MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

Formation-R-perfectionnement

Module Initiation à l'écriture d'applications Shiny

Sommaire

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

0.1 Avant-propos

Ce diaporama de formation a été rédigé dans le but d'être le support visuel des formations dispensées au SSM AgricultureA.

Ces formations s'adressent à des agents qui ont suivi **la formation R initialisation**.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

0.2 Avant-propos

Elles sont données en présentiel sur une durée **de trois jours**, les modules de cette formation sont ajustables suivants le choix des agents.

Champ couvert par cette formation

Ce support couvre l'initation à l'écriture d'application Shiny dans l'environnement du Minsitère.

Pour information, ce module de la formation R-perfectionnement se décompose en :

1. Introduction
2. Créer une interface
3. Partie interface graphique
4. Partie serveur
5. Réactivité
6. Le programme complet

Ce module est orienté pour être utile aux agents du SSM Agriculture et se concentre sur une utilisation de R via RStudio qui est mise à disposition des agents sur la plateforme interne Cerise basée sur RStudio Workbench.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1 Introduction

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

1.1 Introduction

Pourquoi créer une interface graphique ?

⇒ Cela permet de pouvoir lancer un programme, modifier des paramètres, visualiser des résultats, sans avoir à manipuler de code

On utilise le package `Shiny`. Ce package permet :

- de séparer la partie interface de la partie traitement
- de créer des interfaces web très variées et personnalisables
- de déployer des applications web

Une application Shiny a vocation à être mise sous Gitlab et comporter 3 environnements (branches) :

- Développement,
- Recette,
- Production

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

2 Créer une interface

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

2.1 Créer une interface

La création d'une application Shiny se fait en **2 étapes**.

1. Interface graphique : créer et agencer les différents éléments (boutons, champs de saisie, affichage de graphiques ou tableaux, …)
2. Serveur : récupérer les paramètres saisies dans l'interface, écrire les traitements R, et afficher les résultats

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

2.2 Créer une interface

- Les utilisateurs manipulent l'interface, le serveur actualise l'affichage des résultats sur l'interface via du code R.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3 Partie interface graphique

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

3.1 Partie interface graphique

- On créer un objet **ui**, auquel on affecte une « page »

```
ui <- fluidPage(….)
```

Il existe plusieurs manières de créer une page, suivant comment on veut la présenter : *fluidPage, fixedPage, bootstrapPage, pageWithSidebar, …*

- On prend ici l'exemple d'une `fluidPage`.
- En paramètre de la page, on indique les éléments qu'elle va contenir.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.2 Partie interface graphique

Une page peut être découpée en « lignes » :

```
ui <- fluidPage(

fluidRow(),

fluidRow()

)
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

3.3 Partie interface graphique

Chaque ligne peut être découpée en « colonnes » :

```
ui <-fluidPage(

  fluidRow(

    column(),

    column()
  )

)
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.4 Partie interface graphique

- L'interface utilisateur (**ui.R**) est la vitrine de l'application. C'est une fenêtre dans laquelle l'utilisateur va « saisir » des informations : **les inputs**. Les valeurs des inputs sont définies par manipulation de **widgets**.
- Les résultats (*outputs*) produits par l'application sont généralement affichés dans cette interface.
- L'interface est une page Web « composée » par shiny.
- Elle peut prendre plusieurs formes : la mise en page (*layout*). L'une des plus classiques est la *sidebarLayout*.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.5 Partie interface graphique

