

# Procesamiento de datos con Python

Proyecto final

Eliseo Orellan Anguiano



# Eliseo Orellan Anguiano

@eliseo5775

<https://github.com/eliseo5775>

[https://www.linkedin.com/in/eliseoor  
ellan](https://www.linkedin.com/in/eliseoor<br/>ellan)

MBA Associate / Fleet Strategy

@Aeromexico

Python & R



# Objetivos

- Obtener datos de una fuente remota
- Crear un proceso de ETL (Extracción, Transformación y Carga)
- Utilizar funciones de Python Standard Library
- Crear funciones de análisis con ``filter`` & ``map``
- Utilizar ``Jupyter Notebook``
- Entornos virtuales de Python3
- Utilizar Pandas & Matplotlib
- Usar github

# Proyecto

Vivimos en épocas difíciles, COVID19 ha cambiado la manera de ver las cosas en diferentes aspectos, las redes sociales como Twitter han capturado muchos de los mensajes publicados por las personas alrededor del mundo.

El proyecto que trabajarás será analizar las publicaciones de usuarios de twitter relacionadas al tema del momento.


1.- Repositorio en github.com

# Repositorio en github.com

Con tu cuenta de github crea un nuevo  
repositorio público con el nombre:


data-analysis-project


 [eliseo5775](#) / [data-analysis-project](#)

 Unwatch ▾

1

 Code

 Issues


 Pull requests

 Actions

 Projects

 Wiki

 Security

 master ▾

 1 branch

 0 tags

Go to file


Add file ▾

 Code ▾



eliseo5775 Cuenta URL

2bc5fcb 11 minutes ago

 25 commits



Dataset

Descarga de csv

5 days ago



.gitignore

Commit inicial

5 days ago



000\_ProyectoCompleto.py

Cuenta URL

11 minutes ago



001\_Dias\_transcurridos.py

Cuenta URL

11 minutes ago



002\_Dist\_Geografica.py

Ejercicio 4 Commit

1 hour ago



[003\\_Dist\\_Usuarios.py](#)

