Ex. 7

1.1

1. Supervised Learning e.g Classification, Regression algortihms
2. Unsupervised Learning e.g K-Means and EM algorithms
3. Reinforcement Learning e.g Open-source RL algorithms
4. Deep learning e.g Generative Adversarial Networks, Self-Organizing Maps
5. Representation learning e.g Greedy layer-wise unsupervised pretraining protocol
6. Hybrid and ensemble learning approaches e.g Ensemble learning algorithms

3 K-Nearest Neighbours

$||P - A|| = ||(17; 1; 4)|| = \sqrt{306}$
$||P - B|| = ||(10; 2; 3)|| = \sqrt{113}$
$||P - C|| = ||(13; 1; 3)|| = \sqrt{179}$
$||P - D|| = ||(11; -1; 0)|| = \sqrt{122}$
$||P - E|| = ||(14; -1; 1)|| = \sqrt{198}$
$||P - F|| = ||(16; -1; 1)|| = \sqrt{258}$
$\implies$ B is nearest neighbor and P is therefore in class 1.

4 K-Means Clustering

In [2]:
```
library(tidyverse)
```

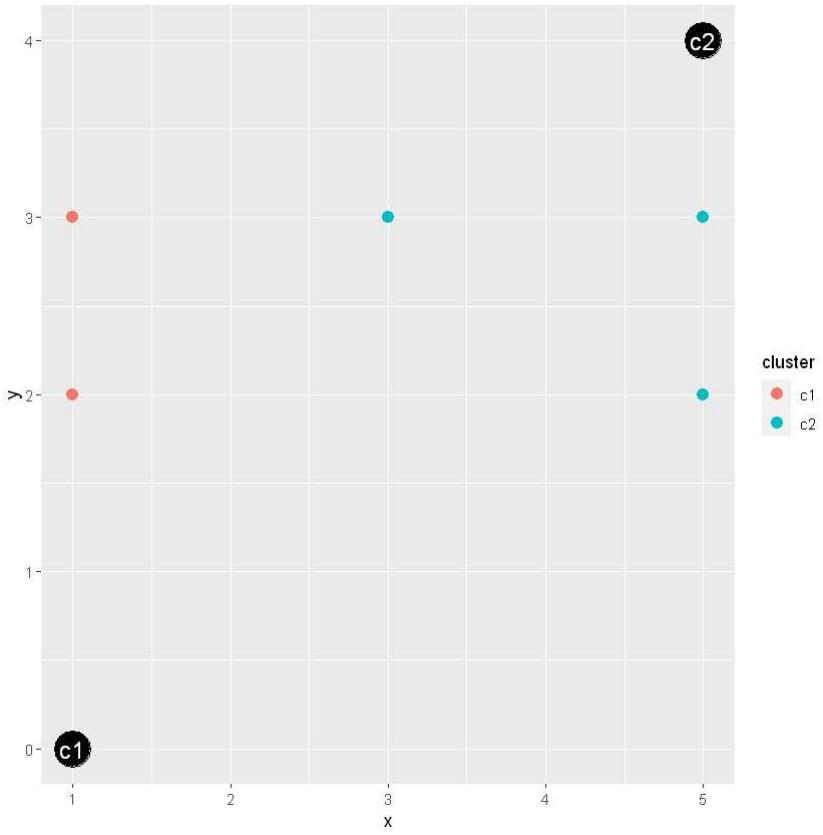
```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
✓ ggplot2 3.4.0      ✓ purrr   0.3.5
✓ tibble  3.1.8      ✓ dplyr   1.0.10
✓ tidyr   1.2.1      ✓ stringr 1.5.0
✓ readr   2.1.3      ✓ forcats 0.5.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

In [97]:
```
data = as_tibble(data.frame(x = c(1,3,5,1,5), y=c(3,3,3,2,2)))
centers = as_tibble(data.frame(name=c('c1', 'c2'), x=c(1,5), y=c(0, 4)))
nearest_cluster <- Vectorize(function(xd,yd) {
    (centers %>% mutate(dist=((x-xd)^2 + (y-yd)^2))%>% slice(which.min(dist)))$name
})
data['cluster']=apply(data[,c('x', 'y')], 1, function(x) nearest_cluster(x[1], x[2]
head(data)
```

A tibble: 5 × 3

