Diplomado_Mod13_Ejercicio2

September 28, 2023

1. Descargar varios archivos referente a un tema en particular

[17]:
```
import pandas as pd

dfc = pd.read_csv("https://gitlab.com/dgtic5/res/-/raw/main/
 ↪aprendizajeSupervizado/
 ↪Yemen%20Cholera%20Outbreak%20Epidemiology%20Data%20-%20Data_Governorate_Level.
 ↪csv" \
                  ,header = 0,dtype ={'Date': str, 'Governorate': str,
                            'Cases': str,
                            'Deaths': int,
                            'CFR (%)': float,
                            'Attack Rate (per 1000)': float,
                            'COD Gov English': str,
                            'COD Gov Arabic': str,
                            'COD Gov Pcode': str} \
                  ,keep_default_na=False)
```

[18]:
```
dfc
```

[18]:

	Date	Governorate	Cases	Deaths	CFR (%)	Attack Rate (per 1000)	\
0	2018-02-18	Amran	103965	176	0.17	89.582	
1	2018-02-18	Al Mahwit	62887	151	0.24	86.122	
2	2018-02-18	Al Dhale'e	47136	81	0.17	64.438	
3	2018-02-18	Hajjah	121287	422	0.35	52.060	
4	2018-02-18	Sana'a	76250	123	0.16	51.859	
…	…	…	…	…	…	…	
2909	2017-05-22	Raymah	549	4	0.70	0.870	
2910	2017-05-22	Aden	489	12	2.50	0.510	
2911	2017-05-22	Al_Jawf	189	3	1.60	0.290	
2912	2017-05-22	Lahj	168	0	0.00	0.160	
2913	2017-05-22	Ma'areb	2	0	0.00	0.010	

	COD Gov English	COD Gov Arabic	COD Gov Pcode
0	Amran		29
1	Al Mahwit		27
2	Al Dhale'e		30
3	Hajjah		17

```
4              Sana'a                          23
…                …              …              …
2909           Raymah                          31
2910             Aden                          24
2911          Al Jawf                          16
2912             Lahj                          25
2913            Marib                          26

[2914 rows x 9 columns]
```

[20]:
```
dfc['Cases']
```

[20]:
```
0       103965
1        62887
2        47136
3       121287
4        76250
         …
2909       549
2910       489
2911       189
2912       168
2913         2
Name: Cases, Length: 2914, dtype: object
```

[21]:
```
# Define the convert_to_int function.
def convert_to_int(x):
    """
    Converts a string to an int, removing commas if necessary.

    Args:
        x (str): The string to convert.

    Returns:
        int: The converted int.
    """

    # Remove commas.
    x = x.replace(',', '')

    # Try to convert to int.
    try:
        return int(x)
    except ValueError:
        # Return the original string if the conversion fails.
        return x
```

```
# Convert the 'Cases' column to int.
dfc['Cases'] = dfc['Cases'].apply(convert_to_int)
```

[23]:
```
dfc.describe()
```

[23]:

	Cases	Deaths	CFR (%)	Attack Rate (per 1000)
count	2914.000000	2914.000000	2914.000000	2914.000000
mean	26067.229581	87.131434	0.383253	18.652564
std	28246.793106	96.037509	0.380705	17.531846
min	2.000000	0.000000	0.000000	0.000000
25%	3336.250000	7.000000	0.150000	5.090250
50%	16522.000000	59.000000	0.300000	14.601000
75%	40385.000000	140.000000	0.500000	25.633750
max	155908.000000	422.000000	9.000000	89.582000

