

Stata hybrids: updates & ideas

James Fiedler
Stata Conference 2014

Abstract:

At last year's Stata conference, I presented projects that facilitate the combined use of Stata and Python. One project provides the ability to use Python within Stata via a C plugin. The other project provides a custom Python class that can be used to open, modify, and save Stata datasets. In this talk, I will begin by describing some modifications and extensions to these projects. I will then present a few new ideas for useful combinations of Stata with other tools. Some of these ideas can be realized using last year's Python projects, some using JavaScript and a web browser.

Support

where I work

NASA Johnson Space Center

who I work for

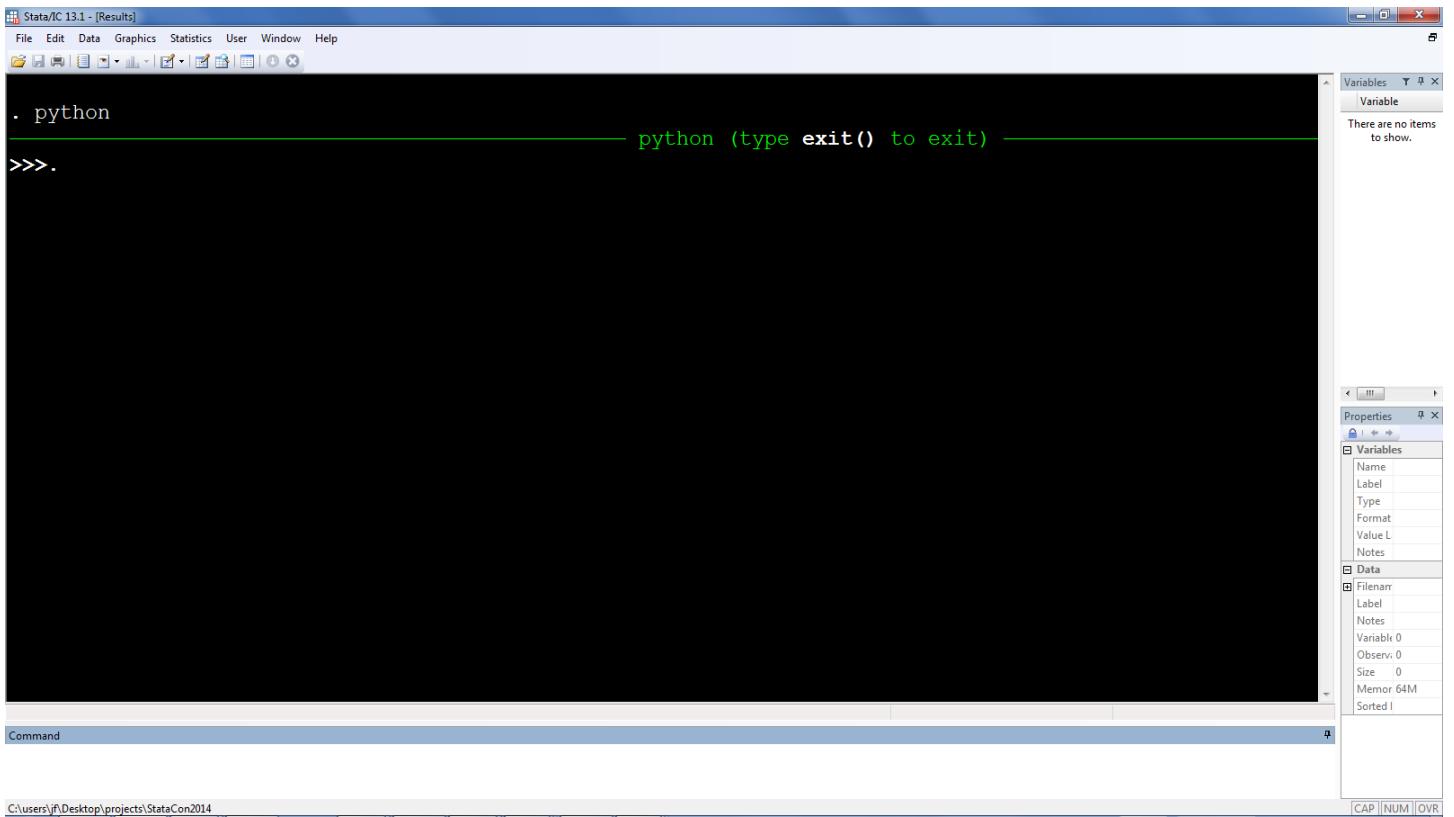
Universities Space Research Association

Last year ...

2

The past couple years I've been playing with the idea of combining Stata with other software to extend its functionality. I've mostly been using the Python programming language and third-party Python modules.

Last year I demonstrated two Stata/Python projects.



The first project from last year is a plugin for using Python directly in Stata. As shown here, the `python` command puts the user in an interactive Python session.

```

Stata/IC 13.1 - [Results]
File Edit Data Graphics Statistics User Window Help
python (type exit() to exit)
Variables
Variable
There are no items to show.

. python
>>>. from stata_dta import open_dta
>>>. dta = open_dta("auto_copy.dta")
(1978 Automobile Data)

>>>. dta.describe()

      obs:          74
      vars:          12
      size:        3182
                                1978 Automobile Data
                                24 Jul 2014 13:51
                                (_dta has notes)

      variable   storage  display    value
      name       type     format   label
                                variable label

make        str18  %18s      Make and Model
price       int     %8.0gc    Price
mpg        int     %8.0g     Mileage (mpg)
rep78      int     %8.0g     Repair Record 1978
headroom    float   %6.1f     Headroom (in.)
trunk       int     %8.0g     Trunk space (cu. ft.)
weight      int     %8.0gc    Weight (lbs.)

Command
C:\users\j\Desktop\projects\StataCon2014

```

The second project from last year is a Python package `stata_dta`, for opening, modifying, and saving `dta` files in Python.

Here I use the interactive Python session to demonstrate `stata_dta`. First I import the function `open_dta`, then use that function to create a Python variable `dta` containing the information in `auto_copy.dta`, and then finally call the `describe` method on `dta`. All of this is being done in Python. The variables panel shows that there is no dataset loaded in Stata.

Changes

The `python` command will now search for Python files in the ado path.

A new `st_mirror()` function which returns a dynamic view onto the Stata dataset. (The `st_view()` function returns a static view.)

3

I've made a few changes to last year's projects. Most of these changes are in line with making these projects more convenient.

I will demonstrate the new `st_mirror()` function on the next page.

