TP0 : Introduction au logiciel R

CIV4760 – Gestion de données en transport

Jean-Simon Bourdeau
Associé de recherche
jean-simon.bourdeau@polymtl.ca
B-344

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Plan de la présentation

- Introduction
- Types de données
- Structures de données
  - Les vecteurs
  - Les lists
  - Les factors
  - Les matrices
  - Les data frames
- Importation de fichiers
- Création de graphiques
- Fonctions statistiques

Introduction

- Ceci est une INTRODUCTION au logiciel R.
- Ressources pertinentes :
  - https://github.com/sahirbhatnagar/atelier-R-GERAD
  - Livre de Patrice Goulet : https://cran.r-project.org/doc/contrib/Goulet_introduction_programmation_R.pdf

Qu’est-ce que R ?

- R est un **logiciel libre** de traitement des données et d'analyse statistiques mettant en œuvre le langage de programmation S.
- C'est un projet fondé sur l'environnement développé dans les laboratoires Bell par John Chambers et ses collègues.
- Depuis plusieurs années, deux nouvelles versions apparaissent au printemps et à l'automne.
- Il dispose de nombreuses fonctions graphiques.

Source:

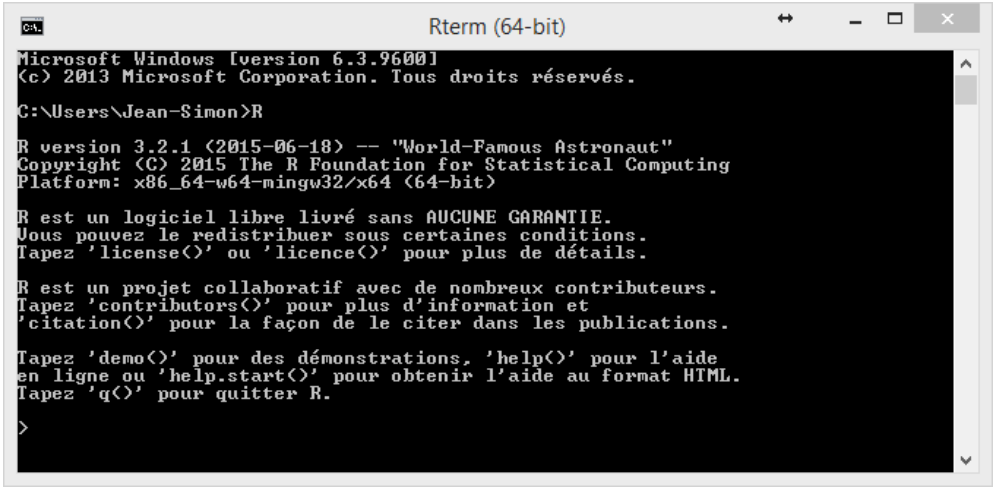
https://fr.wikipedia.org/wiki/R_(langage_de_programmation_et_environnement_statistique)

Pourquoi R

- http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages
- http://www.nature.com/news/programming-tools-adventures-with-r-1.16609