[4]:
```
ls -la
```

```
total 364
drwxr-xr-x  19 root root    4096 Sep 28 02:46  ./
drwxr-xr-x  19 root root    4096 Sep 28 02:46  ../
-rw-r--r--   1 root root   38689 Sep 22 06:28  amba-censo-csv
-rw-r--r--   1 root root   39212 Sep 22 06:48  bes2017_part1.csv
-rw-r--r--   1 root root   39606 Sep 22 06:48  bes2017_part2.csv
-rw-r--r--   1 root root   32598 Sep 22 06:47  bes2017_part3.csv
-rw-r--r--   1 root root   32598 Sep 22 06:47  bes2017_part4.csv
lrwxrwxrwx   1 root root       7 Aug 14 21:37  bin ->
usr/bin/
drwxr-xr-x   4 root root    4096 Sep  2 00:19  boot/
-rw-r--r--   1 root root   39212 Sep 22 06:43  british-election-study-csv-files
-rw-r--r--   1 root root     646 May 28  2021  copyright
drwxr-xr-x  14 root root    2800 Sep 28 02:46  dev/
drwxr-xr-x 102 root root    4096 Sep 28 16:32  etc/
-rw-r--r--   1 root root   32598 Sep 22 06:33
'file?filename=BES2017_W13_Panel_v1.0-3.csv'
-rw-r--r--   1 root root   32598 Sep 22 06:43
'file?filename=BES2017_W13_Panel_v1.0-4.csv'
drwxrwxr-x   7 root hadoop  4096 Sep  9 16:34  hadoop/
drwxr-xr-x   5 root root    4096 Sep 23 18:21  home/
lrwxrwxrwx   1 root root       7 Aug 14 21:37  lib ->
usr/lib/
lrwxrwxrwx   1 root root       9 Aug 14 21:37  lib32 ->
usr/lib32/
lrwxrwxrwx   1 root root       9 Aug 14 21:37  lib64 ->
usr/lib64/
lrwxrwxrwx   1 root root      10 Aug 14 21:37  libx32 ->
usr/libx32/
drwx------   2 root root   16384 Aug 14 21:36  lost+found/
drwxr-xr-x   2 root root    4096 Aug 14 21:37  media/
```

```
drwxr-xr-x   2 root root    4096 Aug 14 21:37  mnt/
drwxr-xr-x   9 root root    4096 Sep  9 16:34  opt/
dr-xr-xr-x 175 root root       0 Sep 28 02:46  proc/
drwx------   8 root root    4096 Sep 22 06:28  root/
drwxr-xr-x  31 root root     880 Sep 28 09:16  run/
lrwxrwxrwx   1 root root       8 Aug 14 21:37  sbin ->
usr/sbin/
drwxr-xr-x   2 root root    4096 Aug 14 21:37  srv/
dr-xr-xr-x  13 root root       0 Sep 28 02:46  sys/
drwxrwxrwt  39 root root    4096 Sep 28 18:46  tmp/
drwxr-xr-x  14 root root    4096 Aug 14 21:37  usr/
drwxr-xr-x  12 root root    4096 Sep  2 00:48  var/
```

Descargamos el archivo en nuestro Hadoop

[10]:
```
import urllib.request

url = 'https://gitlab.com/dgtic5/res/-/raw/main/aprendizajeSupervizado/
 ↪Yemen%20Cholera%20Outbreak%20Epidemiology%20Data%20-%20Data_Governorate_Level.
 ↪csv'
filename = 'Yemen_epidemiology_Data_Governorate_Level.csv'

urllib.request.urlretrieve(url, filename)
```

[10]:
```
('Yemen_epidemiology_Data_Governorate_Level.csv',
 <http.client.HTTPMessage at 0x7fcd71d7c820>)
```

