**DATA VISUALIZATION**

| [doc matplotlib.pyplot](https://matplotlib.org/stable/api/pyplot_summary.html)  [matplotlib API ref](https://matplotlib.org/stable/api/index)  [doc pandas visualization](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html)  [doc seaborn visualization](http://seaborn.pydata.org/examples/index.html)  [seaborn API ref](https://seaborn.pydata.org/api.html)  [doc plotline](https://plotly.com/python/) |  | - EJE X: representa la categoría a medir (cualquier valor)  - EJE Y: representa la medición  (valores continuos) |
| --- | --- | --- |

**PANDAS:**

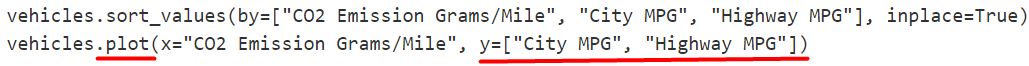
**→ import matplotlib.pyplot as plt**

* df**.plot.***TYPE***(x='**...**', y=**...**', \*\*kwargs);**
* df**.plot(***kind***= “**line**”, x='**...**', y=**...**', \*\*kwargs);**

****

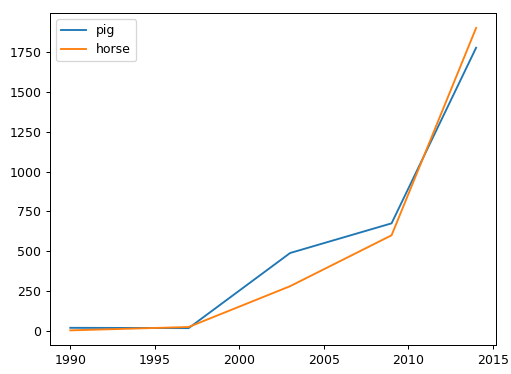
**Múltiples valores en un eje:**

→ Poniéndolo en un **LIST**, o creando el gráfico del dataFrame entero

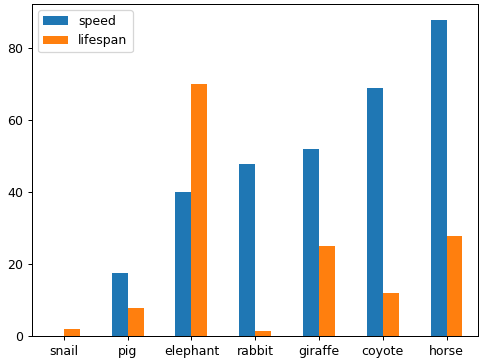
****

**TYPES:**

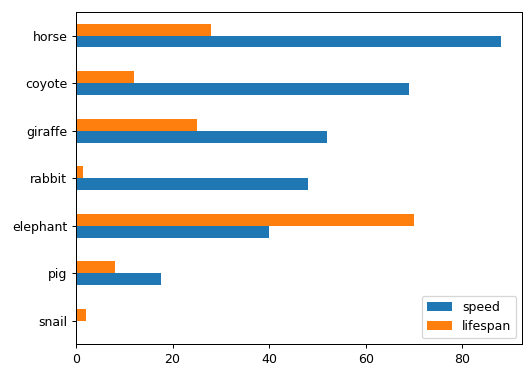
* **‘line’ (LINEAS):** line plot (default)



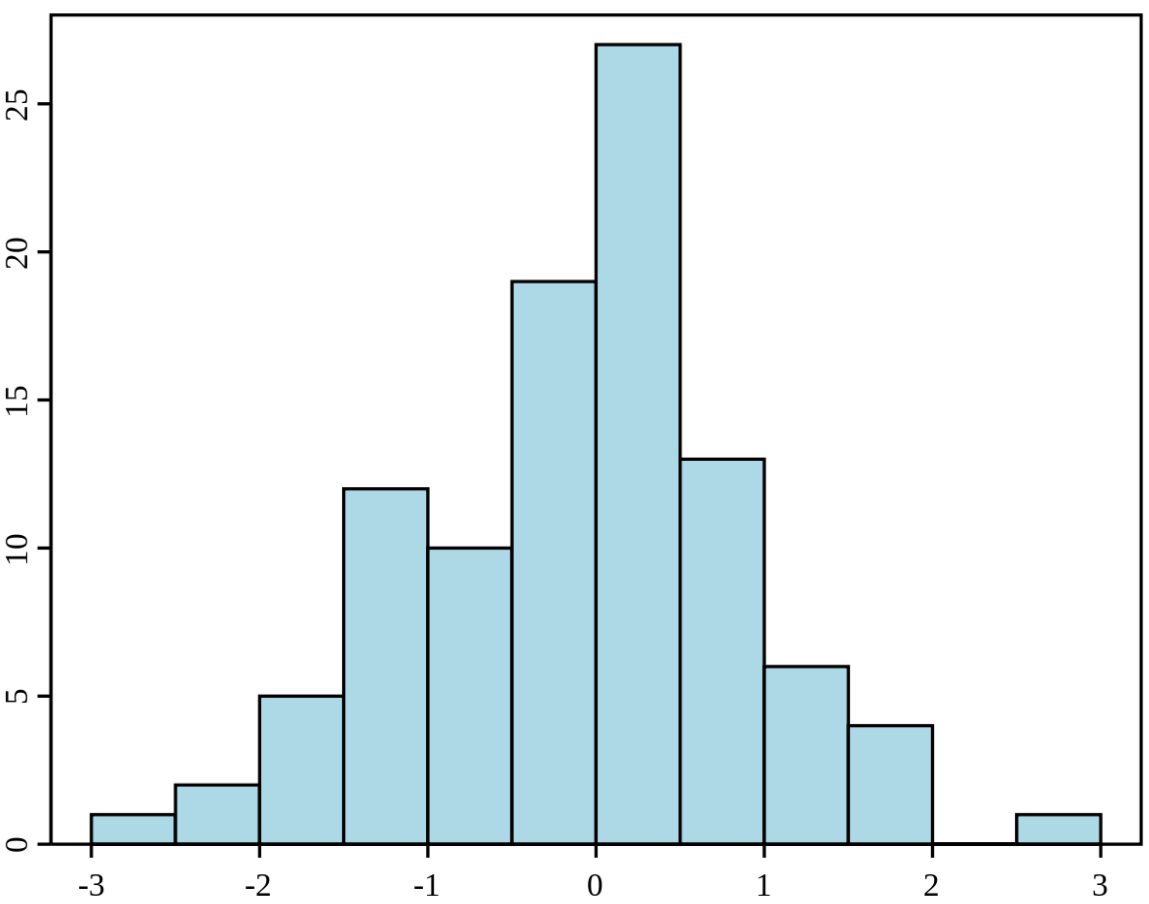
* **‘bar’ (BARRAS VERTICALES):** vertical bar plot → ÚTIL para variables CATEGÓRICAS



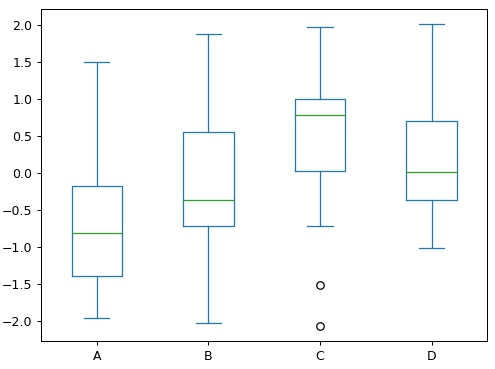
* **‘barh’ (BARRAS HORIZONTALES):** horizontal bar plot



* **‘hist’ (HISTOGRAMA):** histogram (si se lo aplicas a un único valor te muestra su frecuencia) → ÚTIL para variables CONTINUAS

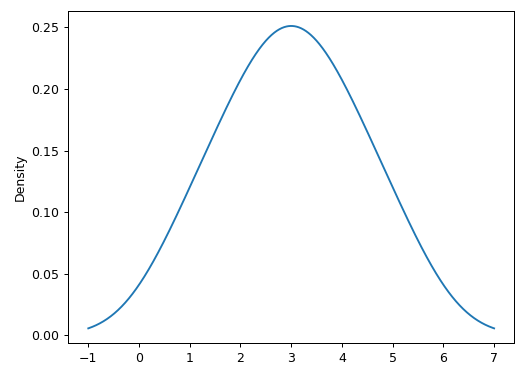


* **‘box’ (CAJAS):** boxplot → muestra la distribución de datos cuantitativos a través de percentiles en el que los valores atípicos están en los extremos (la mediana en el medio)

