

Visualisation et analyse des données de surveillance avec le logiciel R

Yann Le Strat

Département des Maladies Infectieuses
Institut de Veille Sanitaire. France

1er février 2016

Introduction

Le but de cette présentation est de présenter certains packages R permettant la visualisation et l'analyse des données de surveillance.

R en quelques mots

- ▶ Création en 1993, par Ihaka et Gentleman, professeurs de statistiques à l'université d'Auckland
- ▶ Logiciel libre
- ▶ Repose sur des packages, développés par des utilisateurs et mis à la disposition de tous.
- ▶ Package : ensemble de fonctions, accompagnées souvent d'un ou plusieurs jeux de données, couvrant un domaine particulier.

Installation de R

Installation de R, à partir du site *The Comprehensive R Archive Network (CRAN)* : <https://cran.r-project.org/>



[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

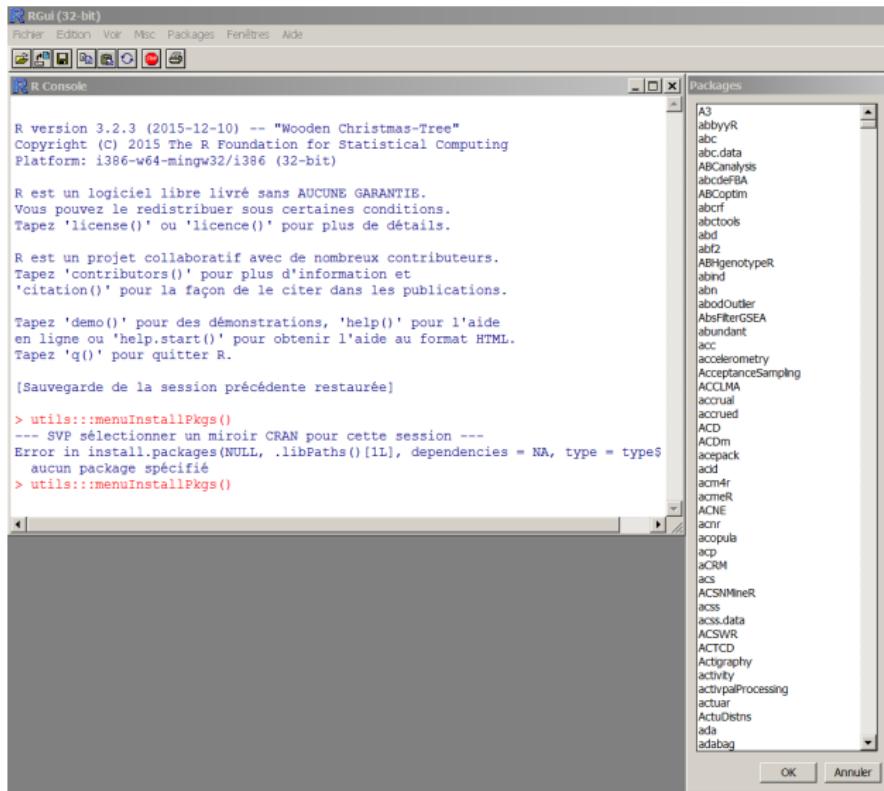
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-08-14, Fire Safety) [R-3.2.2.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.

Installation de librairies



Quelques éditeurs pour R

- ▶ Utilisation d'un éditeur pour écrire un programme (suite de commandes), même si on peut utiliser n'importe quoi (Bloc-notes par exemple)
- ▶ Certains éditeurs offrent plus de fonctionnalités :
 - ▶ Tinn-R :
<http://nbcgib.uesc.br/lec/software/editores/tinn-r/en>
 - ▶ RStudio
<https://www.rstudio.com/products/rstudio/download/>

Environnement de Tinn-R

The screenshot shows the Tinn-R IDE interface. On the left is a file browser window titled "Tools" with tabs for Misc, Markup, Results, Spell, Database, and R. The R tab is selected, showing a list of mirrors. The main area contains an R script and an R console.

R Script Content:

```
<<setup, include=FALSE, cache>>
library(knitr)
# set global chunk options
opts_chunk$set(fig.path='figs',
               fig.align='center',
               fig.show='hold')

options(replace.assign=TRUE,
        width=90)
@

\title{A Minimal Demo of knitr}
\author{Tinn-R Team (based on RStudio)}
\maketitle

To generate the HTML output
The associated shortcut is:

You can test if \textbf{kni}!
get started with some boring

<<boring-random>>=
set.seed(1121)
(x=rnorm(18))
mean(x)
var(x)
@
```

R Console Output:

```
> (x=rnorm(18))
[1] 0.1449583 0.4383221 0.1531912
[4] 1.0849426 1.9995449 -0.8118832
[7] 0.1602680 0.5858923 0.3600880
[10] -0.0253084 0.1508809 0.1100824
[13] 1.3596812 -0.3269946 -0.7163819
[16] 1.8097690 0.5084011 -0.5274603

> mean(x)
[1] 0.3587774

> var(x)
[1] 0.6217067

> sd
function (x, na.rm = FALSE)
sqrt(var(if (is.vector(x)) x else as.double(x)))
<bytecode: 0x00000000208b9730>
<environment: namespace:stats>

mean
function (x, ...)
<bytecode: 0x0000000014342030>
<environment: namespace:base>
```

Bottom Status Bar:

ANSI WIN 61/87: 10 Editing Normal R hotkeys active R -> TCP/IP [stats] <rnorm>

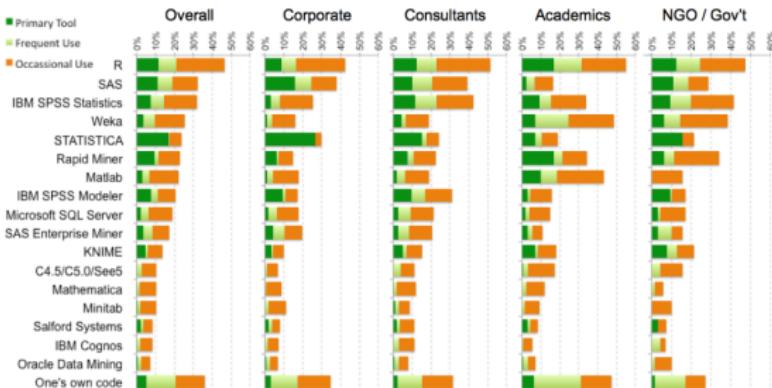
Environnement de RStudio

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Displays a script named "08_syndrome.R" containing R code for calculating epidemic periods and creating a line plot.
- Environment View:** Shows the global environment with objects like `d`, `data`, `data.week`, `values`, `bigrL`, and `l`.
- Plot View:** A line plot titled "Proportion de syndromes grippaux" showing the proportion of influenza-like illnesses from July 2010 to July 2015. The plot includes three lines: a blue line for "baseline", a green line for "serf95", and a red line for "proportion". Shaded yellow areas indicate epidemic periods.
- Console:** Displays the R code used to generate the plot, including the creation of a dygraph object and the addition of shaded regions for periods where the proportion exceeds the baseline by more than 5%.