Environnement de travail

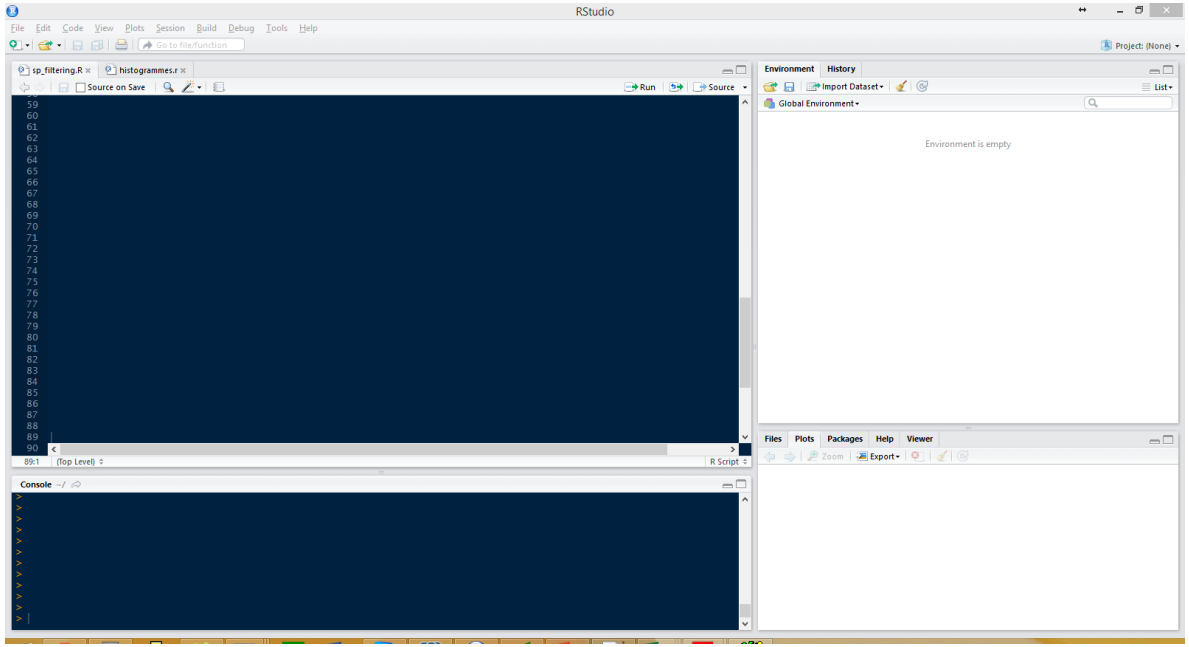
- Pour vérifier si R est installé :



- Pour installer R :
  - https://cran.r-project.org/ (Comprehensive R Archive Network)

Environnement de travail

- Nous allons travailler avec un logiciel :
  - R Studio
  - https://www.rstudio.com/
- Il s'agit d'un environnement de développement intégré (IDE).



Gestion du répertoire de travail

- Pour obtenir l'environnement de travail actuel :

```
> getwd()
[1] "C:/Python27/Lib/site-packages/rpy2"
```

- Pour le modifier :

```
> setwd("C:/Users/Jean-Simon/Dropbox")
```

Quelques notions

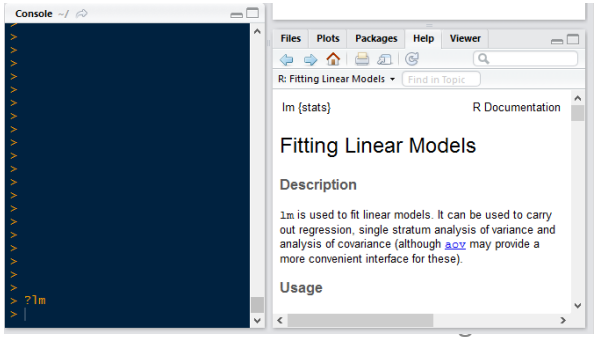
- Pour commenter :
  - Utilisation du # :

```
> #ceci ne fera rien
>
```

- Pour référencer une variable : « <- » :

```
> valeur
Error: object 'valeur' not found
>
>
> valeur <- 5
>
>
> valeur
[1] 5
```

- Pour obtenir de l'aide sur une fonction : ? + nom de la fonction



Opérations de base

- Addition

  5+5
- Soustraction

  5 - 5
- Multiplication

  3 * 5
- Division

  (5 + 5)/2
- Exposant

  2^5
- Modulo

  28%%6

Types de données

- Numeric (4.5)
- Integers (4)
- Logical (T,F,TRUE,FALSE) (Attention : R est sensible à la casse)
- Character ("quatre")

- La fonction class() permet de déterminer le type de données

Types de données

- Une dimension :
  - *Vector*
  - List
  - Factor
- Deux dimensions :
  - *Matrix*
  - Data Frame

Les vecteurs

- Pour créer un vecteur, il suffit d'utiliser la fonction concaténation. Ex :
  - c(1,2,3)
- Types de vecteurs les plus fréquents :
  - Double
  - Integer
  - Character
  - Logical
- Si on fait la concaténation de plusieurs types de données, les données seront converties au type le plus flexible
- On peut assigner des noms à un objet à l'aide de la fonction names()

Les vecteurs (suite)

- Il est possible d'additionner des vecteurs, tant que ceux- ci sont de mêmes dimensions

```
> a <- c(1,2,3)
> b <- c(4,5,6)
>
> a+b
[1] 5 7 9
>
> c <- c(7,8)
>
> b + c
[1] 11 13 13
Warning message:
In b + c : longer object length is not a multiple of shorter object length
```

