Econometrics 3, Problems 4

Oliver Snellman

07.05.2020

Contents

Introduction: Cross sectional, panel and time series structures **1**

Panel data **2**

Panel data in matrix form 4

Models for panel data **6**

How to choose a suitable model for panel data? 6

OLS 7

Random Effects Estimator 10

Fixed Effects Estimator 12

First-Difference Estimator 14

Endogeneity 15

Anderson-Hsiao Estimator 15

Arellano-Bond Estimator 16

Monte Carlo study **17**

The data simulation function 17

The MC-algorithm 18

Results: Average estimated parameter values 21

Results: Distribution of estimated parameters 22

Autocorrelation of the residuals η_{it}^{k} 24

Increasing the time dimension 25

Introduction: Cross sectional, panel and time series structures

Assume that the following data inherently has a panel structure, the explained variable can be thought of as the first of the K variables. Let's take a look at how the three main representations (CS, panel and TS) interpretes this data.

$$\begin{array}{ccc} \text{Cross Sectional} & \text{Panel} & \text{Time Series} \\ y_i = x_i'\beta + \varepsilon_i^{CS} & y_{it} = x_{it}'\beta + f_i + u_{it} & y_t = x_t'\beta + \varepsilon_t^{TS} \\ x_{ik} & x_{itk} & x_{tk} \end{array}$$

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ \vdots & \vdots & & \vdots \\ x_{T1} & x_{T2} & \cdots & x_{TK} \\ x_{(T+1)1} & x_{(T+1)2} & \cdots & x_{(T+1)K} \\ \vdots & \vdots & & \vdots \\ x_{(2\times T)1} & x_{(2\times T)2} & \cdots & x_{(2\times T)K} \\ \vdots & \vdots & & \vdots \\ x_{[(N-1)\times T]1} & x_{[(N-1)\times T]2} & \cdots & x_{[(N-1)\times T]K} \\ \vdots & \vdots & & \vdots \\ x_{(N\times T)1} & x_{(N\times T)2} & \cdots & x_{(N\times T)K} \end{bmatrix} \begin{bmatrix} x_{111} & x_{112} & \cdots & x_{11K} \\ \vdots & \vdots & & \vdots \\ x_{1T1} & x_{1T2} & \cdots & x_{1TK} \\ x_{211} & x_{212} & \cdots & x_{21K} \\ \vdots & \vdots & & \vdots \\ x_{2T1} & x_{2T2} & \cdots & x_{2TK} \\ \vdots & \vdots & & \vdots \\ x_{N11} & x_{N12} & \cdots & x_{N1K} \\ \vdots & \vdots & & \vdots \\ x_{NT1} & x_{NT2} & \cdots & x_{NTK} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ \vdots & \vdots & & \vdots \\ x_{T1} & x_{T2} & \cdots & x_{TK} \end{bmatrix}$$

Representation		
Cross Sectional	Panel	Time Series
$N \times T$ observations with data of K variables	N observations, each with K variables from T time periods	One (representative) observation with data of K variables from T time periods
Misses dynamic features.	Best of both worlds	Misses individual heterogeneity. Transformation from panel to TS is only possible for an ergodic system.

Forcing a panel dataset into a cross sectional form destroys the possibility to uncover the time-series properties of the data generating process. On the other hand, compressing the data into a time series destroys the heterogeneity of the observations.

Panel data

Panel data is a compromize between cross sectional and time series data structures, with the potential to capture the best of both worlds.

It becomes crucial to use appropriate panel data models especially when there is endogeneity present due to missing variables, which can't be fixed with instrumental variables in cross sectional setting.

Typical panel data models include the Fixed and Random effect models, and the First differeces model. These models are designed to take into account (or get rid of) certain features which are commonly present in panel data.

General form

The basic panel data has the following structure:

$$
\begin{aligned}
y_{it} &= x_{it}'\beta + v_{it} \\
v_{it} &= f_i + u_{it}
\end{aligned}
$$

where the scalar y_{it} denotes the value of the explanatory variable of individual $i = 1, \ldots, N$ at time $t = 1, \ldots, T$, the $K \times 1$ vector x_{it} includes the values of $k = 1, \ldots, K$ explanatory variables for that same observation i at time period t. The composite error term v_{it} is assumed to decompose into a time-invariant fixed effect (individual effect) f_i and a random effect $u_{i,t}$ components. f_i can also be interpreted as an omitted variable.

Panel data is inherently three dimensional, $T \times K \times N$, where each observation (person) has own matrix, consisting of data from K explanatory (and one explained) variables over T time periods.

Our DGP

In this exercise the data generating process (DGP) to be simulated is

$$
y_{i,t} = \beta y_{i,t-1} + f_i + u_{i,t}
$$

where $f_i \sim N(0,1)$ is the time-invariant fixed effect and $u_{i,t} \sim N(0,1)$ is the random effect. In the instructions the variables where actually defined as $f_i =: \alpha_i$ and $u_{it} =: \varepsilon_{it}$, but I will use the symbols from course material for notational convenience.

Our data generating process has an autoregressive feature of order one with persistence parameter $\beta \in \{0.1, 0.5, 0.9\}$. The dimensions are $N = 100$, $T = 10$ and $K = 1$, as the explanatory variables only includes the lags of y itself.

In order to conduct the simulation, an unobserved zeroth period data for y had to be randomly generated (from the standard normal), but is omitted from the following analysis.

Panel data in matrix form

In general

Let's first take a closer look at how the panel data is packed in a matrices in general form.

Explained variables, vectors y_i for observation i, include values for $t = 1, \dots, T$.

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{1,T} \\ \\ y_{2,1} \\ \vdots \\ y_{2,T} \\ \\ \vdots \\ \\ y_{N,1} \\ \vdots \\ y_{N,T} \end{bmatrix} \qquad \text{with} \qquad y_i = \begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,T} \end{bmatrix}$$

Explanatory variable X with (K > 1), each observation has a matrix of data, the values of each K variables for each T time periods. The (balanced) panel data can then be packed in a matrix

$$X = \begin{bmatrix} X_1 \\ \\ \\ X_2 \\ \\ \\ \vdots \\ \\ \\ X_N \end{bmatrix} = \begin{bmatrix} x_{111} & x_{112} & \cdots & x_{11K} \\ \vdots & \vdots & & \vdots \\ x_{1T1} & x_{1T2} & \cdots & x_{1TK} \\ \\ x_{211} & x_{212} & \cdots & x_{21K} \\ \vdots & \vdots & & \vdots \\ x_{2T1} & x_{2T2} & \cdots & x_{2TK} \\ \\ \vdots & \vdots & & \vdots \\ \\ x_{N11} & x_{N12} & \cdots & x_{N1K} \\ \vdots & \vdots & & \vdots \\ x_{NT1} & x_{NT2} & \cdots & x_{NTK} \end{bmatrix}$$

with the elements indexed as x_{itk}.

An individual panel X_i of observation i is given by

$$X_i = \begin{bmatrix} x_{i11} & x_{i12} & \dots & x_{i1K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{iT1} & x_{iT2} & \dots & x_{iTK} \end{bmatrix}$$

which I have highlighted in the larger matrix X by adding space between panels.

Whereas balanced panels have the same length of t for each observation i, in unbalanced panels this is not the case. A typical example of an unbalanced panel arises with macrodata from different countries. US can have some data available from 1950-2020, whereas in Mexico the same data might only be recorded form 1990-2015.

Our DGP

In our case, each of the vectors of explanatory variables y_i corresponding to the panel i, include only values $t = 2, \dots, T$. The explained data has to start from the second time period in order for there to be a corresponding value for explanatory variable $x_2 = lag(y_2) = y_1$.

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} y_{1,2} \\ \vdots \\ y_{1,T} \\ \\ y_{2,2} \\ \vdots \\ y_{2,T} \\ \\ \vdots \\ \\ y_{N,2} \\ \vdots \\ y_{N,T} \end{bmatrix} \qquad \text{with} \quad y_i = \begin{bmatrix} y_{i,2} \\ \vdots \\ y_{i,T} \end{bmatrix}$$

Because the only observable explanatory variable is y’s own lags, it is defined for $t = 1, ..., T - 1$

$$X = lag(y) = \begin{bmatrix} lag(y_1) \\ lag(y_2) \\ \vdots \\ lag(y_N) \end{bmatrix} = \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{1,T-1} \\ \\ y_{2,1} \\ \vdots \\ y_{2,T-1} \\ \\ \vdots \\ \\ y_{N,1} \\ \vdots \\ y_{N,T-1} \end{bmatrix}$$

In our case the whole panel data of explanatory variables can be expressed with a vector because $K = 1$, but it should not be mistaken for cross-sectional data.

