As a graph search:
- verts are core points
- edges when 2 core points are w/in ε of eachother

Find connected components
- each component is a cluster

Finally - add in border pts w/ in ε

Computational complexity

High level
1. $\forall \vec{x} \in D$
- compute $N_\varepsilon(\vec{x})$
- check if $\vec{x}$ is a core point
- init $\vec{x}$ as not in a cluster

2. $\forall$ core point
find all density connected points

for each point

time proportional to

$O(n)$ in high dim

$O(\log n)$ in low dim (hidden d's)

$O(1)$

$O(1)$

$O(n)$

not going to dominate

Total time:
- $O(n^2)$ in high dim
- or $O(n \log n)$ in low dim

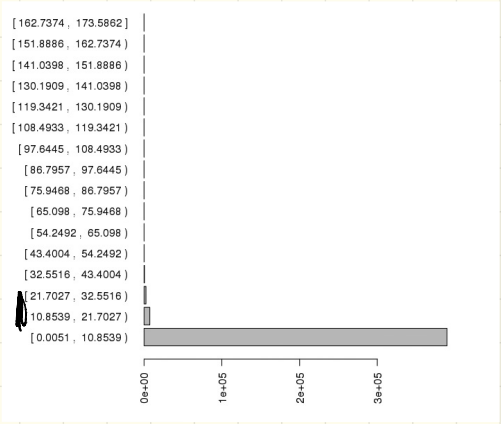
Things to consider
- user supplied params minpts, ε
- only 1 ε value is allowed (can't handle varying densities)

Soln 1: OPTICS - improvement of OB Scan

Soln 2: in practice

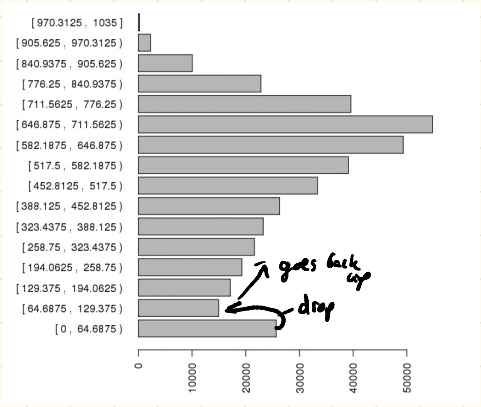
https://github.com/alitouka/spark_dbscan/wiki/Choosing-parameters-of-DBSCAN-algorithm

dist to nearest neighbor



pick $\varepsilon = 22$

fix $\varepsilon = 22$ look at # of core pts for varying minpts



choose min pts = 129