[11]:
```
ls -la
```

```
total 540
drwxr-xr-x  19 root root     4096 Sep 28 19:11  ./
drwxr-xr-x  19 root root     4096 Sep 28 19:11  ../
-rw-r--r--   1 root root   180155 Sep 28 19:11
Yemen_epidemiology_Data_Governorate_Level.csv
-rw-r--r--   1 root root    38689 Sep 22 06:28  amba-censo-csv
-rw-r--r--   1 root root    39212 Sep 22 06:48  bes2017_part1.csv
-rw-r--r--   1 root root    39606 Sep 22 06:48  bes2017_part2.csv
-rw-r--r--   1 root root    32598 Sep 22 06:47  bes2017_part3.csv
-rw-r--r--   1 root root    32598 Sep 22 06:47  bes2017_part4.csv
lrwxrwxrwx   1 root root        7 Aug 14 21:37  bin ->
usr/bin/
drwxr-xr-x   4 root root     4096 Sep  2 00:19  boot/
-rw-r--r--   1 root root    39212 Sep 22 06:43  british-election-study-csv-files
-rw-r--r--   1 root root      646 May 28  2021  copyright
drwxr-xr-x  14 root root     2800 Sep 28 02:46  dev/
drwxr-xr-x 102 root root     4096 Sep 28 16:32  etc/
-rw-r--r--   1 root root    32598 Sep 22 06:33
'file?filename=BES2017_W13_Panel_v1.0-3.csv'
-rw-r--r--   1 root root    32598 Sep 22 06:43
```

```
'file?filename=BES2017_W13_Panel_v1.0-4.csv'
drwxrwxr-x   7 root hadoop   4096 Sep  9 16:34  hadoop/
drwxr-xr-x   5 root root     4096 Sep 23 18:21  home/
lrwxrwxrwx   1 root root        7 Aug 14 21:37  lib ->
usr/lib/
lrwxrwxrwx   1 root root        9 Aug 14 21:37  lib32 ->
usr/lib32/
lrwxrwxrwx   1 root root        9 Aug 14 21:37  lib64 ->
usr/lib64/
lrwxrwxrwx   1 root root       10 Aug 14 21:37  libx32 ->
usr/libx32/
drwx------   2 root root    16384 Aug 14 21:36  lost+found/
drwxr-xr-x   2 root root     4096 Aug 14 21:37  media/
drwxr-xr-x   2 root root     4096 Aug 14 21:37  mnt/
drwxr-xr-x   9 root root     4096 Sep  9 16:34  opt/
dr-xr-xr-x 174 root root        0 Sep 28 02:46  proc/
drwx------   8 root root     4096 Sep 22 06:28  root/
drwxr-xr-x  31 root root      880 Sep 28 09:16  run/
lrwxrwxrwx   1 root root        8 Aug 14 21:37  sbin ->
usr/sbin/
drwxr-xr-x   2 root root     4096 Aug 14 21:37  srv/
dr-xr-xr-x  13 root root        0 Sep 28 02:46  sys/
drwxrwxrwt  39 root root     4096 Sep 28 18:46  tmp/
drwxr-xr-x  14 root root     4096 Aug 14 21:37  usr/
drwxr-xr-x  12 root root     4096 Sep  2 00:48  var/
```

Creamos las carpetas necesarias para subir el archivo a Hadoop

[13]:
```
import subprocess

command = 'hdfs dfs -mkdir -p /tmp/dcd/Yemen/input'
subprocess.run(command, shell=True)
```

[13]:
```
CompletedProcess(args='hdfs dfs -mkdir -p /tmp/dcd/Yemen/input', returncode=0)
```

