

# Data Visualization

## Libraries:

### ***Scientific computing:***

Pandas = data structure and tools, 2D dataframes

NumPy = arrays and matrices

SciPy = optimization and solving differential equations

### ***Data Visualization:***

Matplotlib = plots, graphs and figures

Seaborn = heat maps, times series and other plots

### ***ML Algorithmic Development:***

Scikit-lesrn = ML, regression, classification, clustering analysis

Statsmodels = Explore data, estimation of statistical models, statistical analysis

In [5]:

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (0.11.2)
```

```
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
```

```
You should consider upgrading via the 'C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

```
Requirement already satisfied: matplotlib>=2.2 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (3.5.1)
```

```
Requirement already satisfied: pandas>=0.23 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (1.3.4)
```

```
Requirement already satisfied: numpy>=1.15 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (1.21.4)
```

```
Requirement already satisfied: scipy>=1.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from seaborn) (1.7.3)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (4.28.5)
Requirement already satisfied: packaging>=20.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (21.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (9.0.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (3.0.6)
Requirement already satisfied: cycler>=0.10 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from pandas>=0.23->seaborn) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib>=2.2->seaborn) (1.16.0)
```

In [5]:

```
import seaborn as sns
```

Matplotlib is building the font cache; this may take a moment.

In [6]:

```
!pip install matplotlib
```

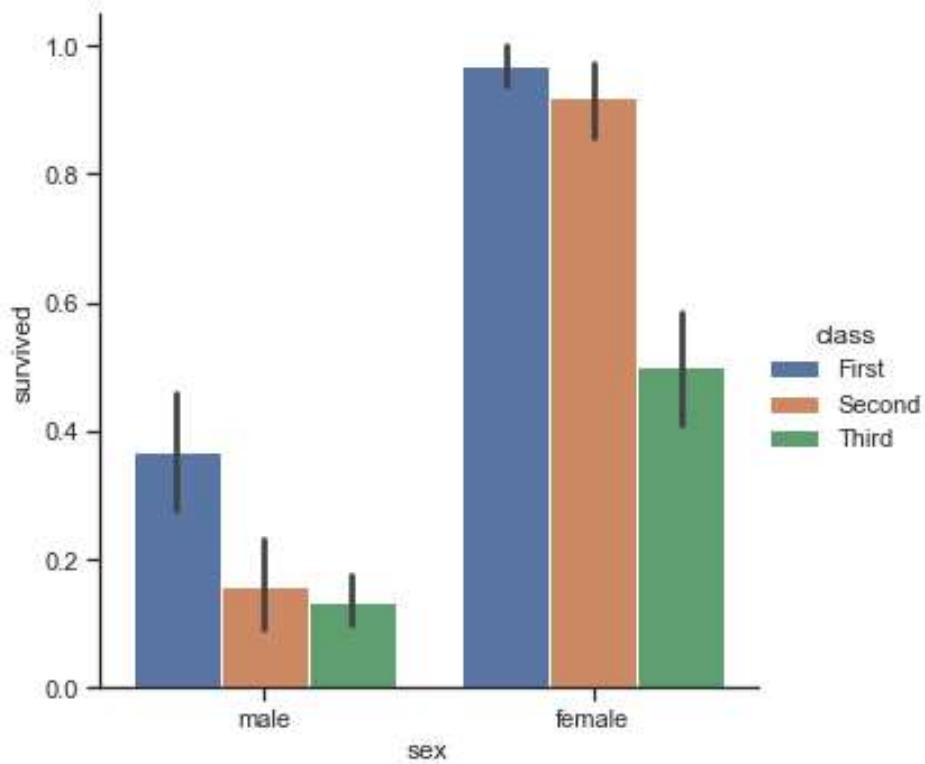
```
Requirement already satisfied: matplotlib in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (3.5.1)
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
Requirement already satisfied: python-dateutil>=2.7 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (9.0.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (4.28.5)
Requirement already satisfied: cycler>=0.10 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.17 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (1.21.4)
```

```
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (3.0.6)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: six>=1.5 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

```
In [7]: import matplotlib.pyplot as plt
```

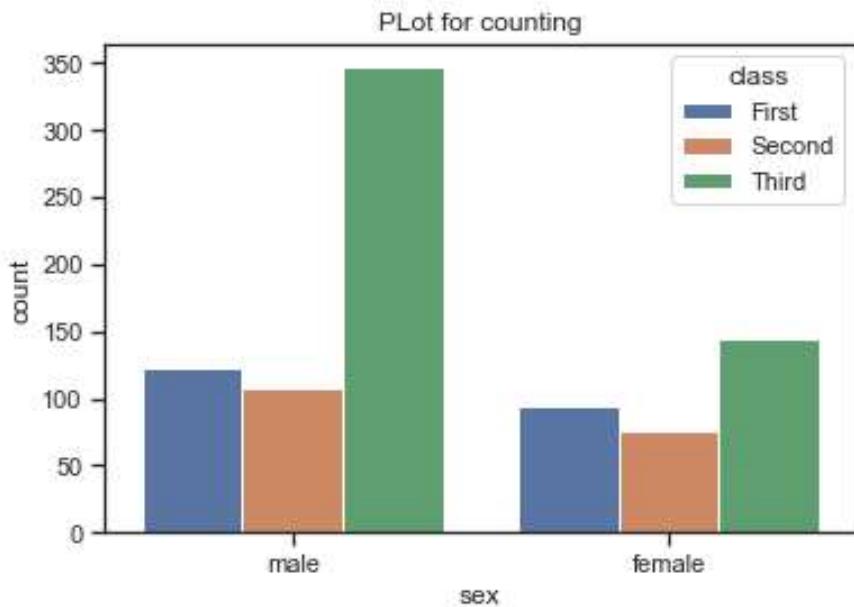
```
In [8]: sns.set_theme(style='ticks', color_codes=True)
titanic= sns.load_dataset('titanic')
sns.catplot(x='sex',y='survived',hue='class',kind='bar',data=titanic)
```

