Lab 1: from data to a first table

Your project, your data, your first regression

Magnus Lodefalk
Örebro University

Economics Research and Communication (NA310G)

22 October 2026



Today’s goals

By the end of today you can:

1. Set up a project that anyone can rerun, and check data before using them
2. Build an estimation sample and variables, including a merge you can trust
3. Run a first regression and read an AI dummy in a log outcome correctly

From last time

Answer on your own first, then compare with a neighbour

1. A panel is many short what?
2. Why do we cluster standard errors by firm?

How today works

- **I type in Stata and say why; you type the same** in your own do-files
- **Stops** like this slide: the few things you must not miss
- **After each block:** `do "$root/check_lab1.do"` says OK, or what differs
- **Lost?** The catch-up file puts you back in step; the full code is in `lab1-catchup/`
- **Today's question is a demo** nobody has in the report: AI adoption and financial expenses, chosen to show what the methods do. Your own question comes at the end

---

Typing it yourself is the point: by 15:00 your report has its skeleton.

The report in one slide

- **Your code** (six characters, on Blackboard) is your identity: cover page, file names
- **Your question:** one outcome, one industry group, one geography; the key regressor is `ai_firm`, 1 if the firm advertised for AI skills that year
- **Your data:** your own dataset, from the course site's "Your data" page
- **Real roots:** a twin of a research panel: Serrano firm accounts and Platsbanken job ads, the data we use in the lab

---

Today's demo is your report's skeleton, on another question.

Setup, step by step

1. **Unzip:** right-click the zip, *Extract All* (Mac: double-click). Never work inside the zip
2. **Put the folder in Documents**, not in OneDrive, Dropbox or iCloud: they can lock files
3. **Data:** copy the two course data files into `data/raw/`
4. **Double-click `master.do`:** Stata opens in your project folder and runs it
5. **Check your version:** type `display c(stata_version)`; you need 18 or newer

Hands up when master.do has run once. Then we start together.

Your project folder

```
na310g-starter/
   master.do          double-click it: runs everything, in order
   dofiles/0_settings.do   finds the folder, sets the version
   dofiles/1_load_check.do ... 4_regressions.do
   data/raw/          the data as you got them: never edit
   data/processed/    what your do-files save
   output/tables/, output/logs/, lab1-catchup/
```

Stop: raw data are the evidence. We never edit them; we save copies.

The do-file editor: five habits

- **Type in the do-file, never in the Command window:** if it is not in the do-file, it did not happen
- **Run a selection:** mark lines, then Ctrl+D (Mac: Cmd+Shift+D); after an error, run the whole file from the top
- **Comments:** `*` starts a comment line, `//` a comment after code, `///` continues a command on the next line
- **Help and look:** `help isid`; `browse` shows the data
- **Tab completes variable names:** type `bslov` and press Tab

Every time you open Stata: double-click master.do first. It sets `$root`, the project folder.

Words for today

- **Firm-year:** one row: one firm in one year. A **panel** is many firms over many years
- **Key:** the variables that identify a row (`firmID year`)
- **Merge:** put two files side by side by a key; **m:1** = many rows here match one row there
- **Log points:** a difference in $\ln y$; for a dummy, $100(e^b - 1)$ gives per cent
- **Clustered standard errors:** allow a firm's errors to be related across its years

Stop: one row per firm and year?

```
use "$root/data/raw/na310g_data_course.dta", clear
describe, short
isid firmID year
```

- `isid` says nothing when the key is fine, and stops with an error when it is not

---

Check the key first: if it is wrong, every step after it is wrong.

Stop: garbage in, garbage out

- 390 firm-years with zero employees: what kind of firm is that? (`browse if bslov_antanst == 0`)
- 96 firm-years with missing accounts
- Costs are stored with a minus sign: financial expenses, and salaries too (yet 129 firm-years have positive salaries)

Every dataset has these. Look with summarize, count if and browse, before Stata decides for you.

After block A

```
do "$root/check_lab1.do"
```

```
A: OK, ... firm-years, one row per firm and year.
```

- Not OK? Read the message; it says what to look at. Read the log: it is your lab notebook
- Lost? `do "$root/lab1-catchup/catchup_after_A.do"`, then go on

Your industry and your counties

```
gen division = floor(bransch_sni07 / 1000)
gen industry = .
replace industry = 1 if inrange(division, 10, 33)
drop if inlist(ser_laen, 8, 24)
```

- Type the manufacturing line; copy the other two groups from `lab1-catchup/2_sample_vars.do`
- Firms outside the three groups (energy, health, no code) get no group: by design

Stop: change one thing, run, look. Change three, and you will not know which broke it.

Stop: flip the sign, then log only positive numbers

```
gen finexp = -rr09_finkostn
gen ln_fin = ln(finexp)
label variable ln_fin "Log financial expenses"
count if missing(ln_fin)
```

- The log of zero or a negative number is missing: those rows leave the regression silently
- Your salaries outcome is a cost too: flip the sign the same way

In symbols

$\ln(0)$ and $\ln(-x)$ are undefined: Stata returns missing (.), not an error

How many firm-years fail to merge?

