# SEABORN

galería: https://seaborn.pydata.org/examples/index.html (https://seaborn.pydata.org/examples/index.html)

tutorial youtube: https://www.youtube.com/watch?v=GcXcSZ0gQps (https://www.youtube.com/watch?v=GcXcSZ0gQps)

intro: https://seaborn.pydata.org/tutorial/function_overview.html (https://seaborn.pydata.org/tutorial/function_overview.html)

1. Numerical Data Plotting

- relplot()
- scatterplot()
- lineplot()

2. Categorical Data Ploting

- catplot()
- boxplot()
- stripplot()
- swarmplot()
- etc…

3. Visualizing Distribution of the Data

- distplot()
- kdeplot()
- jointplot()
- rugplot()

4. Linear Regression and Relationship

- regplot()
- lmplot()

5. Controlling Plotted Figure Aesthetics

- figure styling
- axes styling
- color palettes

## 1. Numerical Data Plotting

- sns.relplot(x=, y=, data=, hue=, style=, palette=, size=, sizes= (n, n), kind=, sort=)

**Hue** cambia el color de los puntos, **style** cambia la forma de los puntos, **size** cambia el tamaño de los puntos. **Sizes** determina el rango de tamaños de los puntos en la gráfica. **Kind** determina el tipo de representación (por defecto es un scatter plot). **Sort**: si es True ordena los puntos a partir de ejeX, si es False no se ordena

In [2]:
```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:

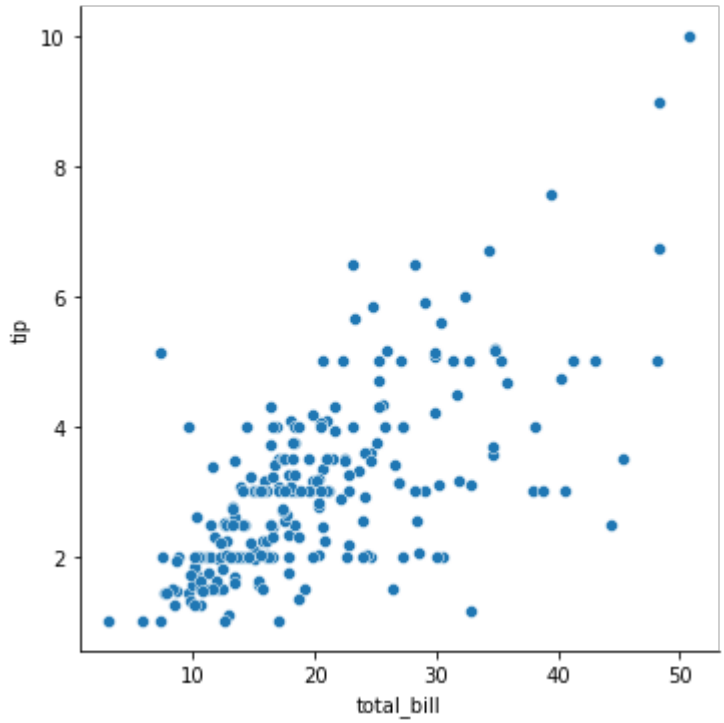
In [4]:
```
tips = sns.load_dataset('tips')
```

Out[4]:

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **239** | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 |

In [5]:

Out[5]:
```
<seaborn.axisgrid.FacetGrid at 0x1f016dbde80>
```




In [6]:

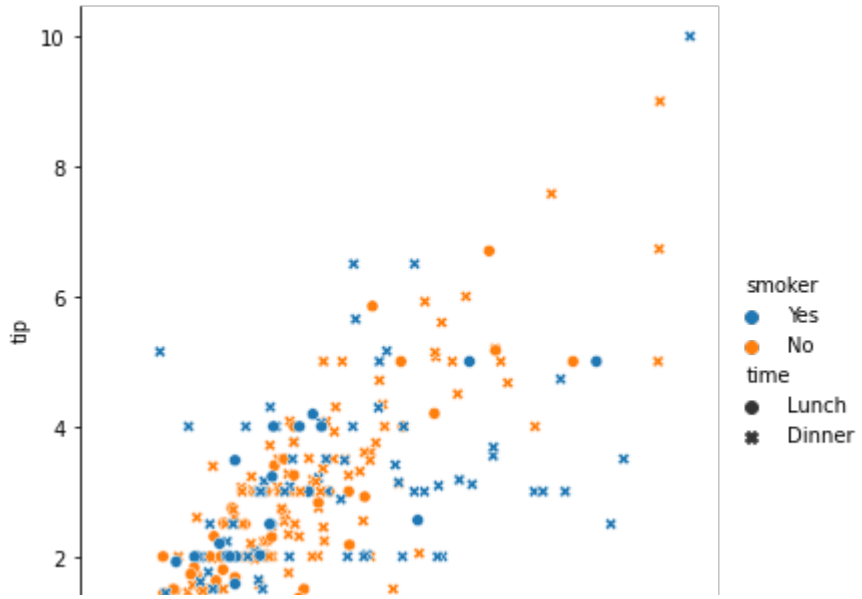
Out[6]:

```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_bottom_axes',
 '_clean_axis',
 '_facet_color',
 '_facet_plot',
 '_finalize_grid',
 '_get_palette',
 '_inner_axes',
 '_left_axes',
 '_legend_out',
 '_margin_titles',
 '_not_bottom_axes',
 '_not_left_axes',
 '_update_legend_data',
 'add_legend',
 'ax',
 'axes',
 'axes_dict',
 'despine',
 'facet_axis',
 'facet_data',
 'fig',
 'legend',
 'map',
 'map_dataframe',
 'savefig',
 'set',
 'set_axis_labels',
 'set_titles',
 'set_xlabels',
 'set_xticklabels',
```

'set_ylabels'.

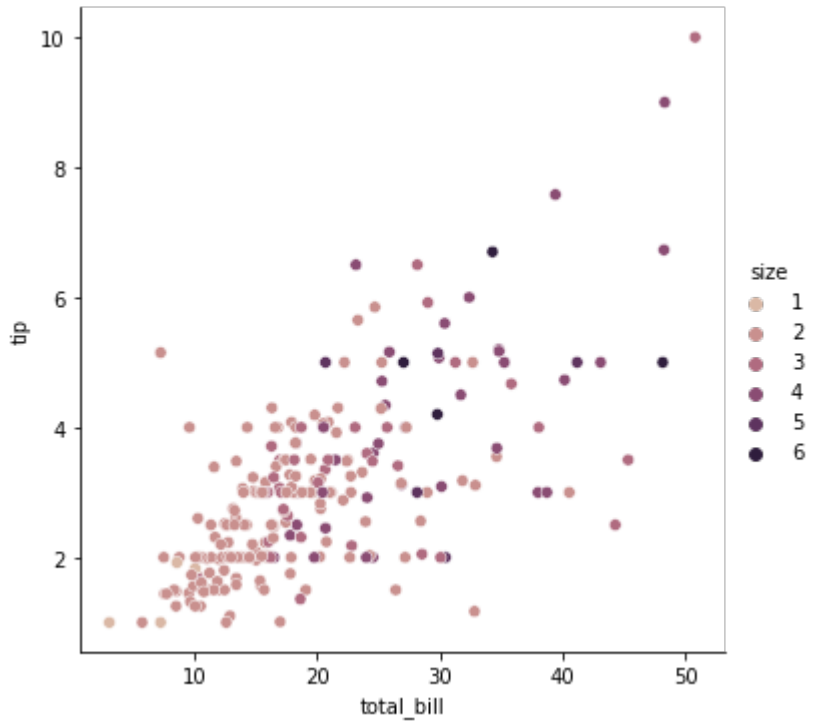
In [9]:
```
sns.relplot(x = 'total_bill', y = 'tip', data=tips, hue = 'smoker', style
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x1f016f93190>
```




In [11]:
```
sns.relplot(x = 'total_bill', y = 'tip', data=tips, hue = 'size', palette
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x1f0175f6bb0>
```




https://www.youtube.com/watch?v=6GUZXDef2U0 (https://www.youtube.com/watch?v=6GUZXDef2U0)

# Distribution Plots

Provide a distribution for one variable (univariate distribution).

- kde= : si quieres la línea de densidad o no.
- bins= : como de anchas queremos las barras del historiograma.

In [3]:
```
crash_df = sns.load_dataset('car_crashes')
```

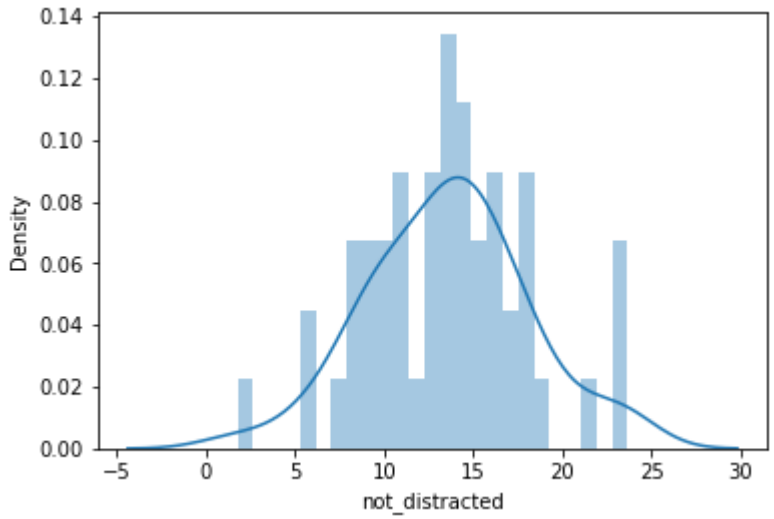
Out[3]:

| | total | speeding | alcohol | not_distracted | no_previous | ins_premium | ins_losses | abbrev |
|---|---|---|---|---|---|---|---|---|
| **0** | 18.8 | 7.332 | 5.640 | 18.048 | 15.040 | 784.55 | 145.08 | AL |
| **1** | 18.1 | 7.421 | 4.525 | 16.290 | 17.014 | 1053.48 | 133.93 | AK |
| **2** | 18.6 | 6.510 | 5.208 | 15.624 | 17.856 | 899.47 | 110.35 | AZ |
| **3** | 22.4 | 4.032 | 5.824 | 21.056 | 21.280 | 827.34 | 142.39 | AR |
| **4** | 12.0 | 4.200 | 3.360 | 10.920 | 10.680 | 878.41 | 165.63 | CA |

In [22]:
```
sns.distplot(crash_df['not_distracted'],bins=25)
```

```
C:\Users\elgab\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[22]: <AxesSubplot:xlabel='not_distracted', ylabel='Density'>



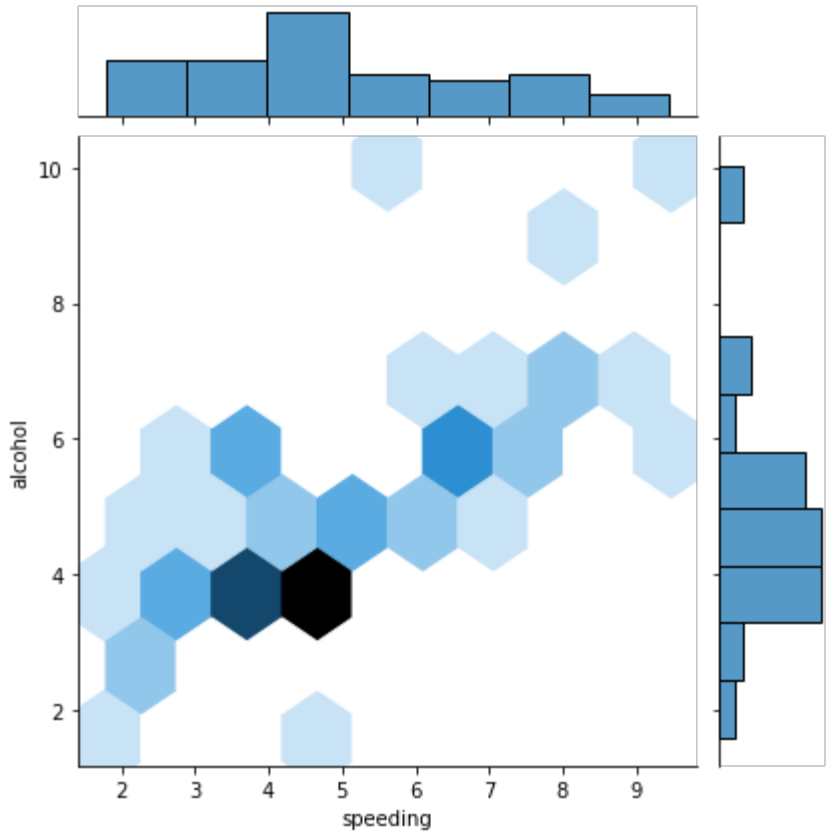

# Joint Plot

Used to compare two variables.

