www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

Le Bulletin de la Dialyse à Domicile

Introduction à l'analyse de données avec le logiciel R

Claire Della Vedova

1087 chemin de Sainte Roustagne, 04100 MANOSQUE, France

NDLR : Le RDPLF a pour but principal d'être une aide pour permettre aux équipes de dialyse à domicile d'évaluer leurs pratiques cliniques et également conduire des études à partir d'exports anonymisés des données qu'elles saisissent. Toute évaluation ou étude à partir de données biologiques ou cliniques nécessite le recours à des calculs statistiques. Si les modèles complexes relèvent du spécialiste, nombre de calculs de base, tels que moyenne, médiane, etc.. sont à la portée de tous moyennant un minimum effort. Il existe un logiciel gratuit, «R», extrêmement puissant, qu'il est facile d'installer sur tout ordinateur et qui permet les calculs les plus simples comme les plus compliqués en fonction de ses compétences.

Nous avons pensé que les lecteurs du BDD, infirmières comme médecins seraient intéressés à suivre une formation de base qui leur donnera l'autonomie pour des études simples ou compliquées qu'ils ou elles souhaiteraient mener. Le RDPLF se tient par ailleurs à leur disposition pour leur adresser tout fichier d'export anonymisé.

Nous débutons avec ce numéro une série d'articles rédigée par Claire Della Vedova que nous remercions vivement pour l'aide apportée.

Claire est Ingénieur en biostatistique / data analyste, Elle utilise quotidiennement le logiciel R pour analyser des données. Elle a travaillé pendant plus de 15 ans dans les domaines de l'environnement et de la santé, et a formé de nombreux étudiants et chercheurs à l'utilisation de R. Elle anime depuis novembre 2017 le blog Statistique et Logiciel R dont le but est d'aider les débutants à mieux appréhender les méthodes statistiques classiques et à utiliser le logiciel R plus efficacement, notamment au travers de tutoriels : https://statistique-et-logiciel-r.com/.

Ce premier article est réservé à l'instatllation du logiciel gratuit R.

La formation totale se fera sur 15 mois, au rythme d'un article par trimestre à chaque parution du BDD. Cela laissera largement le temps d'assimiler et tester les connaissances acquises entre chaque article. Pour ceux qui souhaiteront aller plus vite, ils pourront aller sur le blog (https://statistique-et-logiciel-r.com/).

Dates des prochaines parutions :
- article 2 (septembre 2019) : la réalisation de représentations visuelles avec l'add on esquisse
- article 3 (Décembre 2019) : une initiation à ggplot2
- article 4 (Avril 2020) : la réalisation de rapports d'analyses statistiques automatisées avec Rmarkdown
- article 5 (Juin 2020) : la manipulation de données (avec dplyr, notamment les fonction group_by et summarise)
- article 6 (Septembre 2020) : la réalisation d'analyses descriptives (paramètres statistiques et graphes) sous forme de dashboard avec le package flexboard

Mots clés : biostatistique, épidémiologie, logiciel R, RDPLF

journal officiel du Registre de Dialyse Péritonéale de Langue Française RDPLF www.rdplf.org

www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

1. C'est quoi R ?

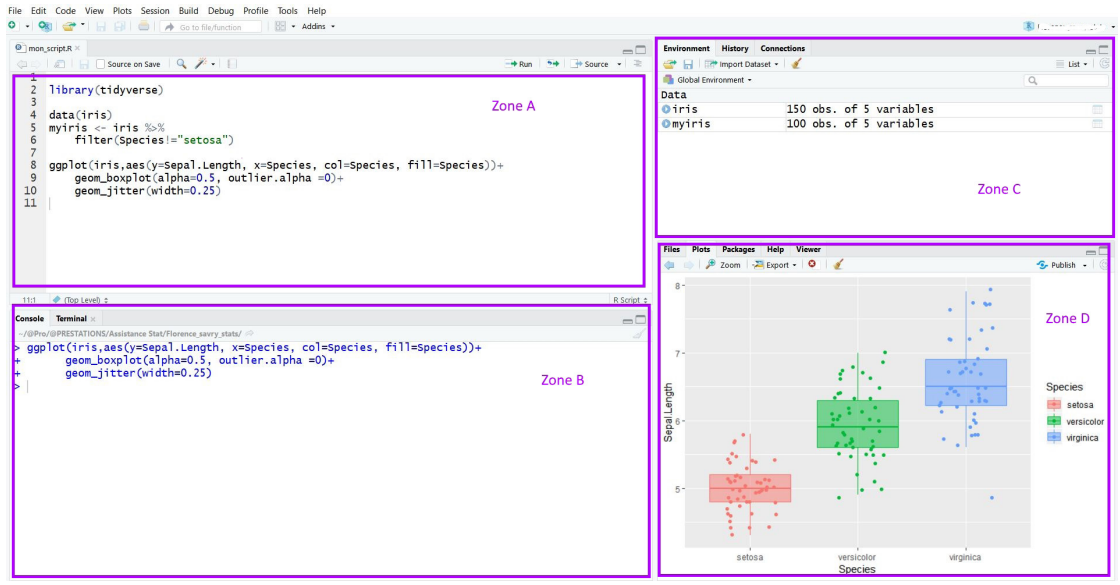
A l'origine, R était un logiciel de statistique. Aujourd'hui, compte tenu de son évolution, il est davantage qualifié de logiciel de data science, car on peut également l'utiliser pour faire de la data visualisation, du machine learning, de la cartographie, des rapports d'analyse automatisés, des dashboards, ou encore des applications web (avec shiny).

R est très largement utilisé dans le domaine médical, car il permet de réaliser des analyses descriptives performantes, d'employer de multiples tests d'hypothèses (pour comparer des moyennes ou des pourcentages par exemple), ou d'employer de nombreux modèles de régression (régression linéaire multiple, régression logistique, modèles de survie etc…).

Mais R est aussi un langage de programmation ! En pratique cela veut dire qu'il fonctionne, non pas avec des boutons, ou des menus déroulants (comme Excel par exemple), mais principalement avec des lignes de commandes. Ces lignes de commandes comportent des sortes de "mots clé" qui sont des fonctions implémentées (c'est-à-dire prêtes à être employées). Par exemple, la fonction mean() permet de calculer une moyenne. Il est, bien sûr, également possible de coder ses propres fonctions.
Cet usage des lignes de commandes peut faire un peu peur au début. Mais d'une part, il existe un environnement pour R, qui s'appelle R Studio, qui facilite grandement son utilisation. Et d'autre part, mon expérience dans l'accompagnement des débutants, me montre qu'en investissant un peu de temps dans l'apprentissage du langage, il est tout à fait possible de devenir autonome assez rapidement.
Enfin, R est un logiciel gratuit, qui est disponible pour les systèmes d'exploitation Windows, MacOS et Linux.

