

Utilisation de la bibliothèque Shiny & Esquisse pour la création d'applications web

Mame Thierno Ndiaye {*élève ingénieur statisticien économiste*}

Année 2022-2023

Présentation de R Shiny

Structure d'une application Shiny

Exécution d'une application

Création d'une interface d'utilisateur (UI)

Ajouter des widgets de contrôle

Affichage de la sortie réactive

Introduction sur shinydashboard:

Présentation de R Shiny

Présentation de R Shiny

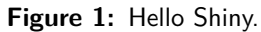
Shiny est un package utilisé pour créer des applications Web interactive avec R. L'application Web peut être intégrée dans un document rmarkdown, une page Web, de manière autonome ou en tant que tableau de bord.

C'est un outil puissant avec lequel nous pouvons créer des applications et des tableaux de bord assez rapidement.

Dans cet exposé, Nous allons parcourir les bases de Shiny, comment créer des applications simples et présenter le concept de réactivité.

Exemples d'application Shiny

L'emballage Shiny comporte onze exemples intégrés qui démontrent chacun comment fonctionne Shiny. Chaque exemple est une application Shiny autonome. L'exemple **Hello Shiny** trace un histogramme du jeu de données de R avec un nombre configurable de bacs. Les utilisateurs peuvent modifier le nombre de bacs avec une barre de défilement, et l'application répondra immédiatement à leur entrée. Nous utiliserons **Hello Shiny** pour explorer la structure d'une application Shiny et créer notre première application.



Structure d'une application Shiny

Structure d'une application Shiny

Les applications Shiny sont contenues dans un script unique appelé **app.R**. Le script **app.R** réside dans un répertoire (par exemple, **Exposé_Shiny**) et l'application peut être exécutée avec **runApp("Exposé_Shiny")**.

app.R comporte trois volets :

- un objet d'interface utilisateur (**ui**)
- une fonction serveur (**server**)
- un appel à la fonction **shinyApp**

L'objet **ui** contrôle la disposition et l'apparence de l'application. La fonction **server** contient les instructions dont l'ordinateur a besoin pour générer notre application. Enfin, la fonction **shinyApp** crée des objets d'application Shiny à partir d'une paire **UI/serveur** explicite.

Une fonctionnalité intéressante des applications à fichier unique est que vous pouvez copier et coller l'application entière dans la console R, ce qui facilite le partage rapide de code pour que d'autres puissent l'expérimenter. Par exemple, si vous copiez et collez le code ci-dessus dans la ligne de commande R, il démarrera une application Shiny.

Exécution d'une application

Exécution d'une application

Chaque application Shiny a la même structure : un fichier **app.R** qui contient **ui** et **server** . On peut créer une application Shiny en créant un nouveau répertoire et en enregistrant un fichier **app.R** à l'intérieur. Il est recommandé que chaque application vive dans son propre répertoire unique.

On peut exécuter une application Shiny en donnant le nom de son répertoire à la fonction **runApp**.

Remarque : **runApp** est similaire à **read.csv**, **read.table** , et à de nombreuses autres fonctions de R. Le premier argument de **runApp** est le chemin d'accès de votre répertoire de travail au répertoire de l'application.

Création d'une interface d'utilisateur (UI)

Disposition

Disposition

Shiny utilise la fonction **fluidPage** pour créer un affichage qui s'ajuste automatiquement aux dimensions de la fenêtre du navigateur de votre utilisateur. Nous disposons de l'interface utilisateur de notre application en plaçant des éléments dans la fonction **fluidPage**.

titlePanel et **sidebarLayout** sont les deux éléments les plus populaires à ajouter à **fluidPage**. Ils créent une application Shiny de base avec une barre latérale.

sidebarLayout prend toujours deux arguments :

- sidebarPanel**: Sortie de fonction
- mainPanel**: Sortie de fonction

Ces fonctions placent le contenu dans la barre latérale ou dans les panneaux principaux.



On peut également créer des mises en page plus avancées. On peut utiliser **navbarPage** pour doter l'application d'une interface utilisateur de plusieurs pages comprenant une barre de navigation. Ou on peut utiliser **fluidRow** et construire une mise en page **column** à partir d'un système de grille. Pour en savoir plus sur ces options avancées, vous pouvez utiliser le lien ci-dessous:

<https://shiny.rstudio.com/articles/layout-guide.html>

Contenu HTML

Contenu HTML

On peut ajouter du contenu à l'application Shiny en la plaçant dans une fonction **Panel**. Par exemple, les applications ci-dessus affichent une chaîne de caractères dans chacun de leurs panneaux.

Fonction Shiny	Crée
P	Un paragraphe de texte
h1	Un en-tête de premier niveau
h2	Un en-tête de deuxième niveau
h3	Un en-tête de troisième niveau
h4	Un en-tête de quatrième niveau
h5	Un en-tête de cinquième niveau
h6	Un en-tête de sixième niveau
a	Un lien hypertexte
br	Un saut de ligne

div	Une division de texte avec un style uniform
img	Une image
strong	Texte en gras
em	Texte en italique

Texte mis en forme

Texte mis en forme

Shiny offre de nombreuses fonctions de balise pour la mise en forme du texte. On peut utiliser par exemple les fonction précédentes pour faire des mises en forme.

Images

Images

Pour insérer une image, on donne à la fonction **img** le nom de notre fichier image comme argument.

Important: L'image doit d'abord être dans le dossier **WWW** de shiny.

Ajouter des widgets de contrôle

Ajouter des widgets de contrôle

Qu'est-ce qu'un widget ?: Élément Web avec lequel les utilisateurs peuvent interagir. Les widgets permettent aux utilisateurs d'envoyer des messages à l'application Shiny.

