---

marker_label_options — Options for specifying marker labels

Description

Marker labels are labels that appear next to (or in place of) markers. Markers are the ink used to mark where points are on a plot.

Syntax

marker_label_options	Description
mlabel(*varname*)	specify marker variable
mlabstyle(*markerlabelstyle*)	overall style of label
mlabposition(*clockposstyle*)	where to locate the label
mlabvposition(*varname*)	where to locate the label 2
mlabgap(*size*)	gap between marker and label
mlabangle(*anglestyle*)	angle of label
mlabtextstyle(*textstyle*)	overall style of text
mlabsize(*textsizestyle*)	size of label
mlabcolor(*colorstyle*)	color and opacity of label
mlabformat(*%fmt*)	format of label
mlabprefix(*text*)	prefix for marker label
mlabsuffix(*text*)	suffix for marker label

All options are *rightmost*; see [G-4] **Concept: repeated options**.

Sometimes—such as when used with scatter—lists are allowed inside the arguments. A list is a sequence of the elements separated by spaces. Shorthands are allowed to make specifying the list easier; see [G-4] *stylelists*. When lists are allowed, option mlabel() allows a *varlist* in place of a *varname*.

Options

mlabel(*varname*) specifies the (usually string) variable to be used that provides, observation by observation, the marker "text". For instance, you might have

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)

. list mpg weight make in 1/4
```

	mpg	weight	make
1.	22	2,930	AMC Concord
2.	17	3,350	AMC Pacer
3.	22	2,640	AMC Spirit
4.	20	3,250	Buick Century

Typing

```
. scatter mpg weight, mlabel(make)
```

would draw a scatter of mpg versus weight and label each point in the scatter according to its make. (We recommend that you include "in 1/10" on the above command. Marker labels work well only when there are few data.)

mlabstyle(*markerlabelstyle*) specifies the overall look of marker labels, including their position, their size, their text style, etc. The other options documented below allow you to change each attribute of the marker label, but mlabstyle() is the starting point. See [G-4] *markerlabelstyle*.

You need not specify mlabstyle() just because there is something you want to change about the look of a marker and, in fact, most people seldom specify the mlabstyle() option. You specify mlabstyle() when another style exists that is exactly what you desire or when another style would allow you to specify fewer changes to obtain what you want.

mlabposition(*clockposstyle*) and mlabvposition(*varname*) specify where the label is to be located relative to the point. mlabposition() and mlabvposition() are alternatives; the first specifies a constant position for all points and the second specifies a variable that contains *clockposstyle* (a number 0–12) for each point. If both options are specified, mlabvposition() takes precedence.

If neither option is specified, the default is mlabposition(3) (3 o'clock)—meaning to the right of the point.

mlabposition(12) means above the point, mlabposition(1) means above and to the right of the point, and so on. mlabposition(0) means that the label is to be put directly on top of the point (in which case remember to also specify the msymbol(i) option so that the marker does not also display; see [G-3] *marker_options*).

mlabvposition(*varname*) specifies a numeric variable containing values 0–12, which are used, observation by observation, to locate the labels relative to the points.

See [G-4] *clockposstyle* for more information on specifying *clockposstyle*.

mlabgap(*size*) specifies how much space should be put between the marker and the label. See [G-4] *size*.

mlabangle(*anglestyle*) specifies the angle of text. The default is usually mlabangle(horizontal). See [G-4] *anglestyle*.

mlabtextstyle(*textstyle*) specifies the overall look of text of the marker labels, which here means their size and color. When you see [G-4] *textstyle*, you will find that a *textstyle* defines much more, but all of those other things are ignored for marker labels. In any case, the mlabsize() and mlabcolor() options documented below allow you to change the size and color, but mlabtextstyle() is the starting point.

As with mlabstyle(), you need not specify mlabtextstyle() just because there is something you want to change. You specify mlabtextstyle() when another style exists that is exactly what you desire or when another style would allow you to specify fewer changes to obtain what you want.

mlabsize(*textsizestyle*) specifies the size of the text. See [G-4] *textsizestyle*.

mlabcolor(*colorstyle*) specifies the color and opacity of the text. See [G-4] *colorstyle*.

mlabformat(*%fmt*) specifies the format of the text. This option is most useful when the marker labels are numeric. See [D] **format**.

mlabprefix(*text*) specifies that *text* be used as a prefix for the marker label.

mlabsuffix(*text*) specifies that *text* be used as a suffix for the marker label.

Remarks and examples

Remarks are presented under the following headings:

Typical use
Eliminating overprinting and overruns
Advanced use
Using marker labels in place of markers
Adding a prefix and suffix to the marker labels

Typical use

Markers are the ink used to mark where points are on a plot, and marker labels optionally appear beside the markers to identify the points. For instance, if you were plotting country data, marker labels would allow you to have "Argentina", "Bolivia", ..., appear next to each point. Marker labels visually work well when there are few data.

To obtain marker labels, you specify the mlabel(*varname*) option, such as mlabel(country). *varname* is the name of a variable that, observation by observation, specifies the text with which the point is to be labeled. *varname* may be a string or numeric variable, but usually it is a string. For instance, consider a subset of the life-expectancy-by-country data:

```
. use https://www.stata-press.com/data/r19/lifeexp
(Life expectancy, 1998)

. list country lexp gnppc if region==2
```

	country	lexp	gnppc
45.	Canada	79	19170
46.	Cuba	76	.
47.	Dominican Republic	71	1770
48.	El Salvador	69	1850
49.	Guatemala	64	1640
50.	Haiti	54	410
51.	Honduras	69	740
52.	Jamaica	75	1740
53.	Mexico	72	3840
54.	Nicaragua	68	1896
55.	Panama	74	2990
56.	Puerto Rico	76	.
57.	Trinidad and Tobago	73	4520
58.	United States	77	29240

We might graph these data and use labels to indicate the country by typing

```
. scatter lexp gnppc if region==2, mlabel(country)
```
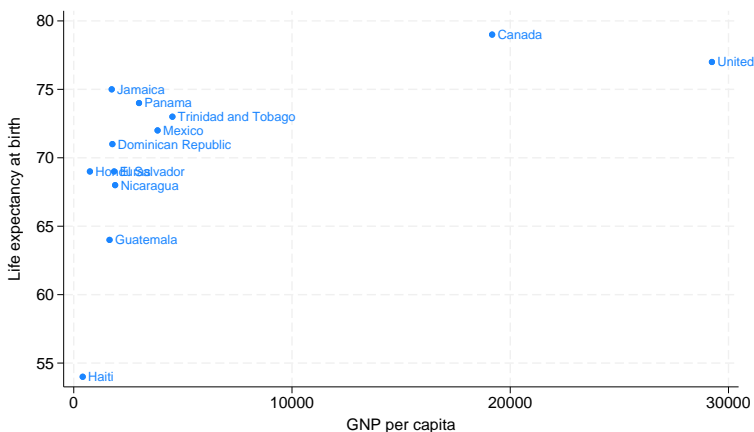

Eliminating overprinting and overruns

In the graph, the label "United States" runs off the right edge and the labels for Honduras and El Salvador are overprinted. Problems like that invariably occur when using marker labels. The mlabposition() allows specifying where the labels appear, and we might try

```
. scatter lexp gnppc if region==2, mlabel(country) mlabpos(9)
```

to move the labels to the 9 o'clock position, meaning to the left of the point. Here, however, that will introduce more problems than it will solve. You could try other clock positions around the point, but we could not find one that was satisfactory.

If our only problem were with "United States" running off the right, an adequate solution might be to widen the x axis so that there would be room for the label "United States" to fit:

```
. scatter lexp gnppc if region==2, mlabel(country)
                       xscale(range(35000))
```
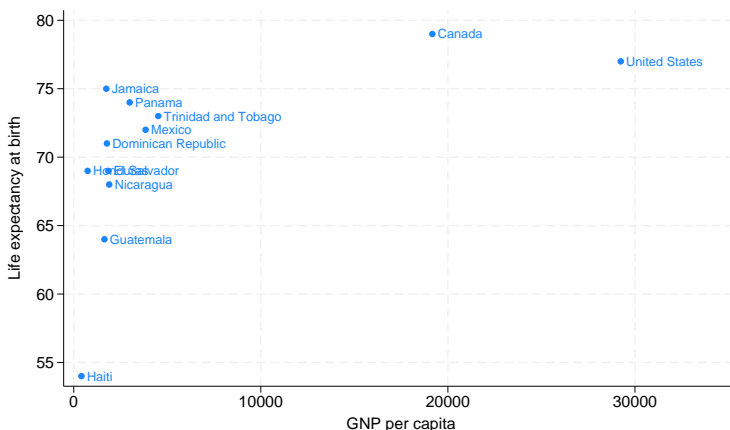
That would solve one problem but will leave us with the overprinting problem. The way to solve that problem is to move the Honduras label to being to the left of its point, and the way to do that is to specify the option `mlabvposition(varname)` rather than `mlabposition(clockposstyle)`. We will create new variable `pos` stating where we want each label:

```
. generate pos = 3
. replace pos = 9 if country=="Honduras"
(1 real change made)
. scatter lexp gnppc if region==2, mlabel(country) mlabv(pos)
                         xscale(range(35000))
```
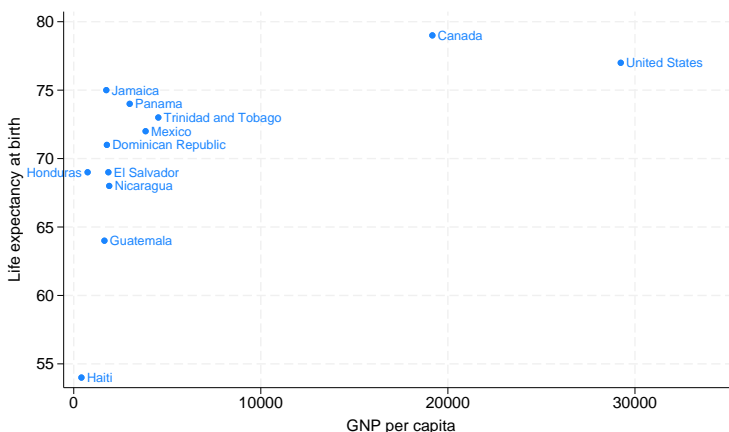

We are near a solution: Honduras is running off the left edge of the graph, but we know how to fix that. You may be tempted to solve this problem just as we solved the problem with the United States label: expand the range, say, to `range(-500 35000)`. That would be a fine solution.

Here, however, we will increase the margin between the left edge of the plot area and the y axis by adding the option plotregion(margin(l+9)); see [G-3] *region_options*. plotregion(margin(l+9)) says to increase the margin on the left by 9%, and this is really the "right" way to handle margin problems:

```
. scatter lexp gnppc if region==2, mlabel(country) mlabv(pos)
            xscale(range(35000))
            plotregion(margin(l+9))
```
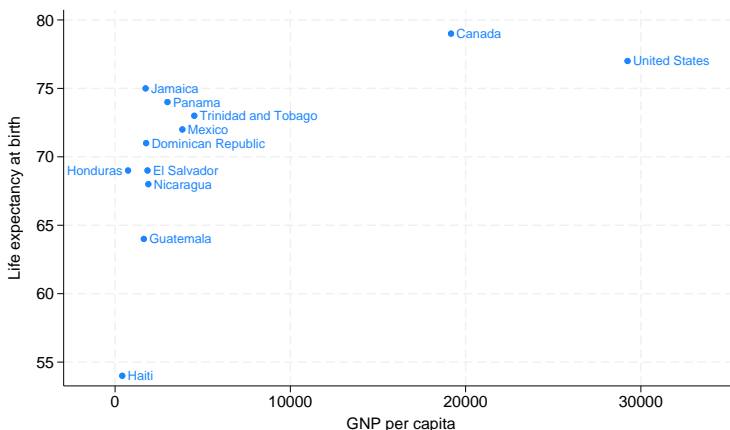

The overall result is adequate. Were we producing this graph for publication, we would move the label for United States to the left of its point, just as we did with Honduras, rather than widening the x axis.

Advanced use

Let us now consider properly graphing the life-expectancy data and graphing more of it. This time, we will include South America, as well as North and Central America, and we will graph the data on a log(GNP) scale.

```
. use https://www.stata-press.com/data/r19/lifeexp, clear
(Life expectancy, 1998)
. keep if region==2 | region==3                                       (note 1)
. replace gnppc = gnppc / 1000
. label var gnppc "GNP per capita (thousands of dollars)"             (note 2)

. generate lgnp = log(gnp)
. quietly reg lexp lgnp
. predict hat
. label var hat "Linear prediction"                                   (note 3)

. replace country = "Trinidad" if country=="Trinidad and Tobago"
. replace country = "Para" if country == "Paraguay"                   (note 4)

. generate pos = 3
. replace pos = 9 if lexp > hat                                       (note 5)

. replace pos = 3 if country == "Colombia"
. replace pos = 3 if country == "Para"
. replace pos = 3 if country == "Trinidad"
. replace pos = 9 if country == "United States"                      (note 6)
```
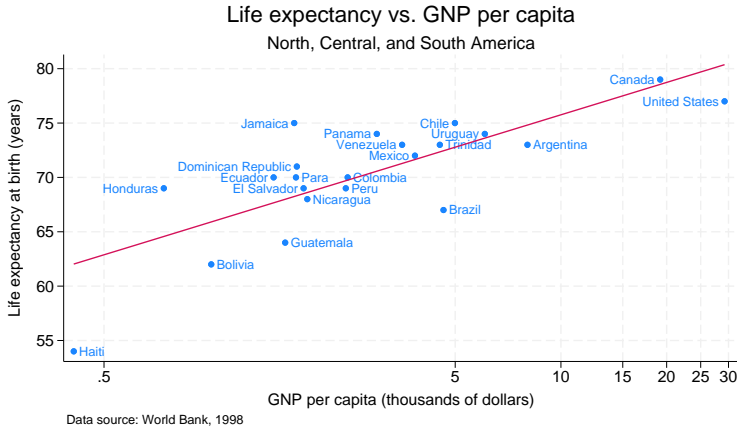
```
. twoway (scatter lexp gnppc, mlabel(country) mlabv(pos))
         (line hat gnppc, sort)
       , xscale(log) xlabel(.5 5 10 15 20 25 30) legend(off)
         title("Life expectancy vs. GNP per capita")
         subtitle("North, Central, and South America")
         note("Data source: World Bank, 1998")
         ytitle("Life expectancy at birth (years)")
```

Life expectancy vs. GNP per capita

North, Central, and South America

Data source: World Bank, 1998

Notes:

1. In these data, region 2 is North and Central America, and region 3 is South America.

2. We divide gnppc by 1,000 to keep the x axis labels from running into each other.

3. We add a linear regression prediction. We cannot use graph twoway lfit because we want the predictions to be based on a regression of log(GNP), not GNP.

4. The first time we graphed the results, we discovered that there was no way we could make the names of these two countries fit on our graph, so we shortened them.

5. We are going to place the marker labels to the left of the marker when life expectancy is above the regression line and to the right of the marker otherwise.

6. To keep labels from overprinting, we need to override rule (5) for a few countries.

Also see [G-3] *scale_option* for another rendition of this graph. In that rendition, we specify one more option—scale(1.1)—to increase the size of the text and markers by 10%.

Using marker labels in place of markers

In addition to specifying where the marker label goes relative to the marker, you can specify that the marker label be used instead of the marker. mlabposition(0) means that the label is to be centered where the marker would appear. To suppress the display of the marker as well, specify option msymbol(i); see [G-3] *marker_options*.

Using the labels in place of the points tends to work well in analysis graphs where our interest is often in identifying the outliers. Below we graph the entire `lifeexp.dta` data:

```
. scatter lexp gnppc, xscale(log) mlab(country) m(i)
```
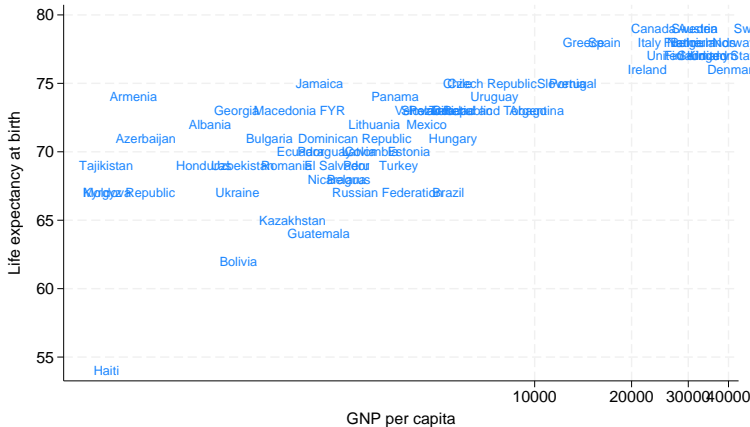

In the above graph, we also specified `xscale(log)` to convert the x axis to a log scale. A log x scale is more appropriate for these data, but had we used it earlier, the overprinting problem with Honduras and El Salvador would have disappeared, and we wanted to show how to handle the problem.

Adding a prefix and suffix to the marker labels

You may add a prefix or suffix to the marker labels with the `mlabprefix()` or `mlabsuffix()` option, respectively. For example, we have daily data recording the values for the S&P 500 in 2001. Below, we plot the opening price for the first few observations and label each marker with the closing price. Our dates are stored numerically, and to prevent the label for the rightmost point from running off the graph, we widen the x axis with option `xlabel(14977(1)14988)`. These are the numeric values for the first 10 dates. Instead of displaying the numeric values for the dates, we use the `format()` option to specify that the month and day be displayed. And we use the `mlabprefix()` option to add a dollar sign before each marker label.

```
. sysuse sp500, clear
. scatter open date in 1/9, mlabel(close)
         xlabel(14977(1)14988, format(%tdnn/dd)) mlabprefix("$")
```
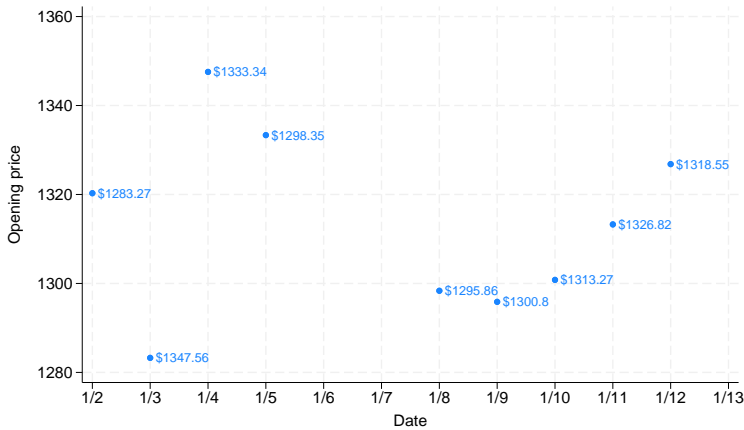
Similarly, we can add a suffix to the marker labels. For example, we would like to label each data point with the trading volume, which is measured in the thousands and ranges from about 4,000 to 23,000. First, we generate a new variable to record the trading volume in millions. Then, we use this variable to label the markers. We add the `mlabsuffix()` option to add an "M", for millions, to each marker label:

```
. generate double volume_million = volume/1000
. scatter open date in 1/9, mlabel(volume_million)
        xlabel(14977(1)14988, format(%tdnn/dd)) mlabsuffix("M")
```

Also see