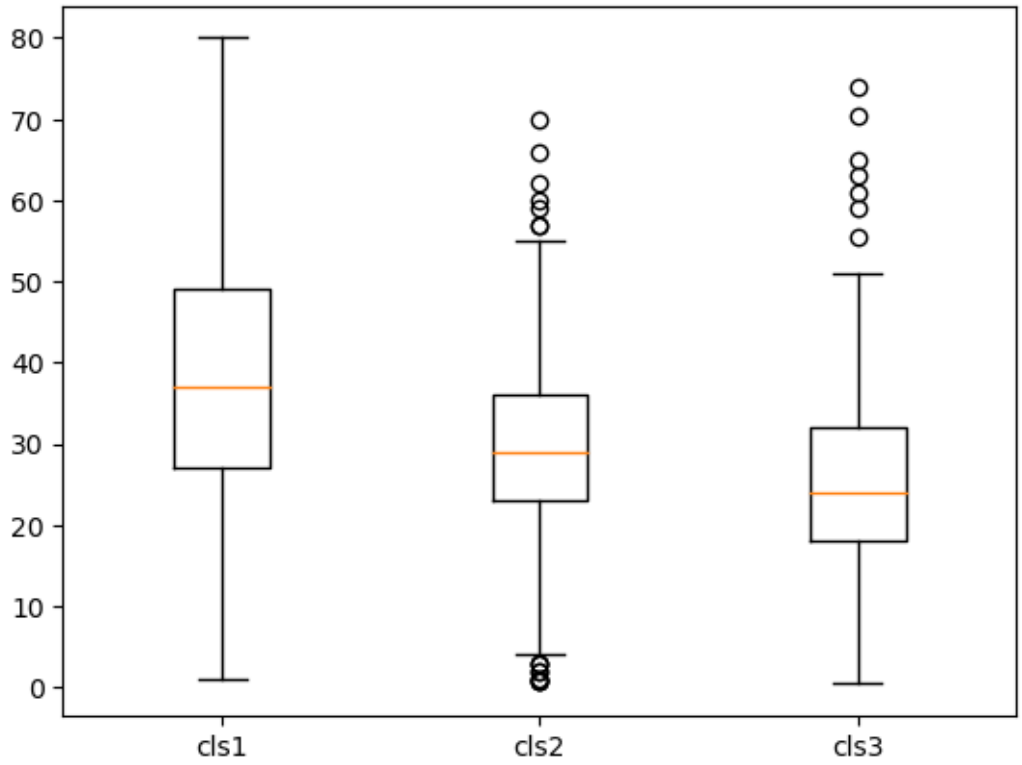
day-4-630

February 15, 2024

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
df_titanic = pd.read_csv('titanic_train.csv')
cls1=df_titanic[df_titanic['Pclass']==1]['Age'].dropna()
cls2=df_titanic[df_titanic['Pclass']==2]['Age'].dropna()
cls3=df_titanic[df_titanic['Pclass']==3]['Age'].dropna()
```

[2]:
```
l1=[cls1,cls2,cls3]
plt.boxplot(l1 ,labels=["cls1","cls2","cls3"])
```

[2]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x214062a0310>,
  <matplotlib.lines.Line2D at 0x21406651250>,
  <matplotlib.lines.Line2D at 0x2140665d850>,
  <matplotlib.lines.Line2D at 0x2140665e390>,
  <matplotlib.lines.Line2D at 0x2140666e810>,
  <matplotlib.lines.Line2D at 0x2140666f350>],
 'caps': [<matplotlib.lines.Line2D at 0x21406614d10>,
  <matplotlib.lines.Line2D at 0x21406652ad0>,
  <matplotlib.lines.Line2D at 0x2140665ef50>,
  <matplotlib.lines.Line2D at 0x2140665fa90>,
  <matplotlib.lines.Line2D at 0x2140666ff10>,
  <matplotlib.lines.Line2D at 0x21406674ad0>],
 'boxes': [<matplotlib.lines.Line2D at 0x2140663ec50>,
  <matplotlib.lines.Line2D at 0x2140665cd50>,
  <matplotlib.lines.Line2D at 0x2140666dcd0>],
 'medians': [<matplotlib.lines.Line2D at 0x214066536d0>,
  <matplotlib.lines.Line2D at 0x2140666c590>,
  <matplotlib.lines.Line2D at 0x214066755d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x2140665c110>,
  <matplotlib.lines.Line2D at 0x2140666d090>,
  <matplotlib.lines.Line2D at 0x214066760d0>],
 'means': []}
```



[3]:
```
df_titanic.rename(columns={'Sex':'Gender'})
```

[3]:

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	
..	…	…	…	
886	887	0	2	
887	888	1	1	
888	889	0	3	
889	890	1	1	
890	891	0	3	

