Notions de modélisation statistique

Benoit Crabbé

Motivations

- Introduction à la modélisation statistique
  - Modèles de régression (linéaire ici)
  - Modèles de classification (logistique et arbres de décision ici)
- Très utilisés en TAL et en linguistique empirique. Les usages diffèrent par endroits : ici c'est la version linguistique empirique qui préside, on focalise sur l'interprétation du modèle plus que sur sa capacité à réaliser une tâche de classification le mieux possible.

Exemple

- Alternance dative en anglais :
  1. John gave a book to Mary (V-NP-PP)
  2. John gave Mary a book (dative shift, V-NP-NP)
- Thème ; Bénéficiaire

Problème trop difficile ?

Choix : quels facteurs interviennent pour préférer tel ou tel ordonnancement ?

Principales Variables

- **Accessibilité dans le discours** (*given,new,accessible*) (pour le thème et le bénéficiaire)
- **Définitude** (pour le thème et le bénéficiaire)
- **Pronominalité des dépendants** (pour le thème et le bénéficiaire)
- **Animacité des dépendants** (pour le thème et le bénéficiaire)
- **Classe sémantique du verbe** *abstrait, transfert de possession, futur transfert de possession, prévention de possession,communication*
- **Interaction de complexité** entre le thème et le bénéficiaire : différence de longueur (manipulée au *log*() pour écraser les outliers)
- **Personne des dépendants** (pour les pronominaux)
- **Parallélisme dans le dialogue** priming ?

Devoir

- Récupérer le papier :

  Joan Bresnan, Anna Cueni, Tatiana Nikitina, and Harald Baayen. 2007. "Predicting the Dative Alternation." In Cognitive Foundations of Interpretation, *ed. by G. Boume, I. Kraemer, and J. Zwarts. Amsterdam: Royal Netherlands Academy of Science, pp. 69–94.*

- `http://www.stanford.edu/~bresnan/CFI04.pdf`
- Résumé (2 pages)
- Sera suivi d'un travail qui demande de répliquer certaines analyses dans ce papier...

Plan

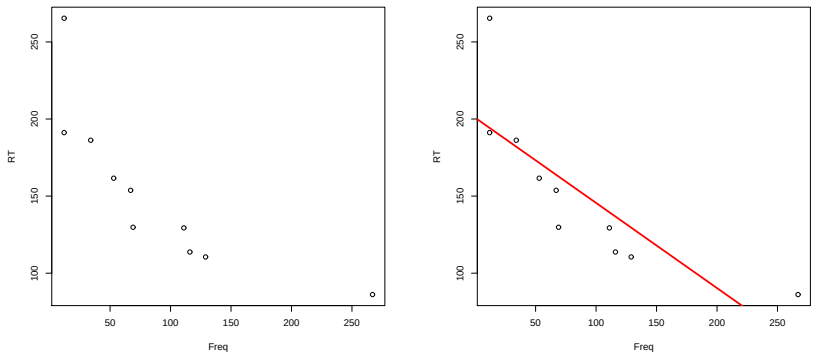
Exemple (artificiel)

- Imaginons que l'on ait des observations sous forme de table (temps de lecture et fréquence pour différents mots)

	rt	freq
1	265.38	12.00
2	191.17	12.00
3	186.23	34.00
4	161.61	53.00
5	153.75	67.00
6	129.79	69.00
7	129.38	111.00
8	113.73	116.00
9	110.54	129.00
10	86.14	267.00

Exemple artificiel (continué)

- On peut en faire une représentation graphique :



- Le calcul de régression consiste à
  - calculer une droite qui passe le mieux possible entre les points
  - utiliser cette droite pour prédire au mieux de nouvelles valeurs

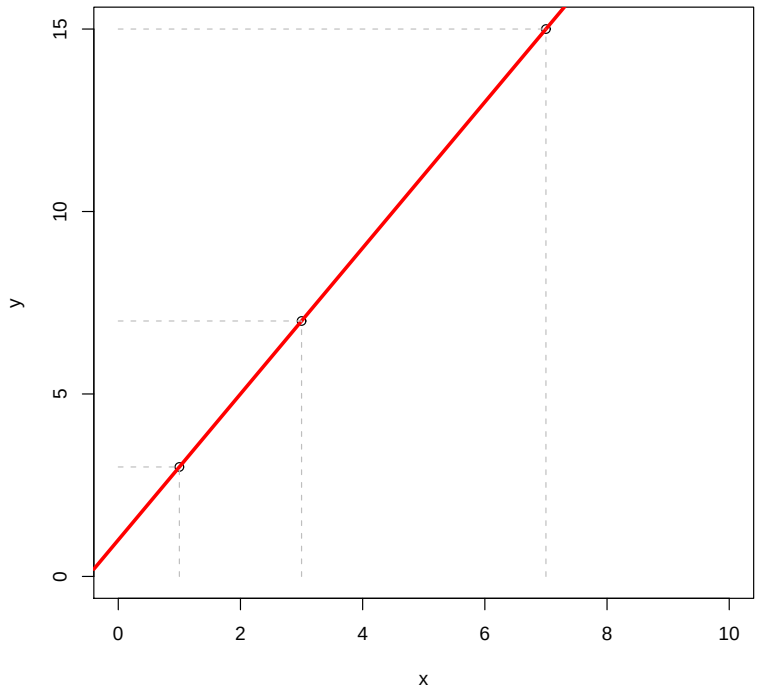
Interprétation

Les points représentent des données (d'entrainement) ; la droite représente un modèle plus abstrait construit à partir des données qui permet de prédire le temps de lecture à partir de la fréquence pour de nouvelles valeurs

Plan

La fonction comme prédicteur

- Une fonction au sens mathématique, permet de prédire une valeur y à partir d'une valeur x:

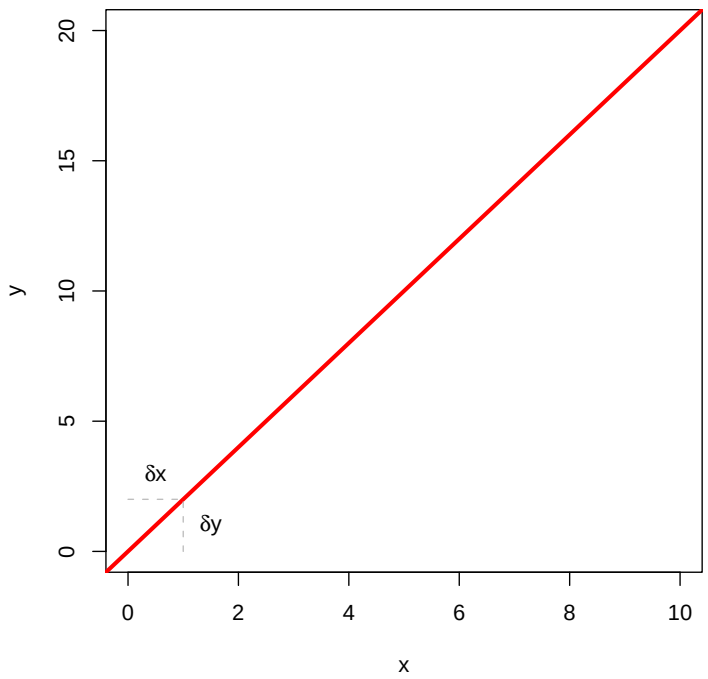


$$f(x) = 2x + 1$$

$$f(1) = 2 + 1 = 3;$$
$$f(3) = 6 + 1 = 7;$$
$$f(7) = 14 + 1 = 15$$

Anatomie d'une fonction linéaire (pente)

- Une fonction linéaire se caractérise par son coefficient de pente β et son décalage à l'origine α

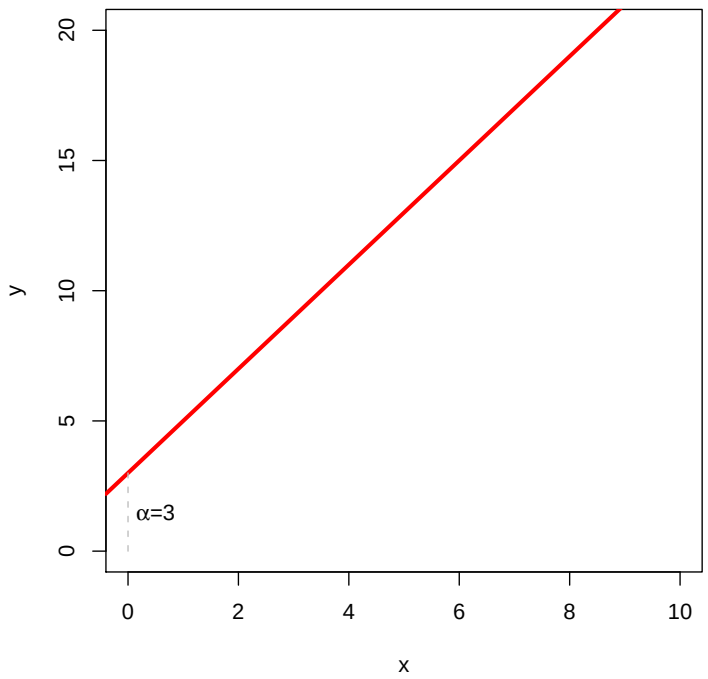


Coefficient de pente

- Le coefficient de pente $\beta = \frac{\delta y}{\delta x}$ indique l'inclinaison de la droite. Plus le coefficient est élevé plus la droite est inclinée.
- Dans le cas où $\delta x = 1$, on a que $\beta = \delta y$: le coefficient indique donc de combien d'unité y varie par incrément unitaire de x

Anatomie d'une fonction linéaire (intercept)

- L'intercept de la droite représente la translation de la droite par rapport à l'origine :



Coefficient de pente

- L'intercept de la pente α indique la translation de la droite par rapport à l'origine : nul, la droite passe par l'origine, positif, la droite passe au-dessus de l'origine négatif, la droite passe sous l'origine.

Anatomie d'une droite

- Une droite est une fonction de la forme :

$$y = \alpha + \beta x$$

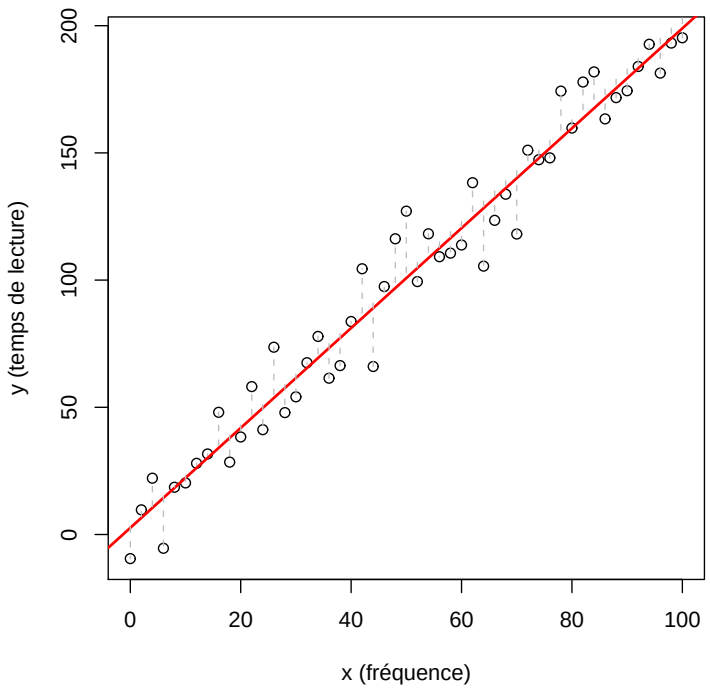
- α est le coefficient d'interception (de l'axe y)
- β est le coefficient de pente

Calcul de la régression

- La droite à trouver a les coordonnées:

$$y = \alpha + x\beta$$

- Chaque point y_i est éloigné de la droite idéale d'une distance $y_i - \hat{y} = \epsilon_i$. On appelle ϵ_i un **résidu** de régression.



Calcul de la droite

- Pour faire le calcul, on fait l'hypothèse que les résidus sont distribués normalement autour de la droite $\epsilon \sim \mathcal{N}(0, \sigma)$.
- Le calcul consiste à trouver α, β telle que la somme des carrés des résidus soit minimale:

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha,\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \underset{\alpha,\beta}{\operatorname{argmin}} \sum_{i=1}^{n} ((\alpha + \beta x) - y_i)^2$$

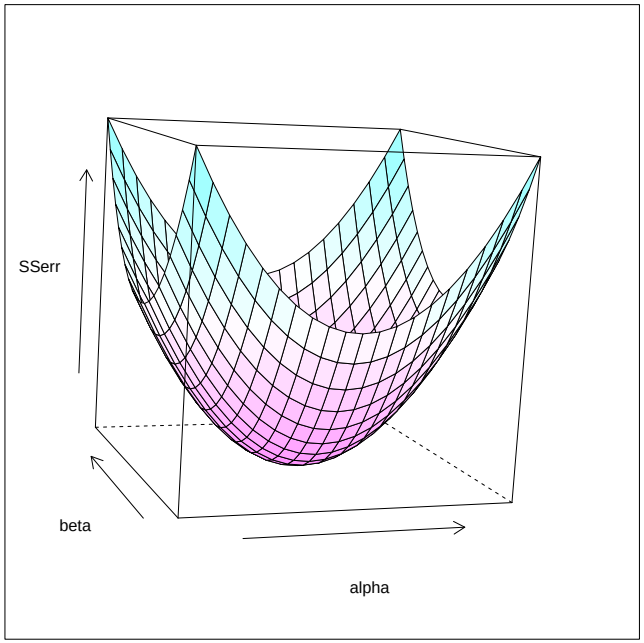
- Cette méthode de calcul s'appelle méthode des moindres carrés.

Problèmes d'optimisation

Les moindres carrés est sans doute un des problèmes d'optimisation les plus simples. Beaucoup d'algorithmes d'apprentissage cherchent à minimiser des fonctions (souvent plus complexes) du même genre.

Fonction convexe

- La fonction à minimiser (la somme du carré des erreurs, SS_{err}) est une fonction à deux variables α et β dont on cherche les valeurs de α et β qui la minimisent :



Solution analytique

Dans le cas décrit ici on a une solution analytique:

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Calculer une droite de régression avec R

```
#Donnees artificielles
> x <- rnorm(100,5,2)
> y <- rnorm(100,3,1)
> plot(x,y)
#Regression
> reg <- lm(y ~ x) #fait le calcul pour vous
> abline(reg)
```

Modèle linéaire

- Le modèle linéaire a la forme :

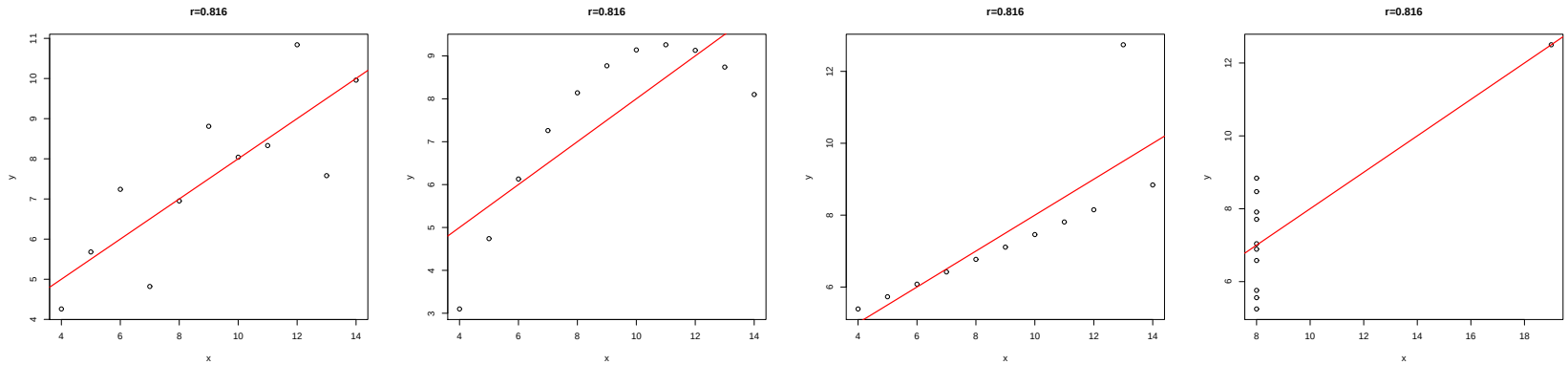
$$y = \alpha + \beta x + \epsilon$$

- En pratique, quand on calcule une régression, il faut vérifier (entre autres) que les hypothèses du calcul sont vérifiées, en particulier la normalité des résidus : $\epsilon \sim \mathcal{N}(0, \sigma)$
- Quand on a trouvé la droite et vérifié que le modèle est valide, on peut interpréter les coefficients, par exemple si β est positif, on sait qu'à chaque incrément unitaire de x, y augmente de β.

Plan

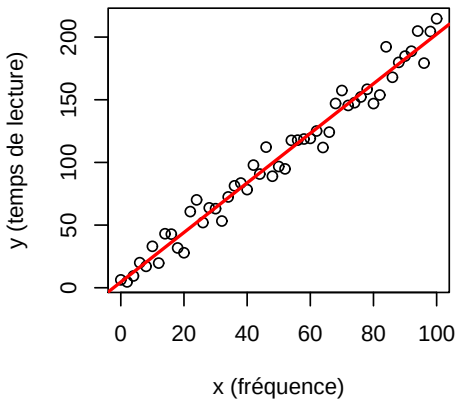
Vérifier l'allure des données

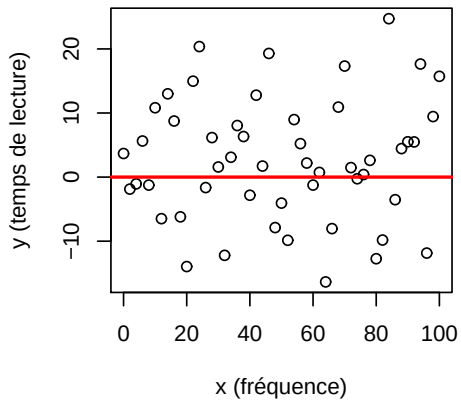
Toutes les relations entre variables ne sont pas nécessairement linéaires, ou parfois les données peuvent présenter des anormalités

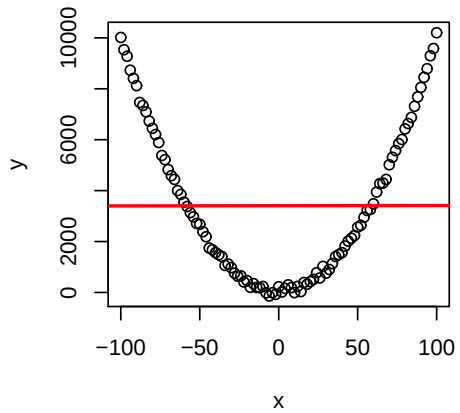


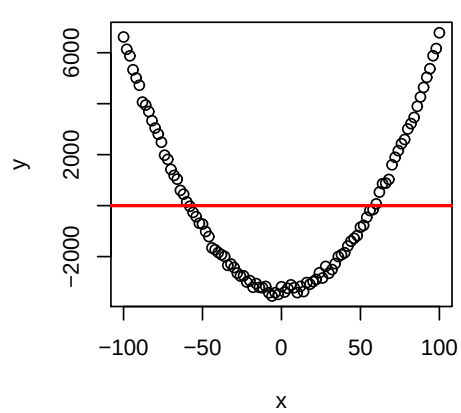
Vérification de la normalité de la variance

- Cela peut se faire de manière graphique :







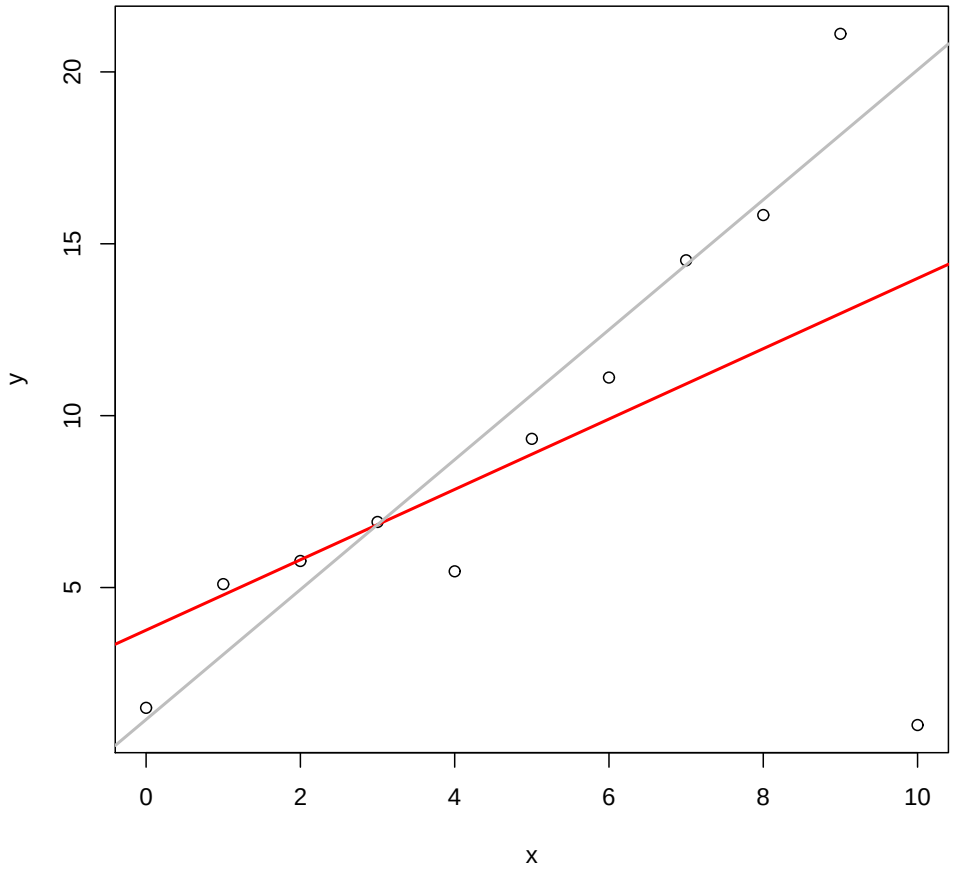


Vérification de la normalité de la variance des résidus avec R

```
#Donnees artificielles
> x <- rnorm(100,5,2)
> y <- rnorm(100,3,1)
> plot(x,y)
#Regression
> reg <- lm(y ~ x) # Fait le calcul pour vous
> abline(reg)
> plot(x,reg$residuals)
> abline(h=0)
```

Vérification des points de levier

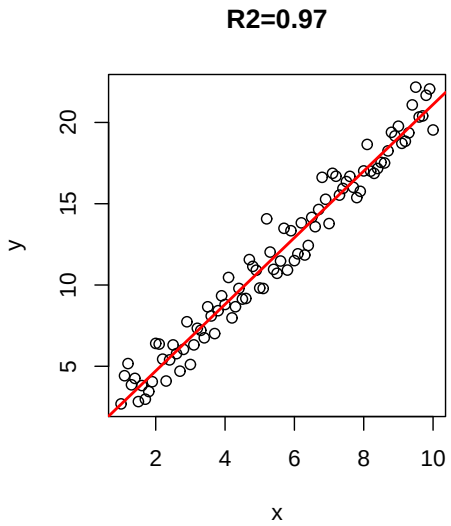
- Le calcul est sensible aux points extrêmes (outliers)
- Important de vérifier : (rouge avec outlier, gris sans outlier)

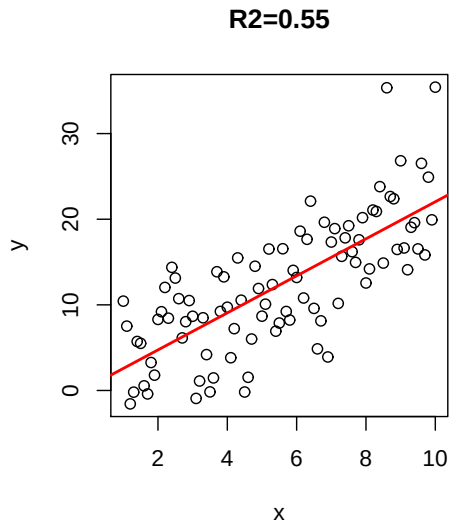


Plan

Vérification de la qualité de la régression

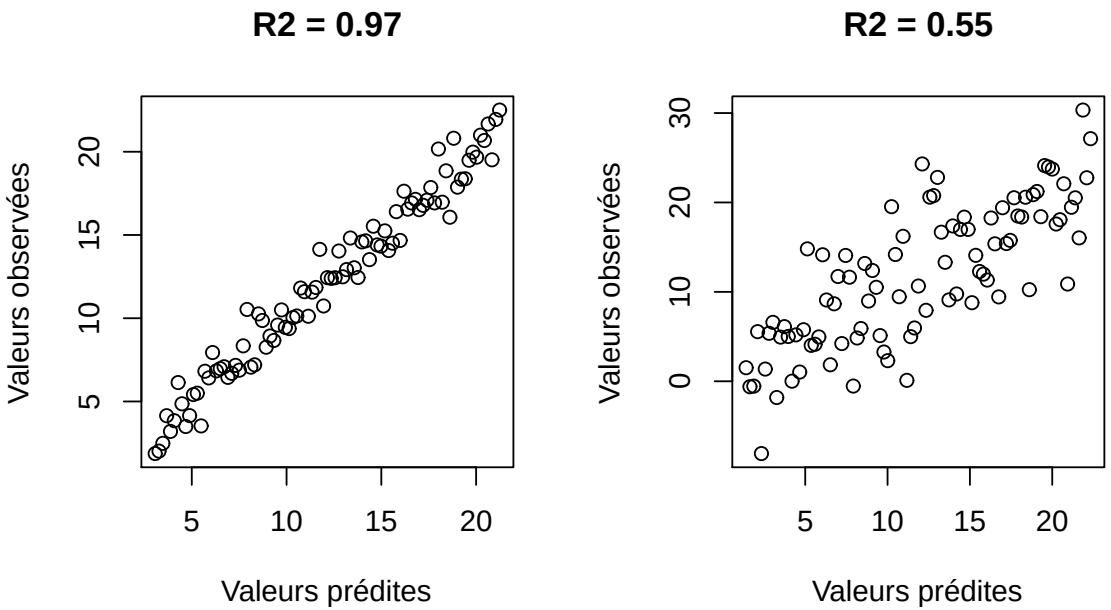
- Vérifier si le modèle explique bien les données : intuition on a une bonne régression si la variance est faible et mauvaise si la variance est importante :





Evaluer la qualité de la régression = R^2

- Pour évaluer la qualité de la régression, une méthode classique est de faire un graphique de la corrélation entre les valeurs prédites $\hat{y}$ en fonction des valeurs effectivement observées y.



Valeurs prédites en fonction des valeurs observées

```
#Donnees artificielles
> x <- rnorm(100,5,2)
> y <- rnorm(100,3,1)
> plot(x,y)
#Regression
> reg <- lm(y ~ x) # Fait le calcul pour vous
#Predire de nouvelles valeurs
> ndata <- data.frame(x)
> py <- predict(reg,newdata=ndata)
> plot(y,py)
#R-square
> cor(y,py)**2
```

- On utilise R^2 qui est le carré de $r(y, \hat{y})$ (coefficient de corrélation de Pearson) pour donner une idée de la qualité de la régression
- $R^2 \in [0, 1] \subset \mathbb{R}$
- Proche de 1, régression parfaite, proche de 0, régression catastrophique

- La somme des carrés totale est la variabilité des données par rapport à la moyenne :

$$SS_{\text{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- La somme des carrés résiduels (cf. infra) est :

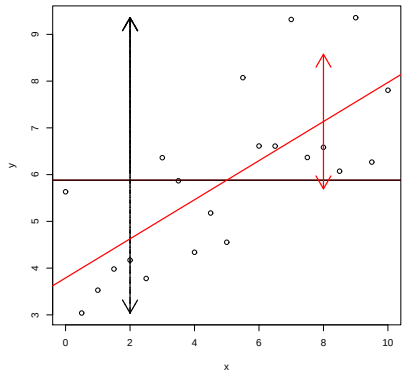
$$SS_{\text{err}} = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

- Coefficient de détermination est fonction du ratio de la somme des erreurs sur la somme totale

$$R^2 = 1 - \frac{SS_{\text{err}}}{SS_{\text{tot}}}$$

R^2 (interprétation graphique)

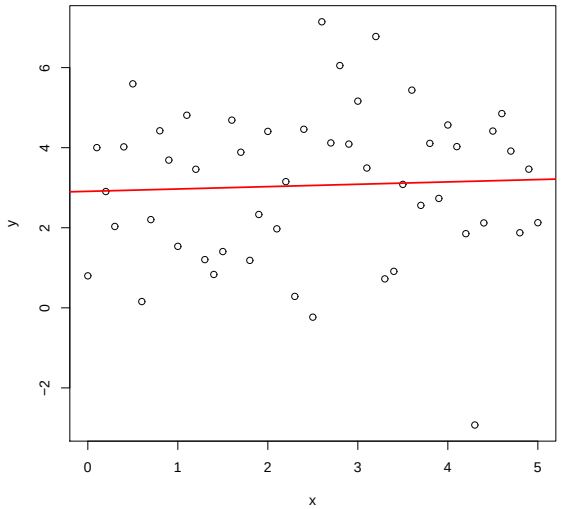
- Lorsqu'on ne connait pas la valeur de x, le choix naturel est de prendre la valeur $\bar{y}$ mais on a une grande incertitude sur la valeur prédite (flèches noires).
- Si on connait la valeur de x alors on prend $\hat{y}$, et on espère une incertitude plus faible sur la valeur prédite (flèches rouges).
- Si il y a une vraie relation entre les variables, alors on espère que $SS_{err} << SS_{tot}$; si la relation est faible ou non existante alors $SS_{err} \approx SS_{tot}$.



Plan

Tests d'hypothèse sur les coefficients (motivation)

- Parfois, on peut avoir une variable prédictrice (x) qui ne permet pas de prédire véritablement la valeur de la variable prédite (y), ce cas de figure est illustré ici :



- C'est le cas ou la droite de régression est horizontale, on a une droite de la forme $y = \alpha + 0x$. Cela signifie que x et y sont indépendantes.

Test sur le coefficient

- On peut tester si le coefficient de pente β est significativement différent de 0.
- H_0 stipule que le coefficient est nul, H_a le contraire
- La statistique de test est le t-test à $n-2$ degrés de liberté:

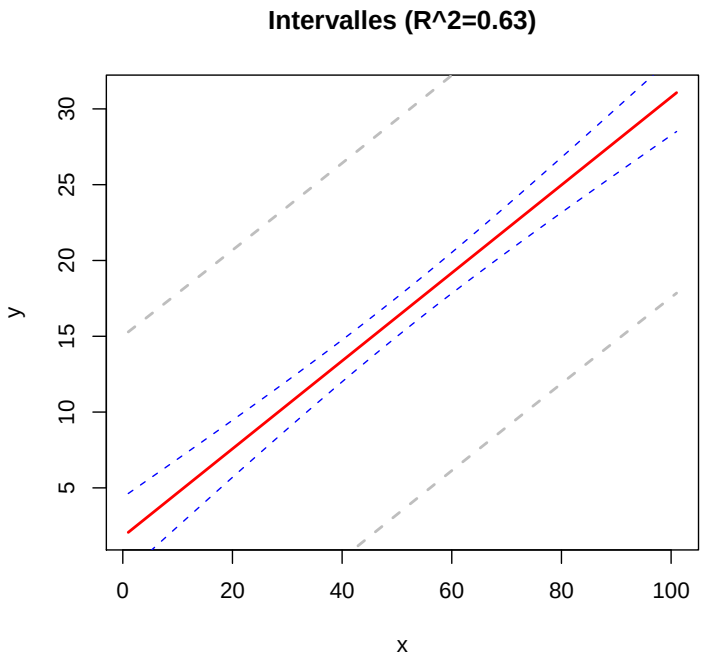
$$t = \frac{\hat{\beta} - 0}{\sigma(\hat{\beta})}$$

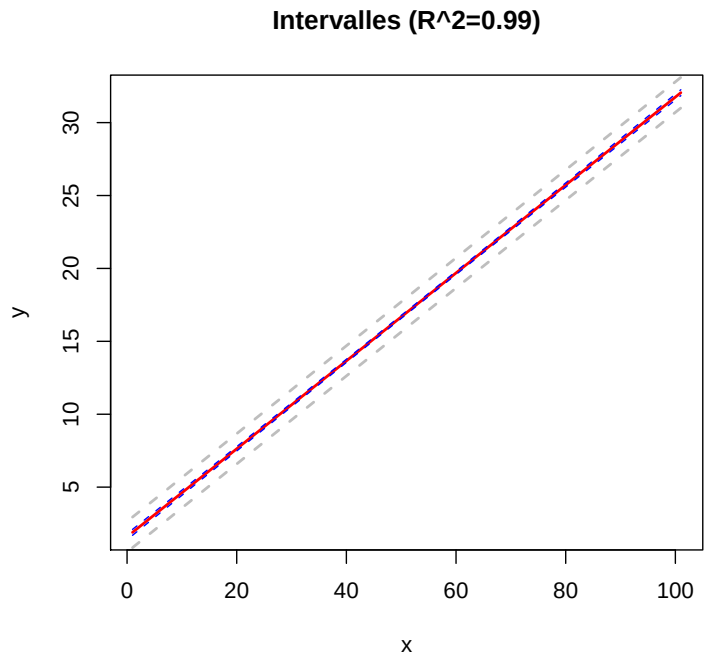
où l'erreur standard $\sigma(\hat{\beta})$ est :

$$\sigma(\hat{\beta}) = \sqrt{\frac{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

Incertitude sur le coefficient

- Intervalles de confiance (bleu): représente l'incertitude sur la zone dans laquelle la droite aurait pu être estimée (95% de chances de résider dans cette zone).
- Intervalles de prédiction (gris): représente la zone d'incertitude dans laquelle les points ont 95% de chances d'apparaître avec la droite estimée.





La régression linéaire avec R

```
> data(english)
> plot(english$lexdec~english$WrittenFrequency)
> my.model <- lm(lexdec~WrittenFrequency,data=english)
> abline(my.model,lwd=2,col="red")
> summary(my.model)
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.0894 -1.3107  0.2657  1.3102  4.0816

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.90925    0.53777   5.410 1.87e-06 ***
x            0.05892    0.18537   0.318    0.752
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.949 on 49 degrees of freedom
Multiple R-squared: 0.002057,Adjusted R-squared: -0.01831
F-statistic: 0.101 on 1 and 49 DF,  p-value: 0.752
```

Plan

Modélisation à plusieurs variables

- Le travail de modélisation est généralement réalisé avec plusieurs variables
- Le but de la modélisation est d'inférer un modèle théorique qui prédit le mieux possible les données et dont la variance inexpliquée est la plus faible possible.
- Ce modèle trouvé, on peut alors l'analyser

Cas d'étude

- Tiré de Baayen : Baayen, R.H., Feldman, L. and Schreuder, R. (2006) Morphological influences on the recognition of monosyllabic monomorphemic words, Journal of Memory and Language, 53, 496-512.
  - Ici : prédire la familiarité des mots à partir de la catégorie et la fréquence des mots à l'écrit

```
> library(languageR)
> data(english)
> head(english)
#Creates a data frame for the class
> exo <- data.frame(Familarity = english$Familiarity,
                    WrittenFrequency=english$WrittenFrequency,
                    WordCategory=english$WordCategory,
                    RT=english$RTnaming)
```

Exemple (suite)

- Prédire la familiarité en fonction de la fréquence des mots, de leur catégorie et de leur temps de lecture :

	Familarity	WrittenFrequency	WordCategory	RT
1	2.37	3.91	N	6.15
2	4.43	4.52	N	6.25
3	5.60	6.51	N	6.14
4	3.87	5.02	N	6.13
5	3.93	4.89	N	6.20
6	3.27	4.77	N	6.17
7	3.73	6.38	N	6.12
8	5.67	7.16	N	6.10
9	3.10	4.89	N	6.12
10	4.43	5.93	N	6.18

- Il s'agit de prédire la variable (Familiarity) à partir de trois variables (WrittenFrequency,RT,WordCategory)

Plan

Fonction de plusieurs variables

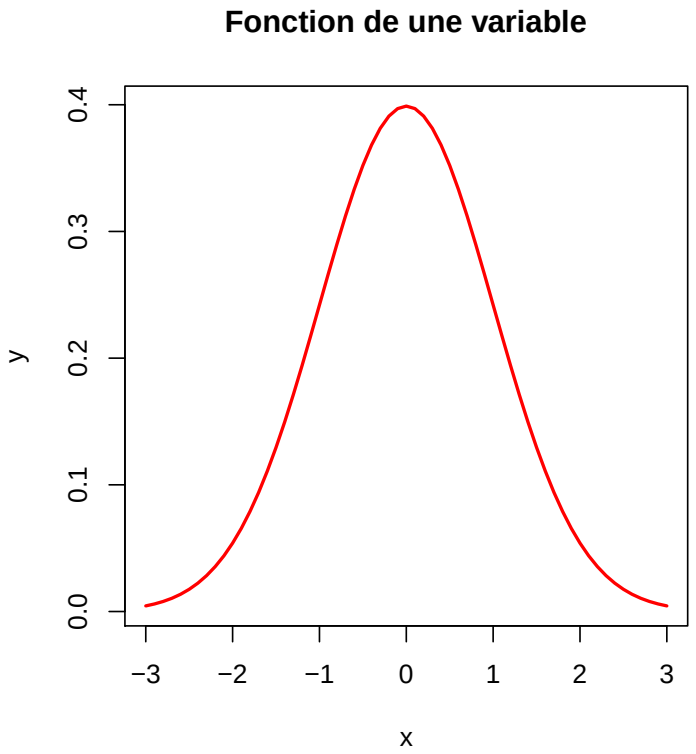
- La généralisation des méthodes qui précèdent aux fonctions de plusieurs variables est assez directe d'un point de vue formel, on fitte désormais des fonctions du type,

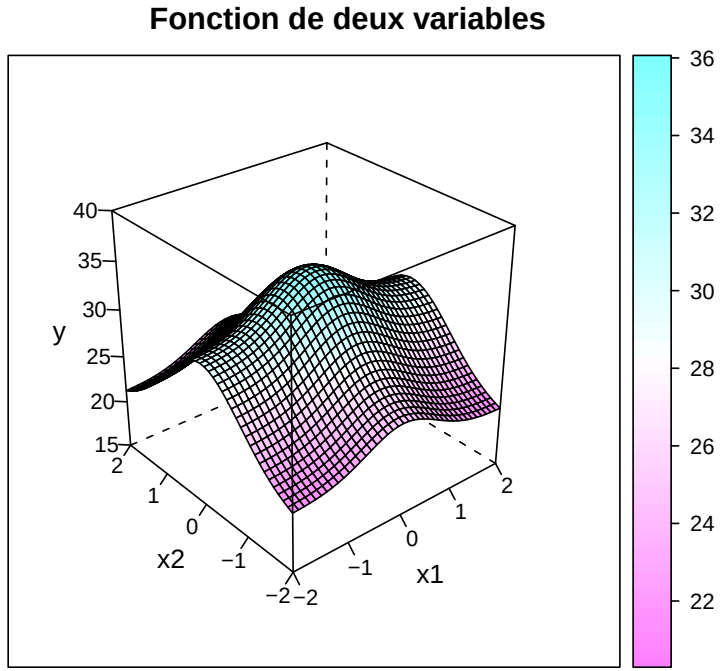
$$y = \alpha + \beta_1 X_1 + \beta_i X_i + \beta_n X_n + \epsilon = \alpha + \beta \mathbf{X} + \epsilon$$

- Par contre, d'un point de vue méthodologique ça se complique un peu...

Rappel : fonction de plusieurs variables

- Une variable : fonction de $\mathbb{R}$ dans $\mathbb{R}$
- Deux variables : fonction de $\mathbb{R}^2$ dans $\mathbb{R}$
- Trois variables :fonction de $\mathbb{R}^3$ dans $\mathbb{R}$ (plus de représentation graphique)





La régression multivariée facile (et incorrecte. . .)

- Imaginons que l'on veuille prédire la `Familiarity` en fonction de `RT` et de `WrittenFrequency`
- On pourrait imaginer décomposer en deux sous problèmes:

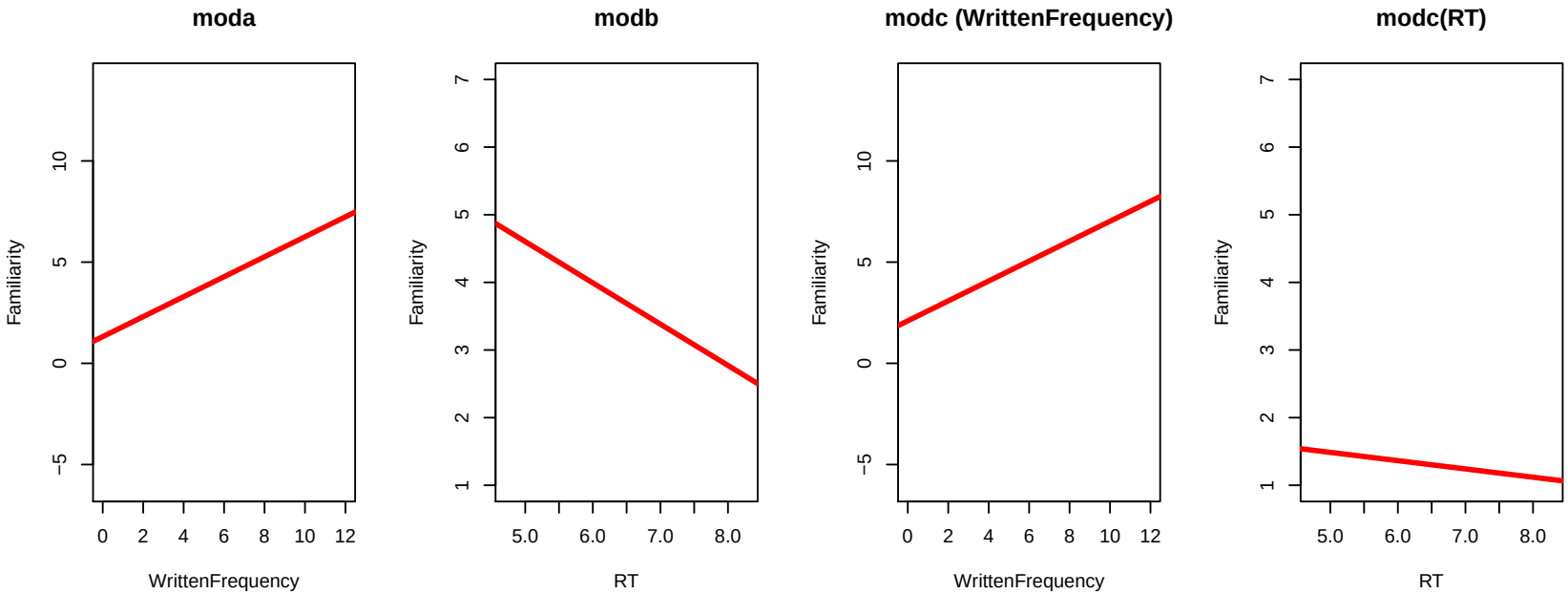
$$\begin{aligned} \text{Familiarity} &= \alpha + \beta\ \text{RT} + \epsilon \\ \text{Familiarity} &= \alpha + \beta\ \text{WrittenFrequency} + \epsilon \end{aligned}$$

- Plutôt qu'un seul :

$$\text{Familiarity} = \alpha + \beta_1\ \text{RT} + \beta_2\ \text{WrittenFrequency} + \epsilon$$

```
> moda <- lm(Familarity ~ WrittenFrequency, data=exo)
> modb <- lm(Familarity ~ RT, data=exo)
> modc <- lm(Familarity ~ WrittenFrequency+RT, data=exo)
> summary(moda)
> summary(modb)
> summary(modc)
```

Illustration



Observations

- Les coefficients diffèrent !
- Il se trouve que les deux variables `RT` et `WrittenFrequency` ne sont pas parfaitement indépendantes

```
> cor(exo$RT,exo$WrittenFrequency)
> pairs(exo)
> summary(modc,corr=T)
```

- Les variables s'influencent l'une l'autre ! (corrélation faible mais réelle)

La plaie : corrélations

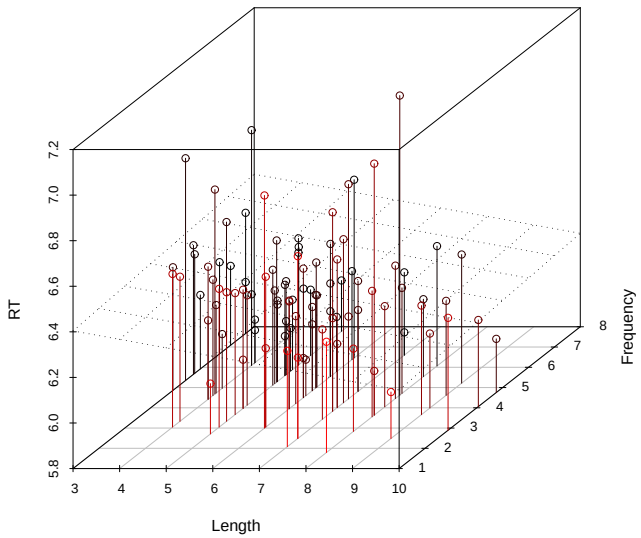
Dans le cas multivarié, l'interprétation des coefficients est plus difficile dans la mesure où ils sont corrélés.

Travailler avec des fonctions à plusieurs variables

- En réalité la régression à n variables revient à calculer les coordonnées d'un hyperplan à n dimensions
- Supposons que l'on veuille prédire le RT en fonction de la fréquence (x_1) et de la longueur (x_2) des mots, le modèle est :

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

- Illustration :



Modélisation à plusieurs variables

Buts classiques

- Produire un modèle qui prédit le "mieux possible" les données
- Identifier un sous-ensemble de variables qui permet de prédire les données (toutes les variables ne sont pas toujours utiles)
- Identifier les variables qui ont un effet sur la prédiction

Méthode

1. S'assurer que le modèle prédit bien les données (**qualité du fit**)
   - Variance pas trop élevée
   - Généralise correctement (pas d'overfitting)
2. Vérifier les corrélations entre les variables
   - Mise à l'échelle des variables
3. Compacter le modèle (Comparaison de modèles)
4. Interpréter les coefficients

Plan

Tests de sanité publique

J'en illustre deux :

- Vérifier que la distribution des résidus ϵ est bien normale
  - Peut se faire graphiquement (test de Shapiro-Wilk)

```
#Plot des residus
> plot(modc$residuals)
> abline(h=0,lwd=3,col="red")
#Alternativement
> hist(modc$residuals)
```

- Vérifier qu'il n'y a pas d'overfitting
- ...

But de ces tests

Vérifier que le modèle généralise correctement les données : le premier verifie que votre modèle est bien adapté pour modéliser vos données, le second que votre modèle ne reste pas collé dans les données

Test de surentrainement (overfitting)

- Par validation croisée (cross-validation)
  - On divise le corpus en k (ex. k = 50) morceaux et on répète k fois le processus d'entrainement sur $k-1$ morceaux (test sur le $k$ième morceau) en prenant à chaque fois un test différent.
- Par bootstrapping :
  - Pour un corpus C de k observations on génère un nouveau corpus C' de $k' = k$ observations telle que chaque observation dans C' soit tirée au sort dans C (avec remplacement, ie. on peut tirer plusieurs fois la même observation dans C)

```
>library(Design)
> mod <- ols(Familiarity ~ WrittenFrequency+RT,
                          data=exo,x=T,y=T)
#Par bootstrapping (1000 runs)
>validate(mod, method="boot",B=1000)
#Par validation croisee (100 runs)
>validate(mod, method="crossvalidation",B=1000)
```

Diagnostics de surentrainement

Typiquement on obtient qqch du genre:

```
> validate(mod,method="boot",B=100)
          index.orig  training        test      optimism index.corrected    n
R-square   0.6264435 0.6265450 0.626202632  0.0003424013     0.626101148 1000
MSE        0.4933415 0.4928831 0.493659638 -0.0007765382     0.494118006 1000
Intercept  0.0000000 0.0000000 0.001007654 -0.0010076542     0.001007654 1000
Slope      1.0000000 1.0000000 0.999703546  0.0002964543     0.999703546 1000
```

- On vérifie la ligne R^2:
  - Le R^2 d'entrainement est le R^2 moyen calculé sur les corpus bootstrappés $C'_{1...B}$
  - Le R^2 de test est le R^2 moyen qui est calculé sur les données originales C avec chaque modèle appris sur un corpus C'_i
  - Optimism est la différence entre les deux
  - index.corrected est la valeur du R^2 original (sur toutes les données) moins la valeur d'optimisme

Plan

Qualité de la régression (goodness of fit)

Avant d'analyser un modèle, il faut vérifier qu'il soit good enough, c'est-à-dire qu'il est capable de prédire correctement les données

- Graphiques de corrélation (observés / prédits)
- R^2 ajusté

Graphiques de corrélation

- On peut faire des graphiques de corrélation des valeurs prédites en fonction des valeurs observées :

```
> mod <- lm(Familarity ~ WrittenFrequency+RT,
                          data=exo)
> plot(exo$Familarity,mod$fitted.values)
```

R^2 ajusté

- Le R^2 tel que défini jusqu'ici tend à augmenter lorsque le nombre de variables augmente.
- R^2 ajusté est un R^2 adapté (généralisé) aux cas à plusieurs variables prédictrices (en tenant compte du nombre de variables du modèle)
- Pour m observations et un modèle à n variables $X_1 \dots X_n$, on définit la variance de la variable observée Y:

$$SS_{tot} = \frac{\sum_{i=1}^{m} (\bar{y} - y_i)^2}{m - 1}$$

- Et la variance des résidus :

$$SS_{err} = \frac{\sum_{i=1}^{m} (\hat{y} - y_i)^2}{m - n}$$

- Et le R^2 ajusté:

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}}$$

Plan

Vérifier les corrélations

- On peut calculer toutes les corrélations entre les variables d'une dataframe (numérique) comme suit :

```
# Enlever la colonne de categorie
# dans la table exo
> df <- data.frame(Fam=exo$Familarity,
                   RT=exo$RT,
                   WF=exo$WrittenFrequency)
> cor(df)
```

- On peut visualiser les distributions deux par deux comme suit :

```
pairs(df)
```

- Pour les variables nominales, on peut utiliser un test de χ^2 pour tester l'indépendance

Collinéarité et Variance Inflation Factors (VIFs)

- Il se peut qu'une variable X_i soit corrélée à une combinaison linéaire de variables $X_j + \ldots + X_m$
- Calcul des VIFs (Variance Inflation Factors)
- Soit un modèle de la forme : $Y = \alpha + \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon$
- Pour chaque prédicteur X_i de coefficient β_i:
  1. Calculer la régression en ommettant $\beta_i X_i$:

$$Y = \alpha + \beta_1 X_1 + \ldots \beta_{i-1} X_{i-1} + \beta_{i+1} X_{i+1} + \ldots + \beta_n X_n + \epsilon$$

  2. Calculer R_i^2 le coefficient de détermination de cette régression:

$$\text{VIF}(\beta_i) = \frac{1}{1 - R_i^2}$$

si le R^2 sans la variable est grand $\Rightarrow$ VIF important $\Rightarrow$ indicateur de multicollinéarité

- Les VIFs supérieur à 5 (parfois à 10) sont considérés comme des indicateurs sérieux de multicollinéarité

Calcul des Variance Inflation Factors avec R

Se calcule avec une fonction de la librairie `Design`:

```
> library(Design)
> mod <- ols(Familarity ~ WrittenFrequency+RT,
             data=exo)
> vif(mod)
WrittenFrequency                       RT
        1.009301                 1.009301
```

Plan

Comparer des modèles enchâssés

- Question : est-ce que toutes le variables contribuent à la prédiction ?
- Méthode : comparer différents modèles enchassés (modèle plus petit dont les variables sont strictement incluses dans modèle plus gros) ; est-ce que les variables supplémentaires apportent qqch ?

Rasoir d'Occam

- Guillaume d'Occam (1285-1349) :

  Numquam ponenda est pluralitas sine necessitate.

  ≈ Il n'est pas nécessaire d'avoir des pluralités sans besoin.
- On cherche les modèles les plus petits et les plus explicatifs possibles, pas besoin de faire des modèles avec des variables qui n'apportent rien.

Méthode

- Rappels :
  - Plus faible la variance des erreurs, meilleur le modèle

    $SS_{err} = \sum (y_i - \hat{y})^2$ (= variance des erreurs à normalisation près)

- Pour un corpus de m observations, on pose $M_{\subset}$ à $n_{\subset}$ variables et M_{grand} à n_{grand} variables, on compare leurs variances comme suit :

$$F = \frac{\left(\frac{SS_{\subset} - SS_{grand}}{n_{grand} - n_{\subset}}\right)}{\left(\frac{SS_{grand}}{m - n_{grand}}\right)}$$

- F est la statistique du rapport de variance et suit la loi F (Loi de Fisher Snedecor)

Intuition

- $SS_{\subset} > SS_{grand}$
- Plus grande la différence entre la variance de M_{grand} avec $M_{\subset}$, plus la statistique F est grande.

Test d'hypothèse

- On peut comparer deux modèles en comparant la variance de leurs erreurs (résidus).
- La statistique F suit une loi de probabilité
- Test d'hypothèse (F-Test)
  - H_0 : La différence des variances est nulle
  - H_A : La différence des variances est significativement différente de 0.
- Donne un moyen de décider si les variables supplémentaires de M_{grand} apportent une information significative.

Attention

Ce test ne fait pas sens si la distribution des résidus de chacun des modèles n'est pas normale.

- La fonction anova fait tous les calculs pour vous:

```
> modLarge <- lm(Familarity ~ RT+WrittenFrequency,data=exo
> modSmall <- lm(Familarity ~ WrittenFrequency,data=exo)
> anova(modSmall,modLarge)
Analysis of Variance Table

Model 1: Familarity ~ WrittenFrequency
Model 2: Familarity ~ RT + WrittenFrequency
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1   4566 2255.7
2   4565 2253.6  1    2.1605 4.3765 0.03649 *
```

- où RSS dénote la somme des carrés résiduels (noté ici SS_{err})

Plan

Interprétation des coefficients

- Un coefficient positif favorise l'augmentation de la variable prédite
  - Une augmentation de 1 pour la variable prédictrice X_i fait augmenter Y de β_i
- Un coefficient négatif favorise la diminution de la variable prédite
  - Une augmentation de 1 pour la variable prédictrice X_i fait diminuer Y de β_i
- Notons que les coefficients sont fonction de l'échelle des variables respectives, pour comparer des coefficients, il faut des variables à la même échelle

```
#Effet de l'echelle des variables sur les coefficients
> exo$WrittenFrequency2 <- exo$WrittenFrequency*100
> head(exo)
> mod1 <- lm(Familarity ~ WrittenFrequency+RT, data=exo)
> mod2 <- lm(Familarity ~ WrittenFrequency2+RT, data=exo)
> summary(mod1)
> summary(mod2)
```

Mise à l'échelle des variables

- Pour comparer différents coefficients entre eux, il faut pouvoir mettre les différentes variables à la même échelle
- Méthode standard (pour les variables normalement distribuées) :

$$z = \frac{x - \bar{x}}{\sigma}$$

- Avec R cela peut se faire comme suit:

```
#Check normality
> hist(exo$WrittenFrequency) # ok.
> hist(exo$RT) #Hum ... hum...
> hist(exo$Familarity) # ok.
#scale
> exo$WFscaled <- scale(exo$WrittenFrequency,center=T,scale=T)
> exo$RTscaled <- scale(exo$RT,center=T,scale=T)
> exo$Famscaled <- scale(exo$Familarity,center=T,scale=T)
# Build model
> lm(Famscaled ~ RTscaled+WFscaled,data=exo)
```

Mise à l'échelle de variables

- Exercice: à chaque fois, comparer les coefficients (modèle non échelonné):
  - Standardiser la variable prédite
  - Standardiser les variables prédictrices

Mise à l'échelle de variables

- Permet de comparer les coefficients
- Interprétation des valeurs des coefficients un peu plus difficile
- Ne change pas la significativité des coefficients
- Autres transformations possibles (non détaillé ici) : $\log(x), x^2 \ldots$

Plan

Codage des variables nominales

- Les problèmes de langage naturel font intervenir habituellement des variables nominales plutôt que numériques
- Il faut alors pouvoir coder les variables nominales sous formes numérique
- On appelle facteur une variable nominale codée sous forme numérique, l'ensemble des valeurs possibles d'un facteur est appelé niveau.

```
#Forcer le codage d'une variable comme facteur
> exo$WordCategory <- factor(exo$WordCategory)
# Observer les niveaux
> exo$WordCategory
> levels(exo$WordCategory)
# Observer le codage
> contrasts(exo$WordCategory)
```

Codage d'une variable binaire

- Dans le cas d'une variable binaire, on peut coder les valeurs de X par la variable $\phi(X)$ à valeurs binaires: $\{0, 1\}$
- Exemple :

$$\phi(X) = \begin{cases} 1 & \text{si } X = V \\ 0 & \text{si } X = N \end{cases}$$

- Dans ce cas le coefficient associé à $\phi(X)$ indique l'effet que la valeur positive (ici V) a sur la variable prédite Y.

Codage d'une variable n-aire

- Soit une variable nominale à k valeurs différentes
- Diviser la variable en $k-1$ variables binaires (à valeurs dans $\{1, 0\}$)
- Exemple : $X = \{bleu, blanc, rouge, jaune\}$:

Niveau	$\phi(X_1)$	$\phi(X_2)$	$\phi(X_3)$
bleu	0	0	0
blanc	1	0	0
rouge	0	1	0
jaune	0	0	1

- Chacune des 3 variables $\phi(X_i)$ encode l'effet d'une des valeurs de X. Le coefficient qui y sera associé correspond à l'effet de cette valeur sur Y.

Dummy coding

Cette manière de coder s'appelle le "dummy coding" : méthode standard, utilisée par défaut par R. Il existe un grand nombre de manières de coder les variables nominales (codage par contrastes notamment)

R contient des fonctions préféfinies

```
> contrasts(exo$WordCategory)
 V
N 0
V 1
> mod <- lm(Familarity ~ RT+WrittenFrequency+WordCategory,data=exo)
> summary(mod)
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.94659    0.36778   5.293 1.26e-07 ***
RT               -0.11313    0.05753  -1.966   0.0493 *
WrittenFrequency  0.49116    0.00557  88.184  < 2e-16 ***
WordCategoryV     0.26924    0.02124  12.677  < 2e-16 ***
#changer le niveau de reference (qui etait N)
> exo$WordCategory  <- relevel (exo$WordCategory,ref="V")
> contrasts(exo$WordCategory)
  N
V 0
N 1
> mod2 <- lm(Familarity ~ RT+WrittenFrequency+WordCategory,data=exo)
> summary(mod2)
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.21583    0.36770   6.026 1.81e-09 ***
RT               -0.11313    0.05753  -1.966   0.0493 *
WrittenFrequency  0.49116    0.00557  88.184  < 2e-16 ***
WordCategoryN    -0.26924    0.02124 -12.677  < 2e-16 ***
```

Modèles sur corpus vs Modèles expérimentaux

- Dans le cas de modèles construits sur corpus, lorsqu'une vraie corrélation persiste, on ne peut contrôler les variables (on ne manipule pas le corpus)
- Dans le cas de modèles expérimentaux, on peut contrôler certaines (idéalement toutes) variables de manière à réduire les corrélations, ce qui permet de mieux contrôler l'effet d'une variable prédictrice donnée sur la variable prédite.