`parmest` and extensions

With examples from the Avon Longitudinal Study of Parents and Children (ALSPAC) cohort study at Bristol University, UK
http://www.bristol.ac.uk/alspac/

Roger B. Newson
r.newson@imperial.ac.uk
http://www.imperial.ac.uk/nhli/r.newson/

National Heart and Lung Institute
Imperial College London

14th UK Stata Users' Group Meeting, 8–9 September, 2008
Downloadable from the conference website at
http://ideas.repec.org/s/boc/usug08.html

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

`parmest` and its resultssets

- The `parmest` package (Newson, 2003)[3] has been downloadable from SSC since 1998.
- It contains 4 modules: the original `parmest`, and the additional `parmby`, `metaparm` and `parmcip`.
- These are used to produce output datasets (or resultssets), with 1 observation for each of a set of estimated statistical parameters.
- The essential variables are the parameter identification variables, and the parameter estimate, standard error, and (optionally) degrees of freedom.
- Usually, there are also derived variables, containing confidence limits, t– or z–test statistics, and P-values.
- And other variables may also be added, containing other parameter attributes.

Creating a typical `parmest` resultsset

In the `auto` data, a regression model is fitted, predicting car weight from repair record. The `parmest` command creates a resultsset in memory, overwriting the original data.

```
. sysuse auto, clear;
(1978 Automobile Data)

. xi: regress weight i.rep78, nohead;
i.rep78           _Irep78_1-5         (naturally coded; _Irep78_1 omitted)
------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Irep78_2 |     253.75   574.2138     0.44   0.660    -893.3739    1400.874
   _Irep78_3 |         199    530.436     0.38   0.709    -860.6676    1258.668
   _Irep78_4 |        -230   541.3739    -0.42   0.672    -1311.519    851.5187
   _Irep78_5 |  -777.2727   558.3338    -1.39   0.169    -1892.673    338.1273
       _cons |        3100   513.5924     6.04   0.000     2073.981    4126.019
------------------------------------------------------------------------------

. parmest, label norestore escal(N);
```

A typical `parmest` resultsset: Essential variables

These are the parameter name, estimate, standard error, and degrees of freedom.

```
. list parm estimate stderr dof, clean noobs;

      parm     estimate      stderr   dof
  _Irep78_2       253.75   574.21376    64
  _Irep78_3          199   530.43595    64
  _Irep78_4         -230   541.37392    64
  _Irep78_5   -777.27273    558.3338    64
     _cons         3100    513.5924    64
```

A typical `parmest` resultsset: Test statistics and *P*–values

The variables `t` and `p` are derived from the estimate, standard error, and degrees of freedom.

```
. list parm estimate stderr dof t p, clean noobs;
```

parm	estimate	stderr	dof	t	p
_Irep78_2	253.75	574.21376	64	.4419086	.660045
_Irep78_3	199	530.43595	64	.37516311	.70877962
_Irep78_4	-230	541.37392	64	-.42484499	.67237463
_Irep78_5	-777.27273	558.3338	64	-1.3921291	.16870208
_cons	3100	513.5924	64	6.0359149	8.796e-08

A typical `parmest` resultsset: 95 percent confidence limits

The variables `min95` and `max95` are also derived from the estimate, standard error, and degrees of freedom.

```
. list parm estimate stderr dof min95 max95, clean noobs;
```

parm	estimate	stderr	dof	min95	max95
_Irep78_2	253.75	574.21376	64	-893.37386	1400.8739
_Irep78_3	199	530.43595	64	-860.66763	1258.6676
_Irep78_4	-230	541.37392	64	-1311.5187	851.51874
_Irep78_5	-777.27273	558.3338	64	-1892.6727	338.12727
_cons	3100	513.5924	64	2073.9812	4126.0188

A typical `parmest` resultsset: Optional extras

`label` contains the parameter variable label. `es_1` contains the estimation result `e(N)` (the sample number).

```
. list parm estimate stderr dof label es_1, clean noobs;
```

parm	estimate	stderr	dof	label	es_1
_Irep78_2	253.75	574.21376	64	rep78==2	69
_Irep78_3	199	530.43595	64	rep78==3	69
_Irep78_4	-230	541.37392	64	rep78==4	69
_Irep78_5	-777.27273	558.3338	64	rep78==5	69
_cons	3100	513.5924	64	Constant	69

A typical `parmest` resultsset: Summary of variables

The SSC package `descsave` is an extended version of `describe`. Here, it is used to list the variables of the `parmest` resultsset.

```
. descsave, list(name type varlab, clean noobs subvar abbr(32));

    variable name   storage type   variable label
    parm            str9           Parameter name
    label           str8           Parameter label
    estimate        double         Parameter estimate
    stderr          double         SE of parameter estimate
    dof             byte           Degrees of freedom
    t               double         t-test statistic
    p               double         P-value
    min95           double         Lower 95% confidence limit
    max95           double         Upper 95% confidence limit
    es_1            byte           e(N)
```

What use are `parmest` resultssets?

