Creating plots and tables of estimation results using `parmest` and friends

Roger Newson (King's College, London, UK)

`roger.newson@kcl.ac.uk`

- Why save estimation results?
- The `parmest` package: `parmest` and `parmby`.
- Creating confidence interval plots using `descsave` and `factext`.
- Concatenating multiple analysis results using `dsconcat`.
- Plotting *P*-values using `smileplot`.

Why save estimation results?

- Statisticians make their living by producing confidence intervals and P-values.
- Unfortunately, the confidence intervals and P-values in a Stata log are in no fit state for delivery to an end user.
- At the very least, they need to be formatted and tabulated to be fit for publication.
- And, for immediate impact, it is even better to plot them.
- Former SAS users in particular are accustomed to being able to produce output data sets, and want to do the same in Stata.

The `parmest` package: `parmest` and `parmby`

- My first response to this problem was `parmest`, which saves the results of the most recent Stata estimation command as a data set with 1 observation per parameter and data on parameter names, labels, estimates, confidence limits and P-values.

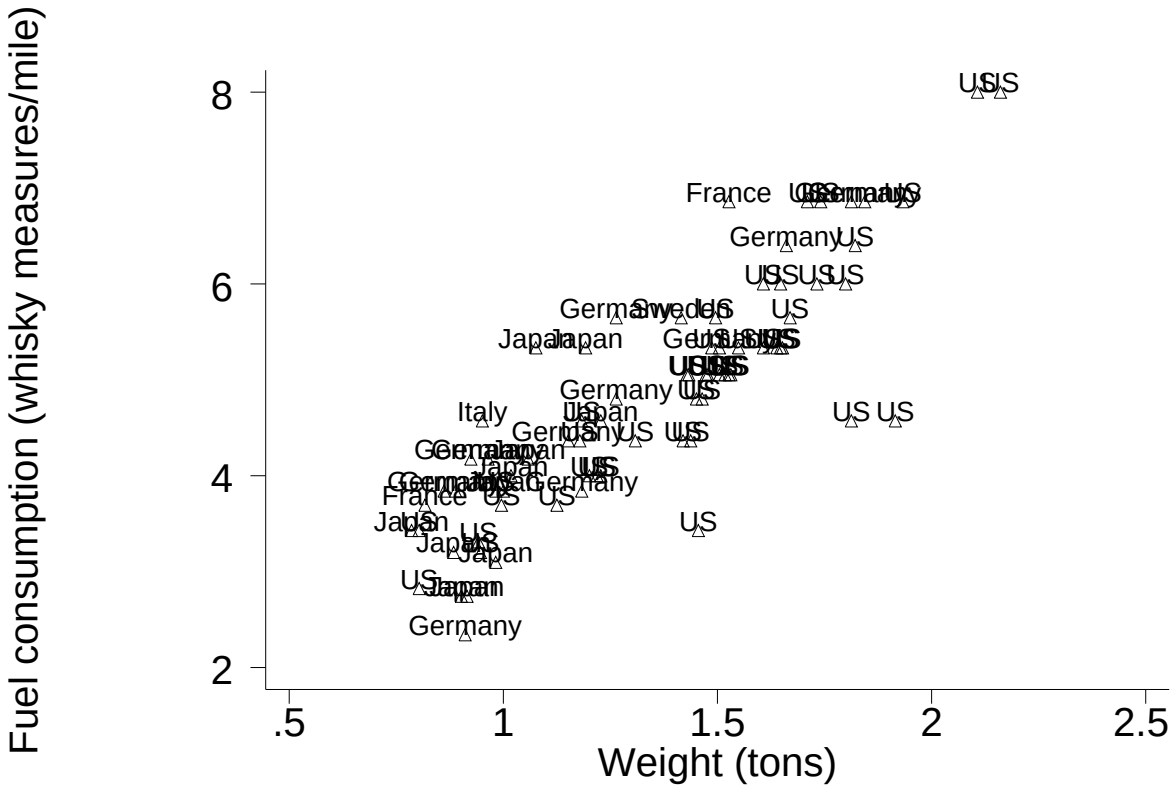
- Nowadays, I use `parmby`, a "quasi-byable" front end to `parmest` (although the `by` option is not compulsory). `parmby` calls a Stata estimation command (such as `regress`), and creates an output data set with 1 observation per parameter or 1 observation per parameter per by-group, and data on a wide range of estimation results.