Popularité

Une enquête menée par Rexter Analytics auprès de 1 300 analystes montre que R est le logiciel le plus souvent utilisé lorsqu'il s'agit d'un travail en entreprise, dans le monde académique, au sein d'organismes publics ou d'ONG et chez les analystes travaillant comme consultants.



Popularité

AUSTRIAN JOURNAL OF STATISTICS
Volume 41 (2012), Number 1, 59–66

Are There Too Many R Packages?

Kurt Hornik

Wirtschaftsuniversität Wien

Abstract: The number of R extension packages available from the CRAN repository has tremendously grown over the past 10 years. We look at this phenomenon in more detail, and discuss some of its consequences. In particular, we argue that the statistical computing community needs a more common understanding of software quality, and better domain-specific semantic resources.

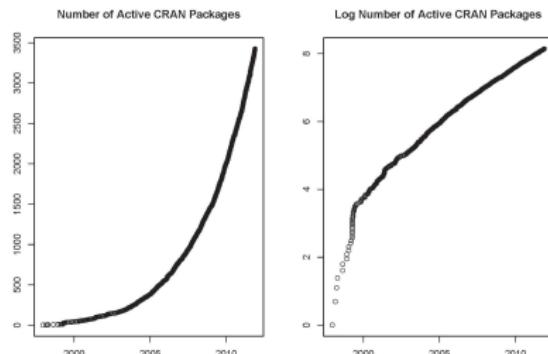


Figure 1: Development of the estimated number of active CRAN packages on a raw (left) and log (right) scale.

Visualisation des données

Un exemple marquant et pionnier

En 2006, présentation de Hans Rosling (médecin/statisticien à l'institut Karolinska en Suède) à la conférence TED :

<https://www.youtube.com/watch?v=usdJgEwMinM>

Graphiques interactifs pour le Web

Création de graphiques interactifs pour les insérer dans des pages web. Accessible maintenant pour ordinateurs et appareils mobiles.

Quelques exemples :

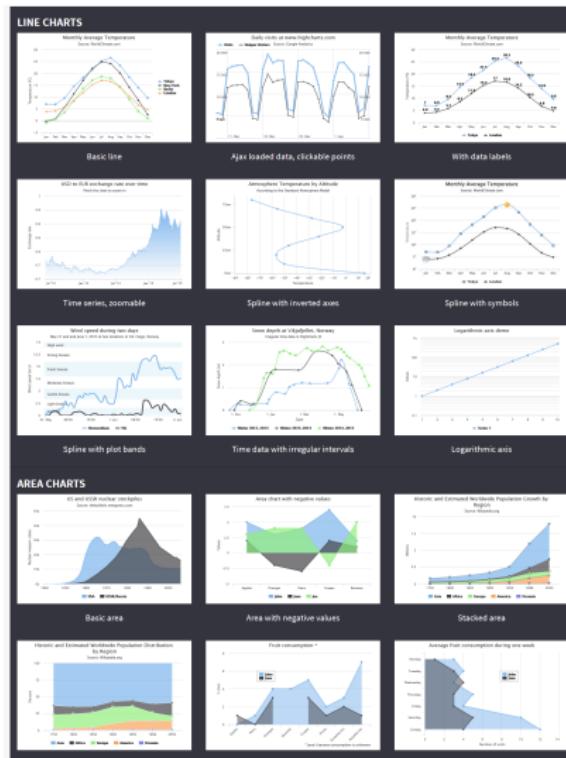
- ▶ Google
- ▶ Highcharts
- ▶ Chart.js
- ▶ ZingChart

Google charts



<https://developers.google.com/chart/interactive/docs/gallery>

Highcharts



<http://www.highcharts.com/demo>

Et R dans tout ca ?

- ▶ Jusqu'en 2014, ces outils web interactifs et R n'étaient pas liés (ou très peu).
- ▶ D'un côté on avait un logiciel statistique hyper-puissant (nécessitant des compétences statistiques) mais sans interface.
- ▶ De l'autre côté on avait des graphiques interactifs magnifiques (nécessitant une connaissance en langage du web (javascript, ...) mais rien derrière.
- ▶ Et l'épidémiologue ne profitait ni de l'un ni de l'autre.
- ▶ On doit le rapprochement de R et du web à l'entreprise RStudio.

RStudio

- ▶ Entreprise privée américaine, fondée par JJ Allaire
- ▶ 25 personnes (Boston, Seattle, ...)
- ▶ Commercialisent une version payante de RStudio (l'éditeur), du package Shiny, et de l'hébergement d'applications web sur leur serveur.
- ▶ Depuis peu, RStudio crée un "écosystème" de packages dans une logique d'intégration dans des applications web.



RStudio - packages (1)



R Markdown

R Markdown lets you insert R code into a markdown document. R then generates a final document that replaces the R code with its results.

[Project Site Link >](#)



Shiny

Shiny makes it incredibly easy to build interactive web applications with R. Automatic "reactive" binding between inputs and outputs and extensive pre-built widgets.

[Project CRAN Site Link >](#)



packrat

A dependency management tool for R to make your R projects more isolated, portable, and reproducible.

[Project GitHub Link >](#)



tidyr

tidyR is new package that makes it easy to "tidy" your data. Tidy data is data that's easy to work with: it's easy to munge (with dplyr), visualise (with ggplot2 or ggviz) and model (with R's hundreds of modelling packages).

