

# Guide d'installation ou de mise à jour de R et RStudio

Sophie Baillargeon, Université Laval

2021-01-13

## Table des matières

<b>1</b>	<b>Notes préliminaires</b>	<b>2</b>
1.1	Système d'exploitation . . . . .	2
1.2	Utilitaires pour automatiser les installations . . . . .	2
1.3	Nécessaire au développement de packages R . . . . .	2
1.4	LaTeX . . . . .	2
<b>2</b>	<b>Logiciel R</b>	<b>2</b>
2.1	Installation . . . . .	3
2.1.1	Détails pour Windows . . . . .	3
2.1.2	Détails pour Linux Ubuntu . . . . .	3
2.2	Mise à jour . . . . .	4
2.2.1	Détails pour Windows . . . . .	4
2.3	Être informé de la disponibilité d'une nouvelle version . . . . .	4
<b>3</b>	<b>RStudio</b>	<b>4</b>
3.1	Installation . . . . .	4
3.1.1	Détails pour Windows . . . . .	4
3.2	Configurations de RStudio . . . . .	4
3.3	Mise à jour . . . . .	5
<b>4</b>	<b>Packages R</b>	<b>5</b>
4.1	Installation . . . . .	5
4.1.1	À partir du CRAN . . . . .	5
4.1.2	À partir d'un dépôt informatique autre que le CRAN : . . . . .	5
4.1.3	À partir d'un fichier local . . . . .	5
4.1.4	Emplacement des packages installés . . . . .	6
4.2	Mise à jour . . . . .	6
4.3	Packages et mise à jour de R . . . . .	6
<b>5</b>	<b>Rtools (Windows seulement)</b>	<b>6</b>
5.1	Installation . . . . .	7
5.1.1	Ajout des Rtools au PATH . . . . .	7
<b>6</b>	<b>Xcode command line tools (macOS seulement)</b>	<b>7</b>
<b>7</b>	<b>Résolution de problèmes</b>	<b>7</b>

---

# 1 Notes préliminaires

## 1.1 Système d'exploitation

Cette fiche explique les étapes générales d'installation de R et RStudio pour tous les systèmes d'exploitation :

- Windows,
- macOS (représentant ici également ses prédécesseurs Mac OS X et OS X) ;
- Linux (ou Unix).

Quelques explications plus détaillées sont parfois fournies pour un système d'exploitation spécifique (en particulier Windows). La documentation officielle du *R core* concernant l'installation de la plus récente version de R est disponible ici : <https://cran.r-project.org/doc/manuals/r-release/R-admin.html>

## 1.2 Utilitaires pour automatiser les installations

Pour les utilisateurs de Windows, le [package `installr`](#) propose des fonctions pour installer ou mettre à jour R, RStudio ou les Rtools simplement en lançant une commande dans la console R. <https://CRAN.R-project.org/package=installr>

Pour les utilisateurs de macOS, le [package `updateR`](#) propose l'équivalent, mais limité à l'installation de R. <https://github.com/AndreaCirilloAC/updateR>

## 1.3 Nécessaire au développement de packages R

Afin de développer des packages R, quelques outils supplémentaires au logiciel R sont nécessaires, notamment un compilateur C/C++.

- Sous Windows, tous les outils requis sont réunis dans les [Rtools](#), qui sont gratuitement distribués sur internet. Ce document explique [comment installer les Rtools](#).
- Sous macOS, il faut avoir les « Xcode command line tools », disponibles gratuitement, s'ils ne sont pas déjà installés. Ce document explique [comment installer les Xcode command line tools](#).
- Sous Linux, il faut s'assurer d'avoir installé R accompagné des ses « development tools » (`r-base-dev`).

## 1.4 LaTeX

Pour compiler des documents R Markdown vers un format PDF, une installation de LaTeX est requise. De plus, une des commandes R pour le développement de packages (`R CMD check`) a besoin d'un compilateur LaTeX pour tester la création de la documentation en format PDF. Ainsi, plusieurs utilisateurs de R auront un jour également besoin de LaTeX.

Voici quelques suggestions de compilateurs LaTeX gratuits :

- MiKTeX : <https://miktex.org/download>
- MacTeX (macOS seulement) : <http://www.tug.org/mactex/>
- [TinyTeX](#), installable à partir de R grâce au package R `tinytex` (pensé spécifiquement pour les utilisateurs de R, peut-être moins intéressant pour ceux qui utilisent aussi LaTeX en dehors de R) : <https://CRAN.R-project.org/package=tinytex>

Notez que les systèmes Linux viennent habituellement par défaut avec une distribution de LaTeX.

