

Nuts and Bolts

Matthew Gentzkow
Jesse M. Shapiro

Chicago Booth and NBER

Introduction

- We have focused on the statistical / econometric issues that arise with big data
- In the time that remains, we want to spend a little time on the *practical* issues...

Introduction

- We have focused on the statistical / econometric issues that arise with big data
- In the time that remains, we want to spend a little time on the *practical* issues...
 - E.g., where do you actually put a 2 TB dataset?

Introduction

- We have focused on the statistical / econometric issues that arise with big data
- In the time that remains, we want to spend a little time on the *practical* issues...
 - E.g., where do you actually put a 2 TB dataset?
- Goal: Sketch some basic computing ideas relevant to working with large datasets.

Introduction

- We have focused on the statistical / econometric issues that arise with big data
- In the time that remains, we want to spend a little time on the *practical* issues...
 - E.g., where do you actually put a 2 TB dataset?
- Goal: Sketch some basic computing ideas relevant to working with large datasets.
- Caveat: We are all amateurs

The Good News

- Much of what we've talked about here you can do on your laptop
 - Your OS knows how to do parallel computing (multiple processors, multiple cores)
 - Many “big” datasets are < 5 GB
 - Save the data to local disk, fire up Stata or R, and off you go...

How Big is Big?

Congressional record text (1870-2010)	≈50 GB
Congressional record pdfs (1870-2010)	≈500 GB
Nielsen scanner data (34k stores, 2004-2010)	≈5 TB
Wikipedia (2013)	≈6 TB
20% Medicare claims data (1997-2009)	≈10 TB
Facebook (2013)	≈100,000 TB
All data in the world	≈2.7 billion TB

Outline

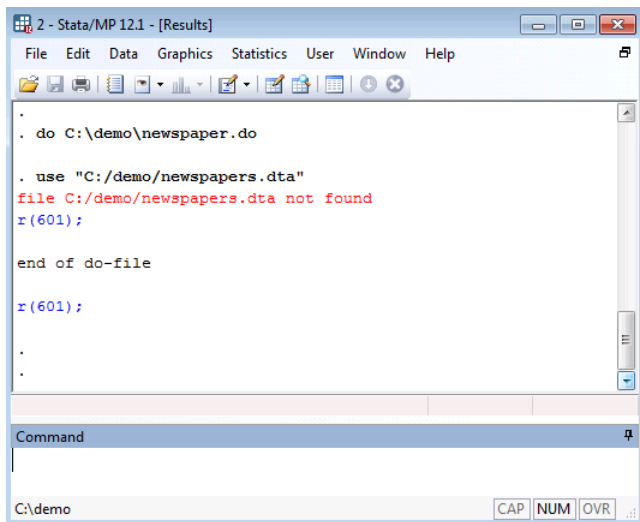
- Software engineering for economists
- Databases
- Cluster computing
- Scenarios

Software Engineering for Economists

Motivation

- A lot of the time spent in empirical research is writing, reading, and debugging code.
- Common situations...

Broken Code

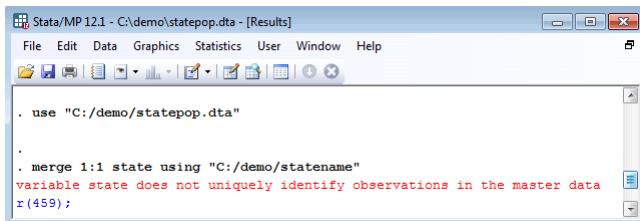


The screenshot shows a Stata/MP 12.1 Results window. The window title is "2 - Stata/MP 12.1 - [Results]". The menu bar includes File, Edit, Data, Graphics, Statistics, User, Window, and Help. The toolbar contains icons for file operations and analysis. The main text area displays the following code and output:

```
.  
. do C:\demo\newspaper.do  
  
. use "C:/demo/newspapers.dta"  
file C:/demo/newspapers.dta not found  
r(601);  
  
end of do-file  
  
r(601);  
  
.  
.
```

Below the main text area is a "Command" input field, which is currently empty. At the bottom of the window, the current directory is shown as "C:\demo", and there are buttons for "CAP", "NUM", "OVR", and a menu icon.

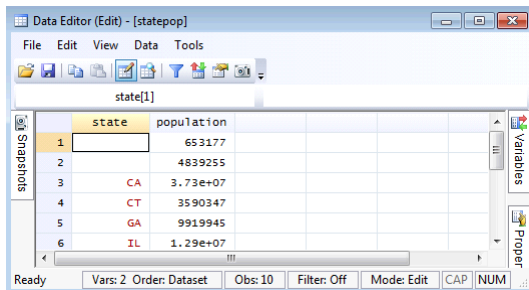
Incoherent Data



Stata/MP 12.1 - C:\demo\statepop.dta - [Results]

```
File Edit Data Graphics Statistics User Window Help
```

```
. use "C:/demo/statepop.dta"  
  
. merge 1:1 state using "C:/demo/statename"  
variable state does not uniquely identify observations in the master data  
r(459);
```



Data Editor (Edit) - [statepop]

File Edit View Data Tools

state[1]

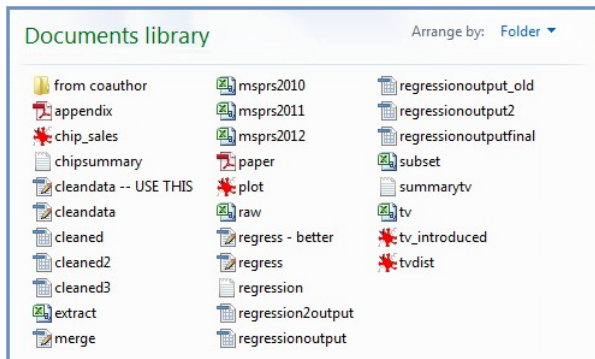
	state	population
1		653177
2		4839255
3	CA	3.73e+07
4	CT	3590347
5	GA	9919945
6	IL	1.29e+07

Ready Vars: 2 Order: Dataset Obs: 10 Filter: Off Mode: Edit CAP NUM

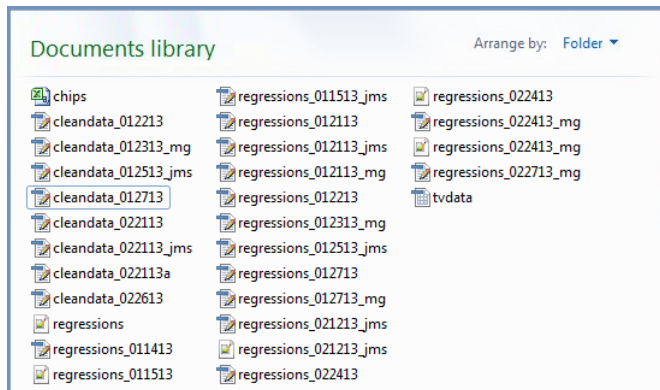
Rampant Duplication

```
Do-file Editor - reg_duplication
File Edit Tools View
reg_duplication reg_locals
1 use "C:/demo/data.dta", clear
2
3 regress probgen coll pharm hlthworker hlthmajor scimajor i.market
4 regress probgen coll pharm hlthworker hlthmajor scimajor i.year
5 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.year
6 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.year age
7 regress probgen coll pharm hlthworker hlthmajor scimajor i.market income
8 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.product
9 regress probgen coll pharm hlthworker hlthmajor scimajor i.year age income
10 regress probgen coll pharm hlthworker hlthmajor scimajor i.market age income
11 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.product i.year age income
12 regress probgen coll pharm hlthworker hlthmajor scimajor i.product i.year age income
13 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.year income
14
15
16
17
Ready Line: 17, Col: 0 CAP NUM OVR
```

Replication Impossible



Tons of Versions



This Talk

- We are not software engineers or computer scientists.
- But we have learned that most common problems in social sciences have analogues in these fields and there are standard solutions.
- Goal is to highlight a few of these that we think are especially valuable to researchers.
- Focus on incremental changes: one step away from common practice.

Automation

Raw Data

Data from original source...

chips_tv - Micr...

File Home Insert Page Layout Formulas Data Review Window Help

Paste Font Alignment Number Styles Cells

Clipboard

D2 1012

	A	B	C	D	E	F
1	county	state	year	chip sales		
2	Autauga	AL	1940	1012		
3	Autauga	AL	1941	1020		
4	Autauga	AL	1942	1034		
5	Autauga	AL	1943	1058		
6	Autauga	AL	1944	1085		
7	Autauga	AL	1945	1148		
8	Autauga	AL	1946	1205		
9	Autauga	AL	1947	1287		
10	Autauga	AL	1948	1299		
11	Autauga	AL	1949	1344		
12	Autauga	AL	1950	1365		
13	Autauga	AL	1951	1397		
14	Autauga	AL	1952	1455		
15	Autauga	AL	1953	1501		
16	Autauga	AL	1954	1582		
17	Autauga	AL	1955	1656		
18	Autauga	AL	1956	1723		
19	Autauga	AL	1957	1795		
20	Autauga	AL	1958	1878		

Ready tv chips Sheet3 100%

chips_tv - Micr...