1.2. Comment ça fonctionne

Pour utiliser R, il faut, en premier lieu, télécharger et installer R, puis faire de même avec R Studio (cette démarche est explicitée en détail plus loin). Ensuite, R est utilisé au travers de R Studio. Il s'agit d'une interface graphique qui se compose de quatre fenêtres (ou zones) :



La zone A est la zone dédiée à l'édition de codes R. C'est à son niveau que se déroule l'ouverture, la création ou la modification de scripts de commandes R. Ces scripts sont en .R ou .Rmd.

La zone B est la console du logiciel R ; elle permet l'exécution de codes. Les lignes de commandes peuvent être directement entrées dans la console, ou bien transférées de la zone A à la zone B par un copier-coller, ou par le raccourci Ctrl+Entrée après s'être positionné sur la ligne.

www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

La zone C permet notamment d'avoir accès aux objets présents dans la mémoire de R, ainsi que les jeux de données importées ou créés. Cette zone contient également un outil d'importation des données via le menu déroulant Import Dataset.

La zone D permet, entre autres, d'avoir accès à l'outil de téléchargement des packages, à une fenêtre permettant de visualiser des graphiques, un navigateur de fichiers (comme sous Windows), ou encore d'accéder à la page d'aide des fonctions.

Les fonctions utilisées par R sont regroupées dans des paquets, nommés packages. Ces packages sont développés individuellement, par des spécialistes du domaine auquel ils s'intéressent. Ils sont mis à disposition de la communauté R, généralement sur le site de CRAN (dans l'onglet package du menu de gauche), mais parfois aussi sur le compte GitHub du développeur. Les packages contenant les fonctions de base sont téléchargés et installés automatiquement avec le logiciel R (c'est le cas des packages stats, graphics, grDevices, datasets, methods, base). Les autres packages doivent être téléchargés et installés volontairement. Nous installerons dans la suite de cet article, le package funModelling afin d'utiliser certaines des fonctions qu'il contient pour réaliser une analyse descriptive de variables qualitatives (ou catégorielles).

1.3. Pourquoi utiliser R plutôt qu'Excel pour analyser des données ?

D'abord, parce que Excel n'est pas particulièrement intuitif pour faire des analyses statistiques, qu'elles soient descriptives ou comparatives. Ensuite, parce que les analyses statistiques réalisables sous Excel sont extrêmement limitées. Il en est de même des possibilités graphiques.
Du côté de R, les possibilités sont quasi illimitées ! Non seulement en termes d'analyses statistiques, mais aussi en termes de data visualisation, de reporting (dashboard, rapports d'analyses automatisés), et de ressources libre d'accès.

Et puis R est très employé dans le domaine médical ; savoir l'utiliser est devenue une compétence importante et recherchée.

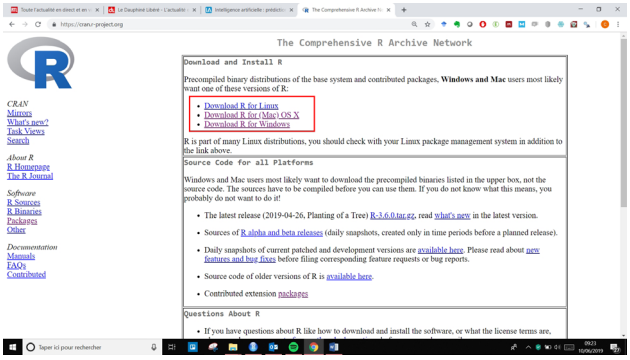
Et enfin, comme dit précédemment R est gratuit !

2. Installation de R et RStudio

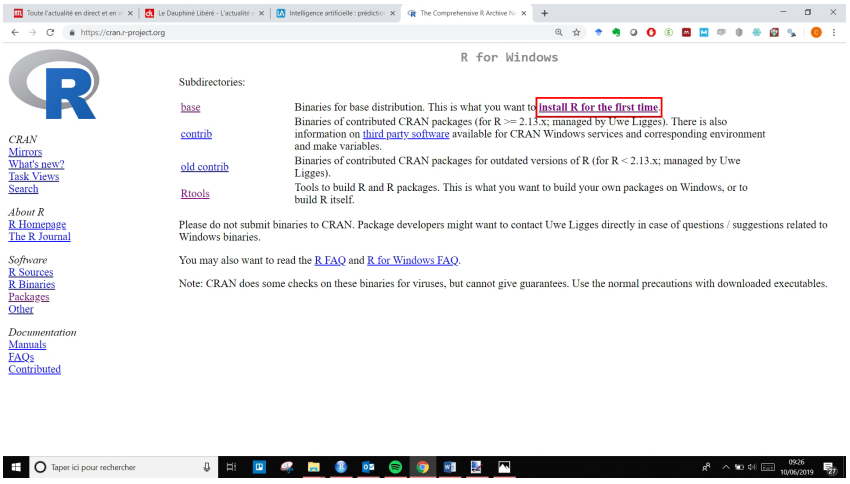
L'installation de R et R studio se déroule en 5 étapes :
1) Téléchargement du logiciel R
2) Installation du logiciel R
3) Téléchargement de R studio
4) Installation de R studio
5) Ouverture de R studio

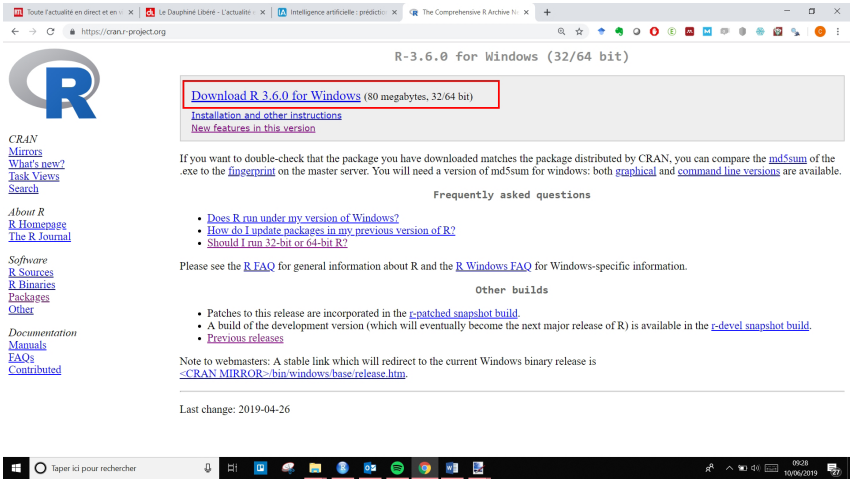
2.1 Téléchargement du logiciel R:

R est accessible sur le site du CRAN : The Comprehensive R Archive Network (https://cran.r-project.org/). Pour le télécharger, il suffit de ce rendre sur ce site, et de suivre la démarche présentée ici en pas à pas pour Windows. La procédure pour les autres systèmes d'exploitation est sensiblement identique :



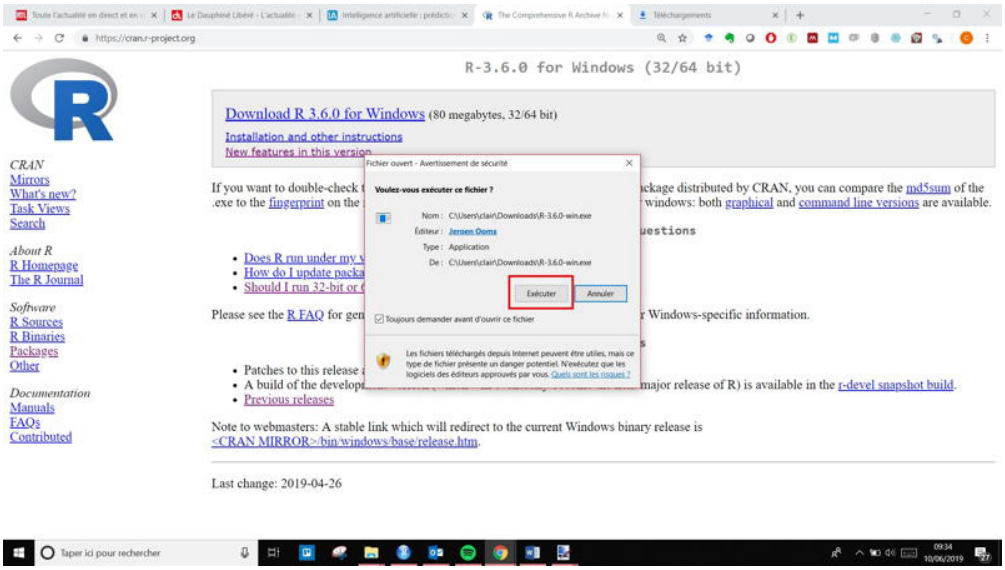
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917



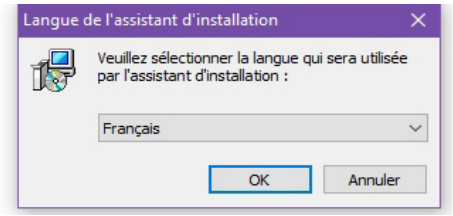


2.2 Installation du logiciel R

Une fois le fichier téléchargé, il est nécessaire d'aller le chercher dans le dossier de téléchargement, et de double-cliquer dessus pour l'exécuter.



www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

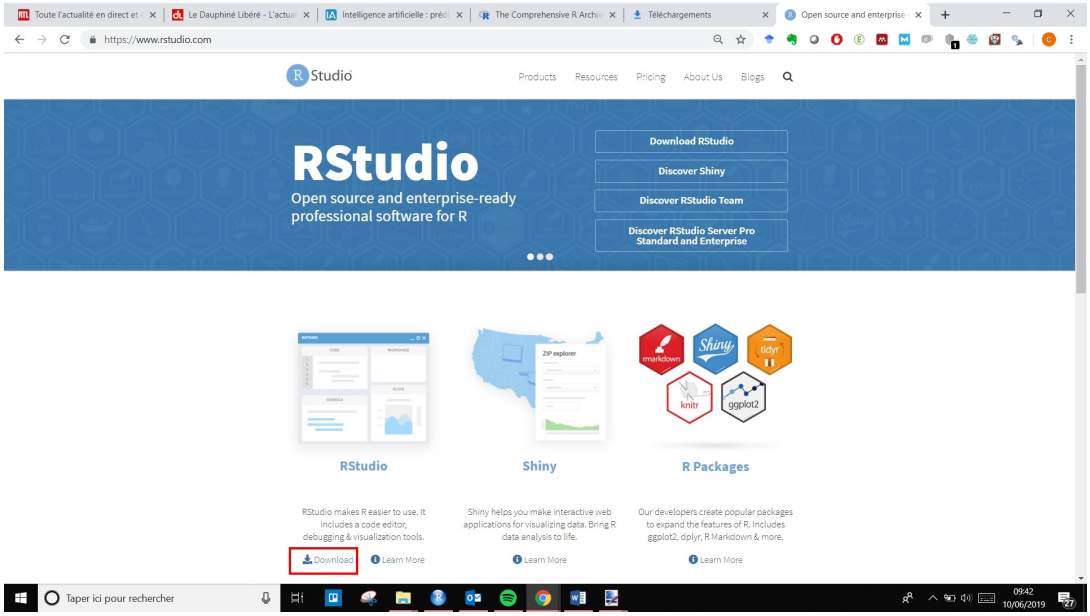


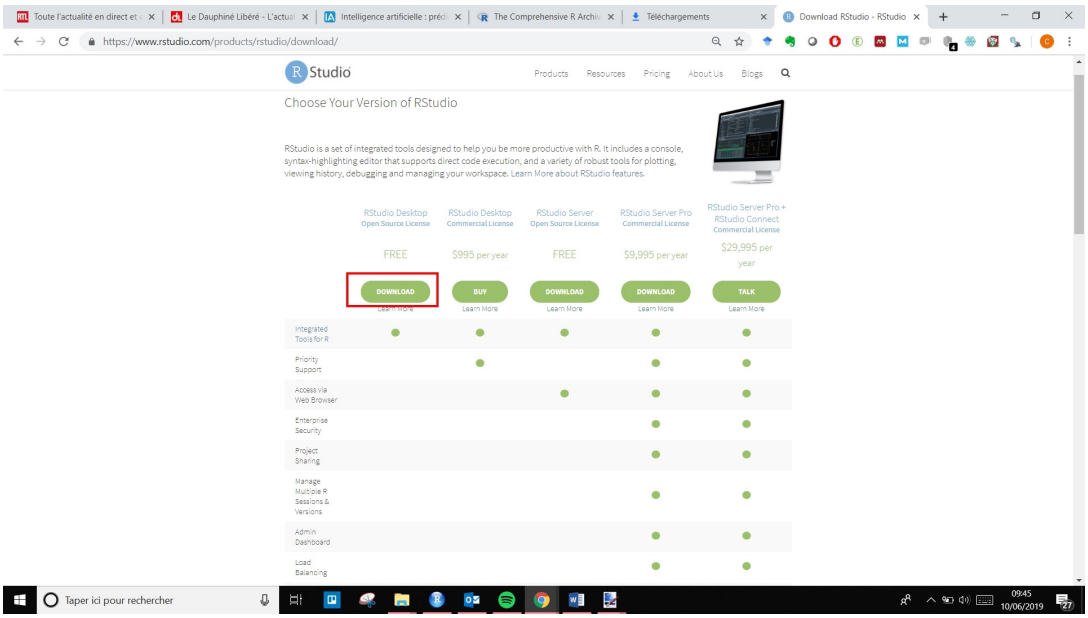
Normalement, la langue sélectionnée par défaut est le français. Si ce n'est pas le cas, vous pouvez choisir "Français" dans le menu déroulant.