The screenshot shows the Stata/IC 13.1 interface with the 'Results' window active. In the command window, the following Python code is run:

```
. python
>>>. v = st_view()
>>>. m = st_mirror()
>>>. v.rows
()
>>>. m.rows
()
>>>.
```

A green horizontal bar at the top of the code editor indicates the current line of execution. To the right of the code editor, there are two floating windows: a 'Variables' window showing 'There are no items to show.' and a 'Properties' window for a variable named 'v' which lists its properties: Name, Label, Type, Format, Value L, Notes, Data, and Filenam. The 'Data' section shows 'Variable 0', 'Observ: 0', 'Size 0', and 'Memor 64M'. A status bar at the bottom shows the path 'C:\users\j\Desktop\projects\StataCon2014'.

The Python plugin includes a function `st_view`, which, like Mata's `st_view`, provides a *static* view of the Stata dataset. The plugin now includes a function `st_mirror` to provide a *dynamic* view. Here I will demonstrate the difference

In the picture above, no dataset is loaded in Stata. I create Python variable `v` using `st_view` and Python variable `m` using `st_mirror`. I query the `rows` in each, and for both there are no rows.

```
>>>. v.rows
()
>>>. m.rows
()
>>>. exit()

. sysuse auto
(1978 Automobile Data)

. python
python (type exit() to exit)
>>>. v.rows
()
>>>. m.rows
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73)
>>>.
```

Continuing from the last page, here I exit Python, load the auto dataset, and re-enter Python.

I again query the rows in `v` and `m`. Again the view `v` contains no rows, but `m` contains all the rows in the loaded dataset. This is the main difference between `st_view` and `st_mirror`. The number of rows and columns in `v` is fixed based on what was loaded when it was created. The number of rows and columns in `m` always reflects what is current in Stata.

Changes

A new `stata_math` module containing math functions that know what to do with Stata missing values.

Various other small changes. See documentation for more info.

4

The last major addition to mention here is the `stata_math` module.

Documentation for the Python plugin can be found at

https://github.com/jrfiedler/python-in-stata/raw/master/python_plugin.pdf

New ideas

5

Now for a few new ideas about extending Stata functionality.

Physical units

There's no dedicated place for recording physical units.

There's no simple way to convert from one unit to another.

6

First idea: Add functionality for recording and converting physical units.

```

. python
python (type exit() to exit)
>>>. from units_dta import UDta
>>>. dta = UDta("auto_copy.dta")
(1978 Automobile Data)
>>>. dta.describe()

      obs:          74           1978 Automobile Data
      vars:          12           24 Jul 2014 13:51
      size:        3182           (_dta has notes)

variable   storage   display   value
name      type     format    label      variable label
make      str18   %-18s      Make and Model
price     int      %8.0gc    Price
mpg       int      %8.0g     Mileage (mpg)
rep78    int      %8.0g     Repair Record 1978
headroom  float    %6.1f     Headroom (in.)
trunk    int      %8.0g     Trunk space (cu. ft.)
weight   int      %8.0gc    Weight (lbs.)

```

To explore this idea, I added on to the `stata_dta` module described earlier to create the module `units_dta`.

Here I import `UDta` from `units_dta`, use `UDta` to open `auto_copy.dta`, and call the `describe` method. (So far, this demonstration matches the demonstration for `stata_dta`.)

Notice that in the auto dataset units have been recorded in the variable label.

```

make          str18   %18s      Make and Model
price         int     %8.0gc    Price
mpg          int     %8.0g     Mileage (mpg)
rep78        int     %8.0g     Repair Record 1978
headroom     float   %6.1f     Headroom (in.)
trunk        int     %8.0g     Trunk space (cu. ft.)
weight       int     %8.0gc    Weight (lbs.)
length       int     %8.0g     Length (in.)
turn         int     %8.0g     Turn Circle (ft.)
displacement int     %8.0g     Displacement (cu. in.)
gear_ratio   float   %6.2f     Gear Ratio
foreign      byte    %8.0g     origin   Car type
Sorted by: foreign

>>>. dta.units_set("head", "inch")
>>>. dta.units_set("mpg", "mpg")
>>>. dta.rename("mpg", "efficiency")
>>>.

```

Variables

Variable
make
price
mpg
rep78
headroom
trunk
weight
length
turn
displacement
gear_ratio
foreign

Properties

Variables
Name make
Label Make and Model
Type str18
Format %18s
Value L
Notes
Data
Filenam: auto.dta
Label 1978 Auto
Notes
Variable 12
Observ: 74
Size 3.11K
Memor 64M
Sorted I foreign

I will move units out of the variable label and into a dedicated place. One part of the added functionality is the `units_set` method. Here (continuing from the `describe` on the previous page) I use `units_set` to say that `headroom` is measured in inches and `mpg` is measured in mpg.

Predicting that I might convert the `mpg` variable to other units, I change its name to `efficiency`.

Stata/IC 13.1 - C:\Program Files (x86)\Stata13\ado\base\auto.dta - [Results]

File Edit Data Graphics Statistics User Window Help

```
>>>. dta = UDTa("auto_units.dta")
(1978 Automobile Data)

>>>. dta.describe()

      obs:          74          1978 Automobile Data
      vars:          12          13 Apr 2013 17:45
      size:        3182          (_dta has notes)

      storage   display   value
variable name    type    format   label     variable label

make           str18  %-18s  Make and Model
price          int    %8.0gc  Price
efficiency    int    %8.0g  Mileage
rep78          int    %8.0g  Repair Record 1978
headroom       float   %6.1f  Headroom
trunk           int    %8.0g  Trunk space
weight          int    %8.0gc Weight
length          int    %8.0g  Length
turn            int    %8.0g  Turn circle
displacement   int    %8.0g  Displacement
gear_ratio     float   %6.2f  Gear Ratio
```

Variables

Variable	
make	
price	
mpg	
rep78	
headroom	
trunk	
weight	
length	
turn	
displacement	
gear_ratio	
foreign	

Properties

Name	make
Label	Make and Model
Type	str18
Format	%-18s
Value	L
Notes	

Data

Filename:	auto.dta
Label:	1978 Auto
Notes:	
Variable:	12
Observ:	74
Size:	3.11K
Memor:	64M
Sorted:	I foreign

Command

C:\users\j\Desktop\projects\StataCon2014 CAP NUM OVR

I could continue setting units on the other variables, but instead I will load a dataset where that has already been done. Also, I have removed the units information from the variable labels.

The screenshot shows the Stata/IC 13.1 interface with the following details:

- Variables View:** Shows a list of variables: headroom, trunk, weight, length, turn, displacement, gear_ratio, foreign, price, mpg, headroom, trunk, weight, length, turn, displacement, gear_ratio, and foreign.
- Properties View:** Shows the properties for the variable "make".
 - Name: make
 - Label: Make and Type
 - Type: str18
 - Format: %18s
 - Value: L
 - Notes:
- Data View:** Shows file information for "auto.dta".
 - Filename: auto.dta
 - Label: 1978 Auto
 - Notes:
 - Variables: 12
 - Observations: 74
 - Size: 3.11K
 - Memory: 64M
 - Sorted by: foreign
- Command History:** Shows the command "dta.units_list()".
- Status Bar:** Shows the path "C:\users\j\Desktop\projects\StataCon2014" and the CAP|NUM|OVR indicator.

