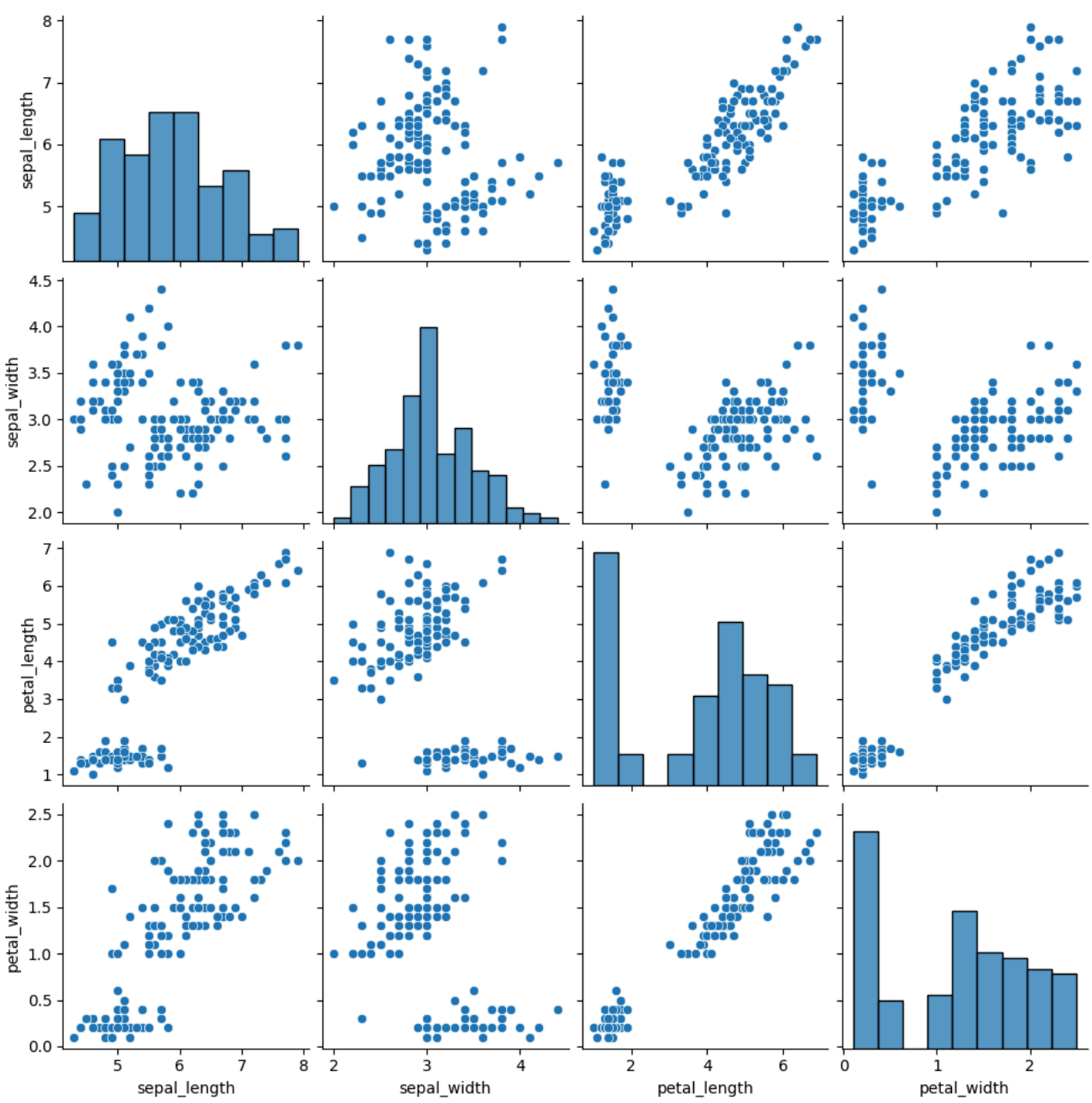
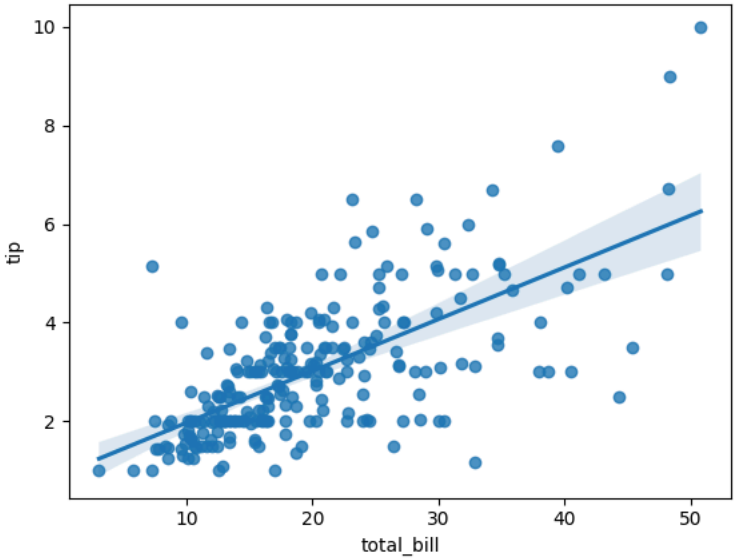
```
import matplotlib.pyplot as plt
import seaborn as sns
df=sns.load_dataset('iris')
sns.pairplot(df,kind="scatter")
plt.show()
```




```
import seaborn as sb
from matplotlib import pyplot as pyplot
df=sb.load_dataset('tips')
sb.regplot(x="total_bill",y="tip",data=df)
plt.show()
```




```
import matplotlib.pyplot as plt
from scipy import  stats
```

```
x=[5,7,8,7,2,17,2,9,4,11,12,9,6]
y=[99,86,87,88,111,86,103,87,94,78,77,85,86]
```

```
slope,intercept,r,p,std_err=stats.linregress(x,y)
```

