"promoted_content": null,
"query": "#DinahJane1",
"tweet_volume": 23757
},
{
"name": "#ShivSena",
"url":
"http://twitter.com/search?q=#ShivSena",
"promoted_content": null,
"query": "#ShivSena",
"tweet_volume": null
},
{
"name": "#DuyguAsena",
"url":
"http://twitter.com/search?q=#DuyguAsena",
"promoted_content": null,
"query": "#DuyguAsena",
"tweet_volume": null
},
{
"name": "#TheJudasInMyLife",

```

```

"url":
"http://twitter.com/search?q=#TheJ
udasInMyLife",
"promoted_content": null,
"query": "#TheJudasInMyLife",
"tweet_volume": null
},
{
"name": "#Jersey",
"url":
"http://twitter.com/search?q=#Jerse
y",
"promoted_content": null,
"query": "#Jersey",
"tweet_volume": 20509
},
{
"name":
"#\u0627\u063a\u0644\u0627\u0642
2_BBM",
"url": "http://twitter.com/search?q=#
اغلاق_BBM",
"promoted_content": null,
"query": "#اغلاق_BBM",
"tweet_volume": 17055
},
{
"name": "#19aprile",
"url":
"http://twitter.com/search?q=#19apr
ile",
"promoted_content": null,
"query": "#19aprile",
"tweet_volume": null
},
{
"name": "#CHlvLIO",
"url":
"http://twitter.com/search?q=#CHlv
LIO",
"promoted_content": null,
"query": "#CHlvLIO",
"tweet_volume": null
},
{
"name": "#Karfreitag",
"url":
"http://twitter.com/search?q=#Karfr
eitag",
"promoted_content": null,
"query": "#Karfreitag",
"tweet_volume": null
},
{
"name": "#JunquerasACN",
"url":
"http://twitter.com/search?q=#Junq
uerasACN",
"promoted_content": null,
"query": "#JunquerasACN",
"tweet_volume": null
},
{
"as_of": "2019-04-19T08:43:43Z",
"created_at": "2019-04-
19T08:39:15Z",
"locations": [
{
"name": "Worldwide",
"woeid": 1
}
]
}
US trends:

```

```

[
{
"trends": [
{
"name": "#WeLoveTheEarth",
"url":
"http://twitter.com/search?q=#WeLo
veTheEarth",
"promoted_content": null,
"query": "#WeLoveTheEarth",
"tweet_volume": 159698
},
{
"name": "#DragRace",
"url":
"http://twitter.com/search?q=#Drag
Race",
"promoted_content": null,
"query": "#DragRace",
"tweet_volume": 37166
},
{
"name": "Lil Dicky",
"url":
"http://twitter.com/search?q=Lil+Di
cky",
"promoted_content": null,
"query": "Lil+Dicky",
"tweet_volume": 42461
},
{
"name": "Derrick White",
"url":
"http://twitter.com/search?q=Derric
k+White",
"promoted_content": null,
"query": "Derrick+White",
"tweet_volume": 27104
},
{
"name": "#CUZILOVEYOU",
"url":
"http://twitter.com/search?q=#CUZI
LOVEYOU",
"promoted_content": null,
"query": "#CUZILOVEYOU",
"tweet_volume": null
},
{
"name": "Kevin Durant",
"url":
"http://twitter.com/search?q=Kevin
+Durant",
"promoted_content": null,
"query": "Kevin+Durant",
"tweet_volume": 21870
},
{
"name": "#StarTrekDiscovery",
"url":
"http://twitter.com/search?q=#StarT
rekDiscovery",
"promoted_content": null,
"query": "#StarTrekDiscovery",
"tweet_volume": null
},
{
"name": "#GSWvsLAC",
"url":
"http://twitter.com/search?q=#GSW
vsLAC",
"promoted_content": null,
"query": "#GSWvsLAC",
"tweet_volume": null
},

```

```

{
"name": "Oshie",
"url":
"http://twitter.com/search?q=Oshie"
,
"promoted_content": null,
"query": "Oshie",
"tweet_volume": null
},
{
"name": "Seth Abramson",
"url":
"http://twitter.com/search?q=Seth+
Abramson",
"promoted_content": null,
"query": "Seth+Abramson",
"tweet_volume": null
},
{
"name": "Lyra McKee",
"url":
"http://twitter.com/search?q=Lyra+
McKee",
"promoted_content": null,
"query": "Lyra+McKee",
"tweet_volume": 17606
},
{
"name": "Silky",
"url":
"http://twitter.com/search?q=Silky",
"promoted_content": null,
"query": "Silky",
"tweet_volume": 12881
},
{
"name": "Kazuo Koike",
"url":
"http://twitter.com/search?q=Kazu
o+Koike",
"promoted_content": null,
"query": "Kazuo+Koike",
"tweet_volume": null
},
{
"name": "Game 6",
"url":
"http://twitter.com/search?q=Game
+6",
"promoted_content": null,
"query": "Game+6",
"tweet_volume": null
},
{
"name": "Yvie",
"url":
"http://twitter.com/search?q=Yvie",
"promoted_content": null,
"query": "Yvie",
"tweet_volume": 10680
},
{
"name": "Gallant",
"url":
"http://twitter.com/search?q=Gallant
",
"promoted_content": null,
"query": "Gallant",
"tweet_volume": null
},
{
"name": "Lone Wolf and Cub",
"url":
"http://twitter.com/search?q=Lone+
Wolf+and+Cub",

```