- kind= : diferentes tipos de gráfica.

In [38]:

Out[38]: <seaborn.axisgrid.JointGrid at 0x1f01c4ecdf0>

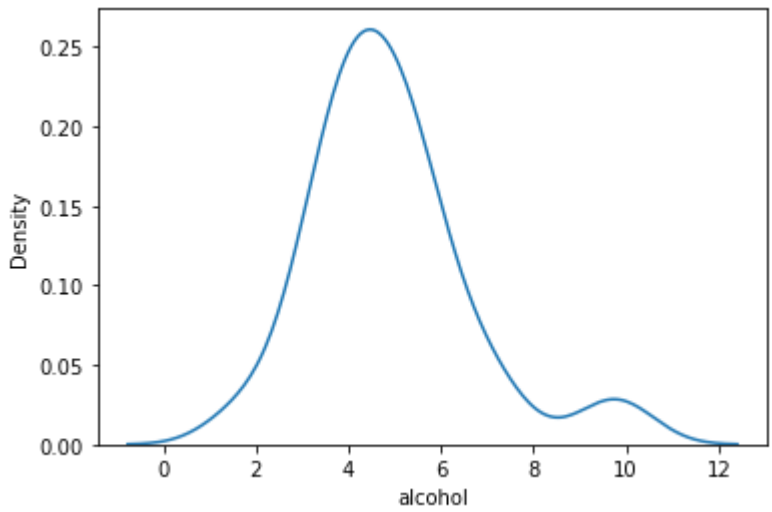



# KDE Plot

Representa la curva de densidad.

In [25]:

Out[25]: <AxesSubplot:xlabel='alcohol', ylabel='Density'>

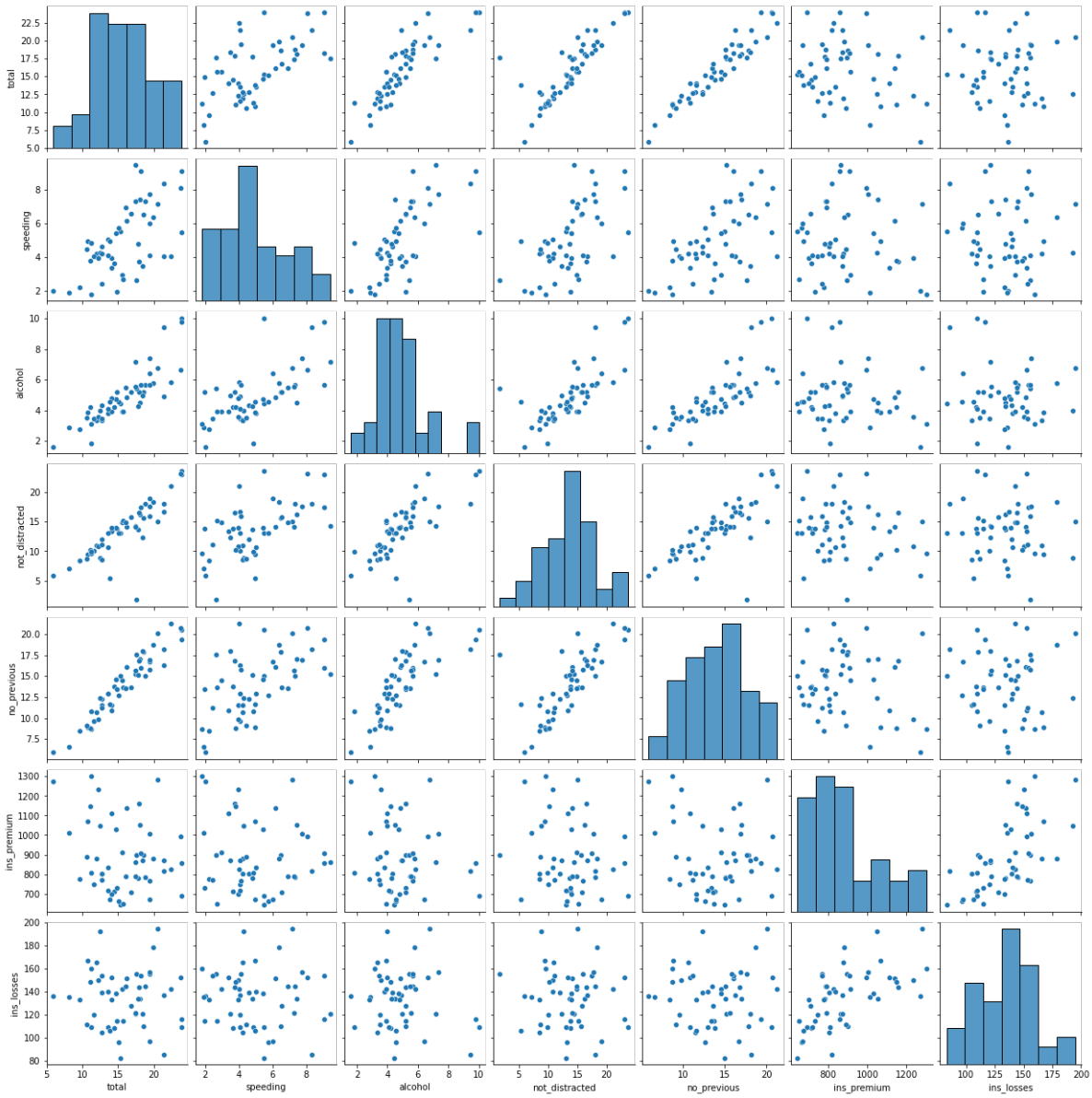



# Pair Plots

Plots the relationship across the entire dataframe's numerical values.

