Canadian Journal of Regional Science
Revue canadienne des sciences régionales

Tests de randomisation : une façon plus flexible de tester la significativité avec le logiciel Stata

Jean Dubé, Diego Cardenas and Marie-Pier Champagne

Volume 46, Number 3, 2023

URI: https://id.erudit.org/iderudit/1108482ar
DOI: https://doi.org/10.7202/1108482ar

See table of contents

Publisher(s)

Canadian Regional Science Association / Association canadienne des sciences régionales

ISSN

0705-4580 (print)
1925-2218 (digital)

Explore this journal

Cite this article

Dubé, J., Cardenas, D. & Champagne, M.-P. (2023). Tests de randomisation : une façon plus flexible de tester la significativité avec le logiciel Stata. *Canadian Journal of Regional Science / Revue canadienne des sciences régionales*, *46*(3), 50–56. https://doi.org/10.7202/1108482ar

Article abstract

In the field of regional science, several empirical conclusions rely on significance tests. The statistical significance of parameter estimates in regression models is particularly important for researchers. However, the calculation of classical significance tests (t-test of F-test) relies on assumptions that, if not respected, may lead to bias in the tests themselves. The purpose of this technical note is to present an alternative approach, a non-parametric test, that allows us to test for significance using a randomization test. Such an approach is more flexible than conventional parametric tests and its application is relatively simple. To show the relevance of this alternative approach, we present a detailed application in Stata using synthetic data.

CANADIAN JOURNAL OF REGIONAL SCIENCE
REVUE CANADIENNE DES SCIENCES RÉGIONALES
CANADIAN REGIONAL SCIENCE ASSOCIATION
ASSOCIATION CANADIENNE DE SCIENCE RÉGIONALE

TESTS DE RANDOMISATION : UNE FAÇON PLUS FLEXIBLE DE TESTER LA SIGNIFICATIVITÉ AVEC LE LOGICIEL STATA

Jean Dubé, Diego Cardenas, Marie-Pier Champagne

Jean Dubé
Université Laval
2325 rue des Bibliothèques
Pavillon Félix-Antoine-Savard
Québec, Québec, Canada, G1V 0A6
jean.dube@esad.ulaval.ca

Diego Cardenas
Candidat au Ph.D. en ATDR
Université Laval

Marie-Pier Champagne
Candidate au Ph.D. en ATDR
Université Laval

Soumission : 2023-04-18
Accepté : 2023-06-14

Résumé : En sciences régionales, plusieurs conclusions empiriques reposent sur des tests de significativité. La significativité des paramètres dans des modèles de régression revêt souvent une importance capitale pour les chercheurs. Or, le calcul des tests statistiques classiques (test-t ou test-F) repose sur un certain nombre d'hypothèses qui, si elles s'avèrent non-respectées, peuvent entraîner un biais dans les calculs des statistiques. Le but de cette note technique est de présenter une approche alternative, un test non-paramétrique, permettant de tester la significativité à partir d'un test de randomisation. Cette approche est plus flexible que les tests paramétriques conventionnels et son application est relativement simple. Afin de démontrer le potentiel de la méthode, une application détaillée dans le logiciel Stata est présentée sur la base de données fictives.

Mots clés : Significativité; Pseudo-significativité; Approche non-paramétrique; Tests de randomisation.

Abstract: In the field of regional science, several empirical conclusions rely on significance tests. The statistical significance of parameter estimates in regression models is particularly important for researchers. However, the calculation of classical significance tests (t-test of F-test) relies on assumptions that, if not respected, may lead to bias in the tests themselves. The purpose of this technical note is to present an alternative approach, a non-parametric test, that allows us to test for significance using a randomization test. Such an approach is more flexible than conventional parametric tests and its application is relatively simple. To show the relevance of this alternative approach, we present a detailed application in Stata using synthetic data.

Keywords: Significance; pseudo-significance; non-parametric approach; randomization tests.

INTRODUCTION

En analyses quantitatives, les tests de significativité occupent une place importante dans les conclusions que tirent les chercheurs. La plupart des études visent à vérifier s'il existe une relation entre une variable spécifique, dite variable d'intérêt ou indépendante, et une variable dépendante, mesurant une performance quelconque. Un test statistique doit permettre de rendre un jugement sur une hypothèse nulle (H_0: la variable indépendante n'a aucun effet sur la variable dépendante) par rapport à une hypothèse alternative (H_1: la variable indépendante a un effet sur la variable dépendante).

En régression linéaire, les applications reposent sur un test de significativité d'un paramètre, désigné par le test-t, ou encore sur un test de significativité global, désigné par le test-F (ou test du khi-deux). Simples, ces statistiques ont l'avantage de référer à des bases et des pratiques reconnues et largement documentées. Or, la validité des statistiques repose sur quelques hypothèses fondamentales liées à la forme du test.

Dans le cas d'un test de significativité d'un paramètre - test-t -, il faut premièrement que le coefficient estimé soit sans biais. Deuxièmement, la variance estimée doit être homogène, ou minimalement corrigée pour la présence d'hétéroscédasticité. Troisièmement, pour que le test puisse s'appliquer, les termes résiduels de la régression doivent être indépendants et identiquement distribués. La validité du test statistique est donc liée au comportement du numérateur et/ou du dénominateur de la statistique.[1]

D'autres alternatives existent afin d'éviter le recours à certaines hypothèses nécessaires à la réalisation des tests d'hypothèses. Une de ces alternatives consiste à développer des tests non-paramétriques. Dans ce contexte, le test de randomisation s'avère un outil intéressant.