```

"promoted_content": null,
"query": ""Lone+Wolf+and+Cub"",
"tweet_volume": null
},
{
"name": "George Conway",
"url":
"http://twitter.com/search?q=Georg
e+Conway",
"promoted_content": null,
"query": ""George+Conway"",
"tweet_volume": 27458
},
{
"name": "David Fletcher",
"url":
"http://twitter.com/search?q=David
+Fletcher",
"promoted_content": null,
"query": ""David+Fletcher"",
"tweet_volume": null
},
{
"name": "Derry",
"url":
"http://twitter.com/search?q=Derry",
"promoted_content": null,
"query": "Derry",
"tweet_volume": 28234
},
{
"name": "Mike Anderson",
"url":
"http://twitter.com/search?q=Mike+
Anderson",
"promoted_content": null,
"query": ""Mike+Anderson",
"tweet_volume": null
},
{
"name": "Shy Glizzy",
"url":
"http://twitter.com/search?q=Shy+
Glizzy",
"promoted_content": null,
"query": ""Shy+Glizzy",
"tweet_volume": null
},
{
"name": "Tomas Hertl",
"url":
"http://twitter.com/search?q=Toma
s+Hertl",
"promoted_content": null,
"query": ""Tomas+Hertl",
"tweet_volume": null
},
{
"name": "Servais",
"url":
"http://twitter.com/search?q=Servai
s",
"promoted_content": null,
"query": "Servais",
"tweet_volume": null
},
{
"name": "WE LOVE THE EARTH",
"url":
"http://twitter.com/search?q=WE+L
OVE+THE+EARTH",
"promoted_content": null,
"query":
""WE+LOVE+THE+EARTH",
"tweet_volume": null

```

```

},
{
"name": ""Earth",
"url":
"http://twitter.com/search?q=Earth",
"promoted_content": null,
"query": ""Earth",
"tweet_volume": 338417
},
{
"name": "#DinahJane1",
"url":
"http://twitter.com/search?q=#Dinah
Jane1",
"promoted_content": null,
"query": "#DinahJane1",
"tweet_volume": 23757
},
{
"name":
"#WhatStopsYouFromGoingHome",
"url":
"http://twitter.com/search?q=#What
StopsYouFromGoingHome",
"promoted_content": null,
"query":
"#WhatStopsYouFromGoingHome",
"tweet_volume": null
},
{
"name": "#MakeAMovieSensual",
"url":
"http://twitter.com/search?q=#Make
AMovieSensual",
"promoted_content": null,
"query": "#MakeAMovieSensual",
"tweet_volume": null
},
{
"name": "#DontChangeOutNow",
"url":
"http://twitter.com/search?q=#Dont
ChangeOutNow",
"promoted_content": null,
"query": "#DontChangeOutNow",
"tweet_volume": null
},
{
"name": "#BLACKPINKxCorden",
"url":
"http://twitter.com/search?q=#BLAC
KPINKxCorden",
"promoted_content": null,
"query": "#BLACKPINKxCorden",
"tweet_volume": 253605
},
{
"name": "#WorldofWarcraftMains",
"url":
"http://twitter.com/search?q=#World
ofWarcraftMains",
"promoted_content": null,
"query": "#WorldofWarcraftMains",
"tweet_volume": null
},
{
"name": "#MyDrunkUncleSays",
"url":
"http://twitter.com/search?q=#MyDr
unkUncleSays",
"promoted_content": null,
"query": "#MyDrunkUncleSays",

```

```

"tweet_volume": null
},
{
"name": "#WGAMIX",
"url":
"http://twitter.com/search?q=#WGA
MIX",
"promoted_content": null,
"query": "#WGAMIX",
"tweet_volume": null
},
{
"name": "#Earth",
"url":
"http://twitter.com/search?q=#Earth",
"promoted_content": null,
"query": "#Earth",
"tweet_volume": 13655
},
{
"name":
"#TheLegendOfVoxMachina",
"url":
"http://twitter.com/search?q=#TheL
egendOfVoxMachina",
"promoted_content": null,
"query":
"#TheLegendOfVoxMachina",
"tweet_volume": null
},
{
"name": "#AFLNorthDons",
"url":
"http://twitter.com/search?q=#AFLN
orthDons",
"promoted_content": null,
"query": "#AFLNorthDons",
"tweet_volume": null
},
{
"name": "#FridayFeeling",
"url":
"http://twitter.com/search?q=#Frida
yFeeling",
"promoted_content": null,
"query": "#FridayFeeling",
"tweet_volume": 19510
},
{
"name": "#MyInnerDemonSaid",
"url":
"http://twitter.com/search?q=#MyIn
nerDemonSaid",
"promoted_content": null,
"query": "#MyInnerDemonSaid",
"tweet_volume": null
},
{
"name": "#rupaulsdragrace",
"url":
"http://twitter.com/search?q=#rupau
lsdragrace",
"promoted_content": null,
"query": "#rupaulsdragrace",
"tweet_volume": null
},
{
"name": "#ConCalmaRemix",
"url":
"http://twitter.com/search?q=#ConC
almaRemix",
"promoted_content": null,
"query": "#ConCalmaRemix",
"tweet_volume": 37846

```

```

},
{
  "name": "#TimeTolmpeach",
  "url":
    "http://twitter.com/search?q=#TimeTolmpeach",
  "promoted_content": null,
  "query": "#TimeTolmpeach",
  "tweet_volume": 21732
},
{
  "name": "#NRLBulldogsSouths",
  "url":
    "http://twitter.com/search?q=#NRLBulldogsSouths",
  "promoted_content": null,
  "query": "#NRLBulldogsSouths",
  "tweet_volume": null
},
{
  "name": "#CriticalRoleSpoilers",
  "url":
    "http://twitter.com/search?q=#CriticalRoleSpoilers",
  "promoted_content": null,
  "query": "#CriticalRoleSpoilers",
  "tweet_volume": null
},
{
  "name": "#GossipShouldBe",
  "url":
    "http://twitter.com/search?q=#GossipShouldBe",
  "promoted_content": null,
  "query": "#GossipShouldBe",
  "tweet_volume": null
},
{
  "name": "#LilDicky",
  "url":
    "http://twitter.com/search?q=#LilDicky",
  "promoted_content": null,
  "query": "#LilDicky",
  "tweet_volume": null
},
{
  "name": "#RPDR",
  "url":
    "http://twitter.com/search?q=#RPDR",
  "promoted_content": null,
  "query": "#RPDR",
  "tweet_volume": null
},
{
  "name": "#WeirdDateStories",
  "url":
    "http://twitter.com/search?q=#WeirdDateStories",
  "promoted_content": null,
  "query": "#WeirdDateStories",
  "tweet_volume": null
},
{
  "name": "#HustleAndSoul",
  "url":
    "http://twitter.com/search?q=#HustleAndSoul",
  "promoted_content": null,
  "query": "#HustleAndSoul",
  "tweet_volume": null
},
{
  "name": "#fridaymotivation",
  "url":
    "http://twitter.com/search?q=#fridaymotivation",
  "promoted_content": null,
  "query": "#fridaymotivation",
  "tweet_volume": null
},
{
  "as_of": "2019-04-19T08:43:43Z",
  "created_at": "2019-04-19T08:39:15Z",
  "locations": [
    {
      "name": "United States",
      "woeid": 23424977
    }
  ]
}]

```

## Conclusion

- We have an array of trend objects having: the name of the trending topic, the query parameter that can be used to search for the topic on Twitter-Search, the search URL and the volume of tweets for the last 24 hours, if available. (The trends get updated every 5 mins.)
- At query time **#BeratKandili**, **#GoodFriday** and **#WeLoveTheEarth** were trending WW.
- *"tweet\_volume"* tell us that **#WeLoveTheEarth** was the most popular among the three.
- Results are not sorted by *"tweet\_volume"*.
- Some trends are unique to the US.

## 7.3 Task 3: Finding common trends

It's easy to skim through the two sets of trends and spot common trends, but let's not do "manual" work. We can use Python's **set** data structure to find common trends we can iterate through the two trends objects, cast the lists of names to sets, and call the intersection method to get the common names between the two sets.

### Initial code

