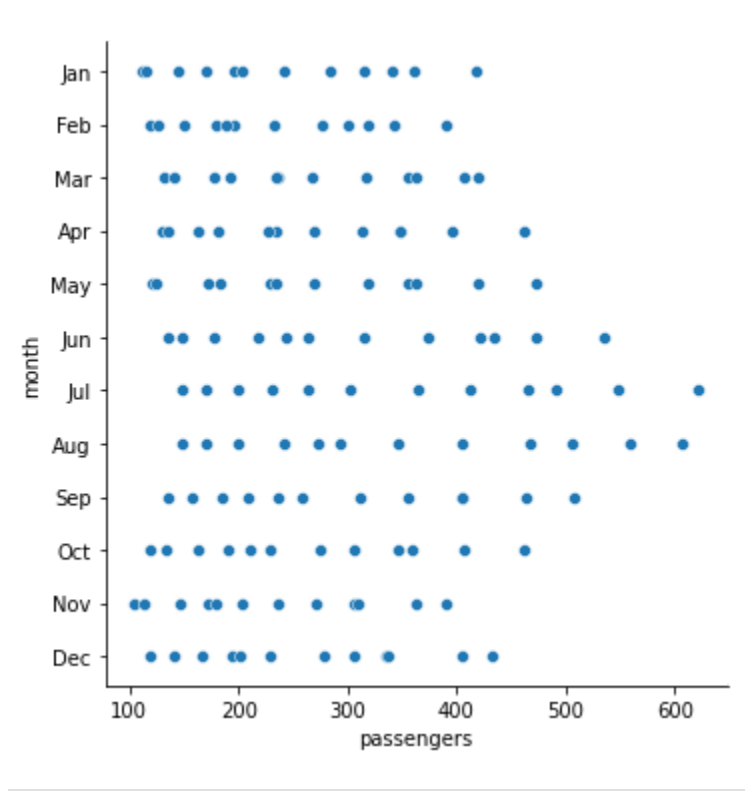
a. Visualizing statistical relationship:

statistical analysis *The process of understanding relationships between variables of a dataset. and how this relationship intern depend upon other variable is known as* **statistical analysis**
relplot: is figure level function that makes you the two other axes functions for visualizing statistical relationships.
These axes level functions are scatter plot and line plot which can be specified using the kind parameter of the relplot