[Project Paper Link >](#)



Leaflet

Leaflet is one of the most popular open-source JavaScript libraries for interactive maps. This R package makes it easy to integrate and control Leaflet maps in R.

[Project Site Link >](#)



DT

The R package DT provides an R interface to the JavaScript library DataTables. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and DataTables provides filtering, pagination, sorting, and many other features in the tables.

[Project GitHub Link >](#)



ggplot2

An enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

[Project Site Link >](#)



knitr

Elegant, flexible and fast dynamic report generation that combines R with TeX, Markdown, or HTML.

[Project Site Link >](#)



dplyr

dplyr is the next iteration of plyr, focussing on only data frames. dplyr is faster and has a more consistent API.

[Project GitHub Link >](#)

RStudio - packages (2)



Haven

Haven allows you to load foreign data formats (SAS, Spss and Stata) in to R by wrapping the fantastic ReadStat library.

[GitHub Site Link >](#)



devtools

devtools removes the pains and bottlenecks of package development. The aim of devtools is to make your life as a package developer easier by providing R functions that simplify many common tasks.

[Project Site Link >](#)



readxl

The readxl package makes it easy to get data out of Excel and into R. Compared to many of the existing packages (e.g. gdata, xlst, xlsReadWrite) readxl has no external dependencies so it's easy to install and use on all operating systems. It is designed to work with tabular data stored in a single sheet.

[Project GitHub Link >](#)



ggvis

ggvis is the next iteration of the popular ggplot2 graphics package. ggvis creates dynamic, interactive data visualizations.

[Project Site Link >](#)



testthat

Testing your code is normally painful and boring. testthat tries to make testing as fun as possible, so that you get a visceral satisfaction from writing tests. Testing should be fun, not a drag, so do it all the time.

[Project GitHub Link >](#)



htmlwidgets

html widgets brings the best of JavaScript data visualization to R. You can use JavaScript visualization libraries at the R console, just like plots, embed widgets in R Markdown documents and Shiny web applications, and develop new widgets using a framework that seamlessly bridges R and JavaScript.

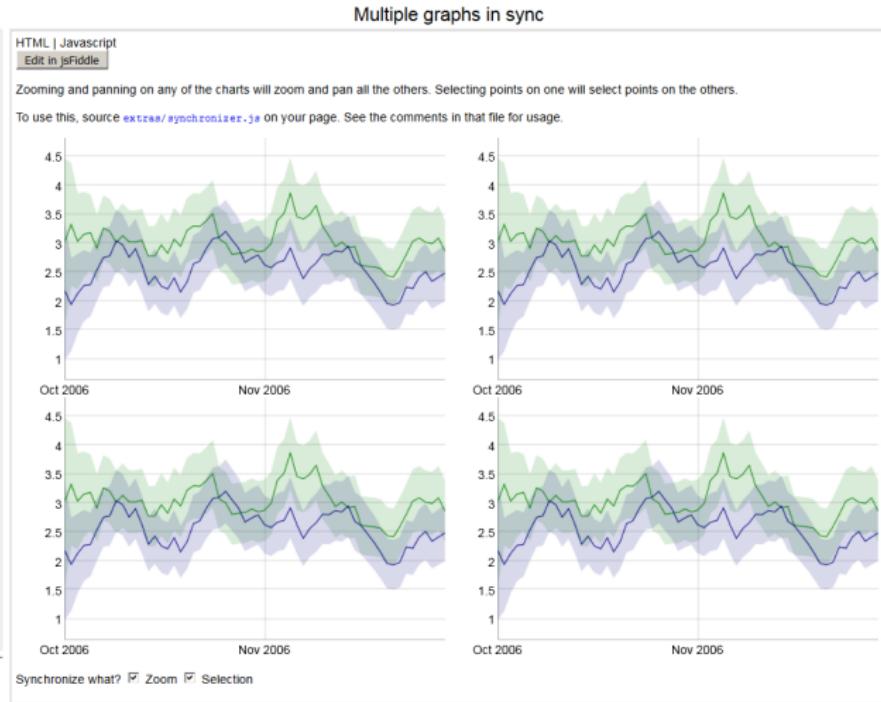
[Project Website Link >](#)

Package *dygraphs*

dygraphs (librairie JavaScript)

Dygraphs Gallery

- Annotations
- Time Series Drawing Demo
- Dynamic Update
- Highlight Closest Series
- Highlighted Region
- Highlighted Weekends
- Independent Series
- Function Plotter
- Link Interaction
- Per-series properties
- Synchronization**
- Range Selector
- Resizable Graph
- Stock Chart Demo
- CSS label styling
- Roll Periods and Error Bars
- Custom interaction models
- Linear Regressions
- Edge Padding



<http://dygraphs.com/gallery/>

dygraphs (librairie R)

dygraphs for R

Home

USING

R Console

R Markdown

Shiny

GALLERY

Series Options

Series Highlighting

Axis Options

Second Y Axis

Plot Labels

Plot Legend

Time Zones

CSS Styling

Range Selector

Synchronization

Roll Periods

Shaded Regions

Event Lines

Annotations

Upper/Lower Bars

dygraphs for R

The dygraphs package is an R interface to the [dygraphs](#) JavaScript charting library. It provides rich facilities for charting time-series data in R, including:

- Automatically plots `xts` time series objects (or any object convertible to `xts`).
- Highly configurable axis and series display (including optional second Y-axis).
- Rich interactive features including `zoom/pan` and series/point `highlighting`.
- Display `upper/lower bars` (e.g. prediction intervals) around series.
- Various graph overlays including `shaded regions`, `event lines`, and `point annotations`.
- Use at the R console just like conventional R plots (via RStudio Viewer).
- Seamless embedding within `R Markdown` documents and `Shiny` web applications.

Installation

You can install the `dygraphs` package from CRAN as follows:

```
install.packages("dygraphs")
```

You can use dygraphs at the R console, within R Markdown documents, and within Shiny applications. See the usage documentation linked to from the sidebar for more details. There are a few demos of dygraphs below as well as quite a few others in the gallery of examples.

