

Exercices d'initiation à la manipulation de données avec R

Gilles San Martin - Centre Wallon de Recherche Agronomique (CRA-W)

14 November 2015

Contents

Exercices de prise en main de R	2
Exercices de manipulation de formats de données	8
Exercice 1	8
Exercice 2	12
Exercices d'importation de données	16
Exercices subsidiaires (pas de R)	24
Exercices sur l'extraction de données (Subscripting)	25
Exercices de manipulation de caractères	56
Exercice 1	56
Exercice 2	64
Structures de contrôle : fonctions, boucles, exécutions conditionnelles,...	65
Exercice 0	65
Exercice 1	67
Exercice 2	68
Exercice 3	74
Exercice 4	76
Exercice 5	78
Agrégation et reshaping	83
Exercice 1	83
Exercice 2	92

Exercices de prise en main de R

Le but de cette série d'exercices est de se familiariser avec l'utilisation basique de R, d'acquérir le réflexe de chercher dans l'aide et de comprendre et utiliser la vectorisation. On vous demandera donc par moment d'utiliser des fonctions qui n'ont pas été vues dans la partie théorique. Pas de panique, cherchez simplement dans l'aide...

- 1) Calculer les racines carrées arrondies à 2 décimales de tous les nombres de 1 à 100 (fonctions round et sqrt)

```
round(sqrt(1:100), 2)
```

```
## [1] 1.00 1.41 1.73 2.00 2.24 2.45 2.65 2.83 3.00 3.16 3.32
## [12] 3.46 3.61 3.74 3.87 4.00 4.12 4.24 4.36 4.47 4.58 4.69
## [23] 4.80 4.90 5.00 5.10 5.20 5.29 5.39 5.48 5.57 5.66 5.74
## [34] 5.83 5.92 6.00 6.08 6.16 6.24 6.32 6.40 6.48 6.56 6.63
## [45] 6.71 6.78 6.86 6.93 7.00 7.07 7.14 7.21 7.28 7.35 7.42
## [56] 7.48 7.55 7.62 7.68 7.75 7.81 7.87 7.94 8.00 8.06 8.12
## [67] 8.19 8.25 8.31 8.37 8.43 8.49 8.54 8.60 8.66 8.72 8.77
## [78] 8.83 8.89 8.94 9.00 9.06 9.11 9.17 9.22 9.27 9.33 9.38
## [89] 9.43 9.49 9.54 9.59 9.64 9.70 9.75 9.80 9.85 9.90 9.95
## [100] 10.00
```

```
round((1:100)^0.5, 2) # résultat identique
```

```
## [1] 1.00 1.41 1.73 2.00 2.24 2.45 2.65 2.83 3.00 3.16 3.32
## [12] 3.46 3.61 3.74 3.87 4.00 4.12 4.24 4.36 4.47 4.58 4.69
## [23] 4.80 4.90 5.00 5.10 5.20 5.29 5.39 5.48 5.57 5.66 5.74
## [34] 5.83 5.92 6.00 6.08 6.16 6.24 6.32 6.40 6.48 6.56 6.63
## [45] 6.71 6.78 6.86 6.93 7.00 7.07 7.14 7.21 7.28 7.35 7.42
## [56] 7.48 7.55 7.62 7.68 7.75 7.81 7.87 7.94 8.00 8.06 8.12
## [67] 8.19 8.25 8.31 8.37 8.43 8.49 8.54 8.60 8.66 8.72 8.77
## [78] 8.83 8.89 8.94 9.00 9.06 9.11 9.17 9.22 9.27 9.33 9.38
## [89] 9.43 9.49 9.54 9.59 9.64 9.70 9.75 9.80 9.85 9.90 9.95
## [100] 10.00
```

```
round(sqrt(seq(1,100,1)), 2) # résultat identique
```

```
## [1] 1.00 1.41 1.73 2.00 2.24 2.45 2.65 2.83 3.00 3.16 3.32
## [12] 3.46 3.61 3.74 3.87 4.00 4.12 4.24 4.36 4.47 4.58 4.69
## [23] 4.80 4.90 5.00 5.10 5.20 5.29 5.39 5.48 5.57 5.66 5.74
## [34] 5.83 5.92 6.00 6.08 6.16 6.24 6.32 6.40 6.48 6.56 6.63
## [45] 6.71 6.78 6.86 6.93 7.00 7.07 7.14 7.21 7.28 7.35 7.42
## [56] 7.48 7.55 7.62 7.68 7.75 7.81 7.87 7.94 8.00 8.06 8.12
## [67] 8.19 8.25 8.31 8.37 8.43 8.49 8.54 8.60 8.66 8.72 8.77
## [78] 8.83 8.89 8.94 9.00 9.06 9.11 9.17 9.22 9.27 9.33 9.38
## [89] 9.43 9.49 9.54 9.59 9.64 9.70 9.75 9.80 9.85 9.90 9.95
## [100] 10.00
```

- 2) Calculer la surface des cercles de rayon variant de 0 à 250 cm par pas de 10 cm (26 cercles donc) (fonction seq)

```
rayon <- seq(0, 250, 10)
surface <- pi*rayon^2
surface
```

```
## [1] 0.0000 314.1593 1256.6371 2827.4334 5026.5482
## [6] 7853.9816 11309.7336 15393.8040 20106.1930 25446.9005
## [11] 31415.9265 38013.2711 45238.9342 53092.9158 61575.2160
## [16] 70685.8347 80424.7719 90792.0277 101787.6020 113411.4948
## [21] 125663.7061 138544.2360 152053.0844 166190.2514 180955.7368
## [26] 196349.5408
```

- 3) Ecrivez les phrases suivantes : “Un cercle de 0 cm de rayon a une surface de 0 m²”, “Un cercle de 10 cm de rayon a une surface de 0.03 m²”, etc... pour tous les cercles de rayon variant de 0 à 250 cm par pas de 10 cm (fonction paste)

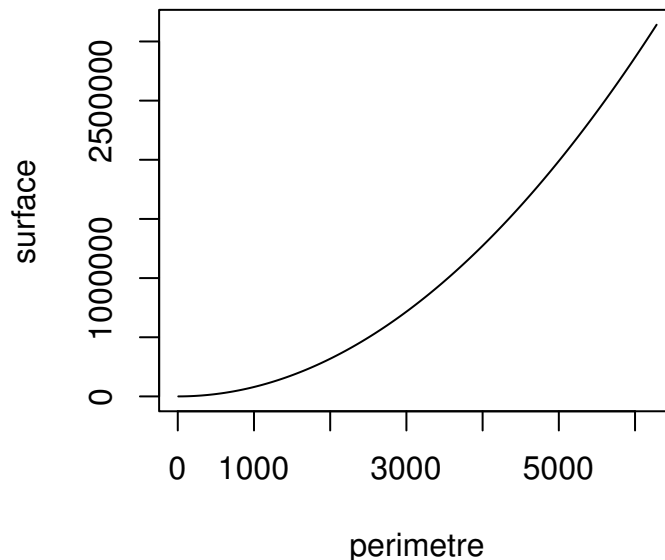
```
paste("Un cercle de ", rayon, " cm de rayon a une surface de ", round(surface/10000, 2),  
      " m2", sep= " ")
```

```
## [1] "Un cercle de 0 cm de rayon a une surface de 0 m2"  
## [2] "Un cercle de 10 cm de rayon a une surface de 0.03 m2"  
## [3] "Un cercle de 20 cm de rayon a une surface de 0.13 m2"  
## [4] "Un cercle de 30 cm de rayon a une surface de 0.28 m2"  
## [5] "Un cercle de 40 cm de rayon a une surface de 0.5 m2"  
## [6] "Un cercle de 50 cm de rayon a une surface de 0.79 m2"  
## [7] "Un cercle de 60 cm de rayon a une surface de 1.13 m2"  
## [8] "Un cercle de 70 cm de rayon a une surface de 1.54 m2"  
## [9] "Un cercle de 80 cm de rayon a une surface de 2.01 m2"  
## [10] "Un cercle de 90 cm de rayon a une surface de 2.54 m2"  
## [11] "Un cercle de 100 cm de rayon a une surface de 3.14 m2"  
## [12] "Un cercle de 110 cm de rayon a une surface de 3.8 m2"  
## [13] "Un cercle de 120 cm de rayon a une surface de 4.52 m2"  
## [14] "Un cercle de 130 cm de rayon a une surface de 5.31 m2"  
## [15] "Un cercle de 140 cm de rayon a une surface de 6.16 m2"  
## [16] "Un cercle de 150 cm de rayon a une surface de 7.07 m2"  
## [17] "Un cercle de 160 cm de rayon a une surface de 8.04 m2"  
## [18] "Un cercle de 170 cm de rayon a une surface de 9.08 m2"  
## [19] "Un cercle de 180 cm de rayon a une surface de 10.18 m2"  
## [20] "Un cercle de 190 cm de rayon a une surface de 11.34 m2"  
## [21] "Un cercle de 200 cm de rayon a une surface de 12.57 m2"  
## [22] "Un cercle de 210 cm de rayon a une surface de 13.85 m2"  
## [23] "Un cercle de 220 cm de rayon a une surface de 15.21 m2"  
## [24] "Un cercle de 230 cm de rayon a une surface de 16.62 m2"  
## [25] "Un cercle de 240 cm de rayon a une surface de 18.1 m2"  
## [26] "Un cercle de 250 cm de rayon a une surface de 19.63 m2"
```

- 4) Tracer un graphique montrant la relation entre le périmètre et la surface d'un cercle. Ajoutez un titre explicite au graphique

```
rayon <- 1:1000  
surface <- pi*rayon^2  
perimetre <- 2*pi*rayon  
plot(surface ~ perimetre, type="l",  
      main = "Relation entre la surface et \nle périmètre d'un cercle")
```

Relation entre la surface et le périmètre d'un cercle



NB à la place du paramètre graphique "main", on peut aussi utiliser la fonction "title"

- 5) Utilisez les 3 vecteurs ci-dessous : “jour”, “mois”, “année” pour créer un vecteur nommé “date” sous la forme 28/9/2012. (fonction paste) Essayez de comprendre comment sont construits ces 3 vecteurs (regardez l’aide des fonctions utilisées: floor, runif, set.seed).

```
set.seed(123)
jour <- floor(runif(30, 1, 31))
mois <- floor(runif(30, 1, 12))
année <- floor(runif(30, 1900, 2012))
```

```
date <- paste(jour, mois, année, sep="/")
date
```

```
## [1] "9/11/1974" "24/10/1910" "13/8/1943" "27/9/1930" "29/1/1991"
## [6] "2/6/1950" "16/9/1990" "27/3/1990" "17/4/1988" "14/3/1949"
## [11] "29/2/1984" "14/5/1970" "21/5/1979" "18/5/1900" "4/2/1953"
## [16] "27/2/1924" "8/3/1942" "2/6/1968" "10/3/1939" "29/10/1912"
## [21] "27/1/1927" "21/5/1974" "20/9/1946" "30/2/1988" "20/7/1911"
## [26] "22/3/1948" "17/2/2010" "18/9/2000" "9/10/1999" "5/5/1919"
```

*# On a utilisé la fonction runif qui génère des nombres aléatoire selon une distribution
uniforme. Pour les 3 vecteurs on demande à la fonction de générer 30 de ces nombres.
On génère des nombres entre 1 et 30 pour les jours, entre 1 et 12 pour les mois et entre
1900 et 2012 pour les années.
Pour obtenir des nombres entiers (pour éliminer les décimales) on utilise la fonction
floor qui garde uniquement la partie entière des nombres (sans les arrondir).
La fonction set.seed sert simplement à ce que les nombres aléatoires générés soient
toujours les mêmes.*

- 6) Créez 2 variables aléatoires de distribution normale (fonction rnorm) ayant chacune une moyenne différente. Réalisez ensuite un test t de student pour comparer les moyennes de ces deux variables (cherchez dans l’aide la

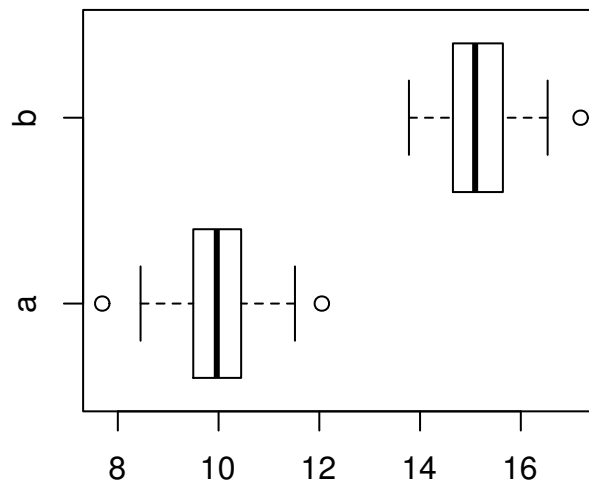
fonction adéquate...).

Faites un boxplot horizontal de ces deux variables sur le même graphique. Vous devez pour ce faire d'abord les coller côte à côte dans un même objet avec la fonction cbind (par exemple cbind(a,b))

```
a <- rnorm(n=30, mean= 10, sd=1)
b <- rnorm(30, 15)
# Taper ??student pour trouver dans l'aide la fonction faisant le test de t
# (ou cherchez sur internet)
t.test(a, b)

##
## Welch Two Sample t-test
##
## data: a and b
## t = -23.068, df = 57.097, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -5.655180 -4.751822
## sample estimates:
## mean of x mean of y
## 9.965332 15.168834

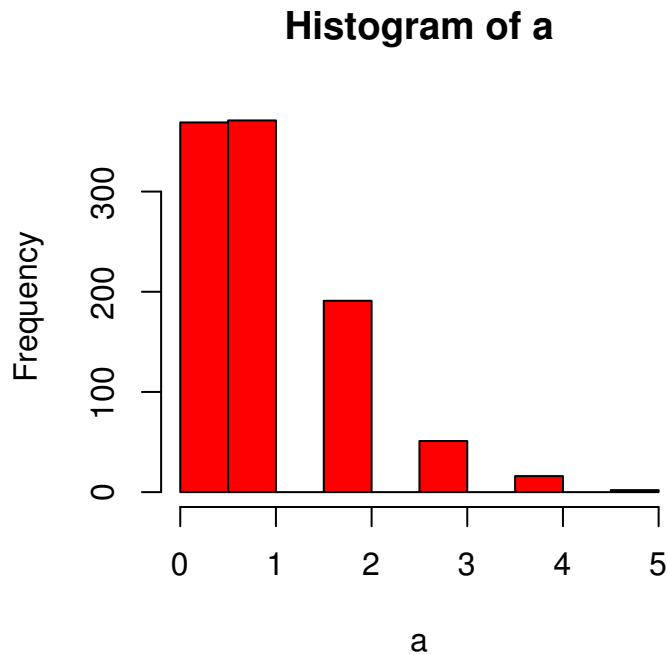
# Taper ?boxplot pour voir les différents paramètres possibles pour cette fonction et
# notamment celui qui permet de tracer des boxplots horizontaux.
boxplot(cbind(a,b), horizontal =TRUE)
```



- 7) Créez une variable aléatoire avec une distribution de Poisson (une fonction similaire à rnorm et runif). Tracez un histogramme de cette variable de couleur rouge.

```
# La seule difficulté ici est de trouver la bonne fonction en utilisant l'aide.
# Voir par exemple l'aide de ?rnorm : dans la section "see also" il y a un lien vers
```

```
# toutes les distributions fournies par R
a <- rpois(1000, 1)
hist(a, col = "red")
```



8) Voici deux vecteurs représentant le nombre de mâles et de femelles observées dans une population et dans 5 catégories de couleurs : noir, jaune, rouge, bleu, blanc.

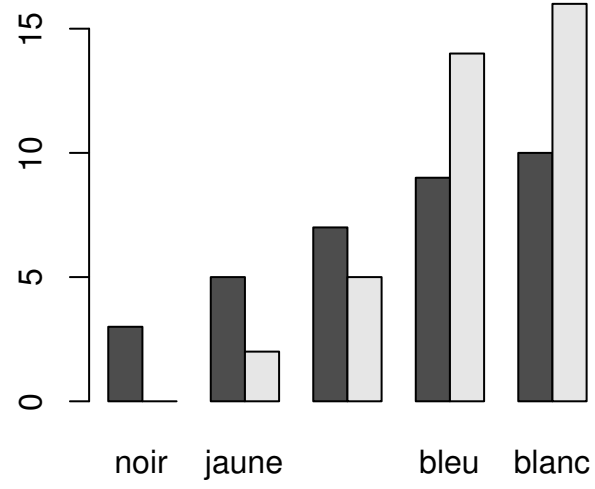
Tracez un barplot représentant côte à côte les valeurs des mâles et des femelles pour chaque couleur (et avec le nom des couleurs dans les étiquettes de l'axe x).

Pour ce faire commencez par coller les 2 vecteurs de manière à former une matrice à 2 lignes et 5 colonnes (fonction rbind). Ajoutez les 5 couleurs dans les noms de colonnes (fonction colnames) et ensuite, tracez le graphique.

```
males <- c(3, 5, 7, 9, 10)
femelles <- c(0, 2, 5, 14, 16)
```

```
mat <- rbind(males, femelles)
colnames(mat) <- c("noir", "jaune", "rouge", "bleu", "blanc")

barplot(mat, beside = TRUE)
```



Exercices de manipulation de formats de données

Exercice 1

L'exercice suivant n'est pas très réaliste. Le but est de vous faire manipuler dans des cas simples les formats de données. Voici 2 vecteurs de 10 nombres a (10, 20, 30, etc) et b (5, 5, 5, etc).

```
a <- seq(10,100,10)
b <- rep(c(5,10), c(7,3))
```

- 1) Faites la somme de chaque vecteur (somme a = 550, somme b = 65)

```
sum(a)
```

```
## [1] 550
```

```
sum(b)
```

```
## [1] 65
```

- 2) Concaténez ces deux vecteurs l'un à la suite de l'autre de façon à obtenir un vecteur de longueur 20 et dont la somme vaut 615

```
c(a,b)
```

```
## [1] 10 20 30 40 50 60 70 80 90 100 5 5 5 5 5 5 5
## [18] 10 10 10
```

```
length(c(a,b))
```

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

```
sum(c(a,b))
```

```
## [1] 615
```

- 3) Collez les nombres de ces vecteurs a et b 2 à 2 (pour obtenir 105, 205, 305 etc) et sauvez le résultat dans un nouveau vecteur "ab". Ce vecteur a une longueur de 10 éléments. Quel est son mode ?

```
ab <- paste(a, b, sep="")
ab
```

```
## [1] "105" "205" "305" "405" "505" "605" "705" "8010"
## [9] "9010" "10010"
```

```
mode(ab)
```

```
## [1] "character"
```

- 4) Faites la somme de ce nouveau vecteur (réponse = 29865).


```
sum(ab) # ne fonctionne pas
```

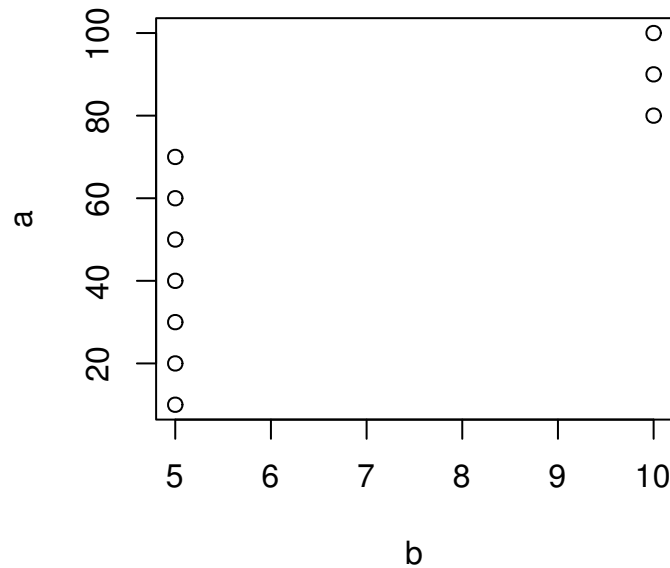
```
## Error in sum(ab): 'type' (character) de l'argument incorrect
```

```
sum(as.numeric(ab))
```

```
## [1] 29865
```

- 5) Faites un graphique de a en fonction de b (plot(a~b))

```
plot(a~b)
```

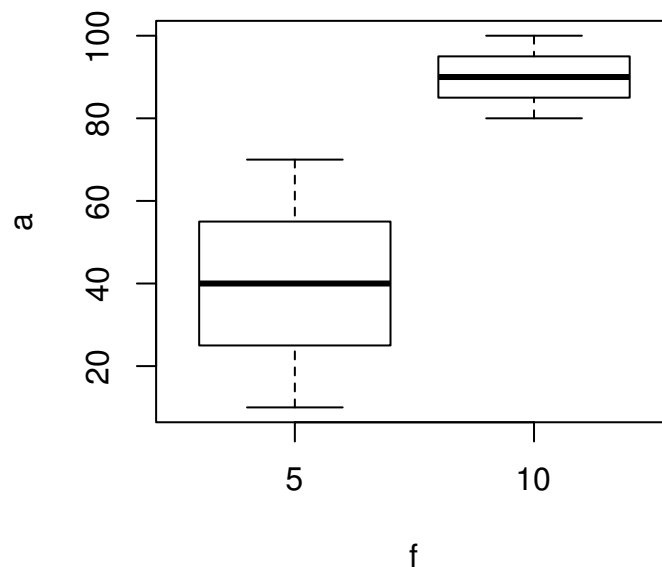


- 6) Transformez le vecteur b en facteur et sauvez ce résultat dans un nouveau vecteur appelé “f”

```
f <- as.factor(b)
```

- 7) Faites un graphique de a en fonction de f

```
plot(a~f)
```



- 8) Faites la somme de f (ça ne fonctionne pas...)

```
sum(f) # ne fonctionne pas
```

```
## Error in Summary.factor(structure(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, : 'sum' not meaningful for factors
```

- 9) Retransformez f en numérique pour pouvoir en faire la somme (vous devrez passer par un format intermédiaire avant de le transformer en numérique). Vous devriez obtenir le même résultat que sum(b) soit 65 (et pas 13).

```
sum(as.numeric(f)) # ne donne pas la bonne réponse
```

```
## [1] 13
```

```
sum(as.numeric(as.character(f)))
```

```
## [1] 65
```

- 10) A partir du vecteur a, créez une matrice de 5 colonnes et 2 lignes avec sur la première ligne les valeurs 10, 20, 30, etc... (et pas 10, 30, 50 etc...). Sauvez cette matrice dans un objet appelé "mat".

```
matrix(a, ncol = 5, nrow=2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  10  30  50  70  90
## [2,]  20  40  60  80 100
```

```
mat <- matrix(a, ncol = 5, nrow=2, byrow=TRUE)
mat
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  10  20  30  40  50
## [2,]  60  70  80  90 100
```

- 11) Ajoutez comme nom de colonnes col1, col2, col3 etc... à cette matrice.

```
colnames(mat) <- c("col1", "col2", "col3", "col4", "col5")
colnames(mat) <- paste0("col", 1:ncol(mat)) # même résultat mais plus flexible
```

- 12) Au moyen de l'opérateur \$ essayez d'extraire la colonne 2 (mat\$col2). Que faut-il faire pour que cela fonctionne ? (dans quel format faut-il transformer mat?)

```
mat$col2 # ne fonctionne pas car $ ne fonctionne qu'avec des listes ou des data.frame
```

```
## Error in mat$col2: $ operator is invalid for atomic vectors
```

```
mat <- as.data.frame(mat)
mat$col2
```

```
## [1] 20 70
```

- 13) Créez un objet à 2 colonnes et 3 lignes contenant dans la première colonne les valeurs A, B et C et dans la deuxième colonne 3, 6 et 9.

```
a <- c("A", "B", "C")
b <- c(3,6,9)
df <- data.frame(a=a, b=b)
df
```

```
##   a b
## 1 A 3
## 2 B 6
## 3 C 9
```

- 14) Faites la somme de la deuxième colonne de l'objet ainsi créé

```
sum(df$b)
```

```
## [1] 18
```

```
# On aurait pu être tenté de procéder comme suit mais ça ne fonctionne pas car si on
# mélange du texte et des nombres dans une matrice, les nombres sont transformés en texte
mat <- matrix(c("A", "B", "C", 3,6,9), 3, 2)
colnames(mat) <- c("a", "b")
mat <- as.data.frame(mat)
sum(mat$b)
```

```
## Error in Summary.factor(structure(1:3, .Label = c("3", "6", "9"), class = "factor"), : 'sum' not meaningful for factor
```

```
# on peut par contre retransformer cette colonne en numérique à la volée :
sum(as.numeric(mat$b))
```

```
## [1] 6
```

Exercice 2

Dans cet exercice plus réaliste, vous devrez à de nombreuses reprises passer d'un format de données à un autre (dates, character, numeric,...) pour pouvoir effectuer les opérations demandées.

Chargez le fichier "biche.txt" avec la commande suivante : `d <- read.table("biche.txt")` après avoir défini votre répertoire de travail où se trouve le fichier "biche.txt" avec la fonction `setwd`, par exemple : `setwd("/home/gilles/exercices")` (NB : vous devez utiliser des slashes "/" et pas de backslashes "\" dans le chemin du répertoire).

Ce fichier contient la date, l'heure et les points gps (coordonnées x y) de position d'une biche. Les colonnes `datet0`, `timet0`, `xt0`, `yt0`, contiennent pour chaque ligne les positions et temps de l'observation (point GPS) précédente. L'objectif général est de déterminer la période de l'année (saison) pour chaque point et de calculer la vitesse de déplacement entre deux points successifs.

- 1) Visualisez le contenu de votre jeu de données avec `summary(d)`, `head(d)` et `str(d)`. Identifiez le type de chaque variable.

```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
d <- read.table("data/biche.txt")
summary(d)
```

```
##           id           date           time           x
## Min.      : 0   5/02/2012 : 86   0:00:54 : 48   Min.    :284406
## 1st Qu.:1106 29/10/2011: 80   20:00:54: 40   1st Qu.:285488
## Median :2212 15/10/2011: 79   4:00:54 : 36   Median :285975
## Mean   :2212 6/02/2012 : 78   0:00:55 : 31   Mean   :285882
## 3rd Qu.:3319 21/05/2011: 77   20:00:53: 30   3rd Qu.:286293
## Max.    :4425 9/02/2011 : 76   0:00:53 : 27   Max.    :287348
##           (Other) :3950   (Other) :4214
##           y           datet0           timet0           xt0
## Min.    :128157   5/02/2012 : 86   0:00:54 : 48   Min.    :284406
## 1st Qu.:132342   29/10/2011: 80   20:00:54: 40   1st Qu.:285488
## Median :132588   15/10/2011: 79   4:00:54 : 36   Median :285975
## Mean   :132540   6/02/2012 : 78   0:00:55 : 31   Mean   :285882
## 3rd Qu.:132905   21/05/2011: 77   20:00:53: 30   3rd Qu.:286293
## Max.    :133691   (Other)    :4025   (Other) :4240   Max.    :287348
##           NA's      : 1   NA's      : 1   NA's      :1
##           yt0
## Min.    :128157
## 1st Qu.:132342
## Median :132587
## Mean   :132540
## 3rd Qu.:132904
## Max.    :133691
## NA's    :1
```

```
head(d)
```

```
##   id   date   time   x   y   datet0   timet0   xt0   yt0
## 1  0 1/02/2011 20:13:23 286202 132124      <NA>      <NA>    NA    NA
## 2  1 2/02/2011 0:01:26 286340 132181 1/02/2011 20:13:23 286202 132124
## 3  2 2/02/2011 8:01:23 286586 132035 2/02/2011 0:01:26 286340 132181
## 4  3 2/02/2011 12:01:42 286581 132024 2/02/2011 8:01:23 286586 132035
## 5  4 2/02/2011 16:01:24 286323 132248 2/02/2011 12:01:42 286581 132024
## 6  5 3/02/2011 8:01:41 286315 132246 2/02/2011 16:01:24 286323 132248
```

- 2) Passez les colonnes `time` et `timet0` en format date-temps (fonction `strptime`). Vous devrez au préalable coller la date et le temps dans une même variable (fonction `paste`). Utilisez la fonction `summary` pour visualiser la transformation du type de données

```
d$time <- paste(d$date, d$time)
d$time <- strptime(d$time, format = "%d/%m/%Y %H:%M:%S")

d$timet0 <- paste(d$datet0, d$timet0)
d$timet0 <- strptime(d$timet0, format = "%d/%m/%Y %H:%M:%S")

summary(d)
```

```
##           id           date           time
## Min.      : 0    5/02/2012 : 86    Min.      :2011-02-01 20:13:23
## 1st Qu.:1106   29/10/2011: 80    1st Qu.:2011-03-16 15:04:42
## Median :2212   15/10/2011: 79    Median :2011-07-26 12:09:14
## Mean    :2212   6/02/2012 : 78    Mean    :2011-07-30 00:52:05
## 3rd Qu.:3319   21/05/2011: 77    3rd Qu.:2011-11-17 00:26:31
## Max.     :4425   9/02/2011 : 76    Max.     :2012-03-29 04:00:55
##          (Other) :3950
##           x           y           datet0
## Min.     :284406   Min.     :128157   5/02/2012 : 86
## 1st Qu.:285488   1st Qu.:132342   29/10/2011: 80
## Median :285975   Median :132588   15/10/2011: 79
## Mean     :285882   Mean     :132540   6/02/2012 : 78
## 3rd Qu.:286293   3rd Qu.:132905   21/05/2011: 77
## Max.     :287348   Max.     :133691   (Other)    :4025
##                               NA's      : 1
##           timet0           xt0           yt0
## Min.      :2011-02-01 20:13:23   Min.      :284406   Min.      :128157
## 1st Qu.:2011-03-16 15:00:53   1st Qu.:285488   1st Qu.:132342
## Median :2011-07-26 12:02:49   Median :285975   Median :132587
## Mean     :2011-07-29 23:32:58   Mean     :285882   Mean     :132540
## 3rd Qu.:2011-11-17 00:15:16   3rd Qu.:286293   3rd Qu.:132904
## Max.     :2012-03-29 00:00:55   Max.     :287348   Max.     :133691
## NA's      :1                   NA's      :1       NA's      :1
```

- 3) Créez une colonne pour l'année et le mois (par exemple avec la fonction `strptime`). Pour rappel, vous pouvez ajouter une colonne "mois" au dataframe "d" par exemple comme ceci : `d$mois <- c(1, 3, 5, 7)`
Utilisez la fonction `head` pour visualiser les premières lignes du jeu de données

```
d$year <- as.numeric(strptime(d$time, "%Y"))
d$month <- as.numeric(strptime(d$time, "%m"))
head(d)
```

```
##    id    date           time    x    y    datet0
## 1  0 1/02/2011 2011-02-01 20:13:23 286202 132124    <NA>
## 2  1 2/02/2011 2011-02-02 00:01:26 286340 132181 1/02/2011
## 3  2 2/02/2011 2011-02-02 08:01:23 286586 132035 2/02/2011
## 4  3 2/02/2011 2011-02-02 12:01:42 286581 132024 2/02/2011
## 5  4 2/02/2011 2011-02-02 16:01:24 286323 132248 2/02/2011
## 6  5 3/02/2011 2011-02-03 08:01:41 286315 132246 2/02/2011
##           timet0    xt0    yt0 year month
## 1           <NA>     NA     NA 2011     2
## 2 2011-02-01 20:13:23 286202 132124 2011     2
## 3 2011-02-02 00:01:26 286340 132181 2011     2
## 4 2011-02-02 08:01:23 286586 132035 2011     2
## 5 2011-02-02 12:01:42 286581 132024 2011     2
## 6 2011-02-02 16:01:24 286323 132248 2011     2
```

- 4) Sur base de votre colonne "mois", créez une colonne "saison" divisée en 3 périodes : nourrissage de janvier à avril (inclus), "été" de mai à septembre (inclus) et "chasse" de octobre à décembre.
Vous pouvez utiliser la fonction `cut` pour ce faire. Faites un `summary()` de cette nouvelle colonne. Vous devriez avoir 2192 données en période de nourrissage, 969 en été et 1265 en période de chasse.

```
d$season <- cut(d$month, breaks = c(0,4,9,12), labels = c("nourrissage", "été", "chasse"))
summary(d$season)
```

```
## nourrissage      été      chasse
##           2192      969      1265
```

NB on aurait pu utiliser ici la fonction cut sur une date au format Date (cfr ?cut.Date)

- 5) Calculez le temps écoulé entre 2 points gps successifs (2 lignes successives).
Lorsque vous faites la différence entre deux heures (classe `POSIXct`), Vous obtenez une variable au format "difftime". Vous pouvez voir les unités utilisées au moyen de la fonction `units`. Voir l'aide pour `difftime` pour plus d'infos. Les 3 premières valeurs sont : NA, 3.800833 et 7.999167 heures.

```
d$deltatime <- d$time - d$timet0
units(d$deltatime)
```

```
## [1] "mins"
```

```
d$deltatime <- as.numeric(d$deltatime)/60
head(d$deltatime)
```

```
## [1]      NA  3.800833  7.999167  4.005278  3.995000 16.004722
```

- 6) Calculez pareillement la distance parcourue. Les coordonnées x et y sont des lambert belges (coordonnées projetées) en mètres.
Pour Rappel, le théorème de Pythagore pour un triangle rectangle : $\text{Hypoténuse}^2 = \text{base}^2 + \text{hauteur}^2$
Les 3 premières valeurs sont NA, 0.149308406 et 0.286062930 km

```
d$dist <- sqrt((d$x - d$xt0)^2 + (d$y - d$yt0)^2)/1000
head(d$dist)
```

```
## [1]      NA  0.149308406  0.286062930  0.012083046  0.341672358  0.008246211
```

- 7) Calculez la vitesse en km/h entre deux points successifs (attentions aux unités). Quelle est la vitesse maximale ? (réponse : 3.312303 km/h et pas NA ! Regardez dans l'aide en cas de besoin...)

```
d$speed <- d$dist/d$deltatime
max(d$speed, na.rm = TRUE)
```

```
## [1] 3.312303
```

```
summary(d)
```

```
##      id      date      time
## Min.   :  0  5/02/2012 :  86  Min.   :2011-02-01 20:13:23
## 1st Qu.:1106 29/10/2011:  80  1st Qu.:2011-03-16 15:04:42
## Median :2212 15/10/2011:  79  Median :2011-07-26 12:09:14
## Mean   :2212  6/02/2012 :  78  Mean   :2011-07-30 00:52:05
## 3rd Qu.:3319 21/05/2011:  77  3rd Qu.:2011-11-17 00:26:31
## Max.   :4425  9/02/2011 :  76  Max.   :2012-03-29 04:00:55
##      (Other) :3950
##      x      y      datet0
## Min.   :284406  Min.   :128157  5/02/2012 :  86
```

```
## 1st Qu.:285488 1st Qu.:132342 29/10/2011: 80
## Median :285975 Median :132588 15/10/2011: 79
## Mean :285882 Mean :132540 6/02/2012 : 78
## 3rd Qu.:286293 3rd Qu.:132905 21/05/2011: 77
## Max. :287348 Max. :133691 (Other) :4025
## NA's : 1
## timet0 xt0 yt0
## Min. :2011-02-01 20:13:23 Min. :284406 Min. :128157
## 1st Qu.:2011-03-16 15:00:53 1st Qu.:285488 1st Qu.:132342
## Median :2011-07-26 12:02:49 Median :285975 Median :132587
## Mean :2011-07-29 23:32:58 Mean :285882 Mean :132540
## 3rd Qu.:2011-11-17 00:15:16 3rd Qu.:286293 3rd Qu.:132904
## Max. :2012-03-29 00:00:55 Max. :287348 Max. :133691
## NA's :1 NA's :1 NA's :1
## year month season deltatime
## Min. :2011 Min. : 1.000 nourrissage:2192 Min. : 0.0325
## 1st Qu.:2011 1st Qu.: 2.000 été : 969 1st Qu.: 0.2483
## Median :2011 Median : 5.000 chasse :1265 Median : 0.2572
## Mean :2011 Mean : 5.749 Mean : 2.2849
## 3rd Qu.:2011 3rd Qu.:10.000 3rd Qu.: 1.2492
## Max. :2012 Max. :12.000 Max. :96.0047
## NA's :1 NA's :1
## dist speed
## Min. :0.000000 Min. :0.000000
## 1st Qu.:0.009487 1st Qu.:0.02438
## Median :0.029732 Median :0.05494
## Mean :0.138638 Mean :0.12598
## 3rd Qu.:0.127012 3rd Qu.:0.13322
## Max. :2.565983 Max. :3.31230
## NA's :1 NA's :1
```

head(d)

```
## id date time x y datet0
## 1 0 1/02/2011 2011-02-01 20:13:23 286202 132124 <NA>
## 2 1 2/02/2011 2011-02-02 00:01:26 286340 132181 1/02/2011
## 3 2 2/02/2011 2011-02-02 08:01:23 286586 132035 2/02/2011
## 4 3 2/02/2011 2011-02-02 12:01:42 286581 132024 2/02/2011
## 5 4 2/02/2011 2011-02-02 16:01:24 286323 132248 2/02/2011
## 6 5 3/02/2011 2011-02-03 08:01:41 286315 132246 2/02/2011
## timet0 xt0 yt0 year month season deltatime
## 1 <NA> NA NA 2011 2 nourrissage NA
## 2 2011-02-01 20:13:23 286202 132124 2011 2 nourrissage 3.800833
## 3 2011-02-02 00:01:26 286340 132181 2011 2 nourrissage 7.999167
## 4 2011-02-02 08:01:23 286586 132035 2011 2 nourrissage 4.005278
## 5 2011-02-02 12:01:42 286581 132024 2011 2 nourrissage 3.995000
## 6 2011-02-02 16:01:24 286323 132248 2011 2 nourrissage 16.004722
## dist speed
## 1 NA NA
## 2 0.149308406 0.0392830710
## 3 0.286062930 0.0357615914
## 4 0.012083046 0.0030167810
## 5 0.341672358 0.0855249957
## 6 0.008246211 0.0005152361
```

Exercices d'importation de données

Le but de ces exercices est de se familiariser avec les cas les plus fréquents d'importation de données dans R (données dans un tableur ou dans un fichier texte). Les premiers exercices sont assez simples mais on a progressivement et volontairement introduit de nombreux pièges. L'objectif n'est point de jouer du plaisir sadique de vous voir patauger mais bien de vous confronter aux problèmes fréquents que l'on rencontre pour importer ses données. Quand vous devez importer un fichier texte, ouvrez le d'abord dans un bon éditeur de texte (par exemple avec Rstudio) et regardez comment il est structuré (est-ce qu'il y a des entêtes de colonne, des valeurs manquantes, comment sont séparés les champs, quels sont les séparateur de décimales, etc...).

Pour les fichiers xls, exportez les en format texte en respectant les recommandations vues dans la partie théorique. Après l'importation visualisez systématiquement votre jeu de données avec `summary()` et en imprimant le jeu de données.

Les deux derniers exercices sont subsidiaires et ne sont pas à proprement parler des exercices de R, l'essentiel du travail devant se faire dans le tableur. Ils montrent des fichiers Excel bien structurés et clairs mais assez mal adaptés à l'analyse et à l'importation dans R (ou tout autre logiciel). Le but est de montrer qu'il vaut mieux structurer ses fichiers de données brutes en réfléchissant au traitement des données.

- 1) Importez le fichier `data1.txt`. Calculez la moyenne de la variable "haraxy" (réponse : 5.85)

```
# Peu de difficultés ici. Il faut bien spécifier na.rm = TRUE dans la fonction mean sans  
# quoi elle retourne une valeur manquante NA  
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")  
mydata <- read.table("data/data1.txt", sep=";", header = TRUE)  
summary(mydata)
```

```
## tree   landsc   site   date   haraxy  
## L:27   R:27   L'île de Nivernais:18   Min.   :1.000   Min.   : 0.000  
## M:27   S:27   Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000  
## P:27   U:27   Mazée :12   Median :3.000   Median : 0.000  
## Nîmes :15   Mean :2.667   Mean : 5.856  
## Nivernais :12   3rd Qu.:3.000   3rd Qu.: 9.100  
## Olroy s/ Viroin : 9   Max. :4.000   Max. :53.600  
## Treignes : 3   NA's :2  
##  
## adabip   adadec   calqua   caldec  
## Min.   : 0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000  
## 1st Qu.: 6.10   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.000  
## Median : 31.80   Median : 6.70   Median : 0.000   Median : 0.000  
## Mean : 31.78   Mean :10.42   Mean : 7.192   Mean : 8.847  
## 3rd Qu.: 51.10   3rd Qu.:16.70   3rd Qu.: 7.450   3rd Qu.:13.800  
## Max. :100.00   Max. :50.00   Max. :60.000   Max. :66.700  
## NA's :2   NA's :2   NA's :2   NA's :2  
##  
## exoqua   halsed   oencon   harqua  
## Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00  
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00  
## Median : 0.000   Median : 0.000   Median : 0.000   Median : 0.00  
## Mean : 4.416   Mean : 3.099   Mean : 3.954   Mean :13.37  
## 3rd Qu.: 3.400   3rd Qu.: 0.000   3rd Qu.: 3.050   3rd Qu.:17.75  
## Max. :54.500   Max. :50.000   Max. :75.000   Max. :77.30  
## NA's :2   NA's :2   NA's :2   NA's :2  
##  
## myroct   aphobl   anaoce  
## Min.   : 0.000   Min.   : 0.000   Min.   :0.0000  
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000  
## Median : 0.000   Median : 0.000   Median :0.0000  
## Mean : 6.786   Mean : 3.778   Mean :0.4937  
## 3rd Qu.: 3.550   3rd Qu.: 0.000   3rd Qu.:0.0000  
## Max. :57.100   Max. :33.000   Max. :7.1000  
## NA's :2   NA's :2   NA's :2
```



```
mean(mydata$haraxy, na.rm = TRUE)
```

```
## [1] 5.855696
```

- 2) Importez le fichier data2.txt. Faites la moyenne de la variable “adabip” (=31.77) Utilisez la fonction plot pour faire un graphique de adabip en fonction de site : plot(adabip ~ site, data=mydata). Vous obtenez un boxplot. Faites de même pour adabip en fonction de date : plot(adabip ~ date, data=mydata). Vous n’obtenez pas un boxplot parce que date n’est pas dans le bon format. Transformez la date en facteur de façon à ce que: plot(adabip ~ date, data=mydata) vous donne un boxplot.

Il faut bien spécifier le type de séparateur, le caractère des décimales et des valeurs manquantes.

