xtrobreg

A new command for pairwise difference robust panel estimators
... and more

Vincenzo Verardi
xtrobreg is joint work with Ben Jann

FNRS, Unamur, ULB

UK Stata users meeting, 2021

Introduction

The command **xtrobreg** was born to estimate **robust pairwise differences estimators** for panel data (see Aquaro and Cížek, 2013). It works by applying the **robreg** command (see Jann, 2021a) to appropriately transformed data. However, the command can be used for a wider range of situations

It can be used for pairwise based differences descriptive statistics

It can be used for pairwise based average descriptive statistics

It can be used for semiparametric models with a partially linear index

...

In this presentation we will start by presenting the **pairwise differences robust estimator** and then illustrate how the command can be used in different contexts.

Remark: The **code lines** presented in this talk are a **very first draft** and provided for **illustrative purpose** only. We advice to **revise them** thoroughly **before eventually using** them. Furthermore these **slides** have to be understood as being a **support for an oral presentation** were additionnal information has been provided. They should **not** be considered as a **stand alone document**.

Within-groups estimators

The **structure of this talk** is the following

1. Revision of the within transformation in Panel Data
2. Revision of difference estimators in Panel Data
   1. First-difference
   2. Within individuals pairwise difference
3. Robust Panel data regressions
   1. S-estimators and MM-estimators
   2. LTS-estimator and RLTS-estimatos
4. Pairwise based dispersion and location statistics
5. Pairwise based Logit and Poisson partial linear models (as stated by Honoré and Powell, 2005, the term entering non-linealy can represent "true nonlinearity" or may be the result of sample selection)
6. References

Plan of the talk

The linear **fixed effects** panel data model can be written as

$$y_{it} = \alpha_i + \mathbf{x}_{it}^{\mathrm{T}}\beta + \varepsilon_{it}, \quad i = 1, \ldots, n; t = 1, \ldots, T,$$

where y_{it} denotes the dependent variable, $\mathbf{x}_{it} = \left(x_{it}^{(1)}, \ldots, x_{it}^{(p)}\right)^{\mathrm{T}} \in \mathbb{R}^p$ contains observable covariates, and $\beta \in \mathbb{R}^p$ is the vector of the parameters of interest. Subscript i refers to individuals whereas t refers to time. The terms α_i are the unobservable time-invariant individual-specific effects.

These **individual effects** can be **partialled-out** by applying a simple linear (within groups) transformation:

1. compute the time averages for each individual,
   $\bar{y}_{i.} = \frac{1}{T}\sum_{t=1}^{T} y_{it}, \quad \overline{\boldsymbol{x}}_{i.} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_{it}$
2. subtract them from the original values: $\widetilde{y}_{it} = y_{it} - \bar{y}_{i.}$ and $\widetilde{\mathbf{x}}_{it} = \mathbf{x}_{it} - \overline{\mathbf{x}}_{i.}$
3. regress $\widetilde{y}_{it}$ on $\widetilde{\mathbf{x}}_{it}$ by least squares in the model
   $\widetilde{y}_{it} = \widetilde{\mathbf{x}}_{it}^{\mathrm{T}}\beta + \widetilde{\varepsilon}_{it}, \quad i = 1, \ldots, n; t = 1, \ldots, T$

Difference based estimators

Instead of using the mean centering to partial out the individual effects, a **first difference transformation** can be applied:

$$\underbrace{\Delta y_{it}}_{y_{it}-y_{i,t-1}} = \underbrace{\Delta \mathbf{x}_{it}^{\mathrm{T}}}_{\left(\mathbf{x}_{it}^{\mathrm{T}}-\mathbf{x}_{i,t-1}^{\mathrm{T}}\right)}\beta + \underbrace{\Delta \varepsilon_{it}}_{(\varepsilon_{it}-\varepsilon_{i,t-1})}$$

for $i = 1, \ldots, n$ and $t = 2, \ldots, T$. Under strict exogeneity assumption β is consistently estimated by LS.

Alternatively, Aquaro and Cížek (2013) propose to eliminate individual effects by taking all **pairwise differences within each individual** (for all $s = 1, ..., t-1$):

$$\underbrace{\Delta^s y_{it}}_{y_{it}-y_{i,t-s}} = \underbrace{\Delta^s \mathbf{x}_{it}^{\mathrm{T}}}_{\left(\mathbf{x}_{it}^{\mathrm{T}}-\mathbf{x}_{i,t-s}^{\mathrm{T}}\right)}\beta + \underbrace{\Delta^s \varepsilon_{it}}_{(\varepsilon_{it}-\varepsilon_{i,t-s})}$$

they then propose to fit the model using an LTS and a one-step reweighted RLTS estimator to get a robust estimator.

Pairwise-difference based estimators

In **xtrobreg**, the transformation advocated by Aquaro and Cížek (2013) is applied

$$\Delta^s y_{it} = \Delta^s \mathbf{x}_{it}^{\mathrm{T}} \beta + \Delta^s \varepsilon_{it}$$

and the **robreg** command is then used on the transformed data to get a robust estimation.