	Name	Gender	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	
..	…	…	…	…	

```
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[4]:
```
df_titanic.rename(columns={'Sex':'Gender'},inplace=True)
df_titanic
```

[4]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …       …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                 Name  Gender   Age  SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                              Heikkinen, Miss. Laina  female  26.0      0
3        Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                            Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
```

```
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …       …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[5]:
```
df_titanic['Gender']=df_titanic['Gender'].map({'male':0,'female':1})
df_titanic
```

[5]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …       …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                 Name  Gender   Age  SibSp  \
0                             Braund, Mr. Owen Harris       0  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…       1  38.0      1
2                                Heikkinen, Miss. Laina       1  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)       1  35.0      1
4                               Allen, Mr. William Henry       0  35.0      0
..                                                   …     …   …    …
886                              Montvila, Rev. Juozas       0  27.0      0
887                       Graham, Miss. Margaret Edith       1  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"       1   NaN      1
889                              Behr, Mr. Karl Howell       0  26.0      0
890                                Dooley, Mr. Patrick       0  32.0      0
```

```
     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

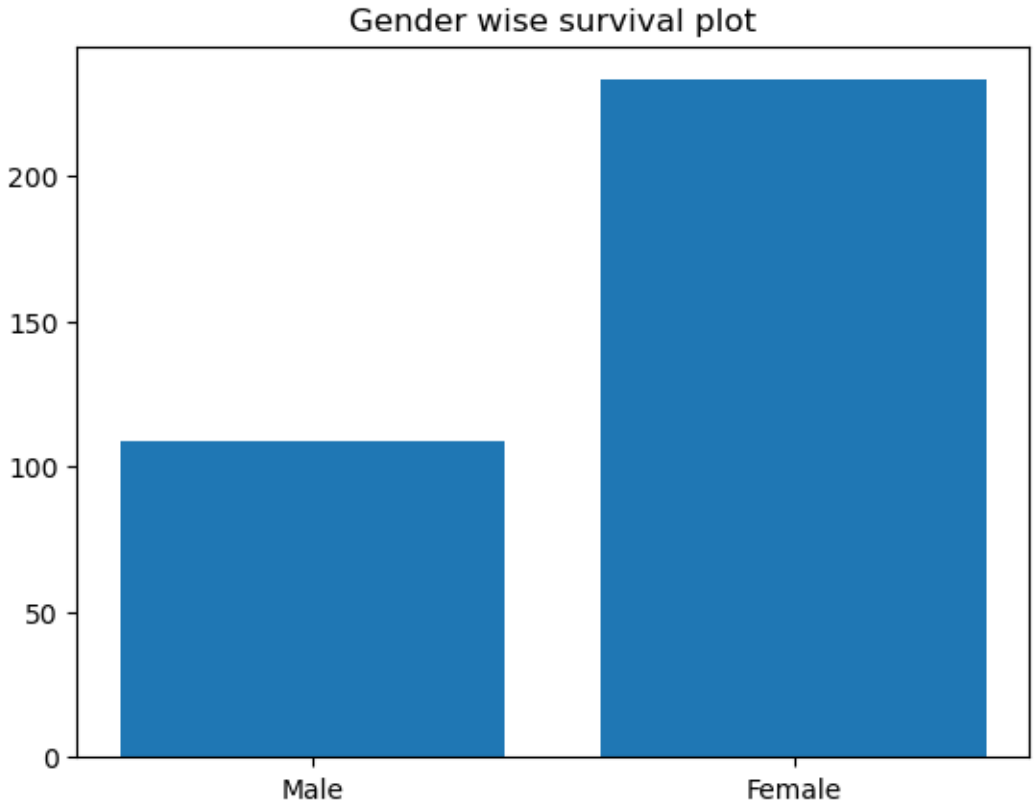
[6]:
```
df_titanic[(df_titanic['Gender']==1) & (df_titanic['Age'] < 25)]
```

[6]:
```
     PassengerId  Survived  Pclass                                     Name  \
9             10         1       2    Nasser, Mrs. Nicholas (Adele Achem)
10            11         1       3        Sandstrom, Miss. Marguerite Rut
14            15         0       3   Vestrom, Miss. Hulda Amanda Adolfina
22            23         1       3            McGowan, Miss. Anna "Annie"
24            25         0       3          Palsson, Miss. Torborg Danira
..           …       …     …                                        …
855          856         1       3             Aks, Mrs. Sam (Leah Rosen)
858          859         1       3  Baclini, Mrs. Solomon (Latifa Qurban)
875          876         1       3       Najib, Miss. Adele Kiamie "Jane"
882          883         0       3           Dahlberg, Miss. Gerda Ulrika
887          888         1       1           Graham, Miss. Margaret Edith

     Gender   Age  SibSp  Parch   Ticket     Fare Cabin Embarked
9         1  14.0      1      0   237736  30.0708   NaN        C
10        1   4.0      1      1  PP 9549  16.7000    G6        S
14        1  14.0      0      0   350406   7.8542   NaN        S
22        1  15.0      0      0   330923   8.0292   NaN        Q
24        1   8.0      3      1   349909  21.0750   NaN        S
..      …   …    …     …      …       …   …      …
855       1  18.0      0      1   392091   9.3500   NaN        S
858       1  24.0      0      3     2666  19.2583   NaN        C
875       1  15.0      0      0     2667   7.2250   NaN        C
882       1  22.0      0      0     7552  10.5167   NaN        S
887       1  19.0      0      0   112053  30.0000   B42        S

