Applied Economics I

David Strömberg, Department of Economics, SU

Goals

- Provide a structure for conducting research projects aimed at
  - Replication
  - Efficiency
  - Accuracy: preventing errors
  - (Why is learning this difficult? Tacit knowledge.)
- Present tools
  - Programming languages: Stata, R, Python
  - Programs for backup, tracking changes (version control), text editors.
  - Machine Learning:
    - Unsupervised: clustering and principal component.
    - Supervised: model assessment and selection.
  - GIS
  - Text as data
  - Webscraping
- This is not a course in theoretical econometrics.
  - Evaluation by problem sets with hands-on exercises.

Overview

Lecture	Topic	Task
1	Workflow and project organization	Set up project and replicate AK (1991).
2	Stata, Documentation	Improve AK (1991) code.
3	Backup, Version Control.	Set up backup and version control systems.
4	ML: Unsupervised, Model Assessment and Selection	Unsupervised learning: World Values Survey.
5	ML: Linear model selection and regularization	Run two million growth regressions.
6	ML: Classification and text analysis	Classify US congressmen ideology by speeches.
7	ML: Tree-based methods	TBA
8	ML: Deep Learning	TBA
9	GIS	Map Swedish municipal borders and railways.
10	Python	Install + solve simple problem.
11	Web scraping with Python	Download and clean data.

Workflow and project organization

- Research Structure
  1. Project organization.
  2. Data management
  3. Programming practice.
  4. Backup and tracking changes.
  5. Documentation.

Replication/reproduction is a must

1. How you plan, document, write programs and save results should anticipate the need to replicate.
2. Minimal requirement for claiming a result.
   - Without replication, knowledge does not accumulate and veracity is questioned.
   - Growing concern: journals are requiring replication files, funding agencies are making requirements.
3. Increases efficiency and accuracy.

Why is replication hard?

1. The curse of dimensionality: research involves 1000’s of decisions that must be reproduced exactly.
    - What exact income measure to use.
    - How to treat missing values.
2. Missing, ambigous, obsolete files.

Some useful references

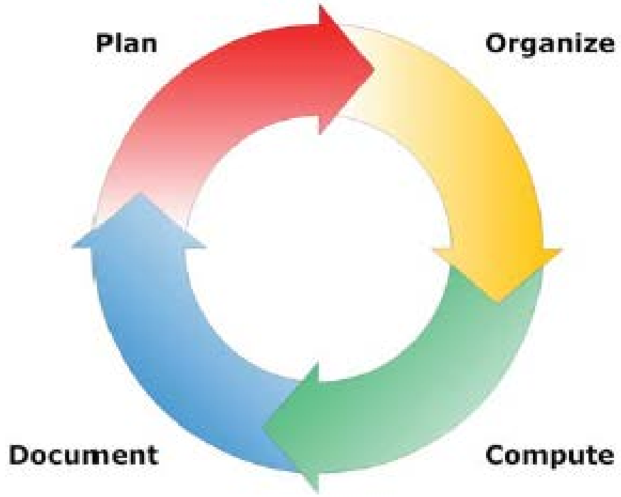
- * Gentzkow, Matthew, and Jesse M. Shapiro. "Code and data for the social sciences: A practitioner’s guide." *University of Chicago mimeo. Last updated January* (2014).
- Wilson, Greg, et al. "Good enough practices in scientific computing." *PLoS computational biology* 13.6 (2017): e1005510.
- Long, J. Scott, and J. Scott Long. *The workflow of data analysis using Stata*. College Station, TX: Stata Press, 2009.
- Bowers, Jake, and Maarten Voors. "HOW TO IMPROVE YOUR RELATIONSHIP WITH YOUR FUTURE SELF." *Revista de Ciencia Política* 36.3 (2016).

Workflow (from Scott Long's book)

- This is not a cook-book recipe.
- May help you to think through the project process.
  - Spend more time planning/organizing/documenting.
  - Get more things done.

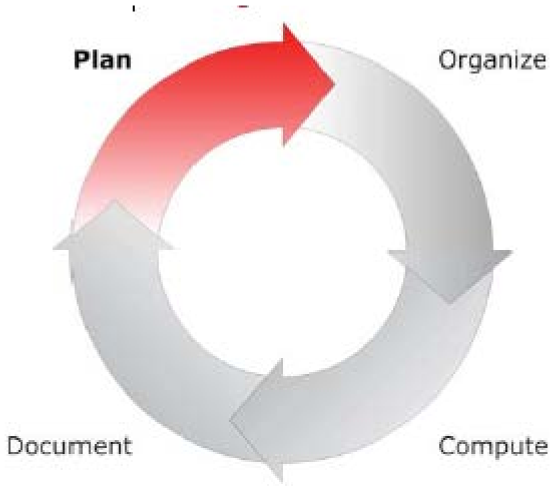
- Workflow steps
  1. Goals
  2. Data management.
  3. Data analysis.
  4. Presentation.

Tasks within each step



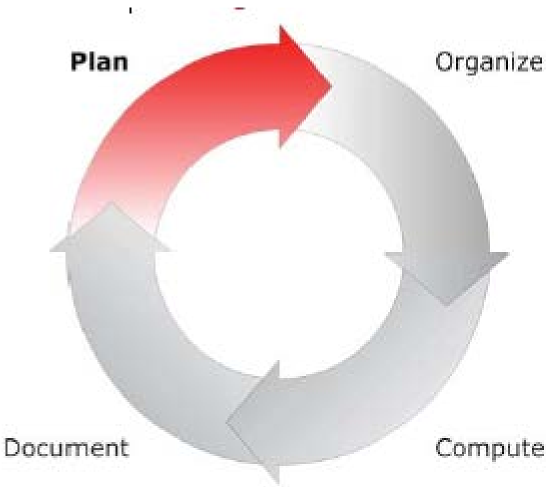
Plan

1. Goals.
   - Plan – is it worth is?
     - What do we want to learn?
     - What data, specification/method?
     - What is the addition to previous research?
     - How high can it fly, what journal?
     - Start by writing the abstract.
   - Plan how to do it.
     - What precise data and how to get it.
       - What y-variables, what x-variables?
     - What precise analysis.
       - What tables and graphs?
     - Timeline and division of labor.



Plan

2. Data management.
    - Plan
        - Timeline and division of labor.
        - Names and labels.
        - etc.



Organization

- Folder structure, file names, master file.
- Objectives
  1. Find things.
  2. Avoid duplication.
  3. Facilitate replication and collaboration.
- Signs of poor organization.
  - You **can’t find** a file and think you deleted it.
  - You and a colleague are **working on different versions of the same paper**. You changed what she changed an now you have three versions of the same paper.
  - You need the **final** version of the paper that was submitted for review, but you have two (or 16) files with ”final” in the name.

Organization: folder structure

- Rule 1: Every project has a project folder that is self-contained.
  - You can run all data preparation and analysis from raw data to results using files in this directory only. The project is portable.
  - Good for replication, collaboration, back-up, branching,…

Organization: folder structure

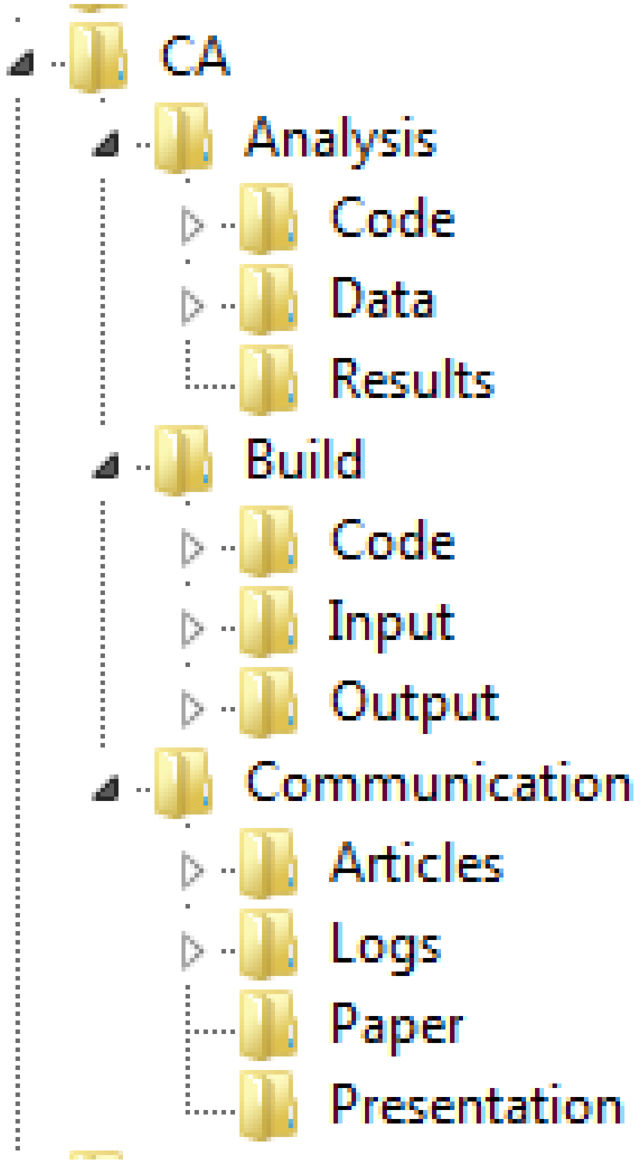
- Rule 2: Split data management and analysis.
  - Why?
    - Easy to redo the analysis without re-running the data build.
    - Analysis sparks creative energy that encourages poor data management.
    - Avoid duplication.
    - Helps identify and contain mistakes.

Separating data-manipulation and data-analysis is an example of modularity. ... The logic for this is simple. Lots of things can go wrong. You want to be able to isolate what went wrong. You also want to be able to isolate what went right. (Jonathan Nagler of NYU)

  - Are data management and analysis distinct?
    - No, but try to keep it separate.
    - The first analysis a paper of mine uses principal components to measure media bias and estimating a pca is analysis. This is used as input in other analysis.

A project of mine

I sometimes also include a Pre-filder that loads external raw data into the project.