In [1]:
```
# relplot function to plot various statistical relationship
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

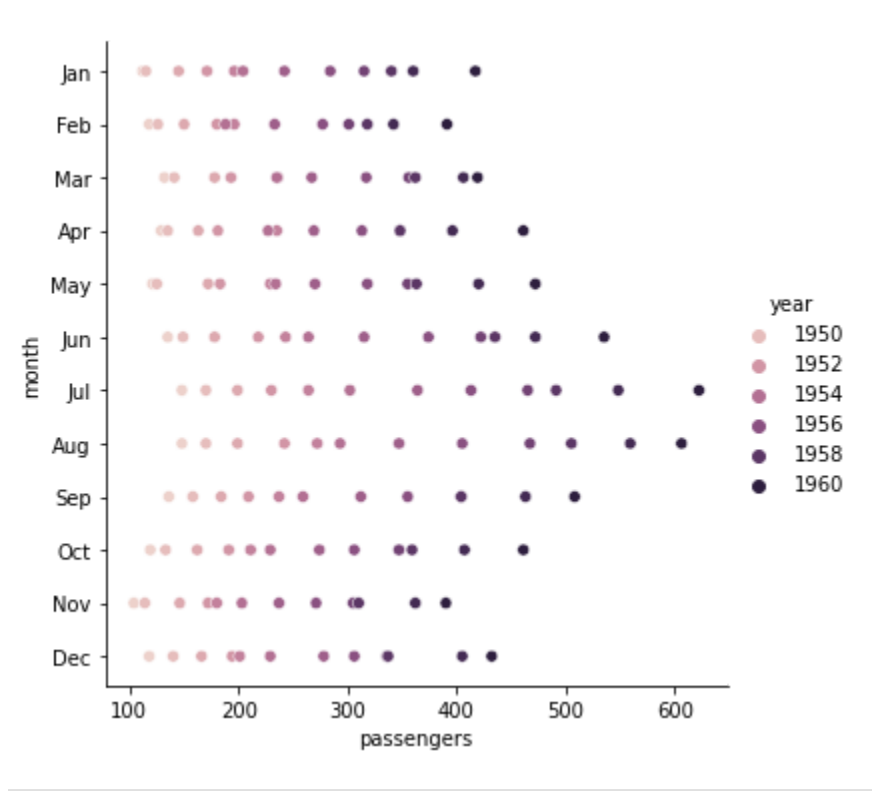
In [2]:
```
#scatter plot
a=sns.load_dataset("flights") #flights data is direct come from github(mwaskom/seaborn-data)
sns.relplot(x="passengers", y="month", data = a) #there present x-axis or y-axis that means it has 2d
```

Out[2]:
```
<seaborn.axisgrid.FacetGrid at 0x2312c4f4d90>
```



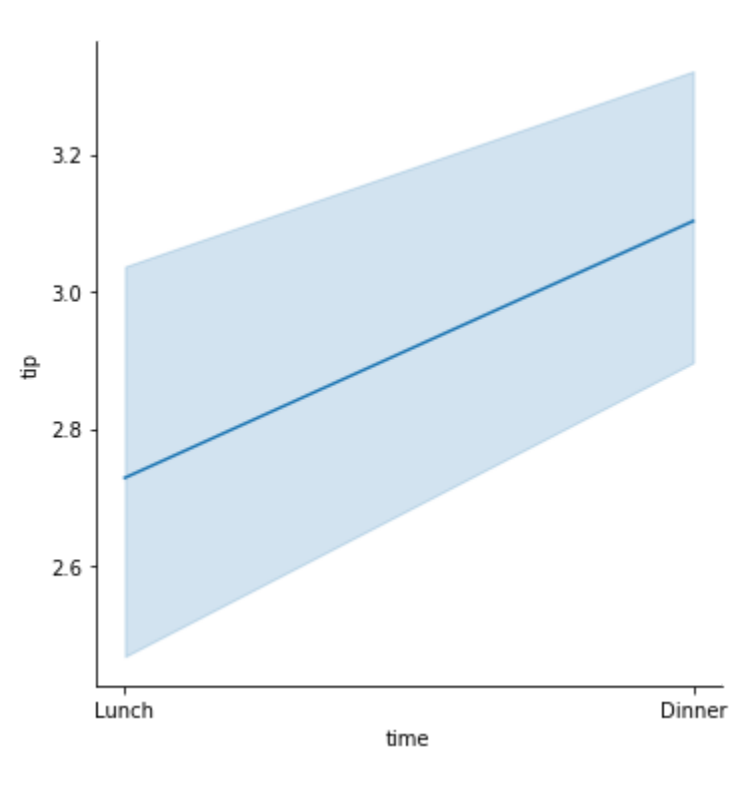
In [3]:
```
#now we are adding another dimensions
a=sns.load_dataset("flights") #flights data is direct come from github(mwaskom/seaborn-data)
sns.relplot(x="passengers", y="month", hue="year", data = a) #it is 3-dimensional
```

Out[3]:
```
<seaborn.axisgrid.FacetGrid at 0x2314c36b280>
```



In [4]:
```
#line plot
b=sns.load_dataset("tips") #this tips data also import from github
sns.relplot(x="time", y="tip", data=b, kind="line")
```

Out[4]:
```
<seaborn.axisgrid.FacetGrid at 0x2314f557550>
```

