Big Data Tools II
M1 IREF

Laurent R. Bergé*
Projet 2023-2024

Date de rendu max Webscraping :	Jeudi 1er février 2024, 23h59
Date de rendu max Shiny :	Mardi 6 février 2024, 23h59
Date de présentation :	Jeudi 8 février 2024

Résumé. L'objectif de ce projet est de créer une application de recherche d'emploi. Cette application offrira une interface intuitive à l'utilisateur pour lui proposer des offres d'emploi adaptées à son profil. Les offres d'emploi proposées seront tirées de sites webs existants proposant ces offres.

Instructions :

- Il faut rendre les projets avant les dates butoir définies. Il y a 0.5 point de pénalité par 12h de retard. Premières 12h gratuites.
- le code doit être commenté afin d'être compréhensible par tous
- la qualité du code compte : formattage impeccable, clarté dans ce qui est fait et consistence des noms de variable. Un code illisible donnera des points de pénalité. Un code très clair et facile à lire donnera des points bonus.
- le rendu des projets se fait exclusivement sur le site Moodle du cours : M2 IREF – Data analysis II

1. Général

Ce projet se fait en deux parties :

1. webscraping
2. shiny

Dans la première partie vous collecterez les données et les formatterez correctement en une base de données.

Dans la deuxième partie, vous créerez une application pour exploiter ces données.

Les attendus de ces deux parties et la façon dont elles sont notées sont très différentes. Prêtez bien attention aux instructions.

*B$_x$SE, UMR CNRS 6060, Université de Bordeaux, *email* : laurent.berge@u-bordeaux.fr

2. Webscraping

L'objectif est de recueillir des données sur les offres d'emploi dans le domaine de la "*science des données*"[2] en provenance de différents sites webs les recensant.

2.1. Instructions générales

date

- vous devrez rendre le code et les données de la partie **webscraping** avant le Jeudi 1er février 2024, 23h59

stratégie

- vous devrez rendre un fichier PDF qui synthétise les sites choisis et les résultats obtenus
- vous donnerez les stratégies établies pour extraire et formatter les données de façon robuste
- éventuellement vous pouvez ajouter une section détaillant les difficultés rencontrées

code

- vous rendrez autant de fichiers **`.R`** ou **`.py`** que nécessaires pour répliquer l'analyse. Il faut que **tout** le code utilisé soit contenu dans ces fichiers. Les fichiers ne contenant que des fonctions ou des classes (python) devront être préfixés par **`src_`**. Le ou les fichiers principaux à lancer pour répliquer l'analyse doivent se nommer **`main.R`**/**`main.py`**, ou être préfixés par **`main_`**.[3] Le code pour inclure tous les fichiers sources doit être présent dans les fichier **`main`**. Chaque fichier **`main`** ne doit pas dépendre de variables ou de fonctions définies dans d'autres fichiers **`main`**. Chaque fichier **`main`** doit pouvoir être lancé indépendemment des autres et fonctionner sur mon ordinateur.
- modulo l'installation de packages : les fichiers **`main`** doivent pouvoir tourner sans erreur sur ma machine. (Bien sûr ce n'est pas grave si la seule raison pour laquelle ils ne fonctionnent pas est l'absence de fichiers intermédiaires, comme le HTML des sites téléchargés par exemple.)
- dans le cas où le scraping requière que vous vous connectiez au site avec des identifiants, ne les mettez pas en clair dans le code. Mettez les identifiants dans un fichier, par exemple **`credentials.py`** ou **`credentials.R`**, que vous chargerez et utiliserez ensuite dans le code. Vous expliquerez en commentaire dans les fichiers **`main`** comment remplir ces fichiers pour que le code marche.
- chaque fichier devra contenir les premières lignes suivantes :

```
# Projet : Webscraping
# Objet : _BUT DE CE DOCUMENT_
# Auteurs :
# - Nom1, Prénom1
# - Nom2, Prénom2
# - Nom3, Prénom3
# - Nom4, Prénom4
```

données

- vous devrez rendre l'ensemble de la base de données finale en format **`.tsv`** ou **`.csv`**. Si la base de données est trop grosse : vous tronquerez[4] la variable de description du poste (c'est celle qui prendra beaucoup de place).

[2] Vous êtes libre de faire les recherches que vous voulez du moment que cela reste dans le domaine.

[3] Vous pouvez séparer le **`main`** de sorte à représenter des étapes. Cela peut être un **`main`** par site web par exemple. Vous pouvez également séparer le scraping du formattage. Dans le cas de plusieurs **`main`**, numérotez les dans leur noms pour clarifier leur ordre. Par exemple : **`01_main_linkedin_scrape.R`**, **`02_main_linkedin_format.R`**.

- ne déposez pas les données intermédiaires, seulement les données finales

2.2. Acquisition de données d'offre d'emploi

Ici l'objectif est de faire des recherches en ligne automatisées et de recueillir et structurer les informations sur les offres d'emploi. Les recherches se porteront sur la science des données (data scientist, big data, etc.).

Les sites d'offres d'emploi que vous chercherez peuvent être les suivants :
- linkedin
- indeed
- jobteaser
- hellowork
- tout autre site approprié

2.3. Objectifs

L'objectif est de créer une base de donnée avec les informations suivantes :

- recherche effectutée qui a amené à cette offre (quel mots clefs ?)
- intitulé du poste
- description du poste
- fourchette de salaire (si disponible)
- entreprise
- secteur de l'entreprise (si disponible)
- compétences demandées
- lieu d'exercice
- type d'emploi : CDI, CDD, stage
- durée de l'emploi (si stage ou CDD)
- site source de l'annonce (chaine de caractère)
- lien vers la page de l'annonce

Vous êtes libre d'ajouter plus d'informations à la liste ci-dessus.

Vous êtes également libre d'y apparier des données externes (comme des informations de la base SIRENE par exemple).

2.4. Notation

Il faudra scraper au moins deux sites mentionnés en Section 2.2. L'objectif est d'avoir au moins 1000 offres d'emploi.

- 10 points par site scrapé (oui, si 3 sites : 30 points) si les données finales obtenues sont correctes et toutes les variables demandées sont fournies (malus pour variables manquantes ou incorrectes)
- **pour chaque site**, extraire au moins 500 offres. Si en dessous de 500 : chaque offre vaut 0.02 point. Au dessus, chaque offre vaut 0.004, max 2 points bonus pour un total de 1000 offres. Je me réserve le droit de demander des preuves matérielles pour l'origine de chaque offre, donc conservez les données intermédiaires (le HTML scrapé).
- **pour chaque site**, la qualité du code pour formatter les données, ainsi que celle pour scraper, sont notées sous forme de bonus/malus (jusqu'à −3 en malus et +2 en bonus)

[4] Attention : tronquer ne veut pas dire supprimer !

3. Shiny

Cette partie du projet porte sur l'exploitation des données obtenues dans le travail précédent.

Vous créerez une application qui permettra de rapidement accéder aux offres d'emploi en fonction de mots clefs. Cette partie est très libre : essayez de faire une application la plus intéressante possible, qui donne le plus possible de valeur ajoutée à l'aggrégation des offres d'emploi.

3.1. Instructions

dates

- vous devrez rendre le code de l'application **shiny** avant le Mardi 6 février 2024, 23h59
- vous ferez une démonstration de votre application devant l'ensemble de la classe le Jeudi 8 février 2024

code

- vous rendrez une seule archive **`.zip`** qui contiendra tous les fichers nécessaires à faire tourner votre application, ceci incluant la base de données
- separate the application into a **`ui.R`** and a **`server.R`** file. Like we've done in class. Exception : if you think there is value added to pass arguments to the app, you can write an app-function, but it has to be justified.
- on the data : put it in the **`www/`** folder and access it within the server with, e.g., **`fread("www/my-data.csv)"`**
- l'application devra pouvoir tourner sur ma machine sans manipulation de ma part (à l'exception de l'installation de packages)
- chaque fichier devra contenir, en commentaire de début de fichier, le nom de chaque participant et le but du fichier (voir Section 2.2)

3.2. Notation

3.2.1. Contenu

Le contenu de l'application est libre. Néanmoins voici quelques éléments qu'elle se **doit** de **contenir** :

1. welcome page : page qui décrit somairement comment utiliser l'app (à noter qu'il y a des applications qui permettent de créer des "pas à pas" facilement en shiny)
2. page sur les données d'offre d'emploi avec des options pour facilement les filtrer/trier et offrir différentes façons de les afficher.[5]
3. une page où l'utilisateur peut écrire librement ses compétences (ou quelque chose du genre), et lorsqu'il valide apparaît les 3 offres qui lui correspondent le plus (par exemple sous la forme de cartes descriptives – peut être aussi sous la forme d'une base de données)

Voici d'autres idées que vous pouvez inclure :

- une page avec la géolocalisation des offres d'emploi, possiblement filtrées.

Astuce

Utilisez le package **`leaflet`** pour tout ce qui est cartographie web.

- une page où l'utilisateur peut uploader son CV en PDF. L'application lui renverra alors les meilleures offres correspondant à son CV (similaire au point 2 au dessus mais avec un input différent)

[5] A noter que pour chaque offre d'emploi, il faudra donner le nombre de sites dans lequel elle apparaît.

Astuce

Utilisez le package **`pdftools`** pour manipuler des documents PDF.

- dans l'affichage de la base de donnée, créer pour chaque ligne un bouton qui, une fois cliqué, affiche la fiche de poste formattée en fenêtre pop-up. Une version plus difficile est de faire la même chose mais sans bouton, juste quand l'utilisateur clique sur une ligne.

Astuce

Utilisez le package **`DT`** pour **`dataTableOutput`**/**`renderDataTable`** et utilisez l'argument **`escape = FALSE`** dans **`renderDataTable`**.

- tout autre idée que vous pouvez avoir et qui est pertinente

3.2.2. Esthétique

A noter que dans le cadre de cette partie l'aspect esthétique et l'expérience utilisateur est primordiale. Ne priviligiez pas la technique au détriment de l'UX et de l'esthétique.

Dans le contenu mentionné au dessus, sur les trois permiers points obligatoires, ça n'a l'air de rien mais il y a mille façons de faire : des façons très léchées et des façons bof bof. Je peux tout à fait mettre 20 à des personnes qui n'ont fait que ces trois points mais extrêmement bien. Faire les mêmes trois points sans réfléchir ne donnera pas plus de 5/20.

L'objectif ici est que vous soyez créatifs. Profitez d'être en groupe pour être très critiques vis à vis de votre travail.

3.2.3. Autres aspects

La qualité du code sera inspectée. Il y aura des bonus/malus (-3/+2) de la même façon que pour la partie webscrapping.

4. Mots de la fin

C'est un projet ambitieux, mais c'est vous qui avez choisi ! 😉