Rapport : Régression linéaire

Compte-rendu du projet d'étude

Le modèle de regression linéaire

MAM 3
Année 2024 - 2025

Polytech Nice Sophia

Rédigé par :
Katell Nio et Charlotte Prouzet

Sommaire

1 Introduction et objectifs

La régression linéaire est une méthode statistique utilisée pour modéliser la relation entre une variable dépendante et une ou plusieurs variables indépendantes (variables explicatives). L'objectif de cette méthode est de construire un modèle capable de réaliser des prédictions à partir de données observées.

D'une part, lorsqu'une seule variable explicative est considérée, on parle de régression linéaire simple. Cela revient à ajuster une droite "au plus près" des points d'un nuage de données afin de représenter au mieux la tendance générale. Cette droite d'ajustement permet d'estimer la valeur de la variable dépendante à partir d'une valeur donnée de la variable explicative. Nous pourrons alors déterminer l'influence d'une variable indépendante sur une variable dépendante. D'autre part, en présence de plusieurs variables explicatives, on parle de régression linéaire multiple. Le principe reste le même, mais l'ajustement est réalisé dans un espace de dimension supérieure. L'objectif reste de modéliser la relation linéaire entre les variables, afin d'expliquer ou de prévoir le comportement de la variable cible. Nous pourrons alors analyser l'influence de plusieurs variables indépendantes sur une variable dépendante.

L'objectif de ce projet d'étude est de comprendre l'utilisation de la régression linéaire pour modéliser et prédire des variables. Nous étudierons la régression simple et multiple, puis nous nous pencherons sur l'évaluation de la qualité de nos modèles via des indicateurs comme le coefficient R^2 ajusté, ou encore via l'analyse des résidus. Nous compléterons nos prédictions en étudiant les intervalles de confiance. Enfin, nous procéderons à la sélection de variables afin d'optimiser l'ajustement de notre modèle.

2 Création du jeu de données

La première étape de ce projet a été de créer un jeu de données réelles. Nous avons utilisé le jeu de données nommé "mtcars" disponible dans R en le constituant de la manière suivante :

```
1  A = mtcars
2  F1 = as.factor(A[,8])
3  A[,8] = F1
4  set.seed(17)
5  set.seed(17 * floor(100 * runif(1, 0, 3)))
6  set1 = sample(1:32, 1)
7  B = A[-set1, ]
8  Y = B[,1]
9  u = 1:11
10 v = u[-c(1,8,9)]
11 set2 = c(8, sample(v, 6, replace = FALSE))
12 X = B[,set2]
```

Voilà une interprétation de chacune de ces lignes :

- (Ligne 1) On crée une variable `A` dans laquelle on charge le jeu de données `mtcars`.
- (Ligne 2) On transforme la $8^{\text{ème}}$ colonne de `A` en facteur. Cela permet au modèle de traiter correctement la variable `vs`, qui est qualitative.
- (Ligne 3) On remplace la $8^{\text{ème}}$ colonne de `A` par sa version factorisée, pour garantir que l'analyse prendra bien en compte son type catégoriel.
- (Ligne 4) On fixe la graine aléatoire à 17. Cela permet de reproduire les mêmes tirages aléatoires et donc de travailler à partir du même jeu de données.
- (Ligne 5) On génère un nombre aléatoire entre 0 et 3, que l'on multiplie par 100 et que l'on arrondit à l'entier inférieur. Ce résultat est ensuite multiplié par 17 pour fixer une nouvelle graine, introduisant un tirage légèrement variable.
- (Ligne 6) On tire au hasard un entier entre 1 et 32 que l'on stocke dans la variable `set1`.

- (Ligne 7) On crée un nouveau jeu de données `B` identique à `A`, mais sans la ligne d'indice `set1`. Ainsi, `B` contient 31 lignes contre 32 pour `A`.
- (Ligne 8) On définit la variable réponse `Y` comme étant la première colonne de `B`, soit la variable `mpg`.
- (Ligne 9) On crée le vecteur `u`, contenant les indices de 1 à 11, correspondant aux colonnes du jeu de données.
- (Ligne 10) On crée le vecteur `v`, obtenu en retirant de `u` les indices 1, 8 et 9. Ces colonnes sont exclues car :
  - la colonne 1 (`mpg`) est déjà stockée dans `Y`,
  - les colonnes 8 et 9 sont qualitatives (respectivement `vs` et `am`).
- (Ligne 11) On forme l'ensemble `set2`, qui contient l'indice 8 (on impose la présence de la variable qualitative `vs`), ainsi que 6 indices tirés aléatoirement dans `v`. `set2` contient donc 7 variables explicatives.
- (Ligne 12) Enfin, on extrait de `B` les colonnes désignées par `set2` pour construire la matrice `X`, qui servira dans la suite du projet.

Une fois toutes ces modifications apportées, notre jeu de données est alors composé de 31 lignes et 7 colonnes. Chaque ligne correspond à un modèle de voiture, tandis que chaque colonne correspond à une variable technique. Dans notre cas, les colonnes sont les suivantes :

Variable	`vs`	`disp`	`qsec`	`wt`	`hp`	`drat`	`carb`
Nom complet	Type de moteur	Cylindrée du moteur	Temps pour 1/4 mile	Poids du véhicule	Puissance moteur	Rapport de transmission arrière	Nombre de carburateurs
Unité	–	pouces cubes	secondes	1000 lbs	chevaux-vapeur	–	–
Type	Qualitative (facteur)	Quantitative continue	Quantitative continue	Quantitative continue	Quantitative continue	Quantitative continue	Quantitative discrète

3 Régression linéaire simple

3.1 Objectif de la méthode

La régression linéaire simple est une méthode statistique permettant de modéliser la relation entre deux variables :

- une **variable indépendante** (ou explicative),
- et une **variable dépendante** (ou à expliquer).

L'objectif est de déterminer une droite d'ajustement qui représente au mieux la tendance générale des données. Cette droite permet d'estimer la valeur de la variable dépendante en fonction d'une valeur donnée de la variable explicative.

Mathématiquement, le modèle s'écrit :

$$Y = aX + b + \varepsilon$$

où :

- Y est la variable à prédire,
- X est la variable explicative,
- a est le coefficient directeur (la pente),

- b est l'ordonnée à l'origine (interception),
- ε représente le bruit.

Notre droite de régression linéaire est déterminée selon la méthode des moindres carrés, c'est-à-dire en minimisant la somme des carrés des écarts entre les points observés et les points prédits.

Démonstration des expressions de $\hat{a}_n$ et $\hat{b}_n$:

Le cadre mathématique du modèle de régression linéaire est le suivant :
Pour tout i compris entre 1 et n :

$$\hat{Y}_i = \hat{a}_n x_i + \hat{b}_n + \varepsilon_i \quad \Leftrightarrow \quad \varepsilon_i = \hat{Y}_i - \hat{a}_n x_i - \hat{b}_n$$

On note S la somme des carrés des bruits ε_i, que l'on cherche à minimiser.

$$S = \sum_{i=1}^{n} (\varepsilon_i)^2 \Leftrightarrow S(a,b) = \sum_{i=1}^{n} (\hat{Y}_i - \hat{a}_n x_i - \hat{b}_n)^2$$

Minimiser $S(a,b)$ revient à annuler les dérivées partielles par rapport à a et par rapport à b :

$$\frac{\partial S}{\partial a} = -2 \sum_{i=1}^{n} x_i (Y_i - a x_i - b) = 0$$
$$\frac{\partial S}{\partial b} = -2 \sum_{i=1}^{n} (Y_i - a x_i - b) = 0$$

On obtient alors le système suivant :

$$\sum_{i=1}^{n} (x_i Y_i - a x_i^2 - b x_i) = 0$$
$$\sum_{i=1}^{n} (Y_i - a x_i - b) = 0$$

qui est équivalent à :

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

En posant :

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

On peut alors réécrire l'équation (2) comme :

$$\bar{Y}_n = a \bar{x}_n + b \Rightarrow \boxed{b = \bar{Y}_n - a \bar{x}_n}$$

Ainsi, en remplaçant b dans l'équation (1), on trouve :

$$\sum x_i Y_i - a\sum x_i^2 - (\bar{Y}_n - a\bar{x}_n)\sum_{i=1}^{n} x_i = 0$$
$$\sum x_i Y_i - a\sum x_i^2 - \bar{Y}_n \sum_{i=1}^{n} x_i + a\bar{x}_n \sum_{i=1}^{n} x_i = 0$$

Sachant que $\sum_{i=1}^{n} x_i = n\bar{x}_n$, on a donc :

$$\sum x_i Y_i - a\sum x_i^2 - \bar{Y}_n n\bar{x}_n + a\bar{x}_n n\bar{x}_n = 0$$
$$\sum x_i Y_i - \bar{Y}_n n\bar{x}_n = a\left(\sum x_i^2 - n\bar{x}_n^2\right)$$

Finalement, on obtient donc les coefficients suivants :

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

Remarque : On a $\sum_{i=1}^{n} x_i^2 - n\bar{x}_n^2 = \sum_{i=1}^{n}(x_i - \bar{x}_n)^2$.

Finalement, la solution du problème des moindres carrés est la suivante :

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

où :

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

Ce modèle est utilisé lorsque l'on suppose l'existence d'une relation linéaire entre les deux variables. Ainsi, pour pouvoir tracer notre droite de régression, nous devons au préalable vérifier la validité de notre modèle de régression linéaire simple. Pour ce faire, nous allons procéder à deux méthodes de vérification : la première via l'étude du coefficient R^2, et la seconde via l'étude des tests sur le paramètre a.

Remarque - démonstration du caractère sans biais de $\hat{a}_n$ et $\hat{b}_n$:

- Caractère sans biais de $\hat{a}_n$:

  On considère le modèle linéaire suivant :

  $$Y_i = ax_i + b + \varepsilon_i \quad \text{où } \mathbb{E}[\varepsilon_i] = 0 \text{ et } x_i \text{ sont fixés (non aléatoires).}$$

  $$x_i Y_i = x_i(ax_i + b + \varepsilon_i) = ax_i^2 + bx_i + x_i\varepsilon_i$$

  d'où :

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

De même, on a :

$$\bar{Y}_n = \frac{1}{n}\sum Y_i = a\bar{x}_n + b + \bar{\varepsilon}_n \quad \text{où } \bar{\varepsilon}_n = \frac{1}{n}\sum \varepsilon_i$$

donc :

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

On en déduit que le numérateur de $\hat{a}_n$ vaut :

$$\begin{aligned}\sum x_iY_i - n\bar{x}_n\bar{Y}_n &= \left(a\sum x_i^2 + b\sum x_i + \sum x_i\varepsilon_i\right) - \left(an\bar{x}_n^2 + bn\bar{x}_n + n\bar{x}_n\bar{\varepsilon}_n\right) \\ &= a\left(\sum x_i^2 - n\bar{x}_n^2\right) + b\left(\sum x_i - n\bar{x}_n\right) + \left(\sum x_i\varepsilon_i - n\bar{x}_n\bar{\varepsilon}_n\right)\end{aligned}$$

Or, $\sum x_i = n\bar{x}_n$, donc :

$$\sum x_iY_i - n\bar{x}_n\bar{Y}_n = a\sum (x_i - \bar{x}_n)^2 + \sum x_i\varepsilon_i - n\bar{x}_n\bar{\varepsilon}_n$$

Donc :

$$\begin{aligned}\mathbb{E}[\hat{a}_n] &= \frac{\mathbb{E}\left[a\sum (x_i - \bar{x}_n)^2 + \sum x_i\varepsilon_i - n\bar{x}_n\bar{\varepsilon}_n\right]}{\sum (x_i - \bar{x}_n)^2} \\ &= \frac{a\sum (x_i - \bar{x}_n)^2 + \mathbb{E}\left[\sum x_i\varepsilon_i\right] - \mathbb{E}\left[n\bar{x}_n\bar{\varepsilon}_n\right]}{\sum (x_i - \bar{x}_n)^2}\end{aligned}$$

Or :

$$\mathbb{E}\left[\sum x_i\varepsilon_i\right] = \sum x_i\mathbb{E}[\varepsilon_i] = 0, \quad \text{et } \mathbb{E}\left[\bar{\varepsilon}_n\right] = 0 \Rightarrow \mathbb{E}[n\bar{x}_n\bar{\varepsilon}_n] = 0$$

Donc :

$$\mathbb{E}[\hat{a}_n] = \frac{a\sum (x_i - \bar{x}_n)^2}{\sum (x_i - \bar{x}_n)^2} = a \quad \Rightarrow \quad \hat{a}_n \text{ est bien un estimateur sans biais de } a.$$

- Caractère sans biais de $\hat{b}_n$:

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

Or :

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

Donc :

$$\mathbb{E}[\hat{b}_n] = (a\bar{x}_n + b) - \bar{x}_n a = b \quad \Rightarrow \quad \hat{b}_n \text{ est bien un estimateur sans biais de } b.$$

3.2 Validité de la régression linéaire

3.2.1 Le coefficient R^2

Tout d'abord, nous avons décidé d'étudier le coefficient R^2 afin d'avoir une première idée de la validité (ou non) de notre modèle. En effet, plus notre coefficient R^2 est proche de 1, et plus la régression linéaire est une bonne modélisation. À savoir que l'on a toujours la relation suivante :

$$0 \leq R^2 \leq 1$$

En temps normal, on a :

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{Y}_i - Y_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y}_n)^2}$$

où :

- Y_i : la valeur observée de la variable dépendante pour l'observation i,
- $\hat{Y}_i$: la valeur prédite par le modèle pour l'observation i, soit $\hat{Y}_i = \hat{a}_n x_i + \hat{b}_n$,
- $\bar{Y_n}$: la moyenne des valeurs observées, soit $\bar{Y_n} = \frac{1}{n}\sum_{i=1}^{n} Y_i$,
- n : le nombre total d'observations.

