

Le marché des chaussures de course à pied d'occasion

Emilien HARTMANN et Brice KUBLER

28-12-2024

I. Contextualisation

- ▶ En 2024, 12,4 millions de Français déclarent courir, ce qui représente 25 % de la population.
- ▶ La hausse est intégralement portée par les marcheurs ou coureurs réguliers : + 7 points, de 25 % à 32 %.
- ▶ Exemple : Une Nike Alphafly a 310 € ou encore une Asics Gel-Nimbus a 200 €

==> **Les chaussures de running n'ont jamais été aussi chères.**

- ▶ Paradoxalement : Les chaussures sont des biens rapidement usables et fabriquées avec des matières plastiques.
- ▶ Solution économique et écologique ? → Site de revente : Vinted, eBay.

II. Problématique

Comment être alerté en temps réel par des annonces pertinentes postées par des particuliers ?

III. Construction du code : Sélection des packages

► Sélection des bibliothèques et packages pertinents :

1. `library(blastula)`
2. `library(keyring)`
3. `library(rvest)`
4. `library(dplyr)`
5. `library(cronR)`

III. Construction du code : Justification du choix des packages

- ▶ **blastula** : Création et envoi d'emails avec des options avancées (personnalisation, inclusion d'images, etc.) → notifications par email.
- ▶ **keyring** : Stocker en toute sécurité les informations sensibles comme les identifiants ou mots de passe (ici pour la configuration SMTP Gmail).
- ▶ **rvest** : Facilite l'extraction de données d'un site web (web scraping). Utilisé pour récupérer les informations (nom, prix, lien) depuis les annonces eBay.
- ▶ **dplyr** : Indispensable pour manipuler, nettoyer et transformer les données sous forme de tableaux.
- ▶ **cronR** : Le package cronR permet de planifier et d'automatiser l'exécution de scripts R à des intervalles réguliers en utilisant le système de tâches cron.

IV. Listing des URL de recherche

Le listing des URL de recherche sert à *définir les pages web spécifiques* à partir desquelles les données (comme les annonces de chaussures) seront extraites pour le traitement et l'analyse.

V Extraire et filtrer les données brutes depuis chaque URL.

1. *Lecture de la page HTML* : “**read_html(url)**” charge le contenu HTML de l’URL.
2. *Extraction des données* : “**html_nodes()**identifie” les sections HTML pertinentes (titres, prix, liens).
3. *Filtrage des données* : Les résultats non pertinents contenant “**Shop on eBay**” sont exclus.
4. *Formatage* : Les résultats filtrés sont stockés dans un data frame pour une manipulation facile.

VI. Application de la fonction et combinaison des résultats

- ▶ **“lapply()”** applique la fonction **“extract_filtered_data”** à chaque URL de la liste.
- ▶ Les résultats de chaque URL sont combinés avec **“do.call(rbind, ...)”**, créant une table unique contenant toutes les annonces.

VII. Nettoyage et transformation des données

Objectif : Préparer les données pour analyse

- ▶ *Extraction des prix numériques* : Les caractères non numériques sont supprimés.
- ▶ *Remplacement des virgules par des points pour convertir les prix*
- ▶ *Filtrage des prix* : Annonces dont le prix est **inférieur à 60 EUR** ou **supérieur à 150 EUR** sont exclues.
- ▶ *Sélection des colonnes finales* : Seules les colonnes nécessaires (**Name, Price, Price_num, Link**) sont conservées.

VIII. Création d'un corps de message email et envoi de l'email

Objectif : Générer un texte contenant les annonces filtrées. Chaque ligne contient le nom, le prix et un lien cliquable vers l'annonce.

- ▶ **SMTP (Simple Mail Transfer Protocol)**: Un protocole standart pour l'envoi de mails. Dans R, l'utilisation d'un serveur SMTP permet d'automatiser l'envoi d'e-mails directement depuis le script.
- ▶ Utilisation de **keyring** pour stocker les identifiants en toute sécurité
- ▶ Envoi via le serveur SMTP de Gmail Paramètres : **from**), **to**, **subject**, et configuration sécurisée des identifiants via "keyring".

IX. Automatisation du programme

Objectif : Envoie d'une notification contenant les liens, les prix et les modèles des chaussures trouvées sur ebay.fr

Fréquence: Tous les jours à 8h du matin

`cron_add(command = cmd, frequency = "daily", at = "08:00", id = "chaussures_daily", description = "Mail quotidien des chaussures de course à pied sur ebay.fr"`

Utilisation de la commande **cronR** : scraping du site ebay.fr quotidiennement

X. Conclusion

- ▶ Il est désormais possible **d'exécuter le script directement depuis RStudio** ou en ligne de commande pour démarrer l'extraction des données. Le script se charge de tout : il va **extraire les annonces de chaussures de sport sur eBay**, les **filtrer selon les critères définis**, puis envoyer un email à l'adresse spécifiée.

Concernant la personnalisation des paramètres :

- ▶ Changez l'URL dans la liste des URLs pour explorer d'autres pages eBay. Mettez à jour la configuration SMTP dans le fichier pour personnaliser les paramètres d'email (comme votre adresse Gmail).
- ▶ **Exécution automatique:** Vous pouvez programmer le script pour s'exécuter régulièrement en utilisant un planificateur de tâches sur votre système (par exemple, cron sur Linux ou Task Scheduler sur Windows).