Models for panel data

How to choose a suitable model for panel data?

The choice is based on assumptions regarding the relationship between the error components f_i and u_{it} and the explanatory variables x_{it}. The relation between X_i and u_{it} can exert strict or weak exogeneity, or in other words, the regressors X_i can be strictly or weakly exogenous.

$$\mathbb{E}[u_{it} \mid X_i,\ f_i] = \mathbb{E}[u_{it}] = 0 \quad | \quad \text{Strict exogeneity}$$

$$\mathbb{E}[u_{it} \mid \{x_{is}\}_{t \geq s},\ f_i] = 0 \quad | \quad \text{Weak exogeneity}$$

Here the term “fixed effect” for f_i becomes puzzling, because it can actually have fixed or random effects.

$$\mathbb{E}[f_i \mid X_i] = \mathbb{E}[f_i] = 0 \quad | \quad \text{Random effects}$$

$$\mathbb{E}[f_i \mid X_i] \neq 0 \qquad\qquad | \quad \text{Fixed effects}$$

It might be easier to just think of f_i as the individual effect.

Random sampling is assumed in all of them:

$$[u_{it}|\{X_j, f_j\}_{j=1,\ldots,N}] = [u_{it}|X_i, f_i] \quad | \quad \text{Random sampling}$$

meaning that the errors of observation i, u_{it}, are not correlated with data of some other observation j.

Under strict exogeneity and random effects, Random Effects (RE) model is appropriate, but even OLS can be applicable. Strict exogeneity and fixed effects point towards Fixed Effects (FE) or First Differences (FD) models.

If strict exogeneity fails, as is common in econometric settings, then Anderson-Hsiao (AH) or Arellano-Bond (AB) can save the day. Also FE can become asymptotically consistent under weak exogeneity, but typically there is not enough data for this result to be useful.

	Relation of errors and regressors	
	Strict exogeneity	Weak exogeneity
Random effects	RE, OLS	?
Fixed effects	FE, FD	AH, FE

Our data does not fulfil the strict exogeneity assumption due to the lagged dependent variable. I find that only Anderson-Hsiao is pre-asymptotically consistent.

Next I'll present the most common panel data models. Let's start off with a common benchmark, the good ol' OLS.

OLS

Let's discuss the interpretation for running OLS on the panel data in matrix form:

$$y = X\beta + v$$

$$\iff$$

$$\begin{bmatrix} y_{1,1} \\ \vdots \\ y_{1,T} \\ \\ y_{2,1} \\ \vdots \\ y_{2,T} \\ \vdots \\ y_{N,1} \\ \vdots \\ y_{N,T} \end{bmatrix} = \begin{bmatrix} x_{111} & x_{112} & \cdots & x_{11K} \\ \vdots & \vdots & & \vdots \\ x_{1T1} & x_{1T2} & \cdots & x_{1TK} \\ \\ x_{211} & x_{212} & \cdots & x_{21K} \\ \vdots & \vdots & & \vdots \\ x_{2T1} & x_{2T2} & \cdots & x_{2TK} \\ \vdots & \vdots & & \vdots \\ x_{N11} & x_{N12} & \cdots & x_{N1K} \\ \vdots & \vdots & & \vdots \\ x_{NT1} & x_{NT2} & \cdots & x_{NTK} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \\ \vdots \\ \\ \beta_K \end{bmatrix} + \begin{bmatrix} v_{1,1} \\ \vdots \\ v_{1,T} \\ \\ v_{2,1} \\ \vdots \\ v_{2,T} \\ \vdots \\ v_{N,1} \\ \vdots \\ v_{N,T} \end{bmatrix} \qquad \text{Panel data}$$

Panel data has N observations, each with values from T periods. Applying OLS means that the data is treated as cross-sectional with $N \times T$ observations from one time period. Hence, the 3D structure of panel data is lost, along with possible time series elements of the DGP.

$$y = X\beta + v$$

$$\iff$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \\ y_{T+1} \\ \vdots \\ y_{(N\times T)-1} \\ y_{N\times T} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \vdots & & \vdots \\ x_{T1} & x_{T2} & \cdots & x_{TK} \\ x_{(T+1)1} & x_{(T+1)2} & \cdots & x_{(T+1)K} \\ \vdots & \vdots & & \vdots \\ x_{(N\times T-1)1} & x_{(N\times T-1)2} & \cdots & x_{(N\times T-1)K} \\ x_{(N\times T)1} & x_{(N\times T)2} & \cdots & x_{(N\times T)K} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \\ \vdots \\ \\ \beta_K \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_T \\ v_{T+1} \\ \vdots \\ v_{(N\times T)-1} \\ v_{N\times T} \end{bmatrix} \qquad \text{Cross sectional data}$$

In our case the OLS entails

$$y_i = x'\beta + v_i$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} y_{1,2} \\ \vdots \\ y_{1,T} \\ \\ \vdots \\ \\ y_{N,2} \\ \vdots \\ y_{N,T} \end{bmatrix} = \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{1,T-1} \\ \\ \vdots \\ \\ y_{N,1} \\ \vdots \\ y_{N,T-1} \end{bmatrix} * \beta + \begin{bmatrix} f_1 + u_{1,2} \\ \vdots \\ f_1 + u_{1,T} \\ \\ \vdots \\ \\ f_N + u_{N,2} \\ \vdots \\ f_N + u_{N,T} \end{bmatrix} \qquad \text{Panel data}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} y_{1,2} \\ \vdots \\ y_{1,T} \\ \\ \vdots \\ \\ y_{N,2} \\ \vdots \\ y_{N,T} \end{bmatrix} = \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{1,T-1} \\ \\ \vdots \\ \\ y_{N,1} \\ \vdots \\ y_{N,T-1} \end{bmatrix} * \beta + \begin{bmatrix} v_{1,2} \\ \vdots \\ v_{1,T} \\ \\ \vdots \\ \\ v_{N,2} \\ \vdots \\ v_{N,T} \end{bmatrix} \qquad \text{Panel data}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} y_2 \\ \vdots \\ y_T \\ y_{T+1} \\ \\ \vdots \\ \\ y_{(N\times T)-1} \\ y_{N\times T} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{T-1} \\ y_T \\ \\ \vdots \\ \\ y_{(N\times T)-2} \\ y_{(N\times T)-1} \end{bmatrix} * \beta + \begin{bmatrix} v_2 \\ \vdots \\ v_T \\ v_{T+1} \\ \\ \vdots \\ \\ v_{(N\times T)-1} \\ v_{N\times T} \end{bmatrix} \qquad \text{Cross sectional data}$$

Transposes are not needed for x, as β is a scalar.

Using OLS plainly on the panel data is not suitable, if weak exogeneity and random effects requirements are not met. In that case we ought to use a panel data model. Such a model first conducts a transformation on the data to take care of some particular issue at hand, and afterwards uses OLS to estimate the parameters. Next we'll take a look at some of these models.

Random Effects Estimator

If it is the case, that

$$\mathbb{E}[f_i|x] = 0 \quad | \quad \text{Random effects}$$

then OLS is unbiased and consistent, but the standard errors are incorrect.

The following transformation, using the covariance matrix of the composite error term, can be applied on the model, creating the Random effects -model. The model can be estimated using OLS after the transformation.

$$V_i = \mathbb{E}[v_i v_i' | x_i] = \sigma_u^2 I_T + \sigma_f^2 J_T$$

where I_T is a T dimensional identity matrix and J_T is $T \times T$ dimensional matrix of ones.

