

# Capsule 4

## Objectifs de la capsule

À la fin de cette capsule, vous serez en mesure de:

1. Comprendre qu'est-ce qu'un bon jeu de données.
2. Faire la lecture des principaux types de fichiers .dat, .csv, .txt.
3. Comprendre ce qu'est-ce qu'un *data frame* et connaître les principales fonctions utilisées sur cette classe.
4. Exporter des données dans un fichier depuis R.

## Capsule vidéo

[https://youtu.be/555\\_kicuHJg](https://youtu.be/555_kicuHJg)

## Exercices

### **i** Note

Veillez noter qu'il est possible d'avoir plus d'une bonne réponse par question. Vous pouvez reprendre chaque exercice grâce aux boutons "Start Over". Le bouton "Indice" est là pour être utilisé!

## Les jeux de données

Le jeu de données suivant (produit par le *Pew Research Center*) explore la relation entre le salaire et la religion aux États-Unis.

religion	below10k	from10to20k	from20to30k	from30to40k
Agnostic	27	34	60	81
Atheist	12	27	37	52
Buddhist	27	21	30	34
Catholic	418	617	732	670
Don't know/refused	15	14	15	11
Evangelical Prot	575	869	1064	982

### **i** Note

## Importer des données dans R

En utilisant la fonction `read.table()`, écrivez une ligne de code qui permet de lire correctement le fichier `BI01006_H2018_Echantillon_espace.dat` dont la structure ressemble à ceci.

*Ne pas changer le nom du fichier, seulement remplacer ... avec la bonne réponse.*

```
couleur_yeux  taille_cm  nombre_choisi  cours_par_session  peur_des_biostats
piece_monnaie genre
Noir 174 92 6 Non Face Fille
Bleu 168 20 5 Pas_sur Face Fille
Noisette 158 11 5 Pas_sur Face Fille
Autre 177 99 5 Non Face Gars
Noisette 152 87 5 Pas_sur Pile Fille
Bleu 177 7 5 Non Pile Gars
Bleu 163 25 5 Pas_sur Pile Fille
Noir 186 27 4 Non Pile Gars
Bleu 150 87 5 Non Face Gars
```

```
{webr}
#| autorun: false
read.table(file = "https://tinyurl.com/bio1006dat", ...)
```

### Indice

Le paramètre `sep` permet de spécifier le caractère utilisé pour séparer les colonnes de variables.

⚠ Solution

```
read.table(file = "https://tinyurl.com/bio1006dat", sep = " ")
```

	V1	V2	V3	V4	V5
1	couleur_yeux	taille_cm	nombre_choisi	cours_par_session	peur_des_biostats
2	Noir	174	92	6	Non
3	Bleu	168	20	5	Pas_sur
4	Noisette	158	11	5	Pas_sur
5	Autre	177	99	5	Non
6	Noisette	152	87	5	Pas_sur
7	Bleu	177	7	5	Non
8	Bleu	163	25	5	Pas_sur
9	Noir	186	27	4	Non
10	Bleu	150	87	5	Non
11	Noisette	170	6	5	Non
12	Noir	165	88	5	Pas_sur
13	Bleu	180	6	3	Oui
14	Bleu	153	42	4	Pas_sur
15	Bleu	177	9	5	Non
16	Bleu	1.53	8	5	Pas_sur
17	Bleu	156	7	5	Pas_sur
18	Autre	152	27	4	Pas_sur
19	Noisette	165	20	4	Oui
20	Noisette	168	7	5	Non
21	Noisette	158	9	5	Oui
22	Noir	173	8	5	Oui
23	Autre	177	0	4	Pas_sur
24	Noisette	164	1	5	Pas_sur
25	Autre	165.54	11	5	Non
26	Noisette	170	4	5	Pas_sur
27	Bleu	155	22	5	Non
28	Autre	195	69	5	Non
29	Autre	174	66	4	Pas_sur
30	Noisette	183	18	5	Pas_sur
31	Autre	164	66	4	Pas_sur
32	Bleu	153	1	5	Non
33	Bleu	173	13	4	Pas_sur
34	Bleu	161	4	5	Non
35	Autre	170	14	4	Non
36	Autre	164	16	5	Pas_sur
37	Noisette	152	7	4	Non
38	Bleu	170	4	5	Oui
39	Noisette	170	72	4	Pas_sur
40	Bleu	159	18	4	Pas_sur
41	Bleu	182	42	5	Pas_sur
42	Autre	161	44	5	Non
43	Noisette	186	5	5	Non
44	Bleu	168	69	5	Non
45	Bleu	175	31	6	Pas_sur
46	Bleu	162	24	5	Non
47	Autre	164	29	5	Pas_sur
48	Bleu	195.68	16	5	Pas_sur
49	Bleu	155	22	5	Oui
50	Noir	182	42	4	Non