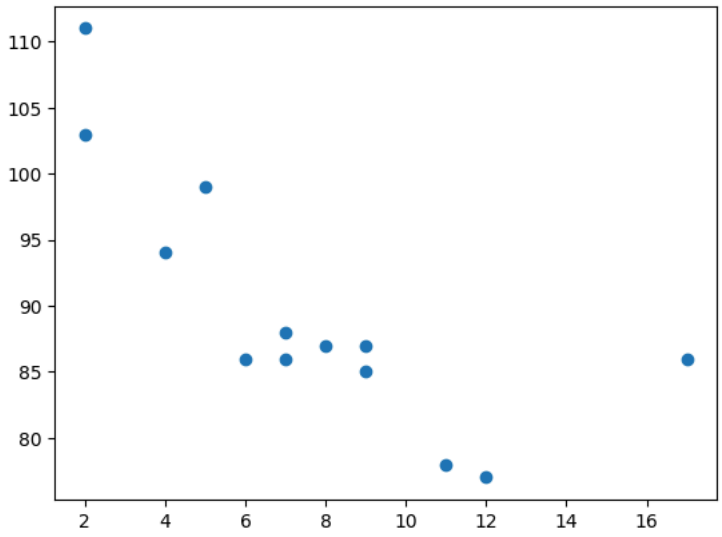
```
def myfunc(x):
  return slope*x+intercept
```

```
mymodel=list(map(myfunc,x))
```

Draw the original scatter plot:

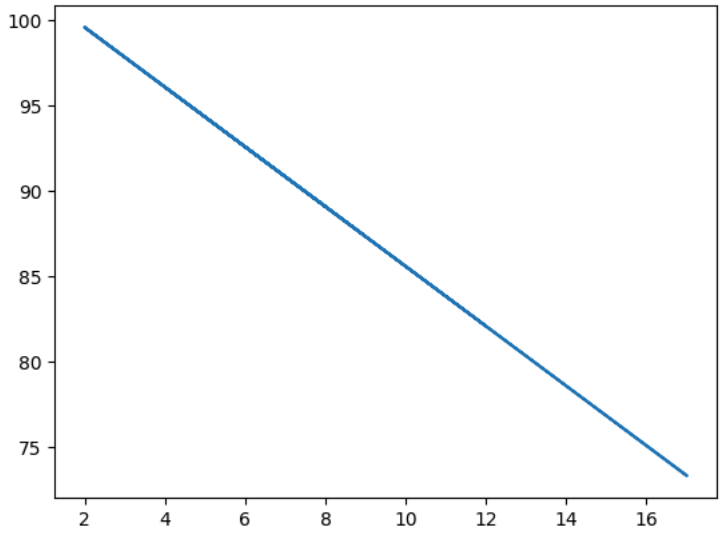
```
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x7a96795f5d80>
```




```
plt.plot(x,mymodel)
```

```
[<matplotlib.lines.Line2D at 0x7a9671b4ff40>]
```




```
from re import M
import numpy as nmp
import matplotlib.pyplot as mtplt
def estimate_coeff(p,q):
  n1=nmp.size(p)
  m_p=nmp.mean(p)
  m_q=nmp.mean(q)
  SS_pq=nmp.sum(q*p)-n1*m_q*m_p
  SS_pp=nmp.sum(p*p)-n1*m_p*m_p

  b_1=SS_pq / SS_pp
  b_0=m_q-b_1*m_p
  return(b_0,b_1)
  def plot_regression_line(p,q,b):
    mtplt.scatter(p,q,color="m",marker="o",s=30)
    q_pred=b[0]+b[1]*p
    mtplt.plot(p,q_qred,color="g")
    mtplt.xlabel('p')
    mtplt.ylabel('q')
    mtplt.show()
  def main():
    p=np.array([10,11,12,13,14,15,16,17,18,19])
    q=np.array([11,13,12,15,17,18,18,19,20,22])
    b=estimate_coeff(p,q)
    print("Estimated coefficients are:\nb 0={} \ \nb_1={}".format(b[0],b[1]))
    plot_regression_line(p,q,b)


  if __name__=="__main__":
```