Cependant, nous voulons ici tester la validité de notre modèle de régression linéaire avant de procéder aux calculs des estimateurs $\hat{a}_n$ et $\hat{b}_n$. Nous n'allons donc pas utiliser la formule classique de R^2, puisqu'elle requiert la connaissance de $\hat{Y}_i$, qui dépend directement de ces estimateurs.
À la place, nous allons déterminer R^2 en utilisant la formule du coefficient de corrélation de Pearson :

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

En régression linéaire simple, le coefficient de détermination R^2 est simplement le carré de r :

$$R^2 = r^2$$

où :

- x_i, y_i : les observations des variables X et Y,
- $\bar{x}, \bar{y}$: les moyennes respectives des séries x_i et y_i,
- n : le nombre total d'observations.

En R, le coefficient de corrélation de Pearson est prédéfini via la fonction `cor`. Afin d'obtenir le coefficient de détermination R^2 pour plusieurs variables quantitatives en fonction d'une variable Y, nous avons écrit la fonction suivante :

```
deter_r=function(){
  n=length(X)
  R=rep(c(0),times=n)
  for (i in 2:n){
    x=X[,i]
    R[i]=cor(x,Y)
  }
  R=R^2
  R
}
```

Listing 1: Fonction pour calculer R^2 à partir de Pearson

En testant cette fonction sur toutes les colonnes de X, nous obtenons les résultats :

Indice	1	2	3	4	5	6	7
R^2	0.0000000	0.7173685	0.1943166	0.7562557	0.6003721	0.4604115	0.2998048

Table 1: Valeurs du coefficient de détermination R^2 pour chaque variable (indices 1 à 7)

On observe donc que le "meilleur" R^2, soit celui qui est le plus proche de 1, est obtenu avec la 4ème colonne de X, tandis que le "moins bon" R^2 est obtenu avec la 3ème colonne de X.

3.2.2 Test du paramètre a

Une autre manière pour étudier la validité du modèle de régression linéaire simple consiste à effectuer des tests sur le paramètre a. Cependant, lorsque l'on parle de test, la notion de loi est nécessaire. Il faut donc vérifier l'hypothèse de gaussianité du bruit. Nous allons donc procéder à deux vérifications : l'une sur les résidus standardisés, et la seconde sur les résidus Studentisés, puisque si le bruit est gaussien, alors les résidus le sont aussi !

a) Résidus Standardisés

Les résidus standardisés sont tous de même loi, mais avec une loi connue asymptotiquement. Ainsi, une manière de valider l'hypothèse de gaussianité du bruit consiste à vérifier si les résidus standardisés suivent approximativement une loi normale standard.
Pour ce faire, nous avons utilisé la fonction lm(), qui est justement utilisée en R pour ajuster des modèles de régression linéaire. Nous utilisons alors le code suivant :

```
1 m = lm(Y ~ X[,4])
2 rst1 = rstandard(m)
3 qqnorm(rst1, main="Q-Q plot des residus standardises")
4 qqline(rst1, col="red", lwd=2)
```

Listing 2: Q-Q plot des résidus standardisés (pour la 4ème colonne de X)

Explication du code :

- (Ligne 1) On crée un modèle de régression linéaire simple dans lequel la variable explicative est la 4ème colonne de X (correspond à la colonne pour laquelle on a le "meilleur" R^2), et la variable à expliquer est Y.
- (Ligne 2) On calcule les résidus standardisés de notre modèle m.
- (Ligne 3) On trace un Q-Q plot des résidus standardisés. On peut alors vérifier visuellement si ces résidus suivent bien une loi Normale Standard ou non.
- (Ligne 4) On ajoute une droite de référence. Si les points tracés précédemment suivent cette droite, alors les résidus suivront bien approximativement une loi Normale Standard.

Nous avons donc obtenu les 3 graphiques suivants pour le choix de différentes colonnes de X, à savoir la 3ème, la 4ème et la 6ème qui correspondent respectivement à celles pour lesquelles nous avons obtenu le "meilleur" R^2, le "moins bon" R^2, et un R^2 intermédiaire.

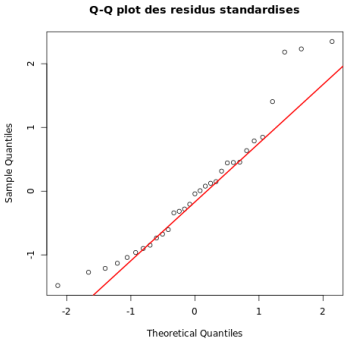

Figure 1: Q-Q plot des résidus standardisés pour X[,4]

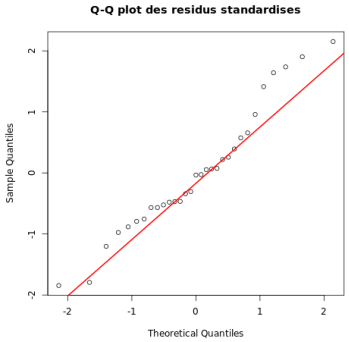

Figure 2: Q-Q plot des résidus standardisés pour X[,3]

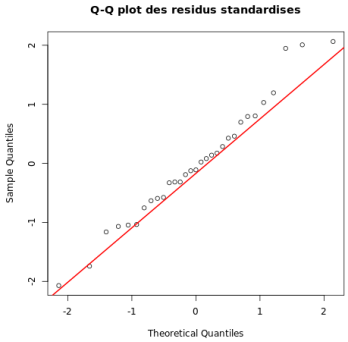

Figure 3: Q-Q plot des résidus standardisés pour X[,6]

Nous observons que pour les 3 colonnes sélectionnées, peu importe le R^2 auquel ils sont associés, les résidus standardisés suivent bien la droite de référence. Nous en déduisons donc que les résidus standardisés suivent une loi Normale Standard, et donc que l'hypothèse de gaussianité du bruit est valide.

b) Résidus Studentisés

Vérifier l'hypothèse de gaussianité du bruit pour les résidus Studentisés revient à montrer que ces derniers suivent une loi de Student à (n-3) degré de liberté. C'est donc ce que nous avons mis en œuvre en effectuant un test d'adéquation de Kolmogorov.

Nous avons donc tracé nos résidus Studentisés ainsi que la courbe caractéristique d'une loi de Student à (n-3) degré de liberté. Le code utilisé est le suivant :

```
1  m = lm(formula = Y ~ X[,4])
2  rst4 = rstudent(m)
3  srst4 = sort(rst4)
4  nt4 = length(rst4)
5  yst4 = 1 / nt4 * (1:nt4)
6
7  xt = seq(srst4[1], srst4[nt4], 0.01)
8  yt = pt(xt, length(Y) - 3)
9
10 plot(srst4, yst4,
11      main = "Residus Studentises",
12      type = 's',
13      col = 'blue',
14      xlim = c(srst4[1], srst4[nt4]),
15      ylim = c(0, 1))
16
17 lines(xt, yt, col = 'red')
18
19 ks.test(rst4, "pt", length(Y) - 3)
```

Listing 3: Comparaison des résidus Studentisés à une loi de Student (pour la 4ème colonne de X)

Explication du code :

- (Ligne 1) On crée m le modèle de régression linéaire simple avec Y pour variable dépendante et la 4ème colonne de X pour variable explicative.
- (Ligne 2) On calcule les résidus Studentisés de notre modèle.
- (Ligne 3) On trie les résidus Studentisés dans l'ordre croissant pour ensuite pouvoir les comparer à une loi théorique.
- (Ligne 4) On pose nt4 le nombre total d'observations.
- (Ligne 5) On calcule les valeurs empiriques de la fonction de répartition des résidus. Ce résultat sera utilisé pour construire la fonction de répartition empirique.
- (Ligne 7) On pose "xt" une séquence de valeurs allant du minimum au maximum des résidus.
- (Ligne 8) On calcule la fonction de répartition théorique de la loi de Student à (n-3) degré de liberté.
- (Ligne 10) On trace la fonction de répartition empirique des résidus
- (Ligne 17) On trace la courbe théorique de la loi de Student
- (Ligne 19) On effectue un test de Kolmogorov ce qui nous permet d'obtenir la p-valeur

Nous avons donc obtenu les 3 graphiques suivants pour le choix de différentes colonnes de X, à savoir la 3ème, la 4ème et la 6ème qui correspondent respectivement à celles pour lesquelles nous avons obtenu le "meilleur" R^2, le "moins bon" R^2, et un R^2 intermédiaire.

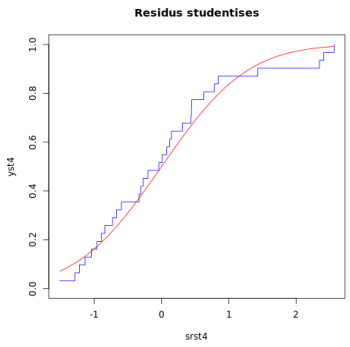

Figure 4: Plot des résidus Studentisés pour X[,4]

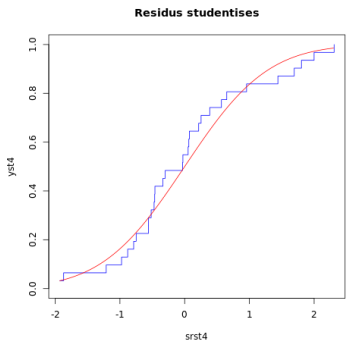

Figure 5: Plot des résidus Studentisés pour X[,3]

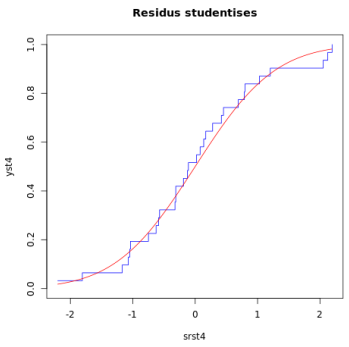

Figure 6: Plot des résidus Studentisés pour X[,6]

Indice	X[,4]	X[,3]	X[,6]
p-valeur	0.8667	0.7544	0.9835

Table 2: p-valeur pour chaque variable (colonnes 4, 3 et 6 de X)

Nous observons que pour les 3 colonnes sélectionnées, peu importe le R^2 auquel ils sont associés, l'allure de la fonction de répartition des résidus Studentisés suit bien la courbe de référence. Par ailleurs, les p-valeurs obtenues sont toutes supérieures à α=5%. Nous en déduisons donc que les résidus Studentisés suivent bien la loi de Student à (n-3) degré de liberté, et donc que l'hypothèse de gaussianité du bruit est valide.

Remarque :
les résidus pourraient-ils suivre une autre loi, comme la loi Uniforme par exemple ?
Afin de répondre à cette interrogation, nous avons rédigé le code suivant qui compare la fonction de répartition des résidus à la courbe théorique de la loi Uniforme.

```
m = lm(formula = Y ~ X[,4])
rst4 = rstudent(m)
srst4 = sort(rst4)
nt4 = length(rst4)
yst4 = 1 / nt4 * (1:nt4)

xt = seq(min(srst4), max(srst4), length.out = 1000)
yt_unif = punif(xt, min = min(rst4), max = max(rst4))

plot(srst4, yst4, type = "s", col = "blue", ylim = c(0, 1),
     main = "Residus Studentises vs Loi Uniforme",
     xlab = "Residus",
     ylab = "Fonction de repartition")

lines(xt, yt_unif, col = "green", lwd = 2)

ks.test(rst4, "punif", min = min(rst4), max = max(rst4))
```

Listing 4: Comparaison des résidus Studentisés à une loi Uniforme (pour la 4^e colonne de X)

Nous avons donc obtenu les 3 graphiques suivants pour le choix de différentes colonnes de X, à savoir la 3ème, la 4ème et la 6ème qui correspondent respectivement à celles pour lesquelles nous avons obtenu le "meilleur" R^2, le "moins bon" R^2, et un R^2 intermédiaire.