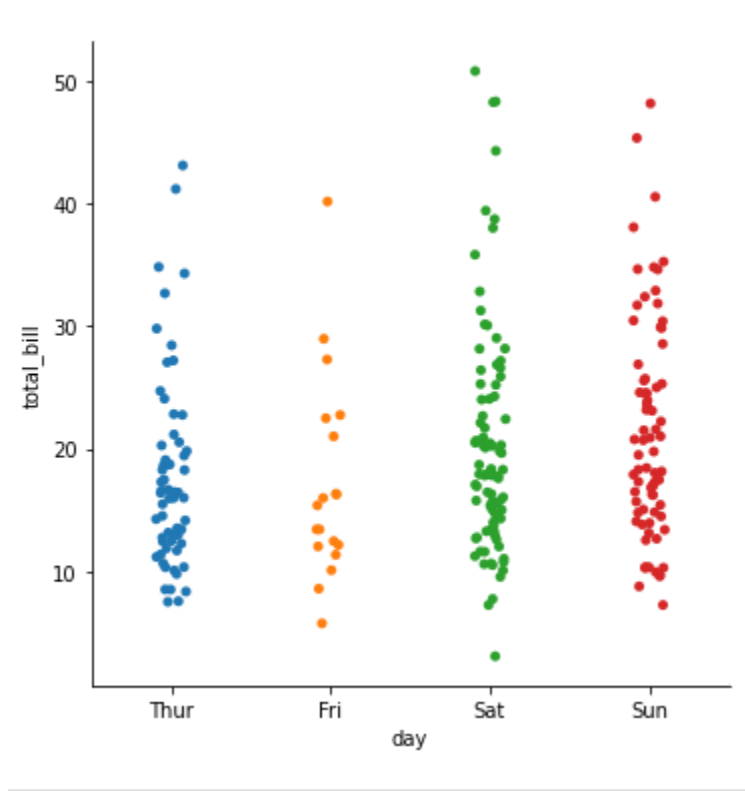


b. Ploting with categorical data:

The approach basically comes in a picture when a main variable is divided into discerete groups.
catplot *like relplot, catplot is also a figure level function caracterised by three families of axes level functions,which are scatter plots,distribution plots or estimate plots.*

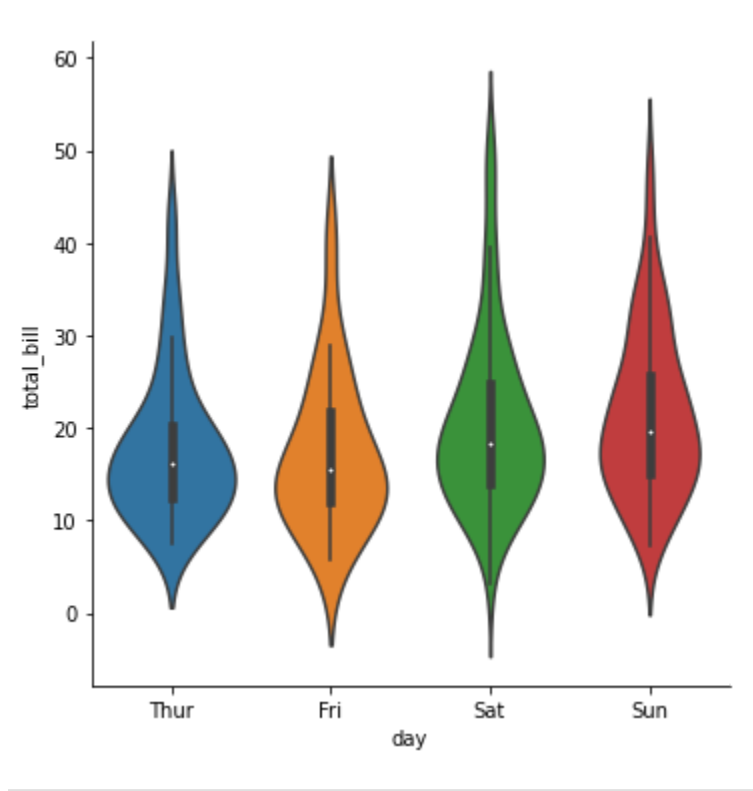
In [5]:
```
# To achive this we will be using the catplot function available in seaborn
sns.catplot(x="day", y="total_bill", data=b) #basically this is our scatter plot
```

Out[5]:
```
<seaborn.axisgrid.FacetGrid at 0x2314f5fea00>
```



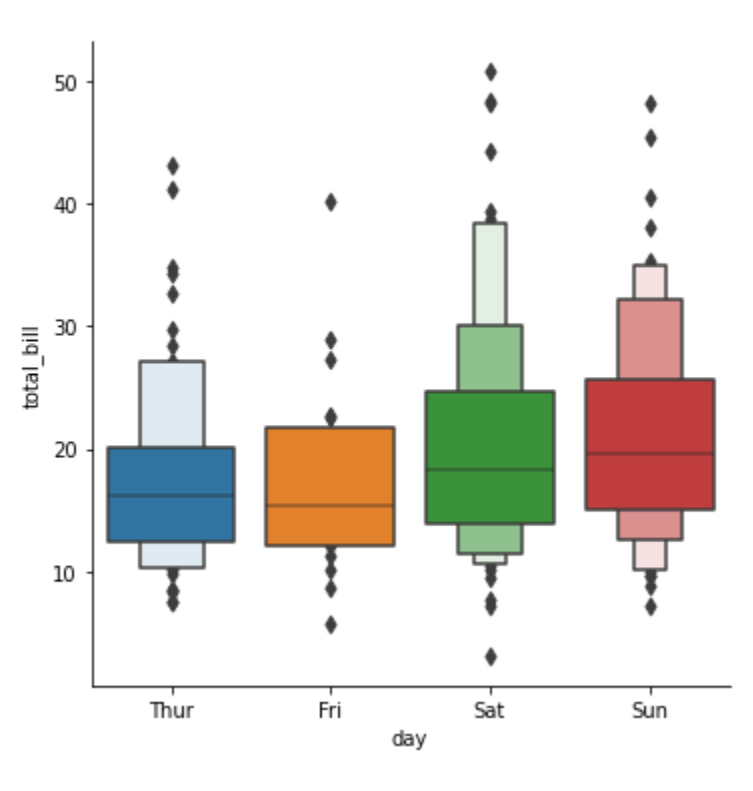
In [6]:
```
# violin plot
sns.catplot(x="day", y="total_bill",kind="violin", data=b) # this is our violin plot
```

Out[6]:
```
<seaborn.axisgrid.FacetGrid at 0x2314f5f8c10>
```



In [7]:
```
# boxen plot
sns.catplot(x="day", y="total_bill",kind="boxen", data=b)
```

Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x2314f670460>
```



