

Introduction à plm

Yves Croissant

30 octobre 2006

1 Introduction

L'objectif du package `plm` est de permettre d'estimer simplement les modèles de panel. Le package `nlme` (*non-linear mixed effect models*) permet d'estimer certains modèles de panel, mais pas de manière intuitive pour un économètre. `plm` fournit des fonctions pour lire des données de panel, pour estimer différents modèles et pour réaliser des tests.

Ce package est chargé en utilisant :

```
> library(plm)
```

Ce document présente différents exemples d'utilisation de `plm` en utilisant des données disponibles dans le package `Ecdat`.

```
> library(Ecdat)
```

Ces données sont utilisées dans BALTAGI (2001).

2 Lecture des données

Avec `plm`, les données de panel sont stockées dans un objet de type `pdata.frame`, qui est un `data.frame` auquel plusieurs attributs ont été ajoutés de manière à stocker des informations sur la structure des données utiles pour les estimations. Un `pdata.frame` peut être créé à partir d'un `data.frame` ordinaire (commande `pdata.frame`) ou directement à partir d'un fichier texte (commande `pread.table`).

2.1 Lire les données à partir d'un data.frame

Nous illustrerons la commande `pdata.frame` en utilisant les données `Produc` :

```
> data(Produc)
> pdata.frame(Produc, "state", "year", "pprod")
```

La commande `pdata.frame` attend 4 arguments :

- le nom du `data.frame`,
- `id` : la variable qui identifie les individus,
- `time` : la variable qui identifie les périodes,

– `name` : le nom sous lequel sera stocké le `pdata.frame`

Les observations doivent être classées par individu, puis par période. Le troisième argument est optionnel, en son absence, une nouvelle série appelée `time` est ajoutée aux données. Le quatrième argument est également optionnel, en son absence, le `pdata.frame` est stocké sous le même nom que le `data.frame`. S'il n'y a pas d'index temporel, seul l'argument `id` peut être renseigné, dans ce cas une variable appelée `time` est ajoutée au `pdata.frame`.

```
> data(Hedonic)
> pdata.frame(Hedonic, "townid")
```

Enfin, il est possible, si le panel est cylindré, d'indiquer comme deuxième argument non pas le nom de l'index individuel, mais le nombre d'individus. Dans ce cas, deux séries supplémentaires `id` et `time` sont créées :

```
> data(Wages)
> pdata.frame(Wages, 595)
```

Un résumé des données est obtenu en appliquant la méthode `summary` :

```
> summary(Hedonic)
```

```
-----
----- Indexes -----
-----
Individual index : townid
Time index      : time
-----
----- Panel Dimensions -----
-----
Unbalanced Panel
Number of Individuals      : 92
Number of Time Observations : from 1 to 30
Total Number of Observations : 506
-----
----- Time/Individual Variation -----
-----
no time variation : zn indus rad tax ptratio
-----
----- Descriptive Statistics -----
-----
```

mv	crim	zn	indus	chas
Min. : 8.517	Min. : 0.00632	Min. : 0.00	Min. : 0.46	no :471
1st Qu.: 9.742	1st Qu.: 0.08205	1st Qu.: 0.00	1st Qu.: 5.19	yes: 35
Median : 9.962	Median : 0.25651	Median : 0.00	Median : 9.69	
Mean : 9.942	Mean : 3.61352	Mean : 11.36	Mean :11.14	
3rd Qu.:10.127	3rd Qu.: 3.67708	3rd Qu.: 12.50	3rd Qu.:18.10	
Max. :10.820	Max. :88.97620	Max. :100.00	Max. :27.74	

nox	rm	age	dis
Min. :14.82	Min. :12.68	Min. : 2.90	Min. :0.1219
1st Qu.:20.16	1st Qu.:34.64	1st Qu.: 45.02	1st Qu.:0.7420
Median :28.94	Median :38.55	Median : 77.50	Median :1.1655
Mean :32.11	Mean :39.99	Mean : 68.57	Mean :1.1880
3rd Qu.:38.94	3rd Qu.:43.87	3rd Qu.: 94.07	3rd Qu.:1.6464

Max. :75.86 Max. :77.09 Max. :100.00 Max. :2.4954

rad	tax	ptratio	blacks
Min. :0.000	Min. :187.0	Min. :12.60	Min. :0.00032
1st Qu.:1.386	1st Qu.:279.0	1st Qu.:17.40	1st Qu.:0.37538
Median :1.609	Median :330.0	Median :19.05	Median :0.39144
Mean :1.868	Mean :408.2	Mean :18.46	Mean :0.35667
3rd Qu.:3.178	3rd Qu.:666.0	3rd Qu.:20.20	3rd Qu.:0.39623
Max. :3.178	Max. :711.0	Max. :22.00	Max. :0.39690

lstat	townid	time
Min. :-4.0582	29 : 30	1 : 92
1st Qu.:-2.6659	84 : 23	2 : 75
Median :-2.1747	5 : 22	3 : 60
Mean :-2.2342	83 : 19	4 : 50
3rd Qu.:-1.7744	41 : 18	5 : 39
Max. :-0.9684	28 : 15	6 : 33
	(Other):379	(Other):157

L’affichage se décompose en quatre sections :

- **indexes** indique le nom des index individuels et temporels,
- **panel dimensions** donne des informations sur la dimension du panel,
- **Time/individual variation** indique les variables pour lesquelles il n’y a pas de variation temporelle ou individuelle,
- **Descriptive statistics** présente des statistiques descriptives sur les différentes variables.