```
In [0]: # Extracting all the WW trend names from WW_trends
world_trends = set([... for trend in ...])

# Extracting all the US trend names from US_trends
us_trends = set([... for trend in ...])

# Getting the intersection of the two sets of trends
common_trends = ...

# Inspecting the data
print(world_trends, "\n")
print(us_trends, "\n")
print (len(common_trends), "common trends:", common_trends)
```

## Steps Performed

- Extracted the name field, trend['name'], from the list of trends in WW\_trends and US\_trends using list comprehension. used WW\_trends[0]['trends'] and US\_trends[0]['trends'] for iterations to get the names because the trends objects are lists with only one element.
- Called the intersection() method on world\_trends with us\_trends as an input parameter to get the common items between the two; stored the output in the variable called common\_trends and printed it.

## Final code

```
In [3]: # Extracting all the WW trend names from WW_trends
world_trends = set([trend['name'] for trend in WW_trends[0]['trends']])

# Extracting all the US trend names from US_trends
us_trends = set([trend['name'] for trend in US_trends[0]['trends']])

# Getting the intersection of the two sets of trends
common_trends = world_trends.intersection(us_trends)

# Inspecting the data
print(world_trends, "\n")
print(us_trends, "\n")
print (len(common_trends), "common trends:", common_trends)
```

## Methods Used

- intersection()  
The intersection() method returns a set that contains the similarity between two or more sets. The returned set contains only items that exist in both sets, or all sets if the comparison is done with more than two sets.
- set()  
converts the datatype explicitly to set

## Output

```
{'#GoodFriday', '#ProtestoEdiyorun', 'Berat Kandilimiz', '#اغلاق_BBM', '#195TLdenTTVerilir', '#DragRace', '#JunquerasACN', '歩行者', '#BLACKPINKxCorden', '#Jersey', '#KpuJanganCurang', '#ConCalmaRemix', '#HanumanJayanti', '#WeLoveTheEarth', 'Lyra McKee', 'Priyanka Chaturvedi', '#ViernesSanto', '重体の女性と女兒', '免許返納', '#BeratKandili', 'Lil Dicky', '#HayırlıKandiller', '#19aprire', 'グレア', 'örgütdeğil arkadaşgrubu', '#HayırlıCumalar', '高齢者', '刀ステ', '十二国記', '#Karfreitag', 'プリウス', '#IndonesianElectionHeroes', 'Shiv Sena', '브이알', '#Ontas', '#TheJudasInMyLife', '池袋の事故', '#NRLBulldogsSouths', 'Derry', '#NikahUmurBerapa', '#AFLNorthDons', 'Hemant Karkare', 'Derrick White', '#HardikPatel', '#DinahJane1', '#DuyguAsena', '#ShivSena', '#CHIVLIO', 'يوم_الحم', '東京・池袋衝突事故'}
{'#HustleAndSoul', '#GSWvsLAC', '#LilDicky', '#CriticalRoleSpoilers', 'Kazuo Koike', '#DragRace', 'Lone Wolf and Cub', 'Seth Abramson', '#Earth', 'Oshie', '#BLACKPINKxCorden', '#CUZILOVEYOU', '#StarTrekDiscovery',
```



```
'#WhatStopsYouFromGoingHome', '#ConCalmaRemix', 'Shy Glizzy', '#RPDR', 'George Conway', '#WeLoveTheEarth', 'Lyra McKee', 'Gallant', '#MyInnerDemonSaid', 'Lil Dicky', 'David Fletcher', '#WeirdDateStories', 'Game 6', 'Silky', '#TheLegendOfVoxMachina', 'Servais', '#TimeToImpeach', '#MyDrunkUncleSays', '#MakeAMovieSensual', '#WorldofWarcraftMains', 'Mike Anderson', 'Kevin Durant', 'TomasHertl', '#NRLBulldogsSouths', '#WGAMIX', '#rupaulsdragrace', 'Earth', '#GossipShouldBe', '#fridaymotivation', 'Derry', 'Derrick White', '#AFLNorthDons', '#DinahJane1', 'WE LOVE THE EARTH', '#DontChangeOutNow', 'Yvie', '#FridayFeeling'}
```

11

```
common trends: {'Derry', '#ConCalmaRemix', 'Derrick White', '#AFLNorthDons', '#WeLoveTheEarth', 'Lyra McKee', '#DinahJane1', '#DragRace', 'Lil Dicky', '#BLACKPINKxCorden', '#NRLBulldogsSouths'}
```

## Conclusion

- From the intersection we can see that, out of the two sets of trends (each of size 50), we have 11 overlapping topics.

### 7.4 Exploring the hot trend

We have found a hot-trend, #WeLoveTheEarth. Now let's see what story it is screaming to tell us! If we query Twitter's search API with this hashtag as a query parameter, we get back actual tweets related to it. We have a response from the search API stored in the datasets folder as 'WeLoveTheEarth.json'. So let's load this dataset and do a deep dive into this trend.

Initial code

```
In [0]: # Loading the data
        tweets = ...