- CA
  - Analysis
    - Code
    - Data
    - Results
  - Build
    - Code
    - Input
    - Output
  - Communication
    - Articles
    - Logs
    - Paper
    - Presentation

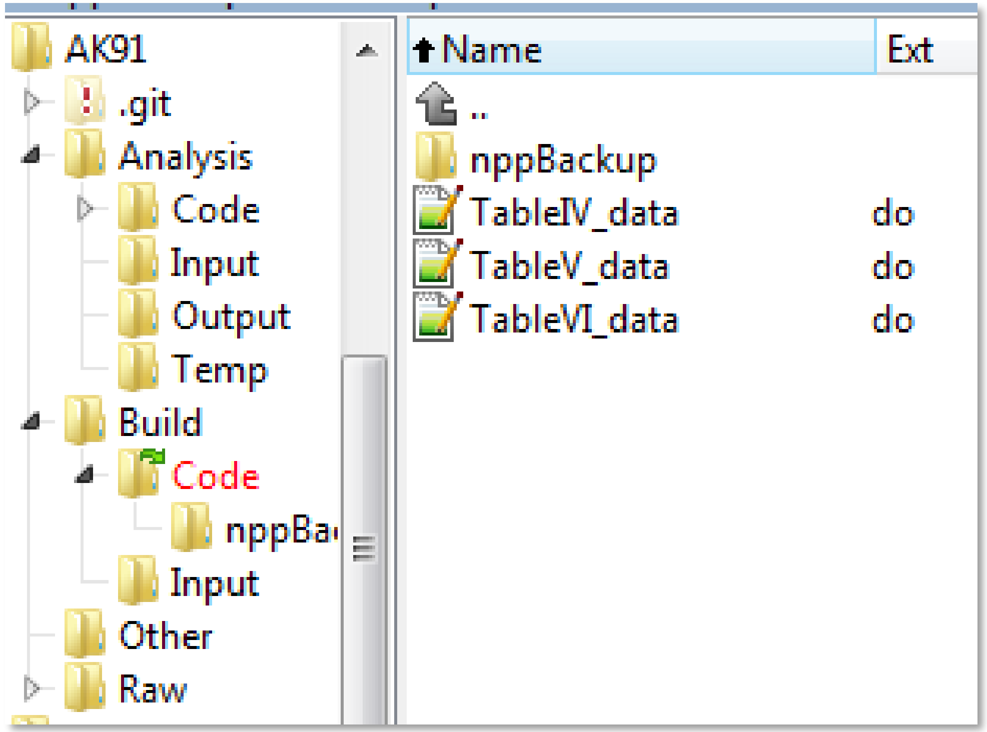
Names are critical

1. Names provide implicit documentation.
2. Make it possible to find things.

- Plan for names of main files and directories at the start of the project.
  - Should contain critical information, but not be too long.
  - Should be consistent within and across projects.
  - Don’t rename files or directories. This makes it difficult to trace changes.

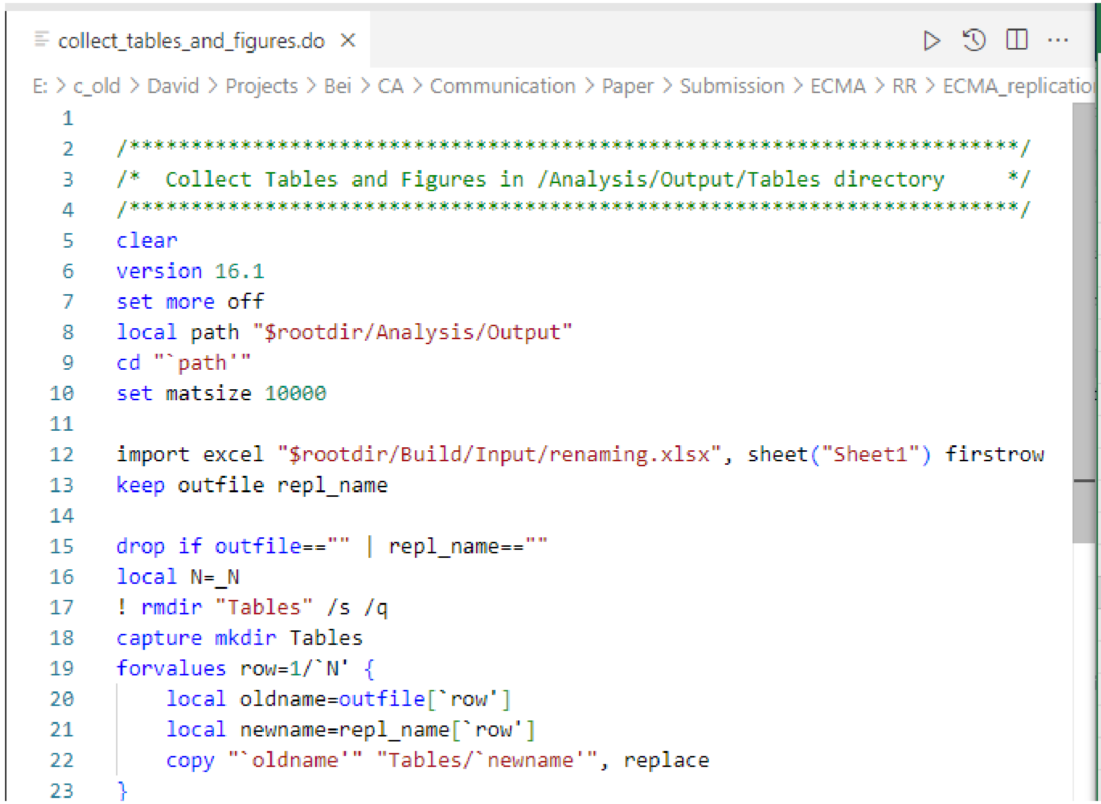
Names

- Project keyword (main folder name)
  - Pick mneumonic for your project: short and unique.
  - AK91 stands for Angrist and Krueger (1991).
- Dofiles
  - Try to name do-files in the build stage after the data set they create.
  - Don’t use names such as TableIV, since table numbers change all the time. (I just do this since I replicate a published paper.)
  - For replication: a separate do-file
    - Creates a new replication directory
    - Copies input data and dofiles.
    - Runs the master file and renames tables and figures.



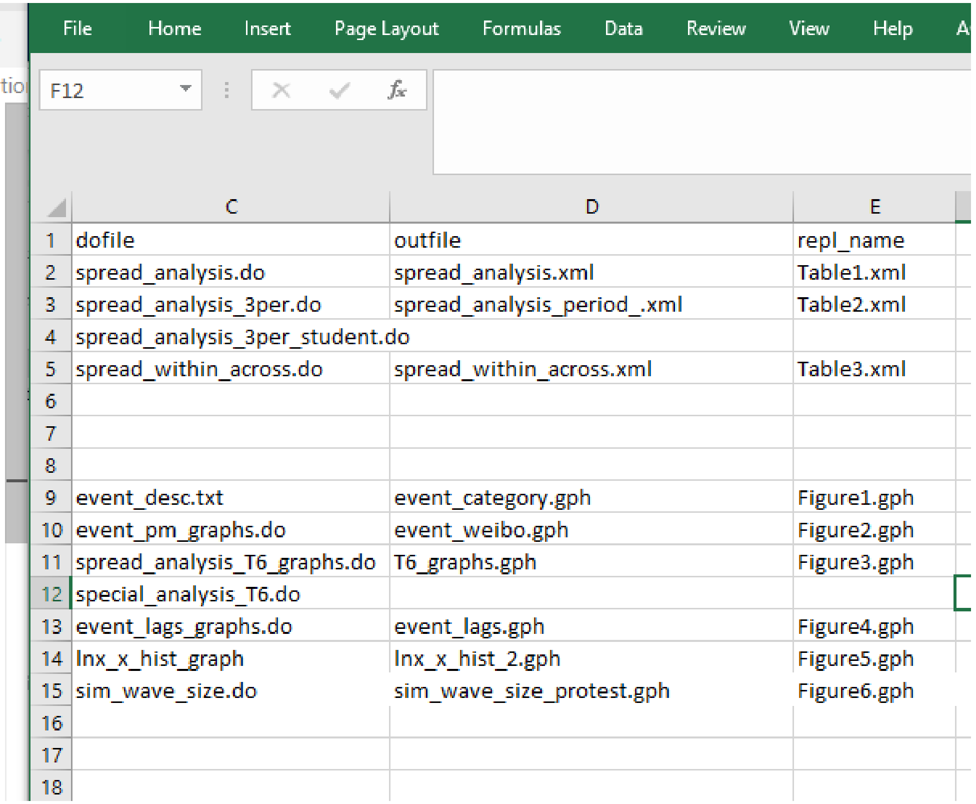
Name	Ext
..	
nppBackup	
TableIV_data	do
TableV_data	do
TableVI_data	do

Example: renaming output files.



E: > c_old > David > Projects > Bei > CA > Communication > Paper > Submission > ECMA > RR > ECMA_replicatio

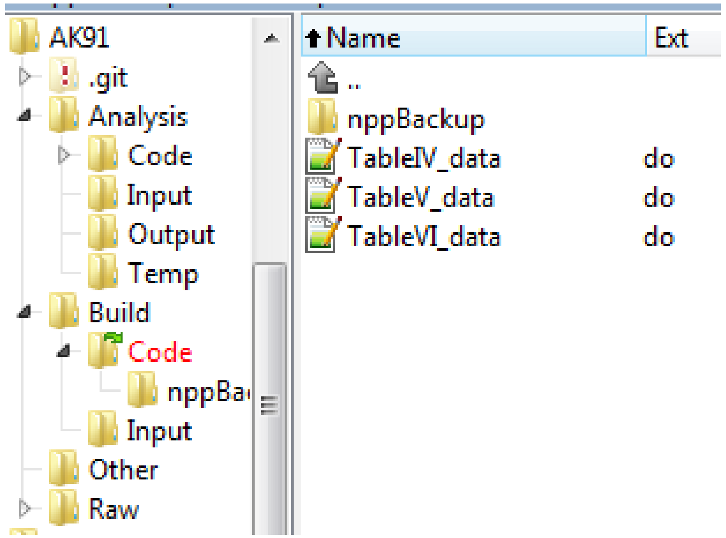
```

/*********************************************************************/
/*  Collect Tables and Figures in /Analysis/Output/Tables directory    */
/*********************************************************************/
clear
version 16.1
set more off
local path "$rootdir/Analysis/Output"
cd "`path'"
set matsize 10000

import excel "$rootdir/Build/Input/renaming.xlsx", sheet("Sheet1") firstrow
keep outfile repl_name

drop if outfile=="" | repl_name==""
local N=_N
! rmdir "Tables" /s /q
capture mkdir Tables
forvalues row=1/`N' {
    local oldname=outfile[`row']
    local newname=repl_name[`row']
    copy "`oldname'" "Tables/`newname'", replace
}
```



	C	D	E
1	dofile	outfile	repl_name
2	spread_analysis.do	spread_analysis.xml	Table1.xml
3	spread_analysis_3per.do	spread_analysis_period_.xml	Table2.xml
4	spread_analysis_3per_student.do		
5	spread_within_across.do	spread_within_across.xml	Table3.xml
6			
7			
8			
9	event_desc.txt	event_category.gph	Figure1.gph
10	event_pm_graphs.do	event_weibo.gph	Figure2.gph
11	spread_analysis_T6_graphs.do	T6_graphs.gph	Figure3.gph
12	special_analysis_T6.do		
13	event_lags_graphs.do	event_lags.gph	Figure4.gph
14	lnx_x_hist_graph	lnx_x_hist_2.gph	Figure5.gph
15	sim_wave_size.do	sim_wave_size_protest.gph	Figure6.gph
16			
17			
18			

