

A collection of approximately 18 squares in various shades of blue and grey, scattered across the top half of the slide.

Visualización

Data Mining

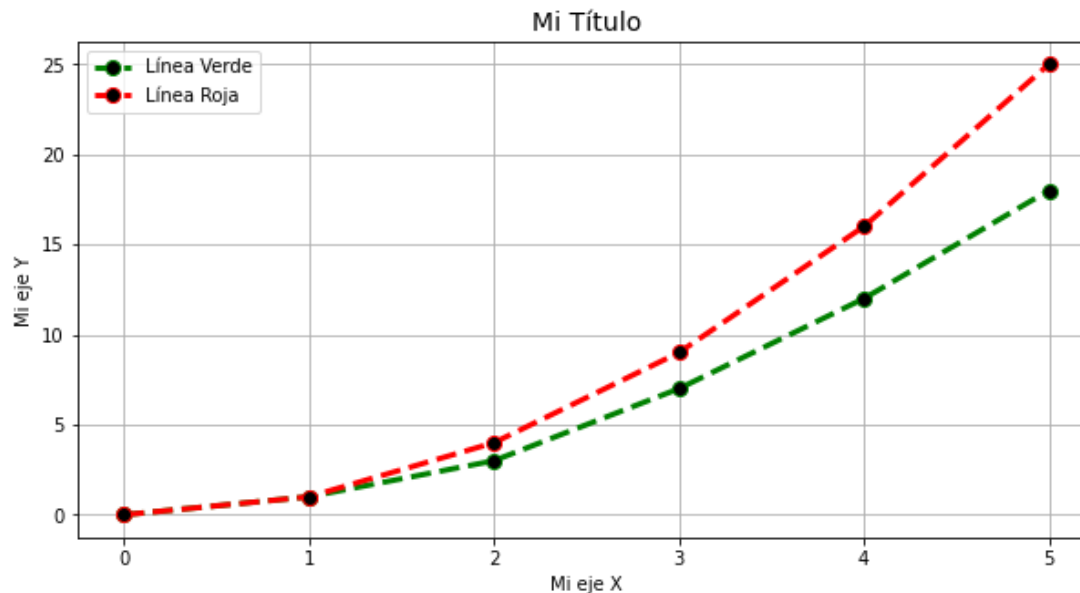
Ester Vidaña Vila

Matplotlib

- Es una librería para crear visualizaciones estáticas, animadas o interactivas en Python.
- Recordad que para utilizar las librerías primero debemos importarlas:

```
import matplotlib.pyplot as plt
```

- Ejemplo de visualización de un gráfico:



Matplotlib – gráfico de líneas

```
plt.figure(figsize=(10,5))

plt.plot(x, y, color="green", label="Línea Verde", lw=3, ls='--', marker="o",
markersize=8, markerfacecolor="black")

plt.plot(x, z, color="red", label="Línea Roja", lw=3, ls='--', marker="o",
markersize=8, markerfacecolor="black")

plt.title("Mi Título", fontsize=14)

plt.xlabel('Mi eje X')
plt.ylabel('Mi eje Y')

plt.grid(True)

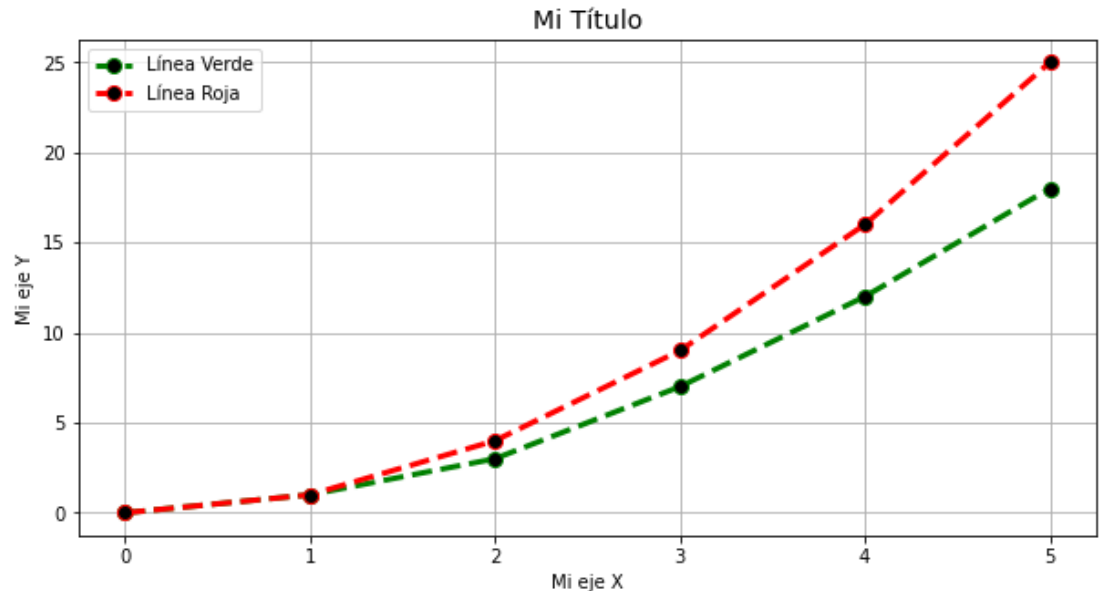
plt.legend()
```