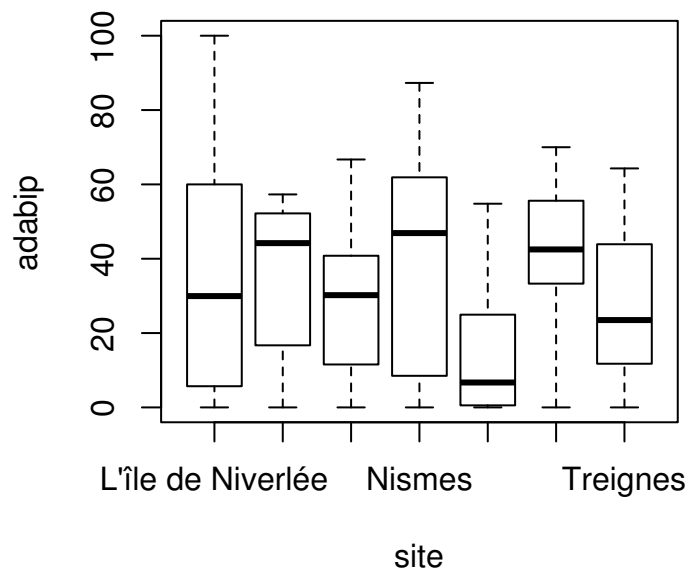
```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
mydata <- read.table("data/data2.txt", sep="\t", dec=".", na.string="", header = TRUE)
summary(mydata)
```

```
## tree   landsc   site      date      haraxy
## L:27   R:27    L'île de Niverrlée:18   Min.    :1.000   Min.    : 0.000
## M:27   S:27    Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000
## P:27   U:27    Mazée                   :12   Median :3.000   Median : 0.000
##                               Nismes                   :15   Mean    :2.667   Mean    : 5.856
##                               Niverrlée                  :12   3rd Qu.:3.000   3rd Qu.: 9.100
##                               Olloy s/ Viroin            : 9   Max.    :4.000   Max.    :53.600
##                               Treignes                   : 3   NA's    :2
##      adabip      adadec      calqua      caldec
## Min.    : 0.00   Min.    : 0.00   Min.    : 0.000   Min.    : 0.000
## 1st Qu.: 6.10   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.000
## Median : 31.80   Median : 6.70   Median : 0.000   Median : 0.000
## Mean    : 31.78   Mean    :10.42   Mean    : 7.192   Mean    : 8.847
## 3rd Qu.: 51.10   3rd Qu.:16.70   3rd Qu.: 7.450   3rd Qu.:13.800
## Max.    :100.00   Max.    :50.00   Max.    :60.000   Max.    :66.700
## NA's    :2       NA's    :2       NA's    :2       NA's    :2
##      exoqua      halsed      oencon      harqua
## Min.    : 0.000   Min.    : 0.000   Min.    : 0.000   Min.    : 0.00
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00
## Median : 0.000   Median : 0.000   Median : 0.000   Median : 0.00
## Mean    : 4.416   Mean    : 3.099   Mean    : 3.954   Mean    :13.37
## 3rd Qu.: 3.400   3rd Qu.: 0.000   3rd Qu.: 3.050   3rd Qu.:17.75
## Max.    :54.500   Max.    :50.000   Max.    :75.000   Max.    :77.30
## NA's    :2       NA's    :2       NA's    :2       NA's    :2
##      myroct      aphobl      anaoce
## Min.    : 0.000   Min.    : 0.000   Min.    :0.0000
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000
## Median : 0.000   Median : 0.000   Median :0.0000
## Mean    : 6.786   Mean    : 3.778   Mean    :0.4937
## 3rd Qu.: 3.550   3rd Qu.: 0.000   3rd Qu.:0.0000
## Max.    :57.100   Max.    :33.000   Max.    :7.1000
## NA's    :2       NA's    :2       NA's    :2
```

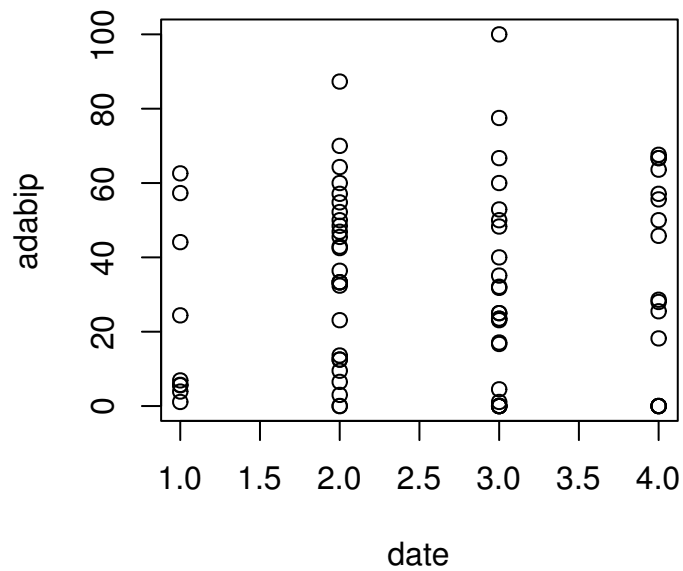
```
mean(mydata$adabip, na.rm = TRUE)
```

```
## [1] 31.77595
```

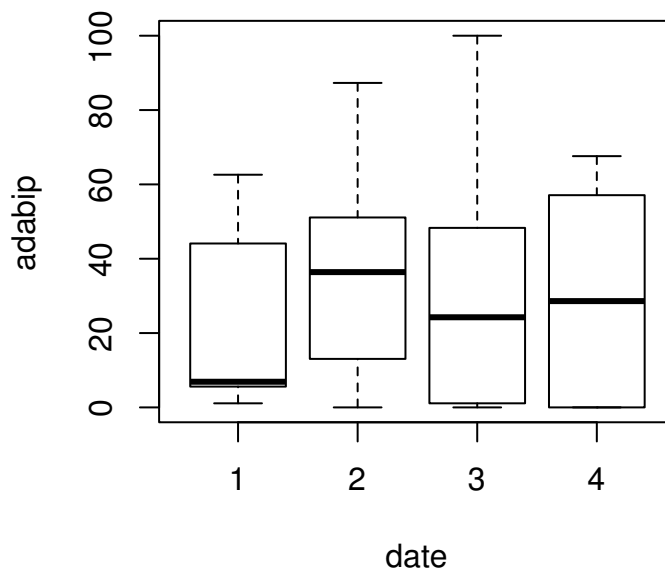
```
plot(adabip ~ site, data= mydata)
```



```
plot(adabip ~ date, data= mydata) # ce n'est pas ce qu'on veut
```



```
mydata$date <- as.factor(mydata$date)
plot(adabip ~ date, data= mydata)
```



*# Une même analyse statistique ne donnera pas les mêmes résultats selon que les variables
sont des facteurs ou non...*

```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
mydata <- read.table("data/data2.txt", sep="\t", dec=",", na.string="", header = TRUE)
lm(adabip ~ date, data=mydata)
```

```
##
## Call:
## lm(formula = adabip ~ date, data = mydata)
##
## Coefficients:
## (Intercept)      date
##    29.1889      0.9779
```

```
mydata$date <- as.factor(mydata$date)
lm(adabip ~ date, data=mydata)
```

```
##
## Call:
## lm(formula = adabip ~ date, data = mydata)
##
## Coefficients:
## (Intercept)      date2      date3      date4
##    23.511    12.467     5.485    10.218
```

- 3) Importez le fichier data2bis.csv. Avec la ligne de code suivante (où “mydata” est le nom de votre jeu de données), calculez la moyenne d’adabip pour le site “l’île de Niverlée”. Vous devez obtenir 33.35.
`mean(mydata[mydata$site == "L’île de Niverlée", "adabip"])`
 Si ça ne fonctionne pas, n’oubliez pas de regarder le jeu de données que vous avez importé pour essayer de comprendre la source du problème.

```

# Si vous êtes sous windows la colonne site ne sera normalement pas importée correctement.
# Tous les caractères accentués sont remplacés par d'autres caractères.
# Ceci est dû à l'encodage des caractères qui est UTF-8 pour ce fichier alors que sous
# Windows l'encodage par défaut est latin1 (qui porte aussi d'autres noms).
# Vous devez donc soit modifier l'encodage des caractères dans un éditeur de texte avancé
# soit utiliser l'option fileEncoding de read.tables.
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
mydata <- read.table("data/data2bis.csv", sep=";", dec=".", header = TRUE,
                    fileEncoding = "UTF-8")
summary(mydata)

```

```

## tree   landsc      site      date      haraxy
## L:27   R:27   L'île de Niverlée:18   Min.   :1.000   Min.   : 0.000
## M:27   S:27   Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000
## P:27   U:27   Mazée :12   Median :3.000   Median : 0.000
##                               Nismes :15   Mean   :2.667   Mean   : 5.856
##                               Niverlée :12   3rd Qu.:3.000   3rd Qu.: 9.100
##                               Olloy s/ Viroin : 9   Max.   :4.000   Max.   :53.600
##                               Treignes : 3   NA's   :2
##      adabip      adadec      calqua      caldec
## Min.   : 0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000
## 1st Qu.: 6.10   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.000
## Median : 31.80   Median : 6.70   Median : 0.000   Median : 0.000
## Mean   : 31.78   Mean   :10.42   Mean   : 7.192   Mean   : 8.847
## 3rd Qu.: 51.10   3rd Qu.:16.70   3rd Qu.: 7.450   3rd Qu.:13.800
## Max.   :100.00   Max.   :50.00   Max.   :60.000   Max.   :66.700
## NA's   :2       NA's   :2       NA's   :2       NA's   :2
##      exoqua      halsed      oencon      harqua
## Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00
## Median : 0.000   Median : 0.000   Median : 0.000   Median : 0.00
## Mean   : 4.416   Mean   : 3.099   Mean   : 3.954   Mean   :13.37
## 3rd Qu.: 3.400   3rd Qu.: 0.000   3rd Qu.: 3.050   3rd Qu.:17.75
## Max.   :54.500   Max.   :50.000   Max.   :75.000   Max.   :77.30
## NA's   :2       NA's   :2       NA's   :2       NA's   :2
##      myroct      aphobl      anaoce
## Min.   : 0.000   Min.   : 0.000   Min.   :0.0000
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000
## Median : 0.000   Median : 0.000   Median :0.0000
## Mean   : 6.786   Mean   : 3.778   Mean   :0.4937
## 3rd Qu.: 3.550   3rd Qu.: 0.000   3rd Qu.:0.0000
## Max.   :57.100   Max.   :33.000   Max.   :7.1000
## NA's   :2       NA's   :2       NA's   :2

```

```
mean(mydata[mydata$site == "L'île de Niverlée", "adabip"])
```

```
## [1] 33.35
```

- 4) Importez le fichier data3.xls. Calculez la moyenne de adabip (=31.77) et adadec (=10.42)

```

# Depuis le tableur, exporter le fichier en format texte et ensuite l'importer avec
# read.table.
# Petite difficulté : il y a eu une erreur d'encodage dans adadec : un point virgule à la
# place d'une virgule. Cette colonne est donc reconnue comme du texte et importée comme
# un facteur comme on le voit dans le résultat d'un summary()
# La solution la plus simple dans ce cas-ci est sans doute d'aller modifier le fichier
# d'origine. Il est possible aussi de le faire dans R si on a pas la possibilité de
# modifier les données sources.
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")

```

```
mydata <- read.table("data/Exercices_import/data3.csv", header=TRUE, sep="\t",
                     dec=",", na.string="")
summary(mydata)
```

```
## tree   landsc      site      date      haraxy
## L:27   R:27   L'île de Niverlée:18   Min.    :1.000   Min.    : 0.000
## M:27   S:27   Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000
## P:27   U:27   Mazée                :12   Median :3.000   Median : 0.000
##                               Nismes                :15   Mean    :2.667   Mean    : 5.856
##                               Niverlée               :12   3rd Qu.:3.000   3rd Qu.: 9.100
##                               Olloy s/ Viroin        : 9   Max.    :4.000   Max.    :53.600
##                               Treignes              : 3   NA's    :2
##      adabip      adadec      calqua      caldec
## Min.    : 0.00   0      :27   Min.    : 0.000   Min.    : 0.000
## 1st Qu.: 6.10   14,3   : 3   1st Qu.: 0.000   1st Qu.: 0.000
## Median : 31.80   16,7   : 3   Median : 0.000   Median : 0.000
## Mean    : 31.78   2,2    : 2   Mean    : 7.192   Mean    : 8.847
## 3rd Qu.: 51.10   22,2   : 2   3rd Qu.: 7.450   3rd Qu.:13.800
## Max.    :100.00   (Other):42   Max.    :60.000   Max.    :66.700
## NA's    :2      NA's    : 2   NA's    :2      NA's    :2
##      exoqua      halsed      oencon      harqua
## Min.    : 0.000   Min.    : 0.000   Min.    : 0.000   Min.    : 0.00
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00
## Median : 0.000   Median : 0.000   Median : 0.000   Median : 0.00
## Mean    : 4.416   Mean    : 3.099   Mean    : 3.954   Mean    :13.37
## 3rd Qu.: 3.400   3rd Qu.: 0.000   3rd Qu.: 3.050   3rd Qu.:17.75
## Max.    :54.500   Max.    :50.000   Max.    :75.000   Max.    :77.30
## NA's    :2      NA's    :2      NA's    :2      NA's    :2
##      myroct      aphobl      anaoce
## Min.    : 0.000   Min.    : 0.000   Min.    :0.0000
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000
## Median : 0.000   Median : 0.000   Median :0.0000
## Mean    : 6.786   Mean    : 3.778   Mean    :0.4937
## 3rd Qu.: 3.550   3rd Qu.: 0.000   3rd Qu.:0.0000
## Max.    :57.100   Max.    :33.000   Max.    :7.1000
## NA's    :2      NA's    :2      NA's    :2
```

```
mean(mydata$adabip, na.rm = TRUE)
```

```
## [1] 31.77595
```

```
mean(mydata$adadec, na.rm = TRUE)
```

```
## Warning in mean.default(mydata$adadec, na.rm = TRUE): argument is not
## numeric or logical: returning NA
```

```
## [1] NA
```

```
# Avec les données corrigées dans le fichier orriginal
```

```
mydata <- read.table("data/Exercices_import/data3_corrige.csv", header=TRUE, sep="\t",
                     dec=",", na.string="")
summary(mydata)
```

```
## tree   landsc      site      date      haraxy
## L:27   R:27   L'île de Niverlée:18   Min.    :1.000   Min.    : 0.000
## M:27   S:27   Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000
## P:27   U:27   Mazée                :12   Median :3.000   Median : 0.000
```

```
##           Nismes           :15   Mean   :2.667   Mean   : 5.856
##           Niverlée         :12   3rd Qu.:3.000   3rd Qu.: 9.100
##           Olloy s/ Viroin   : 9   Max.    :4.000   Max.    :53.600
##           Treignes         : 3                               NA's    :2
##           adabip           adadec           calqua           caldec
##   Min.    : 0.00   Min.    : 0.00   Min.    : 0.000   Min.    : 0.000
##   1st Qu.: 6.10   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.000
##   Median : 31.80   Median : 6.70   Median : 0.000   Median : 0.000
##   Mean    : 31.78   Mean    :10.42   Mean    : 7.192   Mean    : 8.847
##   3rd Qu.: 51.10   3rd Qu.:16.70   3rd Qu.: 7.450   3rd Qu.:13.800
##   Max.    :100.00   Max.    :50.00   Max.    :60.000   Max.    :66.700
##   NA's    :2       NA's    :2       NA's    :2       NA's    :2
##           exoqua           halsed           oencon           harqua
##   Min.    : 0.000   Min.    : 0.000   Min.    : 0.000   Min.    : 0.00
##   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00
##   Median : 0.000   Median : 0.000   Median : 0.000   Median : 0.00
##   Mean    : 4.416   Mean    : 3.099   Mean    : 3.954   Mean    :13.37
##   3rd Qu.: 3.400   3rd Qu.: 0.000   3rd Qu.: 3.050   3rd Qu.:17.75
##   Max.    :54.500   Max.    :50.000   Max.    :75.000   Max.    :77.30
##   NA's    :2       NA's    :2       NA's    :2       NA's    :2
##           myroct           aphobl           anaoce
##   Min.    : 0.000   Min.    : 0.000   Min.    :0.0000
##   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000
##   Median : 0.000   Median : 0.000   Median :0.0000
##   Mean    : 6.786   Mean    : 3.778   Mean    :0.4937
##   3rd Qu.: 3.550   3rd Qu.: 0.000   3rd Qu.:0.0000
##   Max.    :57.100   Max.    :33.000   Max.    :7.1000
##   NA's    :2       NA's    :2       NA's    :2
```

```
mean(mydata$adabip, na.rm = TRUE)
```

```
## [1] 31.77595
```

```
mean(mydata$adadec, na.rm = TRUE)
```

```
## [1] 10.42152
```

- 5) Importez le fichier data4.xls. Avec la fonction mean, calculez la moyenne de calqua (=7.19) et oencon (=3.95) (NB : Inspiré de faits réels...)

```
# Plusieurs difficultés ici. Le fichier Excel comprend des valeurs "fantômes" invisibles
# sur certaines lignes et colonnes en dehors de la matrice de données.
# Le plus simple pour éviter ce genre de problèmes est de sélectionner la matrice de
# données et de la copier-coller dans une feuille vierge avant l'exportation en format
# texte.
# Ensuite le titre de la colonne OENCON a été encodé comme OENCON, le deuxième "O" a été
# remplacé par un zéro...
# Il faut donc corriger le titre au bien l'utiliser avec le zéro.
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
mydata <- read.table("data/Exercices_import/data4.csv", header=TRUE, sep="\t", dec=".",
                     na.string="")
summary(mydata)
```

```
## tree   landsc           site           date           HARAXY
## L:27   R:27   L'île de Niverlée:18   Min.   :1.000   Min.   : 0.000
## M:27   S:27   Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000
## P:27   U:27   Mazée                       :12   Median :3.000   Median : 0.000
##                               Nismes           :15   Mean    :2.667   Mean    : 5.856
```

```
##           Niverlée           :12  3rd Qu.:3.000  3rd Qu.: 9.100
##           Olloy s/ Viroin    : 9  Max.      :4.000  Max.      :53.600
##           Treignes           : 3                      NA's      :2
##           ADABIP             ADADEC             CALQUA             CALDEC
## Min.      : 0.00  Min.      : 0.00  Min.      : 0.000  Min.      : 0.000
## 1st Qu.   : 6.10  1st Qu.   : 0.00  1st Qu.   : 0.000  1st Qu.   : 0.000
## Median    : 31.80  Median    : 6.70  Median    : 0.000  Median    : 0.000
## Mean      : 31.78  Mean      :10.42  Mean      : 7.192  Mean      : 8.847
## 3rd Qu.   : 51.10  3rd Qu.   :16.70  3rd Qu.   : 7.450  3rd Qu.   :13.800
## Max.      :100.00  Max.      :50.00  Max.      :60.000  Max.      :66.700
## NA's      :2      NA's      :2      NA's      :2      NA's      :2
##           EXOQUA             HALSED             OENCON             HARQUA
## Min.      : 0.000  Min.      : 0.000  Min.      : 0.000  Min.      : 0.00
## 1st Qu.   : 0.000  1st Qu.   : 0.000  1st Qu.   : 0.000  1st Qu.   : 0.00
## Median    : 0.000  Median    : 0.000  Median    : 0.000  Median    : 0.00
## Mean      : 4.416  Mean      : 3.099  Mean      : 3.954  Mean      :13.37
## 3rd Qu.   : 3.400  3rd Qu.   : 0.000  3rd Qu.   : 3.050  3rd Qu.   :17.75
## Max.      :54.500  Max.      :50.000  Max.      :75.000  Max.      :77.30
## NA's      :2      NA's      :2      NA's      :2      NA's      :2
##           MYROCT             APHOBL             ANAOCE
## Min.      : 0.000  Min.      : 0.000  Min.      :0.0000
## 1st Qu.   : 0.000  1st Qu.   : 0.000  1st Qu.   :0.0000
## Median    : 0.000  Median    : 0.000  Median    :0.0000
## Mean      : 6.786  Mean      : 3.778  Mean      :0.4937
## 3rd Qu.   : 3.550  3rd Qu.   : 0.000  3rd Qu.   :0.0000
## Max.      :57.100  Max.      :33.000  Max.      :7.1000
## NA's      :2      NA's      :2      NA's      :2
```

```
mean(mydata$CALQUA, na.rm = TRUE)
```

```
## [1] 7.192405
```

```
mean(mydata$OENCON, na.rm = TRUE)
```

```
## [1] 3.95443
```

- 6) Importez le fichier data5.xls et vérifiez que l'importation s'est faite correctement. Si l'importation ne s'est pas faite correctement, essayez d'importer le fichier sans la dernière colonne et essayez de comprendre l'origine du problème

```
# La difficulté ici vient du fait qu'il y a un champ "mémo" qui contient des informations
# diverses encodées par les utilisateurs.
# On a typiquement des problèmes avec ce genre de champs parce qu'ils contiennent des
# caractères spéciaux qui interfèrent avec la lecture du fichier :
# point virgule, tabulation (plus rare), retour à la ligne, guillemets (fermés ou ouverts).
# La première défense contre ce problème est de bien demander au logiciel lors de
# l'exportation au format texte d'entourer les champs textes par des guillemets.
# Dans bien des cas ça suffit. Mais dans certains cas il faut beaucoup plus chipoter...
# Une mesure radicale est de ne pas importer ces colonnes si elles ne sont pas nécessaires.
# Si non R indique en général le numéro de la ligne qui pose problème. On peut alors
# essayer de comprendre ce qui se passe en ouvrant le fichier dans un éditeur de texte.
# Dans bien des cas il suffit de faire quelques corrections à la main. Si ce n'est pas
# possible, il faudra utiliser readLines ou scan et des expressions régulières pour
# traiter le fichier.
#
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
mydata <- read.table("data/Exercices_import/data5.csv", header=TRUE, sep="\t",
                    dec=",", na.string="")
summary(mydata)
```

```

## tree   landsc      site      date      haraxy
## L:27   R:27   L'île de Niverlée:18   Min.    :1.000   Min.    : 0.000
## M:27   S:27   Matagne-la-Petite:12   1st Qu.:2.000   1st Qu.: 0.000
## P:27   U:27   Mazée                      :12   Median  :3.000   Median  : 0.000
##                      Nismes                      :15   Mean    :2.667   Mean    : 5.856
##                      Niverlée                    :12   3rd Qu.:3.000   3rd Qu.: 9.100
##                      Olloy s/ Viroin              : 9   Max.    :4.000   Max.    :53.600
##                      Treignes                     : 3   NA's    :2
##      adabip      adadec      calqua      caldec
## Min.    : 0.00   Min.    : 0.00   Min.    : 0.000   Min.    : 0.000
## 1st Qu.: 6.10   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.000
## Median : 31.80   Median : 6.70   Median : 0.000   Median : 0.000
## Mean    : 31.78   Mean    :10.42   Mean    : 7.192   Mean    : 8.847
## 3rd Qu.: 51.10   3rd Qu.:16.70   3rd Qu.: 7.450   3rd Qu.:13.800
## Max.    :100.00   Max.    :50.00   Max.    :60.000   Max.    :66.700
## NA's    :2      NA's    :2      NA's    :2      NA's    :2
##      exoqua      halsed      oencon      harqua
## Min.    : 0.000   Min.    : 0.000   Min.    : 0.000   Min.    : 0.00
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00
## Median : 0.000   Median : 0.000   Median : 0.000   Median : 0.00
## Mean    : 4.416   Mean    : 3.099   Mean    : 3.954   Mean    :13.37
## 3rd Qu.: 3.400   3rd Qu.: 0.000   3rd Qu.: 3.050   3rd Qu.:17.75
## Max.    :54.500   Max.    :50.000   Max.    :75.000   Max.    :77.30
## NA's    :2      NA's    :2      NA's    :2      NA's    :2
##      myroct      aphobl      anaoce
## Min.    : 0.000   Min.    : 0.000   Min.    :0.0000
## 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000
## Median : 0.000   Median : 0.000   Median :0.0000
## Mean    : 6.786   Mean    : 3.778   Mean    :0.4937
## 3rd Qu.: 3.550   3rd Qu.: 0.000   3rd Qu.:0.0000
## Max.    :57.100   Max.    :33.000   Max.    :7.1000
## NA's    :2      NA's    :2      NA's    :2
##      memo
## pluie;vent;froid      : 2
## beaucoup de vent      : 1
## Branches basses temps froid: 1
## individus très "schnoll" : 1
## pluie                  : 1
## (Other)                : 5
## NA's                   :70

```

Exercices subsidiaires (pas de R)

- 7) Importez les données contenues dans le fichier “insectes_feuilles.xls”. Calculez la proportion de feuilles F5 où les insectes étaient présents tous traitements et toutes variétés confondus NB : l’essentiel du travail se fait ici dans le tableur
L’essentiel du travail consiste ici à réorganiser le fichier xls dans le tableur pour qu’il soit exploitable. Il faut éliminer les sous-totaux, fusionner les deux feuilles de calcul, rajouter une colonne “traitement” et une colonne “variété. Cette manière d’encoder les données n’est pas idéale pour le traitement. Il vaut mieux en général encoder les données brutes et faire les statistiques descriptives à part (par exemple avec des tableaux croisés dynamiques/pilotes de données) Lors de l’importation dans R on prendra garde au fait que les cases vides correspondent à des données manquantes (argument na.string de read.table).
- 8) Importez les données contenues dans le fichier “météo.xlsx” issu d’un site internet. Calculez la température minimale moyenne et la quantité totale de précipitations sur la période. NB : l’essentiel du travail se fait ici dans le tableur
Ici aussi il y a pas mal de boulot à faire dans le tableur pour pouvoir sauvegarder ce fichier dans un format texte exploitable. Il faut défusionner les cellules fusionnées, ensuite faire un tri sur la date pour séparer les lignes contenant les heures de prise de mesure des mesures proprement-dites. Il faut éliminer les unités “°C” et “mm” avec des chercher-remplacer. Attention dans certains cas (Excel 2007 ?) après un rechercher remplacer sur “°C”, il reste un caractère invisible qu’il faut également éliminer.

Exercices sur l'extraction de données (Subscripting)

Chargez le fichier "ladybirds.txt". Il s'agit de données de comptage de coccinelles sur 3 espèces d'arbres (colonne "tree") : pins (P), tilleuls (L cfr "Lime"), érables (M cfr "Maple") dans 3 types de paysages (landsc) : Urbain (U), Suburbain (S) et Rural (R), sur une série de sites et à 4 dates. Les colonnes après la quatrième correspondent aux espèces de coccinelles (nombre d'individus).

```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
d <- read.table("data/ladybirds.txt", sep="\t", header = TRUE)
summary(d)
```

```
## tree   landsc   site   date   haraxy   adabip
## L:27   R:27    LR1    : 3   date1: 9   Min.    : 0.00   Min.    : 0.00
## M:27   S:27    LR2    : 3   date2:27  1st Qu.: 0.00   1st Qu.: 1.00
## P:27   U:27    LR3    : 3   date3:27  Median : 0.00   Median : 4.00
##                               LS1    : 3   date4:18  Mean    : 2.21   Mean    : 13.51
##                               LS2    : 3               3rd Qu.: 2.00   3rd Qu.: 12.00
##                               LS3    : 3               Max.    :30.00   Max.    :212.00
##                               (Other):63
##      adadec      calqua      caldec      exoqua
## Min.    : 0.000   Min.    : 0.0000   Min.    : 0.000   Min.    : 0
## 1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0
## Median : 1.000   Median : 0.0000   Median : 0.000   Median : 0
## Mean    : 3.778   Mean    : 0.9506   Mean    : 1.136   Mean    : 1
## 3rd Qu.: 3.000   3rd Qu.: 2.0000   3rd Qu.: 2.000   3rd Qu.: 1
## Max.    :82.000   Max.    :13.0000   Max.    :14.000   Max.    :15
##
##      halsed      oencon      harqua      myroct
## Min.    :0.0000   Min.    : 0.000   Min.    : 0.000   Min.    : 0.000
## 1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000
## Median :0.0000   Median : 0.000   Median : 0.000   Median : 0.000
## Mean    :0.2593   Mean    : 1.049   Mean    : 7.741   Mean    : 3.951
## 3rd Qu.:0.0000   3rd Qu.: 1.000   3rd Qu.: 10.000   3rd Qu.: 2.000
## Max.    :2.0000   Max.    :11.000   Max.    :106.000   Max.    :54.000
##
##      aphobl      anaoce
## Min.    : 0.000   Min.    :0.0000
## 1st Qu.: 0.000   1st Qu.:0.0000
## Median : 0.000   Median :0.0000
## Mean    : 2.074   Mean    :0.2469
## 3rd Qu.: 0.000   3rd Qu.:0.0000
## Max.    :29.000   Max.    :5.0000
##
```

- 1) Sélectionnez toutes les colonnes sauf la 6, 9 et 12

```
d[, -c(6, 9, 12)]
```

```
## tree landsc site date haraxy adadec calqua exoqua halsed harqua myroct
## 1 P R PR1 date1 0 0 0 0 0 36 0
## 2 P R PR2 date1 0 1 2 0 2 24 3
## 3 P R PR3 date1 0 3 0 1 0 22 0
## 4 P S PS1 date1 0 6 0 0 1 21 43
## 5 P S PS2 date1 0 2 0 4 1 106 41
## 6 P S PS3 date1 0 7 0 0 0 35 1
## 7 P U PU1 date1 16 13 0 0 0 38 7
## 8 P U PU2 date1 22 52 0 2 0 22 12
## 9 P U PU3 date1 2 82 0 0 0 64 10
## 10 L R LR1 date2 0 1 1 0 0 0 0
## 11 L R LR2 date2 0 5 0 0 0 0 0
```

## 12	L	R	LR3	date2	0	8	2	0	0	0	0
## 13	L	S	LS1	date2	1	4	2	0	0	0	0
## 14	L	S	LS2	date2	0	4	3	1	0	0	0
## 15	L	S	LS3	date2	0	2	2	0	0	0	0
## 16	L	U	LU1	date2	0	6	0	0	0	0	0
## 17	L	U	LU2	date2	2	0	3	0	0	0	0
## 18	L	U	LU3	date2	5	15	2	0	0	0	0
## 19	M	R	MR1	date2	0	1	3	0	0	0	0
## 20	M	R	MR2	date2	0	1	6	2	0	0	0
## 21	M	R	MR3	date2	1	2	13	1	0	0	0
## 22	M	S	MS1	date2	0	1	1	0	1	0	0
## 23	M	S	MS2	date2	0	0	2	0	2	0	0
## 24	M	S	MS3	date2	0	1	3	0	2	0	0
## 25	M	U	MU1	date2	6	2	1	2	0	0	0
## 26	M	U	MU2	date2	2	2	2	1	0	0	0
## 27	M	U	MU3	date2	2	0	2	0	1	0	0
## 28	P	R	PR1	date2	0	0	0	1	0	17	1
## 29	P	R	PR2	date2	0	0	0	0	0	13	5
## 30	P	R	PR3	date2	0	2	0	0	0	14	0
## 31	P	S	PS1	date2	0	0	0	0	0	13	10
## 32	P	S	PS2	date2	0	0	0	0	0	36	54
## 33	P	S	PS3	date2	0	1	0	3	0	20	7
## 34	P	U	PU1	date2	4	1	0	0	0	9	5
## 35	P	U	PU2	date2	6	6	0	0	0	10	2
## 36	P	U	PU3	date2	0	6	0	0	0	4	2
## 37	L	R	LR1	date3	0	1	0	0	1	0	0
## 38	L	R	LR2	date3	0	6	1	0	0	0	0
## 39	L	R	LR3	date3	0	2	1	7	0	0	0
## 40	L	S	LS1	date3	1	7	3	1	0	0	0
## 41	L	S	LS2	date3	2	4	0	0	1	0	0
## 42	L	S	LS3	date3	0	0	1	0	0	0	0
## 43	L	U	LU1	date3	3	3	0	1	1	0	0
## 44	L	U	LU2	date3	9	2	2	0	0	0	0
## 45	L	U	LU3	date3	6	3	2	0	1	0	0
## 46	M	R	MR1	date3	0	0	0	0	0	0	0
## 47	M	R	MR2	date3	0	0	0	1	0	0	0
## 48	M	R	MR3	date3	0	1	0	0	0	0	0
## 49	M	S	MS1	date3	2	1	0	1	0	0	0
## 50	M	S	MS2	date3	0	0	1	1	2	0	0
## 51	M	S	MS3	date3	0	2	0	1	1	0	0
## 52	M	U	MU1	date3	2	1	0	0	0	0	0
## 53	M	U	MU2	date3	30	2	0	2	0	0	0
## 54	M	U	MU3	date3	0	0	0	0	0	0	0
## 55	P	R	PR1	date3	0	0	0	4	0	14	2
## 56	P	R	PR2	date3	0	0	0	0	0	16	9
## 57	P	R	PR3	date3	0	5	0	0	0	20	0
## 58	P	S	PS1	date3	0	0	0	0	0	3	20
## 59	P	S	PS2	date3	0	0	0	0	0	15	44
## 60	P	S	PS3	date3	0	0	0	0	0	47	21
## 61	P	U	PU1	date3	1	0	0	0	0	3	5
## 62	P	U	PU2	date3	0	6	0	0	0	0	6
## 63	P	U	PU3	date3	0	4	0	0	0	4	10
## 64	L	R	LR1	date4	0	0	2	0	2	0	0
## 65	L	R	LR2	date4	0	2	3	0	0	0	0
## 66	L	R	LR3	date4	0	0	0	6	1	0	0
## 67	L	S	LS1	date4	2	2	1	0	0	0	0
## 68	L	S	LS2	date4	2	2	2	0	0	0	0
## 69	L	S	LS3	date4	0	0	1	0	0	0	0
## 70	L	U	LU1	date4	2	2	0	0	0	0	0
## 71	L	U	LU2	date4	6	1	1	15	0	0	0
## 72	L	U	LU3	date4	4	6	2	6	0	0	0
## 73	M	R	MR1	date4	0	0	0	0	0	0	0
## 74	M	R	MR2	date4	0	0	0	1	1	0	0

## 75	M	R	MR3 date4	1	2	1	0	0	0	0
## 76	M	S	MS1 date4	2	0	0	0	0	0	0
## 77	M	S	MS2 date4	2	0	0	0	0	0	0
## 78	M	S	MS3 date4	0	0	0	2	0	0	0
## 79	M	U	MU1 date4	10	2	1	6	0	0	0
## 80	M	U	MU2 date4	22	0	2	8	0	1	0
## 81	M	U	MU3 date4	1	0	0	0	0	0	0
##	aphobl	anaoce								
## 1	11	3								
## 2	2	0								
## 3	8	0								
## 4	15	3								
## 5	19	0								
## 6	11	0								
## 7	2	0								
## 8	0	0								
## 9	0	0								
## 10	0	0								
## 11	0	0								
## 12	0	0								
## 13	0	0								
## 14	0	0								
## 15	0	0								
## 16	0	0								
## 17	0	0								
## 18	0	0								
## 19	0	0								
## 20	0	0								
## 21	0	0								
## 22	0	0								
## 23	1	0								
## 24	0	0								
## 25	0	0								
## 26	0	0								
## 27	0	0								
## 28	0	0								
## 29	0	0								
## 30	3	0								
## 31	3	0								
## 32	6	0								
## 33	9	3								
## 34	0	0								
## 35	10	0								
## 36	0	0								
## 37	0	0								
## 38	0	0								
## 39	0	0								
## 40	0	0								
## 41	0	0								
## 42	0	0								
## 43	0	0								
## 44	0	0								
## 45	0	0								
## 46	0	0								
## 47	0	0								
## 48	0	0								
## 49	0	0								
## 50	0	0								
## 51	0	0								
## 52	0	0								
## 53	0	0								
## 54	0	0								
## 55	6	2								

```
## 56      2      1
## 57     12      2
## 58      6      0
## 59     29      0
## 60     13      5
## 61      0      0
## 62      0      0
## 63      0      0
## 64      0      0
## 65      0      0
## 66      0      0
## 67      0      0
## 68      0      0
## 69      0      0
## 70      0      0
## 71      0      0
## 72      0      0
## 73      0      0
## 74      0      0
## 75      0      0
## 76      0      0
## 77      0      0
## 78      0      0
## 79      0      0
## 80      0      1
## 81      0      0
```

- 2) Sélectionnez les 10 premières lignes et uniquement les colonnes correspondant aux espèces

```
d[1:10, 5:ncol(d)]
```

```
##      haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## 1         0      3      0      0      0      0      0      0      0      36      0
## 2         0     11      1      2      0      0      2      0      0      24      3
## 3         0      2      3      0      0      1      0      0      0      22      0
## 4         0      1      6      0      0      0      1      0      0      21     43
## 5         0      7      2      0      0      4      1      0     106     41
## 6         0      4      7      0      0      0      0      0      0      35      1
## 7        16     67     13      0      0      0      0      9      38      7
## 8        22    184     52      0      0      2      0      0      22     12
## 9         2    212     82      0      0      0      0      0      64     10
## 10        0      1      1      1      0      0      0      0      0      0      0
##      aphobl anaoce
## 1         11      3
## 2          2      0
## 3          8      0
## 4         15      3
## 5         19      0
## 6         11      0
## 7          2      0
## 8          0      0
## 9          0      0
## 10         0      0
```

```
d[1:10, -c(1:4)] # équivalent
```

```
##      haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## 1         0      3      0      0      0      0      0      0      36      0
## 2         0     11      1      2      0      0      2      0      24      3
## 3         0      2      3      0      0      1      0      0      22      0
```

```
## 4      0      1      6      0      0      0      1      0      21      43
## 5      0      7      2      0      0      4      1      0     106      41
## 6      0      4      7      0      0      0      0      0      35       1
## 7     16     67     13      0      0      0      0      9      38       7
## 8     22    184     52      0      0      2      0      0      22      12
## 9      2    212     82      0      0      0      0      0      64      10
## 10     0      1      1      1      0      0      0      0      0       0
##      aphobl anaoce
## 1      11      3
## 2       2      0
## 3       8      0
## 4      15      3
## 5      19      0
## 6      11      0
## 7       2      0
## 8       0      0
## 9       0      0
## 10      0      0
```

- 3) Sélectionnez les données de l'espèce "haraxy" sur pin.

```
d[d$tree == "P", "haraxy"]
```

```
## [1] 0 0 0 0 0 0 16 22 2 0 0 0 0 0 0 4 6 0 0 0 0 0 0
## [24] 0 1 0 0
```

- 4) Ne gardez que les observations (lignes) où l'espèce "anaoce" était présente

```
d[d$anaoce >0,]
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 1      P      R  PR1 date1      0      3      0      0      0      0      0
## 4      P      S  PS1 date1      0      1      6      0      0      0      1
## 33     P      S  PS3 date2      0      3      1      0      0      3      0
## 55     P      R  PR1 date3      0      0      0      0      0      4      0
## 56     P      R  PR2 date3      0      0      0      0      0      0      0
## 57     P      R  PR3 date3      0      0      5      0      0      0      0
## 60     P      S  PS3 date3      0      1      0      0      0      0      0
## 80     M      U  MU2 date4     22     13      0      2      0      8      0
##      oencon harqua myroct aphobl anaoce
## 1      0      36      0     11      3
## 4      0      21     43     15      3
## 33     0      20      7      9      3
## 55     0      14      2      6      2
## 56     0      16      9      2      1
## 57     0      20      0     12      2
## 60     0      47     21     13      5
## 80     4       1      0      0      1
```

- 5) Ne gardez que les observations (lignes) "rurales" où "anaoce" était présente

```
d[d$landsc == "R" & d$anaoce >0,]
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 1      P      R  PR1 date1      0      3      0      0      0      0      0
## 55     P      R  PR1 date3      0      0      0      0      0      4      0
## 56     P      R  PR2 date3      0      0      0      0      0      0      0
## 57     P      R  PR3 date3      0      0      5      0      0      0      0
```

```
##      oencon harqua myroct aphobl anaocce
## 1      0      36      0      11      3
## 55     0      14      2      6      2
## 56     0      16      9      2      1
## 57     0      20      0     12      2
```

- 6) Sélectionnez les données des sites PS1, MU2, LS3, PU2, LR3, LR2, MR1

```
d[d$site %in% c("PS1", "MU2", "LS3", "PU2", "LR3", "LR2", "MR1"),]
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 4      P      S  PS1 date1      0      1      6      0      0      0      1
## 8      P      U  PU2 date1     22     184     52      0      0      2      0
## 11     L      R  LR2 date2      0      9      5      0      0      0      0
## 12     L      R  LR3 date2      0     15      8      2      0      0      0
## 15     L      S  LS3 date2      0      3      2      2      2      0      0
## 19     M      R  MR1 date2      0      3      1      3      0      0      0
## 26     M      U  MU2 date2      2     17      2      2      0      1      0
## 31     P      S  PS1 date2      0      0      0      0      0      0      0
## 35     P      U  PU2 date2      6     32      6      0      0      0      0
## 38     L      R  LR2 date3      0      4      6      1      6      0      0
## 39     L      R  LR3 date3      0      4      2      1      3      7      0
## 42     L      S  LS3 date3      0      1      0      1      2      0      0
## 46     M      R  MR1 date3      0      0      0      0      0      0      0
## 53     M      U  MU2 date3     30     18      2      0      1      2      0
## 58     P      S  PS1 date3      0      6      0      0      0      0      0
## 62     P      U  PU2 date3      0      8      6      0      0      0      0
## 65     L      R  LR2 date4      0      0      2      3      0      0      0
## 66     L      R  LR3 date4      0      2      0      0      2      6      1
## 69     L      S  LS3 date4      0      0      0      1      2      0      0
## 73     M      R  MR1 date4      0      0      0      0      0      0      0
## 80     M      U  MU2 date4     22     13      0      2      0      8      0
##      oencon harqua myroct aphobl anaocce
## 4      0      21      43     15      3
## 8      0      22      12      0      0
## 11     0      0      0      0      0
## 12     0      0      0      0      0
## 15     0      0      0      0      0
## 19     0      0      0      0      0
## 26     7      0      0      0      0
## 31     0      13      10      3      0
## 35     0      10      2     10      0
## 38     0      0      0      0      0
## 39     0      0      0      0      0
## 42     0      0      0      0      0
## 46     0      0      0      0      0
## 53     3      0      0      0      0
## 58     0      3      20      6      0
## 62     0      0      6      0      0
## 65     0      0      0      0      0
## 66     0      0      0      0      0
## 69     0      0      0      0      0
## 73     0      0      0      0      0
## 80     4      1      0      0      1
```

- 7) Sélectionnez toutes les données sauf celles de ces sites (PS1, MU2, LS3, PU2, LR3, LR2, MR1)

```
d[!d$site %in% c("PS1", "MU2", "LS3", "PU2", "LR3", "LR2", "MR1"),]
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 1      P      R  PR1 date1      0      3      0      0      0      0      0
```

## 2	P	R	PR2 date1	0	11	1	2	0	0	2
## 3	P	R	PR3 date1	0	2	3	0	0	1	0
## 5	P	S	PS2 date1	0	7	2	0	0	4	1
## 6	P	S	PS3 date1	0	4	7	0	0	0	0
## 7	P	U	PU1 date1	16	67	13	0	0	0	0
## 9	P	U	PU3 date1	2	212	82	0	0	0	0
## 10	L	R	LR1 date2	0	1	1	1	0	0	0
## 13	L	S	LS1 date2	1	21	4	2	2	0	0
## 14	L	S	LS2 date2	0	2	4	3	6	1	0
## 16	L	U	LU1 date2	0	16	6	0	2	0	0
## 17	L	U	LU2 date2	2	55	0	3	0	0	0
## 18	L	U	LU3 date2	5	12	15	2	2	0	0
## 20	M	R	MR2 date2	0	3	1	6	1	2	0
## 21	M	R	MR3 date2	1	15	2	13	0	1	0
## 22	M	S	MS1 date2	0	4	1	1	0	0	1
## 23	M	S	MS2 date2	0	1	0	2	2	0	2
## 24	M	S	MS3 date2	0	3	1	3	0	0	2
## 25	M	U	MU1 date2	6	24	2	1	0	2	0
## 27	M	U	MU3 date2	2	4	0	2	0	0	1
## 28	P	R	PR1 date2	0	3	0	0	0	1	0
## 29	P	R	PR2 date2	0	0	0	0	0	0	0
## 30	P	R	PR3 date2	0	2	2	0	0	0	0
## 32	P	S	PS2 date2	0	3	0	0	0	0	0
## 33	P	S	PS3 date2	0	3	1	0	0	3	0
## 34	P	U	PU1 date2	4	17	1	0	0	0	0
## 36	P	U	PU3 date2	0	10	6	0	0	0	0
## 37	L	R	LR1 date3	0	0	1	0	0	0	1
## 40	L	S	LS1 date3	1	13	7	3	8	1	0
## 41	L	S	LS2 date3	2	1	4	0	14	0	1
## 43	L	U	LU1 date3	3	7	3	0	5	1	1
## 44	L	U	LU2 date3	9	62	2	2	0	0	0
## 45	L	U	LU3 date3	6	14	3	2	2	0	1
## 47	M	R	MR2 date3	0	2	0	0	0	1	0
## 48	M	R	MR3 date3	0	3	1	0	2	0	0
## 49	M	S	MS1 date3	2	9	1	0	2	1	0
## 50	M	S	MS2 date3	0	2	0	1	6	1	2
## 51	M	S	MS3 date3	0	0	2	0	2	1	1
## 52	M	U	MU1 date3	2	9	1	0	0	0	0
## 54	M	U	MU3 date3	0	3	0	0	0	0	0
## 55	P	R	PR1 date3	0	0	0	0	0	4	0
## 56	P	R	PR2 date3	0	0	0	0	0	0	0
## 57	P	R	PR3 date3	0	0	5	0	0	0	0
## 59	P	S	PS2 date3	0	0	0	0	0	0	0
## 60	P	S	PS3 date3	0	1	0	0	0	0	0
## 61	P	U	PU1 date3	1	3	0	0	0	0	0
## 63	P	U	PU3 date3	0	6	4	0	0	0	0
## 64	L	R	LR1 date4	0	0	0	2	1	0	2
## 67	L	S	LS1 date4	2	15	2	1	5	0	0
## 68	L	S	LS2 date4	2	4	2	2	4	0	0
## 70	L	U	LU1 date4	2	8	2	0	2	0	0
## 71	L	U	LU2 date4	6	48	1	1	0	15	0
## 72	L	U	LU3 date4	4	7	6	2	0	6	0
## 74	M	R	MR2 date4	0	2	0	0	0	1	1
## 75	M	R	MR3 date4	1	8	2	1	0	0	0
## 76	M	S	MS1 date4	2	8	0	0	0	0	0
## 77	M	S	MS2 date4	2	0	0	0	2	0	0
## 78	M	S	MS3 date4	0	7	0	0	2	2	0
## 79	M	U	MU1 date4	10	27	2	1	2	6	0
## 81	M	U	MU3 date4	1	0	0	0	0	0	0
##	oencon harqua myroct aphobl anaoce									
## 1	0	36	0	11	3					
## 2	0	24	3	2	0					
## 3	0	22	0	8	0					

