Dingel Assignment 2 2020

Chase Abram

October 26, 2020

1 Table 1

Table 1: Table 1

	reg	xtreg	areg	reghdfe
	(1)	(2)	(3)	(4)
	log(Flows)	log(Flows)	log(Flows)	log(Flows)
log(Distance)	-1.658***	-1.658***	-1.658***	-1.658***
	(0.00875)	(0.00875)	(0.00875)	(0.00875)
Contiguous countries (binary)	0.974***	0.974***	0.974***	0.974***
	(0.0396)	(0.0396)	(0.0396)	(0.0396)
Common language (binary)	0.906***	0.906***	0.906***	0.906***
	(0.0184)	(0.0184)	(0.0184)	(0.0184)
N	156248	156248	156248	156178
R^2	0.718	0.586	0.718	0.718
Adj. R2			0.713	
Time (sec)	169.886	53.068	40.339	2.248

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

1. Are the point estimates and standard errors numerically identical across the different estimators? Should they be?

   ....

   They are identical and they should be. All the regressions above are econometrically identical, but we are just using different numerical implementations. For example, `reg` is simple, so we just fed it a bunch of dummies for the fixed effects, and it eventually figured things out. Both `xtreg` and `areg` can only directly do one-way fixed effects, so we fed them one of exporter-year or importer-year as the panel variables to be absorbed, and the other as a dummy. Finally, `reghdfe` can directly handle two-way fixed effects.

2. Are the number of observations and R-squared statistics identical? Should they be?

   ....

   The number of observations is only different for `reghdfe`. The R^2 is only different for `xtreg`. A priori, it's not obvious why all the estimations should not match on these fronts, but I have theories. For the number of observations, `reghdfe` is "smarter" than the other estimators, in that it is directly built for higher-dimensional fixed effects. My guess is that part of this construction is that it completes checks and occasionally omits observations that are judged to be problematic for the regression. The other techniques are probably too simplistic to consider the need to reject any of the observations. For the R^2, my guess is that something similar is going on, wherein `xtreg` is calculating standard errors based on some assumptions which don't actually hold when we force it to run two-way FE manually.

3. How do the relative computation times of these estimators depend on the dimensionality and size of the data?

   ....

   We don't really have many counterfactuals in terms of dimensionality and size of data, so it's a bit difficult to properly answer this question. However, it is quite clear that `reg` (the most naive) is the slowest of the bunch, and `reghdfe` (the most advanced) is the fastest, by an order of magnitude in either direction from the other two regressors.

2 Table 2

Table 2: Table 2

	reghdfe	reghdfe	reghdfe	ppml	poi2hdfe	ppml_panel_sg	ppmlhdfe	ppmlhdfe
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	log(Flows)	log(1+Flows)	log(1+Flows)	flow	flow	flow	flow	flow
main								
log(Distance)	-1.704***	-1.069***	-0.890***	-0.899	-0.899***	-0.899***	-0.899***	-0.899***
	(0.0136)	(0.00756)	(0.00652)	(.)	(0.0157)	(0.0157)	(0.0157)	(0.0157)
Contiguous countries (binary)	0.971***	1.017***	1.118***	0.462	0.462***	0.462***	0.462***	0.464***
	(0.0617)	(0.0343)	(0.0320)	(.)	(0.0377)	(0.0377)	(0.0377)	(0.0376)
Common language (binary)	0.975***	0.470***	0.433***	0.212	0.212***	0.212***	0.212***	0.211***
	(0.0286)	(0.0159)	(0.0132)	(.)	(0.0358)	(0.0358)	(0.0358)	(0.0357)
N	67350	67350	91652	91685	91652	91685	91652	67350
R^2	0.718	0.758	0.742	0.907		0.907		
Adj. R2	0.713	0.754	0.739					
Include zero flows?	No	No	Yes	Yes	Yes	Yes	Yes	No
log(flows+1)?	No	Yes	Yes	No	No	No	No	No
Approach	Log-Linear	Log-Linear	Log-Linear	PPML	PPML	PPML	PPML	PPML
Time (sec)	1.63	.963	.8280000000000001	1456.17	34.025	299.682	7.527	3.669

Standard errors in parentheses
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

1. Are your results sensitive to the omission of zeros?

   ....

   Yes, but only for the log-linear approaches. Notice the difference between columns (2) and (3), which only differ by the inclusion of zeros. We find an approximately sixteen percent difference in the estimate of how distance effects flows. If we instead compare columns (7) and (8), our results barely change at all (certainly not enough that we care). So it seems that the log-linear approaches are quite sensitive to inclusion of zeros, whereas the PPML routines are more robust to the inclusion.

