How does it work?

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(clusterlab)

normalize <- function(x){((x-min(x))/(max(x)-min(x)))}
```

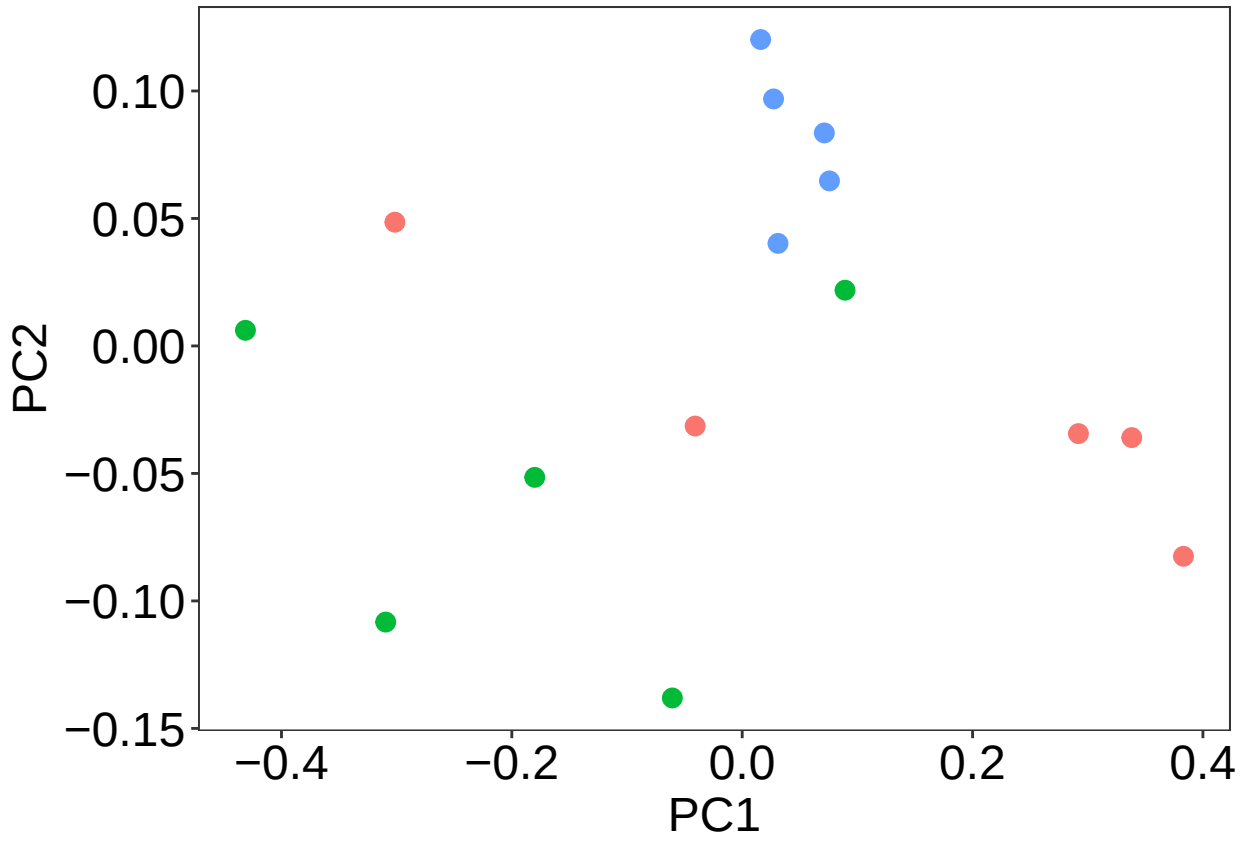
We are goint yo use the following dataset:

```
## ***clusterlab***
```

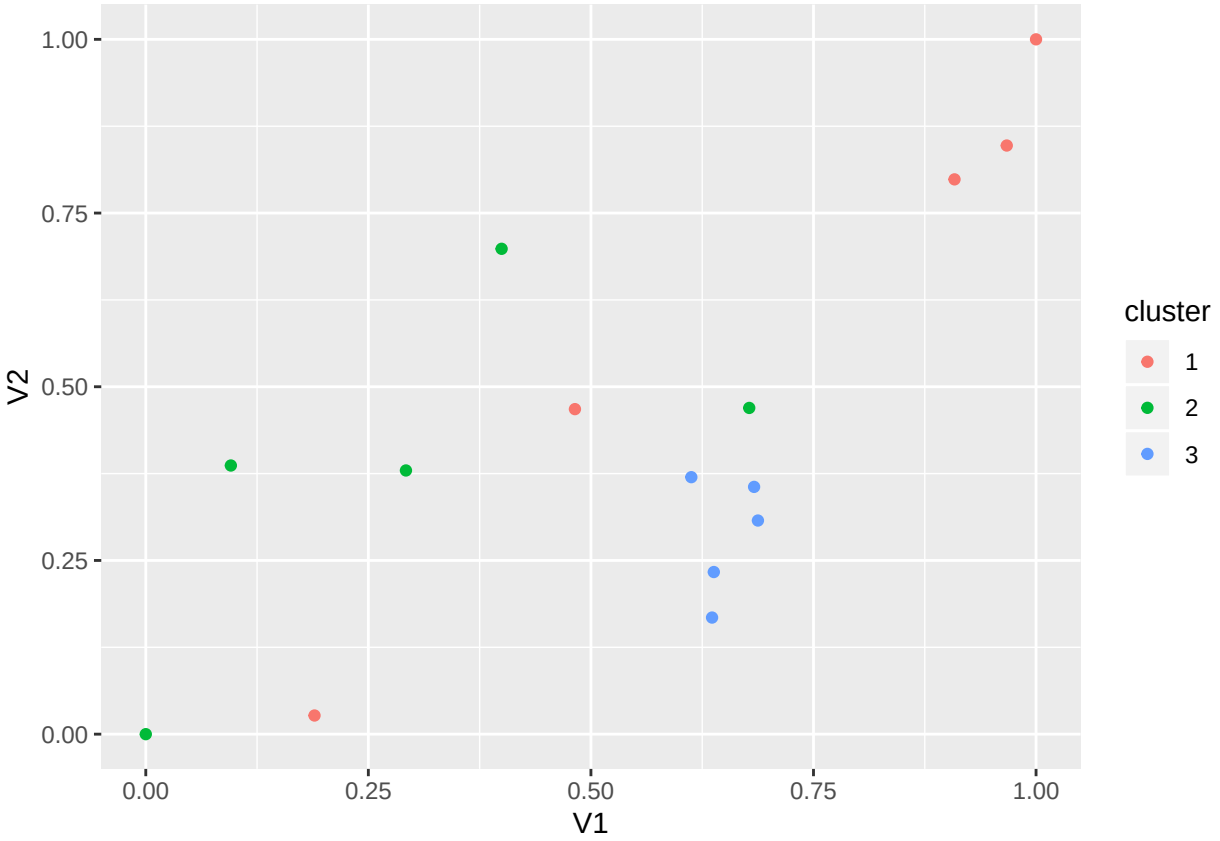
```
## mode: circle
```

```
## simulating clusters...
```

```
## finished.
```




```
##         [,1]         [,2]         [,3]         [,4]         [,5]
## V1      "0.1895123"  "0.9084296"  "1.0000000"  "0.4820551"  "0.9670973"
## V2      "0.02686126" "0.79848540" "1.00000000" "0.46780742" "0.84717833"
## cluster "1"          "1"          "1"          "1"          "1"
##         [,6]         [,7]         [,8]         [,9]         [,10]
## V1      "0.0955437"  "0.6778422"  "0.0000000"  "0.3996768"  "0.2922267"
## V2      "0.38667930" "0.46953135" "0.00000000" "0.69853537" "0.37953080"
## cluster "2"          "2"          "2"          "2"          "2"
##         [,11]        [,12]        [,13]        [,14]        [,15]
## V1      "0.6875977"  "0.6833037"  "0.6380608"  "0.6360855"  "0.6127692"
## V2      "0.30746601" "0.35589478" "0.23335390" "0.16782195" "0.36987061"
## cluster "3"          "3"          "3"          "3"          "3"
```



Assignation

Update

After the assignation step a dataset with all the data + an additional column `cluster` containing the cluster to which the object has been assigned is available.

Now it is time to update the cluster.

1. For each distance function d_ℓ, a matrix M_{d_ℓ} containing the distance between each pair of objects o_i and o_j $d_\ell(o_i, o_j)$ is stored
2. For each cluster j, the matrices M are filtered to keep only the points that belong to the cluster j
3. For each point, the distance to all other points is added, obtaining a single number for each pair of object and distance function
4. The vector containing all the points of the dataset and their respective pair values are ranked from closest to fruthest
5. Borda account is applied to all the rankings obtained
6. The borda winner is the one chosen as centroid