Les vecteurs (suite)

- Pour accéder à un (ou plusieurs) élément du vecteur : []

```
> a <- c(1,2,3)
> a
[1] 1 2 3
> names(a) <- c("un","deux","trois")
> a
  un  deux trois
   1     2     3
> a[0]
named numeric(0)
> a[1]
un
 1
> a["un"]
un
 1
```

```
> a[1:3]
[1] 1 2 3
```

Les factors

- Qu'est-ce qu'un factor ?
  - Variable catégorielle
  - Exemple : le genre d'une personne (Homme/Femme)
- Création d'un factor : factor()
- On peut aussi ordonner les catégories :

```
> temperature <- c("basse","élevée","moyenne")
>
> factor(temperature, order=TRUE, levels = c("basse","moyenne","élevée"))
[1] basse   élevée  moyenne
Levels: basse < moyenne < élevée
```

- Le concept de niveau permet d'associer une valeur à chaque facteur
  - Levels(factor) <- value
- On peut aussi obtenir des statistiques sur les facteurs avec summary()

Les matrices

- On utilise la fonction matrix() :

```
> matrix(1:9, byrow=TRUE, nrow=3)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

Les matrices (suite)

- Une matrice peut être construite avec des vectors, qui peuvent devenir des lignes ou des colonnes :

```
> b <- c(3,4)
> a <- c(8,9)
>
> matrix(c(a,b), nrow=2, byrow=TRUE)
     [,1] [,2]
[1,]    8    9
[2,]    3    4
```

- Comme avec les vecteurs, il est possible de donner des noms aux lignes – rownames() , et aux colonnes – colnames()

Les matrices (suite)

- Fonctions d'addition :
  - Somme de lignes : rowSums()
  - Somme de colonnes : colSums()
- Concaténation des vecteurs/matrices :
  - Par colonnes : cbind()
  - Par lignes : rbind()
- Pour accéder aux éléments d'une matrice :
  - matrice[ligne(s),colonne(s)]

- Opérations sur les matrices :
  - Additions/soustractions
  - Multiplications/divisions

Les Data Frames

- Définition :
  - Liste de variables ayant chacune le même nombre de lignes
- Data Frame existant dans R : mtcars
- Fonctions de partitionnement :
  - head()
  - tail()
- Fonctions pour résumer les données
  - str()
- Pour créer un data frame : data.frame()
- Pour sélectionner des éléments : même chose qu'avec les matrices, sauf que pour sélectionner une variable, on utilise le symbole $ (ex : dataframe$variable)

Les Data Frames

- Exemple de sélection :

```
> mtcars[mtcars$mpg>30,]
                mpg cyl disp  hp drat    wt  qsec vs am gear carb
Fiat 128       32.4   4 78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic    30.4   4 75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla 33.9   4 71.1  65 4.22 1.835 19.90  1  1    4    1
Lotus Europa   30.4   4 95.1 113 3.77 1.513 16.90  1  1    5    2
```

- Ou on peut utiliser la fonction subset()

Les Lists

- Les lists sont comme des vecteurs, sauf qu'ils peuvent contenir plusieurs types de données (en fait, on peut tout insérer dans une list).
- Il suffit d'utiliser la fonction list() :

```
> list(1,2,3)
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3
```

Les lists

- Sélection d'éléments

```
> l <- list(c(1,2,3),c(4,5,6))
>
> l[[1]]
[1] 1 2 3
> l[[2]][2]
[1] 5
```

Importation de fichiers

- Comment faire pour importer un fichier csv ?
  - read.table()

- Exercice :
  - Utilisez la fonction read.table() pour créer un data frame avec les données d'Environnement Canada (https://meteo.gc.ca/canada_f.html)
  - Attention aux arguments de la fonction !

Productions de graphiques

- Fonctions
  - plot()
  - boxplot()
  - hist()
- Pour enregistrer les graphiques :
  - pdf()
  - png()

Fonctions statistiques

- Toujours de la forme y ~ model
- Ex : lm - régression linéaire

Autres librairies (« packages ») intéressantes

- Création dynamique de documents :
  - http://rmarkdown.rstudio.com/
  - http://shiny.rstudio.com/gallery/
- Production avancée de graphiques :
  - http://ggplot2.org/

- Pour installer un package : install.packages("nom_du_package")
  - Après il faut l'importer : library(nom_du_package)