Rapport : Probabilités et Statistiques

Compte-rendu du projet d'étude

Modèle de régression linaire

MAM 3
Année 2024 - 2025

Polytech Nice Sophia

Rédigé par :
Elsa Catteau, Zineb Ziad et Safia Zaari Jabri

Sommaire

1 Introduction et objectifs

Bien souvent, on essaye de comprendre l'influence de certains facteurs sur une notion particulière que l'on souhaiterait expliquer. Par exemple, on pourrait étudier les conséquences de la météo sur une culture de légumes, ou encore chercher le lien entre le nombre d'heures de revision et la note qui pourrait être obtenue. Dans ce dernier cas, un modéle de régression linéaire simple est le suivant :

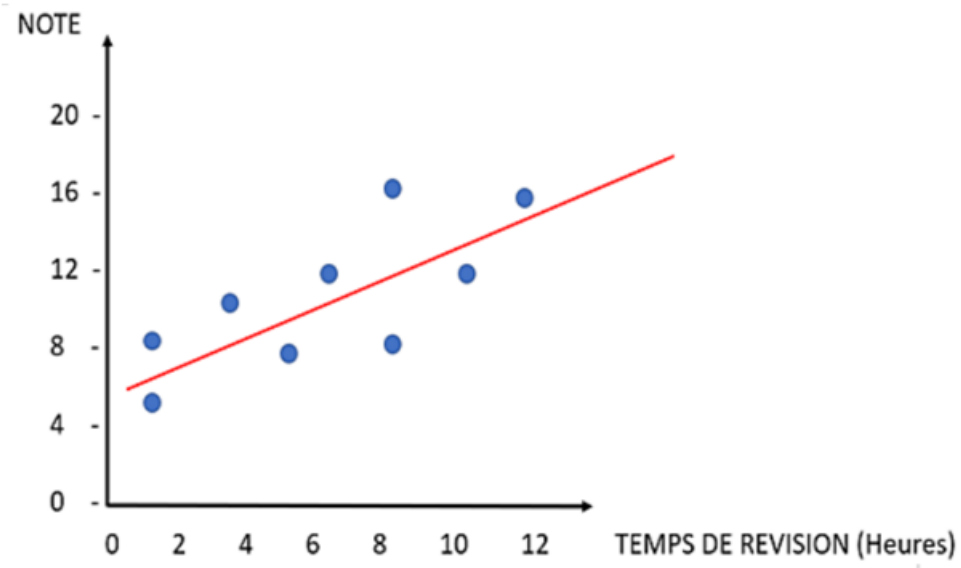


Figure 1: Exemple tiré du site ipgrade.com

A partir d'une série d'observations (les points bleus), on souhaite donc déterminer la droite (la courbe rouge) qui passe au plus près des points.
La régression linéaire simple se distingue de la régression linéaire multiple. Dans les exemples cités ci-dessus, il s'agissait de modéle de régression linéaire simple car on considère l'influence d'un seul facteur seulement. La régression linéaire multiple, quant à elle, va tenir compte non pas d'un seul facteur mais de plusieurs. On sait notamment qu'une culture de légumes ne dépend pas uniquement de la météo, mais aussi de la qualité des sols, de la pollution de l'eau, des pesticides, mais aussi des régions dans lesquelles les productions sont menées, etc.
En termes mathématiques, la régression linéaire simple ou multiple a pour but de modéliser la relation linéaire entre une ou des variables explicatives et une variable à expliquer. L'objectif est alors de trouver la meilleure approximation linéaire de la relation entre ces deux variables.

2 Régression linaire simple

On cherche à savoir s'il existe une relation fonctionnelle entre la variable explicative x_i et la variable réponse y_i, autrement dit, s'il existe une fonction f telle que les y_i peuvent être approximés par une fonction $f(x_i)$.
On considère alors le risque empirique défini par :

$$R_n(g) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - g(X_i))^2$$

On va donc vouloir minimiser ce risque pour des fonctions du type g du type $g(x) = ax + b$.

2.1 Modèle théorique

Le modèle de régression linéaire simple s'écrit :

$$Y_i = a\,x_i + b + \varepsilon_i$$

où a et b sont les paramètres à estimer, et ε_i représente le terme d'erreur.
Les erreurs ε_i sont supposées indépendantes, de moyenne nulle $\mathbb{E}[\varepsilon_i] = 0$ et de variance constante $\mathbb{V}[\varepsilon_i] = \sigma^2$. Dans l'hypothèse de gaussianité du bruit, les variables ε_i sont des variables aléatoires indépendantes et de même loi et les Y_i sont également indépendantes mais pas de même loi.
De plus, on a souvent l'hypothèse

2.2 Estimation des paramètres

Le principe des moindres carrés correspond à la minimisation du risque empirique vu précédemment. Ce principe nous permet donc de déterminer les estimateurs $\hat{a}_n$ de a et $\hat{b}_n$ de b.

Détermination de $\hat{b}_n$:
On cherche à minimiser le risque empirique défini par :

$$R_n(g) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - ax_i - b)^2$$

Calculons la dérivée partielle de R_n par rapport à b :

$$\frac{\partial R_n}{\partial b} = \frac{1}{n}\sum_{i=1}^{n} 2(Y_i - ax_i - b)(-1) = -\frac{2}{n}\sum_{i=1}^{n}(Y_i - ax_i - b)$$

Pour minimiser R_n, on annule cette dérivée :

$$-\frac{2}{n}\sum_{i=1}^{n}(Y_i - ax_i - b) = 0 \quad \Longrightarrow \quad \frac{1}{n}\sum_{i=1}^{n}(Y_i - ax_i - b) = 0$$

On obtient alors :

$$\frac{1}{n}\sum_{i=1}^{n} Y_i - a\frac{1}{n}\sum_{i=1}^{n} x_i - b = 0 \iff \bar{Y}_n - a\bar{x}_n - b = 0$$

D'où le résultat :

$$\boxed{\hat{b}_n = \bar{Y}_n - \hat{a}_n\,\bar{x}_n}$$

Détermination de $\hat{a}_n$:
On part du risque empirique :

$$R_n(g) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - ax_i - b)^2$$

On dérive par rapport à a :

$$\frac{\partial R_n}{\partial a} = -\frac{2}{n}\sum_{i=1}^{n} x_i(Y_i - ax_i - b)$$

On annule la dérivée :

$$\sum_{i=1}^{n} x_i Y_i = a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i$$

On sait que $b = \bar{Y}_n - a\bar{x}_n$:

$$\begin{aligned}
&\sum_{i=1}^{n} x_i Y_i = a\sum_{i=1}^{n} x_i^2 + \bar{Y}_n\sum_{i=1}^{n} x_i - a\bar{x}_n\sum_{i=1}^{n} x_i \\
\iff\quad &\sum_{i=1}^{n} x_i Y_i - \bar{Y}_n\sum_{i=1}^{n} x_i = a\left(\sum_{i=1}^{n} x_i^2 - \bar{x}_n\sum_{i=1}^{n} x_i\right) \\
\iff\quad &\sum_{i=1}^{n} x_i Y_i - n\bar{Y}_n\bar{x}_n = a\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}_n^2\right) \\
\iff\quad &a = \frac{\sum_{i=1}^{n} x_i Y_i - n\bar{Y}_n\bar{x}_n}{\sum_{i=1}^{n} x_i^2 - n\bar{x}_n^2}
\end{aligned}$$

D'où le résultat :

$$\boxed{\hat{a}_n = \frac{\sum_{i=1}^n x_i Y_i - n\bar{Y}_n \bar{x}_n}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$$

2.3 Montrons que les estimateurs sont sans biais

Pour $\hat{a}_n$:

On considère le modèle suivant :

$$Y_i = ax_i + b + \varepsilon_i, \quad \text{avec} \quad \mathbb{E}[\varepsilon_i] = 0$$

On rappelle que :

$$\hat{a}_n = \frac{\sum_{i=1}^n x_i Y_i - n\bar{x}_n \bar{Y}_n}{\sum_{i=1}^n (x_i - \bar{x}_n)^2} \quad \text{et} \quad \hat{b}_n = \bar{Y}_n - \hat{a}_n \bar{x}_n$$

