CASSION · FORMATION

Régression appliquée aux données de programme

Modèles linéaires, logistiques et multiniveaux ajustés sur des données d'enquête et de routine, avec une interprétation rédigée pour un responsable de programme non statisticien.

Formateur	Pierre Robentz Cassion
Niveau	Avancé
Heures apprenant	20 h
Enseignée en	Python · R
Mise à jour	29 juillet 2026
Secteurs	Éducation · Protection · Nutrition · Sécurité alimentaire
Normes et méthodologies	Enquête SMART · Enquête démographique et de santé (EDS) · Définitions d'indicateurs de l'UNICEF · Critères d'évaluation du CAD de l'OCDE

Consulter et exécuter la formation en ligne sur data-analysis.cassion.dev/fr/cours/regression-for-programmes

Ce que vous saurez faire

— Lire un coefficient de régression comme la comparaison de moyennes qu'il est, et dire laquelle

— Distinguer l'ajustement, qui déplace l'estimation, du regroupement en grappes, qui déplace l'erreur-type

— Ajuster un modèle logistique et le rapporter en différence de risques plutôt qu'en rapport de cotes que le lecteur interprétera de travers

— Décider quelles covariables ont leur place dans un modèle et lesquelles en détruisent la réponse par leur seule présence

— Ajuster une constante aléatoire quand le protocole a attribué au-dessus du niveau de la ligne

— Pondérer un modèle selon son plan de sondage, et rendre compte d'un modèle qui s'ajuste mal plutôt que de l'écarter

Prérequis

— Statistiques appliquées aux programmes

Sommaire

I Un modèle est une comparaison **1**

1 Le coefficient est l'écart **2**
1.1 La même comparaison, écrite deux fois . . . 2
1.2 Ce que signifie chaque terme . . . 3
1.3 Où les deux chemins diffèrent, et de combien . . . 3
1.4 Trois prédicteurs, trois lectures . . . 4
1.5 Ce que le modèle ne dit pas . . . 4
1.6 Rapportez-le en entier . . . 5
1.7 La suite . . . 5

2 L'ajustement déplace l'estimation ; les grappes déplacent l'erreur-type **6**
2.1 Le tableau autour duquel ce cours est bâti . . . 6
2.2 Ce que « ajuster sur » fait réellement . . . 7
2.3 Pourquoi les covariables n'ont presque rien déplacé . . . 8
2.4 Lire un tableau de coefficients sans se faire tromper . . . 8
2.5 Quand l'ajustement ne peut pas aider . . . 8
2.6 Rapportez-le en entier . . . 9
2.7 La suite . . . 9

II Des résultats en oui ou non **10**

3 Un rapport de cotes de 0,43 pour un rapport de risques de 0,58 **11**
3.1 Ajustez-le, et lisez ce qu'il imprime . . . 11
3.2 Pourquoi ils diffèrent, et quand cela empire . . . 12
3.3 Le piège qui attrape les analystes, pas seulement les lecteurs . . . 12
3.4 Quand le rapport de cotes est le bon nombre . . . 13
3.5 La phrase à écrire, et trois à ne pas écrire . . . 13
3.6 Rapportez-le en entier . . . 14
3.7 La suite . . . 14

4 Trente-huit cas, pas un rapport de cotes **15**
4.1 Reconvertissez le modèle en probabilités . . . 15
4.2 Pourquoi un rapport de cotes fait plusieurs différences de risques . . . 15
4.3 Un intervalle sur l'effet marginal . . . 16
4.4 Multipliez-le par la charge de cas . . . 17
4.5 Quatre façons de se tromper . . . 17
4.6 Rapportez-le en entier . . . 17
4.7 La suite . . . 18

III Quelles covariables ont leur place 19

5 La covariable qui améliore chaque statistique et détruit la réponse 20
5.1 Une covariable qui paraît manifestement juste . . . 20
5.2 Pourquoi elle retire l'effet . . . 20
5.3 Quatre sortes de covariables, et ce que chacune fait . . . 21
5.4 Le tracer avant d'ajuster . . . 21
5.5 Le collisionneur, qui est pire . . . 22
5.6 Rapportez les deux, et dites lequel est lequel . . . 22
5.7 La suite . . . 23

6 Trois réponses honnêtes et une qui est seule de son côté 24
6.1 Le même coefficient, quatre erreurs-types . . . 24
6.2 Ce qu'ajoute une constante aléatoire . . . 25
6.3 Choisir parmi les trois . . . 25
6.4 Des prédicteurs à deux niveaux . . . 25
6.5 Trois niveaux, et quand s'arrêter . . . 26
6.6 Rapportez-le en entier . . . 26
6.7 La suite . . . 27

IV Le plan et l'ajustement 28

7 Deux coefficients traversent le seuil et un le retraverse 29
7.1 Le plan sous lequel l'échantillon a été tiré . . . 29
7.2 Ce que les poids font à une estimation . . . 29
7.3 Le même modèle, pondéré et non . . . 30
7.4 Poids, grappes et strates sont trois choses distinctes . . . 31
7.5 Quand laisser les poids de côté . . . 31
7.6 Rapportez-le en entier . . . 31
7.7 La suite . . . 32

8 Un R^2 de 0,03, et le constat le plus utile du rapport 33
8.1 Un modèle bâti pour répondre à une question de ciblage . . . 33
8.2 Ne le supprimez pas . . . 34
8.3 Des diagnostics qui changent une décision . . . 34
8.4 Ce que le R^2 est et n'est pas . . . 35
8.5 Rapportez-le en entier . . . 35
8.6 La suite . . . 36

À propos de cette formation

Une régression n'est pas un instrument plus sophistiqué que les comparaisons de *Statistiques appliquées aux programmes*. C'est la même comparaison avec plus d'une chose maintenue fixe, et la première leçon le démontre : ajustez un modèle à une seule variable indicatrice et le coefficient **est** l'écart entre deux moyennes de groupe, à la décimale près.

C'est l'ossature du cours. Un coefficient est toujours une comparaison, et le travail consiste à dire laquelle.

Trois résultats le structurent, tous ajustés sur des fichiers déjà commités.

L'ajustement et le regroupement en grappes sont deux problèmes distincts aux remèdes distincts. L'effet de la cantine vaut 4,9 points avec une erreur-type de 0,9 en ajustement naïf. Ajouter toutes les covariables individuelles du registre déplace l'estimation à 4,4 et laisse l'erreur-type tranquille. Ajouter un terme pour l'école laisse l'estimation tranquille et porte l'erreur-type à 1,5. Confondre les deux est l'erreur la plus courante d'une régression de programme, et c'est pourquoi elles sont enseignées dans le même cours.

Un rapport de cotes de 0,43 décrit un rapport de risques de 0,58. Les cas signalant un handicap aboutissent à 26,4 % contre 45,5 %. Le rapport de cotes est ce qu'imprime un modèle logistique et la différence de risques est ce que le lecteur croit qu'on lui dit, et l'écart entre les deux se creuse précisément quand le résultat est fréquent — ce qui, dans des données de programme, est le cas habituel.

Ajuster sur la mauvaise covariable retire 42 % de l'effet. L'aboutissement d'une orientation passe par une porte — le fait qu'une orientation ait été émise — et les cas signalant un handicap franchissent cette porte à 53,8 % contre 71,5 %. Mettez la porte dans le modèle et l'écart selon le handicap tombe de 19,4 points à 11,2, non parce que l'estimation s'est améliorée mais parce que le modèle s'est mis à répondre à une autre question.

Python et R tout du long. Il vous faut *Statistiques appliquées aux programmes* d'abord ; ce cours suppose que vous savez déjà poser un intervalle sur un nombre et dire ce qu'une valeur p ne vous apprend pas.

Première partie

Un modèle est une comparaison

CHAPITRE 1

Le coefficient est l'écart

Leçon 1 sur 8 · 150 min

Ajustez la présence sur une seule indicatrice de cantine. La constante vaut 85,21 % — la moyenne des écoles sans programme. Le coefficient vaut 4,94 points — l'écart entre les deux moyennes, à la quatrième décimale. Une régression à une indicatrice est une comparaison de deux groupes, rien de plus.

1.1 La même comparaison, écrite deux fois

Le cours de statistiques a comparé la présence dans les écoles avec cantine à celle des écoles sans, et a obtenu 90,15 % contre 85,21 %. Ajustez cela comme une régression et regardez ce qui revient.

```python
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf

attendance = pd.read_csv("school-attendance-2024.v1.csv")
roster = pd.read_csv("school-roster-2024.v1.csv").drop_duplicates(
    "student_id", keep="last")          # two never-de-registered transfers

MARKS = {"true": True, "Y": True, "false": False, "N": False}
marked = attendance[attendance["present"].isin(MARKS)].copy()
marked["attended"] = marked["present"].map(MARKS)

per_student = (marked.groupby("student_id")["attended"].mean()
               .rename("rate").reset_index()
               .merge(roster, on="student_id"))

model = smf.ols("rate ~ feeding_programme", data=per_student).fit()
print(model.summary().tables[1])
```

```r
library(dplyr)

per_student |> lm(rate ~ feeding_programme, data = _) |> summary()
```

Terme	Coefficient	ET	t
Constante	**0,8521**	0,0070	121,30
feeding_programme[True]	**0,0494**	0,0088	5,63

```python
means = per_student.groupby("feeding_programme")["rate"].mean()
print(means)
print(f"difference: {means[True] - means[False]:.4f}")
```

```r
per_student |> summarise(rate = mean(rate), .by = feeding_programme)
```

Moyenne sans programme : 0,8521. Avec : 0,9015. Écart : 0,0494.

La constante est le premier nombre et le coefficient le troisième, exactement. **Ce n'est ni une approximation ni une coïncidence** — avec un seul prédicteur binaire et rien d'autre, les moindres carrés n'ont aucune latitude pour faire autre chose que reproduire les deux moyennes de groupe.

1.2 Ce que signifie chaque terme

Trois phrases couvrent toute la lecture d'un tableau de coefficients, et elles valent pour chaque modèle de ce cours.

