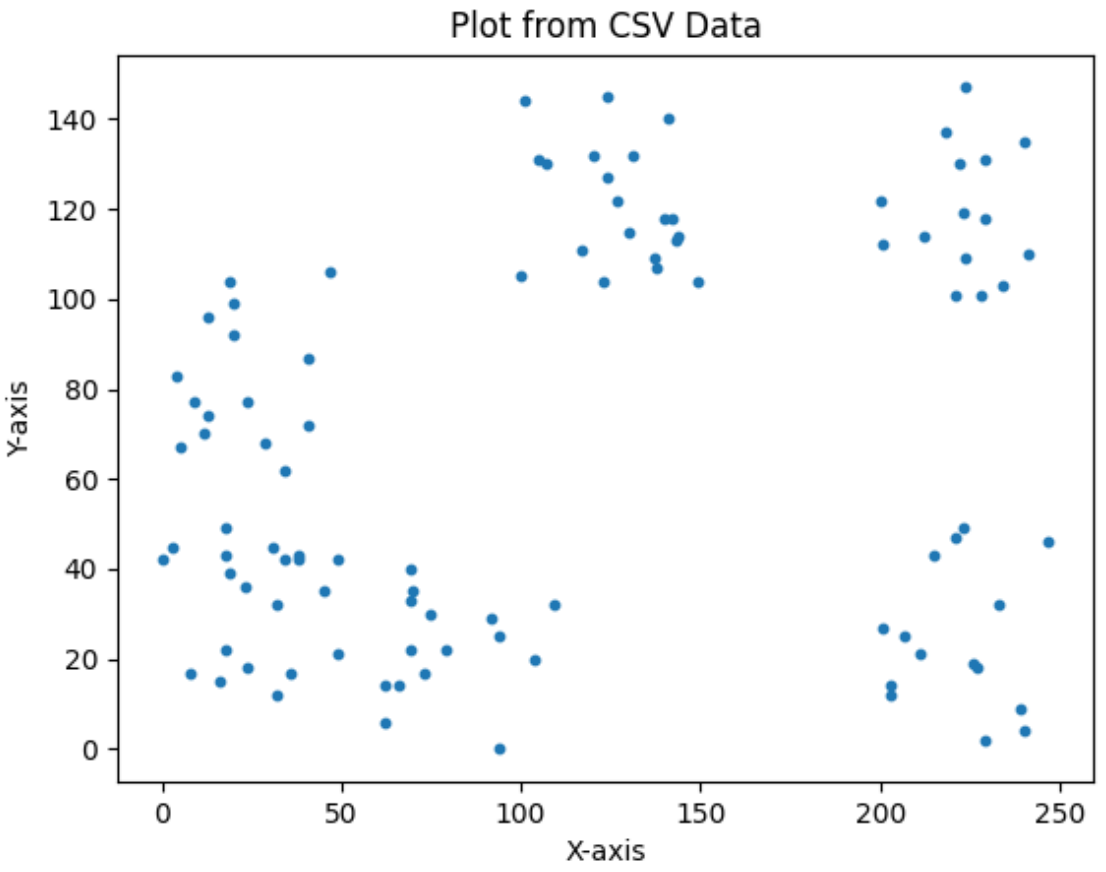
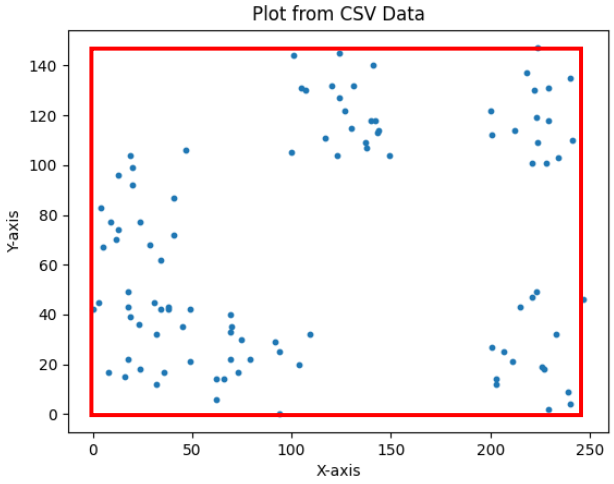
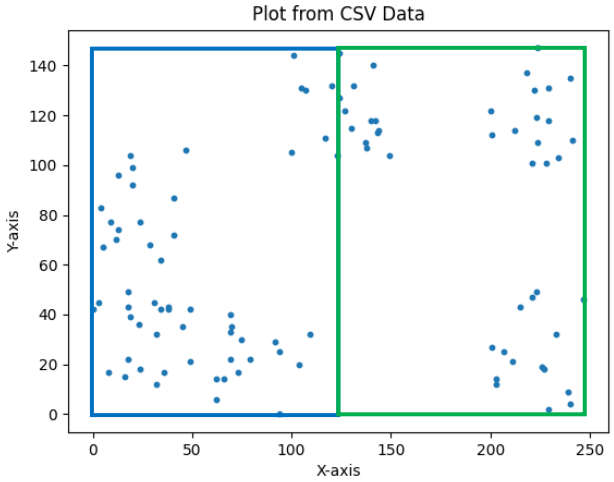
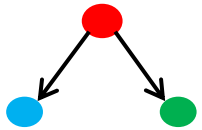