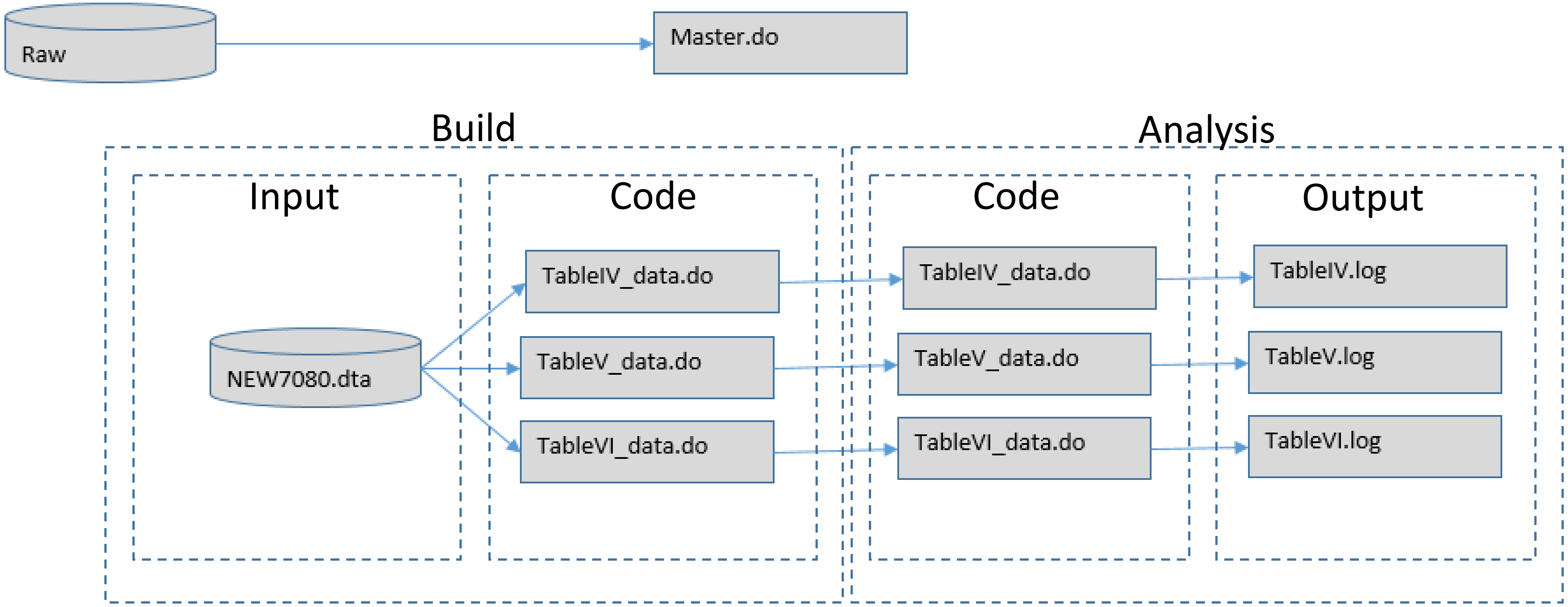
Project map

- Provides an overview, shows dependencies, simplifies finding files, updating and revisions, etc.



AK91
.git
Analysis
Code
Input
Output
Temp
Build
Code
nppBa
Input
Other
Raw

Name	Ext
..	
nppBackup	
TableIV_data	do
TableV_data	do
TableVI_data	do



Raw

Master.do

Build

Input

NEW7080.dta

Code

TableIV_data.do

TableV_data.do

TableVI_data.do

Analysis

Code

TableIV_data.do

TableV_data.do

TableVI_data.do

Output

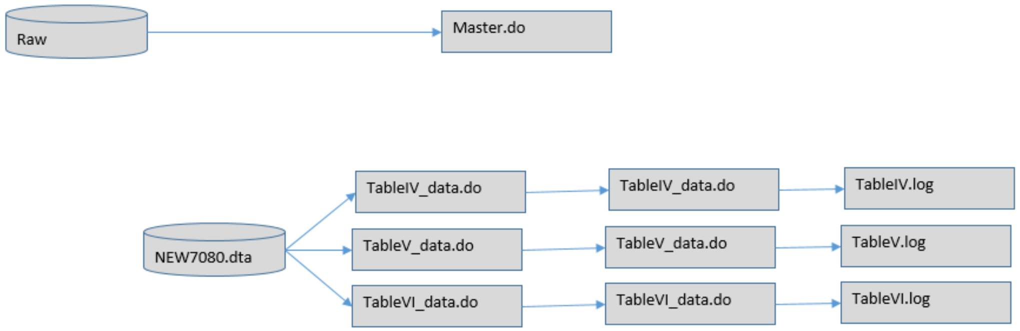
TableIV.log

TableV.log

TableVI.log

Creating a project map using Excel

1. Clear gridlines (View, tick off gridlines)
2. Add shapes (Insert, Shapes, Flowchart)
3. Add hyperlinks.
4. Turn off annoying warnings (Google : Enable Or Disable Hyperlink Warnings For Office)



Raw

Master.do

NEW7080.dta

TableIV_data.do

TableIV_data.do

TableIV.log

TableV_data.do

TableV_data.do

TableV.log

TableVI_data.do

TableVI_data.do

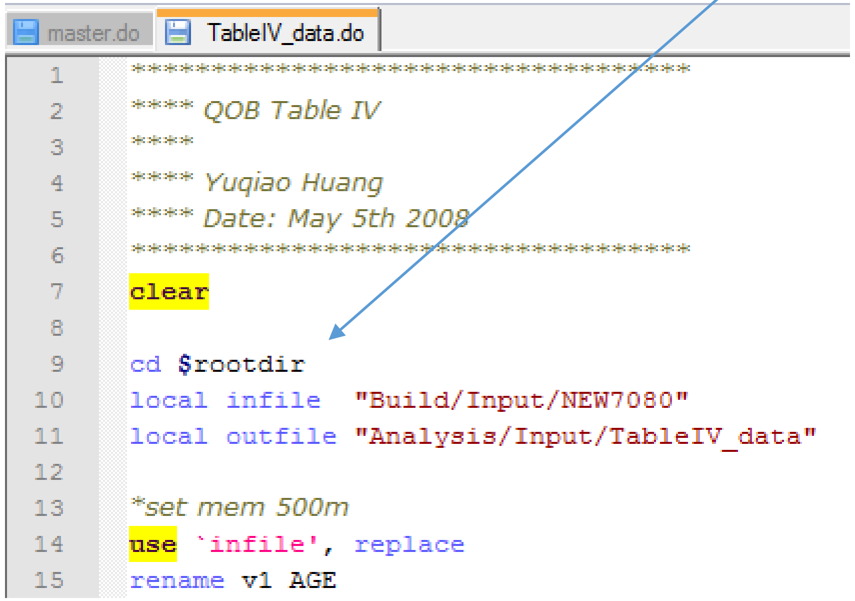
TableVI.log

Master file: automation

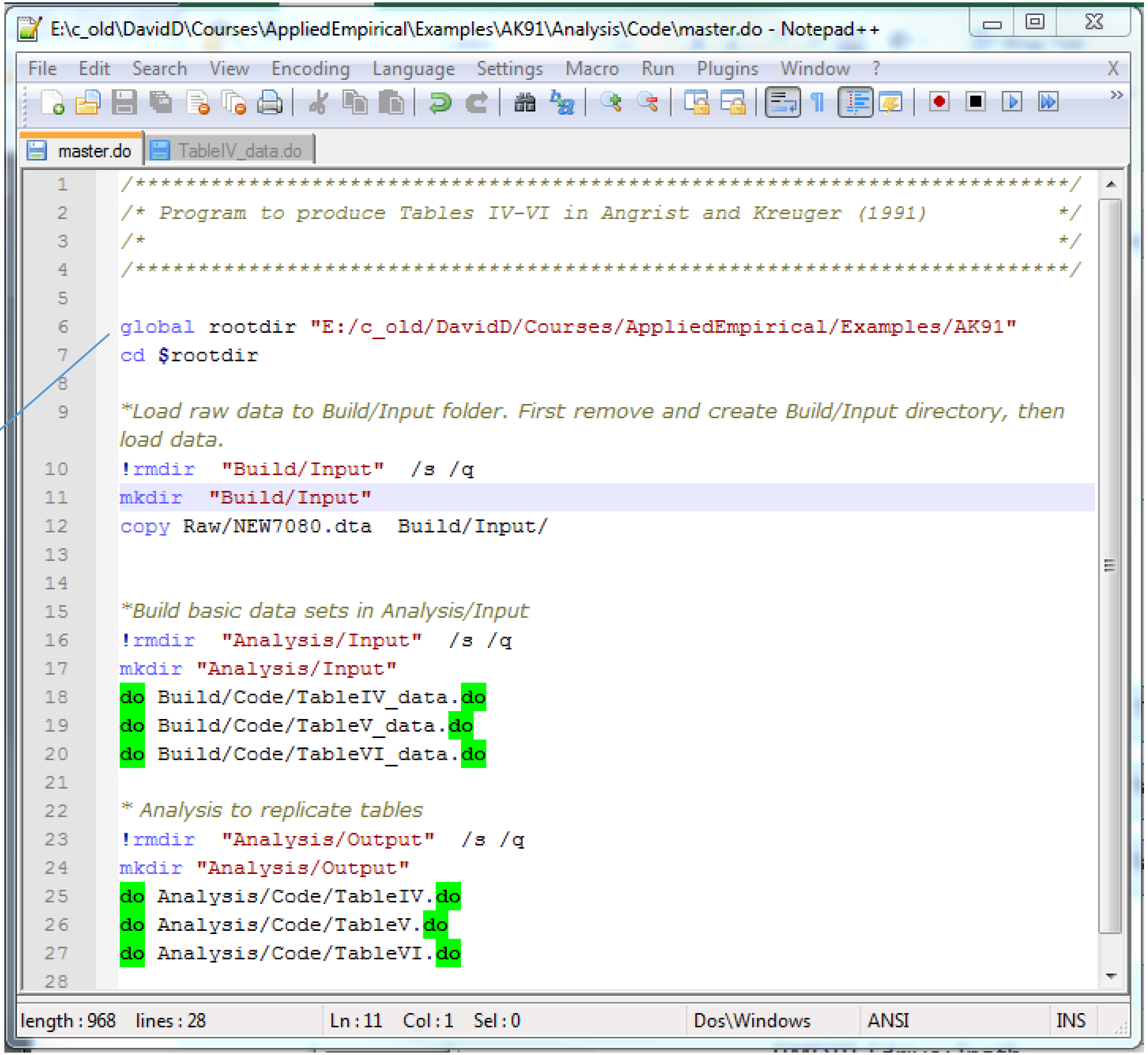
- Automate every step of the analysis from start to end.
- Write a singe script that executes all code
  from raw data to final output (tables, figures, numbers, etc.).
- Stata commands to do this.

Master file

- root dir in global macro
- ! is to execute dos commands in stata
- rmdir removes directory
- mkdir makes directory

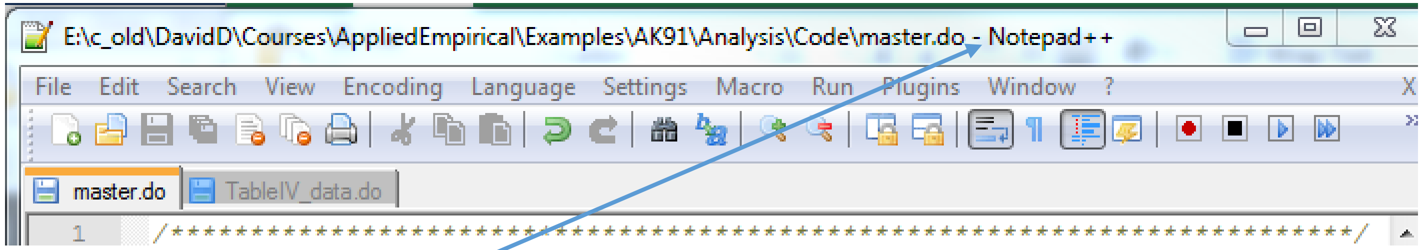


```
************************************
**** QOB Table IV
****
**** Yuqiao Huang
**** Date: May 5th 2008
************************************
clear

cd $rootdir
local infile  "Build/Input/NEW7080"
local outfile "Analysis/Input/TableIV_data"

*set mem 500m
use `infile', replace
rename v1 AGE
```

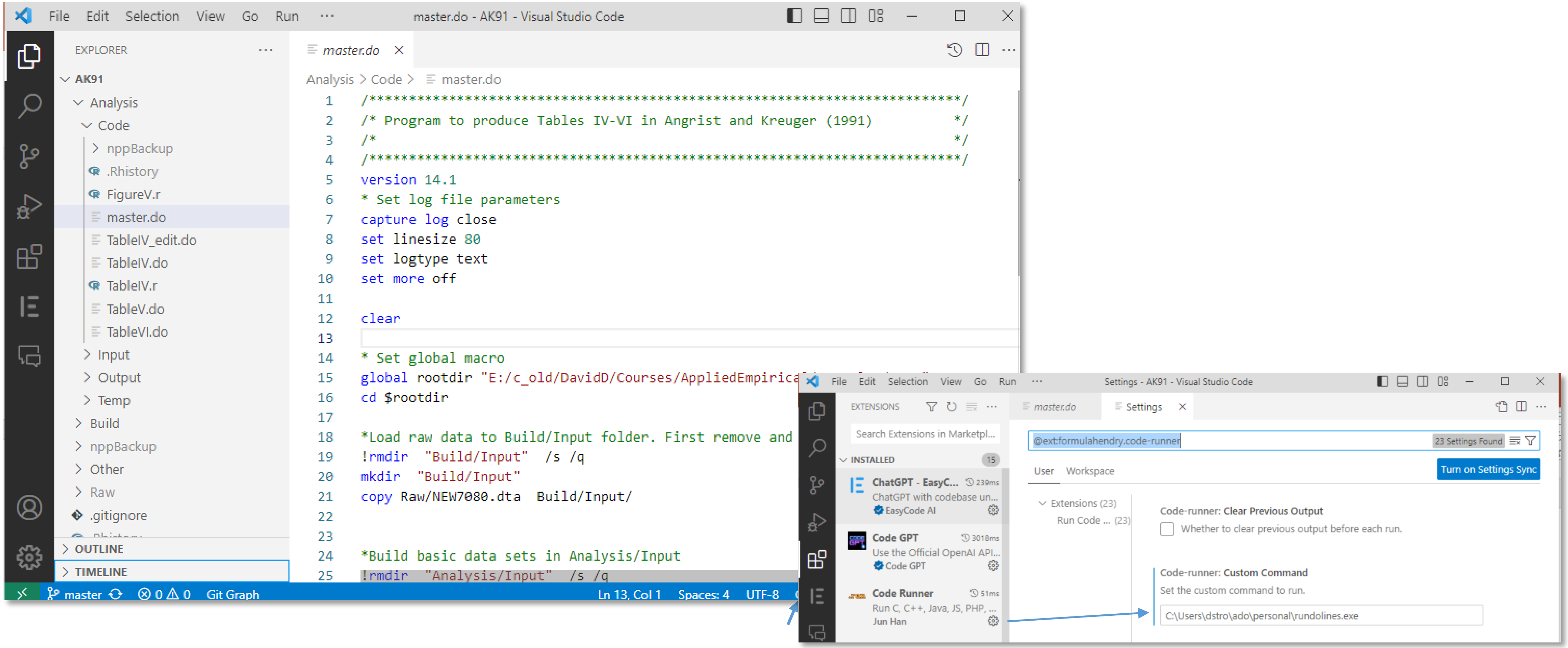


```
/*************************************************************************/
/* Program to produce Tables IV-VI in Angrist and Kreuger (1991)        */
/*                                                                       */
/*************************************************************************/

global rootdir "E:/c_old/DavidD/Courses/AppliedEmpirical/Examples/AK91"
cd $rootdir

*Load raw data to Build/Input folder. First remove and create Build/Input directory, then load data.
!rmdir  "Build/Input"  /s /q
mkdir  "Build/Input"
copy Raw/NEW7080.dta  Build/Input/


*Build basic data sets in Analysis/Input
!rmdir  "Analysis/Input"  /s /q
mkdir "Analysis/Input"
do Build/Code/TableIV_data.do
do Build/Code/TableV_data.do
do Build/Code/TableVI_data.do

* Analysis to replicate tables
!rmdir  "Analysis/Output"  /s /q
mkdir "Analysis/Output"
do Analysis/Code/TableIV.do
do Analysis/Code/TableV.do
do Analysis/Code/TableVI.do

```

Text editor



- An external editor does not have wait for Stata to finish execution, and does not crash when Stata does.
- I use Notepad++
- see ” Using external text editors to write do files” by Friedrich Huebler http://huebler.blogspot.com/2005/03/integrating-stata-and-external-text.html



```
/*************************************************************************/
/* Program to produce Tables IV-VI in Angrist and Kreuger (1991)         */
/*                                                                       */
/*************************************************************************/
version 14.1
* Set log file parameters
capture log close
set linesize 80
set logtype text
set more off

clear

* Set global macro
global rootdir "E:/c_old/DavidD/Courses/AppliedEmpirica
cd $rootdir

*Load raw data to Build/Input folder. First remove and
!rmdir  "Build/Input"  /s /q
mkdir  "Build/Input"
copy Raw/NEW7080.dta  Build/Input/


*Build basic data sets in Analysis/Input
!rmdir  "Analysis/Input"  /s /q
```

@ext:formulahendry.code-runner

Code-runner: **Clear Previous Output**

☐ Whether to clear previous output before each run.

Code-runner: **Custom Command**

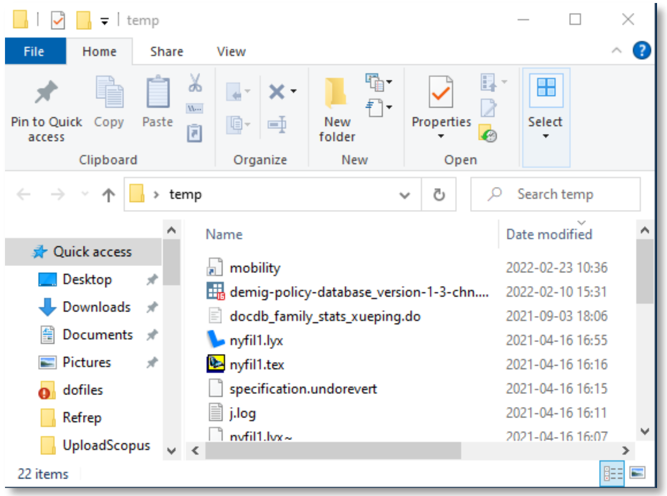
Set the custom command to run.

C:\Users\dstro\ado\personal\rundolines.exe

- You can also use, for example, Visual Studio Code
- To integrate VSC with Stata, I use the extension Code Runner, combined with rundolines/rundo by Friedrich Huebler.

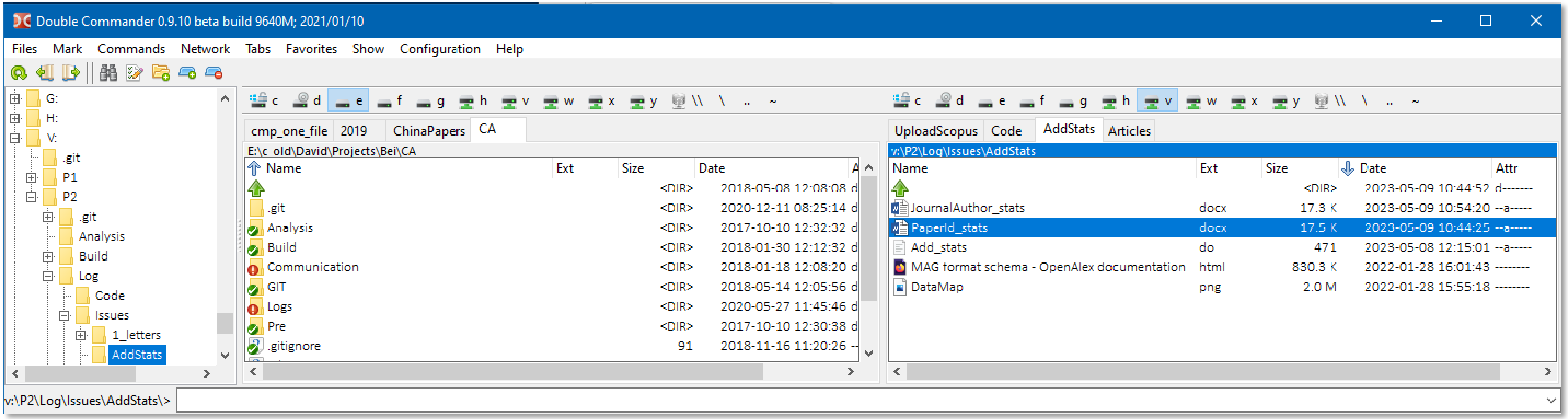
File manager

- *Windows Explorer* not well suited for data work.



Name	Date modified
mobility	2022-02-23 10:36
demig-policy-database_version-1-3-chn....	2022-02-10 15:31
docdb_family_stats_xueping.do	2021-09-03 18:06
nyfil1.lyx	2021-04-16 16:55
nyfil1.tex	2021-04-16 16:16
specification.undorevert	2021-04-16 16:15
j.log	2021-04-16 16:11
nyfil1.lyx~	2021-04-16 16:07

- I use *Double Commander,* a free cross platform open source file manager.



E:\c_old\David\Projects\Bei\CA

Name	Ext	Size	Date	A
..		<DIR>	2018-05-08 12:08:08	d
.git		<DIR>	2020-12-11 08:25:14	d
Analysis		<DIR>	2017-10-10 12:32:32	d
Build		<DIR>	2018-01-30 12:12:32	d
Communication		<DIR>	2018-01-18 12:08:20	d
GIT		<DIR>	2018-05-14 12:05:56	d
Logs		<DIR>	2020-05-27 11:45:46	d
Pre		<DIR>	2017-10-10 12:30:38	d
.gitignore		91	2018-11-16 11:20:26	-

v:\P2\Log\Issues\AddStats

Name	Ext	Size	Date	Attr
..		<DIR>	2023-05-09 10:44:52	d-------
JournalAuthor_stats	docx	17.3 K	2023-05-09 10:54:20	--a-----
PaperId_stats	docx	17.5 K	2023-05-09 10:44:25	--a-----
Add_stats	do	471	2023-05-08 12:15:01	--a-----
MAG format schema - OpenAlex documentation	html	830.3 K	2022-01-28 16:01:43	--------
DataMap	png	2.0 M	2022-01-28 15:55:18	--------

v:\P2\Log\Issues\AddStats\>

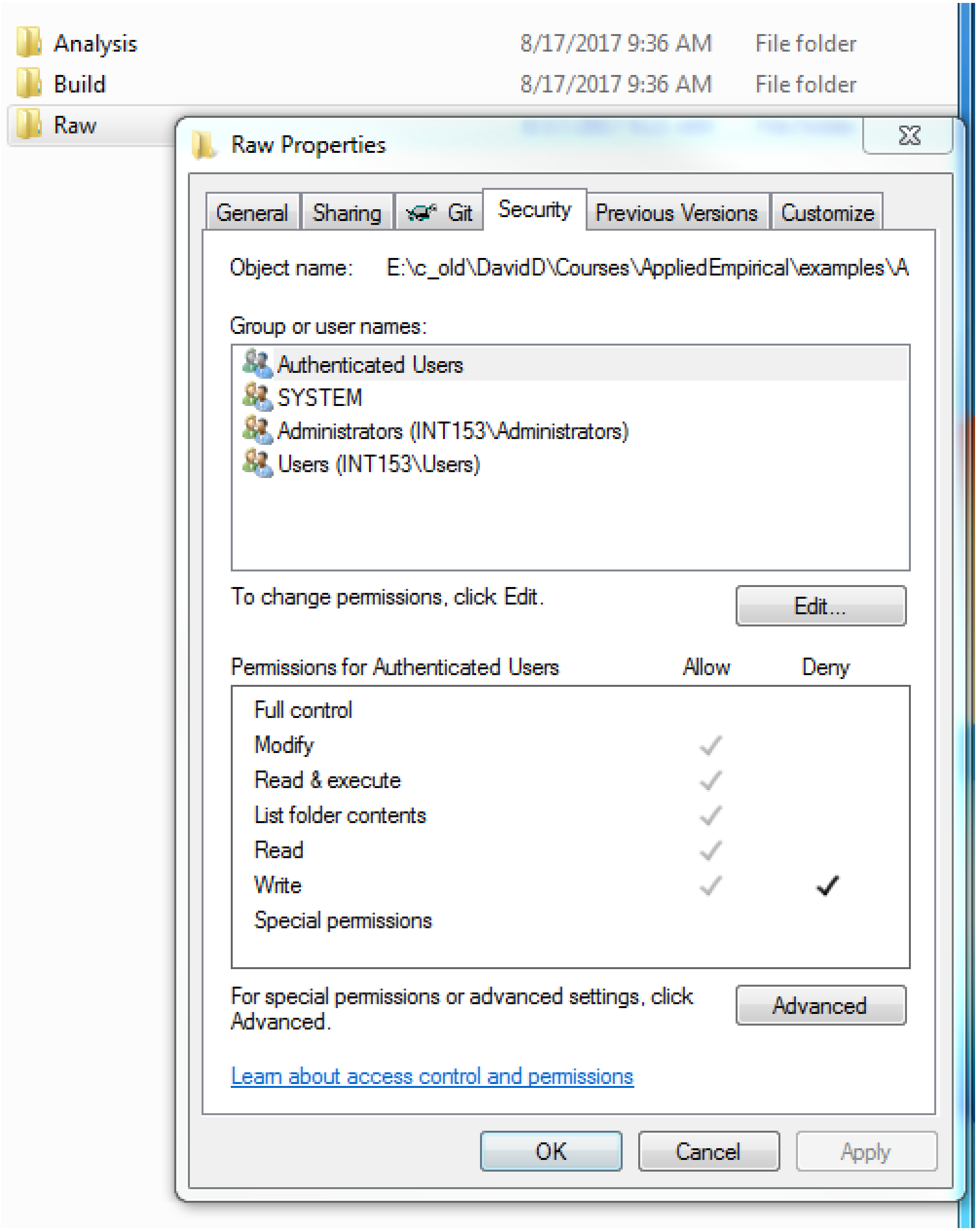
Data management

- Save raw data. Never change (deny write in raw data directory).
- Structuring data sets
  - Keys and tidy data.
  - Keep data in normal form.
  - No duplication.
    - Never same variable in two places.
- Data cleaning.
  - Verify data.
  - Generate variables
- (Backup and documentation discussed separately).

Never change raw data

- Deny writing to raw data directory

- In Windows Explorer, right-click the file or folder you want to work with.
- From the pop-up menu, select Properties, and then in the Properties dialog box click the Security tab.
- In the Name list box, select the user, contact, computer, or group whose permissions you want to view. If the permissions are dimmed, it means the permissions are inherited from a parent object.



Analysis	8/17/2017 9:36 AM	File folder
Build	8/17/2017 9:36 AM	File folder
Raw		

Object name: E:\c_old\DavidD\Courses\AppliedEmpirical\examples\A

Group or user names:

- Authenticated Users
- SYSTEM
- Administrators (INT153\Administrators)
- Users (INT153\Users)

To change permissions, click Edit.

Permissions for Authenticated Users	Allow	Deny
Full control		
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	✓
Special permissions		

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

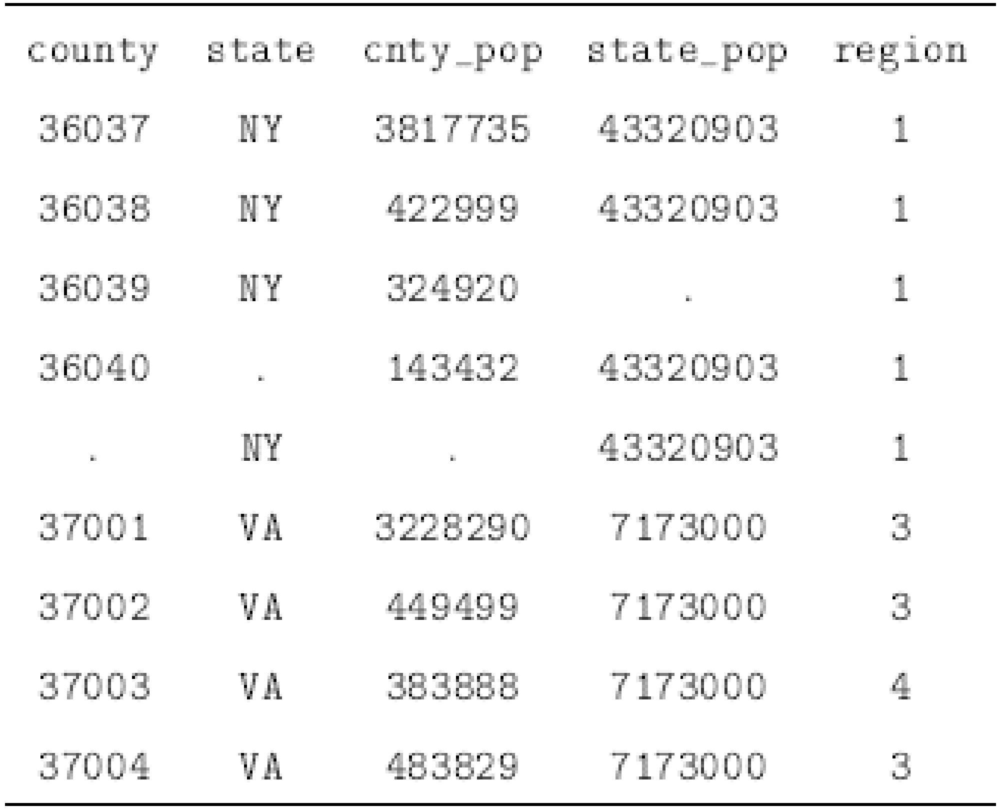
Keys and tidy data.

Rules

1. Store data in tables with unique, non-missing keys.
2. Keep data normalized as far into your code pipepline as you can.

Messy table example

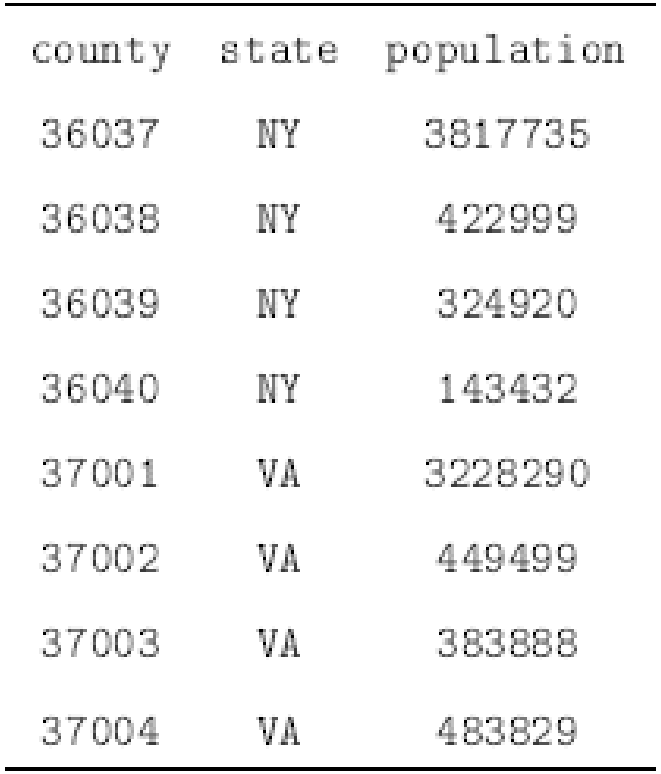
- Population for NY is 43 million for one observation and missing for another.
- County is missing – what does the observation mean?
- Why does region take different values for VA?
- Why is state missing for one county?



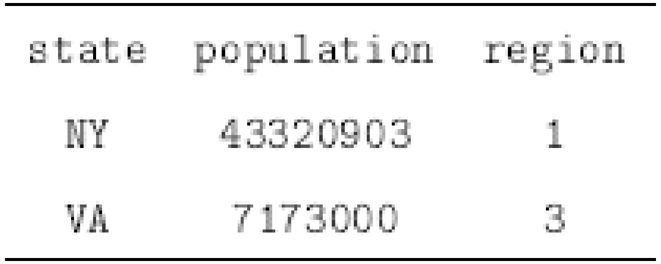
county	state	cnty_pop	state_pop	region
36037	NY	3817735	43320903	1
36038	NY	422999	43320903	1
36039	NY	324920	.	1
36040	.	143432	43320903	1
.	NY	.	43320903	1
37001	VA	3228290	7173000	3
37002	VA	449499	7173000	3
37003	VA	383888	7173000	4
37004	VA	483829	7173000	3

Relational database

People who work professionally with databases would present the data in this form.



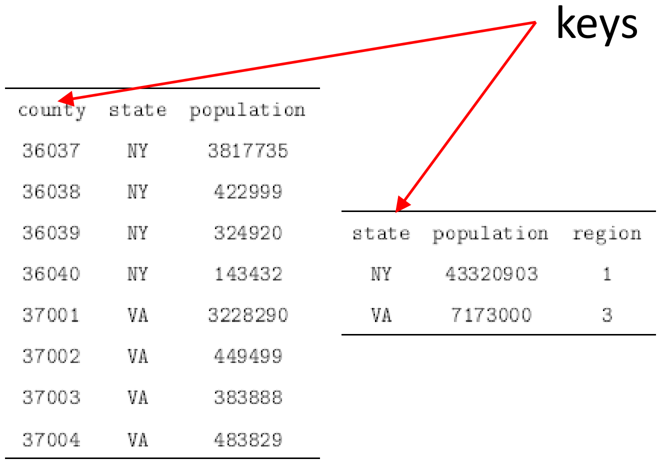
county	state	population
36037	NY	3817735
36038	NY	422999
36039	NY	324920
36040	NY	143432
37001	VA	3228290
37002	VA	449499
37003	VA	383888
37004	VA	483829



state	population	region
NY	43320903	1
VA	7173000	3

Relational database

- Each table has a key:
  a variable (or a set of variables) that
  uniquely identifies the elements of a table.
  - Keys never take on missing value.
  - A key's value is never duplicated across rows of a table.



