In [3]:
```
diabetes
```

Out[3]:
```
{'data': array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
          0.01990749, -0.01764613],
        [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
         -0.06833155, -0.09220405],
        [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
          0.00286131, -0.02593034],
        ...,
        [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
         -0.04688253,  0.01549073],
        [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
          0.04452873, -0.02593034],
        [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
         -0.00422151,  0.00306441]]),
 'target': array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 3
10., 101.,
         69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
         68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
         87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
        259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
```

In [1]:
```
from sklearn import datasets
```

In [2]:
```
diabetes = datasets.load_diabetes()
```

In [4]:
```
print(diabetes.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values

  :Target: Column 11 is a quantitative measure of disease progression one
year after baseline

  :Attribute Information:
      - age     age in years
      - sex
      - bmi     body mass index
      - bp      average blood pressure
      - s1      tc, total serum cholesterol
      - s2      ldl, low-density lipoproteins
      - s3      hdl, high-density lipoproteins
      - s4      tch, total cholesterol / HDL
      - s5      ltg, possibly log of serum triglycerides level
      - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scal
ed by the standard deviation times the square root of `n_samples` (i.e. t
he sum of squares of each column totals 1).

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html (https://www4.s
tat.ncsu.edu/~boos/var.select/diabetes.html)

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004)
"Least Angle Regression," Annals of Statistics (with discussion), 407-49
9.
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```

In [6]:
```
print(diabetes.feature_names)
#print features name once again
```

```
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

In [7]:
```
X = diabetes.data
Y = diabetes.target
```

In [8]:
```
X.shape, Y.shape
```

Out[8]:
```
((442, 10), (442,))
```

In [9]:
```
from sklearn.model_selection import train_test_split
#import necessary library for data split
```

In [10]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
#data split 20% goes to the test set
```

In [11]:
```
X_train.shape, Y_train.shape
#data dimension i.e 80% training data
```

Out[11]:
```
((353, 10), (353,))
```

In [13]:
```
from sklearn import linear_model
#import library for model
from sklearn.metrics import mean_squared_error, r2_score
#import library for computing model
```

In [15]:
```
model = linear_model.LinearRegression()
#Defines the regression model
```

In [17]:
```
model.fit(X_train, Y_train)
# Build actual training model
```

Out[17]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]:
```
Y_pred = model.predict(X_test)
# Apply trained model to make prediction (on test set)
```

In [21]:
```
# **Print model performance**
print('Coefficients:', model.coef_)
#coeffiecient is stored in model.coef
print('Intercept:', model.intercept_)
#intercept stored in model.intercept
print('Mean squared error (MSE): %.2f'
      % mean_squared_error(Y_test, Y_pred))
#mean square stored in mean_squared_error, parameter passed
#Y_test is the actual value, Y_pred is the predicted value.
print('Coefficient of determination (R^2): %.2f'
     % r2_score(Y_test, Y_pred))
```