- `parmby` is therefore like `statsby`, except that it creates an observation *per parameter* per by-group, instead of an observation per by-group.

- The data set created by `parmest` or `parmby` can be saved to disk, stored in memory (overwriting the pre-existing data), or both.

Fuel consumption and weight by country for cars in the `auto` data

- In the `auto` data, we define a numeric variable `country`, encoding a car's country of origin.
- We also define two variables `wmpm` (fuel consumption in whisky measures per mile) and `tons` (weight in tons).
- The graph plots `wmpm` against `tons`, labelling data points by `country`.



An example program using `parmby`

The following program uses `parmby` to fit a regression model of fuel consumption with respect to weight and country of origin, and to store the results in memory, overwriting the existing data. It then specifies a sensible format for the confidence intervals, describes the data set, and lists the confidence intervals.

```
parmby "xi:regress wmpm tons i.country,nohead",label;
format estimate min95 max95 %8.2f;
describe;
list parm label estimate min95 max95,noobs;
```

Output of the example program (1)

`parmby` calls `regress`, prints the results in the usual Stata log format, and saves them.

```
. parmby "xi:regress wmpm tons i.country,nohead",label;
Command: xi:regress wmpm tons i.country,nohead
i.country          _Icountry_1-6       (naturally coded; _Icountry_1 omitted)
------------------------------------------------------------------------------
        wmpm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        tons |   3.400347   .2313002    14.70   0.000      2.93867    3.862024
 _Icountry_2 |    .549941   .1911327     2.88   0.005     .1684385    .9314435
 _Icountry_3 |   .4380153   .2245559     1.95   0.055    -.0102001    .8862307
 _Icountry_4 |   1.280447    .428687     2.99   0.004     .4247846     2.13611
 _Icountry_5 |   1.328566   .6041275     2.20   0.031     .1227229     2.53441
 _Icountry_6 |   .8254644   .5920764     1.39   0.168     -.356325    2.007254
       _cons |   .0094969   .3515232     0.03   0.979    -.6921464    .7111402
------------------------------------------------------------------------------
```

Note that cars typically consume 2.94 to 3.86 extra whisky measures of petrol per additional ton-mile. The other parameters are country effects and an intercept (in whisky measures/mile).

Output of the example program (2)

The data set created by `parmby` has one observation per parameter, and variables as shown:

```
. describe;
Contains data from C:_06002t.tmp
  obs:             7
 vars:             9                          18 Apr 2002 15:11
 size:           539 (99.3% of memory free)
------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
parmseq         byte   %12.0g                 Parameter sequence number
parm            str11  %11s                   Parameter name
label           str13  %13s                   Parameter label
estimate        double %8.2f                  Parameter estimate
stderr          double %10.0g                 SE of parameter estimate
t               double %10.0g                 t-test statistic
p               double %10.0g                 P-value
min95           double %8.2f                  Lower 95% confidence limit
max95           double %8.2f                  Upper 95% confidence limit
------------------------------------------------------------------------------
Sorted by:  parmseq
```

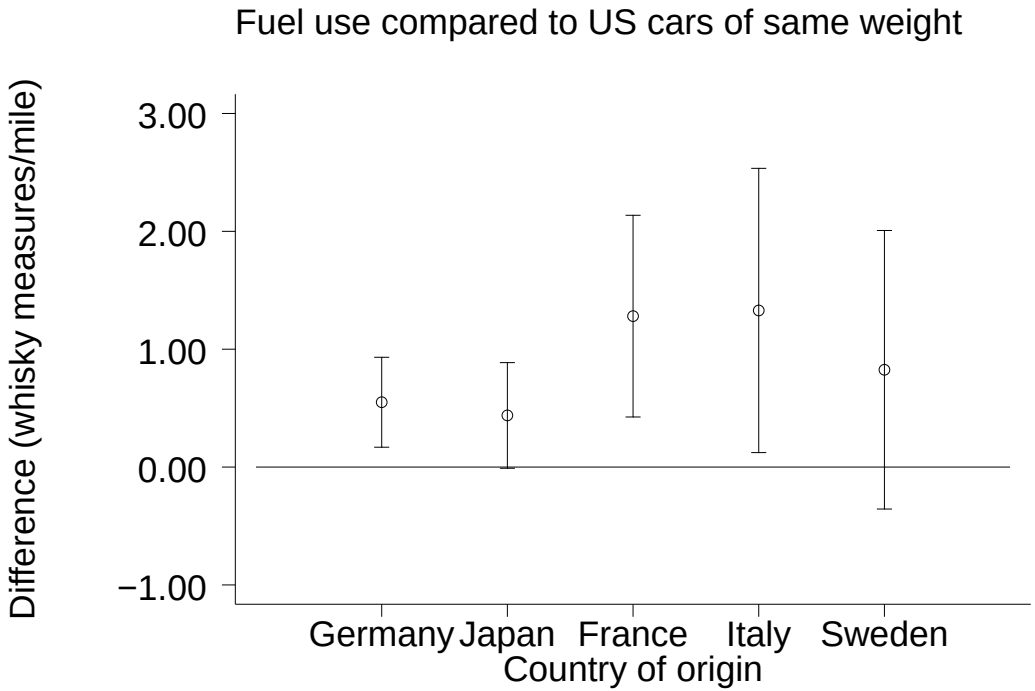
Output of the example program (3)

The parameters are listed with labels and formats, and are a bit more user-friendly than before. The variable `label` contains the variable label of the X-variable corresponding to the parameter, which may be a dummy variable created by `xi`.

```
. list parm label estimate min95 max95,noobs;
       parm           label  estimate     min95     max95
       tons  Weight (tons)      3.40      2.94      3.86
_Icountry_2     country==2      0.55      0.17      0.93
_Icountry_3     country==3      0.44     -0.01      0.89
_Icountry_4     country==4      1.28      0.42      2.14
_Icountry_5     country==5      1.33      0.12      2.53
_Icountry_6     country==6      0.83     -0.36      2.01
      _cons                     0.01     -0.69      0.71
```

We could cut and paste these results into a Word (or TEX) table, possibly using tools such as `outsheet` (official Stata), or `ciform` and/or `listtex` (downloadable from SSC). *However* ...

- . . . it would be better if we knew, at a glance, which dummy variable belonged to which country.
- And it would be even better if we could plot the confidence intervals, instead of just tabulating them.
- This graph shows the expected differences in fuel consumption between non-US cars and US cars of the same weight.



Creating confidence interval plots using `descsave` and `factext`

- `descsave` is an extension of official Stata's `describe`. It lists variable attributes (types, formats, variable labels and value labels), and produces output files.

- One of these output files is a Stata do-file, which reconstructs these attributes when called, if variables exist with the same names and modes (numeric or string).

- `factext` is a program which can read factor values from string variables (such as `label` in the `parmby` output) containing `xi`-style dummy variable labels. It may run a do-file created by `descsave` to reconstruct the variable attributes of the factors.

- If `parmby` is used together with `xi`, `descsave` and `factext`, then factors in the input data set, used with `xi`, can be reconstructed in the output data set, with values from the dummy variable labels created by `xi`. These factors can then be used in confidence interval plots and tables.

A simple program using `descsave`, `parmby`, `xi` and `factext`

This program uses `descsave` to save the attributes of the variable `country` to a temporary do-file, then uses `parmby` and `xi` to carry out the same regression analysis as before, then uses `factext` to reconstruct the variable `country` in the output data set using the temporary do-file, then lists the confidence intervals, and finally produces the CI plot that we saw earlier.

```
tempfile tf0;
descsave country,do(‘tf0’);
parmby "xi:regress wmpm tons i.country,nohead",label;
factext,do(‘tf0’);
format estimate min95 max95 %8.2f;
list parm label country estimate min95 max95,noobs;
grap estimate min95 max95 country,
 xsc(1,7) xlab(2(1)6) ylab ylin(0) s(O..) c(.II)
 t1("Fuel use compared to US cars of same weight")
 l1("Difference (whisky measures/mile)");
```

Output data set created using `descsave`, `parmby`, `xi` and `factext`

This time, the output data set contains a new variable `country`, similar to the one in the input data set. This was reconstructed by `factext`, using dummy variable labels stored in the variable `label` and the do-file stored by `descsave`.

```
. list parm label country estimate min95 max95,noobs;
```

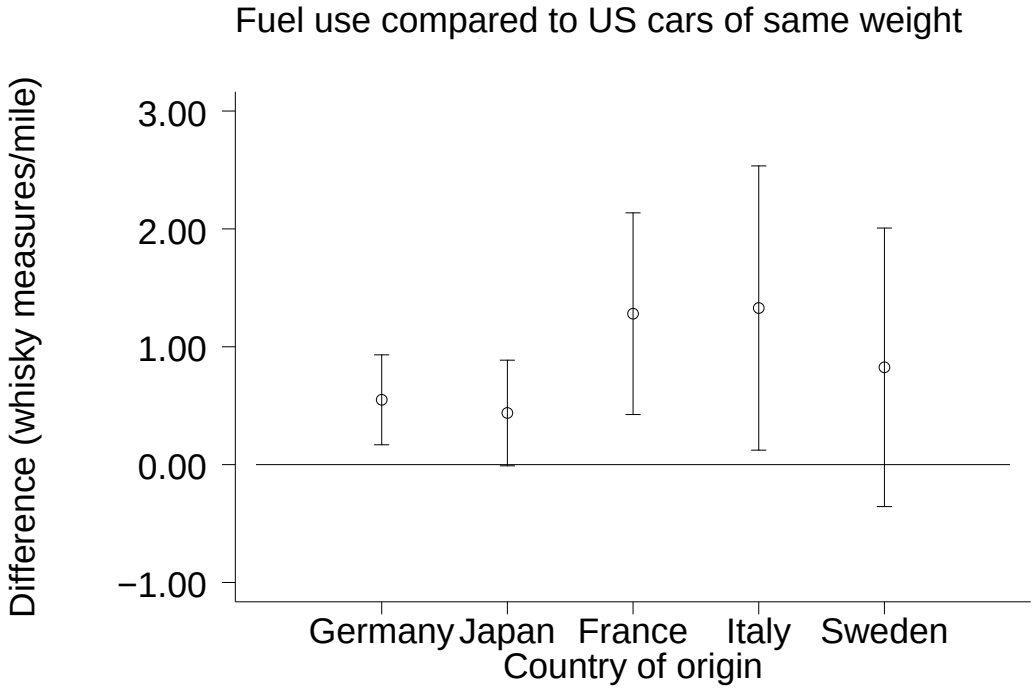
parm	label	country	estimate	min95	max95
tons	Weight (tons)	.	3.40	2.94	3.86
_Icountry_2	country==2	Germany	0.55	0.17	0.93
_Icountry_3	country==3	Japan	0.44	-0.01	0.89
_Icountry_4	country==4	France	1.28	0.42	2.14
_Icountry_5	country==5	Italy	1.33	0.12	2.53
_Icountry_6	country==6	Sweden	0.83	-0.36	2.01
_cons		.	0.01	-0.69	0.71

Plot from the output data set created using `descsave`, `parmby`, `xi` and `factext`

- Finally, the confidence interval variables `estimate`, `min95` and `max95` are plotted against the reconstructed variable `country`.

- Note that the variable and value labels for `country` were automatically reconstructed by `factext`, and did not have to be restated anywhere in the program.

- Therefore, if we change the variable and value labels in the `auto` data and re-run our program, then the changes will appear automatically in the graph.



Using `parmby` and `dsconcat` to save multiple analyses

- Usually, we do multiple analyses in a Stata do-file, instead of just one as in the previous examples. So we would like to use the original data set a few times before finally overwriting it.

- Fortunately, `parmby` output data sets can be saved to disk using the `saving` option, leaving the original data intact. So multiple calls to `parmby` can produce multiple output files, possibly temporary.

- These multiple output files can be concatenated into the memory to form one long data set, using the program `dsconcat`.

- In this long data set, we want to know which analysis each fitted parameter belongs to. `parmby` can help us by creating numeric and string identifier variables `idnum` and `idstr` in the output data set for each analysis.

A program using `parmby` and `dsconcat`

This program carries out unadjusted and adjusted regression analyses of fuel consumption with respect to weight and US origin. It uses `parmby` to save the results of each analysis in a temporary output file (identified by values of the variables `idnum` and `idstr`), and then concatenates the output files using `dsconcat`:

```
tempfile tf1 tf2 tf3;
parmby "regress wmpm tons,nohead",label idnum(1) idstr(Unadj.)
 saving(‘tf1’,replace);
parmby "regress wmpm us,nohead",label idnum(2) idstr(Unadj.)
 saving(‘tf2’,replace);
parmby "regress wmpm tons us,nohead",label idnum(3) idstr(Adj.)
 saving(‘tf3’,replace);
dsconcat ‘tf1’ ‘tf2’ ‘tf3’;
format estimate min95 max95 %8.2f;
sort idnum idstr parmseq;
by idnum idstr:list parm label estimate min95 max95,noobs;
```

Output data set created by `dsconcat` from multiple `parmby` outputs

There is one observation per parameter. The different analyses are identified by the variables `idnum` (numeric ID) and `idstr` (string ID).

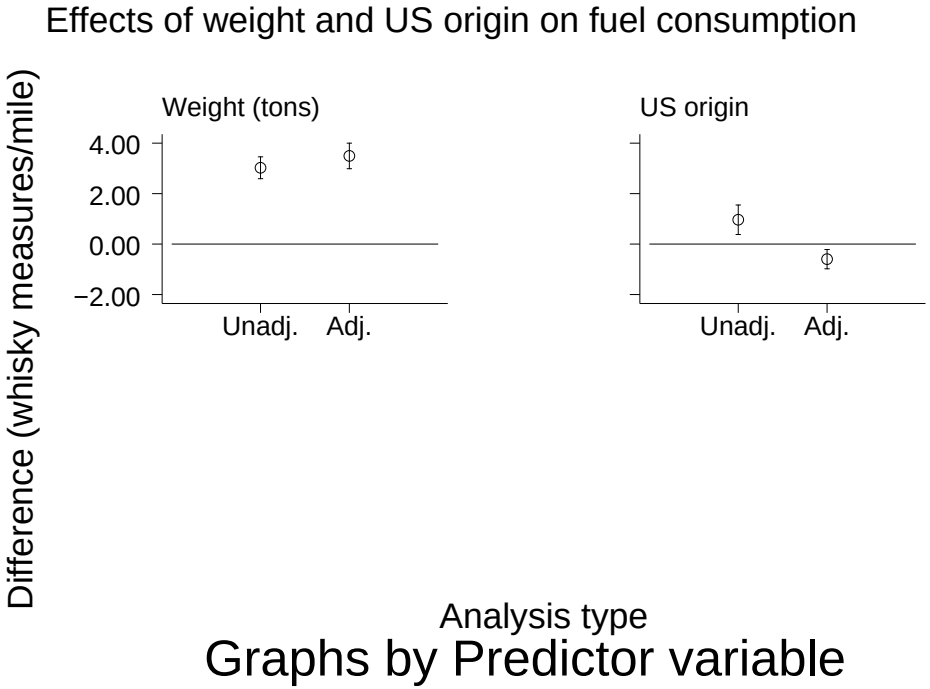
```
. by idnum idstr:list parm label estimate min95 max95,noobs;

-------------------------------------------------------------------------------
-> idnum = 1, idstr = Unadj.
     parm          label  estimate     min95     max95
     tons  Weight (tons)      3.03      2.59      3.46
    _cons                     0.74      0.14      1.34

-------------------------------------------------------------------------------
-> idnum = 2, idstr = Unadj.
     parm          label  estimate     min95     max95
       us      US origin      0.97      0.38      1.55
    _cons                     4.14      3.65      4.63

-------------------------------------------------------------------------------
-> idnum = 3, idstr = Adj.
     parm          label  estimate     min95     max95
     tons  Weight (tons)      3.50      2.99      4.00
       us      US origin     -0.60     -0.98     -0.21
    _cons                     0.53     -0.06      1.11
```

Plot from the output data set created using `parmby` and `dsconcat`

- With a few more lines of Stata code, we can create these plots from the data in the long data set created by `dsconcat`.

- Note that each plot contains parameters from two analyses (unadjusted and adjusted).

- We see that US cars consume more fuel per mile than non-US cars, but less than non-US cars of the same weight.



Plotting P-values using `smileplot`

- As well as saving estimates and confidence limits, `parmby` also saves P-values. (And many other estimation results, if requested by the user.)

- This is very useful if we are carrying out multiple analyses, and we want to know whether a result is still "significant" as one result out of many.

- The program `smileplot` is used on data sets created using `parmby`. It plots P-values on the Y-axis against parameter estimates on the X-axis.

- The P-values are plotted on a reverse log scale. (So, the higher they are, the more significant they are.)

Example: Red blood cell fatty acid composition and oily fish consumption in pregnant women (ALSPAC study, Bristol University)

- 4720 pregnant women contributed 1-6 blood samples each, and also reported current fish consumption on a food frequency questionnaire.

- The blood samples were assayed, using chromatography, for composition of the red blood cell membrane (40 different fatty acids as a percent of total fatty acids).

- Consumption of oily fish (eg mackerel) was reported as never/rarely, once per fortnight, 1-3 times per week, or over 3 times per week.

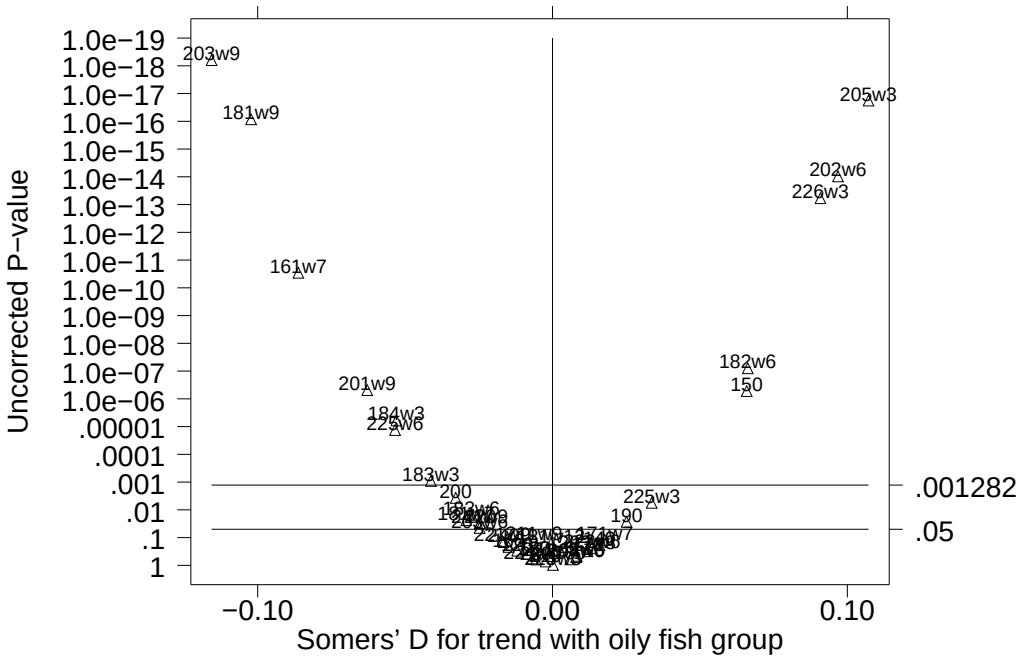
- The association of each fatty acid percentage with reported oily fish consumption was measured using Somers' D, which is the difference between two probabilities. Given two samples from women with different reported oily fish consumption, these are the probability that the woman consuming *more* oily fish had the higher fatty acid percentage, and the probability that the woman consuming *less* oily fish had the higher fatty acid percentage.

Smile plot of Somers' D for fatty acid percentages with respect to oily fish consumption group

- The data points are 40 individual fatty acid percentages, each with a Somers' D estimate for trend with oily fish group.

- The X-axis reference line indicates a Somers' D of zero under the null hypothesis.

- The Y-axis reference lines indicate the P-value thresholds, uncorrected (lower) and Sidak-corrected for 40 parameter estimates (upper). The upper reference line is called the **parapet line**.



Unofficial Stata packages mentioned or used in this presentation

These packages are all downloadable from SSC using the `ssc` command (see `help ssc`).

Package	**Description**
`ciform`	Format three numeric variables as a confidence interval for tabulation
`descsave`	Extension of `describe`, producing output files
`dsconcat`	Concatenate a list of Stata data files into the memory
`factext`	Extract values of factors from string variables, eg `label` in a `parmby` output
`listtex`	Output variables to a file to be inserted into a general TEX or HTML table
`parmest`	Save estimation results as a data set with 1 obs. per parameter (includes `parmby`)
`sencode`	Extension of `encode`, with string values coded in order of appearance
`smileplot`	Create a smile plot of P-values against parameter estimates