Tamaño del plot

Datos a mostrar en el plot

Cuadrícula

Leyenda



Matplotlib – funciones interesantes para gráficos de líneas

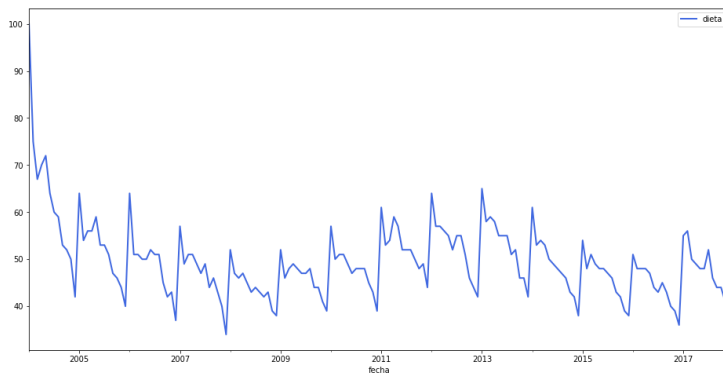
■ Rolling:

- *Pandas dispone de una función para hacer un promedio de muestras en una determinada ventana para poder visualizar mejor las tendencias.*

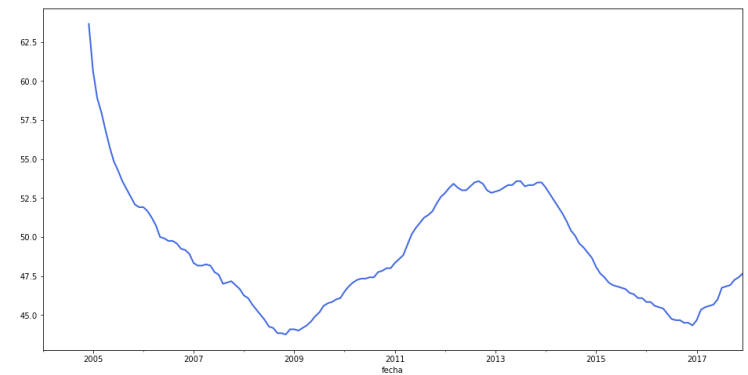
<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rolling.html>

```
df["dieta"].rolling(12).mean().plot(figsize=(16,8), color="royalblue", lw=2);
```

Datos sin aplicar rolling



Mismos datos tras aplicar rolling



Matplotlib – funciones interesantes para gráficos de líneas

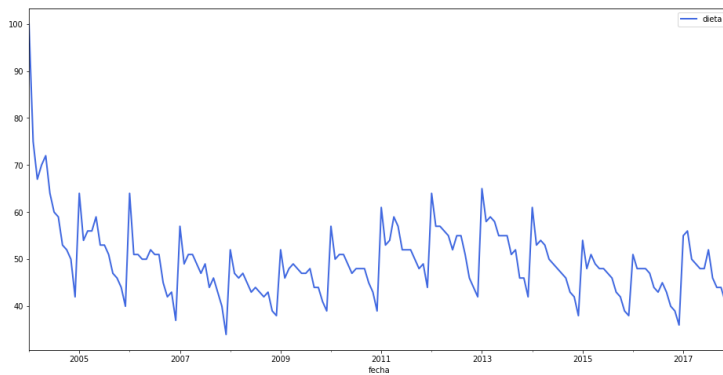
■ Diff:

- *Pandas dispone de una función para calcular la diferencia entre puntos consecutivos, lo que nos permite ver mejor la estacionalidad.*

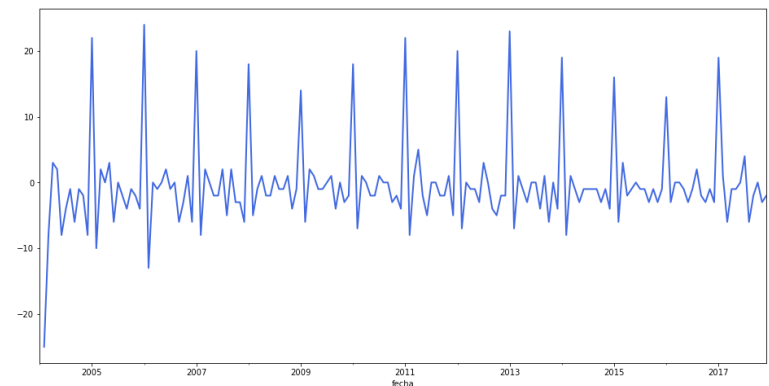
<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.diff.html>

```
df["dieta"].diff().plot(figsize=(16,8), color="royalblue", lw=2);
```