L'origine des tests de randomisation est souvent attribuée à Fischer (1935). Or, il semble qu'il faille plutôt attribuer la paternité du concept à Pitman (1937) et Welch (1937) (pour un historique complet, voir Onghena, 2018). Fischer faisait essentiellement référence à l'échantillonnage aléatoire, alors que peu de passages de son œuvre font explicitement référence à la randomisation. Pitman (1937), pour sa part, fait explicitement référence au point que l'important dans un test de randomisation n'est pas la population, mais plutôt l'échantillon disponible. Le test de randomisation est valide pour tous types d'échantillons, et ce, peu importe la façon dont l'échantillon est obtenu (Edgington & Onghena, 2020).

Souvent utilisés de manière interchangeable, il existe une différence importante entre les termes tests de randomisation et les tests de permutations (Edgington & Onghena, 2020). Les tests de permutations sont habituellement utilisés pour des bases de données très petites, où il est possible d'effectuer le calcul avec l'ensemble des permutations possibles (Lopez-Castro et *al.*, 2019).[2] S'il est simple d'effectuer l'ensemble des permutations pour un petit nombre d'observations il est nettement plus difficile (et long) de le faire avec une base de données volumineuse. Il est alors plus facile d'effectuer des permutations sur un échantillon des possibilités. La méthode est alors qualifiée de test de randomisation (Good, 2005; Manly 2007).

Le test de randomisation ne nécessite pas que les individus de l'échantillon soient représentatifs de la population (Nuzzo, 2017). Des études ont montré que les tests de randomisation sont aussi puissants, sinon plus, que les autres tests non-paramétriques et paramétriques (Ludbrook & Dudley, 1998; Nuzzo, 2017). Selon Good (1994), les tests issus de ces approches sont les plus puissants lorsque les tests paramétriques ne sont pas valides. Il est possible d'arriver à des conclusions statistiques pour un échantillon donné sans que les paramètres soient associés à une population donnée (Pitman, 1937).

L'hypothèse nulle du test de randomisation est qu'en l'absence d'effet particulier, la variable d'intérêt peut être interchangée de manière aléatoire sans affecter les résultats obtenus. En d'autres termes, l'effet de traitement (d'une variable) est nul. Dans ce cas, le mouvement des observations au sein d'un vecteur donné ne devrait pas changer le résultat obtenu.

Deux approches sont habituellement retenues dans les applications empiriques selon l'hypothèse nulle à tester. Si l'hypothèse nulle consiste à vérifier s'il existe une relation significative entre une variable dépendante et un ensemble de variables indépendantes ($H_0: \beta_1 = \beta_2 = \ldots = \beta_K = 0$), alors les valeurs de la variable dépendante sont interchangées de manière aléatoire. De cette manière, la corrélation entre les variables indépendantes demeure intacte. Le but est alors de vérifier si les valeurs obtenues des coefficients via le vrai ensemble de données sont différentes de ce que l'on peut obtenir par le remaniement des valeurs dans le vecteur de la variable dépendante.

Si l'hypothèse nulle à tester est l'absence d'effet d'une variable donnée ($H_0: \beta_k = 0$), alors la variable indépendante d'intérêt, x_{ik}, est permutée de manière aléatoire. Sous l'hypothèse d'absence d'effet statistique, la valeur obtenue du paramètre ne devrait pas être sensiblement différente d'autres valeurs que nous pouvons obtenir en remaniant les valeurs contenues dans le vecteur de la variable indépendante. En d'autres mots, la valeur estimée avec l'ensemble des données originales devrait se fondre dans la distribution de valeurs obtenues par le biais de jeux de données où les valeurs sont allouées de manière aléatoire.

La pratique consiste habituellement à effectuer la permutation des valeurs pour la variable retenue (dépendante ou indépendante) un certain nombre de fois (habituellement 999 ou 9 999 fois) afin d'obtenir une distribution des possibles valeurs du coefficient d'intérêt. Cette distribution peut ensuite être comparée à la valeur obtenue avec le jeu de données original. Un écart important du paramètre avec les vraies données suggère le rejet de l'hypothèse nulle au détriment de l'hypothèse alternative (la permutation des valeurs n'apporte pas un effet similaire). À l'inverse, l'absence d'écart suggère que l'on ne peut rejeter l'hypothèse nulle (absence d'effet).

Les tests de randomisation sont largement mobilisés en analyse spatiale. Ils ont été notamment introduits par Anselin (1995) afin de tester la significativité des indices locaux d'association spatiale (ou LISA), mais également les tests d'autocorrélation globale. Ils sont largement mobilisés dans la littérature portant sur les distributions de points dans l'espace (*point pattern analysis* - Arbia et *al.*, 2020), mais également dans la littérature portant sur les régressions par procédure d'ajustement quadratique (quadratic assignment procedure - QAP) (Mantel, 1967; Hubert, 1986; Krackhardt, 1988; Dekker et *al.*, 2007)

Nombreuses sont les applications ayant utilisé cette approche pour produire des intervalles de confiance (Lopez-Castro et *al.*, 2019; Marcon & Peuch, 2010; Duranton & Overman, 2005). D'autres applications reposent sur l'analyse de l'activité criminelle (Grunt & Densley, 2012), l'analyse des transactions de fusion et acquisition d'entreprises (Cardenas-Morales & Dubé, 2019; 2023) et la localisation des nouvelles constructions (Champagne et *al.* 2022)

Le but de cet article est de présenter l'approche des tests de randomisation de manière intuitive et de formaliser l'application à partir du logiciel `Stata`. De cette manière, le lecteur pourra facilement mobiliser cette approche pour différentes applications.