La constante est la valeur ajustée quand tous les prédicteurs valent zéro. Ici c'est `feeding_programme = False`, donc la constante est la moyenne des écoles sans programme. Ce n'est pas « la présence moyenne » et ce n'est une référence en aucun sens programmatique — c'est la moyenne d'un groupe précis, et lequel dépend entièrement du codage des prédicteurs.

Un coefficient est la variation du résultat pour une variation d'une unité de ce prédicteur, les autres étant maintenus fixes. Pour une indicatrice, « une unité » vaut False → True, donc le coefficient est un écart entre deux groupes.

Une erreur-type est la même erreur-type que celle calculée au cours précédent. La colonne `t` vaut le coefficient sur l'erreur-type, et l'intervalle vaut le coefficient plus ou moins environ deux erreurs-types. Rien de neuf n'est introduit.

```
coef = model.params["feeding_programme[T.True]"]
se = model.bse["feeding_programme[T.True]"]
print(f"{coef:+.4f}  95% CI [{coef - 1.96*se:+.4f}, {coef + 1.96*se:+.4f}]")
```

```
confint(model)
```

+0,0494, IC à 95 % +0,0322 à +0,0665 — le même intervalle, atteint depuis les mêmes données par un autre chemin.

1.3 Où les deux chemins diffèrent, et de combien

Le t de la régression vaut 5,63 et le test t de Welch du cours de statistiques donnait 5,41. Cette différence n'est une erreur ni de l'un ni de l'autre.

Les moindres carrés ordinaires supposent une seule variance résiduelle pour les deux groupes. C'est le test t à variance commune écrit sous forme matricielle. Le test de Welch ne la suppose pas, ce qui est la raison pour laquelle le cours précédent recommandait Welch par défaut.

```
from scipy import stats

fed = per_student[per_student["feeding_programme"]]["rate"]
unfed = per_student[~per_student["feeding_programme"]]["rate"]
print(f"variances: {fed.var():.5f} and {unfed.var():.5f}")
print("pooled t:", stats.ttest_ind(fed, unfed).statistic.round(2))
print("Welch  t:", stats.ttest_ind(fed, unfed, equal_var=False).statistic.round(2))
```

```
t.test(rate ~ feeding_programme, data = per_student, var.equal = TRUE)
t.test(rate ~ feeding_programme, data = per_student)
```

La régression reproduit exactement le t à variance commune. Quand les variances de groupe sont proches les deux diffèrent à peine ; ici elles valent 0,019 et 0,025, et l'écart est la raison d'être de Welch. **On ne peut pas demander « Welch » à une régression** — le remède est une erreur-type robuste, que la leçon 6 introduit pour une raison voisine.

1.4 Trois prédicteurs, trois lectures

L'utilité d'une régression commence quand le prédicteur n'est pas binaire. Chaque type se lit différemment et chacun est une comparaison.

Prédicteur	Le coefficient se lit
Binaire (`feeding_programme`)	L'écart entre les deux groupes
Continu (`age_years`)	La variation par année
Catégoriel à k niveaux	k−1 coefficients, chacun un écart contre le niveau omis

```python
enrolment = pd.read_csv("school-enrolment-2024.v1.csv")
current = enrolment[enrolment["school_year"] == 2024]
joined = per_student.merge(
    current[["student_id", "sex", "age_years", "admin2"]], on="student_id")

print(smf.ols("rate ~ admin2", data=joined).fit().params.round(4))
print(joined.groupby("admin2")["rate"].mean().round(4))
```

```r
lm(rate ~ admin2, data = joined) |> coef()
joined |> summarise(rate = mean(rate), .by = admin2)
```

Un prédicteur catégoriel produit un coefficient par niveau sauf un, et chaque coefficient est une comparaison contre ce niveau omis. Changez le niveau omis et chaque nombre de la colonne change alors que le modèle est identique — premier signe qu'un coefficient ne signifie rien sans savoir à quoi il est comparé.

Énoncez le niveau de référence dans la légende du tableau. « Contre Artibonite » fait deux mots et c'est la différence entre un tableau lisible et un tableau que le lecteur doit deviner.

1.5 Ce que le modèle ne dit pas

Deux choses qu'un coefficient ne porte jamais, et que les lecteurs fournissent d'eux-mêmes.

Ce n'est pas un effet. Le coefficient de 4,9 points est l'écart entre deux ensembles d'écoles qui n'ont pas été randomisées vers une cantine. Chaque régression de ce cours est une comparaison de groupes qui différaient déjà ; la leçon 5 porte sur celles de ces différences que vous pouvez et ne pouvez pas retirer.

Ce n'est pas une prédiction qui vaille la peine. La régression est enseignée ailleurs comme une machine à prédire, et pour des données de programme ce cadrage fait des dégâts : il invite à comparer des modèles par leur ajustement plutôt que par la question de savoir si la comparaison est celle dont le rapport a besoin. **Ajustez le modèle qu'implique votre question, puis lisez le coefficient pour lequel il a été ajusté**, et traitez le R^2 comme un diagnostic et non comme une note — la leçon 8 porte sur un modèle de R^2 0,03 qui est le constat le plus utile de son rapport.

1.6 Rapportez-le en entier

```
Attendance and school feeding, February to April 2024

  Linear model, one predictor, 1,200 students.

  Intercept (no programme)   85.21%
  Feeding programme          +4.94 points   95% CI +3.22 to +6.65

  The coefficient is the difference between the two group means and is
  reported as such. Schools were not randomised into the programme; the
  comparison is observational.

  The standard error assumes independent students. It is not, and lesson 6
  gives the corrected version.
```

La dernière ligne est une reconnaissance de dette que le cours honore, et l'écrire vaut mieux que d'ignorer qu'elle est due. Un modèle rapporté avec une réserve que vous savez nommer se porte mieux qu'un modèle rapporté sans.

1.7 La suite

Un prédicteur reproduit une comparaison que vous auriez pu faire sans modèle. La leçon suivante ajoute le deuxième prédicteur, là où une régression commence à gagner sa place — et où « ajuster sur » reçoit un sens précis, plus étroit que ce que supposent la plupart des rapports.

CHAPITRE 2

L’ajustement déplace l’estimation ; les grappes déplacent l’erreur-type

Leçon 2 sur 8 · 150 min

Ajouter toutes les covariables individuelles du registre fait passer le coefficient de cantine de 4,47 à 4,37 et laisse l’erreur-type à 0,9. Ajouter un terme pour l’école laisse le coefficient tranquille et porte l’erreur-type à 1,5. Ce sont deux problèmes distincts et on les confond couramment.

2.1 Le tableau autour duquel ce cours est bâti

```python
import pandas as pd
import statsmodels.formula.api as smf

attendance = pd.read_csv("school-attendance-2024.v1.csv")
roster = pd.read_csv("school-roster-2024.v1.csv").drop_duplicates(
    "student_id", keep="last")
MARKS = {"true": True, "Y": True, "false": False, "N": False}
marked = attendance[attendance["present"].isin(MARKS)].copy()
marked["attended"] = marked["present"].map(MARKS)
per_student = (marked.groupby("student_id")["attended"].mean()
               .rename("rate").reset_index().merge(roster, on="student_id"))

enrolment = pd.read_csv("school-enrolment-2024.v1.csv")
current = enrolment[enrolment["school_year"] == 2024]

d = per_student.merge(
    current[["student_id", "sex", "age_years", "disability_reported",
             "displacement_status", "admin2"]], on="student_id").dropna()
print(f"complete cases: {len(d)}")

m1 = smf.ols("rate ~ feeding_programme", data=d).fit()
m2 = smf.ols("rate ~ feeding_programme + sex + age_years"
             " + disability_reported + displacement_status", data=d).fit()
m3 = smf.ols("rate ~ feeding_programme + sex + age_years"
             " + disability_reported + displacement_status + admin2", data=d).fit()
m4 = m1.get_robustcov_results(cov_type="cluster", groups=d["school_id"])

for name, m in [("child covariates", m2), ("+ district", m3)]:
    print(f"{name:18} {m.params['feeding_programme[T.True]']:+.4f}")
```

```r
library(dplyr)

m1 <- lm(rate ~ feeding_programme, data = d)
m2 <- lm(rate ~ feeding_programme + sex + age_years +
           disability_reported + displacement_status, data = d)
m3 <- update(m2, . ~ . + admin2)
```

Modèle	Coefficient cantine	ET	t
M1 un prédicteur	**+0,0447**	0,0089	5,02
M2 + sexe, âge, handicap, déplacement	**+0,0437**	0,0090	4,85
M3 + district	**+0,0360**	0,0100	3,60
M4 = M1 avec ET groupées par école	**+0,0447**	**0,0154**	**2,90**

Lisez les colonnes plutôt que les lignes, car elles disent deux choses différentes.

Ajouter des covariables a changé le coefficient et laissé l'erreur-type tranquille. De M1 à M3, l'estimation passe de 4,5 points à 3,6 tandis que l'erreur-type va de 0,9 à 1,0.

Le regroupement a changé l'erreur-type et laissé le coefficient intact. M4 est la *même droite ajustée* que M1 — coefficient identique à toutes les décimales — avec une erreur-type 1,7 fois plus grande.

Ce sont deux problèmes distincts. L'un porte sur le caractère comparable des groupes ; l'autre sur la quantité d'information que contient réellement l'échantillon. Résoudre l'un ne fait rien pour l'autre, et un modèle qui ne résout ni l'un ni l'autre est faux de deux façons indépendantes.

2.2 Ce que « ajuster sur » fait réellement

L'expression figure dans chaque rapport de programme et signifie quelque chose de plus étroit que ce que supposent les lecteurs.

