

Agrupamiento de datos

Clustering

Diana L. Giraldo

Febrero 2021

1. Holi!
2. Métodos de agrupamiento
3. ¿Cómo elegir k ?

Holi!

¿Qué hago?

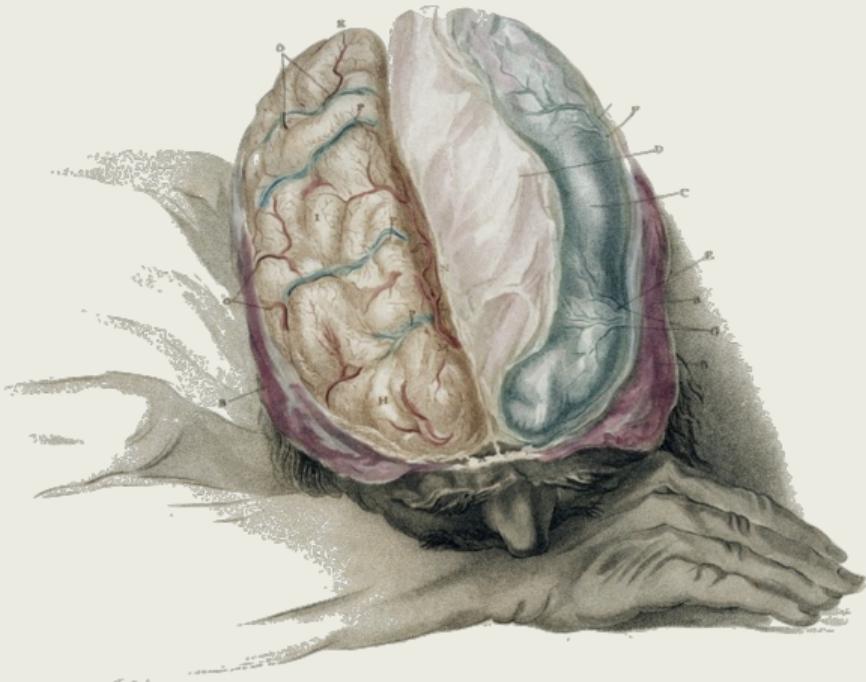
En mi tesis de doctorado trabajo con información de:

- Neuroimágenes
- Variables clínicas
- Evaluación neuropsicológica

para la caracterización de patrones asociados a la

enfermedad de Alzheimer,

la causa más común de demencia.





Antes de la demencia, cuando los pacientes muestran alguna disfunción cognitiva, se diagnostica el

Deterioro Cognitivo Leve

Mild cognitive impairment (MCI).

Un conjunto de tests neurpsicológicos evalúan las funciones cognitivas en diferentes dominios: memoria, lenguaje, habilidades visuo-espaciales, etc.

¿qué vamos a hacer hoy?

Usar puntajes por dominio para **encontrar sub-grupos** de pacientes con Deterioro Cognitivo Leve.

Los datos de prueba, código y esta presentación están en

https://github.com/diagiraldo/RLadies_bta_2021/

¿qué vamos a hacer hoy?

Usar puntajes por dominio para **encontrar sub-grupos** de pacientes con Deterioro Cognitivo Leve.

Los datos de prueba, código y esta presentación están en

https://github.com/diagiraldo/RLadies_bta_2021/

En RStudio:

```
A <- read.csv("datos/neuropsychological_data.csv")  
head(A)
```

	id	edad	edu	genero	dx	progresion	t_progresion	t_seguimiento	MEMORY	LANGUAGE	EXECUTIVE	VISUOSPATIAL	ORIENTATION	ATTENTION
1	1	79.3	16	Male	CN	TRUE	24	84	0.1983688	-0.03209976	0.32784032	-0.4572322	-0.5705812	-0.3139389
2	2	89.5	18	Female	CN	TRUE	60	60	-1.1210747	0.29438495	-0.36517101	1.7030370	-0.8349403	0.2254791
3	3	86.8	18	Male	MCI	FALSE	NA	6	-1.3938079	-0.41181495	-0.46242071	-1.1384656	-0.5148650	2.0804932
4	4	77.6	18	Female	CN	FALSE	NA	84	-1.2479705	-0.67085719	-0.34398220	-0.7515376	-0.8615245	-0.2979084
5	5	76.7	14	Male	CN	FALSE	NA	60	-0.5485783	-0.04031867	0.05542082	-0.4822031	-0.9414579	-0.2955286
6	6	82.7	18	Female	CN	FALSE	NA	96	-1.3993788	-0.29124856	-0.21407394	0.5718734	0.1803515	-0.3387569

Variables

Puntajes por dominio:

- MEMORY
- LANGUAGE
- EXECUTIVE
- VISUOSPATIAL
- ORIENTATION
- ATTENTION

Información demográfica:

- edad
- género
- edu: años de educación.

Información clínica:

- dx: diagnóstico
- CN: control
- MCI: deterioro cognitivo leve
- progresion: diagnóstico de demencia en futuras visitas?
- t_progresión: meses entre visita y progresión.
- t_seguimiento: meses de seguimiento a la persona.

Vamos a usar los **puntajes por dominio** para encontrar sub-grupos de pacientes con Deterioro Cognitivo Leve.

Vamos a usar los **puntajes por dominio** para encontrar sub-grupos de pacientes con Deterioro Cognitivo Leve.

En RStudio:

Vamos a usar los **puntajes por dominio** para encontrar sub-grupos de pacientes con Deterioro Cognitivo Leve.

En RStudio:

Cargar librerías:

```
library(dplyr)
library(reshape2)
library(ggplot2)
library(cluster)
library(factoextra)
library(NbClust)
```

Vamos a usar los **puntajes por dominio** para encontrar sub-grupos de pacientes con Deterioro Cognitivo Leve.