```
Coefficients: [ -14.26812489 -294.66323133  556.98765558  284.78133515 -9
73.00174526
  591.72330419  240.24164297  310.43219074  766.60619727   81.88776627]
Intercept: 152.50905379010226
Mean squared error (MSE): 2829.91
Coefficient of determination (R^2): 0.49
```

In [22]:
```
#equation of linear regression model is y=-14.26812489(age) -294.66323133 (
## **String formatting**
r2_score(Y_test, Y_pred)
```

Out[22]:
```
0.49021677588018786
```

In [23]:
```
r2_score(Y_test, Y_pred).dtype
```

Out[23]:
```
dtype('float64')
```

In [24]:
```
'%.2f' %0.49021677588018786
```

Out[24]:
```
'0.49'
```

In [25]:
```
#Now make scatterplot
# **Import library**
import seaborn as sns
```

In [26]:
```
Y_test
#look at data
```

Out[26]:
```
array([202., 220., 199., 121., 109., 249., 252., 142., 252.,  52., 179.,
       272., 308.,  48., 163., 268., 219.,  83., 131.,  70., 217., 206.,
        49.,  89.,  57.,  90., 104., 142., 258., 150.,  91., 181., 310.,
       174.,  78., 171.,  71.,  75., 185., 142., 225., 200., 152., 212.,
       101., 296.,  87.,  60., 236.,  37., 102., 263.,  44., 111., 202.,
        53., 131., 121., 183., 189., 166., 168., 137., 197., 297., 244.,
       132., 128., 122.,  74., 232., 116.,  84.,  42.,  97., 144.,  53.,
        70.,  45.,  59.,  40., 265.,  65.,  72., 198., 206., 196., 115.,
       186.])
```

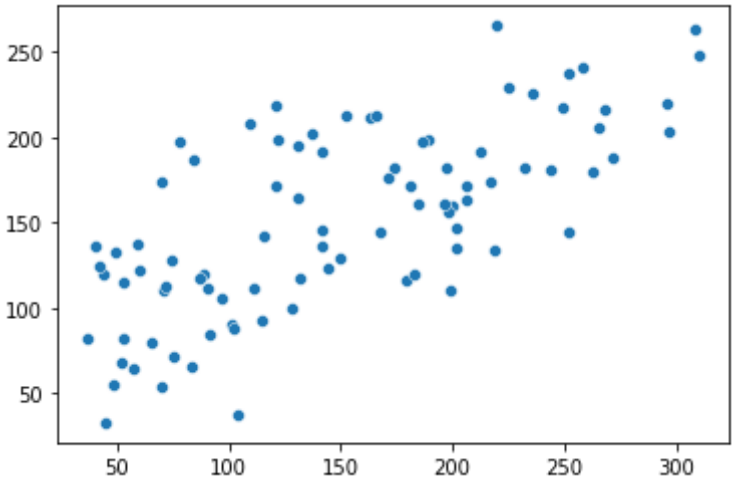
In [27]:
```
Y_pred
```

Out[27]:
```
array([135.46158784, 265.41399678, 110.37504865, 171.90965885,
       208.05689615, 217.01382941, 145.06972288, 136.34426482,
       237.98372468,  68.3983279 , 116.68512593, 187.88530374,
       263.56553173,  55.15170817, 211.44458988, 216.47475712,
       134.06499871,  65.55012555, 164.50100619, 174.43876211,
       173.35929596, 171.58167701, 132.5712416 , 120.15964196,
        63.93857131, 111.54262507,  36.83380647, 191.09431101,
       241.10242247, 129.08523114,  84.23977999, 171.90441026,
       247.996688  , 182.31523389, 198.01618097, 176.21876071,
       110.53464054,  72.08634669, 160.4785526 , 145.5239425 ,
       229.2654962 , 160.22285002, 212.8514772 , 191.5221397 ,
        89.96589371, 219.79560145, 117.4723048 , 122.71198415,
       226.2219284 ,  82.17995131,  88.48149813, 179.61498644,
       120.15456018, 111.05451485, 146.31540844,  82.56015461,
       195.63156153, 218.99097859, 120.07273776, 199.164119  ,
       213.32176127, 144.2632872 , 202.16360781, 182.29432575,
       203.18881599, 180.72013835, 117.07614913, 100.2325263 ,
       199.0939836 , 127.87902882, 182.43881278, 142.324929  ,
       186.35730407, 124.36997395, 105.73926758, 123.46424494,
       115.2077598 ,  54.34426743,  32.79476599, 137.94889864,
       135.78005204, 205.14189993,  79.90002847, 112.95278348,
       156.00285675, 163.6377009 , 161.3666516 ,  92.36343668,
       197.01004328])
```

In [28]:
```
sns.scatterplot(Y_test, Y_pred)
```

```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-pack
ages/seaborn/_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

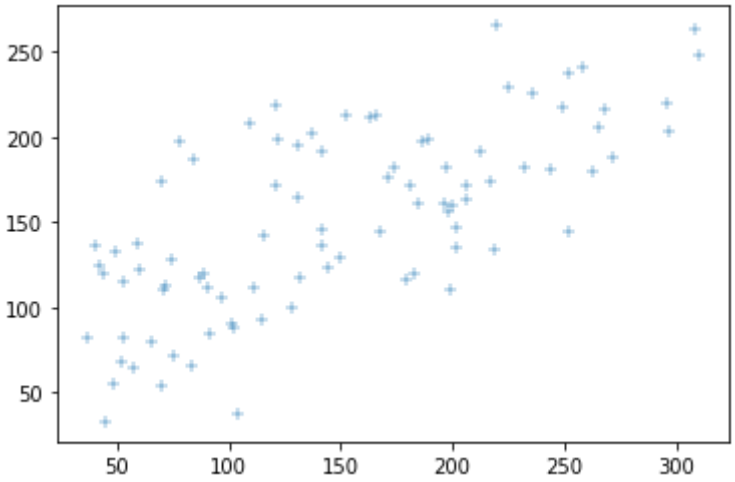
Out[28]:
```
<AxesSubplot:>
```



In [30]:
```
sns.scatterplot(Y_test, Y_pred, marker="+")
```

```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

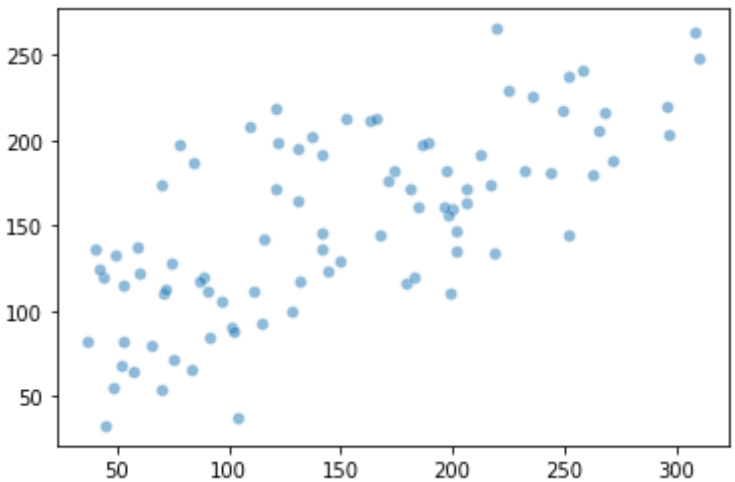
Out[30]: <AxesSubplot:>



In [34]:
```
sns.scatterplot(Y_test, Y_pred, alpha=0.5)
```

```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[34]: <AxesSubplot:>



In [ ]: