2.1 Columnas:

Google Colab

```
1 df.columns
```

```
['fecha reporte web',
 'ID de caso',
 'Fecha de notificación',
 'Código DIVIPOLA departamento',
 'Nombre departamento',
 'Código DIVIPOLA municipio',
 'Nombre municipio',
 'Edad',
 'Unidad de medida de edad',
 'Sexo',
 'Tipo de contagio',
 'Ubicación del caso',
 'Estado',
 'Código ISO del país',
 'Nombre del país',
 'Recuperado',
 'Fecha de inicio de síntomas',
 'Fecha de muerte',
 'Fecha de diagnóstico',
 'Fecha de recuperación',
 'Tipo de recuperación',
 'Pertenencia étnica',
 'Nombre del grupo étnico']
```

JupyterHub

```
In [5]: df.columns

['fecha reporte web', 'ID de caso', 'Fecha de notificación', 'Código DIVIPOLA departamento', 'Nombre departamento', 'Código DIV
IPOLA municipio', 'Nombre municipio', 'Edad', 'Unidad de medida de edad', 'Sexo', 'Tipo de contagio', 'Ubicación del caso', 'Es
tado', 'Código ISO del país', 'Nombre del país', 'Recuperado', 'Fecha de inicio de síntomas', 'Fecha de muerte', 'Fecha de diag
nóstico', 'Fecha de recuperación', 'Tipo de recuperación', 'Pertenencia étnica', 'Nombre del grupo étnico']
```

2.2 Tipos de datos:

Google Colab

```
1 df.printSchema()
```

```
root
 |-- fecha reporte web: string (nullable = true)
 |-- ID de caso: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- Nombre departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- Nombre municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Nombre del grupo étnico: string (nullable = true)
```

JupyterHub

```
In [6]: df.printSchema()

root
 |-- fecha reporte web: string (nullable = true)
 |-- ID de caso: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- Nombre departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- Nombre municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Nombre del grupo étnico: string (nullable = true)
```

2.3 Seleccionando columnas

GoogleColab

```
1 df.select('fecha reporte web', 'ID de caso', 'Sexo', 'Tipo de contagio').show(10, False)

+-----------------+----------+----+----------------+
|fecha reporte web|ID de caso|Sexo|Tipo de contagio|
+-----------------+----------+----+----------------+
|6/3/2020 0:00:00 |1         |F   |Importado       |
|9/3/2020 0:00:00 |2         |M   |Importado       |
|9/3/2020 0:00:00 |3         |F   |Importado       |
|11/3/2020 0:00:00|4         |M   |Relacionado     |
|11/3/2020 0:00:00|5         |M   |Relacionado     |
|11/3/2020 0:00:00|6         |F   |Relacionado     |
|11/3/2020 0:00:00|7         |F   |Importado       |
|11/3/2020 0:00:00|8         |F   |Importado       |
|11/3/2020 0:00:00|9         |F   |Importado       |
|12/3/2020 0:00:00|10        |F   |Importado       |
+-----------------+----------+----+----------------+
only showing top 10 rows
```

JupyterHub

```
In [7]: df.select('fecha reporte web', 'ID de caso', 'Sexo', 'Tipo de contagio').show(10, False)

+-----------------+----------+----+----------------+
|fecha reporte web|ID de caso|Sexo|Tipo de contagio|
+-----------------+----------+----+----------------+
|6/3/2020 0:00:00 |1         |F   |Importado       |
|9/3/2020 0:00:00 |2         |M   |Importado       |
|9/3/2020 0:00:00 |3         |F   |Importado       |
|11/3/2020 0:00:00|4         |M   |Relacionado     |
|11/3/2020 0:00:00|5         |M   |Relacionado     |
|11/3/2020 0:00:00|6         |F   |Relacionado     |
|11/3/2020 0:00:00|7         |F   |Importado       |
|11/3/2020 0:00:00|8         |F   |Importado       |
|11/3/2020 0:00:00|9         |F   |Importado       |
|12/3/2020 0:00:00|10        |F   |Importado       |
+-----------------+----------+----+----------------+
only showing top 10 rows
```

2.4 Renombrando Columnas

GoogleColab

```
1 df=df.withColumnRenamed('fecha reporte web', 'fecha_reporte')

1 df=df.withColumnsRenamed({'ID de caso': 'ID', 'Nombre del país': 'pais', 'Nombre departamento': 'departamento', 'Nombre municipio': 'municipio', 'Ubicación del caso': 'Ubicación'})
```

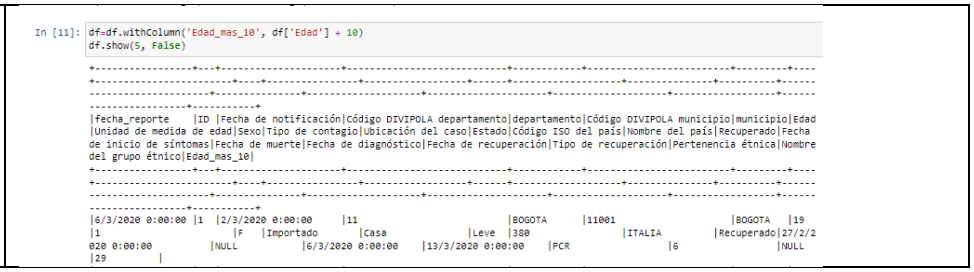
JupyterHub

```
In [8]: df=df.withColumnRenamed('fecha reporte web', 'fecha_reporte')

In [9]: df=df.withColumnsRenamed({'ID de caso': 'ID', 'Nombre del país': 'pais', 'Nombre departamento': 'departamento', '

In [10]: df.printSchema()

root
 |-- fecha_reporte: string (nullable = true)
 |-- ID: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Nombre del grupo étnico: string (nullable = true)
```

