Seminar Corporate Governance:
Topics on Data Analysis with STATA

Yuhao Zhu

y.zhu@ese.eur.nl

22 November 2017

Contents

Part I
Introductory

1 Why we are here and how we get there?

Who am I?

- Yuhao (Hanan) Zhu.

- Final-year PhD Students at Erasmus School of Economics.

- Corporate Governance, Asset pricing, and Behavioral Finance.

Why this topic?

- Not knowing where to collect desired data.

- Lacking sufficient knowledge about STATA.

- Improperly resenting the results.

How to approach?

- Popular databases and what you can find from them.

- Basic knowledge of STATA commands, essential issues, and ability to read help documents.

- Design of the tables of results.

What to expect?

- Better data quality for your thesis.

- Correct results for your analysis.

- Clear way of showing your results.

Above all

You can improve your thesis grade by at least 1.0 if you do can reach these three criterion.

2 What to learn today?

Databases

- Popular databases.

- EDSC.

- WRDS.

STATA

- Layout.

- Idea behind STATA commands (functions). Link them to other computer languages.

- Basic commands for panel regressions.

- Essential issues and common mistakes.

- How to read STATA help document?

Tables

- How to get beautiful tables from STATA.

- How to get them into Excel?

- What items to show in your thesis?

- Title and captions.

Note

- Important sections and points are preceeded by an asterisk $*$.

- There are two kinds of questions: questions and good questions. So feel free to ask.

- I will also ask questions during the lecture.

Part II
Databases

3   Fantastic databases and how to find them

$*$ **Erasmus Data Service Center**

- When you do not know which databases meet your need, go to the website of EDSC.

- `https://www.eur.nl/ub/en/edsc/databases/financial_databases/`

Slide 12   **EDSC - Financial databases**

Financial databases

The Datateam provides access and supports a number of financial databases. If you already know the name of the database you want tot use, you can use the alphabetical list to get access.

You can also get a list of databases by subject (like stock prices, IPOs, CEO's) by entering keywords in the search box op this page. The titles, content descriptions and keywords of all UL-databases will be searched. Please note: you can't search in the databases.

There are **36** hits.
A - B - C - D - E - F - G - H - I - J - K - L - M - N - O - P - Q - R - S - T - U - V - W - X - Y - Z

 # Title

🔒 **Audit Analytics (via WRDS)**
Provides detailed audit information on over 1,200 accounting firms and 15,000 publicly registered companies. Read more

🔒 **Bloomberg Finance**
Bloomberg is the standard for financial data. Read more

🔒 **BoardEx**
BoardEx contains data from publicly listed companies about their board members. Read more

🔒 **Capital IQ People Intelligence (via WRDS)**
People Intelligence covers over 2.4 million people including private and public company executives, board members, and investment professionals, globally. Read more

🔒 **Compustat global (via WRDS)**
CompuStat Global consists of annual and quarterly report data of listed companies, with an

Slide 13   **Hand-collected data sets**

- Sometimes, hand collected data sets are also important.

- Not all data you need for your thesis is available through data sets.

- You can collect them by hand or by spiders.

4   ∗ WRDS

Slide 14   **WRDS**

- Wharton Research Data Services.

- `https://wrds-web.wharton.upenn.edu/wrds/`

- You can get permission to WRDS through EDSC.

Slide 15   **Databases**

- Compustat: Fundamentals of the firm.

- Execucomp: Executive compensation.

- CRSP; Stock prices.

- Event Study: A nice tool to do event study.

Part III
STATA

5   Why STATA?

Too many choices

- Excel (VBA): We know them when we are very young.

- SPSS: The first statistical software you met in your bachelor.

- EViews

- R

- Matlab

- Python

Advantages of STATA

- Ready-to-use packages and commands that are written and revised by previous researchers.

- Reusable codes and programs.

- Professional in handling panel data sets.

- In-design statistics.

Alternatives

- Excel (VBA and Python add-ins): Pre-process of the raw data sets. Some data sets are not clean enough for STATA.

- EViews: Professional in handling time-series analysis.

- R: Data visualization.

- Matlab: Simulations.

- Python: Spiders.

Job synergy

- Catch online data with Python.

- Pre-process and clean the data sets using Excel.

- Data visualization using R.

- Panel analysis using STATA.

6 STATA basics, commands, and do-files

Layout
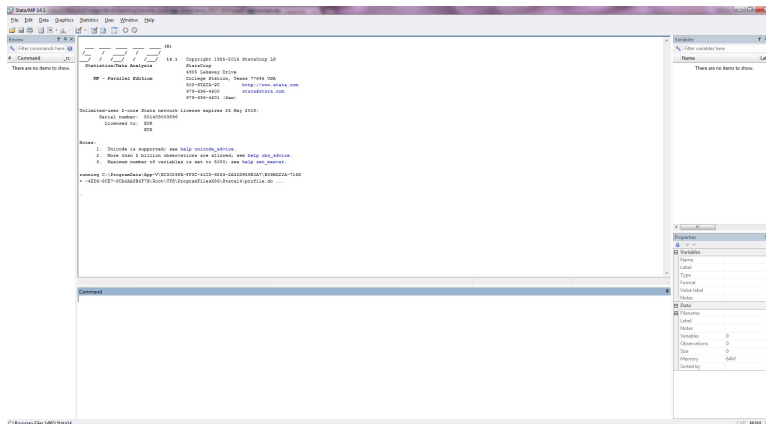
Windows

- ∗ Input window: you give command to STATA.
- ∗ Output window: STATA gives you results.
- Review window: commands you typed before. You can reuse them by clicking.
- Variables window: List of variables within the data set.
- Property window: Property of a certain variable, e.g., name, label, and data type.

Give commands

- There are always two ways to give command:
- Click buttons in the menu bars or ribbons (like what you do in SPSS).
- Type your command and press "Enter" (like what you do in CMD or Terminal).

Preferred way

- Typing command is preferred to clicking. Why?
- Clicking is annoying.
- Commands can be re-used later.
- We have "do-file".

Commands

- Commands are the most essential advantage of STATA.
- Clicking menu bars is first translated to commands, and then executed by STATA.
- Commands are well-defined functions!
- When you give commands, think that you are programming!

Idea behind commands

- Typed-in commands are conceptually equivalent to functions used in other computer languages.

- $Y = Function\,(X_1, X_2, X_3, \cdots \mid \theta)$.

- X_1, X_2, X_3 and so on are the input arguments (independent variables).

- θ is the parameters (optional variables).

- Y is the output (dependent variable).

- *Function* is well-defined sequential calculations and actions. Define once, and can be re-used many times.

Structure of the STATA commands

- `command [arg1 arg2 ...]  [if expression] [, options]`

- Command name.

- main argument.

- optional arguments.

- sample constraints.

- options.

- "Commands" are "Functions" without parenthesis.

An example: OLS estimate

- What if we calculate the OLS estimate by hand?

- Independent variable(s): Matrix X.

- Dependent variable: Y.

- OLS estimate $b = (X'X)^{-1} X'Y$.

- This formular can be defined as a function: `regress`

An example: STATA

- We type the command: `regress Y X`

- STATA analyzes your command:

- The function is `regress`. The OLS estimate should be used.

- The first argument is `Y`. It is the dependent variable.

- The second argument is `X`. It is the independent variable.

- STATA conducts calculation in the background: $b = (X'X)^{-1} X'Y$.

- b is printed in the output region.

An extended example: STATA

- We type the command: `regress Y X if year == 2000, vce(robust)`

- The function is `regress`. The OLS estimate should be used.

- The first argument is `Y`. It is the dependent variable.

- The second argument is `X`. It is the independent variable.

- STATA sees `if`. So the sample is constrained to observations with year equal to 2000.

- STATA sees `,`. So `vce(robust)` is option: using robust standard error.

∗ **The most important things to learn about STATA commands**

- The purpose of the command (function).

- The main arguments (variable) of the function.

- Which observations (sub-sample) are used?

- The options.

Example

- `summarize age income if gender == 1, detail`

- The purpose of the function is to summarize the variables.

- The variables we want to summarize are `age` and `income`.

- Which observations are used: males.