Demos

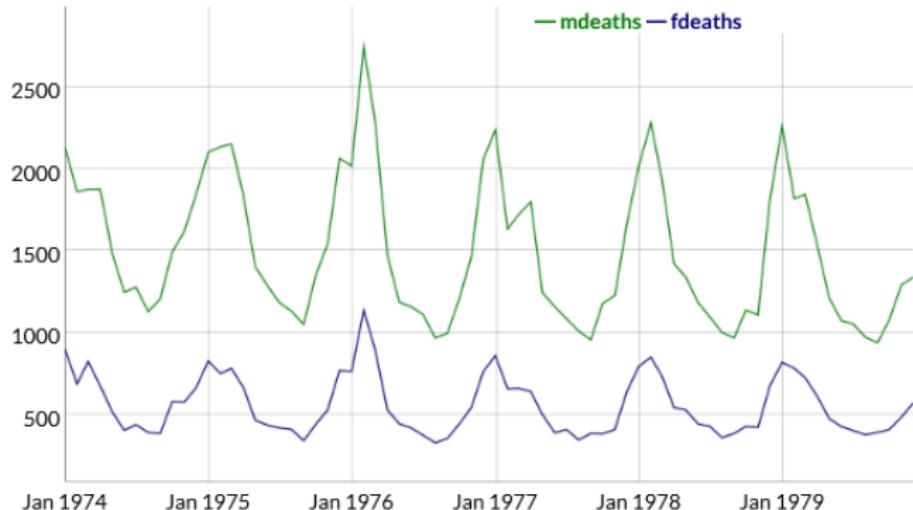
Here's a simple dygraph created from a multiple time series object:

```
library(dygraphs)
lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths)
```

<https://rstudio.github.io/dygraphs/>

dygraphs - exemple

```
library(dygraphs)
lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths)
```



TP

Package *Leaflet*

Leaflet (librairie JavaScript)

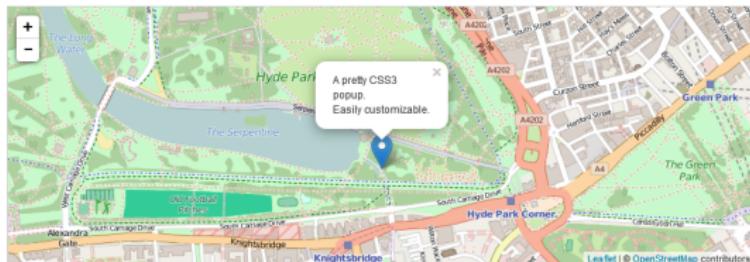


an open-source JavaScript library
for mobile-friendly interactive maps

[Overview](#) [Tutorials](#) [Docs](#) [Download](#) [Plugins](#) [Blog](#)

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about [33 KB of JS](#), it has all the mapping [features](#) most developers ever need.

Leaflet is designed with *simplicity, performance and usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of [plugins](#), has a beautiful, easy to use and [well-documented API](#) and a simple, readable [source code](#) that is a joy to [contribute](#) to.



<http://leafletjs.com/>

Exemples de sites utilisant Leaflet

- ▶ The New York Times
- ▶ The Washington Post
- ▶ flickr

- ▶ Distribution géographique des 417 homicides survenus à Chicago en 2013

Leaflet (librairie R)

Leaflet for R

Introduction

The Map Widget

Basemaps

Markers

Popups

Lines and Shapes

JSON

Raster Images

Shiny Integration

Colors

Legends

Show/Hide Layers

Introduction

Leaflet is one of the most popular open-source JavaScript libraries for interactive maps. It's used by websites ranging from [The New York Times](#) and [The Washington Post](#) to [GitHub](#) and [Flickr](#), as well as GIS specialists like [OpenStreetMap](#), [Mapbox](#), and [CartoDB](#).

This R package makes it easy to integrate and control Leaflet maps in R.

Features

- Interactive panning/zooming
- Compose maps using arbitrary combinations of:
 - Map tiles
 - Markers
 - Polygons
 - Lines
 - Popups
 - GeoJSON
- Create maps right from the R console or RStudio
- Embed maps in [knitr/R Markdown](#) documents and [Shiny](#) apps
- Easily render `Spatial` objects from the `sf` package, or data frames with latitude/longitude columns
- Use map bounds and mouse events to drive Shiny logic

Fork me on GitHub

<https://rstudio.github.io/leaflet/>

Exemples de couches (parmi une centaine disponible)



<http://leaflet-extras.github.io/leaflet-providers/preview/>

Exemple en épidémiologie

Development of an open source tool for mapping disease clusters

Catherine M Smith, Helen Maguire, Charlotte Anderson, Andrew C Hayward

Abstract

Published Online
November 13, 2015

Department of Infectious Disease Informatics, Farr Institute of Health Informatics Research, University College London, London, UK
(C M Smith MSc, C Hayward MD); Field Epidemiology Service - South London, Public Health England, London, UK
(H Maguire MBBS, MSc); and Research Department Infection and

Population Health, for Infectious Disease Epidemiology, University College London, London, UK
(H Maguire)

Correspondence to:
ss Catherine M Smith,
t of Infectious Disease
atics, Farr Institute of
informatics Research,
ston Road, University
le London, London
NW1 2DA, UK
l.smith.13@ucl.ac.uk

Background Mapping is an important step in investigations of infectious disease outbreaks. Despite advances in molecular characterisation of pathogens that allow delineation of outbreaks with greater precision, geospatial methods that complement these analyses are rarely used in real time. Barriers include lack of flexibility, expense of specialised software, and the need for trained personnel. This study aimed to develop a tool with free, open source software that could be used by public health professionals to display clusters of disease on a map, and to demonstrate its utility through the exemplar of molecular clusters of tuberculosis in London.

Methods We developed a bespoke interactive mapping tool using Shiny, a web application framework for the statistical software package, R. Mapping was enabled using Leaflet, a JavaScript library for interactive mapping, through the R package leaflet. Data were extracted from the Enhanced Tuberculosis Surveillance System for cases of tuberculosis with a residential postcode in London, who were part of a molecular cluster, and were notified between Jan 1, 2011, and Dec 31, 2013.

Findings Data subsets can be interactively displayed on the basis of cluster name, notification date, demographics and reported risk factors. In addition to mapping, epidemic curves and summary tables are automatically produced. The tool was used to explore 3194 cases of tuberculosis in 767 clusters in London, and allowed rapid overview of the geographical, temporal, and epidemiological features to support cluster investigation.

