

7	abus.subst	-0.143	0.28	6.1e-01	0.089
8	determination	0.902	0.31	5.5e-03	0.345
9	factor(type.centre)2	0.650	0.41	1.2e-01	0.050
10	factor(type.centre)3	0.824	0.38	3.0e-02	0.048

La fonction `with(glm(...))` ❶ estime le même modèle logistique sur chacun des 5 jeux de données imputées contenus dans `smp.mice`. La fonction `pool()` fait une synthèse des 5 séries de résultats. Parmi les résultats disponibles il y a : les log *odds-ratios* moyens ❷, leurs erreurs types obtenues à partir de la somme des variances intra-modèles et des variances inter-modèles ❸. La FMI (*Fraction of Missing Information*) quantifie, pour chaque variable, la part de la variance due aux données manquantes.

Le *bootstrap*

En quelques mots : dans un modèle logistique, les « p » ainsi que les intervalles de confiance des log *odds-ratios* sont calculés à partir des « dérivées partielles secondes de la log vraisemblance », calculs dont la fiabilité est en principe garantie pour des tailles d'échantillons tendant vers l'infini. Voilà qui n'est, pour le moins, ni intuitif, ni rassurant.

Il existe une alternative à cette approche, beaucoup moins mathématique, mais beaucoup plus calculatoire. L'idée est la suivante : une façon de conceptualiser l'intervalle de confiance d'un paramètre (à 95% par exemple), c'est d'imaginer que l'on réplique l'étude qui a conduit à l'échantillon analysé un grand nombre de fois, de calculer pour chaque réplique le paramètre en question, puis d'écarter les 2,5% des valeurs les plus basses et les 2,5% des valeurs les plus élevées ; on obtiendrait ainsi un intervalle de fluctuations, dont la largeur devrait être une bonne référence pour le calcul d'un intervalle de confiance.

Bien sûr il est impensable de *vraiment* répliquer l'étude. Il est cependant possible de le faire virtuellement. L'astuce consiste à partir de l'échantillon de départ E_0 . Si E_0 compte n sujets, alors un nouvel échantillon E_1 va être constitué en tirant au sort avec remise n sujets dans E_0 (« avec remise » signifie qu'un même sujet de E_0 peut être tiré au sort plusieurs fois ; E_1 est donc presque toujours différent de E_0). Un échantillon E_2 va être également constitué de la sorte, idem pour E_3 , E_4 , ..., E_k (k « grand », $k = 10000$ par exemple). La statistique d'intérêt (par exemple le log *odds-ratio*) va ensuite être calculée dans chaque E_i , puis les 2,5% des valeurs les plus basses et les 2,5% des valeurs les plus élevées de cette statistique vont être écartées et les valeurs restantes délimiteront un intervalle de confiance à 95% ⁽¹⁾. Cette technique est dénommée *bootstrap* ⁽²⁾.

¹ Ici, la méthode dite « percentile » du *bootstrap* est présentée ; il existe d'autres variantes pour le calcul de l'intervalle de confiance, notamment la méthode BCa qui lui est parfois préférable ; cependant la méthode percentile reste la plus simple et est assez robuste.

² En référence au baron de Münchhausen qui parvint à s'extirper de sables mouvants en tirant fort sur les lacets de ses bottes.

Le *bootstrap* présente deux avantages importants : il ne repose pas sur des considérations mathématiques impénétrables et il relève d'une approche non paramétrique beaucoup plus robuste que l'approche de dérivation seconde du maximum de vraisemblance, notamment en cas de violation des hypothèses des modèles statistiques utilisés (additivité des effets, linéarité, etc.). Comme la plupart des méthodes statistiques, le *bootstrap* repose cependant très fortement sur l'indépendance des observations ré-échantillonnées. Les « observations » ré-échantillonnées peuvent cependant correspondre à un grappe (*cluster*) de données corrélées entre elles, telles que l'ensemble des mesures répétées chez un même patient ou dans un même centre.

