

# Régression linéaire simple

*Timothée Flutre*

*01/02/2015*

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Préambule</b>                          | <b>1</b> |
| <b>2</b> | <b>Contexte</b>                           | <b>3</b> |
| <b>3</b> | <b>Introduction</b>                       | <b>3</b> |
| <b>4</b> | <b>Ecrire le modèle</b>                   | <b>3</b> |
| 4.1      | Notations . . . . .                       | 3        |
| 4.2      | Vraisemblance . . . . .                   | 4        |
| <b>5</b> | <b>Simuler des données</b>                | <b>4</b> |
| <b>6</b> | <b>Réaliser l'inférence</b>               | <b>7</b> |
| 6.1      | Dérivation mathématique . . . . .         | 7        |
| 6.2      | Implémentation (R stats) . . . . .        | 7        |
| 6.3      | Implémentation (R base) . . . . .         | 9        |
| <b>7</b> | <b>Explorer les simulations possibles</b> | <b>9</b> |
| <b>8</b> | <b>Perspectives</b>                       | <b>9</b> |
| <b>9</b> | <b>Appendix</b>                           | <b>9</b> |

## 1 Préambule

Ce document a été généré à partir d'un fichier texte au format Rmd utilisé avec le logiciel libre [R](#). Pour exporter un tel fichier vers les formats HTML et PDF, installez le paquet [rmarkdown](#) disponible sur CRAN (il va vraisemblablement vous être demandé d'installer d'autres paquets), puis ouvrez R et entrez:

```
library(rmarkdown)
render("myanalysis.Rmd", "all")
```

Il est généralement plus simple d'utiliser le logiciel libre [RStudio](#), mais ce n'est pas obligatoire.

Pour plus de détails, lisez [cette page](#), et regardez les exemples ci-dessous.

Exemple de liste non ordonnée:

- aaa
- bbb

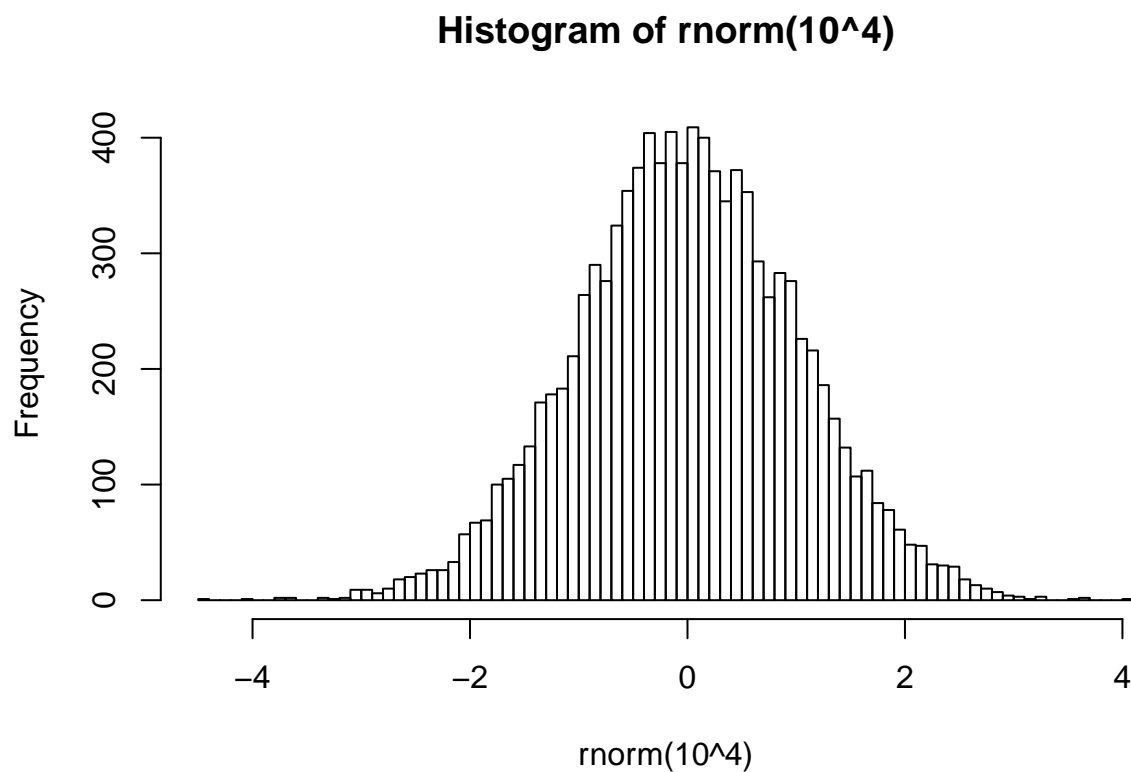
Exemple de liste ordonnée:

1. ccc
2. ddd

Exemple de mot en **gras** ou *italique*.

Exemple de graphique à partir de R:

```
hist(rnorm(10^4), breaks="FD")
```



Exemple de tableau à partir de R:

```
knitr::kable(mtcars[1:5,])
```

|                   | mpg  | cyl | disp | hp  | drat | wt   | qsec | vs | am | gear | carb |
|-------------------|------|-----|------|-----|------|------|------|----|----|------|------|
| Mazda RX4         | 21.0 | 6   | 160  | 110 | 3.90 | 2.62 | 16.5 | 0  | 1  | 4    | 4    |
| Mazda RX4 Wag     | 21.0 | 6   | 160  | 110 | 3.90 | 2.88 | 17.0 | 0  | 1  | 4    | 4    |
| Datsun 710        | 22.8 | 4   | 108  | 93  | 3.85 | 2.32 | 18.6 | 1  | 1  | 4    | 1    |
| Hornet 4 Drive    | 21.4 | 6   | 258  | 110 | 3.08 | 3.21 | 19.4 | 1  | 0  | 3    | 1    |
| Hornet Sportabout | 18.7 | 8   | 360  | 175 | 3.15 | 3.44 | 17.0 | 0  | 0  | 3    | 2    |

Exemple d'équation écrite en [LaTeX](#) ([livre en ligne](#)):  $y_i \sim \mathcal{N}(\mu, \sigma^2)$

## 2 Contexte

Ce document fait partie de l'atelier "Prédiction Génomique" organisé et animé par Jacques David et Timothée Flutre en 2015 à [Montpellier SupAgro](#) dans le cadre de l'option [APIMET](#) (Amélioration des Plantes et Ingénierie végétale Méditerranéennes et Tropicales) couplée à la spécialité SEPMET (Semences Et Plantes Méditerranéennes Et Tropicales) du [Master 3A](#) (Agronomie et Agroalimentaire).

Ce document a pour but d'introduire concrètement les étudiants à l'un des aspects de la modélisation statistique, la simulation. Il prend comme exemple la régression linéaire simple, historiquement développée par [Galton \(1886\)](#) pour étudier l'hérédité de la taille dans l'espèce humaine.

## 3 Introduction

"Essentially, all models are wrong, but some are useful." (Box, 1987). Cette célèbre citation illustre parfaitement le fait qu'un modèle est une simplification du phénomène étudié, mais qu'après tout, si cette simplification nous apporte des enseignements et nous permet de prendre de bonnes décisions, cela importe tout autant.

Il est donc important de rappeler que la première question à se poser, en tant que modélisateur, concerne la validité du modèle. Bien que cela paraisse évident, ceci consiste avant tout à vérifier que les données à analyser correspondent bien à la question à laquelle on veut répondre ([Gelman & Hill, 2006](#)).

Il est également utile, pour mieux comprendre le processus de modélisation statistique, de distinguer le "monde réel", dans lequel vivent les données, du "monde théorique", dans lequel vivent les modèles: "When we use a statistical model to make a statistical inference, we implicitly assert that the variation exhibited by data is captured reasonably well by the statistical model, so that the theoretical world corresponds reasonably well to the real world." ([Kass, 2011](#)).

En particulier, il ne faut pas confondre les données avec des variables aléatoires, même si on fait souvent le raccourci: "In both approaches [frequentist and Bayesian], a statistical model is introduced and we may say that the inference is based on what *would* happen if the data *were* to be random variables distributed according to the statistical model. This modeling assumption would be reasonable if the model *were* to describe accurately the variation in the data." ([Kass, 2011](#)).

## 4 Ecrire le modèle

### 4.1 Notations

- $n$ : nombre d'individus (diploïdes, supposés non-apparentés)
- $i$ : indice indiquant le  $i$ -ème individu, donc  $i \in \{1, \dots, n\}$
- $y_i$ : phénotype de l'individu  $i$  pour la caractéristique d'intérêt
- $\mu$ : moyenne globale des phénotypes des  $n$  individus
- $f$ : fréquence de l'allèle minoritaire à ce SNP (sur un autosome)
- $x_i$ : génotype de l'individu  $i$  au marqueur SNP d'intérêt, codé comme le nombre de copie(s) de l'allèle minoritaire à ce SNP chez cet individu ( $\forall i, x_i \in \{0, 1, 2\}$ )
- $\beta$ : effet additif de chaque copie de l'allèle minoritaire en unité du phénotype

- $\epsilon_i$ : erreur pour l'individu  $i$
- $\sigma^2$ : variance des erreurs
- $\mathcal{N}(\mu, \sigma^2)$ : distribution Normale univariée de moyenne  $\mu$  et variance  $\sigma^2$

## 4.2 Vraisemblance

$$\forall i \quad y_i = \mu + \beta x_i + \epsilon_i \text{ avec } \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

## 5 Simuler des données

Afin de simuler des données, nous allons utiliser un générateur de nombres pseudo-aléatoires. Le mot “pseudo” est là pour rappeler que les générateurs informatiques sont déterministes et peuvent donc être initialisés avec une graine (*seed*), très utile pour la reproductibilité des analyses:

```
seed <- 1859
set.seed(seed)
```

Commençons par fixer le nombre d'individus:

```
n <- 500
```

Puis la moyenne globale (de manière arbitraire, ce n'est pas très important car on peut toujours centrer les phénotypes en début d'analyse):

```
mu <- 50
```

Pour simuler les génotypes, nous allons supposer que la population est à l'équilibre d'Hardy-Weinberg:

```
##' Calculate the genotype frequencies (AA, Aa, aa) at Hardy-Weinberg equilibrium.
##'
##' https://en.wikipedia.org/wiki/Hardy%E2%80%93Weinberg_principle
##' @param maf frequency of the minor allele (a)
##' @return vector of genotype frequencies
##' @author Timothée Flutre
calc.geno.freq <- function(maf){
  geno.freq <- c((1 - maf)^2,
                2 * (1 - maf) * maf,
                maf^2)
  names(geno.freq) <- c("AA", "Aa", "aa")
  return(geno.freq)
}

f <- 0.3
genotypes <- sample(x=c(0,1,2), size=n, replace=TRUE, prob=calc.geno.freq(f))
```

Le morceau de code ci-dessus vous montre aussi les bonnes pratiques de programmation:

- choisir des noms de fonctions et variables clairs et explicites;

- documenter le code;
- citer des référence si nécessaire.s

Regardons à quoi ressemblent les génotypes que nous venons de simuler:

```
table(genotypes)
```

```
## genotypes
##    0    1    2
## 267 192  41
```

```
sum(genotypes) / (2 * n)
```

```
## [1] 0.274
```

```
var(genotypes)
```

```
## [1] 0.413
```

Tirons une valeur pour l'effet du génotype sur le phénotype:

```
(beta <- rnorm(n=1, mean=1, sd=1))
```

```
## [1] 1.51
```

Simulons des erreurs:

```
sigma <- 1
errors <- rnorm(n=n, mean=0, sd=sigma)
```

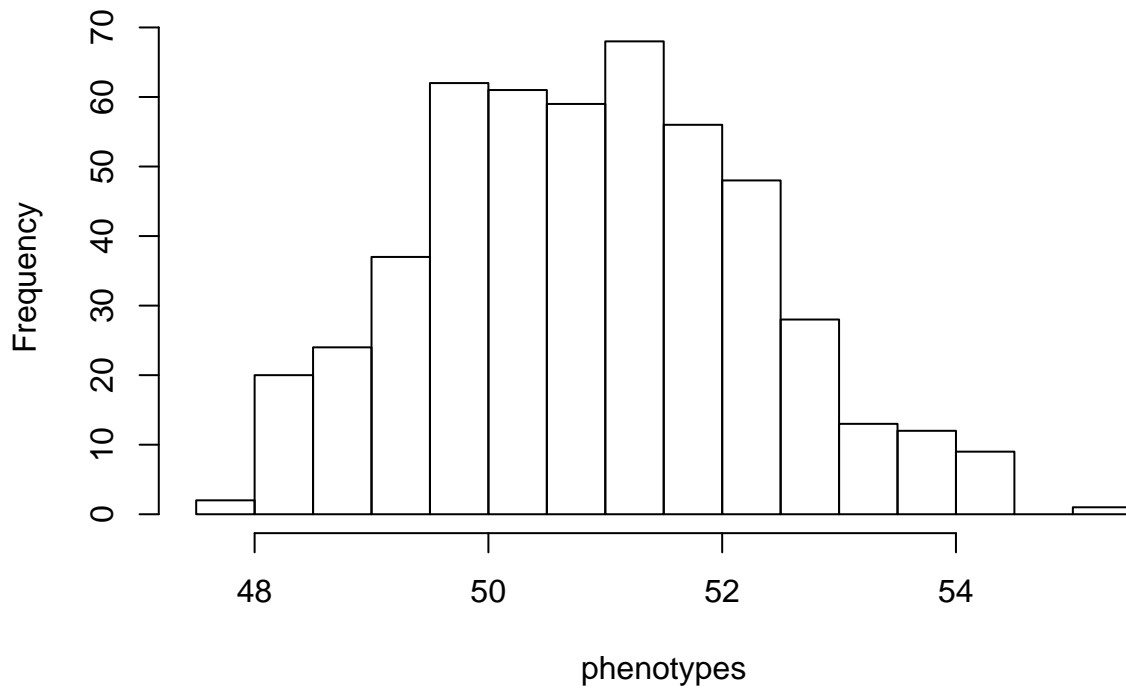
Nous avons maintenant tout ce qu'il faut pour simuler les phénotypes:

```
phenotypes <- mu + beta * genotypes + errors
```

Regardons à quoi ils ressemblent:

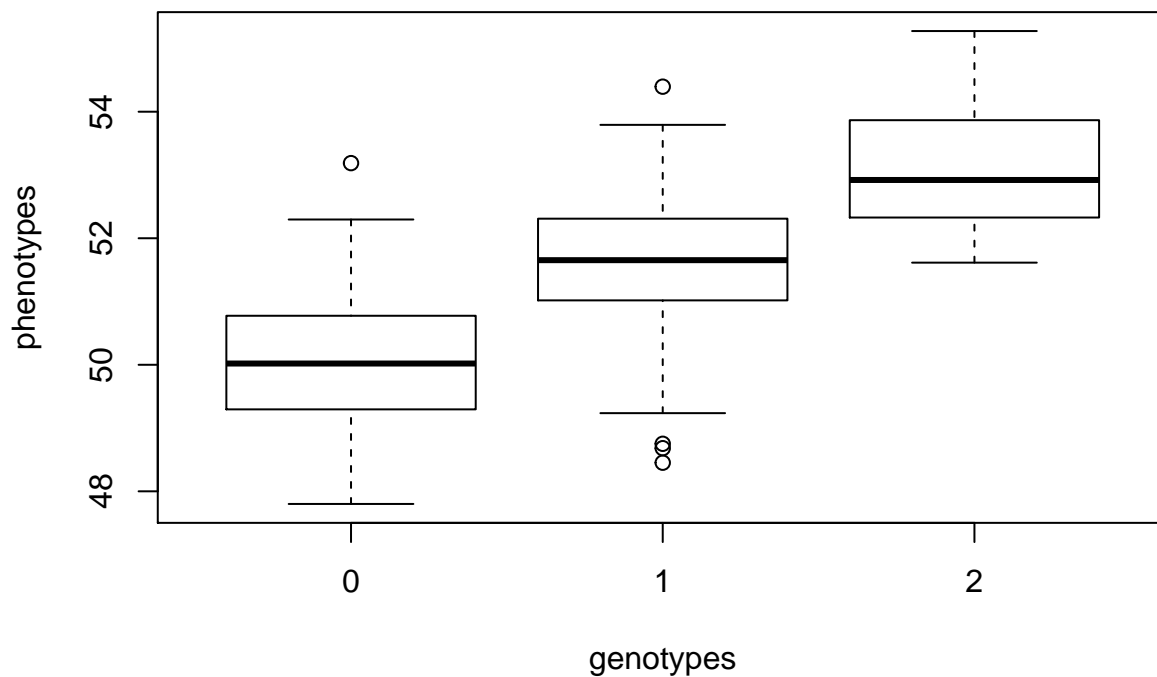
```
hist(phenotypes, breaks="FD")
```

## Histogram of phenotypes



Comme ce qui nous intéresse ici, ce ne sont pas uniquement les phénotypes, mais bien la relation entre les génotypes et les phénotypes, un autre type de graphique semble plus approprié:

```
boxplot(phenotypes ~ genotypes, xlab="genotypes", ylab="phenotypes")
```



Pour la suite, il est habituel dans R d'organiser les données dans un tableau:

```
dat <- data.frame(x=genotypes, y=phenotypes)
summary(dat)
```

```
##           x           y
## Min.      :0.000   Min.   :47.8
## 1st Qu.:0.000   1st Qu.:49.8
## Median :0.000   Median :50.8
## Mean     :0.548   Mean    :50.9
## 3rd Qu.:1.000   3rd Qu.:51.8
## Max.     :2.000   Max.    :55.3
```

## 6 Réaliser l'inférence

### 6.1 Dérivation mathématique

...

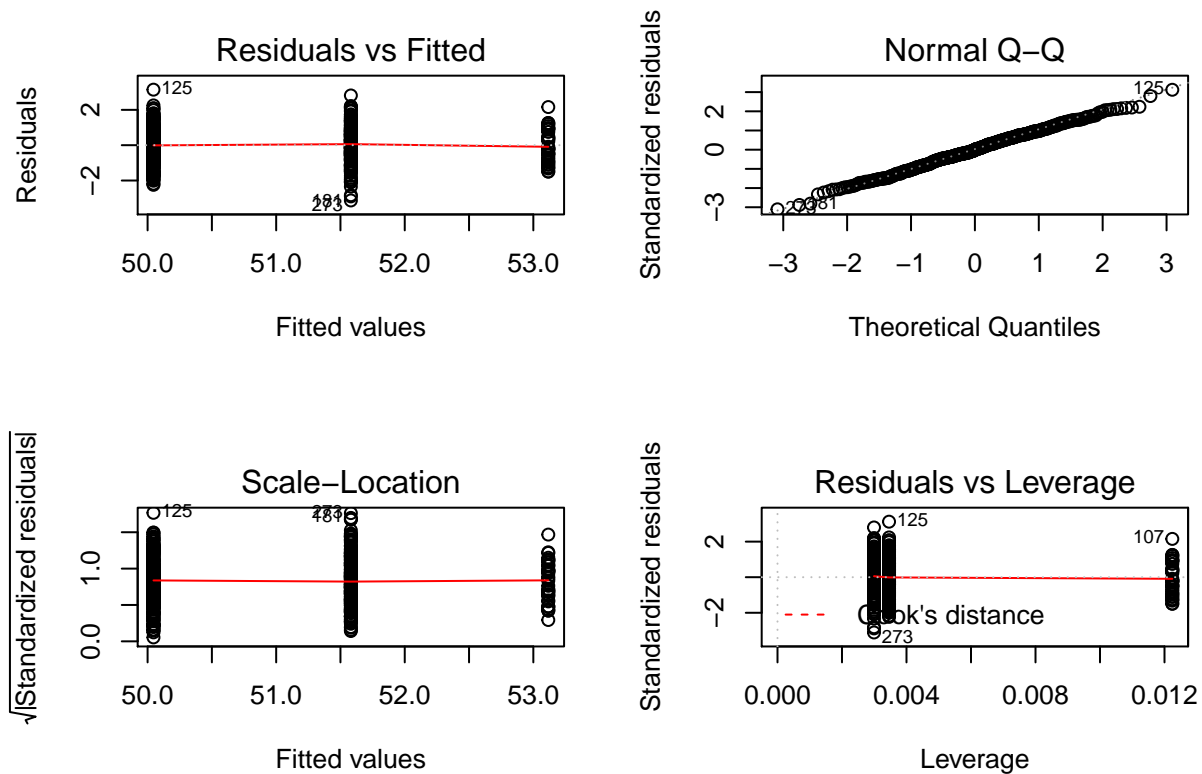
### 6.2 Implémentation (R stats)

Bien entendu, R a nativement une fonction implémentant l'estimation par maximum de vraisemblance d'un modèle linéaire:

```
fit <- lm(y ~ x, data=dat)
```

Avant toute autre chose, il nous faut vérifier que les hypothèses du modèles sont vérifiées (homoscédasticité, Normalité, indépendance):

```
par(mfrow=c(2,2))
plot(fit)
```



Ceci semble être bien le cas (évidemment puisque nous avons simulé nous-même les données...).

Nous pouvons donc extraire de l'objet les quantités qui nous intéressent et les comparer aux vraies valeurs utilisées pour simuler les données:

```
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1284 -0.6718 -0.0062  0.7347  3.1419
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  50.0458    0.0593   844.1   <2e-16 ***
## x             1.5352    0.0703   21.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.01 on 498 degrees of freedom
## Multiple R-squared:  0.489, Adjusted R-squared:  0.488
## F-statistic: 477 on 1 and 498 DF, p-value: <2e-16
```

```
c(mu, beta, sigma)
```

```
## [1] 50.00  1.51  1.00
```



```
(mu.hat <- coefficients(fit)[1])

## (Intercept)
##          50

(beta.hat <- coefficients(fit)[2])

##      x
## 1.54

(sigma.hat <- sqrt((1/(n-2) * sum(fit$residuals^2))))

## [1] 1.01
```

### 6.3 Implémentation (R base)

Il est bon, surtout dans ce cas simple, d'implémenter par nous même cette méthode, histoire de mieux la comprendre.

TODO

## 7 Explorer les simulations possibles

La simulation est un outil particulièrement utile pour explorer comment un modèle répond à des changements dans les données.

Maintenant, c'est à vous: que voudriez-vous explorer en premier?

## 8 Perspectives

Naturellement, l'activité de modélisation statistique ne se limite pas à simuler des données sur ordinateur. Bien au contraire, elle est au coeur de l'activité de recherche en ce qu'elle vise à identifier les caractéristiques saillantes d'un phénomène naturel afin d'en réaliser l'inférence.

Concernant le thème de l'atelier, la prédiction génomique, quelles sont les limites du modèle exploré ci-dessus? Que proposez-vous pour y remédier?

## 9 Appendix

```
print(sessionInfo(), locale=FALSE)

## R version 3.1.2 (2014-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
##
## other attached packages:
## [1] MASS_7.3-37      knitr_1.9         rmarkdown_0.4.2
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.8      evaluate_0.5.5    formatR_1.0       htmltools_0.2.6
## [5] stringr_0.6.2     tools_3.1.2       yaml_2.1.13
```