Advanced Econometrics
Assignment 1

Tibo Lachaert
Nicolás Romero Díaz

June 25, 2024

1 OLS Estimation

- *Replicate OLS results for the following cigarette demand model*

$$\ln(C_{it}) = \alpha + \rho \ln(C_{i,t-1}) + \beta_1 \ln(P_{it}) + \beta_2 \ln(Pn_{it}) + \beta_3 \ln(Y_{it}) + \mu_{it} \tag{1}$$

- File: `OwnFunctions.R`
  - Line 4 in this file has the code for the `OLS_own` function, which manually implements OLS estimation.
- Implementation: `Case assignment - Dynamic panel data.R`
  - We present the results of our own implementation of the OLS estimator. Table 1 shows the results from running the code in this file until line 45.

	Coefs	Std. Dev	t-stats	p-values
$\ln(C_{i,t-1})$	0.969	0.006	157.669	0.000
$\ln(P_{it})$	-0.090	0.015	-6.183	0.000
$\ln(Pn_{it})$	0.024	0.013	1.827	0.068
$\ln(Y_{it})$	-0.031	0.006	-5.089	0.000

Table 1: OLS results from applying the `OLS_own` function.

2 Fixed Effects estimator

- *Write a Matlab/R function that implements the FE estimator*
- *Replicate the FE results reported by Baltagi*

2.1 Implementing the FE estimator

- File: `OwnFunctions.R`
  - Line 39 in this file has the code for the implementation of the Fixed Effects (FE) estimator. The function is called `FE_own`
- Implementation: `Case assignment - Dynamic panel data.R`
  - Running the code in the file above up to line 67, we obtain the estimates presented in Table 2

	Coefs	Std. Dev	t-stats	p-values
$\ln(C_{i,t-1})$	0.833	0.013	66.269	0.000
$\ln(P_{it})$	-0.299	0.024	-12.638	0.000
$\ln(Pn_{it})$	0.034	0.027	1.238	0.216
$\ln(Y_{it})$	0.100	0.024	4.200	0.000

Table 2: FE results from applying the `FE_own` function.

2.2 Simulating the properties of the FE estimator

- *Set up a Monte Carlo experiment to numerically illustrate the properties of the FE estimator in a dynamic setting*

$$y_{it} = \rho y_{i,t-1} + \varepsilon_{it} \tag{2}$$
$$\varepsilon \sim \mathcal{N}(0,1)$$

  - *Use a burn-in of 25 periods*
  - *Use different values of* $\rho = 0, 0.5, 0.9$
  - *Use all combinations of* $T = 4, 10, 20, 50$ *and* $N = 10, 100$

- Implementation: `Case assignment - Dynamic panel data.R`
  - Continuing to execute the code in this file until line 131 will implement a Monte Carlo simulation with the respective number of individuals (N) and time periods (T) for each of the ρ parameters.
  - Figure 1 summarizes the results obtained. For a better comparison, the y-axis shows the estimated values $\hat{\rho}$ centered around the analytical value, i.e. $\hat{\rho} - \rho$. This value is represented by a red line in every subplot. The blue circle indicates the results for the panel with $N = 10$, while the orange crosses indicate results for $N = 100$.
  - For fixed values of T, Figure 1 does not show any obvious difference in the estimations of $\hat{\rho}$ for values of $N = 10, 100$. On the other hand, we can see that increasing values of T result in closer estimations of the $\hat{\rho}$ parameter. This result was expected due to the analytical form of the Nickell bias, which is inversely proportional to T. As a result, the bias decreases with an increasing T. This bias is also shown graphically in Figure 1, since estimations for all T, N considered lie below the analytical value.