Interpretation Geographical displays of molecular data can enhance understanding of disease transmission. Advantages of the tool developed in this study include: flexibility, allowing user-defined subsets of data by geographical location to be displayed; potential adaptability to other geocoded health or other contextual data via a web interface, which is user-friendly, needs minimal training to operate, and does not require upload of identifying information to the internet. A potential limitation is maintenance of code, although the use of open source software also benefits from support and improvement from the wider developer community. Future work will involve testing of the tool with potential users to assess its utility in practice.

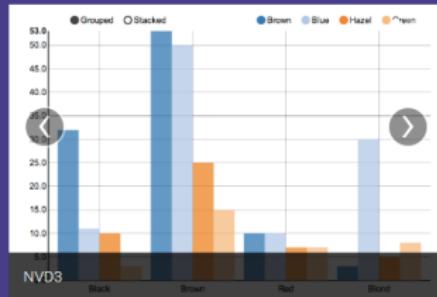
Package *rcharts*

rcharts

[rCharts](#) [Home](#) [Quick Start](#) [Gallery](#) [Libraries](#) [Star 924](#) [Fork 594](#)

 rCharts
is an R package to create, customize and publish interactive javascript visualizations from R using a familiar lattice style plotting interface.

[Learn more »](#)



A stacked bar chart titled "NVD3" showing the distribution of hair colors (Black, Brown, Red, Blond) across four eye colors (Brown, Hazel, Green, Blue). The Y-axis represents frequency from 0.0 to 53.0. The X-axis categories are Black, Brown, Red, and Blond. The legend indicates "Grouped" (solid bars) and "Stacked" (stacked bars). The data shows that Brown eyes are most common overall, followed by Blue eyes.

Hair Color	Brown	Hazel	Green	Blue
Black	~48	~10	~10	~10
Brown	~50	~25	~15	~10
Red	~10	~5	~5	~10
Blond	~28	~5	~5	~10

Familiar Plotting Interface

rCharts uses a plotting interface that R users are already familiar with. You can use a formula interface as used in the lattice package, or a more traditionally interface used with base graphics.

[View Details](#)

Multiple Charting Libraries

rCharts supports multiple javascript charting libraries, each with its own strengths. Each of these libraries has multiple customization options, most of which are supported within rCharts.

[View Details](#)

Easy to Share

rCharts allows you to share your visualization in multiple ways. You can save it as a standalone page, embed it in a shiny application, or even include it as a part of a blog post or tutorial.

[View Details](#)

démo

Package *googleVis*

googleVis

Basé sur les outils développés par Google

The screenshot shows the Google Developers Charts Gallery page. The left sidebar contains a navigation menu with sections like Overview, Hello, Charts!, Quickstart, Load the Libraries, Prepare the Data, Customize the Chart, Draw the Chart, Chart Types, Chart Gallery, Annotation Charts, Area Charts, Bar Charts, Bubble Charts, Calendar Charts, Candlestick Charts, Column Charts, Combo Charts, Diff Charts, Donut Charts, Gantt Charts, Gauge Charts, and Geo Charts. The main content area displays seven examples of charts: a Geo Chart showing France with regions highlighted in blue; a Scatter Chart showing a positive correlation between two variables; a Column Chart with blue and yellow bars; a Histogram with blue bars; a Bar Chart with blue bars; and a Combo Chart featuring a bar chart with orange and green bars and a line chart overlaid.

<https://developers.google.com/chart/interactive/docs/gallery>

googleVis

googleVis examples

It may take a little while to load all charts. Please be patient. All charts require an Internet connection.

These examples are taken from the googleVis demo. You can execute the demo via

```
library(googlevis)
demo(googlevis)
```

For more details about the charts and further examples see the helpfiles of the individual googleVis function and review the [Google Charts API documentation](#) and [Terms of Service](#).

Line chart

```
df=data.frame(country=c("US", "GB", "BR"),
              val1=c(10,13,14),
              val2=c(23,12,32))
```

```
Line <- gvisLineChart(df)
plot(Line)
```

démo

Package *DT*

DataTables

Module d'extension (plug-in) de JQuery, une librairie JavaScript



DataTables Table plug-in for jQuery

DataTables is a plug-in for the [jQuery](#) Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, and will add advanced interaction controls to any HTML table.

- Pagination, instant search and multi-column ordering
- Supports almost any data source:
 - DOM, Javascript, Ajax and [server-side processing](#)
- Easily theme-able: [DataTables](#), [jQuery UI](#), [Bootstrap](#), [Foundation](#)
- Wide variety of [extensions](#) inc. [Editor](#), [TableTools](#), [FixedColumns](#) and more
- Extensive options and a beautiful, expressive [API](#)
- Fully [internationalisable](#)
- Professional quality: backed by a suite of 2900+ unit tests
- Free open source software ([MIT license](#))! [Commercial support available](#).
- [Show more features...](#)

How easy is it to use DataTables? Take a peek at the code below: a single function call to initialise the table is all it takes!

```
1 | $(document).ready(function(){  
2 |     $('#myTable').DataTable();  
3 |});
```

Getting started with DataTables is as simple as including two files in your web-site, the CSS styling and the DataTables script itself. These two files are available on the [DataTables CDN](#):

CSS //cdn.datatables.net/1.10.10/css/jquery.dataTables.css

JS //cdn.datatables.net/1.10.10/js/jquery.dataTables.js

An example of DataTables in action is shown below.

<http://datatables.net/>

DT (librairie R)

The DT package provides an R interface to the JavaScript library DataTables. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and DataTables provides filtering, pagination, sorting, and many other features in the tables.

