Paneldaten in R

Empirische Wirtschaftsforschung

Juni 2022

Hier demonstriere ich anhand eines sehr einfachen Beispiels, wie mit R einfache Panel-Modelle geschätzt werden können, d.h. Modelle für Daten, die sowohl eine Querschnitts- als auch eine Zeitdimension haben.

Datenformate

Wir beginnen mit einem sehr einfachen Datensatz, nämlich für 4 Individuen (A, B, C, D) über drei Zeitperioden (2020, 2021, 2022)

Prinzipiell unterscheidet man zwischen dem für Querschnittsanalysen üblichen `wide`-Format

Tabelle 1:

	year	x_A	y_A	x_B	y_B	x_C	y_C	x_D	y_D
1	2020	1	8	5	2	4	5	2	2
2	2021	2	5	3	3	2	3	4	7
3	2022	3	6	3	2	4	9	5	4

und dem für Panel-Analysen erforderlichem `long`-Format

Tabelle 2:

	indiv	year	x	y
1	A	2020	1	8
2	A	2021	2	5
3	A	2022	3	6
4	B	2020	5	2
5	B	2021	3	3
6	B	2022	3	2
7	C	2020	4	5
8	C	2021	2	3
9	C	2022	4	9
10	D	2020	2	2
11	D	2021	4	7
12	D	2022	5	4

Üblicherweise werden die Daten aus einer Datenquelle eingelesen, aber für dieses einfache Beispiel legen wir uns manuell einen sehr einfachen Datensatz (`data.frame`) im `wide`-Format an

```
dat_wide <- data.frame(year = 2020:2022,
                       x_A = c(1, 2, 3), y_A = c(8, 5, 6),
                       x_B = c(5, 3, 3), y_B = c(2, 3, 2),
                       x_C = c(4, 2, 4), y_C = c(5, 3, 9),
                       x_D = c(2, 4, 5), y_D = c(2, 7, 4)
                       )
dat_wide
```

```
##   year x_A y_A x_B y_B x_C y_C x_D y_D
## 1 2020   1   8   5   2   4   5   2   2
## 2 2021   2   5   3   3   2   3   4   7
## 3 2022   3   6   3   2   4   9   5   4
```

das heißt, wir beobachten zwei Variablen, `x`, `y` für jedes der vier Individuen (A, B, C, D) über drei Jahre (2020 - 2022).

Vom `wide`-Format zum `long`-Format

Um diese Daten in ein für Panel-Schätzungen erforderliches *long*-Format zu bringen (d.h. die Daten zu *stacken*) kann z.B. das `reshape2 package` verwendet werden.

Das `R`-package `reshape2` stellt zwei zentralen Funktionen `melt()` und `dcast()` zur Verfügung. Mit `melt()` werden die Daten im `wide`-Format in ein `long`-Format mit einer einzigen `value`-Spalte *`geschmolzen´*, und mitdcast()wird daraus das gewünschtelong‘-Format mit einer Spalte für jede Variable geformt.

Es sei vorausgeschickt, dass der Weg vom `wide`-Format zum `long`-Format deutlich umständlicher ist als der umgekehrte Weg, und dass dieser Schritt in vielen Fällen nicht erforderlich ist, da Daten oft bereits im `long`-Format vorliegen.

Falls die Daten nur im `wide` Format vorliegen müssen wir die Information, um welches Individuum es sich handelt, in den Variablennamen kodiert sein, deshalb müssen wir die Individuendaten aus den Variablennamen extrahieren. In unserem Beispiel machen wir es uns zunutze, dass der erste Teil des Variablennamens die Variable bezeichnet (x,y), und der zweite Teil die Namen der Individuuen (A, B, C, D). Die zwei Teile sind in diesem Beispiel durch das Unterstreichungszeichen (`_, underscore`) getrennt, und dieses verwenden wir, um mit `colspit()` zwei neue Datenreihen mit diesen Bestanteilen anzufügen.

Wir verwenden das package `reshape2` mit der darin bereitgestellten Funktion `melt()`, als *identifier* wählen wir das Jahr. Mit der R Funktion `colsplit()` erzeugen wir aus den Variablennamen zwei neu Variablen, die wir `var` und `indiv` nennen, und fügen diese als neue Spalten mittels `cbind()` hinzu

```
library(reshape2)

dat_long1 <- melt(dat_wide, id = "year")
dat_long1[1:5, ]  # nur Zeilen 1:5 und alle Spalten zeigen
```

```
##   year variable value
```

```
## 1 2020      x_A     1
## 2 2021      x_A     2
## 3 2022      x_A     3
## 4 2020      y_A     8
## 5 2021      y_A     5
```

```
dat_long2 <- cbind(dat_long1, colsplit(dat_long1$variable, "_",
                                       names = c("var", "indiv")))
## alternativ:
#dat_long2$var   <- substr(dat_long1$variable, 1, 1)
#dat_long2$indiv <- substr(dat_long1$variable, 3, 3)

dat_long2[1:5, ]
```

```
##   year variable value var indiv
## 1 2020      x_A     1   x     A
## 2 2021      x_A     2   x     A
## 3 2022      x_A     3   x     A
## 4 2020      y_A     8   y     A
## 5 2021      y_A     5   y     A
```

```
dat_long2$variable <- NULL # variable entfernen (löschen)
# gewünschtes Format erzeugen
dat_long <- dcast(dat_long2, indiv + year ~ var)
dat_long
```

```
##    indiv year x y
## 1      A 2020 1 8
## 2      A 2021 2 5
## 3      A 2022 3 6
## 4      B 2020 5 2
## 5      B 2021 3 3
## 6      B 2022 3 2
## 7      C 2020 4 5
## 8      C 2021 2 3
## 9      C 2022 4 9
## 10     D 2020 2 2
## 11     D 2021 4 7
## 12     D 2022 5 4
```

Panel Modelle

Hier stellen wir nur die allereinfachsten Modelle vor und vergleichen diese mit den entsprechenden *Least squared dummy variable* (LSDV) Modellen. Für eine ausführliche Einführung/Diskussion siehe jedes einführende Lehrbuch oder insbesondere Croissant and Millo (2018).

Das derzeit verbreiteste `R`-package zur Schätzung von Panelmodellen ist `plm` (Millo, 2017). Dieses muss wie üblich einmalig installiert werden und mit `library(plm)` aktiviert werden.

```
library(plm)
```

Mit Hilfe dieses packages können alle üblichen Modelle geschätzt und getestet werden. Zuerst legen wir uns mit Hilfe des `plm`–packages einen `pdata.frame` an, der die spezielle Datenstruktur berücksichtigt. Dazu müssen wir die zwei Indizes (*identifier*) angeben, immer zuerst den Individuum– und dann den Zeitindex

```
paneldat <- pdata.frame(dat_long, index = c("indiv", "year"))
paneldat
```

```
##        indiv year x y
## A-2020     A 2020 1 8
## A-2021     A 2021 2 5
## A-2022     A 2022 3 6
## B-2020     B 2020 5 2
## B-2021     B 2021 3 3
## B-2022     B 2022 3 2
## C-2020     C 2020 4 5
## C-2021     C 2021 2 3
## C-2022     C 2022 4 9
## D-2020     D 2020 2 2
## D-2021     D 2021 4 7
## D-2022     D 2022 5 4
```

Man sieht, dass der Index die Panel-Struktur korrekt abbildet.

Pooling-Modell

Die einfachste Modell besteht einfach darin, OLS auf die Daten im `long`-Format anzuwenden.

$$y_{it} = \alpha + \beta x_{it} + \varepsilon_{it}$$

Die Koeffizienten α und β werden über alle Beobachtungen geschätzt und haben deshalb keine Indizes.

Für die Darstellung des Ergebnisses wird das `stargazer package` (Hlavac, 2018) verwendet.

```
eq_pool <- plm(y ~ x, model = "pooling", data = paneldat)
# Vergleich mit OLS
eq_pool_lm <- lm(y ~ x, data = dat_long)
stargazer(eq_pool, eq_pool_lm, type = "latex", title = "Pooling-Modell",
          header = FALSE)
```

Dieses Modell wird eher selten verwendet. Das vermutlich am häufigsten geschätzte Panel-Modell berücksichtigt Individuen-spezifische fixe Effekte, d.h., das *within*-Modell.

'Between' - Modell

Beim *between-model* werden (meist) Individuen-Durchschnitte über die Zeit berechnet, und auf diese Durchschnitte wird OLS angewandt

Tabelle 3: Pooling-Modell

	Dependent variable:	
	y	
	panel linear	*OLS*
	(1)	(2)
x	−0.132	−0.132
	(0.604)	(0.604)
Constant	5.085**	5.085**
	(2.047)	(2.047)
Observations	12	12
R^2	0.005	0.005
Adjusted R^2	−0.095	−0.095
Residual Std. Error		2.537 (df = 10)
F Statistic (df = 1; 10)	0.048	0.048
Note:	*p<0.1; **p<0.05; ***p<0.01	

$$\bar{y}_{i\bullet} = \alpha + \beta \bar{x}_{i\bullet} + \varepsilon_i$$

```
# OLS, indiv-means
avg <- aggregate(dat_long, list(dat_long$indiv), mean, simplify = TRUE)

## alternativ mit tidyverse
# library(dplyr)
# avg1 <- dat_long %>% group_by(dat_long$indiv) %>% summarise_all(mean)

eq_ols      <- lm(y ~ x, data = avg)
eq_betw  <- plm(y ~ x, model = "between", data = paneldat)

stargazer(eq_ols, eq_betw, type = "latex",
          title = "Between-Modell",
          header = FALSE)
```

Individuen-fixe Effekte (within-Modell)

Dieses Modell haben wir bereits im Abschnitt zu Dummy-Variablen (Kapitel *Deskriptive Regressionsanalyse*) diskutiert.

Da individuenspezifische Interzepte zugelassen werden wird dies häufig kurz geschrieben als

$$y_{it} = \alpha_i + \beta x_{it} + \varepsilon_{it}$$

Tabelle 4: Between-Modell

	Dependent variable:	
	y	
	OLS	*panel linear*
	(1)	(2)
x	−1.647	−1.647
	(1.056)	(1.056)
Constant	9.882	9.882
	(3.420)	(3.420)
Observations	4	4
R^2	0.549	0.549
Adjusted R^2	0.324	0.324
Residual Std. Error	1.451 (df = 2)	
F Statistic (df = 1; 2)	2.435	2.435
Note:	*p<0.1; **p<0.05; ***p<0.01	

Hinweis: Dies ist lediglich eine verkürzte Schreibweise für

$$y_{it} = \alpha_1 + \sum_{h=2}^{n} \alpha_h d_h + \beta x_{it} + \varepsilon_{it}$$

wobei d die Individuendummies sind.

Dieses Modell kann entweder mit Hilfe individuenspezifischer Dummy-Variablen, oder einfacher und numerisch äquivalent (siehe *Frisch-Waugh-Lovell* Theorem), mittels *Mittelwerttransformierter Daten* geschätzt werden.

```
eq_within <- plm(y ~ x, model = "within", data = paneldat)
# Vergleich mit OLS und Dummies
eq_within_lm <- lm(y ~ x + factor(indiv), data = dat_long)
stargazer(eq_within, eq_within_lm, type = "latex",
          title = "Individuen-fixe Effekte (within-Modell)",
          header = FALSE)
```

Die individuen-spezifischen Interzepte (Koeffizienten der Dummies) werden standardmäßig nicht ausgegeben, da sie selten von Interesse sind, können aber einfach mit `fixef()` angefordert werden

```
fixef(eq_within, type = "dfirst")
```

```
##       B       C       D
## -4.9722 -1.4444 -2.9722
```

Tabelle 5: Individuen-fixe Effekte (within-Modell)

	Dependent variable:	
	y	
	panel linear	*OLS*
	(1)	(2)
x	0.583	0.583
	(0.623)	(0.623)
factor(indiv)B		-4.972^{**}
		(2.045)
factor(indiv)C		-1.444
		(1.948)
factor(indiv)D		-2.972
		(2.045)
Constant		5.167^{**}
		(1.762)
Observations	12	12
R^2	0.111	0.496
Adjusted R^2	-0.396	0.208
Residual Std. Error		2.157 (df = 7)
F Statistic	0.877 (df = 1; 7)	1.723 (df = 4; 7)
Note:		*p<0.1; **p<0.05; ***p<0.01

First differences

Eine alternative Methode indidividuen-fixe Effekte zu berücksichtigen ist die Schätzung in ersten Differenzen. Dabei ist zu beachten, dass keine Differenzen *zwischen* Individuen gebildet werden dürfen, und deshalb die erste Beobachtung jedes Individuums verloren geht.

```
eq_fd <- plm(y ~ x, model = "fd", data = paneldat)
# Vergleich mit OLS
x_diff <- paneldat$x - lag(paneldat$x)
y_diff <- paneldat$y - lag(paneldat$y)
eq_fd_lm <- lm(y_diff ~ x_diff)
stargazer(eq_fd, eq_fd_lm, type = "latex",
          title = "First differences", header = FALSE)
```

Tabelle 6: First differences

	Dependent variable:	
	y	y_diff
	panel linear	*OLS*
	(1)	(2)
x	0.979	
	(0.790)	
x_diff		0.979
		(0.790)
Constant	0.133	0.133
	(1.217)	(1.217)
Observations	8	8
R^2	0.204	0.204
Adjusted R^2	0.071	0.071
Residual Std. Error		3.338 (df = 6)
F Statistic (df = 1; 6)	1.537	1.537
Note:	*p<0.1; **p<0.05; ***p<0.01	

Zeit-fixe Effekte (within-Modell)

Analog können auch Zeit-fixe Effekte geschätzt werden, diese allein zu schätzen ist allerdings weniger gebräuchlich

$$y_{it} = \gamma_t + \beta x_{it} + \varepsilon_{it}$$

```
eq_within_time <- plm(y ~ x, effect = "time", model = "within", data = dat_long)
# Vergleich mit OLS
eq_within_lm_time <- lm(y ~ x + factor(year), data = dat_long)
```

```
stargazer(eq_within_time, eq_within_lm_time, type = "latex",
          title = "Zeit-fixe Effekte", header = FALSE)
```

Tabelle 7: Zeit-fixe Effekte

	Dependent variable:	
	y	
	panel linear	*OLS*
	(1)	(2)
x	−0.274	−0.274
	(0.703)	(0.703)
factor(year)2021		0.181
		(1.966)
factor(year)2022		1.206
		(2.028)
Constant		5.073*
		(2.524)
Observations	12	12
R^2	0.019	0.052
Adjusted R^2	−0.349	−0.304
Residual Std. Error		2.769 (df = 8)
F Statistic	0.152 (df = 1; 8)	0.145 (df = 3; 8)
Note:		*p<0.1; **p<0.05; ***p<0.01

Zwei-Weg Modell: Individuen- und Zeit-fixe Effekte (twoways-Modell)

Dies entspricht numerisch wieder einer einfachen OLS-Schätzung mit z.B. Länder- *und* Jahres-Dummies

$$y_{it} = \alpha_i + \gamma_t + \beta x_{it} + \varepsilon_{it}$$

```
eq_twoways <- plm(y ~ x, effect = "twoways", data = paneldat)
# Vergleich mit OLS
eq_twoways_lm <- lm(y ~ x + factor(indiv) + factor(year), data = dat_long)
stargazer(eq_twoways, eq_twoways_lm, type = "latex", header = FALSE,
          title = "Individuen- und Zeit-fixe Effekte (twoways-Modell)"
          )
```

Da Paneldaten sowohl eine Querschnitts- als auch eine Zeitdimension haben ist sowohl *Heteroskedastizität* als auch *Autokorrelation* zu beachten.

Tabelle 8: Individuen- und Zeit-fixe Effekte (twoways-Modell)

	Dependent variable:	
	y	
	panel linear	*OLS*
	(1)	(2)
x	0.517	0.517
	(0.805)	(0.805)
factor(indiv)B		−4.862
		(2.460)
factor(indiv)C		−1.356
		(2.324)
factor(indiv)D		−2.862
		(2.460)
factor(year)2021		0.379
		(1.797)
factor(year)2022		0.612
		(1.885)
Constant		4.969*
		(2.316)
Observations	12	12
R^2	0.076	0.507
Adjusted R^2	−1.032	−0.084
Residual Std. Error		2.525 (df = 5)
F Statistic	0.412 (df = 1; 5)	0.857 (df = 6; 5)
Note:		*p<0.1; **p<0.05; ***p<0.01

In fast allen Fällen empfiehlt sich die Verwendung *robuster Standardfehler*, und wenn genügend Cluster (bei Individueneffekten die Anzahl der Individuen) vorliegen, die Verwendung *Cluster robuster Standardfehler* Standard.

In diesem Beispiel ist die Stichprobe natürlich viel zu klein, aber nur um die Befehle und den Output zu demonstrieren führen wir mit Hilfe des `plm`packages einen Breusch Pagan Test auf Heteroskedastizität durch und berechnen Cluster-robuste Standardfehler.

```
# Breusch Pagan Test
#library(lmtest)
plmtest(eq_within, type=c("bp"))
```

```
##
##  Lagrange Multiplier Test - (Breusch-Pagan) for balanced panels
##
## data:  y ~ x
## chisq = 0.11393, df = 1, p-value = 0.7357
## alternative hypothesis: significant effects
```

```
# Robuste Standardfehler
coeftest(eq_within, vcov=vcovHC(eq_within, type="sss", cluster="group"))
```

```
##
## t test of coefficients:
##
##   Estimate Std. Error t value Pr(>|t|)
## x  0.58333    0.54291  1.0745   0.3183
```

Random-effects Modelle

Bei `random-effects` Modellen wird angenommen, dass die individuenspezifischen Dummies keinen systematischen Auswirkungen auf die abhängige y Variable haben, und deshalb nicht im systematischen Teil der Regression berücksichtigt werden müssen. Individuenspezifische Unterschiede sind deshalb Teil des Störterms, und nicht Teil des systematischen Teils der Regression!

Trotzdem können sich die *Varianzen* zwischen den Individuen unterscheiden. Das `random-effects` Modell ist im wesentlichen ein *feasable generalized least squares estimator* (FGLS Modell), welches für die Schätzung der Varianz-Kovarianz Matrix individuenspezifische Varianzen zulässt, aber davon ausgeht, dass sich die Koeffizienten nicht zwischen den Individuen unterscheiden.

```
# random effects
eq_re <- plm(y ~ x, model = "random", data = paneldat)
stargazer(eq_re, type = "latex",
          title = "Random Effects", header = FALSE)
```

Wenn die Annnahme richtig ist, dass sich die Koeffizienten *nicht* zwischen den Individuen unterscheiden, sind sowohl das `within'-Modell als auch dasrandom-effectsModell konsistent`, `aber dasrandom-effects' Modell ist darüber hinaus effizient, da keine Freiheitsgrade durch die Schätzung der Individuendummies (bzw. Mittelwerttransformation) verloren gehen.

Ist hingegen die Annahme falsch, so ist das `random-effects` Modell weder effizient noch konsistent (*omitted variable bias*, d.h. systematisch verzerrt!), hingegen ist das 'within'-Modell in diesem Fall

Tabelle 9: Random Effects

	Dependent variable:
	y
x	0.007
	(0.595)
Constant	4.643**
	(2.049)
Observations	12
R^2	0.00002
Adjusted R^2	−0.100
F Statistic	0.0002
Note:	*p<0.1; **p<0.05; ***p<0.01

sowohl effizient als auch konsistent!

Da in ökonomischen Zusammenhängen selten davon ausgegangen werden kann, dass sich die Individuen in allen zeitinvarianten Eigenschaften identisch sind, werden `random-effects` Modelle eher seltener geschätzt.

Die gemeinsame Signifikanz aller individuenspezifischen Dummies kann mit einem einfachen F-Test getestet werden. Allerdings existiert für plm-Objekte keine anova Methode, deshalb greife ich hier auf die OLS Modelle zurück

```
eq_restr <- lm(y ~ x, data = dat_long)  ## pool
eq_unrestr <- lm(y ~ x + factor(indiv), data = dat_long) ## within

anova(eq_restr, eq_unrestr)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x
## Model 2: y ~ x + factor(indiv)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     10 64.358
## 2      7 32.583  3    31.775 2.2755 0.1669
```

Das Ergebnis zeigt, dass die individuenspezifischen Dummies auch gemeinsam nicht signifikant von Null verschieden sind ($p = $0.1669), was allerdings bei dieser Beobachtungszahl kein Wunder ist.

Mit einem *Hausman-Test* kann die Annahme, dass sich die Koeffizienten *nicht* zwischen den Individuen unterscheiden (d.h., dass sowohl OLS als auch das `random-effects` Modell konsistent sind), getestet werden.

```r
# random effects, hausman test
eq_wi <- plm(y ~ x, model = "within", data = dat_long)
eq_re <- plm(y ~ x, model = "random", data = dat_long)
phtest(eq_wi, eq_re)
```

```
##
##  Hausman Test
##
## data:  y ~ x
## chisq = 9.858, df = 1, p-value = 0.001691
## alternative hypothesis: one model is inconsistent
```

Ein Hausman Test (auch Durbin-Wu-Hausman-Test genannt) erlaubt auf Endogenität[1] zu testen, in diesem Fall kann er als ein Test auf *omitted Variables* interpretiert werden.

Das Resultat zeigt, dass mit hoher Wahrscheinlichkeit eines der beiden Modell inkonsistent ist, und da OLS in beiden Fällen konsistent ist, kann dies nur das *random effects* Modell sein. Wir vermuten einen *omitted variable bias* und würden vermutlich das *within* Modell bevorzugen. Allerdings ist der Hausman Test nur asymptotisch gültig, und deshalb bei dieser Beobachtungszahl nicht anwendbar!

Literatur

Croissant, Y. and G. Millo. 2018. *Panel Data Econometrics with R*. Wiley.

Hlavac, Marek. 2018. *stargazer: Well-Formatted Regression and Summary Statistics Tables*. Bratislava, Slovakia: Central European Labour Studies Institute (CELSI). R package version 5.2.2. **URL:** *https://CRAN.R-project.org/package=stargazer*

Millo, Giovanni. 2017. "Robust Standard Error Estimators for Panel Models: A Unifying Approach." *Journal of Statistical Software* 82(3):1–27.

[1] genauer auf Endogenität im ökonometrischen Sinne, d.h. inwieweit eine stochastische Abhängigkeit zwischen Regressoren und Störtermen existiert.