1 Pour simple rappel, la statistique t dans une régression prend la forme suivante : $t = \frac{\hat{\beta}}{SE(\hat{\beta})}$, où $\hat{\beta}$ est le paramètre estimé et $SE(\hat{\beta})$ est l'écart-type du paramètre estimé.

2 Le nombre de permutations total correspond à la factorielle du nombre d'observations contenues dans la base de données.

La suite de l'article est divisée en trois sections. La première section introduit la méthodologie retenue afin de présenter l'intuition des tests de randomisation. Cette approche s'inspire des expériences de types de Monte Carlo, permettant de créer le processus générateur des données (PGD) sur la base de paramètres connus, afin de comparer les estimations issues du vrai jeu de données et des jeux de données permutés. La deuxième section propose une application issue d'un ensemble de données simulé afin de présenter l'intuition et la formalisation du test de permutations. Une première sous-section est dédiée à la présentation intuitive de l'approche, c'est-à-dire l'estimation du paramètre d'intérêt dans le cas d'une seule simulation. Par la suite, la seconde sous-section formalise le test de permutations pour un nombre de répétitions fixé. La dernière section propose une courte conclusion.

CONSTRUCTION D'UN ENSEMBLE DE DONNÉES

L'exercice proposé est issu d'un ensemble de données fictif. Cet exercice permet de fixer la valeur des paramètres ainsi que le processus générateur des données (PGD) et de connaître les vraies valeurs que l'on cherche à recouvrer. Après tout, le processus d'estimation vise à estimer la valeur de paramètres dont on ignore la vraie valeur. Sur la base de certaines hypothèses, on espère que les paramètres estimés permettent de s'approcher de la vraie valeur du paramètre qui demeure, en théorie et en pratique, inconnue.

Pour fin de l'exercice, le PGD s'inspire d'un modèle de type différence-en-différences (DID). La variable dépendante représente une mesure d'un résultat d'intérêt (*outcome*), notée y_{it}. Cette variable est supposée liée à une seule variable indépendante, x_{it}, ainsi qu'à deux effets fixes distincts : i) un effet propre au groupe visé (traitement ou contrôle - D_{it}); et ii) un effet propre au moment où les données sont observées (avant ou après un certain changement - T_{it}). Dans ce type d'exercice, D_{it} est égal à 1 si l'observation donnée fait partie du groupe de traitement et 0 autrement, alors que T_{it} est égal à 1 si l'observation est récoltée après le changement exogène à l'étude et 0 autrement.

L'effet de traitement moyen (*average treatment effect – ATE*) est habituellement isolé par le croisement des deux effets fixes ($D_{it} \times T_{it}$)[3], c'est-à-dire l'effet propre au groupe de traitement une fois que celui-ci subit le traitement, alors que ε_{it} représente un terme d'erreur, habituellement supposé de moyenne nulle et de variance homogène. Les termes de ce vecteur sont également supposés idépendants et identiquement distribués (équation 1).

$$y_{it} = \alpha + x_{it}\beta + D_{it}\theta + T_{it}\gamma + (D_{it} \times T_{it})\delta + \varepsilon_{it} \qquad (1)$$

Dans le PGD, l'ATE est mesuré par le paramètre δ. Dans ce modèle, un test d'hypothèse pour la significativité du paramètre δ permet habituellement de juger de la présence de l'effet de traitement moyen ($H_0 : \delta = 0$ contre $H_1 : \delta \neq 0$). Les autres paramètres ne sont pas spécifiquement d'intérêt. Ils permettent essentiellement de contrôler pour les caractéristiques observables, β, l'appartenance aux groupes (traitement ou contrôle), θ, et le moment où les observations sont collectées (avant ou après), γ.

Pour fin de simulations, nous supposons que la variable indépendante est distribuée de manière normale avec une moyenne nulle et une variance unitaire. L'appartenance aux groupes et le moment de récolte des données sont tous les deux issus de distributions uniformes. Le groupe de traitement et de contrôle représentent tous deux environ la moitié des observations, alors que l'on suppose que le quart des observations est récolté après le changement. Le terme d'erreur est également supposé distribué normalement avec une moyenne nulle et une variance unitaire. Les valeurs des paramètres sont fixées de sorte que $\beta = 1$, $\theta = 0{,}5$, $\gamma = 0{,}5$ et $\delta = 0{,}5$. La taille de l'échantillon est fixée à un total de 1000 observations.

EXERCICES EMPIRIQUES

La première étape nécessaire à la suite de l'exercice est de construire une base de données pour laquelle on connaît exactement le PGD.[4] Pour le bien de l'exercice, les étapes préliminaires consistent à préparer le logiciel afin de créer la base de données de départ. Le code est préparé sous le logiciel `Stata`.[5] Avant de faire quelques opérations que ce soit, il est nécessaire de libérer l'espace mémoire (ligne 1), d'éviter le bris de commandes (ligne 2) et d'assurer une réplicabilité des variables (ligne 3).

```
1. clear all
2. set more off
3. set seed 790118
```

On peut ensuite identifier le répertoire dans lequel on souhaite enregistrer l'information (ligne 4) afin de récupérer les bases de données créées et ensuite identifier le répertoire de travail de manière formelle (ligne 5).

```
4. global repertoire C:\Documents\Temp
5. cd "$repertoire"
```

À noter que les deux étapes précédentes ne sont pas strictement nécessaires : le logiciel enregistre par défaut l'information dans un répertoire existant. Cette procédure vise uniquement à éviter de surcharger le répertoire par défaut et, surtout, de se repérer rapidement dans les nombreux répertoires existants.

