Statistics/Data Analysis

Charts for comparing results between many categories

Ulrich Kohler, WZB
kohler@wzb.eu

Presentation for the 6th German Stata Users' Group Meeting on Friday June 27th 2008 at WZB

Contents

An example

This presentation is about comparing statistics between many groups using graphs. Here is an example:

run example0.do

Notes
- **o** Takes some time to run (loads many programs)
- **o** Looks impressive
- **o** Shows many stats (using dots, arrows, bars, vertical lines)
- **o** Shows stats for many groups (surveys, countries, sampling method)
- **o** Concepts: overlays, resultssets, categorical axis, juxtaposition

Charts with overlays

Limits of graph bar

Charts for comparing results between groups with many categories can be produced from individual data out of the box:

run example1a.do

```
use ESS, clear
graph bar (mean) stflife, over(cntry, sort((mean) stflife))
```

However, producing such plots with **graph bar** has some limitations, and sometimes they arise pretty soon:

run example1b.do

```
summarize stflife, meanonly
graph bar (mean) stflife
  , over(cntry, sort((mean) stflife)) yline(‘=r(mean)’)
```

Charts with overlays, continued

Introducing graph twoway bar

graph bar can be reproduced with **graph twoway bar**. Therefore revert to the "resultsset" approach

run example2.do

```
collapse (mean) stflife, by(cntry)

encode cntry, gen(ctrnum)
sum stflife, meanonly
gen meanstflife = r(mean)

graph twoway
  || bar stflife ctrnum
  || line meanstflife ctrnum
```

Charts with overlays, continued

The order problem

To clone the figure from **graph bar** with **graph twoway** you must solve the order problem.

run example3.do

```
egen axis = axis(stflife), label(ctrnum)

graph twoway
  || bar stflife axis, barwidth(0.8)
  || line meanstflife axis, lcolor(black)
  || , xlab(1/25, valuelabels) xtitle("") legend(off)
```

Charts with overlays, continued

Twoway bar is powerful

Once you moved to **twoway bar** you can do a lot of things

run example4.do

```
format stflife %2.1f

gen xnordic = axis if inlist(cntry,"NO","FI","IS","SE","DK")
gen ynordic = stflife if !mi(xnordic)
gen ystart = 8.7                <- Trial and Error
gen xstart = axis[_N-4]         <- Trial and Error
```

```
graph twoway
  || bar stflife axis, barwidth(0.8)
  || line meanstflife axis, lcolor(black)
  || scatter stflife axis
     , ms(i) mlab(stflife) mlabpos(6) mlabcolor(white)
  || scatteri ‘=meanstflife[1]’ 1 "Average", mlabpos(1) ms(i)
  || pcarrow ystart xstart ynordic xnordic
  || scatteri ‘=ystart[1]’ ‘=xstart[1]’ "Nordic Exceptionalism?"
    , ms(i) mlabpos(12)
  || , xlab(1/25, valuelabels) xtitle("") legend(off)
```

More on resultssets

Produce a resultsset with countries’ average life satisfaction after controlling for individual characteristics.

run example5.do

```
use ESS, clear
gen men = gndr==1 if !mi(gndr)
gen unemp = uempla==1 if !mi(uempla)
egen hhinc = xtile(hinctnt), p(20(20)80) by(cntry)
tab hhinc, gen(hhinc)

levelsof cntr, local(K)
foreach k of local K {
        gen byte ‘k’ = cntry == "‘k’" if !mi(cntry)
}

reg stflife men hhinc2-hhinc5 AT-UA, hascons

parmest, fast
drop if parm == "men" | strpos(parm,"hhinc") | parm == "_cons"
```

On resultssets, continued

Once you made a resultsset, using -twoway- is easy:

run example6.do

```
egen axis = axis(estimate), label(parm)
graph twoway
  || bar estimate axis, barwidth(0.8)
  || rcap min95 max95  axis, lcolor(black)
  || , xlab(1/25, valuelabels) xtitle("") legend(off)
```

Hence: Producing graphs is often just a data management problem. For details on resultssets consider Roger Newson’s site at http://www.imperial.ac.uk/nhli/r.newson/usergp/

Designing the categorical axis

Long labels

Comparing results between many groups often requires long category labels. Better to use the vertical axis for this:

run example7.do

```
egen ctrname = iso3166(parm), origin(codes)
egen axiswide = axis(estimate), label(ctrname) reverse
```

```
graph twoway
  || scatter axiswide estimate, mcolor(black)
  || rcap min95 max95 axiswide, lcolor(black) horizontal
  || , ylab(1/25, valuelabels angle(0) grid gstyle(dot))
     ytitle("") legend(off)
```

Aside: Cleveland (1994) recommends dot charts instead of bar charts.

Designing the categorical axis, continued

Grouping

It is often sensible to group the categorical information into broader categories.

run example8.do

```
gen group:gr=2 if inlist(parm,"UA","HU","SK","EE","PL","CZ","SI")
replace group = 3 if inlist(parm,"TR","NO","CH","IS")
replace group = 1 if mi(group)
lab def gr 1 "EU-15" 2 "Post socialist"  3 "Other"

egen axisgroup = axis(group estimate), gap label(ctrname) reverse
egen groupmeans = mean(estimate), by(group)

levelsof axisgroup, local(ylab)
graph twoway
  || line axisgroup groupmeans if group==1, lcolor(gs8)
  || line axisgroup groupmeans if group==2, lcolor(gs8)
  || line axisgroup groupmeans if group==3, lcolor(gs8)
  || scatter axisgroup estimate, mcolor(black) ms(O)
  || rcap min95 max95 axisgroup, lcolor(black) horizontal
  || , ylab(`ylab', valuelabels angle(0) grid gstyle(dot))
     ytitle("") legend(off)
```

Juxtaposition

A problem with by()

Juxtaposition can be achieved with graph option -by()-. However -by()- does not work with graphs of different sizes.

run example9.do

```
graph twoway
  || line axisgroup groupmeans, lcolor(gs8)
  || scatter axisgroup estimate, mcolor(black) ms(O)
  || rcap min95 max95 axisgroup, lcolor(black) horizontal
  || , ylab(, valuelabel angle(0) grid) ytitle("")
     by(group, col(1) yrescale note("")legend(off) )
```

Juxtaposition, continued

Combine with force

Use the "forced size options" and -graph combine- to tune the individual graphs' sizes.

run example10.do

```
local opt `"yti("") xtic(4(1)8) nodraw leg(off) xsc(r(3.7 8.3))"'
local fsizes "50 27 23" // Trial and Error

forv i = 1/3 {
        local xlab = cond(`i'<3,"xlab(none)","xlab(4(1)8)")
        local marg = cond(`i'<3,"","graphregion(margin(l=10))")
```

```
        levelsof axisgroup if group == ‘i’, local(ylab)
        graph twoway
          || line axisgroup groupmeans, lcolor(gs8)
          || scatter axisgroup estimate, mcolor(black) ms(O)
          || rcap min95 max95 axisgroup, lcolor(black) horizontal
          || if group == ‘i’ , ‘opt’ ‘xlab’ ‘marg’
         ylab(‘ylab’, valuelabel angle(0) grid)
          title(‘:label (group) ‘i’’, bexpand pos(12) box)
          fysize(‘:word ‘i’ of ‘fsizes’’) name(g‘i’, replace)
}

graph combine g1 g2 g3, cols(1) ysize(6) imargin(t=0)
```

Additional hints

Rules for designing statistical graphs are given by

o Cleveland, W.S. (1994), The elements of graphing data. Summit, NJ: Hobart Press.

On a related topic:

o Cox, N.J. (in press), Speaking Stata: Between tables and graphs. Stata Journal 8 (2) pp X–XX