AgroParisTech

Régression linéaire

Eric Marcon

23 février 2024

Régression linéaire simple

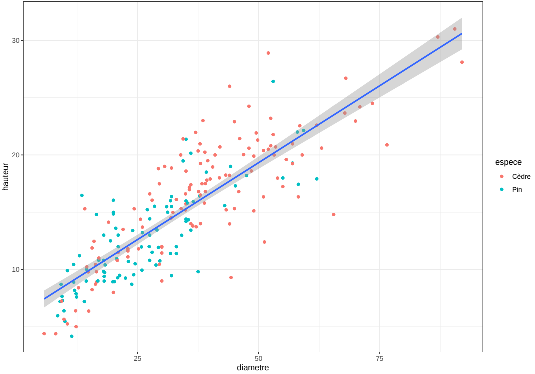
Ventoux

Données du projet de dendrométrie 2020, Mont Ventoux.

```
read_csv2("data/Inv_GEEFT_Ventoux_09-2020.csv") |>
  rename(
    espece = Espèce,
    diametre = `Diamètre (cm)`,
    hauteur = `Hauteur réelle (m)`
  ) |>
  mutate(
    espece = case_match(
      espece,
      "P" ~ "Pin",
      "C" ~ "Cèdre"
    )
  ) -> ventoux
```

Graphique hauteur ∼ diamètre


```
ventoux |>
  ggplot(aes(x = diametre, y = hauteur)) +
  geom_point(aes(col = espece)) +
  geom_smooth(method = "lm")
```



Théorie

Modèle linéaire simple :

$$Y = \beta_0 + \beta_1 X + \mathrm{E}$$

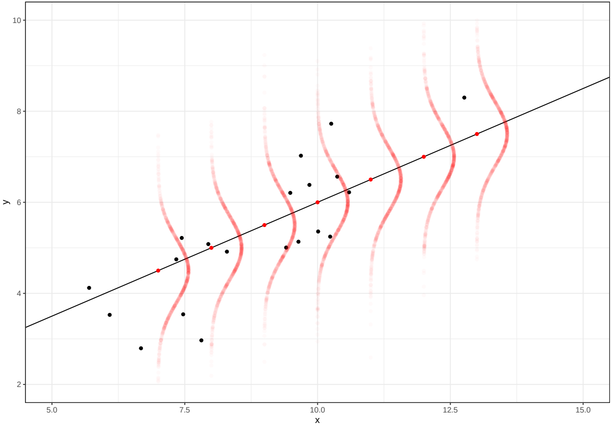
Y et X sont des vecteurs : $Y = \{y_i\}$ est l'ensemble des observations. Par abus d'écriture, Y est aussi la variable aléatoire dont les y_i sont des réalisations.

Vocabulaire : variable expliquée, exogène, coefficients, constante (intercept)...

$\mathrm{E} = \{\epsilon_i\}$ est l'erreur du modèle. $\mathrm{E} \sim \mathcal{N}(0, \sigma^2)$

Représentation

Le modèle prédit une densité de probabilité des valeurs de Y pour toute valeur de X distribuée normalement autour de la droite de régression.



Hypothèses

- Indépendance des erreurs : $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$. Assurée par le design expérimental.
- Exogénéité : X n'est pas corrélé à E.
- Homoscédasticité : la variance de l'erreur est constante sur l'étendue de X.
- Normalité des termes d'erreur : $\mathrm{E} \sim \mathcal{N}(0, \sigma^2)$.

Exemple

Générer les données du modèle.

Coefficients :

```
beta_0 <- 1
beta_1 <- 0.5
sigma <- 1
```

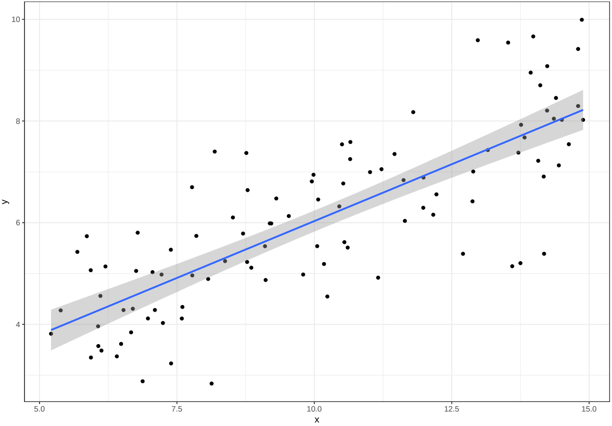
Tirage :

```
n <- 100
x <- runif(n, min = 5, max = 15)
# Jeu de points
mod_l1 <- tibble(x, y = rnorm(n, mean = beta_0 + beta_1*x, sd = sigma))
```

Estimation

Commencer par une figure.

```
mod_l1 |>
  ggplot(aes(x = x, y = y)) + geom_point() + geom_smooth(method = lm)
```



Estimation

La fonction `lm()` du package *stats* estime le modèle et permet de tester les hypothèses.

```
mod_l1_lm <- lm(y ~ x, data = mod_l1)
```

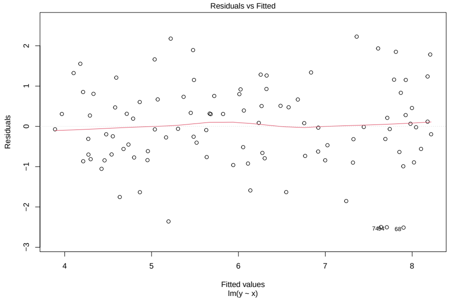
Syntaxe de la formule :

- variable expliquée à gauche, covariables à droite de `~`
- constante implicite `y ~ x` est identique à `y ~ 1 + x` alors que `y ~ 0 + x` force la constante à 0.
- possibilité de transformer les variables : `log(y) ~ I(x^2)` (Attention : `log(y) ~ x^2` est interprété comme l'interaction de `x` avec lui-même, c'est-à-dire `x`)

Homoscédasticité et indépendance des erreurs

Graphique $\mathrm{E} \sim Y^{\star}$

```
plot(mod_l1_lm, which = 1)
```

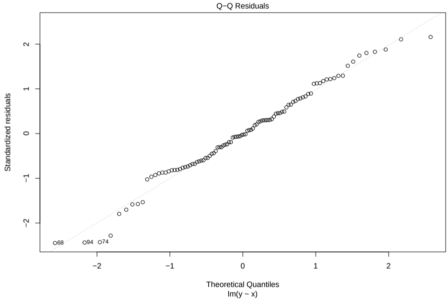


Les erreurs doivent être centrée sur 0 et uniformément réparties.

Normalité des erreurs

Graphique quantile - quantile (`?qqplot`)

```
plot(mod_l1_lm, which = 2)
```



La non-normalité des résidus implique la non-normalité des estimateurs des coefficients.

Test de normalité

Utiliser le test de Shapiro-Wilk :

```
mod_l1_lm |> residuals() |> shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mod_l1_lm)
## W = 0.98344, p-value = 0.2439
```

La p-value est la probabilité de se tromper en rejetant l'hypothèse nulle de normalité des données. Attention : la puissance du test augmente avec la taille de l'échantillon (limité à 5000).

Test de Kolmogorov-Smirnov

Teste l'hypothèse que deux échantillons sont issus de la même distribution :

```
mod_l1_lm |> residuals() %>% ks.test(rnorm(length(.), 0, var(.)))
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov
##  test
##
## data:  . and rnorm(length(.), 0, var(.))
## D = 0.11, p-value = 0.5806
## alternative hypothesis: two-sided
```

Plus général que Shapiro-Wilk.

→ tester un tirage dans une loi uniforme contre une distribution normale. Combien de valeurs faut-il pour rejeter H0 ?

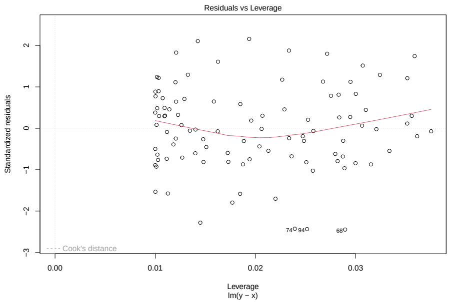
AgroParisTech

Régression linéaire

Eric Marcon

Effet de levier

```r
plot(mod_l1_lm, which = 5)
```



Les points avec fort effet de levier forte erreur ($\rightarrow$ grande distance de Cook) posent problème.

Rectification des données

Affaire d'expérience.

- Éliminer les points (réellement) aberrants ;
- Transformer Y si :
  - la relation n'est pas linéaire (ex.: quadratique) ;
  - l'erreur augmente avec $Y^\star$ ($\rightarrow$ racine carrée ou logarithme).
- Revoir les hypothèses à l'origine du modèle, le design expérimental...

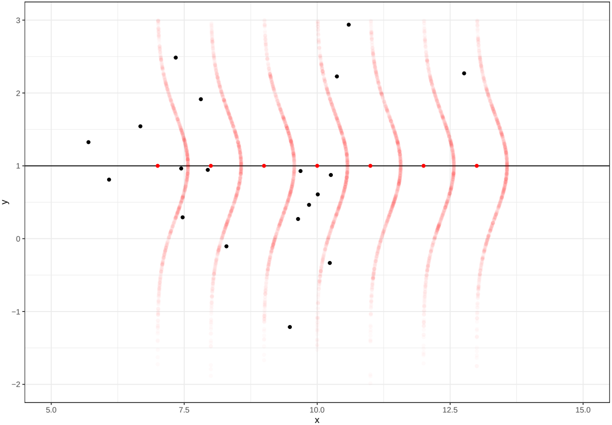
Interprétation des résultats : summary


```
##
## Call:
## lm(formula = y ~ x, data = mod_l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51229 -0.69782 -0.02673  0.68502  2.22763
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.55838    0.37138   4.196 5.97e-05
## x            0.44729    0.03525  12.689  < 2e-16
##
## (Intercept) ***
## x           ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.042 on 98 degrees of freedom
## Multiple R-squared:  0.6216, Adjusted R-squared:  0.6178
## F-statistic:   161 on 1 and 98 DF,  p-value: < 2.2e-16
```

Statistique F

La statistique F décrit la probabilité que le modèle n'explique rien.

Modèle nul: $Y = \bar{Y} = \beta_0$



R^2

R^2 mesure la proportion de la variance de Y expliquée par le modèle :

$$R^2 = \frac{\mathrm{Var}(Y^\star)}{\mathrm{Var}(Y)} = 1 - \frac{\sigma^2}{\mathrm{Var}(Y)}$$

$\rightarrow$ Que devient R^2 en doublant σ ? Estimer rapidement puis re-simuler le modèle pour vérifier.

R^2 ajusté pénalise le R^2 par le nombre de paramètres du modèle.

Les degrés de liberté sont le nombre d'observations moins le nombre de paramètres moins 1.

Estimation des coefficients

Les coefficients sont estimés par la méthode des moindres carrés : minimisation des écarts

$$\sum (y_i - y_i^\star)^2$$

.

Résultat identique à la maximisation de la vraisemblance

$$\prod f(\epsilon_i)$$

où $f(\dot{})$ est la densité de $\mathcal{N}(0, \sigma^2)$.

Estimation des coefficients

L'estimateur de chaque coefficient est sa valeur la plus probable.

L'estimateur est distribué normalement (quand E est normal) :

$$\hat{\beta}_1 \sim \mathcal{N}(0.447, \sigma_1^2)$$

où σ_1 est l'écart-type de l'estimateur. `lm()` donne son erreur standard, c'est-à-dire $\sigma_1/\sqrt{n}$.

Un test de Student donne la probabilité de se tromper en affirmant que l'estimateur n'est pas nul.

Synthèse 1/2

Un bon modèle a un grand R^2 et des petites p-values.

- R^2 diminue avec la variance de l'erreur ;
- L'écart-type des estimateur diminue comme $\sqrt{n}$.

Mais les deux dépendent du design expérimental.

Design expérimental

Quadrupler l'effort d'échantillonnage divise par deux l'intervalle de confiance

```
mod_l1x4 <- tibble(
  x = rnorm(n * 4, mean = 10, sd = 2), # x est calculé avant y
  y = rnorm(n * 4, mean = beta_0 + beta_1 * x, sd =sigma) # y utilise x
)
mod_l1x4_lm <- lm(y ~ x, data = mod_l1x4)
summary(mod_l1x4_lm)$coefficients
```

```
##              Estimate Std. Error   t value
## (Intercept) 1.4740364 0.26412629  5.580801
## x           0.4558641 0.02579445 17.672952
##                 Pr(>|t|)
## (Intercept) 4.434776e-08
## x           5.182547e-52
```

Choix économique.

Design expérimental

Retirer les valeurs intermédiaires de X augmente le R^2 (*design factoriel*) alors que σ ne change pas.

```
mod_l1x4 |>
  filter(x < 6 | x >14) %>% # pas |> pour "data = ."
  lm(y ~ x, data = .) |>
  summary() |>
  pluck("r.squared")
```

```
## [1] 0.8281414
```

contre 0.4396996 avec toutes les données.

Design expérimental

Le R^2 d'un modèle avec des données individuelles est plus faible qu'avec des données agrégées.

$\rightarrow$ Estimer le modèle hauteur $\sim$ diamètre des données Ventoux.

$\rightarrow$ Regrouper les données par espèce.

$\rightarrow$ Estimer le modèle à nouveau.

Synthèse 2/2

Considérer R^2 et p-values en fonction du modèle :

- beaucoup de données individuelles $\rightarrow$ faible R^2 mais petites p-values pour montrer l'influence d'un facteur ;
- possibilité d'un très grand R^2 sans aucun coefficient significatif si peu de points ;
- un grand R^2 et des petites p-values permettent de faire des prédictions.

Prédictions

`predict()` permet d'extrapoler le modèle.

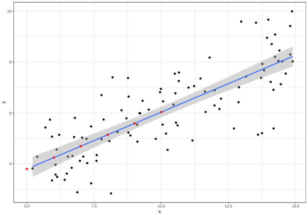
```
mod_l1_lm |> predict(newdata = data.frame(x = 5:10))
```

```
##        1        2        3        4        5
## 3.794805 4.242090 4.689375 5.136661 5.583946
##        6
## 6.031231
```

Prédictions

Ajout des points sur la figure :

```
# Estimation du modèle
mod_l1 |>
  ggplot(aes(x, y)) + geom_point() + geom_smooth(method = lm) ->
  mod_l1_ggplot
# Choix des x pour lesquels y est à prédire
mod_l1_predict <- data.frame(x = 5:10)
# Ajout des prédictions
mod_l1_predict$y <- predict(mod_l1_lm, newdata = mod_l1_predict)
# Ajout des points à la figure précédente
mod_l1_ggplot +
  geom_point(data = mod_l1_predict, aes(x = x, y = y), col = "red")
```

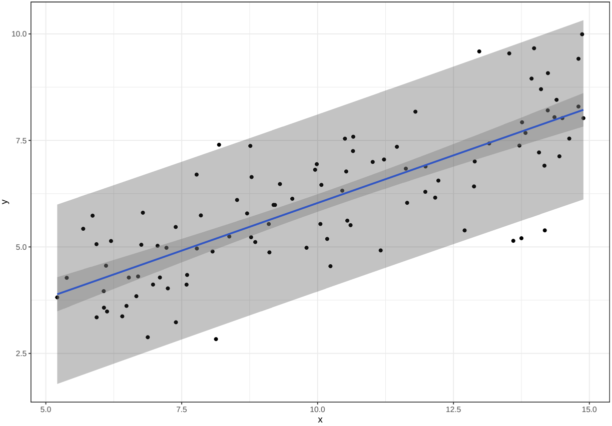


Intervalles de confiance et de prédiction

La zone grisée de `geom_smooth` est l'intervalle de confiance de l'espérance de $Y|X$, c'est-à-dire de la moyenne des prédictions. Il est bien plus étroit que l'intervalle de prédiction, qui correspond à 95% des prédictions :

```
mod_l1_predict <- data.frame(
  x = seq(from = min(mod_l1$x), to = max(mod_l1$x), length.out = 50)
)
mod_l1_predict <- cbind(
  mod_l1_predict,
  predict(
    mod_l1_lm,
    newdata = mod_l1_predict,
    interval = "prediction"
  )
)
mod_l1_ggplot +
  geom_ribbon(
    data = mod_l1_predict,
    aes(y = fit, ymin = lwr, ymax = upr),
    alpha = 0.3
  ) -> mod_l1_ggplot_predict
```

Intervalles de confiance et de prédiction



Régression linéaire multiple

Théorie

Modèle linéaire multiple :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \mathrm{E}$$

Y et X_j sont des vecteurs : $X_1 = \{x_{i,1}\}$ est l'ensemble des valeurs du premier prédicteur (= variable explicative, variable exogène ou covariable).

Représentation

Multidimensionnelle donc plus difficile.

Le dimension de Y est égale au nombre de covariables moins 1: le modèle linéaire réduit la dimension des données.

Ajout d'un coefficient à l'exemple précédent :

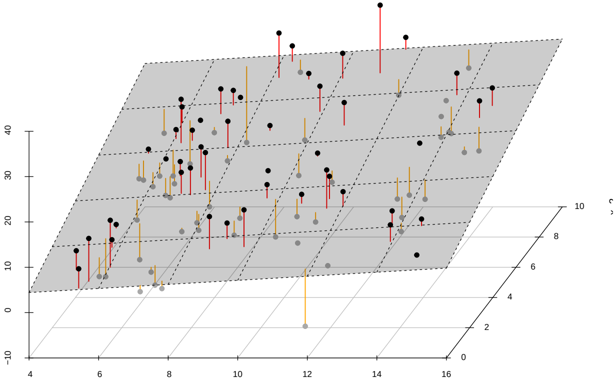
```
beta_0 <- 1
beta_1 <- 0.5
beta_2 <- 2
sigma <- 5
```

Tirage :

```
n <- 100
x_1 <- runif(n, min = 5, max = 15)
x_2 <- runif(n, min = 0, max = 10)
# Jeu de points
mod_l2 <- tibble(
  x_1, x_2,
  y = rnorm(n, mean = beta_0 + beta_1*x_1 + beta_2*x_2, sd = sigma)
)
```

Représentation

```
mod_l2_lm  <- lm(y ~ x_1 + x_2, data = mod_l2)
```



Hypothèses

En plus des précédentes :

- Non colinéarité des covariables.

Si une des covariables est une combinaison linéaire des autres, le modèle ne peut pas être estimé.

En pratique, les covariables doivent être aussi peu corrélées que possible.

Interactions

On peut tester l'effet de l'interaction de deux variables :

```
lm(y ~ x_1 + x_2 + x_1:x_2, data = mod_l2) |> # Identique à x_1*x_2
  summary()
```

```
##
## Call:
## lm(formula = y ~ x_1 + x_2 + x_1:x_2, data = mod_l2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8144  -3.7527   0.0555   3.4683  14.3491
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.07576    3.40857   1.782   0.0778
## x_1          0.07438    0.35391   0.210   0.8340
## x_2          0.96892    0.65142   1.487   0.1402
## x_1:x_2      0.08070    0.06674   1.209   0.2296
##
## (Intercept) .
## x_1
## x_2
## x_1:x_2
```

Standardisation

Il est possible de standardiser toutes les variables pour comparer les effets des covariables.

```
lm(scale(y) ~ 0 + scale(x_1) + scale(x_2), data = mod_l2) |> summary()
```

```
##
## Call:
## lm(formula = scale(y) ~ 0 + scale(x_1) + scale(x_2), data = mod_l2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.35612 -0.53383 -0.02476  0.48399  2.10158
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## scale(x_1)  0.19407    0.07379   2.630  0.00991
## scale(x_2)  0.64478    0.07379   8.738 6.58e-14
##
## scale(x_1) **
## scale(x_2) ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Standardisation

Dans un modèle linéaire standardisé simple, le coefficient égale la corrélation.

```
lm(scale(y) ~ scale(x), data = mod_l1) |> summary()
```

```
##
## Call:
## lm(formula = scale(y) ~ scale(x), data = mod_l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49102 -0.41415 -0.01586  0.40655  1.32207
##
## Coefficients:
##               Estimate Std. Error t value
## (Intercept) -1.163e-16  6.182e-02    0.00
## scale(x)     7.884e-01  6.214e-02   12.69
##             Pr(>|t|)
## (Intercept)        1
## scale(x)      <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test de la corrélation

La significativité de la corrélation entre deux variables est celle du coefficient de la régression standardisée.

Plus simplement :

```
with(mod_l1, cor.test(x, y))
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 12.689, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7005033 0.8527907
## sample estimates:
##       cor
## 0.7884389
```

Régression sur les rangs

Théorie

Si les résidus ne sont pas normaux, il est possible de faire la régression sur les rangs des variables :

- régression simple : revient à tester la corrélation de Spearman.

Exemple

Modèle univarié :

```r
lm(rank(y) ~ rank(x), data = mod_l1)|> summary()
```

```
##
## Call:
## lm(formula = rank(y) ~ rank(x), data = mod_l1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.203 -11.867   1.019  11.140  41.659
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.22667    3.54507   2.885  0.00482
## rank(x)      0.79749    0.06095  13.085  < 2e-16
##
## (Intercept) **
## rank(x)     ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.59 on 98 degrees of freedom
```

Transformation de variables

Principe

Le modèle linéaire permet de traiter des modèles non linéaires en transformant les variables.

Exemple : le volume V d'un arbre est lié à son diamètre D à la puissance β_1

$\rightarrow$ Modèle :

$$\ln(V) = \beta_0 + \beta_1 \ln(D) + \mathrm{E}$$

Pratique

Dans la formule de `lm()`, certains opérateurs sont compris, d'autres non : essayer.

```
# Données
n <- 100
D <- runif(n, min = 10, max = 50)
V <- exp(2.5 * log(D)) + rnorm(n)
# Modèle
lm(log(V) ~ log(D))
```

```
##
## Call:
## lm(formula = log(V) ~ log(D))
##
## Coefficients:
## (Intercept)       log(D)
##    0.000175     2.499946
```

Pratique

Autres écritures :

- `I(D^2.5)` : le contenu de `I()` peut être n'importe quel calcul valide

```
lm(V ~ I(D^2.5))
```

- `poly(D, degree = 3)` : toutes les puissances de D jusqu'à 3.

```
lm(V ~ poly(D, degree = 3))
```

Pourquoi transformer

Modèle mécaniste : la relation entre volume et diamètre est à la puissance 3 si l'arbre est un cylindre.

Contrôle de la variance : dans certains cas, la variance augmente avec $Y^\star$. On peut essayer de régresser $\sqrt{Y}$ ou $\ln(Y)$.

Mais on ne doit pas tenter à l'aveugle toutes les transformations possibles : voir le problème des tests multiples dans le cours sur l'Anova.

Ventoux transformé

Les modèles allométriques prévoient que la hauteur des arbres est liée au diamètre à une puissance inférieure à 1 : plus l'arbre est grand, moins il a besoin d'investir en hauteur et plus en diamètre.

Le modèle est alors

$$\ln(Y) = \beta_0 + \beta_1 \log(X_1) + \mathrm{E}$$

- Y est la hauteur des arbres ;
- X_1 est leur diamètre ;
- β_1 est la puissance dans le modèle $Y \sim X_1^{\beta_1}$

Ventoux transformé

```
(ventoux_lm <- lm(log(hauteur) ~ log(diametre), data = ventoux))
```

```
##
## Call:
## lm(formula = log(hauteur) ~ log(diametre), data = ventoux)
##
## Coefficients:
##   (Intercept)  log(diametre)
##        0.5453         0.6177
```

Ce modèle s'ajuste mieux aux données.

Il sera étudié en cours de dendrométrie.

Ancova

Théorie

Modèle de régression multiple avec des covariables catégorielles, codées sous forme d'indicatrices (autant d'indicatrices que de modalités - 1).

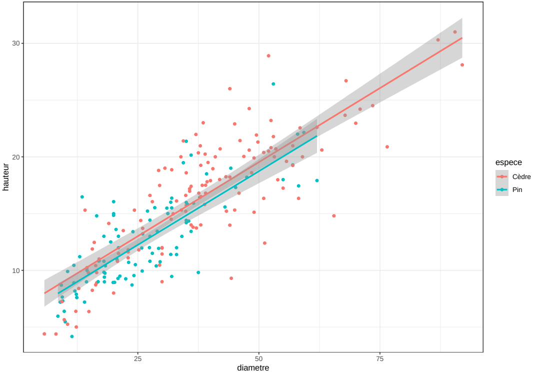
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \mathrm{E}$$

Exemple du Ventoux :

- Y est la hauteur des arbres ;
- X_1 est leur diamètre ;
- L'espèce est codée par une variable indicatrice, par exemple $X_2 = \mathbb{1}('Cedre')$.

Exemple


```
ventoux |>
  ggplot(aes(x = diametre, y = hauteur, color = espece)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Estimation

La figure représente *deux régressions séparées* : les pentes pourraient être différentes. Une Ancova est donc appropriée.

`lm` crée automatiquement des indicatrices pour les variables catégorielles.

```
(ventoux_lm <- lm(hauteur ~ diametre + espece, data = ventoux))
```

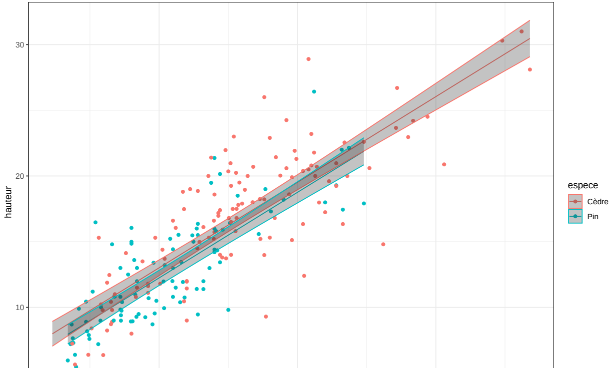
```
##
## Call:
## lm(formula = hauteur ~ diametre + espece, data = ventoux)
##
## Coefficients:
## (Intercept)     diametre    especePin
##      6.5018       0.2605      -0.7696
```

Ici, l'indicatrice vaut 1 pour les pins, 0 pour les cèdres.

Représentation

La figure doit être construite manuellement

```
ventoux |>
  bind_cols(predict(ventoux_lm, interval = "confidence")) |>
  ggplot(aes(x = diametre, color = espece)) +
    geom_point(aes(y = hauteur)) +
    geom_line(aes(y = fit)) +
    geom_ribbon(aes(y = fit, ymin = lwr, ymax = upr),  alpha = 0.3)
```



Sélection de modèle

Théorie

Un modèle avec trop peu de covariables est sous-ajusté. Il explique mal Y, avec une erreur qui ne diminue pas quand le nombre d'observations augmente : on parle de *biais*. Cas extrême : $Y = \beta_0 + \mathrm{E}$

Un modèle avec trop de covariables est sur-ajusté. Avec le même nombre d'observations qu'un modèle plus simple, ses coefficients sont une *variance* plus grande. Cas extrême : $Y = \beta_0 + \sum_{i=1}^{n-1} \beta_i X_i + \mathrm{E}$

Théorie

Beaucoup de méthodes pour choisir le “meilleur” modèle, solide support théorique.

Critère d'Information d'Akaike (AIC) : $2K - 2\ln(L)$ où L est la vraisemblance et K le nombre de paramètres (les β_i et σ).

Critère AICc pour de petits échantillons :

$$2K\frac{n}{n-K-1} - 2\ln(L)$$

Exemple

Faut-il ajouter le paramètre espèce au modèle Ventoux ?

On peut calculer l'AICc d'un modèle

```
library("AICcmodavg")
lm(hauteur ~ diametre, data = ventoux) |>
  AICc()
```

```
## [1] 1108.679
```

Pour la comparaison, une liste de modèles est nécessaire :

```
ventoux_lm_list <- list(
  nul = lm(hauteur ~ 1, data = ventoux),
  diametre = lm(hauteur ~ diametre, data = ventoux),
  diamespece = lm(hauteur ~ diametre + espece, data = ventoux),
  complet = lm(hauteur ~ diametre*espece, data = ventoux)
)
```

Exemple

```
aictab(ventoux_lm_list)
```

```
##
## Model selection based on AICc:
##
##            K    AICc Delta_AICc AICcWt Cum.Wt
## diamespece 4 1107.28       0.00   0.54   0.54
## diametre   3 1108.68       1.40   0.27   0.81
## complet    5 1109.37       2.09   0.19   1.00
## nul        2 1382.78     275.50   0.00   1.00
##                 LL
## diamespece -549.55
## diametre   -551.28
## complet    -549.55
## nul        -689.36
```

Le meilleur modèle est celui avec l'espèce mais sans l'interaction.

Les poids permettent des prédictions multi-modèles.

Exemple

Prédiction pour de nouvelles valeurs:

```
ventoux_nouveau <- data.frame(
  diametre = c(20, 50),
  espece = c("Pin", "Cèdre")
)
modavgPred(ventoux_lm_list, newdata = ventoux_nouveau)
```

```
##
## Model-averaged predictions on the response scale
## based on entire model set and 95% confidence interval:
##
##   mod.avg.pred uncond.se lower.CL upper.CL
## 1       11.032     0.329   10.387   11.677
## 2       19.474     0.301   18.884   20.064
```

Sélection systématique

Sélection (*backward*) :

- Estimer le modèle complet,
- Retirer la covariable qui fait le plus diminuer l'AIC jusqu'à ce qu'il ne diminue plus

Élimination (*forward*) :

- Estimer le modèle nul,
- Ajouter la covariable qui fait le plus diminuer l'AIC jusqu'à ce qu'il ne diminue plus.

Mixte (*stepwise*):

- Élimination puis sélection successives.

AgroParisTech

Exemple

Régression linéaire

Eric Marcon

Critère AIC, pas AICc :

```
library("MASS")
stepAIC(lm(hauteur ~ diametre * espece, data = ventoux))
```

```
## Start:  AIC=474.25
## hauteur ~ diametre * espece
##
##                    Df Sum of Sq    RSS    AIC
## - diametre:espece  1  0.020844 1804.4 472.25
## <none>                         1804.4 474.25
##
## Step:  AIC=472.25
## hauteur ~ diametre + espece
##
##            Df Sum of Sq    RSS    AIC
## <none>                  1804.4 472.25
## - espece    1      28.3 1832.7 473.72
## - diametre  1    3694.8 5499.2 718.76

##
## Call:
## lm(formula = hauteur ~ diametre + espece, data = ventoux)
##
```