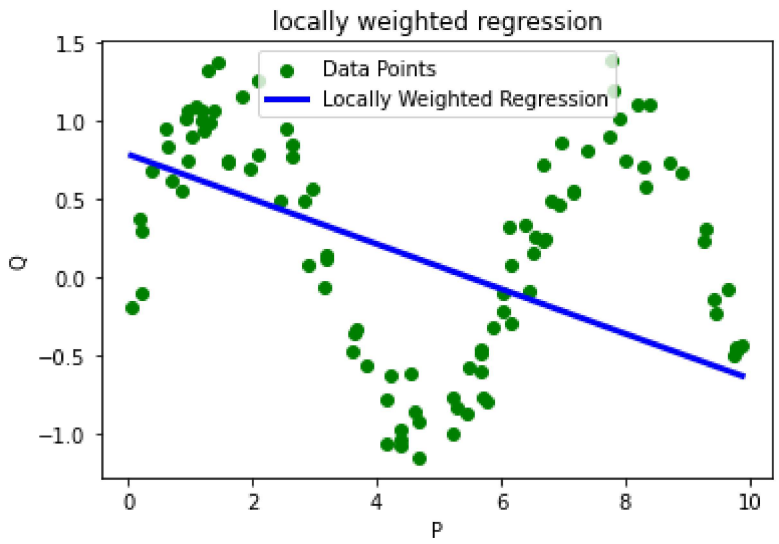
In [12]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

In [32]:
```
def local_regression(w,P,Q,alpha):
    """perform local regression at point w"""
    weights = np.exp( - (P-w)*2 / (2*alpha*2)).ravel()
    #fit a weighted linearmodel
    model = LinearRegression()
    model.fit(P,Q,sample_weight= weights)
    #predict values of w
    return model.predict(np.array([[w]]))
```

In [33]:
```
#sample data
np.random.seed(0)
P = np.sort(np.random.rand(100)*10).reshape(-1,1)
Q = np.sin(P).ravel() + np.random.normal(0,0.2,100)
```

In [34]:
```
#fit local regression  for ech point in P
alpha = 1.0 #bandwidthprameter
Q_pred = np.array([local_regression(w,P,Q,alpha) for w in P.ravel()])
```

In [36]:
```
#PLOT THE RESULTS
plt.scatter(P,Q,label ="Data Points",color="green")
plt.plot(np.sort(P.ravel()),Q_pred[np.argsort(P.ravel())], label = "Locally Weighte
plt.title("locally weighted regression")
plt.xlabel("P")
plt.ylabel("Q")
plt.legend()
plt.show()
```




In [ ]:
```
```