En pratique : dans la section sur le modèle linéaire, p. 123, nous avons estimé un modèle expliquant la variable « durée de l'entretien » réalisée lors de l'étude santé mentale en prison. En vérifiant les conditions de validité de ce modèle, nous avons constaté graphiquement que les résidus s'écartaient quelque peu de la normalité, ce qui faisait planer un léger doute sur la validité des résultats obtenus :

```
> mod <- lm(duree.interv~schizophrenie+depression+abus.subst+
             gravite+caractere+trauma.enfant+age+
             factor(type.centre), data=smp)

> summary(mod)
[...]
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	42.4900	4.0000	10.62	< 2e-16 ***
schizophrenie	3.0642	2.8100	1.09	0.27591 ❶
depression	6.7127	1.6479	4.07	5.2e-05 ***
abus.subst	4.6004	1.7985	2.56	0.01076 *
gravite	1.0624	0.5655	1.88	0.06074 .
caractere	1.6255	0.9354	1.74	0.08272 .
trauma.enfant	-0.6681	1.6693	-0.40	0.68914
age	0.2079	0.0607	3.43	0.00065 ***
factor(type.centre)2	4.0954	2.5340	1.62	0.10655
factor(type.centre)3	-1.2968	2.4416	-0.53	0.59551

```
[...]
```

	2.5 %	97.5 %
(Intercept)	34.635	50.34
schizophrenie	-2.454	8.58 ❷
depression	3.477	9.95
abus.subst	1.069	8.13
gravite	-0.048	2.17
caractere	-0.211	3.46
trauma.enfant	-3.946	2.61
age	0.089	0.33
factor(type.centre)2	-0.881	9.07
factor(type.centre)3	-6.091	3.50

Voyons ce que donnerait une estimation par *bootstrap* en commençant, pour simplifier, par négliger la dépendance des observations d'un même centre :

```

> library(boot)

> lm.boot <- function(data, index) {
  smp.boot <- data[index,]
  mod <- lm(duree.interv ~ schizophrénie + depression +
    abus.subst + gravite + caractere + trauma.enfant +
    age + factor(type.centre), data = smp.boot)
  coefficients(mod)
}
> set.seed(10)
> resboot <- boot(smp, lm.boot, 10000)
> hist(resboot$t[,2], breaks=40, main="",
  xlab="Regression coefficient of variable scz.cons")
> box()

```

L'étape la plus délicate d'une procédure de *bootstrap* programmée en R concerne la définition de la fonction qui estime les paramètres d'intérêt calculés pour chaque jeu de données répliqué. Cette fonction a deux arguments : le premier ❸ définit le nom du jeu de données, le second ❹ repère les observations qui devront être tirées au sort avec remise (il s'agit presque toujours des lignes du jeu de données, d'où la syntaxe ❺). Le modèle linéaire est ensuite estimé et les coefficients obtenus stockés dans ❻.

La fonction `boot()` est ensuite appelée avec, successivement, le nom du jeu de données original, la fonction à « bootstrapper » définie précédemment ainsi que le nombre de réplifications. Si l'on est plus spécialement intéressé par la variable « schizophrénie » il faut choisir le deuxième paramètre du tableau de résultats ❸ (la variable était la première à apparaître dans le modèle, le premier coefficient correspondant à l'ordonnée à l'origine ou *intercept*). L'histogramme des 10000 estimations du coefficient de la variable `schizophrénie` est présenté Fig. 2.28.

L'intervalle de confiance à 95% ainsi que le « p » du test de significativité s'obtiennent très simplement :

```

> boot.ci(resboot, index=2, type= "bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = resboot, type = "bca", index = 2)

