Seaborn Practice

1.Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. 2.used for data visualization and based on Matplotlib. 3.Seaborn allows the creation of statistical graphics

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
a=sns.load_dataset('flights')
```

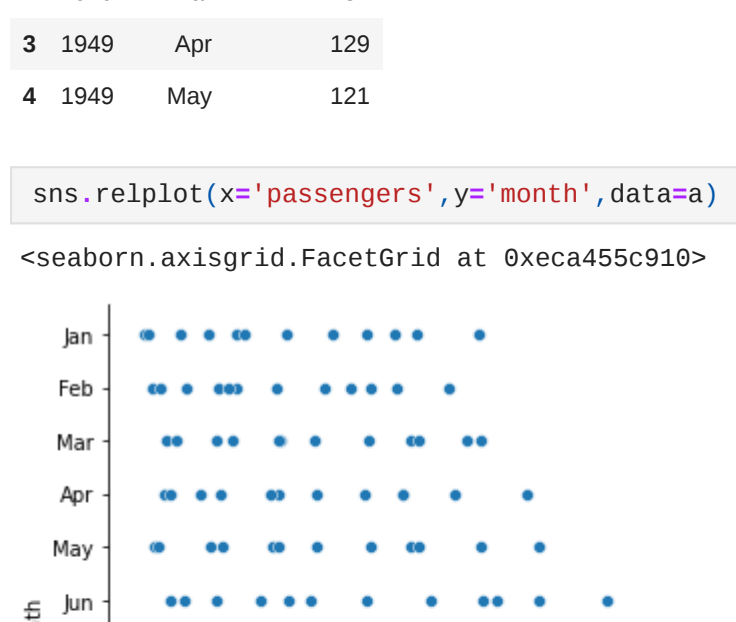
In [6]:
```
a.head()
```

Out[6]:

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

In [9]:
```
sns.relplot(x='passengers',y='month',data=a)
```

Out[9]: <seaborn.axisgrid.FacetGrid at 0xeca455c910>

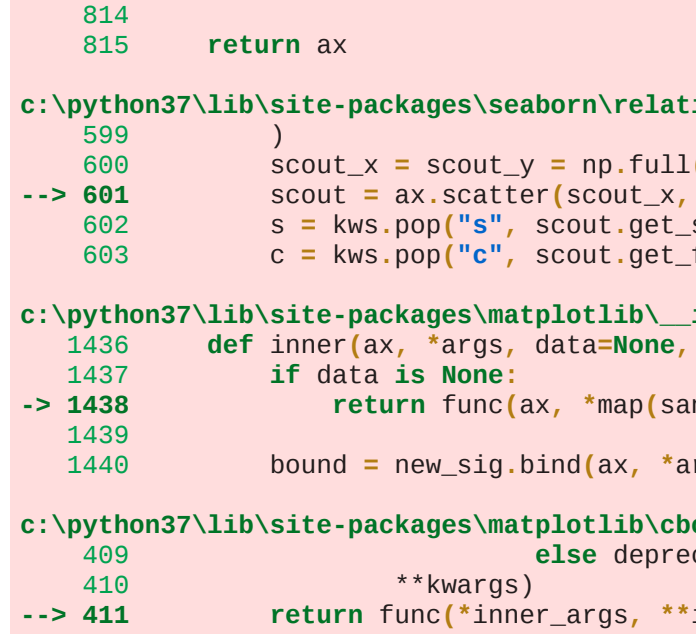


To add another dimension

In [10]:
```
a=sns.load_dataset('flights')
```

In [18]:
```
sns.relplot(x='passengers',y='month',z='year',data=a)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-18-b911b9499beb> in <module>
----> 1 sns.relplot(x='passengers',y='month',z='year',data=a)