File Home Insert Page Layout Formulas Data Review Window Help

Paste Font Alignment Number Styles Cells

Clipboard

C32

	A	B	C	D
1	county	state	year_tv_introduced	
2	Autauga	AL		1940
3	Baldwin	AL		1935
4	Barbour	AL		1942
5	Bibb	AL		1942
6	Blount	AL		1939
7	Bullock	AL		1945
8	Butler	AL		1942
9	Calhoun	AL		1936
10	Chambers	AL		1940
11	Cherokee	AL		1939
12	Chilton	AL		1941
13	Choctaw	AL		1942
14	Clarke	AL		1940
15	Clay	AL		1941
16	Cleburne	AL		1943
17	Coffee	AL		1936
18	Colbert	AL		1937
19	Conecuh	AL		1940
20	Coosa	AL		1943

Ready tv chips Sheet3 100%

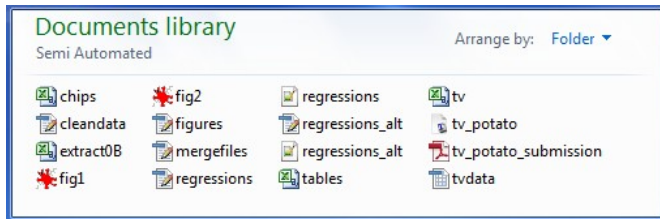
Manual Approach

- Open spreadsheet
- Output to text files
- Open Stata
- Load data, merge files
- Compute $\log(\text{chip sales})$
- Run regression
- Copy results to MS Word and save

Manual Approach

- Two main problems with this approach
 - Replication: how can we be sure we'll find our way back to the exact same numbers?
 - Efficiency: what happens if we change our mind about the right specification?

Semi-automated Approach



- Problems
 - Which file does what?
 - In what order?

Fully Automated Approach

```
File:  rundirectory.bat
stattransfer export_to_csv.stc
statase -b mergefiles.do
statase -b cleandata.do
statase -b regressions.do
statase -b figures.do
pdflatex tv_potato.tex
```

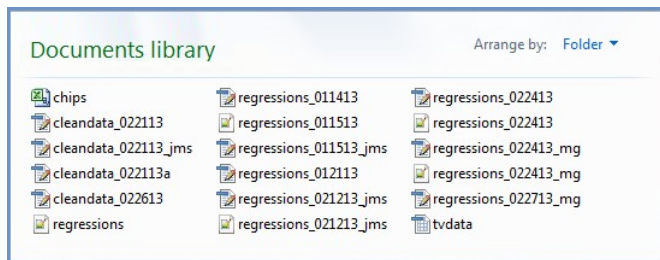
- All steps controlled by a shell script
- Order of steps unambiguous
- Easy to call commands from different packages

Make

- Framework to go from source to target
- Tracks dependencies and revisions
- Avoids rebuilding components that are up to date
- Used to build executable files

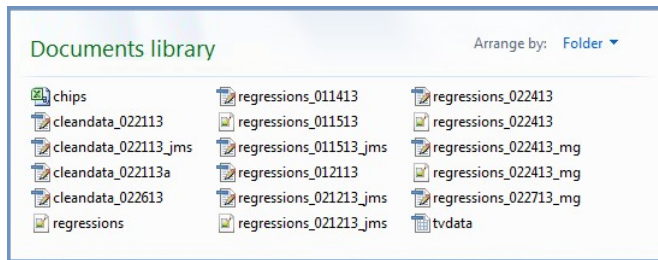
Version Control

After Some Editing



- Dates demarcate versions, initials demarcate authors
- Why do this?
 - Facilitates comparison
 - Facilitates “undo”

What's Wrong with the Approach?









- Why not do this?
 - It's a pain: always have to remember to “tag” every new file
 - It's confusing:
 - Which log file came from `regressions_022713_mg.do`?
 - Which version of `cleandata.do` makes the data used by `regressions_022413.do`?
 - It fails the market test: No software firm does it this way

Version Control

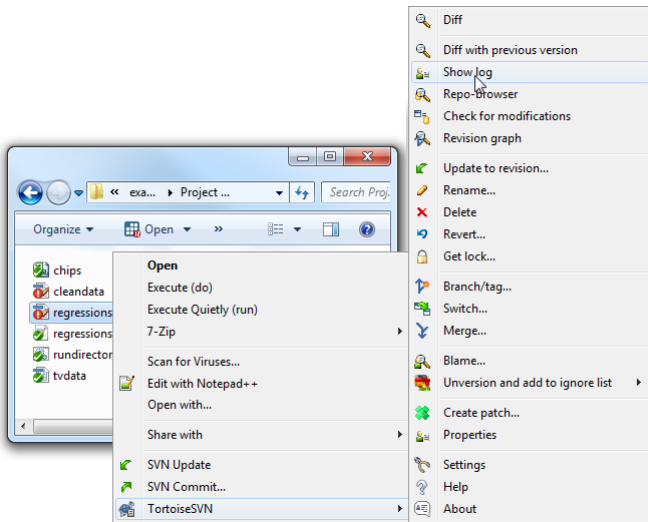
- Software that sits “on top” of your filesystem
 - Keeps track of multiple versions of the same file
 - Records date, authorship
 - Manages conflicts
- Benefits
 - Single authoritative version of the directory
 - Edit without fear: an undo command for everything

Life After Version Control

Documents library

-  chips
-  cleandata
-  regressions
-  regressions
-  rundirectory
-  tvdata

Life After Version Control



Life After Version Control

C:\Users\mwong4\Documents\CODA\slides\examples\Proj...\cleandata.do - Log Messages - TortoiseSVN

Filter by Messages, Paths, Authors, Revisions, Bug-IDs, Date, Date Range From: 7/ 1/2013 To: 7/ 1/2013

Revision	Actions	Author	Date	Message
20657		mwong4	Monday, July 01, 2013 3:28:11 PM	[CODA-15] Added variables for household consumption
20647		mwong4	Monday, July 01, 2013 1:54:40 PM	[CODA-15] Add example directories

[CODA-15] Added variables for household consumption

Path	Action	Copy from path	Revision
/trunk/slides/Code and Data/examples/Project Directory/cleandata.do	Modified		

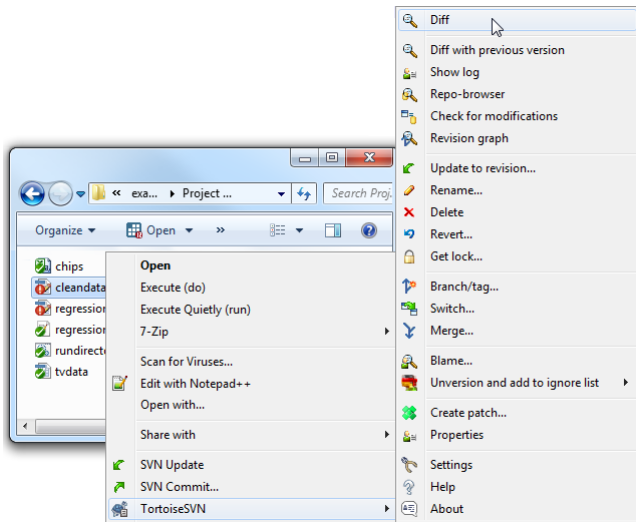
Showing 2 revision(s), from revision 20647 to revision 20657 - 1 revision(s) selected, showing 1 changed paths

Show only affected paths Stop on copy/rename Include merged revisions

Statistics Help OK

Show All Next 100 Refresh

Life After Version Control



Life After Version Control

cleandata.do Revision 20657 - TortoiseMerge

File Edit Navigate View Help

cleandata.do Revision 20656

```
1 * Per capita consumption within state
2 egen total_pc_potato = total(pc_potato),
  by(state)
3 egen total_obs = count(pc_potato), by(state)
4 gen leaveout_state_pc_potato = (total_pc_potato
  - pc_potato)
5 / (total_obs - 1)
6
7 * Per capita consumption within metro area
8 egen total_pc_potato = total(pc_potato),
  by(metroarea)
9 egen total_obs = count(pc_potato), by(state)
10 gen leaveout_metro_pc_potato = (total_pc_potato
  - pc_potato)
11 / (total_obs - 1)
.
```