Par la suite, il est important de fixer les conditions générales de l'exercice. Par le biais de macro (`global`) permettant de conserver en mémoire les informations tout au long de l'exercice, on peut : i) spécifier la taille de l'échantillon (ligne 6); ii) spécifier le nombre de simulations que l'on souhaite effectuer pour les tests de randomisations (ligne 7); iii) fixer la proportion d'observations qui fait partie du groupe traitement (ligne 8); et iv) fixer la proportion d'observations qui sont récoltées avant le changement (ligne 9).

```
6. global nobs 1000
7. global simul = 999
8. global dlimit = 0.5
9. global tlimit = 0.75
```

On doit ensuite fixer les valeurs des différents paramètres mobilisés dans la construction du PGD (voir équation 1).

```
10. global beta = 1
11. global theta = 0.5
12. global gamma = 0.5
13. global delta = 0.5
```

Une fois le cadre général fixé, il suffit de préparer le logiciel à recevoir le nombre d'observations qui permettront de calculer le PGD (ligne 14). À noter que le rappel des informations enregistrées dans les macros `global` se fait avec l'aide du signe de dollar (`$`) sous `Stata`.

```
14. set obs $nobs
```

3 En termes matriciels, la multiplication des deux vecteurs terme-à-terme est formalisée par l'opérateur du produit de Hadamard, $\odot$: $D_{it} \odot T_{it}$.

4 Évidemment, l'exercice peut être mené à partir d'une base de données réelle : un ensemble de données que l'on observe dans un cas spécifique.

5 Le code peut facilement être transposé en R à partir de l'application ChatGPT, qui fournit une traduction efficace des codes.

On peut ensuite procéder à la création de variables qui servent à reconstruire le PGD. Après avoir créé un identifiant pour chacune des observations (ligne 15), on génère la variable indépendante, qui peut être vue ici comme le résultat d'une composante principale unique centrée et réduite qui synthétise un ensemble de variables (ligne 16). La création des deux autres variables indépendantes (traitement/contrôle et avant/après) repose sur une loi uniforme (lignes 17 et 18).

```
15. quietly generate id = _n
16. generate x = rnormal(0,1)
17. generate traitement = uniform()>$dlimit
18. generate apres = uniform()>$tlimit
```

Une fois les variables indépendantes créées, on peut reconstruire la variable dépendante en formalisant le PGD auquel on ajoute un terme d'erreur de loi normale de moyenne nulle et de variance unitaire (lignes 19 à 22).[6]

```
19. #delimit;
20. generate y=x*$beta+traitement*$theta+apres*$gamma+
21. (traitement*apres)*$delta + rnormal(0,1);
22. #delimit cr
```

On peut ensuite enregistrer la base de données résultantes (ligne 23) et estimer la régression (ligne 24) afin de vérifier que l'on retrouve bel et bien les valeurs des paramètres fixés au départ (voir lignes 10 à 13).

```
23. save "DonneesOriginales.dta", replace
24. regress y x traitement##apres
```

À noter que l'utilisation des sigles ## permet d'introduire directement le croisement des variables. Les valeurs estimées et les intervalles de confiance suggèrent que la base de données permet de retracer les valeurs réelles des paramètres (Figure 1), alors que le paramètre de l'ordonnée à l'origine n'est pas statistiquement significatif.

Figure 1. Résultats d'estimation des paramètres avec les données simulées

Source	SS	df	MS		
				Number of obs	= 1,000
				F(4, 995)	= 357.15
Model	1317.36997	4	329.342494	Prob > F	= 0.0000
Residual	917.532392	995	.922143108	R-squared	= 0.5895
				Adj R-squared	= 0.5878
Total	2234.90237	999	2.23713951	Root MSE	= .96028

y	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
x	1.025354	.0299325	34.26	0.000	.9666162	1.084092
1.traitement	.514921	.0708923	7.26	0.000	.3758054	.6540366
1.apres	.4645178	.0986403	4.71	0.000	.2709509	.6580846
traitement#apres 1 1	.5884167	.1375331	4.28	0.000	.3185286	.8583048
_cons	-.0082007	.0502007	-0.16	0.870	-.1067121	.0903107

Une fois les données simulées, il est possible de passer à l'application d'intérêt : le test de randomisation.