c:\python37\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **kwargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46         return f(**kwargs)
     47     return inner_f
     48

c:\python37\lib\site-packages\seaborn\relational.py in relplot(x, y, hue, size, style, data, row, col, col_wrap, row_order, col_order, palette, hue_order, hue
_norm, sizes, size_order, size_norm, markers, dashes, style_order, legend, kind, height, aspect, facet_kws, units, **kwargs)
   1014
   1015     # Draw the plot
-> 1016     g.map_dataframe(func, **plot_kws)
   1017
   1018     # Label the axes

c:\python37\lib\site-packages\seaborn\axisgrid.py in map_dataframe(self, func, *args, **kwargs)
    748
    749             # Draw the plot
--> 750             self._facet_plot(func, ax, args, kwargs)
    751
    752         # Finalize the annotations and layout

c:\python37\lib\site-packages\seaborn\axisgrid.py in _facet_plot(self, func, ax, plot_args, plot_kwargs)
    772                 plot_kwargs[key] = val
    773             plot_args = []
--> 774         func(*plot_args, **plot_kwargs)
    775
    776         # Sort out the supporting information

c:\python37\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **kwargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46         return f(**kwargs)
     47     return inner_f
     48

c:\python37\lib\site-packages\seaborn\relational.py in scatterplot(x, y, hue, style, size, data, palette, hue_order, hue_norm, sizes, size_order, size_norm, m
arkers, style_order, x_bins, y_bins, units, estimator, ci, n_boot, alpha, x_jitter, y_jitter, legend, ax, **kwargs)
    811     p._attach(ax)
    812
--> 813     p.plot(ax, kwargs)
    814
    815     return ax

c:\python37\lib\site-packages\seaborn\relational.py in plot(self, ax, kws)
    599         )