Multiplying the equation by the square root (in matrix sense) of V gives the transformed equation

$$\begin{gathered} V^{\frac{1}{2}} y_i = V_i^{\frac{1}{2}} X_i \beta + V_i^{\frac{1}{2}} v_i \\ \iff \\ \tilde{y}_i = \tilde{X}_i \beta + \tilde{v}_i \end{gathered}$$

The estimate β^{RE} can be obtained by running OLS on the transformed system. As noted above in the OLS-section, the transformed panel data is now treated as cross sectional in the estimation of Random effects estimator.

$$\tilde{y} = \tilde{x}\beta + \tilde{v}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} \tilde{y}_{1,2} \\ \vdots \\ \tilde{y}_{1,T} \\ \\ \vdots \\ \\ \tilde{y}_{N,2} \\ \vdots \\ \tilde{y}_{N,T} \end{bmatrix} = \begin{bmatrix} \tilde{y}_{1,1} \\ \vdots \\ \tilde{y}_{1,T-1} \\ \\ \vdots \\ \\ \tilde{y}_{N,1} \\ \vdots \\ \tilde{y}_{N,T-1} \end{bmatrix} * \beta \ + \begin{bmatrix} \tilde{v}_{1,2} \\ \vdots \\ \tilde{v}_{1,T} \\ \\ \vdots \\ \\ \tilde{v}_{N,2} \\ \vdots \\ \tilde{v}_{N,T} \end{bmatrix} \qquad \text{Panel data}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} \tilde{y}_{2} \\ \vdots \\ \tilde{y}_{T} \\ \\ \vdots \\ \\ \tilde{y}_{N \times T} \end{bmatrix} = \begin{bmatrix} \tilde{y}_{1} \\ \vdots \\ \tilde{y}_{T-1} \\ \\ \vdots \\ \\ \tilde{y}_{(N \times T)-1} \end{bmatrix} * \beta \ + \begin{bmatrix} \tilde{v}_{2} \\ \vdots \\ \tilde{v}_{T} \\ \\ \vdots \\ \\ \tilde{v}_{N \times T} \end{bmatrix} \qquad \text{Cross sectional data}$$

Fixed Effects Estimator

The starting point in fixed effects model is, that

$$\mathbb{E}[f_i|x] \neq 0 \quad | \quad \text{Fixed effects}$$

This causes endogeneity, and invalidates the use of OLS.

The impact of the fixed effect can be removed by centering the model.

$$\begin{gathered}(y_{it} - \bar{y}_i) = (x_{it} - \bar{x}_i)\beta + (v_{it} - \bar{v}_i) \\ \iff \\ \tilde{y}_i = \tilde{X}_i\beta + \tilde{v}_i\end{gathered}$$

Let's look more closely at the composite errors $v_{it} = f_i + u_{it}$. The fixed effects f_i are constant in time, and hence disappear in this procedure:

$$\begin{aligned}v_{it} - \bar{v}_{it} &= f_i + u_{it} - f_i + \bar{u}_{it} \\ &= u_{it} - \bar{u}_{it} \\ &=: \bar{u}_{it}\end{aligned}$$

The Fixed effects estimator β^{FE} can be estimated by applying OLS on the centered data, similarly as in the previous case of RE-estimator.

$$\tilde{y} = \tilde{x}\beta + \tilde{v}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} y_{1,2} - \bar{y}_1 \\ \vdots \\ y_{1,T} - \bar{y}_1 \\ \\ \vdots \\ \\ y_{N,2} - \bar{y}_N \\ \vdots \\ y_{N,T} - \bar{y}_N \end{bmatrix} = \begin{bmatrix} y_{1,1} - \bar{y}_1 \\ \vdots \\ y_{1,T-1} - \bar{y}_1 \\ \\ \vdots \\ \\ y_{N,1} - \bar{y}_N \\ \vdots \\ y_{N,T-1} - \bar{y}_N \end{bmatrix} * \beta \ + \ \begin{bmatrix} v_{1,2} - \bar{v}_1 \\ \vdots \\ v_{1,T} - \bar{v}_1 \\ \\ \vdots \\ \\ v_{N,2} - \bar{v}_N \\ \vdots \\ v_{N,T} - \bar{v}_N \end{bmatrix} \qquad \text{Panel data}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} \tilde{y}_2 \\ \vdots \\ \tilde{y}_T \\ \\ \vdots \\ \\ \tilde{y}_{N\times T} \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{T-1} \\ \\ \vdots \\ \\ \tilde{y}_{(N\times T)-1} \end{bmatrix} * \beta \ + \ \begin{bmatrix} \tilde{u}_2 \\ \vdots \\ \tilde{u}_T \\ \\ \vdots \\ \\ \tilde{u}_{N\times T} \end{bmatrix} \qquad \text{Cross sectional interpretation}$$

First-Difference Estimator

Another way to remove the impact of the fixed effect f_i is to use first differences.

$$\Delta y_i = \Delta X_i \beta + \Delta v_i$$

$$\iff$$

$$(y_{i,t} - y_{i,t-1}) = (x_{i,t} - x_{i,t-1})\beta + (v_{i,t} - v_{i,t-1})$$

$$\iff$$

$$\begin{bmatrix} y_{1,3} - y_{1,2} \\ \vdots \\ y_{1,T} - y_{1,T-1} \\ \\ \vdots \\ \\ y_{N,3} - y_{N,2} \\ \vdots \\ y_{N,T} - y_{N,T-1} \end{bmatrix} = \begin{bmatrix} y_{1,2} - y_{1,1} \\ \vdots \\ y_{1,T-1} - y_{1,T-2} \\ \\ \vdots \\ \\ y_{N,2} - y_{N,1} \\ \vdots \\ y_{N,T-1} - y_{N,T-2} \end{bmatrix} * \beta \ + \begin{bmatrix} v_{1,3} - v_{1,2} \\ \vdots \\ v_{1,T} - v_{1,T-1} \\ \\ \vdots \\ \\ v_{N,3} - v_{N,2} \\ \vdots \\ v_{N,T} - v_{N,T-1} \end{bmatrix} \qquad \text{Panel data}$$

$$\iff$$

$$\begin{bmatrix} \Delta y_2 \\ \vdots \\ \Delta y_T \\ \\ \vdots \\ \\ \Delta y_{N \times T} \end{bmatrix} = \begin{bmatrix} \Delta y_1 \\ \vdots \\ \Delta y_{T-1} \\ \\ \vdots \\ \\ \Delta y_{N \times T-1} \end{bmatrix} * \beta \ + \begin{bmatrix} \Delta u_2 \\ \vdots \\ \Delta u_T \\ \\ \vdots \\ \\ \Delta u_{N \times T} \end{bmatrix} \qquad \text{Cross sectional interpretation}$$

Likewise with FE-estimator, also here the composite error simplifies to the random effect:

$$\begin{aligned} v_{it} - v_{it-1} &= f_i + u_{it} - f_i + u_{it-t} \\ &= u_{it} - u_{it-1} \\ &=: \Delta u_{it} \end{aligned}$$

We also lose all observations from one time-period when differencing.

Endogeneity

If strict exogeneity falls, FD becomes biased. Let's take another look at our data.

$$y_{i,t} = \beta y_{i,t-1} + f_i + u_{i,t}$$

The FD model conducts the transformation

$$\begin{aligned}\Delta y_{it} &= \beta \Delta y_{it-1} + \Delta v_{it} \\ y_{it} - y_{it-1} &= \beta(y_{it-1} - y_{it-2}) + u_{it} - u_{it-1}\end{aligned}$$

where the fixed effects f_i cancels. Alternative expressions for the LHS (used later) are

$$\begin{aligned}y_{it} - y_{it-1} &= \beta y_{i,t-1} + f_i + u_{i,t} - y_{it-1} \\ &= (\beta - 1) y_{i,t-1} + f_i + u_{i,t} && | \quad \text{or opening both} \\ &= \beta(y_{i,t-1} - y_{i,t-2}) + u_{i,t} - u_{it-1}\end{aligned}$$

It is apparent, that strict (and weak?) exogeneity fails in FD model, as

$$\begin{aligned}\mathbb{E}[\Delta x_{it} \Delta v_{it}] &= \mathbb{E}[\Delta y_{it-1} \Delta v_{it}] \\ &= \mathbb{E}[(y_{it-1} - y_{it-2})(u_{it} - u_{it-1})] \\ &= \mathbb{E}[((\beta - 1) y_{i,t-2} + f_i + u_{i,t-1})(u_{it} - u_{it-1})] && | \quad \mathbb{E}[u_{it-1} u_{it-1}] \neq 0 \\ &\neq 0\end{aligned}$$

as there are u_{it-1} in both terms.