- We want to show more detailed summary: `, detail`

What is do-file?

- A sequence of commands just like a program.

- Automatically run from the beginning to the end. Or run the selected parts.

- Easy to re-use the codes.

- Ready to show to others with comments.

- Loops.

Always use do-files

- Always use do-files when you use STATA.

- `Ctrl + D` on PC, or `Shift + Cmd + D` on Mac to run selected commands.

7 ∗ Basic commands with Demo

Basic functions

- We will go through basic commands (functions) for panel analysis by:

- The idea behind commands.
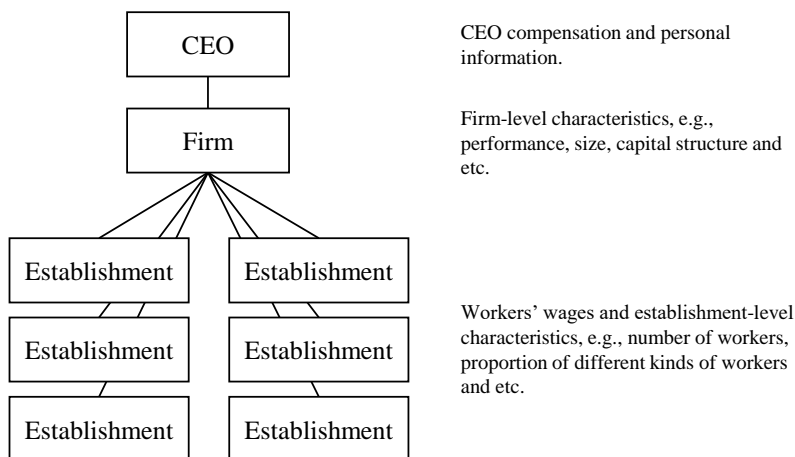
- The structure of a command.

- A real example.

Learning by doing

- Now we introduce the most basic commands in STATA. They are frequently utilized when doing corporate finance studies. Get familiar with them for your thesis.

- Example: a German panel data set.

- Working paper: *The real costs of CEO compensation: the effect of behindness aversion of employees.*

- Purpose: Relationship between CEO compensation and workers' pay.

Real example

- Data set 1: Firm-level information on 100 largest German firms (CEO compensation and performance)

- Data set 2: Branch-level (establishment-level) information (workers' pay and labor structure)

- Data set 3: Match book of the firm ID and the branch (establishment) ID.

Data structure

CEO compensation and personal information.

Firm-level characteristics, e.g., performance, size, capital structure and etc.

Workers' wages and establishment-level characteristics, e.g., number of workers, proportion of different kinds of workers and etc.

Organize your folder

- Organize your folder for better readability.

- A parent folder, and several sub-folders.

- /orig: Contains original data sets. Do not change data sets in this folder.

- /data: Contains modified or intermediate data sets saved for further analysis.

- /prog: Do files.

- Other sub-folders if needed.

Create a do-file

- We need to create a do-file.

- Save it under "analysis/prog/"

- Save the file in time.

Locate the path of the parent folder

- Locate your parent folder.

- For example, it is named "analysis".

- `cd "C:\Data\...\guest_lecture\analysis"`

Open a data set

- We begin with open a data set.

- `use "orig/firm_level_data.dta", clear`

Generate variables

- `generate ln_market_capital = ln(market_cap)`
  `gen return_on_asset = ebitda / total_asset`

Summarize

- Summarize the variables.

- Missing values are not summarized.

- With option `detail`, you can obtain more detailed descriptions including quantiles.

- `summarize ceo_total market_cap total_sales`
  `sum ceo_total, detail // Show more details.`

Correlation matrix

- You can create correlation matrix of many variables.

- `correlate ceo_total ceo_cash board_total`
  `corr market_cap total_sales employees`

Sort

- Sort by variable name(s).

- ```
  sort market_cap
  sort market_cap total_sales
  sort id_iab // Sort firm id.
  ```

Merge data sets

- Step 1: Find the common key(s)

- Step 2: Identify the matching mode: 1-to-1, m-to-1, or m-to-m.

- Step 3: Decide the master and the using data sets.

