

# Python para Análisis de Datos

Módulo 05

# Pandas

# Gráficos

Pandas ofrece muchas facilidades para hacer gráficos de manera sencilla. Se basa en Matplotlib, el principal módulo de gráficos de Python, que estudiaremos la próxima clase, para realizar los gráficos.

Podemos acceder a las funcionalidades para graficar a través del método `plot`. Permite acceder a distintos tipos de gráficos de forma sencilla. Por ejemplo, se pueden hacer histogramas con `plot.hist()`, gráficos de barras con `plt.bar`, y de torta con `plt.pie()`.

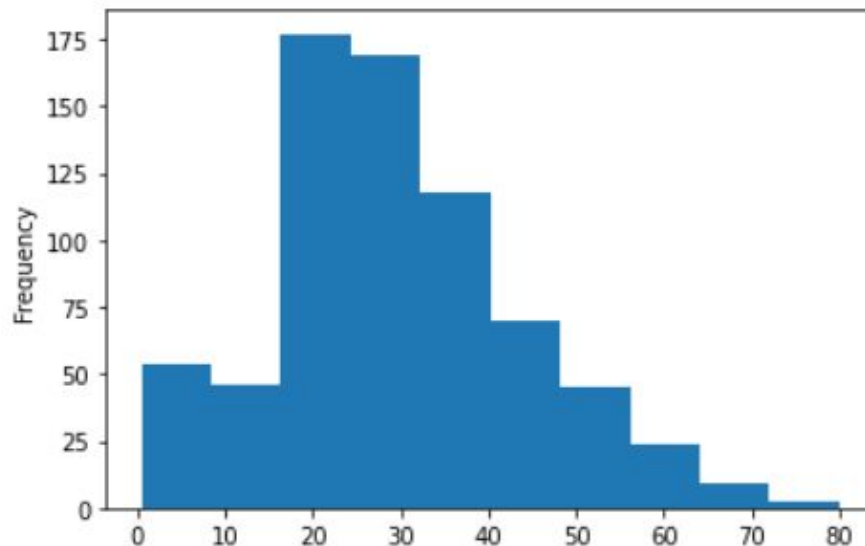


# Histogramas

Un **histograma** es una representación de la distribución de un conjunto de números. El rango se divide en barras y la altura de la barra es proporcional a la cantidad de datos que entran en el rango de la barra (ancho). Se puede definir la cantidad de barras a mostrar.

```
data["Age"].plot.hist()
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f8026329a30>

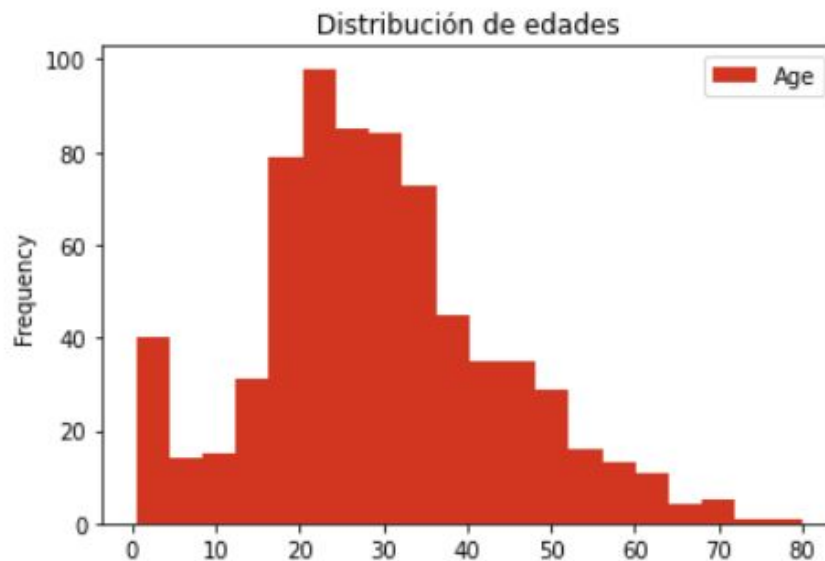


# Histogramas

Los métodos vienen con números parámetros para controlar el aspecto del gráfico.