To see what units have been defined, I use the `units_list` method.

```

Stata/IC 13.1 - C:\Program Files (x86)\Stata13\ado\base\auto.dta - [Results]
File Edit Data Graphics Statistics User Window Help
turn: ft
displacement: inch**3

>>. dta.summ()

  Variable    Units      Obs      Mean   Std. Dev.      Min      Max
  make          0
  price        USD1978    74    6165.26    2949.5    3291    15906
  efficiency    mpg        74    21.2973    5.7855     12      41
  rep78         rep78      69    3.4058    0.989932     1       5
  headroom      inch       74    2.99324    0.845995    1.5       5
  trunk          ft**3      74    13.7568    4.2774      5      23
  weight         lb         74    3019.46    777.194   1760    4840
  length         inch       74    187.932    22.2663   142     233
  turn            ft         74    39.6486    4.39935    31      51
  displacement   inch**3    74    197.297   91.8372    79     425
  gear_ratio      74    3.01486    0.456287    2.19     3.89
  foreign         74    0.297297    0.460188     0       1

>>.

Command

```

The screenshot shows the Stata 13.1 interface with the 'auto.dta' dataset loaded. The main window displays the results of the 'dta.summ()' command, showing summary statistics for various variables including 'make', 'price', 'efficiency', 'rep78', 'headroom', 'trunk', 'weight', 'length', 'turn', 'displacement', 'gear_ratio', and 'foreign'. The 'Variables' pane on the right lists these variables with their types and formats. The 'Properties' pane provides detailed information for the 'make' variable, such as its name, label ('Make and Model'), type ('str18'), format ('%18s'), and value labels ('Value L'). The status bar at the bottom indicates the file path 'C:\users\j\Desktop\projects\StataCon2014'.

If we go to the trouble of recording units, it might be helpful to be reminded of them when we use common Stata commands. The `units_dta` module has a modified version of `summarize` that displays units in the second column, as demonstrated above.

```

Stata/IC 13.1 - C:\Program Files (x86)\Stata13\ado\base\la\auto.dta - [Results]
File Edit Data Graphics Statistics User Window Help
foreign | 74 0.297297 0.460188 0 1

>>>. dta.summ("head eff")

      Variable    Units     Obs      Mean   Std. Dev.      Min      Max
headroom |    inch     74  2.99324  0.845995     1.5       5
efficiency |    mpg     74  21.2973   5.7855     12      41

>>>. dta.units_convert("head", "cm")
>>>. dta.units_convert("eff", "lp100km")
>>>. dta.summ("head eff")

      Variable    Units     Obs      Mean   Std. Dev.      Min      Max
headroom |     cm     74  7.60284  2.14883     3.81     12.7
efficiency | lp100km  74  11.8061  3.01041   5.73694    19.6012

>>.

```

Variables

- Variable
- make
- price
- mpg
- re78
- headroom
- trunk
- weight
- length
- turn
- displacement
- gear_ratio
- foreign

Properties

- Variables
- Name: make
- Label: Make and Type
- Type: str18
- Format: %18s
- Value L
- Notes
- Data
- Filenam: auto.dta
- Label: 1978 Auto
- Notes
- Variable: 12
- Observ: 74
- Size: 3.11K
- Memor: 64M
- Sorted: foreign

Now suppose you give this dataset to someone outside of the US. They might prefer that headroom be measured in something other than inches, and they might prefer efficiency to be measured in something other than mpg.

In the picture above I first called `summarize` on just `headroom` and `efficiency` (this will be useful in a moment). I then used the `units_convert` method to convert the units on `headroom` from `inch` to `cm` and the units on `efficiency` from `mpg` to `lp100km` (my abbreviation for *liters per 100 km*). Finally, I called `summarize` again to compare to the previous `summarize`. The labeled units have changed, but the values in the dataset have also been converted.

Side note: I predefined `lp100km` in `units_dta`. If I hadn't I would first have to use the method `units_define` to define it.

Physical units

Made with

1. the `stata_dta` module
2. the `Sympy` module

7

As I said earlier, this example was built on top of my `stata_dta` module. Most of the work to convert units is done by the `Sympy` module.

Multimedia spreadsheet

Stata 13 added the ability to include images, videos, pdfs, etc. in our dta files.

But there's no convenient way to view them.

8

Second idea: A multimedia spreadsheet that allows us to view embedded images as images, hear embedded audio, etc, in the same spreadsheet as the other values of our dataset.

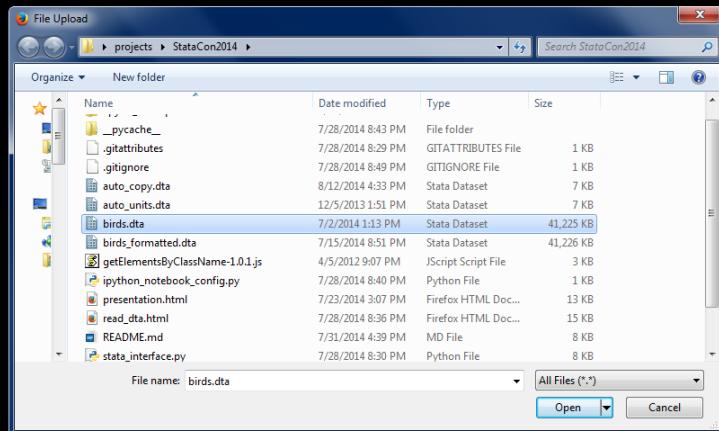
No file selected.

In fact, all of us have a rich multimedia viewer on our computers: our web browsers.

So, to explore this idea I use an html page and a third-party spreadsheet library.

The html page is pictured above.

No file selected.



I click on the “Browse” button, and choose `birds.dta`, a Stata dataset with embedded photos and audio.