2.5 Añadiendo columnas

Google Colab

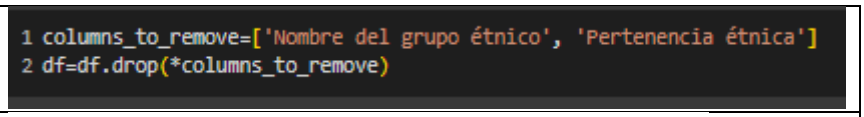
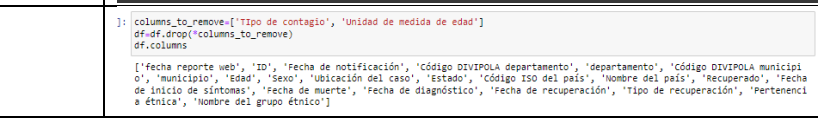
```
1 df=df.withColumn('Edad_mas_10', df['Edad'] + 10)
2 df.show(5, False)

o|Ubicación|Estado|Código ISO del país|pais  |Recuperado|Fecha de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperación|Pertenencia étnica|Nombre del grupo étnico|Edad_mas_10
 |Casa     |Leve  |380                |ITALIA|Recuperado|27/2/2020 0:00:00          |NULL           |6/3/2020 0:00:00    |13/3/2020 0:00:00    |PCR                 |6                 |NULL                   |29
 |Casa     |Leve  |724                |ESPAÑA|Recuperado|4/3/2020 0:00:00           |NULL           |9/3/2020 0:00:00    |19/3/2020 0:00:00    |PCR                 |5                 |NULL                   |44
 |Casa     |Leve  |724                |ESPAÑA|Recuperado|29/2/2020 0:00:00          |NULL           |9/3/2020 0:00:00    |15/3/2020 0:00:00    |PCR                 |6                 |NULL                   |60
 |Casa     |Leve  |NULL               |NULL  |Recuperado|6/3/2020 0:00:00           |NULL           |11/3/2020 0:00:00   |26/3/2020 0:00:00    |PCR                 |6                 |NULL                   |65
 |Casa     |Leve  |NULL               |NULL  |Recuperado|8/3/2020 0:00:00           |NULL           |11/3/2020 0:00:00   |23/3/2020 0:00:00    |PCR                 |6                 |NULL                   |35
```

JupyterHub	

```
In [11]: df=df.withColumn('Edad_mas_10', df['Edad'] + 10)
         df.show(5, False)
```

```
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+
|fecha_reporte   |ID |Fecha de notificación|Código DIVIPOLA departamento|departamento|Código DIVIPOLA municipio|municipio|Edad|
|Unidad de medida de edad|Sexo|Tipo de contagio|Ubicación del caso|Estado|Código ISO del país|Nombre del país|Recuperado|Fecha
de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperación|Pertenencia étnica|Nombre
del grupo étnico|Edad_mas_10|
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+
|6/3/2020 0:00:00 |1  |2/3/2020 0:00:00    |11                         |BOGOTA     |11001                  |BOGOTA   |19  |
|1                |F  |Importado           |Casa               |Leve |380                |ITALIA   |Recuperado|27/2/2
020 0:00:00       |NULL          |6/3/2020 0:00:00   |13/3/2020 0:00:00   |PCR               |6               |NULL
|29        |
```

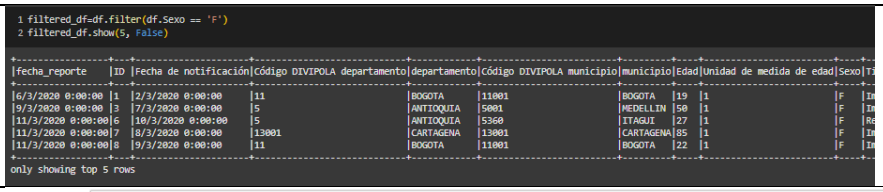
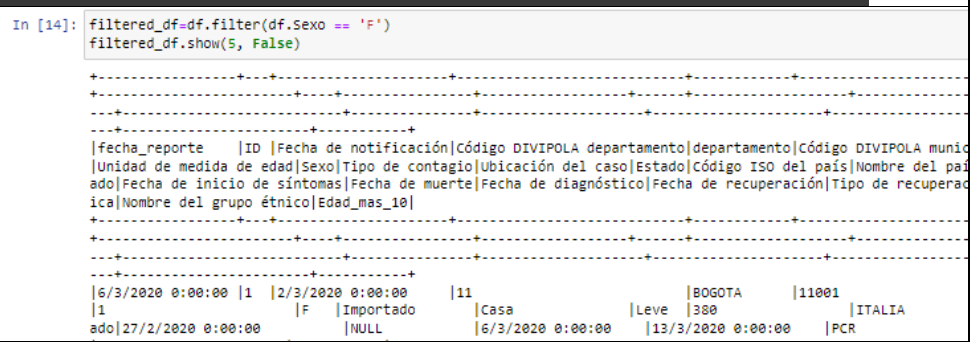
2.6 Eliminando columnas

GoogleColab	
JupyterHub	

```
1 columns_to_remove=['Nombre del grupo étnico', 'Pertenencia étnica']
2 df=df.drop(*columns_to_remove)
```

```
]: columns_to_remove=['Tipo de contagio', 'Unidad de medida de edad']
   df=df.drop(*columns_to_remove)
   df.columns

   ['fecha reporte web', 'ID', 'Fecha de notificación', 'Código DIVIPOLA departamento', 'departamento', 'Código DIVIPOLA municipi
   o', 'municipio', 'Edad', 'Sexo', 'Ubicación del caso', 'Estado', 'Código ISO del país', 'Nombre del país', 'Recuperado', 'Fecha
   de inicio de síntomas', 'Fecha de muerte', 'Fecha de diagnóstico', 'Fecha de recuperación', 'Tipo de recuperación', 'Pertenenci
   a étnica', 'Nombre del grupo étnico']
```