To deal with the **serial correlation** of the errors induced by the pairwise-difference transformation, **standard-errors**, estimated relying on the influence function, are **clustered** by default at the **cross-sectional identifier level** (similarly to what is done in Adams et al., 2019). An additional level of clustering is allowed.

The robust estimators available are those programmed in **robreg**.

xtrobreg restores the original data after estimation.

Alternatively, **xtrobreg convert** can be used to transform the data permanently and then apply **robreg** (or other commands) manually (typically including option `noconstant` and eventually using generated weights if the panel is unbalanced and clustering the S.E.).

Pairwise-difference based estimators

Stata example

```
.#delimit ;

.xtarsim y x1 eta, n(200) t(4) g(0) b(1) r(0) sn(5) seed(1234);

.est clear; drawnorm x2 x3; replace y=y+x2+x3;

.xtreg y x*, cluster(ivar) fe; est store FE_Clean;

.xtrobreg s y x*; est store PWDSE_Clean;

.xtrobreg mm y x*; est store PWDMME_Clean;

.replace x1=x1+10 if uniform()>0.9;

.xtreg y x*, cluster(ivar) fe; est store FE_Contaminated;

.xtrobreg s y x*; est store PWDS_Contaminated;

.xtrobreg mm y x*; est store PWDMM_Contaminated;

.estout *, drop(scale:  S: _cons) cells(b(star fmt(3)) t(par
fmt(2))) collabels(none) stats(hausman_p) style(tex); #delimit cr
```

Note: `xtarsim` is from Bruno (2005) and `estout` is from Jann (2021b)

Pairwise-difference based estimators

Stata example

	FE	PWD-S	PWD-MM	FE	PWD-S	PWD-MM
		Clean			Contaminated	
x1	0.997***	1.016***	0.993***	0.353***	0.999***	0.996***
	(51.87)	(29.87)	(51.14)	(11.46)	(38.84)	(47.10)
x2	0.956***	0.975***	0.958***	0.914***	0.997***	0.979***
	(23.15)	(11.98)	(21.72)	(9.45)	(16.67)	(21.22)
x3	0.950***	0.920***	0.920***	1.044***	0.865***	0.881***
	(19.56)	(12.79)	(19.58)	(11.92)	(13.50)	(17.64)
Hausman p		0.883	0.649		0.000	0.904

Finite Sample Performances of estimators

Simulations

To see the finite sample performances of **LS**, **M**, **S** and **MM** estimators applied to pairwise-difference data, we run some simple Montecarlo simulations using Bruno's (2005) `xtarsim` command.

Unbalancedeness is generated by randomly removing 10% of the observations (`drop if uniform()>0.9`) before an eventual contamination. We consider four setups.

1. #id=200, #t=3, no contamination

```
.xtarsim y x1 eta, n(200) t(3) g(0) b(1) r(0) sn(5) oneway(corr 5)
```

2. #id=200, #t=10, no contamination

```
.xtarsim y x1 eta, n(200) t(10) g(0) b(1) r(0) sn(5) oneway(corr 5)
```