Un coefficient avec des covariables dans le modèle est la comparaison à l'intérieur des niveaux de ces covariables, mise en commun sur ces niveaux. « Ajusté sur le district » signifie : comparer élèves nourris et non nourris *à l'intérieur* de chaque district, puis moyenner ces comparaisons.

```python
within = (d.groupby("admin2")
          .apply(lambda g: g[g["feeding_programme"]]["rate"].mean()
                 - g[~g["feeding_programme"]]["rate"].mean()))
print(within.round(4))
print(f"unadjusted: {m1.params['feeding_programme[T.True]']:.4f}")
print(f"adjusted:   {m3.params['feeding_programme[T.True]']:.4f}")
```

```r
d |> summarise(gap = mean(rate[feeding_programme]) -
                     mean(rate[!feeding_programme]), .by = admin2)
```

District	Écart	Avec cantine	Sans
Artibonite	+3,8 pts	146	141
Centre	**−4,1 pts**	244	**42**
Nord-Ouest	+4,9 pts	251	45
Sud	+7,0 pts	104	178

Le coefficient ajusté est une moyenne pondérée de ces quatre écarts, et les voir séparément vaut les deux lignes que cela coûte — parce que l'un d'eux va dans l'autre sens. Les écoles avec cantine du Centre sont présentes 4,1 points *de moins* que celles sans, contre une estimation groupée de +3,6.

Un coefficient unique moyenné sur un désaccord reste un nombre, et ce n'est plus un résumé. Avant de rapporter le 3,6 ajusté, dites que le Centre est en désaccord et que sa comparaison repose sur 42 élèves sans cantine — assez peu pour que le renversement puisse être du bruit, et assez peu pour qu'il faille le vérifier plutôt que le moyenner.

2.3 Pourquoi les covariables n'ont presque rien déplacé

M2 ajoute sexe, âge, handicap et statut de déplacement et déplace le coefficient d'un dixième de point. Ce n'est pas un échec des covariables — c'est un fait sur le protocole.

La cantine a été attribuée par école. Rien concernant un *enfant* n'a décidé s'il recevait des repas scolaires, si bien que les caractéristiques individuelles ne peuvent pas être des facteurs de confusion de cette comparaison, et ajuster dessus ne peut pas beaucoup déplacer l'estimation.

Le district, lui, l'a déplacée, de 4,5 à 3,6, et pour la raison inverse : les écoles avec cantine ne sont pas réparties uniformément entre districts, donc le district est un vrai facteur de confusion pour une exposition de niveau école.

La règle qu'implique le protocole : les facteurs de confusion vivent au niveau de l'attribution. Pour un programme de niveau école, cherchez les différences de niveau école et district ; ajouter des covariables individuelles n'est ici ni nuisible ni utile, et signale surtout que l'analyste a pris ce que le fichier contenait.

2.4 Lire un tableau de coefficients sans se faire tromper

Quatre habitudes, chacune prévenant une mélecture précise.

Rapportez la taille d'échantillon du modèle, non celle du fichier. L'ajustement sur cas complets a écarté 49 élèves ici — 1 200 lignes deviennent 1 151 — et l'écart est silencieux. Chaque modèle du tableau ci-dessus doit être ajusté sur les *mêmes* lignes, sans quoi la comparaison entre modèles est contaminée par les lignes que chacun a gardées.

```python
print(f"file {len(per_student)}, model {int(m3.nobs)}")
```

```r
c(file = nrow(per_student), model = nobs(m3))
```

Ne lisez pas les coefficients des covariables comme des constats. Ils sont ajustés sur un ensemble de variables choisi pour répondre à une question sur la cantine, non sur le sexe ou le déplacement. Un coefficient n'est interprétable que pour l'exposition autour de laquelle le modèle a été bâti — c'est le « sophisme du tableau 2 », et c'est ainsi qu'un coefficient de déplacement ajusté comme contrôle finit cité comme un constat sur le déplacement.

Vérifiez si une covariable ajoutée a changé l'échantillon. Une covariable avec des valeurs manquantes écarte silencieusement des lignes, donc un coefficient qui bouge quand vous l'ajoutez a peut-être bougé parce que l'échantillon a changé et non parce que l'ajustement a fait quelque chose.

Rapportez aussi l'estimation non ajustée. La paire est ce qui montre au lecteur le travail qu'a fait l'ajustement, et un rapport ne donnant que le nombre ajusté a caché la seule comparaison qui n'exige aucune hypothèse.

2.5 Quand l'ajustement ne peut pas aider

Trois cas où ajouter une covariable n'est pas la réponse, tous présents plus loin dans ce cours.

Le facteur de confusion n'est pas dans le fichier. Ajuster sur ce que vous avez mesuré ne dit rien de ce que vous n'avez pas mesuré, et « ajusté » n'est pas synonyme de « sans confusion ». Le cours d'épidémiologie faisait ce point avec la standardisation sur l'âge ; la régression n'y ajoute rien.

La covariable est sur le chemin causal. La leçon 5 montre une covariable qui retire 42 % de l'effet en se plaçant entre l'exposition et le résultat, et l'ajouter détériore la réponse d'une façon qu'aucune statistique d'ajustement ne révèle.

Le problème est l'erreur-type, non l'estimation. Aucune covariable ne corrige le regroupement en grappes. C'est la ligne M1 vers M4, et la leçon 6 y répond entièrement.

2.6 Rapportez-le en entier

```
Attendance and school feeding, adjusted analysis

  Linear model, 1,151 students with complete covariates (49 dropped).

  Unadjusted                    +4.47 points   95% CI +2.73 to +6.22
  Adjusted for district         +3.60 points   95% CI +1.64 to +5.56
  Child-level covariates (sex, age, disability, displacement) change the
  estimate by less than 0.1 points and are retained only to show that.

  The district gaps are +3.8, -4.1, +4.9 and +7.0 points. Centre runs the
  other way on 42 unfed students and is not averaged away silently.

  Standard errors above assume independent students and are too small; the
  school-clustered version of the unadjusted model gives +4.47 (95% CI
  +1.45 to +7.49).

  Observational. Schools were not randomised into the programme, and
  district adjustment does not make the remaining comparison causal.
```

Les deux estimations, et l'intervalle groupé, en six lignes. Un lecteur qui ne voit que le nombre ajusté ne peut pas savoir si l'ajustement a compté, et un lecteur qui ne voit que l'intervalle naïf s'entend dire que le résultat est deux fois plus précis qu'il ne l'est.

2.7 La suite

Tout jusqu'ici avait un résultat continu. La leçon suivante prend un résultat en oui ou non — cette orientation a-t-elle atteint un service — et rencontre le nombre que ce public interprète de travers plus sûrement qu'aucun autre.

Deuxième partie

Des résultats en oui ou non

CHAPITRE 3

Un rapport de cotes de 0,43 pour un rapport de risques de 0,58

Leçon 3 sur 8 · 150 min

Les cas signalant un handicap aboutissent à 26,4 % contre 45,5 %. C'est 58 % du taux, et le modèle logistique imprime 0,43. Les deux nombres décrivent les mêmes données, un seul est ce que le lecteur croit lire, et l'écart se creuse précisément quand le résultat est fréquent.

3.1 Ajustez-le, et lisez ce qu'il imprime

```python
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf

referrals = pd.read_csv("protection-referrals-2024.v1.csv")
referrals["disability"] = referrals["disability_reported"].map(
    {"true": 1, "Yes": 1, "false": 0, "No": 0})

consenting = referrals[referrals["consent_to_refer"]].copy()
consenting["completed"] = (consenting["referral_accepted"]
                           & consenting["days_to_first_service"].notna()).astype(int)
d = consenting.dropna(subset=["disability", "case_category", "age_band",
                              "sex", "service_requested", "admin1"])

crude = smf.logit("completed ~ disability", data=d).fit(disp=0)
print(np.exp(crude.params).round(3))
print(np.exp(crude.conf_int()).round(3))
```

```r
library(dplyr)

crude <- glm(completed ~ disability, data = d, family = binomial())
exp(cbind(OR = coef(crude), confint(crude)))
```

Rapport de cotes 0,430, IC à 95 % 0,308 à 0,601.
Calculez maintenant ce que les données disent simplement.

```python
rates = d.groupby("disability")["completed"].agg(["sum", "size", "mean"])
print(rates.round(4))
p1, p0 = rates.loc[1, "mean"], rates.loc[0, "mean"]
print(f"risk ratio        {p1 / p0:.3f}")
print(f"risk difference   {p1 - p0:+.4f}")
print(f"odds ratio        {(p1/(1-p1)) / (p0/(1-p0)):.3f}")
```

```r
d |> summarise(k = sum(completed), n = n(), rate = mean(completed), .by = disability)
```

Groupe	Aboutis	n	Taux
Handicap signalé	52	197	**26,4 %**
Non signalé	629	1 384	**45,5 %**

Mesure	Valeur	Se lit
Rapport de risques	**0,581**	58 % de chances d'aboutir
Différence de risques	**−19,1 points**	19 aboutissements de moins pour 100 cas
Rapport de cotes	**0,430**	—

Le modèle a imprimé 0,430 et la réponse vaut 0,581. Les deux sont corrects ; ce sont des réponses à des questions différentes, et une seule est la question que quelqu'un a posée.

3.2 Pourquoi ils diffèrent, et quand cela empire

Une cote vaut `p / (1 − p)`. Quand `p` est petit, le dénominateur est proche de 1 et les cotes sont proches des risques, si bien que les deux rapports concordent presque. Quand `p` est grand, non.

```
for p0 in (0.02, 0.10, 0.25, 0.45, 0.70):
    p1 = 0.6 * p0                      # a true risk ratio of 0.6 throughout
    odds = (p1 / (1 - p1)) / (p0 / (1 - p0))
    print(f"baseline {p0:5.0%}   risk ratio 0.60   odds ratio {odds:.3f}")
```

```
# One true risk ratio, five baselines, five different odds ratios.
```

Risque de base	Vrai rapport de risques	Rapport de cotes
2 %	0,60	0,59
10 %	0,60	0,57
25 %	0,60	0,53
45 %	**0,60**	**0,45**
70 %	0,60	0,31

Le rapport de cotes n'est pas une distorsion fixe du rapport de risques — il dépend du niveau de base. À 2 % de résultat les deux sont interchangeables, ce qui explique la survie du rapport de cotes en épidémiologie, où les résultats sont rares.

Les résultats de programme ne sont pas rares. L'aboutissement des orientations vaut 43 %, la présence 88 %, l'inscription 97 %, et à ces niveaux le rapport de cotes est très loin de ce que le lecteur y lira. Ce n'est pas une subtilité — c'est le cas ordinaire dans ce métier.