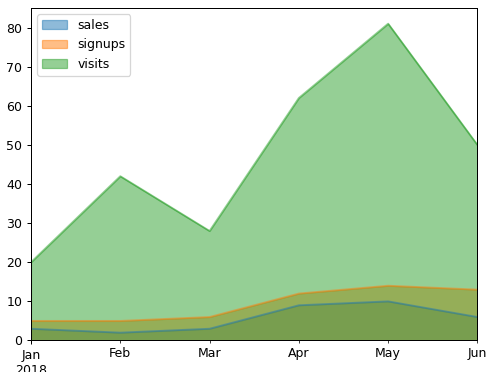


* **‘kde’ (CAMPANA DE GAUSS “probabilidad”):** Kernel Density Estimation plot

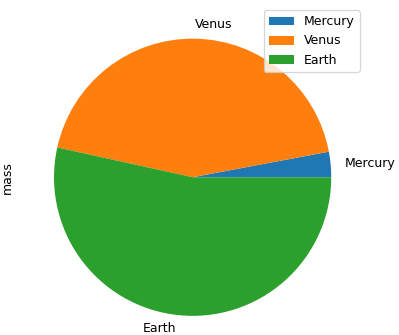
*ò* **‘density’ (CAMPANA DE GAUSS “probabilidad”):** same as ‘kde’



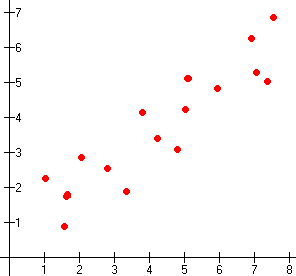
* **‘area’ (AREAS):** area plot



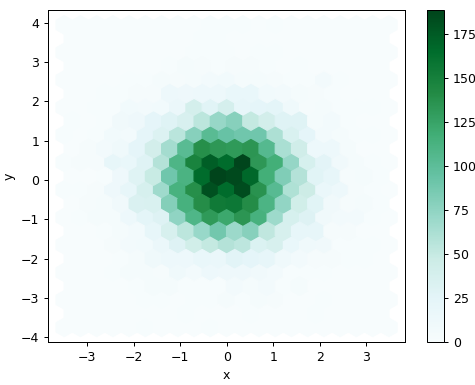
* **‘pie’ (TARTA) :** pie plot



* **‘scatter’ (DISPERSIÓN):** scatter plot (DataFrame only)



* **‘hexbin’ (HEXÁGONOS):** hexbin plot (DataFrame only)

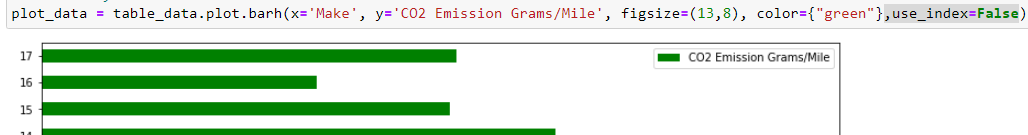


**KWARGS:**

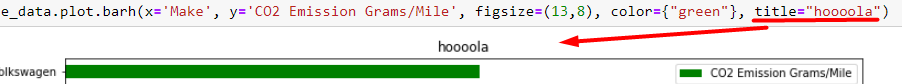
* **ax:** matplotlib axes object → default None. (Un eje de la figura actual)
* **subplots:** bool → default False. (subparcelas separadas para cada columna)
* **sharex:** bool → default True if ax is None else False (En el caso subplots=True, comparta el eje “x” y configure algunas etiquetas del eje x como invisibles)
* **sharey:** bool → default False (En el caso subplots=True, comparta el eje “y” y configure algunas etiquetas del eje y como invisibles.)
* **layout:** tuple → optional ((filas, columnas) para el diseño de subparcelas)
* **figsize:** a tuple (width, height). (INDICA EL TAMAÑO DEL GRÁFICO)



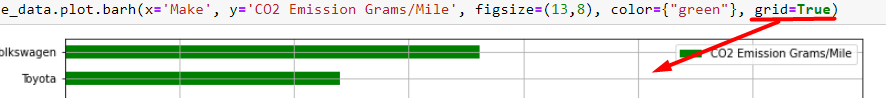
* **use\_index:** bool → default True (deja de utilizar el index (en este caso el nombre de los coches) y pone el índice numérico inicial (0,1,2,3.,...)



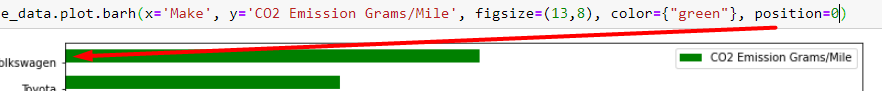
* **title:** str or list



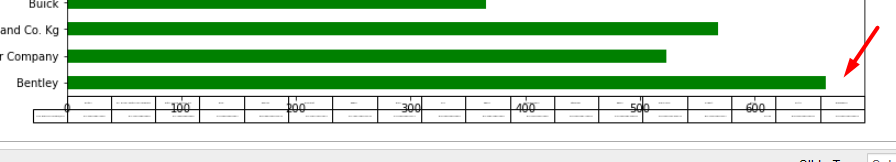
* **grid:** bool → default None (matlab style default) (LÍNEAS DE CUADRÍCULA de los ejes)



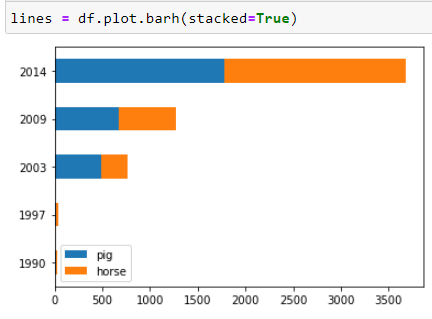
* **legend:** bool or {‘reverse’}
* **style:** list or dict. Si se ponen estilos no se puede poner el arg de “colors”
* **logx:** bool or ‘sym’ → default False. (Pone el eje x en escala logarítmica)
* **logy:** bool or ‘sym’ → default False. (Pone el eje y en escala logarítmica)
* **loglog:** bool or ‘sym’ → default False (Pone ambos ejes en escala logarítmica)
* **xticks:** sequence
* **yticks:** sequence
* **xlim:** 2-tuple/list (Estable los límites X de los ejes actuales.)
* **ylim:** 2-tuple/list (Estable los límites Y de los ejes actuales.)
* **xlabel:** label → optional (Etiqueta/título para el eje X)
* **ylabel:** label → optional (Etiqueta/título para el eje Y)
* **rot:** int → default None (Rotación de ticks (xticks para gráficos verticales, yticks para gráficos horizontales)
* **fontsize:** int → default None (Tamaño de fuente para xticks e yticks)
* **colormap:** str or matplotlib colormap object → default None (Mapa de colores para seleccionar colores. Si es una cadena, cargue el mapa de colores con ese nombre de matplotlib)
* **colorbar:** bool → optional (Si es Verdadero, trace la barra de colores (solo relevante para los gráficos de 'dispersión' y 'hexbin')
* **position:** float → (Posición de las barras respecto a su leyenda → va de 0 a 1 y el defecto es 0.5)



* **table:** bool, Series or DataFrame → default False (Se dibuja una tabla con los datos debajo del gráfico)



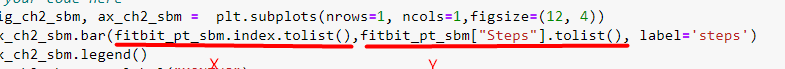
* **yerr:** DataFrame, Series, array-like, dict and str
* **xerr:** DataFrame, Series, array-like, dict and str
* **stacked:** bool → default False in line and bar plots, and True in area plot (cuando hay múltiples áreas o barras, podemos agruparlas y que se superpongan)



* **sort\_columns:** bool → default False (Ordenar los nombres de las columnas para determinar el orden de las parcelas)
* **secondary\_y:** bool or sequence → default False (EJES SECUNDARIOS → hay que dar una list o tuple indicando que ejes serán los secundarios)
* **mark\_right:** bool → default True (Cuando utiliza un eje Y secundario, marca automáticamente las etiquetas de las columnas con "(derecha)" en la leyenda)
* **include\_bool:** bool → default is False
* **backend:** str → default None

**MATPLOTLIB**

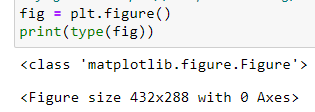
**→ import matplotlib.pyplot as plt**



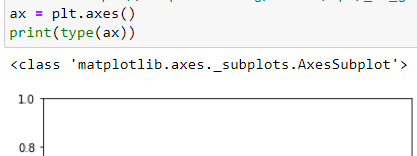
*→ Se puede hacer lo que queramos, pero es más “arduo”. Primero creamos el “CANVAS” y luego le AÑADIMOS “ELEMENTOS”*

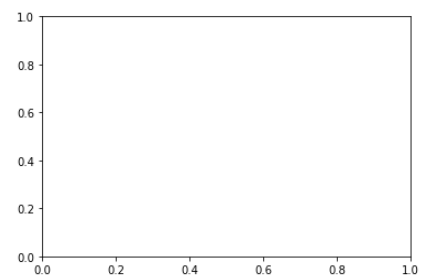
1. para generar un **gráfico vacío**  donde meteremos todos sus elementos