c. Visualizing the distribution of a dataset:

Visualizing dataset basically deals with understanding datasets with context to being ***Univariate or Divariate*** .

In [8]:
```
from scipy import stats
```

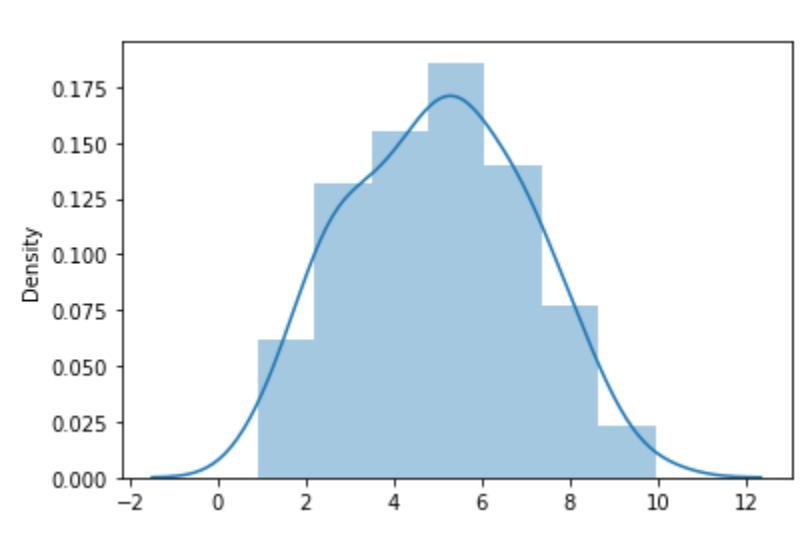
In [9]:
```
c=np.random.normal(loc=5,size=100,scale=2)
```

Univariate distribution

In [12]:
```
# Using distplot function for univariate distribution
sns.distplot(c) #this is an exaple of univariate distribution
```

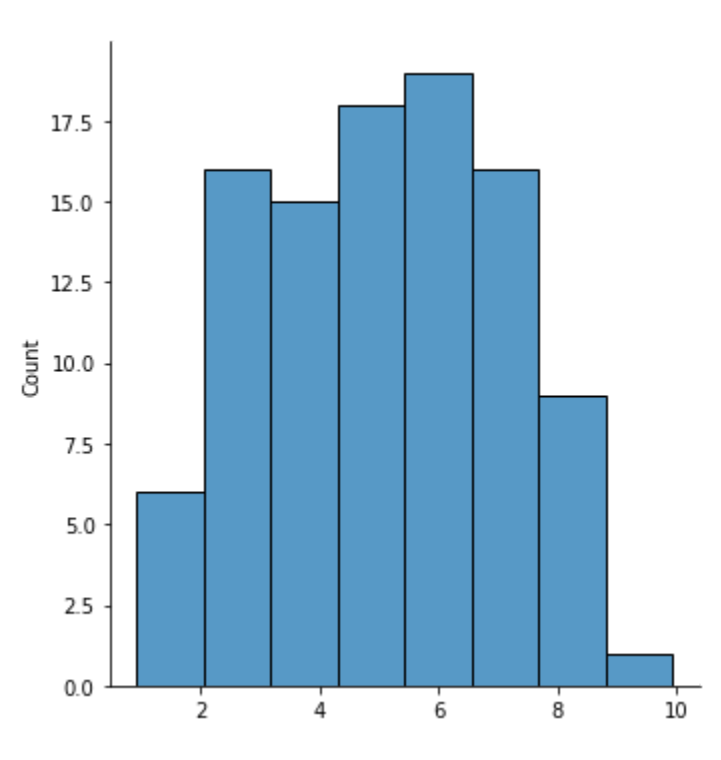
```
c:\python39\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[12]:
```
<AxesSubplot:ylabel='Density'>
```



In [13]:
```
# `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
sns.displot(c)
```

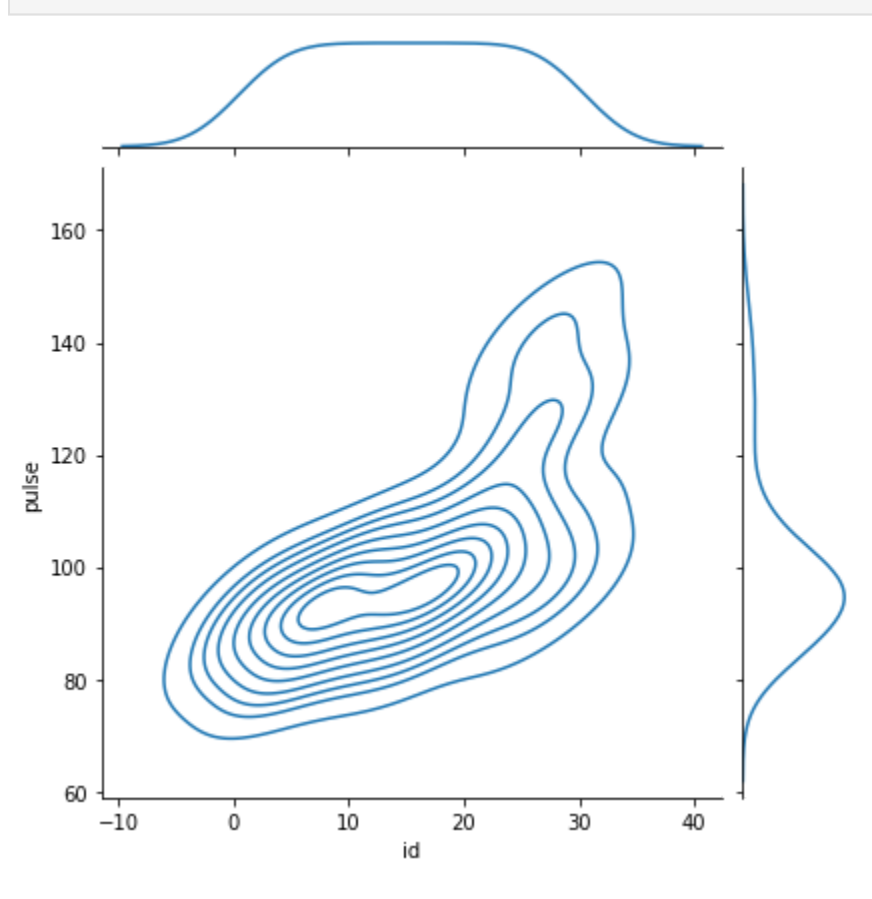
Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x231518bb580>
```



