

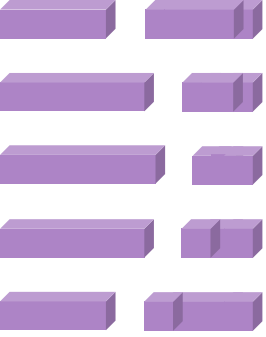


Diplomado en Ciencia de Datos UNAM

Modulo 13 Datos Masivos

Septiembre de 2023

Sergio Ibarra



Contenido

1. Descargar varios archivos referente a un tema en particular
2. Cargar los archivos al cluster de HADOOP
3. Crear un Data Frame en Spark
4. Registrar como tabla Spak SQL
5. Generar consultas
6. Guarda el nuevo DF en HDFS y en el Bucket

Descargar varios archivos referente a un tema en particular

carbide-trees-398419 > cluster-sir-dcd05

jupyter Diplomado_Mod13_Ejercicio2 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | PySpark

1. Descargar varios archivos referente a un tema en particular

In [17]:

```
import pandas as pd

dfc = pd.read_csv("https://gitlab.com/dgtic5/res/-/raw/main/aprendizajeSupervizado/Yemen%20Cholera%20Outbreak%20Epidemiology%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)
```

In [18]: dfc

Out[18]:

	Date	Governorate	Cases	Deaths	CFR (%)	Attack Rate (per 1000)	COD Gov English	COD Gov Arabic	COD Gov Pcode
0	2018-02-18	Amran	103965	176	0.17	89.582	Amran	امران	29
1	2018-02-18	Al Mahwit	62887	151	0.24	86.122	Al Mahwit	المحويت	27
2	2018-02-18	Al Dhale'e	47136	81	0.17	64.438	Al Dhale'e	الضالع	30
3	2018-02-18	Hajjah	121287	422	0.35	52.060	Hajjah	حجة	17

carbide-trees-398419 > cluster-sir-dcd05

jupyter Diplomado_Mod13_Ejercicio2 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | PySpark

In [10]:

```
import urllib.request

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

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

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

In [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/

Cargar los archivos al cluster de HADOOP

```
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
```

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

In [16]: `import subprocess`

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

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

In [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
```

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

2. Cargar los archivos al cluster de HADOOP

In [22]: `import subprocess`

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

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

In [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
```

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

Crear un Data Frame en Spark

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

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

3. Crear un dataframe en Spark

In [24]: Yemen_df = spark.read.csv('hdfs:///tmp/dcd/Yemen/input', header=True, in

In [25]: Yemen_df

Out[25]: 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]
```

```
In [24]: Yemen_df = spark.read.csv('hdfs:///tmp/dcd/Yemen/input', header=True, in
```

```
In [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	عمران		
2018-02-18 00:00:00	Al Mahwit	62887	151	0.24	
86.122	Al Mahwit	27	المحويت		
2018-02-18 00:00:00	Al Dhale'e	47136	81	0.17	
64.438	Al Dhale'e	30	الضالع		
2018-02-18 00:00:00	Hajjah	121287	422	0.35	
52.06	Hajjah	17	حجة		
2018-02-18 00:00:00	Sana'a	76250	123	0.16	
51.859	Sana'a	23	صنعاء		
2018-02-18 00:00:00	Dhamar	103214	161	0.16	
51.292	Dhamar	20	ذمار		
2018-02-18 00:00:00	Abyan	28243	35	0.12	
49.477	Abyan	12	أبين		
2018-02-18 00:00:00	Al Hudaydah	155908	282	0.18	
48.147	Al Hudaydah	18	الحديدة		
2018-02-18 00:00:00	Al Baydal	30568	36	0.12	

Registrar como tabla Spak SQL

4. Registrar como tabla Spak SQL

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

```
Out[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]
```

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

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

	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	6.18988704

Generar consultas

5. Generar consultas

5.1 Aquellas ciudades con mas casos y mas muertes

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

only showing top 10 rows

5.2 Aquellas ciudades con mayor 'Attack Rate'

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

--	--	--	--	--	--	--	--	--

5.2 Aquellas ciudades con mayor 'Attack Rate'

```
In [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	عمران
2018-02-11 00:00:00	Amran	103814	176	0.17	89.452	Amran	29	عمران
2018-02-04 00:00:00	Amran	103556	176	0.17	89.229	Amran	29	عمران
2018-01-28 00:00:00	Amran	103285	176	0.17	88.996	Amran	29	عمران
2018-01-21 00:00:00	Amran	102917	175	0.0	88.679	Amran	29	عمران
2018-01-14 00:00:00	Amran	102231	175	0.17	88.088	Amran	29	عمران
2018-01-07 00:00:00	Amran	101793	174	0.17	87.71	Amran	29	عمران
2017-12-31 00:00:00	Amran	100981	174	0.17	87.011	Amran	29	عمران
2018-02-18 00:00:00	Al Mahwit	62887	151	0.24	86.122	Al Mahwit	27	المحويت
2018-02-11 00:00:00	Al Mahwit	62606	151	0.24	85.737	Al Mahwit	27	المحويت

Guarda el nuevo DF en HDFS y en el Bucket

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

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

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

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

```
In [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/Yemen/output1/part-00000-07715bba-34b5-4494-84ad
-4717d1467385-c000.csv
```

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

```
In [ ]:
```

```
In [ ]:
```