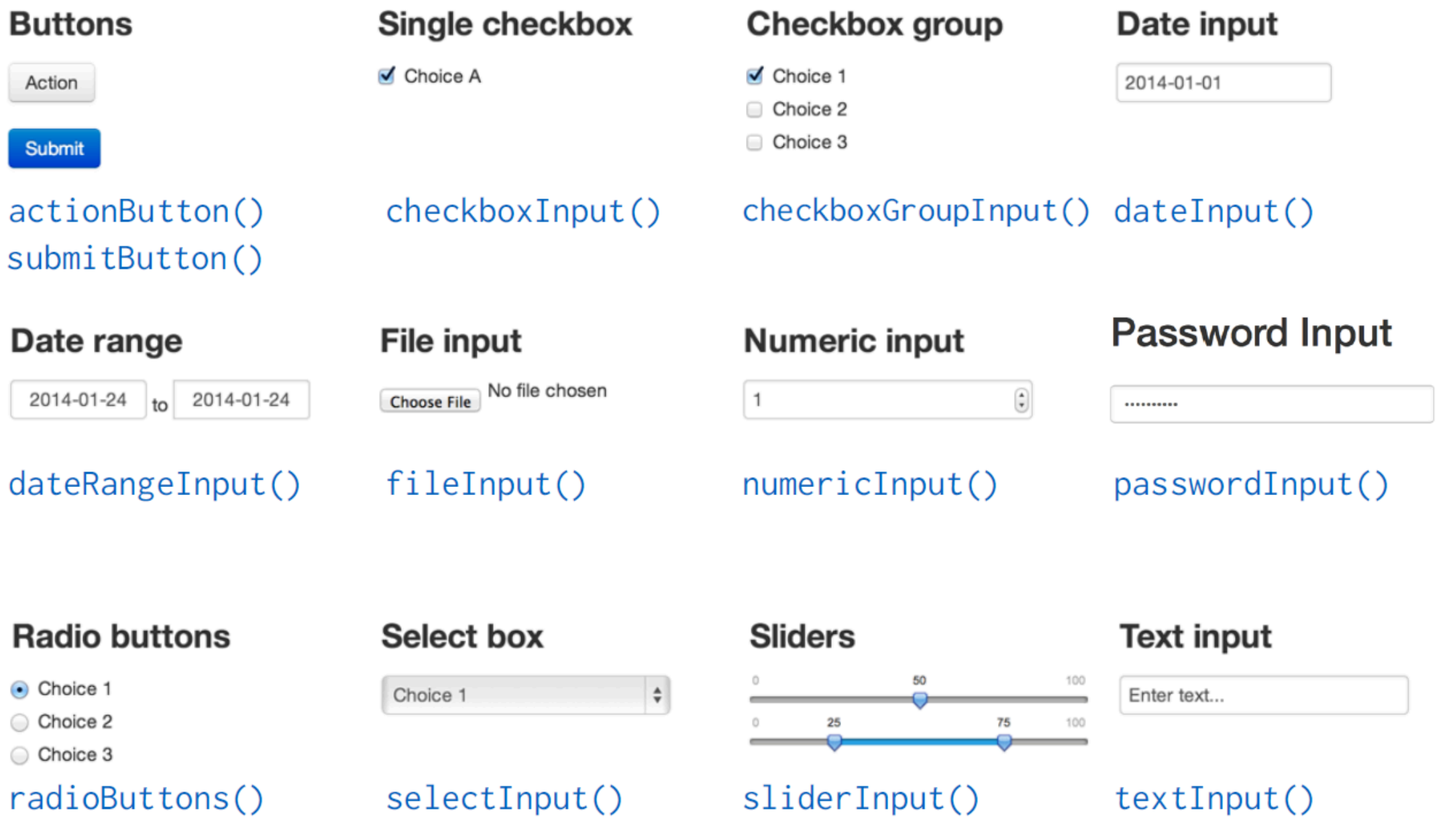
Pour permettre à l'utilisateur de saisir des données en entrée, on utilise des **widgets.** Les plus courants sont :

- **selectInput** : liste déroulante
- **radioButtons** : boutons radio
- **checkboxInput** : case à cocher
- **sliderInput** : bouton à déplacer sur une barre
- **numericInput** : zone de saisie pour une variable numérique
- **textInput** : zone de saisie pour une variable texte
- **actionButton** : bouton pour effectuer une action
- **fileInput** : bouton pour sélectionner un fichier

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

3.6 Partie interface graphique

Voir http://shiny.rstudio.com/gallery/widget-gallery.html



MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

3.7 Partie interface graphique

Les widgets sont en fait des fonctions qui attendent plusieurs arguments. Les 2 premiers sont :

- le nom du widget (inputId)
- le label, c'est à dire le texte qui guidera l'utilisateur dans ses choix.

Exemple :

```
selectInput(inputId="select1",

  label="Veuillez faire un choix dans la liste",

  choices = c("choix 1" = 1, "choix 2" = 2, "choix 3" = 3)

)
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.8 Partie interface graphique

Chaque colonne contiendra des éléments :

- Des inputs (actionButton, selectInput, textInput, fileInput, …)
- Des outputs (tableOutput, plotOutput, downloadButton, …)

On indique en premier paramètre de la colonne la taille de celle-ci, sachant que la taille totale d'une ligne (somme des colonnes) est de **12**.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.9 Partie interface graphique

```
ui <-fluidPage(

  fluidRow(

    column(8,

           selectInput()

    ),

    column(4,

           tableOutput()

    )