    600         scout_x = scout_y = np.full(scout_size, np.nan)
--> 601         scout = ax.scatter(scout_x, scout_y, **kws)
    602         s = kws.pop("s", scout.get_sizes())
    603         c = kws.pop("c", scout.get_facecolors())

c:\python37\lib\site-packages\matplotlib\__init__.py in inner(ax, data, *args, **kwargs)
   1436     def inner(ax, *args, data=None, **kwargs):
   1437         if data is None:
-> 1438             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1439
   1440         bound = new_sig.bind(ax, *args, **kwargs)

c:\python37\lib\site-packages\matplotlib\cbook\deprecation.py in wrapper(*inner_args, **inner_kwargs)
    409                          else deprecation_addendum,
    410                 **kwargs)
--> 411         return func(*inner_args, **inner_kwargs)
    412
    413     return wrapper

c:\python37\lib\site-packages\matplotlib\axes\_axes.py in scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, verts, edgecolors, plot
nonfinite, **kwargs)
   4496         )
   4497         collection.set_transform(mtransforms.IdentityTransform())
-> 4498         collection.update(kwargs)
   4499
   4500         if colors is None:

c:\python37\lib\site-packages\matplotlib\artist.py in update(self, props)
    994                     func = getattr(self, f"set_{k}", None)
    995                     if not callable(func):
--> 996                         raise AttributeError(f"{type(self).__name__!r} object "
    997                                              f"has no property {k!r}")
    998                     ret.append(func(v))

AttributeError: 'PathCollection' object has no property 'z'
```

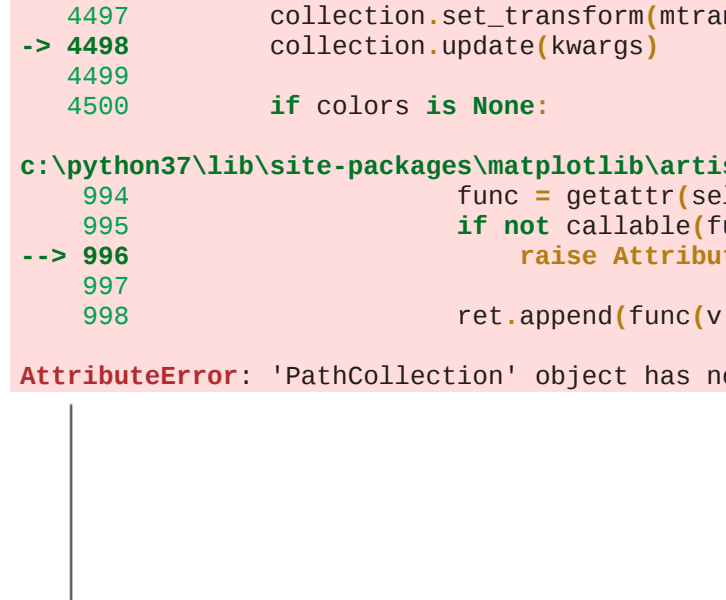


For line plot

In [7]:
```
b=sns.load_dataset('tips')
```

In [8]:
```