- `parmest` resultssets can be saved to memory (overwriting the original dataset) and/or listed and/or saved to disk files and/or subsetted and/or merged and/or concatenated with other resultssets, creating new resultssets.
- They can also be used to create tables of results, using the SSC package `listtex`.
- *However*, a principal advantage of resultssets is that the results can be plotted, using Stata graphics programs.
- *Also*, a resultsset with multiple *P*–values (possibly from a genome scan) can be input into a selection of multiple–test procedures, using the SSC package `smileplot` (Newson *et al.*, 2003)[2].
- *Also*, we can derive resultssets containing results for linear combinations and transformations of parameters, using the `metaparm` and `parmcip` modules of `parmest`.

What use are `parmest` resultssets?

- `parmest` resultssets can be saved to memory (overwriting the original dataset) and/or listed and/or saved to disk files and/or subsetted and/or merged and/or concatenated with other resultssets, creating new resultssets.
- They can also be used to create tables of results, using the SSC package `listtex`.
- *However*, a principal advantage of resultssets is that the results can be plotted, using Stata graphics programs.
- *Also*, a resultsset with multiple *P*–values (possibly from a genome scan) can be input into a selection of multiple–test procedures, using the SSC package `smileplot` (Newson *et al.*, 2003)[2].
- *Also*, we can derive resultssets containing results for linear combinations and transformations of parameters, using the `metaparm` and `parmcip` modules of `parmest`.

What use are `parmest` resultssets?

- `parmest` resultssets can be saved to memory (overwriting the original dataset) and/or listed and/or saved to disk files and/or subsetted and/or merged and/or concatenated with other resultssets, creating new resultssets.
- They can also be used to create tables of results, using the SSC package `listtex`.
- *However*, a principal advantage of resultssets is that the results can be plotted, using Stata graphics programs.
- *Also*, a resultsset with multiple *P*–values (possibly from a genome scan) can be input into a selection of multiple–test procedures, using the SSC package `smileplot` (Newson *et al.*, 2003)[2].
- *Also*, we can derive resultssets containing results for linear combinations and transformations of parameters, using the `metaparm` and `parmcip` modules of `parmest`.

What use are `parmest` resultssets?

- `parmest` resultssets can be saved to memory (overwriting the original dataset) and/or listed and/or saved to disk files and/or subsetted and/or merged and/or concatenated with other resultssets, creating new resultssets.
- They can also be used to create tables of results, using the SSC package `listtex`.
- *However*, a principal advantage of resultssets is that the results can be plotted, using Stata graphics programs.
- *Also*, a resultsset with multiple *P*–values (possibly from a genome scan) can be input into a selection of multiple–test procedures, using the SSC package `smileplot` (Newson *et al.*, 2003)[2].
- *Also*, we can derive resultssets containing results for linear combinations and transformations of parameters, using the `metaparm` and `parmcip` modules of `parmest`.

What use are `parmest` resultssets?

- `parmest` resultssets can be saved to memory (overwriting the original dataset) and/or listed and/or saved to disk files and/or subsetted and/or merged and/or concatenated with other resultssets, creating new resultssets.
- They can also be used to create tables of results, using the SSC package `listtex`.
- *However*, a principal advantage of resultssets is that the results can be plotted, using Stata graphics programs.
- *Also*, a resultsset with multiple *P*–values (possibly from a genome scan) can be input into a selection of multiple–test procedures, using the SSC package `smileplot` (Newson *et al.*, 2003)[2].
- *Also*, we can derive resultssets containing results for linear combinations and transformations of parameters, using the `metaparm` and `parmcip` modules of `parmest`.

What use are `parmest` resultssets?

- `parmest` resultssets can be saved to memory (overwriting the original dataset) and/or listed and/or saved to disk files and/or subsetted and/or merged and/or concatenated with other resultssets, creating new resultssets.
- They can also be used to create tables of results, using the SSC package `listtex`.
- *However*, a principal advantage of resultssets is that the results can be plotted, using Stata graphics programs.
- *Also*, a resultsset with multiple *P*–values (possibly from a genome scan) can be input into a selection of multiple–test procedures, using the SSC package `smileplot` (Newson *et al.*, 2003)[2].
- *Also*, we can derive resultssets containing results for linear combinations and transformations of parameters, using the `metaparm` and `parmcip` modules of `parmest`.

