Webscraping sur R et Python

Laurent Bergé*

7 février 2022

1 Problème

Votre patron cherche à recruter quelqu'un et souhaiterais collecter les informations publiques le concernant, c'est à dire tout ce que la personne a pu laisser d'elle sur internet. Il vous contacte pour que vous collectiez ces informations. Mais par souci de confidentialité, il ne souhaite pas vous révéler le nom de la personne à étudier. Il vous charge alors de créer un bot qui collectera automatiquement les informations publiques pour cette personne et éditera automatiquement un rapport. A vous de créer une fonction efficace et facile d'utilisation pour votre patron.

*BxSE, University of Bordeaux, laurent.berge@u-bordeaux.fr.

2 Outils

— Pour ce projet, utilisez R ou Python, comme vous le souhaitez. Il n'y a pas forcément besoin de gérer des pages dynamiques, mais ça peut être utile également.

3 Résultats attendus

— un fichier nommé `source.R` ou `source.py`, contenant :
  — une fonction `scrape_public`
    — cette fonction recherche une personne en ligne et **écrit un rapport** sur cette recherche (voir Contenu du rapport)
    — cette fonction doit avoir au moins les trois arguments suivants : [1]
      1. `nom` : le nom de la personne
      2. `prenom` : son prénom
      3. `out` : chemin d'accès où sera écrit le rapport
  — tout le code source (et donc potentiellement toutes les sous-fonctions) permettant à la fonction `scrape_public` de tourner. **Cette fonction devra tourner sur mon ordinateur !**
— un fichier nommé `example.R` ou `example.py`, contenant :
  — un exemple d'appel de la fonction `scrape_public`. A part le chemin (dans `out`), cet exemple devrait fonctionner sur mon ordinateur.
— un exemple de rapport rendu avec `scrape_public` que vous aurez fait tourné sur vous ou un de vos collègues de la promo (si possible avec des résultats !)

4 Contenu du rapport

Le rapport doit avoir la forme d'une page HTML. Cette page doit contenir (si possible, voir Notation) les sections suivantes :

Google Résultats d'une recherche Google (ou équivalent). Seuls les **résultats pertinents** doivent apparaître.

1. Elle peut en avoir plus si vous trouvez cela utile.

Facebook Résultat d'une recherche de cette personne sur Facebook : nom de profil, informations publiques disponibles (âge, statut, etc), quelques posts (max 2).

Twitter Nom de profil, description contenue dans le profil de la personne et quelques tweets (max 2).

LinkedIn Informations du profil (description, études, travail s'il y en a).

Autres Quelqu'autre site qui vous semble oppurtun d'étudier (Instagram, etc).

Si possible, les liens vers les pages devront êtres inclus dans les résultats. Pour vous donner une idée de ce qui est attendu, et pour vous aider, vous pouvez utilisez le template du rapport. Le template n'est qu'un exemple, vous pouvez faire ce que vous voulez du moment où vous votre fonction `scrape_public` écrit un rapport en HTML.

A noter que vous avez une **obligation de moyen**, pas une obligation de résultat. Cela veut dire que vous devez mettre en oeuvre une stratégie raisonnable pour retrouver une personne en ligne. Mais si votre algorithme ne trouve pas la personne alors qu'elle a effectivement une présence en ligne, ce n'est pas grave.

5 Organisation

— La date de rendu est le **2 mars 2022**. Passé cette date, une pénalité de 1 point par jour de retard est appliquée.

— Rendre les devoirs sur Moodle (M2 IREF – Data analysis II).

6 Notation

10+ pts	Scraping	*Code pour scraper. Google (ou eq.) + 2 réseaux sociaux de votre choix valent 10 points. Ensuite chaque source supplémentaire vaut 3 point.*
3 pts	Qualité du rapport	*Pertinence du contenu du rapport et aspect esthétique*
4 pts	Modularité du code	*Si le code est écrit en fonctions, et ces fonctions suffisamment générales*
3 pts	Clarté et qualité du code	