commit de funcion de 4

5 hours ago



004\_Dist\_Tiempo.py

Cuenta URL

12 minutes ago



005\_Metadata.py

Cuenta URL

12 minutes ago



READ\_ME.md

Commit inicial

5 days ago



requirements.txt

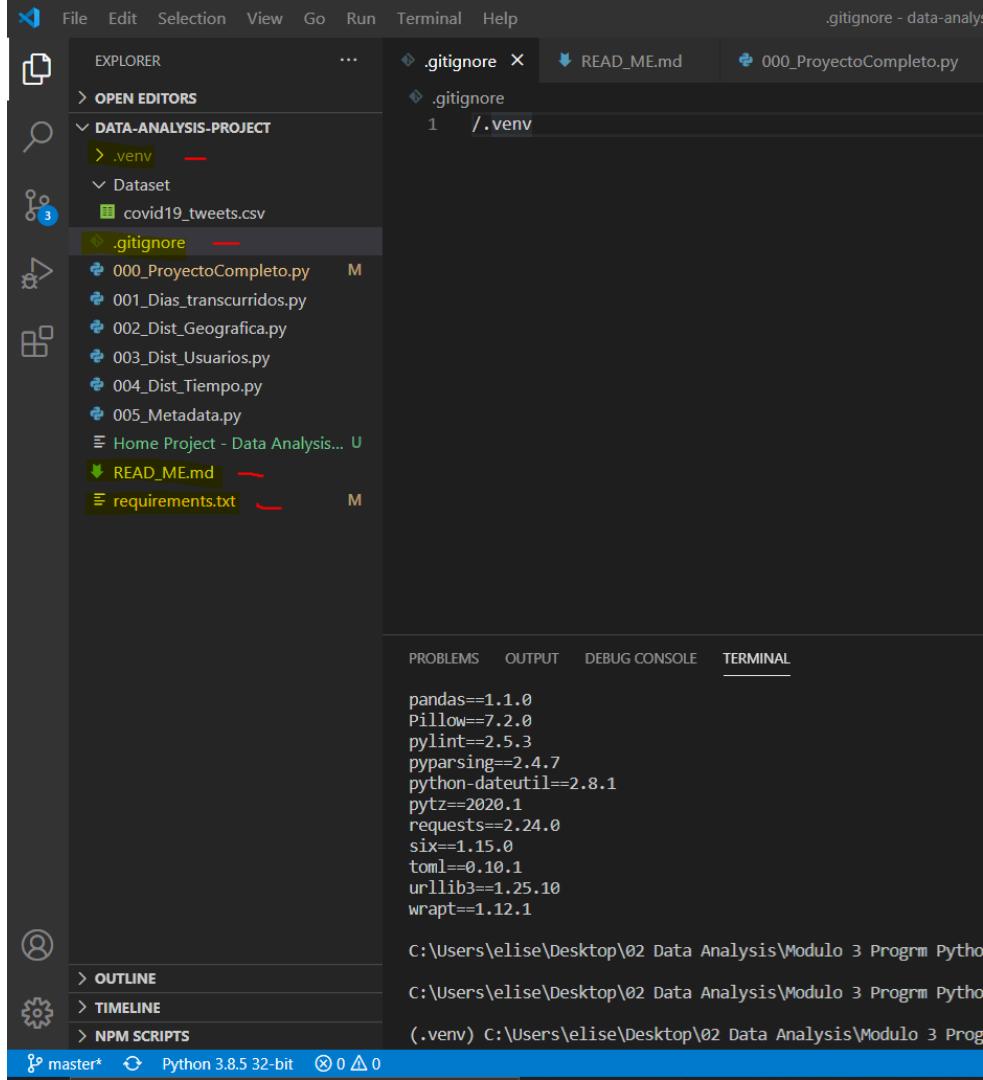
modificacion de vsnv

4 days ago

## 2.- Proyecto de python

# Proyecto de python

1. Crea un folder en tu computador,
2. dentro de él crearás un entorno virtual de python.
3. Cuando lo tengas, agrega al proyecto los archivos de:
  1. requerimientos,
  2. el archivo para ignorar,
  3. así como el archivo de lectura en formato markdown.





## 4.- Obtención de información

# Obtención de información

Crea un script en python que se encargue de descargar y guardar en memoria el dataset que encontrarás en la siguiente url:

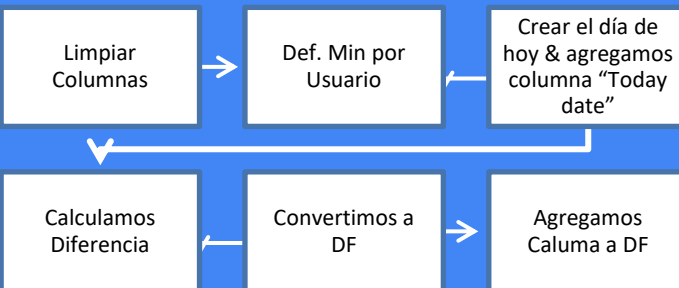
[http://galileoguzman.com/data/covid19\\_tweets.csv](http://galileoguzman.com/data/covid19_tweets.csv)

```
000_ProyectoCompleto.py • 001_Dias_transcurridos.py 002_Dist_Geografica.py 003_Dist_Usuarios.py
000_ProyectoCompleto.py > ...
1
2 import requests
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 from datetime import datetime, date
6
7 # Descarga de data de dataset de twetts http://galileoguzman.com/data/covid19_tweets.csv
8
9 df = requests.get('http://galileoguzman.com/data/covid19_tweets.csv')
10 print(df.status_code)
11
12 #Ejercicio 1 #####
13
14 #leer de CSV
15
16 FILENAME = 'Dataset/covid19_tweets.csv'
17 df = pd.read_csv(FILENAME)
18
19 #Transformas la data
20 df["user_name"] = df['user_name'].astype('string')
21 df["user_location"] = df['user_location'].astype('string')
```