3.3 Le piège qui attrape les analystes, pas seulement les lecteurs

Ajouter des covariables change un rapport de cotes *même sans aucune confusion*, ce qui surprend qui n'a travaillé qu'avec des modèles linéaires.

```
adjusted = smf.logit(
    "completed ~ disability + case_category + age_band + sex"
    " + service_requested + admin1", data=d).fit(disp=0)
print(f"crude OR    {np.exp(crude.params['disability']):.3f}")
print(f"adjusted OR {np.exp(adjusted.params['disability']):.3f}")
```

```
adjusted <- glm(completed ~ disability + case_category + age_band + sex +
                  service_requested + admin1, data = d, family = binomial())
exp(coef(adjusted))["disability"]
```

	Rapport de cotes	Différence de risques
Brut	0,430	−19,05 points
Ajusté	**0,388**	**−19,41 points**

Le rapport de cotes a bougé de 10 % et la différence de risques d'un tiers de point. La lecture habituelle — « l'ajustement a révélé un effet plus fort » — est fausse ici. Rien n'a été démêlé d'une confusion ; le rapport de cotes n'est simplement pas *effondrable*.

Une mesure effondrable est égale à la moyenne des mesures de sous-groupes. Les différences et les rapports de risques le sont ; les rapports de cotes non. Ajoutez une covariable qui prédit le résultat et le rapport de cotes conditionnel s'éloigne de 1 même si la covariable est sans lien avec l'exposition.

Un rapport de cotes ajusté et un rapport de cotes brut ne peuvent donc pas être comparés pour juger d'une confusion. Cette comparaison est le geste standard de tout tutoriel de régression et elle ne fonctionne pas pour les modèles logistiques. Comparez plutôt des différences de risques, que la leçon suivante calcule.

3.4 Quand le rapport de cotes est le bon nombre

Trois cas, et il vaut la peine d'être précis parce que la réponse n'est pas « jamais ».

Une étude cas-témoins. Cas et témoins sont échantillonnés séparément, si bien que les risques ne sont pas estimables du tout et que le rapport de cotes est le seul rapport que le protocole autorise. C'est ce pour quoi la mesure a été inventée.

Un résultat rare. En dessous d'environ 10 %, le rapport de cotes approche le rapport de risques d'assez près pour que la distinction cesse de compter. Dites quand même lequel vous avez calculé.

Une comparaison avec une littérature publiée en rapports de cotes. Rapportez alors les deux, et menez avec la différence de risques.

```
def report(label, p1, p0):
    return (f"{label}: {p1:.1%} vs {p0:.1%} -- "
            f"risk difference {p1-p0:+.1%}, risk ratio {p1/p0:.2f}, "
            f"odds ratio {(p1/(1-p1))/(p0/(1-p0)):.2f}")

print(report("Referral completion, disability reported", p1, p0))
```

```
# Print all three. The reader picks; you do not pick for them by omission.
```

3.5 La phrase à écrire, et trois à ne pas écrire

Pas ceci : « Les cas signalant un handicap avaient 57 % de chances en moins d'aboutir (RC 0,43). » Deux erreurs en une phrase — un rapport de cotes n'est pas une probabilité, et 57 % n'est pas la réduction.

Ni ceci : « Le handicap a divisé par deux les cotes d'aboutissement. » Arithmétique correcte, et « divisé par deux » sera lu comme un risque par tous ceux qui ne sont pas statisticiens.

Ni ceci : « RC 0,43 (IC à 95 % 0,31–0,60, p < 0,001). » Complet, vérifiable, et cela n'apprend rien d'actionnable à un responsable de programme.

Ceci : « Les cas signalant un handicap ont abouti à 26,4 % contre 45,5 % pour les cas sans handicap signalé — 19,1 points de pourcentage de moins (IC à 95 % −25,7 à −12,4), soit 58 % du taux d'aboutissement. Ajusté sur la catégorie de cas, l'âge, le sexe, le service demandé et le département, l'écart vaut 19,4 points. Le rapport de cotes ajusté vaut 0,39 ; il est indiqué ici pour la comparabilité avec les études publiées et ne doit pas être lu comme un risque. »

La dernière proposition est celle qui travaille. Un rapport de cotes dans un rapport de programme sans traduction à côté sera lu comme un risque par presque tous ceux qui le verront, y compris ceux qui ont commandé l'analyse.

3.6 Rapportez-le en entier

```
Referral completion by disability status, 1,581 consenting cases

  Disability reported     26.4%   52 of 197
  Not reported            45.5%   629 of 1,384

  Risk difference   -19.1 points   95% CI -25.7 to -12.4
  Risk ratio          0.58
  Odds ratio          0.43 (crude), 0.39 (adjusted)

  The odds ratio is reported for comparability only. Completion is a common
  outcome (43% overall), so the odds ratio overstates the relative
  difference: 0.43 in odds is 0.58 in risk.

  The adjusted odds ratio differs from the crude one partly because the odds
  ratio is not collapsible, not only because of confounding. The adjusted
  risk difference is -19.4 points, essentially unchanged from crude.

  Fitted on the 1,581 consenting cases with complete covariates. The
  statistics course reported this gap on all 1,638 consenting cases and got
  -19.4 points; the 57-case difference is the complete-case restriction.
```

Le dernier paragraphe fait deux lignes et prévient la surinterprétation la plus courante — celle qui veut que l'ajustement ait « renforcé » un constat quand la mesure a simplement bougé pour des raisons arithmétiques.

3.7 La suite

Un modèle logistique imprime des coefficients sur une échelle où personne ne pense. La leçon suivante les reconvertit en probabilités — le nombre qu'un responsable de programme peut multiplier par une charge de cas — et montre ce que le modèle dit des deux portes que la filière d'orientation possède réellement.

CHAPITRE 4

Trente-huit cas, pas un rapport de cotes

Leçon 4 sur 8 · 150 min

Un seul rapport de cotes de 0,39 produit un écart de 22,7 points là où l'aboutissement est fréquent et de 15,2 points là où il est rare. La probabilité prédite est ce qu'un modèle logistique affirme réellement, et multipliée par la charge de cas elle devient 38 cas qui n'ont pas atteint un service.

4.1 Reconvertissez le modèle en probabilités

Un coefficient logistique vit sur l'échelle du log des cotes, où personne ne pense. Le remède tient en une ligne et c'est la même à chaque fois : prédire, et moyenner.

```python
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf

model = smf.logit("completed ~ disability + case_category + age_band + sex"
                  " + service_requested + admin1", data=d).fit(disp=0)

everyone_with = d.assign(disability=1)
everyone_without = d.assign(disability=0)

p1 = model.predict(everyone_with).mean()
p0 = model.predict(everyone_without).mean()
print(f"{p1:.4f} vs {p0:.4f}   average marginal effect {p1 - p0:+.4f}")
```

```r
library(marginaleffects)

avg_comparisons(model, variables = "disability")
```

26,11 % contre 45,52 %, une différence de −19,41 points.

C'est l'**effet marginal moyen** : prendre chaque cas des données, demander au modèle ce qu'il prédit si ce cas signalait un handicap, redemander s'il n'en signalait pas, et moyenner l'écart. C'est la réponse du modèle dans l'unité où la question a été posée.

Comparez-la à l'écart brut de la leçon précédente : −19,05 points. Le rapport de cotes ajusté est passé de 0,430 à 0,388 et la *différence de risques* ajustée n'a presque pas bougé. L'effet marginal moyen est le nombre qui se comporte comme un lecteur attend qu'une estimation ajustée se comporte.

4.2 Pourquoi un rapport de cotes fait plusieurs différences de risques

C'est la propriété qui rend l'effet marginal nécessaire plutôt que simplement commode.

```python
profile = dict(case_category="child-protection", age_band="0-11", sex="f",
               admin1="Artibonite")
rows = []
for service in ["health", "psychosocial", "legal", "livelihood-support"]:
    with_d = pd.DataFrame([{**profile, "service_requested": service, "disability": 1}])
```

```
    without = pd.DataFrame([{**profile, "service_requested": service, "disability": 0}])
    rows.append((service, model.predict(without)[0], model.predict(with_d)[0]))

print(pd.DataFrame(rows, columns=["service", "no disability", "disability"]).round(3))
```

```
predictions(model, newdata = datagrid(service_requested = unique(d$service_requested),
                                      disability = 0:1))
```

Service demandé	Sans handicap	Handicap signalé	Écart
Santé	69,2 %	46,5 %	**−22,7 pts**
Psychosocial	64,5 %	41,3 %	−23,2 pts
Juridique	41,0 %	21,2 %	−19,8 pts
Soutien aux moyens d'existence	28,7 %	13,5 %	**−15,2 pts**

Le rapport de cotes vaut 0,39 sur chacune de ces lignes. Le modèle a été ajusté avec un seul coefficient de handicap, donc par construction le rapport de cotes ne varie pas — et la différence de risques varie tout de même de 15,2 à 23,2 points.

Un rapport de cotes constant n'est pas un effet constant. Là où un résultat est déjà fréquent, le même rapport de cotes déplace plus de points de pourcentage ; là où il est rare, moins. Donc « l'effet du handicap » n'a pas de réponse unique en probabilité sans dire *à quel niveau de base* — ce que fait précisément l'effet marginal moyen, en moyennant sur les niveaux que la charge de cas possède réellement.

4.3 Un intervalle sur l'effet marginal

L'effet marginal est une fonction de tous les coefficients, si bien que son intervalle n'est pas dans le tableau des coefficients. Deux façons de l'obtenir, et la seconde est celle vers laquelle se tourner.

```
# statsmodels: delta method
margins = model.get_margeff(at="overall")
print(margins.summary())
```

```
avg_comparisons(model, variables = "disability")   # delta method, with CI
```

```
# bootstrap, when the delta method is awkward or the estimator is custom
rng = np.random.default_rng(20260729)
draws = []
for _ in range(400):
    sample = d.sample(len(d), replace=True, random_state=int(rng.integers(1e9)))
    m = smf.logit(model.model.formula, data=sample).fit(disp=0)
    draws.append(m.predict(sample.assign(disability=1)).mean()
                 - m.predict(sample.assign(disability=0)).mean())
print(np.percentile(draws, [2.5, 97.5]).round(4))
```

```
# 400 resamples is enough for a reportable interval; 2,000 for a published one.
```

−19,41 points, IC à 95 % −26,1 à −13,2 sur 400 rééchantillonnages bootstrap.