Plot from CSV Data
Y-axis
X-axis
0
20
40
60
80
100
120
140
0
50
100
150
200
250



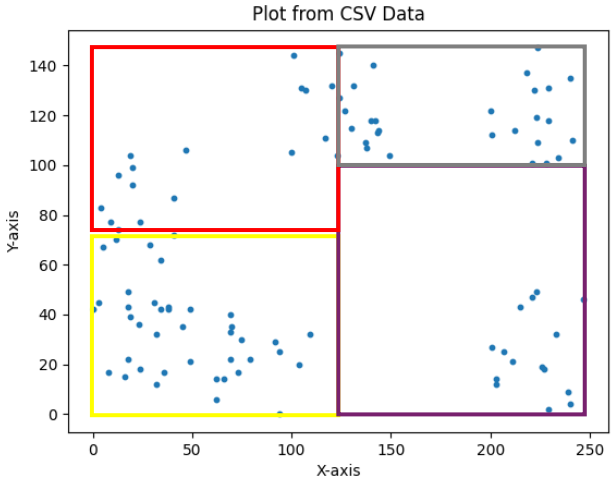
Root



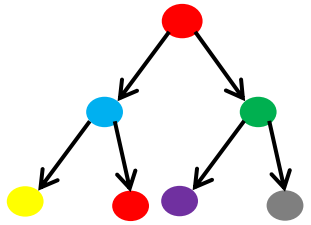
Root



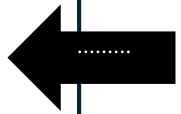
