

# Introduction à



*Mai, juin 2024*

# PROGRAMME

## 1. Séance n°1 :

- **Introduction : la programmation, R et RStudio**, *comprendre les principes de la programmation, et les spécialités du langage R et de l'interface RStudio ;*
- **Importer des données ;**

## 2. Séance n°2 :

- **Nettoyer et transformer des données : le tidyverse**, *connaître les principales fonctions pour nettoyer et manipuler les données, avec des exercices de mise en pratique ;*

## 3. Séance n°3 :

- **Visualiser des données : ggplot2**

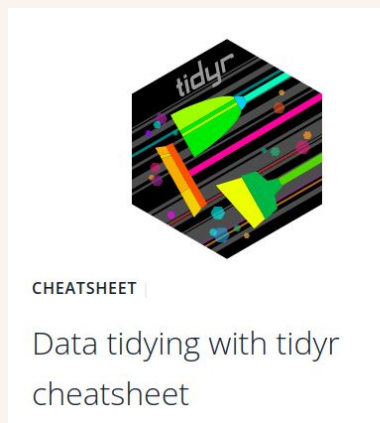
# Séance n°1

# La programmation, R et RStudio

```
    }).done(function(response) {  
      for (var i = 0; i < response.length; i++) {  
        var layer = L.marker(  
          [response[i].latitude, response[i].longitude]  
          // , {icon: myIcon}  
        );  
        layer.addTo(group);  
  
        layer.bindPopup(  
          "<p>" + "Species: " + response[i].species + "<br>" +  
          "<p>" + "Description: " + response[i].description + "<br>" +  
          "<p>" + "Seen at: " + response[i].latitude + " " + response[i].longitude + "<br>" +  
          "<p>" + "On: " + response[i].sighted_at + "</p>"  
        );  
      }  
  
      $('select').change(function() {  
        species = this.value;  
      });  
    });  
  }  
  $.ajax({  
    url: queryURL,  
    method: "GET"  
  }).done(function(response) {  
    for (var i = 0; i < response.length; i++) {  
      var layer = L.marker(  
        [response[i].latitude, response[i].longitude]  
        // , {icon: myIcon}  
      );  
      layer.addTo(group);  
    }  
  });  
}
```

# Avant de commencer :

## ❖ les fiches tuto de POSIT



<https://posit.co/resources/cheatsheets/>

## ❖ un support complet pour approfondir les notions



[https://dianethy.github.io/cours\\_R/Introduction\\_R.html](https://dianethy.github.io/cours_R/Introduction_R.html)

# La programmation, tout un concept :

- **Définition** : ensemble des activités qui permettent l'écriture de programmes informatiques.
- **Synonymes** : codage, développement.
- **Contexte** : révolution informatique, 2.5 trillions d'octets de données créés chaque jour.
- **Utilité** : obtenir des informations et connaissances à partir de données.
- **Usage** : de nombreux langages existent (HTML, Javascript, Java, C, Python, R...)