Pour la suite, il suffit de toujours cliquer sur suivant, en acceptant les options par défaut, jusqu'à la fin de l'installation.

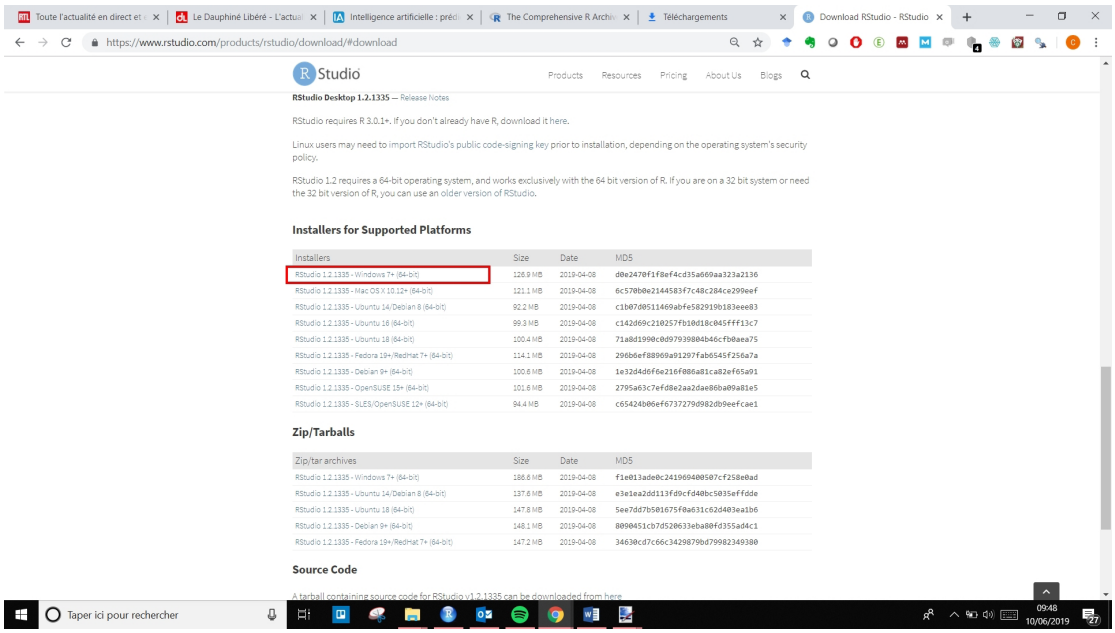
2.3 Téléchargement de R Studio

R studio se télécharge à partir du site de R studio (https://www.rstudio.com/). Là encore, la démarche est présentée en pas à pas, pour Windows.

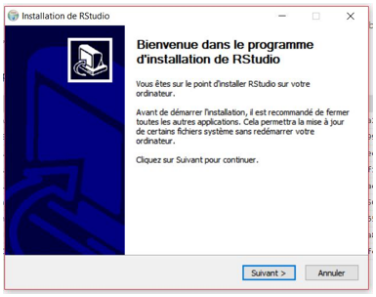




www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917



2.4 Installlation de RStudio

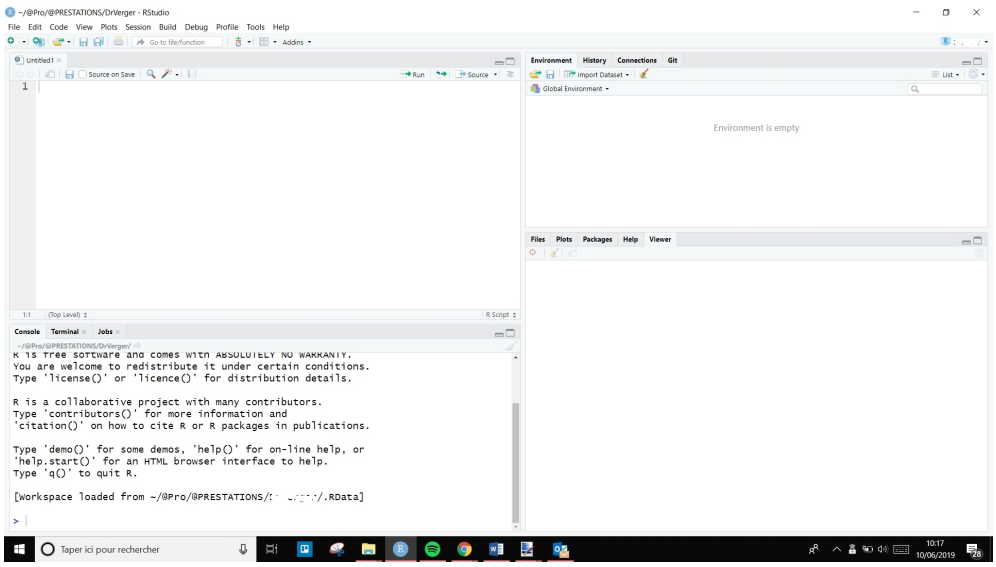


A ce stade, il faut aller chercher, dans le dossier de Téléchargement, le fichier de R studio que l'on vient de télécharger, puis de double cliquer dessus pour commencer son installation.

2.5 Ouverture de R et RStudio

Pour utiliser R, à partir de R studio, il suffit à présent d'ouvrir R studio. Si l'installation s'est correctement déroulée, une icône de R studio est à présent visible dans le menu déroulant de Windows, ou bien dans la liste des applications installées. Ouvrez R studio en cliquant dessus.

Vous devriez obtenir un écran similaire à celui-ci :



www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

Pas de panique si vous n'avez pas quatre fenêtres, il suffit de cliquer sur l'icône en haut à droite (de la fenêtre) pour l'ouvrir. Ne vous inquiétez pas non plus s'il vous manque l'onglet Git dans la zone C (en haut à gauche), c'est normal !

Rassurez-vous, tout est prêt pour utiliser R et RStudio.

