

Calcul de domaines vitaux : un exemple d'utilisation de Shiny avec des cartes interactives.

Cyril Bernard, CEFE-CNRS

Apéro technique OSU-OREME, 15 décembre 2016

Problématique

Objectif : pouvoir calculer le **domaine vital** d'un animal à part de **positions GPS** issues d'un tracker, et le visualiser sur une **carte**.

L'utilisateur doit pouvoir entrer ses propres données.

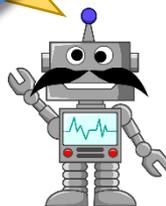
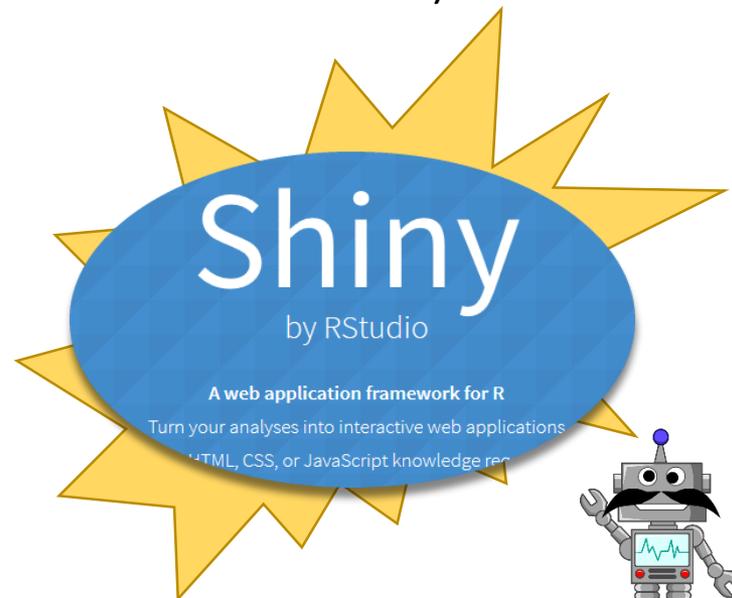
Comment : utilisation de R et du package **adehabitatHR** (*A collection of tools for the estimation of animals home range*) avec la méthode par estimation de noyau de densité.

L'utilisateur doit pouvoir faire varier quelques paramètres basiques.

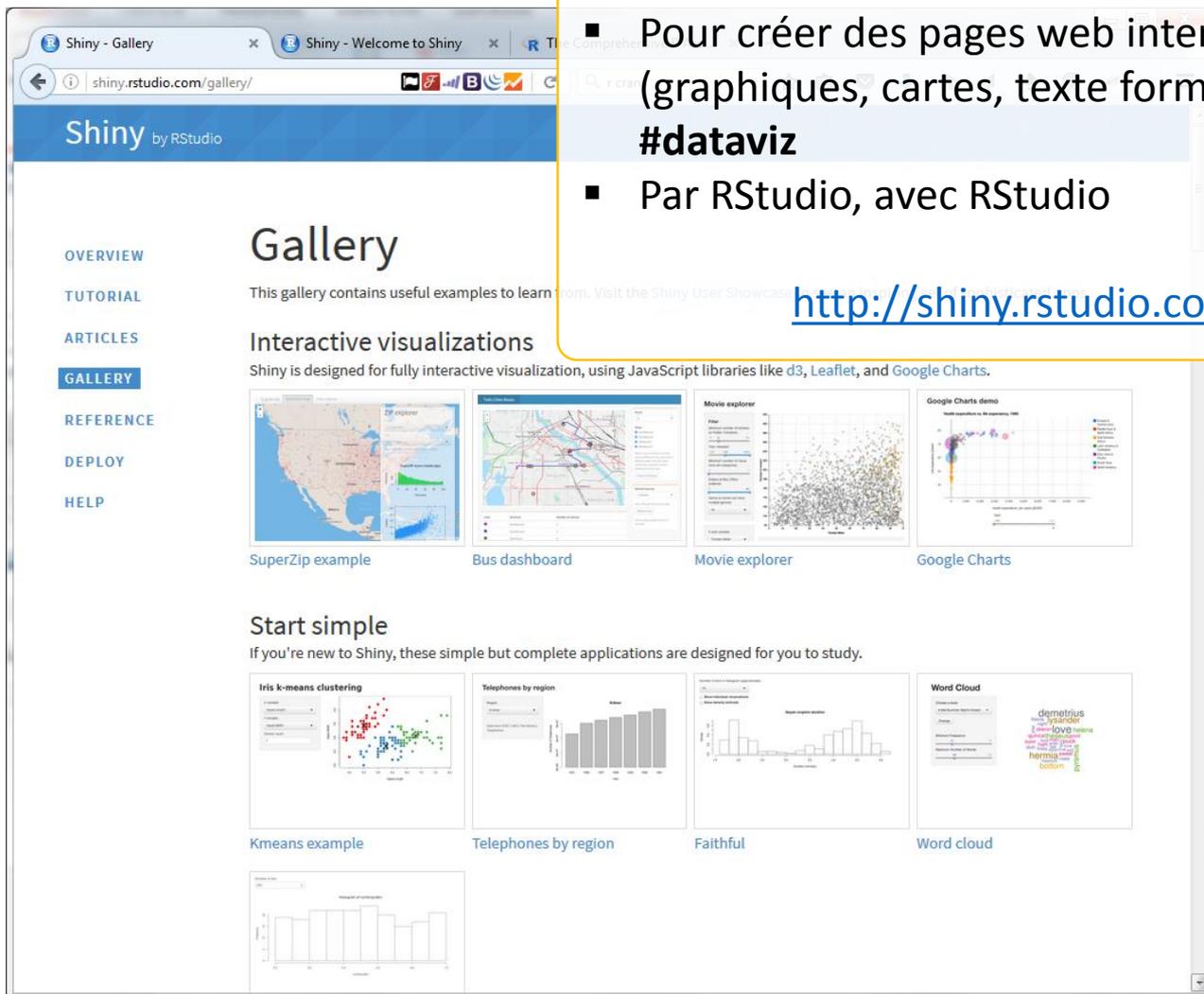
Pour qui : un public de non-initiés à R (exemple : dans le cadre d'un cours)