2.7 Filtrando datos

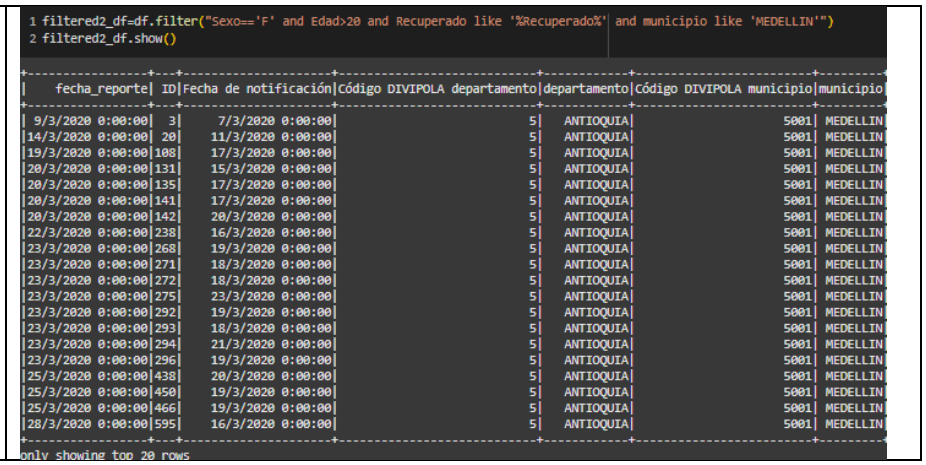
Google Colab	
JupyterHub	

```
1 filtered_df=df.filter(df.Sexo == 'F')
2 filtered_df.show(5, False)
```

```
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+----------------------+----+
|fecha_reporte   |ID |Fecha de notificación|Código DIVIPOLA departamento|departamento|Código DIVIPOLA municipio|municipio|Edad|Unidad de medida de edad|Sexo|T
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+----------------------+----+--
|6/3/2020 0:00:00 |1  |2/3/2020 0:00:00    |11                         |BOGOTA     |11001                  |BOGOTA   |19  |1                     |F   |Im
|9/3/2020 0:00:00 |3  |7/3/2020 0:00:00    |5                          |ANTIOQUIA  |5001                   |MEDELLIN |50  |1                     |F   |Im
|11/3/2020 0:00:00|6  |10/3/2020 0:00:00   |5                          |ANTIOQUIA  |5360                   |ITAGUI   |27  |1                     |F   |Re
|11/3/2020 0:00:00|7  |8/3/2020 0:00:00    |13001                      |CARTAGENA  |13001                  |CARTAGENA|85  |1                     |F   |Im
|11/3/2020 0:00:00|8  |9/3/2020 0:00:00    |11                         |BOGOTA     |11001                  |BOGOTA   |22  |1                     |F   |Im
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+----------------------+----+--
only showing top 5 rows
```

```
In [14]: filtered_df=df.filter(df.Sexo == 'F')
         filtered_df.show(5, False)
```

```
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+
|fecha_reporte   |ID |Fecha de notificación|Código DIVIPOLA departamento|departamento|Código DIVIPOLA munic
|Unidad de medida de edad|Sexo|Tipo de contagio|Ubicación del caso|Estado|Código ISO del país|Nombre del pai
ado|Fecha de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperac
ica|Nombre del grupo étnico|Edad_mas_10|
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+----+
|6/3/2020 0:00:00 |1  |2/3/2020 0:00:00    |11                         |BOGOTA     |11001
|1                |F  |Importado           |Casa               |Leve |380                |ITALIA
ado|27/2/2020 0:00:00          |NULL          |6/3/2020 0:00:00   |13/3/2020 0:00:00   |PCR
```

Google Colab	

```
1 filtered2_df=df.filter("Sexo=='F' and Edad>20 and Recuperado like '%Recuperado%' and municipio like 'MEDELLIN'")
2 filtered2_df.show()
```

```
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+
|    fecha_reporte| ID|Fecha de notificación|Código DIVIPOLA departamento|departamento|Código DIVIPOLA municipio|municipio|
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+
| 9/3/2020 0:00:00|  3|     7/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|14/3/2020 0:00:00| 20|    11/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|19/3/2020 0:00:00|108|    17/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|20/3/2020 0:00:00|131|    15/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|20/3/2020 0:00:00|135|    17/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|20/3/2020 0:00:00|141|    17/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|20/3/2020 0:00:00|142|    20/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|22/3/2020 0:00:00|238|    16/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|268|    19/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|271|    18/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|272|    18/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|275|    23/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|292|    19/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|293|    18/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|294|    21/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|23/3/2020 0:00:00|296|    19/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|25/3/2020 0:00:00|438|    20/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|25/3/2020 0:00:00|450|    19/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|25/3/2020 0:00:00|466|    19/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
|28/3/2020 0:00:00|595|    16/3/2020 0:00:00|                          5|   ANTIOQUIA|                   5001| MEDELLIN|
+----------------+---+--------------------+---------------------------+-----------+-----------------------+---------+
only showing top 20 rows
```

