Université Catholique de Louvain

LMAT1271 - Calcul de Probabilité et Analyse Statistique

Projet

Professeur :
Mr. R von Sachs

Travail réalisé par :
Marie Determe
Augustin Lambotte
Amaury Laridon

École de Physique

Faculté des Sciences

26 mai 2023

Table des matières

1 Partie I - Estimation ponctuelle

1.1 Contexte et analyse préparatoire

Question (a)

Par définition, soit p un nombre entre 0 et 1 et T une variable aléatoire de densité $f_{\alpha,\beta}$. Le nombre q_p tel que $P(T \leq q_p) = p$ est le quantile d'ordre p de T.
On doit ainsi trouver la solution à l'équation $F_{\alpha,\beta}(q_p) = p$. On calcule donc d'abord explicitement la fonction de répartition [1],

$$F_{\alpha,\beta}(t) = P(T \leq t) = \int_0^t f_{\alpha,\beta}(x)dx \quad \text{Pour } x \in [0,1] \tag{1.1}$$

$$= 0 \quad \text{sinon} \tag{1.2}$$

Donc pour $x \in [0,1]$,

$$F_{\alpha,\beta}(t) = \int_0^t \alpha\beta x^{\alpha-1}(1-x^\alpha)^{\beta-1}dx \tag{1.3}$$

$$= -\beta \int_1^{1-t^\alpha} du\, u^{\beta-1} \tag{1.4}$$

Avec à la dernière ligne le changement de variable $u = 1 - x^\alpha$ ce qui donne finalement,

$$F_{\alpha,\beta}(t) = 1 - (1-t^\alpha)^\beta \tag{1.5}$$

On cherche donc à résoudre maintenant

$$F_{\alpha,\beta}(q_p) = 1 - (1-q_p^\alpha)^\beta = p. \tag{1.6}$$

Ce qui donne finalement,

$$\boxed{q_p = \sqrt[\alpha]{1 - \sqrt[\beta]{1-p}}} \tag{1.7}$$

Question (b)

Par définition on a que,

$$\mathbb{E}(T^k) = \int_{-\infty}^{\infty} t^k f_{\alpha\beta}(t)dt \tag{1.8}$$

$$= \alpha\beta \int_0^1 t^{\alpha+k-1}(1-t^\alpha)^{\beta-1}dt \quad \text{Pour } t \in [0,1] \tag{1.9}$$

Avec bien sûr $\mathbb{E}(T^k) = 0$ pour $t \notin [0,1]$.
On peut poser $y = t^\alpha$, on a alors $dt = \alpha^{-1}t^{1-\alpha}dy$, donc toujours pour $t \in [0,1]$,

$$\mathbb{E}(T^k) = \beta \int_0^1 y^{k/\alpha}(1-y)^{\beta-1}dy \tag{1.10}$$

$$= \beta \int_0^1 y^{(k/\alpha+1)-1}(1-y)^{\beta-1}dy \tag{1.11}$$

$$= \beta \frac{\Gamma(\frac{k}{\alpha}+1)\Gamma(\beta)}{\Gamma(\frac{k}{\alpha}+\beta+1)} \tag{1.12}$$

1. On prend la notation d'écrire comme suggéré dans l'énoncé la fonction de densité de probabilité de la variable aléatoire T de la façon suivante $f_{\alpha,\beta}$ bien que dans le cours la notation usuelle soit de mettre en indice la variable aléatoire dont la fonction est la fonction de densité de probabilité. Ici on met en indice les paramètres dont dépendent la fonction de densité de probabilité de la variable aléatoire T. Puisque l'exercice traite qu'une seule variable aléatoire il n y a pas d'ambiguïté sur les notations.

Or, $\Gamma(x+1) = x\Gamma(x)$ donc finalement,

$$\boxed{\mathbb{E}(T^k) = \frac{\beta k}{k+\alpha\beta}\frac{\Gamma(\frac{k}{\alpha})\Gamma(\beta)}{\Gamma(\frac{k}{\alpha}+\beta)}} \tag{1.13}$$

1.2 Estimation

Question (a)

Puisque la statistique est ordonnée de $T_1, \ldots, T_n$ on peut sans aucun calcul proposer un premier estimateur q_S de $q_{0.95}$ de la façon suivante [2],

$$\boxed{q_S = \frac{\left(T_{\lceil 0.95n \rceil} + T_{\lfloor 0.95n \rfloor}\right)}{2}} \tag{1.14}$$

Question (b)

On va appliquer la méthode des moments sur T^{α_M} afin d'obtenir un estimateur $(\hat{\alpha}_M, \hat{\beta}_M)$ de (α, β).

On doit avoir :

$$\begin{cases} \frac{1}{n}\sum_{i=1}^{n} T_i^{\hat{\alpha}_M} = \quad \mathbb{E}(T^{\hat{\alpha}_M}; \hat{\alpha}_M, \hat{\beta}_M) \\ \frac{1}{n}\sum_{i=1}^{n} T_i^{2\hat{\alpha}_M} = \quad \mathbb{E}(T^{2\hat{\alpha}_M}; \hat{\alpha}_M, \hat{\beta}_M) \end{cases} \tag{1.15}$$

Par (1.13), on sait que

$$\mathbb{E}(T^{\hat{\alpha}_M}) = \frac{\hat{\beta}_M\hat{\alpha}_M}{\hat{\alpha}_M+\hat{\alpha}_M\hat{\beta}_M}\frac{\Gamma(1)\Gamma(\hat{\beta}_M)}{\Gamma(1+\hat{\beta}_M)} \tag{1.16}$$

et que

$$\mathbb{E}(T^{2\hat{\alpha}_M}) = \frac{2\hat{\beta}_M\hat{\alpha}_M}{2\hat{\alpha}_M+\hat{\alpha}_M\hat{\beta}_M}\frac{\Gamma(2)\Gamma(\hat{\beta}_M)}{\Gamma(2+\hat{\beta}_M)} \tag{1.17}$$

En utilisant le fait que $\Gamma(x+1) = x\Gamma(x)$, on obtient par calcul :

$$\begin{cases} \frac{1}{n}\sum_{i=1}^{n} T_i^{\hat{\alpha}_M} = \frac{1}{\hat{\beta}_M+1} \\ \frac{1}{n}\sum_{i=1}^{n} T_i^{2\hat{\alpha}_M} = \frac{2}{(\hat{\beta}_M+1)(\hat{\beta}_M+2)} \end{cases} \tag{1.18}$$

Par la première ligne de (1.18), on a donc :

$$\boxed{\hat{\beta}_M = \left(\frac{1}{\frac{1}{n}\sum_{i=1}^{n} T_i^{\hat{\alpha}_M}}\right) - 1} \tag{1.19}$$

En utilisant la première ligne de (1.18) et en remplaçant $\hat{\beta}_M$ dans la 2ème ligne de (1.18) par la valeur trouvée en (1.19), on a :

$$\frac{1}{n}\sum_{i=1}^{n} T_i^{2\hat{\alpha}_M} = \frac{2}{n}\sum_{i=1}^{n} T_i^{\hat{\alpha}_M} \cdot \frac{1}{\left(\frac{1}{\frac{1}{n}\sum_{i=1}^{n} T_i^{\hat{\alpha}_M}}\right)+1} \tag{1.20}$$

2. On utilise les notations suivantes $\forall x \in \mathbb{R}, \lfloor x \rfloor = y$ avec y le plus grand entier inférieur à x et la notation $\lceil x \rceil = z$ avec z le plus petit entier supérieur à x.

qui est équivalent à

$$\boxed{\left(\frac{1}{n}\sum_{i=1}^{n} T_i^{2\hat{\alpha}_M}\right)\cdot\left(n+\sum_{i=1}^{n} T_i^{\hat{\alpha}_M}\right)\cdot\left(\sum_{i=1}^{n} T_i^{\hat{\alpha}_M}\right)^{-2}=2} \tag{1.21}$$

Question (c)

