

```
pip install TwitterSearch
```

```
Collecting TwitterSearch
```

```
  Downloading https://files.pythonhosted.org/packages/52/77/3731f0d25c97ef1d1e
Requirement already satisfied: requests>=1.0.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: requests-oauthlib>=0.3.0 in /usr/local/lib/pyth
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/c
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dis
Building wheels for collected packages: TwitterSearch
  Building wheel for TwitterSearch (setup.py) ... done
  Created wheel for TwitterSearch: filename=TwitterSearch-1.0.2-cp37-none-any.
  Stored in directory: /root/.cache/pip/wheels/4e/98/87/016442cb92cf56a0e262e7
Successfully built TwitterSearch
Installing collected packages: TwitterSearch
Successfully installed TwitterSearch-1.0.2
```

```
from TwitterSearch import *
```

```
import time
```

```
import json
```

```
import tweepy
```

```
import pandas as pd
```

```
# Definición diccionario:
```

```
def extract_place(row):
```

```
    if row['Place Info']:
```

```
        return row['Place Info'].full_name
```

```
    else:
```

```
        return None
```

```
try:
```

```
    consumer_key = 'ZxWvLYiEjuaYF2tcLJ8Bxs1rx'
```

```
    consumer_secret = 'HqjuFRM78m9ay274oiKksddn5x9VAowOQeGcOKoBn4q2Bt6dOP'
```

```
    access_token = '805863249191178241-DNW0knLtRLjLPGBull9Q5tFmSvypNO7'
```

```
    access_token_secret = 'p10rsJOxiPKHSQGLZNAx3Jl0yS9zQOEri4nAL9SFz7J0f'
```

```
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
```

```
    auth.set_access_token(access_token, access_token_secret)
```

```
    # con este objeto realizaremos todas las llamadas al API
```

```
    #print (json.dumps(tweet._json, indent=4))
```

```
    api = tweepy.API(auth,
```

```
                        wait_on_rate_limit=True,
```

```
                        wait_on_rate_limit_notify=True)
```

```
    conta =0
```

```
    usuarios=[]
```

```
    enlaces=[]
```

```
    descripcion=[]
```

```
    total_palabras=[]
```

```
    hashtags=[]
```

```
    num_retweet=[]
```

```
    num_favor=[]
```

```
    print('antes')
```

```
    for tweet in tweepy.Cursor(api.search, q="Guillermo Lasso",tweet_mode="extended
```

```
        #print('usuario ->', tweet.user.screen_name,': enlaces externos ->' ,tweet.us
```

```
        #print('texto ->', tweet.full text)
```

```

# print( % ( tweet['user']['screen_name']['url']['listed_count'], tweet['text']
# print('hashtag ->', tweet._json['entities']['hashtags'])
# print('conteo retuit ->', tweet.retweet_count, 'favoritos ->', tweet.favorite_count)
user= tweet.user.screen_name
usuarios.append(user)
url = tweet.user.url
enlaces.append(url)
texto = tweet.full_text
descripcion.append(texto)
total_palabras.append(len(texto)-1)
hashtag= tweet._json['entities']['hashtags']
hashtags.append(hashtag)
retuit= tweet.retweet_count
num_retweet.append(retuit)
favorito = tweet.favorite_count
num_favor.append(favorito)
conta=conta+1
time.sleep(2)
print(conta)
print('fin primero')
except TwitterSearchException as e:
    print('error al hacer el scraping' + e)

```

☞ **Streaming output truncated to the last 5000 lines.**

```

3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035

```

```

3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059

```

```
print(len(usuario))
```

```
8000
```

```

import pandas as pd
from google.colab import drive
drive.mount('/content/drive/')

```

Drive already mounted at /content/drive/; to attempt to forcibly remount, call

```

datos = {'usuario': usuarios, 'enlace_externo': enlaces, 'texto_descriptivo': descrip
df_datos = pd.DataFrame(datos)
df_datos.to_csv(r'/content/drive/MyDrive/Simulacion/tweet2.csv', index=False)

```

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-3-ed1c00ecda0a> in <module>()
      1
----> 2 datos = {'usuario': usuarios, 'enlace_externo':
enlaces, 'texto_descriptivo': descripcion, 'total_palabras': total_palabras,
'hashtag': hashtags, 'retweet': num_retweet, 'favorito': num_favor}
      3 df_datos = pd.DataFrame(datos)
      4 df_datos.to_csv(r'/content/drive/MyDrive/Simulacion/tweet2.csv',
index=False)