JupyterHub

```
filtered2_df=df.filter("Sexo=='F' and Edad>20 and Recuperado like '%Recuperado%' and municipio like 'MEDELLIN'")
filtered2_df.show()
```

```
|   fecha_reporte| ID|Fecha de notificación|Código DIVIPOLA departamento|departamento|Código DIVIPOLA municipio|municipio|Edad
|Unidad de medida de edad|Sexo|Tipo de contagio|Ubicación del caso|Estado|Código ISO del país|     Nombre del país|Recuperado|F
echa de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperación|Pertenencia étnica|N
ombre del grupo étnico|Edad_mas_10|
| 9/3/2020 0:00:00|  3|     7/3/2020 0:00:00|                           5|   ANTIOQUIA|                     5001| MEDELLIN|  50
|                       1|   F|       Importado|              Casa|  Leve|                724|              ESPAÑA|Recuperado|
29/2/2020 0:00:00|           NULL|    9/3/2020 0:00:00|    15/3/2020 0:00:00|                 PCR|                 6|
NULL|         60|
|14/3/2020 0:00:00| 20|    11/3/2020 0:00:00|                           5|   ANTIOQUIA|                     5001| MEDELLIN|  26
|                       1|   F|     Relacionado|              Casa|  Leve|               NULL|                NULL|Recuperado|
9/3/2020 0:00:00|          NULL|   14/3/2020 0:00:00|    24/3/2020 0:00:00|                 PCR|                 6|
NULL|         36|
|19/3/2020 0:00:00|108|    17/3/2020 0:00:00|                           5|   ANTIOQUIA|                     5001| MEDELLIN|  57
|                       1|   F|       Importado|              Casa|  Leve|                724|              ESPAÑA|Recuperado|
14/3/2020 0:00:00|          NULL|   19/3/2020 0:00:00|    28/3/2020 0:00:00|                 PCR|                 6|
NULL|         67|
|20/3/2020 0:00:00|131|    15/3/2020 0:00:00|                           5|   ANTIOQUIA|                     5001| MEDELLIN|  22
|                       1|   F|       Importado|              Casa|  Leve|                724|              ESPAÑA|Recuperado|
13/3/2020 0:00:00|          NULL|   20/3/2020 0:00:00|    28/3/2020 0:00:00|                 PCR|                 6|
NULL|         32|
|20/3/2020 0:00:00|135|    17/3/2020 0:00:00|                           5|   ANTIOQUIA|                     5001| MEDELLIN|  44
|                       1|   F|       Importado|              Casa|  Leve|                276|             ALEMANIA|Recuperado|
16/3/2020 0:00:00|          NULL|   20/3/2020 0:00:00|     1/4/2020 0:00:00|                 PCR|                 6|
NULL|         54|
|20/3/2020 0:00:00|141|    17/3/2020 0:00:00|                           5|   ANTIOQUIA|                     5001| MEDELLIN|  62
|                       1|   F|       Importado|              Casa|  Leve|                191|             CROACIA|Recuperado|
```

2.8 Usando lambda

Google Colab

```
1 age_udf=udf(lambda age: "kid" if age < 18 else ('young adult' if age<=30 else 'senior'), StringType())
2 df=df.withColumn("age_group", age_udf(df.Edad))
3 df.filter(df['age_group']=='young adult').orderBy('Edad', ascending=False).show(10, False)
```

```
stado|Código ISO del país|Nombre del país   |Recuperado|Fecha de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperación|Pertenencia étnica|Nombre del grupo étnico|age_group  |
eve  |NULL   |NULL    |Recuperado|9/6/2020 0:00:00  |NULL |19/6/2020 0:00:00 |12/7/2020 0:00:00 |PCR    |6 |NULL |young adult|
eve  |NULL   |NULL    |Recuperado|17/3/2020 0:00:00 |NULL |21/3/2020 0:00:00 |2/4/2020 0:00:00  |PCR    |6 |NULL |young adult|
eve  |NULL   |NULL    |Recuperado|14/6/2020 0:00:00 |NULL |19/6/2020 0:00:00 |23/7/2020 0:00:00 |Tiempo |5 |NULL |young adult|
eve  |NULL   |NULL    |Recuperado|17/3/2020 0:00:00 |NULL |27/3/2020 0:00:00 |15/4/2020 0:00:00 |PCR    |6 |NULL |young adult|
eve  |NULL   |NULL    |Recuperado|8/6/2020 0:00:00  |NULL |19/6/2020 0:00:00 |11/7/2020 0:00:00 |PCR    |6 |NULL |young adult|
eve  |724    |ESPAÑA  |Recuperado|15/3/2020 0:00:00 |NULL |22/3/2020 0:00:00 |25/3/2020 0:00:00 |PCR    |6 |NULL |young adult|
eve  |NULL   |NULL    |Recuperado|11/6/2020 0:00:00 |NULL |19/6/2020 0:00:00 |14/7/2020 0:00:00 |Tiempo |6 |NULL |young adult|
eve  |724    |ESPAÑA  |Recuperado|9/3/2020 0:00:00  |NULL |13/3/2020 0:00:00 |23/3/2020 0:00:00 |PCR    |6 |NULL |young adult|
eve  |NULL   |NULL    |Recuperado|3/6/2020 0:00:00  |NULL |19/6/2020 0:00:00 |6/7/2020 0:00:00  |PCR    |6 |NULL |young adult|
eve  |214    |REPÚBLICA DOMINICANA|Recuperado|16/3/2020 0:00:00 |NULL |23/3/2020 0:00:00 |31/3/2020 0:00:00 |PCR |6 |NULL |young adult|
```