```
Out[8]: <seaborn.axisgrid.FacetGrid at 0x25e2cac8580>
```



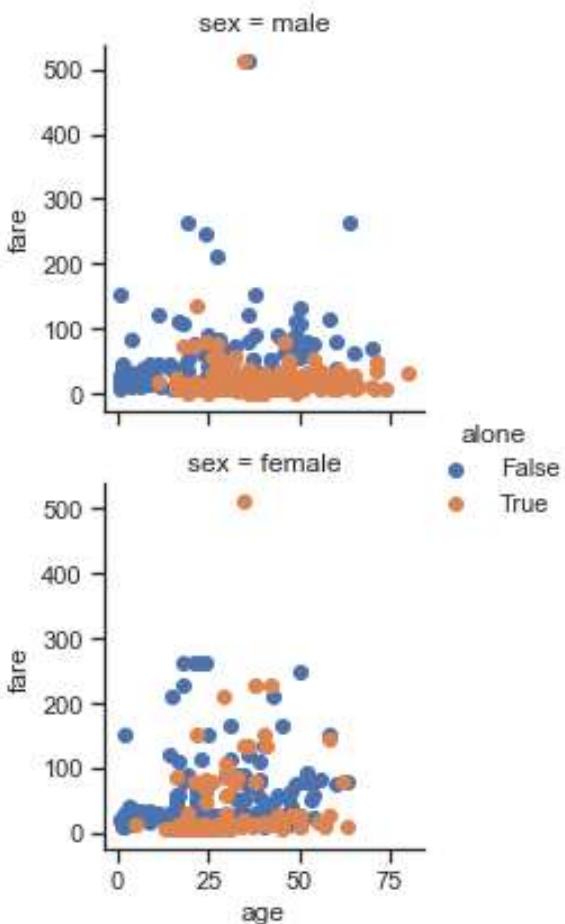
```
In [9]: p1 = sns.countplot(x='sex',data=titanic,hue='class')
p1.set_title('PLOT for counting')
```

```
Out[9]: Text(0.5, 1.0, 'PLOT for counting')
```



In [12]:

```
g = sns.FacetGrid(titanic, row='sex', hue='alone')
g = (g.map(plt.scatter, 'age', 'fare').add_legend())
```



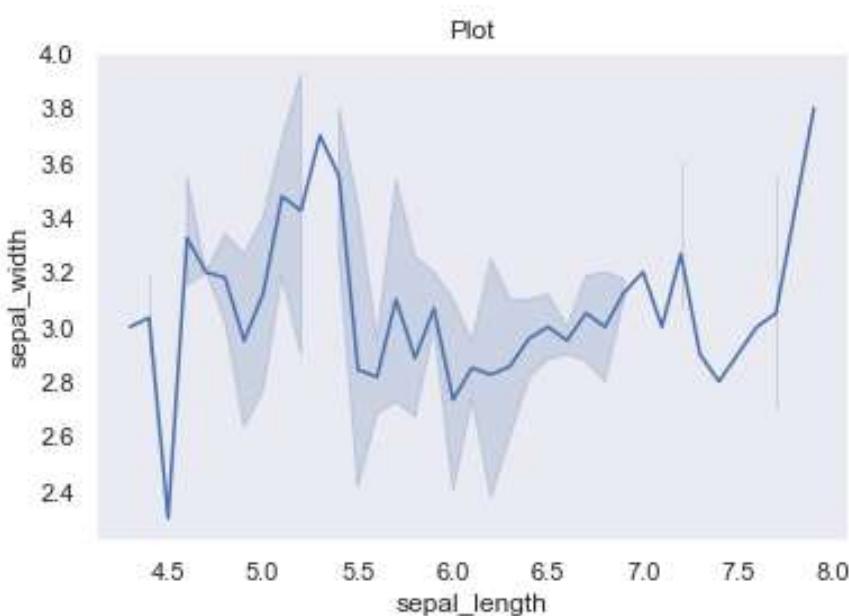
## Line plot

In [16]:

```
import seaborn as sns      #import libraries
import matplotlib.pyplot as plt
sns.set_theme(style='dark')    #theme-style of graph(#styles=darkgrid, whi
```

```
ds= sns.load_dataset('iris')    #Load dataset
ds
p = sns.lineplot(x = 'sepal_length', y = 'sepal_width', data = ds) #aesthetic
p.set_title('Plot') #title
#plt.xlim(2) #limits(x,y)
#plt.ylim(1)
plt.figure(figsize=(10,8))      #fig_size
```

Out[16]: <Figure size 720x576 with 0 Axes>



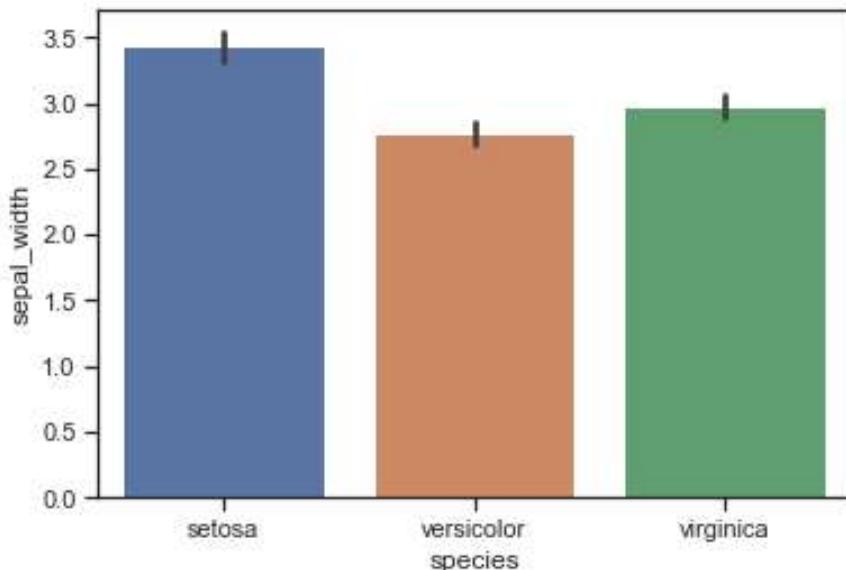
<Figure size 720x576 with 0 Axes>

## Bar Plot

In [21]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style = 'ticks', color_codes=True)

d = sns.load_dataset('iris')
d
p = sns.barplot(x = 'species', y = 'sepal_width', data=d)
plt.show()
```

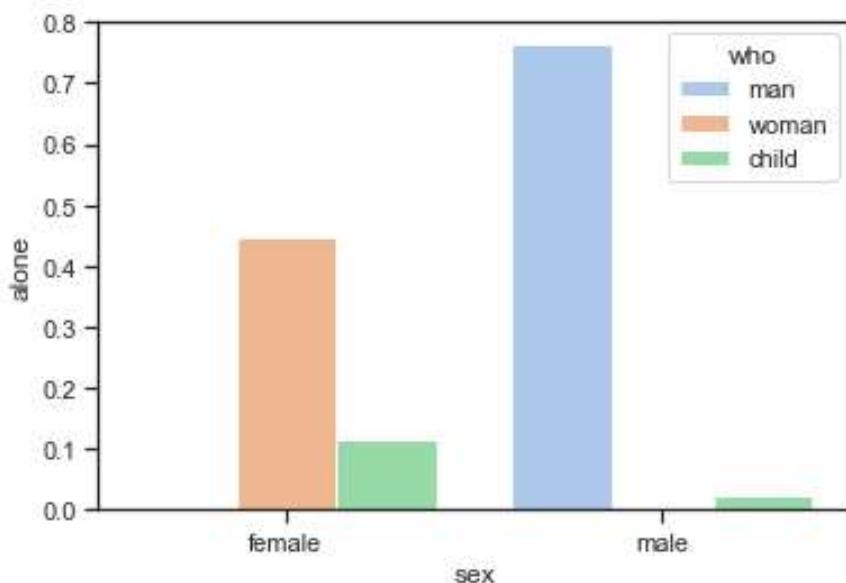


In [28]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style='ticks', color_codes=True)

d = sns.load_dataset('titanic')
d
p = sns.barplot(x = 'sex', y ='alone', hue ='who' , data =d,
                 order=['female','male'], color = 'red', ci= None ,
                 palette = 'pastel')

