

# Détection de communautés

L. Rouvière et B. Thieurmél

[laurent.rouviere@univ-rennes2.fr](mailto:laurent.rouviere@univ-rennes2.fr)

[benoit.thieurmél@datastorm.fr](mailto:benoit.thieurmél@datastorm.fr)

13 juin 2017

- **Données** : individus sur lesquels on a mesuré des quantités ou défini des proximités.

## Détection de communautés

Mettre en évidence des **groupes** possédant des **caractéristiques similaires**.

## Objectifs multiples

- Identifier des profils
  - Effectuer des actions ciblées (recommandation...)
  - ...
- 
- Problème **récent**, techniques pas forcément récentes.

- **Données** : individus sur lesquels on a mesuré des quantités ou défini des proximités.

## Détection de communautés

Mettre en évidence des **groupes** possédant des **caractéristiques similaires**.

### Objectifs multiples

- Identifier des profils
  - Effectuer des actions ciblées (recommandation...)
  - ...
- 
- Problème **récent**, techniques pas forcément récentes.

- **Données** : individus sur lesquels on a mesuré des quantités ou défini des proximités.

## Détection de communautés

Mettre en évidence des **groupes** possédant des **caractéristiques similaires**.

## Objectifs multiples

- Identifier des profils
  - Effectuer des actions ciblées (recommandation...)
  - ...
- 
- Problème **récent**, techniques pas forcément récentes.

- **Données** : individus sur lesquels on a mesuré des quantités ou défini des proximités.

## Détection de communautés

Mettre en évidence des **groupes** possédant des **caractéristiques similaires**.

## Objectifs multiples

- Identifier des profils
  - Effectuer des actions ciblées (recommandation...)
  - ...
- 
- Problème **récent**, techniques pas forcément récentes.

# Un exemple jouet

- On considère  $n = 40$  utilisateurs qui ont visionné (ou pas)  $p = 30$  films ;
- Un extrait des données :

```
> donnees[1:5,1:15]
```

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
1	1	0	1	1	0	0	0	1	0	0	0	0	1	1	0
2	0	1	1	0	0	1	1	0	1	0	1	0	0	0	1
3	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
4	0	1	1	0	1	0	0	0	1	0	0	0	0	0	1
5	1	1	1	1	0	0	0	1	0	0	1	0	1	0	1

## Objectif

Extraire des communautés d'utilisateurs qui ont des goûts similaires.

# Un exemple jouet

- On considère  $n = 40$  utilisateurs qui ont visionné (ou pas)  $p = 30$  films;
- Un extrait des données :

```
> donnees[1:5,1:15]
```

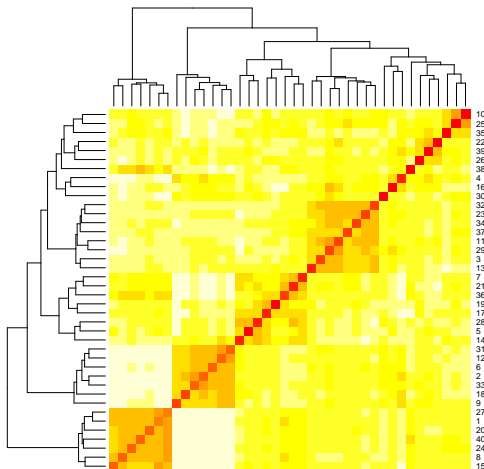
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
1	1	0	1	1	0	0	0	1	0	0	0	0	1	1	0
2	0	1	1	0	0	1	1	0	1	0	1	0	0	0	1
3	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
4	0	1	1	0	1	0	0	0	1	0	0	0	0	0	1
5	1	1	1	1	0	0	0	1	0	0	1	0	1	0	1

## Objectif

Extraire des communautés d'utilisateurs qui ont des goûts similaires.

# Similarités

- Mesurer la similarité entre utilisateurs (primordial pour l'analyse)  
> DD <- as.matrix(dist(donnees))
- Visualiser à l'aide d'un **heatmap** :  
> heatmap(DD)





- **Clustering** : trouver des groupes d'individus proches (CAH, kmeans)...  
mais "1 communauté n'est pas vraiment un cluster".

1 communauté=ensemble d'individus proches sur un sous-ensemble de variables.

- **Graphe** :
  - un nœud=un individu.
  - 1 arête=1 lien entre deux individus.
  - $\implies$  1 communauté=1 sous-ensemble de nœuds où on a "beaucoup" d'arêtes.

- **Clustering** : trouver des groupes d'individus proches (CAH, kmeans)...  
mais "1 communauté n'est pas vraiment un cluster".

1 communauté=ensemble d'individus proches sur un sous-ensemble de variables.

- **Graphe** :
  - un nœud=un individu.
  - 1 arête=1 lien entre deux individus.
  - $\implies$  1 communauté=1 sous-ensemble de nœuds où on a "beaucoup" d'arêtes.

- **Clustering** : trouver des groupes d'individus proches (CAH, kmeans)...  
mais "1 communauté n'est pas vraiment un cluster".

1 communauté=ensemble d'individus proches sur un sous-ensemble de variables.

- **Graphe** :
  - un nœud=un individu.
  - 1 arête=1 lien entre deux individus.
  - $\implies$  1 communauté=1 sous-ensemble de nœuds où on a "beaucoup" d'arêtes.

- **Clustering** : trouver des groupes d'individus proches (CAH, kmeans)...  
mais "1 communauté n'est pas vraiment un cluster".

1 communauté=ensemble d'individus proches sur un sous-ensemble de variables.

- **Graphe** :
  - un nœud=un individu.
  - 1 arête=1 lien entre deux individus.
  - $\implies$  1 communauté=1 sous-ensemble de nœuds où on a "beaucoup" d'arêtes.

- **Clustering** : trouver des groupes d'individus proches (CAH, kmeans)...  
mais "1 communauté n'est pas vraiment un cluster".

1 communauté=ensemble d'individus proches sur un sous-ensemble de variables.

- **Graphe** :
  - un nœud=un individu.
  - 1 arête=1 lien entre deux individus.
  - $\implies$  1 communauté=1 sous-ensemble de nœuds où on a "beaucoup" d'arêtes.

- **Clustering** : trouver des groupes d'individus proches (CAH, kmeans)...  
mais "1 communauté n'est pas vraiment un cluster".

1 communauté=ensemble d'individus proches sur un sous-ensemble de variables.

- **Graphe** :
  - un nœud=un individu.
  - 1 arête=1 lien entre deux individus.
  - $\implies$  1 communauté=1 sous-ensemble de nœuds où on a "beaucoup" d'arêtes.

1 Approche clustering

2 Approche graphe

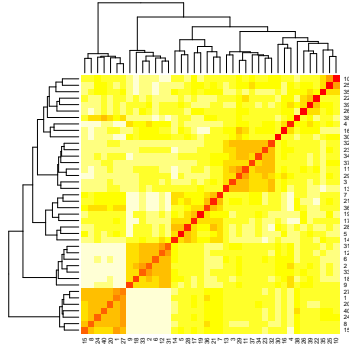
3 Le projet Edenred

1 Approche clustering

2 Approche graphe

3 Le projet Edenred





- De nombreuses **méthodes de clustering** permettent de définir des **cluster** à partir d'une **matrice de similarité**.
- 1 communauté  $\neq$  1 cluster.
- 1 communauté = ensemble d'individus très proche vis-vis d'**un petit groupe de variables**.
- 1 individu peut **appartenir à plusieurs communautés**.

- 1 **bicluster** = 1 sous-ensemble d'individus et de variables.

[Kaiser, 2011]

A typical situation to calculate bicluster are a **high dimensional** dataset with many variables, so that normal cluster algorithms lead to **diffuse results** due to many uncorrelated variables. Also **biclustering** is useful if there is a assumed connection of objects and some of the variables in the dataset, e.g. some objects have 'similar' patterns for a given set of variables.

Objectif

Détecter des pattern **locaux**.

- 1 **bicluster** = 1 sous-ensemble d'individus et de variables.

[Kaiser, 2011]

A typical situation to calculate bicluster are a **high dimensional** dataset with many variables, so that normal cluster algorithms lead to **diffuse results** due to many uncorrelated variables. Also **biclustering** is useful if there is a assumed connection of objects and some of the variables in the dataset, e.g. some objects have 'similar' patterns for a given set of variables.

## Objectif

Détecter des pattern **locaux**.

- Les données :

$$A = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$$

- **Objectif** : Trouver des sous-groupes  $A_{IJ}$  tels que les individus dans  $I = \{i_1, \dots, i_k\}$  sont proches pour le groupe de variables  $J = \{j_1, \dots, j_\ell\}$ .
- $BC = A_{IJ} = \{I, J\}$ .

Trouver un bicluster revient à trouver un sous-ensemble des lignes et des colonnes de  $A$ .

- Les données :

$$A = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$$

- **Objectif** : Trouver des sous-groupes  $A_{IJ}$  tels que les individus dans  $I = \{i_1, \dots, i_k\}$  sont proches pour le groupe de variables  $J = \{j_1, \dots, j_\ell\}$ .
- $BC = A_{IJ} = \{I, J\}$ .

Trouver un bicluster revient à trouver un sous-ensemble des lignes et des colonnes de  $A$ .

- Les données :

$$A = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$$

- **Objectif** : Trouver des sous-groupes  $A_{IJ}$  tels que les individus dans  $I = \{i_1, \dots, i_k\}$  sont proches pour le groupe de variables  $J = \{j_1, \dots, j_\ell\}$ .
- $BC = A_{IJ} = \{I, J\}$ .

Trouver un bicluster revient à trouver un sous-ensemble des lignes et des colonnes de  $A$ .

- Les données :

$$A = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$$

- **Objectif** : Trouver des sous-groupes  $A_{IJ}$  tels que les individus dans  $I = \{i_1, \dots, i_k\}$  sont proches pour le groupe de variables  $J = \{j_1, \dots, j_\ell\}$ .
- $BC = A_{IJ} = \{I, J\}$ .

Trouver un bicluster revient à trouver un sous-ensemble des lignes et des colonnes de  $A$ .

# Différents types de bicluster

- En fonction de la nature des données (et surtout du problème), on considèrera différents types de bicluster.
- Valeurs constantes : partout, ligne, colonnes

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

- Valeurs constantes additives ou multiplicatives

$$\begin{pmatrix} 1 & 2 & 5 & 0 \\ 2 & 3 & 6 & 1 \\ 4 & 5 & 8 & 3 \\ 5 & 6 & 9 & 4 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 0.5 & 1.5 \\ 2 & 4 & 1 & 3 \\ 4 & 8 & 2 & 6 \\ 3 & 6 & 1.5 & 4.5 \end{pmatrix}$$



# Différents types de bicluster

- En fonction de la nature des données (et surtout du problème), on considèrera différents types de bicluster.
- Valeurs constantes : partout, ligne, colonnes

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

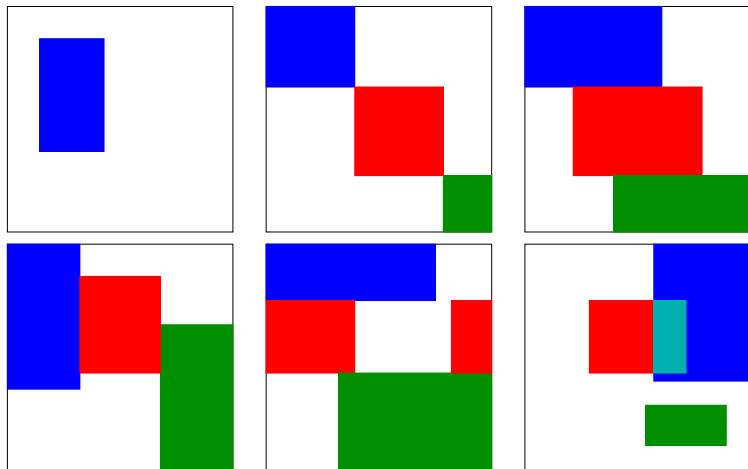
$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

- Valeurs constantes additives ou multiplicatives

$$\begin{pmatrix} 1 & 2 & 5 & 0 \\ 2 & 3 & 6 & 1 \\ 4 & 5 & 8 & 3 \\ 5 & 6 & 9 & 4 \end{pmatrix}$$

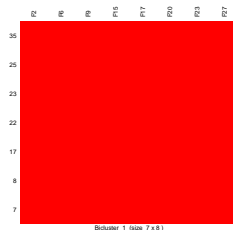
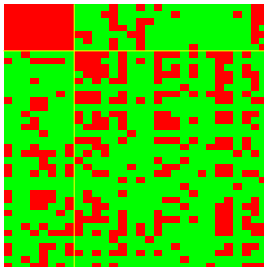
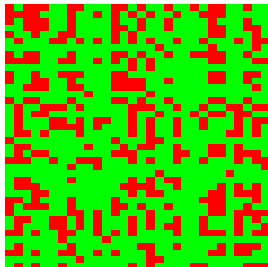
$$\begin{pmatrix} 1 & 2 & 0.5 & 1.5 \\ 2 & 4 & 1 & 3 \\ 4 & 8 & 2 & 6 \\ 3 & 6 & 1.5 & 4.5 \end{pmatrix}$$

# Différentes structures de bicluster



# Notre cas

- On s'intéresse ici à une **matrice binaire** (exemple des films).
- Un **biclusteur** : une sous-matrice de dimension (au moins)  $minr \times minc$  qui ne contient que des 1.

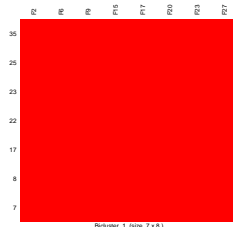
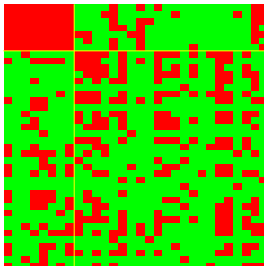
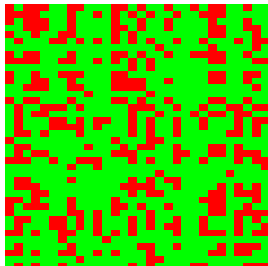


Bimax [Prelic et al., 2006]

Algorithme qui permet d'identifier des sous-groupes de 1 dans la matrice

# Notre cas

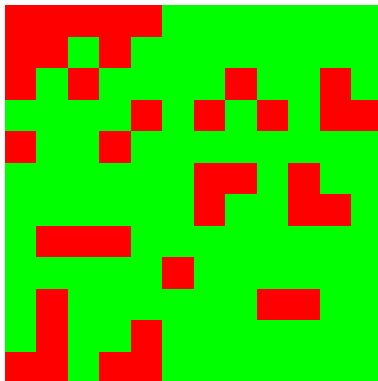
- On s'intéresse ici à une **matrice binaire** (exemple des films).
- Un **biclust** : une sous-matrice de dimension (au moins)  $minr \times minc$  qui ne contient que des 1.



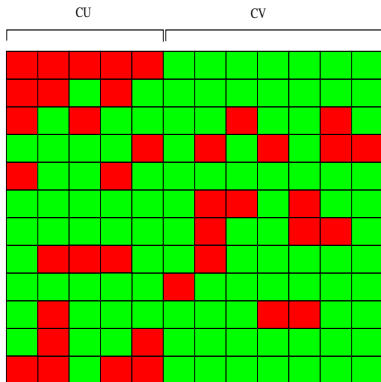
Bimax [Prelic et al., 2006]

Algorithme qui permet d'identifier des sous-groupes de 1 dans la matrice

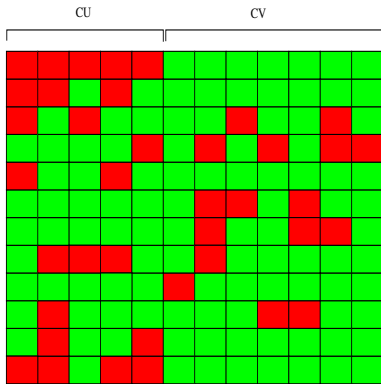
- **Idée** : Partitionner la matrice en 3 sous-matrices dont une ne contient que des 0 (et sera éliminée à l'étape suivante).



- ① On fixe  $minr$  et  $minc$ .
- ② Choix d'une ligne  $i^*$  au hasard (qui contient un nombre  $minc$  suffisant de 1).
- ③ Diviser les colonnes en deux groupes  $CU$  et  $CV$  tels que
  - $CU = \{j : A[i^*, j] = 1\}$  et  $CV = \{1, \dots, j\} - CU$ .

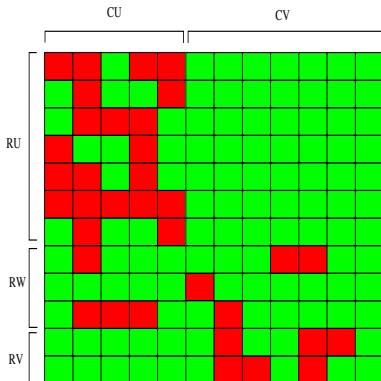


- ① On fixe  $minr$  et  $minc$ .
- ② Choix d'une ligne  $i^*$  au hasard (qui contient un nombre  $minc$  suffisant de 1).
- ③ Diviser les colonnes en deux groupes  $CU$  et  $CV$  tels que
  - $CU = \{j : A[i^*, j] = 1\}$  et  $CV = \{1, \dots, j\} - CU$ .



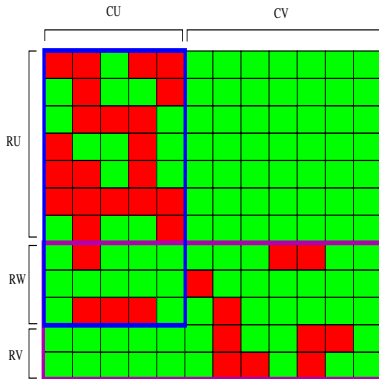
② Diviser les lignes en 3 ensembles :

- $RU = \{i : \exists j \in CU A[i,j] = 1 \text{ et } A[i,j] = 0 \forall j \in CV\}$
- $RV = \{i : \exists j \in CV A[i,j] = 1 \text{ et } A[i,j] = 0 \forall j \in CU\}$
- $RW$  les autres



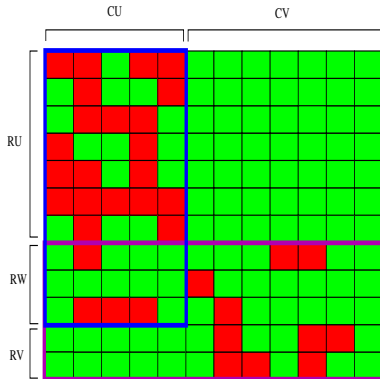


- ③ On pose :  $U = [RU \cup RW, CU]$  et  $V = [RV \cup RW, CU \cup CV]$



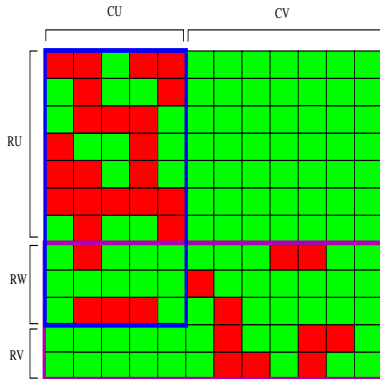
- ④ On itère le process sur  $U$  et  $V$  en ajoutant une contrainte sur les bicluster de  $V$  (ils doivent dépendre de  $CV$ ).
- ⑤ **Sortie** : les matrices de taille minimales  $minr$  et  $minc$  qui ne contiennent que des 1.

- ③ On pose :  $U = [RU \cup RW, CU]$  et  $V = [RV \cup RW, CU \cup CV]$



- ④ On itère le process sur  $U$  et  $V$  en ajoutant une contrainte sur les bicluster de  $V$  (ils doivent dépendre de  $CV$ ).
- ⑤ **Sortie** : les matrices de taille minimales *minr* et *minc* qui ne contiennent que des 1.

- ③ On pose :  $U = [RU \cup RW, CU]$  et  $V = [RV \cup RW, CU \cup CV]$



- ④ On itère le process sur  $U$  et  $V$  en ajoutant une contrainte sur les bicluster de  $V$  (ils doivent dépendre de  $CV$ ).
- ⑤ **Sortie** : les matrices de taille minimales  $minr$  et  $minc$  qui ne contiennent que des 1.

# Le package biclust

```
> library(biclust)
> biclust(x, method=BCBimax(), minr=2, minc=2, number=100)
```

## Paramètres

$\text{minr} \searrow, \text{minc} \searrow \implies \text{nb de bicluster} \nearrow$ .

```
> biclust(A,method=BCBimax(),minr=2,minc=2)
```

An object of class Biclust

call:

```
biclust(x = A, method = BCBimax(), minr = 2, minc = 2)
```

Number of Clusters found: 9

First 5 Cluster sizes:

	BC 1	BC 2	BC 3	BC 4	BC 5
Number of Rows:	3	4	4	3	2
Number of Columns:	2	2	2	3	4

```
> biclust(A,method=BCBimax(),minr=3,minc=3)
```

An object of class Biclust

call:

```
biclust(x = A, method = BCBimax(), minr = 3, minc = 3)
```

There was one cluster found with  
3 Rows and 3 columns

```
> biclust(A,method=BCBimax(),minr=4,minc=4)
```

An object of class Biclust

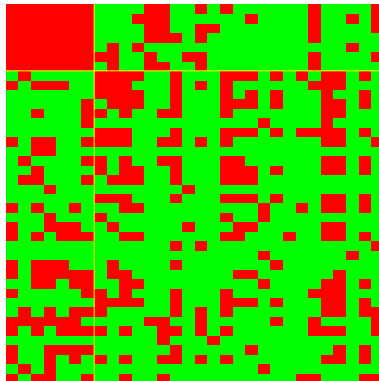
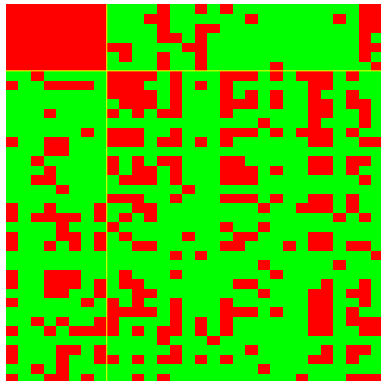
call:

```
biclust(x = A, method = BCBimax(), minr = 4, minc = 4)
```

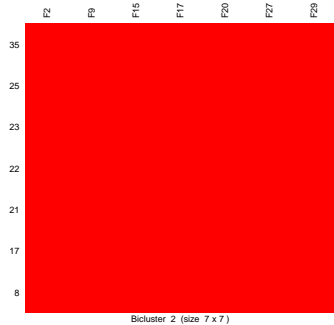
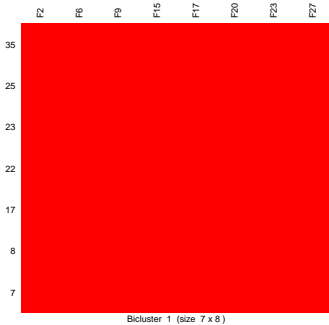
There was no cluster found

# Visualisation

```
> res1 <- biclust(donnees,method=BCBimax(),minr=7,minc=7)
> res1
Number of Clusters found: 5
First 5 Cluster sizes:
              BC 1 BC 2 BC 3 BC 4 BC 5
Number of Rows:      7   7   7   7   7
Number of Columns:    8   7   7   7   7
> drawHeatmap2(donnees,res1,1)
> drawHeatmap2(donnees,res1,2)
```

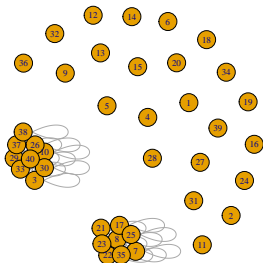


```
> drawHeatmap(donnees,res1,1)
> drawHeatmap(donnees,res1,2)
```



- **Avantages** : procédure simple qui permet de trouver des petits groupes de personnes aux goûts très proches (sur une partie des variables).
- **Inconvénient** : structure de bicluster **pas toujours pertinentes**, beaucoup de bicluster vont être très proches.

```
> BICLUST
[[1]]
[1] 7 8 17 22 23 25 35
[[2]]
[1] 8 17 21 22 23 25 35
[[3]]
[1] 3 10 26 29 30 33 40
[[4]]
[1] 10 26 29 30 33 37 40
[[5]]
[1] 10 26 29 33 37 38 40
```

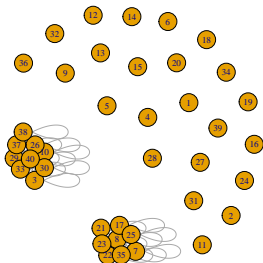


Etape préliminaire à une étude de **graphe**.



- **Avantages** : procédure simple qui permet de trouver des petits groupes de personnes aux goûts très proches (sur une partie des variables).
- **Inconvénient** : structure de bicluster **pas toujours pertinentes**, beaucoup de bicluster vont être très proches.

```
> BICLUST
[[1]]
[1] 7 8 17 22 23 25 35
[[2]]
[1] 8 17 21 22 23 25 35
[[3]]
[1] 3 10 26 29 30 33 40
[[4]]
[1] 10 26 29 30 33 37 40
[[5]]
[1] 10 26 29 33 37 38 40
```



Etape préliminaire à une étude de **graphe**.

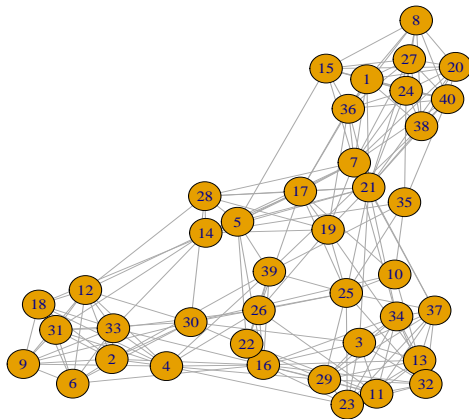
1 Approche clustering

2 Approche graphe

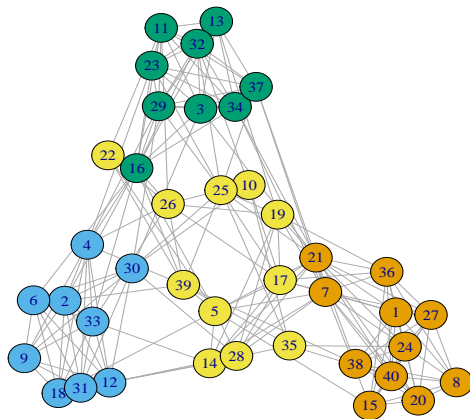
3 Le projet Edenred

# Objectif

- Obtenir une **structure de graphe** :



- Sur laquelle on puisse déduire des communautés.



# Construction du graphe

- Le graphe se construit souvent à l'aide d'une **matrice d'adjacence**  $A$  définie à partir de la matrice de similarités  $S$  ;

## Graphe par plus proches voisins

- G-ppv** :  $A_{i,j} = 1$  si  $X_j$  est parmi les  $k$ -ppv de  $X_i$ , sinon  $A_{i,j} = 0$  ;
- G-ppv mutuel** :  $A_{i,j} = 1$  si  $X_j$  est parmi les  $k$ -ppv de  $X_i$  et  $X_i$  parmi les  $k$ -ppv de  $X_j$ , sinon  $A_{i,j} = 0$

## Choix de $k$

$k$  est bien entendu un **paramètre à calibrer** :  $k$  grand  $\implies$  beaucoup d'individus liés et réciproquement si  $k$  est petit.

# Construction du graphe

- Le graphe se construit souvent à l'aide d'une **matrice d'adjacence**  $A$  définie à partir de la matrice de similarités  $S$  ;

## Graphe par plus proches voisins

- G-ppv** :  $A_{i,j} = 1$  si  $X_j$  est parmi les  $k$ -ppv de  $X_i$ , sinon  $A_{i,j} = 0$  ;
- G-ppv mutuel** :  $A_{i,j} = 1$  si  $X_j$  est parmi les  $k$ -ppv de  $X_i$  et  $X_i$  parmi les  $k$ -ppv de  $X_j$ , sinon  $A_{i,j} = 0$

## Choix de $k$

$k$  est bien entendu un **paramètre à calibrer** :  $k$  grand  $\implies$  beaucoup d'individus liés et réciproquement si  $k$  est petit.

# Construction du graphe

- Le graphe se construit souvent à l'aide d'une **matrice d'adjacence**  $A$  définie à partir de la matrice de similarités  $S$  ;

## Graphe par plus proches voisins

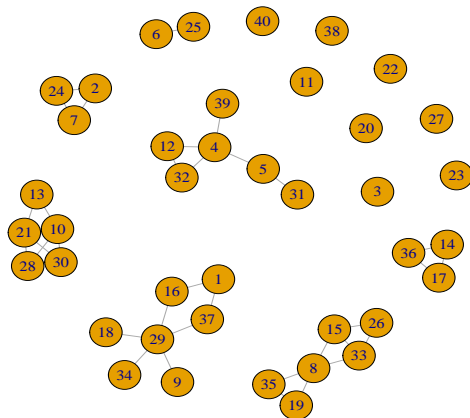
- G-ppv** :  $A_{i,j} = 1$  si  $X_j$  est parmi les  $k$ -ppv de  $X_i$ , sinon  $A_{i,j} = 0$  ;
- G-ppv mutuel** :  $A_{i,j} = 1$  si  $X_j$  est parmi les  $k$ -ppv de  $X_i$  et  $X_i$  parmi les  $k$ -ppv de  $X_j$ , sinon  $A_{i,j} = 0$

## Choix de $k$

$k$  est bien entendu un **paramètre à calibrer** :  $k$  grand  $\implies$  beaucoup d'individus liés et réciproquement si  $k$  est petit.

# Exemple

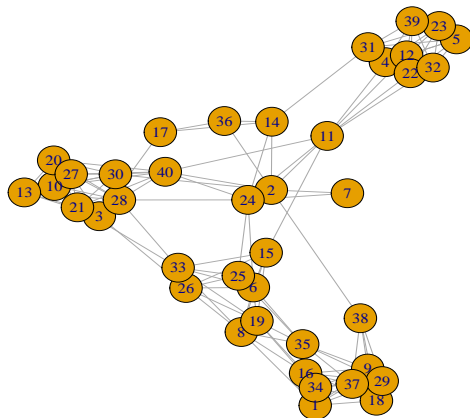
```
> gg1 <- nng(dx=DD,k=2,mutual=TRUE)  
> plot(gg1)
```





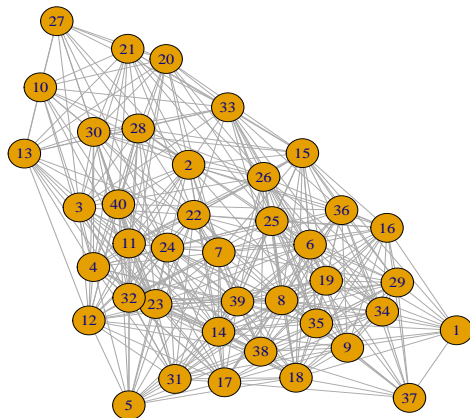
# Exemple

```
> gg2 <- nng(dx=DD,k=8,mutual=TRUE)  
> plot(gg2)
```



# Exemple

```
> gg3 <- nng(dx=DD,k=20,mutual=TRUE)  
> plot(gg3)
```



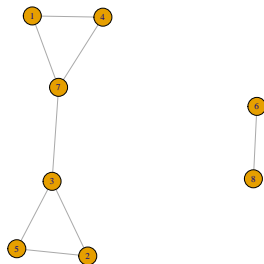
- On se restreint à l'étude des graphes **non orientés**.
- $G(V, E)$  un **graphe**,  $V$  est l'ensemble des **nœuds** de cardinal  $n$ ,  $E$  l'ensemble des **arêtes** de cardinal  $m$ .
- **Degré de centralité d'un nœud  $v_i$**  : nombre d'arêtes qui lui est associé (ou nombre de voisins adjacents) :  $d_i = \sum_{j=1}^n A_{ij}$  où  $A$  désigne la matrice d'adjacence associée au graphe.

- On se restreint à l'étude des graphes **non orientés**.
- $G(V, E)$  un **graphe**,  $V$  est l'ensemble des **nœuds** de cardinal  $n$ ,  $E$  l'ensemble des **arêtes** de cardinal  $m$ .
- **Degré de centralité d'un nœud  $v_i$**  : nombre d'arêtes qui lui est associé (ou nombre de voisins adjacents) :  $d_i = \sum_{j=1}^n A_{ij}$  où  $A$  désigne la matrice d'adjacence associée au graphe.

- On se restreint à l'étude des graphes **non orientés**.
- $G(V, E)$  un **graphe**,  $V$  est l'ensemble des **nœuds** de cardinal  $n$ ,  $E$  l'ensemble des **arêtes** de cardinal  $m$ .
- **Degré de centralité d'un nœud  $v_i$**  : nombre d'arêtes qui lui est associé (ou nombre de voisins adjacents) :  $d_i = \sum_{j=1}^n A_{ij}$  où  $A$  désigne la matrice d'adjacence associée au graphe.

# Exemple

```
> A
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    0    0    0    1    0    0    1    0
[2,]    0    0    1    0    1    0    0    0
[3,]    0    1    0    0    1    0    1    0
[4,]    1    0    0    0    0    0    1    0
[5,]    0    1    1    0    0    0    0    0
[6,]    0    0    0    0    0    0    0    1
[7,]    1    0    1    1    0    0    0    0
[8,]    0    0    0    0    0    1    0    0
> g <- graph_from_adjacency_matrix(A,mode="undirected")
> plot(g)
```



- $n = 8; m = 8$
- $d_1 = 2, d_2 = 2, \dots, d_6 = 1 \dots$

- Détecter des communautés revient à partitionner l'ensemble des nœuds  $\{v_1, \dots, v_n\}$  en  $K$  groupes (communautés) que l'on notera  $\mathcal{C}_1, \dots, \mathcal{C}_K$ .
- Idéal :
  - beaucoup de connections entre les nœuds d'une même communauté ;
  - peu de connections entre les nœuds de communautés différentes.

Nécessité de se donner un critère qui permette de mesurer la performance d'une partition.

- Détecter des communautés revient à partitionner l'ensemble des nœuds  $\{v_1, \dots, v_n\}$  en  $K$  groupes (communautés) que l'on notera  $\mathcal{C}_1, \dots, \mathcal{C}_K$ .
- Idéal :
  - beaucoup de connections entre les nœuds d'une même communauté ;
  - peu de connections entre les nœuds de communautés différentes.

Nécessité de se donner un critère qui permette de mesurer la performance d'une partition.



- Détecter des communautés revient à partitionner l'ensemble des nœuds  $\{v_1, \dots, v_n\}$  en  $K$  groupes (communautés) que l'on notera  $\mathcal{C}_1, \dots, \mathcal{C}_K$ .
- Idéal :
  - beaucoup de connections entre les nœuds d'une même communauté ;
  - peu de connections entre les nœuds de communautés différentes.

Nécessité de se donner un critère qui permette de mesurer la performance d'une partition.

- Détecter des communautés revient à partitionner l'ensemble des nœuds  $\{v_1, \dots, v_n\}$  en  $K$  groupes (communautés) que l'on notera  $\mathcal{C}_1, \dots, \mathcal{C}_K$ .
- Idéal :
  - beaucoup de connections entre les nœuds d'une même communauté ;
  - peu de connections entre les nœuds de communautés différentes.

Nécessité de se donner un critère qui permette de mesurer la performance d'une partition.

- Il s'agit d'un des critères **les plus utilisés** pour mesurer la performance de communautés.
- **L'idée** : comparer la performance de la partition sur le **graphe** à sa performance sur un **graphe "aléatoire"** (graphe dans lequel les arêtes seraient distribuées au "hasard").

- Il s'agit d'un des critères **les plus utilisés** pour mesurer la performance de communautés.
- **L'idée** : comparer la performance de la partition sur le **graphe** à sa performance sur un **graphe "aléatoire"** (graphe dans lequel les arêtes seraient distribuées au "hasard").

- Soit  $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  une partition des nœuds de  $G(V, E)$ .
- **Modularité** de  $\mathcal{C}$  :

$$\mathcal{M}(\mathcal{C}) = \frac{1}{2m} \sum_{1 \leq i, j \leq n} (A_{ij} - P_{ij}) \delta(\mathcal{C}(v_i), \mathcal{C}(v_j))$$

où

- $\delta(\mathcal{C}(v_i), \mathcal{C}(v_j)) = 1$  si  $v_i$  et  $v_j$  sont dans le même élément de la partition, 0 sinon
- $P_{ij}$  représente l'**espérance du nombre d'arêtes** entre  $v_i$  et  $v_j$  sous le modèle nul (graphe aléatoire).

## Interprétation

- $-1 \leq \mathcal{M}(\mathcal{C}) \leq 1$  ;
- $\mathcal{M}(\mathcal{C}) \nearrow$  **plus d'arêtes** dans les communautés que le modèle nul (**bonnes communautés**) et réciproquement lorsque  $\mathcal{M}(\mathcal{C}) \searrow$ .

- Soit  $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  une partition des nœuds de  $G(V, E)$ .
- **Modularité** de  $\mathcal{C}$  :

$$\mathcal{M}(\mathcal{C}) = \frac{1}{2m} \sum_{1 \leq i, j \leq n} (A_{ij} - P_{ij}) \delta(\mathcal{C}(v_i), \mathcal{C}(v_j))$$

où

- $\delta(\mathcal{C}(v_i), \mathcal{C}(v_j)) = 1$  si  $v_i$  et  $v_j$  sont dans le même élément de la partition, 0 sinon
- $P_{ij}$  représente l'**espérance du nombre d'arêtes** entre  $v_i$  et  $v_j$  sous le modèle nul (graphe aléatoire).

## Interprétation

- $-1 \leq \mathcal{M}(\mathcal{C}) \leq 1$  ;
- $\mathcal{M}(\mathcal{C}) \nearrow$  **plus d'arêtes** dans les communautés que le modèle nul (**bonnes communautés**) et réciproquement lorsque  $\mathcal{M}(\mathcal{C}) \searrow$ .

# Le modèle nul

- Il peut être spécifier de **plusieurs façons** (voir [Fortunato, 2010]).
- **Première approche** : les  $m$  arêtes sont **distribuées uniformément** entre les paires de nœuds :

$$P_{ij} = \frac{2m}{n(n-1)}, \quad 1 \leq i, j \leq n.$$

- **Seconde approche** : générer aléatoirement les arêtes en conservant les degrés de centralité des nœuds :

$$P_{ij} = \frac{d_i d_j}{2m}, \quad 1 \leq i, j \leq n.$$

On a alors

$$\mathcal{M}(C) = \frac{1}{2m} \sum_{1 \leq i, j \leq n} \left( A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C(v_i), C(v_j))$$

- Il peut être spécifier de **plusieurs façons** (voir [Fortunato, 2010]).
- **Première approche** : les  $m$  arêtes sont **distribuées uniformément** entre les paires de nœuds :

$$P_{ij} = \frac{2m}{n(n-1)}, \quad 1 \leq i, j \leq n.$$

- **Seconde approche** : générer aléatoirement les arêtes en conservant les degrés de centralité des nœuds :

$$P_{ij} = \frac{d_i d_j}{2m}, \quad 1 \leq i, j \leq n.$$

On a alors

$$\mathcal{M}(C) = \frac{1}{2m} \sum_{1 \leq i, j \leq n} \left( A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C(v_i), C(v_j))$$



- Il peut être spécifier de **plusieurs façons** (voir [Fortunato, 2010]).
- **Première approche** : les  $m$  arêtes sont **distribuées uniformément** entre les paires de nœuds :

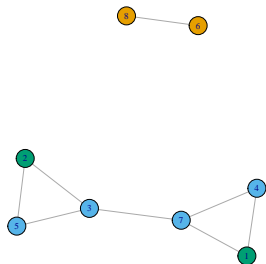
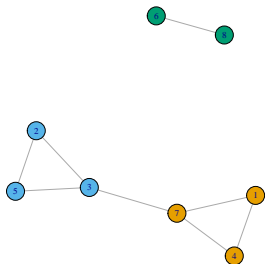
$$P_{ij} = \frac{2m}{n(n-1)}, \quad 1 \leq i, j \leq n.$$

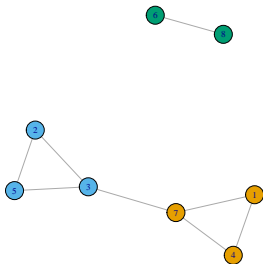
- **Seconde approche** : générer aléatoirement les arêtes en conservant les degrés de centralité des nœuds :

$$P_{ij} = \frac{d_i d_j}{2m}, \quad 1 \leq i, j \leq n.$$

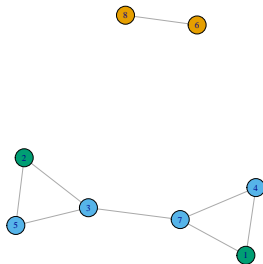
On a alors

$$\mathcal{M}(\mathcal{C}) = \frac{1}{2m} \sum_{1 \leq i, j \leq n} \left( A_{ij} - \frac{d_i d_j}{2m} \right) \delta(\mathcal{C}(v_i), \mathcal{C}(v_j))$$





```
> modularity(g,c11)
[1] 0.4765625
```

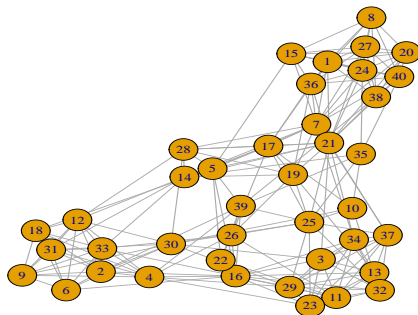


```
> modularity(g,c12)
[1] 0.03125
```

- ① Calculer la modularité pour l'ensemble des partitions ;
- ② Choisir la partition pour laquelle la modularité est maximale.

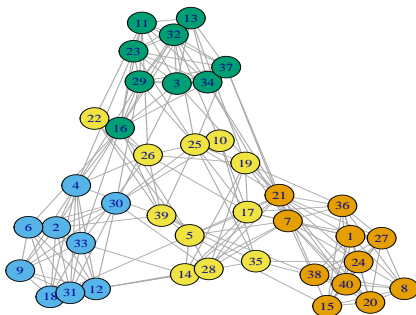
# Approche exhaustive

- 1 Calculer la modularité pour l'ensemble des partitions ;
- 2 Choisir la partition pour la quelle la modularité est maximale.



# Approche exhaustive

- 1 Calculer la modularité pour l'ensemble des partitions ;
- 2 Choisir la partition pour la quelle la modularité est maximale.



```
> cl <- cluster_optimal(gg)
> V(gg)$color <- membership(cl)
> perm_ligne <- sample(ntot)
```

- L'approche exhaustive ne peut être utilisée que sur des **petits graphes** (problème NP hard).
- Il existe plusieurs algorithmes alternatifs permettant de tomber sur des **extrema locaux** de modularité.
- Ces approches sont le plus souvent construites de manière récursive, certaines fournissent des suites de **partitions emboîtées** (même principe que la CAH).

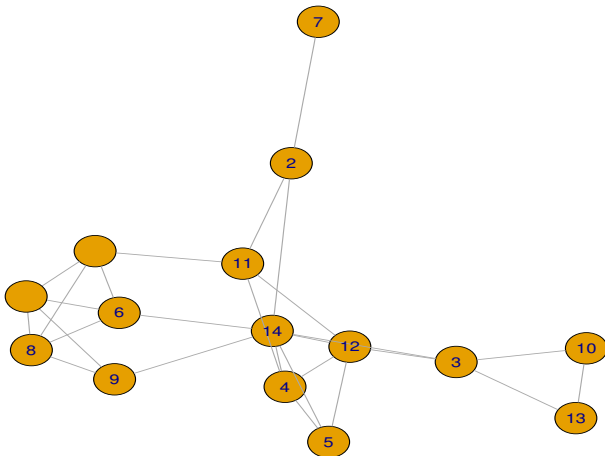
- Proposée par des chercheurs qui se sont retrouvés à Louvain [Blondel et al., 2008]...
- Procédure en deux phases répétées plusieurs fois.



- Proposée par des chercheurs qui se sont retrouvés à Louvain [Blondel et al., 2008]...
- Procédure en **deux phases** répétées plusieurs fois.

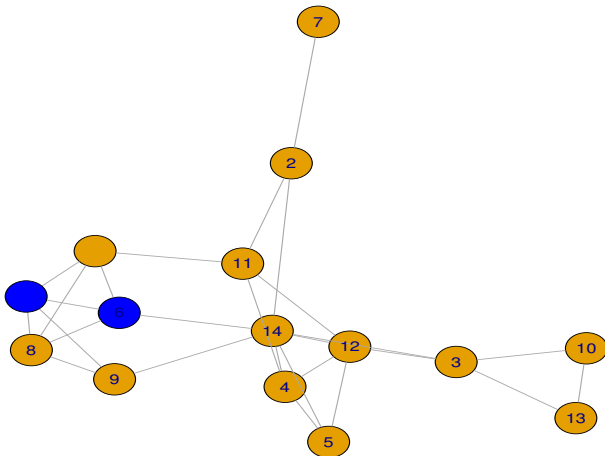
# Phase 1

- Tous les nœuds forment une communauté ;
- Chaque nœud est placé dans la communauté qui maximise le gain de modularité ;



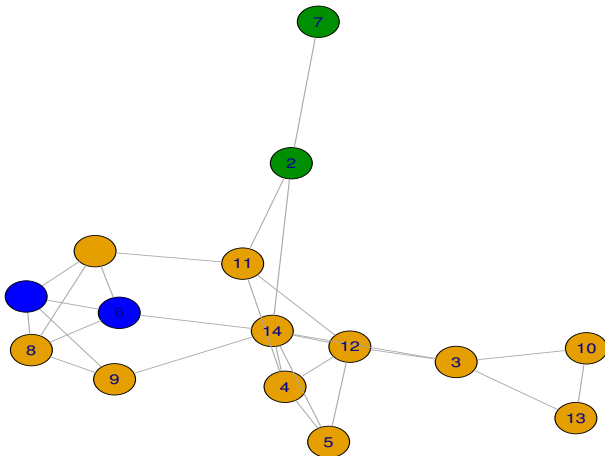
# Phase 1

- Tous les nœuds forment une communauté ;
- Chaque nœud est placé dans la communauté qui maximise le gain de modularité ;



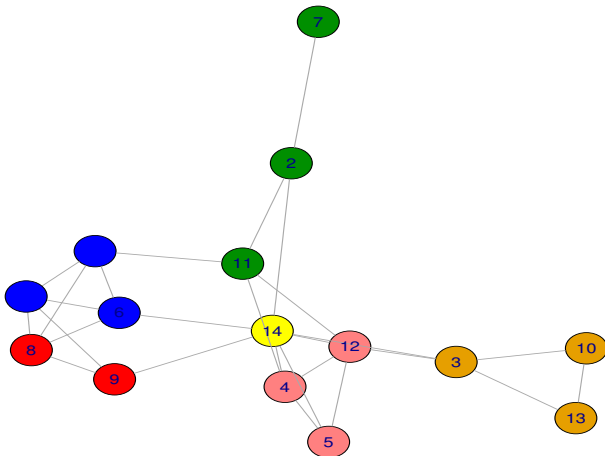
# Phase 1

- Tous les nœuds forment une communauté ;
- Chaque nœud est placé dans la communauté qui maximise le gain de modularité ;

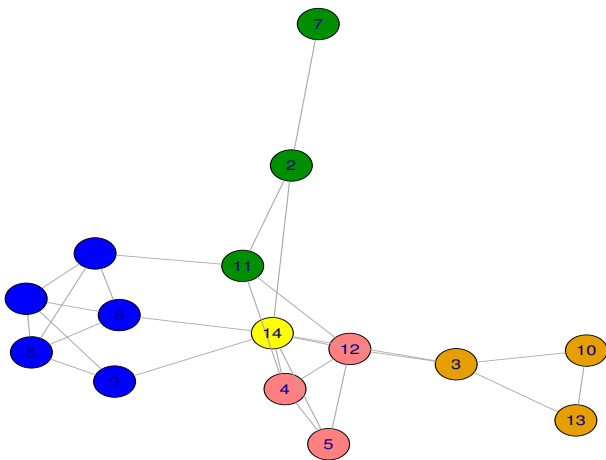


# Phase 1

- Tous les nœuds forment une communauté ;
- Chaque nœud est placé dans la communauté qui maximise le gain de modularité ;

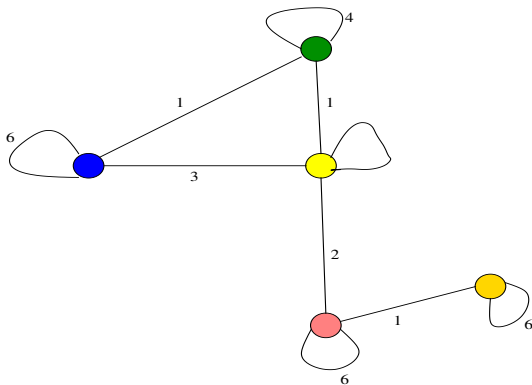


- Le procédé est itéré jusqu'à ce qu'il n'y ait plus d'amélioration (fin de la phase 1)



## Phase 2

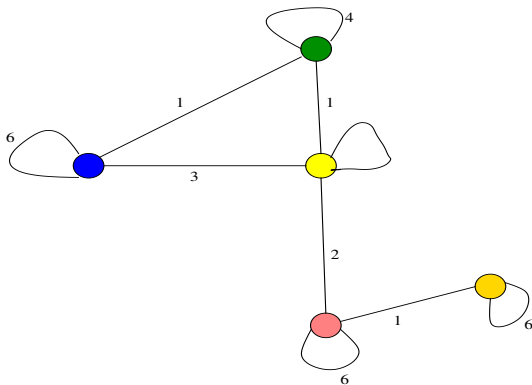
- Construction d'un nouveau graphe dont les **nœuds** sont les **communautés** du graphe de la phase 1 :



- Fin de la première passe - retour à la phase 1 avec ce nouveau graphe.

## Phase 2

- Construction d'un nouveau graphe dont les **nœuds sont les communautés** du graphe de la phase 1 :

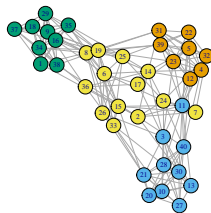
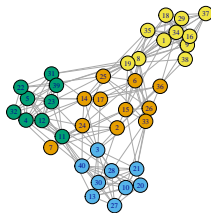
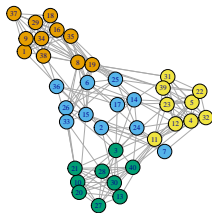


- Fin de la première passe - retour à la phase 1 avec ce nouveau graphe.



# Exemple

```
> cl1 <- cluster_optimal(gg1)
> cl2 <- cluster_louvain(gg1)
> cl3 <- fastgreedy_community(gg1)
```



1 Approche clustering

2 Approche graphe

3 Le projet Edenred

# Présentation de Edenred

Inventeur de Ticket Restaurant et Ticket Kadéos, **Edenred** est le **leader mondial** des services prépayés aux entreprises.

## Histoire

- **1954** : Invention du concept de Titres Restaurant
- **1976** : Début de l'exportation de la formule à l'étranger
- **1985** : Diversification des programmes
- **2010** : Création d'Edenred

## Quelques chiffres

- 2,9 milliards d'Euros de volume d'émission
- 6,7 millions d'utilisateurs (Salariés)
- 120 000 Commerces/Restaurants

# Présentation de Edenred

Inventeur de Ticket Restaurant et Ticket Kadéos, **Edenred** est le **leader mondial** des services prépayés aux entreprises.

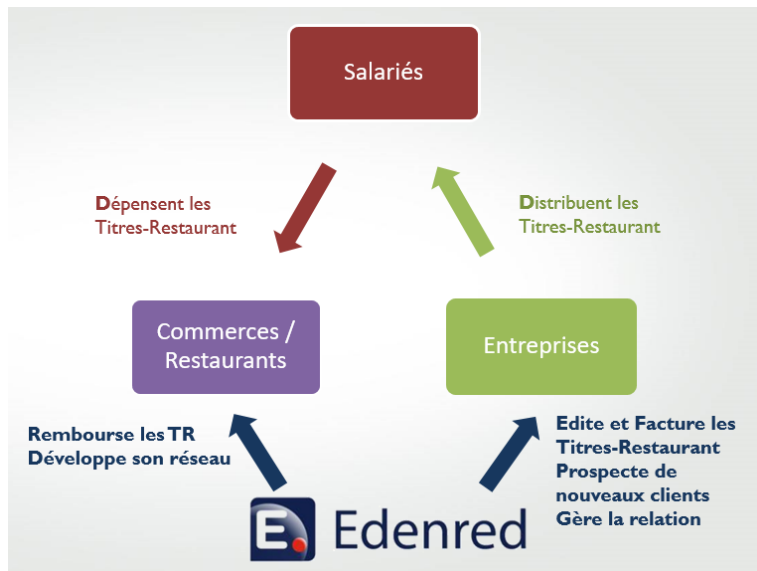
## Histoire

- **1954** : Invention du concept de Titres Restaurant
- **1976** : Début de l'exportation de la formule à l'étranger
- **1985** : Diversification des programmes
- **2010** : Création d'Edenred

## Quelques chiffres

- **2,9 milliards** d'Euros de volume d'émission
- **6,7 millions** d'utilisateurs (Salariés)
- **120 000** Commerces/Restaurants

# Activité historique : un positionnement purement B2B



# Un repositionnement dû à la digitalisation

La dématérialisation, encouragée par l'état, est autorisée par décret depuis le 2 avril 2014.

Les titres-restaurant dématérialisés viennent de dépasser le cap des 2 ans d'existence avec :

- **+ de 20 millions** de transactions réalisées avec la carte
- **160 000** utilisateurs
- **300** transactions à la minute entre 12H et 14H

Actuellement, seulement un peu plus de 10% des bénéficiaires utilisent le support carte, avec une majorité d'entreprises nouvellement couvertes.

# Un repositionnement dû à la digitalisation

La dématérialisation, encouragée par l'état, est autorisée par décret depuis le 2 avril 2014.

Les titres-restaurant dématérialisés viennent de dépasser le cap des 2 ans d'existence avec :

- **+ de 20 millions** de transactions réalisées avec la carte
- **160 000** utilisateurs
- **300** transactions à la minute entre 12H et 14H

Actuellement, seulement un peu plus de 10% des bénéficiaires utilisent le support carte, avec une majorité d'entreprises nouvellement couvertes.

Effectuent des  
transactions

Salariés



Edenred



Débitent un solde  
sur les cartes

Commerces /  
Restaurants

Rembourse  
directement  
les montants  
des transactions

Gère la commande/  
Charge les cartes

Entreprises



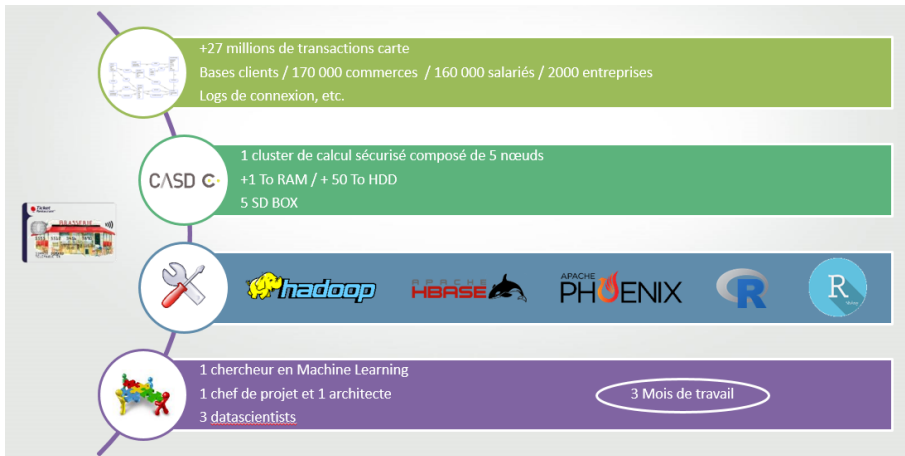
**Les données issues de la digitalisation permettent-elles d'augmenter la proposition de valeur qu'Edenred peut faire à ses différents interlocuteurs ?**

- Développer une offre proactive auprès des salariés bénéficiaires
- Mise en place d'offres promotionnelles ciblées
- Prévoir les comportements pour faciliter le pilotage
- Offrir des leviers de développement aux restaurateurs et commerçants
- Proposer aux entreprises de s'appuyer sur cette relation *alimentaire* pour enrichir la relation salarié-employeur

Les données issues de la digitalisation permettent-elles d'augmenter la proposition de valeur qu'Edenred peut faire à ses différents interlocuteurs ?

- Développer une offre proactive auprès des salariés bénéficiaires
- Mise en place d'offres promotionnelles ciblées
- Prévoir les comportements pour faciliter le pilotage
- Offrir des leviers de développement aux restaurateurs et commerçants
- Proposer aux entreprises de s'appuyer sur cette relation *alimentaire* pour enrichir la relation salarié-employeur

# Le dispositif expérimental mis en place



- Peut-on segmenter les comportements des utilisateurs de la carte ?
- Existe-t-il des comportements communautaires parmi les utilisateurs ?  
Si oui, existe-t-il des leaders au sein de ces communautés ?
- La dématérialisation a-t-elle changé le comportement des gens ? Cela impacte-t-il l'activité d'Edenred ? Le chiffre d'affaires ?
- Les applications en ligne associées à la digitalisation entraînent-elles des comportements spécifiques ?
- ...

## Fonctionnalités

- analyses comparées des populations
- suivi de l'activité
- visualisation des segmentations réalisées
- identifications des leaders et des communautés

## Architecture

- données indexées via [HBASE](#) et [PHOENIX](#)
- [R](#) comme moteur de calcul
- restitution avec [R-Shiny](#) et l'utilisation de composants [JavaScripts](#) dynamiques

## Fonctionnalités

- analyses comparées des populations
- suivi de l'activité
- visualisation des segmentations réalisées
- identifications des leaders et des communautés

## Architecture

- données indexées via [HBASE](#) et [PHOENIX](#)
- **R** comme moteur de calcul
- restitution avec [R-Shiny](#) et l'utilisation de composants [JavaScripts](#) dynamiques

## Ciblage des populations

624

No affilia

62667

No transactions

613412

VE

9930

CA (L-PMV)

3741

No affilia

66490

No transactions

905309

VE

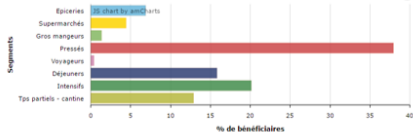
29808

CA (L-PMV)

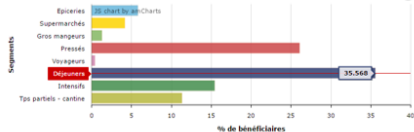
Vue globale

Vue mensuelle

## Typologie des bénéficiaires

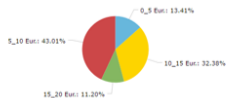


## Typologie des bénéficiaires



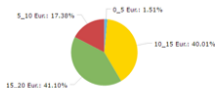
3D chart by amCharts

## Montants des transactions



3D chart by amCharts

## Montants des transactions

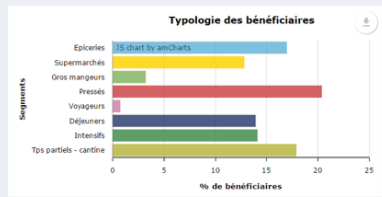
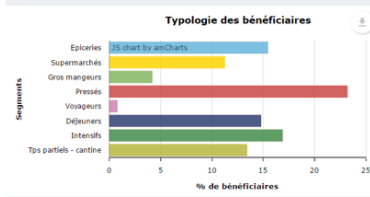
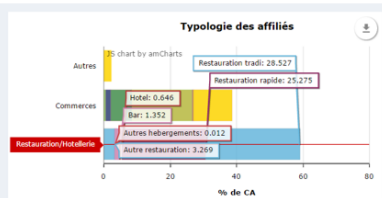
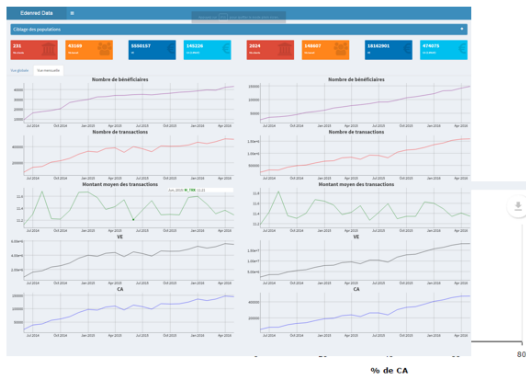


## Impacts météorologiques



## Impacts météorologiques







On s'intéresse à l'identification de "leader" au sein d'une ou plusieurs entreprises.

Qu'est-ce qu'un leader ?

- Un leader est la personne qui va motiver ses collègues à aller manger ensemble
- Et donc un individu qui a tendance à utiliser souvent sa carte au même endroit et au même moment que d'autres individus de la même entreprise

## 1 - Identification des leaders

- Construction d'une matrice d'affinité :
  - en ligne les individus
  - en colonne des créneaux de transaction **X** les lieux de consommation.