## 5	0	106	41	19	0
## 6	0	35	1	11	0
## 7	9	38	7	2	0
## 9	0	64	10	0	0
## 10	0	0	0	0	0
## 13	0	0	0	0	0
## 14	0	0	0	0	0
## 16	4	0	0	0	0
## 17	3	0	0	0	0
## 18	1	0	0	0	0
## 20	0	0	0	0	0
## 21	0	0	0	0	0
## 22	1	0	0	0	0
## 23	0	0	0	1	0
## 24	0	0	0	0	0
## 25	11	0	0	0	0
## 27	2	0	0	0	0
## 28	0	17	1	0	0
## 29	0	13	5	0	0
## 30	0	14	0	3	0
## 32	0	36	54	6	0
## 33	0	20	7	9	3
## 34	4	9	5	0	0
## 36	0	4	2	0	0
## 37	0	0	0	0	0
## 40	4	0	0	0	0
## 41	0	0	0	0	0
## 43	2	0	0	0	0
## 44	5	0	0	0	0
## 45	1	0	0	0	0
## 47	0	0	0	0	0
## 48	0	0	0	0	0
## 49	0	0	0	0	0
## 50	0	0	0	0	0
## 51	0	0	0	0	0
## 52	5	0	0	0	0
## 54	0	0	0	0	0
## 55	0	14	2	6	2
## 56	0	16	9	2	1
## 57	0	20	0	12	2
## 59	0	15	44	29	0
## 60	0	47	21	13	5
## 61	1	3	5	0	0
## 63	0	4	10	0	0
## 64	0	0	0	0	0
## 67	2	0	0	0	0
## 68	0	0	0	0	0
## 70	0	0	0	0	0
## 71	0	0	0	0	0
## 72	0	0	0	0	0
## 74	0	0	0	0	0
## 75	0	0	0	0	0
## 76	2	0	0	0	0
## 77	0	0	0	0	0
## 78	0	0	0	0	0
## 79	11	0	0	0	0
## 81	3	0	0	0	0

- 8) Sélectionnez les données sur Pins à la date 1 et sur feuillus (tilleuls et érables) pour toutes les dates sauf la date1


```
d[(d$tree == "P" & d$date == "date1") | (d$tree %in% c("M", "L") & d$date != "date1"),]
```

##	tree	landsc	site	date	haraxy	adabip	adadec	calqua	caldec	exoqua	halsed
## 1	P	R	PR1	date1	0	3	0	0	0	0	0
## 2	P	R	PR2	date1	0	11	1	2	0	0	2
## 3	P	R	PR3	date1	0	2	3	0	0	1	0
## 4	P	S	PS1	date1	0	1	6	0	0	0	1
## 5	P	S	PS2	date1	0	7	2	0	0	4	1
## 6	P	S	PS3	date1	0	4	7	0	0	0	0
## 7	P	U	PU1	date1	16	67	13	0	0	0	0
## 8	P	U	PU2	date1	22	184	52	0	0	2	0
## 9	P	U	PU3	date1	2	212	82	0	0	0	0
## 10	L	R	LR1	date2	0	1	1	1	0	0	0
## 11	L	R	LR2	date2	0	9	5	0	0	0	0
## 12	L	R	LR3	date2	0	15	8	2	0	0	0
## 13	L	S	LS1	date2	1	21	4	2	2	0	0
## 14	L	S	LS2	date2	0	2	4	3	6	1	0
## 15	L	S	LS3	date2	0	3	2	2	2	0	0
## 16	L	U	LU1	date2	0	16	6	0	2	0	0
## 17	L	U	LU2	date2	2	55	0	3	0	0	0
## 18	L	U	LU3	date2	5	12	15	2	2	0	0
## 19	M	R	MR1	date2	0	3	1	3	0	0	0
## 20	M	R	MR2	date2	0	3	1	6	1	2	0
## 21	M	R	MR3	date2	1	15	2	13	0	1	0
## 22	M	S	MS1	date2	0	4	1	1	0	0	1
## 23	M	S	MS2	date2	0	1	0	2	2	0	2
## 24	M	S	MS3	date2	0	3	1	3	0	0	2
## 25	M	U	MU1	date2	6	24	2	1	0	2	0
## 26	M	U	MU2	date2	2	17	2	2	0	1	0
## 27	M	U	MU3	date2	2	4	0	2	0	0	1
## 37	L	R	LR1	date3	0	0	1	0	0	0	1
## 38	L	R	LR2	date3	0	4	6	1	6	0	0
## 39	L	R	LR3	date3	0	4	2	1	3	7	0
## 40	L	S	LS1	date3	1	13	7	3	8	1	0
## 41	L	S	LS2	date3	2	1	4	0	14	0	1
## 42	L	S	LS3	date3	0	1	0	1	2	0	0
## 43	L	U	LU1	date3	3	7	3	0	5	1	1
## 44	L	U	LU2	date3	9	62	2	2	0	0	0
## 45	L	U	LU3	date3	6	14	3	2	2	0	1
## 46	M	R	MR1	date3	0	0	0	0	0	0	0
## 47	M	R	MR2	date3	0	2	0	0	0	1	0
## 48	M	R	MR3	date3	0	3	1	0	2	0	0
## 49	M	S	MS1	date3	2	9	1	0	2	1	0
## 50	M	S	MS2	date3	0	2	0	1	6	1	2
## 51	M	S	MS3	date3	0	0	2	0	2	1	1
## 52	M	U	MU1	date3	2	9	1	0	0	0	0
## 53	M	U	MU2	date3	30	18	2	0	1	2	0
## 54	M	U	MU3	date3	0	3	0	0	0	0	0
## 64	L	R	LR1	date4	0	0	0	2	1	0	2
## 65	L	R	LR2	date4	0	0	2	3	0	0	0
## 66	L	R	LR3	date4	0	2	0	0	2	6	1
## 67	L	S	LS1	date4	2	15	2	1	5	0	0
## 68	L	S	LS2	date4	2	4	2	2	4	0	0
## 69	L	S	LS3	date4	0	0	0	1	2	0	0
## 70	L	U	LU1	date4	2	8	2	0	2	0	0
## 71	L	U	LU2	date4	6	48	1	1	0	15	0
## 72	L	U	LU3	date4	4	7	6	2	0	6	0
## 73	M	R	MR1	date4	0	0	0	0	0	0	0
## 74	M	R	MR2	date4	0	2	0	0	0	1	1
## 75	M	R	MR3	date4	1	8	2	1	0	0	0
## 76	M	S	MS1	date4	2	8	0	0	0	0	0
## 77	M	S	MS2	date4	2	0	0	0	2	0	0

## 78	M	S	MS3	date4	0	7	0	0	2	2	0
## 79	M	U	MU1	date4	10	27	2	1	2	6	0
## 80	M	U	MU2	date4	22	13	0	2	0	8	0
## 81	M	U	MU3	date4	1	0	0	0	0	0	0
##	oencon	harqua	myroct	aphobl	anaoce						
## 1	0	36	0	11	3						
## 2	0	24	3	2	0						
## 3	0	22	0	8	0						
## 4	0	21	43	15	3						
## 5	0	106	41	19	0						
## 6	0	35	1	11	0						
## 7	9	38	7	2	0						
## 8	0	22	12	0	0						
## 9	0	64	10	0	0						
## 10	0	0	0	0	0						
## 11	0	0	0	0	0						
## 12	0	0	0	0	0						
## 13	0	0	0	0	0						
## 14	0	0	0	0	0						
## 15	0	0	0	0	0						
## 16	4	0	0	0	0						
## 17	3	0	0	0	0						
## 18	1	0	0	0	0						
## 19	0	0	0	0	0						
## 20	0	0	0	0	0						
## 21	0	0	0	0	0						
## 22	1	0	0	0	0						
## 23	0	0	0	1	0						
## 24	0	0	0	0	0						
## 25	11	0	0	0	0						
## 26	7	0	0	0	0						
## 27	2	0	0	0	0						
## 37	0	0	0	0	0						
## 38	0	0	0	0	0						
## 39	0	0	0	0	0						
## 40	4	0	0	0	0						
## 41	0	0	0	0	0						
## 42	0	0	0	0	0						
## 43	2	0	0	0	0						
## 44	5	0	0	0	0						
## 45	1	0	0	0	0						
## 46	0	0	0	0	0						
## 47	0	0	0	0	0						
## 48	0	0	0	0	0						
## 49	0	0	0	0	0						
## 50	0	0	0	0	0						
## 51	0	0	0	0	0						
## 52	5	0	0	0	0						
## 53	3	0	0	0	0						
## 54	0	0	0	0	0						
## 64	0	0	0	0	0						
## 65	0	0	0	0	0						
## 66	0	0	0	0	0						
## 67	2	0	0	0	0						
## 68	0	0	0	0	0						
## 69	0	0	0	0	0						
## 70	0	0	0	0	0						
## 71	0	0	0	0	0						
## 72	0	0	0	0	0						
## 73	0	0	0	0	0						
## 74	0	0	0	0	0						
## 75	0	0	0	0	0						
## 76	2	0	0	0	0						

```
## 77      0      0      0      0      0
## 78      0      0      0      0      0
## 79     11      0      0      0      0
## 80      4      1      0      0      1
## 81      3      0      0      0      0
```

- 9) Sélectionnez les données collectées sur érable ou tilleul en milieu suburbain et pour les espèces haraxy, calqua, exoqua, harqua et aphobl

```
d[(d$tree == "M" | d$tree == "L") & d$landsc == "S",
  c("tree", "landsc", "date", "haraxy", "calqua", "exoqua", "harqua", "aphobl")]
```

```
##      tree landsc  date haraxy calqua exoqua harqua aphobl
## 13      L      S date2      1      2      0      0      0
## 14      L      S date2      0      3      1      0      0
## 15      L      S date2      0      2      0      0      0
## 22      M      S date2      0      1      0      0      0
## 23      M      S date2      0      2      0      0      1
## 24      M      S date2      0      3      0      0      0
## 40      L      S date3      1      3      1      0      0
## 41      L      S date3      2      0      0      0      0
## 42      L      S date3      0      1      0      0      0
## 49      M      S date3      2      0      1      0      0
## 50      M      S date3      0      1      1      0      0
## 51      M      S date3      0      0      1      0      0
## 67      L      S date4      2      1      0      0      0
## 68      L      S date4      2      2      0      0      0
## 69      L      S date4      0      1      0      0      0
## 76      M      S date4      2      0      0      0      0
## 77      M      S date4      2      0      0      0      0
## 78      M      S date4      0      0      2      0      0
```

- 10) Sélectionnez 10 lignes aléatoirement (à l'aide de la fonction sample)

```
d[sample(1:nrow(d), 10),]
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 34      P      U  PU1 date2      4     17      1      0      0      0      0
## 80      M      U  MU2 date4     22     13      0      2      0      8      0
## 21      M      R  MR3 date2      1     15      2     13      0      1      0
## 40      L      S  LS1 date3      1     13      7      3      8      1      0
## 7       P      U  PU1 date1     16     67     13      0      0      0      0
## 62      P      U  PU2 date3      0      8      6      0      0      0      0
## 29      P      R  PR2 date2      0      0      0      0      0      0      0
## 60      P      S  PS3 date3      0      1      0      0      0      0      0
## 15      L      S  LS3 date2      0      3      2      2      2      0      0
## 69      L      S  LS3 date4      0      0      0      1      2      0      0
##      oencon harqua myroct aphobl anaoce
## 34      4      9      5      0      0
## 80      4      1      0      0      1
## 21      0      0      0      0      0
## 40      4      0      0      0      0
## 7       9     38      7      2      0
## 62      0      0      6      0      0
## 29      0     13      5      0      0
## 60      0     47     21     13      5
## 15      0      0      0      0      0
## 69      0      0      0      0      0
```

- 11) Sélectionnez une ligne sur 5 (à l'aide de la fonction seq)

```
d[seq(1, nrow(d), 5),]
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 1      P      R  PR1 date1      0      3      0      0      0      0      0
## 6      P      S  PS3 date1      0      4      7      0      0      0      0
## 11     L      R  LR2 date2      0      9      5      0      0      0      0
## 16     L      U  LU1 date2      0     16      6      0      2      0      0
## 21     M      R  MR3 date2      1     15      2     13      0      1      0
## 26     M      U  MU2 date2      2     17      2      2      0      1      0
## 31     P      S  PS1 date2      0      0      0      0      0      0      0
## 36     P      U  PU3 date2      0     10      6      0      0      0      0
## 41     L      S  LS2 date3      2      1      4      0     14      0      1
## 46     M      R  MR1 date3      0      0      0      0      0      0      0
## 51     M      S  MS3 date3      0      0      2      0      2      1      1
## 56     P      R  PR2 date3      0      0      0      0      0      0      0
## 61     P      U  PU1 date3      1      3      0      0      0      0      0
## 66     L      R  LR3 date4      0      2      0      0      2      6      1
## 71     L      U  LU2 date4      6     48      1      1      0     15      0
## 76     M      S  MS1 date4      2      8      0      0      0      0      0
## 81     M      U  MU3 date4      1      0      0      0      0      0      0
##      oencon harqua myroct aphobl anaoce
## 1          0      36      0      11      3
## 6          0      35      1      11      0
## 11         0      0      0      0      0
## 16         4      0      0      0      0
## 21         0      0      0      0      0
## 26         7      0      0      0      0
## 31         0     13     10      3      0
## 36         0      4      2      0      0
## 41         0      0      0      0      0
## 46         0      0      0      0      0
## 51         0      0      0      0      0
## 56         0     16      9      2      1
## 61         1      3      5      0      0
## 66         0      0      0      0      0
## 71         0      0      0      0      0
## 76         2      0      0      0      0
## 81         3      0      0      0      0
```

- 12) Transformez les données d'abondance en données binaires (présence/absence). (n'écrasez pas le jeu de donnée)

```
sp <- d[, -c(1:4)]
sp[sp>0] <- 1
cbind(d[, 1:4], sp)
```

```
##      tree landsc site  date haraxy adabip adadec calqua caldec exoqua halsed
## 1      P      R  PR1 date1      0      1      0      0      0      0      0
## 2      P      R  PR2 date1      0      1      1      1      0      0      1
## 3      P      R  PR3 date1      0      1      1      0      0      1      0
## 4      P      S  PS1 date1      0      1      1      0      0      0      1
## 5      P      S  PS2 date1      0      1      1      0      0      1      1
## 6      P      S  PS3 date1      0      1      1      0      0      0      0
## 7      P      U  PU1 date1      1      1      1      0      0      0      0
## 8      P      U  PU2 date1      1      1      1      0      0      1      0
## 9      P      U  PU3 date1      1      1      1      0      0      0      0
## 10     L      R  LR1 date2      0      1      1      1      0      0      0
## 11     L      R  LR2 date2      0      1      1      0      0      0      0
## 12     L      R  LR3 date2      0      1      1      1      0      0      0
## 13     L      S  LS1 date2      1      1      1      1      1      0      0
```

## 14	L	S	LS2	date2	0	1	1	1	1	1	0
## 15	L	S	LS3	date2	0	1	1	1	1	0	0
## 16	L	U	LU1	date2	0	1	1	0	1	0	0
## 17	L	U	LU2	date2	1	1	0	1	0	0	0
## 18	L	U	LU3	date2	1	1	1	1	1	0	0
## 19	M	R	MR1	date2	0	1	1	1	0	0	0
## 20	M	R	MR2	date2	0	1	1	1	1	1	0
## 21	M	R	MR3	date2	1	1	1	1	0	1	0
## 22	M	S	MS1	date2	0	1	1	1	0	0	1
## 23	M	S	MS2	date2	0	1	0	1	1	0	1
## 24	M	S	MS3	date2	0	1	1	1	0	0	1
## 25	M	U	MU1	date2	1	1	1	1	0	1	0
## 26	M	U	MU2	date2	1	1	1	1	0	1	0
## 27	M	U	MU3	date2	1	1	0	1	0	0	1
## 28	P	R	PR1	date2	0	1	0	0	0	1	0
## 29	P	R	PR2	date2	0	0	0	0	0	0	0
## 30	P	R	PR3	date2	0	1	1	0	0	0	0
## 31	P	S	PS1	date2	0	0	0	0	0	0	0
## 32	P	S	PS2	date2	0	1	0	0	0	0	0
## 33	P	S	PS3	date2	0	1	1	0	0	1	0
## 34	P	U	PU1	date2	1	1	1	0	0	0	0
## 35	P	U	PU2	date2	1	1	1	0	0	0	0
## 36	P	U	PU3	date2	0	1	1	0	0	0	0
## 37	L	R	LR1	date3	0	0	1	0	0	0	1
## 38	L	R	LR2	date3	0	1	1	1	1	0	0
## 39	L	R	LR3	date3	0	1	1	1	1	1	0
## 40	L	S	LS1	date3	1	1	1	1	1	1	0
## 41	L	S	LS2	date3	1	1	1	0	1	0	1
## 42	L	S	LS3	date3	0	1	0	1	1	0	0
## 43	L	U	LU1	date3	1	1	1	0	1	1	1
## 44	L	U	LU2	date3	1	1	1	1	0	0	0
## 45	L	U	LU3	date3	1	1	1	1	1	0	1
## 46	M	R	MR1	date3	0	0	0	0	0	0	0
## 47	M	R	MR2	date3	0	1	0	0	0	1	0
## 48	M	R	MR3	date3	0	1	1	0	1	0	0
## 49	M	S	MS1	date3	1	1	1	0	1	1	0
## 50	M	S	MS2	date3	0	1	0	1	1	1	1
## 51	M	S	MS3	date3	0	0	1	0	1	1	1
## 52	M	U	MU1	date3	1	1	1	0	0	0	0
## 53	M	U	MU2	date3	1	1	1	0	1	1	0
## 54	M	U	MU3	date3	0	1	0	0	0	0	0
## 55	P	R	PR1	date3	0	0	0	0	0	1	0
## 56	P	R	PR2	date3	0	0	0	0	0	0	0
## 57	P	R	PR3	date3	0	0	1	0	0	0	0
## 58	P	S	PS1	date3	0	1	0	0	0	0	0
## 59	P	S	PS2	date3	0	0	0	0	0	0	0
## 60	P	S	PS3	date3	0	1	0	0	0	0	0
## 61	P	U	PU1	date3	1	1	0	0	0	0	0
## 62	P	U	PU2	date3	0	1	1	0	0	0	0
## 63	P	U	PU3	date3	0	1	1	0	0	0	0
## 64	L	R	LR1	date4	0	0	0	1	1	0	1
## 65	L	R	LR2	date4	0	0	1	1	0	0	0
## 66	L	R	LR3	date4	0	1	0	0	1	1	1
## 67	L	S	LS1	date4	1	1	1	1	1	0	0
## 68	L	S	LS2	date4	1	1	1	1	1	0	0
## 69	L	S	LS3	date4	0	0	0	1	1	0	0
## 70	L	U	LU1	date4	1	1	1	0	1	0	0
## 71	L	U	LU2	date4	1	1	1	1	0	1	0
## 72	L	U	LU3	date4	1	1	1	1	0	1	0
## 73	M	R	MR1	date4	0	0	0	0	0	0	0
## 74	M	R	MR2	date4	0	1	0	0	0	1	1
## 75	M	R	MR3	date4	1	1	1	1	0	0	0
## 76	M	S	MS1	date4	1	1	0	0	0	0	0

## 77	M	S	MS2 date4	1	0	0	0	1	0	0
## 78	M	S	MS3 date4	0	1	0	0	1	1	0
## 79	M	U	MU1 date4	1	1	1	1	1	1	0
## 80	M	U	MU2 date4	1	1	0	1	0	1	0
## 81	M	U	MU3 date4	1	0	0	0	0	0	0
##	oencon	harqua	myroct	aphobl	anaoce					
## 1	0	1	0	1	1					
## 2	0	1	1	1	0					
## 3	0	1	0	1	0					
## 4	0	1	1	1	1					
## 5	0	1	1	1	0					
## 6	0	1	1	1	0					
## 7	1	1	1	1	0					
## 8	0	1	1	0	0					
## 9	0	1	1	0	0					
## 10	0	0	0	0	0					
## 11	0	0	0	0	0					
## 12	0	0	0	0	0					
## 13	0	0	0	0	0					
## 14	0	0	0	0	0					
## 15	0	0	0	0	0					
## 16	1	0	0	0	0					
## 17	1	0	0	0	0					
## 18	1	0	0	0	0					
## 19	0	0	0	0	0					
## 20	0	0	0	0	0					
## 21	0	0	0	0	0					
## 22	1	0	0	0	0					
## 23	0	0	0	1	0					
## 24	0	0	0	0	0					
## 25	1	0	0	0	0					
## 26	1	0	0	0	0					
## 27	1	0	0	0	0					
## 28	0	1	1	0	0					
## 29	0	1	1	0	0					
## 30	0	1	0	1	0					
## 31	0	1	1	1	0					
## 32	0	1	1	1	0					
## 33	0	1	1	1	1					
## 34	1	1	1	0	0					
## 35	0	1	1	1	0					
## 36	0	1	1	0	0					
## 37	0	0	0	0	0					
## 38	0	0	0	0	0					
## 39	0	0	0	0	0					
## 40	1	0	0	0	0					
## 41	0	0	0	0	0					
## 42	0	0	0	0	0					
## 43	1	0	0	0	0					
## 44	1	0	0	0	0					
## 45	1	0	0	0	0					
## 46	0	0	0	0	0					
## 47	0	0	0	0	0					
## 48	0	0	0	0	0					
## 49	0	0	0	0	0					
## 50	0	0	0	0	0					
## 51	0	0	0	0	0					
## 52	1	0	0	0	0					
## 53	1	0	0	0	0					
## 54	0	0	0	0	0					
## 55	0	1	1	1	1					
## 56	0	1	1	1	1					
## 57	0	1	0	1	1					

```
## 58      0      1      1      1      0
## 59      0      1      1      1      0
## 60      0      1      1      1      1
## 61      1      1      1      0      0
## 62      0      0      1      0      0
## 63      0      1      1      0      0
## 64      0      0      0      0      0
## 65      0      0      0      0      0
## 66      0      0      0      0      0
## 67      1      0      0      0      0
## 68      0      0      0      0      0
## 69      0      0      0      0      0
## 70      0      0      0      0      0
## 71      0      0      0      0      0
## 72      0      0      0      0      0
## 73      0      0      0      0      0
## 74      0      0      0      0      0
## 75      0      0      0      0      0
## 76      1      0      0      0      0
## 77      0      0      0      0      0
## 78      0      0      0      0      0
## 79      1      0      0      0      0
## 80      1      1      0      0      1
## 81      1      0      0      0      0
```

- 13) Remplacez les données de plus de 150 individus par des valeurs manquantes

```
# NB dans ce cas les messages d'avertissement (warnings) sont sans importance
d[d>150] <- NA
```

```
## Warning in Ops.factor(left, right): '>' not meaningful for factors
```

```
## Warning in Ops.factor(left, right): '>' not meaningful for factors
```

```
## Warning in Ops.factor(left, right): '>' not meaningful for factors
```

```
## Warning in Ops.factor(left, right): '>' not meaningful for factors
```

- 14) Remplacez les données d'abondance d'espèce du site PS1 à la date 3 par des valeurs manquantes

```
d[d$site == "PS1" & d$date == "date3", -c(1:4)] <- NA
```

- 15) Multipliez les valeurs des sites sur pins par 5/4

```
d[d$tree == "P", -c(1:4)] <- d[d$tree == "P", -c(1:4)] * 5/4
```

- 16) Réordonnez le jeu de données de manière à ce qu'on ait d'abord la colonne date puis site puis landsc puis tree suivies des colonnes espèces

```
d[, c(4:1, 5: ncol(d))]
```

```
##      date site landsc tree haraxy adabip adadec calqua caldec exoqua halsed
## 1 date1 PR1      R   P   0.00   3.75   0.00   0.0      0   0.00   0.00
## 2 date1 PR2      R   P   0.00  13.75   1.25   2.5      0   0.00   2.50
## 3 date1 PR3      R   P   0.00   2.50   3.75   0.0      0   1.25   0.00
## 4 date1 PS1      S   P   0.00   1.25   7.50   0.0      0   0.00   1.25
```

## 5	date1	PS2	S	P	0.00	8.75	2.50	0.0	0	5.00	1.25
## 6	date1	PS3	S	P	0.00	5.00	8.75	0.0	0	0.00	0.00
## 7	date1	PU1	U	P	20.00	83.75	16.25	0.0	0	0.00	0.00
## 8	date1	PU2	U	P	27.50	NA	65.00	0.0	0	2.50	0.00
## 9	date1	PU3	U	P	2.50	NA	102.50	0.0	0	0.00	0.00
## 10	date2	LR1	R	L	0.00	1.00	1.00	1.0	0	0.00	0.00
## 11	date2	LR2	R	L	0.00	9.00	5.00	0.0	0	0.00	0.00
## 12	date2	LR3	R	L	0.00	15.00	8.00	2.0	0	0.00	0.00
## 13	date2	LS1	S	L	1.00	21.00	4.00	2.0	2	0.00	0.00
## 14	date2	LS2	S	L	0.00	2.00	4.00	3.0	6	1.00	0.00
## 15	date2	LS3	S	L	0.00	3.00	2.00	2.0	2	0.00	0.00
## 16	date2	LU1	U	L	0.00	16.00	6.00	0.0	2	0.00	0.00
## 17	date2	LU2	U	L	2.00	55.00	0.00	3.0	0	0.00	0.00
## 18	date2	LU3	U	L	5.00	12.00	15.00	2.0	2	0.00	0.00
## 19	date2	MR1	R	M	0.00	3.00	1.00	3.0	0	0.00	0.00
## 20	date2	MR2	R	M	0.00	3.00	1.00	6.0	1	2.00	0.00
## 21	date2	MR3	R	M	1.00	15.00	2.00	13.0	0	1.00	0.00
## 22	date2	MS1	S	M	0.00	4.00	1.00	1.0	0	0.00	1.00
## 23	date2	MS2	S	M	0.00	1.00	0.00	2.0	2	0.00	2.00
## 24	date2	MS3	S	M	0.00	3.00	1.00	3.0	0	0.00	2.00
## 25	date2	MU1	U	M	6.00	24.00	2.00	1.0	0	2.00	0.00
## 26	date2	MU2	U	M	2.00	17.00	2.00	2.0	0	1.00	0.00
## 27	date2	MU3	U	M	2.00	4.00	0.00	2.0	0	0.00	1.00
## 28	date2	PR1	R	P	0.00	3.75	0.00	0.0	0	1.25	0.00
## 29	date2	PR2	R	P	0.00	0.00	0.00	0.0	0	0.00	0.00
## 30	date2	PR3	R	P	0.00	2.50	2.50	0.0	0	0.00	0.00
## 31	date2	PS1	S	P	0.00	0.00	0.00	0.0	0	0.00	0.00
## 32	date2	PS2	S	P	0.00	3.75	0.00	0.0	0	0.00	0.00
## 33	date2	PS3	S	P	0.00	3.75	1.25	0.0	0	3.75	0.00
## 34	date2	PU1	U	P	5.00	21.25	1.25	0.0	0	0.00	0.00
## 35	date2	PU2	U	P	7.50	40.00	7.50	0.0	0	0.00	0.00
## 36	date2	PU3	U	P	0.00	12.50	7.50	0.0	0	0.00	0.00
## 37	date3	LR1	R	L	0.00	0.00	1.00	0.0	0	0.00	1.00
## 38	date3	LR2	R	L	0.00	4.00	6.00	1.0	6	0.00	0.00
## 39	date3	LR3	R	L	0.00	4.00	2.00	1.0	3	7.00	0.00
## 40	date3	LS1	S	L	1.00	13.00	7.00	3.0	8	1.00	0.00
## 41	date3	LS2	S	L	2.00	1.00	4.00	0.0	14	0.00	1.00
## 42	date3	LS3	S	L	0.00	1.00	0.00	1.0	2	0.00	0.00
## 43	date3	LU1	U	L	3.00	7.00	3.00	0.0	5	1.00	1.00
## 44	date3	LU2	U	L	9.00	62.00	2.00	2.0	0	0.00	0.00
## 45	date3	LU3	U	L	6.00	14.00	3.00	2.0	2	0.00	1.00
## 46	date3	MR1	R	M	0.00	0.00	0.00	0.0	0	0.00	0.00
## 47	date3	MR2	R	M	0.00	2.00	0.00	0.0	0	1.00	0.00
## 48	date3	MR3	R	M	0.00	3.00	1.00	0.0	2	0.00	0.00
## 49	date3	MS1	S	M	2.00	9.00	1.00	0.0	2	1.00	0.00
## 50	date3	MS2	S	M	0.00	2.00	0.00	1.0	6	1.00	2.00
## 51	date3	MS3	S	M	0.00	0.00	2.00	0.0	2	1.00	1.00
## 52	date3	MU1	U	M	2.00	9.00	1.00	0.0	0	0.00	0.00
## 53	date3	MU2	U	M	30.00	18.00	2.00	0.0	1	2.00	0.00
## 54	date3	MU3	U	M	0.00	3.00	0.00	0.0	0	0.00	0.00
## 55	date3	PR1	R	P	0.00	0.00	0.00	0.0	0	5.00	0.00
## 56	date3	PR2	R	P	0.00	0.00	0.00	0.0	0	0.00	0.00
## 57	date3	PR3	R	P	0.00	0.00	6.25	0.0	0	0.00	0.00
## 58	date3	PS1	S	P	NA	NA	NA	NA	NA	NA	NA
## 59	date3	PS2	S	P	0.00	0.00	0.00	0.0	0	0.00	0.00
## 60	date3	PS3	S	P	0.00	1.25	0.00	0.0	0	0.00	0.00
## 61	date3	PU1	U	P	1.25	3.75	0.00	0.0	0	0.00	0.00
## 62	date3	PU2	U	P	0.00	10.00	7.50	0.0	0	0.00	0.00
## 63	date3	PU3	U	P	0.00	7.50	5.00	0.0	0	0.00	0.00
## 64	date4	LR1	R	L	0.00	0.00	0.00	2.0	1	0.00	2.00
## 65	date4	LR2	R	L	0.00	0.00	2.00	3.0	0	0.00	0.00
## 66	date4	LR3	R	L	0.00	2.00	0.00	0.0	2	6.00	1.00
## 67	date4	LS1	S	L	2.00	15.00	2.00	1.0	5	0.00	0.00

## 68	date4	LS2	S	L	2.00	4.00	2.00	2.0	4	0.00	0.00
## 69	date4	LS3	S	L	0.00	0.00	0.00	1.0	2	0.00	0.00
## 70	date4	LU1	U	L	2.00	8.00	2.00	0.0	2	0.00	0.00
## 71	date4	LU2	U	L	6.00	48.00	1.00	1.0	0	15.00	0.00
## 72	date4	LU3	U	L	4.00	7.00	6.00	2.0	0	6.00	0.00
## 73	date4	MR1	R	M	0.00	0.00	0.00	0.0	0	0.00	0.00
## 74	date4	MR2	R	M	0.00	2.00	0.00	0.0	0	1.00	1.00
## 75	date4	MR3	R	M	1.00	8.00	2.00	1.0	0	0.00	0.00
## 76	date4	MS1	S	M	2.00	8.00	0.00	0.0	0	0.00	0.00
## 77	date4	MS2	S	M	2.00	0.00	0.00	0.0	2	0.00	0.00
## 78	date4	MS3	S	M	0.00	7.00	0.00	0.0	2	2.00	0.00
## 79	date4	MU1	U	M	10.00	27.00	2.00	1.0	2	6.00	0.00
## 80	date4	MU2	U	M	22.00	13.00	0.00	2.0	0	8.00	0.00
## 81	date4	MU3	U	M	1.00	0.00	0.00	0.0	0	0.00	0.00
##	oencon	harqua	myroct	aphobl	anaoce						
## 1	0.00	45.00	0.00	13.75	3.75						
## 2	0.00	30.00	3.75	2.50	0.00						
## 3	0.00	27.50	0.00	10.00	0.00						
## 4	0.00	26.25	53.75	18.75	3.75						
## 5	0.00	132.50	51.25	23.75	0.00						
## 6	0.00	43.75	1.25	13.75	0.00						
## 7	11.25	47.50	8.75	2.50	0.00						
## 8	0.00	27.50	15.00	0.00	0.00						
## 9	0.00	80.00	12.50	0.00	0.00						
## 10	0.00	0.00	0.00	0.00	0.00						
## 11	0.00	0.00	0.00	0.00	0.00						
## 12	0.00	0.00	0.00	0.00	0.00						
## 13	0.00	0.00	0.00	0.00	0.00						
## 14	0.00	0.00	0.00	0.00	0.00						
## 15	0.00	0.00	0.00	0.00	0.00						
## 16	4.00	0.00	0.00	0.00	0.00						
## 17	3.00	0.00	0.00	0.00	0.00						
## 18	1.00	0.00	0.00	0.00	0.00						
## 19	0.00	0.00	0.00	0.00	0.00						
## 20	0.00	0.00	0.00	0.00	0.00						
## 21	0.00	0.00	0.00	0.00	0.00						
## 22	1.00	0.00	0.00	0.00	0.00						
## 23	0.00	0.00	0.00	1.00	0.00						
## 24	0.00	0.00	0.00	0.00	0.00						
## 25	11.00	0.00	0.00	0.00	0.00						
## 26	7.00	0.00	0.00	0.00	0.00						
## 27	2.00	0.00	0.00	0.00	0.00						
## 28	0.00	21.25	1.25	0.00	0.00						
## 29	0.00	16.25	6.25	0.00	0.00						
## 30	0.00	17.50	0.00	3.75	0.00						
## 31	0.00	16.25	12.50	3.75	0.00						
## 32	0.00	45.00	67.50	7.50	0.00						
## 33	0.00	25.00	8.75	11.25	3.75						
## 34	5.00	11.25	6.25	0.00	0.00						
## 35	0.00	12.50	2.50	12.50	0.00						
## 36	0.00	5.00	2.50	0.00	0.00						
## 37	0.00	0.00	0.00	0.00	0.00						
## 38	0.00	0.00	0.00	0.00	0.00						
## 39	0.00	0.00	0.00	0.00	0.00						
## 40	4.00	0.00	0.00	0.00	0.00						
## 41	0.00	0.00	0.00	0.00	0.00						
## 42	0.00	0.00	0.00	0.00	0.00						
## 43	2.00	0.00	0.00	0.00	0.00						
## 44	5.00	0.00	0.00	0.00	0.00						
## 45	1.00	0.00	0.00	0.00	0.00						
## 46	0.00	0.00	0.00	0.00	0.00						
## 47	0.00	0.00	0.00	0.00	0.00						
## 48	0.00	0.00	0.00	0.00	0.00						

```
## 49 0.00 0.00 0.00 0.00 0.00
## 50 0.00 0.00 0.00 0.00 0.00
## 51 0.00 0.00 0.00 0.00 0.00
## 52 5.00 0.00 0.00 0.00 0.00
## 53 3.00 0.00 0.00 0.00 0.00
## 54 0.00 0.00 0.00 0.00 0.00
## 55 0.00 17.50 2.50 7.50 2.50
## 56 0.00 20.00 11.25 2.50 1.25
## 57 0.00 25.00 0.00 15.00 2.50
## 58 NA NA NA NA NA
## 59 0.00 18.75 55.00 36.25 0.00
## 60 0.00 58.75 26.25 16.25 6.25
## 61 1.25 3.75 6.25 0.00 0.00
## 62 0.00 0.00 7.50 0.00 0.00
## 63 0.00 5.00 12.50 0.00 0.00
## 64 0.00 0.00 0.00 0.00 0.00
## 65 0.00 0.00 0.00 0.00 0.00
## 66 0.00 0.00 0.00 0.00 0.00
## 67 2.00 0.00 0.00 0.00 0.00
## 68 0.00 0.00 0.00 0.00 0.00
## 69 0.00 0.00 0.00 0.00 0.00
## 70 0.00 0.00 0.00 0.00 0.00
## 71 0.00 0.00 0.00 0.00 0.00
## 72 0.00 0.00 0.00 0.00 0.00
## 73 0.00 0.00 0.00 0.00 0.00
## 74 0.00 0.00 0.00 0.00 0.00
## 75 0.00 0.00 0.00 0.00 0.00
## 76 2.00 0.00 0.00 0.00 0.00
## 77 0.00 0.00 0.00 0.00 0.00
## 78 0.00 0.00 0.00 0.00 0.00
## 79 11.00 0.00 0.00 0.00 0.00
## 80 4.00 1.00 0.00 0.00 1.00
## 81 3.00 0.00 0.00 0.00 0.00
```

- 17) Ne gardez dans le jeu de données que les espèces pour lesquelles on a plus de 100 individus observés au total. (utilisez la fonction colSums)

```
colSums(d[, -c(1:4)])
```

```
## haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## NA NA NA NA NA NA NA NA NA NA
## aphobl anaoce
## NA NA
```

```
colSums(d[, -c(1:4)], na.rm = TRUE)
```

```
## haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## 191.75 737.75 355.25 77.50 92.00 84.75 22.00 88.50 779.75 375.00
## aphobl anaoce
## 202.25 24.75
```

```
colSums(d[, -c(1:4)], na.rm = TRUE) > 100
```

```
## haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
## aphobl anaoce
## TRUE FALSE
```

```
# plusieurs solutions :
spp <- colSums(d[,-c(1:4)], na.rm = TRUE) >100
spp

## haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
## aphobl anaoce
## TRUE FALSE
```

```
d[, c(TRUE, TRUE, TRUE, TRUE, spp)]
```

	tree	landsc	site	date	haraxy	adabip	adadec	harqua	myroct	aphobl
## 1	P	R	PR1	date1	0.00	3.75	0.00	45.00	0.00	13.75
## 2	P	R	PR2	date1	0.00	13.75	1.25	30.00	3.75	2.50
## 3	P	R	PR3	date1	0.00	2.50	3.75	27.50	0.00	10.00
## 4	P	S	PS1	date1	0.00	1.25	7.50	26.25	53.75	18.75
## 5	P	S	PS2	date1	0.00	8.75	2.50	132.50	51.25	23.75
## 6	P	S	PS3	date1	0.00	5.00	8.75	43.75	1.25	13.75
## 7	P	U	PU1	date1	20.00	83.75	16.25	47.50	8.75	2.50
## 8	P	U	PU2	date1	27.50	NA	65.00	27.50	15.00	0.00
## 9	P	U	PU3	date1	2.50	NA	102.50	80.00	12.50	0.00
## 10	L	R	LR1	date2	0.00	1.00	1.00	0.00	0.00	0.00
## 11	L	R	LR2	date2	0.00	9.00	5.00	0.00	0.00	0.00
## 12	L	R	LR3	date2	0.00	15.00	8.00	0.00	0.00	0.00
## 13	L	S	LS1	date2	1.00	21.00	4.00	0.00	0.00	0.00
## 14	L	S	LS2	date2	0.00	2.00	4.00	0.00	0.00	0.00
## 15	L	S	LS3	date2	0.00	3.00	2.00	0.00	0.00	0.00
## 16	L	U	LU1	date2	0.00	16.00	6.00	0.00	0.00	0.00
## 17	L	U	LU2	date2	2.00	55.00	0.00	0.00	0.00	0.00
## 18	L	U	LU3	date2	5.00	12.00	15.00	0.00	0.00	0.00
## 19	M	R	MR1	date2	0.00	3.00	1.00	0.00	0.00	0.00
## 20	M	R	MR2	date2	0.00	3.00	1.00	0.00	0.00	0.00
## 21	M	R	MR3	date2	1.00	15.00	2.00	0.00	0.00	0.00
## 22	M	S	MS1	date2	0.00	4.00	1.00	0.00	0.00	0.00
## 23	M	S	MS2	date2	0.00	1.00	0.00	0.00	0.00	1.00
## 24	M	S	MS3	date2	0.00	3.00	1.00	0.00	0.00	0.00
## 25	M	U	MU1	date2	6.00	24.00	2.00	0.00	0.00	0.00
## 26	M	U	MU2	date2	2.00	17.00	2.00	0.00	0.00	0.00
## 27	M	U	MU3	date2	2.00	4.00	0.00	0.00	0.00	0.00
## 28	P	R	PR1	date2	0.00	3.75	0.00	21.25	1.25	0.00
## 29	P	R	PR2	date2	0.00	0.00	0.00	16.25	6.25	0.00
## 30	P	R	PR3	date2	0.00	2.50	2.50	17.50	0.00	3.75
## 31	P	S	PS1	date2	0.00	0.00	0.00	16.25	12.50	3.75
## 32	P	S	PS2	date2	0.00	3.75	0.00	45.00	67.50	7.50
## 33	P	S	PS3	date2	0.00	3.75	1.25	25.00	8.75	11.25
## 34	P	U	PU1	date2	5.00	21.25	1.25	11.25	6.25	0.00
## 35	P	U	PU2	date2	7.50	40.00	7.50	12.50	2.50	12.50
## 36	P	U	PU3	date2	0.00	12.50	7.50	5.00	2.50	0.00
## 37	L	R	LR1	date3	0.00	0.00	1.00	0.00	0.00	0.00
## 38	L	R	LR2	date3	0.00	4.00	6.00	0.00	0.00	0.00
## 39	L	R	LR3	date3	0.00	4.00	2.00	0.00	0.00	0.00
## 40	L	S	LS1	date3	1.00	13.00	7.00	0.00	0.00	0.00
## 41	L	S	LS2	date3	2.00	1.00	4.00	0.00	0.00	0.00
## 42	L	S	LS3	date3	0.00	1.00	0.00	0.00	0.00	0.00
## 43	L	U	LU1	date3	3.00	7.00	3.00	0.00	0.00	0.00
## 44	L	U	LU2	date3	9.00	62.00	2.00	0.00	0.00	0.00
## 45	L	U	LU3	date3	6.00	14.00	3.00	0.00	0.00	0.00
## 46	M	R	MR1	date3	0.00	0.00	0.00	0.00	0.00	0.00
## 47	M	R	MR2	date3	0.00	2.00	0.00	0.00	0.00	0.00
## 48	M	R	MR3	date3	0.00	3.00	1.00	0.00	0.00	0.00
## 49	M	S	MS1	date3	2.00	9.00	1.00	0.00	0.00	0.00

## 50	M	S	MS2	date3	0.00	2.00	0.00	0.00	0.00	0.00
## 51	M	S	MS3	date3	0.00	0.00	2.00	0.00	0.00	0.00
## 52	M	U	MU1	date3	2.00	9.00	1.00	0.00	0.00	0.00
## 53	M	U	MU2	date3	30.00	18.00	2.00	0.00	0.00	0.00
## 54	M	U	MU3	date3	0.00	3.00	0.00	0.00	0.00	0.00
## 55	P	R	PR1	date3	0.00	0.00	0.00	17.50	2.50	7.50
## 56	P	R	PR2	date3	0.00	0.00	0.00	20.00	11.25	2.50
## 57	P	R	PR3	date3	0.00	0.00	6.25	25.00	0.00	15.00
## 58	P	S	PS1	date3	NA	NA	NA	NA	NA	NA
## 59	P	S	PS2	date3	0.00	0.00	0.00	18.75	55.00	36.25
## 60	P	S	PS3	date3	0.00	1.25	0.00	58.75	26.25	16.25
## 61	P	U	PU1	date3	1.25	3.75	0.00	3.75	6.25	0.00
## 62	P	U	PU2	date3	0.00	10.00	7.50	0.00	7.50	0.00
## 63	P	U	PU3	date3	0.00	7.50	5.00	5.00	12.50	0.00
## 64	L	R	LR1	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 65	L	R	LR2	date4	0.00	0.00	2.00	0.00	0.00	0.00
## 66	L	R	LR3	date4	0.00	2.00	0.00	0.00	0.00	0.00
## 67	L	S	LS1	date4	2.00	15.00	2.00	0.00	0.00	0.00
## 68	L	S	LS2	date4	2.00	4.00	2.00	0.00	0.00	0.00
## 69	L	S	LS3	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 70	L	U	LU1	date4	2.00	8.00	2.00	0.00	0.00	0.00
## 71	L	U	LU2	date4	6.00	48.00	1.00	0.00	0.00	0.00
## 72	L	U	LU3	date4	4.00	7.00	6.00	0.00	0.00	0.00
## 73	M	R	MR1	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 74	M	R	MR2	date4	0.00	2.00	0.00	0.00	0.00	0.00
## 75	M	R	MR3	date4	1.00	8.00	2.00	0.00	0.00	0.00
## 76	M	S	MS1	date4	2.00	8.00	0.00	0.00	0.00	0.00
## 77	M	S	MS2	date4	2.00	0.00	0.00	0.00	0.00	0.00
## 78	M	S	MS3	date4	0.00	7.00	0.00	0.00	0.00	0.00
## 79	M	U	MU1	date4	10.00	27.00	2.00	0.00	0.00	0.00
## 80	M	U	MU2	date4	22.00	13.00	0.00	1.00	0.00	0.00
## 81	M	U	MU3	date4	1.00	0.00	0.00	0.00	0.00	0.00

```
cbind(d[,1:4], d[, -c(1:4)][, spp])
```