3. Premiers pas avec R :

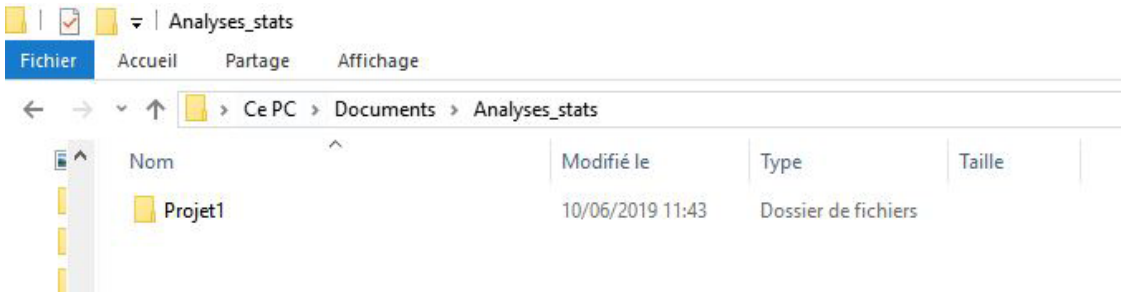
En guise de premiers pas, nous allons importer un jeu de données et faire une première analyse descriptive de celui-ci.

Mais avant de commencer à proprement parlé, nous allons créer un projet R. Un projet R est simplement un moyen d'encapsuler son travail. Ca consiste à associer un dossier de travail de l'ordinateur à R Studio. Cela est très pratique, car par défaut, le répertoire de travail du logiciel R sera à la racine de ce dossier. Ainsi, à chaque fois que vous sauvegarderez un script ou un graphique, par défaut cela se fera dans ce dossier.

3.1 Création d'un projet R

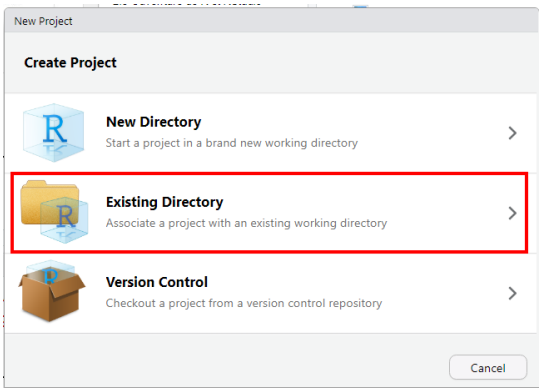
- 3.1.1 Création d'un dossier de travail

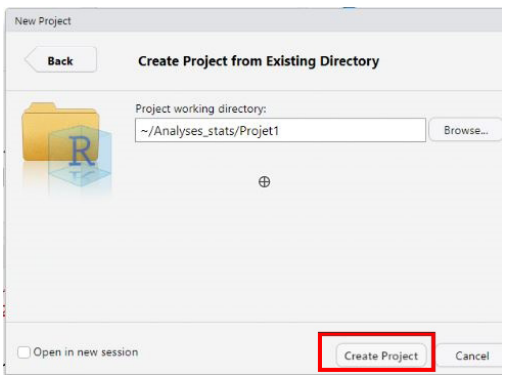
Pour cela, créez, où vous le souhaitez sur votre ordinateur, un dossier de travail.
Par exemple, ici, j'ai créé (avec l'explorateur Windows) un dossier "Analyses_stats" dans "Document". Et dans ce dossier "Analyses_stats", j'ai créé un dossier "Projet1". C'est ce dossier que je vais associer à R.



- 3.1.2 Association du dossier à R Studio

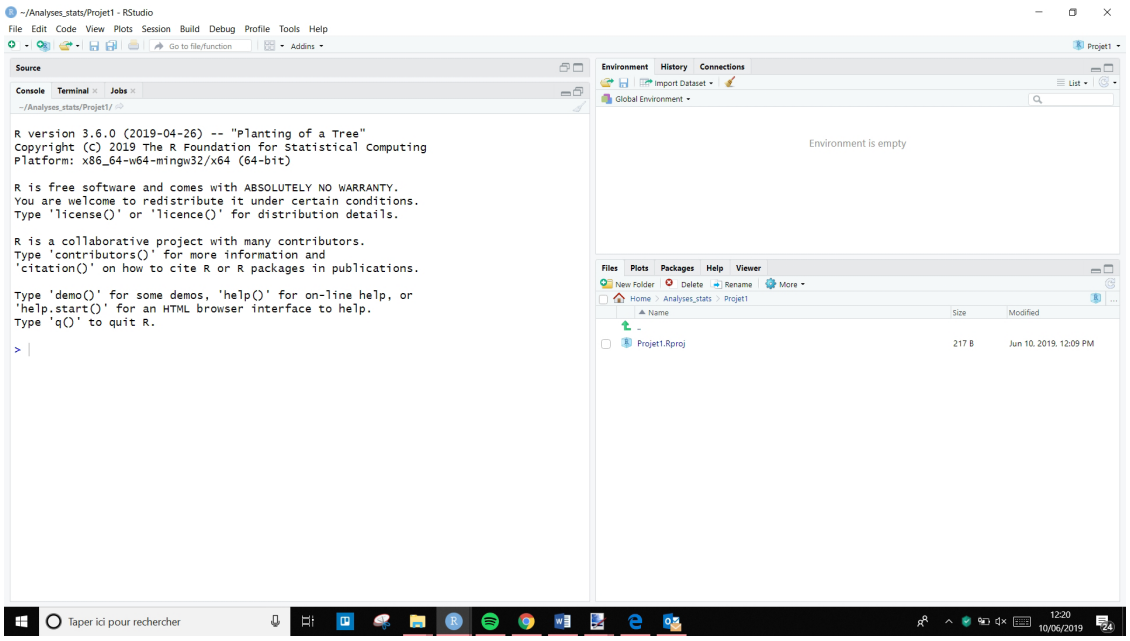
Pour cela, dans R Studio, allez dans le menu File (en haut à gauche), puis choisissez "New Project",et indiquez votre dossier de travail :





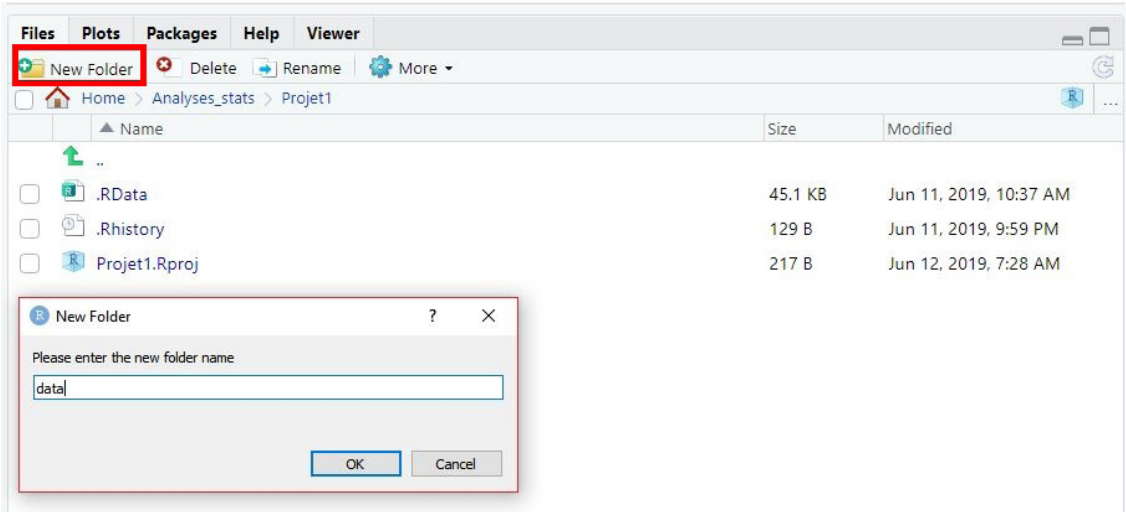
www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