You may install the stable version from CRAN, or the development version using `devtools::install_github('rstudio/DT')` if necessary (this website reflects the development version of DT):

```
if (!require("DT")) install.packages("DT")  
sessionInfo()
```

```
## R version 3.2.2 (2015-08-14)  
## Platform: x86_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 15.04  
##  
## locale:  
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C  
## [3] LC_TIME=en_US.UTF-8       LC_COLLATE=en_US.UTF-8  
## [5] LC_MONETARY=en_US.UTF-8   LC_MESSAGES=en_US.UTF-8  
## [7] LC_PAPER=en_US.UTF-8     LC_NAME=C  
## [9] LC_ADDRESS=C              LC_TELEPHONE=C  
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

<https://rstudio.github.io/DT/>

Analyses des données - Package surveillance

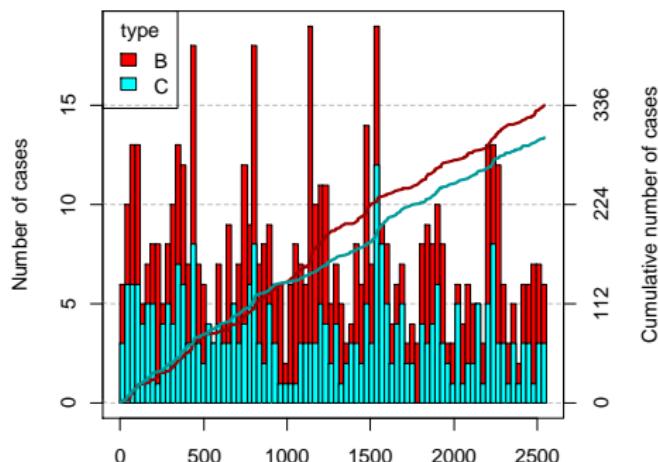
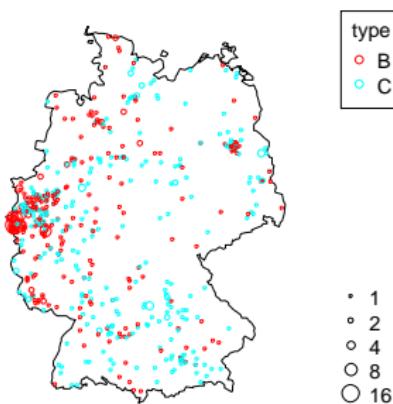
surveillance

- ▶ Package existant depuis plus de 10 ans, mis à jour régulièrement.
- ▶ Créateur et principal contributeur : Michael Höhle
- ▶ Package pour la modélisation temporelle et spatio-temporelle de phénomènes épidémiques.

[pdf \(321 pages\)](#)

Exemple 1 : 636 cas d'infections invasives à méningocoque en Allemagne

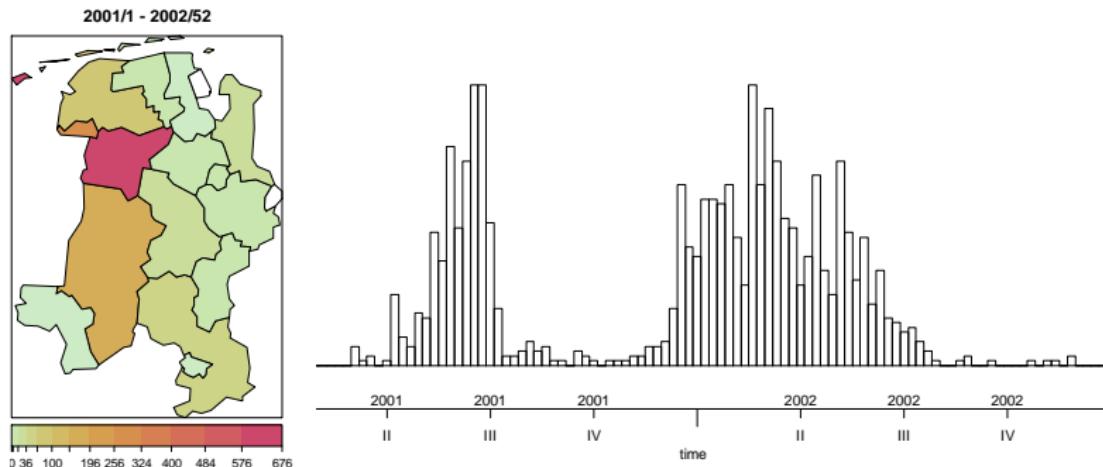
```
library(surveillance)
data("imdepi")
plot(imdepi,"space")
plot(imdepi,"time")
```



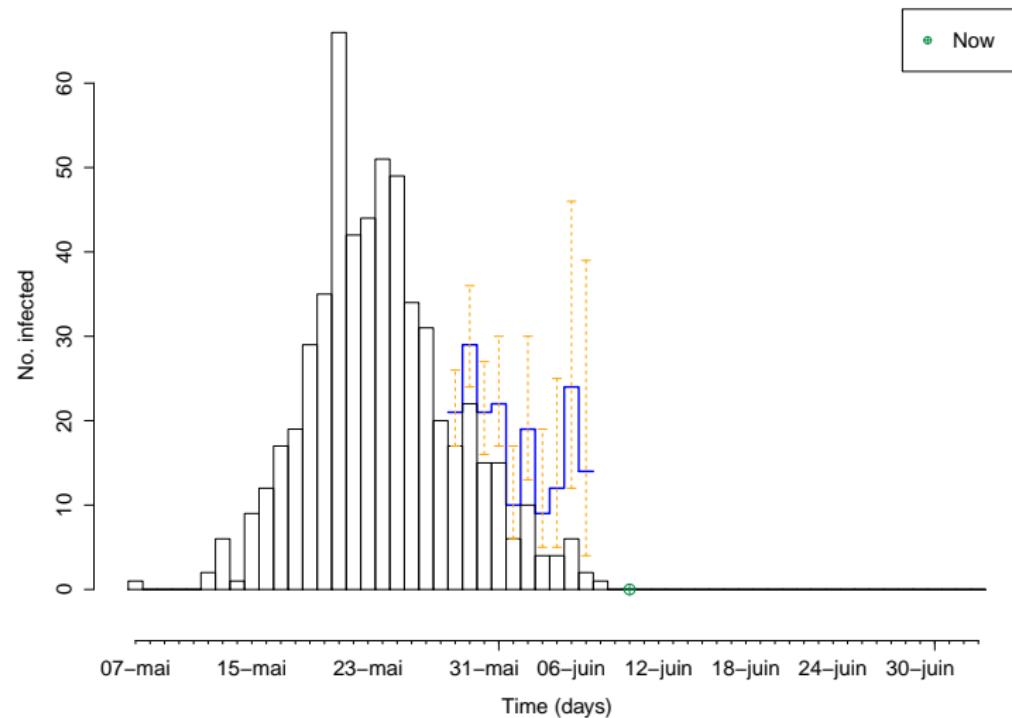
animation

Exemple 2 : Nombre hebdomadaire rapporté de rougeole dans la région Weser-Ems - Allemagne 2001-2002

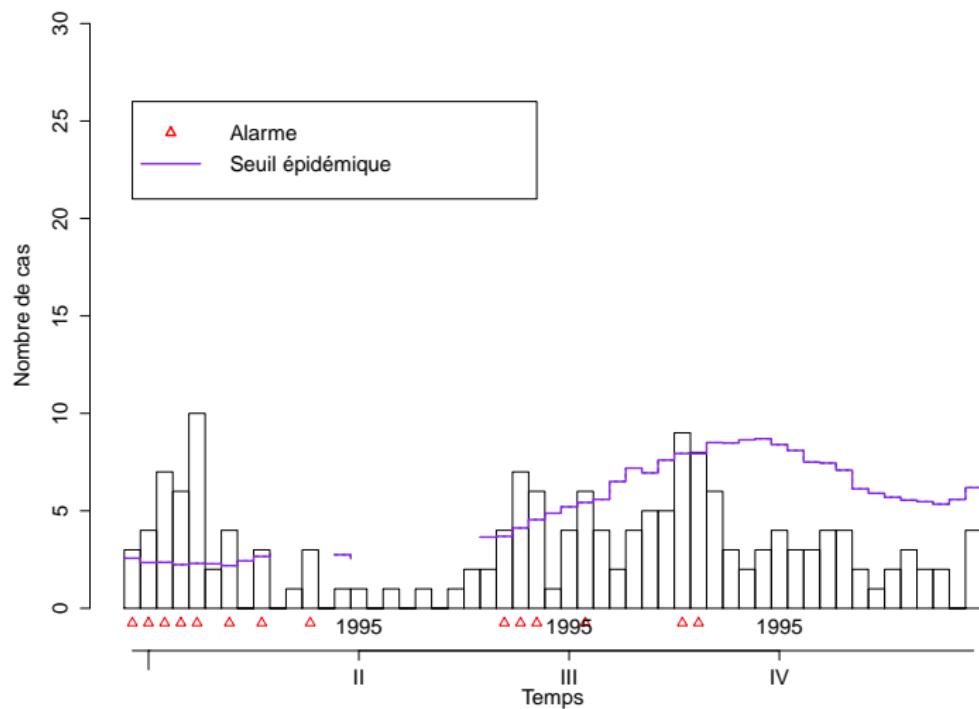
```
data("measlesWeserEms")
plot(measlesWeserEms,type=observed unit)
plot(measlesWeserEms,type=observed time)