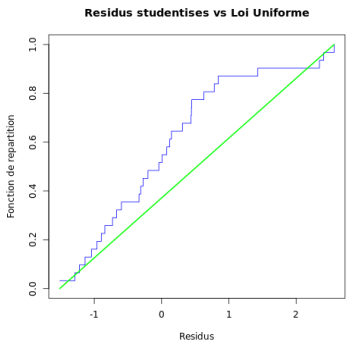

Figure 7: Résidus Studentisés VS loi Uniforme pour X[,4]

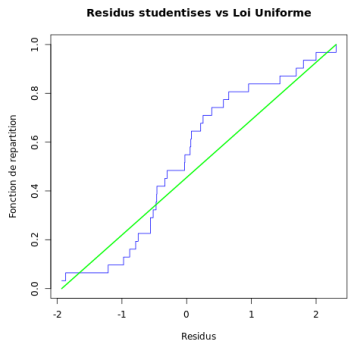

Figure 8: Résidus Studentisés VS loi Uniforme pour X[,3]

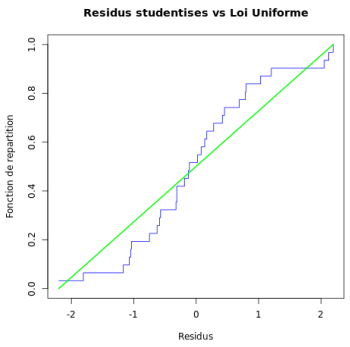

Figure 9: Résidus Studentisés VS loi Uniforme pour X[,6]

Indice	X[,4]	X[,3]	X[,6]
p-valeur	0.00743	0.1547	0.2937

Table 3: p-valeurs pour chaque variable (colonnes 4, 3 et 6 de X)

Au vue de ces graphiques, on remarque bien que la fonction de répartition des résidus Studentisés ne suit pas la courbe théorique d'une loi Uniforme. Par ailleurs, la p-valeur obtenue pour X[,4] (qui est la colonne pour laquelle on a le "meilleur" R^2) est inférieure à α=5% ce qui montre bien statistiquement que les résidus suivent une loi de Student et non une loi Uniforme. En effet, dans le cas contraire, on ne pourrait pas effectuer de test sur le paramètre a.

Ayant donc validé l'hypothèse de gaussianité du bruit pour les résidus standardisés et Studentisés, et ayant des p-valeurs supérieures à α=5%, nous pouvons à présent effectuer des tests sur le paramètre a.

c) Test sur le paramètre a

Dans le cadre d'un modèle de régression linéaire simple :

$$Y_i = ax_i + b + \varepsilon_i \quad \text{avec} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Nous souhaitons tester si la variable explicative X a un effet significatif sur Y, autrement dit, si le coefficient a est différent de 0.

Hypothèses du test :

- H_0 : $a = 0$ (pas d'effet de X sur Y)
- H_1 : $a \neq 0$

On utilise la statistique :

$$T_a = \frac{\widehat{a}_n}{\widehat{\sigma}_n \cdot \sqrt{\dfrac{1}{\sum (x_i - \bar{x}_n)^2}}}$$

où :

- $\widehat{a}_n$ est l'estimateur du coefficient a,
- $\widehat{\sigma}^2 = \dfrac{1}{n-2} \sum (Y_i - \widehat{Y}_i)^2$ est l'estimateur de la variance résiduelle.

Sous H_0, la statistique T_a suit une loi de Student à $n-2$ degrés de liberté :

$$T_a \sim \mathcal{T}(n-2)$$

Proposition :
On rejette H_0 au seuil α si :

$$|T_a| > t_{1-\alpha/2,\, n-2}$$

ou, équivalemment, si la p-valeur est inférieure à α. Tout au long de ce projet, nous considérons α=5%
Le code correspondant à la prise de décision ci-dessus est le suivant :

D'une part, on pose les coefficients estimés $\widehat{a}_n$ et $\widehat{b}_n$ selon la méthode des moindres carrés :

```
an_chap = (sum(X[,4]*Y) - 31*mean(X[,4])*mean(Y)) / sum((X[,4] - mean(X[,4]))^2)
bn_chap = mean(Y) - an_chap * mean(X[,4])
```

D'autre part, on calcule la statistique de test T_a pour tester l'hypothèse $H_0 : a = 0$ dans notre régression linéaire simple, et ce, en utilisant le code suivant :

```
n = length(X[,4])
Y_chap = an_chap * X[,4] + bn_chap
on2_chap = 1/(n - 2) * sum((Y - Y_chap)^2)
on_chap = sqrt(on2_chap)

Ta = an_chap / (on_chap * sqrt(1 / sum((X[,4] - mean(X[,4]))^2)))
Ta = abs(Ta)
PT = pt(Ta, df = n - 2)
p_val = 1 - PT
```

Explication du code :

- (Ligne 1) On détermine la taille de notre échantillon, soit le nombre d'observations.
- (Ligne 2) On calcule $\widehat{Y}_i$ telle que : $\widehat{Y}_i = \widehat{a}_n x_i + \widehat{b}_n$
- (Ligne 3) On calcule l'estimation de la variance du bruit, que l'on note $\widehat{\sigma}^2$ telle que : $\widehat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$
- (Ligne 4) On en déduit l'estimateur de l'écart-type
- (Ligne 6) On calcule la statistique de test T_a telle que :

$$T_a = \frac{|\widehat{a}_n|}{\widehat{\sigma} \cdot \sqrt{\dfrac{1}{\sum(X_i - \bar{X})^2}}}$$

- (Ligne 7) On garde la valeur absolue de T_a
- (Ligne 8) On calcule la p-valeur associée grâce à la fonction prédéfinie "pt".

Le code nous renvoie les p-valeur suivantes :

Indice	X[,4]	X[,3]	X[,6]
p-valeur	1.07838e-10	0.006530888	1.361856e-05

Table 4: p-valeurs pour chaque variable (colonnes 4, 3 et 6 de X)

On observe que dans les 3 cas, la p-valeur obtenue est bien inférieure à α=5%. On a donc p-valeur $< \alpha$.
Cela implique la décision suivante :
On décide $H_1 : a \neq 0$

3.3 Affichage graphique de notre modèle de régression linéaire simple

Dans les parties précédentes, nous avons montré de deux manières différentes la validité de notre modèle de régression linéaire : en utilisant le coefficient de détermination R^2, et un test sur le paramètre a. Nous allons donc à présent pouvoir tracer notre nuage de points, puis le comparer à la droite linéaire associée, à savoir : $\hat{Y}_i = \hat{a}_n x_i + \hat{b}_n$

Nous rédigeons alors le code suivant :

```
1  an_chap = (sum(X[,4]*Y) - 31*mean(X[,4])*mean(Y)) / sum((X[,4] - mean(X[,4]))^2)
2  bn_chap = mean(Y) - an_chap*mean(X[,4])
3
4  plot(X[,4], Y,
5       main = "Regression lineaire simple",
6       xlab = "Variable explicative : wt",
7       ylab = "Variable reponse",
8       pch = 19, col = "blue")
9
10 abline(a = bn_chap, b = an_chap, col = "red", lwd = 2)
11
12 legend("topright",
13        legend = c("Donnees", "Droite de regression"),
14        col = c("blue", "red"),
15        pch = c(19, NA),
16        lty = c(NA, 1),
17        lwd = c(NA, 2))
```

Listing 5: Plot de notre modèle de régression linéaire simple

Explication du code :

- (Ligne 1) On définit le coefficient directeur $\widehat{a}_n$ de la droite de régression par la formule des moindres carrés.
- (Ligne 2) On définit l'ordonnée à l'origine $\widehat{b}_n$ de la droite de régression par la formule des moindres carrés.
- (Ligne 4) On affiche nos données sous la forme d'un nuage de points.
- (Ligne 10) On trace notre droite de régression.
- (Ligne 12) Ajout de la légende.

Ainsi, nous obtenons les graphes suivants :

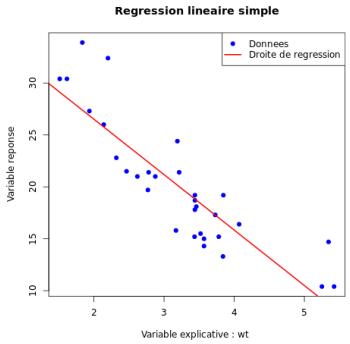

Figure 10: Modèle de régression linéaire simple pour X[,4]

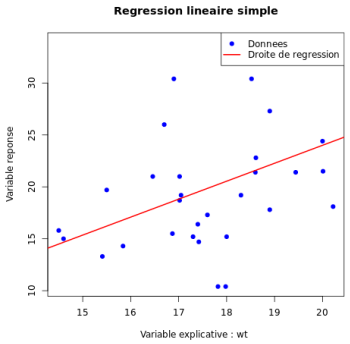

Figure 11: Modèle de régression linéaire simple pour X[,3]

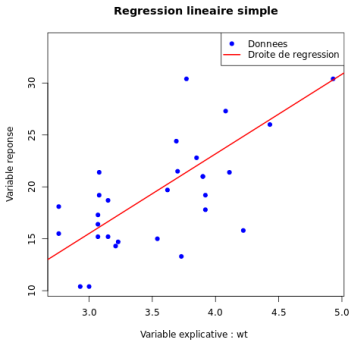

Figure 12: Modèle de régression linéaire simple pour X[,6]

On remarque bien que le nuage de points est proche de la droite de régression linéaire dans le cas où l'on prend la 4ème colonne de X, soit lorsque l'on obtient le R^2 le plus proche de 1. Il y a donc bel et bien un lien entre la valeur de R^2 et la proximité des points par rapport à la droite de régression linéaire tracée.

3.4 Vérification

Afin de vérifier l'exactitude de nos représentations graphiques, nous avons décidé d'utiliser la fonction "lm()" prédéfinie dans R.
Nous allons ainsi comparer un graphe pour lequel on a défini $\widehat{a}_n$ et $\widehat{b}_n$ avec la formule des moindres carrés, et un graphe avec lequel on a simplement utilisé la fonction "lm()".

Le code du premier graphe est donné par le code ci-dessus.
le code du second graphe qui utilise la fonction "lm()" est le suivant :

```
1 m = lm(Y ~ X[,4])
2 plot(X[,4], Y, main = "Regression avec lm()", pch = 19, col = "blue")
3 abline(m, col = "red", lwd = 2)
```

Listing 6: Modèle de régression linéaire simple avec lm()

Explication du code :

- (Ligne 1) On crée un modèle de régression linéaire noté m dans lequel Y est la variable réponse, et X[,4] est la variable indépendante. L'object "m" contient alors les coefficients estimés, les résidus...
- (Ligne 2) On dessine le nuage de points
- (Ligne 3) On trace la droite de régression linéaire estimée à partir de "m"

Avec nos deux méthodes, nous obtenons alors les graphes suivants :

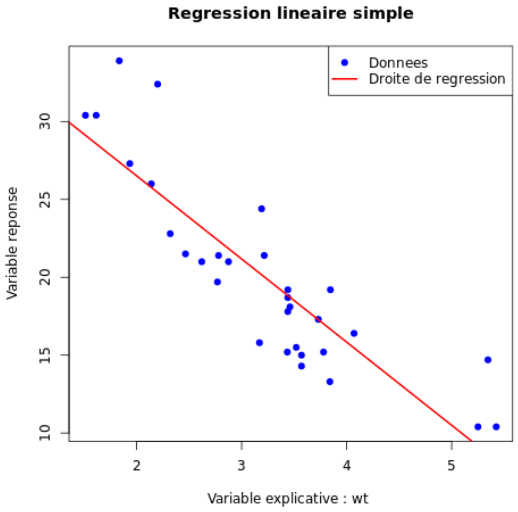

Figure 13: Modèle de régression linéaire simple en utilisant les coefficients $\widehat{a}_n$ et $\widehat{b}_n$ pour X[,4]

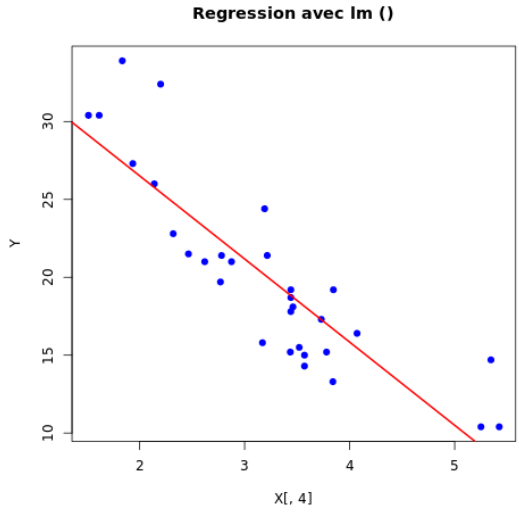

Figure 14: Modèle de régression linéaire simple en utilisant la fonction "lm()" pour X[,4]

On observe que les deux graphes sont identiques, ce qui confirme la validité de notre modèle obtenu avec les coefficients $\widehat{a}_n$ et $\widehat{b}_n$ (méthode des moindres carrés).