X-Axis Split



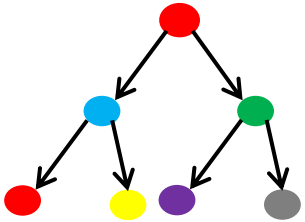
Root

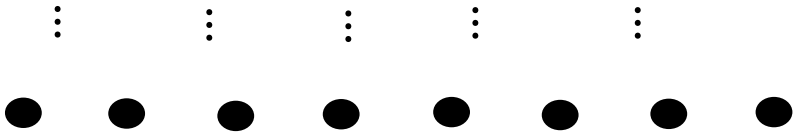


Y-Axis Split



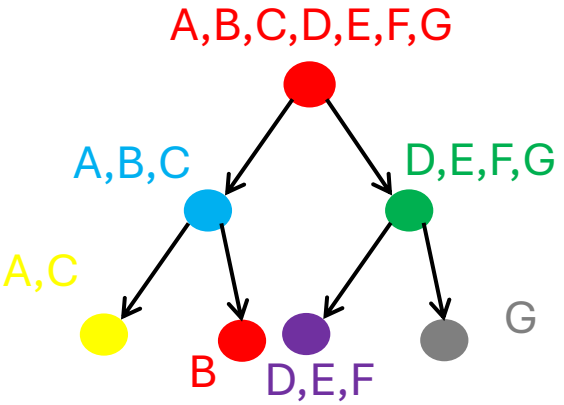
Root

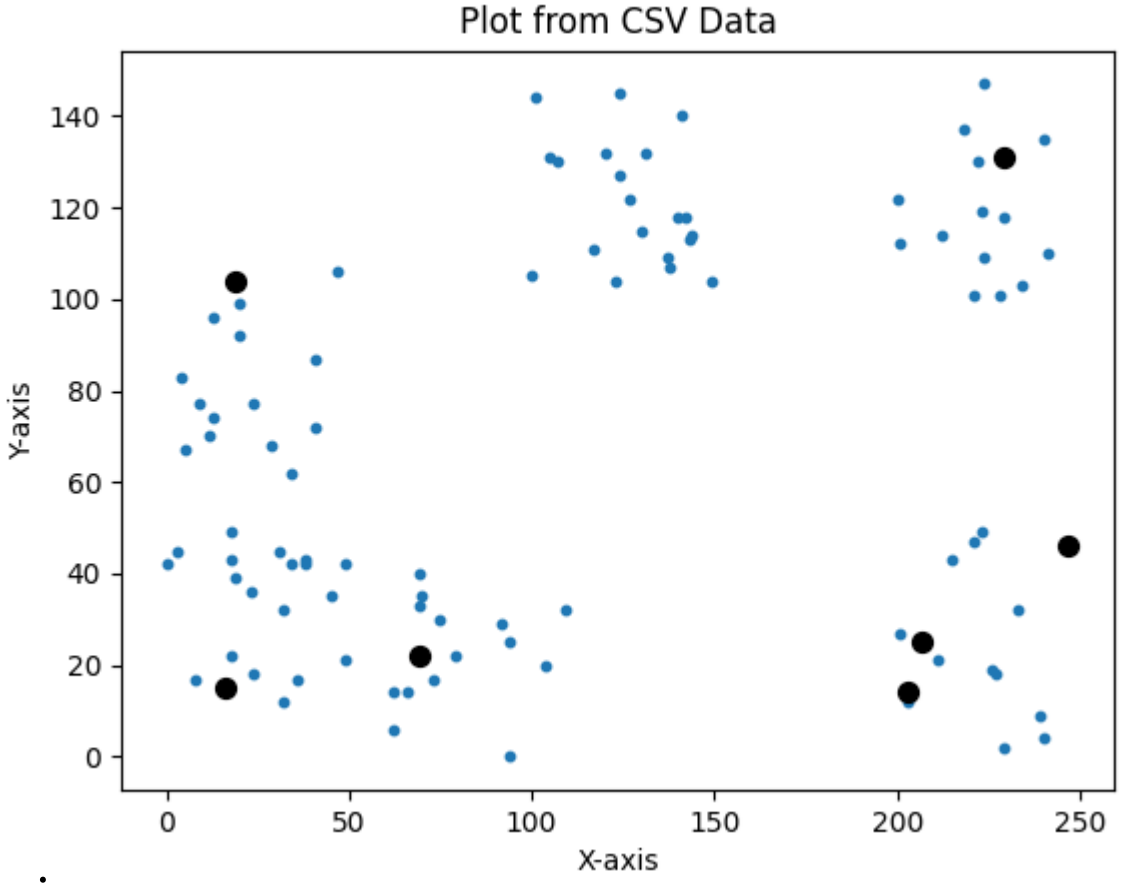