Fixez la graine et dites combien de rééchantillonnages. Un intervalle bootstrap qu'on ne peut pas reproduire n'est pas un intervalle, et la règle de cette plateforme sur les

nombres produits vaut autant pour ceux que produit un modèle que pour ceux que produit un générateur.

4.4 Multipliez-le par la charge de cas

C'est la phrase que lit un responsable de programme, et le cours précédent a établi pourquoi : un effet doit arriver dans l'unité où se prend la décision.

```
n_disability = int((d["disability"] == 1).sum())
comparator = d[d["disability"] == 0]["completed"].mean()
observed = d[d["disability"] == 1]["completed"].sum()
print(f"{n_disability} cases reporting a disability")
print(f"expected at the comparator rate: {comparator * n_disability:.0f}")
print(f"observed: {int(observed)}  shortfall: {comparator * n_disability - observed:.0f}")
```

```
# Three lines, and it is the only line of the analysis a manager will quote.
```

197 cas signalant un handicap. 90 auraient abouti au taux de comparaison ; 52 l'ont fait. Trente-huit cas de moins.

Trente-huit est un nombre sur lequel une équipe de protection peut agir : cela fait environ trois cas par mois, cela nomme une charge de cas plutôt qu'un pourcentage, et cela se compare à ce que coûterait une correction. « RC 0,39 » ne fait rien de tout cela.

Énoncez l'arithmétique, non le seul résultat. Un lecteur qui voit $197 \times 45{,}5\ \% - 52$ peut vérifier ; un lecteur à qui l'on donne seulement « 38 cas » doit faire confiance.

4.5 Quatre façons de se tromper

Prédire à la moyenne au lieu de moyenner les prédictions. Fixer chaque covariable à sa moyenne et prédire une fois donne l'effet pour un cas qui n'existe pas — un cas à 0,6 femme et 0,3 juridique. Moyennez plutôt les prédictions sur les cas réels ; c'est ce que font `at="overall"` et `avg_comparisons`.

Rapporter un effet marginal issu d'un modèle avec interaction sans dire où. Si le modèle laisse l'effet du handicap varier selon le service, la moyenne est une moyenne sur une différence réelle et le tableau des profils est la sortie honnête.

Prendre le déficit pour un effet. Trente-huit cas est ce à quoi correspond l'écart observé, non ce que rapporterait sa fermeture. La comparaison reste observationnelle.

Extrapoler au-delà des données. Le modèle prédira volontiers une probabilité d'aboutissement pour un cas juridique de 80 ans dans un département où aucun n'a été enregistré. Prédisez sur des profils que les données contiennent, et dites lesquels.

4.6 Rapportez-le en entier

```
Referral completion by disability status, adjusted

  Logistic model, 1,581 consenting cases with complete covariates.
  Adjusted for case category, age band, sex, service requested, department.

  Average marginal effect   -19.4 points   95% CI -26.1 to -13.2
                                           (400 bootstrap resamples, seed 20260729)
  Predicted completion      26.1% with a disability reported
                            45.5% without
```

```
Applied to the 197 cases reporting a disability, the gap is 38 cases that
did not reach a service and would have at the comparator rate.

The gap is not constant: it is 22.7 points for health referrals, where
completion is common, and 15.2 points for livelihood support, where it is
not. The odds ratio (0.39) is the same on both.

Observational. Cases were not randomised, and the model adjusts only for
what the register records.
```

L'avant-dernier paragraphe est celui qui empêche de surappliquer la moyenne. Un nombre unique est ce qui sera cité ; la phrase qui dit où il vaut et où il ne vaut pas est ce qui garde la citation honnête.

4.7 La suite

Chaque covariable jusqu'ici a été choisie parce qu'elle paraissait pertinente. La leçon suivante en ajoute une qui est pertinente, défendable et fausse — une variable située sur le chemin causal qui retire 42 % de l'écart tout en améliorant chaque statistique d'ajustement.

Troisième partie

Quelles covariables ont leur place

CHAPITRE 5

La covariable qui améliore chaque statistique et détruit la réponse

Leçon 5 sur 8 · 150 min

Ajoutez le fait qu'une orientation ait été émise et l'AIC chute de 595, le pseudo-R^2 quadruple, et 42,5 % de l'écart selon le handicap disparaît. Tout critère de sélection de modèle dit de la garder. Toute considération causale dit qu'elle doit sortir, et aucune statistique d'ajustement ne vous dira laquelle a raison.

5.1 Une covariable qui paraît manifestement juste

La filière d'orientation comporte une porte. Un cas est consenti, puis une orientation est *émise*, puis elle est acceptée, puis un service est atteint. `referral_made` est enregistré, il est fortement lié à l'aboutissement, et l'ajouter au modèle est l'étape naturelle.

```python
import statsmodels.formula.api as smf

BASE = ("completed ~ disability + case_category + age_band + sex"
        " + service_requested + admin1")

adjusted = smf.logit(BASE, data=d).fit(disp=0)
plus_gate = smf.logit(BASE + " + referral_made", data=d).fit(disp=0)

for name, m in [("adjusted", adjusted), ("+ referral_made", plus_gate)]:
    print(f"{name:16} AIC {m.aic:7.1f}   pseudo-R2 {m.prsquared:.3f}")
```

```r
adjusted  <- glm(completed ~ disability + case_category + age_band + sex +
                   service_requested + admin1, data = d, family = binomial())
plus_gate <- update(adjusted, . ~ . + referral_made)
AIC(adjusted, plus_gate)
```

Modèle	Rapport de cotes	Effet marginal moyen	AIC	Pseudo-R^2
Brut	0,430	−19,05 pts	2138,5	0,012
Ajusté	0,388	**−19,41 pts**	1999,6	0,089
+ `referral_made`	0,494	**−11,15 pts**	**1404,2**	**0,365**

L'AIC s'améliore de 595 points. Le pseudo-R^2 quadruple. Et 42,5 % de l'effet s'évanouit.

Toute procédure automatique de sélection de modèle jamais écrite garderait cette variable. La sélection pas à pas la garde, l'AIC la garde, l'exactitude validée croisée la garde, et toutes répondent à une question que personne n'a posée.

5.2 Pourquoi elle retire l'effet

```python
print(d.groupby("disability")["referral_made"].agg(["mean", "size"]).round(4))
```

```
d |> summarise(made = mean(referral_made), n = n(), .by = disability)
```

Une orientation est émise pour 71,5 % des cas sans handicap signalé et 53,8 % des cas avec.

Ce n'est pas une différence parasite à neutraliser — **cela fait partie de ce qui est mesuré.** Les cas signalant un handicap atteignent moins souvent un service *en partie parce qu'une orientation est moins souvent émise pour eux au départ.*

Conditionner sur `referral_made` demande : parmi les cas où une orientation a été émise, reste-t-il un écart ? La réponse est oui, 11,2 points — un nombre réel et utile sur l'étape d'*acceptation.* Ce n'est pas l'effet du handicap sur le fait d'atteindre un service, et le rapporter comme tel sous-estime cet effet de 42 %.

Une covariable située sur le chemin causal entre exposition et résultat est un médiateur, et ajuster dessus retire exactement la part de l'effet qui passe par elle.

5.3 Quatre sortes de covariables, et ce que chacune fait

La décision porte sur la structure causale, non sur des statistiques, et elle doit être prise avant d'ajuster le modèle.

Sorte	Se situe	Ajuster ?	Si vous vous trompe
Facteur de confusion	Cause l'exposition et le résultat	**Oui**	Estimation biaisée, dan
Médiateur	Entre l'exposition et le résultat	**Non** (pour un effet total)	Effet sous-estimé
Collisionneur	Causé par l'exposition et le résultat	**Jamais**	Biais créé là où il n'y e
Cause concurrente	Ne cause que le résultat	Facultatif	Précision seulement

Seule la première ligne est une raison d'ajouter une variable. La quatrième est une raison pour laquelle vous pouvez en ajouter une, et elle achète une erreur-type plus petite plutôt qu'une estimation différente. Les deux du milieu sont des raisons de ne pas le faire.

La sorte d'une variable n'est pas visible dans les données. Facteur de confusion, médiateur et collisionneur produisent tous un coefficient qui bouge quand on ajuste, et tous trois améliorent l'ajustement s'ils prédisent le résultat. **Le seul moyen de les distinguer est de savoir ce qui est venu avant et ce qui a causé quoi** — d'où le schéma tracé avant l'ajustement du modèle.

5.4 Le tracer avant d'ajuster

Trois flèches, tirées de ce que vous savez du fonctionnement de la filière.

```
disability ------------------------------> completion
    |                                          ^
    +-------> referral made ----------------------+

department --> disability        department --> completion
```

Le département est un facteur de confusion — le lieu où naît un cas influe à la fois sur qui est enregistré avec un handicap et sur la qualité du système de services — il a donc sa place dans le modèle. **L'émission de l'orientation est un médiateur** — le handicap l'influence et elle influence l'aboutissement — elle n'y a donc pas sa place.

```
TOTAL_EFFECT = ["disability", "case_category", "age_band", "sex",
                "service_requested", "admin1"]        # confounders only
MECHANISM = TOTAL_EFFECT + ["referral_made"]          # a different question
```

```
# Two named model specifications, two questions, both legitimate.
```

Nommez les modèles d'après la question plutôt que d'après les variables. Un modèle appelé `MECHANISM` ne sera pas rapporté par accident comme l'effet total, et un relecteur voit lequel était visé.

5.5 Le collisionneur, qui est pire

Un médiateur sous-estime un effet réel. Un collisionneur en fabrique un qui n'existe pas, et le mécanisme est plus subtil.