  )

```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

3.10 Partie interface graphique

Les différents objets finaux (*input ou output*) ont tous des paramètres différents, mais le premier est toujours le même, **inputId** pour les inputs, et **outputId** pour les output.

→ C'est cet identifiant qui permettra de faire le lien entre l'interface graphique et la partie serveur : la partie traitement (le serveur) peut ainsi récupérer les valeurs saisies dans les inputs et d'afficher les résultats dans les outputs.

> ⚠ **Attention**
>
> Comme on va rapidement imbriquer beaucoup de couches d'éléments, il est très important de bien aérer et indenter son code.

4 Interface graphique pour interface graphique

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

4.1 Interface graphique pour interface graphique

- Le package **{shinyuieditor}** permet de construire l'UI sans avoir à coder https://rstudio.github.io/shinyuieditor/ (version préliminaire)
- Voir aussi le package **{flexdashboard}** pour construire des tableaux de bord avec du Rmarkdown https://rstudio.github.io/flexdashboard/articles/shiny.html

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

4.2 Exercice UI (1/3)

Exercice - partie Interface (ui)

Le but de cet exercice est de créer une interface pour afficher des statistiques simples sur des données.

Partie 1 – Interface

1. Ouvrez la table BC.rds.

Récupérer dans un vecteur la liste des régions.

2. On va maintenant initialiser une page vide. Il nous faut 3 objets :
   - un objet ui, en affectant une fluidPage() vide
   - la fonction server(vide aussi)
   - la commande shinyApp(ui, server)

Exécutez le programme.

(suite sur la prochaine slide…)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

4.3 Exercice UI (2/3)

Exercice - partie Interface (ui)

3. Créez une liste déroulante à l'aide de selectInput. Renseignez les paramètres inputId (identifiant), label (texte affiché), et choices (avec la liste des régions créée précédemment).

Créez un tableOutput.

Exécutez le programme.

4. De la même manière, créez sous la liste déroulante une autre liste, contenant les items « Population », « Altitude », et « Superficie ». Puis encore en-dessous un bouton avec la fonction actionButton.

Exécutez le programme.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

4.4 Exercice UI (3/3)

5 Partie serveur

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.1 Partie serveur

On crée une fonction `server`, qui a au moins 2 paramètres :

```
server <- function(input, output) { ... }
```

Dans le corps de la fonction, on peut placer du code R traditionnel.

Le fichier **server.R** contient les instructions nécessaires pour construire l'application et afficher les résultats dans l'interface.

Plus précisément, il contient le code R à exécuter pour effectuer tous les traitements nécessaires à l'application :

- Chargement de données ;
- Calculs ; sélection, extraction, … ;
- Traçage de graphique, affichage de tableau ;

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

5.2 Partie serveur

Il y a 2 particularités principales liée à Shiny dans le code du serveur :

- Les références aux inputs/outputs : récupérer une valeur saisie, afficher tel tableau à l'écran, …
- Les éléments dits « réactifs » : on peut déclencher un traitement particulier sur une action de l'utilisateur

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.3 Partie serveur

Pour récupérer une valeur saisie par l'utilisateur, par exemple un champ de texte, on utilise le paramètre input, avec l'identifiant du champ de saisie (**inputId**)

- partie ui :

```
textInput(inputId="champ_saisie", label= "Saisir :")
```

- partie server :

```
ma_chaine <- input$champ_saisie
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.4 Partie serveur

Pour placer un tableau ou un graphique dans l'interface, on affecte à l'output la table concernée, en lui appliquant une fonction appelée « render »

- partie ui :

```
tableOutput(outputId="resultat")
```

- partie server :

```
output$resultat <- renderTable({ data })
```

Il existe des fonction **render** pour chaque type d'output (`renderTable, renderPlot, renderText, renderImage, …`).