[117 rows x 12 columns]
```

[7]:
```
gender=['Male','Female']
male = ((df_titanic['Gender']==0) &( df_titanic['Survived'])).sum()
```

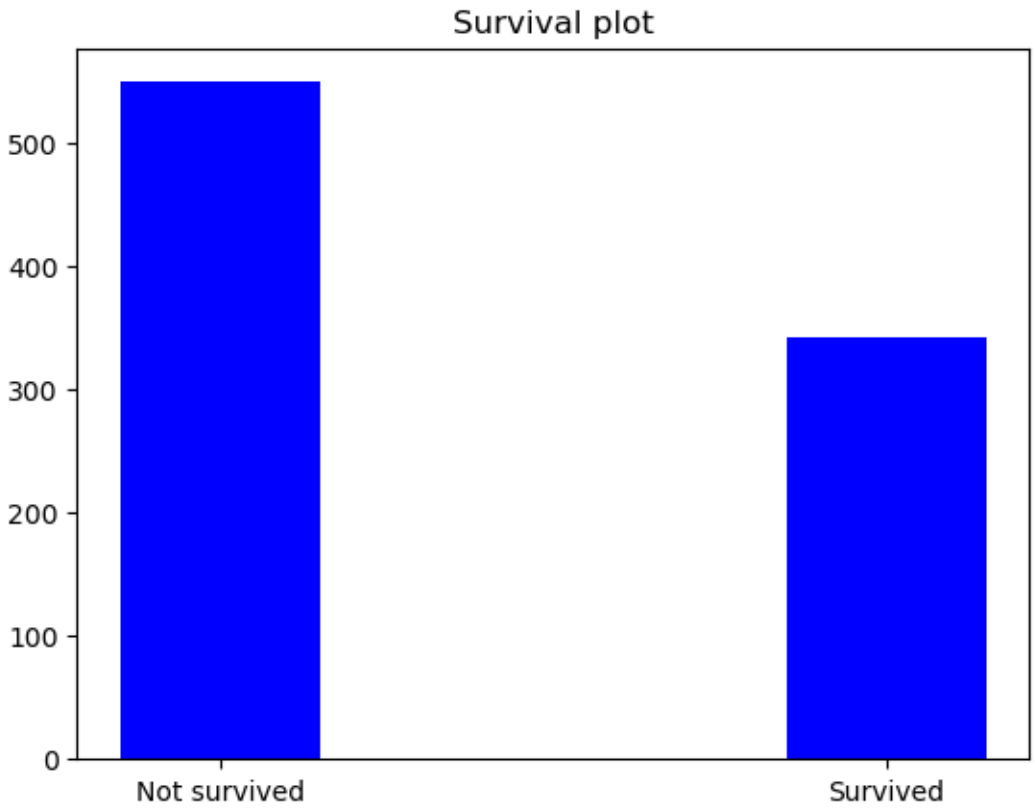
```
female = ((df_titanic['Gender']==1) & (df_titanic['Survived'])).sum()
count=[male,female]
plt.bar(gender,count)
plt.title("Gender wise survival plot")
```

[7]: Text(0.5, 1.0, 'Gender wise survival plot')



[8]:
```
gender=['Not survived','Survived']
not_survived = ((df_titanic['Survived']==0)).sum()
Survive = ((df_titanic['Survived']==1)).sum()
count=[not_survived,Survive]
plt.bar(gender,count,color='blue',width=0.3)
plt.title('Survival plot')
```

[8]: Text(0.5, 1.0, 'Survival plot')



[9]:
```
import seaborn as sns
tips = sns.load_dataset('tips')
tips
```

[9]:

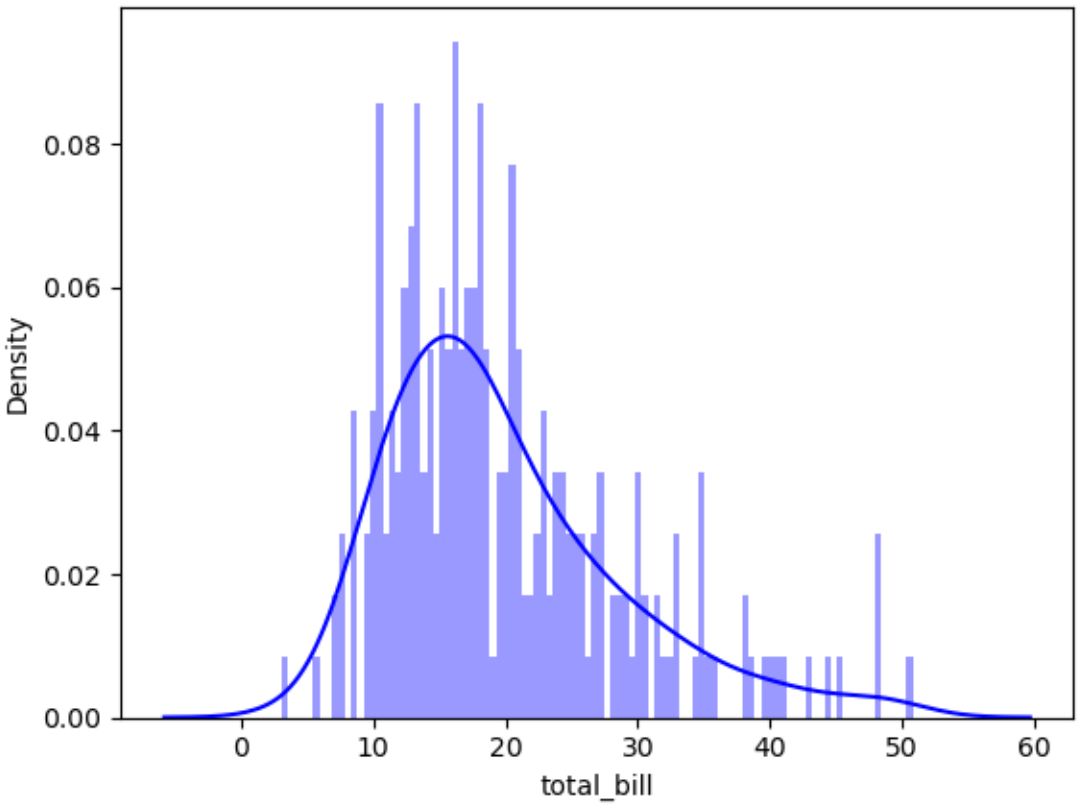
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
..	…	…	…	…	…	…	…
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	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 x 7 columns]

[10]:
```
sns.distplot(tips['total_bill'],bins=100,kde=True,hist=True,color='blue')
```

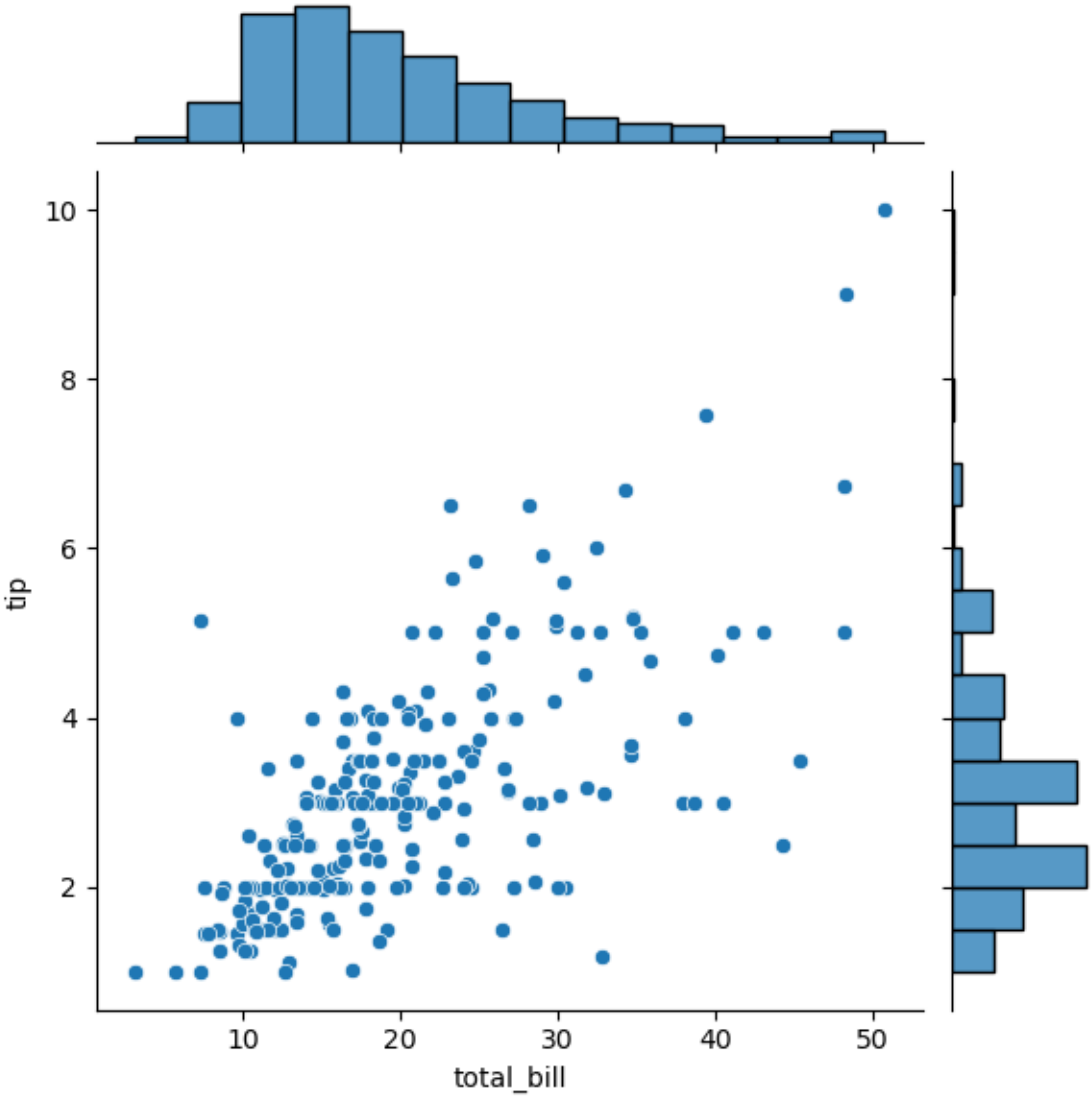
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_12768\83553870.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(tips['total_bill'],bins=100,kde=True,hist=True,color='blue')
```