Se restreindre aux cas clos est la façon la plus courante dont cela arrive ici, parce qu'un cas clos est un cas où *quelque chose* a été résolu — ce qui est en aval à la fois du handicap et de l'aboutissement.

```
closed = d[d["case_status"].isin(["closed-resolved", "closed-lost-contact"])]
print(f"closed cases: {len(closed)}")
print(closed.groupby("disability")["completed"].agg(["mean", "size"]).round(4))
```

```
d |> filter(case_status %in% c("closed-resolved", "closed-lost-contact")) |>
  summarise(rate = mean(completed), n = n(), .by = disability)
```

Échantillon	n	Écart (effet marginal moyen)
Tous les cas consentis	1 581	−19,4 points
Cas clos seulement	**860**	**−21,1 points**

Se restreindre aux cas clos porte l'écart à 21,1 points, et le mouvement est produit par la restriction plutôt que par quoi que ce soit concernant le handicap. Les cas ouverts — ceux encore en cours — sont exclus sur un critère que les deux variables influencent.

Sélectionner des lignes, c'est ajuster sur une variable. Écarter les cas ouverts, ne garder que les formulaires complets, n'analyser que les ménages ayant répondu à tout : chacun est un conditionnement, et chacun peut être un collisionneur. « Nous avons restreint l'analyse aux cas clos par souci d'exhaustivité » est une phrase qui change une estimation sans que personne remarque qu'il s'agissait d'un choix de modélisation.

5.6 Rapportez les deux, et dites lequel est lequel

La médiation vaut d'être rapportée quand le mécanisme est la question du programme, ce qui est le cas ici : le cours de protection situait l'écart selon le handicap à l'émission et à l'acceptation, et cette décomposition en est la version chiffrée.

```
Referral completion by disability status, decomposition

  Total gap                                  -19.4 points
    of which runs through referral-making    -8.3 points  (42.5%)
    remaining, among referrals made          -11.2 points
```

```
Referrals are made for 53.8% of cases reporting a disability against 71.5%
of others. Both gates contribute; neither explains the other away.

The -11.2 figure is conditional on a referral having been made and must not
be reported as the effect of disability on reaching a service.
```

La dernière ligne est celle qui garde la décomposition honnête. Les deux nombres sont réels, ils répondent à des questions différentes, et le mode d'échec n'est pas de calculer le mauvais — c'est de calculer le bon et de l'étiqueter comme l'autre.

5.7 La suite

Chaque modèle de ce cours a jusqu'ici supposé que chaque ligne était une observation indépendante. La leçon suivante donne au modèle un terme pour ce à l'intérieur de quoi les lignes sont regroupées, et trouve trois chemins honnêtes vers la même réponse là où le naïf était seul de son côté.

CHAPITRE 6

Trois réponses honnêtes et une qui est seule de son côté

Leçon 6 sur 8 · 150 min

MCO naïfs, erreurs-types robustes aux grappes, constante aléatoire et agrégation au niveau école donnent des erreurs-types de 0,88, 1,46, 1,48 et 1,53 point. Trois d'entre elles concordent et la quatrième est celle que produit tout ajustement par défaut.

6.1 Le même coefficient, quatre erreurs-types

Le cours de statistiques a établi que la cantine avait été attribuée à 24 écoles et mesurée sur 1 200 enfants. Voici ce que produit chaque façon de traiter cela.

```python
import pandas as pd
import statsmodels.formula.api as smf

naive = smf.ols("rate ~ feeding_programme", data=per_student).fit()
robust = naive.get_robustcov_results(cov_type="cluster",
                                     groups=per_student["school_id"])

mixed = smf.mixedlm("rate ~ feeding_programme", data=per_student,
                    groups=per_student["school_id"]).fit()

by_school = (per_student.groupby(["school_id", "feeding_programme"])["rate"]
             .mean().reset_index())
aggregated = smf.ols("rate ~ feeding_programme", data=by_school).fit()
```

```r
library(lme4); library(sandwich); library(lmtest)

naive      <- lm(rate ~ feeding_programme, data = per_student)
robust     <- coeftest(naive, vcov = vcovCL, cluster = ~school_id)
mixed      <- lmer(rate ~ feeding_programme + (1 | school_id), data = per_student)
aggregated <- lm(rate ~ feeding_programme, data = by_school)
```

Approche	Coefficient	ET	t	n
MCO naïfs	+4,93 pts	**0,88**	5,63	1 200
ET robustes aux grappes	+4,93 pts	1,46	3,38	1 200
Constante aléatoire	+5,10 pts	1,48	3,44	1 200
MCO au niveau école	+5,21 pts	1,53	3,41	24

Trois des quatre concordent et une non. Les coefficients s'étalent sur 0,3 point ; les trois erreurs-types honnêtes sur 0,07 ; et la naïve est 40 % plus petite que n'importe laquelle d'entre elles.

C'est la forme à attendre. Le regroupement en grappes change rarement beaucoup une estimation et change couramment beaucoup sa précision, et les trois corrections sont trois chemins vers la même destination plutôt que trois réponses concurrentes.

6.2 Ce qu'ajoute une constante aléatoire

Les erreurs-types robustes aux grappes corrigent l'inférence et ne disent rien de la structure. Une constante aléatoire l'estime.

```
print(mixed.summary())
print(f"between-school variance: {float(mixed.cov_re.iloc[0, 0]):.5f}")
print(f"residual variance:       {mixed.scale:.5f}")
```

```
VarCorr(mixed)
```

Modèle	Variance inter-écoles	Variance résiduelle	CCI
Constante seule	0,00142	0,02042	**0,065**
+ cantine	**0,00081**	0,02042	0,038

La cantine explique 42,8 % de la variance inter-écoles et rien de la variance intra-école, ce qui est exactement ce que devrait faire un programme de niveau école et vaut d'être vérifié parce que c'est une façon d'attraper un modèle mal spécifié.

Le CCI à constante seule de 0,065 est la même quantité que le cours de statistiques calculait à la main depuis les carrés moyens, avec 0,060. **Deux estimateurs, deux réponses légèrement différentes, une conclusion** — environ six pour cent de la variation de présence est entre écoles, ce qui, à cinquante enfants par école, suffit à quadrupler la variance d'une comparaison naïve.

6.3 Choisir parmi les trois

Les trois corrections ne sont pas interchangeables, et le choix se décide généralement par le nombre de grappes et par ce qui varie à quel niveau.

Employez	Quand	Coû
Agréger à la grappe	Peu de grappes ; l'exposition est de niveau grappe	Une
ET robustes aux grappes	Beaucoup de grappes (30+) ; vous voulez le modèle individuel	Peu
Constante aléatoire	Vous voulez les composantes de variance, ou des prédicteurs aux deux niveaux	Supp

Avec 24 grappes, agrégez ou ajustez la constante aléatoire, et dites laquelle. Le sandwich robuste est l'outil standard et c'est celui qui se comporte le plus mal ici : son asymptotique exige plus de groupes que ce protocole n'en a, et il s'exécutera sans vous prévenir.

Quand l'exposition ne varie qu'entre grappes, les trois convergent, ce que montre le tableau ci-dessus. Elles se séparent quand un prédicteur varie *à l'intérieur* des grappes — et la constante aléatoire fait alors un travail que les deux autres ne peuvent pas faire.

6.4 Des prédicteurs à deux niveaux

C'est là qu'un modèle multiniveau cesse d'être une correction et devient un modèle.

```
two_level = smf.mixedlm(
    "rate ~ feeding_programme + age_years + disability_reported",
    data=d, groups=d["school_id"]).fit()
print(two_level.summary().tables[1])
```

```
lmer(rate ~ feeding_programme + age_years + disability_reported +
       (1 | school_id), data = d)
```

`feeding_programme` ne varie qu'entre écoles. `age_years` et `disability_reported` varient à l'intérieur. Un seul modèle peut porter les deux, et la constante aléatoire est ce qui le permet : les prédicteurs intra-école sont estimés depuis la variation intra-école et le prédicteur inter-écoles depuis la variation inter-écoles, sans que l'un contamine l'erreur-type de l'autre.

Le mode d'échec est de mettre une exposition de niveau grappe dans un modèle sans terme de grappe. C'est la ligne naïve du premier tableau, et c'est le comportement par défaut de tout logiciel.

6.5 Trois niveaux, et quand s'arrêter

Les enfants sont dans des ménages, les ménages dans des villages, les villages dans des districts.

```
# Two grouping levels: households nested within enumeration areas.
smf.mixedlm("outcome ~ x", data=survey,
            groups=survey["ea_id"], re_formula="1").fit()
```

```
lmer(outcome ~ x + (1 | ea_id / household_id), data = survey)
```

Ajoutez un niveau quand quelque chose est attribué ou mesuré à ce niveau et que vous en avez assez d'unités. Trois niveaux avec six districts au sommet n'estimeront pas une variance de district digne d'être rapportée — le niveau supérieur a besoin d'assez de groupes pour la même raison que la leçon sur les comparaisons multiples avait besoin d'assez de tests.

N'ajoutez pas un niveau par souci d'ordre. Un niveau à quatre groupes ajoute un paramètre inestimable et le faux sentiment d'avoir traité quelque chose.

6.6 Rapportez-le en entier

```
Attendance and school feeding, multilevel analysis

  Linear mixed model, 1,200 students in 24 schools.
  rate ~ feeding_programme + (1 | school_id)

  Feeding programme    +5.10 points   95% CI +2.20 to +8.00

  Between-school SD    0.028      Residual SD  0.143
  ICC 0.038 with the programme term; 0.065 without it, so the programme
  accounts for 43% of the between-school variance.

  The naive single-level model gives +4.93 points with a standard error of
  0.88 rather than 1.48. It is not reported: it treats 50 children in one
  school as 50 independent observations.

  Cluster-robust standard errors on the single-level model (+4.93, SE 1.46)
  and school-level aggregation (+5.21, SE 1.53) give the same answer. With
  24 clusters the robust sandwich is at the edge of its assumptions and is
  reported as a check rather than as the headline.
```

```
Observational. Schools were not randomised into the programme.
```

Rapporter les trois fait quatre lignes et coupe court à l'objection évidente — que le résultat dépendrait de la méthode. Il n'en dépend pas, et le montrer coûte moins cher que d'en débattre.