Example from the ALSPAC cohort: 110 FFQ diet–disease associations

- In the Avon Longitudinal Study of Parents and Children (ALSPAC), mothers of 12008 children completed a food frequency questionnaire (FFQ) on their diet during pregnancy.
- 5 dietary pattern scores were derived from the FFQ data, using principal component analysis (Northstone *et al.*, 2008)[4].
- Associations were measured (using linear or logistic regression) with 22 child disease outcomes, implying $22 \times 5 = 110$ associations.
- Each of these associations was measured with adjustment for 3 nested confounder sets, described as "None" (unadjusted), "All" (adjusted for the full list of confounders), and "Non-causal" (adjusted for a restricted subset of confounders).
- The confidence intervals, *t*– or *z*–statistics, and *P*–values were saved in `parmest` resultssets, plotted, and entered into multiple–test procedures.

Example from the ALSPAC cohort: 110 FFQ diet–disease associations

- In the Avon Longitudinal Study of Parents and Children (ALSPAC), mothers of 12008 children completed a food frequency questionnaire (FFQ) on their diet during pregnancy.
- 5 dietary pattern scores were derived from the FFQ data, using principal component analysis (Northstone *et al.*, 2008)[4].
- Associations were measured (using linear or logistic regression) with 22 child disease outcomes, implying $22 \times 5 = 110$ associations.
- Each of these associations was measured with adjustment for 3 nested confounder sets, described as "None" (unadjusted), "All" (adjusted for the full list of confounders), and "Non-causal" (adjusted for a restricted subset of confounders).
- The confidence intervals, t– or z–statistics, and P–values were saved in `parmest` resultssets, plotted, and entered into multiple–test procedures.

Example from the ALSPAC cohort: 110 FFQ diet–disease associations

- In the Avon Longitudinal Study of Parents and Children (ALSPAC), mothers of 12008 children completed a food frequency questionnaire (FFQ) on their diet during pregnancy.
- 5 dietary pattern scores were derived from the FFQ data, using principal component analysis (Northstone *et al.*, 2008)[4].
- Associations were measured (using linear or logistic regression) with 22 child disease outcomes, implying $22 \times 5 = 110$ associations.
- Each of these associations was measured with adjustment for 3 nested confounder sets, described as "None" (unadjusted), "All" (adjusted for the full list of confounders), and "Non-causal" (adjusted for a restricted subset of confounders).
- The confidence intervals, t– or z–statistics, and P–values were saved in `parmest` resultssets, plotted, and entered into multiple–test procedures.

Example from the ALSPAC cohort: 110 FFQ diet–disease associations

- In the Avon Longitudinal Study of Parents and Children (ALSPAC), mothers of 12008 children completed a food frequency questionnaire (FFQ) on their diet during pregnancy.
- 5 dietary pattern scores were derived from the FFQ data, using principal component analysis (Northstone *et al.*, 2008)[4].
- Associations were measured (using linear or logistic regression) with 22 child disease outcomes, implying $22 \times 5 = 110$ associations.
- Each of these associations was measured with adjustment for 3 nested confounder sets, described as "None" (unadjusted), "All" (adjusted for the full list of confounders), and "Non-causal" (adjusted for a restricted subset of confounders).
- The confidence intervals, t– or z–statistics, and P–values were saved in `parmest` resultssets, plotted, and entered into multiple–test procedures.