## Les data frame

**i** Note

**i** Note

Le *data frame* sondage\_bio1006 qui présente le résultat d'un sondage effectué dans le cours de biostatistiques BIO-1006 du programme de baccalauréat en Biologie de l'Université Laval.

sondage\_bio1006

	couleur_yeux	taille_cm	nombre_choisi	cours_par_session	peur_des_biostats
1	Noir	174.00	92	6	Non
2	Bleu	168.00	20	5	Pas_sur
3	Noisette	158.00	11	5	Pas_sur
4	Autre	177.00	99	5	Non
5	Noisette	152.00	87	5	Pas_sur
6	Bleu	177.00	7	5	Non
7	Bleu	163.00	25	5	Pas_sur
8	Noir	186.00	27	4	Non
9	Bleu	150.00	87	5	Non
10	Noisette	170.00	6	5	Non
11	Noir	165.00	88	5	Pas_sur
12	Bleu	180.00	6	3	Oui
13	Bleu	153.00	42	4	Pas_sur
14	Bleu	177.00	9	5	Non
15	Bleu	1.53	8	5	Pas_sur
16	Bleu	156.00	7	5	Pas_sur
17	Autre	152.00	27	4	Pas_sur
18	Noisette	165.00	20	4	Oui
19	Noisette	168.00	7	5	Non
20	Noisette	158.00	9	5	Oui
21	Noir	173.00	8	5	Oui
22	Autre	177.00	0	4	Pas_sur
23	Noisette	164.00	1	5	Pas_sur
24	Autre	165.54	11	5	Non
25	Noisette	170.00	4	5	Pas_sur
26	Bleu	155.00	22	5	Non
27	Autre	195.00	69	5	Non
28	Autre	174.00	66	4	Pas_sur
29	Noisette	183.00	18	5	Pas_sur
30	Autre	164.00	66	4	Pas_sur
31	Bleu	153.00	1	5	Non

32	Bleu	173.00	13	4	Pas_sur
33	Bleu	161.00	4	5	Non
34	Autre	170.00	14	4	Non
35	Autre	164.00	16	5	Pas_sur
36	Noisette	152.00	7	4	Non
37	Bleu	170.00	4	5	Oui
38	Noisette	170.00	72	4	Pas_sur
39	Bleu	159.00	18	4	Pas_sur
40	Bleu	182.00	42	5	Pas_sur
41	Autre	161.00	44	5	Non
42	Noisette	186.00	5	5	Non
43	Bleu	168.00	69	5	Non
44	Bleu	175.00	1	6	Pas_sur
45	Bleu	162.00	24	5	Non
46	Autre	164.00	29	5	Pas_sur
47	Bleu	195.68	16	5	Pas_sur
48	Bleu	155.00	22	5	Oui
49	Noir	182.00	42	4	Non
50	Noisette	180.00	2	4	Oui
51	Noisette	166.00	28	6	Non
52	Noisette	173.00	94	4	Oui
53	Autre	177.00	7	5	Non
54	Noir	161.00	5	4	Pas_sur
55	Bleu	180.00	14	4	Pas_sur
56	Noir	177.00	6	5	Oui
57	Bleu	160.00	34	4	Oui
58	Bleu	165.00	26	4	Pas_sur
59	Noir	165.00	12	5	Pas_sur
60	Noisette	157.00	66	4	Pas_sur

piece\_monnaie genre

1	Face	Fille
2	Face	Fille
3	Face	Fille
4	Face	Gars
5	Pile	Fille
6	Pile	Gars
7	Pile	Fille
8	Pile	Gars
9	Face	Gars
10	Pile	Fille
11	Pile	Fille
12	Pile	Gars
13	Face	Fille
14	Pile	Fille
15	Pile	Autre
16	Pile	Fille
17	Pile	Fille
18	Pile	Fille

```
19      Face Fille
20      Pile Fille
21      Pile Fille
22      Face  Gars
23      Pile Fille
24      Pile Fille
25      Face Fille
26      Face Fille
27      Face  Gars
28      Face Fille
29      Pile  Gars
30      Pile Fille
31      Face Fille
32      Pile Fille
33      Pile Fille
34      Face Fille
35      Pile Fille
36      Pile Fille
37      Face Fille
38      Pile Fille
39      Face Fille
40      Face  Gars
41      Pile Fille
42      Pile  Gars
43      Pile  Gars
44      Pile  Gars
45      Face Fille
46      Pile Fille
47      Face Fille
48      Face Fille
49      Pile  Gars
50      Face  Gars
51      Face Fille
52      Face  Gars
53      Pile  Gars
54      Face Fille
55      Pile Fille
56      Face  Gars
57      Face Fille
58      Pile Fille
59      Face Fille
60      Face Fille
```