Vous devriez obtenir un écran identique à celui-ci:



- 3.1.3 Créer un dossier data

La dernière étape consiste à créer un dossier “data”, à l’intérieur de dossier “Projet1”. Comme son nom l’indique, c’est dans ce dossier “data”, que seront placés les jeux de données à importer dans R pour les analyser. Vous pouvez créer ce dossier “data” dans le dossier “projet1” associé à R Studio, à l’aide de l’explorateur Windows (de la même façon que vous avez créé le dossier Projet1), ou bien en utilisant l’explorateur de RStudio, dans la zone D.



3.2 Quelques éléments à connaître pour bien débuter

Avant de commencer à proprement parler, quelques éléments de l’utilisation de R sont importants à connaitre :

1. Une commande peut être entrée dans la console R lorsque le prompt, c’est-à-dire le signe “>” est présent. Lorsque ce signe est absent, c’est qu’une commande n’est pas achevée. Pour retrouver le prompt, la touche Echap peut être utilisée.

www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

2. R est sensible à la casse. Cela signifie qu'une lettre Majuscule n'est pas équivalente à une lettre minuscule. Il faut donc être attentif sur ce point, lorsqu'on écrit du code. Dans l'exemple ci-dessous lorsque la fonction Mean est écrite avec une majuscule, R renvoie une erreur car mean doit être écrit en minuscule.

3. Il est possible de réutiliser une commande R exécutée précédemment, en utilisant la flèche du haut du clavier.

4. Sous R, les décimales sont des points.

5. La flèche d'assignation est utilisée pour créer un objet. Ici, l'objet "a" prend la valeur 5.

```
a <- 5
```

6. Pour obtenir de l'aide sur une fonction, on utilise la syntaxe ***?fonction***, par exemple avec la fonction ***mean()*** qui calcule la moyenne. La page d'aide s'ouvre alors dans la zone D (en bas à droite).

```
?mean()
```

7. Pour connaître les fonctions disponibles, on utilise la fonction ***apropos()***. Par exemple pour connaître toutes les fonctions disponibles qui contiennent le mot "mean".

```
apropos(«mean»)
##  [1] «.colMeans»      «.rowMeans»      «colMeans»       «cummean»
##  [5] «kmeans»         «mean»           «mean.Date»      «mean.default»
##  [9] «mean.difftime»  «mean.POSIXct»   «mean.POSIXlt»   «mean_cl_boot»
## [13] «mean_cl_normal» «mean_sdl»       «mean_se»        «rowMeans»
## [17] «weighted.mean»
```

3.3 Importation de données

Les données à importer sont contenues dans un fichier csv, nommé "FichierExempleStat" extrait de la base de données du RDPLF. Il est téléchargeable ici : https://www.rdplf.org/exempleR/FichierExempleStat.csv

Placez les dans le dossier "data" créé précédemment.

	A	B	C	D	E	F	G	H	I
1	code post	PAYS	sexe	Age 1ere DP	Charlson	Charlson_mo	Type de DP	Taille	Poids
2	56100	FRANCE	F	20,13	3	3	DPCA	132	30
3	1006	TUNISIE	F	50,74	3	2	DPCA	131	32,5
4	2540	LUXEMBOUR	M	18,86	5	5	DPA quotidie	140	33
5	3200	FRANCE	F	50,86	6	5	DPA quotidie	142	35
6	31603	FRANCE	F	54,07	6	5	DPCA	150	35
7	74374	FRANCE	F	53,89	4	3	DPCA	145	37
8	20420	MAROC	F	22,25	2	2	DPCA	148	37
9	25000	FRANCE	F	86,5	7	3	DPCA	147	38
10	10000	MAROC	F	18,22	2	2	DPA quotidie	136	39
11	83100	FRANCE	F	83,75	7	3	DPCA	139	39

www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

Pour importer le fichier de données au format csv dans R, nous allons ici utiliser la fonction *read.csv2()*, et nommer le fichier importé "mydata". En pratique, le résultat de la fonction *read.csv2()* est assigné à mydata, qui de ce fait va contenir les données.

mydata <- **read.csv2**(«**data/FichierExempleStat.csv**»)

Par défaut, la fonction *read.csv2()*, considère :

- que le jeu de données contient des noms de variables (cela s'appelle des header),
- que le séparateur des données est un point virgule (c'est ce qui est utilisé dans un fichier csv européen),
- que le séparateur de décimales est une virgule (c'est aussi ce qui est employé par défaut dans un fichier csv européen).

Si le fichier à importer ne contient pas ces caractéristiques, alors il sera nécessaire de le préciser dans la fonction. Pour plus d'informations, vous pouvez consulter cet article :

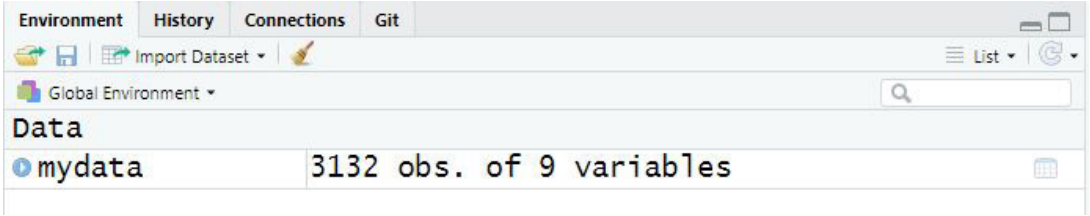
https://statistique-et-logiciel-r.com/nettoyer-et-valider-les-donnees-avec-r/