In [26]:

Out[26]: <seaborn.axisgrid.PairGrid at 0x1f01927cd90>

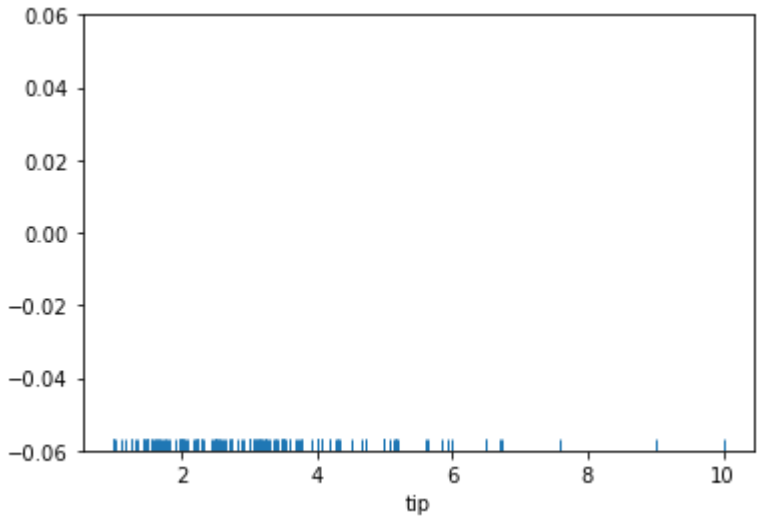



# Rug Plot

Plot a single column of data points in a dataframe as sticks. You'll see a higher density of sticks where the values are more common.

In [27]:

Out[27]: <AxesSubplot:xlabel='tip'>




In [ ]:

# Style

**sns.set_style('')** : el estilo de la gráfica.

Opciones: white, darkgrid, whitegrid, dark, ticks.

**sns.set_context('', font_scale=)** : lo que hay alrededor de la gráfica.

Opciones: paper (jupyter notebook), talk (presentación), poster.

**sns.despint(left=, bottom=, top=, right=)** : los contornos de la gráfica.

True or false.

**.set_theme(context='notebook', style='darkgrid', palette='deep', font='sans-serif', font_scale=1, color_codes=True, rc=None)**

sns.set(rc={'figure.figsize':(11.7,8.27)}) --> **set size**

https://seaborn.pydata.org/generated/seaborn.set_theme.html#seaborn.set_theme (https://seaborn.pydata.org/generated/seaborn.set_theme.html#seaborn.set_theme)

# Set size

plt.figure(figsize=(8,5))

(mirar strip plot)

## Rotar xticks

plt.xticks(rotation=45)

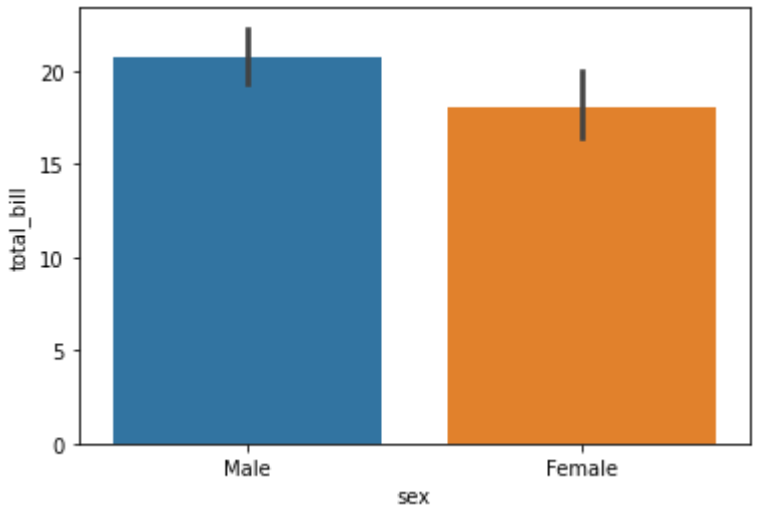
# Categorical Plots

Prestamos atención a datos categóricos en relación a una variable numérica.

## Bar Plots

In [29]:

Out[29]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>

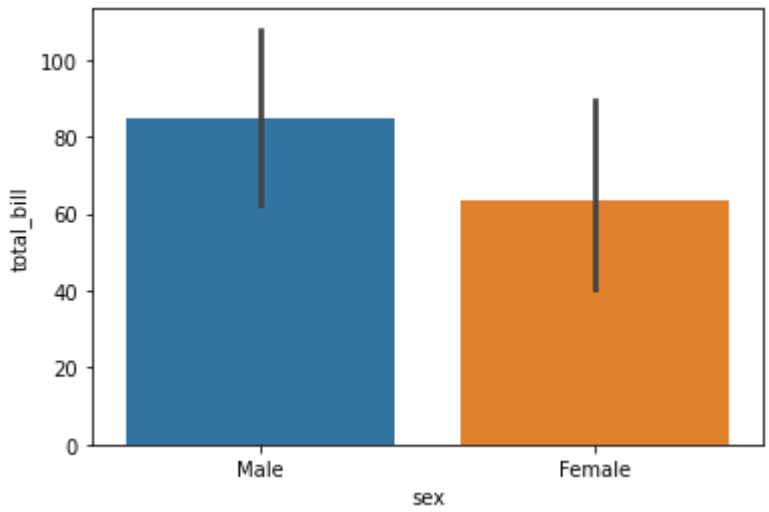



La barrita hace referencia a la varianza.

Las barras se basan por defecto en la media, pero también se puede setear en base a muchas otras funciones de numpy como la mediana, etc.

