****



**Analyse de données textuelles**

Restaurant : Sushishop Vieux-Lille

Reda Belhoti

# **Introduction**

## Présentation du restaurant « Sushishop – Vieux Lille »

Sushi Shop est une chaîne [française](https://fr.wikipedia.org/wiki/France) de [restauration](https://fr.wikipedia.org/wiki/Restaurant) rapide [japonaise](https://fr.wikipedia.org/wiki/Japon), fondée en 1998 par Grégory Marciano et Hervé Louis.

La chaîne de restaurants Sushi Shop est le leader européen sur son créneau avec plus de 160 magasins dans douze pays et un chiffre d'affaires de 202 millions d'euros en 2017, dont plus de la moitié provient de la commande en ligne livrée aux clients.

Sushi Shop propose un service haut de gamme de restauration sur place, à emporter et en livraison de cuisine japonaise auprès des particuliers et des entreprises. le réseau offre un service d'excellence identique dans l'ensemble de ses points de vente. La restauration est un marché hyperconcurrentiel, les clients sont aussi précieux que des diamants : il faut garantir un service irréprochable, à la hauteur des attentes d'une clientèle exigeante, qui entend être reconnue dans l'ensemble de nos points de ventes.

## Présentation du projet

Les avis consommateurs représente une preuve “sociale” de la fiabilité et de la crédibilité d’une marque. Lorsqu’ils sont maitrisés et suivis, les avis clients deviennent un véritable atout pour les sites de e-commerce et permettent de renforcer leur visibilité et leur réputation.

L’étude d’analyse des données textuelles a pour objet le traitement statistique des données textuelles.

Pour cela l’analyse des données textuelles sur la base sushishop-vieux lille comporte 133 avis collectés sur la plateforme TripAdvisor.

Outil de traitement : Logiciel R , Excel .

# **Analyse**

## Présentation « technique »

### Mode d’extraction

Regroupement des avis du le site TripAdvisor dans Tableur Excel selon leurs niveaux de qualité (Excellent - Très Bon – Moyenne - Médiocre - Horrible) pour construire une base de données.

### Volume de la base

133 Avis sur Le restaurant sushishop VieuxLille.

### Statistiques descriptive sur les Avis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Minimum** | **1erQuartile** | **Médiane** | **3 Quartile** | **Moyenne** | **Maximum** |
| 100 | 159 | 240 | 335 | 414 | 2121 |

* En moyenne un avis dans cette base contient 414 mots

### Nettoyage de la base

* Les textes d’avis collectés sur Internet sont toujours plein de fautes (orthographe) et, quel que soit le texte, tous les mots ne nous intéressent pas (il s’agit des mots-outils).
* On enlève du corpus tous les mots-outils du dictionnaire modifié. On importe le dictionnaire sous format csv, on obtient une base dico.
* Il est pratique d’enlever toutes les lettres accentuées (les cédilles, etc.). En particulier dans des avis sur Internet, beaucoup de mots sont écrits avec ou sans accents. Du coup, en enlevant tous les accents, on uniformise le corpus… même si à terme cela peut compliquer la lecture.
* On enlève les ponctuations.
* Aux mots du lexique - en majuscules - on associe leur racine.
* On peut ajouter les fréquences des mots racines. Cela permet par exemple de faire remonter tous les termes qui évoquent ce qui est “petit” (qui englobent toutes les formes fléchies : petit, petits, petite, petites)

## Construction d’un lexique général

### Dimension du corpus

|  |  |
| --- | --- |
| Nombre des mots | Nombre d’avis |
| 1519 | 133 |

### Tableaux des fréquences des mots

**Commentaires :**

Dans la base, le mot « qualité » est répété 39 fois. Ceci signifie que la qualité est un facteur important pour la clientèle de Sushishop.

Sushishop est le leader des restaurants de sushi en termes de livraison. Le service « commander en livraison » est beaucoup mentionné dans les avis.

|  |  |  |
| --- | --- | --- |
| **Mot** | **Racine** | **Fréquence** |
| SUSHI | sush | 133 |
| COMMANDE | command | 81 |
| PLUS | plus | 61 |
| TRES | tre | 60 |
| QUALITE | qualit | 39 |
| SHOP | shop | 37 |
| LIVRAISON | livraison | 37 |
| BIEN | bien | 33 |
| TOUT | tout | 32 |
| SERVICE | servic | 31 |
| PRIX | prix | 29 |
| FOIS | fois | 28 |
| FRAIS | frais | 28 |
| FAIT | fait | 26 |
| PEU | peu | 23 |
| RESTAURANT | restaur | 23 |
| CAR | car | 22 |
| BON | bon | 20 |
| RIEN | rien | 20 |
| PRODUITS | produit | 20 |
| EMPORTER | emport | 20 |
| DEUX | deux | 20 |

### Representation graphique du Lexique

### 

### Tableaux des fréquences des Racines

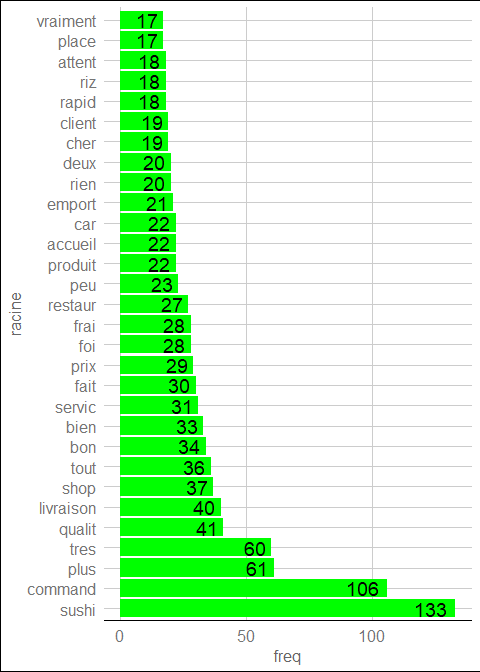
**Commentaires**

Dans les Avis les clientes mentionnent les mots « Bon » « bien » « produit » « prix » « accueil »

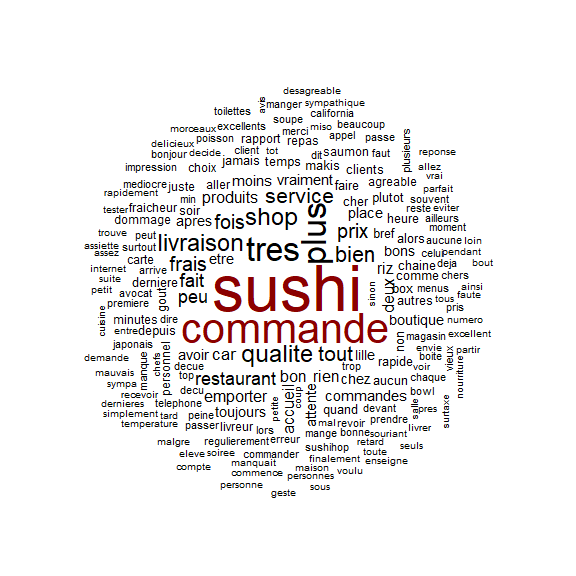
Cela signifie que les attentes des clients sont plutôt exigeantes. Il paraît que les clients attendent un bon rapport qualité/prix mais sans que la qualité du service soit affectée.

|  |  |  |
| --- | --- | --- |
| **Mot** | **Racine** | **Fréquence** |
| sushi | sushi | 133 |
| command | command | 106 |
| plus | plus | 61 |
| tres | tres | 60 |
| qualit | qualit | 41 |
| livraison | livraison | 40 |
| shop | shop | 37 |
| tout | tout | 36 |
| bon | bon | 34 |
| bien | bien | 33 |
| servic | servic | 31 |
| fait | fait | 30 |
| prix | prix | 29 |
| foi | foi | 28 |
| frai | frai | 28 |
| restaur | restaur | 27 |
| peu | peu | 23 |
| produit | produit | 22 |
| accueil | accueil | 22 |
| car | car | 22 |
| emport | emport | 21 |
| rien | rien | 20 |
| deux | deux | 20 |

