SY09 Printemps 2018
TP 1 — Manipulation de données

1 Chargement d'un jeu de données

Les jeux de données sont communément stockés dans des fichiers textes au format dit « csv » (*comma separated value*). Il s'agit d'un format décrivant un tableau individus–variables : une ligne liste les caractéristiques d'un individu, séparées par une virgule ; et une colonne liste les valeurs d'une variable pour tous les individus. Dans certains fichiers, la première ligne est parfois une ligne d'en-tête (ou `header`) spécifiant le nom de chacun des prédicteurs, il faudra donc le spécifier à `R`. Parfois, la première colonne n'est pas un prédicteur mais un identifiant ou un nom d'individu qui n'est pas un prédicteur, il faudra là encore le spécifier à `R`. Les fichiers « csv » ont plusieurs variantes, le séparateur (la virgule pour le fichier « csv ») peut changer. La plupart du temps, le séparateur est une espace, un point virgule ou une tabulation.

Il existe plusieurs fonctions pour charger un fichier texte dans un `data.frame`. La plus générale est la fonction `read.table` qui est utilisée par toutes les autres fonctions. Par défault, la fonction `read.table` suppose qu'il n'y a pas d'en-tête et que le séparateur est une espace.

- `read.csv` Fichier séparé par une virgule avec un point pour séparateur décimal. Pas de `header` par défaut.
- `read.csv2` Fichier séparé par un point virgule avec une virgule pour séparateur décimal. Pas de `header` par défaut.
- `read.delim` Fichier séparé par une tabulation avec un point pour séparateur décimal.
- `read.delim2` Fichier séparé par une tabulation avec une virgule pour séparateur décimal.

Les arguments les plus utiles sont les suivants :

- `sep` Spécifie le séparateur
- `header` Spécifie si la première ligne est une ligne d'en-tête
- `col.names` `col.names = 1` si la première colonne est une colonne nommant les individus

Charger les fichiers `iris1.data`, `iris2.data`, `iris3.data` et `iris4.data` en utilisant les bonnes fonctions avec les bons arguments.

Vous pouvez vérifier que le fichier est correctement chargé avec la fonction suivante :

```
> check <- function(data) {
>    stopifnot(colnames(data)[1] == "Sepal.Length",
>              colnames(data)[2] == "Sepal.Width",
>              colnames(data)[3] == "Petal.Length",
>              colnames(data)[4] == "Petal.Width",
>              colnames(data)[5] == "Species",
>              nrow(data) == 150,
>              ncol(data) == 5,
>              is.numeric(data[,1]),
>              is.numeric(data[,2]),
>              is.numeric(data[,3]),
>              is.numeric(data[,4]),
>              is.factor(data[,5]))
>    print("Chargement OK")
> }
```

Vous pouvez également utiliser les fonctions `summary` ou `head` pour vérifier visuellement le résultat du chargement.

2 Conversion de types

Lors du chargement d'un fichier texte, `R` essaie de deviner la classe des prédicteurs. Les classes les plus communes sont les suivantes

- `factor` : Correspond à une variable qualitative
- `character` : Correspond à une variable qualitative mais stocké dans un vecteur plutôt qu'un facteur
- `numeric` : Correspond à une variable quantitative continue
- `integer` : Correspond à une variable quantitative discrète (les entiers naturels)
- `logical` : Correspond à une variable binaire

On peut vérifier la classe d'un prédicteur avec la fonction `class`.

```
> class(iris[,1])
> class(iris[,5])
```

Pour voir comment `R` essaie de deviner la classe des variables, on pourra exécuter les instructions suivantes. Il s'agit de lire un texte comme s'il s'agissait d'un fichier contenant un jeu de donnée avec un seul prédicteur et un seul individu.

```
> class(read.table(text='TRUE')[,1])
> class(read.table(text='F')[,1])
> class(read.table(text='A')[,1])
> class(read.table(text='3.14')[,1])
> class(read.table(text='2')[,1])
```