3.5 Prédiction

En régression linéaire simple, l'un des objectifs fondamentaux, en plus de comprendre la relation entre deux variables, est de prédire la valeur de notre variable dépendante Y, à partir d'une nouvelle valeur de la variable explicative X.
Ainsi, si l'on note x_{new} une nouvelle observation de notre variable explicative, une prévision de la variable réponse Y sera donnée par :

$$\widehat{y}_{\text{new}} = \widehat{a}_n x_{\text{new}} + \widehat{b}_n$$

Cependant, il est important de prendre en compte le fait que $\widehat{Y}_{\text{new}}$ est une estimation qui est sujette à plusieurs sources d'incertitudes, comme celle sur les coefficients $\widehat{a}_n$ et $\widehat{b}_n$ car ce sont des estimations

C'est pourquoi nous allons construire deux intervalles :

1. L'intervalle de confiance qui entoure la moyenne de Y pour un certain x_{new} :

$$\widehat{y}_{\text{new}} \pm \widehat{\sigma}_n \cdot \sqrt{\frac{1}{n} + \frac{(x_{\text{new}} - \bar{x}_n)^2}{\sum (x_i - \bar{x}_n)^2}} \cdot t_{1-\alpha/2,\, n-2}.$$

2. L'intervalle de prédiction qui entoure la future valeur d'un individu ayant x_{new} :

$$\widehat{y}_{\text{new}} \pm \widehat{\sigma}_n \cdot \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x}_n)^2}{\sum (x_i - \bar{x}_n)^2}} \cdot t_{1-\alpha/2,\, n-2}.$$

Afin de déterminer et de tracer les intervalles de confiance et de prédiction relatifs à notre modèle de régression linéaire simple, nous avons rédigé le code suivant :

```
1  df = data.frame(x = X[, 4], y = Y)
2  m = lm(y ~ x, data = df)
3
4  plot(df$x, df$y,
5       main = "Regression lineaire avec intervalle de confiance a 95%",
6       xlab = "Variable explicative", ylab = "Variable reponse")
7
8  abline(m, col = "red", lwd = 1)
9
10 x_seq = seq(min(df$x), max(df$x), length.out = 100)
11 x_new = data.frame(x = x_seq)
12
13 # Intervalle de confiance
14 pred = predict(m, newdata = x_new, interval = "confidence", level = 0.95)
15 lines(x_seq, pred[, "lwr"], col = "blue", lwd = 1)
16 lines(x_seq, pred[, "upr"], col = "blue", lwd = 1)
17
18 # Intervalle de prediction
19 pred = predict(m, newdata = x_new, interval = "prediction", level = 0.95)
20 lines(x_seq, pred[, "lwr"], col = "green", lwd = 1)
21 lines(x_seq, pred[, "upr"], col = "green", lwd = 1)
```

Listing 7: Prédictions avec intervalles de confiance et de prédiction

Explication du code :

- (Ligne 1) On crée le data frame "df" qui contient "x" la variable explicative (qui correspond à la 4ème colonne de X ici) et "y" la variable réponse.
- (Ligne 2) On ajuste le modèle de régression linéaire simple en utilisant la fonction lm().
- (Ligne 4) On affiche le nuage de points qui représente nos différentes observations.
- (Ligne 8) On ajoute la droite de régression linéaire en rouge.

- (Ligne 10) On crée une "grille" de 100 valeurs de x.
- (Ligne 11) On pose x_{new} notre nouveau jeu de données pour lequel on fait des prédictions.
- (Ligne 14) On calcule les prédictions pour la moyenne Y en utilisant la fonction "predict()" dans laquelle on renseigne bien "intervalle de confiance". On prend un intervalle de confiance à 95%.
- (Ligne 15) On trace en bleu l'incertitude inférieure.
- (Ligne 16) On trace en bleu l'incertitude supérieure.
- (Ligne 19) On calcule les prédictions pour la valeur individuelle de Y en utilisant la fonction "predict()" dans laquelle on renseigne bien "intervalle de prédiction". On prend un intervalle de prédiction à 95%.
- (Ligne 20) On trace en vert l'incertitude inférieure.
- (Ligne 21) On trace en vert l'incertitude supérieure.

Les graphes obtenus sont les suivants :

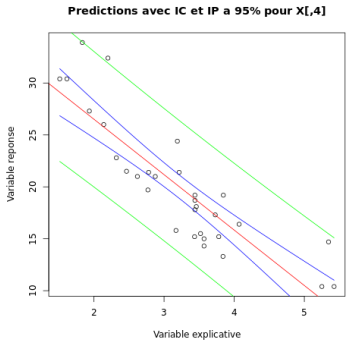

Figure 15: Prédictions avec intervalles de confiance et de prédiction à 95% pour X[,4]

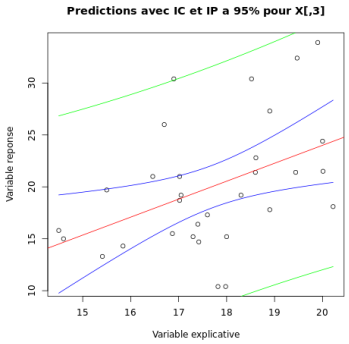

Figure 16: Prédictions avec intervalles de confiance et de prédiction à 95% pour X[,3]

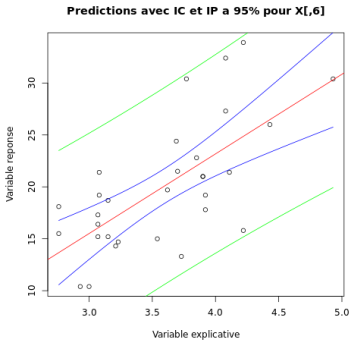

Figure 17: Prédictions avec intervalles de confiance et de prédiction à 95% pour X[,6]

Nous remarquons que les bandes vertes (intervalle de prédiction) sont plus larges que les bandes bleues (intervalle de confiance). Cela s'explique par le fait qu'elles prennent en compte à la fois l'incertitude du modèle, et celle d'une future observation.
De plus, les bandes de confiance sont plus étroites autour de la moyenne de x et s'élargissent aux extrémités. Nous pouvons expliquer cela par le fait que l'incertitude est plus grande pour des valeurs éloignées du centre des données, ce qui induit un élargissement des bandes de confiance aux bords.

3.6 Démonstration de formules

3.6.1 Démonstration des variances des estimateurs en régression linéaire simple

On note $\widehat{a}_n$ et $\widehat{b}_n$ les estimateurs des coefficients a et b, définis par la méthode des moindres carrés :

$$\widehat{a}_n = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \qquad \widehat{b}_n = \bar{Y} - \widehat{a}_n \bar{x},$$

où $\bar{x} = \frac{1}{n}\sum x_i$ et $\bar{Y} = \frac{1}{n}\sum Y_i$.

Variance de $\widehat{a}_n$

En remplaçant $Y_i = ax_i + b + \varepsilon_i$, on obtient :

$$\widehat{a}_n = \frac{\sum(x_i - \bar{x})(ax_i + b + \varepsilon_i - \bar{Y})}{\sum(x_i - \bar{x})^2} = a + \frac{\sum(x_i - \bar{x})\varepsilon_i}{\sum(x_i - \bar{x})^2}$$

En notant $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x}_n)^2$, on obtient :

$$\widehat{a}_n = a + \frac{\sum_{i=1}^{n}(x_i - \bar{x}_n)\varepsilon_i}{S_{xx}}$$

Comme $\mathbb{V}[a] = 0$, on a :

$$\mathbb{V}[\widehat{a}_n] = \mathbb{V}\left[\frac{\sum_{i=1}^{n}(x_i - \bar{x}_n)\varepsilon_i}{S_{xx}}\right] = \frac{1}{S_{xx}^2} \cdot \mathbb{V}\left[\sum_{i=1}^{n}(x_i - \bar{x}_n)\varepsilon_i\right]$$

Les ε_i sont indépendants, donc :

$$\mathbb{V}\left[\sum_{i=1}^{n}(x_i - \bar{x}_n)\varepsilon_i\right] = \sum_{i=1}^{n}(x_i - \bar{x}_n)^2 \cdot \mathbb{V}[\varepsilon_i] = \sigma^2 \cdot S_{xx}$$

Donc :

$$\mathbb{V}[\widehat{a}_n] = \frac{1}{S_{xx}^2} \cdot \sigma^2 S_{xx} = \frac{\sigma^2}{S_{xx}}$$

Finalement :

$$\mathbb{V}[\widehat{a}_n] = \mathbb{V}\left(\frac{\sum(x_i - \bar{x})\varepsilon_i}{\sum(x_i - \bar{x})^2}\right) = \frac{1}{\left(\sum(x_i - \bar{x})^2\right)^2}\sum(x_i - \bar{x})^2\mathbb{V}[\varepsilon_i] = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$$

Variance de $\widehat{b}_n$

On a :

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

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, on peut démontrer que dans le cadre de l'hypothèse de bruit centré :

$$\mathrm{Cov}(\bar{Y}_n, \widehat{a}_n) = 0$$

La moyenne empirique des observations $Y_1, \ldots, Y_n$ est donnée par :

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

où $\bar{\varepsilon}_n = \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i$ est la moyenne des bruits.

Comme les ε_i sont supposés indépendants et de même loi $\mathcal{N}(0, \sigma^2)$, 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 :

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

On utilise alors l'égalité suivante :

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

Finalement :

$$\mathbb{V}[\widehat{b}_n] = \sigma^2 \cdot \frac{\sum_{i=1}^{n} x_i^2}{n \cdot S_{xx}}$$

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

Conclusion : Les variances des estimateurs des coefficients dans le modèle linéaire simple sont donc données par :

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

3.6.2 Démonstration de la formule de l'intervalle de prédiction pour une nouvelle valeur x_{new}

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

$$Y_i = ax_i + b + \varepsilon_i \quad \text{avec } \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

On cherche à prédire une nouvelle valeur aléatoire :

$$Y_{\text{new}} = ax_{\text{new}} + b + \varepsilon_{\text{new}}$$

Estimateur de la prédiction

On utilise le modèle estimé pour prédire :

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

Variance de la prédiction

Comme Y_{new} est aléatoire, on calcule la variance de l'erreur de prédiction :

$$\mathbb{V}[Y_{\text{new}} - \widehat{Y}(x_{\text{new}})] = \mathbb{V}[\widehat{Y}(x_{\text{new}})] + \mathbb{V}[\varepsilon_{\text{new}}]$$

Or :

$$\mathbb{V}[\widehat{Y}(x_{\text{new}})] = \sigma^2 \cdot \left(\frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \right) \quad \text{et} \quad \mathbb{V}[\varepsilon_{\text{new}}] = \sigma^2$$

Donc :

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

Utilisation de la statistique de Student

On construit la statistique :

$$T = \frac{Y_{\text{new}} - \widehat{Y}(x_{\text{new}})}{\widehat{\sigma} \cdot \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum (x_i - \bar{x})^2}}} \sim \mathcal{T}(n-2)$$

Formule de l'intervalle de prédiction

On en déduit l'intervalle de niveau de confiance $1 - \alpha$:

$$IC_{1-\alpha}(x_{\text{new}}) = \left[\widehat{a}_n x_{\text{new}} + \widehat{b}_n \pm t_{1-\alpha/2,\, n-2} \cdot \widehat{\sigma} \cdot \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{\sum (x_i - \bar{x})^2}} \right]$$

4 Régression linéaire multiple

4.1 Objectif de la méthode

Après avoir étudié le cas de la régression linéaire simple, nous généralisons le modèle à plusieurs variables explicatives. La **régression linéaire multiple** est un outil statistique qui permet de modéliser la relation entre une variable à expliquer $Y \in \mathbb{R}^n$ et plusieurs variables explicatives regroupées dans une matrice $X \in \mathbb{R}^{n \times p}$, avec p le nombre de variables et n le nombre d'observations.

Le modèle s'écrit sous forme vectorielle :

$$Y = X\beta + \varepsilon$$

où :

- $Y \in \mathbb{R}^n$ est le vecteur des observations de la variable dépendante ;
- $X \in \mathbb{R}^{n \times p}$ est la matrice des observations des variables explicatives (chaque ligne correspond à une observation, chaque colonne à une variable) ;
- $\beta \in \mathbb{R}^p$ est le vecteur des coefficients inconnus à estimer ;
- $\varepsilon \in \mathbb{R}^n$ est le vecteur des erreurs aléatoires supposées centrées, non corrélées, et de variance constante.

L'objectif est d'estimer le vecteur β tel que la somme des carrés des résidus soit minimisée. La solution du problème des moindres carrés est donnée par :

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y$$

Une fois $\hat{\beta}$ obtenu, nous pouvons calculer les valeurs ajustées par le modèle :

$$\hat{Y} = X\hat{\beta}$$

Le modèle est considéré comme linéaire, car Y est une combinaison linéaire des colonnes de X, pondérées par les coefficients β_j. Ce modèle suppose donc l'existence d'une relation linéaire entre la variable réponse et les variables explicatives.