Commençons par $\hat{a}_n$ et développons son numérateur :

$$\sum_{i=1}^n x_i Y_i = \sum_{i=1}^n x_i(ax_i + b + \varepsilon_i) = a\sum_{i=1}^n x_i^2 + b\sum_{i=1}^n x_i + \sum_{i=1}^n x_i \varepsilon_i$$

Or :

$$\bar{x}_n = \frac{1}{n}\sum_{i=1}^n x_i, \quad \bar{Y}_n = \frac{1}{n}\sum_{i=1}^n Y_i = a\bar{x}_n + b + \bar{\varepsilon}_n$$

$$n\bar{x}_n\bar{Y}_n = n\bar{x}_n(a\bar{x}_n + b + \bar{\varepsilon}_n) = an\bar{x}_n^2 + bn\bar{x}_n + n\bar{x}_n\bar{\varepsilon}_n$$

Donc le numérateur devient :

$$\sum_{i=1}^n x_i Y_i - n\bar{x}_n\bar{Y}_n = a\left(\sum_{i=1}^n x_i^2 - n\bar{x}_n^2\right) + \left(\sum_{i=1}^n x_i\varepsilon_i - n\bar{x}_n\bar{\varepsilon}_n\right)$$

On reconnaît :

$$\sum_{i=1}^n (x_i - \bar{x}_n)^2 = \sum_{i=1}^n x_i^2 - n\bar{x}_n^2$$

Donc :

$$\hat{a}_n = \frac{a\sum_{i=1}^n (x_i - \bar{x}_n)^2 + \sum_{i=1}^n x_i\varepsilon_i - n\bar{x}_n\bar{\varepsilon}_n}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

Espérance de $\hat{a}_n$:

$$\mathbb{E}[\hat{a}_n] = a + \mathbb{E}\left[\frac{\sum_{i=1}^n x_i\varepsilon_i - n\bar{x}_n\bar{\varepsilon}_n}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}\right]$$

Or :

$$\mathbb{E}\left[\sum_{i=1}^n x_i\varepsilon_i\right] = \sum_{i=1}^n x_i\mathbb{E}[\varepsilon_i] = 0 \quad \text{et} \quad \mathbb{E}[\bar{\varepsilon}_n] = 0$$

Donc :

$$\mathbb{E}[\hat{a}_n] = a$$

On en conclu que $\hat{a}_n$ est sans biais.

Maintenant, pour $\hat{b}_n$:

$$\hat{b}_n = \bar{Y}_n - \hat{a}_n\bar{x}_n \quad \Rightarrow \quad \mathbb{E}[\hat{b}_n] = \mathbb{E}[\bar{Y}_n] - \bar{x}_n\mathbb{E}[\hat{a}_n]$$

Or :

$$\mathbb{E}[\bar{Y}_n] = a\bar{x}_n + b, \quad \mathbb{E}[\hat{a}_n] = a$$

Donc :

$$\mathbb{E}[\hat{b}_n] = a\bar{x}_n + b - a\bar{x}_n = b$$

On en conclu que $\hat{b}_n$ est sans biais.

2.4 Démontrons les formules de variances

Pour $\hat{a}_n$:

On remplace $Y_i = ax_i + b + \varepsilon_i$ dans l'expression de $\hat{a}_n$:

$$\hat{a}_n = \frac{\sum_{i=1}^n (x_i - \bar{x}_n)(ax_i + b + \varepsilon_i - \bar{Y}_n)}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

Mais $\bar{Y}_n = a\bar{x}_n + b + \bar{\varepsilon}_n$, donc :

$$Y_i - \bar{Y}_n = a(x_i - \bar{x}_n) + (\varepsilon_i - \bar{\varepsilon}_n)$$

Ainsi :

$$\hat{a}_n = \frac{\sum_{i=1}^n (x_i - \bar{x}_n)\left[a(x_i - \bar{x}_n) + (\varepsilon_i - \bar{\varepsilon}_n)\right]}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

On sépare les deux termes :

$$\hat{a}_n = a + \frac{\sum_{i=1}^n (x_i - \bar{x}_n)(\varepsilon_i - \bar{\varepsilon}_n)}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

$$\hat{a}_n - a = \frac{\sum_{i=1}^n (x_i - \bar{x}_n)(\varepsilon_i - \bar{\varepsilon}_n)}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

Mais comme $\bar{\varepsilon}_n = \frac{1}{n}\sum_{i=1}^n \varepsilon_i$, alors :

$$\hat{a}_n - a = \frac{\sum_{i=1}^n (x_i - \bar{x}_n)\varepsilon_i - \bar{\varepsilon}_n \sum_{i=1}^n (x_i - \bar{x}_n)}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

Or :

$$\sum_{i=1}^n (x_i - \bar{x}_n) = 0 \quad \Rightarrow \quad \hat{a}_n - a = \frac{\sum_{i=1}^n (x_i - \bar{x}_n)\varepsilon_i}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}$$

On a alors :

$$\mathrm{Var}(\hat{a}_n) = \mathrm{Var}\left(\frac{\sum_{i=1}^n (x_i - \bar{x}_n)\varepsilon_i}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}\right)$$

Par linéarité de la variance (et par indépendance des ε_i) :

$$\mathrm{Var}(\hat{a}_n) = \frac{1}{\left(\sum_{i=1}^n (x_i - \bar{x}_n)^2\right)^2} \cdot \sum_{i=1}^n (x_i - \bar{x}_n)^2 \sigma^2$$

Finalement, on a obtient bien cette formule :

$$\boxed{\mathrm{Var}(\hat{a}_n) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$$

Maintent pour $\hat{b}_n$

$$\hat{b}_n = \bar{Y}_n - \hat{a}_n \bar{x}_n$$

Donc :

$$\mathbb{V}[\widehat{b}_n] = \mathbb{V}[\bar{Y}_n] + \bar{x}_n^2 \cdot \mathbb{V}[\widehat{a}_n] - 2\bar{x}_n \cdot \mathrm{Cov}(\bar{Y}_n, \widehat{a}_n)$$

Or, dans le cadre de l'hypothèse de bruit centré, on a :

$$\mathrm{Cov}(\bar{Y}_n, \widehat{a}_n) = 0 \quad \text{et} \quad \mathbb{V}[\bar{Y}_n] = \frac{\sigma^2}{n}$$

De plus, on a :

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i = a\bar{x}_n + b + \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i = a\bar{x}_n + b + \bar{\varepsilon}_n$$

avec $\bar{\varepsilon}_n = \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i$.
Comme les ε_i sont indépendants et de même loi, on a :

$$\mathbb{V}[\bar{\varepsilon}_n] = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{V}[\varepsilon_i] = \frac{1}{n^2}\cdot n\cdot\sigma^2 = \frac{\sigma^2}{n}$$

D'où :

$$\mathbb{V}[\bar{Y}_n] = \mathbb{V}[a\bar{x}_n + b + \bar{\varepsilon}_n] = \mathbb{V}[\bar{\varepsilon}_n] = \frac{\sigma^2}{n}$$

Donc :

$$\mathbb{V}[\widehat{b}_n] = \frac{\sigma^2}{n} + \bar{x}_n^2\cdot\frac{\sigma^2}{\sum_{i=1}^{n}(x_i-\bar{x}_n)^2} = \sigma^2\left(\frac{1}{n} + \frac{\bar{x}_n^2}{\sum_{i=1}^{n}(x_i-\bar{x}_n)^2}\right)$$

On utilise ensuite l'égalité suivante :

$$\sum_{i=1}^{n} x_i^2 = n\cdot\bar{x}_n^2 + \sum_{i=1}^{n}(x_i-\bar{x}_n)^2 \quad\Rightarrow\quad n\bar{x}_n^2 = \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n}(x_i-\bar{x}_n)^2$$

On a donc :

$$\mathbb{V}[\widehat{b}_n] = \frac{\sigma^2}{n}\left(1 + \frac{n\bar{x}_n}{\sum_{i=1}^{n}(x_i-\bar{x}_n)^2}\right) = \frac{\sigma^2}{n}\left(1 + \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n}(x_i-\bar{x}_n)^2} - 1\right)$$