[10]: <Axes: xlabel='total_bill', ylabel='Density'>



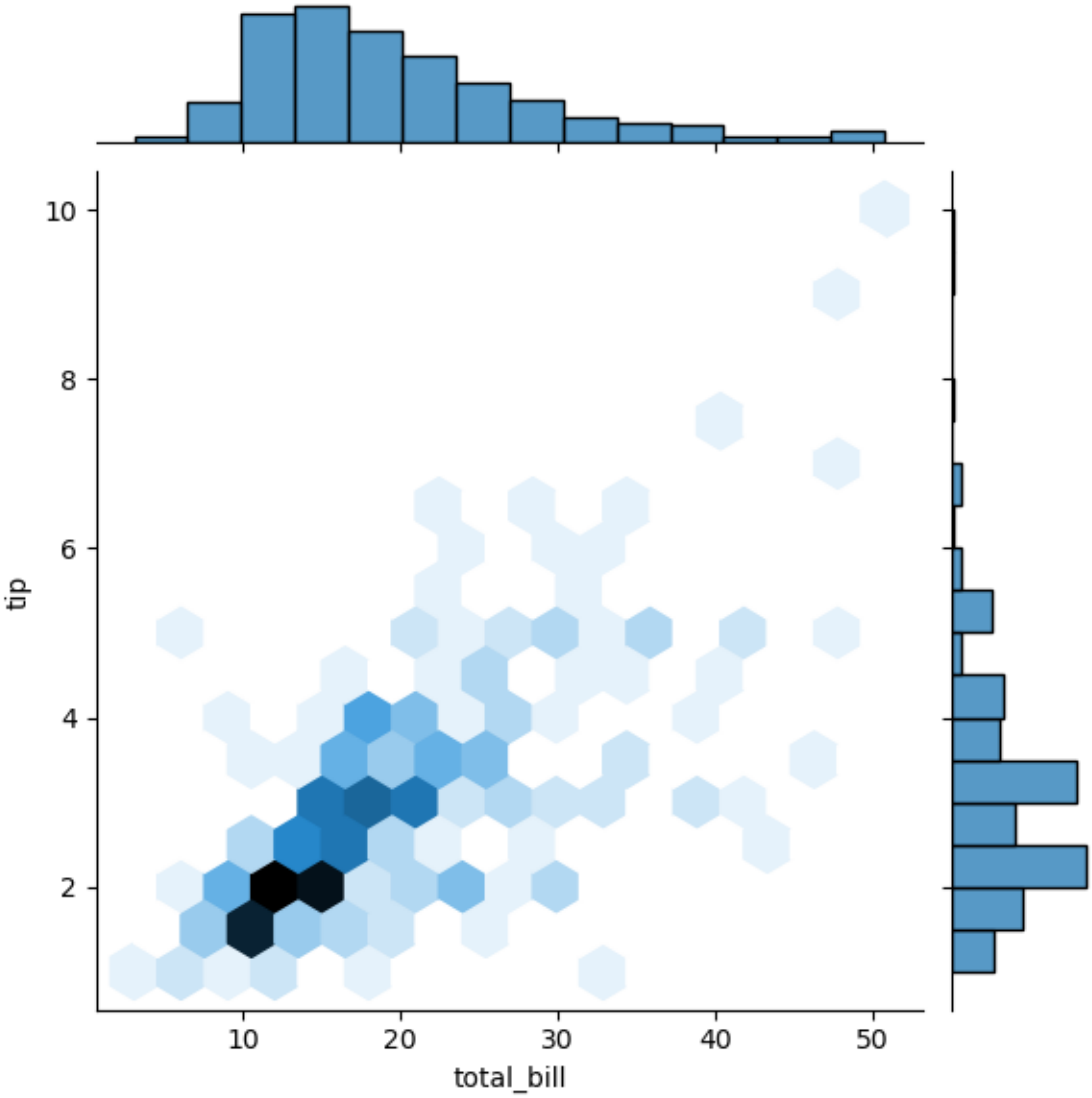
[11]:
```
sns.jointplot(x='total_bill',y='tip',data=tips)
```

[11]: <seaborn.axisgrid.JointGrid at 0x2140988b190>



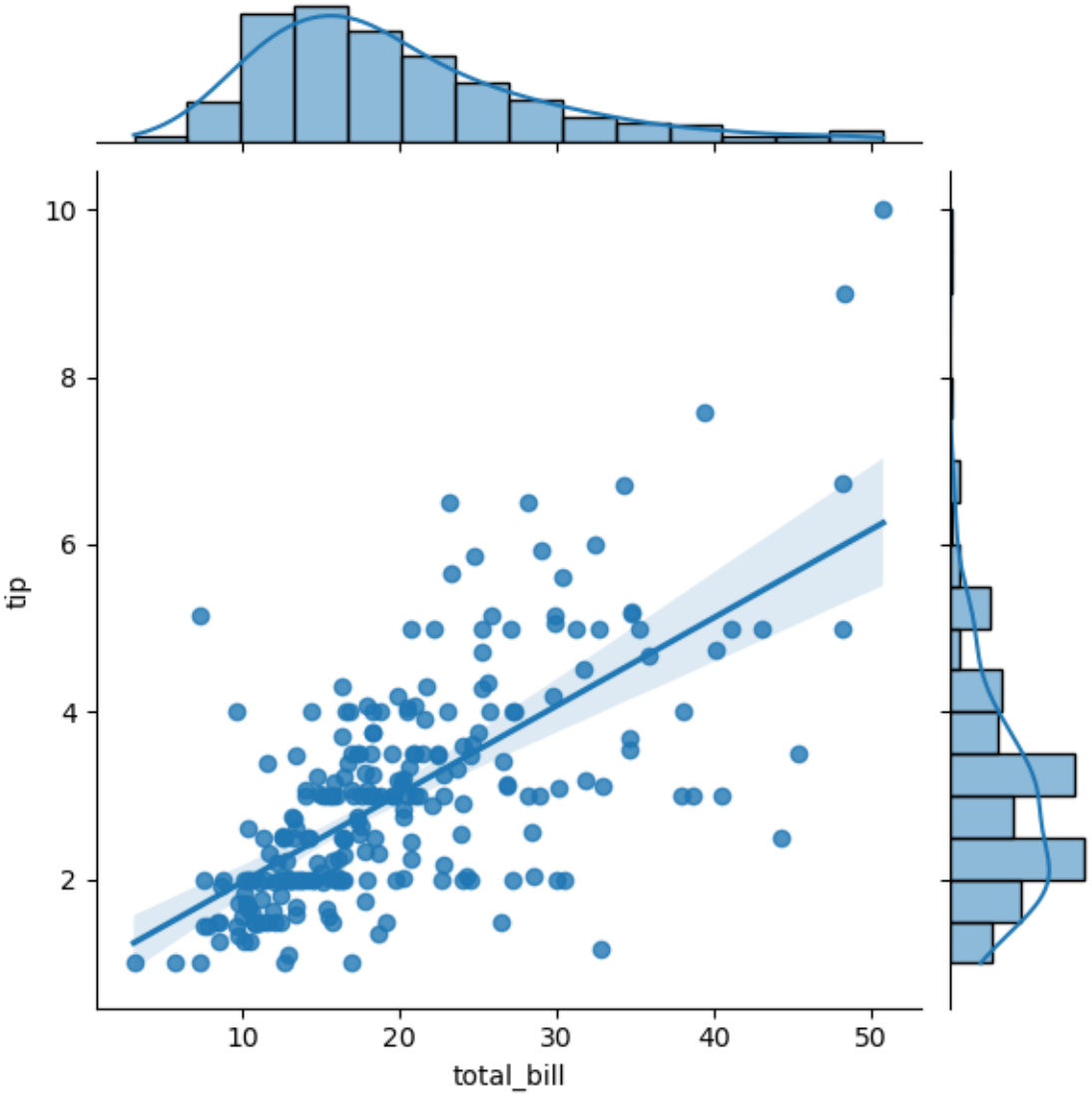
[12]:
```
sns.jointplot(x='total_bill',y='tip',data=tips,kind='hex')
```

