

Robust Inference in Panel Data Microeconometrics, Using R

Giovanni Millo*

January 4, 2021

1 Introduction

This chapter is written as a quick companion to the practicing microeconomist who wants to use R for her analysis. It is concerned with relaxing the restrictive hypotheses of error homoskedasticity and performing robust inference in the context of panel data. I will systematically reference two textbooks for more formal, technical and thorough treatments: Wooldridge (2010) as a world-leading treatment of the subject and the source for the first worked example I present and the general structure of the first half of the paper; and Croissant and Millo (2019) for a comprehensive textbook-level treatment of panel data econometrics with the R software.

This chapter has two main bodies: the first part is introducing the reader to the mechanics of panel data econometrics with R as a necessary step towards the second part, where I expand upon the various robust estimators available in the **plm** package (Croissant and Millo, 2008; Millo, 2017; Croissant and Millo, 2019) and how to employ them in a microeconomic context. To this end, first I present the estimators and tests in the order of Wooldridge, Ch. 10 with a particular focus on “robustifying” the analysis by employing the *robust*, or *clustered*, covariance matrix, which is likely to be the most appropriate choice in the majority of microeconomic applications. Then I present alternative sandwich estimators robust to different kinds of dependence in a data-rich context. Any formality, although minimal, is deferred to this second part in order to make the first as gentle and as practice-oriented as possible: therefore, perhaps unusually, the clustering estimator which is employed in the first part is defined at the beginning of the second.

The main perspective taken is that of fixed T , large N as in Wooldridge’s book and as is most common in microeconometrics; although in the last part of the chapter I will consider a second example with a longer time dimension and enlarge the scope of the treatment to a variety of “robust” covariance estimators appropriate for this setting beyond the popular “clustered” covariance employed in the first part.

In particular I will cover, in this order: estimating the basic models (RE, FE, FD) and using the results to test hypotheses of interest; relaxing the typical

*Insurance and AM Research at Generali Investments, piazza Duca degli Abruzzi 1, Trieste (Italy). Tel.: +39-040-671184, Fax: +39-040-671160, email: giovanni.millo@generali.com.

regularity assumptions on the error covariance to robustify the standard errors of the basic estimators by allowing for unrestricted general heteroskedasticity and within-individual correlation of model errors (so-called “clustered” standard errors); presenting alternative standard error estimators for different kinds of error dependence; applying the latter in combination with different specifications of a model’s individual and/or time effects. The increasing level of detail means that the casual reader can refer to the first part of the chapter and have the main methods “just working” in R, while proceeding to the latter sections once she is interested in understanding the many different options available and how they map to the underlying statistical theory.

I stress that reference to Wooldridge (2010) (or another textbook) is a necessary condition for sensible application of the software methods presented here, and that this chapter, unlike Croissant and Millo (2019), cannot be taken as a standalone primer on basic panel methods, its only purpose being to illustrate software operation and to put it into the perspective of the theoretical methods described elsewhere. The reader must be familiar with the subject to properly understand what is going on when a given command is issued.

1.1 Notation

Package names are in **bold**. Commands and parts thereof (like argument names) are in **typewriter** font. Arguments will follow the type: e.g., character values will be “quoted”, numeric values will not be. Variable (and model) names are in *italics*.

Given that Stata is perhaps the most popular software package in the field of microeconometrics, I will occasionally refer to the equivalent Stata command to help the reader follow what is done in R.

1.2 Software requirements

Before starting, it is necessary to load some additional software materials into the workspace. In fact, the R system is a large collection of software, of which only the base system and some core packages are loaded by default. Specialized extensions for various statistical tasks are loaded on a by-session basis: either in the structured form of *packages*, large standalone collections of functions, documentation and, often, datasets (like **plm**); or under form of individual functions or datasets. In the former case, the **library** function will load a complete package; in the latter, individual pieces of code (typically defining a function) can be *sourced* (i.e., executed) through the **source** function; individual example datasets can be loaded with **data**. The reader should keep in mind that **data** will only work with built-in example datasets contained in some R package: reading external data into the R session will instead require different statements. An example is presented in Section 4.

Next to **plm**, I will employ some other packages of econometric nature, mainly for the purpose of hypothesis testing: **lmtest** (Zeileis and Hothorn, 2002) and **car** (Fox and Weisberg, 2011).

```
> library(plm)
> library(car)
> library(lmtest)
```

Presentation of results and the preparation of documents in general can be made easier by outputting results from **R** directly into \LaTeX . Two useful packages for this purpose, although not the only options, are **xtable** (Dahl et al, 2019) and **texreg** (Leifeld, 2013).

```
> library(texreg)
> library(xtable)
```

The package **wooldridge** (Shea, 2018) contains the datasets from the "other" manual by Jeffrey Wooldridge, *Introductory Econometrics: a Modern Approach*.

```
> library(wooldridge)
> data(jtrain)
```

Lastly, this document – like the vast majority of my papers – has been prepared by dynamically weaving **R** code and \LaTeX markup through the **Sweave** utility (Leisch, 2002).

1.3 Data organization and conventions

Panel data have two dimensions, usually the individual unit of observation (family, firm, state, country...) and the time period. Estimators and tests need the indexing information in order to operate correctly. Some widely used econometric programs whose paradigm requires to operate on one dataset at a time require to organize it at the beginning, once for all, so that the indices are properly specified (like **xtset**-ting the dataset in Stata, or setting the indices of an EViews workfile). **R** is more flexible, allowing the coexistence of many different datasets in the workspace; therefore, the indices cannot but be an attribute of the (individual) dataset itself.

The convention in **plm** is that **data** can be any regular **data.frame**, provided that the individual index sits in the first column, the time index in the second. Alternatively, a regular **data.frame** with a different ordering of the columns can be used, provided that the **index** argument is specified within every call to estimators and tests (see the examples in the next Section). A third possibility is to make and employ a specialized object type, the **pdata.frame**, which will carry over the indexing information. This third way of operating is somewhat more sophisticated and, while it is necessary for operations and tests involving one (or more) panel variable(s), like cross-sectional dependence-, or panel unit root testing, it goes beyond the scope of this chapter. On **pdata.frames**, see Croissant and Millo (2019).

1.4 The robustification framework

R allows to separate the procedural step of obtaining estimates $\hat{\beta}$ from those of estimating their dispersion $SE(\hat{\beta})$ and presenting the well-known table of significance tests. Some words on the mechanics of the software are in order.

An *estimator* function will create an *estimated model object*; the latter containing $\hat{\beta}$ as well as residuals and other quantities of interest. In turn another estimator – this time for the estimators' covariance, and hence for $SE(\hat{\beta})$ – can feed on the estimated model for producing the standard errors *according to the chosen method*. Functions in the **summary** and **coeftest** families can draw on these results to compute and present *t*-tests in the familiar tabular format.

The method chosen to compute the SEs can be passed on to the function producing the t -statistics and diagnostics table in multiple ways. The “cleanest” way is by supplying a function, like `coeftest(mymodel, vcov=vcovHC)`; otherwise, one can provide a computed matrix, perhaps computing it on the fly inside the same call, along the lines of `coeftest(mymodel, vcov=vcovHC(mymodel))`. See Millo (2017), Croissant and Millo (2019, Ch.5) and the documentation for package **plm** for more detail on the robustification framework in the case of panel data methods; and Zeileis (2004, 2006) for the general approach to flexible covariance specification of which the one presented here is one special instance.

All operations in the following therefore share the same structure:

- estimate model of choice
- produce coefficients’ tables or individual statistical tests using either the “standard” (default) or some “special” covariance (e.g., clustering by firm)

2 Basic panel methods in R by way of example

As a worked introduction to panel methods in R, I go through the examples from Wooldridge (2010, Ch. 10) reviewing the practical aspects of estimation. The focus is on applied microeconometrics and on obtaining results; the reader will possibly compare the presented workflow with that of other software packages, e.g. Stata. For a more structured approach to the software package **plm** for panel data econometrics in R, see Croissant and Millo (2019); for another gentle introduction to panel methods in R, see also Henningsen and Henningsen (2019).

The job training grants example (Example 10.4 in Wooldridge, 2010) regards estimating the effect of job training grants on firm scrap rates. The dataset (*JTRAIN1* in the original source, *jtrain* in package “wooldridge”) contains observations on 54 firms for the years 1987, 1988, and 1989. Grants were not awarded in 1987, then some firms received grants in 1988, others received grants in 1989; a firm could not receive a grant twice. As per Wooldridge (2010), “since there are firms in 1989 that received a grant only in 1988, it is important to allow the grant effect to persist one period”.

Let us replicate the results in the original source with R. All references to Examples etc. are relative to Wooldridge (2010, Ch. 10), so I will henceforth omit the citation.