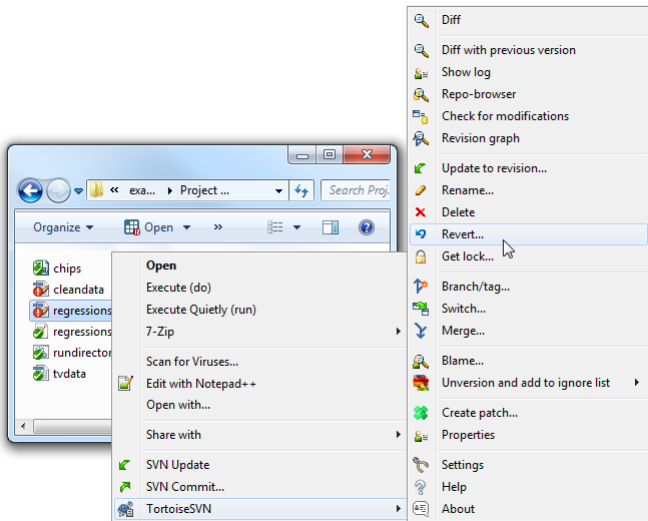
cleandata.do Revision 20657

```
1 * Per capita consumption within state
2 egen total_pc_potato = total(pc_potato),
  by(state)
3 egen total_obs = count(pc_potato), by(state)
4 gen leaveout_state_pc_potato = (total_pc_potato
  - pc_potato)
5 / (total_obs - 1)
6
7 * Per capita consumption within metro area
8 egen total_pc_potato = total(pc_potato),
  by(metroarea)
9 egen total_obs = count(pc_potato), by(state)
10 gen leaveout_metro_pc_potato = (total_pc_potato
  - pc_potato)
11 / (total_obs - 1)
12
13 * Per household consumption within metro area
14 egen total_hh_potato = total(hh_potato),
  by(metroarea)
15 egen total_obs = count(hh_potato), by(state)
16 gen leaveout_metro_hh_potato = (total_hh_potato
  - pc_potato)
```

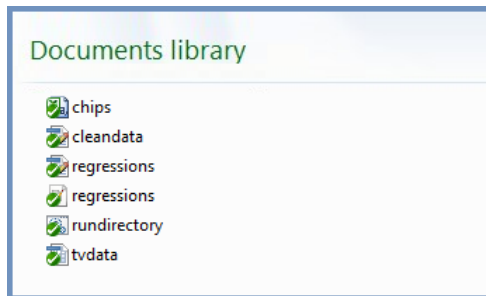
For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel

Left View: ASCII CRLF Right View: ASCII CRLF / + 5 Conflicts: 0 CAP: NUM SCRL

Life After Version Control



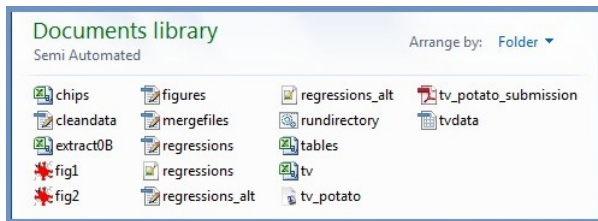
Life After Version Control



- Aside: If you always run `rundirectory.bat` before you commit, you guarantee replicability.

Directories

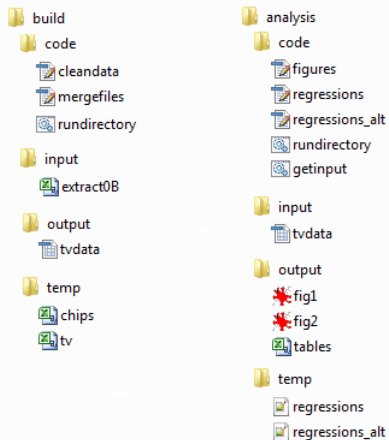
One Directory Does Everything



- Pros: Self-contained, simple
- Cons:
 - Have to rerun everything for every change
 - Hard to figure out dependencies

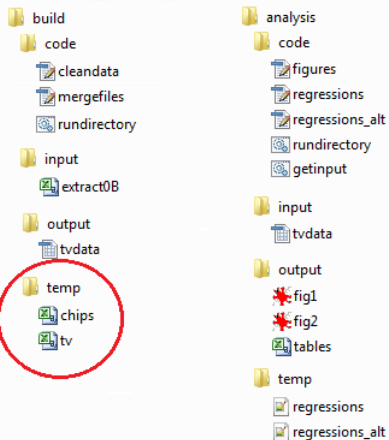
Functional Directories

Documents library

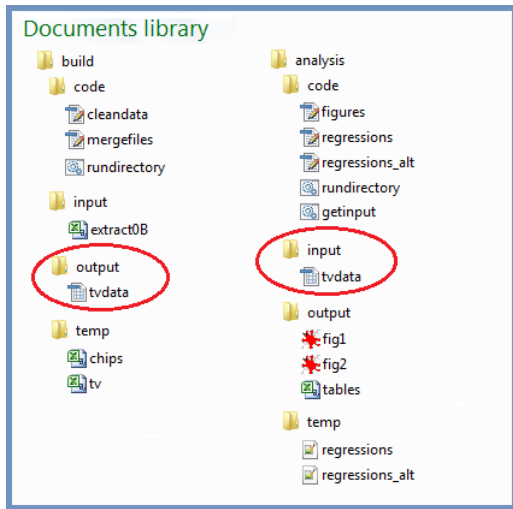


Dependencies Obvious

Documents library



One Resource, Many Projects



Keys

Research Assistant Output

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

Causes for Concern

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 Databases

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

- Each *variable* is an attribute of an *element* of the table
- Each table has a *key*
- Tables are connected by *foreign keys* (state field in the county table)

Steps

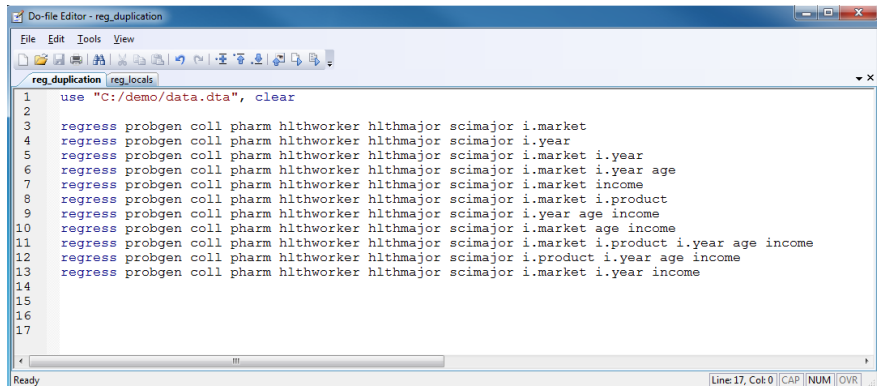
- Store data in normalized format as above
 - Can use flat files, doesn't have to be fancy relational database software
- Construct a second set of files with key transformations
 - e.g., log population
- Merge data together and run analysis

To Come

- What to do with enormous databases?

Abstraction

Rampant Duplication

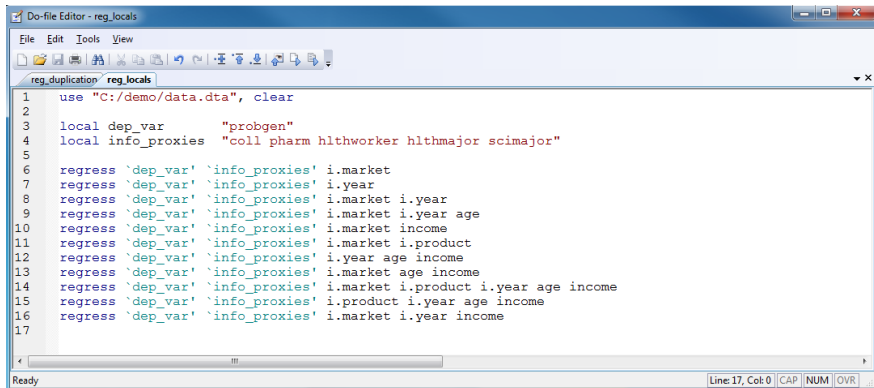


The screenshot shows a Do-file Editor window titled "Do-file Editor - reg_duplication". The window contains a list of 17 lines of Stata commands. Line 1 is "use 'C:/demo/data.dta', clear". Lines 2 through 17 are "regress" commands, each with a different set of variables. The variables are: probgen, coll, pharm, hlthworker, hlthmajor, scimajor, i.market, i.year, i.market i.year, i.market i.year age, i.market income, i.market i.product, i.year age income, i.market age income, i.market i.product i.year age income, i.product i.year age income, and i.market i.year income. The commands are repeated with different combinations of these variables, illustrating the concept of "Rampant Duplication".

```
1 use "C:/demo/data.dta", clear
2
3 regress probgen coll pharm hlthworker hlthmajor scimajor i.market
4 regress probgen coll pharm hlthworker hlthmajor scimajor i.year
5 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.year
6 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.year age
7 regress probgen coll pharm hlthworker hlthmajor scimajor i.market income
8 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.product
9 regress probgen coll pharm hlthworker hlthmajor scimajor i.year age income
10 regress probgen coll pharm hlthworker hlthmajor scimajor i.market age income
11 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.product i.year age income
12 regress probgen coll pharm hlthworker hlthmajor scimajor i.product i.year age income
13 regress probgen coll pharm hlthworker hlthmajor scimajor i.market i.year income
14
15
16
17
```

