Imagining a Stata/Python combination

James Fiedler
Stata Conference, 2012

Abstract:

There are occasions when a task is difficult in Stata but fairly easy in a more general programming language. Python is a popular language for a range of uses. It is easy to use, has many high-quality packages, and programs can be written relatively quickly. Is there any advantage to combining Stata and Python within a single interface? Stata already offers support for user-written programs, which allow extensive control over calculations but somewhat less control over graphics. Also, except for specifying output, the user has minimal programmatic control over the user interface. Python can be used in ways that allow more control over the interface and graphics, and in so doing provide roundabout methods for satisfying some user requests (for example, transparency levels in graphics and the ability to clear the results window). My talk will explore these ideas, present a possible method for combining Stata and Python, and give examples to demonstrate how this combination might be useful.

To define the scope a little more, I'm considering why and how users might want to combine Stata and Python, rather than how and why Stata Corp. might.

I'll be presenting some of my own ideas, but to help direct the brainstorming, I'll also be using requests I found on Statalist.

support

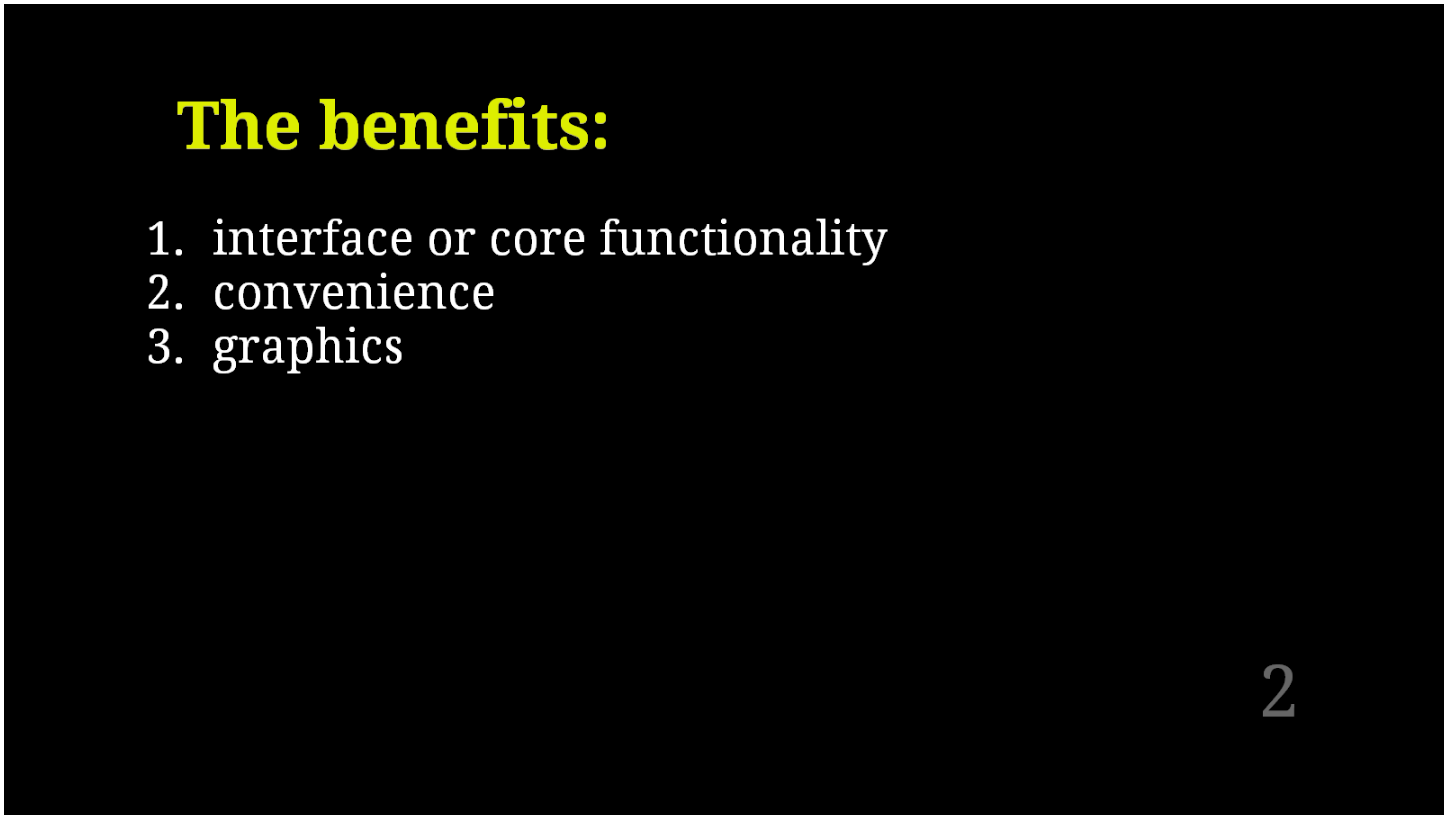
where I work:

NASA Johnson Space Center

who I work for:

Universities Space Research Association

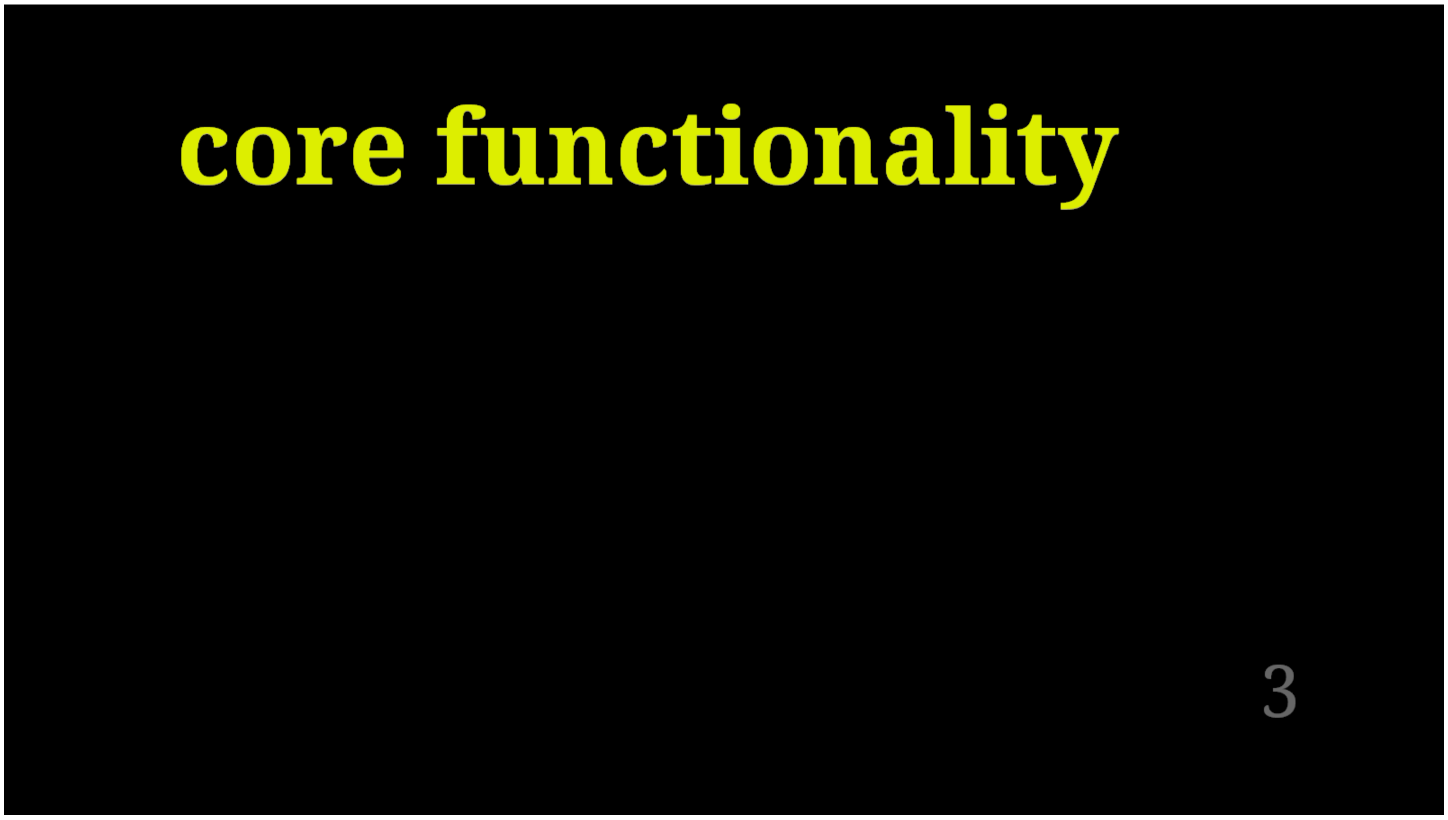
I’ll be discussing three kinds of benefits.



The benefits:

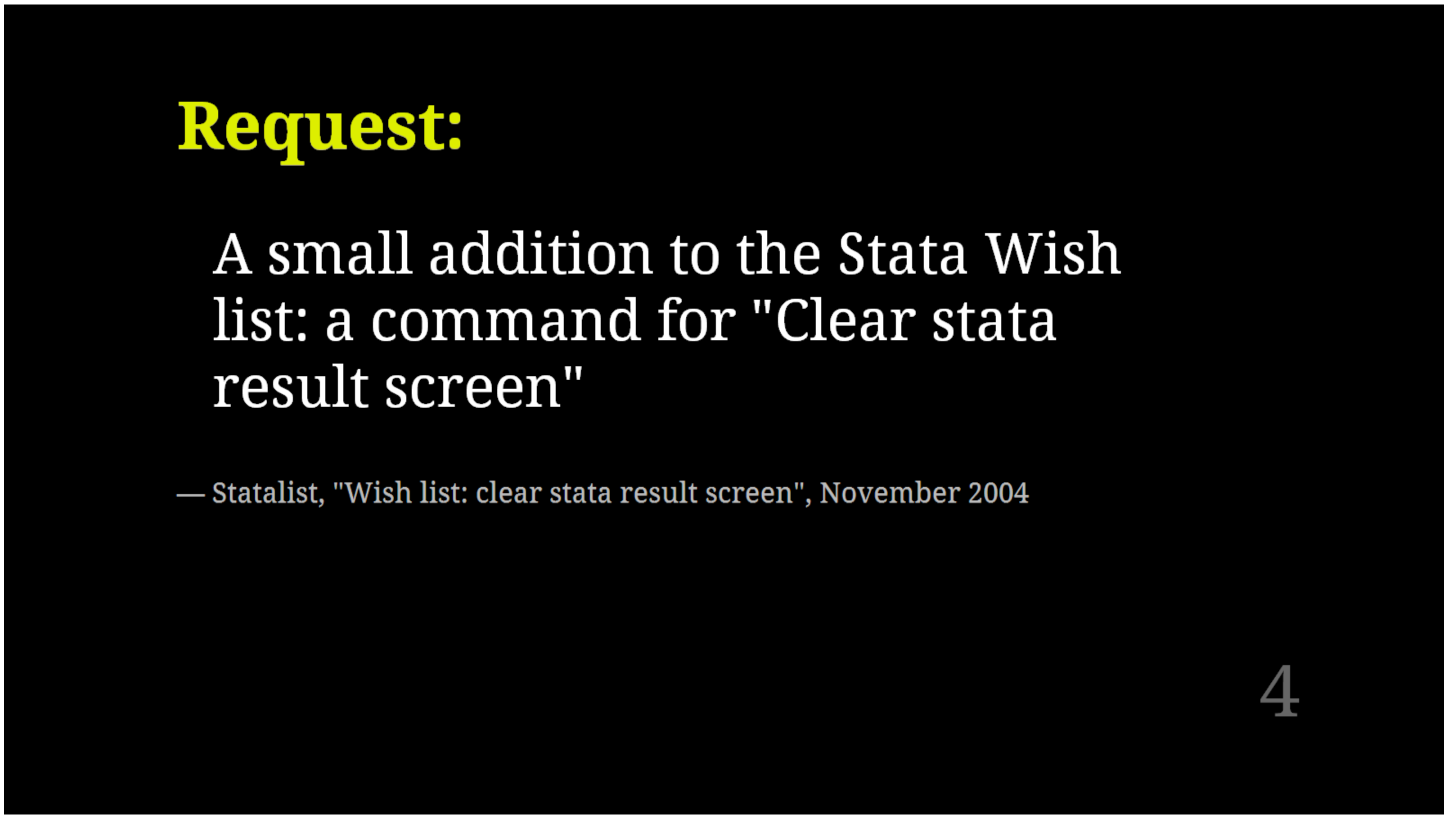
1. interface or core functionality
2. convenience
3. graphics

2



core functionality

3

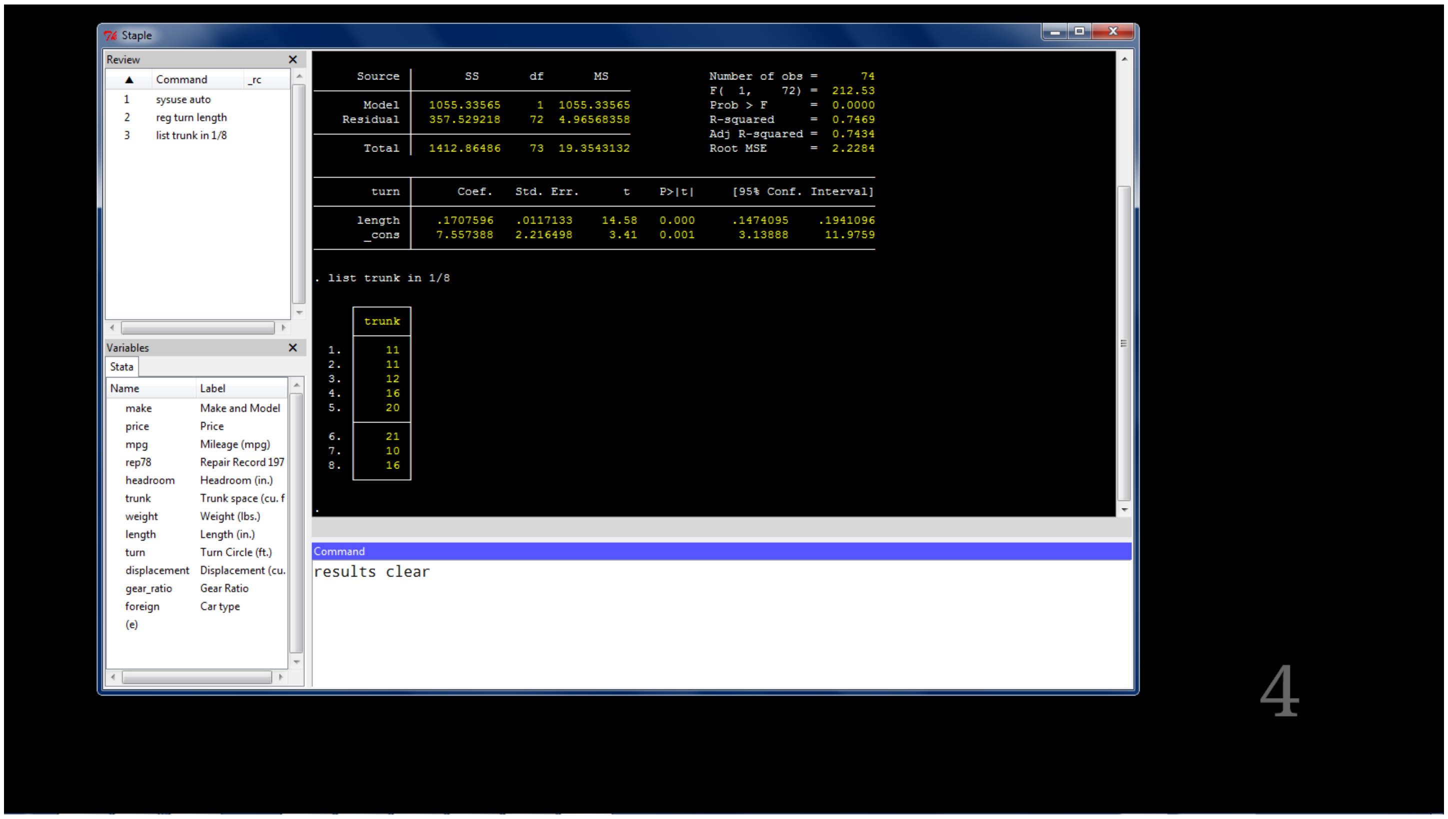


Request:

A small addition to the Stata Wish list: a command for "Clear stata result screen"

— Statalist, "Wish list: clear stata result screen", November 2004

4



Command	_rc
1 sysuse auto	
2 reg turn length	
3 list trunk in 1/8	

Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type
(e)	

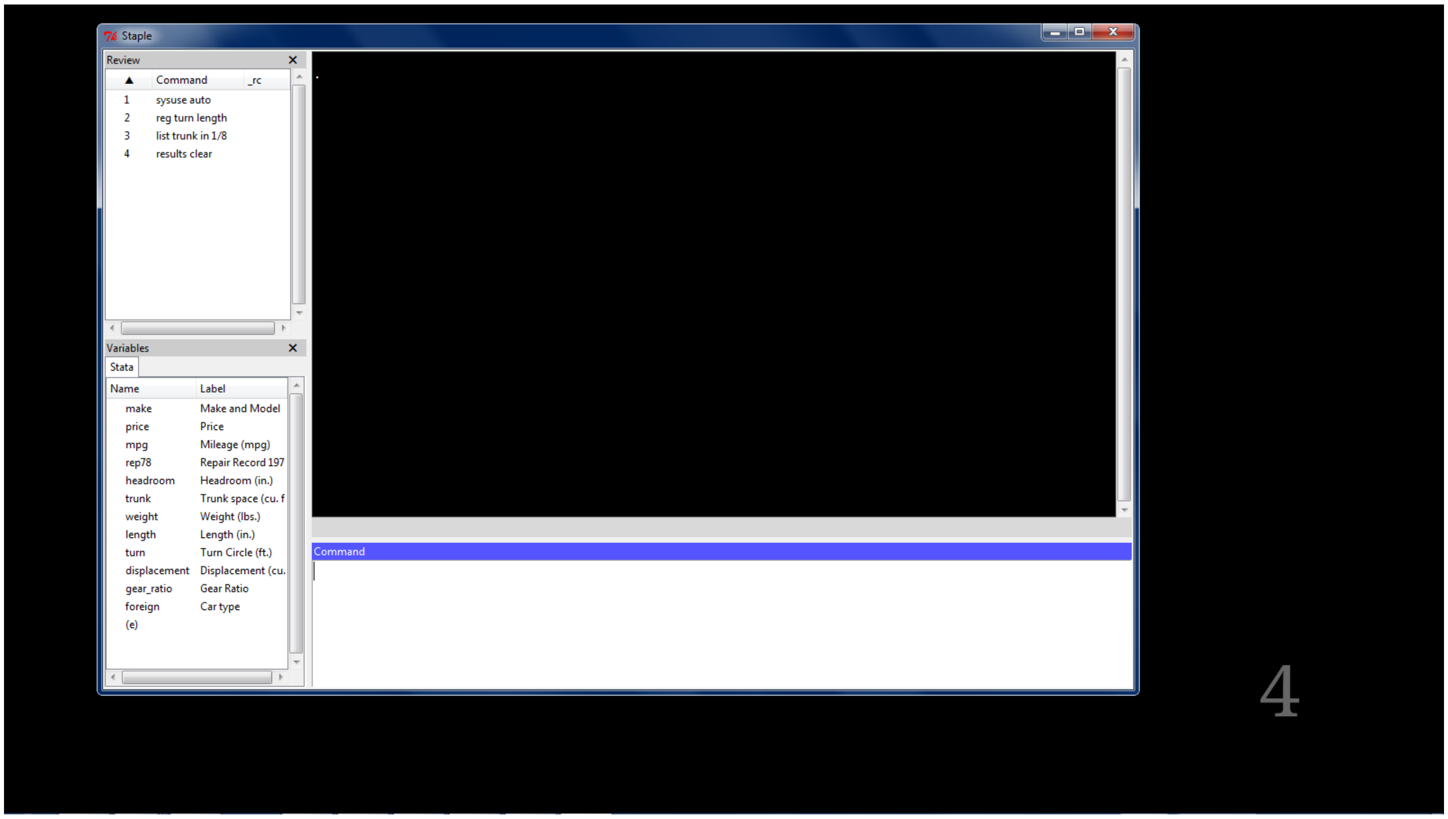
```
      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  1,    72) =  212.53
       Model |  1055.33565     1  1055.33565           Prob > F      =  0.0000
    Residual |  357.529218    72  4.96568358           R-squared     =  0.7469
-------------+------------------------------           Adj R-squared =  0.7434
       Total |  1412.86486    73  19.3543132           Root MSE      =  2.2284

------------------------------------------------------------------------------
        turn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      length |   .1707596   .0117133    14.58   0.000     .1474095    .1941096
       _cons |   7.557388   2.216498     3.41   0.001      3.13888     11.9759
------------------------------------------------------------------------------

. list trunk in 1/8

     +-------+
     | trunk |
     |-------|
  1. |    11 |
  2. |    11 |
  3. |    12 |
  4. |    16 |
  5. |    20 |
     |-------|
  6. |    21 |
  7. |    10 |
  8. |    16 |
     +-------+

.
```

Command

```
results clear
```

4

This user wants the ability to type a command such as `results clear` (as I have done here), and ...

