

Manuel de laboratoire

Biostatistiques appliquées avec R



BIO4558
Automne 2019
Julien Martin
Département de biologie
Université d'Ottawa

Exercices de laboratoire

Les exercices de laboratoire que vous retrouverez dans les pages qui suivent sont conçus de manière à vous permettre de développer une expérience pratique en analyse de données à l'aide d'un logiciel (R). R est un logiciel très puissant, mais comme tous les logiciels, il a des limites. En particulier il ne peut réfléchir à votre place, vous dire si l'analyse que vous tentez d'effectuer est appropriée ou sensée, ou interpréter biologiquement les résultats.

Quelques points importants à retenir

- Avant de commencer une analyse statistique, il faut d'abord vous familiariser son fonctionnement. Cela ne veut pas dire que vous devez connaître les outils mathématiques qui la sous-tendent, mais vous devriez au moins comprendre les principes utilisés lors de cette analyse. Avant de faire un exercice de laboratoire, lisez donc la section correspondante dans les notes de cours. Sans cette lecture préalable, il est très probable que les résultats produits par le logiciel, même si l'analyse a été effectuée correctement, seront indéchiffrables.
- Les laboratoires sont conçus pour compléter les cours théoriques et *vice versa*. À cause des contraintes d'horaires, il se pourrait que le cours et le laboratoire ne soient pas parfaitement synchronisés. N'hésitez donc pas à poser des questions sur le labo en classe ou des questions théoriques au laboratoire.
- Travaillez sur les exercices de laboratoire à votre propre rythme. Certains exercices prennent beaucoup moins de temps que d'autres et il n'est pas nécessaire de compléter un exercice par séance de laboratoire. En fait deux séances de laboratoire sont prévues pour certains des exercices. Même si vous n'êtes pas notés sur les exercices de laboratoire, soyez conscient que ces exercices sont essentiels. Si vous ne les faites pas, il est très peu probable que vous serez capable de compléter les devoirs et l'examen final. Prenez donc ces exercices de laboratoire au sérieux !
- Le premier laboratoire est conçu pour vous permettre d'acquérir ou de réviser le minimum de connaissances requises pour vous permettre de réaliser les exercices de laboratoires avec R. Il y a presque toujours de multiples façons de faire les choses avec R et vous ne trouverez ici que des méthodes simples. Ceux et celles d'entre vous qui y sont enclins pourront trouver en ligne des instructions plus détaillées et complexes. En particulier, je vous conseille :

R pour les débutants

http://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf

Using R for psychological research: A simple guide to an elegant package

<http://www.personality-project.org/r/>

An introduction to R

<http://cran.r-project.org/doc/manuals/R-intro.html>

Si vous préférez des manuels, le site web de CRAN en garde une liste commentée à :

<http://www.r-project.org/doc/bib/R-books.html>

Finalement, comme aide-mémoire à garder sous la main, je vous recommande :

R reference card par Tom Short

<http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

Qu'est-ce que R et pourquoi l'utiliser dans ce cours?

R est un logiciel libre et multiplateforme formant un système statistique et graphique. R est également un langage de programmation spécialisé pour les statistiques. C'est un dialecte du langage S. S-Plus est un autre dialecte, très semblable, et forme un produit commercial qui a une interface graphique que certains trouvent plus conviviale.

R a deux très grands avantages pour ce cours, et un inconvénient embêtant initialement mais qui vous forcera à acquérir des excellentes habitudes de travail.

Le premier avantage est que vous pouvez tous l'installer sur votre (ou vos) ordinateurs personnel gratuitement. C'est important parce que c'est à l'usage que vous apprendrez et maîtriserez réellement les biostatistiques et cela implique que vous devez avoir un accès facile et illimité à un logiciel statistique.

Le deuxième avantage est que R peut tout faire en statistiques. R est conçu pour être extensible et est devenu l'outil de prédilection des statisticiens mondialement. La question n'est plus : " Est-ce que R peut faire ceci? ", mais devient " Comment faire ceci avec R ". Et Google est votre ami.

Aucun autre logiciel n'offre ces deux avantages.

L'inconvénient embêtant initialement est que l'on doit opérer R en tapant des instructions (ou en copiant des sections de code) plutôt qu'en utilisant des menus et en cliquant sur différentes options. Si on ne sait pas quelle commande taper, rien ne se passe. Ce n'est donc pas facile d'utilisation à priori. Cependant, il est possible d'apprendre rapidement à faire certaines des opérations de base (ouvrir un fichier de données, faire un graphique pour examiner ces données, effectuer un test statistique simple). Et une fois que l'on comprend le principe de la chose, on peut assez facilement trouver sur le web des exemples d'analyses ou de graphiques plus complexes et adapter le code à nos propres besoins.

C'est ce que vous ferez dans le premier laboratoire pour vous familiariser avec R.

Pourquoi cet inconvénient est-il d'une certaine façon un avantage? Parce que vous allez sauver du temps en fin de compte. Garanti. Croyez-moi, on ne fait jamais une analyse une seule fois. En cours de route, on découvre des erreurs d'entrée de données, ou que l'on doit faire l'analyse séparément pour des sous-groupes, ou on obtient des données supplémentaires, ou on fait une erreur. On doit alors recommencer l'analyse. Avec une interface graphique et des menus, cela implique recommencer à cliquer ici, entre des paramètres dans des boîtes et sélectionner des boutons. Chaque fois avec possibilité d'erreur. Avec une série de commandes écrites, il suffit de corriger ce qui doit l'être puis de copier-coller l'ensemble pour répéter instantanément. Et vous avez la possibilité de parfaitement documenter ce que vous avez fait. C'est comme cela que les professionnels travaillent et offrent une assurance de qualité de leurs résultats.

Installation

Pour installer R sur un nouvel ordinateur, allez au site <http://cran.r-project.org/>. Vous y trouverez des versions compilées (binaries) pour votre système d'exploitation de prédilection (Windows, MacOS, Linux).

La version que j'ai téléchargée au moment d'écrire ces notes est 3.0.1 pour Windows (36 Mbytes). Je l'ai installée sans changer les options d'installation ou de démarrage par défaut.

Note : R a déjà été installé l'été dernier sur les ordinateurs du laboratoire (la version pourrait être un peu plus ancienne, mais cela devrait être sans conséquences).

Instructions générales pour les laboratoires

- Apporter une clé USB ou son équivalent à chaque séance de laboratoire pour sauvegarder votre travail.
- Lire l'exercice de laboratoire AVANT la séance, lire le code R correspondant et préparer vos questions sur le code.
- Durant les pré-labs, écouter les instructions et posez vos questions au moment approprié.
- Faites les exercices du manuel de laboratoire à votre rythme, en équipe, puis je vous recommande de commencer (compléter?) le devoir. Profitez de la présence du démonstrateur et du prof...
- Pendant vos analyses, copiez-collez des fragments de sorties de R dans un document (par exemple dans votre traitement de texte favori). Annotez abondamment.
- À chaque fois que vous fermez R, sauvegardez l'historique de vos commandes (ex: labo1.1 rHistory, labo1.2.rHistory, etc). Vous pourrez ainsi refaire le labo instantanément, récupérer des fragments de code, ou plus facilement identifier les erreurs dans vos analyses.
- Créez votre propre librairie de fragments de codes (snippets). Annotez-là abondamment. Vous vous en fêlerez plus tard.

Et pour les dilettantes ou ceux qui sont allergiques à la programmation...

Je vous conseille d'essayer d'appivoiser R sans interface graphique. Au début ce sera peut-être difficile, mais cela vous forcera à acquérir de bonnes habitudes. Si vous trouvez cela vraiment trop lourd, ou que vous vous heurtez à des problèmes récurrents qui vous font perdre trop de temps, alors vous pouvez installer (du moins sur votre propre ordi, je ne sais pas si vous pouvez au laboratoire) une interface graphique distribuée par CRAN et qui fonctionne sous Windows, Mac et Linux (à ce qu'il paraît, je n'ai essayé que la version Windows).

Il s'agit de Rcmdr, que vous devez d'abord installer dans R. Vous devez avoir accès à l'Internet. Dans la fenêtre de commande, entrez la commande

```
install.packages("Rcmdr", dependencies=TRUE)
```

Une fenêtre s'ouvrira vous permettant de choisir l'un des sites distribuant Rcmdr. Le téléchargement et l'installation prendront plusieurs minutes car plusieurs autres packages doivent être téléchargés et installés. Quand ce sera terminé, vous pourrez démarrer l'interface graphique en entrant la commande:

```
library(Rcmdr)
```

Une nouvelle fenêtre s'ouvrira, avec des menus pour accéder à la majorité des commandes usuelles. Une des sous-fenêtres de Rcmdr est un historique des commandes R correspondant à vos choix dans les menus. Étudiez-les pour apprendre, peut-être plus facilement, le langage R.

Labo- Introduction à R

Après avoir complété cet exercice de laboratoire, vous pourrez :

- Ouvrir des fichiers de données R déjà existants
- Importer des ensembles de données rectangulaires
- Exporter des données de R vers un fichier texte
- Vérifier si les données ont été correctement importées
- Examiner la distribution des observations d'une variable
- Examiner visuellement et tester la normalité d'une variable
- Calculer des statistiques descriptives d'une variable
- Effectuer des transformations de données

Importer et exporter des données

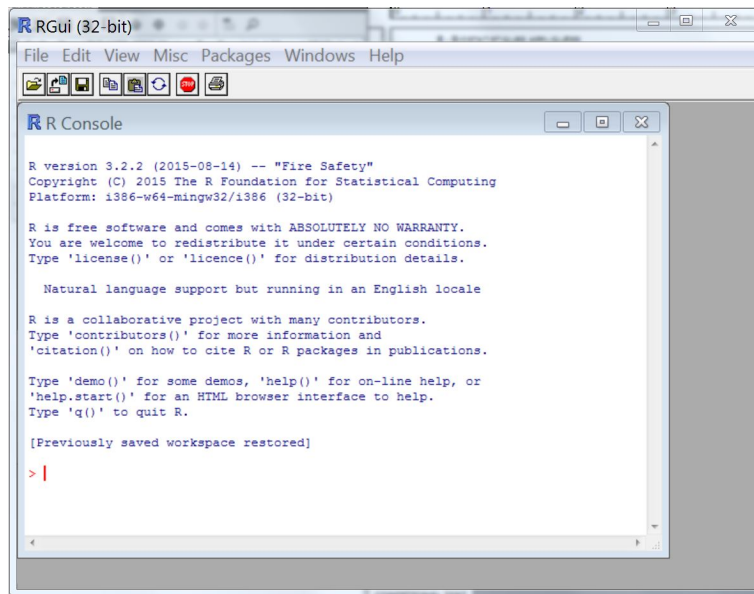
Ouvrir et sauvegarder un fichier de données en format R

Les données pour les exercices de laboratoire et pour les devoirs vous sont fournies déjà en format R (fichiers avec extension .Rdata). Pour ouvrir ces fichiers, vous pouvez cliquer dessus et laisser votre système d'exploitation démarrer une nouvelle session de R ou encore, à partir de la console de R, taper sur une ligne de commande :

```
load(file.choose())
```

ce qui ouvrira une boîte de dialogue vous permettant d'aller choisir un fichier sur votre ordinateur. Après avoir fait votre sélection (choisir le fichier ErablesGatineau.Rdata), vous reviendrez à la fenêtre Rconsole sans changement apparent :

Figure 1.



Pour vérifier si les données ont bel et bien été lues, vous pouvez lister les objets en mémoire avec la fonction `ls()` ou en obtenir une liste avec une description plus détaillée avec `ls.str()`

```
> ls()
[1] "ErablesGatineau"
> ls.str()
ErablesGatineau : 'data.frame': 100 obs. of  3 variables:
 $ station: Factor w/ 2 levels "A      ",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ diam   : num  22.4 36.1 44.4 24.6 17.7 ...
 $ biom   : num  732 1171 673 1552 504 ...
```

R confirme avoir en mémoire l'objet `ErablesGatineau`.

`ErablesGatineau` est un tableau de données rectangulaire (`data.frame`) contenant 100 observations (lignes) de 3 variables (colonnes) : `station`, une variable de type Facteur avec 2 niveaux, et `diam` et `biom` qui sont 2 variables numériques.

Entrer des données

R n'est pas un environnement idéal pour entrer des données. C'est possible, mais la syntaxe est lourde et m'incite à m'arracher les cheveux. Utilisez votre chiffrier préféré pour faire l'entrée de données. Ce sera plus efficace et moins frustrant.

Nettoyer/corriger des données

Une autre opération qui peut être frustrante en R. Mon conseil : ne le faites pas là. Retournez au fichier original, faites la correction là, puis re-exportez les données vers R. Il est finalement plus simple de refaire exécuter les quelques lignes de code par la machine. Vous aurez à la fin une seule version (corrigée) de vos données et un code qui vous permet de refaire votre analyse.

Importer des données à partir d'Excel.

Sauvegarder la matrice rectangulaire de données en dans un fichier en format csv. (**File>Save as**).

Importer ces données en R avec la commande `read.csv`.

Par exemple, pour créer un `data.frame` appelé `age` à partir d'un fichier csv préalablement sauvegardé, on peut taper la commande :

```
>
age<-read.csv(file.choose())
```

(Si vous essayez avec le fichier `age.csv` exporté du fichier excel `age.xls`, vous obtiendrez:

```
> age
  ageclass    sex count
1      0-9 female 17619
2      0-9  male 17538
3     10-19 female 17947
4     10-19  male 18207
5     20-29 female 21344
6     20-29  male 21401
7     30-39 female 19138
8     30-39  male 18837
9     40-49 female 13135
10    40-49  male 12568
11    50-59 female 11617
12    50-59  male 10661
13    60-69 female 11053
14    60-69  male  9374
15    70-79 female  7712
16    70-79  male  5348
17     80+ female  4114
18     80+  male  1926
```

Attrape : Attention si vous travaillez dans une langue utilisant la virgule au lieu du point décimal. Par défaut, R utilise le point décimal et vous n'obtiendrez pas le résultat escompté. Il existe une version modifiée de `read.csv()` appelée `read.csv2()` qui règle ce problème. Googlez-la si vous en avez besoin.

Exporter des données à partir de R.

```
write.csv(mydata, file = "outfilename.csv", row.names =
FALSE)
```


où `mydata` est le nom du `data.frame` à exporter et `outfilename.csv` est le nom du fichier à produire. Notez que ce fichier sera créé dans le répertoire de travail (qui peut être changé par le menu à **File>Change dir**, ou par la commande `setwd()`)

Examen préliminaire des données


La première étape de toute analyse est l'examen des données. Elle nous permet de découvrir si on a bien importé les données, si les nombres enregistrés sont possibles, si toutes les données ont bien été

lues, etc. L'examen préliminaire des données permet souvent aussi d'identifier des observations suspectes, possiblement dûes à des erreurs d'entrée de donnée. Finalement, l'examen graphique préliminaire permet en général de visualiser les tendances principales qui seront confirmées par l'analyse statistique en tant que telle.

Le fichier `sturgeon.Rdata` contient les données d'une étude effectuée sur les esturgeons de la rivière Saskatchewan. Ces données ont été récoltées, entre autres, pour examiner comment la taille des esturgeons varie entre les sexes (`sex`), les sites (`location`), et les années (`year`).


 Pour recommencer avec une ardoise vide, videz la mémoire de R de tout son contenu en tapant la commande `rm(list=ls())`

 Ouvrez le fichier `sturgeon.Rdata`.

 Pour obtenir un aperçu des éléments du fichier qui ont été chargés en mémoire, taper la commande `ls.str()` :

```
sturgeon : 'data.frame':      186 obs. of  9 variables:
 $ fklngth : num  37 50.2 28.9 50.2 45.6 ...
 $ totlngth: num  40.7 54.1 31.3 53.1 49.5 ...
 $ drlngth : num  23.6 31.5 17.3 32.3 32.1 ...
 $ rdwght  : num  15.95 NA 6.49 NA 29.92 ...
 $ age     : num  11 24 7 23 20 23 20 7 23 19 ...
 $ girth   : num  40.5 53.5 31 52.5 50 54.2 48 28.5 44 39 ...
 $ sex     : Factor w/ 2 levels "FEMALE","MALE": 2 1 2 1 2 1 1 2 2 2 ...
 $ location: Factor w/ 2 levels "CUMBERLAND","THE_PAS": 2 2 2 2 2 2 2 2 2 2
 ...
 $ year    : Factor w/ 3 levels "1978","1979",...: 1 1 1 1 1 1 1 1 1 1 ...
```

Sommaire statistique

 Pour un sommaire du contenu du `data.frame` appelé `sturgeon` qui est en mémoire, taper la commande `summary(sturgeon)`

```
      fklngth      totlngth      drlngth
Min.   :24.96   Min.   :28.15   Min.   :14.33
1st Qu.:41.00   1st Qu.:43.66   1st Qu.:25.00
Median :44.06   Median :47.32   Median :27.00
Mean   :44.15   Mean   :47.45   Mean   :27.29
3rd Qu.:48.00   3rd Qu.:51.97   3rd Qu.:29.72
Max.   :66.85   Max.   :72.05   Max.   :41.93
NA's   :        NA's   :85.00   NA's   :13.00

      rdwght      age      girth
Min.   : 4.73   Min.   : 7.00   Min.   :11.50
1st Qu.:18.09   1st Qu.:17.00   1st Qu.:40.00
Median :23.10   Median :20.00   Median :44.00
Mean   :24.87   Mean   :20.24   Mean   :44.33
3rd Qu.:30.27   3rd Qu.:23.50   3rd Qu.:48.80
Max.   :93.72   Max.   :55.00   Max.   :73.70
NA's   : 4.00   NA's   :11.00   NA's   :85.00

      sex      location      year
FEMALE:106   CUMBERLAND: 85   1978:45
MALE   : 80   THE_PAS   :101   1979:68
                                1980:73
```

Pour chaque variable, R donne le minimum, le maximum, la médiane qui est la valeur au milieu de la liste des observations ordonnées (appelée le 50^{ième} percentile), ici, la 93^{ième} valeur des 186 observations, les valeurs au premier (25%) et troisième quartile (75%), et si il y a des valeurs manquantes dans la colonne.

Notez que plusieurs des variables ont des observations manquantes (NA). Donc, seules les variables `fklnth` (longueur à la fourche), `sex`, `location` et `year` ont 186 observations.

Attrape : Attention aux valeurs manquantes. Plusieurs fonctions de R y réagissent mal et on doit souvent faire les analyses sur des sous-ensembles sans valeur manquante, par des commandes ou des options dans les commandes. On y reviendra, mais prenez l'habitude de noter mentalement si il y a des données manquantes et de vous en rappeler en faisant l'analyse

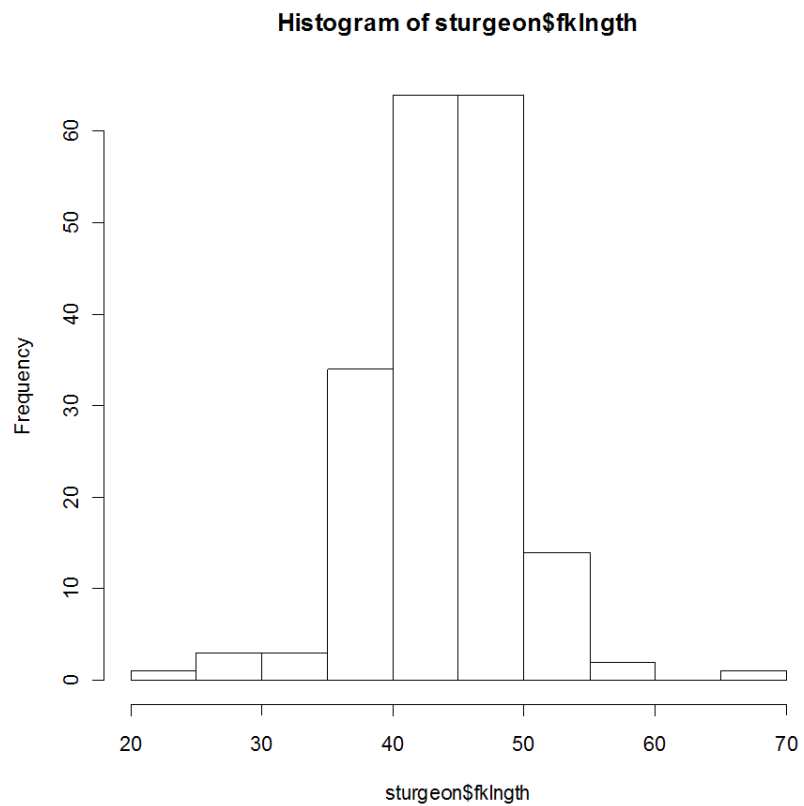
Histogramme, densité de probabilité empirique, boxplot et examen visuel de la normalité

Examinons maintenant de plus près la distribution de `fklnth`.

La commande `hist()` permet de tracer un histogramme de la variable `fklnth` dans le `data.frame` `sturgeon`

```
hist(sturgeon$fklnth)
```

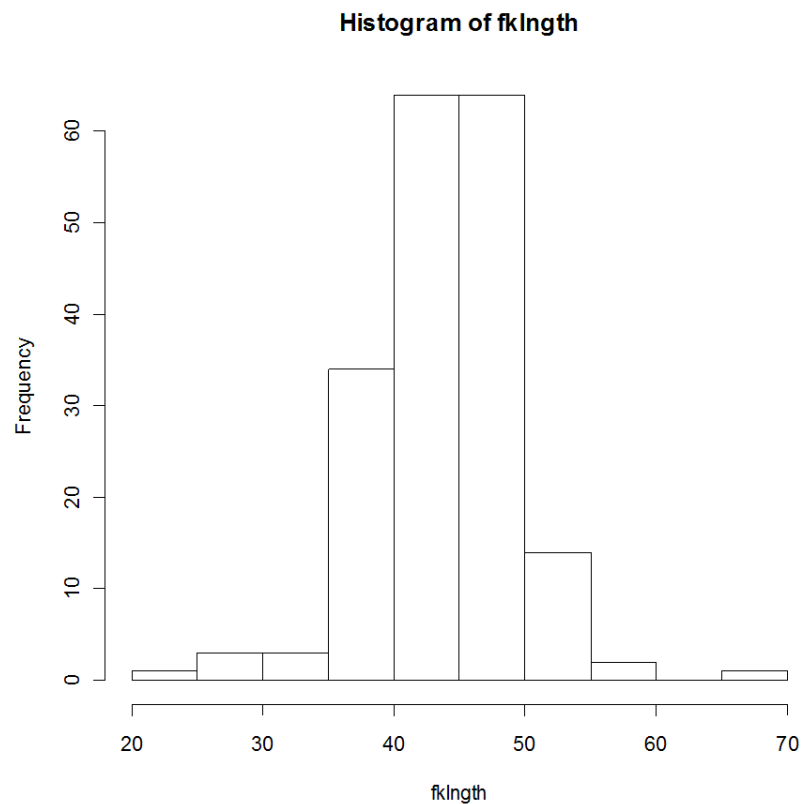
Figure 2.




Les données semblent suivre approximativement une distribution normale. C'est bon à savoir. Cette syntaxe est un peu lourde puisqu'on doit ajouter le préfixe `sturgeon$` devant chaque nom de variable. On peut se faciliter la tâche en utilisant la commande `attach(sturgeon)` qui va nous donner accès directement aux variables contenues dans le data.frame `sturgeon`.

```
attach(sturgeon)  
hist(flenght)
```

Figure 3.



Cet histogramme (Fig. 3) est la représentation classique. Mais les histogrammes ne sont pas parfaits. Leur forme dépend en partie du nombre de catégories utilisées, surtout pour les petits échantillons. On peut faire mieux, particulièrement si on est intéressé à comparer visuellement la distribution des observations à une distribution normale. Mais il faut programmer un peu (ou savoir copier-coller...)

 Copiez-collez le code suivant dans une nouvelle fenêtre script (**File->New script**, ou **Ctrl-n** dans Windows), puis exécutez le :

```
require(ggplot2)
# use "sturgeon" dataframe to make plot called mygraph
# and define x axis as representing fklength
mygraph <- ggplot(sturgeon, aes(x = fklength))
# add data to the mygraph ggplot
mygraph <- mygraph +
  # add data density smooth
  geom_density() +
  # add rug (bars at the bottom of the plot)
  geom_rug() +
  # add black semitransparent histogram
  geom_histogram(aes(y = ..density..),
                 color = "black",
```

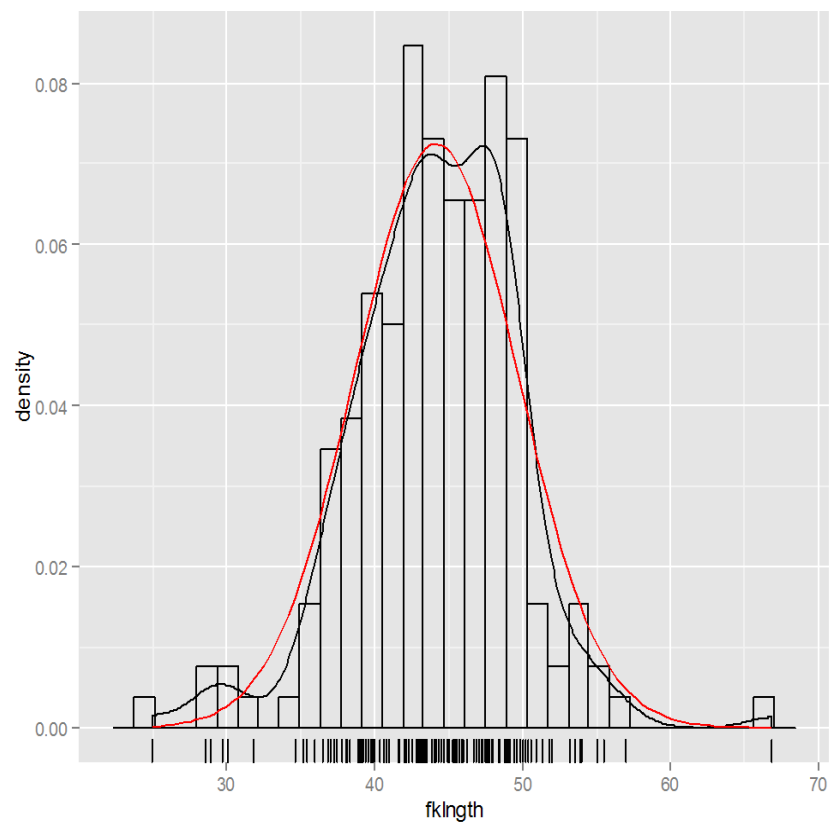
```

                                alpha = 0.3) +
# add normal curve in red, with mean and sd from fklngth
stat_function(fun = dnorm,
              args = list(
                mean = mean(sturgeon$fklngth),
                sd = sd(sturgeon$fklngth)
              ),
              color = "red")
#display graph
mygraph

```

Vous obtiendrez :

Figure 4.

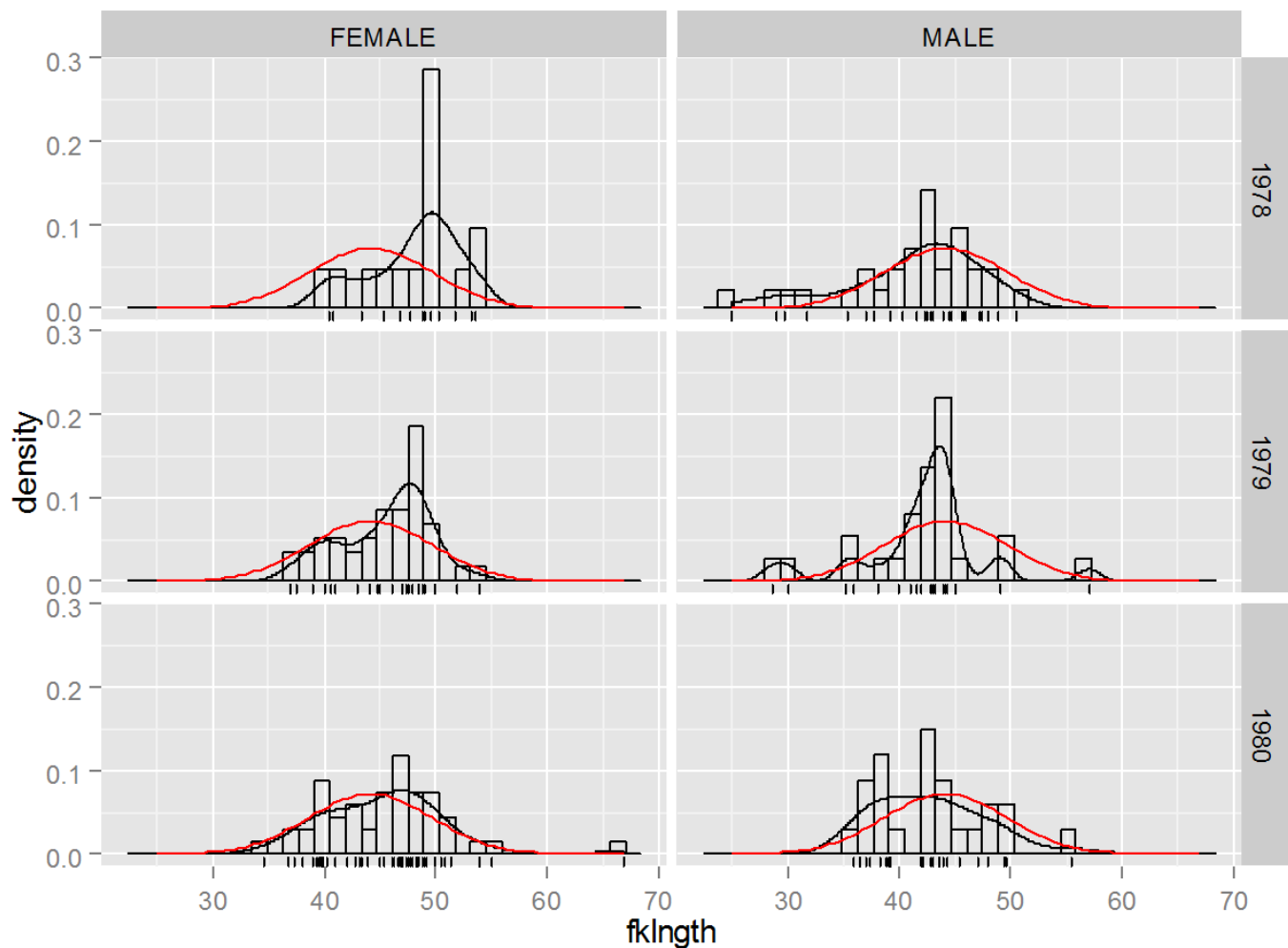


Chaque observation est représentée par une barre sous l'axe des x (rug). En rouge est la distribution normale de données avec la même moyenne et écart-type que les observations. Et l'autre ligne est la densité de probabilité empirique, « lissée » à partir des observations.

Si vous êtes plus aventureux, vous pouvez examiner la distribution des observations de fklngth par sous-groupes (par exemple sex et year) avec :

```
mygraph + facet_grid(year ~ sex)
```

Figure 5.

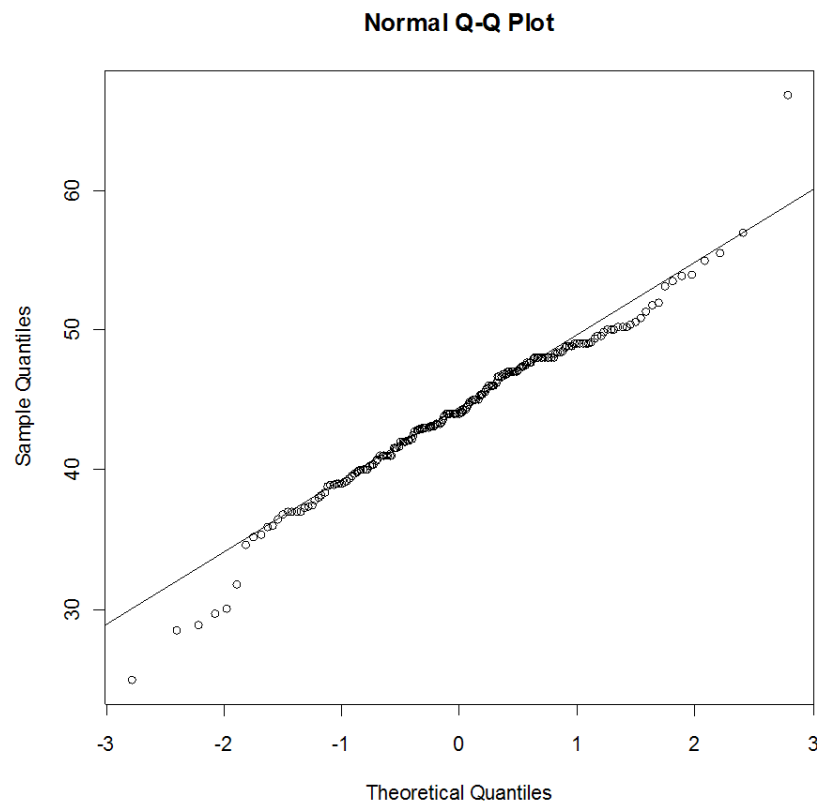


Chaque panneau illustre la distribution pour un sexe cette année-là, et la courbe en rouge récurrente représente la distribution normale pour l'ensemble des données. Cette courbe peut servir à mieux évaluer visuellement les différences entre les panneaux.

Une autre façon d'évaluer la normalité de données visuellement est de faire un QQ plot avec la paire de commandes `qqnorm()` et `qqline()`,

```
qqnorm(fklngth)
qqline(fklngth)
```


Figure 6.



Des données parfaitement normales suivraient la ligne droite diagonale. Ici, il y a des déviations dans les queues de la distribution, et un peu à droite du centre. Comparez cette représentation à celle des deux graphiques précédents. Vous conviendrez sans doute avec moi qu'il est plus facile de visualiser comment la distribution dévie de la normalité sur les histogrammes et les graphiques de la densité empirique de probabilité que sur les QQ plots. Ceci dit, les QQ plots sont souvent utilisés et vous devriez être capable de les interpréter. De plus, on peut facilement éprouver statistiquement l'hypothèse que les données sont distribuées normalement avec R par la commande `shapiro.test()` qui calcule une statistique (W) qui est une mesure de la tendance des points d'un QQ plot à former une ligne parfaite. Si oui, alors $W=1$. Si W s'éloigne de 1 (vers 0), alors les données s'éloignent de la normalité.

Ici :

```
shapiro.test(fklength)
```

```
Shapiro-Wilk normality test
```

```
data: fklength
```

```
w = 0.9722, p-value = 0.0009284
```

W n'est pas très loin de 1, mais suffisamment pour que la différence soit significative.

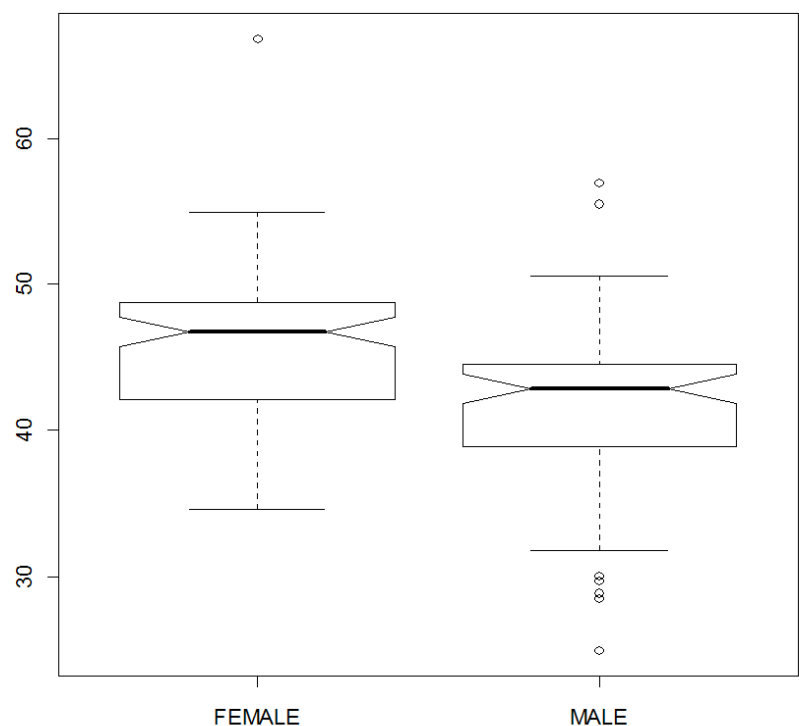
L'examen visuel des grands échantillons est souvent compliqué par le fait que plusieurs points se superposent et qu'il devient plus difficile de bien visualiser la tendance centrale. Les boxplots avec "moustaches" (box and whiskers plots) offrent une alternative intéressante.

La commande

```
boxplot(fklength~sex, notch=TRUE)
```

produit un boxplot de fklength pour chaque niveau de sex, et ajoute les coches.

Figure 7.



La ligne un peu plus épaisse dans la boîte de la Fig. 7 indique la médiane. La coche est proportionnelle à l'incertitude quant à la position de la médiane. On peut visuellement interpréter approximativement les différences entre médianes en examinant si il y a chevauchement entre les coches (ici, il n'y a pas chevauchement, et on conclurait provisoirement que la médiane de fklength pour les

femelles est supérieure à celle des mâles). Les boîtes s'étendent du premier au troisième quartile (du 25^{ième} au 75^{ième} percentile si vous préférez), Les barres (moustaches ou whiskers) au-dessus et en dessous des boîtes s'étendent soit de la valeur minimum à la valeur maximum, ou, si il y a des valeurs extrêmes, de la plus petite à la plus grande valeur à l'intérieur de 1.5x la largeur de l'étendue interquartile . Enfin, les observations qui excèdent les limites des moustaches (donc à plus de 1.5x l'étendue interquartile de chaque côté de la médiane) sont indiquées par des symboles. Ce sont des valeurs qui pourraient être considérées comme extrêmes et possiblement aberrantes.

Diagrammes de dispersion bivariés

En plus des graphiques pour chacune des variables séparément, il est très souvent intéressant de jeter un coup d'oeil aux diagrammes de dispersion . La commande `plot(y~x)` permet de faire le graphique de y sur l'axe vertical (l'ordonnée) en fonction de x sur l'axe horizontal (l'abscisse).


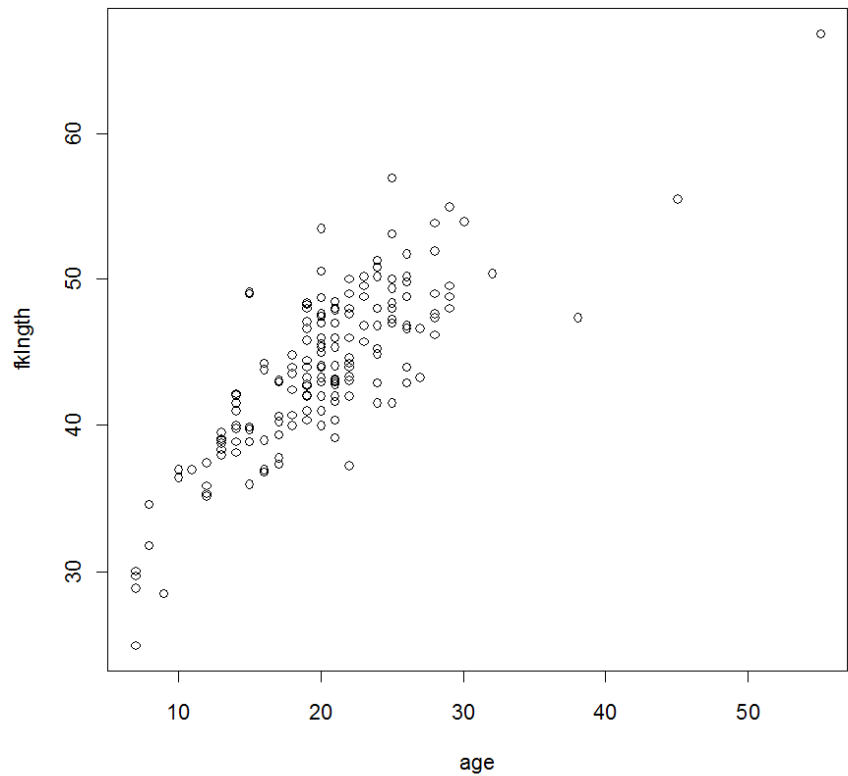
 Faites un graphique de `fklngh` en fonction de `age` avec la commande `plot`. Vous devriez obtenir:

Figure 8.



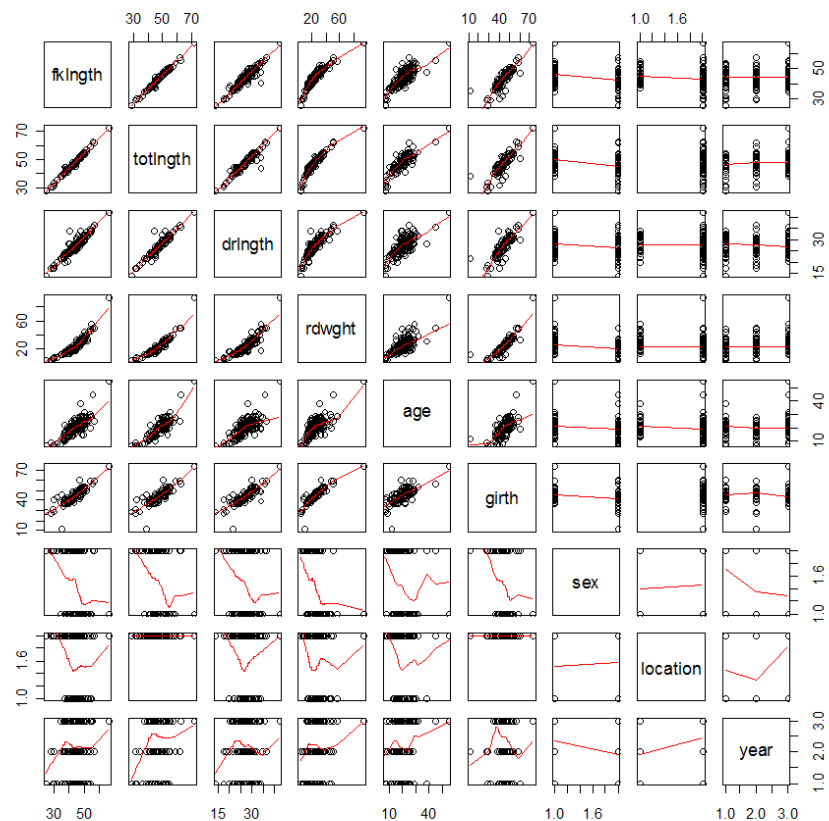
R a une fonction qui permet la création des graphiques de dispersion de toutes les paires de variables (`pairs()`). Une des option de `pairs()` est l'ajout d'une trace lowess qui indique la tendance de la relation entre les variables.

Pour obtenir la matrice de ces graphiques avec la trace lowess pour toutes les variable dans `sturgeon`, entrer la commande

```
pairs(sturgeon, panel=panel.smooth)
```

et vous devriez obtenir

Figure 9.



Créer des sous-ensembles de cas

Il arrive fréquemment qu'une analyse se concentre sur un sous-ensemble des observations contenues dans un fichier de données. Les cas sont d'habitude sélectionnés selon un critère en particulier. Pour utiliser un sous-ensemble de vos données en créant un graphique ou en performant une analyse, on peut utiliser la commande `subset()`. Par exemple, pour créer un sous ensemble des données du tableau `sturgeon` qui ne contient que les femelles capturées en 1978, on peut écrire

```
:
> sturgeon.female.1978<-subset(sturgeon,sex=="FEMALE" &
year=="1978")
> sturgeon.female.1978
```

	flngth	totlngth	drlngth	rdwght	age	girth
2	50.19685	54.13386	31.49606	NA	24	53.5
4	50.19685	53.14961	32.28346	NA	23	52.5
6	49.60630	53.93701	31.10236	35.86	23	54.2
7	47.71654	51.37795	33.97638	33.88	20	48.0
15	48.89764	53.93701	29.92126	35.86	23	52.5
105	46.85039	NA	28.34646	23.90	24	NA
106	40.74803	NA	24.80315	17.50	18	NA
107	40.35433	NA	25.59055	20.90	21	NA

```

109 43.30709      NA 27.95276 24.10 19  NA
113 53.54331      NA 33.85827 48.90 20  NA
114 51.77165      NA 31.49606 35.30 26  NA
116 45.27559      NA 26.57480 23.70 24  NA
118 53.14961      NA 32.67717 45.30 25  NA
119 50.19685      NA 32.08661 33.90 26  NA
123 49.01575      NA 29.13386 37.50 22  NA
      sex      location year
2   FEMALE    THE_PAS 1978
4   FEMALE    THE_PAS 1978
6   FEMALE    THE_PAS 1978
7   FEMALE    THE_PAS 1978
15  FEMALE    THE_PAS 1978
105 FEMALE CUMBERLAND 1978
106 FEMALE CUMBERLAND 1978
107 FEMALE CUMBERLAND 1978
109 FEMALE CUMBERLAND 1978
113 FEMALE CUMBERLAND 1978
114 FEMALE CUMBERLAND 1978
116 FEMALE CUMBERLAND 1978
118 FEMALE CUMBERLAND 1978
119 FEMALE CUMBERLAND 1978
123 FEMALE CUMBERLAND 1978

```

Attrape: Dans ces comparaisons, il faut toujours utiliser `==` pour égal à. Dans ce contexte, si vous utilisez `=` seulement, vous n'obtiendrez pas ce que vous désirez.

Dans le tableau qui suit se trouve une liste de commandes communes que vous allez probablement utiliser pour créer des expressions en R.

Operator	Explication	Operator	Explication
<code>==</code>	Égal à	<code>!=</code>	Pas égal à
<code>></code>	Plus que	<code><</code>	Moins que
<code>>=</code>	Plus que ou égal à	<code><=</code>	Moins que ou égal à
<code>&</code>	Et vectorisé	<code> </code>	Ou vectorisé
<code>&&</code>	Et contrôle	<code> </code>	Ou contrôle
<code>!</code>	Pas		


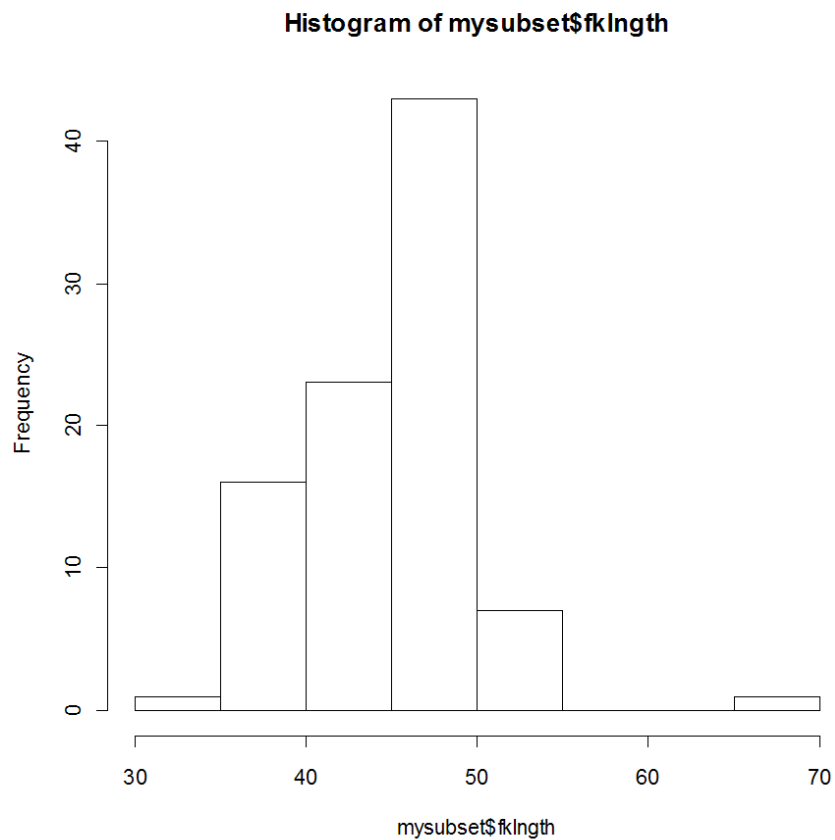
 En utilisant les commandes `subset()` et `hist()`, essayez de faire un histogramme pour le sous-ensemble de cas correspondant aux femelles capturées en 1979 et 1980 (donc `sex=="FEMALE" & (year=="1979" | year=="1980")`)

Figure 10.



Transformations de données

Il est très fréquemment nécessaire d'effectuer des transformations mathématiques sur les données brutes pour mieux satisfaire aux conditions d'application de tests statistiques. R étant aussi un langage de programmation complet, il peut donc effectuer les transformations désirées.

Les fonctions les plus fréquemment utilisées sont:

`log10()`

`sqrt()`

`ifelse()`

On peut employer ces fonctions directement dans les lignes de commandes, ou encore créer de nouvelles variables orphelines ou faisant partie d'un `data.frame`.

Par exemple, pour faire un graphique du logarithme décimal de `fklnth` en fonction de l'âge, on peut écrire

```
plot(log10(fklnth)~age)
```

Pour créer une variable orpheline (i.e. non incluse dans le `data.frame`) appelée `lfklnth` et contenant le logarithme décimal de `fklnth`, on peut écrire

```
lfklnth<-log10(fklnth)
```

Si on veut ajouter cette variable transformée à un tableau de données (`data.frame`), alors, on doit préfixer le nom de la variable par le nom du `data.frame` et du symbole `$`, par exemple, pour ajouter une variable nommée `lfkl` contenant le `log10` de `fklnth` au tableau `sturgeon`, on peut écrire:

```
sturgeon$lfkl<-log10(fklnth)
```

N'oubliez pas de sauvegarder ce tableau modifié si vous voulez avoir accès à cette nouvelle variable dans le futur.

Pour les transformations conditionnelles, on peut utiliser la fonction `ifelse()`. Par exemple, pour créer une nouvelle variable appelée `dummy` qui sera égale à 1 pour les mâles et 0 pour les femelles, on peut écrire:

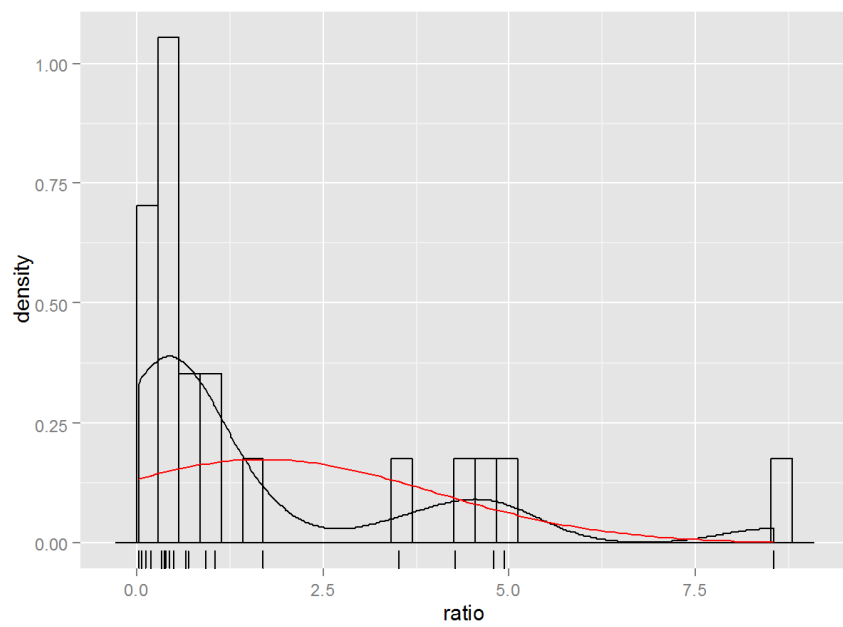
```
dummy<-ifelse(sex=="MALE",1,0)
```

Exercice

Vous trouverez dans le fichier `salmonella`, des valeurs numériques du ratio pour deux milieux (IN VITRO et IN VIVO) pour trois souches.

Examinez les données pour ratio et faites des graphiques pour évaluer la normalité de la distribution des ratios pour la souche SAUVAGE.

Figure 11.



Labo - Analyse de puissance avec G*Power

Après avoir complété cet exercice de laboratoire, vous devriez :

- Pouvoir calculer la puissance d'un test de t avec G*Power
- Pouvoir calculer l'effectif requis pour obtenir la puissance désirée avec un test de t
- Pouvoir calculer la taille de l'effet détectable par un test de t étant donné l'effectif, la puissance et α
- Comprendre comment la puissance change lorsque l'effectif augmente, la taille de l'effet change, ou lorsque α diminue
- Comprendre comment la puissance est affectée lorsque l'on passe d'un test bilatéral à un test unilatéral

La théorie

Qu'est-ce que la puissance?

La puissance est la probabilité de rejeter l'hypothèse nulle quand elle est fausse.

Pourquoi faire une analyse de puissance?

Évaluer l'évidence. L'analyse de puissance effectuée après avoir accepté une hypothèse nulle permet de calculer la probabilité que l'hypothèse nulle soit rejetée si elle était fausse et que la taille de l'effet était d'une valeur donnée. Ce type d'analyse *a posteriori* est très commun.

Planifier de meilleures expériences. L'analyse de puissance effectuée avant de réaliser une expérience (le plus souvent *après* une expérience préliminaire cependant), permet de déterminer le nombre d'observations nécessaires pour détecter un effet d'une taille donnée à un niveau fixe de probabilité (la puissance). Ce type d'analyse *a priori* devrait être réalisé plus souvent.

Estimer la "limite de détection" statistique. L'effort d'échantillonnage est souvent déterminé à l'avance (par exemple lorsque vous héritez de données récoltées par quelqu'un d'autre), ou très sévèrement limité (lorsque les contraintes logistiques prévalent). Que ce soit *a priori* ou *a posteriori* l'analyse de puissance vous permet d'estimer, pour un effort d'échantillonnage donné et un niveau de puissance fixe, quelle est la taille minimale de l'effet qui peut être détecté (comme étant statistiquement significatif).

Facteurs qui affectent la puissance

Il y a 3 facteurs qui affectent la puissance d'un test statistique:

Le critère de décision. La puissance dépend de α , le seuil de probabilité auquel on rejette l'hypothèse nulle. Si ce seuil est très strict (i.e. si α est fixé à une valeur très basse, comme 0.1% ou $p=0.001$), alors la puissance sera plus faible que si le seuil était moins strict.


La taille de l'échantillon. Plus l'échantillon est grand, plus la puissance est élevée. La capacité d'un test à détecter de petites différences comme étant statistiquement significatives augmente avec une augmentation du nombre d'observations.

La taille de l'effet. Plus la taille de l'effet est grande, plus un test a de puissance. Pour un échantillon de taille fixe, la capacité d'un test à détecter un effet comme étant statistiquement significatif est plus élevée si l'effet est grand que s'il est petit. La taille de l'effet est en fait une mesure du degré de fausseté de l'hypothèse nulle.

Qu'est ce que G*Power?

G*Power est un programme gratuit, développé par des psychologues de l'Université de Dusseldorf en Allemagne et disponible à <http://www.gpower.hhu.de/en.html>. Le programme existe en version Mac et Windows.

G*Power vous permettra d'effectuer une analyse de puissance pour la majorité des tests que nous verrons au cours de la session sans avoir à effectuer des calculs complexes ou farfouiller dans des tableaux ou des figures décrivant des distributions ou des courbes de puissance. C'est vraiment un outil très utile que vous devrez maîtriser.

 Téléchargez le programme et installez-le sur votre ordi et votre station de travail au laboratoire (si ce n'est déjà fait).

Comment utiliser G*Power

Principe général

L'utilisation de G*Power implique généralement en trois étapes:

1. Choisir le test approprié
2. Choisir l'un des 5 types d'analyses de puissance disponibles
3. Inscrire les valeurs des paramètres requis et cliquer sur **Calculate**

Types d'analyses de puissance disponibles

A priori. Calcule l'effectif requis pour une valeur de α , β , et de taille d'effet donnée. Ce type d'analyse est utile à l'étape de planification des expériences.

Compromis. Calcule α et β pour un rapport β/α donné, un effectif fixe, et une taille d'effet donnée. Ce type d'analyse est plus rarement utilisé (je ne l'ai jamais fait), mais peut être utile lorsque le rapport β/α est d'intérêt, par exemple lorsque le coût d'une erreur de type I et de type II peut être quantifié.

Critère. Calcule α pour β , effectif et taille d'effet donné. En pratique, je vois peu d'utilité pour ce type de calcul. Contactez-moi si vous en voyez une!

Post-hoc. Calcule la puissance pour α , une taille d'effet et un effectif donné. Très utilisée pour interpréter les résultats d'une analyse statistique non-significative, mais seulement si l'on utilise une taille d'effet biologiquement significative (et non la taille d'effet observée). Peu pertinente lorsque le test est significatif.

Sensitivité. Calcule la taille d'effet détectable pour une valeur d' α , β , et un effectif donné. Très utile également au stade de planification des expériences.

Comment calculer la taille de l'effet

G*Power permet de faire une analyse de puissance pour de nombreux tests statistiques. L'indice de la taille de l'effet qui est utilisé par G*Power pour les calculs dépend du test. Notez que d'autres logiciels peuvent utiliser des indices différents et il est important de vérifier que l'indice que l'on utilise est celui qui convient. G*Power vous facilite la tâche et permet de calculer la taille de l'effet en inscrivant seulement les valeurs pertinentes dans la fenêtre de calcul. Le tableau suivant donne les indices utilisés par G*Power pour les différents tests.

Test	Indice utilisé pour quantifier la taille de l'effet	Formule
Tests de t sur des moyennes	d	$d = \frac{ \mu_1 - \mu_2 }{\sqrt{(s_1^2 + s_2^2)/2}}$

Test de t pour des corrélations r

Autres tests de t f

$$f = \mu_1 / \sigma$$

Tests F (ANOVA) f

$$f = \frac{\sqrt{\frac{\sum_{i=1}^k (\mu_i - \mu)^2}{k}}}{\sigma}$$

Autres tests F f^2

$$f^2 = \frac{R_p^2}{1 - R_p^2}$$

où R_p^2 est le carré du coefficient de corrélation partiel

Test Chi-carré w

$$w = \sqrt{\sum_{i=1}^m \frac{(p_{0i} - p_{1i})^2}{p_{0i}}}$$

où p_{0i} est la proportion de la catégorie i prédite par l'hypothèse nulle (0) et l'hypothèse alternative (1)

Calculs de puissance pour un test de t comparant deux moyennes indépendantes

L'objectif de cette séance de laboratoire est de vous familiariser avec G*Power et de vous aider à comprendre comment les quatre paramètres des analyses de puissance (α , β , effectif et taille de l'effet) sont reliés entre eux. On examinera seulement un des nombreux tests, le test de t permettant de comparer deux moyennes indépendantes. C'est le test le plus communément utilisé par les biologistes, vous l'avez tous déjà utilisé, et il conviendra très bien pour les besoins de la cause. Ce que vous apprendrez aujourd'hui s'appliquera à toutes les autres analyses de puissance que vous effectuerez à l'avenir.

Jaynie Stephenson a étudié la productivité des ruisseaux de la région d'Ottawa. Elle a, entre autres, quantifié la biomasse des poissons dans 18 ruisseaux sur le Bouclier Canadien d'une part, et dans 18 autres ruisseaux de la vallée de la rivière des Outaouais et de la rivière Rideau d'autre part. Elle a observé une biomasse plus faible dans les ruisseaux de la vallée (2.64 g/m^2 , écart-type=3.28) que dans ceux du Bouclier (3.31 g/m^2 , écart-type=2.79.). En faisant un test de t pour éprouver l'hypothèse nulle que la biomasse des poissons est la même dans les deux régions, elle obtient:

Pooled-Variance Two-Sample t-Test

$t = -0.5746$, $df = 34$, $p\text{-value} = 0.5693$

Elle accepte l'hypothèse nulle (puisque p est plus élevé que 0.05) conclue donc que la biomasse moyenne des poissons est la même dans ces deux régions.

Analyse post-hoc

Compte tenu des valeurs des moyennes observées et de leur écart-type, on peut utiliser G*Power pour calculer la puissance du test de t bilatéral pour deux moyennes indépendantes et pour la taille d'effet (i.e. la différence entre la biomasse entre les deux régions, pondérée par les écarts-type) à $\alpha=0.05$.

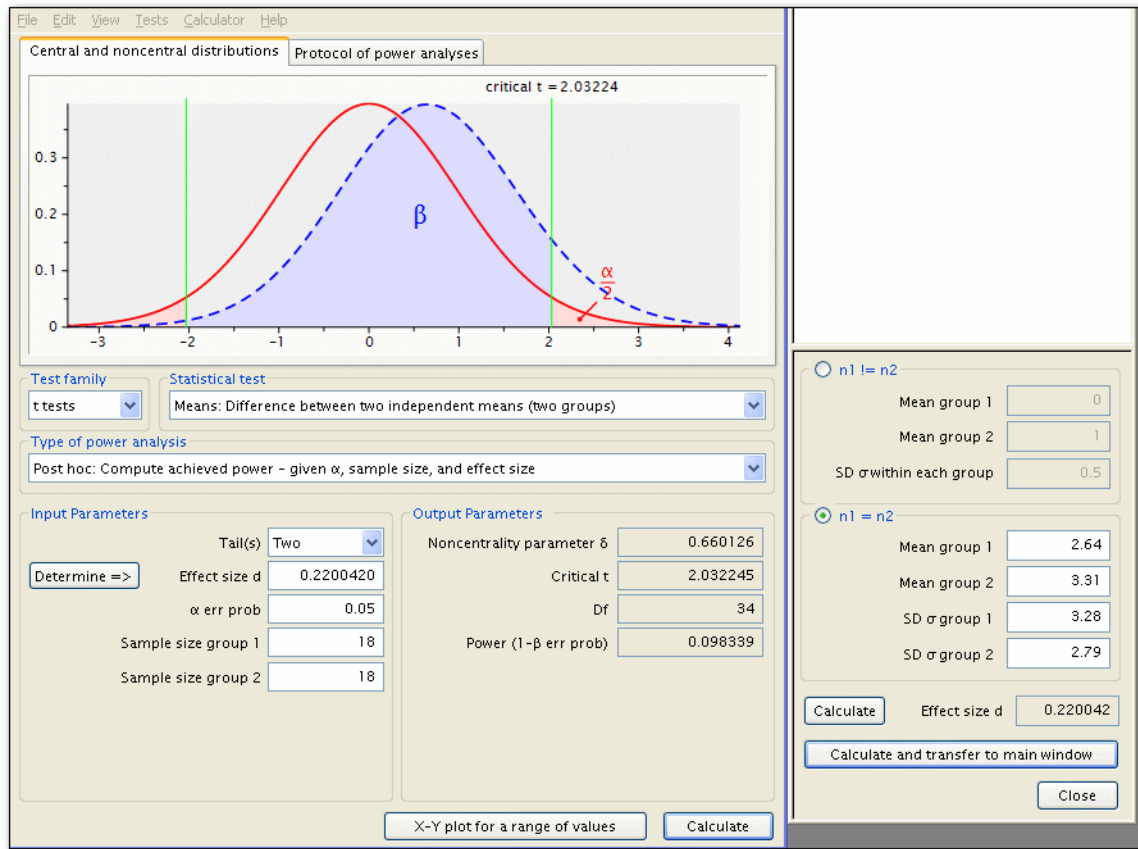
 Démarrer G*Power.

1. À **Test family**, choisir: t tests
2. À **Statistical test**, choisir: Means: Difference between two independent means (two groups)
3. À **Type of power analysis**, choisir: Post hoc: Compute achieved power - given α , sample size, and effect size
4. Dans **Input Parameters**,
à la boîte **Tail(s)**, choisir: Two,
vérifier que **α err prob** est égal à 0.05
Inscrire 18 pour **Sample size group 1** et 2

puis, pour calculer la taille d'effet (Effect size d), cliquer sur le bouton **Determine =>**

5. Dans la fenêtre qui s'ouvre à droite,
sélectionner **$n1 = n2$** , puis
entrer les moyennes (**Mean group 1** et 2)
entrer les écarts types (**SD s group 1** et 2)
et cliquer sur le bouton **Calculate and transfer to main window**
6. Cliquer sur le bouton Calculate dans la fenêtre principale et vous devriez obtenir ceci:

Figure 1.



Étudions un peu ce graphique.

- La courbe de gauche, en rouge, correspond à la distribution de la statistique t si H_0 est vraie (i.e si les deux moyennes étaient égales) compte tenu de l'effectif (18 dans chaque région) et des écarts-types observés.
- Les lignes verticales vertes correspondent aux valeurs critiques de t pour une valeur $\alpha=0.05$ et un effectif total de 36 (2×18).
- Les régions ombrées en rose correspondent aux zones de rejet de H_0 . Si Jaynie avait obtenu une valeur de t en dehors de l'intervalle délimité par les valeurs critiques allant de -2.03224 à 2.03224, alors elle aurait rejeté H_0 , l'hypothèse nulle d'égalité des deux moyennes. En fait, elle a obtenu une valeur de t égale à -0.5746 et conclu que la biomasse est la même dans les deux régions.
- La courbe de droite, en bleu, correspond à la distribution de la statistique t si H_1 est vraie (ici H_1 correspond à une différence de biomasse entre les deux régions de $3.33-2.64=0.69\text{g/m}^2$, compte tenu des écarts-types observés). Cette distribution correspond à ce qu'on devrait s'attendre à observer si H_1 était vraie et que l'on

répétait un grand nombre de fois les mesures dans des échantillons aléatoires de 18 ruisseaux des deux régions en calculant la statistique t à chaque fois. En moyenne, on observerait une valeur de t d'environ 0.6.

- Notez que la distribution de droite chevauche considérablement celle de gauche, et une bonne partie de la surface sous la courbe de droite se retrouve à l'intérieur de l'intervalle d'acceptation de H_0 , délimité par les deux lignes vertes et allant de -2.03224 à 2.03224. Cette proportion, correspondant à la partie ombrée en bleu sous la courbe de droite et dénoté par β correspond au risque d'erreur de type II qui est d'accepter H_0 quand H_1 est vraie.
- La puissance est simplement $1-\beta$, et est ici de 0.098339. Donc, si la biomasse différait de $0.69\text{g}/\text{m}^2$ entre les deux régions, Jaynie n'avait que 9.8% des chances d'être capable de détecter une différence statistiquement significative à $\alpha=5\%$ en échantillonnant 18 ruisseaux de chaque région.

Récapitulons: La différence de biomasse entre les deux régions n'est pas statistiquement significative d'après le test de t . C'est donc que cette différence est relativement petite compte tenu de la précision des mesures. Il n'est donc pas très surprenant que la puissance, i.e. la probabilité de détecter une différence significative, soit faible. Toute cette analyse ne nous informe pas beaucoup.

Une analyse de puissance post hoc avec la taille de l'effet observé n'est pas très utile. On la fera plutôt pour une taille d'effet autre que celle observée quand H_0 est acceptée. Quelle taille d'effet utiliser? C'est la biologie du système étudié qui peut nous guider. Par exemple, en ce qui concerne la biomasse des poissons, on pourrait s'attendre à ce qu'une différence de biomasse du simple au double (disons de 2.64 à $5.28\text{ g}/\text{m}^2$) ait des conséquences écologiques. On voudrait s'assurer que Jaynie avait de bonnes chances de détecter une différence aussi grande que celle-là avant d'accepter ses conclusions que la biomasse est la même entre les deux régions. Quelles étaient les chances de Jaynie de détecter une différence de $2.64\text{ g}/\text{m}^2$ entre les deux régions? G*Power peut nous le dire.


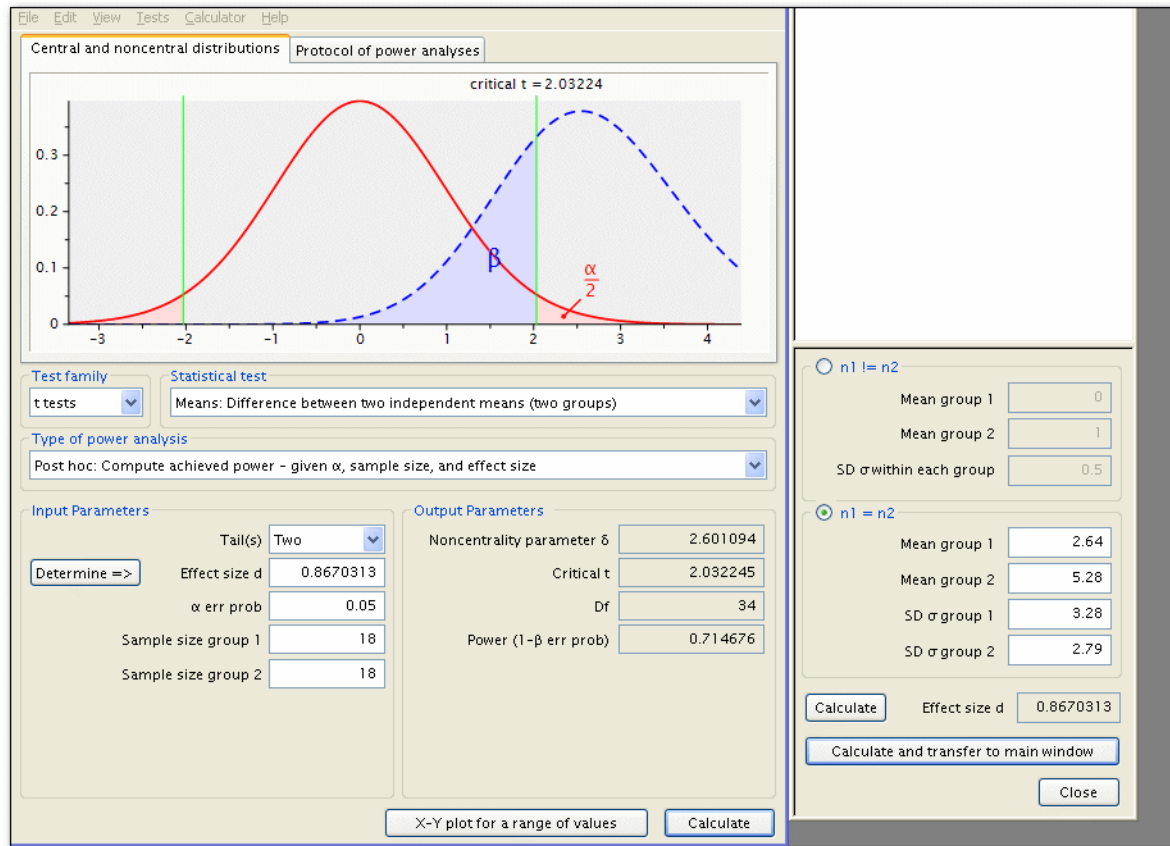
 Changer la moyenne du groupe 2 à 5.28, recalculer la taille d'effet, et cliquer sur Calculate pour obtenir:

Figure 2.



La puissance est de 0.71, donc Jaynie avait une chance raisonnable de détecter une différence du simple au double avec 18 ruisseaux dans chaque région.

Notez que cette analyse de puissance post hoc pour une taille d'effet jugée biologiquement significative est bien plus informative que l'analyse précédente pour la taille d'effet observée (qui est celle effectuée par défaut par bien des néophytes et de trop nombreux logiciels qui essaient de penser pour nous). En effet, Jaynie n'a pu détecter de différences significatives entre les deux régions. Cela pourrait être pour deux raisons: soit qu'il n'y a pas de différences entre les régions, ou soit parce que la précision des mesures est si faible et l'effort d'échantillonnage était si limité qu'il était très peu probable de détecter même d'énormes différences. La deuxième analyse de puissance permet d'éliminer cette seconde possibilité puisque Jaynie avait 71% des chances de détecter une différence du simple au double.

Analyse a priori

Supposons qu'on puisse défendre la position qu'une différence de biomasse observée par Jaynie entre les deux régions, $3.31 - 2.64 = 0.67 \text{ g/m}^2$, soit écologiquement significative. On devrait donc planifier la prochaine saison d'échantillonnage de manière à avoir de bonnes chances de détecter une différence de cette taille. Combien de ruisseaux Jaynie devrait-elle échantillonner pour avoir 80% des chances de la détecter (compte tenu de la variabilité observée)?


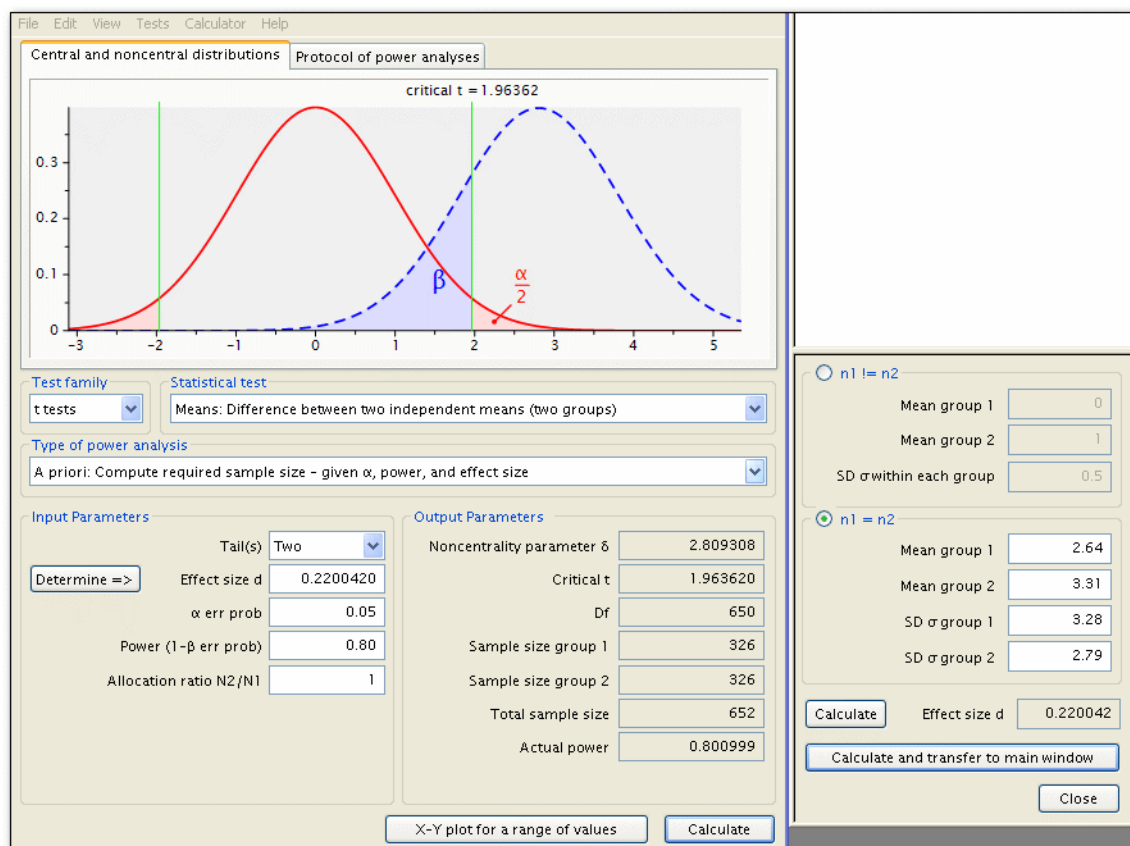
 Changer le type d'analyse de puissance dans G*Power à A priori: **Compute sample size - given α , power, and effect size**. Assurez-vous que les valeurs pour les moyennes et les écarts-type soient celles qu'a obtenu Jaynie, recalculez la taille de l'effet, et inscrivez 0.8 pour la puissance et vous obtiendrez:

Figure 3.



Ouch! Il faudrait échantillonner 326 ruisseaux dans chaque région! Cela coûterait une fortune et exigerait de nombreuses équipes de travail. Sans cela, on ne pourrait échantillonner que quelques dizaines de ruisseaux, et il serait peu probable que l'on puisse détecter une si

faible différence de biomasse entre les deux régions. Ce serait vraisemblablement en vain et pourrait être considéré comme une perte de temps: pourquoi tant d'efforts et de dépenses si les chances de succès sont si faibles.

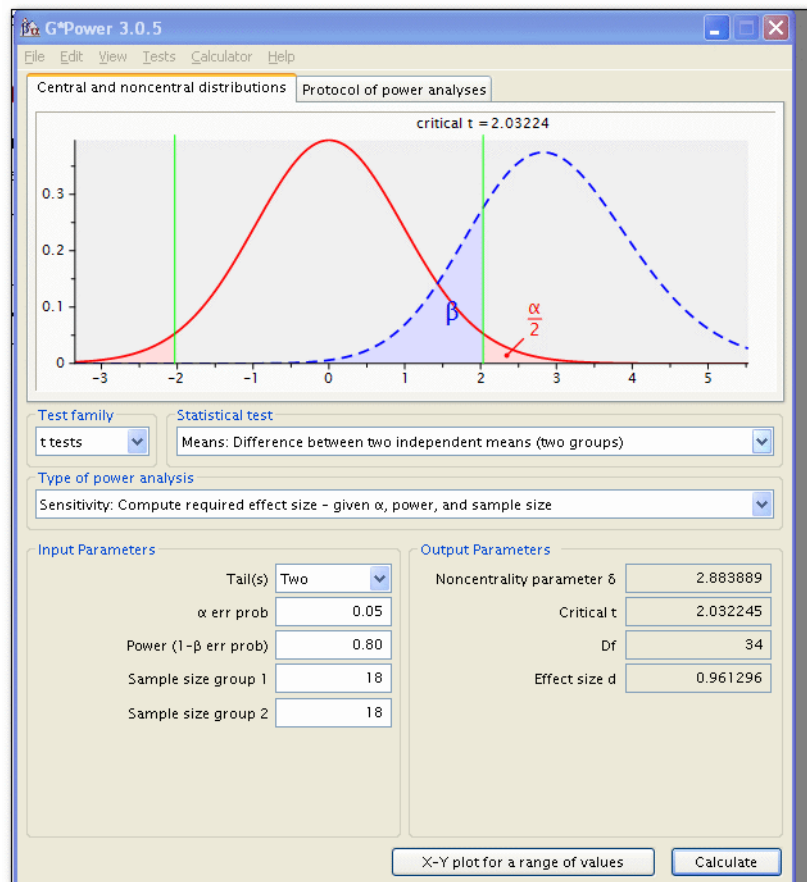
☞ Si on refait le même calcul pour une puissance de 95%, on obtient 538 ruisseaux par région. Augmenter la puissance ça demande plus d'effort.

Analyse de sensibilité - Calculer la taille d'effet détectable

Compte tenu de la variabilité observée, d'un effort d'échantillonnage de 18 ruisseaux par région, et en conservant $\alpha=0.05$, quelle est la taille d'effet que Jaynie pouvait détecter avec 80% de chances ($\beta=0.2$)?

☞ Changez le type d'analyse dans G*Power à Sensitivity: **Compute required effect size - given α , power, and sample size** et assurez-vous que la taille des échantillons est de 18 dans chaque région. Vous obtiendrez:

Figure 4.



La taille d'effet détectable pour cette taille d'échantillon, $\alpha=0.05$ et $\beta=0.2$ (ou une puissance de 80%) est de 0.961296. **Attention**, cette valeur est l'indice d de la taille de l'effet et est pondérée par la variabilité des mesures. Dans ce cas ci, d est approximativement égal à

$$d = \frac{|\bar{X}_1 - \bar{X}_2|}{\sqrt{\frac{s_1^2 + s_2^2}{2}}}$$

Pour convertir cette valeur de d sans unités en une valeur de différence de biomasse détectable, il suffit de multiplier d par le dénominateur de l'équation:

$$\begin{aligned} \text{différence détectable} &= |\bar{X}_1 - \bar{X}_2| = d \sqrt{\frac{s_1^2 + s_2^2}{2}} \\ \text{différence détectable} &= 0.961296 \sqrt{\frac{3.28^2 + 2.79^2}{2}} \\ \text{différence détectable} &= 2.927024 \end{aligned}$$

Donc, avec 18 ruisseaux dans chaque région, pour $\alpha=0.05$ et $\beta=0.2$ (une puissance de 80%), Jaynie pouvait détecter une différence de biomasse de 2.93g/m² entre les régions, un peu plus que du simple au double.

Points à retenir

- L'analyse de puissance post hoc n'est pertinente que lorsque l'on a accepté l'hypothèse nulle. Il est en effet impossible de faire une erreur de type II quand on rejette H_0 .
- Avec de très grands échantillons, on a une puissance quasi infinie et on peut détecter statistiquement de très petites différences qui ne sont pas nécessairement biologiquement significatives.
- En utilisant un critère de signification plus strict ($\alpha < 0.05$) on diminue notre puissance.
- En voulant maximiser la puissance, on augmente l'effort requis, à moins d'utiliser une valeur critique plus libérale ($\alpha > 0.05$)
- Le choix de β est quelque peu arbitraire. On considère que $\beta=0.2$ (puissance de 80%) est relativement élevé.

Exercice

Les larves de mouches noires (Diptera: Simuliidae) ont été échantillonnées en février à l'émissaire de deux lacs des Cantons de l'Est (lacs Orford et Lovering). La longueur de chaque larve a été mesurée. Les données sont dans le fichier `simulies.RData`

La relation entre la longueur (L, en mm) et la masse (M, en μg) pour l'espèce dominante (*P. mixtum/fuscum*) est:

$$M = 1.36L^{3.05}$$

1. Calculer la masse moyenne et l'écart-type à chaque site.
2. En utilisant la masse moyenne de *P. mixtum/fuscum* à Lovering comme référence et les écarts-types observés aux 2 sites, calculer la puissance d'un test de t bilatéral pour moyennes indépendantes
 - a) si la différence de masse est de 5 μg , $\alpha=0.05$, et qu'on échantillonnait 100 larves à chaque site
 - b) si la différence de masse est de 20 μg , $\alpha=0.05$, et qu'on échantillonnait 100 larves à chaque site
 - c) si la différence de masse est de 50 μg , $\alpha=0.05$, et qu'on échantillonnait 100 larves à chaque site
3. Calculer la taille d'échantillon requis pour pouvoir détecter, par un test de t bilatéral pour moyennes indépendantes en tenant compte des écarts-types observés, une différence de 50 μg entre les moyennes
 - a) avec une puissance de 80% et $\alpha=0.05$
 - b) avec une puissance de 80% et $\alpha=0.001$
 - c) avec une puissance de 95% et $\alpha=0.05$
4. Calculer la taille d'effet détectable (d) par un test de t bilatéral pour moyennes indépendantes, compte tenu des écarts-types observés
 - a) avec une puissance de 80%, $\alpha=0.05$, et des mesures sur 10 larves de chaque site
 - b) avec une puissance de 80%, $\alpha=0.05$, et des mesures sur 200 larves à chaque site
 - c) avec une puissance de 80%, $\alpha=0.05$, et des mesures sur 20 larves d'un site et sur 380 larves au second site.

Calculer la différence de masse, en μg , qui est détectable d'après vos estimés de la taille minimale d'effet détectable à 4a, b, et

Labo- Corrélation linéaire et régression linéaire simple

Après avoir complété cet exercice de laboratoire, vous devriez être en mesure de :


- Utiliser R pour produire un diagramme de dispersion pour illustrer la relation entre deux variables avec trace lowess
- Utiliser R pour faire des transformations simples
- Utiliser R pour calculer le coefficient de corrélation de Pearson entre deux variables et en évaluer sa signification statistique.
- Utiliser R pour calculer la corrélation de rang entre des paires de variables avec le r de Spearman et le tau de Kendall.
- Utiliser R pour évaluer la signification de corrélations dans une matrice de corrélation en utilisant les probabilités ajustées par la méthode de Bonferroni.
- Utiliser R pour faire une régression linéaire simple.
- Utiliser R pour évaluer si un ensemble de données remplit les conditions d'application d'une analyse de régression simple.
- Quantifier la taille de l'effet d'une régression simple et effectuer une analyse de puissance avec G*Power.

Diagrammes de dispersion

Les analyses de corrélation et de régression devraient toujours commencer par un examen des données. C'est une étape critique qui sert à évaluer si ce type d'analyse est approprié pour un ensemble de données.

Supposons que nous sommes intéressés à évaluer si la longueur d'esturgeons mâles dans la région de The Pas covarie avec leur poids. Pour répondre à cette question, regardons d'abord la corrélation entre `fklnlngth` et `rdwgth`.

Souvenez-vous qu'une des conditions d'application de l'analyse de corrélation est que la relation entre les deux variables est linéaire. Pour évaluer cela, commençons par un diagramme de dispersion :

 En utilisant le fichier `sturgeon.Rdata`, et après avoir attaché le `data.frame` `sturgeon`, faites un diagramme de dispersion avec une

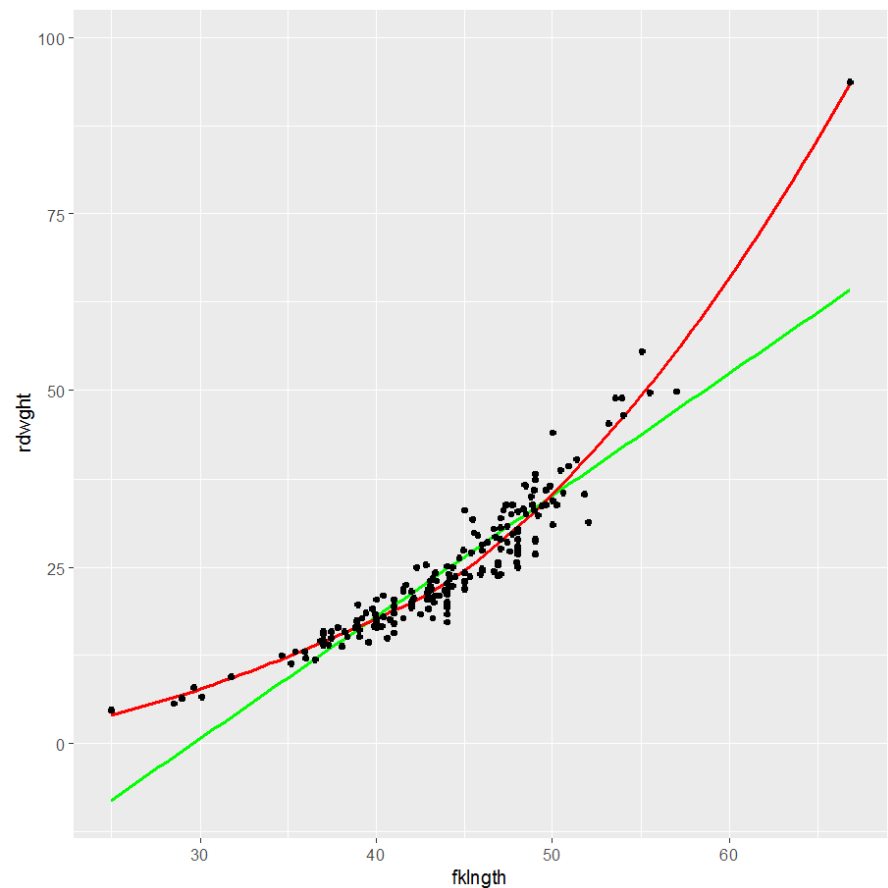
droite de régression et une courbe LOWESS de fklngth en fonction de rdwght en tapant:

```
mygraph<-ggplot(
  data=sturgeon, # source of data
  aes(x=fklngth, y=rdwght)) #aesthetics: x=fklngth, y=rdw-
ght
# plot data points, regression, loess trace
mygraph<-mygraph+
  stat_smooth(method=lm, se=FALSE, color="green") + # add
linear regression, but no SE shading
  stat_smooth(color="red")+ #add loess
  geom_point() # add data points
mygraph # display graph
```

(notez ici la gymnastique requise pour éliminer les données manquantes qui posent problème pour obtenir la trace loess).

☞ Est-ce que la dispersion des points suggère une bonne corrélation entre les deux variables? Est-ce que la relation semble linéaire?

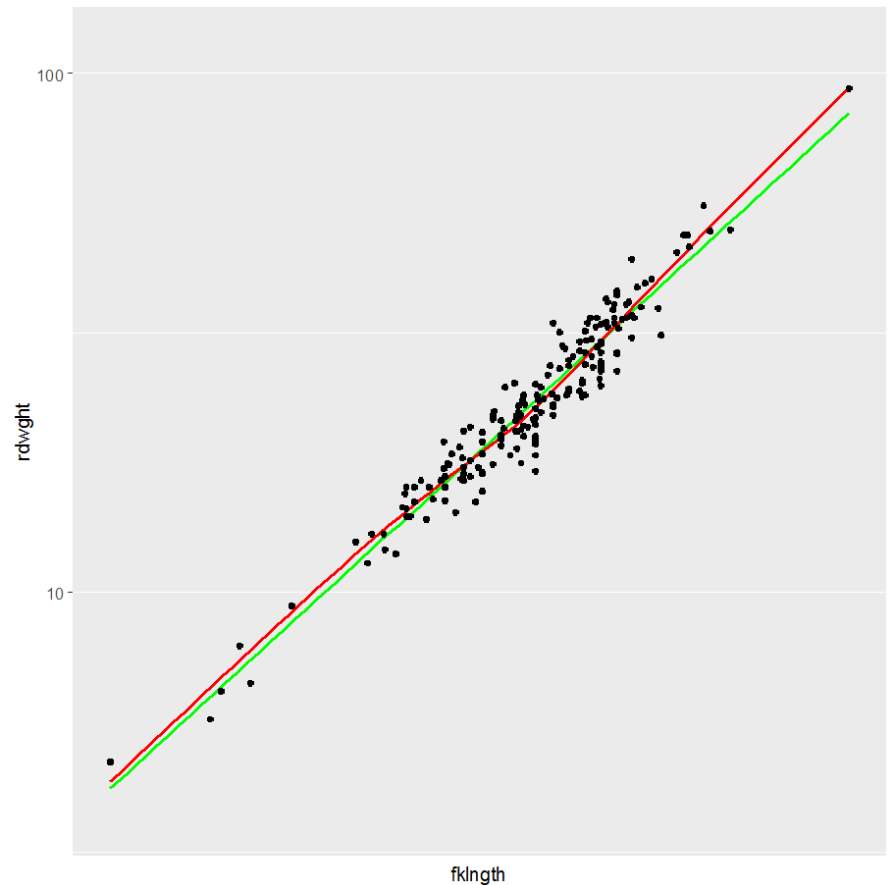
Figure 1.



Ce graphique suggère une tendance plus curvilinéaire que linéaire. Malgré tout, il semble y avoir une forte corrélation entre les deux variables.

☞ Refaites le diagramme de dispersion avec des axes logarithmiques:

```
# apply log transformation on defined graph
mygraph+scale_x_log10()+scale_y_log10()
```




Comparez les diagrammes de dispersion avant et après transformation (Figures 17 et). Comme l'analyse de corrélation présuppose une relation linéaire entre les variables, on devrait donc privilégier l'analyse sur les données log-transformées.

Transformations et le coefficient de corrélation

Une autre condition préalable à l'analyse de corrélation est que les deux variables concernées suivent une distribution normale bidimensionnelle. On peut aisément vérifier la normalité de chacune

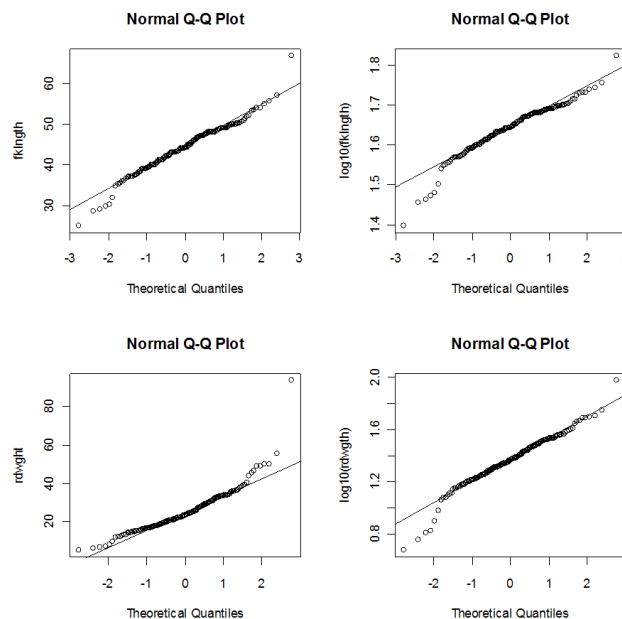
des 2 variables séparément tel que décrit dans le laboratoire précédent. Si les deux variables sont normalement distribuées, on présume généralement qu'elles suivent une distribution normale bidimensionnelle lorsqu'analysées simultanément (notez que ce n'est pas toujours le cas cependant).

 Examinez la distribution des quatre variables (les deux variables originales et les variables transformées). Que concluez-vous de l'inspection visuelle de ces graphiques ?

Les figures ci-dessous sont les diagrammes de probabilité (`qqplots()`). Le code pour produire des graphiques multiples sur une page, comme on voit ci-dessous, est):

```
attach(sturgeon)
par(mfrow = c(2, 2))
qqnorm(fklength, ylab="fklength")
qqline(fklength)
qqnorm(log10(fklength), ylab="log10(fklength)")
qqline(log10(fklength))
qqnorm(rdwght, ylab="rdwght")
qqline(rdwght)
qqnorm(log10(rdwght), ylab="log10(rdwght)")
qqline(log10(rdwght))
par(mfrow = c(1, 1))
```

Figure 2.



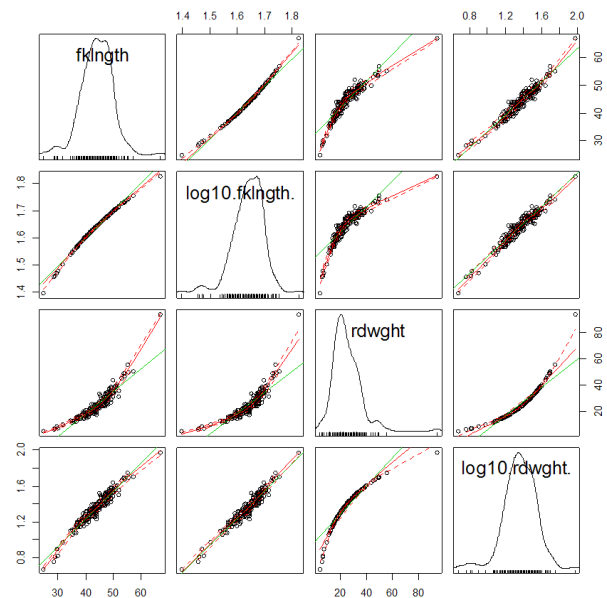
Il n'y a pas grand-chose à redire: aucune des distributions n'est parfaitement normale, mais les déviations semblent mineures.

☞ Générez une matrice de graphiques de dispersion améliorés en utilisant la commande `scatterplot.matrix` de la librairie `car`.

```
scatterplot.matrix(~fklength+log10(fklength)+rdwght+log10(rdwght), reg.line=lm, smooth=TRUE, span=0.5, diagonal = 'density')
```

pour obtenir

Figure 3.



☞ Ensuite, calculez le coefficient de corrélation de Pearson entre chaque paire (variables originales et logtransformées) en utilisant la commande `cor()`. Avant de commencer, on va cependant ajouter les variables transformées au tableau de données `sturgeon`:

```
sturgeon$lfklength <- with(sturgeon, log10(fklength))  
sturgeon$lrdwght <- with(sturgeon, log10(rdwght))
```

Vous pouvez ensuite obtenir la matrice de corrélation par:

```
cor(sturgeon[,c("fklength", "lfklength", "lrdwght", "rdwght")],  
use="complete.obs")
```

Fréquemment, il y a des données manquantes dans un échantillon. En choisissant **`use="complete.obs"`**, toutes les lignes du fichier pour lesquelles les variables ne sont pas toutes mesurées sont éliminées. Dans ce cas, toutes les corrélations seront calculées avec le même nombre de cas. Par contre, en utilisant **`use="pairwise.complete.obs"`**, R élimine une observation que lorsqu'un des deux membres de la

paire a une valeur manquante. Dans ce cas, si les données manquantes pour différentes variables se retrouvent dans un groupe différent d'observation, les corrélations ne seront pas nécessairement calculées sur le même nombre de cas ni sur le même sous-ensemble de cas. En général, vous devriez utiliser l'option **use="complete.obs"** à moins que vous ayez un très grand nombre de données manquantes et que cette façon de procéder élimine la plus grande partie de vos observations.


Pourquoi la corrélation entre les variables originales est-elle la plus faible des trois ?

	fklngh	lfklngh	lrdwght	rdwght
fklngh	1.0000000	0.9921435	0.9645108	0.9175435
lfklngh	0.9921435	1.0000000	0.9670139	0.8756203
lrdwght	0.9645108	0.9670139	1.0000000	0.9265513
rdwght	0.9175435	0.8756203	0.9265513	1.0000000

Il y a plusieurs choses à noter ici. Premièrement, la corrélation entre la longueur à la fourche et le poids rond est élevée, peu importe la transformation: les poissons lourds ont tendance à être longs. Deuxièmement, la corrélation est plus forte pour les données transformées que pour les données brutes. Pourquoi? Parce que le coefficient de corrélation est inversement proportionnel au bruit autour de la relation linéaire. Si la relation est curvilinéaire (comme dans le cas des données non transformées), le bruit est plus grand que si la relation est parfaitement linéaire. Par conséquent, la corrélation est plus faible.

Matrices de corrélations et correction de Bonferroni

Une pratique courante est d'examiner une matrice de corrélation à la recherche des associations significatives. Comme exemple, essayons de tester si la corrélation entre `lfklngh` et `rdwght` est significative (c'est le plus faible coefficient de corrélation de cette matrice).

 De la fenêtre R console, inscrivez le code suivant pour tester la corrélation entre `lfklngh` et `rdwght`:

```
>cor.test(sturgeon$lfklngh, sturgeon$rdwght, alternative="two.sided", method="pearson")
```

Pearson's product-moment correlation

```
data: sturgeon$lfklngh and sturgeon$rdwght
t = 24.3223, df = 180, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8367345 0.9057199
sample estimates:
cor
0.8756203
```


On voit ici que la corrélation est hautement significative ($p < 2.2e-16$), ce qui n'est pas surprenant étant donné la valeur du coefficient de corrélation (0.8756).

Il est important de réaliser que si une matrice contient un grand nombre de corrélations, il n'est pas surprenant d'en trouver au moins une qui soit "significative". En effet, on s'attend à en trouver 5% en moyenne lorsqu'il n'y a en fait aucune corrélation entre les paires de moyennes. Une façon de corriger pour cette tendance est d'ajuster le niveau α critique auquel on attribue une signification statistique en divisant α par le nombre k de corrélations qui sont examinées : $\alpha' = \alpha/k$ (ajustement de Bonferroni). Si initialement $\alpha = 0.05$ et qu'il y a 10 corrélations qui sont examinées, alors $\alpha' = 0.005$. Donc, afin de rejeter l'hypothèse nulle, la valeur de p devra être plus petite que α' , en l'occurrence 0.005.

Dans l'exemple qui précède, on devrait donc ajuster α critique en divisant par le nombre total de corrélations dans la matrice (6 dans ce cas, donc $\alpha' = 0.00833$). Cette correction modifie-t-elle votre conclusion quant à la corrélation entre `lklfl` et `rdwght`?

Corrélations non paramétriques : r de Spearman et tau de Kendall

L'analyse faite à la section précédente avec les esturgeons suggère que l'une des conditions préalables à l'analyse de corrélation, soit la distribution normale bidimensionnelle de données, pourrait ne pas être remplie pour `fklnth` et `rdwght`, ni pour les paires de variables transformées. La recherche d'une transformation appropriée peut parfois être difficile. Pire encore, pour certaines distributions il n'existe pas de transformation qui va normaliser les données. Dans ces cas-là, la meilleure option est de faire une analyse non paramétrique qui ne présume ni de la normalité ni de la linéarité. Ces analyses sont basées sur les rangs. Les deux plus communes sont le coefficient de rang de Spearman et le tau de Kendall.

 Allez à la fenêtre **R console**, et testez la corrélation entre `fklnth` et `rdwght` en utilisant Spearman et Kendall's . Les commandes qui suivent produiront les corrélations:

```
>cor.test(sturgeon$fklnth, sturgeon$rdwght, alternative="two.sided", method="spearman")
```

```
Spearman's rank correlation rho
```

```
data: sturgeon$fklnth and sturgeon$rdwght
S = 47971.33, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
```

```

sample estimates:
      rho 
0.9522546

>cor.test(sturgeon$fklngh, sturgeon$rdwght, alterna-
tive="two.sided", method="kendall")Kendall's rank correla-
tion tau

Kendall's rank correlation tau

data: sturgeon$fklngh and sturgeon$rdwght
z = 16.3578, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau 
0.8208065

```

Comparer les résultats de cette analyse à l'analyse paramétrique.
Pourquoi y-a-t'il une différence ?

Calculez les corrélations non paramétriques sur les paires de variables transformées. Vous devriez voir tout de suite que les corrélations des données transformées et non transformées sont identiques puisque dans les deux cas la corrélation est calculée à partir des rangs qui ne sont pas affectés par la transformation.

Notez que les corrélations obtenues avec le tau de Kendall (0.820) sont plus faibles que celles du coefficient de Spearman (0.952). Le tau pondère un peu plus les grandes différences entre les rangs alors que le coefficient de Spearman donne le même poids à chaque paire d'observations. En général, on préfère le tau de Kendall lorsqu'il y a plus d'incertitude quant aux rangs qui sont près les uns des autres.


Les esturgeons de cet échantillon ont été capturés à l'aide de filets et d'hameçons d'une taille fixe. Quel impact cette méthode de capture peut-elle avoir eu sur la forme de la distribution de `fklngh` et `rdwght` ? Compte tenu de ces circonstances, l'analyse de corrélation est-elle appropriée ?

Rappelez-vous que l'analyse de corrélation présume aussi que *chaque variable est échantillonnée aléatoirement*. Dans le cas de nos esturgeons, ce n'est pas le cas: les hameçons appâtés et les filets ne capturent pas de petits esturgeons (et c'est pourquoi il n'y en a pas dans l'échantillon). Il faut donc réaliser que les coefficients de corrélation obtenus dans cette analyse ne reflètent pas nécessairement ceux de la population totale des esturgeons.

Régression linéaire simple

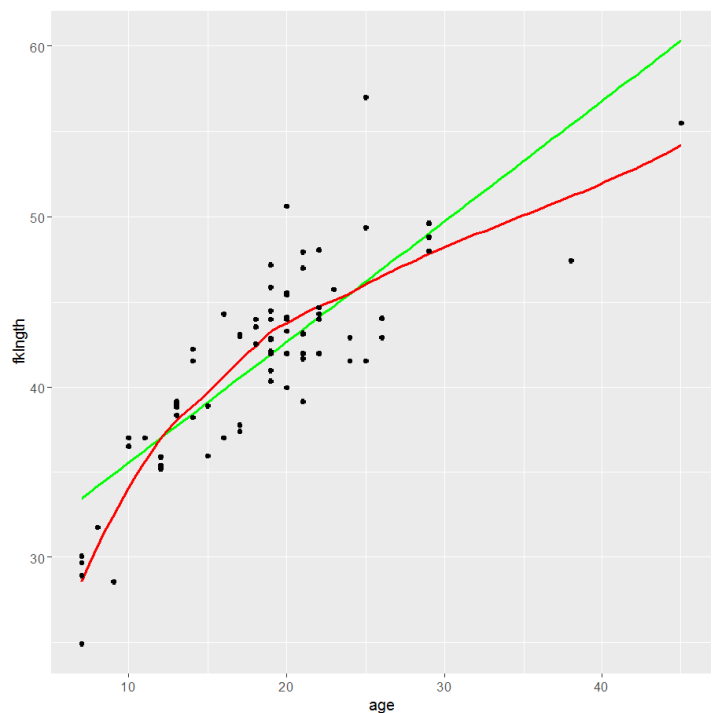
L'analyse de corrélation vise à décrire comment deux variables covarient. L'analyse de régression vise plutôt à produire un modèle permettant de prédire une variable (la variable dépendante) par l'autre (la variable indépendante).

Comme pour l'analyse de corrélation, on devrait commencer en examinant des graphiques. Puisque l'on est intéressé à quantifier la relation entre deux variables, un graphique de la variable dépendante (Y) en fonction de la variable indépendante (X) est tout à fait approprié.

 Le fichier `sturgeon.Rdata` contient des données d'un inventaire des esturgeons récoltés en 1978-1980 à Cumberland House en Saskatchewan et à The Pas au Manitoba. Faites un diagramme de dispersion de `fklnlngth` (la variable dépendante) en fonction de `age` (la variable indépendante) pour les esturgeons mâles et ajoutez-y une régression linéaire et une trace lowess. Que concluez-vous de ce diagramme de dispersion ?

```
sturgeon.male <- subset(sturgeon, subset=sex=="MALE")
lmygraph<-ggplot(
  data=sturgeon.male, # source of data
  aes(x=age, y=fklnlngth)) #aesthetics: y=fklnlngth, x=rdwght
# plot data points, regression, loess trace
mygraph<-mygraph+
  stat_smooth(method=lm, se=FALSE, color="green") + # add linear regression,
  but no SE shading
  stat_smooth(color="red")+ #add loess
  geom_point() # add data points
mygraph # display graph
```

Figure 4.



Ce graphique suggère que la relation n'est pas linéaire.

Supposons que nous désirions estimer le taux de croissance des esturgeons mâles. Un estimé (peut-être pas terrible...) du taux de croissance peut être obtenu en calculant la pente de la régression de la longueur à la fourche sur l'âge.

Ajustons d'abord une régression avec la commande `lm()` et sauvons ces résultats dans un objet appelé `RegModel.1`

```
RegModel.1 <- lm(fklength~age, data=sturgeon.male)
```

Rien n'apparaît à l'écran, mais ne vous inquiétez pas, tout a été sauvegardé en mémoire.

Pour voir ces résultats, tapez:

```
summary(RegModel.1)
```

call:

```
lm(formula = fklength ~ age, data = sturgeon.male) ❶
```

Residuals: ❷

Min	1Q	Median	3Q	Max
-8.4936	-2.2263	0.1849	1.7526	10.8234

Coefficients: ❸

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	28.50359	1.16873	24.39	<2e-16 ***
age	0.70724	0.05888	12.01	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: ❹ 3.307 on 73 degrees of freedom
(5 observations deleted due to missingness)

Multiple R-squared: 0.664 ❺, Adjusted R-squared: 0.6594 ❻

F-statistic: 144.3 on 1 and 73 DF, p-value: < 2.2e-16 ❼

- ❶ Le modèle qui a été ajusté et les données utilisées.
- ❷ Un sommaire statistique des résidus autour du modèle estimé.
- ❸ Valeurs estimées des paramètres du modèle, erreurs-types, statistiques t et probabilités associées.
- ❹ Racine carrée de la variance résiduelle.
- ❺ Coefficient de détermination. Il correspond à la proportion de la variabilité de la variable dépendante qui peut être expliquée par la régression.
- ❻ Le R-carré ajusté tient compte du nombre de paramètres du modèle. Si vous voulez comparer différents modèles qui n'ont pas le même nombre de paramètres, c'est ce qu'il faut utiliser.
- ❼ C'est le test de signification omnibus du modèle. Dans le cas de la régression simple, il est équivalent au test sur la pente de la régression.

La régression estimée est donc:

$$Fklength = 28.50359 + 0.70724 * age$$

Étant donné la valeur très significative du test de F ainsi que pour le test de t pour la pente de la droite, on rejette l'hypothèse nulle qu'il n'y a pas de relation entre la taille et l'âge.

Vérifier les conditions d'application de la régression

La régression simple de modèle I a quatre conditions préalables :

1. il n'y a pas d'erreur de mesure sur la variable indépendante (X),
2. la relation entre Y et X est linéaire,
3. les résidus sont normalement distribués et
4. la variance des résidus est constante pour toutes les valeurs de la variable indépendante.

Procédons maintenant à l'examen post-mortem. La première condition est rarement remplie avec des données biologiques ; il y a presque toujours de l'erreur sur X et sur Y. Cela veut dire qu'en général les pentes estimées sont biaisées, mais que les valeurs prédites ne le sont pas. Toutefois, si l'erreur de mesure sur X est petite par rapport à l'étendue des valeurs de X, le résultat de l'analyse n'est pas dramatiquement influencé. Par contre, si l'erreur de mesure est relativement grande (toujours par rapport à l'étendue des valeurs de X), la droite de régression obtenue par la régression de modèle I est un piètre estimé de la relation fonctionnelle entre X et Y. Dans ce cas, il est préférable de passer à la régression de modèle II, malheureusement au-delà du contenu de ce cours.

Les autres conditions préalables à l'analyse de régression de modèle I peuvent cependant être vérifiées, ou du moins évaluées visuellement. La commande `plot()` permet de visualiser des graphiques diagnostiques pour des modèles linéaires.

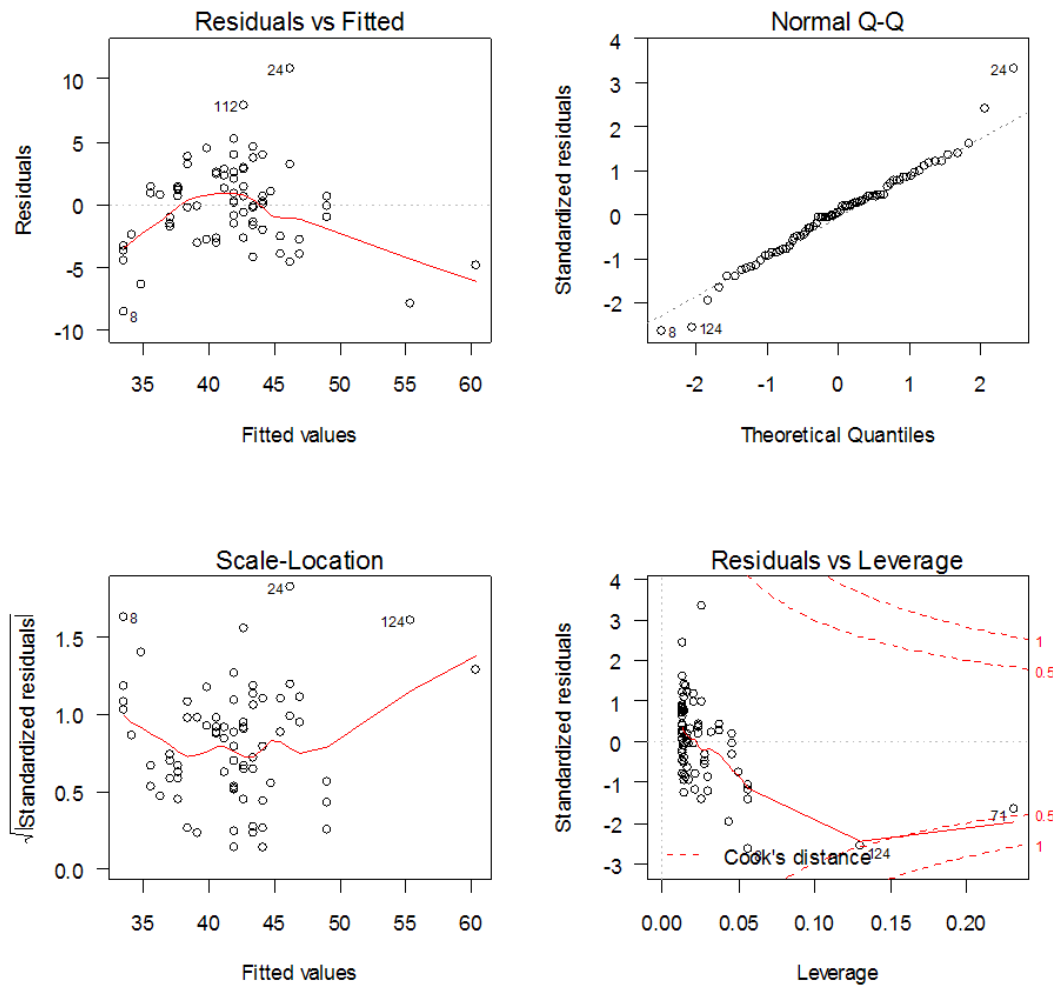
```
par(mfrow = c(2, 2), las=1)  
plot(RegModel.1)
```

La commande `par()` est utilisée pour dire à R de tracer 2 rangées et 2 colonnes de graphiques par page (il y a quatre graphiques diagnostiques qui sont générés automatiquement pour les modèles

linéaires), et la commande las indique à R d'effectuer une rotation des légendes de l'axe des Y pour qu'elles soient perpendiculaires à l'axe (oui. Je sais. Rien de tout ça n'est évident.)

Vous obtiendrez:

Figure 5.



Le premier graphique (en haut à gauche) permet d'évaluer la linéarité, la normalité, et l'homoscédasticité des résidus. Il illustre les déviations autour de la régression en fonction des valeurs prédites. Rappelez-vous que le graphique de `fklngh` vs `age` suggère que la relation entre la longueur à la fourche et l'âge n'est pas linéaire. Les très jeunes et très vieux esturgeons sont sous la droite en général, alors que les esturgeons d'âge moyen sont retrouvés généralement au-dessus de la droite de régression. C'est exactement ce que le graphique des résidus en fonction des valeurs prédites illustre. La ligne en rouge est une

trace lowess au travers de ce nuage de points. Si la relation était linéaire, la trace lowess serait presque plate et près de 0. La dispersion des résidus permet d'évaluer visuellement leur normalité et hétéroscédasticité; mais ce graphique n'est pas optimal pour évaluer ces propriétés. Les deux graphiques suivants sont supérieurs au premier pour cela.

Le deuxième graphique permet d'évaluer la normalité des résidus. C'est un graphique QQ des résidus (QQ plot). Des résidus distribués normalement tomberaient exactement sur la diagonale. Ici, on voit que c'est presque le cas, sauf dans les queues de la distribution.

Le troisième graphique, en bas à gauche, intitulé Scale-Location, permet d'évaluer l'homoscédasticité. On y retrouve sur l'ordonnée (l'axe des y) la racine carrée de la valeur absolue des résidus standardisés (résidus divisés par l'écart-type des résidus) en fonction des valeurs prédites. Le graphique aide à déterminer si la variation des résidus est constante ou non. Si les résidus sont homoscédastiques, la valeur moyenne sur l'axe des y ne va pas changer en fonction de la valeur prédite. Ici, il y a une certaine tendance, mais pas une tendance monotone puisqu'il y a d'abord une baisse puis une hausse.; bref, rien qui soit une forte évidence contre la supposition d'homoscédasticité.

Le quatrième graphique, en bas à droite, montre les résidus en fonction du "leverage" et permet de détecter certaines valeurs extrêmes qui ont une grande influence sur la régression. Le leverage d'un point mesure sa distance des autres points, mais seulement en ce qui concerne les variables indépendantes. Dans le cas d'une régression simple, cela revient à la distance entre le point sur l'axe des x et la moyenne de tous les points sur cet axe. Vous devriez porter une attention particulière aux observations qui ont un leverage plus grand que $2(k+1)/n$, où k est le nombre de variables indépendantes (ici, 1) et n est le nombre d'observations. Dans cet exemple, il y a 75 observations et une variable indépendante et donc les points ayant un leverage plus grand que $4/75 = 0.053$ devraient être considérés avec attention. Le graphique indique également comment la régression changerait si on enlevait un point. Ce changement est mesuré par la distance de Cook, illustrée par les bandes en rouge sur le graphique. Un point ayant une distance de Cook supérieure à 1 a une grande influence.

Attrappe: Notez que R identifie automatiquement les cas les plus extrêmes sur chacun de ces 4 graphiques. Le fait qu'un point soit identifié ne signifie pas nécessairement que c'est une valeur réellement

extrême, ou que vous devez vous en préoccuper. Dans tous les ensembles de données il y aura toujours un résidu plus grand que les autres...

Finalement, quel est le verdict concernant la régression linéaire entre `fklength` et `age`? Elle viole la condition de linéarité, possiblement celle de normalité, remplit la condition d'homoscédasticité, et ne semble pas influencée outre mesure par des valeurs bizarres ou extrêmes.

Tests formels des conditions d'application pour la régression

Personnellement, je n'utilise jamais les tests formels des conditions d'application de la régression et me contente des graphiques des résidus pour guider mes décisions. C'est ce que la plupart des praticiens font. Cependant, lors de mes premières analyses, je n'étais pas toujours certain de bien interpréter les graphiques et j'aurais aimé un indice plus formel ou un test permettant de détecter les violations des conditions d'application de la régression.

Le package `lmtest`, qui ne fait pas partie de l'installation de base, mais qui est disponible sur CRAN, permet de faire plusieurs tests de linéarité et d'homoscédasticité. Et on peut tester la normalité avec le test Shapiro-Wilk test vu précédemment..

 Installez le package `lmtest` de CRAN.

 Exécutez les commandes suivantes:

```
library(lmtest)
bptest(RegModel.1)
dwtest(RegModel.1)
resettest(RegModel.1)
shapiro.test(residuals(RegModel.1))
```

pour obtenir

studentized Breusch-Pagan test **1**

```
data: RegModel.1
BP = 1.1765, df = 1, p-value = 0.2781
```

Durbin-Watson test **2**

```
data: RegModel.1
DW = 2.242, p-value = 0.849
alternative hypothesis: true autocorrelation is greater than 0
```

RESET test **3**

```
data: RegModel.1
RESET = 14.544, df1 = 2, df2 = 71, p-value = 5.082e-06
```

```
> shapiro.test(residuals(RegModel.1))
```

Shapiro-wilk normality test **4**

```
data: residuals(RegModel.1)
W = 0.9804, p-value = 0.2961
```

1 Le test Breusch-Pagan test examine si la variabilité des résidus est constantes lorsque les valeurs prédites changent. Une faible valeur de p suggère de l'hétéroscédasticité. Ici, la valeur p est élevée et suggère que la condition d'application d'homoscédasticité est remplie avec ces données.


2 Le test Durbin-Watson permet de détecter l'autocorrélation sérielle des résidus. En l'absence d'autocorrélation (i.e. d'indépendance des résidus) la valeur attendue de la statistique D est 2. Ce test permet d'éprouver l'hypothèse d'indépendance des résidus, mais ne permet de détecter qu'un type particulier de dépendance. Ici, le test ne permet pas de rejeter l'hypothèse d'indépendance.

3 Le test RESET permet d'éprouver la linéarité. Si la relation est linéaire, alors la statistique RESET sera d'environ 1. Ici, la statistique est beaucoup plus élevée (14.54) et hautement significative. Le test confirme la tendance que nous avons détectée visuellement plus haut: la relation n'est pas linéaire.

4 Le test de normalité Shapiro-Wilk sur les résidus confirme que la déviation par rapport à une distribution normale des résidus n'est pas grande.

Transformation des données en régression

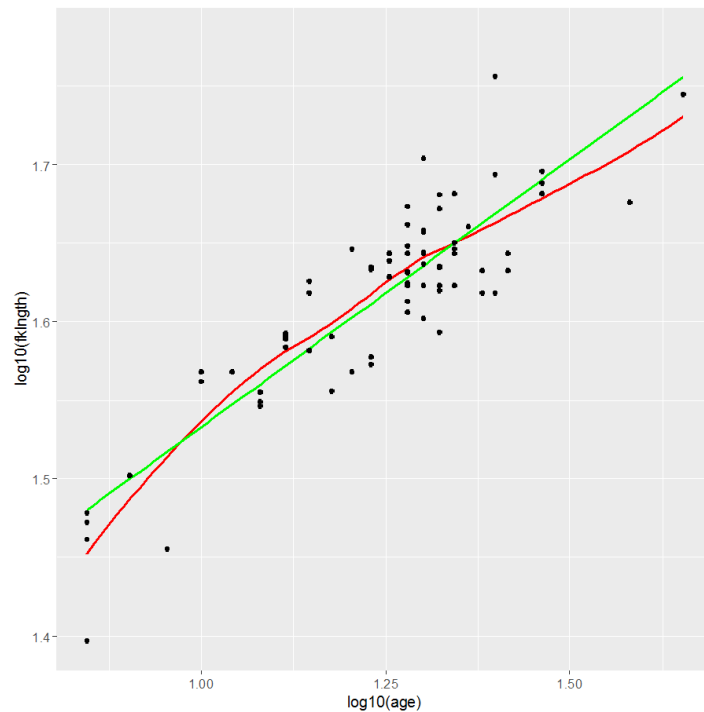
La relation entre `fklength` et `age` n'étant pas linéaire, on devrait donc essayer de transformer les données pour tenter de les linéariser. :

 Voyons ce qu'une transformation log donne:

```
ggplot(data=sturgeon.male,
       aes(x=log10(age), y=log10(fklength))) +
  geom_smooth(color="red") +
  geom_smooth(method="lm", se=FALSE, color="green") +
  geom_point()
```

On obtient:

Figure 6.



Ajustons maintenant une régression simple sur ces données transformées:

```
RegModel.2 <- lm(log10(fklength)~log10(age), data=sturgeon.male)
summary(RegModel.2)
```

```
Call:
lm(formula = log10(fklength) ~ log10(age), data = sturgeon.male)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0827935 -0.0168373 -0.0007188  0.0211016  0.0874465

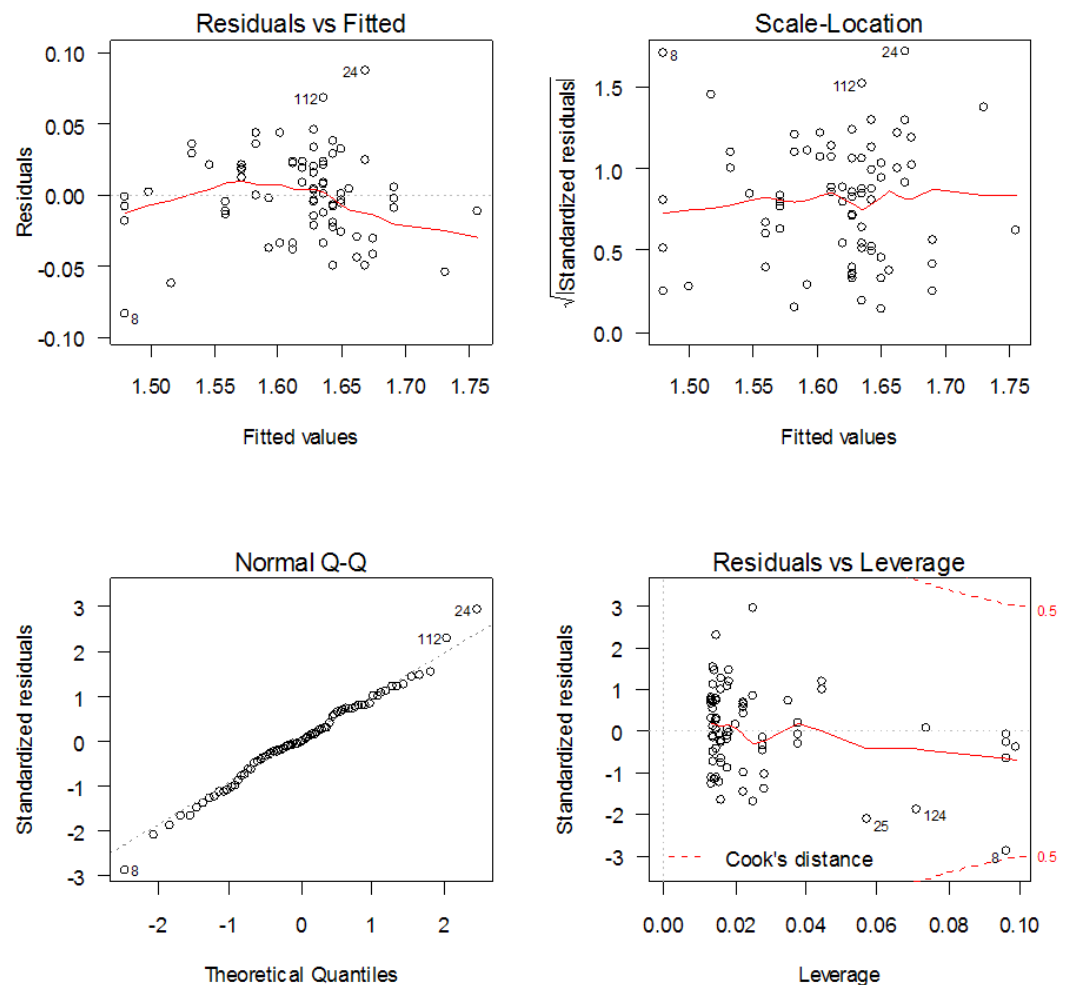
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.19199    0.02723   43.77  <2e-16 ***
log10(age)   0.34086    0.02168   15.72  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03015 on 73 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.772,    Adjusted R-squared:  0.7688
F-statistic: 247.1 on 1 and 73 DF,  p-value: < 2.2e-16
```

Examinons maintenant les graphiques diagnostiques:

```
plot(RegModel.2)
```

Figure 7.



Il y a une certaine amélioration, mais ce n'est pas encore parfait (la perfection n'est pas de ce monde....). Le graphique des résidus en fonction des valeurs prédites suggère encore une certaine non linéarité. Sur le graphique Q-Q les points se retrouvent plus près de la droite diagonale qu'avant, indiquant que les résidus sont encore plus près de la normalité après la transformation log-log. Il n'y a pas d'indice d'hétéroscédasticité. Finalement, même si il reste quelques points avec plus d'influence (leverage) que les autres, aucun n'a une distance de Cook au-delà de 0.5. En résumé, la transformation log a amélioré les choses: relation est plus linéaire, les résidus sont plus normaux, et il y a moins de points avec une influence relativement élevée. Est-ce que les tests formels supportent cette évaluation?

bptest (RegModel . 2)

```

dwtest(RegModel.2)
resettest(RegModel.2)
shapiro.test(residuals(RegModel.2))

> bptest(RegModel.2)

studentized Breusch-Pagan test

data:  RegModel.2
BP = 0.14282, df = 1, p-value = 0.7055

> dwtest(RegModel.2)

Durbin-Watson test

data:  RegModel.2
DW = 2.1777, p-value = 0.6134
alternative hypothesis: true autocorrelation is greater than 0

> resettest(RegModel.2)

RESET test

data:  RegModel.2
RESET = 4.4413, df1 = 2, df2 = 71, p-value = 0.01523

> shapiro.test(residuals(RegModel.2))

Shapiro-Wilk normality test

data:  residuals(RegModel.2)
W = 0.98998, p-value = 0.8246

```

Oui, les conclusions sont les mêmes: les résidus sont encore homoscedastiques (test Breusch-Pagan), ne sont pas autocorrélés (test Durbin-Watson), sont normaux (test Shapiro-Wilk), et sont plus linéaires (la valeur de P du test RESET est maintenant 0.015, au lieu de 0.000005). Donc la linéarité a augmenté, mais cette condition d'application semble encore légèrement violée.

Traitement des valeurs extrêmes

Dans cet exemple, il n'y a pas de valeur vraiment extrême. Oui, je sais, R a quand même identifié les observations 8, 24, et 112 dans le dernier graphique diagnostique. Mais ces valeurs sont encore dans la fourchette de valeurs que je juge "acceptables". Mais comment déterminer objectivement ce qui est acceptable? À quel moment juge t'on qu'une valeur extrême est vraiment trop invraisemblable pour ne pas l'exclure? Il n'y a malheureusement pas de règle absolue là-dessus. Les opinions varient, mais je penche vers le conservatisme sur cette question.

Ma position est que, à moins que la valeur soit impossible ou clairement une erreur d'entrée de données, je n'élimine pas les valeurs extrêmes et j'utilise toutes mes données dans leur analyse. Pourquoi?

Parce que je veux que mes données reflètent bien la variabilité naturelle ou réelle. C'est d'ailleurs parfois cette variabilité qui est intéressante.

L'approche conservatrice qui consiste à conserver toutes les valeurs extrêmes possibles est possiblement la plus honnête, mais elle peut causer certains problèmes. Ces valeurs extrêmes sont souvent la cause des violations des conditions d'application des tests statistiques. La solution suggérée à ce dilemme est de faire l'analyse avec et sans les valeurs extrêmes et de comparer les conclusions. Dans bien des cas, les conclusions seront qualitativement les mêmes et les tailles d'effet ne seront pas très différentes. Toutefois, dans certains cas, la présence des valeurs extrêmes change complètement les conclusions. Dans ces cas, il faut simplement accepter que les conclusions dépendent entièrement de la présence des valeurs extrêmes et sont donc peu concluantes.

Suivant cette approche comparative, refaisons donc l'analyse après avoir enlevé les observations 8, 24, et 112:

```
RegModel.3 <- lm(log10(fklnth)~log10(age), data=sturgeon.male, subset=!(rownames(sturgeon.male) %in% c('8', '24', '112'))))
summary(RegModel.3)
```

pour obtenir:

```
Call:
lm(formula = log10(fklnth) ~ log10(age), data = sturgeon.male,
    subset = !(rownames(sturgeon.male) %in% c("8", "24", "112")))
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.0691634 -0.0173903  0.0009862  0.0185900  0.0476466
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.22676    0.02431   50.46  <2e-16 ***
log10(age)   0.31219    0.01932   16.16  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.02554 on 70 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.7885,    Adjusted R-squared:  0.7855
F-statistic: 261 on 1 and 70 DF,  p-value: < 2.2e-16
```

L'ordonnée à l'origine (Intercept), la pente, et le R carré sont presque les mêmes, et la valeur de p est encore astronomiquement petite.

Enlever les valeurs extrêmes a peu d'effet dans ce cas.

Les graphiques diagnostiques des résidus et les tests formels des conditions d'application sur ce sous-ensemble de données donnent:

```
plot(RegModel.3)
library(lmtest)
```

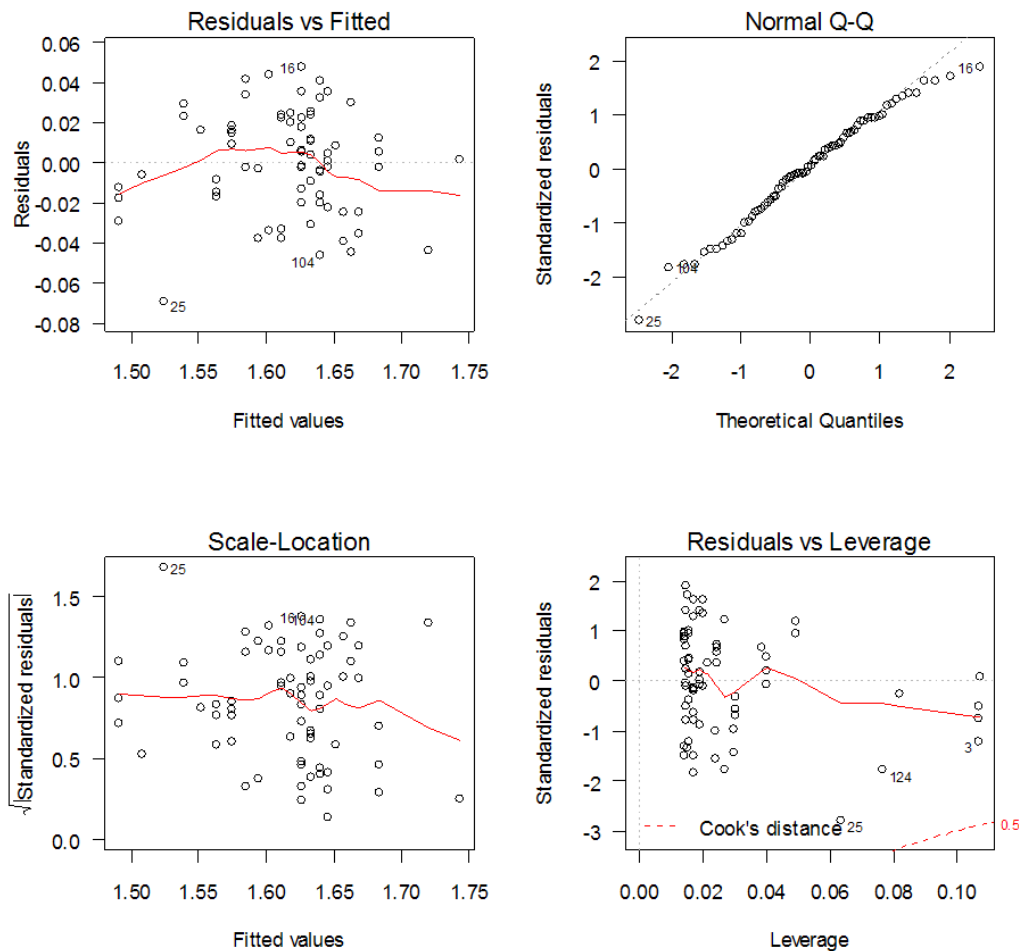


```

sturgeon.male.subset <- subset(sturgeon, subset=!(rownames(sturgeon.male) %in% c('8','24','112')))
bptest(RegModel.3)
dwtest(RegModel.3)
resettest(RegModel.3)
shapiro.test(residuals(RegModel.3))

```

Figure 8.



```

> bptest(RegModel.3)

studentized Breusch-Pagan test

data:  RegModel.3
BP = 0.3001, df = 1, p-value = 0.5838

> dwtest(RegModel.3)

Durbin-Watson test

data:  RegModel.3
DW = 2.0171, p-value = 0.5074
alternative hypothesis: true autocorrelation is greater than 0

> resettest(RegModel.3)

RESET test

```

```
data: RegModel.3
RESET = 3.407, df1 = 2, df2 = 68, p-value = 0.0389

> shapiro.test(residuals(RegModel.3))

      Shapiro-Wilk normality test

data: residuals(RegModel.3)
W = 0.98318, p-value = 0.4502
```

Il n'y a pas vraiment de différence ici non plus avec l'analyse des données en entier. Bref, tout pointe vers la conclusion que les valeurs les plus extrêmes de cet ensemble de donnée n'influencent pas indûment les résultats statistiques.

Quantifier la taille d'effet et analyse de puissance en régression

L'interprétation biologique des résultats n'est pas la même chose que l'interprétation statistique. Dans l'analyse qui précède, on conclue statistiquement que la taille augmente avec l'âge (puisque la pente est positive et $p < 0.05$). Mais cette augmentation "statistique" de la taille avec l'âge ne donne pas d'information sur la différence de taille entre les jeunes et vieux individus. La pente et un graphique sont plus informatifs à ce sujet que la valeur p . La pente (dans l'espace log-log) est 0.34. Cela veut dire que pour chaque unité d'accroissement de X ($\log_{10}(\text{age})$), il y a une augmentation de 0.34 unités of $\log_{10}(\text{fklngh})$. En d'autres mots, quand l'âge est multiplié par 10, la longueur à la fourche est multipliée environ par 2. Donc la longueur des esturgeons augmente plus lentement que leur âge (contrairement à mon tour de taille, semble-t-il...). La valeur de la pente (0.34) est un estimé de la taille de l'effet de l'âge sur la longueur.

Puissance de détecter une pente donnée

Pour les calculs de puissance avec G*Power vous devrez cependant utiliser une autre métrique de la taille de l'effet., calculée à partir de la pente, de son erreur-type, et de la taille de l'échantillon (ce qui facilite les calculs pour G*Power, mais malheureusement pas pour vous ;-). La métrique (d) est calculée comme:

$$d = \frac{b}{s_b \sqrt{n-k-1}}$$

où b est l'estimé de la pente, s_b est l'erreur type de la pente, n est le nombre d'observations, et k est le nombre de variables indépendantes (1 pour la régression linéaire simple).

Vous pouvez calculer approximativement la puissance avec G*Power pour une valeur de pente que vous jugez assez grande pour mériter d'être détectée. Allez à **Tests: Means: One group: difference from constant**, là, vous devrez remplacer la valeur de b dans l'équation pour la taille d'effet (d) par la pente que vous voudriez détecter, mais utiliser l'erreur type calculée à partir de vos données.

Par exemple, supposons que les ichthyologues considèrent qu'une pente de 0.1 pour la relation entre $\log_{10}(\text{fklength})$ et $\log_{10}(\text{age})$ est significative biologiquement, et qu'ils désirent estimer la puissance de détecter une telle pente à partir d'un échantillon de 20 esturgeons. Les résultats de la régression log-log nous fournissent ce dont on a besoin:

summary (RegModel.2)

```
Call:
lm(formula = log10(fklength) ~ log10(age), data = sturgeon.male)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0827935 -0.0168373 -0.0007188  0.0211016  0.0874465

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.19199    0.02723   43.77  <2e-16 ***
log10(age)   0.34086    0.02168   15.72  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03015 on 73 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.772,    Adjusted R-squared:  0.7688

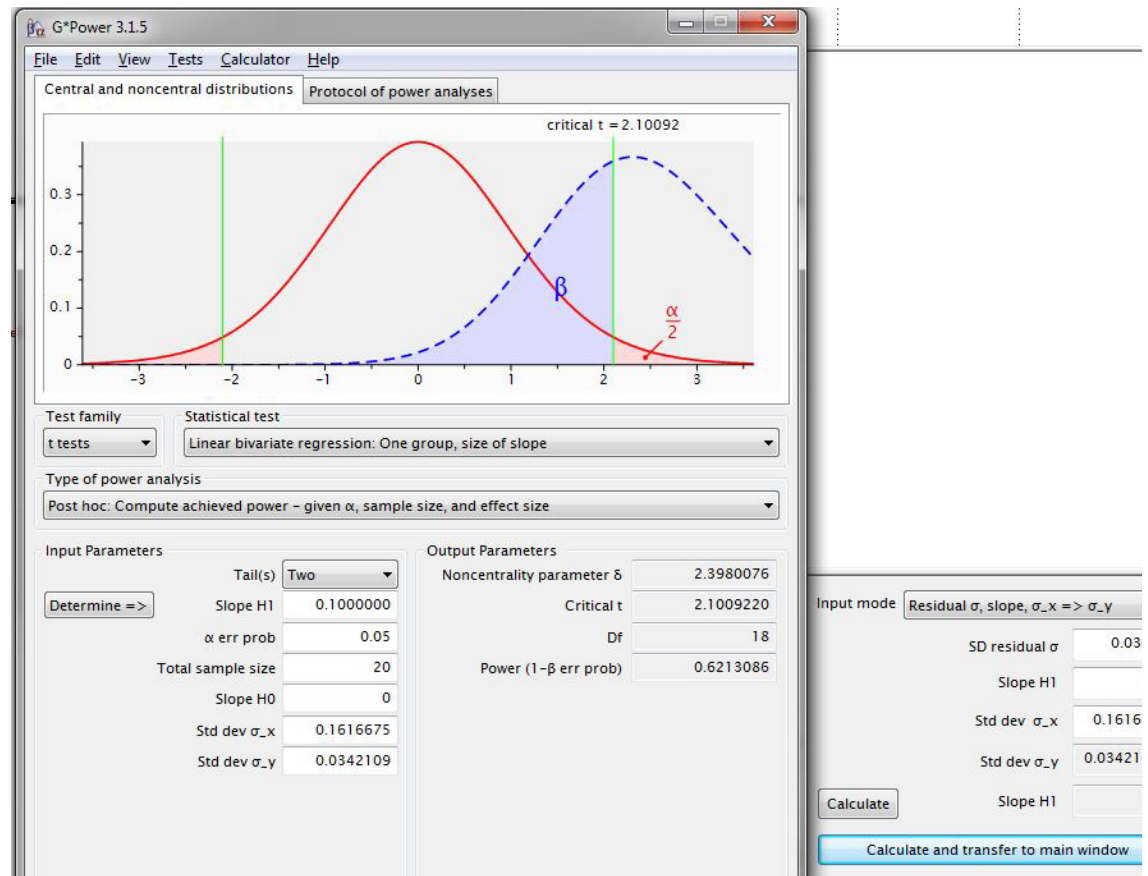
F-statistic: 247.1 on 1 and 73 DF, p-value: < 2.2e-16
```

L'erreur-type de la pente est 0.02168. Il y avait 75 poissons ($n=75$) dans l'échantillon de départ. On peut donc calculer la métrique de taille d'effet pour G*Power

$$d = \frac{b}{s_b \sqrt{n - k - 1}} = \frac{0.1}{0.02168 \sqrt{75 - 1 - 1}} = 0.54$$

Armés de cette taille d'effet (une pente présumée de 0.1 et une variabilité autour de la régression similaire à la régression de fklngth vs age), aller à **Tests: Means: One group: difference from constant**, et entrez la valeur calculée de d , α , et l'effectif de l'échantillon pour calculer la puissance

Figure 9.

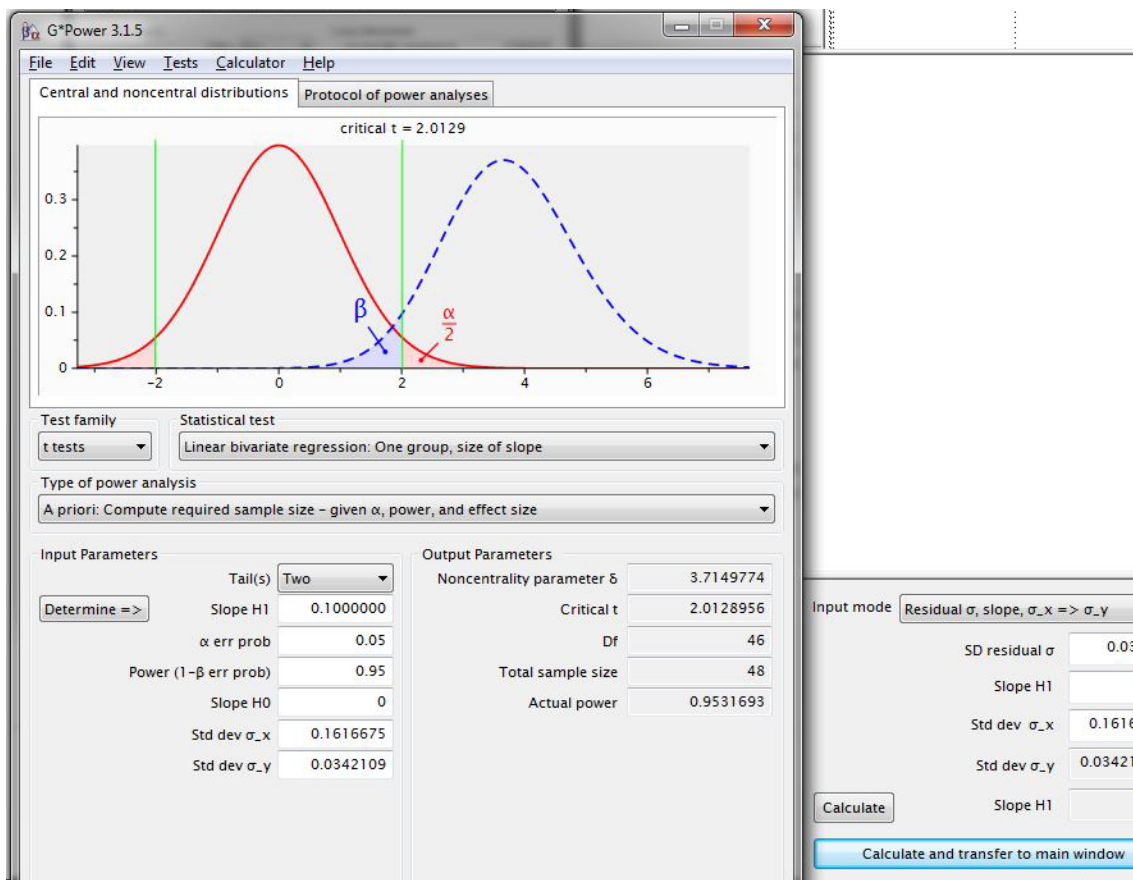


La puissance de détecter une pente comme étant statistiquement significative (au niveau α), si la pente est 0.1, que la variabilité résiduelle autour de la régression est semblable à celle de notre échantillon (ce qui revient à une taille d'effet de 0.54, pour un échantillon de 20 esturgeons et $\alpha=0.05$) est de 0.629. Seulement environ 2/3 des échantillons de cette taille détecteraient un effet significatif de l'âge sur fklngth.

Effectif requis pour atteindre une puissance désirée

Pour estimer la taille d'échantillon (effectif) requis pour avoir une puissance de 99% de détecter un effet de l'âge si la pente est 0.1 (sur une échelle log-log), avec $\alpha=0.05$, on utilise la même valeur de d (0.54):

Figure 10.



En augmentant la taille de l'échantillon à 65, selon le même scénario que précédemment, la puissance augmente à 99%.

Bootstrap en régression simple avec R

Un test non paramétrique pour l'ordonnée à l'origine et la pente d'une régression simple peut être effectué par bootstrap:

```
# Bootstrap 95% CI for regression coefficients
library(boot)
# function to obtain regression weights
bs <- function(formula, data, indices) {
  d <- data[indices,] # allows boot to select sample
  fit <- lm(formula, data=d)
  return(coef(fit))
}
# bootstrapping with 1000 replications
results <- boot(data=sturgeon.male, statistic=bs,
  R=1000, formula=log10(fklength)~log10(age))

# view results
```

results

```
> results
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

```
boot(data = sturgeon.male, statistic = bs, R = 1000, formula = log10(fklngh)
~
log10(age))
```

Bootstrap Statistics :

	original ①	bias ②	std. error ③
t1*	1.1919926	-8.936048e-05	0.03532614
t2*	0.3408557	1.116266e-04	0.02789170

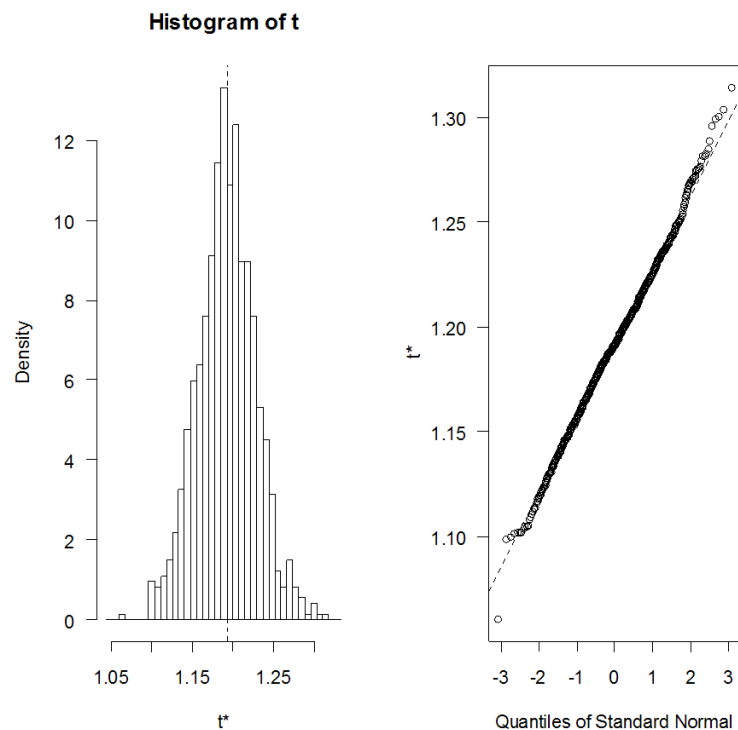
① Pour chaque paramètre du modèle (ici l'ordonnée à l'origine est appelée $t1^*$ et la pente de la régression $t2^*$), R imprime d'abord la valeur estimée sur tout l'échantillon

② "bias" est la différence entre la valeur moyenne des estimés bootstrap et la valeur originale sur tout l'échantillon

③ l'erreur-type de l'estimé bootstrap

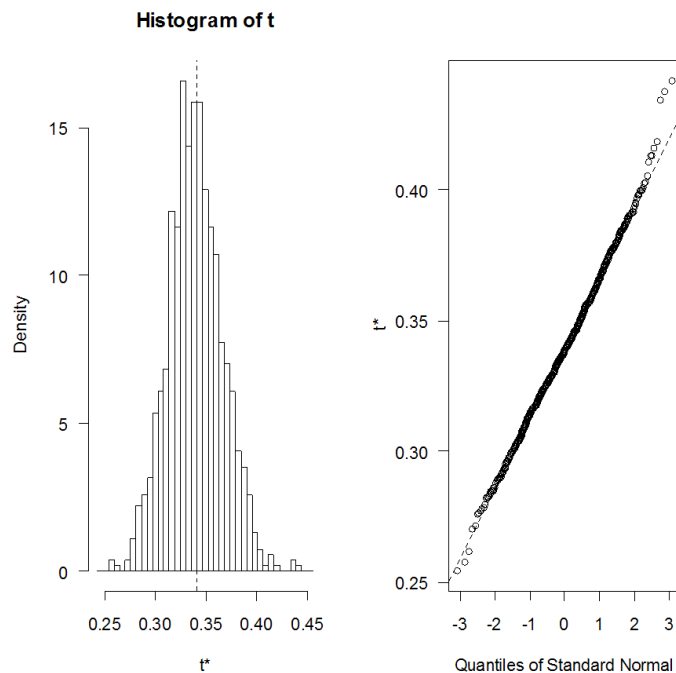
```
plot(results, index=1) # intercept
```

Figure 11.



```
plot(results, index=2) # log10(age)
```

Figure 12.



La distribution des estimés obtenus par bootstrap est assez normale dans cet exemple, avec de petites déviations dans les queue de la distribution (là où ça compte pour les intervalles de confiance...). On pourrait utiliser l'erreur-type des estimés bootstrap pour calculer un intervalle de confiance symétrique (moyenne \pm t ET). Cependant, comme R peut facilement calculer des intervalles de confiance qui corrigent pour le biais (BCa) ou encore des intervalles empiriques à partir des distributions simulées (méthode Percentile) il peut être aussi simple de les calculer selon les 3 méthodes:

```
# get 95% confidence intervals
boot.ci(results, type = "all", index = 1)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "all", index = 1)

Intervals :
Level      Normal          Basic
95%   ( 1.124, 1.256 )   ( 1.123, 1.255 )

Level      Percentile        BCa
95%   ( 1.129, 1.261 )   ( 1.104, 1.247 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
Warning message:
In boot.ci(results, type = "all", index = 1) :
  bootstrap variances needed for studentized intervals
```

```
boot.ci(results, type = "all", index = 2)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
```

Based on 1000 bootstrap replicates

```
CALL :
boot.ci(boot.out = results, type = "all", index = 2)

Intervals :
Level      Normal          Basic
95%   ( 0.2898, 0.3946 )   ( 0.2902, 0.3931 )

Level      Percentile      BCa
95%   ( 0.2886, 0.3915 )   ( 0.2966, 0.4044 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
Warning message:
In boot.ci(results, type = "all", index = 2) :
  bootstrap variances needed for studentized intervals
```

Ici, les 4 types d'intervalles de confiance que R a calculé sont essentiellement semblables. Si les données avaient violé plus sévèrement les conditions d'application de la régression (normalité, homoscedasticité), alors les différentes méthodes (Normal, Basic, Percentile, et BCa) auraient divergé un peu plus. Lequel choisir alors? BCa est celui qui est préféré de la majorité des praticiens, présentement.

Labo- Comparaison de deux échantillons

Après avoir complété cet exercice de laboratoire, vous devriez pouvoir:

- Utiliser R pour examiner visuellement vos données
- Utiliser R pour comparer les moyennes de deux échantillons tirés de populations normales
- Utiliser R pour comparer les moyennes de deux échantillons tirés de populations qui ne sont pas normales
- Utiliser R pour comparer les moyennes de deux échantillons appariés.

Examen visuel des données

Une des premières étapes dans toute analyse de données est l'examen visuel des données par des graphiques et statistiques sommaires pour détecter les distributions sous-jacentes, les valeurs extrêmes et les tendances dans vos données. Cela commence souvent avec des graphiques de vos données (histogrammes, diagrammes de probabilité, Box plots, etc.) qui vous permettent d'évaluer si vos données sont normales, si elles sont corrélées les unes aux autres, ou s'il y a des valeurs suspectes dans le fichier.

Supposons que l'on veuille comparer la distribution en taille des esturgeons de The Pas et Cumberland House. La variable `fklength` dans le fichier `sturgdat.SDD` représente la longueur (en cm) à la fourche de chaque poisson mesurée de l'extrémité de la tête à la base de la fourche de la nageoire caudale. Pour commencer, examinons si cette variable est normalement distribuée. On ne va pas tester pour la normalité à ce stade-ci; la présomption de normalité dans les analyses paramétriques s'applique aux résidus et non aux données brutes. Cependant, si les données brutes ne sont pas normales, vous avez d'habitude une très bonne raison de soupçonner que les résidus vont aussi ne pas avoir une distribution normale.

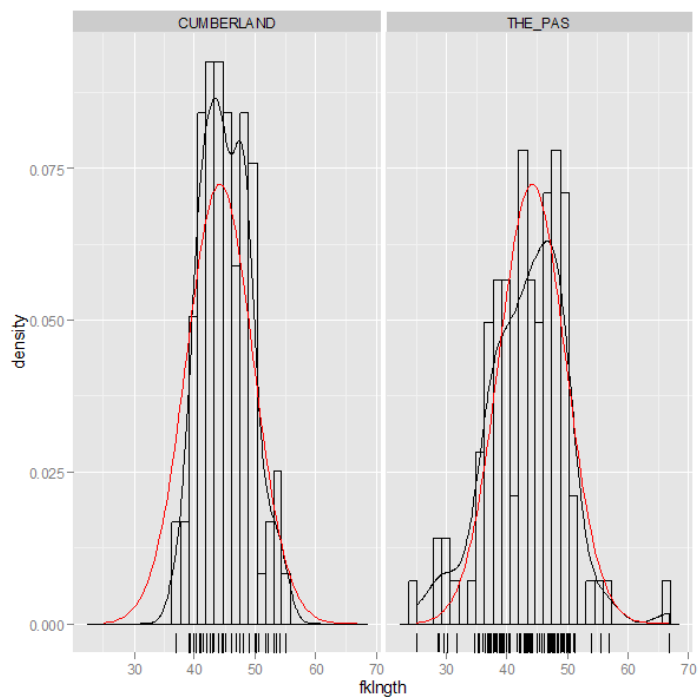
Une excellente façon de comparer visuellement une distribution à la distribution normale est de superposer un histogramme des données observées à une courbe normale. Pour ce faire, il faut procéder en deux étapes : 1) indiquer à R que nous voulons créer un histogramme superposé à une courbe normale, 2) spécifier qu'on veut que les graphiques soient faits pour les deux sites.

En utilisant les données du fichier `sturgeon.Rdata`, générez les pour et les distributions normales ajustées aux données de `fklength` à The Pas et Cumberland House.

```
require(ggplot2)

# use "sturgeon" dataframe to make plot called mygraph
# and define x axis as representing fklength
mygraph <- ggplot(sturgeon, aes(x = fklength, xlab="Fork
length (cm)"))
# add data to the mygraph ggplot
mygraph <- mygraph +
  # add data density smooth
  geom_density() +
  # add rug (bars at the bottom of the plot)
  geom_rug() +
  # add black semitransparent histogram
  geom_histogram(aes(y = ..density..),
    color = "black",
    alpha = 0.3) +
  # add normal curve in red, with mean and sd from fklength
  stat_function(fun = dnorm,
    args = list(
      mean = mean(sturgeon$fklength),
      sd = sd(sturgeon$fklength)
    ),
    color = "red")
#display graph, by location
mygraph+facet_grid(.~location)
```

Figure 1.



Examinez ce graphique et essayez de déterminer si ces deux échantillons sont normalement distribués.

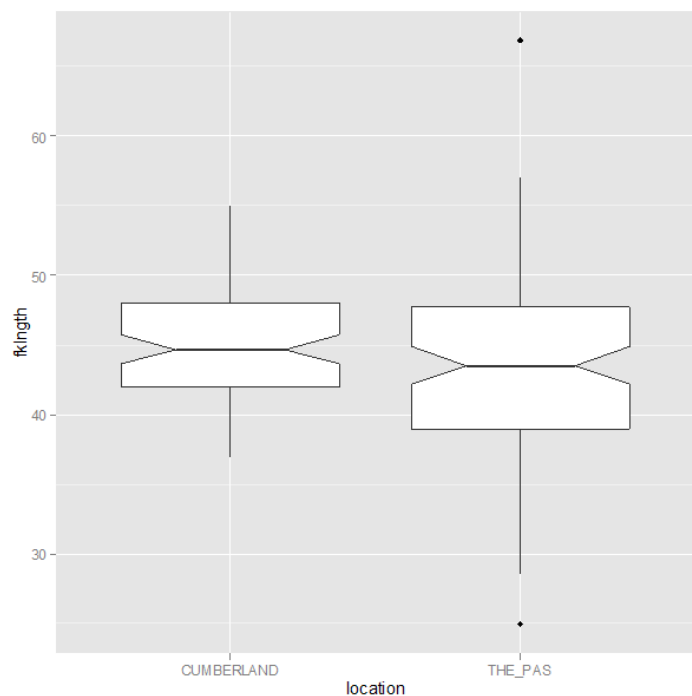
À mon avis, cette variable est approximativement normalement distribuée dans les deux échantillons.

Puisque ce qui nous intéresse est de comparer la taille des poissons de deux sites différents, c'est probablement une bonne idée de créer un graphique qui compare les deux groupes de données. Un Box plot convient très bien pour cette tâche.

 Tracez un boxplot de `fklength` groupé par `location`. Que concluez-vous quant à la différence entre les deux sites?

```
ggplot(data=sturgeon, aes(x=location,
y=fklength)) + geom_boxplot(notch=TRUE)
```

Figure 2.



Il n'y a pas de grande différence de taille entre les deux sites, mais la taille des poissons à The Pas est plus variable ayant une plus large étendue de taille et des valeurs extrêmes (définies par les valeurs qui sont $> 1.5 \times$ l'étendue interquartile) à chaque bout de la distribution.

Comparer les moyennes de deux échantillons indépendants : comparaisons paramétriques et non paramétriques

Éprouvez l'hypothèse nulle d'égalité de la longueur à la fourche à The Pas et Cumberland House. Que concluez-vous?

```
> # t-test assuming equal variances
> t.test(fklength~location, alternative='two.sided',
conf.level=.95,
+ var.equal=TRUE, data=sturgeon)

Two Sample t-test

data: fklength by location
t = 2.1359, df = 184, p-value = 0.03401
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1308307 3.2982615
sample estimates:
mean in group CUMBERLAND    mean in group THE_PAS
      45.08439              43.36984

>
> # t-test assuming unequal variances
> t.test(fklength~location, alternative='two.sided',
conf.level=.95,
+ var.equal=FALSE, data=sturgeon)

welch Two Sample t-test

data: fklength by location
t = 2.2201, df = 169.804, p-value = 0.02774
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1900117 3.2390804
sample estimates:
mean in group CUMBERLAND    mean in group THE_PAS
      45.08439              43.36984

>
> # non-parametric wilcoxon test
> wilcox.test(fklength ~ location, alternative="two.sided",
data=sturgeon)

Wilcoxon rank sum test with continuity correction

data: fklength by location
W = 4973, p-value = 0.06296
alternative hypothesis: true location shift is not equal to 0

>
```

En se fiant au test de t , on rejette donc l'hypothèse nulle. Il y a une différence significative (mais pas hautement significative) entre les deux moyennes des longueurs à la fourche.

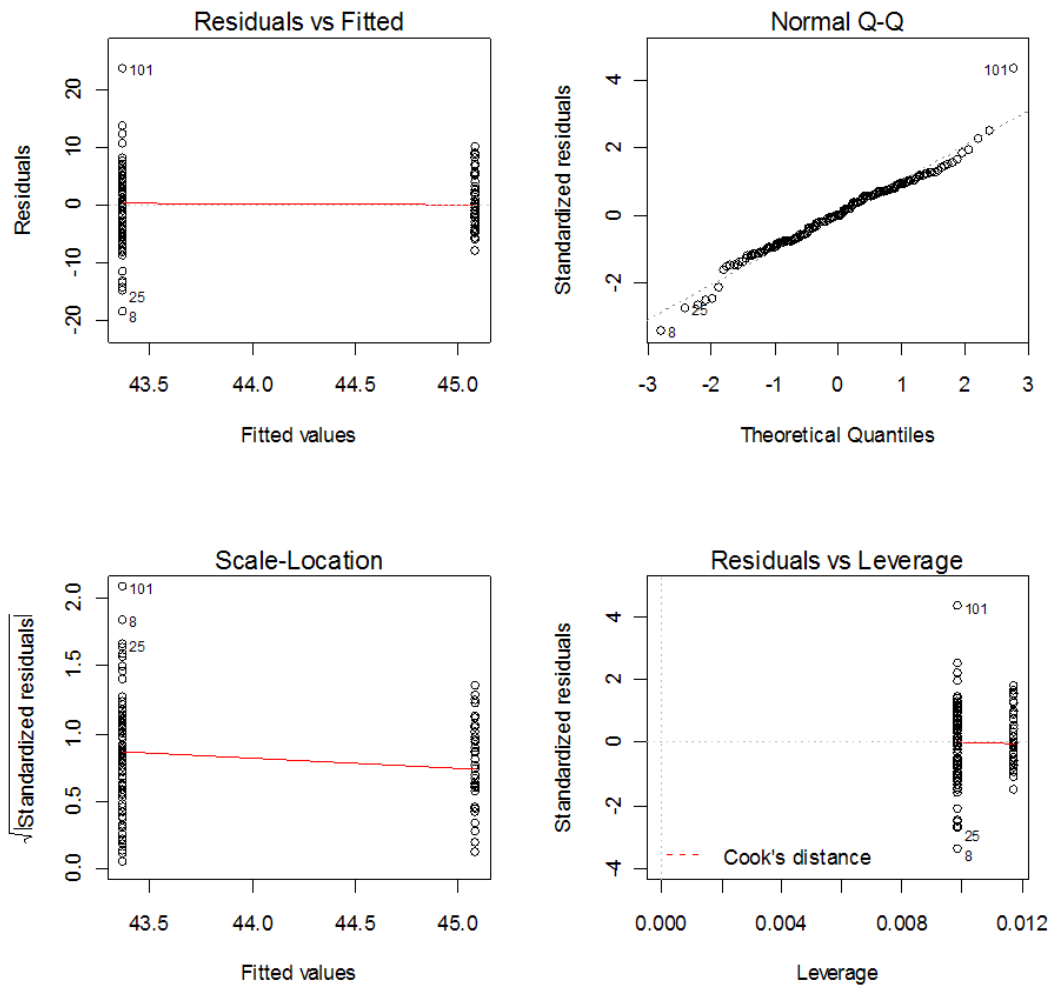
Notez que si l'on se fie au test de Wilcoxon, il faut accepter l'hypothèse nulle. Les deux tests mènent donc à des conclusions contradictoires. La différence significative obtenue par le test de t peut provenir en partie d'une violation des conditions d'application du test (normalité et homoscedasticité). D'un autre côté, l'absence de

différence significative selon le test de Wilcoxon pourrait être due au fait que, pour un effectif donné, la puissance du test non paramétrique est inférieure à celle du test paramétrique correspondant. Compte tenu 1) des valeurs de p obtenues pour les deux tests, et 2) le fait que pour des grands échantillons (des effectifs de 84 et 101 sont considérés grands) le test de t est considéré robuste, il est raisonnable de rejeter l'hypothèse nulle.

Avant d'accepter les résultats du test de t et de rejeter l'hypothèse nulle qu'il n'y a pas de différences de taille entre les deux sites, il est important de déterminer si les données remplissent les conditions de normalité des résidus et d'égalité des variances. L'examen préliminaire suggérait que les données sont à peu près normales mais qu'il y avait peut-être des problèmes avec les variances (puisque l'étendue des données pour The Pas était beaucoup plus grande que celle pour Cumberland). On peut examiner ces conditions d'application plus en détail en examinant les résidus d'un modèle linéaire et en utilisant les graphiques diagnostiques:

```
RAnovaModel.1 <- lm(fklngh ~ location, data=sturgeon)
par(mfrow = c(2, 2))
plot(RAnovaModel.1)
```

Figure 3.



.Le premier graphique ci-dessus montre comment les résidus se distribuent autour des valeurs prédites (les moyennes) pour chaque site et permette de juger si il semble y avoir un problème de normalité ou d'homoscédasticité. Si les variances étaient égales dans les deux sites, l'étendue verticale des résidus tendrait à être la même. Sur le graphique, on voit que l'étendue des résidus est plus grande à gauche (le site où la taille moyenne est la plus faible), ce qui suggère un possible problème d'homogénéité des variances. On peut tester cela plus formellement en comparant la moyenne de la valeur absolue des résidus.(on y reviendra; c'est le test de Levene).

Le second graphique est un graphique de probabilité (graphique Q-Q) des résidus. Comme ici, les points tombent près de la diagonale, il ne semble pas y avoir de problème important avec la normalité. On peut faire un test formel de la condition de normalité par le test de Shapiro-Wilk:

```
shapiro.test(residuals(AnovaModel.1))
```

```
Shapiro-Wilk normality test

data: residuals(AnovaModel.1)
W = 0.9747, p-value = 0.001858
```

Hummm. Ce test indique que les résidus ne sont pas normaux, ce qui contredit notre évaluation visuelle. Cependant, puisque (a) la distribution des résidus ne s'éloigne pas beaucoup de la normalité et (b) le nombre d'observations à chaque site est raisonnablement grand (i.e. >30), on n'a pas à être trop inquiet quant à l'impact de cette violation de normalité sur la fiabilité du test.

Qu'en est-il de l'égalité des variances?

```
> require(car)
> leveneTest(AnovaModel.1)
```

```
Levene's Test for Homogeneity of Variance
      Df F value    Pr(>F)
group  1 11.514 0.0008456 ***
      184
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> require(lmtest)
> bptest(AnovaModel.1)
```

```
studentized Breusch-Pagan test

data: AnovaModel.1
BP = 8.8015, df = 1, p-value = 0.00301

>
```

Les résultats qui précèdent proviennent de 3 des tests disponibles en R (dans les package car et lmtest) qui éprouvent l'hypothèse de l'égalité des variances dans des tests de t ou des modèles linéaires ayant uniquement des variables indépendantes discontinues ou catégoriques. **Il est redondant de faire les 2 tests.** Si ils sont présentés ici, c'est que ces 2 tests sont usuels et qu'il n'y a pas consensus quant au meilleur des deux. Le test de Levene est le plus connu et utilisé. Il compare la moyenne des valeurs absolues des résidus dans les deux groupes. Le test Breusch-Pagan a l'avantage d'être applicable à une plus large gamme de modèles linéaires (il peut être utilisé également avec des variables indépendantes continues, comme en régression). Ici, les deux tests mènent à la même conclusion: la variance diffère entre les deux sites.

Sur la base de ces résultats, on peut conclure qu'il y a évidence (mais faible) pour rejeter l'hypothèse nulle qu'il n'y a pas de différence dans la taille de poissons entre les deux sites. On a utilisé une modification du test de t pour tenir compte du fait que les variances ne sont pas égales et nous sommes satisfaits que la condition de normalité des résidus a été remplie. Alors, `fklength` à Cumberland est plus grande que `fklength` à The Pas.

Bootstrap et tests de permutation pour comparer deux moyennes

Le bootstrap et les tests de permutation peuvent être utilisés pour comparer les moyennes (ou d'autres statistiques). Le principe général est simple et peut être effectué de diverses façons. Ici j'utilise certains des outils disponibles et le fait qu'une comparaison de moyenne peut être représentée par un modèle linéaire. On pourra utiliser un programme similaire plus tard quand on ajustera des modèles plus complexes.

library(boot)

La première section sert à définir une fonction (ici appelée `bs`) qui extraie les coefficients d'un modèle ajusté

:

```
# function to obtain model coefficients for each iteration
bs <- function(formula, data, indices) {
  d <- data[indices, ]
  fit <- lm(formula, data = d)
  return(coef(fit))
}
```

La deuxième section avec la commande `boot()` fait le gros du travail: on prend les données dans `sturgeon`, on les bootstrap $R=1000$ fois, et chaque fois on ajuste le modèle `fklength` vs `location` et on garde les valeurs calculées par la fonction `bs`.

```
# bootstrapping with 1000 replications
results <- boot(data = sturgeon, statistic = bs, R = 1000,
  formula = fklength ~ location)

# view results
results
```

ORDINARY NONPARAMETRIC BOOTSTRAP

```
Call:
boot(data = sturgeon, statistic = bs, R = 1000, formula = fklength ~
  location)
```

```

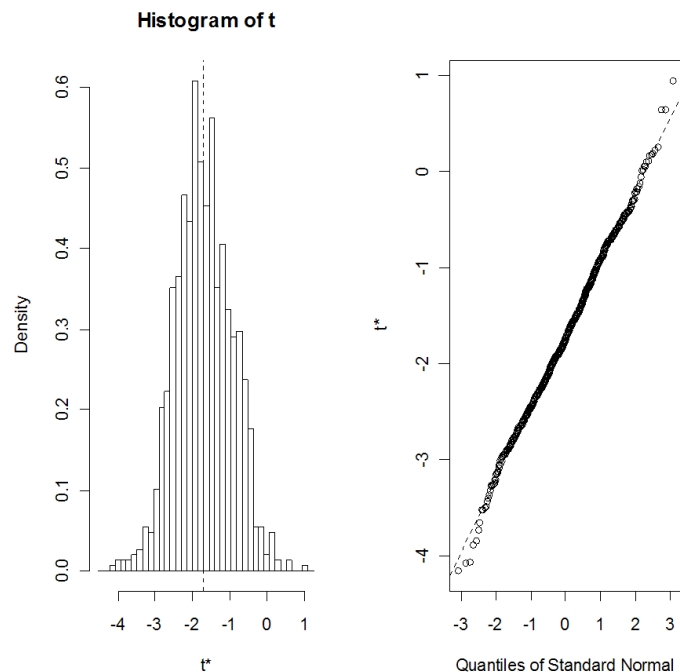
Bootstrap Statistics :
      original      bias      std. error
t1* 45.084391 -0.005634742  0.4164833
t2* -1.714546  0.012011519  0.7510652

```

On obtient les estimés originaux pour les deux coefficients du modèle: la moyenne pour le premier (alphabétiquement) site soit Cumberland, et la différence entre les deux moyennes à Cumberland et The Pas. C'est ce second paramètre, la différence entre les moyennes, qui nous intéresse.

```
plot(results, index = 2)
```

Figure 4.



```
# get 95% confidence intervals
```

```
boot.ci(results, type = "bca", index = 2)
```

```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

```

```

CALL :
boot.ci(boot.out = results, type = "bca", index = 2)

```

```

Intervals :
Level      BCa
95%      (-3.052, -0.211 )
Calculations and Intervals on Original scale

```

Comme l'intervalle de confiance n'inclue pas 0, on conclue que les moyennes ne sont pas les mêmes.

Les tests de permutation pour les modèles linéaires peuvent être effectués à l'aide du package `lmPerm`:

```
require(lmPerm2)

mymodelProb <- lmp(fklength ~ location, data = sturgeon,
  perm = "Prob")
```

La fonction `lmp()` fait tout le travail pour nous. Ici, cette fonction est effectuée avec l'option `perm` pour choisir la règle utilisée pour stopper les calculs. L'option `Probs` arrête les permutations quand la déviation standard estimée pour la p-valeur tombe sous un seuil déterminé. C'est l'une des nombreuses règles qui peuvent possiblement être utilisées pour ne faire les permutations que sur un sous-ensemble des permutations possibles (ce qui prendrait souvent trrrrrès longtemps, même sur votre super ordi).

```
summary(mymodelProb)
```

```
Call:
lmp(formula = fklength ~ location, data = sturgeon, perm = "Prob")

Residuals:
    Min       1Q   Median       3Q      Max
-18.40921  -3.75370  -0.08439   3.76598  23.48055

Coefficients:
              Estimate Iter 1 Pr(Prob) 2
location1    0.8573 1117   0.0824 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 5.454 on 184 degrees of freedom
Multiple R-Squared:  0.02419,    Adjusted R-squared:  0.01889
F-statistic: 4.562 on 1 and 184 DF,  p-value: 0.03401 3
```

1 Ici, la règle a limité à 1117 permutations le calcul. Notez que ce nombre va varier à chaque fois que vous tournerez cet petit bout de code. Ce sont des résultats obtenus par permutations **aléatoires**, donc vous devez vous attendre à de la variabilité. .

2 La p-valeur estimée pour H_0 est 0.0824. La différence observée pour `fklength` between entre les deux sites était plus grande que les valeurs permutées environ $(1-0.0824=\text{presque } 92\%)$ des 1117 permutations. Notez que 1117 permutations ce n'est pas un si grand nombre de permutations que ça, et donc les faibles valeurs de p ne sont pas très précises. Si vous voulez des valeurs précises de p , vous devrez faire plus de permutations.. Vous pouvez ajuster 2 paramètres: `maxIter`, le nombre maximal de permutations (défaut 5000) et `Ca`, le seuil de précision désiré qui arrête les permutations quand l'erreur-type de p est plus petite que $Ca \cdot p$ (défaut=0.1)

③ Le reste est la sortie standard pour un modèle ajusté à des données, avec le test paramétrique. Ici, la p-valeur, présument que toutes les conditions d'application sont remplies, est 0.34.

Comparer les moyennes de deux échantillons appariés

Dans certaines expériences les mêmes individus sont mesurés deux fois, par exemple avant et après un traitement ou encore à deux moments au cours de leur développement. Les mesures obtenues lors de ces deux événements ne sont pas indépendantes, et des comparaisons de ces mesures appariées doivent être faites.

Le fichier `skulldat.Rdata` contient des mesures de la partie inférieure du visage de jeunes filles d'Amérique du Nord prises à 5 ans, puis à 6 ans (données de Newman and Meredith, 1956)

✎ Pour débuter, éprouvons l'hypothèse que la largeur de la figure est la même à 5 ans et à 6 ans en assumant que les mesures viennent d'échantillons indépendants. .

```
> t.test(width~age, alternative='two.sided',
+ conf.level=.95, var.equal=TRUE, data=skulldat )
```

Two Sample t-test

```
data: width by age
t = -1.7812, df = 28, p-value = 0.08573
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.43000032  0.03000032
sample estimates:
mean in group 5 mean in group 6
 7.461333      7.661333
```

> Maintenant, effectuons le test apparié qui est approprié: Que conclure? Comment les résultats diffèrent-ils de la première analyse? Pourquoi?

```
> t.test(skulldat$width5, skulldat$width6, alternative='two.sided',
+ conf.level=.95, paired=TRUE)
```

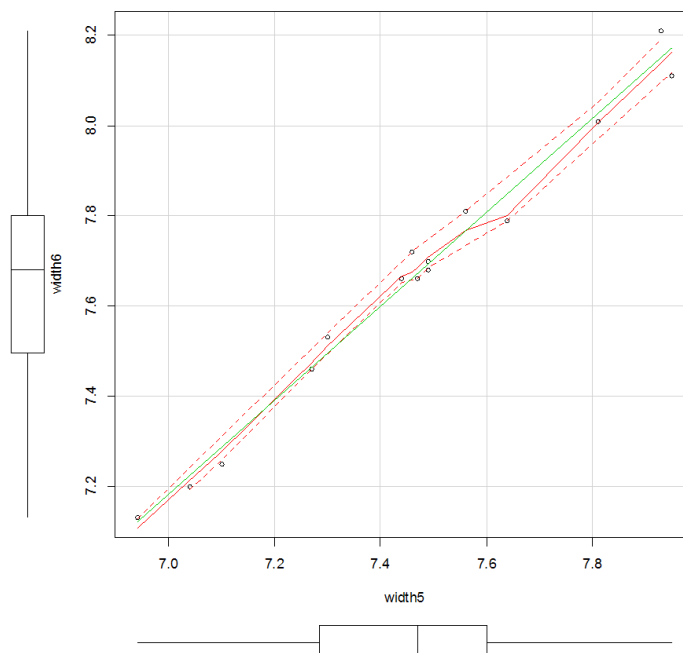
Paired t-test

```
data: skulldat$width5 and skulldat$width6
t = -19.5411, df = 14, p-value = 1.473e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2241710 -0.1798290
sample estimates:
mean of the differences
 -0.202
```

La première analyse a comme supposition que les deux échantillons de filles de 5 et 6 ans sont indépendants, alors que la deuxième analyse a comme supposition que la même fille a été mesurée deux fois, une fois à 5 ans, et la deuxième fois à 6 ans.

Notez que, dans le premier cas, on accepte l'hypothèse nulle, mais que le test apparié rejette l'hypothèse nulle. Donc, le test qui est approprié (le test apparié) indique un effet très significatif de l'âge, mais le test inapproprié suggère que l'âge n'importe pas. C'est parce qu'il y a une très forte corrélation entre la largeur du visage à 5 et 6 ans:

Figure 5.



avec $r = 0.993$. En présence d'une si forte corrélation, l'erreur-type de la différence appariée de largeur du visage entre 5 et 6 ans est beaucoup plus petit que l'erreur-type de la différence entre la largeur moyenne à 5 ans et la largeur moyenne à 6 ans. Par conséquent, la statistique t associée est beaucoup plus élevée pour le test apparié, la puissance du test est plus grande, et la valeur de p plus petite.

🔧 Répétez l'analyse en utilisant l'alternative nonparamétrique, le test Wilcoxon signed-rank. (Que concluez-vous?

```
> wilcox.test(skulldat$width5, skulldat$width6, alterna-
+   tive='two.sided',
+   paired=TRUE)
```

```

Warning in wilcox.test.default(skulldat$width5, skulldat$width6, alternative
= "two.sided", :
  cannot compute exact p-value with ties

      Wilcoxon signed rank test with continuity correction

data:  skulldat$width5 and skulldat$width6
V = 0, p-value = 0.0007211
alternative hypothesis: true location shift is not equal to 0
>

```

Donc on tire la même conclusion qu'avec le test de t apparié et conclue qu'il y a des différences significatives entre la taille des crânes de filles âgées de 5 et 6 ans (quelle surprise!).

Mais, attendez une minute! On a utilisé des tests bilatéraux ici mais, compte tenu de ses connaissances sur la croissance des enfants, une hypothèse unilatérale serait préférable. Ceci peut être accommodé en modifiant l'option "alternative". On utilise l'hypothèse alternative pour décider entre "less" ou "greater". Ici on s'attend que si il y a une différence, width5 va être inférieur à width6, donc on utiliserait "less".

```

> t.test(skulldat$width5, skulldat$width6, alterna-
tive='less',
+   conf.level=.95, paired=TRUE)

      Paired t-test

data:  skulldat$width5 and skulldat$width6
t = -19.5411, df = 14, p-value = 7.363e-12
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -0.1837930
sample estimates:
mean of the differences
 -0.202

>
> wilcox.test(skulldat$width5, skulldat$width6, alterna-
tive='less',
+   exact=TRUE, paired=TRUE)
Warning in wilcox.test.default(skulldat$width5, skulldat$width6, alternative
= "less", :
  cannot compute exact p-value with ties

      Wilcoxon signed rank test with continuity correction

data:  skulldat$width5 and skulldat$width6
V = 0, p-value = 0.0003606
alternative hypothesis: true location shift is less than 0

.

```

Références

- Bumpus, H.C. (1898) The elimination of the unfit as illustrated by the introduced sparrow, *Passer domesticus*. Biological Lectures, Woods Hole Biology Laboratory, Woods Hole, 11 th Lecture: 209 - 226.
- Newman, K.J. and H.V. Meredith. (1956) Individual growth in skeletal bigonial diameter during the childhood period from 5 to 11 years of age. Amer. J. Anat. 99: 157 - 187.

Labo- ANOVA à un critère de classification

Après avoir complété cet exercice de laboratoire, vous devriez être capable de :

- Utiliser R pour effectuer une analyse de variance paramétrique à un critère de classification, suivie de comparaisons multiples
- Utiliser R pour vérifier si les conditions d'application de l'ANOVA paramétrique sont remplies
- Utiliser R pour faire une ANOVA à un critère de classification non-paramétrique
- Utiliser R pour transformer des données de manière à mieux remplir les conditions d'application de l'ANOVA paramétrique.

ANOVA à un critère de classification et comparaisons multiples

L'ANOVA à un critère de classification est l'analogue du test de t pour des comparaisons de moyennes de plus de deux échantillons. Les conditions d'application du test sont essentiellement les mêmes, et lorsque appliqué à deux échantillons ce test est mathématiquement équivalent au test de t .

En 1961-1962, le barrage Grand Rapids était construit sur la rivière Saskatchewan en amont de Cumberland House. On croit que durant la construction plusieurs gros esturgeons restèrent prisonniers dans des sections peu profondes et moururent. Des inventaires de la population d'esturgeons furent faits en 1954, 1958, 1965 et 1966. Au cours de ces inventaires, la longueur à la fourche (`frklngth`) et la masse (`rndwght`) furent mesurées (pas nécessairement sur chaque poisson cependant). Ces données sont dans le fichier `dam10dat.SdD`.

Visualiser les données

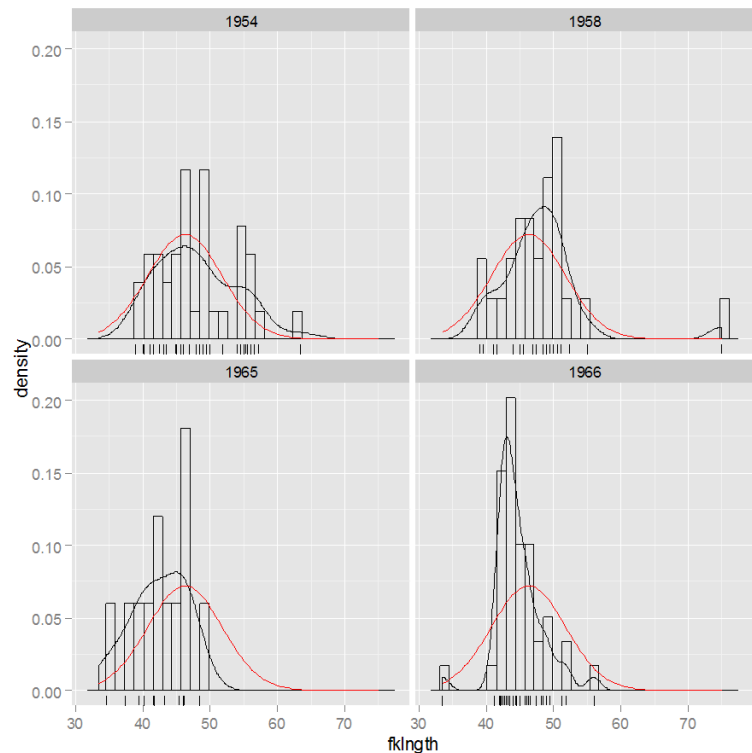
À partir des données du fichier `dam10dat.Rdata`, vous devez d'abord changer le type de donnée de la variable `year`, pour que R traite `YEAR` comme une variable discontinue (`factor`) plutôt que continue.

```
Dam10dat$year<-as.factor(Dam10dat$year)
ls.str()
attach(Dam10dat)
```


📊 Ensuite, visualisez les données comme dans le labo pour les tests de t .
 Créez un histogramme avec ligne de densité, un diagramme de probabilité, et un Box plot par année. Que vous révèlent ces données?

```
require(ggplot2)


# use "Dam10dat" dataframe to make plot called mygraph
# and define x axis as representing fklngth
mygraph <- ggplot(Dam10dat, aes(x = fklngth, xlab="Fork
length (cm)"))
# add data to the mygraph ggplot
mygraph <- mygraph +
  # add data density smooth
  geom_density() +
  # add rug (bars at the bottom of the plot)
  geom_rug() +
  # add black semitransparent histogram
  geom_histogram(aes(y = ..density..),
                 color = "black",
                 alpha = 0.3) +
  # add normal curve in red, with mean and sd from fklngth
  stat_function(fun = dnorm,
               args = list(
                 mean = mean(Dam10dat$fklngth),
                 sd = sd(Dam10dat$fklngth)
               ),
               color = "red")
#display graph, by year
mygraph+facet_wrap(~year,ncol=2)
```



Il semble que la taille des esturgeons est un peu plus petite après la construction du barrage, mais les données sont très variables et les effets ne sont pas parfaitement clairs. Il y a peut-être des problèmes de normalité avec les échantillons de 1954 et 1966, et il y a probablement des valeurs extrêmes dans les échantillons de 1958 et 1966. On va continuer en testant les conditions d'application de l'ANOVA. Il faut d'abord faire l'analyse et examiner les résidus.

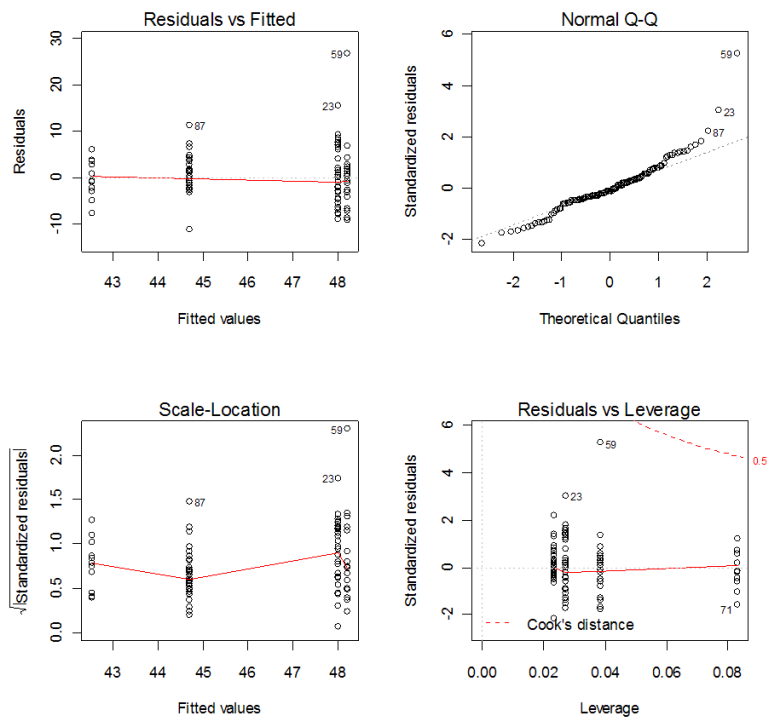
Vérifier si les conditions d'application de l'ANOVA paramétrique sont remplies

L'ANOVA paramétrique a trois conditions principales d'application : 1) les résidus sont normalement distribués, 2) la variance des résidus est égale dans tous les traitements (homoscédasticité) et 3) les résidus sont indépendants les uns des autres. Ces conditions doivent être remplies avant qu'on puisse se fier aux résultats de l'ANOVA paramétrique.

 Faites une ANOVA à un critère de classification sur `fklength` par année et produisez les graphiques diagnostiques


```
# Fit anova model and plot residual diagnostics
```

```
# but first, save current par and set graphic page to hold 4
graphs
opar <- par(mfrow = c(2, 2))
anova.modell <- lm(fklength ~ year, data=Dam10dat)
plot(anova.modell)
par(opar)
```



D'après les graphiques, on peut douter de la normalité et de l'homogénéité des variances. Notez qu'il y a un point qui ressort vraiment avec une forte valeur résiduelle (cas numéro 59) et qu'il ne s'aligne pas bien avec les autres valeurs: c'est la valeur extrême qui avait été détectée plus tôt. Ce point fera sans doute gonfler la variance résiduelle du groupe auquel il appartient.

Des tests formels nous confirmeront ou infirmeront nos conclusions faites à partir de ces graphiques.

 Faites un test de normalité sur les résidus de l'ANOVA.

```
shapiro.test(residuals(anova.modell))
```

```
Shapiro-Wilk normality test
```

```
data: residuals(anova.modell)
w = 0.9157, p-value = 1.63e-06
```

Ce test confirme nos soupçons: les résidus ne sont pas distribués normalement. Il faut cependant garder à l'esprit que la puissance est grande et que même de petites déviations de la normalité sont suffisantes pour rejeter l'hypothèse nulle.

Ensuite, éprouvez l'hypothèse d'égalité des variances (homoscedasticité):

```
require(car)
leveneTest(fklngh ~ year, data=Dam10dat)

Levene's Test for Homogeneity of Variance
  Df F value Pr(>F)
group 3  2.8159 0.04234 *
---
114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

La valeur de p vous dit que vous pouvez rejeter l'hypothèse nulle qu'il n'y a aucune différence dans les variances entre les années. Alors, nous concluons que les variances ne sont pas homogènes.

Faire l'ANOVA

Faites une ANOVA de `fklngh` en choisissant **en** présumant pour l'instant que les conditions d'application sont suffisamment remplies. Que concluez-vous?

```
> summary(anova.model1)

Call:
lm(formula = fklngh ~ year)

Residuals:
    Min       1Q   Median       3Q      Max
-11.2116  -2.6866  -0.7116   2.2103  26.7885

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  48.0243 1      0.8566  56.061 < 2e-16 ***
year1958      0.1872    1.3335   0.140  0.88859
year1965     -5.5077    1.7310  -3.182  0.00189 **
year1966     -3.3127    1.1684  -2.835  0.00542 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.211 2 on 114 degrees of freedom

Multiple R-squared: 0.1355, 3 Adjusted R-squared: 0.1128

F-statistic: 5.957 on 3 and 114 DF, p-value: 0.0008246 4
```

❶ Les 4 coefficients peuvent être utilisés pour obtenir les valeurs prédites par le modèle (i.e. les moyennes de chaque groupe). La `fklngh` moyenne de la première année (1954) est 48.0243. Les coefficients pour les 3 autres années sont la différence entre la moyenne de l'année en question et la moyenne de 1954. La moyenne pour 1965 est $48.0243 - 5.5077 = 42.5166$. Pour chaque coefficient, on a également accès à l'erreur-type, une valeur de `t` et la probabilité qui lui est associée (H_0 que le coefficient est 0). Les poissons étaient plus petits après la construction du barrage qu'en 1954. Vous devez prendre ces `p`-valeurs avec un grain de sel, car elles ne sont pas corrigées pour les comparaisons multiples et. En général, je porte peu d'attention à cette partie des résultats imprimés et me concentre sur ce qui suit.

❷ La racine carrée de la variance des résidus (valeurs observées moins valeurs prédites) qui correspond à la variabilité inexpliquée par le modèle (variation de la taille des poissons capturés la même année).

❸ Le `R`-carré est la proportion de la variabilité de la variable dépendante qui peut être expliquée par le modèle. Ici, le modèle explique 13.5% de la variabilité. Les différences de taille d'une année à l'autre sont relativement petites lorsqu'on les compare à la variation de taille entre les poissons capturés la même année.

❹ La `p`-valeur associée au test "omnibus" que toutes les moyennes sont égales. Ici, `p` est beaucoup plus petit que 0.05 et on rejeterait H_0 pour conclure que `fklngh` varie selon les années.

La commande `anova()` produit le tableau d'ANOVA standard qui contient la plupart de cette information:

```
> anova(anova.model1)
Analysis of Variance Table

Response: fklngh
          Df Sum Sq Mean Sq F value    Pr(>F)
year       3  485.26   161.75   5.9574 0.0008246 ***
Residuals 114 3095.30    27.15
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

La variabilité totale de `fklngh`, mesurée par la somme des carrés des écarts (`Sum sq`) est partitionnée en ce qui peut être expliqué par l'année (485.26) et la variabilité résiduelle inexpliquée (3095.30). L'année explique bien $(485.26 / (3095.30 + 485.26)) = 0.1355$ or 13.55% de la variabilité). Le carré moyen des résidus (`Residual Mean Sq`) est leur variance.

Les comparaisons multiples

La fonction `pairwise.t.test()` peut être utilisée pour comparer des moyennes et ajuster (ou non, si désiré) les probabilités pour le nombre de comparaisons en utilisant l'une des options pour `p.adj`:

```
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "none")
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "bonf")
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "holm")
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "fdr")
```

```
> pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "none")
```

Compare toutes les moyennes, sans ajuster les probabilités.

```
Pairwise comparisons using t tests with pooled SD

data:  Dam10dat$fklngh and Dam10dat$year

      1954  1958  1965
1958 0.8886 -    -
1965 0.0019 0.0022 -
1966 0.0054 0.0079 0.1996
```

Option "bonf" ajuste les p-valeurs avec la correction de Bonferroni. Ici, il y a 6 p-valeurs calculées, et la correction de Bonferroni revient à simplement multiplier la p-valeur par 6 (sauf si le résultat est supérieur à 1. Si tel est le cas, la p-valeur ajustée est 1)

```
> Pairwise comparisons using t tests with pooled SD

data:  Dam10dat$fklngh and Dam10dat$year

      1954  1958  1965
1958 1.000 -    -
1965 0.011 0.013 -
1966 0.033 0.047 1.000

P value adjustment method: bonferroni
```

Option "holm" est la correction séquentielle de Bonferroni dans laquelle les p-valeurs sont ordonnées de ($i=1$) la plus faible à (N) la plus grande. La correction pour les p-valeurs est $(N-i+1)$. Ici, il y a $N=6$ paires de moyennes qui sont comparées. La plus petite valeur de p non corrigée est 0.0019 pour 1954 vs 1965. La p-valeur corrigée est donc $0.0019 \times (6-1+1) = 0.011$. La seconde plus petite p-valeur est 0.0022. Sa p-valeur corrigée est $0.0022 \times (6-2+1) = 0.011$. Pour la p-valeur la plus élevée, la correction est $(N-N+1) = 1$, donc la p-valeur corrigée est égale à la p-valeur brute.

```
> pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "holm")
Pairwise comparisons using t tests with pooled SD
data: Dam10dat$fklngh and Dam10dat$year
      1954 1958 1965
1958 0.889 -    -
1965 0.011 0.011 -
1966 0.022 0.024 0.399
P value adjustment method:holm
```

L'option "fdr" sert à contrôler le "false discovery rate".

```
> pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "fdr")
Pairwise comparisons using t tests with pooled SD
data: Dam10dat$fklngh and Dam10dat$year
      1954 1958 1965
1958 0.8886 -    -
1965 0.0066 0.0066 -
1966 0.0108 0.0119 0.2395
```

Les quatre méthodes mènent ici à la même conclusion: les poissons sont plus gros après la construction du barrage et toutes les comparaisons entre les années 50 et 60 sont significatives alors que les différences entre 54 et 58 ou 65 et 66 ne le sont pas. La conclusion ne dépend pas du choix de méthode.

Dans d'autres situations, vous pourriez obtenir des résultats contradictoires. Alors, quelle méthode choisir? Les p-valeurs qui ne sont pas corrigées sont certainement suspectes lorsqu'il y a plusieurs comparaisons. D'un autre côté, la correction de Bonferroni est conservatrice et le devient encore plus lorsqu'il y a de très nombreuses comparaisons. Des travaux récents suggèrent que la correction fdr est un bon compromis lorsqu'il y a beaucoup de comparaisons.

La méthode de Tukey est l'une des plus populaires et est facile à utiliser en R (notez cependant qu'il y a une sale petit bogue qui se manifeste quand la variable indépendante peut ressembler à un nombre plutôt qu'un facteur, ce qui explique la petite pirouette avec paste dans mon code):

```
myyear <- as.factor(paste("", year))
TukeyHSD(aov(fklngh ~ myyear))

Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = fklngh ~ myyear)

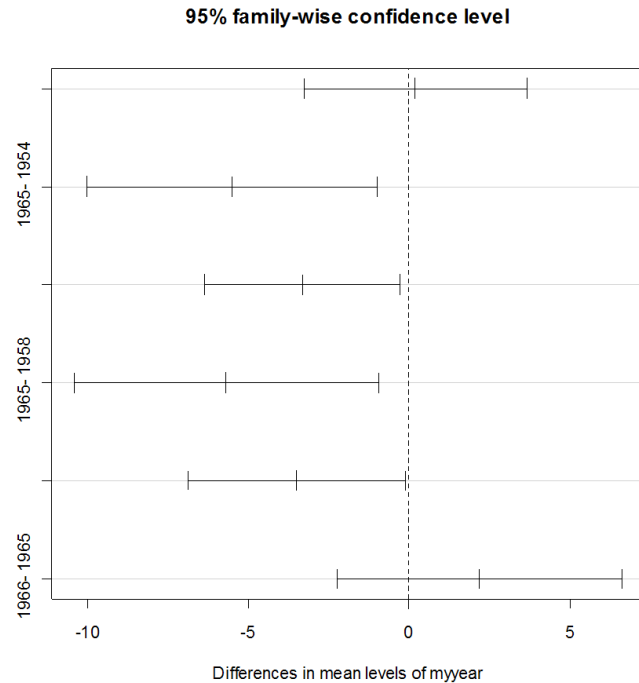
$myyear
      diff      lwr      upr    p adj
1958- 1954 0.1872141 -3.289570 3.6639986 0.9990071
1965- 1954 -5.5076577 -10.021034 -0.9942809 0.0100528
1966- 1954 -3.3126964 -6.359223 -0.2661701 0.0274077
```

```

1965- 1958 -5.6948718 -10.436304 -0.9534397 0.0116943
1966- 1958 -3.4999106 -6.875104 -0.1247171 0.0390011
1966- 1965  2.1949612 -2.240630  6.6305526 0.5710111

```

```
plot(TukeyHSD(aov(fklngh ~ myyear)))
```



Les intervalles de confiance, corrigés pour les comparaisons multiples par la méthode de Tukey, sont illustrés pour les différences entre années. Malheureusement les légendes ne sont pas complètes, mais l'ordre est le même que dans le tableau précédent.

Le package multcomp peut produire de meilleurs graphiques, mais requiert un peu plus de code:

```

# Alternative way to compute Tukey multiple comparisons
# set up a one-way ANOVA

anova.fkl.vs.year <- aov(aov(fklngh ~ year))

# set up all-pairs comparisons for factor `year`
library(multcomp)
meandiff <- glht(anova.fkl.vs.year, linfct = mcp(year =
"Tukey"))

confint(meandiff)

Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts

```



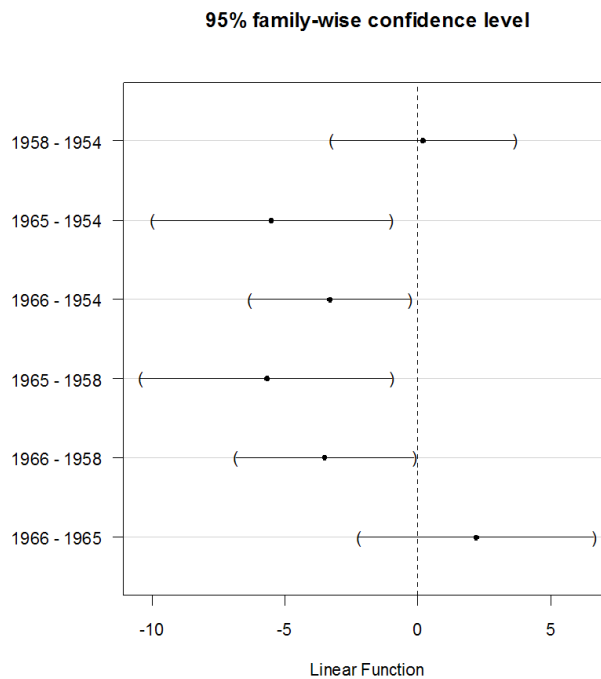
```
Fit: aov(formula = aov(fklength ~ year))
```

```
Estimated Quantile = 2.5936  
95% family-wise confidence level
```

```
Linear Hypotheses:
```

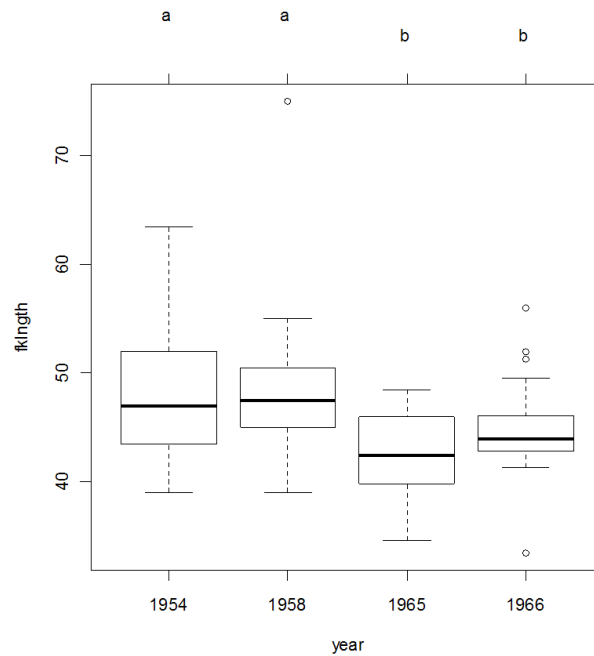
	Estimate	lwr	upr
1958 - 1954 == 0	0.1872	-3.2713	3.6457
1965 - 1954 == 0	-5.5077	-9.9973	-1.0180
1966 - 1954 == 0	-3.3127	-6.3432	-0.2822
1965 - 1958 == 0	-5.6949	-10.4114	-0.9783
1966 - 1958 == 0	-3.4999	-6.8574	-0.1424
1966 - 1965 == 0	2.1950	-2.2173	6.6073

```
old.par <- par(mai = c(1, 1.25, 1, 1))  
plot(meandiff)  
par(old.par)
```



C'est un peu mieux, mais ce qui le serait encore plus c'est un graphique des moyennes, avec leurs intervalles de confiance ajustés pour les comparaisons multiples:


```
# Compute and plot means and Tukey CI  
means <- glht(anova.fkl.vs.year, linfct = mcp(year =  
"Tukey"))  
cimeans <- cld(means)  
# use sufficiently large upper margin  
old.par <- par(mai = c(1, 1, 1.25, 1))  
# plot  
plot(cimeans)  
par(old.par)
```



Notez les lettres au dessus du graphique: les années étiquetées avec la même lettre ne diffèrent pas significativement l'une de l'autre.

Transformations de données et ANOVA non-paramétrique

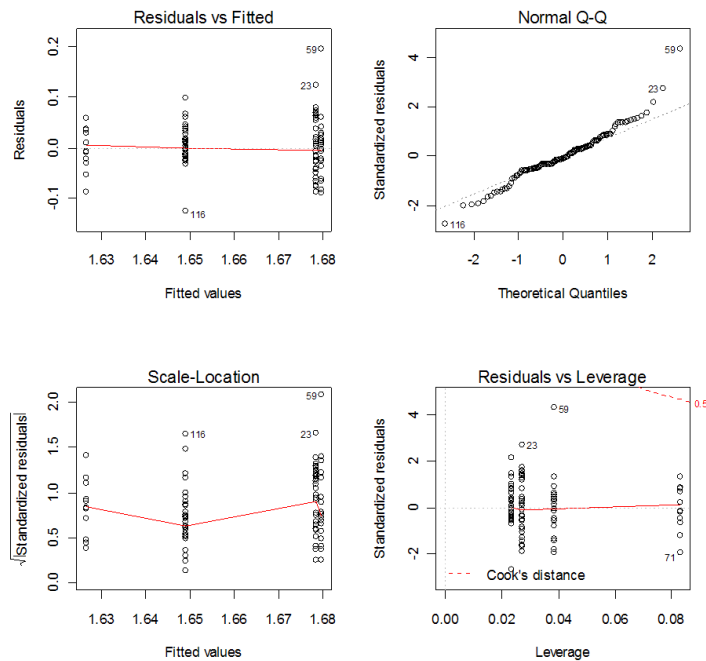
Dans l'exemple précédent sur les différences annuelles de la variable `fklngth`, on a noté que les conditions d'application de l'ANOVA n'étaient pas remplies. Si les données ne remplissent pas les conditions de l'ANOVA paramétrique, il y a 3 options : 1) Ne rien faire. Si les effectifs dans chaque groupe sont grands, on peut relaxer les conditions d'application car l'ANOVA est alors assez robuste aux violations de normalité (mais moins aux violations d'homoscédasticité), 2) on peut transformer les données, ou 3) on peut faire une analyse non-paramétrique.

 Refaites l'ANOVA de la section précédente après avoir transformé `fklngth` en faisant le logarithme à la base de 10. Avec les données transformées, est-ce que les problèmes qui avaient été identifiés disparaissent ?

```
# Fit anova model on log10 of fklngth and plot residual diagnostics
opar <- par(mfrow = c(2, 2))
```

```
anova.model2 <- lm(log10(fklngh) ~ year, data=Dam10dat)
plot(anova.model2)
par(opar)
```

Les graphiques diagnostiques des résidus donnent:



Les graphiques sont à peine mieux ici. Si on fait le test Wilk-Shapiro sur les résidus, on obtient:

```
data: residuals(anova.model2)
w = 0.962, p-value = 0.002048
Si on refait le test de Levene sur les valeurs absolues des résidus, on
obtient:
Levene's Test for Homogeneity of Variance
      Df F value Pr(>F)
group  3  2.6611 0.05148 .
      114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Alors, on a toujours des problèmes avec la normalité et on est juste sur le seuil de décision pour l'égalité des variances. Vous avez le choix à ce point: 1) essayer de trouver une autre transformation pour mieux rencontrer les conditions d'application, 2) assumer que les données sont rencontrant suffisamment les conditions d'application, ou 3) faire une ANOVA non-paramétrique.

 L'analogue non-paramétrique de l'ANOVA à un critère de classification le plus employé est le test de Kruskal-Wallis. Faites ce test (sur

`fklngh` et comparez les résultats à ceux de l'analyse paramétrique. Que concluez-vous?

```
kruskal.test(fklngh, year, data=Dam10dat)
```

```

Kruskal-Wallis rank sum test


data:  fklngh and year
Kruskal-Wallis chi-squared = 15.7309, df = 3, p-value = 0.001288

```

La conclusion est donc la même qu'avec l'ANOVA paramétrique: on rejette l'hypothèse nulle que le rang moyen est le même pour chaque année. Donc, même si les conditions d'application de l'analyse paramétrique n'étaient pas parfaitement rencontrées, les conclusions sont les mêmes, ce qui illustre la robustesse de l'ANOVA paramétrique.

Examen des valeurs extrêmes

Vous devriez avoir remarqué au cours des analyses précédentes qu'il y avait peut-être des valeurs extrêmes dans les données. Ces points étaient évidents dans le Box Plot de `fklngh` by `year` et ont été notés comme les points 59, 23, et 87 dans les diagrammes de probabilité des résidus et dans le diagramme de dispersion des résidus et des valeurs estimées. En général, vous devez avoir de très bonnes raisons pour enlever des valeurs extrêmes de la base de données (i.e. vous savez qu'il y a eu une erreur avec un cas). Cependant, il est quand même toujours valable de voir comment l'analyse change en enlevant des valeurs extrêmes de la base de données.

 Répétez l'ANOVA originale sur `fklngh` et `year` mais faites-le avec un sous-ensemble de données sans les valeurs extrêmes. Est-ce que les conclusions ont changé?

```

Damsubset<-Dam10dat[-c(23,59,87),] #removes obs 23, 59 and 87
aov.Damsubset <- aov(fklngh ~ as.factor(year), Damsubset)
summary(aov.Damsubset)

```

```

              Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(year)  3   367.51   122.50    6.8942 0.000267 ***
Residuals      111  1972.36    17.77
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
shapiro.test(residuals(aov.Damsubset))
```

```

Shapiro-Wilk normality test

data:  residuals(aov.Damsubset)
W = 0.9853, p-value = 0.2448

```

```

leveneTest(log10(fklength) ~ year, Damsubset)
Levene's Test for Homogeneity of Variance
      Df F value    Pr(>F)
group  3  3.8144 0.01206 *
      111
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

L'élimination de trois valeurs extrêmes améliore un peu les choses, mais ce n'est pas parfait. On a toujours un problème avec les variances, mais les résidus sont maintenant normaux. Cependant, le fait que la conclusion qu'on tire de l'ANOVA originale ne change pas en enlevant les points renforce le fait qu'on n'a pas une bonne raison pour enlever les points.

Commandes R pour refaire l'ANOVA sur le sous-ensemble de données

```

Damsubset<-Dam10dat[-c(23,59,87),] # removes obs 23, 59 and 87
aov.Damsubset <- aov(fklength ~ as.factor(year), Damsubset)
summary(aov.Damsubset)
shapiro.test(residuals(aov.Damsubset))
leveneTest(log10(fklength) ~ year, Damsubset)

```

Test de permutation

Commander R pour un test de permutation d'une ANOVA à un critère de classification.

```
#####
# Permutation Test for one-way ANOVA
# modified from code written by David C. Howell
# http://www.uvm.edu/~dhowell/StatPages/More_Stuff/Permutation%20Anova/PermTestsAnova.html

# set desired number of permutations
nreps <- 500
# to simplify reuse of this code, copy desired dataframe to mydata
mydata <- Dam10dat
# copy model formula to myformula
myformula <- as.formula("fklength ~ year")
# copy dependent variable vector to mydep
mydep <- mydata$fklength
# copy independent variable vector to myindep
myindep <- as.factor(mydata$year)

#####
# You should not need to modify code chunk below
#####
# Compute observed F value for original sample
mod1 <- lm(myformula, data=mydata) # Standard Anova
ANOVA <- summary(aov(mod1)) # Save summary to variable
observedF <- ANOVA[[1]]$"F value"[1] # Save observed F value

# Print standard ANOVA results
cat(" The standard ANOVA for these data follows ",
    "\n")
print(ANOVA, "\n")
cat("\n")
cat("\n")

print("Resampling as in Manly with unrestricted sampling of observations. ")
# Now start resampling

Fboot <- numeric(nreps) # initialize vector to receive permuted values
Fboot[1] <- observedF
for (i in 2:nreps) {
  newdependent <- sample(mydep, length(mydep)) # randomize dep var
  mod2 <- lm(newdependent ~ myindep) # refit model
  b <- summary(aov(mod2))
  Fboot[i] <- b[[1]]$"F value"[1] # store F stats
}
permprob <- length(Fboot[Fboot >= observedF])/nreps
cat(" The permutation probability value is: ", permprob,
    "\n")

# end of code chunk for permutation
```

Version lmPerm du test de permutation.

```
## lmPerm version of permutation test
require(lmPerm2)
# for generality, copy desired dataframe to mydata
# and model formula to myformula
mydata<-Dam10dat
myformula<-as.formula("fklength ~ year")

# Fit desired model on the desired dataframe
mymodel <- lm(myformula, data=mydata)
# Calculate permutation p-value
anova(lmp(myformula, data=mydata, perm = "Prob", center=FALSE,
Ca=0.001))
```

Labo- ANOVA à critères multiples : plans factoriels et hiérarchiques

Après avoir complété cet exercice de laboratoire, vous devriez être capable de:

- Utiliser R pour faire une ANOVA paramétrique d'un plan factoriel avec deux facteurs de classification et réplication
- Utiliser R pour faire une ANOVA paramétrique d'un plan factoriel avec deux facteurs de classification sans réplication
- Utiliser R pour faire une ANOVA paramétrique d'un plan hiérarchique avec réplication
- Utiliser R pour faire une ANOVA non paramétrique avec deux facteurs de classification
- Utiliser R pour faire des comparaisons multiples


Il existe une très grande variété de plans (designs) d'ANOVA que R peut analyser. Cet exercice n'est qu'une introduction aux plans les plus communs.

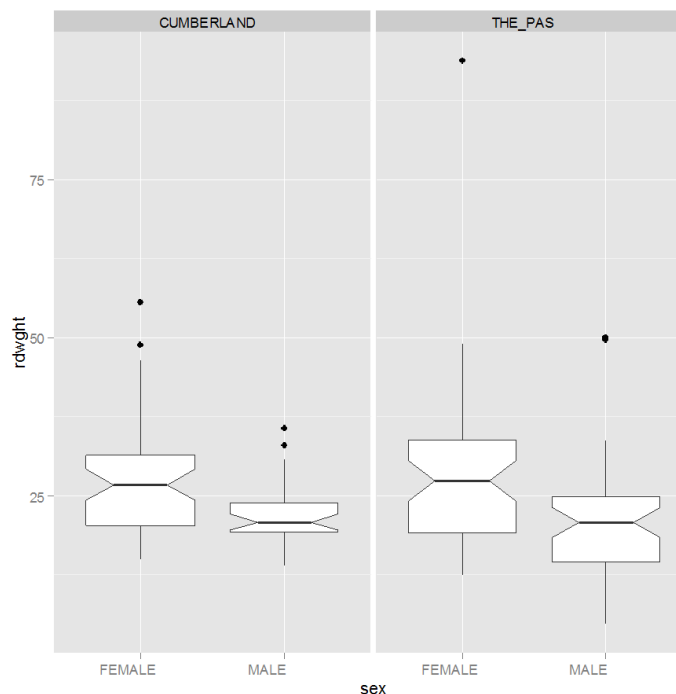
Plan factoriel à deux facteurs de classification et réplication

Il est fréquent de vouloir analyser l'effet de plusieurs facteurs simultanément. L'ANOVA factorielle à deux critères de classification permet d'examiner deux facteurs à la fois, mais la même approche peut être utilisée pour 3, 4 ou même 5 facteurs quoique l'interprétation des résultats devienne beaucoup plus complexe.


Supposons que vous êtes intéressés par l'effet de deux facteurs : site (location, Cumberland House ou The Pas) et sexe (sex, mâle ou femelle) sur la taille des esturgeons. Comme l'effectif n'est pas le même pour tous les groupes, c'est un plan qui n'est pas équilibré. Notez aussi qu'il y a des valeurs manquantes pour certaines variables, ce qui veut dire que chaque mesure n'a pas été effectuée sur chaque poisson.

ANOVA à effets fixes (Modèle I)

 Examinez d'abord les données en faisant des box plots de `rdwght` pour `sex` et `location` des données du fichier `Stu2wdat.Rdata`.



Les graphiques montrent qu'aux deux sites les femelles sont probablement plus grandes que les mâles, mais que les tailles ne varient pas beaucoup d'un site à l'autre. La présence de valeurs extrêmes sur ces graphiques suggère qu'il y aura peut-être des problèmes avec la condition de normalité des résidus.

 Générez les statistiques sommaires pour RDWGHT par sex et Location.

```
aggregate(rdwght~sex+location, data=Stu2wdat, FUN = "summary")
sex      location rdwght.Min. rdwght.1st Qu. rdwght.Median rdw-
ght.Mean rdwght.3rd Qu. rdwght.Max.
1 FEMALE  CUMBERLAND      15.10      20.40      26.80      27.37
31.40      55.60
2 MALE    CUMBERLAND      14.00      19.22      20.85      22.14
23.90      35.60
3 FEMALE  THE_PAS         12.54      19.14      27.39      27.98
33.88      93.72
4 MALE    THE_PAS          4.73      14.63      20.79      20.65
24.94      49.94
```

Ces résultats supportent l'interprétation des box plots: Les femelles sont plus grosses que les mâles, et la différence de taille entre les deux sites sont petites.

 À l'aide du fichier Stu2wdat.Rdata faites une ANOVA factorielle à deux critères de classification:

```
# Fit anova model and plot residual diagnostics
```

```
# but first, save current par and set graphic page to hold 4
graphs
opar <- par(mfrow = c(2, 2))

anova.modell <- lm(rdwght ~ sex + location + sex:location,
  contrasts = list(sex = contr.sum, location = contr.sum),
  data = Stu2wdat)
anova(anova.modell)
```

Analysis of Variance Table

Response: rdwght

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sex	1	1839.6	1839.6	18.6785	2.569e-05 ***
location	1	4.3	4.3	0.0433	0.8355
sex:location	1	48.7	48.7	0.4944	0.4829
Residuals	178	17530.4	98.5		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Attrappe. Attention, R imprime les sommes des carrés séquentielles (Type I) les carrés moyens et probabilités associés. Vous ne pouvez pas vous y fier si votre plan d'expérience n'est pas parfaitement balancé. Dans cet exemple, le nombre de poissons capturés change selon le site et le sexe et le plan d'expérience n'est donc pas balancé.

Vous devez extraire les sommes de carrés partielles (Type III). Le moyen le plus simple que j'ai trouvé est d'utiliser la fonction `Anova()` du package `car` (notez la différence subtile, `Anova()` n'est pas la même chose que `anova()`, R est impitoyable et distingue les majuscules des minuscules). Malheureusement, `Anova()` ne suffit pas; il faut également utiliser la commande `contrasts = list(sex = contr.sum, location = contr.sum) ...`

```
require(car)
```

```
Anova(anova.modell, type = 3)
```

Anova Table (Type III tests)

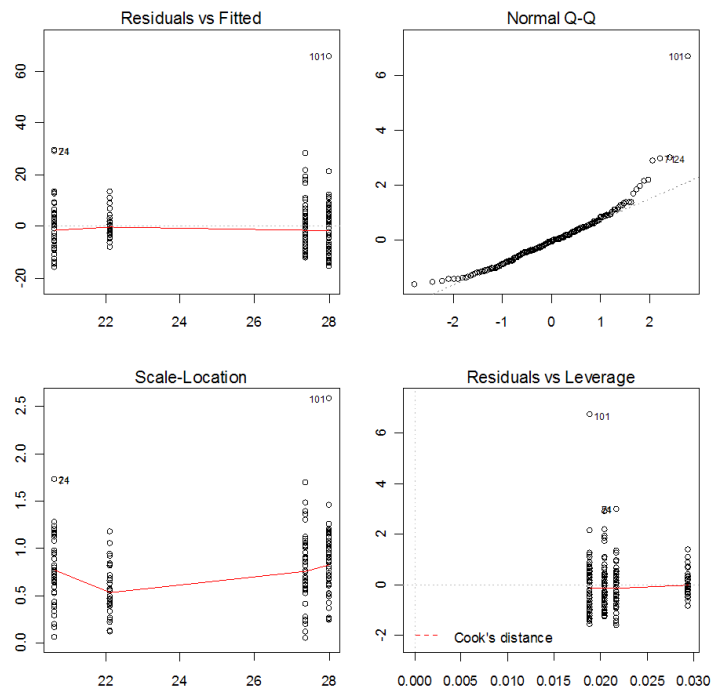
Response: rdwght

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	106507	1	1081.4552	< 2.2e-16 ***
sex	1745	1	17.7220	4.051e-05 ***
location	9	1	0.0891	0.7656
sex:location	49	1	0.4944	0.4829
Residuals	17530	178		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Suite à l'ANOVA, on accepte deux hypothèses nulles: (1) que l'effet du sexe ne varie pas entre les sites (pas d'interaction significative) et (2) qu'il n'y a pas de différence de taille des esturgeons (peu importe le sexe) entre les deux sites. D'un autre côté, on rejette l'hypothèse nulle qu'il n'y a pas de différence de taille entre les esturgeons mâles et les femelles, tel que suggéré par les graphiques.

```
plot(anova.model1)
```



Cependant, on ne peut se fier à ces résultats sans vérifier si les conditions d'application de l'ANOVA étaient remplies. Un examen des graphiques des résidus, en haut, montre que les résidus semblent être distribués plus ou moins normalement, si ce n'est des 3 valeurs extrêmes qui sont notées sur le diagramme de probabilité (cas 101, 24, & 71). D'après le graphique des résidus vs les valeurs prédites, on voit que l'étendue des résidus est plus ou moins égale pour les valeurs estimées, sauf encore pour 2 ou 3 cas. Si on éprouve la normalité, on obtient:

```
shapiro.test(residuals(anova.model1))
```

Shapiro-Wilk normality test


```
data: residuals(anova.model1)
W = 0.8721, p-value = 2.619e-11
```

Alors, il y a évidence que les résidus ne sont pas distribués normalement.

Nous allons utiliser le test de Levene pour examiner l'homoscédasticité des résidus, de la même façon qu'on a fait pour l'ANOVA à un critère de classification.

```
require(car)
leveneTest(rdwght ~ sex * location, data = Stu2wdat)
Levene's Test for Homogeneity of Variance
      Df F value Pr(>F)
group  3  3.8526 0.01055 *
      178
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Si les résidus étaient homoscédastiques, on accepterait l'hypothèse nulle que le absres moyen ne varie pas entre les niveaux de sexe et location (i.e., sexloc). Le tableau d'ANOVA ci-dessus montre que l'hypothèse est rejetée. Il y a donc évidence d'hétéroscédasticité. En bref, nous avons donc plusieurs conditions d'application qui ne sont pas respectées. La question qui reste est: ces violations sont-elles suffisantes pour invalider nos conclusions ?


 Répétez la même analyse avec les données du fichier `stu2mdat.Rdata`. Que concluez-vous? Supposons que vous vouliez comparer la taille des mâles et des femelles. Comment cette comparaison diffère entre les deux ensembles de données ?

Notez que cette fois les femelles sont plus grandes que les mâles à Cumberland House, mais que c'est le contraire à The Pas. Quel est le résultat de l'ANOVA (n'oubliez pas qu'il faut des Type III sums of squares pour les résultats)?

Anova Table (Type III tests)

```
Response: rdwght
      Sum Sq Df F value    Pr(>F)
(Intercept) 106507 1 1081.4552 < 2.2e-16 ***
sex           49  1  0.4944    0.4829
location      9  1  0.0891    0.7656
sex:location 1745  1 17.7220 4.051e-05 ***
Residuals   17530 178
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dans ce cas, le terme de l'interaction (sex:location) est maintenant significatif mais les effets principaux ne le sont pas.

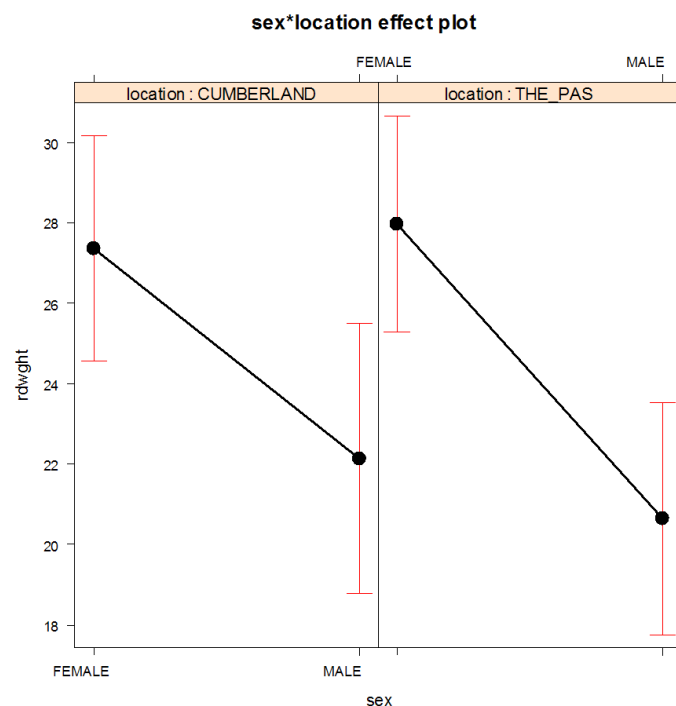
 Vous trouverez utile ici de créer des graphiques pour les deux fichiers de données pour comparer les interactions entre sex et location. Le graphique d'interaction montre les relations entre les moyennes de

chaque combinaison de facteurs (appelées aussi les moyennes des cellules). Générez un graphique illustrant les interactions en utilisant la fonction `allEffects` du package `effects` :

```
library(effects)
allEffects(anova.model1)
model: rdwght ~ sex + location + sex:location

sex*location effect
location
sex      CUMBERLAND  THE_PAS
FEMALE   27.37347    27.97717
MALE     22.14118    20.64652

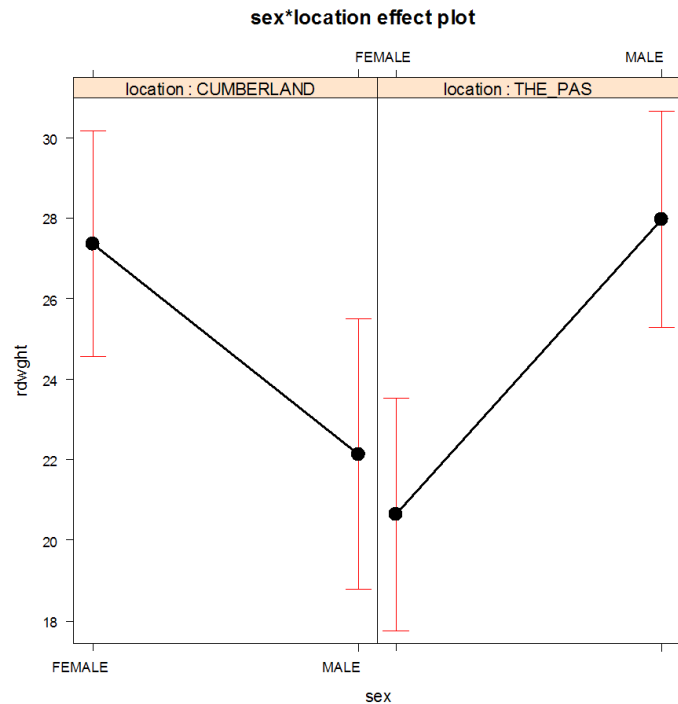
plot(allEffects(anova.model1), 'sex:location')
```



```
allEffects(anova.model2)
model: rdwght ~ sex + location + sex:location

sex*location effect
location
sex      CUMBERLAND  THE_PAS
FEMALE   27.37347    20.64652
MALE     22.14118    27.97717

plot(allEffects(anova.model2), 'sex:location')
```



Il y a une différence importante entre les résultats obtenus avec `stu2wdat` et `stu2mdat`. Dans le premier cas, puisqu'il n'y a pas d'interaction, on peut regrouper les données des deux niveaux d'un facteur (le site, par exemple) pour éprouver l'hypothèse d'un effet de l'autre facteur (le sexe). En fait, si on fait cela et que l'on calcule une ANOVA à un critère de classification (sex), on obtient:

```
Anova(aov(rdwght ~ sex, data = Stu2wdat), type = 3)
Anova Table (Type III tests)
```

```
Response: rdwght
          Sum Sq Df F value    Pr(>F)
(Intercept) 78191  1 800.440 < 2.2e-16 ***
sex          1840  1  18.831 2.377e-05 ***
Residuals   17583 180
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notez que la somme des carrés des résidus (17583) est presque égale à celle du modèle complet (17530) de l'ANOVA factorielle à deux facteurs. C'est parce que dans cette anova factorielle, le terme d'interaction et le terme représentant l'effet du site n'expliquent qu'une partie infime de la variabilité.

D'un autre côté, si on essaie le même truc avec `stu2mdat`, on obtient:

```
Anova(aov(rdwght ~ sex, data = Stu2mdat), type = 3)
Anova Table (Type III tests)
```

```

Response: rdwght
      Sum Sq Df F value Pr(>F)
(Intercept) 55251 1 515.0435 <2e-16 ***
sex          113 1  1.0571 0.3053
Residuals   19309 180
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Ici la somme des carrés des résidus (19309) est beaucoup plus grande que celle de l'ANOVA factorielle (175306) parce qu'une partie importante de la variabilité expliquée par le modèle est associée à l'interaction. Notez que si on n'avait fait que cette analyse, on conclurait que les esturgeons mâles et femelles ont la même taille. Mais en fait leur taille diffère; seulement la différence est à l'avantage des mâles à un site et à l'avantage des femelles à l'autre. Il est donc délicat d'interpréter l'effet principal (sexe) en présence d'une interaction significative...

ANOVA à effets mixtes (Modèle III)

Les analyses qui précèdent négligent un point important: location pourrait être traité comme un facteur aléatoire et sex est fixe. Par conséquent le modèle approprié d'ANOVA est de type III (mixte).

Notez que dans toutes les analyses qui précèdent, R a traité cette ANOVA comme si elle était de Type I, et les termes principaux et celui d'interaction ont été testés en utilisant le carré moyen des résidus comme dénominateur des tests de F. Cependant, pour une ANOVA de type III, ces effets devraient être testés en utilisant le carré moyen du terme d'interaction, ou en combinant la somme des carrés de l'erreur et de l'interaction (selon le statisticien consulté!).

En utilisant `stu2wdat`, refaites un tableau d'ANOVA pour `RDWGHT` en considérant location comme facteur aléatoire et sex comme un facteur fixe. Pour ce faire, vous devrez recalculer les valeurs de F pour sex et location en utilisant le carré moyen de l'interaction sex:location au lieu du carré moyen des résidus comme dénominateur. Le mieux c'est de le faire à la mitaine en travaillant avec les Type III Sums of squares du tableau d'ANOVA.

Anova Table (Type III tests)

```

Response: rdwght
      Sum Sq Df F value    Pr(>F)
(Intercept) 106507 1 1081.4552 < 2.2e-16 ***
sex          1745 1  17.7220 4.051e-05 ***
location      9 1   0.0891  0.7656
sex:location  49 1   0.4944  0.4829
Residuals   17530 178
---

```

Pour sex, la nouvelle valeur de F (le rapport des carrés moyens) est de $(1745/1)/(49/1) = 35.6$

Pour obtenir la p-valeur correspondant à cette statistique F, allez à la fenêtre de commandes et tapez: $1 - \text{pf}(F, df1, df2)$, où F est la valeur de F calculée, et df1 et df2 sont les degrés de liberté du numérateur (sex) et dénominateur (SEX:location)

```
1 - pf(35.6, 1, 1)
[1] 0.1057152
```


Notez que maintenant la valeur de p pour sex n'est plus significative. C'est parce que le carré moyen de l'erreur dans l'ANOVA initiale est plus petit que celui associé à l'interaction, mais surtout parce que le nombre de degrés de liberté pour le dénominateur du test de F est passé de 178 à 1 seulement. En général, c'est beaucoup plus difficile d'obtenir des résultats significatifs quand les degrés de liberté pour le dénominateur sont petits.

Plan factoriel à deux facteurs de classification sans réplication

Dans certains plans d'expérience il n'y a pas de réplicats pour chaque combinaison de facteurs, par exemple parce qu'il serait trop coûteux de faire plus d'une observation. L'ANOVA à deux critères de classification est quand même possible dans ces circonstances, mais il y a une limitation importante.

Comme il n'y a pas de réplicats, on ne peut estimer la variance du terme d'erreur. En effet on ne peut qu'estimer la somme des carrés associés à chacun des facteurs principaux, et la quantité de variabilité qui reste (Remainder Mean Square) représente la somme de la variabilité attribuable à l'interaction et au terme d'erreur. Cela a une implication importante : s'il y a une interaction, seul un modèle II d'ANOVA peut être entièrement testé et dans un modèle III d'ANOVA, seul l'effet fixe peut être testé (il est éprouvé en les comparant au carré moyen associé avec le remainder MS). Dans le cas d'un modèle I ou pour l'effet aléatoire d'un modèle III on ne peut tester les effets principaux que si on est sûr qu'il n'y a pas d'interaction.

Un limnologue qui étudie Round Lake dans le Parc Algonquin prend une seule mesure de température (temp, en degrés C) à 10 profondeurs différentes (depth, en m) à quatre dates (date) au cours de l'été. Ses données sont au fichier nr2wdat.Rdata.

 Effectuez une ANOVA à deux critères de classification en utilisant temp comme variable dépendante et date et depth comme variables indépendantes (vous devez changer le type de données pour DEPTH pour que R traite cette variable comme un facteur et non pas une variable continue).


```
anova.model4<-lm(temp ~ date + as.factor(depth), data =
nr2wdat)
Anova(anova.model4, type = 3)
```

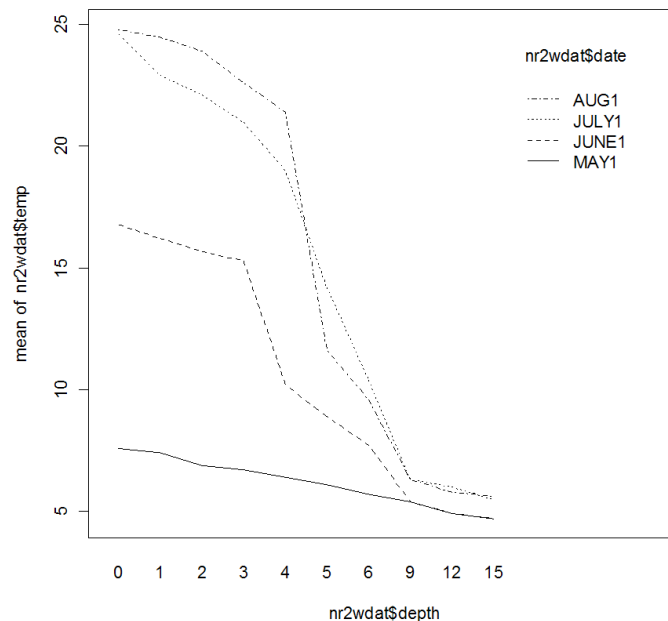
Anova Table (Type III tests)

Response: temp

	Sum Sq	Df	F value	Pr(>F)	
(Intercept)	1511.99	1	125.5652	1.170e-11	***
date	591.15	3	16.3641	2.935e-06	***
as.factor(depth)	1082.82	9	9.9916	1.450e-06	***
Residuals	325.12	27			

Si on suppose que c'est un modèle III d'ANOVA (date aléatoire, Depth fixe), que concluez vous? (Indice: faites un graphique d'interaction température en fonction de la profondeur et la date, pour voir ce qui se passe).

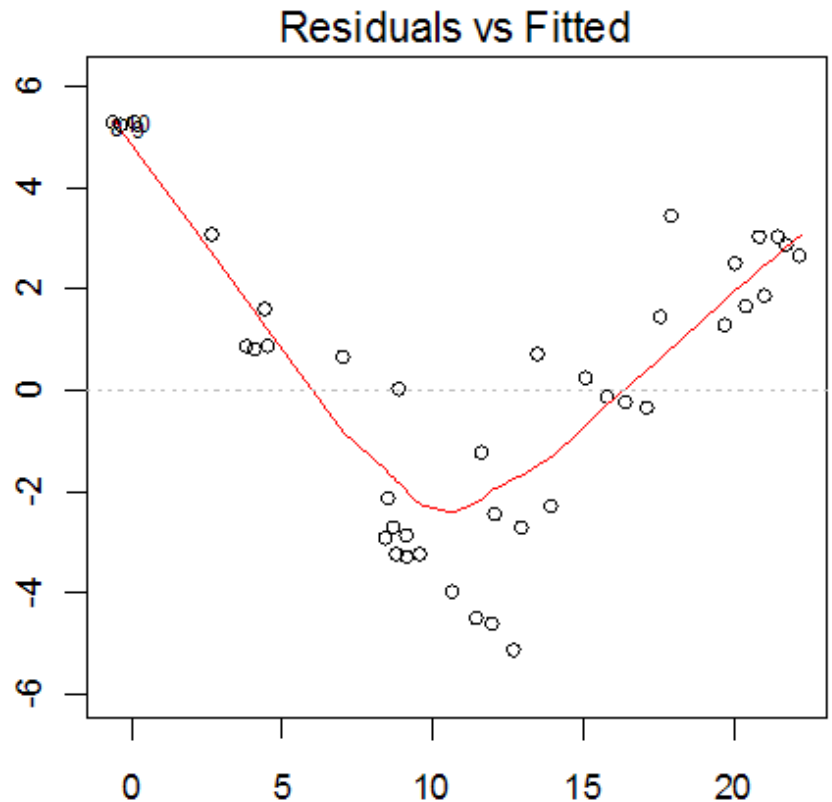
```
interaction.plot(nr2wdat$depth, nr2wdat$date, nr2wdat$temp)
```



La température diminue significativement en profondeur. Pour tester l'effet du mois (le facteur aléatoire), on doit présumer qu'il n'y a pas d'interaction entre la profondeur et le mois (donc que l'effet de la profondeur sur la température est le même à chaque mois). C'est peu probable: si vous faites un graphique de la température en fonction de la profondeur pour chaque mois, vous observerez que le profil de

température change au fur et à mesure du développement de la thermocline. Bref, comme le profil change au cours de l'été, ce modèle ne fait pas de très bonnes prédictions.

Jetez un coup d'œil sur les graphiques des résidus:




Le test de normalité sur les résidus donne $p = 0.16$, donc l'hypothèse de normalité ne semble pas être sérieusement en doute. Pour l'égalité des variances, on peut seulement comparer entre les mois en utilisant les profondeurs comme réplicats (ou l'inverse). En utilisant les profondeurs comme réplicats, on obtient:

```
leveneTest(temp ~ date , data = nr2wdat)
Levene's Test for Homogeneity of Variance
      Df F value    Pr(>F)
group  3 17.979 2.679e-07 ***
```

Il y a donc un problème d'hétéroscédasticité, comme on peut très bien voir dans le graphique des résidus vs les valeurs estimées. Cette analyse n'est donc pas très satisfaisante: il y a des violations des conditions d'application et il semble y avoir une interaction entre depth et date qui pourrait invalider l'analyse

Plans hiérarchiques

Un design expérimental fréquent implique la division de chaque groupe du facteur majeur en sous-groupes aléatoires. Par exemple, une généticienne intéressée par l'effet du génotype sur la résistance à la dessiccation chez la drosophile effectue une expérience. Pour chaque génotype (facteur principal) elle prépare trois chambres de croissance (sous-groupes) avec une température et humidité contrôlées. Dans chaque chambre de croissance, elle place cinq larves, puis mesure le nombre d'heures pendant lesquelles chaque larve survit.

 Le fichier `nestdat.Rdata` contient les résultats d'une expérience semblable. Il contient trois variables : `genotype`, `chamber` et `survival`. Effectuez une ANOVA hiérarchique avec `survival` comme variable dépendante et `genotype` et `chamber/genotype` comme variables indépendantes.

```
nestdat$chamber <- as.factor(nestdat$chamber)
anova.nested <- lm(survival ~ genotype/chamber, data = nest-
dat)
```

Que concluez-vous de cette analyse ? Que devrait être la prochaine étape ? (Indice: si l'effet de `Chamber / genotype` n'est pas significatif, vous pouvez augmenter la puissance des comparaisons entre génotypes en regroupant les chambres de chaque génotype.). Faites-le ! N'oubliez pas de vérifier les conditions d'applications de l'ANOVA!

Analysis of Variance Table

```
Response: survival
      Df Sum Sq Mean Sq F value Pr(>F)
genotype      2 2952.22 1476.11 292.6081 <2e-16 ***
genotype:chamber 6   40.65    6.78   1.3432 0.2639
Residuals     36  181.61    5.04
```

On conclue de cette analyse que la variation entre les chambres de croissance n'est pas significative, mais qu'on doit rejeter l'hypothèse nulle que tous les génotypes ont la même résistance à la dessiccation.

Comme l'effet hiérarchique `chamber / genotype` n'est pas significatif, on peut regrouper les observations pour augmenter le nombre de degrés de liberté:

```
anova.simple <- lm(survival ~ genotype, data = nestdat)
anova(anova.simple)
```

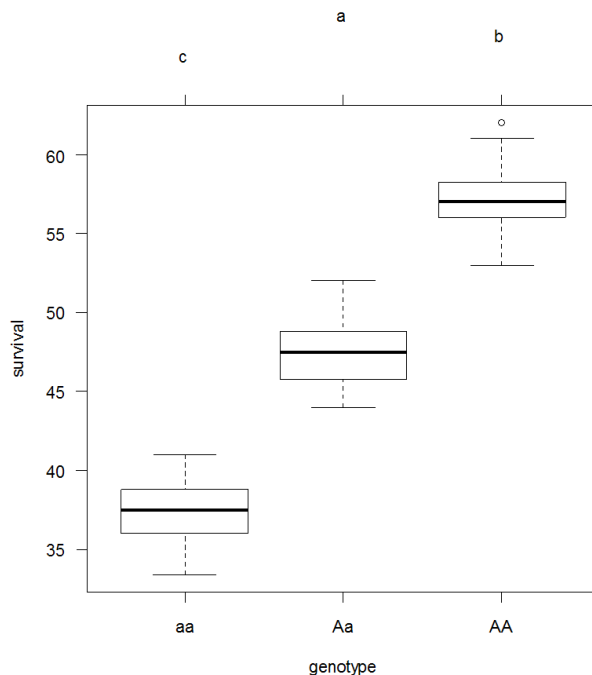
Analysis of Variance Table

```
Response: survival
      Df Sum Sq Mean Sq F value    Pr(>F)
genotype      2 2952.22 1476.11 278.93 < 2.2e-16 ***
Residuals    42  222.26    5.29
```

Donc on conclue qu'il y a une variation significative de résistance à la dessiccation entre les trois génotypes.

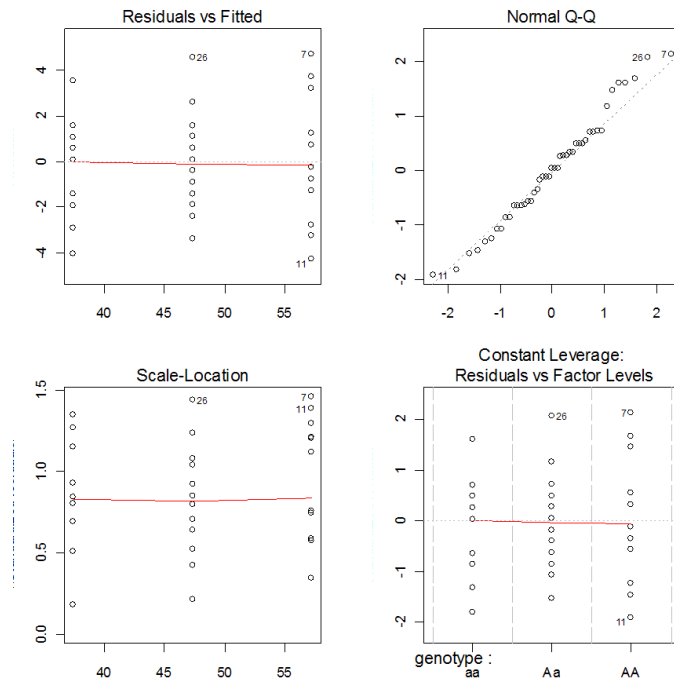
Le graphique de survival en fonction du génotype suggère que la résistance à la dessiccation varie entre chaque génotype. On peut combiner cela avec un test de Tukey

```
par(mfrow=c(1,1))
# Compute and plot means and Tukey CI
means <- glht(anova.simple, linfct = mcp(genotype =
"Tukey"))
cimeans <- cld(means)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(cimeans, las=1) # las option to put y-axis labels as God
intended them
```



On conclue donc que la résistance à la dessiccation (R), telle que mesurée par la survie dans des conditions chaudes et sèches, varie significativement entre les trois génotypes avec $R(AA) > R(Aa) > R(aa)$.

Cependant, avant d'accepter cette conclusion, il faut éprouver les conditions d'application du test. Voici les diagnostics des résidus pour l'ANOVA à un critère de classification (non hiérarchique):



Donc, toutes les conditions d'application semblent être remplies, et on peut donc accepter les conclusions. Notez que si l'on compare le carré moyen des résidus de l'ANOVA hiérarchique et de l'ANOVA à un critère de classification (5.045 vs 5.292), ils sont presque identiques. Cela n'est pas surprenant compte tenu de la faible variabilité associée aux chambres de croissance pour chaque génotype.

ANOVA non paramétrique avec deux facteurs de classification

L'ANOVA non paramétrique à deux critères de classification est une extension de celle à un critère de classification vue précédemment. Elle débute par une ANOVA faite sur les données transformées en rangs. Elle peut se faire sur des données avec ou sans réplicats.

À partir du fichier `stu2wdat.Rdata`, effectuez une ANOVA non paramétrique à deux facteurs de classification pour examiner l'effet de `sex` et `location` sur `rank(rdwght)`.

```
aov.rank <- aov(rank(rdwght) ~ sex * location,
  contrasts = list(sex = contr.sum,
    location = contr.sum), data = Stu2wdat)
Anova(aov.rank, type = 3)
```

Anova Table (Type III tests)

Response: rank(rdwght)				
	Sum Sq	Df	F value	Pr(>F)
(Intercept)	1499862	1	577.8673	< 2.2e-16 ***
sex	58394	1	22.4979	4.237e-06 ***
location	1128	1	0.4347	0.5105
sex:location	1230	1	0.4738	0.4921
Residuals	472383	182		

L'extension de Schreirer-Ray-Hare au test de Kruskal-Wallis se fait ensuite à la main. Il faut d'abord calculer la statistique H égale au rapport de la somme des carrées de l'effet testé, divisée par le carré moyen total. On calcule la statistique H pour chacun des termes. Les statistiques H sont ensuite comparées à une distribution théorique χ^2 en utilisant la commande (dans la fenêtre Commands): `1-pchisq(H, df)`, où H et df sont les statistiques H calculées et les degrés de liberté, respectivement.

Testez l'effet de sex et location sur rdwght. Que concluez-vous ? Comment ce résultat se compare-t-il à celui obtenu en faisant l'ANOVA paramétrique faite précédemment ?

Pour calculer l'extension Schreirer-Ray-Hare au test de Kruskal-Wallis, on doit d'abord calculer le carré moyen total (MS), i.e. la variance des données transformées en rang. Ici, on a 186 observations, donc des rangs; 1, 2, 3, ... 186. La variance de cette série de 186 valeurs peut être calculée simplement par `var(1:186)` (N'est-ce pas que R est fabuleux? Pas toujours évident, mais fabuleux tout de même, non?)

Donc on peut calculer la statistique H pour chaque terme:

$$H_{\text{sex}} = 58394 / \text{var}(1:186) = 20.14$$

$$H_{\text{location}} = 1128 / \text{var}(1:186) = 0.39$$

$$H_{\text{sex:location}} = 1230 / \text{var}(1:186) = 0.42$$

Et convertir ces statistiques en p-valeurs:

$$\text{pour sex: } 1 - \text{pchisq}(20.14, 1) = 7.197556e-06$$

$$\text{location } 1 - \text{pchisq}(0.39, 1) = 0.5322994$$

$$\text{sex:location } 1 - \text{pchisq}(0.42, 1) = 0.516937$$

Ces résultats sont semblables aux résultats de l'ANOVA non-paramétrique à deux critères de classification. Malgré la puissance réduite, il y a encore un effet significatif du sexe, mais ni interaction ni effet du site.

Il y a toutefois une différence importante. Rappelez-vous que dans l'ANOVA paramétrique il y avait un effet significatif de sex en considérant le problème comme un modèle I d'ANOVA. Cependant, si on traite le problème comme un modèle III, l'effet significatif de sex peut en principe disparaître parce que le nombre de dl associés au CM de l'interaction est plus faible que le nombre de dl du CM de l'erreur du modèle I. Dans ce cas ci, cependant, le CM de l'interaction est environ la moitié du CM de l'erreur. Par conséquent, l'effet significatif de sex pourrait devenir encore plus significatif si le problème est analysé (comme il se doit) comme une ANOVA de modèle III. Encore une fois on peut voir l'importance de spécifier le modèle adéquat en ANOVA.

Comparaisons multiples

Les épreuves d'hypothèses subséquentes en ANOVA à plus d'un critère de classification dépendent des résultats initiaux de l'ANOVA. Si vous êtes intéressés à comparer des effets moyens d'un facteur pour tous les niveaux d'un autre facteur (par exemple l'effet du sexe sur la taille des esturgeons peu importe d'où ils viennent), alors vous pouvez procéder exactement tel que décrit dans la section sur les comparaisons multiples suivant l'ANOVA à un critère de classification. Pour comparer les moyennes des cellules entre elles, il faut spécifier l'interaction comme variable qui représente le groupe.

Le fichier `wmdat2.Rdata` contient des mesures de consommation d'oxygène (`o2cons`) de deux espèces (`species`) d'un mollusque (une patelle) à trois concentrations différentes d'eau de mer (`conc`) (ces données sont présentées à la p. 332 de Sokal et Rohlf 1995).

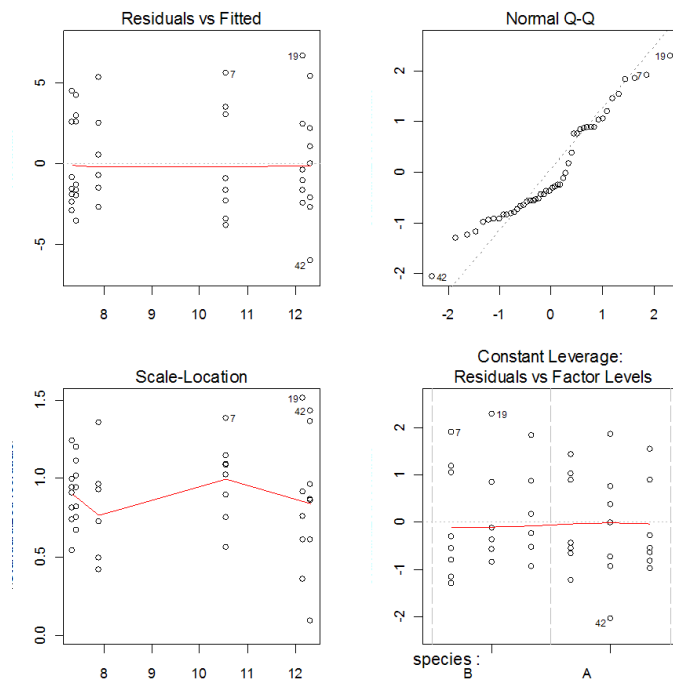
Effectuez une ANOVA factorielle à deux critères de classification sur ces données en utilisant `o2cons` comme variable dépendante et `species` et `conc` comme les facteurs (il va probablement falloir changer le type de données du variable `conc` à facteur). Que concluez-vous ?

Comme l'effectif dans chaque cellule est relativement petit, il faudrait idéalement refaire cette analyse avec une ANOVA non-paramétrique. Pour le moment, contentons nous de la version paramétrique.

Anova Table (Type III tests)

```
Response: o2cons
          Sum Sq Df F value    Pr(>F)
(Intercept) 4441.7  1 464.6163 < 2.2e-16 ***
species      16.6   1   1.7404 0.1942381
conc        181.3   2   9.4833 0.0003993 ***
species:conc  23.9   2   1.2514 0.2965616
Residuals    401.5  42
```

Examinons les graphiques diagnostiques:



Les variances semblent donc égales. Le test de normalité donne:

Shapiro-wilk normality test

```
data: residuals(anova.model15)
w = 0.9369, p-value = 0.01238
```

Il y a donc évidence de non-normalité, mais à part ça tout semble aller. Comme l'ANOVA est relativement robuste à la non-normalité, on va regarder de l'autre côté. (Si vous voulez être plus confiants, vous pouvez tourner une ANOVA non paramétrique. Vous arriverez aux mêmes conclusions.)

🖨️ À la suite des résultats que vous venez d'obtenir, quelles moyennes voudriez-vous comparer ? Pourquoi?

On conclue donc qu'il n'y a pas de différence entre les espèces et que l'effet de la concentration ne dépend pas de l'espèce (il n'y a pas d'interaction). Par conséquent, les seules comparaisons justifiables sont entre les concentrations:

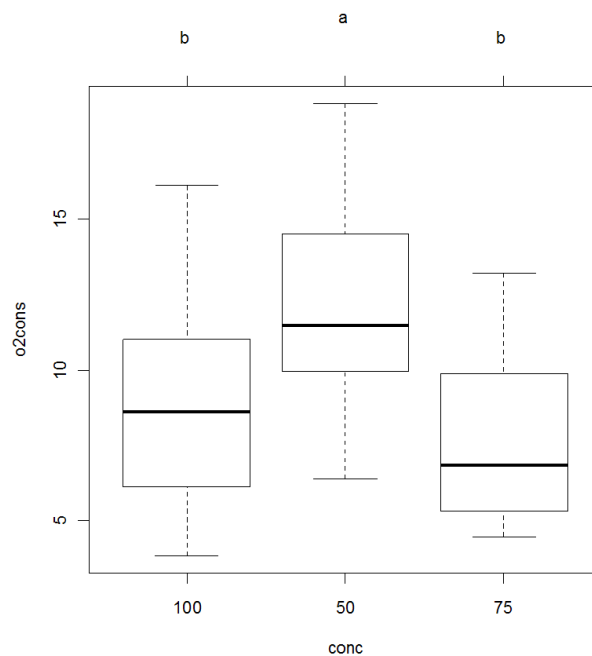

```
# fit simplified model
anova.model6 <- lm (o2cons ~ conc, data = wmcdata2)
# Make Tukey multiple comparisons
TukeyHSD(anova.model6)
```

Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = o2cons ~ conc, data = wmcdata2)

\$conc	diff	lwr	upr	p adj
50-100	3.25500	0.5692518	5.940748	0.0141313
75-100	-1.38125	-4.0669982	1.304498	0.4325855
75-50	-4.63625	-7.3219982	-1.950502	0.0003793

```
par(mfrow=c(1,1))
# Graph of all comparisons for conc
tuk <- glht(anova.model6, linfct = mcp(conc = "Tukey"))
# extract information
tuk.cld <- cld(tuk)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(tuk.cld)
par(old.par)
```



Il y a donc une différence de consommation d'oxygène significative lorsque la salinité est réduite de 50%, mais pas à 25% de réduction.

☞ Répétez les deux analyses précédentes sur les données du fichier `wmc2dat2.Rdata`. Comment les résultats se comparent-ils à ceux obtenus précédemment ?

En utilisant `wmc2dat2.Rdata`, on obtient:

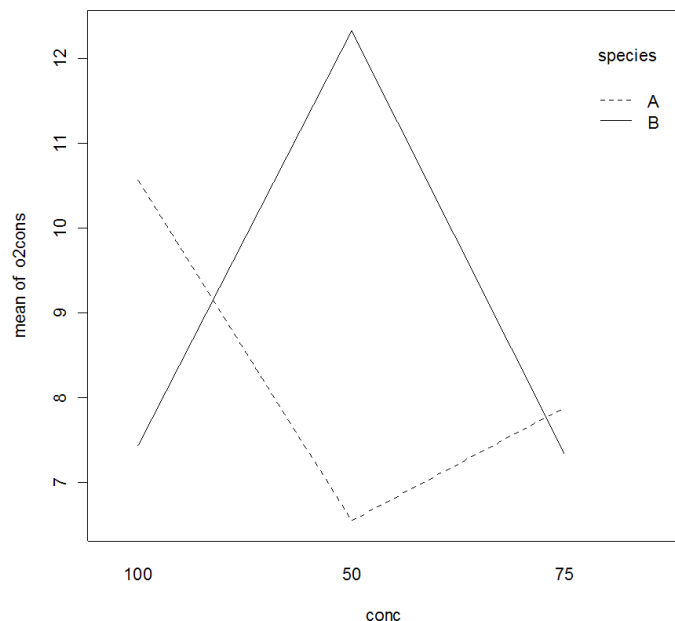
Anova Table (Type III tests)

Response: `o2cons`

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	3618.2	1	381.8998	< 2.2e-16 ***
species	5.8	1	0.6162	0.4368642
conc	29.0	2	1.5283	0.2287237
species:conc	168.2	2	8.8742	0.0006101 ***
Residuals	397.9	42		

Dans ce cas ci, il y a une interaction significative, et il n'est par conséquent pas approprié de comparer les moyennes regroupées par espèce ou concentration. Ceci est clairement visualisé par un graphique d'interaction:

`with(wmc2dat2, interaction.plot(conc, species, o2cons))`



☞ Toujours en utilisant les données de `wmc2dat2.Rdata`, comparez les 6 moyennes avec l'ajustement Bonferroni. Pour ce faire, il sera utile de créer une nouvelle variable qui combine `species` et `conc`:

```
wmc2dat2$species.conc <- paste(wmc2dat2$species,
                               wmc2dat2$conc)
```

ensuite on peut faire les comparaisons de Bonferroni:

```
with(wmc2dat2, pairwise.t.test(o2cons, species.conc,
                               p.adj = "bonf"))
```

Pairwise comparisons using t tests with pooled SD

data: o2cons and species.conc

	A 100	A 50	A 75	B 100	B 50
A 50	1.000	-	-	-	-
A 75	1.000	0.124	-	-	-
B 100	0.737	0.056	1.000	-	-
B 50	1.000	1.000	0.096	0.043	-
B 75	0.647	0.048	1.000	1.000	0.036

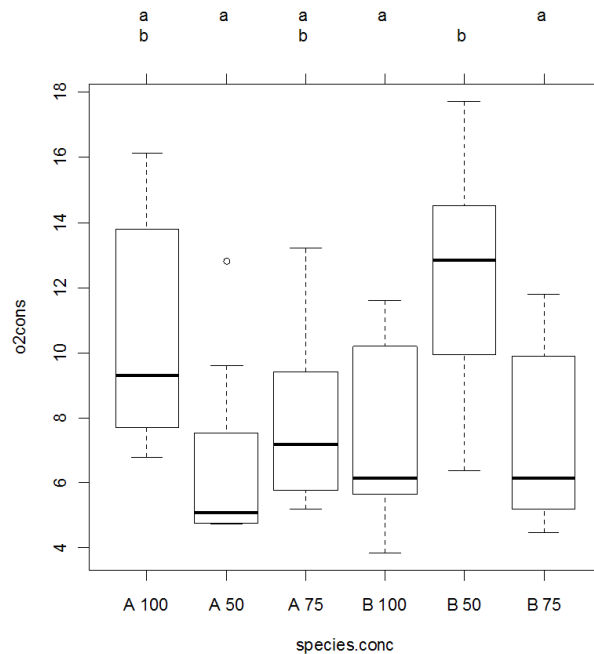
P value adjustment method: bonferroni

Ces comparaisons sont un peu plus difficiles à interpréter, mais l'analyse examine essentiellement les différences entre les concentrations de l'eau dans l'espèce A (nommé adj1) et pour les différences entre les concentrations dans l'espèce B (nommé adj2). Cette analyse indique que la différence principale est entre la concentration de 50% pour l'espèce B et les concentrations de 75 et 100% de l'espèce B, tandis qu'il n'y a aucunes différences significatives pour l'espèce A.

Je trouve ces tableaux de résultats peu satisfaisants parce qu'ils indiquent seulement les p-valeurs sans indices de la taille de l'effet. On peut obtenir à la fois le résultat des tests de comparaison multiple et un indice de la taille de l'effet à l'aide du code suivant:

```
# fit one-way anova comparing all combinations of spe-
cies.conc combinations
anova.modelx <- aov(o2cons ~ species.conc, data = wmc2dat2)

tuk2 <- glht(anova.modelx, linfct = mcp(species.conc =
"Tukey"))
# extract information
tuk2.cld <- cld(tuk2)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(tuk2.cld)
par(old.par)
```



Dans cette analyse on a utilisé le $CM = 9.474$ du modèle d'ANOVA pour comparer les moyennes. En ce faisant, on présume qu'il s'agit d'une situation d'ANOVA de modèle I, ce qui n'est peut-être pas le cas (conc est certainement fixe, mais species peut être fixe ou aléatoire).

Test de permutation pour l'ANOVA à deux facteurs de classification

Quand les données ne rencontrent pas les conditions d'application des tests paramétriques d'ANOVA à un ou plusieurs facteurs de classification, il est possible d'utiliser les tests de permutation comme une alternative aux tests non-paramétriques pour calculer des p-valeurs. Le code suivant est pour un modèle I d'une ANOVA à deux facteurs de classification. Je vous laisse le soin d'adapter ce code pour d'autres modèles. (J'offre même des points boni pour une solution élégante pour des modèles à plusieurs facteurs de classification).

```
#####
# Permutation test for two way ANOVA
# Ter Braak creates residuals from cell means and then per-
# mutes across
# all cells
# This can be accomplished by taking residuals from the full
# model
```

```

# modified from code written by David C. Howell
# http://www.uvm.edu/~dhowell/StatPages/More_Stuff/Permutation%20Anova/PermTestsAnova.html

nreps = 500
dependent <- Stu2wdat$rdwght
factor1 <- as.factor(Stu2wdat$sex)
factor2 <- as.factor(Stu2wdat$location)

my.dataframe <- data.frame(dependent, factor1, factor2)
my.dataframe.noNA <- my.dataframe[complete.cases(my.dataframe),
]

mod <- lm(dependent ~ factor1 + factor2 + factor1:factor2,
  data = my.dataframe.noNA)
res <- mod$residuals
TBint <- numeric(nreps)
TB1 <- numeric(nreps)
TB2 <- numeric(nreps)

ANOVA <- summary(aov(mod))
cat(" The standard ANOVA for these data follows ",
  "\n")
F1 <- ANOVA[[1]]$"F value"[1]
F2 <- ANOVA[[1]]$"F value"[2]
Finteract <- ANOVA[[1]]$"F value"[3]
print(ANOVA)
cat("\n")
cat("\n")

TBint[1] <- Finteract
for (i in 2:nreps) {
  newdat <- sample(res, length(res), replace = FALSE)
  modb <- summary(aov(newdat ~ factor1 + factor2 +
    factor1:factor2,
    data = my.dataframe.noNA))
  TBint[i] <- modb[[1]]$"F value"[3]
  TB1[i] <- modb[[1]]$"F value"[1]
  TB2[i] <- modb[[1]]$"F value"[2]
}
probInt <- length(TBint[TBint >= Finteract])/nreps
prob1 <- length(TB1[TB1 >= F1])/nreps
prob2 <- length(TB2[TB2 >= F2])/nreps
cat("\n")
cat("\n")
print("Resampling as in ter Braak with unrestricted sampling
of cell residuals. ")
cat("The probability for the effect of Interaction is ",
  probInt, "\n")
cat("The probability for the effect of Factor 1 is ",
  prob1, "\n")
cat("The probability for the effect of Factor 2 is ",
  prob2, "\n")

```

Si vous avez la chance d'avoir accès au package `lmPerm`, vous pouvez effectuer le test de permutation beaucoup plus rapidement et facilement:

```
#####

## lmPerm version of permutation test
require(lmPerm2)
# for generality, copy desired dataframe to mydata
# and model formula to myformula
mydata<-Stu2wdat
myformula<-as.formula("rdwght ~ sex+location+sex:location")

# Fit desired model on the desired dataframe
mymodel <- lm(myformula, data=mydata)
# Calculate permutation p-value
anova(lmp(myformula, data=mydata, perm = "Prob", center=FALSE, Ca=0.001))
```

Bootstrap pour l'ANOVA à deux facteurs de classification

Dans la plupart des cas, les tests de permutation seront plus appropriés que le bootstrap pour les designs d'ANOVA. J'ai quand même un bout de code qui pourra servir si vous en avez besoin:

```
#####
#####

# Bootstrap for two-way ANOVA
# You possibly want to edit bootfunction.mod1 to return
other values
# Here it returns the standard coefficients of the fitted
model
# Requires boot library
#

nreps = 5000
dependent <- Stu2wdat$rdwght
factor1 <- as.factor(Stu2wdat$sex)
factor2 <- as.factor(Stu2wdat$location)

my.dataframe <- data.frame(dependent, factor1, factor2)
my.dataframe.noNA <- my.dataframe[complete.cases(my.dataframe), ]

require(boot)
# Fit model on observed data
mod1 <- aov(dependent ~ factor1 + factor2 + factor1:factor2,
            data = my.dataframe.noNA)
```

```

# Bootstrap 1000 time using the residuals bootstrapping
methods to
# keep the same unequal number of observations for each
level of the indep. var.

fit <- fitted(mod1)
e <- residuals(mod1)
X <- model.matrix(mod1)

bootfunction.mod1 <- function(data, indices) {
  y <- fit + e[indices]
  bootmod <- lm(y ~ X)
  coefficients(bootmod)
}

bootresults <- boot(my.dataframe.noNA, bootfunction.mod1,
  R = 1000)

bootresults

# Calculate 90% CI and plot bootstrap estimates separately
# for each model parameter

boot.ci(bootresults, conf = 0.9, index = 1)
plot(bootresults, index = 1)

boot.ci(bootresults, conf = 0.9, index = 3)
plot(bootresults, index = 3)

boot.ci(bootresults, conf = 0.9, index = 4)
plot(bootresults, index = 4)

boot.ci(bootresults, conf = 0.9, index = 5)
plot(bootresults, index = 5)

```

Labo- Régression multiple

Après avoir complété cet exercice de laboratoire, vous devriez pouvoir :

- Utiliser R pour ajuster une régression multiple et comparer des modèles selon l'approche inférentielle et celle de la théorie de l'information
- Utiliser R pour éprouver des hypothèses sur l'effet des variables indépendantes sur la variable dépendante.
- Utiliser R pour évaluer la multicolinéarité entre les variables indépendantes et en évaluer ses effets.
- Utiliser R pour effectuer une régression curvilinéaire (polynomiale).

Conseils généraux

Les variables qui intéressent les biologistes sont généralement influencées par plusieurs facteurs, et une description exacte ou une prédiction de la variable dépendante requiert que plus d'une variable soit incluse dans le modèle. La régression multiple permet de quantifier l'effet de plusieurs variables continues sur la variable dépendante.

Il est important de réaliser que la maîtrise de la régression multiple ne s'acquiert pas instantanément. Les débutants doivent garder à l'esprit plusieurs points importants :

1. Un modèle de régression multiple peut être hautement significatif même si aucun des termes pris isolément ne l'est (ceci est causé par la multicolinéarité),
2. Un modèle peut ne pas être significatif alors que l'un ou plusieurs des termes le sont (ceci est un signe d'un modèle trop complexe ("overfitting")) et,
3. À moins que les variables indépendantes soient parfaitement orthogonales (c'est-à-dire qu'il n'y ait aucune corrélation entre elles et donc pas de multicolinéarité) les diverses approches de sélection des variables indépendantes peuvent mener à des modèles différents.

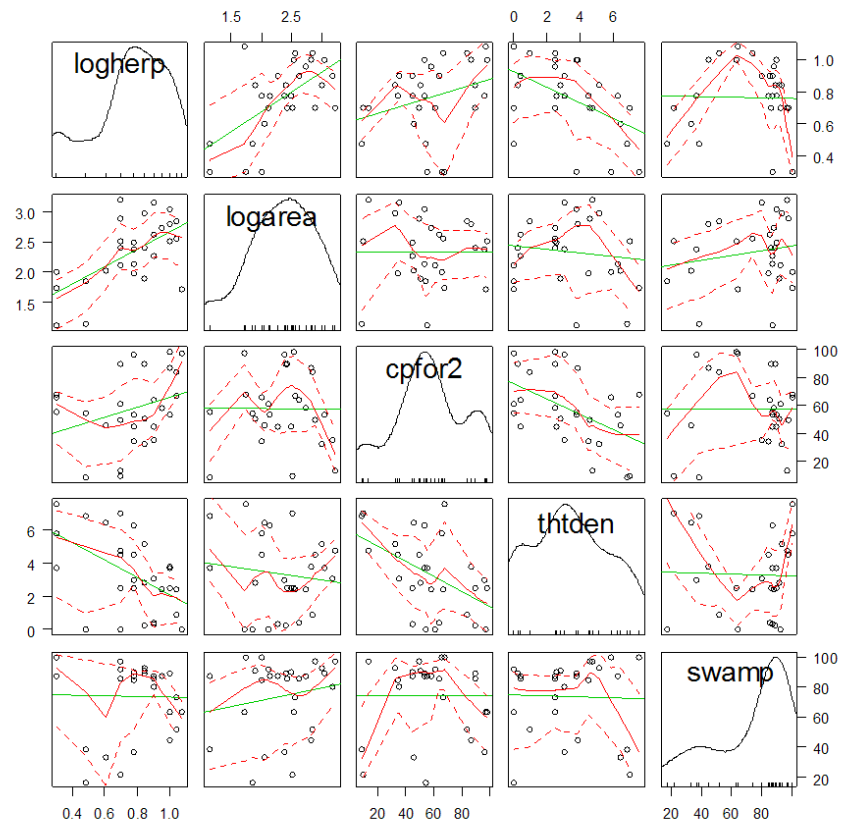
Examen préliminaire des données

Le fichier `Mregdat.Rdata` contient des données de richesse spécifique de quatre groupes d'organismes dans 30 marais de la région Ottawa-Cornwall-Kingston. Les variables sont la richesse spécifique des oiseaux (`bird`, et son logarithme base 10 `logbird`), des mammifères (`mammal`, `logmam`), des amphibiens et reptiles (`herptile`, `logherp`) et celle des vertébrés (`totspp`, `logtot`) ; les coordonnées des sites (`lat`, `long`) ; la superficie du marais (`logarea`), le pourcentage du marais inondé toute l'année (`swamp`) le pourcentage des terres couvertes par des forêts dans un rayon de 1km du marais (`cpfor2`) et la densité des routes pavées (en m/ha) dans un rayon de 1km du marais (`thtden`).

Nous allons nous concentrer sur les amphibiens et les reptiles (`herptile`) pour cet exemple, il est donc avisé d'examiner la distribution de cette variable et les corrélations avec les variables indépendantes potentielles:

```
require(car)
scatterplotMatrix(~logherp + logarea + cpfor2 + thtden +
  swamp,
  reg.line = lm, smooth = TRUE, span = 0.5, diagonal = "density",
  data = mydata)
```

Figure 1.



En utilisant les données de ce fichier, faites la régression simple de logherp sur logarea. Que concluez-vous à partir de cette analyse?

```
model.loga<-lm(logherp ~ logarea, data = mydata)
summary(model.loga)
```

Call:

```
lm(formula = logherp ~ logarea, data = mydata)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.38082	-0.09265	0.00763	0.10409	0.46977

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.18503	0.15725	1.177	0.249996
logarea	0.24736	0.06536	3.784	0.000818 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1856 on 26 degrees of freedom

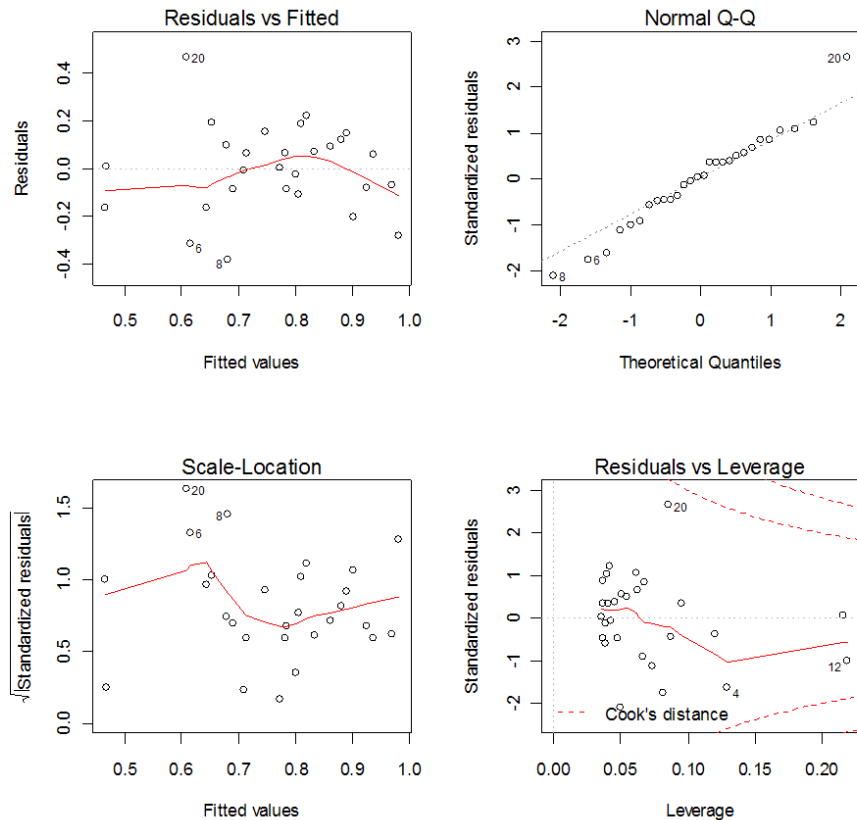
(2 observations deleted due to missingness)

Multiple R-squared: 0.3552, Adjusted R-squared: 0.3304


F-statistic: 14.32 on 1 and 26 DF, p-value: 0.0008185

```
opar <- par(mfrow(2,2))
plot(model.loga)
opar
```

Figure 2.



Il semble donc y avoir une relation positive entre la richesse spécifique des reptiles et des amphibiens et la surface des marais. La régression n'explique cependant qu'environ le tiers de la variabilité ($R^2=0.355$). L'analyse des résidus indique qu'il n'y a pas de problème avec la normalité, l'homoscédasticité, ni l'indépendance.

 Faites ensuite la régression de `logherp` sur `cpfor2`. Que concluez-vous?


```
Call:
lm(formula = logherp ~ cpfor2, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.4909 -0.1027  0.0588  0.1603  0.2516

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.609197   0.104233   5.845 3.68e-06 ***
cpfor2       0.002706   0.001658   1.632  0.115
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2202 on 26 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.09289,    Adjusted R-squared:  0.058
F-statistic: 2.662 on 1 and 26 DF,  p-value: 0.1148
```

Ici, on doit accepter l'hypothèse nulle et conclure qu'il n'y a pas de relation entre la richesse spécifique dans les marais et la proportion de forêts sur les terres adjacentes. Qu'est-ce qui arrive quand on fait une régression avec les 2 variables indépendantes?

 Refaites la régression de `logherp` sur `logarea` et `cpfor2` à la fois, soit que `logherp ~ logarea + cpfor2`. Que concluez-vous?

```
Call:
lm(formula = logherp ~ logarea + cpfor2, data = mydata)


Residuals:
    Min       1Q   Median       3Q      Max
-0.40438 -0.11512  0.01774  0.08187  0.36179

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.027058   0.166749   0.162  0.872398
logarea      0.247789   0.061603   4.022  0.000468 ***
cpfor2       0.002724   0.001318   2.067  0.049232 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.175 on 25 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.4493,    Adjusted R-squared:  0.4052
F-statistic: 10.2 on 2 and 25 DF, p-value: 0.0005774
Residual standard error: 0.175 on 25 degrees of freedom
Multiple R-Squared:  0.4493
F-statistic: 10.2 on 2 and 25 degrees of freedom, the p-value is 0.0005774
2 observations deleted due to missing values
```

On voit donc qu'on peut rejeter les 2 hypothèses nulles que la pente de la régression de `logherp` sur `logarea` est zéro et que la pente de la régression de `logherp` sur `cpfor2` est zéro.

Pourquoi `cpfor2` devient-il un facteur significatif dans la régression multiple alors qu'il n'est pas significatif dans la régression simple? Parce qu'il est parfois nécessaire de contrôler pour l'effet d'une variable pour pouvoir détecter les effets plus subtils d'autres variables. Ici, il y a une relation significative entre `logherp` et `logarea` qui masque l'effet de `cpfor2` sur `logherp`. Lorsque le modèle tient compte des deux variables explicatives, il devient possible de détecter l'effet de `cpfor2`.

 Ajustez un autre modèle, cette fois en remplaçant `cpfor2` par `thtden` (`Logherp ~ Logarea + thtden`). Que concluez-vous?

```
Call:
lm(formula = logherp ~ logarea + thtden, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.31583 -0.12326  0.02095  0.13201  0.31674

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.37634   0.14926   2.521  0.018437 *
logarea      0.22504   0.05701   3.947  0.000567 ***
thtden       0.00000   0.00000   0.000  1.000000
```

```

thtden      -0.04196    0.01345   -3.118 0.004535 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.1606 on 25 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.5358,    Adjusted R-squared: 0.4986
F-statistic: 14.43 on 2 and 25 DF,  p-value: 6.829e-05

```

On rejette donc l'hypothèse nulle que la richesse spécifique n'est pas influencée par la taille des marais (`logarea`) ni par la densité des routes (`thtden`). Notez qu'ici il y a une relation négative significative entre la richesse spécifique des amphibiens et reptiles et la densité des routes sur les terres adjacentes, tandis que la relation est positive pour la taille des marais et pour la densité des forêts (`cpfor2`; résultat de la dernière régression).

Le R^2 de ce modèle est plus élevé que pour le précédent, reflétant une corrélation plus forte entre `logherp` et `thtden` qu'entre `logherp` et `cpfor2`.

La richesse spécifique des reptiles et amphibiens semble donc reliée à la surface de marais (`logarea`), la densité des routes (`thtden`), et possiblement au couvert forestier sur les terres adjacentes aux marais (`cpfor2`). Cependant, les trois variables ne sont peut-être pas nécessaires dans un modèle prédictif. Si deux des trois variables (disons `cpfor2` et `thtden`) sont parfaitement corrélées, alors l'effet de `thtden` ne serait rien de plus que celui de `cpfor2` (et vice-versa) et un modèle incluant l'une des deux variables ferait des prédictions identiques à un modèle incluant ces deux variables (en plus de `logarea`).

 Estimez un modèle de régression avec `logherp` comme variable dépendante et `logarea`, `cpfor2` et `thtden` comme variables indépendantes. Que concluez-vous?

```

call:
lm(formula = logherp ~ logarea + cpfor2 + thtden, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.30729 -0.13779  0.02627  0.11441  0.29582

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.284765   0.191420   1.488 0.149867
logarea      0.228490   0.057647   3.964 0.000578 ***
cpfor2       0.001095   0.001414   0.774 0.446516 ①
thtden      -0.035794   0.015726  -2.276 0.032055 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1619 on 24 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.5471 ②,    Adjusted R-squared: 0.4904

```

F-statistic: 9.662 on 3 and 24 DF, p-value: 0.0002291

Plusieurs choses sont à noter ici:

- ❶ Tel que prédit, le coefficient de régression pour `cpfor2` n'est plus significativement différent de 0. Une fois que la variabilité attribuable à `logarea` et `thtden` est enlevée, il ne reste qu'une fraction non-significative de la variabilité attribuable à `cpfor2`.
- ❷ Le R^2 pour ce modèle (0.547) n'est que légèrement supérieur au R^2 du modèle avec seulement `logarea` et `thtden` (.536), ce qui confirme que `cpfor2` n'explique pas grand-chose de plus.

Notez aussi que même si le coefficient de régression pour `thtden` n'a pas beaucoup changé par rapport à ce qui avait été estimé lorsque seul `thtden` et `logarea` étaient dans le modèle (0-.036 vs -0.042), l'erreur type pour l'estimé du coefficient est plus grand, et ce modèle plus complexe mène à un estimé moins précis. Si la corrélation entre `thtden` et `cpfor2` était plus forte, la décroissance de la précision serait encore plus grande.

On peut comparer les deux derniers modèles (i.e., le modèle incluant les 3 variables et celui avec seulement `logarea` and `thtden`) pour décider lequel privilégier.

```
anova(model.loga.cpfor2.thtden, model.loga.thtden)
```

Analysis of Variance Table

Model 1: `logherp ~ logarea + cpfor2 + thtden`

Model 2: `logherp ~ logarea + thtden`

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	24	0.62937				
2	25	0.64508	-1	-0.01571	0.599	0.4465

Cette comparaison révèle que le modèle à 3 variables ne fait pas de prédictions significativement meilleures que le modèle avec seulement `Logarea` et `thtden`. Ce résultat n'est pas surprenant puisque le test de signification pour `cpfor2` dans le modèle complet indique qu'il faut accepter l'hypothèse nulle.

À la suite de cette analyse, on doit conclure que :


1. Le meilleur modèle est celui incluant `thtden` et `logarea`.
2. Il y a une relation négative entre la richesse spécifique des amphibiens et reptiles et la densité des routes sur les terres adjacentes.
3. Il y a une relation positive entre la richesse spécifique et la taille des marais.

Notez que le “meilleur” modèle n’est pas nécessairement le modèle parfait, seulement le meilleur n’utilisant que ces trois variables indépendantes. Il est évident qu’il y a d’autres facteurs qui contrôlent la richesse spécifique dans les marais puisque, même le “meilleur” modèle n’explique que la moitié de la variabilité.

Régression multiple pas-à-pas (stepwise)

Quand le nombre de variables prédictives est restreint, comme dans l’exemple précédent, il est aisé de comparer manuellement les modèles pour sélectionner le plus adéquat. Cependant, lorsque le nombre de variables indépendantes augmente, cette approche n’est rapidement plus utilisable et il est alors utile d’utiliser une méthode automatisée.

La sélection pas à pas avec R utilise le Critère Informatif de Akaike (Akaike Information Criterion, $AIC = n \ln(RSS) + 2K$ où K le nombre de variables indépendantes, n est le nombre d’observations, et RSS est la somme des carrés des résidus) comme mesure de la qualité d’ajustement des modèles. Cette mesure favorise la précision des prédictions et pénalise la complexité. **Lorsque l’on compare des modèles par AIC, le modèle avec le plus petit AIC est le modèle à préférer.**

 Refaite la régression précédente (logherp vs logarea cpfor2 et thtden), mais cette fois en utilisant la fonction stepAIC pour activer la sélection pas à pas des variables indépendantes:

```
model.loga.cpfor2.thtden<-lm(logherp ~ logarea + cpfor2 +
thtden, data = mydata)
# Stepwise Regression
library(MASS)
step <- stepAIC(model.loga.cpfor2.thtden, direction="both")
step$anova # display results
```

```
Start: AIC=-98.27 1
logherp ~ logarea + cpfor2 + thtden
```

	Df	Sum of Sq	RSS	AIC ²
- cpfor2	1	0.016	0.645	-99.576
<none>			0.629	-98.267
- thtden	1	0.136	0.765	-94.794
- logarea	1	0.412	1.041	-86.167

```
Step: AIC=-99.58
logherp ~ logarea + thtden
```

	Df	Sum of Sq	RSS	AIC ³
<none>			0.645	-99.576
+ cpfor2	1	0.016	0.629	-98.267
- thtden	1	0.251	0.896	-92.376
- logarea	1	0.402	1.047	-88.013

```

> step$anova # display results
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
logherp ~ logarea + cpfor2 + thtden

Final Model:
logherp ~ logarea + thtden

      Step Df  Deviance Resid. Df Resid. Dev      AIC
1              24  0.6293717 -98.26666
2 - cpfor2    1  0.01570813      25  0.6450798 -99.57640
>

```

R nous donne::

- ❶ L'ajustement (mesuré par AIC) du modèle complet en premier lieu.
- ❷ L'AIC des modèles dans lesquels une variable a été enlevée du modèle complet. Notez que c'est seulement en enlevant cpfor2 du modèle qu'on peut réduire l'AIC
- ❸ La valeur de AIC pour les modèles auxquels on enlève ou on ajoute une variable au modèle sélectionné à la première étape.(i.e. $\text{logherp} \sim \text{logarea} + \text{thtden}$). Notez qu'aucun des modèles n'a un AIC inférieur à ce modèle.

Au lieu de débiter par le modèle complet (saturé) et enlever des termes, on peut commencer par le modèle nul et ajouter des termes:

```

# Forward selection approach
model.null<-lm(logherp ~ 1, data = mydata)
step <- stepAIC(model.null, scope=~. + logarea + cpfor2 +
thtden, direction="forward")
step$anova # display results

```

```

Start: AIC=-82.09
logherp ~ 1

```

	Df	Sum of Sq	RSS	AIC
+ logarea	1	0.494	0.896	-92.376
+ thtden	1	0.342	1.047	-88.013
+ cpfor2	1	0.129	1.260	-82.820
<none>			1.390	-82.091

```

Step: AIC=-92.38
logherp ~ logarea

```

	Df	Sum of Sq	RSS	AIC
+ thtden	1	0.251	0.645	-99.576
+ cpfor2	1	0.131	0.765	-94.794
<none>			0.896	-92.376

```

Step: AIC=-99.58
logherp ~ logarea + thtden

```

	Df	Sum of Sq	RSS	AIC
<none>			0.645	-99.576
+ cpfor2	1	0.016	0.629	-98.267


```

> step$anova # display results
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
logherp ~ 1

Final Model:
logherp ~ logarea + thtden

      Step Df  Deviance Resid. Df Resid. Dev      AIC
1              27  1.3895281 -82.09073
2 + logarea    1  0.4935233    26  0.8960048 -92.37639
3 + thtden     1  0.2509250    25  0.6450798 -99.57640
>

```

Le résultat final est le même, mais la trajectoire est différente. Dans ce cas, R débute avec le modèle le plus simple et ajoute une variable indépendante à chaque étape, sélectionnant la variable minimisant AIC à cette étape. Le modèle de départ a donc seulement une ordonnée à l'origine. Puis, logarea est ajouté, suivi de thtden. cpfor2 n'est pas ajouté au modèle, car son addition fait augmenter l'AIC.

Il est recommandé de comparer le résultat final de plusieurs approches. Si le modèle retenu diffère selon l'approche utilisée, c'est un signe que le "meilleur" modèle est possiblement difficile à identifier et que vous devriez être circonspects dans vos inférences. Dans notre exemple, pas de problème: toutes les méthodes convergent sur le même modèle final.

Pour conclure cette section, quelques conseils concernant les méthodes automatisées de sélection des variables indépendantes:

1. Les différentes méthodes de sélection des variables indépendantes peuvent mener à des modèles différents. Il est souvent utile d'essayer plus d'une méthode et de comparer les résultats. Si les résultats diffèrent, c'est presque toujours à cause de multicollinéarité entre les variables indépendantes.
2. Attention à la régression pas-à-pas. Les auteurs de SYSTAT en disent: "Stepwise regression is probably the most abused computerized statistical technique ever devised. If you think you need automated stepwise regression to solve a particular problem, you probably don't. Professional statisticians rarely use automated stepwise regression because it does not necessarily find the "best" fitting model, the "real" model, or alternative "plausible" models. Furthermore, the order in which variables enter or leave a stepwise program is usually of no theoretical significance. You are always better off thinking about why a model could generate your data and then testing that model." En bref, on abuse trop souvent de cette technique.

3. Il faut toujours garder à l'esprit que l'existence d'une régression significative n'est pas suffisante pour prouver une relation causale.

Detecter la multicollinéarité

La multicollinéarité est la présence de corrélations entre les variables **indépendantes**. Lorsqu'elle est extrême (multicollinéarité parfaite) elle empêche l'estimation des modèles statistiques. Lorsqu'elle est grande ou modérée, elle réduit la puissance de détection de l'effet des variables indépendantes individuellement, **mais elle n'empêche pas le modèle de faire des prédictions**.

Un des indices les plus utilisés pour quantifier la multicollinéarité et le facteur d'inflation de la variance (VIF, variance inflation factor). Le fichier d'aide du package HH explique ainsi son calcul:

"A simple diagnostic of collinearity is the variance inflation factor, VIF one for each regression coefficient (other than the intercept). Since the condition of collinearity involves the predictors but not the response, this measure is a function of the X's but not of Y. The VIF for predictor i is $1 / (1 - R_i^2)$, where R_i^2 is the R^2 from a regression of predictor i against the remaining predictors. If R_i^2 is close to 1, this means that predictor i is well explained by a linear function of the remaining predictors, and, therefore, the presence of predictor i in the model is redundant. Values of VIF exceeding 5 are considered evidence of collinearity: The information carried by a predictor having such a VIF is contained in a subset of the remaining predictors. If, however, all of a model's regression coefficients differ significantly from 0 ($p\text{-value} < .05$), a somewhat larger VIF may be tolerable."

Bref, les VIF indiquent de combien l'incertitude de chaque coefficient de régression est augmentée par la multicollinéarité.

Attrappe. Il y a plusieurs fonctions `vif()` (j'en connais au moins trois dans les packages `car`, `HH` et `DAAG`), et je ne sais pas en quoi elles diffèrent.

On peut calculer les VIF avec la fonction `vif()` du package `car` :

```
require(car)
vif(model.loga.cpfor2.thtden)
```

```
logarea  cpfor2  thtden
1.022127  1.344455  1.365970
```

Ici, il n'y a pas d'évidence de multicollinéarité car toutes les valeurs de VIF sont près de 1.

Régression polynomiale

La régression requiert la linéarité de la relation entre les variables dépendante et indépendante(s). Lorsque la relation n'est pas linéaire, il est parfois possible de linéariser la relation en effectuant une transformation sur une ou plusieurs variables. Cependant, dans bien des cas il est impossible de transformer les axes pour rendre la relation linéaire. On doit alors utiliser une forme ou l'autre de régression non-linéaire.

La forme la plus simple de régression non-linéaire est la régression polynomiale dans laquelle les variables indépendantes sont à une puissance plus grande que 1 (Ex : X^2 ou X^3)


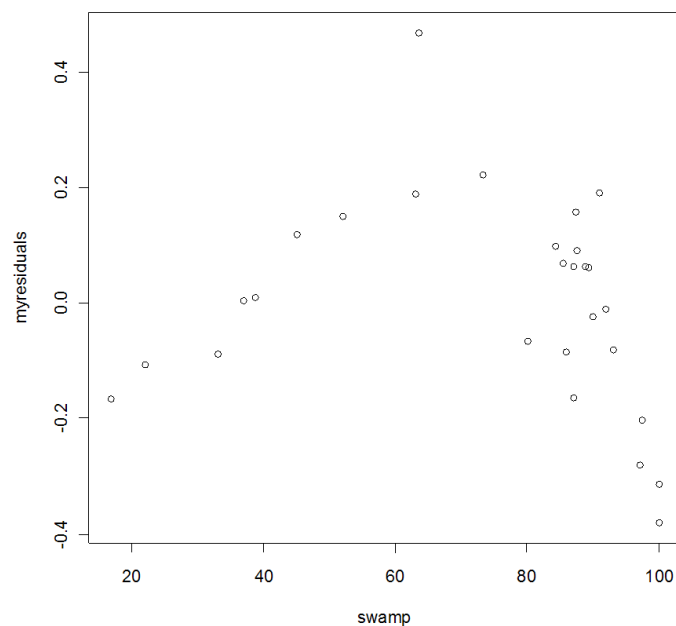

 Faites un diagramme de dispersion des résidus (residual) de la régression logherp-logarea en fonction de swamp.

Figure 3.



 L'examen de ce graphique suggère qu'il y a une forte relation entre les deux variables, mais qu'elle n'est pas linéaire. Essayez de faire une régression de residual sur swamp. Quelle est votre conclusion?

```
call:
lm(formula = myresiduals ~ swamp, data = mydata.noNA)
```


```
Residuals:
    Min       1Q   Median       3Q      Max
-0.350880 -0.138189  0.003131  0.108485  0.458021
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.084571	0.109265	0.774	0.446
swamp	-0.001145	0.001403	-0.816	0.422

Residual standard error: 0.1833 on 26 degrees of freedom
 Multiple R-squared: 0.02498, Adjusted R-squared: -0.01252
 F-statistic: 0.666 on 1 and 26 DF, p-value: 0.4219

En deux mots, l'ajustement est épouvantable! Malgré le fait que le graphique suggère une relation très forte entre les deux variables. Cependant, cette relation n'est pas linéaire... (ce qui est également apparent si vous examinez les résidus du modèle linéaire).

 Refaites la régression d'en haut, mais cette fois incluez un terme pour représenter $(\text{swamp})^2$. L'expression devrait apparaître comme: `residuals ~ SWAMP + SWAMP^2`. Que concluez-vous? Qu'est-ce que l'examen des résidus de cette régression multiple révèle?

Call:

```
lm(formula = myresiduals ~ swamp + I(swamp^2), data = mydata.noNA)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.181185	-0.085350	0.007377	0.067327	0.242455


Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7.804e-01	1.569e-01	-4.975	3.97e-05 ***
swamp	3.398e-02	5.767e-03	5.892	3.79e-06 ***
I(swamp^2)	-2.852e-04	4.624e-05	-6.166	1.90e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1177 on 25 degrees of freedom
 Multiple R-squared: 0.6132, Adjusted R-squared: 0.5823
 F-statistic: 19.82 on 2 and 25 DF, p-value: 6.972e-06

Il devient évident que si on corrige la richesse spécifique pour la taille des marais, une fraction importante de la variabilité résiduelle peut être associée à `swamp`, selon une relation quadratique. Si vous examinez les résidus, vous observerez que l'ajustement est nettement meilleur qu'avec le modèle linéaire.

 En vous basant sur les résultats de la dernière analyse, comment suggérez-vous de modifier le modèle de régression multiple? Quel est, d'après vous, le meilleur modèle? Pourquoi? Ordonnez les différents facteurs en ordre croissant de leur effet sur la richesse spécifique des reptiles.

Suite à ces analyses, il semble opportun d'essayer d'ajuster un modèle incluant `logarea`, `thtdden`, `cpfor2`, `swamp` et `swamp^2`:

Call:

```
lm(formula = logherp ~ logarea + cpfor2 + thtdden + swamp + I(swamp^2),
    data = mydata)
```

```

Residuals:
    Min       1Q   Median       3Q      Max
-0.201797 -0.056170 -0.002072  0.051814  0.205626

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.203e-01  1.813e-01  -1.766   0.0912 .
logarea      2.202e-01  3.893e-02   5.656  1.09e-05 ***
cpfor2       -7.864e-04  9.955e-04  -0.790   0.4380
thtden       -2.929e-02  1.048e-02  -2.795   0.0106 *
swamp        3.113e-02  5.898e-03   5.277  2.70e-05 ***
I(swamp^2)   -2.618e-04  4.727e-05  -5.538  1.45e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1072 on 22 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.8181,    Adjusted R-squared: 0.7767
F-statistic: 19.78 on 5 and 22 DF, p-value: 1.774e-07
the p-value is 1.774e-007
2 observations deleted due to missing values

```

Les résultats de cette analyse suggèrent qu'on devrait probablement exclure `cpfor2` du modèle:

```

Call:
lm(formula = logherp ~ logarea + thtden + swamp + I(swamp^2),
    data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.19621 -0.05444 -0.01202  0.07116  0.21295

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.461e-01  1.769e-01  -1.957   0.0626 .
logarea      2.232e-01  3.842e-02   5.810  6.40e-06 ***
thtden       -2.570e-02  9.364e-03  -2.744   0.0116 *
swamp        2.956e-02  5.510e-03   5.365  1.89e-05 ***
I(swamp^2)   -2.491e-04  4.409e-05  -5.649  9.46e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1063 on 23 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.8129,    Adjusted R-squared: 0.7804
F-statistic: 24.98 on 4 and 23 DF, p-value: 4.405e-08

```

Est-ce qu'il y a possiblement un problème de multicollinéarité?

```

> vif(model.polynomial.reduced)
   logarea   thtden    swamp I(swamp^2)
  1.053193  1.123491 45.845845 45.656453

```

Les valeurs d'inflation de la variance (VIF) pour les deux termes de swamp sont beaucoup plus élevés que le seuil de 5. Cependant, c'est la norme pour les termes polynomiaux et on ne doit pas s'en préoccuper outre mesure, surtout quand les deux termes sont hautement significatifs dans le modèle. Les fortes valeurs de VIF indiquent que les coefficients pour ces deux termes ne sont pas estimés précisément, mais leur utilisation dans le modèle permet tout de même de faire de bonnes prédictions (i.e. ils décrivent la réponse à swamp).

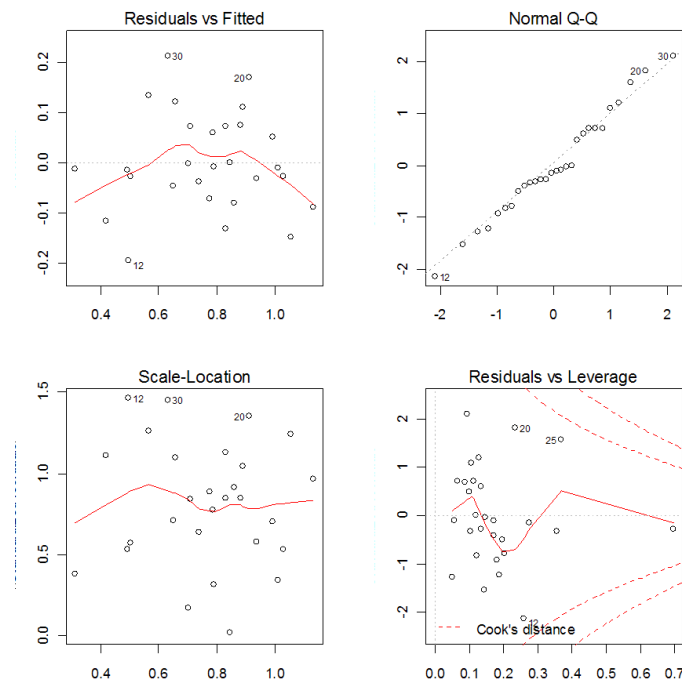
Vérifier les conditions d'application de modèles de régression multiple

Toutes les techniques de sélection des modèles présument que les conditions d'applications (indépendance, normalité, homoscédasticité, linéarité) sont remplies. Comme il y a un grand nombre de modèles qui peuvent être ajustés, il peut paraître quasi impossible de vérifier si les conditions sont remplies à chaque étape de construction.

Cependant, il est souvent suffisant d'examiner les résidus du modèle complet (saturé) puis du modèle final. Les termes qui ne contribuent pas significativement à l'ajustement n'affectent pas beaucoup les résidus et donc les résidus du modèle final sont généralement similaires à ceux du modèle complet.

Examinons donc les graphiques diagnostiques du modèle final:

Figure 4.



Tout semble acceptable dans ce cas. Pour convaincre les sceptiques, on peut faire les tests formels des conditions d'application:

```
shapiro.test(residuals(model.polynomial.reduced))
```

Shapiro-Wilk normality test

```
data: residuals(model.polynomial.reduced)
W = 0.9837, p-value = 0.9278
```

Les résidus ne dévient pas significativement de la normalité. Bien. >

```
> #Homoscedasticity
> #Homoscedasticity
```

```
> bptest(model.polynomial.reduced)
```

```
studentized Breusch-Pagan test
```

```
data: model.polynomial.reduced
BP = 3.8415, df = 4, p-value = 0.4279
```

Pas de déviation d'homoscédasticité non plus. Bien.

```
> #Serial autocorrelation
> dwtest(model.polynomial.reduced)
```

```
Durbin-Watson test
```

```
data: model.polynomial.reduced
DW = 1.725, p-value = 0.2095
```

alternative hypothesis: true autocorrelation is greater than 0

Pas de corrélation sérielle des résidus, donc pas d'évidence de non-indépendance.

```
#Linearity
resettest(model.polynomial.reduced, type = "regressor", data = mydata)
```

```
RESET test
```

```
data: model.polynomial.reduced
RESET = 0.9823, df1 = 8, df2 = 15, p-value = 0.4859
```

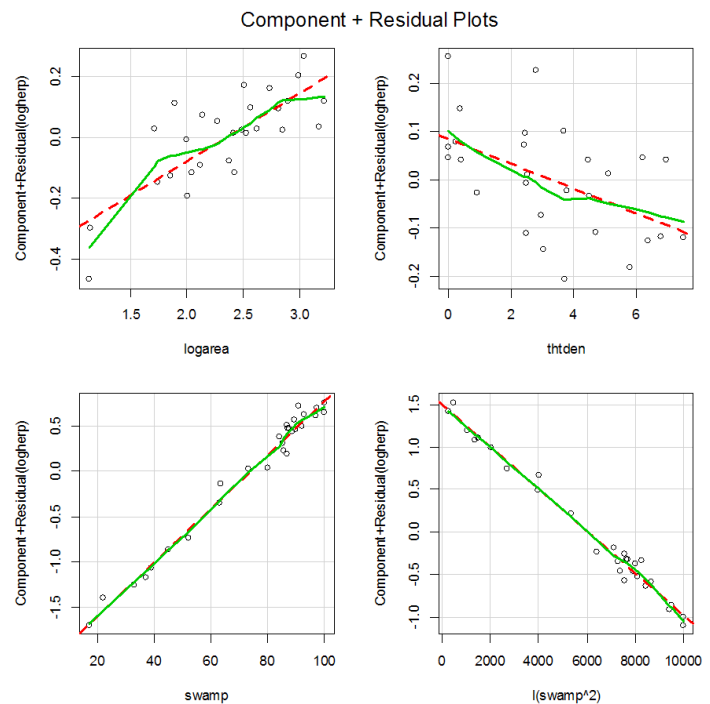
Et finalement, pas de déviation significative de normalité. Donc tout semble acceptable.

Visualiser la taille d'effet

Les coefficients de la régression multiple peuvent mesurer la taille d'effet, quoiqu'il puisse être nécessaire de les standardiser pour qu'ils ne soient pas influencés par les unités de mesure. Mais un graphique est souvent plus informatif. Dans ce contexte, les graphiques des résidus partiels (appelés components+residual plots dans R) sont particulièrement utiles. Ces graphiques illustrent comment la variable dépendante, corrigée pour l'effet des autres variables dans le modèle, varie avec chacune des variables indépendantes du modèle. Voyons voir:

```
# Evaluate visually linearity and effect size
# component + residual plot
cr.plots(model.polynomial.reduced, one.page=TRUE, ask=FALSE)
```

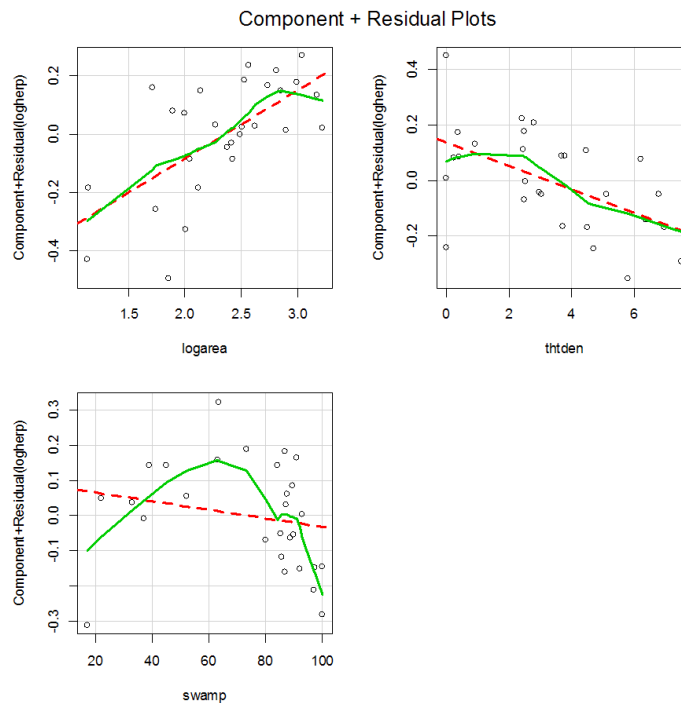
Figure 5.



Notez que l'échelle de l'axe des y varie sur chaque graphique. Pour thtden, la variable dépendante ($\log_{10}(\text{richesse des herptiles})$) varie d'environ 0.4 unités entre la valeur minimum et maximum de thtden. Pour logarea, la variation est d'environ 0.6 unité log. Pour swamp, l'interprétation est plus compliquée parce qu'il y a deux termes qui quantifient son effet, et que ces termes ont des signes opposés (positif pour swamp et négatif pour swamp^2) ce qui donne une relation curvilinéaire de type parabole. Le graphique ne permet pas de bien visualiser cela. Ceci dit, ces graphiques n'indiquent pas vraiment de violation de linéarité.

Pour illustrer ce qui serait visible sur ces graphiques si il y avait une déviation de linéarité, enlevons le terme du second degré pour swamp, puis on va refaire ces graphiques et effectuer le test RESET.

Figure 6.



La relation non-linéaire avec swamp devient évidente. Et le test RESET détecte bien cette non-linéarité:

RESET test

```
data: g
RESET = 6.7588, df1 = 6, df2 = 18, p-value = 0.0007066
```

Tester la présence d'interactions

Lorsqu'il y a plusieurs variables indépendantes, vous devriez toujours garder à l'esprit la possibilité d'interactions. Dans la majorité des situations de régression multiple cela n'est pas évident parce que l'addition de termes d'interaction augmente la multicollinéarité des termes du modèle, et parce qu'il n'y a souvent pas assez d'observations pour éprouver toutes les interactions ou que les observations ne sont pas suffisamment balancées pour faire des tests puissants pour les interactions.

Retournons à notre modèle "final" et voyons ce qui se passe si on essaie d'ajuster un modèle saturé avec toutes les interactions:

```
fullmodel.withinteractions<-lm(logherp ~ logarea * cpfor2 *
thtden * swamp * I(swamp^2), data= Mregdat)

summary(fullmodel.withinteractions)
```

```

Call:
lm(formula = logherp ~ logarea * cpfor2 * thtden * swamp * I(swamp^2),
    data = Mregdat)

Residuals:
ALL 28 residuals are 0: no residual degrees of freedom!

Coefficients: (4 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    -5.948e+03      NA      NA      NA
logarea         3.293e+03      NA      NA      NA
cpfor2          7.080e+01      NA      NA      NA
thtden          9.223e+02      NA      NA      NA
swamp           1.176e+02      NA      NA      NA
I(swamp^2)      -3.517e-01      NA      NA      NA
logarea:cpfor2  -3.771e+01      NA      NA      NA
logarea:thtden  -4.781e+02      NA      NA      NA
cpfor2:thtden   -1.115e+01      NA      NA      NA
logarea:swamp   -7.876e+01      NA      NA      NA
cpfor2:swamp    -1.401e+00      NA      NA      NA
thtden:swamp    -1.920e+01      NA      NA      NA
logarea:I(swamp^2)  5.105e-01      NA      NA      NA
cpfor2:I(swamp^2)  3.825e-03      NA      NA      NA
thtden:I(swamp^2)  7.826e-02      NA      NA      NA
swamp:I(swamp^2)  -2.455e-03      NA      NA      NA
logarea:cpfor2:thtden  5.359e+00      NA      NA      NA
logarea:cpfor2:swamp  8.743e-01      NA      NA      NA
logarea:thtden:swamp  1.080e+01      NA      NA      NA
cpfor2:thtden:swamp  2.620e-01      NA      NA      NA
logarea:cpfor2:I(swamp^2) -5.065e-03      NA      NA      NA
logarea:thtden:I(swamp^2) -6.125e-02      NA      NA      NA
cpfor2:thtden:I(swamp^2) -1.551e-03      NA      NA      NA
logarea:swamp:I(swamp^2) -4.640e-04      NA      NA      NA
cpfor2:swamp:I(swamp^2)  3.352e-05      NA      NA      NA
thtden:swamp:I(swamp^2)  2.439e-04      NA      NA      NA
logarea:cpfor2:thtden:swamp -1.235e-01      NA      NA      NA
logarea:cpfor2:thtden:I(swamp^2)  7.166e-04      NA      NA      NA
logarea:cpfor2:swamp:I(swamp^2)      NA      NA      NA      NA
logarea:thtden:swamp:I(swamp^2)      NA      NA      NA      NA
cpfor2:thtden:swamp:I(swamp^2)      NA      NA      NA      NA
logarea:cpfor2:thtden:swamp:I(swamp^2)      NA      NA      NA      NA

Residual standard error: NaN on 0 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 1, Adjusted R-squared: NaN
F-statistic: NaN on 27 and 0 DF, p-value: NA

```

Notez les coefficients manquants aux dernières lignes: on ne peut inclure les 32 termes si on a seulement 28 observations. Il manque des observations, le R carré est 1, et le modèle "prédit" parfaitement les données.

Si on essaie une méthode automatique pour identifier le "meilleur" modèle dans ce gâchis, R refuse:

```
step(fullmodel.withinteractions)
```

```
Error in step(fullmodel.withinteractions) :
  AIC is -infinity for this model, so 'step' cannot proceed
```

Bon, est-ce qu'on oublie tout ça et qu'on accepte le modèle final sans se soucier des interactions? Non, pas encore. Il y a un compromis possible: comparer notre modèle "final" à un modèle qui contient au moins un sous-ensemble des interactions, par exemple toutes les interactions du second degré, pour éprouver si l'addition de ces interactions améliore beaucoup l'ajustement du modèle.

```
full.model.2ndinteractions<- lm(logherp ~ logarea + cpfor2 +
tthden + swamp + I(swamp^2)
+ logarea:cpfor2 + logarea:tthden + logarea:swamp
+ cpfor2:tthden + cpfor2:swamp
+ tthden:swamp
, data= mydata)
```

```
summary(full.model.2ndinteractions)
```

```
Call:
```

```
lm(formula = logherp ~ logarea + cpfor2 + tthden + swamp + I(swamp^2) +
logarea:cpfor2 + logarea:tthden + logarea:swamp + cpfor2:tthden +
cpfor2:swamp + tthden:swamp, data = mydata)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-0.216880 -0.036534  0.003506  0.042990  0.175490
```

```
Coefficients:
```

```
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.339e-01  6.325e-01   0.686 0.502581
logarea      -1.254e-01  2.684e-01  -0.467 0.646654
cpfor2       -9.344e-03  7.205e-03  -1.297 0.213032
tthden       -1.833e-01  9.035e-02  -2.028 0.059504 .
swamp        3.569e-02  7.861e-03   4.540 0.000334 ***
I(swamp^2)   -3.090e-04  7.109e-05  -4.347 0.000500 ***
logarea:cpfor2 2.582e-03  2.577e-03   1.002 0.331132
logarea:tthden 7.017e-02  3.359e-02   2.089 0.053036 .
logarea:swamp -5.290e-04  2.249e-03  -0.235 0.816981
cpfor2:tthden -2.095e-04  6.120e-04  -0.342 0.736544
cpfor2:swamp  4.651e-05  5.431e-05   0.856 0.404390
tthden:swamp  2.248e-04  4.764e-04   0.472 0.643336
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.108 on 16 degrees of freedom
```

```
(2 observations deleted due to missingness)
```

```
Multiple R-squared: 0.8658, Adjusted R-squared: 0.7735
```

```
F-statistic: 9.382 on 11 and 16 DF, p-value: 4.829e-05
```

Ce modèle s'ajuste un peu mieux aux données que le modèle "final" (il explique 86.6% de la variance de logherp, comparé à 81.2% pour le modèle "final" sans interactions), mais il compte deux fois plus de paramètres. De plus, si vous examinez les coefficients, il se passe d'étranges choses: le signe pour logarea a changé par exemple. C'est un des symptômes de la multicolinéarité. Allons voir les facteurs d'inflation de la variance:

```
vif(full.model.2ndinteractions)
```

```
      logarea      cpfor2      tthden      swamp      I(swamp^2)
49.86060      78.49622     101.42437     90.47389     115.08457
logarea:cpfor2 logarea:tthden logarea:swamp cpfor2:tthden cpfor2:swamp
66.97792      71.69894      67.27034     14.66814      29.41422
tthden:swamp
20.04410
```

Aïe! tous les VIF sont plus grands que 5, pas seulement les termes incluant swamp. Cette forte multicolinéarité empêche de quantifier avec précision l'effet de ces interactions.

De plus, ce modèle avec interactions n'est pas plus informatif que le modèle "final" puisque son AIC est plus élevé (souvenez-vous qu'on privilégie le modèle avec la valeur d'AIC la plus basse):

```
> AIC(full.model)
[1] -38.3433
> AIC(full.model.2ndinteractions)
[1] -34.86123
```

On peut également utiliser la fonction `anova()` pour comparer l'ajustement des deux modèles et vérifier si l'addition des termes d'interaction améliore significativement l'ajustement:

```
anova(full.model, full.model.2ndinteractions)
Analysis of Variance Table

Model 1: logherp ~ logarea + cpfor2 + thtden + swamp + I(swamp^2)
Model 2: logherp ~ logarea + cpfor2 + thtden + swamp + I(swamp^2) + loga-
rea:cpfor2 +
logarea:thtden + logarea:swamp + cpfor2:thtden + cpfor2:swamp +
thtden:swamp
   Res.Df    RSS Df Sum of Sq    F Pr(>F)
1       22 0.252820
2       16 0.186507   6  0.066314 0.9481 0.489
```

Ici, l'addition des termes d'interaction ne réduit pas significativement la variabilité résiduelle du modèle "complet". Qu'en est-il de la si on compare le modèle avec interaction et notre modèle "final"?

```
anova(model.polynomial.reduced, full.model.2ndinteractions)
Analysis of Variance Table

Model 1: logherp ~ logarea + thtden + swamp + I(swamp^2)
Model 2: logherp ~ logarea + cpfor2 + thtden + swamp + I(swamp^2) + loga-
rea:cpfor2 +
logarea:thtden + logarea:swamp + cpfor2:thtden + cpfor2:swamp +
thtden:swamp
   Res.Df    RSS Df Sum of Sq    F Pr(>F)
1       23 0.259992
2       16 0.186507   7  0.073486 0.9006 0.5294
```

Le test indique que ces deux modèles ont des variances résiduelles comparables, et donc que l'addition des termes d'interaction et de `cpfor2` au modèle final n'apporte pas grand chose.

Recherche du meilleur modèle fondée sur la théorie de l'information

Une des principales critiques des méthodes pas-à-pas (stepwise) est que les p-valeurs ne sont pas strictement interprétables à cause du grand nombre de tests qui sont implicites dans le processus. C'est le problème des comparaisons ou tests multiples: en construisant un modèle linéaire (comme une régression multiple) à partir d'un grand nombre de variables et de leurs interactions, il y a tellement de combinaisons possibles qu'un ajustement de Bonferroni rendrait les tests trop conservateurs.

Une alternative, élégamment défendue par Burnham et Anderson (2002, Model selection and multimodel inference: a practical information-theoretic approach. 2nd ed), est d'utiliser l'AIC (ou

mieux encore AICc qui est plus approprié quand le nombre d'observations est inférieur à 40 fois le nombre de variables indépendantes) pour ordonner les modèles et identifier un sous-ensemble de modèles qui sont les meilleurs. On peut ensuite calculer les moyennes des coefficients pondérées par la probabilité que chacun des modèles soit le meilleur pour obtenir des coefficients qui sont plus robustes et moins sensibles à la multicolinéarité.

Cette approche a été suivie dans le package MuMIn.

```
library(MuMIn)
dd <- dredge(full.model.2ndinteractions)
```

L'objet dd contient tous les modèles possibles (i.e. ceux qui ont toutes les combinaisons possibles) en utilisant les termes du modèle full.model.2ndinteractions ajusté précédemment.

On peut ensuite extraire de l'objet dd le sous-ensemble de modèles qui ont un AICc semblable au meilleur modèle (Burnham et Anderson suggèrent que les modèles qui dévient par plus de 4 unités d'AICc du meilleur modèle ont peu de support empirique)

```
# get models within 4 units of AICc from the best model
top.models.1 <- get.models(dd, subset = delta < 4)
avgmodel1 <- model.avg(top.models.1) # compute average parameters
summary(avgmodel1) #display averaged model
confint(avgmodel1) #display CI for averaged coefficients
```

```
Call:
model.avg.default(object = top.models.1)
```

Component models: ①

	df	logLik	AICc	Delta	Weight	②
23457	7	27.78	-35.95	0.00	0.34	
2345	6	25.78	-35.56	0.39	0.28	
123457	8	28.30	-33.02	2.93	0.08	
234578	8	28.26	-32.95	3.00	0.08	
12345	7	26.17	-32.74	3.21	0.07	
23458	7	26.06	-32.51	3.44	0.06	
234567	8	27.88	-32.17	3.78	0.05	
23456	7	25.79	-31.99	3.97	0.05	

Term codes:

	cpfor2	I(swamp^2)	logarea	swamp	thtden
	1	2	3	4	5
logarea:swamp	6	logarea:thtden	7	swamp:thtden	8

Averaged model parameters: ③

	Coefficient	Variance	SE	Unconditional	SE	Lower CI
Model-averaged coefficients:	Estimate	Std. Error	Adjusted SE	z value	Pr(> z)	
(Intercept)	-2.075e-01	2.484e-01	2.593e-01	0.800	0.4236	
logarea	1.314e-01	1.185e-01	1.222e-01	1.076	0.2820	
swamp	3.193e-02	6.125e-03	6.438e-03	4.960	7e-07 ***	
I(swamp^2)	-2.676e-04	4.904e-05	5.154e-05	5.193	2e-07 ***	
thtden	-6.843e-02	5.324e-02	5.459e-02	1.254	0.2100	
logarea:thtden	3.924e-02	2.125e-02	2.251e-02	1.743	0.0813 .	

```

cpfor2      -8.187e-04  9.692e-04  1.027e-03  0.797  0.4253
swamp:thtden -2.402e-04  3.127e-04  3.313e-04  0.725  0.4684
logarea:swamp 4.462e-04  1.664e-03  1.762e-03  0.253  0.8001
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Full model-averaged coefficients (with shrinkage):
(Intercept)      logarea      swamp  I(swamp^2)      thtden logarea:thtden
-2.0751e-01  1.3143e-01  3.1935e-02 -2.6765e-04 -6.8432e-02  2.1388e-02
cpfor2 swamp:thtden logarea:swamp
-1.2017e-04 -3.2771e-05  4.3777e-05

Relative variable importance:
      I(swamp^2)      logarea      swamp      thtden logarea:thtden
      1.00      1.00      1.00      1.00      0.55
      cpfor2 swamp:thtden logarea:swamp
      0.15      0.14      0.10

      2.5 %      97.5 %
(Intercept) -0.7157333646  0.3007147516
logarea      -0.1080048582  0.3708612563
swamp        0.0193158426  0.0445532538
I(swamp^2)   -0.0003686653 -0.0001666418
thtden       -0.1754184849  0.0385545120
logarea:thtden -0.0048800385  0.0833595106
cpfor2       -0.0028313465  0.0011940283
swamp:thtden -0.0008894138  0.0004090457
logarea:swamp -0.0030067733  0.0038991294

```

- ❶ La liste des modèles qui sont à 4 unités ou moins de l'AICc du meilleur modèle. Les variables dans chaque modèle sont codées et on retrouve la clé en dessous du tableau.
- ❷ Pour chaque modèle, en plus de l'AICc, le poids Akaike est calculé. C'est un estimé de la probabilité que ce modèle est le meilleur. Ici on voit que le premier modèle (le meilleur) a seulement 34% des chances d'être vraiment le meilleur.
- ❸ À partir de ce sous-ensemble de modèles, la moyenne pondérée des coefficients (en utilisant les poids Akaike) est calculée, avec in IC à 95%. Notez que les termes absents d'un modèle sont considérés avoir un coefficient de 0 pour ce terme.

Bootstrapping multiple regression

Quand les données ne rencontrent pas les conditions d'application de normalité et d'homoscédasticité et que les transformations n'arrivent pas à corriger ces violations, le bootstrap peut être utilisé pour calculer des intervalles de confiance pour les coefficients. Si la distribution des coefficients bootstrappés est symétrique et approximativement normale, on peut utiliser les percentiles empiriques pour calculer les limites de confiance.

Le code qui suit, utilisant le package `simpleboot`, a été conçu pour être facilement modifiable et calcule les limites des IC à partir des percentiles empiriques.

```
#####
#####
# Bootstrap analysis the simple way with library simpleboot

# Define model to be bootstrapped and the data source used
mymodel<-lm(logherp ~ logarea + thtden + swamp + I(swamp^2),
data = mydata)

# Set the number of bootstrap iterations
nboot<-1000

require(simpleboot)

# R is the number of bootstrap iterations
# Setting rows to FALSE indicates resampling of residuals
mysimpleboot<-lm.boot(mymodel, R = nboot, rows = FALSE)

# Extract bootstrap coefficients
myresults<-sapply(mysimpleboot$boot.list, function (x)
x$coef)

# Transpose matrix so that lines are bootstrap iterations
and columns are coefficients
tmyresults<-t(myresults)

# Plot histograms of bootstrapped coefficients

ncoefs<-length(data.frame(tmyresults))

par(mfrow=c(2,1), mai=c(0.5, 0.5, 0.5, 0.5), ask = TRUE)

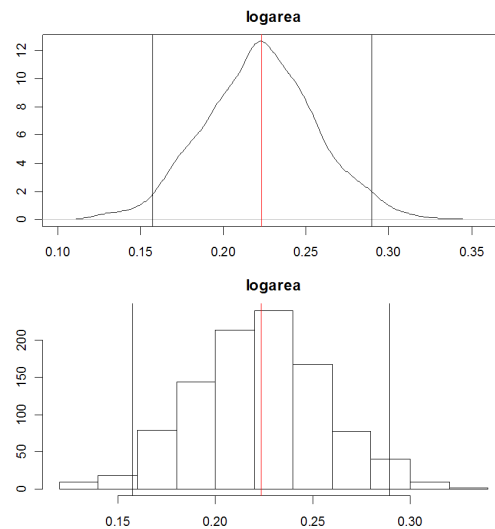
for (i in 1 : ncoefs) {
  lab<-colnames(tmyresults)[i]
  x<-tmyresults[,i]

  plot(density(x),main=lab, xlab="")
  abline(v=mymodel$coef[i], col = "red")
  abline(v=quantile(x, c(0.025, 0.975)))
  hist(x, main=lab, xlab="")
  abline(v=quantile(x, c(0.025, 0.975)))
  abline(v=mymodel$coef[i], col = "red")
}

par(mfrow=c(1,1), ask = FALSE)
```

Lorsque vous tournerez ce code, il y aura une pause pour vous permettre d'examiner la distribution pour chaque coefficient du modèle sur des graphiques:

Figure 7.



Le graphique du haut illustre la densité lissée (kernel density) et celui du bas est l'histogramme des estimés bootstrap du coefficient. La ligne rouge sur le graphique indique la valeur du coefficient ordinaire (pas bootstrap) et les deux lignes verticales noires marquent les limites de l'intervalle de confiance à 95%. Ici l'IC ne contient pas 0, et donc on peut conclure que l'effet de logarea sur logherp est significativement positif.

Les limites précises peuvent être obtenues par:

```
> # Display empirical bootstrap quantiles (not corrected
for bias)
> p <- c(0.005,0.01, 0.025, 0.05, 0.95, 0.975, 0.99, 0.995)
> apply(tmyresults, 2, quantile, p)
      (Intercept) logarea thtden swamp I(swamp^2)
0.5% -0.735605927 0.1321902 -0.048153235 0.01794482 -0.0003529401
1% -0.706786582 0.1444830 -0.045895772 0.01886999 -0.0003384226
2.5% -0.644649399 0.1575567 -0.043016668 0.02015046 -0.0003263934
5% -0.600986378 0.1679613 -0.039140810 0.02166009 -0.0003152855
95% -0.075274798 0.2801058 -0.012076119 0.03777323 -0.0001859188
97.5% -0.035536091 0.2894439 -0.009660800 0.03929003 -0.0001715579
99% -0.004244384 0.3014986 -0.007446500 0.04091320 -0.0001612558
99.5% 0.031621494 0.3095612 -0.006314632 0.04301388 -0.0001566147
```

Ces intervalles de confiances ne sont pas fiables si la distribution des estimés bootstrap n'est pas Gaussienne. Dans ce cas, il vaut mieux calculer des coefficients non-biaisés (bias-corrected accelerated confidence limits, BCa):

```
#####
# Bootstrap analysis in multiple regression with BCa confi-
dence intervals
# Preferable when parameter distribution is far from normal

# Bootstrap 95% BCa CI for regression coefficients
```



```

library(boot)

# function to obtain regression coefficients for each iteration
bs <- function(formula, data, indices) {
  d <- data[indices,] # allows boot to select sample
  fit <- lm(formula, data=d)
  return(coef(fit))
}

# bootstrapping with 1000 replications
results <- boot(data=mydata, statistic=bs,
  R=1000, formula=logherp ~ logarea + thtden + swamp +
  I(swamp^2))

# view results
results
plot(results, index=1) # intercept
plot(results, index=2) # logarea
plot(results, index=3) # thtden
plot(results, index=4) # swamp
plot(results, index=5) # swamp^2

# get 95% confidence intervals
boot.ci(results, type="bca", index=1) # intercept
boot.ci(results, type="bca", index=2) # logarea
boot.ci(results, type="bca", index=3) # thtden
boot.ci(results, type="bca", index=4) # swamp
boot.ci(results, type="bca", index=5) # swamp^2

results

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = mydata, statistic = bs, R = 1000, formula = logherp ~
  logarea + thtden + swamp + I(swamp^2))

Bootstrap Statistics :
      original      bias    std. error
t1*  -0.3460988043  2.613804e-04  2.346177e-01
t2*   0.2232323394 -5.971629e-03  5.222778e-02
t3*  -0.0256957141 -3.946859e-04  9.411536e-03
t4*   0.0295628938  4.057821e-04  7.148861e-03
t5*  -0.0002490924 -2.438465e-06  5.341945e-05

```

Ce code va produire le graphique standard pour chaque coefficient, et les estimés BCa pour l'intervalle de confiance. Pour logarea, cela donne:

```

> boot.ci(results, type="bca", index=2) # logarea
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 2)

```

```
Intervals :
Level      BCa
95%      ( 0.1212,  0.3225 )
Calculations and Intervals on Original Scale
```

Notez que l'intervalle BCa va de 0.12 à 0.32, alors que l'intervalle standard était de 0.16 à 0.29. L'intervalle BCa est ici plus grand du côté inférieur et plus petit du côté supérieur comme il se doit compte tenu de la distribution non-Gaussienne et asymétrique des estimés bootstrap. i

Test de permutation

Les tests de permutations sont plus rarement effectués en régression multiple que le bootstrap. Voici un fragment de code pour le faire tout de même.

```
#####
#####
#   Permutation in multiple regression
#
#   using lperm library

require(lmPerm)

#   Fit desired model on the desired dataframe
mymodel<-lm(logherp ~ logarea + thtden + swamp + I(swamp^2) ,
data = mydata)

mymodelProb<-lmp(logherp ~ logarea + thtden + swamp +
I(swamp^2) , data = mydata, perm="Prob")

summary(mymodel)
summary(mymodelProb)
```


Labo- ANCOVA et GLM

Après avoir complété cet exercice de laboratoire, vous devriez pouvoir:

- Utiliser R pour faire une analyse de covariance (ANCOVA) et ajuster des modèles qui ont des variables indépendantes continues et discontinues (GLM)
- Utiliser R pour vérifier les conditions préalables à l'ANCOVA ou aux GLM
- Utiliser R pour comparer l'ajustement de modèles statistiques
- Utiliser R pour faire des tests de bootstrap et de permutation sur des modèles GLM avec des variables indépendantes continues et discontinues.

GLM

GLM est une abréviation pour General Linear Model. Les GLM sont des modèles statistiques de la forme $Y=XB+E$, où Y est un vecteur (ou matrice) contenant la variable dépendante, B est une matrice de paramètres estimés, et E est un vecteur (ou matrice) de résidus homoscédastiques et normalement distribués. Tous les tests que nous avons étudiés à date (test de t, régression linéaire simple, ANOVA à un facteur de classification, ANOVA à plusieurs facteurs de classification et régression multiple) sont des GLMs. Notez que tous les modèles que nous avons rencontrés à ce jour ne contiennent qu'un type de variable indépendante (soit continue ou discontinue). Dans cet exercice de laboratoire, vous allez ajuster des modèles qui ont les deux types de variables indépendantes.

ANCOVA

ANCOVA est l'abréviation pour l'analyse de covariance. C'est un type de GLM dans lequel il y a une (ou plusieurs) variable indépendante continue (parfois appelé la covariable) et une (ou plusieurs) variable indépendante discontinue. Dans la présentation traditionnelle de l'ANCOVA dans les manuels de biostatistique, le modèle ANCOVA ne contient pas de termes d'interaction entre les variables continues et discontinues. Par conséquent, on doit précéder l'ajustement de ce modèle (réduit parce que sans terme d'interaction), par un test de signification de l'interaction qui correspond à éprouver l'égalité des

pentés (coefficients pour la ou les variables continues) entre les différents niveaux de la ou les variables discontinues (i.e un test d'homogénéité des pentés).


Certaines personnes, moi y compris, utilisons le terme ANVCOVA dans un sens plus large, pour décrire tous les modèles GLM qui contiennent à la fois des variables continues et discontinues. Soyez avertis toutefois que, selon l'auteur, ANCOVA peut correspondre à un modèle GLM avec ou sans interaction.

Homogénéité des pentés

Pour répondre à de nombreuses questions biologiques, il est nécessaire de déterminer si deux (ou plus de deux) régressions diffèrent significativement. Par exemple, pour comparer l'efficacité de deux insecticides on doit comparer la relation entre leur dose et la mortalité. Ou encore, pour comparer le taux de croissance des mâles et des femelles on doit comparer la relation entre la taille et l'âge des mâles et des femelles.

Comme chaque régression linéaire est décrite par deux paramètres, la pente et l'ordonnée à l'origine, on doit considérer les deux dans la comparaison. Le modèle d'ANCOVA, à strictement parler, n'éprouve que l'hypothèse d'égalité des ordonnées à l'origine. Cependant, avant d'ajuster ce modèle, il faut éprouver l'hypothèse d'égalité des pentés (homogénéité des pentés).

Cas 1 - La taille en fonction de l'âge (exemple avec pente commune)

 En utilisant les données du fichier `anc1dat.Rdata`, éprouvez l'hypothèse que le taux de croissance des esturgeons mâles et femelles de The Pas est le même (données de 1978-1980). Comme mesure du taux de croissance, nous allons utiliser la pente de la régression du \log_{10} de la longueur à la fourche, `lfkl`, sur le \log_{10} de l'âge, `lage`

Commençons par examiner les données. Pour faciliter la comparaison, il serait utile de tracer la droite de régression et la trace lowess pour ainsi plus facilement évaluer la linéarité. On peut aussi ajouter un peu de trucs R pour obtenir des légendes plus complètes (remarquez l'utilisation de la commande `expression()` pour obtenir des indices):

```
mydata<-anc1dat
require(ggplot2)
myplot<-ggplot(data=mydata, aes(x=lage,
y=log10(fklngth)))+facet_grid(.~sex)+geom_point()
myplot
myplot<-myplot+
```

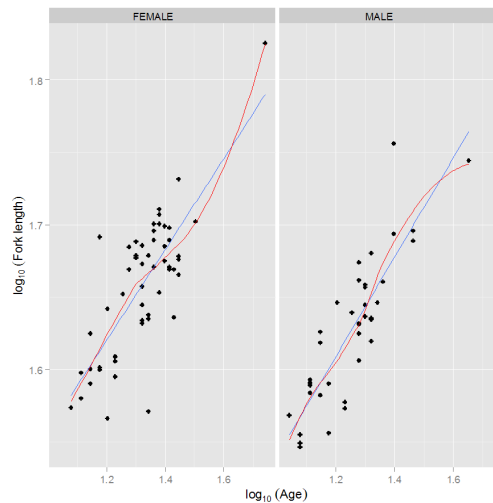
```

stat_smooth(method = lm, se=FALSE)+
stat_smooth(se=FALSE, color="red")

update_labels(myplot, list(
  y = expression(log[10]~(Fork~length)) ,
  x = expression(log[10]~(Age))
))

```

Figure 1.



La transformation log-log rend la relation linéaire et, à première vue, il ne semble pas y avoir de problème évident avec les conditions d'application. Ajoutons donc le modèle complet avec l'interaction:

```

require(car)
model.full<-lm(lfkl ~ sex + lage + sex:lage, data = mydata)
Anova(model.full, type = 3)

```

Anova Table (Type III tests)

Response: lfkl

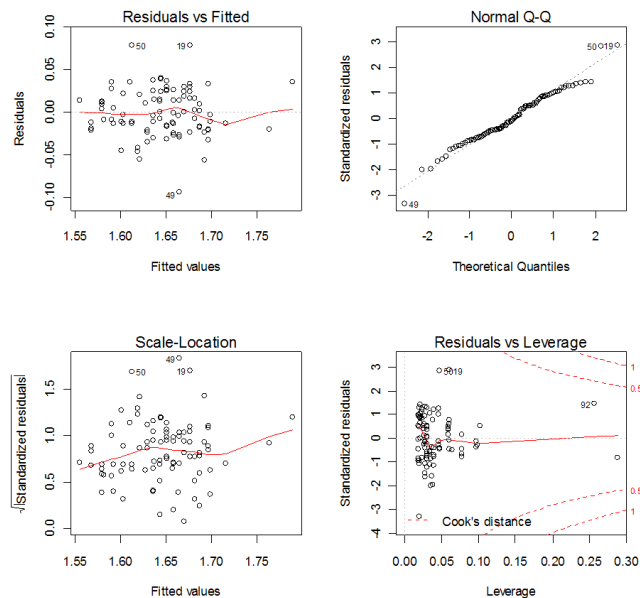
	Sum Sq	Df	F value	Pr(>F)
(Intercept)	0.64444	1	794.8182	< 2.2e-16 ***
sex	0.00041	1	0.5043	0.4795
lage	0.07259	1	89.5312	4.588e-15 ***
sex:lage	0.00027	1	0.3367	0.5632 1
Residuals	0.07135	88		

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
 Residuals 88 0.0713501 0.0008108

1 Probabilité que le terme $lage \times sex$ n'affecte pas la longueur à la fourche (i.e. que la pente ne diffère pas entre les sexes, et que la différence de taille entre les mâles et femelles ne varie pas avec l'âge)

Attrape. Notez que j'ai utilisé la fonction `Anova()` du package `car` avec un "a" majuscule au lieu de la fonction native `anova()` (avec un "a" minuscule") associée aux objets produits par `lm()` pour obtenir les sommes de carrés de type III. Ces sommes des carrés des écarts de type III (partiels) sont calculées comme si la variable était entrée la dernière dans le modèle et correspondent à la différence entre la variance expliquée par le modèle complet et par le modèle dans lequel seule cette variable est omise. La fonction native `anova()` donne les sommes des carrés séquentielles, calculées au fur et à mesure que chaque variable est ajoutée au modèle nul avec seulement une ordonnée à l'origine. Dans de rares cas, les sommes des carrés de type I et III sont égales (quand le design est parfaitement orthogonal ou balancé). Dans la vaste majorité des cas, les sommes des carrés de type I et III sont différentes et je vous conseille de toujours utiliser les sommes des carrés de type III dans vos analyses.

À partir de cette analyse, on devrait accepter les hypothèses nulles (1) d'égalité des pentes pour les deux sexes, et (2) que les ordonnées à l'origine sont les mêmes pour les deux sexes. Mais, avant d'accepter ces conclusions, il faut vérifier si les données rencontrent les conditions d'application, comme d'habitude...



En ce qui concerne la normalité, ça a l'air d'aller quoiqu'il y a quelques points, en haut à droite, qui dévient de la droite. Si on effectue le test de Wilk-Shapiro ($W = .9764$, $p = 0.09329$), on confirme que les résidus ne dévient pas significativement de la normalité. Les résidus semblent

homoscédastiques, mais si vous voulez vous en assurer, vous pouvez l'éprouver par un des tests formels. Ici j'utilise le test Breusch-Pagan, qui est approprié quand certaines des variables indépendantes sont continues (Le test de Levene n'est approprié que lorsqu'il n'y a que des variables discontinues).

```
bptest(model.full)
```

qui donne:

```
studentized Breusch-Pagan test

data: model.full
BP = 0.99979, df = 3, p-value = 0.8013
```

Comme l'hypothèse nulle de ce test est que les résidus sont homoscédastiques, et que p est relativement élevé, le test confirme l'évaluation visuelle.

De plus, il n'y a pas de tendance évidente dans les résidus, suggérant qu'il n'y a pas de problème de linéarité. Ce qui peut également être éprouvé formellement:

```
resettest(model.full, power = 2:3, type = "regressor",
           data = mydata)
RESET test

data: model.full
RESET = 0.5986, df1 = 2, df2 = 86, p-value = 0.5519
```

La dernière condition d'application est qu'il n'y a pas d'erreur de mesure sur la variable indépendante continue. On ne peut vraiment éprouver cette condition,, mais on sait que des estimés indépendants de l'âge des poissons obtenus par différents chercheurs donnent des âges qui concordent avec moins de 1-2 ans d'écart., ce qui est inférieur au 10% de la fourchette observée des âges et donc acceptable pour des analyses de modèles de type I (attention ici on ne parle pas des SC de type I, je sais, c'est facile de confondre...)

Vous noterez qu'il y a une observation qui a un résidu normalisé (studentized residual) qui est élevé, i.e. une valeur extrême (cas numéro 49). Éliminez-la de l'ensemble de données et refaites l'analyse. Vos conclusions changent-elles?

```
model.full.no49<-lm(lfkl ~ sex + lage + sex:lage, data =
mydata[c(-49),])
Anova(model.full.no49, type=3)
```

Anova Table (Type III tests)


```
Response: lfkl
              Sum Sq Df F value Pr(>F)
(Intercept)  0.64255  1 895.9394 <2e-16 ***
sex           0.00038  1   0.5273 0.4697
lage          0.07378  1 102.8746 <2e-16 ***
```



```
sex:lage    0.00022  1    0.3135 0.5770
Residuals   0.06239 87
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La conclusion ne change pas après avoir enlevé la valeur extrême. Comme on n'a pas de bonne raison d'éliminer cette valeur, il est probablement mieux de la conserver. Un examen des conditions d'application après avoir enlevé cette valeur révèle qu'elles sont toutes rencontrées.

Cas 2 - Taille en fonction de l'âge (exemple avec des pentes différentes)

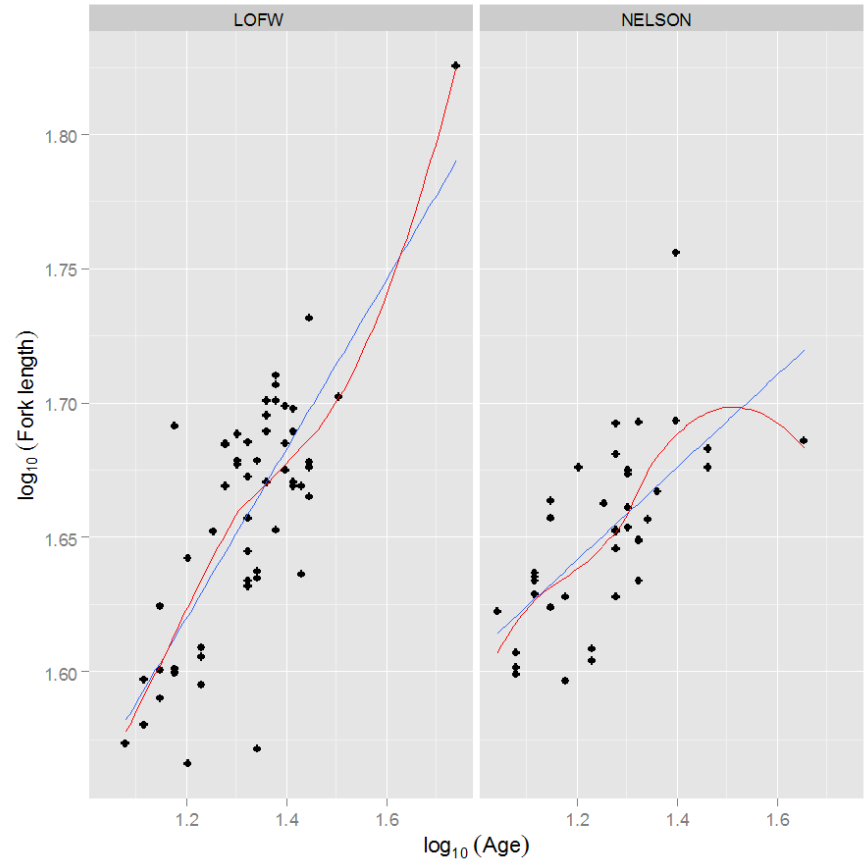
 Le fichier `anc3dat.Rdata` contient des données sur des esturgeons mâles de deux sites (`locate`): Lake of the Woods dans le Nord-Ouest de l'Ontario et Churchill River dans le Nord du Manitoba. En utilisant la même procédure, éprouvez l'hypothèse que la pente de la régression de `lflkl` sur `lage` est la même aux deux sites (alors `locate` est la variable en catégories et non pas `sex`). Que concluez-vous?

```
mydata<-anc3dat

require(ggplot2)
myplot<-ggplot(data=mydata, aes(x=lage,
y=log10(fklngth)))+facet_grid(.~locate)+geom_point()
myplot
myplot<-myplot+
  stat_smooth(method = lm, se=FALSE)+
  stat_smooth(se=FALSE, color="red")

update_labels(myplot, list(
  y = expression(log[10]~(Fork~length)),
  x = expression(log[10]~(Age))
))
```

Figure 2.



```
model.full<-lm(lfkl ~ lage + locate + lage:locate, data =
mydata)
```

```
Anova(model.full, type = 3)
```

Anova Table (Type III tests)

Response: lfkl

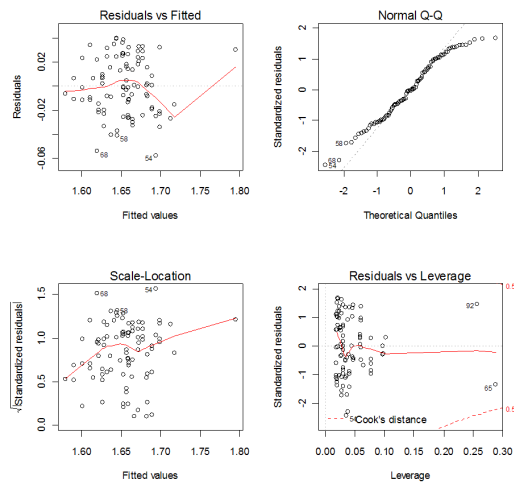
	Sum Sq	Df	F value	Pr(>F)
(Intercept)	0.62951	1	1078.632	< 2.2e-16 ***
lage	0.07773	1	133.185	< 2.2e-16 ***
locate	0.00968	1	16.591	0.0001012 ***
lage:locate	0.00909	1	15.575	0.0001592 ***
Residuals	0.05136	88		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Ici, on rejette les hypothèses nulles (1) que les pentes sont les mêmes dans les deux sites et (2) que les ordonnées à l'origine sont égales. En d'autres mots, si on veut prédire la longueur à la fourche d'un esturgeon à un âge donné précisément, il faut savoir de quel site il provient. Puisque les pentes diffèrent, il faut estimer des régressions séparées.

Mais avant d'accepter ces conclusions, on doit se convaincre que les conditions d'application sont rencontrées:

Figure 3.



Si on examine les résidus selon les méthodes habituelles, on voit qu'il n'y a pas de problème de linéarité, ni d'homoscédasticité ($BP = 2.8721, p = 0.4118$). Cependant, le test de Wilk-Shapiro est significatif ($W=0.97, p = 0.03$). Étant donné la taille assez grande de l'échantillon ($N=92$), ce test a beaucoup de puissance, même si la déviation de normalité ne semble pas très grande. Compte-tenu de la robustesse relative des GLM, de la taille de l'échantillon, on ne devrait pas être trop inquiet de cette déviation de normalité.

Donc, comme les conditions des GLM sont suffisamment remplies, on peut accepter les résultats donnés par R. Tous les termes sont significatifs (location, lage, interaction). Ce modèle complet est équivalent à ajuster des régressions séparées pour chaque site. Pour obtenir les coefficients, on peut ajuster des régressions simples sur chaque sous-ensemble, ou extraire les coefficients ajustés du modèle complet:

model.full

```
Call:
lm(formula = lfk1 ~ lage + locate + lage:locate, data = mydata)
```

Coefficients:

(Intercept)	lage	locateNELSON
1.2284	0.3253	0.2207
lage:locateNELSON		
-0.1656		

Par défaut, la variable `locate` est encodée comme 0 pour le site qui vient le premier en ordre alphabétique (`LofW`) et 1 pour l'autre (`Nelson`). Les régressions pour chaque site deviennent donc:

Pour `LofW`:

$$l_{fkl} = 1.2284 + 0.3253 \text{ lage} + 0.2207 (0) - 0.1656 (0) (\text{lage})$$

$$l_{fkl} = 1.2284 + 0.3253 \text{ lage}$$

Pour Nelson:

$$l_{fkl} = 1.2284 + 0.3253 \text{ lage} + 0.2207 (1) - 0.1656 (1) (\text{lage})$$

$$l_{fkl} = 1.4491 + 0.1597 \text{ lage}$$

Vous pouvez vérifier en ajustant séparément les régressions pour chaque site:

```
by(anc3dat, locate, function(x) lm(lfkl~lage, data=x))
```

```
locate: LOFW
```

```
Call:
```

```
lm(formula = lfkl ~ lage, data = x)
```

```
Coefficients:
```

```
(Intercept)      lage
    1.2284         0.3253
```

```
-----
locate: NELSON
```

```
Call:
```

```
lm(formula = lfkl ~ lage, data = x)
```

```
Coefficients:
```

```
(Intercept)      lage
    1.4491         0.1597
```

Le modèle d'ANCOVA

Si le test d'homogénéité des pentes indique qu'elles diffèrent, alors on devrait estimer des régressions individuelles pour chaque niveau des variables discontinues. Cependant, si on accepte l'hypothèse d'égalité des pentes, l'étape suivante est de comparer les ordonnées à l'origine.

Selon la "vieille école" i.e. l'approche traditionnelle, on ajuste un modèle avec la variable catégorique et la variable continue, mais sans interaction (le modèle ANCOVA sensu stricto) et on utilise la somme des carrés des écarts de type III, disons avec la fonction `Anova()`. C'est ce que la majorité des manuels de biostatistiques présentent

L'autre approche consiste à utiliser les résultats de l'analyse du modèle complet, et tester la signification de chaque terme à partir des sommes des carrés partiels. C'est plus rapide, mais moins puissant. Dans la plupart des cas, cette perte de puissance n'est pas trop préoccupante, sauf lorsque le modèle est très complexe et contient de nombreuses interactions non-significatives. Je vous suggère d'utiliser l'approche simplifiée, et de n'utiliser l'approche traditionnelle que lorsque vous acceptez l'hypothèse d'égalité des ordonnées à l'origine. Pourquoi?

Puisque l'approche simplifiée est moins puissante, si vous rejetez quand même H_0 , alors votre conclusion ne changera pas, mais sera seulement renforcée, en utilisant l'approche traditionnelle.

Ici, je vais comparer l'approche traditionnelle et l'approche simplifiée. Rappelez-vous que vous voulez évaluer l'égalité des ordonnées à l'origine **après avoir déterminé que les pentes étaient égales**. Éprouver l'égalité des ordonnées à l'origine quand les pentes diffèrent (ou, si vous préférez, quand il y a une interaction) est rarement sensé, peut facilement être mal interprété, et ne devrait être effectué que rarement.

De retour aux données de `anc1dat.Rdata`, en comparant la relation entre `lfkl` et `lage` entre les sexes, nous avons obtenu les résultats suivants pour le modèle complet avec interactions:

Anova Table (Type III tests)

```
Response: lfkl
      Sum Sq Df F value    Pr(>F)
(Intercept) 0.64444 1 794.8182 < 2.2e-16 ***
sex          0.00041 1   0.5043   0.4795
lage        0.07259 1  89.5312 4.588e-15 ***
sex:lage     0.00027 1   0.3367   0.5632
Residuals   0.07135 88
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residuals 88 0.0713501 0.0008108
```

On avait déjà conclu que la pente ne varie pas entre les sexes (i.e. l'interaction n'est pas significative). Notez que la p-valeur associée au sexe (0.4795) n'est pas significative non plus.

De l'autre côté, selon l'approche traditionnelle, l'inférence quand à l'effet du sexe se fait à partir du modèle réduit (le modèle ANCOVA sensus stricto):

```
model.ancova<-lm(lfkl ~ sex + lage , data = mydata)
Anova(model.ancova, type = 3)
summary(model.ancova)
```

qui donne:

Anova Table (Type III tests)

```
Response: lfkl
      Sum Sq Df F value    Pr(>F)
(Intercept) 1.13480 1 1410.1232 <2e-16 ***
sex          0.00149 1   1.8513 0.1771
lage        0.14338 1  178.1627 <2e-16 ***
Residuals   0.07162 89
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Call:
lm(formula = lfkl ~ sex + lage, data = mydata)
```

```

Residuals:
    Min       1Q   Median       3Q      Max
-0.0939916 -0.0184574 -0.0008762  0.0224911  0.0811609


Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.225533   0.032636  37.552  <2e-16 ***
sexMALE       -0.008473   0.006228  -1.361   0.177
lage          0.327253   0.024517  13.348  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02837 on 89 degrees of freedom
Multiple R-squared: 0.696,    Adjusted R-squared: 0.6892
F-statistic: 101.9 on 2 and 89 DF,  p-value: < 2.2e-16

```

. Dans ce modèle, sex n'est pas significatif et on conclue donc que l'ordonnée à l'origine ne diffère pas entre les sexes. Notez que la p-valeur est plus petite (0.1771 vs 0.4795), ce qui reflète la puissance accrue de l'approche traditionnelle. Toutefois, les conclusions sont les mêmes: les ordonnées à l'origine ne diffèrent pas.

(

 En examinant les graphiques diagnostiques, vous noterez qu'il y a trois observations dont la valeur absolue du résidu est grande (cas 19, 49, et 50). Ces observations pourraient avoir un effet disproportionné sur les résultats de l'analyse. Éliminez-les et refaites l'analyse. Les conclusions changent-elles ?

```

model.ancova.nooutliers<-lm(lfkl ~ sex + lage ,
                             data = mydata[c(-49, -50, -19),])
Anova(model.ancova.nooutliers, type=3)
summary(model.ancova.nooutliers)

Anova Table (Type III tests)

Response: lfkl
              Sum Sq Df    F value    Pr(>F)
(Intercept)  1.09160  1 1896.5204 < 2e-16 ***
sex           0.00232  1    4.0374 0.04764 *
lage         0.13992  1   243.0946 < 2e-16 ***
Residuals    0.04950 86
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Call:
lm(formula = lfkl ~ sex + lage, data = mydata[c(-49, -50, -19),
])


Residuals:
    Min       1Q   Median       3Q      Max
-0.0583972 -0.0184692 -0.0009756  0.0206964  0.0402875

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.224000   0.028106  43.549  <2e-16 ***
sexMALE       -0.010823   0.005386  -2.009   0.0476 *
lage          0.328604   0.021076  15.591  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
Residual standard error: 0.02399 on 86 degrees of freedom
Multiple R-squared: 0.7706,    Adjusted R-squared: 0.7653
F-statistic: 144.4 on 2 and 86 DF,  p-value: < 2.2e-16
```

Ouch! Les résultats changent. Il faudrait donc rejeter l'hypothèse nulle et conclure que les ordonnées à l'origine diffèrent! Une conclusion qualitativement différente de celle obtenue en considérant toutes les données. Pourquoi? Il y a deux raisons possibles : (1) les valeurs extrêmes influencent beaucoup les régressions ou (2) l'exclusion des valeurs extrêmes permet d'augmenter la puissance de détection d'une différence. La première explication est moins plausible parce que les valeurs extrêmes n'avaient pas une grande influence (leverage faible). Alors, la deuxième explication est plus plausible et vous pouvez le vérifier en faisant des régressions pour chaque sexe sans et avec les valeurs extrêmes. Si vous le faites, vous noterez que les ordonnées à l'origine pour chaque sexe ne changent presque pas alors que leurs erreurs-types changent beaucoup.

 Ajustez une régression simple entre `lfkl` et `lage` pour l'ensemble complet de données et aussi pour le sous-ensemble sans les 3 valeurs déviantes. Comparez ces modèles avec les modèles d'ANCOVA ajustés précédemment. Que concluez-vous ? Quel modèle, d'après vous, a le meilleur ajustement aux données ? Pourquoi ?

Le modèle en excluant les valeurs extrêmes:

```
model.linear.nooutliers<-lm(lfkl ~ lage ,
  data = mydata[c(-49, -50, -19),])
summary(model.linear.nooutliers)
call:
lm(formula = lfkl ~ lage, data = mydata[c(-49, -50, -19), ])

Residuals:
    Min       1Q   Median       3Q      Max
-0.055567 -0.017809 -0.002944  0.021272  0.044972

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.20378    0.02670   45.09  <2e-16 ***
lage         0.34075    0.02054   16.59  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.02441 on 87 degrees of freedom
Multiple R-squared: 0.7598,    Adjusted R-squared: 0.7571
F-statistic: 275.2 on 1 and 87 DF,  p-value: < 2.2e-16
```

Pour la régression simple (sans les valeurs extrêmes) on obtient un R^2 de 0.76 et une erreur-type des résidus de 0.02441, En comparant à l'erreur-type des résidus du modèle d'ANCOVA (0.02399) on réalise que la qualité des prédictions est essentiellement la même, même en ajustant des ordonnées à l'origine différentes pour chaque groupe. Par conséquent, les bénéfices de l'inclusion d'un terme pour les

différentes ordonnées à l'origine sont faibles alors que le coût, en terme de complexité du modèle, est élevé (33% d'augmentation du nombre de termes pour un très faible amélioration de la qualité d'ajustement). Si vous examinez les résidus de ce modèle, vous trouverez qu'ils sont à peu près O.K.)

Si on ajuste une régression simple sur toutes les données, on obtient:

```
model.linear<-lm(lfkl ~ lage , data = mydata)
summary(model.linear)
```

```
Call:
lm(formula = lfkl ~ lage, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.090915 -0.018975 -0.002587  0.021270  0.085273

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.21064    0.03089   39.19  <2e-16 ***
lage         0.33606    0.02376   14.14  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0285 on 90 degrees of freedom
Multiple R-squared:  0.6897,    Adjusted R-squared:  0.6863
F-statistic: 200.1 on 1 and 90 DF,  p-value: < 2.2e-16
```


Encore une fois, l'erreur-type des résidus (0.0285) pour cette régression unique est semblable à la variance du modèle d'ANCOVA (0.02837) et le modèle simplifié prédit presque aussi bien que le modèle plus complexe. (Ici encore, toutes les conditions d'application semblent remplies, si ce n'est de la valeur extrême).

Donc, dans les deux cas (avec ou sans les valeurs extrêmes), l'addition d'un terme supplémentaire pour le sexe n'ajoute pas grand-chose. Il semble donc que le meilleur modèle soit celui de la régression simple. Un estimé raisonnablement précis de la taille des esturgeons peut être obtenu de la régression commune sur l'ensemble des résultats.

Note: Il est fréquent que l'élimination de valeurs extrêmes en fasse apparaître d'autres. C'est parce que ces valeurs extrêmes dépendent de la variabilité résiduelle. Si on élimine les valeurs les plus déviantes, la variabilité résiduelle diminue, et certaines observations qui n'étaient pas si déviantes que cela deviennent proportionnellement plus déviantes. Notez aussi qu'en éliminant des valeurs extrêmes, l'effectif diminue et que la puissance décroît. Il faut donc être prudent.

Comparer l'ajustement de modèles

Comme vous venez de le voir, le processus d'ajustement de modèles est itératif. La plupart du temps il y a plus d'un modèle qui peut être ajusté aux données et c'est à vous de choisir celui qui est le meilleur compromis entre la qualité d'ajustement (qu'on essaie de maximiser) et la complexité (qu'on essaie de minimiser). La stratégie de base en ajustant des modèles linéaires (ANOVA, régression, ANCOVA) est de privilégier le modèle le plus simple si la qualité d'ajustement n'est pas significativement plus mauvaise. R peut calculer une statistique F vous permettant de comparer l'ajustement de deux modèles. Dans ce cas, l'hypothèse nulle est que la qualité d'ajustement ne diffère pas entre les deux modèles.

 En utilisant les données de `Anc1dat` comparez l'ajustement du modèle ANCOVA et de la régression commune:

```
anova(model.ancova,model.linear)
```


Analysis of Variance Table

Model 1: `lfkl ~ sex + lage`

Model 2: `lfkl ~ lage`

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	89	0.071623				
2	90	0.073113	-1	-0.001490	1.8513	0.1771

La fonction `anova()` utilise la différence entre la somme des carrés des deux modèles et la divise par la différence entre le nombre de degrés de liberté pour obtenir un carré moyen. Ce carré moyen est utilisé au numérateur et est divisé par la variance résiduelle du modèle le plus complexe pour obtenir la statistique F. Dans ce cas-ci, le test de F n'est pas significatif, et on conclut que les deux modèles ont une qualité d'ajustement équivalente, et qu'on devrait donc privilégier le modèle le plus simple, la régression linéaire simple.

 Refaites le même processus avec les données de `ANC3DAT`, ajustez le modèle complet avec interaction (`LFKL~LAGE+LOCATE+LAGE:LOCATE`) et sans interaction (`LFKL~LAGE+LOCATE`), Comparez l'ajustement des deux modèles, que concluez vous?

```
model.full.anc3dat<-lm(lfkl ~ lage + locate + lage:locate,
data = anc3dat)
model.ancova.anc3dat<-lm(lfkl ~ lage + locate , data =
anc3dat)
anova(model.full.anc3dat,model.ancova.anc3dat)
```

Analysis of Variance Table

Model 1: `lfkl ~ lage + locate + lage:locate`

Model 2: `lfkl ~ lage + locate`

```

      Res.Df    RSS Df Sum of Sq    F    Pr(>F)
1         88 0.051358
2         89 0.060448 -1 -0.009090 15.575 0.0001592 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Cette fois-ci, le modèle plus complexe s'ajuste significativement mieux aux données. (Pas surprenant puisque nous avons précédemment conclu que l'interaction est significative avec ces données.)

Bootstrap

```

#####
#####
# Bootstrap analysis

# Bootstrap analysis BCa confidence intervals
# Preferable when parameter distribution is far from normal

# Bootstrap 95% BCa CI for regression coefficients
library(boot)

# To simplify future modifications of the code in this file,
# copy the data to a generic mydata dataframe
mydata<-anc3dat

# create a myformula variable containing the formula for the
model to be fitted
myformula<-as.formula(lfkl ~ lage + locate + lage:locate)

# function to obtain regression coefficients for each iteration
bs <- function(formula, data, indices) {
  d <- data[indices, ]
  fit <- lm(formula, data = d)
  return(coef(fit))
}

# bootstrapping with 1000 replications
results <- boot(data = mydata, statistic = bs, R = 1000,
  formula = myformula)

# view results
results
for (i in 1:length(results$t0)) {
  plot(results, index = i)
}

# get 95% confidence intervals

for (i in 1:length(results$t0)) {
  print(boot.ci(results, type = "bca", index = i))
}

```

```
}
```

Permutation test

```
#####
#####
#   Permutation test
#
#   using lmp perm library

#   To simplify future modifications of the code in this file,
#   copy the data to a generic mydata dataframe
mydata<-anc3dat

#   create a myformula variable containing the formula for the
#   model to be fitted
myformula<-as.formula(lfkl ~ lage + locate + lage:locate)

require(lmPerm2)

#   Fit desired model on the desired dataframe
mymodel <- lm(myformula , data = mydata)

#   Calculate p-values for each term by permutation
#   Note that lmp centers numeric variable by default, so to
#   get results that are
#   consistent with standard models, it is necessary to set
#   center=FALSE

mymodelProb <- lmp(myformula, data = mydata, center=FALSE,
  perm = "Prob")

summary(mymodel)
summary(mymodelProb)
```

Lab- Analyse de données de fréquence: Tableaux de contingence, modèles log-linéaires et régression de Poisson

Après avoir complété ce laboratoire, vous devriez être en mesure de:

- Créer et manipuler des jeux de données en R pour analyser des données de fréquences.
- Utiliser R pour éprouver une hypothèse extrinsèque à propos de données de fréquence d'une population.
- Utiliser R pour éprouver l'hypothèse d'indépendance pour des tableaux de contingence à 2 dimensions.
- Utiliser R pour ajuster des régressions de Poisson et des modèles log-linéaires à des données de fréquence.

Organisation des données: 3 formats

Les résultats de certaines expériences sont sous forme de fréquences, par exemple le nombre de plantes infectées par un pathogène sous différents régimes d'infection, ou le nombre de tortues mâles et femelles qui éclosent à différentes températures (oui, chez les tortues le sexe dépend de la température!), etc. La question statistique qui se pose généralement est de savoir si la proportion des observations dans chaque catégorie (infecté vs non infecté, mâle vs femelle, etc) diffère significativement entre les traitements (régime d'infection ou température dans les deux exemples). Pour répondre à cette question, on peut organiser les données de manière à refléter comment les observations se retrouvent dans chaque catégorie. Il existe 3 façons d'organiser ces données. Vous devriez être capable de choisir la manière appropriée pour votre analyse, et savoir convertir entre elles avec R.

Le fichier USPopSurvey.RData contient les données de recensement d'une ville du midwest américain en 1980:

```
# Get US Survey data, in "Frequency" form
load(url("http://www.antoinemorin.com/biostats/2013/USPopSurvey.RData")) #
online version

> USPopSurvey
  ageclass    sex frequency
1     0-9 female    17619
2    10-19 female    17947
3    20-29 female    21344
4    30-39 female    19138
5    40-49 female    13135
6    50-59 female    11617
```

```

7      60-69 female 11053
8      70-79 female 7712
9      80+ female 4114
10     0-9 male 17538
11     10-19 male 18207
12     20-29 male 21401
13     30-39 male 18837
14     40-49 male 12568
15     50-59 male 10661
16     60-69 male 9374
17     70-79 male 5348
18     80+ male 1926
>

```

Notez qu'il y a 18 lignes et 3 colonnes dans ce fichier. Chaque ligne donne le nombre de personnes (frequency) pour un sexe et une classe d'âge. Il y a `sum(USPopSurvey$frequency)=239439` individus qui ont été classifiés selon les 18 catégories (2 sexes x 9 classes d'âge). Cette manière de représenter les données est sous le **format de fréquences (frequency form)**. C'est un format compact permettant d'enregistrer les données quand il y a seulement des variables catégoriques à représenter.

Lorsqu'il y a des variables continues, ce format ne peut être utilisé. Les données doivent être enregistrée sous le **format de cas (case form)** dans laquelle chaque observation (individu) est représenté par une ligne dans le fichier, et où chaque variable est représentée par une colonne. Le package `vcdExtra` contient la fonction `expand.dft()` qui permet de convertir de la forme de fréquence à la forme de cas. Par exemple, pour créer un data frame avec 239439 lignes et 2 colonnes (sex et ageclass) à partir du data frame `USPopSurvey`:

```

USPopSurvey.caseform<-expand.dft(USPopSurvey, freq="frequency")
head(USPopSurvey.caseform)
ageclass sex
1      0-9 female
2      0-9 female
3      0-9 female
4      0-9 female
5      0-9 female
6      0-9 female

tail(USPopSurvey.caseform)
ageclass sex
239534    80+ male
239535    80+ male
239536    80+ male
239537    80+ male
239538    80+ male
239539    80+ male

```

Ces données peuvent finalement être organisées sous le **format de tableau (table form)** de contingence où chacune des n variables est représentée par une dimension d'un tableau n -dimensionnel (dans notre exemple on a 2 variables, sexe et classe d'âge, et les rangées pourraient représenter les classes d'âge et les colonnes chaque sexe). Les cellules de ce tableau contiennent les fréquences. Le format tableau peut être obtenu du format de fréquence ou de cas par la commande `xtabs()` :

```
# convert case form to table form
xtabs(~ageclass+sex,USPopSurvey.caseform)
sex
ageclass female  male
0-9      17619 17538
10-19    17947 18207
20-29    21344 21401
30-39    19138 18837
40-49    13135 12568
50-59    11617 10661
60-69    11053  9374
70-79     7712  5348
80+       4114  1926

# convert frequency form to table form
xtabs(frequency~ageclass+sex, data=USPopSurvey)
sex
ageclass female  male
0-9      17619 17538
10-19    17947 18207
20-29    21344 21401
30-39    19138 18837
40-49    13135 12568
50-59    11617 10661
60-69    11053  9374
70-79     7712  5348
80+       4114  1926
```

Figure 1. Fonctions permettant la conversion de données de fréquences entre les différents formats.

à ce format			
De ce format	Cas	Fréquence	Tableau
Cas		<code>xtabs(~A+B)</code>	<code>table(A,B)</code>
Fréquence	<code>expand.dft(X)</code>		<code>xtabs(count~A+B)</code>
Tableau	<code>expand.dft(X)</code>	<code>as.data.frame(X)</code>	

Graphiques Visualiser graphiquement les tableaux de contingence et test d'indépendance

Les tableaux de contingence peuvent servir à éprouver l'hypothèse d'indépendance des observations. Ceci équivaut à répondre à la question: est-ce que la classification des observations selon une variable (par exemple sex) indépendante de la classification par une autre variable (par exemple ageclass). En autres mots, est-ce que la proportion des mâles et femelles indépendante de l'âge ou varie avec l'âge?

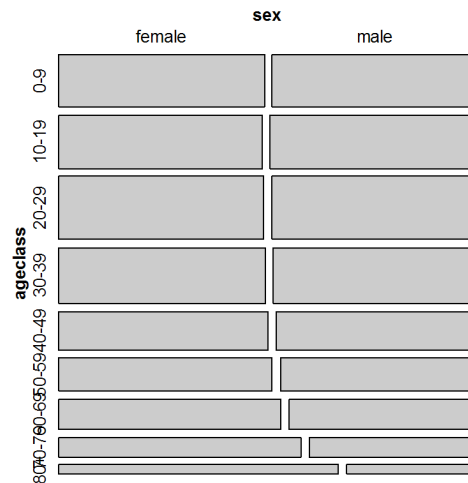
Le package vcd inclut la fonction `mosaic()` qui permet de représenter graphiquement le contenu d'un tableau de contingence:

```
library(vcd)

USTable<-xtabs(frequency~ageclass+sex, data=USPopSurvey) # save the table
form as USTable dataframe

# Mosaic plot of the contingency table
mosaic(USTable)
```

Figure 2.



Cette mosaïque représente la proportion des observations dans chaque combinaison de catégories (ici il y a 18 catégories, 2 sexes x 9 classes d'âge). Les catégories contenant une plus grande proportion d'observations sont représentées par de plus grands rectangles. Visuellement, on peut voir que la proportion des mâles et femelles est approximativement égale chez les jeunes, mais que la proportion des femelles augmente chez les personnes âgées.

Le test de Chi carré permet d'éprouver l'hypothèse nulle que la proportion des mâles et femelles ne change pas avec l'âge (est indépendante de l'âge):

```
# Test of independence

chisq.test(USTable) # runs chi square test of independence of sex and age
class

Pearson's Chi-squared test

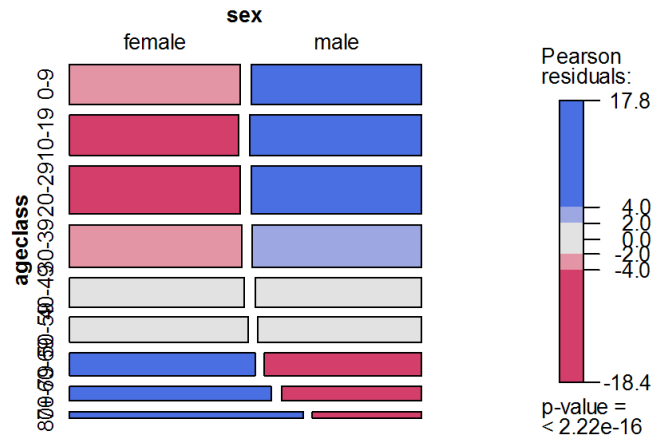
data:  USTable
X-squared = 1162.584, df = 8, p-value < 2.2e-16
```

La valeur p étant très faible, on rejette donc l'hypothèse nulle que âge et sexe sont indépendants.

Ces graphiques mosaïques peuvent être colorés pour souligner les catégories qui contribuent le plus à cette dépendance:

```
# Mosaic plot of the contingency table with shading
mosaic(USTable, shade=TRUE)
```

Figure 3.



La couleur de chaque rectangle est proportionnelle à la déviation des fréquences observées de ce qui serait attendu si l'âge et le sexe étaient indépendants. Les classes d'âge 40-49 et 50-59 ont un rapport des sexe approximativement égal à celui de toutes les classes d'âge réunies. Il y a plus de jeunes mâles et de femmes âgées que si le rapport des sexe ne variait pas avec l'âge, et ces rectangles sont colorés en bleu. De l'autre côté, il y a moins de jeunes femmes et de mâles âgés que si le rapport des sexe était indépendant de l'âge, et ces rectangles sont en rouge. La valeur p à la droite de la figure est pour le test de Chi carré qui éprouve l'hypothèse nulle d'indépendance pour l'ensemble des observations, toutes classes d'âge confondues.

L'estimation de la valeur p associée à la statistique du Chi carré est approximative lorsque les fréquences attendues sont faibles dans certaines cellules, et ce particulièrement pour les tableaux de contingence 2x2. Deux options permettant des valeurs p plus exactes sont préférées dans ce cas, et le choix dépend du nombre total d'observations. Pour de grands échantillons (comme ici avec plus de 200,000 observations!), une approche par simulation de type Monte Carlo est suggérée et peut être obtenue en ajoutant `simulate.p.value=TRUE` comme argument à la fonction `chisq.test()` :

```
# Monte-carlo estimation of p value (better for small n)
chisq.test(UStable, simulate.p.value=TRUE, B=10000)

Pearson's Chi-squared test with simulated p-value (based on 10000
replicates)
```



```
data: USTable
X-squared = 1162.584, df = NA, p-value = 9.999e-05
```

Ici, la simulation a été faite $B=10000$ fois, et la valeur de Chi carré observée avec les données réelles n'a jamais été observée. Par conséquent, p a été estimé à $1/10001=9.999e-05$, qui est beaucoup plus élevé que la valeur p estimée à partir de la distribution théorique de Chi carré ($p < 2.2e-16$). Cette différence est due au moins en partie à un artéfacts de la simulation. Pour obtenir des valeurs p de l'ordre de $1e-16$, il faut effectuer au moins 10^{16} simulations. Et je ne suis pas aussi patient que ça!

Pour de petits tableaux de contingence avec des fréquences attendues petites, le test exact de Fisher peut servir à estimer la valeur p associée à l'hypothèse d'indépendance. Mais ce test ne peut être effectué avec de grands échantillons, comme ici:

```
> #Fisher exact test for contingency tables (small samples and small tables)
> fisher.test(USTable) # fails here because too many observations
Error in fisher.test(USTable) : FEXACT error 40.
Out of workspace.
> fisher.test(USTable, simulate.p.value=TRUE, B=10000)
```

```
Fisher's Exact Test for Count Data with simulated p-value (based on
10000 replicates)
```

```
data: USTable
p-value = 9.999e-05
alternative hypothesis: two.sided
```

Régression de Poisson: une alternative au test de Chi carré pour les tableaux de contingence

Rendu à ce stade, vous devriez avoir appris à apprécier la flexibilité et la généralité des modèles linéaires, et réaliser que le test de t est un cas spécial d'un modèle linéaire avec une variable indépendante catégorique. L'analyse des tableaux de contingence par le test du Chi carré peut également être généralisé. Un modèle linéaire généralisé pour une distribution de Poisson peut être utilisé quand la variable dépendante est une fréquence d'observations et les variables indépendantes sont catégorique (comme pour les tableaux de contingence, on parle alors de modèles log-linéaires), continue (régression Poisson), ou une combinaison de variables indépendante continues et catégoriques (aussi appelé régression de Poisson, mais avec des variables catégoriques en plus, analogue à l'ANCOVA sensu largo).

Ces modèles prédisent le logarithme naturel de la fréquence des observations en fonction des variables indépendantes. Comme pour les modèles linéaires qui présument de la normalité des résidus, on peut évaluer la qualité d'ajustement du modèle (par AICc par exemple)

et la signification statistique des termes du modèle (par exemple en comparant l'ajustement d'un modèle "complet" et celui d'un modèle qui exclue un terme à tester). On peut également obtenir des estimés des paramètres pour chaque terme dans le modèle, avec des intervalles de confiance et des valeurs p pour l'hypothèse nulle que ce terme n'a pas d'influence sur la fréquence.

La fonction `glm()` avec l'option `family=poisson()` permet l'estimation, par la méthode du maximum de vraisemblance, de modèles linéaires pour des fréquences.

Comparativement aux modèles linéaires vus précédemment, une des particularités de ces modèles est que seuls les termes d'**interaction** sont d'intérêt.

En partant des données de recensement en forme tableau, on peut ajuster un glm aux fréquences observées par sexe et classe d'âge par:

```
mymodel<-glm(frequency~sex*ageclass,family=poisson(),data=USPopSurvey)
> summary(mymodel)
```

```
Call:
glm(formula = frequency ~ sex * ageclass, family = poisson(),
    data = USPopSurvey)
```

```
Deviance Residuals:
[1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    9.776733   0.007534 1297.730 < 2e-16 ***
sexmale        -0.004608   0.010667  -0.432  0.6657
ageclass10-19   0.018445   0.010605   1.739  0.0820 .
ageclass20-29   0.191793   0.010179  18.842 < 2e-16 ***
ageclass30-39   0.082698   0.010441   7.921 2.36e-15 ***
ageclass40-49  -0.293697   0.011528 -25.477 < 2e-16 ***
ageclass50-59  -0.416508   0.011951 -34.850 < 2e-16 ***
ageclass60-69  -0.466276   0.012134 -38.428 < 2e-16 ***
ageclass70-79  -0.826200   0.013654 -60.511 < 2e-16 ***
ageclass80+    -1.454582   0.017316 -84.004 < 2e-16 ***
sexmale:ageclass10-19  0.018991   0.014981   1.268  0.2049
sexmale:ageclass20-29  0.007275   0.014400   0.505  0.6134
sexmale:ageclass30-39 -0.011245   0.014803  -0.760  0.4475
sexmale:ageclass40-49 -0.039519   0.016416  -2.407  0.0161 *
sexmale:ageclass50-59 -0.081269   0.017136  -4.742 2.11e-06 ***
sexmale:ageclass60-69 -0.160154   0.017633  -9.083 < 2e-16 ***
sexmale:ageclass70-79 -0.361447   0.020747 -17.422 < 2e-16 ***
sexmale:ageclass80+   -0.754343   0.029598 -25.486 < 2e-16 ***
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)
```

```
Null deviance: 5.3611e+04 on 17 degrees of freedom
Residual deviance: -1.6116e-12 on 0 degrees of freedom
AIC: 237.31
```

```
Number of Fisher Scoring iterations: 2
```

L'ajustement du modèle complet, avec l'interaction triple `sex:ageclass`, permet à la proportion des mâles et femelles de changer entre les classes d'âge, et donc d'estimer exactement les fréquences observées pour chaque combinaison de sexe et classe d'âge (notez que les résidus (deviance residuals) sont tous 0, et que l'estimé de déviance résiduelle est également approximativement zéro).

Un masochiste peut utiliser le tableau des coefficients pour obtenir la fréquence prédite pour les différentes catégories. Les fréquences prédites, comme pour l'ANOVA à critères multiple, sont obtenus en additionnant les coefficients appropriés. Puisque, en R, le premier niveau d'une variable catégorique (facteur) en ordre alphabétique est utilisé comme référence, l'ordonnée à l'origine (9.776733) est la valeur prédite pour le logarithme naturel de la fréquence des femelles dans la première classe d'âge (0 to 9). En effet, $e^{9.776733}$ est approximativement égal à 17619, le nombre observé de femelles dans cette classe d'âge. Pour les mâles dans la classe d'âge 80+, il faut calculer l'antilog du coefficient pour l'ordonnée à l'origine (pour les femelles dans la première classe d'âge), plus le coefficient pour `sexmale` (égal à la différence du log de la fréquence entre les femelles et les mâles), plus le coefficient pour la classe d'âge 80+ qui correspond à la différence de fréquence entre cette classe d'âge et la classe d'âge de référence, plus le coefficient pour l'interaction `sexmale:ageclass80+` (qui correspond à la différence de proportion de mâles dans cette classe d'âge par rapport à la classe d'âge de référence). Ceci donne:
 $\ln(\text{frequency}) = 9.776733 - 0.004608 - 1.454582 - 0.754343 = 7.5632$, et la fréquence est égale à $e^{7.5632} = 1926$

Il y a de nombreuses valeurs p dans ce tableau, mais elles ne sont en général pas très utiles. Pour éprouver l'hypothèse que l'effet du sexe sur la fréquence est identique dans chaque classe d'âge (i.e. que sexe et âge sont indépendants), vous devez ajuster un modèle qui exclut cette interaction (`sex:ageclass`) et déterminer comment l'ajustement du modèle est affecté.

La fonction `Anova()` du package `car` permet de prendre un raccourci:

```
> library(car)
> Anova(mymodel, type=3, test = "LR")
Analysis of Deviance Table (Type III tests)

Response: frequency
          LR Chisq Df Pr(>Chisq)
sex           0.2   1   0.6657
ageclass    21074.6  8  <2e-16 ***
sex:ageclass  1182.2  8  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les arguments `type=3` and `test="LR"` font en sorte que le test effectué pour comparer le modèle complet aux modèles réduits est le test de Chi carré sur le rapport de vraisemblance (Likelihood Ratio Chi-Square) à partir de la variance résiduelle, et que c'est un test partiel, et non séquentiel.

Selon ces tests, il n'y a pas d'effet principal de sexe ($p=0.667$) mais il y a un effet principal de ageclass et une interaction significative `sex:ageclass`. L'interaction significative signifie que l'effet du sexe sur la fréquence varie selon les classes d'âge, bref que le rapport des sexe varie avec l'âge. L'effet principal de ageclass signifie que la fréquence des individus varie avec l'âge dans la population recensée (i.e. que certaines classes d'âge sont plus peuplées que d'autres). L'absence d'un effet principal du sexe suggère qu'il y a approximativement le même nombre de mâles et femelles dans l'échantillon (quoique, puisqu'il y a une interaction, vous devez être prudents en faisant cette déclaration. C'est "vrai" au total, mais semble incorrect pour certaines classes d'âge).

Tester une hypothèse extrinsèque

Le test d'indépendance ci-dessus éprouve une hypothèse **intrinsèque** parce que les proportions utilisées pour calculer les valeurs attendues et tester l'indépendance sont celles observées (i.e. la proportion des mâles et femelles dans tout l'échantillon, et la proportion des individus dans chaque classe d'âge).

Pour éprouver l'hypothèse (**extrinsèque**) que le rapport des sexes est 1:1 pour les individus les plus jeunes (ageclass 0-9), on doit produire le tableau 2X2 des fréquences observées et attendues. Les fréquences attendues sont obtenues simplement en divisant le total des mâles et femelles par 2.

Code R pour créer et analyser un tableau de contingence 2X2 et éprouver une hypothèse extrinsèque

```
### Produce a table of obs vs exp for 0-9 age class
Popn0.9 <- rbind(c(17578, 17578), c(17619, 17538))
### Run x2 test on above table
chisq.test(Popn0.9, correct=F)### X2 without Yates
chisq.test(Popn0.9) ### X2 with Yates
```

🖨️ Éprouvez l'hypothèse nulle que la proportion de mâles et femelles à la naissance est égale. Que concluez-vous? Croyez-vous que ces données sont appropriées pour tester cette hypothèse?

```
> chisq.test(Popn0.9, correct=F)### X2 without Yates

Pearson's Chi-squared test

data:  Popn0.9
X-squared = 0.0933, df = 1, p-value = 0.76

> chisq.test(Popn0.9) ### X2 with Yates

Pearson's Chi-squared test with Yates' continuity correction

data:  Popn0.9
X-squared = 0.0888, df = 1, p-value = 0.7658
```

Notez que pour un tableau 2X2, on devrait utiliser une correction de Yates ou un test de Fisher. Le test de Fisher ne pouvant être utilisé lorsque l'échantillon dépasse 200, on utilise la correction de Yates. Selon cette analyse, on accepte l'hypothèse nulle que le rapport des sexes est 1:1 à la naissance. Ceci dit, ces données ne sont pas très appropriées pour éprouver l'hypothèse car la première classe d'âge est trop grossière. Il est possible que le rapport des sexes à la naissance soit différent de 1:1 mais que la mortalité différentielle des deux sexes compense au cours des 9 premières années (par exemple si il y a plus de mâles à la naissance, mais que les jeunes garçons ont une survie plus faible au cours de leurs 9 premières années). Dans un tel cas, le rapport des sexes n'est PAS de 1:1 à la naissance, mais on accepte l'hypothèse nulle à partir des données dans la classe d'âge 0-9.

Régression de Poisson pour l'analyse de tableaux de contingence à plusieurs critères

Le principe d'éprouver l'indépendance en examinant les interactions peut être utilisé avec les tableaux de contingence à plusieurs critères. Par exemple, examinons si la température (2 niveaux: base et haute) et l'éclairage (2 niveaux: bas et haut) affectent si des plantes sont infectées (2 niveaux: infecté et non-infecté) par un pathogène. On peut représenter ces données par un tableau de contingence à 3 critères (température, lumière, statut d'infection).

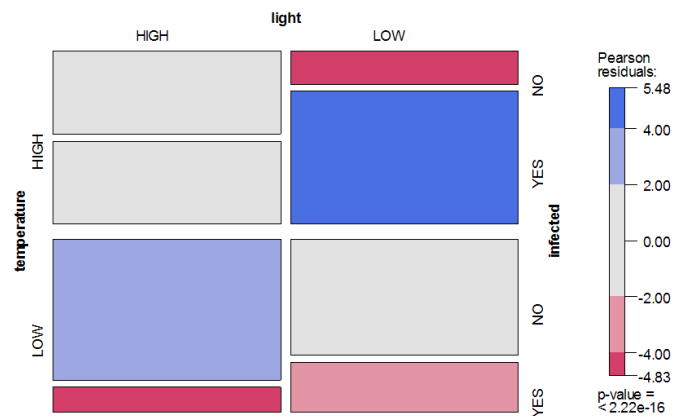
L'ajustement de modèles log-linéaires à des données de fréquence implique que l'on éprouve plusieurs modèles en les comparant au modèle complet (saturé). Une série de modèles contenant tous les termes sauf une des interactions qui nous intéressent est produite, et l'ajustement de chaque modèle est comparé à celui du modèle complet. Si la réduction de la qualité d'ajustement n'est pas significative, cela implique que l'interaction manquante contribue peu à la qualité de l'ajustement. Par contre, si le modèle réduit s'ajuste nettement moins bien aux données, alors l'interaction manquante

contribue beaucoup à l'ajustement du modèle complet. Comme pour les tableaux de contingence 2X2, les termes qui nous intéressent le plus sont les interactions, pas les effets principaux, si l'on teste pour l'indépendance des différents facteurs.

Le fichier `loglin.Rdata` contient les fréquences (`frequency`) des plantes infectées ou non infectées (`infected`) à basse et haute température (`temperature`) à basse et haute lumière (`light`). Pour visualiser ces données et déterminer si le taux d'infection dépend de la lumière et de la température, on peut faire une figure mosaïque et ajuster un modèle log-linéaire:

```
# Convert from frequency form to table form for mosaic plot
loglinTable<-xtabs(frequency~temperature+light+infected, data=loglin)
# Create mosaic plot to look at data
library(vcd)
mosaic(loglinTable, shade=TRUE)
```

Figure 4.



Cette expérience contrôlée avec le même nombre de plantes à chaque niveau de lumière et de température produit une mosaïque où la surface occupée par les observations dans les quatre quadrants est égale. Ce qui nous intéresse, le taux d'infection par le pathogène, semble varier entre les quadrants (i.e. les niveaux de température et de lumière). Le rectangle rouge dans le coin en bas à gauche indique que le nombre de plantes infectées à basse température et haute lumière est plus faible qu'attendu si ces deux facteurs n'influencent pas le taux d'infection. Même chose pour les conditions de basse lumière et de haute température (coin supérieur droit). La valeur p au bas de l'échelle représente un test d'indépendance équivalent à comparer le modèle complet au modèle excluant toutes les interactions et ne contenant que les effets principaux de la température, la lumière, et le statut d'infection sur le logarithme naturel du nombre d'observations.

```
# Fit full model
full.model<-glm(frequency~temperature*light*infected, family=poisson(),
data=loglin)
```

```
# Test partial effect of terms in full model
require(car)
Anova(full.model, type=3, test="LR")
```

Analysis of Deviance Table (Type III tests)

Response: frequency

	LR	Chisq	Df	Pr(>Chisq)
temperature	9.1786	1	0.0024487	**
light	13.2829	1	0.0002678	***
infected	0.0000	1	1.0000000	
temperature:light	5.6758	1	0.0172008	*
temperature:infected	29.0612	1	7.013e-08	***
light:infected	20.2687	1	6.729e-06	***
temperature:light:infected	1.0840	1	0.2978126	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Interpretation of S-PLUS output

Les probabilités associées à chaque terme sont ici calculées en comparant l'ajustement du modèle complet à un modèle qui exclue seulement le terme d'intérêt. Plusieurs des termes sont ici sans véritable intérêt puisque les fréquences sont partiellement contrôlées dans notre expérience. Puisque la question biologique porte sur le taux d'infection, les seuls termes d'intérêt sont les termes d'interactions qui incluent le statut d'infection:

temperature:infected, light:infected, l'interaction triple temperature:light:infected.

- L'interaction significative temperature:infected implique que le taux d'infection n'est pas indépendant de la température. D'ailleurs il est apparent dans la mosaïque que le taux d'infection (le nombre relatif de plantes infectées) est supérieur à haute température.
- L'interaction significative light:infected implique que le taux d'infection dépend de la lumière. La mosaïque illustre que la proportion des plantes infectées est plus élevée en basse lumière.
- L'interaction temperature:light:infected n'est pas significative. Cela implique que l'effet de la température et de la lumière sur le taux d'infection sont indépendants. Autrement dit, l'effet de la lumière sur le taux d'infection ne dépend pas de la température, et vice versa.

Exercice

Le fichier `sturmdat` contient les données qui vous permettront d'éprouver l'hypothèse que le nombre d'esturgeons capturé est indépendants du site, de l'année, et du sexe. Avant de commencer l'analyse, les données devront être réorganisées pour pouvoir ajuster un modèle log-linéaire:

🖨️ Ouvrez `sturmdat.Rdata`, puis utilisez la fonction `table()` pour obtenir les fréquence d'individus capturés par `sex`, `location`, et `year`. Sauvegardez ce tableau comme `sturmdat.table`. Faites une figure mosaïque de ces données:

```
load(url("http://www.antoinemorin.com/biostats/2010/Sturmdat.Rdata")) #
online version, data in case form

# Reorganize data from case form to table form
sturmdat.table<-with(sturmdat, table(sex,year,location))
# display the table
sturmdat.table
```

```
, , location = CUMBERLAND

      year
sex    1978    1979    1980
FEMALE    10     30     11
MALE      14     14      6
```

```
, , location = THE_PAS

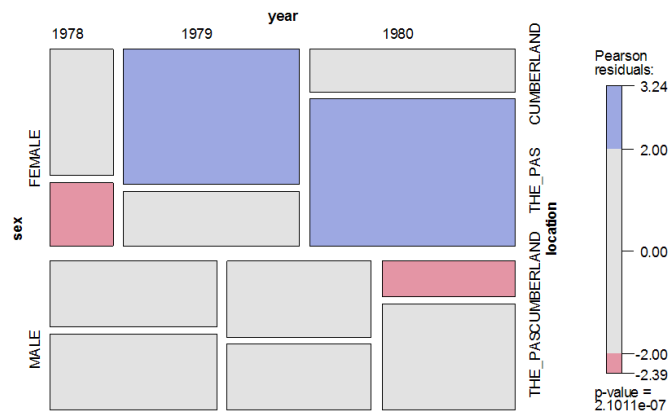
      year
sex    1978    1979    1980
FEMALE     5     12     38
MALE      16     12     18
```

```
# Create data frame while converting from table form to frequency form
sturmdat.freq<-as.data.frame(sturmdat.table)
#display data frame
sturmdat.freq
```

	sex	year	location	Freq
1	FEMALE	1978	CUMBERLAND	10
2	MALE	1978	CUMBERLAND	14
3	FEMALE	1979	CUMBERLAND	30
4	MALE	1979	CUMBERLAND	14
5	FEMALE	1980	CUMBERLAND	11
6	MALE	1980	CUMBERLAND	6
7	FEMALE	1978	THE_PAS	5
8	MALE	1978	THE_PAS	16
9	FEMALE	1979	THE_PAS	12
10	MALE	1979	THE_PAS	12
11	FEMALE	1980	THE_PAS	38
12	MALE	1980	THE_PAS	18

```
# Look at the data as mosaic plot
library(vcd)
# mosaic using the table created above
mosaic(sturmdat.table, shade=TRUE)
```


Figure 5.



À partir de ces données en format de fréquence, ajustez le modèle log-linéaire complet et le tableau d'anova avec les statistique de Chi carré pour les termes du modèles. Est-ce que l'interaction triple ($location:year:sex$) est significative? Est-ce que le rapport des sexes varien entre les sites ou d'une année à l'autre?.

```
# Fit full model
full.model<-glm(Freq~sex*year*location, data=sturgdat.freq, family="poisson")
summary(full.model)
library(car)
Anova(full.model, type=3)
```

Call:
glm(formula = Freq ~ sex * year * location, family = "poisson",
data = sturgdat.freq)

Deviance Residuals:
[1] 0 0 0 0 0 0 0 0 0 0 0 0

Coefficients:

	value	Pr(> z)	Estimate	Std. Error	z
(Intercept)	7.281	3.3e-13 ***	2.30259	0.31623	
sexMALE	0.813	0.41641	0.33647	0.41404	
year1979	3.009	0.00262 **	1.09861	0.36515	
year1980	0.218	0.82732	0.09531	0.43693	
locationTHE_PAS	1.266	0.20569	-0.69315	0.54772	-
sexMALE : year1979	2.090	0.03658 *	-1.09861	0.52554	-
sexMALE : year1980	1.439	0.15011	-0.94261	0.65498	-
sexMALE : locationTHE_PAS	1.255	0.20950	0.82668	0.65873	
year1979 : locationTHE_PAS	0.346	0.72957	-0.22314	0.64550	-
year1980 : locationTHE_PAS	2.992	0.00277 **	1.93284	0.64593	
sexMALE : year1979 : locationTHE_PAS	0.077	0.93875	-0.06454	0.83986	-

```

sexMALE      :year1980      :locationTHE_PAS      -0.96776      0.87942      -
1.100  0.27114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 5.7176e+01 on 11 degrees of freedom
Residual deviance: 4.6629e-15 on 0 degrees of freedom
AIC: 77.28
Analysis of Deviance Table (Type III tests)


Response: Freq
      LR Chisq Df Pr(>Chisq)
sex      0.6698  1  0.4131256
year     13.8895  2  0.0009637 ***
location   1.6990  1  0.1924201
sex:year    4.6930  2  0.0957024 .
sex:location 1.6323  1  0.2013888
year:location 25.2580  2  3.276e-06 ***
sex:year:location 1.6677  2  0.4343666
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Ce tableau a trois critères: *sex*, *location* et *year*. Donc le modèle complet (saturé) contient 7 termes: trois effets principaux (*sex*, *location* et *year*), trois interactions du second degré (double) (*sex:year*, *sex:location* et *year:location*) et une interaction du troisième degré (triple) (*sex:year:location*). La déviance nulle est 57.17574, la déviance résiduelle du modèle complet est, sans surprise, 0. La déviance pouvant être attribuée à l'interaction triple est 1.6677, non significative.

Qu'est ce que cela implique? S'il y a des interactions doubles, alors elles ne dépendent pas de la troisième variable. Par exemple, si le rapport des sexes des esturgeons varie d'une année à l'autre (une interaction *sex:year*), alors cette tendance est la même aux 2 stations.

Puisqu'il n'y a pas d'interaction triple, il est (statistiquement) justifié de combiner les données pour éprouver les interactions du second degré. Par exemple, pour tester l'effet *sex:location*, on peut combiner les années. Pour tester l'effet *sex:year*, on peut combiner les sites. Cette aggrégation a pour effet d'augmenter la puissance, et est analogue à la stratégie en ANOVA à critères multiples. L'approche de la régression de Poisson permet de faire l'équivalent simplement en ajustant le modèle sans l'interaction du troisième degré.

 Ajustez le modèle en excluant l'interaction du troisième degré:

```


Analysis of Deviance Table (Type III tests)

Response: Freq
      LR Chisq Df Pr(>Chisq)
sex      1.8691  1  0.1715807
year     15.1289  2  0.0005186 ***
location   1.5444  1  0.2139568
sex:year   15.5847  2  0.0004129 ***
sex:location 2.1762  1  0.1401583
year:location 28.3499  2  6.981e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

L'interaction `sex:location` n'explique pas une portion significative de la déviance, alors que les deux autres sont significatives. Le rapport des sexes ne varie pas entre les sites, mais il varie selon les années. L'interaction `year:location` est aussi significative (voir plus bas pour son interprétation).

Devriez vous tenter de simplifier le modèle encore plus? Les vrais statisticiens sont divisés sur cette question. Tous s'entendent cependant sur le fait que conserver des interactions non significatives dans un modèle peut réduire la puissance. De l'autre côté, le retrait des interactions non significatives peut rendre l'interprétation plus délicate lorsque les observations ne sont pas bien balancées (i.e. il y a de la colinéarité entre les termes du modèle).

 Ajustez le modèle sans l'interaction `sex:location`:

Analysis of Deviance Table (Type III tests)

Response: Freq

	LR	Chisq	Df	Pr(>Chisq)
sex	5.0970	1	0.0239677	*
year	16.1226	2	0.0003155	***
location	0.2001	1	0.6546011	
sex:year	13.9883	2	0.0009173	***
year:location	26.7534	2	1.551e-06	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Les deux interactions sont significatives et ce modèle semble le meilleur.

Ce modèle est:

$$\ln[f_{(ijk)}] = \text{location} + \text{sex} + \text{year} + \text{sex:year} + \text{location:year}.$$

Comment ces effets peuvent-ils être interprétés biologiquement? Souvenez vous que, comme dans les test d'indépendance, on n'est pas vraiment intéressé aux effets principaux, seulement par les interactions. Par exemple, l'effet principal de `location` nous dit que le nombre total d'esturgeons capturé (le total des 2 sexes pendant les 3 années d'échantillonnage) diffère entre les 2 sites. Cela n'est pas vraiment surprenant et peu intéressant en l'absence d'information sur l'effort de pêche. Cependant, l'interaction `sex:year` nous dit que le rapport des sexes a changé d'une année à l'autre. Et puisque l'interaction du troisième degré n'est pas significative, on sait que ce changement dans le temps est approximativement le même dans les deux sites. Un résultat possiblement intéressant. Pourquoi? Comme l'expliquer?

L'interaction `location:year` nous dit que le nombre d'esturgeons n'a pas seulement varié d'une année à l'autre, mais que la tendance dans le temps diffère entre les deux sites. Ceci pourrait refléter une

différence d'effort de pêche à un des sites durant l'une des campagnes d'échantillonnage, ou un impact à seulement un des deux sites la dernière année par exemple. Mais cette tendance est la même pour les mâles et les femelles (donc n'a pas affecté le rapport des sexes) puisque l'interaction triple n'est pas significative.