Datos sin aplicar diff



Mismos datos tras aplicar diff



Resolución de los gráficos

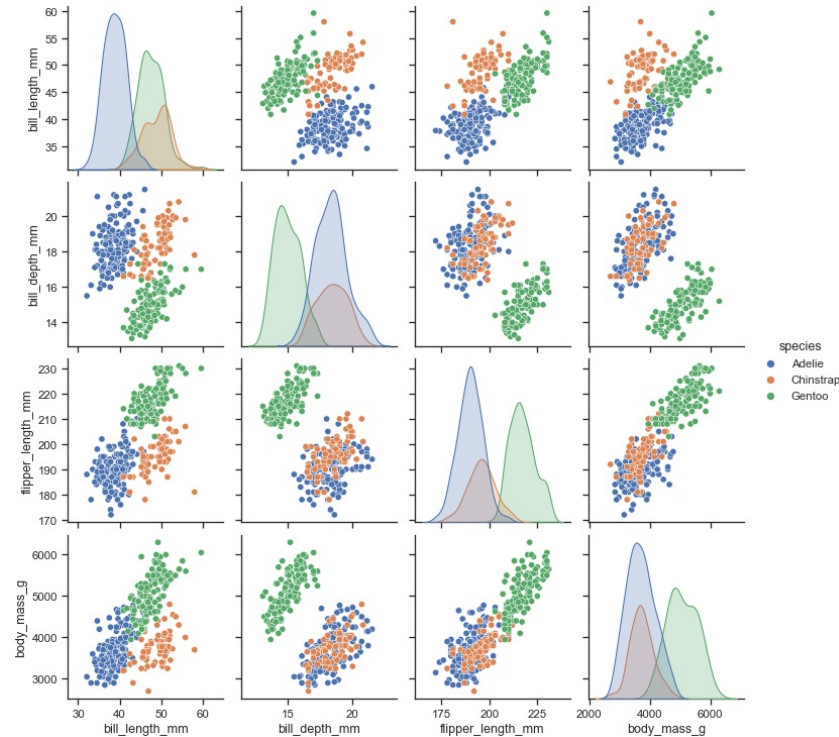
- Podemos modificar la resolución de los gráficos utilizando:

```
from IPython.display import set_matplotlib_formats  
set_matplotlib_formats("jpeg")
```

Seaborn

- Seaborn es una librería de Python basada en matplotlib que nos da una interfície para dibujar gráficos estadísticos visualmente atractivos.

```
import seaborn as sns
```



Datasets predeterminados

- Seaborn dispone de varios datasets en su github:

<https://github.com/mwaskom/seaborn-data>

- Los podemos importar directamente en Python con la función `load_dataset`:

Ejemplo:

```
import seaborn as sns  
titanic = sns.load_dataset("titanic")
```


Paletas de colores

- Si no nos gustan los colores por defecto de seaborn, podemos cambiar la paleta de colores:

https://seaborn.pydata.org/generated/seaborn.color_palette.html

paleta = sns.color_palette() #nos devuelve la paleta que estamos utilizando

sns.palplot(paleta) #para printar la paleta y ver qué colores contiene

Podemos utilizar paletas pre-determinadas de seaborn...

paleta_predeterminada = sns.color_palette("rainbow", 10) #paleta de 10 colores

O crear nuestras propias paletas:

mi_paleta = ["#9b59b6", "#3498db", "#95a5a6", "#e74c3c", "#34495e", "#2ecc71"]

paleta_propia = sns.color_palette(mi_paleta) #paleta de 6 colores

Temas

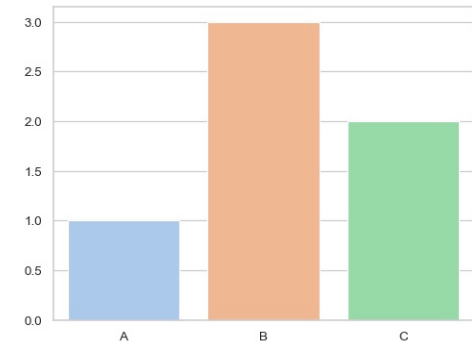
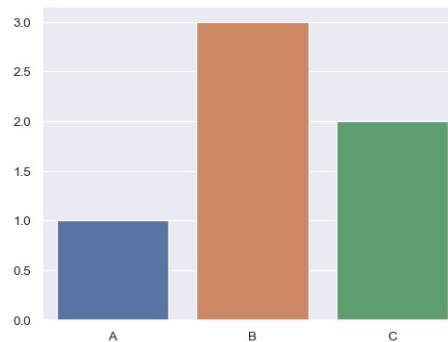
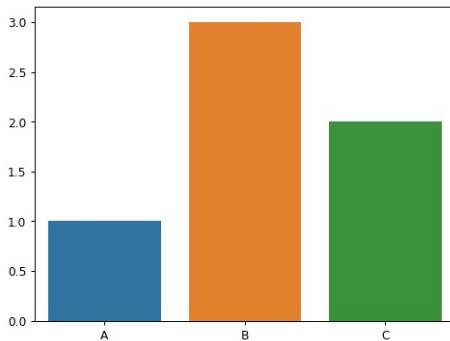
- También se pueden cambiar los temas en general de los gráficos:

https://seaborn.pydata.org/generated/seaborn.set_theme.html

```
sns.set_theme(style="white")
```

```
sns.set_theme(style="darkgrid")
```

```
sns.set_theme(style="whitegrid",palette="pastel")
```



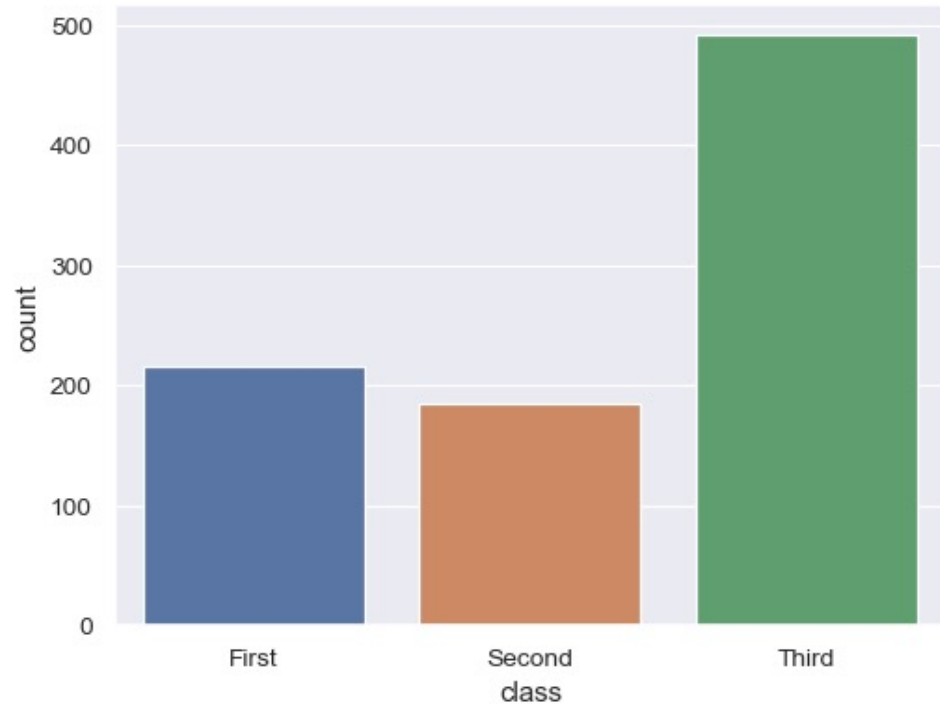
Tipos de gráficos: gráfico de contaje

- Muestra la cantidad de observaciones de cada categoría utilizando barras.

<https://seaborn.pydata.org/generated/seaborn.countplot.html>

Ejemplo:

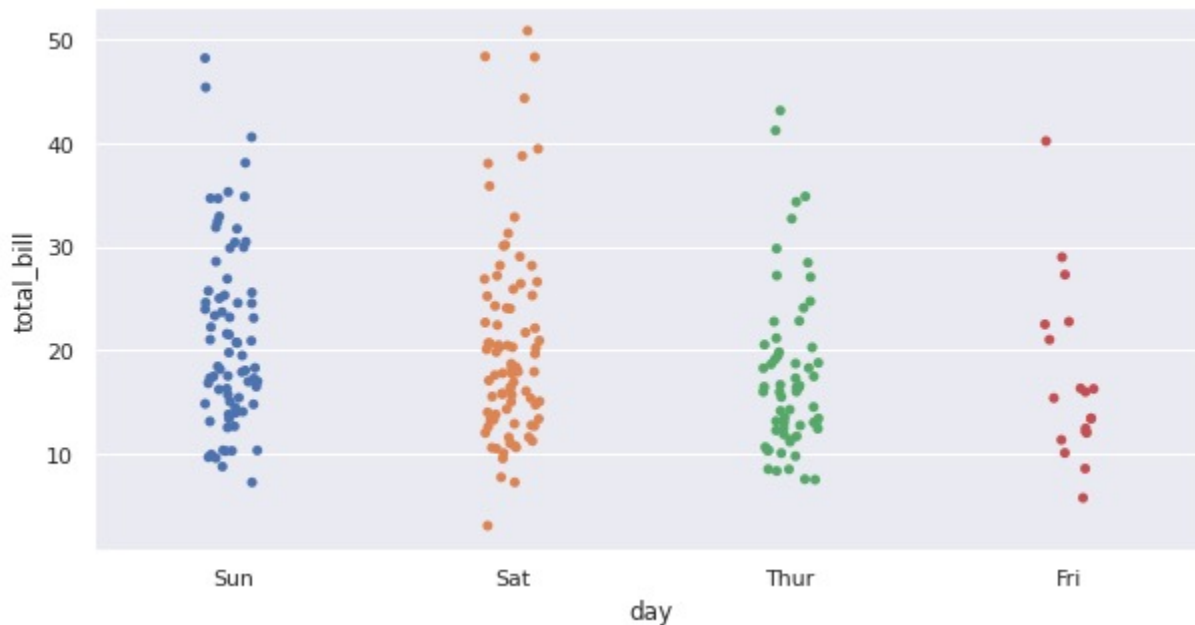
Cantidad de pasajeros de
Primera, segunda y tercera clase
del dataset Titanic.



