

# Tidyverse package

Tarik H.

12/5/2020

## dplyr Package

Le package dplyr est l'un des packages les plus puissants et les plus populaires de R. Ce package sert à nettoyer et transformer les données. Cela rend l'exploration et la manipulation de données faciles et rapides dans R. Le package "dplyr" comprend de nombreuses fonctions qui effectuent des opérations de manipulation de données telles que l'application de filtre, la sélection de colonnes spécifiques, le tri des données, l'ajout ou la suppression de colonnes et l'agrégation de données.

## dplyr Practical Examples

### Example 1 : Selecting Random N Rows

```
sample_n(mydata, 3)
```

### Example 2 : Remove Duplicate Rows based on all the variables

La fonction distinct est utilisée pour éliminer les doublons.

```
x1 = distinct(mydata)
```

### Example 3 : Selecting Variables

Supposons que l'on nous demande de ne sélectionner que quelques variables. Le code ci-dessous sélectionne les variables "Index", les colonnes de "State" à "Y2008".

```
mydata2 = select(mydata, Index, State:Y2008)
```

### Example 4 : Dropping Variables

Le signe mathématique "moins" avant une variable indique à R de "droper" la variable.

```
mydata = select(mydata, -Index, -State)
```

### Example 5 : Filter Rows

Supposons que nous avons besoin des sous-ensembles de données. Nous voulons filtrer les lignes et ne retenir que les valeurs pour lesquelles l'Index est égal à A.

```
mydata7 = filter(mydata, Index == "A")
```

## readr Package

L'objectif de "readr" est de fournir un moyen rapide et convivial de lire des "rectangular data", des données transversales à plusieurs variables dans lesquelles chaque colonne est une variable et chaque ligne est un cas ou un enregistrement (fichiers "csv", "tsv" et "fwf"). Il est conçu pour analyser de manière flexible de nombreux types de données trouvées dans la nature, tout en échouant proprement lorsque les données changent de manière inattendue.

## Example :

```
library(readr)
teams <- read_csv("data/team_standings.csv")
```

# ggplot Package

Le package ggplot est un package pour la data visualization. R propose différentes méthodes grâce à des graphiques intégrés et des logiciels puissants tels que ggplot2. La première version aide à la création de graphiques standards tandis que la seconde aide à la création de graphiques professionnels / personnalisés.

## Example :

```
ggplot(data = mpg, aes(x=class)) + geom_bar()
```

“data = mpg” Data : ensemble de données mpg : dataframe

“aes(x=class)” Aesthetics : indique les variables x et y. Elle est également utilisée pour indiquer à R comment les données sont affichées dans un tracé, par exemple la couleur, la taille et la forme des points, etc

class : x-axis

“geom\_bar()” Geometry : se réfère au type de graphique (diagramme à barres, histogramme, box plot, line plot, density plot, dot plot, etc.)

geom\_bar : bar-chart

## Important plots & important functions :

### Scatter plot :

geom\_point(), geom\_smooth(), stat\_smooth()

### Bar chart :

geom\_bar(), geom\_errorbar()

### Histogram :

geom\_histogram(), stat\_bin(), position\_identity(), position\_stack(), position\_dodge()

### Box plot :

geom\_boxplot(), stat\_boxplot(), stat\_summary()

### Line plot :

geom\_line(), geom\_step(), geom\_path(), geom\_errorbar

### Pie chart :

coord\_polar()

## Example 1 : Creating Histogram

Firstly we consider the ‘iris data’ to create histogram and scatter plot. Considering the iris data. Creating a histogram

```
ggplot(data = iris, aes( x = Sepal.Length)) + geom_histogram( )
```

## Example 2 : Various groups in density plot

```
ggplot(iris, aes(x=Sepal.Length, color=Species)) + geom_density( )
```

# caret Package

En R, le package caret signifie Classification And REgression Training. Il facilite la modélisation prédictive. Il peut exécuter la plupart des techniques de modélisation prédictive avec une cross validation. Il peut également effectuer des étapes de modélisation de données de découpage et de prétraitement des données.

```
library(caret)  
lm1 <- train(annual_pm~., data = air, method = "lm")  
rf1 <- train(annual_pm~., data = air, method = "rf")
```