[15]:
```
import subprocess

command = 'hdfs dfs -ls /tmp/dcd'
subprocess.run(command, shell=True)
```

```
Found 12 items
drwxr-xr-x   - root              hadoop          0 2023-09-23 18:24
/tmp/dcd/OnTimeDB
drwxr-xr-x   - root              hadoop          0 2023-09-28 19:20
/tmp/dcd/Yemen
drwxr-xr-x   - root              hadoop          0 2023-09-22 07:15
/tmp/dcd/british
drwxr-xr-x   - root              hadoop          0 2023-09-22 08:39
/tmp/dcd/british2
```

```
drwxr-xr-x   - root              hadoop          0 2023-09-09 19:52
/tmp/dcd/ecobici
drwxr-xr-x   - root              hadoop          0 2023-09-23 15:24 /tmp/dcd/job
drwxr-xr-x   - root              hadoop          0 2023-09-23 01:14
/tmp/dcd/particion
drwxr-xr-x   - root              hadoop          0 2023-09-23 00:55
/tmp/dcd/pyspark
drwxr-xr-x   - sergio_ibarra1795 hadoop          0 2023-09-23 00:08
/tmp/dcd/sirilo
drwxr-xr-x   - sergio_ibarra1795 hadoop          0 2023-09-23 16:25
/tmp/dcd/streamdat
drwxr-xr-x   - sergio_ibarra1795 hadoop          0 2023-09-23 16:35
/tmp/dcd/streamdata
drwxr-xr-x   - sergio_ibarra1795 hadoop          0 2023-09-09 19:33
/tmp/dcd/wordcount
```

[15]:
```
CompletedProcess(args='hdfs dfs -ls /tmp/dcd', returncode=0)
```

[16]:
```
import subprocess

command = 'hdfs dfs -mkdir -p /tmp/dcd/Yemen/output'
subprocess.run(command, shell=True)
```

[16]:
```
CompletedProcess(args='hdfs dfs -mkdir -p /tmp/dcd/Yemen/output', returncode=0)
```

[17]:
```
import subprocess

command = 'hdfs dfs -ls /tmp/dcd/Yemen/'
subprocess.run(command, shell=True)
```

```
Found 2 items
drwxr-xr-x   - root hadoop          0 2023-09-28 19:20 /tmp/dcd/Yemen/input
drwxr-xr-x   - root hadoop          0 2023-09-28 19:23 /tmp/dcd/Yemen/output
```

[17]:
```
CompletedProcess(args='hdfs dfs -ls /tmp/dcd/Yemen/', returncode=0)
```

[ ]:
```

```

2. Cargar los archivos al cluster de HADOOP

[22]:
```
import subprocess

command = 'hdfs dfs -put Yemen_epidemiology_Data_Governorate_Level.csv /tmp/dcd/
↪Yemen/input/Yemen.csv'
subprocess.run(command, shell=True)
```

[22]:
```
CompletedProcess(args='hdfs dfs -put
Yemen_epidemiology_Data_Governorate_Level.csv /tmp/dcd/Yemen/input/Yemen.csv',
returncode=0)
```

[23]:
```
import subprocess

command = 'hdfs dfs -ls /tmp/dcd/Yemen/input'
subprocess.run(command, shell=True)
```

```
Found 1 items
-rw-r--r--   2 root hadoop     180155 2023-09-28 19:30
/tmp/dcd/Yemen/input/Yemen.csv
```

[23]:
```
CompletedProcess(args='hdfs dfs -ls /tmp/dcd/Yemen/input', returncode=0)
```

3. Crear un dataFrame en Spark

[24]:
```
Yemen_df = spark.read.csv('hdfs:///tmp/dcd/Yemen/input', header=True,␣
↪inferSchema=True)
```