Tipos de gráficos: strip plot

- Muestra scatterplots que relacionan una variable numérica y una variable categórica.

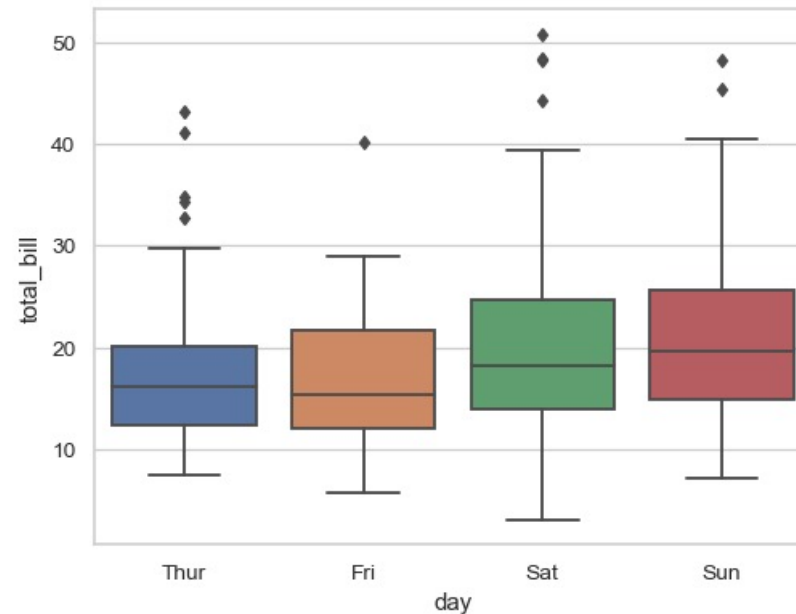
<https://seaborn.pydata.org/generated/seaborn.stripplot.html>