## 5.- Tareas de análisis

## 5.1.- Días transcurridos

Ejecuta una función que calcule cuantos días transcurridos han pasado hasta el día que se ejecute, desde la primera vez que un usuario publicó un tweet acerca del CoronaVirus.



```
001_Dias_transcurridos.py X 002_Dist_Geografica.py 003_Dist_Usuarios.py 004_Dist_Tiempo.py 005_Metadata.py
001_Dias_transcurridos.py > ...

34 def diastranscurridos(df_in001):
35     # limpiar el data set y quitar columnas feas, ademas transoformamos
36     df_cl_in001 = df_in001.drop(columns=['user_followers','user_location','user_friends','user_favourites','us
37     #date min por usuario
38     df_grp_min = df_cl_in001.loc[df_in001.groupby('user_name').date.idxmin()]
39     #agregamos el current date
40     current_date = pd.to_datetime(date.today())
41     df_grp_min.insert(2,'today_date',current_date)
42     #calculamos los dias entre dos fechas y agregamos columna
43     df_dias_trans = df_grp_min['today_date']-df_grp_min['date']
44     # convertimos a frame
45     df_dias_trans = df_dias_trans.to_frame()
46     # agregar la columna de dias trans
47     df_grp_min["dias_transcurridos"] = df_dias_trans
48     df_grp_min = df_grp_min.reset_index()
49     return(df_grp_min)
50
51 #Ejecutamos función
52 df_dias_trans = diastranscurridos(df)
53 print(f'Los dias transcurridos desde el primer tweet de cada usuario son los siguientes \n\n{df_dias_trans}')
54
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

index	user_name	date	today_date	dias_transcurridos
0	18134 "Fantacy Tv" Channel ( A Mirror of News Agency	2020-07-26 08:43:05	2020-09-04	39 days 15:16:55
1	28117 "John"	2020-07-27 04:53:02	2020-09-04	38 days 19:06:58
2	46153 "Star Grammy Unleashed" DMs will not be answered	2020-07-31 18:48:32	2020-09-04	34 days 05:11:28
3	49651 "The Director" Chris Keeling	2020-07-31 17:45:23	2020-09-04	34 days 06:14:37
4	75786 "We Need a Lincoln and We Have a Buchanan"	2020-08-06 16:17:45	2020-09-04	28 days 07:42:15
...	...	...	...	...
50379	79858 snap! crackle! pop! Gene/CRISPRs!	2020-08-06 15:14:48	2020-09-04	2 8 days 08:45:12
50380	159311 Andrew Clark	2020-08-22 07:15:49	2020-09-04	1 2 days 16:44:11
50381	145705 Murphy	2020-08-17 04:37:18	2020-09-04	17 days 19:22:42
50382	13013 VANTA BLACK	2020-07-25 03:08:20	2020-09-04	4 0 days 20:51:40
50383	52939 Stodius	2020-08-01 18:47:02	2020-09-	04 33 days 05:12:58

[50384 rows x 5 columns]

## 5.2.- Distribución geográfica

- Crea una función que se encargue de mostrar cuántos tweets por ciudad han sido publicados.
- Crea una función que se encargue de mostrar una gráfica de barras con la información obtenida de la función anterior.