Browse... birds.dta

commo...	wikip...	photo	photo...	photo...	photo...	binom...	audio...	audio...	audio_sample	audio...	photo...
sedge ...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Cistot...	http:/...		[object Blob]		Jan 2,...
downy ...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Picoid...	http:/...	http:/...	[object Blob]	[objec...	Jan 2,...
red-ta...	http:/...	[object Blob]	http:/...	http:/...	Forest...	Buteo ...	http:/...		[object Blob]		Jan 3,...
common...	http:/...	[object Blob]	http:/...	http:/...	Dick D...	Phasia...	http:/...	http:/...	[object Blob]	[objec...	Jan 3,...
common...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Quisca...	http:/...		[object Blob]		Jan 2,...
red-wi...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Agelai...	http:/...	http:/...	[object Blob]	[objec...	Jan 2,...
common...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Acanth...	http:/...	http:/...	[object Blob]	[objec...	Jan 2,...
marsh ...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Cistot...	http:/...		[object Blob]		Jan 2,...
americ...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Turdus...	http:/...	http:/...	[object Blob]	[objec...	Jan 2,...
house ...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Haemor...	http:/...		[object Blob]		Jan 2,...
indigo...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Passer...	http:/...	http:/...	[object Blob]	[objec...	Jan 2,...
savann...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Passer...	http:/...		[object Blob]		Jan 2,...
americ...	http:/...	[object Blob]	http:/...	http:/...	Larry ...	Spizel...	http:/...	http:/...	[object Blob]	[objec...	Jan 2,...
dickci...	http:/...					Spiza ...	http:/...	http:/...	[object Blob]	[objec...	
harris...	http:/...					Zonotr...	http:/...		[object Blob]		

When the dataset first opens it looks like this. Most of the columns contain plain text. Notice the third column, photo, which contains binary data. At the moment the spreadsheet only knows that it's binary data (Blob = binary large object).

Browse... birds.dta

commo...	wikip...	photo	▼ photo...	photo...	photo...	binom...	audio...	audio...	audio_sample	audio...	photo...
sedge ... http://...	[object Blob]	▼ Sort Ascending	arry ... Cistot...	arry ... Cistot...	arry ... Cistot...	[object Blob]	Jan 2,...				
downy ... http://...	[object Blob]	▼ Sort Descending	arry ... Picoid...	arry ... Picoid...	arry ... Picoid...	[object Blob]					
red-ta...	[object Blob]		as image	as image	as image	Buteo ...	Buteo ...	Buteo ...	[object Blob]	[object Blob]	Jan 3,...
common...	[object Blob]		as audio	as audio	as audio	Phasia...	Phasia...	Phasia...	[object Blob]	[object Blob]	[object Blob]
common...	[object Blob]					Quisca...	Quisca...	Quisca...	[object Blob]	[object Blob]	Jan 2,...
red-wi...	[object Blob]					Larry ...	Larry ...	Larry ...	[object Blob]	[object Blob]	[object Blob]
common...	[object Blob]					Agelai...	Agelai...	Agelai...	[object Blob]	[object Blob]	[object Blob]
marsh ...	[object Blob]					Larry ...	Larry ...	Larry ...	[object Blob]	[object Blob]	Jan 2,...
americ...	[object Blob]					Cistot...	Cistot...	Cistot...	[object Blob]	[object Blob]	[object Blob]
house ...	[object Blob]					Turdus...	Turdus...	Turdus...	[object Blob]	[object Blob]	[object Blob]
indigo...	[object Blob]					Haemor...	Haemor...	Haemor...	[object Blob]	[object Blob]	Jan 2,...
savann...	[object Blob]					Passer...	Passer...	Passer...	[object Blob]	[object Blob]	[object Blob]
americ...	[object Blob]					Spizel...	Spizel...	Spizel...	[object Blob]	[object Blob]	[object Blob]
dickci...	[object Blob]					Spiza ...	Spiza ...	Spiza ...	[object Blob]	[object Blob]	[object Blob]
harris...	[object Blob]					Zonotr...	Zonotr...	Zonotr...	[object Blob]	[object Blob]	[object Blob]

In the header, next to “photo”, I can click to open a menu. The menu includes the options to decode “as image” or “as audio”.

Browse... birds.dta

common_na...	wikip...	photo	photo...	photo...	photo...	binom...	audio...	audio...	audio_sample	audio...	photo...
sedge wren	http://...		http://...	http://...	Larry ... Cistot...	http://...	[object Blob]			Jan 2,...	
downy woo...	http://...		http://...	http://...	Larry ... Picoid...	http://...	[object Blob]	[objec...	Jan 2,...		
red-taile...	http://...		Buteo ...	http://...		[object Blob]			Jan 3,...		
common ph...	http://...		Phasia...	http://...	http://...	[object Blob]	[objec...	Jan 3,...			
common gr...	http://...		Quisca...	http://...		[object Blob]			Jan 2,...		
red-winge...	http://...		Agelai...	http://...	http://...	[object Blob]	[objec...	Jan 2,...			
common re...	http://...		Acanth...	http://...	http://...	[object Blob]	[objec...	Jan 2,...			
marsh wren	http://...		Cistot...	http://...		[object Blob]			Jan 2,...		
american ...	http://...		Turdus...	http://...	http://...	[object Blob]	[objec...	Jan 2,...			
house fin...	http://...		Haemor...	http://...		[object Blob]			Jan 2,...		
indigo bu...	http://...		http://...	http://...	Larry ... Passer...	http://...	[object Blob]	[objec...	Jan 2,...		
savannah ...	http://...		http://...	http://...	Larry ... Passer...	http://...	[object Blob]			Jan 2,...	
american ...	http://...		http://...	http://...	Larry ... Spizel...	http://...	[object Blob]	[objec...	Jan 2,...		
dickcissel	http://...		Spiza ...	http://...	http://...	[object Blob]			[objec...		
harris's ...	http://...		Zonotr...	http://...		[object Blob]					

When I choose to decode as image, the decoded images appear. The images are small, so I put my mouse over one to show it larger (the cursor is invisible here).