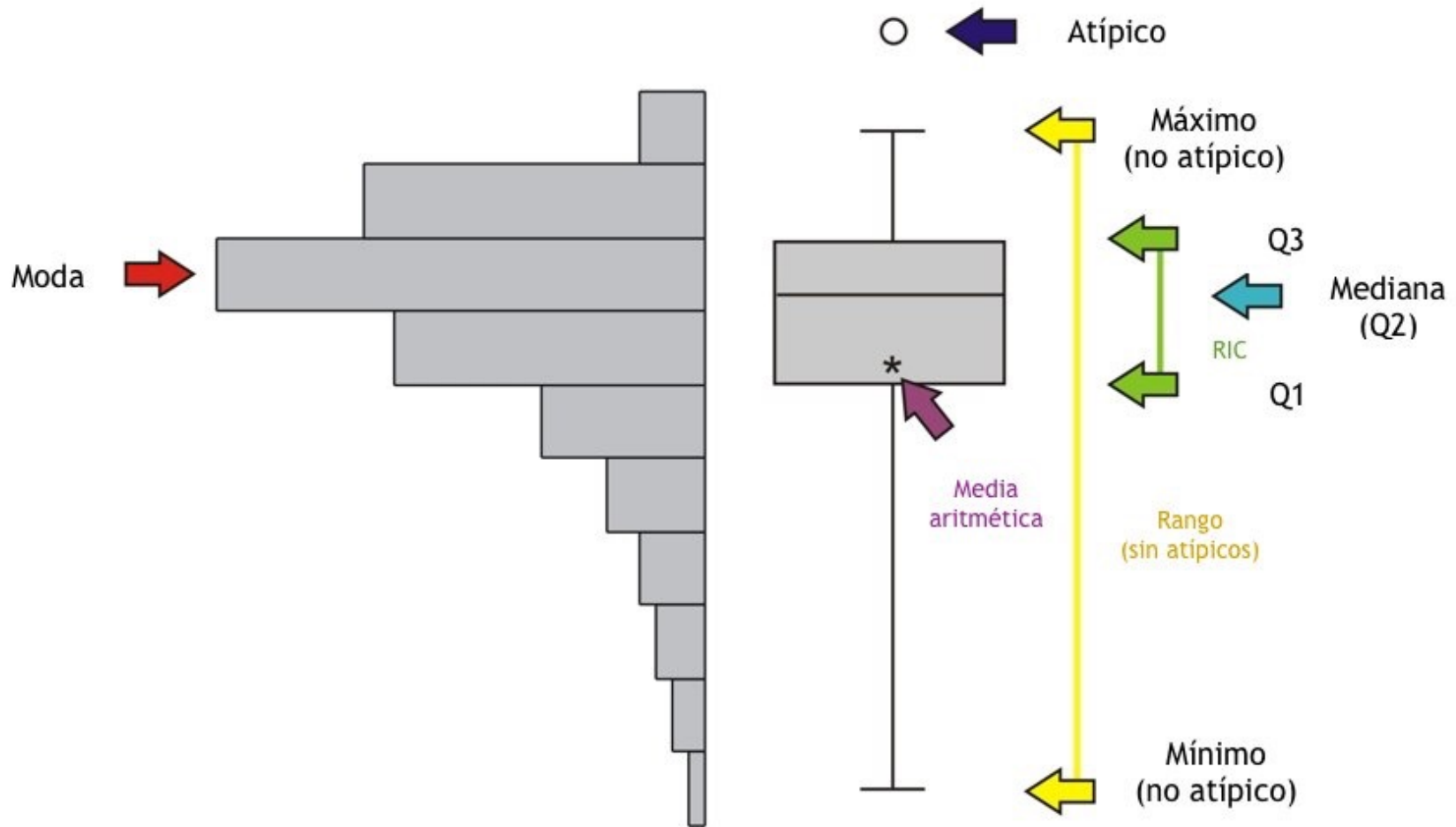
Tipos de gráficos: boxplot

- Nos permite ver la distribución de probabilidad de una variable a partir de un conjunto de datos.

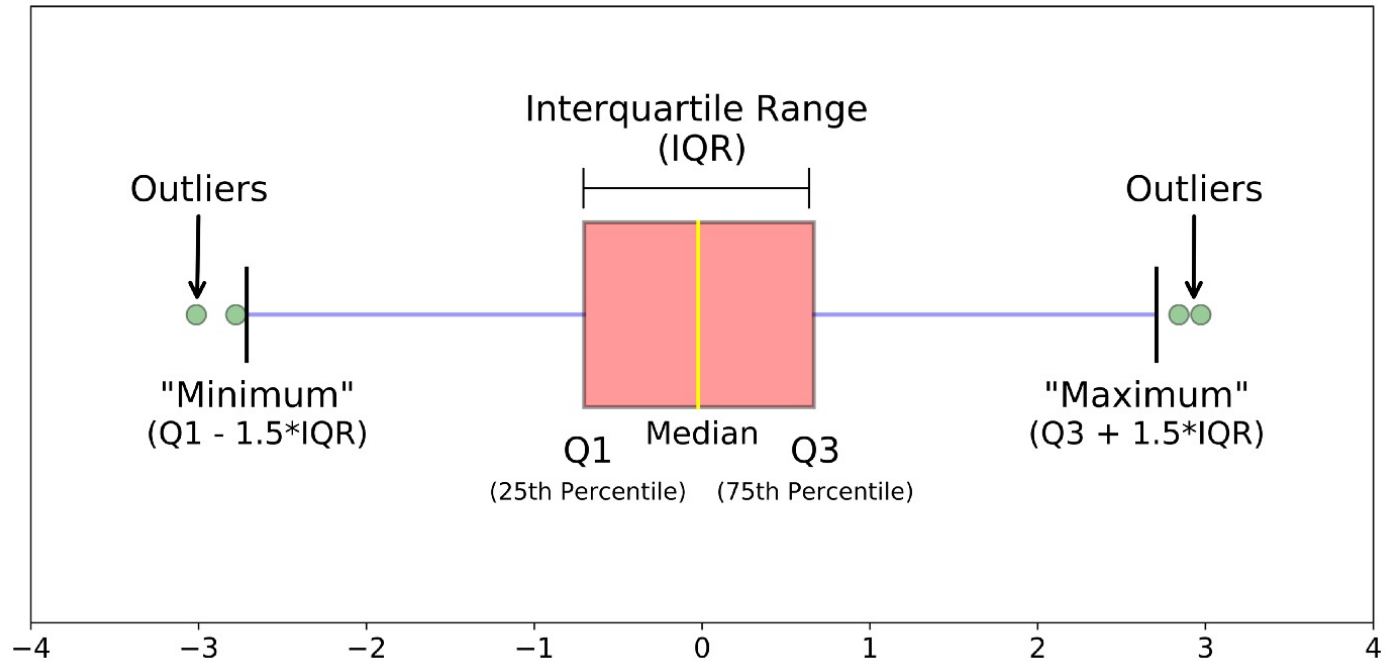
<https://seaborn.pydata.org/generated/seaborn.boxplot.html>