Comme dans le cas simple, il est nécessaire de vérifier la validité du modèle ajusté. Pour cela, nous analyserons d'abord le coefficient R^2 ajusté, puis nous vérifierons les hypothèses du modèle via l'analyse des résidus.

4.2 Validité de la régression linéaire

4.2.1 $X^\top X$ inversible

Contrairement à la régression linéaire simple, nous travaillons ici sur des matrices. Dans la formule de β chapeau, il est question d'inverser une matrice. Nous avons donc vérifié, dans un premier temps, que la matrice $X^\top X$ était inversible en contrôlant que :

- le déterminant était non nul ;
- le rang de la matrice était égal au nombre de colonnes.

Notre code :

```r
#avec X1 = X[,c(-1,-7)], sans le vs et carb (variables qualitatives)
X1 = X[,c(-1,-7)]
df1 = data.frame(X1)
X1mat = model.matrix(~ ., data = df1)

XtX1 = t(X1mat) %*% X1mat
det(XtX1)         # Doit etre different 0
rangX1 = qr(X1mat)$rank    # Doit etre egal a ncol(X1mat)
```

Listing 8: Vérification $X^\top X$ inversible (ex avec X1)

4.2.2 Validation par le coefficient $R^2_{\text{ajusté}}$

Pour évaluer la qualité de nos modèles, nous avons utilisé le coefficient de détermination $R^2_{\text{ajusté}}$ qui dépend de R^2, mais qui permet de prendre en compte le nombre de variables explicatives. Ce dernier permet d'éviter le surajustement en pénalisant les modèles trop complexes.

Nous avons adopté une **méthode pas à pas** pour sélectionner les variables les plus pertinentes :

- Nous sommes partis de la matrice X contenant les variables explicatives, après avoir retiré les variables qualitatives (`vs` et `carb`) ;
- Nous avons progressivement retiré les colonnes (variables quantitatives), une par une, en observant l'évolution du $R^2_{\text{ajusté}}$;
- À chaque étape, nous avons conservé la configuration donnant la valeur de $R^2_{\text{ajusté}}$ la plus élevée ;
- Le processus s'est arrêté lorsque le retrait d'une variable n'améliorait plus le score. Le modèle final X_4 contient alors uniquement les 3 variables les plus pertinentes.

Pour chaque modèle (X1, X2, X3), nous avons calculé les coefficients de détermination :

$$R^2 = 1 - \frac{\sum(Y-\hat{Y})^2}{\sum(Y-\bar{Y})^2} \qquad R^2_{\text{ajusté}} = 1 - \frac{(n-1)}{(n-p)}(1-R^2)$$

Nous avons d'abord enlevé les colonnes une par une en notant le meilleur $R^2_{\text{ajusté}}$, puis nous avons décidé de faire une fonction afin d'optimiser notre code :

```
find_Ra=function(X_initial){
  df=data.frame(X_initial) #creation d'un data frame
  Xmat=model.matrix(~ ., data=df) #creation de la matrice du modele
  XtX=t(Xmat) %*% Xmat #on calcule le produit matriciel
  rangX = qr(Xmat)$rank
  XtY=t(Xmat) %*% Y
  beta_chap=solve(XtX) %*% XtY #on calcule les coefficients estimes
  Y_chap=Xmat %*% beta_chap #on calcule les valeurs predites
  n=length(Y)
  R=1-sum((Y_chap - Y)^2) / sum((Y - mean(Y))^2) #coeff de determination R^2
  Ra=1-((n-1)/(n-rangX))*(1-R) #R^2 ajuste
  return (Ra)
}


selection = function(){
  X_initial = X[, -c(1, 7)] # on enleve les colonnes qualitatives
  Ra_initial = find_Ra(X_initial) #calcul du R^2 ajuste de depart
  long = ncol(X_initial) #nombre initial de variables
  cat("R^2 ajuste initial :", round(Ra_initial,5), "\n")
  while (long > 1) { #tant qu'il reste plus d'une variable a tester, on passe
      dans la boucle
    best_Ra = Ra_initial
    best_i = NA
    for (i in 1:long) { #pour chaque variable, on teste le modele sans elle et
        on recalcule R^2 ajuste
      X_new = X_initial[, -i, drop = FALSE]
      Ra_new = find_Ra(X_new)
      if (Ra_new > best_Ra) { #si le nouveau R^2 ajuste est meilleur, alors on
          note l'index de la variable a supprimer et on met a jour le R^2 de
          reference
        best_Ra = Ra_new
        best_i = i
      }
```

```r
    }
    # si aucune amelioration n'a ete trouvee, on s'arrete
    if (is.na(best_i)) {
      break
    }
    cat("Suppression de la variable :", names(X_initial)[best_i], " -> Ra=",
        round(best_Ra,5), "\n")
    # on met a jour X_initial et Ra_initial
    X_initial = X_initial[, -best_i, drop = FALSE]
    Ra_initial = best_Ra
    long = ncol(X_initial)
  }
  cat("Variables finales selectionnees :", names(X_initial), "\n")
  cat("R^2 ajuste final :",round(best_Ra,5), "\n")
}
```

Listing 9: Fonction pour déterminer le meilleur $R^2_{\text{ajusté}}$: méthode pas à pas

Ce script R nous permet de réduire notre modèle de manière stratégique en supprimant progressivement des variables explicatives dès lors que le R^2 ajusté augmente. Par ailleurs, cela nous permet d'améliorer la qualité d'ajustement du modèle, tout en réduisant sa complexité.

Afin de calculer le coefficient R^2, il est nécessaire de connaître les valeurs prédites $\hat{Y}$. Pour cela, nous avons d'abord déterminé l'estimateur $\hat{\beta}$ des coefficients du modèle, en utilisant la méthode des moindres carrés.

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y$$

Une fois $\hat{\beta}$ obtenu, nous avons pu calculer les valeurs ajustées par le modèle selon :

$$\hat{Y} = X\hat{\beta}$$

Ces valeurs prédites $\hat{Y}$ sont ensuite utilisées pour évaluer la qualité de l'ajustement via le coefficient $R^2_{\text{ajusté}}$.

Table 5: Comparaison des modèles successifs

Modèle	R^2	$R^2_{\textbf{ajusté}}$	**Variables conservées**
X_1	0.853	0.823	Retrait de `vs` et `carb` considérées comme qualitatives
X_2	0.850	0.827	Retrait de `hp`
X_3	0.847	**0.830**	Retrait de `hp` et `disp`

Le modèle X_3 s'est avéré être le plus pertinent : il présente le $R^2_{\text{ajusté}}$ le plus élevé avec seulement trois variables explicatives. Cela montre qu'il est possible d'obtenir un modèle plus simple sans perte de performance, ce qui est souvent préférable en pratique.

Sélection exhaustive des variables explicatives

En complément de notre méthode pas à pas manuelle basée sur le $R^2_{\text{ajusté}}$, nous avons appliqué une approche automatique de sélection de variables à l'aide de la fonction `regsubsets()` du package `leaps`.

Cette méthode effectue une recherche exhaustive parmi toutes les combinaisons possibles de variables explicatives, et fournit, pour chaque nombre de variables, le meilleur sous-ensemble.

Dans notre cas, nous avons comparé les évolutions de R^2 et $R^2_{\text{ajusté}}$ en fonction du nombre de variables sélectionnées.

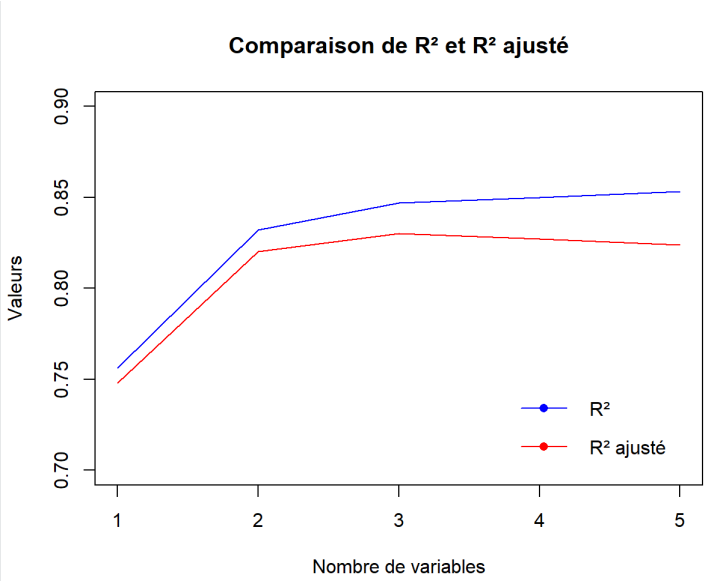

Figure 18: Évolution de R^2 et $R^2_{\text{ajusté}}$ selon le nombre de variables

On observe que le $R^2_{\text{ajusté}}$ atteint un maximum pour un sous-ensemble de taille 3, ce qui confirme la pertinence du modèle X_3 retenu précédemment dans notre sélection manuelle.

La méthode pas à pas présente l'avantage de réduire considérablement la complexité de l'algorithme de recherche du meilleur modèle.

En effet, contrairement à la méthode exhaustive qui teste l'ensemble des 2^p modèles possibles (soit $2^5 = 32$ modèles dans notre cas), la méthode pas à pas n'en teste qu'un nombre limité. Lorsqu'on commence avec $p = 5$ variables, le nombre maximal de modèles testés est donné par :

$$5 + 4 + 3 + 2 + 1 = 15$$

Cela permet un gain de temps et de performance non négligeable, tout en obtenant des résultats souvent très proches de ceux issus d'une sélection exhaustive.

4.2.3 Tests sur les coefficients du modèle de régression linéaire multiple

Une autre manière de valider la qualité de nos modèles de régression linéaire multiple est de tester si les coefficients estimés sont significativement différents de zéro.

Cependant, avant d'effectuer ces tests, nous devons impérativement vérifier une hypothèse centrale du modèle : l'hypothèse de gaussianité du bruit (les erreurs doivent suivre une loi normale centrée). Pour que ces tests soient valides, deux conditions préalables doivent être vérifiées concernant les résidus :

- Les **résidus standardisés** doivent suivre une loi normale centrée réduite ;
- Les **résidus studentisés** doivent suivre une loi de Student à $n - rang(X) - 1$ degrés de liberté.

a) Résidus Standardisés

Comme pour la régression linéaire simple, la distribution des résidus standardisés devrait approcher celle d'une loi $\mathcal{N}(0, 1)$ si l'hypothèse de bruit gaussien est valide.
Nous avons utilisé la fonction `rstandard()` pour extraire ces résidus à partir du modèle ajusté avec `lm()`, puis tracé les Q-Q plots associés. Voici le code correspondant pour le modèle X_1 :

```
m3 = lm(Y ~ X3mat)
rstm3 = rstandard(m3)
qqnorm(rstm3, main = "Q-Q plot des residus standardises")
```

```r
qqline(rstm3, col = "red", lwd = 2)
```

Listing 10: Q-Q plot des résidus standardisés (modèle X3)

Cette procédure a été répétée pour les modèles X_2 et X_1.

Nous avons ensuite tracé les graphes pour X1, X2 et X3 afin d'interpréter les résultats :

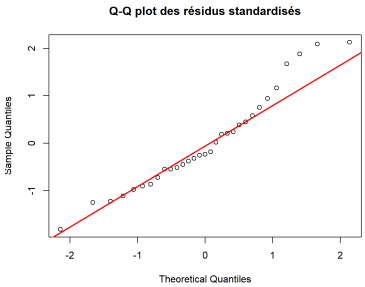

Figure 19: Q-Q plot des résidus standardisés pour X1

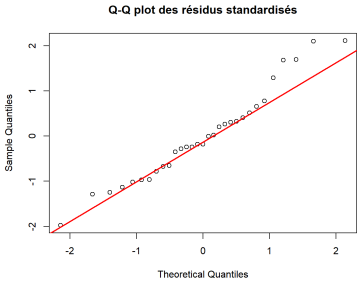

Figure 20: Q-Q plot des résidus standardisés pour X2

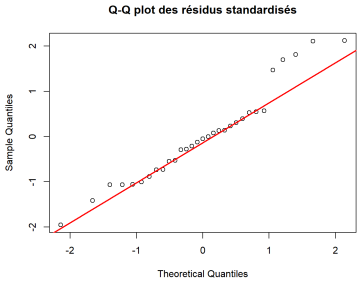

Figure 21: Q-Q plot des résidus standardisés pour X3

Les résultats graphiques sont satisfaisants : les points suivent globalement la droite théorique, indiquant que les résidus standardisés sont bien distribués selon une loi normale centrée réduite. On observe que c'est d'autant plus flagrant pour X3 (meilleur $R^2_{\text{ajusté}}$)

b) Résidus Studentisés

Les résidus Studentisés suivent, théoriquement, une loi de Student à $n - rang(X) - 1$ degrés de liberté, sous l'hypothèse de bruit gaussien.
Nous avons tracé la fonction de répartition empirique des résidus Studentisés et l'avons comparée à la courbe théorique de la loi de Student.