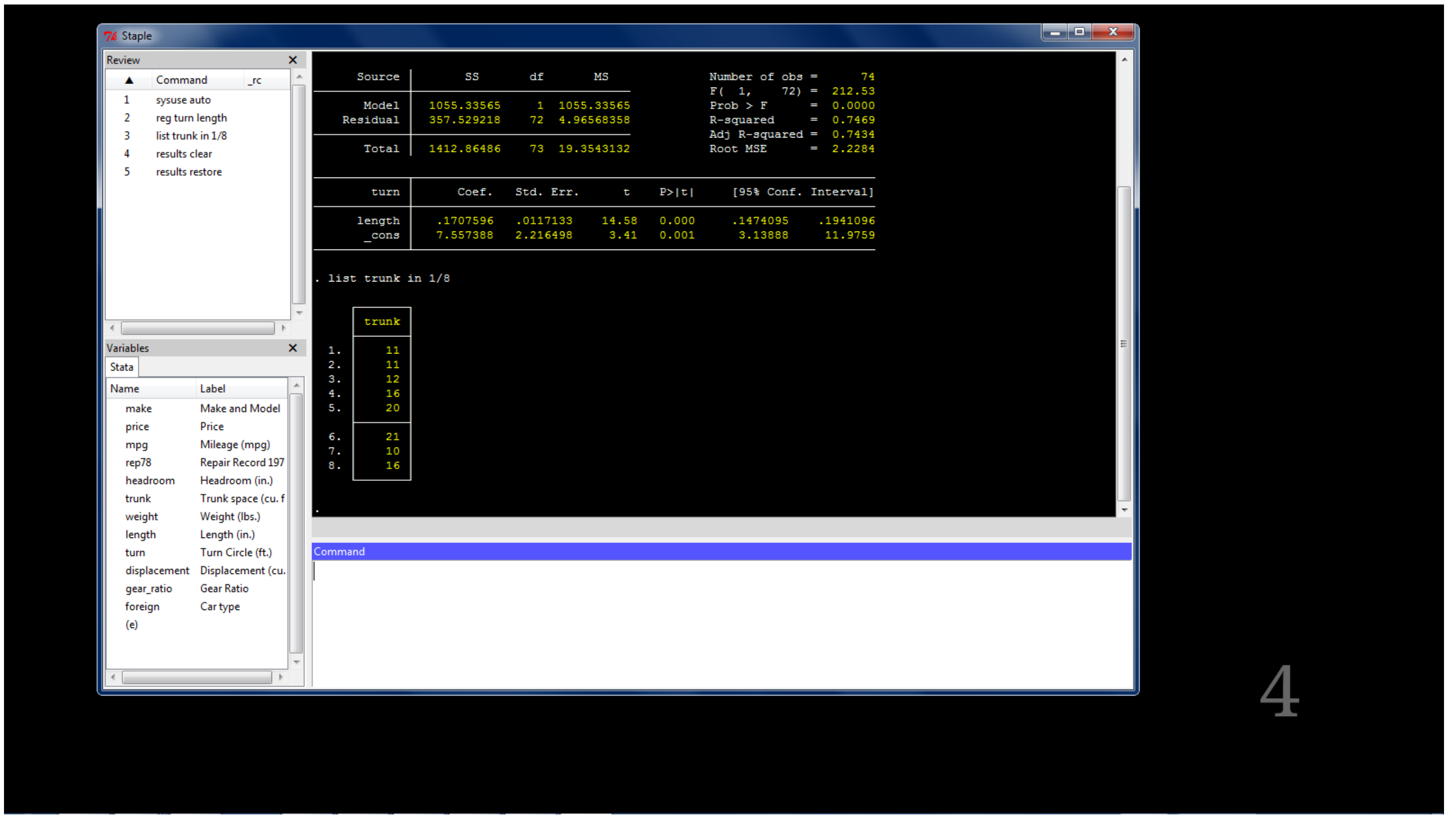


▲	Command	_rc
1	sysuse auto	
2	reg turn length	
3	list trunk in 1/8	
4	results clear	

Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type
(e)	

… when the command is submitted, the results window is cleared.

There are ways to imitate this. For instance, the user could insert blank lines until the old output is no longer visible. However, the user can’t remove the blank lines to make the old output reappear.

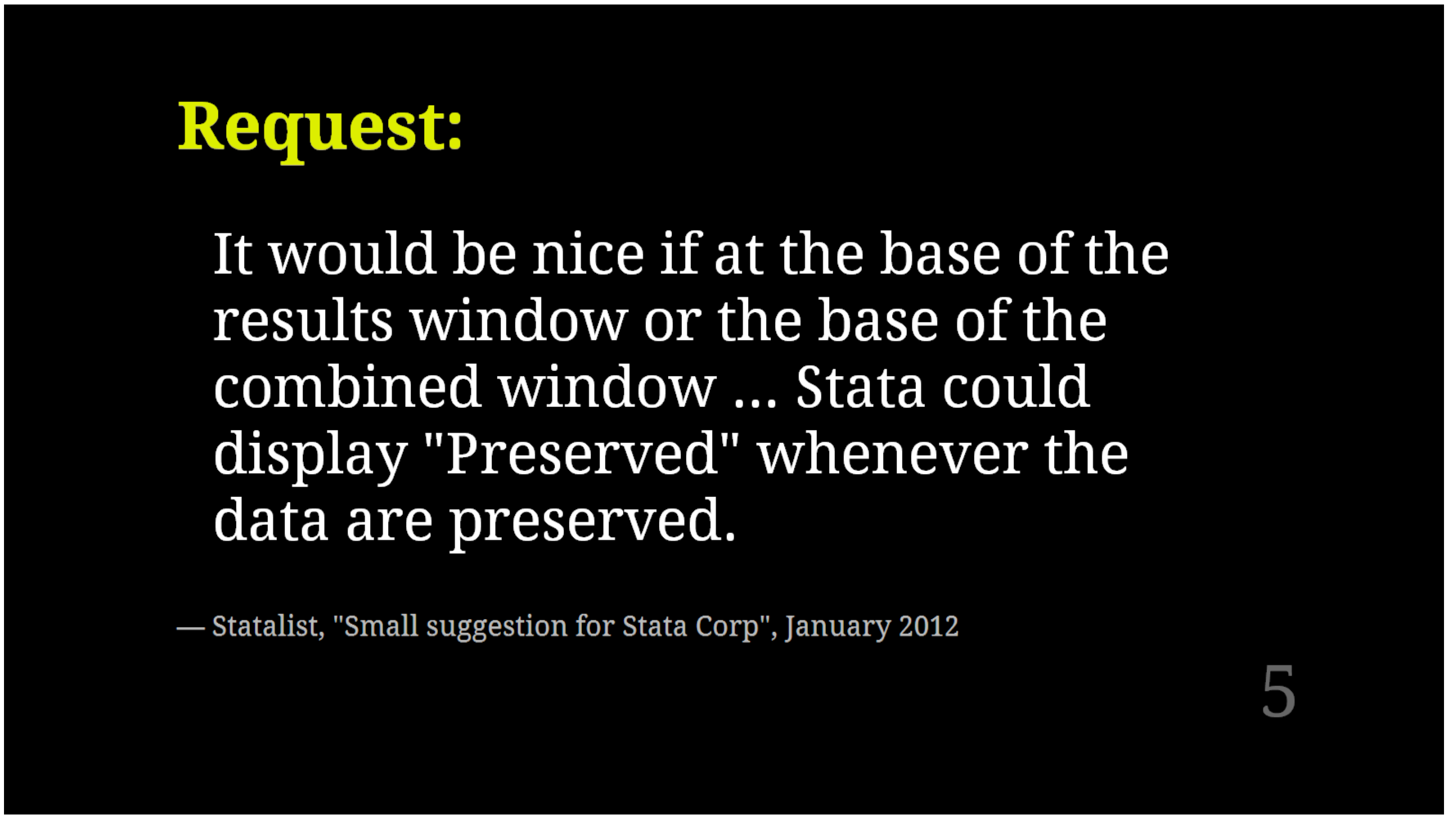


	Command	_rc
1	sysuse auto	
2	reg turn length	
3	list trunk in 1/8	
4	results clear	
5	results restore	

Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type
(e)	

```
      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  1,    72) =  212.53
       Model |  1055.33565     1  1055.33565           Prob > F      =  0.0000
    Residual |  357.529218    72  4.96568358           R-squared     =  0.7469
-------------+------------------------------           Adj R-squared =  0.7434
       Total |  1412.86486    73  19.3543132           Root MSE      =  2.2284

------------------------------------------------------------------------------
        turn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      length |   .1707596   .0117133    14.58   0.000     .1474095    .1941096
       _cons |   7.557388   2.216498     3.41   0.001      3.13888     11.9759
------------------------------------------------------------------------------

. list trunk in 1/8

     +-------+
     | trunk |
     |-------|
  1. |    11 |
  2. |    11 |
  3. |    12 |
  4. |    16 |
  5. |    20 |
     |-------|
  6. |    21 |
  7. |    10 |
  8. |    16 |
     +-------+

.
```

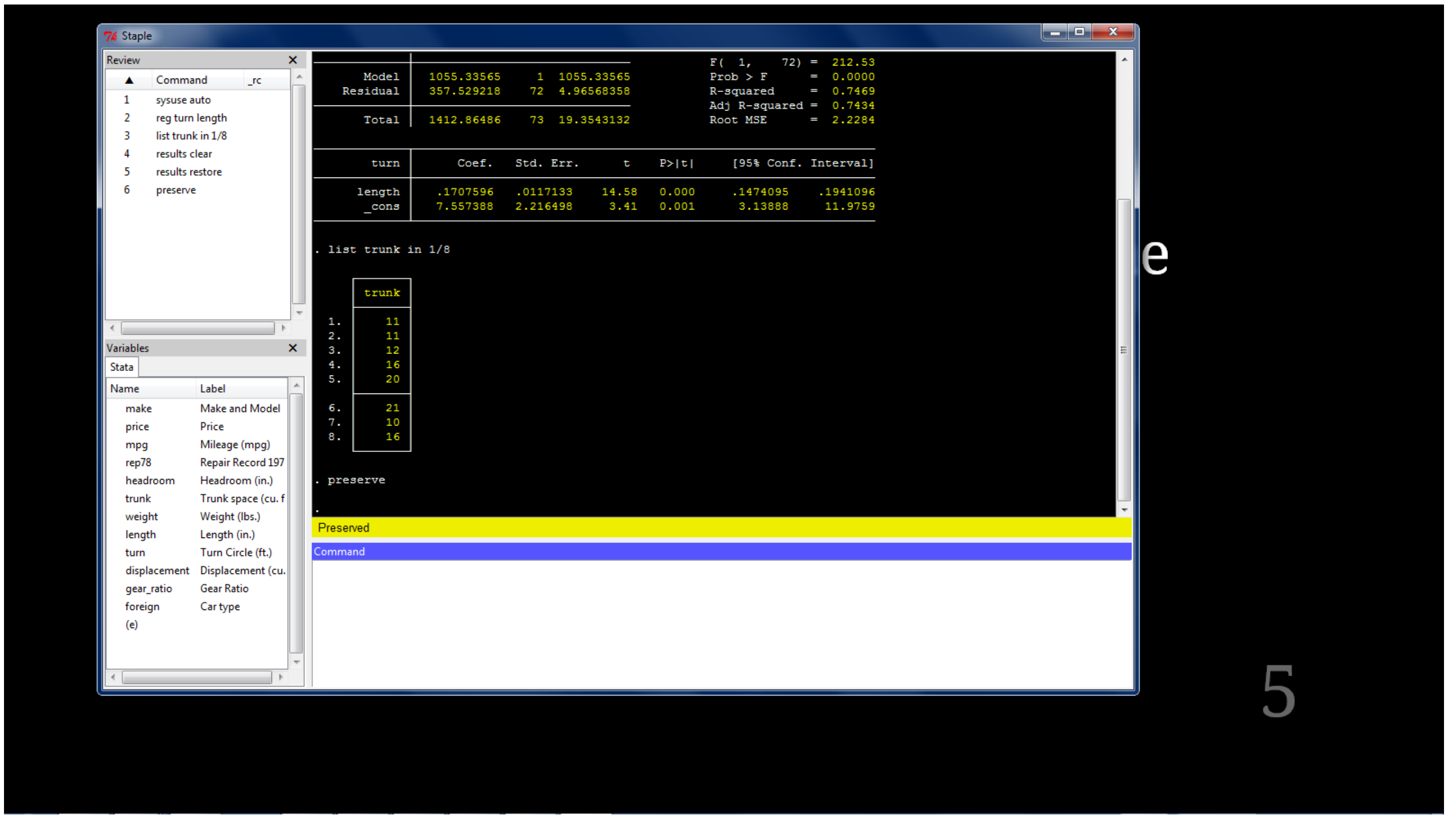
Sometimes it will be useful to have a command like `results restore` to bring the results back (as has been done here).


Request:

It would be nice if at the base of the results window or the base of the combined window ... Stata could display "Preserved" whenever the data are preserved.

— Statalist, "Small suggestion for Stata Corp", January 2012

Another request, this time for a visual reminder that a dataset has been preserved.



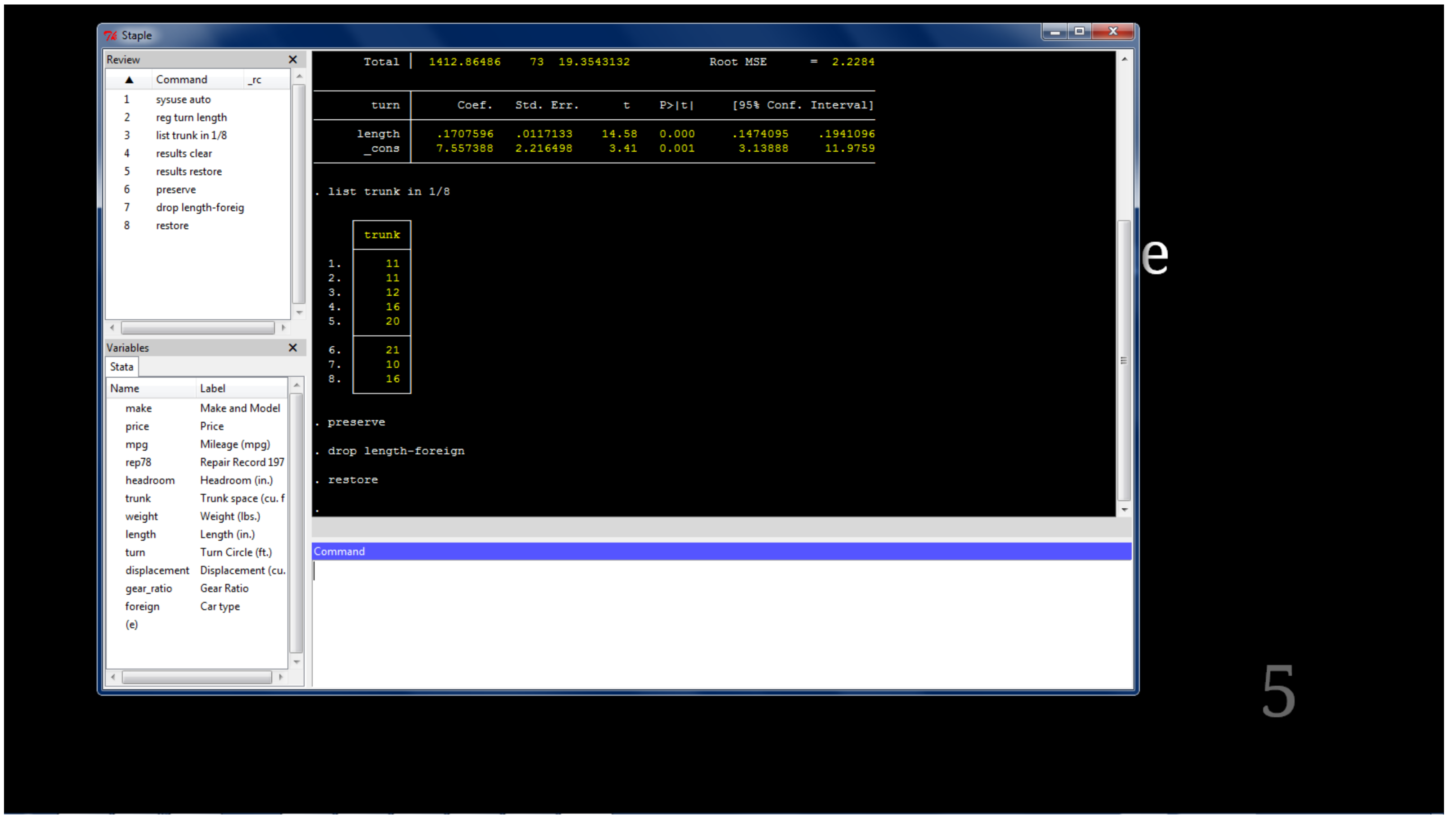
	Command
1	sysuse auto
2	reg turn length
3	list trunk in 1/8
4	results clear
5	results restore
6	preserve

Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type
(e)	

```
       Model |  1055.33565     1  1055.33565           F(  1,    72) =  212.53
    Residual |  357.529218    72  4.96568358           Prob > F      =  0.0000
                                                       R-squared     =  0.7469
                                                       Adj R-squared =  0.7434
       Total |  1412.86486    73  19.3543132           Root MSE      =  2.2284

        turn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]

      length |   .1707596   .0117133    14.58   0.000     .1474095    .1941096
       _cons |   7.557388   2.216498     3.41   0.001      3.13888     11.9759

. list trunk in 1/8

     | trunk |
     |-------|
  1. |    11 |
  2. |    11 |
  3. |    12 |
  4. |    16 |
  5. |    20 |
     |-------|
  6. |    21 |
  7. |    10 |
  8. |    16 |

. preserve

.
```

Preserved

Command



	Command
1	sysuse auto
2	reg turn length
3	list trunk in 1/8
4	results clear
5	results restore
6	preserve
7	drop length-foreig
8	restore

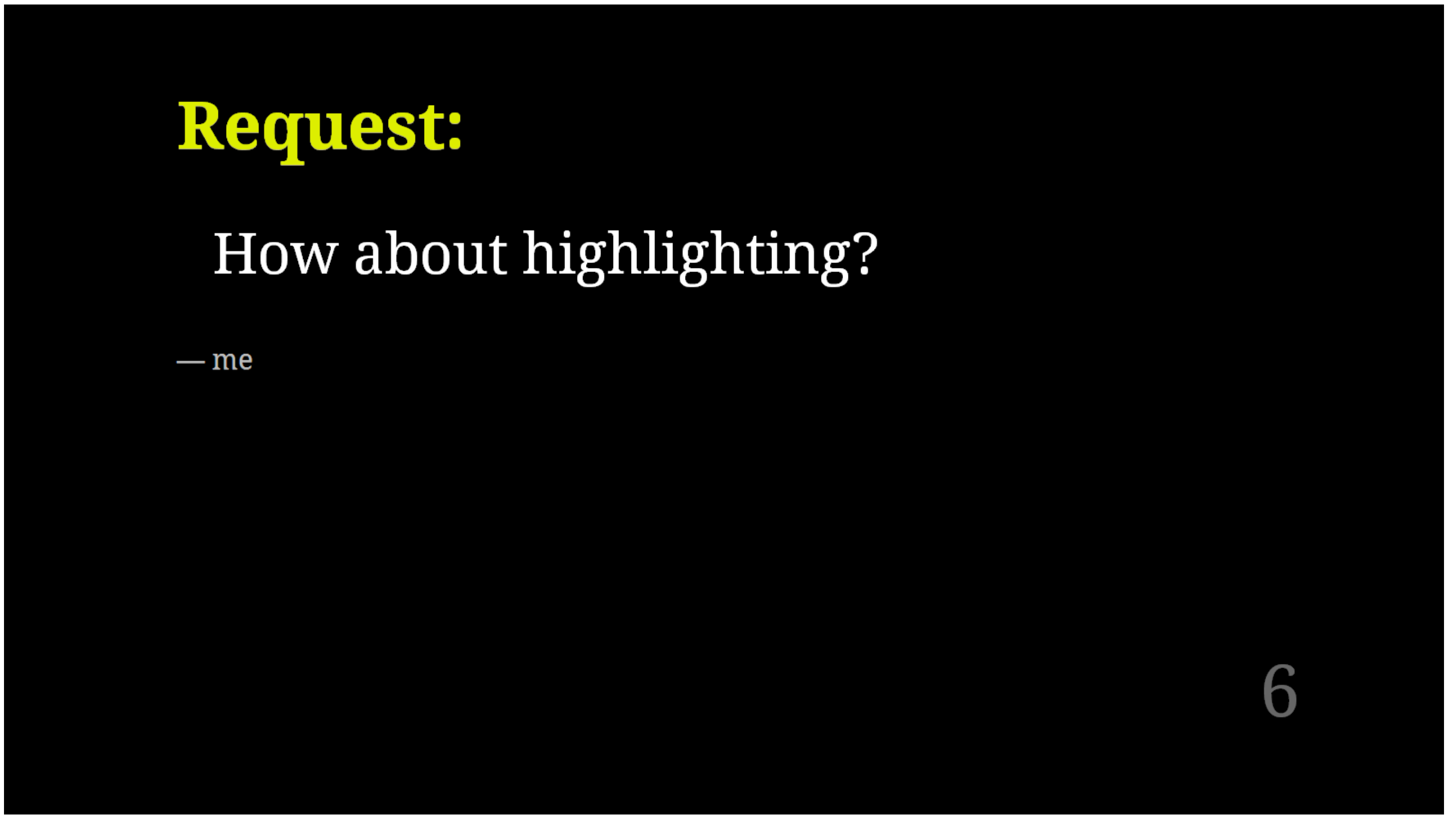
Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type
(e)	

```
       Total |  1412.86486    73  19.3543132           Root MSE      =  2.2284

        turn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]

      length |   .1707596   .0117133    14.58   0.000     .1474095    .1941096
       _cons |   7.557388   2.216498     3.41   0.001      3.13888     11.9759

. list trunk in 1/8

     | trunk |
     |-------|
  1. |    11 |
  2. |    11 |
  3. |    12 |
  4. |    16 |
  5. |    20 |
     |-------|
  6. |    21 |
  7. |    10 |
  8. |    16 |

. preserve

. drop length-foreign

. restore

.
```

Command

In the top slide, the status bar below the results window turns yellow and the status is set to "Preserved" after submitting a `preserve` command.