Command `joinby`

- Syntex: `joinby [varlist] using filename [, options]`

- ```
  sort id_iab // sort before joinby
  joinby id_iab using "orig/match_book.dta"
  ```

Command `merge`

- Syntex: `merge m:m varlist using filename [, options]`

- ```
  * Many-to-many matching
  merge m:m betnr year using "data.dta"
  * Keep only mathced observations
  keep if _merge==3
  * Drop the auto-created variable
  drop _merge
  ```

Rename variables

- You maybe want to rename variables to make them easily recognized.

- ```
  rename id_iab firm_id
  rename betnr branch_id
  ```

Handle duplicates

- Sometimes there are duplicates within sample.

- For example, for each branch and each year, there should be only one variable.

- But there are some times multiple values (mistakes during data collection, or just change of id).

- ```
  duplicates drop branch_id year, force
  ```

Generate dummy variables

- We need dummy variables for descriptive information.

- For example, we want to create a dummy indicating whether the union within the firm is "igmetall".

- ```
  * Generate a dummy with value 0.
  gen is_igmetall = 0
  * Change it to 1 under certain conditions.
  replace is_igmetall = 1 if union == "igmetall"
  ```

Generate fixed effects dummies

- If you want to do regressions with fixed effects, you need to create dummies.

- STATA create N new dummy variables for N groups.

- ```
  tabulate year, generate(year_fe_)
  tabulate id_iab, generate(firm_fe_)
  tabulate state_branch, generate(state_fe_)
  ```

- For example, year dummies: *year_fe_1*, *year_fe_2*, and so on.

Save modified data sets

- After modifications, we can save our data sets that are ready to be used for analysis.

- The option `replace` is important!

- ```
  save "data/panel_data_zhu.dta", replace
  ```

Begin regressions

- Now we can use the modified data sets to do regressions.

- ```
  clear all
  use data/panel_data_zhu.dta, clear
  ```

Declare data structure

- Before panel analysis, you need to declare the data structure.

- Time series, cross-sectional, or panel data set.

- For panel data set: `xtset [group] [time]`.

- For time series data set: `tsset [time]`.

- This enables time operators `l.year`, `f.year`, or `l2.year`.

- ```
  xtset branch_id year
  ```

OLS

- Example 1: Normal OLS with 1 explanatory variable.

- `regress ln_worker_wage ln_ceo_total`

OLS

- Example 2: Normal OLS with multiple explanatory variables.

- `reg ln_worker_wage ln_ceo_total ///`
  `return_on_asset leverage_ratio`

OLS

- Example 3: OLS when year is after 2006.

- `reg ln_worker_wage ln_ceo_total ///`
  `return_on_asset leverage_ratio ///`
  `if year >= 2006`

OLS

- Example 4: OLS when year is after 2006, using robust standard errors.

- `reg ln_worker_wage ln_ceo_total ///`
  `return_on_asset leverage_ratio ///`
  `if year >= 2006, vce(robust)`

OLS

- Example 5: OLS when year is after 2006, using robust standard erros, with firm fixed effects.

- `reg ln_worker_wage ln_ceo_total ///`
  `return_on_asset leverage_ratio ///`
  `firm_fe_* ///`
  `if year >= 2006, vce(robust)`

OLS

- Example 6: OLS when year is after 2006, using robust standard erros, with multiple fixed effects.

- `reg ln_worker_wage ln_ceo_total ///`
  `return_on_asset leverage_ratio ///`
  `firm_fe_* year_fe_* state_fe_* ///`
  `if year >= 2006, vce(robust)`

OLS

- Example 7: One-year lagged OLS when year is after 2006, using robust standard erros, with multiple fixed effects.

- ```
reg ln_worker_wage l.ln_ceo_total ///
return_on_asset leverage_ratio ///
firm_fe_* year_fe_* state_fe_* ///
if year >= 2006, vce(robust)
```

# 8 ∗ Essential issues and common mistakes

**Essential issues**

- Some issues are very essential when doing analysis or writing do-files with STATA.

- If you neglect these issues (or even ignore them), you may get error reports.

- Of course, your results are probably wrong! If your supervisor or co-reader detect this, then...

