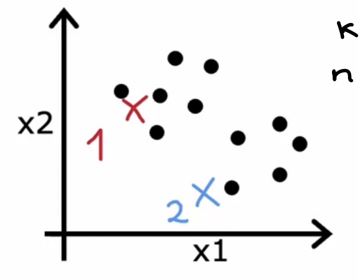
First randomly initialize $\underbrace{\text{cluster centroids}}_{\mu_1, \mu_2, \ldots\ldots, \mu_K}$

centroid will have the same dimension as training set.



$K = 2$ (previous example)
$n = 2$ (means μ_1 and μ_2 are two dimensional)

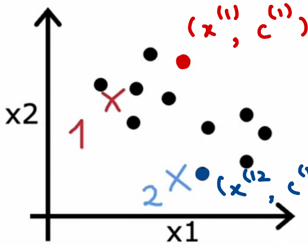
First step : Assign points to cluster

basically checking which cluster centroid is close to each point by its index.

```
for i = 1 to m
    /* $c^{(i)}$ :- index (from 1 to K) of
       cluster centroid closest to $x^{(i)}$ */
    $c^{(i)} := \min_K \| x^{(i)} - \mu_K \|^2$
}
```

finding value of K such μ_K is the nearest centroid for all K.

eg.



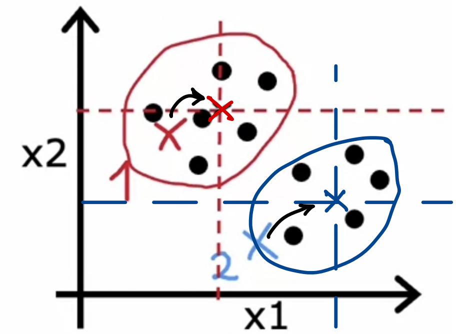
$(x^{(1)}, c^{(1)}) = \underline{1}$ → closer to centroid 1 ∴ assigned 1.

$(x^{(12}, c^{(12)}) = 2$ → closer to centroid 2.

# Second step : Rearrange cluster points by moving the centroid

for $K = 1$ to K

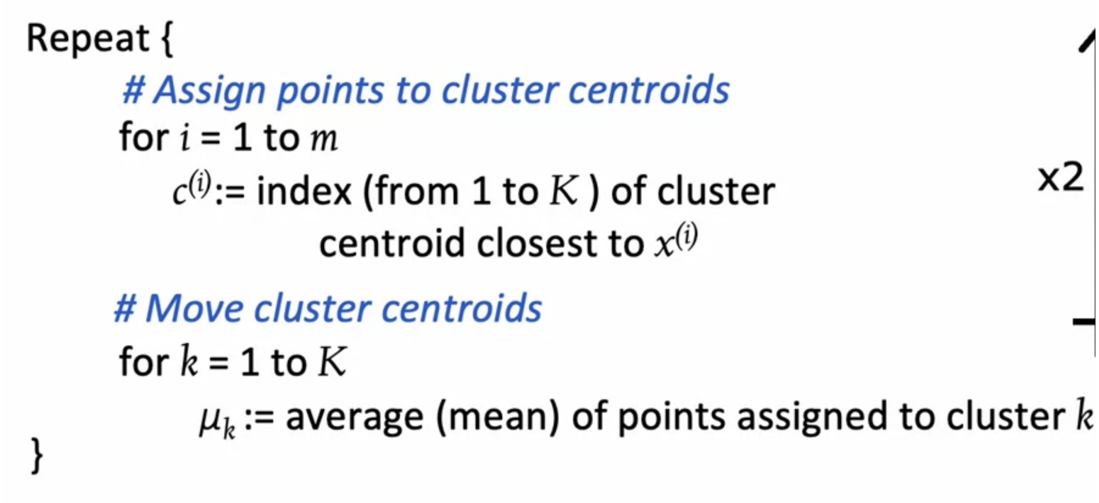
μ_K :- average mean of points assigned to cluster K.



We'll look at the points' average location on horizontal and vertical axis.

Mathematically, an average would be $$\underbrace{\frac{x^{(1)}+x^{(2)}+x^{(3)}+x^{(4)}}{4}}_{\text{new location of } \mu_1}$$

In summary, the algorithm works like this:

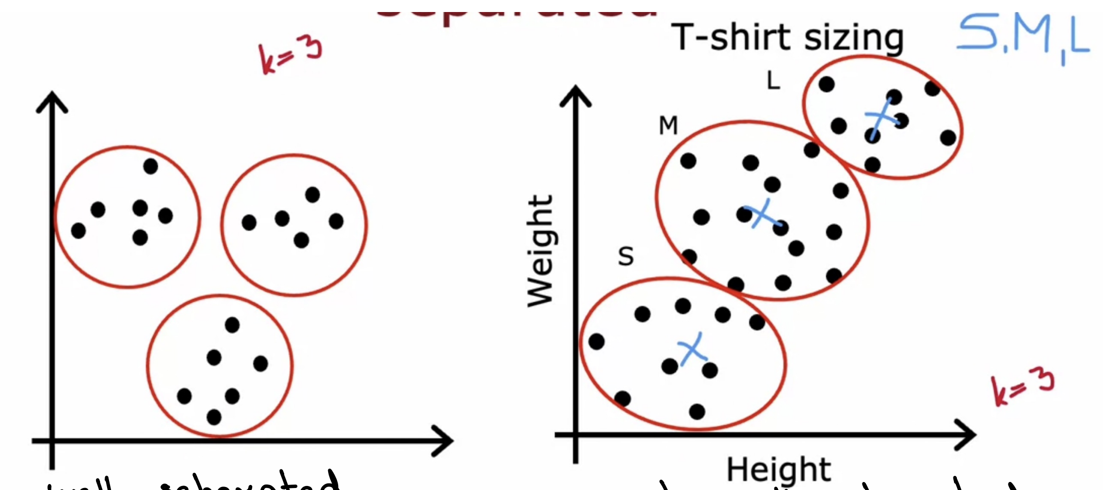


```
Repeat {
    # Assign points to cluster centroids
    for $i$ = 1 to $m$
        $c^{(i)}$ := index (from 1 to $K$ ) of cluster
                    centroid closest to $x^{(i)}$

    # Move cluster centroids
    for $k$ = 1 to $K$
        $\mu_k$ := average (mean) of points assigned to cluster $k$
}
```

If no points are assigned to a cluster centroid then it is removed i.e. ~~K = 2~~ K = 1 or we reinitialize the $\mu_1 \ldots \mu_K$ hoping it gets assigned some points but in majority cases we just remove it

K means will also work for points that are not well separated



well separated

not well separated
eg. T-shirt sizing