Intervals :
Level      BCa
95%      (-1.7,  7.9 )
Calculations and Intervals on Original Scale
> 2*sum(resboot$t[,2]<=0)/10000
[1] 0.22

```

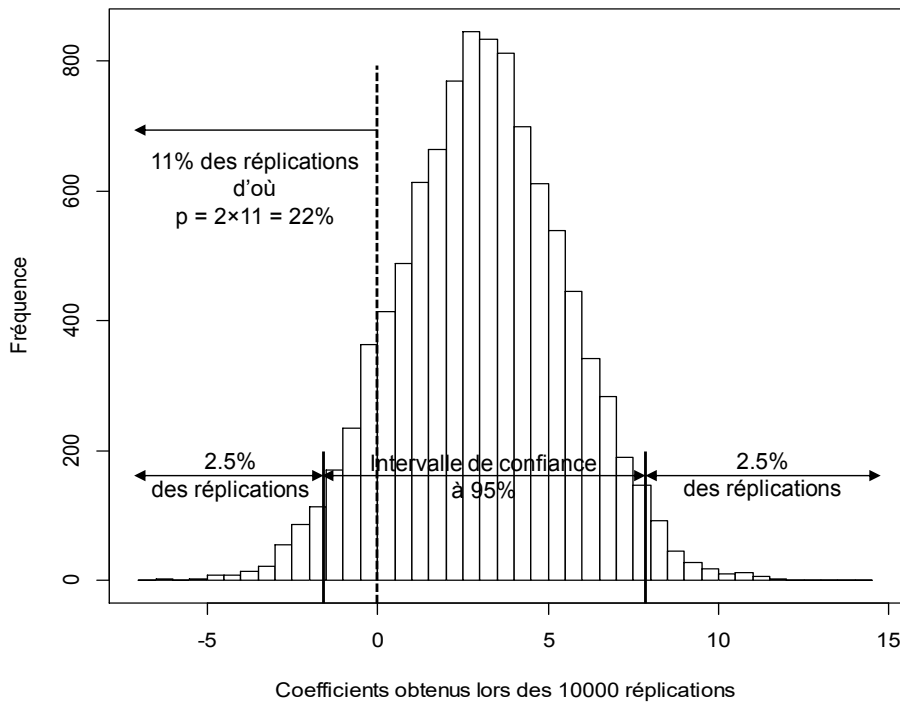


Fig. 2.28 — Histogramme des 10000 coefficients de régression de la variable « schizophrénie » estimés via bootstrap, c'est-à-dire issus de 10000 jeux de données obtenus par tirage au sort avec remise à partir du jeu de données original.

L'instruction `index=2` ❶ précise que c'est la variable « schizophrénie » qui nous intéresse. L'estimateur naïf de l'intervalle de confiance tel que suggéré sur la Fig. 2.28 étant légèrement biaisé sur des petits échantillons, il lui est préféré une variante que l'on obtient avec l'instruction ❷.

L'intervalle estimé ici est égal à $[-1,7 ; 7,9]$ ❸, il est proche de celui obtenu par dérivation des dérivées seconde de la log vraisemblance : $[-2.454 ; 8.58]$ (voir ❷ au niveau des résultats précédents). Le « p », obtenu selon la méthode percentile vaut, quant à lui, 0,22 ❹, ce qui est proche de la valeur 0,275 obtenue précédemment ❶.

Nous pouvons affiner le modèle en prenant en compte la dépendance intra-centre des observations ; pour cela, nous allons ré-échantillonner les centres plutôt que les sujets :

```
> # on définit le jeu de colonnes utiles pour alléger
> # le temps de calcul
> cols <- c("duree.interv", "schizophrenie", "depression",
            "abus.subst", "gravite", "caractere",
            "trauma.enfant", "age", "centre")
> # on crée une liste de data frames
> centres <- split(smp[,cols], smp$centre) ❶
> length(centres) ❶
[1] 20
```