Pour utiliser la méthode du maximum de vraisemblance on calcule tout d'abord la fonction de vraisemblance,

$$L((\alpha,\beta)) \equiv L((\alpha,\beta)|t_1,\ldots,t_n) = f_T(t_1,\ldots,t_n;(\alpha,\beta)) \tag{1.22}$$

et puisque $T_1,\ldots,T_n$ sont i.i.d on a,

$$L((\alpha,\beta)) = \Pi_{i=1}^{n} f_T(t_i;(\alpha,\beta)). \tag{1.23}$$

Donc,

$$L((\alpha,\beta)) = \Pi_{i=1}^{n} f_{\alpha,\beta}(t_i) = \Pi_{i=1}^{n}(\alpha\beta t_i^{\alpha-1}(1-t_i^{\alpha})^{\beta-1}\mathbb{I}(0<t_i<1)) \tag{1.24}$$

$$= (\alpha\beta)^n \Pi_{i=1}^{n}(t_i^{\alpha-1}(1-t_i^{\alpha})^{\beta-1}\mathbb{I}(0<t_i<1)) \tag{1.25}$$

On définit ensuite,

$$l((\alpha,\beta)) = \ln(L((\alpha,\beta)) \tag{1.26}$$

et on cherche à résoudre $(\hat{\alpha}_L,\hat{\beta}_L) = arg\,max(l(\alpha,\beta))$ avec $(\alpha,\beta)\in\Theta$. On calcule,

$$l((\alpha,\beta)) = n\ln(\alpha)+n\ln(\beta)+\sum_{i=1}^{n}\ln(t_i^{\alpha-1}(1-t_i^{\alpha})^{\beta-1} \quad \text{pour} \quad (0<t_i<1). \tag{1.27}$$

Pour trouver $(\hat{\alpha}_L,\hat{\beta}_L)$ on calcule,

$$\nabla_{(\alpha,\beta)} l((\alpha,\beta)) = 0 \tag{1.28}$$

ou encore,

$$\frac{\partial l(\alpha,\beta)}{\partial\alpha}|_{\hat{\alpha}_L,\hat{\beta}_L} = 0 \tag{1.29}$$

$$\frac{\partial l(\alpha,\beta)}{\partial\beta}|_{\hat{\alpha}_L,\hat{\beta}_L} = 0 \tag{1.30}$$

ce qui après quelques lignes d'algèbre donne,

$$\frac{n}{\hat{\alpha}_L}+\sum_{i=1}^{n}\frac{T_i^{\hat{\alpha}_L-1}ln(T_i)(1-T_i^{\hat{\alpha}_L})^{\hat{\beta}_L-1}-T_i^{2\hat{\alpha_L}-1}(\hat{\beta}_L-1)ln(T_i)(1-T_i^{\hat{\alpha}_L})^{\hat{\beta_L}-2}}{T_i^{\hat{\alpha_L}-1}(1-T_i^{\hat{\alpha}_L})^{\hat{\beta}_L-1}} = 0 \tag{1.31}$$

$$\frac{n}{\hat{\beta}_L}+\sum_{i=1}^{n}\ln(1-T_i^{\hat{\alpha}_L}) = 0. \tag{1.32}$$

De la dernière équation on peut isoler,

$$\boxed{\hat{\beta}_L = \frac{-n}{\sum_{i=1}^{n}\ln(1-T_i^{\hat{\alpha}_L})}.} \tag{1.33}$$

Finalement on peut simplifier l'équation (1.31) et en injectant l'équation (1.32) on obtient à nouveau après plusieurs lignes de calcul

$$\boxed{\frac{1}{\hat{\alpha}_L}+\left(\frac{1}{\sum_{i=1}^{n}\ln(1-T_i^{\hat{\alpha}_L})}+\frac{1}{n}\right)\sum_{i=1}^{n}\frac{T_i^{\hat{\alpha}_L}\ln(T_i)}{1-T_i^{\hat{\alpha}_L}} = -\frac{1}{n}\sum_{i=1}^{n}\ln(T_i)} \tag{1.34}$$

1.3 Simulation

Le code du script `point_est.r` utilisé pour faire toutes les simulations associées à cette partie est accessible dans l'Annexe.

Question (a)

Voici les résultats obtenus :

$\hat{q}_S$	$\hat{q}_M$	$\hat{q}_L$
0.7067	0.7769	0.8643

TABLE 1 – *Valeur obtenues des trois estimateurs $\hat{q}_S$, $\hat{q}_M$, $\hat{q}_L$ issues des trois méthodes différentes. Valeurs obtenues par le script* `point_est.r` *disponible sur le GitHub.*

Le code `R` correspondant à cette question se trouve dans une sous-section de l'annexe.

Question (b)

Les histogrammes et boxplot de $\hat{q}_S, \hat{q}_M, \hat{q}_L$ sont représentés par les figures (1-6) et peuvent être trouvés en annexe du rapport. Après analyse de ceux-ci on peut conclure que :

- Les estimateurs $\hat{q}_S, \hat{q}_L$ et $\hat{q}_M$ ont une valeur moyenne proche de 0.8
- Les trois estimateurs ont une distribution légèrement asymétrique avec une queue plus longue pour les valeurs inférieures à leurs moyennes.
- Les valeurs de l'estimateur $\hat{q}_M$ sont légèrement plus concentrées autour de la valeur moyenne
- On voit pour les trois estimateurs au moins une valeur extrême très faible autour de 0.3 qui dépasse $1,5$ fois l'espace interquartile.

Question (c)

Toute la partie fonctionnelle du code `R` utilisée pour répondre à cette question a déjà été affichée à la question (b) dans l'annexe.
Sur les figures (1-6) les valeurs respectives du biais, de la variance et de la *mean squared error* (MSE) sont affichées et reprises dans le tableau suivant,

	$\hat{q}_S$	$\hat{q}_M$	$\hat{q}_L$
Biais	-0.1089	-0.051	-0.0534
Variance	0.0183	0.0129	0.0135
MSE	0.0301	0.0155	0.0163

TABLE 2 – *Biais, Variance et MSE pour les trois estimateurs $\hat{q}_S$, $\hat{q}_M$, $\hat{q}_L$ issues des trois méthodes différentes. Valeurs obtenues par le script* `point_est.r` *disponible sur le GitHub.*

- Les biais sont tous les 3 négatifs
- Étant donné que les MSE des 3 estimateurs sont assez faibles, on peut espérer que ce soient des estimateurs consistants et on pourra mieux regarder cela dans la prochaine sous-question en augmentant la taille des échantillons et en comparant leurs comportements asymptotiques.
- En terme de MSE, $\hat{q}_S$ semble être un estimateur moins efficace que les 2 autres estimateurs. $\hat{q}_L$ et $\hat{q}_M$ sont plus équivalents bien que $\hat{q}_M$ soit légèrement meilleur.

Question (d)

Pour obtenir les figures et les valeurs à analyser il suffit de changer la valeur de la variable `n` dans le script `point_est.r` déjà affiché en annexe.
Les figures (Fig(7)-Fig(15)) montrent l'évolution du biais, de la variance et de la MSE pour

différentes tailles d'échantillon pour chacun des estimateurs.
Pour comparer quel estimateur est le meilleur nous reprenons dans le tableau ci-dessous les valeurs de la MSE pour chacun des estimateurs pour les différentes valeurs de la taille de l'échantillon n.