2.2 Lire les données à partir d’un fichier texte

La commande `pread.table` est proposée afin de lire des données de panel directement à partir d’un fichier texte. La syntaxe est la suivante :

```
pread.table("c:/mes documents/essai/donnees.txt",
            "firm", "year", "dataname", header=T, sep=";", dec=",")
```

Les arguments de `pread.table` sont les suivants :

- le fichier texte qui contient les données,
- **id** : la variable qui identifie les individus,
- **time** : la variable qui identifie les périodes,
- **name** : le nom du `pdata.frame` qui sera créé (si l’argument est nul, le nom du `pdata.frame` est le nom du fichier sans l’extension et sans le chemin d’accès),
- d’autres arguments éventuels qui seront passés à la fonction `read.table` (ici, on précise que la première ligne du fichier contient le nom des variables, que le séparateur de champs est le point-virgule et que le séparateur décimal est la virgule).

3 Estimer un modèle

L’estimation d’un modèle de panel est obtenu avec la fonction `plm`.

3.1 Utilisation simple de plm

Il y a deux possibilités pour utiliser `plm` : la première consiste à estimer une liste de modèles (comportement par défaut), la seconde à estimer un modèle. Dans le premier cas, les modèles estimés sont :

- le modèle à effets fixes (`within`),
- les moindres carrés ordinaires (`pooling`),
- le modèle estimé sur les moyennes individuelles ou temporelles (`between`),
- le modèle à erreurs composées (`random`).

Il peut être pertinent d'estimer directement ces quatre modèles car beaucoup de tests impliquent plusieurs d'entre eux. L'utilisation la plus simple de `plm` consiste à indiquer une formule décrivant le modèle à estimer et le `pdata.frame` qui contient les données¹ :

```
> zz <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp,
+ data = pprod)
```

Le résultat de l'estimation est stocké dans un objet de classe `plms`. Celui-ci est une liste contenant les résultats des 4 modèles estimés. Chacun de ceux-ci est un objet de classe `plm` et peut ensuite être extrait :

```
> zz$within <- zz$within
```

On peut également préciser un modèle à estimer en fixant l'argument `model` de `plm` à : `within`, `between`, `random` ou `pooling`.

```
> zzra <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp,
+ data = pprod, model = "random")
```

Les objets de classe `plm` et `plms` sont dotés d'une méthode `print` qui ressemble à celle de la commande `lm` :

```
> print(zzra)
```

Model Formula: log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp

Coefficients:

(intercept)	log(pcap)	log(pc)	log(emp)	unemp
2.1354110	0.0044386	0.3105484	0.7296705	-0.0061725

Ils disposent également d'une méthode `summary` :

- pour les objets de classe `plms`, les coefficients et les écarts-type des modèles à effets fixes et à erreurs composées sont affichés, ainsi que plusieurs tests (test d'Hausman, test de multiplicateur de Lagrange, test de F).
- pour les objets de classe `plm`, le tableau des coefficients est affiché, ainsi que différentes statistiques.

```
> summary(zz)
```

¹L'exemple présenté ci-dessous est utilisé par BALTAGI (2001), pp. 25–28.

```

-----
----- Model Description -----
Oneway (individual) effect

Model Formula      : log(gsp) ~ log(pcap) + log(pc) + log(emp) +
                      unemp

-----
----- Panel Dimensions -----
Balanced Panel
Number of Individuals      : 48
Number of Time Obserbations : 17
Total Number of Observations : 816

-----
----- Coefficients -----
              within      wse      random      rse
(intercept)      .      .      2.13541100 0.1335
log(pcap) -0.02614965 0.02900158 0.00443859 0.0234
log(pc)      0.29200693 0.02511967 0.31054843 0.0198
log(emp)      0.76815947 0.03009174 0.72967053 0.0249
unemp        -0.00529774 0.00098873 -0.00617247 0.0009

-----
----- Tests -----
Hausman Test      : chi2(4) = 9.525416 (p.value=0.04922762)
F Test           : F(47,764) = 75.8204 (p.value=0)
Lagrange Multiplier Test : chi2(1) = 4134.961 (p.value=0)

-----
> summary(zzra)

```

```

-----
----- Model Description -----
Oneway (individual) effect
Random Effect Model (Swamy-Arora's transformation)
Model Formula      : log(gsp) ~ log(pcap) + log(pc) +
                      log(emp) + unemp

-----
----- Panel Dimensions -----
Balanced Panel
Number of Individuals      : 48
Number of Time Obserbations : 17
Total Number of Observations : 816

-----
----- Effects -----
              var      std.dev      share
idiosyncratic 0.0014544 0.0381371 0.1754
individual    0.0068377 0.0826905 0.8246
theta      : 0.88884

-----
----- Residuals -----
              Min.      1st Qu.      Median      Mean      3rd Qu.      Max.