Tipos de gráficos: boxplot



Tipos de gráficos: boxplot → outliers detection

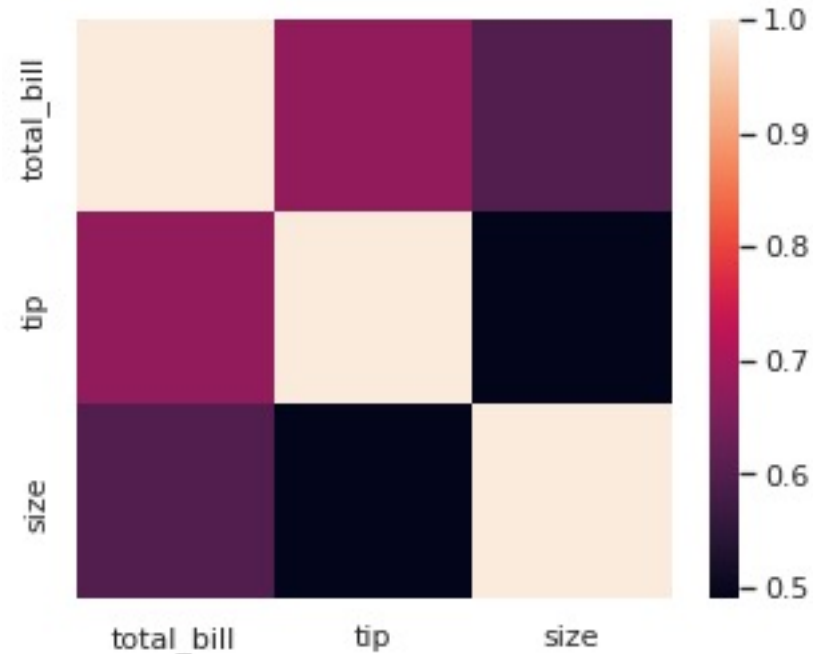


Fuente: <https://towardsdatascience.com/understanding-boxplots-5e2df7bc51>

Tipos de gráficos: heatmap

- Muestra una matriz codificada y con colores

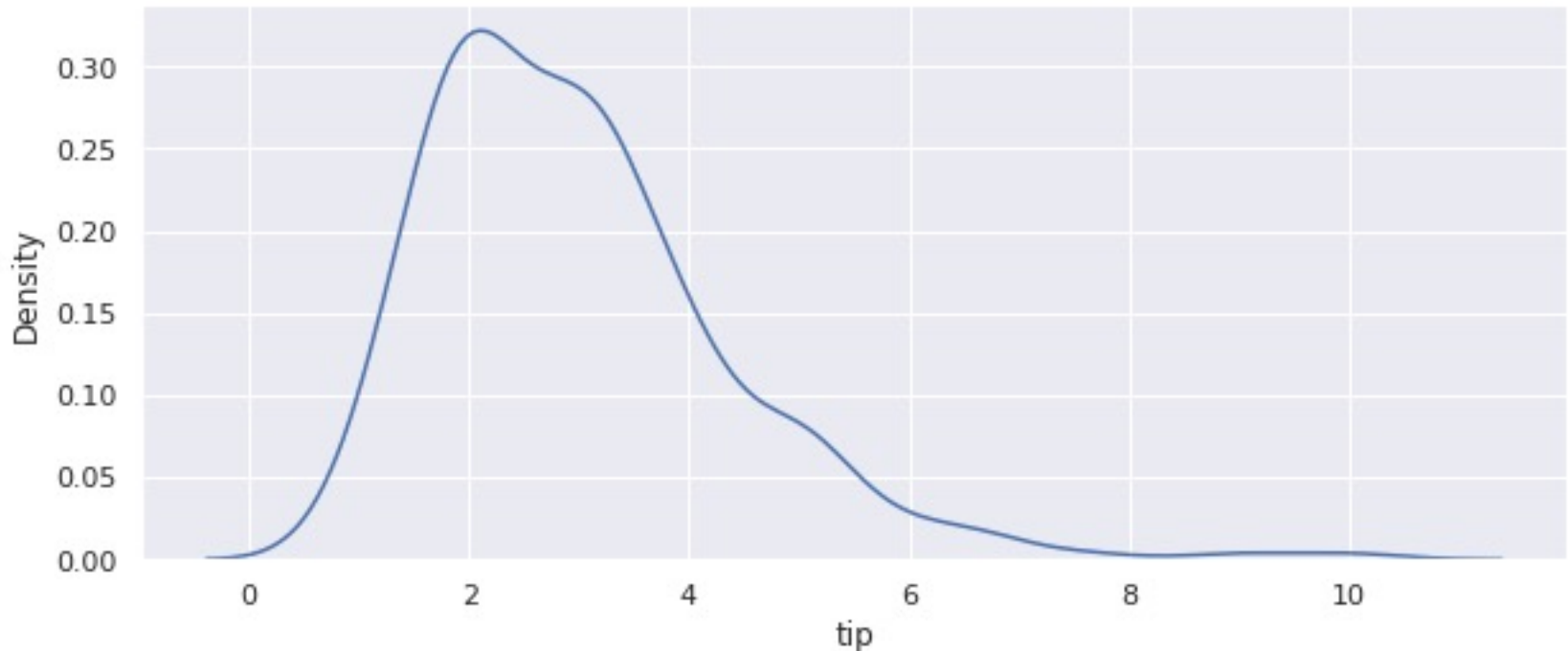
<https://seaborn.pydata.org/generated/seaborn.heatmap.html>