```

> lm.boot.centre <- function(centres, index) {
  centres2 = centres[index] ❷
  for(i in 1:length(centres)) {
    # on renomme les centres pour qu'
    # un centre sélectionné deux fois
    # ait deux numéros différents
    centres2[[i]]$centre = i ❸
  }
  # on colle toutes les données des centres
  # en un seul data frame
  data <- do.call(rbind, centres2)
  mod <- lm(duree.interv ~ schizophrénie + depression +
    abus.subst + gravite + caractere + trauma.enfant +
    age + factor(centre), data = data)
  # on renvoie la durée moyenne d'intervention
  # et l'effet de la schizophrénie
  c(moyenne=mean(data$duree.interv, na.rm=TRUE),
    coefficients(mod) ["schizophrénie"]) ❹
}
> set.seed(10)
> ctrboot <- boot(data=centres, lm.boot.centre, R=10000)
> # on compare les intervalles de confiance de la
> # durée moyenne d'intervention entre la méthode
> # naïve et le bootstrap
> boot.ci(ctrboot, index=1, type="bca")
[...
95%      (56, 68 ) ❺
[...
> t.test(smp$duree.interv)$conf.int
[1] 61 64 ❺
attr(,"conf.level")
[1] 0.95
>
> # et pour l'effet de la schizophrénie sur la durée
> confint(lm(duree.interv ~ schizophrénie + depression +
  abus.subst + gravite + caractere + trauma.enfant +
  age + factor(centre), data = smp)
, parm="schizophrénie")
      2.5 % 97.5 %
schizophrénie  0.17    8.4 ❻
> c(ctrboot$t0[2], boot.ci(ctrboot, index=2, type="bca")$bca[4:5])
schizophrénie
      4.3      1.9      9.2 ❻
> v <- ctrboot$t[,2]
> hist(v, breaks=100, xlab="Effet schizophrénie",
  ylab="Nombre d'échantillons de bootstrap", main="",
  xlim=c(-5,15)) ❼
> qqnorm(v); qqline(v); abline(v=c(-2,2)) ❼

```

Le code est plus lourd, et pourrait être remplacé par un appel direct à la fonction `clusbootglm` de la bibliothèque `ClusterBootstrap`, mais illustre la manière générale de procéder pour des statistiques plus complexes.

En ❶, le jeu de données est segmenté par centres, conduisant à une liste de longueur 20 (nombre de centres) contenant 20 petits *data frames*, contenant chacun les observations du centre correspondant. Le nombre de centres est un peu faible pour garantir avec certitude la validité du *bootstrap*, mais cela peut être encore acceptable si l'effet « centre » est modeste. En effet, il faut suffisamment d'observations (ici les centres) pour que le tirage au sort avec remise simule correctement la distribution recherchée. La fonction de calcul statistique est ensuite définie, sélectionnant spécifiquement les centres qui ont été échantillonnés par le tirage au sort ❷ et les renumérotant ❸. Après avoir estimé le modèle avec un ajustement sur l'effet centre (en effet fixe), un vecteur numérique comprenant deux valeurs ❹ est renvoyé : la durée moyenne d'interview et l'effet de la schizophrénie sur la durée d'interview dans le modèle linéaire.

Après avoir lancé la procédure de *bootstrap* et récupéré les résultats en ❺, on calcule l'intervalle de confiance de la durée moyenne d'interview, d'une part avec la *bootstrap* prenant en compte la corrélation intra-centre, et d'autre part, avec la méthode naïve de Student ignorant cette corrélation. La différence de largeur d'intervalle de confiance est impressionnante car il existe un effet centre majeur sur la durée d'interview, possiblement du fait d'un effet caché de l'interviewer (variable manquante), mais aussi peut-être du fait du cadre et des conditions imposées par les agents du centre. L'hypothèse d'indépendance des centres sur laquelle repose ces calculs pourrait être remise en cause si un même interviewer était susceptible de visiter plusieurs centres, auquel cas la variable « interviewer » aurait dû être renseignée et prise en compte dans la procédure de *bootstrap*. Enfin, en ❻ on calcule les intervalles de confiance naïf et « bootstrappé » de l'effet ajusté de la schizophrénie sur la durée d'interview. La différence entre les deux largeurs d'intervalles de confiance est très modeste parce qu'il n'y a probablement pas d'interaction majeure entre le centre et la schizophrénie sur la durée d'interview, c'est-à-dire que l'effet de la schizophrénie n'est pas hétérogène entre les centres.