Comment faire exécuter un script R à quelqu'un ...

- qui veut utiliser ses propres données ?
- qui ne connaît pas le logiciel R ?
- qui n'a peut-être pas R sur sa machine ?



Shiny : un framework web sous



The screenshot shows the Shiny gallery website. The browser address bar displays 'shiny.rstudio.com/gallery/'. The page title is 'Shiny by RStudio'. A navigation menu on the left includes 'OVERVIEW', 'TUTORIAL', 'ARTICLES', 'GALLERY' (highlighted), 'REFERENCE', 'DEPLOY', and 'HELP'. The main content area is titled 'Gallery' and contains the text: 'This gallery contains useful examples to learn from. Visit the Shiny User Showcase'. Below this, there is a section for 'Interactive visualizations' with the text: 'Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.' This section features four thumbnails: 'SuperZip example' (a map of the USA), 'Bus dashboard' (a map with bus routes), 'Movie explorer' (a scatter plot), and 'Google Charts demo' (a line chart). Below this is a 'Start simple' section with the text: 'If you're new to Shiny, these simple but complete applications are designed for you to study.' This section features four thumbnails: 'Kmeans example' (a scatter plot), 'Telephones by region' (a bar chart), 'Faithful' (a histogram), and 'Word cloud' (a word cloud). A fifth thumbnail is partially visible at the bottom left.

- Pour créer des pages web interactives (graphiques, cartes, texte formaté) avec R **#dataviz**
- Par RStudio, avec RStudio

<http://shiny.rstudio.com>

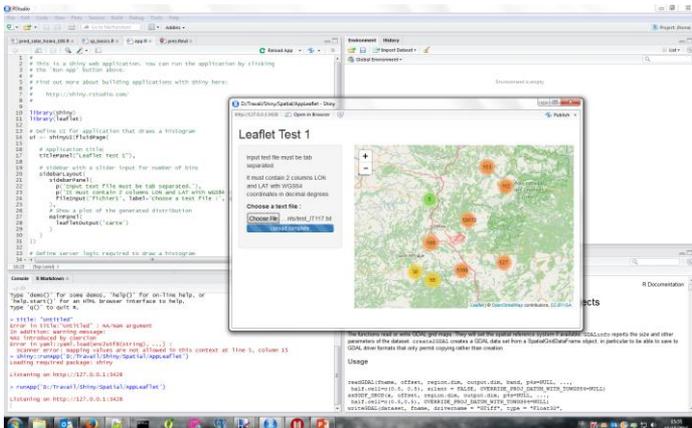
Shiny : comment créer et déployer une application ?

Prérequis : connaître R, avoir un poste avec Rstudio (à jour)

Etape 1 : installez le package shiny

```
> install.packages("shiny")
```

Etape 2 : créez, testez et déboguez votre app dans RStudio



Etape 3 : déployez votre app sur <http://www.shinyapps.io/> (ou votre propre serveur Shiny)

Physionomie d'une app Shiny

The screenshot shows a web browser window displaying a Shiny application. The browser's address bar shows the URL `shiny.rstudio.com`. The application's main content area features a heading "Here" followed by a red-bordered box containing the text "2 fichiers .R". Below this, the text "Shiny apps are easy to write. No web development skills are required." is displayed. Two orange arrows point from the "2 fichiers .R" box to two circular callouts labeled "ui.R" and "server.R".