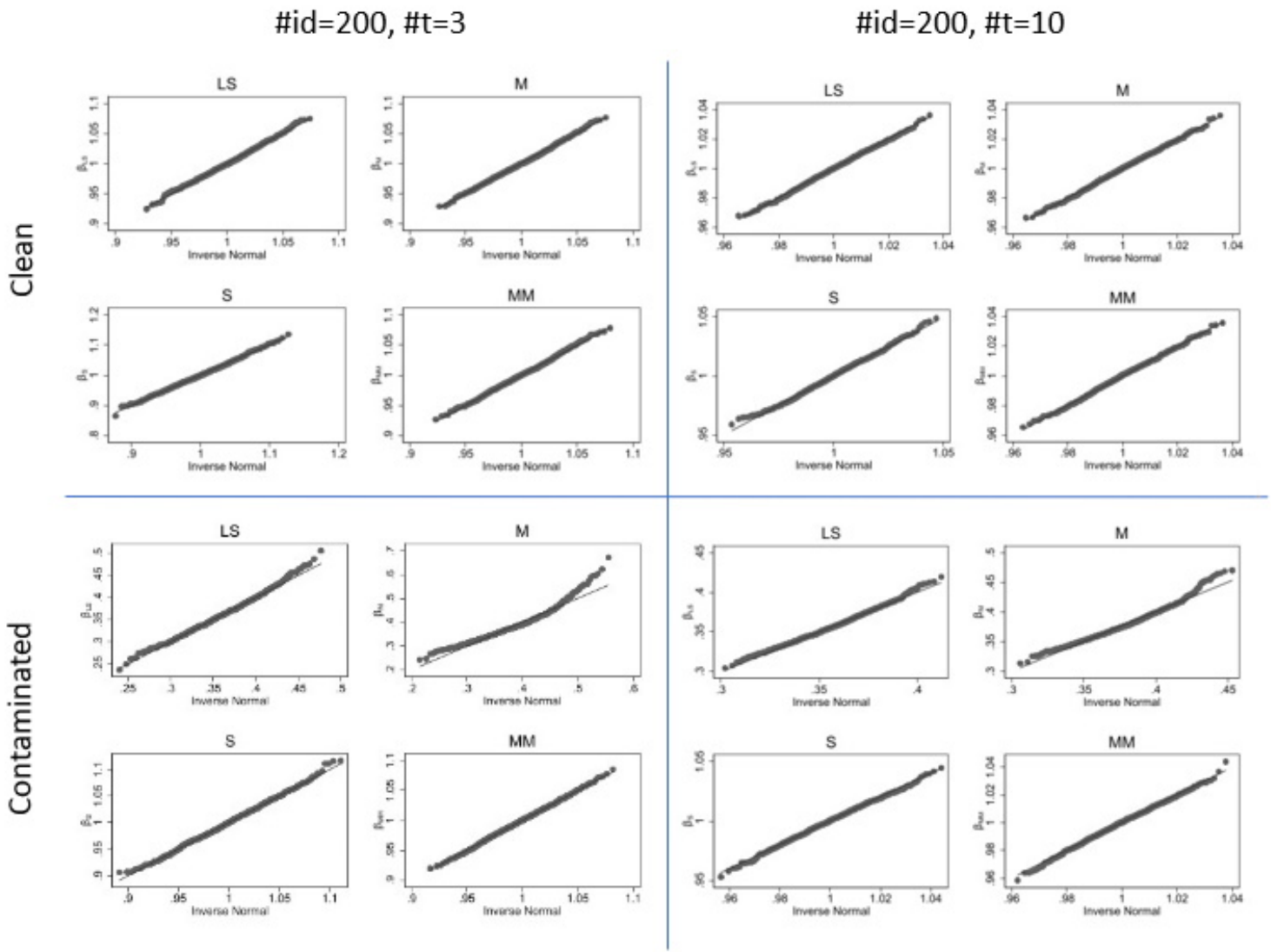
3. #id=200, #t=3, 10% contamination at x=x+10

```
.xtarsim y x1 eta, n(200) t(3) g(0) b(1) r(0) sn(5) oneway(corr 5)
.replace x1=x1+10 if uniform()>0.9
```

4. #id=200, #t=10, 10% contamination at x=x+10

```
.xtarsim y x1 eta, n(200) t(10) g(0) b(1) r(0) sn(5) oneway(corr 5)
.replace x1=x1+10 if uniform()>0.9
```

Normality of βs, qq-plots, 1000 simulations



Simulations results

Simulations: 1000

	n=200, T=3			n=200, T=10		
Clean	B	S.D.	mean S.E.	B	S.D.	mean S.E.
LS	1.00113	0.02369	0.02410	1.00010	0.01131	0.01120
M	1.00124	0.02416	0.02458	1.00007	0.01148	0.01132
S	1.00208	0.04047	0.04080	1.00006	0.01510	0.01484
MM	1.00139	0.02529	0.02583	1.00003	0.01180	0.01161
	n=200, T=3			n=200, T=10		
Contaminated	B	S.D.	mean S.E.	B	S.D.	mean S.E.
LS	0.35826	0.03827	0.03765	0.35715	0.01784	0.01838
M	0.38519	0.05523	0.04864	0.37942	0.02377	0.02272
S	1.00128	0.03549	0.03633	1.00038	0.01419	0.01385
MM	0.99943	0.02667	0.02753	0.99994	0.01222	0.01201

B: Mean estimated β

S.D.: Monte Carlo standard deviation of estimated β

mean S.E.: Mean of estimated standard error of estimated β

Aquaro and Cížek (2013) estimator

RLTS is an LTS estimator where the BDP is set to the number of identified outliers in the preliminary LTS estimator.

The **cutoff point** used to identify outliers is **data dependent** and is obtained by comparing two distribution functions: F^{+}_{nT} related to the estimated standardized absolute residuals (that is estimated relying on the data) and the F^{+}_{0} distribution function assumed for the error term (generally the normal).

The **maximum difference** between F^{+}_{nT} and F^{+}_{0} **in the tail of the distributions** can be measured by

$$\widehat{d}_{nT} = \sup_{v \geq \eta} \left\{ \left[F^{+}_{0}(v) - F^{+}_{nT}(v) \right] \cdot I\left(F^{+}_{0}(v) - F^{+}_{nT}(v) \geq 0 \right) \right\}$$

where η is a large quantile of F^{+}_{0}, for example, $\eta = 2.5$ for Gaussian errors (see Gervini and Yohai, 2002).

The cutoff point $\widehat{v}_{nT}$ is then defined as the $\left(1 - \widehat{d}_{nT}\right)$ th quantile of the distribution F^{+}_{nT}.