En RStudio:

Cargar librerías:

```
library(dplyr)  
library(reshape2)  
library(ggplot2)  
library(cluster)  
library(factoextra)  
library(NbClust)
```

Filtrar datos:

```
B <- A %>%  
  filter(dx == "MCI") %>%  
  select(MEMORY, LANGUAGE, EXECUTIVE,  
         VISUOSPATIAL, ORIENTATION, ATTENTION)
```

B tiene 680 observaciones con 6 variables

Métodos de agrupamiento

Agrupamiento de datos (Clustering)

Conjunto de métodos **no-supervisados** para dividir un conjunto de datos en sub-grupos de tal forma que:

- Observaciones muy parecidas pertenecen a un mismo grupo.
- Observaciones muy diferentes pertenecen a distintos sub-grupos.

Agrupamiento de datos (Clustering)

Conjunto de métodos **no-supervisados** para dividir un conjunto de datos en sub-grupos de tal forma que:

- Observaciones muy parecidas pertenecen a un mismo grupo.
- Observaciones muy diferentes pertenecen a distintos sub-grupos.

Algunos de los métodos más conocidos:

- k -means
- k -medoids
- Hierarchical clustering
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Agrupamiento de datos (Clustering)

Conjunto de métodos **no-supervisados** para dividir un conjunto de datos en sub-grupos de tal forma que:

- Observaciones muy parecidas pertenecen a un mismo grupo.
- Observaciones muy diferentes pertenecen a distintos sub-grupos.

Algunos de los métodos más conocidos:

- *k-means*
- *k-medoids*
- Hierarchical clustering
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

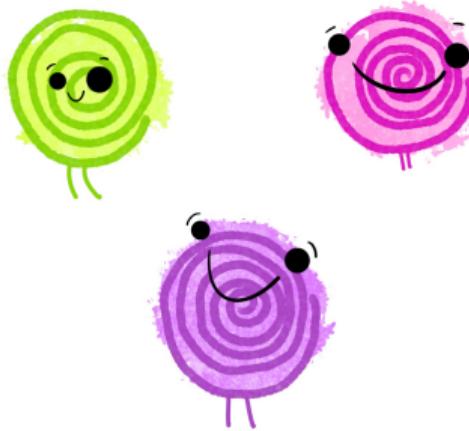
k-means clustering

OBSERVATIONS



- assign each observation to one of k clusters based on the nearest cluster centroid.

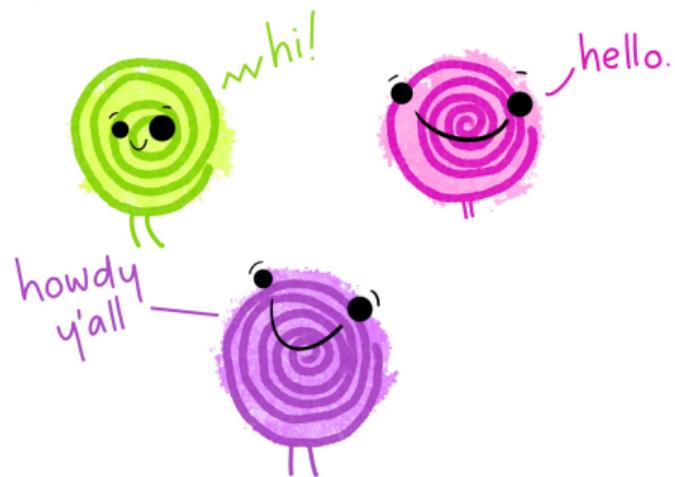
cluster
CENTROIDS



①

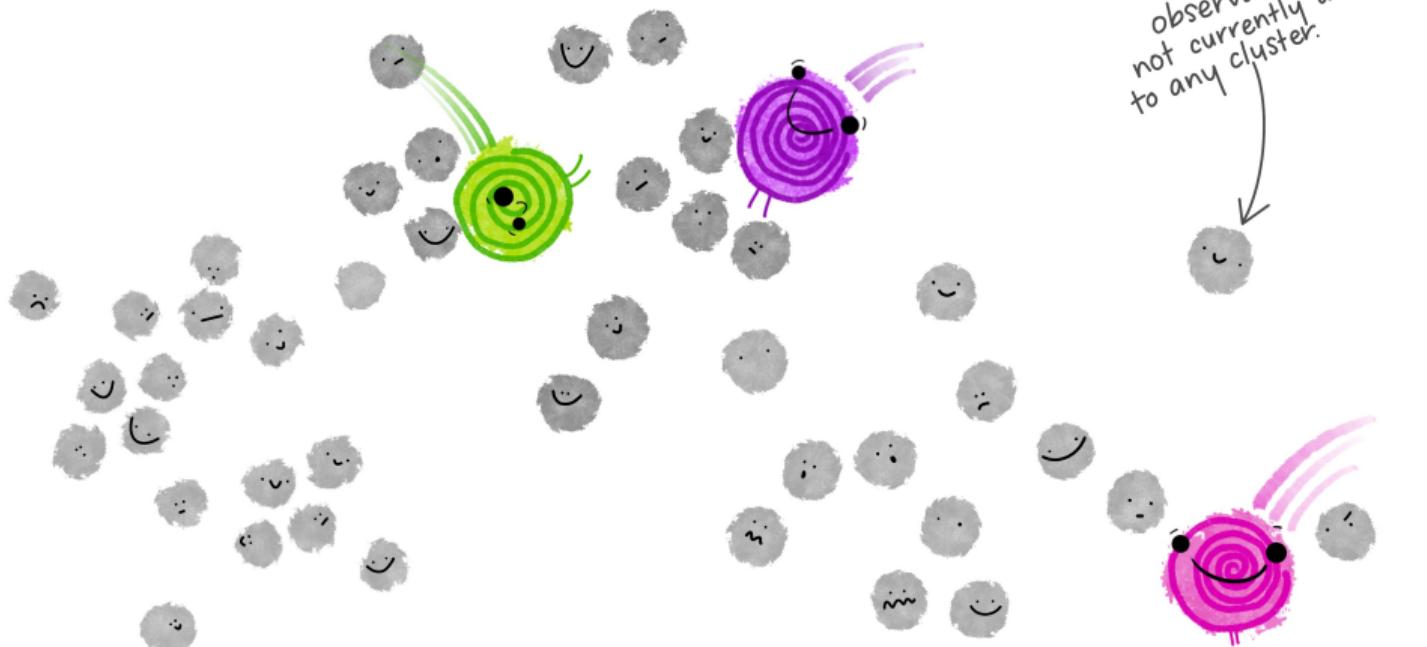
Specify the number of clusters (in this example, $k=3$).