- Sometimes, remember these issues will save your a lot of time and energy. Time is money.

**Write comments**

- Comments start with *, //, or be blocked by /* and */.

- Sometimes you forget what you wrote yesterday. So use comments to remind yourself.

- If you want to show your do-file to others, comments may help.

**Example: comments**

```
* This is a single-line comment.
[commands]
// This is also a single-line comment.
[commands]
/*
This is a block of comments.
*/
```

Line break

- Sometimes it is hard to write everything in a single line.

- Use [///] (one space and three slashes) to break lines for your commands.

14

`merge` or `joinby`

- Sometimes you need to think which merging command to choose.

- `merge` and `joinby` should generate the same results.

- However, under certain circumstances, one is superior to the other.

- `joinby` creates all pair-wise matching. Suppose that you think you are doing 1-to-1, m-to-1, or 1-to-m matching. However, there is duplicate observations in your sample. `joinby` will not report his error, but `merge` does.

- `merge` is not encouraged to do m-to-m matching! The matching is unstable! You get different results every time you re-run this command.

Decision: `merge` or `joinby`

- `merge` for 1-to-1, m-to-1, or 1-to-m matching.

- `joinby` for m-to-m matching.

- Remember: I warned you...

Missing values

- Missing values should be treated carefully!

- Otherwise, you make mistakes.

Missing values: problem 1

- If most of the observations for `prop_female` are missing.

- You dropped missing values during data preparation. Compare the following codes:

- ```
  use data/panel_data_zhu.dta, clear
  regress ln_worker_wage ln_ceo_total
  ```

- ```
  use data/panel_data_zhu.dta, clear
  keep if prop_female == .
  regress ln_worker_wage ln_ceo_total
  ```

- Your sample size will shrink! You are measuring only local treatment effect.

- So, do not drop missing variable if the variable is a trivial one!. STATA automatically drop them during regressions.

∗ **Missing values: problem 2**

- I want to regress for branches where more than half of the workers are female. Compare the following codes:

- ```
  use data/panel_data_zhu.dta, clear
  keep if prop_female != .
  keep if prop_female > 0.5
  regress ln_worker_wage ln_ceo_total
  ```

15

- use data/panel_data_zhu.dta, clear
  ```
  regress ln_worker_wage ln_ceo_total ///
  if prop_female > 0.5
  ```

- use data/panel_data_zhu.dta, clear
  ```
  regress ln_worker_wage ln_ceo_total ///
  if prop_female > 0.5 & prop_female != .
  ```

- Wrong sample during regression! Why?

Slide 72 ∗ **Missing values are treated as very large numbers**

- Missing values . are treated as very large values in STATA. Compare the following codes:

- use data/panel_data_zhu.dta, clear
  ```
  keep if prop_female > 1
  count prop_female if prop_female == .
  ```

- use data/panel_data_zhu.dta, clear
  ```
  keep if prop_female < 1
  count prop_female if prop_female == .
  ```

- Do always treat missing values carefully!

Slide 73 **Fixed effects**

- What is fixed effects?

- How we add them into the regression?

- Fixed effects. i. or fe_*

Slide 74 **Do not use command `xtreg`**

- Many students love to use the command xtreg.

- Do not use it!

- use "orig/firm_level_data.dta", clear
  ```
  xtset id_iab year
  xtreg ceo_total employees, fe
  reg ceo_total employees i.id_iab
  ```

- Drawback 1: xtreg does not report overal R-square and adjusted R-square. Not comparable to OLS.

- Drawback 2: xtreg does not report coefficients for FE dummies.

- Drawback 3: It is hard to incorporate multiple fixed effects with xtreg.

- Drawback 4: *t-statistics* are more robust using FE dummies than using xtreg.

16

Fixed effect regressions: alternative methods

- Choice 1: create dummies and include them in the regression.

- `firm_fe_*` means all variables starting with `firm_fe_`.

- `reg ln_worker_wage ln_ceo_total ///`
  `firm_fe_* year_fe_* state_fe_*`

Fixed effect regressions: alternative methods

- Choice 2: use operator `i.`

- It only applies to numerical catagorical variables.