```
data["Age"].plot.hist(bins=20, title="Distribución de edades",  
                      legend=True, color="#D23620")
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f8024b73070>

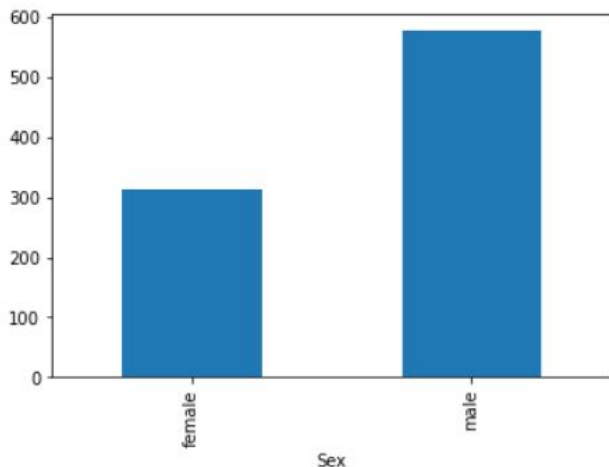


# Gráfico de barras

Este tipo de gráficos muestra barras de altura variable. En general, son útiles para mostrar estadísticas tomadas por categorías por medio de groupby.

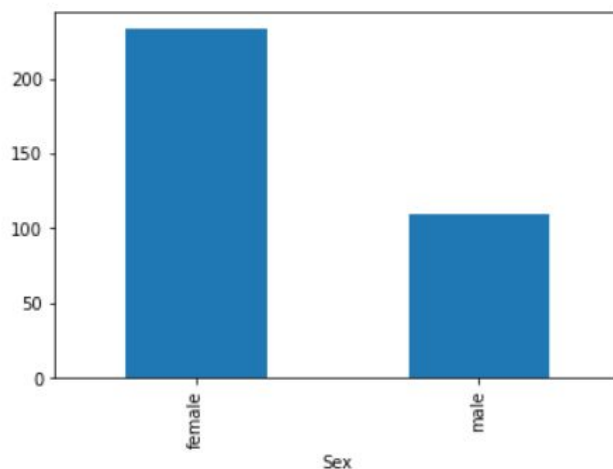
```
data.groupby("Sex").size().plot.bar()
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f801d2aleb0>



```
data.groupby("Sex")["Survived"].sum().plot.bar()
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f8021fcfa90>



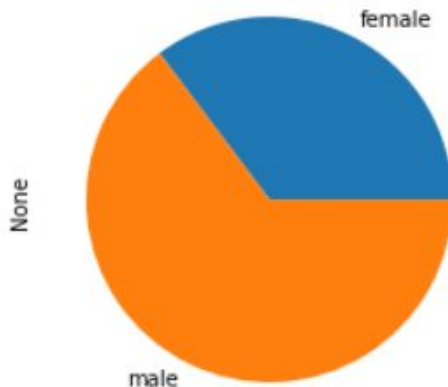
# Gráfico de torta

Otra forma de representar este tipo de información es el **gráfico de tortas**, donde el área de la cuña es proporcional al número que representa.

```
data.groupby("Sex").size().plot.pie(title="Proporcion de hombres y mujeres")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f801b7fbf70>
```