Then imagine k cluster centroids are created.



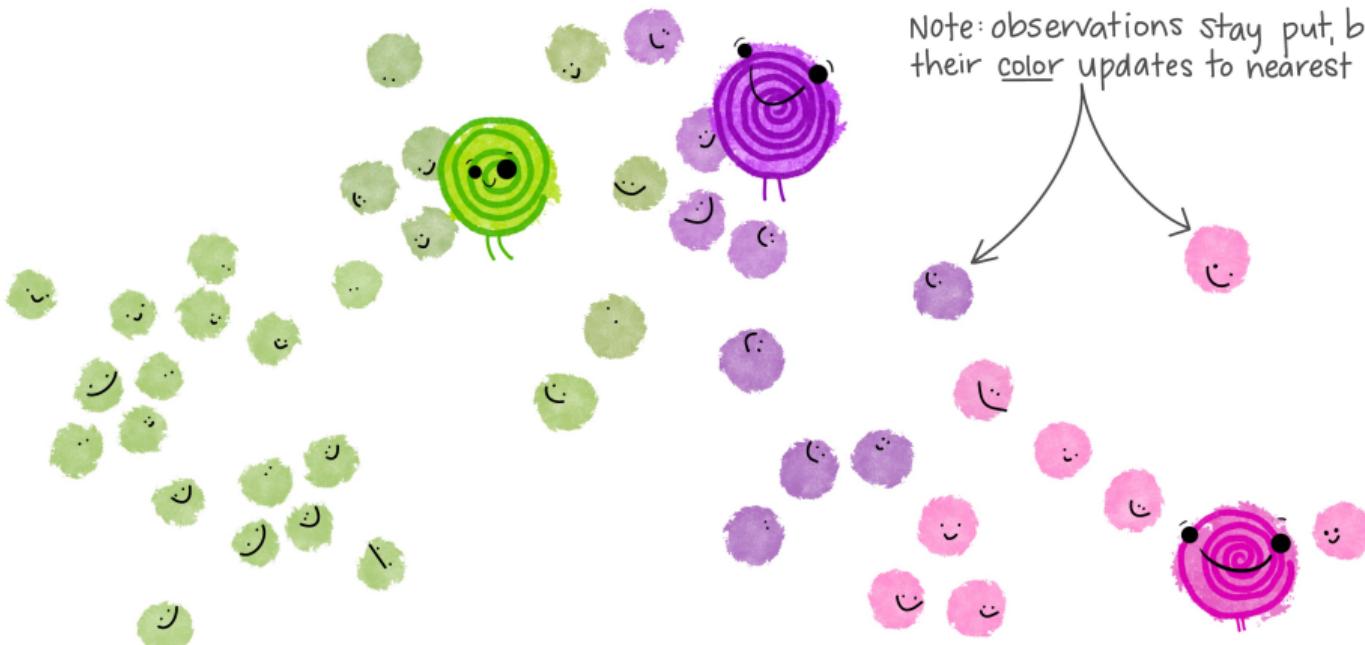
②

Those k centroids get randomly placed in your space.