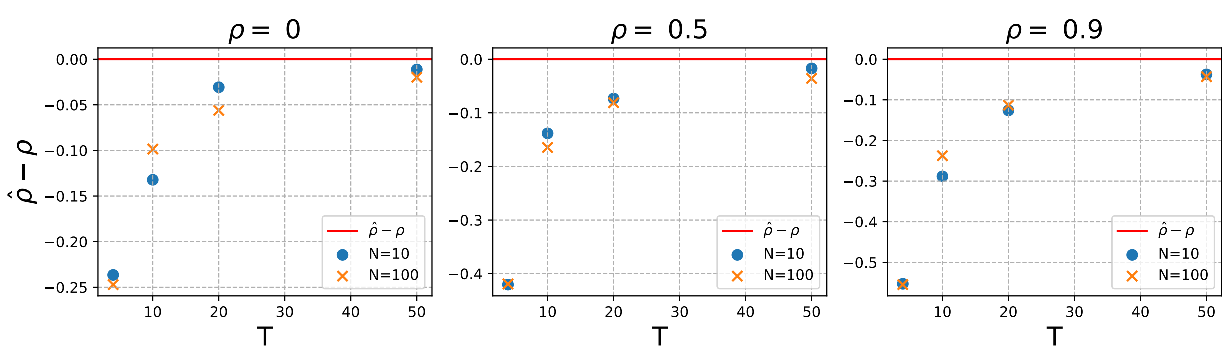

Figure 1: Results of Monte Carlo simulations of the FE estimator. Each subplot corresponds to different analytical values of ρ. The red line indicates the true value of the parameter. Blue dots indicate results for $N = 10$, while orange crosses present results for $N = 100$.

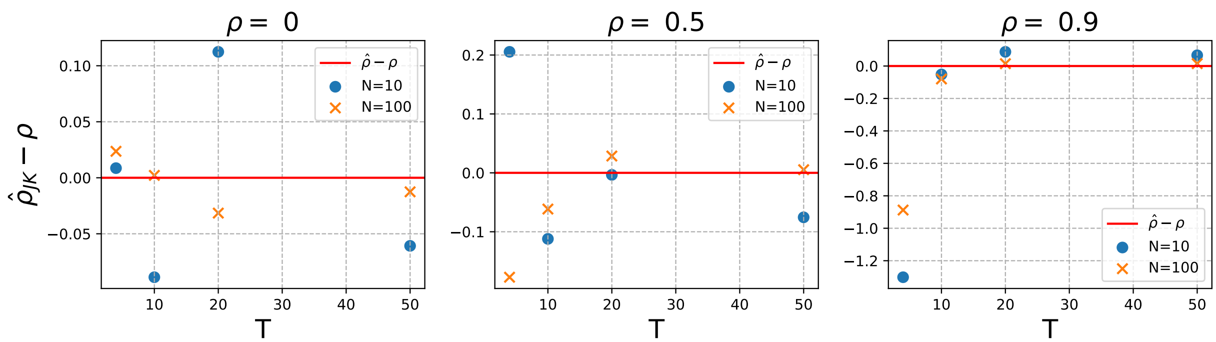

Figure 2: Results of the half-panel jackknife bias correction. Each subplot corresponds to different analytical values of ρ. The red line indicates the true value of the parameter. Blue dots indicate results for $N = 10$, while orange crosses present results for $N = 100$.

- COMPARE MEAN OF ESTIMATED STANDARD ERROR $\hat{se}(\hat{\rho}_{FE})$ TO TRUE STANDARD ERROR $se(\hat{\rho}_{FE})$
- COMPUTE THE SIZE OF A T-TEST FOR ρ BASED ON THE ANALYTICAL SE

2.3 Simulation methods for bias-correction and inference

In this section, we implement a half-panel jackknife to bias-correct a FE estimator.