##	tree	landsc	site	date	haraxy	adabip	adadec	harqua	myroct	aphobl
## 1	P	R	PR1	date1	0.00	3.75	0.00	45.00	0.00	13.75
## 2	P	R	PR2	date1	0.00	13.75	1.25	30.00	3.75	2.50
## 3	P	R	PR3	date1	0.00	2.50	3.75	27.50	0.00	10.00
## 4	P	S	PS1	date1	0.00	1.25	7.50	26.25	53.75	18.75
## 5	P	S	PS2	date1	0.00	8.75	2.50	132.50	51.25	23.75
## 6	P	S	PS3	date1	0.00	5.00	8.75	43.75	1.25	13.75
## 7	P	U	PU1	date1	20.00	83.75	16.25	47.50	8.75	2.50
## 8	P	U	PU2	date1	27.50	NA	65.00	27.50	15.00	0.00
## 9	P	U	PU3	date1	2.50	NA	102.50	80.00	12.50	0.00
## 10	L	R	LR1	date2	0.00	1.00	1.00	0.00	0.00	0.00
## 11	L	R	LR2	date2	0.00	9.00	5.00	0.00	0.00	0.00
## 12	L	R	LR3	date2	0.00	15.00	8.00	0.00	0.00	0.00
## 13	L	S	LS1	date2	1.00	21.00	4.00	0.00	0.00	0.00
## 14	L	S	LS2	date2	0.00	2.00	4.00	0.00	0.00	0.00
## 15	L	S	LS3	date2	0.00	3.00	2.00	0.00	0.00	0.00
## 16	L	U	LU1	date2	0.00	16.00	6.00	0.00	0.00	0.00
## 17	L	U	LU2	date2	2.00	55.00	0.00	0.00	0.00	0.00
## 18	L	U	LU3	date2	5.00	12.00	15.00	0.00	0.00	0.00
## 19	M	R	MR1	date2	0.00	3.00	1.00	0.00	0.00	0.00
## 20	M	R	MR2	date2	0.00	3.00	1.00	0.00	0.00	0.00
## 21	M	R	MR3	date2	1.00	15.00	2.00	0.00	0.00	0.00
## 22	M	S	MS1	date2	0.00	4.00	1.00	0.00	0.00	0.00
## 23	M	S	MS2	date2	0.00	1.00	0.00	0.00	0.00	1.00
## 24	M	S	MS3	date2	0.00	3.00	1.00	0.00	0.00	0.00
## 25	M	U	MU1	date2	6.00	24.00	2.00	0.00	0.00	0.00
## 26	M	U	MU2	date2	2.00	17.00	2.00	0.00	0.00	0.00

## 27	M	U	MU3	date2	2.00	4.00	0.00	0.00	0.00	0.00
## 28	P	R	PR1	date2	0.00	3.75	0.00	21.25	1.25	0.00
## 29	P	R	PR2	date2	0.00	0.00	0.00	16.25	6.25	0.00
## 30	P	R	PR3	date2	0.00	2.50	2.50	17.50	0.00	3.75
## 31	P	S	PS1	date2	0.00	0.00	0.00	16.25	12.50	3.75
## 32	P	S	PS2	date2	0.00	3.75	0.00	45.00	67.50	7.50
## 33	P	S	PS3	date2	0.00	3.75	1.25	25.00	8.75	11.25
## 34	P	U	PU1	date2	5.00	21.25	1.25	11.25	6.25	0.00
## 35	P	U	PU2	date2	7.50	40.00	7.50	12.50	2.50	12.50
## 36	P	U	PU3	date2	0.00	12.50	7.50	5.00	2.50	0.00
## 37	L	R	LR1	date3	0.00	0.00	1.00	0.00	0.00	0.00
## 38	L	R	LR2	date3	0.00	4.00	6.00	0.00	0.00	0.00
## 39	L	R	LR3	date3	0.00	4.00	2.00	0.00	0.00	0.00
## 40	L	S	LS1	date3	1.00	13.00	7.00	0.00	0.00	0.00
## 41	L	S	LS2	date3	2.00	1.00	4.00	0.00	0.00	0.00
## 42	L	S	LS3	date3	0.00	1.00	0.00	0.00	0.00	0.00
## 43	L	U	LU1	date3	3.00	7.00	3.00	0.00	0.00	0.00
## 44	L	U	LU2	date3	9.00	62.00	2.00	0.00	0.00	0.00
## 45	L	U	LU3	date3	6.00	14.00	3.00	0.00	0.00	0.00
## 46	M	R	MR1	date3	0.00	0.00	0.00	0.00	0.00	0.00
## 47	M	R	MR2	date3	0.00	2.00	0.00	0.00	0.00	0.00
## 48	M	R	MR3	date3	0.00	3.00	1.00	0.00	0.00	0.00
## 49	M	S	MS1	date3	2.00	9.00	1.00	0.00	0.00	0.00
## 50	M	S	MS2	date3	0.00	2.00	0.00	0.00	0.00	0.00
## 51	M	S	MS3	date3	0.00	0.00	2.00	0.00	0.00	0.00
## 52	M	U	MU1	date3	2.00	9.00	1.00	0.00	0.00	0.00
## 53	M	U	MU2	date3	30.00	18.00	2.00	0.00	0.00	0.00
## 54	M	U	MU3	date3	0.00	3.00	0.00	0.00	0.00	0.00
## 55	P	R	PR1	date3	0.00	0.00	0.00	17.50	2.50	7.50
## 56	P	R	PR2	date3	0.00	0.00	0.00	20.00	11.25	2.50
## 57	P	R	PR3	date3	0.00	0.00	6.25	25.00	0.00	15.00
## 58	P	S	PS1	date3	NA	NA	NA	NA	NA	NA
## 59	P	S	PS2	date3	0.00	0.00	0.00	18.75	55.00	36.25
## 60	P	S	PS3	date3	0.00	1.25	0.00	58.75	26.25	16.25
## 61	P	U	PU1	date3	1.25	3.75	0.00	3.75	6.25	0.00
## 62	P	U	PU2	date3	0.00	10.00	7.50	0.00	7.50	0.00
## 63	P	U	PU3	date3	0.00	7.50	5.00	5.00	12.50	0.00
## 64	L	R	LR1	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 65	L	R	LR2	date4	0.00	0.00	2.00	0.00	0.00	0.00
## 66	L	R	LR3	date4	0.00	2.00	0.00	0.00	0.00	0.00
## 67	L	S	LS1	date4	2.00	15.00	2.00	0.00	0.00	0.00
## 68	L	S	LS2	date4	2.00	4.00	2.00	0.00	0.00	0.00
## 69	L	S	LS3	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 70	L	U	LU1	date4	2.00	8.00	2.00	0.00	0.00	0.00
## 71	L	U	LU2	date4	6.00	48.00	1.00	0.00	0.00	0.00
## 72	L	U	LU3	date4	4.00	7.00	6.00	0.00	0.00	0.00
## 73	M	R	MR1	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 74	M	R	MR2	date4	0.00	2.00	0.00	0.00	0.00	0.00
## 75	M	R	MR3	date4	1.00	8.00	2.00	0.00	0.00	0.00
## 76	M	S	MS1	date4	2.00	8.00	0.00	0.00	0.00	0.00
## 77	M	S	MS2	date4	2.00	0.00	0.00	0.00	0.00	0.00
## 78	M	S	MS3	date4	0.00	7.00	0.00	0.00	0.00	0.00
## 79	M	U	MU1	date4	10.00	27.00	2.00	0.00	0.00	0.00
## 80	M	U	MU2	date4	22.00	13.00	0.00	1.00	0.00	0.00
## 81	M	U	MU3	date4	1.00	0.00	0.00	0.00	0.00	0.00

```
d[, c(1:4, which(spp) + 4)]
```

##	tree	landsc	site	date	haraxy	adabip	adadec	harqua	myroct	aphobl
## 1	P	R	PR1	date1	0.00	3.75	0.00	45.00	0.00	13.75
## 2	P	R	PR2	date1	0.00	13.75	1.25	30.00	3.75	2.50
## 3	P	R	PR3	date1	0.00	2.50	3.75	27.50	0.00	10.00

## 4	P	S	PS1 date1	0.00	1.25	7.50	26.25	53.75	18.75
## 5	P	S	PS2 date1	0.00	8.75	2.50	132.50	51.25	23.75
## 6	P	S	PS3 date1	0.00	5.00	8.75	43.75	1.25	13.75
## 7	P	U	PU1 date1	20.00	83.75	16.25	47.50	8.75	2.50
## 8	P	U	PU2 date1	27.50	NA	65.00	27.50	15.00	0.00
## 9	P	U	PU3 date1	2.50	NA	102.50	80.00	12.50	0.00
## 10	L	R	LR1 date2	0.00	1.00	1.00	0.00	0.00	0.00
## 11	L	R	LR2 date2	0.00	9.00	5.00	0.00	0.00	0.00
## 12	L	R	LR3 date2	0.00	15.00	8.00	0.00	0.00	0.00
## 13	L	S	LS1 date2	1.00	21.00	4.00	0.00	0.00	0.00
## 14	L	S	LS2 date2	0.00	2.00	4.00	0.00	0.00	0.00
## 15	L	S	LS3 date2	0.00	3.00	2.00	0.00	0.00	0.00
## 16	L	U	LU1 date2	0.00	16.00	6.00	0.00	0.00	0.00
## 17	L	U	LU2 date2	2.00	55.00	0.00	0.00	0.00	0.00
## 18	L	U	LU3 date2	5.00	12.00	15.00	0.00	0.00	0.00
## 19	M	R	MR1 date2	0.00	3.00	1.00	0.00	0.00	0.00
## 20	M	R	MR2 date2	0.00	3.00	1.00	0.00	0.00	0.00
## 21	M	R	MR3 date2	1.00	15.00	2.00	0.00	0.00	0.00
## 22	M	S	MS1 date2	0.00	4.00	1.00	0.00	0.00	0.00
## 23	M	S	MS2 date2	0.00	1.00	0.00	0.00	0.00	1.00
## 24	M	S	MS3 date2	0.00	3.00	1.00	0.00	0.00	0.00
## 25	M	U	MU1 date2	6.00	24.00	2.00	0.00	0.00	0.00
## 26	M	U	MU2 date2	2.00	17.00	2.00	0.00	0.00	0.00
## 27	M	U	MU3 date2	2.00	4.00	0.00	0.00	0.00	0.00
## 28	P	R	PR1 date2	0.00	3.75	0.00	21.25	1.25	0.00
## 29	P	R	PR2 date2	0.00	0.00	0.00	16.25	6.25	0.00
## 30	P	R	PR3 date2	0.00	2.50	2.50	17.50	0.00	3.75
## 31	P	S	PS1 date2	0.00	0.00	0.00	16.25	12.50	3.75
## 32	P	S	PS2 date2	0.00	3.75	0.00	45.00	67.50	7.50
## 33	P	S	PS3 date2	0.00	3.75	1.25	25.00	8.75	11.25
## 34	P	U	PU1 date2	5.00	21.25	1.25	11.25	6.25	0.00
## 35	P	U	PU2 date2	7.50	40.00	7.50	12.50	2.50	12.50
## 36	P	U	PU3 date2	0.00	12.50	7.50	5.00	2.50	0.00
## 37	L	R	LR1 date3	0.00	0.00	1.00	0.00	0.00	0.00
## 38	L	R	LR2 date3	0.00	4.00	6.00	0.00	0.00	0.00
## 39	L	R	LR3 date3	0.00	4.00	2.00	0.00	0.00	0.00
## 40	L	S	LS1 date3	1.00	13.00	7.00	0.00	0.00	0.00
## 41	L	S	LS2 date3	2.00	1.00	4.00	0.00	0.00	0.00
## 42	L	S	LS3 date3	0.00	1.00	0.00	0.00	0.00	0.00
## 43	L	U	LU1 date3	3.00	7.00	3.00	0.00	0.00	0.00
## 44	L	U	LU2 date3	9.00	62.00	2.00	0.00	0.00	0.00
## 45	L	U	LU3 date3	6.00	14.00	3.00	0.00	0.00	0.00
## 46	M	R	MR1 date3	0.00	0.00	0.00	0.00	0.00	0.00
## 47	M	R	MR2 date3	0.00	2.00	0.00	0.00	0.00	0.00
## 48	M	R	MR3 date3	0.00	3.00	1.00	0.00	0.00	0.00
## 49	M	S	MS1 date3	2.00	9.00	1.00	0.00	0.00	0.00
## 50	M	S	MS2 date3	0.00	2.00	0.00	0.00	0.00	0.00
## 51	M	S	MS3 date3	0.00	0.00	2.00	0.00	0.00	0.00
## 52	M	U	MU1 date3	2.00	9.00	1.00	0.00	0.00	0.00
## 53	M	U	MU2 date3	30.00	18.00	2.00	0.00	0.00	0.00
## 54	M	U	MU3 date3	0.00	3.00	0.00	0.00	0.00	0.00
## 55	P	R	PR1 date3	0.00	0.00	0.00	17.50	2.50	7.50
## 56	P	R	PR2 date3	0.00	0.00	0.00	20.00	11.25	2.50
## 57	P	R	PR3 date3	0.00	0.00	6.25	25.00	0.00	15.00
## 58	P	S	PS1 date3	NA	NA	NA	NA	NA	NA
## 59	P	S	PS2 date3	0.00	0.00	0.00	18.75	55.00	36.25
## 60	P	S	PS3 date3	0.00	1.25	0.00	58.75	26.25	16.25
## 61	P	U	PU1 date3	1.25	3.75	0.00	3.75	6.25	0.00
## 62	P	U	PU2 date3	0.00	10.00	7.50	0.00	7.50	0.00
## 63	P	U	PU3 date3	0.00	7.50	5.00	5.00	12.50	0.00
## 64	L	R	LR1 date4	0.00	0.00	0.00	0.00	0.00	0.00
## 65	L	R	LR2 date4	0.00	0.00	2.00	0.00	0.00	0.00
## 66	L	R	LR3 date4	0.00	2.00	0.00	0.00	0.00	0.00

## 67	L	S	LS1	date4	2.00	15.00	2.00	0.00	0.00	0.00
## 68	L	S	LS2	date4	2.00	4.00	2.00	0.00	0.00	0.00
## 69	L	S	LS3	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 70	L	U	LU1	date4	2.00	8.00	2.00	0.00	0.00	0.00
## 71	L	U	LU2	date4	6.00	48.00	1.00	0.00	0.00	0.00
## 72	L	U	LU3	date4	4.00	7.00	6.00	0.00	0.00	0.00
## 73	M	R	MR1	date4	0.00	0.00	0.00	0.00	0.00	0.00
## 74	M	R	MR2	date4	0.00	2.00	0.00	0.00	0.00	0.00
## 75	M	R	MR3	date4	1.00	8.00	2.00	0.00	0.00	0.00
## 76	M	S	MS1	date4	2.00	8.00	0.00	0.00	0.00	0.00
## 77	M	S	MS2	date4	2.00	0.00	0.00	0.00	0.00	0.00
## 78	M	S	MS3	date4	0.00	7.00	0.00	0.00	0.00	0.00
## 79	M	U	MU1	date4	10.00	27.00	2.00	0.00	0.00	0.00
## 80	M	U	MU2	date4	22.00	13.00	0.00	1.00	0.00	0.00
## 81	M	U	MU3	date4	1.00	0.00	0.00	0.00	0.00	0.00

- 18) Triez les lignes par tree, landsc et date (dans cet ordre). Vous devrez utiliser la fonction order.

```
d[order(d$tree, d$landsc, d$date), ]
```

##	tree	landsc	site	date	haraxy	adabip	adadec	calqua	caldec	exoqua	halsed
## 10	L	R	LR1	date2	0.00	1.00	1.00	1.0	0	0.00	0.00
## 11	L	R	LR2	date2	0.00	9.00	5.00	0.0	0	0.00	0.00
## 12	L	R	LR3	date2	0.00	15.00	8.00	2.0	0	0.00	0.00
## 37	L	R	LR1	date3	0.00	0.00	1.00	0.0	0	0.00	1.00
## 38	L	R	LR2	date3	0.00	4.00	6.00	1.0	6	0.00	0.00
## 39	L	R	LR3	date3	0.00	4.00	2.00	1.0	3	7.00	0.00
## 64	L	R	LR1	date4	0.00	0.00	0.00	2.0	1	0.00	2.00
## 65	L	R	LR2	date4	0.00	0.00	2.00	3.0	0	0.00	0.00
## 66	L	R	LR3	date4	0.00	2.00	0.00	0.0	2	6.00	1.00
## 13	L	S	LS1	date2	1.00	21.00	4.00	2.0	2	0.00	0.00
## 14	L	S	LS2	date2	0.00	2.00	4.00	3.0	6	1.00	0.00
## 15	L	S	LS3	date2	0.00	3.00	2.00	2.0	2	0.00	0.00
## 40	L	S	LS1	date3	1.00	13.00	7.00	3.0	8	1.00	0.00
## 41	L	S	LS2	date3	2.00	1.00	4.00	0.0	14	0.00	1.00
## 42	L	S	LS3	date3	0.00	1.00	0.00	1.0	2	0.00	0.00
## 67	L	S	LS1	date4	2.00	15.00	2.00	1.0	5	0.00	0.00
## 68	L	S	LS2	date4	2.00	4.00	2.00	2.0	4	0.00	0.00
## 69	L	S	LS3	date4	0.00	0.00	0.00	1.0	2	0.00	0.00
## 16	L	U	LU1	date2	0.00	16.00	6.00	0.0	2	0.00	0.00
## 17	L	U	LU2	date2	2.00	55.00	0.00	3.0	0	0.00	0.00
## 18	L	U	LU3	date2	5.00	12.00	15.00	2.0	2	0.00	0.00
## 43	L	U	LU1	date3	3.00	7.00	3.00	0.0	5	1.00	1.00
## 44	L	U	LU2	date3	9.00	62.00	2.00	2.0	0	0.00	0.00
## 45	L	U	LU3	date3	6.00	14.00	3.00	2.0	2	0.00	1.00
## 70	L	U	LU1	date4	2.00	8.00	2.00	0.0	2	0.00	0.00
## 71	L	U	LU2	date4	6.00	48.00	1.00	1.0	0	15.00	0.00
## 72	L	U	LU3	date4	4.00	7.00	6.00	2.0	0	6.00	0.00
## 19	M	R	MR1	date2	0.00	3.00	1.00	3.0	0	0.00	0.00
## 20	M	R	MR2	date2	0.00	3.00	1.00	6.0	1	2.00	0.00
## 21	M	R	MR3	date2	1.00	15.00	2.00	13.0	0	1.00	0.00
## 46	M	R	MR1	date3	0.00	0.00	0.00	0.0	0	0.00	0.00
## 47	M	R	MR2	date3	0.00	2.00	0.00	0.0	0	1.00	0.00
## 48	M	R	MR3	date3	0.00	3.00	1.00	0.0	2	0.00	0.00
## 73	M	R	MR1	date4	0.00	0.00	0.00	0.0	0	0.00	0.00
## 74	M	R	MR2	date4	0.00	2.00	0.00	0.0	0	1.00	1.00
## 75	M	R	MR3	date4	1.00	8.00	2.00	1.0	0	0.00	0.00
## 22	M	S	MS1	date2	0.00	4.00	1.00	1.0	0	0.00	1.00
## 23	M	S	MS2	date2	0.00	1.00	0.00	2.0	2	0.00	2.00
## 24	M	S	MS3	date2	0.00	3.00	1.00	3.0	0	0.00	2.00
## 49	M	S	MS1	date3	2.00	9.00	1.00	0.0	2	1.00	0.00

## 50	M	S	MS2 date3	0.00	2.00	0.00	1.0	6	1.00	2.00
## 51	M	S	MS3 date3	0.00	0.00	2.00	0.0	2	1.00	1.00
## 76	M	S	MS1 date4	2.00	8.00	0.00	0.0	0	0.00	0.00
## 77	M	S	MS2 date4	2.00	0.00	0.00	0.0	2	0.00	0.00
## 78	M	S	MS3 date4	0.00	7.00	0.00	0.0	2	2.00	0.00
## 25	M	U	MU1 date2	6.00	24.00	2.00	1.0	0	2.00	0.00
## 26	M	U	MU2 date2	2.00	17.00	2.00	2.0	0	1.00	0.00
## 27	M	U	MU3 date2	2.00	4.00	0.00	2.0	0	0.00	1.00
## 52	M	U	MU1 date3	2.00	9.00	1.00	0.0	0	0.00	0.00
## 53	M	U	MU2 date3	30.00	18.00	2.00	0.0	1	2.00	0.00
## 54	M	U	MU3 date3	0.00	3.00	0.00	0.0	0	0.00	0.00
## 79	M	U	MU1 date4	10.00	27.00	2.00	1.0	2	6.00	0.00
## 80	M	U	MU2 date4	22.00	13.00	0.00	2.0	0	8.00	0.00
## 81	M	U	MU3 date4	1.00	0.00	0.00	0.0	0	0.00	0.00
## 1	P	R	PR1 date1	0.00	3.75	0.00	0.0	0	0.00	0.00
## 2	P	R	PR2 date1	0.00	13.75	1.25	2.5	0	0.00	2.50
## 3	P	R	PR3 date1	0.00	2.50	3.75	0.0	0	1.25	0.00
## 28	P	R	PR1 date2	0.00	3.75	0.00	0.0	0	1.25	0.00
## 29	P	R	PR2 date2	0.00	0.00	0.00	0.0	0	0.00	0.00
## 30	P	R	PR3 date2	0.00	2.50	2.50	0.0	0	0.00	0.00
## 55	P	R	PR1 date3	0.00	0.00	0.00	0.0	0	5.00	0.00
## 56	P	R	PR2 date3	0.00	0.00	0.00	0.0	0	0.00	0.00
## 57	P	R	PR3 date3	0.00	0.00	6.25	0.0	0	0.00	0.00
## 4	P	S	PS1 date1	0.00	1.25	7.50	0.0	0	0.00	1.25
## 5	P	S	PS2 date1	0.00	8.75	2.50	0.0	0	5.00	1.25
## 6	P	S	PS3 date1	0.00	5.00	8.75	0.0	0	0.00	0.00
## 31	P	S	PS1 date2	0.00	0.00	0.00	0.0	0	0.00	0.00
## 32	P	S	PS2 date2	0.00	3.75	0.00	0.0	0	0.00	0.00
## 33	P	S	PS3 date2	0.00	3.75	1.25	0.0	0	3.75	0.00
## 58	P	S	PS1 date3	NA	NA	NA	NA	NA	NA	NA
## 59	P	S	PS2 date3	0.00	0.00	0.00	0.0	0	0.00	0.00
## 60	P	S	PS3 date3	0.00	1.25	0.00	0.0	0	0.00	0.00
## 7	P	U	PU1 date1	20.00	83.75	16.25	0.0	0	0.00	0.00
## 8	P	U	PU2 date1	27.50	NA	65.00	0.0	0	2.50	0.00
## 9	P	U	PU3 date1	2.50	NA	102.50	0.0	0	0.00	0.00
## 34	P	U	PU1 date2	5.00	21.25	1.25	0.0	0	0.00	0.00
## 35	P	U	PU2 date2	7.50	40.00	7.50	0.0	0	0.00	0.00
## 36	P	U	PU3 date2	0.00	12.50	7.50	0.0	0	0.00	0.00
## 61	P	U	PU1 date3	1.25	3.75	0.00	0.0	0	0.00	0.00
## 62	P	U	PU2 date3	0.00	10.00	7.50	0.0	0	0.00	0.00
## 63	P	U	PU3 date3	0.00	7.50	5.00	0.0	0	0.00	0.00
##	oencon harqua myroct aphobl anaocce									
## 10	0.00	0.00	0.00	0.00	0.00					
## 11	0.00	0.00	0.00	0.00	0.00					
## 12	0.00	0.00	0.00	0.00	0.00					
## 37	0.00	0.00	0.00	0.00	0.00					
## 38	0.00	0.00	0.00	0.00	0.00					
## 39	0.00	0.00	0.00	0.00	0.00					
## 64	0.00	0.00	0.00	0.00	0.00					
## 65	0.00	0.00	0.00	0.00	0.00					
## 66	0.00	0.00	0.00	0.00	0.00					
## 13	0.00	0.00	0.00	0.00	0.00					
## 14	0.00	0.00	0.00	0.00	0.00					
## 15	0.00	0.00	0.00	0.00	0.00					
## 40	4.00	0.00	0.00	0.00	0.00					
## 41	0.00	0.00	0.00	0.00	0.00					
## 42	0.00	0.00	0.00	0.00	0.00					
## 67	2.00	0.00	0.00	0.00	0.00					
## 68	0.00	0.00	0.00	0.00	0.00					
## 69	0.00	0.00	0.00	0.00	0.00					
## 16	4.00	0.00	0.00	0.00	0.00					
## 17	3.00	0.00	0.00	0.00	0.00					
## 18	1.00	0.00	0.00	0.00	0.00					


```

## 43 2.00 0.00 0.00 0.00 0.00
## 44 5.00 0.00 0.00 0.00 0.00
## 45 1.00 0.00 0.00 0.00 0.00
## 70 0.00 0.00 0.00 0.00 0.00
## 71 0.00 0.00 0.00 0.00 0.00
## 72 0.00 0.00 0.00 0.00 0.00
## 19 0.00 0.00 0.00 0.00 0.00
## 20 0.00 0.00 0.00 0.00 0.00
## 21 0.00 0.00 0.00 0.00 0.00
## 46 0.00 0.00 0.00 0.00 0.00
## 47 0.00 0.00 0.00 0.00 0.00
## 48 0.00 0.00 0.00 0.00 0.00
## 73 0.00 0.00 0.00 0.00 0.00
## 74 0.00 0.00 0.00 0.00 0.00
## 75 0.00 0.00 0.00 0.00 0.00
## 22 1.00 0.00 0.00 0.00 0.00
## 23 0.00 0.00 0.00 1.00 0.00
## 24 0.00 0.00 0.00 0.00 0.00
## 49 0.00 0.00 0.00 0.00 0.00
## 50 0.00 0.00 0.00 0.00 0.00
## 51 0.00 0.00 0.00 0.00 0.00
## 76 2.00 0.00 0.00 0.00 0.00
## 77 0.00 0.00 0.00 0.00 0.00
## 78 0.00 0.00 0.00 0.00 0.00
## 25 11.00 0.00 0.00 0.00 0.00
## 26 7.00 0.00 0.00 0.00 0.00
## 27 2.00 0.00 0.00 0.00 0.00
## 52 5.00 0.00 0.00 0.00 0.00
## 53 3.00 0.00 0.00 0.00 0.00
## 54 0.00 0.00 0.00 0.00 0.00
## 79 11.00 0.00 0.00 0.00 0.00
## 80 4.00 1.00 0.00 0.00 1.00
## 81 3.00 0.00 0.00 0.00 0.00
## 1 0.00 45.00 0.00 13.75 3.75
## 2 0.00 30.00 3.75 2.50 0.00
## 3 0.00 27.50 0.00 10.00 0.00
## 28 0.00 21.25 1.25 0.00 0.00
## 29 0.00 16.25 6.25 0.00 0.00
## 30 0.00 17.50 0.00 3.75 0.00
## 55 0.00 17.50 2.50 7.50 2.50
## 56 0.00 20.00 11.25 2.50 1.25
## 57 0.00 25.00 0.00 15.00 2.50
## 4 0.00 26.25 53.75 18.75 3.75
## 5 0.00 132.50 51.25 23.75 0.00
## 6 0.00 43.75 1.25 13.75 0.00
## 31 0.00 16.25 12.50 3.75 0.00
## 32 0.00 45.00 67.50 7.50 0.00
## 33 0.00 25.00 8.75 11.25 3.75
## 58 NA NA NA NA NA
## 59 0.00 18.75 55.00 36.25 0.00
## 60 0.00 58.75 26.25 16.25 6.25
## 7 11.25 47.50 8.75 2.50 0.00
## 8 0.00 27.50 15.00 0.00 0.00
## 9 0.00 80.00 12.50 0.00 0.00
## 34 5.00 11.25 6.25 0.00 0.00
## 35 0.00 12.50 2.50 12.50 0.00
## 36 0.00 5.00 2.50 0.00 0.00
## 61 1.25 3.75 6.25 0.00 0.00
## 62 0.00 0.00 7.50 0.00 0.00
## 63 0.00 5.00 12.50 0.00 0.00

```

- 19) Comment faire pour que la colonne landsc se classe dans cet ordre : U, S, R ? (utilisez la fonction factor et le paramètre labels= pour changer l'ordre des niveaux du facteur landsc. Ensuite réutilisez la même ligne de

code que ci-dessus.)

```
d$landsc <- factor(d$landsc, levels = c("U", "S", "R"), ordered = TRUE)
d[order(d$tree, d$landsc, d$date), ]
```

##	tree	landsc	site	date	haraxy	adabip	adadec	calqua	caldec	exoqua	halsed
## 16	L	U	LU1	date2	0.00	16.00	6.00	0.0	2	0.00	0.00
## 17	L	U	LU2	date2	2.00	55.00	0.00	3.0	0	0.00	0.00
## 18	L	U	LU3	date2	5.00	12.00	15.00	2.0	2	0.00	0.00
## 43	L	U	LU1	date3	3.00	7.00	3.00	0.0	5	1.00	1.00
## 44	L	U	LU2	date3	9.00	62.00	2.00	2.0	0	0.00	0.00
## 45	L	U	LU3	date3	6.00	14.00	3.00	2.0	2	0.00	1.00
## 70	L	U	LU1	date4	2.00	8.00	2.00	0.0	2	0.00	0.00
## 71	L	U	LU2	date4	6.00	48.00	1.00	1.0	0	15.00	0.00
## 72	L	U	LU3	date4	4.00	7.00	6.00	2.0	0	6.00	0.00
## 13	L	S	LS1	date2	1.00	21.00	4.00	2.0	2	0.00	0.00
## 14	L	S	LS2	date2	0.00	2.00	4.00	3.0	6	1.00	0.00
## 15	L	S	LS3	date2	0.00	3.00	2.00	2.0	2	0.00	0.00
## 40	L	S	LS1	date3	1.00	13.00	7.00	3.0	8	1.00	0.00
## 41	L	S	LS2	date3	2.00	1.00	4.00	0.0	14	0.00	1.00
## 42	L	S	LS3	date3	0.00	1.00	0.00	1.0	2	0.00	0.00
## 67	L	S	LS1	date4	2.00	15.00	2.00	1.0	5	0.00	0.00
## 68	L	S	LS2	date4	2.00	4.00	2.00	2.0	4	0.00	0.00
## 69	L	S	LS3	date4	0.00	0.00	0.00	1.0	2	0.00	0.00
## 10	L	R	LR1	date2	0.00	1.00	1.00	1.0	0	0.00	0.00
## 11	L	R	LR2	date2	0.00	9.00	5.00	0.0	0	0.00	0.00
## 12	L	R	LR3	date2	0.00	15.00	8.00	2.0	0	0.00	0.00
## 37	L	R	LR1	date3	0.00	0.00	1.00	0.0	0	0.00	1.00
## 38	L	R	LR2	date3	0.00	4.00	6.00	1.0	6	0.00	0.00
## 39	L	R	LR3	date3	0.00	4.00	2.00	1.0	3	7.00	0.00
## 64	L	R	LR1	date4	0.00	0.00	0.00	2.0	1	0.00	2.00
## 65	L	R	LR2	date4	0.00	0.00	2.00	3.0	0	0.00	0.00
## 66	L	R	LR3	date4	0.00	2.00	0.00	0.0	2	6.00	1.00
## 25	M	U	MU1	date2	6.00	24.00	2.00	1.0	0	2.00	0.00
## 26	M	U	MU2	date2	2.00	17.00	2.00	2.0	0	1.00	0.00
## 27	M	U	MU3	date2	2.00	4.00	0.00	2.0	0	0.00	1.00
## 52	M	U	MU1	date3	2.00	9.00	1.00	0.0	0	0.00	0.00
## 53	M	U	MU2	date3	30.00	18.00	2.00	0.0	1	2.00	0.00
## 54	M	U	MU3	date3	0.00	3.00	0.00	0.0	0	0.00	0.00
## 79	M	U	MU1	date4	10.00	27.00	2.00	1.0	2	6.00	0.00
## 80	M	U	MU2	date4	22.00	13.00	0.00	2.0	0	8.00	0.00
## 81	M	U	MU3	date4	1.00	0.00	0.00	0.0	0	0.00	0.00
## 22	M	S	MS1	date2	0.00	4.00	1.00	1.0	0	0.00	1.00
## 23	M	S	MS2	date2	0.00	1.00	0.00	2.0	2	0.00	2.00
## 24	M	S	MS3	date2	0.00	3.00	1.00	3.0	0	0.00	2.00
## 49	M	S	MS1	date3	2.00	9.00	1.00	0.0	2	1.00	0.00
## 50	M	S	MS2	date3	0.00	2.00	0.00	1.0	6	1.00	2.00
## 51	M	S	MS3	date3	0.00	0.00	2.00	0.0	2	1.00	1.00
## 76	M	S	MS1	date4	2.00	8.00	0.00	0.0	0	0.00	0.00
## 77	M	S	MS2	date4	2.00	0.00	0.00	0.0	2	0.00	0.00
## 78	M	S	MS3	date4	0.00	7.00	0.00	0.0	2	2.00	0.00
## 19	M	R	MR1	date2	0.00	3.00	1.00	3.0	0	0.00	0.00
## 20	M	R	MR2	date2	0.00	3.00	1.00	6.0	1	2.00	0.00
## 21	M	R	MR3	date2	1.00	15.00	2.00	13.0	0	1.00	0.00
## 46	M	R	MR1	date3	0.00	0.00	0.00	0.0	0	0.00	0.00
## 47	M	R	MR2	date3	0.00	2.00	0.00	0.0	0	1.00	0.00
## 48	M	R	MR3	date3	0.00	3.00	1.00	0.0	2	0.00	0.00
## 73	M	R	MR1	date4	0.00	0.00	0.00	0.0	0	0.00	0.00
## 74	M	R	MR2	date4	0.00	2.00	0.00	0.0	0	1.00	1.00
## 75	M	R	MR3	date4	1.00	8.00	2.00	1.0	0	0.00	0.00
## 7	P	U	PU1	date1	20.00	83.75	16.25	0.0	0	0.00	0.00
## 8	P	U	PU2	date1	27.50	NA	65.00	0.0	0	2.50	0.00

## 9	P	U	PU3 date1	2.50	NA	102.50	0.0	0	0.00	0.00
## 34	P	U	PU1 date2	5.00	21.25	1.25	0.0	0	0.00	0.00
## 35	P	U	PU2 date2	7.50	40.00	7.50	0.0	0	0.00	0.00
## 36	P	U	PU3 date2	0.00	12.50	7.50	0.0	0	0.00	0.00
## 61	P	U	PU1 date3	1.25	3.75	0.00	0.0	0	0.00	0.00
## 62	P	U	PU2 date3	0.00	10.00	7.50	0.0	0	0.00	0.00
## 63	P	U	PU3 date3	0.00	7.50	5.00	0.0	0	0.00	0.00
## 4	P	S	PS1 date1	0.00	1.25	7.50	0.0	0	0.00	1.25
## 5	P	S	PS2 date1	0.00	8.75	2.50	0.0	0	5.00	1.25
## 6	P	S	PS3 date1	0.00	5.00	8.75	0.0	0	0.00	0.00
## 31	P	S	PS1 date2	0.00	0.00	0.00	0.0	0	0.00	0.00
## 32	P	S	PS2 date2	0.00	3.75	0.00	0.0	0	0.00	0.00
## 33	P	S	PS3 date2	0.00	3.75	1.25	0.0	0	3.75	0.00
## 58	P	S	PS1 date3	NA	NA	NA	NA	NA	NA	NA
## 59	P	S	PS2 date3	0.00	0.00	0.00	0.0	0	0.00	0.00
## 60	P	S	PS3 date3	0.00	1.25	0.00	0.0	0	0.00	0.00
## 1	P	R	PR1 date1	0.00	3.75	0.00	0.0	0	0.00	0.00
## 2	P	R	PR2 date1	0.00	13.75	1.25	2.5	0	0.00	2.50
## 3	P	R	PR3 date1	0.00	2.50	3.75	0.0	0	1.25	0.00
## 28	P	R	PR1 date2	0.00	3.75	0.00	0.0	0	1.25	0.00
## 29	P	R	PR2 date2	0.00	0.00	0.00	0.0	0	0.00	0.00
## 30	P	R	PR3 date2	0.00	2.50	2.50	0.0	0	0.00	0.00
## 55	P	R	PR1 date3	0.00	0.00	0.00	0.0	0	5.00	0.00
## 56	P	R	PR2 date3	0.00	0.00	0.00	0.0	0	0.00	0.00
## 57	P	R	PR3 date3	0.00	0.00	6.25	0.0	0	0.00	0.00
##	oencon harqua myroct aphobl anaoce									
## 16	4.00	0.00	0.00	0.00	0.00					
## 17	3.00	0.00	0.00	0.00	0.00					
## 18	1.00	0.00	0.00	0.00	0.00					
## 43	2.00	0.00	0.00	0.00	0.00					
## 44	5.00	0.00	0.00	0.00	0.00					
## 45	1.00	0.00	0.00	0.00	0.00					
## 70	0.00	0.00	0.00	0.00	0.00					
## 71	0.00	0.00	0.00	0.00	0.00					
## 72	0.00	0.00	0.00	0.00	0.00					
## 13	0.00	0.00	0.00	0.00	0.00					
## 14	0.00	0.00	0.00	0.00	0.00					
## 15	0.00	0.00	0.00	0.00	0.00					
## 40	4.00	0.00	0.00	0.00	0.00					
## 41	0.00	0.00	0.00	0.00	0.00					
## 42	0.00	0.00	0.00	0.00	0.00					
## 67	2.00	0.00	0.00	0.00	0.00					
## 68	0.00	0.00	0.00	0.00	0.00					
## 69	0.00	0.00	0.00	0.00	0.00					
## 10	0.00	0.00	0.00	0.00	0.00					
## 11	0.00	0.00	0.00	0.00	0.00					
## 12	0.00	0.00	0.00	0.00	0.00					
## 37	0.00	0.00	0.00	0.00	0.00					
## 38	0.00	0.00	0.00	0.00	0.00					
## 39	0.00	0.00	0.00	0.00	0.00					
## 64	0.00	0.00	0.00	0.00	0.00					
## 65	0.00	0.00	0.00	0.00	0.00					
## 66	0.00	0.00	0.00	0.00	0.00					
## 25	11.00	0.00	0.00	0.00	0.00					
## 26	7.00	0.00	0.00	0.00	0.00					
## 27	2.00	0.00	0.00	0.00	0.00					
## 52	5.00	0.00	0.00	0.00	0.00					
## 53	3.00	0.00	0.00	0.00	0.00					
## 54	0.00	0.00	0.00	0.00	0.00					
## 79	11.00	0.00	0.00	0.00	0.00					
## 80	4.00	1.00	0.00	0.00	1.00					
## 81	3.00	0.00	0.00	0.00	0.00					
## 22	1.00	0.00	0.00	0.00	0.00					

```
## 23  0.00  0.00  0.00  1.00  0.00
## 24  0.00  0.00  0.00  0.00  0.00
## 49  0.00  0.00  0.00  0.00  0.00
## 50  0.00  0.00  0.00  0.00  0.00
## 51  0.00  0.00  0.00  0.00  0.00
## 76  2.00  0.00  0.00  0.00  0.00
## 77  0.00  0.00  0.00  0.00  0.00
## 78  0.00  0.00  0.00  0.00  0.00
## 19  0.00  0.00  0.00  0.00  0.00
## 20  0.00  0.00  0.00  0.00  0.00
## 21  0.00  0.00  0.00  0.00  0.00
## 46  0.00  0.00  0.00  0.00  0.00
## 47  0.00  0.00  0.00  0.00  0.00
## 48  0.00  0.00  0.00  0.00  0.00
## 73  0.00  0.00  0.00  0.00  0.00
## 74  0.00  0.00  0.00  0.00  0.00
## 75  0.00  0.00  0.00  0.00  0.00
## 7   11.25 47.50  8.75  2.50  0.00
## 8    0.00 27.50 15.00  0.00  0.00
## 9    0.00 80.00 12.50  0.00  0.00
## 34   5.00 11.25  6.25  0.00  0.00
## 35   0.00 12.50  2.50 12.50  0.00
## 36   0.00  5.00  2.50  0.00  0.00
## 61   1.25  3.75  6.25  0.00  0.00
## 62   0.00  0.00  7.50  0.00  0.00
## 63   0.00  5.00 12.50  0.00  0.00
## 4    0.00 26.25 53.75 18.75  3.75
## 5    0.00 132.50 51.25 23.75  0.00
## 6    0.00 43.75  1.25 13.75  0.00
## 31   0.00 16.25 12.50  3.75  0.00
## 32   0.00 45.00 67.50  7.50  0.00
## 33   0.00 25.00  8.75 11.25  3.75
## 58    NA    NA    NA    NA    NA
## 59   0.00 18.75 55.00 36.25  0.00
## 60   0.00 58.75 26.25 16.25  6.25
## 1    0.00 45.00  0.00 13.75  3.75
## 2    0.00 30.00  3.75  2.50  0.00
## 3    0.00 27.50  0.00 10.00  0.00
## 28   0.00 21.25  1.25  0.00  0.00
## 29   0.00 16.25  6.25  0.00  0.00
## 30   0.00 17.50  0.00  3.75  0.00
## 55   0.00 17.50  2.50  7.50  2.50
## 56   0.00 20.00 11.25  2.50  1.25
## 57   0.00 25.00  0.00 15.00  2.50
```

- 20) Réordonnez colonnes du jeu de données de manière à ce que les espèces soient présentées de la plus abondante à la moins abondante (ie harqua, adabip, myroct, adadec, etc...)

```
or <- order(colSums(d[, -c(1:4)]), na.rm = TRUE), decreasing = TRUE)
or
```

```
## [1] 9 2 10 3 11 1 5 8 6 4 12 7
```

```
d[, c(1:4, or+4)]
```

```
##      tree landsc site  date harqua adabip myroct adadec aphobl haraxy caldec
## 1      P      R  PR1 date1  45.00   3.75  0.00  0.00 13.75  0.00    0
## 2      P      R  PR2 date1  30.00  13.75  3.75  1.25  2.50  0.00    0
## 3      P      R  PR3 date1  27.50   2.50  0.00  3.75 10.00  0.00    0
## 4      P      S  PS1 date1  26.25   1.25 53.75  7.50 18.75  0.00    0
```