Utilisez la fonction `head` pour faire afficher les 4 premières lignes du *data frame* nommé `sondage_bio1006`.

```
{webr}
#| autorun: false
sondage_biol006 <- read.csv("https://tinyurl.com/biol006csv")
head()
```

### ⚠ Solution

```
sondage_biol006 <- read.csv("https://tinyurl.com/biol006csv")
head(sondage_biol006)
```

```
couleur_yeux taille_cm nombre_choisi cours_par_session peur_des_biostats
1      Noir      174          92          6          Non
2      Bleu      168          20          5      Pas_sur
3  Noisette      158          11          5      Pas_sur
4      Autre      177          99          5          Non
5  Noisette      152          87          5      Pas_sur
6      Bleu      177           7          5          Non
piece_monnaie genre
1      Face Fille
2      Face Fille
3      Face Fille
4      Face Gars
5      Pile Fille
6      Pile Gars
```

### i Note

 Solution

```
sondage_biol006[, -1]
```

```
  taille_cm nombre_choisi cours_par_session peur_des_biostats piece_monnaie
1    174.00           92             6             Non             Face
2    168.00           20             5             Pas_sur          Face
3    158.00           11             5             Pas_sur          Face
4    177.00           99             5             Non             Face
5    152.00           87             5             Pas_sur          Pile
6    177.00            7             5             Non             Pile
7    163.00           25             5             Pas_sur          Pile
8    186.00           27             4             Non             Pile
9    150.00           87             5             Non             Face
10   170.00            6             5             Non             Pile
11   165.00           88             5             Pas_sur          Pile
12   180.00            6             3             Oui             Pile
13   153.00           42             4             Pas_sur          Face
14   177.00            9             5             Non             Pile
15     1.53            8             5             Pas_sur          Pile
16   156.00            7             5             Pas_sur          Pile
17   152.00           27             4             Pas_sur          Pile
18   165.00           20             4             Oui             Pile
19   168.00            7             5             Non             Face
20   158.00            9             5             Oui             Pile
21   173.00            8             5             Oui             Pile
22   177.00            0             4             Pas_sur          Face
23   164.00            1             5             Pas_sur          Pile
24   165.54           11             5             Non             Pile
25   170.00            4             5             Pas_sur          Face
26   155.00           22             5             Non             Face
27   195.00           69             5             Non             Face
28   174.00           66             4             Pas_sur          Face
29   183.00           18             5             Pas_sur          Pile
30   164.00           66             4             Pas_sur          Pile
31   153.00            1             5             Non             Face
32   173.00           13             4             Pas_sur          Pile
33   161.00            4             5             Non             Pile
34   170.00           14             4             Non             Face
35   164.00           16             5             Pas_sur          Pile
36   152.00            7             4             Non             Pile
37   170.00            4             5             Oui             Face
38   170.00           72             4             Pas_sur          Pile
39   159.00           18             4             Pas_sur          Face
40   182.00           42             5             Pas_sur          Face
41   161.00           44             5             Non             Pile
42   186.00            5             5             Non             Pile
43   168.00           69             5             Non             Pile
44   175.00            1             6             Pas_sur          Pile
45   162.00           24             8             5             Non             Face
46   164.00           29             5             Pas_sur          Pile
47   195.68           16             5             Pas_sur          Face
48   155.00           22             5             Oui             Face
49   182.00           42             4             Non             Pile
50   180.00            2             4             Oui             Face
```

## Matériel accompagnateur

### Qu'est-ce qu'un bon jeu de données?

Avant de voir comment importer des données dans R, il est bon de bien comprendre ce qu'est un bon jeu de données. Avoir un jeu de données bien organisé est nécessaire pour faire des analyses statistiques ou des graphiques. Il y a deux propriétés fondamentales qui caractérisent un bon jeu de données:

1. Chaque *ligne* représente une *observation*.
2. Chaque *colonne* représente une *variable*.

Imaginons que vous devez faire le recensement de certaines espèces animales à trois différentes stations (station1, station2 et station3). À chacune d'elles, il y a trois espèces différentes (espece1, espece2 et espece3). Notre réflexe est de créer un tableau de données comme suit, avec une colonne par espèce où chaque chiffre correspond au dénombrement noté. C'est ce qu'on appelle un tableau de données au *format large* (*wide layout* en anglais).