Pour terminer, on vérifie sur un histogramme ainsi que sur un Q-Q plot de normalité les fluctuations empiriques d'échantillonnages de l'effet moyen de la schizophrénie sur la durée d'interview ❼. Cet effet moyen présente une modeste mais non-négligeable asymétrie à droite (Fig. 2.29) avec une approximation normale qui ne serait pas bien valide entre -2 et $+2$ erreurs types, ce qui explique l'asymétrie de l'intervalle de confiance BCa ❽ ; même si la méthode BCa est conçue pour tolérer une asymétrie assez nette, celle-ci, lorsqu'elle est forte, peut faire douter de la validité du *bootstrap*.

Les conditions de validité du *bootstrap* sont assez légères en théorie :

- 1/ les observations ré-échantillonnées (bootstrappées) doivent être indépendantes ;
- 2/ leur nombre doit être suffisamment grand pour que les fluctuations de ré-échantillonnages empiriques s'approchent assez des fluctuations d'échantillonnages de la population ;
- 3/ les fluctuations d'échantillonnages ne doivent pas être discrètes et
- 4/ doivent être soumises à une stabilisation asymptotique.

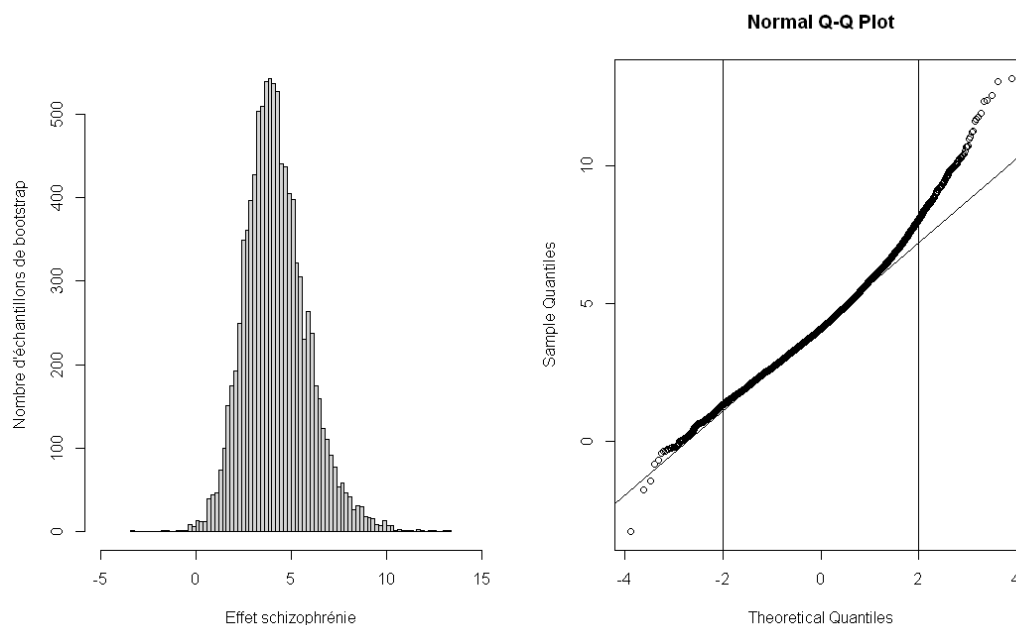


Fig. 2.29 — Histogramme et Q-Q plot de normalité des 10000 coefficients de régression de la variable « schizophrénie » estimés via bootstrap des centres, c'est-à-dire issus de 10000 jeux de données obtenus par tirage au sort avec remise des centres dans le jeu de données original.

La première condition, nécessaire pour presque tous les outils statistiques n'est généralement vérifiable que par la connaissance de la méthodologie d'échantillonnage. Les sources de corrélation cachées, telles que l'identité de l'interviewer dans notre analyse de la durée d'interview, ne sont rectifiables a posteriori que si les variables nécessaires avaient été préalablement recueillies.

La seconde condition (échantillon suffisamment grand) est bien plus complexe à apprécier même si quelques points de repères peuvent être fournis et que le *bootstrap* facilite les vérifications a posteriori des conditions de validité. Pour des statistiques simples, telles que les estimations d'un modèle linéaire, une trentaine d'observations suffit souvent, à condition qu'il n'y ait pas d'*outlier* influent, c'est-à-dire d'observations suffisamment extrême pour influencer de manière substantielle la statistique d'intérêt ⁽¹⁾.

¹ Malgré le fait que ces observations soient peu fréquentes, leur caractère extrême compenserait leur rareté. Elles sont à distinguer des *outliers* non influents dont la rareté est plus forte que le caractère extrême. Le pire cas de figure est relatif aux *outliers* influents invisibles, c'est-à-dire n'apparaissant aucune fois dans l'échantillon bien qu'existant dans la population. Les outils de validation *a posteriori* des conditions de validité peuvent alors transmettre un faux signal rassurant alors que, puisque l'*outlier* est absent de l'échantillon, il ne contribue plus aux fluctuations d'échantillonnages empiriques du *bootstrap* qui sous-estiment alors les vraies fluctuations d'échantillonnages de manière substantielle.

Un exemple caricatural serait l'estimation de la rentabilité moyenne d'une assurance basée sur un échantillon de seulement une trentaine d'assurés dont aucun n'aurait eu de sinistre. La moyenne est pourtant très influencée par les rares assurés qui ont des sinistres onéreux pour l'assurance ; leur absence dans l'échantillon ne doit pas en faire oublier l'existence. Le cas des variables binaires dont la proportion est proche de zéro est assez semblable, puisque seule une petite fraction des valeurs (par exemple quelques pourcents) est entièrement responsable de la valeur non-nulle de la moyenne.

Le *bootstrap* est adapté aux variables dont les fluctuations d'échantillonnages sont continues et non discrètes. Par exemple, une médiane calculée sur une variable discrète ne doit pas être « bootstrappée ». L'utilisation du *bootstrap* est plutôt à réserver aux statistiques asymptotiquement normales, qui représentent la majorité des statistiques utilisées dans le domaine de la santé. Pour l'estimation d'un simple pourcentage il peut aussi y avoir des problèmes si le nombre d'événements est faible.