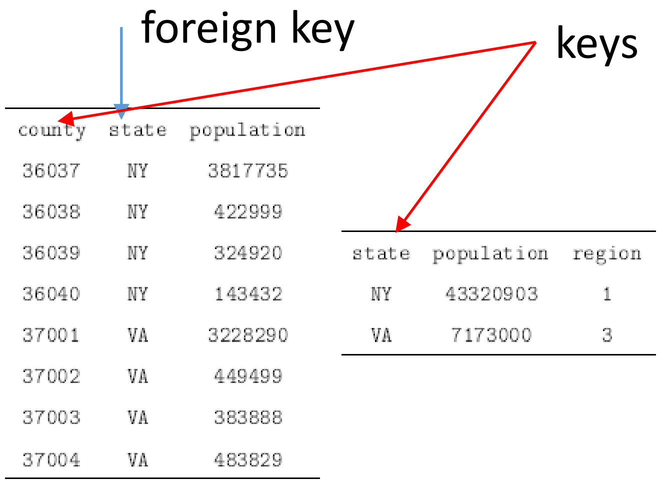
keys

county	state	population
36037	NY	3817735
36038	NY	422999
36039	NY	324920
36040	NY	143432
37001	VA	3228290
37002	VA	449499
37003	VA	383888
37004	VA	483829

state	population	region
NY	43320903	1
VA	7173000	3

Relational database

- Each table has a key=
  a variable (or a set of variables) that
  uniquely identifies the elements of a table.
  - Keys never take on missing value.
  - A key's value is never duplicated across rows of a table.
- The other variables are attributes of the key.
- Tables are connected by foreign keys.

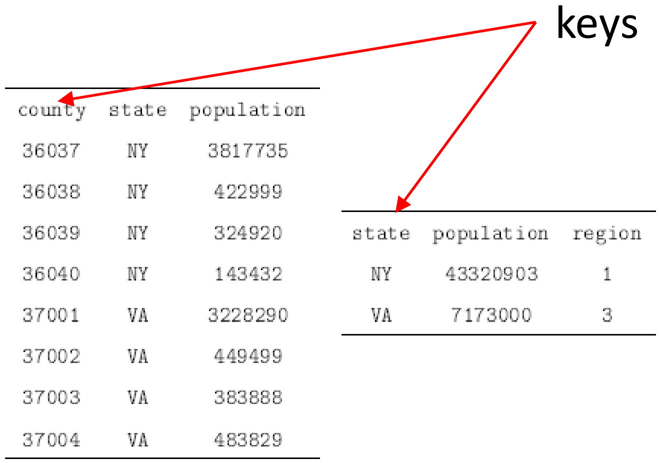


county	state	population
36037	NY	3817735
36038	NY	422999
36039	NY	324920
36040	NY	143432
37001	VA	3228290
37002	VA	449499
37003	VA	383888
37004	VA	483829

state	population	region
NY	43320903	1
VA	7173000	3

Relational database

- Data stored in this form are normalized.
  - Each fact is expressed only once (no duplication).
  - Avoids inconsistencies.
  - Facilitates data merging.
  - Efficient when working with large data sets.



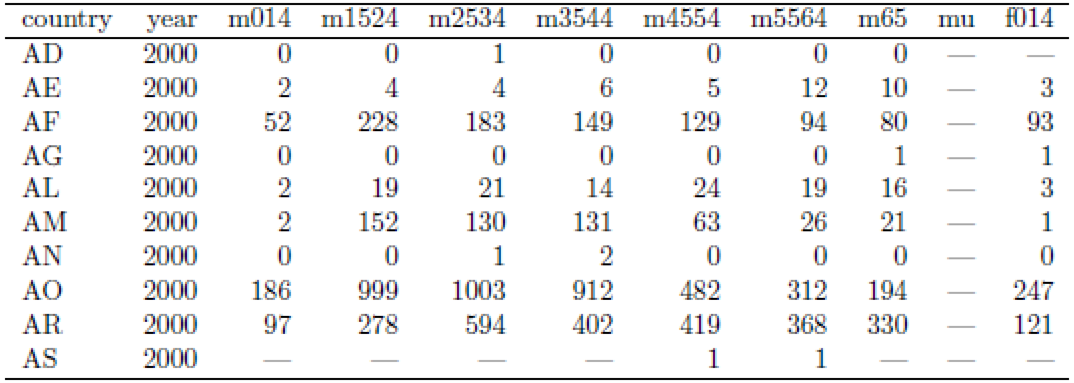
county	state	population
36037	NY	3817735
36038	NY	422999
36039	NY	324920
36040	NY	143432
37001	VA	3228290
37002	VA	449499
37003	VA	383888
37004	VA	483829

state	population	region
NY	43320903	1
VA	7173000	3

- Common forms of messy data
  - Key is missing or not unique.
    - Fatal. Always check!
    - Stata commands:
      - duplicates report
      - duplicates tag, gen() to fix unique id issues
      - isid or isid, missok
      - bysort key: gen N=_N
        tab N
  - Multiple types stored in one table (previous example).

Common forms of messy data

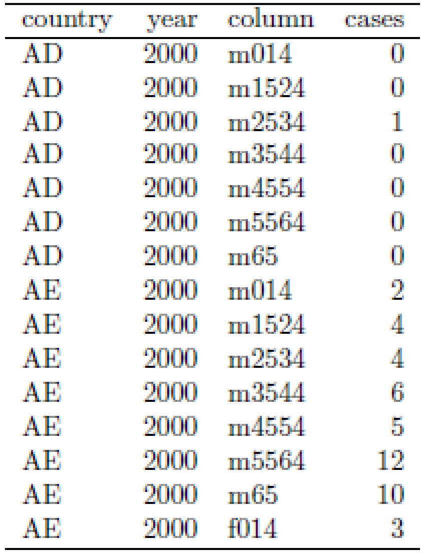
- Variables stored in column names.



country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Table 9: Original TB dataset. Corresponding to each 'm' column for males, there is also an 'f' column for females, `f1524`, `f2534` and so on. These are not shown to conserve space. Note the mixture of 0s and missing values (—). This is due to the data collection process and the distinction is important for this dataset.

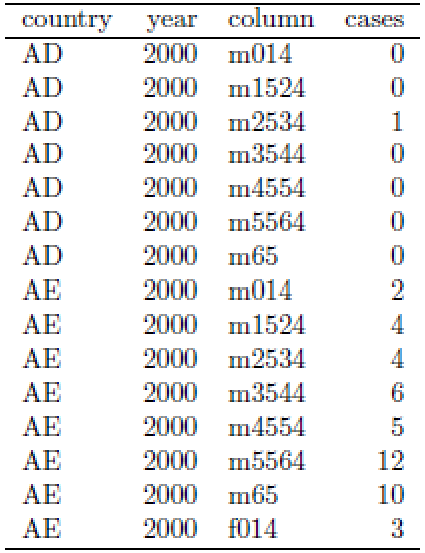
Stata: reshape long



country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

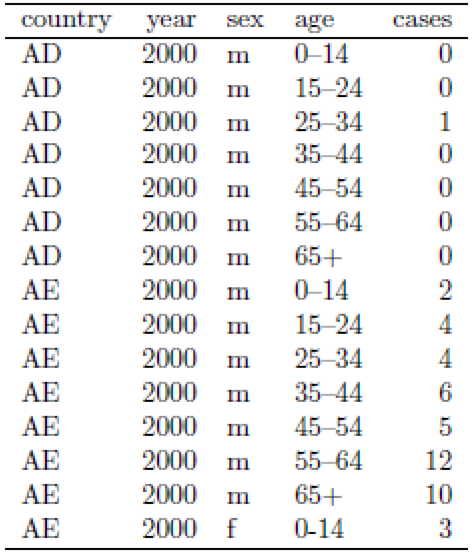
(a) Molten data

- Common forms of messy data
  - Multiple variables stored in one variable



country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

(a) Molten data



country	year	sex	age	cases
AD	2000	m	0–14	0
AD	2000	m	15–24	0
AD	2000	m	25–34	1
AD	2000	m	35–44	0
AD	2000	m	45–54	0
AD	2000	m	55–64	0
AD	2000	m	65+	0
AE	2000	m	0–14	2
AE	2000	m	15–24	4
AE	2000	m	25–34	4
AE	2000	m	35–44	6
AE	2000	m	45–54	5
AE	2000	m	55–64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

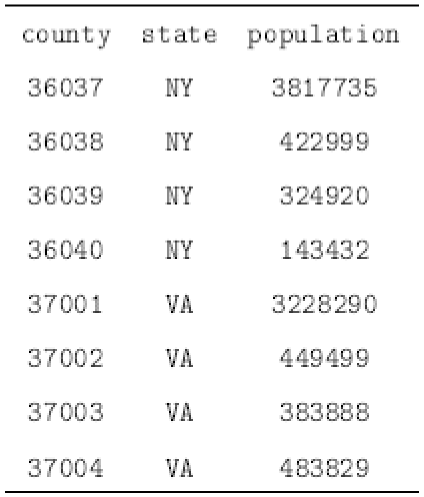
(b) Tidy data

Stata: substr or regex

Merging normalized data sets

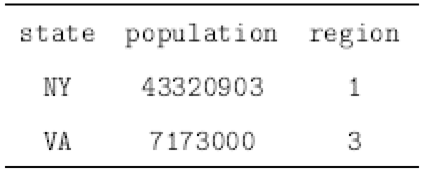
- Combining data sets with the same key will always be 1:1 merge.
- Merging with a foreign key implies merging m:1 or 1:m.
- Never merge m:m!

county_pop.dta



county	state	population
36037	NY	3817735
36038	NY	422999
36039	NY	324920
36040	NY	143432
37001	VA	3228290
37002	VA	449499
37003	VA	383888
37004	VA	483829

state_pop_.dta



state	population	region
NY	43320903	1
VA	7173000	3

- use county_pop, replace
- merge m:1 state using state_pop

Cleaning data

- When my kids ask me what I do, I say

  ”I clean data”.