La matrice est codée 1 si une transaction pour l'*individu i* sur le créneau et dans le lieu *j* a été observée, 0 sinon

- Biclustering de cette matrice : détecter les individus ayant utilisé leur carte au même endroit sur un même créneau horaire
  - paramètres : nombre minimum d'individus et de repas, ainsi que le nombre de groupes

Les leaders sont ceux apparaissant le plus de fois dans un groupe.

## 1 - Identification des leaders

- Construction d'une matrice d'affinité :
  - en ligne les individus
  - en colonne des créneaux de transaction **X** les lieux de consommation.

La matrice est codée 1 si une transaction pour l'*individu i* sur le créneau et dans le lieu *j* a été observée, 0 sinon

- Biclustering de cette matrice : détecter les individus ayant utilisé leur carte au même endroit sur un même créneau horaire
  - **paramètres** : nombre minimum d'individus et de repas, ainsi que le nombre de groupes

Les leaders sont ceux apparaissant le plus de fois dans un groupe.

## 2 - Identification des communautés

Cette étape se fait pour un leader donné, avec l'identification des communautés rattachées.

- ① **Base de départ** : le leader et tous les bénéficiaires ayant co-consommé avec lui au moins 1 fois.
- ② **Analyse par graphe**
  - **noeud** : représente un bénéficiaire
  - **lien** : nombre de repas pris ensemble
- ③ **Clustering du réseau** : maximisation d'un critère statistique afin de constituer des groupes où les gens ont beaucoup mangé ensemble et peu avec les autres groupes.

## 2 - Identification des communautés

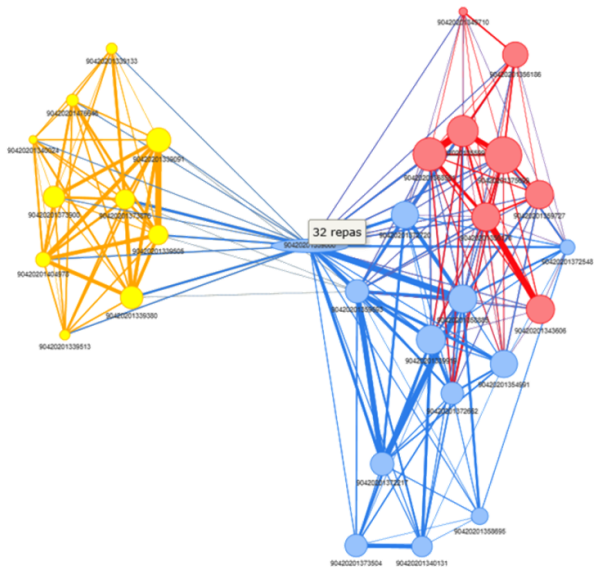
Cette étape se fait pour un leader donné, avec l'identification des communautés rattachées.

- ① **Base de départ** : le leader et tous les bénéficiaires ayant co-consommé avec lui au moins 1 fois.
- ② **Analyse par graphe**
  - **noeud** : représente un bénéficiaire
  - **lien** : nombre de repas pris ensemble
- ③ **Clustering du réseau** : maximisation d'un critère statistique afin de constituer des groupes où les gens ont beaucoup mangé ensemble et peu avec les autres groupes.

## 2 - Identification des communautés

Cette étape se fait pour un leader donné, avec l'identification des communautés rattachées.

- ① **Base de départ** : le leader et tous les bénéficiaires ayant co-consommé avec lui au moins 1 fois.
- ② **Analyse par graphe**
  - **noeud** : représente un bénéficiaire
  - **lien** : nombre de repas pris ensemble
- ③ **Clustering du réseau** : maximisation d'un critère statistique afin de constituer des groupes où les gens ont beaucoup mangé ensemble et peu avec les autres groupes.



- **Biclustering** : le package **biclust**, avec différentes méthodes
  - BCBimax : Groups with ones in binary matrix
  - BCCC : Constant values
  - BCPlaid : Constant values over rows or columns
  - BCSpectral : Coherent values over rows and columns
  - BCXmotifs : Coherent correlation over rows and columns
- **Traitements de réseaux** : le package **igraph**, avec notamment différents algorithmes de détection de communautés :
  - `cluster_fast_greedy`, `cluster_walktrap`, `cluster_spinglass`,  
...
- **Visualisation interactive** de réseaux : le package **visNetwork**



# Example

```
# Generate random data
set.seed(1234) ; n <- 1000 ; p <- 100

data <- matrix(abs(rnorm(n*p)), n, p)
rownames(data) <- paste0("Person ", 1:nrow(data))
colnames(data) <- paste0("Eat-time ", 1:ncol(data))

# binarize
data_bin <- biclust::binarize(data, 2.1)
head(data_bin[1:4, 1:4])
```

##	Eat-time 1	Eat-time 2	Eat-time 3	Eat-time 4
## Person 1	0	0	0	0
## Person 2	0	0	0	0
## Person 3	0	0	0	0
## Person 4	1	0	0	0

```
# find bicluster
```

```
res <- biclust::biclust(x = data_bin,  
                        method=BCBimax(), number = 500,  
                        minr = 2, minc = 2)
```

```
# get bicluster
```

```
info_biclust <- biclust::bicluster(data_bin, res)  
info_biclust[1]
```

```
## $Bicluster1
```

```
##           Eat-time 75 Eat-time 100  
## Person 506           1           1  
## Person 870           1           1  
## Person 1000          1           1
```

```
# get leaders
```

```
leader <- rowSums(res@RowxNumber)
```

```
names(leader) <- rownames(data_bin)
```

```
leader <- sort(leader[leader != 0], decreasing = T)
```

```
head(leader, n = 4)
```

```
## Person 718 Person 271 Person 899 Person 802
```

```
##          40          38          38          26
```

```
# get subdata for network
```

```
leader_1 <- names(leader)[1]
```

```
eat_l1 <- which(data_bin[leader_1, ] > 0)
```

```
friends_l1 <- which(rowSums(data_bin[, eat_l1]) > 0)
```

```
data_net <- data_bin[friends_l1, eat_l1]
```

```
# remove only one 1 (random data ....!)
```

```
data_net <- data_net[which(rowSums(data_net) > 1), ]
```

```
# edges
```

```
mat_edges <- data_net %*% t(data_net)  
head(mat_edges[1:5, 1:4])
```

##	Person 22	Person 38	Person 44	Person 45
## Person 22	2	0	0	1
## Person 38	0	2	0	0
## Person 44	0	0	2	0
## Person 45	1	0	0	2
## Person 65	0	0	1	1

```
# use igraph to do clustering
```

```
ig <- igraph::graph_from_adjacency_matrix(  
  mat_edges, weighted = TRUE,  
  mode = "upper", diag = FALSE)  
fg <- igraph::cluster_fast_greedy(ig)
```

```

# prepare for visNetwork
data_vis <- toVisNetworkData(ig)
nodes <- data_vis$nodes ; edges <- data_vis$edges
edges$value <- edges$weight*5 # size
nodes$group <- membership(fg)

nodes$shape <- "dot"
nodes[nodes$id %in% leader_1, "shape"] <- "ellipse"

head(nodes, n = 1) ; head(edges, n = 1)

```

```

##              id      label group shape
## Person 22 Person 22 Person 22      2   dot

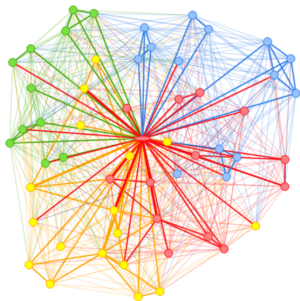
##      from      to weight value
## 1 Person 22 Person 45      1     5

```

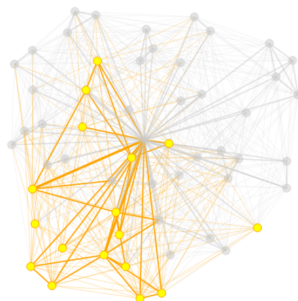
*# And visualize !*





```
visNetwork(nodes, edges) %>% visIgraphLayout() %>%  
  visOptions(highlightNearest = T, selectedBy = "group")
```

Select by group ▾



3 ▾



-  Blondel, V. D., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008).  
Fast unfolding of communities in large networks.  
*Journal of Statistical Mechanics : Theory and Experiment*.
-  Fortunato, S. (2010).  
Community detection in graphs.  
*Physics report*, 486 :75–174.
-  Kaiser, S. (2011).  
Biclustering : Methods, software and application.  
[https://edoc.ub.uni-muenchen.de/13073/1/Kaiser\\_Sebastian.pdf](https://edoc.ub.uni-muenchen.de/13073/1/Kaiser_Sebastian.pdf).
-  Prelic, A. and Bleuler, S., Zimmermann, P., Wil, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. (2006).  
A systematic comparison and evaluation of biclustering methods for gene expression data.  
*Bioinformatics*, 22 :1122–1129.