JupyterHub

```
age_udf=udf(lambda age: "kid" if age < 18 else ('young adult' if age<=30 else 'senior'), StringType())
df=df.withColumn("age_group", age_udf(df.Edad))
df.filter(df['age_group']=='young adult').orderBy('Edad', ascending=False).show(10, False)
```

```
|fecha_reporte     |ID   |Fecha de notificación|Código DIVIPOLA departamento|departamento|Código DIVIPOLA municipio|municipio
|Edad|Unidad de medida de edad|Sexo|Tipo de contagio|Ubicación del caso|Estado|Código ISO del país|Nombre del país     |Recuper
ado|Fecha de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperación|Pertenencia étn
ica|Nombre del grupo étnico|Edad_mas_10|age_group  |
|19/6/2020 0:00:00|62406|10/6/2020 0:00:00    |11                          |BOGOTA      |11001                    |BOGOTA
|30  |1                       |F   |Relacionado     |Casa              |Leve  |NULL               |NULL                |Recuper
ado|9/6/2020 0:00:00           |NULL           |19/6/2020 0:00:00   |12/7/2020 0:00:00    |PCR                 |6
|NULL                   |40         |young adult|
|21/3/2020 0:00:00|192  |20/3/2020 0:00:00    |76                          |VALLE       |76001                    |CALI
|30  |1                       |F   |Relacionado     |Casa              |Leve  |NULL               |NULL                |Recuper
ado|17/3/2020 0:00:00          |NULL           |21/3/2020 0:00:00   |2/4/2020 0:00:00     |PCR                 |6
|NULL                   |40         |young adult|
|19/6/2020 0:00:00|62398|17/6/2020 0:00:00    |11                          |BOGOTA      |11001                    |BOGOTA
|30  |1                       |M   |Relacionado     |Casa              |Leve  |NULL               |NULL                |Recuper
ado|14/6/2020 0:00:00          |NULL           |19/6/2020 0:00:00   |23/7/2020 0:00:00    |Tiempo              |5
|NULL                   |40         |young adult|
|27/3/2020 0:00:00|517  |23/3/2020 0:00:00    |11                          |BOGOTA      |11001                    |BOGOTA
|30  |1                       |F   |Relacionado     |Casa              |Leve  |NULL               |NULL                |Recuper
ado|17/3/2020 0:00:00          |NULL           |27/3/2020 0:00:00   |15/4/2020 0:00:00    |PCR                 |6
|NULL                   |40         |young adult|
|19/6/2020 0:00:00|62380|10/6/2020 0:00:00    |11                          |BOGOTA      |11001                    |BOGOTA
|30  |1                       |F   |Comunitaria     |Casa              |Leve  |NULL               |NULL                |Recuper
ado|8/6/2020 0:00:00           |NULL           |19/6/2020 0:00:00   |11/7/2020 0:00:00    |PCR                 |6
```

Punto 3 en Google Colab

3.1

Spark Dataframes

```
1 df.groupBy('departamento').count().orderBy('count', ascending=False).show(10, False)
```

```
+------------+-----+
|departamento|count|
+------------+-----+
|BOGOTA      |30016|
|BARRANQUILLA|13065|
|ATLANTICO   |10994|
|VALLE       |10404|
|CARTAGENA   |8333 |
|ANTIOQUIA   |4554 |
|NARIÑO      |3520 |
|CUNDINAMARCA|2827 |
|AMAZONAS    |2317 |
|CHOCO       |1636 |
+------------+-----+
only showing top 10 rows
```

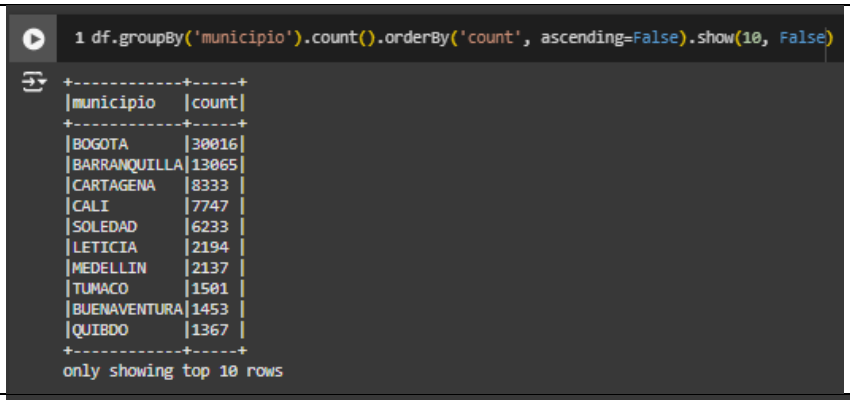
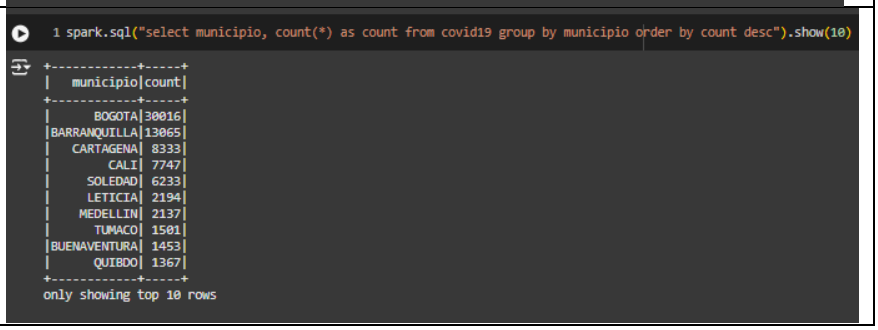
SparkSQL	

```
[ ]  1 df.createOrReplaceTempView("covid19")
```

```
 1 spark.sql("select departamento, count(*) as count from covid19 group by departamento order by count desc").show(10)
```

```
+------------+-----+
|departamento|count|
+------------+-----+
|      BOGOTA|30016|
|BARRANQUILLA|13065|
|   ATLANTICO|10994|
|       VALLE|10404|
|   CARTAGENA| 8333|
|   ANTIOQUIA| 4554|
|      NARIÑO| 3520|
|CUNDINAMARCA| 2827|
|    AMAZONAS| 2317|
|       CHOCO| 1636|
+------------+-----+
only showing top 10 rows
```