        # Inspecting some tweets
        tweets[0:2]
```

#### Steps Performed

- used the open() method with 'datasets/WeLoveTheEarth.json' as an input parameter to open the file
- called the read() method on the opened file to read its content
- passed the read JSON string to the json.loads() method as input parameter for decoding it
- stored the decoded output in tweets.
- Printed first 2 tweets

#### Final code

```
In [4]: # Loading the data
        tweets = json.loads(open('datasets/WeLoveTheEarth.json').read())

        # Inspecting some tweets
        tweets[0:2]
```

#### Output

```
[{'created_at': 'Fri Apr 19 08:46:48 +0000 2019',
'id': 1119160405270523904,
'id_str': '1119160405270523904',
'text': 'RT @lildickytweets: 🌐 out now #WeLoveTheEarth https://t.co/L22XsoT5P1',
'truncated': False,
'entities': {'hashtags': [{'text': 'WeLoveTheEarth', 'indices': [30, 45]}],
'symbols': [],
'user_mentions': [{'screen_name': 'lildickytweets',
'name': 'LD',
'id': 1209516660,
'id_str': '1209516660',
'indices': [3, 18]}],
'urls': [{'url': 'https://t.co/L22XsoT5P1',
'expanded_url': 'https://youtu.be/pvuN_WvF1to',
```



```

'display_url': 'youtu.be/pvuN_WvF1to',
'indices': [46, 69]]],
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for
Android</a>',
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None,
'user': {'id': 212375312,
'id_str': '212375312',
'name': 'fake smile',
'screen_name': 'Pati95Poland',
'location': 'SWAGLAND ',
'description': '"you just got knocked the fuck out"',
'url': 'https://t.co/nxzISyYZSK',
'entities': {'url': {'urls': [{'url': 'https://t.co/nxzISyYZSK',
'expanded_url': 'http://loveeeujdb.tumblr.com/',
'display_url': 'loveeeujdb.tumblr.com',
'indices': [0, 23]]}},
'description': {'urls': []}},
'protected': False,
'followers_count': 2306,
'friends_count': 697,
'listed_count': 28,
'created_at': 'Fri Nov 05 22:25:29 +0000 2010',
'favourites_count': 5552,
'utc_offset': None,
'time_zone': None,
'geo_enabled': True,
'verified': False,
'statuses_count': 185750,
'lang': 'pl',
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': 'FFFFFF',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme18/bg.gi
f',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme18/
bg.gif',
'profile_background_tile': False,
'profile_image_url': 'http://pbs.twimg.com/profile_images/1093929135183937537/hQuxt
wKq_normal.jpg',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/109392913518393753
7/hQuxtwKq_normal.jpg',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/212375312/1522705183',
'profile_link_color': 'ABB8C2',
'profile_sidebar_border_color': 'FFFFFF',
'profile_sidebar_fill_color': 'F6F6F6',
'profile_text_color': '333333',
'profile_use_background_image': True,
'has_extended_profile': True,
'default_profile': False,
'default_profile_image': False,
'following': False,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'regular'},
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'retweeted_status': {'created_at': 'Fri Apr 19 04:22:29 +0000 2019',
'id': 1119093888524754946,
'id_str': '1119093888524754946',
'text': '🌍 out now #WeLoveTheEarth https://t.co/L22XsoT5P1',
'truncated': False,
'entities': {'hashtags': [{'text': 'WeLoveTheEarth', 'indices': [10, 25]]},
'symbols': [],
'user_mentions': [],
'urls': [{'url': 'https://t.co/L22XsoT5P1',
'expanded_url': 'https://youtu.be/pvuN_WvF1to',
'display_url': 'youtu.be/pvuN_WvF1to',

```

```

'indices': [26, 49]]},
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None,
'user': {'id': 1209516660,
'id_str': '1209516660',
'name': 'LD',
'screen_name': 'lildickytweets',
'location': 'Earth',
'description': 'Rapper/Actor/Comedian/Model #WeLoveTheEarth',
'url': 'https://t.co/aFrPkkJKqs',
'entities': {'url': {'urls': [{'url': 'https://t.co/aFrPkkJKqs',
'expanded_url': 'https://LilDicky.lnk.to/Earth',
'display_url': 'LilDicky.lnk.to/Earth',
'indices': [0, 23]]}},
'description': {'urls': []}},
'protected': False,
'followers_count': 503111,
'friends_count': 945,
'listed_count': 657,
'created_at': 'Fri Feb 22 19:06:15 +0000 2013',
'favourites_count': 7696,
'utc_offset': None,
'time_zone': None,
'geo_enabled': False,
'verified': True,
'statuses_count': 14430,
'lang': 'en',
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': '000000',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme14/bg.gif',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme14/bg.gif',
'profile_background_tile': False,
'profile_image_url': 'http://pbs.twimg.com/profile_images/1119087366679846912/XSa4fpQA_normal.png',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1119087366679846912/XSa4fpQA_normal.png',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/1209516660/1555646206',
'profile_link_color': '858585',
'profile_sidebar_border_color': 'FFFFFF',
'profile_sidebar_fill_color': 'DDEEF6',
'profile_text_color': '333333',
'profile_use_background_image': True,
'has_extended_profile': False,
'default_profile': False,
'default_profile_image': False,
'following': False,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'none'},
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 7482,
'favorite_count': 13317,
'favorited': False,
'retweeted': False,
'possibly_sensitive': False,
'lang': 'en',
'is_quote_status': False,
'retweet_count': 7482,
'favorite_count': 0,
'favorited': False,
'retweeted': False,

```