MSE	$\hat{q}_S$	$\hat{q}_M$	$\hat{q}_L$
$n = 20$	0.0292	0.0173	0.0178
$n = 50$	0.0076	0.0051	0.0054
$n = 100$	0.0036	0.0020	0.0018
$n = 250$	0.0013	0.0007	0.0006
$n = 400$	0.0008	0.0005	0.0004

TABLE 3 – *MSE pour les trois estimateurs $\hat{q}_S$, $\hat{q}_M$, $\hat{q}_L$ issues des trois méthodes différentes pour des tailles d'échantillons différents. Valeurs obtenues par le script* `point_est.r` *disponible sur le GitHub.*

- En regardant les figures (Fig(7)-Fig(15)) on observe premièrement (que ce soit dans le cas de la variance, du biais ou de la MSE) que ces trois mesures tendent vers zéro quand la taille de l'échantillon augmente.
- Puisque la variance des trois estimateurs tend vers zéro pour des tailles d'échantillon plus grandes, cela implique que les valeurs des estimateurs sont de plus en plus concentrées autour de leurs valeurs moyennes.
- On observe que la MSE des trois estimateurs tend vers zéro puisque leur biais et leur variance tendent chacun vers zéro.
- Par définition, le fait que la MSE tende vers zéro suggère que nos trois estimateurs sont consistants.
- Vis à vis de la tendance dans la diminution de ces mesures, il semble que les trois grandeurs décroissent de manière inversement proportionnelle (en valeur absolue pour le biais) à n.
- Pour $n = 20$ on voit notamment dans le Tableau 3 que $\hat{q}_M$ a la MSE la plus petite. C'est également le cas pour $n = 50$. Toutefois à partir des données de $n = 100$ c'est $\hat{q}_L$ qui a la plus petite MSE.
- Comme indiqué dans le cours[1] nous pouvons définir le "meilleur" estimateur comme celui avec la MSE la plus faible. Donc pour les valeurs de $n = 20, 50$ le meilleur estimateur est $\hat{q}_M$ tandis que pour des valeurs de $n = 100, 250, 400$ c'est $\hat{q}_L$ qui est le meilleur estimateur.
- Ceci pouvait être attendu étant donné que les estimateurs construits par la méthode du maxima de vraisemblance sont particulièrement plus performants pour des échantillons de plus grande taille.
- On voit que l'estimateur $\hat{q}_S$ est le moins bon pour toutes les valeurs de n. Il garde malgré tout une MSE relativement faible à l'image des deux autres estimateurs construits par des méthodes plus complexes. Ceci se voit notamment dans la valeur finale de la MSE pour la taille d'échantillon $n = 400$.

Question (e)

Les figures (Fig(16)-Fig(18) montrent les trois histogrammes de $\sqrt{n}(\hat{q}_L - q_{0.95})$ pour différente taille d'échantillon.

- Les trois histogrammes sont concentrés autour de zéro ce qui est cohérent avec le fait que le biais tend vers zéro.
- On observe toutefois un comportement asymétrique (notamment pour $n = 20$) qui tend à diminuer lorsque n augmente.
- On peut remarquer que plus n augmente, plus le graphe de $\sqrt{n}(\hat{q}_L - q_{0.95})$ semble se rapprocher du graphe de la densité d'une normale $\mathcal{N}(0, \sigma^2)$ où σ^2 est un paramètre inconnu égal à la variance de q_l. Ce phénomène est dû au théorème central limite

qui implique que quand n augmente, la distribution de $\hat{q}_l - q_{0.95}$ tend vers $\mathcal{N}(\mathbb{E}(q_l - q_{0.95}, \mathbb{V}(q_l - q_{0.95}))$

2 Partie II - Régression

Le script `regression.r` utilisé pour analyser les données associée à cette partie se trouve dans l'Annexe du document. Il existe des fonctions déjà implémentées (comme la fonction `lm()`) dans `R` qui permettent rapidement et facilement d'obtenir les résultats d'une régression linéaire. L'objectif ici de refaire les calculs explicitement. Dans un dernier temps nous utiliserons aussi ces fonctions déjà implémentées afin de vérifier nos résultats.

2.1 Question (a)

Il nous est demandé d'ajuster un modèle de régression linéaire où Y - le logarithme de la consommation de carburant $[l/km]$ - est la variable de réponse et X - la puissance du moteur de la voiture considérée - est la variable explicative d'un échantillon de $n = 100$ voitures.
Autrement dit nous cherchons à formuler un modèle linéaire de la forme suivante,

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{2.1}$$

de paramètres β_0 et β_1 avec $\epsilon \sim \mathcal{N}(0, \sigma^2)$ le terme d'erreur.
On a $\sigma^2 = \mathbb{V}(\epsilon_i)$ qui n'est pas connu. Un estimateur est donnée par (preuve dans le cours),

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 = \frac{S_{yy} - \hat{\beta}_1 S_{xy}}{(n-2)} \tag{2.2}$$

où S_{yy} et S_{xx} sont tels que,

$$S_{yy} = \sum_{i=1}^{n}(Y_i - \bar{Y})^2. \tag{2.3}$$

$$S_{xx} = \sum_{i=1}^{n}(X_i - \bar{X})^2. \tag{2.4}$$

Des estimateurs $\hat{\beta}_0, \hat{\beta}_1$ des paramètres β_0 et β_1 peuvent être trouvé à l'aide de la méthode des moindres carrés ordinaires (MCO) comme vue en cours. On a alors (résultat démontré dans le cours) que,

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}, \tag{2.5}$$

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}, \tag{2.6}$$

avec les notations suivantes,

$$S_{xy} = \sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}), \tag{2.7}$$

Dans le début du script `regression.R` nous calculons explicitement les deux estimateurs de nos paramètres de la régression linéaire par la méthode MCO. Cela nous donne les valeurs estimées $\hat{\beta}_0 = -1.565$ et $\hat{\beta}_1 = 0.004$. Sur base des valeurs également calculées dans le script de S_{yy} et S_{xy}, nous obtenons la valeur suivante pour l'estimation de l'écart type de l'erreur : $\hat{\sigma} = 0.327$
Les données des coefficients de la corrélation linéaire obtenue sont les suivants,

$Slope = \hat{\beta}_1$	$Interception = \hat{\beta}_0$
0.0040152777099277115	−1.5649816030460726

TABLE 4 – Valeurs des coefficients de la corrélation linéaire sur le jeux de données `dataproject.txt`. Valeurs calculées dans le script `regression.r` disponible sur le GitHub.

Ce résultat dérivé manuellement est confirmé avec les mêmes valeurs de $\hat{\beta}_0$, $\hat{\beta}_1$ et $\hat{\sigma}$ si on utilise la fonction `lm()` déjà implémentée en `R`. La Fig.(20) montre le résumé de l'appel de cette fonction `lm()` de `R`.

Au niveau de l'interprétation, premièrement la pente positive de la régression linéaire nous indique qu'à mesure que les voitures ont plus de puissance dans l'ensemble de données, le logarithme de leur consommation (et donc leur consommation) augmente en même temps (bien que cette augmentation soit assez faible). Notons aussi que si l'on s'intéressait à une voiture de "zéro chevaux", on s'attendrait à une consommation nulle. Or la valeur de l'interception nous donnerait une consommation de 0.21l/100km pour cette voiture. Ceci montre que nous ne pourrions utiliser ce modèle pour interpoler la consommation d'une voiture ayant une toute petite puissance.

Nous pouvons déjà représenter graphiquement sur la Figure(19) le modèle de régression linéaire obtenu ici ainsi que le nuage de point correspondant. Visuellement nous pouvons voir que le modèle de régression linéaire est pertinent par rapport au jeu de données ici à disposition. De plus on observe que les données sont plus dispersées et moins bien représentées par le modèle linéaire pour les plus faibles valeurs de puissance des moteurs. Au contraire, pour les plus grandes puissances de moteur (plus de 700 hp) les données sont très faiblement dispersées autour du modèle linéaire.