1º fig = **plt.figures()** → creación de ejes, contenido a nivel de figura

****

2º ax = **plt.axes()** → la mayoría de los métodos de trazado, etiquetas de ejes, estilo de ejes, …

****

****

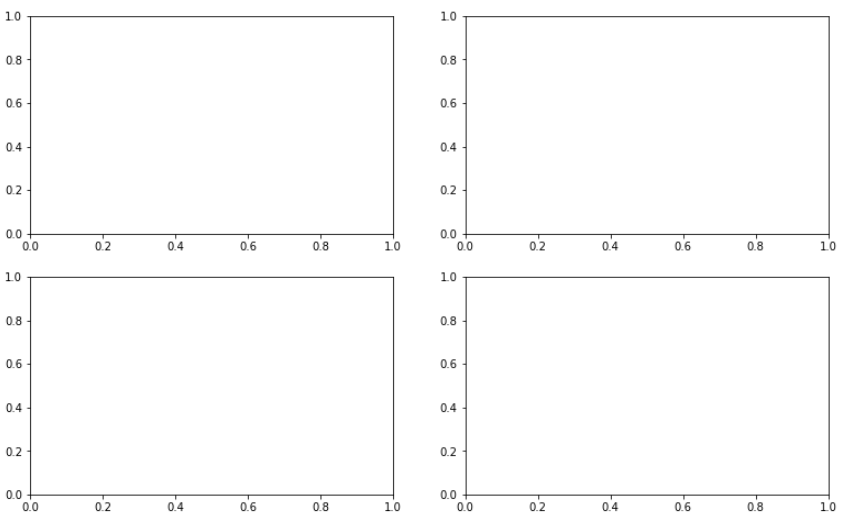
\* ax.get\_children() → da información sobre el gráfico

1. para **genera un canvas** donde meteremos todos los elementos de los gráficos
2. *CANVAS EN BLANCO:*

→ fig, ax = **plt.subplots(**ncols=1, nrows=1, figsize=(13,8)**)**:

(este método de matplotlib genera dos objetos (figure y axes). Si solo le asignaremos una variable, esta sería una tupla con dos objetos, por eso mejor crear dos variables directamente)



****

**→ ax[0,0], ax[0,1], ax[1,0]** y **ax[1,1]**

**→** El gráfico en este ejemplo se almacenaría en la variable **“fig”**

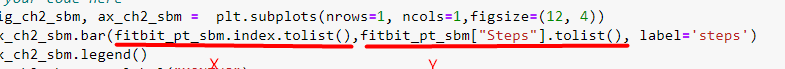
También podríamos crearlo de la siguiente manera:

****

**→ ax\_test1** y **ax\_test2**

1. *ELEMENTOS DEL CANVAS*
2. Damos valor a los EJES: ax.**bar(**x\_valores,y\_valores**), ax.barh(**x,y**)**

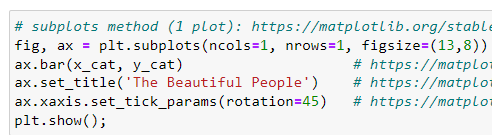
¡IMPORTANTE! No utilizamos dataFrame, sino **List** o **ARR**

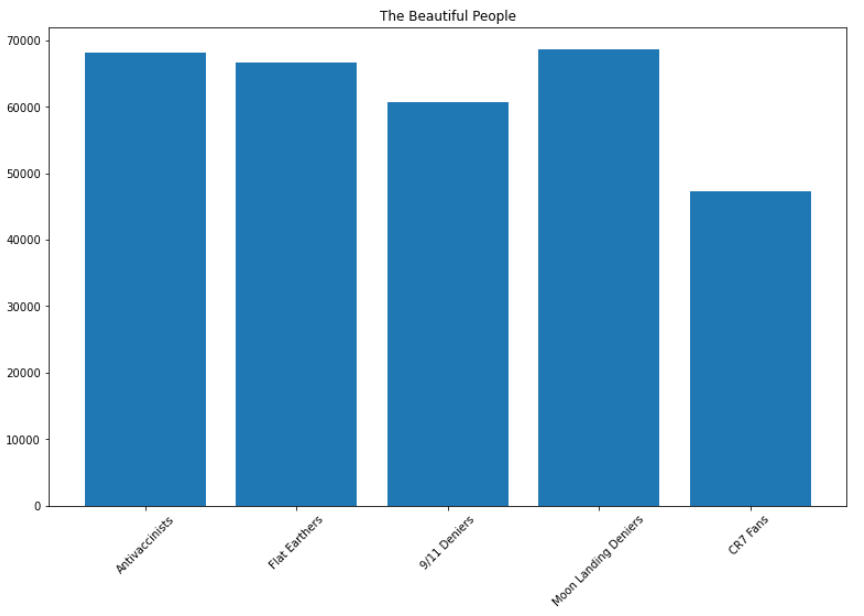


1. demás elementos que queramos: TÍTULO, TICKS, …

* ax[0,0].set\_title()
* ax[0,0].set\_xlabel(), ax[0,0].set\_ylabel(), …

*Para el siguiente ejemplo creamos un único “ax”*





\* Para mostrar el resultado de la celda ponemos **plt.show()**

**ALMACENAR** EL GRÁFICO EN UNA IMG O PDF



**Cambiar los valors del texto de los ejes → set\_xticklabels(), set\_xticklabels()**



**INCLUIR** EL VALOR DE LAS BARRAS (o cualquier tipo de gráfico)





**SEABORN**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

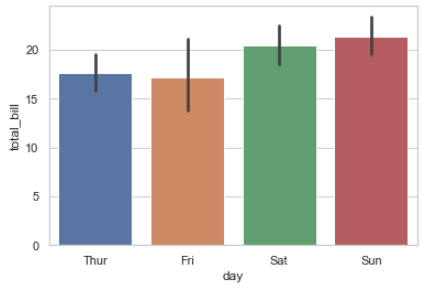
**sns.set()**

* **sns.***TYPE* **(x=**” “**,** **y=**” “**,** **data=** “List, array or dataFrame”**, resto de args)**

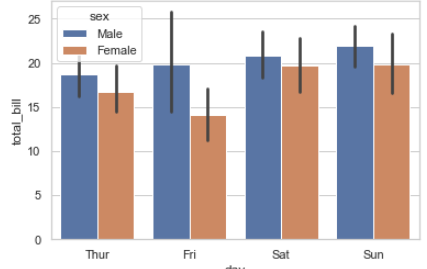
# EJEMPLO:

# 

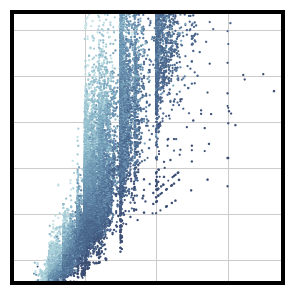
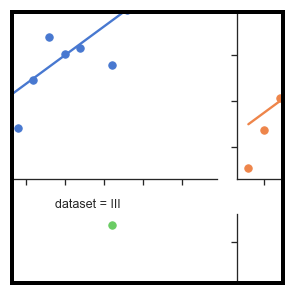
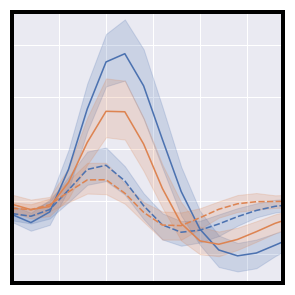
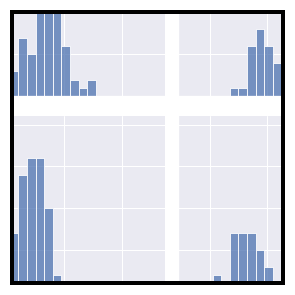
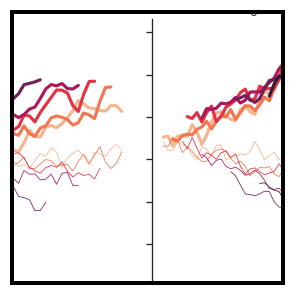
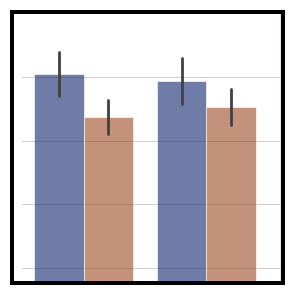
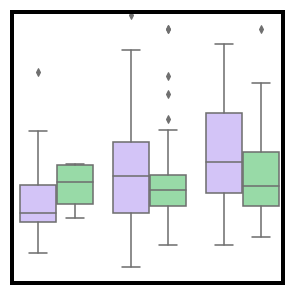
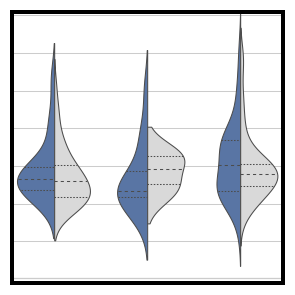
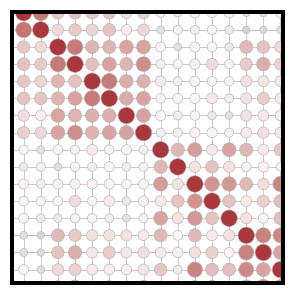
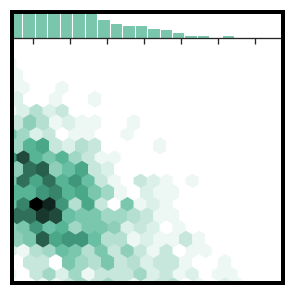
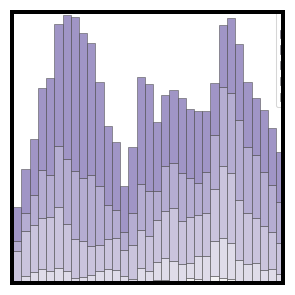
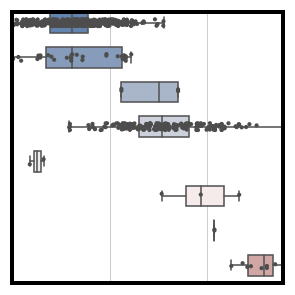
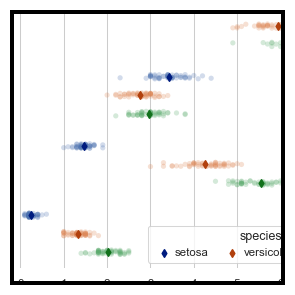
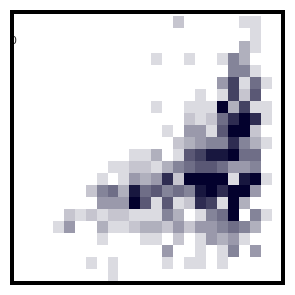
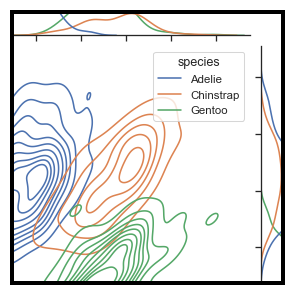
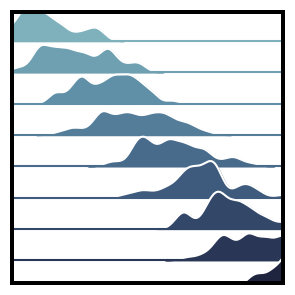
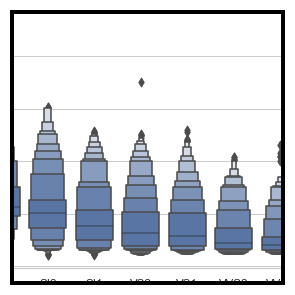
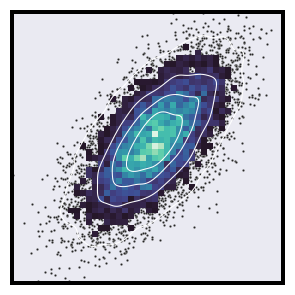
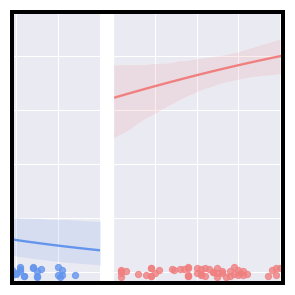
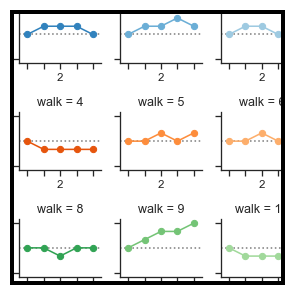
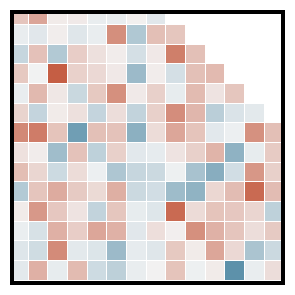
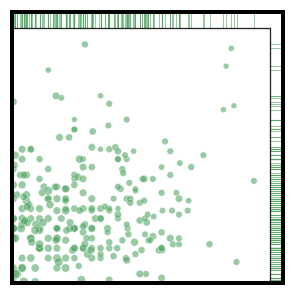
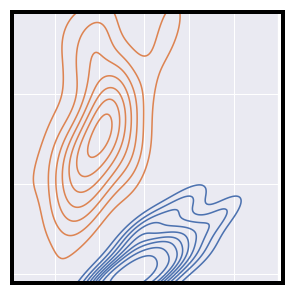
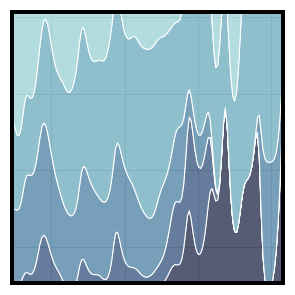
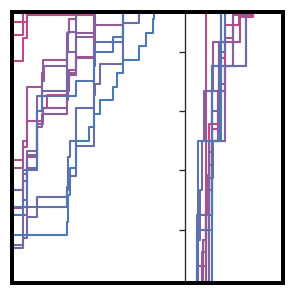
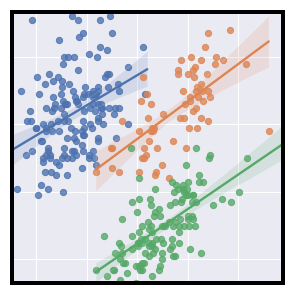
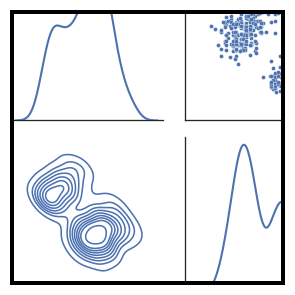
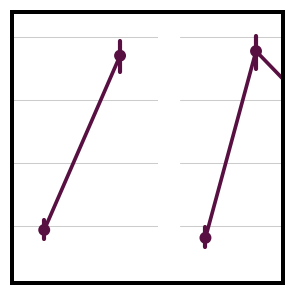
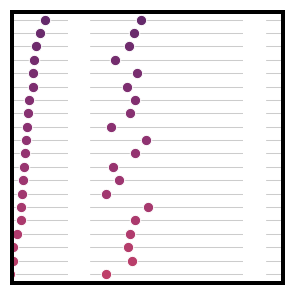
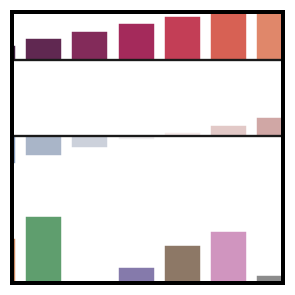
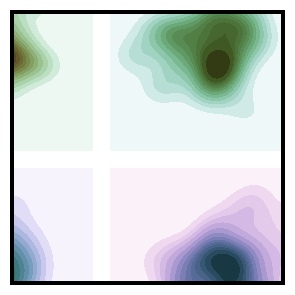
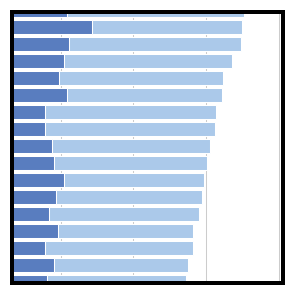
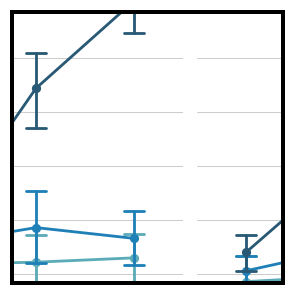
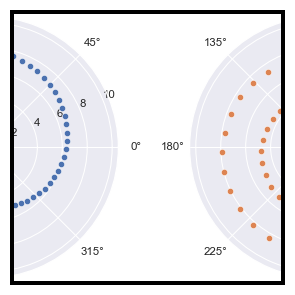
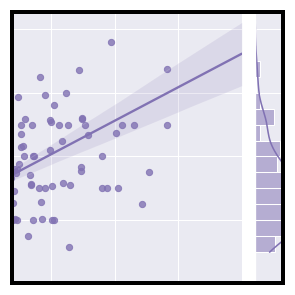
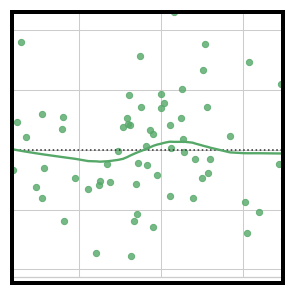
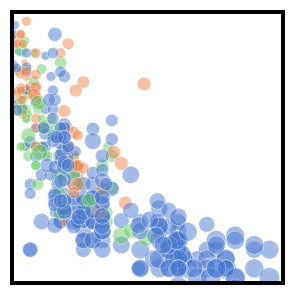
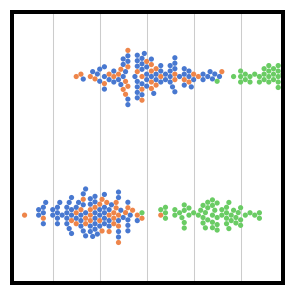
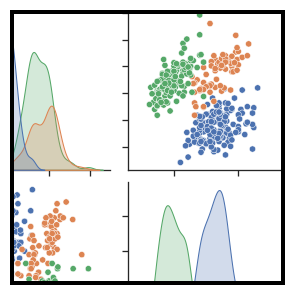
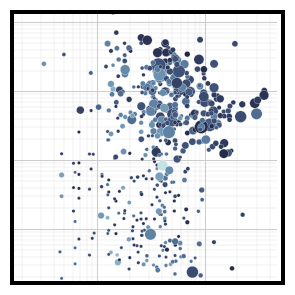
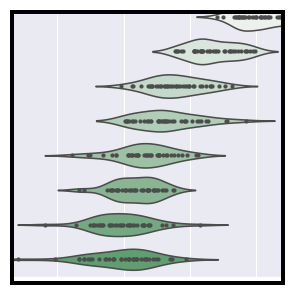
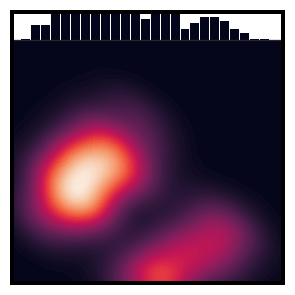
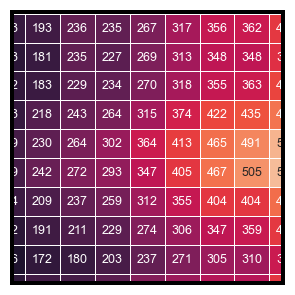
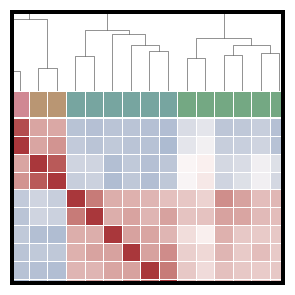
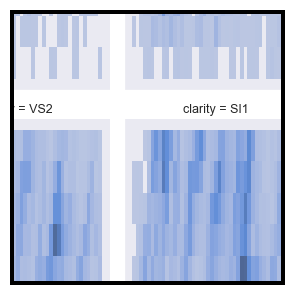
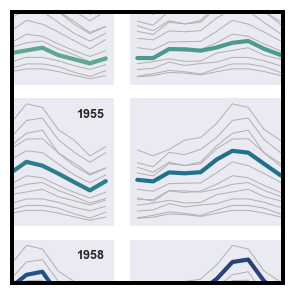
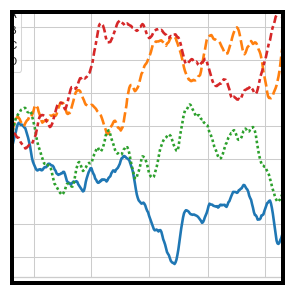
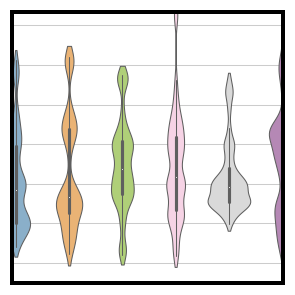
# 