The application interface includes a control panel on the left with the following elements:

- A dropdown menu for "Number of bins in histogram (approximate):" set to "20".
- Two checkboxes: "Show individual observations" (unchecked) and "Show density estimate" (checked).
- A plot titled "Geyser eruption duration" showing a histogram of eruption durations with a blue density curve overlaid. The x-axis is labeled "Duration (minutes)" and ranges from 1.5 to 5.0. The y-axis is labeled "Density" and ranges from 0.0 to 0.6.
- A slider for "Bandwidth adjustment:" ranging from 0.2 to 2, with the current value set to 1.

On the right side of the interface, a code editor displays the following R code:

```
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)  
  )  
))
```

Définir les entrées / sorties dans **ui.R** (présentation)

*Input functions

*Output functions

The screenshot displays a Shiny application in a web browser. The browser tabs show "Shiny" and "Shiny - Welcome to Shiny". The address bar is "shiny.rstudio.com". The main content area has the heading "Here is a Shiny app" and the text "Shiny apps are easy to write. No web development skills are required."

The application interface includes the following elements:

- Input controls:**
 - A dropdown menu labeled "Number of bins in histogram (approximate):" with the value "20" selected.
 - Two checkboxes: "Show individual observations" (unchecked) and "Show density estimate" (checked).
 - A slider control labeled "Bandwidth adjustment:" with a range from 0.2 to 2 and a current value of 1.
- Output:**
 - A histogram titled "Geyser eruption duration" showing the density of eruption durations. The x-axis is "Duration (minutes)" ranging from 1.5 to 5.0, and the y-axis is "Density" ranging from 0.0 to 0.6. A blue density curve is overlaid on the histogram bars.
- Code Editor:**
 - The editor shows the `ui.R` file with the following code:

```
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)  
  )  
))
```

Interactions entrée/sortie dans `server.R`

Rendre les sorties **réactives** aux entrées

Traiter les données, calculer (possibilité d'utiliser des packages externes)

The screenshot displays a Shiny application window with the following components:

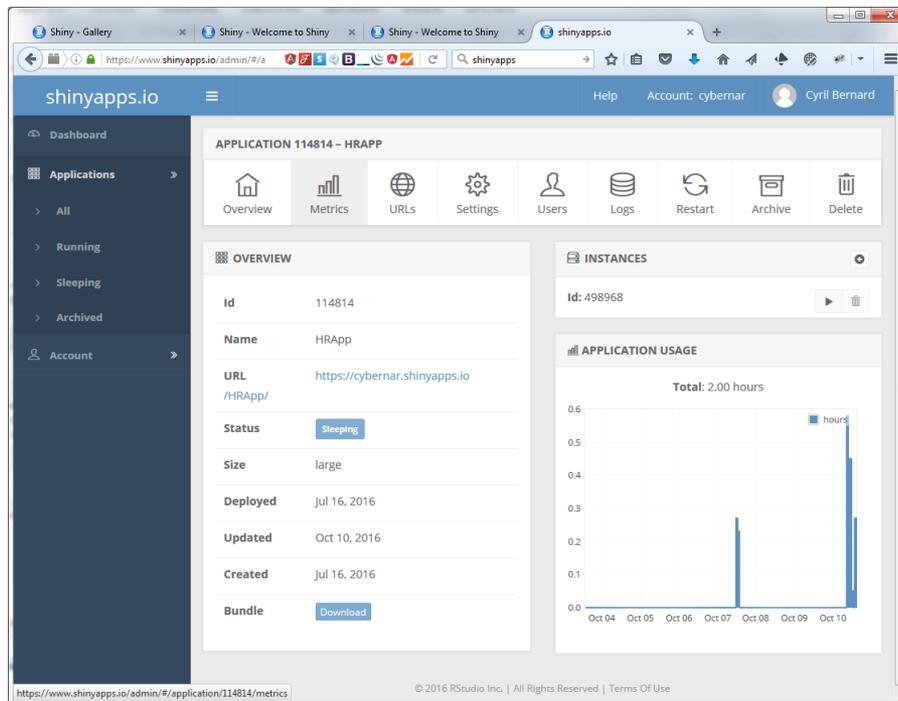
- Browser:** Shows the URL `shiny.rstudio.com` and several open tabs.
- UI (Left Panel):**
 - Control: "Number of bins in histogram (approximate):" with a dropdown menu set to "20".
 - Checkboxes: "Show individual observations" and "Show density estimate".
 - Plot: A histogram titled "Geyser eruption duration" with a blue density curve overlaid. The x-axis is "Duration (minutes)" ranging from 1.5 to 5.0, and the y-axis is "Density" ranging from 0.0 to 0.6.
 - Control: "Bandwidth adjustment:" with a slider ranging from 0.2 to 2.0, currently set at 1.0.
- Server (Right Panel):**
 - Code editor showing the `server.R` file with the following code:

```
shinyServer(function(input, output) {  
  output$main_plot <- renderPlot({  
    hist(faithful$eruptions,  
         probability = TRUE,  
         breaks = as.numeric(input$nb_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")  
    }  
  })  
})
```