2.2 Question (b)

Pour tester si notre modèle linéaire est pertinent par rapport aux données on fait un test statistique pour la pente. Il n'est pas nécessaire d'effectuer un test pour l'ordonnée à l'origine étant donné que si le test sur la pente n'est pas concluant, le modèle linéaire devrait de toute manière être rejetée indépendamment du résultat du test sur β_0. Nous considérons l'hypothèse nulle notée H_0 comme étant celle où le modèle linéaire n'est pas concluant et l'hypothèse alternative notée H_1 où le modèle linéaire est concluant. Pour le paramètre de la pente β_1 cela se formule comme suit,

$$H_0 : \beta_1 = 0, H_1 : \beta_1 \neq 0. \tag{2.8}$$

Nous sommes dans une situation où on ne connaît pas σ^2 et donc nous utilisons son estimation précédemment calculée $\hat{\sigma}^2$ ce qui nous conduit à choisir le test statistique suivant,

$$T_0 = \frac{\hat{\beta}_1 - 0}{\hat{\sigma}} \sqrt{S_{xx}} \sim t_{n-2} \tag{2.9}$$

Nous calculons la *p-value* comme étant la probabilité d'observer une valeur de notre test statistique au moins aussi extrême que celle observée, sous l'hypothèse H_0.

$$\mathbb{P}(|T_0| > |T_{obs}|; \beta_1 = 0) = 2\mathbb{P}(T_0 > |T_{obs}|; \beta_1 = 0) \tag{2.10}$$

En remplaçant dans la formule avec les valeurs déjà calculées, on trouve $T_{obs} = 28.09$. Étant donné que $T_0 \sim t_{n-2}$, sur base du code trouvable en annexe, on a :

$$2\mathbb{P}(T_0 > |T_{obs}|; \beta_1 = 0) = 2\mathbb{P}(T_0 < -|T_{obs}|; \beta_1 = 0) = 1.11 \cdot 10^{-48} \tag{2.11}$$

On a donc $p = 1.11 \cdot 10^{-48} < \alpha = 0.001$, ce qui signifie que β_1 est significativement différent de 0. Autrement dit, sur base des données à disposition, on rejette l'hypothèse que la modélisation linéaire ne soit pas bonne.

2.3 Question (c)

Pour rappel, notre modèle est donné par,

$$Y = \hat{\beta}_0 + \hat{\beta}_1 X + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \hat{\sigma}^2) \tag{2.12}$$

En utilisant les valeurs calculées au point (a) nous trouvons comme valeur prédite pour le logarithme de la consommation de carburant d'une voiture de $X^* = 1000$ chevaux : $Y^* = 2.450296$ $[l/100km]$.
Puisqu'on ne connait pas exactement σ^2 mais seulement un estimateur $\hat{\sigma}^2$, un intervalle de prédiction à 95% de confiance noté $IP_{95\%}$ est donné par,

$$IP_{95\%} = \left[\hat{\beta}_0 + \hat{\beta}_1 X^* \pm t_{n-2;\alpha/2}\hat{\sigma}\sqrt{1 + 1/n + \frac{(X^* - \bar{x})^2}{S_{xx}}}\right] \tag{2.13}$$

$$= \left[(-1.564) + (0.004).(1000) \pm (1.984)(0.327)\sqrt{1 + 1/100 + \frac{(1000 - 570.34)^2}{5\,242\,742.5}}\right] \tag{2.14}$$

$$= [1.786, 3.114] \tag{2.15}$$

La valeur prédite ainsi que les minimums et maximums de l'intervalle de prédiction à 95% sont affichés sur tab.5. On remarque que l'intervalle n'est très regroupé autour de la valeur *fit* : il est de taille 1.328 alors que le logarithme de la consommation d'essence varie entre -1 et 2. Nous rappelons la signification de l'intervalle de prédiction : contrairement à un intervalle de confiance à 95% qui contient avec une probabilité de 95% la vraie valeur de Y pour $X = 1000$, il contient avec une probabilité de 95% la valeur prédite de la réponse Y^*.

Dernièrement nous pouvons nous demander quelle est la qualité de notre ajustement des données par un modèle linéaire. Une mesure de cela est le coefficient $R^2 = \frac{S_{xy}^2}{S_{xx}S_{yy}}$ qui dans notre cas vaut $R^2 = 0.889$. Or au plus R^2 s'approche de 1 au plus la qualité de l'ajustement linéaire est bonne puisque R^2 représente la proportion de la variation totale dans les Y_i qui est expliqué dans la variable X dans un modèle linéaire simple. Autrement dit on peut stipuler qu'ici, 88.9% de la variance du logarithme de la consommation de carburant est expliqué par la puissance du moteur de la voiture considérée dans le cadre d'un modèle linéaire.

Prediction	fit	lower	upper
log(*fuel consumption*)	2.450296	1.786	3.114
fuel consumption	11.591777	5.966	22.511

TABLE 5 – Interpolation linéaire, bornes inférieures et supérieures de l'intervalle de confiance à 95% pour une voiture de 1000 chevaux.

3 Annexe

3.1 Code R

Un repository GitHub a été crée afin d'archiver le code R contenu dans deux scripts distincts `point_est.r` pour la première partie et *regression.r* pour la seconde. Le tout est accessible à l'adresse suivante : `https://github.com/AmauryLaridon/LMAT1271-Project`

Question 1.3.(a)

Voici la partie du code correspondant à cette question.

```r
#############################################################################
#                                                                           #
#                                  LMAT 1271                                #
#                                Projet 2022-2023                           #
#          Auteurs : Marie Determe, Augustin Lambotte, Amaury Laridon       #
#          Script R utiliser pour le projet instructions et ressources      #
#       disponibles à : https://github.com/AmauryLaridon/LMAT1271-Project   #
#                                                                           #
#############################################################################

rm(list=ls(all=TRUE)) # Nettoyage mémoire R Studio
set.seed(2023)

############################ - Partie 1.3 Simulations - ########################

n <- 400 # size of the sampling
nbr_replication <- 100 # number of replication of the sampling process
a_0 <- 0.5 # alpha parameter of the density function of T
b_0 <- 1.2 # beta parameter of the density function of T

#Density function of random variable T
f_a_0b_0 <- function(t) {
  if(t > 0 & t < 1) {
    return(a_0*b_0*t^(a_0-1)*(1-t^a_0)^(b_0-1))
  } else {
    return(0)
  }
}

#Quantile of order p, a and b are unknown parameters
q_p_a_b <- function(p,a,b) {
  return ((1-(1-p)^(1/b))^(1/a))
}

#Quantile of order p
q_p <- function(p) {
  return (q_p_a_b(p,a_0,b_0))
}

#First estimator of q_0.95
q_s <- function(T_i) {
  size <- length(T_i)
  T_sort <- sort(T_i)
  return (T_sort[ceiling(0.95*size)]+(T_sort[floor(0.95*size)]
                                      -T_sort[ceiling(0.95*size)])/2)
}

#Estimator of a_m and b_m, using the method of moments
q_m <- function(T_i) {
  #On cherche une racine à la première équation de 1.2.b pour obtenir
  #a_M, puis on calcule b_M et q_M
  f1 <- function(a) {
    return ((sum(T_i^(2*a)))*(n+sum(T_i^(a)))*(sum(T_i^(a)))^(-2)-2)
  }
  solution <- uniroot(f1,c(1e-9,10))
  a_m <- solution$root
  b_m <- n/(sum(T_i^(a_m)))-1
  return (q_p_a_b(0.95,a_m,b_m))
}

#Estimator of a_l and b_l, using the maximum likelihood estimator
q_l <- function(T_i) {
  #On cherche une racine à la première équation de 1.2.c pour obtenir a_L
  #puis on calcule b_L et q_L
  f1 <- function(a) {
    return (1/a + (1/(sum(log(1-T_i^a)))+(1/n))*sum((T_i^(a)
```

```
                                                    *log(T_i))/(1-T_i^(a)))+(1
    /n)*sum(log(T_i)))
  }
  solution_l <- uniroot(f1,c(1e-9,10))
  a_l <- solution_l$root
  b_l <- -n/(sum(log(1-T_i^(a_l))))

  return (q_p_a_b(0.95,a_l,b_l))
}

### - 1.3 (a) - ###

#Generation of a iid sample of size 20 from the density f_a_0b_0 and
#estimation of q_0.95 in 3 different ways
generate_sample <- function() {
  U <- runif(n)
  T_i <- sapply(U,q_p)
  q_s_sample <- q_s(T_i)
  q_m_sample <- q_m(T_i)
  q_l_sample <- q_l(T_i)
  return (c(q_s_sample,q_m_sample,q_l_sample))
}

######################## - Execution - ########################

#Vraie valeur de q_0.95
q_p_true <- q_p(0.95)

#1.3.a
generate_sample()
```