```

'possibly_sensitive': False,
'lang': 'en'),
{'created_at': 'Fri Apr 19 08:46:48 +0000 2019',
'id': 1119160404876206080,
'id_str': '1119160404876206080',
'text': '💚🌍💚 #WeLoveTheEarth _',
'truncated': False,
'entities': {'hashtags': [{'text': 'WeLoveTheEarth', 'indices': [5, 20]}],
'symbols': [],
'user_mentions': [],
'urls': []},
'metadata': {'iso_language_code': 'und', 'result_type': 'recent'},
'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for i
Phone</a>',
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None,
'user': {'id': 72150460,
'id_str': '72150460',
'name': 'Alfonsina del Mar (Dianishka Prietishka)',
'screen_name': 'Diana____X',
'location': 'México',
'description': '🌍 🇲🇽 _u200d🌊 ★★★★★ 🇲🇽 ',
'url': None,
'entities': {'description': {'urls': []}},
'protected': False,
'followers_count': 223,
'friends_count': 513,
'listed_count': 0,
'created_at': 'Sun Sep 06 23:26:24 +0000 2009',
'favourites_count': 750,
'utc_offset': None,
'time_zone': None,
'geo_enabled': True,
'verified': False,
'statuses_count': 1668,
'lang': 'es',
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': '642D8B',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme10/bg.gi
f',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme10/
bg.gif',
'profile_background_tile': True,
'profile_image_url': 'http://pbs.twimg.com/profile_images/1072296818531278848/tgn0e
2h4_normal.jpg',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/107229681853127884
8/tgn0e2h4_normal.jpg',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/72150460/1545198367',
'profile_link_color': '8400FF',
'profile_sidebar_border_color': '65B0DA',
'profile_sidebar_fill_color': '7AC3EE',
'profile_text_color': '3D1957',
'profile_use_background_image': True,
'has_extended_profile': True,
'default_profile': False,
'default_profile_image': False,
'following': False,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'none'},
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 0,
'favorite_count': 0,
'favorited': False,
'retweeted': False,
'lang': 'und']}

```

## Conclusion

- Printing the first two tweet items makes us realize that there's a lot more to a tweet than what we normally think of as a tweet — there is a lot more than just a short text!

## 7.5 Digging deeper

let's not get overwhelmed by all the information in a tweet object! Let's focus on a few interesting fields and see if we can find any hidden insights there.

Initial code

```
In [0]: # Extracting the text of all the tweets from the tweet object
texts = [... for tweet in tweets]

# Extracting screen names of users tweeting about #WeLoveTheEarth
names = [... for tweet in tweets for ... in ...]

# Extracting all the hashtags being used when talking about this topic
hashtags = ...

# Inspecting the first 10 results
print (json.dumps(texts[0:10], indent=1), "\n")
print (json.dumps(names[0:10], indent=1), "\n")
print (json.dumps(hashtags[0:10], indent=1), "\n")
```

## Steps Performed

- For each tweet in the tweets object, extracted its text field, tweet['text'], using list comprehension. Stored all the output texts in a list called texts.
- For each tweet in tweets, created an inner loop to iterate through User mentions, tweet['entities']['user\_mentions']. From each user\_mention extract its screen name field, user\_mention['screen\_name']. Stores the output in names.
- For each tweet in tweets, created an inner loop to iterate through hashtags, tweet['entities']['hashtags']. From each hashtag extract its text field, hashtag['text']. Stores the output in hashtags

## Final Code

```
In [5]: # Extracting the text of all the tweets from the tweet object
texts = [tweet['text'] for tweet in tweets]

# Extracting screen names of users tweeting about #WeLoveTheEarth
names = [user_mention['screen_name'] for tweet in tweets for user_mention in tweet['entities']['user_mentions']]

# Extracting all the hashtags being used when talking about this topic
hashtags = [hashtag['text'] for tweet in tweets for hashtag in tweet['entities']['hashtags']]

# Inspecting the first 10 results
print (json.dumps(texts[0:10], indent=1), "\n")
print (json.dumps(names[0:10], indent=1), "\n")
print (json.dumps(hashtags[0:10], indent=1), "\n")
```

## Output

```
[
"RT @lildickytweets: \ud83c\udf0e out now #WeLoveTheEarth https://t.co/L22XsoT5P1",
"\ud83d\udc9a\ud83c\udf0e\ud83d\udc9a #WeLoveTheEarth \ud83d\udc47\ud83c\udfffc",
"RT @cabeyoomoon: Ta piosenka to bop, wpada w ucho i dochody z niej id\ud0105 na dob
ry cel, warto s\ud0142ucha\ud0107 w k\ud00f3\ud0142ko i w k\ud00f3\ud0142ko gdziekolwiek si
\ud0119 ty\ud2026",
"#WeLoveTheEarth \nCzemu ja si\ud0119 pop\ud0142aka\ud0142am",
"RT @Spotify: This is epic. @lildickytweets got @justinbieber, @arianagrande, @halse
y, @sanbenito, @edsheeran, @SnoopDogg, @ShawnMendes, @Kr\ud2026",
"RT @biebercentineo: Justin : are we gonna die? \nLil dicky: you know bieber we might
die \n\nBTCH IM CRYING #EARTH #WeLoveTheEarth #WELOVEEART\ud2026",
"RT @dreamsiinflate: #WeLoveTheEarth \u201ci am a fat fucking pig\ud201d okay brendon
urie https://t.co/FdJmq31xZc",
"Literally no one:\n\nMe in the past 4 hours:\n\nI'm a koala and I sleep all the tim
e, so what, it's cute \ud83c\udfb6\n\n#WeLoveTheEarth #EdSheeranTheKoala",
"RT @Yuuupthatsme: Mia\ud0142e\ud015b by\ud0107 \ud017cyra\ud0105 #WeLoveTheEarth http
s://t.co/0kNCpU8o6q",
"RT @jaguareffects: eu prestando aten\ud00e7\ud00e3o no \ud00e1udio pra identificar cada
artista\n\n#WeLoveTheEarth https://t.co/0cDtIV2t1E"
]
[
"lildickytweets",
"cabeyoomoon",
"Spotify",
"lildickytweets",
"justinbieber",
"ArianaGrande",
"halsey",
"sanbenito",
"edsheeran",
"SnoopDogg"
]
[
"WeLoveTheEarth",
"WeLoveTheEarth",
"WeLoveTheEarth",
"EARTH",
"WeLoveTheEarth",
"WeLoveTheEarth",
"WeLoveTheEarth",
"EdSheeranTheKoala",
"WeLoveTheEarth",
"WeLoveTheEarth"
]
]
```

## Conclusion

- We are talking about a song about loving the Earth.
- A lot of big artists are the forces behind this Twitter wave, especially Lil Dicky.
- Ed Sheeran was some cute koala in the song — "EdSheeranTheKoala" hashtag!

## 7.6 Frequency analysis

Observing the first 10 items of the interesting fields gave us a sense of the data. We can now take a closer look by doing a simple, but very useful, exercise — computing frequency distributions. Starting simple with frequencies is generally a good approach; it helps in getting ideas about how to proceed further.

## Initial code

```
In [0]: # Importing modules
# ... YOUR CODE FOR TASK 6 ...