2. How well does making the dependent variable log(x+1) perform?

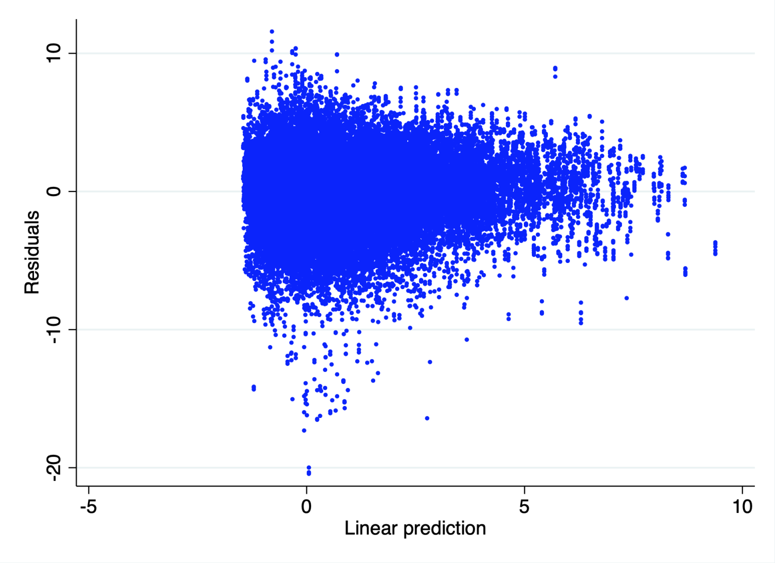
. . . .

It performs surprisingly well for the coefficient on distance, provided we include the zeros (hence the whole reason we would consider $\log(1 + x)$ over $\log(x)$ at all). However, for the other coefficients we are nowhere near the PPML estimates. So it's probably somewhat lucky that $\log(1 + x)$ worked well for the distance coefficient, and I would not expect this result to generalize.

3. Examine the residuals from your log-linear regression. Are they heteroskedastic? Report a Breusch–Pagan test statistic and a scatterplot of the residuals that addresses this question.

. . . .

The residuals seems to be quite heteroskedatic, hence why we use heteroskedastic-robust standard errors. The Breusch-Pagan test statistic is 8415.97, so deep into the rejection region for the null hypothesis of homoskedastic errors. This result is so prevalent, in fact, that even just a glance at the below scatter plot would convince most people that the variance in the errors appears to be decreasing as our predicted flows increase.



4. How do the computation times compare?

. . . .

Plain-vanilla `ppml` is the clear loser in a speed race. All the `reghdfe`s are the fastest, but of course it seems they are wrong, so fast but incorrect is not particularly helpful. The other three `ppml` estimators are all comparable[1], and it's again not surprising that the estimator designed for high-dimensional fixed effects was able to become the victor.

3 Table 3

1. Verify that reghdfe, FixedEffectModels, and fixest return identical estimates. Are the standard errors identical?

. . . .

[1] I think my computer goofed on `ppml_panel_sg`, based on conversations with my classmates, but I felt re-running the estimates (basically because they didn't look how I wanted) might be scientifically questionable.

Table 3: Table 3

	Stata	R	Julia
	(1)	(2)	(3)
	log(Flows)	log(Flows)	log(Flows)
log(Distance)	-1.325***	-1.325***	-1.325***
	(0.00361)	(0.004)	(0.004)
Contiguous countries (binary)	0.550***	0.550***	0.550***
	(0.0151)	(0.016)	(0.016)
Common language (binary)	0.762***	0.762***	0.762***
	(0.00731)	(0.008)	(0.008)
N	709248	709573	709248
R^2	0.702		0.702
Adj. R2	0.694	0.694	0.694
Time (sec)	7.403	0.472	2.625

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

The estimates are identical, and the standard errors are nearly identical, in that they only differ up to the third or fourth decimal place.

2. Which estimator is faster? By what magnitude?

. . . .

The `fixest` approach is the fastest by perhaps one order of magnitude (i.e. 10 times the time for `fixest` is somewhat close to the times for `R` and `Julia`). Practically, though (and perhaps I am being naive), it's not clear why we would deeply care about this difference. For example, even if I was running 100 of these regressions, it would take less than 15 minutes for the `Stata` approach, so it might well be that the trade-off for using a software which is more comfortable more than makes up for the time lost in the actual runtime. However, I don't mean to suggest this as a defense for `Stata`. I will suggest it as a defense for `Julia`, however, as `Julia` is already so diverse and useful for non-stats problems, and thus for a given project it might make sense to just also use `Julia` for the empirical component, rather than juggle across languages.