Question 1.3.(b)

Voici la partie du code correspondant à cette question (le code de la partie (c) est aussi inclue dans la fonction principale de cette partie du script).

```

### - 1.3 (b) - ###

generate_estimation_q_p <- function(affichage) {
  estimation_q_p <- replicate(nbr_replication,generate_sample())
  estimation_q_s <- estimation_q_p[1,]
  estimation_q_m <- estimation_q_p[2,]
  estimation_q_l <- estimation_q_p[3,]


  ### - 1.3 (c) - ###

  #Approximation de l'espérance par la moyenne empirique
  esperance_q_s <- mean(estimation_q_s)
  esperance_q_m <- mean(estimation_q_m)
  esperance_q_l <- mean(estimation_q_l)

  #Calcul du biais
  bias_s <- esperance_q_s - q_p_true
  bias_m <- esperance_q_m - q_p_true
  bias_l <- esperance_q_l - q_p_true

  #Calcul de la variance
  variance_s <- mean(estimation_q_s^2)-esperance_q_s^2
  variance_m <- mean(estimation_q_m^2)-esperance_q_m^2
  variance_l <- mean(estimation_q_l^2)-esperance_q_l^2

  #Calcul du MSE
```

```
  mse_s <- bias_s^2 + variance_s
  mse_m <- bias_m^2 + variance_m
  mse_l <- bias_l^2 + variance_l

  if (affichage == TRUE) {
    hist(estimation_q_s,breaks=10, main="")
    title(main = paste("Histogram of estimated q_s\n", "n =", n, " N =",
                       nbr_replication, " a_0 =", a_0, " b_0 =", b_0),
                       sub= paste("Biais =", round(bias_s, digits=4),
                       " Variance=", round(variance_s, digits=4), " MSE=",
                       round(mse_s, digits=4)))
    boxplot(estimation_q_s, xlab = "q_s", ylab="Value")
    title(main = paste("Boxplot of estimated q_s\n","n =", n, " N =",
                       nbr_replication, " a_0 =", a_0, " b_0 =", b_0),
                       sub= paste("Biais =", round(bias_s, digits=4),
                       " Variance=", round(variance_s, digits=4), " MSE=",
                       round(mse_s, digits=4)))


    hist(estimation_q_m,breaks=10, main = "")
    title(main = paste("Histogram of estimated q_m\n","n =", n, " N =",
                       nbr_replication, " a_0 =", a_0, " b_0 =", b_0),
                       sub= paste("Biais =", round(bias_m, digits=4),
                       " Variance=", round(variance_m, digits=4), " MSE=",
                       round(mse_m, digits=4)))
    boxplot(estimation_q_m, xlab = "q_m", main = "", ylab="Value")
    title(main = paste("Boxplot of estimated q_m\n","n =", n, " N =",
                       nbr_replication, " a_0 =", a_0, " b_0 =", b_0),
                       sub= paste("Biais =", round(bias_m, digits=4),
                       " Variance=", round(variance_m, digits=4), " MSE=",
                       round(mse_m, digits=4)))


    hist(estimation_q_l,breaks=10, main = "")
    title(main = paste("Histogram of estimated q_l\n","n =", n, " N =",
                       nbr_replication, " a_0 =", a_0, " b_0 =", b_0),
                       sub= paste("Biais =", round(bias_l, digits=4),
                        " Variance=", round(variance_l, digits=4), " MSE=",
                        round(mse_l, digits=4)))
    boxplot(estimation_q_l, xlab = "q_l" , main = "", ylab="Value")
    title(main = paste("Boxplot of estimated q_l\n","n =", n, " N =",
                       nbr_replication, " a_0 =", a_0, " b_0 =", b_0),
                       sub= paste("Biais =", round(bias_l, digits=4),
                       " Variance=", round(variance_l, digits=4), " MSE=",
                       round(mse_l, digits=4)))

  }

  return (c(c(bias_s,bias_m,bias_l),
            c(variance_s,variance_m,variance_l),c(mse_s,mse_m,mse_l)))
}

######################## - Execution - ##########################

#1.3.b et 1.3.c
generate_estimation_q_p(TRUE)
```

Question 1.3.(d)

```
### - 1.3.d - ###

n_value <- c(20,50,100,250,400)
results <- array(0, dim = c(3, 3, length(n_value)))
for (i in 1:length(n_value)) {
```

```
  n<-n_value[i]
  results[,,i]=t(generate_estimation_q_p(FALSE))
}

results
#Graphiques

print(results[3,,])

#Biais
plot(n_value,results[1,1,],xlab="n",ylab="Bias",main="Bias of q_s") #q_s
plot(n_value,results[2,1,],xlab="n",ylab="Bias",main="Bias of q_m") #q_m
plot(n_value,results[3,1,],xlab="n",ylab="Bias",main="Bias of q_l") #q_l

#Variance
plot(n_value,results[1,2,],xlab="n",ylab="Variance",main="Variance of q_s") #q_
    s
plot(n_value,results[2,2,],xlab="n",ylab="Variance",main="Variance of q_m") #q_
    m
plot(n_value,results[3,2,],xlab="n",ylab="Variance",main="Variance of q_l") #q_
    l

#MSE
plot(n_value,results[1,3,],xlab="n",ylab="MSE",main="MSE of q_s") #q_s
plot(n_value,results[2,3,],xlab="n",ylab="MSE",main="MSE of q_m") #q_m
plot(n_value,results[3,3,],xlab="n",ylab="MSE",main="MSE of q_l") #q_l
```

Question 1.3.(e)

```
### - 1.3.e - ###

n_values <- c(20,100,400)
for (i in 1:3) {
  n <- n_values[i]
  factor <- sqrt(n)
  estimation_q_p <- replicate(nbr_replication,generate_sample())
  estimation_q_l <- estimation_q_p[3,]
  hist(factor*(estimation_q_l-q_p_true),main=paste("Histogram of n^(1/2)(q_l-q_
    0.95) for n=",n),xlabel="n^(1/2)(q_l-q_0.95)")
}
```

Question 2.(a)

```
#############################################################################
#                                                                           #
#                                  LMAT 1271                                #
#                               Projet 2022-2023                            #
#          Auteurs : Marie Determe, Augustin Lambotte, Amaury Laridon       #
#          Script R utiliser pour le projet instructions et ressources      #
#        disponibles à : https://github.com/AmauryLaridon/LMAT1271-Project  #
#                                                                           #
#############################################################################

rm(list=ls(all=TRUE)) # Nettoyage mémoire R Studio
set.seed(2023) # génération clef aléatoire

##### Chargement librairies #####
#install.packages("lattice")
#install.packages("ggplot2")
#install.packages("esquisse")
```

```r
library(lattice)
library(ggplot2)
#library(esquisse)

########################## - Partie 2 Regression - ##########################

### Importation des données ###

dataproject <- read.csv("/home/amaury/Bureau/LMAT1271 - Calcul des probabilités et analyse statistique/Projet/LMAT1271-Project/Data/dataproject.txt", sep=";")
#View(dataproject)
fuel <- dataproject$Y
hrspw <- dataproject$X
n <- length(fuel)

### Question (a) ###

mean_hp <- mean(hrspw)
mean_fuel <- mean(fuel)


## Calcul de S_xx ##

S_xx <- sum((hrspw - mean_hp)^2)

## Calcul de S_xy ##

S_xy <- sum((hrspw-mean_hp)*(fuel-mean_fuel))

## Calcul de S_yy ##

S_yy <- sum((fuel-mean_fuel)^2)

## Calcul des estimateurs de beta_0 et beta_1 à l'aide de leur définition par la méthode MCO ##

beta_1 <- S_xy/S_xx
beta_0 <- mean_fuel - beta_1*mean_hp
print(beta_0)
print(beta_1)

## Calcul de l'estimateur de la variance des erreurs ##

var_err <- (S_yy-(beta_1*S_xy))/(n-2)
std_err <- sqrt(var_err)

## Création du modèle linéaire sur base des paramètres estimés ##

lm <- beta_0+(beta_1*hrspw)
```