Dans le paramètre de la fonction, dans les accolades, on peut mettre simplement un objet R, ou un bloc de code :

```
renderTable({ head(data, n=10 })
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

5.5 Partie serveur

De la même façon que nous insérons des données dans l'application via les *widget*, il faut s'occuper d'afficher à l'écran les résultats des traitements.

On utilise pour cela les fonctions render* :

- `renderPlot ;`
- `renderImage ;`
- `renderTable, ...`

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.6 Partie serveur

fonction	prend	crée
renderDataTable	tout objet équivalent à un tableau	dataTables.js table
renderImage	liste d'attributs d'image	image HTML
renderPlot	plot	plot
renderPrint	tout output imprimé	texte
renderTable	tout objet équivalent à un tableau	table
renderText	chaîne de caractères	texte
renderUI	objet Shiny tag ou HTML	élément UI(HTML)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.7 Partie serveur

Les fonctions **render*** coté server et **Output* coté ui vont de pair :

ui.R	server.R
imageOutput	renderImage
plotOutput	renderPlot
tableOutput	renderTable
DTOutput	renderDT
textOutput	renderText
uiOutput	renderUI
verbatimTextOutput	renderPrint

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

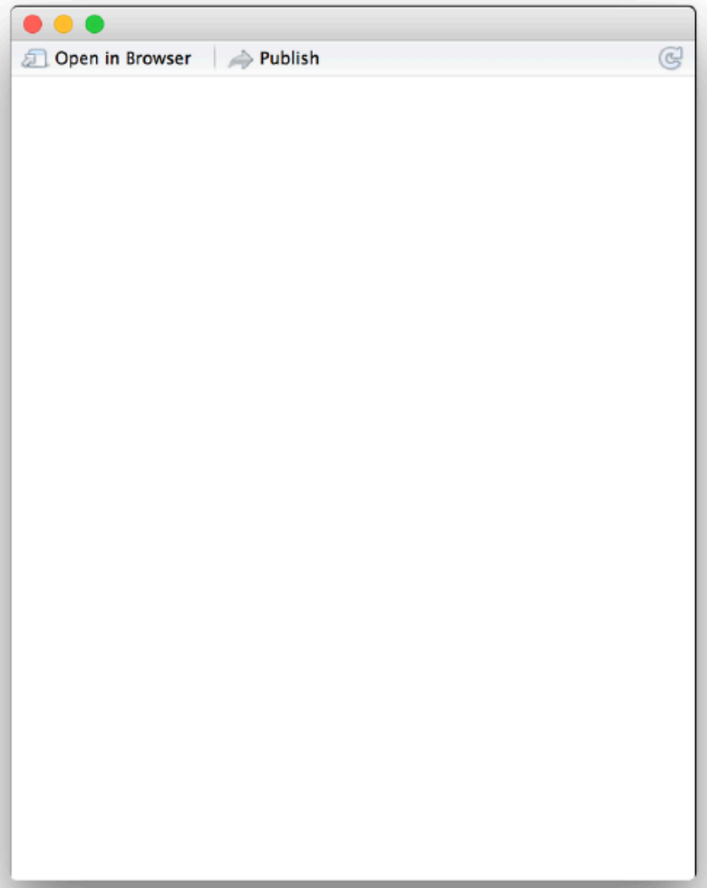
6 Exemples

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.1 Exemple : pas à pas

```
library(shiny)

ui <- fluidPage(



)

server <- function(input, output) {



}

shinyApp(ui = ui, server = server)
```

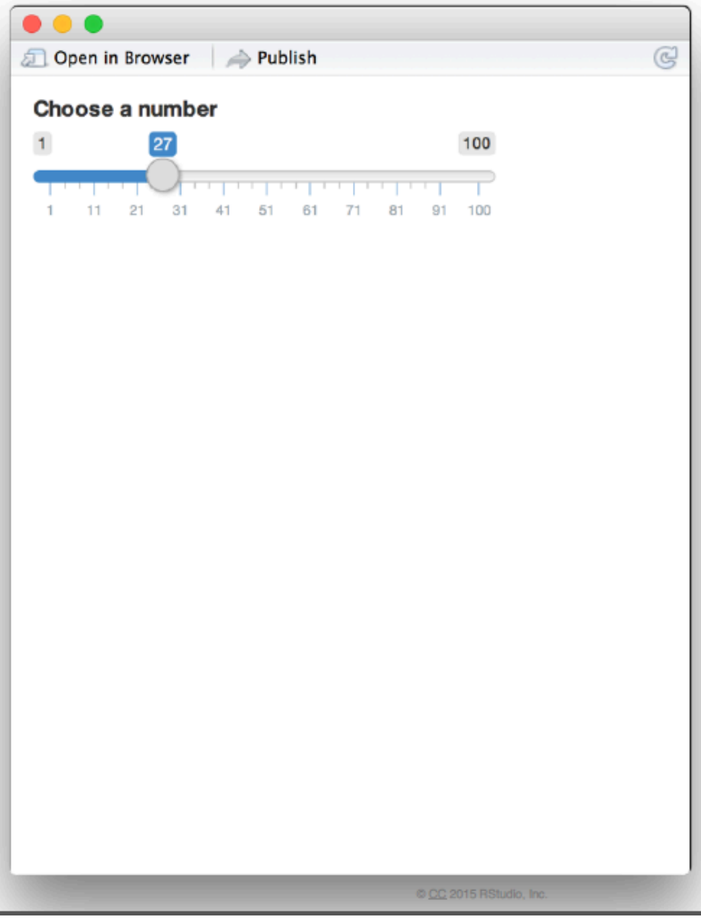


MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.2 Exemple : pas à pas

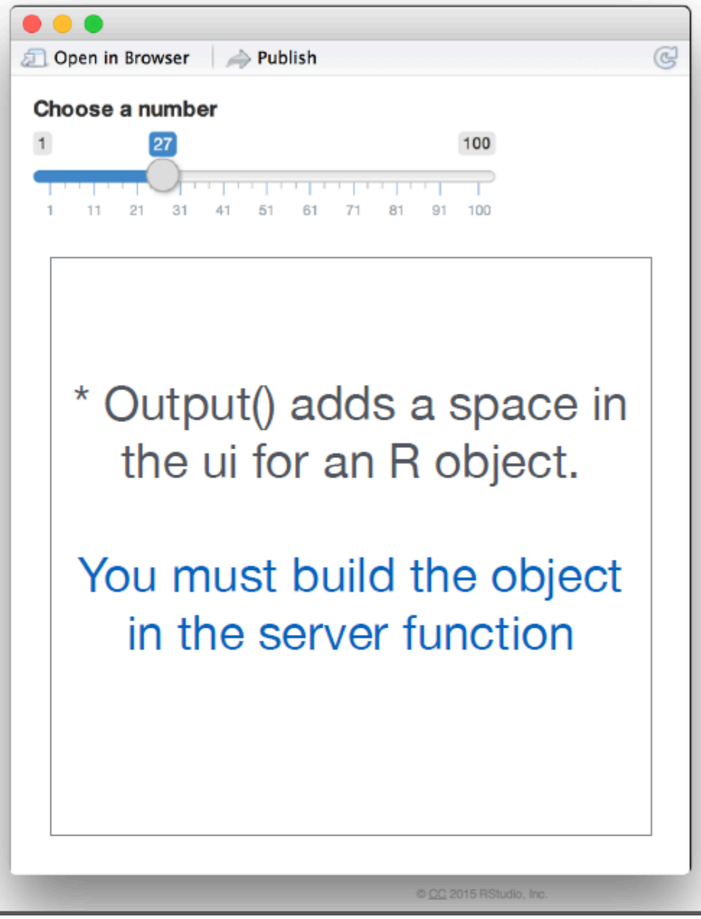
```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100)