NameError: name 'usuarios' is not defined

```

```
df = pd.read_csv('/content/drive/MyDrive/Simulacion/tweet2.csv')
```

```
df.columns = ['usuario', 'enlace_externo', 'texto_descriptivo', 'total_palabras', 'hashtag']

df.shape

(8001, 7)

df.head()
```

	usuario	enlace_externo	texto_descriptivo	total_palabras	hashtag
0	usuario	enlace_externo	texto_descriptivo	total_palabras	
1	Raagsanc	NaN	RT @eluniversocom: De la Fundación Ecuador Lib...	139	
2	CEDENMA	https://t.co/4outo4EZnw	Este #DiaMundialDelMedioAmbiente revisamos los... @RonSangines	252	'DiaM'

```
df.fillna(0, inplace=True)
df.describe()
```

	usuario	enlace_externo	texto_descriptivo	total_palabras	hashtag
count	8001	8001	8001	8001	8001
unique	4528	638	2195	309	418
top	HaroldCaiced	0	RT @eluniversocom: El presidente Guillermo Las...	139	[]

```
df.drop([''])

[]
1 1 1

#filtered_data = df[(df['total_palabras']) & (df['retweet'])]
import matplotlib.pyplot as pp
from pandas import Series
%matplotlib inline
import numpy as np
from numpy.random import randn
```

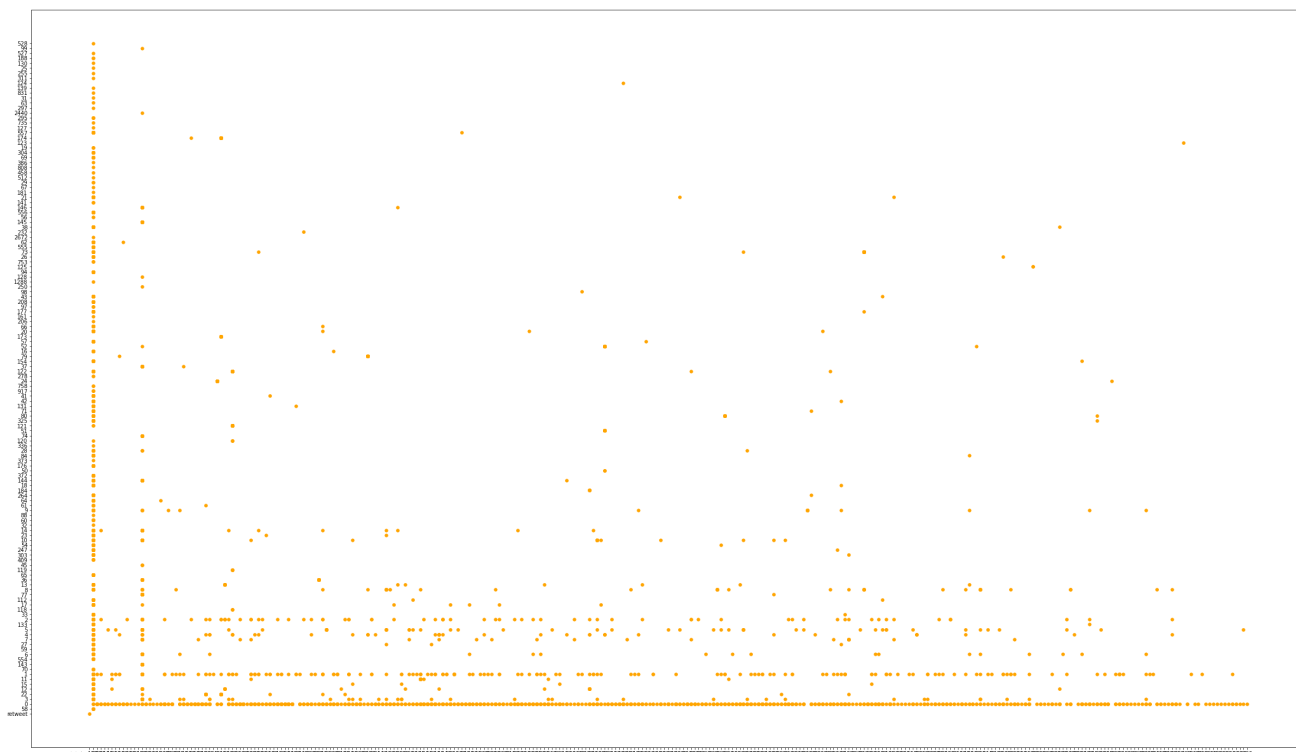
```
colores=['orange']
```

