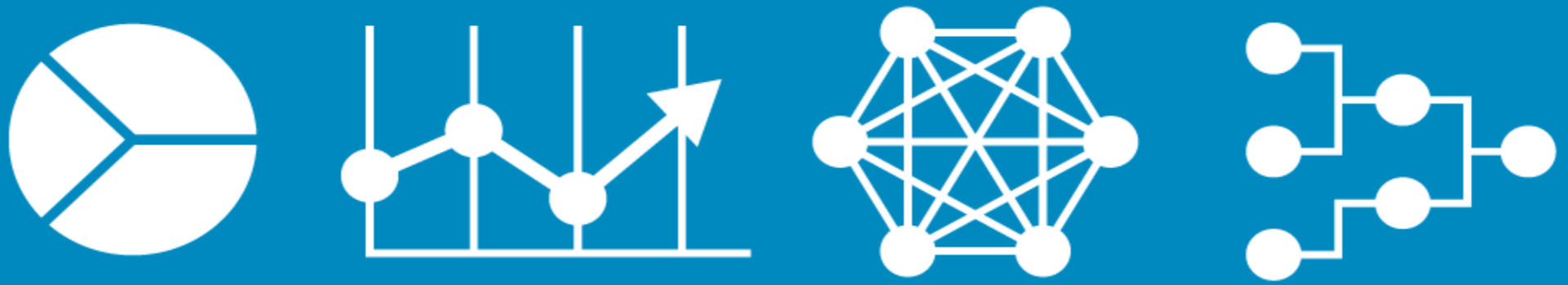
ctrl F Agency



# Astuces en text-mining : removeSparseTerms

## Text-mining Basics : Tips and Tricks R(S)

Par Thibaut LOMBARD

Décembre 2016

ctrl F Agency

- Founder of ctrl+f agency (and all stuff related)
- Web/mobile Dev,
- Data science,
- Research
- Teaching

Age : 30 Years Old

ctrl F Agency

# Parcimonie Document/Fréquence

**Définition :**

La parcimonie est la précision relative au rapport document/fréquence d'un ou plusieurs termes contenu dans une matrice (triplet).

Cette précision dépends :

- Du nombre de termes contenu dans le document
- Du nombre de document(s) dans laquelle la fonction est exécutée.

**Formule**

$$df_j > N * (1 - \theta)$$

Pour

- $1 < \theta < 0$
- La lettre j (le terme)
- N le nombre de documents

ctrl F Agency

# Sparse.r

```
packages <- function(paquets)
{
        new.paquets <- paquets[!(paquets %in% installed.packages()[, "Package"])] if
        (length(new.paquets))
        install.packages(new.paquets, dependencies = TRUE, repos='http://cran.rstudio.com/')
        sapply(paquets, require, character.only = TRUE)
}
packages(c("NLP", "tm", "NMF", "proxy"))
```

ctrl F Agency

# Rscript sparse.r

```
monTexte <- c("Bâchez la queue du wagon-taxi avec les pyjamas du fakir.",
              "la matrice du wagon-taxi",
              "le fakir est dans la matrice")
monCorpus <- Corpus(VectorSource(monTexte))
maTdm <- DocumentTermMatrix(monCorpus, control = list(minWordLength = 1))
```

ctrl F Agency

# inspect(maTdm)

```
[1] "**********************************************************"
[1] "Inspection de la matrice Term document maTdm avec as.matrix()"
[1] "**********************************************************"
<<DocumentTermMatrix (documents: 3, terms: 11)>>
Non-/sparse entries: 13/20
Sparsity           : 61%
Maximal term length: 10
Weighting          : term frequency (tf)
    Terms
Docs avec bâchez dans est fakir fakir. les matrice pyjamas queue wagon-taxi
   1    1      1    0   0     0      1   1        0       1     1          1
   2    0      0    0   0     0      0   0        1       0     0          1
   3    0      0    1   1     1      0   0        1       0     0          0
```

ctrl F Agency

# Le calcul de sparsity (parcimonie) 50%

```
c <- removeSparseTerms(maTdm, 0.5)
```

```
<<DocumentTermMatrix (documents: 3, terms: 2)>>
Non-/sparse entries: 4/2
Sparsity           : 33%
Maximal term length: 10
Weighting          : term frequency (tf)

    Terms
Docs matrice wagon-taxi
   1       0          1
   2       1          1
   3       1          0
```

ctrl F Agency

# Le calcul de sparsity (parcimonie) 90%

```
d <- removeSparseTerms(maTdm, 0.9)

<<DocumentTermMatrix (documents: 3, terms: 11)>>
Non-/sparse entries: 13/20
Sparsity           : 61%
Maximal term length: 10
Weighting          : term frequency (tf)

    Terms
Docs avec bâchez dans est fakir fakir. les matrice pyjamas queue wagon-taxi
   1    1      1    0   0     0      1   1       0       1     1          1
   2    0      0    0   0     0      0   0       1       0     0          1
   3    0      0    1   1     1      0   0       1       0     0          0
```