)

server <- function(input, output) {



}

shinyApp(ui = ui, server = server)
```



MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

6.3 Exemple : pas à pas

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {



}

shinyApp(ui = ui, server = server)
```



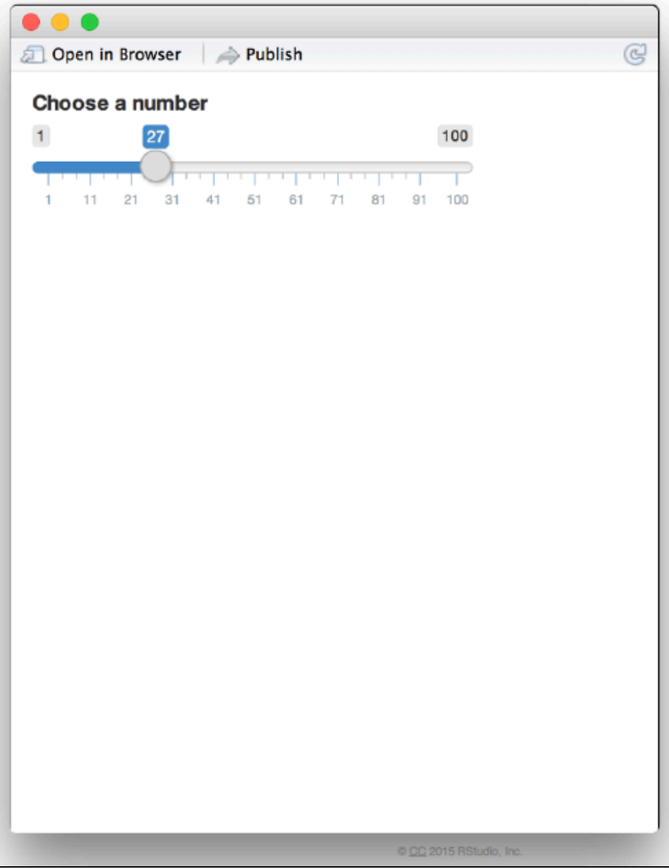
MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.4 Exemple : pas à pas

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <-


}

shinyApp(ui = ui, server = server)
```

1



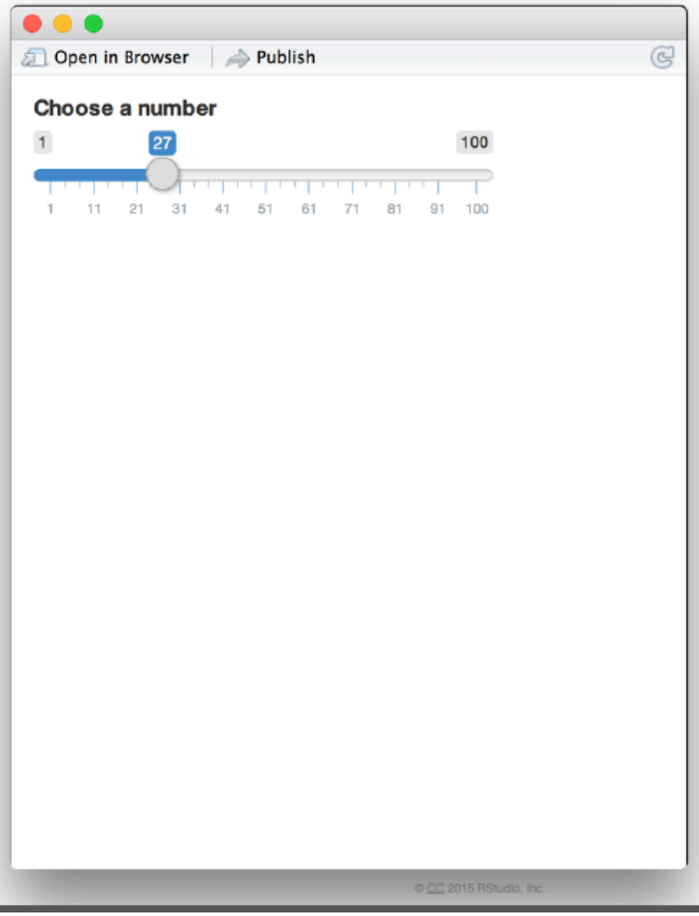
MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.5 Exemple : pas à pas

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({

  })
}

shinyApp(ui = ui, server = server)
```