```



Exemple 3 : Nowcasting pour l'épidémie de STEC O104 :H4, Allemagne, 2011



Exemple 4 : Détection d'événements inhabituels -
Salmonella Newport, Allemagne, 1995



Intégration de tous ces outils dans un site web

- ▶ Tous ces outils (graphiques, statistiques) ont une réelle utilité dans le domaine de la surveillance épidémiologique
- ▶ Il serait intéressant s'ils pouvaient être à la disposition d'utilisateurs ne connaissant ni le langage de programmation web, ni R.
- ▶ A l'image de l'outil de requêtes du RKI : [SurvStat@RKI 2.0](#)
- ▶ C'est possible en R, grâce au package shiny développé par RStudio.

Création de site web - Package *shiny*

Shiny by RStudio

[OVERVIEW](#)

[TUTORIAL](#)

[ARTICLES](#)

GALLERY

[REFERENCE](#)

[DEPLOY](#)

[HELP](#)

Gallery

This gallery contains useful examples to learn from. Visit the [Shiny User Showcase](#) to see an inspiring set of sophisticated apps.

Interactive visualizations

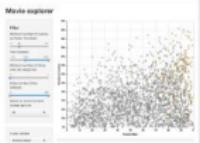
Shiny is designed for fully interactive visualization, using JavaScript libraries like [d3](#), [Leaflet](#), and [Google Charts](#).



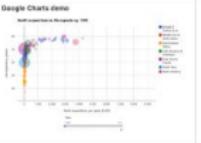
SuperZip example



Bus dashboard



Movie explorer



Google Charts

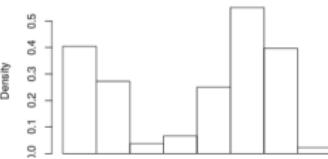
Le principe général

2 programmes : user-interface (ui.R) et server.R

Number of bins in histogram (approximate):
10

Show individual observations
 Show density estimate

Geyser eruption duration



Density

Duration (minutes)	Density
[0, 1]	~0.35
[1, 2]	~0.25
[2, 3]	~0.05
[3, 4]	~0.05
[4, 5]	~0.25
[5, 6]	~0.55
[6, 7]	~0.35
[7, 8]	~0.05

ui.R server.R