## 5	P	S	PS2 date1	132.50	8.75	51.25	2.50	23.75	0.00	0
## 6	P	S	PS3 date1	43.75	5.00	1.25	8.75	13.75	0.00	0
## 7	P	U	PU1 date1	47.50	83.75	8.75	16.25	2.50	20.00	0
## 8	P	U	PU2 date1	27.50	NA	15.00	65.00	0.00	27.50	0
## 9	P	U	PU3 date1	80.00	NA	12.50	102.50	0.00	2.50	0
## 10	L	R	LR1 date2	0.00	1.00	0.00	1.00	0.00	0.00	0
## 11	L	R	LR2 date2	0.00	9.00	0.00	5.00	0.00	0.00	0
## 12	L	R	LR3 date2	0.00	15.00	0.00	8.00	0.00	0.00	0
## 13	L	S	LS1 date2	0.00	21.00	0.00	4.00	0.00	1.00	2
## 14	L	S	LS2 date2	0.00	2.00	0.00	4.00	0.00	0.00	6
## 15	L	S	LS3 date2	0.00	3.00	0.00	2.00	0.00	0.00	2
## 16	L	U	LU1 date2	0.00	16.00	0.00	6.00	0.00	0.00	2
## 17	L	U	LU2 date2	0.00	55.00	0.00	0.00	0.00	2.00	0
## 18	L	U	LU3 date2	0.00	12.00	0.00	15.00	0.00	5.00	2
## 19	M	R	MR1 date2	0.00	3.00	0.00	1.00	0.00	0.00	0
## 20	M	R	MR2 date2	0.00	3.00	0.00	1.00	0.00	0.00	1
## 21	M	R	MR3 date2	0.00	15.00	0.00	2.00	0.00	1.00	0
## 22	M	S	MS1 date2	0.00	4.00	0.00	1.00	0.00	0.00	0
## 23	M	S	MS2 date2	0.00	1.00	0.00	0.00	1.00	0.00	2
## 24	M	S	MS3 date2	0.00	3.00	0.00	1.00	0.00	0.00	0
## 25	M	U	MU1 date2	0.00	24.00	0.00	2.00	0.00	6.00	0
## 26	M	U	MU2 date2	0.00	17.00	0.00	2.00	0.00	2.00	0
## 27	M	U	MU3 date2	0.00	4.00	0.00	0.00	0.00	2.00	0
## 28	P	R	PR1 date2	21.25	3.75	1.25	0.00	0.00	0.00	0
## 29	P	R	PR2 date2	16.25	0.00	6.25	0.00	0.00	0.00	0
## 30	P	R	PR3 date2	17.50	2.50	0.00	2.50	3.75	0.00	0
## 31	P	S	PS1 date2	16.25	0.00	12.50	0.00	3.75	0.00	0
## 32	P	S	PS2 date2	45.00	3.75	67.50	0.00	7.50	0.00	0
## 33	P	S	PS3 date2	25.00	3.75	8.75	1.25	11.25	0.00	0
## 34	P	U	PU1 date2	11.25	21.25	6.25	1.25	0.00	5.00	0
## 35	P	U	PU2 date2	12.50	40.00	2.50	7.50	12.50	7.50	0
## 36	P	U	PU3 date2	5.00	12.50	2.50	7.50	0.00	0.00	0
## 37	L	R	LR1 date3	0.00	0.00	0.00	1.00	0.00	0.00	0
## 38	L	R	LR2 date3	0.00	4.00	0.00	6.00	0.00	0.00	6
## 39	L	R	LR3 date3	0.00	4.00	0.00	2.00	0.00	0.00	3
## 40	L	S	LS1 date3	0.00	13.00	0.00	7.00	0.00	1.00	8
## 41	L	S	LS2 date3	0.00	1.00	0.00	4.00	0.00	2.00	14
## 42	L	S	LS3 date3	0.00	1.00	0.00	0.00	0.00	0.00	2
## 43	L	U	LU1 date3	0.00	7.00	0.00	3.00	0.00	3.00	5
## 44	L	U	LU2 date3	0.00	62.00	0.00	2.00	0.00	9.00	0
## 45	L	U	LU3 date3	0.00	14.00	0.00	3.00	0.00	6.00	2
## 46	M	R	MR1 date3	0.00	0.00	0.00	0.00	0.00	0.00	0
## 47	M	R	MR2 date3	0.00	2.00	0.00	0.00	0.00	0.00	0
## 48	M	R	MR3 date3	0.00	3.00	0.00	1.00	0.00	0.00	2
## 49	M	S	MS1 date3	0.00	9.00	0.00	1.00	0.00	2.00	2
## 50	M	S	MS2 date3	0.00	2.00	0.00	0.00	0.00	0.00	6
## 51	M	S	MS3 date3	0.00	0.00	0.00	2.00	0.00	0.00	2
## 52	M	U	MU1 date3	0.00	9.00	0.00	1.00	0.00	2.00	0
## 53	M	U	MU2 date3	0.00	18.00	0.00	2.00	0.00	30.00	1
## 54	M	U	MU3 date3	0.00	3.00	0.00	0.00	0.00	0.00	0
## 55	P	R	PR1 date3	17.50	0.00	2.50	0.00	7.50	0.00	0
## 56	P	R	PR2 date3	20.00	0.00	11.25	0.00	2.50	0.00	0
## 57	P	R	PR3 date3	25.00	0.00	0.00	6.25	15.00	0.00	0
## 58	P	S	PS1 date3	NA	NA	NA	NA	NA	NA	NA
## 59	P	S	PS2 date3	18.75	0.00	55.00	0.00	36.25	0.00	0
## 60	P	S	PS3 date3	58.75	1.25	26.25	0.00	16.25	0.00	0
## 61	P	U	PU1 date3	3.75	3.75	6.25	0.00	0.00	1.25	0
## 62	P	U	PU2 date3	0.00	10.00	7.50	7.50	0.00	0.00	0
## 63	P	U	PU3 date3	5.00	7.50	12.50	5.00	0.00	0.00	0
## 64	L	R	LR1 date4	0.00	0.00	0.00	0.00	0.00	0.00	1
## 65	L	R	LR2 date4	0.00	0.00	0.00	2.00	0.00	0.00	0
## 66	L	R	LR3 date4	0.00	2.00	0.00	0.00	0.00	0.00	2
## 67	L	S	LS1 date4	0.00	15.00	0.00	2.00	0.00	2.00	5

## 68	L	S	LS2 date4	0.00	4.00	0.00	2.00	0.00	2.00	4
## 69	L	S	LS3 date4	0.00	0.00	0.00	0.00	0.00	0.00	2
## 70	L	U	LU1 date4	0.00	8.00	0.00	2.00	0.00	2.00	2
## 71	L	U	LU2 date4	0.00	48.00	0.00	1.00	0.00	6.00	0
## 72	L	U	LU3 date4	0.00	7.00	0.00	6.00	0.00	4.00	0
## 73	M	R	MR1 date4	0.00	0.00	0.00	0.00	0.00	0.00	0
## 74	M	R	MR2 date4	0.00	2.00	0.00	0.00	0.00	0.00	0
## 75	M	R	MR3 date4	0.00	8.00	0.00	2.00	0.00	1.00	0
## 76	M	S	MS1 date4	0.00	8.00	0.00	0.00	0.00	2.00	0
## 77	M	S	MS2 date4	0.00	0.00	0.00	0.00	0.00	2.00	2
## 78	M	S	MS3 date4	0.00	7.00	0.00	0.00	0.00	0.00	2
## 79	M	U	MU1 date4	0.00	27.00	0.00	2.00	0.00	10.00	2
## 80	M	U	MU2 date4	1.00	13.00	0.00	0.00	0.00	22.00	0
## 81	M	U	MU3 date4	0.00	0.00	0.00	0.00	0.00	1.00	0
##	oencon exoqua calqua anaoce halsed									
## 1	0.00	0.00	0.0	3.75	0.00					
## 2	0.00	0.00	2.5	0.00	2.50					
## 3	0.00	1.25	0.0	0.00	0.00					
## 4	0.00	0.00	0.0	3.75	1.25					
## 5	0.00	5.00	0.0	0.00	1.25					
## 6	0.00	0.00	0.0	0.00	0.00					
## 7	11.25	0.00	0.0	0.00	0.00					
## 8	0.00	2.50	0.0	0.00	0.00					
## 9	0.00	0.00	0.0	0.00	0.00					
## 10	0.00	0.00	1.0	0.00	0.00					
## 11	0.00	0.00	0.0	0.00	0.00					
## 12	0.00	0.00	2.0	0.00	0.00					
## 13	0.00	0.00	2.0	0.00	0.00					
## 14	0.00	1.00	3.0	0.00	0.00					
## 15	0.00	0.00	2.0	0.00	0.00					
## 16	4.00	0.00	0.0	0.00	0.00					
## 17	3.00	0.00	3.0	0.00	0.00					
## 18	1.00	0.00	2.0	0.00	0.00					
## 19	0.00	0.00	3.0	0.00	0.00					
## 20	0.00	2.00	6.0	0.00	0.00					
## 21	0.00	1.00	13.0	0.00	0.00					
## 22	1.00	0.00	1.0	0.00	1.00					
## 23	0.00	0.00	2.0	0.00	2.00					
## 24	0.00	0.00	3.0	0.00	2.00					
## 25	11.00	2.00	1.0	0.00	0.00					
## 26	7.00	1.00	2.0	0.00	0.00					
## 27	2.00	0.00	2.0	0.00	1.00					
## 28	0.00	1.25	0.0	0.00	0.00					
## 29	0.00	0.00	0.0	0.00	0.00					
## 30	0.00	0.00	0.0	0.00	0.00					
## 31	0.00	0.00	0.0	0.00	0.00					
## 32	0.00	0.00	0.0	0.00	0.00					
## 33	0.00	3.75	0.0	3.75	0.00					
## 34	5.00	0.00	0.0	0.00	0.00					
## 35	0.00	0.00	0.0	0.00	0.00					
## 36	0.00	0.00	0.0	0.00	0.00					
## 37	0.00	0.00	0.0	0.00	1.00					
## 38	0.00	0.00	1.0	0.00	0.00					
## 39	0.00	7.00	1.0	0.00	0.00					
## 40	4.00	1.00	3.0	0.00	0.00					
## 41	0.00	0.00	0.0	0.00	1.00					
## 42	0.00	0.00	1.0	0.00	0.00					
## 43	2.00	1.00	0.0	0.00	1.00					
## 44	5.00	0.00	2.0	0.00	0.00					
## 45	1.00	0.00	2.0	0.00	1.00					
## 46	0.00	0.00	0.0	0.00	0.00					
## 47	0.00	1.00	0.0	0.00	0.00					
## 48	0.00	0.00	0.0	0.00	0.00					

## 49	0.00	1.00	0.0	0.00	0.00
## 50	0.00	1.00	1.0	0.00	2.00
## 51	0.00	1.00	0.0	0.00	1.00
## 52	5.00	0.00	0.0	0.00	0.00
## 53	3.00	2.00	0.0	0.00	0.00
## 54	0.00	0.00	0.0	0.00	0.00
## 55	0.00	5.00	0.0	2.50	0.00
## 56	0.00	0.00	0.0	1.25	0.00
## 57	0.00	0.00	0.0	2.50	0.00
## 58	NA	NA	NA	NA	NA
## 59	0.00	0.00	0.0	0.00	0.00
## 60	0.00	0.00	0.0	6.25	0.00
## 61	1.25	0.00	0.0	0.00	0.00
## 62	0.00	0.00	0.0	0.00	0.00
## 63	0.00	0.00	0.0	0.00	0.00
## 64	0.00	0.00	2.0	0.00	2.00
## 65	0.00	0.00	3.0	0.00	0.00
## 66	0.00	6.00	0.0	0.00	1.00
## 67	2.00	0.00	1.0	0.00	0.00
## 68	0.00	0.00	2.0	0.00	0.00
## 69	0.00	0.00	1.0	0.00	0.00
## 70	0.00	0.00	0.0	0.00	0.00
## 71	0.00	15.00	1.0	0.00	0.00
## 72	0.00	6.00	2.0	0.00	0.00
## 73	0.00	0.00	0.0	0.00	0.00
## 74	0.00	1.00	0.0	0.00	1.00
## 75	0.00	0.00	1.0	0.00	0.00
## 76	2.00	0.00	0.0	0.00	0.00
## 77	0.00	0.00	0.0	0.00	0.00
## 78	0.00	2.00	0.0	0.00	0.00
## 79	11.00	6.00	1.0	0.00	0.00
## 80	4.00	8.00	2.0	1.00	0.00
## 81	3.00	0.00	0.0	0.00	0.00

Exercices de manipulation de caractères

Quelques rappels :

- les fonctions `grep` et `grepl` ont un argument `ignore.case` qui permet d'ignorer la casse des caractères quand on le désire
- dans les expressions régulières, si vous voulez rechercher par exemple un point “.” il faudra chercher “\.” car le point a une signification particulière dans la syntaxe des expressions régulières.

Exercice 1

Voici un jeu de données sur lequel on va travailler (espèce, nombre d'individus, site)

```
d <- data.frame(
  sp = c("COCCINELLA SEPTEMPUNCTATA", "Coccinella septempunctata",
        "Coccinella Septempunctata Brucki", "Adalia spp", "COCCINELLIDAE",
        "coccinella hieroglyphica", "ADALIA BIPUNCTATA REVIELERI",
        "Adalia bipunctata bipunctata", "Adalia decempunctata", "Formicidae",
        "Harmonia spp.", "Adalia bipunctata", "Coccinella quinquepunctata",
        "Coccinella sp.", "Myrmica speciosides", "NOMADA BISPINOSA",
        "NOMADA sp.", "Formicinae", "Apoidea", "Dinocampus coccinellae", "COLEOPTERA",
        "Aphaenogaster spinosa", "MYRMICA SP.", "Lasius niger", "Formicinae"),
  n = c(1, 5, 10, 33, 2, 1, 4, 6, 1, 3, 1, 1, 2, 1, 9, 3, 1, 1, 3, 2, 6, 1, 2, 1, 3),
  site = c("Mazée", "Namur", "Res. Nat de Matagne", "Treignes",
           "res nat du coupu tienne", "Vaucelles", "Rés. Nat. Roche Madou", " Vierves",
           "natoye", "Olloy", "Le Mesnil", "resnat du Chamousias", "Natoye", "Mazée",
           "Réserve Naturelle de la Haie Gabaux", "Dourbes", "Doische",
           "Oignies", "Mazée", "Vaucelles", "Rés. nat. de la Montagne de la Carrière",
           "Regniessart", "Vireux", "Haybes", "res nat Al Florée")
)
summary(d)
```

```
##              sp              n              site
## Formicinae          : 2   Min.   : 1.00   Mazée      : 3
## Adalia bipunctata    : 1   1st Qu.: 1.00   Vaucelles: 2
## Adalia bipunctata bipunctata: 1   Median : 2.00   Doische   : 1
## ADALIA BIPUNCTATA REVIELERI : 1   Mean    : 4.12   Dourbes   : 1
## Adalia decempunctata    : 1   3rd Qu.: 4.00   Haybes    : 1
## Adalia spp            : 1   Max.    :33.00   Le Mesnil: 1
## (Other)               :18                      (Other) :16
```

```
d

##              sp  n
## 1      COCCINELLA SEPTEMPUNCTATA 1
## 2      Coccinella septempunctata 5
## 3      Coccinella Septempunctata Brucki 10
## 4              Adalia spp 33
## 5      COCCINELLIDAE 2
## 6      coccinella hieroglyphica 1
## 7      ADALIA BIPUNCTATA REVIELERI 4
## 8      Adalia bipunctata bipunctata 6
## 9      Adalia decempunctata 1
## 10             Formicidae 3
## 11             Harmonia spp. 1
## 12             Adalia bipunctata 1
## 13      Coccinella quinquepunctata 2
```



```
## 14          Coccinella sp. 1
## 15      Myrmica specioidea 9
## 16      NOMADA BISPINOSA 3
## 17          NOMADA sp. 1
## 18          Formicinae 1
## 19          Apoidea 3
## 20      Dinocampus coccinellae 2
## 21          COLEOPTERA 6
## 22      Aphaenogaster spinosa 1
## 23          MYRMICA SP. 2
## 24          Lasius niger 1
## 25          Formicinae 3
##
##              site
## 1              Mazée
## 2              Namur
## 3      Res. Nat de Matagne
## 4              Treignes
## 5      res nat du coupu tienne
## 6              Vaucelles
## 7      Rés. Nat. Roche Madou
## 8              Vierves
## 9              natoye
## 10             Olloy
## 11             Le Mesnil
## 12      resnat du Chamousias
## 13             Natoye
## 14             Mazée
## 15      Réserve Naturelle de la Haie Gabaux
## 16             Dourbes
## 17             Doische
## 18             Oignies
## 19             Mazée
## 20             Vaucelles
## 21 Rés. nat. de la Montagne de la Carrière
## 22             Regniessart
## 23             Vireux
## 24             Haybes
## 25      res nat Al Florée
```

Rappels sur la nomenclature des noms d'espèces et la taxonomie.

Le nom scientifique d'une espèce est toujours composé du nom de genre suivi du nom d'espèce. Par exemple : *Adalia bipunctata*, *Adalia* étant le nom de genre, *bipunctata*, le nom d'espèce. Lorsqu'on ne sait pas identifier une espèce jusqu'à l'espèce, on indique en general le nom de genre suivi de "sp" ou "spp" ("species"). Par exemple *Adalia* sp. Lorsqu'il y a trois noms à la suite (pex *Adalia bipunctata bipunctata*), le 3ème désigne une sous-espèce. Lorsqu'on a un seul nom terminant par "inae", "idae", "oidea", "era",... il s'agit de niveaux taxonomiques supérieurs au nom de genre (sous-famille, famille, ordre,...)

- 1) Dans la colonne sp, la casse des noms scientifiques n'est pas uniforme. Mettez tous les caractères en majuscule.

```
d$sp <- toupper(d$sp)
```

- 2) Sélectionnez les données du genre *Adalia* (n=5)

```
d[grepl("ADALIA", d$sp), ]
```

```
##              sp  n              site
## 4      ADALIA SPP 33      Treignes
## 7  ADALIA BIPUNCTATA REVIELERI 4 Rés. Nat. Roche Madou
```

```
## 8 ADALIA BIPUNCTATA BIPUNCTATA 6 Vierves
## 9 ADALIA DECEMPUNCTATA 1 natoye
## 12 ADALIA BIPUNCTATA 1 resnat du Chamousias
```

- 3) Sélectionnez les données correspondant au genre *Coccinella* (attention aux parasites !). (n = 6) Comment faire pour éviter de sélectionner *Dinocampus coccinellae* (un parasite de coccinelles...) sans changer la casse des caractères ?

```
d[grepl("COCCINELLA", d$sp), ] # pas bon car on inclut Dinocampus coccinellae
```

```
##           sp n           site
## 1 COCCINELLA SEPTEMPUNCTATA 1 Mazée
## 2 COCCINELLA SEPTEMPUNCTATA 5 Namur
## 3 COCCINELLA SEPTEMPUNCTATA BRUCKI 10 Res. Nat de Matagne
## 6 COCCINELLA HIEROGLYPHICA 1 Vaucelles
## 13 COCCINELLA QUINQUEPUNCTATA 2 Natoye
## 14 COCCINELLA SP. 1 Mazée
## 20 DINOCAMPUS COCCINELLAE 2 Vaucelles
```

```
# il suffit de spécifier que Coccinella doit être en début de chaîne et éventuellement
# de spécifier qu'il est suivi d'une espace
```

```
d[grepl("^COCCINELLA ", d$sp), ]
```

```
##           sp n           site
## 1 COCCINELLA SEPTEMPUNCTATA 1 Mazée
## 2 COCCINELLA SEPTEMPUNCTATA 5 Namur
## 3 COCCINELLA SEPTEMPUNCTATA BRUCKI 10 Res. Nat de Matagne
## 6 COCCINELLA HIEROGLYPHICA 1 Vaucelles
## 13 COCCINELLA QUINQUEPUNCTATA 2 Natoye
## 14 COCCINELLA SP. 1 Mazée
```

- 4) Sélectionnez les données des genres *Adalia* et *Coccinella* (n = 11)

```
d[grepl("^ADALIA |^COCCINELLA ", d$sp), ]
```

```
##           sp n           site
## 1 COCCINELLA SEPTEMPUNCTATA 1 Mazée
## 2 COCCINELLA SEPTEMPUNCTATA 5 Namur
## 3 COCCINELLA SEPTEMPUNCTATA BRUCKI 10 Res. Nat de Matagne
## 4 ADALIA SPP 33 Treignes
## 6 COCCINELLA HIEROGLYPHICA 1 Vaucelles
## 7 ADALIA BIPUNCTATA REVIELERI 4 Rés. Nat. Roche Madou
## 8 ADALIA BIPUNCTATA BIPUNCTATA 6 Vierves
## 9 ADALIA DECEMPUNCTATA 1 natoye
## 12 ADALIA BIPUNCTATA 1 resnat du Chamousias
## 13 COCCINELLA QUINQUEPUNCTATA 2 Natoye
## 14 COCCINELLA SP. 1 Mazée
```

- 5) Transformez le vecteur de noms scientifiques "sp" de manière à avoir la première lettre en majuscule et les suivantes en minuscule. Vous pouvez simplement utiliser ici la fonction `substring` (avec `paste`, `toupper` et `tolower`). Les expressions régulières ne sont nécessaires (il est aussi possible d'utiliser les expressions régulières mais c'est plus compliqué dans ce cas).

```
d$sp <- paste(toupper(substring(d$sp, 1, 1)), tolower(substring(d$sp, 2, 1000)), sep="")
d$sp
```

```
## [1] "Coccinella septempunctata"      "Coccinella septempunctata"
## [3] "Coccinella septempunctata brucki" "Adalia spp"
## [5] "Coccinellidae"                  "Coccinella hieroglyphica"
## [7] "Adalia bipunctata revielieri"    "Adalia bipunctata bipunctata"
## [9] "Adalia decempunctata"            "Formicidae"
## [11] "Harmonia spp."                   "Adalia bipunctata"
## [13] "Coccinella quinquepunctata"      "Coccinella sp."
## [15] "Myrmica speciosus"               "Nomada bispinosa"
## [17] "Nomada sp."                      "Formicinae"
## [19] "Apoidea"                         "Dinocampus coccinellae"
## [21] "Coleoptera"                      "Aphaenogaster spinosa"
## [23] "Myrmica sp."                     "Lasius niger"
## [25] "Formicinae"
```

autre solution avec les expressions régulières :

```
paste0(toupper(gsub("(^.).*", "\\1", d$sp)), tolower(gsub("(^.).*", "\\1", d$sp)))
```

```
## [1] "Coccinella septempunctata"      "Coccinella septempunctata"
## [3] "Coccinella septempunctata brucki" "Adalia spp"
## [5] "Coccinellidae"                  "Coccinella hieroglyphica"
## [7] "Adalia bipunctata revielieri"    "Adalia bipunctata bipunctata"
## [9] "Adalia decempunctata"            "Formicidae"
## [11] "Harmonia spp."                   "Adalia bipunctata"
## [13] "Coccinella quinquepunctata"      "Coccinella sp."
## [15] "Myrmica speciosus"               "Nomada bispinosa"
## [17] "Nomada sp."                      "Formicinae"
## [19] "Apoidea"                         "Dinocampus coccinellae"
## [21] "Coleoptera"                      "Aphaenogaster spinosa"
## [23] "Myrmica sp."                     "Lasius niger"
## [25] "Formicinae"
```

- 6) Sélectionnez uniquement les 6 données qui n'ont PAS été identifiées au moins jusqu'au genre (familles, sous-familles, etc...). Piste : il s'agit des les éléments de la colonne "sp" contenant au moins une espace (utilisez regexpr ou grepl).

```
d[!grepl(" ", d$sp), ] # solution 1
```

```
##           sp n           site
## 5 Coccinellidae 2      res nat du coupu tienne
## 10 Formicidae 3      Olloy
## 18 Formicinae 1      Oignies
## 19 Apoidea 3      Mazée
## 21 Coleoptera 6 Rés. nat. de la Montagne de la Carrière
## 25 Formicinae 3      res nat Al Florée
```

```
d[regexpr(" ", d$sp) < 0, ] # solution 2
```

```
##           sp n           site
## 5 Coccinellidae 2      res nat du coupu tienne
## 10 Formicidae 3      Olloy
## 18 Formicinae 1      Oignies
## 19 Apoidea 3      Mazée
## 21 Coleoptera 6 Rés. nat. de la Montagne de la Carrière
## 25 Formicinae 3      res nat Al Florée
```

- 7) Sélectionnez les données qui ont été identifiées au niveau générique ou infra générique (genre espèces, sous-espèces). NB : ne cherchez pas midi à 14 heures. Il suffit normalement de rajouter ou de changer un seul caractère à la ligne de code précédente.

```
d[grepl(" ", d$sp), ] # solution 1
```

```
##           sp n           site
## 1 Coccinella septempunctata 1 Mazée
## 2 Coccinella septempunctata 5 Namur
## 3 Coccinella septempunctata brucki 10 Res. Nat de Matagne
## 4 Adalia spp 33 Treignes
## 6 Coccinella hieroglyphica 1 Vaucelles
## 7 Adalia bipunctata revielieri 4 Rés. Nat. Roche Madou
## 8 Adalia bipunctata bipunctata 6 Vierves
## 9 Adalia decempunctata 1 natoye
## 11 Harmonia spp. 1 Le Mesnil
## 12 Adalia bipunctata 1 resnat du Chamousias
## 13 Coccinella quinquepunctata 2 Natoye
## 14 Coccinella sp. 1 Mazée
## 15 Myrmica specioidea 9 Réserve Naturelle de la Haie Gabaux
## 16 Nomada bispinosa 3 Dourbes
## 17 Nomada sp. 1 Doische
## 20 Dinocampus coccinellae 2 Vaucelles
## 22 Aphaenogaster spinosa 1 Regniessart
## 23 Myrmica sp. 2 Vireux
## 24 Lasius niger 1 Haybes
```

```
d[!regexpr(" ", d$sp) < 0, ] # solution 2
```

```
##           sp n           site
## 1 Coccinella septempunctata 1 Mazée
## 2 Coccinella septempunctata 5 Namur
## 3 Coccinella septempunctata brucki 10 Res. Nat de Matagne
## 4 Adalia spp 33 Treignes
## 6 Coccinella hieroglyphica 1 Vaucelles
## 7 Adalia bipunctata revielieri 4 Rés. Nat. Roche Madou
## 8 Adalia bipunctata bipunctata 6 Vierves
## 9 Adalia decempunctata 1 natoye
## 11 Harmonia spp. 1 Le Mesnil
## 12 Adalia bipunctata 1 resnat du Chamousias
## 13 Coccinella quinquepunctata 2 Natoye
## 14 Coccinella sp. 1 Mazée
## 15 Myrmica specioidea 9 Réserve Naturelle de la Haie Gabaux
## 16 Nomada bispinosa 3 Dourbes
## 17 Nomada sp. 1 Doische
## 20 Dinocampus coccinellae 2 Vaucelles
## 22 Aphaenogaster spinosa 1 Regniessart
## 23 Myrmica sp. 2 Vireux
## 24 Lasius niger 1 Haybes
```

```
d[regexpr(" ", d$sp) > 0, ] # solution 3
```

```
##           sp n           site
## 1 Coccinella septempunctata 1 Mazée
## 2 Coccinella septempunctata 5 Namur
## 3 Coccinella septempunctata brucki 10 Res. Nat de Matagne
## 4 Adalia spp 33 Treignes
## 6 Coccinella hieroglyphica 1 Vaucelles
## 7 Adalia bipunctata revielieri 4 Rés. Nat. Roche Madou
## 8 Adalia bipunctata bipunctata 6 Vierves
## 9 Adalia decempunctata 1 natoye
## 11 Harmonia spp. 1 Le Mesnil
## 12 Adalia bipunctata 1 resnat du Chamousias
```

```
## 13      Coccinella quinquepunctata 2          Natoye
## 14      Coccinella sp. 1          Mazée
## 15      Myrmica speciosus 9 Réserve Naturelle de la Haie Gabaux
## 16      Nomada bispinosa 3          Dourbes
## 17      Nomada sp. 1          Doische
## 20      Dinocampus coccinellae 2          Vauzelles
## 22      Aphaenogaster spinosa 1          Regniessart
## 23      Myrmica sp. 2          Vireux
## 24      Lasius niger 1          Haybes
```

- 8) Créez un code composé des 3 premières lettres du genre suivies des 3 premières lettres de l'espèce. Pour les données identifiées au niveau supragénérique remplacez le code obtenu par une chaîne de caractère vide.
2 pistes :

- a) utilisez une combinaison de substring et de regexr pour trouver la position du premier caractère espace
- b) utiliser gsub et les possibilités de capture des expressions régulières

```
# Solution a
esp_position <- regexr(" ", d$sp)
code <- paste(substring(d$sp, 1, 3), substring(d$sp, esp_position+1, esp_position + 3),
             sep="")
code[esp_position < 0] <- ""
code
```

```
## [1] "Cocsep" "Cocsep" "Cocsep" "Adaspp" "" "Cochie" "Adabip"
## [8] "Adabip" "Adadec" "" "Harspp" "Adabip" "Cocqui" "Cocsp."
## [15] "Myrspe" "Nombis" "Nomsp." "" "Dincoc" ""
## [22] "Aphspi" "Myrsp." "Lasnig" ""
```

```
# Solution b
code <- gsub("(.{3}).* (.{3}).*", "\\1\\2", d$sp)
code[!grepl(" ", d$sp)] <- ""
code
```

```
## [1] "Cocsep" "Cocsep" "Cocbru" "Adaspp" "" "Cochie" "Adarev"
## [8] "Adabip" "Adadec" "" "Harspp" "Adabip" "Cocqui" "Cocsp."
## [15] "Myrspe" "Nombis" "Nomsp." "" "Dincoc" ""
## [22] "Aphspi" "Myrsp." "Lasnig" ""
```

- 9) Sélectionner les données qui ont été identifiées au niveau de la sous-espèce. (NB : utilisez grep ou grepl mais pas regexr)

```
# On crée une expression régulière correspondant au début de la chaîne suivi d'un nombre
# quelconque de caractères,
# puis une espace puis un nombre quelconque de caractères puis une deuxième espace puis
# encore un nombre quelconque de
# caractères.
d[grepl("^.* .* .*", d$sp),]
```

```
##          sp n          site
## 3 Coccinella septempunctata brucki 10 Res. Nat de Matagne
## 7 Adalia bipunctata revielieri 4 Rés. Nat. Roche Madou
## 8 Adalia bipunctata bipunctata 6 Vierves
```

```
# On pourrait en fait utiliser gregexpr pour identifier les noms contenant plus d'une
# espace mais le résultat est sous forme de liste
# Et on a pas encore vu à ce stade comment par exemple aller mesurer la longueur des
# éléments d'une liste (avec sapply).
# De plus, la solution avec grep est plus simple.
sapply(gregexpr(" ", d$sp), length) # nombre d'espace dans chaque élément)
```

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

```
d[sapply(gregexpr(" ", d$sp), length) > 1, ]
```

```
##               sp n               site
## 3 Coccinella septempunctata brucki 10 Res. Nat de Matagne
## 7      Adalia bipunctata revieri  4 Rés. Nat. Roche Madou
## 8      Adalia bipunctata bipunctata 6                Vierves
```

- 10) Pour ces espèces identifiées au niveau de la sous-espèce, transformer le nom de manière à ne garder que le Genre et l'espèce. Utilisez gsub et les possibilités de capture des expressions régulières

```
gsub("(^.* .*).*", "\\1", d$sp)
```

```
## [1] "Coccinella septempunctata" "Coccinella septempunctata"
## [3] "Coccinella septempunctata" "Adalia spp"
## [5] "Coccinellidae" "Coccinella hieroglyphica"
## [7] "Adalia bipunctata" "Adalia bipunctata"
## [9] "Adalia decempunctata" "Formicidae"
## [11] "Harmonia spp." "Adalia bipunctata"
## [13] "Coccinella quinquepunctata" "Coccinella sp."
## [15] "Myrmica speciosus" "Nomada bispinosa"
## [17] "Nomada sp." "Formicinae"
## [19] "Apoidea" "Dinocampus coccinellae"
## [21] "Coleoptera" "Aphaenogaster spinosa"
## [23] "Myrmica sp." "Lasius niger"
## [25] "Formicinae"
```

- 11) Sélectionnez uniquement les données qui ont été identifiées au niveau générique mais pas spécifique (ie Adalia sp., Adalia spp, etc...). NB : il y en a 5, pas 7 !

```
# Le code suivant sélectionne les taxons contenant une espace suivie de sp mais ce n'est
# pas suffisant car il sélectionne aussi Myrmica speciosus et Aphaenogaster spinosa
d[grepl(" sp", d$sp), "sp"]
```

```
## [1] "Adalia spp" "Harmonia spp." "Coccinella sp."
## [4] "Myrmica speciosus" "Nomada sp." "Aphaenogaster spinosa"
## [7] "Myrmica sp."
```

```
# Avec le code suivant, on sélectionne les taxons contenant une espace suivie de sp suivi
# de 0 ou 1 "p" suivi de 0 ou 1 point (le double backslash \\ est nécessaire pour
# échapper à la signification du point tout seul qui vaut pour n'importe quel caractère)
d[grepl(" spp?\\.?", d$sp), "sp"]
```

```
## [1] "Adalia spp" "Harmonia spp." "Coccinella sp." "Nomada sp."
## [5] "Myrmica sp."
```

- 12) Sélectionnez toutes les données observées dans des réserves naturelles. (NB il y en a 7, pas 9) Dans la colonne "site" les données observées dans des réserves naturelles sont systématiquement indiquées mais de manière peu uniformisée (res. nat, résnat, etc...)

```
d[grep("r[eé]s.* ?nat.*", d$site, ignore.case=TRUE),]
```

```
##                               sp  n
## 3 Coccinella septempunctata brucki 10
## 5           Coccinellidae         2
## 7      Adalia bipunctata revieleri  4
## 12           Adalia bipunctata    1
## 15           Myrmica specioidea    9
## 21           Coleoptera           6
## 25           Formicinae           3
##
##                               site
## 3           Res. Nat de Matagne
## 5           res nat du coupu tienne
## 7           Rés. Nat. Roche Madou
## 12          resnat du Chamousias
## 15  Réserve Naturelle de la Haie Gabaux
## 21 Rés. nat. de la Montagne de la Carrière
## 25          res nat Al Florée
```

```
# plus simple mais pas correct car on sélectionne aussi les données de Natoye
d[grep("nat", d$site, ignore.case=TRUE),]
```

```
##                               sp  n
## 3 Coccinella septempunctata brucki 10
## 5           Coccinellidae         2
## 7      Adalia bipunctata revieleri  4
## 9      Adalia decempunctata    1
## 12           Adalia bipunctata    1
## 13  Coccinella quinquepunctata    2
## 15           Myrmica specioidea    9
## 21           Coleoptera           6
## 25           Formicinae           3
##
##                               site
## 3           Res. Nat de Matagne
## 5           res nat du coupu tienne
## 7           Rés. Nat. Roche Madou
## 9                               natoye
## 12          resnat du Chamousias
## 13                               Natoye
## 15  Réserve Naturelle de la Haie Gabaux
## 21 Rés. nat. de la Montagne de la Carrière
## 25          res nat Al Florée
```

- 13) Sélectionnez les données du genre Adalia observées dans des réserves naturelles (2 données). Il faut impérativement utiliser ici grepl, la version de grep qui retourne un vecteur logique

```
d[grepl("r[eé]s.* ?nat.*", d$site, ignore.case=TRUE) &
  grepl("^Adalia ", d$sp, ignore.case=TRUE),]
```

```
##                               sp  n                               site
## 7 Adalia bipunctata revieleri 4 Rés. Nat. Roche Madou
## 12 Adalia bipunctata 1 resnat du Chamousias
```

Exercice 2

Voici un vecteur de coordonnées géographiques en Degrés Minutes Secondes (DMS).

```
DMS <- c("50°45'58.32\"N", "50°51'12.73\"S", "4°32'5.66\"E", "3°34'53.66\"W",
        "2°28'7.26\"O", "50°36'23.88\"", "50°36'23\"", "50°36'23.65\"'N (N2)",
        "50°36'23.65\"'N (N2)", "50°36'07,26\"N", " 50 ° 45 ' 58,32 \" N")
```

On veut le transformer en degrés décimaux ($DD = D + M/60 + S/(60*60)$). Les 4 première coordonnées sont correctes et cohérentes. Les suivantes n'ont pas toutes été encodée de la même manière (NB : inspiré de faits réels...)

Utilisez les expressions régulières pour extraire les valeurs correspondant aux degrés, aux minutes, aux secondes et à l'hémisphère (par défaut on considère qu'on est dans l'hémisphère nord et à l'est du méridien de Greenwich).

Conseil : Commencez par transformer les 4 premières valeurs puis coplexifiez progressivement vos expressions pour prendre en compte tous les cas particuliers...

Vous devriez obtenir les valeurs suivantes :

```
c(50.7662, -50.853536, 4.534906, -3.581572, -2.468683, 50.606633, 50.606389, 50.606569, 50.606569, 50.602017, 50.7662)
```

```
D <- as.numeric(gsub(pattern = " *([0-9]{1,2}) *°.*", "\\1", x = DMS))
M <- as.numeric(gsub(pattern = ".*° *([0-9]{1,2}) *'.*", "\\1", x = DMS))

S <- gsub(pattern = ".*°.*' *([0-9]{1,2}[\\.,\\,]{0,1}[0-9]{0,2}) *([\\\"' ]).*",
             "\\1", x = DMS)
S
```

```
## [1] "58.32" "12.73" "5.66" "53.66" "7.26" "23.88" "23" "23.65"
## [9] "23.65" "07,26" "58,32"
```

```
# remplacer la virgule par un point puis transformer en nombres
```

```
S <- as.numeric(gsub(",", "\\.", S))
S
```

```
## [1] 58.32 12.73 5.66 53.66 7.26 23.88 23.00 23.65 23.65 7.26 58.32
```

```
sign <- gsub(pattern = ".*°.*'.*([\\\"' ])*([NSEOW]).*", "\\1", x = DMS)
sign
```

```
## [1] "N" "S" "E" "W"
## [5] "O" "50°36'23.88\"" "50°36'23\"" "N"
## [9] "N" "N" "N"
```

```
sign <- ifelse(sign %in% c("S", "O", "W"), -1, 1)
sign
```

```
## [1] 1 -1 1 -1 -1 1 1 1 1 1 1
```

```
DD <- (D + M/60 + S/(60*60)) * sign
DD
```

```
## [1] 50.766200 -50.853536 4.534906 -3.581572 -2.468683 50.606633
## [7] 50.606389 50.606569 50.606569 50.602017 50.766200
```


Structures de contrôle : fonctions, boucles, exécutions conditionnelles,...

Exercice 0

Petits exercices simples de mise en jambes...