# args → Example gallery

[](http://seaborn.pydata.org/examples/anscombes_quartet.html)[](http://seaborn.pydata.org/examples/different_scatter_variables.html)[](http://seaborn.pydata.org/examples/errorband_lineplots.html)[](http://seaborn.pydata.org/examples/faceted_histogram.html)[](http://seaborn.pydata.org/examples/faceted_lineplot.html)[](http://seaborn.pydata.org/examples/grouped_barplot.html)[](http://seaborn.pydata.org/examples/grouped_boxplot.html)[](http://seaborn.pydata.org/examples/grouped_violinplots.html)[](http://seaborn.pydata.org/examples/heat_scatter.html)[](http://seaborn.pydata.org/examples/hexbin_marginals.html)[](http://seaborn.pydata.org/examples/histogram_stacked.html)[](http://seaborn.pydata.org/examples/horizontal_boxplot.html)[](http://seaborn.pydata.org/examples/jitter_stripplot.html)[](http://seaborn.pydata.org/examples/joint_histogram.html)[](http://seaborn.pydata.org/examples/joint_kde.html)[](http://seaborn.pydata.org/examples/kde_ridgeplot.html)[](http://seaborn.pydata.org/examples/large_distributions.html)[](http://seaborn.pydata.org/examples/layered_bivariate_plot.html)[](http://seaborn.pydata.org/examples/logistic_regression.html)[](http://seaborn.pydata.org/examples/many_facets.html)[](http://seaborn.pydata.org/examples/many_pairwise_correlations.html)[](http://seaborn.pydata.org/examples/marginal_ticks.html)[](http://seaborn.pydata.org/examples/multiple_bivariate_kde.html)[](http://seaborn.pydata.org/examples/multiple_conditional_kde.html)[](http://seaborn.pydata.org/examples/multiple_ecdf.html)[](http://seaborn.pydata.org/examples/multiple_regression.html)[](http://seaborn.pydata.org/examples/pair_grid_with_kde.html)[](http://seaborn.pydata.org/examples/paired_pointplots.html)[](http://seaborn.pydata.org/examples/pairgrid_dotplot.html)[](http://seaborn.pydata.org/examples/palette_choices.html)[](http://seaborn.pydata.org/examples/palette_generation.html)[](http://seaborn.pydata.org/examples/part_whole_bars.html)[](http://seaborn.pydata.org/examples/pointplot_anova.html)[](http://seaborn.pydata.org/examples/radial_facets.html)[](http://seaborn.pydata.org/examples/regression_marginals.html)[](http://seaborn.pydata.org/examples/residplot.html)[](http://seaborn.pydata.org/examples/scatter_bubbles.html)[](http://seaborn.pydata.org/examples/scatterplot_categorical.html)[](http://seaborn.pydata.org/examples/scatterplot_matrix.html)[](http://seaborn.pydata.org/examples/scatterplot_sizes.html)[](http://seaborn.pydata.org/examples/simple_violinplots.html)[](http://seaborn.pydata.org/examples/smooth_bivariate_kde.html)[](http://seaborn.pydata.org/examples/spreadsheet_heatmap.html)[](http://seaborn.pydata.org/examples/structured_heatmap.html)[](http://seaborn.pydata.org/examples/three_variable_histogram.html)[](http://seaborn.pydata.org/examples/timeseries_facets.html)[](http://seaborn.pydata.org/examples/wide_data_lineplot.html)

**VISUALIZACIÓN DINÁMICA/INTERACTIVA:**

**a) CUFFLINKS**

Cufflinks es un envoltorio alrededor de Plotly que le permite producir fácilmente visualizaciones interactivas

**import cufflinks as cf**

**cf.go\_offline()**

* df**.iplot(***kind***= “**'hist'**”, x=”...”, y=”...”, xTitle='**...**', yTitle='**...**', Title='**...**', );**