Aquaro and Cížek (2013) estimator

The **RLTS** estimator is obtained by estimating :

$$\widehat{\beta}^{(\mathrm{RLTS})} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{j=1}^{\widehat{h}_{nT}} r^2_{(j)}(\beta)$$

where

$$\widehat{h}_{nT} = \sum_{i=1}^{n} \sum_{t=1}^{T} I\left(\left| r_{it}\left(\widehat{\beta}_{nT}\right) / \widehat{\sigma}_{nT} \right| < \widehat{v}_{nT} \right)$$

Aquaro and Cížek (2013) prove that LTS and RLTS estimators in this context:

- **keep** their **equivariance properties**
- have **BDP**$\rightarrow 1/4$
- are **asymptotically normal** and provide their asymptotic variance

Aquaro and Cížek (2013) in Stata (balanced case)

The **RLTS** estimator:

```
. xtarsim y x eta, n(200) t(3) g(0) b(1) r(0) sn(9) oneway(corr 5)
. local T=3 ↑ Generate the data (balanced panel, no weights will be needed) ↑
. replace x=x+10 if uniform()>0.9 ← Generate ± 10% of outliers randomly located
. preserve
. xtrobreg convert y x*, clear ← Do pairwise differences
. robreg lts y x*, nocons ← Run LTS estimation (no weights are needed)
. predict res, rstandard
. gen ares=abs(res) ← Get absolute standardized residuals
. cumul ares, gen(Fnplus) ← Empirical CDF of absolute residuals
. gen Fplus=2*normal(ares)-1 ← Theoretical CDF of absolute residuals for normal errors
. gen d=(Fplus-Fnplus)*(ares>2.5)*(Fplus>Fnplus) ← Differences of CDFs in the tail
. qui sum d
. local lambda=max(1-r(max),0.5)*100 ← % of inliers. If set to 50, it become the LTS
. qui centile ares, cent(‘lambda’) ← Identify cut-off point
. gen inlier=(ares<=r(c_1))
. qui sum inlier
. local bp=max(1,100-r(mean)*100) ← alternatively local bp=max(1,100-‘lambda’)
. robreg lts y x*, nocons bp(‘bp’) ← LTS with new bdp
. predict subset, subset
. qui reg y x* [aweight=subset], nocons ← Same as LTS (note that s.e. are incorrect)
. predict resRLTS, res
. gen aresRLTS=abs(resRLTS)
```

Aquaro and Cížek (2013) in Stata (balanced case)

The asymptotic variance of **RLTS**:

```
. qui centile aresRLTS, cent(‘lambda’)
. local q=r(c_1)
. gen q=‘q’
. gen mq=-‘q’ ↓Simplifying assumption (homoskedasticity)↓
. kdens resRLTS, gen(fp) at(q) nograph k(g) ← Kernel density evaluated at q
. kdens resRLTS, gen(fm) at(mq) nograph k(g) ← Kernel density evaluated at -q
. gen f=fm+fp
. mata { ↓ Tentative Mata code to estimate the asymptotic variance (to be confirmed) ↓
. res=st_data(., "resRLTS") ; y=st_data(., "y"); N = rows(y); n=N/‘T’
. X=st_data(., "x*"); f=st_data(., "f") ; Id=diag(st_data(., "inlier"));
. e=Id:*res; Xe=cross(X,e); S=cross(Xe’,Xe’)/N ; Q=cross(X,diagonal(Id),X)/N
. J=-(‘q’*cross(X,f,X))/N ; V=(luinv(Q+J)*S*luinv(Q+J))/n; st_matrix("V",V)
. }
. program aquarocizek, eclass
.     tempname b V
.     mat ‘b’ = e(b)
.     mat ‘V’=V
.     eret repost b=‘b’ V=‘V’, rename ← Replace b and V of last estimation
.     est store RLTS
.     est replay RLTS
. end
. aquarocizek ← Run the program
```

Aquaro and Cížek (2013) in Stata

The **RLTS** estimator:

```
Model RLTS

      Source |       SS       df       MS      Number of obs   =       497
-------------+------------------------------   F(1, 496)       =   4231.24
       Model |  9614.58041     1  9614.58041   Prob > F        =    0.0000
    Residual |  1127.05248   496  2.27228322   R-squared       =    0.8951
-------------+------------------------------   Adj R-squared   =    0.8949
       Total |  10741.6329   497  21.6129434   Root MSE        =    1.5074

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
           x |    1.004118   .0306178    32.80   0.000     .9439618    1.064275
------------------------------------------------------------------------------
. restore
. xtrobreg ls y x*
converting data ... done

Pairwise-differences LS regression              Number of obs =       600
                                                Wald chi2(1)  =    224.70
                                                Prob > chi2   =    0.0000
                                  Scale               =  3.3844357
                                  Number of groups    =       200

                                (Std. err. adjusted for 200 clusters in ivar)
------------------------------------------------------------------------------
           y | Coefficient  Robust std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
           x |   .5431641   .0362349    14.99   0.000     .4717104    .6146178
------------------------------------------------------------------------------
```