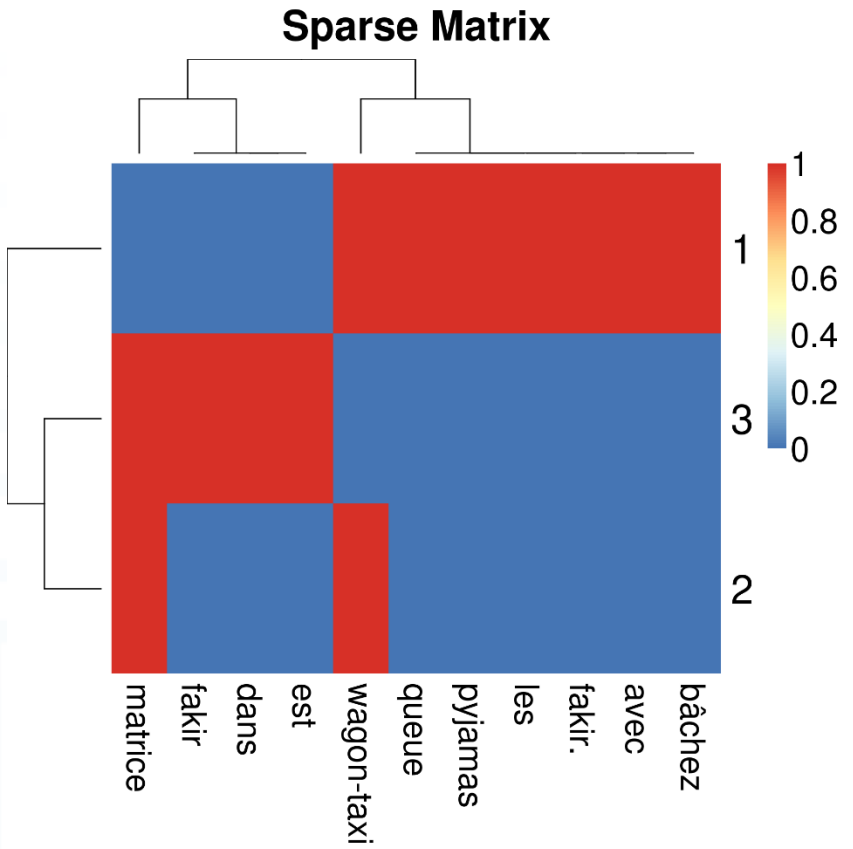
ctrl F Agency

# Création des annotated Heatmap

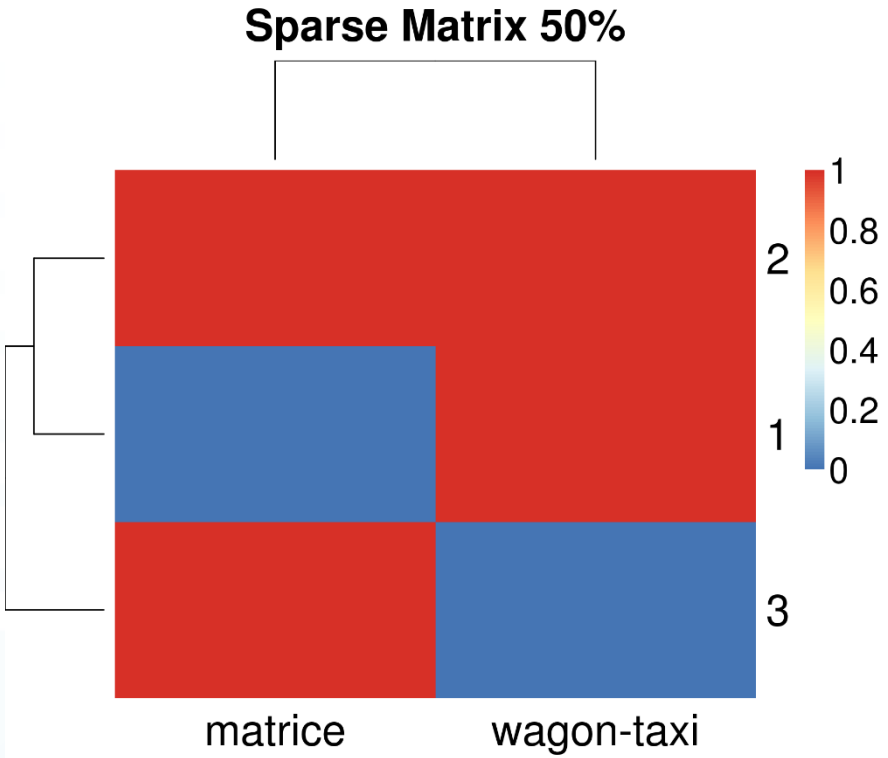
```
b <- as.matrix(maTdm)
ch <- as.matrix(removeSparseTerms(maTdm, 0.5))
dh <- as.matrix(removeSparseTerms(maTdm, 0.9))

aheatmap(b, filename = "sparse-heatmap-matrix.png")
aheatmap(ch, filename = "sparse-heatmap-50percent.png")
aheatmap(dh, filename = "sparse-heatmap-99percent.png")
```