- *Implement a half-panel jackknife bias-corrected FE estimator*
- *Use the Monte Carlo setup to compare the mean of* $\hat{\rho}_{FE}$ *to the population parameter* ρ
- Implementation: `jackknife_bias_correction.R`
  - The code in this file starts with the function `generate_ar1_panel`, which generates an AR1 panel dataset as described by equation (2).
  - In line 42, the function `create_half_panel` splits a panel into two halves across the time dimension, under the condition that the total time T is even.
  - Line 118 defines the `block_bootstrap` function, which ensures that the resampling procedure in the bootstrap does not interfere with the persistence pattern ρ in the AR1 process.
  - Line 158 applies the jackknife procedure for the values of ρ, T, and N given in question 2.2
  - Figure 2 presents the results for the bias-corrected estimator. This figure reproduces the presentation of results for question 2.2, yet, in this case, there is no apparent pattern in the estimated $\hat{\rho}$ as N and T change. We can see that in several occasions, the jackknife procedure results in an over-correction of the estimated persistence parameter.
  - To better compare between the non-corrected FE estimator and the bias-corrected version, we present a histogram of the difference between the estimated values and the true parameter $(\hat{\rho} - \rho)$. The distribution of this error is presented in Figure 3. We can see that the distribution of the bias-corrected estimator is less skewed than for the vanilla FE. Other than 2 outliers, the jackknife procedure generates a relatively symmetric distribution around the analytical value of ρ.

- *Compare the mean of the bootstrapped standard error* $se^b\left(\hat{\rho}_{FE_j}\right)$ to the true standard error $se\left(\hat{\rho}_{FE_j}\right)$
- Implementation: `jackknife_bias_correction.R`
  - In our case, the mean of the bootstrapped se and the true se are practically equal, with each value being $se^b\left(\hat{\rho}_{FE_j}\right) = 0.0577$ and $se\left(\hat{\rho}_{FE_j}\right) = 0.0579$. These results are calculated in lines 213 and 216 of the file denoted above.

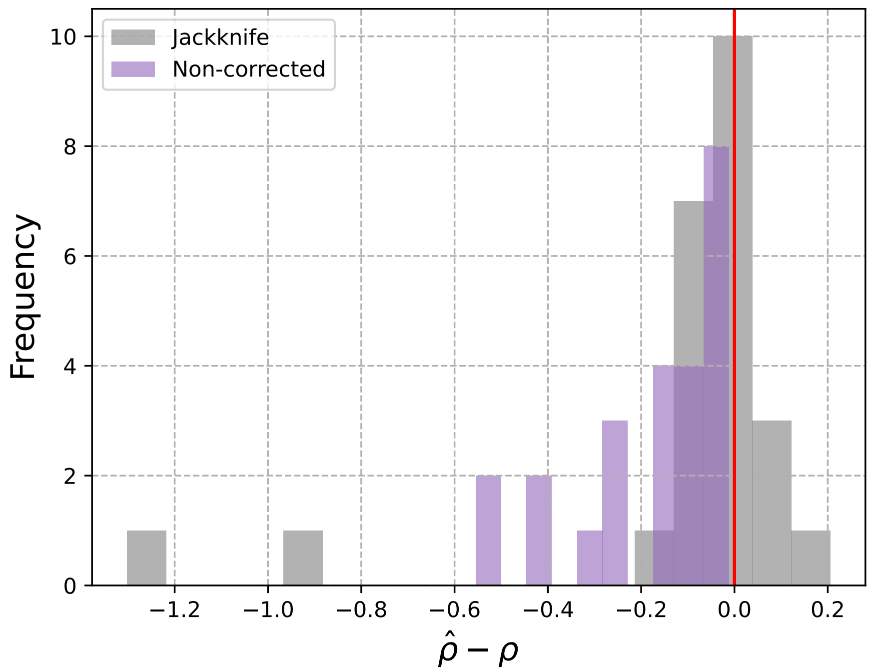

Figure 3: Distribution of the estimated $\hat{\rho}$ for jackknife (grey) and non-corrected FE (purple). The distribution is centered around the analytical value of ρ, which translates all estimated values around zero.

3 Implementing the panel GMM estimator

- *Write an R function that implements the GMM estimator*
- *Results should include one-step GMM estimates, t-stats, p-values*
- *For overidentified models: Two-step GMM estimates, t-stats, p-values, Sargan/Hansen test*
- Implementation: `Case assignment - Dynamic panel data.R`