

Analyse du nombre de visiteurs avec une application Shiny

CHOISY Alexis

DEGAT Teddy

GRONDIN David

MOUSSAMIH Elias

LEAUTHAUD Matthieu

Plan

- 1.Introduction
- 2.Description de l'application Shiny
 - 2.1.Stockage des données
 - 2.2.Interface Utilisateur (UI)
 - 2.3.Partie serveur fonctionnalités principales
- 3.Méthodologie
 - 3.1.Apprentissage de la bibliothèque R Shiny
 - 3.2.Les packages
- 4.Installation du server shiny ainsi que l'intégration de la l'application R
 - 4.1.Installation via Docker
- 5.Conclusion

1.Introduction

Dans le cadre de ce projet, nous avons développé une application Shiny visant a effectuer un jeux de données sur le nombre de visiteurs de notre site web de ticketing. L'application permet la visualisation des données, le calcul de statistiques et de probabilités telles que la moyenne, la médiane et l'écart type, ainsi que la possibilité d'ajouter, modifier ou supprimer des données. Toute ces fonctionnalité son accessible directement sur notre application uniquement par l'administrateur web dans un onglet prévue a cette effet.

2.Description de l'application Shiny

L'application est structurée en différentes sections :

2.1.Stockage des données

- Les données utilisées dans notre application Shiny sont stockées dans un fichier au format CSV, représentant les informations sur le nombre de visiteurs par mois et par année.
- Nous avons choisis de stockées nos données dans un fichier csv pour sa simplicité d'utilisation, de compréhension et de manipulation. En raison des données fictive nous n'avons pas utiliser de système de gestion de base de donnée(SGBD), ce qui nous simplifie le processus de stockage et de gestion des données.
- Lors de l'ajout, de la modification ou de la suppression de données via l'interface de l'application, les opérations de manipulation de données sont effectuées sur ce fichier CSV à l'aide des fonctionnalités de lecture `read.csv()` et d'écriture `write.table()` de R.

2.2.Interface Utilisateur (UI)

- L'interface est séparée en trois pages : la principale pour les statistiques, une autre pour le calcul de probabilité et enfin une partie qui concerne la modification des données.
- La section principale de l'interface utilisateur comporte des panneaux latéraux pour sélectionner la période de début et de fin pour l'analyse.
- Des options sont disponibles pour choisir les statistiques à calculer : moyenne, médiane, écart type.
- Un panneau principal affiche les résultats des calculs ainsi qu'un graphique représentant le nombre de visiteurs par mois pour la période sélectionnée.

2.3.Partie serveur fonctionnalités principales

- **Analyse des données** : Les données sont filtrées en fonction de la période sélectionnée et les statistiques choisies sont calculées et affichées.
- **Graphique interactif** : Un graphique représentatif du nombre de visiteurs par mois est généré. Des lignes pour les statistiques calculées peuvent être ajoutées au graphique.
- **Calcul de probabilité** : En fonction des données du fichier csv, l'admin peut étudier la probabilité du nombre de visiteurs en fonction de la moyenne.
- **Gestion des données** : La possibilité d'ajouter, de modifier ou de supprimer des données existantes est intégrée à l'application.

3.Méthodologie

3.1.Apprentissage de la bibliothèque R Shiny

Pour commencer ce projet, nous nous sommes initiés à la bibliothèque R Shiny en parcourant la documentation officielle, des tutoriels en ligne et des exemples de code. Cette phase d'apprentissage nous a permis de comprendre les principes fondamentaux de Shiny, y compris la structure de l'application, la création d'interfaces utilisateur réactives et la gestion des éléments de l'interface. Shiny nous permet donc de créer des applications web interactives en utilisant le langage de programmation R, tout en exploitant la puissance du HTML et du CSS.

3.2.Les packages

Pour parvenir à ces fonctionnalités, nous avons utilisé les packages `shiny`, `dplyr`, `ggplot2`, et `shinythemes`. Nous avons également créé des fonctions réactives pour effectuer les calculs et mettre à jour l'interface en fonction des interactions de l'utilisateur. Toutes ces bibliothèques ont été utilisées pour ajouter des éléments visuels à notre interface pour que l'utilisateur ait une meilleure interaction. Parmi ces éléments, on retrouve un graphique et aussi beaucoup de CSS.