Measure of dispersion

Alternative measures of dispersion

- Standard formula for the variance: $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$
- Pairwise alternative formulas for the variance are
  - $s^2 = \frac{1}{2n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2$ or
  - $s^2 = \frac{1}{2n_c}\sum_{i=1}^{n_c}\left\{(x_i - x_j)^2 ; i < j\right\}$ where $n_c = \begin{pmatrix} n \\ 2 \end{pmatrix}$
- Alternatively, one could use a robustified version (see e.g. Rousseeuw and Croux, 1993, citing Shamos, 1976 and Bickel and Lehmann, 1976) such as $1.0843 med\left\{|x_i - x_j| ; i < j\right\}$. However this estimator has a 29% breakdown point.
- Rousseeuw and Croux (1993) propose to use $Qn = 2.2219\left\{|x_i - x_j| ; i < j\right\}_{(k)}$ instead, which is the *kth* order statistic (with $k = n_c/4$) of the n_c inter-point distances, which has a 50% BDP (note that the latter is programmed more efficiently from a computational complexity viewpoint in the `robstat` command).

Dispersion

Alternative measures of dispersion

```
. clear
. qui set obs 1000
. set seed 1234
. drawnorm x
. gen id=1
. gen t=_n
. qui tsset id t
. qui xtrobreg convert x, clear
. gen ax=abs(x)
. qui sum ax, d
. local S0=1.0843*r(p50)
. local c=1/(sqrt(2)*invnorm(5/8))
. local Qn=`c´*r(p25)
. noi di `S0´
1.0625958
. noi di `Qn´
1.0286252
```

Measure of location

Alternative measures of location

Hodges and Lehmann (1963) introduced an alternative estimator of location to the mean and the median that has the advantage of being robust to outliers and being relatively efficient.

The Hodges-Lehmann estimator is defined by

$$\mathrm{HL}^{(n)} = \mathrm{med}_{i<j} \left\{ \frac{x_i + x_j}{2} \right\}.$$

The Hodges-Lehmann estimator is therefore the median of the arithmetic means of the pairs $i < j$.

For this estimator we hence need the **pairwise averages** rather than pairwise differences.

It would certainly be much more efficient to use the `robstat` command, but this can also be done relying on `xtrobreg`

Measure of location

Hodges and Lehmann location estimator

In this example we will show how to calculate pairwise means and subsequently get the HL estimator.

The real undercover objective of this slide is however to show how to link the original and transformed datasets using `xtrobreg` and `frames`.

```
.set obs 500
```

`.gen ivar=1` ← **Pairwise differences are done within the individual**

`.gen tvar=_n` ← **Pairwise differences are done over time**

```
.tsset ivar tvar
```

`.drawnorm x y z` ← **Generate the data**

`.gen t=tvar` ← **Create time indicator that will be needed for merging**

`.frame rename default dataset` ← **Rename the original dataset**

`.frame copy dataset difference` ← **Create a frame for transformed variables**

`.frame change difference` ← **Switch to the soon-to-be transformed dataset**

```
.drop t
```

`.xtrobreg convert x y z, clear t0(t)` ← **Transform the variables**

Measure of location

Hodges and Lehmann location estimator

```
.frlink m:1 ivar t, frame(dataset)
```
← **Link the datasets**

```
.frget x y z, from(dataset) suffix(_1)
```
← **Join datasets by ivar and t**

```
.foreach v in x y z {
.    replace `v'=0.5*`v'_1+0.5*(`v'_1+`v')
```
← **Calculate pairwise averages**

```
.}
.robstat x y z, stat(median) nose
```
← **Calculate the median pairwise average**

```
.frame change dataset
```
← **Switch to the original dataset**

```
.robstat x y z, stat(hl)
```
← **Compare to the efficiently programmed estimator**

```
. robstat x y z, stat(median) nose

Robust Statistics                       Number of obs = 124,750
```

median	Coefficient
x	-.0386149
y	-.0005888
z	-.0083099

```
. frame change dataset

. robstat x y z, stat(hl)

Robust Statistics                       Number of obs = 500
```

HL	Coefficient	Std. err.	[95% conf.	interval]
x	-.0386149	.0473029	-.1315523	.0543224
y	-.0005888	.0478467	-.0945947	.093417
z	-.0083099	.0466702	-.1000043	.0833845

Partially Linear Logit (from Honoré and Powell, 2005)