ctrl F Agency

# Annotated Heatmap (matrice maTdm)

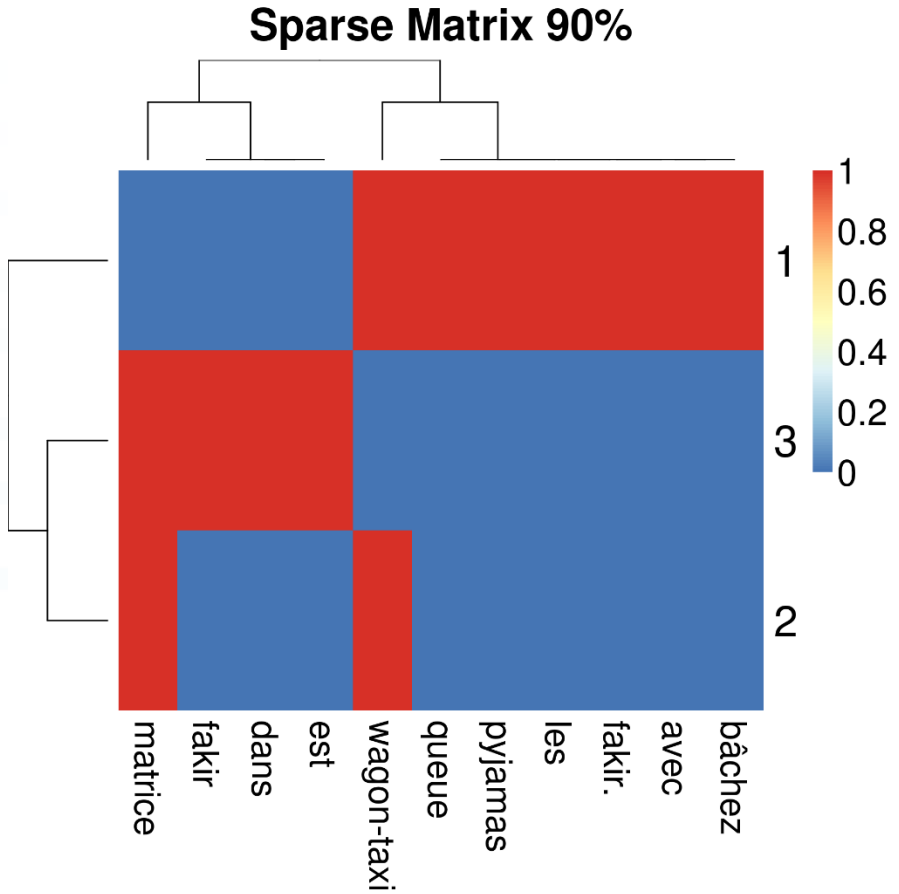


# Annotated Heatmap Parcimonie à 50%



ctrl F Agency

## Annotated Heatmap Parcimonie à 90%



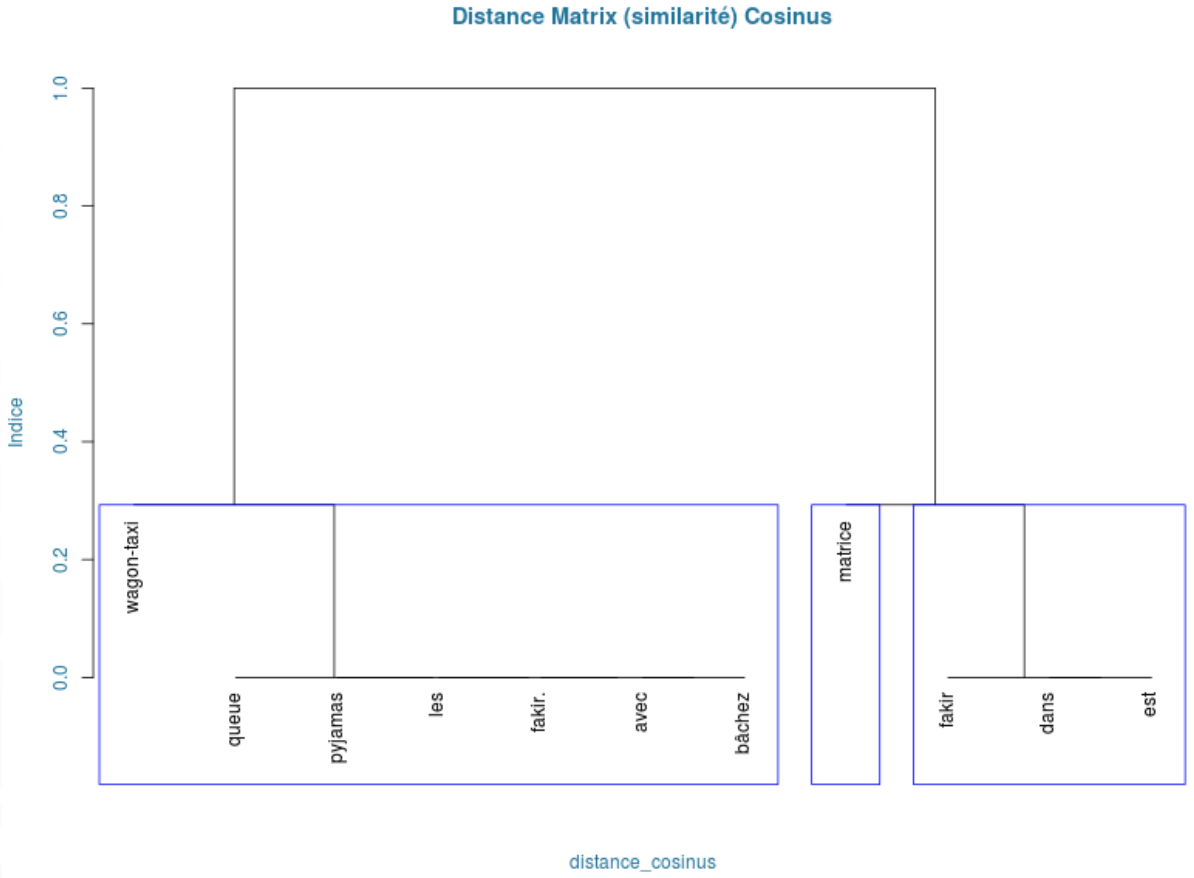
## Matrice de similarité (méthode cosinus) Eisen et al. 1998

$$d_{eisen}(\mathbf{x},\mathbf{y}) = 1 - \frac{\mathbf{x}'\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{\left|\sum_{i=1}^{m} x_i y_i\right|}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}}$$

```
distance_cosinus <- dist(as.matrix(t(b)), method = "cosine")
distance_cosinus
```

```
           avec    bâchez      dans       est     fakir    fakir.
bâchez    0.0000000
dans      1.0000000 1.0000000
est       1.0000000 1.0000000 0.0000000
fakir     1.0000000 1.0000000 0.0000000 0.0000000
fakir.    0.0000000 0.0000000 1.0000000 1.0000000 1.0000000
les       0.0000000 0.0000000 1.0000000 1.0000000 1.0000000 0.0000000
matrice   1.0000000 1.0000000 0.2928932 0.2928932 0.2928932 1.0000000
pyjamas   0.0000000 0.0000000 1.0000000 1.0000000 1.0000000 0.0000000
queue     0.0000000 0.0000000 1.0000000 1.0000000 1.0000000 0.0000000
wagon-taxi 0.2928932 0.2928932 1.0000000 1.0000000 1.0000000 0.2928932
```