3.2

Spark Dataframes	
Spark SQL	

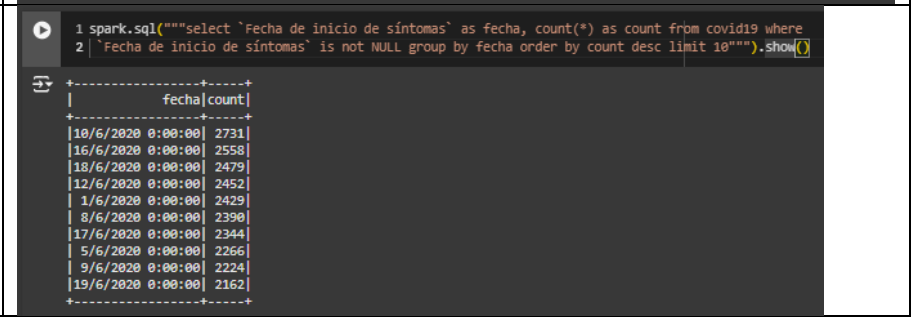
```
 1 df.groupBy('municipio').count().orderBy('count', ascending=False).show(10, False)
```

```
+------------+-----+
|municipio   |count|
+------------+-----+
|BOGOTA      |30016|
|BARRANQUILLA|13065|
|CARTAGENA   |8333 |
|CALI        |7747 |
|SOLEDAD     |6233 |
|LETICIA     |2194 |
|MEDELLIN    |2137 |
|TUMACO      |1501 |
|BUENAVENTURA|1453 |
|QUIBDO      |1367 |
+------------+-----+
only showing top 10 rows
```

```
 1 spark.sql("select municipio, count(*) as count from covid19 group by municipio order by count desc").show(10)
```

```
+------------+-----+
|   municipio|count|
+------------+-----+
|      BOGOTA|30016|
|BARRANQUILLA|13065|
|   CARTAGENA| 8333|
|        CALI| 7747|
|     SOLEDAD| 6233|
|     LETICIA| 2194|
|    MEDELLIN| 2137|
|      TUMACO| 1501|
|BUENAVENTURA| 1453|
|      QUIBDO| 1367|
+------------+-----+
only showing top 10 rows
```

3.3

Dataframes	
SparkSQL	

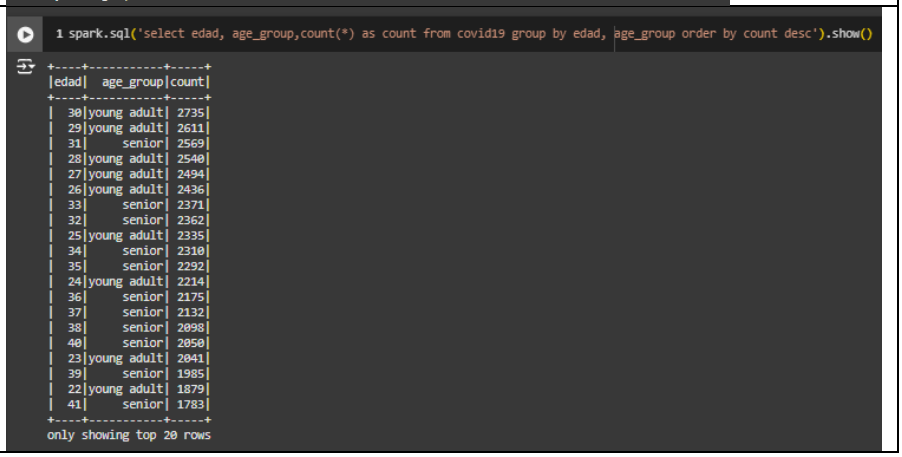
```
 1 df.dropna(subset=['Fecha de inicio de síntomas']).groupBy('Fecha de inicio de síntomas').count().orderBy('count', ascending=False).show(10, False)
```

```
+-------------------------+-----+
|Fecha de inicio de síntomas|count|
+-------------------------+-----+
|10/6/2020 0:00:00        |2731 |
|16/6/2020 0:00:00        |2558 |
|18/6/2020 0:00:00        |2479 |
|12/6/2020 0:00:00        |2452 |
|1/6/2020 0:00:00         |2429 |
|8/6/2020 0:00:00         |2390 |
|17/6/2020 0:00:00        |2344 |
|5/6/2020 0:00:00         |2266 |
|9/6/2020 0:00:00         |2224 |
|19/6/2020 0:00:00        |2162 |
+-------------------------+-----+
only showing top 10 rows
```

```
 1 spark.sql("""select `Fecha de inicio de síntomas` as fecha, count(*) as count from covid19 where
 2 `Fecha de inicio de síntomas` is not NULL group by fecha order by count desc limit 10""").show()
```

```
+-----------------+-----+
|            fecha|count|
+-----------------+-----+
|10/6/2020 0:00:00| 2731|
|16/6/2020 0:00:00| 2558|
|18/6/2020 0:00:00| 2479|
|12/6/2020 0:00:00| 2452|
| 1/6/2020 0:00:00| 2429|
| 8/6/2020 0:00:00| 2390|
|17/6/2020 0:00:00| 2344|
| 5/6/2020 0:00:00| 2266|
| 9/6/2020 0:00:00| 2224|
|19/6/2020 0:00:00| 2162|
+-----------------+-----+
```