---

# 2 Logiciel R

Si, lors de l'installation de RStudio, R est déjà présent sur l'ordinateur, son emplacement devrait être détecté automatiquement par RStudio. Il est donc préférable de débiter par l'installation de R.

## 2.1 Installation

Suivez d'abord ces étapes :

- allez sur le site <https://www.r-project.org/> ;
- cliquez sur « CRAN » dans le menu à gauche ;
- sélectionnez le site miroir du CRAN le plus près de chez vous ;
- dans la case intitulée « Download and Install R », cliquez sur le lien correspondant à votre système d'exploitation.

Sous Windows ou macOS, il suffit de télécharger un installateur et de double-cliquer sur le fichier pour démarrer l'installation. Sous Linux, les instructions varient en fonction de la distribution utilisée.

### 2.1.1 Détails pour Windows

Cliquez sur « base », puis sur « Download R x.y.z for Windows » où x.y.z est le numéro de la version la plus à jour de R. Le téléchargement du fichier « R-x.y.z-win.exe » débutera. Il s'agit d'un installateur, soit un programme permettant d'installer R. Il vous suffit ensuite de double-cliquer sur ce fichier et l'installation débutera. *Vous pouvez accepter toutes les options par défaut offertes par l'installateur.*

Pour télécharger l'installateur d'une version antérieure de R, il faut cliquer sur « Previous releases », puis sur la version désirée.

### 2.1.2 Détails pour Linux Ubuntu

Vous pouvez procéder à l'installation de R en suivant les instructions décrites sur cette page web :

- <http://basicgroundwork.blogspot.ca/2015/05/install-r-in-ubuntu-1404.html>

La première étape est d'ajouter un des miroirs du CRAN à la liste des dépôts informatiques depuis lesquels votre instance peut acquérir des logiciels. Cette liste se trouve dans le fichier `/etc/apt/sources.list`. Vous devez y ajouter le miroir `http://cran.rstudio.com/bin/linux/ubuntu` ou un autre miroir du CRAN. Pour effectuer cet ajout, le blogue cité plus haut suggère d'ouvrir le fichier avec `vi`, de l'éditer pour ajouter la ligne : `deb http://cran.rstudio.com/bin/linux/ubuntu trusty/`, puis d'enregistrer et fermer le fichier. Encore plus simplement, il est possible de réaliser cette édition en une seule commande soumise dans le terminal, la suivante :

```
printf 'deb http://cran.rstudio.com/bin/linux/ubuntu trusty/\n'| sudo tee -a \
/etc/apt/sources.list
```

Il faut ensuite mettre à jour la liste des fichiers disponibles dans les dépôts énumérés dans le fichier `/etc/apt/sources.list` avec la commande suivante :

```
sudo apt-get update
```

Si une erreur similaire à la suivante est générée :

```
W: GPG error: http://cran.rstudio.com/bin/linux/ubuntu trusty/ Release: The following
signatures couldn't be verified because the public key is not available: NO_PUBKEY
51716619E084DAB9
```

alors les commandes suivantes devraient régler le problème (en utilisant le même numéro de clé publique que dans l'erreur).

```
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-key 51716619E084DAB9
gpg -a --export 51716619E084DAB9| sudo apt-key add -
sudo apt-get update
```

Finalement, vous serez en mesure de procéder à l'installation de R et de ses « development tools » avec la commande suivante :

```
sudo apt-get install r-base r-base-dev
```

## 2.2 Mise à jour

Si une version antérieure de R est déjà installée sur votre ordinateur et que vous souhaitez la remplacer par une nouvelle version, *désinstaller l'ancienne version avant d'installer la nouvelle comporte quelques avantages*. Le principal avantage, sous Windows, est que les associations entre des extensions et R demeurent fonctionnelles. C'est le cas, par exemple, pour l'extension `.Rdata`. (L'extension `.R` n'est pas affectée si RStudio est installé, car elle est associée à RStudio plutôt qu'à R.) Aussi, les raccourcis sur le Bureau ou le menu démarrer ne sont pas perdus.

Cependant, sachez que plusieurs versions de R peuvent être installées simultanément sur votre ordinateur. Par défaut, chaque version de R est installée dans un sous-dossier portant le numéro de la version. Avoir plus d'une version de R est utile, par exemple, lorsqu'un de vos programmes R fonctionne seulement avec une version antérieure de R.

### 2.2.1 Détails pour Windows

Pour désinstaller un logiciel sous Windows, une façon simple de procéder est la suivante :

- allez dans les paramètres de Windows, puis dans « Applications » (ou l'équivalent selon votre version de Windows) ;
- sélectionnez « R for Windows u.v.w » où u.v.w est le numéro de la version de R à désinstaller ;
- cliquez sur « Désinstaller ».

Vous pouvez aussi aller dans les fichiers d'installation de R (par défaut "C:/Program Files/R/R-u.v.w") et double-cliquer sur le fichier `unins000.exe`.

La désinstallation démarrera.

## 2.3 Être informé de la disponibilité d'une nouvelle version

Il y a typiquement entre 4 et 6 mises à jour de R par année. Pour être informé lorsqu'une nouvelle version est disponible, vous pouvez vous abonner à la liste courriel R-announce (<https://stat.ethz.ch/mailman/listinfo/r-announce>). Les abonnés reçoivent un courriel lors de la mise en ligne d'une nouvelle version de R.

---

## 3 RStudio

### 3.1 Installation

Vous pouvez télécharger gratuitement la version la plus à jour de RStudio (RStudio Desktop Open Source License) à partir du site web : <https://www.rstudio.com/products/rstudio/download/>

Le fichier d'installation correspondant à votre système d'exploitation vous sera proposé. Vous pouvez aussi trouver le bon fichier dans le tableau intitulé « All Installers ».

#### 3.1.1 Détails pour Windows

Le fichier téléchargé est un installateur. Il suffit de double-cliquer sur ce fichier et de répondre aux quelques questions. Encore une fois, *vous pouvez accepter toutes les options par défaut offertes par l'installateur*.

### 3.2 Configurations de RStudio

Une fois RStudio installé, vous désirerez peut-être modifier quelques-unes de ses configurations. Celles-ci peuvent être changées par le menu « Tools > Global Options... ». Par exemple, il peut être intéressant de :

- dans « General », sélectionner « Never » pour l'option « Save workspace to .RData on exit » ;
- dans « Code > Saving », sélectionner « UTF-8 » comme « Default text encoding » ;

- dans « Code > Display », cocher « Show line numbers » ;
- autres selon vos goûts.

### 3.3 Mise à jour

Pour mettre à jour RStudio, il est inutile de désinstaller l'ancienne version comme il est conseillé de le faire pour R. Cette différence vient du fait que la nouvelle version de RStudio porte exactement le même nom que l'ancienne (il n'y a pas de numéro de version dans son nom) et elle est, par défaut, installée exactement dans le même emplacement que l'ancienne version. Les anciens fichiers sont donc remplacés par les nouveaux.

Le menu « Help > Check for Updates » permet de vérifier facilement si une nouvelle version de RStudio est disponible. Si c'est le cas, il suffit de l'installer de la même façon que lors d'une première installation.

Les configurations de RStudio ne sont pas perdues lors d'une mise à jour.

## 4 Packages R

### 4.1 Installation

#### 4.1.1 À partir du CRAN

Le [CRAN](#) est le dépôt informatique de packages R géré par le *R core team*. C'est là que la majorité des packages R sont rendus disponibles publiquement.

Pour installer un package à partir d'un des miroirs du CRAN, il suffit de :

- s'assurer d'être connecté à internet,
- démarrer une session R et
- lancer la commande suivante en R :

```
install.packages("nomPkg")
```

En RStudio, la commande `install.packages` peut être lancée par le bouton « Install » de la sous-fenêtre « Packages », en sélectionnant les options appropriées.

#### 4.1.2 À partir d'un dépôt informatique autre que le CRAN :

En plus du CRAN, des packages R sont partagés à d'autres endroits sur le web, notamment sur :

- le dépôt informatique de packages en bio-informatique [Bioconductor](#),
- un service web d'hébergement et de gestion de versions tel que [GitHub](#), [GitLab](#) ou [BitBucket](#).

Le [package remotes](#) offre des fonctions pour télécharger et installer directement à partir de ces sites (p. ex. fonctions `install_bioc`, `install_github`, `install_gitlab` et `install_bitbucket`).

#### 4.1.3 À partir d'un fichier local

Pour installer un package à partir d'un fichier local, il faut plutôt lancer la commande suivante en R :

```
install.packages("cheminAcces/nomPkg_NoVersion.zip", repos = NULL)
```

Dans cet exemple, l'extension du fichier doit être adaptée selon le système d'exploitation utilisé. L'extension `.zip` est celle d'un package pour Windows, `.tgz` est l'extension d'un package pour macOS et `.tar.gz` est l'extension d'un package source, qui est aussi la version requise avec Linux.

#### 4.1.4 Emplacement des packages installés

Le répertoire dans lequel les packages R sont installés par défaut est identifié dans le premier élément du vecteur retourné par la commande R suivante :

```
.libPaths()
```

Les autres éléments de ce vecteur sont d'autres répertoires dans lesquels R peut accéder à des installations de packages.

## 4.2 Mise à jour

Il ne faut pas oublier que les packages R sont eux aussi mis à jour périodiquement, tout comme R et RStudio. Pour avoir accès aux dernières corrections et aux derniers ajouts de fonctionnalités des packages que vous utilisez, il est conseillé de vérifier fréquemment si de nouvelles versions sont disponibles et de les installer si c'est le cas.

Une mise à jour d'un package R se fait avec la commande `install.packages`, exactement comme une installation. Il n'y a rien à désinstaller préalablement.

En RStudio, le bouton « Update » de la sous-fenêtre « Packages » permet de facilement vérifier si des mises à jour de packages sont disponibles.

## 4.3 Packages et mise à jour de R

Après une mise à jour de R, il est possible que vous n'ayez plus accès, avec la nouvelle version de R, aux packages que vous aviez installés avec une version antérieure de R. Cela dépend en fait du répertoire d'installation des packages.

S'ils sont installés directement dans le répertoire d'installation de R, il est certain que vous n'y aurez plus accès avec la nouvelle version de R. C'est que le nom du répertoire d'installation de packages dépend alors du numéro de version de R.

Afin de ne pas forcer l'utilisateur à mettre à jour les packages qu'il utilise à chaque fois qu'il met R à jour, le comportement par défaut de R est d'installer les packages dans un répertoire local à l'utilisateur, dont le nom dépend seulement des deux premiers numéros composant le numéro de version. Par exemple, avec mon installation de R sous Windows, les packages que j'installe se retrouvent par défaut dans le répertoire `"C:/Users/Sophie/Documents/R/win-library/u.v"`. Lors d'une mise à jour mineure de R, seul le dernier numéro composant le numéro de version change. Par exemple, le 10 octobre 2020, la version 4.0.3 de R est sortie, améliorant la version 4.0.2 sortie en juin 2020. Le répertoire d'installation de packages par défaut est donc demeuré le même entre les deux versions. Cependant, lorsqu'une mise à jour plus importante de R sortira, pour laquelle le deuxième numéro dans le numéro de version sera incrémenté, alors la mise à jour provoquera une perte d'accès aux packages installés précédemment, selon le comportement par défaut de R. Vous pourriez alors réagir en :

- réinstallant tous les packages dont vous avez besoin avec la nouvelle version de R, du coup les packages sont mis à jour ;
- ajouter l'ancien répertoire d'installation de packages dans le chemin de recherche de packages avec la commande suivante, en remplaçant la chaîne de caractères par le chemin d'accès approprié :

```
.libPaths(c(.libPaths(), "C:/Users/Sophie/Documents/R/win-library/u.v"))
```

---

## 5 Rtools (Windows seulement)

Comme mentionné dans les [notes préliminaires](#), le développement de packages R nécessite des outils non fournis dans l'installation de R. La version pour Windows de ces outils se trouve dans les [Rtools](#).

## 5.1 Installation

Vous pouvez vous rendre sur le site de téléchargement des Rtools comme suit :

- allez sur le site <http://www.r-project.org/> ;
- cliquez sur « CRAN » dans le menu à gauche ;
- sélectionnez le site miroir du CRAN le plus près de chez vous ;
- dans la case intitulée « Download and Install R », cliquez sur « Download R for Windows » ;
- cliquez sur « Rtools ».