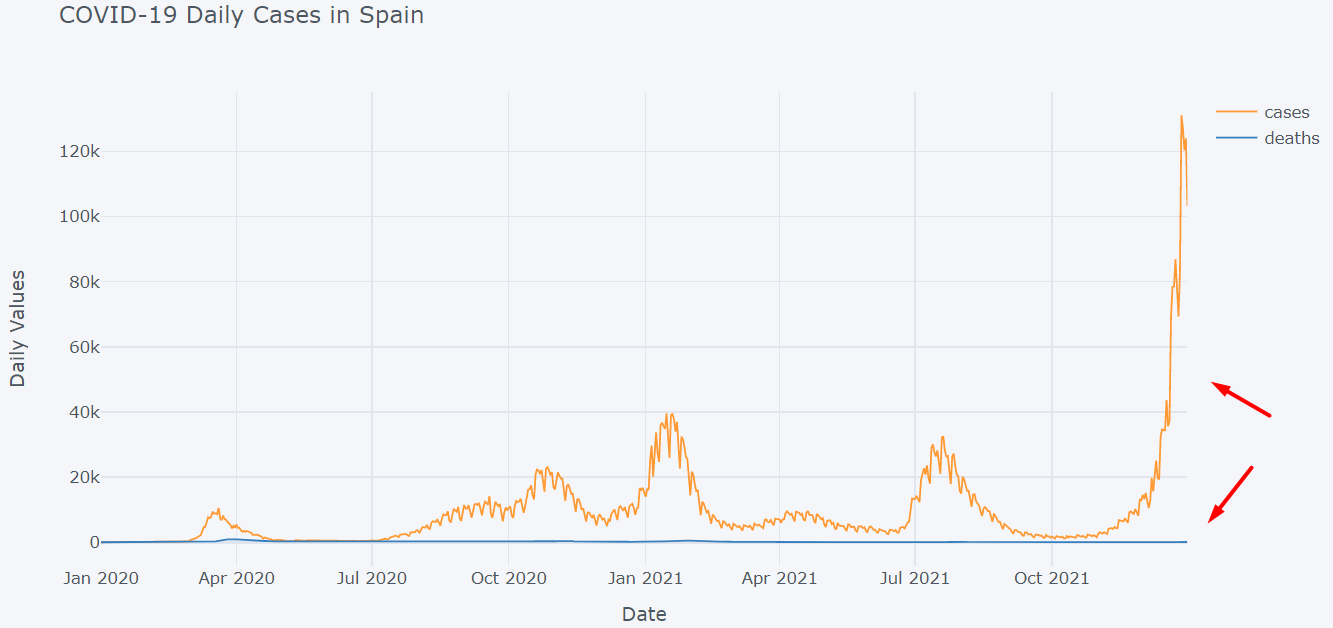
EJEMPLOS

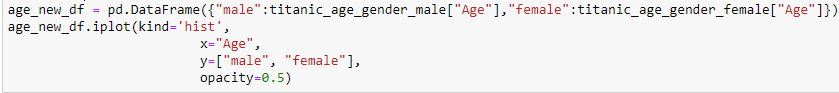
**UNO SOLO**

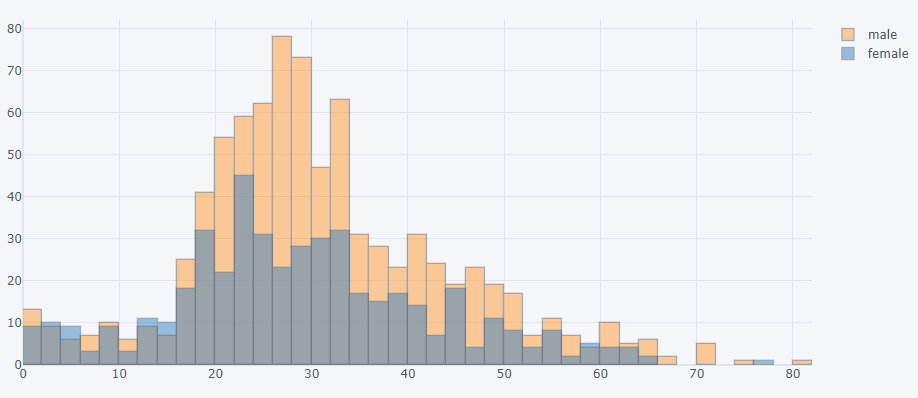
****

**DOS JUNTOS**

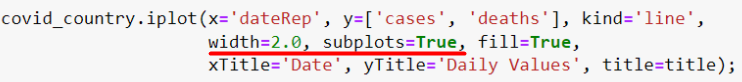
****

****

****

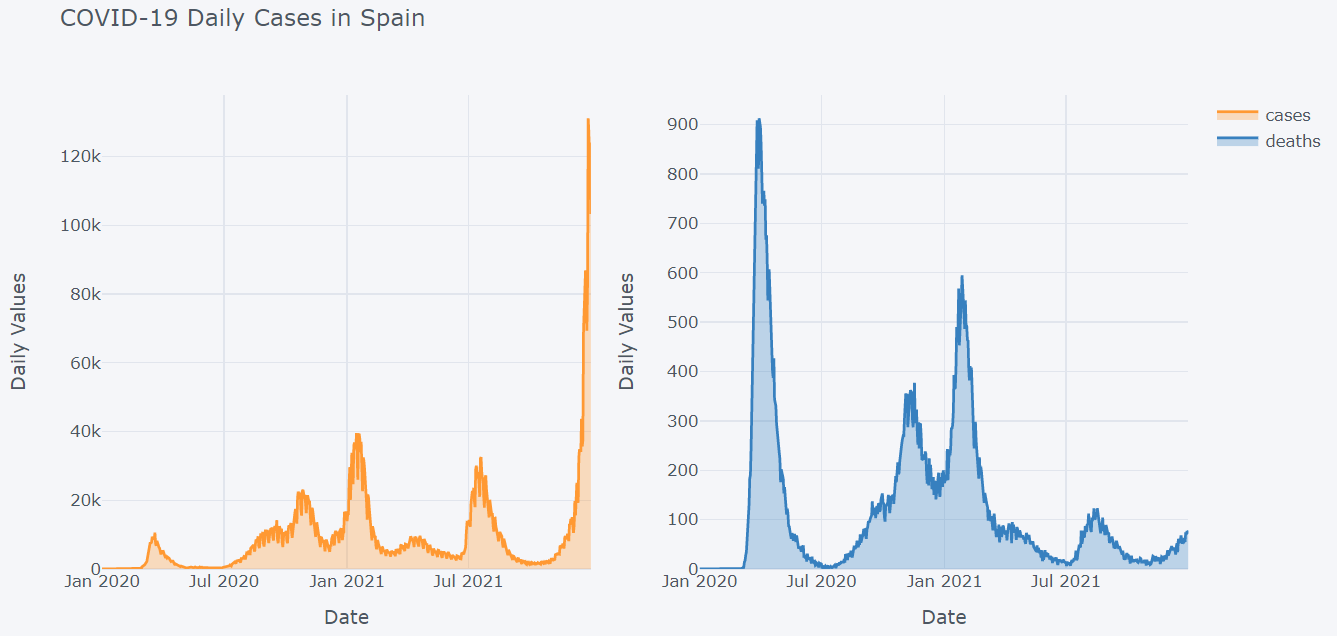
****

**SUBPLOTS**

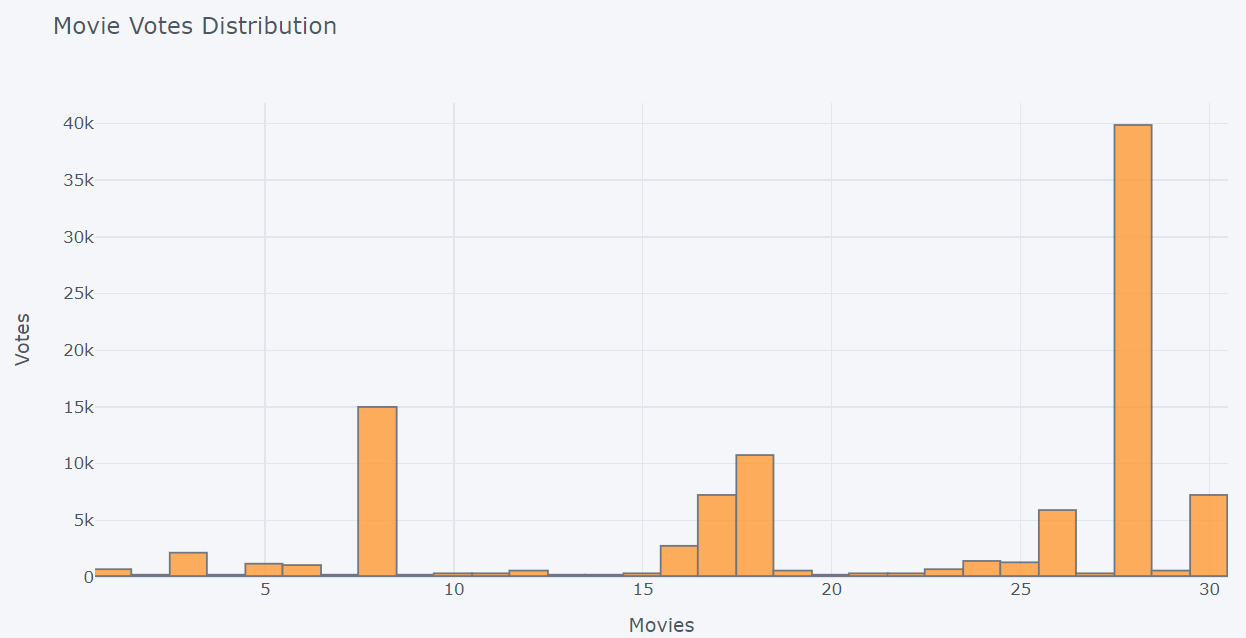


**\* subplots=True** → crea una cuadrícula de gráficos

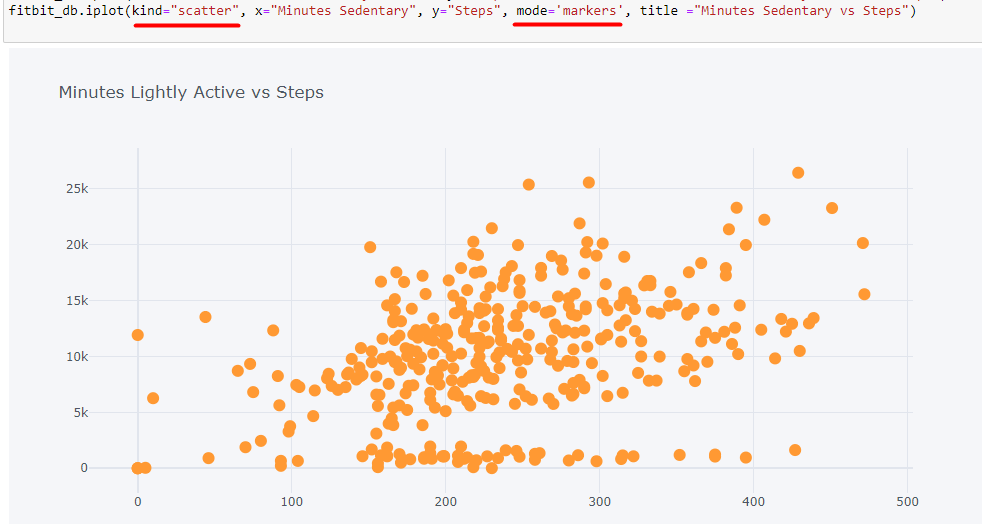
**\* fill=True** → rellena el área entre la línea y el eje

****

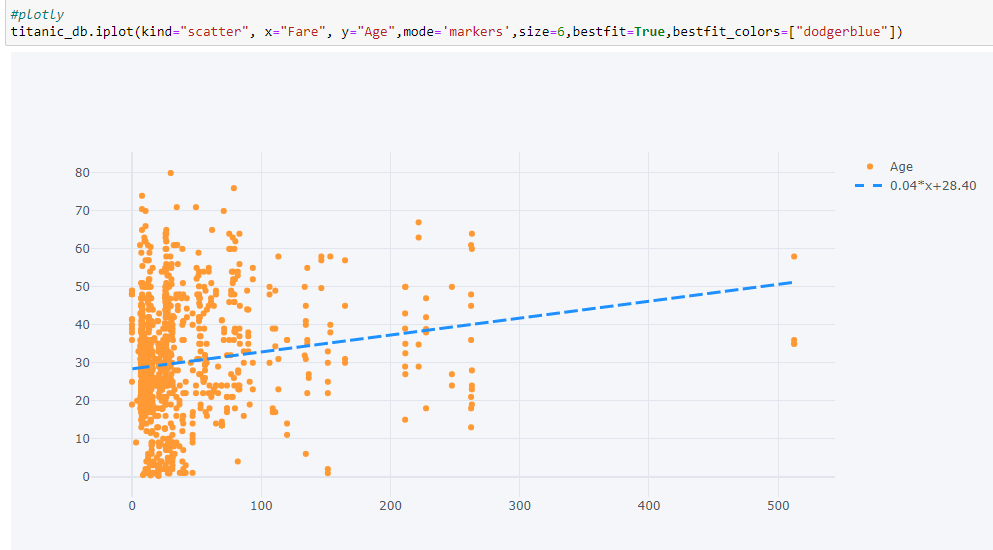