2.1 Random effects

Below I produce the estimates in Example 10.4: random effects. The random effects (RE) estimator is obtained from the `plm` function, setting the `model` argument to “random”. As observed above, the argument `index=c("fcode", "year")` must be set in order to let the program know the correct panel indices, much like Stata’s `xtset`. With respect to the latter, though, the `index` will have to be specified with every command; unless the columns of the dataset are ordered in the “standard” way: individual index first, time index second, then the data proper; in which case it is not necessary to specify anything.

```
> ## Example 10.4 (RE Estimation of the Effects of Job Training Grants)
> fm <- log(scrap) ~ d88 + d89 + union + grant + grant_1
```

```
> ## RE model
> re10.4 <- plm(fm, jtrain, index=c("fcode","year"), model="random")
```

The output from the estimating function is, as usual with R, not appearing immediately in the session log. It is rather stored into a *model* object, which can be inspected with the appropriate functions. To see the overall result of estimation, one can issue a call to `summary`.

```
> summary(re10.4)
```

```
Oneway (individual) effect Random Effect Model
(Swamy-Arora's transformation)
```

```
Call:
```

```
plm(formula = fm, data = jtrain, model = "random", index = c("fcode",
"year"))
```

```
Balanced Panel: n = 54, T = 3, N = 162
```

```
Effects:
```

```
                var std.dev share
idiosyncratic 0.2477  0.4977 0.114
individual    1.9322  1.3900 0.886
theta: 0.7975
```

```
Residuals:
```

```
      Min.   1st Qu.   Median   3rd Qu.   Max.
-2.546798 -0.223892  0.040554  0.255287  1.549791
```

```
Coefficients:
```

```
                Estimate Std. Error t-value Pr(>|t|)
(Intercept)  0.414833   0.243432  1.7041  0.09035 .
d88          -0.093452   0.109156 -0.8561  0.39324
d89          -0.269834   0.131650 -2.0496  0.04207 *
union         0.547802   0.410625  1.3341  0.18413
grant        -0.214696   0.147784 -1.4528  0.14830
grant_1      -0.377070   0.205352 -1.8362  0.06823 .
```

```
---
```

```
Signif. codes:  0
```

The statements defining the model formula, performing the estimation and lastly inspecting the results have been kept separate here, for the sake of exposition; and it can be good practice to do so as well. Nevertheless, it is common practice to do everything in one line, like

```
> summary(plm(log(scrap) ~ d88 + d89 + union + grant + grant_1,
+           jtrain, index=c("fcode","year"), model="random"))
```

The economic result of interest is whether either *grant* or its lagged value, *grant_1*, are significant. A joint significance test – in either χ^2 or *F* form – is performed in the following way, calling the `waldtest` function and supplying

the order numbers (or the names) of the regressors to test. For the sake of illustration, below I supply names in the χ^2 test (default), order numbers in the F test:

```
> ## Wald test for joint significance of grant and grant_1
> wald10.4 <- waldtest(re10.4, c("grant", "grant_1"))
> wald10.4.F <- waldtest(re10.4, 4:5, test="F")
```

As is customary in R, the testing function creates a test object in the workspace but does not automatically produce an output in the log. It is enough to call the object by name (or just some part of it, like e.g. the p -value) to display it (here, the test in χ^2 form):

```
> wald10.4

Wald test

Model 1: log(scrap) ~ d88 + d89 + union + grant + grant_1
Model 2: log(scrap) ~ d88 + d89 + union
  Res.Df Df  Chisq Pr(>Chisq)
1     156
2     158 -2  3.6645    0.1601
```

but it is perhaps more common, in practice – and it is what I will do henceforth in the paper, for the sake of compactness – to output the result on the fly, without assigning an object (below, the test in F -form):

```
> waldtest(re10.4, 4:5, test="F")

Wald test

Model 1: log(scrap) ~ d88 + d89 + union + grant + grant_1
Model 2: log(scrap) ~ d88 + d89 + union
  Res.Df Df      F Pr(>F)
1     156
2     158 -2  1.8323 0.1635
```

2.1.1 Robustifying inference

Wooldridge (10.4.2) suggests using a robust covariance matrix for the parameter estimates. The robust covariance estimator is the one described in Wooldridge (2010, Ch. 7) and known as the “clustered”, or “Arellano” (from Arellano, 1987) covariance. This method is robust to arbitrary heteroskedasticity, both between and within individuals, and to arbitrary serial correlation within the same individual; on the contrary, it rules out any correlation between different individuals. Its statistical properties depend on the number of individuals being “large” with respect to that of time periods. For all these reasons, it is usually the estimator of choice in the context of micro panels, all the more so if individuals are sampled independently from a large population. It is implemented as the default in the appropriate panel method `vcovHC.plm` (Millo, 2017) of the generic function `vcovHC` (Zeileis, 2006), and as such it will be automatically applied by R to panel models and tests whenever `vcovHC` is invoked.

I will come back to this more formally in the following: for now I just show how this is to be done within the framework of **plm** (the results are the same one would obtain using the Stata option **robust** to the **xtreg** command). It is enough to specify the **vcov** argument to **vcovHC** (for “heteroskedasticity-consistent”), thus imposing to use a robust estimator in place of the standard one:

```
> ## 10.4.2, Ex. 10.4 with robust variance
> summary(re10.4, vcov=vcovHC)
```

```
Oneway (individual) effect Random Effect Model
(Swamy-Arora's transformation)
```

Note: Coefficient variance-covariance matrix supplied: **vcovHC**

```
Call:
plm(formula = fm, data = jtrain, model = "random", index = c("fcode",
"year"))
```

Balanced Panel: n = 54, T = 3, N = 162

Effects:

```
          var std.dev share
idiosyncratic 0.2477  0.4977 0.114
individual    1.9322  1.3900 0.886
theta: 0.7975
```

Residuals:

```
      Min.   1st Qu.   Median   3rd Qu.   Max.
-2.546798 -0.223892  0.040554  0.255287  1.549791
```

Coefficients:

```
          Estimate Std. Error t-value Pr(>|t|)
(Intercept)  0.414833  0.260766  1.5908  0.11367
d88          -0.093452  0.091489 -1.0215  0.30862
d89          -0.269834  0.183842 -1.4677  0.14419
union         0.547802  0.392385  1.3961  0.16467
grant        -0.214696  0.127866 -1.6791  0.09514
grant_1      -0.377070  0.260807 -1.4458  0.15024
```

Signif. codes: 0

Again, I leave this to intuition for now: but the Wald test (in χ^2 form)¹ can be robustified in the very same way as the t -statistics, specifying to employ a robust covariance matrix in the calculations.

```
> waldtest(re10.4, 4:5, vcov=vcovHC)
```

¹If Assumption RE.3 is violated, the sum of squared residuals form of the F statistic is not valid (Wooldridge, 2010, p.263).

Wald test

```
Model 1: log(scrap) ~ d88 + d89 + union + grant + grant_1
Model 2: log(scrap) ~ d88 + d89 + union
      Res.Df Df  Chisq Pr(>Chisq)
1         156
2         158 -2  2.8215      0.244
```

2.1.2 Testing for unobserved effects

The unobserved effects test from 10.4.4 is a distribution-free procedure which tests whether the off-diagonal elements in the sub-matrix Σ_i of the overall errors covariance matrix Σ are zero (i.e., whether there is any correlation between errors pertaining to the same individual at different times).

```
> ## 10.4.4 Testing for the Presence of an Unobserved Effect
> pwtest(fm, jtrain, index=c("fcode", "year"))
```

Wooldridge's test for unobserved individual effects

```
data: formula
z = 4.1451, p-value = 3.396e-05
alternative hypothesis: unobserved effect
```

The test confirms the presence of unobserved effects. It must be stressed that this piece of evidence does not imply that the effects be necessarily of the RE type, as the test does as well have power against, say, time-decaying serial error correlation. In turn, any of these deviations from error sphericity is consistent with the robust inference methods presented here.

2.2 Fixed effects

Below, I estimate the FE model from Example 10.5 (FE Estimation of the Effects of Job Training Grants) by specifying the `model` argument to "within" (another name for the FE estimator, due to the fact that it only uses within-individual variation). Notice that this is the default choice for the `model` argument, so that I might have omitted it altogether.

```
> ## Example 10.5 (FE Estimation of the Effects of Job Training Grants)
>
> ## FE model
> fe10.5 <- plm(fm, jtrain, index=c("fcode", "year"), model="within")
```

The Wald test is performed just as above. The model object `fe10.5` contains all the necessary information. This is in a sense similar to what happens with Stata's *post estimation* commands, only here the model object will persist in the workspace for later use (again, this is analogous to using `estimates store` in Stata).

```
> ## Wald test for joint significance of grant and grant_1,
> ## Chi2 or F version
> wald10.5 <- waldtest(fe10.5, 4:5)
> wald10.5.F <- waldtest(fe10.5, 4:5, test="F")
```


2.2.1 LSDV

The same estimate as $\hat{\beta}_{FE}$ can be obtained by explicitly augmenting the original regression with individual dummies, which is called the *least squares dummy variable* estimator (LSDV). It is very easy to do this augmentation because R will include any qualitative variable in a linear model under form of a set of dummies, therefore one just has to include the individual index in the model formula. Notice, in the following syntax, the use of `update` for easily adding the set of dummies to the formula; and the `as.factor` transformation, which makes sure that the individual index is treated as a categorical variable. In fact, if the index were a set of labels, like e.g. country names, there would be no need to do this; but if it is a numeric, as is the `fcode` here, I must specify lest R understands it as just another numerical variable. Moreover, notice that while I insert a full set of dummies I specify not to include an intercept in the model (`-1` in the formula); this is cleaner but not strictly necessary, as otherwise one of the dummies would be automatically dropped.

```
> ## Dummy variable regression
> lsdv10.5 <- plm(update(fm, . ~ . + as.factor(fcode) - 1),
+               jtrain, index=c("fcode", "year"), model="pooling")
```

I can now check that the estimates from FE and LSDV are actually the same:

```
> coef(fe10.5)[c("grant", "grant_1")]
      grant      grant_1
-0.2523149 -0.4215895

> coef(lsdv10.5)[c("grant", "grant_1")]
      grant      grant_1
-0.2523149 -0.4215895
```

The LSDV estimator automatically provides estimates of the individual effects, for as good as they can be (in a short panel, being T -consistent, their estimates are not dependable). In the FE case, Stata produces an average intercept (whose interpretation is not straightforward); `plm` does not. If one wants to recover the estimated individual effects, anyway, there is an extractor function `fixef` available: so that the “overall intercept”, for what it is worth, would simply be `mean(fixef(fe10.5))`. Rather than this, though, as the model postulates a different intercept for each individual, I will look at the population of these.

