Creado por:

Isabel Maniega

Importamos las dependecias

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

Creo un Dataframe para el ejemplo

In [2]:
```
df = pd.DataFrame({"X": [10, 20, 30, 40, 50], "Y":[15, 5, 10, 8, 6]})
df.head()
```

Out[2]:

	X	Y
0	10	15
1	20	5
2	30	10
3	40	8
4	50	6

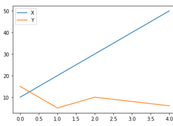
Gráfico rápidos con pandas

.plot()

In [3]:
```
df.plot?
```

In [4]:
```
df.plot()
```

Out[4]:
```
<AxesSubplot:>
```



.plot() y plt.show()

In [5]:
```
df.plot()
plt.show()
```

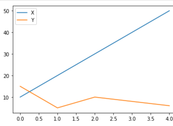


Gráfico de barras

In [7]:
```
df.plot(kind="bar")
plt.show()
```

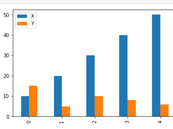
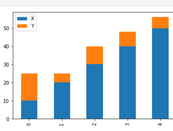


Gráfico apilado

In [8]:
```
df.plot(kind="bar", stacked=True)
plt.show()
```

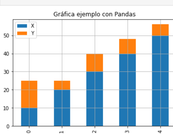


Buscamos más parámetros de df.plot

In [9]:
```
# df.plot?
```

Le pongo título y rejilla para verlo mejor

In [10]:
```
df.plot(kind="bar", stacked=True,
        grid=True, title="Gráfica ejemplo con Pandas")
plt.show()
```



value_counts()

In [12]:
```
df
```

Out[12]:

	X	Y
0	10	15
1	20	5
2	30	10
3	40	8
4	50	6

In [14]:
```
df.Y.value_counts()
```

Out[14]:
```
8     1
10    1
5     1
6     1
15    1
Name: Y, dtype: int64
```

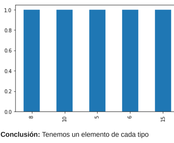
In [15]:
```
type(df.Y.value_counts())
```

Out[15]:
```
pandas.core.series.Series
```

Simplemente añadimos .plot y obtenemos la gráfica

In [17]:
```
df.Y.value_counts().plot(kind="bar")
```

Out[17]:
```
<AxesSubplot:>
```



Conclusión: Tenemos un elemento de cada tipo

Si fueran más se vería de otra forma obviamente

Imaginamos otro df

In [19]:
```
df_2 = pd.DataFrame({"X": [10, 20, 30, 40, 50, 60, 70, 80],
                     "y": [10, 20, 10, 40, 20, 30, 20, 10]})
df_2
```

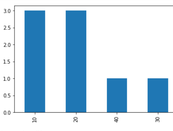
Out[19]:

	X	y
0	10	10
1	20	20
2	30	10
3	40	40
4	50	20
5	60	30
6	70	20
7	80	10

In [21]:
```
df_2.y.value_counts()
```

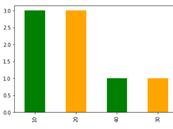
Out[21]:
```
10    3
20    3
40    1
30    1
Name: y, dtype: int64
```

In [22]:
```
df_2.y.value_counts().plot(kind="bar")
plt.show()
```



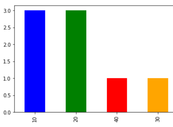
Con color diferente...

In [23]:
```
df_2.y.value_counts().plot(kind="bar",
                           color=["green", "orange"])
plt.show()
```



o incluso...

In [25]:
```
df_2.y.value_counts().plot(kind="bar",
                           color=["blue", "green", "red", "orange"])
plt.show()
```



Creado por:

Isabel Maniega