# order= order of variables on axis
# ci = confidence interval(error bars)
# palette = colour scheme
```



In [33]:

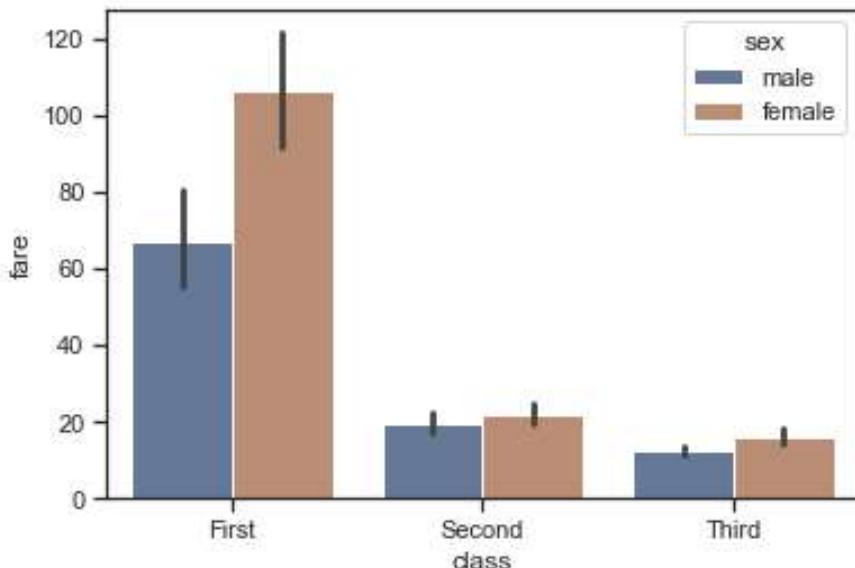
```
import seaborn as sns
import matplotlib.pyplot as plt
from numpy import mean
sns.set_theme(style='ticks', color_codes=True)

d = sns.load_dataset('titanic')
d
```

```

p = sns.barplot(x = 'class', y ='fare', hue ='sex' , data =d, estimator=m
#saturation = color saturation of graph
#estimator = estimate of a given quantity

```



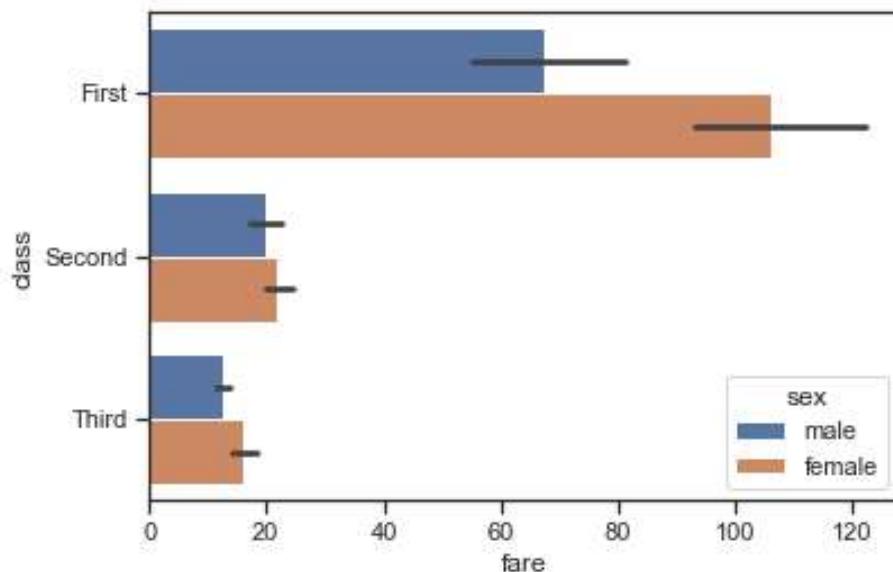
In [36]:

```

import seaborn as sns
import matplotlib.pyplot as plt

d = sns.load_dataset('titanic')
d
p = sns.barplot(x = 'fare', y = 'class',hue='sex', data= d)

```



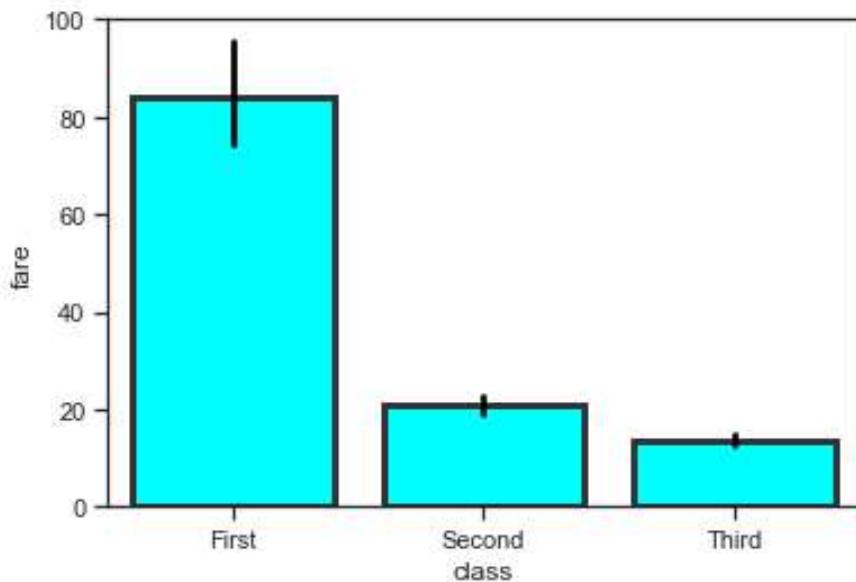
In [58]:

```

import seaborn as sns
import matplotlib.pyplot as plt
d = sns.load_dataset('titanic')
d
p = sns.barplot(x ='class', y = 'fare', data = d,
                 linewidth = 3, facecolor =(0,1,1,1),
                 errcolor ='0', edgecolor= '.2')
#errcolor = (0-black,1-white)

```

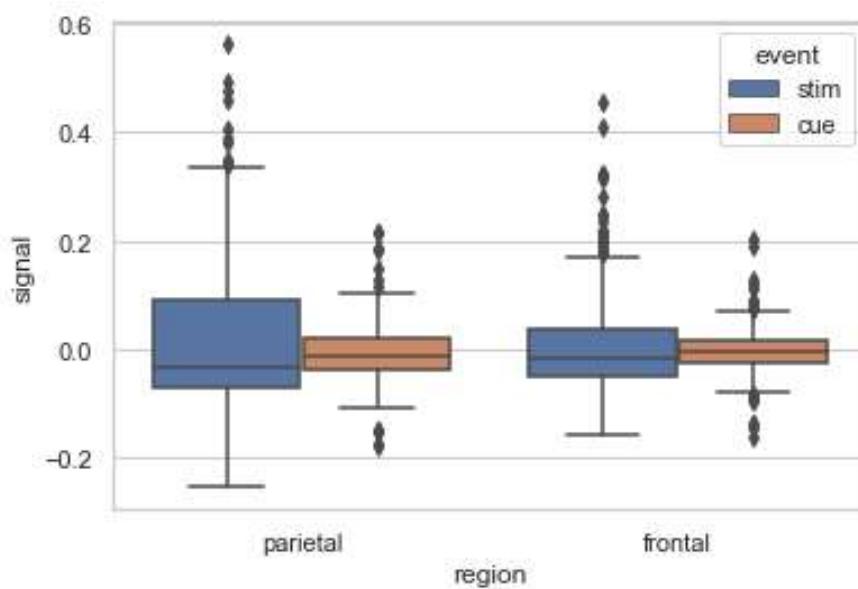
```
#edgecolor = border of bars  
#
```



## Box plot

In [60]:

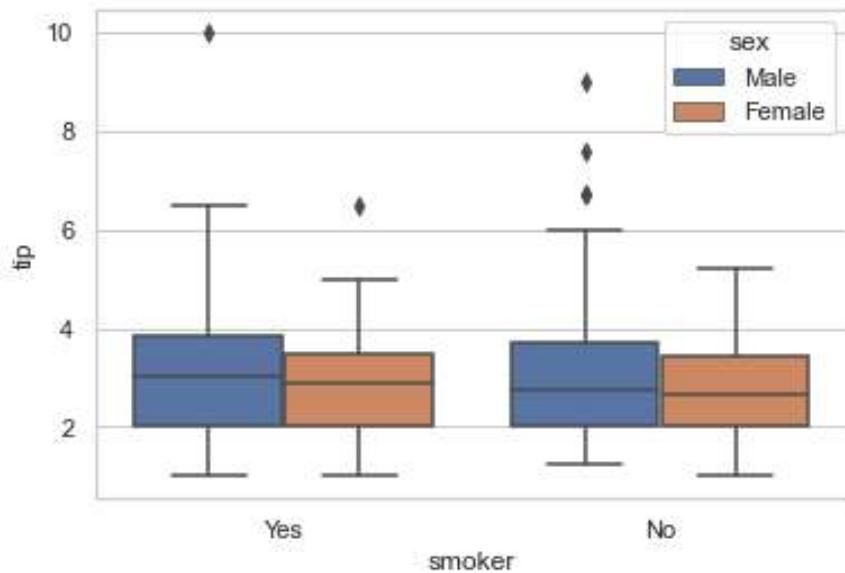
```
import seaborn as sns  
import matplotlib.pyplot as plt  
sns.set_theme(style = 'whitegrid')  
  
d = sns.load_dataset('fmri')  
d  
p = sns.boxplot(x='region', y ='signal', hue ='event', data = d)
```



In [62]:

```
import seaborn as sns  
sns.set(style='whitegrid')  
d = sns.load_dataset('tips')
```

```
d  
p = sns.boxplot(x = 'smoker', y ='tip', hue ='sex', data=d)
```



In [78]:

```
import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
d = sns.load_dataset('tips')  
d
```

Out[78]:

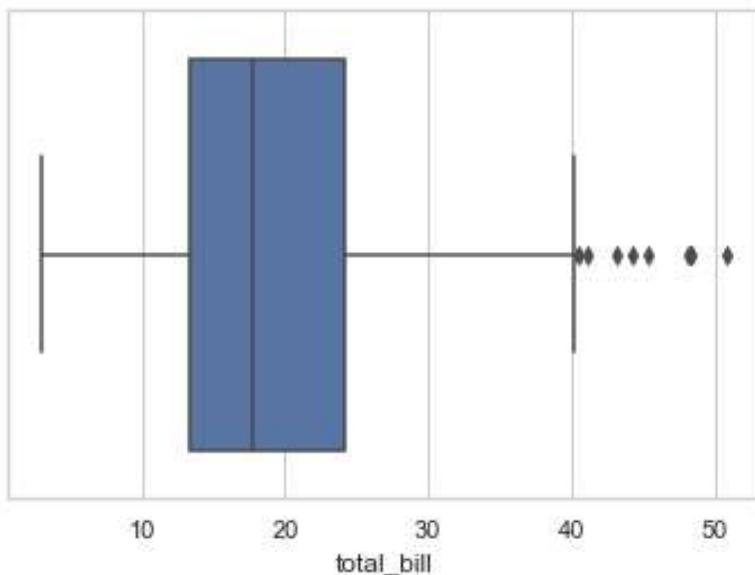
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	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

244 rows × 7 columns

In [79]:

```
sns.boxplot(x = d['total_bill'])
```

```
Out[79]: <AxesSubplot:xlabel='total_bill'>
```

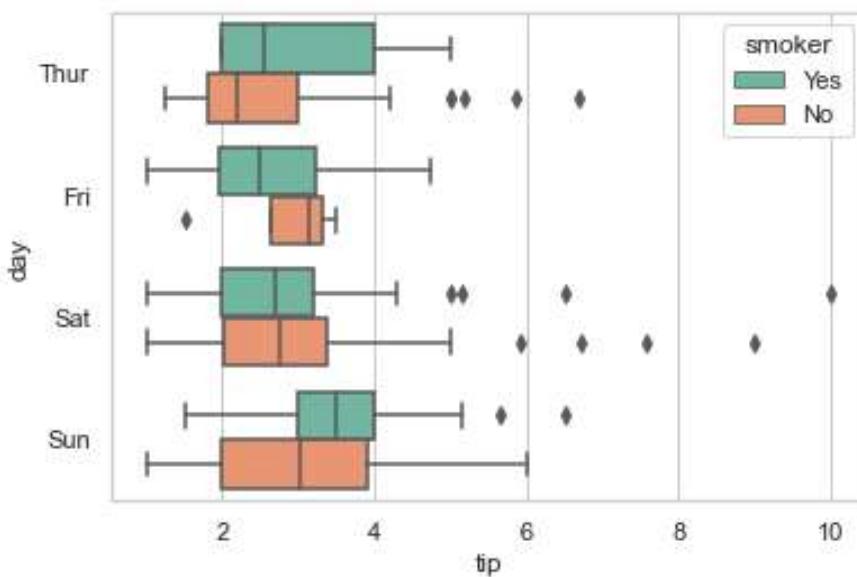