<b>station</b>	<b>espece1</b>	<b>espece2</b>	<b>espece3</b>
station1	12	32	65
station2	34	14	85
station3	54	32	5

Le problème avec cette structure est qu'il y a trois colonnes pour représenter les espèces. Les colonnes *espece1*, *espece2* et *espece3* ne sont pas des variables, *mais plutôt des valeurs* d'une variable/colonne que l'on pourrait nommer tout simplement *espece*. Pour créer un jeu de données bien organisé, il faut retenir que chaque colonne représente une variable. C'est ce qu'on appelle le *format long* (*long layout* en anglais).

#### Tip

N'oubliez pas qu'il faut éviter d'utiliser des caractères accentués dans les noms de variables. C'est pourquoi on nommera la colonne *espece* et non *espèce*.

La façon appropriée de représenter ces données consiste à créer seulement trois colonnes:

1. Une colonne *station* contenant le nom de la station. Même si celle-ci existait déjà, remarquez comment les valeurs seront dupliquées autant que nécessaire!
2. Une colonne *espece* contenant les noms des espèces. Remarquez l'omission volontaire des accents, nous y reviendrons.
3. Une colonne *decompte* contenant le dénombrement de chacune des espèces à chaque station.

<b>station</b>	<b>espece</b>	<b>decompte</b>
station1	espece1	12
station1	espece2	32
station1	espece3	65
station2	espece1	34
station2	espece2	14
station2	espece3	85
station3	espece1	54
station3	espece3	32
station3	espece3	5

### Importer des données dans R

La plupart du temps, les données que vous aurez à importer dans R seront présentées dans de simples fichiers *texte*. Les extensions de ces fichiers les plus souvent rencontrées sont: `.txt`, `.dat`, `.tab` et `.csv`. (Note: il est plus simple d'entrer les données dans un tableur de type "Excel" ou Google sheets et de sauvegarder le fichier dans ces formats). La fonction R de base à utiliser pour ouvrir ces fichiers est `read.table()`. Les principaux paramètres de cette fonction que vous devriez connaître par coeur sont (`help(read.table)`):

1. `file`: Chemin d'accès vers le fichier à ouvrir sur votre ordinateur.
2. `skip`: Nombre de lignes à passer/sauter au début de fichier avant de commencer à lire les données. En effet les premières lignes des fichiers de données sont souvent des commentaires sur le contexte, l'échantillonnage, etc. (métadonnées).
3. `header`: Est-ce que la première ligne de données du fichier contient les noms des colonnes? C'est le cas la grande majorité du temps!
4. `dec`: Caractère utilisé pour les valeurs décimales ("," en Anglais ou ";" en Français).
5. `sep`: Caractère utilisé pour séparer les colonnes de variables.

Note sur les chemins d'accès (`file`): La structure du chemin d'accès à un fichier dépend de votre système d'exploitation. Les deux exemples suivants montrent quels pourraient être les chemins d'accès pour un fichier texte nommé `nom_du_fichier.txt` se situant sur le bureau (Desktop) d'ordinateur Linux/Mac et Windows. Ici, `nom_utilisateur` correspond à **votre nom d'utilisateur**

**sur votre système!** De plus, notez que vous devriez toujours utiliser le signe / plutôt que \ dans un chemin d'accès.

Linux/Mac

```
/home/nom_utilisateur/Desktop/nom_du_fichier.txt
```

Microsoft Windows

```
c:/Users/nom_utilisateur/Desktop/nom_du_fichier.txt
```

#### Tip

La fonction `read.table()` est une fonction générique qui peut lire la majorité des fichiers *texte*. Il existe cependant certaines variantes avec des valeurs de paramètres déjà spécifiées pour lire différents types de fichiers afin de nous faciliter la vie. Les deux fonctions les plus souvent utilisées sont:

- `read.csv()`: Pour lire les fichiers où le marqueur de décimale est un point “.” et les colonnes sont séparées par une virgule “,” (système générique anglais).
- `read.csv2()`: Pour lire les fichiers où le marqueur de décimale est une virgule “,” et les colonnes sont séparées par un point virgule “;”. Cette fonction devrait être utilisée si votre système d'exploitation est configuré en Français.