6.7 La suite

Cette leçon a accordé le modèle à la façon dont le *programme* a été attribué. La suivante l'accorde à la façon dont l'*échantillon* a été tiré, sur une enquête aux strates délibérément inégales — et trouve des coefficients qui traversent le seuil de significativité dans les deux sens quand les poids entrent.

Quatrième partie

Le plan et l'ajustement

CHAPITRE 7

Deux coefficients traversent le seuil et un le retraverse

Leçon 7 sur 8 · 150 min

La strate rurale isolée pèse 11 % des ménages et un tiers des entretiens. Ajustez le modèle sans pondération et l'élevage ressemble à un facteur de risque significatif ; pondérez-le et il cesse de l'être, tandis que la taille du ménage et le statut de retourné se mettent à l'être.

7.1 Le plan sous lequel l'échantillon a été tiré

```python
import pandas as pd

survey = pd.read_csv("household-survey-2025.v1.csv")
frame = pd.read_csv("household-survey-frame-2025.v1.csv")

print(frame.groupby("stratum")["households"].sum())
print(survey.groupby("stratum").size())
```

```r
library(dplyr)
frame |> summarise(households = sum(households), .by = stratum)
survey |> count(stratum)
```

Strate	Ménages du sondage	Part	Entretiens	Part
Rurale accessible	28 958	51,3 %	337	33,8 %
Urbaine	21 270	37,7 %	316	31,7 %
Rurale isolée	**6 200**	**11,0 %**	**343**	**34,4 %**

La strate rurale isolée pèse 11,0 % de la population et 34,4 % de l'échantillon. C'est un choix de plan délibéré — le cours sur les enquêtes explique pourquoi on suréchantillonne une petite strate dont on a besoin d'estimations — et cela signifie que chaque nombre non pondéré tiré de ce fichier décrit une population qui n'existe pas.

7.2 Ce que les poids font à une estimation

```python
stratum_households = frame.groupby("stratum")["households"].sum()
base_weight = stratum_households / (25 * 14)
interviewed = frame[frame["selected"]].set_index("ea_id")["households_interviewed"]

survey["weight"] = (survey["stratum"].map(base_weight)
                    * 14 / survey["ea_id"].map(interviewed))
assert abs(survey["weight"].sum() - frame["households"].sum()) < 1

unweighted = survey["food_insecure"].mean()
weighted = np.average(survey["food_insecure"], weights=survey["weight"])
print(f"unweighted {unweighted:.1%}   weighted {weighted:.1%}")
```

```r
survey <- survey |>
  mutate(weight = base_weight[stratum] * 14 / households_interviewed)

c(unweighted = mean(survey$food_insecure),
  weighted   = weighted.mean(survey$food_insecure, survey$weight))
```

32,8 % sans pondération, 29,1 % avec. La strate rurale isolée a l'insécurité alimentaire la plus élevée (46,4 %), et la surreprésenter trois fois tire le chiffre national vers le haut de 3,7 points.

Une régression ajustée sur ce fichier sans poids hérite exactement de ce problème, et cela ne se corrige pas en ajoutant `stratum` comme covariable — cela change la question de « quelle est l'association dans la population » à « quelle est-elle à l'intérieur des strates ». Les deux sont légitimes ; une seule est ce que signifie une estimation nationale.

7.3 Le même modèle, pondéré et non

```python
import statsmodels.api as sm
import statsmodels.formula.api as smf

FORMULA = ("food_insecure ~ improved_water_source + displacement_status"
           " + main_livelihood + household_size")

plain = smf.glm(FORMULA, data=d, family=sm.families.Binomial()).fit(
    cov_type="cluster", cov_kwds={"groups": d["ea_id"]})

weighted = smf.glm(FORMULA, data=d, family=sm.families.Binomial(),
                   freq_weights=d["weight"]).fit(
    cov_type="cluster", cov_kwds={"groups": d["ea_id"]})
```

```r
library(survey)

design <- svydesign(ids = ~ea_id, strata = ~stratum, weights = ~weight,
                    data = survey, nest = TRUE)
svyglm(food_insecure ~ improved_water_source + displacement_status +
         main_livelihood + household_size, design = design,
       family = quasibinomial())
```

Terme	RC non pondéré	RC pondéré	Bouge ?
Source d'eau améliorée	0,75 (0,55–1,03)	0,74 (0,51–1,08)	—
Déplacement : retourné	0,56 (0,25–1,26)	**0,39 (0,15–0,97)**	**devient significatif**
Moyen d'existence : agriculture	1,55 (1,03–2,34)	1,69 (1,04–2,75)	—
Moyen d'existence : élevage	**1,51 (1,04–2,18)**	1,26 (0,73–2,18)	**cesse d'être significatif**
Moyen d'existence : salarié	0,41 (0,22–0,77)	0,45 (0,24–0,87)	—
Taille du ménage	1,04 (0,98–1,11)	**1,08 (1,01–1,16)**	**devient significatif**

Trois coefficients changent de côté du seuil, dans les deux sens. La pondération n'est pas une correction qui affaiblit tout ou renforce tout — elle repondère les ménages depuis lesquels l'association est estimée, et la réponse bouge partout où ces ménages diffèrent.

Les moyens d'existence par l'élevage sont concentrés dans la strate isolée suréchantillonnée. Sans pondération, ils portent un tiers de l'influence de l'échantillon ; pondérés, ils portent leurs 11 % réels, et l'association se dissipe.

7.4 Poids, grappes et strates sont trois choses distinctes

Ils arrivent ensemble et font trois travaux différents, et un modèle peut aisément en réussir un et rater les deux autres.

Élément du plan	Corrige	Si vous l'omettez
Poids	L'estimation	Les estimations décrivent l'échantillon, non la population
Grappes	L'erreur-type	Erreurs-types trop petites, comme à la leçon précédente
Strates	L'erreur-type	Erreurs-types un peu trop grandes — l'erreur conservatrice

Les poids changent le coefficient ; les grappes et les strates changent son intervalle. C'est la même distinction qu'ajustement contre grappes à la leçon 2, arrivant par une autre porte, et c'est celle à retenir.

```
print(f"clusters: {d['ea_id'].nunique()} enumeration areas")
print(f"strata:   {d['stratum'].nunique()}")
print(f"weights:  {d['weight'].min():.1f} to {d['weight'].max():.1f}")
```

```
# survey::svydesign wants all three; a glm with weights= handles only one.
```

`glm(..., weights=)` n'est pas une régression pondérée par le plan. Elle applique les poids et calcule les erreurs-types comme si chaque observation était indépendante et les poids des effectifs. Employez `survey::svyglm` en R, et en Python passez à la fois `freq_weights` et une covariance par grappes — ou acceptez que l'intervalle soit faux et dites-le.

7.5 Quand laisser les poids de côté

Deux cas, et aucun n'est « les poids ont fait disparaître mon résultat ».

Une question intra-strate. « Parmi les ménages ruraux isolés, une source d'eau améliorée va-t-elle avec moins d'insécurité alimentaire ? » est une question sur cette strate, et le poids de sondage à l'intérieur d'une strate est de toute façon quasi constant.

Un modèle dont les covariables contiennent tout ce dont les poids sont faits. Si la strate et la probabilité de sélection sont entièrement captées par les covariables, les estimations pondérée et non pondérée répondent à la même question et la non pondérée est plus précise. **Vérifiez-le plutôt que de le supposer** : ajustez les deux, et si elles diffèrent sensiblement, les covariables n'ont pas capté le plan.

```
print(f"unweighted {plain.params['main_livelihood[T.livestock]']:.3f}")
print(f"weighted   {weighted.params['main_livelihood[T.livestock]']:.3f}")
```

```
# A large gap between the two is evidence the design is not in the model.
```

Un grand écart entre l'ajustement pondéré et le non pondéré est en soi un diagnostic, et c'est la vérification à faire avant de décider que les poids sont facultatifs.

7.6 Rapportez-le en entier

```
Food insecurity, household survey 2025

  976 households with complete covariates (of 996 interviewed),
```

```
75 enumeration areas, 3 strata.
Survey-weighted logistic regression; weights sum to 56,428 households,
which is the frame total. Standard errors account for clustering by
enumeration area and for stratification.

Weighted prevalence      29.1%   (unweighted 32.8%)

Adjusted odds ratios (weighted):
  Improved water source          0.74   95% CI 0.51 to 1.08
  Returnee household             0.39   95% CI 0.15 to 0.97
  Farming livelihood             1.69   95% CI 1.04 to 2.75
  Livestock livelihood           1.26   95% CI 0.73 to 2.18
  Salaried livelihood            0.45   95% CI 0.24 to 0.87
  Household size, per person     1.08   95% CI 1.01 to 1.16

The unweighted model makes livestock livelihoods significant (OR 1.51,
95% CI 1.04 to 2.18) and household size not. The rural remote stratum is
11% of households and 34% of interviews, and livestock is concentrated
there; the unweighted result is an artefact of the sampling design.

Odds ratios are reported because food insecurity is 29% and the ratios are
moderate; the weighted risk difference for salaried livelihoods is -14.2
points against the observed mix of livelihoods, which is the number in
the summary.
```

Le paragraphe qui nomme l'artefact est ce qu'un relecteur vérifiera en premier, et l'écrire vous-même vaut mieux que de le laisser trouver.

7.7 La suite

Chaque modèle jusqu'ici a été ajusté puis lu. La dernière leçon en ajuste un qui explique trois pour cent de son résultat, ne fait pas mieux que deviner, et constitue le constat le plus utile de son rapport — parce que ce qu'il échoue à prédire *est* la réponse opérationnelle.

CHAPITRE 8

Un R^2 de 0,03, et le constat le plus utile du rapport

Leçon 8 sur 8 · 150 min

Douze caractéristiques de ménage expliquent trois pour cent de la variation du PB des enfants. Cibler le dépistage sur les 20 % les moins bien prédits trouve 31 % des enfants malnutris, contre 20 % au hasard. Le modèle a échoué, et son échec est la réponse opérationnelle.

