linktest — Specification link test for single-equation models

Description

`linktest` performs a link test for model specification.

Quick start

Specification link test after a single-equation estimation command without options

```
linktest
```

After `tobit` estimated with right-censoring limit at 24

```
linktest, ul(24)
```

After `stcox` estimated with Efron method for tied failures

```
linktest, efron
```

Perform test on full dataset when estimation used a subset of observations

```
linktest if e(sample) < .
```

Menu

Statistics > Postestimation

Syntax

`linktest` [*if*] [*in*] [, *cmd_options*]

When `if` and `in` are not specified, the link test is performed on the same sample as the previous estimation.
`collect` is allowed; see **[U] 11.1.10 Prefix commands**.

Option

Main

cmd_options must be the same options specified with the underlying estimation command, except the *display_options* may differ.

Remarks and examples

The form of the link test implemented here is based on an idea of Tukey (1949), which was further described by Pregibon (1980), elaborating on work in his unpublished thesis (Pregibon 1979). See *Methods and formulas* below for more details.

▷ Example 1

We want to explain the mileage ratings of cars in our automobile dataset by using the weight, engine displacement, and whether the car is manufactured outside the United States:

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)
. regress mpg weight displ foreign
```

Source	SS	df	MS	Number of obs	=	74
				F(3, 70)	=	45.88
Model	1619.71935	3	539.906448	Prob > F	=	0.0000
Residual	823.740114	70	11.7677159	R-squared	=	0.6629
				Adj R-squared	=	0.6484
Total	2443.45946	73	33.4720474	Root MSE	=	3.4304

mpg	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
weight	-.0067745	.0011665	-5.81	0.000	-.0091011	-.0044479
displacement	.0019286	.0100701	0.19	0.849	-.0181556	.0220129
foreign	-1.600631	1.113648	-1.44	0.155	-3.821732	.6204699
_cons	41.84795	2.350704	17.80	0.000	37.15962	46.53628

On the basis of the R^2, we are reasonably pleased with this model.

If our model really is specified correctly, then if we were to regress `mpg` on the prediction and the prediction squared, the prediction squared would have no explanatory power. This is what `linktest` does:

```
. linktest
```

Source	SS	df	MS	Number of obs	=	74
				F(2, 71)	=	76.75
Model	1670.71514	2	835.357572	Prob > F	=	0.0000
Residual	772.744316	71	10.8837228	R-squared	=	0.6837
				Adj R-squared	=	0.6748
Total	2443.45946	73	33.4720474	Root MSE	=	3.299

mpg	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
_hat	-.4127198	.6577736	-0.63	0.532	-1.724283	.8988434
_hatsq	.0338198	.015624	2.16	0.034	.0026664	.0649732
_cons	14.00705	6.713276	2.09	0.041	.6211539	27.39294

We find that the prediction squared does have explanatory power, so our specification is not as good as we thought.

Although `linktest` is formally a test of the specification of the dependent variable, it is often interpreted as a test that, conditional on the specification, the independent variables are specified incorrectly. We will follow that interpretation and now include weight squared in our model:

```
. regress mpg weight c.weight#c.weight displ foreign
```

Source	SS	df	MS	Number of obs	=	74
				F(4, 69)	=	39.37
Model	1699.02634	4	424.756584	Prob > F	=	0.0000
Residual	744.433124	69	10.7888859	R-squared	=	0.6953
				Adj R-squared	=	0.6777
Total	2443.45946	73	33.4720474	Root MSE	=	3.2846

mpg	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
weight	-.0173257	.0040488	-4.28	0.000	-.0254028	-.0092486
c.weight# c.weight	1.87e-06	6.89e-07	2.71	0.008	4.93e-07	3.24e-06
displacement	-.0101625	.0106236	-0.96	0.342	-.031356	.011031
foreign	-2.560016	1.123506	-2.28	0.026	-4.801349	-.3186832
_cons	58.23575	6.449882	9.03	0.000	45.36859	71.10291

Now, we perform the link test on our new model:

```
. linktest
```

Source	SS	df	MS	Number of obs	=	74
				F(2, 71)	=	81.08
Model	1699.39489	2	849.697445	Prob > F	=	0.0000
Residual	744.06457	71	10.4797827	R-squared	=	0.6955
				Adj R-squared	=	0.6869
Total	2443.45946	73	33.4720474	Root MSE	=	3.2372

mpg	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
_hat	1.141987	.7612218	1.50	0.138	-.3758456	2.659821
_hatsq	-.0031916	.0170194	-0.19	0.852	-.0371272	.0307441
_cons	-1.50305	8.196444	-0.18	0.855	-17.84629	14.84019

We now pass the link test.

◁

Example 2

Above, we followed a standard misinterpretation of the link test—when we discovered a problem, we focused on the explanatory variables of our model. We might consider varying exactly what the link test tests. The link test told us that our dependent variable was misspecified. For those with an engineering background, mpg is indeed a strange measure. It would make more sense to model energy consumption—gallons per mile—in terms of weight and displacement:

```
. gen gpm = 1/mpg
. regress gpm weight displ foreign
```

Source	SS	df	MS	Number of obs	=	74
				F(3, 70)	=	76.33
Model	.009157962	3	.003052654	Prob > F	=	0.0000
Residual	.002799666	70	.000039995	R-squared	=	0.7659
				Adj R-squared	=	0.7558
Total	.011957628	73	.000163803	Root MSE	=	.00632

gpm	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
weight	.0000144	2.15e-06	6.72	0.000	.0000102	.0000187
displacement	.0000186	.0000186	1.00	0.319	-.0000184	.0000557
foreign	.0066981	.0020531	3.26	0.002	.0026034	.0107928
_cons	.0008917	.0043337	0.21	0.838	-.0077515	.009535

This model looks every bit as reasonable as our original model:

```
. linktest
```

Source	SS	df	MS	Number of obs	=	74
				F(2, 71)	=	117.06
Model	.009175219	2	.004587609	Prob > F	=	0.0000
Residual	.002782409	71	.000039189	R-squared	=	0.7673
				Adj R-squared	=	0.7608
Total	.011957628	73	.000163803	Root MSE	=	.00626

gpm	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
_hat	.6608413	.515275	1.28	0.204	-.3665877	1.68827
_hatsq	3.275857	4.936655	0.66	0.509	-6.567553	13.11927
_cons	.008365	.0130468	0.64	0.523	-.0176496	.0343795

Specifying the model in terms of gallons per mile also solves the specification problem and results in a more parsimonious specification.

◁

Example 3

The link test can be used with any single-equation estimation procedure, not solely regression. Let's turn our problem around and attempt to explain whether a car is manufactured outside the United States by its mileage rating and weight. To save paper, we will specify `logit`'s `nolog` option, which suppresses the iteration log:

```
. logit foreign mpg weight, nolog
Logistic regression                                     Number of obs =     74
                                                        LR chi2(2)    =  35.72
                                                        Prob > chi2   = 0.0000
Log likelihood = -27.175156                             Pseudo R2     = 0.3966

------------------------------------------------------------------------------
     foreign | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         mpg |  -.1685869   .0919175    -1.83   0.067    -.3487418     .011568
      weight |  -.0039067   .0010116    -3.86   0.000    -.0058894    -.001924
       _cons |   13.70837   4.518709     3.03   0.002     4.851859    22.56487
------------------------------------------------------------------------------
```

When we run `linktest` after `logit`, the result is another logit specification:

```
. linktest, nolog
Logistic regression                                     Number of obs =     74
                                                        LR chi2(2)    =  36.83
                                                        Prob > chi2   = 0.0000
Log likelihood = -26.615714                             Pseudo R2     = 0.4090

------------------------------------------------------------------------------
     foreign | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        _hat |   .8438531   .2738759     3.08   0.002     .3070661     1.38064
      _hatsq |  -.1559115   .1568642    -0.99   0.320    -.4633596     .1515366
       _cons |   .2630557   .4299598     0.61   0.541      -.57965     1.105761
------------------------------------------------------------------------------
```

The link test reveals no problems with our specification.

If there had been a problem, we would have been virtually forced to accept the misinterpretation of the link test—we would have reconsidered our specification of the independent variables. When using `logit`, we have no control over the specification of the dependent variable other than to change likelihood functions.

We admit to having seen a dataset once for which the link test rejected the logit specification. We did change the likelihood function, refitting the model using `probit`, and satisfied the link test. Probit has thinner tails than logit. In general, however, you will not be so lucky.

◁

❑ Technical note

You should specify the same options with `linktest` that you do with the estimation command, although you do not have to follow this advice as literally as we did in the preceding example. `logit`'s `nolog` option merely suppresses a part of the output, not what is estimated. We specified `nolog` both times to save space.

If you are testing a tobit model, you must specify the censoring points just as you do with the `tobit` command.

If you are not sure which options are important, duplicate exactly what you specified on the estimation command.

If you do not specify `if` *exp* or `in` *range* with `linktest`, Stata will by default perform the link test on the same sample as the previous estimation. Suppose that you omitted some data when performing your estimation but want to calculate the link test on all the data, which you might do if you believe the model is appropriate for all the data. You would type `linktest if e(sample) < .` to do this.

❑

Stored results

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

Scalars

`r(t)`	t statistic on `_hatsq`
`r(df)`	degrees of freedom

`linktest` is *not* an estimation command in the sense that it leaves previous estimation results unchanged. For instance, after running a regression and performing the link test, typing `regress` without arguments after the link test still replays the original regression.

For integrating an estimation command with `linktest`, `linktest` assumes that the name of the estimation command is stored in `e(cmd)` and that the name of the dependent variable is stored in `e(depvar)`. After estimation, it assumes that the number of degrees of freedom for the t test is given by `e(df_m)` if the macro is defined.

If the estimation command reports z statistics instead of t statistics, `linktest` will also report z statistics. The z statistic, however, is still returned in `r(t)`, and `r(df)` is set to a missing value.

Methods and formulas

The link test is based on the idea that if a regression or regression-like equation is properly specified, you should be able to find no additional independent variables that are significant except by chance. One kind of specification error is called a link error. In regression, this means that the dependent variable needs a transformation or "link" function to properly relate to the independent variables. The idea of a link test is to add an independent variable to the equation that is especially likely to be significant if there is a link error.

Let

$$\mathbf{y} = f(\mathbf{X}\boldsymbol{\beta})$$

be the model and $\widehat{\boldsymbol{\beta}}$ be the parameter estimates. `linktest` calculates

$$\texttt{\_hat} = \mathbf{X}\widehat{\boldsymbol{\beta}}$$

and

$$\texttt{\_hatsq} = \texttt{\_hat}^2$$

The model is then refit with these two variables, and the test is based on the significance of `_hatsq`. This is the form suggested by Pregibon (1979) based on an idea of Tukey (1949). Pregibon (1980) suggests a slightly different method that has come to be known as "Pregibon's goodness-of-link test". We prefer the older version because it is universally applicable, straightforward, and a good second-order approximation. It can be applied to any single-equation estimation technique, whereas Pregibon's more recent tests are estimation-technique specific.

References

Pregibon, D. 1979. Data analytic methods for generalized linear models. PhD diss., University of Toronto.

———. 1980. Goodness of link tests for generalized linear models. *Journal of the Royal Statistical Society*, C ser., 29: 15–24. https://doi.org/10.2307/2346405.

Tukey, J. W. 1949. One degree of freedom for non-additivity. *Biometrics* 5: 232–242. https://doi.org/10.2307/3001938.

Also see

[R] **regress postestimation** — Postestimation tools for regress