Ready Line: 17, Col: 0 CAP NUM OVR

Abstracted

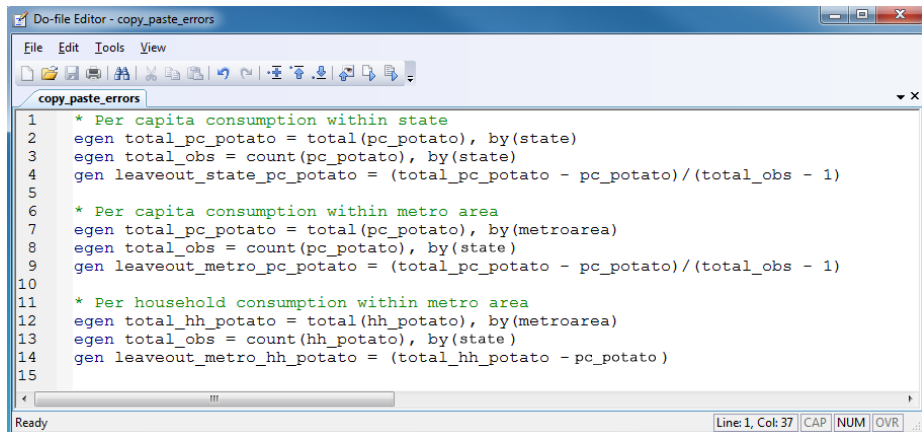


The screenshot shows a Do-file Editor window titled "Do-file Editor - reg_locals". The window contains a list of Stata commands for running regressions on a dataset named "data.dta". The code uses local macros to abstract the dependent variable and the list of independent variables. The commands are as follows:

```
1 use "C:/demo/data.dta", clear
2
3 local dep_var      "probgen"
4 local info_proxies "coll pharm hlthworker hlthmajor scimajor"
5
6 regress `dep_var' `info_proxies' i.market
7 regress `dep_var' `info_proxies' i.year
8 regress `dep_var' `info_proxies' i.market i.year
9 regress `dep_var' `info_proxies' i.market i.year age
10 regress `dep_var' `info_proxies' i.market income
11 regress `dep_var' `info_proxies' i.market i.product
12 regress `dep_var' `info_proxies' i.year age income
13 regress `dep_var' `info_proxies' i.market age income
14 regress `dep_var' `info_proxies' i.market i.product i.year age income
15 regress `dep_var' `info_proxies' i.product i.year age income
16 regress `dep_var' `info_proxies' i.market i.year income
17
```

The status bar at the bottom of the window indicates "Ready" and "Line: 17, Col: 0" with buttons for CAP, NUM, and OVR.

Three Leave-Out Means

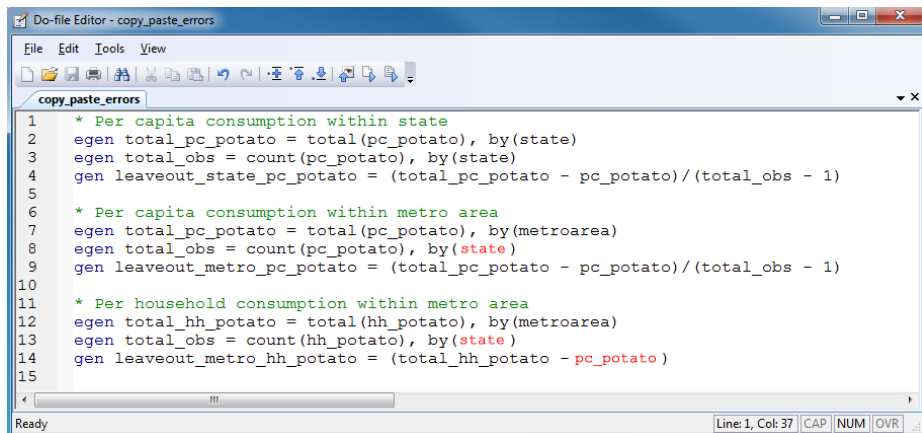


The screenshot shows a Do-file Editor window titled "Do-file Editor - copy_paste_errors". The window contains the following Stata code:

```
1 * Per capita consumption within state
2 egen total_pc_potato = total(pc_potato), by(state)
3 egen total_obs = count(pc_potato), by(state)
4 gen leaveout_state_pc_potato = (total_pc_potato - pc_potato)/(total_obs - 1)
5
6 * Per capita consumption within metro area
7 egen total_pc_potato = total(pc_potato), by(metroarea)
8 egen total_obs = count(pc_potato), by(state)
9 gen leaveout_metro_pc_potato = (total_pc_potato - pc_potato)/(total_obs - 1)
10
11 * Per household consumption within metro area
12 egen total_hh_potato = total(hh_potato), by(metroarea)
13 egen total_obs = count(hh_potato), by(state)
14 gen leaveout_metro_hh_potato = (total_hh_potato - pc_potato)
15
```

The status bar at the bottom indicates "Ready" and "Line: 1, Col: 37".

Copy and Paste Errors

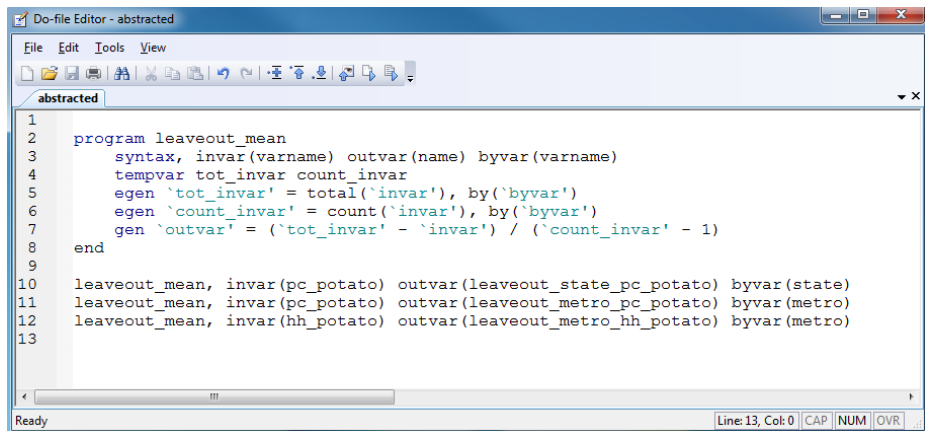


The screenshot shows a window titled "Do-file Editor - copy_paste_errors". The code is as follows:

```
1  * Per capita consumption within state
2  egen total_pc_potato = total(pc_potato), by(state)
3  egen total_obs = count(pc_potato), by(state)
4  gen leaveout_state_pc_potato = (total_pc_potato - pc_potato)/(total_obs - 1)
5
6  * Per capita consumption within metro area
7  egen total_pc_potato = total(pc_potato), by(metroarea)
8  egen total_obs = count(pc_potato), by(state)
9  gen leaveout_metro_pc_potato = (total_pc_potato - pc_potato)/(total_obs - 1)
10
11 * Per household consumption within metro area
12 egen total_hh_potato = total(hh_potato), by(metroarea)
13 egen total_obs = count(hh_potato), by(state)
14 gen leaveout_metro_hh_potato = (total_hh_potato - pc_potato)
15
```

The status bar at the bottom indicates "Ready" and "Line: 1, Col: 37".