```

```
-1.07e-01 -2.46e-02 -2.37e-03 -9.93e-19 2.17e-02 2.00e-01
```

```
-----
                        Coefficients
-----
              Estimate Std. Error z-value Pr(>|z|)
(intercept)  2.13541100  0.13346149 16.0002 < 2.2e-16 ***
log(pcap)    0.00443859  0.02341732  0.1895  0.8497
log(pc)      0.31054843  0.01980475 15.6805 < 2.2e-16 ***
log(emp)     0.72967053  0.02492022 29.2803 < 2.2e-16 ***
unemp       -0.00617247  0.00090728 -6.8033 1.023e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
-----
                        Overall Statistics
-----
Total Sum of Squares      : 29.209
Sum of Squares Residuals  : 1.1879
Rsq                       : 0.95933
F                         : 4782.77
P(F>0)                   : 8.76231e-08
```

Pour le modèle `random`, la commande `summary` donne des précisions sur la variance des différents éléments du terme d'erreur.

Enfin, les objets issus de la commande `plm` supportent l'utilisation de la commande `update`. Par exemple, pour enlever `log(emp)` et `unemp` et ajouter `emp` à l'estimation précédente, on utilisera la commande suivante :

```
> zzwithmod <- update(zzwith, . ~ . - unemp - log(emp) + emp)
> zzmod <- update(zz, . ~ . - unemp - log(emp) + emp)
> summary(zzwithmod)
```

```
-----
                        Model Description
-----
Oneway (individual) effect
```

```
Model Formula      : log(gsp) ~ log(pcap) + log(pc) + emp
```

```
-----
                        Panel Dimensions
-----
Balanced Panel
Number of Individuals      : 48
Number of Time Observations : 17
Total Number of Observations : 816
```

```
-----
                        Coefficients
-----
              within      wse      random      rse
(intercept)      .          . 7.1982e-01  0.1846
log(pcap)  1.7888e-01 4.0690e-02 3.4357e-01  0.0322
log(pc)    6.9975e-01 2.9154e-02 6.0369e-01  0.0256
emp        3.7909e-05 8.7824e-06 5.0924e-05 8.218e-06
```

```
-----
                        Tests
-----
Hausman Test      : chi2(3) = 80.35868 (p.value=0)
```

```
F Test : F(47,765) = 101.9109 (p.value=0)
Lagrange Multiplier Test : chi2(1) = 4355.292 (p.value=0)
```

Les effets fixes peuvent être extraits facilement en utilisant la commande FE qui prend comme argument soit un modèle estimé de type `within`, soit un objet de classe `plms` :

```
> FE(zzmod)
```

ALABAMA	ARIZONA	ARKANSAS	CALIFORNIA	COLORADO
1.1717531	1.3062389	1.1877004	1.6191982	1.4582149
CONNECTICUT	DELAWARE	FLORIDA	GEORGIA	IDAHO
1.7060341	1.2035746	1.5564969	1.4460171	1.1002049
ILLINOIS	INDIANA	IOWA	KANSAS	KENTUCKY
1.5496106	1.3451714	1.2323038	1.1735476	1.3492604
LOUISIANA	MAINE	MARYLAND	MASSACHUSETTS	MICHIGAN
1.1652834	1.2659480	1.6011871	1.7384231	1.5290312
MINNESOTA	MISSISSIPPI	MISSOURI	MONTANA	NEBRASKA
1.3654287	1.1545345	1.4809262	0.7960951	1.0905033
NEVADA	NEW_HAMPSHIRE	NEW_JERSEY	NEW_MEXICO	NEW_YORK
1.0627992	1.4138235	1.7420589	1.0925399	1.6694387
NORTH_CAROLINA	NORTH_DAKOTA	OHIO	OKLAHOMA	OREGON
1.5048751	0.7663694	1.4985974	1.2784660	1.3345094
PENNSYLVANIA	RHODE_ISLAND	SOUTH_CAROLINA	SOUTH_DAKOTA	TENNESSE
1.4972243	1.5948140	1.2344011	0.8705826	1.3123010
TEXAS	UTAH	VERMONT	VIRGINIA	WASHINGTON
1.3230328	1.2464927	1.1804339	1.6175357	1.3492922
WEST_VIRGINIA	WISCONSIN	WYOMING		
1.0129871	1.4860561	0.7842841		

3.2 Options concernant le modèle à erreurs composées

Le modèle à erreurs composées est estimé en réalisant une estimation linéaire sur les données en quasi-différences. Le coefficient de cette quasi-différence est obtenu à partir d'estimations préalables des variances des différents éléments du terme d'erreur. Quatre possibilités existent pour estimer ce coefficient, en fixant l'argument `theta.method` à :

- `swar` : d'après SWAMY et ARORA (1972),
- `walhus` : d'après WALLACE et HUSSAIN (1969),
- `amemiya` : d'après AMEMIYA (1971),
- `nerlove` : d'après NERLOVE (1971).