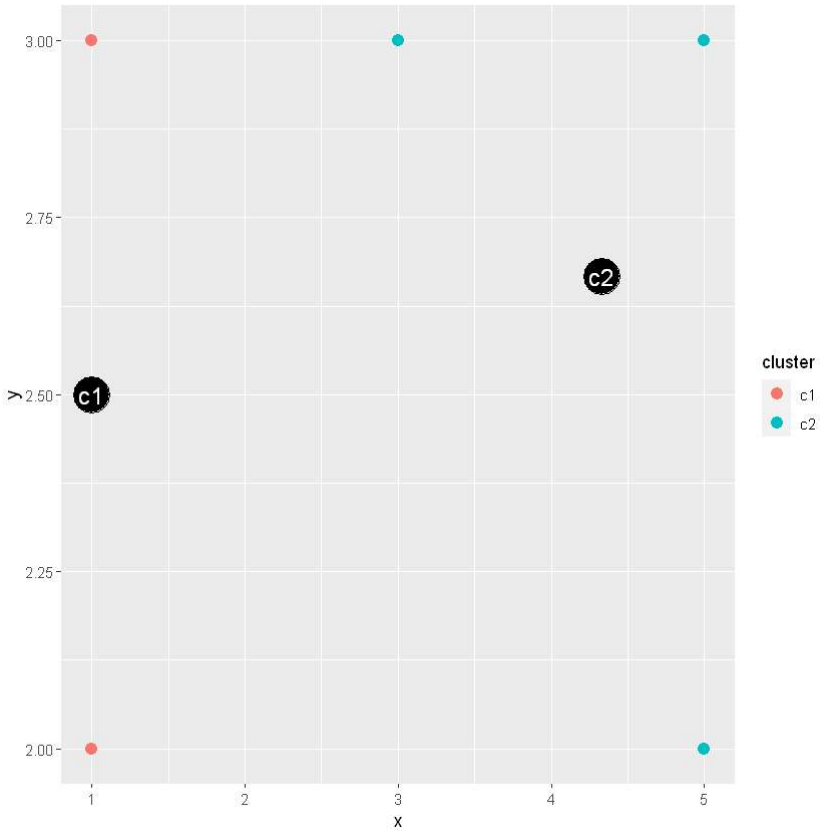
x	y	cluster
<dbl>	<dbl>	<chr>
1	3	c1
3	3	c2
5	3	c2
1	2	c1
5	2	c2

In [98]:
```
ggplot() +
    geom_point(data = data, mapping = aes(x=x, y=y, colour=cluster), size=3) +
    geom_point(data = centers, mapping = aes(x=x, y=y), colour="black", size=10) +
    geom_text(data = centers, mapping = aes(x = x, y=y, label=name), color="white",
```



In [99]:
```
for (x in 1:50) {
    centers$x = apply(centers["name"], 1, function(name) mean(filter(data, cluster=
    centers$y = apply(centers["name"], 1, function(name) mean(filter(data, cluster=
    data['cluster']=apply(data[,c('x', 'y')], 1, function(x) nearest_cluster(x[1],
}
```

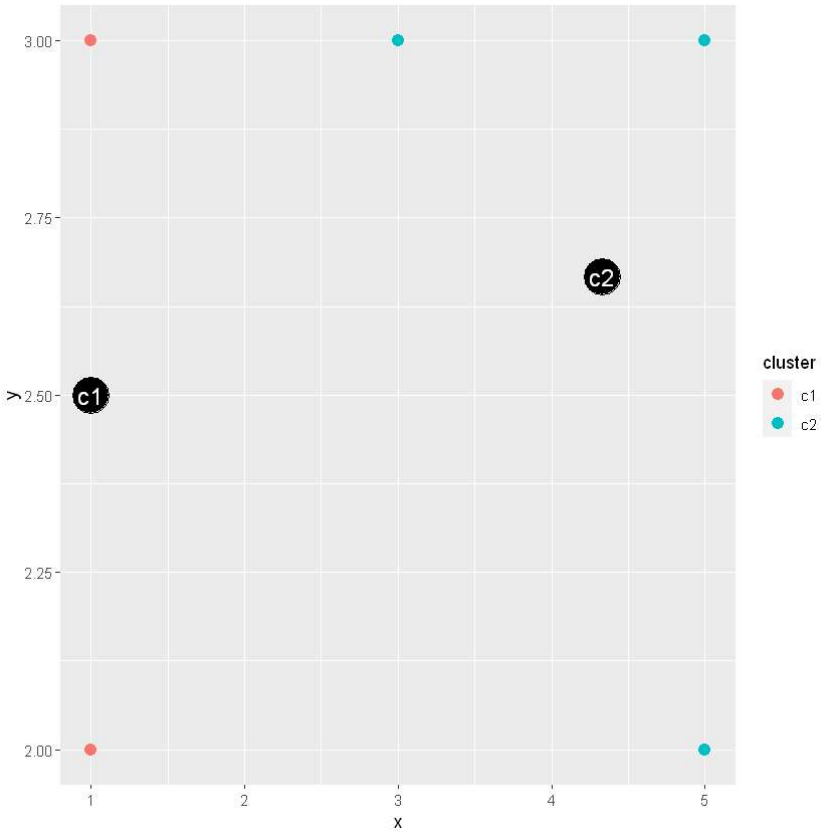
In [100…
```
ggplot() +
    geom_point(data = data, mapping = aes(x=x, y=y, colour=cluster), size=3) +
    geom_point(data = centers, mapping = aes(x=x, y=y), colour="black", size=10) +
    geom_text(data = centers, mapping = aes(x = x, y=y, label=name), color="white",
```



In [101…

```
for (x in 1:100) {
    centers$x = apply(centers["name"], 1, function(name) mean(filter(data, cluster=
    centers$y = apply(centers["name"], 1, function(name) mean(filter(data, cluster=
    data['cluster']=apply(data[,c('x', 'y')], 1, function(x) nearest_cluster(x[1],
}

ggplot() +
    geom_point(data = data, mapping = aes(x=x, y=y, colour=cluster), size=3) +
    geom_point(data = centers, mapping = aes(x=x, y=y), colour="black", size=10) +
    geom_text(data = centers, mapping = aes(x = x, y=y, label=name), color="white",
```



looks like we are stable