Déploiement d'une app Shiny

Sur shinyapps.io : gratuit pour 5 apps max. (25 h d'utilisation / mois)
-> déploiement facile, interface d'administration

```
library(rsconnect)  
deployApp()
```



Shiny Server sur mon serveur Linux : version 'de base' gratuite, pas de limite d'utilisation. Exemple : http://data.oreme.org/plankton/plankton_thau_R

Physionomie de l'application HRApp

* File input *

Nb points, href

H input

* Calculate HR *

* % UD input *

HR area

Home Range App v1

Select input text file :

Parcourir... test_IT117.txt
Upload complete

N points = 15071
href = 1756.1

h =
1756

Estimate HR

% of UD in HR
50 55 60 65 70 75 80 85 90 95

HR Size (km2) = 847.4876

Download

About

App created by Cyril Bernard, SIE, CEFE-CNRS.

The data processing made in this app is based on the [adehabitatHR](#) package.

Calenge, C. (2006) The package adehabitat for the R software: a tool for the analysis of space and habitat use by animals. *Ecological Modelling*, 197, 516-519

How to use

Map output
(leaflet)

* REACTIVE INPUT *

INPUT

OUTPUT

Diagramme séquences de l'application HRApp

packages : shiny, leaflet,
sp, raster, rgdal, adehabitatHR

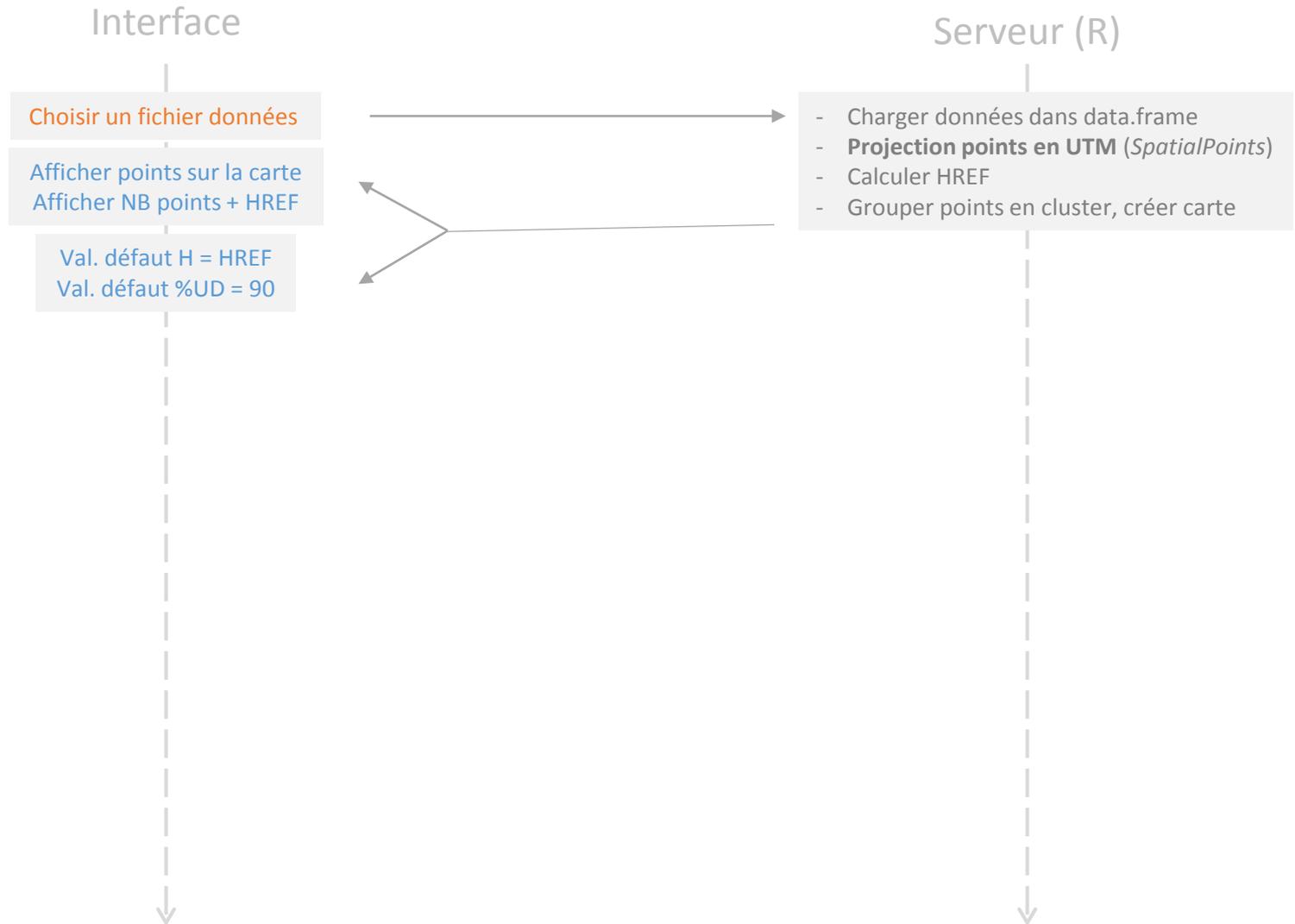


Diagramme séquences de l'application HRApp

packages : shiny, leaflet,
sp, raster, rgdal, adehabitatHR

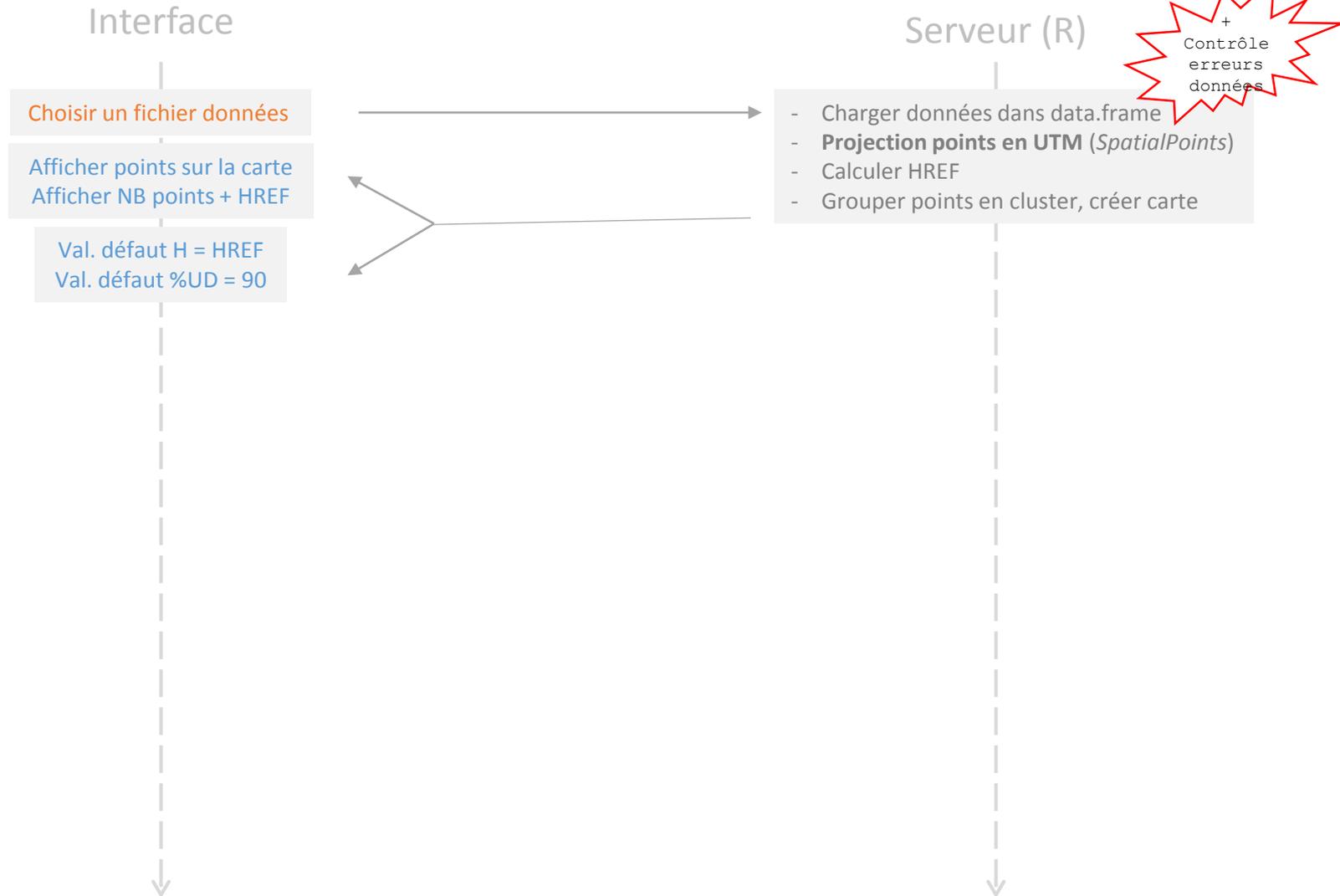


Diagramme séquences de l'application HRApp

packages : shiny, leaflet,
sp, raster, rgdal, adehabitatHR

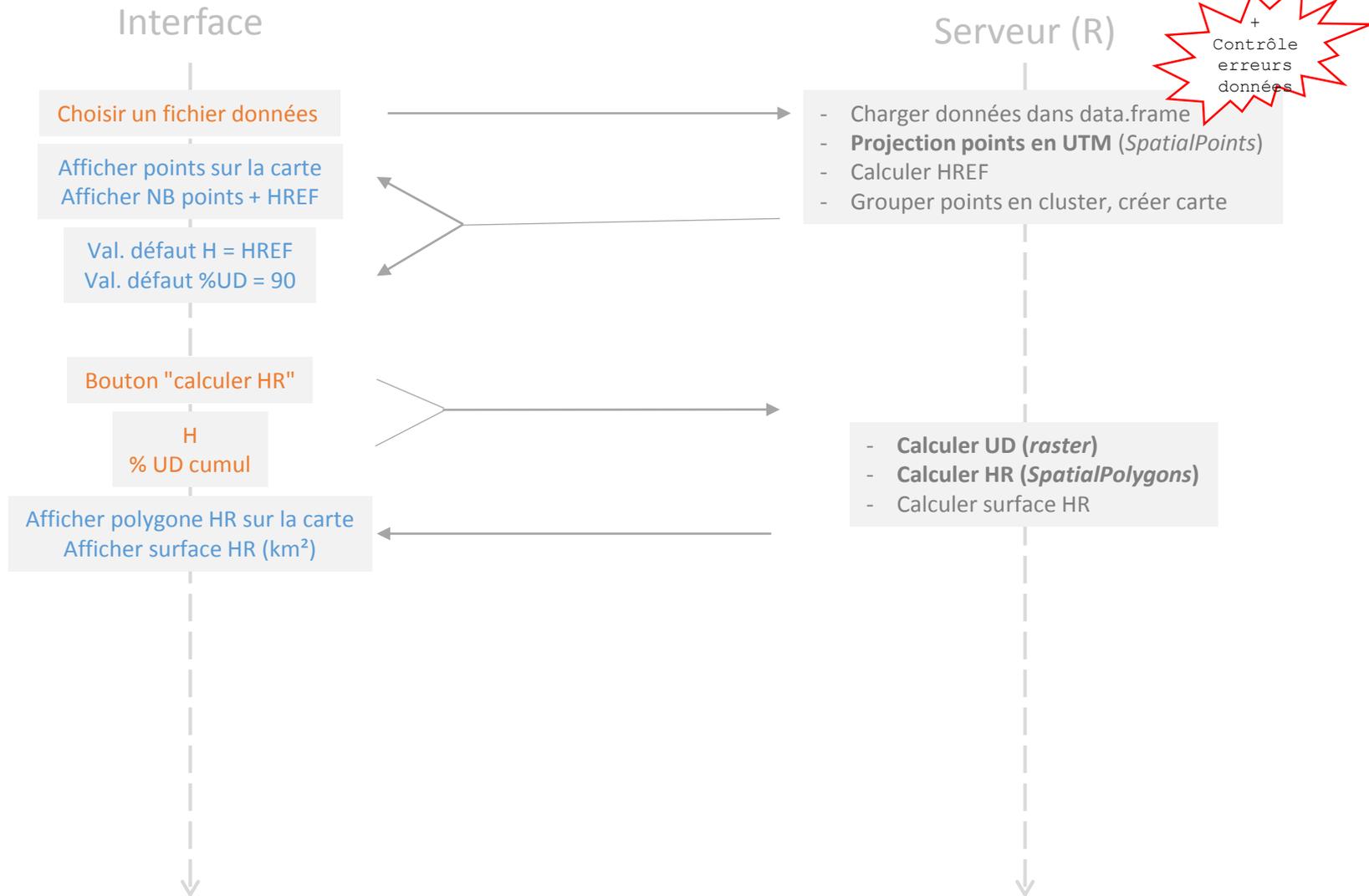


Diagramme séquences de l'application HRApp

packages : shiny, leaflet,
sp, raster, rgdal, adehabitatHR

