STID 1 - Programmation Statistique
TP2
Manipuler un dataframe

Anthony SARDELLITTI

2023-01-01

Contents

1 Ressources documentaires

Pour réaliser ce TP, vous aurez besoin des ressources suivantes :

- Importer un fichier excel
- Fonctions de tests et comparaisons
- Indexation
- Filtres et sélection
- Les fonctions de tests et opérateurs de comparaison
- Trier
- Agréger

2 Exercices

Pour rendre ce TP, vous pouvez envoyer un mail à *anthony.sardellitti@hotmail.fr* **OU** compléter votre repository GitHub.

Pensez à commenter votre code.

On utilise le fichier `pokemon.xlsx` qui décrit les statistiques des pokemon des deux premières générations. Le fichier est issu du site Kaggle. Il a été adapté pour ce TP. Pour réaliser ce TP, télécharger le fichier en cliquant ici. Voici une description des données :

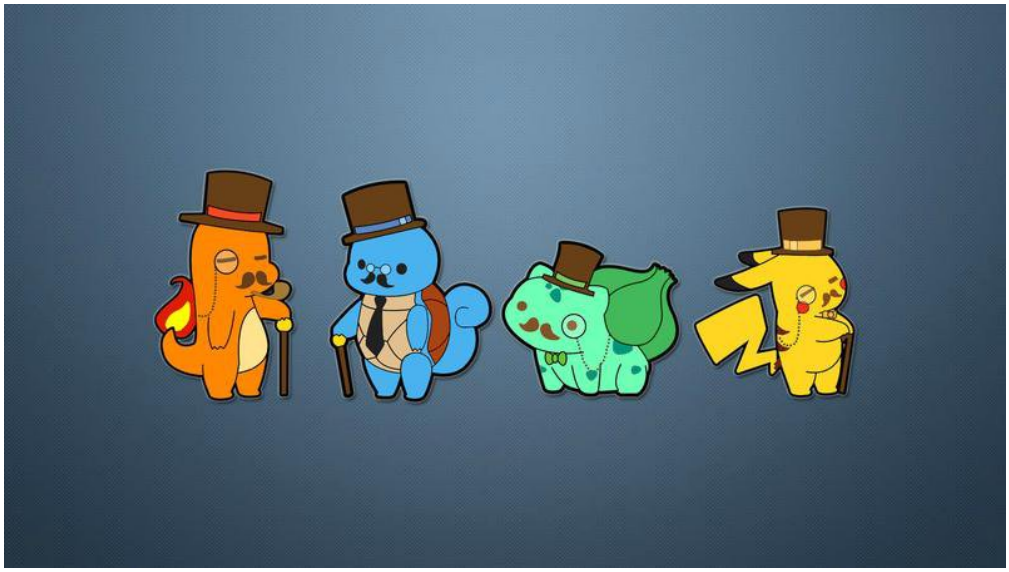

Figure 1: Pokemon

- `pokedex_number` : numéro du pokemon
- `nom` : nom du pokemon
- `generation` : le numéro de génération dont est issu le pokemon
- `is_legendary` : Oui / Non si le pokemon est légendaire
- `type` : le type du pokemon
- `weight_kg` : le poids du pokemon en kg
- `height_m` : la taille du pokemon en mètre
- `attack` : la puissance d'attaque du pokemon
- `defense` : la puissance de défense du pokemon
- `speed` : la vitesse du pokemon

2.1 Exercice 1 : Importer les données

a. Importez le jeu de données `pokemon.xlsx` à l'aide du package `readxl`.
b. Combien de lignes, colonnes sont présentes dans ce dataset (utilisez les fonctions adaptées) ?
c. Affichez le nom des colonnes.
d. Affichez le type des colonnes avec la fonction adaptée.
e. On souhaite analyser les variables `generation`, `is_legendary`, et `type` en tant que variables qualitatives. Modifier le type de ces variables pour les transformer en type factor.
f. Combien de niveaux (*levels*) sont présents dans ces variables ?
g. Affichez un résumé des données avec la fonction adaptée.

2.2 Exercice 2 : Statistiques descriptives

a. Déterminer la moyenne de la variable `weight_kg`.
b. Déterminer la médiane de la variable `weight_kg`.
c. Déterminer les quartiles de la variable `height_m`.
d. Déterminer les déciles de la variable `height_m`.
e. Déterminer la variance et l'écart-type de la variable `weight_kg`.
f. Déterminer un tri à plat pour compter les effectifs des modalités de chaque variable *factor* en triant chaque sortie par ordre décroissant.

2.3 Exercice 3 : Tris et Selections

Pour chaque question suivante, affectez le résultat de la requête dans un objet puis calculez sa dimension. Exemple :

```
#Selectionnez les deux premières colonnes du data frame
requete_0 <- pokemon[,1:2]
dim(requete_0)
```

```
## [1] 251   2
```

a. Sélectionnez la colonne `nom` et `is_legendary`.
b. Sélectionnez les 50 premières lignes et les deux premières colonnes.
c. Sélectionnez les 10 premières lignes et toutes les colonnes.
d. Sélectionnez toutes les colonnes sauf la dernière.
e. Triez le dataset par ordre alphabétique et afficher le `nom` du pokemon dela première ligne.
f. Triez le dataset par `weight_kg` en ordre **décroissant**, et afficher le `nom`du pokemon de la première ligne
g. Triez le dataset par `attack` en ordre **décroissant** puis par `speed` en ordre **croissant**, et afficher le `nom` des pokemons des 10 premières lignes.

2.4 Exercice 4 : Tris et Filtres

Pour chaque question suivante, affectez le résultat de la requête dans un objet puis calculez sa dimension. Pour faciliter la lecture, sélectionnez la colonne `nom`et les colonnes concernées par le filtre. Exemple :

```
#Selectionnez les pokemons de type feu
requete_0 <- pokemon[ pokemon$type == "fire", c("nom","type")]
dim(requete_0)
```

```
## [1] 20  2
```

a. Filtrez sur les pokemons qui ont 150 ou plus d'`attack` puis trier le résultat par ordre décroissant d'`attack`.
b. Filtrez sur les pokemons de `type` *dragon*,*ghost*,*psychic* et *dark*
c. Filtrez sur les pokemons de `type` *fire* avec plus de 100 d'`attack`, puis trier le résultat par ordre décroissant d'`attack`.
d. Filtrez sur les pokemons qui ont entre 100 et 150 de `speed`. Les trier par `speed` décroissant.
e. Filtrez sur les pokémons qui ont des valeurs manquantes sur la variable `height_m`.
f. Filtrez sur les pokemons qui ont des valeurs renseignées à la fois pour la variable `weight_kg` **et** la variable `height`.
g. Filtrez sur les pokemons pesant plus de 250 kg et affichez le résultat pour vérifier.

2.5 Exercice 5 : Agrégations

Pour chaque question suivante, affectez le résultat de la requête dans un objet puis calculez sa dimension. Exemple :

```
#Calculez la vitesse moyenne par generation
requete_0 <- aggregate(x = speed ~ generation, data = pokemon , FUN = mean)
dim(requete_0)
```

```
## [1] 2 2
```

a. Calculez l'`attack` moyenne en fonction de la variable `type`, puis filtrez sur les 3 types avec les moyennes les plus élevées.
b. Calculez le nombre de pokemon par `type` , puis triez par ordre décroissant ces effectifs.
c. Calculez la médiane de `weight_kg` par `type`.
d. Calculez le nombre de pokemon par `type` et `generation`
e. Calculez la moyenne de chaque critère (`weight_kg`, `height_m`, `attack`, `defense` et `speed`) en fonction de chaque `type`.