

Rapport Probabilités

Shiny

Shiny est un package opensource de RStudio permettant la création d'application web à partir du langage R.

Ainsi, avec Shiny, il est possible de créer une application web nous permettant d'analyser des statistiques par rapport à une application ou bien de créer des simulations interactives.

Une application Shiny se compose de deux parties : la partie UI (interface utilisateur) et la partie serveur. L'interface utilisateur définit la présentation visuelle de l'application. Il s'agit de l'interface que l'utilisateur peut voir et avec laquelle interagir. La partie serveur gère la logique sous-jacente et les calculs. Cette collaboration aboutit à des applications interactives de manière simple et efficace.

Dans la partie UI, nous définissons la disposition des éléments visuels. Ces éléments peuvent être le résultat de calculs, tels que des graphiques ou des tableaux, ou des entrées permettant à l'utilisateur d'initier des actions via des boutons ou de modifier des éléments via le curseur. Ces entrées et sorties sont appelées ou créées via des fonctions spécifiques au package Shiny. Ces fonctions sont utilisées de manière similaire aux fonctions R classiques, mais dans un langage propre au développement web, comme HTML ou CSS.

Le Serveur quant à lui va transformer l'interaction utilisateur en action. Il réagit aux entrées de l'UI, effectue des calculs, manipule les données et met à jour l'interface en conséquence.

Notre Application

Dans le cadre de notre SAE, nous devons réaliser une application Shiny utilisant un jeu de données en rapport avec notre application web de ticketing.

Nous avons ainsi décidé de créer une application fournissant un diagramme représentant le nombre de tickets créés en fonctions du mois et de l'année.

Comment fonctionne notre application ?

Pour commencer, notre application a besoin de charger différents packages qui sont les packages Shiny, dplyr et ggplot2 utiles au bon fonctionnement de l'application.

- Le package shiny nous permettra donc d'utiliser les fonctions nécessaires à la création d'une application web interactive en R.
- Le package dplyr va permettre la manipulation et le filtrage des données des tickets
- Le package ggplot2 va lui permettre de générer des graphiques pour visualiser la répartition des tickets par mois.

Pour le package shiny, on utilise les fonctions suivantes :

- `fluidPage()` : Cette fonction crée une page Shiny avec une mise en page fluide qui s'adapte à différentes tailles d'écran. C'est la base de la mise en page de l'application Shiny.
- `titlePanel()` : Utilisée pour définir le titre de la page de l'application Shiny.
- `sidebarLayout()` : Il est utilisé pour créer une disposition à deux volets dans l'application Shiny, avec un volet latéral (`sidebarPanel()`) et un volet principal (`mainPanel()`).
- `sidebarPanel()` : Utilisé pour définir le contenu du volet latéral de l'application, tel que les entrées utilisateur, les boutons, etc.
- `selectInput()` : Cette fonction crée un menu déroulant (input) dans le volet latéral de l'application, permettant à l'utilisateur de sélectionner des options à partir d'une liste prédéfinie.
- `plotOutput()` : Utilisé pour définir l'emplacement où un graphique généré dynamiquement sera affiché dans le volet principal de l'application.
- `textOutput()` : Cette fonction est utilisée pour définir l'emplacement où du texte généré dynamiquement sera affiché dans le volet principal de l'application.
- `renderPlot()` : Utilisée dans la définition de la logique du serveur pour générer un graphique dynamique en fonction des entrées de l'utilisateur.
- `renderText()` : Utilisée dans la définition de la logique du serveur pour générer du texte dynamique en fonction des entrées de l'utilisateur.
- `shinyApp()` : Cette fonction est utilisée pour lancer l'application Shiny en spécifiant l'interface utilisateur (UI) et la logique du serveur.

Pour le package dplyr :

- `filter()` : Utilisé pour filtrer les lignes de données en fonction de certaines conditions. Par exemple, dans le code, il est utilisé pour filtrer les tickets en fonction de l'année et du mois sélectionnés par l'utilisateur.

- `group_by()` : Il est utilisé pour regrouper les données en fonction des variables spécifiées. Dans le code, il est utilisé pour regrouper les tickets par mois.
- `summarise()` : Cette fonction est utilisée pour résumer les données en calculant des statistiques agrégées, telles que le nombre total de tickets pour chaque mois dans le code.
- `n()` : Utilisé pour compter le nombre d'observations (dans ce cas, le nombre de tickets) dans chaque groupe.

Pour le package `ggplot2` :

- `ggplot()` : C'est la fonction principale pour créer des graphiques à l'aide de `ggplot2`. Elle initialise un nouveau graphique.
- `aes()` : Utilisé pour définir les mappings esthétiques (aesthetic mappings) qui relient les variables aux éléments visuels du graphique, tels que les axes, les couleurs, etc.
- `geom_bar()` : Utilisé pour créer un diagramme à barres. Dans le code, il est utilisé pour afficher le nombre de tickets par mois.
- `labs()` : Utilisé pour définir les étiquettes des axes, du titre et d'autres éléments du graphique.
- `theme_minimal()` : Utilisé pour définir le thème du graphique, dans ce cas, un thème minimaliste.