Tipos de gráficos: KDE

- Muestra un gráfico de estimación de densidad de Kernel, que representa la distribución de las estimaciones del dataset de forma continua. Así pues, es como un gráfico análogo al histograma.

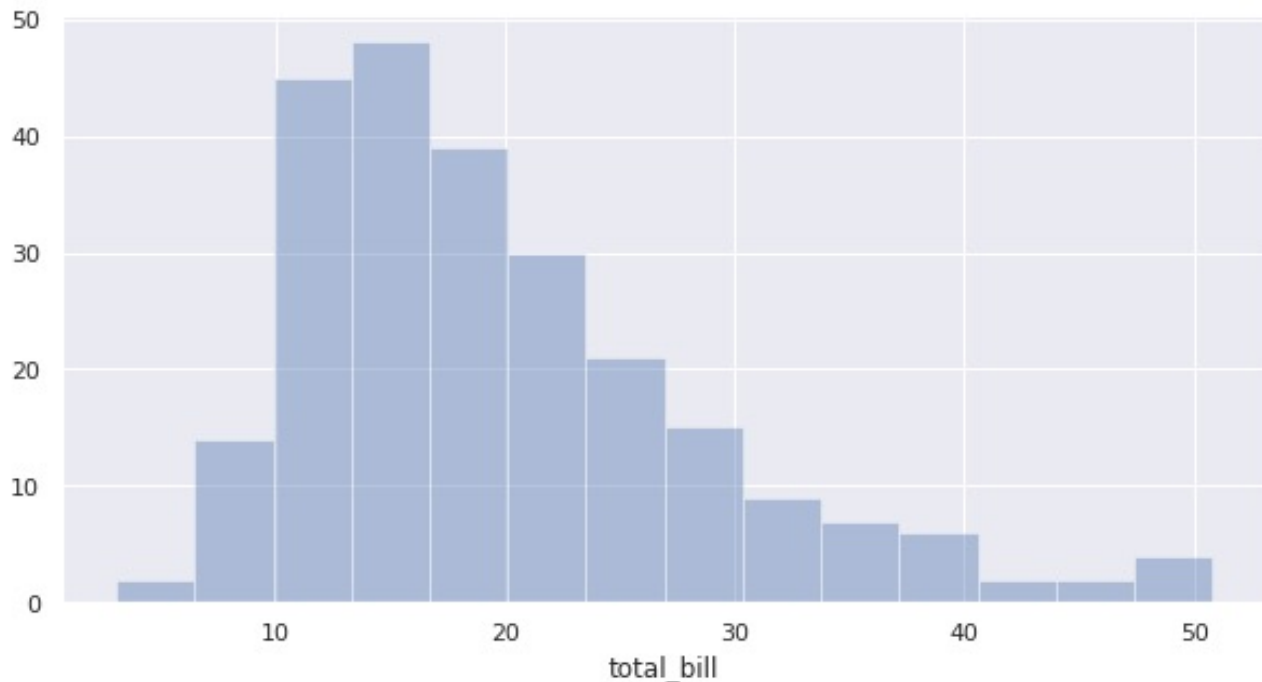
<https://seaborn.pydata.org/generated/seaborn.kdeplot.html>



Tipos de gráficos: histograma

- Muestra un gráfico de barras en las que se muestra la suma de observaciones para cada valor o grupo de valores distintos.

<https://seaborn.pydata.org/generated/seaborn.histplot.html>



Tipos de gráficos: scatter plot

- Muestra puntos con posibilidad de agruparlos de distintas formas.

<https://seaborn.pydata.org/generated/seaborn.scatterplot.html>