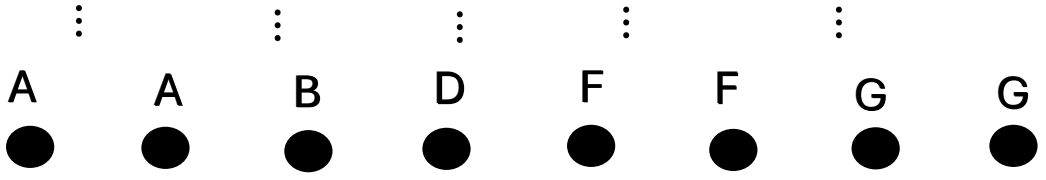


Leaf = Points

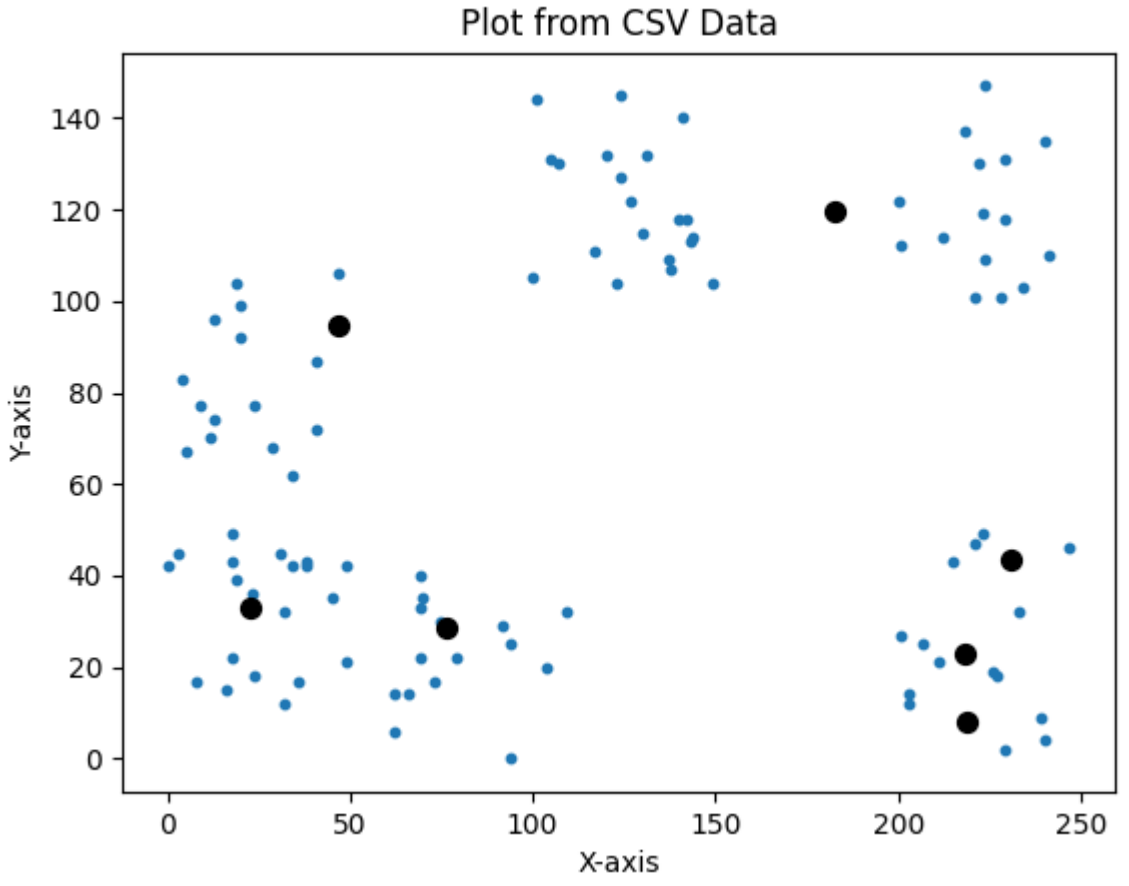
Centroids:
A = (16, 15) E = (247, 46)
B = (69, 22) F = (203, 14)
C =(19, 104) G = (229, 131)
D =(207, 25)

