Statistics/Data Analysis

help for **drm** version 0.5 (Caspar Kaiser)

## Diagonal Reference Models (DRM)

## Syntax

**drm** *depvar* *rowvar* *colvar* [*varlist*] [**if** *exp*] [*weight*] [, *options*]

| *options* | Description |
|---|---|
| General [+] | |
| **vce(***vcetype***)** | set standard error type |
| **wgt(***str***)** | one of **cons**, **row** or **col**; specifies if weights are assumed to be constant (default) or dependent on *rowvar* (**row**) or *colvar* (**col**) |
| **intervars(***varlist***)** | interact weights p and q with *varlist* |
| **iterate(***#***)** | specifies the maximum number of iterations |
| **level(***#***)** | set confidence level |
| **coeflegend** | specifies that the legend of the coefficients and how to specify them in an expression be displayed rather than displaying the statistics for the coefficients |
| **keep** | prevents deletion of generated dummies for *rowvar* and *colvar* after estimation |
| **old** | forces **drm** to behave as it did until version 0.4, i.e. to generate illegible temporary variable names for *rowvar* and *colvar* |
| Maximum likelihood [+] | |
| **link(***str***)** | one of **linear**, **logit** or **probit**; specifies link function; default is **link(**linear**)** |
| **tech()** | specifies maximization algorithm; see ml for options |
| **difficult** | specifies difficult option; see maximize |
| **sobel** | use Sobel's method to find initial value; the default |
| **classic** | use classic initial values for mu |
| **alternative** | use alternative initial values for mu |
| **ownconstrains(***str***)** | specifies further user-written constrains |
| Least squares [+] | |
| **by(***varlist***)** | specify variables on which estimates of mu[i,i] and mu[j,j] are made conditional. If more than one variable is specified, every combination of *varlist* is taken. |
| **constrain** | explicitly constrains p to lie on [0,1] |

**pweights**, **aweights**, **fweights**, and **iweights** are allowed; see weight.
factor variables are allowed, but time-series operators are not (yet) supported.
Typing **drm** without arguments redisplays previous results.

## Introduction

**drm** is a module to estimate several versions of Sobel's (1981; 1985) diagonal reference model. Diagonal reference models are especially suited for the estimation of effects of movements across levels of categorical variables like education or social class. **drm** allows for a number of extensions that go beyond Sobel's most simple model. In particular, weights are allowed to vary conditional on 'destinations' and 'origins' and may be interacted with an arbitrary linear combination of covariates. Furthermore, diagonal population means may be estimated conditional on a further (set of) variable(s). Finally, next to the linear link function, **drm** allows for logit as well as probit links to estimate models with a binary dependent variable.

**drm** was inspired by and is an alternative to Lizardo's (2007) **diagref** command (which is no longer available online).

At minimum, **drm** requires Stata version 12.

## Description

**drm** standardly uses maximum likelihood to estimate parameters and returns in e() all that ml returns. However, by specifying the **nl** option, estimation may also be done with non-linear least squares. In this case, **drm** returns in e() whatever nl returns. See below on why outputs will look different between **nl** and **ml** estimation.

The basic model can be written as:

```
y[i,j,k] = p*mu[i,i] + q*mu[j,j] + e[i,j,k] (1)
```

Where:

```
p+q=1 and 0<=p<=1
```

Here, y[i,j,k] is the value of depvar of the [k]th observation in the [i,j]th cell. mu[i,i] and mu[j,j] are estimated population means of y in the [i,j]th cell. Cell positions [i,j] are indices in e.g. a mobility table with an origin variable (*rowvar*) with values {1,...,i,...R} and a destination variable (*colvar*) with values {1,...,j,...C}. It is necessary that R=C. p and q are weight parameters to be estimated.

The model of equation (1) is quite restrictive. Therefore, **drm** allows for five extensions. First, the assumption of constant weights may be relaxed. Weights may be made specific to a respondent's value on *rowvar* or *colvar*, i.e. specific to values of i or j. Thus, it is possible to estimate one of:

```
y[i,j,k] = p[i]*mu[i,i] + q[i]*mu[j,j] + e[i,j,k] (2)
```

or

```
y[i,j,k] = p[j]*mu[i,i] + q[j]*mu[j,j] + e[i,j,k] (3)
```

Second, any number of covariates may be entered linearly. Extending (2), this yields:

```
y[i,j,k] = p[i]*mu[i,i] + q[i]*mu[j,j] + XB + e[i,j,k] (4)
```

Where X is a vector of covariates and B a vector of parameters.

Third, mu[i,i] and mu[j,j] may be replaced with mu[i,i,c] and mu[j,j,c]. In other words, estimated population means on the diagonal may be specific to some (set of) variable(s) *byvar* that is indexed by c. This may be useful when one has data with multiple levels (e.g. persons nested in countries) and would like to have mobility tables be specific to each country c.

Building on (4), this extension yields:

```
y[i,j,c,k] = p[i]*mu[i,i,c] + q[i]*mu[j,j,c] + XB + e[i,j,c,k] (5)
```

Currently, this option is only supported with least-squares estimation.

Fourth, weights p[i] and q[i] may be interacted with a linear combination of variables XB_inter. As an extension of (2), this yields:

```
y[i,j,k] = (p[i]+(XB_inter))*mu[i,i] + (q[i]-(XB_inter))*mu[j,j] + e[i,j,k] (6)
```

This extension follows e.g. De Graaf, Nieuwbeerta, Heath (1995).

Fifth, in cases where *depvar* is binary, it may be useful to estimate a logit or probit variant of the diagonal reference model. Thus, users may estimate:

```
pr(y[i,j,k]=1)=logistic(drm)
```

or

```
pr(y[i,j,k]=1)=normal(drm)
```

for the logit or probit link, respectively. Here, logistic(x)=1/(1+e^-x) and normal(x) is the cdf of the normal distribution. Moreover, drm=p[i]*mu[i,i] + q[i]*mu[j,j] + XB + e[i,j,k], or one of the other variants described above.

**Options**

General

**vce(***vcetype***)** set standard error type. See vce_option, nl, and ml for options.

**wgt(***str***)** one of **cons**, **row** or **col**. Specifies if weights are assumed to be constant (default) or dependent on *rowvar* (**row**) or *colvar* (**col**). See equations (2) and (3) in the description.

**intervars(***varlist***)** interact weights p and q with *varlist*. See equation (6) in the description.

**iterate(**#**)** specifies the maximum number of iterations; default is **iterate(1000)**

**level(**#**)** set confidence level; default is **level(95)**

**coeflegend** specifies that the legend of the coefficients and how to specify them in an expression be displayed rather than displaying the statistics for the coefficients.

**keep** prevents **drm** from deleting dummies for each level of *rowvar* and *colvar* that were generated for estimation.

**old** forces **drm** to behave as it did until version 0.4, i.e. to generate illegible temporary variable names for *rowvar* and *colvar*.

Maximum likelihood

N.b. When **nl** is specified, all maximum likelihood options are ignored. See description.

**link(***str***)** one of **linear**, **logit** or **probit**. Specifies link function; default is **link(***linear***)**. Using **link(***linear***)** or specifying **nl** gives equivalent results, though the resulting output will look somewhat different. See difference between nl and ml.

**sobel** implements variants of the method documented in appendix A of Sobel (1985) to find initial values; the default.

**classic** uses (1/R)*(depvar[i,i]}/(depvar[1,1]+...+depvar[R,R])) as initial values for mu[i,i] and 0.5 as initial values for p.

**alternative** uses exp((1/R)*(depvar[i,i]}/(depvar[1,1]+...+depvar[R,R]))) as initial values for mu[i,i] and 0.5 as initial values for p.

**tech()** specifies maximization algorithm. Default is **nr**. Alternatives are **bhhh**, **dfp** and **bfgs**. This option may help when convergence can't be achieved with the default settings. See maximize for further help.

**difficult** specifies **difficult** option for ml. This option may help when convergence can't be achieved with the default settings. See maximize for further help.

**ownconstrains(***str***)** specifies further user-written constrains. Syntax is **[***exp* **=** *exp***]** [**[***exp* **=** *exp***]** ...], where exp typically contains: **[**eq_name**]***varname*. A typical use of **ownconstrains(***str***)** is to constrain weights to lie on the unit interval. Say we fitted a model and found p, i.e. the weight on *rowvar*, to be greater than 1:

```
. drm depvar rowvar colvar control1 control2, link(linear)
```

To force p=1, we specify a constraint as such:

```
. drm depvar rowvar colvar control1 control2, link(linear) ownc([p]_cons=1)
```

If we wanted additional constraints, e.g. **control1**=**control2** we could write:

```
. drm depvar rowvar colvar control1 control2, link(linear) ownc([p]_cons=1
[xb]control1=[xb]control2)
```

Least squares

N.b. When **nl** is not specified, these options are ignored. See introduction.

**by(***varlist***)** specify variables on which estimates of mu[i,i] and mu[j,j] are made conditional. If more than one variable is specified, every combination of varlist is taken. See equation (5) in the description.

**constrain** explicitly constrains p to lie on [0,1]. This is achieved by replacing parameter p in e.g. equation (2) with exp(gamma/(1+gamma)), where gamma is a parameter to be estimated and exp(.) is the exponential function. If specified, parameter estimates for p and q are obtained using nlcom.

**Difference between nl and ml estimation**

The model of equation (1) may be equivalently rewritten as:

y[i,j,k] = alpha + p*mu[i,i] + q*mu[j,j] + e[i,j,k] (1a)

Here, alpha is a constant and the constraint mu[1,1]+...+mu[R,R]=0 is set. When **nl** is not specified and **drm** thus uses maximum likelihood, (variants of) equation (1a) are estimated. When **nl** is specified and hence non-linear least squares are used, **drm** estimates (variants of) equation (1).

**Finding overall weights when intervars option is used**

Note that when **intervars(***intervars***)** is used, parameters p and q only give the overall weights on mu[i,i] and mu[j,j] when all variables in *intervars* are zero. To find e.g. the overall weight on mu[i,i] for other values of variables x1,...,xn in *intervars*, type:

```
. lincom (_b[p:_cons]+(_b[rho:x1]*x1+...+_b[rho:xn]*xn))
```

When p and q are made specific to levels of i (or j), to find e.g. p[2], just write:

```
. lincom (_b[p2:_cons]+(_b[rho:x1]*x1+...+_b[rho:xn]*xn))
```

Concretely, suppose we estimated a model like this:

```
. drm depvar rowvar colvar, wgt(col) intervars(intervar1 intervar2 intervar3)
```

To find the overall weight on rowvar when e.g. rowvar=3, intervar1=3, intervar2=5, intervar3=12, we must write:

```
. lincom
(_b[p3:_cons]+(_b[rho:intervar1]*3+_b[rho:intervar2]*5+_b[rho:intervar3]*12))
```

You may find it useful to use the **coeflegend** option to display the names of parameters as they need to be referred to in postestimation commands like lincom.

**References**

De Graaf, N.D.; Nieuwbeerta, P.; Heath, A. (1995). Class Mobility and Political Preferences: Individual and Contextual Effects. The American Journal of Sociology, 100(4), 997-1027.

Lizardo, O. (2007). Gaussian, Logit, Probit and Poisson Diagonal Reference models.

Sobel, M. (1981). Diagonal Mobility Models: A Substantively Motivated Class of Designs for the Analysis of Mobility Effects. American Sociological Review, 46(6), 893-906.

Sobel, M. (1985). Social Mobility and Fertility Revisited: Some New Models for the Analysis of the Mobility Effects Hypothesis. American Sociological Review, 50(5), 699-712.

**Author/Citation**

Caspar Kaiser
Department of Social Policy and Intervention
Nuffield College, University of Oxford
caspar.kaiser@nuffield.ox.ac.uk

If you use **drm** for your research, please cite:
Kaiser, C. (2018). DRM Diagonal Reference Model Stata. Open Science Framework. doi:10.17605/OSF.IO/KFDP6.
or the suggested RePEc entry.

## Feedback

**drm** will be updated. Any feedback or questions are more than welcome. If you have ideas for additional features (or would be interested in adding any), please feel free to contact me.

## Planned features:

-allow **wgt()** when using ml
-3-dimensional or N-dimensional mobility tables
-full compatibility with predict and margins
-multinomial logit
-ordered logit/probit
-random effects

## New in version 0.4:

-parameter q is now explicitly estimated when using ml. This fixes repeated convergence problems.
-ml estimation is now the default
-user-written constrains are now allowed
-Sobel's (1985) method to find initial values is now implemented and set to be the default. This speeds up estimation considerably and helps with convergence.

## New in version 0.5:

-parameter estimates for each level of *rowavar* and *colvar* are now displayed in legible form and associated dummies are (optionally) saved.
-some users found the display of the ancillary paramter sigma when using the linear link fucntion confusing. This parameter estimate is no longer displayed.