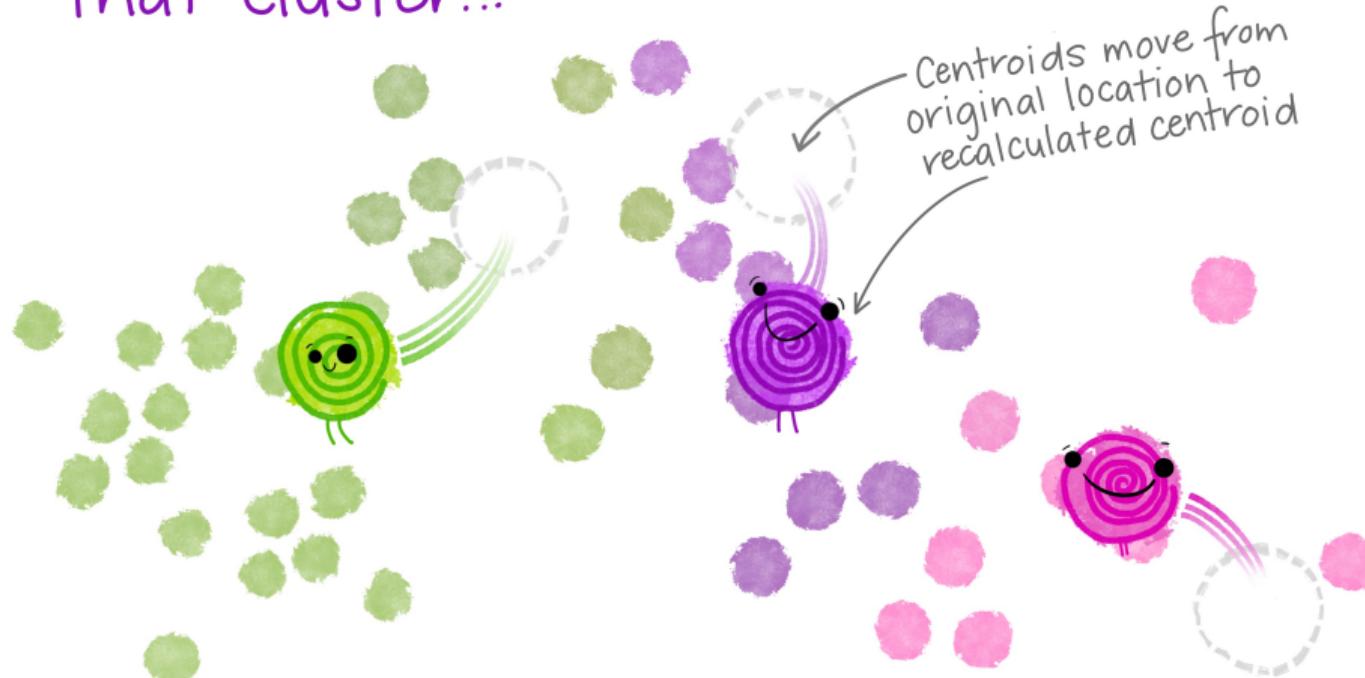
③

Each observation gets temporarily "assigned" to its closest centroid.
(e.g. by Euclidean distance)



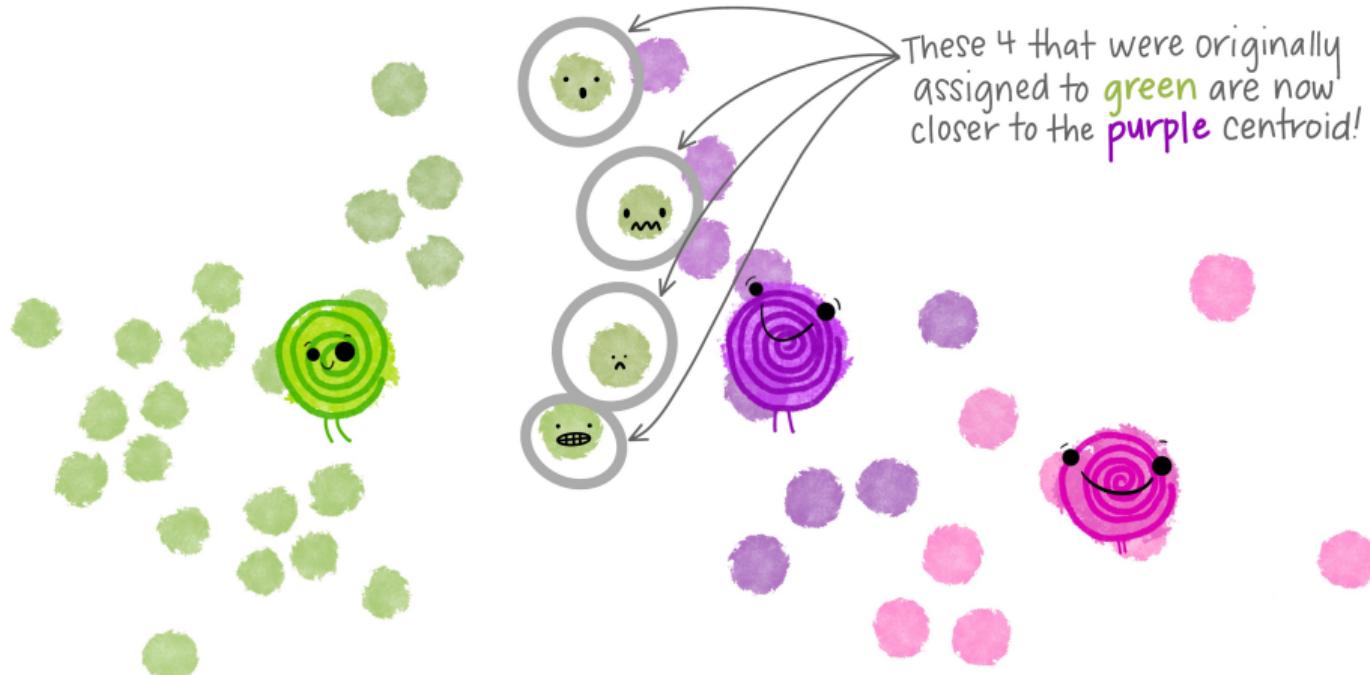
④

Then the centroid of each cluster is calculated based on all observations assigned to that cluster...





UH OH. Now that the cluster centroids have moved, some of the observations are now closer to a different centroid!