Voici notre code pour le modèle X_3 :

```r
#pourX3
rstd3=rstudent(m3)
srst3=sort(rstd3)
nt3=length(rstd3)
yst3=1/nt3*(1:nt3)
xt3=seq(srst3[1],srst3[nt3],0.01)
yt3=pt(xt3, n-rangX3-1)
plot(srst3,yst3,type='s',col='blue',xlim=c(srst3[1],srst3[nt3]),ylim=c(0,1))
lines(xt3,yt3,col='red')

ks.test(rstd3,'pt',n-rangX3-1) # p-value = 0.6424 >> 0.05
```

Listing 11: Comparaison des résidus Studentisés à une loi de Student

Nous avons donc obtenu les trois graphiques suivants pour X1, X2 et X3.

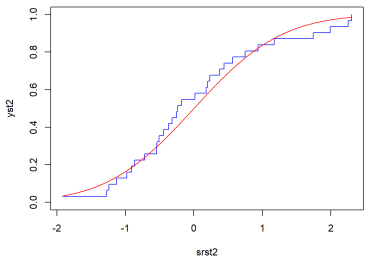

Figure 22: Plot des résidus Studentisés pour X1

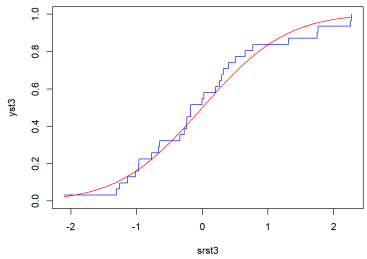

Figure 23: Plot des résidus Studentisés pour X2

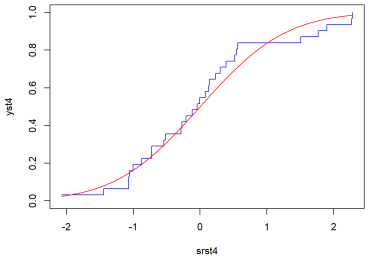

Figure 24: Plot des résidus Studentisés pour X3

Modèles	X1	X2	X3
p-valeur	0.7283	0.9485	0.6424

Table 6: p-valeurs pour chaque modèle (5, 4 et 3 variables quantitatives respectivement)

Nous observons que pour les 3 modèles peu importe le R^2 ajusté auquel ils sont associés, l'allure de la fonction de répartition des résidus Studentisés suit bien la courbe de référence. Par ailleurs, les p-valeurs obtenues sont toutes supérieures à α=5%. Nous en déduisons donc que les résidus Studentisés suivent bien la loi de Student à (n-rang(X) -1) degré de liberté, et donc que l'hypothèse de gaussianité du bruit est valide.

c) Test global sur les coefficients du modèle

Ayant validé l'hypothèse de bruit gaussien grâce aux résidus standardisés et studentisés, nous pouvons maintenant effectuer des tests sur l'ensemble des coefficients estimés du modèle de régression linéaire multiple.

Dans un modèle de régression linéaire multiple, le modèle s'écrit :

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i \quad \text{avec} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Nous souhaitons tester si les variables explicatives ont un effet globalement significatif sur la variable réponse Y. Autrement dit, nous testons si tous les coefficients associés aux variables explicatives sont nuls.

Hypothèses du test global :

- $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$ (aucune variable explicative n'a d'effet),
- $H_1 : \exists j \in \{1, \ldots, p\}$ tel que $\beta_j \neq 0$.

On utilise la statistique de test de Fisher suivante :

$$F = \frac{\|\widehat{Y} - \bar{Y}\|^2 / (rang(X) - 1)}{\|Y - \widehat{Y}\|^2 / (n - rang(X))}$$

où :

- $\widehat{Y}$ est le vecteur des valeurs prédites par le modèle,
- $\bar{Y}$ est la moyenne de la variable réponse,
- n est le nombre total d'observations,

Sous H_0, la statistique F suit une loi de Fisher $\mathcal{F}(rang(X)-1, n-rang(X))$. On rejette H_0 si la p-valeur est inférieure à $\alpha = 5\%$.

Voici un exemple de code utilisé pour le modèle X_3 :

```r
#Pour X3
F3 = (norm(Y_chap3 - mean(Y), type = c("2"))^2/(rangX3 - 1))/(norm(Y - Y_chap3,
    type = c("2"))^2/(n-rangX3))
pf3=pf(F3, rangX3-1, n - rangX3)
p_val3 = 1 - pf3 # = 3.872591e-11
#On decide H1
```

Listing 12: Test de Fisher pour le modèle X3

Ce code a été répété pour les modèles X_1 et X_2. Voici les résultats obtenus :

Modèle	p-valeur
X_1	$\approx 1.77 \times 10^{-9}$
X_2	$\approx 2.33 \times 10^{-10}$
X_3	$\approx 3.87 \times 10^{-11}$

Table 7: Test global de significativité des coefficients

Dans tous les cas, la p-valeur est très inférieure à $\alpha = 5\%$, donc on décide H_1 : il existe bien au moins une variable explicative ayant un effet significatif sur Y. Nos modèles sont donc statistiquement pertinents.

4.3 Affichage graphique du modèle de régression multiple

En régression linéaire simple, la droite de régression peut être représentée graphiquement dans le plan (X, Y). En revanche, en régression linéaire multiple, lorsque plusieurs variables explicatives sont en jeu, il devient impossible de visualiser directement le modèle dans un espace à plus de trois dimensions.

Cependant, il reste pertinent d'évaluer la qualité du modèle à l'aide d'une représentation graphique des valeurs prédites. Pour cela, nous avons tracé les valeurs réelles de la variable dépendante Y en fonction des valeurs prédites $\hat{Y}$, obtenues à partir de nos différents modèles.

Ce type de graphique permet de visualiser facilement si le modèle fournit de bonnes prédictions :

- Si les points sont proches de la diagonale $Y = \hat{Y}$, cela signifie que le modèle prédit correctement la variable d'intérêt.
- Une dispersion importante autour de cette diagonale indique une mauvaise qualité de prédiction.

Nous rédigeons alors le code suivant :

```r
plot(Y, Y_chap3,
     xlab="Y reel", ylab="Y predit",
     main="Regression multiple : Y vs Y_chap",
     pch=19, col="darkblue")
abline(0, 1, col="red", lwd=2)  # Ligne ideale : Y = Y_chap
```

Listing 13: Plot de notre modèle de régression multiple : Y vs $\hat{Y}$

Ce graphique est donc un outil complémentaire aux indicateurs numériques comme R^2 ajusté, en fournissant une évaluation visuelle simple de la performance du modèle.

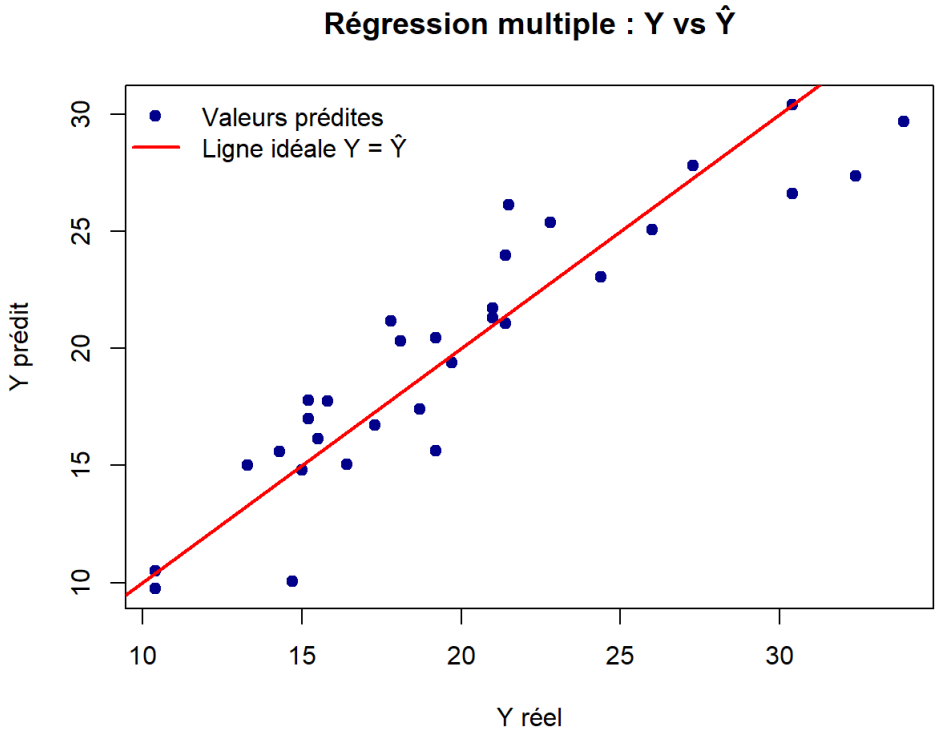

Figure 25: Régression multiple : comparaison entre Y réel et $\hat{Y}$ prédit

Le nuage de points représentant les valeurs observées Y en fonction des valeurs prédites $\hat{Y}$ pour le modèle X_3 est globalement aligné avec la droite idéale $Y = \hat{Y}$. Cela signifie que les prédictions fournies par notre modèle sont proches des valeurs réelles, ce qui traduit une bonne qualité d'ajustement.

Cette observation visuelle est cohérente avec le coefficient de détermination ajusté $R^2_{\text{ajusté}}$, qui atteint ici une valeur élevée (environ 0,83). Cela indique que le modèle X_3, obtenu après sélection des variables, permet d'expliquer une grande partie de la variabilité de la variable réponse tout en évitant le sur-ajustement.

4.4 Prédiction

Pour rappel, effectuer des prédictions consiste à estimer la valeur de la variable réponse (Y) lorsqu'une nouvelle donnée est disponible pour la variable explicative.
Ainsi, si l'on note x_{new} une nouvelle observation de la variable explicative, alors la prédiction associée de la variable réponse est donnée par la relation :

$$y_{\text{new}} = (1 \;\; x_{\text{new}}) \cdot \widehat{\beta}_n$$

où :

- $(1 \;\; x_{\mathbf{new}})$ est un vecteur ligne qui contient :
  - le **1** pour prendre en compte l'ordonnée à l'origine (c'est-à-dire la constante du modèle),
  - et $x_{\mathbf{new}}$, la nouvelle valeur de la variable explicative que l'on utilise pour faire la prédiction.
- $\widehat{\beta}_n$ est un vecteur colonne contenant les deux coefficients estimés par la régression :
  - le premier est l'ordonnée à l'origine (notée $\widehat{b}_n$),
  - le second est le coefficient directeur (la pente, notée $\widehat{a}_n$).

Leur produit correspond à l'équation de la droite de régression : $\widehat{y}_{\text{new}} = \widehat{a}_n x_{\text{new}} + \widehat{b}_n$.

Le modèle ajusté peut maintenant être utilisé pour faire des prédictions de consommation `mpg` à partir de plusieurs variables techniques de la voiture. Ces prédictions sont d'autant plus fiables que les hypothèses du modèle sont respectées.