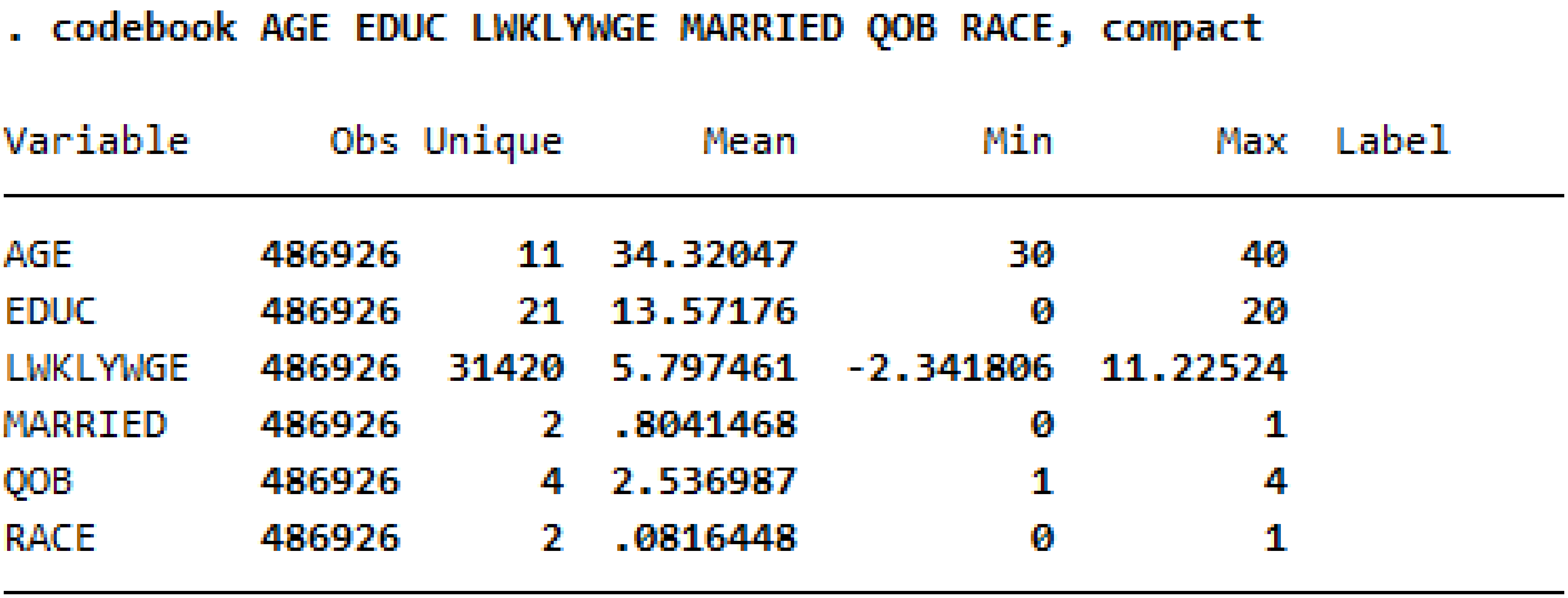
  - IIES Assistant Professor.

Cleaning data: Verifying variables

- Some of your data will be wrong.
  Find out which before you invest time and energy in incorrect results.

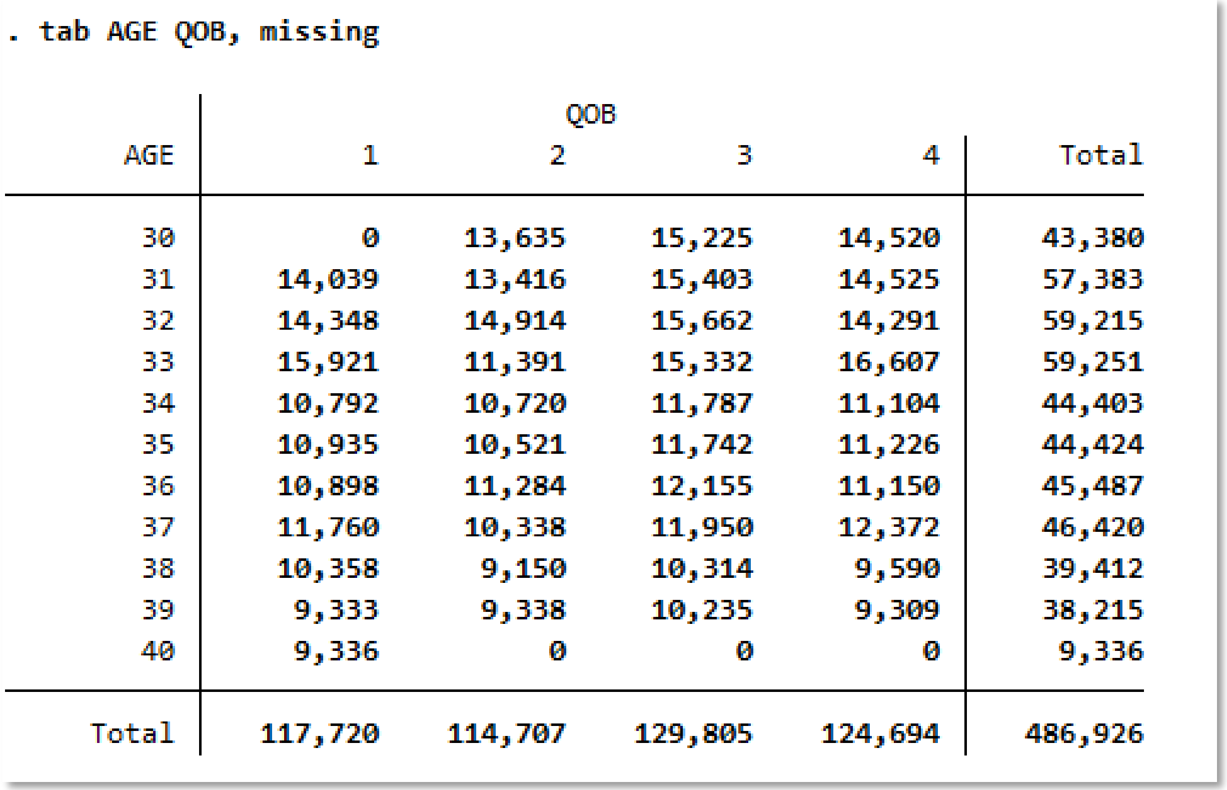
- Values review
  - Check descriptive stats:
    # missing observations, # of unique values, mean, min, max.
  - Check values with tab1 or dotplot.
  - Use tabulate or scatter to examing pairs of variables.

Values review: summary statistics, missing values, unique values



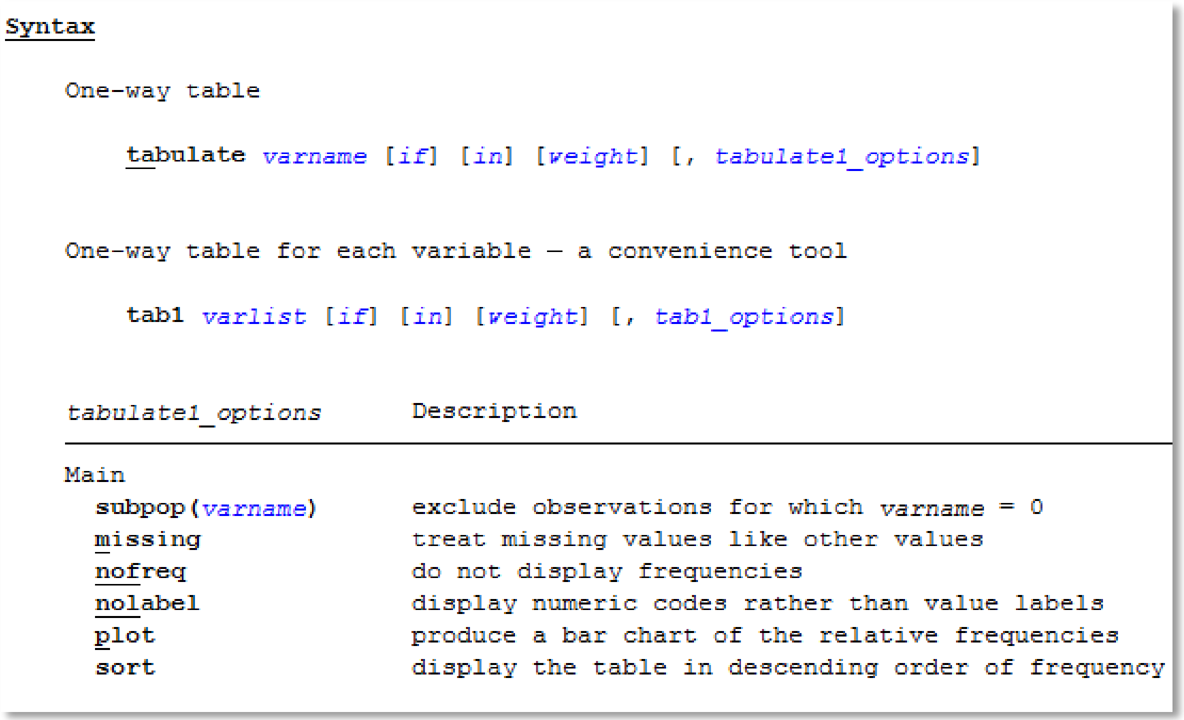
```
. codebook AGE EDUC LWKLYWGE MARRIED QOB RACE, compact

Variable       Obs Unique      Mean        Min       Max  Label
------------------------------------------------------------------
AGE         486926     11  34.32047         30        40
EDUC        486926     21  13.57176          0        20
LWKLYWGE    486926  31420  5.797461  -2.341806  11.22524
MARRIED     486926      2  .8041468          0         1
QOB         486926      4  2.536987          1         4
RACE        486926      2  .0816448          0         1
------------------------------------------------------------------
```

Values review: discrete variables



```
. tab AGE QOB, missing

           |                    QOB
       AGE |         1          2          3          4 |     Total
-----------+--------------------------------------------+----------
        30 |         0     13,635     15,225     14,520 |    43,380
        31 |    14,039     13,416     15,403     14,525 |    57,383
        32 |    14,348     14,914     15,662     14,291 |    59,215
        33 |    15,921     11,391     15,332     16,607 |    59,251
        34 |    10,792     10,720     11,787     11,104 |    44,403
        35 |    10,935     10,521     11,742     11,226 |    44,424
        36 |    10,898     11,284     12,155     11,150 |    45,487
        37 |    11,760     10,338     11,950     12,372 |    46,420
        38 |    10,358      9,150     10,314      9,590 |    39,412
        39 |     9,333      9,338     10,235      9,309 |    38,215
        40 |     9,336          0          0          0 |     9,336
-----------+--------------------------------------------+----------
     Total |   117,720    114,707    129,805    124,694 |   486,926
```