Example from the ALSPAC cohort: 110 FFQ diet–disease associations

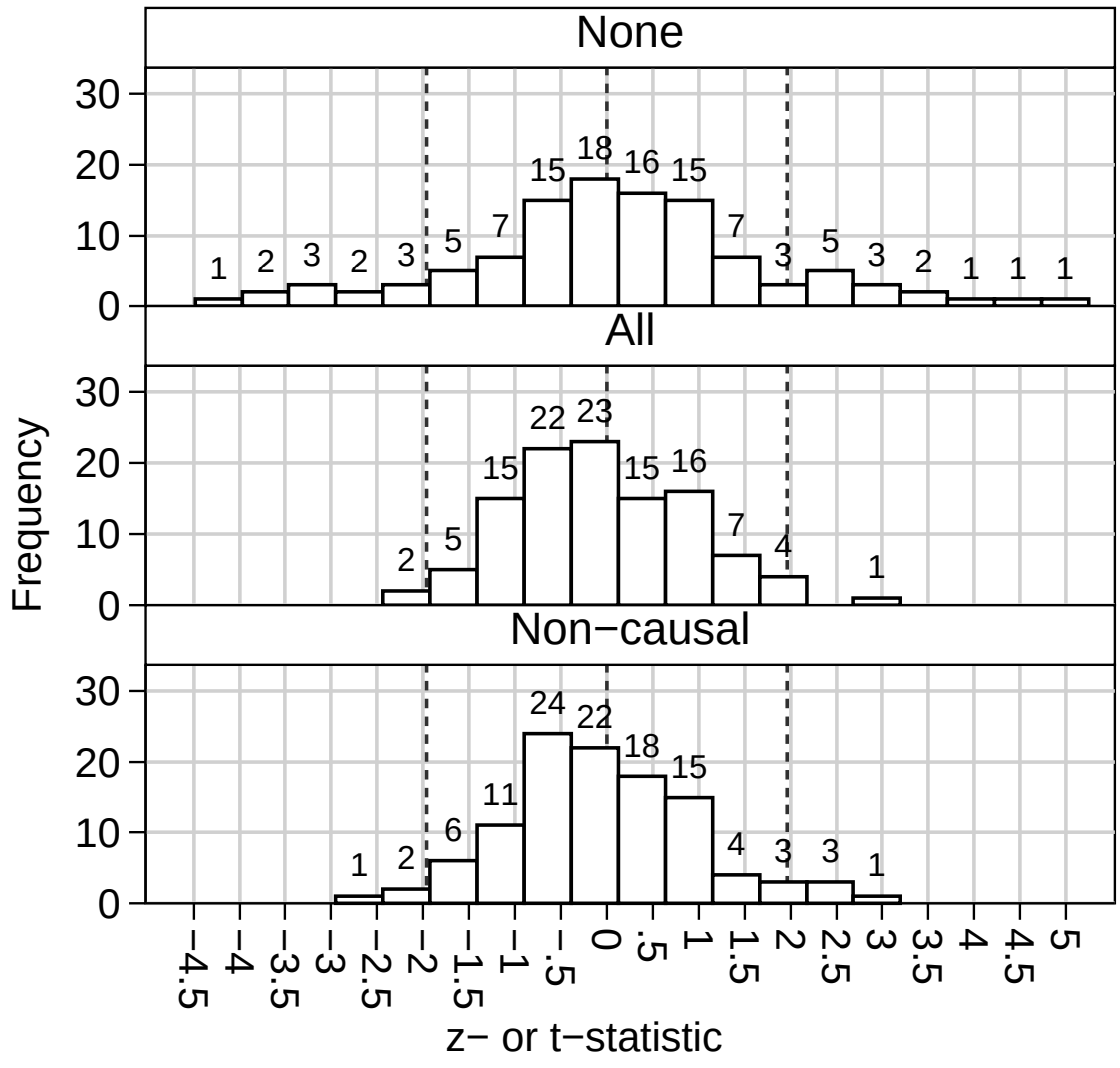
- In the Avon Longitudinal Study of Parents and Children (ALSPAC), mothers of 12008 children completed a food frequency questionnaire (FFQ) on their diet during pregnancy.
- 5 dietary pattern scores were derived from the FFQ data, using principal component analysis (Northstone *et al.*, 2008)[4].
- Associations were measured (using linear or logistic regression) with 22 child disease outcomes, implying $22 \times 5 = 110$ associations.
- Each of these associations was measured with adjustment for 3 nested confounder sets, described as "None" (unadjusted), "All" (adjusted for the full list of confounders), and "Non-causal" (adjusted for a restricted subset of confounders).
- The confidence intervals, *t*– or *z*–statistics, and *P*–values were saved in `parmest` resultssets, plotted, and entered into multiple–test procedures.

Example from the ALSPAC cohort: 110 FFQ diet–disease associations

- In the Avon Longitudinal Study of Parents and Children (ALSPAC), mothers of 12008 children completed a food frequency questionnaire (FFQ) on their diet during pregnancy.
- 5 dietary pattern scores were derived from the FFQ data, using principal component analysis (Northstone *et al.*, 2008)[4].
- Associations were measured (using linear or logistic regression) with 22 child disease outcomes, implying $22 \times 5 = 110$ associations.
- Each of these associations was measured with adjustment for 3 nested confounder sets, described as "None" (unadjusted), "All" (adjusted for the full list of confounders), and "Non-causal" (adjusted for a restricted subset of confounders).
- The confidence intervals, *t*– or *z*–statistics, and *P*–values were saved in `parmest` resultssets, plotted, and entered into multiple–test procedures.