# Counting occurrences/ getting frequency dist of all names and hashtags
for item in [names, hashtags]:
    c = ...
    # Inspecting the 10 most common items in c
    print (c.most_common(10), "\n")
```

## Steps Performed

- Imported the Counter module from collections.
- Called the Counter() method with the item from the for loop as an input parameter.

## Final Code

```
In [6]: # Importing modules
# ... YOUR CODE FOR TASK 6 ...
from collections import Counter

# Counting occurrences/ getting frequency dist of all names and hashtags
for item in [names, hashtags]:
    c = Counter(item)
    # Inspecting the 10 most common items in c
    print (c.most_common(10), "\n")
```

## Methods used

- Collection library/package  
This module implements specialized container datatypes providing alternatives to Python's general-purpose built-in containers, dict, list, set, and tuple.
- Counter()  
A Counter is a dict subclass for counting hashable objects. It is a collection where elements are stored as dictionary keys and their counts are stored as dictionary values. Counts are allowed to be any integer value including zero or negative counts. The Counter class is similar to bags or multisets in other languages.
- most\_common()  
Return a list of the  $n$  most common elements and their counts from the most common to the least. If  $n$  is omitted or None, most\_common() returns *all* elements in the counter. Elements with equal counts are ordered in the order first encountered:

## Output

```
[('lildickytweets', 102), ('LeoDiCaprio', 44), ('ShawnMendes', 33), ('halsey', 31), ('ArianaGrande', 30), ('justinbieber', 29), ('Spotify', 26), ('edsheeran', 26), ('sanbenito', 25), ('SnoopDogg', 25)]
```

```
[('WeLoveTheEarth', 313), ('4future', 12), ('19aprile', 12), ('EARTH', 11), ('fridaysforfuture', 10), ('EarthMusicVideo', 3), ('ConCalm aRemix', 3), ('Earth', 3), ('aliens', 2), ('AvengersEndgame', 2)]
```

## Conclusion

- We can more safely say that this was a music video about Earth (hashtag 'EarthMusicVideo') by Lil Dicky.
- DiCaprio is not a music artist, but he was involved as well.
- We can also say that the video was released on a Friday; very likely on April 19th.

## 7.7 Activity around the trend

- Let's further analyze the data to find patterns in the activity around the tweets — **did all retweets occur around a particular tweet?**
- If a tweet has been retweeted, the *'retweeted\_status'* field gives many interesting details about the original tweet itself and its author.
- We can measure a tweet's popularity by analyzing the *retweetcount* and *favoritecount* fields. But let's also extract the number of followers of the tweeter — we have a lot of celebs in the picture, so **can we tell if they're advocating for #WeLoveTheEarth influenced a significant proportion of their followers?**

## Initial code

```
In [0]: # Extracting useful information from retweets
retweets = [(..., ..., ..., ..., ...) for tweet in tweets if 'retweeted_status' in tweet]
```

## Steps Performed

Got 'retweet\_count', 'retweeted\_status\favorite\_count', 'retweeted\_status\user\followers\_count', 'retweeted\_status\user\screen\_name', and 'text' fields for each tweet from the given for loop, respecting this order

## Final Code

```
In [7]: # Extracting useful information from retweets
retweets = [(tweet['retweet_count'], tweet['retweeted_status']['favorite_count'],
               tweet['retweeted_status']['user']['followers_count'], tweet['retweeted_status']['user']['screen_name'],
               tweet['text']) for tweet in tweets if 'retweeted_status' in tweet]
```



## 7.8 A table that speaks a 1000 words

Let's manipulate the data further and visualize it in a better and richer way — "looks matter!"

### Initial Code

```
In [0]: # Importing modules
import matplotlib.pyplot as plt
import pandas as pd

# Create a DataFrame and visualize the data in a pretty and insightful format
df = ...

df.style.background_gradient()
```

### Steps Performed

- Created a DataFrame using the `pd.DataFrame()` constructor by passing `retweets` object as input. Also set the additional input parameter `columns` to `['Retweets', 'Favorites', 'Followers', 'ScreenName', 'Text']`
- Called `groupby()` on the resulting DataFrame with `['ScreenName', 'Text', 'Followers']` as input parameter
- Then called `sum()` on the results of the `groupby` to compute an aggregate of the numerical columns.
- Finally, called `sort_values()` with input parameters by set to `['Followers']` and ascending to `False` to sort the table by decreasing the number of followers.

### Final Code

```
In [8]: # Importing modules
import matplotlib.pyplot as plt
import pandas as pd

