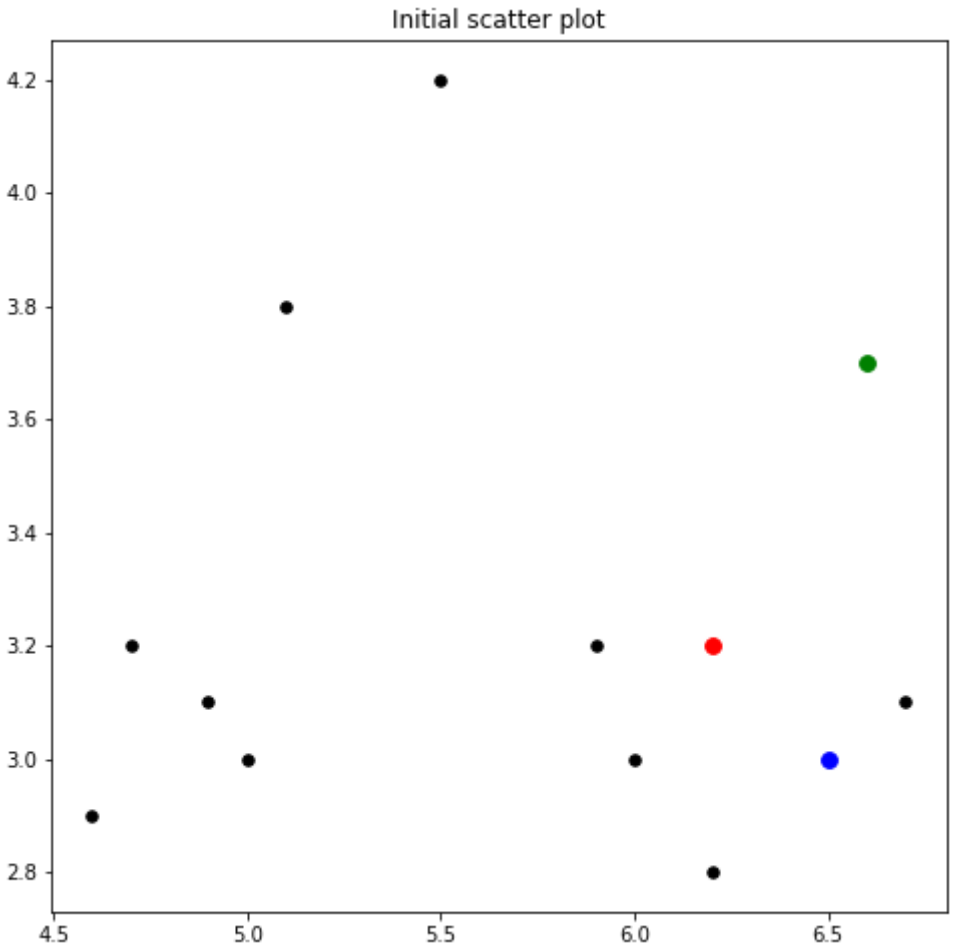
In [1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
X = np.array([[5.9, 3.2], [4.6, 2.9], [6.2, 2.8], [4.7, 3.2], [5.5, 4.2], [
              [4.9, 3.1], [6.7, 3.1], [5.1, 3.8], [6.0, 3.0]])

c1=[6.2,3.2]
c2=[6.6,3.7]
c3=[6.5,3.0]
```

In [3]:
```
# Initial Scatter plot
fig = plt.figure(figsize=(8, 8))
for i in X:
    plt.scatter(i[0],i[1],c = "black",s=30)
plt.title('Initial scatter plot')
plt.scatter(c1[0],c1[1], c='red',s=60)
plt.scatter(c2[0],c2[1],c='green',s=60)
plt.scatter(c3[0],c3[1], c='blue',s=60)
```

Out[3]: <matplotlib.collections.PathCollection at 0x7f8a6cb08760>



In [5]:
```
# itr = 2 # Tune this variable for better out put
# # c = [0]*10
# # a = c.copy()
# for t in range(0,itr):