How can we combat endogeneity? Anderson-Hsiao estimator offers an answer.

Anderson-Hsiao Estimator

AH-estimator uses $y_{i,t-2}$ as an instrument for the endogenous regressor $\Delta x_{i,t} = \Delta y_{i,t-1} = y_{it-1} - y_{it-2}$.

The choice makes sense, because the instrument is correlated with the regressor $\Delta y_{i,t-1}$

$$\begin{aligned}\mathbb{E}[y_{it-2}\Delta x_{it}] &= \mathbb{E}[y_{it-2}\Delta y_{it-1}] \\ &= \mathbb{E}[y_{it-2}(y_{it-1} - y_{it-2})] \\ &\neq 0\end{aligned}$$

and the regressand $\Delta y_{i,t}$

$$\begin{aligned}\mathbb{E}[y_{it-2}\ \Delta y_{it}] &= \mathbb{E}[y_{it-2}\ (y_{it} - y_{it-1})] \\ &= \mathbb{E}[y_{it-2}\ (\beta(y_{i,t-1} - y_{i,t-2}) + u_{i,t} - u_{it-1})] \\ &\neq 0\end{aligned}$$

but is uncorrelated with the error term $\Delta v_{i,t}$

$$\begin{aligned}\mathbb{E}[y_{it-2}\ \Delta v_{it}] &= \mathbb{E}[y_{it-2}\ (u_{it} - u_{it-1})] \\ &= \mathbb{E}[y_{it-2}\ u_{i,t}] - \mathbb{E}[y_{it-2} u_{it-1})] \\ &= 0\end{aligned}$$

The AH estimator is given by

$$\begin{aligned}\hat{\beta}^{IV} &= \frac{cov(z_{it}, \Delta y_{it})}{cov(z_{it}, \Delta x_{it})} \\ &= \frac{cov(y_{i,t-2}, \Delta y_{it})}{cov(y_{i,t-2}, \Delta y_{it-1})}\end{aligned}$$

This makes it adept in removing endogeneity caused by the lagged dependent variable. As it turns out, that AH-estimator is the best estimator (among the ones tried) for our data.

Arellano-Bond Estimator

Multiple orthogonality conditions in GMM setting.

Monte Carlo study

Next I simulate panel data and use it to estimate different models to compare their performance. To recap, the DGP is

$$y_{i,t} = \beta y_{i,t-1} + f_i + u_{i,t}$$

$$\text{s.t.}$$

$$\begin{aligned} y_{i,0} &\sim N(0,1) \\ f_i &\sim N(0,1) \\ u_{i,t} &\sim N(0,1) \\ \beta &\in \{0.1, 0.5, 0.9\} \end{aligned}$$

The zeroth period y-values are drawn from the normal distribution with the fixed effect f_i added on top, in order to get the recursive process going. The rest of the data is generated according to the GDP described above. The zeroth period is omitted from the output of the data simulation function.

The data simulation function

```
simulate_panel <- function(N, T, beta){

    # Generate a Nx1 vector of fixed effects
    alpha <- rnorm(n=N, mean=0, sd=1)

    # Generate T+1 vectors of dimension Nx1. The first is period zero values
    epsilon <- matrix(rnorm(n=N*T, mean=0, sd=1), nrow=N, byrow=F)

    # Generate the actual data, T+1 vectors of Nx1
    data <- matrix(NA, nrow=N, ncol=(T+1))
    data[,1] <- rnorm(n=N, mean=0, sd=1) + alpha # Period zero values amended by the fixed effect

    for(i in 2:(T+1)){
        data[,i] <- beta*data[,i-1] + alpha + epsilon[,i-1]
    }

    return(data[,-1])
}
data <- simulate_panel(N=100, T=10, beta=0.1)
colnames(data) <- c(1:10)
as_tibble(data)
```

```
# A tibble: 100 x 10
      `1`    `2`    `3`    `4`     `5`     `6`     `7`     `8`    `9`
```

```
    <dbl>  <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>
 1 -1.58  -2.68  -0.970 -0.921 -4.30   -1.71   -2.43   -2.92   -2.39
 2  2.70   2.41   2.20   1.13   0.770   2.60    1.85    2.40    2.59
 3  0.356  0.907 -1.03   0.683 -0.768 -2.47   -1.99   -0.602  -0.254
 4  0.196  0.816  1.26   1.58   0.0117  0.255   1.78    1.71   -0.152
 5 -2.31  -0.195  1.73  -0.561  1.83    0.0227 -0.936  -0.259  -1.07
 6 -0.554  0.625 -1.23  -0.959 -0.751 -1.16   -0.0875 -0.825   0.732
 7  1.72   1.41   1.53   0.622  0.856   0.253   1.84    1.99    1.47
 8 -1.21  -0.743 -1.91  -0.487  1.01   -0.335  -1.65   -0.647  -1.10
 9 -2.15   0.886  0.330 -0.118 -0.0461  0.298  -1.34    0.0522  1.03
10  2.05   1.03   1.46   2.61   3.26    0.393   3.10    2.05    2.19
# ... with 90 more rows, and 1 more variable: `10` <dbl>
```

The MC-algorithm

```
conduct_mc <- function(R, T, N, beta){
    #start <- Sys.time()
    set.seed(1917)

    rounds <- length(beta)

    # Arrays to save the all the results
    all <- list(ols               = list(),
                fixed_effects     = list(),
                first_differences = list(),
                anderson_hsiao    = list()
    )

    # Arrays for the averages
    array1 <- matrix(NA, nrow=2, ncol=rounds)
    averages <- list(ols               = array1,
                     fixed_effects     = array1,
                     first_differences = array1,
                     anderson_hsiao    = array1
    )

    # Arrays for residual autocorrelations
    residual_ac <- list(ols               = list(),
                        fixed_effects     = list(),
                        first_differences = list(),
                        anderson_hsiao    = list()
    )

    for(j in 1:rounds){
        # These matrices are just placeholders for in-loop results
        array2 <- matrix(NA, nrow=2, ncol=R)
        models <- list(ols               = array2,
                       fixed_effects     = array2,
                       first_differences = array2,
                       anderson_hsiao    = array2
                        )
        residuals <- list(ols               = c(),
                       fixed_effects     = c(),
                       first_differences = c(),
                       anderson_hsiao    = c()
                        )

        for(i in 1:R){
            # Simulate a NxT set of data
            data <- simulate_panel(N=N, T=T, beta=beta[[j]])

            # OLS
            y <- c(data[,-1])
            x <- c(data[,-T])
            model_ols <- lm(y ~ x)
            models$ols[,i] <- model_ols$coefficients # Save the coefficients
            #residuals$ols[,i] <- acf(model_ols$residuals, plot = FALSE)$acf[2:3] # Save residual autoc

            res_ols <- matrix(model_ols$residuals, nrow=T-1)

            residuals$ols[i] <- lm(c(res_ols[-1,]) ~ c(res_ols[-nrow(res_ols),]))$coefficients[[2]]

            #res_ols2 <- c()
            #for(l in 1:ncol(res_ols)){
            #    res_ols2[[l]] <- acf(res_ols[,l], plot = FALSE)$acf[2]
            #}
            #residuals$ols[,i] <- mean(res_ols2) # Save residual autocorrelations for lags 1 and 2


            # Fixed effects: Refract time averages by rows, convert to CS and run OLS
            time_means <-apply(X=data, MARGIN=1, FUN=mean)
            data_fe <- data - time_means
            y_fe <- c(data_fe[,-1])
            x_fe <- c(data_fe[,-T])
            model_fe <- lm(y_fe ~ x_fe)
            models$fixed_effects[,i] <- model_fe$coefficients
            #residuals$fixed_effects[,i] <- acf(model_fe$residuals, plot = FALSE)$acf[2:3] # Residual a

            res_fe <- matrix(model_fe$residuals, nrow=T-1)
            residuals$fixed_effects[i] <- lm(c(res_fe[-1,]) ~ c(res_fe[-nrow(res_fe),]))$coefficients[[

            #res_fe2 <- c()
            #for(l in 1:ncol(res_fe)){
            #    res_fe2[[l]] <- acf(res_fe[,l], plot = FALSE)$acf[2]
            #}
            #residuals$fixed_effects[,i] <- mean(res_fe2)

            # First differences: Transform to first differences, convert to CS and run OLS
            diff <- t(apply(X=data, MARGIN=1, FUN=diff))
            y_fd <- c(diff[,-1])
            x_fd <- c(diff[,-(T-1)])
```

```
            model_fd <- lm(y_fd ~ x_fd)
            models$first_differences[,i] <- model_fd$coefficients
            #residuals$first_differences[,i] <- acf(model_fd$residuals, plot = FALSE)$acf[2:3] # Residu

            res_fd <- matrix(model_fd$residuals, nrow=T-2)
            residuals$first_differences[i] <- lm(c(res_fd[-1,]) ~ c(res_fd[-nrow(res_fd),]))$coefficien


            #res_fd2 <- c()
            #for(l in 1:ncol(res_fd)){
            #    res_fd2[[l]] <- acf(res_fd[,l], plot = FALSE)$acf[2]
            #}
            #residuals$first_differences[,i] <- mean(res_fd2)


            # Anderson-Hsiao: Here we use the second lag of as instrument for y= delta y and x= delta y
            y_instrument <- c(data[,-c((T-1):T)])
            models$anderson_hsiao[,i] <- cov(y_instrument, y_fd)/cov(y_instrument, x_fd)
            ah_residuals <- models$anderson_hsiao[1,i]*x_fd - y_fd # Calculate predictions and residual
            #residuals$anderson_hsiao[,i] <- acf(ah_residuals, plot = FALSE)$acf[2:3]

            res_ah <- matrix(ah_residuals, nrow=T-2)
            residuals$anderson_hsiao[i] <- lm(c(res_ah[-1,]) ~ c(res_ah[-nrow(res_ah),]))$coefficients[

            #res_ah2 <- c()
            #for(l in 1:ncol(res_ah)){
            #    res_ah2[[l]] <- acf(res_ah[,l], plot = FALSE)$acf[2]
            #}
            #residuals$anderson_hsiao[,i] <- mean(res_ah2)

            # Random effects: Divide by the square root of covariance matrix or some shit
            #grun.fe <- plm(inv~value+capital, data = Grunfeld, model = "within")

        }

        for(k in 1:length(models)){
            all[[k]][[j]] <- models[[k]]
            residual_ac[[k]][[j]] <- residuals[[k]] # Save the R residual autocorrelation for beta on r
        }
    }

    # Calculate the averages
    for(j in 1:rounds){
        for(k in 1:length(all)){
            averages[[k]][,j] <- apply(X=all[[k]][[j]], MARGIN=1, FUN=mean)
            rownames(averages[[k]]) <- c("Intercept", "Coefficient")
            colnames(averages[[k]]) <- c("Beta=0.1", "Beta=0.5", "Beta=0.9")
        }
    }

    #end <- Sys.time()

    return(list(all=all, averages=averages, residual_autocorrelations=residual_ac))
```

```
}
```