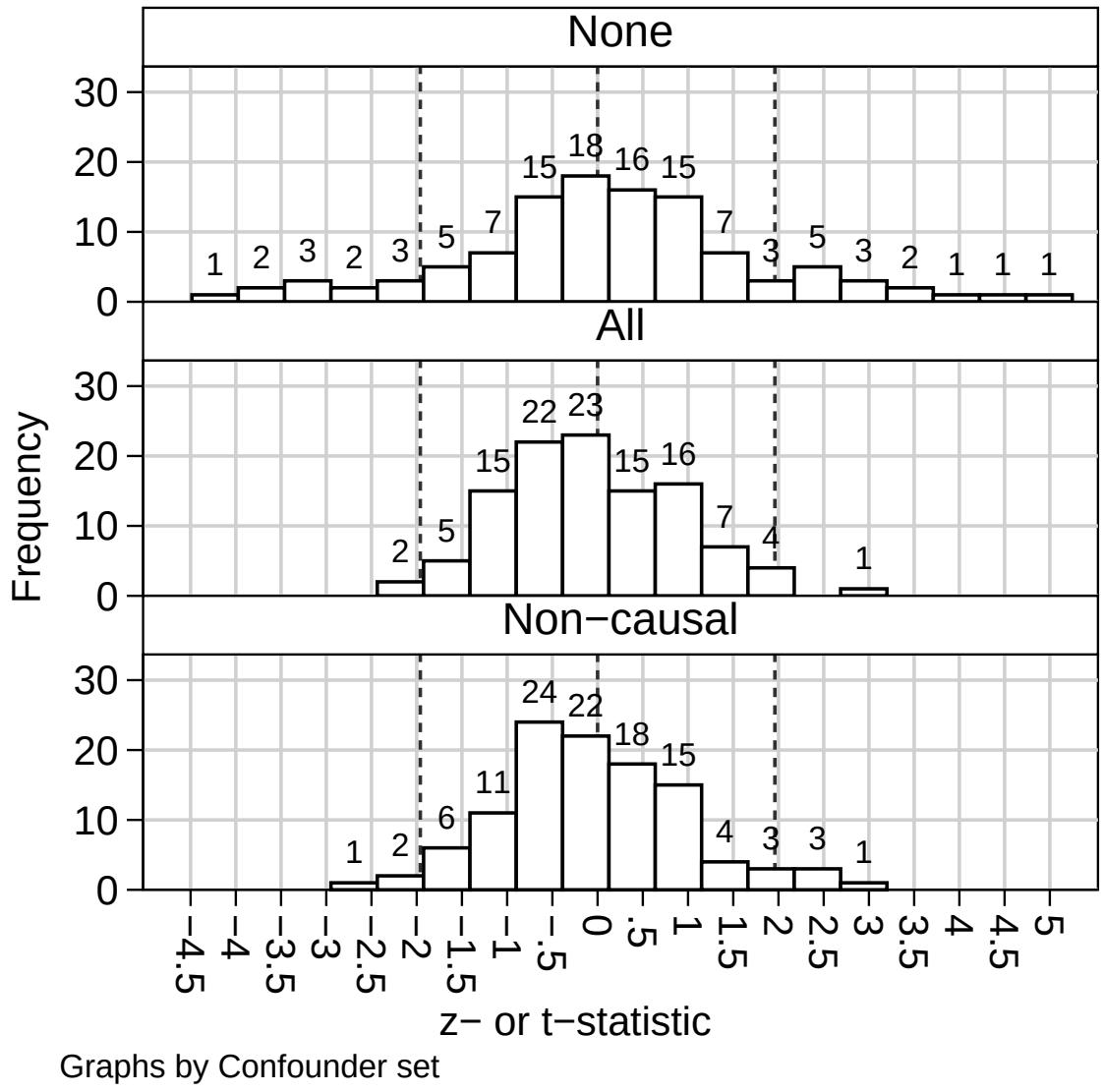
Histograms of test statistics for 110 diet–disease associations

- The unadjusted test statistics (top panel) have more extreme values than would be expected by chance in 110 associations.
- The 2 sets of confounder–adjusted test statistics, by contrast, look "standard Normal".
- Therefore, the unadjusted associations are not due to chance, but *may* be due to confounding.