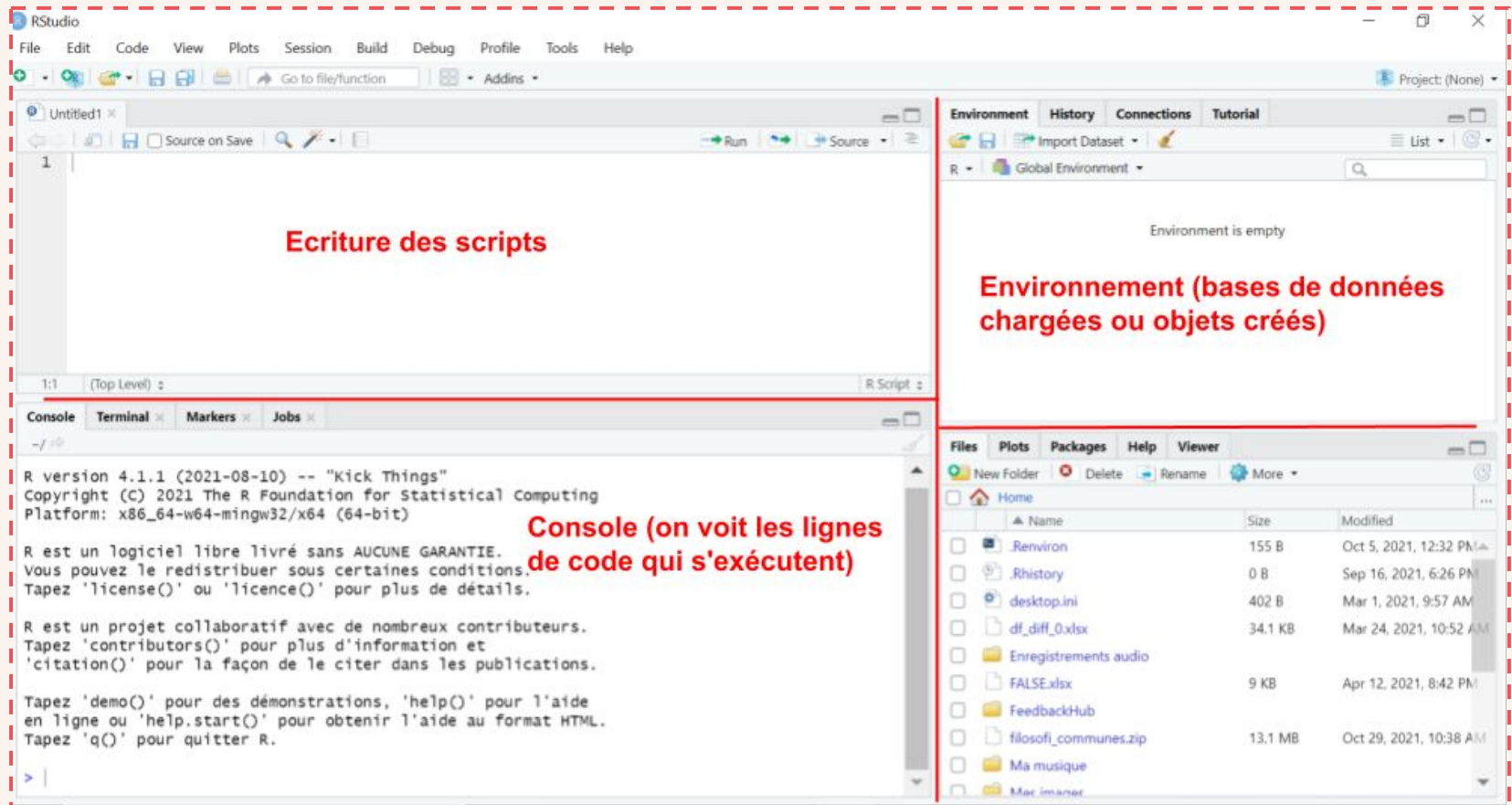
# R, un langage de programmation **destiné aux statistiques**

**If statistics programs/languages were cars...**



Gratuit, puissant, design, avec une communauté importante ([Stack Overflow](#), [R-bloggers](#) etc.)

# RStudio, une interface permettant d'interagir avec R





# Les concepts clés :


漢

- run
- package
- library
- script

Abc

- **exécuter** (une ligne de code)
- **ensemble de fonctions**
- **endroit où sont gardés les packages**
- **document où l'on code** (extension *.R*)

# Assez parlé, ouvrons RStudio !

- **Créer un nouveau script** via *File > New File > R script*
- **Commencer à interagir par de simples opérations arithmétiques:**  $2+3$
- **Assigner des valeurs à des objets:**
  - $x \leftarrow 2$  (ou  $x = 2$  mais la bonne pratique est d'utiliser ' $\leftarrow$ ')
  - $y \leftarrow 7$
  - $resultat \leftarrow x * y$
  - $resultat$
- **Exécuter les lignes de code:** *Ctrl/Enter* ou bouton 
- **Commenter le code** via le #