3.4

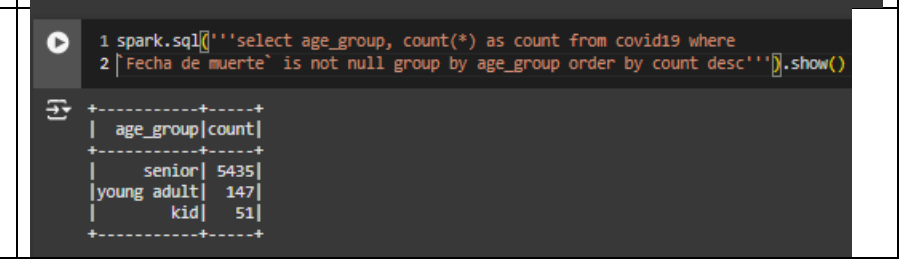
Dataframes	
SparkSQL	

```
1 df.select('edad', 'age_group', 'ID').groupBy('edad', 'age_group').count().orderBy('count', ascending=False).show()

+----+-----------+-----+
|edad|  age_group|count|
+----+-----------+-----+
|  30|young adult| 2735|
|  29|young adult| 2611|
|  31|     senior| 2569|
|  28|young adult| 2540|
|  27|young adult| 2494|
|  26|young adult| 2436|
|  33|     senior| 2371|
|  32|     senior| 2362|
|  25|young adult| 2335|
|  34|     senior| 2310|
|  35|     senior| 2292|
|  24|young adult| 2214|
|  36|     senior| 2175|
|  37|     senior| 2132|
|  38|     senior| 2098|
|  40|     senior| 2050|
|  23|young adult| 2041|
|  39|     senior| 1985|
|  22|young adult| 1879|
|  41|     senior| 1783|
+----+-----------+-----+
only showing top 20 rows
```

```
1 spark.sql('select edad, age_group,count(*) as count from covid19 group by edad, age_group order by count desc').show()

+----+-----------+-----+
|edad|  age_group|count|
+----+-----------+-----+
|  30|young adult| 2735|
|  29|young adult| 2611|
|  31|     senior| 2569|
|  28|young adult| 2540|
|  27|young adult| 2494|
|  26|young adult| 2436|
|  33|     senior| 2371|
|  32|     senior| 2362|
|  25|young adult| 2335|
|  34|     senior| 2310|
|  35|     senior| 2292|
|  24|young adult| 2214|
|  36|     senior| 2175|
|  37|     senior| 2132|
|  38|     senior| 2098|
|  40|     senior| 2050|
|  23|young adult| 2041|
|  39|     senior| 1985|
|  22|young adult| 1879|
|  41|     senior| 1783|
+----+-----------+-----+
only showing top 20 rows
```

3.5

Muertes por grupo de edad

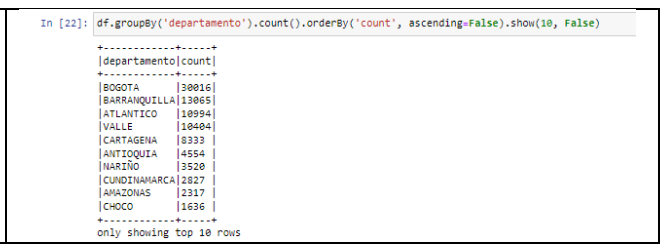
Dataframes	
SparkSQL	

```
[31] 1 from pyspark.sql.functions import col
     2 df.filter(col('Fecha de muerte').isNotNull()).groupBy('age_group').count().orderBy('count', ascending=False).show()

+-----------+-----+
|  age_group|count|
+-----------+-----+
|     senior| 5435|
|young adult|  151|
|        kid|   47|
+-----------+-----+
```

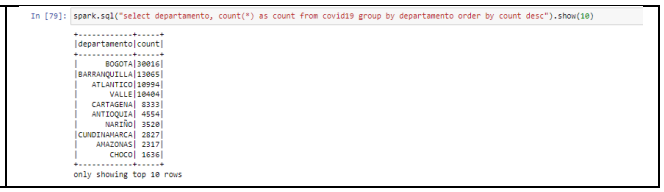
```
1 spark.sql('''select age_group, count(*) as count from covid19 where
2 `Fecha de muerte` is not null group by age_group order by count desc''').show()

+-----------+-----+
|  age_group|count|
+-----------+-----+
|     senior| 5435|
|young adult|  147|
|        kid|   51|
+-----------+-----+
```

Punto 3 JupyterHub

3.1

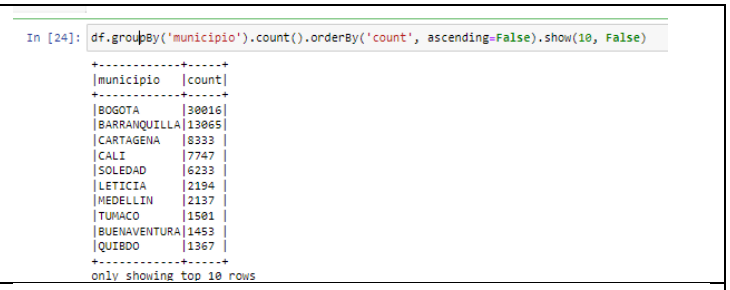
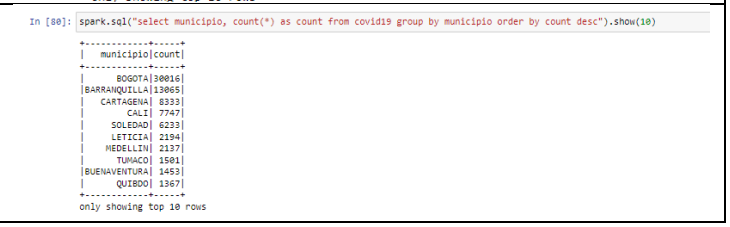
Spark Dataframes	

```
In [22]: df.groupBy('departamento').count().orderBy('count', ascending=False).show(10, False)

+------------+-----+
|departamento|count|
+------------+-----+
|BOGOTA      |30016|
|BARRANQUILLA|13065|
|ATLANTICO   |10994|
|VALLE       |10404|
|CARTAGENA   |8333 |
|ANTIOQUIA   |4554 |
|NARIÑO      |3520 |
|CUNDINAMARCA|2827 |
|AMAZONAS    |2317 |
|CHOCO       |1636 |
+------------+-----+
only showing top 10 rows
```

SparkSQL	

```
In [79]: spark.sql("select departamento, count(*) as count from covid19 group by departamento order by count desc").show(10)

+------------+-----+
|departamento|count|
+------------+-----+
|      BOGOTA|30016|
|BARRANQUILLA|13065|
|   ATLANTICO|10994|
|       VALLE|10404|
|   CARTAGENA| 8333|
|   ANTIOQUIA| 4554|
|      NARIÑO| 3520|
|CUNDINAMARCA| 2827|
|    AMAZONAS| 2317|
|       CHOCO| 1636|
+------------+-----+
only showing top 10 rows
```

