Nice log (and log-like) scaled axes

Mark Chatfield

11th Oceania Stata Conference
1 February 2024

Structure

Part 1. Nice log-scaled axes

Q. How to create graph commands that nicely label a log-scaled axis?

Part 2. Nice log-like-scaled axes

Q. How to produce a nice log-like-scaled axis showing 0 and/or ∞?

I hope this presentation

- helps you
- nudges StataCorp to help us get nice log-scaled axes *more easily*

Part 1. Nice log-scaled axes

Q. How to create graph commands that nicely label a log-scaled axis?

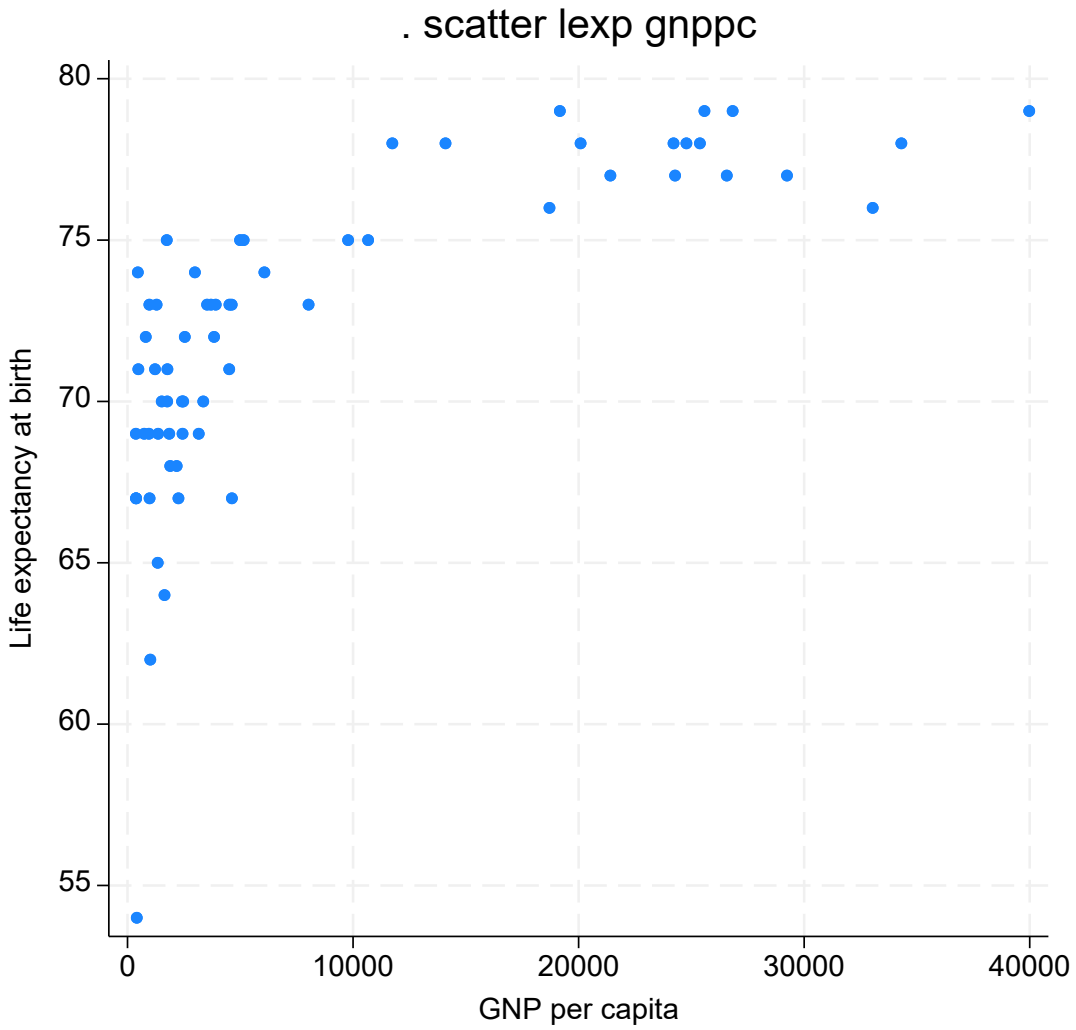
Two approaches:

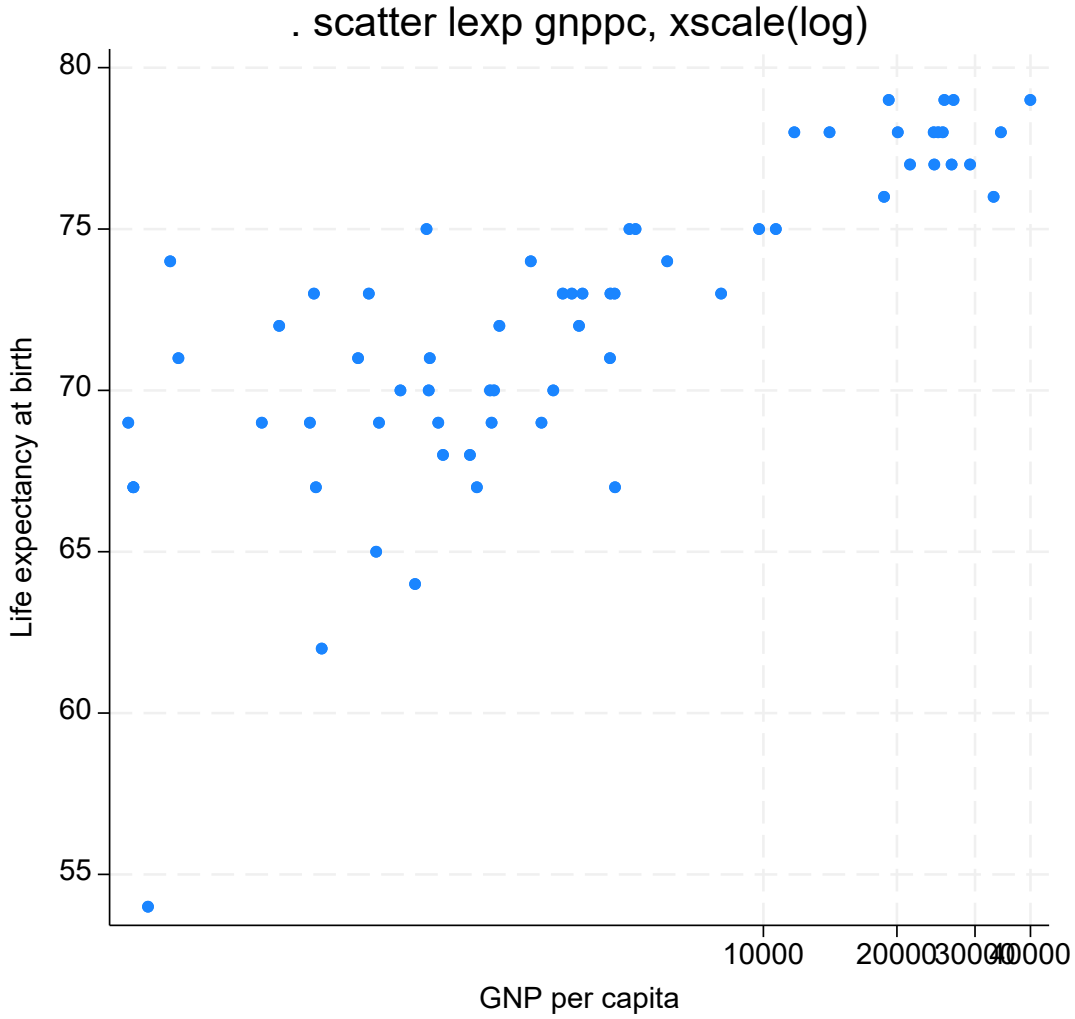
- Plot data specifying the xscale(log) option
- Plot log-transformed data and do tricks

I'll present a simple program (-decidelabels-) to help with either approach

- I've used similar code in a couple of graph commands I've released

xscale(log) approach to produce a log-scaled axis





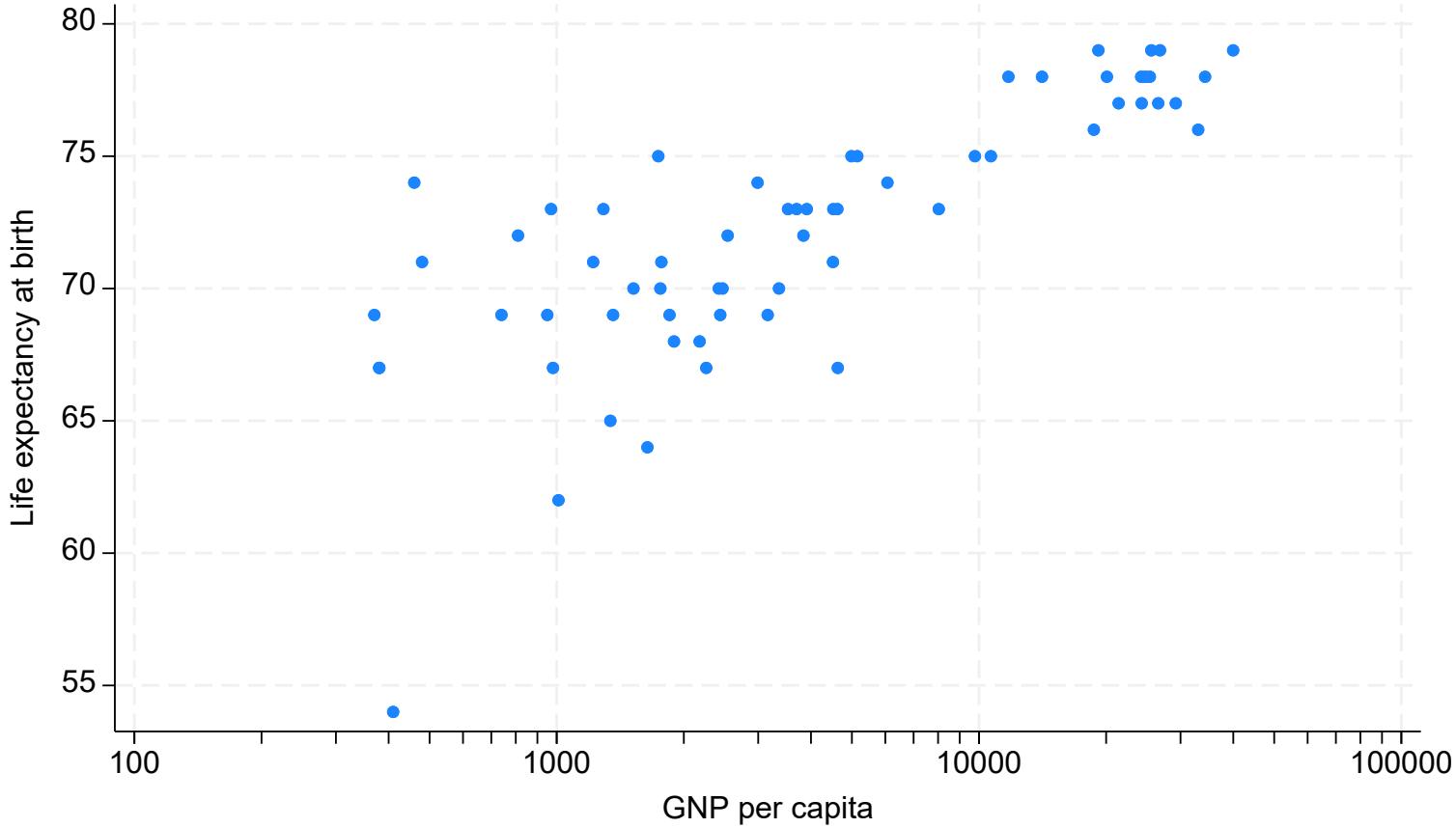
Source: these graphs appear in [G-3] axis_scale_options