Par défaut, l'estimateur `swar` est utilisé. Pour, par exemple, utiliser l'estimateur `amemiya`, on utilisera :

```
> zzra <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp,
+ data = pprod, model = "random", theta.method = "amemiya")
```

3.3 Préciser les effets

Par défaut, le modèle estimé par `plm` contient des effets individuels. Il est également possible, via l'argument `effect` d'introduire :

- des effets temporels `effect="time"`,
- des effets temporels et individuels `effect="twoways"`.

Par exemple, pour estimer un modèle à erreurs composées à effets temporels et individuels pour les données `Grunfeld`, on utilisera :

```
> data(Grunfeld)
> pdata.frame(Grunfeld, "firm", "year")
> z <- plm(inv ~ value + capital, data = Grunfeld, effect = "twoways",
+   theta.method = "amemiya")
> summary(z$random)
```

```
-----
----- Model Description -----
Twoways effects
Random Effect Model (Swamy-Arora's transformation)
Model Formula      : inv ~ value + capital
-----
----- Panel Dimensions -----
Balanced Panel
Number of Individuals      : 10
Number of Time Obserbations : 20
Total Number of Observations : 200
-----
----- Effects -----
              var  std.dev  share
idiosyncratic 2675.426   51.725 0.2738
individual    7095.252   84.233 0.7262
time          0.000     0.000 0.0000
theta : 0.86397 (id) 0 (time) 0 (total)
-----
----- Residuals -----
      Min.   1st Qu.   Median     Mean   3rd Qu.   Max.
-1.77e+02 -1.98e+01  4.60e+00  8.77e-16  1.95e+01  2.53e+02
-----
----- Coefficients -----
              Estimate Std. Error z-value Pr(>|z|)
(intercept) -57.865377  29.393359 -1.9687  0.04899 *
value        0.109790   0.010528 10.4285 < 2e-16 ***
capital      0.308190   0.017171 17.9483 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
-----
----- Overall Statistics -----
Total Sum of Squares      : 2376000
Sum of Squares Residuals  : 547910
Rsq                       : 0.7694
F                         : 328.647
P(F>0)                   : 0.0030381
-----
```


Dans la section “effects” du résultat est désormais précisé l’importance des trois éléments du terme d’erreur, ainsi que les trois paramètres utilisés dans la transformation (associés aux moyennes individuelle, temporelle et globale).

L’estimation du modèle à double effets (individuels et temporels) n’est pour l’instant disponible que pour les panels cylindrés.

3.4 Modèle de Hausman–Taylor

Le modèle de HAUSMAN–TAYLOR (1981) peut être estimé à l’aide de la commande `plm` en fixant l’argument `model` à “`ht`” et en précisant en second argument de `plm` une formule présentant les variables qui sont utilisées comme instruments ² :

```
> data(Wages)
> pdata.frame(Wages, 595)
> form = lwage ~ wks + south + smsa + married + exp + I(exp^2) +
+       bluecol + ind + union + sex + black + ed
> ht = plm(form, ~sex + black + bluecol + south + smsa + ind, data = Wages,
+       model = "ht")
> summary(ht)
```

```
-----
----- Model Description -----
Oneway (individual) effect
Hausman-Taylor Model
Model Formula           : lwage ~ wks + south + smsa + married +
                        exp + I(exp^2) + bluecol + ind +
                        union + sex + black + ed
Instrumental Variables   : ~sex + black + bluecol + south +
                        smsa + ind
Time--Varying Variables
  exogenous variables   : bluecolyes,southyes,smsayes,ind
  endogenous variables  : wks,marriedyes,exp,I(exp^2),unionyes
Time--Invariant Variables
  exogenous variables   : sexmale,blackyes
  endogenous variables  : ed
-----
----- Panel Dimensions -----
Balanced Panel
Number of Individuals    : 595
Number of Time Obserbations : 7
Total Number of Observations : 4165
-----
----- Effects -----
                        var std.dev share
idiosyncratic 0.023044 0.151803 0.0253
individual    0.886993 0.941803 0.9747
theta       : 0.93919
-----
```