**De una columna de un dataframe**

****

**SCATTERN**

****

**\* mode='markers' →** para que sean puntos, si no serían lineas

****

**\* bestfit=True →** para crear la línea de tendencia

**b) PLOTLY**

Plotly es una biblioteca de visualización de código abierto

**import plotly.express as px**

**import plotly.io as pio**

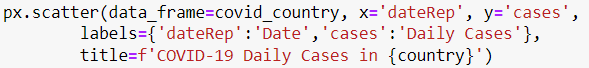
**from plotly.subplots import make\_subplots**

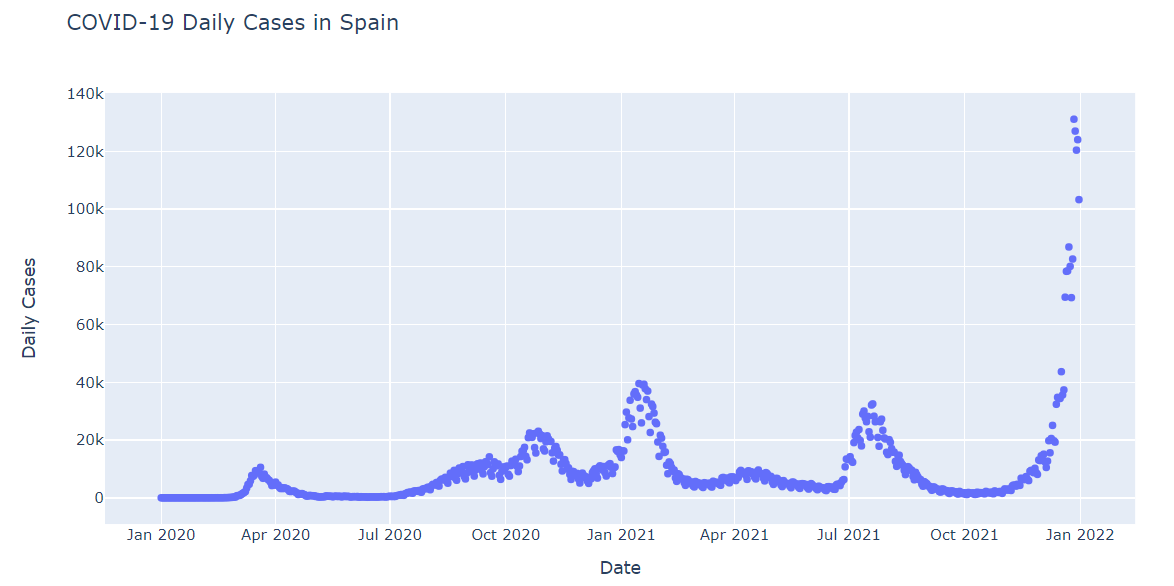
**import plotly.graph\_objects as go**

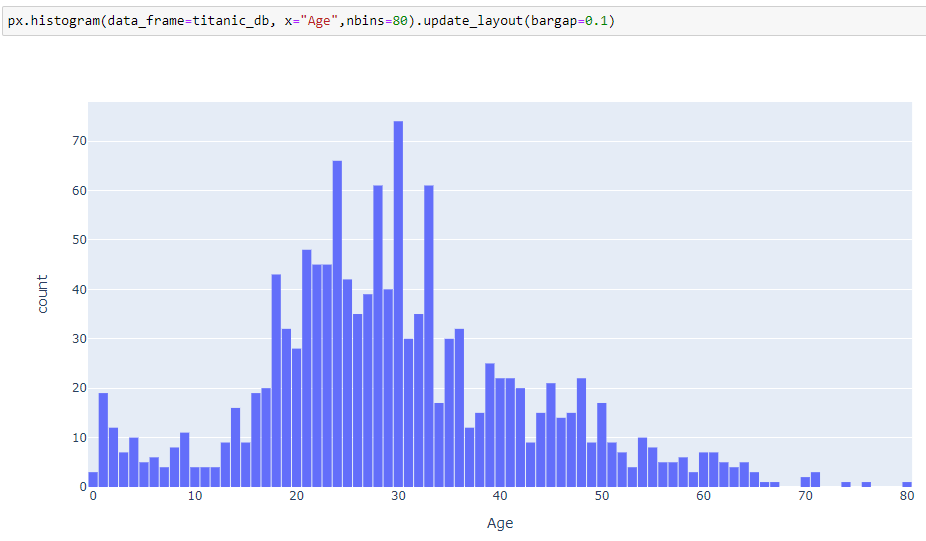
**px.***TYPE***(data\_frame='**...**, x='**...**', y='**...**', title='**...**', labels={**'xname'**:**'xvalue'**,**'yname'**:**'yvalue'**} );**