In [31]:

Out[31]: <AxesSubplot:xlabel='sex', ylabel='total_bill'>

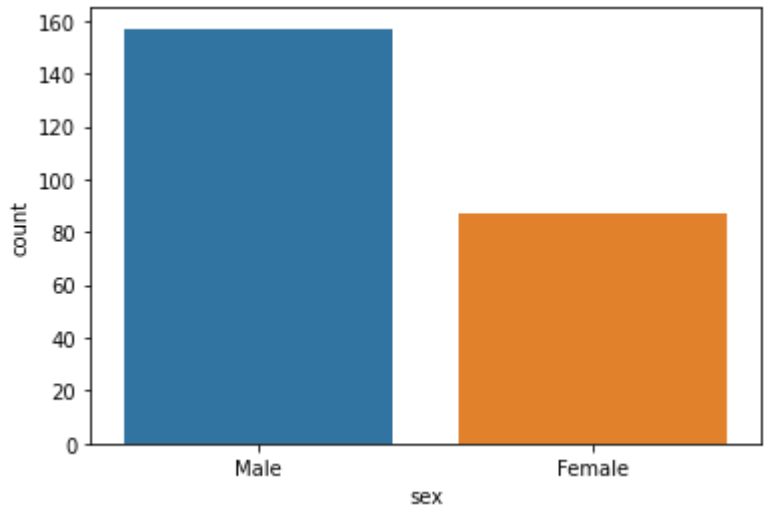



## Count Plot

Es parecido a un bar plot pero va a representar en base al Nº of ocurrences (count).

In [32]:

Out[32]: <AxesSubplot:xlabel='sex', ylabel='count'>




## Box Plot

Compare different variables and show the quartiles of the data.

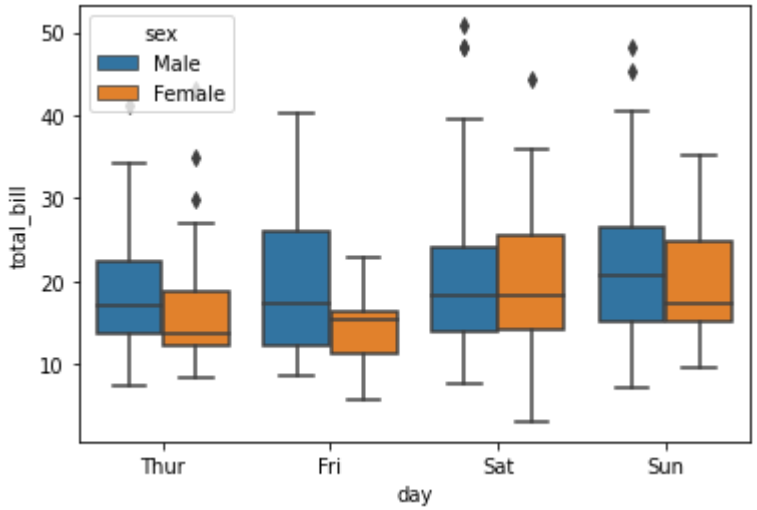
La barrita de en medio representa la media.

La barra equivale a una desviación estándar respecto a la media.

Los bigotes (whiskers) son el resto del dato excepto los outliers, los cuales podemos ver en forma

In [34]:

Out[34]: <AxesSubplot:xlabel='day', ylabel='total_bill'>

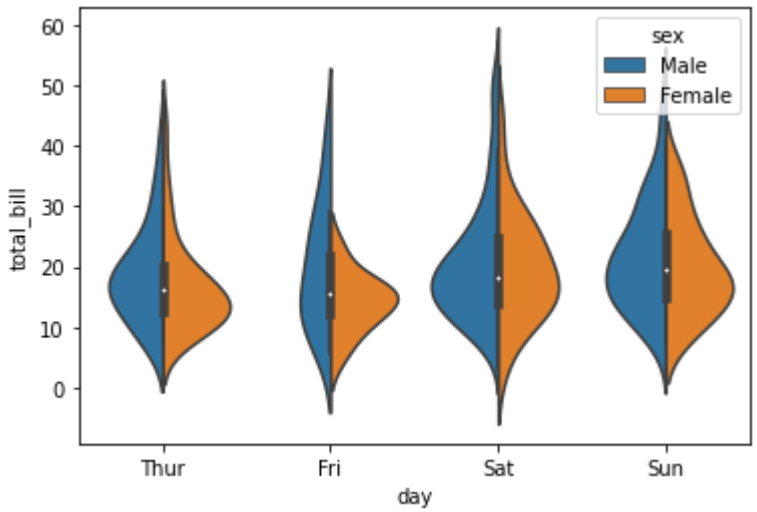



## Violin Plot

Es una combinación de un box plot y un kde plot. (confuso)

In [36]:

Out[36]: <AxesSubplot:xlabel='day', ylabel='total_bill'>

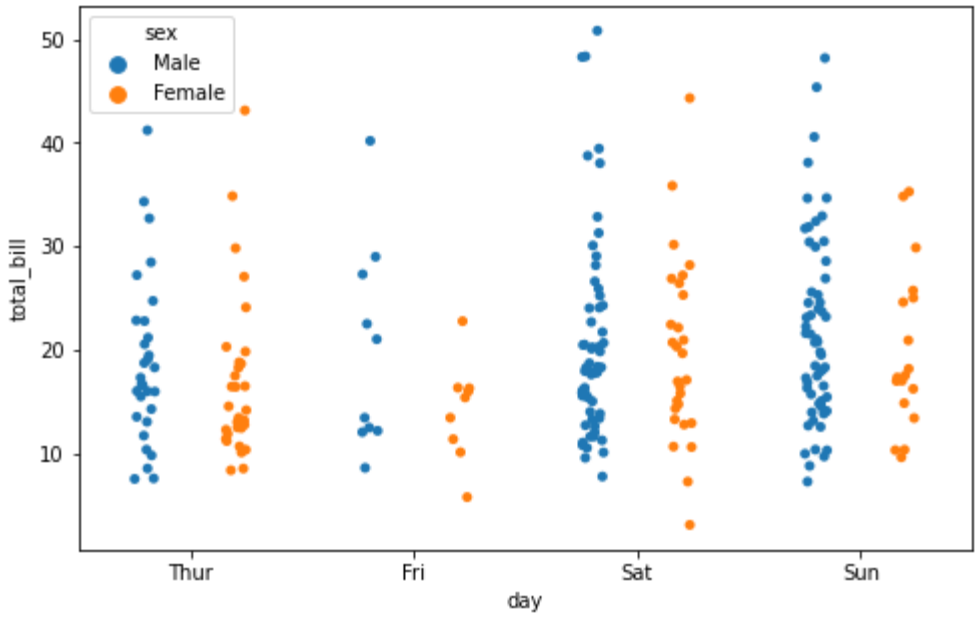



## Strip Plot

Scattarplot where one variable is categorical. Normalmente va asociado a un boxplot. Representar el avarage distribution.