```
001_Dias_transcurridos.py X 002_Dist_Geografica.py X 003_Dist_Usuarios.py 004_Dist_Tiempo.py
002_Dist_Geografica.py > ...
21 #limpiamos NA
22 df = df.dropna(axis = 0)
23
24 #- Crea una función que se encargue de mostrar cuántos tweets por ciudad han sido publicados
25
26 def agrupa_ubicacion_geografica(df):
27     df_agrupado = df.groupby(['user_location']).size().reset_index(name="count")
28     df_ordenado = df_agrupado.sort_values(['count'],ascending=[False])
29     return df_ordenado
30
31 tweets_por_ciudad = agrupa_ubicacion_geografica(df)
32 print(f'Tweets por ciudad \n{tweets_por_ciudad}')
33
34 #- Crea una función que se encargue de mostrar una gráfica de barras con la información obtenida
35 def grafica_ubicacion_geografica(df_in002):
36     df_in002.plot(kind='bar',x='user_location',y='count', title = '# Tweets por ciudad')
37     return plt.show()
38
39 grafica_ubicacion_geografica(tweets_por_ciudad.iloc[1:50,:])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Versión 10.0.18362.1016]

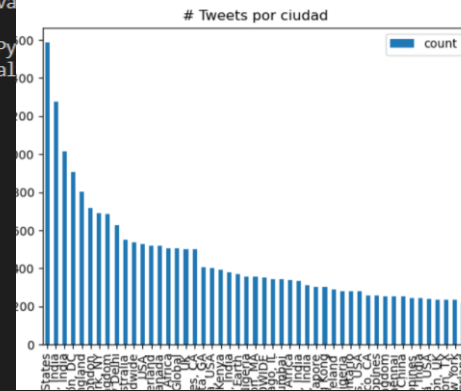
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\elise\Desktop\02 Data Analysis\Modulo 3 Program Python>

/Desktop/02 Data Analysis/Modulo 3 Program Python/data-analysis

Tweets por ciudad

	user_location	count
7334	India	2746
16613	United States	1586
11050	New Delhi, India	1277
10477	Mumbai, India	1014
17236	Washington, DC	905
...	...	...
8033	Kearney, MO	1
8035	Kedah, Malaysia	1
8036	Keep passing the open windows	1
8037	Keeping my distance!	1
19694		1



## 5.3.- Distribución por usuarios

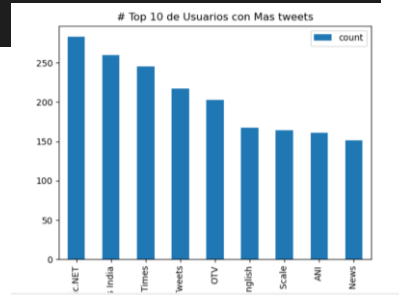
- Crea una función que muestre el resultado de cuántos usuarios por ciudad hay con publicación.
- Crea una función que muestre cuales son los usuarios que han publicado más tweets.

```
31 def agrupa_rt_location(df):
32     #filtrar los que no son retweet
33     df_isrt = df['is_retweet'] == False
34     df_filtrado = df[df_isrt]
35     #filtrar usuarios unicos
36     df_filtrado = df_filtrado.drop_duplicates(subset=['user_name'])
37     #agrupar por user location
38     df_grouped = df_filtrado.groupby(['user_location']).size().reset_index(name="count")
39     df_order = df_grouped.sort_values(['count'],ascending=[False])
40
41     df_order = df_order.iloc[1:20,:]
42     return df_order
```



#- Crea una función que muestre cuales son los usuarios que han publicado más tweets.

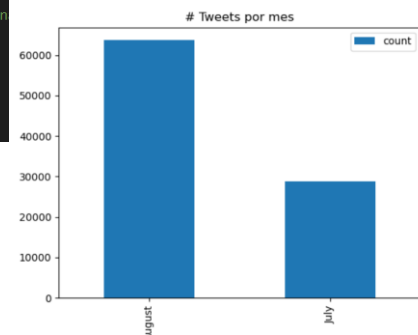
```
def agrupa_top_user(df):
    #filtrar los que no son retweet
    df_grouped = df.groupby(['user_name']).size().reset_index(name="count")
    df_order = df_grouped.sort_values(['count'],ascending=[False])
    df_order1 = df_order.iloc[1:10,:]
    return df_order1
```