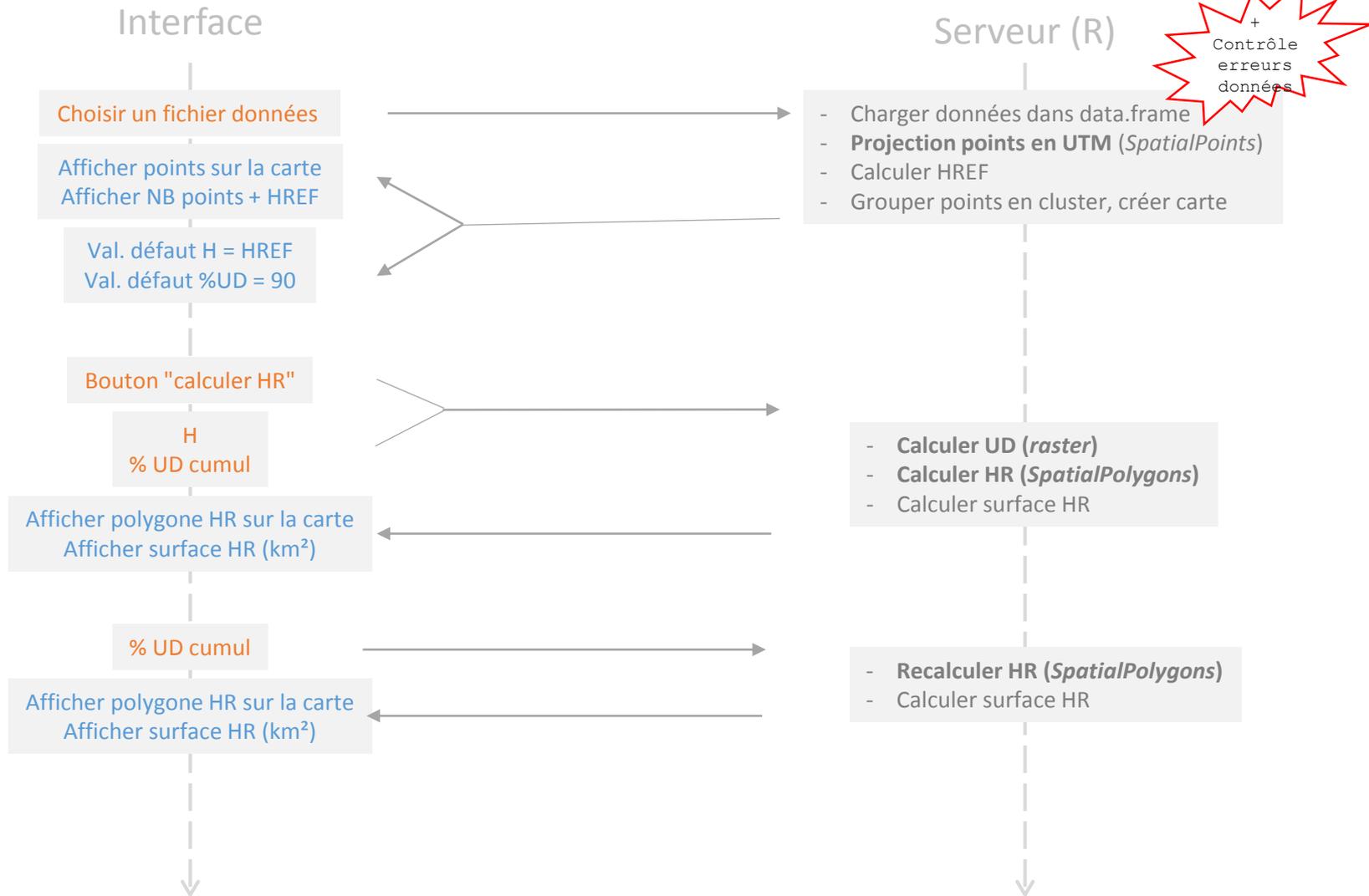
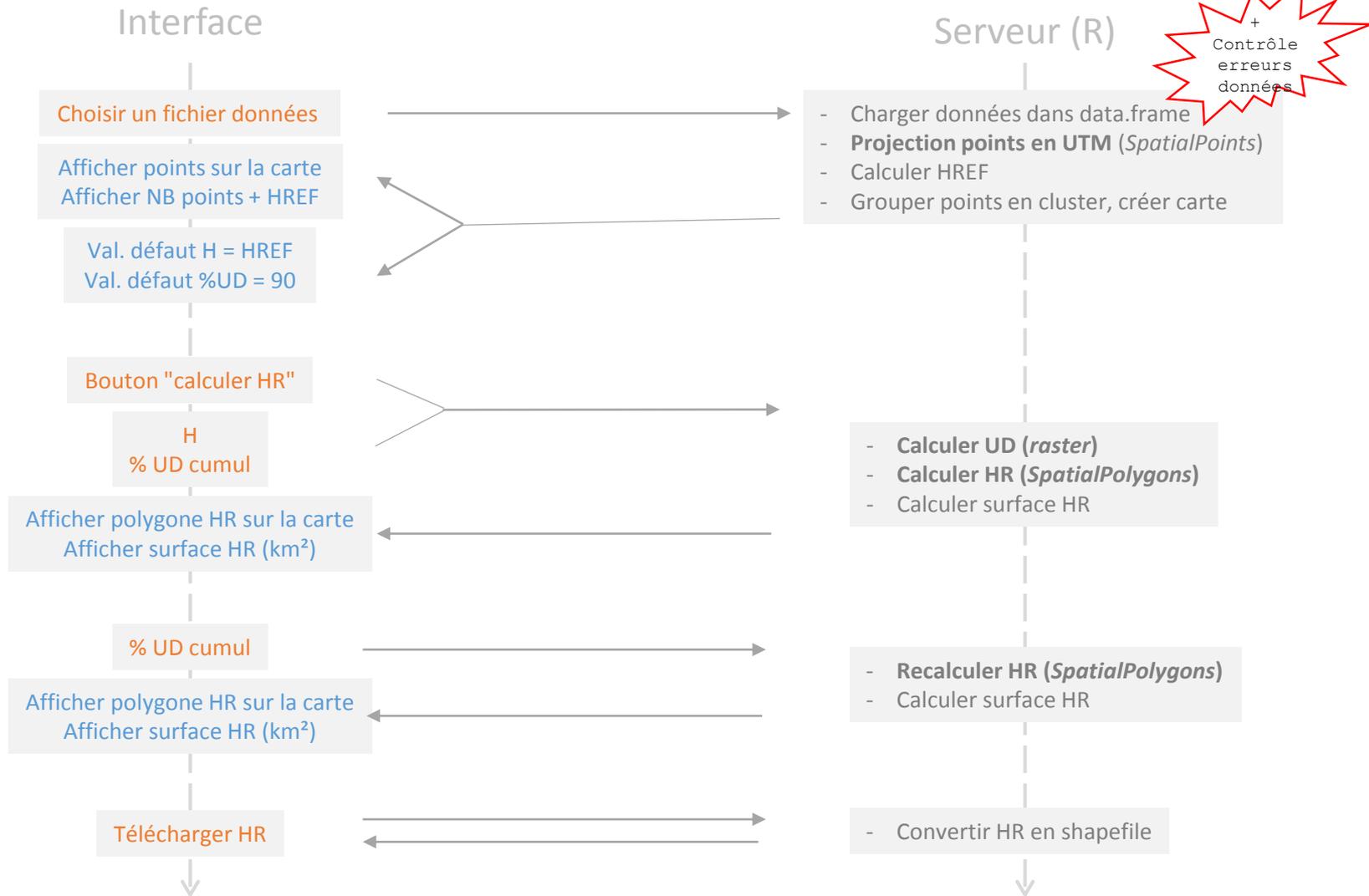


Diagramme séquences de l'application HRApp

packages : shiny, leaflet,
sp, raster, rgdal, adehabitatHR





Démo !!

<https://cybernar.shinyapps.io/HRApp/>

Conclusion : tutoriel et articles

Home Range App v1 x Shiny x Shiny - Hosting a... x How To Set Up Shi... x Shiny - Welcome to... x

shiny.rstudio.com/tutorial/lesson1/

Shiny by RStudio

OVERVIEW < Lesson 1 2 3 4 5 6 7 >

TUTORIAL

ARTICLES

GALLERY

REFERENCE

DEPLOY

HELP

LESSON 1

Welcome to Shiny

Shiny is an R package that makes it easy to build interactive web applications (apps) straight from R. This lesson will get you started building Shiny apps right away.

If you still haven't installed the Shiny package, open an R session, connect to the internet, and run

```
> install.packages("shiny")
```

Note that the preview release of RStudio IDE contains new features designed for Shiny. [Download it here.](#)

Examples

Hello Shiny!

Number of bins: 1 30 50

Histogram of x

Frequency

x