Your bet first: hands up. The answer is on the next slide.

A. Almost none
B. About a tenth
C. About a third

Stop: a merge you can trust

```
rename daioe_pctl daioe_avg
gen ssyk2012_4 = ssyk_main
merge m:1 ssyk2012_4 using ///
  "$root/data/raw/na310g_occupations_ssyk2012.dta", ///
  keepusing(daioe_pctl) keep(master match)
```

- `_merge`: 7,128 matched; 3,147 firm-years had no ads, so no main occupation
- Rename first: the data already hold a variable called `daioe_pctl`, and merge never overwrites it

Stop: how the sample was built

	Dropped	Left
Firm-years in the data		10,492
Kalmar and Västerbotten	217	10,275
No positive financial expenses	440	9,835
No employees	268	9,567
No capital (buildings, machinery)	745	8,822

The same table goes in your report, with a sentence on who drops out and why it may matter.

The descriptive table, straight to Word

```
dtable ln_fin ln_emp ln_cap ai_firm, ///
  continuous(, statistics(count mean p50 sd))
collect layout (var) (result)
collect export ///
  "$root/output/tables/descriptives.docx", replace
```

- In `3_descriptives.do`; the full lines are in the catch-up file
- Then `do "$root/check_lab1.do"`: `B: OK`? Then lunch

After lunch: double-click master.do first, or nothing finds its files.

Pooled OLS with year effects, clustered by firm

```
regress ln_fin ai_firm ln_emp ln_cap ///
    i.year, vce(cluster firmID)
```

```
     ai_firm |   .3007456    .156736     1.92   0.055
      ln_emp |   .4210508   .0643936     6.54   0.000
      ln_cap |   .4908312    .020946    23.43   0.000
```

AI firms have higher financial expenses than firms of the same size and capital, the same year. How much higher, and how sure are we? (A demo: not a finding about AI.)

Stop: an AI dummy in a log outcome

- The coefficient 0.301 is in log points, not per cent
- In per cent: AI firms' financial expenses are about 35.1 per cent higher
- A difference between firms, not an effect of AI: we have not shown that AI *caused* it

In symbols: key

$$\ln y = \beta_0 + \beta_1 D + \ldots \;\Rightarrow\; \%\Delta y = 100\,(e^{\beta_1} - 1) = 100\,(e^{0.301} - 1) \approx 35.1$$

Robust instead of clustered: the standard error is...

Your bet first: hands up. The answer is on the next slide.

A. About the same
B. Larger
C. Smaller

Robust or clustered: same coefficient

	Coefficient	Standard error	p-value
Robust	0.301	0.085	0.000
Clustered by firm	0.301	0.157	0.055

Clustering nearly doubles the standard error (1.9 times), and the result is no longer significant at 5 per cent. Clustering fixes standard errors, not bias. The same check I run in my own papers.

The regression table, straight to Word

```
collect create regtable
estimates restore clustered
quietly collect _r_b _r_se _r_p e(N) e(r2), ///
    tag(model[(1)] method[cl]): estimates replay
* ... model (2), labels, stars, layout, notes
collect export ///
  "$root/output/tables/regressions.docx", replace
```

- `collect`: Stata's own table system (`etable` and `dtable` are built on it)
- Paste the block from the catch-up file; change only the labels and the note

A table a journal would accept, with no number typed by hand.

The AI task: check what the AI wrote

- Open `ai_task_lab1.pdf` in your project folder; answers in Word or on paper
- A chatbot wrote the results paragraph for our regression; it has at least three mistakes
- Find them, say why, correct them, without a chatbot; 15 minutes, then we compare

AI writes fluent results paragraphs. Checking them is your job, and the exam's.

Now your own data and question

1. **Copy the whole folder** and name it `na310g-` plus your code: the demo stays intact
2. Download your data and question (course site, "Your data") into `data/raw/` of the copy
3. Today's goal: your data load, `isid` passes, your counties and industry are set, and `1_load_check` and `2_sample_vars` run

The self-check is for the demo: on your data it will complain. The rest comes in Lab 2.

Exit: with firm fixed effects, the AI coefficient...

Your bet first: hands up. The answer is on the next slide.

A. Gets larger
B. Stays about the same
C. Gets smaller

Summary

1. A project anyone can rerun: one folder, one master file, raw data never touched; check the key and know your data
2. A sample you can explain: industry and counties by rule, signs and logs handled, a merge read through `_merge`, and a table from data to sample
3. A first regression: clustered by firm, the AI dummy read in per cent, and as an association, not an effect

Test yourself before next time

Without AI

1. Why save a copy instead of changing the raw data file?
2. What does `_merge == 1` mean after `merge m:1`, and why did we keep those rows?
3. A dummy has coefficient 0.10 in a log outcome. What is the difference in per cent?
4. Why is the clustered standard error larger than the robust one here?