The logit model with fixed effects is given by $y_{it} = I\left\{\alpha_i + \mathbf{x}_{it}^{\mathrm{T}}\beta + \varepsilon_{it} \geq 0\right\}$, $t = 1,2; i = 1,\ldots,n$ where ε_{it} are i.i.d. logistically distributed. As stated (among others) by Honoré and Powell (2005), β can be estimated by maximizing the conditional log-likelihood

$$\mathscr{L}_n(\boldsymbol{b}) \equiv \sum_{i:y_{i1}\neq y_{i2}} \left\{-y_{i1}\ln\left(1+\exp\left(\left(\mathbf{x}_{i2}^{\mathrm{T}}-\mathbf{x}_{i1}^{\mathrm{T}}\right)\boldsymbol{b}\right)\right) - y_{i2}\ln\left(1+\exp\left(\left(\mathbf{x}_{i1}^{\mathrm{T}}-\mathbf{x}_{i2}^{\mathrm{T}}\right)\boldsymbol{b}\right)\right)\right\}$$

They then show that in the partially linear logit model (in a cross section) $y_i = I\left\{\mathbf{x}_i^{\mathrm{T}}\beta + g(w_i) + \varepsilon_i \geq 0\right\}, i = 1,\ldots,n$ for observations with w_i close to w_j, $g(w_i)$ and $g(w_j)$ are almost like fixed effects if g is smooth. They suggest estimating β by maximizing

$$L_n(\boldsymbol{b}) = c \sum_{\substack{i<j \\ y_i\neq y_j}} -K\left(\frac{w_i - w_j}{h_n}\right)\left(y_i\ln\left(1+\exp\left(\left(\mathbf{x}_j^{\mathrm{T}}-\mathbf{x}_i^{\mathrm{T}}\right)\boldsymbol{b}\right)\right) + y_j\ln\left(1+\exp\left(\left(\mathbf{x}_i^{\mathrm{T}}-\mathbf{x}_j^{\mathrm{T}}\right)\boldsymbol{b}\right)\right)\right)$$

where $c = \binom{n}{2}^{-1}\frac{1}{h_n}$, $K(\cdot)$ is a kernel which gives some weight to the pair (i,j), and h_n is a bandwidth that shrinks when n increases.

Partially Linear Logit - Stata code and simulations

Generate the data and run a standard logit neglecting the non-linear effect

```
.#delimit ; program drop _all; program mysim; frame reset;
.clear; local N=300; set obs ‘N’;
.drawnorm x w; replace w=0.5*w+0.5*x; gen e=rlogistic(0,1);
.gen ivar=1; gen tvar=_n; tsset ivar tvar;
.gen y=5*w^2+x+e>0; gen t=tvar;
.logit y x w; scalar b1 = _b[x]-1; ← Misspecified Logit
```

Create the **pairwise dataset**

```
.frame rename default dataset; frame copy dataset difference;
.frame change difference; drop t;
.xtrobreg convert y x w, clear t0(t);
.rename y dy; rename x dx; rename w dw;
.frlink m:1 ivar t, frame(dataset); frget y1=y, from(dataset);
.gen y2=dy+y1;
```

Partially Linear Logit - Stata code and simulations

Estimate the model (we focus on the point estimate, see Honoré and Powell, 2005, for a discussion on the asymptotic variance not programmed here)

```
.kdens dw, nograph kernel(epan); local bw=r(width); ← See Jann (2005)
.kernel dw, out(K) value(0) bw(‘bw’); ← See Chavez Juarez (2014)
.mlexp(-y1*ln(1+exp((dx)*{x}))-y2*ln(1+exp((-dx*{x})))) [iweight=K] if dy!=0;
.scalar b2 = _b[x:  _cons]-1; end;
```

Run simulations, estimate the **bias** and **MSE**

```
.simulate b1=b1 b2=b2 , reps(1000) seed(123):  mysim;
.sum b1; local Bias1=r(mean); local MSE1=r(mean)^2+r(Var);
.sum b2; local Bias2=r(mean); local MSE2=r(mean)^2+r(Var);
.matrix sim=((‘Bias1’,‘MSE1’)\(‘Bias2’,‘MSE2’));
.matrix colnames sim="Bias" "RMSE"; matrix rownames sim="Logit" "PL-Logit"
.matrix list sim; #delimit
```

	Bias	MSE
Logit	-.2186008	.0901421
PL-Logit	-.00264441	.05734509

Partially Linear Poisson (from Honoré and Powell, 2005)

Honoré and Powell (2005) extend the reasoning (among others) to partially linear Poisson regressions. In the Poisson regression model with fixed effects $y_{it} \sim \text{po}\left(exp\left(\alpha_i + \mathbf{x}_{it}^{\text{T}}\beta\right)\right) \quad t = 1,2; \quad i = 1,\ldots,n$ they state that the coefficients β can be estimated by maximizing:

$$\mathscr{L}_n(\boldsymbol{b}) = \sum_i \left\{ -y_{i1} \ln\left(1 + exp\left(\left(\mathbf{x}_{i2}^{\text{T}} - \mathbf{x}_{i1}^{\text{T}}\right)\boldsymbol{b}\right)\right) - y_{i2} \ln\left(1 + exp\left(\left(\mathbf{x}_{i1}^{\text{T}} - \mathbf{x}_{i2}^{\text{T}}\right)\boldsymbol{b}\right)\right)\right\}$$