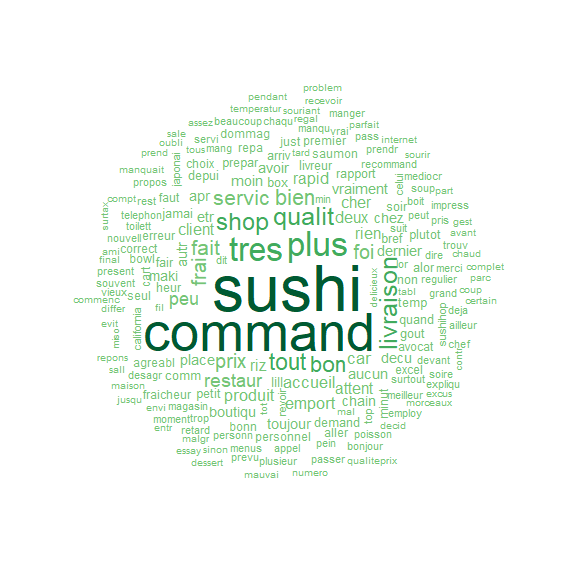
### Représentation graphique du Lexique racine



### Représentation graphique du Lexique (nuage des mots)



### Représentation graphique du Lexique (nuage des mots)



## Étude de mots importants (en lien avec votre problématique): mesures d’association

### Corrélation entre les Mots choisis

« Sushi , commande ,livraison , qualite , accueil , plus »

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **sushi** | **commande** | **livraison** | **qualite** | **accueil** | **plus** |
| **sushi** | 1.00 | 0.25 | 0.08 | 0.05 | 0.09 | 0.29 |
| **commande** | 0.25 | 1.00 | 0.31 | -0.02 | -0.03 | 0.24 |
| **livraison** | 0.08 | 0.31 | 1.00 | -0.16 | -0.12 | 0.10 |
| **qualite** | 0.05 | -0.02 | -0.16 | 1.00 | 0.21 | 0.07 |
| **accueil** | 0.09 | -0.03 | -0.12 | 0.21 | 1.00 | 0.09 |
| **plus** | 0.29 | 0.24 | 0.10 | 0.07 | 0.09 | 1.00 |

**Commentaires :**

Les mots « livraison » et « commande » apparait ensemble avec un coefficient de 0,31.

Le mot « accueil » parait avec le mot « service » dans un avis avec un coefficient de 0,21

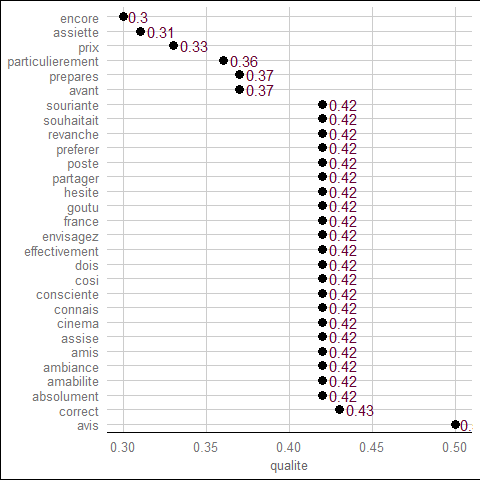
Le mot « accueil » parait avec le mot « commande » dans un avis avec un coefficient

De -0.03 ceci signifie qu’ils n’ont jamais ensemble dans le même avis.

Beaucoup de commande passent par téléphone ou sur internet et pas en sur place.

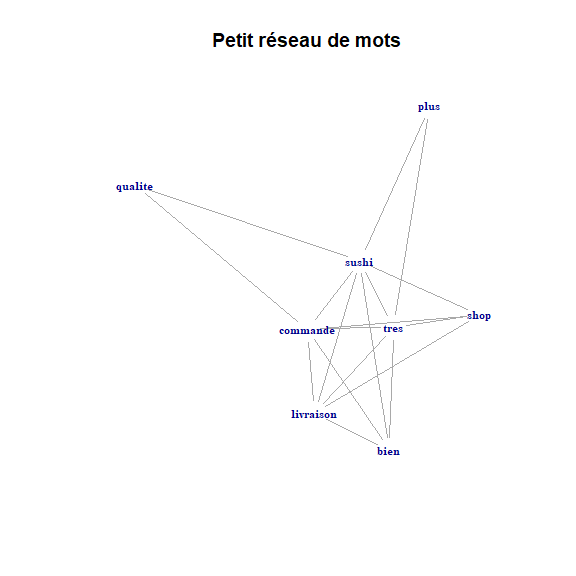
**Représentation des coefficients de corrélation**

### Association des mots avec le mot « qualité



**Commentaire :**

Le mots qualité est beaucoup associé avec le mot correct. Cela pourrait signifier que beaucoup de clients trouvent que la qualité est correcte.



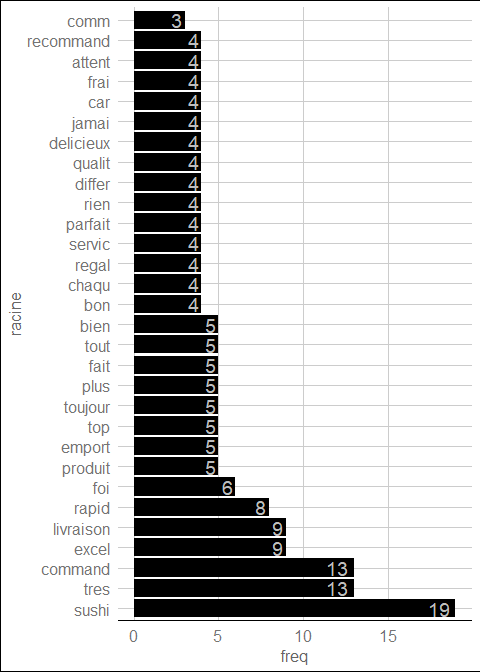
## Comparaison de sous-corpus

Extraire deux sous corpus .

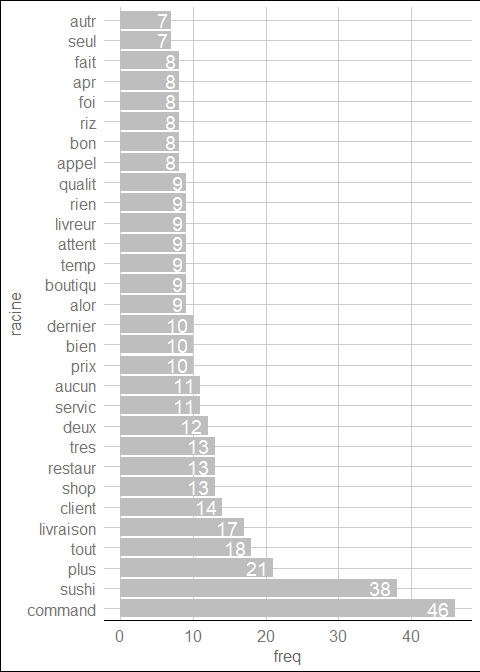
|  |  |  |
| --- | --- | --- |
| **Sous-corpus** | **Nombre des mots** | **Nombre d’Avis** |
| **Avis\_Excellent** | 315 | 26 |
| **Avis\_Horrible** | 776 | 38 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Avis Excellent | | | Avis Horrible | | |
|  | racine | Fréquence |  | racine | Fréquence |
| sushi | sushi | 19 | command | command | 46 |
| tres | tres | 13 | sushi | sushi | 38 |
| command | command | 13 | plus | plus | 21 |
| excel | excel | 9 | tout | tout | 18 |
| livraison | livraison | 9 | livraison | livraison | 17 |
| rapid | rapid | 8 | client | client | 14 |
| foi | foi | 6 | shop | shop | 13 |
| **Commentaire :**  Les mots qui ont une occurrence de plus de 6 fois, dans les deux corpus, signifient que pour les avis excellents, les clients donnent beaucoup d’importance à la rapidité livraison et l’excellence.  Pour le corpus des avis horribles, les clients utilisent des termes qui reflètent les problèmes de livraison comme le temp d’attente, la qualité du riz et l’expérience lors de la passation des commandes au téléphone. | | | restaur | restaur | 13 |
| tres | tres | 13 |
| deux | deux | 12 |
| servic | servic | 11 |
| prix | prix | 10 |
| bien | bien | 10 |
| dernier | dernier | 10 |
| boutiqu | boutiqu | 9 |
| temp | temp | 9 |
| attent | attent | 9 |
| livreur | livreur | 9 |
| qualit | qualit | 9 |
| appel | appel | 8 |
| bon | bon | 8 |
| riz | riz | 8 |
| autr | autr | 7 |
| minut | minut | 7 |
| employ | employ | 7 |
| arriv | arriv | 6 |
| produit | produit | 6 |
| desagr | desagr | 6 |
| faut | faut | 6 |
| moin | moin | 6 |
| personn | personn | 6 |
| avoir | avoir | 6 |
| telephon | telephon | 6 |
| accueil | accueil | 6 |