[12]: <seaborn.axisgrid.JointGrid at 0x2140a3ea410>



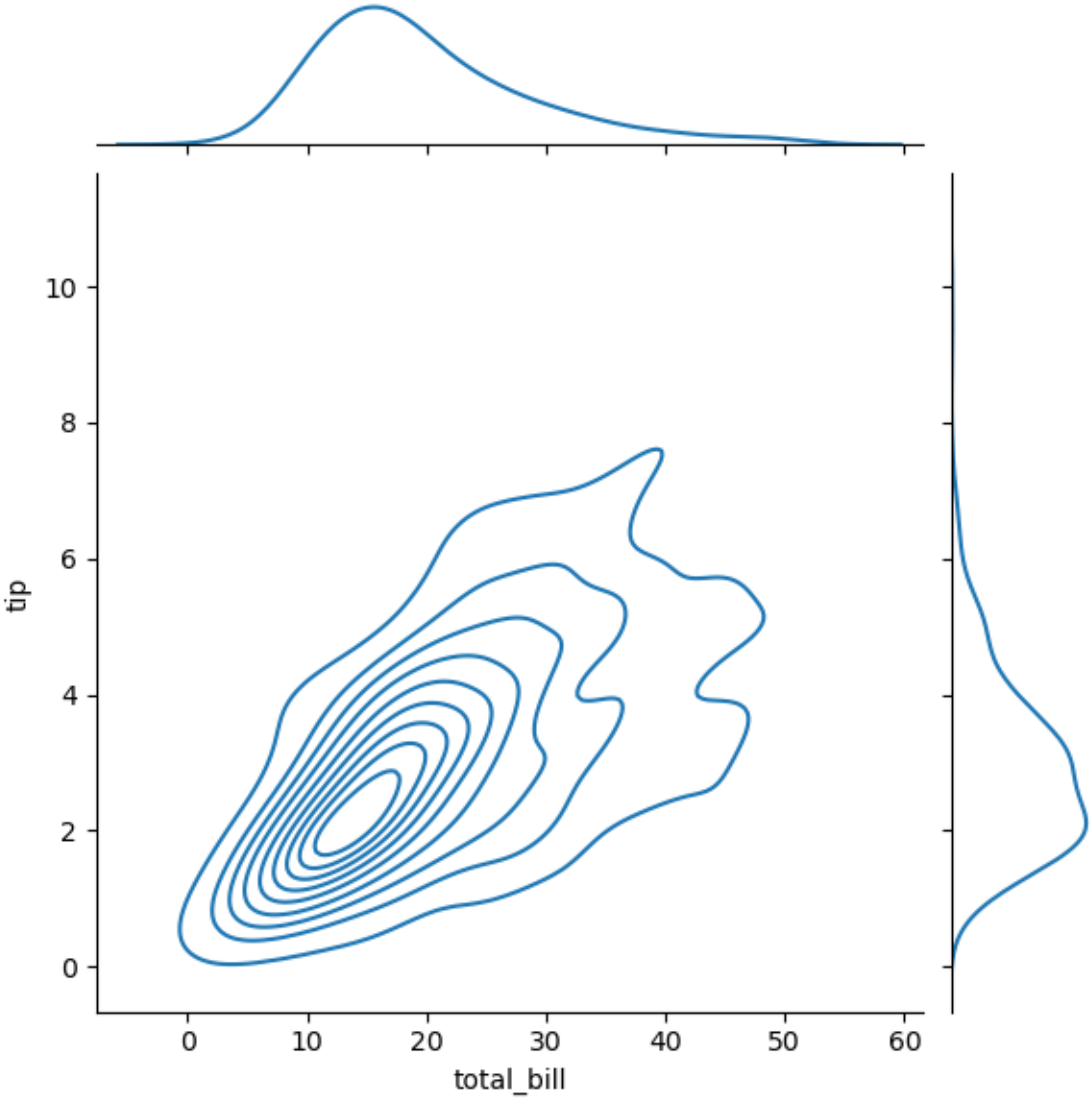
[13]:
```
sns.jointplot(x='total_bill',y='tip',data=tips,kind='reg')
```