It can be interesting, in particular, to summarize or plot the distribution of the individual effects “to get some idea of how much heterogeneity is in the population” (Wooldridge, 2010, p. 273-274). For reasons related to the object orientation of R – which go beyond the scope of the present chapter, but are detailed in Croissant and Millo (2019) – to do so requires transforming the recovered fixed effects to numeric: else instead of the usual `summary`, producing range, quartiles and the mean, a significance table like that of the coefficients would be produced.

```

> ## Recover c_i from FE estimation
> c.i <- fixef(fe10.5)
> summary(as.numeric(c.i))

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.9002 -0.2571  0.6103  0.5974  1.7132  3.3144

> ## else summary(c.i) displays a significance table

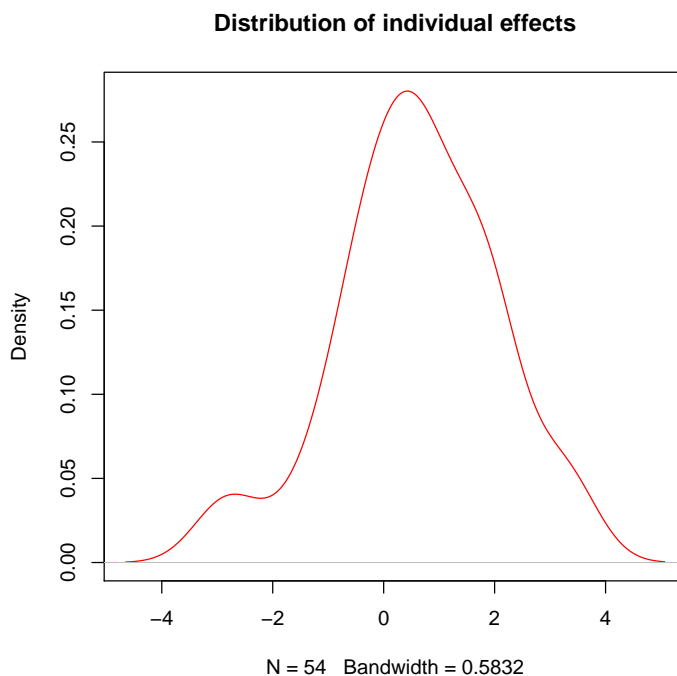
```

Next, I plot the empirical distribution of the individual effects, smoothed out with a kernel-based smoother:

```

> plot(density(c.i), col="red",
+      main="Distribution of individual effects")

```



2.2.2 Serial correlation in the FE model

In his book, Wooldridge (2010, p. 275) proposed a test for serial correlation in FE panel models based on taking into account the serial correlation induced in the model errors by the time-demeaning procedure, so that if the original model's errors are uncorrelated, the transformed errors are correlated with a coefficient of $-\frac{1}{T-1}$, and this is the hypothesis that can actually be tested on the observed residuals. The procedure he suggests is implemented in the R function `pwartest`:

```

> ## Test for serial error correlation
> pwartest(fm, jtrain, index=c("fcode", "year"))

```

Wooldridge's test for serial correlation in FE panels

```
data: plm.model
F = 52.745, df1 = 1, df2 = 106, p-value = 6.667e-11
alternative hypothesis: serial correlation
```

The hypothesis of no serial correlation in the idiosyncratic errors of the original model is strongly rejected: therefore it is appropriate to allow for serial correlation when estimating the standard errors of the estimates. This can be done, as above in the RE case, by specifying a robust estimator for the parameters' covariance through the `vcov` argument to either `summary` or `coeftest`. Below, I extract only the standard errors from the summaries, presenting them just like in the continuation of Example 10.5, Wooldridge (2010, p. 276):

```
> ## FE 10.5 without or with robust covariance (report only SEs)
> rbind(
+   round(coef(fe10.5), 3),
+   round(summary(fe10.5)$coef[,2], 3),
+   round(summary(fe10.5, vcov=vcovHC)$coef[,2], 3)
+ )

      d88      d89  grant grant_1
[1,] -0.080 -0.247 -0.252 -0.422
[2,]  0.109  0.133  0.151  0.210
[3,]  0.096  0.193  0.140  0.276
```

As observed in the book, the change in the SEs from allowing for robustness is unpredictable: the estimated confidence interval on *grant* becomes narrower, that on *grant_1* wider.

The Wald test (in χ^2 form) can be robustified in the same way:

```
> ## Wald test for joint significance of grant and grant_1,
> ## Chisq version (F version would be inappropriate here)
> ## wald10.5.hc <-
> waldtest(fe10.5, 4:5, vcov=vcovHC)
```

Wald test

```
Model 1: log(scrap) ~ d88 + d89 + union + grant + grant_1
Model 2: log(scrap) ~ d88 + d89 + union
  Res.Df Df  Chisq Pr(>Chisq)
1     104
2     106 -2  3.2585    0.1961
```

2.3 First Difference

The *first difference* (FD) estimator consists in applying OLS to the first-differenced data (Wooldridge, 2010, 10.6). This is another way, next to FE, of eliminating unobserved, correlated heterogeneity. The FD estimator is implemented in the `plm` package as the `model='fd'` argument to the `plm` function, by which the data are automatically transformed, just as happens for FE. Sometimes, though,

one does not want to difference some particular variable, such as in this case the time dummies; or in general she wants a more flexible specification. In order to replicate Example 10.6 (FD Estimation of the Effects of Job Training Grants) exactly, instead of differencing at the model level I do so at the formula level: i.e., I specify a formula in differences – model formulae in R, and particularly in **plm**, do support a variety of transformation functions – and apply the pooled OLS estimator to the explicitly transformed data, leaving the 1988 dummy *d88* out (because I lose the first time period in the transformation) and leaving the 1989 dummy *d89* as it is:

```
> ## Example 10.6: First difference estimation
> ## exact formulation as in Wooldridge p.282
> dfm <- diff(log(scrap)) ~ d89 + diff(grant) + diff(grant_1)
> fd10.6 <- plm(dfm, jtrain,
+             index=c("fcode", "year"), model="pooling")
```

The SEs from the FD estimator can be robustified in the usual way. Next, I report the coefficients, the classic and robust SEs as done in Wooldridge:

```
> rbind(
+   round(coef(fd10.6), 3),
+   round(summary(fd10.6)$coef[,2], 3),
+   round(summary(fd10.6, vcov=vcovHC)$coef[,2], 3)
+ )
```

	(Intercept)	d89	diff(grant)	diff(grant_1)
[1,]	-0.091	-0.096	-0.223	-0.351
[2,]	0.091	0.125	0.131	0.235
[3,]	0.088	0.111	0.129	0.265

The R^2 will be part of the usual model `summary`, and will be printed out accordingly; or can be recovered specifically with

```
> summary(fd10.6)$r.squared

      rsq      adjrsq
0.036517701 0.008724942
```

The Wald test can be performed as usual (here, in F form):

```
> waldtest(fd10.6, 2:3, test="F")
```

Wald test

```
Model 1: diff(log(scrap)) ~ d89 + diff(grant) + diff(grant_1)
Model 2: diff(log(scrap)) ~ d89
  Res.Df Df    F Pr(>F)
1     104
2     106 -2  1.5295 0.2215
```

2.3.1 Serial correlation in the FD model

The FD transformation, just as happens with FE, will modify the serial correlation properties of the model errors, inducing a serial correlation with coefficient -0.5 . In this case (Wooldridge, 2010, 10.6.3), one could be interested either in testing for incorrelation of the original errors – to test the model specification – or of the differenced errors – to test Assumption FD.3. See also below, Comparison between FD and FE. Both versions of the Wooldridge first-difference test are implemented in the `pwfdtest` function, where an argument `h0`, defaulting to `"fd"`, allows to test either the original incorrelation hypothesis (on the untransformed model errors) or the derived hypothesis (on the differenced ones). For now, following Example 10.6 (cont.d), I test the differenced errors.

```
> ## Example 10.6 (continued): testing for serial correlation
> pwfdtest(fm, jtrain, index=c("fcode","year"))
```

Wooldridge's first-difference test for serial correlation in panels

```
data: plm.model
F = 2.8015, df1 = 1, df2 = 52, p-value = 0.1002
alternative hypothesis: serial correlation in differenced errors
```

As per Wooldridge (2010, p. 283), "a finding of significant serial correlation in the [transformed errors] warrants computing the robust variance matrix for the FD estimator". Although this is not the case here, this can be done the usual way and I report it for the sake of completeness in the same form as above, comparing classical and robust standard errors.

```
> ## FD 10.6 without or with robust covariance (coefs and SEs)
> rbind(
+   round(coef(fd10.6), 3),
+   round(summary(fd10.6)$coef[,2], 3),
+   round(summary(fd10.6, vcov=vcovHC)$coef[,2], 3)
+ )
```

	(Intercept)	d89	diff(grant)	diff(grant_1)
[1,]	-0.091	-0.096	-0.223	-0.351
[2,]	0.091	0.125	0.131	0.235
[3,]	0.088	0.111	0.129	0.265

2.4 Comparison between FD and FE

Given that both FE and FD can effectively deal with correlated unobserved heterogeneity, the question arises which one to prefer in a given setting. Wooldridge (2010, 10.7.1) addresses the subject from the point of view of efficiency: if the original models' errors follow a random walk, then the first-differenced errors will be stationary and FD will be more efficient than FE; if they are instead stationary, then the preferred choice will be FE because it induces less correlation than FD into the transformed errors.

Another possible concern is the lack of strict exogeneity. Correlation between the errors and past values of the regressors can be solved by including

(backward) lags of the explanatory variables in a distributed lags specification (Wooldridge, 2010, p. 284): in this respect, remember that **plm** supports panel operators in formulae, so that using the `lag` method will work consistently with the panel structure (e.g., the initial observation *for each individual* will be NA). An example follows below. Technically, the same is possible with *forward* lags (a forward lag is obtained setting the argument to a negative value, e.g. $k = -1$); although Wooldridge observes that such a model seldom has a sound economic interpretation.

Testing for strict exogeneity can be done with auxiliary regressions, obtained (in the FD case) augmenting the “explicit FD” formula with the untransformed regressors from the original model, dummies excluded:

```
> ## testing strict exogeneity for FD using a regression-based test
> auxmod.h.fd <- plm(update(dfm, .~. + grant + grant_1),
+                   jtrain, model="pooling")
```

and testing for joint significance of the added variables.

```
> waldtest(auxmod.h.fd, 4:5, test="F")
```

Wald test

```
Model 1: diff(log(scrap)) ~ d89 + diff(grant) + diff(grant_1) + grant +
      grant_1
Model 2: diff(log(scrap)) ~ d89 + diff(grant) + diff(grant_1)
      Res.Df Df      F Pr(>F)
1         45
2         47 -2 0.5413 0.5857
```

A similar procedure is suggested for testing strict exogeneity in the FE setting. Now the model is augmented by forward lags and estimated by FE:

```
> ## testing strict exogeneity for FE using a regression-based test
> auxmod.h.fe <- plm(update(fm, .~. + lag(grant, -1)
+                   + lag(grant_1, -1)),
+                   jtrain, model="within")
> waldtest(auxmod.h.fe, 4:5, test="F")
```

Wald test

```
Model 1: log(scrap) ~ d88 + d89 + union + grant + grant_1 + lag(grant,
-1) + lag(grant_1, -1)
Model 2: log(scrap) ~ d88 + d89 + union + lag(grant, -1) + lag(grant_1,
-1)
      Res.Df Df      F Pr(>F)
1         154
2         156 -2 0.7104 0.493
```

2.5 The relationship between RE and FE

Wooldridge (2010, 10.7.2) shows how the RE estimator can be obtained by applying OLS on *quasi*-time demeaned data, i.e. subtracting only a fraction λ

of the time mean from each individual's data; so that in this framework, $\lambda = 1$ yields the FE estimator (and $\lambda = 0$, i.e. no demeaning, obviously yields OLS).

Below, I illustrate this point recovering $\hat{\lambda}$ from the estimated RE model, both directly and through the estimates of the two error variance components $\hat{\sigma}_c^2$ and $\hat{\sigma}_u^2$, which demonstrates the calculation in Wooldridge (2010, Eq. 10.77):

```
> ## Example 10.7 (Job Training Grants):
> ## extract the quasi-time demeaning parameter from the RE estimates
> ## 1) demonstrate calculations in Wooldridge, p.287
> sigma2.u <- re10.4$ercomp$sigma2[1]
> sigma2.c <- re10.4$ercomp$sigma2[2]
> t. <- length(unique(attr(re10.4$model, "index")[[2]]))
> 1-sqrt(1/(1+t.*(sigma2.c/sigma2.u)))

      id
0.7975426

> ## 2) directly:
> re10.4$ercomp$theta

      id
0.7975426
```

For the sake of illustration, in Table 1 I also report a comparison between OLS, RE and FE which shows how $\hat{\lambda}$ close to 1 produces RE estimates closer to FE than to pooled OLS. To this end, I estimate the pooled OLS specification setting the model argument to 'pooling'.

```
> ols10 <- plm(fm, jtrain, index=c("fcode", "year"), model="pooling")
> texreg(list("OLS"=ols10, "RE"=re10.4, "FE"=fe10.5),
+         caption = "Comparison of estimators", label="tab:modelcomp")
```

2.5.1 Testing serial correlation in RE errors

As per Wooldridge (2010, p. 288), the random effects hypothesis by which, after subtracting the random effects, the remainder errors are incorrelated (RE.3a) can be tested by using the residuals from the regression on partially demeaned data and applying, for example, the Breusch (1978); Godfrey (1978) procedure. This is automated in **plm** through the **pbgttest** wrapper function (see also Croissant and Millo, 2008, 2019):

```
> pbgttest(re10.4)

Breusch-Godfrey/Wooldridge test for serial correlation in panel models

data: fm
chisq = 3.5853, df = 3, p-value = 0.3099
alternative hypothesis: serial correlation in idiosyncratic errors
```

No evidence of serial correlation is found in the remainder errors.

	OLS	RE	FE
(Intercept)	0.42 (0.22)	0.41 (0.24)	
d88	-0.24 (0.31)	-0.09 (0.11)	-0.08 (0.11)
d89	-0.47 (0.33)	-0.27* (0.13)	-0.25 (0.13)
union	0.54* (0.25)	0.55 (0.41)	
grant	0.21 (0.33)	-0.21 (0.15)	-0.25 (0.15)
grant_1	-0.02 (0.43)	-0.38 (0.21)	-0.42* (0.21)
R ²	0.05	0.15	0.20
Adj. R ²	0.02	0.12	-0.24
Num. obs.	162	162	162
s_idios		0.50	
s_id		1.39	

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

Table 1: Comparison of estimators

2.6 Comparison between FE and RE: the Hausman test

The Hausman test is a popular procedure for testing assumption RE.1b (incorrelation between regressors and individual effects) (Wooldridge, 2010, 10.7.3). Rejection will favour a fixed effects analysis, while non-rejection is usually interpreted as evidence in favour of the RE specification.²

The traditional form of the Hausman test is computed in `plm` by the function `phptest`, which can take as arguments either `formula` and `data` or two estimated model objects. I employ this latter syntax to compare the RE and FE models:

```
> ## "traditional" Hausman test
> phptest(fe10.5, re10.4)

Hausman Test

data: fm
chisq = 2.8308, df = 4, p-value = 0.5865
alternative hypothesis: one model is inconsistent
```

It is easy to construct the Hausman test statistic from Wooldridge (2010, Eq. 10.78) “by hand” in plain R (remembering to exclude both the intercept and the time-invariant `union` from `re10.4`):

```
> b.re <- coef(re10.4)[-c(1, 4)]
> b.fe <- coef(fe10.5)
```

²As Wooldridge observes (p. 289), RE.3 is generally assumed as well under the null; but while this latter hypothesis (which implies that RE is the efficient estimator) is necessary for the correct distribution of the Hausman statistic in the traditional form, the latter has no power against failures of RE.3.


```

> V.re <- vcov(re10.4)[-c(1, 4),-c(1, 4) ]
> V.fe <- vcov(fe10.5)
> q <- b.fe - b.re
> Vq <- V.fe - V.re
> H <- crossprod(q, solve(Vq, q))
> 1 - pchisq(abs(H), df=length(q))

      [,1]
[1,] 0.5865327

```

In this case, the test on one single parameter (or a subset of parameters) only can be performed by restricting q and Vq to the relevant row(s) and column(s). For example, below I do a partial Hausman test of *grant*:

```

> H.part <- crossprod(q[3], solve(Vq[3, 3], q[3]))
> 1 - pchisq(abs(H.part), df=1)

      [,1]
[1,] 0.196689

```

The regression-based version of the Hausman test from Wooldridge (2010, Eq. 10.79) is based on augmenting the regression on quasi-time demeaned data (the one from which the RE estimates were obtained) with the within (i.e., time-demeaned) transformations of the variables to test for. The F-test on the added variables is the relevant Hausman statistic.

This procedure can in turn be reproduced easily through the data transformation infrastructure of **plm**. In fact, the time-demeaned and the quasi-time demeaned data needed for Wooldridge (2010, Eq. 10.79) can be retrieved as follows (the reader can check that regressing $y.re$ on $X.re - 1$ will reproduce model *re10.4*); then the Hausman statistic, under form of an F restriction test:

```

> y.re <- pmodel.response(re10.4)
> X.re <- model.matrix(re10.4)
> X.fe <- model.matrix(fe10.5)
> auxmod <- lm(y.re ~ cbind(X.re, X.fe))

> re.ss <- sum(resid(re10.4)^2)
> aux.ss <- sum(resid(auxmod)^2)
> df.auxmod <- fe10.5$df
> m <- dim(X.fe)[[2]]
> aH <- (re.ss-aux.ss)/aux.ss * df.auxmod/m
> pf(aH, df1=m, df2=df.auxmod)

[1] 0.1195879

```

3 Alternative SE estimators for panel data

In the previous Section I have compared inference obtained using either classical standard errors or their most obvious “robust” counterpart: clustering SEs by individual. This is the safest and most obvious choice in the context of the

typical large microeconomic panel, and the default in **plm** for R as well as in other software.

The standard clustering estimator presented above, although arguably the most widely used of covariance estimators at least in the “large panel” contexts typical of microeconometrics, is not the only possible choice. Different estimators have been put forth which are appropriate in some specific context. Despite the practical orientation of the chapter, a minimum of formality will be unavoidable to give an intuition of the relevant alternatives. While referring the interested reader to textbooks, or to the comprehensive computing-oriented treatment in Millo (2017), I sketch the main ideas below here.

Consider a linear model $y = X\beta + u$ and the OLS estimator $\hat{\beta}_{OLS} = (X^\top X)^{-1}X^\top y$. If the error terms u are independent and identically distributed, then the covariance matrix of the OLS estimator takes the following form: $Var(\hat{\beta}) = \sigma^2(X^\top X)^{-1}$, which can be made operational substituting $\sigma^2 = Var(u)$ by an estimate $\hat{\sigma}^2$ of the error variance. This assumption, which often turns out to be too restrictive, will in particular seldom be appropriate in panel data, where errors – especially those belonging to the same individual – can be expected to correlate for a variety of reasons. In the following I will address the two main “robust” approaches to standard error estimation - heteroskedasticity-consistent (HC) errors a la White, and heteroskedasticity- and autocorrelation-consistent (HAC) errors a la Newey-West - without assuming a panel data structure. Then I will review the specific extensions and combinations of the two above methods which have been proposed in the field of panel data.

3.1 Robust estimation in general

“Robust” estimation entails relaxing the assumptions of incorrelation and/or homoskedasticity without imposing any particular structure to the errors’ variance or interdependence. The first breakthrough in this respect is the so-called *sandwich estimator*, due to Halbert White.

The covariance matrix of the OLS estimator is, in general,

$$V(\hat{\beta}_{OLS}) = (X^\top X)^{-1}(X^\top[\sigma^2\Omega]X)(X^\top X)^{-1}$$

where $Var(u) = \sigma^2\Omega$ is the (scaled) error covariance matrix, which is generally unknown. Under homoskedasticity and incorrelation, $\Omega = I_n$ and $Var(u)$ is unknown only up to the variance parameter σ^2 , which is easily estimated from the OLS residuals; but relaxing these hypotheses means allowing for an arbitrary structure where the only requirement is that Ω be positive definite and symmetric, so that in principle it may contain up to $n(n+1)/2$ free parameters.

Yet, as per White (1980), in order to consistently estimate $V_{\hat{\beta}}$ it is not necessary to estimate all the $n(n+1)/2$ unknown elements in the Ω matrix but only the $K(K+1)/2$ ones in

$$X^\top[\sigma^2\Omega]X = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} \mathbf{x}_i \mathbf{x}_j^\top$$

which is often referred to as the *meat* of the sandwich, the two $(X^\top X)^{-1}$ being the *bread*. *Pointwise consistent* estimates of the unobservable errors u

are required, which is satisfied by the OLS residuals \hat{u} by consistency of the estimator $\hat{\beta}_{OLS}$. This structure is still too general but yields feasible estimators in two important special cases.

In the heteroskedasticity case, correlation between different observations is ruled out and the *meat* reduces to

$$S_0 = \sum_{i=1}^n \sigma_i^2 \mathbf{x}_i \mathbf{x}_i^\top$$

where the n unknown σ_i^2 s can be substituted, for estimation purposes, by the squared residuals \hat{u}_i^2 so that $\hat{S}_0 = \sum_{i=1}^n \hat{u}_i^2 \mathbf{x}_i \mathbf{x}_i^\top$. This is the HC estimator of White, which is appropriate in both cross-sectional and time-series settings:

$$\hat{V}_{White} = (X^\top X)^{-1} \hat{S}_0 (X^\top X)^{-1}$$

For the specific case of time series, Newey and West (1987) devise a heteroskedasticity *and autocorrelation* consistent estimator that allows to relax the hypothesis of error independence, and works based on the assumption of correlation dying out "quickly enough" as the distance between observations increases. The Newey-West HAC estimator for the *meat* takes that of White and adds a sum of covariances between the different residuals, smoothed out by a *kernel function* giving weights decreasing with distance:

$$\hat{S}_{Newey-West} = \hat{S}_0 + \sum_{l=1}^L \sum_{t=l+1}^T w_l \hat{u}_t \hat{u}_{t-l} (\mathbf{x}_t \mathbf{x}_{t-l}^\top + \mathbf{x}_{t-l} \mathbf{x}_t^\top)$$

with w_l the weight from the kernel smoother, e.g., the Bartlett kernel function: $w_l = 1 - \frac{l}{L+1}$ (for a discussion of alternative kernels see Zeileis, 2006)). The lag l is usually truncated well below sample size: one popular rule of thumb is $L = n^{1/4}$ (see Greene, 2003; Driscoll and Kraay, 1998). The Newey-West HAC estimator is then:

$$\hat{V}_{Newey-West} = (X^\top X)^{-1} \hat{S}_{Newey-West} (X^\top X)^{-1}$$

3.2 Clustered SEs

This framework has been extended to a panel data setting where, thanks to added dimensionality, various combinations of the two above structures turn out to be able to accommodate very general types of dependence. Besides heteroskedasticity, for example, the added dimensionality allows to obtain robustness against totally unrestricted timewise *or* cross-sectional correlation, provided this is along the "smaller" dimension. I.e., next to heteroskedasticity, wide panels can be robustified against serial correlation, long panels against cross-sectional correlation. In other words, (an appropriate variant of-) the HC estimator can become HAC in a panel context.

In the most common case of a wide panel, the *meat* of the clustering estimator of Liang and Zeger (1986) and Arellano (1987) takes the form:

$$\hat{S}_{ci} = \sum_{i=1}^n X_i^\top \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i^\top X_i$$

where X_i is the part of the design matrix, and $\hat{\mathbf{u}}_i$ that of the residuals' vector, pertaining to the i -th individual; so that the full *sandwich* covariance is:

$$\hat{V}_{cluster,i} = (X^\top X)^{-1} \hat{S}_{ci} (X^\top X)^{-1}$$

It is symmetric (summing over t) in the case of long panels. In wide panels, the big cross-sectional dimension allows robustness against serial correlation (Arellano, 1987); "long" panels, on the converse, can be robustified against cross-sectional correlation thanks to the large T . As a general rule, the estimator is asymptotic in the number of clusters. This is the "robust" covariance used in Section 2, and what most applied econometricians refer to when they report "robust" or "clustered" standard errors in panel data analysis.

3.2.1 Double clustering

In suitably large datasets, *double clustering* can be performed (Petersen, 2009; Cameron et al, 2011; Thompson, 2011) in order to account for serial and at the same time for cross-sectional or spatial correlation. In fact, this estimator combining both individual and time clustering relies on a combination of the asymptotics of each: the minimum number of clusters along the two dimensions must diverge. Any kind of dependence is allowed within each group *or* within each time period, while cross-serial correlations between observations belonging to different groups *and* time periods are ruled out. The double-clustered estimator is calculated by summing up the group-clustering and the time-clustering ones, then subtracting the standard White estimator:

$$\hat{V}_{cluster,it} = \hat{V}_{cluster,i} + \hat{V}_{cluster,t} - \hat{V}_{White}$$

3.3 Newey-West and SCC

As cited above, in a time series context Newey and West (1987) have proposed an estimator that is robust to serial correlation as well as to heteroskedasticity. This estimator takes into account the covariance between units by weighting it through a kernel smoother function giving less weight to more distant terms.

The Newey-West HAC method can be applied time-wise within each individual in the context of a panel dataset. In this case, the resulting estimator will be robust to general heteroskedasticity and to within-individual serial correlation, provided that the latter is of the time-decaying type and that it dies out "fast enough". The *meat* of the NW estimator for the panel case can be expressed as the sum of the "pure" White estimator (applied to the entire dataset) plus a weighted sum over all lags up to l of weighted time-covariance, by individual terms:

$$\hat{S}_{NW} = \hat{S}_0 + \sum_{l=1}^L w_l \left[\sum_{t=1}^T \sum_{i=1}^n \mathbf{x}_{it} \hat{u}_{it} \hat{u}_{i,t-l}^\top \mathbf{x}_{i,t-l}^\top + \sum_{t=1}^T \sum_{i=1}^n [\mathbf{x}_{it} \hat{u}_{it} \hat{u}_{i,t-l}^\top \mathbf{x}_{i,t-l}^\top]^\top \right]$$

so that $\hat{V}_{NW} = (X^\top X)^{-1} \hat{S}_{NW} (X^\top X)^{-1}$. Notice how – unlike the clustering estimator – it does not make any sense to apply the NW estimator symmetrically, over the cross-section, because the latter does not have a natural ordering.

Driscoll and Kraay (1998) have adapted the Newey-West panel time series estimator to include not only serial correlation between errors of the same individual in different time periods but also cross-serial correlation between different

individuals in different times and, within the same period, cross-sectional correlation (see also Arellano, 2003).

The Driscoll and Kraay estimator (SCC for "spatial correlation consistent"), can be calculated as the time-clustering plus a sum of lagged cross-serial covariance terms, again weighted by a distance-decreasing kernel function w_l :

$$\hat{S}_{SCC} = \hat{S}_{ct} + \sum_{l=1}^L w_l \left[\sum_{t=1}^T X_t^\top \hat{\mathbf{u}}_t \hat{\mathbf{u}}_{t-l}^\top X_{t-l} + \sum_{t=1}^T [X_t^\top \hat{\mathbf{u}}_t \hat{\mathbf{u}}_{t-l}^\top X_{t-l}]^\top \right] \quad (1)$$

and, as usual, $\hat{V}_{SCC} = (X^\top X)^{-1} \hat{S}_{SCC} (X^\top X)^{-1}$.

Again, the SCC covariance estimator requires the serial and cross-serial dependence to decrease "fast enough" with the T dimension, which is therefore supposed to be fairly large: Driscoll and Kraay (1998), based on Montecarlo simulation, put the practical minimum at $T > 20 - 25$; the n dimension is irrelevant in this respect and is allowed to grow at any rate relative to T .

3.4 In practice

Summing up, heteroskedasticity can be taken care of in panel data just like one would in cross-sections or time series; but the panel data econometrician will often be concerned about some kind of dependence as well – and, thanks to the richer structure of panel data, she has a number of weapons at her disposal.

If – next to heteroskedasticity – one is worried about serial correlation within individuals of a panel, if the time dimension T is "small" relative to the cross-sectional n she will want to employ the usual clustering estimator $\hat{V}_{cluster,i}$, which allows for arbitrary serial correlation; on the contrary, \hat{V}_{NW} will be the estimator of choice if T is fairly large and one is willing to make the additional assumption that the time dependence is of the distance-decaying type. Both the above exclude any cross-sectional correlation of the errors.³

Symmetrically to the by-individual clustering case, "long" panels of relatively few time series can be made robust to cross-sectional correlation by employing $\hat{V}_{cluster,t}$, provided that any serial correlation is ruled out.⁴ The NW estimator instead has no natural counterpart in this setting because the cross-section does not, in general, have a univariate natural ordering.

Lastly, cross-sectionally dependent panels with potentially large n and a sizeable T (see above, say at least $T > 20 - 25$) can be robustified against heteroskedasticity, unrestricted cross-sectional correlation and time-decaying serial *and cross-serial* correlation by employing \hat{V}_{SCC} , at the only cost of assuming that serial and cross-serial correlations fade out after L lags, with L "much shorter" than T .

³In order to mitigate the cross-sectional correlation problem, the researcher might want to include time dummies in the estimated model so as to control for omitted common time effects. Expanding on this is out of the scope of this chapter; the reader is referred to the literature on panels with common factors and cross-sectional dependence. Neglecting to include time dummies can also induce serial correlation (see Wooldridge, 2010, p.261).

⁴Specifying a distributed lag model is a common solution to residual serial correlation, although the assumptions of the basic panel estimators, in particular FE, can be at odds with dynamic specifications. Again, this is out of the scope of the present chapter: see the literature on dynamic panels.

4 The House Pricing example

In the following I will apply the above described robust standard error estimators in a context where most of the different “flavours” have chances of being meaningful, thanks to the peculiar dimension of the dataset and the fact that the units of observation are sampled in space.

Holly et al (2010) address the evolution of house prices, showing that state-level real housing prices are driven by economic fundamentals, such as real per capita disposable income, as well as by common shocks, such as changes in interest rates, oil prices, and technological change. They employ CCE estimators (Pesaran, 2006) to account for said common factors.

Baltagi and Li (2014) replicate the house pricing model of Holly et al (2010) over a denser network of spatial units: i.e., instead of 49 states over the years 1975 to 2003 as in the original paper, they consider a panel of 381 Metropolitan Statistical Areas (MSAs) observed over the years 1975 to 2011. The model contains three observed regressors: the log of real per capita income, the growth rate of the population and the real cost of borrowing. The log of the real housing price index is the dependent variable.

The original exercises do consider prices and income in levels and assess stationarity and cointegration before proceeding to estimating an error correction specification. Nonstationary data are ill-suited to the standard panel techniques considered here, therefore I consider a short-term version of their model, relating the changes in housing prices to the changes in income (see also Aquaro et al, 2019). I will estimate the model first by pooled OLS, and then accounting for individual and/or time fixed (or random) effects. In particular, time fixed effects are likely to be necessary to account for common shocks, at least as a first approximation with respect to the more sophisticated and flexible approach predominant in the current literature.⁵

4.1 The dataset

The dataset can be found in text format in the Data Archive of the *Journal of Applied Econometrics*, among the materials accompanying the Baltagi and Li (2014) paper, and easily imported into R as follows:

```
> ## read in data
> msa <- read.table(file="msa_4v.txt", header=TRUE)
```

The dataset contains the following variables: the log of the real house price index (*lhpi_real*), the log of real per capita disposable income (*lpci_real*), the population growth rate (*pgr*) and the real cost of borrowing (*rcb*).

I look at the dataset as a panel, so to perform every analysis with **plm**; non-panel estimators, like OLS, are nested subcases and can be easily obtained.

⁵Delving into the possible sources of cross-sectional dependence is out of the scope of the present chapter. In a nutshell: if any common variable drives the outcomes of the entire cross section, period by period, omitting it will - at a minimum - induce cross-sectional correlation in the errors. Time dummies are often enough to capture the influence of unobserved cross-sectional heterogeneity; but, with respect to the CCE estimator, a time fixed effects specification will constrain all factor loadings measuring the elasticity of every individual observation unit to changes in the common factor(s) to be homogeneous; see e.g. Pesaran (2006).

Notice that the dataset already complies with the simplest conventions of the package, having the firm identifier in the first column and the time identifier in the second, and can be employed as it is.⁶ The dataset contains 13716 observations from 381 units over 36 time periods:

```
> pdim(msa)
```

```
Balanced Panel: n = 381, T = 36, N = 13716
```

The model formula is:

```
> blfm <- diff(lhpi_real) ~ diff(lpci_real) + pgr + rcb
```

Notice the seamless treatment of differencing inside the model formula. As already observed, the **plm** package contains a complete data transformation suite of functions performing lagging, differencing and more typical panel data operations like the *within* or *between* transformations in a consistent way.

4.2 Different SEs for the OLS estimates

I will first address estimating a model by OLS and, which is the main focus, obtaining various estimates of the parameter standard errors $SE(\hat{\beta}_{OLS})$ according to the different methods presented above.

Below I estimate the model by OLS and present the results in the typical significance table, using the “classical” standard errors.

```
> ## OLS
> olsmod <- plm(blfm, data = msa, model = "pooling")
> summary(olsmod)
```

Pooling Model

Call:

```
plm(formula = blfm, data = msa, model = "pooling")
```

```
Balanced Panel: n = 381, T = 35, N = 13335
```

Residuals:

	Min.	1st Qu.	Median	3rd Qu.	Max.
	-0.13153230	-0.01337601	0.00052791	0.01367578	0.17130641

Coefficients:

	Estimate	Std. Error	t-value	Pr(> t)
(Intercept)	0.02210665	0.00034344	64.3675	<2e-16 ***
diff(lpci_real)	0.35639153	0.00767517	46.4343	<2e-16 ***
pgr	-0.02245680	0.01581916	-1.4196	0.1557
rcb	-0.79460176	0.00373741	-212.6075	<2e-16 ***

Signif. codes: 0

⁶Otherwise one would have had to specify the `index` argument, a character vector of length 2 containing, in this order, the names of the individual and the time index.

or, equivalently (and not reported) through the `coefstest` function:

```
> olsmod.ols <- coefstest(olsmod)
```

Now I address how to compute t -statistics using different kinds of “robust” standard errors. Only as a first illustration, I present heteroskedasticity-consistent SEs according to White (1980) (without taking into account any panel structure). Notice how the specific covariance estimator to use can be passed on either as a user-specified function or as a precalculated matrix; in either case, function definition or matrix calculation can take place within the very same call to `summary` or to `coefstest`:

```
> ## White, no clustering
> vcovW <- function(x) vcovHC(x, method="white1")
> summary(olsmod, vcov=vcovW)
```

Pooling Model

Note: Coefficient variance-covariance matrix supplied: `vcovW`

Call:

```
plm(formula = blfm, data = msa, model = "pooling")
```

Balanced Panel: n = 381, T = 35, N = 13335

Residuals:

	Min.	1st Qu.	Median	3rd Qu.	Max.
	-0.13153230	-0.01337601	0.00052791	0.01367578	0.17130641

Coefficients:

	Estimate	Std. Error	t-value	Pr(> t)
(Intercept)	0.02210665	0.00043337	51.0110	<2e-16 ***
<i>diff(lpci_real)</i>	0.35639153	0.01117625	31.8883	<2e-16 ***
<i>pgr</i>	-0.02245680	0.01992010	-1.1273	0.2596
<i>rcb</i>	-0.79460176	0.00587330	-135.2904	<2e-16 ***

Signif. codes: 0

or, equivalently, (and not reported)

```
> coefstest(olsmod, vcovHC(olsmod, method="white1"))
```

In the following I review computing all the kinds of *panel-robust* SEs presented in the preceding Section, (almost) in the same order.

4.2.1 Clustered (one dimension)

The default behaviour of the `vcovHC` covariance applied to a panel model is to compute SEs *clustered by individual*:

```
> ## clustered:
> ## by firm
> olsmod.hci <- coefstest(olsmod, vcovHC)
```


but one can of course specify clustering *by time period*⁷:

```
> ## by time
> olsmod.hct <- coefstest(olsmod, vcovHC(olsmod, cluster="time"))
```

4.2.2 Clustered (two dimensions)

A dedicated function, *vcovDC*, will produce double-clustering SEs:

```
> ## double-clustering
> olsmod.dc <- coefstest(olsmod, vcovDC)
```

4.2.3 Newey-West or Driscoll-Kraay

Another “special” covariance function, *vcovNW*, will compute Newey and West (1987) SEs; as usual, see *?vcovNW* for how to set further optional arguments, most importantly the truncation lag.

```
> ## Newey-West
> olsmod.nw <- coefstest(olsmod, vcovNW)
```

The more general case nesting NW, the SCC estimator of Driscoll and Kraay (1998), will be more appropriate here, where spatial correlation between observations and general cross-sectional dependence due to omitted global factors are very likely to be present:

```
> ## Driscoll-Kraay
> olsmod.scc <- coefstest(olsmod, vcovSCC)
```

4.2.4 Comparison of standard errors

In the following Table 2 we compare the SEs from the different estimators (setting the relevant columns from the tables made with *coefstest* side by side).

```
> compare.ols <- cbind(olsmod.ols[,1:2], olsmod.hci[,2],
+ olsmod.hct[,2], olsmod.dc[,2],
+ olsmod.nw[,2], olsmod.scc[,2])
> dimnames(compare.ols)[[2]][2:7] <- c("OLS", "Cl. i", "Cl. t",
+ "Cl. it", "NW", "SCC")
> xtable(compare.ols, digits=3,
+ caption="Comparison of SE estimates, OLS",
+ label="tab:ols.se")
```

The “fit them all” strategy of computing all different SEs and setting them side by side, perhaps choosing the most conservative ones, has issues from the viewpoint of multiple testing (see Millo, 2017, Section 6) if seen as a formal procedure, but has indeed the merit of highlighting the main sources of correlation. In this case, although qualitative considerations on the significance of regressors are little changed, one can see that the specification of the SEs matters a lot for estimating the precision of $\hat{\beta}$.

⁷Strictly speaking, clustering by time would be inappropriate here as N is much larger than T.

	Estimate	OLS	Cl. i	Cl. t	Cl. it	NW	SCC
(Intercept)	0.022	0.000	0.001	0.003	0.003	0.001	0.004
diff(lpci_real)	0.356	0.008	0.020	0.090	0.091	0.014	0.102
pgr	-0.022	0.016	0.024	0.048	0.049	0.023	0.055
rcb	-0.795	0.004	0.010	0.037	0.038	0.007	0.044

Table 2: Comparison of SE estimates, OLS

Let us focus on the main variable of interest, *lpci_real*. The “classical” errors from the pooled specification are the smallest; they double if allowing for clustering, while the increase of the NW SEs is smaller, hinting at some kind of non-fading correlation (as would be the case for an individual effect: take heed we are simply pooling the data here); but the real increase one has if she allows for cross-sectional correlation, the time-clustering SEs being more than tenfold the OLS ones. Lastly, double-clustering SEs are not far from time-clustering ones, while SCC do increase some more: I take this as evidence that allowing for both within-individual persistence and cross-serial correlation, the lagged influence of other units of observation, is the safest course of action.

This is consistent with the economics of the example, as house prices are very much expected to co-move in the cross section and also quite likely to correlate cross-serially; moreover, the omission of individual effects of any kind in this pooled specification means that any persistent individual heterogeneity will “contaminate” the errors, inducing serial correlation.

4.3 Fixed Effects with robust SEs

The previous example employed a pooled specification for the sake of illustration. By contrast, the economic application at hand obviously calls for taking care of unobserved heterogeneity, both along the individual dimension (unobserved, potentially correlated persistent characteristics of individual MSAs, like being part of an industrial district, or containing some important amenity, or being in the middle of a polluted area) and along time (unobserved common factors driving the behaviour of the whole cross section of MSAs, period by period, like the national economic cycle or credit conditions).

As already noted in Section 2, by its characteristics of robustness in a wide array of situations, the individual FE specification is very popular and hence the default for the `plm` function.

```
> ## individual fixed effects:
> femod <- plm(blfm, data = msa) # model="within" is default
> coeftest(femod)
```

```
t test of coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
diff(lpci_real)	0.3497138	0.0075246	46.4763	< 2e-16	***
pgr	0.0426255	0.0220757	1.9309	0.05352	.
rcb	-0.8032376	0.0037106	-216.4737	< 2e-16	***

```
---
```

```
Signif. codes: 0
```

4.3.1 Individual fixed effects with robust SEs

Robust SEs produced by any `vcovXX` function – e.g., clustered SEs – can be combined with most of the estimators in **plm**. Next to the obvious pooled OLS case: FE, random effects (RE), first difference (FD) and more. Just like in the OLS case, one specifies the estimated model and the covariance estimation method of choice, e.g. clustering by individual:

```
> ## with clustering
> coeftest(femod, vcovHC)

t test of coefficients:

              Estimate Std. Error   t value Pr(>|t|)
diff(lpci_real)  0.349714   0.019332  18.0901 <2e-16 ***
pgr              0.042625   0.039552   1.0777  0.2812
rcb             -0.803238   0.007454 -107.7591 <2e-16 ***
---
Signif. codes:  0
```

and any of the above, just substituting `femod` for `olsmod`: e.g., SCC with individual fixed effects

```
> coeftest(femod, vcovSCC)

t test of coefficients:

              Estimate Std. Error   t value Pr(>|t|)
diff(lpci_real)  0.349714   0.101276   3.4531  0.000556 ***
pgr              0.042625   0.116253   0.3667  0.713877
rcb             -0.803238   0.047750 -16.8217 < 2.2e-16 ***
---
Signif. codes:  0
```

It shall of course be noted, first and foremost, that including (at least) individual FEs in the specification is likely to be necessary for consistency of $\hat{\beta}$. From the point of view of precision, it can be seen that including individual FEs in the specification does very slightly reduce the clustering SE, while the SCC SE remains almost unchanged. In other words, including individual FEs does not “clean away” most of the within-individual error correlation.

4.3.2 Two-way fixed effects with robust SEs: comparison

As observed, in the example at hand it is crucial to control for time FEs as well, in order to account for unobservable common shocks affecting every cross-sectional unit (the MSA) over time. Therefore, the two-ways fixed effects (2FE) – rather than FE – is likely to be the appropriate specification. Combining this with the different SE estimators, as above, yields the following Table 3:

```
> fe2mod <- plm(blfm, data = msa, effect="twoways")
> ## "classical" SEs
> fe2mod.ols <- coeftest(fe2mod)
```

```

> ## cluster by individual
> fe2mod.hci <- coeftest(fe2mod, vcovHC)
> ## by time
> fe2mod.hct <- coeftest(fe2mod, vcovHC(fe2mod, cluster="time"))
> ## double-clustering
> fe2mod.dc <- coeftest(fe2mod, vcovDC)
> ## Newey-West
> fe2mod.nw <- coeftest(fe2mod, vcovNW)
> ## Driscoll-Kraay
> fe2mod.scc <- coeftest(fe2mod, vcovSCC)