- 1) Créez une fonction “salut” qui écrit simplement “Bonjour”. Cette fonction n’a pas d’arguments : salut() écrira “Bonjour”
- 2) Ajoutez un argument “qui” pour pouvoir choisir un nom quelconque: salut(qui = “Gilles”) écrira “Bonjour Gilles”, salut(“Louis”) écrira “Bonjour Louis”
- 3) Ajoutez un argument “english” de type vrai/faux permettant de choisir la langue : salut(qui = “Gilles”, english = FALSE) écrira “Bonjour Gilles” et salut(“Philippe”, english = TRUE) écrira “Hello Philippe”

```
salut <- function() {print("Bonjour")}  
salut()
```

```
## [1] "Bonjour"
```

```
salut <- function(qui = "") {  
  a <- paste("Bonjour", qui)  
  print(a)  
}  
salut()
```

```
## [1] "Bonjour "
```

```
salut("Georges")
```

```
## [1] "Bonjour Georges"
```

```
salut <- function(qui = "", english = FALSE) {  
  if(english == TRUE) {  
    a <- paste("Hello", qui)  
  } else {  
    a <- paste("Bonjour", qui)  
  }  
  print(a)  
}  
salut()
```

```
## [1] "Bonjour "
```

```
salut(english = TRUE)
```

```
## [1] "Hello "
```

```
salut("Philippe")
```

```
## [1] "Bonjour Philippe"
```

```
salut("Philippe", TRUE)
```

```
## [1] "Hello Philippe"
```

- 4) Au moyen d'une boucle for, calculez pour chaque ligne de la matrice suivante, la moyenne et l'écart type (sd). Stockez ces valeurs dans une nouvelle matrice à 2 colonnes et 10 lignes. NB : en pratique on ne devrait pas utiliser de boucle pour ce genre de calculs (on utiliserait rowMeans et apply).

```
mat <- matrix(c(1:40), 10, 4, byrow=TRUE)
```

```
meansd <- matrix(NA, 10,2) # matrice vide pour stocker les résultats
colnames(meansd) <- c("mean", "sd")

for(i in 1:nrow(mat)) {
  meansd[i,1] <- mean(mat[i,])
  meansd[i,2] <- sd(mat[i,])
}
meansd
```

```
##      mean      sd
## [1,]  2.5 1.290994
## [2,]  6.5 1.290994
## [3,] 10.5 1.290994
## [4,] 14.5 1.290994
## [5,] 18.5 1.290994
## [6,] 22.5 1.290994
## [7,] 26.5 1.290994
## [8,] 30.5 1.290994
## [9,] 34.5 1.290994
## [10,] 38.5 1.290994
```

- 5) Utilisation de la fonction merge. On a un dataset “obs” qui contient un nombre d’observations pour une série d’espèces (sp) identifiées par un code à 6 lettres.
On veut ajouter une colonne à ce dataset avec le nom scientifique complet correspondant. Les correspondances code - nom scientifiques se trouvent dans le dataset “tax”.
Attention certains codes de la table obs n’ont pas de correspondance dans “tax”. Veillez malgré tout à ne pas perdre de données dans la table “obs”.

```
obs <- data.frame (
  sp = c("cocund", "cocsep", "cocsep", "scysut", "psyvig", "adabip", "rhychr", "adabip",
        "psyvig", "cocsep"),
  nbr = c(2,3,46,3,4,8,9,20,4,3)
)

tax <- data.frame(
  code = c("adabip", "cocsep", "cocund", "psyvig", "subvig", "epiarg"),
  taxon = c("Adalia bipunctata", "Coccinella septempunctata",
            "Coccinella undecimpunctata", "Psyllobora vigintiduopunctata",
            "Subcoccinella vigintiquatuorpunctata", "Epilachna argus")
)
```

```
merge(obs, tax, by.x = "sp", by.y="code", all.x = TRUE)
```

```
##      sp nbr      taxon
## 1 adabip  20 Adalia bipunctata
```

```
## 2  adabip    8      Adalia bipunctata
## 3  cocsep    3      Coccinella septempunctata
## 4  cocsep    3      Coccinella septempunctata
## 5  cocsep   46      Coccinella septempunctata
## 6  cocund    2      Coccinella undecimpunctata
## 7  psyvig    4      Psyllobora vigintiduopunctata
## 8  psyvig    4      Psyllobora vigintiduopunctata
## 9  rhychr    9      <NA>
## 10 scysut    3      <NA>
```

Exercice 1

- Construisez une fonction qui crée un code composé des 3 premières lettres du nom de genre et des 3 premières lettres du nom d'espèce. L'exercice a déjà été fait précédemment il suffit ici de l'encapsuler dans une fonction. Pour rappel les lignes de code suivant permettaient de faire le travail (en dehors d'une fonction) :
`esp_position <- regexpr(" ", x)`
`code <- paste(substring(x, 1, 3), substring(x, esp_position + 1, esp_position + 3), sep="")`
 N'oubliez pas de créer un vecteur exemple de noms scientifiques pour tester votre fonction.

```
noms <- c("Libellula depressa", "Coenagrion mercuriale", "Ceriagrion tenellum",
          "Lestes sponsa")

codesp <- function (x) {
  esp_position <- regexpr(" ", x)
  code <- paste(substring(x, 1, 3), substring(x, esp_position + 1, esp_position + 3),
               sep="")
  return(code)
}
codesp(noms)
```

```
## [1] "Libdep" "Coemer" "Certen" "Lesspo"
```

- Modifiez ensuite cette fonction de façon à ce qu'on puisse choisir le nombre de lettres prises en compte

```
codesp <- function (x, nblettres = 3) {
  esp_position <- regexpr(" ", x)
  code <- paste(substring(x, 1, nblettres),
               substring(x, esp_position + 1, esp_position + nblettres), sep="")
  return(code)
}
codesp(noms)
```

```
## [1] "Libdep" "Coemer" "Certen" "Lesspo"
```

```
codesp(noms, 5)
```

```
## [1] "Libeldepre" "Coenamercu" "Ceriatenel" "Lestespons"
```

- Modifiez encore votre fonction de façon à ce que l'utilisateur puisse choisir de mettre toutes les lettres du code en majuscule

```
codesp <- function (x, nblettres = 3, uppercase = FALSE) {
  esp_position <- regexpr(" ", x)
  code <- paste(substring(x, 1, nblettres),
```

```

        substring(x, esp_position+1, esp_position + nblettres), sep="")
  if(uppercase) code <- toupper(code)
  return(code)
}
codesp(noms, 5)

```

```
## [1] "Libeldepre" "Coenamercu" "Ceriatenel" "Lestespons"
```

```
codesp(noms, 5, TRUE)
```

```
## [1] "LIBELDEPRE" "COENAMERCU" "CERIATENEL" "LESTESPONS"
```

- Modifiez votre fonction de façon à ce que l'utilisateur puisse choisir de mettre toutes les lettres du code en majuscule ou en minuscule ou de pas y toucher

```

codesp <- function (x, nblettres = 3, case = c("upper", "lower", "asis")) {
  esp_position <- regexpr(" ", x)
  code <- paste(substring(x, 1, nblettres),
               substring(x, esp_position+1, esp_position + nblettres), sep="")
  case <- match.arg(case) #permet d'éviter un message et vérifie que les arg. sont bons
  if(case == "upper") code <- toupper(code) else
    if(case == "lower") code <- tolower(code)
  return(code)
}
codesp(noms, 5)

```

```
## [1] "LIBELDEPRE" "COENAMERCU" "CERIATENEL" "LESTESPONS"
```

```
codesp(noms, 5, "lower")
```

```
## [1] "libeldepre" "coenamercu" "ceriatenel" "lestespons"
```

```
codesp(noms, 5, "asis")
```

```
## [1] "Libeldepre" "Coenamercu" "Ceriatenel" "Lestespons"
```

```
codesp(noms, 5, "nimportequoi")
```

```
## Error in match.arg(case): 'arg' should be one of "upper", "lower", "asis"
```

Exercice 2

- Chargez le fichier “ladybirds_aggr.txt”. Changez les valeurs des deux premières colonnes pour des valeurs plus explicites : tree : M = Mapple, L = Lime, P = Pine ; landsc : U = Urban, S = Suburban, R = Rural (le plus simple est d'utiliser la fonction `levels()` <-)

```

setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
d <- read.table("data/ladybirds_aggr.txt", sep="\t", header=TRUE)

levels(d$landsc) <- c("Rural", "Suburban", "Urban")
levels(d$tree) <- c("Lime", "Mapple", "Pine")

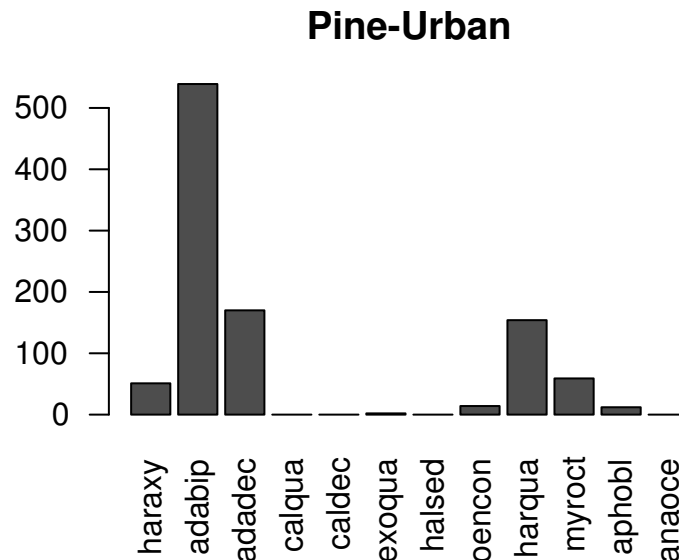
```

- Réordonnez le tableau selon l'espèce d'arbre (tree) et ensuite du paysage (landsc) et dans l'ordre inverse de l'ordre alphabétique.

```
d <- d[order(d$tree, d$landsc,decreasing=TRUE),]
```

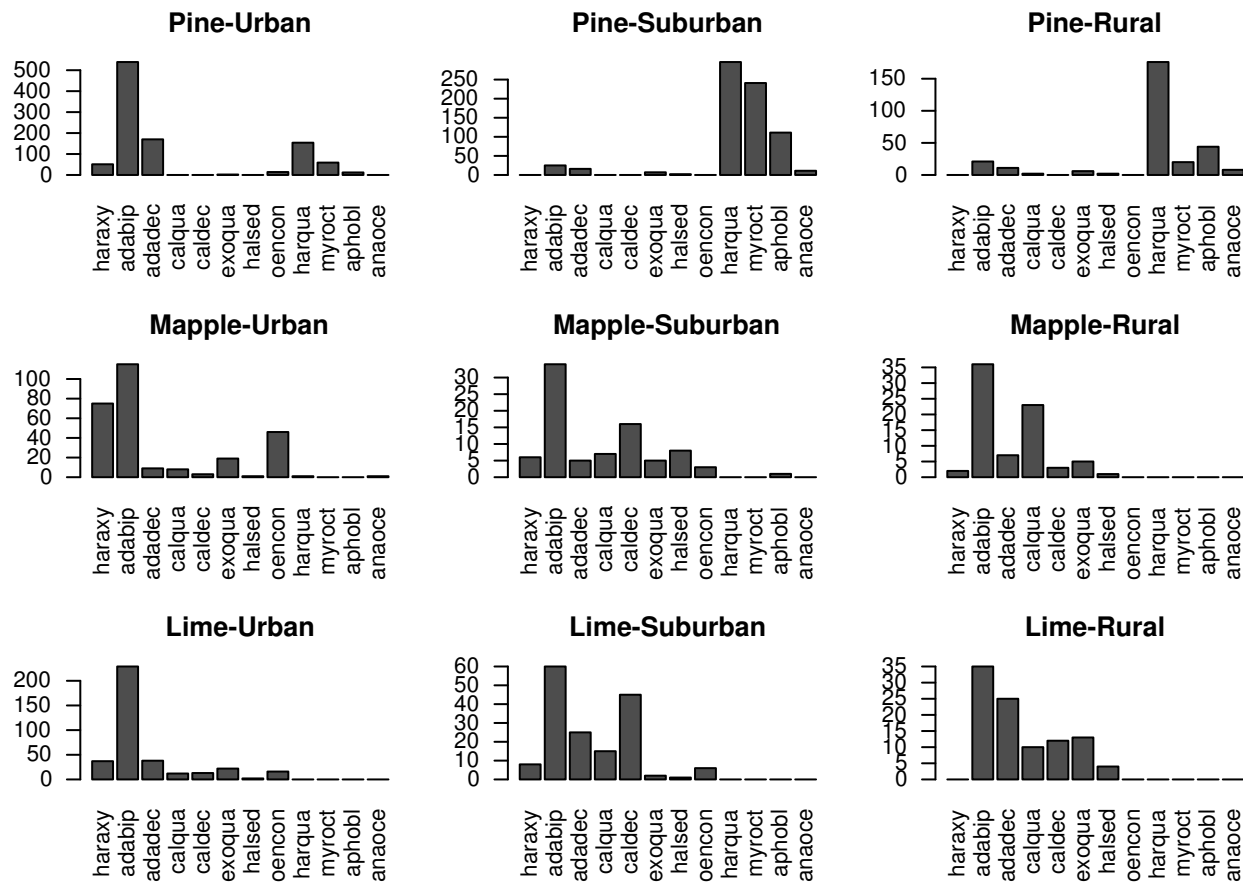
- Tracez un barplot de la première ligne du tableau représentant l'abondance des différentes espèces de coccinelles sur Pins en milieu urbain. Ajoutez dans le titre l'espèce d'arbre et le paysage correspondant (paramètre "main" de la fonction plot).

```
# dev.new(width = 10/2.54, height = 8/2.54)
par(mar = c(4,3,3,1), las=2)
barplot(as.matrix(d[1, -c(1,2)]), main = paste(d[1,1], d[1,2], sep="-"))
```



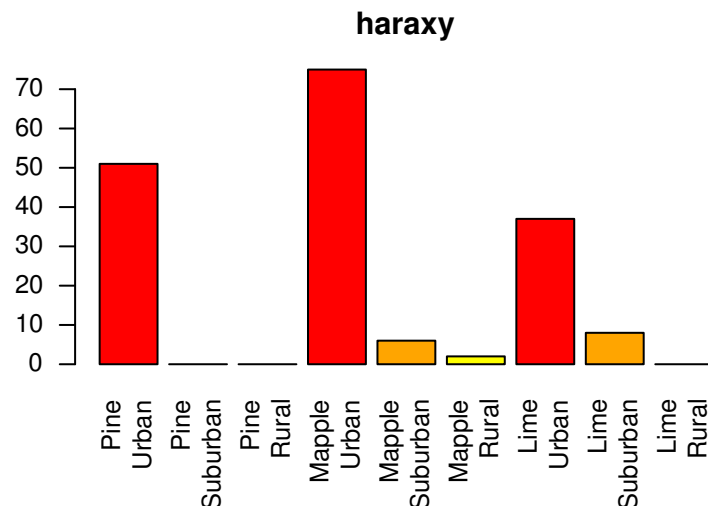
- A l'aide d'une boucle "for" répétez ensuite le même graphique pour chaque ligne du tableau en adaptant les titres. Juste avant la boucle for placez la commande suivante : `par(mfrow = c(3,3), mar = c(4,3,3,1), las=2, cex = 0.7)` `mfrow` divise la fenêtre graphique en 9 parties, `mar` ajuste la taille des marges, `las` force les étiquettes à être perpendiculaires à l'axe et `cex` diminue la taille des caractères (on détaillera ces options plus loin dans la partie sur les graphiques).
Ne vous tracassez pas trop de la mise en forme des graphiques qui sera abordée plus loin.

```
# dev.new(width = 17/2.54, height = 12/2.54)
par(mfrow = c(3,3), mar = c(4,3,3,1), las=2, cex = 0.7)
for (i in 1:nrow(d)){
  barplot(as.matrix(d[i, -c(1,2)]), main = paste(d[i,1], d[i,2], sep="-"))
}
```

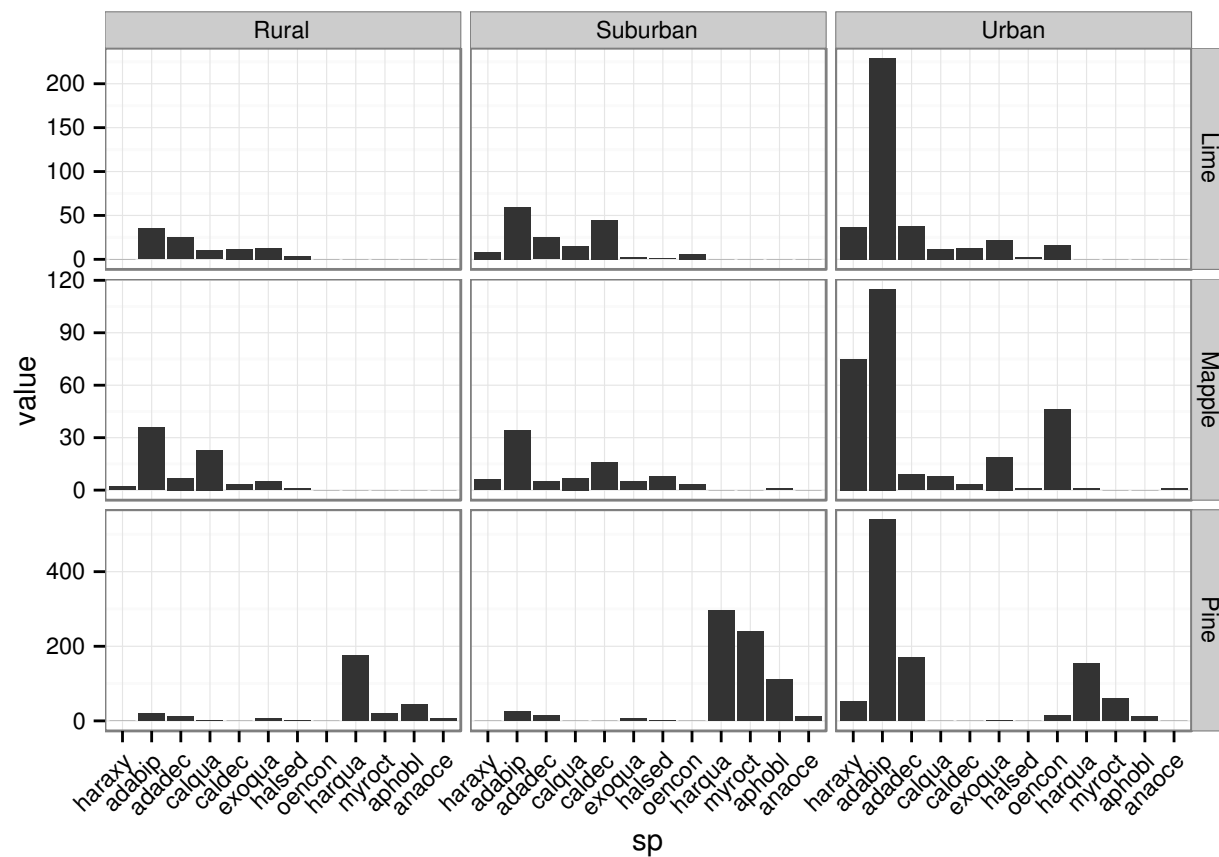


- Faites ensuite un barplot pour une espèce de coccinelle en fonction des différentes combinaisons d'arbre et de paysage (on va travailler sur les colonnes, avec une barre par ligne du tableau de données). Vous devrez ajouter vous-même les étiquettes au moyen de l'argument "names.arg" de la fonction barplot. Pour créer ces étiquettes, collez les valeurs des deux premières colonnes et séparez les par un caractère "\n" qui représente un retour à la ligne.
- Faites ensuite un graphique similaire pour chacune des 12 espèces au moyen d'une boucle. Avant la boucle for, ajoutez la ligne de code suivante : `par(mfrow = c(4,3), mar = c(4.5,2,3,1), las=2, cex = 0.65)` N'oubliez pas d'ajouter les titres appropriés.

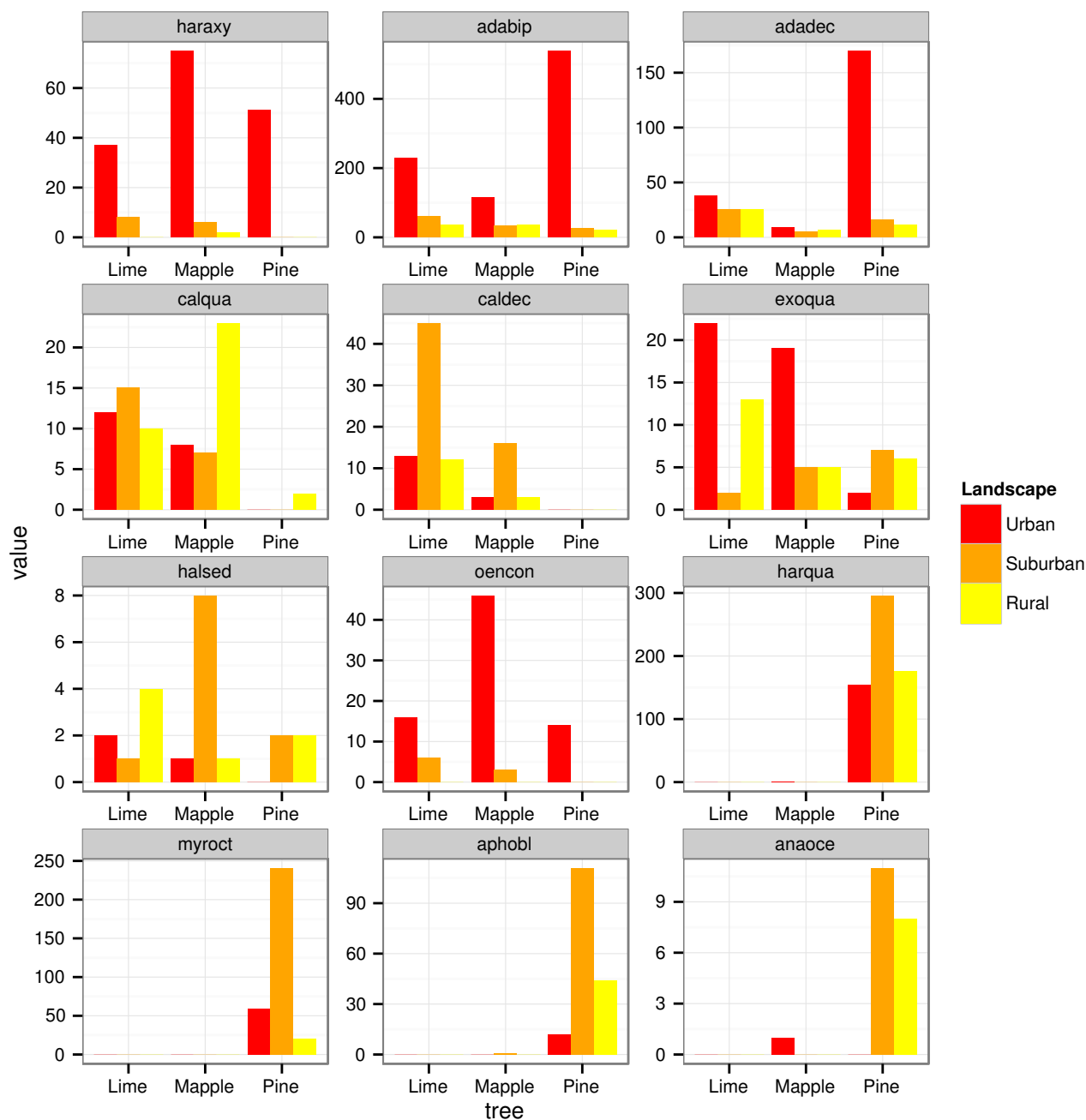
```
# dev.new(width = 10/2.54, height = 7/2.54)
par(mfrow=c(1,1),mar = c(4.5,2,3,1), las=2, cex = 0.8)
barplot(d[, "haraxy"], names.arg = paste(d$tree, d$landsc, sep="\n"),
        main = colnames(d)[3], col = c("red", "orange", "yellow"))
```



```
# dev.new(width = 18/2.54, height = 18/2.54)
par(mfrow = c(4,3), mar = c(4.5,2,3,1), las=2, cex = 0.65)
for(i in 3:ncol(d)){
  barplot(d[,i], names.arg = paste(d$tree, d$landsc, sep="\n"),
    main = colnames(d)[i], col = c("red", "orange", "yellow"))
}
```

```
# dev.new(width = 18/2.54, height = 18/2.54)
mnd$landsc <- factor(mnd$landsc, levels = c("Urban", "Suburban", "Rural"))
ggplot(mnd, aes(y = value, x = tree, fill = landsc)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ sp, scales = "free", nrow = 4) +
  theme_bw(10) +
  scale_fill_manual(name = "Landscape", values = c("red", "orange", "yellow"))
```



Exercice 3

Chargez les fichiers `biche1.csv` (séparateur : tabulations) et `ephemerides.csv` (séparateur : point virgule).

Le fichier “biche” contient les dates, heures et localisations GPS d’une biche. Le fichier éphémérides contient les heures de coucher (sunrise) et de coucher de soleil (sunset) à Uccle.

On veut pour chaque position de la biche, déterminer si il faisait jour ou nuit au moment de l’enregistrement.

Il faut commencer par formater correctement les colonnes date, sunset et sunrise du fichier d’éphémérides et les colonnes date et time du fichier biche. Attention, les éphémérides sont en heure civile pour l’Europe centrale (tz=“CET”) alors que le fichier biche est en temps universel (tz=“UTC”). Afin que les comparaisons et calculs soient corrects, il faut donc bien indiquer le fuseau horaire au moyen de l’argument tz au moment où on formate les temps (avec `strptime`). Pour certaines opérations, R retourne un message d’avertissement quand les temps impliqués ne sont pas dans le même tz mais les comparaisons sont néanmoins correctes.

Fusionnez ensuite les deux tables (merge) et créez une colonne jour/nuit. Il y a 155 données de jour et 288 données de nuit.

nuit dans ce fichier. Vous pouvez le vérifier par un `summary()` sur votre colonne jour/nuit après l'avoir transformée en facteur.

```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
biche <- read.table("data/biche1.csv", sep="\t", header=TRUE)
ephem <- read.table("data/ephemerides.csv", sep=";", header=TRUE)

# On transforme les données de ephem en dates et temps, en spécifiant le fuseau horraire
summary(ephem)
```

```
##           date      sunrise      sunset      length
## 01 01 2011:  1  5:29 : 12  16:37 : 10  Min.   :33.00
## 01 02 2011:  1  8:44 :  7  22:00 : 10  1st Qu.:34.00
## 01 03 2011:  1  8:45 :  7  16:38 :  6  Median :36.00
## 01 04 2011:  1  5:30 :  6  21:59 :  6  Mean   :37.34
## 01 05 2011:  1  8:43 :  5  16:39 :  4  3rd Qu.:39.00
## 01 06 2011:  1  5:31 :  4  16:40 :  4  Max.   :46.00
## (Other) :359 (Other):324 (Other):325
##      dawn_start      dusk_stop
## 4:43 :  8  22:46 : 10
## 8:05 :  8  17:16 :  9
## 4:44 :  6  17:17 :  7
## 8:06 :  6  22:45 :  5
## 8:04 :  5  17:18 :  4
## 4:45 :  4  17:19 :  4
## (Other):328 (Other):326
```

```
ephem$date <- as.Date(ephem$date, format = "%d %m %Y")
ephem$sunrise <- strptime(paste(ephem$date, ephem$sunrise),
                          format = "%Y-%m-%d %H:%M", tz="CET")
ephem$sunset <- strptime(paste(ephem$date, ephem$sunset),
                        format = "%Y-%m-%d %H:%M", tz="CET")

# On transforme les variables temps et date du jeu de données biche
summary(biche)
```

```
##           id      date      time      x
## Min.   : 0.0  9/02/2011:76  0:15:27 : 2  Min.   :285960
## 1st Qu.:110.5 7/02/2011:74  10:00:42: 2  1st Qu.:286262
## Median :221.0 5/02/2011:68  11:00:23: 2  Median :286310
## Mean   :221.0 6/02/2011:67  12:30:44: 2  Mean   :286359
## 3rd Qu.:331.5 8/02/2011:61  1:30:52 : 2  3rd Qu.:286406
## Max.   :442.0 4/02/2011:40  16:30:12: 2  Max.   :287348
##           (Other) :57 (Other):431
##           y      lat      long
## Min.   :130091  Min.   :50.47  Min.   :6.285
## 1st Qu.:132432  1st Qu.:50.49  1st Qu.:6.289
## Median :132604  Median :50.49  Median :6.290
## Mean   :132529  Mean   :50.49  Mean   :6.290
## 3rd Qu.:132807  3rd Qu.:50.49  3rd Qu.:6.291
## Max.   :133212  Max.   :50.49  Max.   :6.304
##
```

```
biche$date <- as.Date(biche$date, format = "%d/%m/%Y")
biche$time <- strptime(paste(biche$date, biche$time),
                      format = "%Y-%m-%d %H:%M:%S", tz="UTC")

# fusion des tables
biche <- merge(biche, ephem[, 1:3], by = "date", all.x=TRUE)
summary(biche)
```

```
##      date            id            time
## Min.   :2011-02-01   Min.   : 0.0   Min.   :2011-02-01 20:13:23
## 1st Qu.:2011-02-05   1st Qu.:110.5   1st Qu.:2011-02-05 13:23:11
## Median :2011-02-07   Median :221.0   Median :2011-02-07 03:45:16
## Mean   :2011-02-06   Mean   :221.0   Mean   :2011-02-07 02:26:20
## 3rd Qu.:2011-02-08   3rd Qu.:331.5   3rd Qu.:2011-02-08 18:38:31
## Max.   :2011-02-10   Max.   :442.0   Max.   :2011-02-10 09:00:46
##      x            y            lat            long
## Min.   :285960   Min.   :130091   Min.   :50.47   Min.   :6.285
## 1st Qu.:286262   1st Qu.:132432   1st Qu.:50.49   1st Qu.:6.289
## Median :286310   Median :132604   Median :50.49   Median :6.290
## Mean   :286359   Mean   :132529   Mean   :50.49   Mean   :6.290
## 3rd Qu.:286406   3rd Qu.:132807   3rd Qu.:50.49   3rd Qu.:6.291
## Max.   :287348   Max.   :133212   Max.   :50.49   Max.   :6.304
##      sunrise            sunset
## Min.   :2011-02-01 08:19:00   Min.   :2011-02-01 17:34:00
## 1st Qu.:2011-02-05 08:13:00   1st Qu.:2011-02-05 17:41:00
## Median :2011-02-07 08:10:00   Median :2011-02-07 17:44:00
## Mean   :2011-02-06 23:00:55   Mean   :2011-02-07 08:34:12
## 3rd Qu.:2011-02-08 08:08:00   3rd Qu.:2011-02-08 17:46:00
## Max.   :2011-02-10 08:05:00   Max.   :2011-02-10 17:50:00
```

```
biche$period <- ifelse(biche$time > biche$sunrise & biche$time < biche$sunset,
                      "day", "night")
```

```
## Warning in check_tzones(e1, e2): 'tzone' attributes are inconsistent
```

```
## Warning in check_tzones(e1, e2): 'tzone' attributes are inconsistent
```

```
biche$period <- as.factor(biche$period)
summary(biche$period)
```

```
##   day night
##  155   288
```

Exercice 4

Un des but de l'exercice précédent était d'utiliser la fonction `merge`. Cependant, il serait plus efficace dans ce cas précis de pouvoir déterminer la période de la journée sans passer par un fichier extérieur. C'est ce que permet de faire la fonction `sunriset` du package `maptools` qui calcule l'heure de coucher ou de lever de soleil sur base des coordonnées géographiques et de la date. La fonction `crepuscule` permet également de déterminer les heures de pénombre.

Ecrivez une fonction qui détermine pour n'importe quelle date-heure en un point précis si il s'agit du jour ou de la nuit. Vous devez utiliser les colonnes longitude et latitude et la méthode appelée “`## S4 method for signature 'matrix,POSIXct'`” dans l'aide de `sunriset`. Attention, par défaut le résultat de `sunriset` est donné dans le fuseau horaire local. Si le fuseau horaire du vecteur date-temps reçu en argument par `sunriset` est bien spécifié, la valeur retournée par `sunriset` sera dans ce fuseau horaire. Il peut y avoir de légères différences entre les résultats de `sunriset` et le fichier `ephemerides.csv`, ce dernier étant calculé pour Uccle (il y a 6 différences). Vous pouvez utiliser les latitude et longitude de l'observatoire d'Uccle comme valeurs par défaut à votre fonction (`long = 4.357886`, `lat = 50.79873`)

```
# long lat par défaut calculé pour l'observatoire d'Uccle
daynight <- function(time, long = 4.357886, lat = 50.79873) {

  require(maptools)

  crd <- as.matrix(cbind(long, lat))
  time <- as.POSIXct(time)
```

```

sunset <- sunriset(crd= crd, dateTime = time, POSIXct.out=TRUE,
                  direction = "sunset"),[2]
sunrise <- sunriset(crd= crd, dateTime = time, POSIXct.out=TRUE,
                   direction = "sunrise"),[2]

ifelse(time > sunrise & time < sunset, "day", "night")
}

biche$periodbis <- daynight(biche$time, biche$long, biche$lat)
biche[1:20,c("time", "sunrise", "sunset", "period", "periodbis")]

```

```

##           time           sunrise           sunset period
## 1  2011-02-01 20:13:23 2011-02-01 08:19:00 2011-02-01 17:34:00 night
## 2  2011-02-02 00:01:26 2011-02-02 08:18:00 2011-02-02 17:35:00 night
## 3  2011-02-02 08:01:23 2011-02-02 08:18:00 2011-02-02 17:35:00 day
## 4  2011-02-02 12:01:42 2011-02-02 08:18:00 2011-02-02 17:35:00 day
## 5  2011-02-02 16:01:24 2011-02-02 08:18:00 2011-02-02 17:35:00 day
## 6  2011-02-03 08:01:41 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 7  2011-02-03 13:45:51 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 8  2011-02-03 15:00:47 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 9  2011-02-03 15:16:03 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 10 2011-02-03 15:31:11 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 11 2011-02-03 15:46:01 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 12 2011-02-03 16:01:13 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 13 2011-02-03 16:31:32 2011-02-03 08:16:00 2011-02-03 17:37:00 day
## 14 2011-02-03 16:46:00 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 15 2011-02-03 17:00:22 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 16 2011-02-03 17:15:27 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 17 2011-02-03 17:31:19 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 18 2011-02-03 17:46:44 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 19 2011-02-03 18:02:00 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 20 2011-02-03 19:30:24 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## periodbis
## 1      night
## 2      night
## 3       day
## 4       day
## 5       day
## 6       day
## 7       day
## 8       day
## 9       day
## 10      day
## 11      day
## 12      day
## 13     night
## 14     night
## 15     night
## 16     night
## 17     night
## 18     night
## 19     night
## 20     night

```

```

sum(biche$period != biche$periodbis) # nombre de différences

```

```

## [1] 6

```

```
# à l'observatoire d'Uccle --> les deux approches correspondent parfaitement
biche$periodbis <- daynight(biche$time)
biche[1:20,c("time", "sunrise", "sunset", "period", "periodbis")]
```

```
##           time           sunrise           sunset period
## 1  2011-02-01 20:13:23 2011-02-01 08:19:00 2011-02-01 17:34:00 night
## 2  2011-02-02 00:01:26 2011-02-02 08:18:00 2011-02-02 17:35:00 night
## 3  2011-02-02 08:01:23 2011-02-02 08:18:00 2011-02-02 17:35:00  day
## 4  2011-02-02 12:01:42 2011-02-02 08:18:00 2011-02-02 17:35:00  day
## 5  2011-02-02 16:01:24 2011-02-02 08:18:00 2011-02-02 17:35:00  day
## 6  2011-02-03 08:01:41 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 7  2011-02-03 13:45:51 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 8  2011-02-03 15:00:47 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 9  2011-02-03 15:16:03 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 10 2011-02-03 15:31:11 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 11 2011-02-03 15:46:01 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 12 2011-02-03 16:01:13 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 13 2011-02-03 16:31:32 2011-02-03 08:16:00 2011-02-03 17:37:00  day
## 14 2011-02-03 16:46:00 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 15 2011-02-03 17:00:22 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 16 2011-02-03 17:15:27 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 17 2011-02-03 17:31:19 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 18 2011-02-03 17:46:44 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 19 2011-02-03 18:02:00 2011-02-03 08:16:00 2011-02-03 17:37:00 night
## 20 2011-02-03 19:30:24 2011-02-03 08:16:00 2011-02-03 17:37:00 night
##      periodbis
## 1          night
## 2          night
## 3           day
## 4           day
## 5           day
## 6           day
## 7           day
## 8           day
## 9           day
## 10          day
## 11          day
## 12          day
## 13          day
## 14         night
## 15         night
## 16         night
## 17         night
## 18         night
## 19         night
## 20         night
```

```
sum(biche$period != biche$periodbis) # nombre de différences
```

```
## [1] 0
```

Exercice 5

Au moyen d'une boucle, modifiez le code des deux exercices précédents de façon à répéter ces opérations pour les 10 fichiers biche1.csv, biche2.csv, ... Votre programme devrait fonctionner pour n'importe quel nombre de fichiers.

Utilisez la fonction `list.files` pour faire la liste des fichiers visés et la stocker dans un vecteur.

Utilisez ensuite ce vecteur pour lire successivement les différents fichiers dans une boucle et ajoutez leur la colonne jour/nuit.

Ajoutez en plus une colonne identifiant chaque biche (chaque fichier).

Vous pouvez stocker chaque jeu de données dans les éléments d'une liste.

```

setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices/data")
fichiers <- list.files(pattern = "^biche[0-9]+\\.csv$")

# liste vide qui va servir à stocker les jeux de données
multibiches <- as.list(vector(length = length(fichiers)))

for (i in 1:length(fichiers)) {

  biche <- read.table(fichiers[i], sep="\t", header=TRUE)

  #' On transforme les variables temps et date
  biche$date <- as.Date(biche$date, format = "%d/%m/%Y")
  biche$time <- strptime(paste(biche$date, biche$time), format = "%Y-%m-%d %H:%M:%S",
                        tz="UTC")

  biche$period <- daynight(time = biche$time , lat = biche$lat, long = biche$long)

  # on ajoute le nom de fichier dans une colonne
  nom <- gsub("(^biche[0-9]+\\.csv$)", "\\1", fichiers[i])
  biche$file <- nom

  multibiches[[i]] <- biche
  # on ajoute aussi le nom de fichier dans les noms des éléments de la liste
  names(multibiches)[i] <- nom

}

summary(multibiches)

```

```

##           Length Class      Mode
## biche10  9      data.frame list
## biche1   9      data.frame list
## biche2   9      data.frame list
## biche3   9      data.frame list
## biche4   9      data.frame list
## biche5   9      data.frame list
## biche6   9      data.frame list
## biche7   9      data.frame list
## biche8   9      data.frame list
## biche9   9      data.frame list

```

```
str(multibiches)
```

```

## List of 10
## $ biche10:'data.frame': 443 obs. of  9 variables:
## ..$ id      : int [1:443] 3983 3984 3985 3986 3987 3988 3989 3990 3991 3992 ...
## ..$ date    : Date[1:443], format: "2012-02-05" ...
## ..$ time    : POSIXlt[1:443], format: "2012-02-05 08:00:11" ...
## ..$ x       : int [1:443] 286370 286374 286373 286368 286374 286374 286374 286379 286371 286302 ...
## ..$ y       : int [1:443] 131783 131785 131778 131777 131777 131779 131775 131772 131772 132090 ...
## ..$ lat     : num [1:443] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long    : num [1:443] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period  : chr [1:443] "day" "day" "day" "day" ...
## ..$ file    : chr [1:443] "biche10" "biche10" "biche10" "biche10" ...
## $ biche1:'data.frame': 443 obs. of  9 variables:
## ..$ id      : int [1:443] 0 1 2 3 4 5 6 7 8 9 ...
## ..$ date    : Date[1:443], format: "2011-02-01" ...
## ..$ time    : POSIXlt[1:443], format: "2011-02-01 20:13:23" ...

```

```

## ..$ x      : int [1:443] 286202 286340 286586 286581 286323 286315 286314 286399 286407 286401 ...
## ..$ y      : int [1:443] 132124 132181 132035 132024 132248 132246 132249 132183 132184 132181 ...
## ..$ lat    : num [1:443] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long   : num [1:443] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period: chr [1:443] "night" "night" "day" "day" ...
## ..$ file   : chr [1:443] "biche1" "biche1" "biche1" "biche1" ...
## $ biche2 : 'data.frame': 443 obs. of  9 variables:
## ..$ id     : int [1:443] 443 444 445 446 447 448 449 450 451 452 ...
## ..$ date   : Date[1:443], format: "2011-02-10" ...
## ..$ time   : POSIXlt[1:443], format: "2011-02-10 09:45:53" ...
## ..$ x      : int [1:443] 286130 286121 286121 286111 286121 286119 286115 286117 286164 286203 ...
## ..$ y      : int [1:443] 132470 132475 132467 132493 132468 132478 132459 132462 132466 132491 ...
## ..$ lat    : num [1:443] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long   : num [1:443] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period: chr [1:443] "day" "day" "day" "day" ...
## ..$ file   : chr [1:443] "biche2" "biche2" "biche2" "biche2" ...
## $ biche3 : 'data.frame': 442 obs. of  9 variables:
## ..$ id     : int [1:442] 886 887 888 889 890 891 892 893 894 895 ...
## ..$ date   : Date[1:442], format: "2011-02-23" ...
## ..$ time   : POSIXlt[1:442], format: "2011-02-23 09:46:22" ...
## ..$ x      : int [1:442] 285677 285669 285657 285651 285666 285697 285672 285674 285704 285703 ...
## ..$ y      : int [1:442] 133137 133134 133138 133136 133125 133117 133132 133072 133044 133044 ...
## ..$ lat    : num [1:442] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long   : num [1:442] 6.28 6.28 6.28 6.28 6.28 ...
## ..$ period: chr [1:442] "day" "day" "day" "day" ...
## ..$ file   : chr [1:442] "biche3" "biche3" "biche3" "biche3" ...
## $ biche4 : 'data.frame': 443 obs. of  9 variables:
## ..$ id     : int [1:443] 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 ...
## ..$ date   : Date[1:443], format: "2011-04-19" ...
## ..$ time   : POSIXlt[1:443], format: "2011-04-19 08:00:38" ...
## ..$ x      : int [1:443] 286102 286288 286202 285875 285886 285880 285864 285873 285872 285867 ...
## ..$ y      : int [1:443] 131787 131767 131731 131646 131644 131663 131642 131638 131650 131670 ...
## ..$ lat    : num [1:443] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long   : num [1:443] 6.29 6.29 6.29 6.28 6.28 ...
## ..$ period: chr [1:443] "day" "day" "day" "day" ...
## ..$ file   : chr [1:443] "biche4" "biche4" "biche4" "biche4" ...
## $ biche5 : 'data.frame': 442 obs. of  9 variables:
## ..$ id     : int [1:442] 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 ...
## ..$ date   : Date[1:442], format: "2011-05-22" ...
## ..$ time   : POSIXlt[1:442], format: "2011-05-22 18:15:22" ...
## ..$ x      : int [1:442] 285944 285874 285760 285769 285790 285776 285769 285766 285763 285803 ...
## ..$ y      : int [1:442] 132468 132441 132249 132239 132242 132313 132309 132318 132317 132240 ...
## ..$ lat    : num [1:442] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long   : num [1:442] 6.28 6.28 6.28 6.28 6.28 ...
## ..$ period: chr [1:442] "day" "day" "night" "night" ...
## ..$ file   : chr [1:442] "biche5" "biche5" "biche5" "biche5" ...
## $ biche6 : 'data.frame': 443 obs. of  9 variables:
## ..$ id     : int [1:443] 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 ...
## ..$ date   : Date[1:443], format: "2011-07-26" ...
## ..$ time   : POSIXlt[1:443], format: "2011-07-26 12:15:40" ...
## ..$ x      : int [1:443] 286338 286343 286296 286191 286181 286177 285909 285832 285829 285831 ...
## ..$ y      : int [1:443] 133069 133040 133045 133027 133040 133047 132939 132796 132803 132829 ...
## ..$ lat    : num [1:443] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long   : num [1:443] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period: chr [1:443] "day" "day" "day" "day" ...
## ..$ file   : chr [1:443] "biche6" "biche6" "biche6" "biche6" ...
## $ biche7 : 'data.frame': 442 obs. of  9 variables:
## ..$ id     : int [1:442] 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 ...
## ..$ date   : Date[1:442], format: "2011-10-15" ...
## ..$ time   : POSIXlt[1:442], format: "2011-10-15 18:30:23" ...
## ..$ x      : int [1:442] 286333 286320 286320 286317 286306 286302 286315 286315 286316 286327 ...
## ..$ y      : int [1:442] 132526 132507 132496 132468 132473 132478 132469 132469 132474 132461 ...
## ..$ lat    : num [1:442] 50.5 50.5 50.5 50.5 50.5 ...

```



```
## ..$ long : num [1:442] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period: chr [1:442] "night" "night" "night" "night" ...
## ..$ file : chr [1:442] "biche7" "biche7" "biche7" "biche7" ...
## $ biche8 : 'data.frame': 443 obs. of 9 variables:
## ..$ id : int [1:443] 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 ...
## ..$ date : Date[1:443], format: "2011-11-02" ...
## ..$ time : POSIXlt[1:443], format: "2011-11-02 17:00:55" ...
## ..$ x : int [1:443] 286202 286252 286340 286382 286410 286404 286434 286462 286451 286448 ...
## ..$ y : int [1:443] 132363 132360 132339 132319 132325 132326 132362 132383 132387 132393 ...
## ..$ lat : num [1:443] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long : num [1:443] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period: chr [1:443] "night" "night" "night" "night" ...
## ..$ file : chr [1:443] "biche8" "biche8" "biche8" "biche8" ...
## $ biche9 : 'data.frame': 442 obs. of 9 variables:
## ..$ id : int [1:442] 3541 3542 3543 3544 3545 3546 3547 3548 3549 3550 ...
## ..$ date : Date[1:442], format: "2011-12-15" ...
## ..$ time : POSIXlt[1:442], format: "2011-12-15 18:30:52" ...
## ..$ x : int [1:442] 286256 286246 286255 286240 286244 286238 286239 286237 286239 286238 ...
## ..$ y : int [1:442] 132513 132511 132499 132510 132507 132516 132514 132515 132514 132510 ...
## ..$ lat : num [1:442] 50.5 50.5 50.5 50.5 50.5 ...
## ..$ long : num [1:442] 6.29 6.29 6.29 6.29 6.29 ...
## ..$ period: chr [1:442] "night" "night" "night" "night" ...
## ..$ file : chr [1:442] "biche9" "biche9" "biche9" "biche9" ...
```

On peut coller les jeux de données les uns à la suite des autres

```
res <- rbind(multibiches[[1]], multibiches[[2]], multibiches[[3]]) # 3 fichiers
res <- do.call(rbind, multibiches) # tous les fichiers
summary(res)
```

```
##           id           date           time
## Min.      : 0   Min. :2011-02-01   Min. :2011-02-01 20:13:23
## 1st Qu.:1106   1st Qu.:2011-03-16   1st Qu.:2011-03-16 15:04:42
## Median :2212   Median :2011-07-26   Median :2011-07-26 12:09:14
## Mean    :2212   Mean    :2011-07-29   Mean    :2011-07-30 00:14:44
## 3rd Qu.:3319   3rd Qu.:2011-11-17   3rd Qu.:2011-11-17 00:26:31
## Max.    :4425   Max.    :2012-03-29   Max.    :2012-03-29 04:00:55
##           x           y           lat           long
## Min.    :284406   Min.    :128157   Min.    :50.45   Min.    :6.262
## 1st Qu.:285488   1st Qu.:132342   1st Qu.:50.49   1st Qu.:6.278
## Median :285975   Median :132588   Median :50.49   Median :6.285
## Mean    :285882   Mean    :132540   Mean    :50.49   Mean    :6.284
## 3rd Qu.:286293   3rd Qu.:132905   3rd Qu.:50.49   3rd Qu.:6.290
## Max.    :287348   Max.    :133691   Max.    :50.50   Max.    :6.304
##           period           file
## Length:4426           Length:4426
## Class :character      Class :character
## Mode :character       Mode :character
##
##
##
```

```
head(res)
```

```
##           id           date           time           x           y           lat
## biche10.1 3983 2012-02-05 2012-02-05 08:00:11 286370 131783 50.48096
## biche10.2 3984 2012-02-05 2012-02-05 08:15:48 286374 131785 50.48098
## biche10.3 3985 2012-02-05 2012-02-05 08:30:21 286373 131778 50.48092
## biche10.4 3986 2012-02-05 2012-02-05 08:45:42 286368 131777 50.48091
## biche10.5 3987 2012-02-05 2012-02-05 09:00:11 286374 131777 50.48092
## biche10.6 3988 2012-02-05 2012-02-05 09:15:19 286374 131779 50.48094
```

```
##          long period    file
## biche10.1 6.290375    day biche10
## biche10.2 6.290439    day biche10
## biche10.3 6.290414    day biche10
## biche10.4 6.290350    day biche10
## biche10.5 6.290432    day biche10
## biche10.6 6.290434    day biche10
```

Agrégation et reshaping

NB1 : dans ces exercices, on utilisera jamais de boucle explicite.

NB2 : par “créer une fonction à la volée”, on entend créer une fonction à usage unique qui n’est pas sauvée dans un objet. On utilise typiquement ce genre de fonctions dans les fonctions de la famille apply.