[13]:
```
<seaborn.axisgrid.JointGrid at 0x2140a696690>
```



[14]:
```
sns.jointplot(x='total_bill',y='tip',data=tips,kind='kde')
```

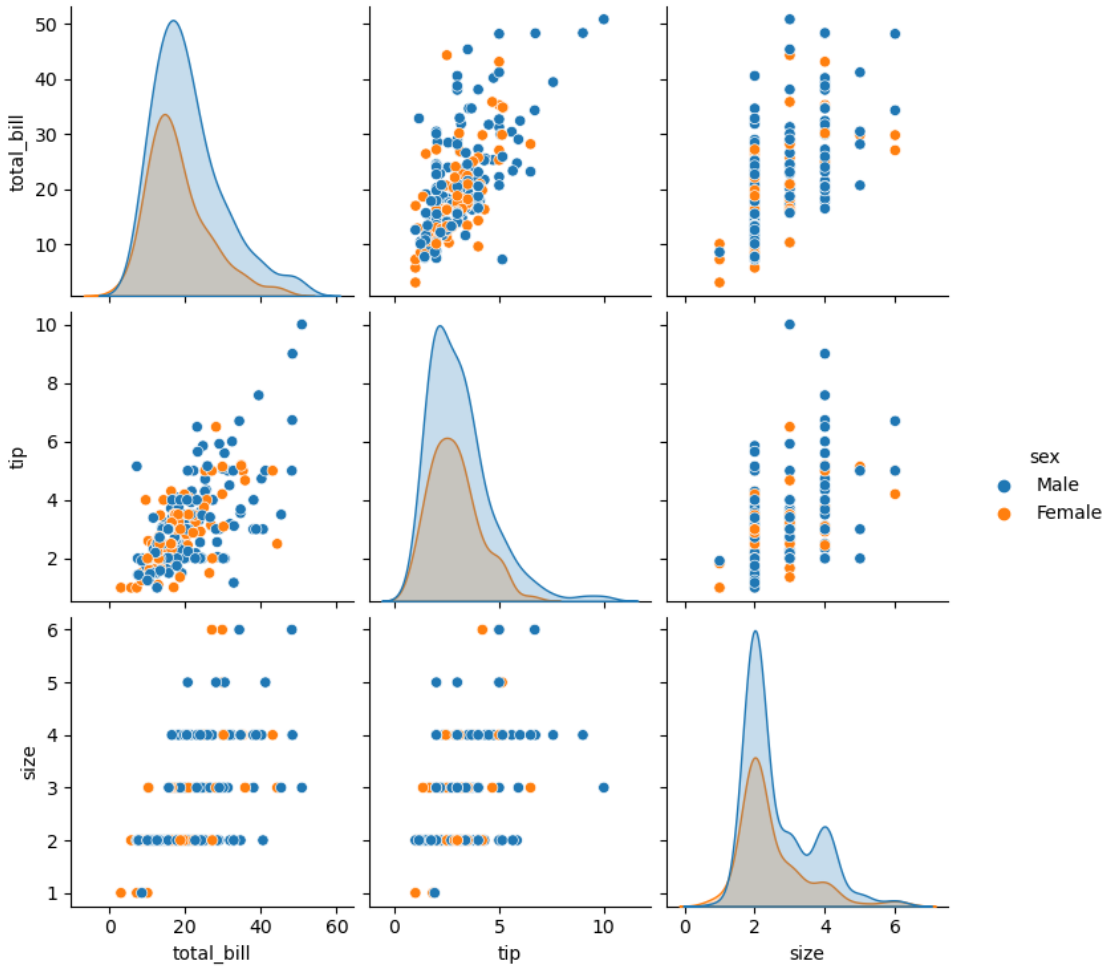
[14]:
```
<seaborn.axisgrid.JointGrid at 0x2140a696310>
```



[15]:
```
sns.pairplot(tips,hue='sex')
```

```
D:\anaconda\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure
layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[15]: <seaborn.axisgrid.PairGrid at 0x2140aed8d10>



[16]:
```
#machine learning algorithms
#logistic regressiion is a statistical method used to model the relationship␣
 ↪between a binary dependent variable and one or more independent variables in␣
 ↪logistic regression
#the dependent variable is binary meaning it can only take two values 0 or 1
#The independent variable is can be either continuous or catagorical
```

[17]:
```
import numpy as np
from sklearn.metrics import␣
 ↪accuracy_score,precision_score,recall_score,f1_score,confusion_matrix
```

[18]:
```
y_pred=np.array([0.3,0.6,0.8,0.2,0.4,0.9,0.1,0.7,0.5,0.6])
y_true=np.array([0,1,1,0,0,1,0,1,0,1])
```

[19]:
```
#accuracy
#accuracy meaures the percentage of correctly classified instancesout of all␣
 ↪instances
```

```
accuracy=accuracy_score(y_true,np.round(y_pred))
accuracy
```

[19]:
```
1.0
```

[20]:
```
# precision
# measures the proportion of true +ve prediction out of all +ve predictions
# precision = true +ve / all +ve
precision = precision_score(y_true,np.round(y_pred))
precision
```

[20]:
```
1.0
```

[21]:
```
# recall
# measures the proportion of true +ve prediction out of all actual true +ve␣
 ↪cases
# recall = true +ve / actual +ve
recall = recall_score(y_true,np.round(y_pred))
recall
```

[21]:
```
1.0
```

[22]:
```
# f1_score
# It is the mean of precision and recall
f1_score = f1_score(y_true,np.round(y_pred))
f1_score
```

[22]:
```
1.0
```

[23]:
```
# confusion matrix
# It is a table that gives the performance of a classification model
# It shows true +ve , true -ve , false +ve , false -ve
matrix = confusion_matrix(y_true,np.round(y_pred))
matrix
```

[23]:
```
array([[5, 0],
       [0, 5]], dtype=int64)
```

[24]:
```
# Example - 1 for logistic regression
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