Ensuite, il y a une lecture du fichier CSV : Le fichier "tickets.csv" est lu et stocké dans un objet appelé "tickets". Puis, la colonne "date_creation" est convertie en classe "Date" à l'aide de la fonction `as.Date`.

Après cela, il y a une définition de l'interface utilisateur en utilisant la fonction `fluidPage`.

Elle comprend un titre, un panneau latéral avec des menus déroulants pour sélectionner une année et un mois, et un panneau principal avec un graphique et un texte.

Puis, la logique du serveur est définie à l'aide de la fonction `server`. Elle spécifie comment les sorties (plots, textes) doivent être générées en fonction des entrées de l'utilisateur.

Le graphique est généré en fonction des sélections de l'utilisateur et le nombre de tickets créés pour le mois sélectionné est affiché sous forme de texte si l'utilisateur choisit un mois spécifique.

Enfin, l'application est lancée en utilisant la fonction shinyApp avec l'interface utilisateur et la logique du serveur comme arguments.

Comment se présente notre application ?

Cette application possède un formulaire demandant de sélectionner l'année et le mois. L'année que l'on peut choisir dépend des tickets créés. En effet, seules les années de créations des tickets sont sélectionnables. Par exemple, si aucun ticket n'a été créé en 2019, alors l'option 2019 ne sera pas écrite.

Pour chaque année, nous pouvons sélectionner un mois. La sélection d'un mois fonctionne de la même manière que pour l'année. Cependant, dans la sélection du mois, il y a l'option tous.

Lors de l'ouverture de l'application, celle-ci s'ouvre avec l'année la plus ancienne et avec tous pour le mois. Quand l'option tous est sélectionnée, l'application affiche un diagramme représentant le nombre de tickets de l'année en fonction du mois. (Annexe 1).

Lorsque l'on sélectionne un mois, l'application affiche le nombre de tickets créer ce mois-ci. (Annexe 2)

Pour cette application, nous utilisons actuellement des tickets fictifs. Effectivement, nous avons créé un fichier csv contenant pour le moment seulement des dates représentant les dates de création des tickets. L'application va ainsi lire ce fichier csv et créer un tableau contenant toutes les dates.

Évolution possible.

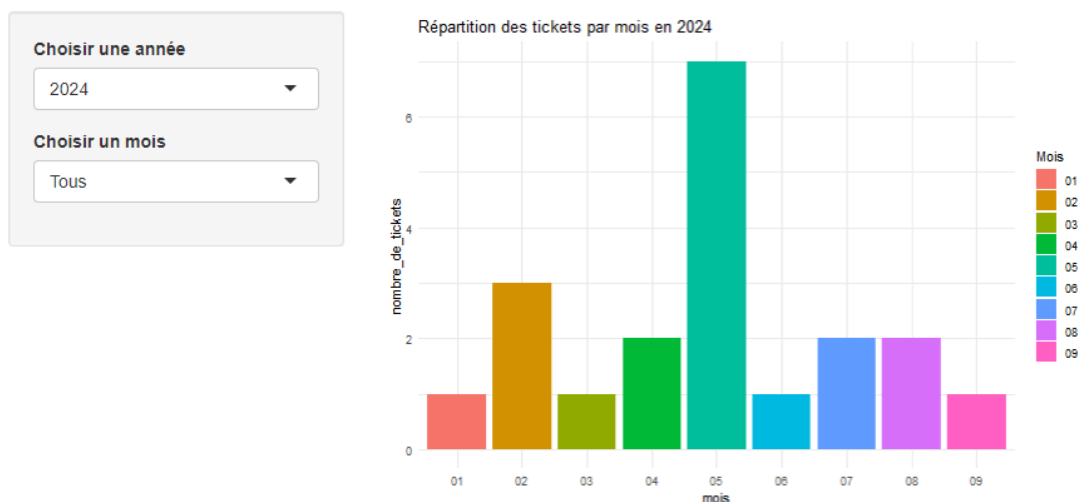
Comme mentionné précédemment, les tickets utilisés sont fictifs, cependant, nous pourrions par la suite récupérer les vrais tickets afin de les mettre dans le fichier csv pour avoir les vraies statistiques de l'application web.

De plus, lors de la sélection d'un mois, nous pourrions ensuite faire la liste des tickets du mois avec leur date complète et leurs titres.

Annexes

Annexe 1 : capture d'écran de l'application pour une année sélectionnée avec tous les mois

Analyse des tickets créés



Annexe 2 : capture d'écran de l'application pour un mois spécifique

Analyse des tickets créés

Choisir une année

2024

Choisir un mois

05

Nombre de tickets créés ce mois-ci : 7