```
shinyUI(bootstrapPage(  
  selectInput(inputId = "n_breaks",  
    label = "Number of bins in histogram (approximate):",  
    choices = c(10, 20, 35, 50),  
    selected = 20),  
  
  checkboxInput(inputId = "individual_obs",  
    label = strong("Show individual observations"),  
    value = FALSE),  
  
  checkboxInput(inputId = "density",  
    label = strong("Show density estimate"),  
    value = FALSE),  
  
  plotOutput(outputId = "main_plot", height = "300px"),  
  
  # Display this only if the density is shown  
  conditionalPanel(condition = "input.density == true",  
    sliderInput(inputId = "bw_adjust",  
      label = "Bandwidth adjustment:",  
      min = 0.2, max = 2, value = 1, step = 0.2)  
)
```

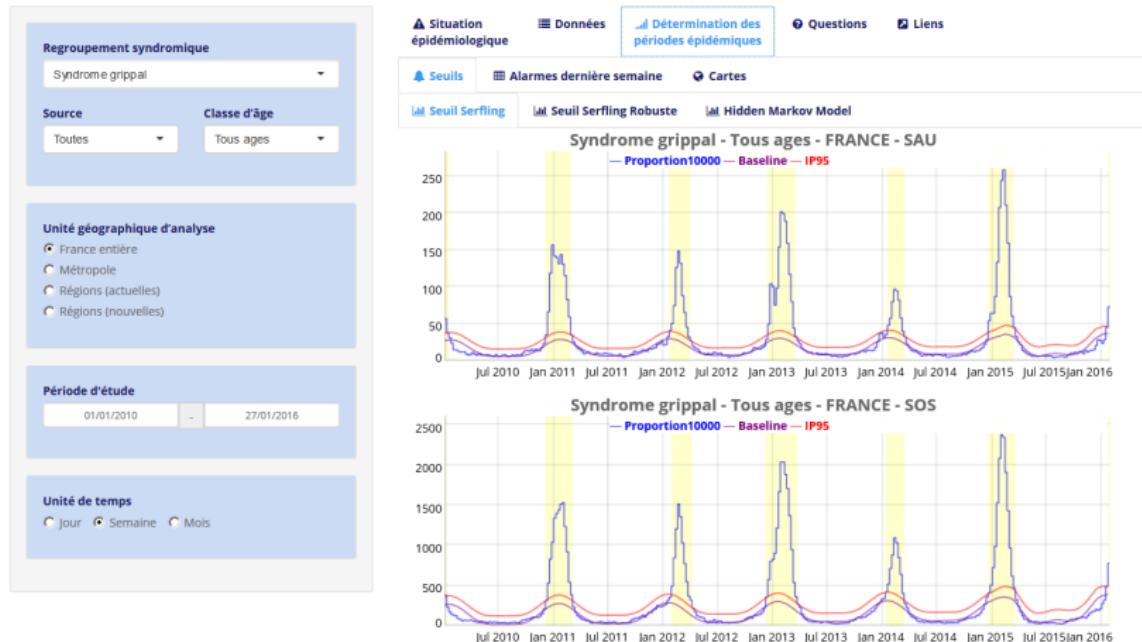
ui.R server.R

```
shinyServer(function(input, output) {  
  
  output$main_plot <- renderPlot({  
  
    hist(faithful$eruptions,  
      probability = TRUE,  
      breaks = as.numeric(input$n_breaks),  
      xlab = "Duration (minutes)",  
      main = "Geyser eruption duration")  
  
    if (input$individual_obs) {  
      rug(faithful$eruptions)  
    }  
  
    if (input$density) {  
      dens <- density(faithful$eruptions,  
        adjust = input$bw_adjust)  
      lines(dens, col = "blue")  
    }  
  })  
})
```

Application MASS

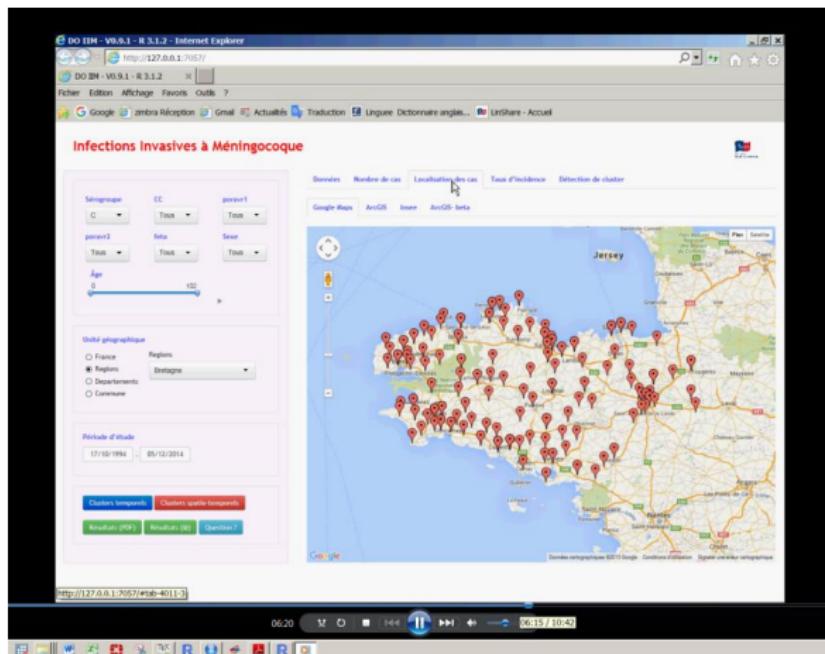
Module d'Analyses Statistiques Sursaud (**MASS**), intégrant des outils de visualisation des données et des méthodes statistiques.

Module d'Analyses Statistiques SurSauD®



Application IIM

Application pour la surveillance des Infections Invasives à Méningocoques (IIM).

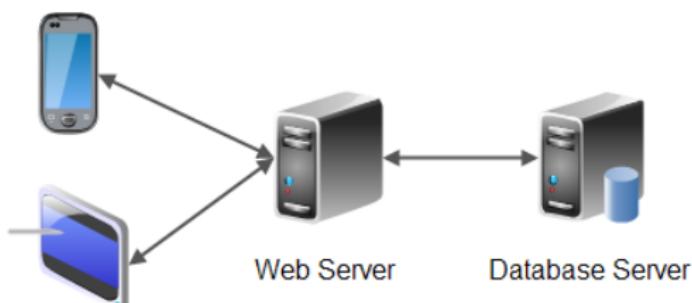


vidéo

Discussion

Les outils dans les 10 prochaines années

- ▶ Recueil de données sur appareils mobiles, ordinateurs.
- ▶ Transmission électronique des données vers un serveur qui héberge une base centralisée.
- ▶ Programmes traitant les données et générant des sorties (graphes, tableaux, cartes, ...)
- ▶ Serveur web permettant de visualiser ces sorties.



Le virage à prendre

L'agence ne doit pas rater :

- ▶ la modernisation du recueil de données
- ▶ le choix des outils d'analyses de données
- ▶ le choix des outils de restitution (aux agents, partenaires, grand public)

Aujourd'hui :

- ▶ La déclaration électronique est en marche
- ▶ La restitution passera obligatoirement par le web
- ▶ Le choix des outils d'analyses des données est encore incertain à ce jour, mais R est un bon candidat.

Les outils à utiliser

Ils doivent être :

- ▶ développés pour répondre au plus près aux missions de l'agence (descriptives, analytiques)
- ▶ ambitieux car le choix des outils ne doit pas reposer sur une restitution minimale mais au contraire maximale, en utilisant les méthodes statistiques et les outils de restitution les plus récents
- ▶ évolutifs, car les missions, les besoins et les techniques évoluent.
- ▶ capables de traiter de grands volumes de données (SurSaud, BDMA par exemple)
- ▶ être rapides.

La question du développement en interne et/ou en externe de ces outils est donc crucial.