Question 2.(b)

```r
### - 2.b - ###

T_obs <- ((beta_1 - 0)/(std_err))*sqrt(S_xx)
p_value <- 2*(pt(-T_obs,df=98))
```

Question 2.(c)

```r
## Calcul des estimateurs de la variance de beta_0 et beta_1 à l'aide de leur définition par la méthode MCO ##

std_beta_1 <- var_err/S_xx

## Calcul de la statique de test T_obs ##

T_obs <- ((beta_1 - 0)/(std_err))*sqrt(S_xx)

## Prévision ##

X.star <- 1000
Y.star <- beta_0 + beta_1*X.star

## Intervalle de prévision à 95% ##

alpha <- 0.05
t <- qt(p=alpha/2, df=n-2, lower.tail = F)

IP_plus <- beta_0+(beta_1*X.star)+(t*std_err)*sqrt(1+(1/n)+((X.star-mean_hp)^2)/S_xx)
IP_min <- beta_0+(beta_1*X.star)-(t*std_err)*sqrt(1+(1/n)+((X.star-mean_hp)^2)/S_xx)

## Calcul p valeur ##

p_value <- 2*(pt(-T_obs,df=98))
print(p_value)

## Affichage ##

xyplot(fuel~hrspw,xlab="Puissance [hP]", ylab="Ln(Consommation d'essence) [l/100km]")
xyplot((beta_0+(beta_1*hrspw))~hrspw)

## Verification à l'aide de la fonction déjà implémentée lm() ##

lm1 <- lm(formula = fuel~hrspw, data = dataproject) #Calcul et création d'un modèle de regression
                                    #linéaire
summary(lm1)

model <- lm(fuel ~ hrspw)
new <- data.frame(hrspw=1000)
predict(lm1,new, interval="confidence")
```

Figures

Toutes les figures utilisées dans ce rapport se trouve également sur le repository GitHub.

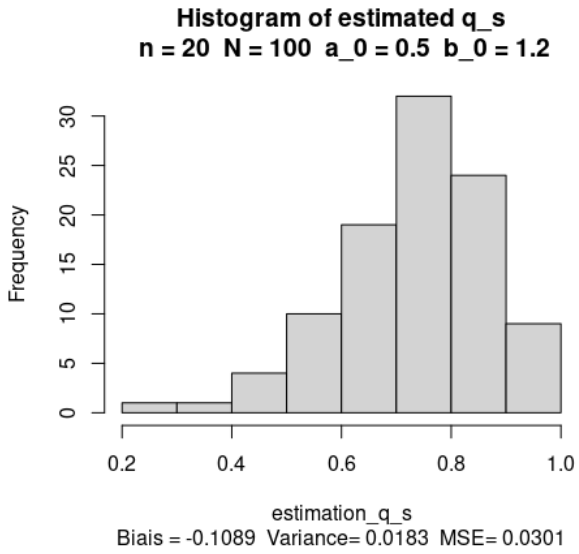

FIGURE 1 – *Histogramme de $\hat{q}_S$ pour $n = 20$. Figure disponible sur le Github produite avec le script* `point_est.r`.

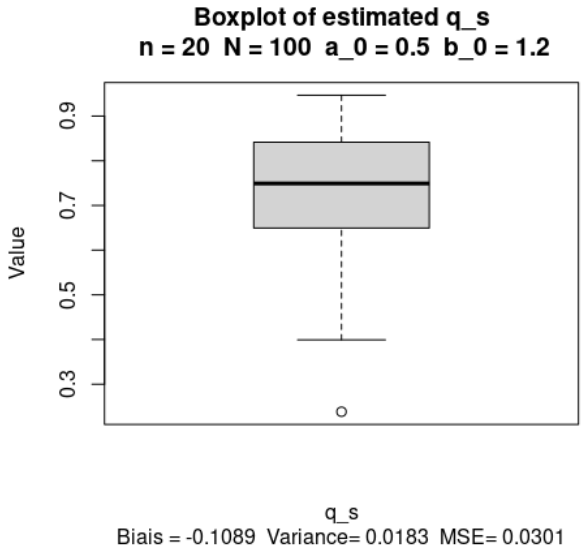

FIGURE 2 – *Boxplot de $\hat{q}_S$ pour $n = 20$. Figure disponible sur le Github produite avec le script* `point_est.r`.

Références

[1] Von Sachs R. *LMAT1271 - Calcul de probabilité et analyse statistique.* UCLouvain, 2023.

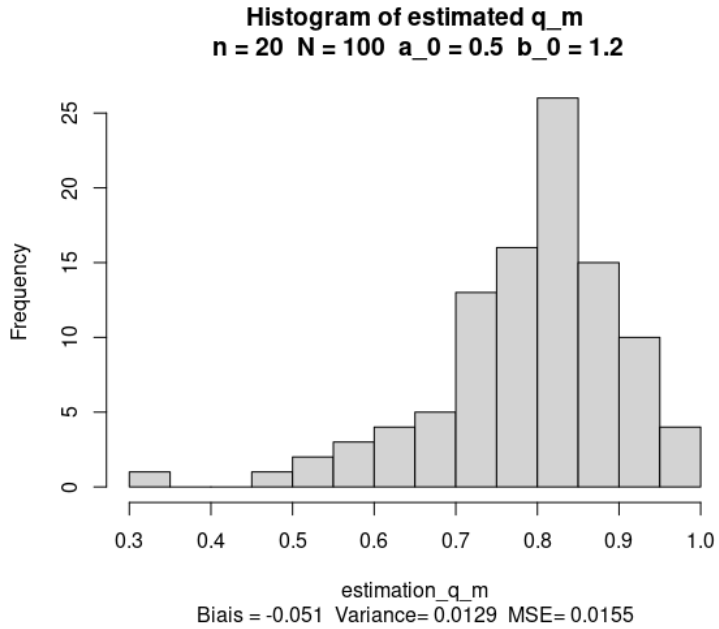

FIGURE 3 – *Histogramme de $\hat{q}_M$ pour $n = 20$. Figure disponible sur le Github produite avec le script* `point_est.r`.

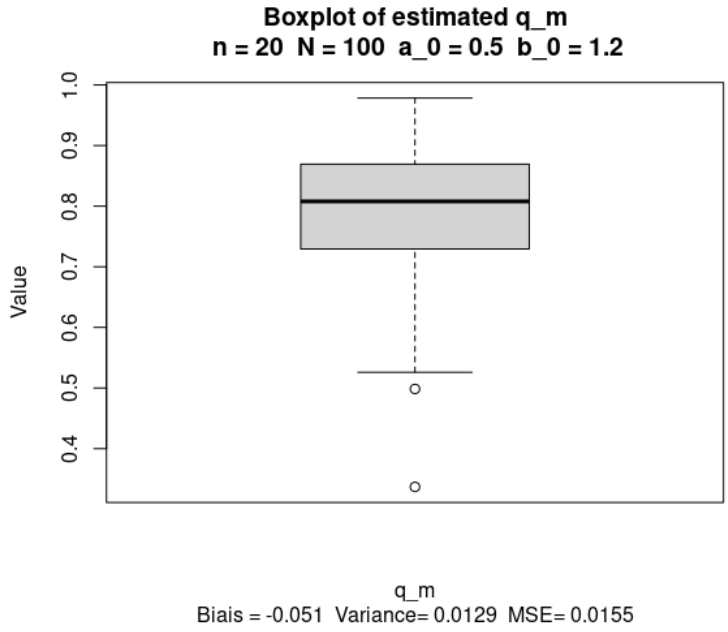

FIGURE 4 – *Boxplot de $\hat{q}_M$ pour $n = 20$. Figure disponible sur le Github produite avec le script* `point_est.r`.