Pour vérifier la classe de tous les prédicteurs, on peut utiliser la commande `summary` ou l'instruction suivante :

```
> sapply(iris, class)
```

Si la classe n'est pas correctement inférée par `R`, on peut les spécifier explicitement lors du chargement du jeu de données avec l'argument `colClasses`. Si cela n'est pas possible, il faut changer le type après chargement avec les fonctions

- `as.numeric`
- `as.logical`
- `as.integer`
- `as.factor`
- `as.ordered`

Charger le fichier binaire `iris.Rdata` qui contient les jeux de données `iris5` et `iris6` avec l'instruction `load` et ajuster les classes des prédicteurs ainsi que les éventuelles erreurs.

3 Recodage de facteurs

Parfois il est nécessaire de changer les modalités d'un facteur ou de changer leur ordre dans le cas d'un facteur ordonné. On pourra utiliser pour ce faire la commande `factor` :

```
> donnees$champ1 <- factor(donnees$champ1, levels=modalites1)
> donnees$champ2 <- factor(donnees$champ2, levels=modalites2, ordered=T)
```

Le fichier sy02-p2016.csv, que l'on chargera, contient des informations relatives aux étudiants ayant suivi l'UV SY02 au semestre de printemps 2016. Certaines des variables (correcteurs, niveau d'études, résultat) sont des variables qualitatives, éventuellement ordonnées : les déclarer comme telles. Attention : les modalités doivent être les mêmes pour les correcteurs du médian et du final.

4 Valeurs spéciales

Le langage R gère des valeurs spéciales qui peuvent modéliser des valeurs manquantes, des erreurs de calculs ou des dépassements de capacité. Ces valeurs spéciales sont

- NA valeur non disponible (*not available*). Il arrive qu'une donnée soit indisponible ou indéterminée. Plutôt que de mettre une valeur impossible (-1 pour des données positives, -666 pour des températures...) on préfère le signifier explicitement
- Inf valeur infinie. Atteinte lors d'un dépassement de capacité

```
> 10^(10^10)
> 1/0
```

- NaN valeur non définie résultant le plus souvent d'une erreur de calcul, par exemple

```
> 0/0
> Inf/Inf
> Inf-Inf
```

5 Chargement du jeu de données babies

Le jeu de données contenu dans le fichier babies23.data est constitué de 1236 bébés décrits par 23 variables. Charger le jeu de données et sélectionner les colonnes 7, 5, 8, 10, 12, 13, 21, 11. Pour changer les noms de colonnes après chargement, on pourra utiliser l'instruction

```
> names(babies) <- c("bwt", "gestation", "parity", "age", "height", "weight",
  ↪  "smoke", "education")
```

Faites l'histogramme des durées de gestation en jours avec l'instruction

```
> hist(babies$gestation)
```

Que remarquez-vous ?

D'une manière générale dans ce jeu de données, lorsque la valeur de certains prédicteurs est inconnue une valeur prédéfinie est utilisée :

- Pour la colonne `bwt`, on utilise 999
- Pour la colonne `gestation`, on utilise 999
- Pour la colonne `age`, on utilise 99
- Pour la colonne `height`, on utilise 99
- Pour la colonne `weight`, on utilise 999
- Pour la colonne `smoke`, on utilise 9
- Pour la colonne `education`, on utilise 9

Remplacer toutes ces valeurs prédéfinies par NA.

Les variables `education` et `smoke` sont en fait des variables qualitatives. Transformer la variable `education` en un facteur ordonné.

Pour la variable `smoke`, la documentation du jeu de données dit

```
smoke: does mother smoke?
0=never,
1=smokes now,
2=until current pregnancy,
3=once did, not now,
9=unknown
```

Recoder la variable `smoke` de manière à ce que la modalité « 1 » soit recodée en `Smoking` et les autres modalités en `NonSmoking`.