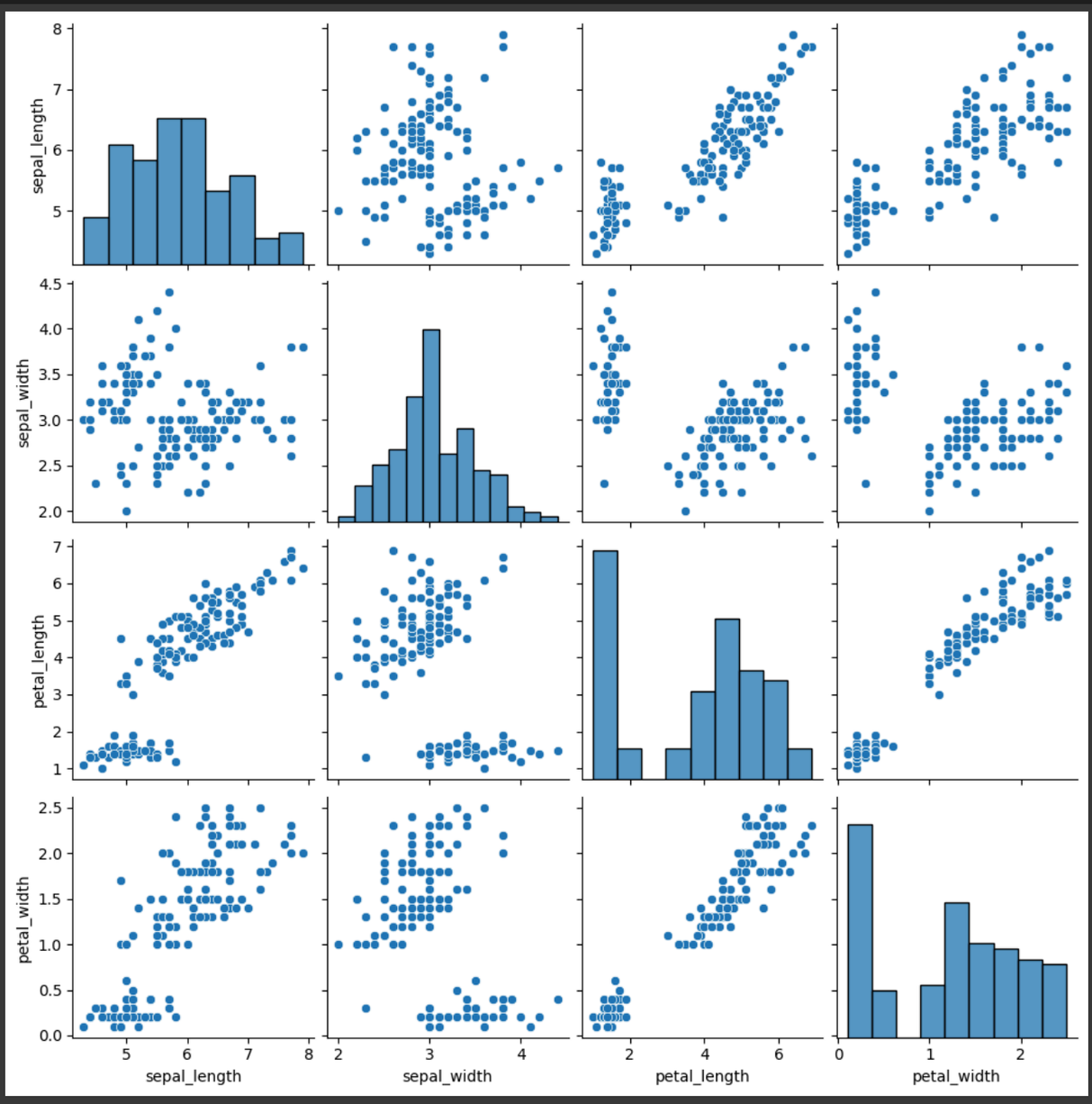
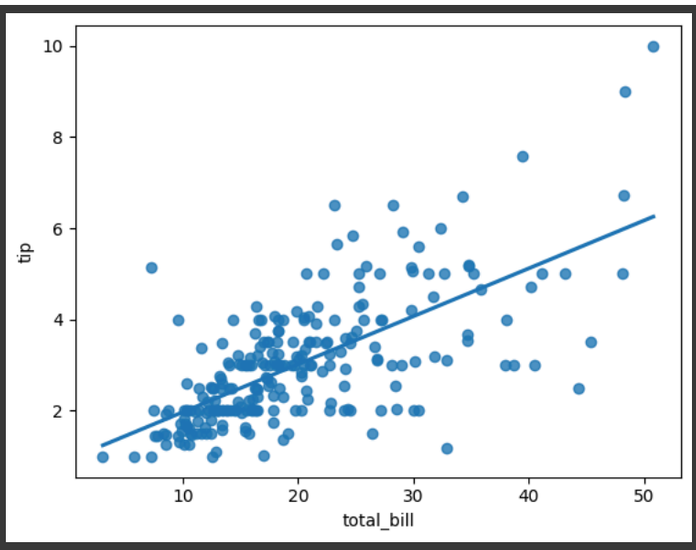
```
import matplotlib.pyplot as plt
import seaborn as sns
df = sns.load_dataset('iris')
sns.pairplot(df, kind="scatter")
plt.show()
df.head()
```