[26]:
```
Yemen_df.show(10)
```

```
+-------------------+---------------+------+------+-------+--------------------
--+---------------+-------------+-------------+
|               Date|     Governorate| Cases|Deaths|CFR (%)|Attack Rate (per
1000)| COD Gov English|COD Gov Arabic|COD Gov Pcode|
+-------------------+---------------+------+------+-------+--------------------
--+---------------+-------------+-------------+
|2018-02-18 00:00:00|           Amran|103965|   176|   0.17|
89.582|           Amran|          |              |29
|2018-02-18 00:00:00|       Al Mahwit| 62887|   151|   0.24|
86.122|       Al Mahwit|          |              |27
|2018-02-18 00:00:00|      Al Dhale'e| 47136|    81|   0.17|
64.438|      Al Dhale'e|          |              |30
|2018-02-18 00:00:00|          Hajjah|121287|   422|   0.35|
52.06|          Hajjah|          |              |17
|2018-02-18 00:00:00|          Sana'a| 76250|   123|   0.16|
51.859|          Sana'a|          |              |23
|2018-02-18 00:00:00|          Dhamar|103214|   161|   0.16|
51.292|          Dhamar|          |              |20
|2018-02-18 00:00:00|           Abyan| 28243|    35|   0.12|
49.477|           Abyan|          |              |12
|2018-02-18 00:00:00|     Al Hudaydah|155908|   282|   0.18|
48.147|     Al Hudaydah|          |              |18
|2018-02-18 00:00:00|        Al Bayda| 30568|    36|   0.12|
40.253|        Al Bayda|          |              |14
|2018-02-18 00:00:00|Amanat Al Asimah|103184|    71|   0.07|
36.489|Amanat Al Asimah|          |              |13
+-------------------+---------------+------+------+-------+--------------------
--+---------------+-------------+-------------+
only showing top 10 rows
```

4. Registrar como tabla Spak SQL

[27]:
```
sub_Yemen_df = Yemen_df.dropDuplicates()
sub_Yemen_df
```

[27]:
```
DataFrame[Date: timestamp, Governorate: string, Cases: string, Deaths: int, CFR
(%): double, Attack Rate (per 1000): double, COD Gov English: string, COD Gov
Arabic: string, COD Gov Pcode: string]
```

[28]:
```
sub_Yemen_df.describe().show()
```

```
[Stage 5:>                                                          (0 + 1) / 1]

+-------+-----------+-----------------+----------------+------------------+---
------------------+--------------+-------------+------------------+
|summary|Governorate|            Cases|          Deaths|               CFR
(%)|Attack Rate (per 1000)|COD Gov English|COD Gov Arabic|     COD Gov Pcode|
+-------+-----------+-----------------+----------------+------------------+---
------------------+--------------+-------------+------------------+
|  count|       2914|             2914|            2914|              2914|
2914|           2914|          2914|              2914|
|   mean|       null|23727.266852812125|87.13143445435827|0.3832532601235424|
18.652564172958154|           null|          null|21.086988573534832|
| stddev|       null|26815.270334195033| 96.0375088723309|0.3807048764015296|
17.531846411491316|           null|          null| 6.189887041320621|
|    min|  AL Mahrah|            1,162|               0|               0.0|
0.0|           #N/A|          #N/A|              #N/A|
|    max|      Taizz|             9996|             422|               9.0|
89.582|          Taizz|          |                 |31
+-------+-----------+-----------------+----------------+------------------+---
------------------+--------------+-------------+------------------+
```

5. Generar consultas

5.1 Aquellas ciudades con mas casos y mas muertes

[35]:
```
sub_Yemen_df.filter("Cases>100 and Deaths>200").show(10)
```

```
+-------------------+-----------+-----+------+-------+----------------------+---
-----------+-------------+-------------+
|               Date|Governorate|Cases|Deaths|CFR (%)|Attack Rate (per 1000)|COD
Gov English|COD Gov Arabic|COD Gov Pcode|
+-------------------+-----------+-----+------+-------+----------------------+---
-----------+-------------+-------------+
|2017-06-27 00:00:00|     Hajjah|24580|   223|    9.0|                  11.1|
Hajjah|           |             |17
```

```
|2017-09-24 00:00:00|     Hajjah|80914|   398|   0.49|                34.731|
Hajjah|           |            |17
|2017-07-12 00:00:00|     Hajjah|35336|   338|    1.0|                 15.9|
Hajjah|           |            |17
|2017-06-29 00:00:00|     Hajjah|25335|   243|    1.0|                 11.4|
Hajjah|           |            |17
|2017-09-05 00:00:00|     Hajjah|67770|   386|   0.57|                29.089|
Hajjah|           |            |17
|2017-07-11 00:00:00|     Hajjah|35310|   338|    1.0|                 15.9|
Hajjah|           |            |17
|2017-09-03 00:00:00|        Ibb|42845|   262|   0.61|                14.489|
Ibb|            |            |11
|2017-09-20 00:00:00|        Ibb|47580|   269|   0.57|                 16.09|
Ibb|            |            |11
|2017-08-18 00:00:00|        Ibb|37347|   252|   0.67|                12.184|
Ibb|            |            |11
|2017-07-05 00:00:00|     Hajjah|30271|   308|    1.0|                 13.6|
Hajjah|           |            |17
+-------------------+-----------+-----+------+-------+----------------------+---
-----------+-------------+-------------+
only showing top 10 rows
```