4.Installation du serveur shiny ainsi que l'intégration de l'application R

Nous avons d'abord envisagé de mettre en place l'application R en installant le paquet officiel de `shiny-server` sur l'adresse https://download3.rstudio.org/ubuntu-18.04/x86_64/shiny-server-VERSION-arm64.deb mais il s'avère qu'il n'existe aucune version précompilée de celui-ci adaptée pour l'architecture de processeur `ARM64` soit l'architecture utilisée par le processeur d'un Raspberry PI. Nous avons donc essayé de compiler depuis la source nous-même sur le Github officiel de `shiny-server` mais nous avons aussi eu des erreurs lors de la compilation :

```
CMake Error at cmake_install.cmake:46 (file):  
  file INSTALL cannot find "/home/pisae/shiny-server/node_modules": No such  
  file or directory.  
  
make: *** [Makefile:120: install] Error 1
```

Alors nous avons finalement choisit de 'conteneuriser' le serveur shiny grâce à [Docker](#) grâce à cette [image](#) trouvée sur Dockerhub.

4.1.Installation via Docker

Pour télécharger l'image nous avons fait la commande suivante :

```
docker pull hvalev/shiny-server-arm
```

Puis d'après le guide d'installation nous avons exécutés les commandes suivante :

```
mkdir ~/shiny-server  
mkdir ~/shiny-server/logs  
mkdir ~/shiny-server/conf  
mkdir ~/shiny-server/apps  
git clone https://github.com/hvalev/shiny-server-arm-docker.git ~/shiny-  
server-arm-docker  
cp ~/shiny-server-arm-docker/shiny-server.conf ~/shiny-server/conf/shiny-  
server.conf  
cp ~/shiny-server-arm-docker/init.sh ~/shiny-server/conf/init.sh  
cp -r ~/shiny-server-arm-docker/hello/ ~/shiny-server/apps/  
rm -rf ~/shiny-server-arm-docker/
```

toujours d'après le guide pour run le container :

```
docker run -d -p 3838:3838 -v ~/shiny-server/apps:/srv/shiny-server/ -v  
~/shiny-server/logs:/var/log/shiny-server/ -v ~/shiny-  
server/conf:/etc/shiny-server/ --name shiny-server hvalev/shiny-server-  
arm:latest
```

en faisant la commande suivante nous pouvons voir que le container tourne bien :

```
docker ps -a
```

nous pouvons aussi aller à l'adresse <http://adresse:3838/hello> pour vérifier que le serveur fonctionne bien, dans notre cas c'est à l'adresse <http://raspb11.dinfo.iut-velizy.uvsq.fr/>.

Il nous manque maintenant les library R nécessaire au bon fonctionnement de notre application, pour cela la commande suivante nous permettra d'accéder au terminal de notre image docker :

```
docker exec -it <ID-container> /bin/sh
```

une fois dedans il suffit d'installer les librarys nécessaire :

```
R -e "install.packages('shinythemes','ggplot2','dplyr'  
repos='http://cran.rstudio.com/')"
```

il faut maintenant quitter le container et commit les changements effectués :

```
docker commit <ID-container> sae-shiny
```

le nom de notre image sera **sae-shiny**

et supprimer l'ancienne image :

```
docker rm -f <ID-container>
```

maintenant il suffit de run le nouveau container qui a été créé en précisant le nom de notre image :

```
docker run -d -p 3838:3838 -v ~/shiny-server/apps:/srv/shiny-server/ -v  
~/shiny-server/logs:/var/log/shiny-server/ -v ~/shiny-  
server/conf:/etc/shiny-server/ --name shiny-server shiny-sae
```

Dans **~/shiny-server/apps** on va créer un nouveau dossier avec notre application

```
mkdir ~/shiny-server/apps/sae/  
mv ~/app.R ~/shiny-server/apps/sae/  
mv ~/sae.csv ~/shiny-server/apps/sae/
```

L'application est maintenant accessible via l'adresse **http://adresse:3838/sae**, pour l'intégrer à notre site de ticketing nous avons juste mit un **<iframe>** qui contient l'adresse qui redirige vers l'application R.

5.Conclusion

Ce projet a permis de développer une application Shiny fonctionnelle pour analyser le nombre de visiteurs d'un site web. L'interface utilisateur intuitive et les fonctionnalités de gestion des données offrent une expérience pratique et interactive pour explorer et comprendre les données relatives aux visites du site.