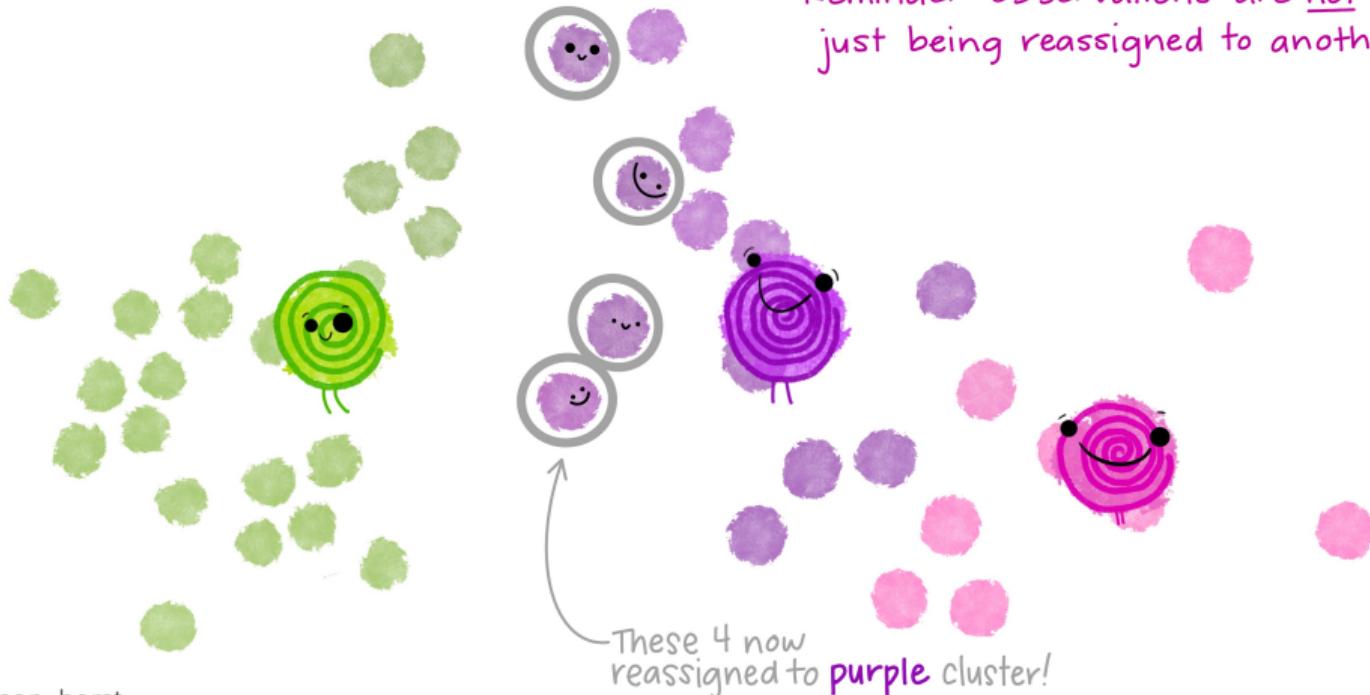




NO PROBLEM!

Observations get reassigned* to a different cluster based on the recalculated centroid.

*Reminder: observations are not moving, just being reassigned to another cluster.



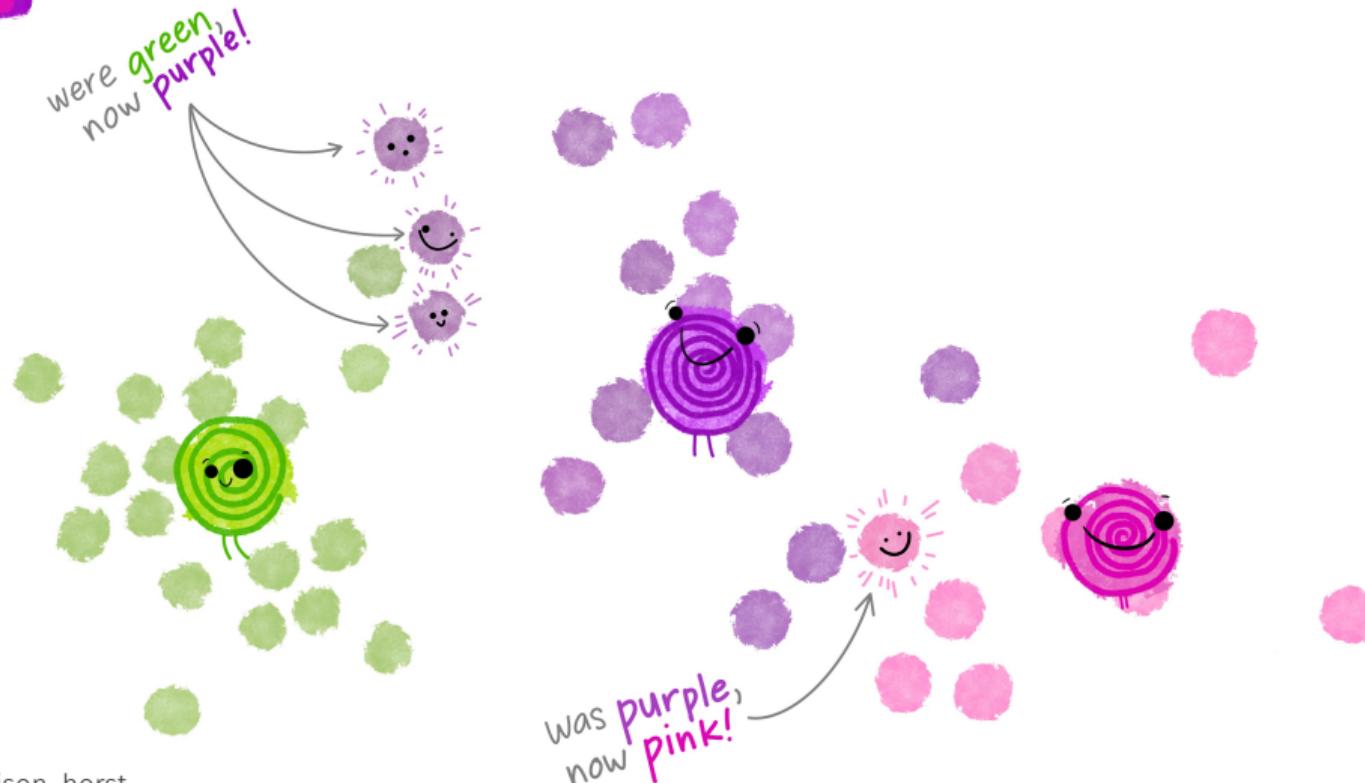
6

But now that observations have been reassigned,
the centroids need to move again [recalculate
centroids from updated clusters]



7

Again, now observations are reassigned as needed to the closest centroid.