Pour nous familiariser avec ces différents paramètres<sup>1</sup>, nous utiliserons un jeu de données contenant 7 variables (colonnes) et 60 observations (lignes). Ce jeu de données est le résultat d'un sondage effectué dans le cours de biostatistiques BIO-1006 du programme de baccalauréat en Biologie de l'Université Laval. Le jeu de données est dans un fichier nommé `BI01006_H2018_Echantillon.csv`.

1. `couleur_yeux`: Couleur des yeux du répondant.
2. `taille_cm`: Taille en cm du répondant.
3. `nombre_choisi`: Nombre choisi par le répondant (entre 0 et 99).
4. `cours_par_session`: Nombre de cours suivis à la session du répondant.
5. `peur_des_biostats`: Réponse à la question “Est-ce que le répondant a peur des biostatistiques” (non, pas sur, oui).
6. `piece_monnaie`: Résultat du lancer d'une pièce de monnaie (pile, face).
7. `genre`: Genre du répondant (gars, fille, autre).

Avant d'importer les données dans R, il est important d'ouvrir le fichier `BI01006_H2018_Echantillon.csv` pour explorer sa *structure*. Une fois ouvert dans votre éditeur de texte préféré, vous observerez les éléments suivants:

- Il n'y a pas de ligne à passer ou sauter en début du fichier.

---

<sup>1</sup>Voir la capsule *Les fonctions de R* pour une explication plus complète de ce que sont les paramètres.

- La première ligne correspond aux noms des variables.
- Les colonnes sont séparées par une virgule (“,”).
- Le séparateur de décimales est le point (“.”).

En connaissant ces éléments, il est possible de paramétrer la fonction `read.table()` correctement pour importer les données. La commande suivante lit les données contenues dans le fichier pour les assigner<sup>2</sup> dans une variable nommée `df`.

```
df <- read.table("../data/BI01006_H2018_Echantillon.csv", header = TRUE, sep =
",", dec = ".")
```

La classe de l’objet `df` qui a été créé est un `data.frame`. Un `data.frame` (ou structure de données) est simplement un tableau de données où chaque colonne peut avoir un type différent (numeric, character, date, etc.).

```
class(df)
```

```
[1] "data.frame"
```

En regardant le contenu de `df`, on constate que la classe de la variable `couleur_yeux` est `factor` alors que celle de la variable `taille_cm` est `numeric` (`dbl` veut dire double pour *double précision*, du jargon informatique concernant les valeurs numériques).

```
df
```

	couleur_yeux	taille_cm	nombre_choisi	cours_par_session	peur_des_biostats
1	Noir	174.00	92	6	Non
2	Bleu	168.00	20	5	Pas_sur
3	Noisette	158.00	11	5	Pas_sur
4	Autre	177.00	99	5	Non
5	Noisette	152.00	87	5	Pas_sur
6	Bleu	177.00	7	5	Non
7	Bleu	163.00	25	5	Pas_sur
8	Noir	186.00	27	4	Non
9	Bleu	150.00	87	5	Non
10	Noisette	170.00	6	5	Non
11	Noir	165.00	88	5	Pas_sur
12	Bleu	180.00	6	3	Oui
13	Bleu	153.00	42	4	Pas_sur
14	Bleu	177.00	9	5	Non
15	Bleu	1.53	8	5	Pas_sur
16	Bleu	156.00	7	5	Pas_sur

<sup>2</sup>Voir la capsule *Les variables dans R* pour bien comprendre ce que signifie une assignation de valeur à une variable ainsi qu’une classe de variable.

17	Autre	152.00	27	4	Pas_sur
18	Noisette	165.00	20	4	Oui
19	Noisette	168.00	7	5	Non
20	Noisette	158.00	9	5	Oui
21	Noir	173.00	8	5	Oui
22	Autre	177.00	0	4	Pas_sur
23	Noisette	164.00	1	5	Pas_sur
24	Autre	165.54	11	5	Non
25	Noisette	170.00	4	5	Pas_sur
26	Bleu	155.00	22	5	Non
27	Autre	195.00	69	5	Non
28	Autre	174.00	66	4	Pas_sur
29	Noisette	183.00	18	5	Pas_sur
30	Autre	164.00	66	4	Pas_sur
31	Bleu	153.00	1	5	Non
32	Bleu	173.00	13	4	Pas_sur
33	Bleu	161.00	4	5	Non
34	Autre	170.00	14	4	Non
35	Autre	164.00	16	5	Pas_sur
36	Noisette	152.00	7	4	Non
37	Bleu	170.00	4	5	Oui
38	Noisette	170.00	72	4	Pas_sur
39	Bleu	159.00	18	4	Pas_sur
40	Bleu	182.00	42	5	Pas_sur
41	Autre	161.00	44	5	Non
42	Noisette	186.00	5	5	Non
43	Bleu	168.00	69	5	Non
44	Bleu	175.00	1	6	Pas_sur
45	Bleu	162.00	24	5	Non
46	Autre	164.00	29	5	Pas_sur
47	Bleu	195.68	16	5	Pas_sur
48	Bleu	155.00	22	5	Oui
49	Noir	182.00	42	4	Non
50	Noisette	180.00	2	4	Oui
51	Noisette	166.00	28	6	Non
52	Noisette	173.00	94	4	Oui
53	Autre	177.00	7	5	Non
54	Noir	161.00	5	4	Pas_sur
55	Bleu	180.00	14	4	Pas_sur
56	Noir	177.00	6	5	Oui
57	Bleu	160.00	34	4	Oui
58	Bleu	165.00	26	4	Pas_sur
59	Noir	165.00	12	5	Pas_sur
60	Noisette	157.00	66	4	Pas_sur