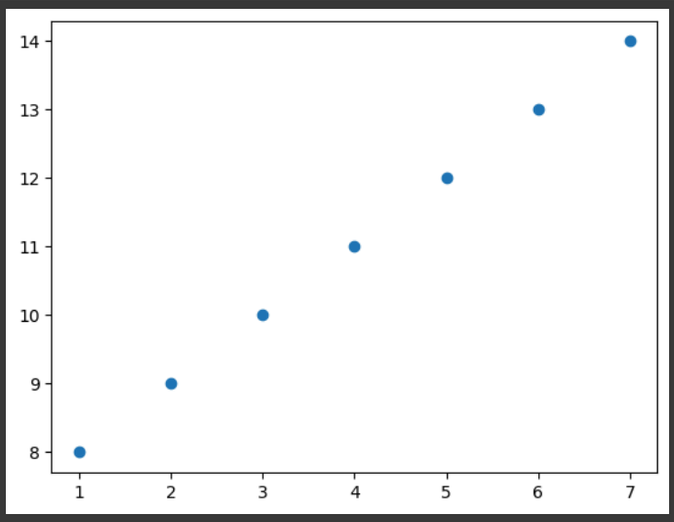



```
df=sns.load_dataset('tips')
sns.regplot(x="total_bill", y="tip", data=df, ci=None)
plt.show() #plotting of scatter plot wit refence to the total tips on x-axis and y value of bill
df.head()
```



	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

```
from scipy import stats
x=[1,2,3,4,5,6,7]
y=[8,9,10,11,12,13,14]
slope, intercept, r, p, std_error=stats.linregress(x,y)
def myfunc(x):
  return slope*x+intercept # sample return of value in the included function
mymodel=list(map(myfunc,x))
plt.scatter(x, mymodel)
plt.show()
```




```
import numpy as np
import matplotlib.pyplot as plt
def esti_coeff(p,q):
  n1=np.size(p) #checking the size of p array values
  np1=np.mean(p) # calculation of mean values of p and q
  nq=np.mean(q) # calculate mean of q
  sspq=np.sum(q*p)-n1*nq*np1 # crosss deviation # is in the form y=mx+c
  sspp=np.sum(p*p)-n1*nq*np1 # deviation p
  b1=sspq/sspp #computation of slope m values by dividing cross dev. value and dev(p^2) value
  b0=np1-b1*nq #is in the form of y=mx+c
  return (b0, b1) # returning the values of sloppe m and intercept c
def plot_regress(p,q,b):
  plt.scatter(p,q, color="m", marker="o", s=30)
  qpred=b[0]+b[1]*p  #b[0] is slope(m)  b[1] is intercept(c) for y=mx+c
  plt.plot(p,q, qpred, color="g")
  plt.xlabel("p")
  plt.ylabel("q")
  plt.show()
```

```
def main():
  p=np.array([10,11,12,13,14,15,16,17,18,19]) #p, q array ensuring for the same size
  q=np.array([11,13,12,15,12,17,18,18,19,20])
  b=esti_coeff(p,q)
  print("Estimated coefficients are: \nb0 = {} \ \nb1={}".format(b[0], b[1]))
  plot_regress(p,q,b)

if __name__ == "__main__":
  main()
```

Estimated coefficients are:
b0 = 35.208 \
b1=-1.336