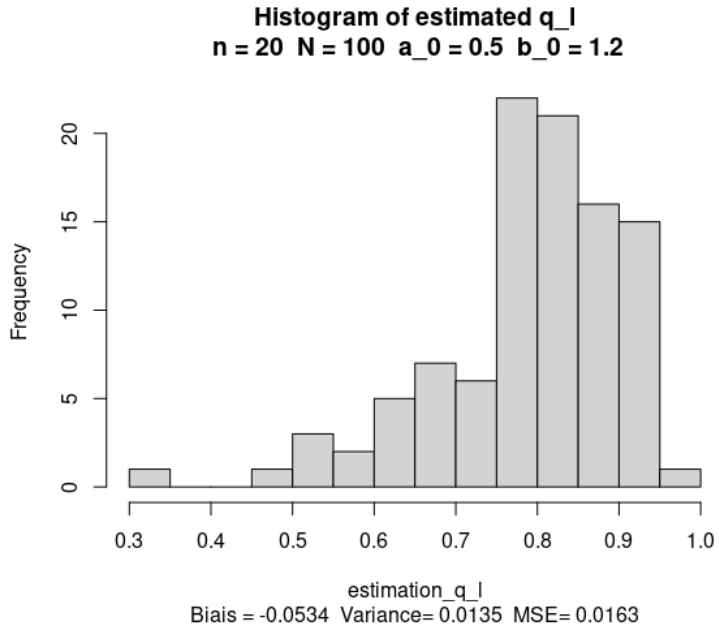

FIGURE 5 – *Histogramme de $\hat{q}_L$ pour $n = 20$. Figure disponible sur le Github produite avec le script* `point_est.r`.

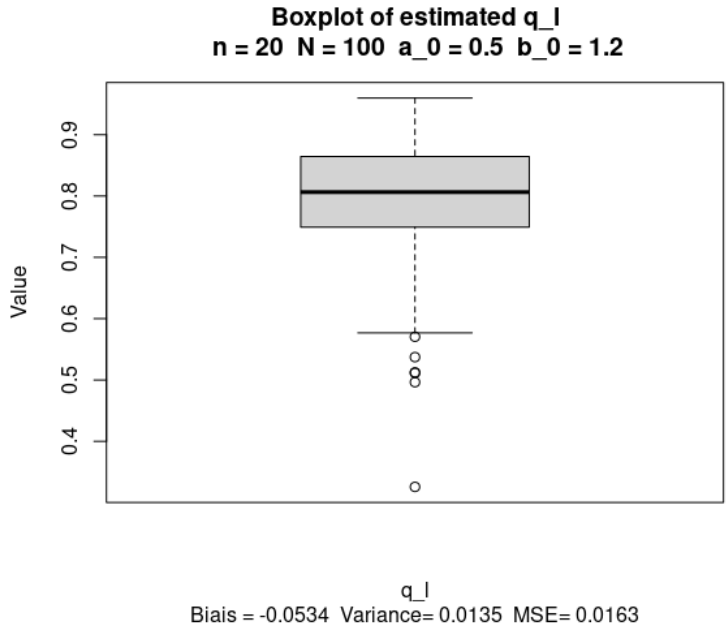

FIGURE 6 – *Boxplot de $\hat{q}_L$ pour $n = 20$. Figure disponible sur le Github produite avec le script* `point_est.r`.

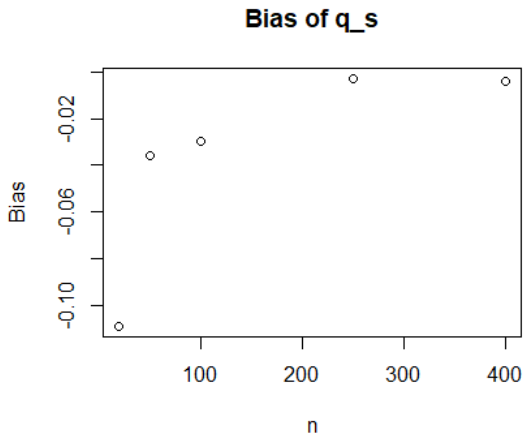

FIGURE 7 – *Biais de $\hat{q}_S$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

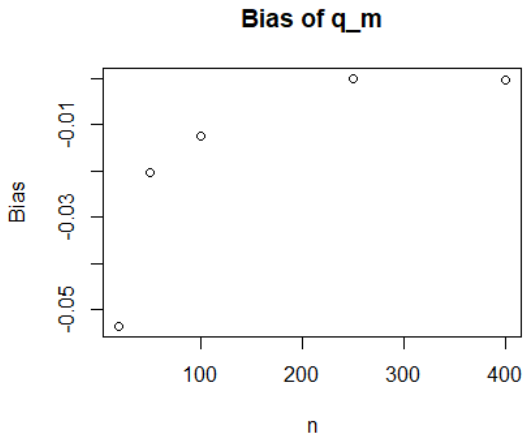

FIGURE 8 – *Biais de $\hat{q}_M$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

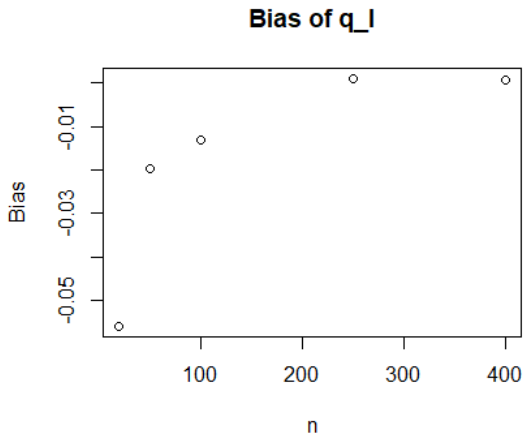

FIGURE 9 – *Biais de $\hat{q}_L$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

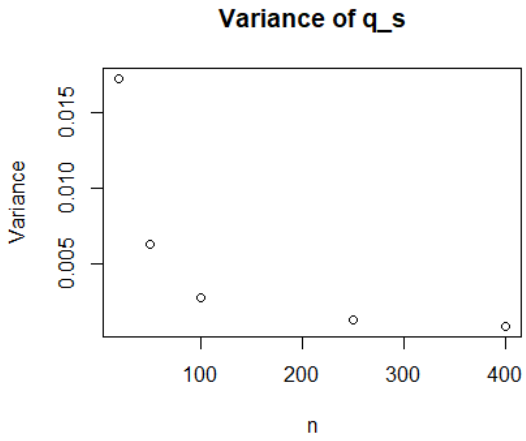

FIGURE 10 – *Variance de $\hat{q}_S$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

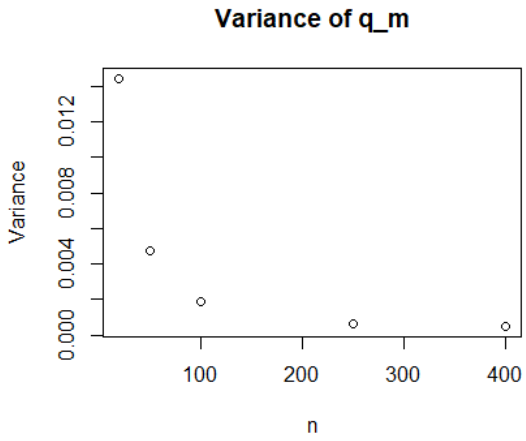

FIGURE 11 – *Variance de $\hat{q}_M$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

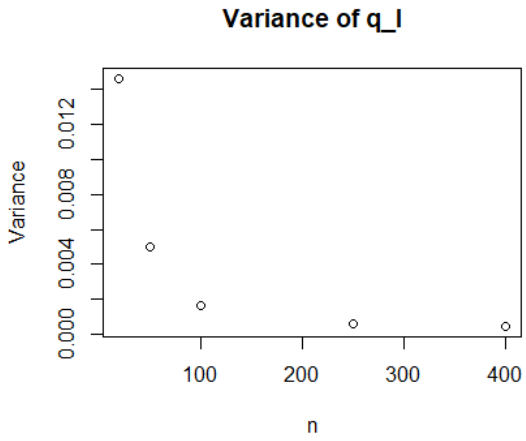

FIGURE 12 – *Variance de $\hat{q}_L$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