Une première approche

Une première approche consiste à permuter la valeur de la variable dépendante afin de tester la significativité globale du modèle. Pour ce faire, il faut modifier de manière aléatoire l'ordre des valeurs observées dans le vecteur original sans modifier les corrélations entre les variables indépendantes. L'opérationnalisation se base sur la création d'une variable aléatoire (ligne 25) afin d'ordonner les observations (ligne 26).[7] On crée ensuite une variable dépendante temporaire (avec valeurs permutées) qui alloue les valeurs en fonction de l'ordre établi par la nouvelle variable et l'ordre original des observations (donnée par l'identifiant unique – ligne 27).

```
25. generate p = uniform()
26. sort p
27. generate yp = y[id]
```

On peut ensuite estimer le modèle de régression avec la variable dépendante permutée et les variables indépendantes originales (ligne 28).

```
28. regress yp x traitement##apres
```

Les résultats de cette régression montrent que les paramètres estimés sont différents de ceux fixés au départ en plus d'être non-statistiquement significatifs (Figure 2). La permutation aléatoire de la variable dépendante génère des coefficients qui sont différents des vraies valeurs. La répétition de cet exercice permet de construire une distribution des valeurs estimées et ainsi de comparer les valeurs recouvrées avec le jeu de données original aux valeurs obtenues par permutations des valeurs du vecteur de la variable dépendante.

Figure 2. Résultats d'estimation pour les paramètres estimés avec une permutation

Source	SS	df	MS		
				Number of obs	= 1,000
				F(4, 995)	= 0.82
Model	7.37481412	4	1.84370353	Prob > F	= 0.5102
Residual	2227.52755	995	2.23872116	R-squared	= 0.0033
				Adj R-squared	= -0.0007
Total	2234.90237	999	2.23713951	Root MSE	= 1.4962

yp	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
x	-.0227102	.0466385	-0.49	0.626	-.1142313	.0688108
1.traitement	-.1280054	.1104587	-1.16	0.247	-.3447642	.0887534
1.apres	-.2022979	.1536934	-1.32	0.188	-.5038983	.0993025
traitement#apres 1 1	.1211746	.214293	0.57	0.572	-.2993434	.5416927
_cons	.5610802	.0782187	7.17	0.000	.4075876	.7145728

Cette première étape n'est pas la fin de l'exercice, mais seulement un exemple qui doit être répété un certain nombre de fois.

Une seconde approche

Une autre approche consiste à effectuer une permutation pour une variable d'intérêt, soit la variable identifiant le groupe de traitement (par rapport au groupe de contrôle - ligne 27), afin de tester la significativité d'un paramètre précis. Cette modification conserve de manière intacte les corrélations existantes entre la variable dépendante et les autres variables indépendantes.

```
27. generate traitementp = traitement[id]
```

Il suffit ensuite d'estimer le modèle de régression avec la variable dépendante originale, les variables indépendantes originales et la variable d'intérêt permutée, incluant la variable d'interaction (ligne 28).

```
28. regress y x traitementp##apres
```

6 À noter que l'option `#delimit` permet de faire en sorte que la fin de la ligne n'indique plus la fin de la commande. Ici, la fin de la commande est donnée par la présence du sigle ;. L'option `#delimit cr` permet de ramener la fin d'une commande à la fin d'une ligne. Pour être opérationnelle, la commande doit exécuter l'ensemble des lignes de manière simultanée. La commande peut également s'écrire sur une seule ligne sans recourir aux options `#delimit`.

7 À noter qu'il est possible que des valeurs se répètent lorsque la base de données est volumineuse. Dans ce cas, il suffit de créer deux variables aléatoires (p1 et p2) à partir de la même commande :
`generate p1 = uniform()`
`generate p2 = uniform()`
L'ordonnancement des observations se fait alors à l'aide des deux variables créées : `sort p1 p2`.

Les résultats de cette régression montrent que les paramètres liés aux variables originales conservent des valeurs proches de celles fixées au départ, mais que les coefficients liés à la variable indépendante permutée ainsi que le produit croisé sont différents des valeurs fixées au départ, en plus ne de pas être statistiquement significatifs (Figure 3).

Figure 3. Résultats d'estimation pour les paramètres estimés avec une permutation

Source	SS	df	MS	Number of obs	=	1,000
				F(4, 995)	=	284.53
Model	1192.42249	4	298.105623	Prob > F	=	0.0000
Residual	1042.47987	995	1.04771847	R-squared	=	0.5335
				Adj R-squared	=	0.5317
Total	2234.90237	999	2.23713951	Root MSE	=	1.0236

y	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
x	1.024048	.0319325	32.07	0.000	.9613854	1.086711
1.traitementp	-.1140356	.0756296	-1.51	0.132	-.2624474	.0343762
1.apres	.8037171	.1017593	7.90	0.000	.6040296	1.003405
traitementp#apres						
1 1	-.0669458	.1468571	-0.46	0.649	-.3551309	.2212393
_cons	.3093297	.0545647	5.67	0.000	.2022547	.4164048

Encore une fois, cet exercice n'est que la première étape d'une série de répétitions basée sur la permutation de la variable d'intérêt.

Une répétition des régressions avec valeurs permutées

Pour être intéressant et utile, le modèle de régression sur les valeurs permutées des variables (dépendantes ou indépendantes) doit être réestimé un certain nombre de fois (habituellement 999 ou 9 999) afin d'obtenir une distribution de la valeur du paramètre d'intérêt. C'est l'ensemble de ces valeurs que l'on doit comparer à la valeur originale obtenue afin de tirer une conclusion quant à l'impact possible de la variable d'intérêt sur la relation globale.