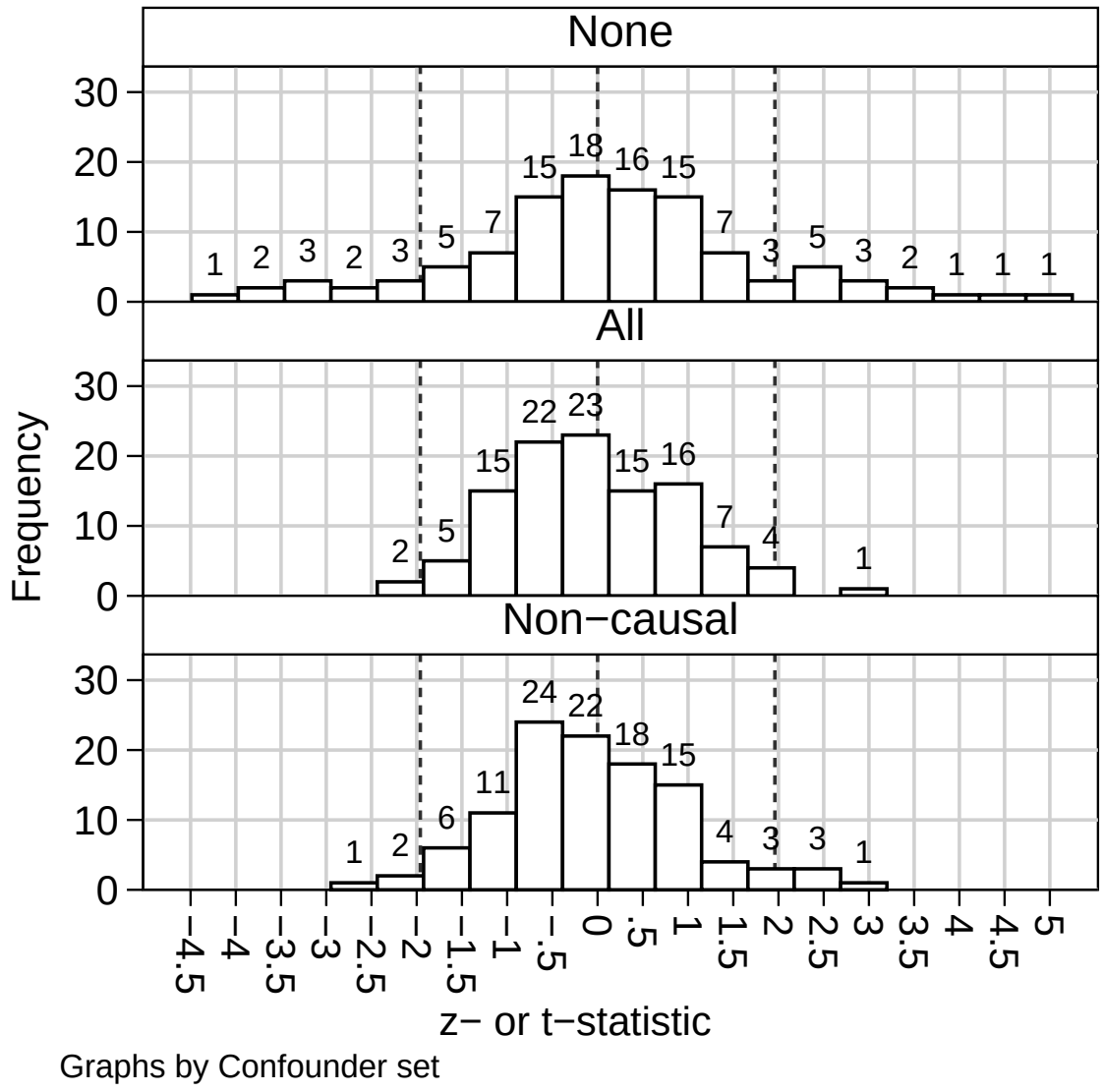
Histograms of test statistics for 110 diet–disease associations

- The unadjusted test statistics (top panel) have more extreme values than would be expected by chance in 110 associations.
- The 2 sets of confounder–adjusted test statistics, by contrast, look "standard Normal".
- Therefore, the unadjusted associations are not due to chance, but *may* be due to confounding.