Finalement :

$$\boxed{\mathbb{V}[\widehat{b}_n] = \frac{\sigma^2\sum x_i^2}{n\sum(x_i-\bar{x})^2}}$$

2.5 Démonstrons la formule de l'intervalle de prédiction

On se place toujours dans le cadre du modèle de régression linéaire simple :

$$Y_i = ax_i + b + \varepsilon_i$$

Pour une nouvelle valeur x_{new}, la prédiction ponctuelle est :

$$\hat{Y}_{\text{new}} = \hat{a}_n x_{\text{new}} + \hat{b}_n$$

Estimateur de la prédiction:

$$\widehat{Y}(x_{\text{new}}) = \widehat{a}_n x_{\text{new}} + \widehat{b}_n$$

L'erreur de prédiction s'écrit :

$$\begin{aligned} Y_{\text{new}} - \hat{Y}_{\text{new}} &= (ax_{\text{new}} + b + \varepsilon_{\text{new}}) - (\hat{a}_n x_{\text{new}} + \hat{b}_n) \\ &= (a-\hat{a}_n)x_{\text{new}} + (b-\hat{b}_n) + \varepsilon_{\text{new}} \end{aligned}$$

La variance de l'erreur de prédiction s'écrit :

$$\begin{aligned} \mathbb{V}(Y_{\text{new}} - \hat{Y}_{\text{new}}) &= \mathbb{V}\left((a-\hat{a}_n)x_{\text{new}} + (b-\hat{b}_n) + \varepsilon_{\text{new}}\right) \\ &= \mathbb{V}(\hat{a}_n x_{\text{new}} + \hat{b}_n) + \mathbb{V}(\varepsilon_{\text{new}}) \end{aligned}$$

Puisque ε_{new} est indépendant des estimateurs $\hat{a}_n$ et $\hat{b}_n$, les covariances entre ε_{new} et les autres termes sont nulles. Ainsi,

$$\mathbb{V}(Y_{\text{new}} - \hat{Y}_{\text{new}}) = x_{\text{new}}^2\mathbb{V}(\hat{a}_n) + \mathbb{V}(\hat{b}_n) + 2x_{\text{new}}\text{Cov}(\hat{a}_n,\hat{b}_n) + \sigma^2$$

En utilisant les expressions connues :

$$\mathbb{V}(\hat{a}_n) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$\mathbb{V}(\hat{b}_n) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right)$$
$$\mathrm{Cov}(\hat{a}_n, \hat{b}_n) = -\frac{\sigma^2 \bar{x}}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

En substituant :

$$\begin{aligned}\mathbb{V}(Y_{\text{new}} - \hat{Y}_{\text{new}}) &= \sigma^2 \left[\frac{x_{\text{new}}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} + \frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} - \frac{2 x_{\text{new}} \bar{x}}{\sum_{i=1}^n (x_i - \bar{x})^2} + 1 \right] \\ &= \sigma^2 \left[1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]\end{aligned}$$

L'erreur standardisée suit une loi de Student :

$$\frac{Y_{\text{new}} - \hat{Y}_{\text{new}}}{\sigma \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}} \sim \mathcal{N}(0,1)$$

et comme $\frac{(n-2)\hat{\sigma}_n^2}{\sigma^2} \sim \chi^2(n-2)$, on a :

$$\frac{Y_{\text{new}} - \hat{Y}_{\text{new}}}{\hat{\sigma}_n \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}} \sim t_{n-2}$$

On a donc :

$$P\left(-t_{1-\alpha/2;n-2} \leq \frac{Y_{\text{new}} - \hat{Y}_{\text{new}}}{\hat{\sigma}_n \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}} \leq t_{1-\alpha/2;n-2} \right) = 1 - \alpha$$

Ce qui donne finalement l'intervalle de confiance pour la prédiction de Y:

$$\boxed{\hat{a}_n x_{\text{new}} + \hat{b}_n \pm \hat{\sigma}_n \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}} \cdot t_{1-\alpha/2;n-2}}$$

2.6 Vérification de l'hypothèse de gaussianité du bruit

Avant de se lancer dans une régression linéaire, il est important de savoir si celle-ci sera pertinente. Pour cela, il faut d'abord vérifier si le bruit est normalement distribué. On examine donc si les résidus suivent une loi normale.

$$\hat{\varepsilon}_{i,\text{sd}} = \frac{Y_i - \hat{Y}_i}{\hat{\sigma}_n \sqrt{1 - h_i}}$$

où h_i est le i-ème terme diagonal de la matrice de projection, et où la matrice des observations $\mathbf{X}$ est :

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

On a alors :

$$\hat{\varepsilon}_{i,\text{sd}} \xrightarrow{(\mathcal{L})} \mathcal{N}(0,1)$$

Afin de valider l'hypothèse de gaussianité du bruit, on vérifie que les résidus standardisés suivent approximativement une loi normale standard à l'aide d'un graphe `qqnorm`.
On obtient alors ce graphe :

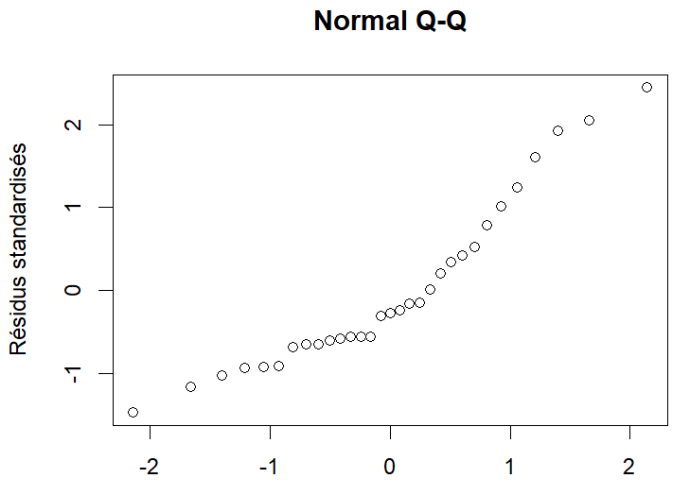


Figure 2: QQ-norm sur les résidus standardisés

On remarque que le nuage de point est bien proche de la première bissectrice. Mais, nous ne comparons qu'‘a une loi asymptotique, il faut donc introduire les résidus studentisés. Pour obtenir les résidus studentisés, nous avons alors utilisé la fonction rstudent. On peut aussi effectuer un test d'adéquation de Kolmogorov afin de verifier l'hypothèse de gaussianité sur les résidus studentisés qui suivent une loi de Student à (n-3) degrés de liberté

$$\hat{\varepsilon}_{i,\mathrm{st}} = \frac{Y_i - \hat{Y}_i}{\hat{\sigma}_{n,i}\sqrt{1-h_i}}$$

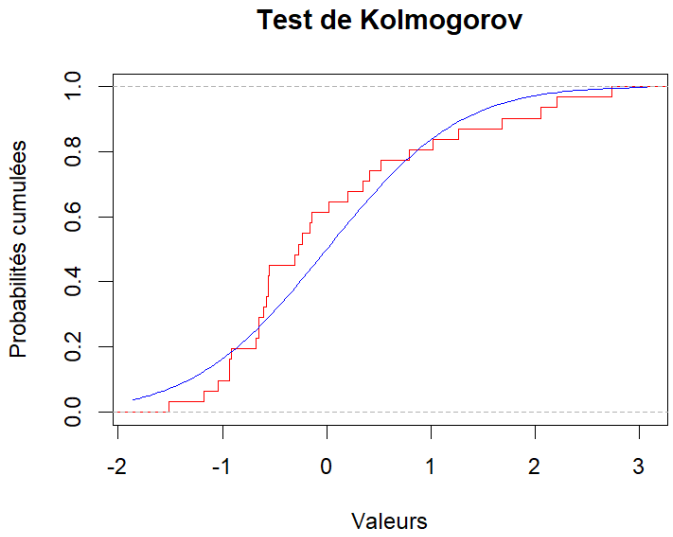


Figure 3: Test de Kolmogorov sur les résidus studentisés

Les résidus de Student suivent bien une loi de Student à n-3 degrés de liberté. Pour vérifier cela on calcule une p-valeur, la notre égale à 0,3079 au dessus de 0,05, on considère bien que les observations peuvent être modélisées par la loi théorique considérée. Maintenant que l'on considère bien que le bruit est gaussien il est donc possible dans le cas de nos observations, d'effectuer une régression lineaire simple.