8.1 Un modèle bâti pour répondre à une question de ciblage

Un programme veut dépister moins d'enfants. Si les caractéristiques des ménages prédisaient quels enfants sont malnutris, le dépistage pourrait viser les ménages qui en abritent le plus probablement, et l'enquête porte toutes les variables qu'une telle règle emploierait.

```python
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf

survey = pd.read_csv("household-survey-2025.v1.csv")
d = survey.dropna(subset=["child_muac_mm", "child_age_months", "child_sex",
                          "household_size", "displacement_status",
                          "main_livelihood", "food_insecure",
                          "improved_water_source"])

model = smf.ols(
    "child_muac_mm ~ child_age_months + child_sex + household_size"
    " + displacement_status + main_livelihood + food_insecure"
    " + improved_water_source", data=d).fit()

print(f"n = {int(model.nobs)}   R2 = {model.rsquared:.4f}"
      f"   adjusted R2 = {model.rsquared_adj:.4f}")
print(f"residual SD {np.sqrt(model.scale):.2f}  outcome SD {d['child_muac_mm'].std():.2f}"
    )
```

```r
model <- lm(child_muac_mm ~ child_age_months + child_sex + household_size +
              displacement_status + main_livelihood + food_insecure +
              improved_water_source, data = d)
summary(model)
```

n = 641. R^2 = 0,030, R^2 ajusté = 0,011. Écart-type résiduel 15,95 mm contre un écart-type du résultat de 16,04 mm.

Le modèle a retiré moins d'un millimètre des seize dont il partait. Un seul coefficient se distingue de zéro.

Terme	Coefficient	IC à 95 %
Insécurité alimentaire	**−4,70 mm**	−7,45 à −1,95
Moyen d'existence salarié	−3,49 mm	−8,44 à +1,45
Moyen d'existence agricole	−2,76 mm	−6,16 à +0,64
Âge de l'enfant, par mois	−0,03 mm	−0,11 à +0,05
Source d'eau améliorée	−1,09 mm	−3,76 à +1,58

8.2 Ne le supprimez pas

L'instinct est d'écarter le modèle, d'essayer plus de variables, ou de se tourner vers quelque chose de non linéaire. Les trois sont faux ici, et la raison est que **la question n'était pas « puis-je modéliser le PB » — c'était « puis-je cibler le dépistage ».**

```python
d = d.assign(predicted=model.fittedvalues, mam=(d["child_muac_mm"] < 125))
for frac in (0.20, 0.30, 0.50):
    cut = d["predicted"].quantile(frac)
    caught = d.loc[d["predicted"] <= cut, "mam"].sum()
    print(f"screen worst {frac:.0%}: catches {caught}/{d['mam'].sum()}"
          f" = {caught / d['mam'].sum():.1%} of cases")
```

```r
d |> mutate(predicted = fitted(model), mam = child_muac_mm < 125) |>
  arrange(predicted) |>
  summarise(caught = mean(head(mam, n() * 0.2)) * sum(mam))
```

Règle de dépistage	Enfants dépistés	PB < 125 mm trouvés
20 % les moins bien prédits	129	28 sur 90 = **31,1 %**
30 % les moins bien prédits	193	41 sur 90 = 45,6 %
50 % les moins bien prédits	321	54 sur 90 = 60,0 %
20 % au hasard	129	**20 %** attendus

Le ciblage par le modèle trouve 31 % des enfants malnutris en en dépistant 20 %. Dépister 20 % au hasard en trouve 20 %. Le modèle fait onze points mieux que le hasard et manque sept enfants sur dix parmi ceux qu'un programme existe pour atteindre.

C'est cela, le constat. Non pas « le modèle n'a pas marché » — *les caractéristiques des ménages n'identifient pas assez bien les enfants malnutris pour cibler le dépistage, donc le dépistage doit rester systématique.* Un programme peut agir sur cette phrase, elle coûte une somme réelle, et elle est étayée par un modèle de R² 0,03.

8.3 Des diagnostics qui changent une décision

Quatre vérifications, chacune assortie d'une décision plutôt que d'un seuil.

Des valeurs ajustées hors de la plage possible. Pour un résultat binaire ajusté par régression linéaire, c'est immédiat.

```python
lpm = smf.ols("completed ~ disability + case_category + age_band + sex"
              " + service_requested + admin1", data=referrals).fit()
print(f"fitted values from {lpm.fittedvalues.min():.3f}"
      f" to {lpm.fittedvalues.max():.3f}")
print(f"outside [0, 1]: {((lpm.fittedvalues < 0) | (lpm.fittedvalues > 1)).sum()}")
```

```r
range(fitted(lm(completed ~ ., data = referrals)))
```

Huit des 1 581 probabilités ajustées tombent sous zéro, la plus basse à −0,060. Le modèle logistique sur les mêmes données reste entre 0,065 et 0,738. **La décision : employez le logistique quand les prédictions comptent, et le modèle linéaire de probabilité quand seule la différence moyenne compte** — le coefficient du modèle linéaire est directement une différence de risques, ce qui explique sa survie en économie.

Les résidus contre les valeurs ajustées. Cherchez une forme en éventail, qui signifie que la dispersion dépend du niveau.

```python
import matplotlib.pyplot as plt
plt.scatter(model.fittedvalues, model.resid, s=6)
```

```r
plot(model, which = 1)
```

La décision : l'hétéroscédasticité ne biaise pas le coefficient, elle biaise l'erreur-type — le remède est donc une erreur-type robuste, non un résultat transformé.

L'influence. Une poignée de lignes peut porter un coefficient.

```python
influence = model.get_influence().cooks_distance[0]
print(f"rows with Cook's D > 4/n: {(influence > 4 / len(d)).sum()}")
```

```r
sum(cooks.distance(model) > 4 / nobs(model))
```

La décision : regardez les lignes signalées avant de faire quoi que ce soit. Les onze ménages du cours EAH avec une erreur d'unité sont exactement le genre de ligne qui ressort ici, et la réponse est de corriger les données, non de sous-pondérer le point.

La linéarité d'un prédicteur continu. Ajustez-le par classes et voyez si les coefficients progressent régulièrement.

```python
d["age_band"] = pd.cut(d["child_age_months"], [0, 12, 24, 36, 48, 60])
print(smf.ols("child_muac_mm ~ C(age_band)", data=d).fit().params.round(2))
```

```r
lm(child_muac_mm ~ cut(child_age_months, c(0, 12, 24, 36, 48, 60)), data = d)
```

La décision : si les classes ne progressent pas régulièrement, le terme linéaire répond à la mauvaise question, et le découpage en classes est généralement une meilleure réponse qu'un polynôme, parce que les classes sont interprétables par ceux qui lisent le rapport.

8.4 Ce que le R^2 est et n'est pas

Le R^2 est la part de variance dont le modèle rend compte. Dans des données de programme il est couramment petit, et c'est un fait sur les gens, non sur l'analyste.

Ce n'est pas une mesure de la justesse d'un coefficient. Un essai randomisé avec un effet de traitement tranché peut avoir un R^2 de 0,02, parce que la variation individuelle noie une vraie différence moyenne. Juger une estimation causale par le R^2 est une erreur de catégorie.

C'est une mesure de la valeur des prédictions. C'est l'usage ci-dessus, et c'est l'usage honnête : la question du ciblage est une question de prédiction, et le R^2 y a répondu.

Le R^2 ajusté pénalise les termes ajoutés, et il est passé de 0,030 à 0,011 ici — l'écart entre les deux mesure combien des douze variables payaient leur place. Presque aucune.

8.5 Rapportez-le en entier

```
Predicting child MUAC from household characteristics

  Linear model, 641 children with complete data.
  R-squared 0.030, adjusted 0.011. Residual SD 15.95 mm against an outcome
  SD of 16.04 mm.

  Only food insecurity is distinguishable from zero: -4.70 mm (95% CI -7.45
  to -1.95). Household size, displacement status, water source, child sex
  and child age are not.

  Used as a targeting rule, screening the 20% of children the model ranks
  worst would find 31% of the children with MUAC below 125 mm, against 20%
  expected from screening 20% at random. Screening half the children would
  find 60%.

  Recommendation: do not target screening on household characteristics. The
  survey's 14.0% prevalence of MUAC below 125 mm is not concentrated in
  households that a registration form can identify, and blanket screening
  remains the only rule that reaches the caseload.

  This model is reported because it does not fit. A negative result about
  targeting is a budget decision, and dropping the model would have left the
  proposal to assume targeting works.
```

Le dernier paragraphe est la raison d'être de cette section. Un modèle qui s'ajuste mal est une preuve, et le tiroir où il finit d'ordinaire est l'endroit où l'hypothèse d'un programme survit sans être contestée.

8.6 La suite

C'est le cours. Huit leçons, une idée : **un coefficient est une comparaison, et le travail consiste à dire laquelle, entre quelles unités, ajustée sur quoi, avec une erreur-type qui épouse la façon dont les données sont arrivées.**

Le laboratoire met les quatre décisions dans une seule analyse. Puis *Méthodes d'évaluation d'impact* prend en charge la question que ce cours a différée dans le dernier paragraphe de chaque leçon — laquelle de ces comparaisons peut s'écrire comme une cause.

À propos de cette édition

Ce PDF est produit à partir de la même source que la version web de la formation ; les deux ne peuvent donc pas diverger. L'édition en ligne comporte le code exécutable, les jeux de données téléchargeables et les corrections publiées depuis la production de ce fichier.

> Tous les jeux de données cités dans cette formation sont synthétiques ou entièrement anonymisés. Aucun enregistrement ne renvoie à une personne réelle.

Cette formation est publiée en anglais et en français. L'édition française est une adaptation professionnelle reprenant la terminologie employée par l'UNICEF, l'OMS, le PAM, OCHA et les ministères nationaux — et non une traduction littérale.

Consulter et exécuter la formation en ligne sur data-analysis.cassion.dev/fr/cours/regression-for-programmes

Généré le 29 juillet 2026.