Browse... birds.dta

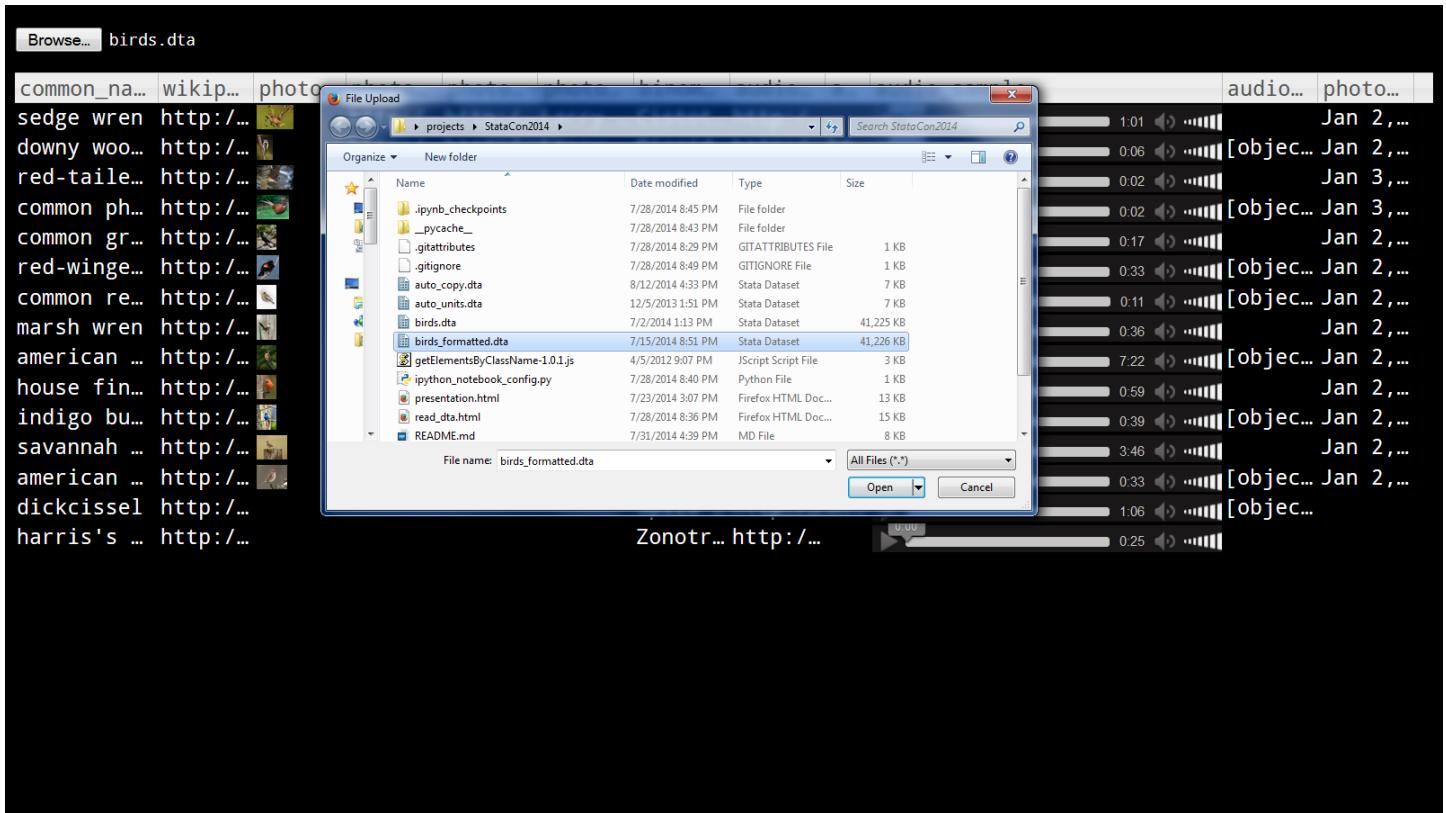
common_na...	wikip...	photo	photo...	photo...	photo...	binom...	audio...	audio...	audio_sample	▼ audio...	photo...
sedge wren	http:/...		http:/...	http:/...	Larry ... Cistot...	http:/...	[object Blob]			▲ Sort Ascending	
downy woo...	http:/...		http:/...	http:/...	Larry ... Picoid...	http:/...	http:/...	[object Blob]		▼ Sort Descending	
red-taile...	http:/...		http:/...	http:/...	Forest...	Buteo ...	http:/...	[object Blob]		as image	
common ph...	http:/...		http:/...	http:/...	Dick D...	Phasia...	http:/...	http:/...	[object Blob]	as audio	
common gr...	http:/...		http:/...	http:/...	Larry ... Quisca...	http:/...		[object Blob]		[objec...	Jan 2,...
red-winge...	http:/...		http:/...	http:/...	Larry ... Agelai...	http:/...	http:/...	[object Blob]		[objec...	Jan 2,...
common re...	http:/...		http:/...	http:/...	Larry ... Acanth...	http:/...	http:/...	[object Blob]		Jan 2,...	
marsh wren	http:/...		http:/...	http:/...	Larry ... Cistot...	http:/...		[object Blob]		[objec...	Jan 2,...
american ...	http:/...		http:/...	http:/...	Larry ... Turdus...	http:/...	http:/...	[object Blob]		[objec...	Jan 2,...
house fin...	http:/...		http:/...	http:/...	Larry ... Haemor...	http:/...		[object Blob]		Jan 2,...	
indigo bu...	http:/...		http:/...	http:/...	Larry ... Passer...	http:/...	http:/...	[object Blob]		[objec...	Jan 2,...
savannah ...	http:/...		http:/...	http:/...	Larry ... Passer...	http:/...		[object Blob]		Jan 2,...	
american ...	http:/...		http:/...	http:/...	Larry ... Spizel...	http:/...	http:/...	[object Blob]		[objec...	Jan 2,...
dickcissel	http:/...					Spiza ...	http:/...	http:/...	[object Blob]	[objec...	
harris's ...	http:/...					Zonotr...	http:/...		[object Blob]		

Likewise, the third-from-last column, `audio_sample`, contains audio, and can be decoded using the same menu.

Browse... birds.dta

common_na...	wikip...	photo	photo...	photo...	photo...	binom...	audio...	a...	audio_sample	audio...	photo...
sedge wren	http://...		http://...	http://...	Larry ... Cistot...	http://...		1:01		[objec...	Jan 2,...
downy woo...	http://...		http://...	http://...	Larry ... Picoid...	http://...		0:06		[objec...	Jan 2,...
red-taile...	http://...		http://...	http://...	Forest... Buteo ...	http://...		0:02		[objec...	Jan 3,...
common ph...	http://...		http://...	http://...	Dick D... Phasia...	http://...		0:02		[objec...	Jan 3,...
common gr...	http://...		http://...	http://...	Larry ... Quisca...	http://...		0:17		[objec...	Jan 2,...
red-winge...	http://...		http://...	http://...	Larry ... Agelai...	http://...		0:33		[objec...	Jan 2,...
common re...	http://...		http://...	http://...	Larry ... Acanth...	http://...		0:11		[objec...	Jan 2,...
marsh wren	http://...		http://...	http://...	Larry ... Cistot...	http://...		0:36		[objec...	Jan 2,...
american ...	http://...		http://...	http://...	Larry ... Turdus...	http://...		7:22		[objec...	Jan 2,...
house fin...	http://...		http://...	http://...	Larry ... Haemor...	http://...		0:59		[objec...	Jan 2,...
indigo bu...	http://...		http://...	http://...	Larry ... Passer...	http://...		0:39		[objec...	Jan 2,...
savannah ...	http://...		http://...	http://...	Larry ... Passer...	http://...		3:46		[objec...	Jan 2,...
american ...	http://...		http://...	http://...	Larry ... Spizel...	http://...		0:33		[objec...	Jan 2,...
dickcissel	http://...				Spiza ...	http://...		1:06		[objec...	
harris's ...	http://...				Zonotr...	http://...		0:25		[objec...	

When `audio_sample` is decoded as audio, audio controls appear.



Let's take this a step further. Here I again clicked on the “Browse” button and will open another dta file, `birds_formatted.dta`. This is the exact same dataset, except that some formatting information has been added.

Browse... birds_formatted.dta

Sedge Wren
Cistothorus platensis

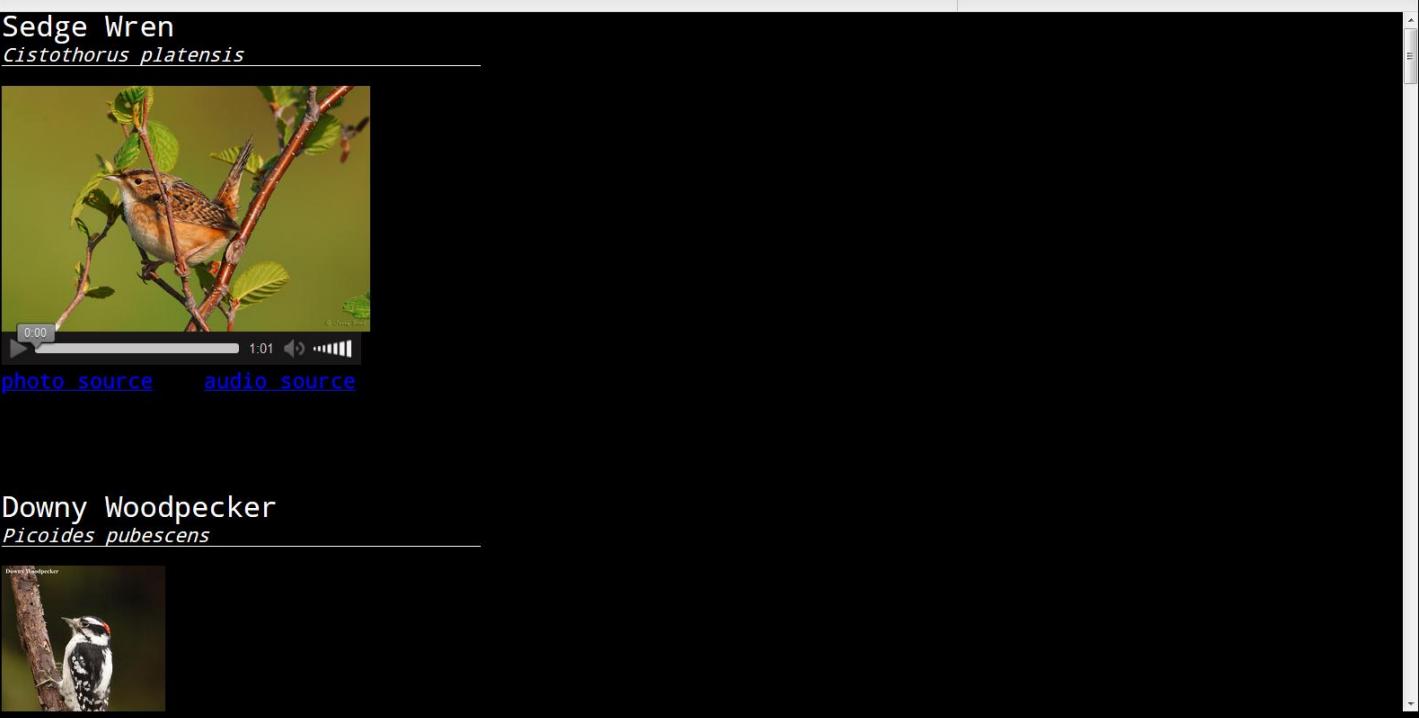


photo source audio source

Downy Woodpecker
Picoides pubescens

When `birds_formatted.dta` is opened, it looks like this. The big changes here are in the layout and in the fact that images and audio were decoded as such without the user having to inform the spreadsheet.

The formatting information has specified

1. that `photo` should be decoded as image and `audio_sample` as audio
2. layout information
3. text size and formatting
4. links be created from data URLs
5. which data appear in the display and which do not

This kind of functionality opens up the possibility of making automatic, multimedia slideshow presentations of the Stata dataset.

Multimedia spreadsheet

Made with

1. the `SlickGrid` JavaScript library
2. a new JavaScript library for opening `.dta` files