2



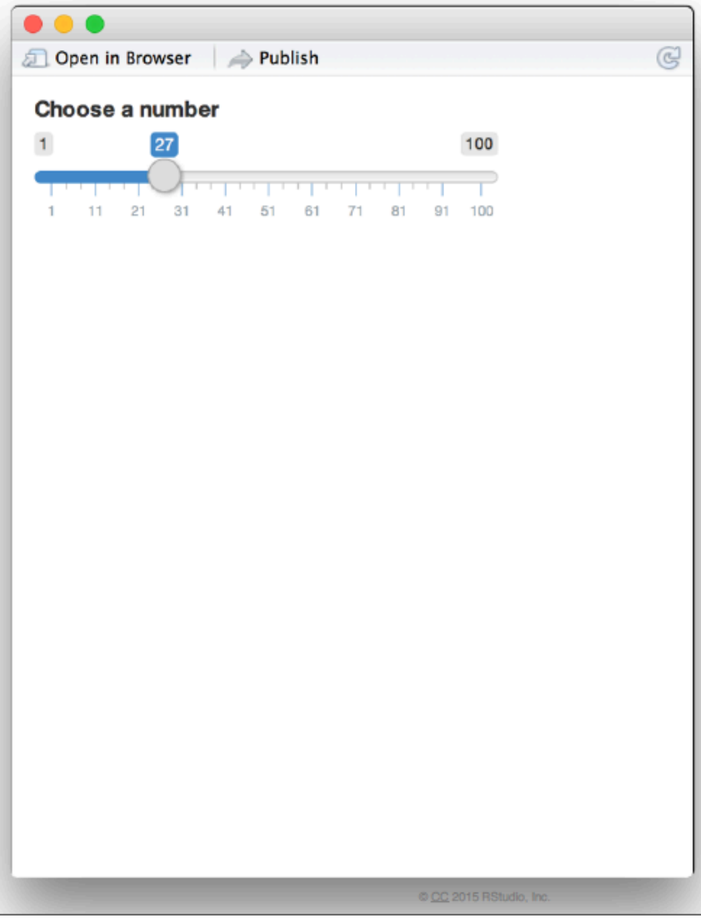
MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.6 Exemple : pas à pas

