

In [3]:

```
import pandas as pd

df1 = pd.read_csv("/content/casos.csv", sep=",", engine="python" )

df1.head(6)
```

Out[3]:

	numero_de_caso	fecha_apertura_snvs	fecha_toma_muestra	fecha_clasificacion	provincia	barrio
596	5286275	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	Buenos Aires	NaN
597	5287632	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	Buenos Aires	NaN
598	5287907	15DEC2020:00:00:00.000000	14DEC2020:00:00:00.000000	14DEC2020:00:00:00.000000	Buenos Aires	NaN

In [4]:

```
df1.tail(3)
```

Out[4]:

	numero_de_caso	fecha_apertura_snvs	fecha_toma_muestra	fecha_clasificacion	provincia	barrio
596	5286275	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	Buenos Aires	NaN
597	5287632	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	15DEC2020:00:00:00.000000	Buenos Aires	NaN
598	5287907	15DEC2020:00:00:00.000000	14DEC2020:00:00:00.000000	14DEC2020:00:00:00.000000	Buenos Aires	NaN

In [13]:

```
df2 = df1[["provincia", "genero", "edad"]]
cantidadDeCeldas = df2.size
print(cantidadDeCeldas)

cantidadDeFilas = df2["provincia"].size
print(cantidadDeFilas)
```

1797
599

In [15]:

```
df2.sort_values(['edad'], ascending=False)
```

Out[15]:

	provincia	genero	edad
547	Buenos Aires	femenino	99
389	Buenos Aires	femenino	97
355	CABA	masculino	93
455	Buenos Aires	femenino	91
369	CABA	femenino	90
...

291	Buenos Aires	femenino	1
285	Buenos Aires	masculino	1
180	Buenos Aires	femenino	1
447	Buenos Aires	femenino	0
491	Buenos Aires	femenino	0

599 rows x 3 columns

In [18]:

```
df2.edad
```

Out[18]:

```
0      30
1      17
2      12
3      55
4      84
..
594    23
595    37
596    86
597    36
598    42
Name: edad, Length: 599, dtype: int64
```

In [130]:

```
type(df2.edad)
```

Out[130]:

pandas.core.series.Series

In []:

```
condicion = df2.provincia.str.contains('f')
condicion = df2.provincia.str.endswith('s')
condicion = df2.provincia.str.startswith('s')

df2[condicion]
```

In [59]:

```
condicion = df2.edad == 45
df2[condicion]
```

Out[59]:

	provincia	genero	edad
47	Buenos Aires	femenino	45
75	Buenos Aires	femenino	45
84	Buenos Aires	masculino	45
85	Buenos Aires	femenino	45
151	Buenos Aires	femenino	45
231	Buenos Aires	masculino	45
309	Buenos Aires	masculino	45
341	Buenos Aires	masculino	45
344	Buenos Aires	masculino	45
406	Buenos Aires	masculino	45
421	Buenos Aires	masculino	45

	provincia	genero	edad
493	Buenos Aires	masculino	45
494	Buenos Aires	masculino	45
511	Buenos Aires	femenino	45
577	Buenos Aires	masculino	45

In [71]:

```
condicion = df2.edad <= 18
df2[condicion]
```

Out[71]:

	provincia	genero	edad
1	Buenos Aires	femenino	17
2	Buenos Aires	femenino	12
27	Buenos Aires	femenino	18
65	Buenos Aires	masculino	13
77	Buenos Aires	femenino	16
99	Buenos Aires	masculino	18
103	Buenos Aires	femenino	13
104	Buenos Aires	femenino	15
108	Buenos Aires	masculino	13
135	Buenos Aires	femenino	18
149	Buenos Aires	femenino	18
156	Buenos Aires	masculino	18
180	Buenos Aires	femenino	1
182	Buenos Aires	femenino	18
187	Buenos Aires	masculino	6
201	Buenos Aires	femenino	18
203	Buenos Aires	masculino	11
204	Buenos Aires	masculino	12
206	Buenos Aires	masculino	18
209	Buenos Aires	femenino	12
213	Buenos Aires	masculino	2
229	Buenos Aires	masculino	18
232	Buenos Aires	femenino	17
243	Buenos Aires	masculino	16
253	Buenos Aires	masculino	13
263	Buenos Aires	masculino	2
283	Buenos Aires	femenino	15
284	Buenos Aires	masculino	14
285	Buenos Aires	masculino	1
286	Buenos Aires	femenino	13
291	Buenos Aires	femenino	1
301	Buenos Aires	femenino	15
305	Buenos Aires	femenino	15
306	Buenos Aires	femenino	18
314	Buenos Aires	femenino	11