Exercice 1

Chargez le jeu de données “ladybirds.txt”. On a déjà utilisé à plusieurs reprises un jeu de données similaire. Les 4 premières colonnes contiennent : l’espèce d’arbre (tree : Pine, Mapple ou Lime), le paysage (landsc : Urban, Suburban, Rural), un code identifiant le site (site) et la période d’échantillonnage (date). Les colonnes suivantes contiennent des abondances de coccinelles identifiées par un code à 6 lettres.

```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
d <- read.table("data/ladybirds.txt", header=TRUE, sep = "\t")
summary(d)
```

```
## tree   landsc   site   date   haraxy   adabip
## L:27   R:27    LR1    : 3    date1: 9    Min.    : 0.00    Min.    : 0.00
## M:27   S:27    LR2    : 3    date2:27   1st Qu.: 0.00    1st Qu.: 1.00
## P:27   U:27    LR3    : 3    date3:27   Median : 0.00    Median : 4.00
##                               LS1    : 3    date4:18   Mean    : 2.21    Mean    : 13.51
##                               LS2    : 3                3rd Qu.: 2.00    3rd Qu.: 12.00
##                               LS3    : 3                Max.    :30.00    Max.    :212.00
##                               (Other):63
##      adadec      calqua      caldec      exoqua
## Min.    : 0.000    Min.    : 0.0000    Min.    : 0.000    Min.    : 0
## 1st Qu.: 0.000    1st Qu.: 0.0000    1st Qu.: 0.000    1st Qu.: 0
## Median : 1.000    Median : 0.0000    Median : 0.000    Median : 0
## Mean    : 3.778    Mean    : 0.9506    Mean    : 1.136    Mean    : 1
## 3rd Qu.: 3.000    3rd Qu.: 2.0000    3rd Qu.: 2.000    3rd Qu.: 1
## Max.    :82.000    Max.    :13.0000    Max.    :14.000    Max.    :15
##
##      halsed      oencon      harqua      myroct
## Min.    :0.0000    Min.    : 0.000    Min.    : 0.000    Min.    : 0.000
## 1st Qu.:0.0000    1st Qu.: 0.000    1st Qu.: 0.000    1st Qu.: 0.000
## Median :0.0000    Median : 0.000    Median : 0.000    Median : 0.000
## Mean    :0.2593    Mean    : 1.049    Mean    : 7.741    Mean    : 3.951
## 3rd Qu.:0.0000    3rd Qu.: 1.000    3rd Qu.: 10.000    3rd Qu.: 2.000
## Max.    :2.0000    Max.    :11.000    Max.    :106.000    Max.    :54.000
##
##      aphobl      anaoce
## Min.    : 0.000    Min.    :0.0000
## 1st Qu.: 0.000    1st Qu.:0.0000
## Median : 0.000    Median :0.0000
## Mean    : 2.074    Mean    :0.2469
## 3rd Qu.: 0.000    3rd Qu.:0.0000
## Max.    :29.000    Max.    :5.0000
##
```

- 1) faites le total du nombre d’individus pour chaque espèce et par site. Utilisez la fonction aggregate et sauvez le résultat dans un dataset appelé “site”. Gardez dans ce jeu de données les colonnes tree, landsc et site dont vous aurez besoin dans les exercices suivants.

```
# Deux notations possibles pour aggregate avec des résultats identiques :
aggregate(.~tree + landsc + site, data=d[, -4], FUN = sum)
```

```
## tree landsc site haraxy adabip adadec calqua caldec exoqua halsed
## 1    L    R LR1    0      1      2      3      1      0      3
```

```
## 2    L    R LR2    0    13    13    4    6    0    0
## 3    L    R LR3    0    21    10    3    5    13    1
## 4    L    S LS1    4    49    13    6    15    1    0
## 5    L    S LS2    4    7    10    5    24    1    1
## 6    L    S LS3    0    4    2    4    6    0    0
## 7    L    U LU1    5    31    11    0    9    1    1
## 8    L    U LU2    17   165    3    6    0    15    0
## 9    L    U LU3    15    33    24    6    4    6    1
## 10   M    R MR1    0    3    1    3    0    0    0
## 11   M    R MR2    0    7    1    6    1    4    1
## 12   M    R MR3    2    26    5    14    2    1    0
## 13   M    S MS1    4    21    2    1    2    1    1
## 14   M    S MS2    2    3    0    3    10    1    4
## 15   M    S MS3    0    10    3    3    4    3    3
## 16   M    U MU1    18    60    5    2    2    8    0
## 17   M    U MU2    54    48    4    4    1    11    0
## 18   M    U MU3    3    7    0    2    0    0    1
## 19   P    R PR1    0    6    0    0    0    5    0
## 20   P    R PR2    0    11    1    2    0    0    2
## 21   P    R PR3    0    4    10    0    0    1    0
## 22   P    S PS1    0    7    6    0    0    0    1
## 23   P    S PS2    0    10    2    0    0    4    1
## 24   P    S PS3    0    8    8    0    0    3    0
## 25   P    U PU1    21    87    14    0    0    0    0
## 26   P    U PU2    28   224    64    0    0    2    0
## 27   P    U PU3    2    228    92    0    0    0    0
##      oencon harqua myroct aphobl anaoce
## 1      0      0      0      0      0
## 2      0      0      0      0      0
## 3      0      0      0      0      0
## 4      6      0      0      0      0
## 5      0      0      0      0      0
## 6      0      0      0      0      0
## 7      6      0      0      0      0
## 8      8      0      0      0      0
## 9      2      0      0      0      0
## 10     0      0      0      0      0
## 11     0      0      0      0      0
## 12     0      0      0      0      0
## 13     3      0      0      0      0
## 14     0      0      0      1      0
## 15     0      0      0      0      0
## 16    27      0      0      0      0
## 17    14      1      0      0      1
## 18     5      0      0      0      0
## 19     0     67      3     17      5
## 20     0     53     17      4      1
## 21     0     56      0     23      2
## 22     0     37     73     24      3
## 23     0    157    139     54      0
## 24     0    102     29     33      8
## 25    14     50     17      2      0
## 26     0     32     20     10      0
## 27     0     72     22      0      0
```

```
site <- aggregate(d[,5:ncol(d)], d[, c("tree", "landsc", "site")], FUN = sum)
```

- 2) Au moyen de la fonction apply, calculez le nombre moyen et l'écart-type de chaque espèce de coccinelles entre les sites (autrement dit : la moyenne et l'écart-type des colonnes espèces du dataset "site").
NB : on pourrait utiliser ici la fonction colMeans pour la moyenne mais il n'y a pas d'équivalent pour l'écart-type.
haraxy : moyenne = 6.62, sd = 12.22

```
apply(site[, 4:ncol(site)], 2, mean)
```

```
##      haraxy      adabip      adadec      calqua      caldec      exoqua
## 6.6296296 40.5185185 11.3333333  2.8518519  3.4074074  3.0000000
##      halsed      oencon      harqua      myroct      aphobl      anaoce
## 0.7777778  3.1481481 23.2222222 11.8518519  6.2222222  0.7407407
```

```
apply(site[, 4:ncol(site)], 2, sd)
```

```
##      haraxy      adabip      adadec      calqua      caldec      exoqua      halsed
## 12.225912 63.662796 20.405505  3.109584  5.590234  4.197069  1.086042
##      oencon      harqua      myroct      aphobl      anaoce
## 6.286149 39.712751 29.851818 13.118787  1.852080
```

- 3) Utilisez les fonctions `lapply` et puis `sapply` pour obtenir le même résultat que la fonction précédente (la présentation peut être différente mais pas les chiffres). Le jeu de donnée étant un `data.frame`, on peut le traiter comme une liste dont les éléments sont les colonnes du `data.frame`, ce qui permet d'utiliser les fonctions `sapply` et `lapply`.

```
lapply(site[, 4:ncol(site)], mean)
```

```
## $haraxy
## [1] 6.62963
##
## $adabip
## [1] 40.51852
##
## $adadec
## [1] 11.33333
##
## $calqua
## [1] 2.851852
##
## $caldec
## [1] 3.407407
##
## $exoqua
## [1] 3
##
## $halsed
## [1] 0.7777778
##
## $oencon
## [1] 3.148148
##
## $harqua
## [1] 23.22222
##
## $myroct
## [1] 11.85185
##
## $aphobl
## [1] 6.222222
##
## $anaoce
## [1] 0.7407407
```

```
sapply(site[, 4:ncol(site)], mean)
```

```
##      haraxy      adabip      adadec      calqua      caldec      exoqua
## 6.6296296 40.5185185 11.3333333 2.8518519 3.4074074 3.0000000
##      halsed      oencon      harqua      myroct      aphobl      anaoce
## 0.7777778 3.1481481 23.2222222 11.8518519 6.2222222 0.7407407
```

- 4) Au moyen de la fonction `apply`, calculez le nombre total de coccinelles (toutes espèces confondues) pour chaque site. (on veut donc faire la somme des lignes du dataset “site”)
NB1 : on devrait de préférence utiliser ici la fonction `rowSums`
NB2 : impossible ici d’utiliser `sapply` ou `lapply` car on travaille sur les lignes

```
apply(site[, 4:ncol(site)], 1, sum)
```

```
## [1] 10 36 53 94 52 16 64 214 91 7 20 50 35 24 26 122 138
## [18] 18 103 91 96 151 367 191 205 380 416
```

- 5) Ecrivez une fonction qui calcule l’indice de diversité de Simpson d’un site sur base du nombre d’individus de chaque espèce. Cet indice se calcule comme 1 moins la somme du carré des abondances relatives de chaque espèce. Calculez ensuite cet indice de simpson pour chaque site. (indice pour le site LR1 = 0.76)

```
simpson <- function(x) { 1-sum((x/sum(x))^2) }
apply(site[, 4:ncol(site)], 1, simpson)
```

```
## [1] 0.7600000 0.6990741 0.7347811 0.6736080 0.7159763 0.7187500 0.7006836
## [8] 0.3919120 0.7605362 0.6122449 0.7400000 0.6376000 0.6106122 0.7569444
## [15] 0.7751479 0.6808654 0.7074144 0.7283951 0.5406730 0.6081391 0.5891927
## [22] 0.6768124 0.6509960 0.6563965 0.7336109 0.6081440 0.6179271
```

- 6) En partant du dataset “site”, calculez l’abondance moyenne et l’écart type pour chaque combinaison d’arbre et de paysage.
Vous devrez utiliser la fonction `aggregate`. Le résultat est un tableau à 9 lignes et 14 colonnes.

```
aggregate(site[,4:ncol(site)], site[, c("tree","landsc")], FUN = mean)
```

```
##      tree landsc      haraxy      adabip      adadec      calqua      caldec
## 1      L      R 0.0000000 11.666667 8.333333 3.333333 4.000000
## 2      M      R 0.6666667 12.000000 2.333333 7.666667 1.000000
## 3      P      R 0.0000000 7.000000 3.666667 0.666667 0.000000
## 4      L      S 2.6666667 20.000000 8.333333 5.000000 15.000000
## 5      M      S 2.0000000 11.333333 1.666667 2.333333 5.333333
## 6      P      S 0.0000000 8.333333 5.333333 0.000000 0.000000
## 7      L      U 12.333333 76.333333 12.666667 4.000000 4.333333
## 8      M      U 25.000000 38.333333 3.000000 2.666667 1.000000
## 9      P      U 17.000000 179.666667 56.666667 0.000000 0.000000
##      exoqua      halsed      oencon      harqua      myroct      aphobl      anaoce
## 1 4.3333333 1.3333333 0.000000 0.000000 0.000000 0.000000 0.000000
## 2 1.6666667 0.3333333 0.000000 0.000000 0.000000 0.000000 0.000000
## 3 2.0000000 0.6666667 0.000000 58.666667 6.666667 14.666667 2.666667
## 4 0.6666667 0.3333333 2.000000 0.000000 0.000000 0.000000 0.000000
## 5 1.6666667 2.6666667 1.000000 0.000000 0.000000 0.3333333 0.000000
## 6 2.3333333 0.6666667 0.000000 98.666667 80.333333 37.000000 3.666667
## 7 7.3333333 0.6666667 5.333333 0.000000 0.000000 0.000000 0.000000
## 8 6.3333333 0.3333333 15.333333 0.333333 0.000000 0.000000 0.333333
## 9 0.6666667 0.000000 4.666667 51.333333 19.666667 4.000000 0.000000
```

```
aggregate(site[,4:ncol(site)], site[, c("tree","landsc")], FUN = sd)
```

```
##   tree landsc   haraxy   adabip   adadec   calqua   caldec   exoqua
## 1    L      R 0.000000 10.066446 5.686241 0.5773503 2.645751 7.5055535
## 2    M      R 1.154701 12.288206 2.309401 5.6862407 1.000000 2.0816660
## 3    P      R 0.000000 3.605551 5.507571 1.1547005 0.000000 2.6457513
## 4    L      S 2.309401 25.159491 5.686241 1.0000000 9.000000 0.5773503
## 5    M      S 2.000000 9.073772 1.527525 1.1547005 4.163332 1.1547005
## 6    P      S 0.000000 1.527525 3.055050 0.0000000 0.000000 2.0816660
## 7    L      U 6.429101 76.794097 10.598742 3.4641016 4.509250 7.0945989
## 8    M      U 26.210685 27.790886 2.645751 1.1547005 1.000000 5.6862407
## 9    P      U 13.453624 80.276605 39.513711 0.0000000 0.000000 1.1547005
##   halsed   oencon   harqua   myroct   aphobl   anaoce
## 1 1.5275252 0.000000 0.0000000 0.000000 0.0000000 0.0000000
## 2 0.5773503 0.000000 0.0000000 0.000000 0.0000000 0.0000000
## 3 1.1547005 0.000000 7.3711148 9.073772 9.7125349 2.0816660
## 4 0.5773503 3.464102 0.0000000 0.000000 0.0000000 0.0000000
## 5 1.5275252 1.732051 0.0000000 0.000000 0.5773503 0.0000000
## 6 0.5773503 0.000000 60.0694043 55.365453 15.3948043 4.0414519
## 7 0.5773503 3.055050 0.0000000 0.000000 0.0000000 0.0000000
## 8 0.5773503 11.060440 0.5773503 0.000000 0.0000000 0.5773503
## 9 0.0000000 8.082904 20.0333056 2.516611 5.2915026 0.0000000
```

- 7) Faites la même opération que dans la question précédente mais avec un résultat arrondi à 2 décimales (vous pouvez le faire uniquement pour la moyenne). Essayez de répondre à cette question en utilisant les 2 approches différentes suivantes :
 - Appliquez simplement la fonction round au résultat de la fonction aggregate de la question précédente (moins les 2 premières colonnes).
 - Au lieu d'utiliser la fonction mean seule dans l'aggregate, créez à la volée une fonction qui calcule la moyenne et l'arrondit à 2 décimales.

Solution 1 :

```
a <- aggregate(site[,4:ncol(site)], site[, c("tree","landsc")], FUN = mean)
round(a[, -c(1,2)], 2)
```

```
##   haraxy adabip adadec calqua caldec exoqua halsed oencon harqua myroct
## 1    0.00 11.67  8.33  3.33  4.00  4.33  1.33  0.00  0.00  0.00
## 2    0.67 12.00  2.33  7.67  1.00  1.67  0.33  0.00  0.00  0.00
## 3    0.00 7.00  3.67  0.67  0.00  2.00  0.67  0.00 58.67  6.67
## 4    2.67 20.00  8.33  5.00 15.00  0.67  0.33  2.00  0.00  0.00
## 5    2.00 11.33  1.67  2.33  5.33  1.67  2.67  1.00  0.00  0.00
## 6    0.00 8.33  5.33  0.00  0.00  2.33  0.67  0.00 98.67 80.33
## 7   12.33 76.33 12.67  4.00  4.33  7.33  0.67  5.33  0.00  0.00
## 8   25.00 38.33  3.00  2.67  1.00  6.33  0.33 15.33  0.33  0.00
## 9   17.00 179.67 56.67  0.00  0.00  0.67  0.00  4.67 51.33 19.67
##   aphobl anaoce
## 1    0.00  0.00
## 2    0.00  0.00
## 3   14.67  2.67
## 4    0.00  0.00
## 5    0.33  0.00
## 6   37.00  3.67
## 7    0.00  0.00
## 8    0.00  0.33
## 9    4.00  0.00
```

Solution 2 :

```
aggregate(site[,4:ncol(site)], site[, c("tree","landsc")], FUN = function(x) {round(mean(x), 2)})
```

```
##   tree landsc haraxy adabip adadec calqua caldec exoqua halsed oencon
## 1    L      R   0.00  11.67   8.33   3.33   4.00   4.33   1.33   0.00
## 2    M      R   0.67  12.00   2.33   7.67   1.00   1.67   0.33   0.00
## 3    P      R   0.00   7.00   3.67   0.67   0.00   2.00   0.67   0.00
## 4    L      S   2.67  20.00   8.33   5.00  15.00   0.67   0.33   2.00
## 5    M      S   2.00  11.33   1.67   2.33   5.33   1.67   2.67   1.00
## 6    P      S   0.00   8.33   5.33   0.00   0.00   2.33   0.67   0.00
## 7    L      U  12.33  76.33  12.67   4.00   4.33   7.33   0.67   5.33
## 8    M      U  25.00  38.33   3.00   2.67   1.00   6.33   0.33  15.33
## 9    P      U  17.00 179.67  56.67   0.00   0.00   0.67   0.00   4.67
##   harqua myroct aphobl anaoce
## 1     0.00   0.00   0.00   0.00
## 2     0.00   0.00   0.00   0.00
## 3    58.67   6.67  14.67   2.67
## 4     0.00   0.00   0.00   0.00
## 5     0.00   0.00   0.33   0.00
## 6    98.67  80.33  37.00   3.67
## 7     0.00   0.00   0.00   0.00
## 8     0.33   0.00   0.00   0.33
## 9    51.33  19.67   4.00   0.00
```

- 8) Si vous le désirez, explorez les possibilités de la fonction cast du package reshape. Cette fonction permet de faire tout ce que aggregate et tapply font mais avec une syntaxe beaucoup plus simple. Vous devrez commencer par passer le jeu de données en format “long” avec la fonction melt()

```
library(reshape)
md <- melt(d, id = 1:4, variable_name = "sp")
cast(md, tree + landsc ~ sp, fun = sum)
```

```
##   tree landsc haraxy adabip adadec calqua caldec exoqua halsed oencon
## 1    L      R     0     35     25     10     12     13     4     0
## 2    L      S     8     60     25     15     45     2     1     6
## 3    L      U    37    229     38     12     13     22     2    16
## 4    M      R     2     36     7     23     3     5     1     0
## 5    M      S     6     34     5     7     16     5     8     3
## 6    M      U    75    115     9     8     3     19     1    46
## 7    P      R     0     21    11     2     0     6     2     0
## 8    P      S     0     25    16     0     0     7     2     0
## 9    P      U    51    539    170     0     0     2     0    14
##   harqua myroct aphobl anaoce
## 1      0      0      0      0
## 2      0      0      0      0
## 3      0      0      0      0
## 4      0      0      0      0
## 5      0      0      1      0
## 6      1      0      0      1
## 7    176     20     44     8
## 8    296    241    111    11
## 9    154     59     12     0
```

```
cast(md, tree + landsc + site ~ sp, fun = sum)
```

```
##   tree landsc site haraxy adabip adadec calqua caldec exoqua halsed
## 1    L      R LR1      0      1      2      3      1      0      3
## 2    L      R LR2      0     13     13     4      6      0      0
```


## 3	L	R	LR3	0	21	10	3	5	13	1
## 4	L	S	LS1	4	49	13	6	15	1	0
## 5	L	S	LS2	4	7	10	5	24	1	1
## 6	L	S	LS3	0	4	2	4	6	0	0
## 7	L	U	LU1	5	31	11	0	9	1	1
## 8	L	U	LU2	17	165	3	6	0	15	0
## 9	L	U	LU3	15	33	24	6	4	6	1
## 10	M	R	MR1	0	3	1	3	0	0	0
## 11	M	R	MR2	0	7	1	6	1	4	1
## 12	M	R	MR3	2	26	5	14	2	1	0
## 13	M	S	MS1	4	21	2	1	2	1	1
## 14	M	S	MS2	2	3	0	3	10	1	4
## 15	M	S	MS3	0	10	3	3	4	3	3
## 16	M	U	MU1	18	60	5	2	2	8	0
## 17	M	U	MU2	54	48	4	4	1	11	0
## 18	M	U	MU3	3	7	0	2	0	0	1
## 19	P	R	PR1	0	6	0	0	0	5	0
## 20	P	R	PR2	0	11	1	2	0	0	2
## 21	P	R	PR3	0	4	10	0	0	1	0
## 22	P	S	PS1	0	7	6	0	0	0	1
## 23	P	S	PS2	0	10	2	0	0	4	1
## 24	P	S	PS3	0	8	8	0	0	3	0
## 25	P	U	PU1	21	87	14	0	0	0	0
## 26	P	U	PU2	28	224	64	0	0	2	0
## 27	P	U	PU3	2	228	92	0	0	0	0
##	oencon	harqua	myroct	aphobl	anaoce					
## 1	0	0	0	0	0					
## 2	0	0	0	0	0					
## 3	0	0	0	0	0					
## 4	6	0	0	0	0					
## 5	0	0	0	0	0					
## 6	0	0	0	0	0					
## 7	6	0	0	0	0					
## 8	8	0	0	0	0					
## 9	2	0	0	0	0					
## 10	0	0	0	0	0					
## 11	0	0	0	0	0					
## 12	0	0	0	0	0					
## 13	3	0	0	0	0					
## 14	0	0	0	1	0					
## 15	0	0	0	0	0					
## 16	27	0	0	0	0					
## 17	14	1	0	0	1					
## 18	5	0	0	0	0					
## 19	0	67	3	17	5					
## 20	0	53	17	4	1					
## 21	0	56	0	23	2					
## 22	0	37	73	24	3					
## 23	0	157	139	54	0					
## 24	0	102	29	33	8					
## 25	14	50	17	2	0					
## 26	0	32	20	10	0					
## 27	0	72	22	0	0					

```
cast(md, sp + landsc ~ date, fun = sum)
```

##	sp	landsc	date1	date2	date3	date4
## 1	haraxy	R	0	1	0	1
## 2	haraxy	S	0	1	5	8
## 3	haraxy	U	40	27	51	45
## 4	adabip	R	16	51	13	12
## 5	adabip	S	12	40	33	34

```
## 6  adabip      U   463   187   130   103
## 7  adadec      R     4    20    15     4
## 8  adadec      S    15    13    14     4
## 9  adadec      U   147    38    21    11
## 10 calqua      R     2    25     2     6
## 11 calqua      S     0    13     5     4
## 12 calqua      U     0    10     4     6
## 13 caldec      R     0     1    11     3
## 14 caldec      S     0    12    34    15
## 15 caldec      U     0     4     8     4
## 16 exoqua      R     1     4    12     7
## 17 exoqua      S     4     4     4     2
## 18 exoqua      U     2     3     3    35
## 19 halsed      R     2     0     1     4
## 20 halsed      S     2     5     4     0
## 21 halsed      U     0     1     2     0
## 22 oencon      R     0     0     0     0
## 23 oencon      S     0     1     4     4
## 24 oencon      U     9    32    17    18
## 25 harqua      R    82    44    50     0
## 26 harqua      S   162    69    65     0
## 27 harqua      U   124    23     7     1
## 28 myroct      R     3     6    11     0
## 29 myroct      S    85    71    85     0
## 30 myroct      U    29     9    21     0
## 31 aphobl      R    21     3    20     0
## 32 aphobl      S    45    19    48     0
## 33 aphobl      U     2    10     0     0
## 34 anaoce      R     3     0     5     0
## 35 anaoce      S     3     3     5     0
## 36 anaoce      U     0     0     0     1
```

```
cast(md, landsc ~ date | sp, fun = sum)
```

```
## $haraxy
##   landsc date1 date2 date3 date4
## 1      R     0     1     0     1
## 2      S     0     1     5     8
## 3      U    40    27    51    45
##
## $adabip
##   landsc date1 date2 date3 date4
## 1      R    16    51    13    12
## 2      S    12    40    33    34
## 3      U   463   187   130   103
##
## $adadec
##   landsc date1 date2 date3 date4
## 1      R     4    20    15     4
## 2      S    15    13    14     4
## 3      U   147    38    21    11
##
## $calqua
##   landsc date1 date2 date3 date4
## 1      R     2    25     2     6
## 2      S     0    13     5     4
## 3      U     0    10     4     6
##
## $caldec
##   landsc date1 date2 date3 date4
## 1      R     0     1    11     3
## 2      S     0    12    34    15
```

```

## 3      U      0      4      8      4
##
## $exoqua
##   landsc date1 date2 date3 date4
## 1      R      1      4     12      7
## 2      S      4      4      4      2
## 3      U      2      3      3     35
##
## $halsed
##   landsc date1 date2 date3 date4
## 1      R      2      0      1      4
## 2      S      2      5      4      0
## 3      U      0      1      2      0
##
## $oencon
##   landsc date1 date2 date3 date4
## 1      R      0      0      0      0
## 2      S      0      1      4      4
## 3      U      9     32     17     18
##
## $harqua
##   landsc date1 date2 date3 date4
## 1      R     82     44     50      0
## 2      S    162     69     65      0
## 3      U    124     23      7      1
##
## $myroct
##   landsc date1 date2 date3 date4
## 1      R      3      6     11      0
## 2      S     85     71     85      0
## 3      U     29      9     21      0
##
## $aphobl
##   landsc date1 date2 date3 date4
## 1      R     21      3     20      0
## 2      S     45     19     48      0
## 3      U      2     10      0      0
##
## $anaoce
##   landsc date1 date2 date3 date4
## 1      R      3      0      5      0
## 2      S      3      3      5      0
## 3      U      0      0      0      1

```

```
cast(md, sp ~ tree + landsc, fun = sum)
```

```

##      sp L_R L_S L_U M_R M_S M_U P_R P_S P_U
## 1 haraxy  0  8 37  2  6 75  0  0 51
## 2 adabip 35 60 229 36 34 115 21 25 539
## 3 adadec 25 25 38  7  5  9 11 16 170
## 4 calqua 10 15 12 23  7  8  2  0  0
## 5 caldec 12 45 13  3 16  3  0  0  0
## 6 exoqua 13  2 22  5  5 19  6  7  2
## 7 halsed  4  1  2  1  8  1  2  2  0
## 8 oencon  0  6 16  0  3 46  0  0 14
## 9 harqua  0  0  0  0  0  1 176 296 154
## 10 myroct 0  0  0  0  0  0 20 241 59
## 11 aphobl 0  0  0  0  1  0 44 111 12
## 12 anaoce 0  0  0  0  0  1  8 11  0

```

Exercice 2

Chargez le jeu de données “dff.csv” et visualisez sa structure. Il s’agit d’un jeu de données faunistiques (de 15000 lignes) typique des jeu de données récoltés par des naturalistes volontaires : espèces observée (sp), position géographique (utm1, utm5 : mailles de 1 et 5 km², x, y : coordonnées géographiques), date (dat2), la plante hôte (plante), le nombre d’individus (n) et diverses infos comme la méthode de capture (capt), le stade de développement (devl), le type de support (micr).

```
# On lit le jeu de données, on affiche un résumé et on visualise les 10 premières lignes
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
d <- read.table("data/dff.csv", header=TRUE, sep=";", quote="\")
summary(d)
```

```
##           sp           utm1           utm5           x
## sp_1 :1133 31UES0845: 236 31UES075475: 238 Min. : 22480
## sp_2 : 867 31UES2533: 162 31UES275325: 226 1st Qu.:114833
## sp_3 : 674 31UES7647: 78 31UES775475: 174 Median :161837
## sp_4 : 669 31UFS2002: 69 31UFS225025: 156 Mean :158206
## sp_5 : 519 31UFS2003: 61 31UES975325: 88 3rd Qu.:202747
## sp_6 : 376 31UES7547: 53 31UES975375: 86 Max. :290158
## (Other): 762 (Other) :4341 (Other) :4032 NA's :3
##           y           dat2           devl           capt
## Min. : 22807 Min. :19980408 0 : 109 0 : 1
## 1st Qu.:132327 1st Qu.:20020626 A :4452 Fauch: 384
## Median :169686 Median :20040826 L : 315 Filet: 9
## Mean :156216 Mean :20043617 P : 53 Obs :1673
## 3rd Qu.:184690 3rd Qu.:20060806 X : 5 Parap:1102
## Max. :241822 Max. :20111031 NA's: 66 PF : 46
## NA's :3 NA's :1785
##           plante           micr           n
## Urtica dioica : 398 Feuil : 151 Min. : 1.000
## Crataegus monogyna: 299 Fleur : 115 1st Qu.: 1.000
## Quercus robur : 277 Tronc : 18 Median : 1.000
## Pinus sylvestris : 209 RT : 13 Mean : 4.601
## Betula pendula : 181 Fruit : 11 3rd Qu.: 3.000
## Pinus nigra : 167 (Other): 6 Max. :1000.000
## (Other) :3469 NA's :4686
```

```
d[1:10,]
```

```
##           sp           utm1           utm5           x           y           dat2 devl capt
## 1 sp_3 31UDS6858 31UDS675575 22480.21 197685.4 20080911 <NA> <NA>
## 2 sp_2 31UDS6954 31UDS675525 23406.90 193666.2 20061014 L <NA>
## 3 sp_1 31UDS6955 31UDS675575 23425.26 194666.4 20061028 A <NA>
## 4 sp_1 31UDS6958 31UDS675575 23480.40 197667.0 20010713 A Obs
## 5 sp_2 31UDS6959 31UDS675575 23498.78 198667.2 20080911 <NA> <NA>
## 6 sp_7 31UDS6960 31UDS675625 23517.14 199667.4 20030925 A Fauch
## 7 sp_3 31UDS6960 31UDS675625 23517.14 199667.4 20050429 A Obs
## 8 sp_5 31UDS6960 31UDS675625 23517.14 199667.4 20030925 A Obs
## 9 sp_4 31UDS6960 31UDS675625 23517.14 199667.4 20050429 A Obs
## 10 sp_2 31UDS7057 31UDS725575 24462.23 196648.4 20080817 A <NA>
##           plante micr n
## 1 Urtica dioica <NA> 1
## 2 Sambucus nigra <NA> 4
## 3 Urtica dioica <NA> 1
## 4 Urtica dioica <NA> 1
## 5 Fraxinus excelsior <NA> 10
## 6 Acer pseudoplatanus <NA> 1
## 7 Ammophila arenaria <NA> 20
## 8 Cirsium vulgare <NA> 4
```

```
## 9  Ammophila arenaria <NA> 10
## 10 Artemisia vulgaris <NA> 1
```

- 1) Combien a-t-on d'observations par espèce ?
Essayez d'arriver au même résultat avec la fonction `table` (la plus simple dans ce cas) et `aggregate`. Pour `aggregate` utilisez la fonction "length" pour compter le nombre d'observations.
(réponse : `sp_01` : 1133 obs, etc)

```
table(d$sp)
```

```
##
## sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 1133 867 674 669 519 376 346 263 153
```

```
# même résultat avec aggregate (le premier argument peut être n'importe quelle colonne)
aggregate(d$sp, list(d$sp), length)
```

```
## Group.1 x
## 1 sp_1 1133
## 2 sp_2 867
## 3 sp_3 674
## 4 sp_4 669
## 5 sp_5 519
## 6 sp_6 376
## 7 sp_7 346
## 8 sp_8 263
## 9 sp_9 153
```

- 2) Combien a-t-on de données par espèce ? On définira la donnée comme une combinaison unique de `sp` x `utm1` x `dat2`.
Utilisez la fonction "unique" pour agréger les valeurs identiques suivie de `table`.
(réponse : `sp_01` : 946 données etc)

```
prov <- unique(d[,c("sp", "utm1", "dat2")])
table(prov$sp)
```

```
##
## sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 946 668 548 554 456 332 312 239 138
```

- 3) Quels sont les 30 carrés utm de 5 km² (`utm5`) avec le plus d'espèces ? (réponse : 31UER775775 et 16 autres carrés avec 9 espèces)

```
prov <- unique(d[,c("sp", "utm5")]) # combinaison unique de espèces x carrés
sort(table(prov$utm5), decreasing = TRUE) [1:30]
```

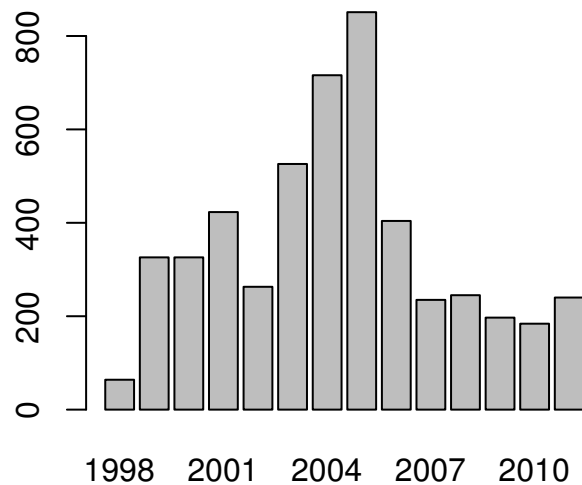
```
##
## 31UER775775 31UES475575 31UES625625 31UES775475 31UES925375 31UES975325
##          9          9          9          9          9          9
## 31UES975375 31UFR175525 31UFR225525 31UFR375675 31UFR475975 31UFS025325
##          9          9          9          9          9          9
## 31UFS225025 31UFS475725 31UFS725075 31UFS825125 31UGR025975 31UES425675
##          9          9          9          9          9          8
## 31UES475325 31UES725475 31UES975275 31UFR225975 31UFR275625 31UFS075525
##          8          8          8          8          8          8
## 31UFS175525 31UFS475525 31UFS525525 31UFS525775 31UFS675825 31UFS875325
##          8          8          8          8          8          8
```

- 4) Extrayez l'année, le mois et le jour de dat2 (utilisez simplement substring) et stockez les dans 3 nouvelles colonnes du jeu de données

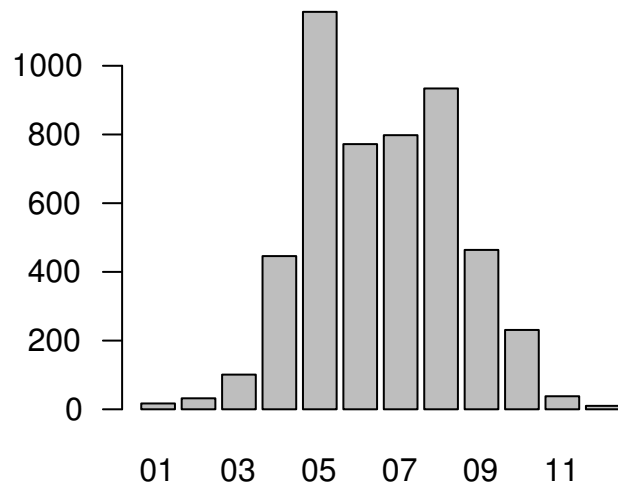
```
d$Y <- substring(d$dat2, 1,4)
d$M <- substring(d$dat2, 5,6)
d$D <- substring(d$dat2, 7,8)
```

- 5) Faites un graphique (par exemple barplot) de l'évolution du nombre total d'observations au cours des années (240 observations en 2011).
Faites de même un graphique montrant la phénologie de toutes les espèces confondues par mois (17 observations en Janvier).

```
barplot(table(d$Y))
```

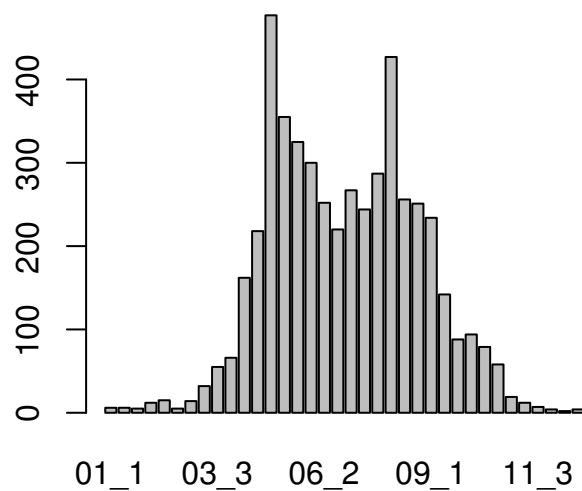


```
barplot(table(d$M), las=1)
```



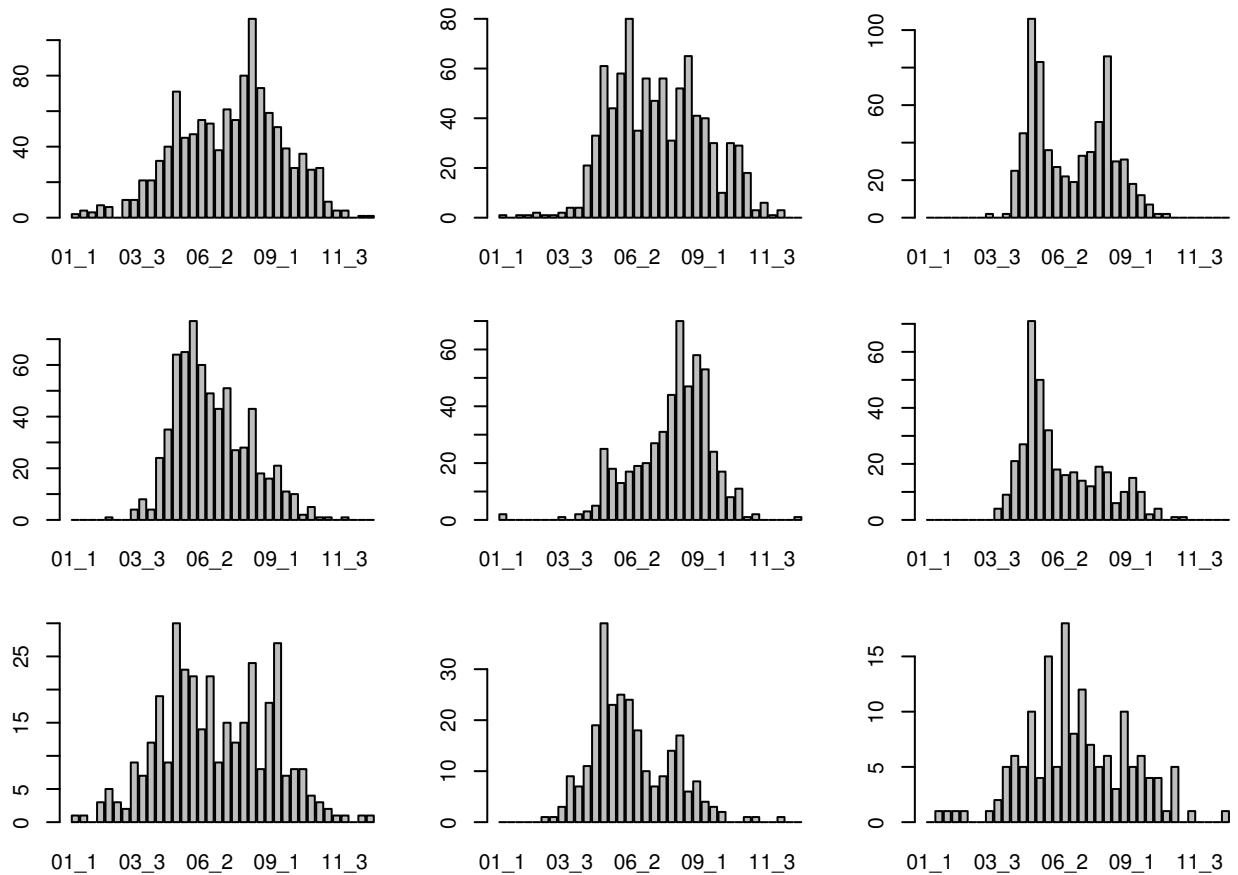
- 6) Faites un graphique phénologique par décade (toutes espèces confondues). Vous pouvez par exemple utiliser la fonction `cut` pour subdiviser le vecteur “jours” en 3 groupes et coller ce nouveau vecteur avec le vecteur mois. (3ème décade de Mars : 55 données)

```
d$decade <- cut(as.numeric(d$D), breaks = c(0,11,21,31), labels = c("1", "2", "3"))
d$decade <- as.factor(paste(d$M, d$decade, sep="_"))
barplot(table(d$decade))
```



- 7) Faites un tableau espèces x décade. Appliquez la fonction barplot à chaque ligne de ce tableau (fonction apply).
Avant la ligne de code apply, insérez la commande suivante qui divise la fenêtre graphique en 9 zones et modifie les marges :
`par(mfrow=c(3,3), mar = c(3,3,1,1))`
NB : il n'y a pas de manière très directe d'ajouter un titre explicite aux graphiques en utilisant apply (en récupérant les noms de colonne). Dans ce cas une boucle peut être plus adaptée.

```
# dev.new(width = 17/2.54, height = 12/2.54)
pheno <- table(d$sp, d$decade)
par(mfrow=c(3,3), mar = c(3,3,1,1))
apply(pheno, 1, barplot)
```



```
##
##      sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## [1,]  0.7  0.7  0.7  0.7  0.7  0.7  0.7  0.7  0.7
## [2,]  1.9  1.9  1.9  1.9  1.9  1.9  1.9  1.9  1.9
## [3,]  3.1  3.1  3.1  3.1  3.1  3.1  3.1  3.1  3.1
## [4,]  4.3  4.3  4.3  4.3  4.3  4.3  4.3  4.3  4.3
## [5,]  5.5  5.5  5.5  5.5  5.5  5.5  5.5  5.5  5.5
## [6,]  6.7  6.7  6.7  6.7  6.7  6.7  6.7  6.7  6.7
## [7,]  7.9  7.9  7.9  7.9  7.9  7.9  7.9  7.9  7.9
## [8,]  9.1  9.1  9.1  9.1  9.1  9.1  9.1  9.1  9.1
## [9,] 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3
## [10,] 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5
## [11,] 12.7 12.7 12.7 12.7 12.7 12.7 12.7 12.7 12.7
## [12,] 13.9 13.9 13.9 13.9 13.9 13.9 13.9 13.9 13.9
## [13,] 15.1 15.1 15.1 15.1 15.1 15.1 15.1 15.1 15.1
```



```
## [14,] 16.3 16.3 16.3 16.3 16.3 16.3 16.3 16.3 16.3 16.3
## [15,] 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5 17.5
## [16,] 18.7 18.7 18.7 18.7 18.7 18.7 18.7 18.7 18.7 18.7
## [17,] 19.9 19.9 19.9 19.9 19.9 19.9 19.9 19.9 19.9 19.9
## [18,] 21.1 21.1 21.1 21.1 21.1 21.1 21.1 21.1 21.1 21.1
## [19,] 22.3 22.3 22.3 22.3 22.3 22.3 22.3 22.3 22.3 22.3
## [20,] 23.5 23.5 23.5 23.5 23.5 23.5 23.5 23.5 23.5 23.5
## [21,] 24.7 24.7 24.7 24.7 24.7 24.7 24.7 24.7 24.7 24.7
## [22,] 25.9 25.9 25.9 25.9 25.9 25.9 25.9 25.9 25.9 25.9
## [23,] 27.1 27.1 27.1 27.1 27.1 27.1 27.1 27.1 27.1 27.1
## [24,] 28.3 28.3 28.3 28.3 28.3 28.3 28.3 28.3 28.3 28.3
## [25,] 29.5 29.5 29.5 29.5 29.5 29.5 29.5 29.5 29.5 29.5
## [26,] 30.7 30.7 30.7 30.7 30.7 30.7 30.7 30.7 30.7 30.7
## [27,] 31.9 31.9 31.9 31.9 31.9 31.9 31.9 31.9 31.9 31.9
## [28,] 33.1 33.1 33.1 33.1 33.1 33.1 33.1 33.1 33.1 33.1
## [29,] 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
## [30,] 35.5 35.5 35.5 35.5 35.5 35.5 35.5 35.5 35.5 35.5
## [31,] 36.7 36.7 36.7 36.7 36.7 36.7 36.7 36.7 36.7 36.7
## [32,] 37.9 37.9 37.9 37.9 37.9 37.9 37.9 37.9 37.9 37.9
## [33,] 39.1 39.1 39.1 39.1 39.1 39.1 39.1 39.1 39.1 39.1
## [34,] 40.3 40.3 40.3 40.3 40.3 40.3 40.3 40.3 40.3 40.3
## [35,] 41.5 41.5 41.5 41.5 41.5 41.5 41.5 41.5 41.5 41.5
## [36,] 42.7 42.7 42.7 42.7 42.7 42.7 42.7 42.7 42.7 42.7
```