piece\_monnaie genre

1	Face Fille
2	Face Fille
3	Face Fille

4	Face	Gars
5	Pile	Fille
6	Pile	Gars
7	Pile	Fille
8	Pile	Gars
9	Face	Gars
10	Pile	Fille
11	Pile	Fille
12	Pile	Gars
13	Face	Fille
14	Pile	Fille
15	Pile	Autre
16	Pile	Fille
17	Pile	Fille
18	Pile	Fille
19	Face	Fille
20	Pile	Fille
21	Pile	Fille
22	Face	Gars
23	Pile	Fille
24	Pile	Fille
25	Face	Fille
26	Face	Fille
27	Face	Gars
28	Face	Fille
29	Pile	Gars
30	Pile	Fille
31	Face	Fille
32	Pile	Fille
33	Pile	Fille
34	Face	Fille
35	Pile	Fille
36	Pile	Fille
37	Face	Fille
38	Pile	Fille
39	Face	Fille
40	Face	Gars
41	Pile	Fille
42	Pile	Gars
43	Pile	Gars
44	Pile	Gars
45	Face	Fille
46	Pile	Fille
47	Face	Fille
48	Face	Fille
49	Pile	Gars
50	Face	Gars
51	Face	Fille

```
52      Face  Gars
53      Pile  Gars
54      Face Fille
55      Pile Fille
56      Face  Gars
57      Face Fille
58      Pile Fille
59      Face Fille
60      Face Fille
```

## Les data frame

Voici certaines fonctions très utiles pour travailler avec les *data frame*, illustrées ici en utilisant le jeu de données contenu dans la variable `df`.

```
names(df) # Affiche les noms de colonnes (des variables)
```

```
[1] "couleur_yeux"      "taille_cm"         "nombre_choisi"
[4] "cours_par_session" "peur_des_biostats" "piece_monnaie"
[7] "genre"
```

```
str(df) # Affiche la structure du data frame; une sorte de résumé des variables
```

```
'data.frame':  60 obs. of  7 variables:
 $ couleur_yeux      : chr  "Noir" "Bleu" "Noisette" "Autre" ...
 $ taille_cm        : num  174 168 158 177 152 177 163 186 150 170 ...
 $ nombre_choisi     : int   92 20 11 99 87 7 25 27 87 6 ...
 $ cours_par_session: int    6 5 5 5 5 5 5 4 5 5 ...
 $ peur_des_biostats: chr   "Non" "Pas_sur" "Pas_sur" "Non" ...
 $ piece_monnaie     : chr   "Face" "Face" "Face" "Face" ...
 $ genre            : chr   "Fille" "Fille" "Fille" "Gars" ...
```

```
nrow(df) # Nombre de lignes (observations)
```

```
[1] 60
```

```
ncol(df) # Nombre de colonnes (variables)
```

```
[1] 7
```

```
head(df) # Affiche les 6 premières lignes (observations)
```

```

couleur_yeux taille_cm nombre_choisi cours_par_session peur_des_biostats
1      Noir      174      92      6      Non
2      Bleu      168      20      5      Pas_sur
3      Noisette  158      11      5      Pas_sur
4      Autre     177      99      5      Non
5      Noisette  152      87      5      Pas_sur
6      Bleu      177      7      5      Non
piece_monnaie genre
1      Face Fille
2      Face Fille
3      Face Fille
4      Face Gars
5      Pile Fille
6      Pile Gars