sns.relplot(x='time',y='tip',data=b,kind='line')
```

Out[8]: <seaborn.axisgrid.FacetGrid at 0xeca44ce700>

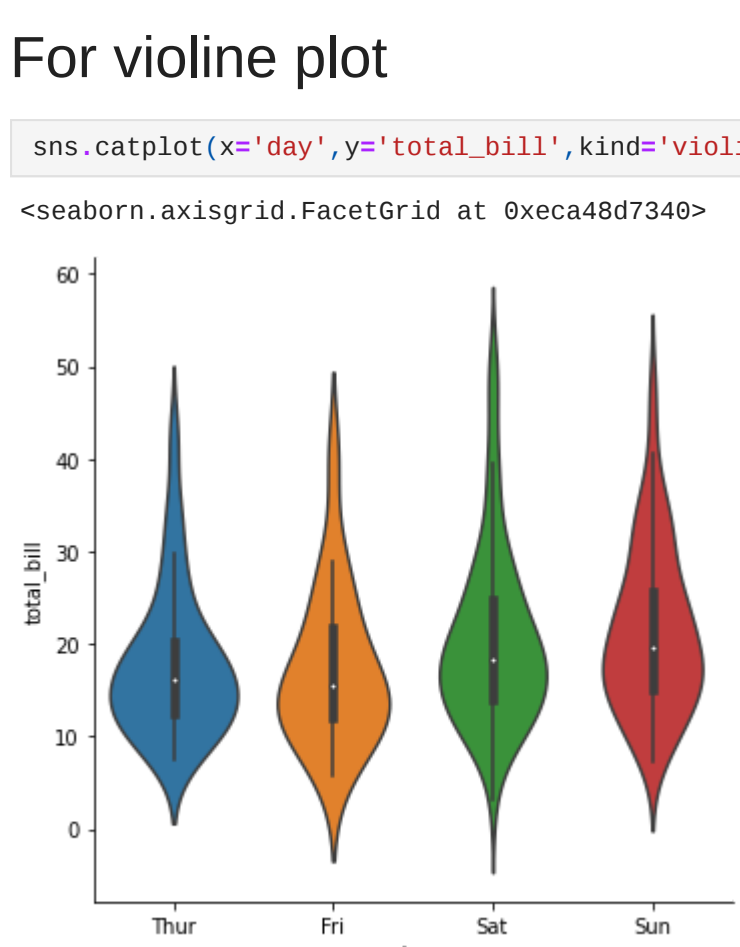


Plotting with categorical data

In [13]:
```
b=sns.load_dataset('tips')
```

In [15]:
```
sns.catplot(x='day',y='total_bill',data=b)
```

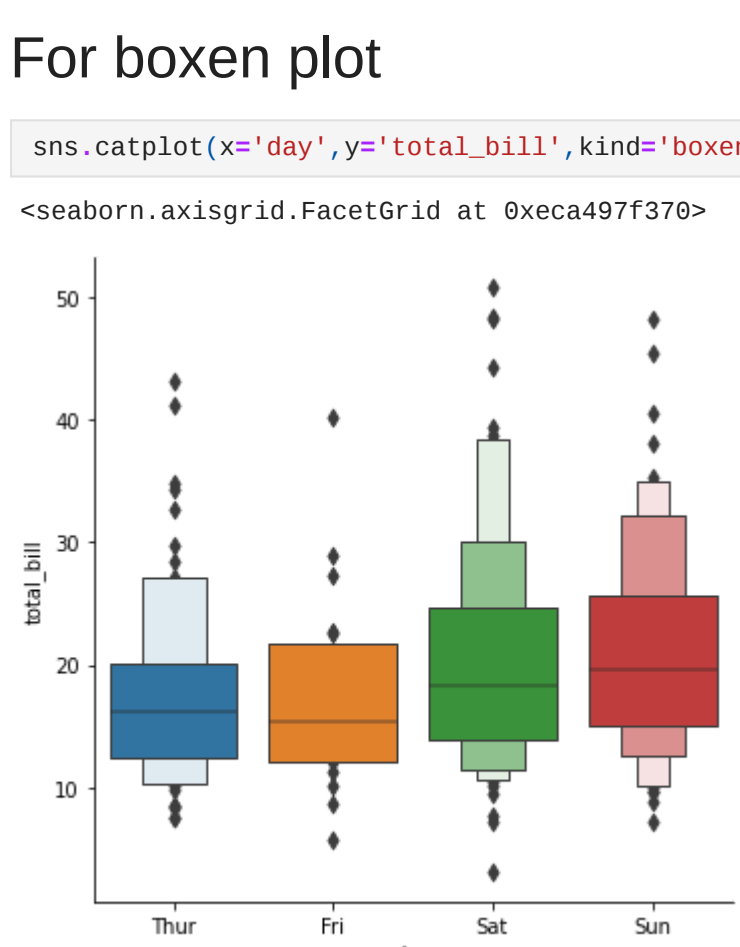
Out[15]: <seaborn.axisgrid.FacetGrid at 0xeca4842400>



For violine plot

In [17]:
```
sns.catplot(x='day',y='total_bill',kind='violin',data=b)
```

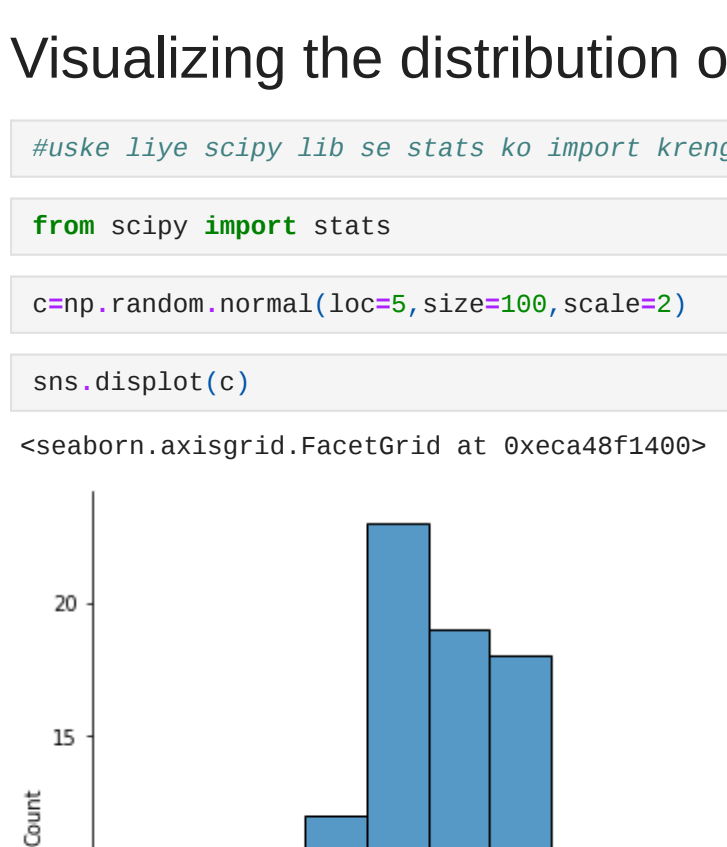
Out[17]: <seaborn.axisgrid.FacetGrid at 0xeca48d7340>



For boxen plot

In [19]:
```
sns.catplot(x='day',y='total_bill',kind='boxen',data=b)
```

Out[19]: <seaborn.axisgrid.FacetGrid at 0xeca497f370>



Visualizing the distribution of a dataset

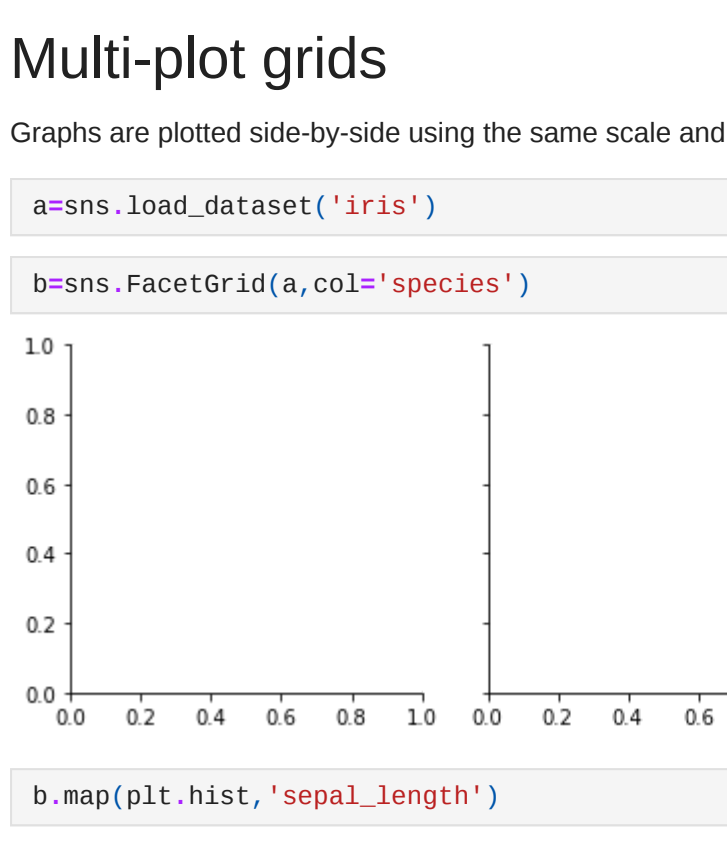
In [20]:
```
#uske liye scipy lib se stats ko import krenge
```

In [21]:
```
from scipy import stats
```