In [7]:
```
tips_df = sns.load_dataset('tips')
plt.figure(figsize=(8,5))

#jitter esparce los puntos, dodge hace que separe los puntos en base al h
sns.stripplot(x='day', y='total_bill', data=tips_df, jitter=True, hue='se
```

Out[7]: <AxesSubplot:xlabel='day', ylabel='total_bill'>




In [ ]:

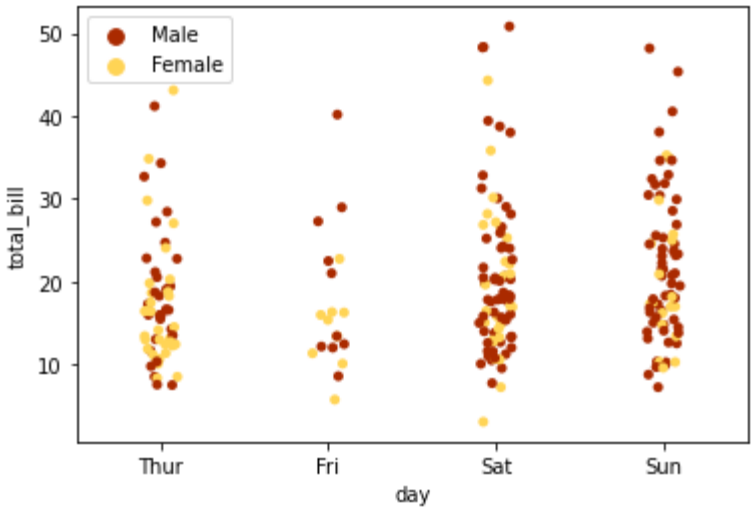
# PALETTES

Colormaps: https://matplotlib.org/stable/tutorials/colors/colormaps.html (https://matplotlib.org/stable/tutorials/colors/colormaps.html)

Legends: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html (https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html)

In [9]:
```
sns.stripplot(x='day', y='total_bill', data=tips_df, hue='sex', palette='
plt.legend(loc=0)#CAMBIA EL LUGAR DE LA LEYENDA DE MALE/FEMALE
```

Out[9]:
```
<matplotlib.legend.Legend at 0x1bd904b1c40>
```




# MATRIX PLOTS

## HEATMAPS

Representa la correlación entre las variables!!!

- fmt: formatear las anotaciones
- tutorial completito de heatmaps: https://indianaiproduction.com/seaborn-heatmap/ (https://indianaiproduction.com/seaborn-heatmap/)

Códigos de formateo:

The available integer presentation types are:

| Type | Meaning |
|---|---|
| `'b'` | Binary format. Outputs the number in base 2. |
| `'c'` | Character. Converts the integer to the corresponding unicode character before printing. |
| `'d'` | Decimal Integer. Outputs the number in base 10. |
| `'o'` | Octal format. Outputs the number in base 8. |
| `'x'` | Hex format. Outputs the number in base 16, using lower-case letters for the digits above 9. |
| `'X'` | Hex format. Outputs the number in base 16, using upper-case letters for the digits above 9. In case `'#'` is specified, the prefix `'0x'` will be upper-cased to `'0X'` as well. |
| `'n'` | Number. This is the same as `'d'`, except that it uses the current locale setting to insert the appropriate number separator characters. |

| | |
|---|---|
| None | The same as `'d'`. |

https://docs.python.org/3/library/string.html?highlight=string#formatspec (https://docs.python.org/3/library/string.html?highlight=string#formatspec)

In [14]:
```
plt.figure(figsize= (8,6))
sns.set_context('paper', font_scale=1.4)

#annot hace que aparezcan los números en medio de las casillas
#cmap se refiere al colormap
```

Out[14]: <AxesSubplot:>


In [3]:
```
import seaborn as sns
import pandas as pd
flights = sns.load_dataset('flights')
```

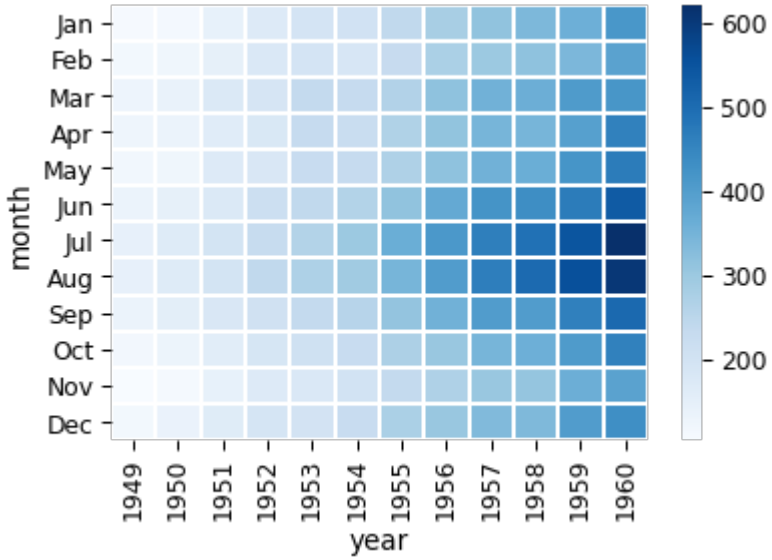
Out[3]:

| | year | month | passengers |
|---|---|---|---|
| **0** | 1949 | Jan | 112 |
| **1** | 1949 | Feb | 118 |
| **2** | 1949 | Mar | 132 |
| **3** | 1949 | Apr | 129 |
| **4** | 1949 | May | 121 |
| **...** | ... | ... | ... |

| | year | month | passengers |
|---|---|---|---|
| **139** | 1960 | Aug | 606 |
| **140** | 1960 | Sep | 508 |
| **141** | 1960 | Oct | 461 |
| **142** | 1960 | Nov | 390 |
| **143** | 1960 | Dec | 432 |