2.7 Visualisation

Voici le graphe de notre régression linéaire pour la variable disp :

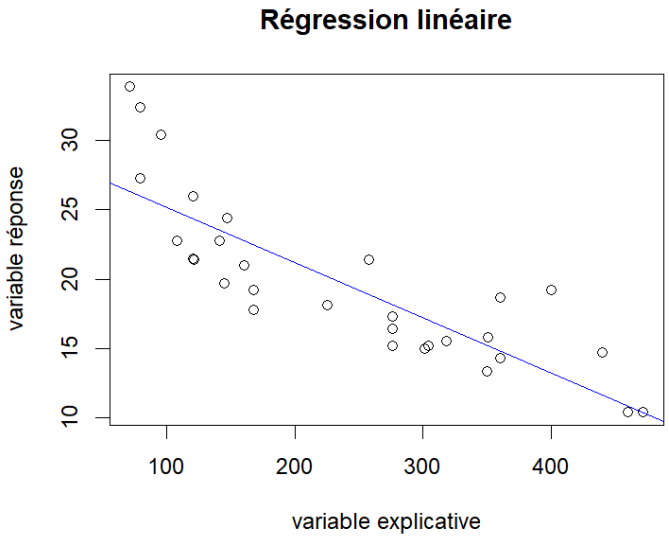


Figure 4: Régression linéaire simple

On observe le nuage de points sur lequel a été ajoutée la droite de régression linéaire.

```
Call:
lm(formula = Yi ~ xi)

Residuals:
   Min     1Q Median     3Q    Max
-4.671 -2.065 -0.862  1.473  7.586

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 29.145800   1.268635  22.974  < 2e-16 ***
xi          -0.039825   0.004791  -8.313 3.65e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.219 on 29 degrees of freedom
Multiple R-squared:  0.7044,    Adjusted R-squared:  0.6942
F-statistic: 69.11 on 1 and 29 DF,  p-value: 3.652e-09
```

Figure 5: Informations graphique sur la régression linéaire

Voici un listing d'informations que l'on obtient et qui nous serviront dans la suite de notre étude.On retrouve notamment les estimations des paramètres a et b

2.8 Validation de la régression linéaire

La question qu'il se pose maintenant et est-ce que notre régression linaire est-elle bien valide ? Il existe plusieurs moyens de le vérifier.Le premier étant à l'aide du coefficient de détermination R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Plus R^2 est proche de 1, plus la régression linéaire est une bonne modélisation. Ici nous avons un R^2=0,7044, que nous lisons sur les informations précedemment obtenus.Ainsi cela vient confirmer la validation de notre régression linéaire.

Un deuxième moyen de le vérifier consiste au test du paramètre a. Ici on teste deux hypothèses soit H_0 ou H_1

Test d'hypothèse pour le paramètre a :

$$H_0 : a = 0 \quad \text{vs} \quad H_1 : a \neq 0$$

Sous H_0, la statistique de test est :

$$T_a = \frac{\hat{a}_n}{\hat{\sigma}_n \sqrt{\frac{1}{\sum_{i=1}^{n}(x_i - \bar{x}_n)^2}}} \sim \mathcal{T}(n-2)$$

Ici on ne calcule pas de seuil théorique mais plutôt une p-valeur, celle de T_a tel que la p-valeur vaut :

$$p\text{-valeur} = P_{H_0}\left(|T| > |T_{a,\text{obs}}|\right)$$

La décision est basée sur la p-valeur :

- Si la p-valeur $< \alpha$, alors on décide H_1.
- Si la p-valeur $> \alpha$, alors on ne rejette pas H_0.

On pose α=0,05. Nous obtenons une p-valeur égale à 1.825833e-09 Ainsi on on décide H1, ce qui est logique dans notre Ainsi toutes ses méthodes nous permettent de valider notre régression linéaire et montrer que celle -ci est valide.

2.9 Intervalle de confiance

Si on note x_{new} une nouvelle observation de la variable explicative, une prévision de la variable réponse est donnée par : $\hat{y}_{\text{new}} = \hat{a}_n \cdot x_{\text{new}} + \hat{b}_n$.Cependant, les estimateurs $\hat{a}_n$ et $\hat{b}_n$ dépendent du jeu de données d'apprentissage. Pour en tenir compte, on affiche l'intervalle de confiance à 95% (en bleu) mais aussi l'intervalle de prédiction à 95%(en vert).

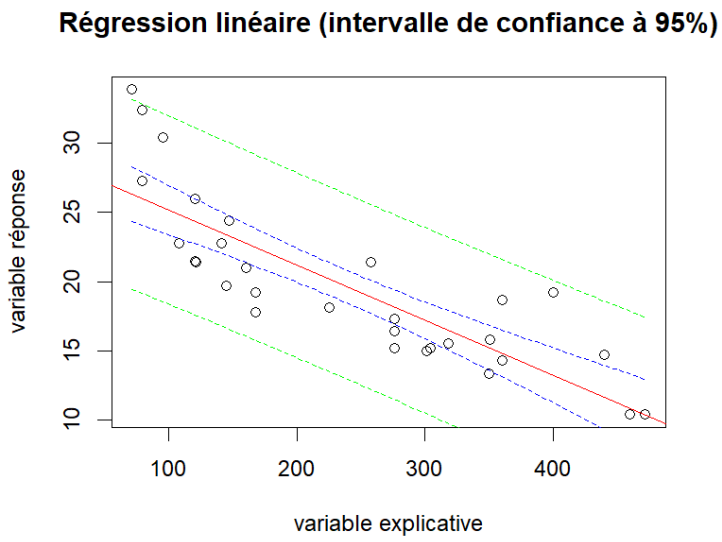


Figure 6: Intervalle de confiance à 95%

Voici le code correspondant à notre graphe :

```
df=data.frame(xi,Yi)
plot(df$xi, df$Yi, main = "R gression lin aire (intervalle de confiance
    95%)", xlab = "variable explicative", ylab = "variable r ponse")
x_seq=seq(min(df$xi),max(df$xi),length.out = 100)
```

```r
abline(lm(Yi ~ xi,data=df), col = "red") #droite de r gression lin aire
x_new= data.frame(xi=x_seq)
pred = predict(lm(Yi~xi), newdata = x_new, interval = "confidence", level =
    0.95)
lines(x_seq, pred[,"lwr"], col = "blue", lty=2) # borne inf rieure
lines(x_seq, pred[,"upr"], col = "blue", lty=2) # borne sup rieure
#Pr diction
pred = predict(lm(Yi~xi), newdata = x_new, interval = "prediction", level =
    0.95)
lines(x_seq, pred[,"lwr"], col = "green", lty=2) # borne inf rieure
lines(x_seq, pred[,"upr"], col = "green", lty=2) # borne sup rieure
```

3 Régression linéaire multiple

On garde les variables qui sont quantitatives dans notre jeu de données, il y en a trois : `qsec`, `drat` et `disp`.

