

Documentation Technique - Projet Velov

Introduction

Cette documentation technique fournit des informations détaillées sur le projet Velov, un tableau de bord interactif développé en utilisant R et la bibliothèque Shiny. Le projet utilise des données Velov (vélos en libre-service) pour afficher des informations utiles sur les stations Velov dans la région de Lyon, France.

Fichiers du Projet

Le projet Velov est composé des fichiers suivants :

1. **Projet-SD2-Velov.R** - Le code principal du projet est implémenté dans ce fichier R. Il contient la logique métier, l'interface utilisateur et les gestionnaires d'événements pour le tableau de bord Velov.
2. **Code_Postaux.csv** - Ce fichier CSV a été créé à partir d'un géocodage tidygéocoder des données Velov grâce aux positions (lat-long). Il est utilisé pour stocker des informations sur les codes postaux liés aux stations Velov. Il est chargé dans le projet pour effectuer des jointures de données et pour améliorer la convivialité du tableau de bord.
3. **style.css** - Le fichier CSS a été créé par nous-mêmes pour personnaliser l'apparence du tableau de bord Shiny. Il est lié au code R pour appliquer des styles spécifiques à l'interface utilisateur.

Packages et Bibliothèques

Le projet Velov utilise une variété de packages et de bibliothèques R pour ses fonctionnalités. Voici la liste des packages et leur utilisation :

1. **Shiny** - Cette bibliothèque est au cœur du projet Velov, permettant de créer une application Web interactive en R.
2. **leaflet** - Utilisé pour afficher des cartes interactives avec des marqueurs pour les stations Velov.

3. **plotly** - Pour générer des graphiques interactifs basés sur les données Velov.
4. **DT** - Utilisé pour afficher des tables de données interactives dans le tableau de bord.
5. **dplyr et tidyverse** - Ces packages sont utilisés pour manipuler et traiter les données Velov de manière efficace.
6. **shinydashboard** - Utilisé pour créer la structure de base du tableau de bord, y compris le menu et les onglets.
7. **kableExtra** - Pour personnaliser la sortie des tableaux dans le tableau de bord.
8. **orca** - Permet de générer des images statiques à partir de graphiques Plotly.
9. **webshot** - Utilisé pour capturer des images statiques de graphiques interactifs.
10. **jsonlite et RMySQL** - Pour extraire des données Velov à partir d'une source de données externe (API ou base de données).

Source de Données

Les données utilisées dans le projet Velov proviennent de plusieurs sources :

1. **Données Velov via une API** : Le code R récupère les données de Velov à l'aide d'une API appartenant à Velov. Ces données comprennent des informations sur les stations Velov, telles que leur nom, leur capacité, le nombre de vélos disponibles, les coordonnées géographiques, etc.
2. **Fichier CSV Code_Postaux** : Ce fichier a été généré à partir d'un géocodage tidygéocoder des données Velov grâce aux positions (lat-long). Il est utilisé pour enrichir les données Velov avec des informations sur les codes postaux des stations.

Traitement des Données

Les données Velov sont soumises à un traitement avant d'être utilisées dans le tableau de bord. Le traitement des données comprend :

- Extraction des données Velov à partir de l'API Velov à l'aide du package jsonlite.
- Jointure gauche des données Velov avec le fichier CSV Code_Postaux.csv pour ajouter des informations sur les codes postaux.
- Sélection des colonnes pertinentes pour l'affichage dans le tableau de bord.

- Préparation des données pour les graphiques interactifs et les tables.

Les données sont traitées de manière efficace à l'aide des packages dplyr et tidyverse pour répondre aux besoins du tableau de bord Velov.

Fonctionnalités Principales

Le tableau de bord Velov comprend les fonctionnalités suivantes :

- Affichage de la carte interactive avec des marqueurs pour les stations Velov.
- Possibilité de filtrer les stations par code postal.
- Affichage de graphiques interactifs montrant la distribution des vélos par code postal.
- Affichage de graphiques en camembert montrant la répartition des vélos disponibles.
- Exportation des données filtrées et de l'ensemble des données Velov en format CSV.

Personnalisation

Les développeurs peuvent personnaliser le tableau de bord Velov en modifiant le fichier style.css. Ils peuvent ajuster les styles, les couleurs et les polices pour correspondre à leurs préférences.

Structure de l'Interface

L'interface utilisateur du projet Velov est organisée en plusieurs onglets pour permettre à l'utilisateur de naviguer entre différentes vues. La structure de base de l'interface est construite avec shinydashboard, offrant une barre latérale de menu et une zone centrale pour le contenu spécifique à chaque onglet.

Composants UI

1. **Onglet Stations Proches** : Cet onglet permet à l'utilisateur de rechercher des stations Velov à proximité en filtrant par code postal. Les composants UI comprennent un menu déroulant pour choisir la commune, un filtre pour le code postal, un tableau interactif, et une carte affichant les stations Velov correspondantes.

2. **Onglet Statistiques Graphiques** : Cet onglet affiche des statistiques graphiques basées sur les données Velov. Les composants UI comprennent des graphiques interactifs, une case à cocher pour filtrer les données, et un bouton pour télécharger les graphiques.

Malheureusement, le bouton pour télécharger les graphiques ne fonctionne pas.

3. **Onglet Base de Données** : Cet onglet affiche une table contenant l'ensemble des données Velov, et il permet à l'utilisateur d'exporter les données filtrées ou complètes en format CSV.

Malheureusement, les boutons pour télécharger en CSV ne fonctionnent pas.

Serveur (Server)

Fonctions du Serveur

Le serveur du projet Velov est responsable de gérer les interactions entre l'interface utilisateur et les données. Voici les principales fonctions du serveur :

1. **updatedVelovData** : Cette fonction est déclenchée par l'utilisateur lorsqu'il appuie sur le bouton "Rafraîchir". Elle récupère les dernières données Velov en utilisant une requête vers l'API Velov.
2. **filtered_data** : Cette fonction filtre les données Velov en fonction du code postal sélectionné par l'utilisateur. Elle gère également la mise à jour du slider pour ajuster le nombre de lignes à afficher dans la DataTable.
3. **downloadFilteredCSV** : Cette fonction permet à l'utilisateur de télécharger les données filtrées (CSV) lorsque le bouton "Exporter en CSV data filtrée" est déclenché.
Malheureusement elle ne fonctionne pas et donc les boutons d'export non plus.
4. **downloadAllCSV** : Cette fonction permet à l'utilisateur de télécharger l'ensemble des données Velov (CSV) lorsque le bouton "Exporter en CSV data complète" est déclenché.

Réactivité

La réactivité dans le projet Velov est gérée en utilisant la bibliothèque Shiny. Les fonctions du serveur sont activées en réponse à des événements spécifiques, tels que les clics sur les boutons, les sélections de menu, etc. La réactivité est essentielle pour mettre à jour les composants UI en temps réel en fonction des actions de l'utilisateur.

Par exemple, lorsque l'utilisateur clique sur le bouton "Rafraîchir", la fonction `updatedVelovData` est déclenchée pour récupérer les dernières données Velov et mettre à jour

la carte et la table dans l'onglet "Stations Proches". La réactivité est également gérée pour mettre à jour les données filtrées en fonction des sélections de l'utilisateur, pour ajuster les graphiques en fonction des filtres, et pour mettre à jour le nombre de lignes dans les DataTables.