Results: Average estimated parameter values

Let's take a look at the average estimates from the Monte Carlo run

```
mc_run <- conduct_mc(R=500, T=10, N=100, beta=c(0.1, 0.5, 0.9))

ols <- t(mc_run$averages$ols[2,])
fe <- t(mc_run$averages$fixed_effects[2,])
fd <- t(mc_run$averages$first_differences[2,])
ah <- t(mc_run$averages$anderson_hsiao[1,])
coeffs <- rbind(ols, fe, fd, ah)
rownames(coeffs) <- c("OLS", "Ficed effects", "First differences", "Anderson-Hsiao")

kable(round(coeffs, rounding), "latex", booktabs=T, caption="MC-results") %>%
      kable_styling(latex_options = "hold_position")
```

Table 1: MC-results

	Beta=0.1	Beta=0.5	Beta=0.9
OLS	0.590	0.881	1.073
Ficed effects	-0.026	0.313	0.753
First differences	-0.451	-0.247	0.175
Anderson-Hsiao	0.098	0.500	0.905

OLS estimator seems to overestimate the real value, whereas the Fixed effects and the First differences estimators underestimate the value, the former managing to perform much better than the latter. The Anderson-Hsiao estimator, however, clears the table in accuracy, which seems to be spot on.

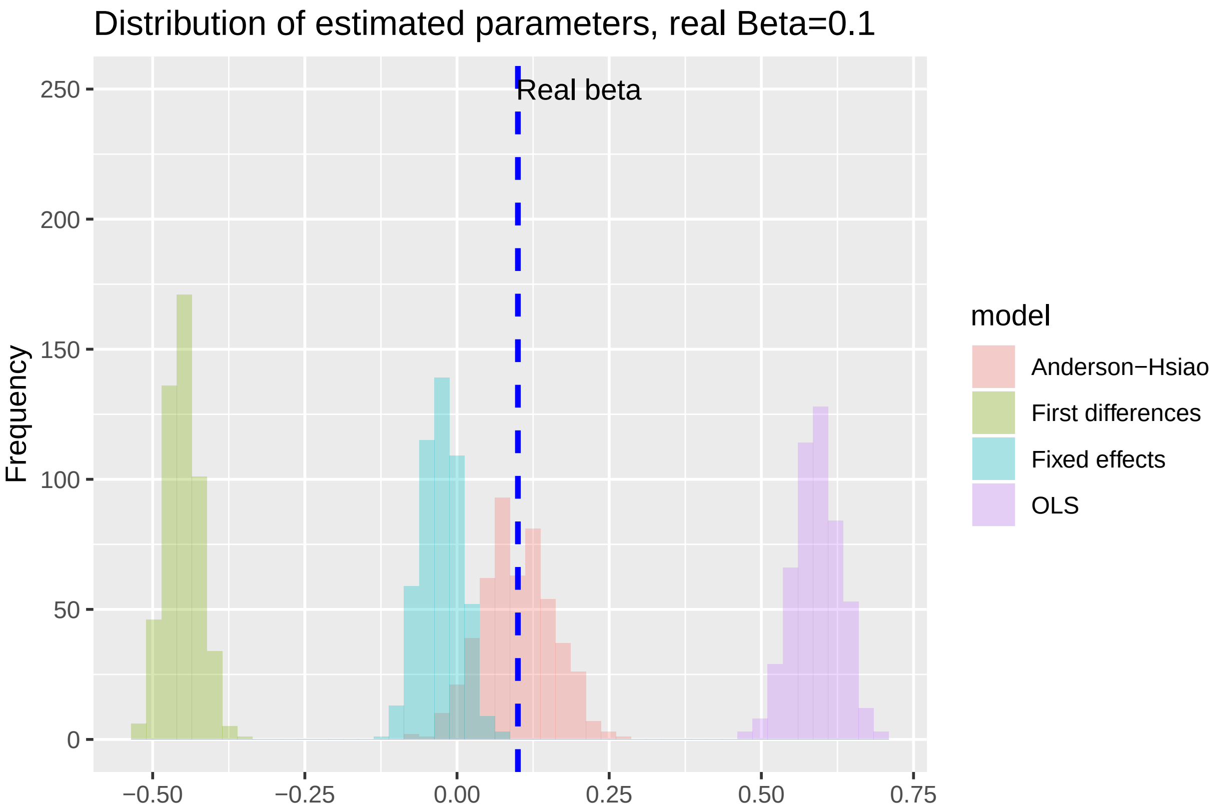
It is quite clear, that OLS is incapable of learning the true autoregressive parameter due to the endogeneity caused by the fixed individual effects $f_i = \alpha_i$ correlating with explanatory variables $x_{it} = lag(y_it - 1) = y_{it-1} + f_i + u_{it-1}$, because they include the common term. However, it shouldn't be impossible for the OLS to pick up the AR(1) component, as the regression was built up in a manner, where the dynamic relation is explicit.