> compare.fe2 <- cbind(fe2mod.ols[,1:2], fe2mod.hci[,2],
+                      fe2mod.hct[,2], fe2mod.dc[,2],
+                      fe2mod.nw[,2], fe2mod.scc[,2])
> dimnames(compare.fe2)[[2]][2:7] <- c("OLS", "Cl. i", "Cl. t",
+                                     "Cl. it", "NW", "SCC")
> xtable(compare.fe2, digits=3,
+         caption="Comparison of SE estimates, 2FE",
+         label="tab:fe2.se")

```

	Estimate	OLS	Cl. i	Cl. t	Cl. it	NW	SCC
diff(lpci.real)	0.160	0.005	0.012	0.021	0.024	0.008	0.030
pgr	-0.018	0.012	0.024	0.026	0.032	0.016	0.030
rcb	-0.927	0.002	0.009	0.009	0.012	0.005	0.012

Table 3: Comparison of SE estimates, 2FE

One can see that, despite the qualitative considerations of the preceding Section 4.2.4 still being widely applicable, the cross-sectional dependence has been substantially reduced: the double-clustering SEs now being only double the individual clustering ones. Analogously, the SCC SEs are still the largest – at six times the OLS ones – but now are one third of the SCC SEs for the pooling estimates.

4.3.3 Other panel estimators combined with clustering or NW/SCC

As observed, any specification among: pooled, FE or RE with individual, time or both kinds of effects, or FD; can be combined with any SE estimator in: one (individual or time) or two-way clustering, NW, SCC and some more combinations (see Millo, 2017).

Another potentially meaningful example among the many possible is: looking for efficiency combining random effects along the individual dimension (i.e., treating the many MSAs as random samples from a larger population) with fixed time effects (RE+TFE). (Notice that the output of `coeftest` is limited to the first 4 rows to avoid reporting all the estimated year dummies.)

```

> rtfemod <- plm(update(blfm, . ~ . + as.factor(year)), data=msa,
+                model="random")
> ## e.g., RE+TFE with SCC
> coeftest(rtfemod, vcovSCC)[1:4,]

```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.004486737	0.000583317	7.6917649	1.553023e-14
diff(lpci_real)	0.161429689	0.032593391	4.9528350	7.403828e-07
pgr	-0.026897148	0.030574309	-0.8797304	3.790213e-01
rcb	-0.924707300	0.011407218	-81.0633489	0.000000e+00

The latter specification can be compared with a more standard two-way fixed effects (2FE) by means of the Hausman test. The time FEs, explicitly added to the model formula, are not called into question; the test will instead compare the choice of random vs. fixed for the individual effects:

```
> phtest(update(blfm, . ~ . + as.factor(year)), data = msa)
```

Hausman Test

```
data: update(blfm, . ~ . + as.factor(year))
chisq = 101.51, df = 37, p-value = 6.391e-08
alternative hypothesis: one model is inconsistent
```

Despite the apparent similarity of the RE+TFE and 2FE estimates, the test rejects the consistency of the former. 2FE is preferred.

5 Conclusions

I have reviewed the implementation of the basic panel data estimators and tests in the R language, with a particular focus on estimating the standard errors according to various procedures, each of which is “robust” to general heteroskedasticity *and* to some kind of violation of the incorrelation hypothesis.

The **plm** package provides a comprehensive set of tools that allows syntactical consistency and flexibility in combining estimators for the β s and for the SEs thereof at will, thanks to the feature of R that functions are a data type. The latter in fact allows to pass on a statistical procedure (here: the standard error estimator) to any test.

In particular, one- or two-way fixed effects specifications are especially useful in panel data econometrics because they can take into account unobserved individual or timewise heterogeneity, potentially correlated with the regressors, which would render the estimates inconsistent. A fixed (or a random) effects specification can be easily combined with the most popular “clustering” estimators or with variations of the Newey-West procedure: most notably, the “spatial correlation consistent” (SCC) standard errors of Driscoll and Kraay (1998).

In the first part of the paper I have provided a gentle introduction to the subject of panel data econometrics with R by going through the “Basic panel methods” chapter of Wooldridge (2010) and replicating all its examples with functionality from **plm**, in particular showing how to perform testing employing clustered standard errors – which are the most obvious choice for the applied econometrician working on the typical micropanel: many individual units observed over a few time periods. The reader needing a basic primer on robust methods in panel data, perhaps as an R companion to Wooldridge (2010, Ch. 10), can stop here.

In the second half, I have presented a number of different estimators for the standard errors in the light of the two main procedures for heteroskedasticity-

and heteroskedasticity-and-autocorrelation-robust estimation: the HC estimator of White and the HAC estimator of Newey and West; and of their modifications developed for panel data: one- or two-way clustering, and adaptations of the kernel-smoothing-based procedure to panel data. I have illustrated the different procedures through an example drawing on a dataset with a longer time dimension, whereby a number of different estimators for the standard errors become viable. I have combined one- or two-way fixed effects, or combinations of fixed and random effects, with any estimator of the standard errors presented here, showing how the dispersion of estimates can be crucially dependent on the correlation within individuals, within each cross-section or even across different individuals in time; and how making the wrong assumptions can lead to severely overstating the precision of estimates.

References

- Aquaro M, Bailey N, Pesaran MH (2019) Estimation and inference for spatial models with heterogeneous coefficients: an application to us house prices. *Journal of Applied Econometrics* Early View(forthcoming)
- Arellano M (1987) Computing robust standard errors for within group estimators. *Oxford Bulletin of Economics and Statistics* 49:431–434
- Arellano M (2003) *Panel Data Econometrics*. Oxford University Press
- Baltagi BH, Li J (2014) Further evidence on the spatio-temporal model of house prices in the united states. *Journal of Applied Econometrics* 29(3):515–522
- Breusch TS (1978) Testing for autocorrelation in dynamic linear models. *Australian Economic Papers* 17(31):334–355
- Cameron A, Gelbach J, Miller D (2011) Robust inference with multiway clustering. *Journal of Business and Economic Statistics* 29(2):238–249
- Croissant Y, Millo G (2008) Panel data econometrics in r: The 'plm' package. *Journal of Statistical Software* 27(2), URL <http://www.jstatsoft.org/v27/i02/>
- Croissant Y, Millo G (2019) *Panel Data Econometrics with R*. Wiley Online Library
- Dahl DB, Scott D, Roosen C, Magnusson A, Swinton J (2019) xtable: Export Tables to LaTeX or HTML. URL <https://CRAN.R-project.org/package=xtable>, r package version 1.8-4
- Driscoll J, Kraay A (1998) Consistent covariance matrix estimation with spatially dependent panel data. *The Review of Economics and Statistics* 80(4):549–560
- Fox J, Weisberg S (2011) *An R Companion to Applied Regression*, 2nd edn. Sage, Thousand Oaks CA, URL <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion>
- Godfrey LG (1978) Testing against general autoregressive and moving average error models when the regressors include lagged dependent variables. *Econometrica: Journal of the Econometric Society* pp 1293–1301

- Greene W (2003) *Econometric Analysis*, 5th edn. Prentice Hall
- Henningsen A, Henningsen G (2019) Analysis of panel data using r. In: *Panel Data Econometrics*, Elsevier, pp 345–396
- Holly S, Pesaran MH, Yamagata T (2010) A spatio-temporal model of house prices in the usa. *Journal of Econometrics* 158(1):160–173
- Leifeld P (2013) texreg: Conversion of statistical model output in R to \LaTeX and HTML tables. *Journal of Statistical Software* 55(8):1–24, URL <http://dx.doi.org/10.18637/jss.v055.i08>
- Leisch F (2002) Sweave: Dynamic generation of statistical reports using literate data analysis. In: *Compstat*, Springer, pp 575–580
- Liang K, Zeger S (1986) Longitudinal data analysis using generalized linear models. *Biometrika* 73:13–22
- Millo G (2017) Robust standard error estimators for panel models: A unifying approach. *Journal of Statistical Software* 82(1):1–27
- Newey W, West K (1987) A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55(3):703–708
- Pesaran M (2006) Estimation and inference in large heterogeneous panels with a multifactor error structure. *Econometrica* 74(4):967–1012
- Petersen MA (2009) Estimating standard errors in finance panel data sets: Comparing approaches. *Review of Financial Studies* 22(1):435–480
- Shea JM (2018) wooldridge: 111 Data Sets from "Introductory Econometrics: A Modern Approach, 6e" by Jeffrey M. Wooldridge. URL <https://CRAN.R-project.org/package=wooldridge>, r package version 1.3.1
- Thompson S (2011) Simple formulas for standard errors that cluster by both firm and time. *Journal of Financial Economics* 99(1):1–10
- White H (1980) *Asymptotic Theory for Econometricians*. Academic Press, Orlando
- Wooldridge JM (2010) *Econometric analysis of cross section and panel data*. MIT press
- Zeileis A (2004) Econometric computing with hc and hac covariance matrix estimators. *Journal of Statistical Software* 11(10):1–17, URL <http://www.jstatsoft.org/v11/i10/>
- Zeileis A (2006) Object-oriented computation of sandwich estimators. *Journal of Statistical Software* 16(9):1–16, URL <http://www.jstatsoft.org/v16/i09/>
- Zeileis A, Hothorn T (2002) Diagnostic checking in regression relationships. *R News* 2(3):7–10, URL <http://CRAN.R-project.org/doc/Rnews/>