Abstracted



The screenshot shows a window titled "Do-file Editor - abstracted" with a menu bar (File, Edit, Tools, View) and a toolbar. The code is as follows:

```
1
2  program leaveout_mean
3      syntax, invar(varname) outvar(name) byvar(varname)
4      tempvar tot_invar count_invar
5      egen `tot_invar' = total(`invar'), by(`byvar')
6      egen `count_invar' = count(`invar'), by(`byvar')
7      gen `outvar' = (`tot_invar' - `invar') / (`count_invar' - 1)
8  end
9
10 leaveout_mean, invar(pc_potato) outvar(leaveout_state_pc_potato) byvar(state)
11 leaveout_mean, invar(pc_potato) outvar(leaveout_metro_pc_potato) byvar(metro)
12 leaveout_mean, invar(hh_potato) outvar(leaveout_metro_hh_potato) byvar(metro)
13
```

The status bar at the bottom indicates "Ready" and "Line: 13, Col: 0" with buttons for CAP, NUM, and OVR.

Documentation

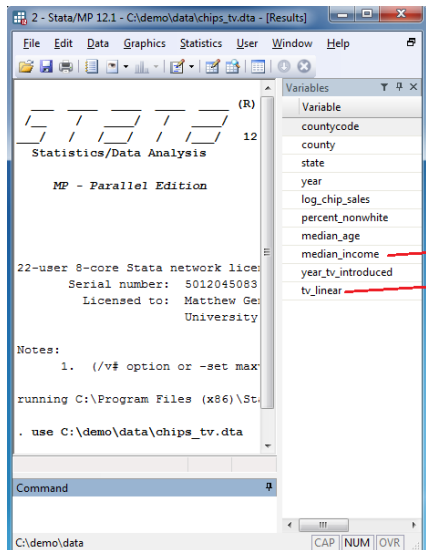
Too Much Documentation

The screenshot shows a Windows Explorer window titled "demo" located at "Computer > OS (C:) > demo". The window displays a list of files and folders. The left sidebar shows the "Computer" view. The main pane shows a table of items:

Name	Date modified	Type	Size
data	6/28/2013 7:13 PM	File folder	
other	6/28/2013 7:13 PM	File folder	
append_years	6/28/2013 6:18 PM	Stata Do-file	1 KB
chips_tv	6/28/2013 7:13 PM	Microsoft Excel C...	2,287 KB
chips_tv_rev	6/28/2013 7:13 PM	Microsoft Excel C...	2,287 KB
log_june20	6/28/2013 12:00 PM	Text Document	3 KB
log_june28	6/28/2013 8:09 PM	Text Document	3 KB
notes_data	6/28/2013 10:05 AM	Text Document	0 KB
notes_data_demo	6/28/2013 10:05 AM	Text Document	0 KB
notes_identification	6/28/2013 10:08 AM	Text Document	0 KB
prepare_data	6/28/2013 6:31 PM	Stata Do-file	1 KB
readme	6/28/2013 10:07 AM	Text Document	0 KB
readme_new	6/28/2013 12:31 PM	Text Document	0 KB
readme_rev	6/28/2013 12:31 PM	Text Document	0 KB
readme_varlist	6/28/2013 8:05 PM	Text Document	1 KB
regressions_june20	6/28/2013 12:32 PM	Microsoft Word D...	101 KB
regressions_june20_mg	6/28/2013 11:49 AM	Microsoft Word D...	101 KB
regressions_june26	6/28/2013 8:31 PM	Microsoft Word D...	43 KB

At the bottom of the window, a status bar indicates: "7 items selected Date modified: 6/28/2013 10:05 AM Date created: 6/28/2013 10:05 AM Size: 222 bytes".

Too Much Documentation



2 - Stata/MP 12.1 - C:\demo\data\chips_tv.dta - [Results]

File Edit Data Graphics Statistics User Window Help

Statistics/Data Analysis

MP - Parallel Edition

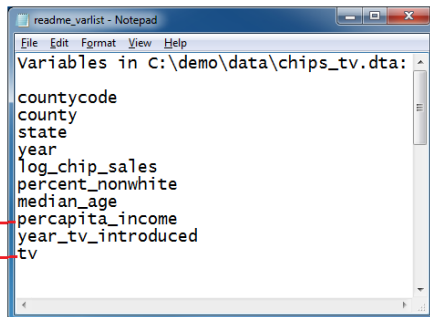
22-user 8-core Stata network license
Serial number: 5012045083
Licensed to: Matthew Ge
University

Notes:
1. (/v# option or -set max
running C:\Program Files (x86)\St
. use C:\demo\data\chips_tv.dta

Command

C:\demo\data

Variable
countycode
county
state
year
log_chip_sales
percent_nonwhite
median_age
median_income
year_tv_introduced
tv_linear



readme_varlist - Notepad

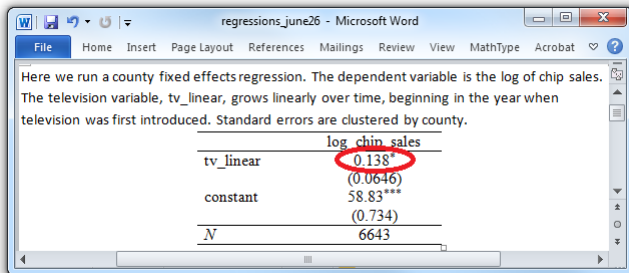
File Edit Format View Help

Variables in C:\demo\data\chips_tv.dta:

countycode
county
state
year
log_chip_sales
percent_nonwhite
median_age
percapita_income
year_tv_introduced
tv

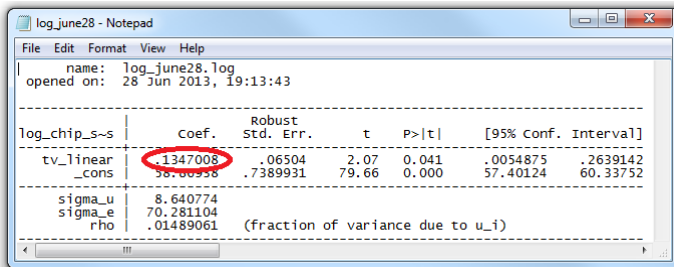
A red question mark is placed to the left of the list, and two red arrows point from it to the 'median_income' and 'tv' entries in the list.

Too Much Documentation



Here we run a county fixed effects regression. The dependent variable is the log of chip sales. The television variable, *tv_linear*, grows linearly over time, beginning in the year when television was first introduced. Standard errors are clustered by county.

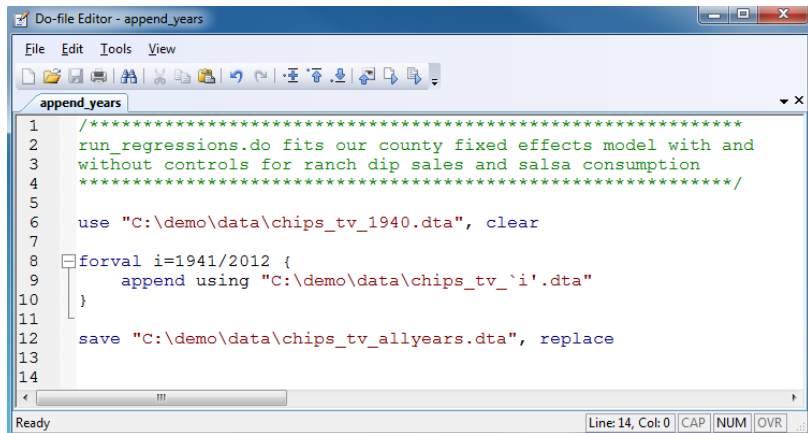
	log_chip_sales
tv_linear	0.138*
	(0.0646)
constant	58.83***
	(0.734)
N	6643



name: log_june28.log
opened on: 28 Jun 2013, 19:13:43

log_chip_s~s	Coef.	Robust Std. Err.	t	P> t	[95% Conf. Interval]	
tv_linear	.1347008	.06504	2.07	0.041	.0054875	.2639142
_cons	58.83338	.7389931	79.66	0.000	57.40124	60.33752
sigma_u	8.640774					
sigma_e	70.281104					
rho	.01489061	(fraction of variance due to u_i)				