De manière plus spécifique, la significativité se calcule à partir du rang de la valeur originale du paramètre par rapport aux valeurs des paramètres obtenus par simulation. Lorsque la valeur observée ($\delta_{original}$) se retrouve dans les 5 percentiles extrêmes ($\delta_{original}$ < 5^{e} percentile ou $\delta_{original}$ > 95^{e} percentile), on peut alors affirmer que le paramètre est statistiquement significatif à un seuil de 95 % ou, plus spécifiquement, que l'effet de traitement n'est pas nul. Cette approche peut aussi être généralisée à l'ensemble des paramètres liés aux variables indépendantes si le but est de tester la significativité globale ($\beta_{original}$, $\theta_{original}$, $\gamma_{original}$, et $\delta_{original}$).

Manly (2007) explique qu'en fonction du nombre de permutations, il est possible de construire des tests de significativité statistique avec un intervalle de confiance de 99 % ou 95 %. Ainsi, un modèle de régression dont les variables sont permutées 4 999 fois ou plus permet de calculer un test de signification de 1 %, tandis que le même modèle de régression basé sur 999 permutations permet d'avoir un test de signification de 5 %.

Afin de généraliser l'application précédente à un nombre de répétitions donné, il faut préparer un programme permettant d'enregistrer les différentes valeurs obtenues pour le nombre de permutations souhaité (lignes 29 et 30).[8] Il est ensuite nécessaire de spécifier l'information que l'on souhaite conserver et créer une base de données dans laquelle l'information sera enregistrée (ligne 31). Pour l'exercice, nous portons notre attention sur la significativité de l'effet de traitement en conservant uniquement le numéro de la simulation (`nsimul`) et la valeur du paramètre estimée pour la variable croisée (`delta`).

```
29. program Permutation, rclass
30. tempname parametre
31. postfile `parametre' nsimul delta using permuta-
tion, replace
```

La première information conservée doit se rapporter à la régression originale (ligne 32). Après avoir estimé le modèle sur les données originales, on conserve les informations souhaitées (lignes 33 et 34). Puisque la valeur est celle issue du vrai jeu de données (ligne 33), le numéro de la simulation est fixé à 0 (ligne 34). Les valeurs sont ensuite enregistrées dans la base de données en construction (ligne 35).

```
32. regress y x traitement##apres
33. scalar delta = _b[1.traitementp##1.apres]
34. scalar nsimul = 0
35. post `parametre' (nsimul) (delta)
```

Par la suite, on crée une boucle qui permet de calculer les valeurs des paramètres pour le nombre de simulations fixé au départ (voir ligne 7). Cette boucle permet de répéter l'ensemble des opérations le nombre de fois voulu (ligne 36) et prend appui sur les opérations couvertes précédemment (voir lignes 25, 26 et 27).[9]

Il s'agit de créer une variable aléatoire (ligne 37) sur laquelle la permutation (ligne 38) prend forme pour créer la variable indépendante permutée (lignes 39). Une fois cette opération effectuée, il suffit d'estimer le modèle de régression (ligne 40) et de conserver : i) la valeur du paramètre pour la variable croisée (ligne 41); et ii) le numéro de la simulation (ligne 42). Ces deux informations sont ensuite enregistrées dans la base de données en construction (ligne 43) avant de supprimer les variables temporaires (ligne 44) afin de répéter l'opération un nombre de fois fixé (ligne 36). La boucle des opérations prend fin lorsque le nombre de calculs atteint la borne supérieure fixée (`$simul`).

```
36. forvalues s = 1/$simul {
37. generate p = uniform()
38. sort p
39. generate traitementp = traitement[id]
40. regress y x traitementp##apres
41. scalar delta = _b[1.traitementp#1.apres]
42. scalar nsimul = `s'
43. post `parametre' (nsimul) (delta)
44. drop p traitementp
45. }
```

Une fois la boucle effectuée et les paramètres enregistrés dans la base de données, on peut fermer le programme créé (lignes 46 et 47) et l'exécuter formellement (ligne 48).

```
46. postclose `parametre'
47. end
48. Permutation
```

Une fois l'exécution terminée, la base de données construite (`permutation.dta`) peut être mobilisée afin de calculer la pseudo-significativité et transposer en image cette approche en exploitant la distribution des valeurs obtenues par la répétition des estimations. Pour ce faire, il est d'abord nécessaire de charger la base de données créée (ligne 49). On peut ensuite obtenir certaines informations telles que : i) la valeur du paramètre obtenue avec le vrai jeu de données (lignes 50 et 51); et ii) les valeurs minimales et maximales observées (lignes 52 à 54). Avec ces valeurs, on peut obtenir une idée du rang de la valeur réelle en comparant à la distribution des valeurs issues des permutations (ligne 55). Visuellement, on peut également construire l'histogramme de distribution des valeurs (lignes 56 à 58).

8 À noter que les variables permutées ne doivent pas déjà exister. Ainsi, le lecteur qui a appliqué l'exercice précédent devra éliminer les variables créées avant de lancer le programme (voir ligne 44).