In the bottom slide, the status bar returns to its default state after submitting a `restore` command.

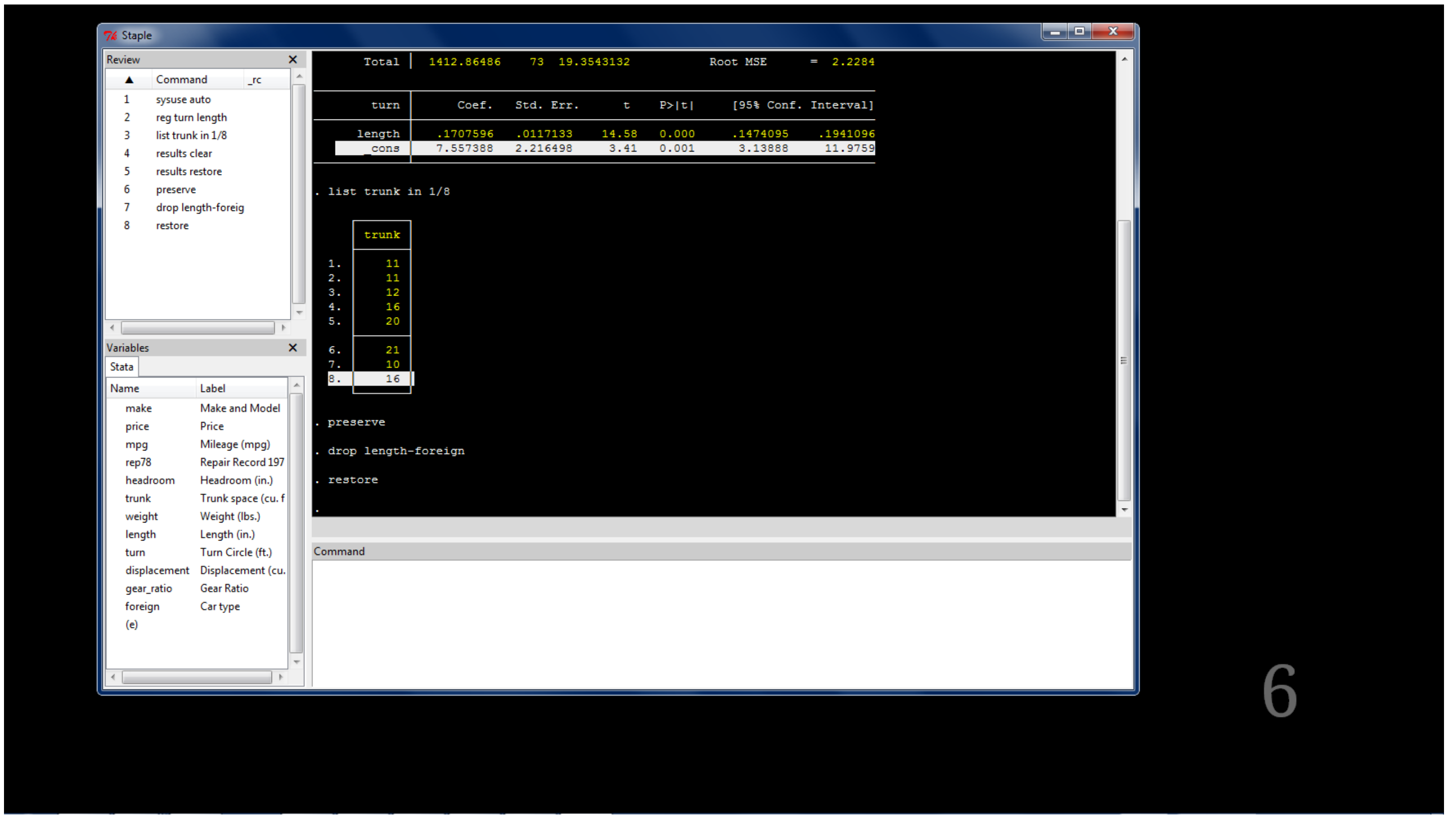


Request:

How about highlighting?

— me

6



Staple

Review

▲	Command	_rc
1	sysuse auto	
2	reg turn length	
3	list trunk in 1/8	
4	results clear	
5	results restore	
6	preserve	
7	drop length-foreig	
8	restore	

Variables

Stata

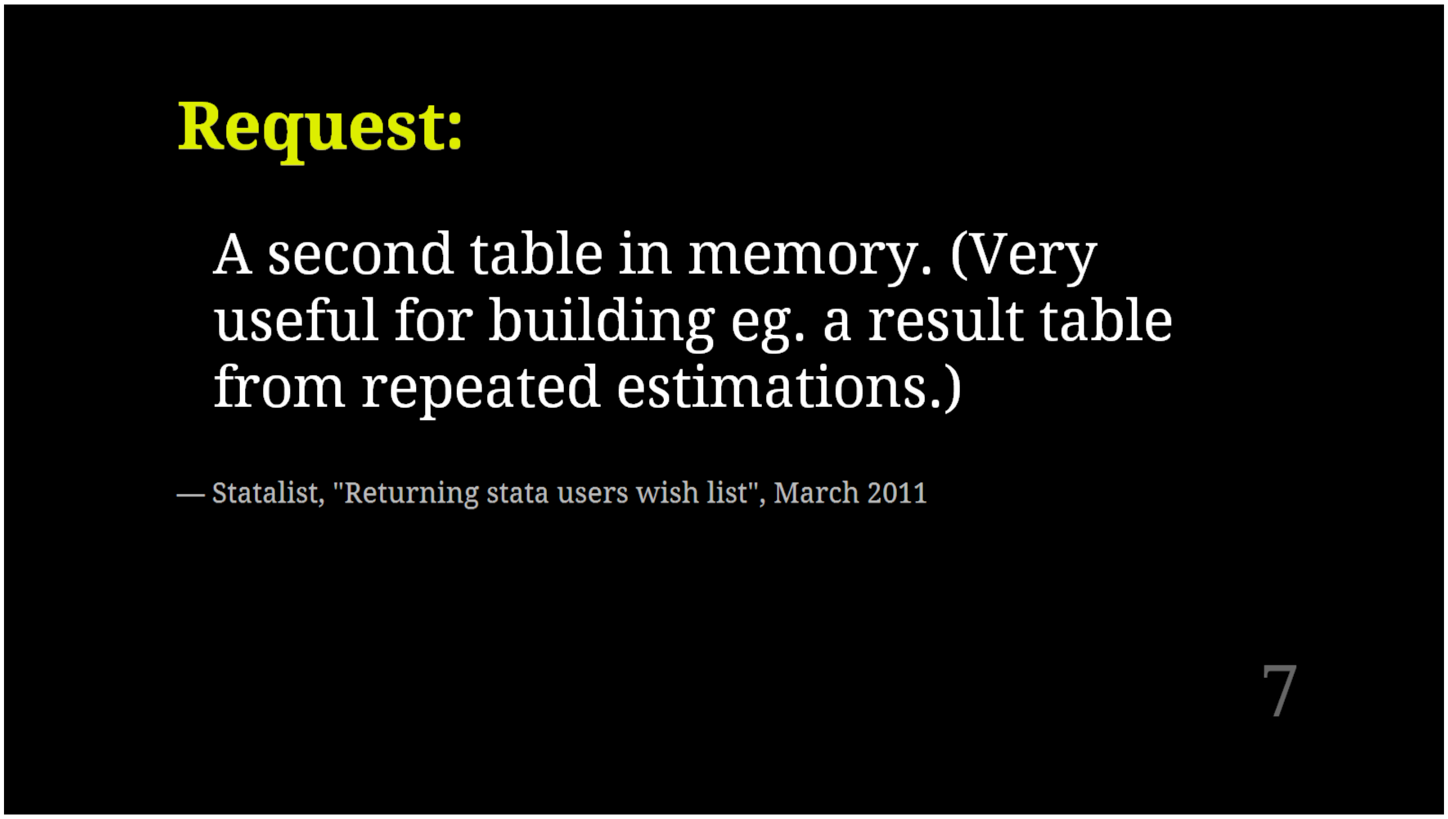
Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type
(e)	

```
       Total |  1412.86486    73  19.3543132           Root MSE      =  2.2284

------------------------------------------------------------------------------
        turn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      length |   .1707596   .0117133    14.58   0.000     .1474095    .1941096
       _cons |   7.557388   2.216498     3.41   0.001      3.13888     11.9759
------------------------------------------------------------------------------

. list trunk in 1/8

     +-------+
     | trunk |
     |-------|
  1. |    11 |
  2. |    11 |
  3. |    12 |
  4. |    16 |
  5. |    20 |
     |-------|
  6. |    21 |
  7. |    10 |
  8. |    16 |
     +-------+

. preserve

. drop length-foreign

. restore

.
```

Command

6

It would also be nice to have interactive highlighting for making some parts of the output more salient. This could be used as a way to bookmark results in a long Stata session or to help draw attention to particular areas when the results are given to someone else. Here, two portions of the result have been highlighted (black text on white background).



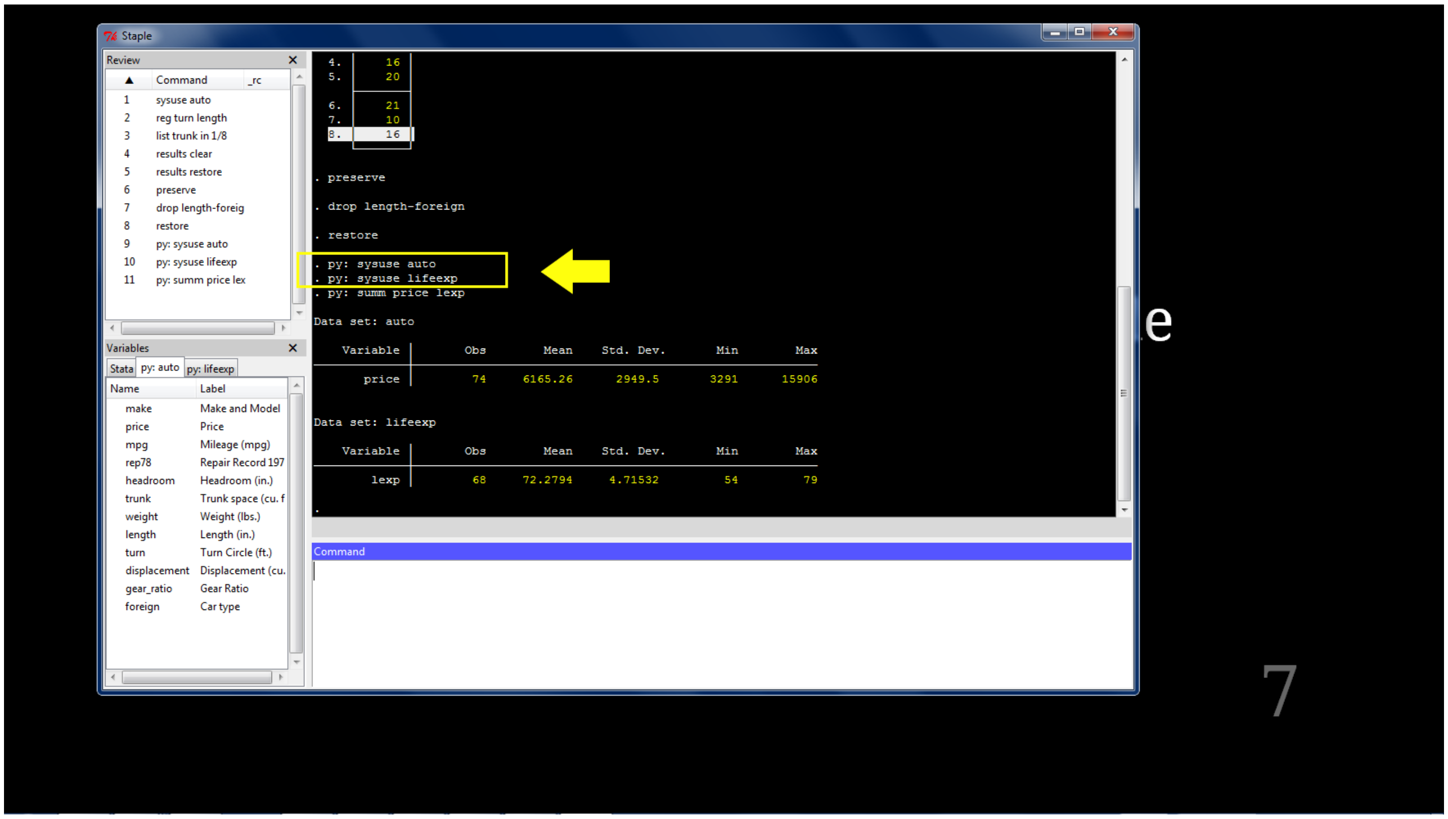
Request:

A second table in memory. (Very useful for building eg. a result table from repeated estimations.)

— Statalist, "Returning stata users wish list", March 2011

7

Stata can hold one data set at a time in memory. If there were a way to load Stata datasets into Python, the user could have simultaneous access to as many datasets as allowed by the memory constraints of the computer.



▲	Command	_rc
1	sysuse auto	
2	reg turn length	
3	list trunk in 1/8	
4	results clear	
5	results restore	
6	preserve	
7	drop length-foreig	
8	restore	
9	py: sysuse auto	
10	py: sysuse lifeexp	
11	py: summ price lex	

Stata | py: auto | py: lifeexp

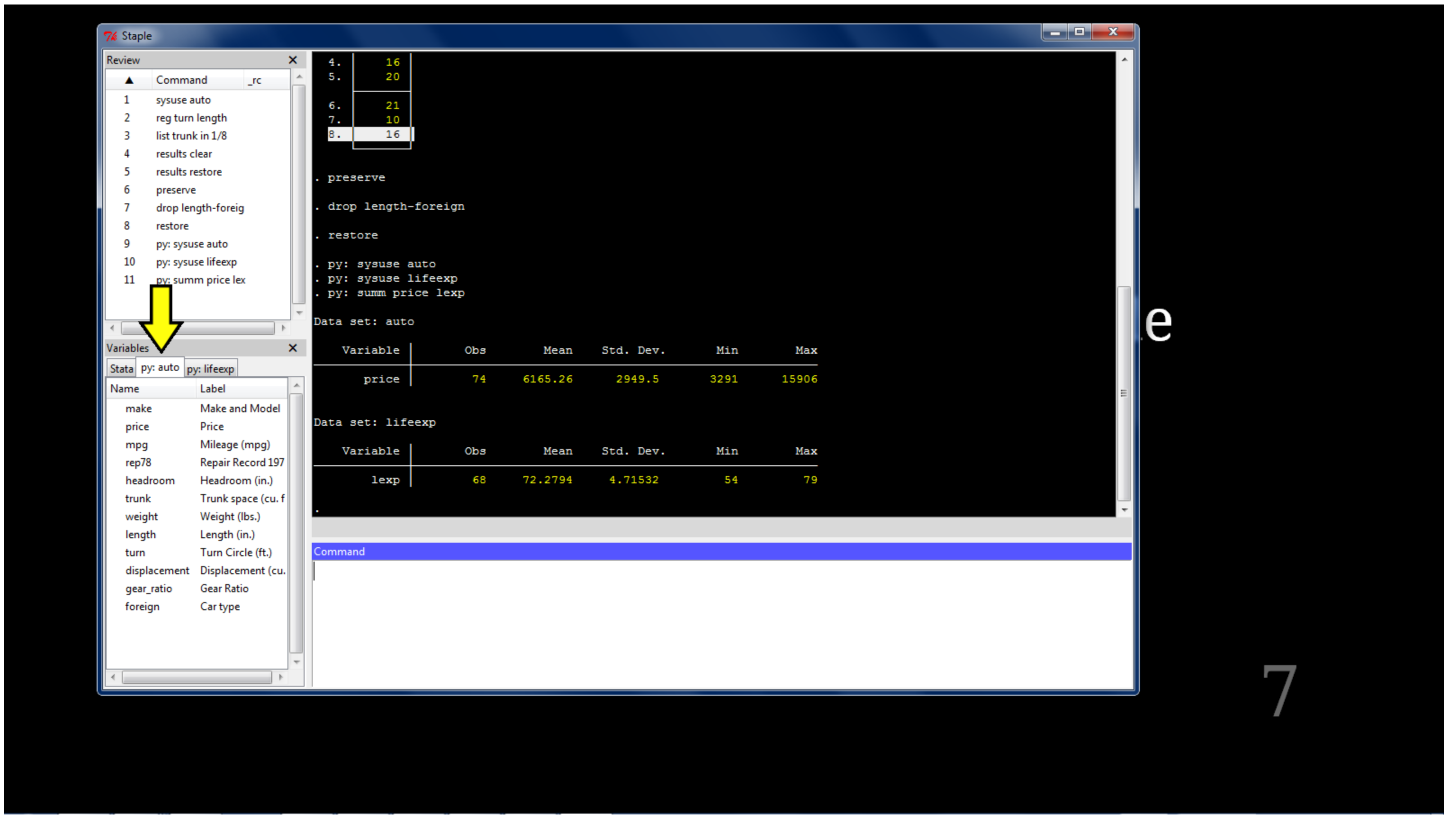
Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type

```
  4.      16
  5.      20

  6.      21
  7.      10
  8.      16


. preserve

. drop length-foreign

. restore

. py: sysuse auto
. py: sysuse lifeexp
. py: summ price lexp

Data set: auto

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       price |        74     6165.26      2949.5       3291      15906


Data set: lifeexp

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        lexp |        68     72.2794     4.71532         54         79

.
```

Here I've entered two commands to load Stata datasets into Python:

```
. py: sysuse auto
. py: sysuse lifeexp
```

Here, `py: sysuse` can be interpreted as “load system dataset into Python”.


Staple

Review

▲	Command	_rc
1	sysuse auto	
2	reg turn length	
3	list trunk in 1/8	
4	results clear	
5	results restore	
6	preserve	
7	drop length-foreig	
8	restore	
9	py: sysuse auto	
10	py: sysuse lifeexp	
11	py: summ price lex	

Variables

Stata | py: auto | py: lifeexp

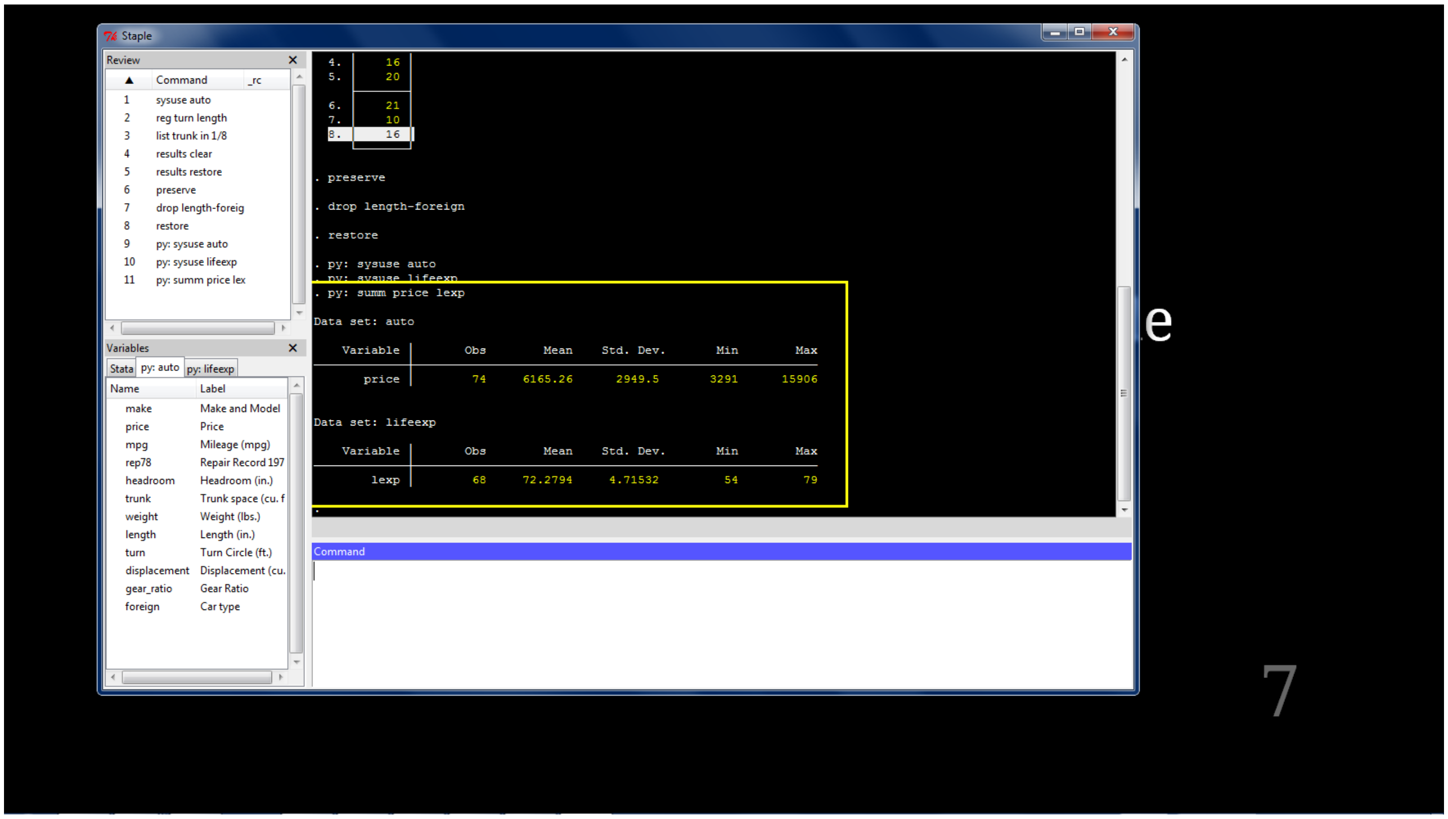
Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type

```
  4.      16
  5.      20

  6.      21
  7.      10
  8.      16


. preserve

. drop length-foreign

. restore

. py: sysuse auto
. py: sysuse lifeexp
. py: summ price lexp

Data set: auto

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       price |         74     6165.26      2949.5       3291      15906


Data set: lifeexp

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        lexp |         68     72.2794     4.71532         54         79

.
```

Command

e

The previous commands also prompted the creation of two new tabs in the variables pane, one for each of the new datasets (there’s also one for the already-open dataset in Stata).


Staple

Review

▲	Command	_rc
1	sysuse auto	
2	reg turn length	
3	list trunk in 1/8	
4	results clear	
5	results restore	
6	preserve	
7	drop length-foreig	
8	restore	
9	py: sysuse auto	
10	py: sysuse lifeexp	
11	py: summ price lex	

Variables

Stata | py: auto | py: lifeexp

Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 197
headroom	Headroom (in.)
trunk	Trunk space (cu. f
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu.
gear_ratio	Gear Ratio
foreign	Car type

```
  4.  |     16 |
  5.  |     20 |
      |--------|
  6.  |     21 |
  7.  |     10 |
  8.  |     16 |
      +--------+

. preserve

. drop length-foreign

. restore

. py: sysuse auto
. py: sysuse lifeexp
. py: summ price lexp

Data set: auto

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       price |        74     6165.26      2949.5       3291      15906


Data set: lifeexp

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        lexp |        68     72.2794     4.71532         54         79

.
```

Command

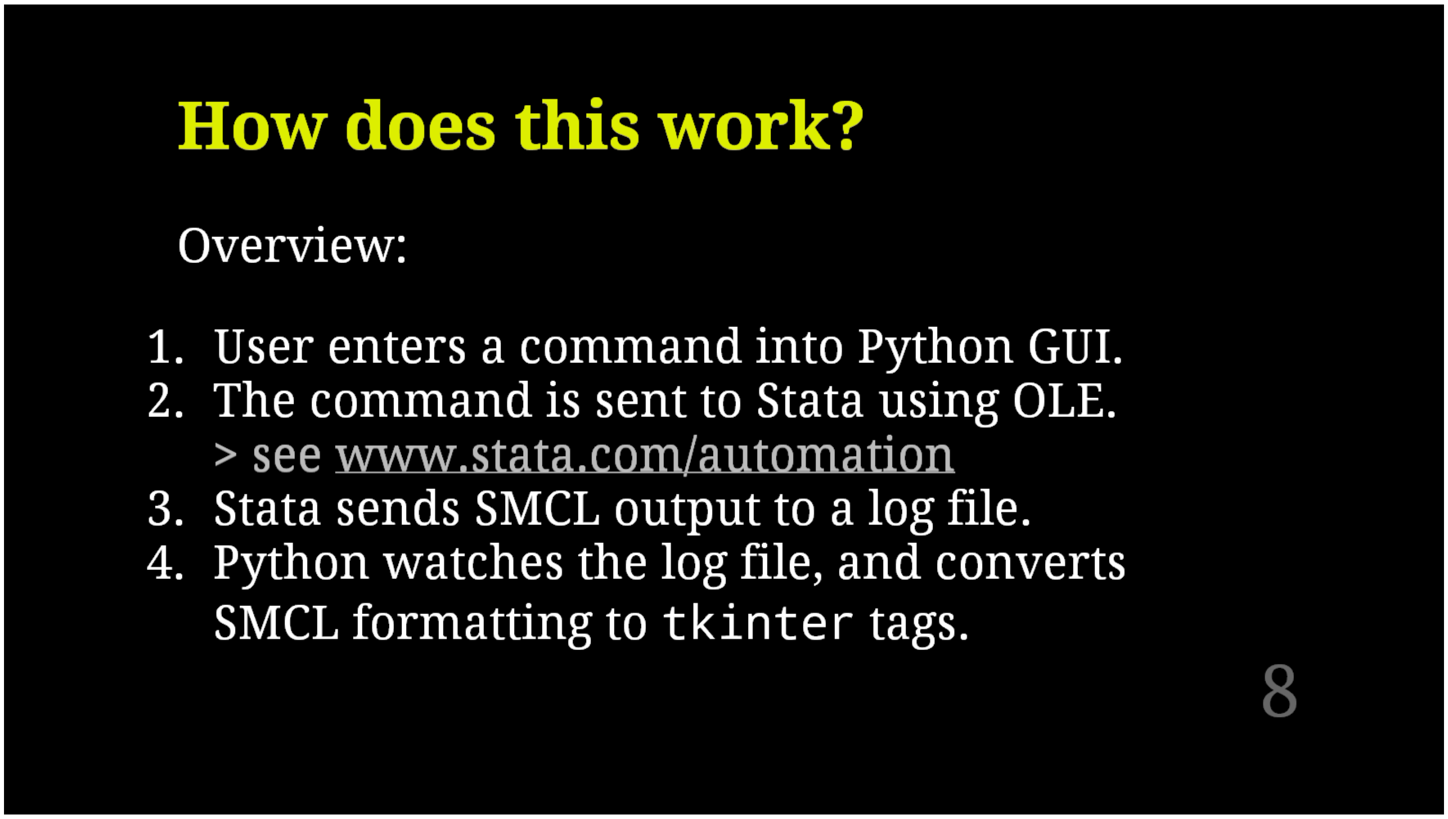
e

7

To prove that these two datasets are in Python simultaneously, I enter the command

```
      . py: summ price lexp
```

The program finds `price` within the auto dataset and `lexp` within the life expectancy dataset, and gives their summaries.



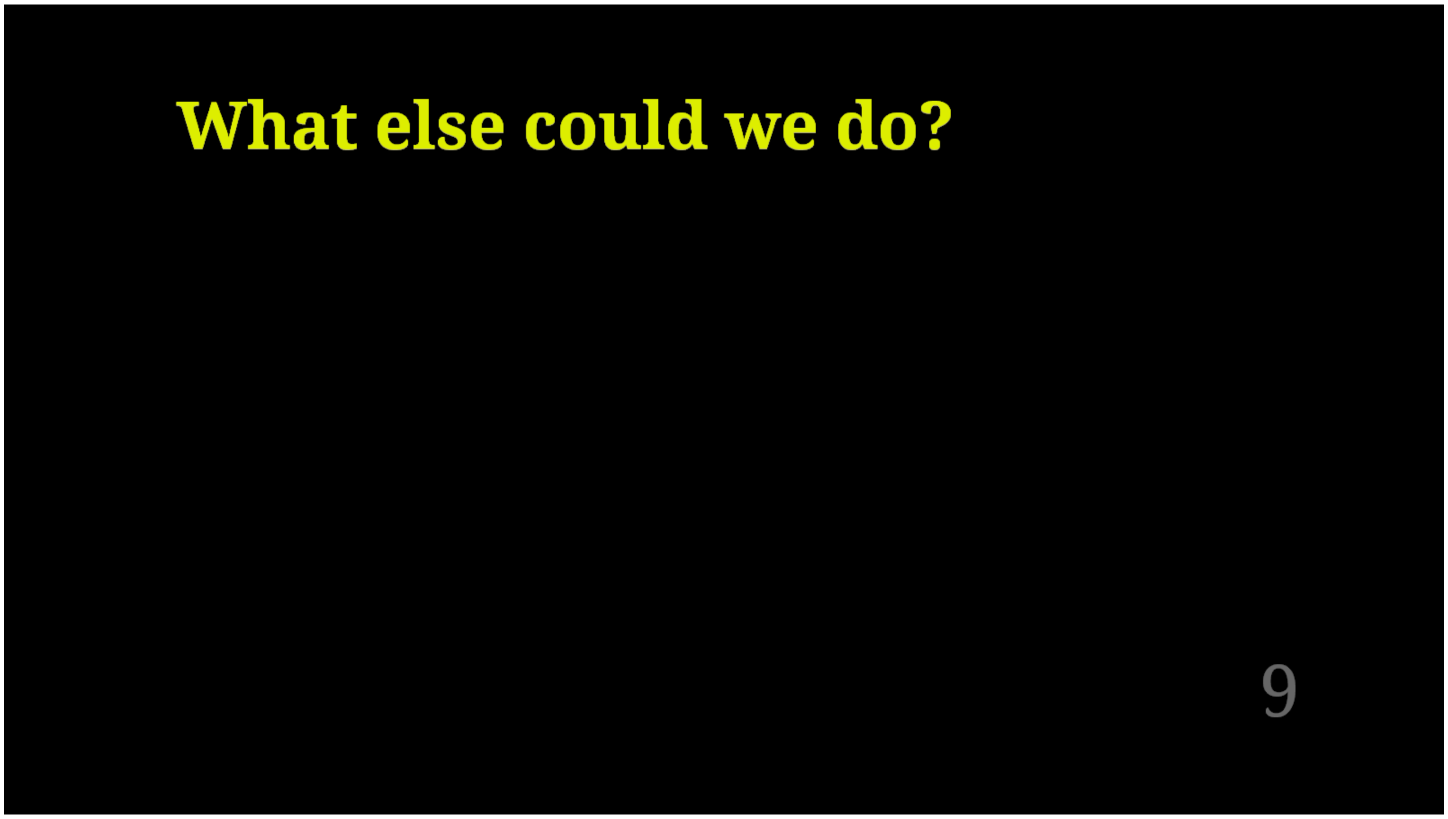
How does this work?

Overview:

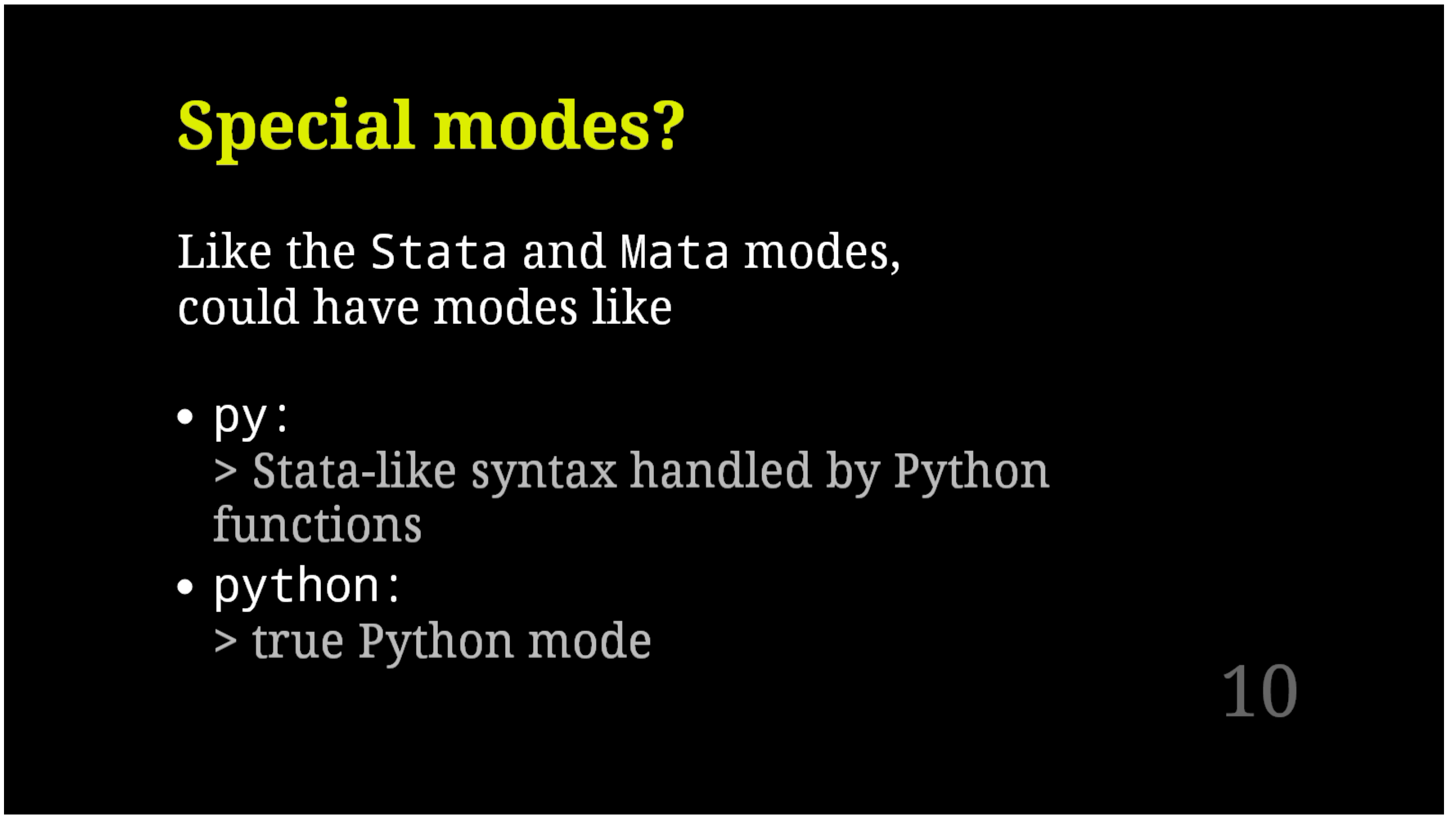
1. User enters a command into Python GUI.
2. The command is sent to Stata using OLE.
   > see www.stata.com/automation
3. Stata sends SMCL output to a log file.
4. Python watches the log file, and converts SMCL formatting to `tkinter` tags.

8

What I've been using here is an imitation of Stata's GUI, made with Python. When I open an instance of my imitation GUI, it generates a new instance of Stata and immediately instructs that Stata instance to start a log file. From then on, my GUI is aware of the Stata instance and its log file. Commands are sent to Stata, and results are retrieved from the log file.



What else could we do?

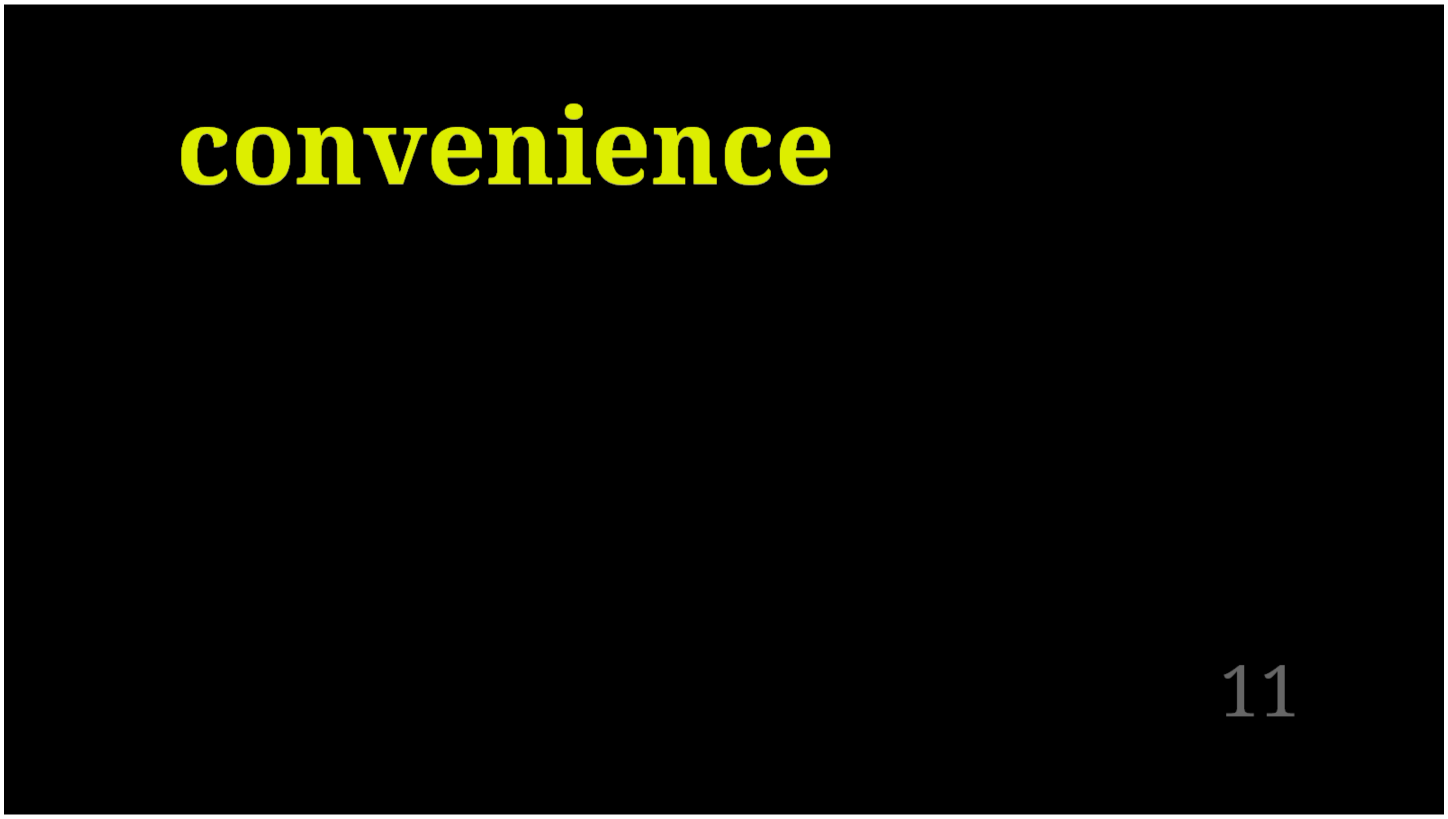


Special modes?

Like the `Stata` and `Mata` modes,
could have modes like

- `py:`
  > Stata-like syntax handled by Python functions
- `python:`
  > true Python mode

For example, the “`py:`” mode was used in the previous slides to load datasets into Python.



convenience

This is the second type of benefit from a Stata/Python combination.

This category includes those things that can already be done with Python and Stata separately, and combining them doesn’t add new functionality. Instead, the benefit would come from being able to issue a single command instead of several separate Python and Stata commands.

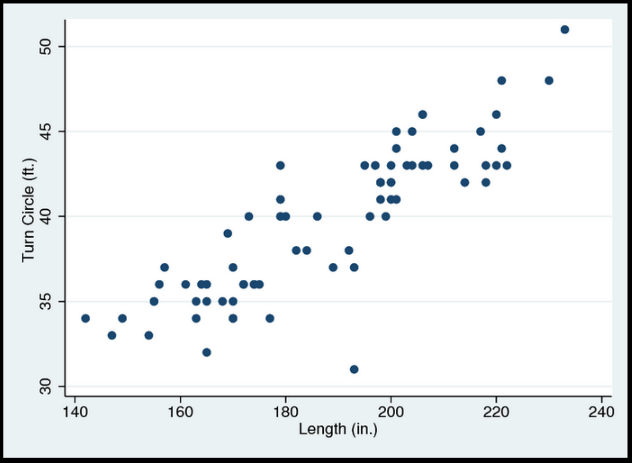
This convenience can already be achieved using OLE automation within Python, as used in the previous examples (the GUI is not required for this). However, since Stata users will tend to be more comfortable with Stata than Python, a Stata-esque environment (like the imitation GUI) might be preferred even if the underlying code is Python.

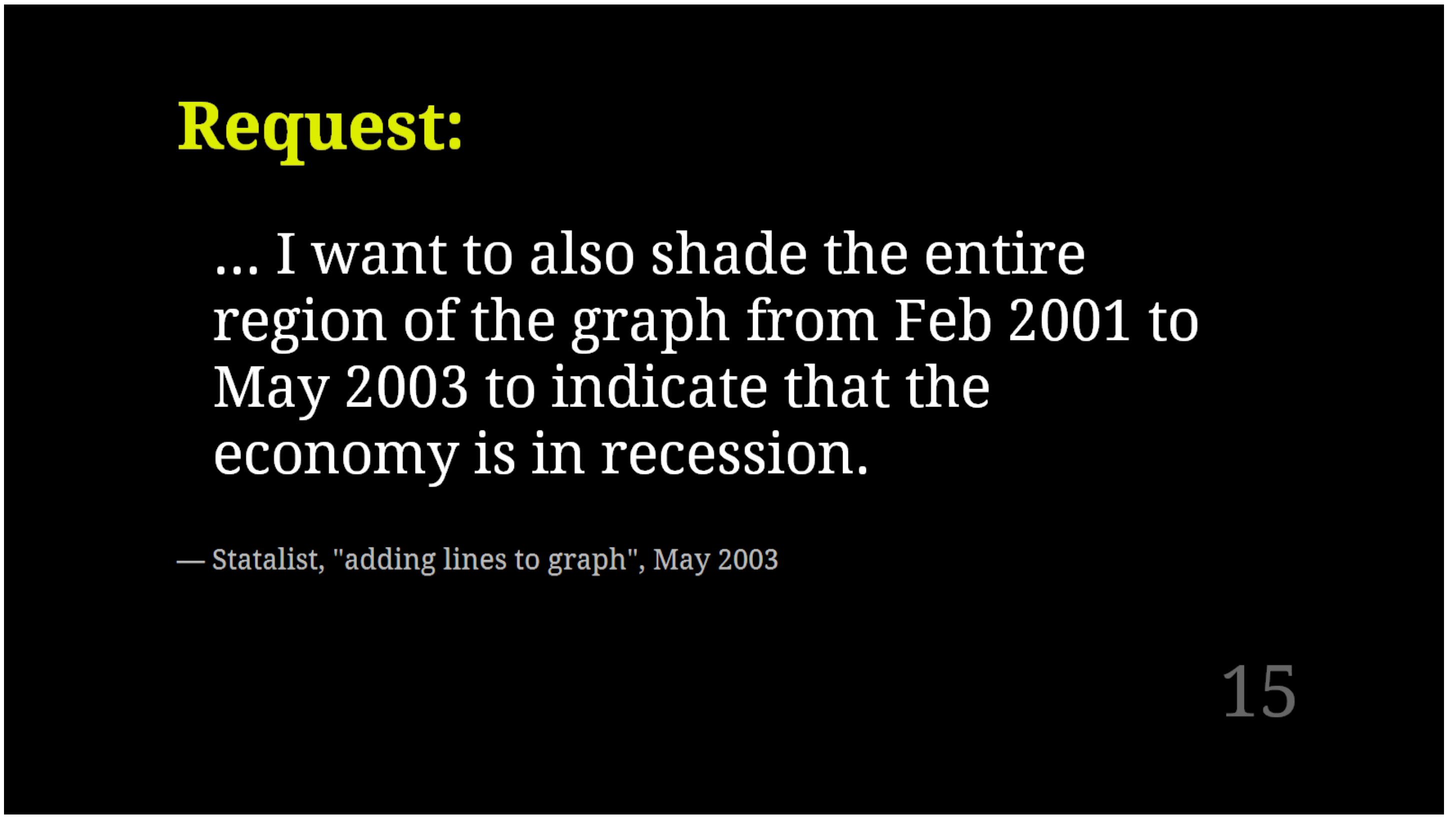
graphics

The third type of benefit.

all examples based on:

```
.sysuse auto
.scatter turn length
```





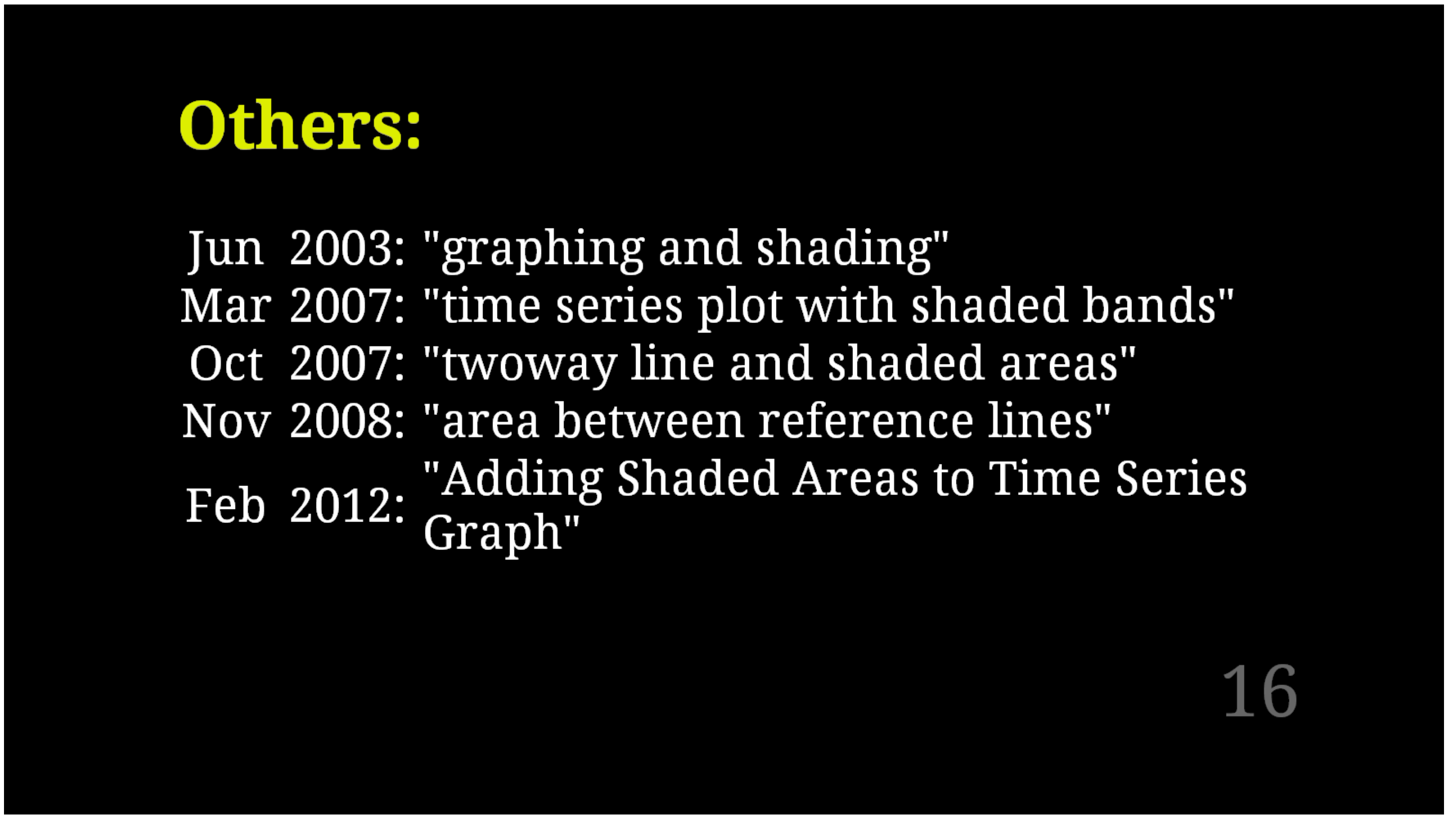
Request:

… I want to also shade the entire region of the graph from Feb 2001 to May 2003 to indicate that the economy is in recession.

— Statalist, "adding lines to graph", May 2003

15

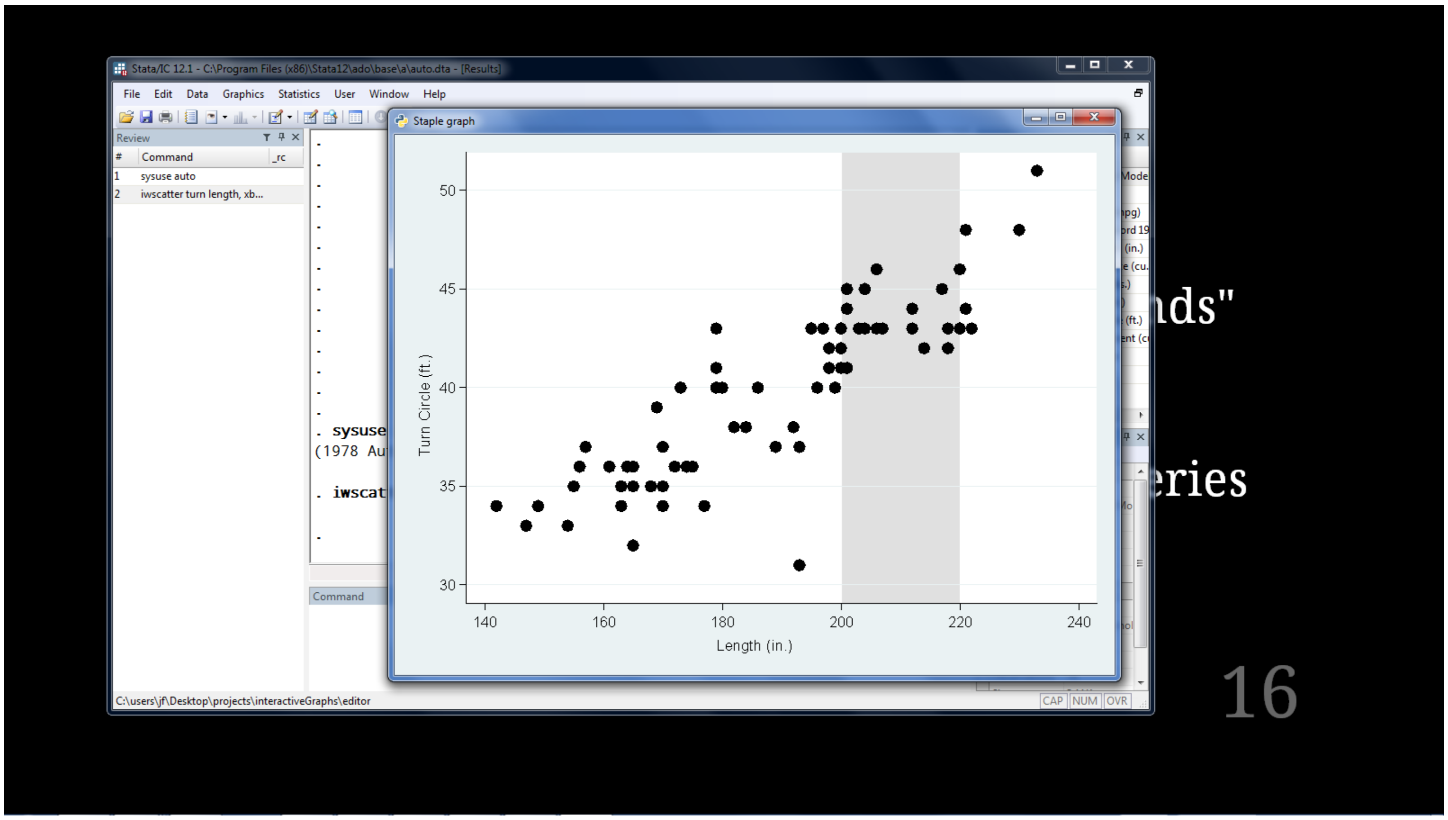
Here a user wants to shade a vertical band between two given x values. This request has come up a few other times:



Others:

Jun	2003:	"graphing and shading"
Mar	2007:	"time series plot with shaded bands"
Oct	2007:	"twoway line and shaded areas"
Nov	2008:	"area between reference lines"
Feb	2012:	"Adding Shaded Areas to Time Series Graph"

16


Stata/IC 12.1 - C:\Program Files (x86)\Stata12\ado\base\a\auto.dta - [Results]
File Edit Data Graphics Statistics User Window Help
Review
Command
_rc
Staple graph
Turn Circle (ft.)
Length (in.)
50
45
40
35
30
140
160
180
200
220
240
C:\users\jf\Desktop\projects\interactiveGraphs\editor
CAP NUM OVR
16

```
1  sysuse auto
2  iwscatter turn length, xb...
```

```
. sysuse
(1978 Au

. iwscat
```

There's a workaround for this where the user plots the data once to find the vertical bounds (or finds these bounds some other way), then generates a graph with a custom-made shaded area in the background. For this to work , the user also has to remove a margin around the inner plot region.

The workaround is a little inconvenient. It would be nice if there were an option similar to `xline` or `yline` that would take care of the details for you.

In the slide above, I've changed

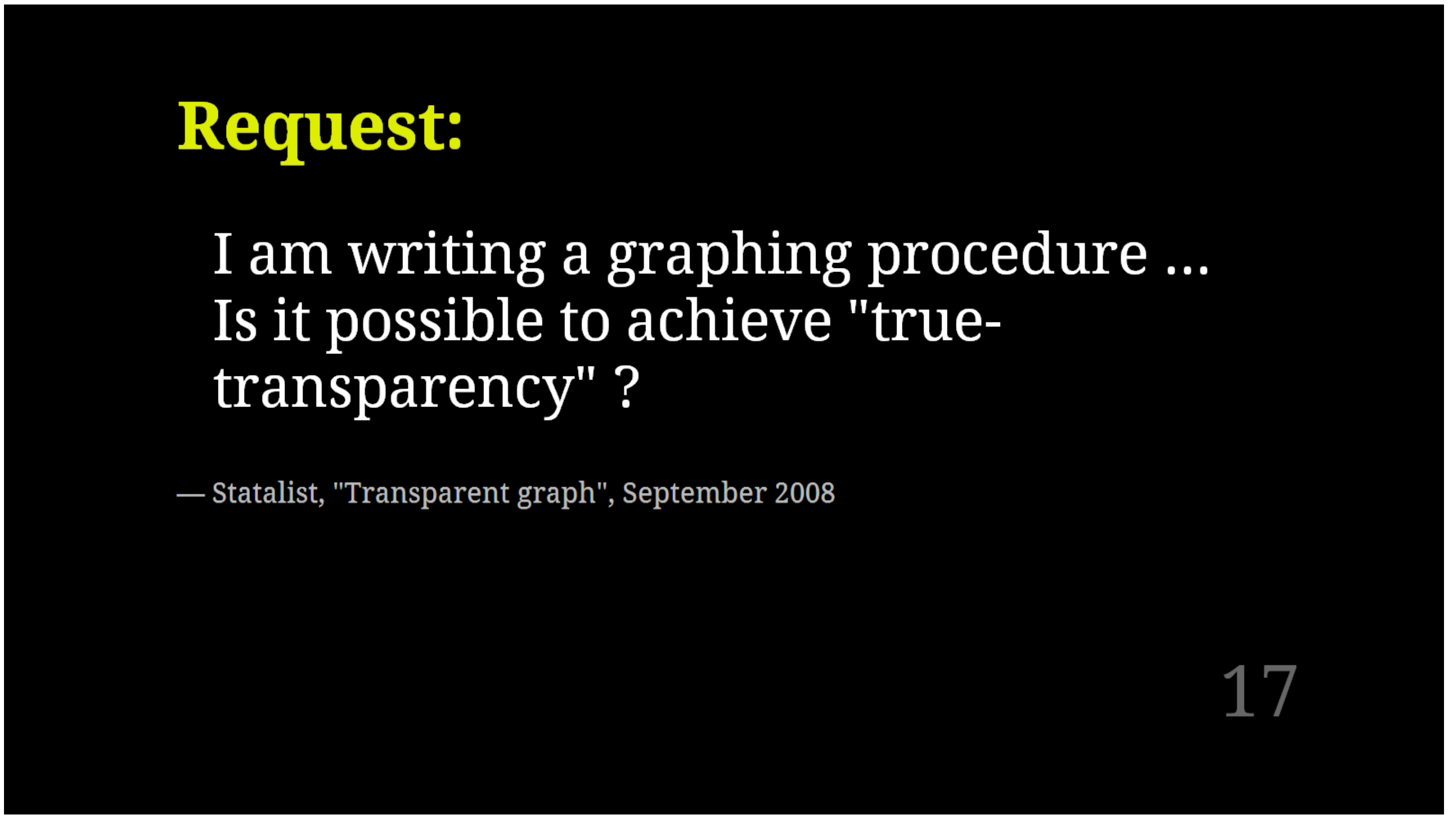
```
. scatter turn length
```

to

```
. iwscatter turn length, xband(200 220, color(gs14))
```

Instead of calling Stata's `scatter` command, I am calling my own command, `iwscatter`, and I've added the option `xband`.

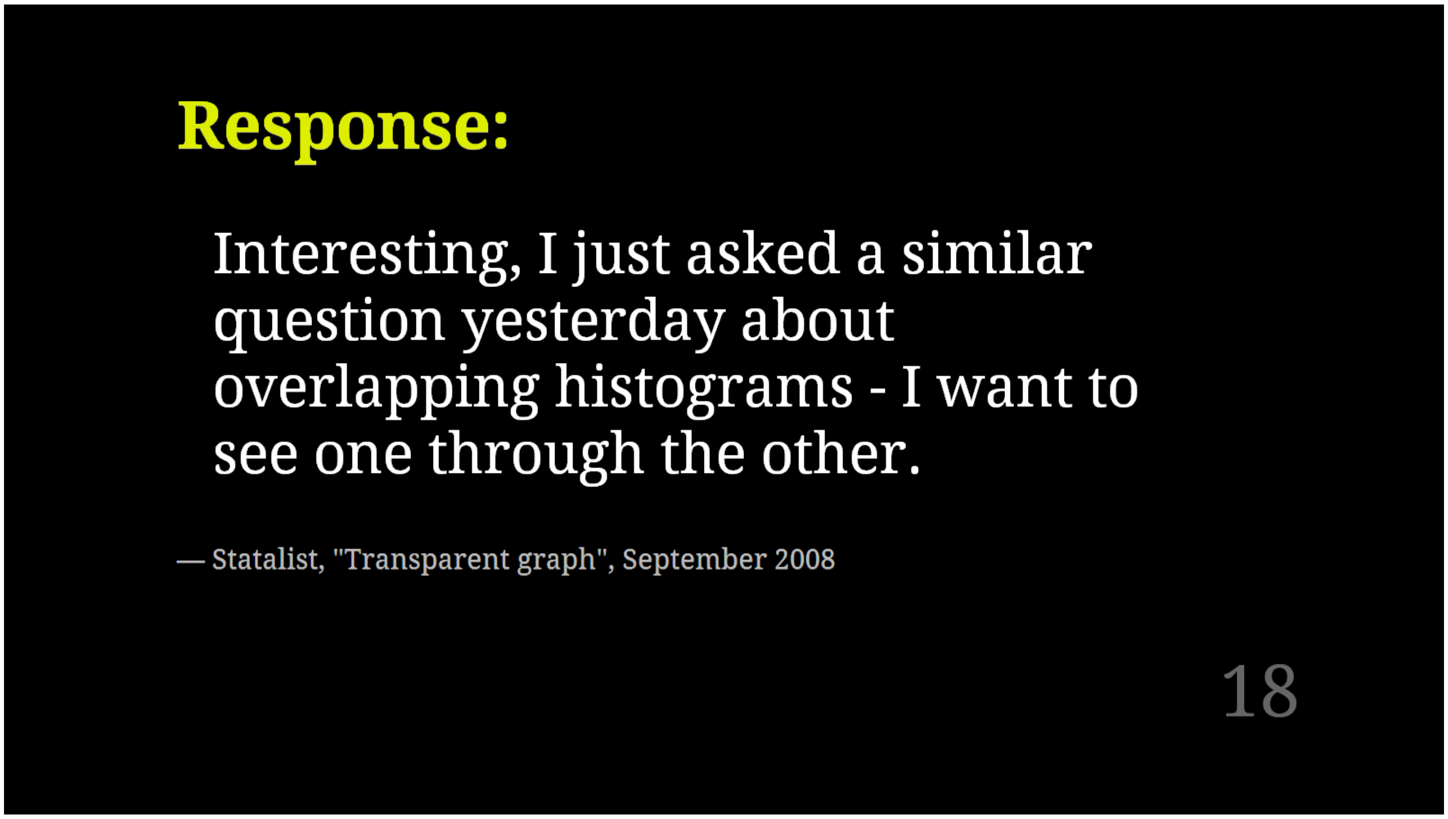
(Note: The `nbercycles` package from SSC can also be used to draw shaded bands.)


Request:

I am writing a graphing procedure … Is it possible to achieve "true-transparency" ?

— Statalist, "Transparent graph", September 2008

17

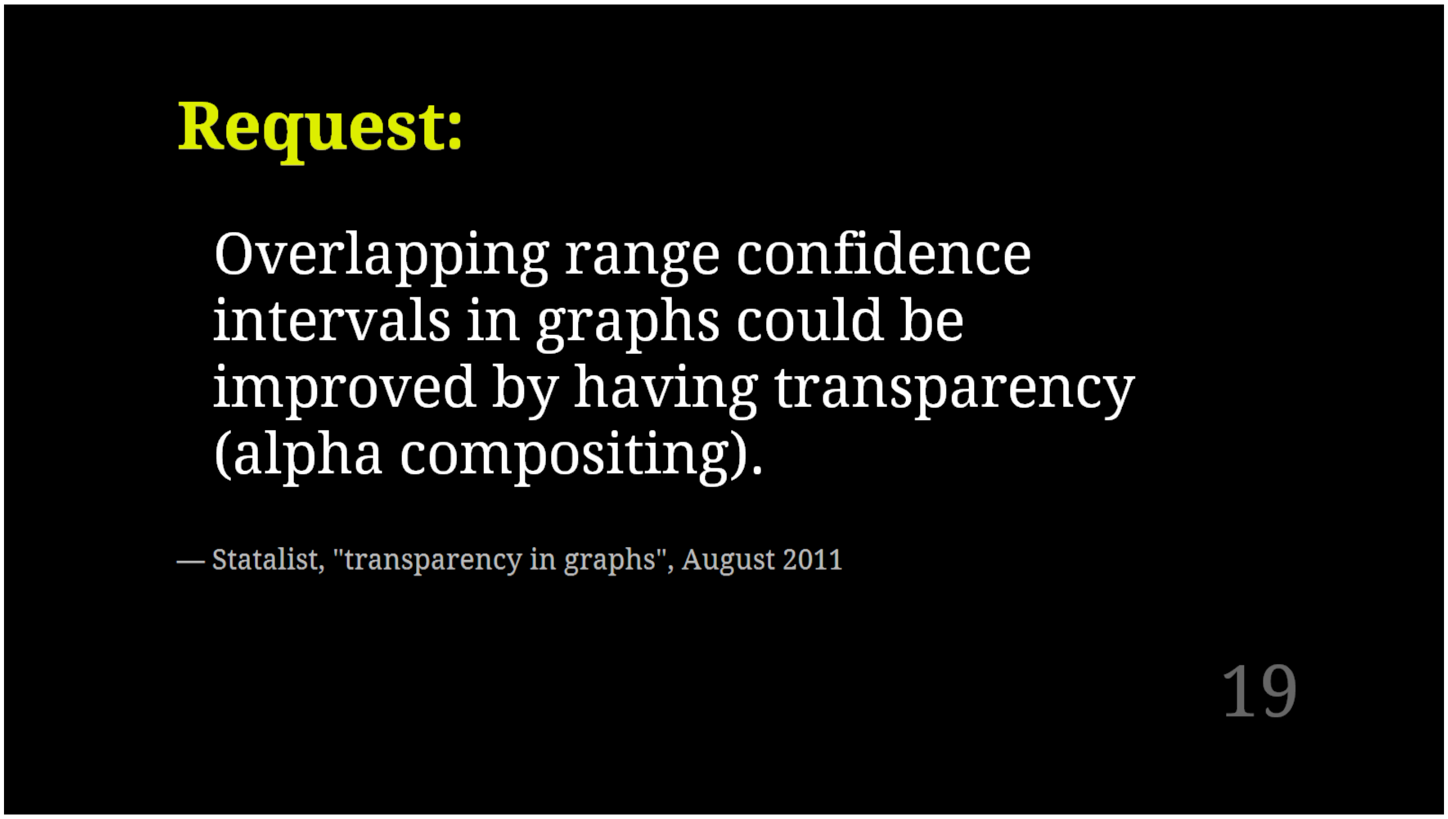

Response:

Interesting, I just asked a similar question yesterday about overlapping histograms - I want to see one through the other.

— Statalist, "Transparent graph", September 2008

18

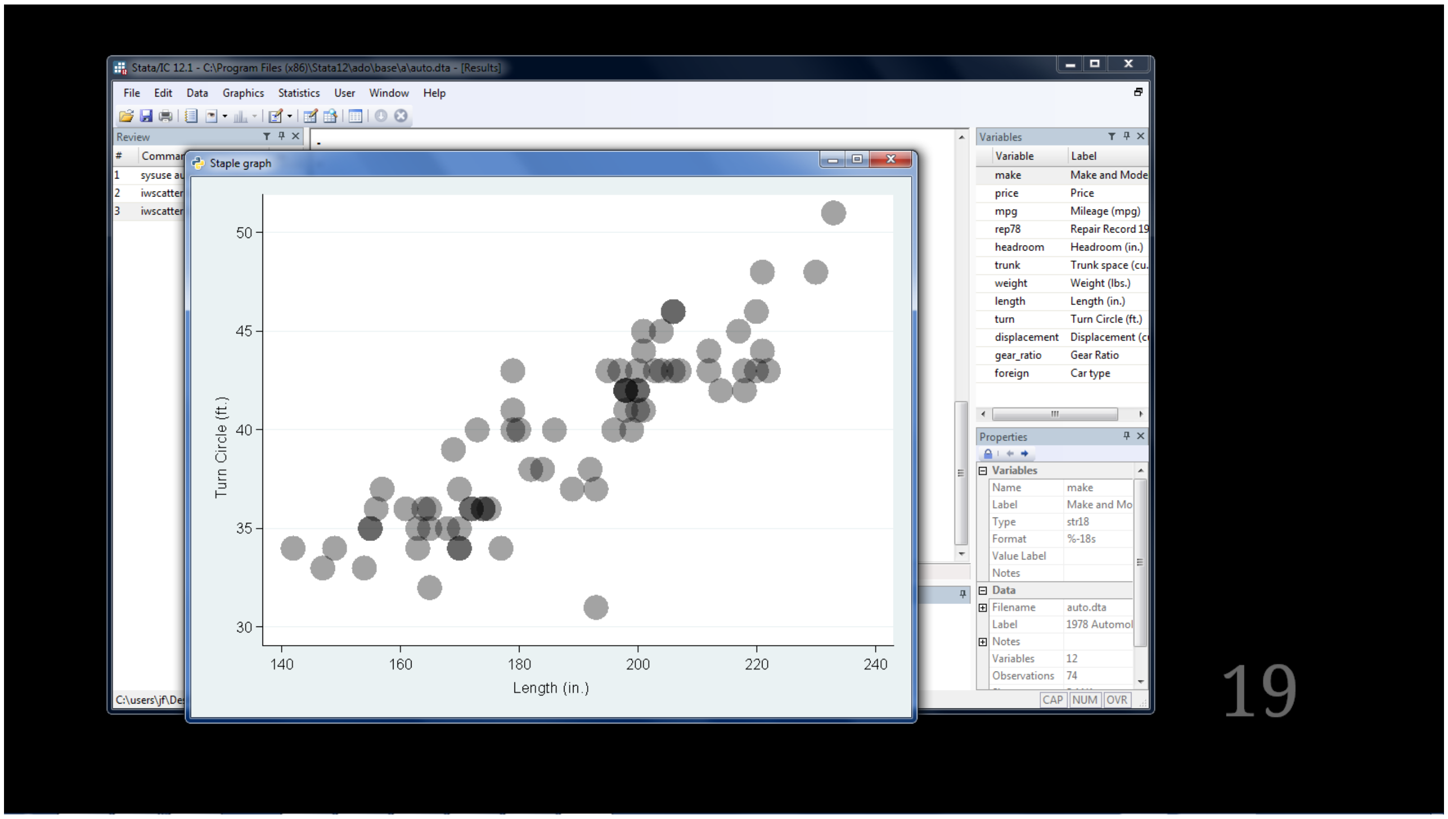
Transparency levels have been requested a few times, for a few different uses.



Request:

Overlapping range confidence intervals in graphs could be improved by having transparency (alpha compositing).

— Statalist, "transparency in graphs", August 2011


Stata/IC 12.1 - C:\Program Files (x86)\Stata12\ado\base\a\auto.dta - [Results]
File Edit Data Graphics Statistics User Window Help
Review
Staple graph
Turn Circle (ft.)
30
35
40
45
50
140
160
180
200
220
240
Length (in.)
Variables
Properties
CAP NUM OVR

#	Comma
1	sysuse au
2	iwscatter
3	iwscatter

Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

Variables	
Name	make
Label	Make and Mo
Type	str18
Format	%-18s
Value Label	
Notes	
Data	
Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74

C:\users\jf\De

Here I am using transparency to show overlap in data points, and I increased the size of the data markers to make the transparency easier to see. This graph was generated with

```
. iwscatter turn length, opacity(0.2) msize(14)
```

Request:

Compared to the myriads of graphs that R can do, Stata can only do simple plots.

— Statalist, "The Future of Statistical Computing", January 2009

Response:

Allow me to disagree on the graphics issue. … the possibilities are endless.

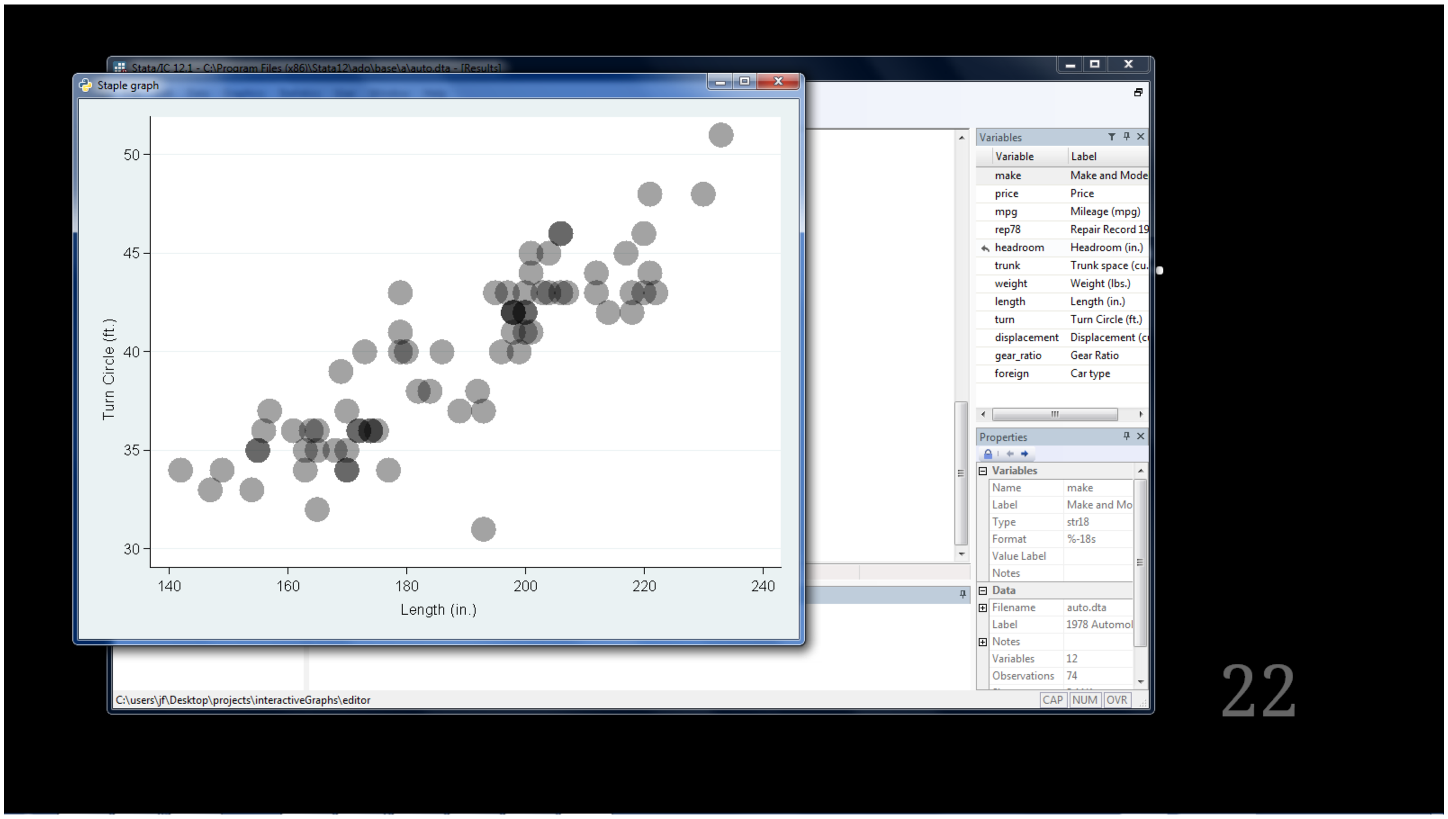
— Statalist, "The Future of Statistical Computing", January 2009

Response:

Yes, Stata graphics are very fexible.

However, wouldn't it be nice if …

— me



Variables

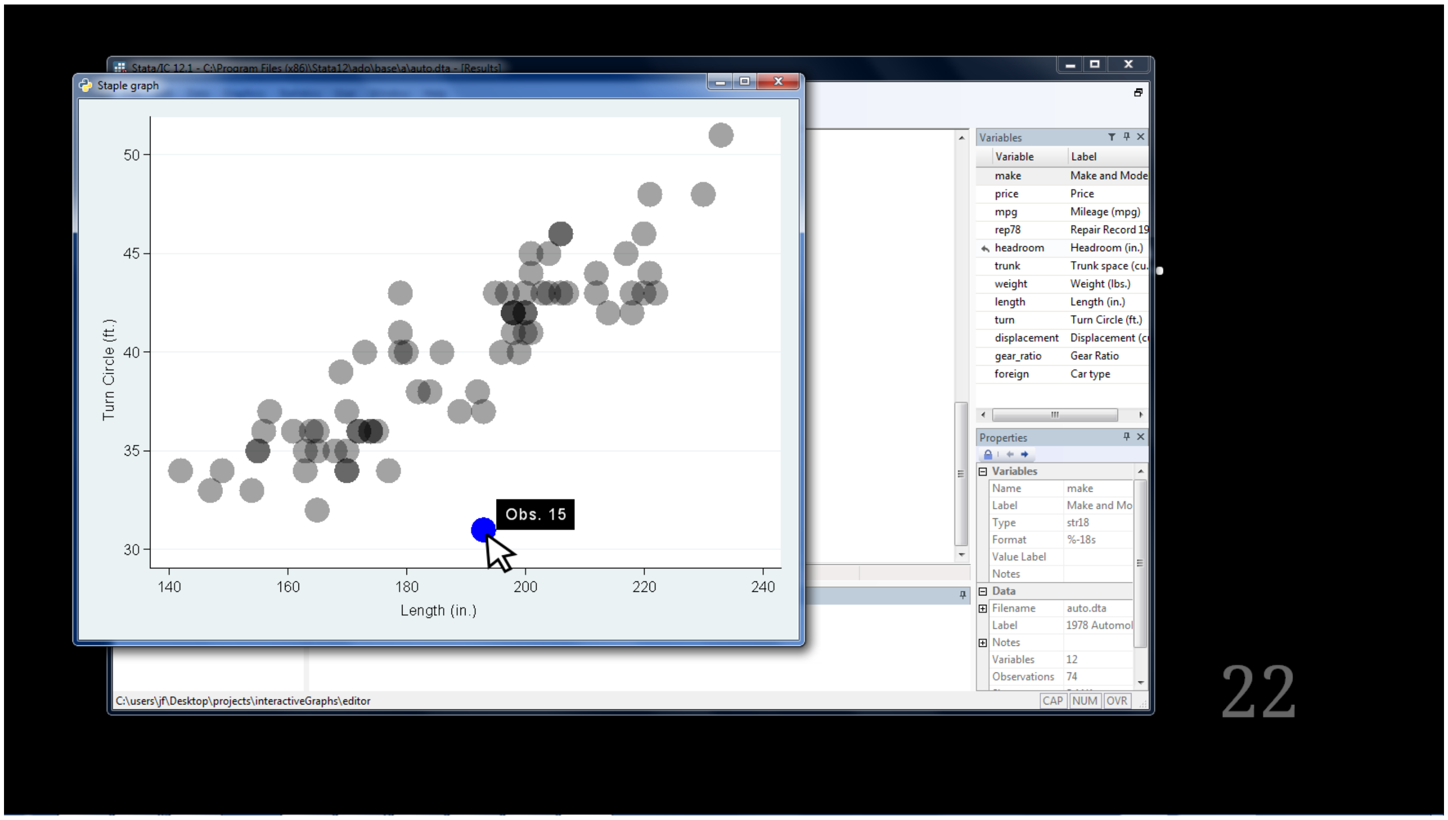
Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

Properties

Variables	
Name	make
Label	Make and Mo
Type	str18
Format	%-18s
Value Label	
Notes	
Data	
Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74

C:\users\jf\Desktop\projects\interactiveGraphs\editor

There's a clear outlier in this plot. When I see an outlier like this, the first thing I want to do is find out which observation it is. In order to do that I could use a few Stata commands, but wouldn't it be nice if you could just put the cursor over the data marker, and a popup would show the observation number?



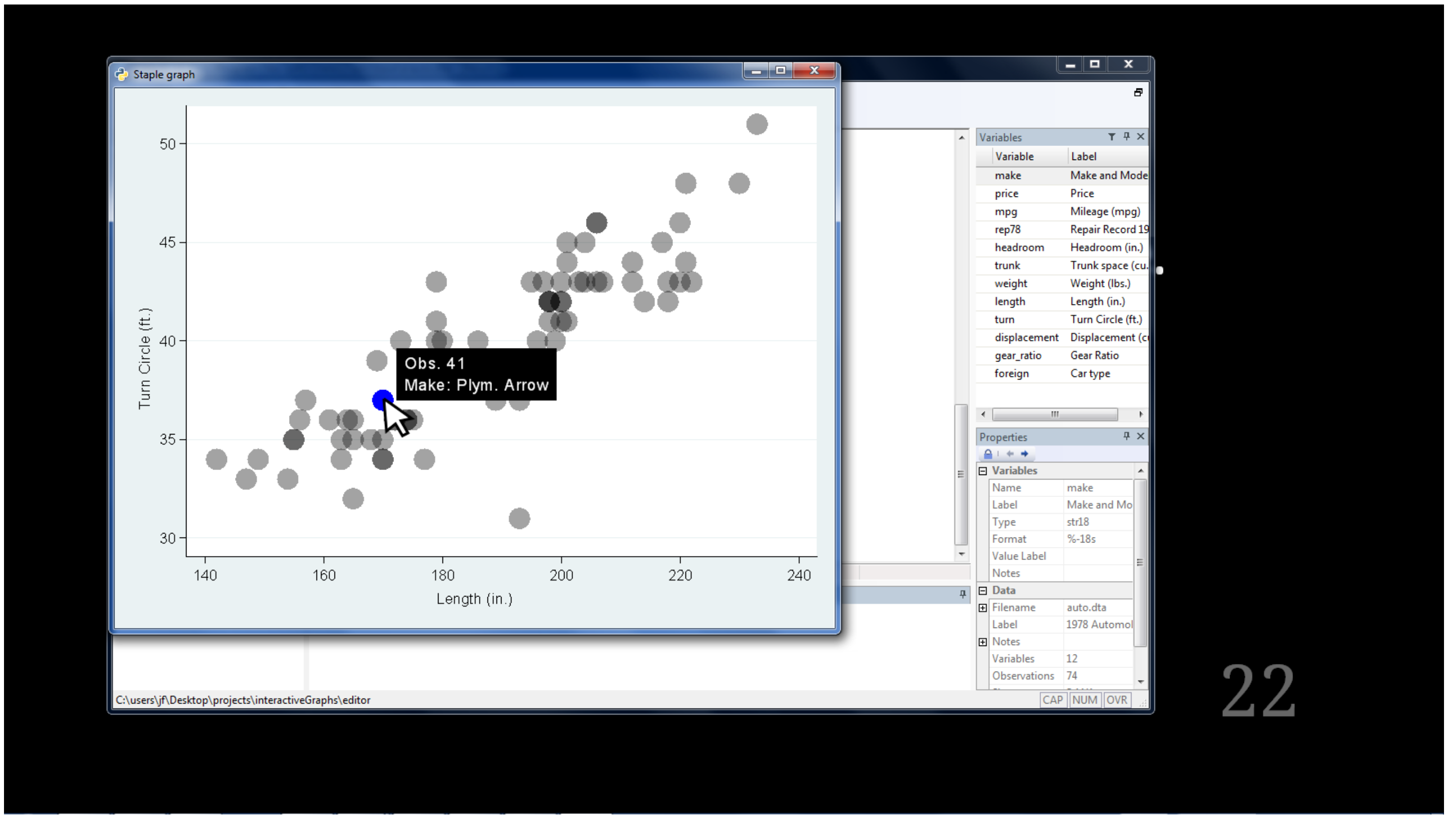
Variables

Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

Properties

Variables	
Name	make
Label	Make and Mo
Type	str18
Format	%-18s
Value Label	
Notes	
Data	
Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74

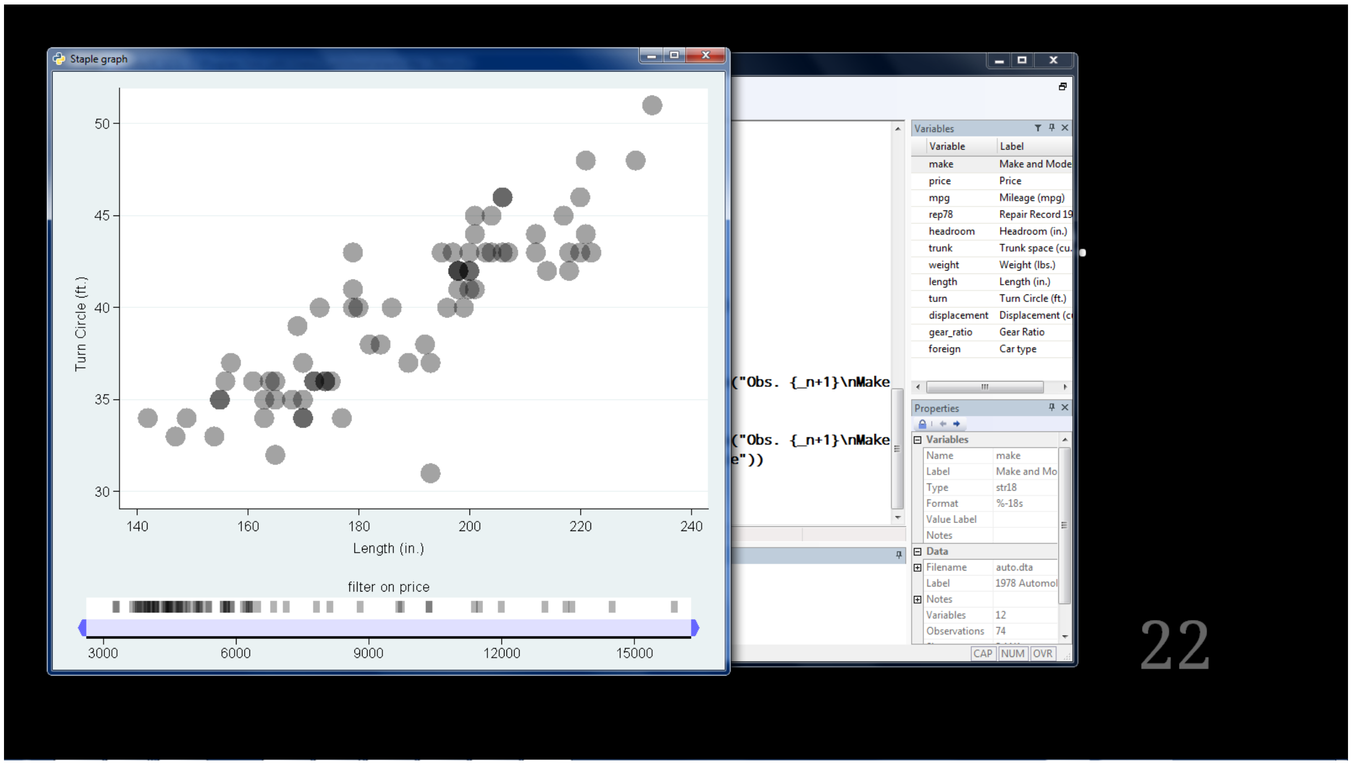
C:\users\jf\Desktop\projects\interactiveGraphs\editor



If we can do that, then it would also be nice if we could customize what appears in the info popup. The graph above is generated with

```
. iwscatter turn length, opacity(0.2) msize(12)
    info("Obs. {_n}\nMake: {make[_n]}")
```

(By the way, the observation numbers in these examples are off by one because Python starts indexing from 0 instead of 1.)



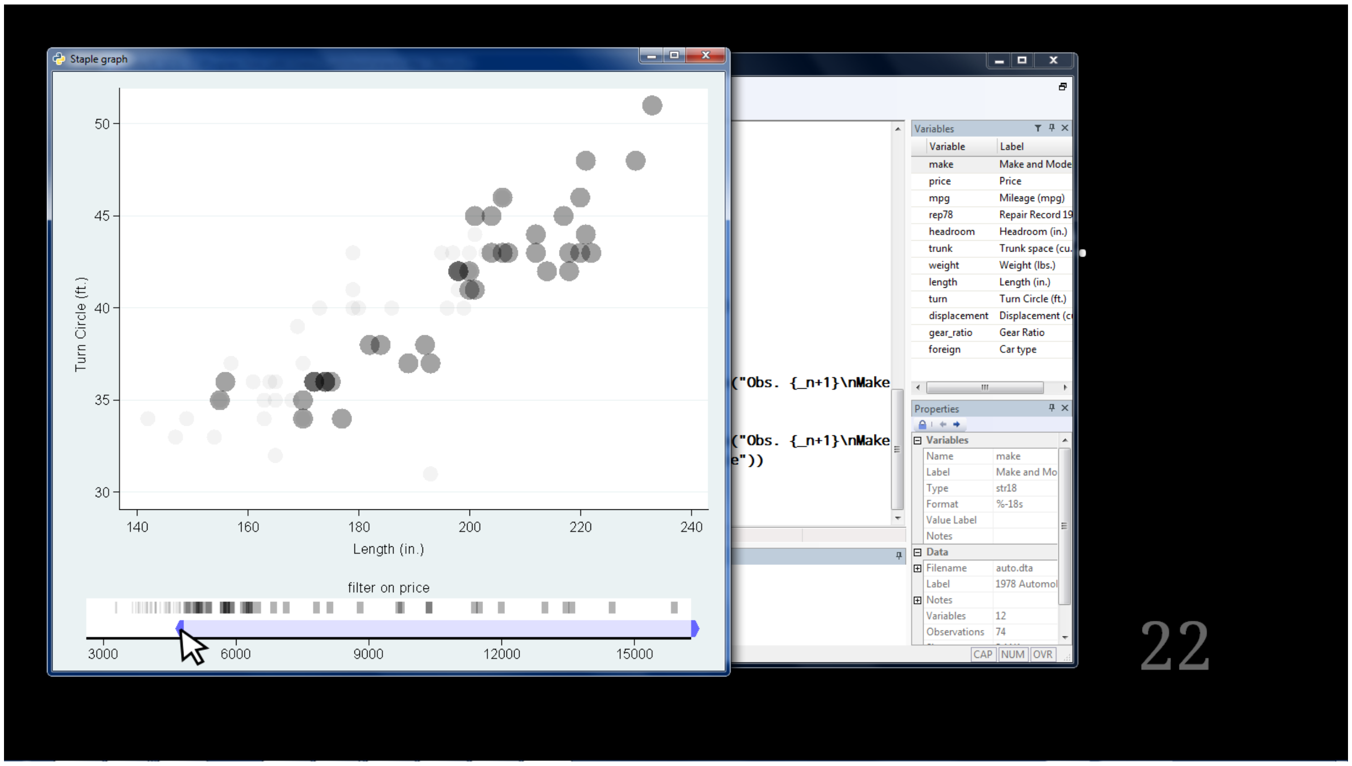
Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

Variables

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

Data

Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74



Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

Variables

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

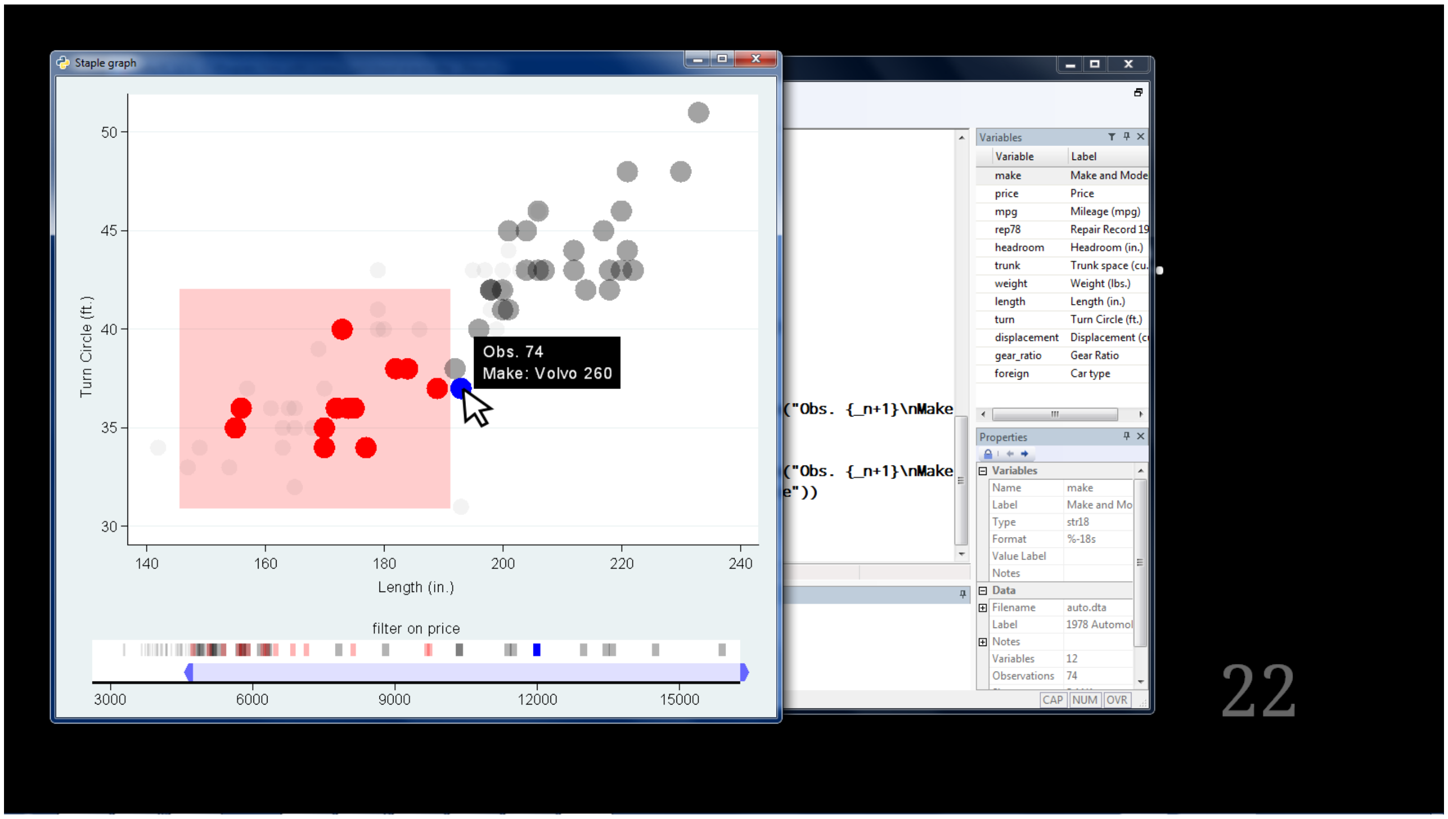
Data

Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74

Another idea: it would be useful to be able to filter based on another variable. Using

```
    . iwscatter turn length, opacity(0.2) msize(12)
        filter(price, title("filter on price"))
```

a rug plot is created under the scatterplot, with an adjustable bar that defines the range of included observations. Excluded observations are drawn in a lighter shade of gray ("ghosted").



```
("Obs. {_n+1}\nMake

("Obs. {_n+1}\nMake
e"))
```

Variables

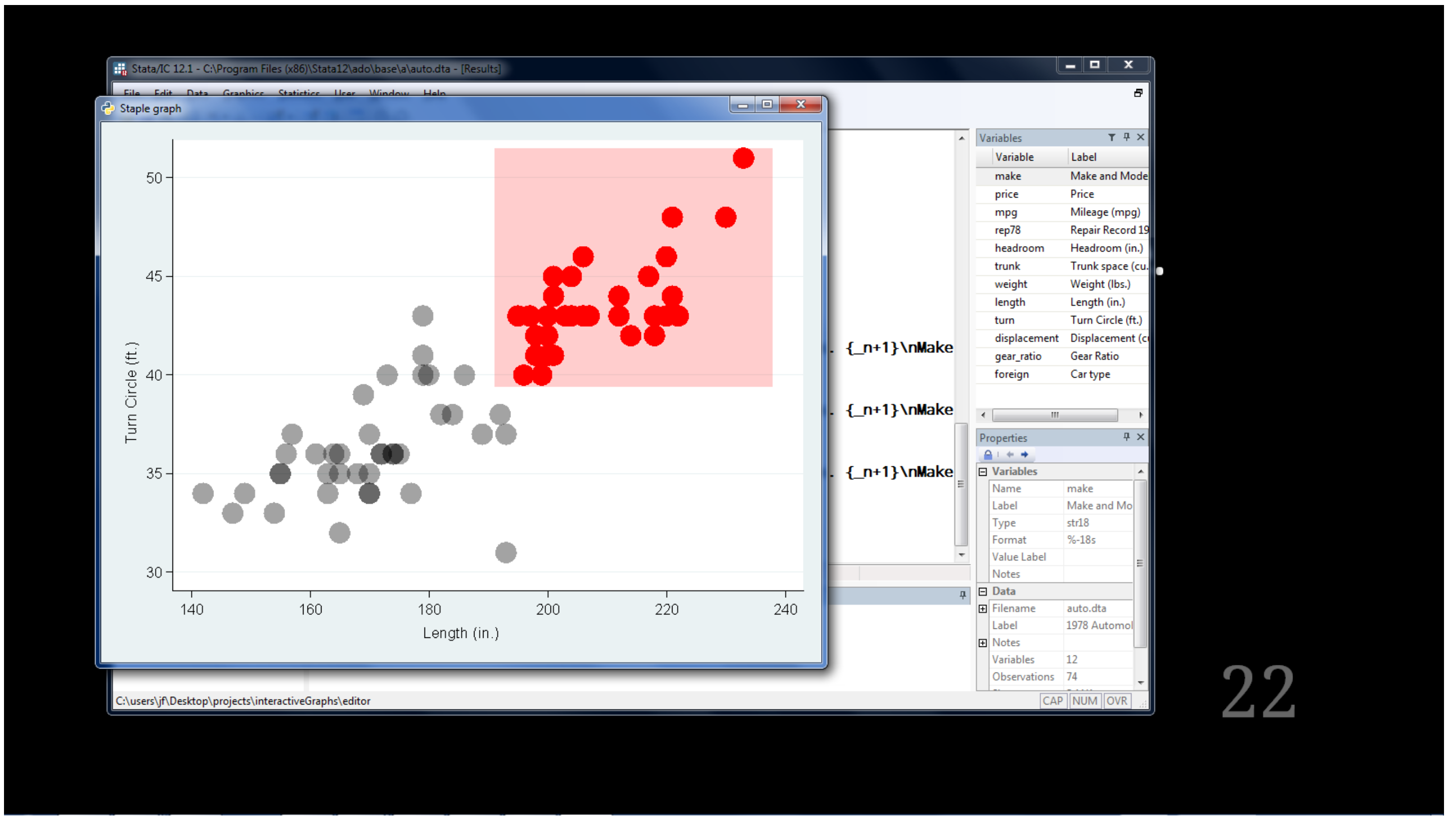
Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

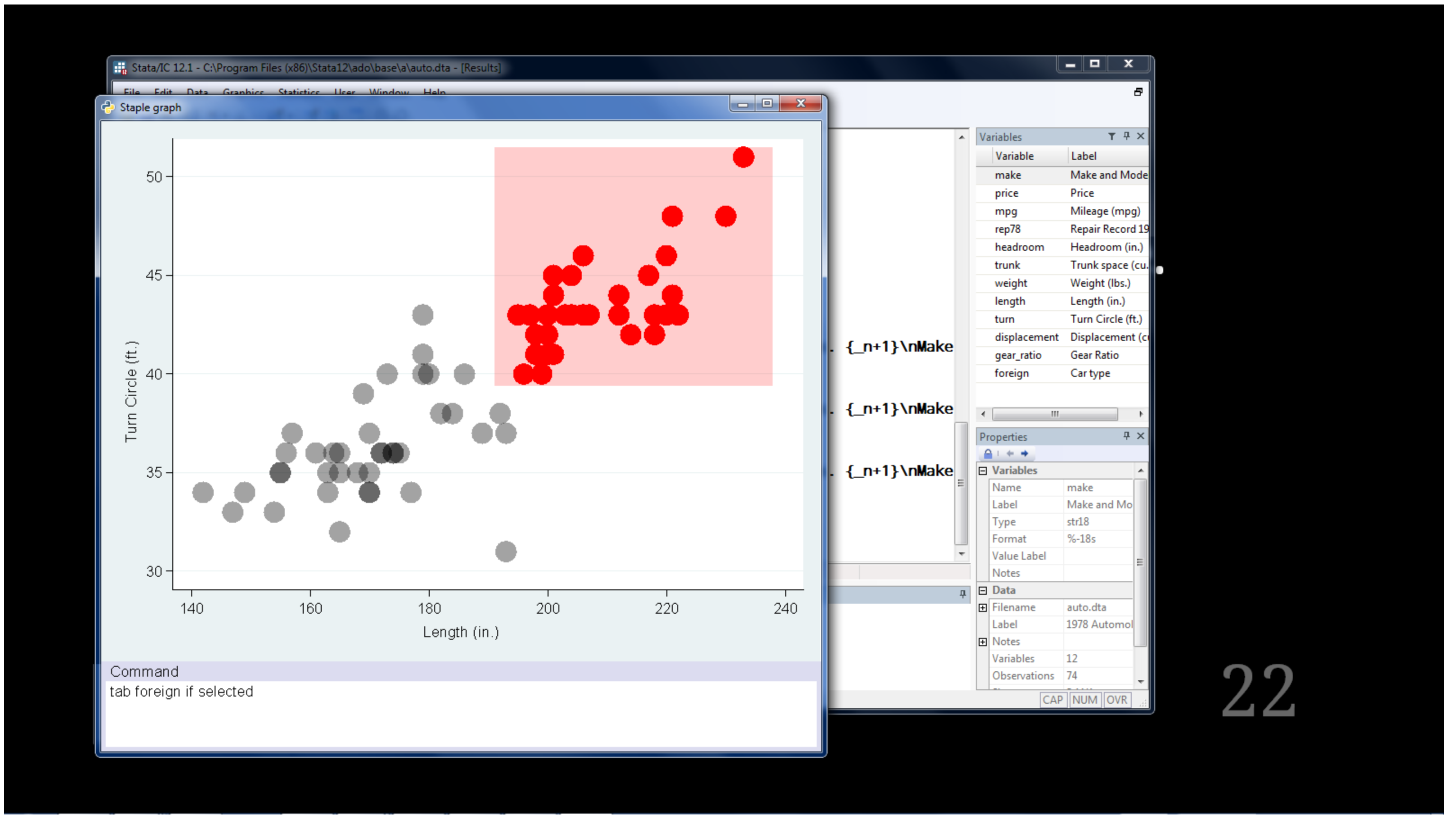
Properties

Variables	
Name	make
Label	Make and Mo
Type	str18
Format	%-18s
Value Label	
Notes	
Data	
Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74

Observations can also be “selected” using a selection rectangle.

Any plots that appear in the same window are automatically “linked”, so that interactions in one plot are mirrored in the others. Notice that the user is interacting with the top scatterplot, but corresponding changes appear in the rug plot.