# Clustering similarité cosinus (Eisen)



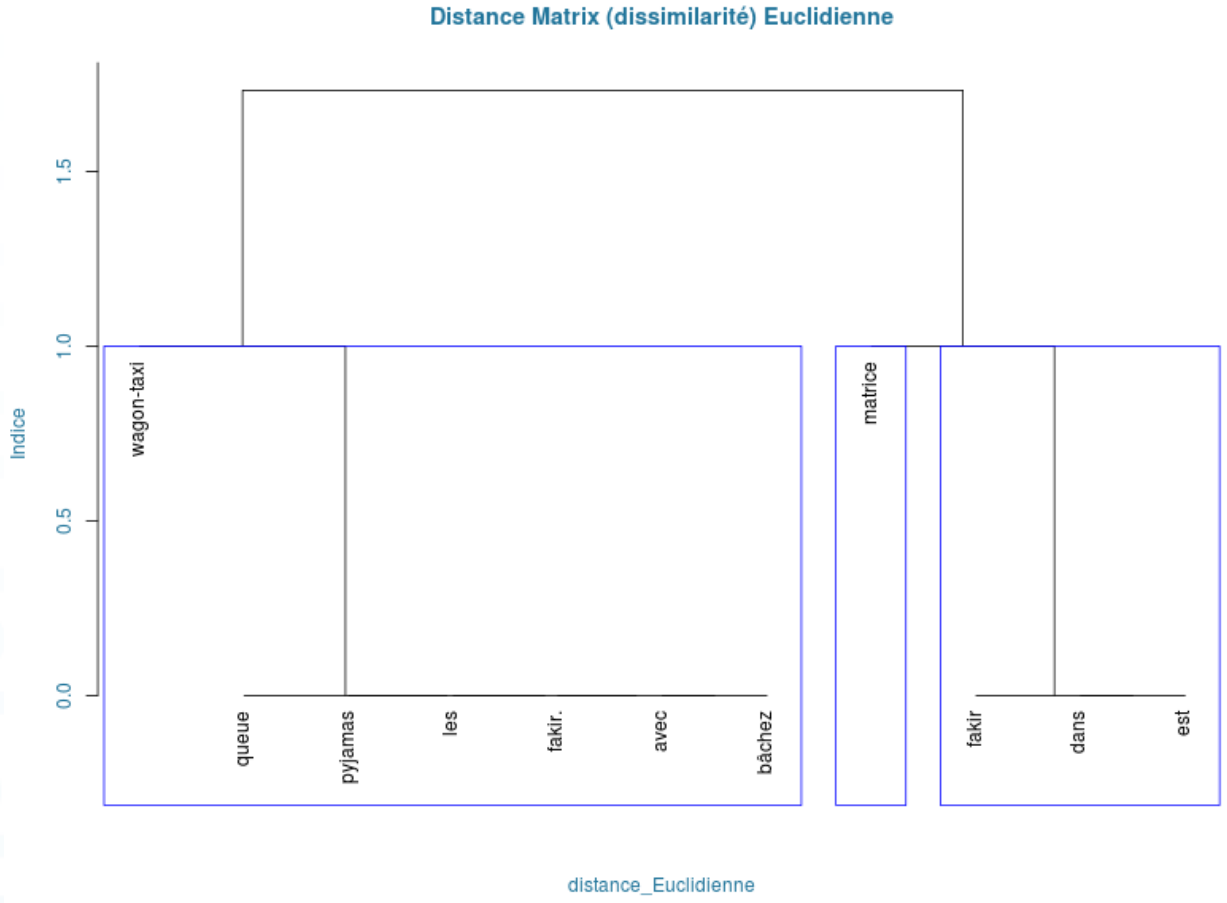
## Matrice de dissimilarité , calcul de distance Euclidienne

$$d_{euc}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

```
distance_Euclidienne <- dist(as.matrix(t(b)), method = "Euclidean")
distance_Euclidienne
```

```
           avec   bâchez     dans      est    fakir   fakir.      les
bâchez     0.000000
dans       1.414214 1.414214
est        1.414214 1.414214 0.000000
fakir      1.414214 1.414214 0.000000 0.000000
fakir.     0.000000 0.000000 1.414214 1.414214 1.414214
les        0.000000 0.000000 1.414214 1.414214 1.414214 0.000000
matrice    1.732051 1.732051 1.000000 1.000000 1.000000 1.732051 1.732051
pyjamas    0.000000 0.000000 1.414214 1.414214 1.414214 0.000000 0.000000
queue      0.000000 0.000000 1.414214 1.414214 1.414214 0.000000 0.000000
wagon-taxi 1.000000 1.000000 1.732051 1.732051 1.732051 1.000000 1.000000
```

# Clustering dissimilarité Euclidienne



ctrl F Agency

# Création d'un graphique de clustering

```
fit <- hclust(distMatrix)
png(filename="sparse-clust.png",width=800,height=600)
plot(fit)
rect.hclust(fit, k = 3,border="blue")
```

ctrl F Agency

# Q/A