TP 1

Analyse de données - Master 1
Lundi 22 septembre

Jeu de données

Tableau **Rio 2016 – athlètes et médailles (niveau athlète)** (CSV en ligne) :
`https://raw.githubusercontent.com/flother/rio2016/master/athletes.csv`

Partie 1 — Bases de R

1. Créez un `character`, un `numeric`, un `logical` et un `NULL`. Affichez `class()` pour chacun.
2. Créez un **vecteur** `v <- c(1,5,6)` et une **liste** `L <- list(1,"a",TRUE)`. Affichez `class(v)`, `class(L)` et l'élément `L[[2]]`.
3. Créez un **factor** pour `c("France","France","Italie")` avec `levels=c("France","Italie","Espagne")` puis affichez `levels()`.
4. **Import CSV** du jeu de données dans `athletes_raw`, puis affichez `head(athletes_raw)` et `str(athletes_raw)`.
5. **Bonus : import et export Excel** : Enregistrer le dataframe au format Excel avec le Package *writexl* sous `athletes_rio2016.xlsx`. Lisez ce fichier avec `le package readxl::read_excel` dans `athletes_xlsx`, puis affichez `head()`.

Partie 2 — Manipulations de données

6. Affichez les **noms de colonnes** (`names()`), le **nombre de lignes/colonnes** (`nrow()`, `ncol()`) de `athletes_raw`.
7. Affichez la **classe** de chaque colonne (`sapply(athletes, class)`).
   Si `height`, `weight`, `gold`, `silver`, `bronze` ne sont pas numériques, convertissez-les en `as.numeric`.
8. **Sous-table au niveau athletes_raw** : créez `athletes` avec les colonnes `name`, `nationality`, `sex`, `date_of_birth`, `height`, `weight`, `sport`, `gold`, `silver`, `bronze`.
   Affichez `head(athletes)`.
9. **Colonne Âge** Ajoutez la variable `age` (à la date du **21/08/2016**). Affichez `summary(athletes$age)`.
   Indication R : `athletes$age <- as.numeric(difftime(as.Date("2016-08-21"), as.Date(athletes$date_of_birth), "days"))/365.25`
10. **Colonne total** Créez la variable `total = gold + silver + bronze`.
    Indication R : `athletes$total <- rowSums(a1[,c("gold","silver","bronze")], na.rm=TRUE)`

11. **Filtre (pays)** : créez `fr` avec les athlètes de nationalité `"FRA"` dans `athletes`.
    Bonus : sélectionnez aussi `"USA"` et `"CHN"`.
12. **Tri** : affichez les **10 athlètes** ayant le plus grand `Total` (colonnes `name`, `nationality`, `sport`, `Total`).
13. **Pourcentage d'or (athlète)** : pour les athlètes `"FRA"`, calculez leur pourcentage de médailles d'or `Gold_pct <- gold / Total * 100` (attention si `Total==0`, soit on mettra NA, soit 0). Affichez les 6 premières lignes de `fr` avec `name`, `sport`, `gold`, `Total`, `Gold_pct`.
14. **Variables numériques** : à partir de `athletes`, créez `athletes_num` en ne gardant que les colonnes numériques. Faites un `summary`. Supprimer les lignes avec des NA si besoin avec na.omit().
15. **Corrélations et redondance d'information (athlète)** : calculez la matrice de corrélation entre `height`, `weight`, `Age`, `gold`, `silver`, `bronze`, `Total`.
    Commentez : quelles paires sont fortement corrélées ? `Total` apporte-t-il une information nouvelle vs. `gold/silver/bronze` ?
16. **Centrage–réduction :** Standardisez `athletes_num` variable par variable (moyenne 0, écart-type 1). Vérifiez rapidement que les moyennes sont ≈ 0 et les écarts-types ≈ 1.
    Indication R : `X_cr <- scale(X); colMeans(X_cr); apply(X_cr,2,sd)`. Pensez à vérifier les types de données en entrée et sortie.
17. **Ajouter des rownames à X_cr** Créez des **rownames** pour `X_cr` en utilisant la variable names
    Indication R :`rownames(X_cr) <- athletes$names`
18. **Distances :** Calculez la distance euclidienne entre athlètes sur `X_cr` puis transformez en matrice.
    Indication R : `d <- dist(X_cr, method="euclidean"); D <- as.matrix(d)`
19. **Voisins d'un athlète :** Choisissez un athlète (`who`) parmi la variable names (par exemple le 1er ou un nom ciblé) et affichez ses 6 plus proches voisins (distance croissante).
    Indication R : `who <- athletes$names[1]; sort(D[who, ])[1:6]`
    Astuce : pour cibler un nom, `who <- athletes$names[grep("PHELPS", athletes$names, ignore.case=TRUE)[1]]`
20. **Interprétation :** En 3–4 lignes, expliquez ce que signifie « être proche » *après* centrage–réduction (mélange de morphologie et résultats), et pourquoi cette étape est nécessaire quand les variables n'ont pas la même échelle (ex. : `Total` vs `height`). Mentionnez une limite possible (athlètes avec `Total=0`, valeurs manquantes, etc.).