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

shinyApp(ui = ui, server = server)
```

3

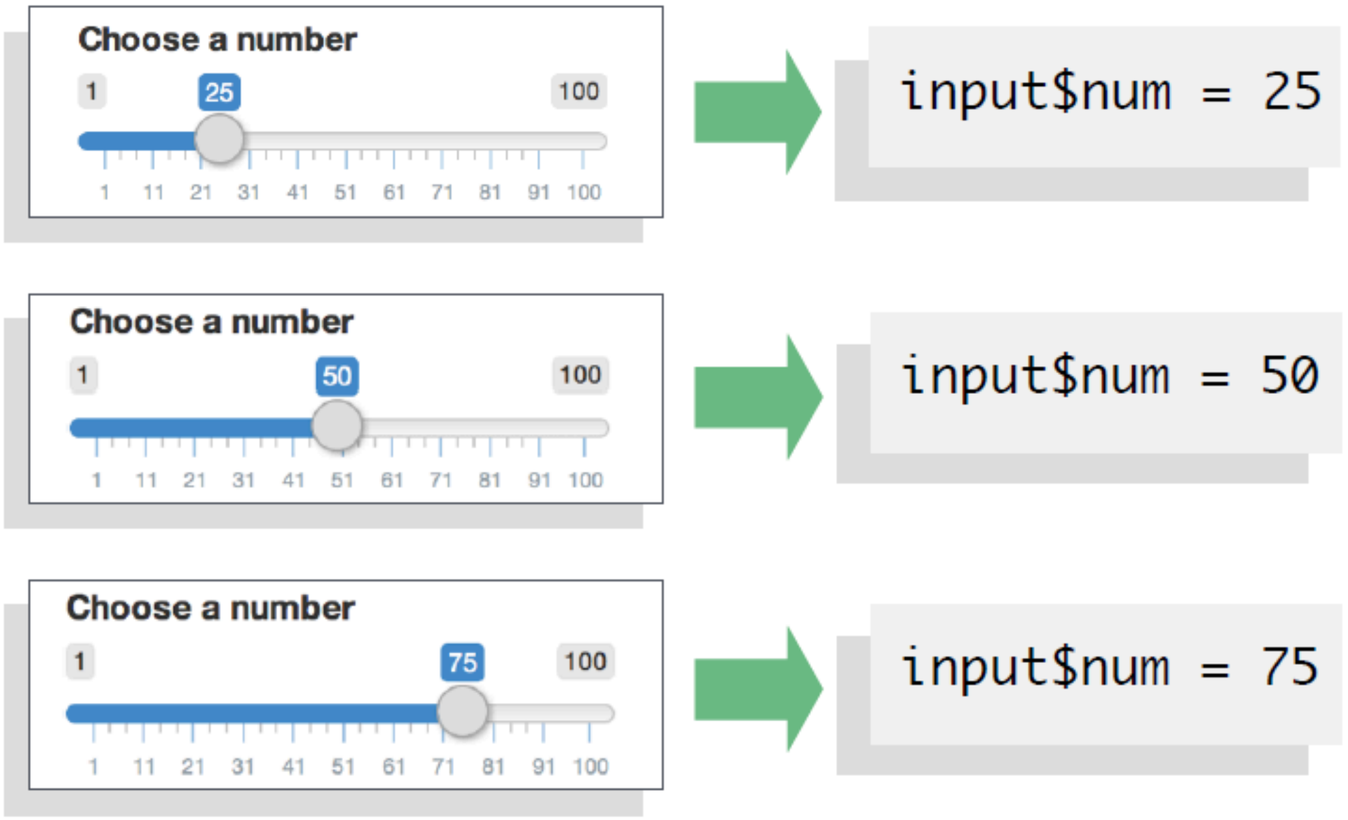


MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.7 Exemple : pas à pas

L'objet **input$num** sera automatiquement modifié :

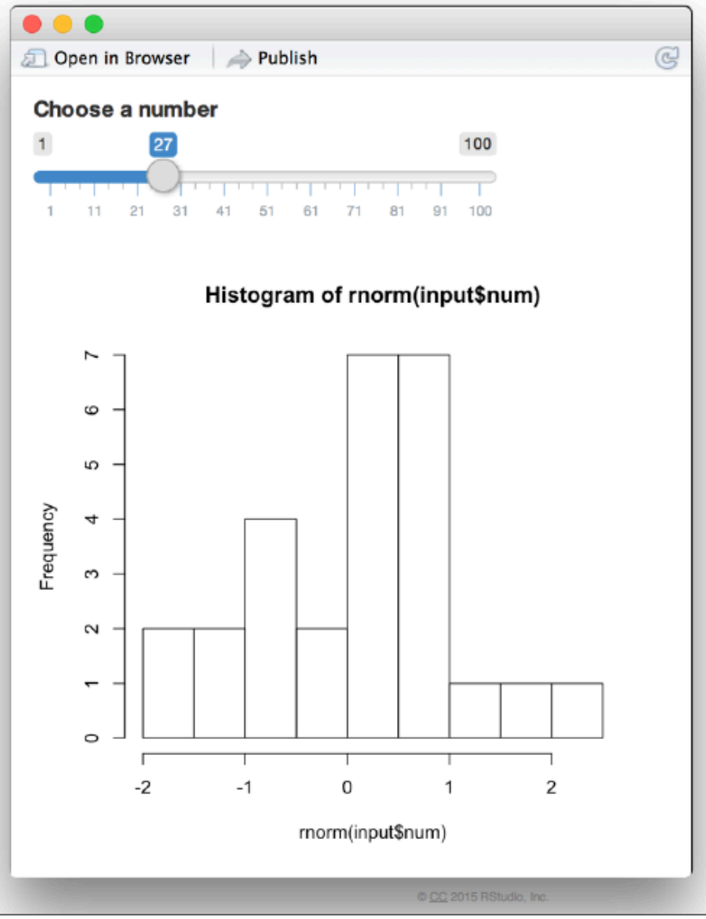


MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

6.8 Exemple : pas à pas

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

shinyApp(ui = ui, server = server)
```

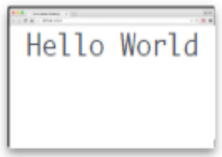


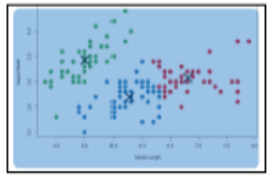
MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