Les outils associés au *bootstrap* permettent de faire des diagnostics basés sur l'analyse graphique de la distribution empirique des fluctuations d'échantillonnages : principalement l'histogramme et le Q-Q plot de normalité. En tant que statisticien, leur usage permet d'acquérir une expérience sur les fluctuations d'échantillonnages des statistiques utilisées ; c'est ainsi un outil pédagogique. Par ailleurs, ils permettent parfois d'identifier des problèmes majeurs de validité que l'on n'avaient pas anticipés (*sanity check*) ou de se rassurer lorsque l'on craint de ne pas être très loin de la limite de validité. Si l'histogramme et le Q-Q plot de normalité de la statistique bootstrappée ne suggèrent pas d'écart substantiel à la normalité sur la zone d'intérêt (souvent -2 à $+2$ erreurs types lorsqu'on s'intéresse aux intervalles de confiance à 95%), cela est fortement rassurant, la variante par approximation normale (la plus fragile des variantes du *bootstrap*) étant *a priori* valide (¹).

Au total, le *bootstrap* est une méthode statistique polyvalente, qui oblige à réfléchir à la méthodologie d'échantillonnage et qui présente un intérêt pédagogique et pratique certain. Le *bootstrap* permet de se recentrer sur le choix de la statistique pertinente pour le problème de recherche (« quel est l'estimateur qui a le plus grand intérêt clinique ? ») plutôt que sur des considérations liées aux calculs d'incertitude (« quel est le modèle qui colle le mieux aux données ? »). En cela, le

Si l'on observe zéro événement sur l'échantillon, le *bootstrap* ré-échantillonnera sans cesse des zéros et considérera que le pourcentage est à 0% avec une variance nulle. Si des dizaines d'*outliers* influents (événements ou non-événements pour les variables binaires de proportion proches de 0% ou 100%) sont présents dans l'échantillon, alors il ne devrait pas y avoir de problème majeur, mais s'il n'y a que quelques événements ou *outliers* visibles dans l'échantillon, la variance estimée par le *bootstrap* peut être trop différente de la vraie variance car la fréquence observée des *outliers* est trop éloignée de la fréquence réelle. La nature de la statistique d'intérêt influence également beaucoup le nombre d'observations nécessaires pour atteindre la validité. La notion d'*outlier* influent est fondamentalement liée à « l'influence », qui est une notion relative à la statistique d'intérêt. Par exemple, pour estimer la différence de pourcentage $1/10 - 30/100$ sur un échantillon de 110 observations indépendantes (10 dans un groupe et 100 dans un autre), l'observation « 1 » dans le groupe de 10 observations a une forte influence sur la statistique de différence de moyennes et son faible nombre ($n = 1$) engendre des problèmes de validité du *bootstrap*.

¹ Il existe cependant le risque d'être faussement rassuré si des *outliers* invisibles existent. La méthode à biais corrigé accéléré (BCa) est moins exigeante en termes de condition de validité, puisqu'il suffit qu'il existe une transformation normalisante pour la variable d'intérêt. Cette méthode, comme la méthode percentile, fournit d'ailleurs un intervalle de confiance indépendant de toute transformation monotone. Parfois une telle transformation peut être anticipée (par exemple arc-tangente hyperbolique pour un ICC ou un coefficient de corrélation de Pearson), ce qui permet de tracer le Q-Q plot de normalité sur la variable transformée. Enfin, de manière plus anecdotique, il est possible d'apprécier l'influence de chacune des observations sur la statistique, avec la fonction `empinf` de la bibliothèque `boost`.

bootstrap entre en concurrence avec des modèles statistiques tels que les modèles à effets mixtes, qui sont parfois plus utilisés pour leur capacité à gérer des mesures répétées que pour l'intelligibilité de leurs résultats.