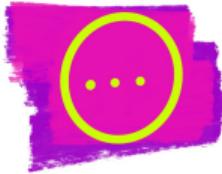




Then the centroid for each cluster
is recalculated...



...which means observations will be reassigned...



That iterative process of

Recalculate cluster centroids

↳ Reassign observations to nearest centroid

↳ Recalculate cluster centroids

↳ Reassign observations to nearest centroid

↳ Recalculate cluster centroids

↳ Reassign observations to nearest centroid



Continues until nothing is moving
or being reassigned anymore!

fin

Which means the iteration is done and each observation is assigned to its final cluster.



Ejemplo k -means con $k = 2$

En la librería `stats`, paquete base de R, tenemos la función `kmeans`

```
k <- 2
k2res <- kmeans(B, k)
str(k2res)
```

List of 9

- \$ cluster : int [1:680] 1 1 1 1 1 1 1 1 1 1 ...
- \$ centers : num [1:2, 1:6] 0.3371 1.1447 0.1176 0.5449 0.0385 ...
- ... attr(*, "dimnames")=List of 2
-\$: chr [1:2] "1" "2"
-\$: chr [1:6] "MEMORY" "LANGUAGE" "EXECUTIVE" "VISUOSPATIAL" ...
- \$ totss : num 4872
- \$ withinss : num [1:2] 1930 1082
- \$ tot.withinss: num 3012
- \$ betweenss : num 1860
- \$ size : int [1:2] 543 137
- \$ iter : int 1
- \$ ifault : int 0
- attr(*, "class")= chr "kmeans"

Ejemplo k -means con $k = 2$

En la librería `stats`, paquete base de R, tenemos la función `kmeans`

```
k <- 2
k2res <- kmeans(B, k)
str(k2res)
```

List of 9
\$ cluster : int [1:680] 1 1 1 1 1 1 1 1 1 1 ...
\$ centers : num [1:2, 1:6] 0.3371 1.1447 0.1176 0.5449 0.0385 ...
... attr(*, "dimnames")=List of 2
... ..\$: chr [1:2] "1" "2"
... ..\$: chr [1:6] "MEMORY" "LANGUAGE" "EXECUTIVE" "VISUOSPATIAL" ...
\$ totss : num 4872
\$ withinss : num [1:2] 1930 1082
\$ tot.withinss: num 3012
\$ betweenss : num 1860
\$ size : int [1:2] 543 137
\$ iter : int 1
\$ ifault : int 0
- attr(*, "class")= chr "kmeans"

Los grupos asignados a las 680 observaciones están en `k2res$cluster`

Ejemplo k -means con $k = 3$

```
k <- 3
k3res <- kmeans(B, k)
str(k3res)
```

List of 9
\$ cluster : int [1:680] 3 3 3 2 3 3 3 3 3 3 ...
\$ centers : num [1:3, 1:6] 1.154 0.767 0.183 0.544 0.583 ...
... attr(*, "dimnames")=List of 2
... ..\$: chr [1:3] "1" "2" "3"
... ..\$: chr [1:6] "MEMORY" "LANGUAGE" "EXECUTIVE" "VISUOSPATIAL" ...
\$ totss : num 4872
\$ withinss : num [1:3] 869 750 772
\$ tot.withinss: num 2390
\$ betweenss : num 2481
\$ size : int [1:3] 117 174 389
\$ iter : int 3
\$ ifault : int 0
- attr(*, "class")= chr "kmeans"

Ejemplo k -means con $k = 3$

```
k <- 3
k3res <- kmeans(B, k)
str(k3res)
```

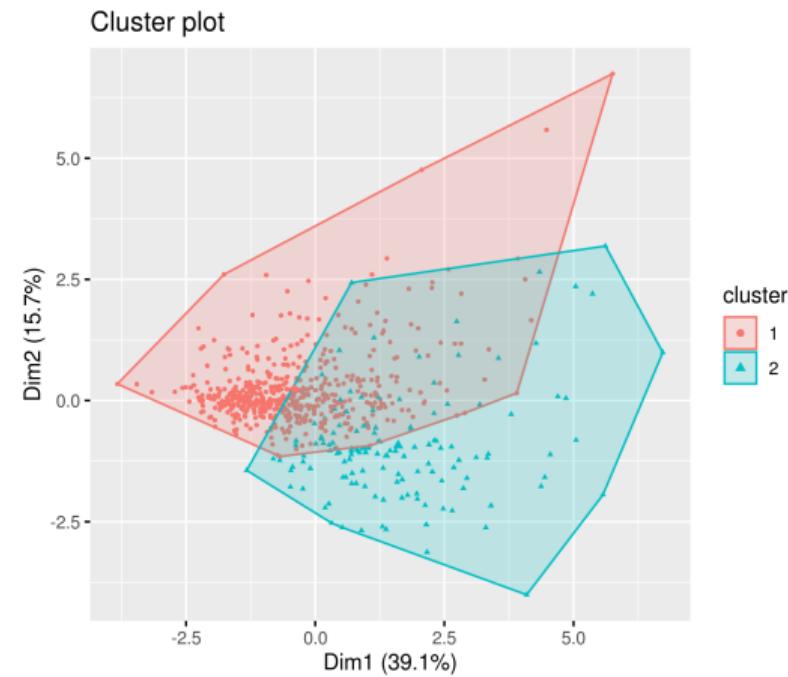
List of 9
\$ cluster : int [1:680] 3 3 3 2 3 3 3 3 3 3 ...
\$ centers : num [1:3, 1:6] 1.154 0.767 0.183 0.544 0.583 ...
... attr(*, "dimnames")=List of 2
... ..\$: chr [1:3] "1" "2" "3"
... ..\$: chr [1:6] "MEMORY" "LANGUAGE" "EXECUTIVE" "VISUOSPATIAL" ...
\$ totss : num 4872
\$ withinss : num [1:3] 869 750 772
\$ tot.withinss: num 2390
\$ betweenss : num 2481
\$ size : int [1:3] 117 174 389
\$ iter : int 3
\$ ifault : int 0
- attr(*, "class")= chr "kmeans"

