The Stata Journal (2007)
7, Number 2, pp. 275–279

Stata tip 47: Quantile–quantile plots without programming

Nicholas J. Cox
Durham University
Durham City, UK
n.j.cox@durham.ac.uk

Quantile–quantile (Q–Q) plots are one of the staples of statistical graphics. Wilk and Gnanadesikan (1968) gave a detailed and stimulating review that still merits close reading. Cleveland (1993, 1994) gave more recent introductions. Here I look at their use for examining fit to distributions. The quantiles observed for a variable, which are just the data ordered from smallest to largest, may be plotted against the corresponding quantiles from some theoretical distribution. A good fit would yield a simple linear pattern. Marked deviations from linearity may indicate characteristics such as skewness, tail weight, multimodality, granularity, or outliers that do not match those of the theoretical distribution. Many consider such plots more informative than individual figures of merit or hypothesis tests and feature them prominently in intermediate or advanced surveys (e.g., Rice 2007; Davison 2003).

Official Stata includes commands for plots of observed versus expected quantiles for the normal (`qnorm`) and chi-squared (`qchi`) distributions. User-written commands can be found for other distributions. You might guess that such graphics depend on the provision of dedicated programs, but much can be done interactively just by combining some basic commands. Indeed, you can easily experiment with variations on the standard plots not yet provided in any Stata program.

Statistical and Stata tradition dictate that we start with the normal distribution and the `auto` dataset. In a departure from tradition, generate `gpm` (gallons per 100 miles) as a reciprocal of `mpg` (miles per gallon) scaled to convenient units and examine its fit to normality. You can calculate the ranks and sample size by using `egen`:

```
. use http://www.stata-press.com/data/r9/auto
(1978 Automobile Data)
. gen gpm = 100 / mpg
. label var gpm "gallons / 100 miles"
. egen rank = rank(gpm)
. egen n = count(gpm)
```

These `egen` functions handle any missing values automatically and can easily be combined with any extra `if` and `in` conditions. You may like to specify the `unique` option with `rank()` if you have many ties on your variable. If you want to fit separate distributions to distinct groups, apply the `by:` prefix, say,

```
. by foreign, sort: egen rank = rank(gpm)
. by foreign: egen n = count(gpm)
```


Next choose a formula for plotting positions given rank i and count n. These positions are cumulative probabilities associated with the data. The formula i/n would imply that no value could be larger than the largest observed in the sample and would render the normal quantile unplottable for the same extreme. The formula $(i-1)/n$ would be similarly objectionable at the opposite extreme. Various alternatives have been proposed, typically $(i-a)/(n-2a+1)$ for some a: you may choose for yourself. `qnorm` has $i/(n+1)$ (i.e., $a = 0$) wired in, but let us take $a = 0.5$ to emphasize our freedom. A minimal plot is now within reach using `invnormal()`, the normal quantile or inverse cumulative distribution function. Figure 1 is our first stab, with separate fits for the two groups of cars.

```
. gen pp = (rank - 0.5) / n
. gen normal = invnormal(pp)
. scatter gpm normal if foreign, ms(oh) ||
> scatter gpm normal if !foreign, ms(S) yla(, ang(h))
> legend(order(1 "Foreign" 2 "Domestic") ring(0) pos(5) col(1))
> xti(standard normal)
```

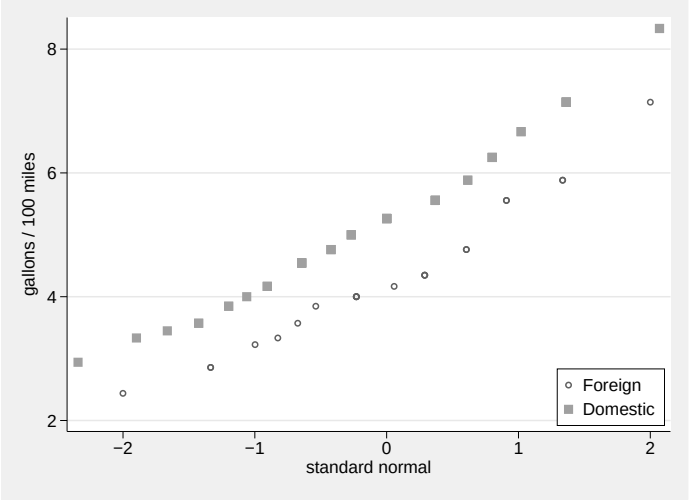

Figure 1: Normal probability plots for gallons per 100 miles for foreign and domestic cars

You might want to fit means and standard deviations explicitly. The easiest way is once again to use `egen`:

```
. by foreign: egen mean = mean(gpm)
. by foreign: egen sd = sd(gpm)
. gen normal2 = mean + sd * normal
. scatter gpm normal2, by(foreign, note("") legend(off)) ||
> function equality = x, ra(normal2) yla(, ang(h))
> xti(fitted normal) yti(gallons / 100 miles)
```

Graph not shown to save space

Already with just a few lines we can do something not available with `qnorm`: plotting two or more groups. We can superimpose, as in figure 1, or juxtapose, as in the last example.

Variants of the basic Q–Q plot are also close at hand. Wilk and Gnanadesikan (1968) suggested some possibilities. As is standard practice in examining model fit, we may subtract the general tilt of the Q–Q plot by looking at the residuals, the differences between observed and expected quantiles. These may be plotted against either the expected quantiles or the plotting positions. The two graphs convey similar information. These difference quantile plots might be called DQ plots for short. DQ plots are in essence more demanding than standard Q–Q plots, as they make discrepancies from expectation more evident. As with residual plots, the reference line is no longer a diagonal line of equality but rather the horizontal line of zero difference or residual. Figure 2 shows the two possibilities mentioned. Although `gpm` is more nearly Gaussian than `mpg`, some marked skewness remains. Lowess or other smoothing could be used to identify any systematic structure.

```
. gen residual = gpm - normal2
. scatter residual normal2 if foreign, ms(oh) ||
> scatter residual normal2 if !foreign, ms(S)
> legend(order(1 "Foreign" 2 "Domestic") pos(5) ring(0) col(1))
> yla(, ang(h)) yli(0) xti(fitted normal) saving(graph1)
(file graph1.gph saved)
. scatter residual pp if foreign, ms(oh) ||
> scatter residual pp if !foreign, ms(S)
> legend(order(1 "Foreign" 2 "Domestic") pos(5) ring(0) col(1))
> yla(, ang(h)) yli(0) xti(plotting position) saving(graph2)
(file graph2.gph saved)
. graph combine graph1.gph graph2.gph
```

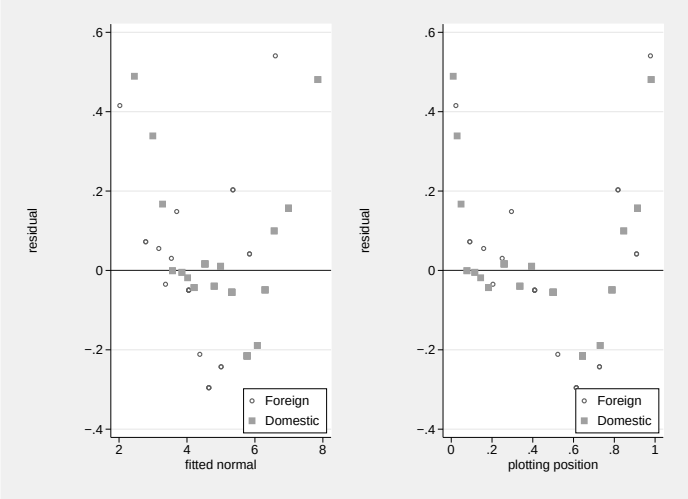

Figure 2: DQ plots for gallons per 100 miles for foreign and domestic cars and normal distribution. Residual versus (left) fitted quantile and (right) plotting position.

The example distribution, the normal, is specified by a location parameter and a scale parameter. This fact gives the flexibility of either fitting parameters or not fitting parameters first. If the theoretical distribution is also specified by one or more shape parameters, we would need to specify those first.

Turning away from the normal, we close with different examples. Q–Q plots and various relatives are prominent in work on the statistics of extremes (e.g., Coles 2001; Reiss and Thomas 2001; Beirlant et al. 2004) and more generally in work with heavy- or fat-tailed distributions. One way of using Q–Q plots is as an initial exploratory device, comparing a distribution, or its more interesting tail, with some reference distribution. For exponential distributions,

```
. generate exponential = -ln(1 - pp)
```

and plot data against that. On such plots, distributions heavier tailed than the exponential will be convex down and those lighter tailed will be convex up (Beirlant et al. 2004). For work with maximums, the Gumbel distribution is a basic starting point.

```
. generate Gumbel = -ln(-ln(pp))
```

Figure 3 is a basic Gumbel plot for annual maximum sea levels at Port Pirie in Australia (data for 1923–1987 from Coles 2001).

```
. scatter level Gumbel, yla(, ang(h)) xti(standard Gumbel)
```

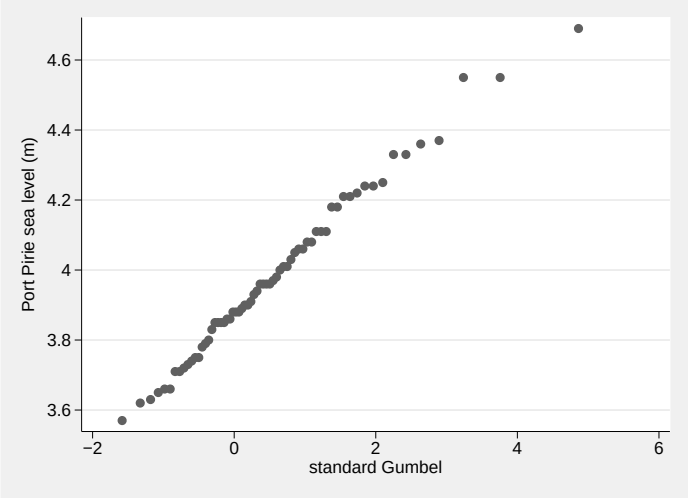

Figure 3: Basic Gumbel plot for annual maximum sea levels at Port Pirie in Australia

The generally good linearity encourages a more formal fit. Convex or concave curves would have pointed to fitting other members of the generalized extreme value distribution family.

References

Beirlant, J., Y. Goegebeur, J. Segers, and J. Teugels. 2004. *Statistics of Extremes: Theory and Applications*. New York: Wiley.

Cleveland, W. S. 1993. *Visualizing Data*. Summit, NJ: Hobart Press.

———. 1994. *The Elements of Graphing Data*. Summit, NJ: Hobart Press.

Coles, S. 2001. *An Introduction to Statistical Modeling of Extreme Values*. London: Springer.

Davison, A. C. 2003. *Statistical Models*. Cambridge: Cambridge University Press.

Reiss, R.-D., and M. Thomas. 2001. *Statistical Analysis of Extreme Values with Applications to Insurance, Finance, Hydrology, and Other Fields*. Basel: Birkhäuser.

Rice, J. A. 2007. *Mathematical Statistics and Data Analysis*. Belmont, CA: Duxbury.

Wilk, M. B., and R. Gnanadesikan. 1968. Probability plotting methods for the analysis of data. *Biometrika* 55: 1–17.

The Stata Journal (2007)
7, Number 2, p. 280

Software Updates

gr0001_3: Generalized Lorenz curves and related graphs: Update for Stata 7. P. van Kerm and S. P. Jenkins. *Stata Journal* 6: 597; 4: 490; 1: 107–112.

With suitable choice of options, `glcurve` returns concentration curve ordinates. In previous versions, the values returned depended on the sort order of the data if there were ties in the ordering variable. `glcurve` has been modified so that, in this case, there is a stable sort, namely in increasing order of the outcome variable within ties of the ordering variable (the maximum concentration case). The effects on results are likely to be negligible with unit-record data, but may be perceptible if there are many ties on the ranking variable used in the `sortvar()` option, as for example when using concentration curve ordinates created by `glcurve` to estimate a concentration coefficient with grouped (banded) data.

st0097_1: Generalized ordered logit/partial proportional odds models for ordinal dependent variables. R. Williams. *Stata Journal* 6: 58–82.

There have been several enhancements to `gologit2`. Different link functions (logit, probit, cloglog, loglog, and cauchit) can now be specified with the `link()` option. Diagnostic tests are done to check for the rare problem of negative predicted probabilities. Many Stata 9 prefix commands (e.g., `nestreg`, `stepwise`) are now supported, while the program continues to run under Stata 8.2. `gologit2` works better with several postestimation commands, such as `mfx`, `mfx2`, `outreg2`, and `estout`. There have been various minor bug fixes. The official support page, http://www.nd.edu/~rwilliam/gologit2/index.html, now includes a trouble-shooting frequently asked question section and more reading material.