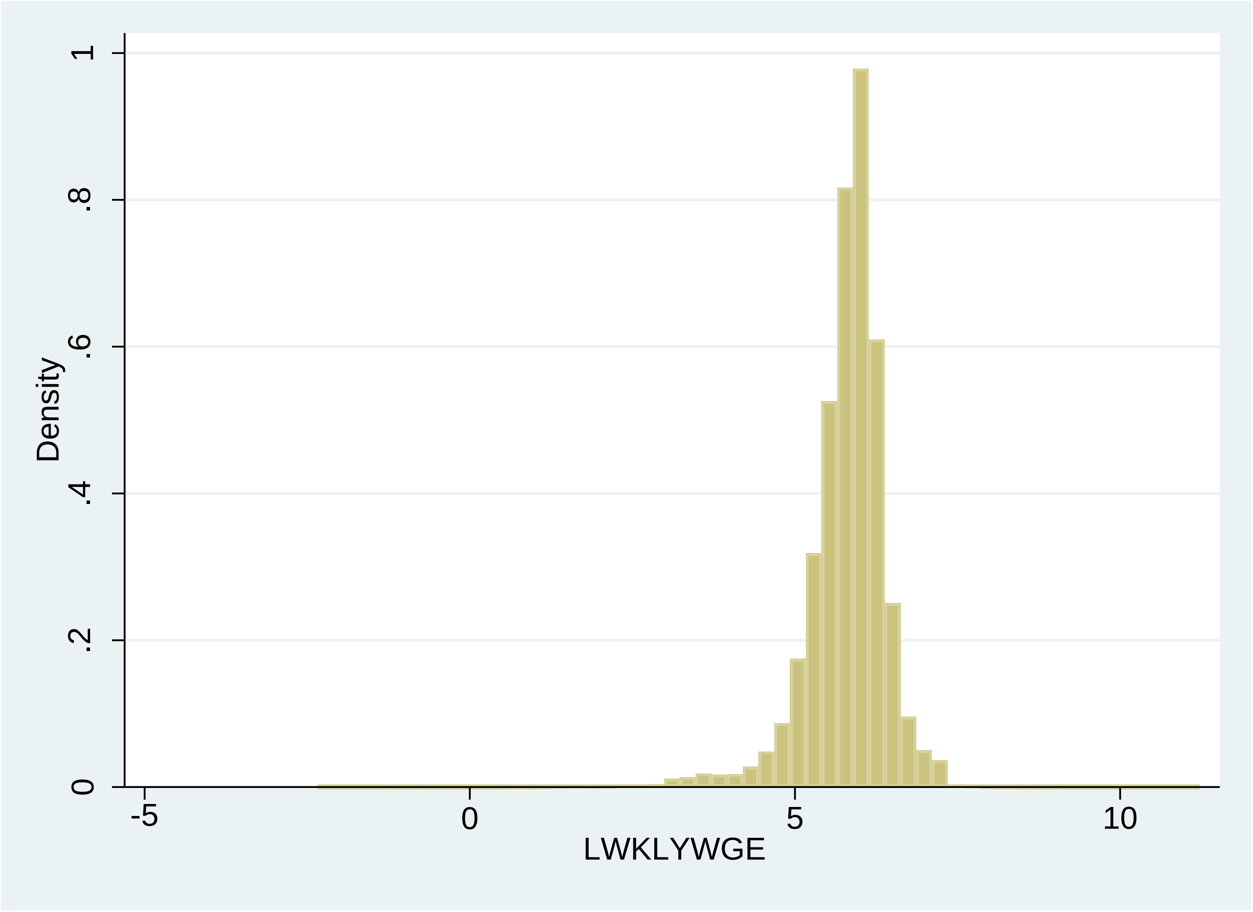
- help tabulate



```
Syntax

    One-way table

        tabulate varname [if] [in] [weight] [, tabulate1_options]


    One-way table for each variable — a convenience tool

        tab1 varlist [if] [in] [weight] [, tab1_options]


    tabulate1_options         Description
    --------------------------------------------------------------------
    Main
      subpop(varname)         exclude observations for which varname = 0
      missing                 treat missing values like other values
      nofreq                  do not display frequencies
      nolabel                 display numeric codes rather than value labels
      plot                    produce a bar chart of the relative frequencies
      sort                    display the table in descending order of frequency
```

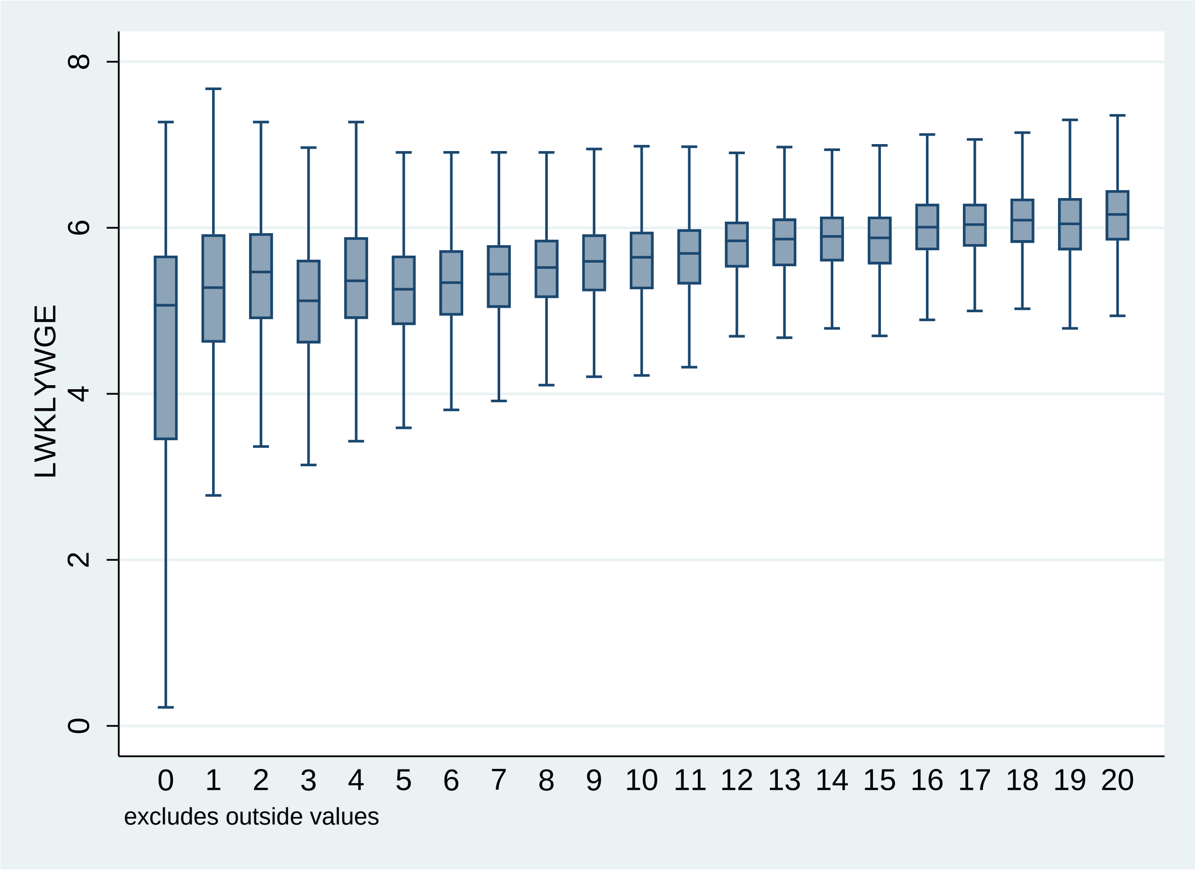
Values review: continuous variables

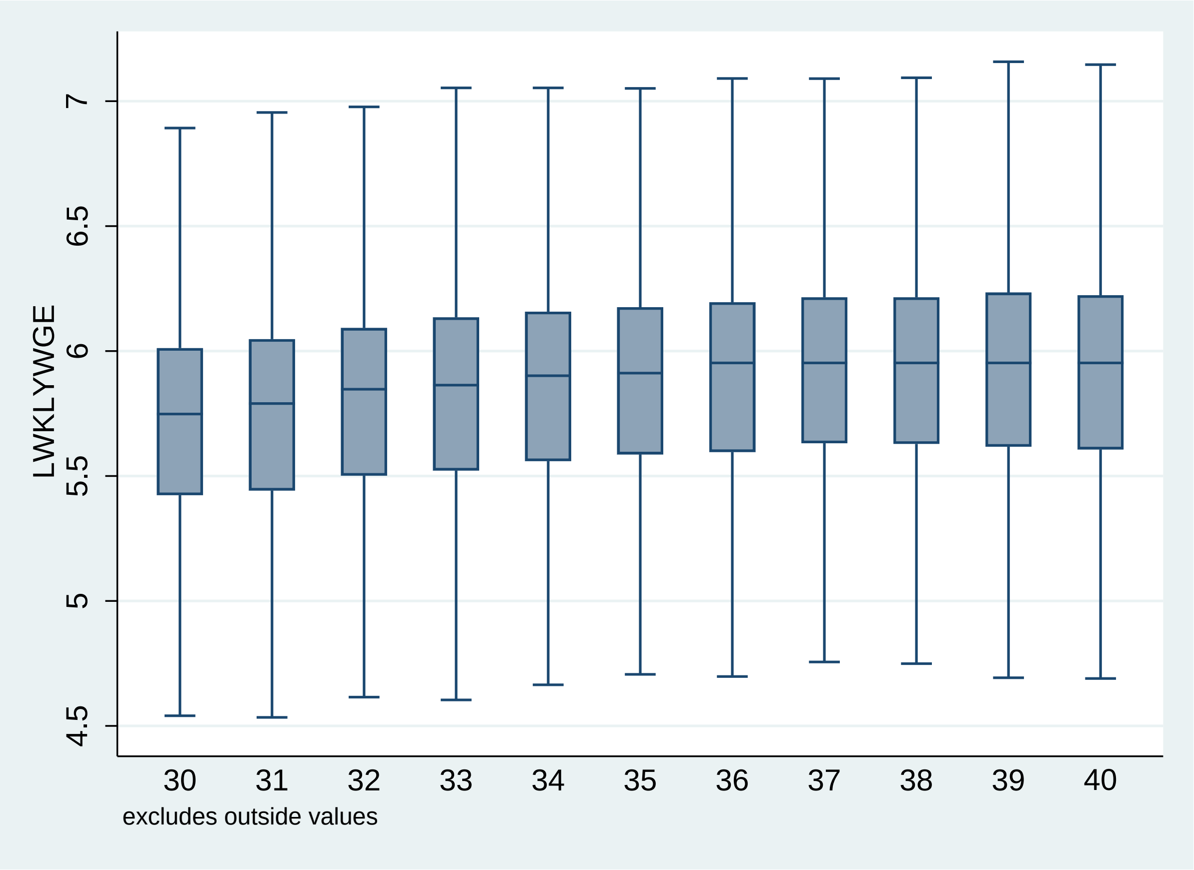
- Wage distribution approximately log normal



```
. hist LWKLYWGE
```

Values review: continuous variables





```
graph box LWKLYWGE, over(EDUC) nooutsides
```

```
graph box LWKLYWGE, over(AGE) nooutsides
```

Clean data: Substantive review

- Internal consistency
  - Do the values make sense?
    - When there are logical links, verify that data is consistent.
    - If someone is not in the labor force, they should not report a wage.
    - If you do not publish, you should not be cited.
    - If education begins at age 5, years of education must be 5 less than age.
  - Examine high frequency values ?
  - Do correlations between variables make sense ?
- External consistency
  - Do means correspond to external data, e.g. national statistics?
  - Can you replicate simple correlations from existing studies?

Fixing inconsistencies

- Coding inconsistent cases as missing or imputing values can introduce selection bias and other distortions.
    - Are some missing values zeroes? Check other sources.
- Always document what you did and why!

Number of observations

- Always keep track of the number of observations.
- Stata treats missing values “." as ∞
    - gen post2000 = year> 2000
        - will code post2000=1 when year==.
    - gen post2000 = year> 2000 if year!=.
        - will work

Task 1: Replicate Angrist & Krueger (1991)

- Download dofiles and data to replicate Angrist & Krueger (1991) from https://economics.mit.edu/people/faculty/josh-angrist/angrist-data-archive
- Run the program and compare the output with the published tables.
- Now set up a folder structure following Rule 1 and Rule 2 above.
- Split the dofiles to separate the data management and analysis stage.
- Create a master file for your replication project.
- Create a project map using Excel (or other similar software).

Task 1

Please submit your assignment to Jinci Liu
by sharing a folder/path on Google Drive with anton.hansing@iies.su.se.

Use the following structure:

- A parental folder named as "your_name“, for example, “John_Smith", for all assignments.
- A subfolder for each task named"task#“ (e.g “task1”), containing all task material, including
- a ReadMeFirst.txt in every subfolder
  - -- for example, the path to key files and the project map, etc.

Submit by giving access to the parental folder, by sending invitation to anton.hansing@iies.su.se.

- I will check/run the files after every deadline.

A one-time set up will be enough for all future submissions.

If you have any questions about this setup, please email anton.hansing@iies.su.se.