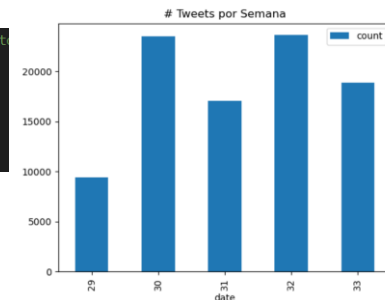
## 5.4.- Distribución por periodos de tiempo

- Crea una función que muestre cuántos tweets han sido publicados por mes, aparte muestrales en una tabla.
- Crea una función que muestre cuántos tweets han sido publicados por semanas, basados en el punto anterior.
- Crea una función que muestre cuales son las horas con más tweets basados en la división del punto anterior, ejemplo:
  - Mañana entre 07-08 horas
  - Tarde entre 15-16 horas
  - Noche entre 21-22 horas

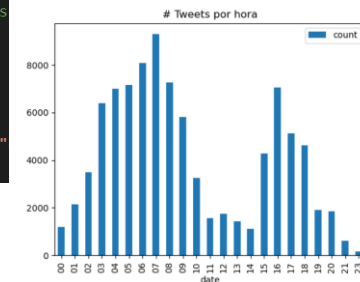
```
#- Crea una función que muestre cuántos tweets han sido publicados por mes, aparte muestrales en un
def agrupa_por_mes (df_in004):
    df_mes = df_in004.groupby(df['date'].dt.strftime('%B')).size().reset_index(name="count")
    df_order = df_mes.sort_values(['count'],ascending=[False])
    return df_order
```



```
38 #- Crea una función que muestre cuántos tweets han sido publicados por semanas, basados en el punto
39
40 def agrupa_por_sem (df_in004):
41     df_mes = df_in004.groupby(df['date'].dt.strftime('%u')).size().reset_index(name="count")
42     return df_mes
43
44 top_sem = agrupa_por_sem(df)
```



```
def agrupa_por_hr (df_in004):
    df_hr = df_in004.groupby(df['date'].dt.strftime('%H')).size().reset_index(name="count")
    return df_hr
```



## 5.5.- Metadata de tweets

- Crea una función que se encargue de mostrar el total de tweets publicados con base en:
  - Publicados con imágenes
  - Publicados con urls
- Crea una función que se encargue de mostrar las palabras más repetidas por país.

```
# - Crea una función que se encargue de mostrar el total de tweets publicados con base en:  
# - Publicados con imágenes  
# - Publicados con urls
```

```
def tiene_url (df):  
    #filtrar los que tienen URL  
    df_url = df['text'].str.contains('https')  
    return df_url
```

```
C:\Users\elise\Desktop\02 Data Analysis\Modulo 3 Progrm Python\data-analysis-project>C:/Users/elise  
/Desktop/02 Data Analysis/Modulo 3 Progrm Python/data-analysis-project/005_Metadata.py  
118893 tweets contienen URL  
Top 10 palabras mas frecuentes  
[('#COVID19', 63996), ('the', 53188), ('to', 42479), ('of', 36502), ('in', 32315), ('and', 23580),
```

```
# - Crea una función que se encargue de mostrar las palabras más repetidas por país.  
def mas_frecuente (df_in006):  
    df_mf = Counter(" ".join(df_in006["text"]).split()).most_common(10)  
    return df_mf  
  
print(f'Top 10 palabras mas frecuentes \n {mas_frecuente(df)}')
```

```
C:\Users\elise\Desktop\02 Data Analysis\Modulo 3 Progrm Python\data-analysis-project>C:/Users/elise  
/Desktop/02 Data Analysis/Modulo 3 Progrm Python/data-analysis-project/005_Metadata.py  
118893 tweets contienen URL  
Top 10 palabras mas frecuentes  
[('#COVID19', 63996), ('the', 53188), ('to', 42479), ('of', 36502), ('in', 32315), ('and', 23580),
```