# Create a DataFrame and visualize the data in a pretty and insightful format
df = pd.DataFrame(retweets, columns=['Retweets', 'Favorites', 'Followers', 'ScreenName', 'Text'])
df = df.groupby(['ScreenName', 'Text', 'Followers'])
df = df.sum()
df = df.sort_values('Followers', ascending=False)
df.style.background_gradient()
```

### Methods Used

- `DataFrame()`  
Two-dimensional, size-mutable, potentially heterogeneous tabular data. The data structure also contains labelled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure
- `groupby()`  
Group DataFrame using a mapper or by a Series of columns.
- `sum()`



Return the sum of the values over the requested axis.

- `sort_values()`  
Sort by the values along either axis.
- `background_gradient()`  
Colour the background in a gradient according to the data in each column (optionally row). Requires matplotlib.

## Output

Note: This is not full output just an example of it since the original output is too large for a document.

| ScreenName      | Text   | Followers | Retweets | Favorites |
|-----------------|--|-----------|----------|-----------|
| katyperry       | RT @katyperry: Sure, the Mueller report is out, but @lildickytweets "Earth" but will be too tonight. Don't say I never tried to save the w...  | 107195569 | 2338     | 10557     |
|                 | RT @lildickytweets: "Earth" but will be too tonight. Don't say I never tried to save the w...  | 107195568 | 2338     | 10556     |
| TheEllenShow    | RT @TheEllenShow: @lildickytweets, @justinbieber, @MileyCyrus, @katyperry, @ArianaGrande and more, all in one music video. My head might e...  | 77474826  | 2432     | 10086     |
| LeoDiCaprio     | RT @LeoDiCaprio: The @dicaprioFdn partners that will benefit from this collaboration include @SharkRayFund, @SolutionsProj, @GreenGrantsFun... | 18988898  | 28505    | 78137     |
|                 | RT @LeoDiCaprio: Thank you to @lildickytweets and all the artists that came together to make this happen. Net profits from the song, video,... | 18988898  | 149992   | 416018    |
| halsey          | RT @halsey: 🐢🐢🐢 meeeeeeeow 🐢 #WeLoveTheEarth https://t.co/Jr6vmKG7U  | 10564842  | 7742     | 69684     |
|                 | RT @halsey: 🐢🐢🐢 meeeeeeeow 🐢 #WeLoveTheEarth https://t.co/Jr6vmKG7U  | 10564841  | 7742     | 69682     |
| scooterbraun    | RT @scooterbraun: Watch #EARTH NOW! #WeLoveTheEarth — https://t.co/OtRtgcSZL   | 3885179   | 5925     | 14635     |
| Spotify         | RT @Spotify: This is epic. @lildickytweets got @justinbieber, @arianagrande, @halsey, @sanbenito, @edsheeran, @SnoopDogg, @ShawnMendes, @K...  | 2973277   | 107222   | 237259    |
| TomHall         | RT @TomHall: 🐢 Look Out! 🐢 #Cheetah @LeoDiCaprio #FridayFeeling #WeLoveTheEarth https://t.co/w1O1NyTbwX  | 590841    | 82       | 252       |
| lildickytweets  | RT @lildickytweets: Earth. 4/18 at 9PM PST. #WeLoveTheEarth pre-save link in my bio https://t.co/7HD2dSFYQ                                     | 503112    | 25098    | 90974     |
|                 | RT @lildickytweets: 🐢 out now #WeLoveTheEarth https://t.co/L22XsoT5P1  | 503112    | 112230   | 199773    |
|                 | RT @lildickytweets: 🐢 out now #WeLoveTheEarth https://t.co/L22XsoT5P1  | 503111    | 119712   | 213072    |
| greenpeacefr    | RT @greenpeacefr: Happening NOW in France : today, to show how much #WeLoveTheEarth, more than 2000 citizens are blocking the headquarters...  | 420789    | 128      | 192       |
| rtthingy        | RT @rtthingy: /rt/ YO GUYS LISTEN TO THIS SONGGGGG #WeLoveTheEarth https://t.co/trUgRG7QBH   | 225977    | 11       | 23        |
| SB_Projects     | RT @SB_Projects: #WeLoveTheEarth out now with @lildickytweets, @justinbieber, @ArianaGrande, @zacbrownband, and 25+ of your other faves htt... | 98418     | 271      | 730       |
| biebernovidade  | RT @biebernovidade: Rio de Janeiro: PRESENTE! #WeLoveTheEarth https://t.co/UrypKqz5f   | 83281     | 651      | 1499      |
| MendesCrewInfo  | RT @MendesCrewInfo: @ShawnMendes represents rhinos in the 'Earth' music video and feature. His line, "We're just some rhinos, horny as hec...  | 61015     | 2815     | 9599      |
| shawmxiabx      | RT @shawmxiabx: La gente piensa que el calentamiento global no es real, ya abran los ojos y actuemos de una manera real, dime ¿qué te cue...   | 53713     | 812      | 1254      |
|                 | RT @shawmxiabx: La humanidad es un asco y nunca me cansaré de decirlos ¿no creen que es momento de cambiar? Salvemos nuestro planeta, sal...   | 53713     | 604      | 1034      |
|                 | RT @shawmxiabx: Vengo a recomendarles a todos esta miniserie que realizó Netflix, en la cual nos muestra cada parte de las especies, lo q...   | 53713     | 138      | 299       |
|                 | RT @shawmxiabx: Espero que con este video se pueda crear un poco más de conciencia propia, simeplemente estamos acabando con nuestro plane...  | 53713     | 300      | 494       |
|                 | RT @SMendesQandA: #WeLoveTheEarth https://t.co/KDlKYVHFVU  | 53282     | 56       | 339       |
| SMendesQandA    | RT @SMendesQandA: Let's help save the Earth. We're the ones with the power of change. 🐢 #WeLoveTheEarth  | 53282     | 122      | 343       |
|                 | RT @SMendesQandA: So you can basically say you have a song with many of the artists we have asked you to collaborate with...smart move mend... | 53282     | 116      | 717       |
|                 | RT @SMendesQandA: "We're just some rhinos, horny as heck" — Shawn's part on #WeLoveTheEarth  | 53282     | 244      | 1420      |
|                 | RT @SMendesQandA: "We're just some rhinos horny as heck" — Shawn Mendes #WeLoveTheEarth https://t.co/URnHb0DTWN                                | 53282     | 768      | 3309      |
| ShawnUpdatesSA  | RT @ShawnUpdatesSA: Shawn Mendes: rinocerontes. #WeLoveTheEarth 'La población de rinocerontes negros está casi extinta, y las otras cuatro...  | 52424     | 850      | 1807      |
| gtbriel         | RT @gtbriel: vimos o cu do justin bieber logo em seguida a ariana sair do cu do justin e logo em seguida a ariana sendo morta pela halse...    | 49215     | 581      | 970       |
|                 | RT @gtbriel: vimos o cu do justin bieber logo em seguida a ariana sair do cu do justin e logo em seguida a ariana sendo morta pela halse...    | 49215     | 581      | 970       |
| MileySmilerNews | RT @MileySmilerNews: Miley's cameo in the Earth song is so cute #WeLoveTheEarth https://t.co/Byk3rkhMpg  | 47840     | 15       | 116       |
| ArianatorFallen | RT @ArianatorFallen: Ariana vocals had saved the earth . This song is so important #WeLoveTheEarth https://t.co/Lg9pPRDSz                      | 44426     | 84       | 258       |
| MendesNotified  | RT @MendesNotified: Rhinos - Shawn Mendes🐢🐢 "The black rhino population is nearly extinct, and the four other species of rhinos found in Af... | 39726     | 258      | 858       |
|                 | RT @MendesNotified: shawn reading his line for the song #WeLoveTheEarth: https://t.co/dcWJFDNmd  | 39726     | 2465     | 10475     |
| badweputation   | RT @badweputation: n adianta vcs comentarem sobre a importância da preservação hj e amanhã já esquecerem e tbn lembrando q o atual presiden... | 38787     | 119      | 183       |
| isbreaks        | RT @isbreaks: Quería dizer que na música do Lil Dicky tem 28 artistas incríveis mais quem eu amo mesmo é um babuino chamado Justin Bieber...   | 37473     | 95       | 215       |
|                 | RT @isbreaks: A música e o dipe tá incrível vamos espalhar o verdadeiro motivo da música que é fazer com que as pessoas pensem nos seus a...   | 37473     | 684      | 1068      |
| NoticiasSmilers | RT @NoticiasSmilers: Soy un elefante, tengo basura en mi trompa. -Pequeño cameo de Miley en la canción Earth de Lil Dicky- #WeLoveTheEarth...  | 32649     | 7        | 4         |
| ThrowbacksBTS   | RT @ThrowbacksBTS: Friendly reminder to use less plastic, recycle and contribute in some way to benefit humanity. The world is dying we nee... | 32498     | 370      | 816       |
| btsargento      | RT @btsargento: • estas son varias cosas que podemos hacer para construir un planeta mejor, sin contaminación y un lugar donde todos aporte... | 31453     | 67       | 143       |
| biebersmaniabrs | RT @biebersmaniabrs: SARU! Ouçam e assistam "Earth", música de Lil Dicky com Justin Bieber e mais 29 artistas. 🐢🐢 Video: https://t.co/oHRR...  | 26097     | 341      | 463       |
| landsrauhl      | RT @landsrauhl: I loved what lil dicky did. There's so many popular artists on the track which hopefully means more of a variety of people...  | 25071     | 128      | 306       |
| biebsrell       | RT @biebsrell: J: ¿Vamos a morir? L: ¿Sabes qué, Bieber? Podríamos morir. No voy a mentirte. Quiero decir, hay mucha gente ahí afuera que n... | 23301     | 730      | 1344      |
|                 | RT @ShawnNewsPoland: "We're just some rhinos, horny as heck" — Shwana tekst w piosence #WeLoveTheEarth https://t.co/NeOmYoe7u                  | 22067     | 97       | 523       |

## Conclusion

- Lil Dicky's followers reacted the most — 42.4% of his followers liked his first tweet.
- Even if celebrities like Katy Perry and Ellen have a huge Twitter following, their followers hardly reacted, e.g., only 0.0098% of Katy's followers liked her tweet.
- While Leo got the most likes and retweets in terms of counts, his first tweet was only liked by 2.19% of his followers.

- The large differences in reactions could be explained by the fact that this was Lil Dicky's music video. Leo still got more traction than Katy or Ellen because he played some major role in this initiative.

## 7.9 Analyzing used languages

Can we find some more interesting patterns in the data? From the text of the tweets, we could spot different languages, so let's create a frequency distribution for the languages.

### Initial Code