```
In [90]:
```

```
import seaborn as sns
sns.set(style = 'whitegrid')
d = sns.load_dataset('tips')
d.describe()
sns.boxplot(x = d['tip'], y = d['day'], hue = 'smoker', data = d,
            palette = 'Set2', dodge = True)

#dodge = separating bars
```

```
Out[90]: <AxesSubplot:xlabel='tip', ylabel='day'>
```



```
In [2]:
```

```
import seaborn as sns
import numpy as np
import pandas as pd

d = sns.load_dataset('titanic')
d.head()
```

```
Out[2]:
```

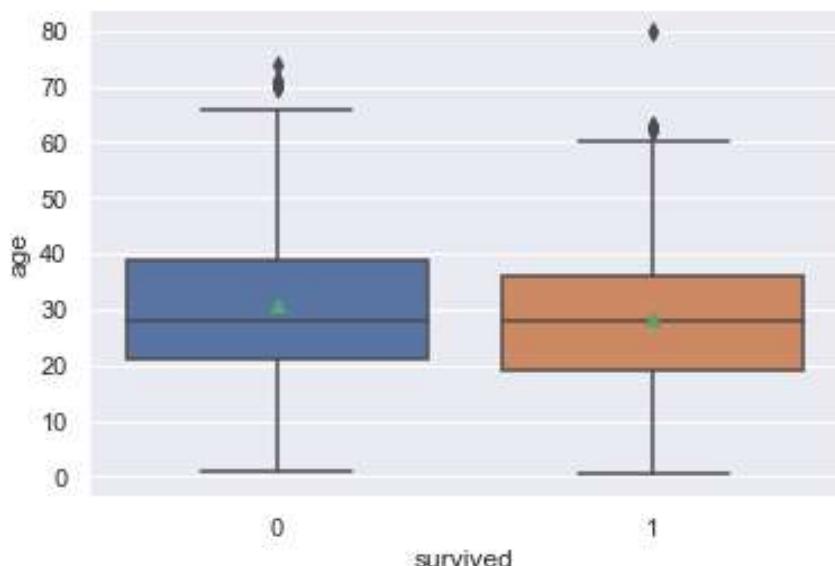
survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adu
----------	--------	-----	-----	-------	-------	------	----------	-------	-----	-----

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adu
0	0	3	male	22.0	1	0	7.2500	S	Third	man	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	

In [6]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
sns.set_theme(style='darkgrid')

d = sns.load_dataset('titanic')
p1 = sns.boxplot(x = 'survived', y = 'age', data = d, showmeans=True)
```



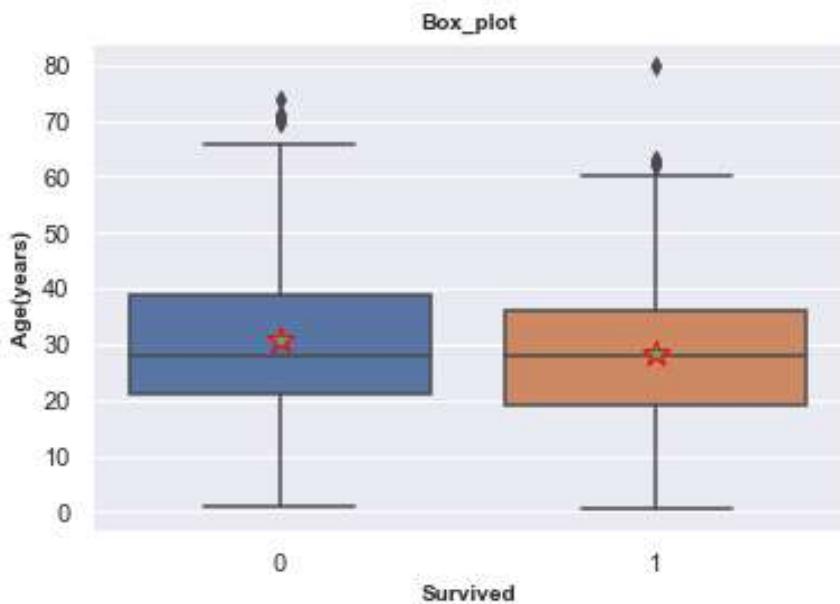
In [9]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
sns.set_theme(style='darkgrid')

d = sns.load_dataset('titanic')

p1 = sns.boxplot(x = 'survived', y = 'age', data = d, showmeans=True,
                  meanprops = {'marker':'*',
                               'markersize':'12',
                               'markeredgecolor':'red'})
plt.xlabel('Survived', size = 10, weight ='bold')
plt.ylabel('Age(years)',size = 10, weight ='bold')
plt.title('Box_plot', size = 10, weight ='bold')
```

Out[9]: Text(0.5, 1.0, 'Box\_plot')



## Other Plots

### Line plot

In [12]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

d = sns.load_dataset('dots')
d.head()
```