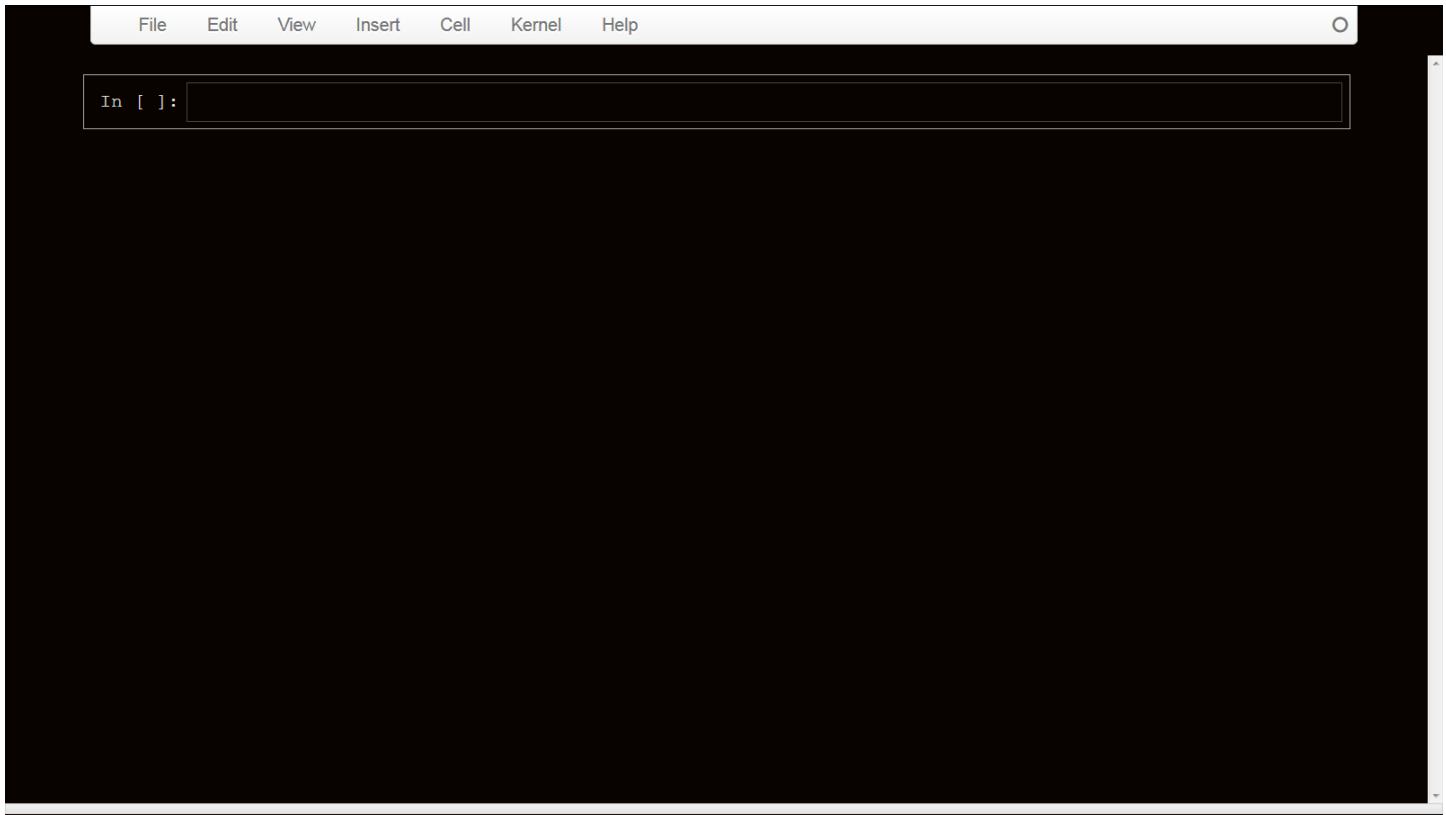
9

This example uses the SlickGrid library for its spreadsheet (the vast majority of the functionality comes from this library), and some custom JavaScript code for opening dta files (available on my GitHub page, see last slide for URL).

Notebook interface

10

Third and final idea: A new kind of interface for Stata. As a kind of preface, the benefit I see for this kind of interface is that notebook interfaces are a kind of editable history of a session. They are simultaneously a log file and a shareable presentation.



Here is the interface I will be using. Rather than try to define “notebook interface”, I will demonstrate some of the features of this particular notebook interface.

Input and output are organized in *cells*. Above you see the input half of a cell, waiting for the user’s input.

In [1]: `sysuse auto`

```
. sysuse auto  
(1978 Automobile Data)
```

In [2]: `describe`

```
. describe  
  
Contains data from C:\Program Files (x86)\Stata12\ado\base/a/auto.dta  
obs: 74 1978 Automobile Data  
vars: 12 13 Apr 2011 17:45  
size: 3,182 (_dta has notes)  
  
+-----  
variable name  storage  display    value  
      type   format     label    variable label  
+-----  
make          str18  %-18s  Make and Model  
price         int    %8.0gc  Price  
mpg          int    %8.0g  Mileage (mpg)  
rep78        int    %8.0g  Repair Record 1978  
headroom      float   %6.1f  Headroom (in.)  
trunk         int    %8.0g  Trunk space (cu. ft.)  
weight        int    %8.0gc  Weight (lbs.)  
length        int    %8.0g  Length (in.)  
turn          int    %8.0g  Turn Circle (ft.)  
displacement  int    %8.0g  Displacement (cu. in.)  
gear_ratio    float   %6.2f  Gear Ratio  
foreign       byte   %8.0g  origin  Car type  
+-----  
Sorted by: foreign
```

Here I've entered two inputs. The output appears immediately below. So far this is not much different from Stata's default interface.