```
In [0]: # Extracting language for each tweet and appending it to the list of languages
tweets_languages = []
for tweet in tweets:
    ...

# Plotting the distribution of languages
%matplotlib inline
# ... YOUR CODE FOR TASK 9 ...
```

### Steps Performed

- For each tweet, the object got its language field, tweet['lang'], and appended it to the list of languages, tweets\_languages using the append() method.
- Call matplotlib's plt.hist() method with tweets\_languages as the input parameter to plot the frequency distribution of languages.

### Final Code

```
In [9]: # Extracting language for each tweet and appending it to the list of languages
tweets_languages = []
for tweet in tweets:
    tweets_languages.append(tweet['lang'])

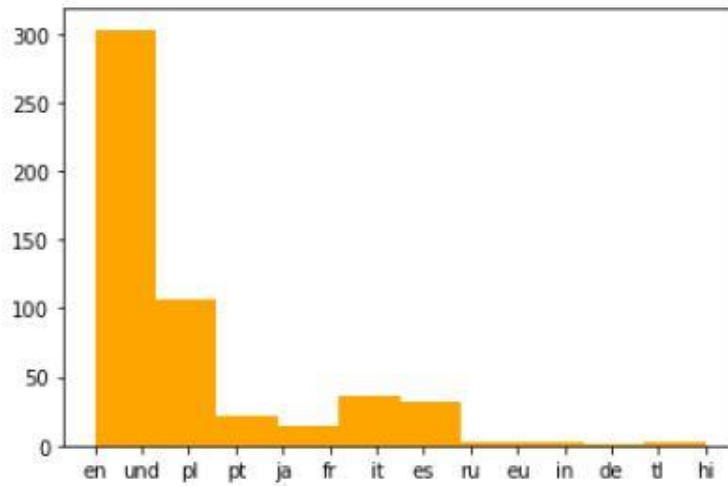
# Plotting the distribution of languages
%matplotlib inline
# ... YOUR CODE FOR TASK 9 ...
plt.hist(tweets_languages,color="orange" )
```

### Method Used

- append()
  - The append() method in python adds a single item to the existing list. It doesn't return a new list of items but will modify the original list by adding the item to the end of the list. After executing the method append on the list the size of the list increases by one.
- hist()
  - Plot a histogram.
  - Color parameter:
    - color or array-like of colors or None, default: None
    - Color or sequence of colors, one per dataset. Default (None) uses the standard line color sequence.

## Output

```
Out[9]: (array([303., 107., 22., 14., 36., 32., 3., 2., 1., 2.]),  
         array([ 0. , 1.3, 2.6, 3.9, 5.2, 6.5, 7.8, 9.1, 10.4, 11.7, 13. ]),  
         <BarContainer object of 10 artists>)
```



## Conclusion

- Most of the tweets were in English.
- Polish, Italian and Spanish were the next runner-ups.
- There were a lot of tweets with a language alien to Twitter (lang = 'und').