Une fois l'importation réalisée, il est important de vérifier que tout s'est bien déroulé. La fonction head() peut alors être employée pour afficher les six premières lignes du jeu de données :
head(mydata)

```
Head (mydata)

##   code.post       PAYS sexe Age.1ere.DP Charlson Charlson_modif
## 1     56100     FRANCE    F       20.13        3              3
## 2      1006    TUNISIE    F       50.74        3              2
## 3      2540 LUXEMBOURG    M       18.86        5              5
## 4      3200     FRANCE    F       50.86        6              5
## 5     31603     FRANCE    F       54.07        6              5
## 6     74374     FRANCE    F       53.89        4              3
##        Type.de.DP Taille Poids
## 1            DPCA    132  30.0
## 2            DPCA    131  32.5
## 3 DPA quotidienne    140  33.0
## 4 DPA quotidienne    142  35.0
## 5            DPCA    150  35.0
## 6            DPCA    145  37.0
```

Ici, tout semble conforme !
RStudio a également un tableau qui lui est intégré, qui permet de visualiser, mais aussi de trier ou de filtrer les données (comme sous Excel). Pour ouvrir ce tableur, il suffit de cliquer sur les données, dans l'onglet Environnement de la zone C:
Le tableau s'ouvre alors dans la zone A :



https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

mydata

Filter

	code.post	PAYS	sexe	Age.1ere.DP	Charlson	Charlson_modif	Type.de.DP	Taille	Poids
1	56100	FRANCE	F	20.13	3	3	DPCA	132	30.0
2	1006	TUNISIE	F	50.74	3	2	DPCA	131	32.5
3	2540	LUXEMBOURG	M	18.86	5	5	DPA quotidienne	140	33.0
4	3200	FRANCE	F	50.86	6	5	DPA quotidienne	142	35.0
5	31603	FRANCE	F	54.07	6	5	DPCA	150	35.0
6	74374	FRANCE	F	53.89	4	3	DPCA	145	37.0
7	20420	MAROC	F	22.25	2	2	DPCA	148	37.0
8	25000	FRANCE	F	86.50	7	3	DPCA	147	38.0
9	10000	MAROC	F	18.22	2	2	DPA quotidienne	136	39.0

4 Vérification et description des données

A présent que les données sont correctement importées, il est nécessaire de les vérifier. En pratique cela signifie contrôler que toutes les lignes et toutes les colonnes sont présentes, que les données ont bien le bon format (par exemple qu'une variable quantitative est bien considérée comme une variable numérique, et pas comme du texte par exemple), et encore de rechercher la présence d'éventuelle données aberrantes.

4.1 Contrôle de la structure avec la fonction str()

str (mydata)

```
> str(mydata)
'data.frame':   3132 obs. of  9 variables:
 $ code.post     : Factor w/ 177 levels "10000","1006",..: 84 2 35 49 48 131 28 34 1 151 ...
 $ PAYS          : Factor w/ 6 levels "BELGIQUE","FRANCE",..: 2 6 3 2 2 2 4 2 4 2 ...
 $ sexe          : Factor w/ 2 levels "F","M": 1 1 2 1 1 1 1 1 1 1 ...
 $ Age.1ere.DP   : num  20.1 50.7 18.9 50.9 54.1 ...
 $ Charlson      : int  3 3 5 6 6 4 2 7 2 7 ...
 $ Charlson_modif: int  3 2 5 5 5 3 2 3 2 3 ...
 $ Type.de.DP    : Factor w/ 4 levels "DPA quotidienne",..: 2 2 1 1 2 2 2 2 1 2 ...
 $ Taille        : num  132 131 140 142 150 145 148 147 136 139 ...
 $ Poids         : num  30 32.5 33 35 35 37 37 38 39 39 ...
```

La fonction "str" permet de vérifier :

- que les données sont bien enregistrées dans un tableau de données (data.frame en R)
- les nombres de lignes (ici 3132) et de colonnes (ici 9)
- la nature des variables :
  - "Factor" pour une variable catégorielle,
  - "num" ou "int" pour des variables numériques.

4.2 Description des données quantatives

La fonction *summary()* est très utile pour décrire les données et mettre en évidence d'éventuelles valeurs aberrantes. En effet, cette fonction renvoie, pour les variables numériques, la plus petite valeur (min), la plus grande valeur (max), la moyenne (mean), la médiane (med), ainsi que le premier et le 3ème quartile..

Pour plus d'informations sur les analyses descriptives de variables quantitatives, vous pouvez consulter cet article : http://bit.ly/2XPUGC9 ainsi que celui-là : http://bit.ly/2wTJQPC

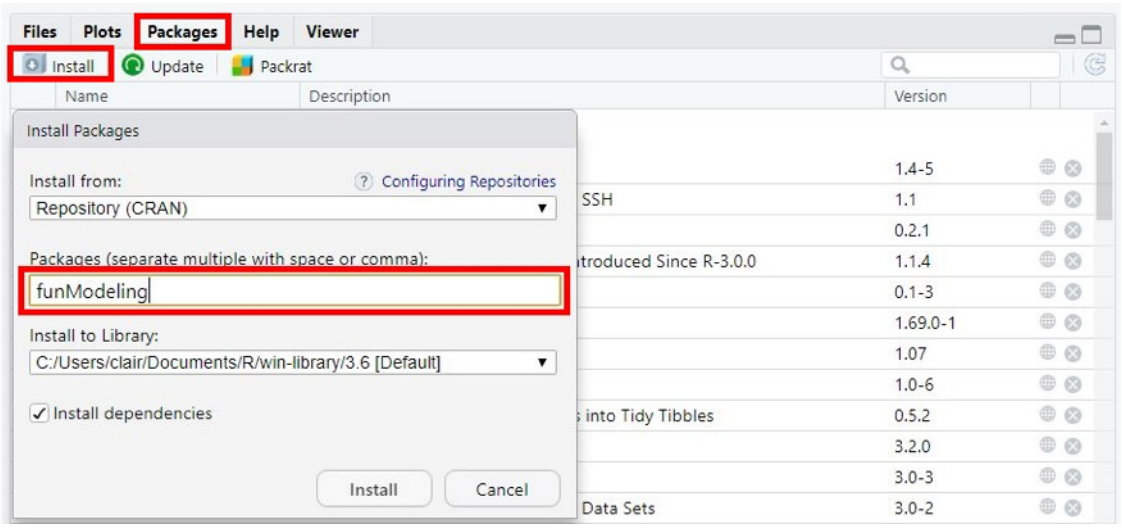
www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

summary(mydata)

```
> summary(mydata)
   code.post             PAYS       sexe      Age.1ere.DP       Charlson
 10000  :  85   BELGIQUE  : 244   F:1265   Min.   :16.01   Min.   : 2.000
 4000   :  70   FRANCE    :2639   M:1867   1st Qu.:54.77   1st Qu.: 4.000
 98849  :  60   LUXEMBOURG:  10            Median :68.10   Median : 6.000
 14033  :  54   MAROC     :  99            Mean   :65.07   Mean   : 5.843
 75877  :  52   SUISSE    :  33            3rd Qu.:77.69   3rd Qu.: 8.000
 63400  :  51   TUNISIE   : 107            Max.   :98.39   Max.   :16.000
 (Other):2760
 Charlson_modif            Type.de.DP       Taille          Poids
 Min.   :-2.000   DPA quotidienne:1348   Min.   :100.0   Min.   : 30.00
 1st Qu.: 2.000   DPCA           :1782   1st Qu.:160.0   1st Qu.: 61.00
 Median : 3.000   DPI            :   1   Median :166.0   Median : 72.00
 Mean   : 3.708   IND            :   1   Mean   :166.1   Mean   : 72.63
 3rd Qu.: 5.000                          3rd Qu.:173.0   3rd Qu.: 83.00
 Max.   :14.000                          Max.   :196.0   Max.   :128.00
```

