

Exercices - 4

Préparation

1. Vérifiez que les options globales de RStudio sont conformes aux options recommandées dans le cours.
2. Créez un projet RStudio intitulé `cours_r_partie_4`. Ce projet sera créé vide. Nous initialiserons un projet git en cochant la case adéquate.

Exercice

Pour cet exercice, nous allons utiliser le même jeu de données issu de <https://data.nantesmetropole.fr/>. L'objectif de cet exercice est de mettre en pratique les librairies découvertes aujourd'hui. Comme lors de la partie 3, n'hésitez pas à réaliser un `commit` après chaque résolution de question, et à `push` régulièrement sur le répertoire distant.

1. Requêtez l'**ensemble** du jeu de données compagnon à l'aide d'`httr2` (sans télécharger le `csv`). Ce jeu contient des informations complémentaires sur chaque boucle. Importer ce jeu de données sous le nom `df_velo_info`. La requête renvoie par défaut le 20 premières lignes uniquement, modifier la `query` pour obtenir l'ensemble du jeu de données (76 lignes).
2. Requêtez une **sous-partie** du jeu de données principal à l'aide d'`httr2` (sans télécharger le `csv`). Limiter la sélection aux **deux directions d'une même boucle** (e.g. 880 et 881 pour *Madeleine vers nord* et *Madeleine vers sud*) et **un mois** (e.g. "2025-09") via le paramètre de `query refine` (cf. doc API). Importer ce jeu de données sous le nom `df_velo_sample`. Vérifier que votre jeu de données comporte le nombre de lignes attendu.
3. Réaliser une jointure entre les jeux de données `df_velo` et `df_velo_info`. Avec la fonction `write.csv`, sauvegardez le jeu de données obtenu au format `csv` dans un sous-dossier `/data`. Dans le fichier `.gitignore`, ajoutez la ligne `data/` puis sauvegardez. À quoi sert cette étape ?
4. Importez le fichier `csv` de la question précédente sans utiliser `readr` (**utiliser la fonction de base `read.delim`**) et en nommant la variable `df_velo`. Convertir le `data.frame` en `tibble` et filtrer les anomalies (probabilité d'anomalie classée `Fort` ou `Faible`, `Total` négatif ou supérieur à 10000). Quelles différences notez-vous par rapport à un `import readr` ?
5. Créer une nouvelle colonne `mois` qui indique le mois de la mesure en toute lettre (janvier, février etc). Calculer le nombre de trajets par mois.
6. Quelles sont les heures de la journée avec le plus de trajet en moyenne ? Pour répondre à cette question, vous pouvez dans un premier temps *pivoter* le `data.frame` pour obtenir une seule colonne `heure`.
7. Utiliser `stringr` et les `regex` pour convertir la colonne `heure` en variable numérique.

8. Visualiser le résultat de la question précédente sous forme de graphique. Adapter le graphique pour afficher l'affluence moyenne par tranche horaire. Quelle est votre analyse ?
9. Séparer les éléments de la colonne `Boucle` de `comptage` pour que le `numéro`, le `nom` de la boucle et sa `direction` (nord, sud etc) soient renseignées dans trois nouvelles colonnes distinctes `numero`, `boucle` et `direction`. Aidez-vous de `stringr` et `dplyr::separate_...`. Quelle est la direction comptant le plus de trajet ?
10. Calculer un flux net entre les deux directions par tranche horaire et par jour. On commencera par sélectionner uniquement nos colonnes d'intérêt. Visualiser la distribution de ces flux nets par jour et par tranche horaire (deux graphiques).
11. (*Bonus*) Export cartographie : À partir des colonnes `geolocalisation.lon` et `geolocalisation.lat` obtenues après jointure, réaliser une geolocalisation de vos bornes avec le package `leaflet`.

Aide pour l'exercice

- Utilisez la fonction d'aide `?<code>` pour déterminer les bons paramètres de lecture.
- La sous-sélection de données peut se faire via la fonction `filter` de `dplyr`.
- Les fonctions `group_by` et `summarise` de `dplyr` peuvent vous aider à calculer des sommes et des moyennes.
- La fonction `mutate` de `dplyr` peut vous aider à créer des nouvelles colonnes à partir de colonnes existantes.
- L'extraction d'information de date peut se faire via la fonction `month` et `ymd` de `lubridate`.
- La famille de fonctions `pivot_...` de `tidyr` peuvent vous aider à manipuler les horaires de passage.
- Pour extraire et manipuler une chaîne de caractère, penser à `{stringr}`
- La famille de fonctions `join` de `dplyr` peuvent vous aider à combiner des `data.frame`.
- La carte des boucles est disponible ici.