- 8) Calculez le nombres d'observations par année pour chaque espèce (sp_1 en 2011 : 94 données). Utilisez deux approches différentes pour obtenir les mêmes chiffres (mais présentés différemment) :
 - fonction table (le plus simple et le plus direct dans ce cas) : vous obtiendrez une table espèces x années à 9 lignes x 14 colonnes
 - fonction aggregate avec comme fonction d'agrégation "length". Les mêmes valeurs seront dans un tableau à 3 colonnes.

```
table(d$sp, d$Y)
```

```
##
##      1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
## sp_1   18   58   83   53   64   70  127  220  141   73   32   39   61
## sp_2    0    0    0    1    9   60  154  176   90   74  104   83   44
## sp_3   16   58   58   44   30   60  125  154   59   21   11    6   17
## sp_4   13   77   43  104   36  154  115   83   13   10    5    7    6
## sp_5    4   33   38   57   48   46   59   55   48   18   37   28   32
## sp_6    5   30   34   49   23   39   43   64   16    6   20   20   10
## sp_7    4   37   38   35   15   33   47   53   14   22   19    8    4
## sp_8    4   18   30   39   19   40   27   39   13    5   15    3    7
## sp_9    0   15    2   41   19   24   19    7   10    6    2    3    3
##
##      2011
## sp_1    94
## sp_2    72
## sp_3    15
## sp_4     3
## sp_5    16
## sp_6    17
## sp_7    17
## sp_8     4
## sp_9     2
```

```
ag <- aggregate(d$sp, d[, c("Y", "sp")], length)
ag[1:20,]
```

```
##      Y    sp    x
## 1 1998 sp_1  18
## 2 1999 sp_1  58
## 3 2000 sp_1  83
## 4 2001 sp_1  53
## 5 2002 sp_1  64
## 6 2003 sp_1  70
## 7 2004 sp_1 127
## 8 2005 sp_1 220
## 9 2006 sp_1 141
## 10 2007 sp_1  73
## 11 2008 sp_1  32
## 12 2009 sp_1  39
## 13 2010 sp_1  61
## 14 2011 sp_1  94
## 15 2001 sp_2   1
## 16 2002 sp_2   9
## 17 2003 sp_2  60
## 18 2004 sp_2 154
## 19 2005 sp_2 176
## 20 2006 sp_2  90
```

- 9) De façon similaire à la question précédente, calculez le nombre maximum d'individus(max(n)) observés pour chaque année et chaque espèce avec les deux approches possibles. NB : Vous devrez utiliser tapply à la place de table ici.

```
tapply(d$n, d[, c("Y", "sp")], max)
```

```
##      sp
## Y    sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 1998    8  NA   2    5   3    4    7    1  NA
## 1999   60  NA  15   66  19   14   30    8    9
## 2000  100  NA  27   12  29   12   17    8    2
## 2001   35  20  25   36  20   13   13   30   41
## 2002   50 300   6   24  15    3    6    6    3
## 2003  100 1000  15  106  10   16   10   41   38
## 2004  200  100  40  100  20   20    5    7    7
## 2005  300  100  43   20  58   25    6    5    8
## 2006   50   50  12    5  21    9   12    1    3
## 2007   23   50  13    8   6    7   10    5    8
## 2008   53  150   3    1   8    6   20    4    1
## 2009   50  100   2    3  35   10    4    3    3
## 2010   25   12   4    2  10   20    3    7    1
## 2011  100  500   4    4  50    3    5    3    1
```

```
ag <- aggregate(d$n, d[, c("Y", "sp")], max)
ag[1:20,]
```

```
##      Y    sp    x
## 1 1998 sp_1    8
## 2 1999 sp_1   60
## 3 2000 sp_1  100
## 4 2001 sp_1   35
## 5 2002 sp_1   50
## 6 2003 sp_1  100
## 7 2004 sp_1  200
## 8 2005 sp_1  300
## 9 2006 sp_1   50
## 10 2007 sp_1   23
## 11 2008 sp_1   53
```

```
## 12 2009 sp_1 50
## 13 2010 sp_1 25
## 14 2011 sp_1 100
## 15 2001 sp_2 20
## 16 2002 sp_2 300
## 17 2003 sp_2 1000
## 18 2004 sp_2 100
## 19 2005 sp_2 100
## 20 2006 sp_2 50
```

- 10) Trouvez la date d'observation la plus précoce pour chaque année et chaque espèce. (= 215 en 1998 pour sp_1) Commencez par créer une colonne au format Date. Utilisez ensuite la fonction aggregate et min. Transformez ensuite la colonne date du résultat de votre aggregate en nombre de jours depuis le premier janvier de chaque année. Vous pouvez utiliser pour ce faire la fonction "format" avec pour argument format "%j" à transformer ensuite en numérique (voir aide de strptime). Il n'est pas nécessaire de passer en format POSIX.
Réorganisez ensuite le jeu de données obtenu en un tableau espèces x années. Unstack ne fonctionne pas bien ici à cause des valeurs manquantes pour certaines années. Vous pouvez utiliser soit la fonction cast du package reshape, soit la fonction tapply avec comme fonction d'agrégation la somme ou la moyenne. Comme il n'y a qu'une seule valeur par combinaison d'espèce x année cette opération ne changera pas les valeurs et ne fera que réorganiser le jeu de données.
NB : Vous pouvez aussi essayer tapply à la place de aggregate dès le début mais le format date est alors directement transformé en numérique et les manipulations sont un peu plus compliquées. Pour pouvoir transformer cette table en format date, il faut d'abord transformer la table en data.frame avec as.data.frame.matrix.
Ensuite on peut transformer ces valeurs en appliquant as.Date à chaque colonne avec lapply et en spécifiant comme origine le 1er janvier 1970.

```
d$date <- as.Date(as.character(d$dat2), format = "%Y%m%d")
res <- aggregate(d$date, list(Y= d$Y, sp = d$sp), min)
res[1:20,]
```

```
##      Y    sp      x
## 1 1998 sp_1 1998-08-03
## 2 1999 sp_1 1999-04-18
## 3 2000 sp_1 2000-04-16
## 4 2001 sp_1 2001-05-01
## 5 2002 sp_1 2002-04-03
## 6 2003 sp_1 2003-01-18
## 7 2004 sp_1 2004-01-15
## 8 2005 sp_1 2005-01-30
## 9 2006 sp_1 2006-01-10
## 10 2007 sp_1 2007-01-14
## 11 2008 sp_1 2008-01-07
## 12 2009 sp_1 2009-04-01
## 13 2010 sp_1 2010-01-23
## 14 2011 sp_1 2011-02-06
## 15 2001 sp_2 2001-09-28
## 16 2002 sp_2 2002-07-30
## 17 2003 sp_2 2003-04-16
## 18 2004 sp_2 2004-03-20
## 19 2005 sp_2 2005-04-01
## 20 2006 sp_2 2006-04-05
```

```
res$x <- as.numeric(format(res$x, "%j"))
```

```
# unstack : pas très adapté ici à cause du fait qu'il y a des valeurs manquantes
# pour certaines années
unstack(res[, c("Y", "x")], x ~ Y)
```

```
## $`1998`
## [1] 215 208 208 216 123 216 98
##
## $`1999`
## [1] 108 113 108 129 114 72 83 134
##
## $`2000`
## [1] 107 112 112 119 109 58 58 155
##
## $`2001`
## [1] 121 271 119 92 121 83 13 74 125
##
## $`2002`
## [1] 93 211 113 88 111 92 83 75 29
##
## $`2003`
## [1] 18 106 107 72 109 95 90 88 91
##
## $`2004`
## [1] 15 80 81 44 108 91 91 68 88
##
## $`2005`
## [1] 30 91 100 82 4 86 76 90 91
##
## $`2006`
## [1] 10 95 110 107 4 109 10 126 18
##
## $`2007`
## [1] 14 31 71 71 92 99 34 111 47
##
## $`2008`
## [1] 7 7 124 124 95 115 41 96 104
##
## $`2009`
## [1] 91 88 104 171 123 108 105 140 193
##
## $`2010`
## [1] 23 118 114 114 114 90 77 84 75
##
## $`2011`
## [1] 37 38 100 121 78 100 38 136 245
```

```
# tapply
tapply(res$x, res[,c("sp", "Y")], sum)
```

```
##      Y
## sp    1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
## sp_1 215 108 107 121 93 18 15 30 10 14 7 91 23
## sp_2 NA NA NA 271 211 106 80 91 95 31 7 88 118
## sp_3 208 113 112 119 113 107 81 100 110 71 124 104 114
## sp_4 208 108 112 92 88 72 44 82 107 71 124 171 114
## sp_5 216 129 119 121 111 109 108 4 4 92 95 123 114
## sp_6 123 114 109 83 92 95 91 86 109 99 115 108 90
## sp_7 216 72 58 13 83 90 91 76 10 34 41 105 77
## sp_8 98 83 58 74 75 88 68 90 126 111 96 140 84
## sp_9 NA 134 155 125 29 91 88 91 18 47 104 193 75
##      Y
## sp    2011
## sp_1 37
## sp_2 38
## sp_3 100
## sp_4 121
```

```
## sp_5 78
## sp_6 100
## sp_7 38
## sp_8 136
## sp_9 245
```

```
# cast
library(reshape)
cast(res, sp ~ Y)
```

```
## Using x as value column. Use the value argument to cast to override this choice
```

```
##      sp 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
## 1 sp_1 215 108 107 121 93 18 15 30 10 14 7 91 23
## 2 sp_2 NA NA NA 271 211 106 80 91 95 31 7 88 118
## 3 sp_3 208 113 112 119 113 107 81 100 110 71 124 104 114
## 4 sp_4 208 108 112 92 88 72 44 82 107 71 124 171 114
## 5 sp_5 216 129 119 121 111 109 108 4 4 92 95 123 114
## 6 sp_6 123 114 109 83 92 95 91 86 109 99 115 108 90
## 7 sp_7 216 72 58 13 83 90 91 76 10 34 41 105 77
## 8 sp_8 98 83 58 74 75 88 68 90 126 111 96 140 84
## 9 sp_9 NA 134 155 125 29 91 88 91 18 47 104 193 75
## 2011
## 1 37
## 2 38
## 3 100
## 4 121
## 5 78
## 6 100
## 7 38
## 8 136
## 9 245
```

```
# avec tapply à la place de aggregate
a <- tapply(d$date, list(Y= d$Y, sp = d$sp), min)
a
```

```
##      sp
## Y      sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 1998 10441 NA 10434 10434 10442 10349 10442 10324 NA
## 1999 10699 NA 10704 10699 10720 10705 10663 10674 10725
## 2000 11063 NA 11068 11068 11075 11065 11014 11014 11111
## 2001 11443 11593 11441 11414 11443 11405 11335 11396 11447
## 2002 11780 11898 11800 11775 11798 11779 11770 11762 11716
## 2003 12070 12158 12159 12124 12161 12147 12142 12140 12143
## 2004 12432 12497 12498 12461 12525 12508 12508 12485 12505
## 2005 12813 12874 12883 12865 12787 12869 12859 12873 12874
## 2006 13158 13243 13258 13255 13152 13257 13158 13274 13166
## 2007 13527 13544 13584 13584 13605 13612 13547 13624 13560
## 2008 13885 13885 14002 14002 13973 13993 13919 13974 13982
## 2009 14335 14332 14348 14415 14367 14352 14349 14384 14437
## 2010 14632 14727 14723 14723 14723 14699 14686 14693 14684
## 2011 15011 15012 15074 15095 15052 15074 15012 15110 15219
```

```
b <- as.data.frame.matrix(a)
b[,1:ncol(b)] <- lapply(b, as.Date, origin = "1970-01-01")
b
```

```
##      sp_1      sp_2      sp_3      sp_4      sp_5      sp_6
## 1998 1998-08-03 <NA> 1998-07-27 1998-07-27 1998-08-04 1998-05-03
```

```
## 1999 1999-04-18      <NA> 1999-04-23 1999-04-18 1999-05-09 1999-04-24
## 2000 2000-04-16      <NA> 2000-04-21 2000-04-21 2000-04-28 2000-04-18
## 2001 2001-05-01 2001-09-28 2001-04-29 2001-04-02 2001-05-01 2001-03-24
## 2002 2002-04-03 2002-07-30 2002-04-23 2002-03-29 2002-04-21 2002-04-02
## 2003 2003-01-18 2003-04-16 2003-04-17 2003-03-13 2003-04-19 2003-04-05
## 2004 2004-01-15 2004-03-20 2004-03-21 2004-02-13 2004-04-17 2004-03-31
## 2005 2005-01-30 2005-04-01 2005-04-10 2005-03-23 2005-01-04 2005-03-27
## 2006 2006-01-10 2006-04-05 2006-04-20 2006-04-17 2006-01-04 2006-04-19
## 2007 2007-01-14 2007-01-31 2007-03-12 2007-03-12 2007-04-02 2007-04-09
## 2008 2008-01-07 2008-01-07 2008-05-03 2008-05-03 2008-04-04 2008-04-24
## 2009 2009-04-01 2009-03-29 2009-04-14 2009-06-20 2009-05-03 2009-04-18
## 2010 2010-01-23 2010-04-28 2010-04-24 2010-04-24 2010-04-24 2010-03-31
## 2011 2011-02-06 2011-02-07 2011-04-10 2011-05-01 2011-03-19 2011-04-10
##          sp_7      sp_8      sp_9
## 1998 1998-08-04 1998-04-08      <NA>
## 1999 1999-03-13 1999-03-24 1999-05-14
## 2000 2000-02-27 2000-02-27 2000-06-03
## 2001 2001-01-13 2001-03-15 2001-05-05
## 2002 2002-03-24 2002-03-16 2002-01-29
## 2003 2003-03-31 2003-03-29 2003-04-01
## 2004 2004-03-31 2004-03-08 2004-03-28
## 2005 2005-03-17 2005-03-31 2005-04-01
## 2006 2006-01-10 2006-05-06 2006-01-18
## 2007 2007-02-03 2007-04-21 2007-02-16
## 2008 2008-02-10 2008-04-05 2008-04-13
## 2009 2009-04-15 2009-05-20 2009-07-12
## 2010 2010-03-18 2010-03-25 2010-03-16
## 2011 2011-02-07 2011-05-16 2011-09-02
```

- 11) Ajoutez au jeu de données un vecteur de dates en nombre de jours depuis le premier janvier de cette année. (format(x, format = "%j") Sur base de ce nouveau vecteur, trouvez les 5 dates les plus précoces par année et par espèce et calculez en la moyenne (pour chaque année et chaque espèce). Vous pouvez utiliser aggregate ou tapply au choix. Vous devrez construire une fonction personnalisée qui trie les dates, extrait les 5 premières et en calcule la moyenne (sans oublier de gérer les NA).

```
d$date2 <- as.numeric(format(d$date, "%j"))

res <- aggregate(d$date2, list(Y= d$Y, sp = d$sp), function(x) mean(sort(x)[1:5],
                                                                    na.rm=TRUE))

res[1:20,]
```

```
##      Y   sp      x
## 1 1998 sp_1 215.6
## 2 1999 sp_1 134.2
## 3 2000 sp_1 111.0
## 4 2001 sp_1 132.2
## 5 2002 sp_1 105.6
## 6 2003 sp_1  74.2
## 7 2004 sp_1  60.4
## 8 2005 sp_1  71.2
## 9 2006 sp_1  46.6
## 10 2007 sp_1  40.6
## 11 2008 sp_1  36.2
## 12 2009 sp_1 115.6
## 13 2010 sp_1  65.6
## 14 2011 sp_1  40.2
## 15 2001 sp_2 271.0
## 16 2002 sp_2 242.8
## 17 2003 sp_2 110.2
## 18 2004 sp_2 103.2
## 19 2005 sp_2 106.8
## 20 2006 sp_2 106.2
```

```
tapply(d$date2, list(Y= d$Y, sp = d$sp), function(x) mean(sort(x)[1:5], na.rm=TRUE))
```

```
##      sp
## Y    sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 1998 215.6  NA 214.6 214.0 226.0 207.8 217.75 186.2500  NA
## 1999 134.2  NA 116.4 115.0 145.8 121.0 114.20 112.0000 156.4000
## 2000 111.0  NA 115.8 117.6 121.4 117.4 106.60 108.4000 197.5000
## 2001 132.2 271.0 120.6 113.0 126.0 112.0 89.40 99.0000 140.6000
## 2002 105.6 242.8 116.2 104.6 121.4 112.4 99.60 86.6000 80.2000
## 2003 74.2 110.2 112.4 82.0 131.8 104.2 104.80 100.0000 104.4000
## 2004 60.4 103.2 102.0 75.2 145.2 108.6 95.40 96.4000 112.0000
## 2005 71.2 106.8 104.0 102.0 110.0 98.6 83.80 103.6000 120.6000
## 2006 46.6 106.2 118.8 118.4 108.0 121.4 73.40 126.8000 112.8000
## 2007 40.6 56.0 97.8 104.2 133.0 122.2 52.40 178.8000 99.8000
## 2008 36.2 64.8 125.4 136.4 140.6 124.4 51.20 124.4000 116.0000
## 2009 115.6 104.8 157.2 195.0 146.2 115.0 156.60 173.3333 202.6667
## 2010 65.6 120.2 131.4 162.8 135.0 114.4 200.00 125.4000 107.3333
## 2011 40.2 96.0 119.4 146.0 137.8 116.8 73.80 183.0000 263.0000
```

- 12) Pour chaque espèce faites une liste des carrés de 5 km² (utm5) où cette espèce a été observée au moins 3 fois. Ensuite comptez les nombres de ces carrés pour chaque espèce. Commencez par faire une table utm5 x sp. Sauvez les noms de lignes (noms des utm5) dans un vecteur. Ensuite avec l'aide de la fonction apply, récupérez pour chaque espèce les noms des utm5 pour lesquelles le nombre d'observations est > 3 (vous devrez construire une fonction à la volée). Le résultat sera une liste. Avec les fonctions lapply puis sapply comptez le nombre d'utm5 pour chaque espèce. Visualisez les différences de format entre ces deux fonctions.

```
utm <- table(d$utm5, d$sp)
head(utm)
```

```
##
##      sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 31UDS675525 0 1 0 0 0 0 0 0 0
## 31UDS675575 2 1 1 0 0 0 0 0 0
## 31UDS675625 0 0 1 1 1 0 1 0 0
## 31UDS725425 1 0 0 0 0 0 0 0 0
## 31UDS725475 1 0 0 0 0 0 0 0 0
## 31UDS725525 1 0 0 1 0 0 0 0 0
```

```
utmnames <- row.names(utm)
l <- apply(utm, 2, function(x) utmnames[x>3])
l[6:9]
```

```
## $sp_6
## [1] "31UDS825325" "31UES475325" "31UES475375" "31UES725325" "31UES775475"
## [6] "31UES925375" "31UES975275" "31UES975325" "31UES975375" "31UFR225525"
## [11] "31UFR375625" "31UFR575525" "31UFS025875" "31UFS125175" "31UFS675525"
## [16] "31UFS725075" "31UFS875125" "31UGR025975" "31UGS025125"
##
## $sp_7
## [1] "31UES275325" "31UES425725" "31UES625625" "31UES775475" "31UES975325"
## [6] "31UES975375" "31UFR075475" "31UFR125475" "31UFR175525" "31UFR175575"
## [11] "31UFR225525" "31UFR325925" "31UFR375525" "31UFR375575" "31UFR375725"
## [16] "31UFR525525" "31UFR525575" "31UFR575525" "31UFS225025" "31UFS275075"
## [21] "31UFS425925" "31UFS775525" "31UGR025975" "31UGS025125"
##
## $sp_8
## [1] "31UDS825325" "31UES475325" "31UES925375" "31UES975325" "31UES975375"
```

```
## [6] "31UFR375675" "31UFS225025" "31UFS675575" "31UFS925125" "31UGR025975"
## [11] "31UGS025125"
##
## $sp_9
## [1] "31UER975825" "31UES975325" "31UES975375" "31UFR325925" "31UFS025325"
## [6] "31UFS175525" "31UFS225025" "31UFS725075" "31UGS025125"
```

```
lapply(1, length)
```

```
## $sp_1
## [1] 90
##
## $sp_2
## [1] 52
##
## $sp_3
## [1] 43
##
## $sp_4
## [1] 42
##
## $sp_5
## [1] 25
##
## $sp_6
## [1] 19
##
## $sp_7
## [1] 24
##
## $sp_8
## [1] 11
##
## $sp_9
## [1] 9
```

```
sapply(1, length)
```

```
## sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 90 52 43 42 25 19 24 11 9
```

- 13) Pour chaque espèce calculez le nombre de carrés utm5 où l'espèce a été vue au moins 0, 1, 2, 3, ... fois. Appliquez simplement (avec apply) la fonction table à chaque colonne de votre tableau utm5 x sp

```
apply(utm, 2, table)
```

```
## $sp_1
##
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 36 43 62
## 289 182 76 34 25 12 16 13 7 4 3 2 2 1 2 1 1 1
##
## $sp_2
##
## 0 1 2 3 4 5 6 7 8 9 10 11 18 20 23 24 27 34
## 389 147 54 29 10 11 9 3 4 2 3 2 1 1 1 1 1 1
## 35 72
## 1 1
##
## $sp_3
```



```
##
## 0 1 2 3 4 5 6 7 8 9 11 25 29 33 38 42
## 412 149 51 16 13 11 6 4 1 2 1 1 1 1 1
##
## $sp_4
##
## 0 1 2 3 4 5 6 7 8 9 10 12 13 15 16 18 19 36
## 431 134 46 18 14 10 4 1 1 2 1 2 1 1 1 1 1 1
## 75
## 1
##
## $sp_5
##
## 0 1 2 3 4 5 6 7 8 10 12 13 16 134
## 476 122 33 15 9 4 5 1 1 1 1 1 1 1
##
## $sp_6
##
## 0 1 2 3 4 5 6 8 12 13 16
## 474 125 28 25 7 5 3 1 1 1 1
##
## $sp_7
##
## 0 1 2 3 4 5 6 7 11 13
## 506 93 30 18 12 3 3 3 1 2
##
## $sp_8
##
## 0 1 2 3 4 5 7 8 15
## 520 98 30 12 3 4 2 1 1
##
## $sp_9
##
## 0 1 2 3 4 5 9 10 14
## 592 54 9 7 3 3 1 1 1
```

- 14) Créez un tableau espèces de plantes x espèces d'insectes donnant le nombre d'observations pour chaque combinaison.
 Eliminez les combinaisons plante x espèce pour lesquelles on a seulement une observation (transformez les 1 en 0) et ensuite éliminez les plantes pour lesquelles on a aucune observations (pex en créant un vecteur logique sur base de rowSums).
 Combien y a-t-il de plantes hôtes par espèce sur base de ce tableau ? (131 plantes pour sp_1)

```
plant <- table(d$plante, d$sp)
plant[plant == 1] <- 0
plant <- plant[rowSums(plant)>0,]
plant[1:20,]
```

```
##
##          sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## Acer campestre      5    8    7    4    4    4    2    3    0
## Acer platanoides     0    4    0    0    2    5    3    0    0
## Acer pseudoplatanus  4   27   16   25   12   19   20   19    2
## Achillea millefolium  9    0    2    0    0    0    0    0    0
## Aegopodium podagraria  0    2   10    0    0    0    0    0    0
## Aesculus hippocastanum 0    0    2    2    0    0    0    0    0
## Alcea rosea          3    0    0    0    0    0    0    0    0
## Alliaria petiolata    4    0    4    0    0    0    0    0    0
## Alnus glutinosa       3   20    7   10    8    5    3    3    0
## Angelica archangelica  0    0    0    2    0    0    0    0    0
## Angelica sylvestris   8   10    6    0    0    0    0    0    0
```

```
## Anthriscus sylvestris 20 8 6 6 2 2 0 0 0
## Apium graveolens 10 0 2 5 0 0 0 0 0
## Arctium lappa 3 2 0 2 2 0 0 0 0
## Arctium minus 3 6 0 8 0 0 0 0 0
## Arrhenatherum elatius 0 0 3 0 0 0 0 0 0
## Artemisia absinthium 0 2 0 0 0 0 0 0 0
## Artemisia vulgaris 13 12 0 5 6 0 0 0 0
## Arum maculatum 0 0 2 0 0 0 0 0 0
## Aster linosyris 4 0 0 0 0 0 0 0 0
```

```
# nombre de plantes hôtes
colSums(plant>0)
```

```
## sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## 131 87 81 72 62 38 34 32 9
```

- 15) Pour chaque espèce, établissez la liste des dix plantes hôtes les plus utilisées avec le nombre d'observations pour chaque plante. Vous devez d'abord transformer votre tableau plante x espèce en data.frame. Si votre tableau est encore au format "table" vous devez utiliser as.data.frame.matrix. Si il est déjà au format "matrix" (ça dépend des opérations faites sur votre table d'origine) vous pouvez utiliser juste as.data.frame ou as.data.frame.matrix. Ensuite vous pouvez utiliser lapply pour appliquer à chaque colonne du tableau plante x espèce une fonction personnalisée qui trie les abondances par ordre décroissant et extrait les 10 premières. Avec une simple fonction comme celle-là vous aurez les nombres d'observations mais vous aurez perdu les noms des lignes (plantes). Il faut donc ajouter un argument à la fonction qui reçoit les noms des plantes et les ajouter au sein de votre fonction personnalisée. Vous pouvez ensuite modifier votre fonction de manière à ajouter par exemple une colonne de % d'observations (nb obs / sum (nb obs)) et à la fin le nombre total d'observations et le nombre total d'espèces de plantes.

```
class(plant)
```

```
## [1] "matrix"
```

```
plant <- as.data.frame.matrix(plant)
class(plant)
```

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

```
# avec le code suivant on a pas les noms de plantes
lapply(plant, function(x) {sort(x, decreasing = TRUE)[1:10]})
```

```
## $sp_1
## [1] 146 58 51 47 40 30 29 27 27 21
##
## $sp_2
## [1] 82 45 42 36 35 29 27 27 27 25
##
## $sp_3
## [1] 66 49 37 27 26 24 18 16 13 12
##
## $sp_4
## [1] 62 36 34 33 25 25 24 22 19 15
##
## $sp_5
## [1] 94 68 31 30 13 12 12 11 10 8
##
## $sp_6
## [1] 100 25 22 20 19 18 18 10 9 7
```

```
##
## $sp_7
## [1] 51 38 27 24 20 14 13 12 11 11
##
## $sp_8
## [1] 23 22 21 19 18 16 14 13 8 8
##
## $sp_9
## [1] 65 53 7 3 3 2 2 2 2 0

# fonction modifiée avec les noms de plantes comme argument
prefered_plants <- function(x, nmax = 10, plnames) {
  names(x) <- plnames
  as.matrix(sort(x, decreasing = TRUE)[1:nmax])
}
lapply(plant, prefered_plants, plnames = row.names(plant), nmax = 5)

## $sp_1
##           [,1]
## Urtica dioica    146
## Pinus sylvestris  58
## Cytisus scoparius  51
## Cirsium arvense   47
## Tanacetum vulgare  40
##
## $sp_2
##           [,1]
## Urtica dioica     82
## Quercus robur     45
## Crataegus monogyna 42
## Betula pendula    36
## Tilia x europaea  35
##
## $sp_3
##           [,1]
## Urtica dioica     66
## Crataegus monogyna 49
## Quercus robur     37
## Betula pendula    27
## Corylus avellana   26
##
## $sp_4
##           [,1]
## Urtica dioica     62
## Betula pendula    36
## Crataegus monogyna 34
## Pinus nigra       33
## Acer pseudoplatanus 25
##
## $sp_5
##           [,1]
## Vitis vinifera    94
## Quercus robur     68
## Potentilla reptans 31
## Heracleum sphondylium 30
## Crataegus monogyna 13
##
## $sp_6
##           [,1]
## Crataegus monogyna 100
## Betula pendula     25
## Quercus robur      22
```

```
## Urtica dioica      20
## Acer pseudoplatanus 19
##
## $sp_7
##                [,1]
## Pinus sylvestris    51
## Quercus robur       38
## Picea abies         27
## Pinus nigra         24
## Acer pseudoplatanus 20
##
## $sp_8
##                [,1]
## Picea abies         23
## Crataegus monogyna  22
## Tilia x europaea    21
## Acer pseudoplatanus 19
## Pinus nigra         18
##
## $sp_9
##                [,1]
## Pinus nigra         65
## Pinus sylvestris    53
## Picea abies         7
## Betula pendula      3
## Pinus strobus       3
```

```
lapply(plant, preferred_plants, plnames = row.names(plant), nmax = 10)
```

```
## $sp_1
##                [,1]
## Urtica dioica      146
## Pinus sylvestris    58
## Cytisus scoparius   51
## Cirsium arvense     47
## Tanacetum vulgare   40
## Calluna vulgaris    30
## Quercus robur       29
## Crataegus monogyna  27
## Heracleum sphondylium 27
## Betula pendula      21
##
## $sp_2
##                [,1]
## Urtica dioica       82
## Quercus robur       45
## Crataegus monogyna  42
## Betula pendula      36
## Tilia x europaea    35
## Corylus avellana     29
## Acer pseudoplatanus 27
## Phragmites australis 27
## Tanacetum vulgare   27
## Pinus sylvestris    25
##
## $sp_3
##                [,1]
## Urtica dioica       66
## Crataegus monogyna  49
## Quercus robur       37
## Betula pendula      27
## Corylus avellana     26
```

```

## Cytisus scoparius      24
## Tanacetum vulgare     18
## Acer pseudoplatanus   16
## Phragmites australis  13
## Fagus sylvatica       12
##
## $sp_4
##                      [,1]
## Urtica dioica         62
## Betula pendula        36
## Crataegus monogyna    34
## Pinus nigra           33
## Acer pseudoplatanus   25
## Tanacetum vulgare     25
## Quercus robur         24
## Salix caprea          22
## Malus sylvestris      19
## Tilia x europaea      15
##
## $sp_5
##                      [,1]
## Vitis vinifera        94
## Quercus robur         68
## Potentilla reptans    31
## Heracleum sphondylium 30
## Crataegus monogyna    13
## Acer pseudoplatanus   12
## Urtica dioica         12
## Filipendula ulmaria    11
## Carpinus betulus      10
## Alnus glutinosa        8
##
## $sp_6
##                      [,1]
## Crataegus monogyna    100
## Betula pendula        25
## Quercus robur         22
## Urtica dioica         20
## Acer pseudoplatanus   19
## Corylus avellana       18
## Tilia x europaea      18
## Salix caprea          10
## Fagus sylvatica        9
## Sorbus aucuparia       7
##
## $sp_7
##                      [,1]
## Pinus sylvestris      51
## Quercus robur         38
## Picea abies           27
## Pinus nigra           24
## Acer pseudoplatanus   20
## Malus sylvestris      14
## Betula pendula        13
## Fagus sylvatica       12
## Crataegus monogyna    11
## Fraxinus excelsior     11
##
## $sp_8
##                      [,1]
## Picea abies           23
## Crataegus monogyna    22
## Tilia x europaea      21

```

```
## Acer pseudoplatanus    19
## Pinus nigra             18
## Betula pendula         16
## Quercus robur          14
## Corylus avellana       13
## Carpinus betulus       8
## Pinus sylvestris       8
##
## $sp_9
##                        [,1]
## Pinus nigra           65
## Pinus sylvestris      53
## Picea abies           7
## Betula pendula        3
## Pinus strobus         3
## Acer pseudoplatanus   2
## Calluna vulgaris      2
## Corylus avellana      2
## Larix decidua         2
## Acer campestre        0
```

```
preferred_plants <- function(x, nmax = 10, plnames) {
  tot <- sum(x)
  nbtaxa <- sum(x>0)
  res <- data.frame(nbobs = x, pct = round(x*100/tot,1) )
  row.names(res) <- as.character(plnames)
  res <- res[order(res$nbobs, decreasing = TRUE),][1:nmax,]
  res <- rbind(res, c( "---", "---"), c( tot, "100"), c(nbtaxa, ""))
  row.names(res)[(nmax+1):(nmax+3)] <- c("---", "Total", "nb taxa")
  return(res)
}
lapply(plant, preferred_plants, plnames = row.names(plant))
```

```
## $sp_1
##                nbobs  pct
## Urtica dioica      146 14.2
## Pinus sylvestris    58  5.7
## Cytisus scoparius   51  5
## Cirsium arvense     47  4.6
## Tanacetum vulgare   40  3.9
## Calluna vulgaris    30  2.9
## Quercus robur       29  2.8
## Crataegus monogyna  27  2.6
## Heracleum sphondylium 27  2.6
## Betula pendula      21  2
## ---              ---  ---
## Total              1025 100
## nb taxa            131
##
## $sp_2
##                nbobs  pct
## Urtica dioica      82 10.3
## Quercus robur       45  5.6
## Crataegus monogyna  42  5.3
## Betula pendula      36  4.5
## Tilia x europaea    35  4.4
## Corylus avellana    29  3.6
## Acer pseudoplatanus 27  3.4
## Phragmites australis 27  3.4
## Tanacetum vulgare   27  3.4
## Pinus sylvestris    25  3.1
```

```

## ---          --- ---
## Total              799 100
## nb taxa            87
##
## $sp_3
##              nbobs  pct
## Urtica dioica        66 11.1
## Crataegus monogyna   49  8.2
## Quercus robur        37  6.2
## Betula pendula       27  4.5
## Corylus avellana     26  4.4
## Cytisus scoparius    24   4
## Tanacetum vulgare    18   3
## Acer pseudoplatanus  16  2.7
## Phragmites australis 13  2.2
## Fagus sylvatica     12   2
## ---          --- ---
## Total              595 100
## nb taxa            81
##
## $sp_4
##              nbobs  pct
## Urtica dioica        62 10.4
## Betula pendula       36   6
## Crataegus monogyna   34  5.7
## Pinus nigra          33  5.5
## Acer pseudoplatanus  25  4.2
## Tanacetum vulgare    25  4.2
## Quercus robur        24   4
## Salix caprea         22  3.7
## Malus sylvestris     19  3.2
## Tilia x europaea     15  2.5
## ---          --- ---
## Total              596 100
## nb taxa            72
##
## $sp_5
##              nbobs  pct
## Vitis vinifera       94 20.4
## Quercus robur        68 14.8
## Potentilla reptans   31  6.7
## Heracleum sphondylium 30  6.5
## Crataegus monogyna   13  2.8
## Acer pseudoplatanus  12  2.6
## Urtica dioica        12  2.6
## Filipendula ulmaria   11  2.4
## Carpinus betulus     10  2.2
## Alnus glutinosa       8  1.7
## ---          --- ---
## Total              461 100
## nb taxa            62
##
## $sp_6
##              nbobs  pct
## Crataegus monogyna   100 29.2
## Betula pendula       25  7.3
## Quercus robur        22  6.4
## Urtica dioica        20  5.8
## Acer pseudoplatanus  19  5.6
## Corylus avellana     18  5.3
## Tilia x europaea     18  5.3
## Salix caprea         10  2.9
## Fagus sylvatica       9  2.6

```

```

## Sorbus aucuparia      7    2
## ---                  ---  ---
## Total                 342  100
## nb taxa               38
##
## $sp_7
##                   nbobs  pct
## Pinus sylvestris     51 16.2
## Quercus robur        38 12.1
## Picea abies          27  8.6
## Pinus nigra          24  7.6
## Acer pseudoplatanus  20  6.3
## Malus sylvestris     14  4.4
## Betula pendula       13  4.1
## Fagus sylvatica      12  3.8
## Crataegus monogyna   11  3.5
## Fraxinus excelsior   11  3.5
## ---                  ---  ---
## Total                 315  100
## nb taxa               34
##
## $sp_8
##                   nbobs  pct
## Picea abies          23  9.8
## Crataegus monogyna   22  9.4
## Tilia x europaea     21   9
## Acer pseudoplatanus  19  8.1
## Pinus nigra          18  7.7
## Betula pendula       16  6.8
## Quercus robur        14   6
## Corylus avellana     13  5.6
## Carpinus betulus      8  3.4
## Pinus sylvestris     8  3.4
## ---                  ---  ---
## Total                 234  100
## nb taxa               32
##
## $sp_9
##                   nbobs  pct
## Pinus nigra          65 46.8
## Pinus sylvestris     53 38.1
## Picea abies          7   5
## Betula pendula       3  2.2
## Pinus strobus        3  2.2
## Acer pseudoplatanus  2  1.4
## Calluna vulgaris     2  1.4
## Corylus avellana     2  1.4
## Larix decidua        2  1.4
## Acer campestre       0   0
## ---                  ---  ---
## Total                 139  100
## nb taxa               9

```

- 16) On veut calculer les incides d'Ellenberg pour chaque espèce. Les indices d'Ellenberg sont des valeurs indicatrices (lumière, température, humidité, richesse du sol, ...) pour une série d'espèces de plantes. On calcule en général un indice synthétique pour un site en multipliant l'abondance des plantes du site par leurs indices d'Ellenberg puis en prenant la somme de ces valeurs divisée par l'abondance totale (une valeur par indice). Il s'agit donc de la moyenne des indices des plantes présentes sur le site pondérée par leur abondance. On peut calculer également ces indices pour les biotopes occupés par des insectes.

– Commencez par charger le fichier Ellenberg.txt, ensuite, fusionnez les noms de lignes de votre tableau

plantes x espèces avec le dataset Ellenberg (merge) afin que les lignes de ce dataset Ellenberg correspondent parfaitement aux lignes de votre tableau plantes x espèces.

- Calculez ensuite les indices d'Ellenberg pour `sp_1` uniquement. Vous pouvez multiplier la matrice d'Ellenberg par le vecteur `sp_1`, diviser par le nombre total d'observations pour `sp_1` et faire la somme de chaque colonne. Attention cependant le nombre total d'observations pour `sp_1` ne devrait prendre en compte que les observations sur les plantes pour lesquelles l'indice d'Ellenberg est disponible. Le dénominateur est donc différent pour chaque indice d'Ellenberg.
- Généralisez ensuite ces calculs pour toutes les espèces (ie les colonnes du tableau plantes x espèces) par exemple au moyen de la fonction `sapply`.

```
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_1_R/Exercices")
ellenberg <- read.table("data/ellenberg.txt", header=TRUE, sep="\t")
summary(ellenberg)
```

```
##           Taxprio_final  Ellenberg_L  Ellenberg_T
## Abies alba           :    1  Min.    :1.000  Min.    :3.000
## Abies nordmanniana   :    1  1st Qu.:6.000  1st Qu.:5.000
## Abutilon theophrasti :    1  Median :7.000  Median :6.000
## Acer campestre       :    1  Mean    :6.875  Mean    :5.899
## Acer negundo         :    1  3rd Qu.:8.000  3rd Qu.:6.000
## Acer platanoides     :    1  Max.    :9.000  Max.    :9.000
## (Other)              :1861  NA's    :306   NA's    :565
## Ellenberg_K          Ellenberg_F    Ellenberg_R    Ellenberg_N
## Min.    :1.000  Min.    : 1.000  Min.    :1.000  Min.    :1.000
## 1st Qu.:3.000  1st Qu.: 4.000  1st Qu.:5.000  1st Qu.:3.000
## Median :3.000  Median : 5.000  Median :7.000  Median :5.000
## Mean    :3.676  Mean    : 5.647  Mean    :6.284  Mean    :4.602
## 3rd Qu.:5.000  3rd Qu.: 7.000  3rd Qu.:8.000  3rd Qu.:6.000
## Max.    :8.000  Max.    :12.000  Max.    :9.000  Max.    :9.000
## NA's    :461   NA's    :368   NA's    :592   NA's    :407
```

```
ellenberg[1:15,]
```

```
##           Taxprio_final Ellenberg_L Ellenberg_T Ellenberg_K
## 1           Actaea spicata           3           5           4
## 2           Adonis aestivalis         6           6           7
## 3           Adonis annua             NA           NA           NA
## 4           Adonis flammea           6           6           6
## 5           Adoxa moschatellina        5           NA           5
## 6           Aegopodium podagraria       5           5           3
## 7           Abutilon theophrasti       NA           NA           NA
## 8           Acer campestre             5           6           4
## 9           Acer negundo              5           6           6
## 10          Acer platanoides           4           6           4
## 11          Acer pseudoplatanus        4           NA           4
## 12          Achillea millefolium       8           NA           NA
## 13          Achillea ptarmica          8           6           3
## 14          Acinos arvensis           9           6           3
## 15 Aconitum lycoctonum subsp.vulparia  3           4           4
## Ellenberg_F Ellenberg_R Ellenberg_N
## 1           5           6           7
## 2           3           8           3
## 3           NA          NA          NA
## 4           3           9           3
## 5           6           7           8
## 6           6           7           8
## 7           NA          NA          NA
## 8           5           7           6
```

```
## 9      6      7      7
## 10     NA     NA     NA
## 11     6     NA     7
## 12     4     NA     5
## 13     8     4     2
## 14     2     5     1
## 15     7     7     8
```

Fusion des jeux de données

```
plantname <- data.frame(name = row.names(plant))
ellenberg <- merge(plantname, ellenberg, by.x = "name", by.y = "Taxprio_final",
  all.x=TRUE)
ellenberg[1:10,]
```

```
##           name Ellenberg_L Ellenberg_T Ellenberg_K Ellenberg_F
## 1      Acer campestre          5          6          4          5
## 2      Acer platanoides         4          6          4         NA
## 3      Acer pseudoplatanus        4         NA          4          6
## 4      Achillea millefolium        8         NA         NA          4
## 5      Aegopodium podagraria        5          5          3          6
## 6      Aesculus hippocastanum       NA         NA         NA         NA
## 7          Alcea rosea          NA         NA         NA         NA
## 8      Alliaria petiolata          NA         NA         NA         NA
## 9          Alnus glutinosa          5          5          3          9
## 10     Angelica archangelica        7          6          5          9
##      Ellenberg_R Ellenberg_N
## 1          7          6
## 2         NA         NA
## 3         NA          7
## 4         NA          5
## 5          7          8
## 6         NA         NA
## 7         NA         NA
## 8         NA         NA
## 9          6         NA
## 10        NA          9
```

Calcul des indices d'Ellenberg pour une espèce

On pourrait être tenté de faire le calcul suivant :

```
colSums(plant[,1] * ellenberg[,-1] / sum(plant[,1]), na.rm = TRUE)
```

```
## Ellenberg_L Ellenberg_T Ellenberg_K Ellenberg_F Ellenberg_R Ellenberg_N
##      5.216585      3.083902      2.243902      3.495610      3.366829      4.294634
```

Avec le calcul ci-dessus, on ne tient pas compte du fait que certaines espèces de plantes n'ont pas de valeurs pour certains indices d'Ellenberg. Le dénominateur n'est donc pas bon on divise systématiquement par un nombre trop grand.

On doit faire pour chaque indice la somme du nombre d'observations pour lesquelles il existe un indice d'ellenberg, comme ceci par exemple :

```
colSums(plant[,1] * ellenberg[,-1], na.rm = TRUE) /
  colSums(plant[,1] * (!is.na(ellenberg[,-1])))
```

```
## Ellenberg_L Ellenberg_T Ellenberg_K Ellenberg_F Ellenberg_R Ellenberg_N
##      7.044796      5.545614      3.814262      5.276878      6.308958      6.130919
```

Calcul des indices d'Ellenberg pour toutes les espèces

```
result <- sapply(plant, function(x) {round(colSums(x * ellenberg[,-1],na.rm = TRUE) /
  colSums(x * (!is.na(ellenberg[,-1]))),1)})
result
```

```
##           sp_1 sp_2 sp_3 sp_4 sp_5 sp_6 sp_7 sp_8 sp_9
## Ellenberg_L 7.0 6.5 6.5 6.7 6.6 6.2 6.1 5.9 6.9
## Ellenberg_T 5.5 5.5 5.3 5.6 5.6 5.3 5.4 5.2 6.6
## Ellenberg_K 3.8 3.9 3.6 3.7 3.9 3.5 4.6 4.1 5.3
## Ellenberg_F 5.3 5.9 5.6 5.3 5.6 4.9 4.7 5.1 3.1
## Ellenberg_R 6.3 7.1 6.6 7.1 7.0 7.3 7.2 7.5 8.8
## Ellenberg_N 6.1 6.5 6.0 6.1 6.5 5.4 5.0 5.4 2.2
```

```
t(result)
```

```
##           Ellenberg_L Ellenberg_T Ellenberg_K Ellenberg_F Ellenberg_R
## sp_1           7.0           5.5           3.8           5.3           6.3
## sp_2           6.5           5.5           3.9           5.9           7.1
## sp_3           6.5           5.3           3.6           5.6           6.6
## sp_4           6.7           5.6           3.7           5.3           7.1
## sp_5           6.6           5.6           3.9           5.6           7.0
## sp_6           6.2           5.3           3.5           4.9           7.3
## sp_7           6.1           5.4           4.6           4.7           7.2
## sp_8           5.9           5.2           4.1           5.1           7.5
## sp_9           6.9           6.6           5.3           3.1           8.8
##           Ellenberg_N
## sp_1           6.1
## sp_2           6.5
## sp_3           6.0
## sp_4           6.1
## sp_5           6.5
## sp_6           5.4
## sp_7           5.0
## sp_8           5.4
## sp_9           2.2
```

```
# strictement équivalent ici :
```

```
result <- apply(plant, 2, function(x) {round(colSums(x * ellenberg[,-1],na.rm = TRUE) /
                                                colSums(x * (!is.na(ellenberg[,-1]))),1)})
```