Correlation

What is Correlation?

Correlation is a bivariate analysis that measures the strength of association between two variables and the direction of the relationship.

Linear correlation is a measure of dependence between two random variables

The statistical relationship between two variables is referred to as their correlation

A correlation could be : Positive : both variables change in the same direction. Neutral : No relationship in the change of the variables. Negative : variables change in opposite directions.

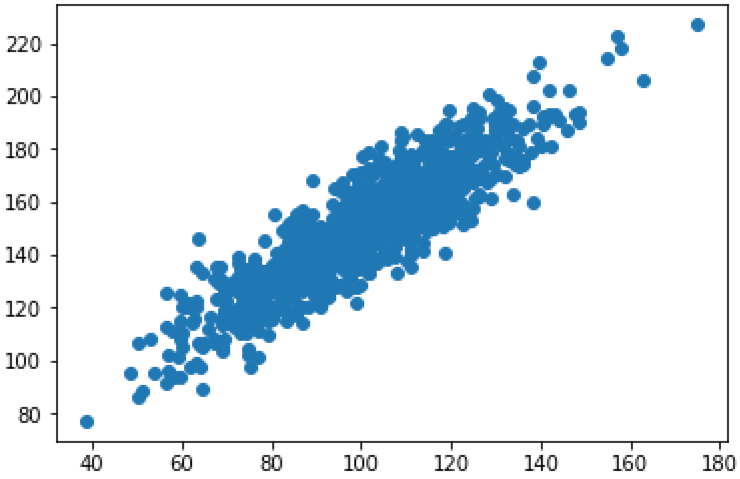
In [5]:

```
# generate related variables
from numpy import mean
from numpy import std
from numpy import cov
from numpy.random import randn
from numpy.random import seed
from matplotlib import pyplot
# seed random number generator
seed(1)
```

Dataset creation

In [13]:
```
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# summarize
print('data1: mean=%.3f stdv=%.3f' % (mean(data1), std(data1)))
print('data2: mean=%.3f stdv=%.3f' % (mean(data2), std(data2)))
# plot
pyplot.scatter(data1, data2)
pyplot.show()
```

```
data1: mean=99.557 stdv=19.432
data2: mean=149.593 stdv=22.175
```



Covariancce

This relationship can be summarized between two variables, called the covariance.

It is calculated as the average of the product between the values from each sample, where the values haven been centered (had their mean subtracted).

The calculation of the sample covariance is as follows:

cov(X, Y) = (sum (x - mean(X)) *(y - mean(Y)))* 1/(n-1)

In [14]:
```
covariance = cov(data1, data2)
print(covariance)
```

```
[[377.99860125 384.19134307]
 [384.19134307 492.24243533]]
```

Problems with covariance:

A problem with covariance as a statistical tool alone is that it is challenging to interpret.

This leads us to the Pearson’s correlation coefficient next.

Pearson (Karl Pearson) correlation coefficient:

For can be used to summarize the strength of the linear relationship between two data samples.

If the two variables are normally distributed, the standard measure of determining the correlation coefficient is the Pearson.

The Pearson’s correlation coefficient is calculated as the covariance of the two variables divided by the product of the standard deviation of each data sample.

It is the normalization of the covariance between the two variables to give an interpretable score.

Pearson's correlation coefficient = covariance(X, Y) / (stdv(X) * stdv(Y))

```
In [12]: # # calculate the Pearson's correlation between two variables
         from numpy.random import randn
         from numpy.random import seed
         from scipy.stats import pearsonr
         # seed random number generator
         seed(1)
         # prepare data
         data1 = 20 * randn(1000) + 100
         data2 = data1 + (10 * randn(1000) + 50)
         # calculate Pearson's correlation
         corr, _ = pearsonr(data1, data2)
         print('Pearsons correlation: %.3f' % corr)
```

```
Pearsons correlation: 0.888
```

The coefficient returns a value between -1 and 1 that represents the limits of correlation from a full negative correlation to a full positive correlation.

A value of 0 means no correlation.

A value below -0.5 or above 0.5 indicates a notable correlation,

Values below those values suggests a less notable correlation.

Problems with Pearson Correlation:

Correlations are very sensitive to outliers; A single unusual observation may have a huge impact on a correlation.

If the data distribution is not normal, a different approach is necessary.

In that case one can rank the set of data for each variable and compare the orderings.

There are two commonly used methods of calculating the rank correlation.

```
Spearman’s
Kendall’s
```

Two variables may be related by a nonlinear relationship, such that the relationship is stronger or weaker across the distribution of the variables.

Spearman’s Correlation

If you are unsure of the distribution and possible relationships between two variables, Spearman correlation coefficient is a good tool to use.

These statistics are calculated from the relative rank of values on each sample.

This is a common approach used in non-parametric statistics, e.g. statistical methods where we do not assume a distribution of the data such as Gaussian.

Spearman's correlation coefficient = covariance(rank(X), rank(Y)) / (stdv(rank(X)) * stdv(rank(Y)))

The spearmanr() SciPy function can be used to calculate the Spearman’s correlation coefficient between two data samples with the same length.

In [16]:

```
# calculate the spearmans's correlation between two variables
from numpy.random import randn
from numpy.random import seed
from scipy.stats import spearmanr
# seed random number generator
seed(1)
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# calculate spearman's correlation
corr, _ = spearmanr(data1, data2)
print('Spearmans correlation: %.3f' % corr)
```

```
Spearmans correlation: 0.872
```

Kendall’s :

```
It is also a rank correlation coefficient,

measuring the association between two measured quantities.

It is harder to calculate than Spearman’s, but it has been argued that confidence in
tervals for Spearman’s are less reliable and less interpretable than confidence inte
rvals for Kendall’s parameters.
```

In [20]:

```python
from numpy.random import randn
from numpy.random import seed
from scipy.stats import kendalltau
# seed random number generator
seed(1)
# prepare data
data1 = 20 * randn(1000) + 100
data2 = data1 + (10 * randn(1000) + 50)
# calculate kendal's correlation
corr, p = kendalltau(data1, data2)
print('Kendals correlation: %.3f' % corr)
```

```
Kendals correlation: 0.688
```