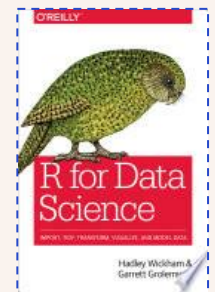
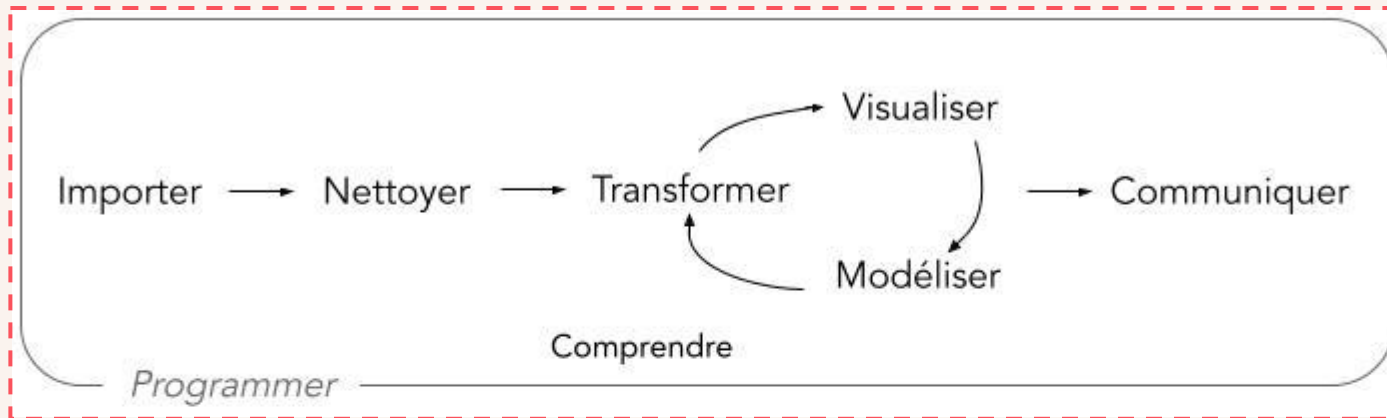
# Le tidyverse



## Installation des packages

- **Stockage des packages R sur le CRAN**
  - `install.packages("readxl")`
  - `install.packages("tidyverse")`

## Schéma d'exploitation des données



source

# Deux manières de coder en R

- **R-base**

- exemple 1 : `exemple[,2]`
- exemple 2 : `f(g(h(exemple)))`

- **Tidyverse**

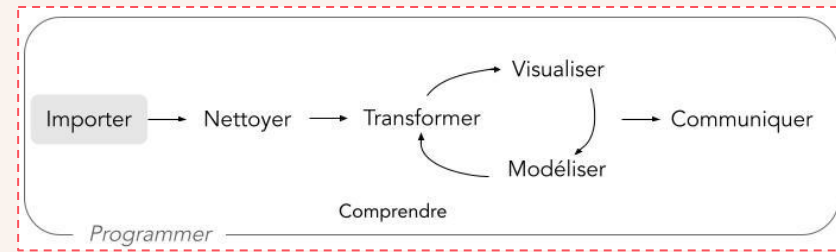
- exemple 1 : `exemple |> select(2)`
- exemple 2 : `exemple |> h() |> g() |> f()`



Le tidyverse fonctionne grâce au pipe (`|>`), qui permet d'appliquer des fonctions à une base de données. L'opération à droite du pipe est appliquée à la valeur située à gauche du pipe, il devient alors possible d'enchaîner les traitements.

# Codons ensemble avec les principales fonctions du tidyverse

# Importer des données



- **Excel**
  - `library(readxl)`
  - `exemple <- read_excel(path = "fichier.xlsx", sheet = "nom_feuille")`
- **CSV : valeurs séparées par des virgules**
  - `library(readr)`
  - `exemple <- read_csv("fichier.csv")`
- **SCSV : valeurs séparées par des points-virgules (“faux” CSV)**
  - `library(readr)`
  - `exemple <- read_delim("fichier.csv", delim = ";")`
- **À vous !** Importer le jeu de données ‘*Prénoms donnés par département - Millésimé - France*’ disponible sur OpenDataSoft :
  - au format CSV en utilisant le lien [**seulement en 1900 !!**]
  - si vous avez fini, au format tableur Excel en exportant le fichier [**seulement en 1900 !!**]

# Importer des données

## Solution

- Importer le jeu de données '**Prénoms donnés par département - Millésimé - France**' disponible sur OpenDataSoft :
  - en CSV en utilisant le lien

```
library(tidyverse)
data <-
read_delim("https://public.opendatasoft.com/api/explore/v2.1/catalog/datasets/demographyref-france-prenoms-departement-millesime/exports/csv?lang=fr&refine=birth_year%3A%221900%22&facet=facet(name%3D%22birth_year%22%2C%20disjunctive%3Dtrue)&timezone=Europe%2FBerlin&use_labels=true&delimiter=%3B",
, delim = ";")
```

- en excel en exportant le fichier

```
library(readxl)
data <-
read_excel("~/Downloads/demographyref-france-prenoms-departement-millesime.xlsx")
```