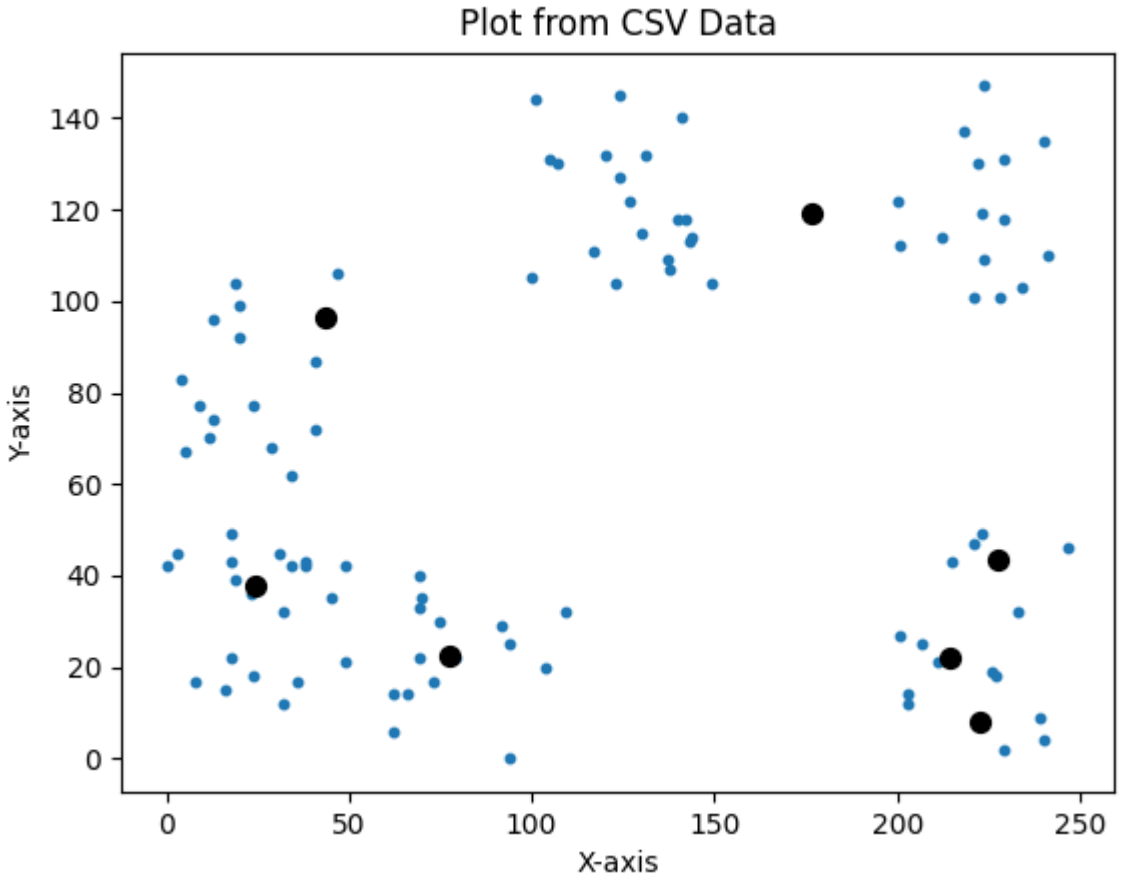


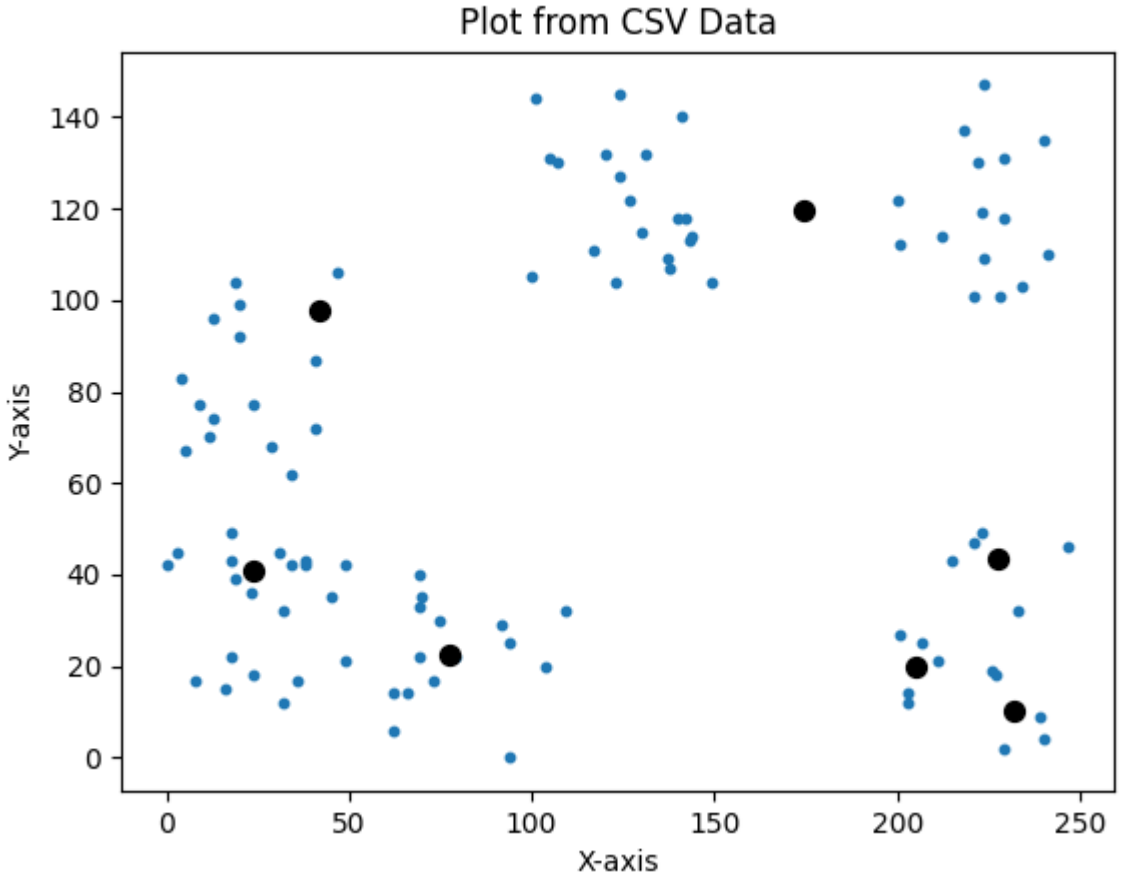




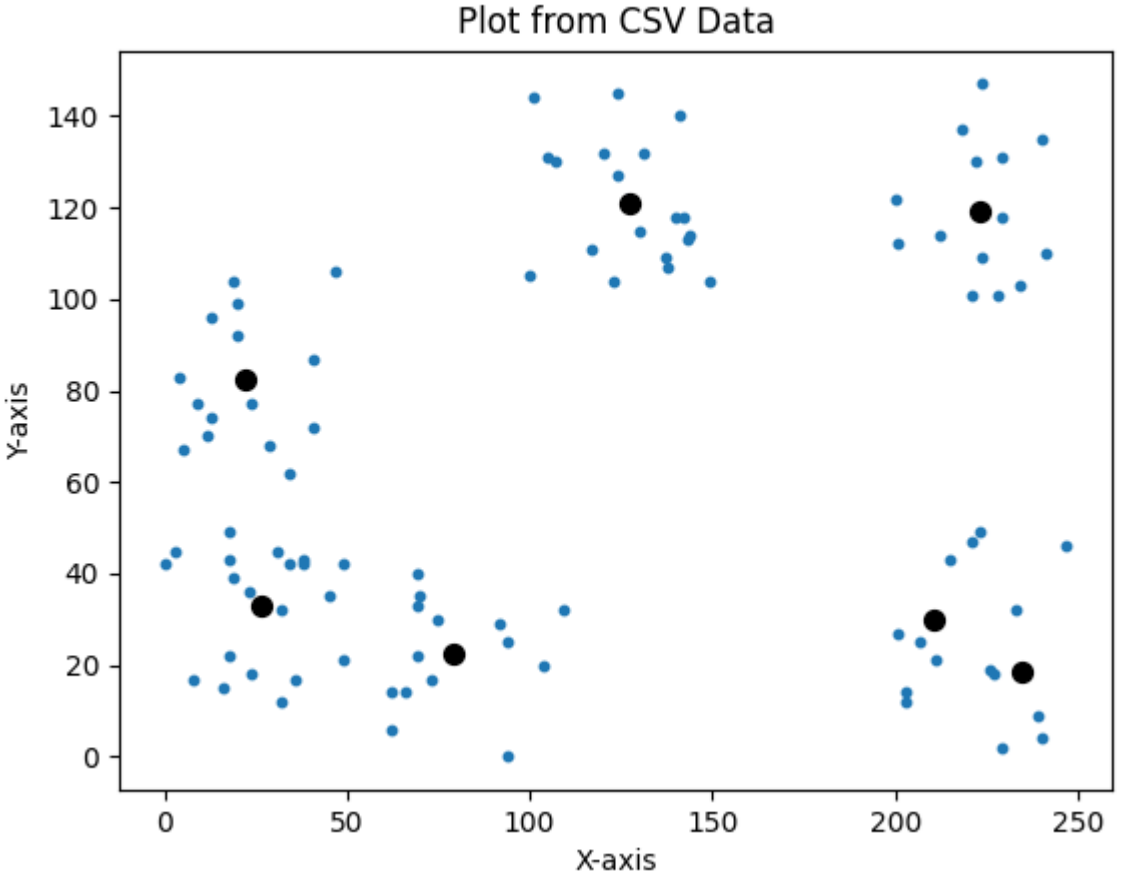
Then each point pulls its centroid near its position (its done with a weighted average)