#     clustr = []

#     for i in X:
#         clustr = dis(i,c1,c2,c3,clustr)

#     for j in range(0,len(X)-1):
#         if(clustr[j] == 1):
#             c1[0] = np.mean(c1[0],X[j][0])
#             c1[1] = np.mean(c1[1],X[j][1])
#         if(clustr[j] == 2):
#             c2[0] = np.mean(c2[0],X[j][0])
#             c2[1] = np.mean(c2[1],X[j][1])
#         if(clustr[j] == 3):
#             c3[0] = np.mean(c3[0],X[j][0])
#             c3[1] = np.mean(c3[1],X[j][1])

#     print("================================================================
#     print("Iteration", t+1)
#     print(clustr)
#     print("Center of c1",c1)
#     print("Center of c2",c2)
#     print("Center of c3",c3)
#     print("================================================================
```

In [6]:
```
def mnX(cl):
    tm1 = 0
    for q in cl:
        tm1 = tm1 + q[0]
    return(tm1)

def mnY(cl):
    tm2 = 0
    for w in cl:
        tm2 = tm2 + w[1]
    return(tm2)
```

In [7]:
```
itr = 5


for j in range(0,itr):
    cl1 = []
    cl2 = []
    cl3 = []

    for i in X:
        dis1 = np.sqrt((i[0] - c1[0]) ** 2 + (i[1] - c1[1]) ** 2)
        dis2 = np.sqrt((i[0] - c2[0]) ** 2 + (i[1] - c2[1]) ** 2)
        dis3 = np.sqrt((i[0] - c3[0]) ** 2 + (i[1] - c3[1]) ** 2)

        minimum = min(dis1,dis2,dis3)

        if(minimum == dis1): cl1.append(i)
        if(minimum == dis2): cl2.append(i)
        if(minimum == dis3): cl3.append(i)

#   Updating centers
    if(len(cl1) != 0):
        t1X = mnX(cl1)
        t1Y = mnY(cl1)
        c1[0] = t1X/len(cl1)
        c1[1] = t1Y/len(cl1)

    if(len(cl2) != 0):
        t2X = mnX(cl2)
        t2Y = mnY(cl2)
        c2[0] = t2X/len(cl2)
        c2[1] = t2Y/len(cl2)

    if(len(cl3) != 0):
        t3X = mnX(cl3)
        t3Y = mnY(cl3)
        c3[0] = t3X/len(cl3)
        c3[1] = t3Y/len(cl3)