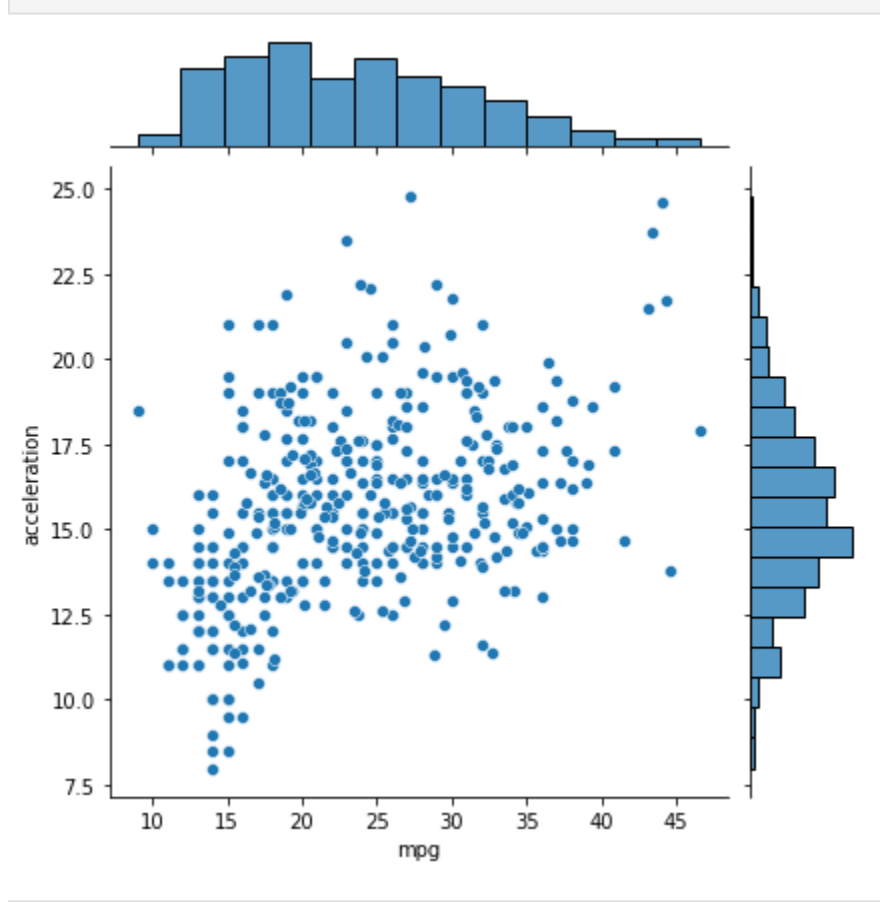
Bivariate distribution

In [18]:
```
# Using jointplot function to plot bivariate distribution
# x=pd.DataFrame({'Day':[1,2,3,4,5,6,7],
#                 'Grocery':[30,80,45,23,51,46,76],
#                 'Clothes':[13,40,34,23,54,67,98],
#                 'Utensils':[12,32,27,56,71,65,80]})
# y=pd.DataFrame({'Day':[8,9,10,11,12,13,14],
#                 'Grocery':[30,80,45,23,51,46,76],
#                 'Clothes':[13,40,34,23,54,67,98],
#                 'Utensils':[12,32,27,56,71,65,80]})
# mean, cov=[0,1],[(1,.5),(.5,1)]
# data = np.random.multivariate_normal(mean, cov, 200)
# with sns.axes_style("white"):
#     sns.jointplot(x=x,y=y, kind="scatter", color="b")
```