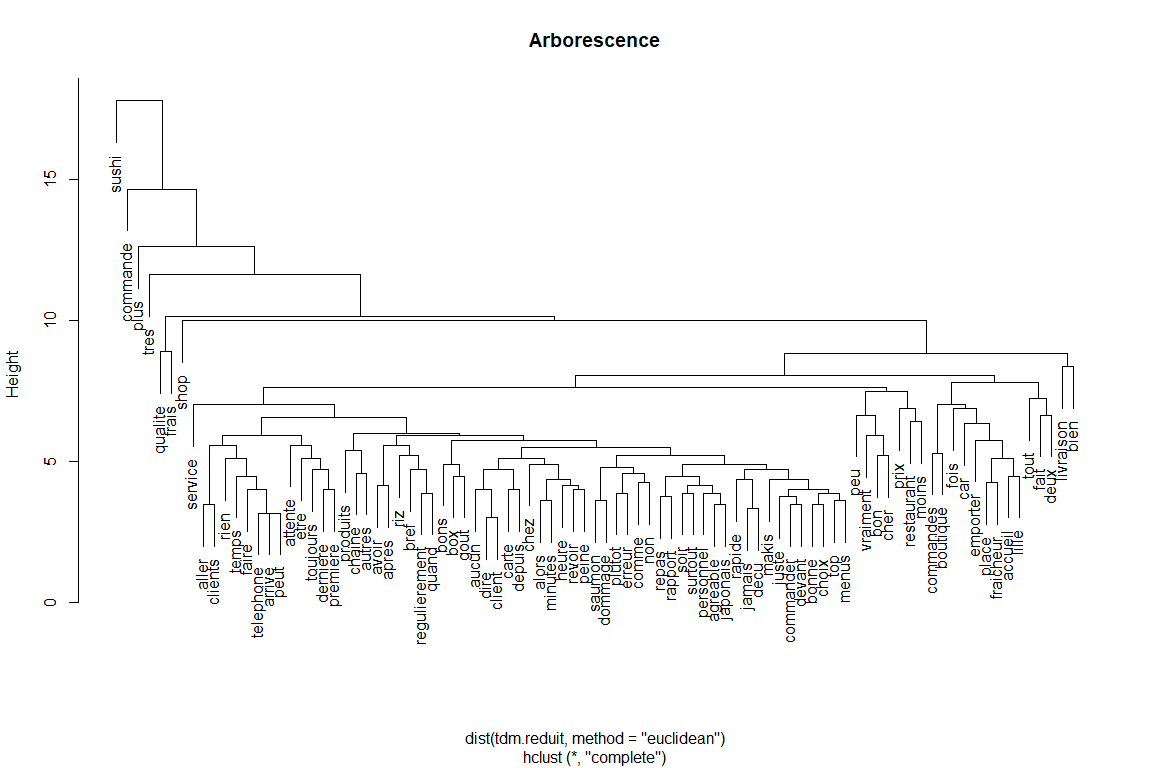
### Avis Excellent



### Avis Excellent



## Conclusion générale



Il est très important d’avoir des outils de diagnostic afin de s’améliorer et prendre de meilleures décisions. L’analyse des avis est toute aussi importante, parce qu’elle permet d’avoir un retour direct des clients et du coup aide les entreprises à améliorer les failles de leurs activités.  
   
Dans le cas des restaurants, la livraison est ce qui rentre le plus d’argent. C’est une part non négligeable du chiffre d’affaires mais qui représente toujours un défi pour les restaurateurs : Il faut bien gérer tout le processus dès la prise de commande jusqu’à l’arrivé de la commande chez le client. Le service doit être excellent, surtout en période de Rush.

# **Annexes**

Table of Contents

[1. Importer la base *1*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458735)

[2. Nettoyer les textes du corpus *3*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458736)

[2.3. Modifier le corpus de textes *3*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458737)

[3. Les lexiques *6*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458738)

[4. Représentations graphiques *13*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458739)

[4.1. Histogrammes des fréquences *13*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458740)

[4.2. Le nuage de mots *20*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458741)

[5. Visualiser les associations entre mots *22*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458742)

[5.1. Calculer les coefficients de corrélation *22*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458743)

[5.2. Représenter les associations sous forme d’un réseau *25*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458744)

[5.3. Regrouper les mots par classification ascendante hiérarchique *27*](file:///C:\Users\REDA\Desktop\ADT\Annexes_sushishop_vieuxlille.docx#_Toc71458745)

Ce fichier systématise la création d’un lexique à partir d’un ensemble de textes courts rassemblés dans un fichiers csv.

Ici on reprend les 133 avis sur Le restaurant sushishop Vieux Lille collectées sur Tripadvisor.

# 1. Importer la base

Ici, il s’agit :

* d’indiquer le répertoire de travail,
* de spécifier qu’il ne faut pas confondre caractères et modalités d’une variable catégorielle
* et d’importer un fichier csv dans R de manière à créer un corpus.

On vérifie que l’importation s’est convenablement réalisée en demandant le nombre de lignes (le nombre de courts textes), le nom des colonnes et les premières lignes du corpus.

setwd("C:/Users/REDA/Desktop/ADT")  
  
options(stringsAsFactors=F)  
  
mabase <- read.csv2(file="avis\_sushishop\_vl.csv", encoding="latin1")  
  
nrow(mabase)

## [1] 133

names(mabase)

## [1] "niveau" "Avis"

head(mabase)

## niveau  
## 1 excellent  
## 2 excellent  
## 3 excellent  
## 4 excellent  
## 5 excellent  
## 6 excellent  
## Avis  
## 1 Comme à chaque fois... un régal ! Un service très rapide et efficace. Rituel du dimanche soir, surtout pendant cette dur période, il est bon de faire vivre les commercants.  
## 2 Parfait comme d’habitude! Rien à redire ! On s’est régalé tous les quatre confinés à la maison ... ??  
## 3 Je commande régulièrement chez Sushi Shop et leurs produits sont très bons, très bonne qualités et les choix sont multiples. Je prend souvent la Box for two gourmet et elle est excellente, l'association des différents sushi est parfaite.  
## 4 Qualité Toujours au top ! Vente à emporter pendant le confinement. Commande prête rapidement et avec soin  
## 5 Merci d'avoir assuré un service minimum pendant cette période. Nous nous sommes régalés de délicieux sushis et cette petite pause évasion nous a mis du baume au coeur.  
## 6 Accueil parfait, personnel dynamique, serviable et souriant malgré l’affluence. Très bons produits. Excellent!

statistique <- summary(nchar(mabase$Avis))  
statistique

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 100 159 240 335 414 2121

# 2. Nettoyer les textes du corpus

Les textes d’avis collectés sur Internet sont toujours plein de fautes (orthographe) et, quel que soit le texte, tous les mots ne nous intéressent pas (il s’agit des mots-outils).

Pour nettoyer les textes, nous allons utiliser deux packages d’analyse textuelle : *tm* et *qdap*.

## 2.3. Modifier le corpus de textes

On enlève du corpus tous les mots-outils du dictionnaire modifié. On importe le dictionnaire sous format csv, on obtient une base dico. Cela a pour avantage de définir sans ambiguïté en “latin1” le vecteur / la colonne dico[,1] de mots-outils (comme déjà les textes de *mabase)*.

dico <- read.csv2(file="mots\_outils\_perso.csv", encoding="latin1")  
# on met les textes en minuscules car les mots outils sont en minuscules  
mabase$Avis <- tolower(mabase$Avis)  
  
# on enlève les mots outils  
mabase$Avis <- removeWords(mabase$Avis, dico[,1])

Il est pratique d’enlever toutes les lettres accentuées (les cédilles, etc.). En particulier dans des avis sur Internet, beaucoup de mots sont écrits avec ou sans accents. Du coup, en enlevant tous les accents, on uniformise le corpus… même si à terme cela peut compliquer la lecture.

Enlever les ponctuations.

Pour l’analyse de textes publiés, normalement sans faute ni coquille, cette opération est moins utile.

