Introduction to the Stata Language, Part 2

Mark Lunt

Centre for Epidemiology Versus Arthritis
University of Manchester

20/12/2022

Summary

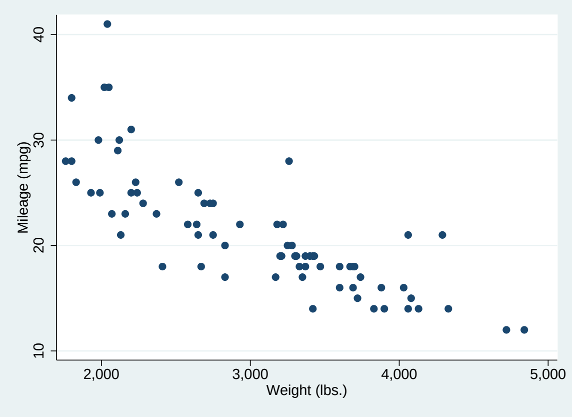
- Graphics
- Summarizing Data
- More Stata Syntax
- Looping
- Reshaping Data
- Other Useful Commands

Graphics

- Scatter plots
- Labelling
- Overlaying plots
- Schemes
- Saving & Exporting

Scatter Plots

```
twoway scatter mpg weight
```



Labelling

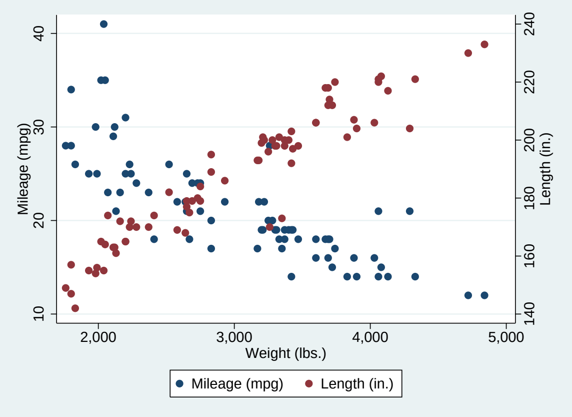
Titles `title(), subtitle(), note(), caption()`

Axis names `xtitle, ytitle`

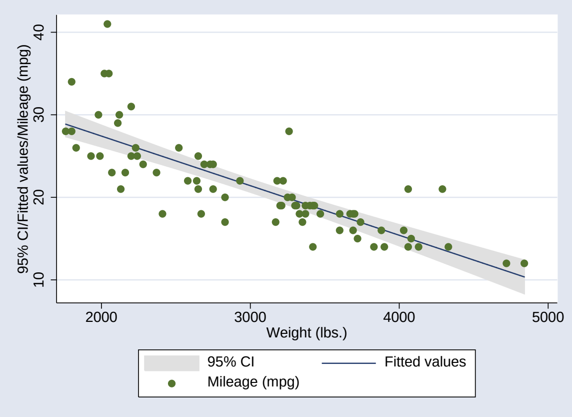
Tick marks `xlabel, ylabel`

Overlaying Graphs

```
twoway (scatter mpg weight) (scatter length
weight, yaxis(2))
```

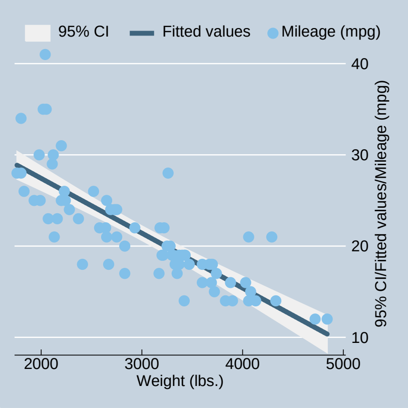


```
twoway lfitci mpg weight || scatter mpg weight
```



Schemes

- Can change appearance of graph:
  - Line thickness
  - Colour or B/W
  - Text size
- Ideal for journal is not ideal for slides
- 11 Schemes provided with stata
- Can write your own by modifying existing ones
- User-written ones also available
- `set scheme` *`scheme_name`*`, [permanently]`
- Option `scheme(`*`scheme_name`*`)`


95% CI
Fitted values
Mileage (mpg)
40
30
20
10
95% CI/Fitted values/Mileage (mpg)
2000
3000
4000
5000
Weight (lbs.)

Saving Graphs

- Save graphs in stata format with `graph save`
- Save graphs in other formats with `graph export`
- Format used defined by
  - Filename suffix
  - Option `as()` to `graph export`
- Use `help graph export` to find out formats available to you (depends on version and OS).

Naming graphs

- By default, every graph called “Graph”
- Can store files in memory by renaming:
  - Option `name()` to `graph` commands
  - `graph rename Graph` *`newname`*
- Recall with `graph display` *`name`*
- Can display multiple graphs as the same time if they have different names

Other Graph Types

`graph bar` Bar charts

`graph box` Box and whisker plots

`graph matrix` Given *n* variables, creates an *n* by *n* matrix of scatterplots, plotting every variable against every other variable.

`twoway histogram` Histograms

`twoway rcap` Given two *y*-values for each *x*-value, plots a line between the two *y*-values, with "caps" at each end. Useful for showing confidence intervals if overlaid.

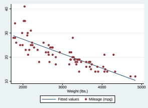
Other Graph Types

`twoway lfit[ci]` Linear regression fit to a scatter plot

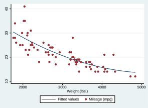
`twoway qfit[ci]` Quadratic regression fit to a scatter plot

`twoway fpfit[ci]` Fractional polynomial fit to a scatter plot

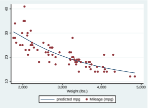
`twoway lowess` Nonparametric smoothed fit to a scatter plot



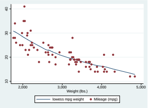
lfit



qfit

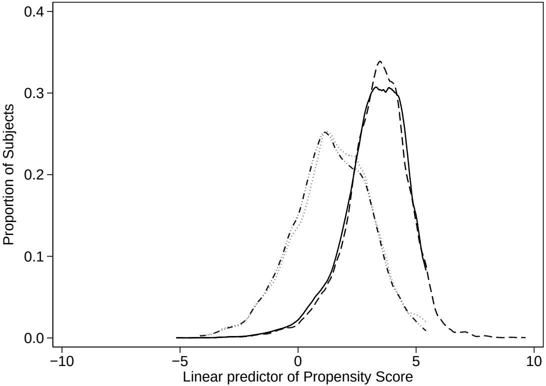


fpfit



lowess

Kernel Density



Summarizing Data

- describe
- codebook
- summarize
- tabulate

describe

- `describe [`*`varlist`*`]`
- Number of observations and variables
- For each variable
    - Name
    - Type
    - Format
    - Labels

codebook

- More detail on each variable:
  - All variables: type, range, unique values, missing values, units
  - Continuous vars: mean, SD, percentiles
  - Categorical vars: frequency table / sample values

summarize

`summarize [varlist]`

- Gives mean, SD, min, max, non-missing values
- Option `detail` gives fuller summary

```
summarize  price mpg headroom trunk

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       price |        74    6165.257    2949.496       3291      15906
         mpg |        74     21.2973    5.785503         12         41
    headroom |        74    2.993243    .8459948        1.5          5
       trunk |        74    13.75676    4.277404          5         23
```

tabulate

- `tabulate variable` gives a frequency table
- `tabulate var1 var2` give a cross-tabulation
- Option `ro` and `co` give row and column percentages respectively
- Option `chi2` gives χ^2-test.

More Stata Syntax

[by *varlist*]: command *varlist* [if *expression*][, options]

- `by` repeats an analysis for each subgroup
- `if` selects a single subgroup to analyse.

Logical Operators

Operator	Meaning
&	and
|	or
==	equal
∼=, !=	not equal
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal

Missing Values

- Missing values are bigger than any “real” value
- Using variables in logical expressions is dangerous if missing values exist
- E.g. (price $>$ 15000) is true if price is missing.
- `gen hi_price = price > 15000 if price < .`
- Be very careful when categorising continuous variables.

The `by` *`varlist`* clause

- Produces results for each subgroup defined by *`varlist`* separately
- Data needs to be sorted for `by` to work
- Command `bysort` will do it for you
- Can replace a lot of `if` clauses
- Complex expression can only be used with `if`
- Does not work with every command

Subscripting

- Square brackets ([]) after a variable name used pick out an observation by its number
- `weight[7]` means the weight of the seventh observation
- `_n` means the number of the current observation
- `_N` means the number of observations in the data (or `by` group)

Lagged Variables

- `varname[_n - 1]` means the value of the variable `varname` in the previous observation
- `bysort idno (fupno): replace haq = haq[_n - 1] if haq == .`
- `bysort idno (fupno): gen diff = haq - haq[_n-1]`

Looping

```
foreach macname in list {
    list of stata commands
}
```

- Opening { must be on first line
- Command(s) must start on next line
- Final } must have its own line

Other forms of `foreach`

- `foreach var of varlist ...`
- `foreach var of newlist ...`
- `foreach num of numlist ...`

Examples of `foreach`

```
foreach visit in 1 2 {
    summarize bp if when == ‘visit’
}  label define yesno 0 "No" 1 "Yes"
```

```
foreach x of varlist *_pain {
    label values ‘x’ yesno
}
```

Reshaping Data

- Long to wide: very easy
- Wide to long: slightly trickier
- Long form more efficient for storage: only need space for followups that exist
- Long form also normally best for analysis

Long Form

ID	Gender	Anniversary	Score
900108	1	1	7
900108	1	2	15
900108	1	5	19
900113	2	1	0
900113	2	2	18
900114	1	1	0
900114	1	2	0

Long to wide

Need to specify:

- Unique identifier which shows which observations belong together: `id`
- Which “repeat” a given observation corresponds to: `anniversary`
- Which variables change between visits: `score`

```
reshape wide score, i(id) j(anniversary)
```

Wide Form

ID	Gender	Score1	Score2	Score5
900108	1	7	15	19
900113	2	0	18	.
900114	1	0	0	.

Wide to long

Need to specify:

- Unique identifier which shows which observations belong together: `id`
- The name of a new variable to contain “repeat” info: `anniversary`
- Which variables are in wide form: `score`
- If suffixes are strings, need to use the `string` option.

```
reshape long score, i(id) j(anniversary)
```

Other Useful Commands

`display`	Make things appear in the results window. Can be used as a calculator
`expand`	Produce multiple copies of each observation
`cmdlog`	Make a do-file of all the commands you are entering.

Expand

Exposed	Cases	Controls
No	20	40
Yes	30	10

exposed	case	frequency
0	0	40
0	1	20
1	0	10
1	1	30