Dans le code suivant, nous allons donc calculer : la valeur prédite pour un nouveau x_{new}, un intervalle de confiance à 95% pour la moyenne prédite, et un intervalle de prédiction à 95% pour une valeur individuelle. Nous réalisons ces calculs pour deux cas différents :

1. Dans le premier, le modèle contient toutes les variables quantitatives, sauf la première.
2. Dans le second, le modèle est réduit puisque l'on enlève la 1ère et la 6ème colonne.

```
1  # 1er cas : on enleve la premiere colonne seulement
2  x_new = A[set1, set2[-1], drop = FALSE]
3  x_new_mat = model.matrix(~ ., data = x_new)
4  y_pred = x_new_mat %*% beta_chap1
5
6  on2_chap = 1 / (n - rangX1) * sum((Y - Y_chap1)^2)
7
8  facteur_IC = diag(x_new_mat %*% solve(XtX1) %*% t(x_new_mat))
9  alpha = 0.05
10 t_alpha = qt(1 - alpha / 2, df = n - rangX1)
11
12 IC_lower = y_pred - sqrt(on2_chap) * sqrt(facteur_IC) * t_alpha
13 IC_upper = y_pred + sqrt(on2_chap) * sqrt(facteur_IC) * t_alpha
14
15 facteur_IP = 1 + facteur_IC
16 IP_lower = y_pred - sqrt(on2_chap) * sqrt(facteur_IP) * t_alpha
17 IP_upper = y_pred + sqrt(on2_chap) * sqrt(facteur_IP) * t_alpha
18
19 cat("Valeur predite :", y_pred, "\n")
20 cat("IC 95% :", round(IC_lower, 3), ",", round(IC_upper, 3), "\n")
21 cat("IP 95% :", round(IP_lower, 3), ",", round(IP_upper, 3), "\n")
```

Listing 14: Prédiction et calcul des intervalles en régression multiple (modèle 1)

Explication du code :

- (Ligne 2) On extrait la ligne "set1" de A et on enlève la première variable de "set2" car elle est qualitative.
- (Ligne 3) On construit la matrice du modèle.
- (Ligne 4) On calcule la valeur prédite : $\widehat{y}_{\text{new}} = \widehat{a}_n x_{\text{new}} + \widehat{b}_n$.
- (Ligne 6) On calcule l'estimation de la variance des erreurs :

$$\widehat{\sigma}_n^2 = \frac{1}{n - \text{rang}(X)} \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i\right)^2$$

- (Ligne 8) On calcule le facteur de variabilité pour l'intervalle de confiance. Ce facteur nous permet de mesurer l'incertitude liée à la nouvelle observation dans l'estimation de la moyenne :

$$\text{facteur}_{\text{IC}} = (1 \;\; x_{\text{new}}) \cdot \left(X^\top X\right)^{-1} \cdot (1 \;\; x_{\text{new}})^\top$$

- (Ligne 9) On pose $\alpha = 5\% = 0{,}05$, ce qui correspond à un niveau de confiance de 95%.
- (Ligne 12 et 13) On calcule les bornes inférieures et supérieures de notre intervalle de confiance : Les bornes de l'intervalle de confiance à 95% pour la valeur moyenne prédite sont données par :

$$\text{IC}_{\text{inf}} = \widehat{y}_{\text{new}} - \widehat{\sigma}_n \cdot \sqrt{\text{facteur}_{\text{IC}}} \cdot t_{1-\alpha/2,\, n-r}$$

$$\text{IC}_{\text{sup}} = \widehat{y}_{\text{new}} + \widehat{\sigma}_n \cdot \sqrt{\text{facteur}_{\text{IC}}} \cdot t_{1-\alpha/2,\, n-r}$$

où :

- $\widehat{y}_{\text{new}}$: la prédiction de la variable réponse pour la nouvelle observation x_{new},
- $t_{1-\alpha/2,\, n-r}$: le quantile de la loi de Student au seuil $1-\alpha/2$, avec $(n-r)$ degrés de liberté,
- $\widehat{\sigma}_n$: l'écart-type estimé des résidus,
- facteur$_{\text{IC}}$: l'incertitude liée à la nouvelle observation.

- (Ligne 15, 16 et 17) On ajoute 1 au facteur de notre intervalle de confiance pour tenir compte de la variabilité aléatoire d'une future valeur de Y, puis on calcule nos bornes inférieures et supérieures à l'aide des formules précédentes. On obtient ainsi notre intervalle de prédiction.
- (Ligne 19, 20 et 21) On affiche les résultats obtenus.

Ainsi, le code nous renvoie les résultats suivants :
La valeur prédite pour x_{new} est : 26.56509
un intervalle de confiance à 95% est : [20.74 , 32.391]
un intervalle de prédiction à 95% est : [18.63 , 34.501]

Interprétation :

- La voiture retirée aurait une consommation prédite d'environ 26.57 mpg.
- On est 95% confiant que la vraie valeur moyenne de "mpg" pour une voiture avec les mêmes caractéristiques est entre 20.74 et 32.391 mpg.
- Néanmoins, une observation individuelle pourrait tomber entre 18.63 et 34.501 mpg. En effet, l'intervalle de prédiction est plus large car plus incertain.

Si l'on se penche maintenant sur le second modèle, le code reste le même, à ceci-près que :

- Pour la ligne 1, on extrait la ligne `set1` de `A` et on enlève la première **et la 7$^{\text{e}}$ variable** de `set2` car elles sont qualitatives (elles correspondent à "vs" et "carb").
- On remplace tous les `beta_chap1` par des `beta_chap2`, de même pour `rangX1`, `Y_chap1` et `XtX1`.

Ainsi, le code nous renvoie les résultats suivants :
La valeur prédite pour x_{new} est : 21.77235
un intervalle de confiance à 95% est : [15.722 , 27.823]
un intervalle de prédiction à 95% est : [13.744 , 29.801]

Interprétation :

- La voiture retirée aurait une consommation prédite d'environ 21.77235 mpg.
- On est 95% confiant que la vraie valeur moyenne de "mpg" pour une voiture avec les mêmes caractéristiques est entre 15.722 et 27.823 mpg.
- Néanmoins, une observation individuelle pourrait tomber entre 13.744 et 29.801 mpg. En effet, l'intervalle de prédiction est plus large car plus incertain.

5 Analyse de la variance à un facteur

5.1 Objectif de la méthode

L'analyse de la variance à un facteur est une méthode statistique qui permet d'évaluer si une variable qualitative, appelée facteur, a une influence significative sur une variable quantitative.

L'idée principale de cette méthode est de comparer les moyennes de plusieurs modalités pour déterminer si elles se comportent de la même façon, ou si elles diffèrent de manière significative. Cela permet de répondre à des questions du type :

- La consommation d'essence (mpg) dépend-elle du type de moteur vs ?
- Le nombre de carburateurs (carb) influence-t-il la consommation d'essence (mpg)?

Principe de fonctionnement :

On considère une variable réponse Y (quantitative) et un facteur A (qualitatif) ayant k modalités, notées $A_1, A_2, \ldots, A_k$. À chaque modalité A_j, on observe un certain nombre n_j de valeurs de Y.

Hypothèses du test :

- H_0 : les moyennes des modalités sont égales. Autrement dit, le facteur A n'a pas d'effet :

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$$

- H_1 : au moins une moyenne est différente :

$$H_1 : \exists (i, j) \text{ tel que } \mu_i \neq \mu_j$$

Prise de décision :

- Si la p-valeur du test est inférieure à α (généralement 5%), on rejette H_0 : le facteur a un effet significatif.
- Sinon, on ne rejette pas H_0 : aucune preuve suffisante que les moyennes diffèrent.

Maintenant que nous avons présenté le principe et les objectifs de l'analyse de la variance à un facteur, nous allons l'appliquer à notre jeu de données. Plus précisément, nous chercherons à savoir si les variables qualitatives `vs` (type de moteur) et `carb` (nombre de carburateurs) ont un effet significatif sur la variable `mpg` (consommation).
Dans un premier temps, nous procéderons à une exploration visuelle à l'aide de boîtes à moustaches, avant de réaliser les tests d'hypothèses associés, suivis de comparaisons multiples (Tukey HSD) pour interpréter plus finement les résultats.

5.2 Visualisation par boîtes à moustaches

Dans un premier temps, nous avons représenté les répartitions de la variable réponse selon les modalités des facteurs étudiés grâce à des boîtes à moustaches. Cela nous permet de visualiser les différences de médianes et de moyennes (ajoutées en rouge).

- Pour la variable qualitative `vs` (type de moteur) :

```
vs = B$vs
boxplot(Y ~ vs,
                  main = "Analyse de la variance : mpg selon vs",
                  xlab = "Type de moteur (vs)",
                  ylab = "mpg",
                  col = c("lightblue", "lightgreen"))

moyenne_vs = tapply(Y, vs, mean)
points(1:2, moyenne_vs, col = "red", pch = 18, cex = 1.5)
```

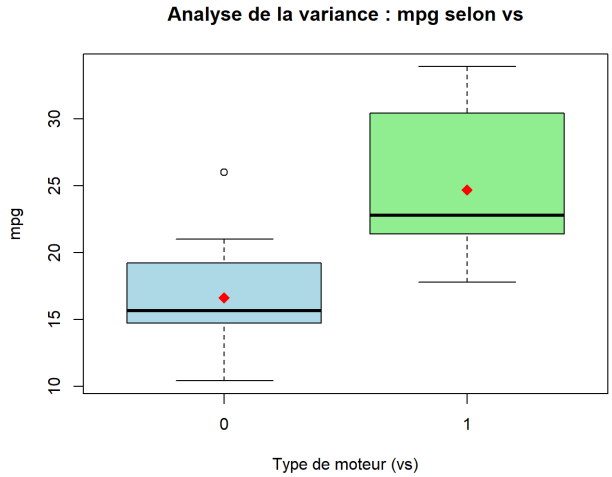

Figure 26: mpg selon le type de moteur (`vs`)

On observe que les moyennes de `mpg` varient visiblement selon le type de moteur (0 = en V, 1 = en ligne), ce qui suggère un effet potentiel du facteur `vs`.

- Pour la variable qualitative `carb` (nombre de carburateurs) :

```r
carb = as.factor(B$carb)
boxplot(Y ~ carb,
        main = "Analyse de la variance : mpg selon le nombre de carburateurs",
        xlab = "Nombre de carburateurs",
        ylab = "mpg",
        col = c("lightblue", "lightgreen","pink","yellow","violet","orange"))

moyenne_carb = tapply(Y, B$carb, mean)
points(1:length(moyenne_carb), moyenne_carb, col = "red", pch = 18, cex = 1.5)
```

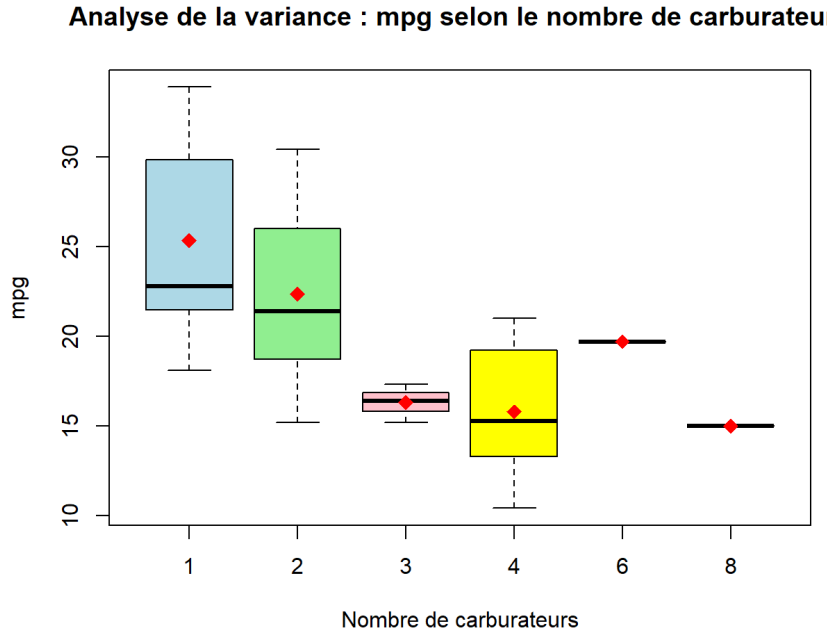

Figure 27: mpg selon le nombre de carburateurs (`carb`)

Ici, les boîtes à moustaches indiquent des écarts de moyenne importants entre certaines modalités, tandis que d'autres ont l'air proches. On s'est alors demandé si on ne pouvait pas regrouper certaines modalités entre elles pour simplifier notre modèle.

5.3 Test sur les moyennes

Nous avons ensuite réalisé un test d'analyse de la variance pour chaque facteur. L'objectif est de tester :

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k \quad \text{vs} \quad H_1 : \exists (i,j) \text{ tel que } \mu_i \neq \mu_j$$

où μ_i représente la moyenne de mpg pour la i-ème modalité du facteur.

- Pour `vs` :

```
L_vs = lm(Y ~ vs)
aov_vs = aov(L_vs)
summary(aov_vs)
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)
vs            1  492.3   492.3    22.8 4.74e-05 ***
Residuals    29  626.2    21.6
```

Figure 28: p-valeur renvoyée : 4.74e-05

La p-valeur du test est très petite ($4.74e-05 < 0.05$), ce qui nous amène à rejeter H_0 : le type de moteur a un effet significatif sur la consommation.

- Pour `carb` :

```
L_carb = lm(Y ~ carb)
aov_carb = aov(L_carb)
summary(aov_carb)
```

```
             Df Sum Sq Mean Sq F value  Pr(>F)
carb          5  493.2   98.63   3.943 0.00898 **
Residuals    25  625.3   25.01
```

Figure 29: p-valeur renvoyée : 8.98e-03

Ici aussi, la p-valeur est très faible ($8.98e-03 < 0.05$), indiquant que le nombre de carburateurs peut influencer significativement la consommation.

5.4 Test de Tukey

Une fois l'analyse de la variance validée, nous avons effectué un test post-hoc de Tukey afin de déterminer quelles modalités sont significativement différentes entre elles. Ce test permet de comparer toutes les paires de modalités deux à deux.

Il repose sur la construction d'intervalles de confiance pour les différences de moyennes deux à deux entre modalités. Ces intervalles sont représentés graphiquement, et leur interprétation est la suivante :

- **Si l'intervalle de confiance contient la valeur 0**, cela signifie qu'il n'y a pas de différence significative entre les deux modalités comparées. Les modalités peuvent être **regroupées**.
- **Si l'intervalle de confiance ne contient pas 0**, cela signifie qu'il existe une différence significative entre les deux modalités. Les modalités doivent être **considérées séparément**.