In [20]:
```
flights = sns.load_dataset('flights')
flights = flights.pivot_table(index='month', columns='year', values='pass
#linecolor + linewidth pone la línea de separación entre cuadritos
sns.heatmap(flights, cmap='Blues', linecolor='white', linewidth=1)
```

Out[20]: <AxesSubplot:xlabel='year', ylabel='month'>

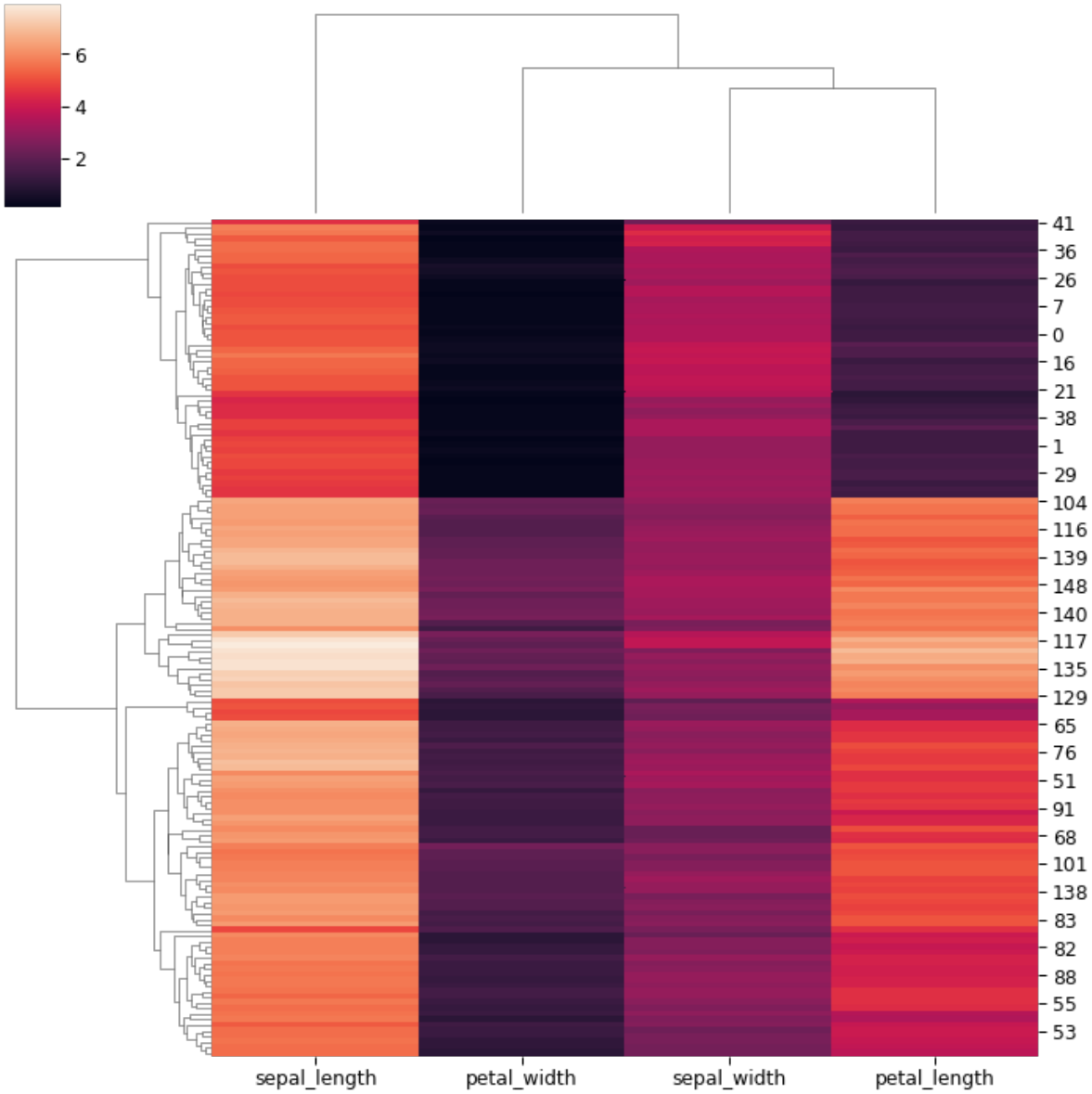



# CLUSTER MAP

((volver a ver esta parte del vídeo más adelante cuando sepa más, min 40 aprox))

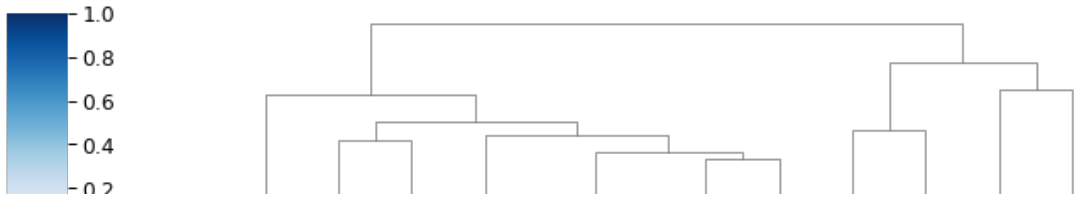
In [23]:
```
iris = sns.load_dataset('iris')
species = iris.pop('species')
```

Out[23]: <seaborn.matrix.ClusterGrid at 0x1bd9048dbb0>




In [24]:

Out[24]: <seaborn.matrix.ClusterGrid at 0x1bd90b8ac10>

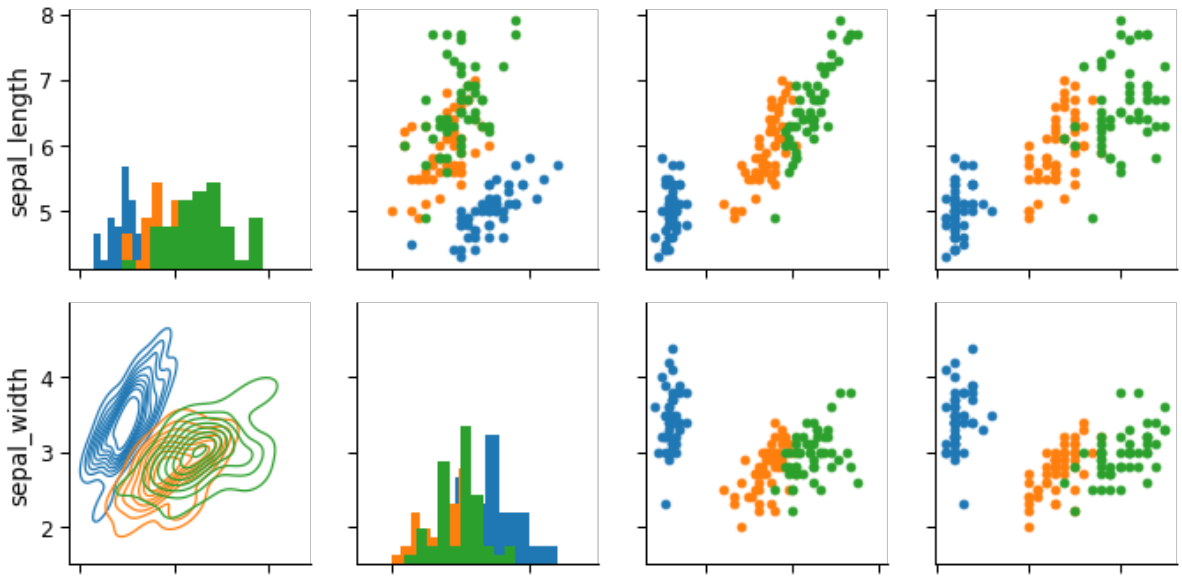



# PAIR GRIDS

Pairplot pero controlado (mirar más tarde también)

In [28]:
```
iris = sns.load_dataset('iris')
iris_g = sns.PairGrid(iris, hue='species')

iris_g.map_diag(plt.hist)
#iris_g.map_offdiag(plt.scatter)
iris_g.map_upper(plt.scatter)
```

Out[28]: <seaborn.axisgrid.PairGrid at 0x1bd93bd0ee0>




# REGRESSION PLOT

Type *Markdown* and LaTeX: $\alpha^2$

https://www.geeksforgeeks.org/how-to-set-axes-labels-limits-in-a-seaborn-plot/ (https://www.geeksforgeeks.org/how-to-set-axes-labels-limits-in-a-seaborn-plot/)

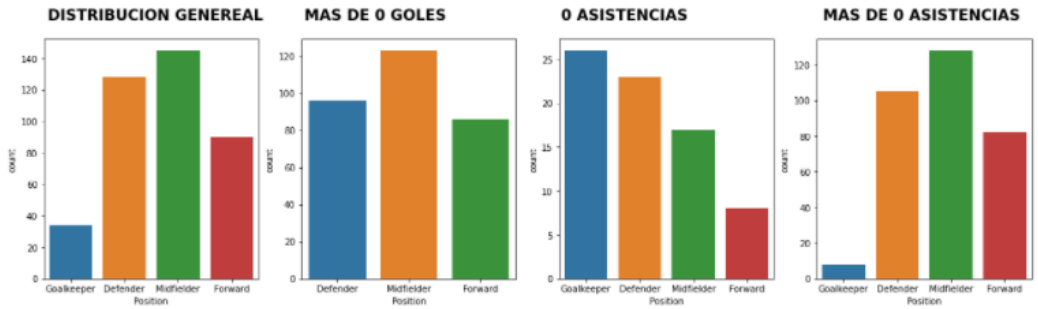
# EXTRA

Annotate subplots: https://www.tutorialspoint.com/annotate-subplots-in-a-figure-with-a-b-c-using-matplotlib (https://www.tutorialspoint.com/annotate-subplots-in-a-figure-with-a-b-c-using-

matplotlib)

```
In [40]: fig, axs = plt.subplots(ncols=4, nrows=1, figsize=(20,5))


sns.countplot(x=premier.Position, ax=axs[0], order=['Goalkeeper', 'Defender', 'Midfielder', 'Forward'])
axs[0].text(.01, .85, 'DISTRIBUCIÓN GENEREAL', transform=ax.transAxes, size=17, weight='bold')

sns.countplot(x=premier[premier.Goals > 0].Position, ax=axs[1], order=['Defender', 'Midfielder', 'Forward'])
axs[1].text(.53, .85, 'MÁS DE 0 GOLES', transform=ax.transAxes, size=17, weight='bold')

sns.countplot(x=premier[premier.Assists == 0].Position, ax=axs[2], order=['Goalkeeper', 'Defender', 'Midfielder', 'Forward'])
axs[1].text(1.05, .85, '0 ASISTENCIAS', transform=ax.transAxes, size=17, weight='bold')

sns.countplot(x=premier[premier.Assists > 0].Position, ax=axs[3], order=['Goalkeeper', 'Defender', 'Midfielder', 'Forward'])
axs[1].text(1.58, .85, 'MÁS DE 0 ASISTENCIAS', transform=ax.transAxes, size=17, weight='bold')
```

```
Out[40]: Text(1.58, 0.85, 'MÁS DE 0 ASISTENCIAS')
```




No hay porteros con más de 0 goles. La mayor parte de jugadores con 0 asistencias son porteros. El número de porteros con asistencias es ínfimo.

Poner un título en cada subplot: https://stackoverflow.com/questions/50232363/how-to-set-title-and-ylims-for-subplots-in-seaborn (https://stackoverflow.com/questions/50232363/how-to-set-title-and-ylims-for-subplots-in-seaborn)

```
scores = []
confusion_matrx = []
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(20, 5))
i=0

for model_name, mp in params.items():
    mp['model'].fit(X_tsample,y_tsample)
    pred = mp['model'].predict(X_test)
    accuracy = metrics.accuracy_score(y_test, pred)
    auc = metrics.roc_auc_score(y_test, pred)
    scores.append({
        'model': model_name,
        'accuracy': accuracy,
        'auc': auc
    })

    sns.heatmap(metrics.confusion_matrix(y_test, pred), annot=True, fmt='.9g', ax=axs[i])
    axs[i].set_title(model_name)
    i+=1

df = pd.DataFrame(scores,columns=['model', 'accuracy', 'auc'])
```