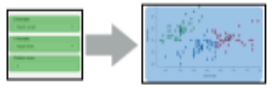
Liberté
Égalité
Fraternité

6.9 Exemple : pas à pas

```
library(shiny)
ui <- fluidPage()
server <- function(input, output) {}
shinyApp(ui = ui, server = server)
```







Dans le **fichier ui.R** :

Ajout d'éléments à afficher via la fonction fluidPage()

Argument en entrées modifiable par l'utilisateur (input)

Ajout de sortie dynamique via l'objet output

Le lien entre les inputs

et les outputs : server.R

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

6.10 Exemple : pas à pas

Dans le **fichier server.R** :

`output$hist <-`

Instanciation de l'objet output qui sera appelé dans l'interface

```
renderPlot({
  hist(rnorm(input$num))
})
```

Construction de l'objet à transmettre via la méthode render*()

`input$num`

Récupération du paramètre d'entrée modifié

Mise en place de l'interactivité : récupération du paramètre en entrée pour construire

la sortie et le stocker dans l'objet output

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

7 Réactivité

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

7.1 Réactivité

Avec Shiny, on peut déclencher des traitements en fonction des actions de l'utilisateur.

« La réactivité d'une application, c'est sa capacité à mettre à jour ses éléments suite à l'action d'un utilisateur. »

On utilise des *« observers »* :

- **observe** → déclenche un traitement quelle que soit l'action
- **observeEvent** → déclenche un traitement quand l'utilisateur modifie un input précis

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

7.2 Réactivité

- Partie UI :

```
selectInput(inputId="ma_liste",

label="Choisissez un pays",

choices=c("Cambodge", "Ecosse", "Corée du Sud", "Islande")

)
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

7.3 Réactivité

- Partie Server :

```
observeEvent(input$ma_liste,{

  pays <- input$ma_liste

  if pays %in% c("Ecosse", "Islande"){

    continent <- "Europe"

  } else{

    continent <- "Asie"

  }

}
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

8 Le programme complet

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

8.1 Le programme complet

- Pour ensuite lancer l'application définie, on utilise la fonction ShinyApp, en appelant les parties ui et server.

```
ui <- fluidPage( … )

server <- function(input, output, session) {
…
}

ShinyApp(ui, server)
```

- On lance ensuite le programme normalement.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

8.2 Le programme complet

On peut rajouter du CSS, du JavaScript, …

- Détails des fonctions : https://shiny.rstudio.com/reference/shiny/latest/
- Des possibilités infinies (voir cette galerie inspirante) : https://shiny.rstudio.com/gallery/

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

8.3 Exercice Server (1/3)

Exercice - partie Server

L'interface est prête, on va maintenant écrire les traitements réalisés par le programme, c'est-à-dire la partie serveur.
5) Dans la fonction server, créez un observeEvent qui se déclenche quand on clique sur le bouton.

6. Filtrez la table sur la région choisie via la liste déroulante (affectez le résultat dans une nouvelle table).

7. En utilisant summarise, créer une table contenant, par département, la moyenne, le maximum, le minimum et la somme des populations (Pop_mun_2011). Affectez cette table à l'outputId de l'interface pour l'afficher (en utilisant renderTable).

Exécutez le programme.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

8.4 Exercice Server (2/3)

Exercice - partie Server

8. A l'aide de conditions, créez la table sur la donnée choisie (Pop_mun_2011, Altitude ou Superficie).
   Exécutez le programme.

9. Dans l'observeEvent, remplacer l'inputId du bouton, par un vecteur contenant les inputId des 2 listes déroulantes.
   Exécutez le programme.

La mise à jour du tableau se fait maintenant dès qu'on change une valeur des listes déroulantes. Vous pouvez supprimer le bouton dans l'interface.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

8.5 Exercice Server (2/3)

Liste des régions :

Provence-Alpes-Côte d'Azur

Variables :

Population

DEP	LIBDEP	minimum	moyenne	somme	maximum
04	Alpes-de-Haute-Provence	2	804.79	160959	22316
05	Hautes-Alpes	10	805.84	138605	40654
06	Alpes-Maritimes	40	6633.40	1081244	344064
13	Bouches-du-Rhône	131	14745.49	1975896	140684
83	Var	11	6619.18	1012735	163974
84	Vaucluse	36	3620.07	546630	90194

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

9 Liens utiles

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

9.1 Liens utiles

- L'aide mémoire:

https://thinkr.fr/pdf/shiny-french-cheatsheet.pdf

- Le tutoriel (en anglais) disponible sur RStudio :

https://shiny.rstudio.com/tutorial/written-tutorial/lesson1/