5.2 Aquellas ciudades con mayor 'Attack Rate'

[36]:
```
sub_Yemen_df.orderBy(sub_Yemen_df['Attack Rate (per 1000)'].desc()).limit(10).
  ↪show()
```

```
+-------------------+-----------+------+------+-------+----------------------+--
-------------+-------------+-------------+
|               Date|Governorate| Cases|Deaths|CFR (%)|Attack Rate (per
1000)|COD Gov English|COD Gov Arabic|COD Gov Pcode|
+-------------------+-----------+------+------+-------+----------------------+--
-------------+-------------+-------------+
|2018-02-18 00:00:00|      Amran|103965|   176|   0.17|                89.582|
Amran|          |            |29
|2018-02-11 00:00:00|      Amran|103814|   176|   0.17|                89.452|
Amran|          |            |29
|2018-02-04 00:00:00|      Amran|103556|   176|   0.17|                89.229|
Amran|          |            |29
|2018-01-28 00:00:00|      Amran|103285|   176|   0.17|                88.996|
Amran|          |            |29
|2018-01-21 00:00:00|      Amran|102917|   175|    0.0|                88.679|
Amran|          |            |29
|2018-01-14 00:00:00|      Amran|102231|   175|   0.17|                88.088|
Amran|          |            |29
|2018-01-07 00:00:00|      Amran|101793|   174|   0.17|                 87.71|
Amran|          |            |29
|2017-12-31 00:00:00|      Amran|100981|   174|   0.17|                87.011|
```

```
Amran|         |              |29
|2018-02-18 00:00:00|  Al Mahwit| 62887|   151|   0.24|                86.122|
Al Mahwit|       |              |27
|2018-02-11 00:00:00|  Al Mahwit| 62606|   151|   0.24|                85.737|
Al Mahwit|       |              |27
+-------------------+-----------+------+------+-------+----------------------+--
-------------+--------------+-------------+
```

6. Guarda el nuevo DF en HDFS y en el Bucket

[41]:
```
sub_Yemen_df.filter("Cases<100 and Deaths<100").write.save("hdfs:///tmp/dcd/
↪Yemen/output1")
```

[ ]:
```
Escribimos en el GS bucket
```

[42]:
```
sub_Yemen_df.filter("Cases<100 and Deaths<100").write.format("csv").save("gs://
↪dcd05-sir-bucket/dcd/Yemen/output1")
```

[44]:
```
import subprocess

command = 'hdfs dfs -ls -R gs://dcd05-sir-bucket/dcd/Yemen/output1'
subprocess.run(command, shell=True)
```

```
-rwx------   3 root root          0 2023-09-28 20:12 gs://dcd05-sir-
bucket/dcd/Yemen/output1/_SUCCESS
-rwx------   3 root root       9086 2023-09-28 20:12 gs://dcd05-sir-bucket/dcd/Y
emen/output1/part-00000-07715bba-34b5-4494-84ad-4717d1467385-c000.csv
```

[44]:
```
CompletedProcess(args='hdfs dfs -ls -R gs://dcd05-sir-bucket/dcd/Yemen/output1',
returncode=0)
```

[ ]:

[ ]: