Title

stata.com

runtest — Test for random order

Syntax

```
runtest varname [in] [, options]
```

options	Description
`continuity`	continuity correction
`drop`	ignore values equal to the threshold
`split`	randomly split values equal to the threshold as above or below the threshold; default is to count as below
`mean`	use mean as threshold; default is median
`threshold(#)`	assign arbitrary threshold; default is median

Menu

Statistics > Nonparametric analysis > Tests of hypotheses > Test for random order

Description

`runtest` tests whether the observations of *varname* are serially independent—that is, whether they occur in a random order—by counting how many runs there are above and below a threshold. By default, the median is used as the threshold. A small number of runs indicates positive serial correlation; a large number indicates negative serial correlation.

Options

`continuity` specifies a continuity correction that may be helpful in small samples. If there are fewer than 10 observations either above or below the threshold, however, the tables in Swed and Eisenhart (1943) provide more reliable critical values. By default, no continuity correction is used.

`drop` directs `runtest` to ignore any values of *varname* that are equal to the threshold value when counting runs and tabulating observations. By default, `runtest` counts a value as being above the threshold when it is strictly above the threshold and as being below the threshold when it is less than or equal to the threshold.

`split` directs `runtest` to randomly split values of *varname* that are equal to the threshold. In other words, when *varname* is equal to threshold, a "coin" is flipped. If it comes up heads, the value is counted as above the threshold. If it comes up tails, the value is counted as below the threshold.

`mean` directs `runtest` to tabulate runs above and below the mean rather than the median.

`threshold(#)` specifies an arbitrary threshold to use in counting runs. For example, if *varname* has already been coded as a 0/1 variable, the median generally will not be a meaningful separating value.

Remarks and examples

stata.com

`runtest` performs a nonparametric test of the hypothesis that the observations of *varname* occur in a random order by counting how many runs there are above and below a threshold. If *varname* is positively serially correlated, it will tend to remain above or below its median for several observations in a row; that is, there will be relatively few runs. If, on the other hand, *varname* is negatively serially correlated, observations above the median will tend to be followed by observations below the median and vice versa; that is, there will be relatively many runs.

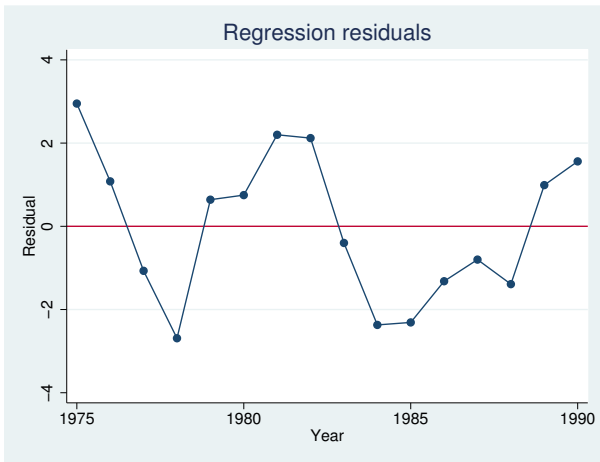
By default, `runtest` uses the median for the threshold, and this is not necessarily the best choice. If `mean` is specified, the mean is used instead of the median. If `threshold(#)` is specified, # is used. Because `runtest` divides the data into two states—above and below the threshold—it is appropriate for data that are already binary; for example, win or lose, live or die, rich or poor, etc. Such variables are often coded as 0 for one state and 1 for the other. Here you should specify `threshold(0)` because, by default, `runtest` separates the observations into those that are greater than the threshold and those that are less than *or equal* to the threshold.

As with most nonparametric procedures, the treatment of ties complicates the test. Observations equal to the threshold value are ties and can be treated in one of three ways. By default, they are treated as if they were below the threshold. If `drop` is specified, they are omitted from the calculation and the total number of observations is adjusted. If `split` is specified, each is randomly assigned to the above- and below-threshold groups. The random assignment is different each time the procedure is run unless you specify the random-number seed; see [R] **set seed**.

▷ Example 1

We can use `runtest` to check regression residuals for serial correlation.

```
. use http://www.stata-press.com/data/r13/run1
. scatter resid year, connect(l) yline(0) title(Regression residuals)
```



The graph gives the impression that these residuals are positively correlated. Excursions above or below zero—the natural threshold for regression residuals—tend to last for several observations. `runtest` can evaluate the statistical significance of this impression.

```
. runtest resid, thresh(0)
 N(resid <= 0) = 8
 N(resid >  0) = 8
           obs = 16
      N(runs) = 5
            z  = -2.07
     Prob>|z| = .04
```

There are five runs in these 16 observations. Using the normal approximation to the true distribution of the number of runs, the five runs in this series are fewer than would be expected if the residuals were serially independent. The p-value is 0.04, indicating a two-sided significant result at the 5% level. If the alternative hypothesis is positive serial correlation, rather than any deviation from randomness, then the one-sided p-value is $0.04/2 = 0.015$. With so few observations, however, the normal approximation may be inaccurate. (Tables compiled by Swed and Eisenhart list five runs as the 5% critical value for a one-sided test.)

`runtest` is a nonparametric test. It ignores the magnitudes of the observations and notes only whether the values are above or below the threshold. We can demonstrate this feature by reducing the information about the regression residuals in this example to a 0/1 variable that indicates only whether a residual is positive or negative.

```
. generate byte sign = resid>0
. runtest sign, thresh(0)
 N(sign <= 0) = 8
 N(sign >  0) = 8
          obs = 16
     N(runs) = 5
           z  = -2.07
    Prob>|z| = .04
```

As expected, `runtest` produces the same answer as before.

◁

❑ Technical note

The run test can also be used to test the null hypothesis that two samples are drawn from the same underlying distribution. The run test is sensitive to differences in the shapes, as well as the locations, of the empirical distributions.

Suppose, for example, that two different additives are added to the oil in 10 different cars during an oil change. The cars are run until a viscosity test determines that another oil change is needed, and the number of miles traveled between oil changes is recorded. The data are

```
. use http://www.stata-press.com/data/r13/additive, clear
. list
```

	additive	miles
1.	1	4024
2.	1	4756
3.	1	7993
4.	1	5025
5.	1	4188
6.	2	3007
7.	2	1988
8.	2	1051
9.	2	4478
10.	2	4232

To test whether the additives generate different distributions of miles between oil changes, we sort the data by `miles` and then use `runtest` to see whether the marker for each additive occurs in random order:

```
. sort miles
. runtest additive, thresh(1)
 N(additive <= 1) = 5
 N(additive >  1) = 5
              obs = 10
         N(runs) = 4
               z  = -1.34
        Prob>|z| = .18
```

Here the additives do not produce statistically different results.

❑

❑ Technical note

A test that is related to the run test is the runs up-and-down test. In the latter test, the data are classified not by whether they lie above or below a threshold but by whether they are steadily increasing or decreasing. Thus an unbroken string of increases in the variable of interest is counted as one run, as is an unbroken string of decreases. According to Madansky (1988), the run test is superior to the runs up-and-down test for detecting trends in the data, but the runs up-and-down test is superior for detecting autocorrelation.

`runtest` can be used to perform a runs up-and-down test. Using the regression residuals from the example above, we can perform a `runtest` on their first differences:

```
. use http://www.stata-press.com/data/r13/run1
. generate resid_D = resid - resid[_n-1]
(1 missing value generated)
. runtest resid_D, thresh(0)
 N(resid_D <= 0) = 7
 N(resid_D >  0) = 8
              obs = 15
         N(runs) = 6
               z  = -1.33
        Prob>|z| = .18
```

Edgington (1961) has compiled a table of the small sample distribution of the runs up-and-down statistic, and this table is reprinted in Madansky (1988). For large samples, the z statistic reported by `runtest` is incorrect for the runs up-and-down test. Let N be the number of observations (15 here), and let r be the number of runs (6). The expected number of runs in the runs up-and-down test is

$$\mu_r = \frac{2N-1}{3}$$

the variance is

$$\sigma_r^2 = \frac{16N-29}{90}$$

and the correct z statistic is

$$\widehat{z} = \frac{r-\mu_r}{\sigma_r}$$

❑

❑ Technical note

`runtest` will tolerate missing values at the beginning or end of a series, as occurred in the technical note above (generating first differences resulted in a missing value for the first observation). `runtest`, however, will issue an error message if there are any missing observations in the interior of the series (in the portion covered by the `in` *range* modifier). To perform the test anyway, simply `drop` the missing observations before using `runtest`.

❑

Stored results

`runtest` stores the following in `r()`:

Scalars

`r(N)`	number of observations	`r(p)`	p-value of z
`r(N_below)`	number below the threshold	`r(z)`	z statistic
`r(N_above)`	number above the threshold	`r(n_runs)`	number of runs
`r(mean)`	expected number of runs	`r(Var)`	variance of the number of runs

Methods and formulas

`runtest` begins by calculating the number of observations below the threshold, n_0; the number of observations above the threshold, n_1; the total number of observations, $N = n_0 + n_1$; and the number of runs, r. These statistics are always reported, so the exact tables of critical values in Swed and Eisenhart (1943) may be consulted if necessary.

The expected number of runs under the null is

$$\mu_r = \frac{2n_0n_1}{N} + 1$$

the variance is

$$\sigma_r^2 = \frac{2n_0n_1\left(2n_0n_1 - N\right)}{N^2\left(N-1\right)}$$

and the normal approximation test statistic is

$$\widehat{z} = \frac{r - \mu_r}{\sigma_r}$$

Acknowledgment

`runtest` was written by Sean Becketti, a past editor of the *Stata Technical Bulletin* and author of the Stata Press book *Introduction to Time Series Using Stata*.

References

Edgington, E. S. 1961. Probability table for number of runs of signs of first differences in ordered series. *Journal of the American Statistical Association* 56: 156–159.

Madansky, A. 1988. *Prescriptions for Working Statisticians*. New York: Springer.

Swed, F. S., and C. Eisenhart. 1943. Tables for testing randomness of grouping in a sequence of alternatives. *Annals of Mathematical Statistics* 14: 66–87.