3.1 Vérification de l'hypothèse de gaussianité du bruit

On vérifie l'hypothèse de gaussianité du bruit de la même manière que pour la régression linéaire simple, c'est-à-dire que l'on vérifie que les résidus standardisés et studentisés sont gaussiens.
Pour les résidus standardisés, on utilise la formule suivante :

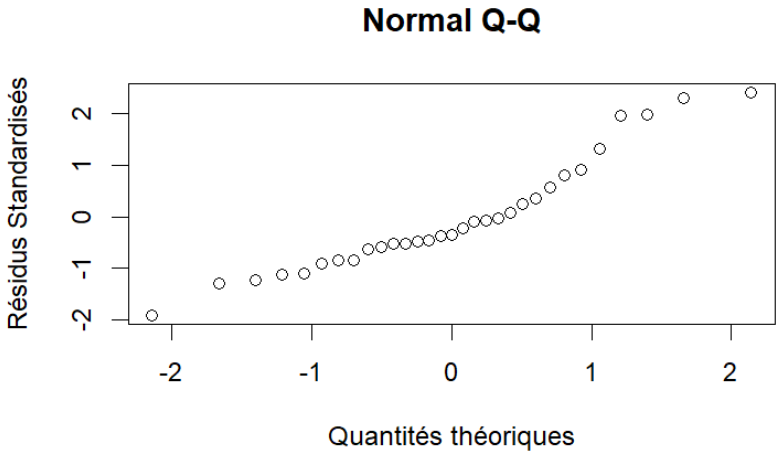
$$\hat{\varepsilon}_{i,\text{sd}} = \frac{Y_i - \hat{Y}_i}{\hat{\sigma}_n \cdot \sqrt{1 - h_i}}$$

avec h_i qui est le i-ème terme diagonal de la matrice $\mathbb{X} \cdot (\mathbb{X}'\mathbb{X})^{-1}\mathbb{X}'$, en vérifiant au préalable que $\mathbb{X}'\mathbb{X}$ est bien inversible.

On a alors :

$$\hat{\varepsilon}_{i,\text{sd}} \xrightarrow{(\mathcal{L})} \mathcal{N}(0,1)$$

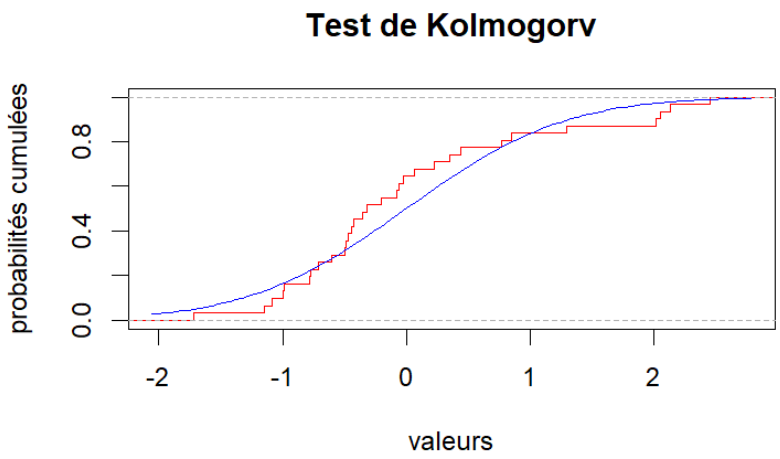
Afin de valider l'hypothèse de gaussianité du bruit, on vérifie que les résidus standardisés suivent approximativement une loi normale standard à l'aide d'un graphe `qqnorm`.
On obtient alors ce graphe :



On voit bien qu'il suit approximativement une loi normale standard.

Mais, nous ne comparons qu'à une loi asymptotique, il faut donc introduire les résidus studentisés. Pour obtenir les résidus studentisés, nous avons alors utilisé la fonction `rstudent`. Puis, on a vérifié l'hypothèse

de gaussianité du bruit en vérifiant que les résidus suivent une loi de Student à $n - \text{rang}(\mathbb{X}) - 1$ degrés de liberté, en réalisant un test d'adéquation de Kolmogorov.
On a alors obtenu ce graphe :



Les résidus de Student suivent bien une loi de Student à $n - \text{rang}(\mathbb{X}) - 1$ degrés de liberté.

3.2 Validation régression linéaire

Tout comme dans la régression lineaire simple nous devons vérifier si celle-ci est bien valide, de même il existe plusieurs méthodes afin de le vérifier.Pour cela il y a la méthode du R^2 que nous verrons un peu plus tard, et celle du test soit :

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0 \quad \text{ou} \quad H_1 : \text{au moins un } \beta_j \neq 0$$

Sous le modèle H_O le modèle s'écrit simplement :

$$\mathbf{Y} = \beta_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \mathcal{U}$$

La statistique de test a effectuer est :

$$F = \frac{\|\hat{Y} - \bar{Y}_n\|^2/(r-1)}{\|Y - \bar{Y}_n\|^2/(n-r)} \overset{H_0}{\sim} \mathcal{F}(r-1, n-r)$$

Tout comme la régression linéaire simple on ne calcule pas un seuil théorique mais une p-valeur afin de choisir si l'hypothèse H_O est bien valide. Pour cela on calcule la p-valeur de F :

- Si la p-valeur $< \alpha$, alors on décide H_1.
- Si la p-valeur $> \alpha$, alors on ne rejette pas H_0.

Dans notre cas on obtiens une p-valeur égale à 1.346396e-07, ainsi notre modèle est bien significatif.

3.3 Intervalle de confiance

Si on note x_{new} une nouvelle observation de la variable explicative, une prévision de la variable réponse est donnée par :

$$\hat{y}_{\text{new}} = \begin{pmatrix} 1 & x_{\text{new}} \end{pmatrix} \hat{\beta}_n.$$

On a choisi de prendre comme x_{new}, la ligne qu'on avait enlevée avec la commande `A[-set1]`, pour que cela corresponde bien à une nouvelle observation. Ensuite, on a utilisé le théorème suivant :

Un intervalle de confiance pour la prédiction de Y pour une nouvelle valeur x_{new} de la variable explicative, de niveau de confiance $100(1-\alpha)\%$, est :

$$\begin{pmatrix}1 & x_{\text{new}}\end{pmatrix}\hat{\beta}_n \pm \hat{\sigma}_n \sqrt{\begin{pmatrix}1 & x_{\text{new}}\end{pmatrix}(\mathbb{X}'\mathbb{X})^{-1}\begin{pmatrix}1 & x_{\text{new}}\end{pmatrix}'} \cdot t_{1-\alpha/2,n-\text{rang}(\mathbb{X})},$$

où :

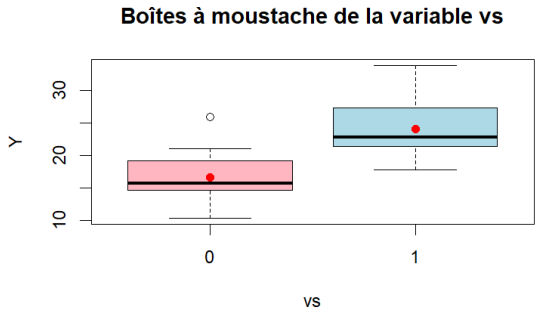
- $\hat{\beta}_n$ est un estimateur de β
- $\hat{\sigma}_n$ est un estimateur sans biais de σ
- $\mathbb{X}$ est la matrice de design,
- $t_{1-\alpha/2,n-\text{rang}(X)}$ est le quantile de la loi de Student à $n - \text{rang}(X)$ degrés de liberté,
- $\alpha = 0.05$ pour un niveau de confiance de 95%.

Au préalable, on a vérifié que $\mathbb{X}'\mathbb{X}$ est inversible. Notre programme affiche une valeur pour la variable réponse qui est de 27.70598 avec un intervalle de confiance à 95% : $[23.72286, 31.68909]$. Voici le code correspondant :

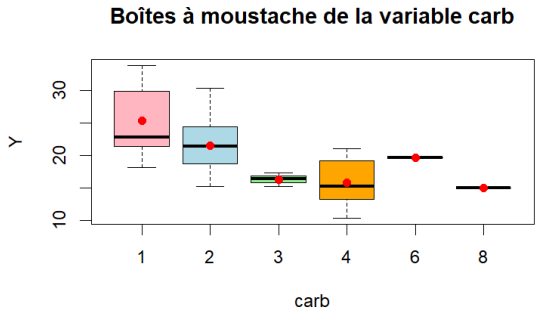
```
# V rifie l'inversibilit  de la matrice
if (abs(det(t(X_matrice) %*% X_matrice)) > 1e-10) {
  beta_chapeau = solve(t(X_matrice) %*% X_matrice) %*% t(X_matrice) %*% Y
} else {
  stop("Matrice non inversible     arr t du programme.")
}

x_new = A[set1, set2[-c(1,2,6,7)]] #on prend la ligne enlev e dans A[-set1,]
    qui correspond    une nouvelle observation
x_new_mat = model.matrix(~ ., data = x_new)
y_new_chapeau = x_new_mat %*% beta_chapeau
val = sigman_chapeau*sqrt((x_new_mat %*% solve(t(X_matrice) %*% X_matrice) %*%
    t(x_new_mat)))
alpha = 0.05
t = qt(1 - alpha/2, df = n - rang) # Quantile
borne_inf = y_new_chapeau - t * val
borne_sup = y_new_chapeau + t * val
cat("Valeur de y_new_chapeau est :", y_new_chapeau, "\n") #affiche 27.70598
cat("Intervalle de confiance    95% : [", borne_inf, ",", borne_sup, "]\n") #
    affiche l'intervalle [ 23.72286 , 31.68909 ]
```