Podemos comparar las dos particiones con `table(k2res$cluster, k3res$cluster)`. Los 3 sub-grupos no son necesariamente particiones de los 2 primeros sub-grupos.

¿Cómo visualizar las particiones?

La función `fviz_cluster`, de la librería `factoextra`, grafica los datos en las 2 direcciones de mayor varianza*:

```
fviz_cluster(k2res, B, geom = "point")
fviz_cluster(k3res, B, geom = "point")
k4res <- kmeans(B, 4)
fviz_cluster(k4res, B, geom = "point")
...
```

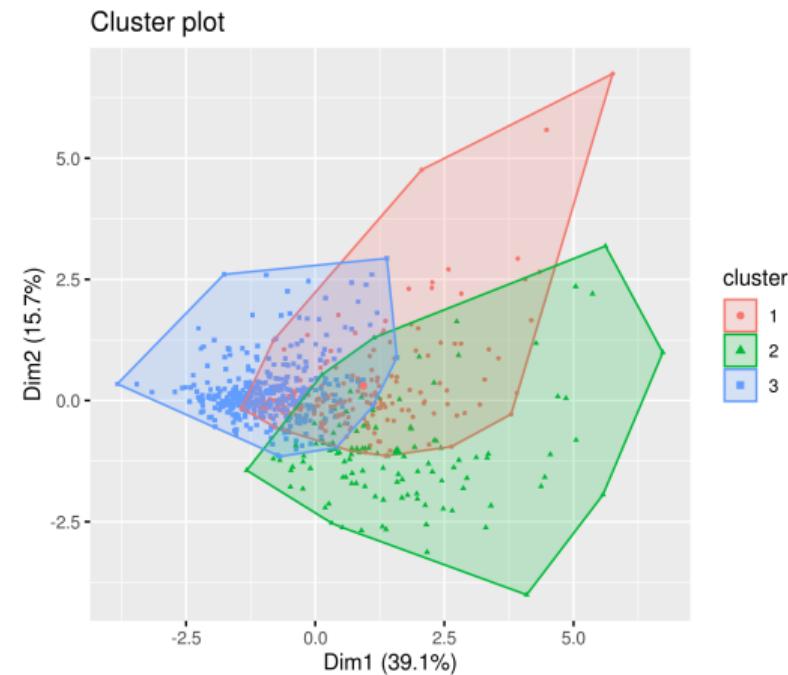


* Usando análisis de componentes principales (PCA)

¿Cómo visualizar las particiones?

La función `fviz_cluster`, de la librería `factoextra`, grafica los datos en las 2 direcciones de mayor varianza*:

```
fviz_cluster(k2res, B, geom = "point")
fviz_cluster(k3res, B, geom = "point")
k4res <- kmeans(B, 4)
fviz_cluster(k4res, B, geom = "point")
...
```

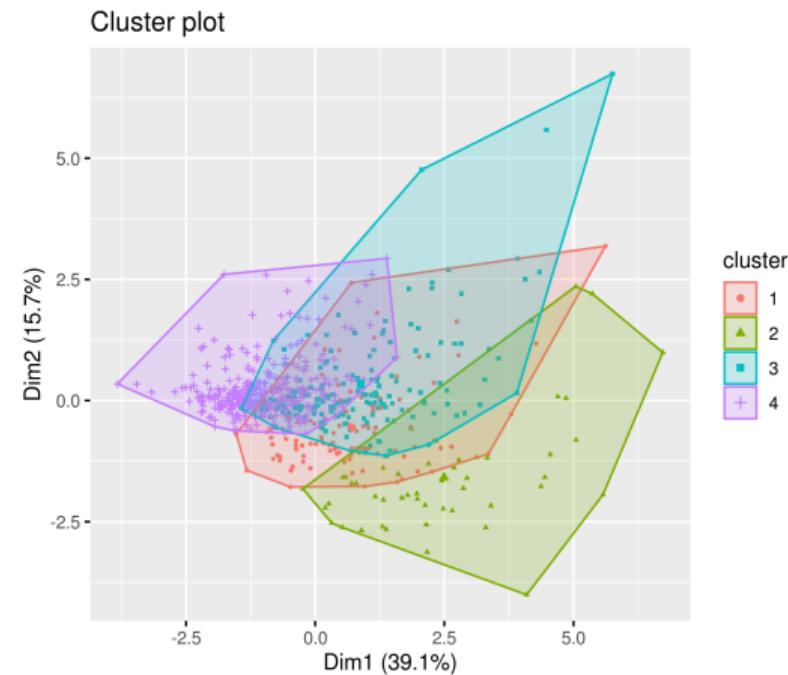


* Usando análisis de componentes principales (PCA)

¿Cómo visualizar las particiones?