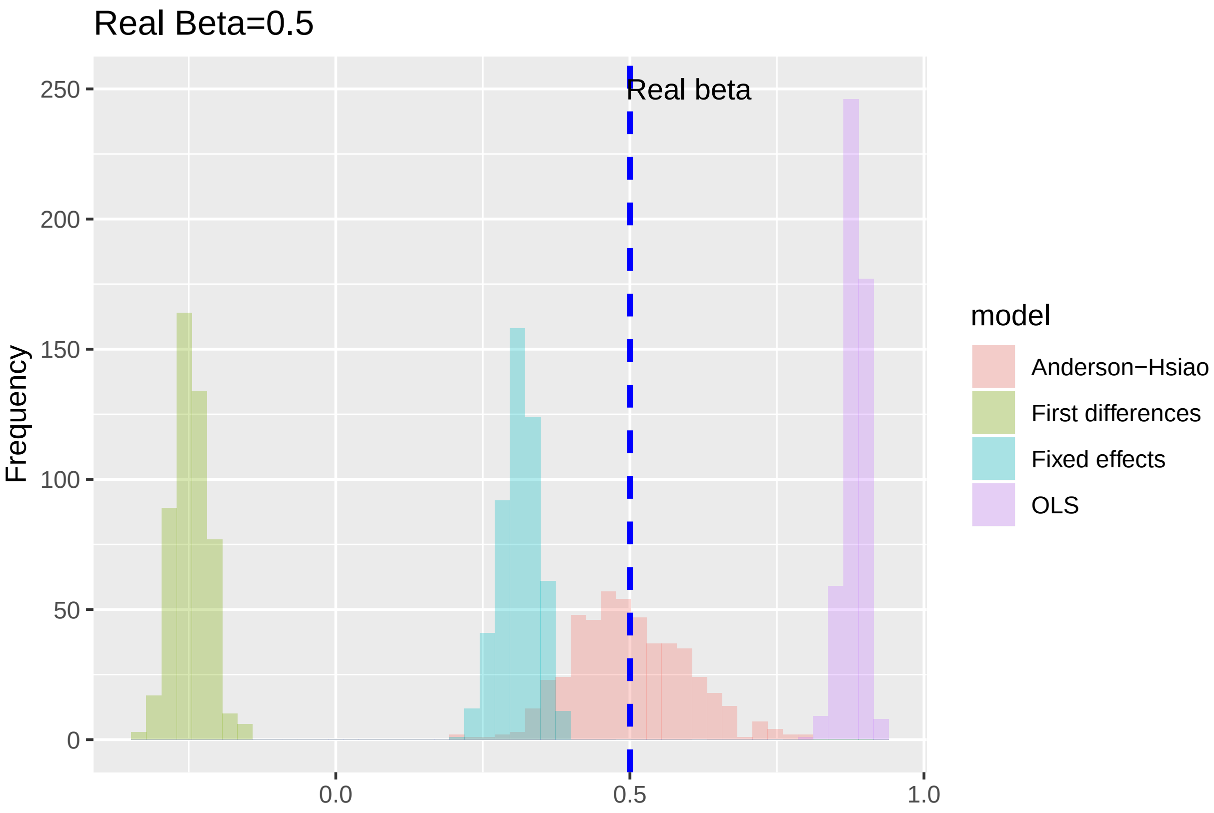
FE and FD manage to remove the fixed effect, by definition, but the under- and overestimate the ar-component, respectively, because . . . ?

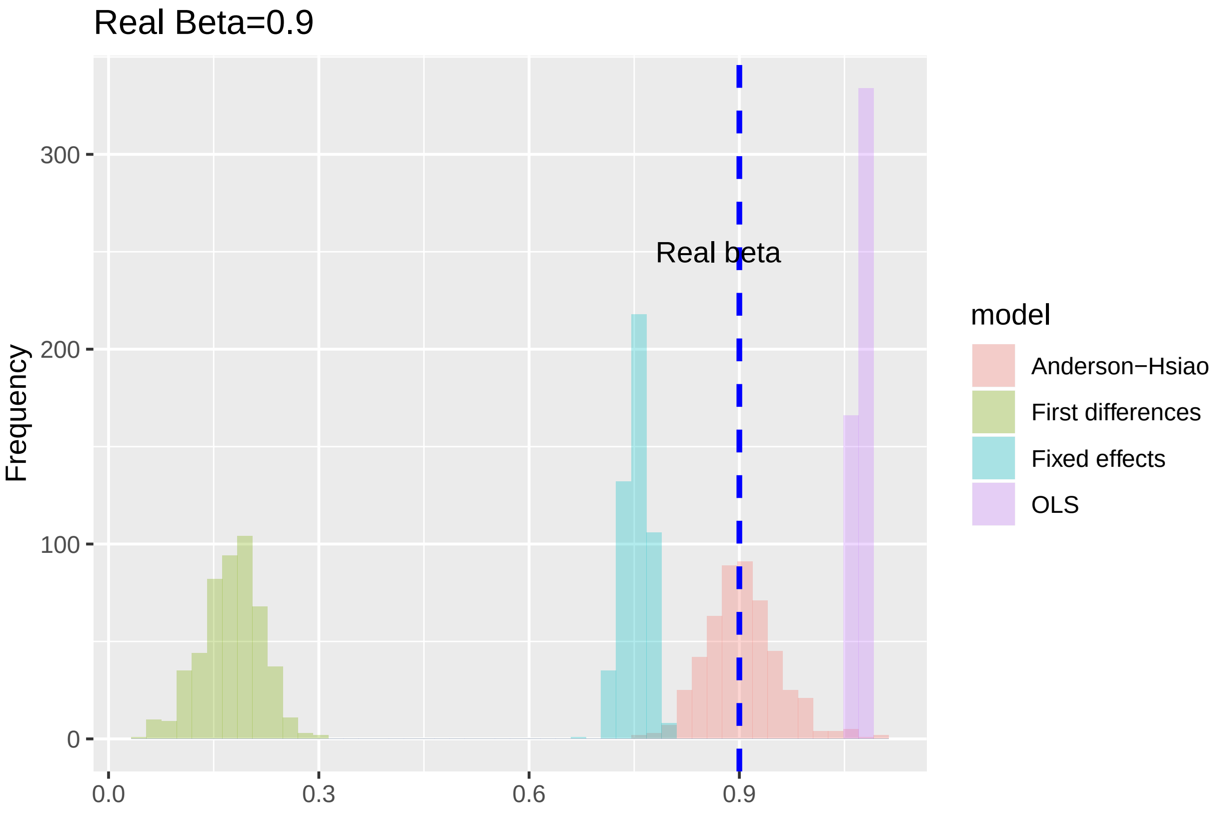
All estimators except HA were estimated alongside an intercept, but I omitted them due to their miniscule values. The intercept could have been omitted also otherwise at least with FD-estimator, but that didn't make a large difference in the results.

Results: Distribution of estimated parameters

Next we'll visualize the distribution of estimated values for the parameters over the 500 MC-rounds




Real Beta=0.5
Real beta
Frequency
250
200
150
100
50
0
0.0
0.5
1.0
model
Anderson−Hsiao
First differences
Fixed effects
OLS


Real Beta=0.9
Real beta
Frequency
300
200
100
0
0.0
0.3
0.6
0.9
model
Anderson−Hsiao
First differences
Fixed effects
OLS

When the real β gets larger, OLS becomes more and FD less precise, although neither are accurate. Fixed effects estimator also seems to improve precision by increase in β. HA has the lowest precision but highest accuracy (widest distribution of estimated values, but centered on the correct location).