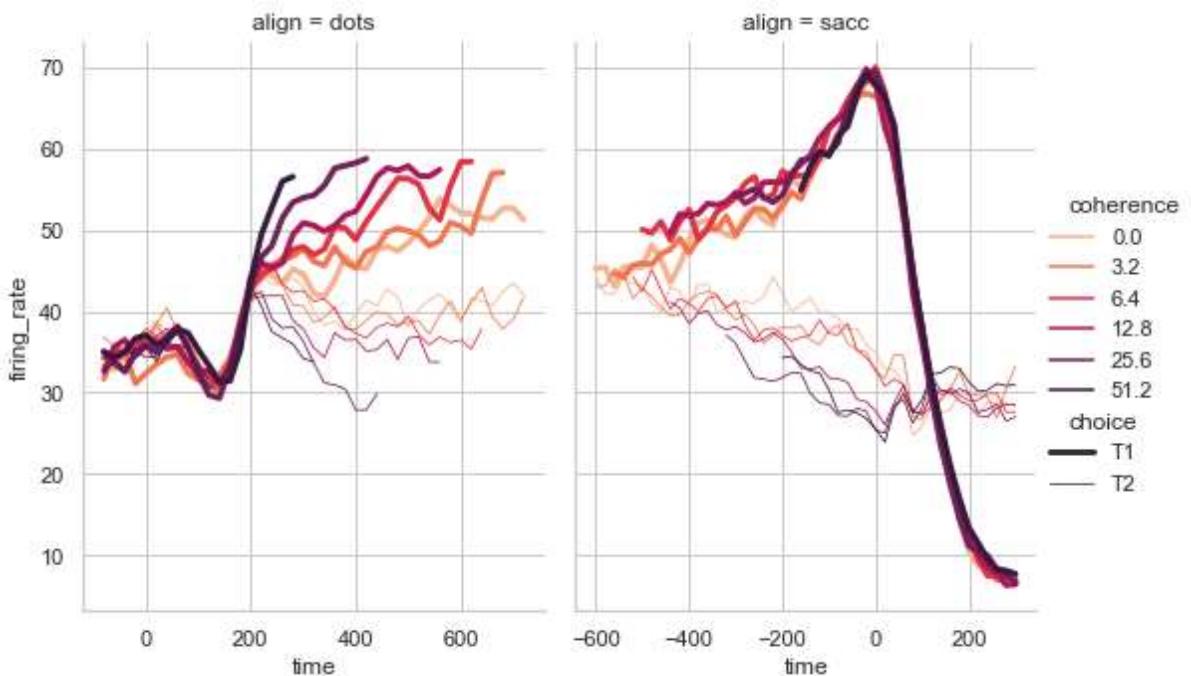
Out[12]:

	<th>choice</th> <th>time</th> <th>coherence</th> <th>firing_rate</th>	choice	time	coherence	firing_rate
0	dots	T1	-80	0.0	33.189967
1	dots	T1	-80	3.2	31.691726
2	dots	T1	-80	6.4	34.279840
3	dots	T1	-80	12.8	32.631874
4	dots	T1	-80	25.6	35.060487

In [20]:

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
sns.set_theme(style='whitegrid')
d = sns.load_dataset('dots')
p = sns.color_palette('rocket_r')
sns.relplot(data = d, x = 'time', y='firing_rate',
            hue = 'coherence', size = 'choice',
            col='align', kind = 'line', size_order=['T1','T2'],
```

```
palette=p, height=5, aspect=.75, facet_kws=dict(sharex=False),  
)  
  
Out[20]: <seaborn.axisgrid.FacetGrid at 0x20aaa89a920>
```



## Num Py

'Intel MKL' or 'OpenBLAS'

### Arrays

```
In [23]: !pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (1.21.4)
```

```
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
```

```
You should consider upgrading via the 'C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

```
In [24]: import numpy as np  
a = np.array([5,5,5])  
a
```

```
Out[24]: array([5, 5, 5])
```

```
In [25]: type(a)
```

```
Out[25]: numpy.ndarray
```

```
In [30]: b = np.array([[1,1,1],[1,1,1],[1,1,1]])  
b
```

```
Out[30]: array([[1, 1, 1],  
                 [1, 1, 1],  
                 [1, 1, 1]])
```

```
In [31]: type(b)
```

```
Out[31]: numpy.ndarray
```

```
In [32]: import numpy as np  
a = np.array([1,2,2,3,4])  
a
```

```
Out[32]: array([1, 2, 2, 3, 4])
```

```
In [34]: b = np.zeros(2)  
b
```

```
Out[34]: array([0., 0.])
```

```
In [35]: b = np.ones(6)  
b
```

```
Out[35]: array([1., 1., 1., 1., 1., 1.])
```

```
In [38]: c = np.empty(3)  
c
```

```
Out[38]: array([ 1.10878006e-311,  0.00000000e+000, -4.94065646e-324])
```

```
In [39]: a = np.arange(6)  
a
```

```
Out[39]: array([0, 1, 2, 3, 4, 5])
```

```
In [41]: a = np.arange(2,20,2)  
a
```

```
Out[41]: array([ 2,  4,  6,  8, 10, 12, 14, 16, 18])
```

```
In [45]: a = np.linspace(0,23,num=5)  
a
```

```
Out[45]: array([ 0. ,  5.75, 11.5 , 17.25, 23. ])
```

```
In [46]: a = np.ones(5,dtype=np.int8)
a
```

```
Out[46]: array([1, 1, 1, 1, 1], dtype=int8)
```

```
In [50]: a = np.zeros(6, dtype=np.float64)
a
```

```
Out[50]: array([0., 0., 0., 0., 0., 0.])
```

```
In [52]: a = np.zeros((5,6))
a
```

```
Out[52]: array([[0., 0., 0., 0., 0., 0.],
 [0., 0., 0., 0., 0., 0.],
 [0., 0., 0., 0., 0., 0.],
 [0., 0., 0., 0., 0., 0.],
 [0., 0., 0., 0., 0., 0.]])
```

```
In [57]: a = np.arange(24).reshape(2,3,4)
a
```

```
Out[57]: array([[[ 0,  1,  2,  3],
 [ 4,  5,  6,  7],
 [ 8,  9, 10, 11]],

 [[12, 13, 14, 15],
 [16, 17, 18, 19],
 [20, 21, 22, 23]]])
```

```
In [8]: import numpy as np
b = np.array([1,2,3,4,5])
b
```

```
Out[8]: array([1, 2, 3, 4, 5])
```

```
In [3]: a = np.array(['a','b','c','d'])
a
```

```
Out[3]: array(['a', 'b', 'c', 'd'], dtype='<U1')
```

```
In [4]: type(a)
```

```
Out[4]: numpy.ndarray
```

```
In [5]:
```

```
len(a)
```

Out[5]: 4

```
a[2]
```

Out[6]: 'c'

```
b.mean()
```

Out[9]: 3.0

```
np.arange(2,20)
```

Out[10]: array([ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])

```
np.arange(2,20,2)
```

Out[11]: array([ 2, 4, 6, 8, 10, 12, 14, 16, 18])

```
np.linspace(0,17,num=4)
```

Out[17]: array([ 0. , 5.66666667, 11.33333333, 17. ])

```
np.ones(4, dtype=np.int64)
```

Out[21]: array([1, 1, 1, 1], dtype=int64)

---

## Array functions

```
a = np.array([10,11,22,11,11,2,33,4.9,4])  
a
```

Out[22]: array([10. , 11. , 22. , 11. , 11. , 2. , 33. , 4.9, 4. ])

```
a.sort()  
a
```