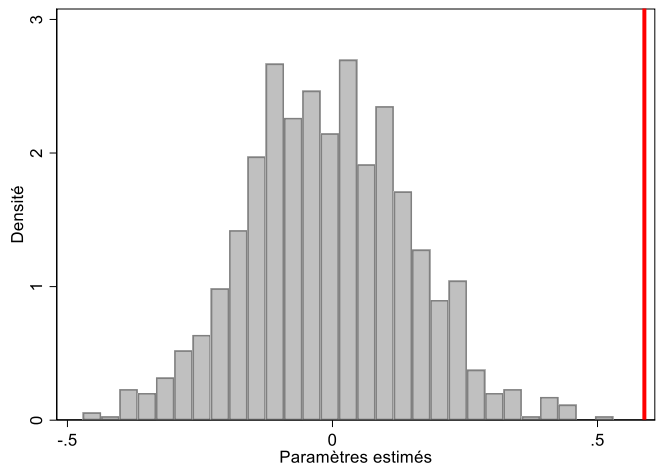
9 À noter que les variables `p` et `traitementp` doivent préalablement être supprimées pour être créées de nouveau.

```
49. use permutation, clear
50. quietly summarize delta if nsimul==0
51. global parestime = r(mean)
52. quietly summarize delta
53. global xmin = r(min)
54. global xmax = r(max)
55. summarize delta if nsimul!=0, detail
56. #delimit;
57. histogram delta if nsimul!=0, fcolor(gs12) lcolor(gs8)
      ytitle(Densité) xtitle(Paramètres estimés)
      xscale(range($xmin $xmax))
      xline($parestime, lwidth(thick) lpattern(solid)
lcolor(red))
      legend(off) scheme(s1color);
58. #delimit cr
```

Les statistiques descriptives permettent de constater que la valeur obtenue est significative puisqu'aucun paramètre estimé du test de randomisation ne permet d'obtenir une valeur supérieure à celle obtenue. On dit alors que le paramètre est statistiquement significatif à un seuil de 95 %. On peut également parler de pseudo-significativité telle que p = 0,001, bien que le nombre de simulations soit faible pour tenir une telle conclusion.

On peut également visualiser le résultat en comparant la valeur obtenue sur le jeu réel de données (la ligne rouge) à la distribution des autres valeurs obtenues par permutations (Figure 4). Dans les deux cas, la conclusion est identique.

Figure 4. Distribution des valeurs des paramètres estimés



CONCLUSION

L'article a pour but de présenter une nouvelle approche permettant d'évaluer la significativité des coefficients à partir d'une approche non-paramétrique. L'approche présentée est issue d'une permutation des valeurs originales de la variable dépendante ou de l'une des variables indépendantes (celle d'intérêt) pour l'ensemble des observations. La permutation des valeurs observées au sein d'un vecteur spécifique pour un nombre donné de réalisations permet d'obtenir une distribution des valeurs des paramètres avec laquelle on peut comparer la valeur du paramètre issus du jeu de données original. Le test de randomisation permet de juger de la pseudo-significativité de manière plus flexible que l'approche paramétrique classique puisqu'elle ne repose sur aucune hypothèse particulière de la statistique de significativité paramétrique (test-t).

La présentation du test de randomisation est formalisée à partir d'un processus générateur des données (PGD), où les valeurs des paramètres sont fixées au départ. L'inspiration des approches de type de Monte Carlo permet de mettre en évidence les différentes étapes permettant d'implémenter ces tests. Après avoir créé une base de données sur une relation imposée, les données sont mobilisées afin de développer les deux approches. Le tout est formalisé à partir du code dans le logiciel `Stata`. L'approche de type pas-à-pas permet au lecteur de mieux comprendre les différentes étapes afin que le code soit facilement applicable sur différents jeux de données.

Le test de randomisation est déjà largement mobilisé en analyse spatiale puisque la forme de la distribution, en présence d'autocorrélation spatiale, ne respecte pas les hypothèses fixées dans le cas de l'hypothèse nulle lorsque vient le temps de dériver la forme du test statistique. Depuis les travaux de Anselin (1995) sur les indicateurs locaux d'association spatiale (LISA), cette approche s'est généralisée, notamment dans le domaine de l'analyse des données spatiales individuelles (*point pattern analysis)* et dans la détection des regroupements spatiaux *(clusters).* Elle permet d'éviter certains problèmes liés au calcul.

En bref, le code présenté permet d'implémenter les tests de randomisation sur les problèmes empiriques rencontrés dans la pratique. Il peut facilement être mobilisé sur différentes applications empiriques.

RÉFÉRENCES

Anselin, L. (1995). Local Indicators of Spatial Association – LISA, *Geographical Analysis*, 27(2): 93-115.

Arbia, G., Espa, G. & Giuliani, D. (2020). *Spatial Microeconometrics*, Routledge.

Cardenas Morales, D. A., & Dube, J. (2019). Schemas origine-destination des activites de fusion et d'acquisition (F&A) au Canada : Une analyse sectorielle des reseaux, 1994–2016. *Canadian Journal of Regional Science/Revue canadienne des sciences régionales*, 42(1), 46–59.

Cardenas Morales. D. A. & Dubé, J. (2023) The evolution and trajectories of the geography of mergers and acquisitions: A city network analysis for Canada, 1994–2016, *Journal of Urban Affairs*, 45:7, 1358-1378, DOI: 10.1080/07352166.2021.1915150

Champagne, M.-P., Dubé, J. & Barla, P. (2022). Build it and they will Come: How Does a New Public Transit Station Influence Building Construction?, *Journal of Transport Geography*, 100: 103320.

Dekker, D., Krackhardt, D., & Snijders, T. A. B. (2007). Sensitivity of MRQAP tests to collinearity and autocorrelation conditions. *Psychometrika*, 72(4), 563–581.

Duranton, G. & Overman, H.G. (2005). Testing for Localization using Micro-Geographic Data, *The Review of Economic Studies*, 72(4): 1077-1106.

Edgington, E.S. & Onghena, P. (2020). *Randomization Tests*, Fourth Edition, CRC Press.