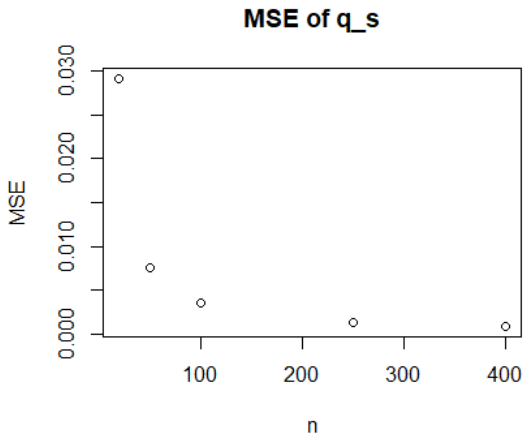

FIGURE 13 – *MSE de $\hat{q}_S$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

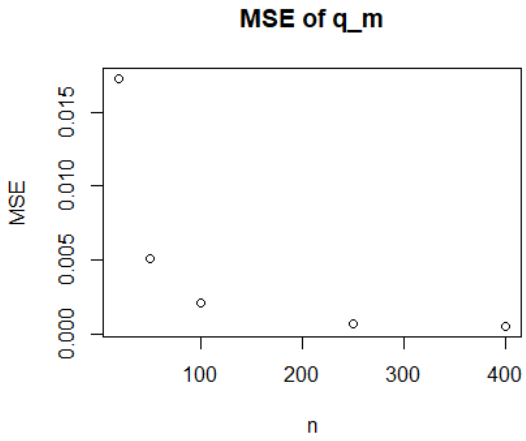

FIGURE 14 – *MSE de $\hat{q}_M$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

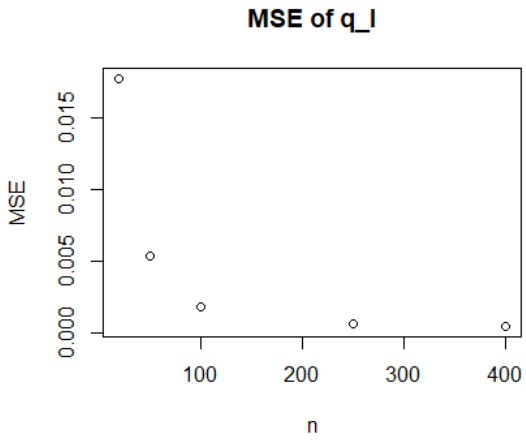

FIGURE 15 – *MSE de $\hat{q}_L$ en fonction de la taille de l'échantillon n. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

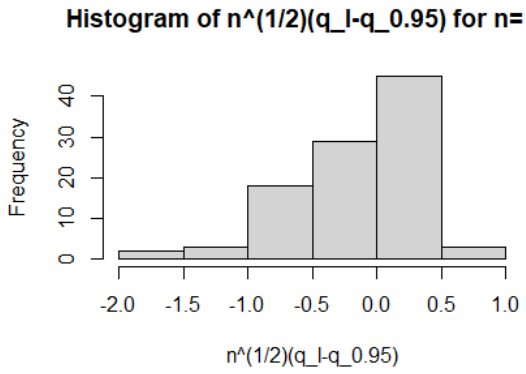

FIGURE 16 – *Histogramme de $\sqrt{n}(\hat{q}_L - q_{0.95})$ pour $n = 20$. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

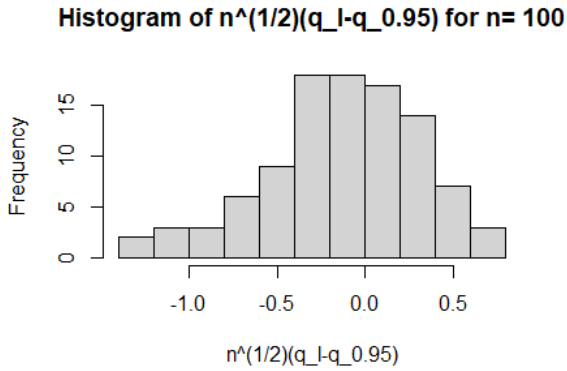

FIGURE 17 – *Histogramme de $\sqrt{n}(\hat{q}_L - q_{0.95})$ pour $n = 100$. Figure disponible sur le GitHub et produite avec le script* `regression.r`.

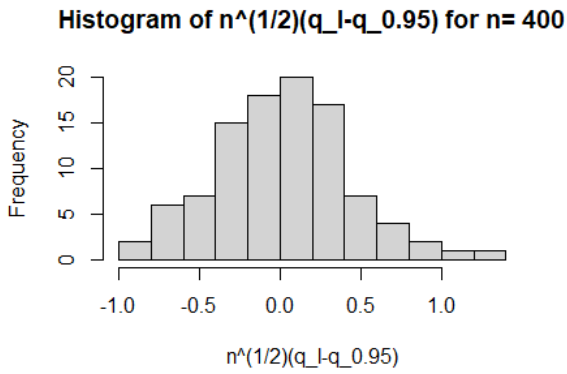

FIGURE 18 – *Histogramme de* $\sqrt{n}(\hat{q}_L - q_{0.95})$ *pour* $n = 400$. *Figure disponible sur le GitHub et produite avec le script* `regression.r`.

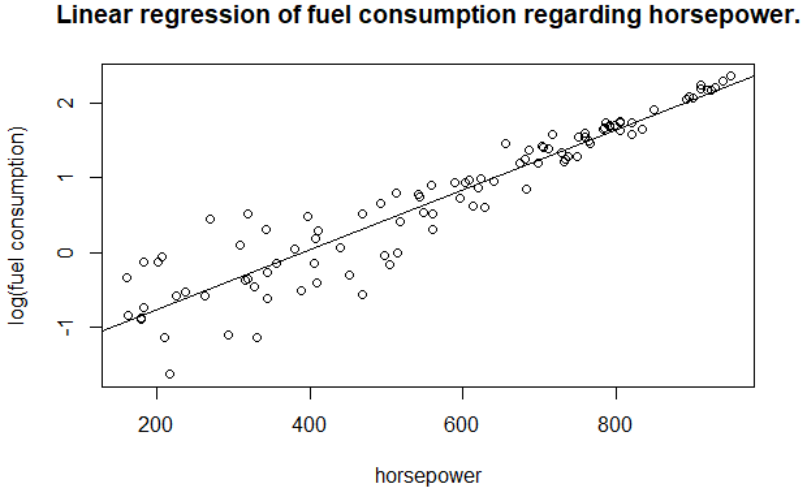

FIGURE 19 – *Nuage de points des données* `dataproject.txt` *et du modèle de régression linéaire obtenu avec la méthode MCO. Figure produite avec le script* `regression.r` *et disponible sur le GitHub.*

```
lm(formula = fuel ~ hrspw, data = dataproject)

Residuals:
     Min       1Q   Median
-0.93947 -0.10972  0.05427
      3Q      Max
 0.11218  0.93006

Coefficients:
             Estimate Std. Error
(Intercept) -1.5649816  0.0878357
hrspw        0.0040153  0.0001429
            t value Pr(>|t|)
(Intercept)  -17.82   <2e-16 ***
hrspw         28.09   <2e-16 ***
---
Signif. codes:
  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’
  0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3272 on 98 degrees of freedom
Multiple R-squared:  0.8896,    Adjusted R-squared:  0.8884
F-statistic: 789.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

FIGURE 20 – *Résumé de l'appel de la fonction* `lm()` *sur base du jeux de donnée. Figure produite avec le script* `regression.r` *et disponible sur le GitHub.*