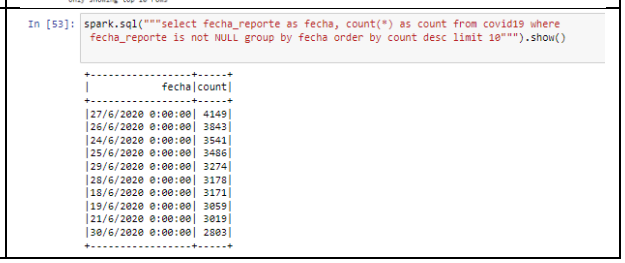
3.2

Spark Dataframes	
Spark SQL	

```
In [24]: df.groupBy('municipio').count().orderBy('count', ascending=False).show(10, False)

+------------+-----+
|municipio   |count|
+------------+-----+
|BOGOTA      |30016|
|BARRANQUILLA|13065|
|CARTAGENA   |8333 |
|CALI        |7747 |
|SOLEDAD     |6233 |
|LETICIA     |2194 |
|MEDELLIN    |2137 |
|TUMACO      |1501 |
|BUENAVENTURA|1453 |
|QUIBDO      |1367 |
+------------+-----+
only showing top 10 rows
```

```
In [80]: spark.sql("select municipio, count(*) as count from covid19 group by municipio order by count desc").show(10)

+------------+-----+
|   municipio|count|
+------------+-----+
|      BOGOTA|30016|
|BARRANQUILLA|13065|
|   CARTAGENA| 8333|
|        CALI| 7747|
|     SOLEDAD| 6233|
|     LETICIA| 2194|
|    MEDELLIN| 2137|
|      TUMACO| 1501|
|BUENAVENTURA| 1453|
|      QUIBDO| 1367|
+------------+-----+
only showing top 10 rows
```

3.3

Dataframes	
SparkSQL	

```
In [71]: df.dropna(subset=['fecha_reporte']).groupBy('fecha_reporte').count().orderBy('count', ascending=False).show(10, False)

+-----------------+-----+
|fecha_reporte    |count|
+-----------------+-----+
|27/6/2020 0:00:00|4149 |
|26/6/2020 0:00:00|3843 |
|24/6/2020 0:00:00|3541 |
|25/6/2020 0:00:00|3486 |
|29/6/2020 0:00:00|3274 |
|28/6/2020 0:00:00|3178 |
|18/6/2020 0:00:00|3171 |
|19/6/2020 0:00:00|3059 |
|21/6/2020 0:00:00|3019 |
|30/6/2020 0:00:00|2803 |
+-----------------+-----+
only showing top 10 rows
```

```
In [53]: spark.sql("""select fecha_reporte as fecha, count(*) as count from covid19 where
          fecha_reporte is not NULL group by fecha order by count desc limit 10""").show()

+-----------------+-----+
|            fecha|count|
+-----------------+-----+
|27/6/2020 0:00:00| 4149|
|26/6/2020 0:00:00| 3843|
|24/6/2020 0:00:00| 3541|
|25/6/2020 0:00:00| 3486|
|29/6/2020 0:00:00| 3274|
|28/6/2020 0:00:00| 3178|
|18/6/2020 0:00:00| 3171|
|19/6/2020 0:00:00| 3059|
|21/6/2020 0:00:00| 3019|
|30/6/2020 0:00:00| 2803|
+-----------------+-----+
```

3.4

Dataframes	

```
In [26]: df.select('edad', 'age_group', 'ID').groupBy('edad', 'age_group').count().orderBy('count', ascending=False).show()

+----+-----------+-----+
|edad|  age_group|count|
+----+-----------+-----+
|  30|young adult| 2735|
|  29|young adult| 2611|
|  31|     senior| 2569|
|  28|young adult| 2540|
|  27|young adult| 2494|
|  26|young adult| 2436|
|  33|     senior| 2371|
|  32|     senior| 2362|
|  25|young adult| 2335|
|  34|     senior| 2310|
|  35|     senior| 2292|
|  24|young adult| 2214|
|  36|     senior| 2175|
|  37|     senior| 2132|
|  38|     senior| 2098|
|  40|     senior| 2050|
|  23|young adult| 2041|
|  39|     senior| 1985|
|  22|young adult| 1879|
|  41|     senior| 1783|
+----+-----------+-----+
only showing top 20 rows
```

SparkSQL	In [30]: `spark.sql('select edad, age_group,count(*) as count from covid19 group by edad, age_group order by count desc').show()`

```
+----+-----------+-----+
|edad|  age_group|count|
+----+-----------+-----+
|  30|young adult| 2735|
|  29|young adult| 2611|
|  31|     senior| 2569|
|  28|young adult| 2540|
|  27|young adult| 2494|
|  26|young adult| 2436|
|  33|     senior| 2371|
|  32|     senior| 2362|
|  25|young adult| 2335|
|  34|     senior| 2310|
|  35|     senior| 2292|
|  24|young adult| 2214|
|  36|     senior| 2175|
|  37|     senior| 2132|
|  38|     senior| 2098|
|  40|     senior| 2050|
|  23|young adult| 2041|
|  39|     senior| 1985|
|  22|young adult| 1879|
|  41|     senior| 1783|
+----+-----------+-----+
only showing top 20 rows
```

3.5

Muertes por grupo de edad

Dataframes	In [77]: `from pyspark.sql.functions import col` `df.filter(col('Fecha de muerte').isNotNull()).groupBy('age_group').count().orderBy('count', ascending=False).show()`

```
+-----------+-----+
|  age_group|count|
+-----------+-----+
|     senior| 5435|
|young adult|  151|
|        kid|   47|
+-----------+-----+
```

SparkSQL	In [36]: `spark.sql('''select age_group, count(*) as count from covid19 where` `` `Fecha de muerte` is not null group by age_group order by count desc''').show() ``

```
+-----------+-----+
|  age_group|count|
+-----------+-----+
|     senior| 5435|
|young adult|  151|
|        kid|   47|
+-----------+-----+
```

URI S3: s3://davidnotebook/jupyter/