In [22]:
```
c=np.random.normal(loc=5,size=100,scale=2)
```

In [23]:
```
sns.displot(c)
```

Out[23]: <seaborn.axisgrid.FacetGrid at 0xeca48f1400>

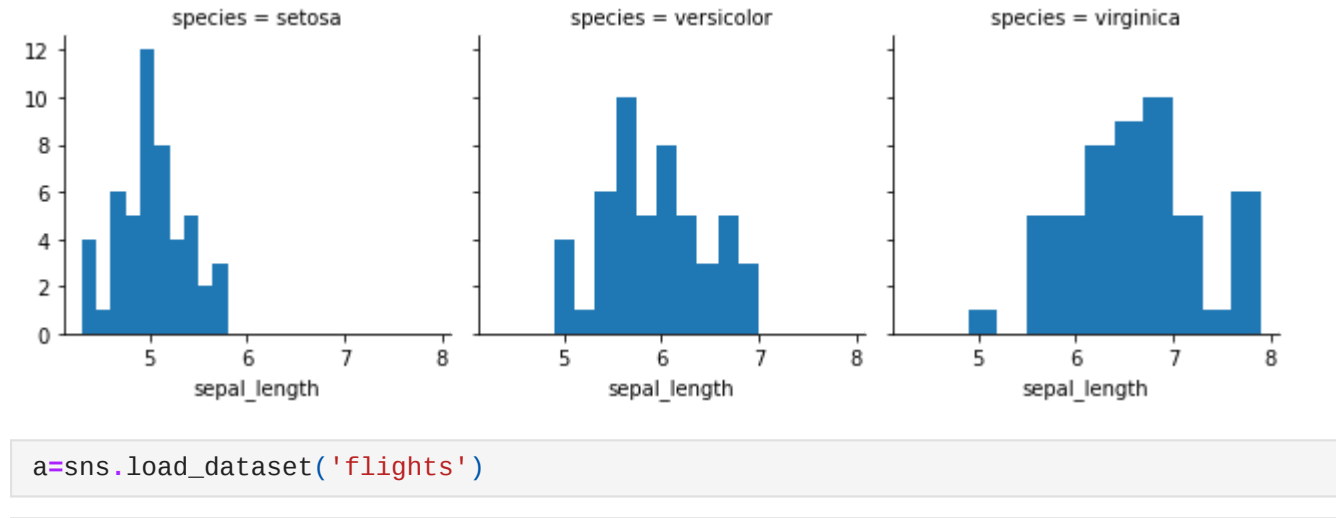


Multi-plot grids

Graphs are plotted side-by-side using the same scale and axes to aid comparison

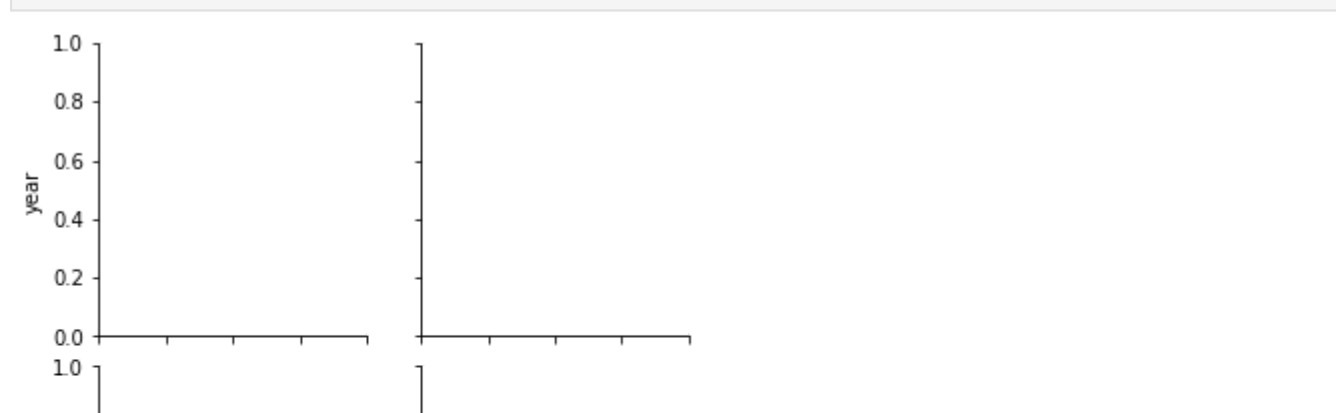
In [24]:
```
a=sns.load_dataset('iris')
```

In [30]:
```
b=sns.FacetGrid(a,col='species')
```



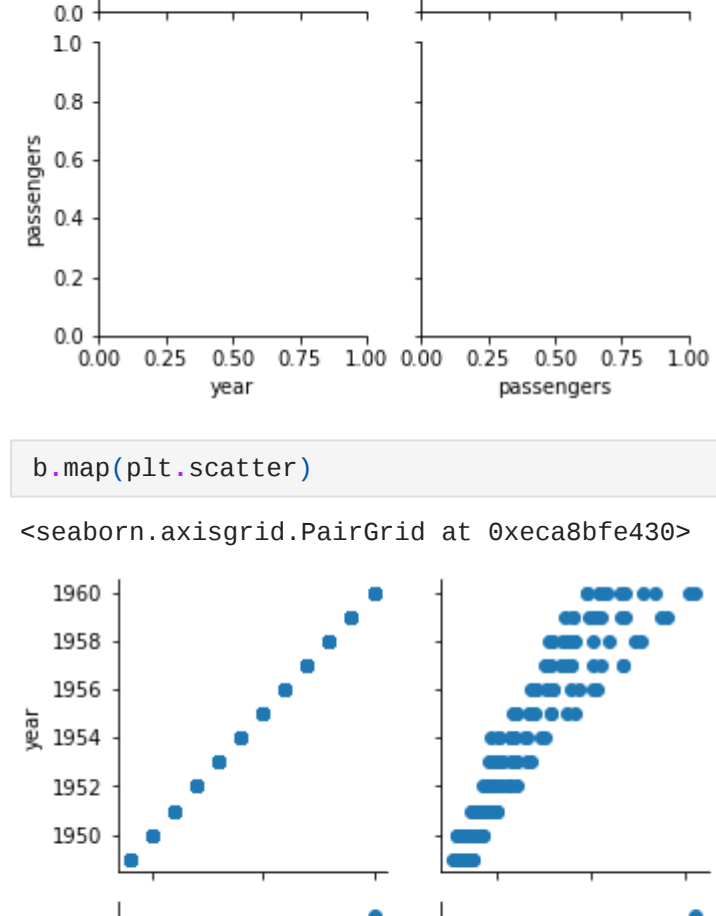
In [31]:
```
b.map(plt.hist,'sepal_length')
```

Out[31]: <seaborn.axisgrid.FacetGrid at 0xeca8d668b0>



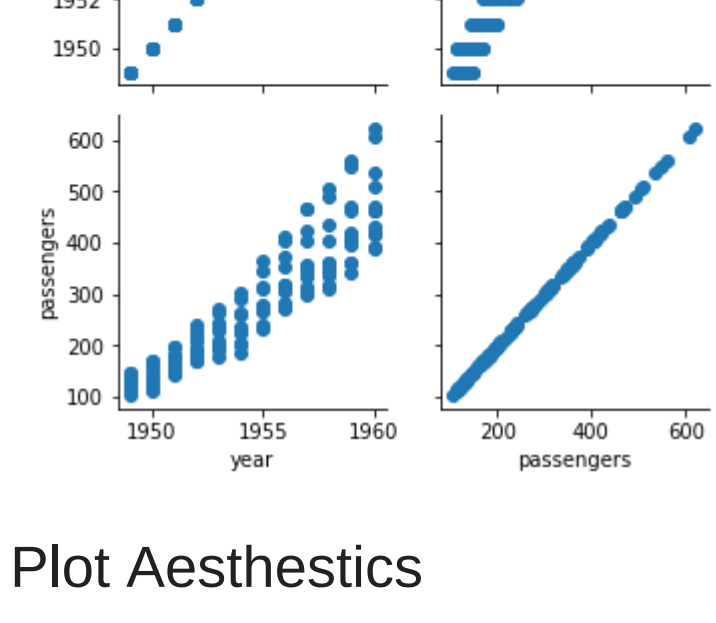
In [32]:
```
a=sns.load_dataset('flights')
```

In [33]:
```
b=sns.PairGrid(a)
```



In [34]:
```
b.map(plt.scatter)
```

Out[34]: <seaborn.axisgrid.PairGrid at 0xeca8bfe430>



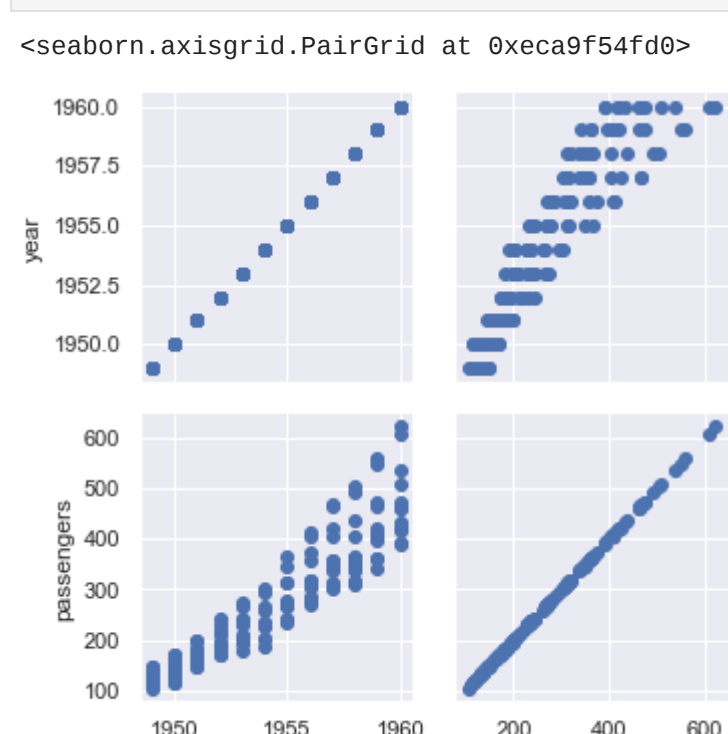
Plot Aesthestics