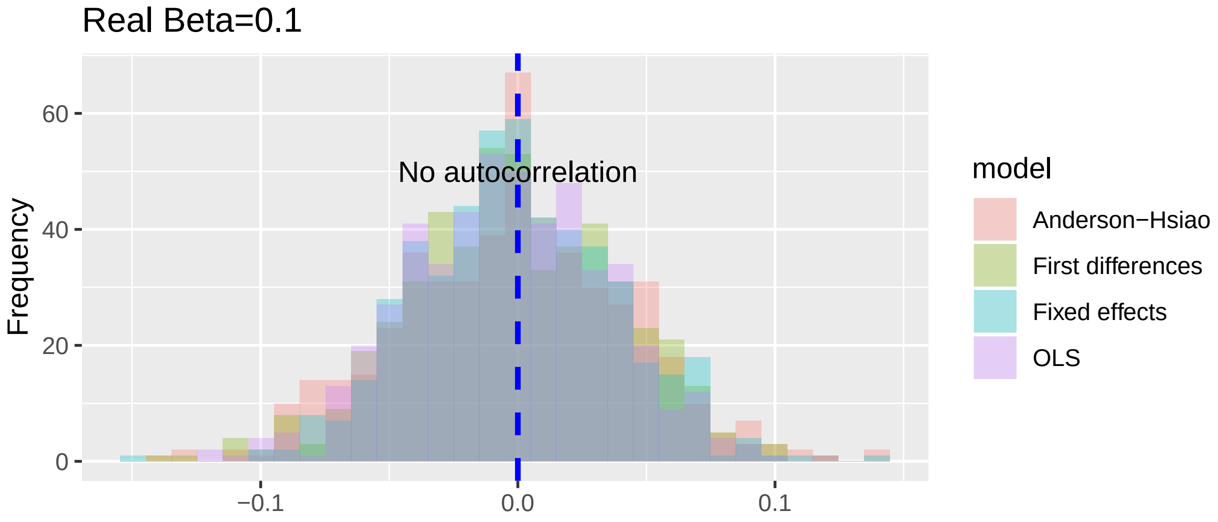
Autocorrelation of the residuals η_{it}^{k}

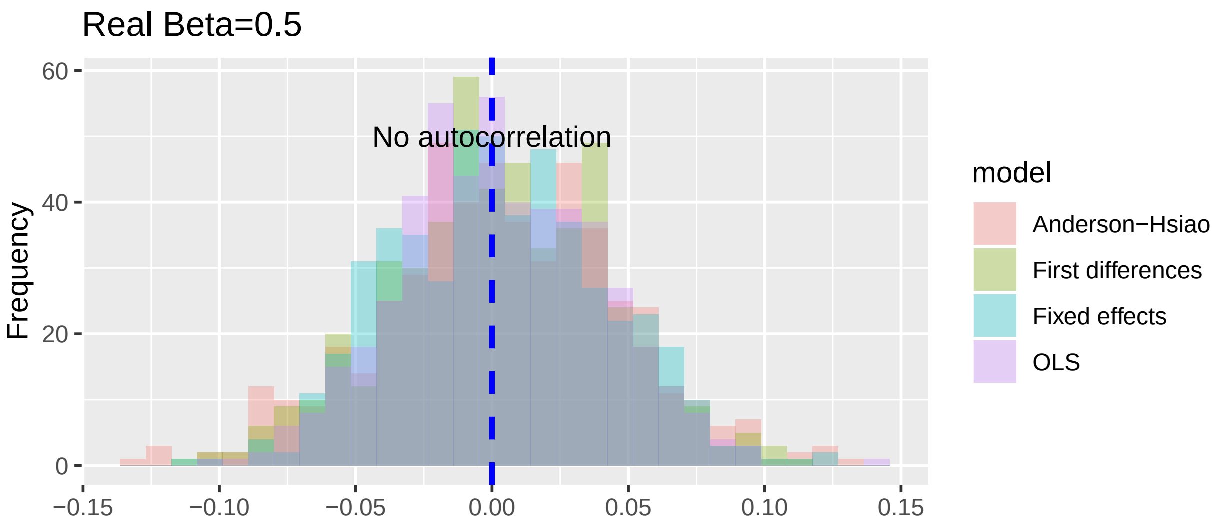
Finally, let's take a look at the residuals of different models, and chech if they exert autocorrelation. I have calculated the first order autocorrelations for residuals from each model, for each of the 500 MC rounds. (Each round's autocorrelation is a mean among the autocorrelations calculated for each of the 100 observations.)

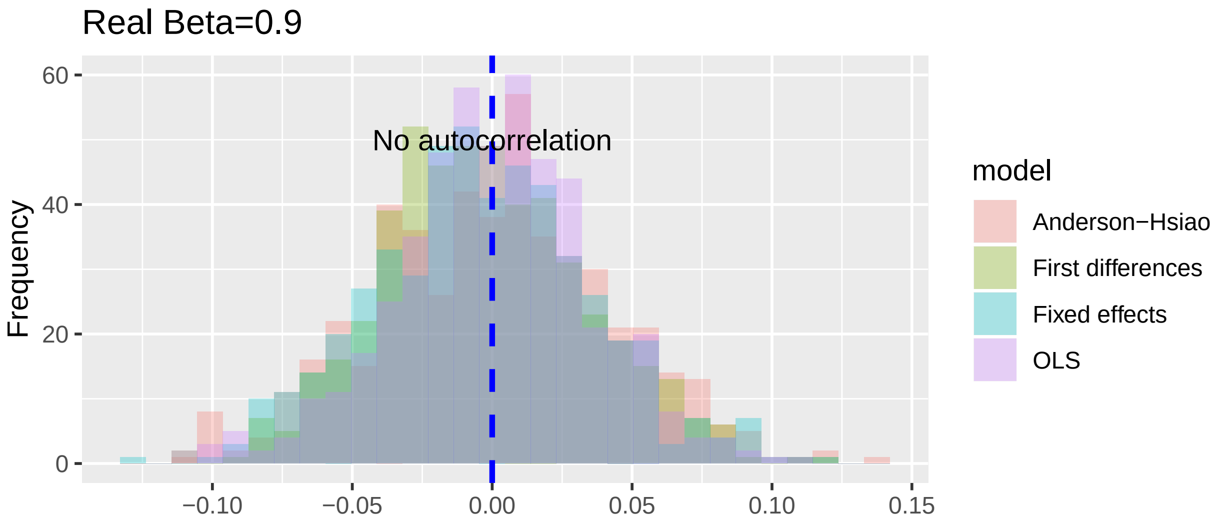
Here are their distributions.

Table 2: Mean residual autocorrelations

	Beta=0.1	Beta=0.5	Beta=0.9
OLS	-0.004	0.003	0.000
Ficed effects	-0.001	0.003	-0.004
First differences	-0.001	0.003	-0.003
Anderson-Hsiao	-0.002	0.003	0.000