Les graphiques générés par la fonction `TukeyHSD()` permettent de visualiser rapidement cette information : chaque ligne correspond à une comparaison entre deux modalités, et les barres horizontales représentent les intervalles de confiance. Si une barre recoupe la verticale en zéro, alors la différence n'est pas significative.

- Pour `vs` :

```
THSD_vs=TukeyHSD(aov_vs)
plot(THSD_vs)
```

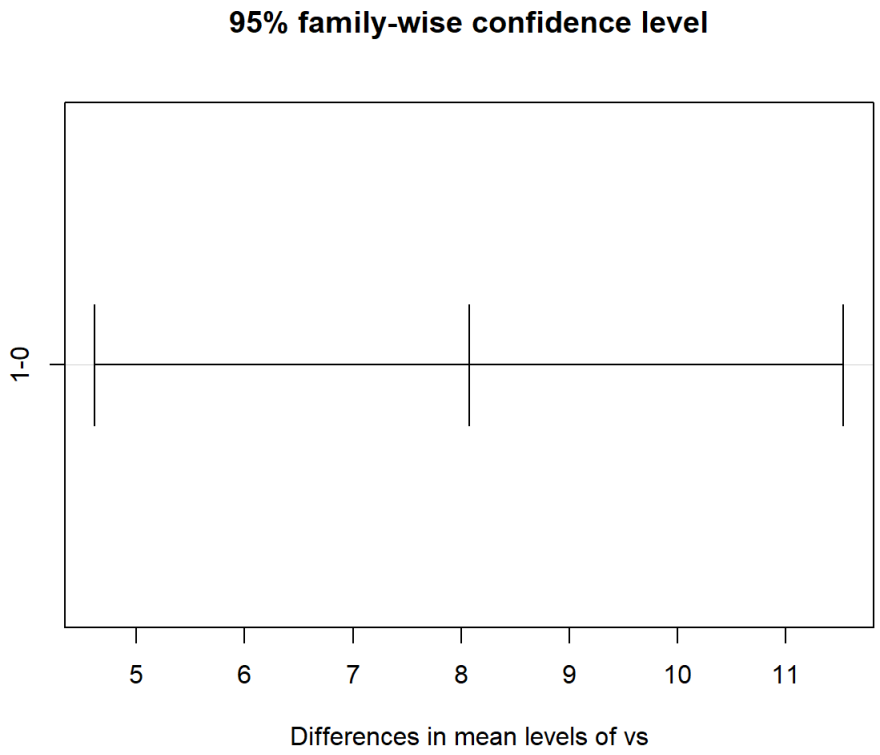

Figure 30: Intervalle de confiance, test de Tukey (variable `vs`)

On remarque donc ici que pour les deux modalités de vs (0 et 1) testées l'une contre l'autre, la valeur zéro n'est pas comprise dans l'intervalle de confiance. Les deux modalités doivent donc être considérées séparément comme supposé avec les boîtes à moustaches.

- Pour `carb` :

```
THSD_carb = TukeyHSD(aov_carb, conf.level = 0.99)
plot(THSD_carb)
```

Afin de prendre en compte le fait que nous faisons plusieurs comparaisons (15 pour `carb`), nous avons fixé un niveau de confiance à 99%, soit $\alpha = \frac{5\%}{15} \approx 0.0033$.

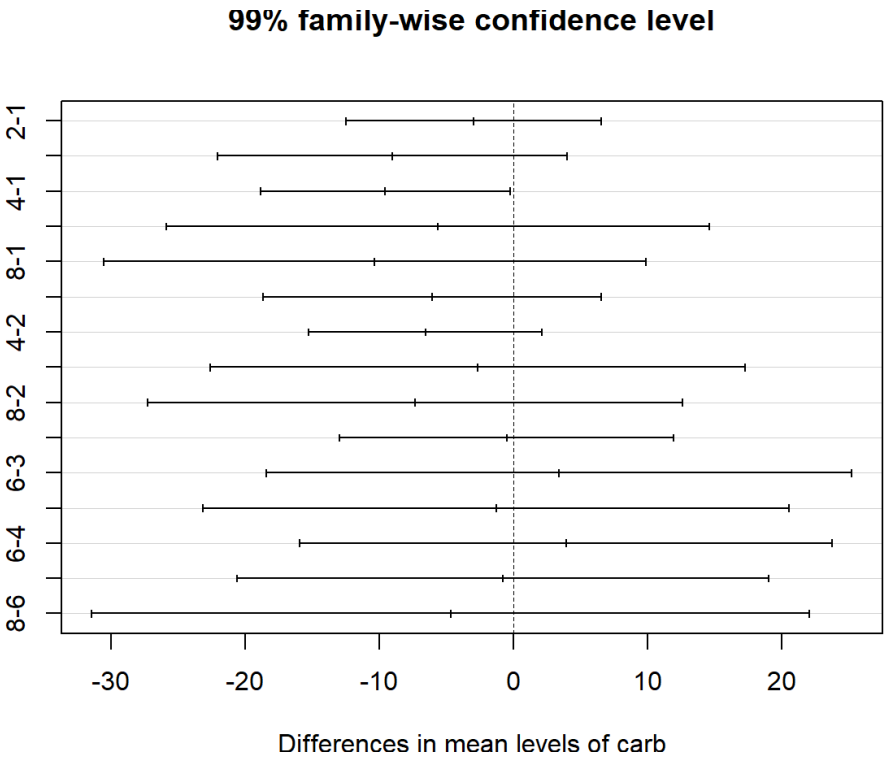

Figure 31: Intervalles de confiance, test de Tukey (variable `carb`)

Pour la variable qualitative `carb`, nous constatons que plusieurs intervalles de confiance issus du test de Tukey incluent la valeur zéro. Cela signifie qu'il n'existe pas de différence significative entre certaines modalités prises deux à deux. Par conséquent, il est pertinent de regrouper ces modalités entre elles afin de simplifier l'analyse, tout en conservant une interprétation statistiquement cohérente.

Dans notre projet, cette méthode nous a permis d'identifier des groupes de modalités qui se comportent statistiquement de manière similaire par rapport à la variable réponse `mpg`. Nous avons donc regroupé certaines modalités afin de simplifier l'interprétation et rendre le modèle plus stable et plus lisible.

5.5 Regroupement des modalités

À l'issue du test de Tukey, certaines modalités ne présentaient pas de différence significative. Nous avons donc décidé de les regrouper pour simplifier le modèle et l'interprétation. Cela permet notamment de stabiliser les résultats en réduisant la complexité.

Nous avons défini deux groupes :

- Groupe A : `carb` = 1, 2, 5
- Groupe B : `carb` = 3, 4, 6

Après regroupement, une nouvelle analyse de la variance a été réalisée avec la variable `carb_group`, confirmant l'existence d'un effet significatif entre les deux groupes.

Visualisation de la pertiance du regroupement :

- 1. Avec les boîtes à moustaches :

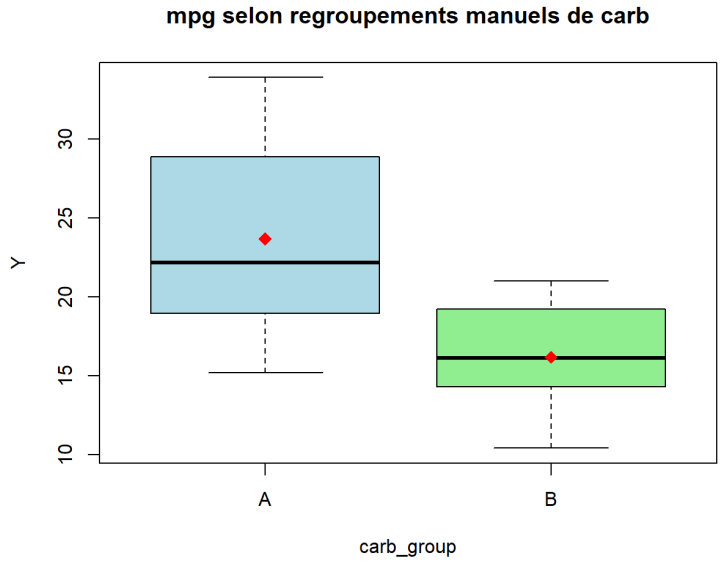

Figure 32: mpg selon les groupes manuels de `carb`

On voit ici les deux boîtes à moustaches qui présentent des médianes et des moyennes différentes. Donc, cela tend à penser que le facteur à une influence sur la variable réponse.

- 2. Test sur les moyennes :

Pour mieux interpréter ce qu'on observe sur les boîtes à moustaches, on fait un test ici sur les moyennes.

```
1 carb_levels = as.character(B$carb)
2 carb_group = rep(NA, length(carb_levels))
3
4 carb_group[carb_levels %in% c("1", "2", "5")] = "A"
5 carb_group[carb_levels %in% c("3", "4", "6")] = "B"
6
7 B$carb_group <- as.factor(carb_group)
8 summary(aov(Y ~ B$carb_group))
```

Explication du code :

- (Ligne 1) On transforme la variable qualitative carb (initialement un facteur) en chaîne de caractères.
- (Ligne 2) On crée un vecteur carb group rempli de NA, de même longueur que le nombre d'observations. Ce vecteur servira à stocker le groupe (A ou B) associé à chaque ligne. Cela sert à initialiser le vecteur carb group avec des cases vides (non encore attribuées).
- (Ligne 4 et 5) On attribue le groupe "A" aux modalités de carb qui valent "1", "2" ou "5". On assigne le groupe "B" aux modalités "3", "4" et "6".
- (Ligne 7) On ajoute à la base de données B une nouvelle variable carb group, convertie en facteur.
- (Ligne 8) On réalise une analyse de la variance sur la variable réponse Y, en fonction des deux groupes (A et B).

```
              Df Sum Sq Mean Sq F value   Pr(>F)
B$carb_group  1  418.2   418.2   17.36 0.000268 ***
Residuals    28  674.4    24.1
```

Figure 33: p-valeur renvoyé : 2.68e-04

La p-valeur du test est très petite ($2.68e-04 < 0.05$), ce qui nous amène à rejeter H0.

- 3. Test de Tukey :

Finalement, on vérifie grâce au test de Tukey que l'intervalle de confiance de nos 2 regroupements ne contient pas la valeur zéro : les modalités regroupées doivent être considérées séparément.

```
TukeyHSD(aov(Y ~ B$carb_group), conf.level = 0.95)
plot(TukeyHSD(aov(Y ~ B$carb_group), conf.level = 0.95))
```

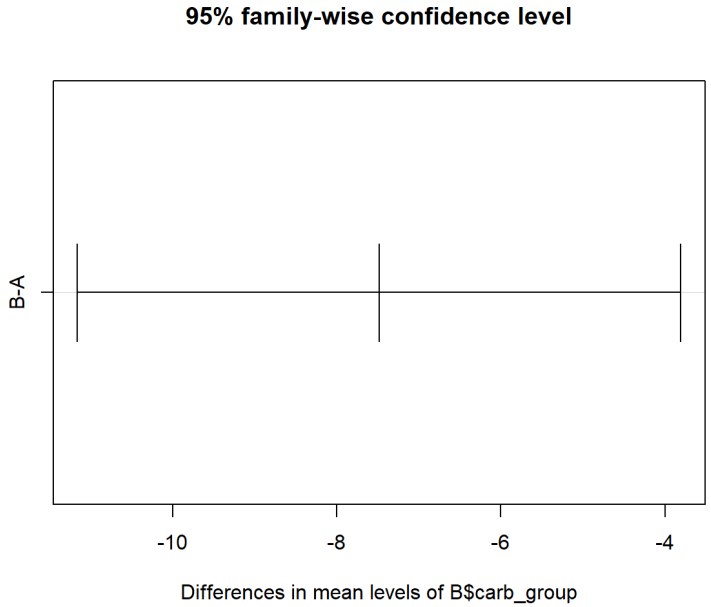

Figure 34: Intervalle de confiance pour les groupes de carb

On remarque donc ici que pour les deux groupes de carb (A et B) testés l'un contre l'autre, la valeur zéro n'est pas comprise dans l'intervalle de confiance. Les deux groupes de modalités doivent donc être considérés séparément comme supposé avec les boîtes à moustaches.

Conclusion : Ce regroupement permet de simplifier l'analyse tout en conservant l'essentiel de l'information.

6 Conclusion

Ce projet nous a permis de comprendre et d'appliquer la régression linéaire, simple et multiple, sur un jeu de données réel. Grâce à différentes analyses (coefficient R^2 , tests statistiques, validation des hypothèses), nous avons pu évaluer à la fois la qualité des modèles que que nous avons construits, et leur capacité à prédire la variable réponse.
Nous avons aussi appris à sélectionner les variables les plus pertinentes pour obtenir un modèle à la fois simple et efficace. Enfin, les représentations graphiques et les intervalles de prédiction nous ont aidées à mieux visualiser et interpréter les résultats.