[25]:
```
iris = load_iris()
iris
```

```
[25]: {'data': array([[5.1, 3.5, 1.4, 0.2],
                      [4.9, 3. , 1.4, 0.2],
                      [4.7, 3.2, 1.3, 0.2],
                      [4.6, 3.1, 1.5, 0.2],
                      [5. , 3.6, 1.4, 0.2],
                      [5.4, 3.9, 1.7, 0.4],
                      [4.6, 3.4, 1.4, 0.3],
                      [5. , 3.4, 1.5, 0.2],
                      [4.4, 2.9, 1.4, 0.2],
                      [4.9, 3.1, 1.5, 0.1],
                      [5.4, 3.7, 1.5, 0.2],
                      [4.8, 3.4, 1.6, 0.2],
                      [4.8, 3. , 1.4, 0.1],
                      [4.3, 3. , 1.1, 0.1],
                      [5.8, 4. , 1.2, 0.2],
                      [5.7, 4.4, 1.5, 0.4],
                      [5.4, 3.9, 1.3, 0.4],
                      [5.1, 3.5, 1.4, 0.3],
                      [5.7, 3.8, 1.7, 0.3],
                      [5.1, 3.8, 1.5, 0.3],
                      [5.4, 3.4, 1.7, 0.2],
                      [5.1, 3.7, 1.5, 0.4],
                      [4.6, 3.6, 1. , 0.2],
                      [5.1, 3.3, 1.7, 0.5],
                      [4.8, 3.4, 1.9, 0.2],
                      [5. , 3. , 1.6, 0.2],
                      [5. , 3.4, 1.6, 0.4],
                      [5.2, 3.5, 1.5, 0.2],
                      [5.2, 3.4, 1.4, 0.2],
                      [4.7, 3.2, 1.6, 0.2],
                      [4.8, 3.1, 1.6, 0.2],
                      [5.4, 3.4, 1.5, 0.4],
                      [5.2, 4.1, 1.5, 0.1],
                      [5.5, 4.2, 1.4, 0.2],
                      [4.9, 3.1, 1.5, 0.2],
                      [5. , 3.2, 1.2, 0.2],
                      [5.5, 3.5, 1.3, 0.2],
                      [4.9, 3.6, 1.4, 0.1],
                      [4.4, 3. , 1.3, 0.2],
                      [5.1, 3.4, 1.5, 0.2],
                      [5. , 3.5, 1.3, 0.3],
                      [4.5, 2.3, 1.3, 0.3],
                      [4.4, 3.2, 1.3, 0.2],
                      [5. , 3.5, 1.6, 0.6],
                      [5.1, 3.8, 1.9, 0.4],
                      [4.8, 3. , 1.4, 0.3],
                      [5.1, 3.8, 1.6, 0.2],
```

```
[4.6, 3.2, 1.4, 0.2],
[5.3, 3.7, 1.5, 0.2],
[5. , 3.3, 1.4, 0.2],
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 4.5, 1.5],
[6.9, 3.1, 4.9, 1.5],
[5.5, 2.3, 4. , 1.3],
[6.5, 2.8, 4.6, 1.5],
[5.7, 2.8, 4.5, 1.3],
[6.3, 3.3, 4.7, 1.6],
[4.9, 2.4, 3.3, 1. ],
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
```

```
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2. ],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
[5.7, 2.5, 5. , 2. ],
[5.8, 2.8, 5.1, 2.4],
[6.4, 3.2, 5.3, 2.3],
[6.5, 3. , 5.5, 1.8],
[7.7, 3.8, 6.7, 2.2],
[7.7, 2.6, 6.9, 2.3],
[6. , 2.2, 5. , 1.5],
[6.9, 3.2, 5.7, 2.3],
[5.6, 2.8, 4.9, 2. ],
[7.7, 2.8, 6.7, 2. ],
[6.3, 2.7, 4.9, 1.8],
[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],
[6.4, 2.8, 5.6, 2.1],
[7.2, 3. , 5.8, 1.6],
[7.4, 2.8, 6.1, 1.9],
[7.9, 3.8, 6.4, 2. ],
[6.4, 2.8, 5.6, 2.2],
[6.3, 2.8, 5.1, 1.5],
[6.1, 2.6, 5.6, 1.4],
[7.7, 3. , 6.1, 2.3],
[6.3, 3.4, 5.6, 2.4],
[6.4, 3.1, 5.5, 1.8],
[6. , 3. , 4.8, 1.8],
[6.9, 3.1, 5.4, 2.1],
[6.7, 3.1, 5.6, 2.4],
```

```
        [6.9, 3.1, 5.1, 2.3],
        [5.8, 2.7, 5.1, 1.9],
        [6.8, 3.2, 5.9, 2.3],
        [6.7, 3.3, 5.7, 2.5],
        [6.7, 3. , 5.2, 2.3],
        [6.3, 2.5, 5. , 1.9],
        [6.5, 3. , 5.2, 2. ],
        [6.2, 3.4, 5.4, 2.3],
        [5.9, 3. , 5.1, 1.8]]),
 'target': 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
 'frame': None,
 'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
 'DESCR': '.. _iris_dataset:\n\nIris plants
dataset\n--------------------\n\n**Data Set Characteristics:**\n\n    :Number of
Instances: 150 (50 in each of three classes)\n    :Number of Attributes: 4
numeric, predictive attributes and the class\n    :Attribute Information:\n
- sepal length in cm\n        - sepal width in cm\n        - petal length in
cm\n        - petal width in cm\n        - class:\n                - Iris-
Setosa\n                - Iris-Versicolour\n                - Iris-Virginica\n
\n    :Summary Statistics:\n\n    ============== ==== ==== ======= =====
====================\n                    Min  Max   Mean    SD   Class
Correlation\n    ============== ==== ==== ======= ===== ====================\n
sepal length:   4.3  7.9   5.84   0.83    0.7826\n    sepal width:    2.0  4.4
3.05   0.43   -0.4194\n    petal length:   1.0  6.9   3.76   1.76    0.9490
(high!)\n    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)\n
============== ==== ==== ======= ===== ====================\n\n    :Missing
Attribute Values: None\n    :Class Distribution: 33.3% for each of 3 classes.\n
:Creator: R.A. Fisher\n    :Donor: Michael Marshall
(MARSHALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThe famous Iris
database, first used by Sir R.A. Fisher. The dataset is taken\nfrom Fisher\'s
paper. Note that it\'s the same as in R, but not as in the UCI\nMachine Learning
Repository, which has two wrong data points.\n\nThis is perhaps the best known
database to be found in the\npattern recognition literature.  Fisher\'s paper is
a classic in the field and\nis referenced frequently to this day.  (See Duda &
Hart, for example.)  The\ndata set contains 3 classes of 50 instances each,
where each class refers to a\ntype of iris plant.  One class is linearly
separable from the other 2; the\nlatter are NOT linearly separable from each
other.\n\n.. topic:: References\n\n   - Fisher, R.A. "The use of multiple
measurements in taxonomic problems"\n     Annual Eugenics, 7, Part II, 179-188
(1936); also in "Contributions to\n     Mathematical Statistics" (John Wiley,
```

```
NY, 1950).\n   - Duda, R.O., & Hart, P.E. (1973) Pattern Classification and
Scene Analysis.\n     (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See
page 218.\n   - Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New
System\n     Structure and Classification Rule for Recognition in Partially
Exposed\n     Environments".  IEEE Transactions on Pattern Analysis and
Machine\n     Intelligence, Vol. PAMI-2, No. 1, 67-71.\n   - Gates, G.W. (1972)
"The Reduced Nearest Neighbor Rule".  IEEE Transactions\n     on Information
Theory, May 1972, 431-433.\n   - See also: 1988 MLC Proceedings, 54-64.
Cheeseman et al"s AUTOCLASS II\n     conceptual clustering system finds 3
classes in the data.\n   - Many, many more …',
 'feature_names': ['sepal length (cm)',
  'sepal width (cm)',
  'petal length (cm)',
  'petal width (cm)'],
 'filename': 'iris.csv',
 'data_module': 'sklearn.datasets.data'}
```

[26]:
```
iris = load_iris()
iris_df = pd.DataFrame(data = iris.data, columns = iris.feature_names)
iris_df
```

[26]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
..	…	…	…	…
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

[150 rows x 4 columns]

[27]:
```
iris = load_iris()
iris_df = pd.DataFrame(data = iris.data, columns = iris.feature_names)
iris_df['target'] = iris.target
iris_df['target_names'] = iris.target_names[iris.target]
iris_df
```

[27]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	

```
4                  5.0               3.6                1.4               0.2
..                 …                 …                  …                 …
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

     target target_names
0         0       setosa
1         0       setosa
2         0       setosa
3         0       setosa
4         0       setosa
..      …           …
145       2    virginica
146       2    virginica
147       2    virginica
148       2    virginica
149       2    virginica

[150 rows x 6 columns]
```

