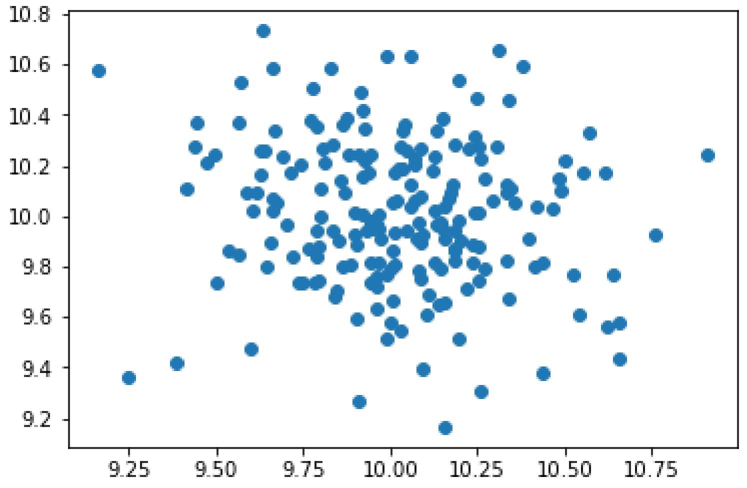
Anomaly Detection_Local Outlier Factor (LOF)

Density-based: The Local Outlier Factor is an algorithm to detect anomalies in observation data. Measuring the local density score of each sample and weighting their scores are the main concept of the algorithm. By comparing the score of the sample to its neighbors, the algorithm defines the lower density elements as anomalies in data.

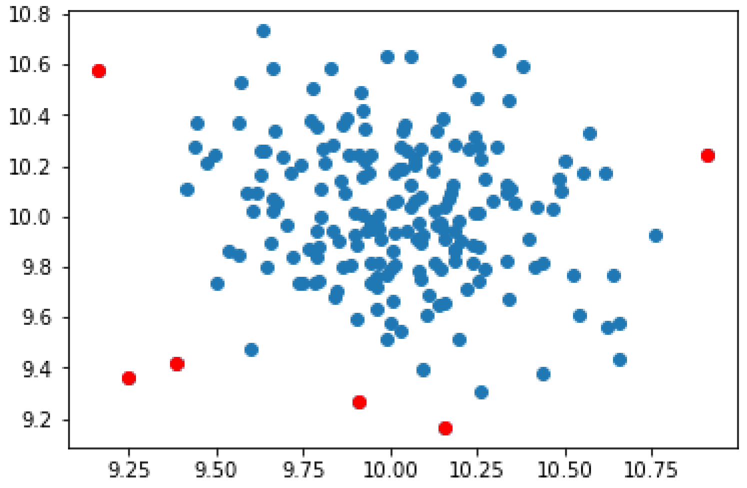
"The local outlier factor is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These are considered to be outliers." ~ Wikipedia

Distance-based: Distance-based outlier detection method consults the neighbourhood of an object, which is defined by a given radius. An object is then considered an outlier if its neighborhood does not have enough other points.

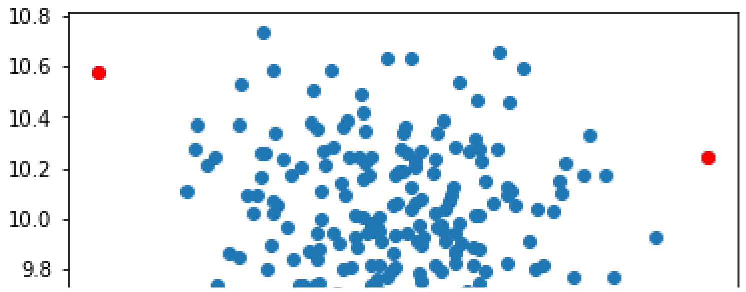
In [1]:
```
from sklearn.neighbors import LocalOutlierFactor
from sklearn.datasets import make_blobs
from numpy import quantile, where, random
import matplotlib.pyplot as plt

random.seed(1)
x, _ = make_blobs(n_samples=200, centers=1, cluster_std=.3, center_box=(10,10))

plt.scatter(x[:,0], x[:,1])
plt.show()

lof = LocalOutlierFactor(n_neighbors=20, contamination=.03)
print(lof)

y_pred = lof.fit_predict(x)

lofs_index=where(y_pred==-1)
values = x[lofs_index]

plt.scatter(x[:,0], x[:,1])
plt.scatter(values[:,0],values[:,1], color='r')
plt.show()

model = LocalOutlierFactor(n_neighbors=20)
print(model)
model.fit_predict(x)

lof = model.negative_outlier_factor_
thresh = quantile(lof, .03)
print(thresh)

index = where(lof<=thresh)
values = x[index]

plt.scatter(x[:,0], x[:,1])
plt.scatter(values[:,0],values[:,1], color='r')
plt.show()
```



```
LocalOutlierFactor(contamination=0.03)
```



```
LocalOutlierFactor()
-1.819148296090704
```



In [2]:
```
import pandas as pd
import numpy as np
from sklearn.neighbors import LocalOutlierFactor
from sklearn.datasets import make_blobs
from numpy import quantile, where, random
import matplotlib.pyplot as plt
```