Too Much Documentation

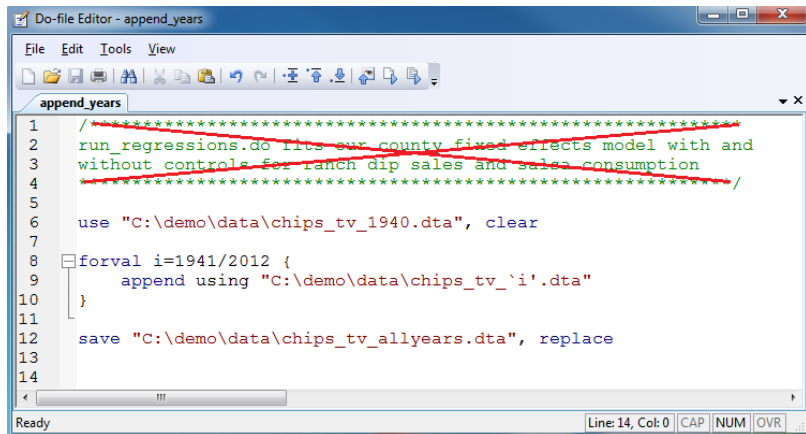


The screenshot shows a window titled "Do-file Editor - append_years" with a menu bar (File, Edit, Tools, View) and a toolbar. The script content is as follows:

```
1  /*****  
2  run_regressions.do fits our county fixed effects model with and  
3  without controls for ranch dip sales and salsa consumption  
4  *****/  
5  
6  use "C:\demo\data\chips_tv_1940.dta", clear  
7  
8  forval i=1941/2012 {  
9      append using "C:\demo\data\chips_tv_`i'.dta"  
10 }  
11  
12 save "C:\demo\data\chips_tv_allyears.dta", replace  
13  
14
```

The status bar at the bottom indicates "Line: 14, Col: 0" and includes buttons for CAP, NUM, and OVR.

Too Much Documentation

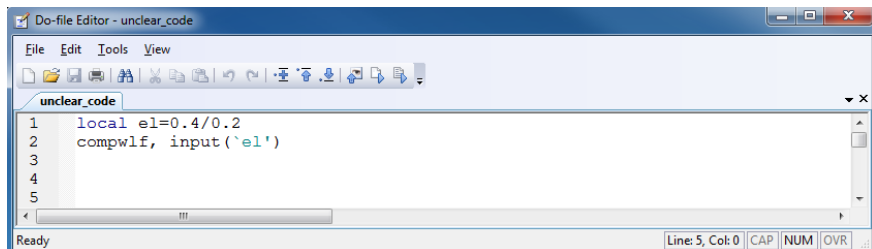


The screenshot shows a window titled "Do-file Editor - append_years" with a menu bar (File, Edit, Tools, View) and a toolbar. The script content is as follows:

```
1  /*****
2  run_regressions.do fits our county fixed effects model with and
3  without controls for ranch dip sales and sales consumption
4  ***** /
5
6  use "C:\demo\data\chips_tv_1940.dta", clear
7
8  forval i=1941/2012 {
9      append using "C:\demo\data\chips_tv_`i'.dta"
10 }
11
12 save "C:\demo\data\chips_tv_allyears.dta", replace
13
14
```

The status bar at the bottom indicates "Line: 14, Col: 0" and "CAP NUM OVR".

Unclear Code



Do-file Editor - unclear_code

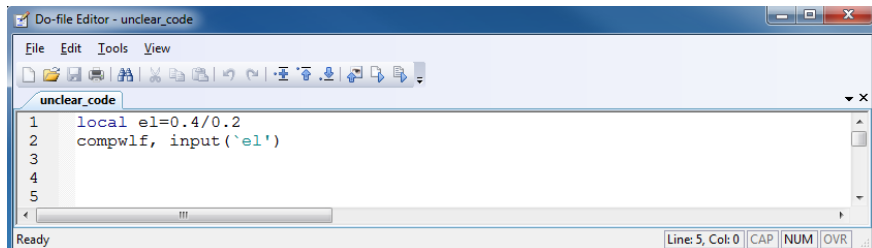
File Edit Tools View

unclear_code

```
1 local e1=0.4/0.2
2 compwlf, input(`e1')
3
4
5
```

Ready Line: 5, Col: 0 CAP NUM OVR

Self-Documenting Code



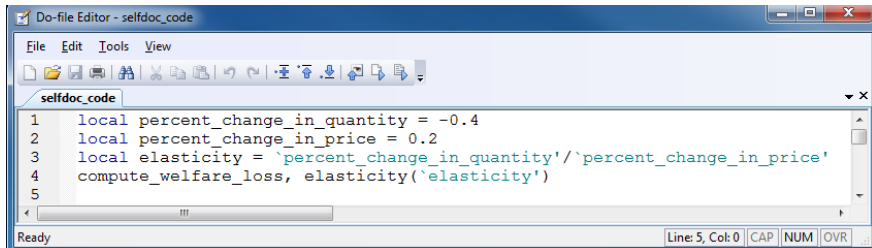
Do-file Editor - unclear_code

File Edit Tools View

unclear_code

```
1 local e1=0.4/0.2
2 compwlf, input(`e1')
3
4
5
```

Ready Line: 5, Col: 0 CAP NUM OVR



Do-file Editor - selfdoc_code

File Edit Tools View

selfdoc_code

```
1 local percent_change_in_quantity = -0.4
2 local percent_change_in_price = 0.2
3 local elasticity = `percent_change_in_quantity' / `percent_change_in_price'
4 compute_welfare_loss, elasticity(`elasticity')
5
```

Ready Line: 5, Col: 0 CAP NUM OVR

Management

A Friendly Chat

potato chips - Message (HTML)

File Message Insert Options Format Text Review

Cut Copy Paste Format Painter Clipboard

Calibri (Body) 16

Basic Text

Address Book Check Names Names

Attach File Attach Item Include

Signature

Follow Up High Importance Low Importance Tags

Zoom

To... Gentzkow, Matthew

Cc...

Subject: potato chips

Send

Hey Matt,

Do you have that robustness check where we control for the amount of ranch dip sold in each county? I am writing the section on dipping sauces and wanted to mention it.

Jesse

See more about: Gentzkow, Matthew.

A Friendly Chat

potato chips - Message (HTML)

File Message Insert Options Format Text Review

Cut Copy Paste Format Painter Clipboard

Calibri (Body) 16 A A

B I U ab? A

Address Book Check Names Attach File Attach Item Signature

Follow Up High Importance Low Importance Tags Zoom

To... Shapiro, Jesse

Cc...

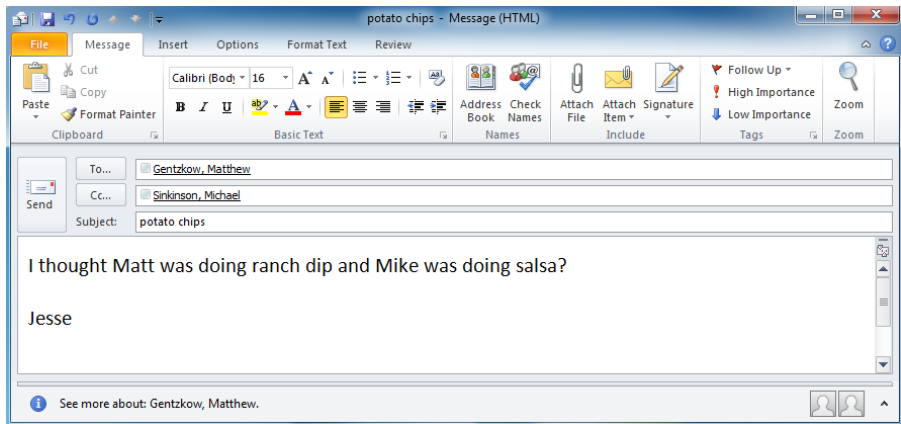
Subject: potato chips

Sorry, I thought you were doing that because it's similar to that other thing you were doing with controlling for salsa sales. Let me know if you want to do it or if you want me to take over.

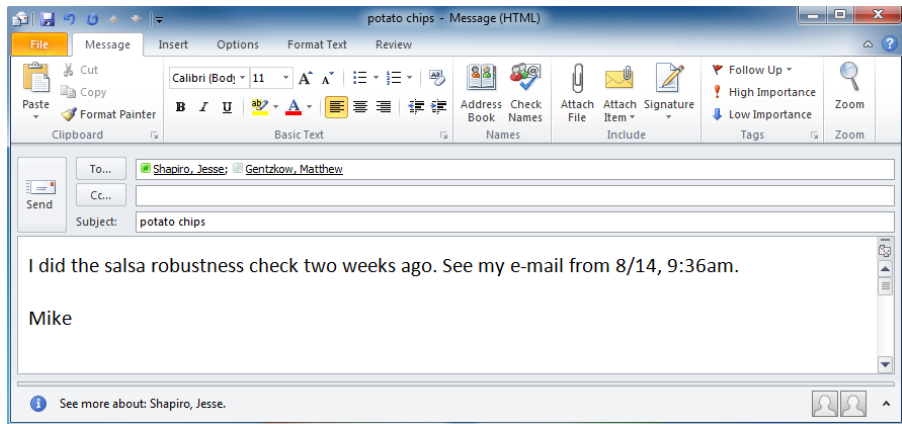
MG

See more about: Shapiro, Jesse.