[28]:
```
x = iris.data
x
```

[28]:
```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
       [5.4, 3.4, 1.7, 0.2],
```

```
[5.1, 3.7, 1.5, 0.4],
[4.6, 3.6, 1. , 0.2],
[5.1, 3.3, 1.7, 0.5],
[4.8, 3.4, 1.9, 0.2],
[5. , 3. , 1.6, 0.2],
[5. , 3.4, 1.6, 0.4],
[5.2, 3.5, 1.5, 0.2],
[5.2, 3.4, 1.4, 0.2],
[4.7, 3.2, 1.6, 0.2],
[4.8, 3.1, 1.6, 0.2],
[5.4, 3.4, 1.5, 0.4],
[5.2, 4.1, 1.5, 0.1],
[5.5, 4.2, 1.4, 0.2],
[4.9, 3.1, 1.5, 0.2],
[5. , 3.2, 1.2, 0.2],
[5.5, 3.5, 1.3, 0.2],
[4.9, 3.6, 1.4, 0.1],
[4.4, 3. , 1.3, 0.2],
[5.1, 3.4, 1.5, 0.2],
[5. , 3.5, 1.3, 0.3],
[4.5, 2.3, 1.3, 0.3],
[4.4, 3.2, 1.3, 0.2],
[5. , 3.5, 1.6, 0.6],
[5.1, 3.8, 1.9, 0.4],
[4.8, 3. , 1.4, 0.3],
[5.1, 3.8, 1.6, 0.2],
[4.6, 3.2, 1.4, 0.2],
[5.3, 3.7, 1.5, 0.2],
[5. , 3.3, 1.4, 0.2],
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 4.5, 1.5],
[6.9, 3.1, 4.9, 1.5],
[5.5, 2.3, 4. , 1.3],
[6.5, 2.8, 4.6, 1.5],
[5.7, 2.8, 4.5, 1.3],
[6.3, 3.3, 4.7, 1.6],
[4.9, 2.4, 3.3, 1. ],
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1. ],
```

```
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2. ],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
[5.7, 2.5, 5. , 2. ],
[5.8, 2.8, 5.1, 2.4],
```

```
       [6.4, 3.2, 5.3, 2.3],
       [6.5, 3. , 5.5, 1.8],
       [7.7, 3.8, 6.7, 2.2],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.2, 5. , 1.5],
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]])
```

[29]:
```
y = iris.target
y
```

[29]:
```
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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

[30]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.
↪2,random_state=101)
x_train.shape
```

[30]:
```
(120, 4)
```

[31]:
```
y_test.shape
```

[31]:
```
(30,)
```

[32]:
```
# To train the algorithm
clf = LogisticRegression()
```

[33]:
```
clf.fit(x_train,y_train)
```

[33]:
```
LogisticRegression()
```

[34]:
```
y_pred = clf.predict(x_test)
y_pred
```

[34]:
```
array([0, 0, 0, 2, 1, 2, 1, 1, 2, 0, 2, 0, 0, 2, 2, 1, 1, 1, 0, 2, 1, 0,
       1, 1, 1, 1, 1, 2, 0, 0])
```

[35]:
```
accuracy = accuracy_score(y_test,y_pred)
accuracy
```

[35]:
```
1.0
```

[36]:
```
data = pd.read_csv('bmi.csv')
data
```

[36]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
..	…	…	…	…
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

[500 rows x 4 columns]

[37]:
```
# project - 1
print('Male',(data['Gender'] == 'Male').sum())
print('Female',(data['Gender'] == 'Female').sum())
```

```
Male 245
Female 255
```

[38]:
```
data['Gender']=data['Gender'].map({'Male':0,'Female':1})
data
```

[38]:

	Gender	Height	Weight	Index
0	0	174	96	4
1	0	189	87	2
2	1	185	110	4
3	1	195	104	3
4	0	149	61	3
..	…	…	…	…
495	1	150	153	5
496	1	184	121	4
497	1	141	136	5
498	0	150	95	5
499	0	173	131	5

[500 rows x 4 columns]

[39]:
```
data = pd.get_dummies(data,columns=['Gender'],dtype=int)
data
```

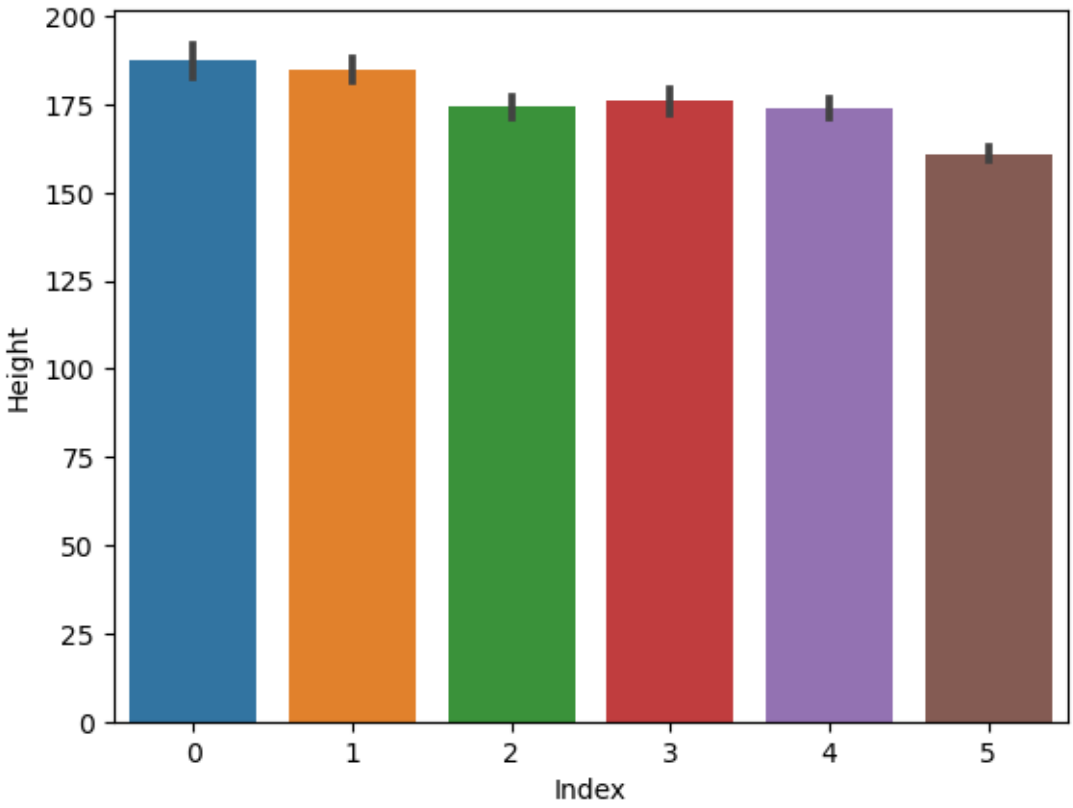
[39]:

	Height	Weight	Index	Gender_0	Gender_1
0	174	96	4	1	0
1	189	87	2	1	0
2	185	110	4	0	1
3	195	104	3	0	1
4	149	61	3	1	0
..	…	…	…	…	…
495	150	153	5	0	1
496	184	121	4	0	1
497	141	136	5	0	1
498	150	95	5	1	0
499	173	131	5	1	0

[500 rows x 5 columns]

[60]:
```
sns.barplot(x='Index',y='Height',data=data)
```

[60]:
```
<Axes: xlabel='Index', ylabel='Height'>
```


200
175
150
125
100
75
50
25
0
Height
0
1
2
3
4
5
Index