

# Kmeans.mrql

```
type point = < X: double, Y: double >;

function distance ( x: point, y: point ): double {
  sqrt(pow(x.X-y.X,2)+pow(x.Y-y.Y,2))
};

aggregation new_centroid (
  \ (p:(point,long),q:(point,long)):(point,long)
  . ( < X: p#0.X+q#0.X, Y: p#0.Y+q#0.Y >,
    p#1+q#1),
  ( < X: 0.0 as double, Y :0.0 as double >, 0 as long )
) : (point,long);

function centroid ( p: (point,long), default: point ): point {
  if p#1 = 0
    then default
  else < X: p#0.X/p#1, Y: p#0.Y/p#1 >
};

repeat centroids = { < X: 0.0 as double, Y: 0.0 as double >,
  < X: 10.0, Y: 0.0 >,
  < X: 0.0, Y: 10.0 >,
  < X: 10.0, Y: 10.0 > }
step select ( centroid(new_centroid(select (p,l as long) from p in s),k), true )
  from s in source(binary,"points.bin")
  group by k: (select c from c in centroids order by distance(c,s))[0]
limit 5;
```