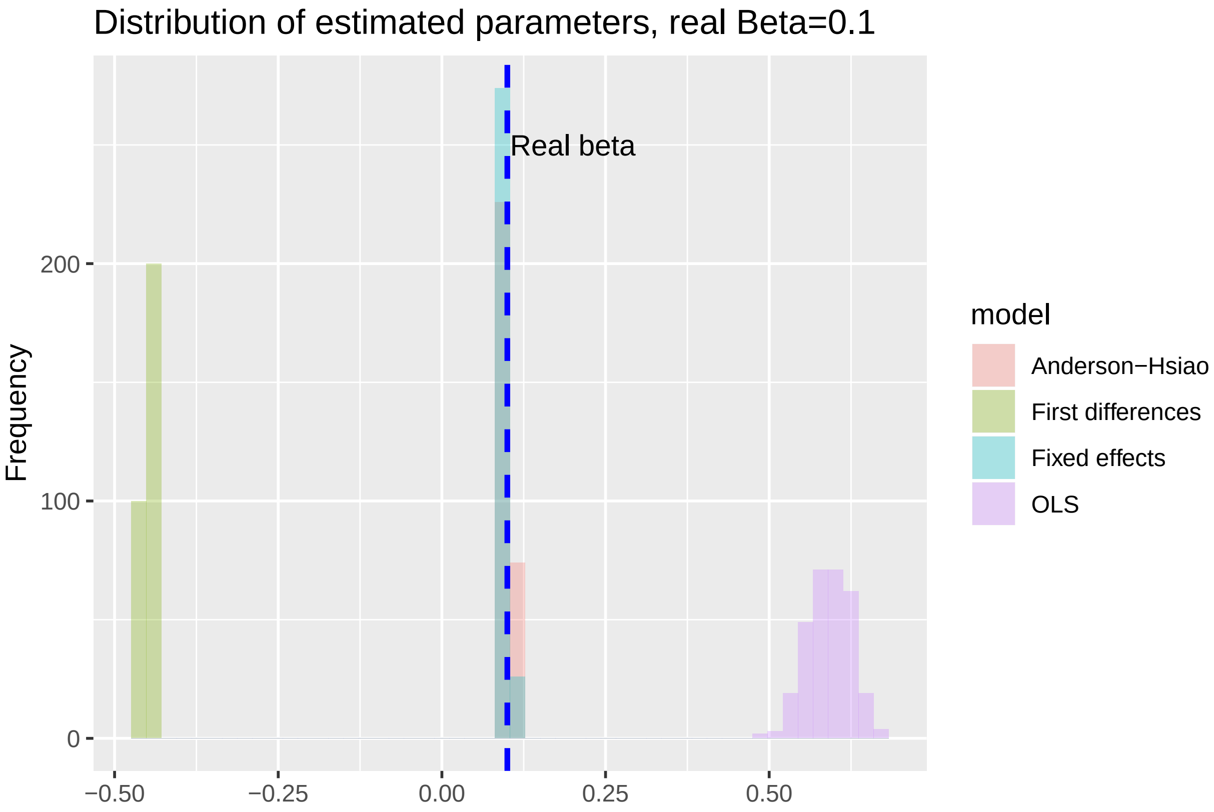
Increasing the time dimension

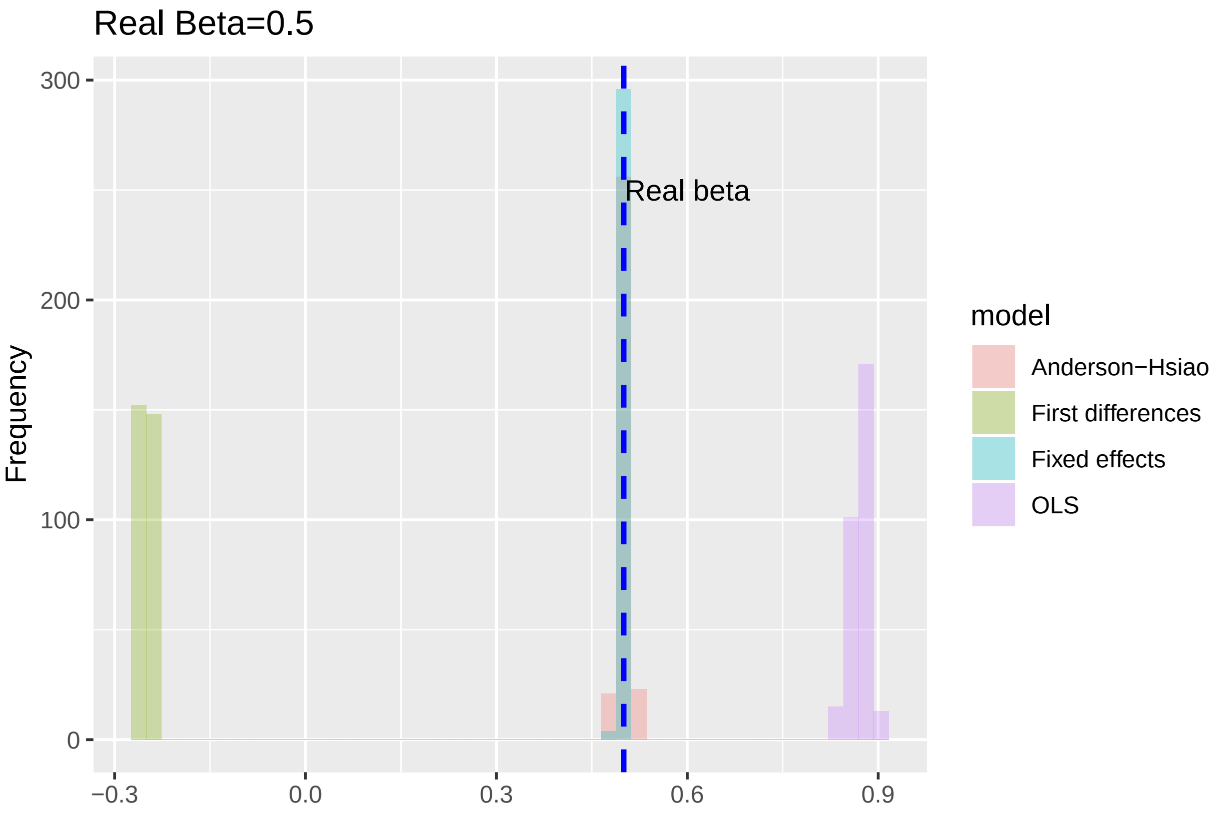
To finish things off, let's see if the estimators' performance improves if we increase the availability of data from the time dimension. Instead of 10 time periods, let's have 500.

```r
mc_run <- conduct_mc(R=300, T=500, N=100, beta=c(0.1, 0.5, 0.9))

ols <- t(mc_run$averages$ols[2,])
fe <- t(mc_run$averages$fixed_effects[2,])
fd <- t(mc_run$averages$first_differences[2,])
ah <- t(mc_run$averages$anderson_hsiao[1,])
coeffs <- rbind(ols, fe, fd, ah)
rownames(coeffs) <- c("OLS", "Ficed effects", "First differences", "Anderson-Hsiao")

kable(round(coeffs, rounding), "latex", booktabs=T, caption="MC-results") %>%
      kable_styling(latex_options = "hold_position")
```

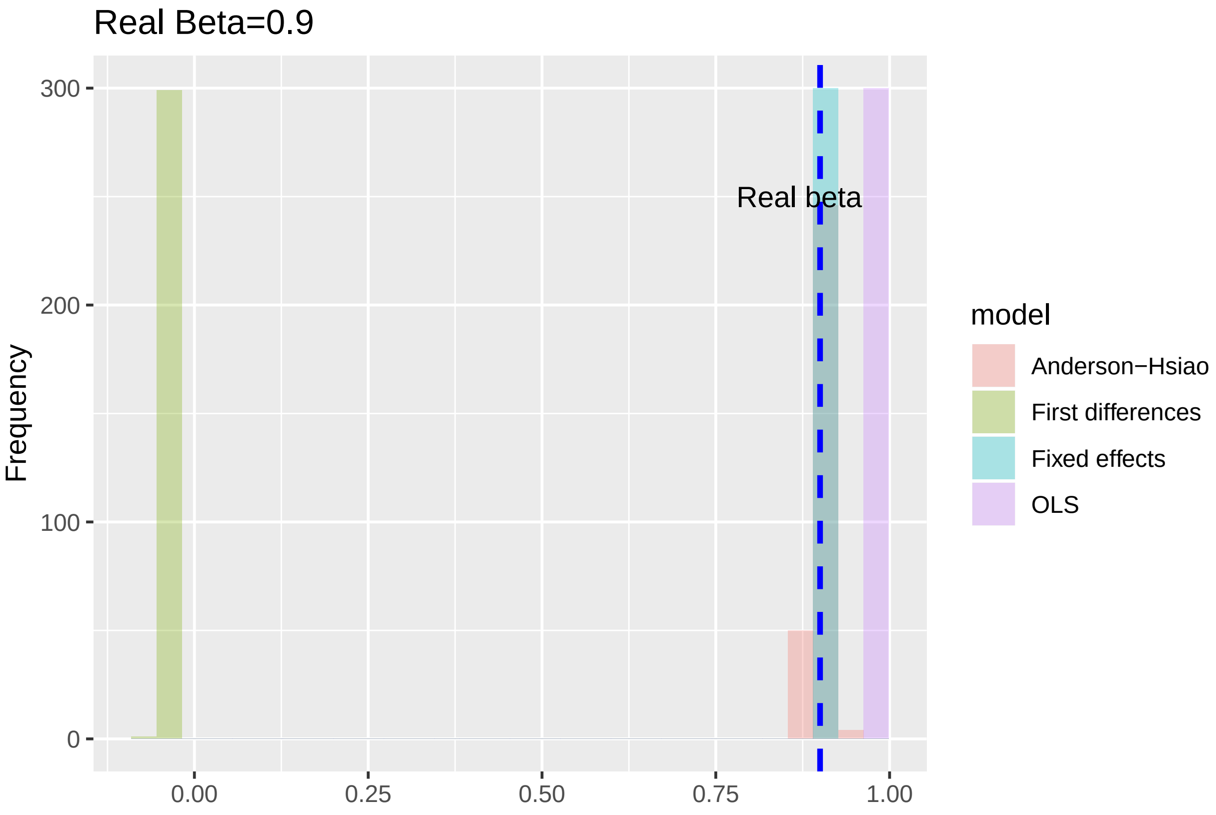
Table 3: MC-results

	Beta=0.1	Beta=0.5	Beta=0.9
OLS	0.592	0.872	0.996
Ficed effects	0.098	0.497	0.896
First differences	-0.450	-0.250	-0.044
Anderson-Hsiao	0.100	0.500	0.900





Real Beta=0.5



Real Beta=0.9