²L’exemple présenté ci-dessous est utilisé par BALTAGI (2001), p. 130.

```

----- Residuals -----
      Min.    1st Qu.    Median      Mean     3rd Qu.      Max.
-1.92e+00 -7.07e-02  6.57e-03 -2.46e-17  7.97e-02  2.03e+00

-----
----- Coefficients -----
      Estimate Std. Error z-value Pr(>|z|)
(intercept)  2.7818e+00  3.0768e-01  9.0411 < 2.2e-16 ***
wks          8.3740e-04  5.9981e-04  1.3961  0.16268
southyes     7.4398e-03  3.1959e-02  0.2328  0.81592
smsayes      -4.1833e-02  1.8960e-02 -2.2064  0.02736 *
marriedyes   -2.9851e-02  1.8982e-02 -1.5726  0.11582
exp          1.1313e-01  2.4713e-03 45.7795 < 2.2e-16 ***
I(exp^2)     -4.1886e-04  5.4605e-05 -7.6709 1.710e-14 ***
bluecolyes   -2.0705e-02  1.3783e-02 -1.5022  0.13304
ind          1.3604e-02  1.5239e-02  0.8927  0.37202
unionyes     3.2771e-02  1.4910e-02  2.1979  0.02796 *
sexmale      1.3092e-01  1.2667e-01  1.0335  0.30135
blackyes     -2.8575e-01  1.5572e-01 -1.8350  0.06651 .
ed           1.3794e-01  2.1251e-02  6.4912 8.518e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

----- Overall Statistics -----
Total Sum of Squares      : 243.04
Sum of Squares Residuals  : 95.947
Rsqr                      : 0.60522
F                          : 530.318
P(F>0)                    : 2.88658e-15

```

3.5 Variables instrumentales

Un ou l'ensemble des modèles de panel peut être estimé avec des variables instrumentales, en précisant la liste des variables utilisées comme instruments. Les instruments peuvent être précisés de deux manières différentes :

- en spécifiant la liste totale des instruments (à l'aide de l'argument `instrument` de `plm`),
- en spécifiant, d'une part la liste des instruments extérieurs au modèle (argument `instrument`) et d'autre part la liste des variables du modèles qui sont considérées comme corrélées avec le terme d'erreur (argument `endog`).

La méthode utilisée peut être spécifiée à l'aide de l'argument `inst.method` :

- `bvk`, d'après BALESTRA et VARADHARAJAN-KRISHNAKUMAR (1987), la valeur par défaut,
- `baltagi`, d'après BALTAGI (1981).

Nous illustrons l'estimation d'un modèle de panel avec variables instrumentales avec les données `Crime`³. La même estimation est réalisée en utilisant la première technique (`cr1`) et la deuxième (`cr2`). Les variables `prbarr` et `polpc`

³Voir BALTAGI (2001), pp.119-120.

sont considérées comme endogènes et il y a deux instruments extérieurs au modèle, `taxpc` et `mix` :

```
> data(Crime)
> pdata.frame(Crime, "county", "year")
> form = log(crmrte) ~ log(prbarr) + log(polpc) + log(prbconv) +
+   log(prbpris) + log(avgsen) + log(density) + log(wcon) + log(wtuc) +
+   log(wtrd) + log(wfir) + log(wser) + log(wmfg) + log(wfed) +
+   log(wsta) + log(wloc) + log(pctymle) + log(pctmin) + region +
+   smsa + year
> inst = ~log(prbconv) + log(prbpris) + log(avgsen) + log(density) +
+   log(wcon) + log(wtuc) + log(wtrd) + log(wfir) + log(wser) +
+   log(wmfg) + log(wfed) + log(wsta) + log(wloc) + log(pctymle) +
+   log(pctmin) + region + smsa + log(taxpc) + log(mix) + year
> inst2 = ~log(taxpc) + log(mix)
> endog = ~log(prbarr) + log(polpc)
> cr = plm(form, data = Crime)
> cr1 = plm(form, inst, data = Crime)
> cr2 = plm(form, inst2, endog, data = Crime)
> summary(cr2$random)
```

```
-----
----- Model Description -----
Oneway (individual) effect
Random Effect Model (Swamy-Arora's transformation)
Instrumental variable estimation (Balestra-Varadharajan-Krishnakumar's transformation)
Model Formula      : log(crmrte) ~ log(prbarr) + log(polpc) +
                    log(prbconv) + log(prbpris) +
                    log(avgsen) + log(density) +
                    log(wcon) + log(wtuc) + log(wtrd) +
                    log(wfir) + log(wser) + log(wmfg) +
                    log(wfed) + log(wsta) + log(wloc) +
                    log(pctymle) + log(pctmin) +
                    region + smsa + year
Endogenous Variables : ~log(prbarr) + log(polpc)
Instrumental Variables : ~log(taxpc) + log(mix)

-----
----- Panel Dimensions -----
Balanced Panel
Number of Individuals      : 90
Number of Time Observations : 7
Total Number of Observations : 630

-----
----- Effects -----
var  std.dev share
idiosyncratic 0.022269 0.149228 0.326
individual    0.046036 0.214561 0.674
theta      : 0.74576

-----
----- Residuals -----
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-5.02e+00	-4.76e-01	2.73e-02	7.11e-16	5.26e-01	3.19e+00

Coefficients				
	Estimate	Std. Error	z-value	Pr(> z)
(intercept)	-0.4538241	1.7029840	-0.2665	0.789864
log(prbarr)	-0.4141200	0.2210540	-1.8734	0.061015 .
log(polpc)	0.5049285	0.2277811	2.2167	0.026642 *
log(prbconv)	-0.3432383	0.1324679	-2.5911	0.009567 **
log(prbpris)	-0.1900437	0.0733420	-2.5912	0.009564 **
log(avgsen)	-0.0064374	0.0289406	-0.2224	0.823977
log(density)	0.4343519	0.0711528	6.1045	1.031e-09 ***
log(wcon)	-0.0042963	0.0414225	-0.1037	0.917392
log(wtuc)	0.0444572	0.0215449	2.0635	0.039068 *
log(wtrd)	-0.0085626	0.0419822	-0.2040	0.838387
log(wfir)	-0.0040302	0.0294565	-0.1368	0.891175
log(wser)	0.0105604	0.0215822	0.4893	0.624620
log(wmfg)	-0.2017917	0.0839423	-2.4039	0.016220 *
log(wfed)	-0.2134634	0.2151074	-0.9924	0.321023
log(wsta)	-0.0601083	0.1203146	-0.4996	0.617362
log(wloc)	0.1835137	0.1396721	1.3139	0.188884
log(pctymle)	-0.1458448	0.2268137	-0.6430	0.520214
log(pctmin)	0.1948760	0.0459409	4.2419	2.217e-05 ***
regionwest	-0.2281780	0.1010317	-2.2585	0.023916 *
regioncentral	-0.1987675	0.0607510	-3.2718	0.001068 **
smsayes	-0.2595423	0.1499780	-1.7305	0.083535 .
year82	0.0132140	0.0299923	0.4406	0.659518
year83	-0.0847676	0.0320008	-2.6489	0.008075 **
year84	-0.1062004	0.0387893	-2.7379	0.006184 **
year85	-0.0977398	0.0511685	-1.9102	0.056113 .
year86	-0.0719390	0.0605821	-1.1875	0.235045
year87	-0.0396520	0.0758537	-0.5227	0.601153

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Overall Statistics	
Total Sum of Squares	: 1354.7
Sum of Squares Residuals	: 557.64
Rsq	: 0.58836
F	: 33.1494
P(F>0)	: 7.77156e-16

3.6 Modèle à coefficients individuels

Si le nombre d'observations temporelles est suffisant, le modèle peut également être spécifié de manière à ce que les différents coefficients soient a priori différents d'un individu à un autre. Dans ce cas, la même équation est estimée pour chaque individu. Cette estimation peut être réalisée avec `plm` de deux manières différentes :

- en utilisant la commande `nopool`,
 - en utilisant la commande `plm` avec l'argument `np` fixé à `TRUE`.
- Avec les données `Grunfeld`, on obtient le résultat suivant :