Out[25]: array([ 2. , 4. , 4.9, 10. , 11. , 11. , 11. , 22. , 33. ])

```
b = np.array([1,2,3,4,45,6,6,6,7.8])
```

```
np.concatenate((a,b))

Out[26]: array([ 2. ,  4. ,  4.9, 10. , 11. , 11. , 11. , 22. , 33. ,  1. ,  2. ,
       3. ,  4. , 45. ,  6. ,  6. ,  6. , 7.8])

In [31]: a = np.array([[12,32,3],[1,2,2]])
a

Out[31]: array([[12, 32,  3],
       [ 1,   2,   2]])

In [32]: b = np.array([[2,3,4],[5,6,7]])
b

Out[32]: array([[2, 3,  4],
       [ 5, 6,  7]])

In [34]: np.concatenate((a,b), axis=0)

Out[34]: array([[12, 32,  3],
       [ 1,   2,   2],
       [ 2,   3,   4],
       [ 5,   6,   7]])

In [35]: np.concatenate((a,b), axis=1)

Out[35]: array([[12, 32,  3,  2,  3,  4],
       [ 1,   2,   2,   5,   6,   7]])

In [36]: a = np.array([[[1,23,4]],[[2,3,4]],[[1,2,3]]])
a

Out[36]: array([[[ 1, 23,  4]],
       [[[ 2,   3,   4]],
        [[ 1,   2,   3]]])

In [37]: a.ndim

Out[37]: 3

In [38]: b = np.array([[9,0,6],[1,2,3],[3,2,2]])
b

Out[38]: array([[9, 0, 6],
       [1, 2, 3],
       [3, 2, 2]])

In [39]: b.ndim
```

```
Out[39]: 2
```

```
In [40]: a.size
```

```
Out[40]: 9
```

```
In [41]: b.shape
```

```
Out[41]: (3, 3)
```

```
In [42]: a = np.array([1,2,3,4,45,43])
a.reshape(3,2)
```

```
Out[42]: array([[ 1,  2],
                 [ 3,  4],
                 [45, 43]])
```

```
In [43]: a = np.array([1,2,3,43,4,5,4])
a
```

```
Out[43]: array([ 1,  2,  3, 43,  4,  5,  4])
```

```
In [44]: a.shape
```

```
Out[44]: (7,)
```

```
In [45]: b = a[np.newaxis, :]      #row_wise 2D conversion
```

```
In [46]: b
```

```
Out[46]: array([[ 1,  2,  3, 43,  4,  5,  4]])
```

```
In [47]: b.shape
```

```
Out[47]: (1, 7)
```

```
In [49]: b = a[:, np.newaxis]      #column_wise 2D conversion
b
```

```
Out[49]: array([[ 1],
                 [ 2],
                 [ 3],
                 [43],
                 [ 4],
```

```
[ 5],  
[ 4]])
```

```
In [50]: a.sum()
```

```
Out[50]: 62
```

```
In [52]: a.mean()
```

```
Out[52]: 8.857142857142858
```

```
In [54]: a.max()
```

```
Out[54]: 43
```

---

## ML Practice

```
In [1]: !pip install scikit-learn
```

```
Collecting scikit-learn  
  Downloading scikit_learn-1.0.2-cp310-cp310-win_amd64.whl (7.2 MB)  
Collecting joblib>=0.11  
  Using cached joblib-1.1.0-py2.py3-none-any.whl (306 kB)  
Requirement already satisfied: numpy>=1.14.6 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from scikit-learn) (1.21.4)  
Requirement already satisfied: scipy>=1.1.0 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from scikit-learn) (1.7.3)  
Collecting threadpoolctl>=2.0.0  
  Using cached threadpoolctl-3.0.0-py3-none-any.whl (14 kB)  
Installing collected packages: threadpoolctl, joblib, scikit-learn  
Successfully installed joblib-1.1.0 scikit-learn-1.0.2 threadpoolctl-3.0.0  
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.  
You should consider upgrading via the 'C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

```
In [2]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model_selection import train_test_split
```

```
In [3]: d = pd.read_csv('C:/Users/Gulfam/Downloads/biostats.csv')
```

```
In [4]: d
```

Out[4]:

	Name	"Sex"	"Age"	"Height (in)"	"Weight (lbs)"
0	Alex	"M"	41	74	170
1	Bert	"M"	42	68	166
2	Carl	"M"	32	70	155
3	Dave	"M"	39	72	167
4	Elly	"F"	30	66	124
5	Fran	"F"	33	66	115
6	Gwen	"F"	26	64	121
7	Hank	"M"	30	71	158
8	Ivan	"M"	53	72	175
9	Jake	"M"	32	69	143
10	Kate	"F"	47	69	139
11	Luke	"M"	34	72	163
12	Myra	"F"	23	62	98
13	Neil	"M"	36	75	160
14	Omar	"M"	38	70	145
15	Page	"F"	31	67	135
16	Quin	"M"	29	71	176
17	Ruth	"F"	28	65	131

In [5]:

```
d.shape
```

Out[5]:

```
(18, 5)
```

In [6]:

```
d.columns
```

Out[6]:

```
Index(['Name', 'Sex', 'Age', 'Height (in)', 'Weight (lbs)'], dtype='object')
```

In [9]:

```
dict = { 'Age': 'age',
         'Height (in)': 'height',
         'Weight (lbs)': 'weight' }
```

```
d.rename(columns=dict, inplace=True)
```

In [10]:

```
d.head()
```

Out[10]:

	Name	"Sex"	age	height	weight
--	------	-------	-----	--------	--------

	Name	"Sex"	age	height	weight
0	Alex	"M"	41	74	170
1	Bert	"M"	42	68	166
2	Carl	"M"	32	70	155
3	Dave	"M"	39	72	167
4	Elly	"F"	30	66	124

```
In [11]: d1 = d[['age', 'height', 'weight']]
```

```
In [12]: d1.head()
```

```
Out[12]:   age  height  weight
```

0	41	74	170
1	42	68	166
2	32	70	155
3	39	72	167
4	30	66	124

```
In [57]: X = d1.iloc[:, :-1].values  
y = d1.iloc[:, 2].values
```

```
In [58]: y
```

```
Out[58]: array([170, 166, 155, 167, 124, 115, 121, 158, 175, 143, 139, 163, 98,  
       160, 145, 135, 176, 131], dtype=int64)
```

```
In [59]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3,
```

```
In [60]: from sklearn.linear_model import LinearRegression  
regressor = LinearRegression()  
regressor.fit(X_train, y_train)
```