3.4 Analyse de la variance à un facteur

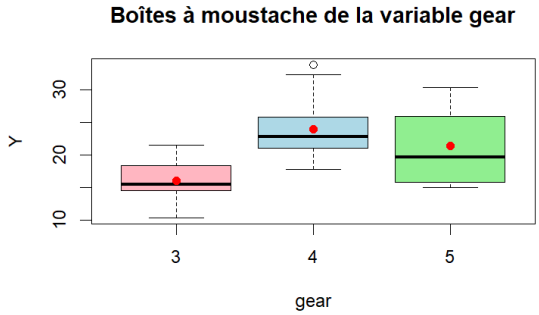
Avant de lancer une analyse de la variance à un facteur, on peut commencer par réaliser une visualisation afin de voir si en effet, le facteur pourrait ou non avoir une influence ou non sur la variable réponse. Pour cela on effectue une représentation graphique pour chacune des variables qualitatives qui fait intervenir des boîtes à moustaches. Dans notre jeu de données, nous avons 4 variables qualitatives : `vs`, `carb`, `gear` et `cyl`. De plus, il est intéressant de faire apparaître la moyenne sur les boîtes à moustaches afin de voir si le facteur à une influence sur la variable réponse. Dans nos graphiques la moyenne est représentée par des points rouges.
On obtient alors ces quatres représentations graphiques :



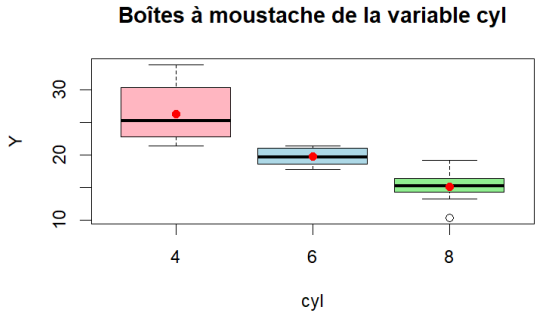
(a) Variable vs



(b) Variable carb



(c) Variable gear



(d) Variable cyl

Le cadre mathématique du modèle d'analyse de la variance à un facteur est le suivant :

$$\forall i \in \{1, \dots, n\}, \quad Y_i = \mu + \sum_{k=1}^{l} \mu_k \cdot \mathbf{1}_{\{x_i = a_k\}} + \epsilon_i$$

avec :

- $a_1, \dots, a_l$: les différentes modalités de la variable explicative
- μ : l'effet commun
- μ_k : effet spécifique de la modalité a_k
- $\forall i \in \{1, \dots, n\}, \quad \mathbb{E}[\epsilon_i] = 0$
- $\forall i \in \{1, \dots, n\}, \quad \mathbb{V}[\epsilon_i] = \sigma^2$
- $\forall i, k \in \{1, \dots, n\}, i \neq k, \quad \text{cov}(\epsilon_i, \epsilon_k) = 0$

Pour mener l'analyse de la variance à un facteur, puisque $\mathbb{X}'\mathbb{X}$ n'est pas une matrice inversible, il faut donc extraire une base au sein de la famille des colonnes de $\mathbb{X}$.
Il y a deux choix à privilégier :

- On supprime la première colonne de $\mathbb{X}$ ce qui revient à faire l'hypothèse $\mu = 0$ (choix mathématique)
- On supprime la seconde colonne de $\mathbb{X}$ ce qui revient à faire l'hypothèse $\mu_1 = 0$ (choix pratique)

Pour tester l'influence ou non du facteur, on va procéder à un test.

Cas $\mu = 0$

Les hypothèses du test sont :

- $\mathcal{H}_0 : \mu_1 = \mu_2 = \ldots = \mu_l$
- $\mathcal{H}_1$: ce n'est pas le cas

Cas $\mu_1 = 0$

Les hypothèses du test sont :

- $\mathcal{H}_0 : \mu_2 = \ldots = \mu_l = 0$
- $\mathcal{H}_1$: ce n'est pas le cas

Ces deux tests sont identiques.

On utilise la commande `summary(aov(lm(`$Y \sim$ variable`)))` qui nous donne un tableau où il y a la donnée `Pr(>F)` qui correspond à la p-valeur.
Ainsi:

- Si p-valeur < 0.05 : on rejette $\mathcal{H}_0$ et on retient $\mathcal{H}_1$
- Si p-valeur ≥ 0.05 : on ne rejette pas $\mathcal{H}_0$

```
L=lm(Y ~ variable)
summary(aov(L))
TUKEY_variable=TukeyHSD(aov(L))
print(TUKEY_variable)
```

En appliquant le code ci-dessus à nos quatres variables qualitatives, on obtient ces quatres tableaux.

```
            Df Sum Sq Mean Sq F value   Pr(>F)
vs           1  423.6   423.6   20.72 8.79e-05 ***
Residuals   29  592.8    20.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(a) Variable vs

```
            Df Sum Sq Mean Sq F value  Pr(>F)
carb         5  462.0   92.39   4.166 0.00685 **
Residuals   25  554.4   22.18
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(b) Variable carb

```
            Df Sum Sq Mean Sq F value   Pr(>F)
gear         2  411.1  205.54   9.509 0.000706 ***
Residuals   28  605.3   21.62
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(c) Variable gear

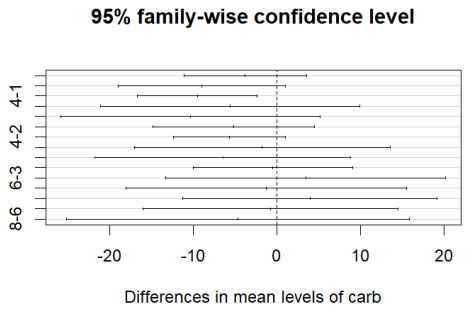
```
            Df Sum Sq Mean Sq F value   Pr(>F)
cyl          2  730.4   365.2   35.77 1.94e-08 ***
Residuals   28  285.9    10.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(d) Variable cyl

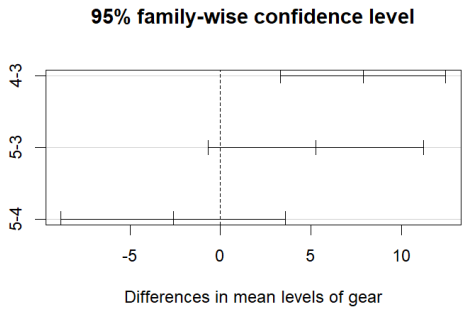
Pour chaque variable on a p-valeur $< 0{,}05$ donc on rejette $\mathcal{H}_0$.
Ensuite, on utilise le test TukeyHSD qui permet d'identifier quelles paires de groupes diffèrent significativement.
Le résultat contient pour chaque paire :

- la différence de moyennes entre les groupes,
- l'intervalle de confiance de cette différence,
- la p-valeur ajustée : si elle est $< 0{,}05$, la différence est significative

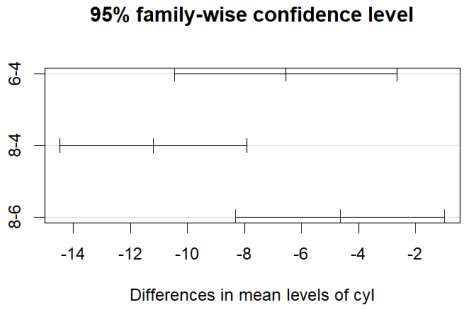
Quand on affiche le graphe correspondant on obtient les intervalles de confiance des différences de moyennes entre les groupes. Si l'intervalle de confiance ne contient pas 0, la différence est significative.