```
> z <- nopool(inv ~ value + capital, data = Grunfeld)
> print(z)
```

	(intercept)	value	capital
1	-149.7824533	0.119280833	0.371444807
2	-49.1983219	0.174856015	0.389641889
3	-9.9563065	0.026551189	0.151693870
4	-6.1899605	0.077947821	0.315718185
5	22.7071160	0.162377704	0.003101737
6	-8.6855434	0.131454842	0.085374274
7	-4.4995344	0.087527198	0.123781407
8	-0.5093902	0.052894126	0.092406492
9	-7.7228371	0.075387943	0.082103558
10	0.1615186	0.004573432	0.437369190

```
> summary(z)
```

	(intercept)	value	capital
Min.	:-149.782	Min. :0.004573	Min. :0.003102
1st Qu.:	-9.639	1st Qu.:0.058518	1st Qu.:0.087132
Median :	-6.956	Median :0.082738	Median :0.137738
Mean :	-21.368	Mean :0.091285	Mean :0.205264
3rd Qu.:	-1.507	3rd Qu.:0.128411	3rd Qu.:0.357513
Max. :	22.707	Max. :0.174856	Max. :0.437369

Le résultat est un objet de classe `nopool`. L'impression de cet objet présente les coefficients estimés pour chaque individu. La méthode `summary` présente des statistiques descriptives pour ces coefficients.

Dans la commande ci-dessous, l'estimation est réalisée à partir de la commande `plm` :

```
> z <- plm(inv ~ value + capital, data = Grunfeld, np = TRUE)
> znp <- z$nopool
```

Le résultat de l'estimation du modèle `nopool` est alors un élément supplémentaire de l'objet renvoyé par `plm`.

3.7 Panel non-cylindré

`plm` propose un support limité pour les panels non-cylindrés. Nous présentons ci-dessous les résultats d'une estimation réalisée sur les données `Hedonic`⁴.

```
> form = mv ~ crim + zn + indus + chas + nox + rm + age + dis +
+       rad + tax + ptratio + blacks + lstat
> ba = plm(form, data = Hedonic)
> summary(ba$random)
```

⁴Voir BALTAGI (2001), p. 174.