- `reg ln_worker_wage ln_ceo_total ///`
  `i.firm_id i.year i.state_branch`

- Following codes give error report:

- `reg ln_worker_wage ln_ceo_total i.union`

Operations within groups

- Sometimes we need to do operations only within groups.

- For example: generate increase rate for each firm.

- Common mistake: inter-group increase rate.

- Always be careful when you are handling lagged data.

Row identifier

- Cells can be located by variable name and row identifier `[_n]`.

- Handling lags with row identifier.

- Remember to sort before handling data.

- `sort id_iab year`
  `gen roa_increase = ///`
  `(roa[_n] - roa[_n-1]) / roa[_n-1]`

- What is the problem left? Inter-group increase rate!

- Drop them!

Alternative ways

- Method 1: Use `by` command to identify groups.

- `by id_iab: gen roa_increase = ///`
  `(roa[_n] - roa[_n-1]) / roa[_n-1]`

- Method 2: Use `xtset` command to specify the panel data set.

- Then use lagging operators.

- `xtset id_iab year`
  `gen roa_increase = (roa - l.roa) / l.roa`

by command

- `by` is equal to looping the command for each group.

- `by id_iab: sum roa`

- We can do this also by looping.

- `levelsof id_iab, local(id)`
  `foreach i of local id {`
  `display "id_iab == 'i'"`
  `sum roa if id_iab == 'i'`
  `}`

Third-party packages

- Some functions are programmed by third-parties.

- For example: IV regression, table output, winsorizing, and etc.

- Can be installed by commands: `ssc install [package]`.

- Be careful in selecting third-party packages, especially about historical versions.

Winsorizing

- Sometimes we need to winsorize the outliers.

- What is winsorizing?

- For the first time you use the package:

- `ssc install winsor`

- Now we conduct single-sided winsorizing to the observations whose value exceed 99% percentile.

- `winsor prop_female ///`
  `, generate(prop_female_winsor) p(0.01) highonly`
  `sum prop_female, detail`
  `sum prop_female_winsor, de`

Help with commands

- When you are not familiar with a new command, read the help documents.

- Type `help [command]`.

- Read syntax in the pop-out window.

- Click "[R] command – purpose of the command" in the pop-out window.

- Read more detailed descriptions in "STATA BASE REFERENCE MANUAL"

Learn to read documentations

- A typical STATA syntax goes as follows.

- `regress depvar [indepvars] [if] [in] [weight] [, options]`

- Also important: Description, Options, Stored results, and of course, Examples.

- Demo.

Part IV
Tables

9 Why tables are important?

Tables

- An organized way of showing your results.

- A good table (figure) is better than 1000 words.

Above all

We do not read your sentences that carefully. We read your tables!

10 ∗ A good table

Structure

- A good title.

- Well-written descriptive words (caption).

- A nice table with necessary statistics.

Title

- Numbering: Table 1, Table 2, and so on.

- The purpose of your table.

- Not too long.

- Example: "Summary statistics".

- Example: "CEO compensation and workers' pay: Baseline regressions".

Caption

- The caption should include all information about what you are testing, so that readers do not need to refer to the main context.

- You need to specify:

- The purpose of your regressions (what relationship you want to test).

- The methods you are using (OLS, IV regression, Difference-in-difference, Probit model).

- The dependent variable and independent variables.

- Special settings.

- How standard errors are treated or clusterd?

- How *t-statistics* and significant levels are expressed.

Caption: an example

This table presents results for regressions with the annual wage of employees as the dependent variable. All independent variables are lagged by one year. See Table 1 for a detailed overview of variable definitions. In specification (3), we consider the observations after 2006 only. We use the White (1980) robust standard errors clustered at firm level. The *t*-statistics are reported below the estimates. ***, ** and * indicate that the value is significantly different from zero at the 1%, 5% and 10% levels.
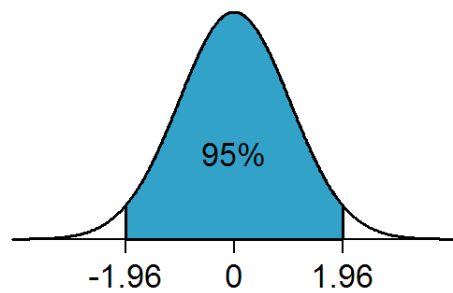
Table

- What information is necessary:

- Dependent variable and independent variables.