    print("==========================================================================
    print("Iteration", j+1)
    print("Center of c1",c1)
    print("Center of c2",c2)
    print("Center of c3",c3)
    print("==========================================================================
```

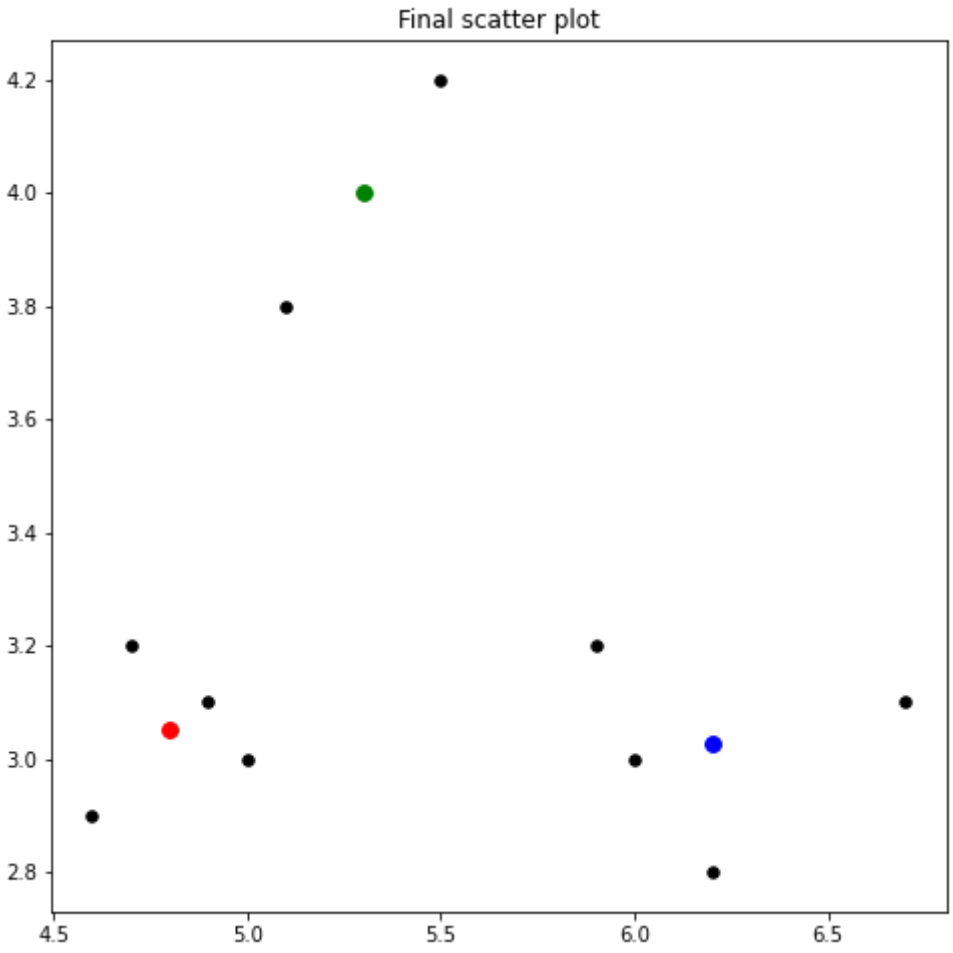
```
==========================================================================
========================
Iteration 1
Center of c1 [5.171428571428572, 3.1714285714285713]
Center of c2 [5.5, 4.2]
Center of c3 [6.45, 2.95]
==========================================================================
========================
==========================================================================
========================
Iteration 2
```

```
Center of c1 [4.800000000000001, 3.05]
Center of c2 [5.3, 4.0]
Center of c3 [6.2, 3.025]
========================================================================
======================
========================================================================
======================
Iteration 3
Center of c1 [4.800000000000001, 3.05]
Center of c2 [5.3, 4.0]
Center of c3 [6.2, 3.025]
========================================================================
======================
========================================================================
======================
Iteration 4
Center of c1 [4.800000000000001, 3.05]
Center of c2 [5.3, 4.0]
Center of c3 [6.2, 3.025]
========================================================================
======================
========================================================================
======================
Iteration 5
Center of c1 [4.800000000000001, 3.05]
Center of c2 [5.3, 4.0]
Center of c3 [6.2, 3.025]
========================================================================
======================
```

In [8]:
```
# # rough cell
# # run function for summing
# # divide by len of cl after fun returns sum
# tmp1,tmp2 = 0,0
# for i in cl1:
# #
#     tmp1 = tmp1 + i[0]
#     tmp2 = tmp2 + i[1]

# tmp = tmp1/len(cl1)
# tmp
```

In [9]:
```
fig = plt.figure(figsize=(8, 8))
for i in X:
    plt.scatter(i[0],i[1],c = "black",s=30)
plt.title('Final scatter plot')
plt.scatter(c1[0],c1[1], c='red',s=60)
plt.scatter(c2[0],c2[1],c='green',s=60)
plt.scatter(c3[0],c3[1], c='blue',s=60)
```

Out[9]: <matplotlib.collections.PathCollection at 0x7f8a6cd06a60>



In [10]:
```
del(c1)
del(c2)
del(c3)
```

In [ ]:
```

```