(a) Variable carb



(b) Variable gear



(c) Variable cyl

On ne fait pas de test de TukeyHSD pour la variable vs car cette variable a que deux modalités donc ça n'apporte rien de faire ce test, on obtiendrait qu'un seul intervalle qui ne passerait pas par 0.

3.5 Sélection de variables

Il existe différentes stratégies pour la sélection de variables. Nous avons choisi de le faire en fonction du R_a^2, c'est-à-dire que nous effectuons une sélection de variables afin d'obtenir le meilleur R_a^2. Pour cela, nous avons utilisé la bibliothèque `leaps` du logiciel R.
Dans cette bibliothèque, la fonction `regsubsets` permet de tester toutes les combinaisons possibles (ici de 1 à 3 variables quantitatives) et de trouver les meilleures sous-sélections pour modéliser notre variable Y.

```
X1=X[,-c(1,2,6,7)]
X1=as.matrix(X1) #transforme X1 en matrice
reg = regsubsets(X1, Y, nvmax = 7, method = "exhaustive")
```

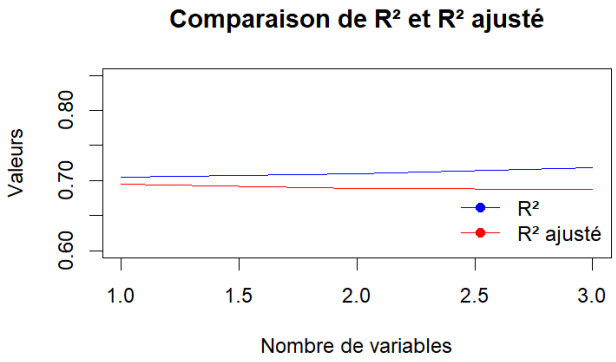
Ensuite, pour connaître le nombre de variables à sélectionner, on peut soit l'observer graphiquement, soit utiliser cette commande qui permet d'identifier le modèle ayant le meilleur R_a^2.

```
summary = summary(reg)
nb = which.max(summary$adjr2)
```

Pour obtenir le graphe comparant R_a^2 et R^2 et nous permettant de voir le nombre de variables à sélectionner, nous avons fait ceci :

```
plot(summary$rsq, type = "l", col = "blue", pch = 16, ylim = c(0.6, 0.85), xlab
    = "Nombre de variables", ylab = "Valeurs",main = "Comparaison de R^2 et R^2
    ajust ") #graphe avec trac  de R_carre
lines(summary$adjr2, type = "l", col = "red", pch = 16) #on trace Ra_carre
legend("bottomright", legend = c("R^2", "R^2 ajust "),col = c("blue", "red"),
   lty=1, pch = 16, bty = "n")
```

Et nous obtenons ce graphe :



Enfin, on peut récupérer le nom des variables à sélectionner en faisant ceci :

```
var = summary$which[nb,-1]
```

On obtient alors cette matrice booléenne :

```
 qsec  disp  drat
FALSE  TRUE FALSE
```

Donc la variable à sélectionner est disp et c'est bien celle qu'on a utilisé dans la régression linéaire simple. Pour la régression linéaire multiple, on peut prendre en compte les 3 variables quantitatives car le R_a^2 décroît très légèrement. Lorsque l'on sélectionne uniquement disp, le R_a^2 vaut 0.694212 alors que si l'on sélection les 3 variables quantitatives(qsec, disp et drat) on a R_a^2 qui vaut 0.6873386.

4 Conclusion

Pour conclure, la régression linéaire est un outil fondamental en statistique permettant de modéliser la relation entre une une variable réponse qui est quantitative, et une ou plusieurs variables explicatives elles aussi quantitatives. Dans le cas de la régression linéaire simple, il y a qu'une seule variable quantitative, le modèle est facile à interpréter, mais limité. Dans le cas de la régression linéaire multiple on utilise plusieurs variables quantitatives, offrant une modélisation plus riche et réaliste. Dans les deux cas, il est essentiel de vérifier les hypothèses du modèle pour garantir la validité des résultats.

5 Annexe : Code

5.1 Régression linéaire simple

```
A=mtcars
F1=as.factor(A[,8]) #prend la 8e colonnne
A[,8]=F1
set.seed(13) #permet de fixer les donn es
set.seed(13*floor(100*runif(1,0,3)))
set1=sample(1:32,1) #donne un nombre al atoire entre 1 et 32
B=A[-set1,] #enl ve la set1- me  ligne de A
Y=B[,1] #1 re  colonne de B
u=1:11 #vecteur 1    11
v=u[-c(1,8,9)] #correspond au vecteur u sans les valeurs 1,8 et 9
set2=c(8,sample(v,6,replace=FALSE)) #vecteur commen ant par 8 puis 6 valeurs
    al atoires qui appartiennent au vecteur v
X=B[,set2] #prend les colonnes de B qui correspondent aux coefficients du
    vecteur set2

xi = X[,4]
Yi = Y
xn_barre=mean(xi)
Yn_barre=mean(Yi)
```

```r
n=length(xi)
an_chapeau = (sum(xi * Yi) - n * xn_barre * Yn_barre) / sum((xi - xn_barre)^2)
bn_chapeau=Yn_barre-an_chapeau*xn_barre
cov=cov(an_chapeau,bn_chapeau)
Yi_chapeau=an_chapeau*xi+bn_chapeau
plot(xi, Yi, main = "R gression lin aire", xlab = "variable explicative",
    ylab = "variable r ponse")
abline(a=bn_chapeau, b=an_chapeau, col = "red")
abline(lm(Yi ~ xi), col = "blue")

#Intervalle de confiance
df=data.frame(xi,Yi)
plot(df$xi, df$Yi, main = "R gression lin aire (intervalle de confiance
    95%)", xlab = "variable explicative", ylab = "variable r ponse")
x_seq=seq(min(df$xi),max(df$xi),length.out = 100)
abline(lm(Yi ~ xi,data=df), col = "red") #droite de r gression lin aire
x_new= data.frame(xi=x_seq)
pred = predict(lm(Yi~xi), newdata = x_new, interval = "confidence", level =
    0.95)
lines(x_seq, pred[,"lwr"], col = "blue", lty=2) # borne inf rieure
lines(x_seq, pred[,"upr"], col = "blue", lty=2) # borne sup rieure
#Pr diction
pred = predict(lm(Yi~xi), newdata = x_new, interval = "prediction", level =
    0.95)
lines(x_seq, pred[,"lwr"], col = "green", lty=2) # borne inf rieure
lines(x_seq, pred[,"upr"], col = "green", lty=2) # borne sup rieure

R_carre = 1 - ( (sum((Yi_chapeau - Yi)^2) )/(sum((Yi - Yn_barre)^2)))
sigman_carre_chapeau=(1/(n-2))*sum((Yi-Yi_chapeau)^2)
sigman_chapeau=sqrt(sigman_carre_chapeau)

#R sidus standardis s
X_matrice = cbind(1, xi) #Cr e la matrice X avec sur la premi re colonne que
    des 1 puis les xi
X_transpose=t(X_matrice)
H = X_matrice %*% solve(t(X_matrice) %*% X_matrice) %*% t(X_matrice)
hi = diag(H)
residus_sd = (Yi - Yi_chapeau) / (sigman_chapeau * sqrt(1 - hi)) #formule des
    r sidus standardis s
qqnorm(residus_sd , main = "Normal Q-Q",ylab="R sidus standardis s",xlab="")

#R sidus studentis s
residus_st = rstudent(lm(Yi ~ xi)) #on r cup re les r sidus studentis s
kolmogorov = ks.test(residus_st , "pt", df = n - 3) #test de Kolomogorov
print(kolmogorov)
plot(ecdf(residus_st), col = "red", main = "Test de Kolmogorov", xlab = "
    Valeurs", ylab = "Probabilit s cumul es", verticals=TRUE, do.points =
    FALSE)
curve(pt(x, df = n - 3), col = "blue", add = TRUE)
```