The screenshot shows a Jupyter Notebook interface with a dark theme. At the top, there's a menu bar with File, Edit, View, Insert, Cell, Kernel, and Help. Below the menu, there's a command-line interface window showing Stata commands: foreign, byte, %8.0g, origin, and Car type, followed by a dashed line and the text "Sorted by: foreign". In the main notebook area, there are two code cells. The first cell, labeled "In [3]:", contains the command "help scatter". The second cell, labeled "Out[3]:", shows the output of the help command, which includes a title section and a syntax section. The title section starts with "[G-2] graph twoway scatter -- Twoway scatterplots". The syntax section starts with "[twoway] scatter varlist [if] [in] [weight] [, options]". There are also some ellipsis dots (...). A new code cell, "In []:", is visible at the bottom left.

```
foreign          byte    %8.0g      origin      Car type
-----
Sorted by: foreign

In [3]: help scatter
Out[3]: Title

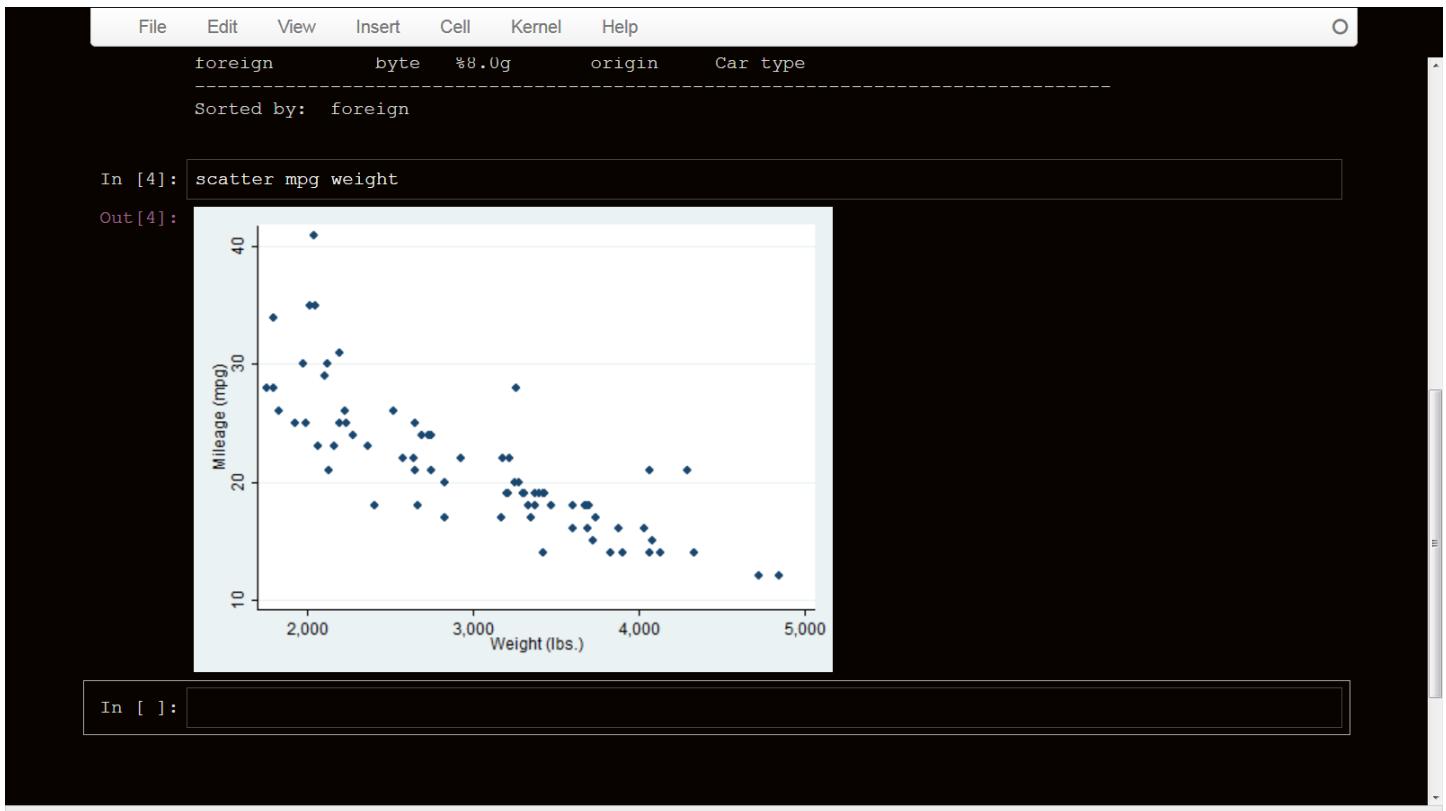
[G-2] graph twoway scatter -- Twoway scatterplots

Syntax

[twoway] scatter varlist [if] [in] [weight] [, options]

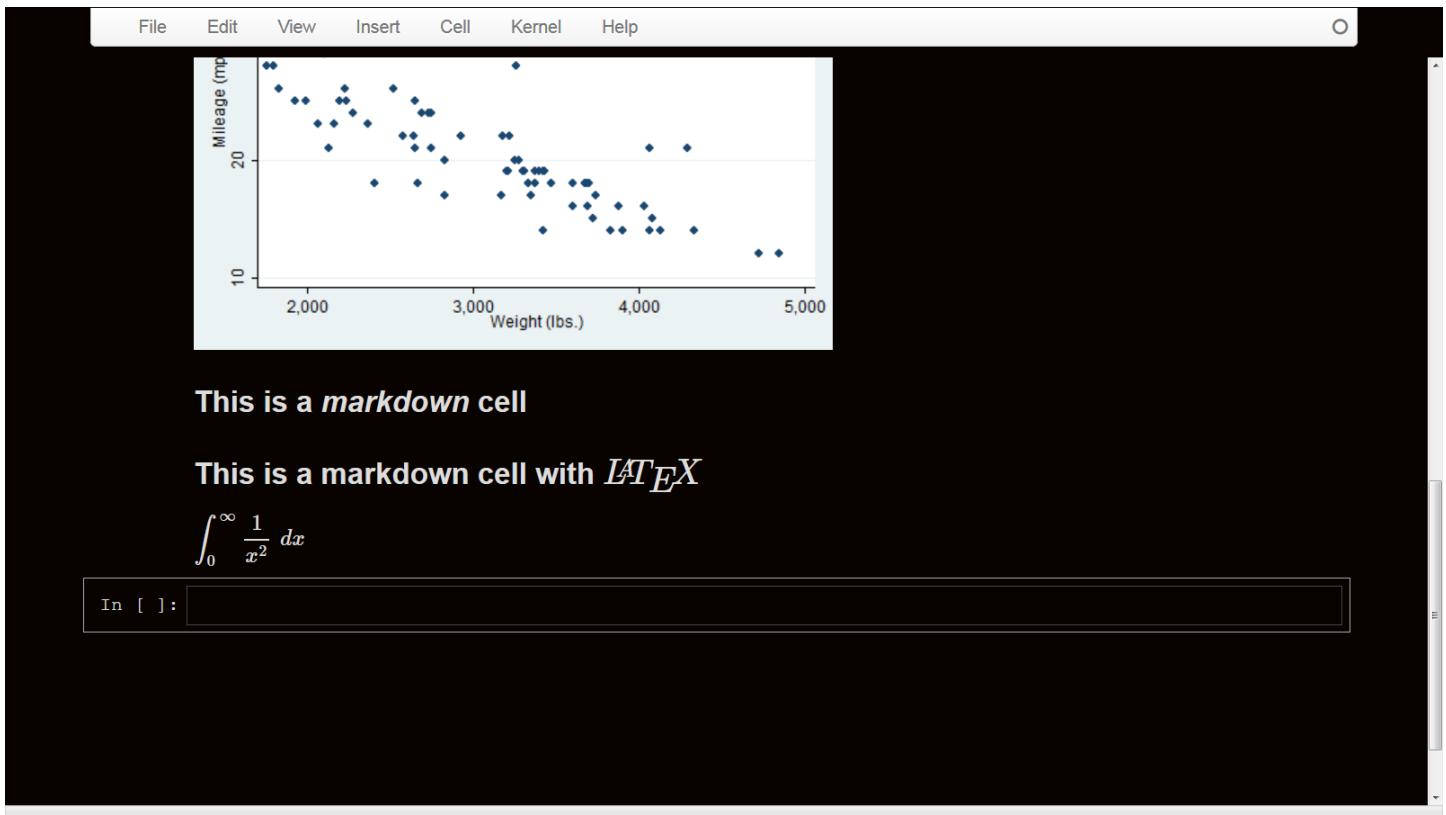
...
In [ ]:
```

Here we see the first difference. Typically, with notebook interfaces all of the output appears in the same window. In Stata's default interface, help files open in another window. Here the help file appears inline. In a sense, the notebook interfaces provides a more complete log of a session. In a Stata log you will see the command for `help scatter`, but the help file itself will not be there.



Here we see a few more differences. On the last page we opened a help file. If we plan on sharing this session later, we probably don't want to advertise that we forgot how to use the `scatter` command. Since the interface is an *editable* history, we just edit that out.

Here we write over that input with our new input, in this case `scatter mpg weight` (notice that the help file began right after the `describe` output, which is where our new `scatter` command is). The new output replaces the old output. Just like the help file, the graph shows up inline rather than in a new window. And again, by including the output of the `scatter` command, this interface is, in a sense, providing a more complete log of the session.



There are many other features that help this notebook interface be an editable, shareable history of your session. Cells can be rearranged or removed (in addition to being written over). You can insert markdown cells for notes or explanations, and you can include LaTeX.

Notebook interface

Made with

1. the IPython notebook
2. a modified version of log2html
by Kit Baum, Nick Cox, and Bill Rising

11

The majority of the functionality shown here comes from the IPython notebook.
The modified version of log2html was used for converting the help file to html.

Resources

Python plugin ssc describe python

Sympy sympy.org/en/index.html

SlickGrid github.com/mleibman/SlickGrid

IPython notebook ipython.org/notebook.html

Example code github.com/jrfiedler/StataCon2014

Contact jrfiedler@gmail.com

12