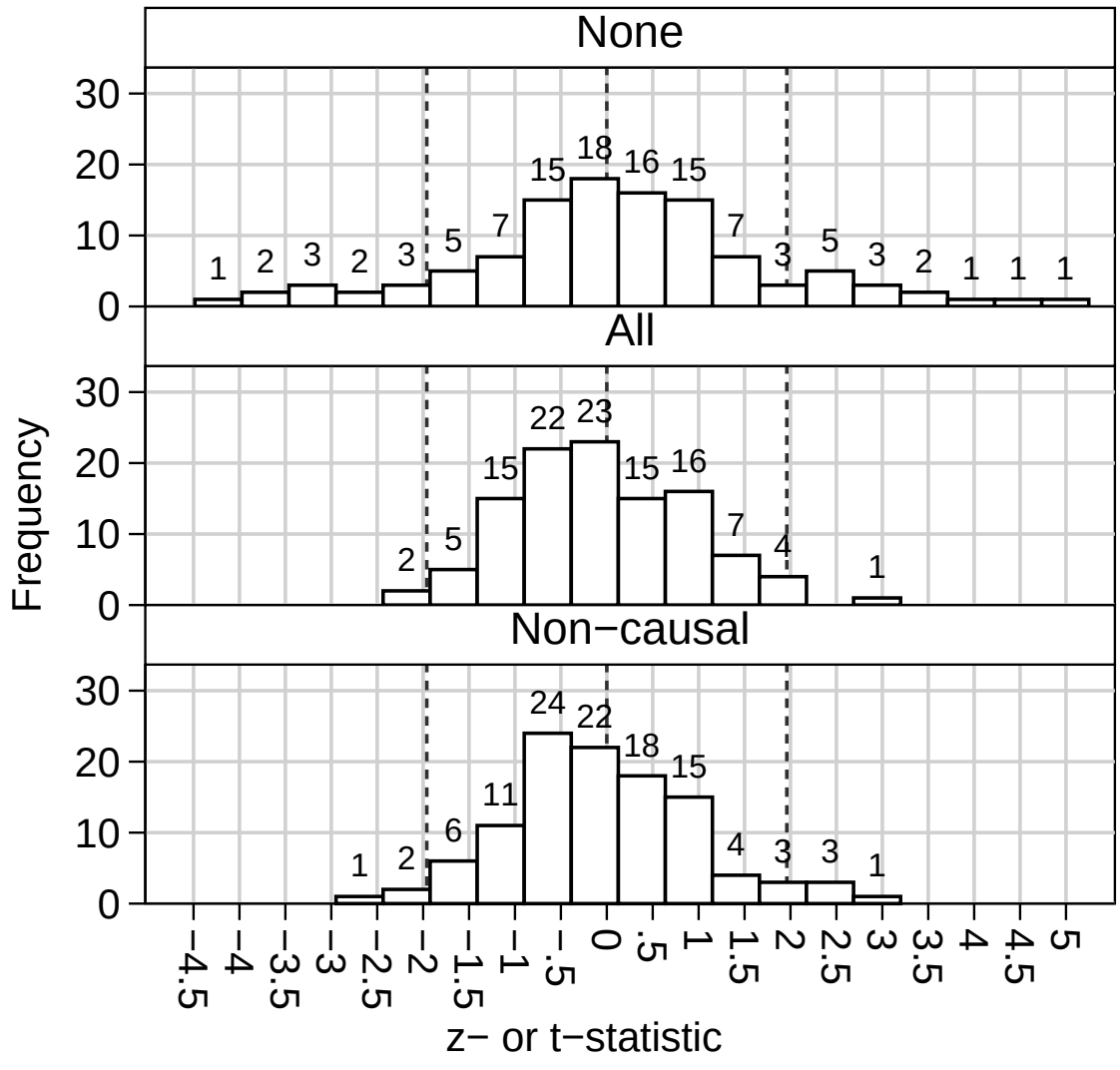
Histograms of test statistics for 110 diet–disease associations

- The unadjusted test statistics (top panel) have more extreme values than would be expected by chance in 110 associations.
- The 2 sets of confounder–adjusted test statistics, by contrast, look "standard Normal".
- Therefore, the unadjusted associations are not due to chance, but *may* be due to confounding.



Histograms of test statistics for 110 diet–disease associations

- The unadjusted test statistics (top panel) have more extreme values than would be expected by chance in 110 associations.
- The 2 sets of confounder–adjusted test statistics, by contrast, look "standard Normal".
- Therefore, the unadjusted associations are not due to chance, but *may* be due to confounding.



The four component modules of `parmest`

The `parmest` package contains 4 component ado–files:

- `parmest` inputs the current set of estimation results, and outputs a resultsset with 1 observation per estimated parameter.
- `parmby` executes an estimation command once for each of a set of by–groups, and outputs a resultsset with 1 observation per parameter per by–group.
- `metaparm` inputs an existing resultsset containing 1 observation for each of a set of uncorrelated parameters, and outputs a new resultsset, containing results for linear combinations of these parameters. Applications include meta–analyses, and also confidence intervals for differences (or ratios) between parameters from independent sets of estimation results.
- `parmcip` does the low–level work. It inputs an existing resultsset containing estimates, standard errors and (optionally) degrees of freedom, and adds new variables, containing t– or z–statistics, P–values, and confidence limits.

The four component modules of `parmest`

The `parmest` package contains 4 component ado–files:

- `parmest` inputs the current set of estimation results, and outputs a resultsset with 1 observation per estimated parameter.
- `parmby` executes an estimation command once for each of a set of by–groups, and outputs a resultsset with 1 observation per parameter per by–group.
- `metaparm` inputs an existing resultsset containing 1 observation for each of a set of uncorrelated parameters, and outputs a new resultsset, containing results for linear combinations of these parameters. Applications include meta–analyses, and also confidence intervals for differences (or ratios) between parameters from independent sets of estimation results.
- `parmcip` does the low–level work. It inputs an existing resultsset containing estimates, standard errors and (optionally) degrees of freedom, and adds new variables, containing t– or z–statistics, P–values, and confidence limits.