Les widgets Shiny collectent une valeur auprès de l'utilisateur. Lorsqu'un utilisateur modifie le widget, la valeur change également. Cela crée des opportunités que nous explorerons dans la suite.

Basic widgets

Buttons

Action

Submit

Single checkbox

☒ Choice A

Checkbox group

☒ Choice 1
☐ Choice 2
☐ Choice 3

Date input

2014-01-01

Date range

2017-06-21 to 2017-06-21

File input

Browse... No file selected

Help text

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

Numeric input

1

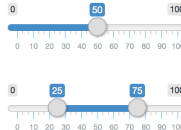
Radio buttons

☒ Choice 1
☐ Choice 2
☐ Choice 3

Select box

Choice 1

Sliders



Text input

Enter text...

Figure 2: widgets.

Widgets de contrôle

Widgets de contrôle

Les widgets Shiny standard sont :

fonction	widget
actionButton	Bouton d'action
checkboxGroupInput	Un groupe de cases à cocher
checkboxInput	Une seule case à cocher
dateInput	Un calendrier pour faciliter la sélection de la date
dateRangeInput	Une paire de calendriers pour sélectionner une plage de dates
fileInput	Un assistant de contrôle de téléchargement de fichiers
helpText	Texte d'aide pouvant être ajouté à un formulaire de saisie
numericInput	Un champ pour entrer des nombres
radioButtons	Un ensemble de boutons radio

selectInput	Une boîte avec des choix
sliderInput	Une barre de défilement
submitButton	Un bouton d'envoi
textInput	Un champ pour saisir du texte

Ajout de widgets

Ajout de widgets

Nous pouvons ajouter des widgets à notre page Web de la même manière qu'on a ajouté d'autres types de contenu HTML précédemment. Pour ajouter un widget à notre application, plaçons une fonction de widget dans **sidebarPanel** ou dans **mainPanel** votre objet **ui**.

Chaque fonction de widget nécessite plusieurs arguments. Les deux premiers arguments de chaque widget sont les suivants :

- **un nom pour le widget** : l'utilisateur ne verra pas ce nom, mais vous pouvez l'utiliser pour accéder à la valeur du **widget**. Le nom doit être une chaîne de caractères.
- **une étiquette** : cette **étiquette** apparaîtra avec le widget dans notre application. Il doit s'agir d'une chaîne de caractères, mais il peut s'agir d'une chaîne vide "".

Les arguments restants varient d'un widget à l'autre. Ils incluent des éléments tels que les **valeurs initiales**, les **plages** et les **incréments**.

Affichage de la sortie réactive

Affichage de la sortie réactive

Il est temps de donner à notre application Shiny une qualité « live »!
On verra comment créer une sortie réactive aux écrans de notre application Shiny.

Deux étapes

Deux étapes

On peut créer une sortie réactive avec un processus en deux étapes.

1. Ajouter un objet R à l'interface utilisateur.

2. Indiquez à Shiny comment construire l'objet dans la fonction **serveur**. L'objet sera réactif si le code qui le génère appelle une valeur de widget.

Étape 1 : Ajouter un objet R à l'interface utilisateur

Shiny fournit une famille de fonctions qui transforment les objets R en sortie pour votre interface utilisateur. Chaque fonction crée un type de sortie spécifique.

Fonction de sortie	Créé
dataTableOutput	DataTable
htmlOutput	HTML brut
imageOutput	image
plotOutput	complot
tableOutput	table
textOutput	SMS
uiOutput	HTML brut
verbatimTextOutput	SMS

Étape 2 : Fournir le code R pour générer l'objet

Étape 2 : Fournir le code R pour générer l'objet

Placer une fonction **output** dans **ui** indique à Shiny où afficher notre objet. Ensuite, on doit dire à Shiny comment construire l'objet.

Pour ce faire, on fournit le code R qui construit l'objet dans la fonction **server**.

La fonction **server** joue un rôle particulier dans le processus Shiny; il génère un objet de type liste nommé **output** qui contient tout le code nécessaire pour mettre à jour les objets R dans notre application. Chaque objet R doit avoir sa propre entrée dans la liste.

On peut créer une entrée en définissant un nouvel élément **output** pour dans la fonction **server**, comme ci-dessous. Le nom de l'élément doit correspondre au nom de l'élément réactif que vous avez créé dans le fichier **ui**.

Chaque entrée à doit contenir la sortie d'une des fonctions *output* de Shiny. Ces fonctions capturent une expression R et effectuent un léger prétraitement sur l'expression *render*. Utilisez la fonction *render* qui correspond au type d'objet réactif que vous créez.

Fonction de
rendu

Crée

renderDataTable DataTable

renderImage images (enregistrées en tant que lien vers un
fichier source)

renderPlot Emplacements

renderPrint toute sortie imprimée

renderTable Trame de données, matrice, autres structures de
type table

renderText chaînes de caractères

Chaque fonction **render** prend un seul argument : une expression R entourée d'accolades, “{ }” . L'expression peut être une simple ligne de texte, ou elle peut impliquer plusieurs lignes de code, comme s'il s'agissait d'un appel de fonction compliqué.

Introduction sur shinydashboard:

Introduction sur shinydashboard:

Grâce à la librairie shiny, et d'autres librairies R, comme **ggplot2** et **shinydashboard**, il est ainsi possible de générer des tableaux de bord, introduisant même des méthodes statistiques approfondies si besoin.

Note : installer les trois packages suivant dès maintenant : shiny, shinydashboard et tidyverse (dans lequel est inclus ggplot2 entre autres). Vous trouverez sous le lien ci-dessous une documentation plus détaillée sur les tableaux de bord Shiny. <https://fxjollois.github.io/cours-2020-2021/lp-sante--tdb/shiny.html>