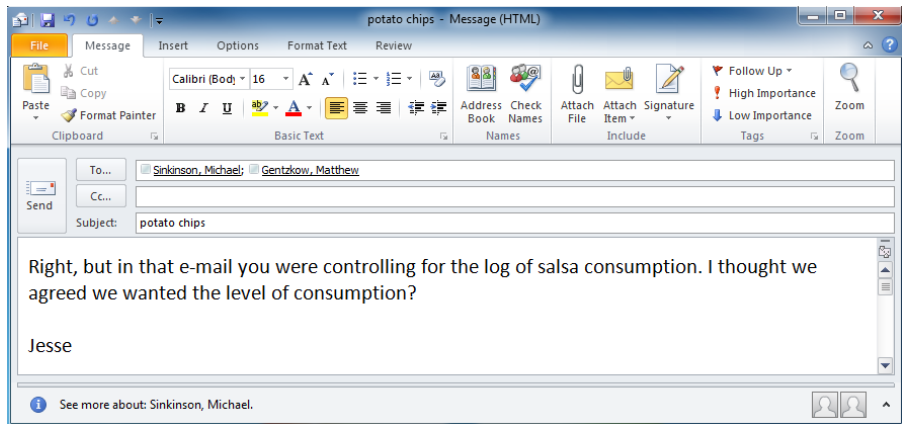
A Friendly Chat



A Friendly Chat



A Friendly Chat



The screenshot shows an Outlook window titled "potato chips - Message (HTML)". The ribbon includes "File", "Message", "Insert", "Options", "Format Text", and "Review". The "Format Text" group is active, showing options for font (Calibri, size 16), bold, italic, underline, text color, background color, and alignment. Other groups include "Clipboard", "Names", "Include", "Tags", and "Zoom".

The email header shows:

- To: Sinkinson, Michael; Gentzkow, Matthew
- CC: [Empty]
- Subject: potato chips

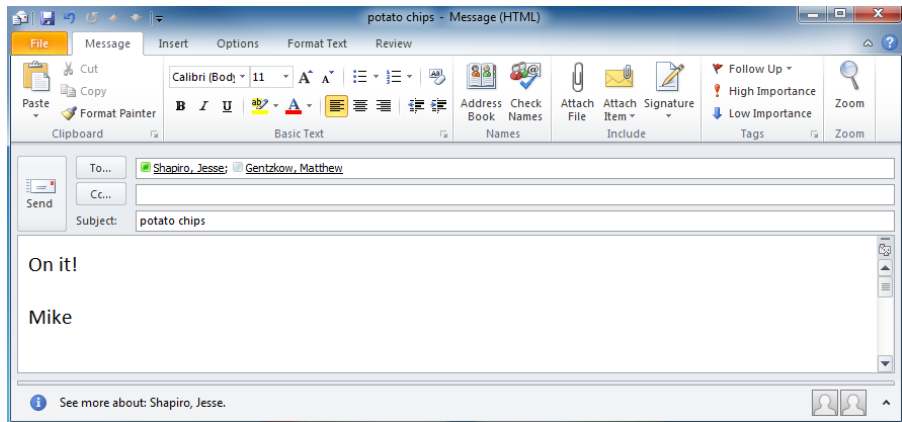
The main body of the email contains the following text:

Right, but in that e-mail you were controlling for the log of salsa consumption. I thought we agreed we wanted the level of consumption?






Jesse

At the bottom, there is a link: "See more about: Sinkinson, Michael." and two profile picture icons.


A Friendly Chat



Task Management

MS Michael S  Due Date    



No Project



 **Salsa Robustness Check**



Run main specifications adding a control for per capita salsa consumption.
Add a line to our robustness table reflecting the results.



[Hide earlier activity](#)



Jesse Shapiro created task. Jun 27
Jesse Shapiro assigned to Michael Sinkinson. Jun 27



 **Michael Sinkinson** On it!
Jun 28 at 2:00pm • 


 **Michael Sinkinson** See the new version of the paper posted in /drafts/Potato Chips and the supporting code in /analysis/Potato Chips. Is this what you had in mind?
Jun 28 at 2:08pm • 

 **Jesse Shapiro** Almost. Our econometric model implies that salsa consumption should enter in levels not logs. Can you revise?
Jun 28 at 2:10pm • 

 **Michael Sinkinson** Ok, how about now?
Jun 28 at 2:12pm • 

 **Jesse Shapiro** Yup, looks good.
Jun 28 at 2:13pm • 

Michael Sinkinson  completed this task
Jun 28 at 2:15pm • 

 Write a comment...

Parting Thoughts

Code and Data

- Data are getting larger
- Research is getting more collaborative
- Need to manage code and data responsibly for collaboration and replicability
- Learn from the pros, not from us

Databases

What is a Database?

- Database Theory
 - Principles for how to store / organize / retrieve data efficiently (normalization, indexing, optimization, etc.)
- Database Software
 - Manages storage / organization / retrieval of data (SQL, Oracle, Access, etc.)
 - Economists rarely use this software because we typically store data in flat files & interact with them using statistical programs
 - When we receive extracts from large datasets (the census, Medicare claims, etc.) someone else often interacts with the database on the back end

Normalization

- “*Database Normalization* is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them.”

Benefits of Normalization

- Efficient storage
- Efficient modification
- **Guarantees coherence**
- **Makes logical structure of data clear**

Indexing

- Medicare claims data for 1997-2010 are roughly 10 TB
- These data are stored at NBER in thousands of zipped SAS files

Indexing

- Medicare claims data for 1997-2010 are roughly 10 TB
- These data are stored at NBER in thousands of zipped SAS files
- To extract, say, all claims for heart disease patients aged 55-65, you would need to read every line of every one of those files
 - THIS IS SLOW!!!



- The obvious solution, long understood for book, libraries, economics journals, and so forth, is to build an index
- Database software handles this automatically
 - Allows you to specify fields that will be often used for lookups, subsetting, etc. to be indexed
 - For the Medicare data, we could index age, gender, type of treatment, etc. to allow much faster extraction

Indexing

- Benefits
 - Fast lookups
 - Easy to police data constraints
- Costs
 - Storage
 - Time
- Database *optimization* is the art of tuning database structure and indexing for a specific set of needs

Data Warehouses

- Traditional databases are optimized for *operational* environments
 - Bank transactions
 - Airline reservations
 - etc.

Data Warehouses

- Traditional databases are optimized for *operational* environments
 - Bank transactions
 - Airline reservations
 - etc.
- Characteristics
 - Many small reads and writes
 - Many users accessing simultaneously
 - Premium on low latency
 - Only care about current state

Data Warehouses

- In analytic / research environments, however, the requirements are different
 - Frequent large reads, infrequent writes
 - Relatively little simultaneous access
 - Value throughput relative to latency
 - May care about history as well as current state
 - Need to create and re-use many custom extracts

Data Warehouses

- In analytic / research environments, however, the requirements are different
 - Frequent large reads, infrequent writes
 - Relatively little simultaneous access
 - Value throughput relative to latency
 - May care about history as well as current state
 - Need to create and re-use many custom extracts
- Database systems tuned to these requirements are commonly called “data warehouses”

Distributed Computing

Distributed Computing

- Definition: Computation shared among many independent processors

Distributed Computing

- Definition: Computation shared among many independent processors
- Terminology
 - Distributed vs. Parallel (latter usually refers to systems with shared memory)
 - Cluster vs. Grid (latter usually more decentralized & heterogeneous)

On Your Local Machine

- Your OS can run multiple processors each with multiple cores
- Your video card has hundreds of cores
- Stata, R, Matlab, etc. can all exploit these resources to do parallel computing

On Your Local Machine

- Your OS can run multiple processors each with multiple cores
- Your video card has hundreds of cores
- Stata, R, Matlab, etc. can all exploit these resources to do parallel computing
- Stata
 - Buy appropriate “MP” version of Stata
 - Software does the rest

On Your Local Machine

- Your OS can run multiple processors each with multiple cores
- Your video card has hundreds of cores
- Stata, R, Matlab, etc. can all exploit these resources to do parallel computing
- Stata
 - Buy appropriate “MP” version of Stata
 - Software does the rest
- R / Matlab
 - Install appropriate add-ins (*parallel* package in R, “parallel computing toolbox” in Matlab)
 - Include parallel commands in code (e.g., *parfor* in place of *for* in Matlab)

