Exemple pas à pas d'analyse d'un jeu de données - DRAFT

Gilles San Martin

01 December 2016

Contents

Introduction

Le but de ce rapport est de présenter pas à pas l'analyse complète d'un jeu de données récolté par des biologistes de terrain dans le but de protéger une espèce menacée de libellule. Le but premier est donc pédagogique et illustratif, ce qui explique la longueur et les nombreux détails donnés. Dans un rapport d'anlayse classique on détaille en général moins les différentes étapes et les différentes possibilités d'analyse... On donne par exemple ici souvent plusieurs manières de représenter la même chose d'abbord avec des lignes de codes simples et un résultat simple puis avec des lignes de code plus complexes pour un résultat plus rafiné...

On a suivi assez étroitement le fil conducteur et les questions posées pour l'exercice qu'on vous demande de réaliser dans le cadre de ce cours (i.e. analyser un jeu de données de distribution d'espèces GBIF). L'ordre des questions a parfois été légèrement modifié en fonction du contexte parce que cet ordre semblait plus approprié pour le traitement réalisé ici. N'hésitez pas à faire de même tout en identifiant clairement les questions correspondantes.

Description du contexte et des questions

```
Q01 - Décrivez brièvement le contexte (ea espèce et/ou variables choisies),
les jeux de données utilisés (GBIF et environnement) et la manière dont ces données
ont été collectées et la ou les questions qu'on se pose (objectifs).
```

Contexte

La demoiselle *Coenagrion mercuriale* (Odonata, Coenagrionidae) est une espèce menacée à l'échelle européenne et inscrite à l'annexe II de la directive européenne 92/43/CEE sur la mise en place du Réseau Natura 2000 en Europe. En pratique cela signifie que les états membres se sont engagés à maintenir en bon état de conservation les habitats de cette espèce et à maintenir voire augmenter par des mesures appropriées la taille des populations ainsi qu'à assurer leur suivi au cours du temps.

D'après la littérature, il s'agit d'une espèce thermophile, vivant dans des petits cours d'eau bien oxygénés, bien éclairés et à courant lent. Une des principales populations en Région Wallone se situe dans la plaine de Focant, près de Beauraing. C'est cette population qui été étudiée ici en juillet 2006.

L'étude a été réalisée principalement par des biologistes du DEMNA (Département D'Etude du Milieu Naturel et Agricole). Le DEMNA est l'organisme publique (Région Wallonne - DGO3) chargé d'évaluer l'état de conservation des habitats Natura 2000, de désigner les sites visés par le décret et d'assurer le suivi de leur état de conservation. Les résultats de cette étude ont été publiés dans Couvreur et al. 2008, Bulletin de la SRBE 144 : 101-115. On ce reportera à cet article pour plus de détails sur la méthodologie.

Objectif - Problème à résoudre

L'objectif général est de déterminer les conditions environnementales favorables à cette espèce de façon à pouvoir proposer des mesures de restauration ou de conservation de son habitat.

Un autre objectif initial était d'évaluer la taille de la population à la mise en place du réseau Natura 2000 afin de pouvoir suivre son évolution au cours du temps. Cet objectif sera laissé de côté ici.

Présentation des données

On a relevé l'abondance des adultes de *Coenagrion mercuriale*, sur une série de drains ainsi qu'une série de facteurs environnementaux suceptibles d'expliquer cette abondance. Voir Couvreur et al. 2008, Bulletin de la SRBE 144 : 101-115 pour plus de détails.

Les drains ont été divisés en 44 tronçons a priori homogènes et les comptages ont tous été fait le même jour (4 juillet 2006) par des naturalistes expérimentés ce qui permet de limiter l'effet des conditions climatiques sur les observations.

Le jeu de données comprend les informations suivantes :

- Abondance des mâles, femelles, tandems (couples appariés) et comportements de ponte le 4 juillet 2006
- Trois identifiants : ID, code et code2. Ce dernier est sans doute le plus utile. Les tronçons paratageant la même première lettre font en fait partie d'un même drain continu.
- x et y : les coordonnées géographiques du centroïde du tronçon (lambert belge 1972, en mètres)
- obs et obsrives : initiales de la personne qui a réalisé les inventaires des insectes ou des caractéristiques environnementales
- longueur (en mètres) et largeur (en cm) du tronçon et hauteur des rives (en cm)
- tourniere : présence/absence d'une tournière en bordure du drain (bande non cultivée)
- recRuisseau : le % de la surface du ruisseau recouvert par de la végétation
- recPhalaris et recBuisson : recouvrement de *Phalaris arundinacea* et de Buissons ou arbres, en classes de recouvrement de Braunn-Blanquet
- Présence/absence d'une série de plantes aquatiques/subaquatiques : Berula, Nasturtium, Callitriche, Glyceria, Groenlandia, Veronica, Carex, Apium
- 3 variables physiques : température, conductivité et pH. 33 troncçons ont été mesurés le 13 septembre 2007 et 3 tronçons supplémentaires (G1, G2, G3) le 17 Octobre 2007. Il y a des valeurs manquantes quand le ruisseau était à sec
- 8 variables physico-chimiques plus approfonides sur base de prélèvements d'eau réalisés en hiver.

Vision du résultat final

```
Q02 - En première approche, sous quelle forme présenterez-vous vos résultats finaux
pour pour pouvoir répondre à votre question ?
```

A la fin de l'analyse on aimerait avoir une liste de variables explicatives liées positivement ou négativement à l'abondance de cette libellule et une série de graphiques représentant ces relations.
On voudrait pouvoir dire par exemple (totalement inventé à ce stade) : "L'abondance de *Coenagrion mercuriale* est liée positivement à la présence de plantes aquatiques du genre *Berula* et négativement à la surface recouverte par des buissons et à la hauteur des berges. L'abondance passe en moyenne de 10 individus à 20 individus en présence de *Berula*, toutes choses étant égales par ailleurs. Lorsque les classes de recouvrement en buisson passent de 0 à 3 (1er et 3ème quartile), l'abondance passe en moyenne de 10 à 0 individus. Il semble que dès que la classe de recouvrement dépasse 2 les drains deviennent très défavorables à ces libellules. La protection de cette espèce requiert donc de limiter le développemnt des buissons. etc. . . "

Biais et problèmes potentiels

```
Q05 - En fonction de ce que vous savez sur la manière dont les jeux de données
ont été récoltés décrivez les problèmes et/ou les biais potentiels dans votre jeu de
données et les conséquences possibles pour la question que vous étudiez.
Est-il possible de mitiger certains de ces problèmes en modifiant le jeu
de données ou en modifiant votre manière d'analyser les données?
```

Le jeu de données a été spécifiquement collecté par des naturalistes expérimentés dans le but de répondre à ces questions. Les variables environnementales mesurées ont été choisie avec soin en fonction de ce qu'on connaît de la biologie de l'espèce. Le jeu de données est donc à la base de très bonne qualité.

Seule la population de la Plaine de Focant a été étudiée et les conclusions ne pourront être extrapolées à d'autres populations qu'avec précaution. Si les résultats trouvés ici sont concordants avec ceux d'autres études réalisées dans d'autres contextes on pourra sans doute les généraliser à d'autres populations.

Un problème potentiel tient au fait qu'on a principalement mesuré l'abondance des adultes qui sont des insectes volants et assez mobiles. On ne peut donc exclure que les zones où les adultes sont les plus abondants ne correspondent pas obligatoirement aux zones où se développent les larves. N'ayant pas de données sur l'abondance des larves on pourrait étudier en plus de l'abondance des adultes la présence/absence (ou l'abondance) des comportements de ponte qui donnent une indication que le tronçon de drain est probablement favorable à la reproduction. Ceci n'est pas idéal non plus car chaque drain n'a été visité qu'une seule fois et l'absence d'observation de comportement de ponte ne veut pas obligatoirement dire que les libellules n'y pondent jamais.

Les variables physico-chimiques n'ont pas été récoltées à la même date que les comptages (l'année suivante...). Pour le pH, conductivité et température 3 tronçons ont de plus été mesurés à une date différente de celles des autres. Si on veut utiliser ces variables ont doit donc faire l'hypothèse d'une certaine stabilité au cours du temps au moins en valeur relative et il faudra interpréter les valeurs absolues avec prudence.
Une mesure unique de la température est souvent un mauvais indicateur tant ce paramètre est variable. De plus les 33 tronçons mesurés la même journée ont vraisemblablement vu leur mesures étalées au cours de la journée et donc dans des conditions de températures très différentes. L'absence d'effet température pourrait donc juste être dû à une trop grande erreur dans les mesures. On a cependant pas de raison de penser a priori que les mesures ont été faite tôt dans la journée sur des sites à forte abondance de *C.mercuriale* et en fin de journée sur des sites à faible abundance. On ne s'attend donc pas à un biais systématique dans les mesures et une relation significative abondance~température pourrait peut-être s'interpréter - mais avec prudence. Malheuereusement l'heure de mesure n'a pas été notée. Cette information aurait peut-être permis d'explorer plus en détails ce problème et peut-être de corriger les valeurs.

Type d'analyses envisagées

```
Q03 - En première approche, vers quel type d'analyse statistique vous orienteriez-vous
à ce stade pour répondre à votre question ?
```

Le but est de prédire une variable (abondance des libellules) en fonction d'une série d'autres variables explicatives. On utilisera donc principalement une méthode supervisée et univariée : les modèles linéaires généralisés (GLMs). Etant donné qu'il s'agit de données de comptage on s'orientera en premère approche vers un GLM avec une distribution de Poisson.
On étudiera aussi le lien entre les mêmes variables explicatives et la présence/absence de comportement de ponte. Pour cette question on s'orientera donc plutôt vers un GLM à distribution binomiale (données binaires).

Import et vérification des données

```
Q04 - Importez les données dans R et vérifiez leur contenu.

* Est-ce que toutes les données ont été importées (nombre de lignes, nombre de colonnes)
* Est-ce qu'il y a des variables qualitatives et si oui ont-elles bien été importées
   comme des facteurs dans R ?
* Est-ce que les variables quantitatives ont bien été importées comme des nombres ?
* Est-ce qu'il y a des valeurs manquantes (dans quelles variables?). Si le jeu de
   données original contenait des cases vides il faut s'assurer qu'elles ont été
   importées correctement (valeurs manquantes, simple champ texte vide)
* Examinez les valeurs pour voir si il n'y a pas manifestement des valeurs impossibles
```

On définit le répertoire de travail principal, on source 2 scripts contenant diverses fonctions utiles et on charge les divers packages que l'on utilisera ici.

```
setwd("/home/gilles/stats/Formation_R_stats/UCL_LBOE2121/mercuriale")
source("/home/gilles/stats/model.select_0.4.1.R")
source("/home/gilles/stats/mytoolbox.R")

library(car)
library(vegan)
library(FactoMineR)
library(visreg)
library(mgcv)

# for maps/GIS
library(rgdal)
library(sp)
library(raster)
library(OpenStreetMap)

# load ggplot, change the default theme and change the locale (language = English)
library(ggplot2)
```

Import du jeu de données

NB : dans le jeu de données original les valeurs manquantes étaient marquées par un point. Ceci est spécifié avec l'argument `na.strings = "."`.

```
d <- read.table("data/mercuriale.csv", sep = ";", dec = ",", header=TRUE,
                na.strings = ".", encoding = "utf8", quote = "\"")
```

`head` permet de voir les première lignes des données et on vérifie qu'on a bien 44 lignes et 37 colonnes comme dans le jeu de données original.

```
dim(d)
```

```
## [1] 44 37
```

```
head(d)
```

```
##   male femelle tandem ponte ID code code2      x     y obs obsrives longueur hauteur largeur
## 1    5       0      0     0 41  J17   B01 197961 91669  XV      JMC   316.63     200     250
```

```
## 2    0       0      0     0 40  J16   B02 198096 91812 JMC      JMC    76.75     200     250
## 3    0       0      0     0 22  J11   B03 198466 92191 JMC      JMC   139.07     200     250
## 4    1       0      0     0 20  J13   B04 198295 92015 JMC      JMC    47.04     200     250
## 5    0       0      0     0 21  J12   B05 198363 92088 JMC      JMC   152.06     200     250
## 6    4       0      0     0 19  J15   B06 198172 91895 JMC      JMC   146.76     200     250
##   tourniere recRuisseau recPhalaris recBuisson Berula Nasturtium Callitriche Glyceria Groenlandia
## 1         0          20           1          3      0          0           0        0           0
## 2         0          20           0          3      0          0           0        0           0
## 3         0          20           0          5      0          0           0        0           0
## 4         0          10           1          3      0          0           0        0           0
## 5         0          10           1          3      0          0           0        0           0
## 6         0          20           2          3      0          0           0        0           0
##   Veronica Carex Apium temp conduct   pH O2dissous saturation nitrates O_Phospate X20.C Alcalin
## 1        0     0     0 22.2     844 8.26     11.45       88.4     43.9       0.36   557    4.07
## 2        0     0     0 22.1     838 8.15     11.45       88.4     43.9       0.36   557    4.07
## 3        0     0     0 22.5     840 8.24     11.45       88.4     43.9       0.36   557    4.07
## 4        0     0     0 22.5     837 8.24     11.45       88.4     43.9       0.36   557    4.07
## 5        0     0     0 22.4     833 8.20     11.45       88.4     43.9       0.36   557    4.07
## 6        0     0     0 22.6     840 8.22     11.45       88.4     43.9       0.36   557    4.07
##     Ca pHCanon
## 1 93.9    8.02
## 2 93.9    8.02
## 3 93.9    8.02
## 4 93.9    8.02
## 5 93.9    8.02
## 6 93.9    8.02
```

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

Un summary permet de vérifier que les variables sont correctement importées (ea pas de fautes de frappe dans les textes) et au bon format et donne de nombreuses informations utiles.

```
summary(d)
```

```
##       male           femelle       tandem           ponte            ID             code
##  Min.   :  0.00   Min.   : 0   Min.   : 0.000   Min.   :0.00   Min.   : 1.00   Biran1 : 1
##  1st Qu.:  3.75   1st Qu.: 0   1st Qu.: 0.000   1st Qu.:0.00   1st Qu.:11.75   Biran2 : 1
##  Median : 15.00   Median : 0   Median : 1.000   Median :0.00   Median :22.50   Biran3 : 1
##  Mean   : 28.55   Mean   : 1   Mean   : 4.455   Mean   :0.25   Mean   :22.50   D1     : 1
##  3rd Qu.: 30.75   3rd Qu.: 1   3rd Qu.: 5.000   3rd Qu.:0.00   3rd Qu.:33.25   D2     : 1
##  Max.   :284.00   Max.   :10   Max.   :40.000   Max.   :4.00   Max.   :44.00   D3     : 1
##                                                                                (Other):38
##      code2          x               y          obs     obsrives    longueur          hauteur
##  B01    : 1   Min.   :193991   Min.   :89717   DT : 3   DT : 2   Min.   :  47.04   Min.   : 20.00
##  B02    : 1   1st Qu.:196270   1st Qu.:90186   JMC:12   JMC:25   1st Qu.: 103.97   1st Qu.: 50.00
##  B03    : 1   Median :197077   Median :90445   PF :11   PF : 2   Median : 193.11   Median :100.00
##  B04    : 1   Mean   :197326   Mean   :90606   XV :18   XV :15   Mean   : 213.25   Mean   : 97.05
##  B05    : 1   3rd Qu.:198115   3rd Qu.:90668                     3rd Qu.: 269.73   3rd Qu.:112.50
##  B06    : 1   Max.   :203624   Max.   :92191                     Max.   :1301.61   Max.   :200.00
##  (Other):38
##     largeur        tourniere       recRuisseau      recPhalaris     recBuisson        Berula
##  Min.   :  0.0   Min.   :0.0000   Min.   :  0.00   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 50.0   1st Qu.:0.0000   1st Qu.: 23.75   1st Qu.:0.00   1st Qu.:0.000   1st Qu.:0.0000
##  Median : 75.0   Median :0.0000   Median : 65.00   Median :1.00   Median :0.500   Median :0.0000
##  Mean   :109.7   Mean   :0.1364   Mean   : 59.66   Mean   :1.25   Mean   :1.455   Mean   :0.2273
##  3rd Qu.:150.0   3rd Qu.:0.0000   3rd Qu.: 91.25   3rd Qu.:2.00   3rd Qu.:3.000   3rd Qu.:0.0000
##  Max.   :250.0   Max.   :1.0000   Max.   :100.00   Max.   :5.00   Max.   :5.000   Max.   :1.0000
```

```
##
##    Nasturtium       Callitriche         Glyceria        Groenlandia         Veronica
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.06818   Mean   :0.06818   Mean   :0.04545   Mean   :0.04545   Mean   :0.04545
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##      Carex             Apium              temp          conduct             pH
##  Min.   :0.00000   Min.   :0.00000   Min.   :12.00   Min.   : 586.0   Min.   :7.130
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:16.60   1st Qu.: 700.0   1st Qu.:7.750
##  Median :0.00000   Median :0.00000   Median :17.40   Median : 742.0   Median :7.850
##  Mean   :0.02273   Mean   :0.06818   Mean   :17.79   Mean   : 758.4   Mean   :7.892
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:18.70   3rd Qu.: 799.0   3rd Qu.:8.150
##  Max.   :1.00000   Max.   :1.00000   Max.   :24.10   Max.   :1351.0   Max.   :8.270
##                                      NA's   :7       NA's   :7        NA's   :7
##    O2dissous       saturation       nitrates       O_Phospate         X20.C           Alcalin
##  Min.   : 5.44   Min.   :37.80   Min.   : 4.90   Min.   : 0.080   Min.   : 557.0   Min.   : 4.070
##  1st Qu.:11.00   1st Qu.:88.40   1st Qu.:32.80   1st Qu.: 0.120   1st Qu.: 607.0   1st Qu.: 5.190
##  Median :11.45   Median :89.30   Median :43.90   Median : 0.180   Median : 636.5   Median : 5.410
##  Mean   :10.97   Mean   :87.11   Mean   :41.67   Mean   : 1.138   Mean   : 710.7   Mean   : 6.079
##  3rd Qu.:11.91   3rd Qu.:95.30   3rd Qu.:45.55   3rd Qu.: 0.360   3rd Qu.: 687.8   3rd Qu.: 5.800
##  Max.   :12.07   Max.   :96.50   Max.   :92.50   Max.   :14.130   Max.   :1769.0   Max.   :17.720
##  NA's   :14      NA's   :14      NA's   :14      NA's   :14       NA's   :14       NA's   :14
##        Ca           pHCanon
##  Min.   : 93.9   Min.   :0.000
##  1st Qu.:118.8   1st Qu.:7.890
##  Median :122.9   Median :8.020
##  Mean   :119.3   Mean   :7.705
##  3rd Qu.:129.5   3rd Qu.:8.027
##  Max.   :137.9   Max.   :8.080
##  NA's   :14      NA's   :14
```

Variables quantitatives

La plupart des valeurs semblent plausibles.

On peut noter cependant qu'au moins un des tronçons de drains à une longueur beaucoup plus grande (1.3 km) que les autres (3/4 des drains font moins de 270m). Le nombre d'individus sur un tronçon de 47m (min) et de 1.3km (max) n'est pas vraiment comparable a priori. On pourrait donc étudier le nombre d'individus par mètre ou par 100 mètres de drain. On aurrait alors plus des nombres entiers ce qui exclut l'utilisation des GLM de Poisson. Une autre possibilité est d'utiliser les nombres bruts et d'inclure la longueur du drain comme variable explicative (ou comme offset) pour corriger cet effet. Pour des données binaires (pontes), c'est de toute façon la seule option. Etant donné la présence d'au moins une grande valeur il faudra envisager une transformation (log pex) pour diminuer l'importance de cette valeur. A garder en tête pour l'exploration graphique des données.

Il est également surprenant que le nombre de femelles soit beaucoup plus faible que celui des mâles. Ce point serait à éclaircir avec les personnes qui ont récolté les données. Une explication plausible est que les femmelles étant beaucoup plus difficiles à identifier que les mâles, elles n'ont pas été comptabilisées systématiquement.
On va travailler sur le nombre total d'individus

On va étudier ici l'abondance globale, on somme donc les mâles, femelles et tandems (x2) pour créer une nouvelle variable "nb" que l'on place dans la première colonne.
Etant donné les doutes sur les femelles on pourrait aussi travailler uniquement sur les mâles mais le nombre de femelles étant très faible, leur effet est sans doute peu important. (on pourrait refaire l'analyse avec et sans les femmelles pour tester la robustesse des résultats)

```
d$nb <- d$male + d$femelle + 2*d$tandem
d <- d[,c(ncol(d), 1:(ncol(d)-1))]
```

On peut voir aussi que la variable “ponte” contient beaucoup de 0 et très peu d’observations dans les autres catégories.

```
table(d$ponte)
```

```
##
##  0  1  2  4
## 38  3  2  1
```

Il semble donc raisonable de la transformer en présence /absence. On a cependant seulement 13% de “présences” ce qui n’est pas beaucoup et risque de poser des problèmes dans l’analyse.

```
d$ponte <- ifelse(d$ponte>0, 1, 0)
mean(d$ponte)
```

```
## [1] 0.1363636
```

Variables qualitatives

La variable ID a été importée comme un nombre alors qu’il s’agit plutôt d’une variable qualitative (la moyenne de ces valeurs n’a pas de sens). Par sécurité on va donc la transformer en facteur (en pratique on utilisera plus cette variable par la suite)

```
d$ID <- factor(d$ID)
```

A part les identifiants, les colonnes obs et obsrives sont les seules variables qualitatives. On voit que les données ont été récoltées principalement par deux observateurs: JMC et XV. On pourrait changer l’ordre des classes alphabétique par défaut) par odre décroissant de nombre d’observation (çà n’a pas beaucoup d’intérrêt ici mais il est souvent important de pouvoir réordonner les niveaux d’un facteur).

```
levels(d$obs)
```

```
## [1] "DT"  "JMC" "PF"  "XV"
```

```
d$obs <- factor(d$obs, levels = c("JMC", "XV", "PF", "DT"))
d$obsrives <- factor(d$obsrives, levels = c("JMC", "XV", "PF", "DT"))
```

Variables binaires

Les variables binaires (présence/absence ici) sont des variables qualitatives mais elles peuvent être sans problème encodée sous la forme de variables numériques composées de 0 ou 1 comme c’est le cas ici (sans les transformer en facteur).
Dans ce cas la moyenne représente la proportion des observations avec des présences. On constate que pour plusieurs de ces variables binaires la proportion de “présences” est très faible (<10%) ce qui peut poser des problèmes dans leur utilisation comme variable explicative (trop peu de variation). C’est le cas par exemple pour toutes les plantes aquatiques à part le genre *Berula*. Il faut garder en tête cette information lors de l’exploration des données pour voir comment on va traiter ce problème.

Variables ordinales

recPhalaris et recBuisson sont des variables ordinales à 6 classes de % de couverture (échelle de Braun-Blanquet). On pourrait conserver ces classes et les traiter comme une variable qualitative (à transformer en facteur). Une approche fréquente est au contraire de transformer ce genre d'échelle en variable continue en attribuant à chaque classe la valeur du milieu de la classe (NB : la définition de ces classes varient parfois d'un groupe de personne à l'autre, ici on s'est inspiré d'un document méthodologique écrit par la même équipe):

- 0 = 0% reste 0
- 1 = >0% et <5% devient 2.5%
- 2 = 5-25% devient 15%,
- 3 = 25-50% devient 37.5%,
- 4 = 50-75% devient 62.5%
- 5 = 75-100% devient 87.5%

Si on veut les transformer en variables qualitatives il suffit de procéder comme suit. NB pour recPhalaris la classe 4 manque (aucune observation), on spécifie explicitement les différentes valeurs de niveau possibles (`levels = 0:5`). Ce qui n'est pas nécessaire pour resBuisson

```
d$recPhalaris <- factor(d$recPhalaris, levels = 0:5)
d$recBuisson <- factor(d$recBuisson)
```

Un problème avec cette approche est que certaines catégories sont très peu représentées. Il va donc être très difficile d'estimer une différence moyenne entre ces catégories rares et les autres (et surtoût d'estimer la précision de telles différences) :

```
table(d$recPhalaris) # Nombre d'observations pour chaque classe
```

```
##
##  0  1  2  3  4  5
## 15 14  6  8  0  1
```

```
table(d$recBuisson)
```

```
##
##  0  1  2  3  4  5
## 22  3  3 12  1  3
```

On peut donc les transformer en variables continues.

```
# On fait une copie des deux variables dans des vecteurs en dehors du data.frame
recPhalarisPct <- d$recPhalaris
recBuissonPct <- d$recBuisson

# On change les valeurs des niveaux
levels(recPhalarisPct)
```

```
## [1] "0" "1" "2" "3" "4" "5"
```

```
levels(recPhalarisPct) <- c("0", "2.5", "15", "37.5", "62.5", "87.5")
levels(recBuissonPct) <- c("0", "2.5", "15", "37.5", "62.5", "87.5")

# On veut ensuite les transformer en variables numériques.
# Attention ceci ne donne pas le résultat escompté :
as.numeric(recBuissonPct)
```

```
##  [1] 4 4 6 4 4 4 4 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 4 1 6 4 5 4 4 4 1 3 1 1 3 1 1 1 1 2 4 6 3
```

```
# Il faut d'abord passer par un format texte puis numérique :
as.numeric(as.character(recBuissonPct))
```

```
##  [1] 37.5 37.5 87.5 37.5 37.5 37.5 37.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  2.5
## [20]  0.0  0.0  2.5  0.0 37.5  0.0 87.5 37.5 62.5 37.5 37.5 37.5  0.0 15.0  0.0  0.0 15.0  0.0  0.0
## [39]  0.0  0.0  2.5 37.5 87.5 15.0
```

```
recPhalarisPct <- as.numeric(as.character(recPhalarisPct))
recBuissonPct <- as.numeric(as.character(recBuissonPct))

# NB on aurait pu arriver au même résultat en remplaçant une valeur à la fois :
tmp <- as.character(d$recPhalaris)
tmp[tmp==1] <- "2.5"
tmp[tmp==2] <- "15"
tmp[tmp==3] <- "37.5"
tmp[tmp==4] <- "62.5"
tmp[tmp==5] <- "87.5"
tmp <- as.numeric(tmp)
tmp
```

```
##  [1]  2.5  0.0  0.0  2.5  2.5 15.0  2.5  0.0  0.0  0.0  0.0 87.5  2.5 37.5  0.0 37.5  2.5  2.5  2.5
## [20]  0.0  0.0  0.0  2.5 37.5 37.5  0.0  0.0  2.5  0.0  2.5  2.5 37.5 15.0  0.0 37.5  0.0 15.0 37.5
## [39]  2.5 15.0 37.5 15.0  2.5 15.0
```

```
recPhalarisPct
```

```
##  [1]  2.5  0.0  0.0  2.5  2.5 15.0  2.5  0.0  0.0  0.0  0.0 87.5  2.5 37.5  0.0 37.5  2.5  2.5  2.5
## [20]  0.0  0.0  0.0  2.5 37.5 37.5  0.0  0.0  2.5  0.0  2.5  2.5 37.5 15.0  0.0 37.5  0.0 15.0 37.5
## [39]  2.5 15.0 37.5 15.0  2.5 15.0
```

Enfin, plutôt que d'utiliser des variables continues on pourrait grouper certaines classes de façon à ce qu'il y ait au moins quelques répétitions pour chaque catégorie (dans le cas le plus extrême on pourrait travailler simplement en présence/absence).

Dans le cas présent travailler avec les variables continues serait sans doute idéal en première approche. Pour des raisons didactiques on va cependant travailler avec les variables qualitatives pour pouvoir illustrer les particularités de telles variables.

```
table(d$recPhalaris)
```

```
##
##  0  1  2  3  4  5
## 15 14  6  8  0  1
```

```
table(d$recBuisson)
```

```
##
##  0  1  2  3  4  5
## 22  3  3 12  1  3
```

```
levels(d$recPhalaris) <- c("0%", "<5%", "5-25%", ">25%", ">25%", ">25%")
levels(d$recBuisson) <- c("0", "<25%", "<25%", "25-50%", ">50%", ">50%")

table(d$recPhalaris)
```

```
##
##    0%   <5% 5-25%  >25%
##    15    14     6     9
```

```
table(d$recBuisson)
```

```
##
##      0   <25% 25-50%   >50%
##     22      6     12      4
```

Valeurs manquantes NA

On remarque qu'il y a des valeurs manquantes dans les dernières variables mais elles ne sont pas toutes sur les mêmes lignes. Pour rappel dans une régression/GLM, le programme élimine toutes les lignes où il y a au moins une valeur manquante. Si on devait utiliser toutes ces variables on perdrait presque la moitié des lignes (on passerait de 44 observation à 25 utilisables) :

```
dim(d)
```

```
## [1] 44 38
```

```
dim(na.omit(d)) # na.omit élimine les lignes où il y a au moins une valeur manquante
```

```
## [1] 25 38
```

Les 8 dernières variables physico-chimiques ont en fait été récoltées seulement à 12 endroits (prélèvement d'eau en hiver + analyse en labo). Elles n'ont vraisemblablement pas été collectées comme variables explicatives à proprement parler mais comme statistiques descriptives supplémentaires pouvant aider à caractériser les sites en général. On ne les utilisera donc pas dans les modèles.

```
unique(d[, (ncol(d)-7):ncol(d)])
```

```
##    O2dissous saturation nitrates O_Phospate X20.C Alcalin    Ca pHCanon
## 1      11.45       88.4     43.9       0.36   557    4.07  93.9    8.02
## 8      11.00       92.7     33.3       0.29   607    5.60 118.8    8.02
## 10        NA         NA       NA         NA    NA      NA    NA      NA
## 12     12.07       96.5     32.8       0.12   626    5.80 122.9    8.08
## 17     12.07       96.5     32.8       0.12   626    5.80 122.9    0.00
## 20      5.44       37.8      4.9      14.13  1769   17.72 131.2    8.07
## 23     11.29       89.7     22.4       0.18   696    6.46 137.9    7.88
## 26     11.41       93.3     39.7       0.08   647    5.37 123.1    7.89
## 27     11.91       95.3     49.2       0.10   661    5.22 124.5    7.99
## 30     12.00       96.0     45.1       0.10   650    5.29 122.7    8.03
## 32     11.19       88.9     45.7       0.21   663    5.35 123.5    8.02
## 38     10.01       77.9     48.0       0.24   647    5.19 120.4    7.89
## 39     10.40       86.9     92.5       0.15   750    5.41 137.2    7.70
```

```
# stockage des 7 dernières variables puis suppression du jeu de données principal
labo <- d[, c(7,(ncol(d)-7):ncol(d))]
d <- d[, -((ncol(d)-7):ncol(d))]
```

Avec les variables pH, conduct et temp on perd 7 lignes sur 44. D'après les métadonnées ce sont des drains où il n'y avait pas d'eau libre disponible. On voit que malgré l'absence d'eau il y avait des libellules, y compris des acouplements (tandems) mais par contre aucun comportement de ponte... Il faut cependant garder en tête que ces mesures ont été faites en 2007 alors que les comptages ont été faits en 2006. Il y avait donc peut-être de l'eau sur ces tronçons en 2006.

```r
d[is.na(d$pH), c("male", "femelle", "tandem", "ponte", "temp", "conduct", "pH")]
```

```
##    male femelle tandem ponte temp conduct pH
## 17   13       0      5     0   NA      NA NA
## 18   29       0      4     0   NA      NA NA
## 20    1       0      0     0   NA      NA NA
## 22    2       0      3     0   NA      NA NA
## 39   17       1      3     0   NA      NA NA
## 40   50       3     12     0   NA      NA NA
## 41    4       0      1     0   NA      NA NA
```

Il faut donc garder en tête que si on inclut ces variables dans l'analyse elle causeront la perte de 16% des données (lignes).
Une autre option serait de ne pas utiliser ces variables ce qui permet de garder toutes les lignes du jeu de données.
NB : les fait de suprimer des lignes à cause des valeurs manquantes n'est pas anodin (même si souvent on ne peut pas faire autrement...). En particulier cela rend l'échantillonnage moins aléatoire (chaque drain n'a pas la même probabilité de se faire échantillonner que les autres). En effet souvent les valeurs manquantes ne sont pas là par hasard... On le voit bien ici où les valeurs manquantes indiquent en fait la présence ou l'absence d'eau.

```
Q05 - En fonction de ce que vous savez sur la manière dont les jeux de données GBIF
ont été récoltés décrivez les problèmes et/ou les biais potentiels dans votre jeu
de données et les conséquences possibles pour la question que vous étudiez.
Est-il possible de mitiger certains de ces problèmes en modifiant le jeu de données
ou en modifiant votre manière d'analyser les données ?
```

```
Q06 - Posez-vous les mêmes questions sur le jeu de données environnemental.
```

NB : ces questions ont déjà été explorées plus haut. On a souligné le fait que l'abondance des adultes à un endroit n'était pas obligatoirement lié à l'abondance des larves même si çà semble plausible. On a proposé d'examiner en complément le comportement de ponte qui indique vraissemblablement des conditions favorables (du moins si les adultes sont capables de choisir de manière optimale leurs sites de pontes).

On a souligné aussi les limites des variables physico-chimiques. Une partie a été éliminée du jeu de données. Pour les 3 restantes (pH, conductivité, température) il n'y a pas de soultion si ce n'est de rester très prudent sur les interprétation des résultats. Des données sur l'heure de mesure aurait peut-être permis de corriger les données de température.

Exploration des données et choix des variables explicatives

Exploration des variables dépendantes

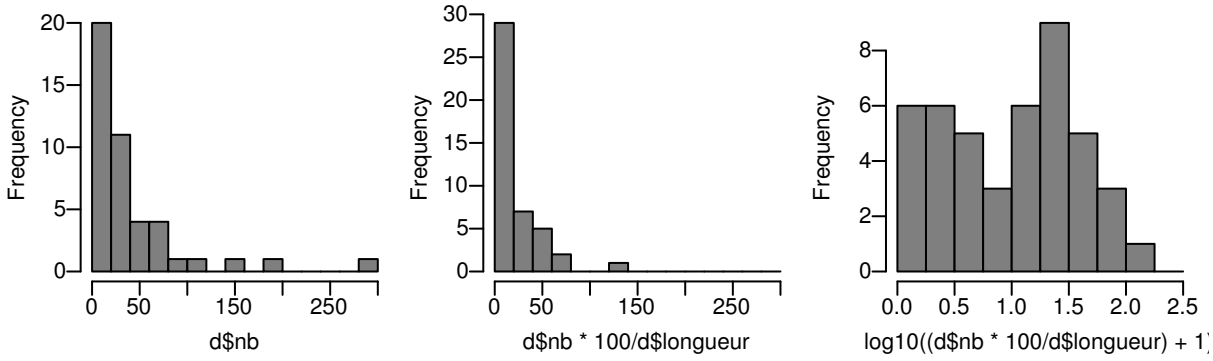
> Q07 - Explorez les données espèces (variables dépendantes/réponses) et commentez vos observations.

On a que deux variables dépendantes ici : le nombre d'adultes et la présence de comportement de ponte.

Le nombre d'adultes a une distribution fortement asymétrique ce qui est attendu avec ce genre de données. On voit que lorsqu'on standardise par la longueur du drain la distribution est moins asymétrique (les très grands nombres se trouvent sur des drains plus longs). Une transformation log10(x+1) rend la distribution à peu près normale (symétrique) avec cependant un clair excès de petites valeurs qui donne une impression de distribution bimodale.

L'argument `breaks` de la fonction `hist` permet de définir la manière dont les valeurs sont divisées en classes avant de compter le nombre d'observations. La fonction `seq` est utilisée pour générer une séquence de nombre par exemple entre 0 et 300 par pas de 20.

```
# dev.new(16/2.54, 5/2.54)
par(mfrow = c(1,3), mar = c(3,3,1,0.5), mgp = c(1.8,0.5,0), cex = 0.7, las = 1)
hist(d$nb, main = "", breaks = seq(0, 300, 20), col = "gray50")
hist(d$nb*100/d$longueur, main = "",  breaks = seq(0, 300, 20), col = "gray50")
hist(log10((d$nb*100/d$longueur)+1), main = "",
     breaks = seq(0, 2.5, 0.25), col = "gray50")
```



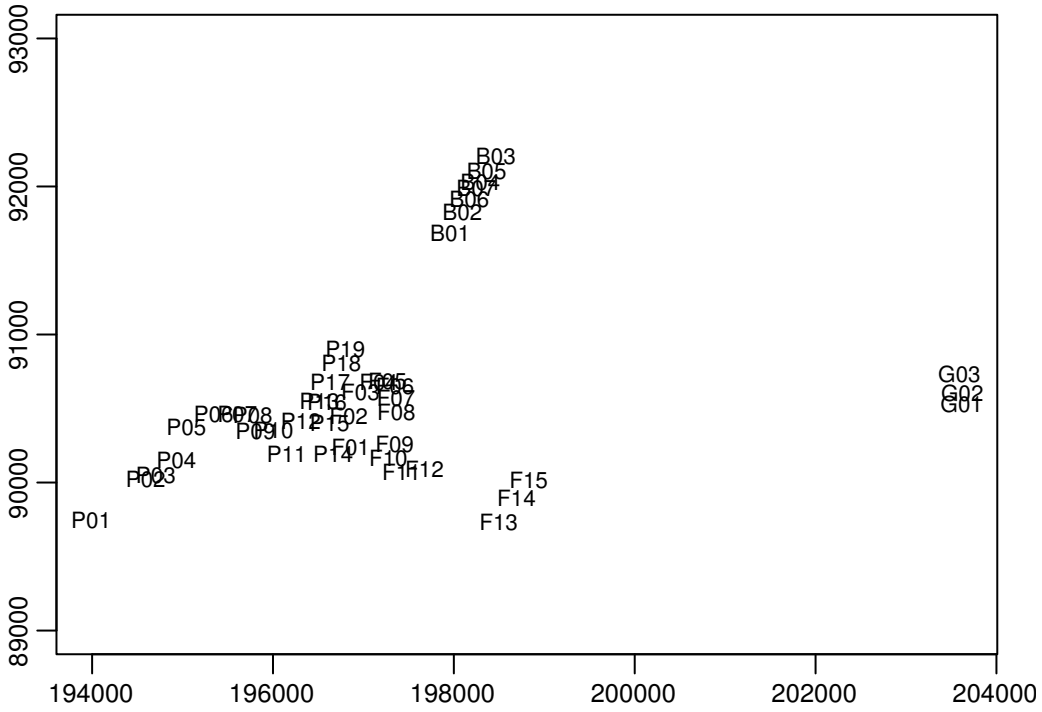
Cartes simples

Il peut être utile de visualiser la distribution spatiale des relevés.

Voici une carte ultra basique qui donne déjà l'essentiel de l'information. On voit bien qu'il y a des groupes d'observations spatialement séparées et que la première lettre de l'identifiant définit des tronçons proches. On voit aussi qu'on travaille sur une petite aire géographique de 2.5 x 10 km environs (les unités sont en mètres). NB: dans une carte plus 'propre" on suprime les axes et on ajoute une échelle.

Avec l'argument `type='n'`, on trace les axes du graphique mais aucune donnée. On ajoute ensuite les identifiants des drains avec la fonction `text`. Les arguments cex permettent de changer la taille des caractères et symbôles. Les paramètres graphiques (`par()`) `mar` et `mgp` définissent la taille des marges et la position des titres et étiquettes des axes. `dev.new` permet d'ouvrir une fenêtre graphique au dimensions voulues (14cm de large, 10cm de haut).

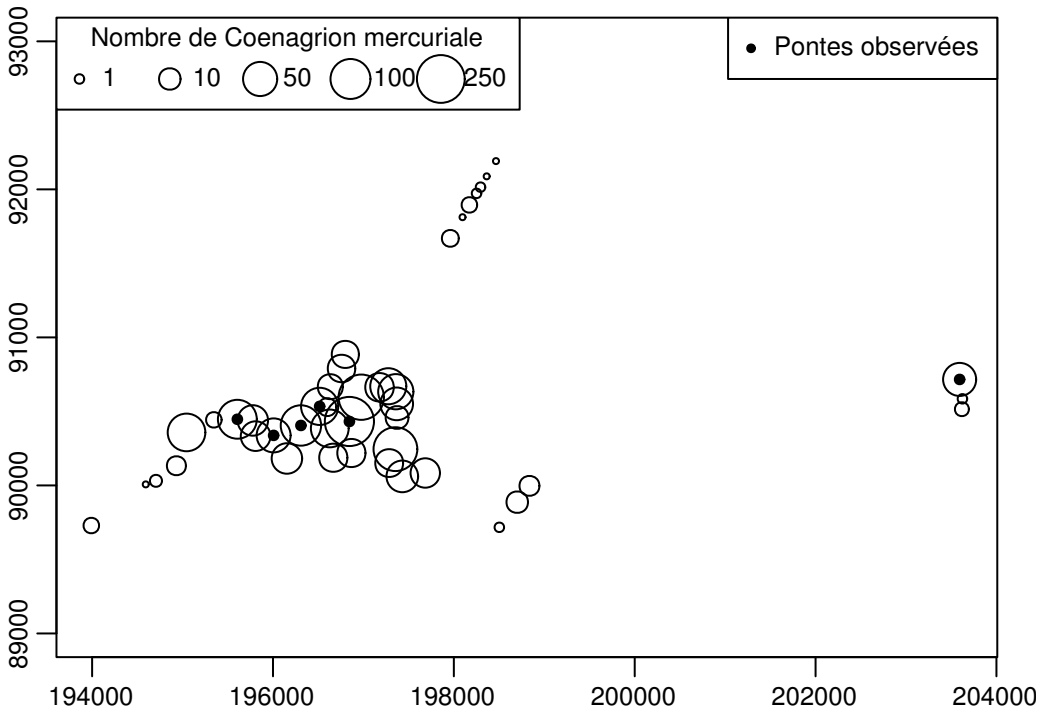
```
# dev.new(14/2.54, 10/2.54)
par(mar = c(2,2,1,1), mgp = c(1.5, 0.5, 0))
plot(x = d$x, y = d$y, ylim = c(89000, 93000),
     type="n", xlab = "", ylab = "", cex.axis=0.8)
text(x = d$x, y = d$y, labels = as.character(d$code2), cex=0.7)
```



On peut aussi utilement représenter l'abondance des libellules et les drains où des pontes ont été observés. Le résultat n'est pas très joli mais permet de voir ce qu'on a besoin et est suffisant pour l'exploration des données. On voit bien qu'il y a un noyeau de population là où les drains sont les plus denses. Il y a cependant un drain à l'extrème est avec un nombre relativement important et un comportement de ponte observé.

A la place de la fonction `text`, on utilise la fonction `points` qui permet de placer des points/symboles sur le graphique. L'argument `pch` ("point character") permet de choisir le type de symbôle (voir `?points` pour les symbôles possibles). L'argument `cex` définit la taille des symbôles. Dans un cas on lui donne comme valeur le nombre d'individus observés (avec transformation log et multiplication par 0.6 pour limiter les écarts de taille). La taille du point sera donc proportionnelle au nombre d'individus observés à cet emplacement.
Les fonctions `legend` permettent d'ajouter les légendes et disposent de très nombreux agruments (voir l'aide).

```r
# dev.new(14/2.54, 10/2.54)
par(mar = c(2,2,1,1), mgp = c(1.5, 0.5, 0))
plot(x = d$x, y = d$y, ylim = c(89000, 93000),
     type="n", xlab = "", ylab = "", cex.axis=0.8)

points(x = d$x, y = d$y, cex= log(d$nb+2)*0.6, pch = 1)
points(x = d[d$ponte == 1, "x"], y = d[d$ponte == 1, "y"], cex= 1, pch = 20)

legend("topright",  pch = 20, horiz = TRUE, cex = 0.8,
       legend = "Pontes observées")
legend("topleft",  pch = 1, horiz = TRUE, cex = 0.8,
       title = "Nombre de Coenagrion mercuriale",
       pt.cex= log(c(1, 10, 50, 100, 250)+2)*0.6,
       legend = as.character(c(1, 10, 50, 100, 250)))
```



Cartes plus avancées

Il existe de nombreuses possibilités de cartographie dans R. Typiquement on veut placer un raster (photo aérienne, plan de rue sous forme d'image disponibles sur Internet) en arrière plan pour pouvoir visualiser les données dans leur contexte géographique.
On peut aussi charger des cartes en format vectoriel (par exemple shapefile de la Belgique) pour montrer où se trouvent les observations de manière plus générale.

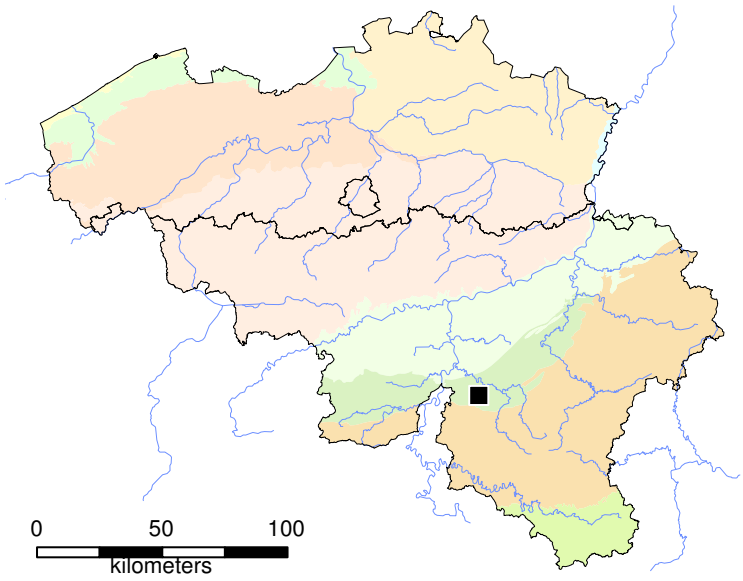
Pour pouvoir utiliser ces options plus avancées il est nécessaire d'avoir installé sur l'ordinateur divers logiciels libres de cartographie (GDAL, PROJ4,...). Le plus simple est sans doute d'installer un SIG libre sur votre ordinateur comme QGIS qui installera automatiquement toutes ces librairies extérieures à R.

Une difficulté des données cartographiques sont les diverses projections utilisées pour afficher le globe terrestre de forme sphéroïde en 2 dimensions. Lorsqu'on utilise des données cartographiques d'origines différentes elles ont en général des projections (Coordinate reference Systems - CRS) différentes. Il faut alors les reprojeter toutes dans le même système de coordonnées sans quoi les données ne s'alignent pas correctement.

Situation générale avec cartes vectorielles .

```
GISfolder <- "/home/gilles/stats/Formation_R_stats/UCL_LBOE2121/GBIF/data/Spatial"

Regions <- readOGR(GISfolder, "regions", p4s = "+init=epsg:31370", verbose = FALSE)
NaturalRegions <- readOGR(GISfolder, "Regions_Naturelles", p4s = "+init=epsg:31370",
                          , verbose = FALSE)
Rivers <- readOGR(GISfolder, "Rivieres", p4s = "+init=epsg:31370", verbose = FALSE)

# choose colors for natural regions
RGB <- as.character(c("#E5FFFF", "#FFFFCC", "#E5FFD9", "#E5FFD9", "#E5FFD9", "#FFE5CC",
                      "#FFE5CC", "#FFEEE1", "#FFEEE1", "#DAF2C2", "#F2FFE5", "#F2FFE5",
                      "#DAF2C2", "#F2FFE5", "#FAE1AF","#E1FAAF", "#FAE1AF", "#FFFFCC",
                      "#FFFFCC", "#FFF2CC"))
NaturalRegions@data$RGB <- RGB
```

```
# dev.new(10/2.54, 8/2.54)
par(mar = c(0,0,0,0))
plot(NaturalRegions, lty = 0,  col = NaturalRegions@data$RGB)
plot(Rivers, col = "#6D90FF", add = TRUE, lwd = 0.5)
plot(Regions, add = TRUE, lwd = 0.5, border = "black")
x <- mean(d$x) ; y <- mean(d$y) # centroïde des points
points(x, y, pch = 22, col = "white", bg = "black", cex = 1.5)

# add a scale - use locator() to find a nice position on the map
raster:::scalebar(d = 100000, xy = c(21000,26000), type = 'bar',
                  divs = 4, below = 'kilometers', cex = 0.7,
                  label = c(0,50, 100))
```




```
# GISTools:::north.arrow(xb= 34914, yb = 80000, len=3000, lab="N")  # North arrow
```

Situation locale avec plan et photo aérienne .

On va ici par exemple les visualiser sur un fond de carte OpenStreetMap.
Il faut dans un premier temps définir les limites de la zone que l'on veut représenter en longitude latitude. Pour cela, on va transformer les Lambert en Longitude - latitude faire un graphique de ces coordonnées et récupérer les extrêmes.

```
d2 <- d # copie du dataset
coordinates(d2) <- ~x+y # transformation en objet spatial
proj4string(d2) <- CRS("+init=epsg:31370") # définition du syst. de coo = lambert belge 72
d2 <- spTransform(d2, CRS("+proj=longlat +datum=WGS84")) # transformation en long lat
```

```
# On fait un graphique juste pour extraire les limites utiles en long lat
plot(as.matrix(coordinates(d2)), asp=NA) # asp=1 pour avoir une échelle égale en x et y
```

```
upleft <- par("usr")[c(4,1)]
lowright <- par("usr")[c(3,2)]

# Ou on défini à la main les coins de la carte
tmp <- as.matrix(coordinates(d2))
tmp <- apply(tmp, 2, range)
pct15 <- (tmp[2,]-tmp[1,])* c(0.05, 0.75)
tmp[1,] <- tmp[1,] - pct15
tmp[2,] <- tmp[2,] + pct15
upleft <- c(tmp[2,2], tmp[1,1])
lowright <- c(tmp[1,2], tmp[2,1])
```

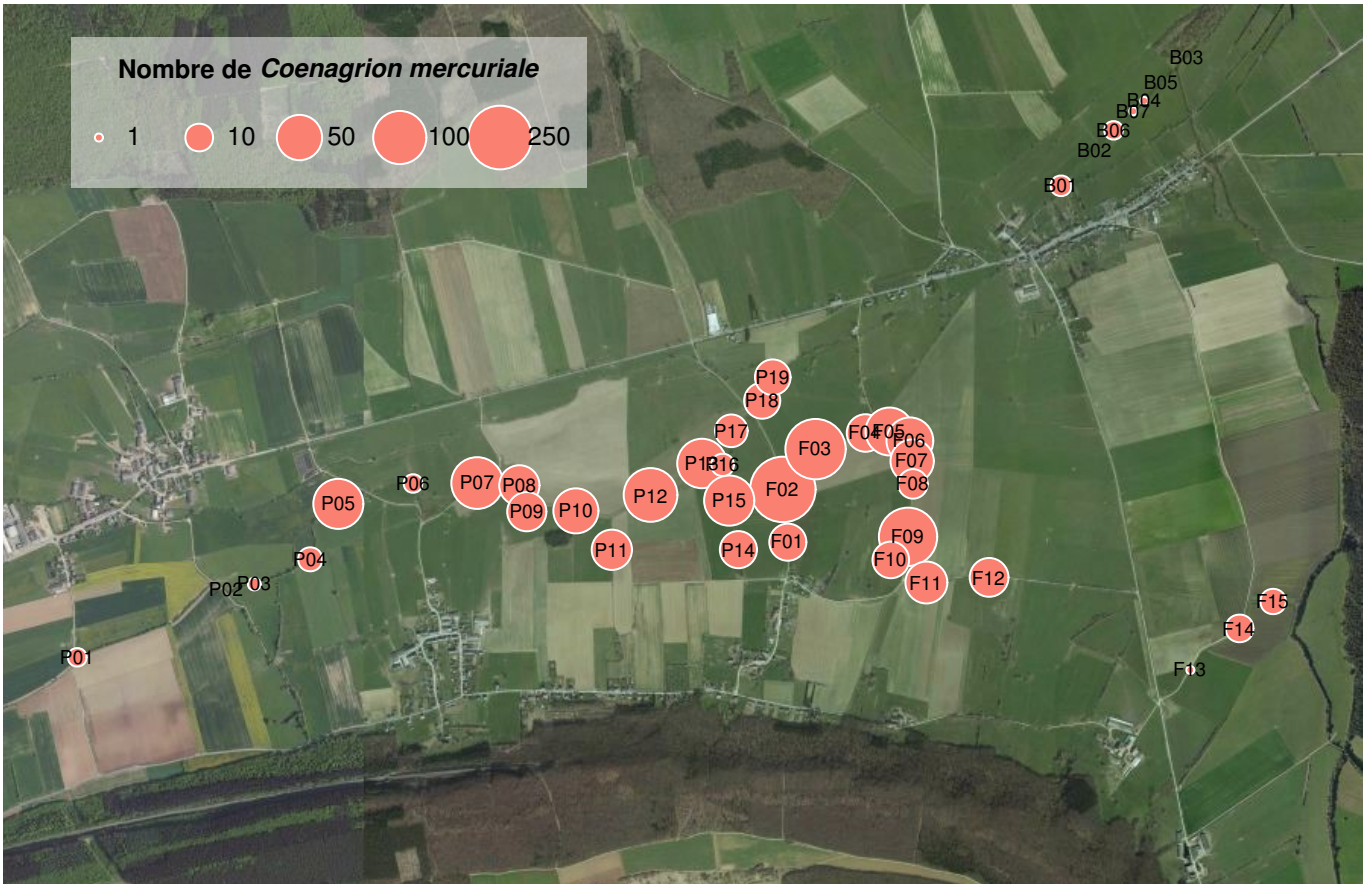
On télécharge la carte depuis OSM (nécessite une connection !). Les fonds cartos sont dans une projection Mercator particulière. On reprojette le fond raster vers le système Lambert Belge 1972 qui est celui de nos données, çà prend un peu de temps à calculer...

```
# dev.new(16/2.54, 8/2.54)
library(OpenStreetMap)
map <- openmap(upleft,lowright, type= "osm")
maplbrt <- openproj(map, projection = "+init=epsg:31370")
plot(maplbrt)
text(x = d$x, y = d$y, labels = as.character(d$code2), cex=0.6)
```

Il existe de nombreux autres fonds carto disponibles. Ici on va par exemple visualiser les données sur des photos aériennes (bing) pour un sous ensemble des données afin d'avoir une échelle plus lisible (ie sans les tronçons G01 - G03 à l'extrême Est). Plutôt que de transformer la carte vers le Lambert Belge, on va ici transformer les données vers la projection Mercator OSM, ce qui est plus rapide... La fonction launchMapHelper() permet de récupérer interactivement les coordonnées des coins de la carte.

La carte permet de voir qu'on est principalement en milieu agricole et dans des zones de cultures.

```
# dev.new(18/2.54, 12/2.54)
# launchMapHelper()
bing = openmap(c(lat= 50.140809906482114,   lon= 4.979639053344727),
               c(lat= 50.10885475929136,   lon= 5.05671501159668),
               type= "bing")
d2 <- spTransform(d2,osm()) # reprojection du jeu de données spatial créé plus haut
plot(bing)
points(coordinates(d2), pch = 21, cex= log(d2$nb+1)*0.8, bg = "salmon", col = "white")
text(coordinates(d2), labels = as.character(d2$code2), cex=0.6)

legend("topleft", pch = 21, pt.bg = "salmon", col = "white", bty = "o", bg = rgb(1,1,1, 0.5),
       cex = 0.8, horiz = TRUE, inset = 0.05, box.lty = 0,  x.intersp = 1,
       title = expression(paste(bold("Nombre de "), bolditalic("Coenagrion mercuriale"))),
       pt.cex= log(c(1, 10, 50, 100, 250)+1)*0.8,
       legend = as.character(c(1, 10, 50, 100, 250)))
```



Cartes interactives en ligne .

NB : Il est aussi très facile de créer des cartes interactives (ea possibilité de zoomer, cliquer pour afficher des informations,...) avec R et leaflet.
NB : le code n'est pas exécuté ici. Nécessite un navigateur internet pour naviguer dans la carte interactive. Un fichier "Mercuriale_Leaflet_map.html" est fourni en annexe avec la carte.

```
library(leaflet)
d2 <- spTransform(d2,CRS("+proj=longlat +datum=WGS84")) # reprojection vers long/lat


pal = colorBin('YlOrRd', domain = range(d$nb),
               bins = c(0, 1, 5, 10, 25, 50, 100, 200, 300))

m <- leaflet(d2) %>%
    addProviderTiles("Esri.WorldImagery", group = "Satelite") %>%
    addProviderTiles("OpenTopoMap", group = "Topo") %>%
    addTiles(group = "OSM") %>%
    addCircles(
        lng=coordinates(d2)[,1], lat=coordinates(d2)[,2],
        popup= paste0(d2$code2, ": ", d2$nb, " indiv."),
        radius = log(d2$nb+2)*20,
        color = ~pal(d2$nb),
        stroke = TRUE, weight = 2, fillOpacity = 0.5,
        group = "Abundance"
    )  %>%
    addCircles(
        lng=coordinates(d2[d2$ponte>0,])[,1], lat=coordinates(d2[d2$ponte>0,])[,2],
        popup= "Egg laying behavior",
        radius = 40,
        color = "black",
        stroke = FALSE, weight = 2, fillOpacity = 0.8,
        group = "Egg laying"
    )  %>%
    addLegend("bottomright", pal = pal, values = d2$nb,
              title = "C. mercuriale",
              opacity = 1, labFormat = labelFormat(suffix = ' indiv.')
    ) %>%
    addLayersControl(
        baseGroups = c("Satelite", "OSM", "Topo"),
        overlayGroups = c("Abundance", "Egg laying"),
        options = layersControlOptions(collapsed = FALSE)
    ) %>%
    setView( 5.030931, 50.13, zoom = 14)
m
```

Exploration des variables explicatives

> Q08 - Explorez les variables explicatives et commentez vos observations.
>
> Vous devez au minimum vous faire une idée des corrélations entre variables explicatives et une idée de leur distribution (histogrammes, density plots) et prendre les dispositions nécessaires en cas de problèmes potentiels pour votre futur modèle.

Plantes auquatiques

On a vu précédemment qu'il y a plusieurs plantes aquatiques ou semi-aquatiques (données de présence/absence) qui sont présntes sur très peu de drains. Le manque de variabilité dans ces variables explicatives (presque uniquement des 0) rend leur utilisation délicate et il est serait sans doute judicieux de les regrouper d'une manière ou d'une autre. Ces plantes aquatiques peuvent être vues soit comme des sites de ponte etde vie pour les larves soit comme de splantes indicatrices des caractéritiques de chaque drain. Selon le point de vue, le regroupement se fera différemment.

Explorons plus en détail le contenu de ces variables.

On peut voir que à part le genre *Berula*, la plupart des plantes ne sont présentes que sur 1 à 3 drains :

```
plants <- c( "Berula", "Nasturtium", "Callitriche", "Glyceria",
             "Groenlandia", "Veronica", "Carex", "Apium")
apply(d[,plants], 2, sum)
```

```
##      Berula  Nasturtium Callitriche    Glyceria Groenlandia    Veronica       Carex       Apium
##          10           3           3           2           2           2           1           3
```

La plupart des drains ne contiennent que 0 ou 1 plante et seuls deux drains contiennent deux plantes :

```
table(apply(d[,plants], 1, sum))
```

```
##
##  0  1  2
## 20 22  2
```

Un drain contient à la fois des *Berula* et des *Glyceria* et un autre contient la fois des *Callitriche* et *Groenlandia*.

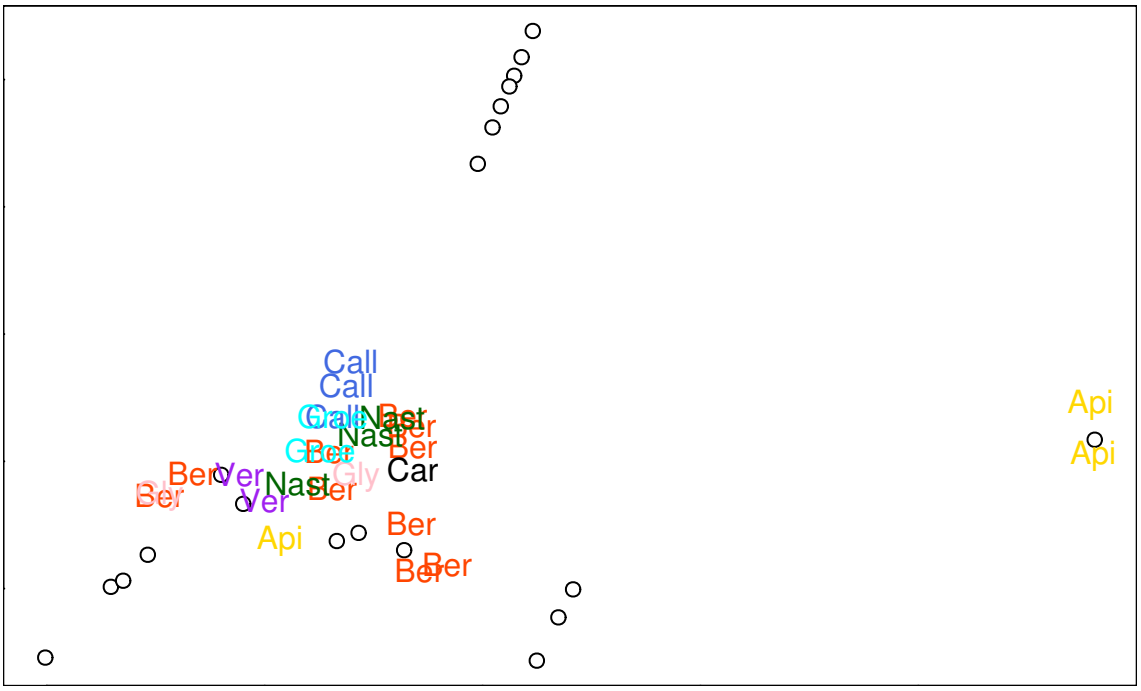
```
d[,plants][apply(d[,plants], 1, sum)>1,]
```

```
##    Berula Nasturtium Callitriche Glyceria Groenlandia Veronica Carex Apium
## 30      1          0           0        1           0        0     0     0
## 42      0          0           1        0           1        0     0     0
```

On peut également visualiser la distribution de ces plantes sur une carte minimaliste. On voit que des tronçons adjacents abritent souvent des genre de plantes identiques.

```
# dev.new(width = 15/2.54, height = 9/2.54)
par(mar = c(0,0,0,0))
prcex <- 1
plot(y ~ x, data = d[rowSums(d[,plants]) ==0,], pch = 21, bg = c("white"))
text(y ~ x, data = d[d$Berula==1,], labels = c("Ber"), cex = prcex, col = "orangered")
text(y ~ x, data = d[d$Nasturtium==1,], labels = c("Nast"), cex = prcex, col = "darkgreen")
text(y ~ x, data = d[d$Callitriche==1,], labels = c("Call"), cex = prcex, col = "royalblue")
text(y ~ x, data = d[d$Apium==1,], labels = c("Api"), cex = prcex, col = "gold")
```

```
text(y ~ x, data = d[d$Glyceria==1,], labels = c("Gly"), cex = prcex, col = "pink")
text(y ~ x, data = d[d$Groenlandia==1,], labels = c("Groe"), cex = prcex, col = "cyan")
text(y ~ x, data = d[d$Veronica==1,], labels = c("Ver"), cex = prcex, col = "purple")
text(y ~ x, data = d[d$Carex==1,], labels = c("Car"), cex = prcex)
```



Si ces plantes sont considérées comme des sites de pontes et de vie pour les larves, le mieux est sans doute de toutes les regrouper dans une seule variable "aquaplants" indiquant la présnece/absence de pplantes auqaitiques. On a alors une variable bien balancée avec presque autant d'absences que de présences.

```
d$aquaplants <- apply(d[,plants], 1, sum)
d$aquaplants <- ifelse(d$aquaplants >= 1, 1, 0)
```

On aurrait pu aussi envisager un regrouppement par gfamille de plantes (mais ici à part les genre Berula et Apium qui sont des Apiaceae, tous les autres genres sont issus de familles différentes) ou par type morphologique de plante.

On peut aussi examiner si certaines plantes ne sont pas des plantes indicatrices (par exemple d'acididté, d'eutrophisation,...).
On peut consulter par exemple les indices d'Ellenberg de ces plantes (NB on affiche pas ici les Carex ni les véroniques qui sont très nombreux et qui ont une écologie très diversifiée).
Les indices les plus intéressants a priori ici sont les indices R (acidité) et N (fertilité - présence de nutriments). On voit par exemple que l'indice N couvre une large gamme de valeurs entre 2 et 9.
Cette information semble cependant difficilement exploitable ici car on ne dispose d'identifications que jusqu'au genre or la variabilité des indices au sein d'un même genre est assez grande. Il est probable que si les plantes avaient été identifiées jusqu'à l'espèce les diffrérences entre indices auraient été moins marquées.

```
ellenb <- read.table("data/ellenberg.txt", sep = "\t", header = TRUE, dec = ",")
colnames(ellenb) <- gsub("Ellenberg_", "", colnames(ellenb))
pattern <- do.call(paste, c(as.list( c( "Berula", "Nasturtium", "Callitriche", "Glyceria",
                                        "Groenlandia", "Apium")), sep = "|"))
pander(ellenb[grep(pattern, ellenb[,1]),])
```

	Taxprio_final	L	T	K	F	R	N
109	Apium inundatum	7	6	2	10	NA	2

	Taxprio_final	L	T	K	F	R	N
110	Apium nodiflorum	7	8	3	10	NA	6
111	Apium repens	9	6	3	7	7	7
184	Berula erecta	8	6	3	10	8	6
241	Callitriche brutia	8	6	3	10	NA	5
242	Callitriche hamulata	8	4	2	10	6	4
243	Callitriche obtusangula	8	6	NA	11	7	7
244	Callitriche palustris	6	NA	NA	11	5	4
245	Callitriche platycarpa	7	6	2	11	7	7
246	Callitriche stagnalis	6	5	NA	10	6	4
247	Callitriche truncata subsp.occidentalis	NA	NA	NA	NA	NA	NA
718	Glyceria declinata	5	6	2	8	6	5
719	Glyceria fluitans	7	NA	3	9	NA	7
720	Glyceria maxima	9	5	NA	10	8	9
721	Glyceria notata	8	5	3	10	8	8
722	Glyceria x pedicellata	NA	NA	NA	NA	NA	NA
728	Groenlandia densa	8	6	2	12	8	5
1036	Nasturtium microphyllum	NA	NA	NA	NA	NA	NA
1037	Nasturtium officinale	7	NA	3	10	7	7

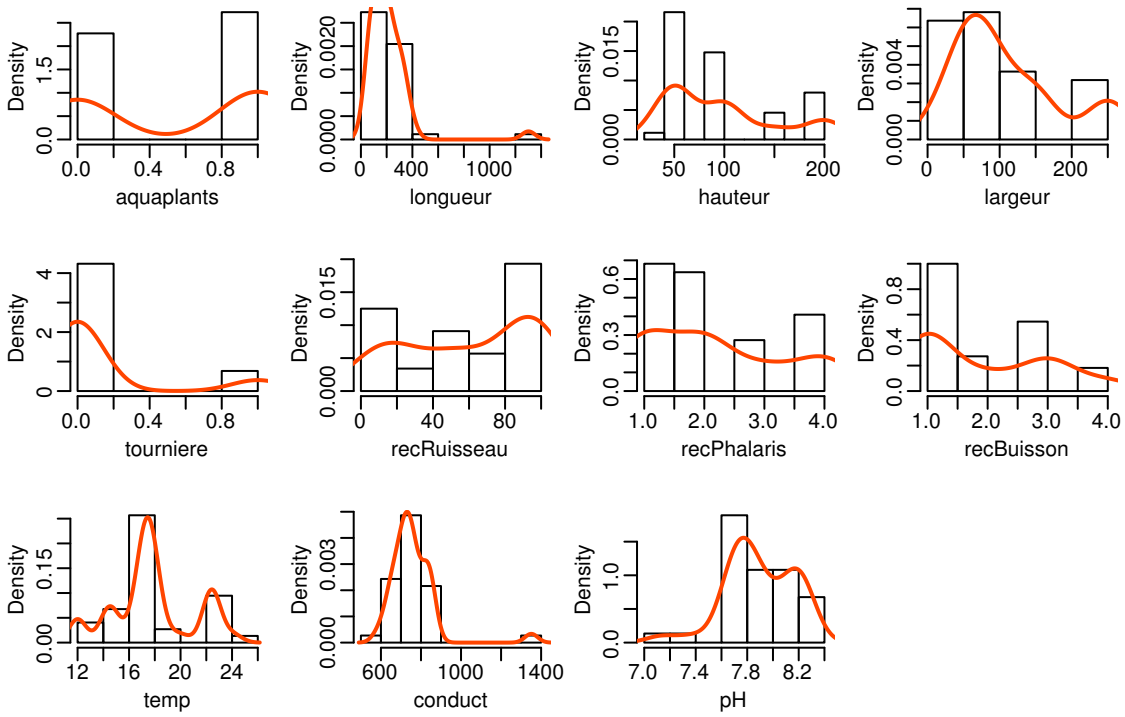
Distribution des variables explicatives

On va mainteant explorer les relations entre les variables et leur distribution. On rajoute au jeu de données les variables de recouvrement (Buisson et Phalaris) quantitatives créées précédement ainsi que le nombre d'individus par 100m de ruisseau. On crée aussi un objet "vars" qui contient le nom des variables qui nous intéressent ici (suite à l'exploration des données réalisée précédemment). après une tranformation log.

```
d$recPhalarisPct <- recPhalarisPct
d$recBuissonPct <- recBuissonPct
d$lognb <- log10((d$nb*100/d$longueur) + 1)
vars <- c("nb", "lognb", "aquaplants", "longueur", "hauteur", "largeur",
          "tourniere", "recRuisseau", "recPhalaris", "recBuisson",
           "temp", "conduct", "pH")
```

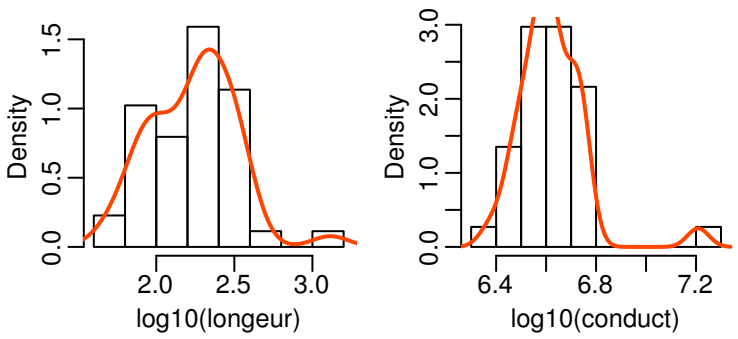
On peut créer un histogramme pour chaque variable pour examiner leur distribution.

```
# dev.new(width = 15/2.54, height = 10/2.54)
par(mfrow = c(3,4), mar = c(3,3,1.5, 0.5), mgp = c(1.6, 0.5,0))
for (i in 1:ncol(d[,vars[-(1:2)]])) {
    tmp <- d[,vars[-(1:2)]]
    hist(as.numeric(na.omit(tmp[,i])), freq = FALSE, main = "" , xlab = colnames(tmp)[i])
    lines(density(as.numeric(na.omit(tmp[,i]))), col = "orangered", lwd = 2)
}
```



On constate que la plupart des distributions sont à peu près correctes. Pour rappel une distribution uniforme est idéale (histogramme plat, même fréquence pour toutes les valeurs) mais une distribution à peu près symétrique est aussi acceptable.
Dans notrecas on voit que les variables longueur et conductivité ont toutes les deux au moins une observation nettement plus grande que les autres. Cette unique observation pourrait avoir une forte influence (non désirée) sur les résultats de notre future analyse. Une tranformation log permet de diminuer l'écart entre ces points et le reste de la distribution. On gardera en tête cette transformation pour la construction des GLMs.

```
# dev.new(width = 10/2.54, height = 5/2.54)
par(mfrow = c(1,2), mar = c(3,3,1.5, 0.5), mgp = c(1.6, 0.5,0), cex = 0.8)

hist(log10(d[, "longueur"]), main = "", xlab = "log10(longeur)", freq = FALSE)
lines(density(log10(d[, "longueur"])), col = "orangered", lwd = 2)

hist(log(d[, "conduct"]), main = "", xlab = "log10(conduct)", freq = FALSE)
lines(density(log(na.omit(d[, "conduct"]))), col = "orangered", lwd = 2)
```




```
d$log10longueur <- log10(d$longueur)
d$log10conduct <- log10(d$conduct)

vars <- c("nb", "lognb", "aquaplants", "log10longueur", "hauteur", "largeur",
          "tourniere", "recRuisseau", "recPhalarisPct", "recBuissonPct",
          "temp", "log10conduct", "pH")
```

Corrélations entre variables explicatives

On peut obtenir facilement une matrice de corrélation dans R mais le résultat est à peu près illisible... Ici un exemple avec seuelement les 6 premières variables.

```
cor(d[,vars[1:6]])
```

```
##                       nb       lognb  aquaplants log10longueur    hauteur    largeur
## nb             1.0000000  0.70912639  0.43711906    0.25348679 -0.2912562 -0.3136896
## lognb          0.7091264  1.00000000  0.66399953    0.03857341 -0.4751661 -0.5051945
## aquaplants     0.4371191  0.66399953  1.00000000    0.06005888 -0.3141106 -0.2739419
## log10longueur  0.2534868  0.03857341  0.06005888    1.00000000 -0.1821712 -0.3546097
## hauteur       -0.2912562 -0.47516611 -0.31411059   -0.18217122  1.0000000  0.8917264
## largeur       -0.3136896 -0.50519452 -0.27394192   -0.35460975  0.8917264  1.0000000
```

On aura donc recours à des méthodes de visualisation pour examiner les relations entrevariables explicatives.

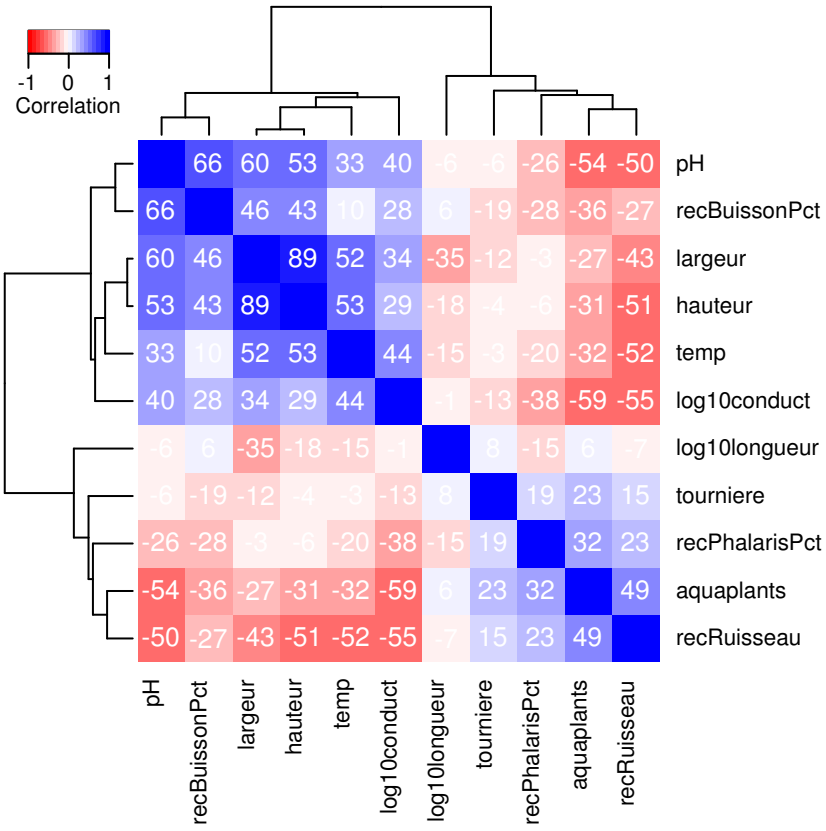
Heatmap de la matrice de corrélation .

Une première approche utilise une combinaison de "heatmap" pour mettre en évidence les corrélations les plus fortes avec des couleurs et de "clustering" pour grouper et réordonner les variables. Cette approche est particulièrement utile quand on a un très grand nombre de variables à examiner en même temps.

La fonction `corheatmap` (dans le script `mytoolbox.R`) permet de produire très facilement ce genre de représentations.

```
# NB : la fonction donne message d'avertissement (warning) pour prévenir
# que les facteurs resPahalaris et recBuisson ont été convertis en valeurs numériques.
# Dans ce cas, puisqu'il s'agit de valeurs ordinales, les coefficients de corrélation
# calulés peuvent être interprétés (ce ne serait pas le cas avec des
# variables nominales)
```

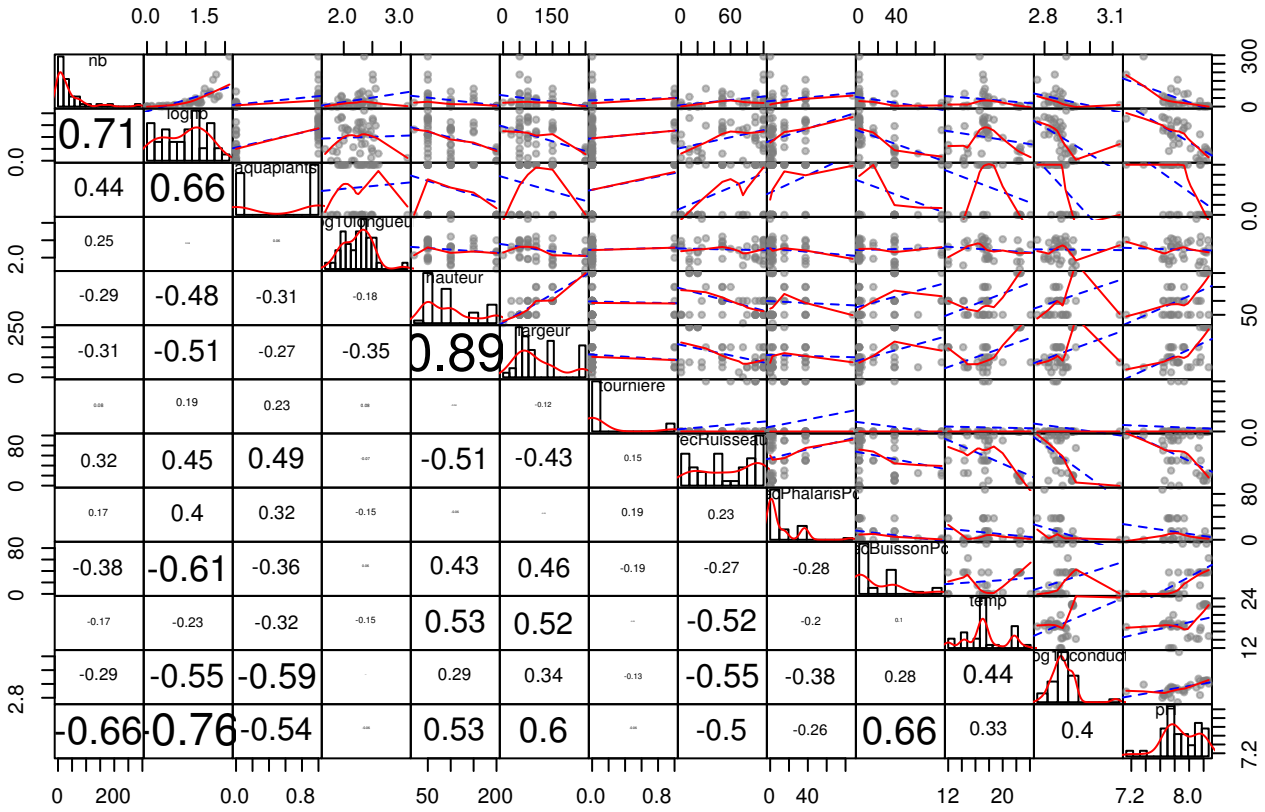
```
# dev.new(11/2.54, 11/2.54)
corheatmap(d[vars[-(1:2)]])
```



Scatter Plot Matrix (SPLOM) .

Lorsque le nombre de variables est plus limité comme ici, un "scatterplot matrix" (SPLOM) donne une représentation beaucoup plus dense en informations particulièrement utiles à ce stade.
On visualise non seulement les corrélations dans les paneaux inférieurs (les coefficients de corrélations les plus grands en valeur absolue sont affichés en plus grand) mais on peut aussi visualiser la forme de la relation grâce aux nuages de points (scaterplots) et leur courbes de lissage (en rouge) dans les panneaux suppérieurs. On a également un rappel de la distribution sur la diagonale.

```
# dev.new(18/2.54, 13/2.54)
pairs2(d[,vars], reorder = FALSE, pt.cex = 1)
```



Les deux premières colonnes et lignes montrent la relation entre les variables que l'on veut étudier (nombre d'individus) et les différentes variables explicatives. Le reste montre les relations entre variables explicatives.

On peut voir que les hauteur des berges et la largeur du drain sont très fortement corrélés (R = 0.89). Ces deux variables sont fortement redondantes. Prendre la moyenne ou la somme de ces variables n'aurait pas beaucoup de sens. On devra donc en laisser tomber une des deux. Le choix ici est totalement arbitraire.

Les autres variables ont des corrélations a priori plus raisonables. Ceci dit les corrélations 2 à 2 sont loin de être proche de 0 dans de nombreux cas. Il est donc possible que les corrélations multiples soient importantes. Ceci devra être vérifié avec les VIFs lors de la construction des GLMs.

Il semble également que les drains avec un pH plus élevé aient aussi tendance à être plus recouverts par des buissons (R = 0.66). Il est possible que les feuilles de ces buissons tombant dans l'eau aient une influence sur le pH.

On voit également que malgré la tranformation log de la conductivité, la relation nombre d'inividus vs conductivité n'est pas très linéaire probablement à cause du point extrème.

Ordinations

Les méthodes d'ordination permettent de représenter dans un nombre de dimensions réduites (en général 2) des données avec de nombreuses variables (= dimensions). D'ou le nom utilisé en écologie "Ordination en espace réduit".

La méthode la plus connue est l'analyse en composante principale (PCA) qui permet à la fois de représenter les corrélations (ou covariances) entre les colonnes du jeu de données et la distance Euclidienne entre les lignes du jeu de données.

D'autres méthodes permettent de représenter d'autres types de distances : distance de Chi carré pour l'analyse des correspondances (CA) et n'importe quelle matrice de distance pour l'analyse en coordonnées principales (PCoA) appellée aussi "Metric Dimensional Scaling" (MDS). La PCA peut aussi représenter d'autres distances que la distances euclidienne (pex : Hellinger, Chord) moyennant une transformation du jeu de données original avant de réaliser une PCA classique (tbPCA : transformation based PCA).

Ces méthodes peuvent s'utiliser dans des objectifs différents : - obtenir des nouvelles variables orthogonales (non corrélées) dont les premières résument au mieux l'information présente dans le jeu de données - représenter en 2 dimensions les corrélations (ou covariances) entre les colonnes du jeu de données - représenter en 2 dimensions les similarités/disimilarités entre les lignes du tableau.

Ce sont ces deux derniers objectifs qui nous intéressent ici et en particulier le dernier (disimilarités entre les drains), la corrélation entre variables ayant déjà été explorée précédemment. Cependant les relations entre les variables permettent d'interpréter pourquoi certains drains sont similaires ou pas.

Principal Component Analysis (PCA) Dans notre cas une simple PCA est adaptée :

- On doit travailler sur la matrice de corrélation (données standardisées) et pas sur la covariance car les données ne sont pas homogènes (les unités des colonnes ne sont pas les mêmes).
- La distance Euclidienne est adaptée parce qu'on a des données quantitatives, ordinales ou binaires (pas de variable qualitative nominale) et parce qu'on a relativement peu de valeurs nulles (pas de problèmes de double 0).

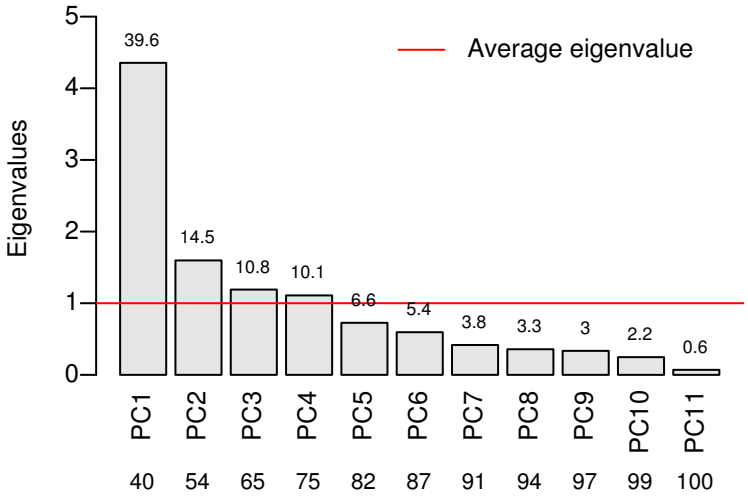
On utilise pas la varible "nombre d'individus" pour construire la PCA mais on peut la rajouter comme variable supplémentaire sur certains graphiques pour visualiser sa corrélation avec les autres variables et les axes.

Il existe de nombreuses fonctions et packages R qui peuvent produire ce genre d'analyses mais chacune a ses options par défaut qui lui sont propres et pratiquement aucune ne donne les mêmes résultats par défaut...
Le package `FactoMineR` a l'avantage de produire assez facilement des graphiques qui sont très corrects sur le plan estéthique. Par contre il ne permet pas de facilement produire les biplots très souvents utilisés en écologie pour représenter variables et observations sur le même graphique. Le package `vegan` permet de représenter les résultats sous forme de biplot mais le résultat par défaut est parfois de qualité médiocre. Plusieurs fonctions fournies dans `mytoolbox.R` facilitent l'exploitation de ces résultats.

```
library(FactoMineR)
row.names(d) <- d$code2
d2 <- na.omit(d[,vars[-1]])
pca <- PCA(d2, quanti.sup = 1, graph = FALSE, ncp = Inf)
```

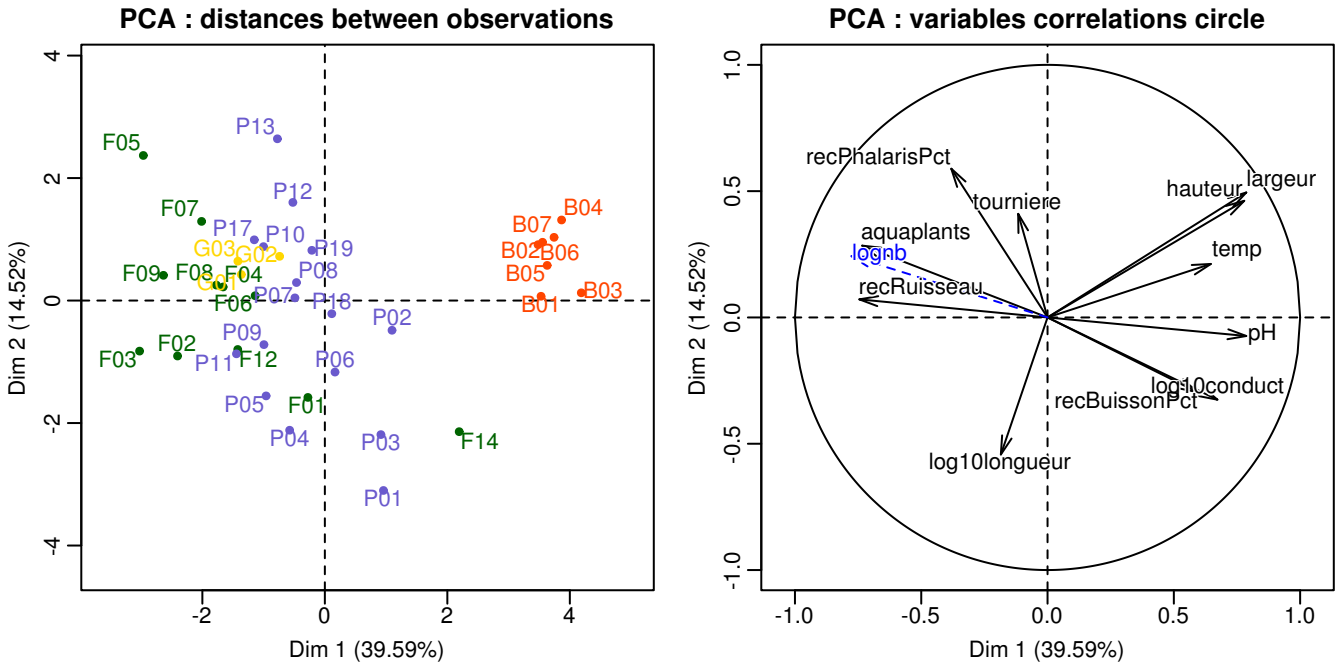
Graphique des éboulis (screeplot)

```
# dev.new(width = 10/2.54, height = 7/2.54)
par( mar=c(4,3,1,0.5), cex = 0.8)
eigenplot(pca$eig[,1]) # fonction dans mytoolbox.R
```



Cercle des corrélations et graphique des distances entre sites

```
# dev.new(18/2.54, 9/2.54)
par(mfrow = c(1,2), mar=c(3,3,2,1), mgp = c(1.8, 0.5, 0), cex = 0.7)
set.seed(123)

mycols <- as.factor(substring(row.names(d2),1,1)) # extraction de la première lettre
mycols <- c("orangered", "darkgreen", "gold", "slateblue")[as.numeric(mycols)]
plot(pca, choix="ind", axes = c(1,2), col.ind = mycols,
     title = "PCA : distances between observations")

plot(pca, choix="var", axes = c(1,2), shadowtext = TRUE,
     title = "PCA : variables correlations circle")
```



Le cos2 donne une indication de la qualité de la représentation des observations et des variables dans le plan factoriel choisi (ici les 2 premières composantes principales). Si le cos2 est = 1 la représentation dans le plan du graphique est parfaite. Il s'agit du % de variation de la variable représentée dans le plan du graphique (plan factoriel).

```
cos <- pca$ind$cos2
round(cos[,1] + cos[,2], 2)
```

```
##  B01  B02  B03  B04  B05  B06  B07  F01  F02  F03  F04  F05  F06  F07  F08  F09  F12  F14  G01  G02
## 0.84 0.95 0.88 0.91 0.96 0.95 0.96 0.31 0.66 0.64 0.49 0.62 0.17 0.68 0.50 0.67 0.46 0.32 0.25 0.09
##  G03  P01  P02  P03  P04  P05  P06  P07  P08  P09  P10  P11  P12  P13  P17  P18  P19
## 0.33 0.52 0.17 0.56 0.63 0.64 0.31 0.05 0.14 0.25 0.32 0.41 0.18 0.48 0.17 0.01 0.33
```

Même PCA avec le package `vegan`

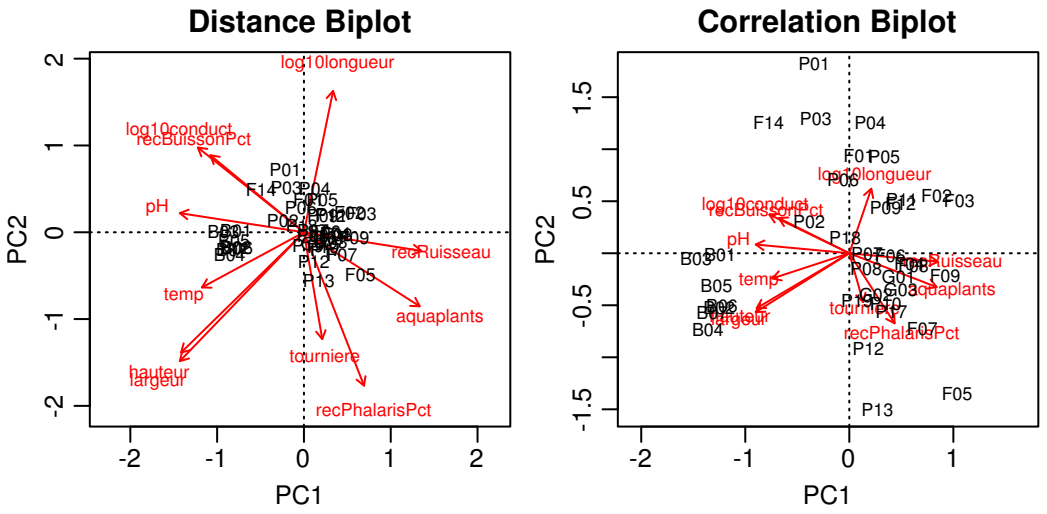
```
library(vegan)
res <- rda(d2[,-1], scale = TRUE)
```

On peut obtenir les mêmes graphiques et informationsqu'avec `FactoMineR` avec les fonctions suivantes (fournies dans `mytoolbox.R`). - Code non exécuté ici.

```
eigenplot(res) # Graphique des éboulis (screeplot)
corplot(res) # cercle des corrélations
corplot(PC = res, Y = d2, suppl = 1) # cercle des corrélations avec variable supplémentaire
cos2(res) # cos2
cos2vars(res) # cos2 pour les variables (colonnes)
cos2obs(res) # cos2 pour les observations (lignes)
```

Les biplots standard ne sont pas toujours très lisibles ni très facile à personaliser...

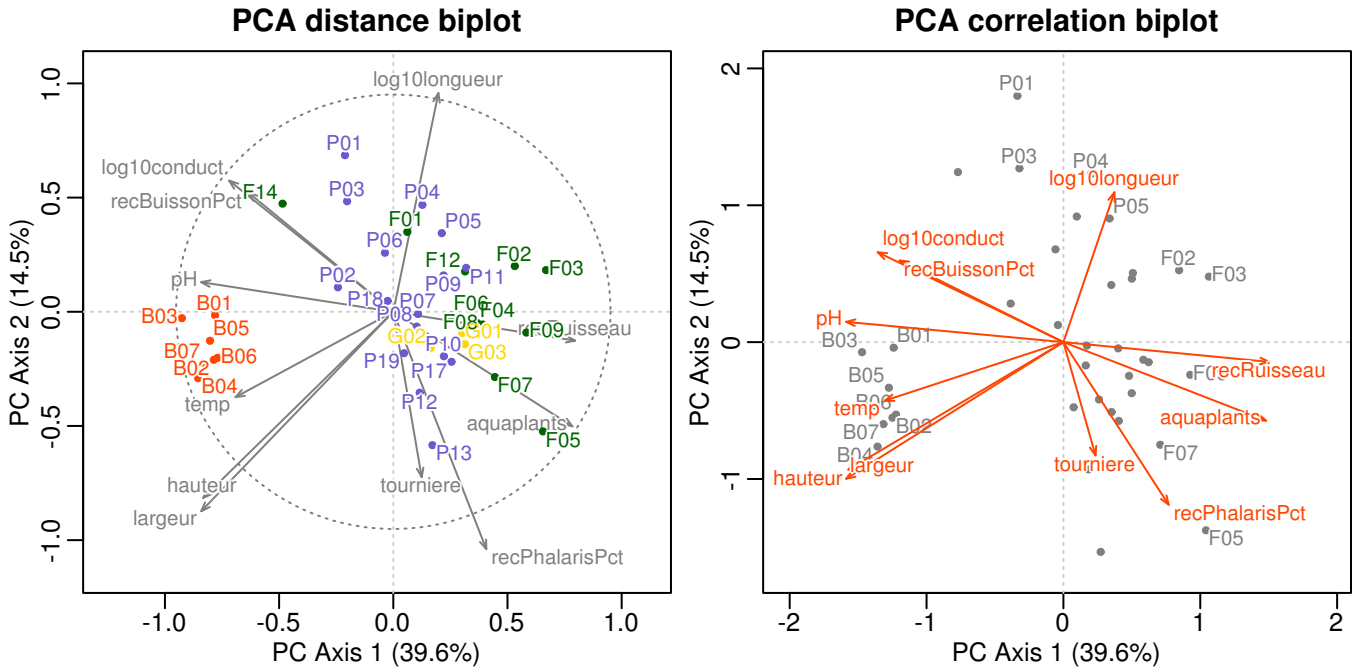
```
# dev.new(14/2.54, 7/2.54)
par(mfrow = c(1,2), mar = c(3,3,2,0.5), mgp = c(1.8, 0.6, 0), cex = 0.8)
biplot(res, scaling = 1, main = "Distance Biplot")
biplot(res, scaling = 2, main = "Correlation Biplot")
```



Biplots personalisés avec `biplot2`.
Dans le graphique de droite, on a affiché les étiquettes uniquement pour les observations correctement représentées (cos2 > 0.5) pour éviter de surcharger le graphique. Les étiquettes sont placées automatiquement pour limiter les recouvrements.
Attention le cercle n'a pas la même interprétation que dans le cercle des corrélations (dans ce cas une flèche touchant le cercle est parfaitement représentée dans le plan). Il s'agit ici d'un cercle de "contribution équilibrée". Le rayon du cercle représente la longueur qu'aurait une variable contribuant de manière égale à toutes les dimensions. Les flèches dépassant le cercle sont donc mieux représentées dans le plan du graphique que dans les autres dimensions.

Rappel : dans le “distance biplot”, la distance entre points est une représentation en 2 dimensions de la distance euclidienne entre les sites (drains ici) L’angle entre les flèches ne doit pas être interprété (les axes d’origine sont toujours à 90° mais projetés en 2D). La projection de la pointe des flèches sur les axes montre leur contribution à ces axes.
Dans le “correlation biplot” l’angle entre les flèches (leur cosinus plus exactement) représente leur corrélation (les axes d’origine ne sont plus à 90°) par contre la distance entre les points ne représente plus leur distance euclidienne.
Dans les deux types de graphique on peut projeter les points à 90° sur les flèches pour avoir une approximation de la valeur de chaque variable (flèches) dans chaque site (points).

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mfrow = c(1,2), mar = c(2.5,2.5,2,0.5), mgp = c(1.5, 0.5, 0), cex = 0.8)
biplot2(res, sc = 0.5, scaling = 1, obs.col = mycols)
biplot2(res, sc = 1.5, scaling = 2, obs.col = "gray50", var.col = "orangered",
          obs.cos2.lim = 0.5)
```



Interprétation

Screeplot : le premier axe résume l’essentiel de la variabilité des données suivi de 3 axes ayant une contribution similaire(10 - 14%). Le premier plan factoriel devrait donc être un relativement bon résumé des données mais il ne capture pas toute la variation (54%).

Le pattern le plus clair est que le groupe de drains “B” au nord de la zone qui se distingue nettement des autres. Ce sont des drains avec vraisemblablement moins de plantes aquatiques, un pH et une température plus élevée et ces drains sont aussi plus larges et plus hauts.

Les graphiques des corrélations donnent une autre représentation des corrélations que les deux autres vues précédemment (SPLOM et heatmap). On voit bien la forte corrélation entre hauteur et largeur. De manière générale on voit qu’il y a un groupe d evariables positivement corrélées : hauteur, largeur, temp, pH, recBuisson, conductivité. Ces variables sont négativement corrélées avec la présnece d eplantes aquatiques et le recouvrement du Ruisseau ainsi qu’avec le nombre d’individus observés. Le nombre d’inidvvidus n’a pas été utilisé pour construire la PCA. Cette variable a simplement été ajoutée comme variable supplémentaire.

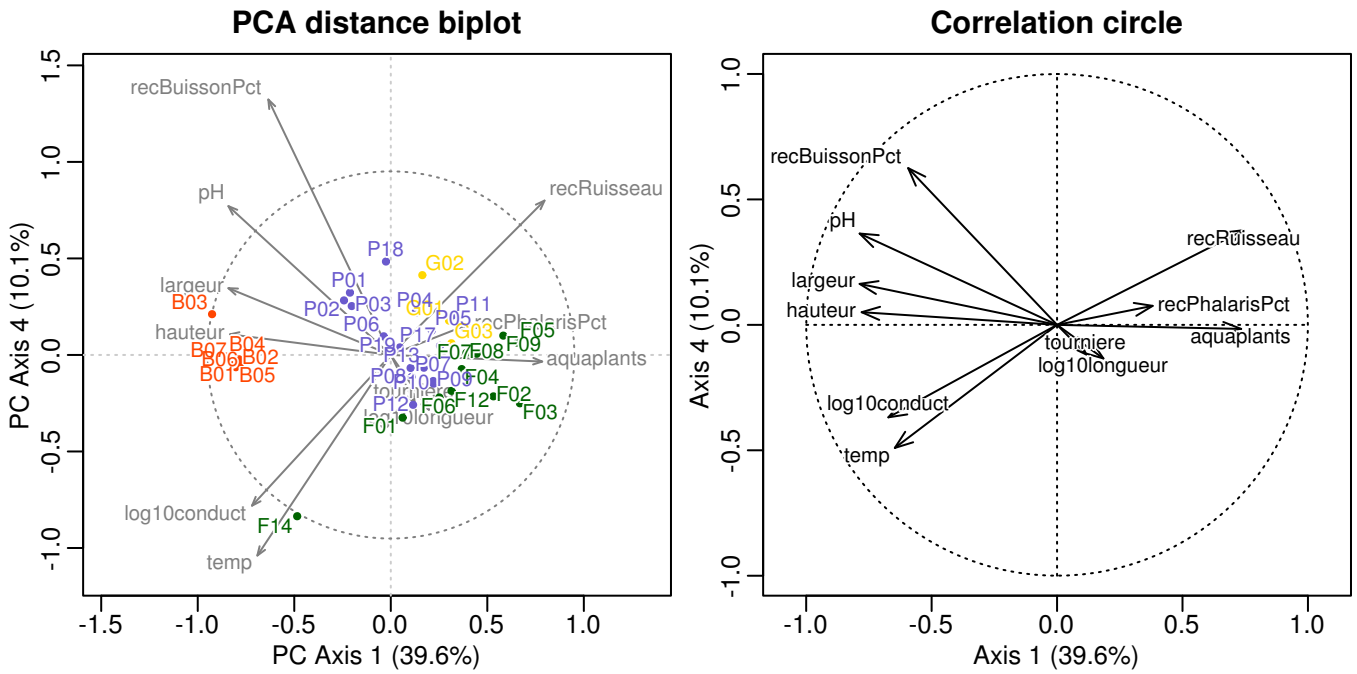
Il faut cependant se méfier de ce genre de représentations simplifiées des données (11 dimensions représentées en 2 dimensions) car elles peuvent donner de fausses impressions. Par exemple on a l’impression que la conductivité et le recouvrement en Buissons sont très corrélées (les deux flèches sont assez longues et se superposent totalement) alors qu’on sait que leur corrélation est seulement de 0.28. recBuisson est par exemple nettement plus corrélée avec le pH ($R = 0.66$). La table des $\cos^2$ donne montre que recBuisson est en fait assez mal représentée dans ce plan. Elle est beaucoup mieux représentée par l’axe 4.

```
cos2(res)[[1]][,1:7]
```

```
##                     PC1    PC2    PC3    PC4    PC5    PC6    PC7
## aquaplants       0.5387 0.0808 0.0384 0.0003 0.1125 0.0566 0.0064
## log10longueur    0.0342 0.2944 0.4424 0.0180 0.0414 0.1254 0.0044
## hauteur          0.6066 0.2133 0.0309 0.0025 0.0024 0.0182 0.0157
## largeur          0.6194 0.2448 0.0012 0.0267 0.0108 0.0068 0.0007
## tourniere        0.0135 0.1683 0.5920 0.0139 0.0109 0.1832 0.0008
## recRuisseau      0.5553 0.0051 0.0156 0.1425 0.0482 0.0027 0.1439
## recPhalarisPct   0.1453 0.3468 0.0008 0.0057 0.4552 0.0044 0.0349
## recBuissonPct    0.3522 0.0875 0.0354 0.3902 0.0014 0.0005 0.0318
## temp             0.4182 0.0449 0.0023 0.2407 0.0290 0.0906 0.0847
## log10conduct     0.4523 0.1060 0.0076 0.1359 0.0072 0.1066 0.0936
## pH               0.6194 0.0054 0.0237 0.1324 0.0075 0.0005 0.0000
```

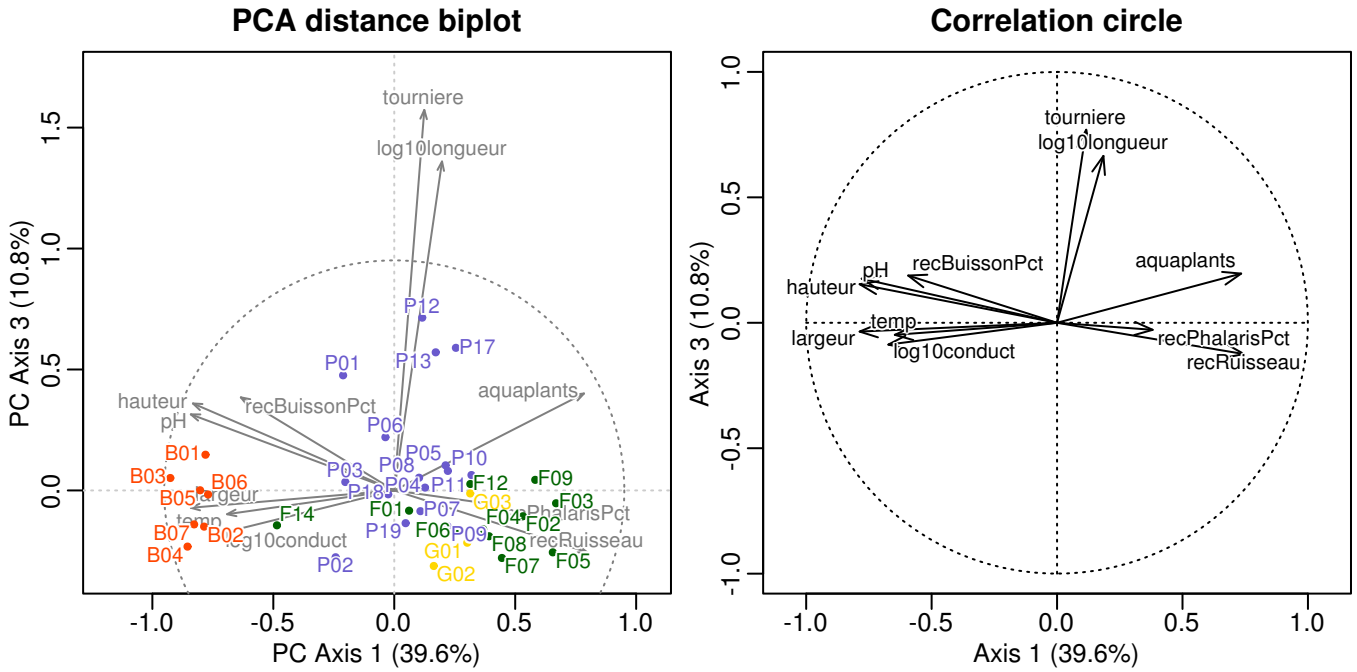
Une représentation dans le plan 1-4 donne une représentation plus fidèle de la réalité. Du point de vue des sites cette représentation n'apporte pas grand chose en plus. Elle montre juste que le drain F14 a une température et une conductivité nettement plus élevée que la moyenne.

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mfrow = c(1,2), mar = c(2.5,2.5,2,0.5), mgp = c(1.5, 0.5, 0), cex = 0.8)
biplot2(res, choices = c(1,4), sc = 0.5, scaling = 1, obs.col = mycols)
corplot(res, choices = c(1,4))
```



L'axe 3 permet surtout de distinguer les 3 drains avec tournière (P12, P13, P17), et le drain qui est nettement plus long que les autres (P01)

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mfrow = c(1,2), mar = c(2.5,2.5,2,0.5), mgp = c(1.5, 0.5, 0), cex = 0.8)
biplot2(res, choices = c(1,3), sc = 0.5, scaling = 1, obs.col = mycols)
corplot(res, choices = c(1,3))
```



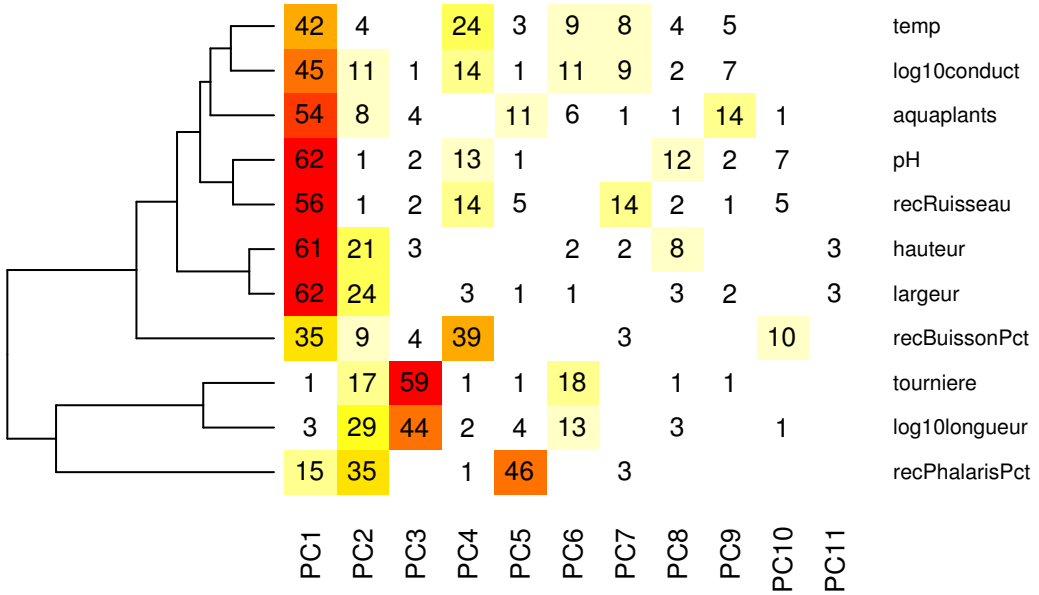
NB : les statistiques $\cos^2$ sont très utiles pour aider à l'interprétation des PCA (et autres ordinations) mais il est plus facile de les interpréter avec une représentation graphique comme une simple heatmap.

La ligne de code suivante donne une telle représentation avec les fonctions de base de R (non exécutée ici)

```
heatmap(cos2(res)[[1]], Colv = NA, col = rev(heat.colors(5)))
```

La fonction suivante (mytoolbox.R) permet d'obtenir une heatmap légèrement améliorée.

```
# dev.new(width = 14/2.54, height = 9/2.54)
cos2heatmap(cos2(res)[[1]], cexRow = 0.8)
```




```
# cos2heatmap(cos2(res)[[2]], reorder = FALSE) # cos² of the observations
```

Non Metric Multidimensional Scaling On le voit il faut parfois aller chercher dans plusieurs dimensions pour distinguer de subtiles différences. Ceci est dû au fait que la PCA n'essaye pas uniquement de reproduire le plus fidèlement possible les distances euclidiennes du jeu de données initial mais qu'elle ajoute la contrainte de représenter un maximum de variance sur les premiers axes et que les axes soient orthogonaux entre eux.

D'autres méthodes comme la NMDS (non Metric Multidimentional Scaling) essayent uniquement de représenter les distances le plus fidèlement possible dans un nombre prédéfini de dimentions (generalement 2). Le lien entre les distances dans l'espace réduit et les distances originales n'est plus linéaire mais simplement monotone. Mais le résultat est souvent une meilleure représentation en 2 dimensions.

On utilise une fonction du package MASS et on pratique ensuite une ACP sur les 2 dimensions uniquement pour effectuer une rotation des axes. NB : on utilise ici une mesure de distance Euclidienne (sur la matrice standardisée) qui est adaptée ici. La distance Euclidienne n'est pas toujours adaptée. La NMDS peut s'utiliser avec n'importe quelle mesure de distance.

```
resNMDS <- MASS::isoMDS(dist(scale(d2[,-1])))
```

```
## initial  value 24.588068
## iter   5 value 15.961641
## final  value 15.832796
## converged
```

```
resNMDS <- princomp(resNMDS$points)
```

On peut vérifier que la corrélation entre les distances originales et les distances en 2 dimensions sont meilleures (corrélation de spearman plus grande) pour le NMDS. Mais la différence est relativement faible ici.

```
# Correlation entre les distances entre observations du jeu de données original
# et les distance dans le plan NMDS
cor(dist(scale(d2[,-1])), dist(-resNMDS$scores), method = "spearman")
```

```
## [1] 0.9333873
```

```
# plot(dist(scale(d2[,-1])),dist(-resNMDS$scores))

# Correlation entre les distances entre observations du jeu de données original
# et les distance dans le plan NMDS
resPCA <- princomp(scale(d2[,-1]))
cor(dist(scale(d2[,-1])),dist(-resPCA$scores[,1:2]), method = "spearman")
```

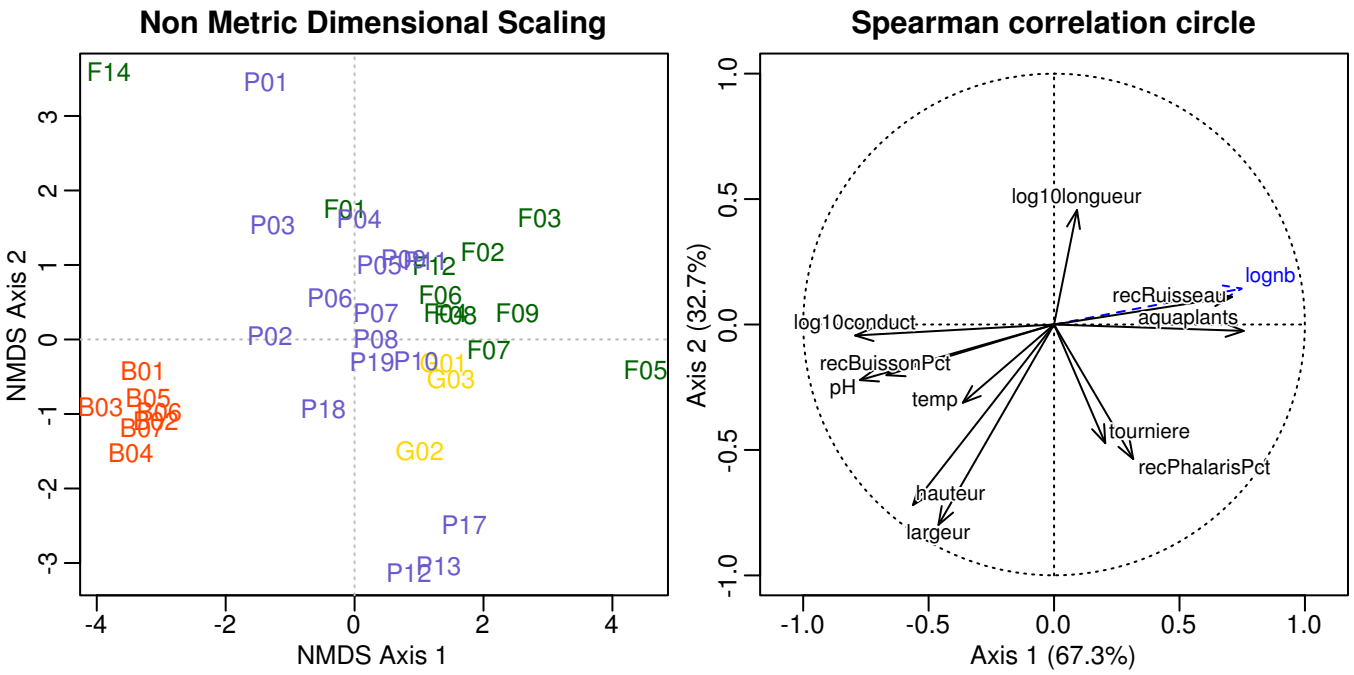
```
## [1] 0.8428645
```

Le résultat est assez similaire à la PCA mais on résume ici en 2 dimensions l'information qu'il a fallu aller chercher dans 4 dimensions dans la PCA, notamment le fait que les points F14, P01, P12, P13, P17 sont assez différents des autres. Le drain F05 semble également se démarquer (dans la PCA il se démarquait sur l'axe 5 !). C'est un drain qui se caractérise notamment par une couverture importante de baldingère (recPhalarisPct).
Le cercle des corrélations aide à l'interprétation (notez qu'il s'agit ici du coefficient de rang de Spearman qui décrit une relation monotone mais pas spécialement linéaire).

Ce résultat permet aussi de confirmer que les 3 drains "G" situés à l'est de l'autre côté de l'autoroute ne sont pas particulièrement différents des autres drains du point de vue de leurs caractéristiques environnementales.

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mfrow = c(1,2), mar = c(2.5,2.5,2,0.5), mgp = c(1.5, 0.5, 0), cex = 0.8)
plot(-resNMDS$scores, type = "n", xlab = "NMDS Axis 1", ylab = "NMDS Axis 2",
     main = "Non Metric Dimensional Scaling")
```

```
abline(h=0, v=0, lty = 3, col = "gray75")
text(-resNMDS$scores, labels = row.names(resNMDS$scores), col = mycols)
corplot(PC = -resNMDS$scores, Y = d2, suppl = 1, Rmethod = "spearman",
        title = "Spearman correlation circle")
```



Clustering et heatmaps

La combinaison du clustering et d'une heatmap apporte à nouveau une quantité d'information très dense mais malgré tout structurée et "digéreable" qui va aider à confirmer et/ou mieux comprendre les observations précédentes.

Le clustering implique 3 catégories de choix :

- le type de clustering (hiérarchique ou non, divisif ou agglomératif, etc...) Le clustering ascendant hiérachique est une des méthodes les plus largement répandues.
- l'algorithme précis : en clustering ascendant hirarchique, les plus courrants sont "Ward", "complete linkage" et "average linkage" (= UPGMA). Ce choix est assez arbitraire. Le but est de trouver des groupes biologiquement interprétables. Ward tend à créer des groupes de tailles similaires en minimisant la variance intra groupe et maximisant la variance intergroupes.
- la matrice de distance. Le choix est ici moins arbitraire. Pour des données environnemntales sans variables qualitatives et avec relativement peu de 0, la distance euclidienne est la plus souvent utilisée pour calculer les distances entre lignes. Voir Legendre & Legendre 2012 pour une guidance détaillée sur le choix des mesures de distance/similarité).

On va travailler ici sur une matrice de données standardisée (car les unités ne sont pas les mêmes entre variables) et avec une distance Euclidienne pour les lignes et une matrice de corrélation pour les colonnes.

```
full <- na.omit(d[,vars])
d2 <- scale(full[,-(1:2)]) # : scale permet de standardiser les variables
```

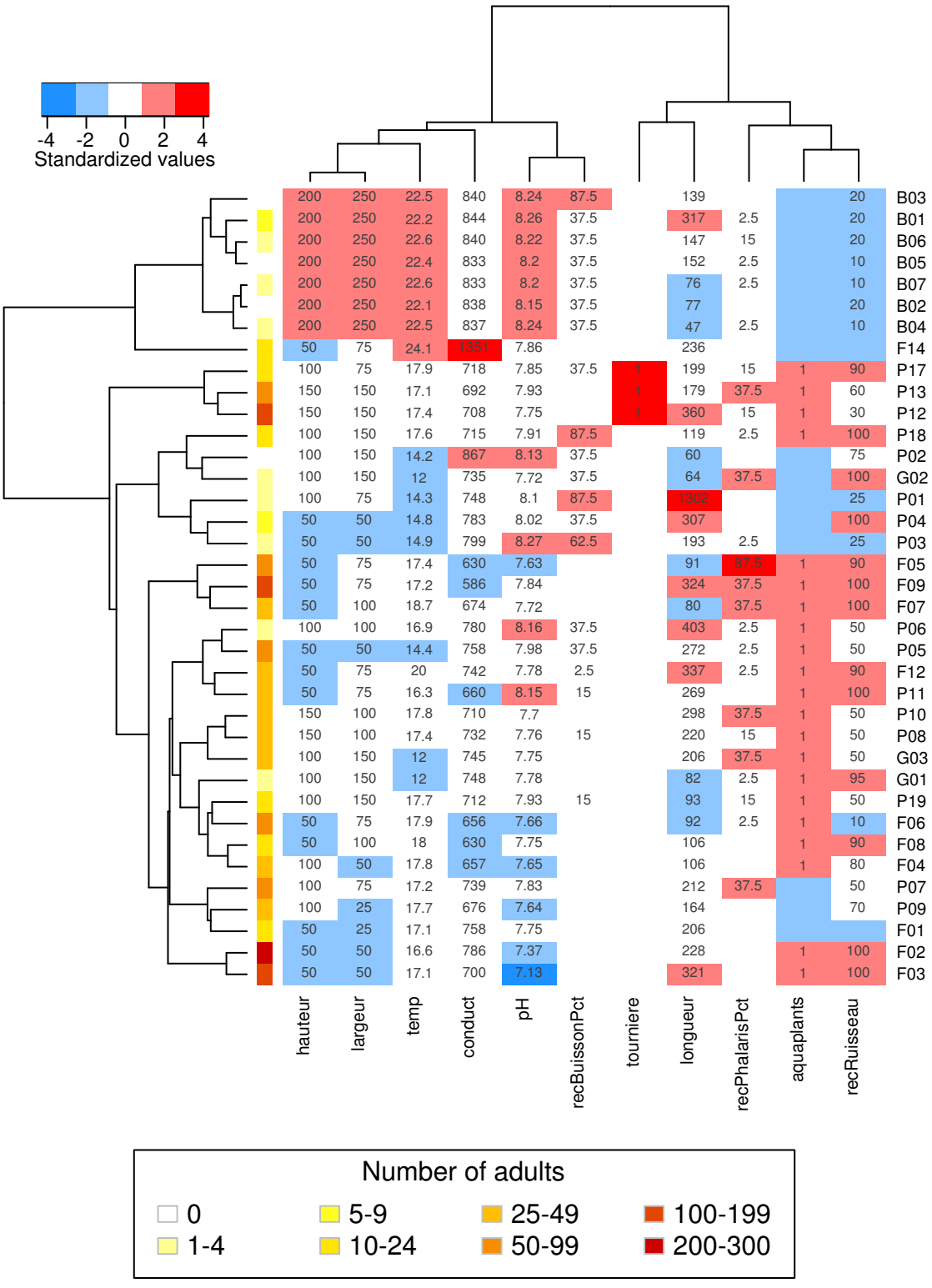
La fonction `heatmap.2` du package `gplots` permet de construire des heatmaps avec dendrogrammes et propose de nombreuses options.
Dans notre cas une version simple mais déjà exploitable pourrait s'obtenir comme suit (pas exécuté ici). Par défaut les dendrogrammes sont construits sur une matrice de distances euclidiennes (y compris pour les colonnes) et un algorithme "complete linkage".

```
heatmap.2(d2, trace = "none", density.info="none",
          col = colorRampPalette(c("dodgerblue", "white", "red"))(5))
```

Un peu de travail supplémentaire permet de personaliser complètement le résultat. Notez qu'on a ajouté une bande avec un gradient blanc-jaune-orange-rouge représentant les abondances de libellules (blanc = 0 individus). Les cases blanches correspondent aux valeurs autour de la moyenne plus ou moins 1 écart type. En rouge et en bleu on a donc les valeurs qui s'écartent plus ou moins de leur moyenne. Les cases sans valeurs sont des valeurs nulles, pas de valeurs manquantes.

```r
# Clustering pour les lignes : ward sur distance euclidienne de la matrice standardisée
cl <- hclust(dist(d2), method = "ward.D2")
rowcl <- as.dendrogram(reorder(cl,  rowMeans((d2))))

# Clustering pour les colonnes : ward sur les corrélations transformées en distances
corenv <- cor(d2)
cl <- hclust(as.dist(1-corenv), method = "ward.D2")
colcl <- as.dendrogram(reorder(cl,  rowMeans((1-corenv))))

# Valeurs à afficher sur la heatmap. Les 2 variables log10 transformées
# sont exprimées dans leur échelle d'origine
# Les valeurs nulles sont remplacées par des valeurs manquantes pour alléger le graphique
values <- full[,-(1:2)]
values$log10conduct <- 10^values$log10conduct
values$log10longueur <- round(10^values$log10longueur,0)
values[values==0] <- NA

colnames(d2)[c(2,10)] <- c("longueur", "conduct")

# Bande de couleurs à ajouter dans la marge pour représenter l'abondance des libellules
# (comme une variable supplémentaire)
tmp <- cut(full$nb,breaks = c(0, 1, 5, 10, 25, 50, 100, 200, 300), right = FALSE)
mycols <- colorRampPalette(c("white", "yellow", "orange", "red3"))(length(levels(tmp)))
grcol <- mycols[tmp]
```

```r
# dev.new(width = 15/2.54, height = 18/2.54)
heatmap.2(d2, Colv = colcl,  Rowv = rowcl,
          col = colorRampPalette(c("dodgerblue", "white", "red"))(5),
          cellnote = values, notecex = 0.8, notecol="gray25",
          cexRow = 0.95, cexCol = 1,
          offsetCol = 0, offsetRow = 0, margins = c(7 , 2.5),
          trace = "none", density.info = "none",
          lhei = c(1,5), lwid = c(2,5),
          key.par=list(mar = c(3,2,3.5,2), mgp = c(1, 0.5, 0)),
          key.title = "", key.xlab =  "Standardized values",
          # add colored band
          RowSideColors = grcol
          )
```


-4 -2 0 2 4
Standardized values
hauteur
largeur
temp
conduct
pH
recBuissonPct
tourniere
longueur
recPhalarisPct
aquaplants
recRuisseau
B03 B01 B06 B05 B07 B02 B04 F14 P17 P13 P12 P18 P02 G02 P01 P04 P03 F05 F09 F07 P06 P05 F12 P11 P10 P08 G03 G01 P19 F06 F08 F04 F07 F09 F01 F02 F03
Number of adults
0
1-4
5-9
10-24
25-49
50-99
100-199
200-300

Interprétation

Les sites se divisent comme suit :

Un premier groupe composé des sites Bxx et du site F14.
Le site F14 déjà mis en avant comme site à part par les ordinations est caractérisé par une conductivité extrèmement élevée. On peut se demander si il ne s'agit pas d'une erreur.
Les sites Bxx sont caractérisés par une hauteur et largeur élevées, une température et un pH élevés, l'absence de plantes aquatiques et des valeurs particulièrement faibles de recRuisseau. Il y a aussi une présencesystématique de buissons. On voit également que ce sont les sites où l'abondance des libellules est parmi la plus faible.

Le groupe suivant est formé des 3 sites P12, P13, P17 déjà mis en évidence por les ordinations également. Leur seule caractéristique distinctive semble ˆetre la présence d'une tournière. Pas très intéressant... On voit que cette variable binaire avec seulement 3 présences est potentiellement problématique.

Le groupe suivant est formé de 6 sites : P18, P02, G02, P01, P04, P03. C'est un groupe peu évident a priori mais intéressant car les abondances de libellules y sont faibles. A part le site P18, il s'agit de sites où la température est particulièrement froide, les buissons sont systématiquement présents et il n'y a aps de plantes aquatiques.

Les drains P01 et F05 mis en évidence par les ordiantions se retrouvent ici plus ou moins au milieu des autres variables et sont caractérisées comme déjà noté par une longueur extrème et un recouvrement en Phalaris très élevé.

Choix préliminaire des variables explicatives

```
Q09 - Suite à la phase d'exploration des données, choisissez un nombre raisonnable
(maximum 20) de variables environnementales que vous pensez pouvoir expliquer
la distribution de votre espèce et qui ne posent pas à priori de problèmes statistiques.
```

Suite à l'exploration des données, on décide d'utiliser les variables explicatives suivantes :

- log10(longueur)
- largeur (on laisse tomber la hauteur qui est très corrélée à cette variable)
- recRuisseau
- recPhalarisPct
- recBuissonPct
- aquaplants (présnece absence de plantes aquatiques)
- temp
- log10(conduct)

On a laissé tomber la variable "tournière" qui présente trop peu de données de présence et on a regroupé les présence/absence de différent genres de plantes dans la variable `aquaplants`. On choisi a priori d'appliquer une tranformation log à la longueur et la connectivité pour limiter l'effet d'une observation extrème dans chaque variable. On a choisi aussi d'utiliser les versions quantitatives de recPhalaris et recBuisson plutôt que leur version qualitative. Si la relation est bien linéaire, le modèle sera plus parcimonieux (un seul paramètre à estimer pour les variables quantitatives contre un nombre égale aux nombres de classes -1 pour les variables qualitatives).

Construction d'un modèle linéaire généralisé

```
Q10 - Construisez un premier modèle et évaluez sa qualité et les conditions d'applications.
Adaptez progressivement votre modèle en conséquence.
```

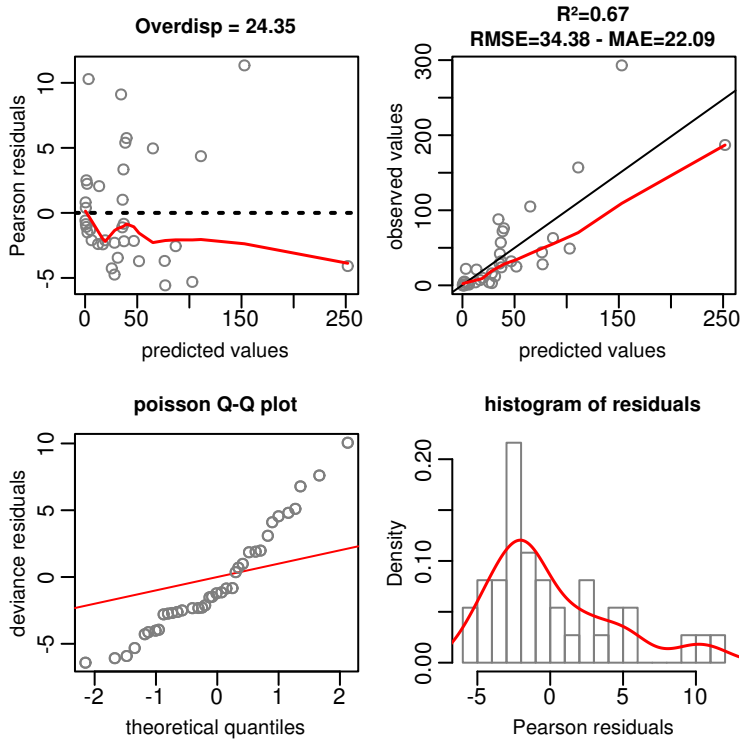
Modèle pour le nombre d'individus

On débute comme prévu avec un modèle de poisson pour le nombre d'individus observés. On voit immédiatement qu'il y a plusieurs problèmes.
Le coefficient de surdispersion est beaucoup trop élevé (Q10a): 24. Cette valeur signifie que les erreurs standard sont environs 5 fois trop petites par rapport à ce qu'elles devraient être. Les p valeurs seront donc beaucoup trop petites. Ceci a une influence sur les erreurs standard uniquement (et donc les inférences, p valeurs etc...) mais pas sur l'estimation des coefficients du modèle.
On voit également grâce au QQ-plot que les résidus ne suivent pas du tout une distribution de Poisson (Q10d).
J'ai essayé d'améliorer le modèle avec certaines transformations des variables x mais sans effet notoire sur ces problèmes.

```
row.names(d) <- d$code2
mod <- glm(nb ~   log10(longueur) + largeur +  recRuisseau + recPhalarisPct +
             recBuissonPct + aquaplants + temp + log10(conduct) + pH, data=d,
           family = poisson )
overdisp(mod)
```

```
## pearsonresid     deviance
##     24.35156     21.74973
```

```
diagplot(mod)
```




```
# diagplot2(mod)
```

NB : dans ce cas plutôt que d'ajouter la longueur comme une variable explicative et estimer son coefficient, on pourrait l'ajouter comme un offset. Dans ce cas son coefficient n'est pas estimé, il est fixé à 1 et cela revient à étudier le nombre

d'individus par mètre de drain (nb/longueur). Nb ceci ne fonctionne que sur une échelle logarithmique ! Cela ne change rien ici aux problèmes observés.
Pour étudier le nombre d'individus par 100m de drain on écrirait par exemple ceci avec un offset:

```
mod <- glm(nb ~   offset(log(longueur)-log(100)) + largeur +  recRuisseau + recPhalarisPct +
               recBuissonPct + aquaplants + temp + log10(conduct) + pH, data=d,
           family = poisson )
```

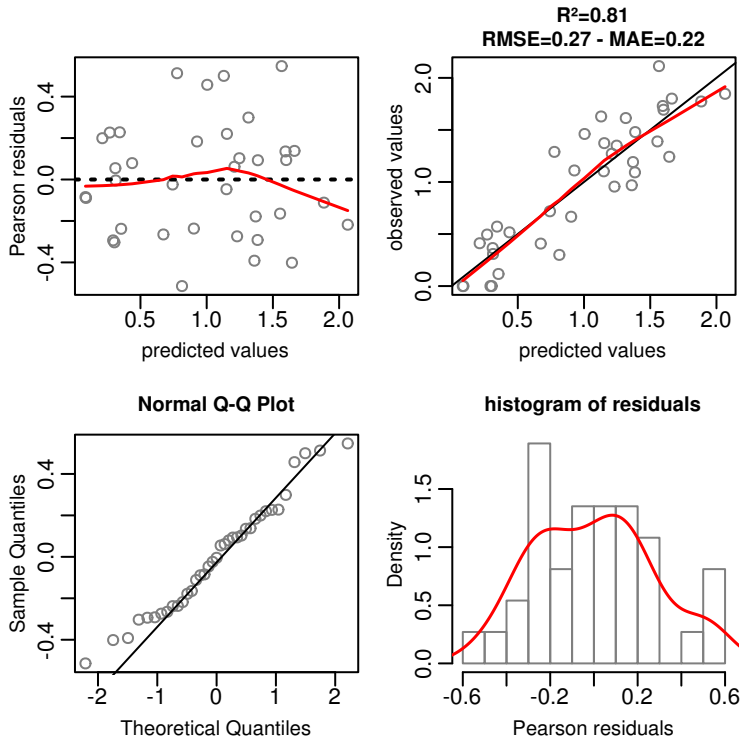
Par contre un modèle gaussien semble déjà bien meilleur. On travaille ici sur le nombre d'individus par 100 m de drain après transformation log10(x+1). Du coup on utilise plus la variable "longueur" comme variable explicative.

La **distribution des résidus** est symétrique et approximativement gaussienne (voir QQ-plot et histogramme des résidus - Q10d).

La **variance des résidus** semble à peu près homogène (pas d'augmentation ou de diminution flagrante quand les valeurs prédites augmentes : voir premier quadrant en haut à gauche - Q10a).

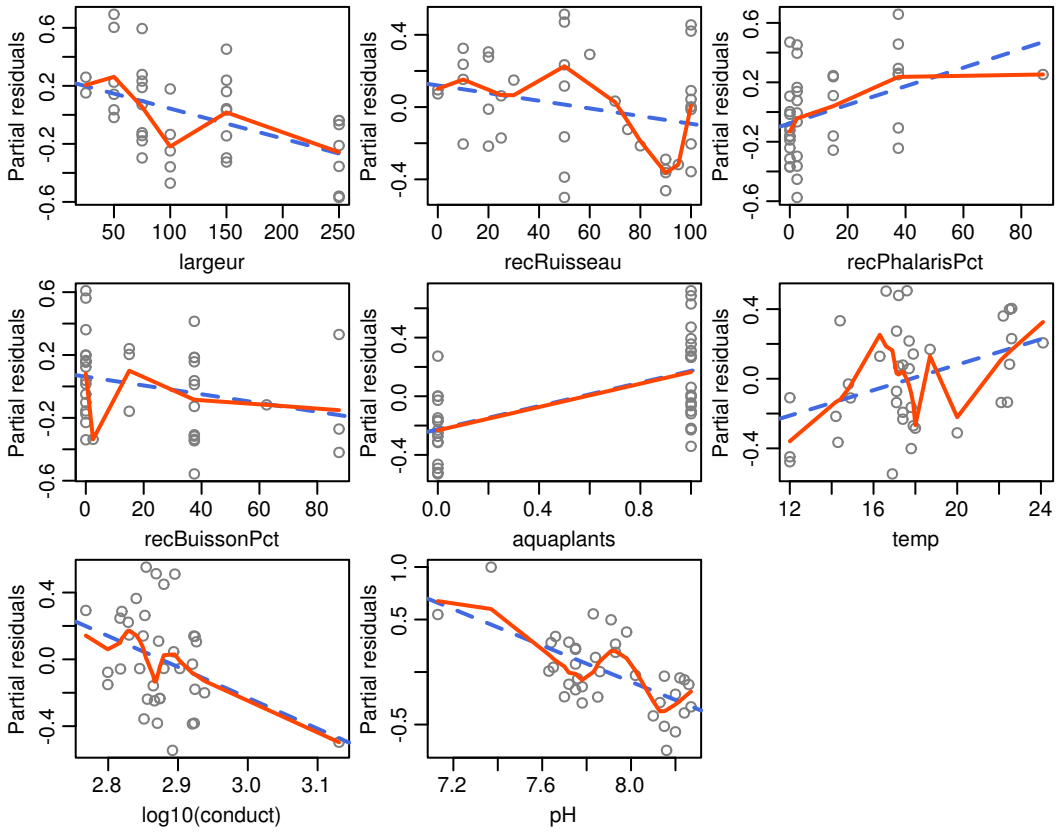
Le modèle a un R^2 de 0.81, il explique donc ~80 % de la variance observée dans le nombre d'individus (sur une échelle logarithmique).

```
d$lognb <- log10((d$nb*100/d$longueur) + 1)
mod <- lm(lognb ~  largeur + recRuisseau + recPhalarisPct +
              recBuissonPct + aquaplants + temp + log10(conduct) + pH, data=d)
diagplot(mod)
```



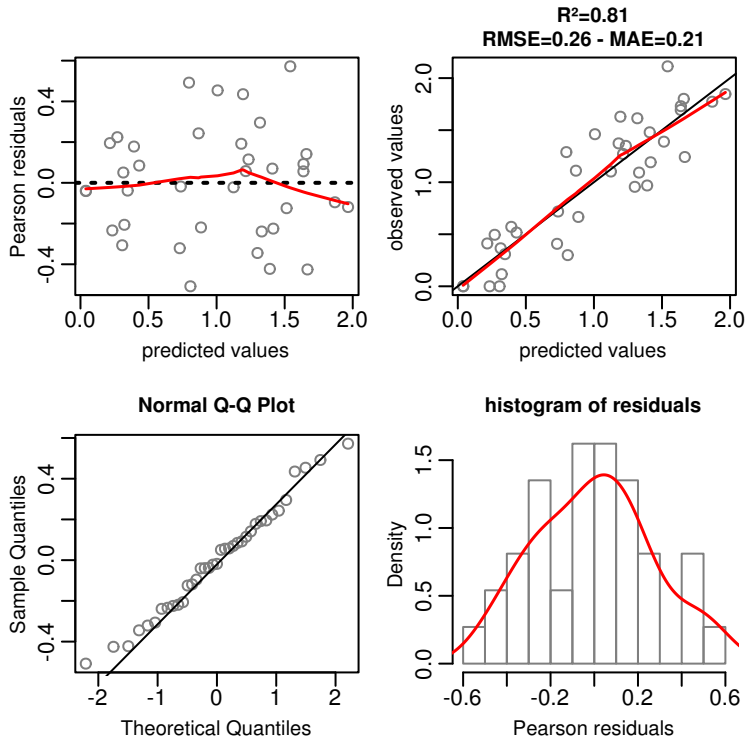
Linéarité et points extrèmes - Q10b
On peut maintenant vérifier si les relations entre y et les x sont à peu près linéaires avec des graphiques résidus vs variables explicatives.
Il n'y a pas de non linéarité flagrante. La variable recPhalarisPct montre une relation positive et des résidus légèrement bombés vers le haut ce qui appelle par exemple à une transformation racine carrée.

```
# dev.new(width = 14/2.54, height = 11/2.54)
diagplot2(mod, mar=c(3,3,0.1,0.1))
```

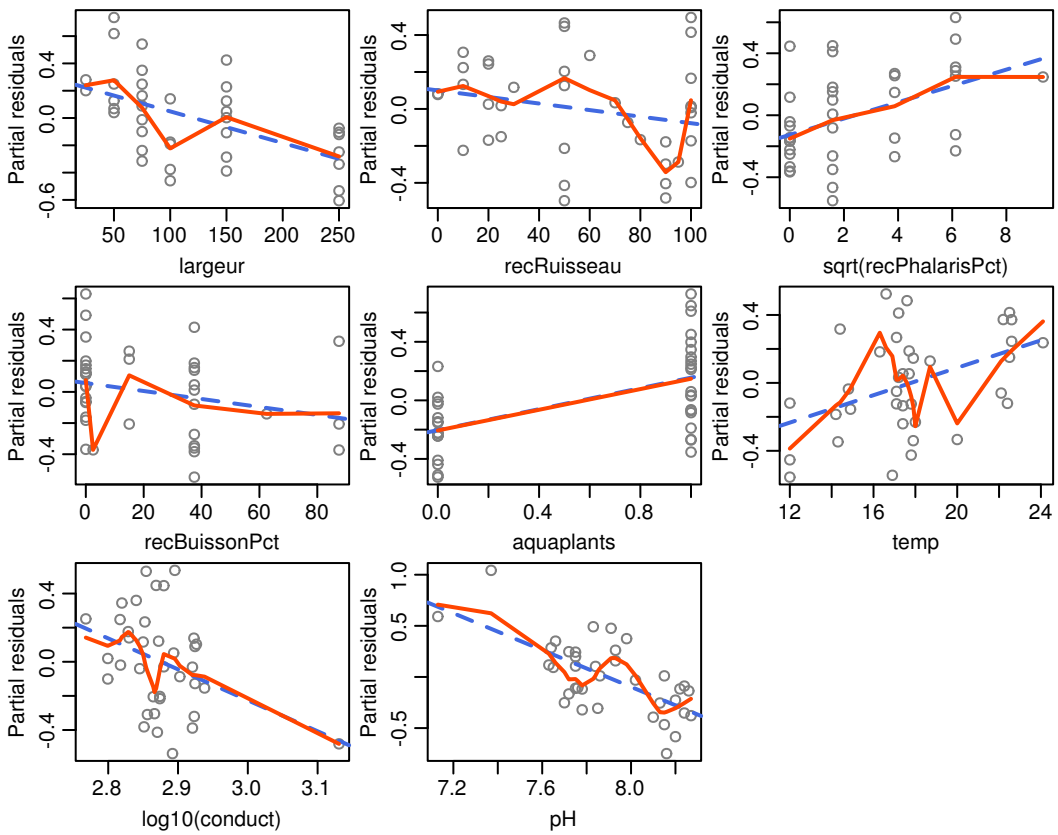


Après transformation racine carrée, la relation lognb ~ recPhalarisPct est effectivement plus linéaire. Les prédiction du modèle sont très légèrement améliorées (Root Mean Squared Error - RMSE - légèrement plus faible).
Cet effet est surtoût dû à un seul point avec un recouvrement en Phalaris >80% (le fameux point F05 repéré dans les ordinations et clustering). Il est donc possible qu'on "overfitte" les données ici avec cette transformation.

```
mod <- lm(lognb ~  (largeur + recRuisseau + sqrt(recPhalarisPct) +
             recBuissonPct + aquaplants + temp + log10(conduct) + pH), data=d)
diagplot(mod)
```




```
# dev.new(width = 14/2.54, height = 11/2.54)
diagplot2(mod, mar=c(3,3,0.1,0.1))
```



Pour la variable conductivité, on remarque que malgré la transformation log il y a toujours un point extrème qui semble tirer la relation vers le bas. Il s'agit du point F14 (voir ordinations et clustering). Il faudrait peut-être réestimer les modèle sans ce point pour voir à quel point il change les résultats (inférences etc. . .).

Les mesures et les graphiques d'influence peuvent être utiles ici...

On voit que le point F14 a effectivement une valeur différente des autres points (hat value élevée) mais lorsqu'on l'enlève du jeu de données on voit qu'il a un effet limité sur l'estimation des coefficients en général (Distance de Cook faible) et en particulier sur le coefficient de la variable "conductivité (dfbetas pas particulièrement élevés).

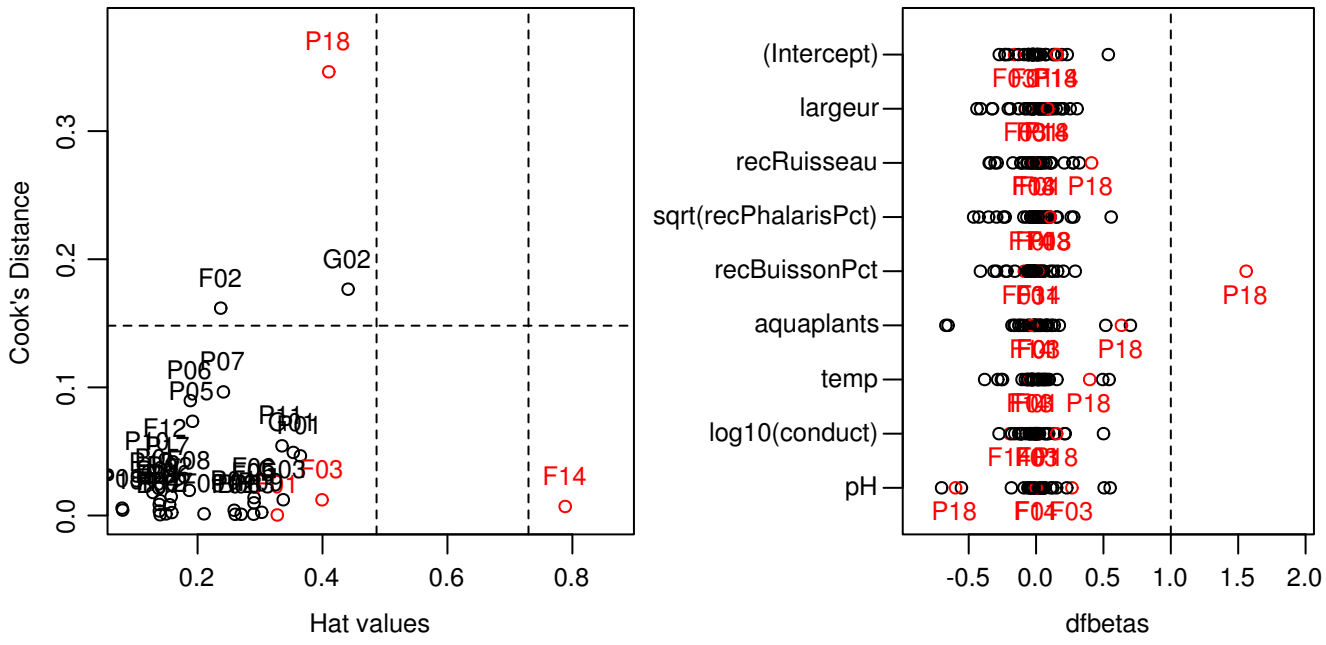
Le drain P18 montre un cas inverse. Il n'a pas une valeur particulièrement exceptionnelle (hat value faible) mais il semble avoir une certaine influence sur l'estimation des coefficients (Cook's distance élevée) et en particulier sur le coefficient de la variable resBuissonPct (dfbeta élevé). L'effet du recouvrement buisson serait plus fort (plus négatif ici) en l'absence de cette observation. Le drain P18 est en effet un point avec un recouvrement en buisson important mais avec un nombre moyen de libellules alors que tous les autres drains avec beaucoup de buisson ont en général des nombres beaucoup plus réduits de libellules. Il n'y a pas de raison de s'inquiéter particulièrement avec ce point qui montre simplement la réalité du terrain. Les cas où on a une combinaison de hat value élevée et distance de Cook élevée sont plus problématiques.

Rappel: la distance de Cook et les dfbetas seront pratiquement "aveugles" si on a 2 ou plusieurs points influents similaires. D'où l'utilité une fois de plus des graphiques de résidus en plus de ces mesures.

```
summary(influence.measures(mod))
```

```
## Potentially influential observations of
##   lm(formula = lognb ~ (largeur + recRuisseau + sqrt(recPhalarisPct) +      recBuissonPct + aquaplants +
##
##     dfb.1_ dfb.lrgr dfb.rcRs dfb.s(PP dfb.rcBP dfb.aqpl dfb.temp dfb.l10( dfb.pH dffit   cov.r
## F01 -0.02   0.02     0.04     0.01     0.01     0.02     0.01     0.02     0.01  -0.06    2.06_*
## F03 -0.16  -0.08    -0.01     0.10    -0.09     0.02    -0.03     0.01     0.27  -0.33    2.19_*
## F14  0.16   0.08    -0.01    -0.04     0.02    -0.03    -0.05    -0.19     0.00  -0.25    6.52_*
## P18  0.15   0.09     0.41     0.10     1.56_*   0.63     0.40     0.15    -0.60   1.89_*  0.49
##     cook.d hat
## F01  0.00   0.33
## F03  0.01   0.40
## F14  0.01   0.79_*
## P18  0.35   0.41
```

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mfrow = c(1,2))
influence.plot(mod)
dfbetas.plot(mod)
```



Multicolinéarité - Q10c
On avait déjà éliminé les paires devariables trop corrélées (colinéaires) on peut maintenant vérifier à quel point chaque variable explicative est liée à l'ensemble des autres variables explicatives.
On calcule les VIFs - Variance Inflation Factors du modèle.

```
vif(mod)
```

```
##              largeur          recRuisseau sqrt(recPhalarisPct)        recBuissonPct
##             2.104715             2.011331             1.375883             2.092159
##           aquaplants                 temp       log10(conduct)                   pH
##             2.021036             1.801765             1.922520             2.693854
```

Chaque erreur standard du modèle est "gonflée" (inflated) à cause de la multicolinéarité selon un facteur correspondant à la racine carrée des VIFs. Le VIF le plus important ici est celui du pH dont l'erreur standard est 1.6 trop grande par rapport à ce qu'elle serait sans colinéarité. Ceci parait très raisonable. Lorsque ce chiffre est trop grand les p valeurs sont trop élevées et on risque de "rater" des effets qui sont significatifs et de plus l'erreur standard étant plus élevée l'estimation des coefficients du modèle est moins bonne. NB : lorsque le modèle contient des variables qualitatives la fonction `vif` donne une colonne GVIF (genrealized VIF) que l'on peut dirrectement untiliser pour l'interprétation (sans en prendre la racine carrée).

```
sqrt(vif(mod))
```

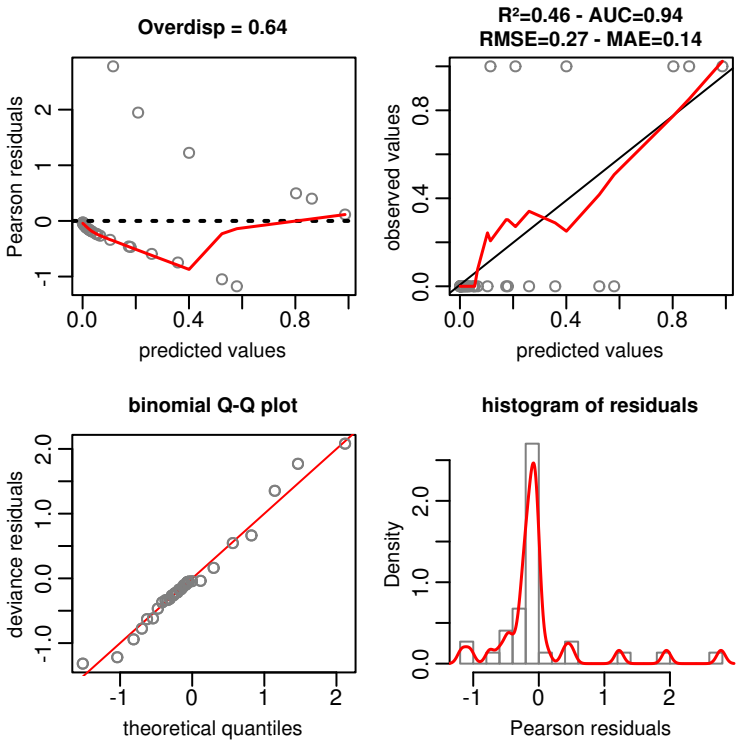
```
##              largeur          recRuisseau sqrt(recPhalarisPct)        recBuissonPct
##             1.450764             1.418214             1.172980             1.446430
##           aquaplants                 temp       log10(conduct)                   pH
##             1.421631             1.342298             1.386550             1.641296
```

Modèle pour la présence/absence de comportement de ponte

On utilise un modèle binomial comme prévu avec les mêmes variables explicatives + la longueur du drain puisqu'ici elle n'est pas incorporée dans la variable dépendante.

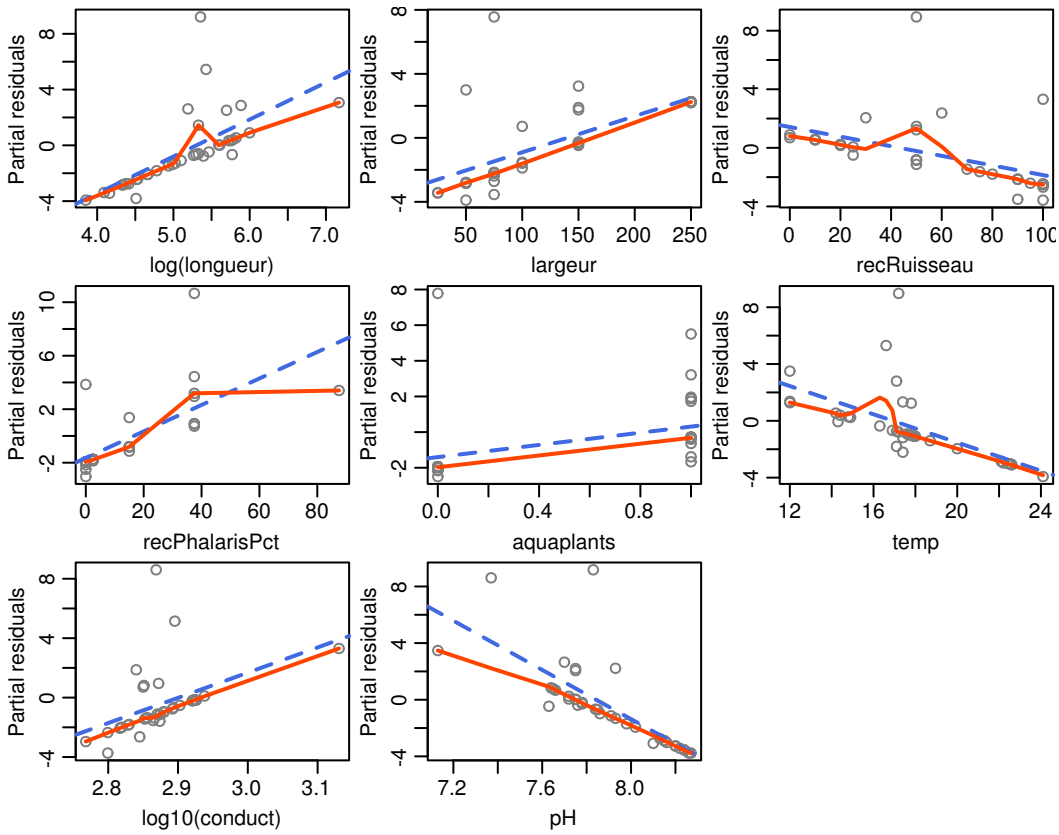
Comme on peut s'y attendre avec des données binaires, le coefficient de surdispersion tourne autour de 1 et ne pose donc pas de problème et la distribution des résidus suit bien une distribution binomiale.

```
mod <- glm(ponte ~  log(longueur) + largeur + recRuisseau + recPhalarisPct +
               aquaplants + temp + log10(conduct) + pH, data=d, family = binomial)
diagplot(mod)
```



Les plots de résidus des modèles binomiaux sont toujours plus difficiles à interpréter en particulier quand il y a très peu de présences comme ici.
On ne voit pas de problème flagrant à part peut-être pour la variable recPhalaris. Des transformations sqrt ou log n'améliorent pas vraiment la situation et provoquent une explosion dans les VIFs. On préfère laisser le modèle tel quel.

```
# dev.new(width = 14/2.54, height = 11/2.54)
diagplot2(mod, mar=c(3,3,0.1,0.1))
```



Les VIFs sont ici plus élevés uniquement à cause de l'ajout de la variable "longueur". Il n'y a pas vraiment de valeur seuil pour définir des VIFs problématiques. On voit parfois des recommendations de ne pas dépasser des valeurs de 5 à 10. On commence à être limite ici... Le risque est d'avoir des coefficients importants non significatifs et moins bien estimés à cause de la colinéarité.

```
vif(mod)
```

```
##  log(longueur)        largeur    recRuisseau recPhalarisPct     aquaplants           temp
##       3.029945       2.509805       4.525255       2.820037       2.046517       2.553692
## log10(conduct)             pH
##       4.179643       3.906843
```

```
sqrt(vif(mod))
```

```
##  log(longueur)        largeur    recRuisseau recPhalarisPct     aquaplants           temp
##       1.740674       1.584236       2.127265       1.679297       1.430565       1.598028
## log10(conduct)             pH
##       2.044418       1.976574
```

Une possibilité serait de passer la longeur comme un offset (son coefficient est fixé à 1) Ce qui reviendrait à estimer un "odd ratio" du comportement de pente standardisé pour une longeur de 100m de drain. Cependant c'est une pratique courrante pour les modèles de Poisson mais l'interprétation en terme de probabilité de ponte pour un modèle binomial est douteuse.

```
tmp <- glm(ponte ~  offset(log(longueur) -log(100)) + largeur + recRuisseau + recPhalarisPct +
             aquaplants + temp + log10(conduct) + pH, data=d, family = binomial)
vif(tmp)
```

```
##        largeur    recRuisseau recPhalarisPct     aquaplants           temp log10(conduct)
##       1.630076       2.937842       1.578753       1.742330       2.199624       2.510014
##             pH
##       2.809231
```

Inférence et sélection de modèle

Tests d'hypothèse nulle

> Q11 - Réalisez un test d'hypothèse nulle pour chaque variable de votre modèle. Quelle est l'hypothèse précise que vous testez ? Quel problème peut-on rencontrer lorsqu'on réalise de tels tests avec un grand nombre de variables explicatives ?

Nombre d'individus

Ici le test de wald donné par `summary` et la table d'analyse de la variance (test de F) demandé à `Anova` sont strictement identiques (modèle gaussien et pas de variables qualitatives). Attention à ne pas utiliser `anova` mais bien `Anova` du package car (ou `drop1(m, test = "F")`)

```
modnb <- lm(lognb ~  pH + aquaplants + sqrt(recPhalarisPct) + largeur + temp +
               recRuisseau + recBuissonPct + log10(conduct), data=d)
summary(modnb)
```

```
##
## Call:
## lm(formula = lognb ~ pH + aquaplants + sqrt(recPhalarisPct) +
##     largeur + temp + recRuisseau + recBuissonPct + log10(conduct),
##     data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50810 -0.21907 -0.01815  0.17820  0.57208
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          12.6989480  4.1970803   3.026  0.00527 **
## pH                   -0.9008903  0.3154525  -2.856  0.00800 **
## aquaplants            0.3584028  0.1426791   2.512  0.01805 *
## sqrt(recPhalarisPct)  0.0526418  0.0229702   2.292  0.02965 *
## largeur              -0.0023182  0.0009984  -2.322  0.02773 *
## temp                  0.0403800  0.0214861   1.879  0.07064 .
## recRuisseau          -0.0017729  0.0020099  -0.882  0.38525
## recBuissonPct        -0.0025578  0.0027054  -0.945  0.35252
## log10(conduct)       -1.8145311  1.1669086  -1.555  0.13118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3024 on 28 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.8148, Adjusted R-squared:  0.7619
## F-statistic:  15.4 on 8 and 28 DF,  p-value: 2.154e-08
```

```
Anova(modnb)
```

```
## Anova Table (Type II tests)
##
## Response: lognb
##                       Sum Sq Df F value   Pr(>F)
## pH                   0.74604  1  8.1560 0.007998 **
## aquaplants           0.57717  1  6.3099 0.018048 *
```

```
## sqrt(recPhalarisPct) 0.48042  1  5.2521 0.029647 *
## largeur              0.49314  1  5.3912 0.027735 *
## temp                 0.32307  1  3.5320 0.070641 .
## recRuisseau          0.07117  1  0.7780 0.385250
## recBuissonPct        0.08176  1  0.8939 0.352518
## log10(conduct)       0.22118  1  2.4180 0.131180
## Residuals            2.56119 28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

le pH, la présence de plantes aquatiques le recouvrement en Phalaris et la largeur sont considérés comme significatifs au seuil 0.05. Pour le pH par exemple on estime que si il n'y avait aucune relation entre le pH et le nombre de libellules, et qu'on recommençait 1000 fois l'expérience, on obtiendrait un coefficient pour le pH aussi grand ou plus grand seulement dans 8 cas. On considère donc qu'il est très peu probable que l'effet du pH observé ici soit dû uniquement au hasard de l'échantillonnage.

Le summary fourni aussi un test statistique pour l'intercept. On teste ici l'hypothèse que lorsque toutes les autres variables explicatives sont fixées à 0, l'abondance moyenne des libellules est 0. Ce test n'a donc pas beaucoup d'intérrêt ici...

Un des problèmes avec ce genre d'approches est que lorsque le nombre de variables explicatives est trop grand par rapport au nombre d'observations, les erreurs standard augmentent (le modèle devient plus variable) et la puissance statistique diminue (overfitting). Par exemple ici si on remplace les variables quantitatives recPhalarisPct et recBuissonPct par leur équivalent qualitatif, le nombre de paramètres total augmente (seulement 4 paramètres en plus) et il n'y a pratiquement plus aucune variable explicative significative.

```
tmp <- lm(lognb ~  pH + aquaplants + recPhalaris + largeur + temp +
                recRuisseau + recBuisson + log10(conduct), data=d)
summary(tmp)
```

```
##
## Call:
## lm(formula = lognb ~ pH + aquaplants + recPhalaris + largeur +
##     temp + recRuisseau + recBuisson + log10(conduct), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44989 -0.13907 -0.00597  0.17416  0.57818
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.3773684  4.7821433   2.170   0.0401 *
## pH               -0.6747195  0.3857502  -1.749   0.0931 .
## aquaplants        0.2911053  0.1848289   1.575   0.1283
## recPhalaris<5%    0.1744304  0.1661531   1.050   0.3043
## recPhalaris5-25%  0.3000599  0.2153515   1.393   0.1763
## recPhalaris>25%   0.3369636  0.1675120   2.012   0.0556 .
## largeur          -0.0022548  0.0010851  -2.078   0.0486 *
## temp              0.0416895  0.0226996   1.837   0.0787 .
## recRuisseau      -0.0002219  0.0025095  -0.088   0.9303
## recBuisson<25%   -0.2233637  0.2341837  -0.954   0.3497
## recBuisson25-50% -0.3432890  0.2272892  -1.510   0.1440
## recBuisson>50%   -0.3331927  0.2571110  -1.296   0.2073
## log10(conduct)   -1.6294465  1.2194071  -1.336   0.1940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3134 on 24 degrees of freedom
```

```
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.8295, Adjusted R-squared:  0.7443
## F-statistic: 9.733 on 12 and 24 DF,  p-value: 1.585e-06
```

On peut aussi vérifier que anova donne des résultats bien différents et qui dépendent de l'ordre des variables dans le modèle : à éviter !

```
tmp <- lm(lognb ~  pH + aquaplants + sqrt(recPhalarisPct) + largeur + temp +
              recRuisseau + recBuissonPct + log10(conduct), data=d)
anova(tmp)
```

```
## Analysis of Variance Table
##
## Response: lognb
##                       Df Sum Sq Mean Sq F value    Pr(>F)
## pH                     1 8.0623  8.0623 88.1404 3.805e-10 ***
## aquaplants             1 1.5852  1.5852 17.3299 0.0002711 ***
## sqrt(recPhalarisPct)   1 0.5247  0.5247  5.7365 0.0235477 *
## largeur                1 0.2743  0.2743  2.9983 0.0943574 .
## temp                   1 0.4797  0.4797  5.2446 0.0297541 *
## recRuisseau            1 0.0430  0.0430  0.4704 0.4984381
## recBuissonPct          1 0.0757  0.0757  0.8279 0.3706495
## log10(conduct)         1 0.2212  0.2212  2.4180 0.1311802
## Residuals             28 2.5612  0.0915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Anova(tmp)

# même modèle mais variables dans un autre ordre
tmp <- lm(lognb ~  temp + recRuisseau + recBuissonPct + log10(conduct) +
              pH + aquaplants + sqrt(recPhalarisPct) + largeur, data=d)
anova(tmp)
```

```
## Analysis of Variance Table
##
## Response: lognb
##                       Df Sum Sq Mean Sq F value    Pr(>F)
## temp                   1 0.7420  0.7420  8.1119 0.0081493 **
## recRuisseau            1 1.7463  1.7463 19.0913 0.0001550 ***
## recBuissonPct          1 4.7583  4.7583 52.0200 7.515e-08 ***
## log10(conduct)         1 1.1068  1.1068 12.0998 0.0016672 **
## pH                     1 1.5084  1.5084 16.4908 0.0003568 ***
## aquaplants             1 0.6582  0.6582  7.1957 0.0121217 *
## sqrt(recPhalarisPct)   1 0.2529  0.2529  2.7653 0.1074855
## largeur                1 0.4931  0.4931  5.3912 0.0277348 *
## Residuals             28 2.5612  0.0915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Anova(tmp)
```

Présence de comportement de ponte

On peut d'abbord voir que la probabilité d'observer un comportement de ponte est très significativement lié à l'abondance des libellules sur le tronçon. Etudier la présence/absence de comportement de ponte n'est donc pas très informatif ici.

On peut cependant se dire que le nombre d'adultes est vraisembablement un bon indice de l'intérrêt du drain comme zone de reproduction.
On continuera avec ce modèle binomial ici surtoût pour montrer les différences avec un modèle gaussien...

```
tmp <- glm(ponte ~  lognb, data=d, family = binomial)
Anova(tmp)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: ponte
##       LR Chisq Df Pr(>Chisq)
## lognb   8.5201  1   0.003512 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ici les test de wald du `summary` n'est pas identique au test de rapport de vraissemblance donné par `Anova`

```
modponte <- glm(ponte ~  recPhalarisPct + pH + log10(longueur)  + largeur +  recRuisseau +
                    aquaplants + temp + log10(conduct), data=d, family = binomial)
summary(modponte)
```

```
##
## Call:
## glm(formula = ponte ~ recPhalarisPct + pH + log10(longueur) +
##     largeur + recRuisseau + aquaplants + temp + log10(conduct),
##     family = binomial, data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3170  -0.3349  -0.1660  -0.0531   2.0812
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -3.52565   62.72338  -0.056   0.9552
## recPhalarisPct   0.07681    0.03993   1.924   0.0544 .
## pH              -7.31589    4.32644  -1.691   0.0908 .
## log10(longueur)  5.23865    3.33752   1.570   0.1165
## largeur          0.02528    0.01962   1.288   0.1976
## recRuisseau     -0.03382    0.04066  -0.832   0.4054
## aquaplants       1.65177    1.94257   0.850   0.3952
## temp            -0.42935    0.35629  -1.205   0.2282
## log10(conduct)  17.73741   18.01464   0.985   0.3248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32.800  on 36  degrees of freedom
## Residual deviance: 16.674  on 28  degrees of freedom
##   (7 observations deleted due to missingness)
## AIC: 34.674
##
## Number of Fisher Scoring iterations: 7
```

```r
Anova(modponte)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: ponte
##                 LR Chisq Df Pr(>Chisq)
## recPhalarisPct    5.3657  1    0.02054 *
## pH                4.3974  1    0.03599 *
## log10(longueur)   4.0352  1    0.04456 *
## largeur           1.9955  1    0.15777
## recRuisseau       0.6034  1    0.43728
## aquaplants        0.7800  1    0.37714
## temp              1.7991  1    0.17982
## log10(conduct)    1.0355  1    0.30887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Il n'y a que trois variables très proches du seuil de significativité : recPhalarisPct, pH et log10(longueur).

Sélection stepwise

> Q12 - Réalisez une sélection du meilleur modèle par une sélection par étape sur base de l'AIC (stepwise AIC selection). Quelles sont les variables sélectionnées par cette méthode ? Quel(s) problème(s) peut-on rencontrer avec cette approche.

Attention dès qu'on travaille avec des AIC il faut se méfier des valeurs manquantes. En effet les AIC ne sont valides que pour comparer des modèles basés sur des y identiques. Lorsqu'il y a des valeurs manquantes le glm var éliminer les lignes avec des valeurs manquantes dans certains modèles mais pas dans d'autres (si la variable avec des NA n'est pas dans le modèle). Il est donc plus prudent de nettoyer le jeu de données en éliminant les valeurs manquantes avant d'estimer le modèle.

```
dtmp <- d[, c("lognb", "ponte", "longueur", "pH" , "aquaplants" , "recPhalarisPct" , "largeur" , "temp" ,
              "recRuisseau" , "recBuissonPct" , "conduct")]
dtmp <- na.omit(dtmp)

modnb <- lm(lognb ~  pH + aquaplants + sqrt(recPhalarisPct) + largeur + temp +
                recRuisseau + recBuissonPct + log10(conduct), data=dtmp)
modponte <- glm(ponte ~  recPhalarisPct + pH + log10(longueur)  + largeur +  recRuisseau +
                    aquaplants + temp + log10(conduct), data=dtmp, family = binomial)
```

Nombre d'individus

La méthode stepwise a seulement éliminé 2 variables : recRuisseau et recBuissonPct et conservé les 6 autres dans le modèle final.

```
step(modnb)
```

```
## Start:  AIC=-80.81
## lognb ~ pH + aquaplants + sqrt(recPhalarisPct) + largeur + temp +
##     recRuisseau + recBuissonPct + log10(conduct)
##
##                         Df Sum of Sq    RSS     AIC
## - recRuisseau            1   0.07117 2.6324 -81.792
## - recBuissonPct          1   0.08176 2.6430 -81.644
## <none>                               2.5612 -80.806
## - log10(conduct)         1   0.22118 2.7824 -79.742
## - temp                   1   0.32307 2.8843 -78.411
## - sqrt(recPhalarisPct)   1   0.48042 3.0416 -76.446
## - largeur                1   0.49314 3.0543 -76.291
## - aquaplants             1   0.57717 3.1384 -75.287
## - pH                     1   0.74604 3.3072 -73.348
##
## Step:  AIC=-81.79
## lognb ~ pH + aquaplants + sqrt(recPhalarisPct) + largeur + temp +
##     recBuissonPct + log10(conduct)
##
##                         Df Sum of Sq    RSS     AIC
## - recBuissonPct          1   0.10796 2.7403 -82.305
## <none>                               2.6324 -81.792
## - log10(conduct)         1   0.17622 2.8086 -81.395
## - temp                   1   0.44057 3.0729 -78.067
## - largeur                1   0.48127 3.1136 -77.580
## - sqrt(recPhalarisPct)   1   0.48981 3.1222 -77.478
## - aquaplants             1   0.51436 3.1467 -77.189
## - pH                     1   0.67813 3.3105 -75.311
```

```
##
## Step:  AIC=-82.31
## lognb ~ pH + aquaplants + sqrt(recPhalarisPct) + largeur + temp +
##     log10(conduct)
##
##                        Df Sum of Sq    RSS     AIC
## <none>                              2.7403 -82.305
## - log10(conduct)        1   0.16080 2.9011 -82.195
## - largeur               1   0.57699 3.3173 -77.235
## - aquaplants            1   0.59178 3.3321 -77.071
## - temp                  1   0.59570 3.3360 -77.027
## - sqrt(recPhalarisPct)  1   0.68043 3.4208 -76.099
## - pH                    1   1.26501 4.0053 -70.262


##
## Call:
## lm(formula = lognb ~ pH + aquaplants + sqrt(recPhalarisPct) +
##     largeur + temp + log10(conduct), data = dtmp)
##
## Coefficients:
##          (Intercept)                    pH            aquaplants  sqrt(recPhalarisPct)
##            12.194570             -0.994212               0.349515               0.060124
##              largeur                  temp        log10(conduct)
##            -0.002471              0.051125              -1.502176
```

Présence de comportement de ponte

Après la procédure de sélection automatique on se retrouve avec les 3 variables qui étaient significatives selon le Test de Rapport de Vraissemblance : recPhalarisPct, pH et log10(longueur) et en plus la variable "largeur".

```
step(modponte)
```

```
## Start:  AIC=34.67
## ponte ~ recPhalarisPct + pH + log10(longueur) + largeur + recRuisseau +
##     aquaplants + temp + log10(conduct)
##
##                   Df Deviance    AIC
## - recRuisseau      1   17.278 33.278
## - aquaplants       1   17.454 33.454
## - log10(conduct)   1   17.710 33.710
## - temp             1   18.473 34.473
## - largeur          1   18.670 34.670
## <none>                 16.674 34.674
## - log10(longueur)  1   20.709 36.709
## - pH               1   21.072 37.072
## - recPhalarisPct   1   22.040 38.040
##
## Step:  AIC=33.28
## ponte ~ recPhalarisPct + pH + log10(longueur) + largeur + aquaplants +
##     temp + log10(conduct)
##
##                   Df Deviance    AIC
## - aquaplants       1   18.153 32.153
## - temp             1   18.947 32.946
## - log10(conduct)   1   19.267 33.267
## <none>                 17.278 33.278
```

```
## - largeur          1   20.254 34.254
## - pH               1   21.085 35.085
## - recPhalarisPct   1   23.297 37.297
## - log10(longueur)  1   23.969 37.969
##
## Step:  AIC=32.15
## ponte ~ recPhalarisPct + pH + log10(longueur) + largeur + temp +
##     log10(conduct)
##
##                   Df Deviance    AIC
## - temp             1   19.367 31.367
## - log10(conduct)   1   19.380 31.380
## <none>                 18.153 32.153
## - largeur          1   21.334 33.334
## - pH               1   23.628 35.628
## - recPhalarisPct   1   23.756 35.756
## - log10(longueur)  1   25.036 37.036
##
## Step:  AIC=31.37
## ponte ~ recPhalarisPct + pH + log10(longueur) + largeur + log10(conduct)
##
##                   Df Deviance    AIC
## - log10(conduct)   1   20.100 30.100
## <none>                 19.367 31.367
## - largeur          1   22.393 32.393
## - pH               1   24.922 34.922
## - recPhalarisPct   1   25.119 35.119
## - log10(longueur)  1   25.726 35.726
##
## Step:  AIC=30.1
## ponte ~ recPhalarisPct + pH + log10(longueur) + largeur
##
##                   Df Deviance    AIC
## <none>                 20.100 30.100
## - largeur          1   23.109 31.109
## - pH               1   25.117 33.117
## - recPhalarisPct   1   25.271 33.271
## - log10(longueur)  1   26.095 34.095


##
## Call:  glm(formula = ponte ~ recPhalarisPct + pH + log10(longueur) +
##     largeur, family = binomial, data = dtmp)
##
## Coefficients:
##     (Intercept)   recPhalarisPct               pH  log10(longueur)          largeur
##        29.89412          0.06042         -6.23317          5.74759          0.02377
##
## Degrees of Freedom: 36 Total (i.e. Null);  32 Residual
## Null Deviance:       32.8
## Residual Deviance: 20.1  AIC: 30.1
```

Le risque avec ces approches est de ne pas sélectionner réellement le modèle avec l'AIC le plus bas mais de tomber sur un optimum local. L'autre problème est qu'on ne garde qu'un seul modèle final alors qu'il y en a sans doute de nombreux autres qui sont quasi aussi bons. On a aussi utilisé ici l'AIC alors qu'il vaut mieux utiliser l'AICc en particulier pour les petits nombres d'observations comme ici.

Sélection de modèles

> Q13 - Calculez tous les modèles possibles, leur AICc et le poids du modèle (model AICc weight). Calculez l'importance relative de chaque variable (variable AICc weight) et les model averaged coefficients. Comparez les résultats avec les deux approches précédentes.

> Q14 - Examinez les coefficients (model averaged coefficients) des variables qui semblent les plus importantes. Comment pouvez-vous interpréter ces coefficients (sans faire de calculs il s'agit juste d'une interprétation très grossière à ce stade) ?

Nombre d'individus

Tous les calculs se font avec la commande suivante et sont stockés dans une liste de 3 éléments. Le premier élément contient un tableau avec tous les modèles, leur AICc leurs différences d'AICc et les "model weights" (AICc.w). Le deuxième contient les variable weights (w) et les variables sont classées par ordre décroissant d'importance. Dans le troisième slot on trouve les "model averaged coefficients" et de nouveau les "variables weights" mais le tableau est dans le même ordre que les coefficients du modèle par défaut.

Vous pouvez aussi utiliser le package `MumIn` ou `AICcmodavg` pour arriver au même résultat.

```
resnb <- model.select(modnb)
```

On peut afficher les 20 meilleurs modèles.
Le meilleur modèle (avec l'AICc le plus faible) contient les mêmes variables que celles qui étaient significatives dans le test de rapport de vraissemblance (y compris "temp" avec p = 0.07).
Cependant si on recommençait l'étude plusieurs fois on estime qu'que ce modèle serait le meilleur modèle seulement dans 13.8% des cas (model weight : AICc.w = 0.138). L'incertitude sur le meilleur modèle est donc assez grande. Le premier modèle est seulement 1.79 fois mieux supporté par les données que le deuxième (rapport des AICc.w : 0.138/0.077).

Il y a cependant seulement un seul autre modèle avec une différence d'AICc (delta.AICc) plus petite que 2. Ce modèle contient en plus la variable explicative "conductivité".

```
resnb[[1]][1:20,-c(2:5)]
```

```
##                                                                                  model   AICc
## 32                                  pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp 28.668
## 160                pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp+ log10(conduct) 29.839
## 96                  pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp+ recBuissonPct 30.750
## 16                                       pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur 31.268
## 8                                           pH+ aquaplants+ sqrt(recPhalarisPct) 31.290
## 64                    pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp+ recRuisseau 31.396
## 72                                  pH+ aquaplants+ sqrt(recPhalarisPct)+ recBuissonPct 31.486
## 80                       pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ recBuissonPct 31.691
## 224 pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp+ recBuissonPct+ log10(conduct) 31.876
## 192   pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp+ recRuisseau+ log10(conduct) 32.024
## 48                          pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ recRuisseau 32.120
## 68                                                 pH+ aquaplants+ recBuissonPct 32.332
## 24                                     pH+ aquaplants+ sqrt(recPhalarisPct)+ temp 32.625
## 196                                   pH+ aquaplants+ recBuissonPct+ log10(conduct) 32.643
## 40                                  pH+ aquaplants+ sqrt(recPhalarisPct)+ recRuisseau 33.025
## 200                pH+ aquaplants+ sqrt(recPhalarisPct)+ recBuissonPct+ log10(conduct) 33.110
## 136                                 pH+ aquaplants+ sqrt(recPhalarisPct)+ log10(conduct) 33.236
## 4                                                                      pH+ aquaplants 33.568
## 112          pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ recRuisseau+ recBuissonPct 33.610
## 152                          pH+ aquaplants+ sqrt(recPhalarisPct)+ temp+ log10(conduct) 33.628
##     AICc.delta AICc.w sum.w
```

```
## 32       0.000  0.138 0.138
## 160      1.171  0.077 0.215
## 96       2.081  0.049 0.264
## 16       2.600  0.038 0.301
## 8        2.621  0.037 0.338
## 64       2.728  0.035 0.374
## 72       2.817  0.034 0.407
## 80       3.023  0.030 0.438
## 224      3.208  0.028 0.466
## 192      3.356  0.026 0.491
## 48       3.452  0.025 0.516
## 68       3.664  0.022 0.538
## 24       3.957  0.019 0.557
## 196      3.975  0.019 0.576
## 40       4.357  0.016 0.592
## 200      4.442  0.015 0.607
## 136      4.567  0.014 0.621
## 4        4.899  0.012 0.633
## 112      4.942  0.012 0.644
## 152      4.960  0.012 0.656
```

Avec l'AIC plutôt que l'AICc on retrouve bien le même modèle qu'avec la méthode stepwise. La méthode stepwise ne trouvait donc pas un optimum local mais bien le meilleur modèle dans ce cas. En utilisant l'AICc qui est plus adapté à des petits jeux de données on trouve cependant un modèle différent.

```
tmp <-  model.select(modnb, srt = "AIC")
tmp[[1]][1,-c(2:5)]
```

```
##                                                                    model    AIC AIC.delta AIC.w
## 160 pH+ aquaplants+ sqrt(recPhalarisPct)+ largeur+ temp+ log10(conduct) 24.696         0 0.109
##     sum.w
## 160 0.109
```

Les variables peuvent être classées par ordre d'importance avec le "variable weight". Les valeurs confirment encore les observations précédentes : pH, aquaplants, sqrt(recPhalarisPct) et largeur on un poids w >0.6. Si on recommençait 1000 fois la récolte de données, on estime par exemple que le pH se retrouverait dans le meilleur modèle 972 fois.

```
resnb[[2]]
```

```
##                      freq     w
## (Intercept)           1.0 1.000
## pH                    0.5 0.972
## aquaplants            0.5 0.926
## sqrt(recPhalarisPct)  0.5 0.784
## largeur               0.5 0.643
## temp                  0.5 0.542
## recBuissonPct         0.5 0.413
## log10(conduct)        0.5 0.407
## recRuisseau           0.5 0.256
```

On peut aussi obtenir les "model averaged coefficients" (av.coef) (qui sont la moyenne des valeurs d'un des coefficients de tous les modèles pondérée par le poids de ce modèle AICc.w). Ils ont été ici triés pour apparaître par ordre décroissant d'importance (w). On voit que le nombre de libellules diminue quand le pH et la largeur augmentent (signe du coefficient négatif). Les libellules sont par contre plus nombreuses en présence de plantes aquatiques et lorsque que recouvrement en Phalaris augmente.

```
resnb <- model.select(modnb, dec = 4)
resnb[[3]][rev(order(resnb$mod.av$w)),1:4]
```

```
##                       freq      w av.coef  av.se
## (Intercept)           1.0 1.0000 10.7371 4.2272
## pH                    0.5 0.9724 -1.0275 0.3212
## aquaplants            0.5 0.9260  0.3784 0.1383
## sqrt(recPhalarisPct)  0.5 0.7843  0.0437 0.0212
## largeur               0.5 0.6425 -0.0014 0.0009
## temp                  0.5 0.5424  0.0212 0.0156
## recBuissonPct         0.5 0.4135 -0.0018 0.0016
## log10(conduct)        0.5 0.4066 -0.7336 0.6646
## recRuisseau           0.5 0.2557 -0.0005 0.0007
```

Il est difficile de dire laquelle des ces variables explicatives supportées par les données est celle qui a le plus d'influence sur l'abondance des libellules, d'une part à cause du fait que le nonbre de libellule est sur une échelle logarithmique (en base 10) et d'autres part parce que toutes ces variables explicatives sont dans des unités différentes. On peut standardiser les variables explicatives pour pouvoir les comparer dirrectement. On voit que çà ne change pas les résultats (w identiques dans les deux cas) mais on peut comparer les coefficients. Il n'y a pas de surprise dans ce cas, le pH et la présence de plantes aquatiques sont bien les deux variables a avoir l'effet le plus importants sur le nombre de libellules. Le recouvrement en Phalaris et la largeur ont un effet deux fois moins importants que la présence de plantes aquatiques (coefficient deux fois plus petit). On voit aussi que le coefficeint pour la largeur n'est pas estimé très précisément (en tenant compte de la sélection de modèles) : son erreur standard non conditionnelle (av.se) est de 0.062 pour un coefficient de -0.099.

```
d2 <- d[, c("lognb", "pH" , "aquaplants" , "recPhalarisPct" , "largeur" , "temp" ,
                    "recRuisseau" , "recBuissonPct" , "conduct")]
d2$sqrtrecPhalarisPct <- sqrt(d2$recPhalarisPct)
d2$log10conduct <- log10(d2$conduct)
d2[,-1] <- scale(d2[,-1])
tmp <- lm(lognb ~  pH + aquaplants + sqrtrecPhalarisPct + largeur + temp +
                recRuisseau + recBuissonPct + log10conduct, data=d2)
res <- model.select(tmp, dec = 4)
res[[3]][rev(order(res$mod.av$w)),1:4]
```

```
##                     freq      w av.coef  av.se
## (Intercept)          1.0 1.0000  0.9920 0.0559
## pH                   0.5 0.9818 -0.2721 0.0849
## aquaplants           0.5 0.9254  0.1900 0.0695
## sqrtrecPhalarisPct   0.5 0.7856  0.1095 0.0531
## largeur              0.5 0.6403 -0.0992 0.0620
## temp                 0.5 0.5476  0.0675 0.0493
## log10conduct         0.5 0.4105 -0.0444 0.0401
## recBuissonPct        0.5 0.4082 -0.0458 0.0411
## recRuisseau          0.5 0.2563 -0.0176 0.0235
```

Présence de comportement de ponte

L'incertitude sur le meilleur modèle est ici plus grande. Les 10 meilleurs modèles sont très proches (delta.AICc <2.2) et contiennent entre 1 et 5 variables explicatives. Le meilleur modèle a seulement 4.5% de chance d'être sélectionné comme meilleur modèle si on pouvait recollecter de nouvelles donénes.

```
responte <- model.select(modponte)
responte[[1]][1:20,-c(2:5)]
```

```
##                                                                  model   AICc AICc.delta AICc.w sum.w
## 16                  recPhalarisPct+ pH+ log10(longueur)+ largeur 32.035      0.000  0.040 0.040
## 6                               recPhalarisPct+ log10(longueur) 32.162      0.127  0.038 0.078
## 8                             recPhalarisPct+ pH+ log10(longueur) 32.359      0.324  0.034 0.112
## 24              recPhalarisPct+ pH+ log10(longueur)+ recRuisseau 33.256      1.220  0.022 0.134
## 2                                                  recPhalarisPct 33.347      1.312  0.021 0.155
## 4                                            recPhalarisPct+ pH 33.680      1.645  0.018 0.173
## 38                   recPhalarisPct+ log10(longueur)+ aquaplants 33.745      1.710  0.017 0.190
## 32     recPhalarisPct+ pH+ log10(longueur)+ largeur+ recRuisseau 33.839      1.804  0.016 0.206
## 20                              recPhalarisPct+ pH+ recRuisseau 33.913      1.878  0.016 0.222
## 70                         recPhalarisPct+ log10(longueur)+ temp 34.095      2.059  0.014 0.236
## 3                                                            pH 34.127      2.091  0.014 0.250
## 144 recPhalarisPct+ pH+ log10(longueur)+ largeur+ log10(conduct) 34.167      2.131  0.014 0.264
## 80             recPhalarisPct+ pH+ log10(longueur)+ largeur+ temp 34.180      2.144  0.014 0.278
## 136          recPhalarisPct+ pH+ log10(longueur)+ log10(conduct) 34.329      2.293  0.013 0.291
## 14                     recPhalarisPct+ log10(longueur)+ largeur 34.367      2.331  0.013 0.303
## 134              recPhalarisPct+ log10(longueur)+ log10(conduct) 34.398      2.362  0.012 0.316
## 22                  recPhalarisPct+ log10(longueur)+ recRuisseau 34.446      2.411  0.012 0.328
## 15                               pH+ log10(longueur)+ largeur 34.521      2.486  0.012 0.339
## 72                    recPhalarisPct+ pH+ log10(longueur)+ temp 34.524      2.488  0.012 0.351
## 7                                          pH+ log10(longueur) 34.587      2.552  0.011 0.362
```

Seuls recPhalarisPct et pH ont un w>0.6, qui reste cependant faible (0.62, 0.7). De plus, le coefficient de pH est estimé avec une très grande imprécision (erreur standard de 2.1 pour un coefficient de -2.9). Seul le recouvrement en Phalaris semble - faiblement - supporté par les données.
On peut noter le contraste entre cette conclusion et la conclusion qu'on aurrait eue si on utilisait le meilleur modèle seul qui comprend 4 variables explicatives : recPhalarisPct, pH, log10(longueur) et largeur. Il faut se rappeler aussi que les VIFs sont un peu élevés ici ce qui ajoute de l'incertitude dans les estimations (les erreurs standard sont "gonflées" à cause de la colinéarité).

```
responte[[3]][rev(order(responte$mod.av$w)),1:4]
```

```
##                 freq     w av.coef  av.se
## (Intercept)      1.0 1.000   8.903 27.740
## recPhalarisPct   0.5 0.691   0.037  0.023
## log10(longueur)  0.5 0.585   2.293  1.679
## pH               0.5 0.556  -2.568  1.962
## largeur          0.5 0.337   0.005  0.006
## recRuisseau      0.5 0.325  -0.007  0.009
## aquaplants       0.5 0.319   0.467  0.583
## temp             0.5 0.309  -0.067  0.081
## log10(conduct)   0.5 0.247   1.481  3.164
```

Interprétation des résultats

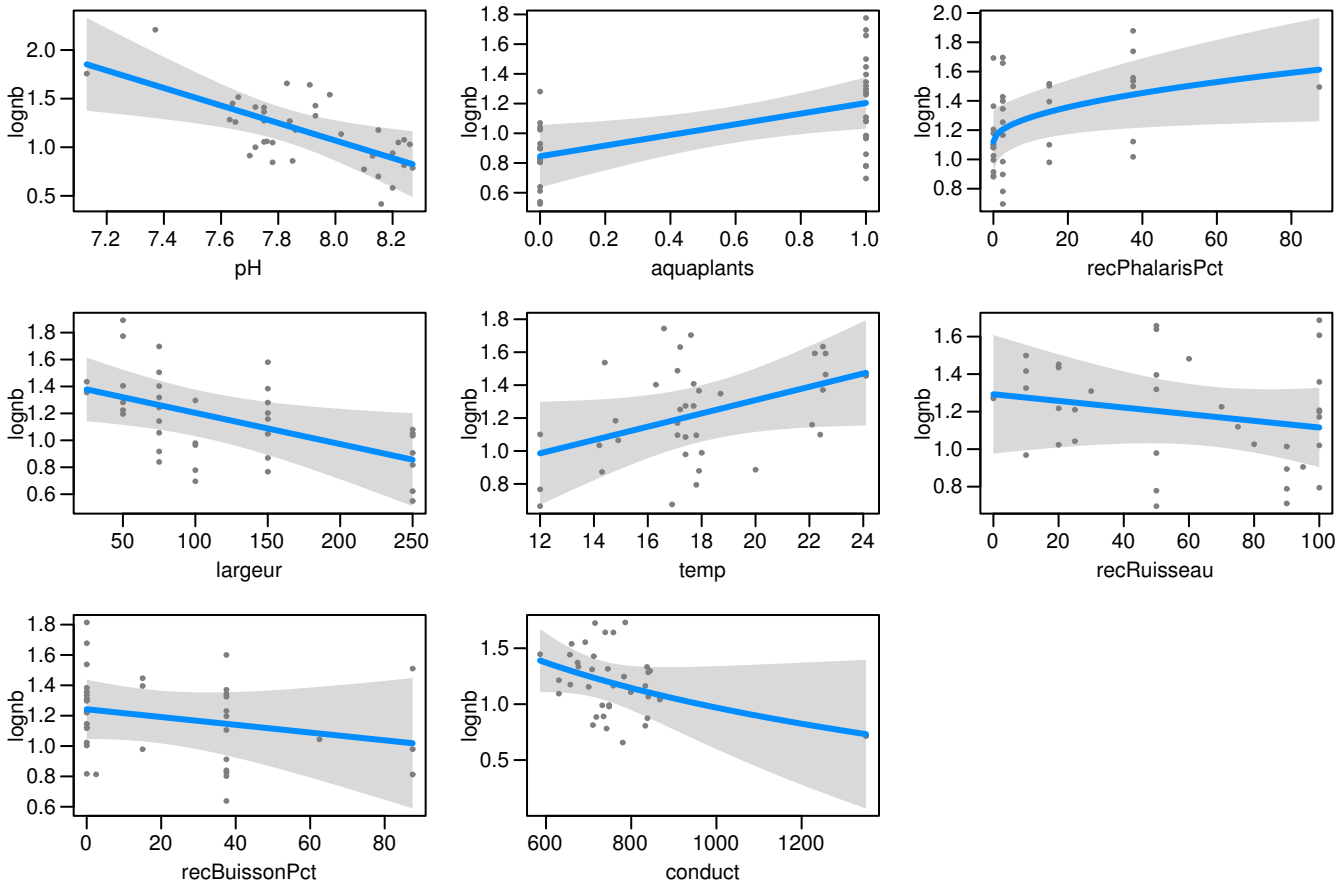
> Q15 - Faites une représentation graphique de vos données et de votre modèle et interprétez les résultats en termes biologiques. Concentrez-vous sur les variables les plus importantes. Il peut être aussi intéressant de discuter les variables qui ne sont pas "significatives" sur le plan statistique mais qui montrent malgré tout une relation avec la variable réponse. Il faut alors s'interroger sur la raison qui fait que ces variables ne sont pas supportées par les données.

Abondance des adultes

On peut facilement visualiser les résultats du modèle complet avec visreg. Par défaut visreg nous montre des résidus partiels (pas les valeurs réellement observées) et les intervales de confiance autour de la droite. L'axe des y représente

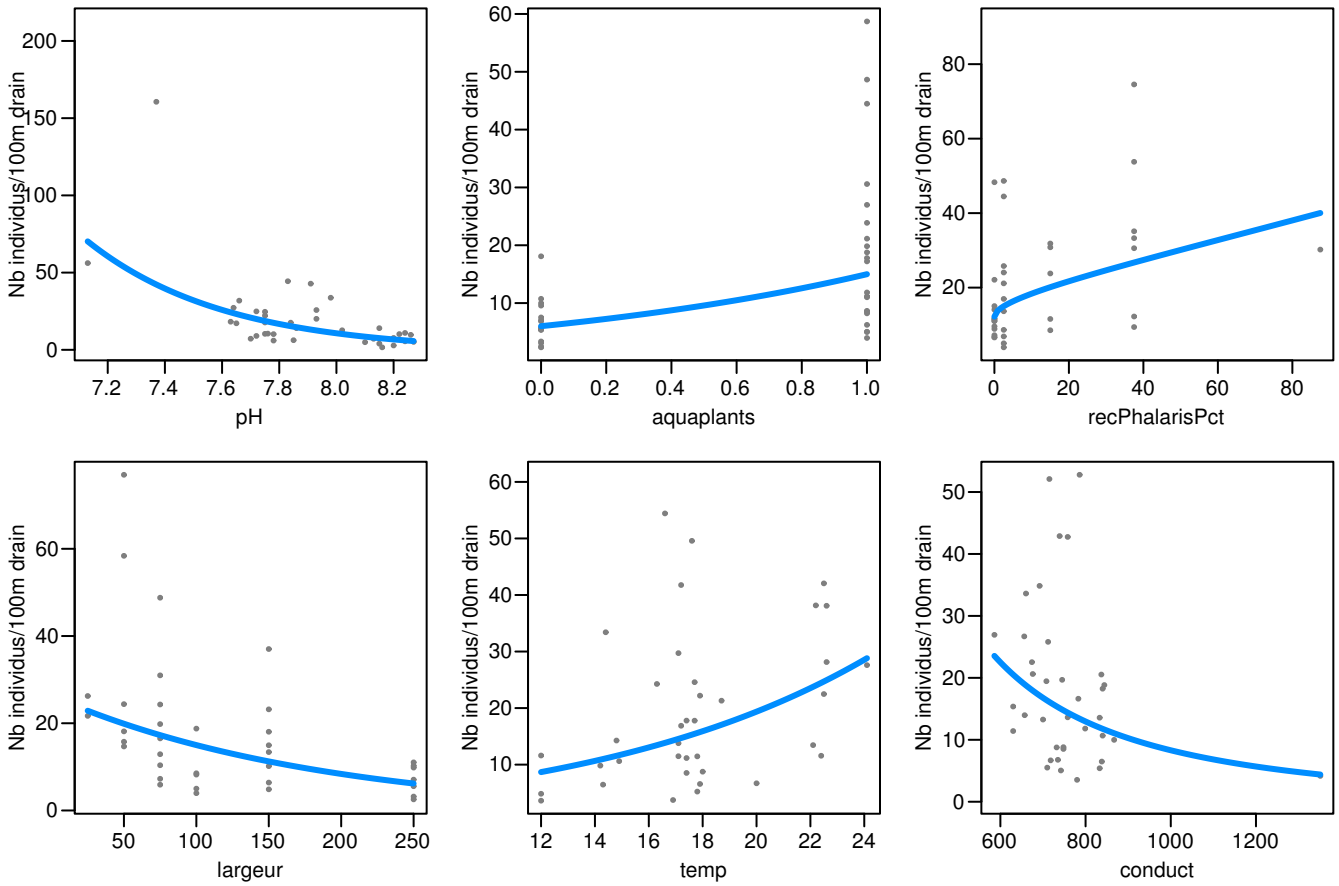
des log de nombre d'individus par 100m de ruisseau ce qui n'est pas très facile à interpréter sur le plan biologique. Les 3 dernières variables (recRuisseau, recBuisson et conduct) sont celles qui sont peu supportées par les données (w <0.41) et leur pente semble en effet très faible en particulier pour recBuissonPct et recRuisseau. On ne représentera plus ces deux dernières par la suite pour gagner de la place.

```
# dev.new(width = 18/2.54, height = 12/2.54)
par(mar=c(3,3,1,1), mgp = c(1.75, 0.6, 0), mfrow = c(3,3))
visreg(modnb)
```



Pour visualiser les résultats sur l'échelle d'origine il faut rétrotransformer les y avec : (10^y)-1. On a retiré les intervales de confiance pour rendre les graphiques plus lisibles.

```
# dev.new(width = 16/2.54, height = 9/2.54)
par(mar=c(3,3,1,1), mgp = c(1.75, 0.6, 0), mfrow = c(2,3))
visreg(modnb, trans = function(x) {(10^x)-1}, partial = TRUE, band = FALSE,
       xvar = c("pH", "aquaplants", "recPhalarisPct", "largeur", "temp", "conduct"),
       ylab = "Nb individus/100m drain")
```

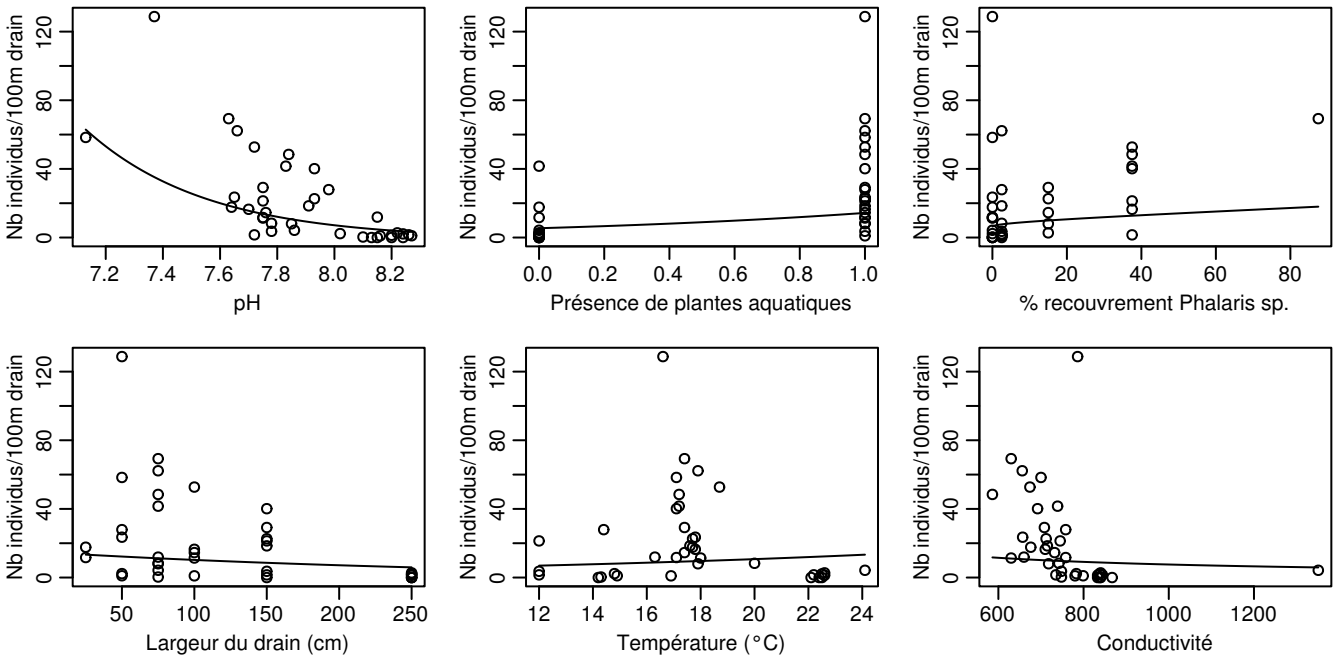


Idéalement il faudrait représenter les modèles sur base des "model averaged coefficients". Viserg ne peut pas le faire mais on peut le faire à la main...
Ici on fixe les variables à leur moyenne (visreg prends la médiane) que l'on répète 100 fois. Ensuite on fait varier la variable X que l'on veut représenter entre sa valeur minimale et maximale.

```
# copie des données nécessaires et élimination des lignes avec des NA
dtmp <- d[, c("lognb", "pH" , "aquaplants" , "recPhalarisPct" , "largeur" , "temp" ,
              "recRuisseau" , "recBuissonPct" , "conduct")]
row.names(dtmp) <- d$code2
dtmp <- na.omit(dtmp)
X <- apply(dtmp, 2, mean) # moyenne pour chaque variable
X <- as.data.frame(matrix(rep(X, each = 100), ncol = ncol(dtmp), byrow = FALSE)) # répétée 100 fois
X[,1] <- 1 # intercept
colnames(X) <- colnames(dtmp)
X[,"recPhalarisPct"] <- sqrt(X[,"recPhalarisPct"])
X[,"conduct"] <- log10(X[,"conduct"])

# séquence de 100 nombres entre le minimum et le maximum pour chaque colonne.
Xseq <- as.data.frame(apply(dtmp, 2, function(x) {seq(min(x), max(x), length.out = 100)}))

# matrice vide pour collecter les valeurs prédites
pred <- X
pred[] <- NA

# coefficient (averaged)
Beta <- resnb[[3]]$av.coef

# prédiction pour le pH puis les 5 variables suivantes
Xtmp <- X
```

```
Xtmp$pH <- Xseq$pH
pred$pH <- 10^(as.matrix(Xtmp) %*% Beta) - 1

Xtmp <- X
Xtmp$aquaplants <- Xseq$aquaplants
pred$aquaplants <- 10^(as.matrix(Xtmp) %*% Beta) - 1

Xtmp <- X
Xtmp$recPhalarisPct <- sqrt(Xseq$recPhalarisPct)
pred$recPhalarisPct <- 10^(as.matrix(Xtmp) %*% Beta) - 1

Xtmp <- X
Xtmp$largeur <- Xseq$largeur
pred$largeur <- 10^(as.matrix(Xtmp) %*% Beta) - 1

Xtmp <- X
Xtmp$temp <- Xseq$temp
pred$temp <- 10^(as.matrix(Xtmp) %*% Beta) - 1

Xtmp <- X
Xtmp$conduct <- log10(Xseq$conduct)
pred$conduct <- 10^(as.matrix(Xtmp) %*% Beta) - 1
```

```
# Graphiques
# dev.new(width = 16/2.54, height = 9/2.54)
par(mar=c(3,3,1,1), mgp = c(1.75, 0.6, 0), mfrow = c(2,3))

nb <- (10^dtmp$lognb)-1
plot(nb ~ dtmp$pH, xlab = "pH", ylab = "Nb individus/100m drain")
lines(pred$pH ~ Xseq$pH)

plot(nb ~ dtmp$aquaplants, xlab = "Présence de plantes aquatiques",
     ylab = "Nb individus/100m drain")
lines(pred$aquaplants ~ Xseq$aquaplants)

plot(nb ~ dtmp$recPhalarisPct, xlab = "% recouvrement Phalaris sp.",
     ylab = "Nb individus/100m drain")
lines(pred$recPhalarisPct ~ Xseq$recPhalarisPct)

plot(nb ~ dtmp$largeur, xlab = "Largeur du drain (cm)",
     ylab = "Nb individus/100m drain")
lines(pred$largeur ~ Xseq$largeur)

plot(nb ~ dtmp$temp, xlab = "Température (°C)",
     ylab = "Nb individus/100m drain")
lines(pred$temp ~ Xseq$temp)

plot(nb ~ dtmp$conduct, xlab = "Conductivité",
     ylab = "Nb individus/100m drain")
lines(pred$conduct ~ Xseq$conduct)
```



La visualisation de ces graphiques sur l'échelle d'origine et avec les valeurs observées peut donner l'impressionj que le modèle n'est pas très bon. On consultera la première série de graphiques sur échalle logarithmique et avec les résidus partiels à la place des valeurs observées pour se convaincre du contraire...
L'avantage est qu'on peut aussi consulter les valeurs prédites sous forme de chiffres ce qui facilite la description des graphiques qui sont un peu "écrasés" ici.

```
head(data.frame(pH = Xseq$pH, "nombre prédit" = pred$pH))
```

```
##         pH nombre.prédit
## 1 7.130000      62.91400
## 2 7.141515      61.19624
## 3 7.153030      59.52466
## 4 7.164545      57.89800
## 5 7.176061      56.31505
## 6 7.187576      54.77465
```

Interprétation

On essaye ici de faire une synthèse des résultats en décrivant le résultat de la sélection de modèle (quelles sont les variables importantes) mais aussi en décrivant les graphiques avec des chiffres biologicuement interprétables. On décrit les graphiques dans la gamme des valeurs les plus fréquentes, ce qui veut dire qu'on exclut parfois certaines valeurs plus extrèmes dans la description.

Trois variables semblent expliquer en partie l'abondance des libellules : le pH, la présence de plantes aquatiques et le taux de recouvrement du drain par la baldingère (*Phalaris arundinacea*). Dans une moindre mesure, l'abondance des adultes semble liée à la largeur du drain et à la température de l'eau.

On observe une plus grande abondance lorsque le pH est plus neutre et moins basique. Le nombre d'individus prédit passe d'une vingtaine individus par 100m de drain pour un pH de 7.6 à seulement 3 individus environs pour un pH de 8.2. Par ailleurs les deux drains avec le pH le plus faible (7.4 et 7.2) abbritent des nombres importants d'individus : 129 et 58 individus par 100m de drain respectivement.

Toutes choses étant égales par ailleurs, l'abondance prédite triple à peu près entre les drains sans plantes aquatiques (5.4 individus par 100m de drain) et les drains avec plantes aquatiques (14 individus)

De même la présence de baldingère semble favorable jusqu'à un certain point. Le nombre d'individus prédit double entre les drains sans baldingère (6.4 individus) et les drains avec 40% de couverture par cette graminée (13 individus).

L'abondance a tendance à être plus faible sur les drains les plus larges et là où la température mesurée est plus faible mais ces résultats sont nettement moins bien supportés par les données.

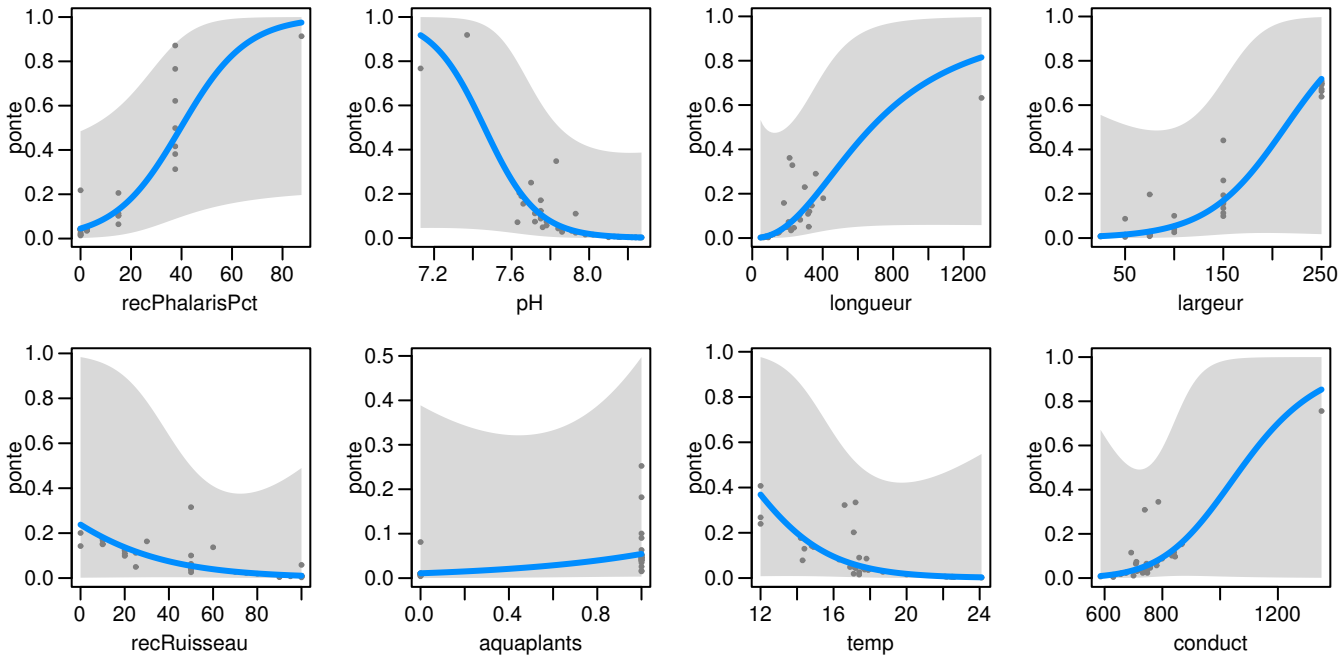
```
# Liste des valeurs prédites utilisées pour décrire les résultats en chiffres (non exécuté ici)
data.frame(nb = nb, "drain" = row.names(dtmp))
data.frame(pH = Xseq$pH, "nombre prédit" = pred$pH)
data.frame(pH = Xseq$aquaplant, "nombre prédit" = pred$aquaplant)
data.frame(pH = Xseq$recPhalarisPct, "nombre prédit" = pred$recPhalarisPct)
data.frame(pH = Xseq$largeur, "nombre prédit" = pred$largeur)
data.frame(pH = Xseq$temp, "nombre prédit" = pred$temp)
```

Présence de comportement de ponte

Le traitement est ici plus restreint car comme déjà évoqué à plusieurs reprises les résultats de ce modèle sont peu intéressants et peu supportés par les données étant donné le peu de comportements observés et la corrélation très forte avec l'abondance des adultes.

La seule variable qui semble liée (faiblement) aux comportements de ponte observés est le recouvrement en baldingère. La probabilité d'observer un comportement de ponte est d'environs 0 en absencede baldingère et passe à 0.4 environs lorsque cette plante recouvre 40% du drain. Ceci semble confirmer l'importance de cette plante déjà observée sur les données d'abondance.

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mar=c(3,3,1,1), mgp = c(1.75, 0.6, 0), mfrow = c(2,4))
visreg(modponte,  partial = TRUE, band = TRUE, scale = "response")
```



Discussion des résultats

Après avoir décrit de manière neutre mais biologiquement interprétable les résultats comme on l'a fait on peut passer à la discussion, moins neutre, plus spéculative en général. Il ne s'agit plus de l'analyse des données et on sort du cadre de ce travail. On ne procèdera donc pas à une discussion approfondie ici.

On peut noter cependant que l'importance de certaines variables semble logique.

La présence de plantes aquatiques peut servir de sites de ponte et de milieu de vie pour les larves. Les baldingères, avec leur structure élevée et leurs pieds généralement dans l'eau pourraient servir d'abbri pour les adultes (ea dortoirs

nocturnes) mais aussi potentiellement de site de sortie pour les larves qui doivent se hisser hors de l'eau au moment de la mue finale et de l'émergence de l'imago.

Le fait que le pH soit la variable qui explique le mieux l'abondance pose question notament vis à vis des réserves que l'on avait émises au départ sur la qualité de cette mesure réalisée un an plus tard. Il semble que même prise dans de mauvaises conditions le pH soit un bon prédicteur. Il faut aussi se rappeler de l'analyse exploratoire qui montrait que les sites avec le pH le plus basique (et donc les conditions les plus défavorables en apparence) se trouvaient à la sortie du village de Focant et souvent en présence de Buissons. On peut donc se demander quelle est l'influence de la pollution éventuelle et des feuilles des buissons sur le pH. Il est aussi tout à fait possible que le pH ne pose aucun problème pour les libellules mais soit simplement un indicateur de certaines conditions environnementales (pollution, matière organique,...).

Les mesures de températures semblent beaucoup moins fiables. Pourtant les résultats - faiblement supportés par les données - semblent en concordance avec la littérature qui fait de cette libellule une espèce plutôt thermophile.

On le voit ce genre d'analyse exploratoire d'études observatives peut être utile pour générer des hypothèses à aller tester sur le terrain par la récolte de nouvelles données.

Bonus ...

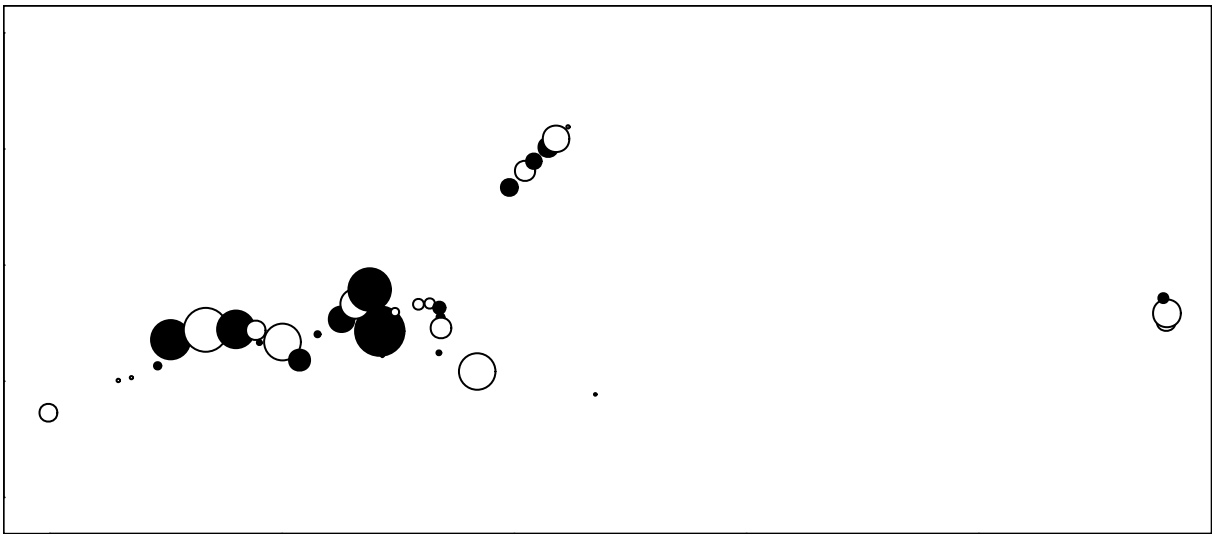
Quelques compléments, très brièvement, sans détails...

Vérification de l'absence de corrélation spatiale dans les résidus

```
d2 <- d[, c("lognb", "x", "y", "pH" , "aquaplants" , "recPhalarisPct" , "largeur" , "temp" ,
            "recRuisseau" , "recBuissonPct" , "conduct")]
row.names(d2) <- d$code2
d2 <- na.omit(d2)
mod <- modnb
```

Un graphique des résidus positionnés dans l'espace ne montre pas de groupes de points noirs (résidus positifs) ou blancs (résidus négatifs) ce qui est bon signe...

```
# dev.new(width = 16/2.54, height = 7/2.54)
mycol <- c("white", "black")[ifelse(resid(mod)<0, 1, 2)]
par(mar = c(0,0,0,0))
plot(d2$x, d2$y, pch = 21,  cex = 6*abs(resid(mod)), bg = mycol,
     asp = 1)
```



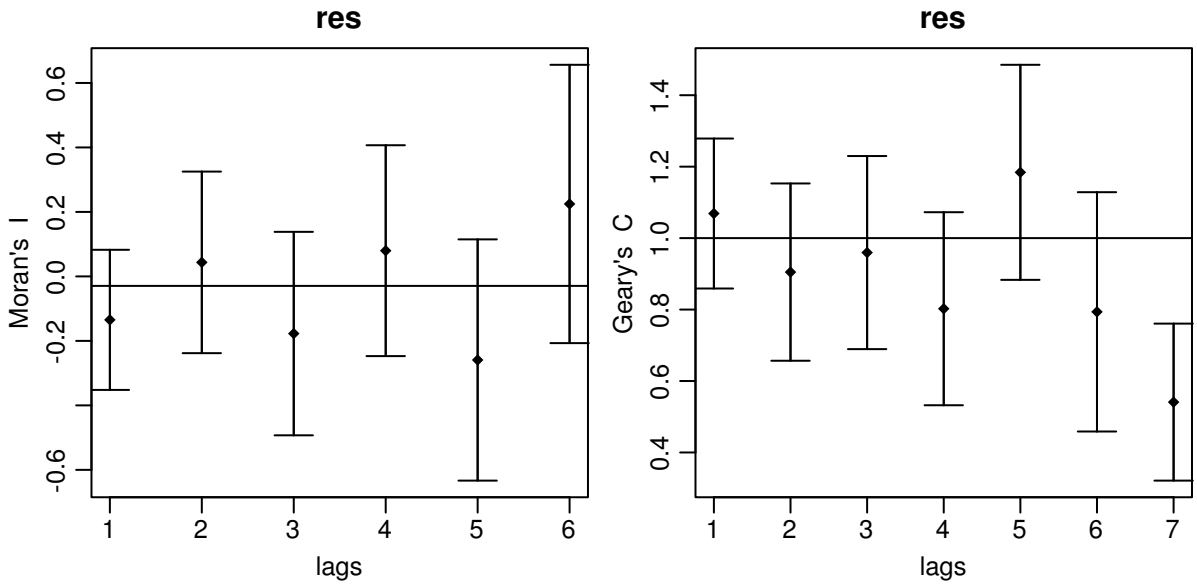
Vérification de la stationarité : pas de tendance globale Nord/sud ou est/ouest dans les résidus.

```
res <- resid(mod)
modres <- lm(res ~ x  * y , data=d2)
Anova(modres)
```

```
## Anova Table (Type II tests)
##
## Response: res
##            Sum Sq Df F value Pr(>F)
## x         0.07691  1  1.0472 0.3136
## y         0.03555  1  0.4840 0.4915
## x:y       0.05002  1  0.6811 0.4151
## Residuals 2.42358 33
```

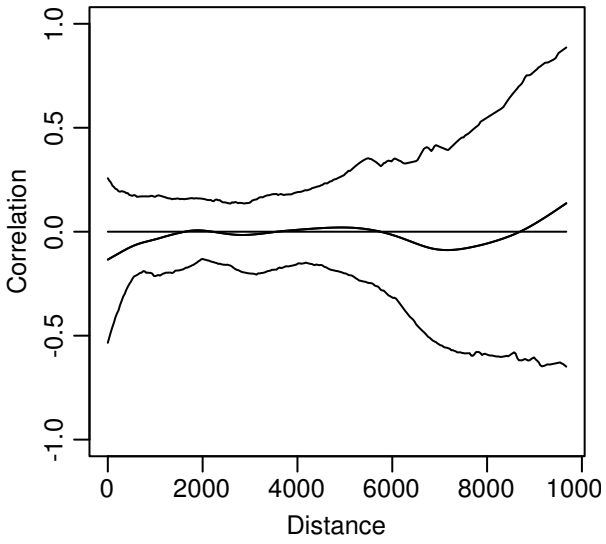
Les corrélogrammes confirment que les corrélations spatiales sont faibles et non significatives (par pas de 500 mètres)

```
# dev.new(width = 16/2.54, height = 8/2.54)
# Correlogram (moran's I) : no correlation related to distance (500m steps)
library(spdep)
xy <- as.matrix(d2[, c("x", "y")])
nb <- dnearneigh(xy, 0, 500)

par(mfrow = c(1,2), mar = c(3,3,2,0.5), mgp = c(1.8, 0.6, 0), cex = 0.8)
correlog <- sp.correlogram(nb, res,  method="I", order=6, zero.policy = TRUE)
# print(correlog, p.adj.method="holm")
plot(correlog)

correlog <- sp.correlogram(nb, res,  method="C", order=7, zero.policy = TRUE)
# print(correlog, p.adj.method="holm")
plot(correlog)
```



On peut aussi obtenir plus facilement un corrélograme lissé directement sur les résidus de la manière suivante

```
# dev.new(width = 8/2.54, height = 8/2.54)
library(ncf)
correlog <- spline.correlog(x = xy[,"x"], y = xy[,"y"], z = resid(mod), quiet = TRUE)
par(mar = c(3,3,2,0.5), mgp = c(1.8, 0.6, 0), cex = 0.8)
plot(correlog)
```



Approches alternatives

Generalized Additive model (GAM)

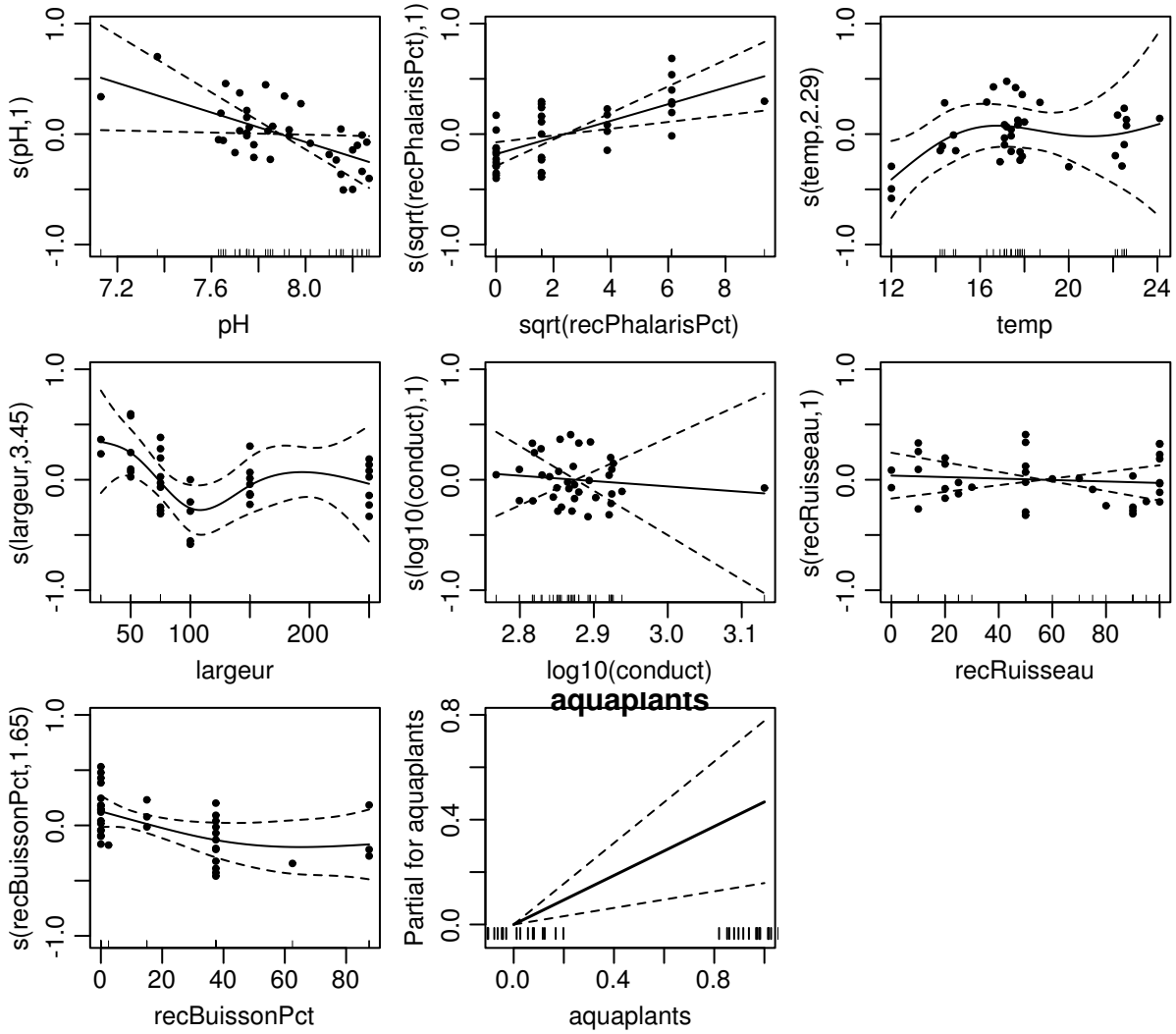
Une approche par GAM confirme l'importance du pH, de la présence de plantes aquatiques et du recouvrement en phalaris. De plus les degrés de liberté du pH et de sqrt(recPhalarisPct) sont = 1 ce qui signifie que la relation est bien linéaire telle qu'elle a été modélisée dans notre GLM. D'autres variables ont des degrés de liberté >1 ce qui indique une relation non linéaire mais ils ne sont pas significatifs pour autant après avoir ajouté une courbe de lissage.

```
library(mgcv)
k <- 5 # pas assez de données pour un lissage plus complexe
mod <- gam(lognb ~  s(pH, k = k) + aquaplants  +
               s(sqrt(recPhalarisPct), k = k) + s(temp, k = k) + s(largeur, k = k) +
               s(log10(conduct), k = k) + s(recRuisseau, k = k) +
               s(recBuissonPct, k = k), data=d)


summary(mod)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## lognb ~ s(pH, k = k) + aquaplants + s(sqrt(recPhalarisPct), k = k) +
##     s(temp, k = k) + s(largeur, k = k) + s(log10(conduct), k = k) +
##     s(recRuisseau, k = k) + s(recBuissonPct, k = k)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.71710    0.09742   7.361 1.48e-07 ***
## aquaplants   0.46767    0.15481   3.021  0.00597 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf Ref.df      F p-value
## s(pH)                   1.000  1.000  4.634 0.04160 *
```

```
## s(sqrt(recPhalarisPct)) 1.000  1.000 11.338 0.00251 **
## s(temp)                 2.286  2.762  1.480 0.14920
## s(largeur)              3.454  3.797  1.879 0.11486
## s(log10(conduct))       1.000  1.000  0.075 0.78726
## s(recRuisseau)          1.000  1.000  0.131 0.72039
## s(recBuissonPct)        1.653  1.971  1.580 0.21693
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.83   Deviance explained = 88.8%
## GCV = 0.10261  Scale est. = 0.065466  n = 37
```

```
# dev.new(width = 16/2.54, height = 14/2.54)
par(mfrow = c(3,3), mar = c(3,3,0.5,0.5), mgp = c(1.8, 0.6, 0), cex = 0.8)
plot(mod, residuals = TRUE, pch = 20, all.terms=TRUE,  cex = 0.75)
```



CART - Classification and Regression Tree

Une approche par arbre de régression confirme l'importancedu pH et de la présence de plantes aquatiques. Lorsque le pH est >= 8 on a un groupe de 13 drains avec une abondance moyenne de 1.1 individus. Lorsque le pH est plus faible, on a un groupe de 9 drains sans plantes aquatiques avec une moyenne de 7.5 individus et un groupe de 24 drains avec une moyenne de

```
library(rpart)
set.seed(3)
res <- rpart(lognb ~  largeur + tourniere + recRuisseau + recPhalarisPct +
                  recBuissonPct +   aquaplants + temp + conduct + pH, data=d)
print(res)
```

```
## n= 44
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 44 15.3222300 0.9860441
##    2) pH>=8 13  1.2192200 0.3226203 *
##    3) pH< 8 31  5.9818810 1.2642540
##      6) aquaplants< 0.5 9  1.8353600 0.9280295 *
##      7) aquaplants>=0.5 22  2.7128790 1.4018010
##       14) pH>=7.735 12  0.9869814 1.2611940 *
##       15) pH< 7.735 10  1.2039630 1.5705290 *
```

```
res$cptable
```

```
##           CP nsplit rel error    xerror      xstd
## 1 0.53002270      0 1.0000000 1.0739229 0.1586781
## 2 0.09356611      1 0.4699773 0.7564637 0.1566615
## 3 0.03406388      2 0.3764112 0.7435635 0.1598270
## 4 0.01000000      3 0.3423473 0.7496865 0.1714418
```

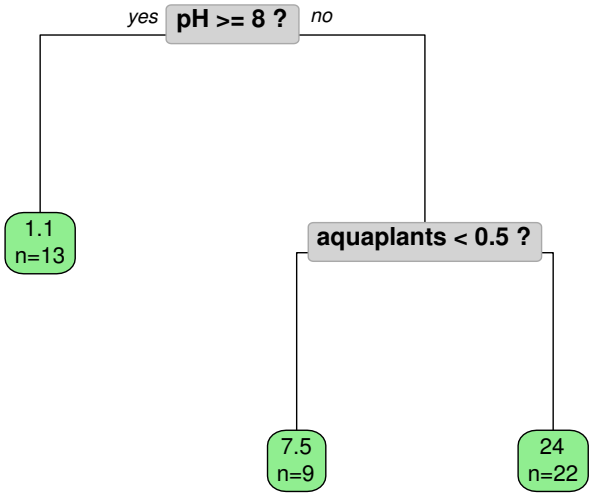
Il faut idéalement "élaguer" l'arbre pour éviter l'overfitting. On prend ici la dimension d'arbre qui minimise l'erreur de cross validation. Cependant les 3 derniers noeuds ont ici une erreur très proche et le résultat change à chaque fois qu'on réestime le modèle (d'où le set.seed ci-dessus de facçon à ce que les résultats soient reproductibels à chaque fois qu'on fait tourner le code). Il ne faut donc interpréter les noeuds après le premier qu'avec précaution. Il s'agit ici plus d'une analyse descriptive du jeu de données qu'un modèle prédictif.

```
res <- prune(res, cp= res$cptable[which.min(res$cptable[,"xerror"]),"CP"])
```

Les graphiques par défaut sont franchement laids. On peut nettement les améliorer avec le package `rpart.plot`

```
# dev.new(width = 10/2.54, height = 10/2.54)
library(rpart.plot)
res$frame$yval <- (10^res$frame$yval)-1
prp(res, main="Nombre d'individus par 100m de drain",
    ycompress=TRUE, extra=1, branch=1, cex=0.7, varlen=0, faclen = 0, digits=2,
    round=2, split.cex=1.1, split.round=.5, under = FALSE,
    box.col = "lightgreen", split.box.col="lightgray", split.border.col = "darkgray",
    yesno.yshift = 0.6, boxes.include.gap = TRUE,
    eq = " is ", lt = " < ", ge = " >= ", split.suffix=" ?", xsep=" | ")
```

Nombre d'individus par 100m de drain




```
## boxes.include.gap is TRUE
```