4.3 Description des données qualitatives

Le package *funModeling* propose des fonctions particulièrement intéressantes pour décrire (et même visualiser) les variables catégorielles (ou qualitatives), ici les variables “code.post”, “PAYS”, “sexe”, et “Type.de.DP”
Nous allons commencer par télécharger et installer ce package. Pour le télécharger, nous pouvons utiliser l’outil présent dans la zone D de R Studio :



Pour pouvoir utiliser les fonctions présentes dans ce package, il est nécessaire, au préalable, de le charger dans R, à l’aide de la commande suivante :
library(funModeling)

www.bdd.rdplf.org Volume 2, n° 2, Juin 2019
https://doi.org/10.25796/bdd.v2i2.20513
ISSN 2607-9917

Ensuite, nous utilisons la fonction freq() afin d'obtenir, pour chaque modalité (ou niveau d'une variable donnée) :

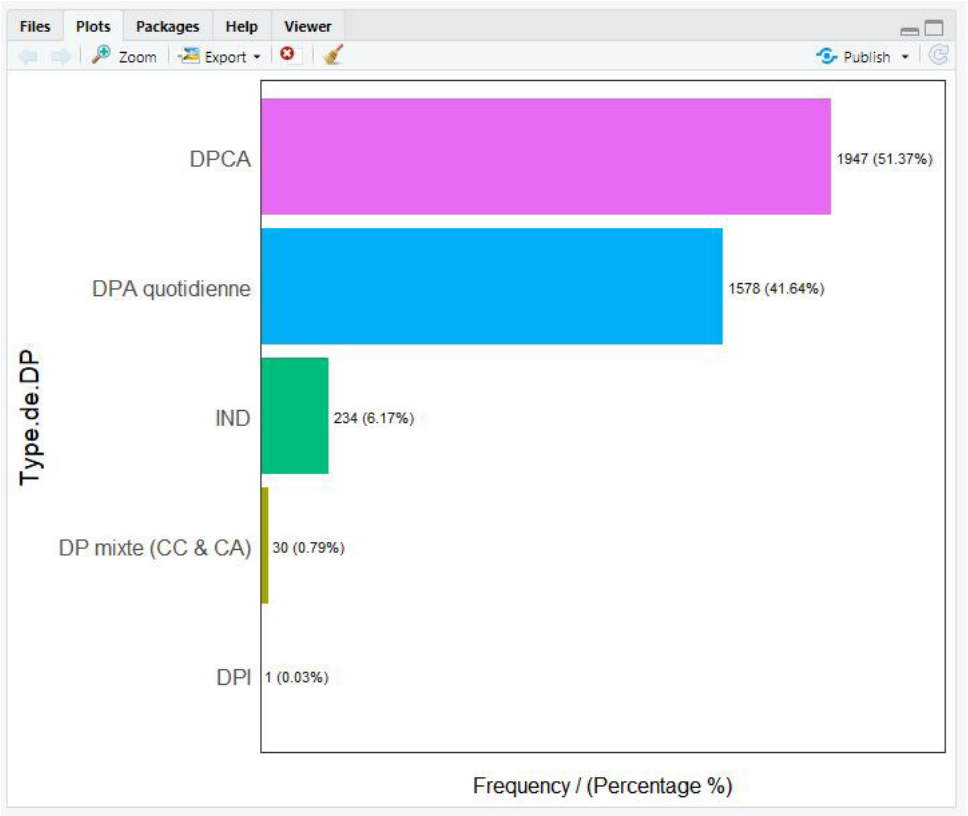
- le nombre de données
- le pourcentage représenté
- le pourcentage cumulé

Dans la ligne de commande ci-dessous, la syntaxe ["Type.de.DP"] permet de préciser la variable que l'on souhaite étudier.

```
freq(mydata[«Type.de.DP»])
```

```
##         Type.de.DP frequency percentage cumulative_perc
## 1             DPCA      1782      56.90           56.90
## 2 DPA quotidienne      1348      43.04           99.94
## 3              DPI         1       0.03           99.97
## 4              IND         1       0.03          100.00
```

La fonction *freq()* créé également, de façon automatique, une représentation graphique des données étudiées



Si aucune variable n'est définie en argument de la fonction freq(), celle-ci est automatiquement appliquée à toutes les variables catégorielles.

5. Pour conclure

J'espère que ce premier article d'introduction à R vous aura convaincu de faire vos premiers pas avec ce logiciel, et qu'il vous aura donné envie d'en savoir plus. Le prochain article sera dédié à la visualisation des données, c'est-à-dire à la réalisation de graphiques. Plus tard, nous verrons comment faire du reporting avec R. Parce qu'analyser ses données, c'est bien. Mais sortir un rapport d'analyse c'est encore mieux! Surtout si cela se fait de façon totalement automatisée.

D'ici là, si vous souhaitez, acquérir de nouvelles connaissances, voici une liste de ressources francophones particulièrement intéressantes :

- La formation "Introduction à la statistique avec R"(http://bit.ly/2XPyOXF) sur la plateforme FUN, dispensée par Bruno Falissard et Christophe Lalanne. Il s'agit d'une formation très complète, sur 5 semaines. Malheureusement, elle n'est ouverte qu'à certaines périodes de l'année.
- Le livre en ligne Contes et Stats R de Lise Vaudor. (http://bit.ly/2RmC8H5)
- Le livre en ligne Introduction à R et au tidyverse de Julien Barnier. (http://bit.ly/2XpIwTH)
- Le blog R-atique de Lise Vaudor. (http://perso.ens-lyon.fr/lise.vaudor/)
- Le site STHDA d'Alboukadel Kassambara. (http://www.sthda.com/french/wiki/wiki.php)
- La page Débutants- commencez ici de mon blog Statistique et logiciel R.(http://bit.ly/2Kmc8ew)

Vous pouvez retrouver une liste plus complète de ressources francophones ici : https://wp.me/p93iR1-m6