- Resources abound
 - University / department computing clusters
 - Non-commercial scientific computing grids (e.g., XSEDE)
 - Commercial grids (e.g., Amazon EC2)

On Cluster / Grid

- Resources abound
 - University / department computing clusters
 - Non-commercial scientific computing grids (e.g., XSEDE)
 - Commercial grids (e.g., Amazon EC2)
- Most of these run Linux w/ distribution handled by a “batch scheduler”
- Write code using your favorite application, then send it to scheduler with a bash script

MapReduce

- MapReduce is a programming model that facilitates distributed computing
 - Developed by Google around 2004, though ideas predate that

MapReduce

- MapReduce is a programming model that facilitates distributed computing
 - Developed by Google around 2004, though ideas predate that
- Most algorithms for distributed data processing can be represented in two steps
 - **Map:** Process individual “chunk” of data to generate an intermediate “summary”
 - **Reduce:** Combine “summaries” from different chunks to produce a single output file

MapReduce

- MapReduce is a programming model that facilitates distributed computing
 - Developed by Google around 2004, though ideas predate that
- Most algorithms for distributed data processing can be represented in two steps
 - **Map:** Process individual “chunk” of data to generate an intermediate “summary”
 - **Reduce:** Combine “summaries” from different chunks to produce a single output file
- If you structure your code this way, MapReduce software will handle all the details of distribution:
 - Partitioning data
 - Scheduling execution across nodes
 - Managing communication between machines
 - Handling errors / machine failures

MapReduce: Examples

- Count words in a large collection of documents
 - Map: Document $i \rightarrow$ Set of $(word, count)$ pairs C_i
 - Reduce: Collapse $\{C_i\}$, summing $count$ within $word$

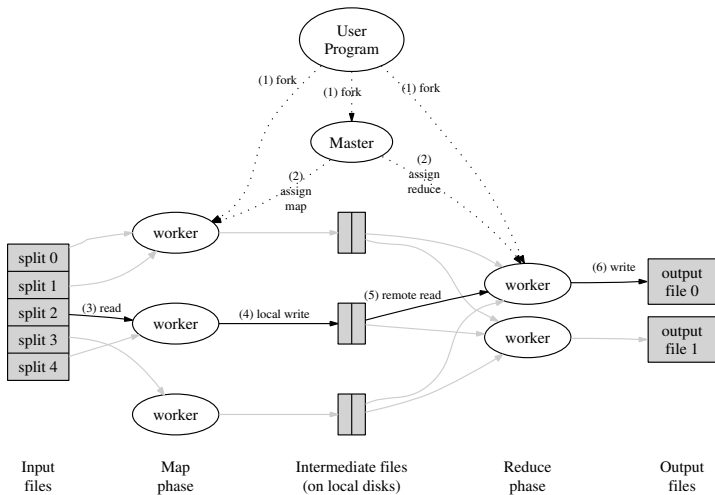
MapReduce: Examples

- Count words in a large collection of documents
 - Map: Document $i \rightarrow$ Set of ($word, count$) pairs C_i
 - Reduce: Collapse $\{C_i\}$, summing $count$ within $word$
- Extract medical claims for 65-year old males
 - Map: Record set $i \rightarrow$ Subset of i that are 65-year old males H_i
 - Reduce: Append elements of $\{H_i\}$

MapReduce: Examples

- Count words in a large collection of documents
 - Map: Document $i \rightarrow$ Set of $(word, count)$ pairs C_i
 - Reduce: Collapse $\{C_i\}$, summing $count$ within $word$
- Extract medical claims for 65-year old males
 - Map: Record set $i \rightarrow$ Subset of i that are 65-year old males H_i
 - Reduce: Append elements of $\{H_i\}$
- Compute marginal regression for text analysis (e.g., Gentzkow & Shapiro 2010)
 - Map: Counts x_{ij} of phrase $j \rightarrow$ Parameters $(\hat{\alpha}_j, \hat{\beta}_j)$ from
$$E(x_{ij}|y_i) = \alpha_j + \beta_j x_{ij}$$
 - Reduce: Append $\{\hat{\alpha}_j, \hat{\beta}_j\}$

MapReduce: Model



MapReduce: Implementation

- MapReduce is the original software developed by Google
- Hadoop is the open-source version most people use (developed by Apache)
- Amazon has a hosted implementation (Amazon EMR)

MapReduce: Implementation

- MapReduce is the original software developed by Google
- Hadoop is the open-source version most people use (developed by Apache)
- Amazon has a hosted implementation (Amazon EMR)
- How does it work?
 - Write your code as two functions called *map* and *reduce*
 - Send code & data to scheduler using bash script

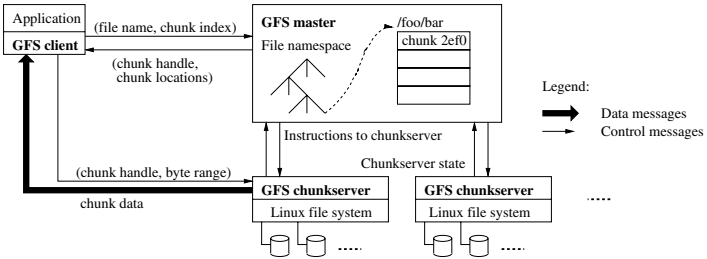
Distributed File Systems

- Data transfer is the main bottleneck in distributed systems
- For big data, it makes sense to distribute data as well as computation
 - Data broken up into chunks, each of which lives on a separate node
 - File system keeps track of where the pieces are and allocates jobs so computation happens “close” to data whenever possible

Distributed File Systems

- Data transfer is the main bottleneck in distributed systems
- For big data, it makes sense to distribute data as well as computation
 - Data broken up into chunks, each of which lives on a separate node
 - File system keeps track of where the pieces are and allocates jobs so computation happens “close” to data whenever possible
- Tight coupling between MapReduce software and associated file systems
 - MapReduce → Google File System (GFS)
 - Hadoop → Hadoop Distributed File System (HDFS)
 - Amazon EMR → Amazon S3

Distributed File Systems



Scenarios

Scenario 1: Not-So-Big Data

- *My data is 100 gb or less*

Scenario 1: Not-So-Big Data

- *My data is 100 gb or less*
- Advice
 - Store data locally in flat files (csv, Stata, R, etc.)
 - Organize data in normalized tables for robustness and clarity
 - Run code serially or (if computation is slow) in parallel

Scenario 2: Big Data, Small Analysis

- *My raw data is > 100 gb, but the extracts I actually use for analysis are $\ll 100$ gb*

Scenario 2: Big Data, Small Analysis

- *My raw data is > 100 gb, but the extracts I actually use for analysis are $\ll 100$ gb*
- Example
 - Medicare claims data \rightarrow analyze heart attack spending by patient by year
 - Nielsen scanner data \rightarrow analyze average price by store by month

Scenario 2: Big Data, Small Analysis

- *My raw data is > 100 gb, but the extracts I actually use for analysis are $\ll 100$ gb*
- Example
 - Medicare claims data \rightarrow analyze heart attack spending by patient by year
 - Nielsen scanner data \rightarrow analyze average price by store by month
- Advice
 - Store data in relational database optimized to produce analysis extracts efficiently
 - Store extracts locally in flat files (csv, Stata, R, etc.)
 - Organize extracts in normalized tables for robustness and clarity
 - Run code serially or (if computation is slow) in parallel

Scenario 2: Big Data, Small Analysis

- *My raw data is > 100 gb, but the extracts I actually use for analysis are $\ll 100$ gb*
- Example
 - Medicare claims data \rightarrow analyze heart attack spending by patient by year
 - Nielsen scanner data \rightarrow analyze average price by store by month
- Advice
 - Store data in relational database optimized to produce analysis extracts efficiently
 - Store extracts locally in flat files (csv, Stata, R, etc.)
 - Organize extracts in normalized tables for robustness and clarity
 - Run code serially or (if computation is slow) in parallel
- Note: Gains to database increase for more structured data. For completely unstructured data, you may be better off using distributed file system + map reduce to create extracts.

Scenario 3: Big Data, Big Analysis

- *My data is > 100 GB and my analysis code needs to touch all of the data*

Scenario 3: Big Data, Big Analysis

- *My data is > 100 GB and my analysis code needs to touch all of the data*
- Example
 - 2 TB of SEC filing text → run variable selection using all data

Scenario 3: Big Data, Big Analysis

- *My data is > 100 GB and my analysis code needs to touch all of the data*
- Example
 - 2 TB of SEC filing text → run variable selection using all data
- Advice
 - Store data in distributed file system
 - Use MapReduce or other distributed algorithms for analysis