Unaccent <- function(Z) {  
 Z <- gsub("['`^~\"]", " ", Z)  
 Z <- iconv(Z, to="ASCII//TRANSLIT//IGNORE")  
 Z <- gsub("['`^~\"]", "", Z)  
 return(Z)  
}  
mabase$Avis <- Unaccent(mabase$Avis)  
#enkever les ponctuations  
mabase$Avis <- removePunctuation(mabase$Avis)  
#créer deux selection selon la qualité d'avis : excellent , horrible   
  
Ex<- mabase[mabase$niveau=="excellent",]  
Hor<-mabase[mabase$niveau=="horrible",]  
  
Exl=data.frame(Ex)  
Horr=data.frame(Hor)

Après avoir extraire deux sous bases : Avis excellent , Avis Horrible

Maintenant, on crée un premier les lexiques “brouillon” (lexic0) qui va nous permettre de repérer les erreurs qu’il faudra corriger ensuite.

Pour cela, il faut créer :

* un objet *corpus* du package *tm,*
* puis le tableau lexical entier (dit *tdm* pour *“term document matrix”*)
* et enfin le lexique avec le calcul des fréquences.

# on transforme Avis en table  
y <- data.frame(doc\_id=seq(1:nrow(mabase)), text=mabase$Avis)  
yex <- data.frame(doc\_id=seq(1:nrow(Exl)), text=Exl$Avis)  
yhor <- data.frame(doc\_id=seq(1:nrow(Horr)), text=Horr$Avis)  
# puis on crée l'objet corpus   
corpus <- SimpleCorpus(DataframeSource(y), control = list(language = "fr"))  
corpus\_yex <- SimpleCorpus(DataframeSource(yex), control = list(language = "fr"))  
corpus\_yhor<- SimpleCorpus(DataframeSource(yhor), control = list(language = "fr"))  
# on enlève les blancs inutiles (deux à la suite, etc.)  
corpus <- tm\_map(corpus, stripWhitespace)  
corpus\_yex <- tm\_map(corpus\_yex, stripWhitespace)  
corpus\_yhor <- tm\_map(corpus\_yhor, stripWhitespace)  
# on enlève les nombres en chiffres  
corpus <- tm\_map(corpus, removeNumbers)  
corpus\_yex <- tm\_map(corpus\_yex, removeNumbers)  
corpus\_yhor <- tm\_map(corpus\_yhor, removeNumbers)  
  
# on vérifie la lecture de l'avis 98  
as.list(corpus)[98]

## $`98`  
## [1] "tout titre grosse arnaque sushis secs tiennent gout plus moyenplateau incomplet prix bien cher bref a fuir"

# on crée le tableau lexical entier  
tdm <-TermDocumentMatrix(corpus)  
tdm\_yex <-TermDocumentMatrix(corpus\_yex)  
tdm\_yhor <-TermDocumentMatrix(corpus\_yhor)  
# on le transforme en objet matrice pour faire des calculs de fréquences  
tdm.mat <-as.matrix(tdm)  
tdm.mat\_yex <-as.matrix(tdm\_yex)  
tdm.mat\_yhor <-as.matrix(tdm\_yhor)  
  
# la dimension de la matrice nous renseigne sur le nombre de mots (formes) repérés dans le corpus  
dim(tdm.mat)

## [1] 1519 133

dim(tdm.mat\_yex)

## [1] 315 26

dim(tdm.mat\_yhor)

## [1] 776 38

# pour obtenir la fréquence d'un mot on somme ses occurences dans le tableau lexical entier (calcul possible car on l'a transformé en un objet matrice)  
term.freq <- rowSums(tdm.mat)  
term.freq\_yex <- rowSums(tdm.mat\_yex)  
term.freq\_yhor <- rowSums(tdm.mat\_yhor)  
  
# on crée la table lexique avec deux colonnes/variables : mot et freq  
lexic0 <-data.frame(mot=names(term.freq), freq=term.freq)  
lexic0\_yex <-data.frame(mot=names(term.freq\_yex), freq=term.freq\_yex)  
lexic0\_yhor <-data.frame(mot=names(term.freq\_yhor), freq=term.freq\_yhor)  
  
write.table(lexic0,"lexic0.csv", sep=";",row.names = F, col.names = T)  
write.table(lexic0\_yex,"lexic0\_yex.csv", sep=";",row.names = F, col.names = T)  
write.table(lexic0\_yhor,"lexic0\_yhor.csv", sep=";",row.names = F, col.names = T)

on repère les chaînes de caractères avec un problème.

On constate que beaucoup d’erreurs viennent du fait que cette méthode ne reconnaît que l’espace comme moyen de délimiter les formes lexicales. Quand dans le texte, il n’y a pas d’espace après une virgule, les deux mots avant et après la virgule ne font qu’une seule forme lexicale avec une virgule au milieu.

problemes <- c("sushis","adoooore","apparament","apparrement,.","appele","aurevoir","aventuree","avocat,saumon","/makis","bol","bols","bolw","bolws","bon,service","dessert.teste","edaname","ensemble.accueil","heureusement...message","jamais,surtout","makis/sushis/sashimis","moyen,plateau","ptit","puisqu","qualite/prix","quemander","qumqats","ramolo","shop...bon","suspect...mangue")  
  
corrections <- c("sushi","adore","apparement","apparement","appel","au revoir","aventure","avocat saumon","makis","bowl","bowls","bowl","bowls","bon service","dessert teste","edamame","ensemble acceuil","heureusement message","jamais surtout","makis sushis sashimis","moyen plateau","petit","puisque","qualite-prix","commander","kumquat","ramollo","shop bon","suspect mangue")  
  
mabase$Avis <- mgsub(problemes,corrections,mabase$Avis)  
Exl$Avis <- mgsub(problemes,corrections,Exl$Avis)  
Hor$Avis <- mgsub(problemes,corrections,Horr$Avis)

# 3. Les lexiques

* On enlève les ponctuations.
* Aux mots du lexique - en majuscules - on associe leur racine

# on transforme Avis en table  
y <- data.frame(doc\_id=seq(1:nrow(mabase)), text=mabase$Avis)  
yex <- data.frame(doc\_id=seq(1:nrow(Exl)), text=Exl$Avis)  
yhor <- data.frame(doc\_id=seq(1:nrow(Horr)), text=Horr$Avis)  
  
# puis on crée l'objet corpus   
corpus <- SimpleCorpus(DataframeSource(y), control = list(language = "fr"))  
corpus\_yex <- SimpleCorpus(DataframeSource(yex), control = list(language = "fr"))  
corpus\_yhor<- SimpleCorpus(DataframeSource(yhor), control = list(language = "fr"))  
  
# on enlève les blancs inutiles (deux à la suite, etc.)  
corpus <- tm\_map(corpus, stripWhitespace)  
corpus\_yex <- tm\_map(corpus\_yex, stripWhitespace)  
corpus\_yhor <- tm\_map(corpus\_yhor, stripWhitespace)  
  
# on enlève les nombres en chiffres  
corpus <- tm\_map(corpus, removeNumbers)  
corpus <- tm\_map(corpus, removeNumbers)  
corpus\_yex <- tm\_map(corpus\_yex, removeNumbers)  
corpus\_yhor <- tm\_map(corpus\_yhor, removeNumbers)  
  
# on enlève la ponctuation  
corpus <- tm\_map(corpus, removePunctuation)  
corpus\_yex <- tm\_map(corpus\_yex, removePunctuation)  
corpus\_yhor <- tm\_map(corpus\_yhor, removePunctuation)  
  
# on crée le tableau lexical entier  
tdm <-TermDocumentMatrix(corpus)  
tdm\_yex <-TermDocumentMatrix(corpus\_yex)  
tdm\_yhor <-TermDocumentMatrix(corpus\_yhor)  
  
# on le transforme en objet matrice pour faire des calculs de fréquences  
tdm.mat <-as.matrix(tdm)  
tdm.mat\_yex <-as.matrix(tdm\_yex)  
tdm.mat\_yhor <-as.matrix(tdm\_yhor)  
  
# la dimension de la matrice nous renseigne sur le nombre de mots (formes) repérés dans le corpus  
dim(tdm.mat)

## [1] 1509 133

dim(tdm.mat\_yex)

## [1] 314 26

dim(tdm.mat\_yhor)

## [1] 776 38

# pour obtenir la fréquence d'un mot on somme ses occurences dans le tableau lexical entier (calcul possible car on l'a transformé en un objet matrice)  
term.freq <- rowSums(tdm.mat)  
term.freq\_yex <- rowSums(tdm.mat\_yex)  
term.freq\_yhor <- rowSums(tdm.mat\_yhor)  
  
# on crée la table lexique avec trois colonnes/variables : mot (en majuscules), racine et freq  
word <- names(term.freq)  
word\_yex <- names(term.freq\_yex)  
word\_yhor <- names(term.freq\_yhor)  
  
# on repère la racine des mots   
racine <- stemDocument(word, language = "french")  
racine\_yex <- stemDocument(word\_yex , language = "french")  
racine\_yhor <- stemDocument(word\_yhor, language = "french")  
  
# on met les mots entiers en majuscules  
lexique <-data.frame(mot=toupper(word), racine, freq=term.freq)  
lexique\_Excellent <-data.frame(mot=toupper(word\_yex),racine\_yex, freq=term.freq\_yex)  
lexique\_Horrible <-data.frame(mot=toupper(word\_yhor), racine\_yhor, freq=term.freq\_yhor)  
  
# on sauvegarde le lexique pour être lu et travaillé sous excel  
write.table(lexique,"lexique.csv", sep=";",row.names = F, col.names = T)  
write.table(lexique\_Excellent,"lexique\_Excellent.csv", sep=";",row.names = F, col.names = T)  
write.table(lexique\_Horrible,"lexique\_Horrible.csv", sep=";",row.names = F, col.names = T)

On peut ajouter les fréquences des mots racines. Cela permet par exemple de faire remonter tous les termes qui évoquent ce qui est “petit” (qui englobent toutes les formes fléchies : petit, petits, petite, petites)

corpus2 <- tm\_map(corpus, stemDocument)  
corpus2\_yex <- tm\_map(corpus\_yex, stemDocument)  
corpus2\_yhor <- tm\_map(corpus\_yhor, stemDocument)  
  
tdm2 <-TermDocumentMatrix(corpus2)  
tdm2\_yex <-TermDocumentMatrix(corpus2\_yex)  
tdm2\_yhor <-TermDocumentMatrix(corpus2\_yhor)  
  
tdm2.avis <-as.matrix(tdm2)  
tdm2.avis\_yex <-as.matrix(tdm2\_yex)  
tdm2.avis\_yhor <-as.matrix(tdm2\_yhor)  
  
dim(tdm2.avis)

## [1] 1296 133

dim(tdm2.avis\_yex)

## [1] 286 26

dim(tdm2.avis\_yhor)

## [1] 693 38

term.freq <- rowSums(tdm2.avis)  
term.freq\_yex <- rowSums(tdm2.avis\_yex)  
term.freq\_yhor <- rowSums(tdm2.avis\_yhor)  
  
f\_racines <-data.frame(racine=names(term.freq), freq=term.freq)  
f\_racines\_yex <-data.frame(racine=names(term.freq\_yex), freq=term.freq\_yex)  
f\_racines\_yhor <-data.frame(racine=names(term.freq\_yhor), freq=term.freq\_yhor)

Imprimons les lignes des lexiques .

library(knitr)   
  
# on trie lexique par odre décroissant des valeurs de la 3e colonne, freq  
lexique <- lexique[order(lexique[,3], decreasing=T),]  
lexique\_Excellent <- lexique\_Excellent[order(lexique\_Excellent[,3], decreasing=T),]  
lexique\_Horrible <- lexique\_Horrible[order(lexique\_Horrible[,3], decreasing=T),]  
  
impression <- lexique[which(lexique$freq > 19 ) ,]  
impression\_Excellent <- lexique\_Excellent[which(lexique\_Excellent$freq >5 ) ,]   
impression\_Horrible <- lexique\_Horrible[which(lexique\_Horrible$freq > 5 ) ,]   
  
  
knitr::kable(impression)

|  |  |  |  |
| --- | --- | --- | --- |
|  | mot | racine | freq |
| sushi | SUSHI | sush | 133 |
| commande | COMMANDE | command | 81 |
| plus | PLUS | plus | 61 |
| tres | TRES | tre | 60 |
| qualite | QUALITE | qualit | 39 |
| shop | SHOP | shop | 37 |
| livraison | LIVRAISON | livraison | 37 |
| bien | BIEN | bien | 33 |
| tout | TOUT | tout | 32 |
| service | SERVICE | servic | 31 |
| prix | PRIX | prix | 29 |
| fois | FOIS | fois | 28 |
| frais | FRAIS | frais | 28 |
| fait | FAIT | fait | 26 |
| peu | PEU | peu | 23 |
| restaurant | RESTAURANT | restaur | 23 |
| car | CAR | car | 22 |
| bon | BON | bon | 20 |
| rien | RIEN | rien | 20 |
| produits | PRODUITS | produit | 20 |
| emporter | EMPORTER | emport | 20 |
| deux | DEUX | deux | 20 |

knitr::kable(impression\_Excellent)

|  |  |  |  |
| --- | --- | --- | --- |
|  | mot | racine\_yex | freq |
| sushi | SUSHI | sush | 19 |
| tres | TRES | tre | 13 |
| commande | COMMANDE | command | 10 |
| livraison | LIVRAISON | livraison | 9 |
| fois | FOIS | fois | 6 |
| rapide | RAPIDE | rapid | 6 |

knitr::kable(impression\_Horrible)

|  |  |  |  |
| --- | --- | --- | --- |
|  | mot | racine\_yhor | freq |
| commande | COMMANDE | command | 35 |
| plus | PLUS | plus | 21 |
| sushis | SUSHIS | sush | 19 |
| sushi | SUSHI | sush | 19 |
| tout | TOUT | tout | 17 |
| livraison | LIVRAISON | livraison | 15 |
| shop | SHOP | shop | 13 |
| tres | TRES | tre | 13 |
| deux | DEUX | deux | 12 |
| service | SERVICE | servic | 11 |
| restaurant | RESTAURANT | restaur | 11 |
| prix | PRIX | prix | 10 |
| bien | BIEN | bien | 10 |
| commandes | COMMANDES | command | 10 |
| alors | ALORS | alor | 9 |
| boutique | BOUTIQUE | boutiqu | 9 |
| temps | TEMPS | temp | 9 |
| attente | ATTENTE | attent | 9 |
| rien | RIEN | rien | 9 |
| qualite | QUALITE | qualit | 9 |
| clients | CLIENTS | client | 8 |
| riz | RIZ | riz | 8 |
| fois | FOIS | fois | 8 |
| livreur | LIVREUR | livreur | 8 |
| apres | APRES | apre | 8 |
| fait | FAIT | fait | 8 |
| derniere | DERNIERE | dernier | 7 |
| aucun | AUCUN | aucun | 7 |
| etre | ETRE | etre | 7 |
| chez | CHEZ | chez | 7 |
| minutes | MINUTES | minut | 7 |
| produits | PRODUITS | produit | 6 |
| client | CLIENT | client | 6 |
| moins | MOINS | moin | 6 |
| avoir | AVOIR | avoir | 6 |
| telephone | TELEPHONE | telephon | 6 |
| lille | LILLE | lill | 6 |
| chaine | CHAINE | chain | 6 |
| accueil | ACCUEIL | accueil | 6 |

On produit aussi le tableau des racines de mots.

# on trie f\_racines par odre décroissant des valeurs de la 2e colonne, freq  
f\_racines <- f\_racines[order(f\_racines[,2], decreasing=T),]  
f\_racines\_yex <- f\_racines\_yex[order(f\_racines\_yex[,2], decreasing=T),]  
f\_racines\_yhor <- f\_racines\_yhor[order(f\_racines\_yhor[,2], decreasing=T),]  
  
impress <- f\_racines[which(f\_racines$freq > 19 ) ,]   
impress\_Excellent <- f\_racines\_yex[which(f\_racines\_yex$freq > 5 ) ,]   
impression\_Horrible <- f\_racines\_yhor[which(f\_racines\_yhor$freq > 5 ) ,]   
  
knitr::kable(impress)

|  |  |  |
| --- | --- | --- |
|  | racine | freq |
| sushi | sushi | 133 |
| command | command | 106 |
| plus | plus | 61 |
| tres | tres | 60 |
| qualit | qualit | 41 |
| livraison | livraison | 40 |
| shop | shop | 37 |
| tout | tout | 36 |
| bon | bon | 34 |
| bien | bien | 33 |
| servic | servic | 31 |
| fait | fait | 30 |
| prix | prix | 29 |
| foi | foi | 28 |
| frai | frai | 28 |
| restaur | restaur | 27 |
| peu | peu | 23 |
| produit | produit | 22 |
| accueil | accueil | 22 |
| car | car | 22 |
| emport | emport | 21 |
| rien | rien | 20 |
| deux | deux | 20 |