EJEMPLOS

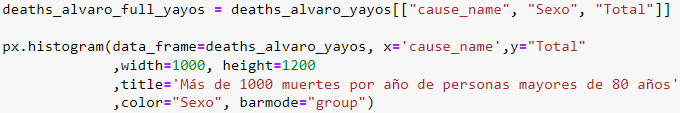
**UNO SOLO**

****

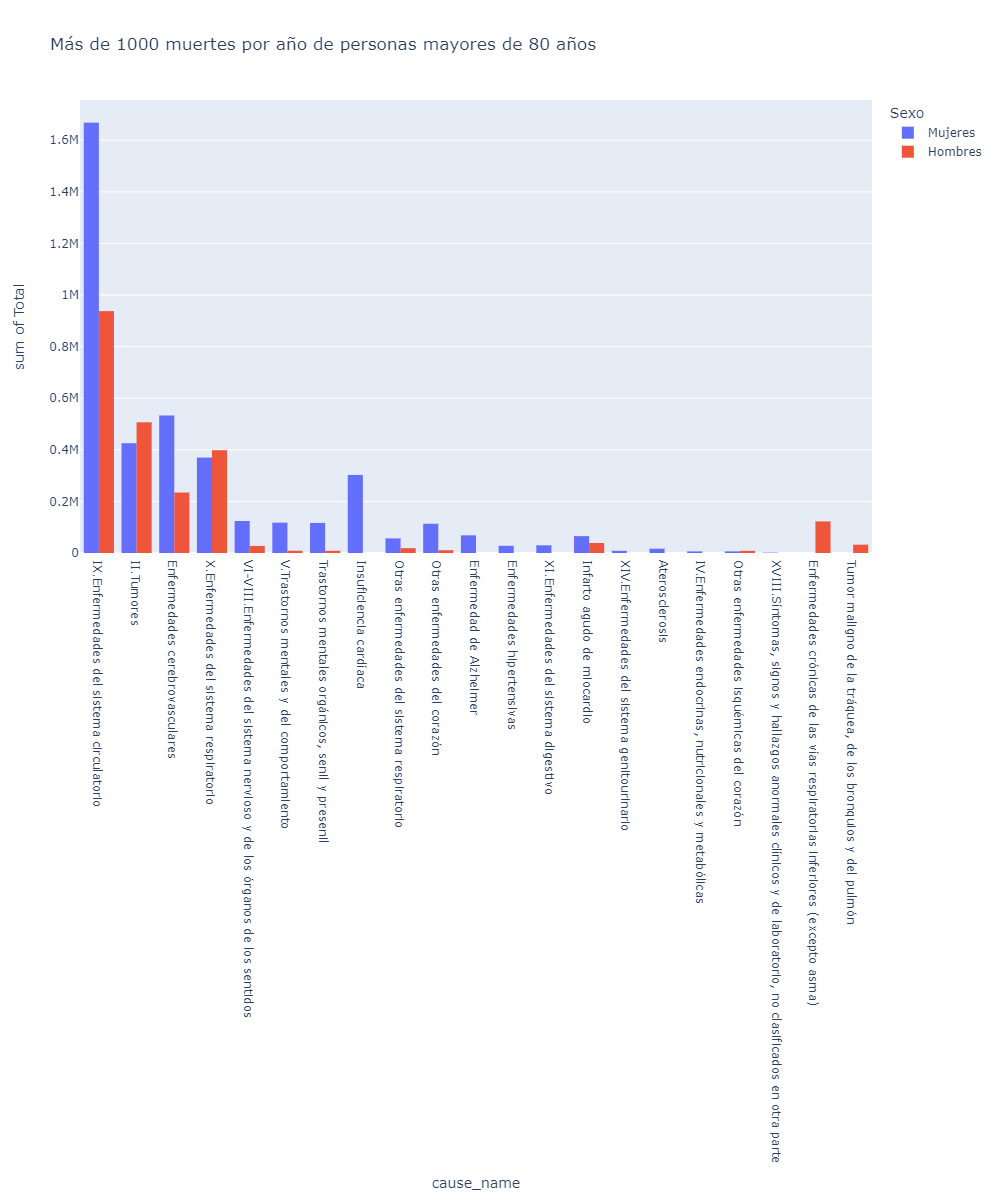
****

****

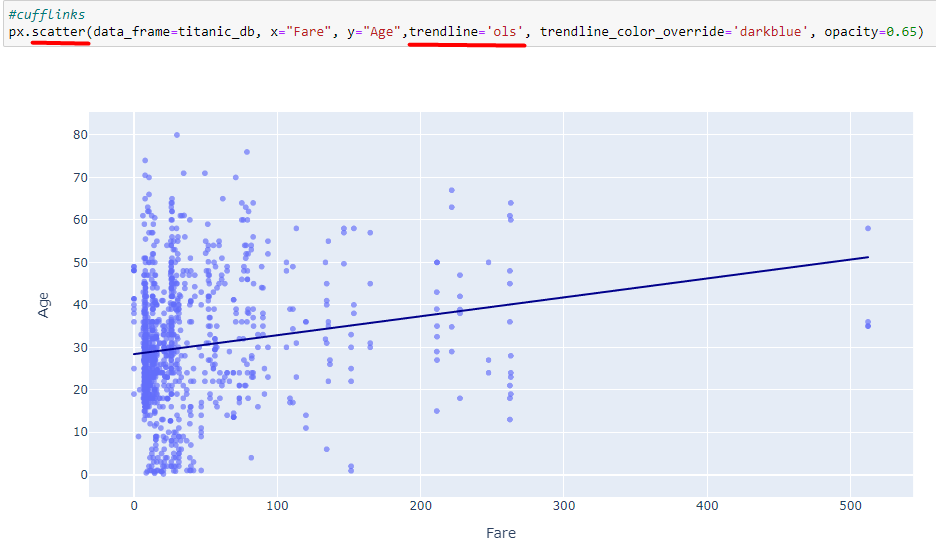
**DOS JUNTOS**

****

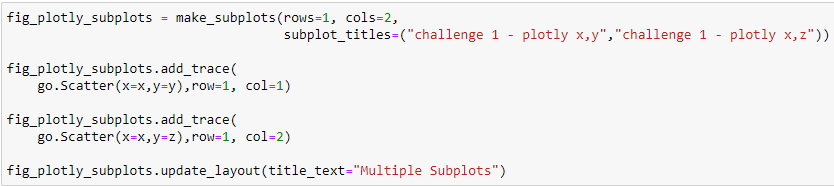
**\* barmode=”group” →** hace que las **barras** se vean **una al lado de la otra**, en vez de una encima de la otra

****

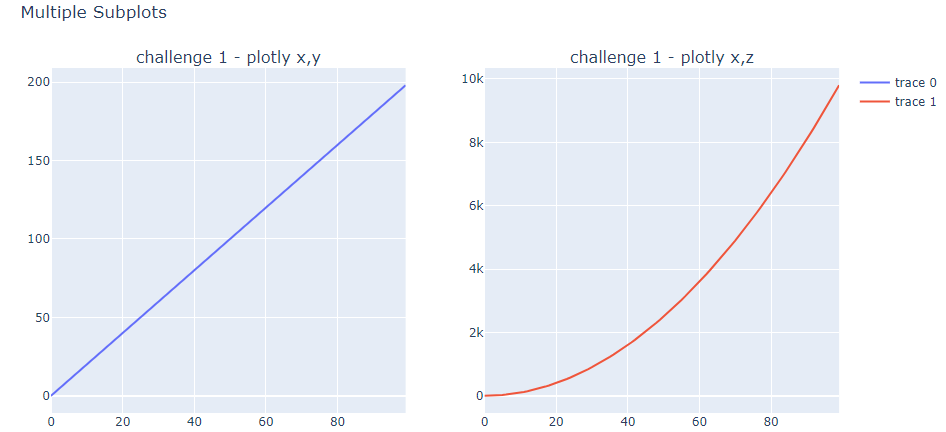
**SCATTER Y TENDENCIA**

****

**SUBPLOTS SEPARADOS**

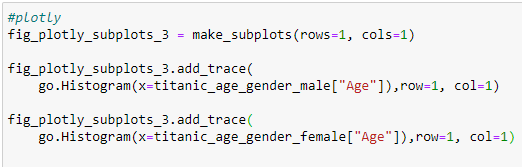
****

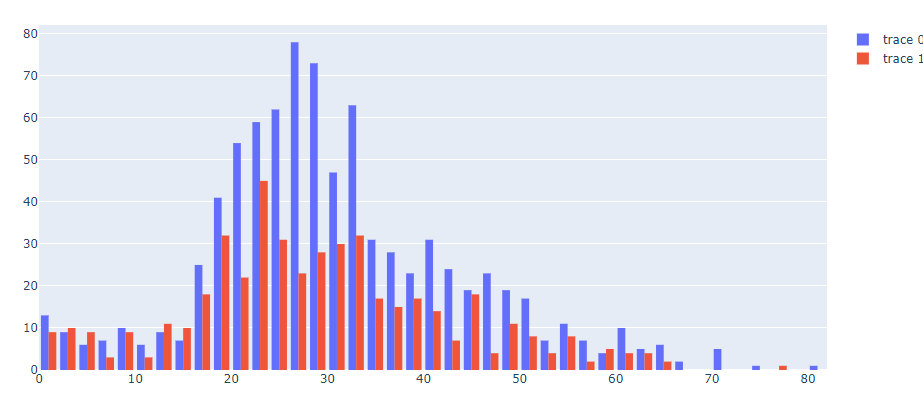
**\* make\_subplots() →** crea una cuadrícula con múltiples subplots

****

**SUBPLOTS UNIDOS (en este caso mejor intentar el ejemplo de “dos juntos”)**

**\* Aunque lo podemos utilizar para un único gráfico que bebe de diferentes dataframes**

****

****