```
Out[60]: LinearRegression()
```

```
In [61]: y_pred = regressor.predict([[41, 74]])
```

```
In [62]: y_pred
```

```
Out[62]: array([168.47242172])
```

```
In [ ]:
```

```
In [63]: d1.height.describe()
```

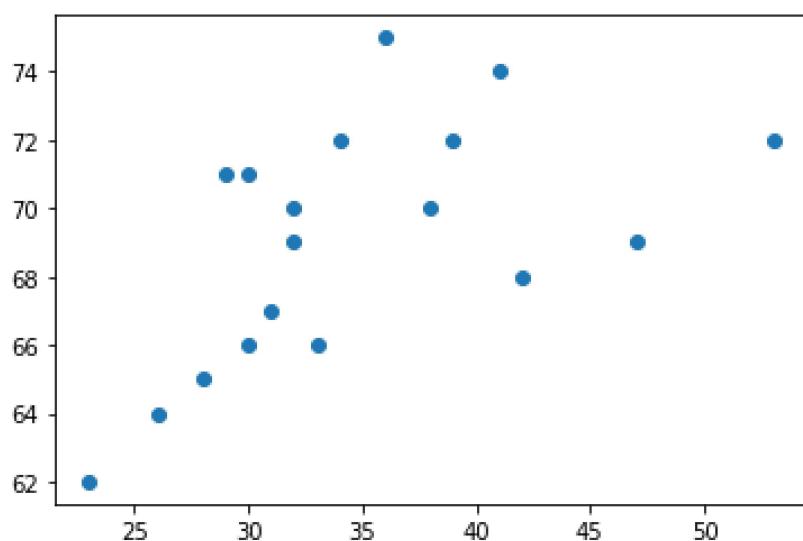
```
Out[63]: count    18.000000
mean     69.055556
std      3.522570
min     62.000000
25%    66.250000
50%    69.500000
75%    71.750000
max     75.000000
Name: height, dtype: float64
```

```
In [64]: d1.head()
```

```
Out[64]:   age  height  weight
0    41      74     170
1    42      68     166
2    32      70     155
3    39      72     167
4    30      66     124
```

```
In [65]: plt.scatter('age','height', data=d1)
```

```
Out[65]: <matplotlib.collections.PathCollection at 0x21a86632dd0>
```



```
In [66]: regressor.fit(d1[['age','height']],d1.weight)
```

```
Out[66]: LinearRegression()
```

```
In [67]: regressor.coef_
```

```
Out[67]: array([0.3559392 , 5.15319898])
```

```
In [68]: regressor.intercept_
```

```
Out[68]: -221.47402175851624
```

```
In [69]: regressor.predict([[35,70]])
```

```
C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\linalg\__init__.py:352: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
```

```
    warnings.warn(
```

```
Out[69]: array([151.70777877])
```

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

```
In [2]: d = sns.load_dataset('iris')
```

```
In [3]: X = d.iloc[:, :-1]
```

```
In [4]: y = d.iloc[:, -1:]
```

```
In [13]: X.tail()
```

```
Out[13]:
```

	sepal_length	sepal_width	petal_length	petal_width
<b>145</b>	6.7	3.0	5.2	2.3
<b>146</b>	6.3	2.5	5.0	1.9
<b>147</b>	6.5	3.0	5.2	2.0
<b>148</b>	6.2	3.4	5.4	2.3
<b>149</b>	5.9	3.0	5.1	1.8

```
In [5]: !pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\gulfam\appdata\loc
```

```

al\programs\python\python310\lib\site-packages (1.0.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\gulfam\app
data\local\programs\python\python310\lib\site-packages (from scikit-learn)
(3.0.0)
Requirement already satisfied: joblib>=0.11 in c:\users\gulfam\appdata\loc
al\programs\python\python310\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\gulfam\appdata\loc
al\programs\python\python310\lib\site-packages (from scikit-learn) (1.7.3)
Requirement already satisfied: numpy>=1.14.6 in c:\users\gulfam\appdata\lo
cal\programs\python\python310\lib\site-packages (from scikit-learn) (1.21.
4)

WARNING: You are using pip version 21.2.3; however, version 21.3.1 is avai
lable.
You should consider upgrading via the 'C:\Users\Gulfam\AppData\Local\Progr
ams\Python\Python310\python.exe -m pip install --upgrade pip' command.

```

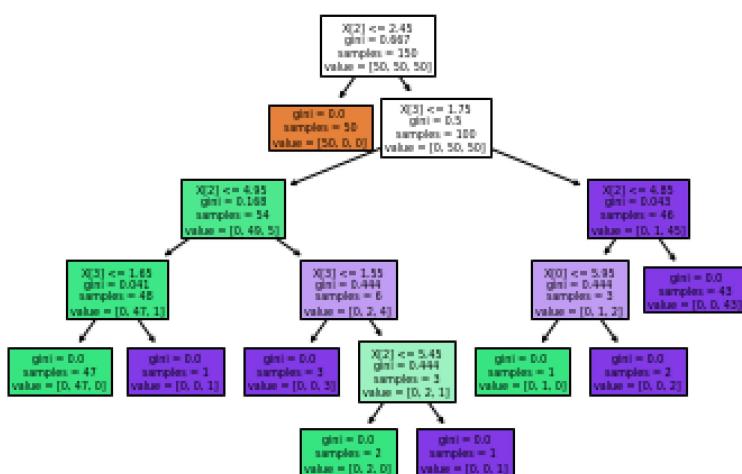
```
In [6]: from sklearn.tree import DecisionTreeClassifier
```

```
In [7]: model = DecisionTreeClassifier().fit(X,y)
```

```
In [8]: from sklearn.tree import plot_tree
```

```
In [21]: plot_tree(model, filled=True)
plt.title('Iris Decision Tree')
plt.show()
plt.savefig('C:/Users/Gulfam/Desktop/Data Sci/ML/IrisDT.png')
```

Iris Decision Tree



<Figure size 432x288 with 0 Axes>

```
In [16]: model.predict([[5.6,3.0,4.1,1.8]])
```

```
C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\lib\site-packages
\sklearn\base.py:450: UserWarning: X does not have valid feature names, bu
t DecisionTreeClassifier was fitted with feature names
    warnings.warn(
        array(['versicolor'], dtype=object)
```

Out[16]:

In [22]:

```
!pip install pandoc
```

```
Collecting pandoc
  Downloading pandoc-2.0.1.tar.gz (29 kB)
Collecting plumbum
  Downloading plumbum-1.7.2-py2.py3-none-any.whl (117 kB)
Collecting ply
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
Requirement already satisfied: pywin32 in c:\users\gulfam\appdata\local\programs\python\python310\lib\site-packages (from plumbum->pandoc) (302)
Using legacy 'setup.py install' for pandoc, since package 'wheel' is not installed.
Installing collected packages: ply, plumbum, pandoc
  Running setup.py install for pandoc: started
  Running setup.py install for pandoc: finished with status 'done'
Successfully installed pandoc-2.0.1 plumbum-1.7.2 ply-3.11
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\Gulfam\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

In [ ]: