Régressions en R

Alison PATOU

Patou.alison@gmail.com

Programme

- Introduction
- Régression linéaire simple
- Régression linéaire multiple

A savoir

- Tout est sur mon GitHub : https://github.com/apatou
- (L'essentiel à retenir du cours, les dataset, exercices, …)

- Merci de m'envoyer à chaque fin de séance vos TPs : patou.alison@gmail.com

- 1 examen à la dernière séance

1

Introduction

Problématique

Existe-t-il une relation entre deux variables ?

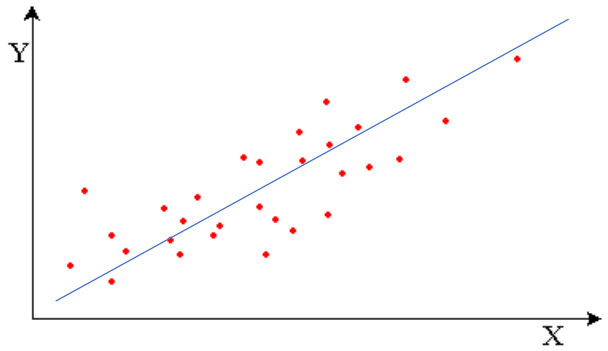
Est-ce qu'il y a une relation entre la taille des enfants à la naissance et leurs poids?
Est-ce que le revenu des parents à une influence sur la réussite scolaire des enfants ?
Existe-t-il un lien entre le prix de vente d'une maison en Californie et la présence d'un porche?

Régression vs Corrélation

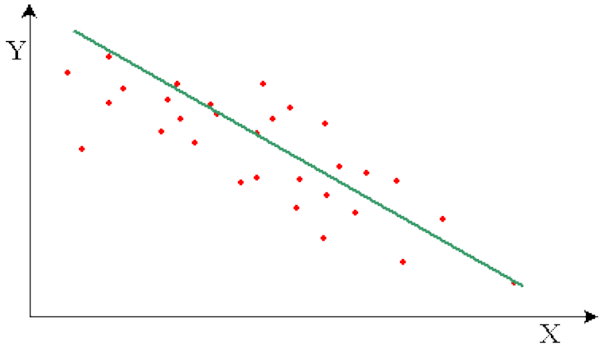
La corrélation statistique est une technique statistique qui s'utilise pour voir si deux variables sont liées.

La corrélation est utilisée pour voir deux choses :

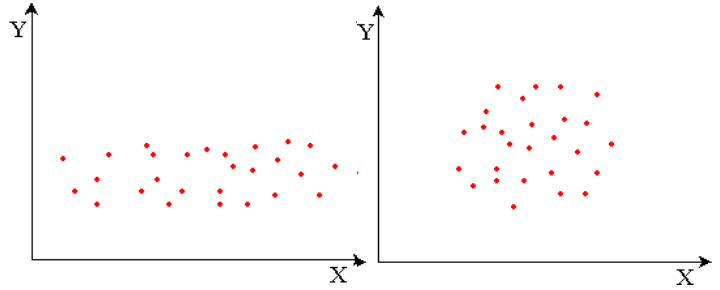
1. si la relation est positive ou négative



Corrélation positive



Corrélation négative



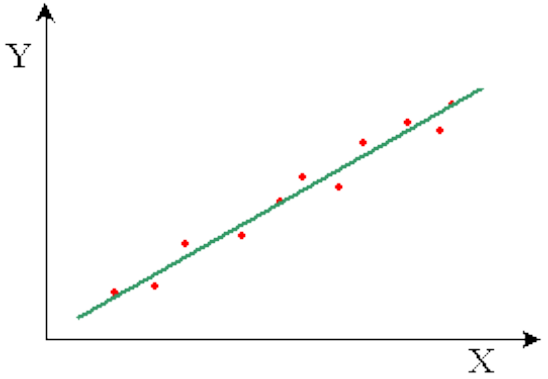
Absence de corrélation

Régression vs Corrélation

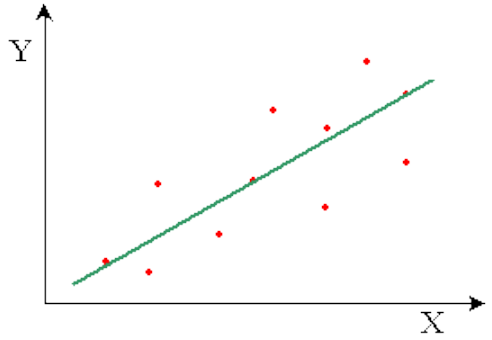
La corrélation statistique est une technique statistique qui s'utilise pour voir si deux variables sont liées.

La corrélation est utilisée pour voir deux choses :

2. la force de la relation (mesuré grâce au coefficient de corrélation r, compris entre -1 et 1)



Corrélation forte



Corrélation faible

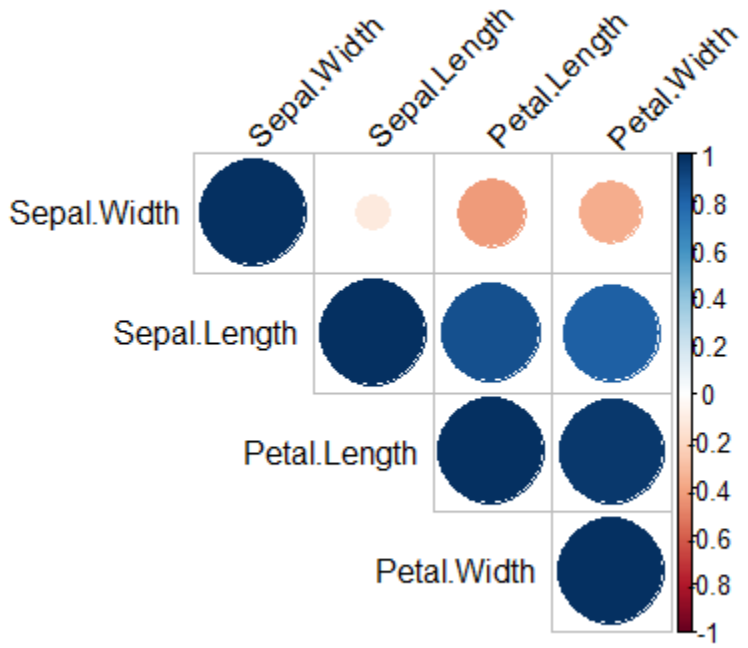
Régression vs Corrélation

Sous R

```
# Matrice de corrélation


cor(iris[,1:4])
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

```
library(corrplot)
corrplot(cor(iris[,1:4]), type="upper", order="hclust",
tl.col="black", tl.srt=45)
```



Régression vs Corrélation

Problème: on ne peut pas considérer qu'un coefficient de 0.85 par exemple est toujours suffisant, ou bien systématiquement nécessaire pour que la liaison entre deux variables soit considérée comme significative.

Lors d'étude DataMining, avoir un coefficient de corrélation proche de 1 nous permet de dire que la probabilité que la relation entre deux variables soit significative est grande.

Globalement, avec un tableau de corrélation, on peut avoir rapidement et visuellement une première idée des pistes à creuser et des réductions de variables possibles si on souhaite aller plus loin dans le Machine learning par exemple.

Régression linéaire simple

Définition

<u>La régression linéaire va nous permettre :</u>

D'être en capacité d'estimer une variable cible numérique et **continue**

<u>Formulation :</u>

$$E(Y) = f(X) = \beta_0 + \beta_1 X \quad \text{ou} \quad Y = \beta_0 + \beta_1 X + \epsilon$$

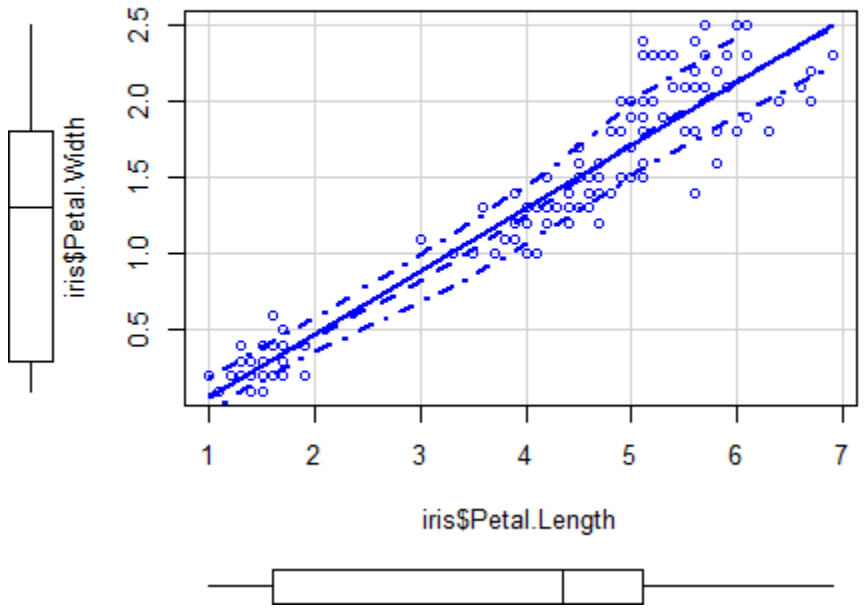
Y : variable qu'on souhaite estimée (de sortie)
X : variable explicative
B0 et β1 : sont les coefficients de la régressions
ε : le résidu (la formule est vrai en moyenne seulement)

Définition

$$E(Y) = f(X) = \beta_0 + \beta_1 X \quad \text{ou} \quad Y = \beta_0 + \beta_1 X + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : variable explicative
B0 et β1 : sont les coefficients de la régressions
ε : le résidu (la formule est vrai en moyenne seulement)

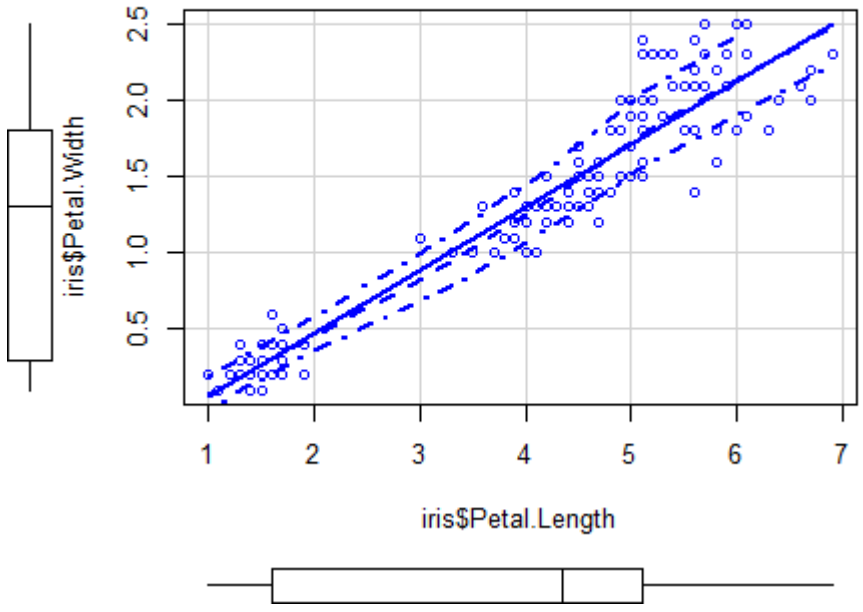
Quelle est la relation entre ma longueur de petal et sa largeur ?
Comment prédire ma largeur de pétal en fonction de ma longueur?



Exemple

$$E(Y) = f(X) = \beta_0 + \beta_1 X \quad \text{ou} \quad Y = \beta_0 + \beta_1 X + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : variable explicative
β0 et β1 : sont les coefficients de la régressions
ε : le résidu (la formule est vrai en moyenne seulement)



Dans notre cas :

variable à expliquer : Y
variable explicative : X
pente de ma droite de régression : β1
constante de ma droite (quand X=0, Y= β0) : β0

Conditions

$$E(Y) = f(X) = \beta_0 + \beta_1 X \quad \text{ou} \quad Y = \beta_0 + \beta_1 X + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : variable explicative
β0 et β1 : sont les coefficients de la régressions
ε : le résidu (la formule est vrai en moyenne seulement)

Pour avoir une régression linéaire qui ait des résultats exploitables, il y a des hypothèses/conditions à avoir :

- Les points doivent le plus suivre une droite
- Les variances des populations sont toutes égales (homoscédasticité) : globalement vos points sont peu éloignés.
- les points doivent avoir suivre une distribution normale

Estimation

$$E(Y) = f(X) = \beta_0 + \beta_1 X \quad \text{ou} \quad Y = \beta_0 + \beta_1 X + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : variable explicative
β0 et β1 : sont les coefficients de la régressions
ε : le résidu (la formule est vrai en moyenne seulement)

Une manière d'estimer les paramètres d'un modèle est de maximiser la fonction de vraisemblance correspondante. Dans le cas de la régression linéaire simple, cette fonction est :

$$\log(L(Y = y|\Theta)) = \log\left(\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right)\right)$$

$$l(Y = y|\Theta) = -n(\log(\sigma) + \log(2\pi)) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Ces deux fonctions ne dépendent finalement que de β0 et β1, qu'il faut finalement trouver pour maximiser ces équations.

Validation

$$E(Y) = f(X) = \beta_0 + \beta_1 X \quad \text{ou} \quad Y = \beta_0 + \beta_1 X + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : variable explicative
β0 et β1 : sont les coefficients de la régressions
ε : le résidu (la formule est vrai en moyenne seulement)

La méthode des moindres carrées va permettre de mesurer que nos points se rapprochent « bien » de notre droite de régression.

Théoriquement cela signifie que :

$$Min\left(\sum_{i=0}^{n}(y_i - \hat{y}_i)^2\right)$$

$$Min\left(\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2\right)$$

Dans cette équation, Yi représente chaque individu (ou point) de votre dataset alors que Ŷi représente la prédiction de votre modèle.
Après plusieurs itérations, votre algorithme est capable de trouver le nombre minimum dans cette formule et donc avoir la meilleure ligne possible qui décrit votre dataset.

En R

```
iris.lm <- lm(Petal.Length~Petal.Width, data=iris)
```

```
> iris.lm

Call:
lm(formula = Petal.Length ~ Petal.Width, data = iris)

Coefficients:
(Intercept)  Petal.Width
      1.084        2.230
```

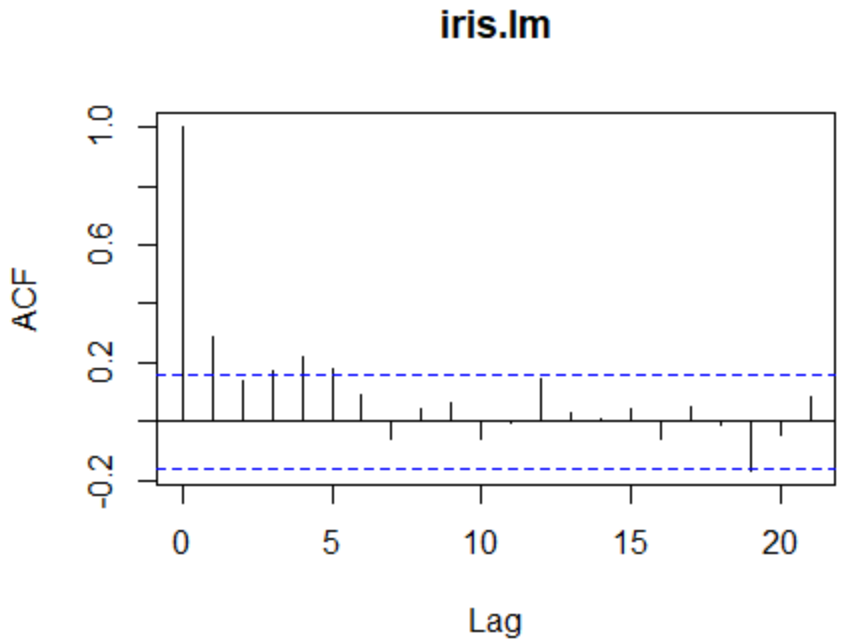
Ici β1 est à 2.230, ce qui signifie que quand la largeur monte d'une unité, la longueur monte de 2.230 unités

En R

Evaluation de l'hypothèse d'indépendances des résidus

On parle d'auto-corrélation des résidus lorsque, par exemple, le résidu d'un point quelconque est liée à celui du point suivant dans le tableau de données

```
> acf(residuals(prest.lm1), main="prest.lm1")
```



Les **pointillées horizontaux**, sont les **intervalles de confiance du coefficient de corrélation égal à 0**.
Les **traits verticaux** représentent **les coefficients de corrélation entre les résidus de chaque point et ceux des points de la ligne suivante (lag=1)**

Ici on a une autocorrélation significative sur les lag 1,2,4,5,6

En R

Evaluation de l'hypothèse d'indépendances des résidus

Un autre moyen de le vérifier est avec le test de DurbinWatson, en vérifiant que la pvalue est >0.05

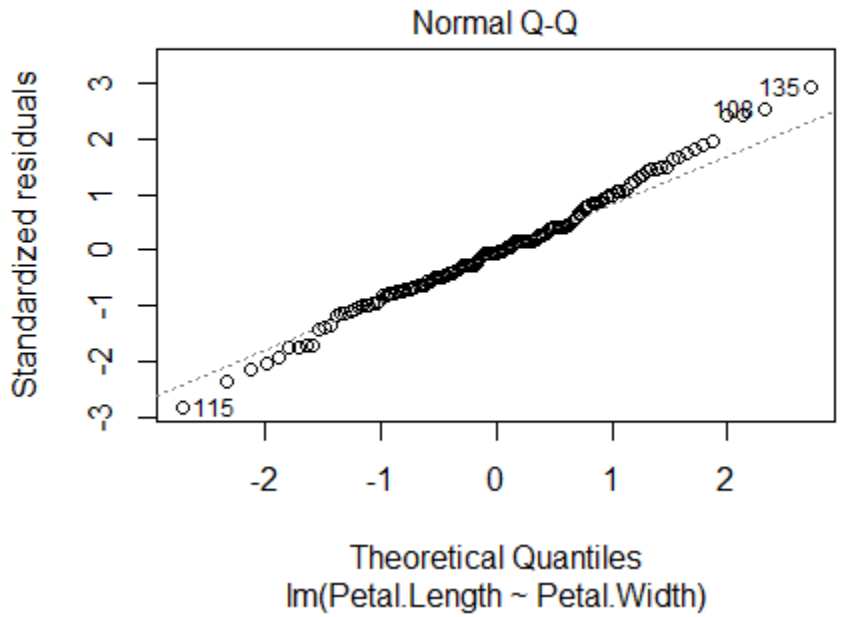
```
> durbinWatsonTest(iris.lm)
 lag Autocorrelation D-W Statistic p-value
   1       0.2849762      1.429552       0
 Alternative hypothesis: rho != 0
```

Ici, le test nous indique qu'il existe une **auto-corrélation significative** entre les résidus d'une ligne du tableau de données et ceux de la ligne suivante

En R

Evaluation de la normalité des résidus

```
> plot(iris.lm,2)
```



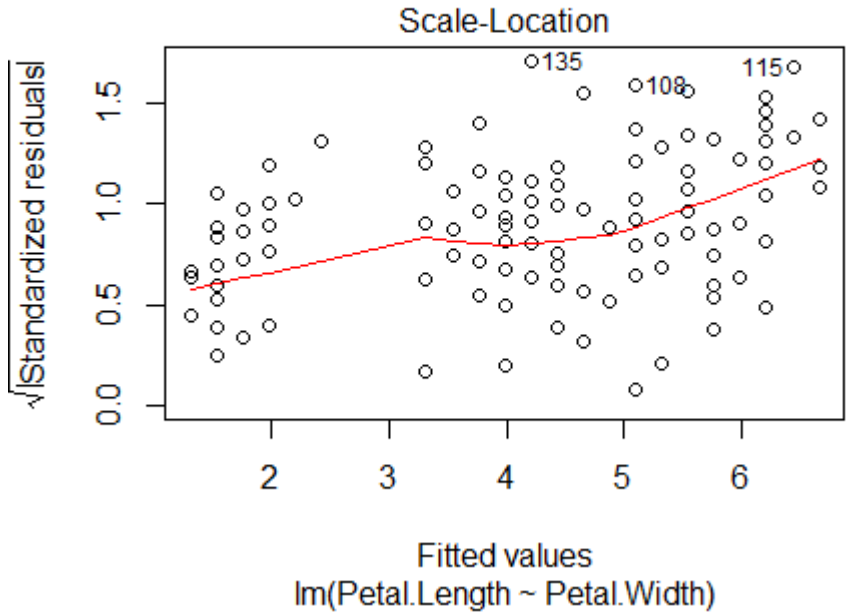

```
> shapiro.test(residuals(iris.lm))

        Shapiro-Wilk normality test

data:  residuals(iris.lm)
W = 0.99011, p-value = 0.3753
```

La normalité des résidus est acceptée (car 0,37>0.05)

En R

Evaluation de l'hypothèse de l'homogéneité des résidus

```
> plot(iris.lm,3)
```




```
> ncvTest(iris.lm)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 28.30794, Df = 1, p = 1.0347e-07
```

L'homogénéité est rejetée car pvalue >0.05

On le voit également sur le graphique : la pente reste ascendante.

Le rejet d'une ou plusieurs des hypothèses n'impacte pas l'estimation des paramètres, mais seulement l'estimation de leur erreur standard.

En R

Maintenant revenons aux résultats :

```
> summary(iris.lm)

Call:
lm(formula = Petal.Length ~ Petal.Width, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.33542 -0.30347 -0.02955  0.25776  1.39453

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.08356    0.07297   14.85   <2e-16 ***
Petal.Width  2.22994    0.05140   43.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4782 on 148 degrees of freedom
Multiple R-squared:  0.9271,    Adjusted R-squared:  0.9266
F-statistic:  1882 on 1 and 148 DF,  p-value: < 2.2e-16
```

La médiane est résidus est autour de 0 : les résidus suivent une loi normale.

En R

La prédiction

On souhaite obtenir la réponse prédite par le modèle de régression, pour une valeur spécifique de la variable prédictive, qui n'a pas été observée. Dans notre exemple imaginons que nous souhaiterions prédire la longueur de la pétale pour une largeur de 2.8

```
> my_df <- data.frame(Petal.Width=c(2.8))
> predict(iris.lm, newdata=my_df)
       1
7.327391
```

1. Je créé un nouveau dataset avec ma nouvelle valeur à prédire
2. Ce nouveau dataset est ensuite passé dans l'argument « newdata » de la fonction predit

Ma longueur de sépal serait 7.32.

Régression linéaire multiple

Définition

Dans la plupart des cas, vous n'aurez pas qu'un seul facteur qui va vous permettre de prédire votre variable dépendante. Par exemple, vous pouvez prédire le salaire de quelqu'un avec son nombre d'années d'expérience mais surement aussi le type de diplôme, le secteur dans lequel la personne travaille, le sexe, le pays etc.

C'est la seule différence entre la régression linéaire simple et multiple. Vous ajoutez des variables indépendantes dans l'équation.

Formulation :

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Définition

Formulation :

$$Y = X\beta + \epsilon$$

Soit en écriture Matricielle

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \dots & x_{np} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \text{et} \quad \epsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Conditions

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Pour avoir une régression linéaire multiple qui ait des résultats exploitables, il y a des hypothèses/conditions à avoir :

- Les points doivent le plus suivre une droite
- Les variances des populations sont toutes égales (homoscédasticité) : globalement vos points sont peu éloignés.
- les points doivent avoir suivre une distribution normale
- la non-colinéarité des variables explicatives

Estimation

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Il s'agit maintenant de sélectionner parmi les p variables explicatives, les q ≤ p variables qui donnent le "meilleur" modèle pour prédire y.

Il faut donc :

- un critère de qualité d'un modèle afin de comparer deux modèles n'ayant pas nécessairement le même nombre de variables explicatives.
- une procédure qui permet de choisir parmi tous les modèles, le meilleur au sens de ce critère. On parle de procédure de choix de modèle.

Estimation

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Les estimateurs à minimiser sont finalement les mêmes que ceux de la régression linéaire.

$$\log(L(Y = y|\Theta)) = \log\left(\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right)\right)$$

$$l(Y = y|\Theta) = -n(\log(\sigma) + \log(2\pi)) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

Ces deux fonctions ne dépendent finalement que de β0 et β1, qu'il faut finalement trouver pour maximiser ces équations.

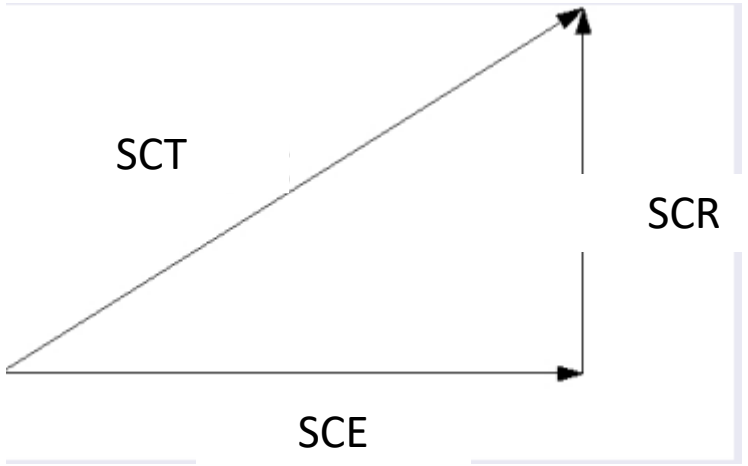
Estimation

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Coefficient de détermination (rappel) :

SCT = SCE + SCR



Estimation

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

R^2 qu'on appelle en anglais R-squared, est une statistique qui quantifie le pouvoir explicatif du modèle par rapport à la variable cible.

$$R^2 = 1 - \frac{SSR}{SST}$$

Estimation

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

$$R^2 = 1 - \frac{SSR}{SST}$$

- Compris entre 0 et 1
- R^2 est un indicateur de performance qui permet uniquement de comparer deux modèles qui ont le même nombre de variables explicatives.
- Si le modèle est peu pertinent, la somme des carrés résiduels SSR sera proche de la somme des carrées totaux SST et R^2sera plus proche de 0
- Si le modèle permet d'expliquer fidèlement la variable cible, alors SSR sera plus proche de 0 et R^2 sera plus proche de 1. Ainsi mécaniquement à chaque ajout de variable au modèle, la prédiction de Y , la variable cible, sera meilleur et R^2 sera plus élevé.

Estimation

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Quelle différence entre R^2 et R^2 ajusté ?

R^2 permet de comparer des modèles ayant le même nombre de variables explicatives et à tendance à augmenter quand le nombre de variables expl. dans le modèle augmente

R^2 ajusté est utilisé pour comparer des modèles avec un nombre de variables explicatives différent et reste globalement stable même avec un nombre de variables explicatives qui augmente.

Pour rappel : R^2 ajusté = $1 - \frac{SSR / (n - p - 1)}{SST / (n - 1)}$

Critères AIC et BIC

$$Y = X\beta + \epsilon$$

Y : variable qu'on souhaite estimée (de sortie)
X : la matrice contenant les variables explicatives 1, x11,x12, …xnp
β : la matrice des coefficients β1, β2, …, βn
ε : vecteur résidu

Le critère d'Akaike (AIC) est un critère permettant de choisir le meilleur modèle. Pour cela on cherche à le minimiser.

$$AIC = -2\ln(L) + 2k$$

$$BIC = -2\ln(L) + k\ln(n)$$

Avec L : le maximum de vraisemblance vu précédemment
k : le nombre de paramètres libres du modèle

En R

```
iris.lm1 <- lm(Petal.Length~Petal.Width+Sepal.Length, data=iris)
```

```
> iris.lm1

Call:
lm(formula = Petal.Length ~ Petal.Width + Sepal.Length, data = iris)

Coefficients:
 (Intercept)   Petal.Width  Sepal.Length
     -1.5071        1.7481        0.5423
```

Ici on a notre vecteur β qui vaut $\begin{bmatrix} 1.74 \\ 0.54 \end{bmatrix}$

En R

Pour l'ensemble des variables :

```
> iris.lm2 <- lm(Petal.Length~., data=iris)
> iris.lm2

Call:
lm(formula = Petal.Length ~ ., data = iris)

Coefficients:
      (Intercept)       Sepal.Length        Sepal.Width        Petal.Width  Speciesversicolor
          -1.1110             0.6080            -0.1805             0.6022             1.4634
 Speciesvirginica
           1.9742
```

Le reste des commandes (validation des hypothèses, prédiction, …) est similaire dans le cas d'une régression linéaire multiple vs régression linéaire simple.

En R

Comparons maintenant deux modèles de régression :

```
iris.lm2 <- lm(Petal.Length~., data=iris)
```

```
> AIC(iris.lm2)
[1] 32.56654
```

```
> BIC(iris.lm2)
[1] 53.64099
```

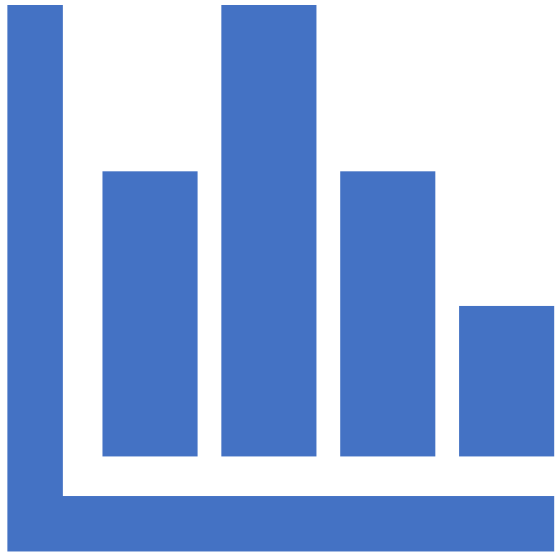
```
> summary(iris.lm2)$r.squared
[1] 0.9785933
```

```
iris.lm1 <- lm(Petal.Length~Petal.Width+Sepal.Length, data=iris)
```

```
> AIC(iris.lm1)
[1] 158.1792
```

```
> BIC(iris.lm1)
[1] 170.2218
```

```
> summary(iris.lm1)$r.squared
[1] 0.9485236
```



Différence ANOVA et régressions ?

- La régression et l'ANOVA (analyse de variance) sont deux méthodes de la théorie statistique permettant d'analyser le comportement d'une variable par rapport à une autre.
- En régression, il s’agit souvent de la variation de la variable dépendante basée sur la variable indépendante
- En ANOVA, il s’agit de la variation des attributs de deux échantillons de deux populations
- ANOVA et Régression sont les deux versions du modèle linéaire général (GLM). L'ANOVA est basée sur des variables prédictives catégorielles, tandis que la régression est basée sur des variables prédictives quantitatives.

Modèles linéaires généralisés (GLM)

Définition

L'ANOVA ou même les régressions, nous permette d'effectuer des analyses (voir des prédictions) sur des variables de types numériques (une taille, un salaire, une superficie, …).

Mais quid des variables non continues ?

→ C'est la que les GLM interviennent

Les GLM sont **principalement** utilisés dans **deux situations** :

1. Lorsque les **données sont de type comptage** (nombre de questionnaire remplis, etc..),
2. Lorsque les **données sont de type binaire** (Malade/non malade ou mort/vivant)

Equation

Lorsqu'on construit un modèle de régression logistique, on suppose qu'il existe une fonction *f* qui lie la variable cible *Y* aux variables explicatives représentées dans la matrice *X* de la manière suivante :

$$P(Y = 1) = f(X) + \epsilon$$

$$f(X) = \frac{1}{1 + exp(-(\beta_0 + X_1\beta_1 + ... + X_p\beta_p))}$$

Avec toujours ε l'erreur

La Régression Logistique Généralisée est très vaste, nous nous focaliserons sur l'étude de la régression logistique ordinaire –la plus courante- de la famille des modèles binomiaux (en R cela signifie que nous utiliserons la fonctions glm() avec comme argument family=binomial(logit)

En R

Mes données sont les suivants :

```
library(questionr)
data(hdv2003)
data = hdv2003
data
```

```
> str(data)
'data.frame':   2000 obs. of  20 variables:
 $ id           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ age          : int  28 23 59 34 71 35 60 47 20 28 ...
 $ sexe         : Factor w/ 2 levels "Homme","Femme": 2 2 1 1 2 2 2 1 2 1 ...
 $ nivetud      : Factor w/ 8 levels "N'a jamais fait d'etudes",..: 8 NA 3 8 3 6 3 6 NA 7 ...
 $ poids        : num  2634 9738 3994 5732 4329 ...
 $ occup        : Factor w/ 7 levels "Exerce une profession",..: 1 3 1 1 4 1 6 1 3 1 ...
 $ qualif       : Factor w/ 7 levels "Ouvrier specialise",..: 6 NA 3 3 6 6 2 2 NA 7 ...
 $ freres.soeurs: int  8 2 2 1 0 5 1 5 4 2 ...
 $ clso         : Factor w/ 3 levels "Oui","Non","Ne sait pas": 1 1 2 2 1 2 1 2 1 2 ...
 $ relig        : Factor w/ 6 levels "Pratiquant regulier",..: 4 4 4 3 1 4 3 4 3 2 ...
 $ trav.imp     : Factor w/ 4 levels "Le plus important",..: 4 NA 2 3 NA 1 NA 4 NA 3 ...
 $ trav.satisf  : Factor w/ 3 levels "Satisfaction",..: 2 NA 3 1 NA 3 NA 2 NA 1 ...
 $ hard.rock    : Factor w/ 2 levels "Non","Oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ lecture.bd   : Factor w/ 2 levels "Non","Oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ peche.chasse : Factor w/ 2 levels "Non","Oui": 1 1 1 1 1 1 2 2 1 1 ...
 $ cuisine      : Factor w/ 2 levels "Non","Oui": 2 1 1 2 1 1 2 2 1 1 ...
 $ bricol       : Factor w/ 2 levels "Non","Oui": 1 1 1 2 1 1 1 2 1 1 ...
 $ cinema       : Factor w/ 2 levels "Non","Oui": 1 2 1 2 1 2 1 1 2 2 ...
 $ sport        : Factor w/ 2 levels "Non","Oui": 1 2 2 2 1 2 1 1 1 2 ...
 $ heures.tv    : num  0 1 0 2 3 2 2.9 1 2 2 ...
```

En R

Nous souhaitons regarder l'effet du sexe, de l'age, du niveau d'étude, de la religion et du nombre d'heures devant la TV par jour sur le fait de pratiquer (ou non) un sport

```
> reg=glm(sport ~ sexe + age + nivetud + relig + heures.tv,data = data, family = binomial(logit))
> summary(reg)

Call:
glm(formula = sport ~ sexe + age + nivetud + relig + heures.tv,
    family = binomial(logit), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6338  -0.8989  -0.5083   1.0288   2.5414

Coefficients:

                                                                Estimate Std. Error z value
(Intercept)                                                     -1.49010    1.05180  -1.417
sexeFemme                                                       -0.38230    0.10984  -3.481
age                                                             -0.02747    0.00405  -6.783
```

En R

Quid de la prédiction ?

Grâce à ma fonction GLM créée précédemment, je souhaiterai la tester et ainsi prédire ma réponse (le fait de pratiquer un sport)

```
> sport.pred <- predict(reg, type = "response", newdata = data)
> head(sport.pred)
        1         2         3         4         5         6
0.6378029        NA 0.1649719 0.6923829 0.0882091 0.3191572
```

En R

Quid de la prédiction ?

Grâce à ma fonction GLM créée précédemment, je souhaiterai la tester et ainsi prédire ma réponse (le fait de pratiquer un sport)

```
> sport.pred <- predict(reg, type = "response", newdata = data)
> head(sport.pred)
        1         2         3         4         5         6
0.6378029        NA 0.1649719 0.6923829 0.0882091 0.3191572
```

Mes résultats sont sous forme de probabilité (entre 0 & 1), usuellement les probabilités sont regroupés en Oui/Non (booléen) selon qu'elles soient supérieures ou inférieures à la moitié :

```
> table(sport.pred > 0.5, data$sport)

       Non  Oui
FALSE 1069  368
TRUE   179  267
```

En R

Interprétation du tableau de contingence

```
> table(sport.pred > 0.5, data$sport)

        Non  Oui
  FALSE 1069  368
  TRUE   179  267
```

	Non	Oui
Faux	Faux Négatif	Vrai Négatif
Vrai	Faux Positif	Vrai positif

Mauvais classement

Nous avons donc 583 (384+199) prédictions incorrectes sur un total de 1993, soit un taux de mauvais classement de 29,3 %.

En R

Optimisation du modèle

La fonction step() permet justement de sélectionner le meilleur modèle par une procédure pas à pas descendante basée sur la minimisation de l'AIC. La fonction affiche à l'écran les différentes étapes de la sélection et renvoie le modèle final.

```
> reg2 <- step(reg)
Start:  AIC=2097.45
sport ~ sexe + age + nivetud + relig + heures.tv

            Df Deviance    AIC
- relig      5   2070.0 2092.0
<none>           2065.4 2097.4
- heures.tv  1   2075.5 2105.5
- sexe       1   2077.6 2107.6
- age        1   2113.5 2143.5
- nivetud    7   2195.8 2213.8

Step:  AIC=2092
sport ~ sexe + age + nivetud + heures.tv

            Df Deviance    AIC
<none>           2070.0 2092.0
- heures.tv  1   2080.2 2100.2
- sexe       1   2081.0 2101.0
- age        1   2115.1 2135.1
- nivetud    7   2200.6 2208.6
```