La función `fviz_cluster`, de la librería `factoextra`, grafica los datos en las 2 direcciones de mayor varianza*:

```
fviz_cluster(k2res, B, geom = "point")
fviz_cluster(k3res, B, geom = "point")
k4res <- kmeans(B, 4)
fviz_cluster(k4res, B, geom = "point")
...
```

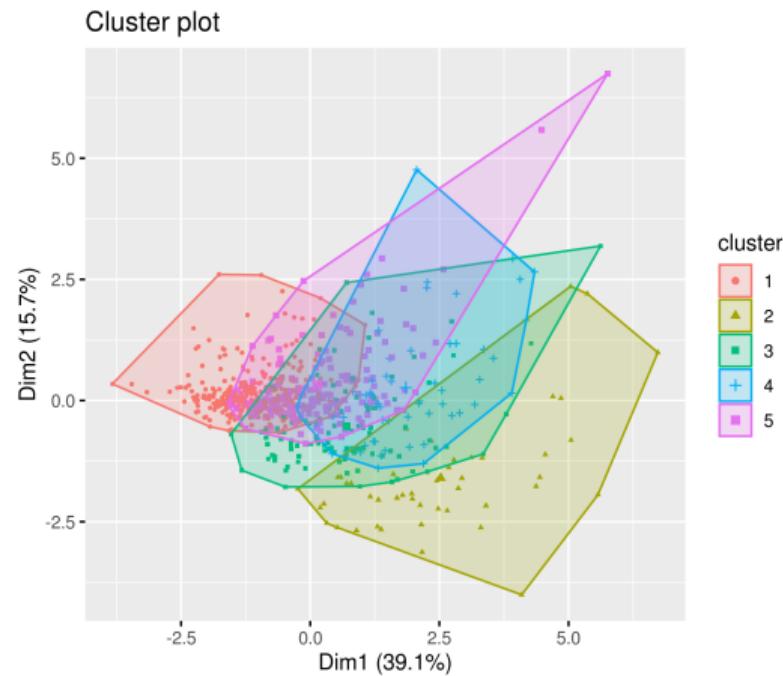


* Usando análisis de componentes principales (PCA)

¿Cómo visualizar las particiones?

La función `fviz_cluster`, de la librería `factoextra`, grafica los datos en las 2 direcciones de mayor varianza*:

```
fviz_cluster(k2res, B, geom = "point")
fviz_cluster(k3res, B, geom = "point")
k4res <- kmeans(B, 4)
fviz_cluster(k4res, B, geom = "point")
...
```



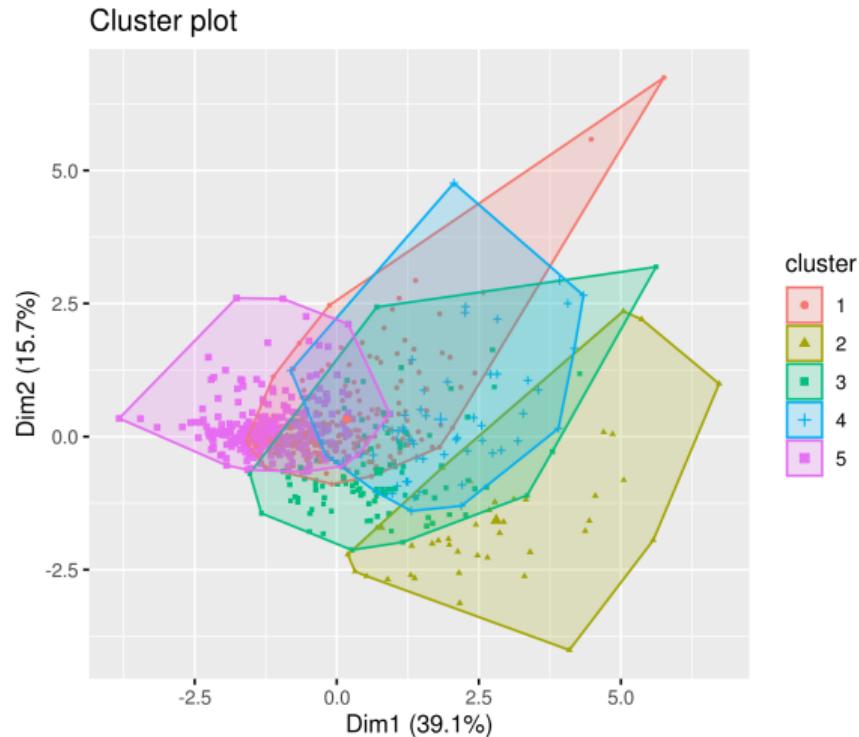
* Usando análisis de componentes principales (PCA)

Importante: muchos (¿todos?) métodos de agrupamiento tienen inicialización aleatoria → el resultado no siempre es el mismo.

Importante: muchos (¿todos?) métodos de agrupamiento tienen inicialización aleatoria → el resultado no siempre es el mismo.

```
> k <- 5  
> set.seed(1987)  
> res1 <- kmeans(B, k)  
> set.seed(2020)  
> res2 <- kmeans(B, k)  
> table(res1$cluster, res2$cluster)
```

	1	2	3	4	5
1	29	0	0	0	263
2	116	0	1	2	19
3	52	0	2	55	0
4	0	1	100	0	0
5	0	39	0	1	0



Varianza intra-grupo: suma de las distancias al centro del grupo → el resultado no siempre es el mismo.

Cuando hay más sub-grupos ($\uparrow k$), las varianzas intra-grupo disminuyen.

>

¿Cómo elegir k ?