As before, the coefficients for a partially linear Poisson regression model in a cross-sectionnal setup $y_i \sim \text{po}\left(exp\left(\mathbf{x}_i^{\text{T}}\beta + g\left(w_i\right)\right)\right) \quad i = 1,\ldots,n$ can be estimated by maximizing

$$L_n(\boldsymbol{b}) = c\sum_{i<j} K\left(\frac{w_i - w_j}{h_n}\right)\left(-y_i ln\left(1 + exp\left(\left(\mathbf{x}_j^{\text{T}} - \mathbf{x}_i^{\text{T}}\right)\boldsymbol{b}\right)\right) - y_j ln\left(1 + exp\left(\left(\mathbf{x}_i^{\text{T}} - \mathbf{x}_j^{\text{T}}\right)\boldsymbol{b}\right)\right)\right)$$

where $c = \begin{pmatrix} n \\ 2 \end{pmatrix}^{-1} \frac{1}{h_n}$, $K(\cdot)$ is a kernel which gives some weight to the pair (i,j), and h_n is a bandwidth that shrinks when n increases.

Partially Linear Poisson model

Generate the data

```
.frame reset
.#delimit ; clear; local N=500; set obs ‘N’; gen ivar=1; gen tvar=_n;
.tsset ivar tvar; drawnorm w x; replace w=0.5*w+0.5*x;
.generate y = rpoisson(exp(x+w^2));
.gen t=tvar; frame rename default dataset;
.frame copy dataset difference;
.frame change difference; drop t;
```

Create the **pairwise dataset**

```
.xtrobreg convert y x w, clear t0(t);
.rename y dy; rename x dx; rename w dw;
.frlink m:1 ivar t, frame(dataset); frget y1=y, from(dataset);
.gen y2=dy+y1;
.drop if _weight==.; drop _*; drop x*;
```