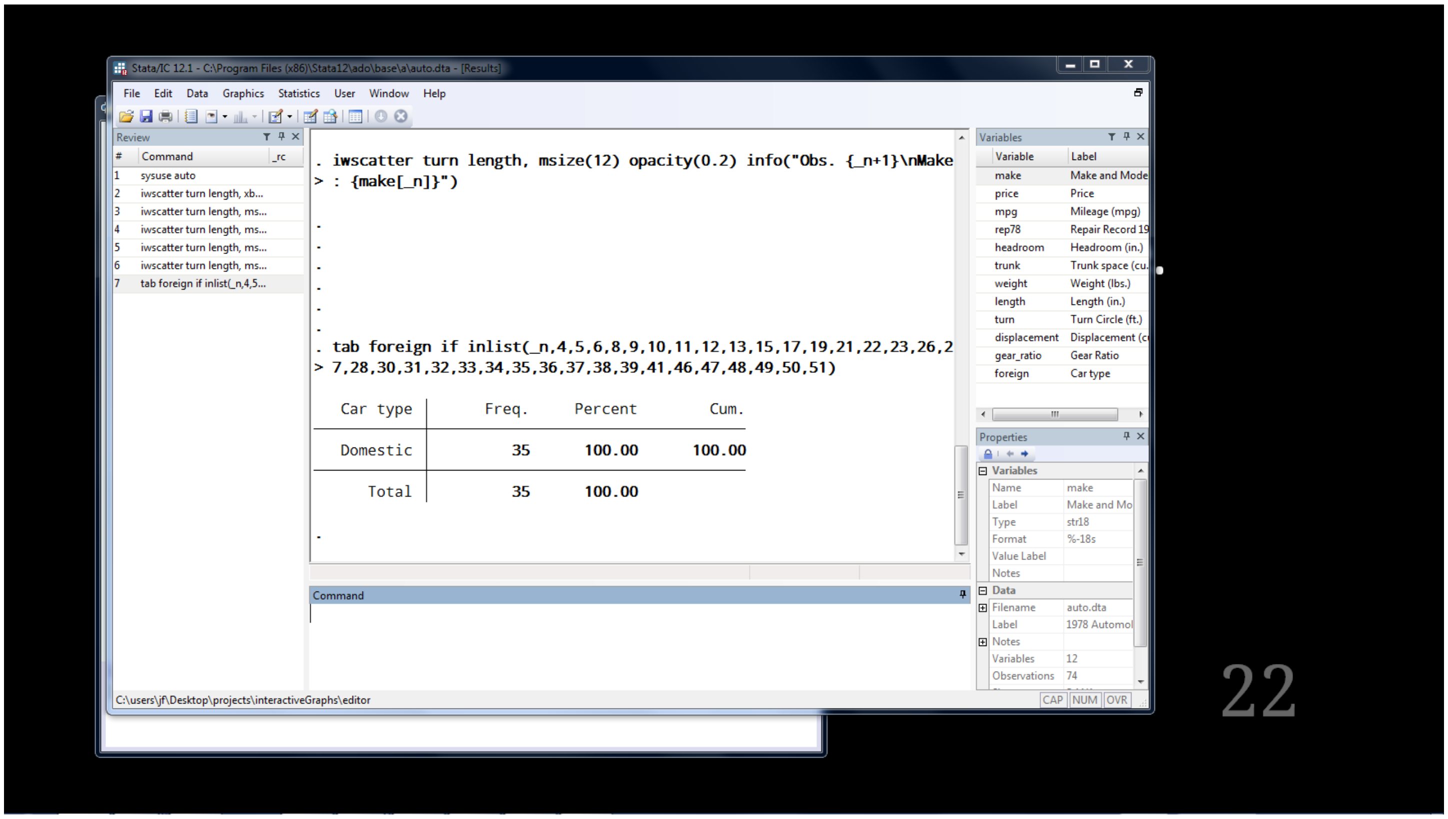

Stata/IC 12.1 - C:\Program Files (x86)\Stata12\ado\base\a\auto.dta - [Results]
Staple graph
Turn Circle (ft.)
Length (in.)
{_n+1}\nMake
Variables
Properties
C:\users\jf\Desktop\projects\interactiveGraphs\editor
22


Stata/IC 12.1 - C:\Program Files (x86)\Stata12\ado\base\a\auto.dta - [Results]
Staple graph
Turn Circle (ft.)
Length (in.)
{_n+1}\nMake
Variables
Properties
22

Command

```
tab foreign if selected
```

It would also be useful to be able to issue Stata commands that are aware of changes in the plot. Control-I opens a “command” box at the bottom of the window. From there commands can be submitted back to Stata.



Stata/IC 12.1 - C:\Program Files (x86)\Stata12\ado\base\a\auto.dta - [Results]

#	Command	_rc
1	sysuse auto	
2	iwscatter turn length, xb...	
3	iwscatter turn length, ms...	
4	iwscatter turn length, ms...	
5	iwscatter turn length, ms...	
6	iwscatter turn length, ms...	
7	tab foreign if inlist(_n,4,5...	

```
. iwscatter turn length, msize(12) opacity(0.2) info("Obs. {_n+1}\nMake
> : {make[_n]}")


.
.
.
.
.
.
. tab foreign if inlist(_n,4,5,6,8,9,10,11,12,13,15,17,19,21,22,23,26,2
> 7,28,30,31,32,33,34,35,36,37,38,39,41,46,47,48,49,50,51)

   Car type |      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         35      100.00      100.00
------------+-----------------------------------
      Total |         35      100.00

.
```

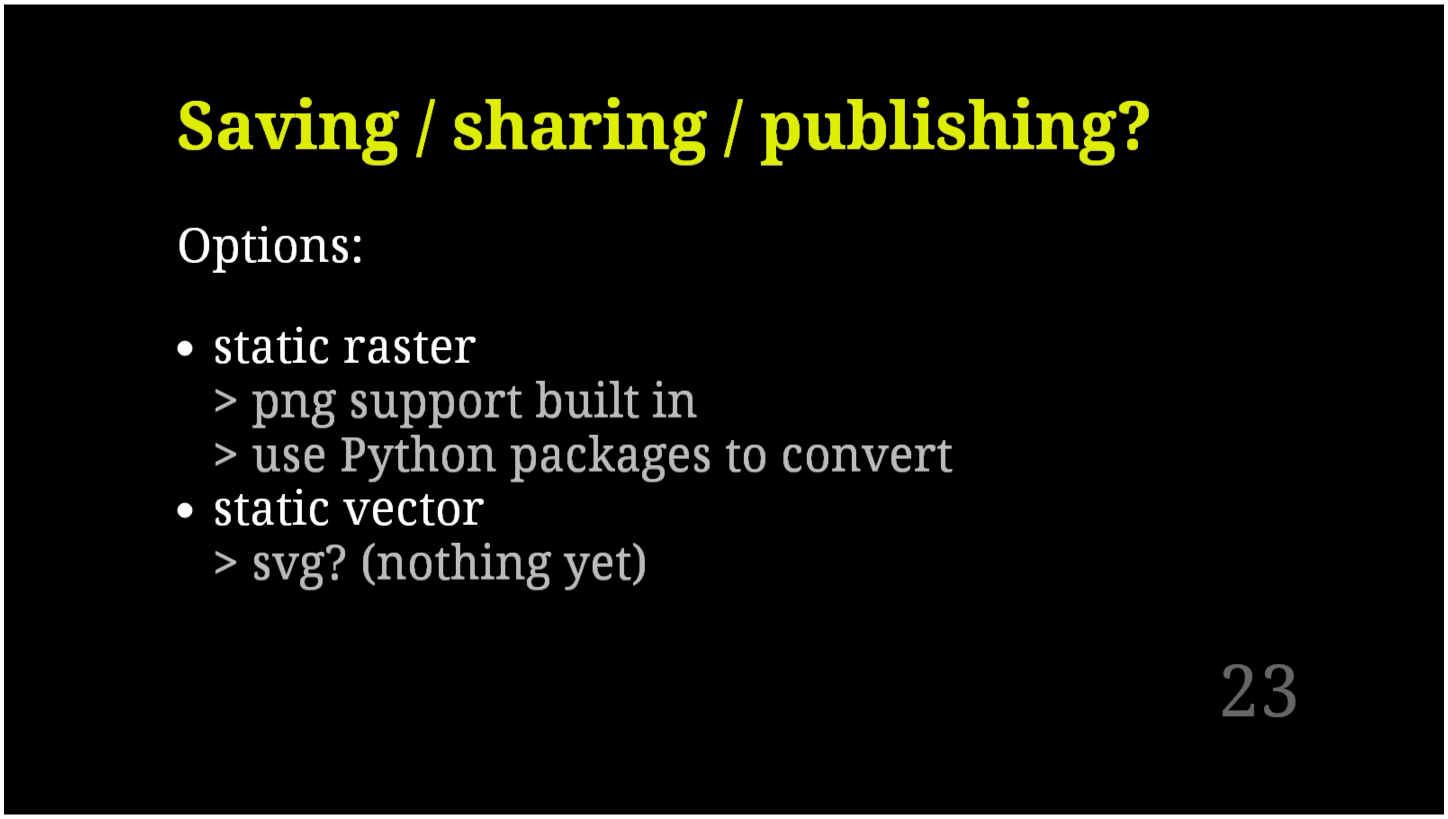
Variable	Label
make	Make and Mode
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 19
headroom	Headroom (in.)
trunk	Trunk space (cu.
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (c
gear_ratio	Gear Ratio
foreign	Car type

Properties	
Variables	
Name	make
Label	Make and Mo
Type	str18
Format	%-18s
Value Label	
Notes	
Data	
Filename	auto.dta
Label	1978 Automol
Notes	
Variables	12
Observations	74

C:\users\jf\Desktop\projects\interactiveGraphs\editor

There are keywords for the types of interaction: `active`, `selected`, and `ghosted`. When a command is submitted, if any of the keywords is present, it is changed to an `inlist` containing the desired observation numbers.

How these examples work: The `iwscatter` command writes a Python file, i.e., a text file with Python commands, and then instructs Python to run the file. The graphs are drawn using a Python module called Pyglet, which is a wrapper for OpenGL functions. Communication back to Stata is accomplished with a module called pywinauto.

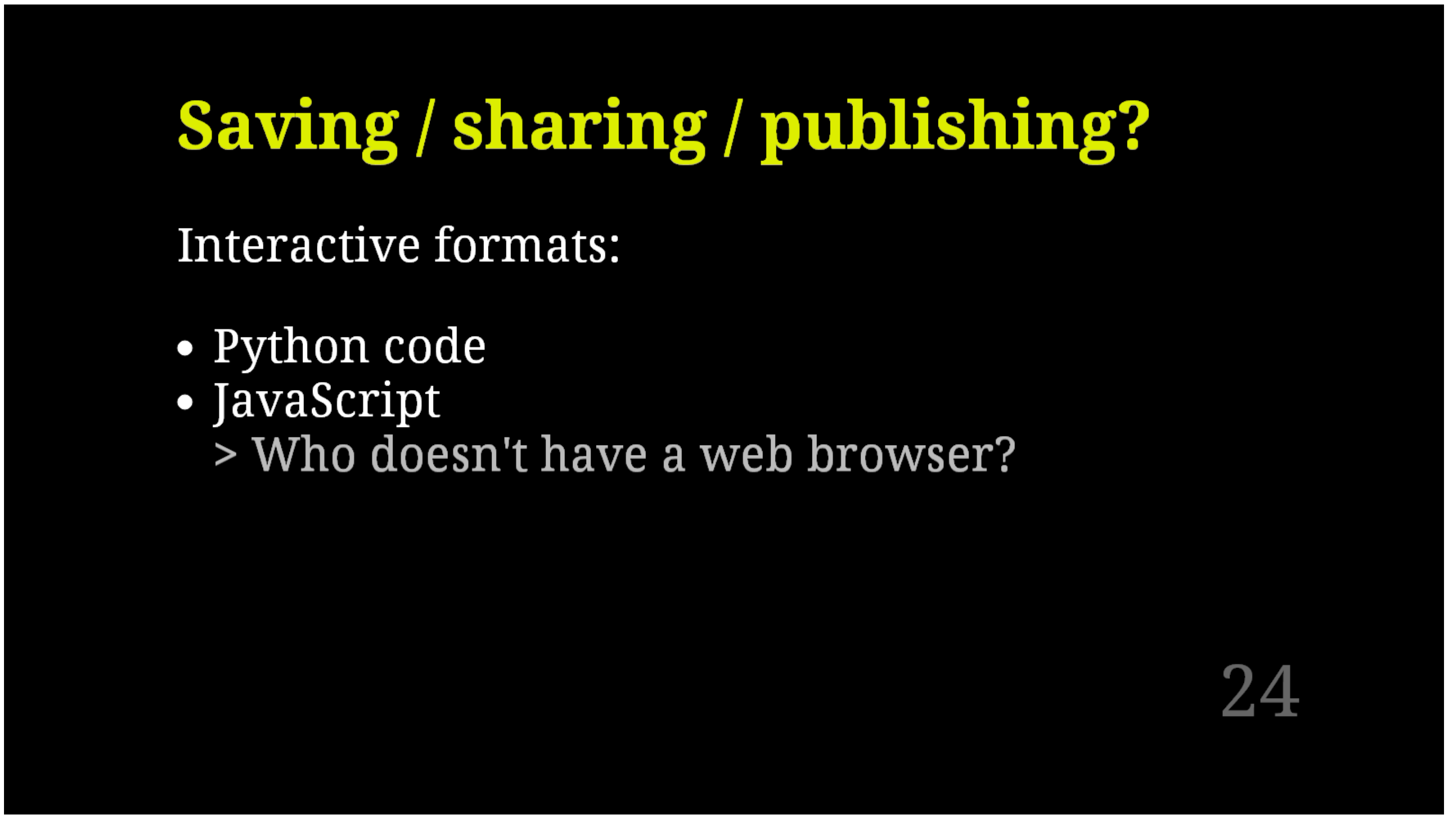


Saving / sharing / publishing?

Options:

- static raster
  > png support built in
  > use Python packages to convert
- static vector
  > svg? (nothing yet)

23



Saving / sharing / publishing?

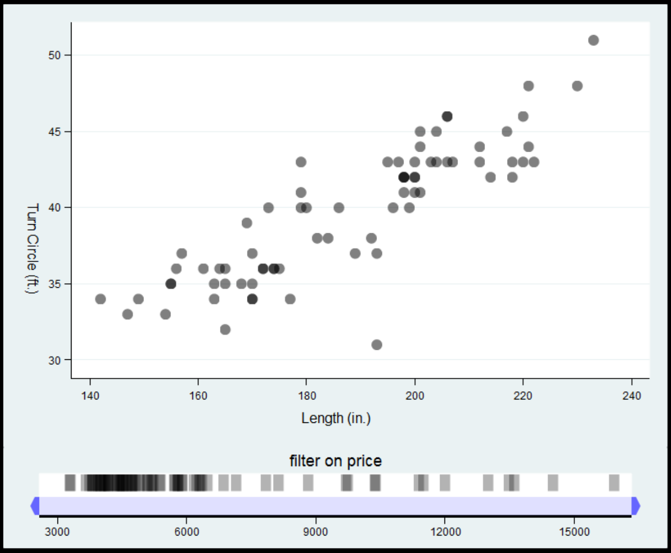
Interactive formats:

- Python code
- JavaScript
  > Who doesn't have a web browser?

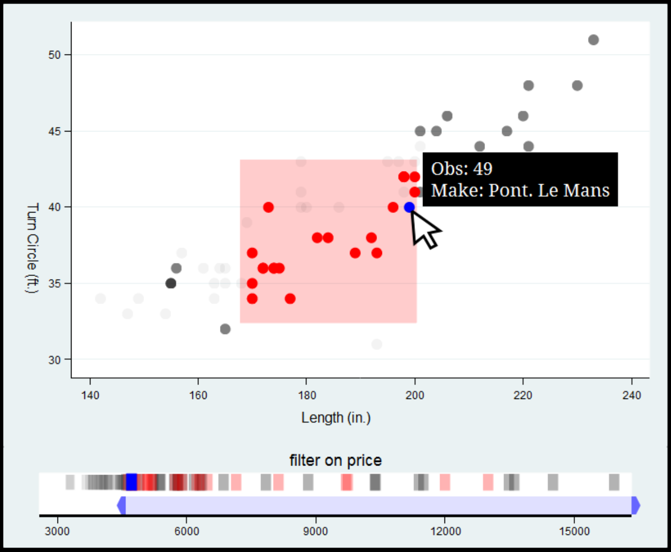
24

Stata graphs serve as tools for both investigation and publication. In order to serve the same purposes, the interactive graphs would need a format for saving and publishing. For fully interactive graphs, users with the required modules could exchange Python files, but for general publication, JavaScript would be the better option due to its ubiquity.

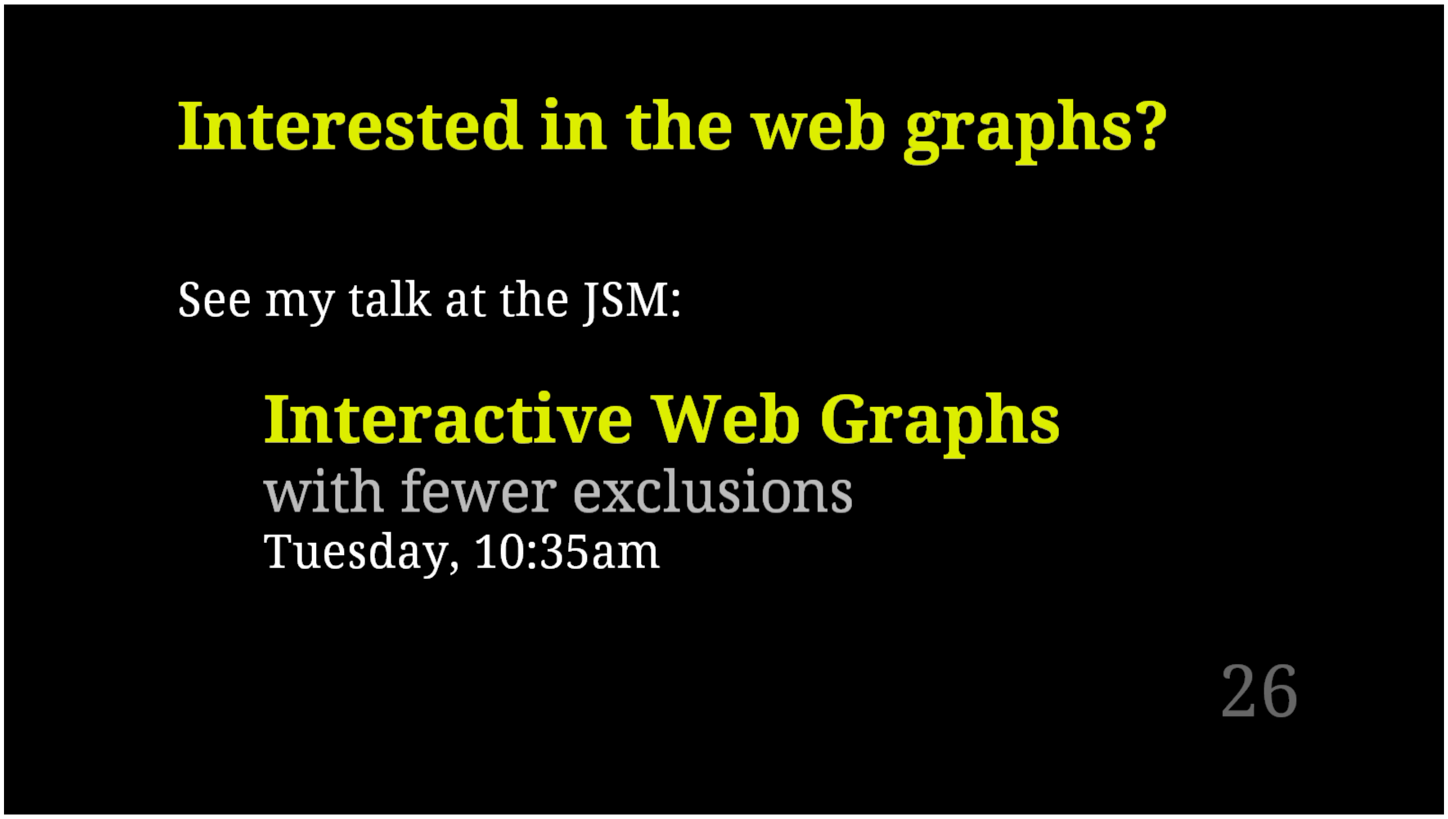
JavaScript



JavaScript



This shows a working JavaScript version of the Python graphs.



Interested in the web graphs?

See my talk at the JSM:

Interactive Web Graphs
with fewer exclusions
Tuesday, 10:35am

26

resources

- Python
  python.org
- Pyglet
  pyglet.org
- pywinauto
  code.google.com/p/pywinauto
- Stata automation
  stata.com/automation

27

Contact: j r f i e d l e r @ gmail.com

I can also be contacted at `james.fiedler-1@nasa.gov`