342	Buenos Aires	masculino	17
383	Buenos Aires	masculino	9
399	Buenos Aires	masculino	18
447	Buenos Aires	femenino	0
454	Buenos Aires	femenino	17
472	Buenos Aires	femenino	6
478	Buenos Aires	masculino	16
491	Buenos Aires	femenino	0
499	Buenos Aires	femenino	5
568	Buenos Aires	masculino	6
569	Buenos Aires	femenino	7
571	Buenos Aires	masculino	2
581	Buenos Aires	femenino	14

In [72]:

```
df2[condicion].index
```

Out[72]:

```
Int64Index([ 1,  2, 27, 65, 77, 99, 103, 104, 108, 135, 149, 156, 180,
            182, 187, 201, 203, 204, 206, 209, 213, 229, 232, 243, 253, 263,
            283, 284, 285, 286, 291, 301, 305, 306, 314, 342, 383, 399, 447,
            454, 472, 478, 491, 499, 568, 569, 571, 581],
           dtype='int64')
```

In [96]:

```
df2["hola"] = pd.Series([])
df2.head(9)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.
    """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    """Entry point for launching an IPython kernel.
```

Out[96]:

	provincia	genero	edad	hola
0	Buenos Aires	masculino	30	NaN
1	Buenos Aires	femenino	17	NaN
2	Buenos Aires	femenino	12	NaN
3	Buenos Aires	femenino	55	NaN
4	Buenos Aires	masculino	84	NaN
5	Buenos Aires	masculino	63	NaN
6	Buenos Aires	masculino	71	NaN
7	Buenos Aires	masculino	29	NaN
8	Buenos Aires	femenino	33	NaN

In [69]:

```
condicion = (df2.edad >= 18) & (df2.provincia == "Buenos Aires")
```

```
condicion = (df2.edad <= 10) | (df2.edad >= 50)
df2[condicion]
```

Out[69]:

```
Int64Index([ 0,  3,  4,  5,  6,  7,  8,  9, 10, 11,
            ...
            588, 590, 591, 592, 593, 594, 595, 596, 597, 598],
           dtype='int64', length=542)
```

In [57]:

```
condicion = (df2.edad <= 10) | (df2.edad >= 50)
df2[condicion]
```

['Buenos Aires' 'Santa Cruz' 'La Rioja' 'Córdoba' 'Misiones' 'Santa Fe' 'Salta' 'San Luis' 'Corrientes' 'Neuquén' nan]

In [83]:

```
condicion = df1.edad.isin([1,3,5])
df2[condicion]
```

Out[83]:

	provincia	genero	edad
180	Buenos Aires	femenino	1
285	Buenos Aires	masculino	1
291	Buenos Aires	femenino	1
499	Buenos Aires	femenino	5

In [84]:

```
condicion = df1.edad.isin([1,3,5])
```

Out[84]:

```
array([1, 5])
```

In [79]:

```
condicion = ~df1.edad.isin([1,3,5])
df2[condicion]
```

Out[79]:

	provincia	genero	edad
0	Buenos Aires	masculino	30
1	Buenos Aires	femenino	17
2	Buenos Aires	femenino	12
3	Buenos Aires	femenino	55
4	Buenos Aires	masculino	84
...
594	Buenos Aires	femenino	23
595	Buenos Aires	masculino	37
596	Buenos Aires	femenino	86
597	Buenos Aires	masculino	36
598	Buenos Aires	femenino	42

595 rows x 3 columns

In [92]:

```
df2.head(4)
```

Out[92]:

	provincia	genero	edad
0	Buenos Aires	masculino	30
1	Buenos Aires	femenino	17
2	Buenos Aires	femenino	12
3	Buenos Aires	femenino	55

In [94]:

```
df2.groupby(["edad"]).agg({'edad': 'count'})
```

Out[94]:

edad	
edad	
0	2
1	3
2	3
5	1
6	3
...	...
90	1
91	1
93	1
97	1
99	1

91 rows x 1 columns

In [63]:

```
df2.agg({'edad': 'max'})
```

Out[63]:

```
edad      99
dtype: int64
```

In [64]:

```
df2.agg({'edad': 'min'})
```

Out[64]:

```
edad      0
dtype: int64
```

In []:

```
pd.merge(df5,df6,on='nombre_variable',how='inner')
pd.merge(df5,df6,on='nombre_variable',how='left')
pd.merge(df5,df6,on='nombre_variable',how='right')
```

In [99]:

```
df2.edad.min()
```

Out[99]:

0

In [100]:

```
df2.edad.max()
```

Out[100]:

99

In []:

```
df6 = df6.drop(["nombre_col"],axis=1)  
df6 = df6.drop(["nro_indice"],axis=0)
```