```

```
tail(df) # Affiche les 6 dernières lignes (observations)
```

```

couleur_yeux taille_cm nombre_choisi cours_par_session peur_des_biostats
55      Bleu      180      14      4      Pas_sur
56      Noir      177      6      5      Oui
57      Bleu      160      34     4      Oui
58      Bleu      165      26     4      Pas_sur
59      Noir      165      12     5      Pas_sur
60      Noisette  157      66     4      Pas_sur
piece_monnaie genre
55      Pile Fille
56      Face Gars
57      Face Fille
58      Pile Fille
59      Face Fille
60      Face Fille

```

Il est important de comprendre que chaque variable (colonne) d'un data frame est en fait un vecteur de données. Pour accéder à ces vecteurs, il suffit d'utiliser le signe de dollar \$ avec le nom du data frame et de la variable. Dans l'exemple suivant, on extrait la variable (vecteur) `taille_cm` du data frame `df`.

```
df$taille_cm
```

```

[1] 174.00 168.00 158.00 177.00 152.00 177.00 163.00 186.00 150.00 170.00
[11] 165.00 180.00 153.00 177.00  1.53 156.00 152.00 165.00 168.00 158.00
[21] 173.00 177.00 164.00 165.54 170.00 155.00 195.00 174.00 183.00 164.00
[31] 153.00 173.00 161.00 170.00 164.00 152.00 170.00 170.00 159.00 182.00

```

```
[41] 161.00 186.00 168.00 175.00 162.00 164.00 195.68 155.00 182.00 180.00
[51] 166.00 173.00 177.00 161.00 180.00 177.00 160.00 165.00 165.00 157.00
```

#### Tip

Dans RStudio, une fois que vous avez tapé le nom du *data frame* et le signe dollar (`df$`), vous pouvez appuyer sur la touche tabulation (TAB; `-> |`) de votre clavier pour afficher la liste complète des variables contenues dans le data frame.

Puisque chaque colonne est un vecteur, on peut utiliser plusieurs fonctions que vous devriez déjà connaître<sup>3</sup>.

```
class(df$taille_cm) # Quelle est la classe du vecteur
```

```
[1] "numeric"
```

```
length(df$taille_cm) # Quelle est la longueur du vecteur
```

```
[1] 60
```

```
mean(df$taille_cm) # Quelle est la moyenne du vecteur
```

```
[1] 165.5792
```

Il est facile de créer de nouvelles variables dans un *data frame* en utilisant l'opérateur `$`. Dans l'exemple suivant, une nouvelle variable `taille_in` représentant la taille en pouces (*inches* en anglais) est créée:

```
# Conversion des centimètres en pouces
df$taille_po <- df$taille_cm * 0.393701

# Afficher les noms de variables dans df (notez la nouvelle colonne)
names(df)
```

```
[1] "couleur_yeux"      "taille_cm"         "nombre_choisi"
[4] "cours_par_session" "peur_des_biostats" "piece_monnaie"
[7] "genre"             "taille_po"
```

---

<sup>3</sup>Voir la capsule *Les fonctions dans R* pour une courte liste des fonctions essentielles. Voir aussi la notice à télécharger.

```
# Afficher le data frame avec la nouvelle variable (cliquez sur les flèches pour  
# naviguer dans les colonnes).
```

```
df
```

	couleur_yeux	taille_cm	nombre_choisi	cours_par_session	peur_des_biostats
1	Noir	174.00	92	6	Non
2	Bleu	168.00	20	5	Pas_sur
3	Noisette	158.00	11	5	Pas_sur
4	Autre	177.00	99	5	Non
5	Noisette	152.00	87	5	Pas_sur
6	Bleu	177.00	7	5	Non
7	Bleu	163.00	25	5	Pas_sur
8	Noir	186.00	27	4	Non
9	Bleu	150.00	87	5	Non
10	Noisette	170.00	6	5	Non
11	Noir	165.00	88	5	Pas_sur
12	Bleu	180.00	6	3	Oui
13	Bleu	153.00	42	4	Pas_sur
14	Bleu	177.00	9	5	Non
15	Bleu	1.53	8	5	Pas_sur
16	Bleu	156.00	7	5	Pas_sur
17	Autre	152.00	27	4	Pas_sur
18	Noisette	165.00	20	4	Oui
19	Noisette	168.00	7	5	Non
20	Noisette	158.00	9	5	Oui
21	Noir	173.00	8	5	Oui
22	Autre	177.00	0	4	Pas_sur
23	Noisette	164.00	1	5	Pas_sur
24	Autre	165.54	11	5	Non
25	Noisette	170.00	4	5	Pas_sur
26	Bleu	155.00	22	5	Non
27	Autre	195.00	69	5	Non
28	Autre	174.00	66	4	Pas_sur
29	Noisette	183.00	18	5	Pas_sur
30	Autre	164.00	66	4	Pas_sur
31	Bleu	153.00	1	5	Non
32	Bleu	173.00	13	4	Pas_sur
33	Bleu	161.00	4	5	Non
34	Autre	170.00	14	4	Non
35	Autre	164.00	16	5	Pas_sur
36	Noisette	152.00	7	4	Non
37	Bleu	170.00	4	5	Oui
38	Noisette	170.00	72	4	Pas_sur
39	Bleu	159.00	18	4	Pas_sur
40	Bleu	182.00	42	5	Pas_sur
41	Autre	161.00	44	5	Non
42	Noisette	186.00	5	5	Non