- Coefficients.

- *t-statistics*

- Significance level.

- R-square or adjusted R-square.

- Number of observations.

STATA table output

- The build-in table is not desirable.

- Only one regression per table. Hard to compare between models.

- Too long if fixed effects are included. We do not need them!

- Standard error, *t-statistics*, and *p-value*: Redundant.

- Confidence interval, *F-statistics*, Root MSE: we do not need them.

- Too long notes of collinearity.

t-statistics* and *p-value

- Reject the null hypothesis at the 95% level if t-statistics is located in the white region.

- To know about significant level, simply count stars.

We design our own table

- Use thrid-party package `estout` to generate concise and beautiful tables.

- Ready to be included in your thesis.

- No redundant output printed on your screen.

Command `estout`

- Install it for the first time.

- Use command `quietly` before `regress`

- Save statistics using `estout store [name]`

- Show the table with following commands.

- ```
estout m_*, ///
cells(b(star fmt(%9.3f)) t(fmt(%9.2f))) ///
style(fixed) title(Regression Results) ///
stats(r2_a N, fmt(%9.3f %9.0f)) ///
starlevels(* 0.10 ** 0.05 *** 0.01) ///
drop(*industry_fe* *year_fe* *state_fe*)
```

**Export STATA table, a demo**

- Copy as table.

- Modifications in Excel.

- Copy into Word / Powerpoint as pdf.

**Outcome**

| Dep. Var. | ln(workers' wage) | | | | | | |
|---|---|---|---|---|---|---|---|
| | model 1 | model 2 | model 3 | model 4 | model 5 | model 6 | model 7 |
| ln(CEO total) | -0.001 | -0.007 | -0.002 | -0.002 | -0.02 | -0.023* | |
| | -0.27 | -1.32 | -0.2 | -0.21 | -1.5 | -1.65 | |
| ln(CEO total) (t-1) | | | | | | | -0.023 |
| | | | | | | | -1.49 |
| ROA | | 0.033 | 0.079* | 0.079* | -0.092 | 0.026 | 0.032 |
| | | 0.93 | 1.73 | 1.74 | -0.97 | 0.27 | 0.28 |
| Leverage ratio | | 0.066*** | 0.117*** | 0.117*** | 0.053 | 0.093 | 0.111 |
| | | 2.64 | 3.32 | 3.29 | 0.44 | 0.76 | 0.78 |
| Market-to-book ratio | | | -0.005** | -0.005** | -0.004 | -0.001 | -0.013** |
| | | | -2.44 | -2.33 | -1 | -0.32 | -2.18 |
| ln(total sales) | | | -0.007 | -0.007 | -0.013 | -0.007 | -0.016 |
| | | | -1.6 | -1.61 | -0.52 | -0.26 | -0.57 |
| After 2006 | No | No | Yes | Yes | Yes | Yes | Yes |
| Robust std error | No | No | No | Yes | Yes | Yes | Yes |
| Firm fixed effects | No | No | No | No | Yes | Yes | Yes |
| Year fixed effects | No | No | No | No | No | Yes | Yes |
| State fixed effects | No | No | No | No | No | Yes | Yes |
| Constant | 9.897*** | 9.940*** | 9.888*** | 9.888*** | 10.067*** | 10.106*** | 10.340*** |
| | 132.36 | 129.94 | 95.01 | 97.69 | 45.74 | 43.85 | 22.38 |
| Adj. R square | 0 | 0 | 0.001 | 0.001 | 0.012 | 0.018 | 0.019 |
| Obs. | 16578 | 16439 | 13224 | 13224 | 13224 | 13224 | 9631 |

**Advantage of `estout`**

- All necessary information is included.

- Comparable between different models.

- Ready to be used in your thesis.

# Part V
# Conclusion

## 11 Concluding remarks

**What we learnt**

- Databases.

- STATA.

- Design your tables.

Slide 100    **Questions**

- Thank you for your attention.
- Please feel free to ask questions.