Fischer, R.A. (1935). *The Design of Experiments*, Oliver & Boyd, Edinburgh.

Good, P. (1994). *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, Springer, New-York.

Good, P. (2005). Permutation, parametric and bootstrap tests of hypotheses. 3rd ed. New York, NY: Springer-Verlag New York.

Grund, T., & Densley, J. A. (2012). Ethnic heterogeneity in the activity and structure of a black street gang. European Journal of Criminology, 9(4), 388–406. https://doi.org/10.1177/1477370812447738

Hubert, L. (1986). Assignment methods in combinational data analysis (Vol. 73). CRC Press.

Krackhardt, D. (1988). Predicting with networks: Nonparametric multiple regression analysis of dyadic data. Social Networks, 10(4), 359–381. https://doi.org/10.1016/0378-8733(88)90004-4

López-Castro, M. A., Thériault, M. & Vandersmissen, M.-H. (2019). A method to test the significance of differences between centrographic measures of dispersion. The Canadian Geographer / Le Géographe canadien, 63, 326-339.

Ludbrook, J. & Dudley, H. (1998). Why Permutation Tests are Superior to the t and F tests in Biomedical Research, *The American Statistician*, 52: 127-132.

Manly, B. F. J. (2007). Randomization, bootstrap and Monte Carlo methods in biology. 3rd ed. Boca Raton, FL: Taylor & Francis Group, LLC.

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. Cancer research, 27(2_Part_1), 209-220.

Marcon, E. & Peuch, F. (2010). Measures of the Geographic Concentration of Industries: Improving Distance-based Methods, *Journal of Economic Geography*, 10(5): 745-762.

Nuzzo, R.L. (2017). Randomization Test: An Alternative Analysis for the Difference of Two Means, *PMR Journal*, 9: 306-310.

Onghena, P. (2018) Chapter 14: Randomization Tests or Permutation Tests? A Historical and Terminological Clarification. In Editor Vance Berger. *Randomization, masking, and allocation concealment* (209-227). Chapman & Hall/CRC Press.

Pitman, E.J.G. (1937). Significance Test Which may be Applied to Samples from any Population, *Journal of Royal Statistical Society*, Series B, 4: 119-130.

Welch, B.L. (1937). On the z-test in Randomized Block and Latin Squares, *Biometrika*, 29: 21-52.

ANNEXE: CODE COMPLET

```
clear all
set more off
set seed 790118 /*Assurer une replicabilite des resultats*/

/*Fixer le nombre d'observations*/
global nobs 1000
set obs $nobs
/*Fixer les valeurs critiques*/
global dlimit = 0.5
global tlimit = 0.75
/*Identifier le nombre de repetitions*/
global simul = 999

/*Simuler les variables (indépendantes) de depart*/
quietly generate id = _n
generate x = rnormal(0,1)
generate traitement = uniform()>$dlimit
generate apres = uniform()>$tlimit

/*Fixer les parametres de la regression*/
global beta = 1
global theta = 0.5
global gamma = 0.5
global delta = 0.5
/*Reconstruire le DGP*/
#delimit;
generate y = x*$beta + traitement*$theta + apres*$gamma +
       (traitement*apres)*$delta + rnormal(0,1);
#delimit cr
/*Enregistrer les donnees de depart*/
save "DataOriginal.dta", replace

/*Debut des tests de permutation*/
/*Preparer le programme de simulation*/
program Permutation, rclass   /*Nom du programme*/
tempname parametre
/*Statistiques a conserver (results)*/
postfile `parametre' nsimul beta gamma theta delta using
permutation, replace

/*Estimer le modele de depart*/
regress y x traitement##apres
scalar beta = _b[x]
scalar gamma = _b[1.apres]
scalar theta = _b[1.traitement]
scalar delta = _b[1.traitement#1.apres]
/*Sauvegarder le numero de la simulation*/
scalar nsimul = 0
/*Sauvegarder les parametres d'interet*/
post `parametre' (nsimul) (beta) (gamma) (theta) (delta)

forvalues s = 1/$simul {
/*Identifier ou est rendu le programme*/
display "Numéro de simulations ->" `s'
quietly {
/*Tests de permutation*/
quietly generate p = uniform()
sort p
generate traitementp = traitement[id]
/*Calculer la regression avec valeurs permutees*/
regress y x traitementp##apres
scalar beta = _b[x]
scalar gamma = _b[1.apres]
scalar theta = _b[1.traitementp]
scalar delta = _b[1.traitementp#1.apres]
/*Sauvegarder le numero de la simulation*/
scalar nsimul = `s'
/*Sauvegarder les paramètres d'intérêt*/
post `parametre' (nsimul) (beta) (gamma) (theta) (delta)
drop p traitementp
}
}
postclose `parametre'
end
Permutation

/*Tester la pseudo-significativite*/
use permutation, clear
quietly summarize delta if nsimul==0
global parestime = r(mean)
summarize delta if delta>=$parestime
quietly summarize delta
global xmin = r(min)
global xmax = r(max)

#delimit;
histogram delta if nsimul!=0, fcolor(gs12) lcolor(gs8)
       ytitle(Densité) xtitle(Paramètres estimés)
       xscale(range($xmin $xmax))
       xline($parestime,  lwidth(thick)  lpattern(solid) lcolor(red))
       legend(off) scheme(s1color);
#delimit cr
```