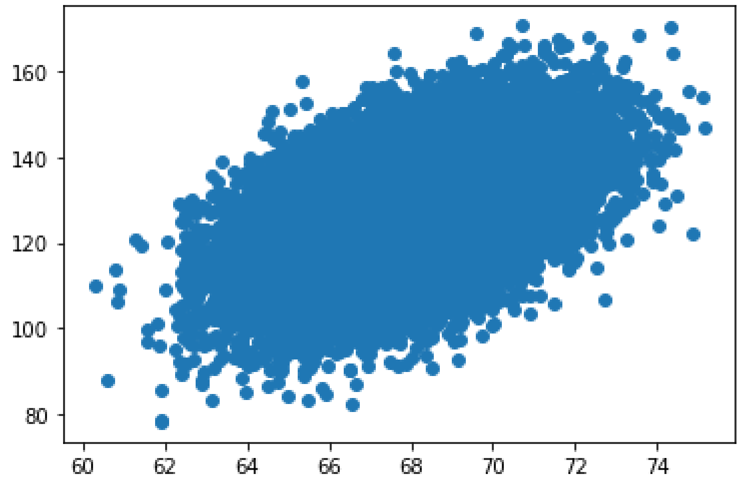
In [3]:
```
x=pd.read_csv('SOCR-HeightWeight.csv')
print(x)
```

```
       Height  Weight
0       61.93   78.01
1       61.91   78.57
2       66.57   82.38
3       63.13   83.09
4       65.47   83.34
...       ...     ...
24995   72.32  168.23
24996   73.52  168.88
24997   69.57  169.13
24998   74.30  170.55
24999   70.71  170.92

[25000 rows x 2 columns]
```

In [4]:
```
x=x[['Height','Weight']]
```

In [5]:
```
x= x.values
```

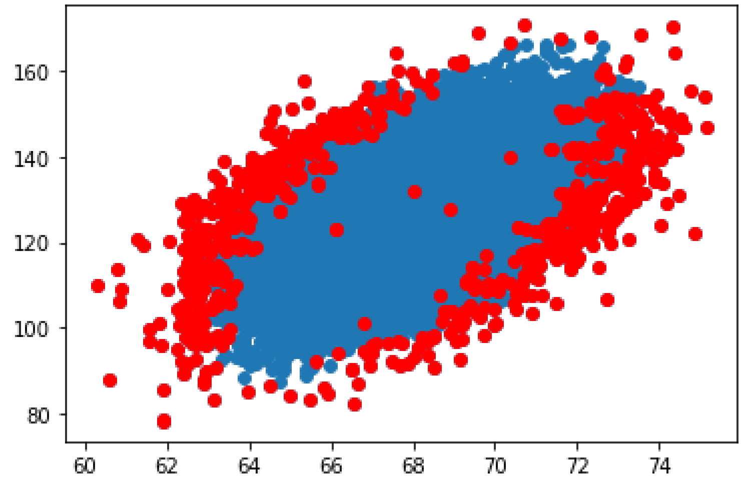
In [6]:
```
plt.scatter(x[:,0], x[:,1])
plt.show()
```



In [7]:
```
lof = LocalOutlierFactor(n_neighbors=20, contamination=.03)
print(lof)

y_pred = lof.fit_predict(x)
```

```
LocalOutlierFactor(contamination=0.03)
```

In [8]:
```
lofs_index=where(y_pred==-1)
values = x[lofs_index]

plt.scatter(x[:,0], x[:,1])
plt.scatter(values[:,0],values[:,1], color='r')
plt.show()
```



In [9]:
```
model = LocalOutlierFactor(n_neighbors=20)
print(model)
model.fit_predict(x)
```

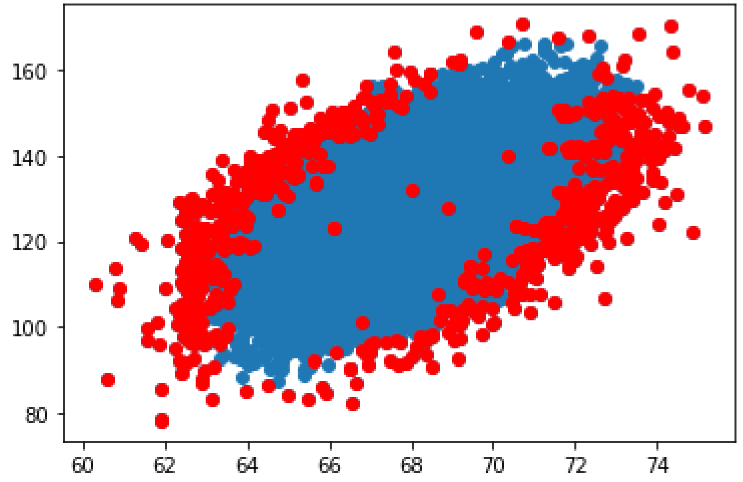
```
LocalOutlierFactor()
```

Out[9]:
```
array([-1, -1, -1, ...,  1, -1, -1])
```

In [10]:
```
lof = model.negative_outlier_factor_
thresh = quantile(lof, .03)
print(thresh)
```

```
-1.1509501957405464
```

In [11]:
```
index = where(lof<=thresh)
values = x[index]

plt.scatter(x[:,0], x[:,1])
plt.scatter(values[:,0],values[:,1], color='r')
plt.show()
```



In [ ]: