

Contrôle continu BASH / Python

Durée : 01h00.

En cas de réalisation par un binôme, on précisera les noms et prénoms des membres du binôme au début de chaque fichier.

L'énoncé présente le minimum attendu. La notation tiendra compte de l'effort fait pour écrire les programmes suivant les bonnes pratiques de développement (commentaires appropriés, découpage en fonctions, ...) et pour ajouter des fonctionnalités intéressantes, y compris pour la gestion des cas d'erreurs.

Nantes Métropole dispose de boucles de comptage des vélos sur les principales pistes cyclables. Le nombre de vélos détectés heure par heure est accessible dans un fichier JSON sur le site data.nantesmetropole.fr. La table 1 montre une entrée de ce fichier.

Un exemplaire du fichier JSON est disponible dans le répertoire `~goualard-f/X1BI030/comptages_vélos.json`.

TABLE 1 – Entrée du fichier JSON de comptage des vélos.

```
{
  "datasetid": "244400404_comptages-velo-nantes-metropole",
  "recordid": "85b8ffd26ba37c81a1ea0bfcc59a51ef7f62d42e",
  "fields": {
    "01": 0,
    "20": 12,
    "04": 0,
    "03": 1,
    "07": 68,
    "22": 7,
    "05": 3,
    "libelle": "De Gaulle sortie Clémenceau vers Nord",
    "18": 41,
    "21": 15,
    "19": 28,
    "16": 24,
    "08": 203,
    "00": 0,
    "11": 11,
    "15": 15,
    "jour": "2022-11-17",
    "dateformat": "2022-11-17",
    "boucle_num": "0669",
    "09": 88,
    "12": 35,
    "10": 26,
    "23": 7,
    "boucle_libelle": "0669 - De Gaulle sortie Clémenceau vers Nord",
    "14": 16,
    "total": 706,
    "vacances_zone_b": "Hors Vacances",
    "jour_de_la_semaine": "4",
    "02": 0,
    "06": 19,
    "17": 49,
    "13": 38,
    "record_timestamp": "2022-11-18T05:26:00+01:00"
  }
}
```

On souhaite afficher un *bar plot* de la fréquentation moyenne sur une boucle pour un jour de la semaine donné (voir figure 1).

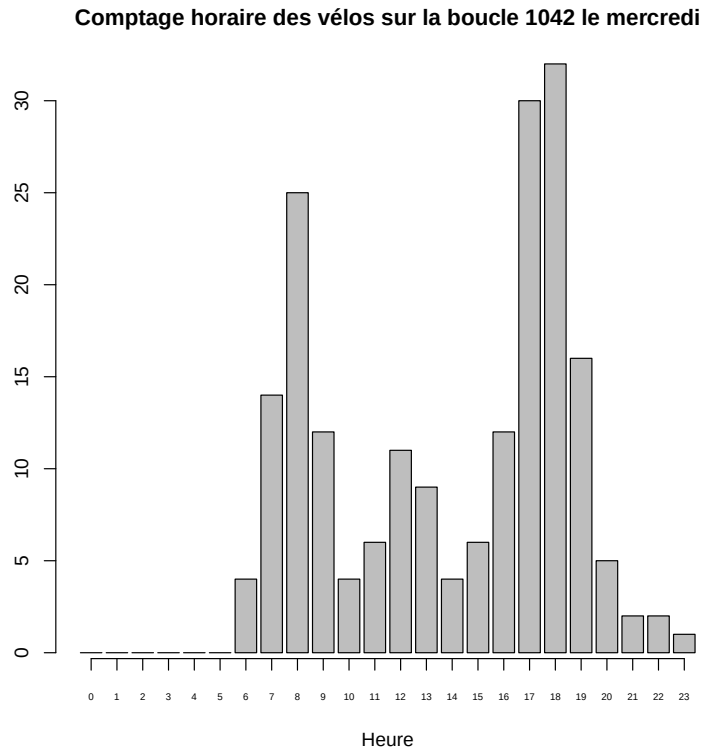


FIGURE 1 – Bar plot du passage moyen heure par heure sur la boucle 1042 le mercredi.

Le programme Python biketraffic.py

On va commencer par écrire le programme Python `biketraffic.py` prenant en paramètres le chemin d'un fichier JSON de comptage, un numéro de boucle (champ `boucle_num`) et un jour de la semaine (lundi, mardi, ...). Le programme doit afficher sur la sortie standard la liste des *nombre de passages moyens* constatés heure par heure pour le jour et la boucle donnés. Exemple :

```
# ./biketraffic.py ~goulard-f/X1BI030/comptages_vélos.json 1042 mercredi
0 0 0 0 0 4 14 25 12 4 6 11 9 4 6 12 30 32 16 5 2 2 1
```

Au minimum, le programme doit :

- Afficher un message d'aide sur la sortie d'erreur standard et retourner le code d'erreur 1 si le nombre de paramètres est incorrect ;
- Afficher un message d'erreur sur la sortie d'erreur standard et retourner le code d'erreur 2 si le troisième paramètre n'est pas un jour de la semaine ;
- Afficher un message d'erreur sur la sortie d'erreur standard et retourner le code d'erreur 3 si le fichier JSON ne peut pas être lu.

Enfin, le programme doit retourner le code d'erreur 4 si aucune donnée de trafic n'a été trouvée.

Le script BASH biketraffic.sh

Le script BASH `biketraffic.sh` sert d'interface avec l'utilisateur. Il accepte en paramètres le chemin d'accès au fichier JSON de données de comptage, le numéro sur quatre chiffres d'une boucle de comptage et le jour de la semaine à considérer. En retour, il doit créer le fichier PDF contenant le *bar plot* ; le nom du fichier est composé du numéro de la boucle et du jour étudié. Exemple :

```
# ./biketraffic.sh ~goulard-f/X1BI030/comptages_vélos.json 1042 mercredi
# Création du fichier 1042_mercredi.pdf
```

Le script ne doit créer aucun fichier PDF si aucune donnée n'est disponible ; il doit, par contre, afficher un message. Exemple :

```
# ./biketraffic.sh comptages_vélos.json 1111 vendredi  
Pas de trafic trouvé pour la boucle 1111.
```

Le *bar plot* est réalisé par le programme `bar.R` [disponible sur madoc](#) à partir de la sortie du programme `biketraffic.py`. Il prend sur son entrée standard la liste des valeurs moyennes rapportées heure par heure et en paramètres le numéro de boucle ainsi que le jour de la semaine.

Le script doit afficher un message d'erreur sur la sortie d'erreur standard et retourner le code d'erreur 1 si le nombre de paramètres est incorrect ; il doit afficher un message d'erreur sur la sortie d'erreur standard et retourner le code d'erreur 2 si le fichier JSON n'est pas lisible ; il doit afficher un message d'erreur et retourner le code d'erreur 3 si les fichiers `biketraffic.py` ou `bar.R` ne sont pas accessibles dans le répertoire courant. Enfin, si un fichier avec le nom du fichier PDF à produire existe déjà, le script doit afficher un message d'erreur et s'arrêter en retournant le code d'erreur 4.