43	Bleu	168.00	69	5	Non
44	Bleu	175.00	1	6	Pas_sur
45	Bleu	162.00	24	5	Non
46	Autre	164.00	29	5	Pas_sur
47	Bleu	195.68	16	5	Pas_sur
48	Bleu	155.00	22	5	Oui
49	Noir	182.00	42	4	Non
50	Noisette	180.00	2	4	Oui
51	Noisette	166.00	28	6	Non
52	Noisette	173.00	94	4	Oui
53	Autre	177.00	7	5	Non
54	Noir	161.00	5	4	Pas_sur
55	Bleu	180.00	14	4	Pas_sur
56	Noir	177.00	6	5	Oui
57	Bleu	160.00	34	4	Oui
58	Bleu	165.00	26	4	Pas_sur
59	Noir	165.00	12	5	Pas_sur
60	Noisette	157.00	66	4	Pas_sur

	piece_monnaie	genre	taille_po
1	Face	Fille	68.5039740
2	Face	Fille	66.1417680
3	Face	Fille	62.2047580
4	Face	Gars	69.6850770
5	Pile	Fille	59.8425520
6	Pile	Gars	69.6850770
7	Pile	Fille	64.1732630
8	Pile	Gars	73.2283860
9	Face	Gars	59.0551500
10	Pile	Fille	66.9291700
11	Pile	Fille	64.9606650
12	Pile	Gars	70.8661800
13	Face	Fille	60.2362530
14	Pile	Fille	69.6850770
15	Pile	Autre	0.6023625
16	Pile	Fille	61.4173560
17	Pile	Fille	59.8425520
18	Pile	Fille	64.9606650
19	Face	Fille	66.1417680
20	Pile	Fille	62.2047580
21	Pile	Fille	68.1102730
22	Face	Gars	69.6850770
23	Pile	Fille	64.5669640
24	Pile	Fille	65.1732635
25	Face	Fille	66.9291700
26	Face	Fille	61.0236550
27	Face	Gars	76.7716950
28	Face	Fille	68.5039740
29	Pile	Gars	72.0472830

```
30      Pile Fille 64.5669640
31      Face Fille 60.2362530
32      Pile Fille 68.1102730
33      Pile Fille 63.3858610
34      Face Fille 66.9291700
35      Pile Fille 64.5669640
36      Pile Fille 59.8425520
37      Face Fille 66.9291700
38      Pile Fille 66.9291700
39      Face Fille 62.5984590
40      Face Gars 71.6535820
41      Pile Fille 63.3858610
42      Pile Gars 73.2283860
43      Pile Gars 66.1417680
44      Pile Gars 68.8976750
45      Face Fille 63.7795620
46      Pile Fille 64.5669640
47      Face Fille 77.0394117
48      Face Fille 61.0236550
49      Pile Gars 71.6535820
50      Face Gars 70.8661800
51      Face Fille 65.3543660
52      Face Gars 68.1102730
53      Pile Gars 69.6850770
54      Face Fille 63.3858610
55      Pile Fille 70.8661800
56      Face Gars 69.6850770
57      Face Fille 62.9921600
58      Pile Fille 64.9606650
59      Face Fille 64.9606650
60      Face Fille 61.8110570
```

### Exporter des données depuis R

Il est possible d'exporter un *data frame* depuis R vers un fichier texte sur votre ordinateur en utilisant la fonction `write.table()`. Les paramètres sont les mêmes que ceux de `read.table()` à l'exception de `x` qui correspond au data frame que l'on veut exporter. Dans l'exemple suivant, le data frame `df` est exporté dans un fichier nommé `sondage_cours_biostats.csv`.

```
write.table(
  x = df,
  file = "/home/user/Desktop/sondage_cours_biostats.csv",
  sep = ",",
  dec = "."
)
```