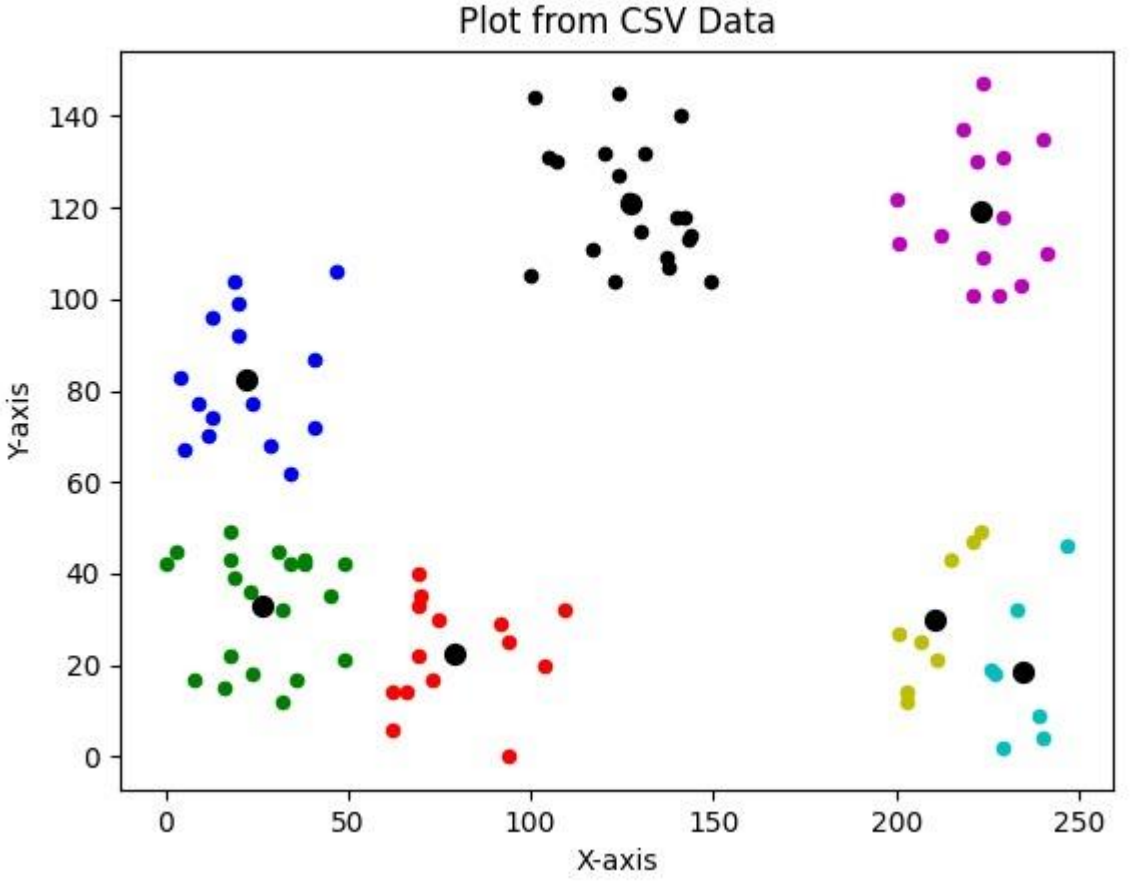

Plot from CSV Data
Y-axis
X-axis
0
20
40
60
80
100
120
140
0
50
100
150
200
250

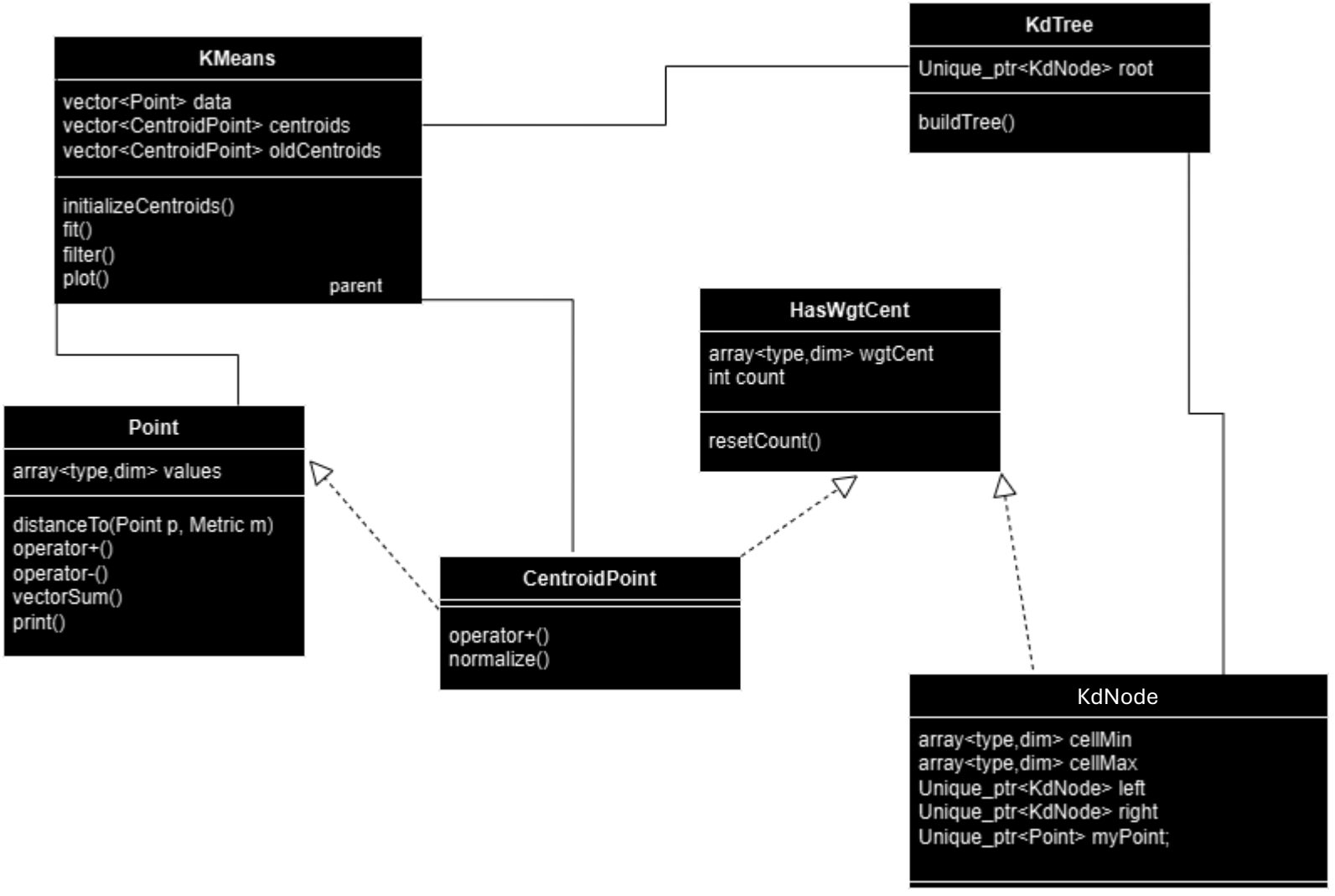

Plot from CSV Data
Y-axis
X-axis
0
20
40
60
80
100
120
140
0
50
100
150
200
250


Plot from CSV Data
Y-axis
X-axis
0
20
40
60
80
100
120
140
0
50
100
150
200
250

Convergence




Plot from CSV Data
Y-axis
X-axis
0
20
40
60
80
100
120
140
0
50
100
150
200
250


KMeans
vector<Point> data
vector<CentroidPoint> centroids
vector<CentroidPoint> oldCentroids
initializeCentroids()
fit()
filter()
plot()
parent
KdTree
Unique_ptr<KdNode> root
buildTree()
HasWgtCent
array<type,dim> wgtCent
int count
resetCount()
Point
array<type,dim> values
distanceTo(Point p, Metric m)
operator+()
operator-()
vectorSum()
print()
CentroidPoint
operator+()
normalize()
KdNode
array<type,dim> cellMin
array<type,dim> cellMax
Unique_ptr<KdNode> left
Unique_ptr<KdNode> right
Unique_ptr<Point> myPoint;