proporcion de hombres y mujeres

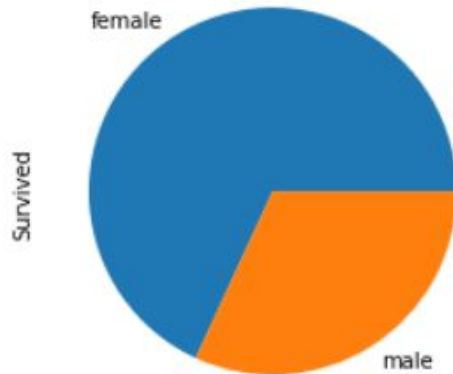


## Gráfico de tortas

```
data.groupby("Sex")["Survived"].sum().plot.pie(title="Proporcion de hombres y mujeres\nentre los supervivientes")
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f801b7af8b0>

Proporcion de hombres y mujeres  
entre los supervivientes





# Gráfico de líneas

Si los datos están ordenados se puede usar `plot` para representarlos sobre una línea en donde el eje **x** es el índice de la serie y el eje **y** es el valor.

Si este método se aplica sobre un dataframe, las columnas numéricas se muestran en el mismo gráfico compartiendo el eje **x**. A continuación creamos un dataset para practicar.

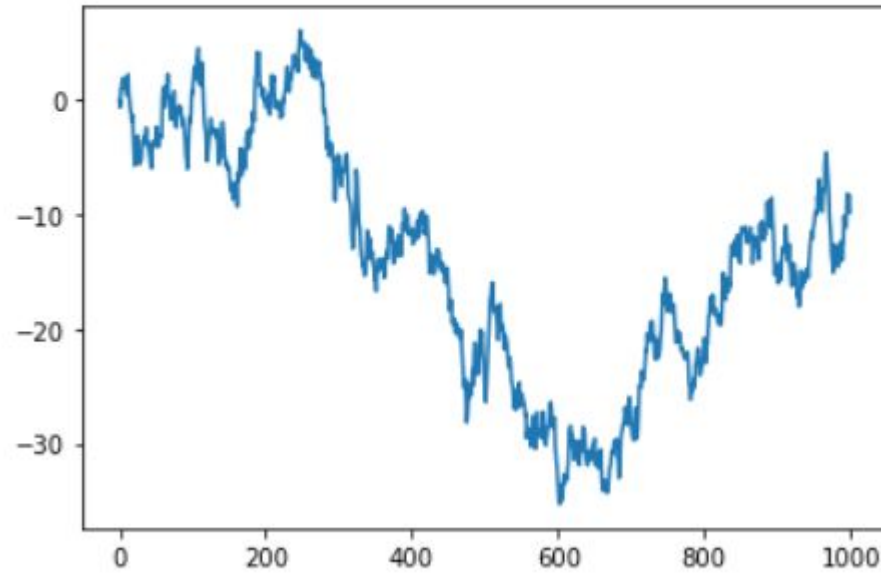
```
arr = np.random.randn(1000,4).cumsum(axis=0)
```

```
df = pd.DataFrame(arr, columns=["A", "B", "C", "D"])
```

## Gráfico de líneas

```
df["A"].plot()
```

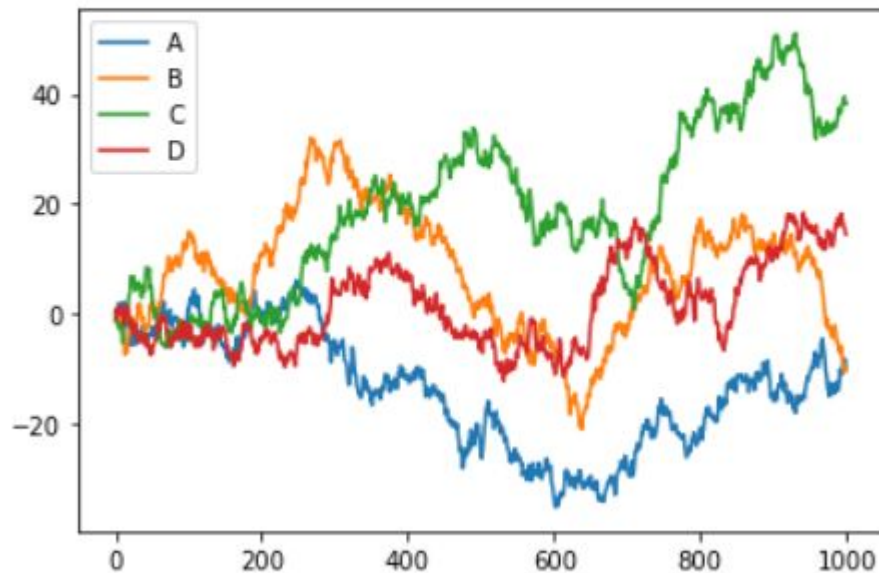
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f801b1bb460>
```



## Gráfico de líneas

```
df.plot()
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f801b109ca0>



# ¡Muchas gracias!

¡Sigamos trabajando!