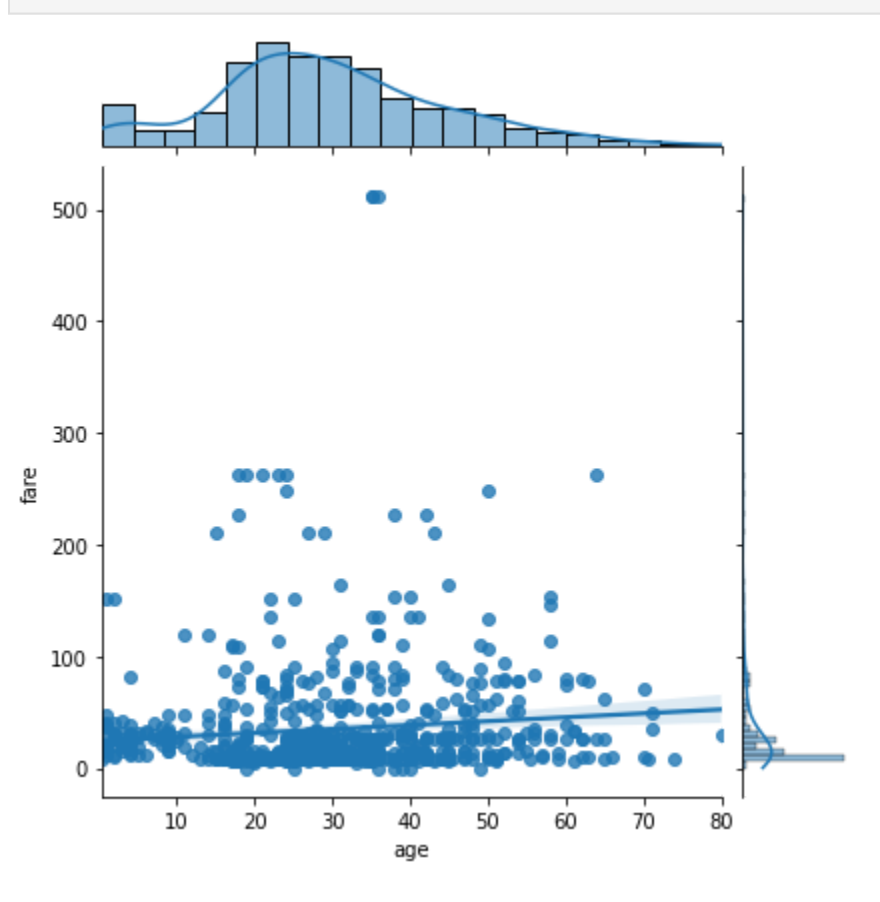
In [22]:
```
# importing required packages
import seaborn as sns
import matplotlib.pyplot as plt

# loading dataset
data = sns.load_dataset("exercise")

# draw jointplot with
# kde kind
sns.jointplot(x = "id", y = "pulse",
              kind = "kde", data = data)
# Show the plot
plt.show()
```



In [19]:
```
# importing required packages
import seaborn as sns
import matplotlib.pyplot as plt

# loading dataset
data = sns.load_dataset("mpg")

# draw jointplot with
# scatter kind
sns.jointplot(x = "mpg", y = "acceleration",
              kind = "scatter", data = data)
# show the plot
plt.show()
```



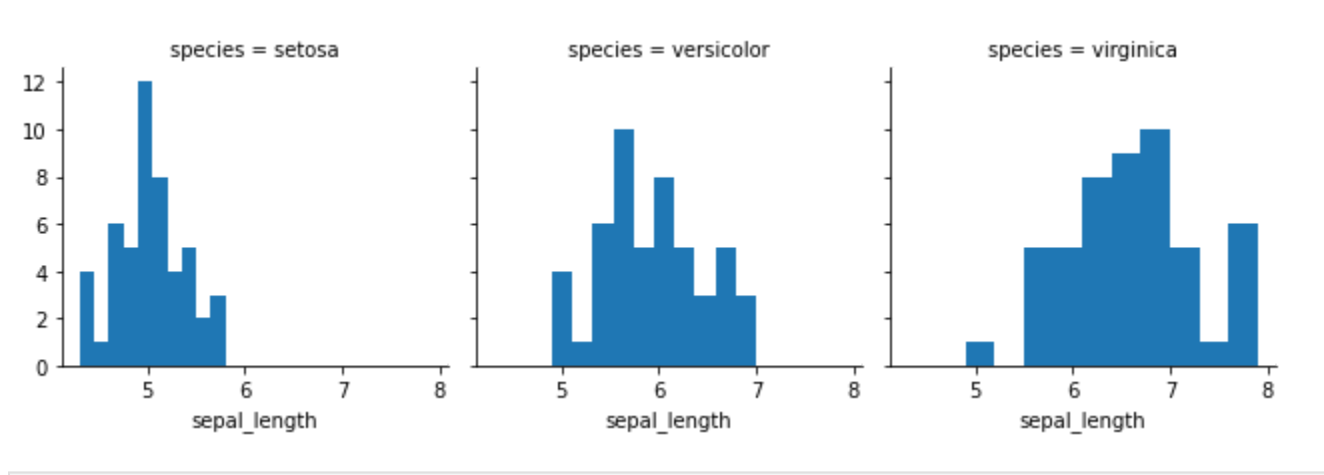
In [24]:
```
# importing required packages
import seaborn as sns
import matplotlib.pyplot as plt

# loading dataset
data = sns.load_dataset("titanic")

# draw jointplot with
# reg kind
sns.jointplot(x = "age", y = "fare",
              kind = "reg", data = data,
              dropna = True)

# show the plot
plt.show()
```



Multi-plot Grid:

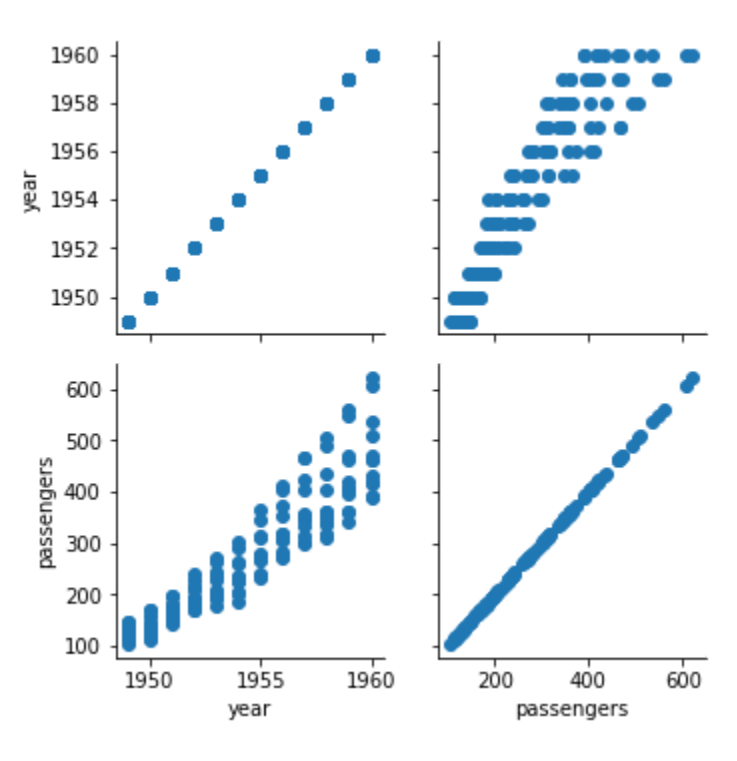
In [26]:
```
a=sns.load_dataset("iris")
b=sns.FacetGrid(a, col="species")
b.map(plt.hist, "sepal_length")
```

Out[26]:
```
<seaborn.axisgrid.FacetGrid at 0x2315335edf0>
```



In [27]:
```
a=sns.load_dataset("flights")
b=sns.PairGrid(a)
b.map(plt.scatter)
```

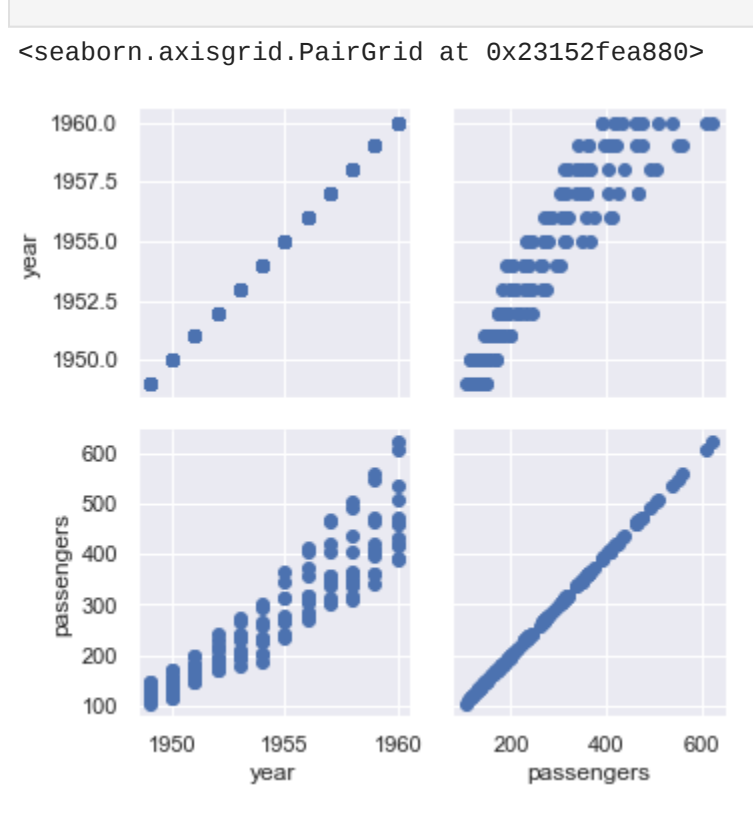
Out[27]:
```
<seaborn.axisgrid.PairGrid at 0x23153376220>
```



Plot-Aesthetics:

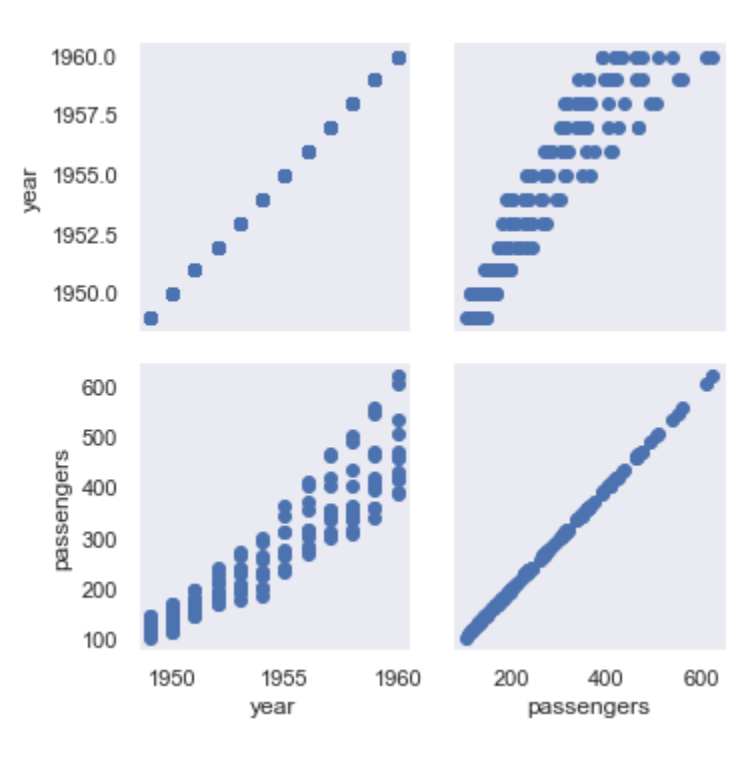
In [28]:
```
sns.set(style="darkgrid")
a=sns.load_dataset("flights")
b=sns.PairGrid(a)
b.map(plt.scatter)
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x23152fea880>
```



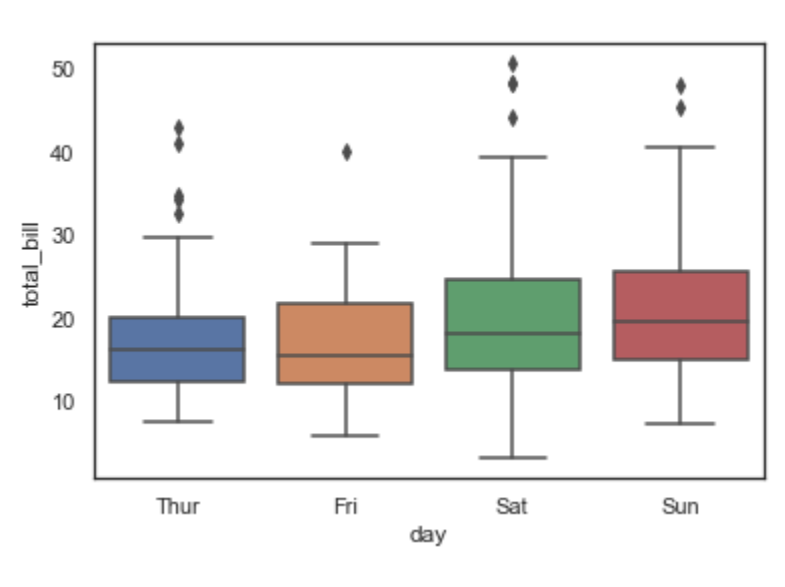
In [29]:
```
sns.set(style="dark")
a=sns.load_dataset("flights")
b=sns.PairGrid(a)
b.map(plt.scatter)
```

Out[29]:
```
<seaborn.axisgrid.PairGrid at 0x231530d76d0>
```

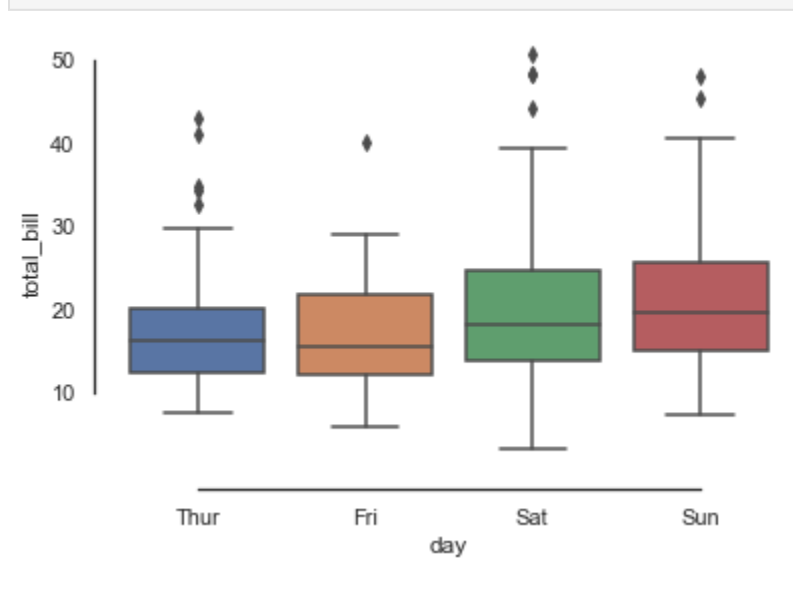


In [30]:
```
sns.set(style="white", color_codes=True)
a=sns.load_dataset('tips')
sns.boxplot(x="day",y="total_bill", data=a)
```

Out[30]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



In [31]:
```
# despine function
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 [32]:
```
# color available in seaborn
c=sns.color_palette()
sns.palplot(c)
```