Ensuite, téléchargez l'installateur approprié pour la version de R installée sur votre ordinateur. Si votre installation de R est à jour, il s'agit d'un des fichiers (64-bit ou 32-bit) directement disponible sur la page atteinte en suivant les instructions précédentes.

En cliquant sur le nom du fichier, le téléchargement de l'installateur débutera. Il vous suffira ensuite de double-cliquer sur le fichier téléchargé et l'installation débutera. Encore une fois, *vous pouvez accepter toutes les options par défaut offertes par l'installateur.*

### 5.1.1 Ajout des Rtools au PATH

Après l'installation des Rtools, il reste encore un détail technique à régler. Il faut ajouter les *make utilities* de Rtools au PATH de votre ordinateur. Le PATH est une variable d'environnement du système qui indique à l'ordinateur dans quels dossiers rechercher les programmes exécutables.

La page web à partir de laquelle vous avez téléchargé l'installateur des Rtools propose, à la section intitulée « Putting Rtools on the PATH », une façon d'effectuer cet ajout. Je vous invite à lire et suivre les indications fournies à cet endroit. Je mentionne seulement ici qu'en résumé, il suffit de lancer la commande suivante dans la console R puis de redémarrer R :

```
writeLines('PATH="%{RTOOLS40_HOME}\\usr\\bin;%{PATH}"', con = "~/.Renvirom")
```

## 6 Xcode command line tools (macOS seulement)

Comme mentionné dans les [notes préliminaires](#), le développement de packages R nécessite des outils non fournis dans l'installation de R. Sur macOS, les « Xcode command line tools » contiennent ces outils. Sur plusieurs versions récentes de macOS, les Xcode command line tools s'obtiennent gratuitement simplement en soumettant la commande suivante dans le Terminal (et en acceptant l'installation si une fenêtre s'ouvre pour vous le demander) :

```
xcode-select --install
```

Certaines versions de macOS pourraient exiger que vous vous inscriviez en tant que développeur Apple. Vous pouvez le faire gratuitement ici : <https://developer.apple.com/programs/register/>

Une autre façon d'obtenir les outils nécessaires au développement de packages R sur macOS est d'installer les « Xcode developer tools » complets, disponibles gratuitement sur le « App Store » : <https://apps.apple.com/ca/app/xcode/id497799835?mt=12>

Voici quelques références pour obtenir plus de détails :

- <https://osxdaily.com/2014/02/12/install-command-line-tools-mac-os-x/>
- <https://r-pkgs.org/setup.html#macos>

---

## 7 Résolution de problèmes

Voici comment régler certains problèmes techniques déjà rencontrés par des étudiants du cours.

### **Les accents s'affichent incorrectement dans RStudio.**

RStudio suppose un certain encodage lors de l'ouverture d'un fichier, par exemple d'un programme (aussi nommé script) R. Pour connaître l'encodage supposé par votre installation de RStudio, allez dans le menu « Tools > Global Options... > Code > Saving ». Le champ « Default text encoding : » sert à spécifier l'encodage par défaut. Les programmes R que vous téléchargerez sur le site web du cours seront toujours en encodage UTF-8.

Pour corriger l'affichage des accents dans un fichier en RStudio, vous pouvez :

- soit rouvrir le fichier avec le bon encodage par le menu « File > Reopen with Encoding... » (et sélectionner « UTF-8 » s'il s'agit d'un de mes fichiers),
- soit modifier la configuration de votre installation de RStudio pour utiliser par défaut l'encodage du fichier. Cette modification peut se faire par menu « File > Reopen with Encoding... » en sélectionnant l'encodage voulu, puis en cochant « Set as default encoding for source files », ou encore par le menu « Tools > Global Options... > Code > Saving », comme mentionné dans la [section concernant les modifications de configuration de RStudio](#).

### **L'explorateur de fichiers Windows n'affiche pas les extensions de certains fichiers.**

Ce problème n'est pas en lien avec l'installation de R ou RStudio, mais on me questionne souvent à ce propos. Par défaut, Windows n'affiche pas les extensions des fichiers dont le type a été associé à un programme. Voici une façon de changer cette option avec Windows 10.

Ouvrez la fenêtre d'« Options des dossiers » . Il est possible d'ouvrir cette fenêtre à partir d'un explorateur de fichiers, par le menu « Fichier > Modifier les options des dossiers et de recherche ». Ensuite, sous l'onglet Affichage, décochez l'option « Masquer les extensions des fichiers dont le type est connu ».