Deals with making our plots more attractive and delightful using various themes and color-pallettes

Lets try::

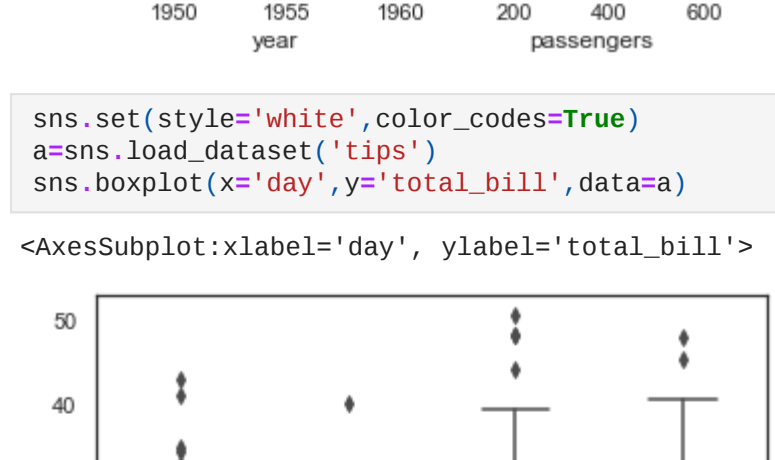
In [37]:
```
sns.set(style='darkgrid')
a=sns.load_dataset('flights')
b=sns.PairGrid(a)
b.map(plt.scatter)
```

Out[37]: <seaborn.axisgrid.PairGrid at 0xeca9f54fd0>



In [38]:
```
sns.set(style='white',color_codes=True)
a=sns.load_dataset('tips')
sns.boxplot(x='day',y='total_bill',data=a)
```

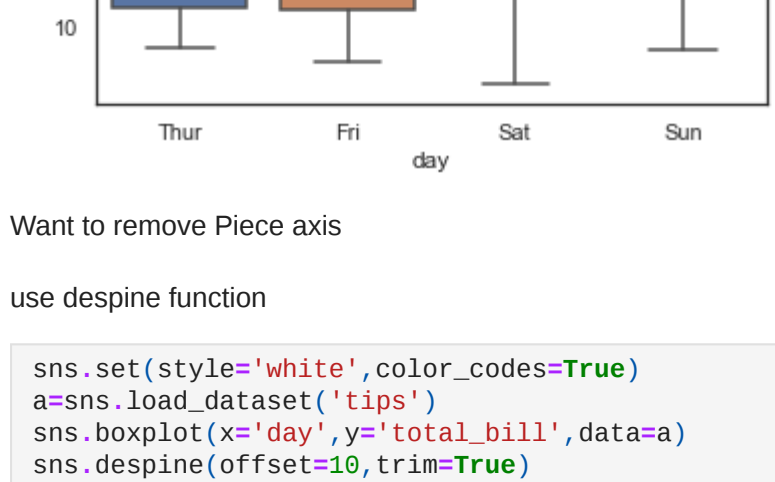
Out[38]: <AxesSubplot:xlabel='day', ylabel='total_bill'>



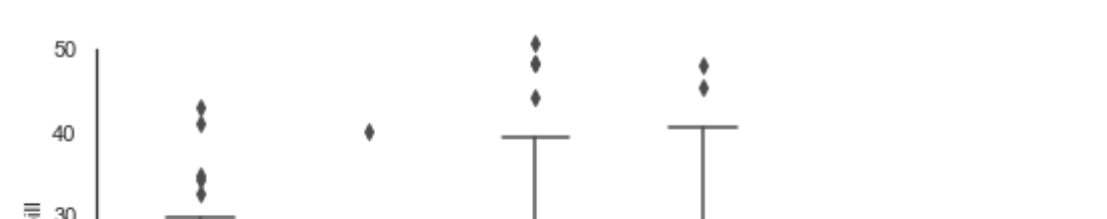
Want to remove Piece axis

use despine function

In [41]:
```
sns.set(style='white',color_codes=True)
a=sns.load_dataset('tips')
sns.boxplot(x='day',y='total_bill',data=a)
sns.despine(offset=10,trim=True)
```



In [43]:
```
c=sns.color_palette()
sns.palplot(c)
```



so these are the color palette currently available on seaborn