knitr::kable(impress\_Excellent)

|  |  |  |
| --- | --- | --- |
|  | racine | freq |
| sushi | sushi | 19 |
| tres | tres | 13 |
| command | command | 13 |
| excel | excel | 9 |
| livraison | livraison | 9 |
| rapid | rapid | 8 |
| foi | foi | 6 |

knitr::kable(impression\_Horrible)

|  |  |  |
| --- | --- | --- |
|  | racine | freq |
| command | command | 46 |
| sushi | sushi | 38 |
| plus | plus | 21 |
| tout | tout | 18 |
| livraison | livraison | 17 |
| client | client | 14 |
| shop | shop | 13 |
| restaur | restaur | 13 |
| tres | tres | 13 |
| deux | deux | 12 |
| servic | servic | 11 |
| aucun | aucun | 11 |
| prix | prix | 10 |
| bien | bien | 10 |
| dernier | dernier | 10 |
| alor | alor | 9 |
| boutiqu | boutiqu | 9 |
| temp | temp | 9 |
| attent | attent | 9 |
| livreur | livreur | 9 |
| rien | rien | 9 |
| qualit | qualit | 9 |
| appel | appel | 8 |
| bon | bon | 8 |
| riz | riz | 8 |
| foi | foi | 8 |
| apr | apr | 8 |
| fait | fait | 8 |
| seul | seul | 7 |
| autr | autr | 7 |
| etr | etr | 7 |
| chez | chez | 7 |
| minut | minut | 7 |
| employ | employ | 7 |
| arriv | arriv | 6 |
| produit | produit | 6 |
| desagr | desagr | 6 |
| faut | faut | 6 |
| moin | moin | 6 |
| personn | personn | 6 |
| avoir | avoir | 6 |
| telephon | telephon | 6 |
| lill | lill | 6 |
| chain | chain | 6 |
| accueil | accueil | 6 |

# 4. Représentations graphiques

On charge les packages suivants.

library(ggplot2)

##   
## Attaching package: 'ggplot2'

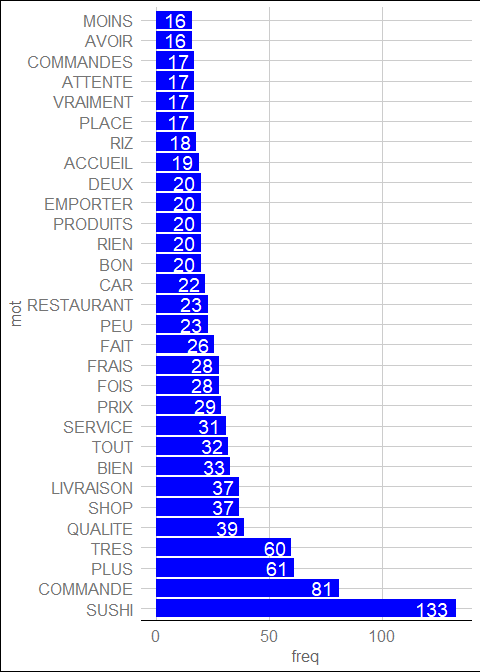
## The following object is masked from 'package:qdapRegex':  
##   
## %+%

## The following object is masked from 'package:NLP':  
##   
## annotate

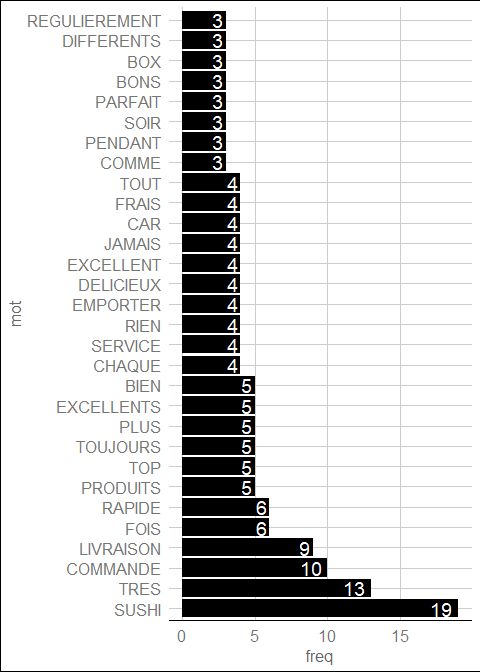
library(ggthemes)  
library(wordcloud)

## 4.1. Histogrammes des fréquences

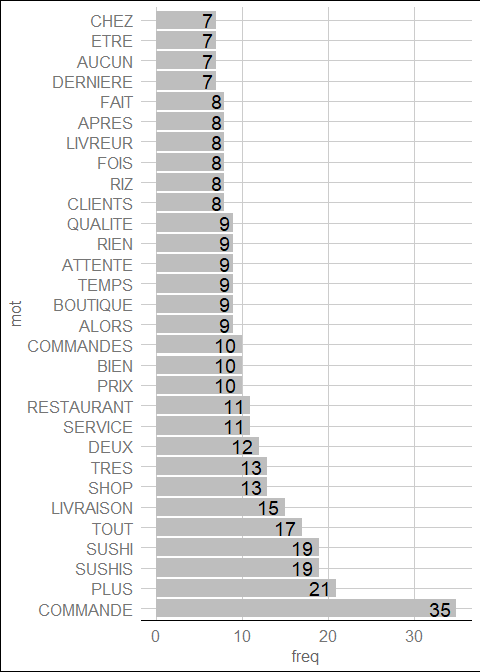
# tri de lexique par odre décroissant des valeurs de la 3e colonne, freq  
lexique <- lexique[order(lexique[,3], decreasing=T),]  
lexique\_Excellent <- lexique\_Excellent[order(lexique\_Excellent[,3], decreasing=T),]  
lexique\_Horrible <- lexique\_Horrible[order(lexique\_Horrible[,3], decreasing=T),]  
  
# création d'une nouvelle table lexfreq avec :  
# transformation de la variable "mot" (chaînes de caractères) en une variable "mot" catégorielle dont chaque modalité reprend exactement la forme des mots du lexique  
  
lexfreq <- lexique  
lexfreq$mot <- factor(lexfreq$mot, levels=unique(as.character(lexfreq$mot)))  
  
lexfreq\_yex <- lexique\_Excellent  
lexfreq\_yex$mot <- factor(lexfreq\_yex$mot, levels=unique(as.character(lexfreq\_yex$mot)))  
  
lexfreq\_yhor <- lexique\_Horrible  
lexfreq\_yhor$mot <- factor(lexfreq\_yhor$mot, levels=unique(as.character(lexfreq\_yhor$mot)))  
  
  
  
ggplot(lexfreq[1:30,], aes(x=mot, y=freq))+geom\_bar(stat="identity",  
fill='blue')+coord\_flip()+theme\_gdocs()+geom\_text(aes(label=freq),  
colour="white",hjust=1.25, size=5.0)



ggplot(lexfreq\_yex[1:30,], aes(x=mot, y=freq))+geom\_bar(stat="identity",  
fill='black')+coord\_flip()+theme\_gdocs()+geom\_text(aes(label=freq),  
colour="white",hjust=1.25, size=5.0)

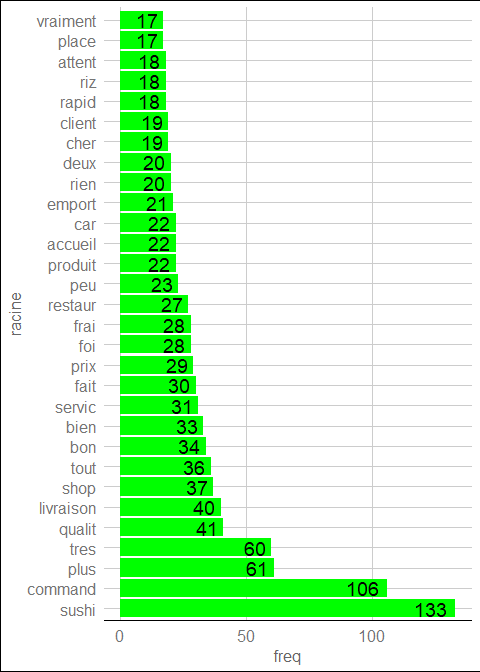


ggplot(lexfreq\_yhor[1:30,], aes(x=mot, y=freq))+geom\_bar(stat="identity",  
fill='grey')+coord\_flip()+theme\_gdocs()+geom\_text(aes(label=freq),  
colour="black",hjust=1.25, size=5.0)