"Additive" labels still! (except 0)

A user will often need to improve the labelling

For example,

```
. scatter lexp gnppc, xscale(log) xlabel(100 1000 10000 100000) ///
  xtick(100(100)1000 1000(1000)10000 10000(10000)100000)
```



Some options for “multiplicative” labels

```
. summarize gnppc

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
       gnppc |         63    8674.857    10634.68        370      39980
```

```
. niceloglabels 370 39980, local(labels) style(1)
1000 10000
```

← 10^k

```
. niceloglabels 370 39980, local(labels) style(2)
512 1024 2048 4096 8192 16384 32768
```

← 2^k

Multiplicative-like labels

```
. niceloglabels 370 39980, local(labels) style(13)
1000 3000 10000 30000
```

← ×3 and $\times 3\frac{1}{3}$

```
. niceloglabels 370 39980, local(labels) style(125)
500 1000 2000 5000 10000 20000
```

← ×2 and ×2.5

Cox, N. J. (2018). Speaking Stata: Logarithmic binning and labelling. *Stata Journal*, 18: 262–286.

Cox, N. J. (2020). Software update for niceloglabels. *Stata Journal*, 20: 1028–1030.

How to decide the “best” labels (in general)?

Useful if you are creating a graph command and you wish to nicely label a log-scaled axis

A simple algorithm for automating the choice of labels on a log-scaled axis:

1) Use powers of 10 such as ..., 100, 1000, 10000, ... (if ≥3 such labels between min and max)
2) Otherwise labels such as ..., 100, 200, 500, 1000, ... (if ≥3 such labels between min and max)
3) Otherwise let Stata use its additive labels

-decidelabels- facilitates this algorithm…

Algorithm aligns fairly well with advice of Cox (2018).

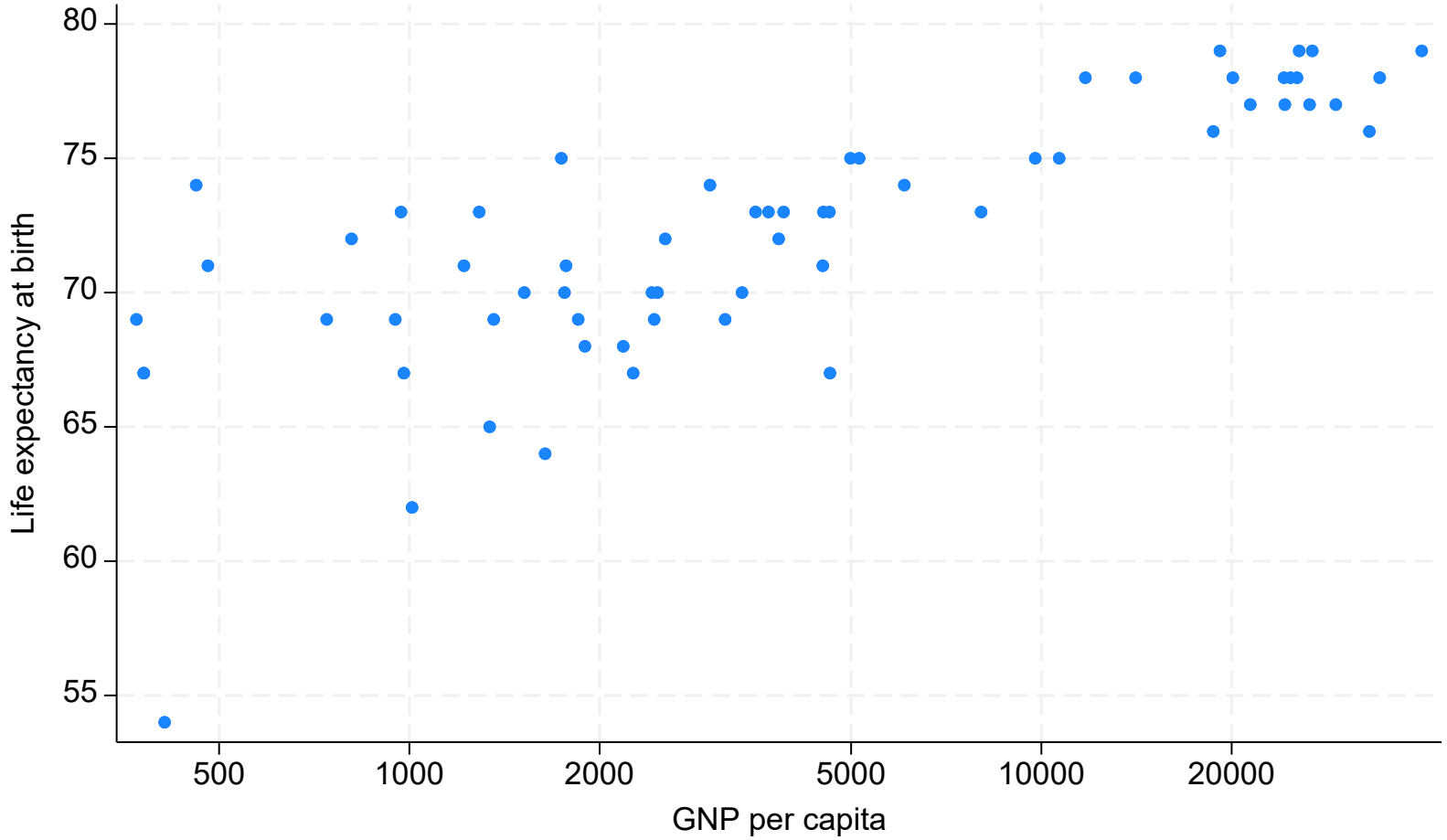
-decidelabels-

```
program define decidelabels, rclass
        args min max

        niceloglabels `min' `max', local(labels125) style(125)
        local countlabels125 = wordcount("`labels125'")

        if `countlabels125' >= 3 {
                niceloglabels `min' `max', local(labels1) style(1)
                local countlabels1 = wordcount("`labels1'")
                if `countlabels1' >= 3 local bestlabels "`labels1'"
                else local bestlabels "`labels125'"
        }

        return local bestlabels "`bestlabels'"
end
```

Program written specifically for this presentation. Feel free to copy, modify, and use it however you would like.

-decidelabels- in action

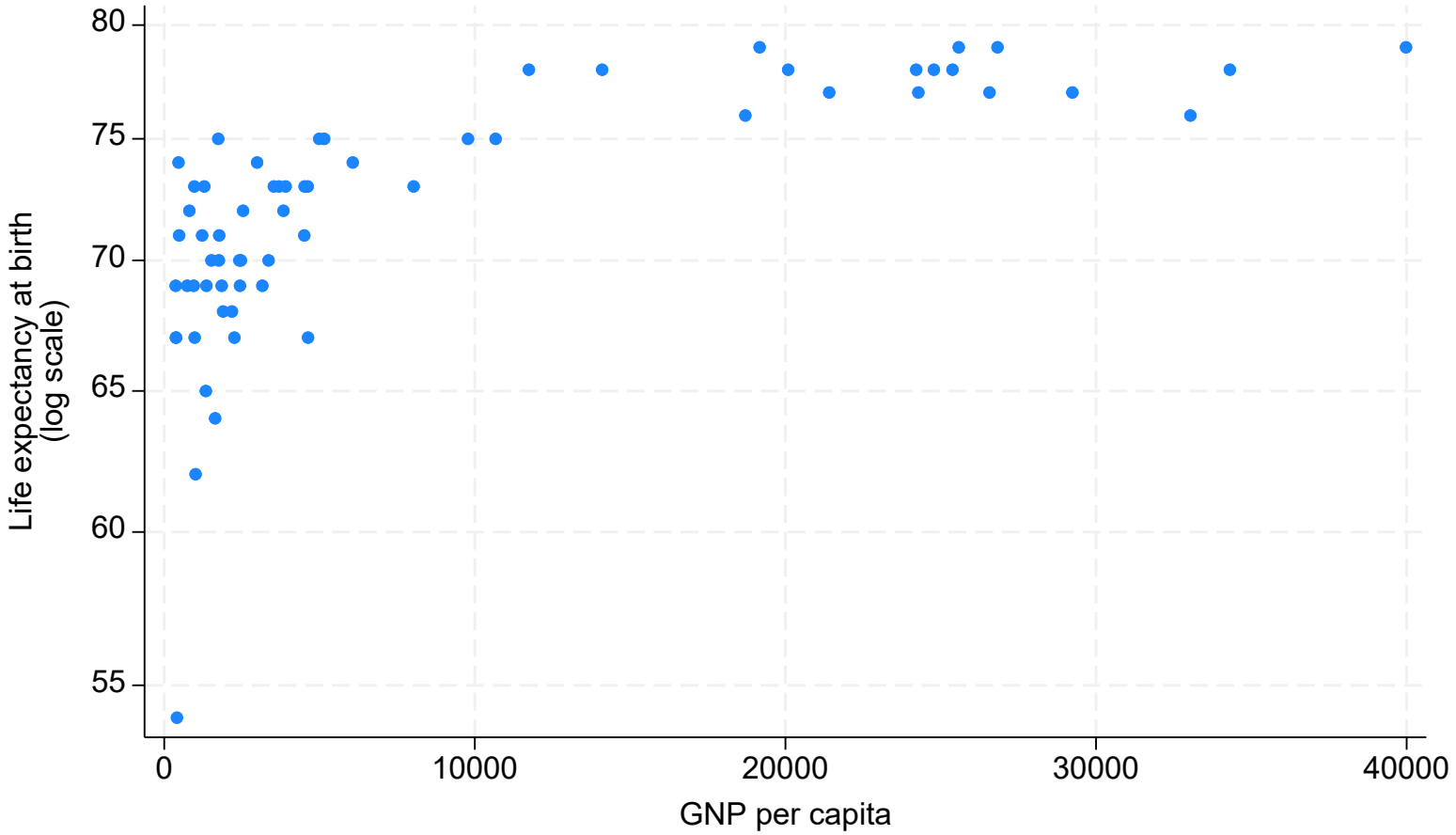
```
. summarize gnppc
. decidelabels `r(min)' `r(max)'
. scatter lexp gnppc, xscale(log) xlabel(`r(bestlabels)')
```



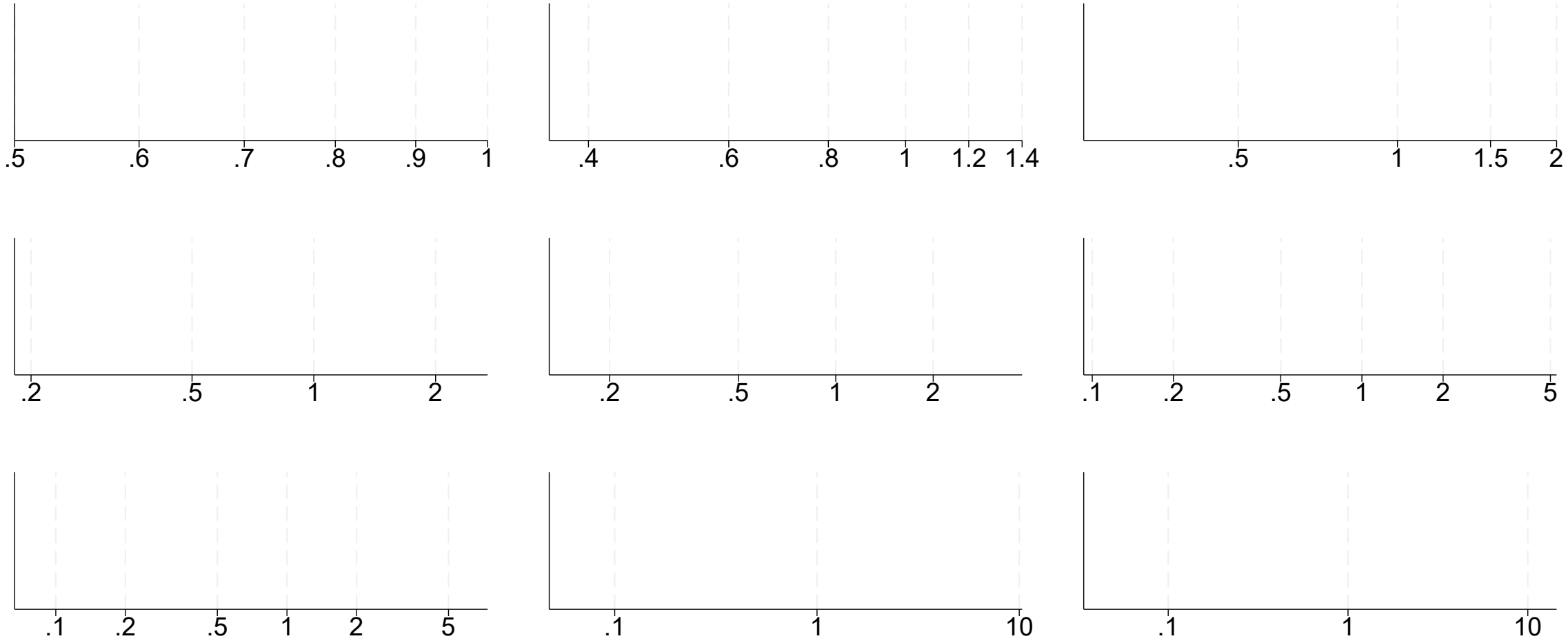
-decidelabels- in action when max/min is small

```
. summarize lexp
. decidelabels `r(min)' `r(max)'
. scatter lexp gnppc, yscale(log) ylabel(`r(bestlabels)') ytitle("(log scale)", suffix)
```

Here,
r(bestlabels) = ""

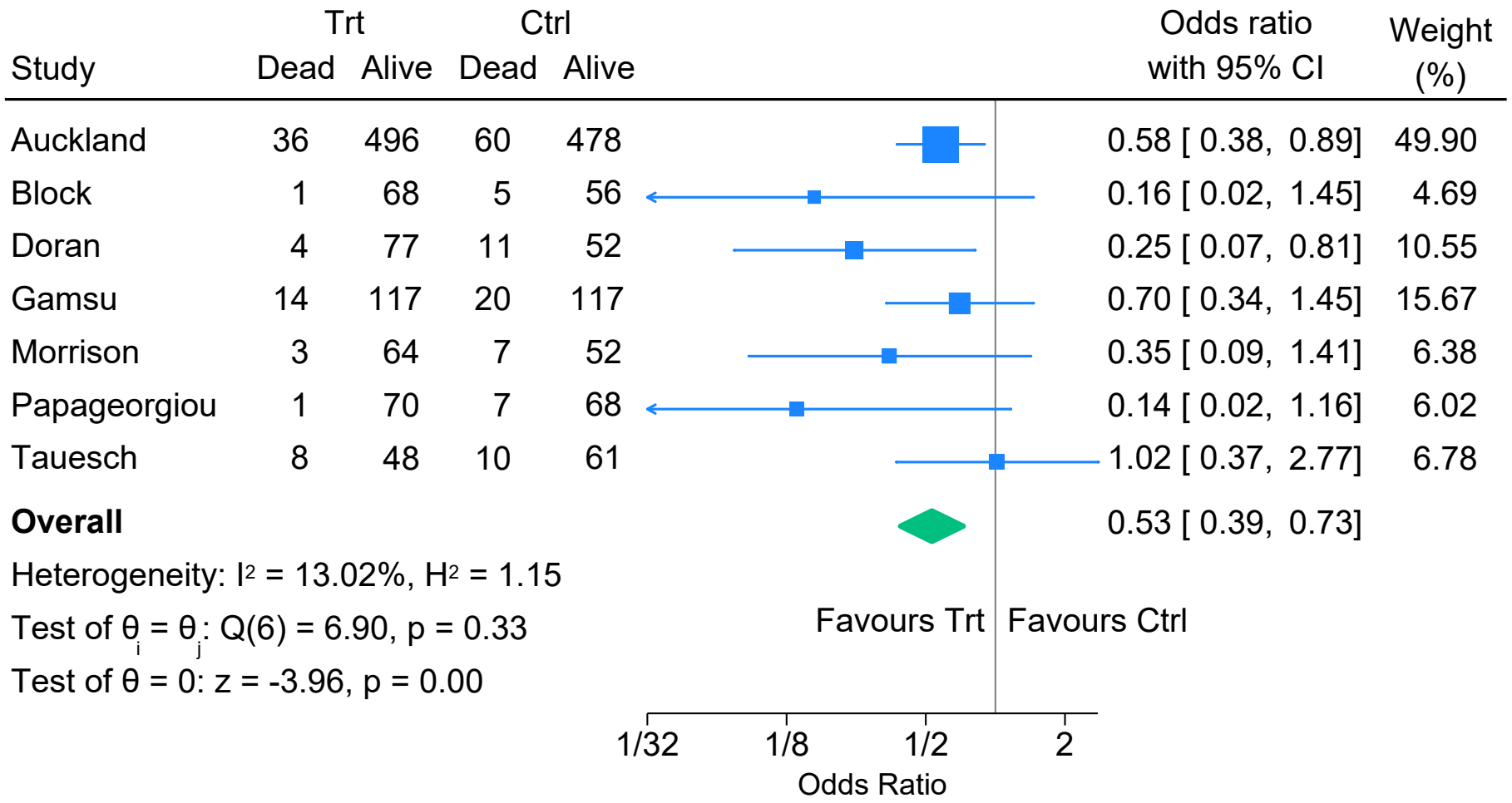


-decidelabels- in action around 1



The only(?) inbuilt Stata command with multiplicative labels

```
. meta forestplot, eform crop(`=1/32' 4) ...
```



Algorithm used by -meta forestplot, eform-

1) Label using a selection of powers of 2 (if ≥2 such labels). Have at most ~5 labels (by default). Achieve that by skipping some labels. And label “1/8” rather than .125 etc.

2) Otherwise simply label min and max

← found in tickset_g.class

```
program .set_ticks_pow2
        args min max nticks

        local ln2min = ceil(log(`min')/log(2))
        local ln2max = floor(log(`max')/log(2))
        ...
        local newmin = 2^(`ln2min')
        local newmax = 2^(`ln2max')

        local range = `ln2max' - `ln2min'
        local skip = floor(`range'/(`nticks'-1))
        if `skip' < 1  local skip 1
        local 2skip = 2^(`skip')      //   <- multiplier
        ...
end
```

Aside. -niceloglabels- can label .125 as:

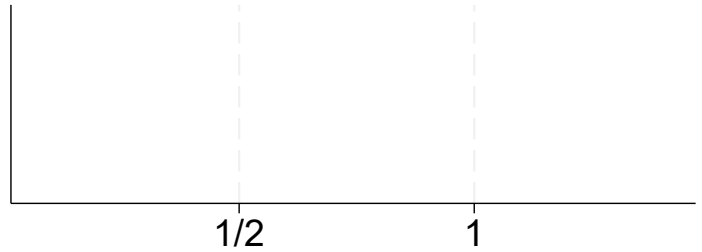
1/8

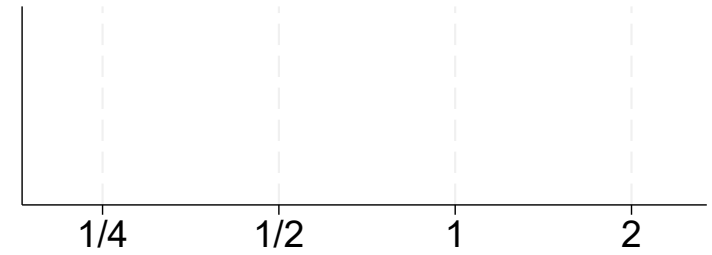
2^{-3}

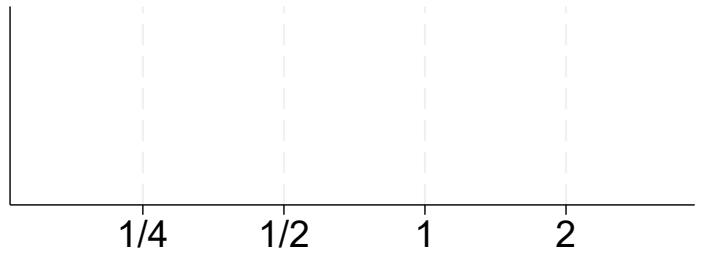
$(1/2)^3$

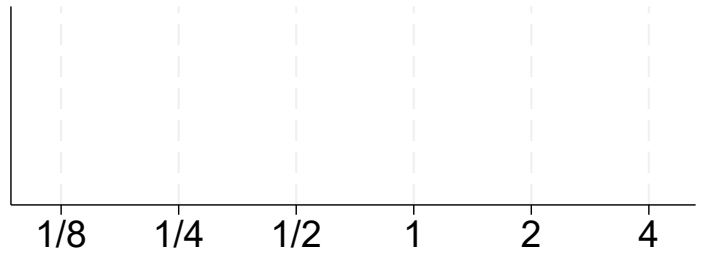
0.125

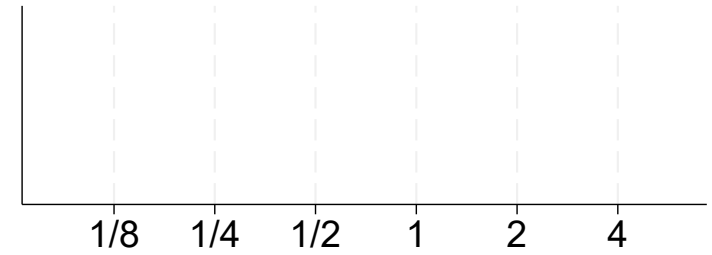
Labelling with -meta forestplot, eform xlabel(#8)-

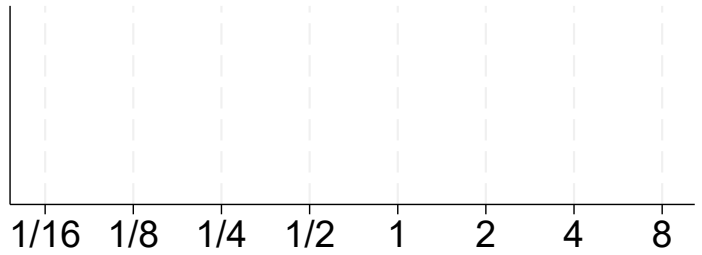


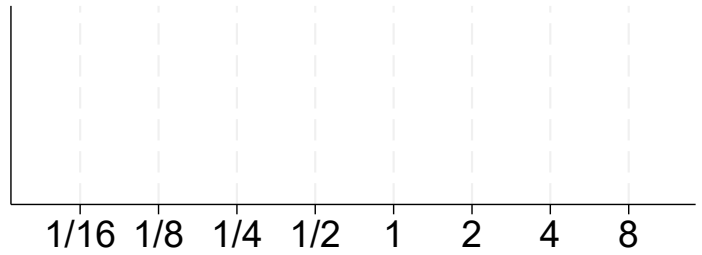










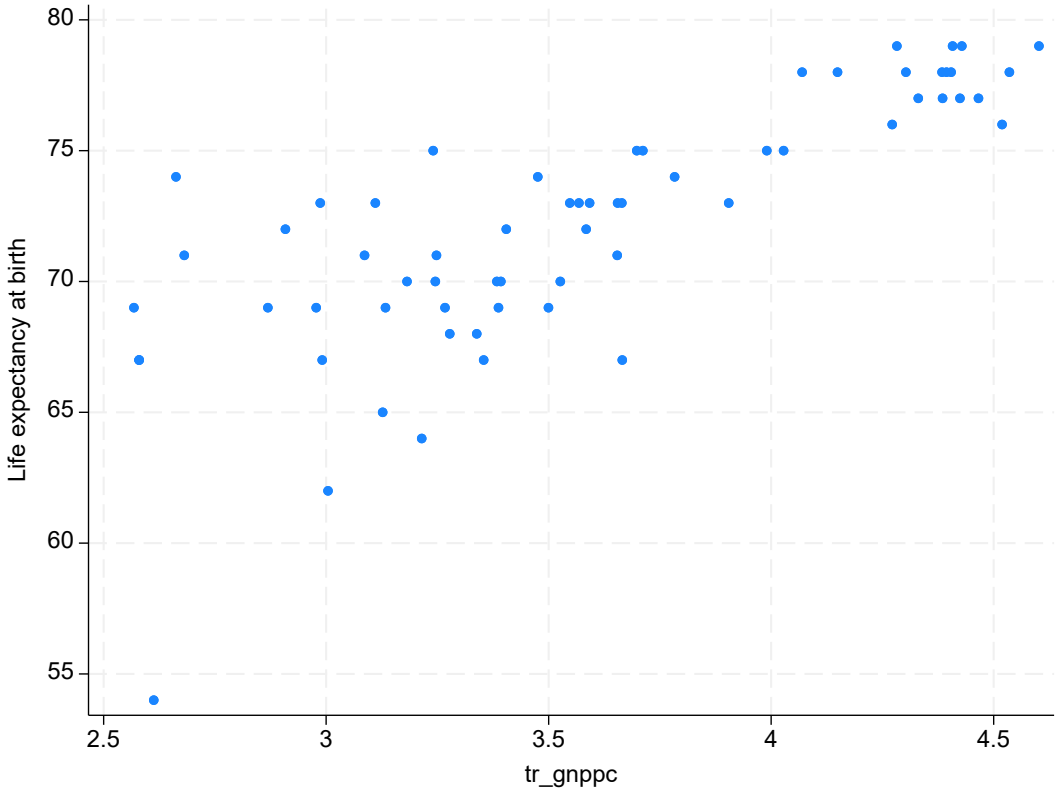


Alternative approach to produce a log-scaled axis

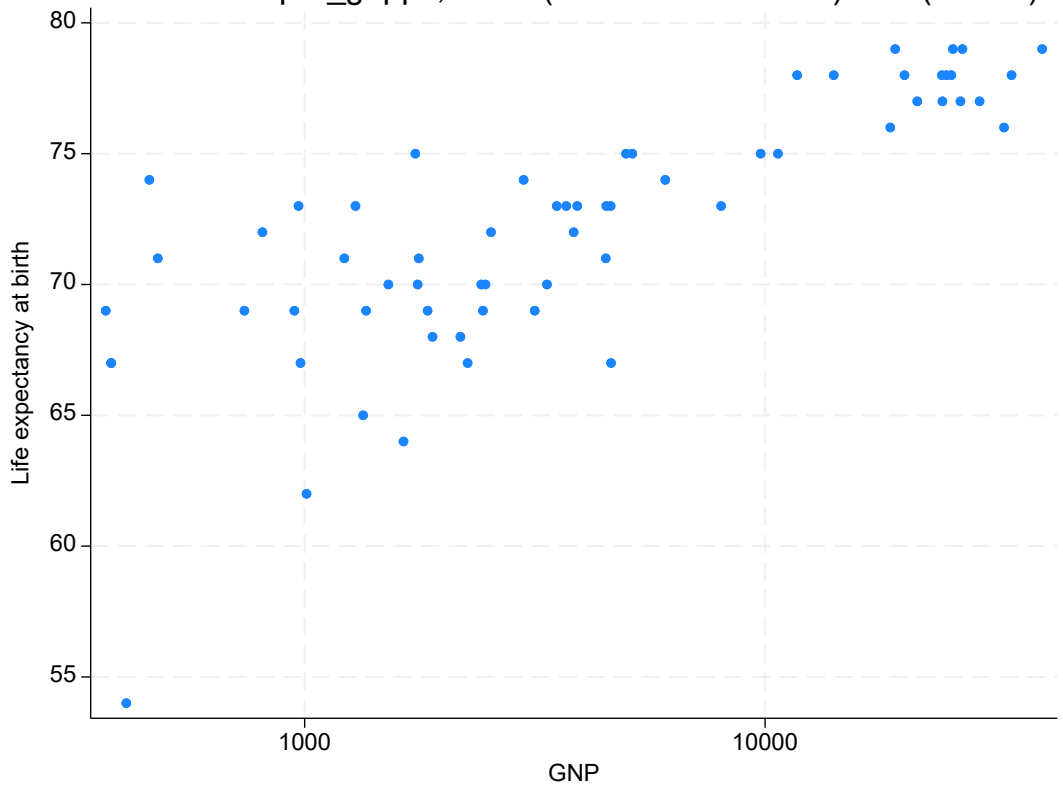
Log transform the data, then plot the transformed data, but 1) label axis with original scale values and 2) change the title on axis

```
. generate tr_gnppc = log10(gnppc)
```

. scatter lexp tr_gnppc



. scatter lexp tr_gnppc, xlabel(3 "1000" 4 "10000") xtitle("GNP")



Automating this approach is harder

1. Decide what labels you want
e.g. 500 1000 2000 5000 10000 20000

2. Transform these labels
e.g. `=log10(500)' `=log10(1000)' ...
2.69 3

3. Combine

Nick Cox's -mylabels- command can do this:

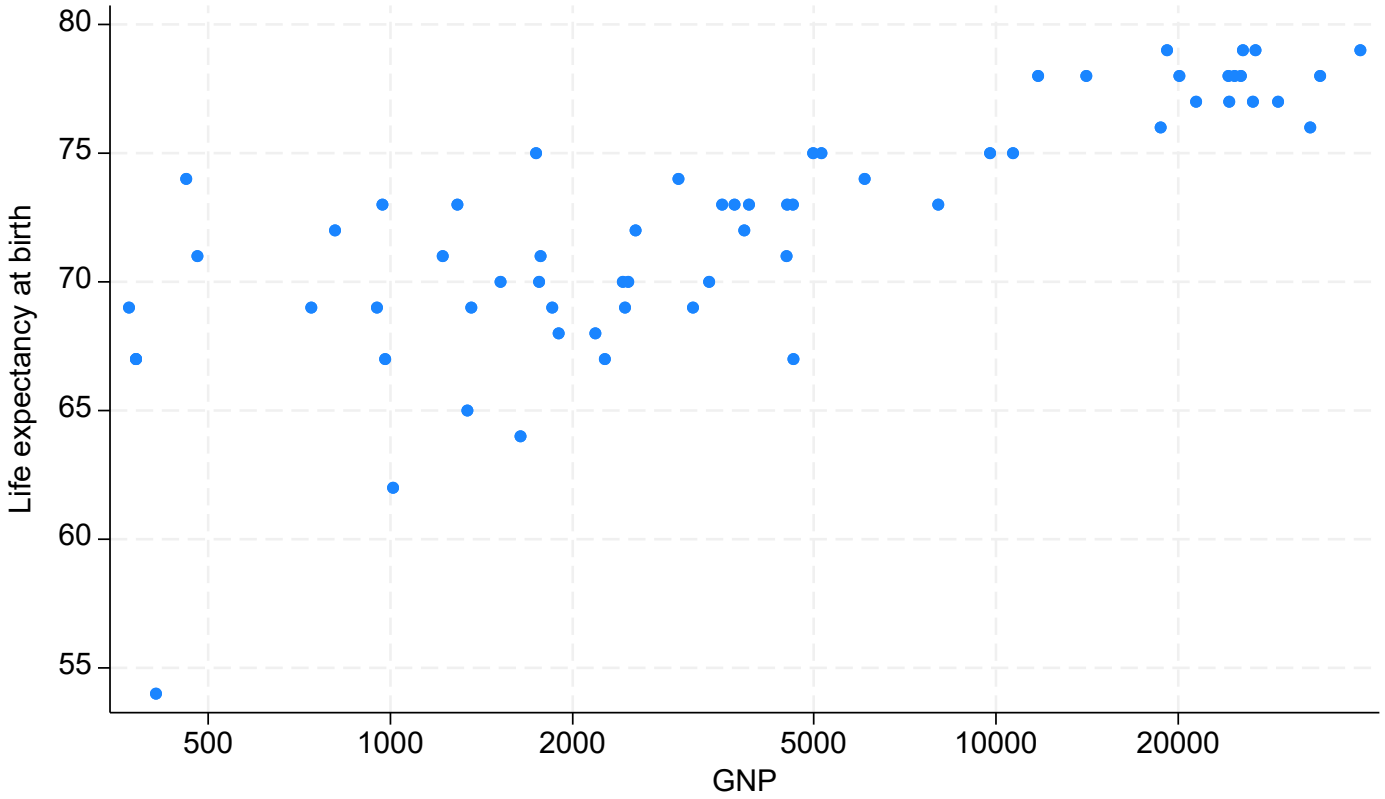
```
. mylabels 500 1000 2000 5000 10000 20000, myscale(log10(@)) local(tricklabels)
2.698970004336018 "500" 3 "1000" 3.301029995663981 "2000" 3.698970004336019 "5000" 4
"10000" 4.30102999566398 "20000"
```

Cox, N. J. (2022). Speaking Stata: Automating axis labels: Nice numbers and transformed scales. *Stata Journal*, 22(4): 975-995.

Cox, N. J. (2024). Software update for mylabels. *Stata Journal*, 24(1): To appear.

Towards automating this approach

```
. generate tr_gnppc = log10(gnppc)
. summarize gnppc
. decidelabels `=r(min)' `r(max)'
. mylabels `=r(bestlabels)', myscale(log10(@)) local(tricklabels)
. scatter lexp tr_gnppc, xlabel(`tricklabels') xtitle("GNP")
```



-decidelabels- needs to be expanded

Why? Need to specify list of “additive labels”

e.g. 55 (5) 80 or 0.2 (0.2) 1

```
. _natscale 54 79  6

. return list

scalars:
                  r(n) =  6
                r(min) =  55
                r(max) =  80
              r(delta) =  5
```

```
. _natscale 0.11 0.8  6

. return list

scalars:
                  r(n) =  5
                r(min) =  0
                r(max) =  .8
              r(delta) =  .2
```

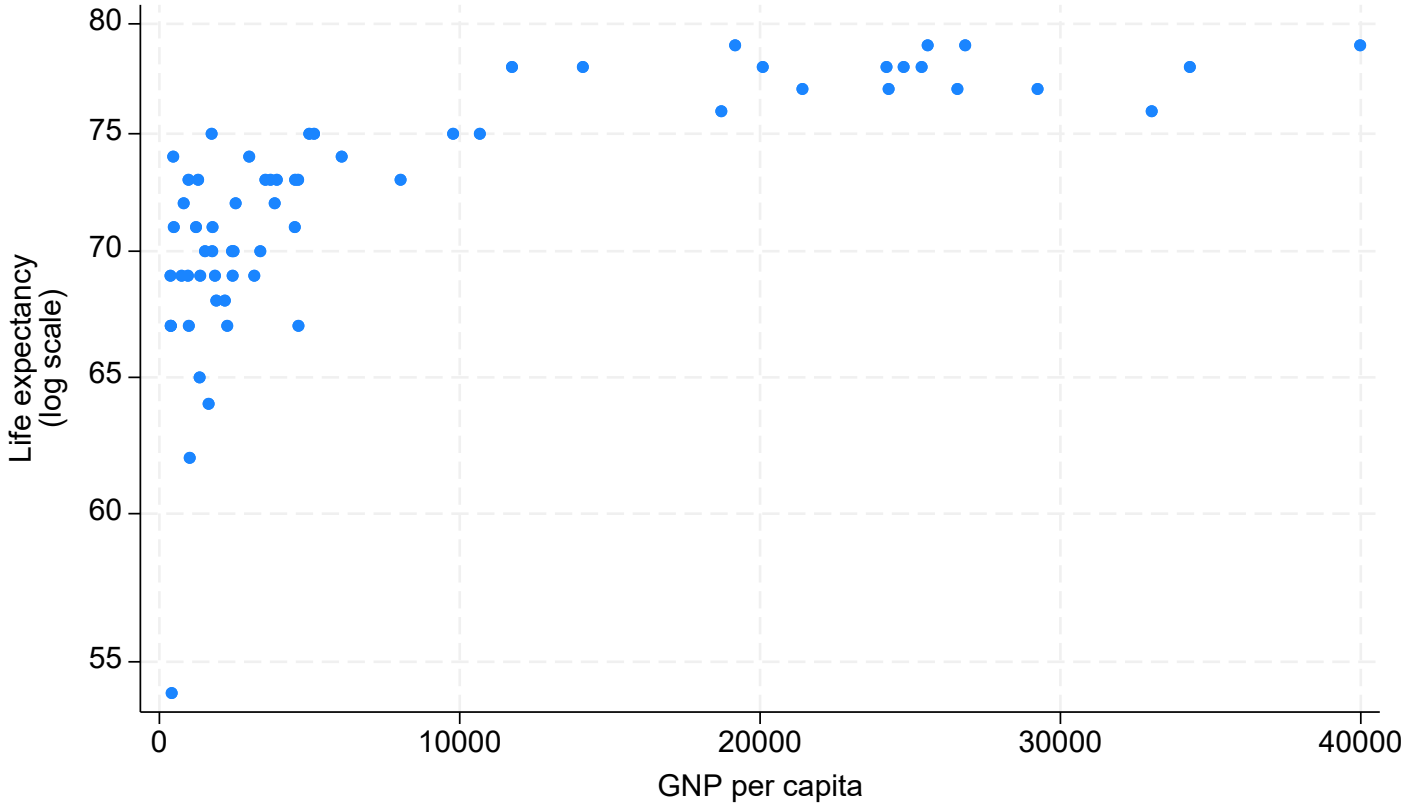
Extra code for -decidelabels-

```
program define decidelabels, rclass
        args min max

        niceloglabels `min' `max', local(labels125) style(125)
        local countlabels125 = wordcount("`labels125'")

        if `countlabels125' >= 3 {
                niceloglabels `min' `max', local(labels1) style(1)
                local countlabels1 = wordcount("`labels1'")
                if `countlabels1' >= 3 local bestlabels "`labels1'"
                else local bestlabels "`labels125'"
        }
        else {
                _natscale `min' `max' 6
                if r(min) == 0  local bestlabels "`=r(delta)' (`=r(delta)') `=r(max)'"
                else  local bestlabels "`=r(min)' (`=r(delta)') `=r(max)'"
        }

        return local bestlabels "`bestlabels'"
end
```

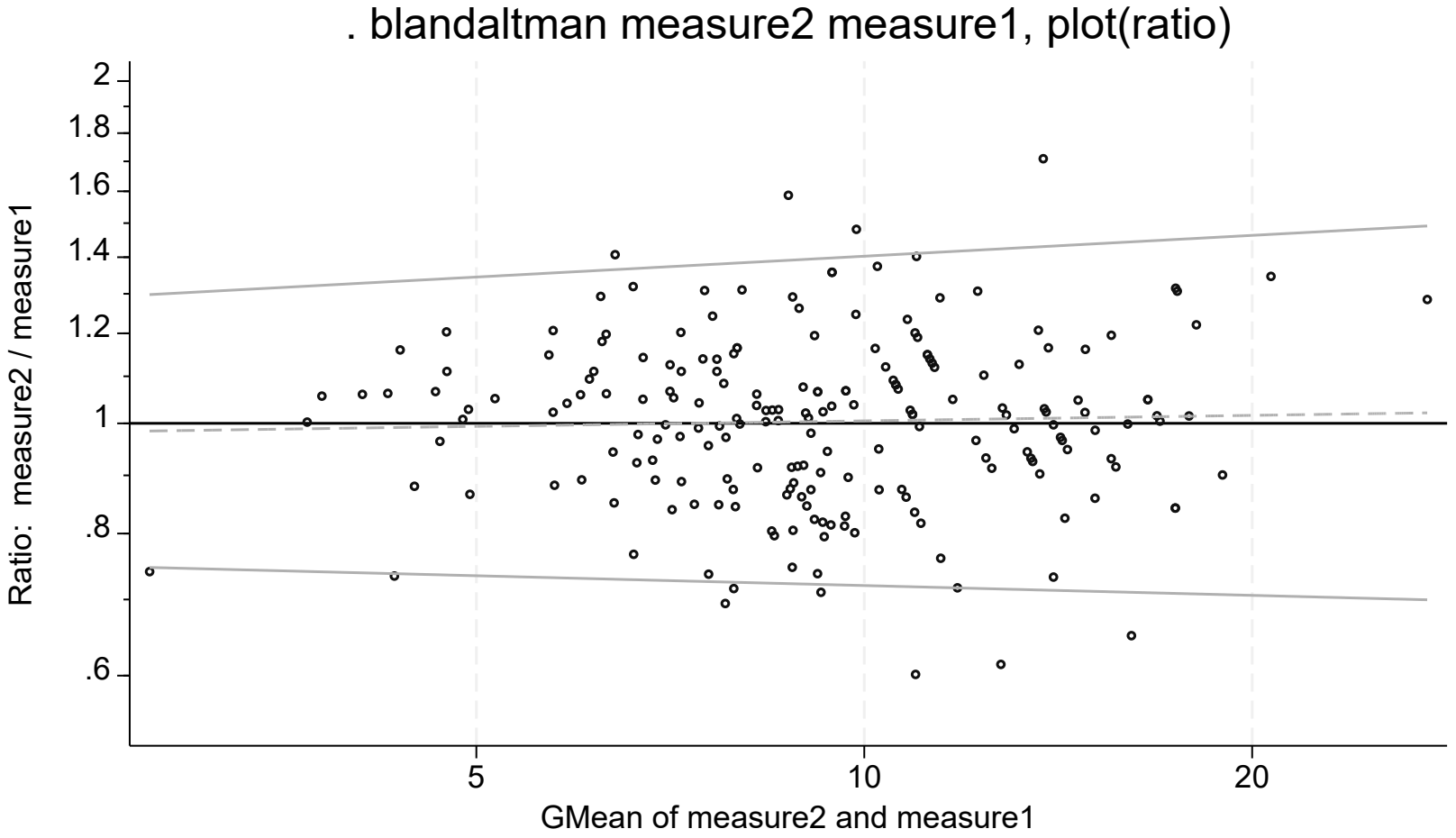
Alternative approach in action when max/min is small

```
. generate tr_lexp = log10(lexp)
. summarize lexp
. decidelabels `r(min)' `r(max)'
. mylabels `=r(bestlabels)', myscale(log10(@)) local(tricklabels)
. scatter tr_lexp gnppc, ylabel(`tricklabels') ytitle("Life expectancy" "(log scale)")
```

Here,
r(bestlabels) = "55 (5) 80"



A graph command internally using xscale(log)

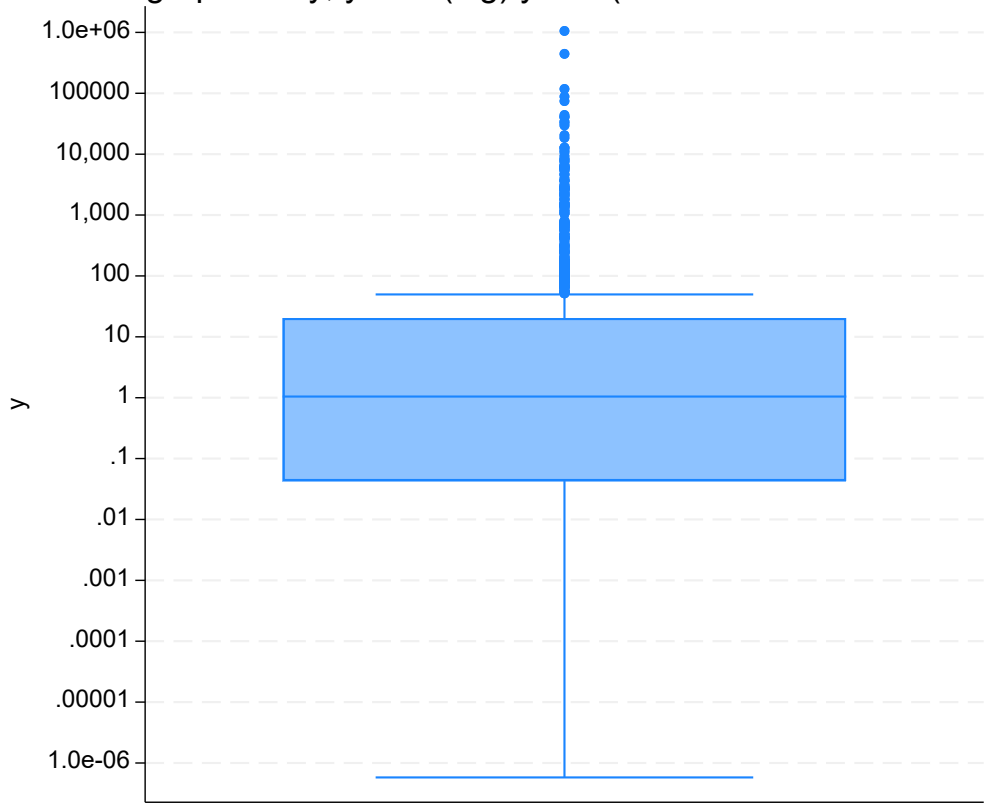


. search blandaltman

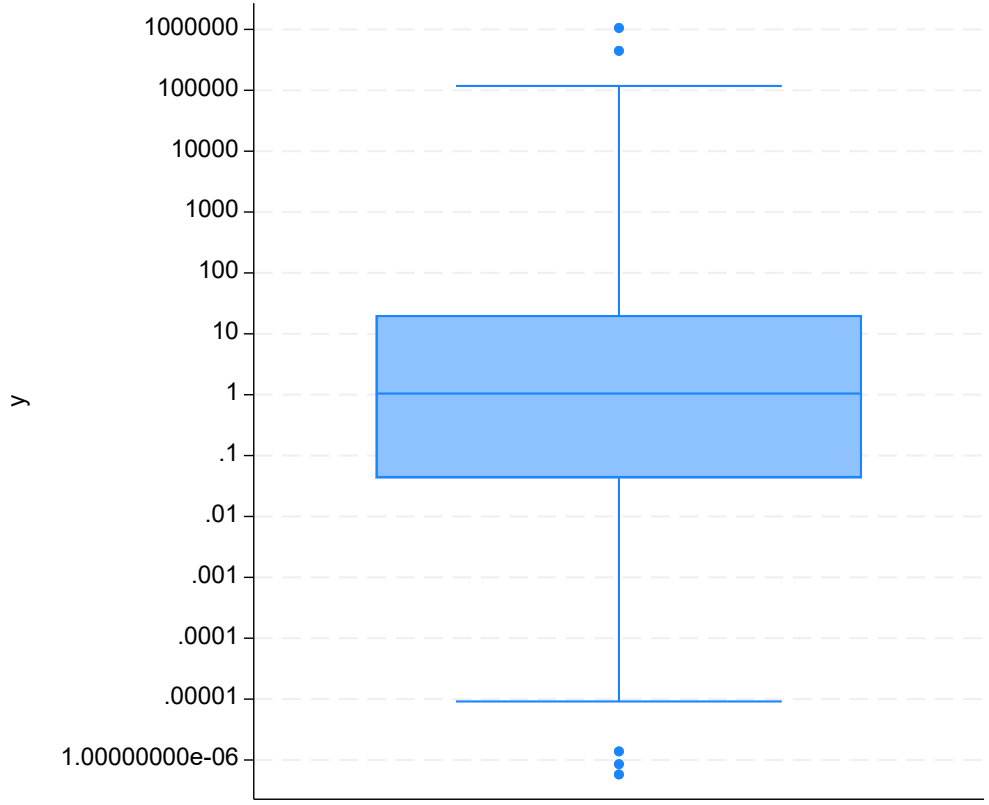
Chatfield M, Cole T, de Vet H, Marquart-Wilson L & Farewell D. blandaltman: a command to create variants of Bland-Altman plots. *Stata Journal* 2023; 23: 851-874.

A graph command internally log-transforming the data

. graph box y, yscale(log) ylabel(.000001 .00001 .0001 ...)



. box_logscale y



For more, . search box_logscale

https://www.stata.com/support/faqs/graphics/box-plots-and-logarithmic-scales/

Tips if you are writing a graph command

1) Avoid the log-transformed data approach if you possibly can – it is harder for you and the user.

   -does the user have to input log10 values if they specify xtick(), yline(), text() etc.?

2) Labels outside the data range? Beware the smallest label can sometimes be too far away.

3) Put a version of -decidelabels- in your ado file.

4) Mention the need to download other community-written programs in your help file if you are using them e.g. -niceloglabels-.

Part 2. Nice log-like-scaled axes

Q. How to produce a nice log-like-scaled axis showing 0 and/or ∞?

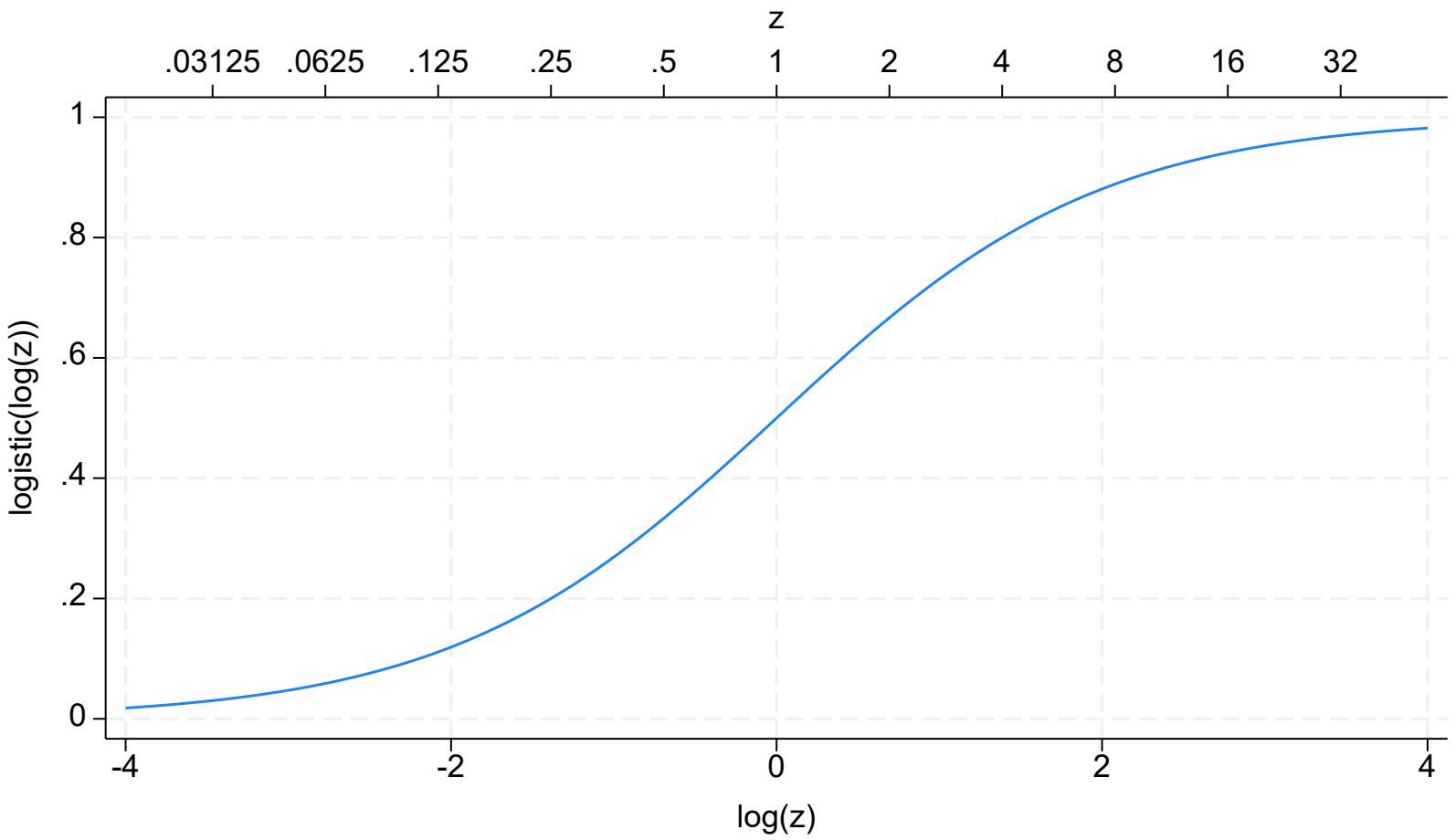
Approach will be:

Transform data

- logistic(#*log(y/#))
- asinh(y/#)

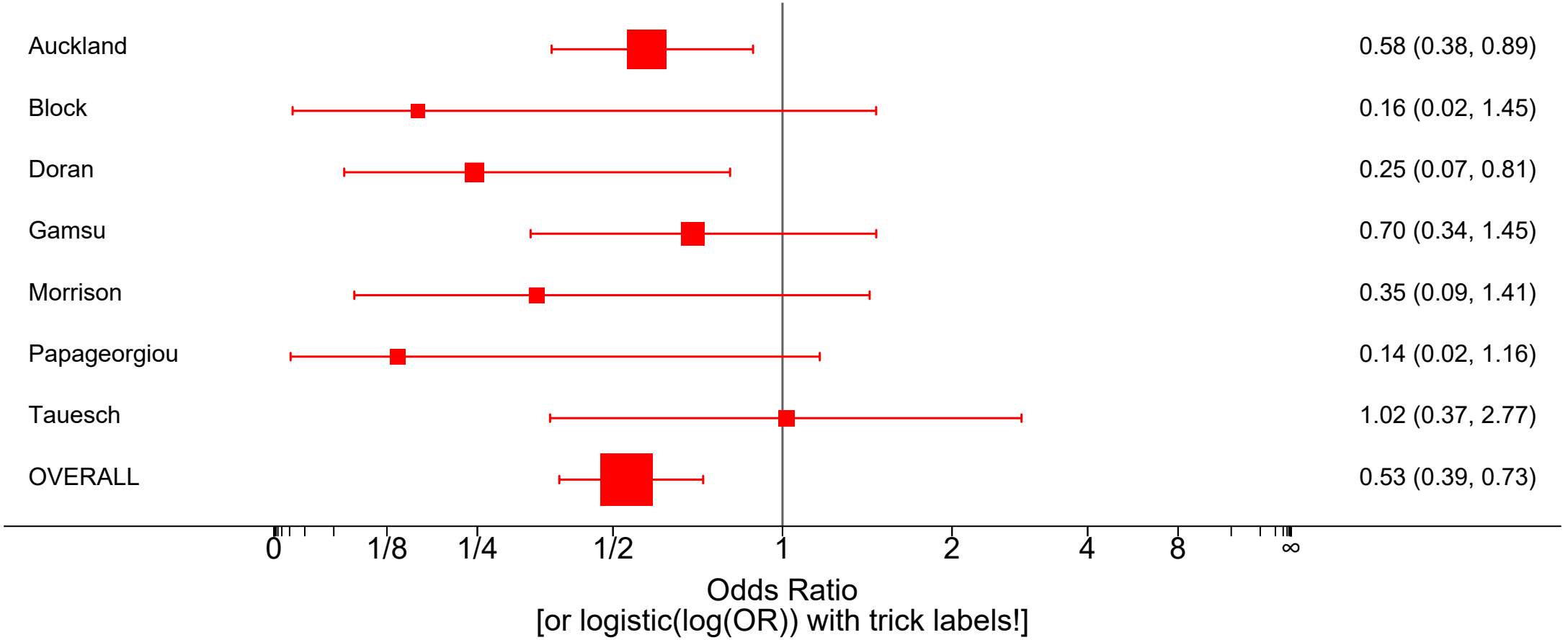
Plot transformed data,
overwriting axis title,
trick labels,
trick ticks

Introducing the logistic(log()) transformation



For 0.125 < z < 8, additive increases in logistic(log(z)) correspond well with multiplicative increases in z

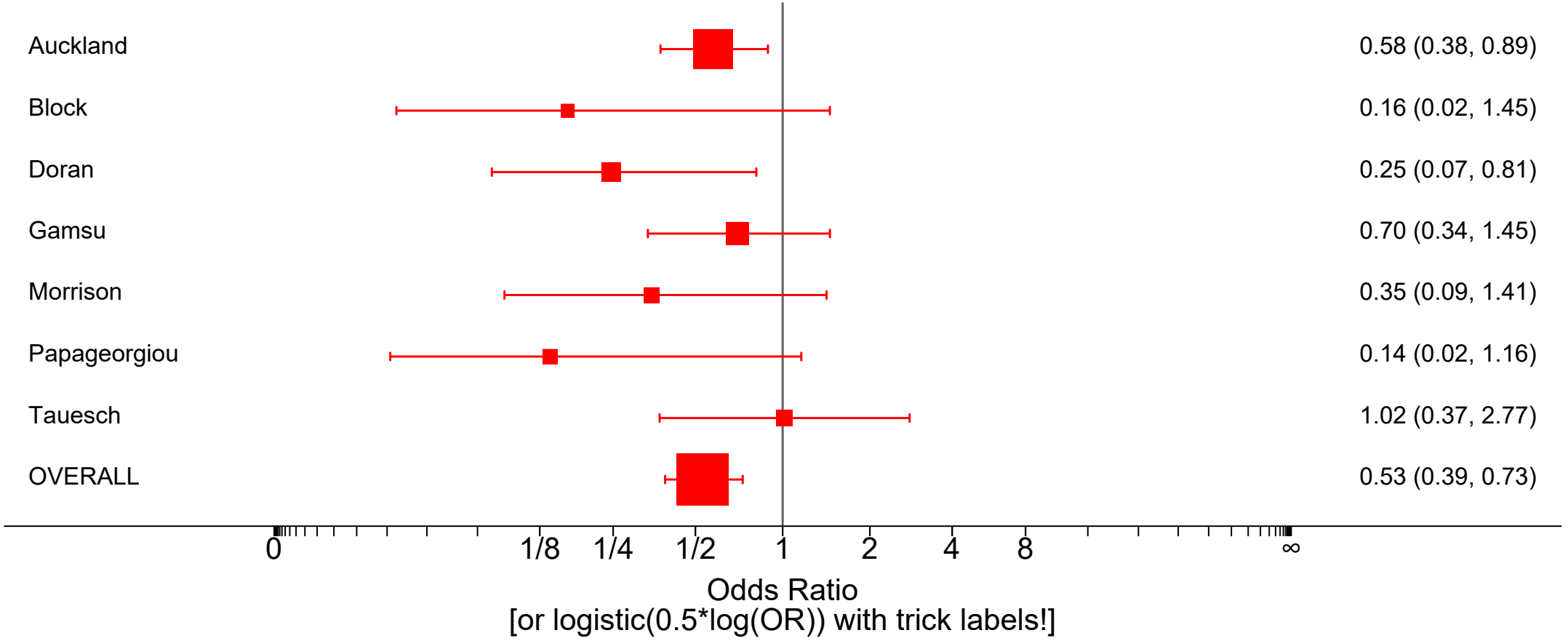
Graph I. Log-like scale ranging from 0 to ∞



Note. meta-analysis done with log odds ratios

No need to crop 95% CIs!

Graph II. Slightly different graph



Note. meta-analysis done with log odds ratios

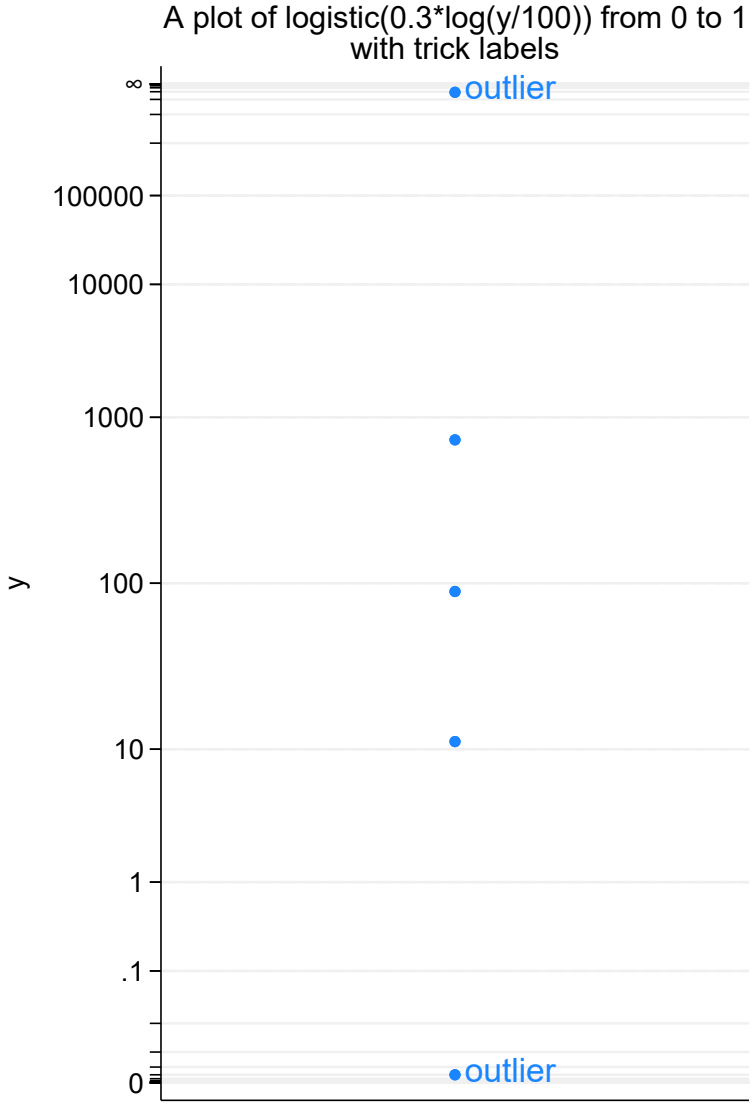
The logistic(0.5*log(z)) transformation moved the positioning of the labels *towards* 1

Syntax to create previous axis

```
. generate tr_OR_es = logistic(0.5*log(OR_es))
. generate tr_OR_cil = logistic(0.5*log(OR_cil))
. generate tr_OR_ciu = logistic(0.5*log(OR_ciu))

. niceloglabels `=1/8' 8, local(labels) style(2)
. mylabels `labels', myscale(logistic(0.5*log(@))) local(tricklabels)

. niceloglabels `=(1/2)^20' `=(2)^20', local(ticks) style(2)
. myticks `ticks', myscale(logistic(0.5*log(@))) local(trickticks)

. scatter ... tr_OR_es, ///
  xtitle("Odds Ratio") ///
  xlabel(`tricklabels' 0 "0" 1 "{&infin}", nogrid) ///
  xtick(`trickticks') ///
  xline(`=logistic(0.5*log(1))', lc(gs6) lp(solid)) ...
```

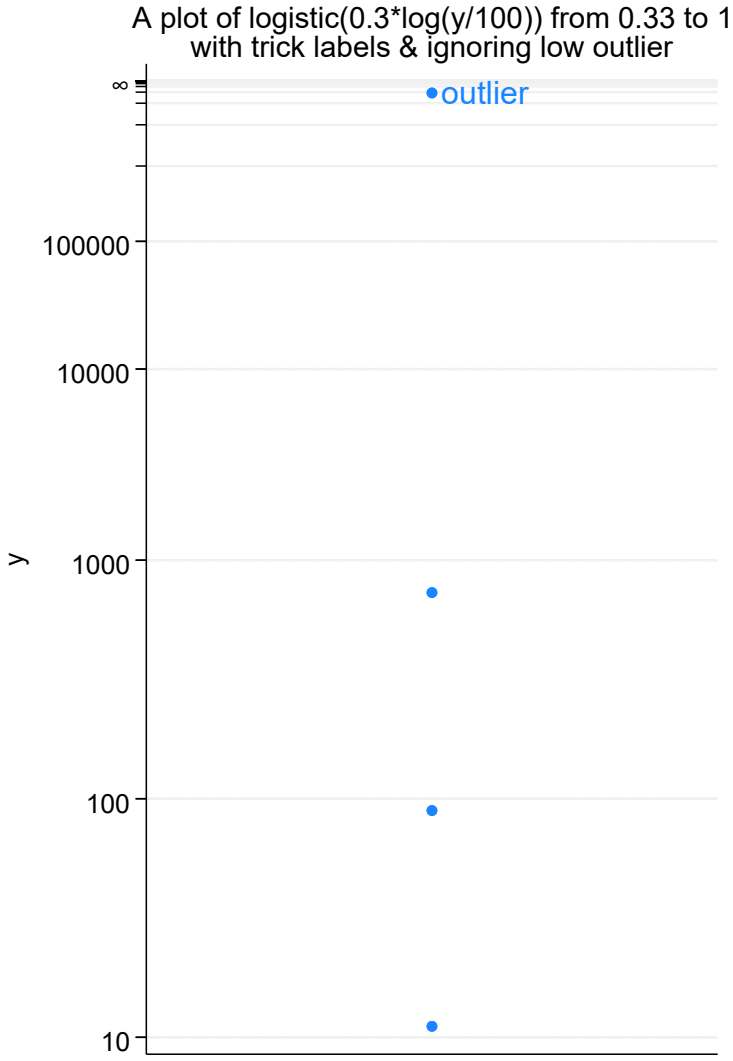
-myticks- and -mylabels- share the same help file

Graph III. Which value halfway between 0 and ∞?



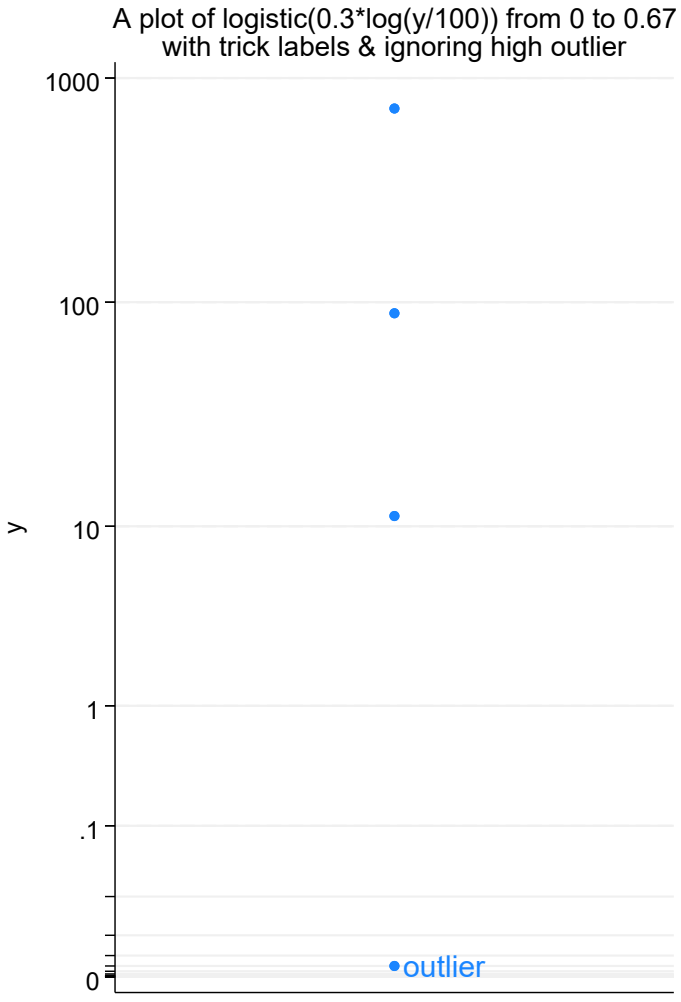

```
. generate tr_y = logistic(0.3*log(y/100))

. niceloglabels .1 100000, local(labels) style(1)
. mylabels `labels', myscale(logistic(0.3*log(@/100))) local(tricklabels)

. niceloglabels `=1/10^20' `=10^20', local(ticks) style(1)
. myticks `ticks', myscale(logistic(0.3*log(@/100))) local(trickticks)

. scatter tr_y x, ///
  ytitle("y") ///
  ylabel(`tricklabels' 0 "0" 1 "{&infin}") ///
  ytick(`trickticks', grid glpattern(solid)) ///
  ...
```

Graph IV. Log-like scale showing ∞ (but not 0)



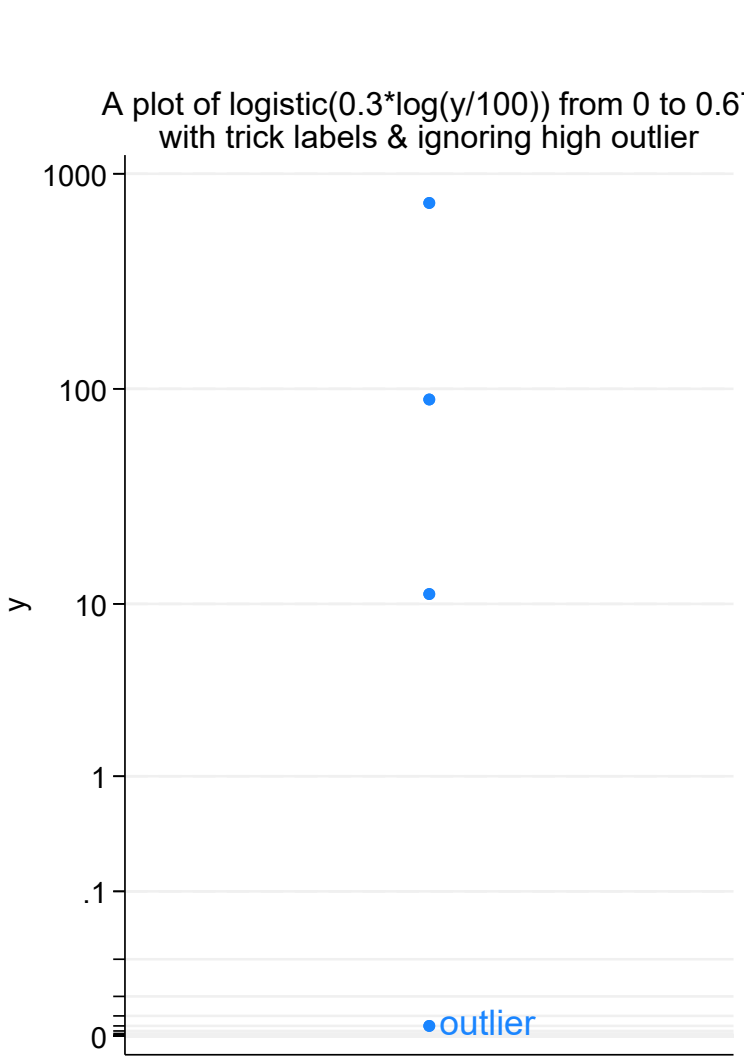

```
. generate tr_y = logistic(0.3*log(y/100))

. niceloglabels 10 100000, local(labels) style(1)
. mylabels `labels', myscale(logistic(0.3*log(@/100))) local(tricklabels)

. niceloglabels 10 `=10^20', local(ticks) style(1)
. myticks `ticks', myscale(logistic(0.3*log(@/100))) local(trickticks)

. scatter tr_y x, ///
  ytitle("y") ///
  ylabel(`tricklabels'       1 "{&infin}") ///
  ytick(`trickticks', glpattern(solid)) ///
  ...
```

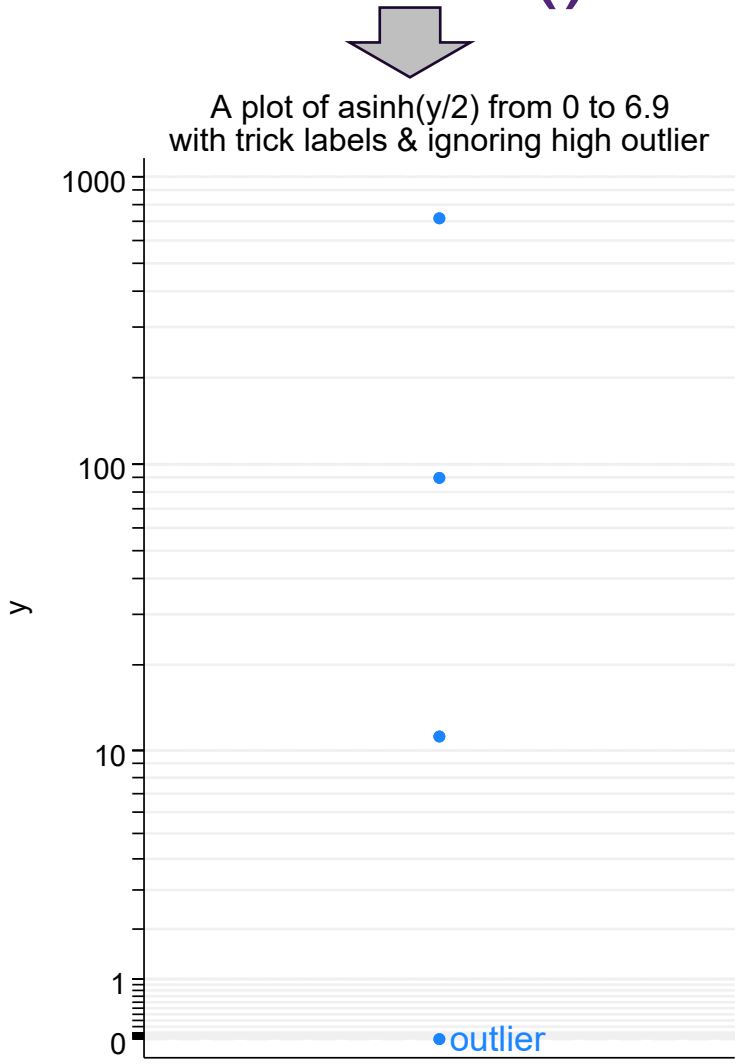
Graph V. Log-like scale showing 0 (but not ∞)




```
. generate tr_y = logistic(0.3*log(y/100))

. niceloglabels .1 1000, local(labels) style(1)
. mylabels `labels', myscale(logistic(0.3*log(@/100))) local(tricklabels)

. niceloglabels `=1/10^20' 1000, local(ticks) style(1)
. myticks `ticks', myscale(logistic(0.3*log(@/100))) local(trickticks)

. scatter tr_y x, ///
  ytitle("y") ///
  ylabel(`tricklabels' 0 "0 " ) ///
  ytick(`trickticks', glpattern(solid)) ///
  ...
```

…or could do with the asinh() transformation





← I put ticks at

100(100)1000

10(10)100

1(1)10

.1(.1)1

.01(.01).1

Linear-like between 0 and 1, otherwise log-like

Tips for nice log-like-scaled axes

- Experiment
  - many transformations you could use

    e.g. `logistic(0.3*log(y/100)), logistic(0.3*log(y/1000)), logistic(0.5*log(y/1000))`
  - many labels and ticks too

- Ticks are important where labels would be too close to each other

- Consider solid grid lines for ticks

  `ytick(..., grid glpattern(solid))`

Questions or Feedback?

Will Stata always have “additive” labels when the user specifies xscale(log) or yscale(log)?

Will there one day be an inbuilt Stata program like -decidelabels- (i.e. a log-scale equivalent of _natscale.ado)?