Partially Linear Poisson model

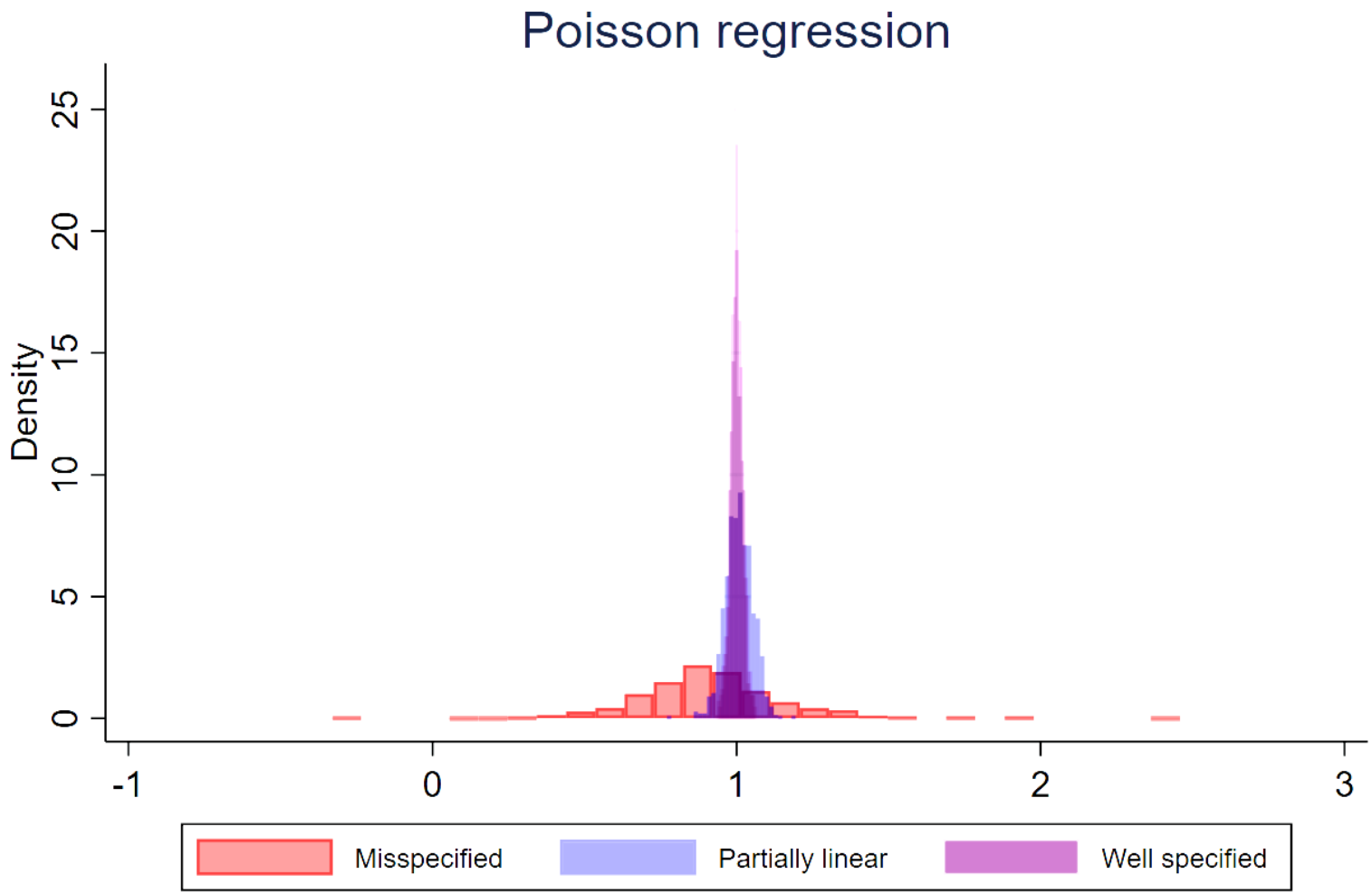
Estimate the model (we focus on the point estimate, the asymptotic variance is not programmed here)

```
.kdens dw, nograph kernel(epan); local bw=r(width); ← See Jann (2005)
.kernel dw, out(K) value(0) bw(‘bw’); ← See Chavez Juarez (2014)
.mlexp(-y1*ln(1+exp((dx)*{x}))-y2*ln(1+exp((-dx*{x})))) [iweight=K]; #delimit cr
```

```
. kdens dw, nograph kernel(epan);
(bandwidth = .08578862)
.  local bw=r(width);
.  kernel dw, out(K) value(0) bw(`bw');
  124,750
  18,775
Used kernel:        Epanechnikov
Kernel saved as:    K
Range of kernel:    [-.191829 |  .191829]
Observations:       124750
Obs. with K(z)>0:   18775
. mlexp(-y2*ln(1+exp((dx)*{x}))-y1*ln(1+exp((-dx*{x})))) [iweight=K];
initial:       log likelihood = -15816.219
...
Log pseudolikelihood = -13517.835                     Number of obs = 18,775
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
/x	1.011857	.0176088	57.46	0.000	.977344	1.046369

Some simulations (1000 replications, same setup)



Note: In the Misspecified Poisson regression model, w is introduced linearly

References and Stata command used

Adams, J., D. Hayunga, S. Mansi, D. Reeb, and Verardi V. (2019). Identifying and treating outliers in finance. Financial Management 48(2): 345-384.

Aquaro, M., and P. Cížek (2013). One-step robust estimation of fixed-effects panel data models. Computational Statistics and Data Analysis (57): 536-548.

Bickel, P. J., and Lehmann, E. L. (1976). Descriptive Statistics for Nonparametric Models IV: Spread in Contributions to Statistics, H*á*jek Memorial Volume, ed. J . Jurečov*á*, Prague: Academia, pp. 33-40.

Bruno, G. (2005). xtarsim: Stata module to perform Monte Carlo analysis for dynamic panel data models. Available from https://ideas.repec.org/c/boc/bocode/s453801.html. Statistical Software Components, Boston College Department of Economics.

Chavez Juarez, F. (2014). kernel: Stata module to compute various kernels. Available from http://ideas.repec.org/c/boc/bocode/s457869.html. Statistical Software Components, Boston College Department of Economics.

Gervini, D. and Yohai, V. (2002). A class of robust and fully efficient regression estimators. Annals of Statistics 30(2): 583-616.

References and Stata command used

Hodges, J. L., Jr., and Lehmann, E.L. (1963). Estimates of location based on rank tests. Annals of Mathematical Statistics 34(2): 598-611.

Jann, B., Verardi, V. and Vermandele, C. (2018). robstat: Stata module to estimate robust univariate statistics. Available from http://ideas.repec.org/c/boc/bocode/s458524.html. Statistical Software Components, Boston College Department of Economics.

Jann, B. (2021a). robreg: Stata module providing robust regression estimators. Available from http://ideas.repec.org/c/boc/bocode/s458931.html. Statistical Software Components, Boston College Department of Economics.

Jann, B. (2021b). estout: Stata module to make regression tables. Available from http://repec.org/bocode/e/estout. Statistical Software Components, Boston College Department of Economics.

Jann, B. (2005). kdens: Stata module for univariate kernel density estimation. Available from http://ideas.repec.org/c/boc/bocode/s456410.html. Statistical Software Components, Boston College Department of Economics.

References and Stata command used

Honoré, B. E., and Powell J.L. (2005). Pairwise Difference Estimators for Nonlinear Models. Identification and Inference for Econometric Models: Essays in Honor of Thomas Rothenberg, edited by Donald W. K. Andrews and James H. Editors Stock, 520–553. Cambridge University Press.

Rousseeuw, P.J and Croux, C. (1993) Alternatives to the Median Absolute Deviation, Journal of the American Statistical Association, 88:424, 1273-1283

Shamos, M. I. (1976), "Geometry and Statistics: Problems at the Interface," in New Directions and Recent Results in Algorithms and Complexity, ed. J. F. Traub, New York: Academic Press, pp. 251-280.