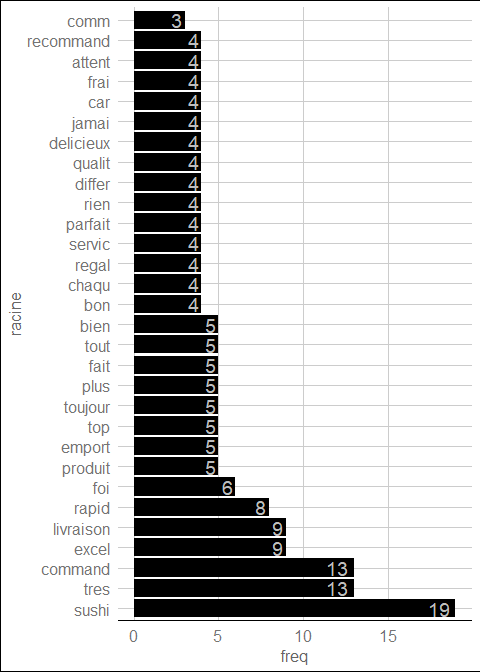


Même chose pour créer un histogramme des racines .

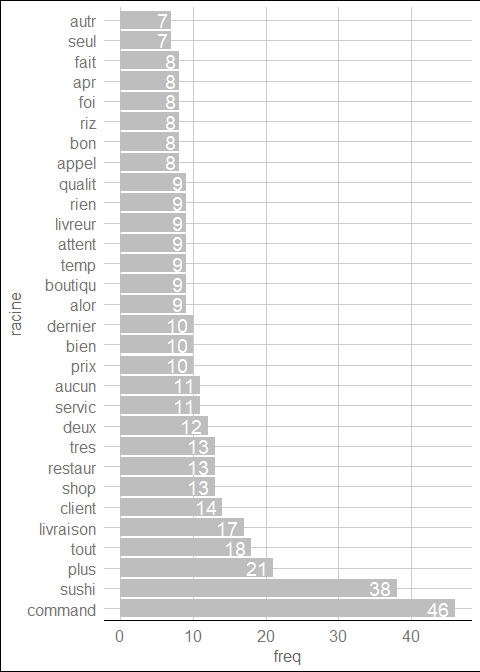
f\_racines <- f\_racines[order(f\_racines[,2], decreasing=T),]  
f\_racines\_yex <- f\_racines\_yex[order(f\_racines\_yex[,2], decreasing=T),]  
f\_racines\_yhor <- f\_racines\_yhor[order(f\_racines\_yhor[,2], decreasing=T),]  
  
f\_rac <- f\_racines  
f\_rac$racine<-factor(f\_rac$racine, levels=unique(as.character(f\_rac$racine)))  
f\_rac\_yex <- f\_racines\_yex  
f\_rac\_yex$racine<-factor(f\_rac\_yex$racine, levels=unique(as.character(f\_rac\_yex$racine)))  
f\_rac\_yhor <- f\_racines\_yhor  
f\_rac\_yhor$racine<-factor(f\_rac\_yhor$racine, levels=unique(as.character(f\_rac\_yhor$racine)))  
  
ggplot(f\_rac[1:30,], aes(x=racine, y=freq))+geom\_bar(stat="identity",  
fill='green')+coord\_flip()+theme\_gdocs()+geom\_text(aes(label=freq),  
colour="black",hjust=1.25, size=5.0)



ggplot(f\_rac\_yex[1:30,], aes(x=racine, y=freq))+geom\_bar(stat="identity",  
fill='black')+coord\_flip()+theme\_gdocs()+geom\_text(aes(label=freq),  
colour="grey",hjust=1.25, size=5.0)

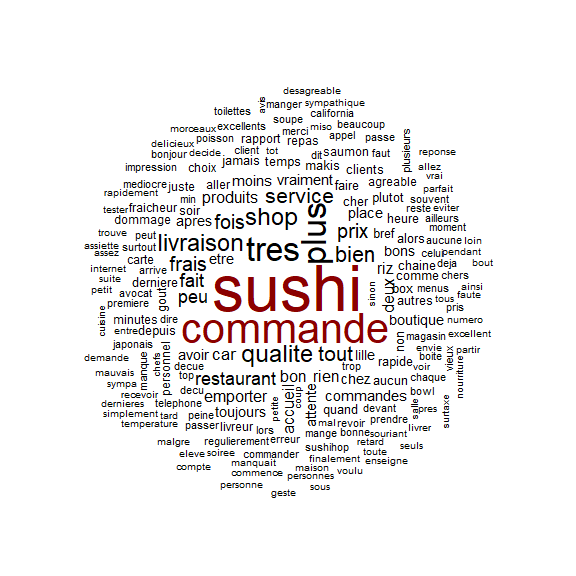


ggplot(f\_rac\_yhor[1:30,], aes(x=racine, y=freq))+geom\_bar(stat="identity",  
fill='grey')+coord\_flip()+theme\_gdocs()+geom\_text(aes(label=freq),  
colour="white",hjust=1.25, size=5.0)



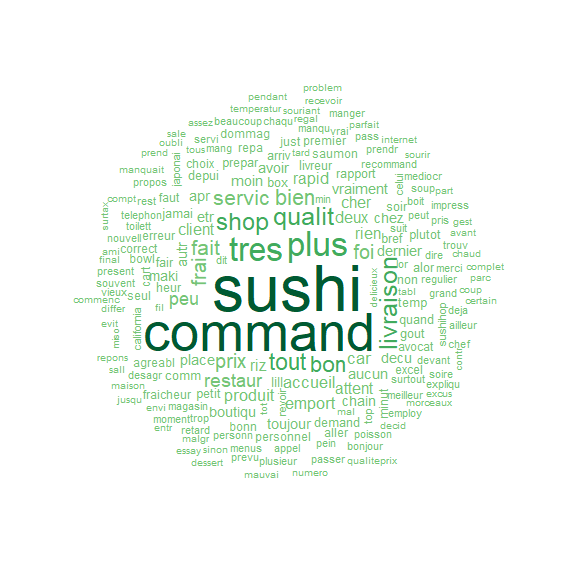
## 4.2. Le nuage de mots

library(wordcloud)  
  
lexfreq <- lexique  
# on passe en minuscules, plus lisible dans un nuage de mots  
lexfreq$mot <- tolower(lexfreq$mot)  
  
wordcloud(lexfreq$mot,lexfreq$freq, max.words = 200, random.order=FALSE, colors=c('black','darkred'))



Même chose avec les racines

pal <- brewer.pal(8, "Greens")  
pal <- pal[-(1:4)]  
  
wordcloud(f\_racines$racine,f\_racines$freq, max.words = 200, random.order=FALSE, colors=pal)



# 5. Visualiser les associations entre mots

## 5.1. Calculer les coefficients de corrélation

tdm.mat[1:20,1:10]

## Docs  
## Terms 1 2 3 4 5 6 7 8 9 10  
## bon 1 0 0 0 0 0 0 0 0 0  
## chaque 1 0 0 0 0 0 1 0 0 0  
## comme 1 1 0 0 0 0 0 0 0 0  
## commercants 1 0 0 0 0 0 0 0 0 0  
## dimanche 1 0 0 0 0 0 0 0 0 0  
## dur 1 0 0 0 0 0 0 0 0 0  
## efficace 1 0 0 0 0 0 0 0 0 0  
## faire 1 0 0 0 0 0 0 0 0 0  
## fois 1 0 0 0 0 0 1 0 0 1  
## pendant 1 0 0 1 1 0 0 0 0 0  
## periode 1 0 0 0 1 0 0 0 0 0  
## rapide 1 0 0 0 0 0 1 1 0 0  
## regal 1 0 0 0 0 0 0 0 0 0  
## rituel 1 0 0 0 0 0 0 0 0 0  
## service 1 0 0 0 1 0 1 0 0 0  
## soir 1 0 0 0 0 0 0 0 0 0  
## surtout 1 0 0 0 0 0 0 0 0 0  
## tres 1 0 2 0 0 1 1 0 0 1  
## vivre 1 0 0 0 0 0 0 0 0 0  
## confines 0 1 0 0 0 0 0 0 0 0

tdm.mat\_yex[1:20,1:10]