```

-----
----- Model Description -----
Oneway (individual) effect
Random Effect Model (Swamy-Arora's transformation)
Model Formula      : mv ~ crim + zn + indus + chas + nox +
                    rm + age + dis + rad + tax +
                    ptratio + blacks + lstat

-----
----- Panel Dimensions -----
Unbalanced Panel
Number of Individuals      : 92
Number of Time Observations : from 1 to 30
Total Number of Observations : 506

-----
----- Effects -----
var  std.dev share
idiosyncratic 0.016965 0.130249 0.502
individual    0.016832 0.129738 0.498
theta :
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0.2915 0.5904 0.6655 0.6499 0.7447 0.8197

-----
----- Residuals -----
      Min.    1st Qu.    Median      Mean    3rd Qu.      Max.
-0.641000 -0.066100 -0.000519 -0.001990 0.069800 0.527000

-----
----- Coefficients -----
      Estimate Std. Error z-value Pr(>|z|)
(intercept)  9.6778e+00 2.0714e-01 46.7207 < 2.2e-16 ***
crim         -7.2338e-03 1.0346e-03 -6.9921 2.707e-12 ***
zn           3.9575e-05 6.8778e-04 0.0575 0.9541153
indus        2.0794e-03 4.3403e-03 0.4791 0.6318706
chasyes      -1.0591e-02 2.8960e-02 -0.3657 0.7145720
nox          -5.8630e-03 1.2455e-03 -4.7074 2.509e-06 ***
rm           9.1773e-03 1.1792e-03 7.7828 7.105e-15 ***
age          -9.2715e-04 4.6468e-04 -1.9952 0.0460159 *
dis          -1.3288e-01 4.5683e-02 -2.9088 0.0036279 **
rad           9.6863e-02 2.8350e-02 3.4168 0.0006337 ***
tax          -3.7472e-04 1.8902e-04 -1.9824 0.0474298 *
ptratio      -2.9723e-02 9.7538e-03 -3.0473 0.0023089 **
blacks       5.7506e-01 1.0103e-01 5.6920 1.256e-08 ***
lstat        -2.8514e-01 2.3855e-02 -11.9533 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

-----
----- Overall Statistics -----
Total Sum of Squares      : 893.08
Sum of Squares Residuals  : 8.6843
Rsq                       : 0.99028
F                          : 3854.18

```

P(F>0) : 0

4 Tests

4.1 Tests de coefficients identiques

On teste ici l'hypothèse que les mêmes coefficients associés à chaque variable sont les mêmes pour tous les individus. Il s'agit d'un test de F classique, comparant les résultats d'un modèle estimé globalement pour l'ensemble des individus et d'un modèle obtenu en estimant une équation pour chaque individu. La commande `pooltest` permet de réaliser ce test. Cette commande prend pour principal argument un objet de classe `plms`. Si le modèle a été estimé par `plm` avec l'argument `np=F`, il faut spécifier un deuxième argument de classe `nopool` (dans le cas inverse, le modèle `nopool` est un élément de l'objet `plms`).

Un troisième argument `effect` doit être fixé à vrai ou faux (valeur par défaut), suivant que sous H_0 , les constantes individuelles sont supposées être différentes ou identiques⁵ :

```
> form = inv ~ value + capital
> znp = nopool(form, data = Grunfeld)
> zplm = plm(form, data = Grunfeld)
> pooltest(zplm, znp)

F statistic

data:  zplm
F = 27.7486, df1 = 27, df2 = 170, p-value < 2.2e-16

> pooltest(zplm, znp, effect = T)

F statistic

data:  zplm
F = 5.7805, df1 = 18, df2 = 170, p-value = 1.219e-10

> z = plm(form, data = Grunfeld, effect = "time", np = TRUE)
> pooltest(z, effect = F)

F statistic

data:  z
F = 1.1204, df1 = 57, df2 = 140, p-value = 0.2928
```

4.2 Tests sur les effets

4.2.1 Tests de multiplicateur de Lagrange

Ces tests de présence d'effets individuels et/ou temporels sont basés sur les résultats du modèle de pooling. La commande `plmtest` permet de calculer ces

⁵Les exemples ci-dessous sont présentés par BALTAGI (2001), pp. 57–58.

tests, en indiquant en argument soit un objet de classe `plms`, soit un objet de classe `plm` (en l'occurrence le résultat d'un modèle `pooling`).

Deux arguments supplémentaires permettent de préciser le test à calculer. `type` peut être fixé à :

- `bp`, test de BREUSCH–PAGAN (1980), la valeur par défaut,
 - `honda`, d'après HONDA (1985),
 - `kw`, d'après KING et WU (1997).
- Le type d'effet testé est précisé grâce à l'argument `effect` fixé à :
- `individual` pour des effets individuels (la valeur par défaut),
 - `time` pour des effets temporels,
 - `twoways` pour des effets individuels et temporels.

Différents exemple d'utilisation de tests de multiplicateur de Lagrange sont présentés ci-dessous⁶.