```
tamanios=[30,60]

f1 = df['total_palabras'].values
f2 = df['retweet'].values

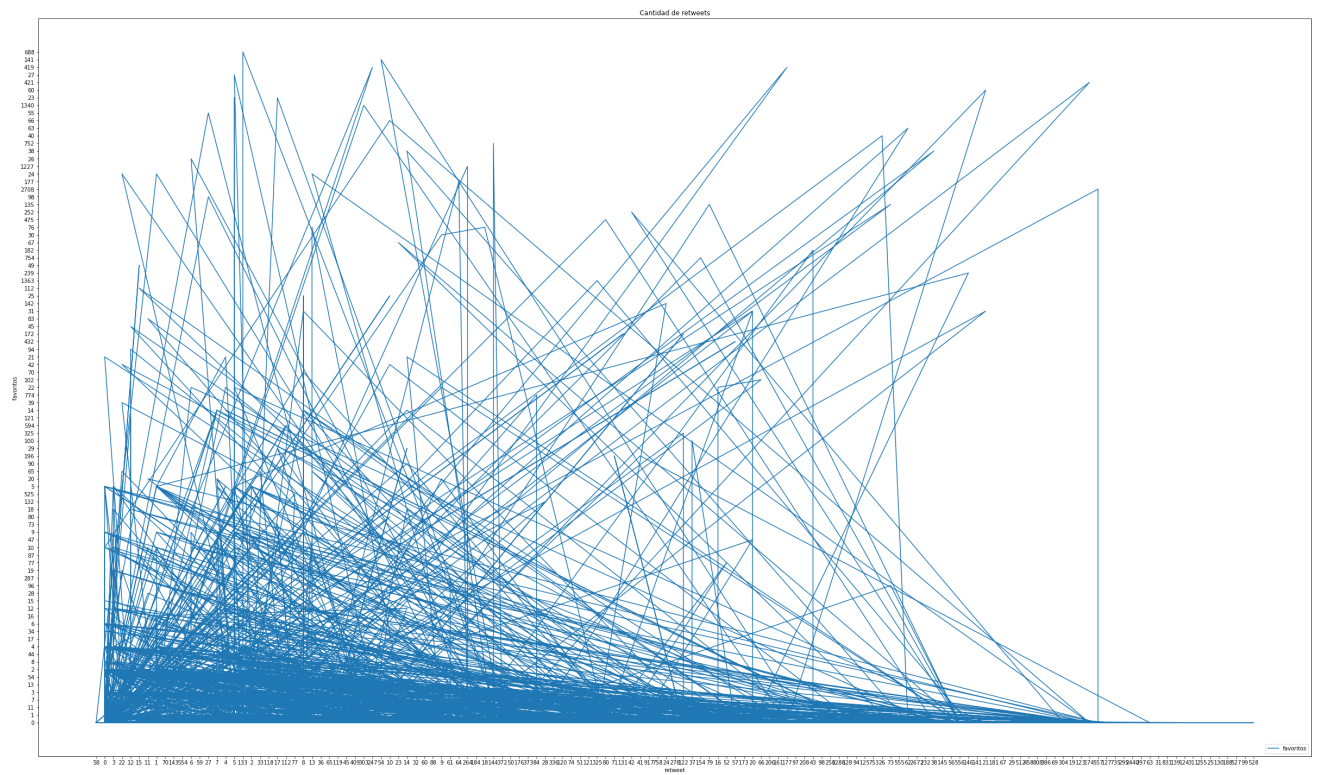
pp.scatter(f1, f2, c=colores, s=tamanios[0])
pp.gcf().set_size_inches(42, 25)

pp.show()
```



▼ Regresión

```
pp.plot(df.retweet[1:], df.favorito[1:])
pp.title('Cantidad de retweets')
pp.ylabel('favoritos ')
pp.xlabel('retweet')
pp.legend(['favoritos', 'retweet'], loc='lower right')
pp.gcf().set_size_inches(42, 25)
pp.show()
```



```
from datetime import datetime, timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model

total_datos = df.loc[:, ['total_palabras', 'retweet', 'favorito']]
```

```
X_train= (total_datos.total_palabras[1:].astype(np.float64))
y_train= (total_datos.retweet[1:].astype(np.float64))

#X_train = np.array(dataX)
#y_train = df['retweet'].values
#objeto regresion
```

```
regres = linear_model.LinearRegression()

regres.fit(np.array(X_train).reshape(-1, 1) ,y_train)

# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', regres.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Intercepcion: \n', regres.intercept_)
# Error Cuadrado Medio
prediccion = regres.predict([[150]])
print('prediccion-->',int(prediccion))
print(' ')
```

```
Coefficients:
[-0.794326]
Intercepcion:
246.43523804208843
prediccion--> 127
```

```
pp.scatter(X_train, y_train)
x_real = np.array(range(1, 50))
pp.gcf().set_size_inches(42, 25)
pp.plot(x_real, regres.predict(x_real.reshape(-1, 1)), color='green')
pp.title('Análisis de los retweet')
pp.xlabel('total de palabras')
pp.ylabel('total de retweet')
pp.show()
```

