# Partial Identification of a Classical Measurement-Error Model

## TheoryGuru applications



---

## Load Economicreasoning package only if it is not already loaded

```
If[Length@Names["PLTools`*"] < 10,
 Get["http://economicreasoning.com"]]
```

### Load other tools by clicking on extras and/or evaluating below

```
If[Not@MemberQ[$ContextPath, "OtherTools`"],
 Get["http://othertools.economicreasoning.com"]]
```

---

## Background

### Some history of classical measurement error

This analysis originated with Frisch (1934). The classic measurement error model is a workhorse in applied econometrics, including Milton Friedman's Nobel Prize winning book Friedman (1957). See Tamer (2010) for a brief exposition of this model from the perspective of "partial identification."

### Dot products and vectors in *Mathematica*

Interpret all vector variables (x, y, $\epsilon$) as demeaned.

In the Wolfram Language, x.y refers to the tensor DOT PRODUCT, NOT scalar multiplication. For TheoryGuru purposes, tensor means vector, so that the result of x.y is a scalar.

```
? SymbolicRegression
```

SymbolicRegression[*depvar*, *indvar1*, *indvar2*, ...]
interprets each argument as a symbolic vector and returns the
formula, experessed in terms of dot products, for the least–squares coefficient vector.
The computation time and formula complexity is exponential in the number of regressors.
E.g., 8 regressors is about 426,000 times more complicated than 2.

# Setup

```
Clear[y, x];
```

```
OLSSlopeForward = First@SymbolicRegression[y, x]
```

$$\frac{x.y}{x.x}$$

```
OLSSlopeReverse = 1 / First@SymbolicRegression[x, y]
```

$$\frac{y.y}{y.x}$$

```
y = xtrue β + ϵy;
x  = xtrue + ϵx;
```

```
ClassicalMeasurementError = {xtrue.ϵx == 0, xtrue.ϵy == 0, ϵx.ϵy == 0};
```

```
YandxtrueAreCorrelated = xtrue.y ≠ 0;
```

```
xHasVariation = x.x > 0;
```

# Results

## Frisch’s identified set:
## the regression parameter $\beta$ is bounded by the forward and reverse regression slopes

```
TheoryGuru[{YandxtrueAreCorrelated, ClassicalMeasurementError},

 0 < OLSSlopeForward ≤ β ≤ OLSSlopeReverse
  ∨
  OLSSlopeReverse ≤ β ≤ OLSSlopeForward < 0]
```

```
True
```

### TheoryPossibilities discovers the hypothesis -- i.e., the above formula for the identified set -- on its own

```
DefineShortVariableNames = {for == OLSSlopeForward, rev == OLSSlopeReverse};
```

```
TheoryPossibilities[
 {YandxtrueAreCorrelated, ClassicalMeasurementError, DefineShortVariableNames},
 {β, for, rev}(* variables to appear in discovered formula *),
 True(* the variables can appear simultaneously *)]
```

```
You are in vector mode: include the hypothesis if it is
  necessary for classifying your assumption symbols as scalars vs. vectors.
```

```
(for < 0 && rev < 0 && β ≤ for && rev ≤ β && β < 0) ||
 (for > 0 && rev > 0 && for ≤ β && β ≤ rev && β > 0)
```

### Confirm that the two formulas above are equivalent

```
TheoryOverlap[{}, %,
 0 < for ≤ β ≤ rev
  ∨
  rev ≤ β ≤ for < 0]
```

```
{(for > 0 ∧ rev > 0 ∧ β > 0 ∧ for ≤ β ∧ β ≤ rev) ∨          are equivalent
  (for < 0 ∧ rev < 0 ∧ β < 0 ∧ rev ≤ β ∧ β ≤ for),
 (0 < for ∧ for ≤ β ∧ β ≤ rev) ∨ (for < 0 ∧ rev ≤ β ∧ β ≤ for)}
```

## The slope inferred from reverse regression has at least as much magnitude

```
TheoryGuru[{YandxtrueAreCorrelated, ClassicalMeasurementError},

 OLSSlopeReverse² ≥ OLSSlopeForward²]
```

```
True
```

```
ImperfectMeasurement = ϵx.ϵx > 0;
```

```
TheoryGuru[{xHasVariation, YandxtrueAreCorrelated,
  ClassicalMeasurementError, ImperfectMeasurement},

 OLSSlopeReverse² > OLSSlopeForward²]
```

```
True
```

## Note that the correlation and classical assumptions guarantee that *x* has variation

i.e., the forward regression will not divide by zero

```
TheoryGuru[{YandxtrueAreCorrelated, ClassicalMeasurementError},
 xHasVariation]
True
```

## Variable interpretations