```
> library(Ecdat)
> g <- plm(inv ~ value + capital, data = Grunfeld)
> plmtest(g)

Lagrange Multiplier Test - individual effects (Breush-Pagan)

data:  g
chi2 = 798.1615, df = 1, p-value < 2.2e-16

> plmtest(g, effect = "time")

Lagrange Multiplier Test - time effects (Breush-Pagan)

data:  g
chi2 = 6.4539, df = 1, p-value = 0.01107

> plmtest(g, type = "honda")

Lagrange Multiplier Test - individual effects (Honda)

data:  g
normal = 28.2518, p-value < 2.2e-16

> plmtest(g, type = "ghm", effect = "twoways")

Lagrange Multiplier Test - two-ways effects (Gourierroux, Holly and
Monfort)

data:  g
chi2 = 798.1615, df = 2, p-value < 2.2e-16

> plmtest(g, type = "kw", effect = "twoways")

Lagrange Multiplier Test - two-ways effects (King and Wu)

data:  g
normal = 21.8322, df = 2, p-value < 2.2e-16
```

⁶Voir BALTAGI (2001), p. 65.

4.2.2 Tests de F

La présence d'effets peut également être testée à partir de la comparaison des résultats des modèles `within` et `pooling`, sous la forme d'un test de F. La commande `pFtest` permet de calculer ce type de test, en indiquant comme argument soit un objet de classe `plms`, soit deux objets de classe `plm`, contenant respectivement les résultats d'un modèle `within` et d'un modèle `pooling`. Différents exemples d'utilisation de `pFtest` sont présentés ci-dessous⁷ :

```
> library(Ecdat)
> gi <- plm(inv ~ value + capital, data = Grunfeld)
> gt <- plm(inv ~ value + capital, data = Grunfeld, effect = "time")
> gd <- plm(inv ~ value + capital, data = Grunfeld, effect = "twoways")
> pFtest(gi)
```

F test for effects

```
data:  gi
F = 49.1766, df1 = 9, df2 = 188, p-value < 2.2e-16
```

```
> pFtest(gi$within, gi$pooling)
```

F test for effects

```
data:  gi$within and gi$pooling
F = 49.1766, df1 = 9, df2 = 188, p-value < 2.2e-16
```

```
> pFtest(gt)
```

F test for effects

```
data:  gt
F = 0.5229, df1 = 9, df2 = 188, p-value = 0.8569
```

```
> pFtest(gd)
```

F test for effects

```
data:  gd
F = 17.4031, df1 = 28, df2 = 169, p-value < 2.2e-16
```

4.3 Test d'Hausman

Le test d'HAUSMAN (1978) est un test basé sur la comparaison de deux modèles :

- sous l'hypothèse nulle, les deux modèles sont convergents et l'un des deux est plus efficace que l'autre,
- sous l'hypothèse alternative, seul le premier des deux modèles est convergent.

⁷Voir BALTAGI (2001), p. 65.

La commande `phptest` permet de calculer le test d’Hausman. Elle peut prendre comme argument un objet de classe `plms`, dans ce cas les deux modèles concernés sont les modèles `within` et `random` (usage le plus courant en économétrie de panel). Elle peut également prendre pour argument deux objets de classe `plm`. Différents exemples d’utilisation du test d’HAUSMAN sont présentés ci-dessous⁸.

```
> g <- plm(inv ~ value + capital, data = Grunfeld)
> phptest(g)
```

Hausman Test

```
data: g
chi2 = 2.3304, df = 2, p-value = 0.3119
```

```
> phptest(g$between, g$random)
```

Hausman Test

```
data: g$between and g$random
chi2 = 2.1314, df = 3, p-value = 0.5456
```

5 Bibliographie

Amemiya, T. (1971), The estimation of the variances in a variance-components model, *International Economic Review*, 12, pp.1–13.

Balestra, P. et J. Varadharajan–Krishnakumar (1987), Full information estimations of a system of simultaneous equations with error components structure, *Econometric Theory*, 3, pp.223–246.

Baltagi, B.H. (1981), Simultaneous equations with error components, *Journal of econometrics*, 17, pp.21–49.

Baltagi, B.H. (2001) *Econometric Analysis of Panel Data*. John Wiley and sons. ltd.

Breusch, T.S. et A.R. Pagan (1980), The Lagrange multiplier test and its applications to model specification in econometrics, *Review of Economic Studies*, 47, pp.239–253.

Gourieroux, C., A. Holly et A. Monfort (1982), Likelihood ratio test, Wald test, and Kuhn–Tucker test in linear models with inequality constraints on the regression parameters, *Econometrica*, 50, pp.63–80.

Hausman, G. (1978), Specification tests in econometrics, *Econometrica*, 46, pp.1251–1271.

Hausman, J.A. et W.E. Taylor (1981), Panel data and unobservable individual effects, *Econometrica*, 49, pp.1377–1398.

⁸Voir BALTAGI (2001), p. 71.

Honda, Y. (1985), Testing the error components model with non-normal disturbances, *Review of Economic Studies*, 52, pp.681–690.

King, M.L. et P.X. Wu (1997), Locally optimal one-sided tests for multiparameter hypotheses, *Econometric Reviews*, 33, pp.523–529.

Nerlove, M. (1971), Further evidence on the estimation of dynamic economic relations from a time-series of cross-sections, *Econometrica*, 39, pp.359–382.

Swamy, P.A.V.B. et S.S. Arora (1972), The exact finite sample properties of the estimators of coefficients in the error components regression models, *Econometrica*, 40, pp.261–275.

Wallace, T.D. et A. Hussain (1969), The use of error components models in combining cross section with time series data, *Econometrica*, 37(1), pp.55–72.