## Docs  
## Terms 1 2 3 4 5 6 7 8 9 10  
## bon 1 0 0 0 0 0 0 0 0 0  
## chaque 1 0 0 0 0 0 1 0 0 0  
## comme 1 1 0 0 0 0 0 0 0 0  
## commercants 1 0 0 0 0 0 0 0 0 0  
## dimanche 1 0 0 0 0 0 0 0 0 0  
## dur 1 0 0 0 0 0 0 0 0 0  
## efficace 1 0 0 0 0 0 0 0 0 0  
## faire 1 0 0 0 0 0 0 0 0 0  
## fois 1 0 0 0 0 0 1 0 0 1  
## pendant 1 0 0 1 1 0 0 0 0 0  
## periode 1 0 0 0 1 0 0 0 0 0  
## rapide 1 0 0 0 0 0 1 1 0 0  
## regal 1 0 0 0 0 0 0 0 0 0  
## rituel 1 0 0 0 0 0 0 0 0 0  
## service 1 0 0 0 1 0 1 0 0 0  
## soir 1 0 0 0 0 0 0 0 0 0  
## surtout 1 0 0 0 0 0 0 0 0 0  
## tres 1 0 2 0 0 1 1 0 0 1  
## vivre 1 0 0 0 0 0 0 0 0 0  
## confines 0 1 0 0 0 0 0 0 0 0

tdm.mat\_yhor[1:20,1:10]

## Docs  
## Terms 1 2 3 4 5 6 7 8 9 10  
## alors 1 0 0 0 0 0 0 0 0 0  
## appele 1 0 0 0 0 0 0 0 1 0  
## arrive 1 0 0 0 0 0 0 0 0 0  
## aussi 1 0 0 0 0 0 0 0 0 0  
## avocat 1 1 0 0 0 0 0 0 0 0  
## bons 1 0 0 0 0 0 0 0 0 0  
## boutique 1 0 0 0 0 0 0 0 0 0  
## clients 1 0 0 0 0 0 0 0 0 0  
## commande 2 1 0 0 2 2 1 2 0 0  
## completement 1 0 0 0 0 0 0 0 0 0  
## decouvrant 1 0 0 0 0 0 0 0 0 0  
## deuxieme 1 0 0 0 0 0 0 0 0 0  
## directement 1 0 0 0 0 0 0 0 0 0  
## disant 1 0 0 0 0 0 0 0 0 0  
## egalement 1 0 0 0 0 0 0 0 0 0  
## excuse 1 0 0 0 0 0 0 0 1 0  
## gamme 1 0 0 0 0 0 0 0 0 0  
## gyozas 2 0 0 0 0 0 0 0 0 0  
## haut 1 0 0 0 0 0 0 0 0 0  
## inacceptable 1 0 0 0 0 0 1 0 0 0

#tdm.mat["commande",]  
#tdm.mat["livraison",]  
#tdm.mat["acceuil",]  
#tdm.mat["emporter",]

Pour quantifier l’association entre deux mots, on calcule le coefficient de corrélation.

motA <- tdm.mat["commande",]   
motB <- tdm.mat["livraison",]   
  
AssociationAB <- cor(motA,motB)  
AssociationAB

## [1] 0.3106079

On construit aisément une matrice de corrélation entre un ensemble de mots.

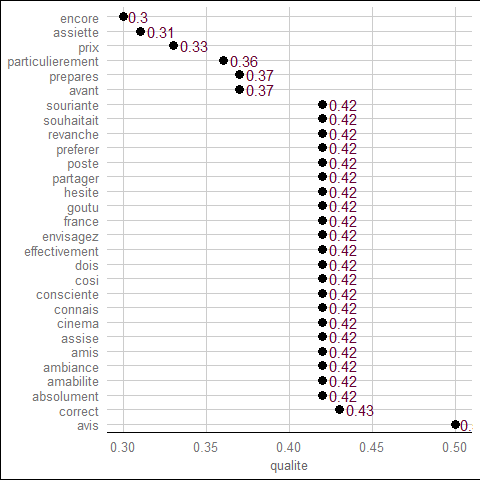
listmots <- c("sushi", "commande", "livraison", "qualite", "accueil","plus")  
extrait <- tdm.mat[listmots,]  
extrait <- t(extrait)  
  
mat.cor <- cor(extrait)  
  
library(knitr)   
knitr::kable(round(mat.cor, 2)) # round(x, 2) = arrondir la valeur x à 2 chiffres après la virgule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | sushi | commande | livraison | qualite | accueil | plus |
| sushi | 1.00 | 0.25 | 0.08 | 0.05 | 0.09 | 0.29 |
| commande | 0.25 | 1.00 | 0.31 | -0.02 | -0.03 | 0.24 |
| livraison | 0.08 | 0.31 | 1.00 | -0.16 | -0.12 | 0.10 |
| qualite | 0.05 | -0.02 | -0.16 | 1.00 | 0.21 | 0.07 |
| accueil | 0.09 | -0.03 | -0.12 | 0.21 | 1.00 | 0.09 |
| plus | 0.29 | 0.24 | 0.10 | 0.07 | 0.09 | 1.00 |

associations <- findAssocs(tdm, 'qualite', 0.3)  
MotsAssoc <- as.data.frame(associations)  
MotsAssoc$termes <- row.names(MotsAssoc)  
MotsAssoc$termes <- factor(MotsAssoc$termes, levels=MotsAssoc$termes)

La même information sous forme d’un graphique :

ggplot(MotsAssoc, aes(y=termes)) +  
geom\_point(aes(x=qualite), data=MotsAssoc,  
size=3)+  
theme\_gdocs()+ geom\_text(aes(x=qualite,label=qualite), colour= "#660033",   
hjust=-.25,size=4)+theme(text=element\_text(size=10),  
axis.title.y=element\_blank())



## 5.2. Représenter les associations sous forme d’un réseau

Pour que le réseau soit lisible, il faut diminuer le nombre de mots. On va garder les mots les plus fréquents. Pour cela :

tdm.mat2 <- data.frame(tdm.mat)  
  
tdm.mat2$tot <- rowSums(tdm.mat2)   
  
tdm.mat2 <- tdm.mat2[order(tdm.mat2$tot, decreasing=T),]  
  
tdm.mat2 <- as.matrix(tdm.mat2[1:8,1:20])

on crée le réseau.

# install.packages("igraph")  
library(igraph)

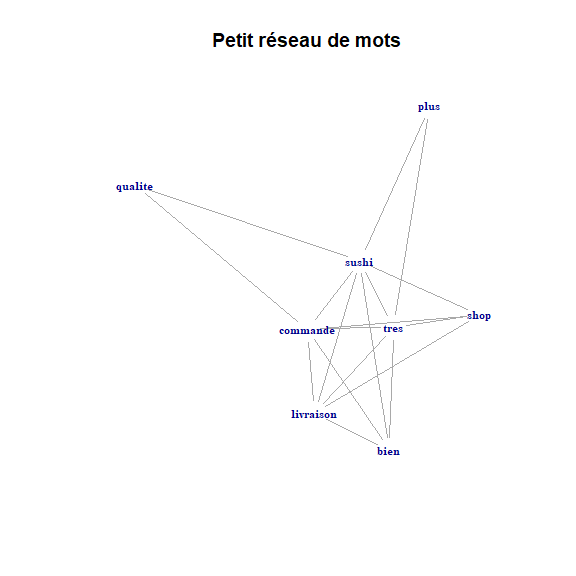
##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:qdap':  
##   
## %>%, diversity

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

tableau.adj <- tdm.mat2 %\*% t(tdm.mat2) # produit de la matrice avec elle-même (transposée)  
tableau.adj <- graph.adjacency(tableau.adj, weighted=TRUE,   
 mode="undirected", diag=T)  
tableau.adj<-simplify(tableau.adj) # élimine les arêtes redondantes  
  
plot.igraph(tableau.adj, vertex.shape="none",  
vertex.label.font=2, vertex.label.color="darkblue",  
vertex.label.cex=.7, edge.color="darkgrey")  
title(main='Petit réseau de mots')



## 5.3. Regrouper les mots par classification ascendante hiérarchique

tdm.reduit <- removeSparseTerms(tdm, sparse=0.95)  
  
arbre <- hclust(dist(tdm.reduit, method="euclidean"), method="complete")  
  
plot(arbre, main='Arborescence')