5.2 Régression linéaire multiple

```r
A=mtcars
F1=as.factor(A[,8]) #prend la 8e colonnne
A[,8]=F1
set.seed(13) #permet de fixer les donn es
set.seed(13*floor(100*runif(1,0,3)))
set1=sample(1:32,1) #donne un nombre al atoire entre 1 et 32
B=A[-set1,] #enl ve la set1- me  ligne de A
Y=B[,1] #1 re  colonne de B
u=1:11 #vecteur 1    11
v=u[-c(1,8,9)] #correspond au vecteur u sans les valeurs 1,8 et 9
```

```r
set2=c(8,sample(v,6,replace=FALSE)) #vecteur commen ant par 8 puis 6 valeurs
    al atoires qui appartiennent au vecteur v
X=B[,set2] #prend les colonnes de B qui correspondent aux coefficients du
    vecteur set2
X1=X[,-c(1,2,6,7)] #on enl ve les variables qualitatives

Yn_barre=mean(Y)
X_matrice=model.matrix(~ ., data = X1) # Cr e la matrice X    partir de X1
n=nrow(X_matrice) #taille

# V rifie l'inversibilit  de la matrice
if (abs(det(t(X_matrice) %*% X_matrice)) > 1e-10) {
  beta_chapeau = solve(t(X_matrice) %*% X_matrice) %*% t(X_matrice) %*% Y
} else {
  stop("Matrice non inversible     arr t du programme.")
}

beta_chapeau = solve(t(X_matrice) %*% X_matrice) %*% t(X_matrice) %*% Y
Y_chapeau=X_matrice%*%beta_chapeau
plot (Y,Y_chapeau,main="^Y en fonction de Y") # Graphe de Y chapeau en fonction
     de Y
abline(a=0,b=1,col="red") #trace la droite

rang=qr(X_matrice)$rank #rang de la matrice
sigman_carre_chapeau=(1/(n-rang))*sum((Y-Y_chapeau)^2)
sigman_chapeau=sqrt(sigman_carre_chapeau)

#Formule de F
F=(sum((Y_chapeau-Yn_barre)^2)/(rang-1))/((sum((Y-Y_chapeau)^2)/(n-rang)))

#R sidus standardis s
H = X_matrice %*% solve(t(X_matrice) %*% X_matrice) %*% t(X_matrice) #matrice H
hi = diag(H)
residus_sd = (Y - Y_chapeau) / (sigman_chapeau * sqrt(1 - hi)) #formule des
    r sidus standardis s
qqnorm(residus_sd, main = "Normal Q-Q",ylab="R sidus Standardis s",xlab="
    Quantit s th oriques")

#R sidus studentis s
X1=as.matrix(X1) #transforme X1 en matrice
residus_st = rstudent(lm(Y ~ X1)) #r cup re les r sidus de student
kolmogorov = ks.test(residus_st, "pt",df=n-rang-1) #test de kolmogorov
plot(ecdf(residus_st),main="Test de Kolmogorv", verticals = TRUE, do.points =
    FALSE, col = "red", xlab = "valeurs", ylab = "probabilit s cumul es")
curve(pt(x, df = n - rang - 1), col = "blue", add = TRUE)

#p-valeur
p_value = 1 - pf(F, df1 = rang - 1, df2 = n - rang)
alpha = 0.05
if (p_value < alpha) {
  cat("On d cide H1\n")
} else {
  cat("On ne rejette pas H0\n")
}

#Intervalle de confiance
x_new = A[set1, set2[-c(1,2,6,7)]] #on prend la ligne enlev e dans A[-set1,]
    qui correspond    une nouvelle observation
x_new_mat = model.matrix(~ ., data = x_new)
y_new_chapeau = x_new_mat %*% beta_chapeau
val = sigman_chapeau*sqrt((x_new_mat %*% solve(t(X_matrice) %*% X_matrice) %*%
    t(x_new_mat)))
alpha = 0.05
```

```r
t = qt(1 - alpha/2, df = n - rang) # Quantile
borne_inf = y_new_chapeau - t * val
borne_sup = y_new_chapeau + t * val
cat("Valeur de y_new_chapeau est :", y_new_chapeau, "\n") #affiche 27.70598
cat("Intervalle de confiance    95% : [", borne_inf, ",", borne_sup, "]\n") #affiche l'intervalle [ 23.72286 , 31.68909 ]

#Analyse de la variance    1 facteur
#On r cup re nos variables qualitatives
vs=B$vs
L1=lm(Y ~ vs)
boxplot(Y ~ vs, main = "Bo tes    moustache de la variable vs", xlab = "vs", ylab = "Y", col = c("lightpink", "lightblue"), names = levels(vs)) #bo te    moustache de vs
moyennes = tapply(Y, vs, mean)
points(1:length(moyennes), moyennes, col = "red", pch = 19)
summary(aov(L1)) #avo(L1) convertit le mod le lin aire L1 en mod le d'analyse de la variance
#summary(aov(L1)) affiche un tableau de d composition de la variance
TUKEY_vs=TukeyHSD(aov(L1)) #applique le test de Tukey
print(TUKEY_vs)
plot(TUKEY_vs)

carb=as.factor(B$carb)
L2=lm(Y ~ carb)
boxplot(Y ~ carb, main = "Bo tes    moustache de la variable carb", xlab = "carb", ylab = "Y", col = c("lightpink", "lightblue","lightgreen", "orange","yellow" ), names = levels(carb)) #bo te    moustache de carb
moyennes = tapply(Y, carb, mean)
points(1:length(moyennes), moyennes, col = "red", pch = 19)
summary(aov(L2))
TUKEY_carb=TukeyHSD(aov(L2))
plot(TUKEY_carb)

gear=as.factor(B$gear)
L3=lm(Y ~ gear)
boxplot(Y ~ gear, main = "Bo tes    moustache de la variable gear", xlab = "gear", ylab = "Y", col = c("lightpink", "lightblue","lightgreen"), names = levels(gear)) #bo te    moustache de gear
moyennes = tapply(Y, gear, mean)
points(1:length(moyennes), moyennes, col = "red", pch = 19)
summary(aov(L3))
TUKEY_gear=TukeyHSD(aov(L3))
plot(TUKEY_gear)

cyl=as.factor(B$cyl)
L4=lm(Y ~ cyl)
boxplot(Y ~ cyl, main = "Bo tes    moustache de la variable cyl", xlab = "cyl", ylab = "Y", col = c("lightpink", "lightblue","lightgreen"), names = levels(cyl)) #bo te    moustache de cyl
moyennes = tapply(Y, cyl, mean)
points(1:length(moyennes), moyennes, col = "red", pch = 19)
summary(aov(L4))
TUKEY_cyl=TukeyHSD(aov(L4))
plot(TUKEY_cyl)

#Graphe pour comparer R_carre et Ra_carre
library(leaps)
R_carre = 1-((sum((Y_chapeau - Y)^2) )/(sum((Y - Yn_barre)^2)))
Ra_carre=1-((n-1)*(1-R_carre)/(n-rang))
reg = regsubsets(X1, Y, nvmax = 7, method = "exhaustive") #fonction de leaps pour effectuer une s lection de variables
summary = summary(reg)
```

```r
nb = which.max(summary$adjr2) #nombre de variables    s lectionner pour avoir le meilleur Ra_carre
var = summary$which[nb,-1] #summary$which est une matrice bool enne indiquant quelles variables sont incluses dans chaque mod le
#var r cup re la ligne de la matrice summary$which qui correspond aux variables    s lectionner pour avoir le meilleur Ra_carre et on enl ve la colonne Intercept
print(var) #3 variables carb, disp et gear
plot(summary$rsq, type = "l", col = "blue", pch = 16, ylim = c(0.6, 0.85), xlab = "Nombre de variables", ylab = "Valeurs",main = "Comparaison de R  et R ajust ") #graphe avec trac  de R_carre
lines(summary$adjr2, type = "l", col = "red", pch = 16) #on trace Ra_carre
legend("bottomright", legend = c("R  ", "R  ajust "),col = c("blue", "red"), lty=1, pch = 16, bty = "n")
```