The four component modules of `parmest`

The `parmest` package contains 4 component ado–files:

- `parmest` inputs the current set of estimation results, and outputs a resultsset with 1 observation per estimated parameter.
- `parmby` executes an estimation command once for each of a set of by–groups, and outputs a resultsset with 1 observation per parameter per by–group.
- `metaparm` inputs an existing resultsset containing 1 observation for each of a set of uncorrelated parameters, and outputs a new resultsset, containing results for linear combinations of these parameters. Applications include meta–analyses, and also confidence intervals for differences (or ratios) between parameters from independent sets of estimation results.
- `parmcip` does the low–level work. It inputs an existing resultsset containing estimates, standard errors and (optionally) degrees of freedom, and adds new variables, containing t– or z–statistics, P–values, and confidence limits.

The four component modules of `parmest`

The `parmest` package contains 4 component ado–files:

- `parmest` inputs the current set of estimation results, and outputs a resultsset with 1 observation per estimated parameter.
- `parmby` executes an estimation command once for each of a set of by–groups, and outputs a resultsset with 1 observation per parameter per by–group.
- `metaparm` inputs an existing resultsset containing 1 observation for each of a set of uncorrelated parameters, and outputs a new resultsset, containing results for linear combinations of these parameters. Applications include meta–analyses, and also confidence intervals for differences (or ratios) between parameters from independent sets of estimation results.
- `parmcip` does the low–level work. It inputs an existing resultsset containing estimates, standard errors and (optionally) degrees of freedom, and adds new variables, containing t– or z–statistics, P–values, and confidence limits.

The four component modules of `parmest`

The `parmest` package contains 4 component ado–files:

- `parmest` inputs the current set of estimation results, and outputs a resultsset with 1 observation per estimated parameter.
- `parmby` executes an estimation command once for each of a set of by–groups, and outputs a resultsset with 1 observation per parameter per by–group.
- `metaparm` inputs an existing resultsset containing 1 observation for each of a set of uncorrelated parameters, and outputs a new resultsset, containing results for linear combinations of these parameters. Applications include meta–analyses, and also confidence intervals for differences (or ratios) between parameters from independent sets of estimation results.
- `parmcip` does the low–level work. It inputs an existing resultsset containing estimates, standard errors and (optionally) degrees of freedom, and adds new variables, containing t– or z–statistics, P–values, and confidence limits.

`metaparm` example: Unequal–variance *t*–tests in the `auto` data

- `metaparm` inputs an existing resultsset, with 1 observation for each of a set of uncorrelated parameter estimates.
- It outputs a new resultsset, with 1 observation for each by–group in the input resultsset, and data on a linear combination of the parameters in the by–group, with weightings defined by a Stata `aweight` or `iweight` expression.
- It can be used to estimate weighted arithmetic or geometric mean parameters for a meta–analysis, or differences (or ratios) between parameter pairs, or differences between differences (or ratios between ratios), sometimes called “interactions”.
- We will demonstrate `metaparm` by calculating confidence intervals for mean differences between weights in car groups, using unequal–variance *t*–tests.

`metaparm` example: Unequal–variance *t*–tests in the `auto` data

- `metaparm` inputs an existing resultsset, with 1 observation for each of a set of uncorrelated parameter estimates.
- It outputs a new resultsset, with 1 observation for each by–group in the input resultsset, and data on a linear combination of the parameters in the by–group, with weightings defined by a Stata `aweight` or `iweight` expression.
- It can be used to estimate weighted arithmetic or geometric mean parameters for a meta–analysis, or differences (or ratios) between parameter pairs, or differences between differences (or ratios between ratios), sometimes called “interactions”.
- We will demonstrate `metaparm` by calculating confidence intervals for mean differences between weights in car groups, using unequal–variance *t*–tests.

`metaparm` example: Unequal–variance *t*–tests in the `auto` data

- `metaparm` inputs an existing resultsset, with 1 observation for each of a set of uncorrelated parameter estimates.
- It outputs a new resultsset, with 1 observation for each by–group in the input resultsset, and data on a linear combination of the parameters in the by–group, with weightings defined by a Stata `aweight` or `iweight` expression.
- It can be used to estimate weighted arithmetic or geometric mean parameters for a meta–analysis, or differences (or ratios) between parameter pairs, or differences between differences (or ratios between ratios), sometimes called "interactions".
- We will demonstrate `metaparm` by calculating confidence intervals for mean differences between weights in car groups, using unequal–variance *t*–tests.

`metaparm` example: Unequal–variance *t*–tests in the `auto` data

- `metaparm` inputs an existing resultsset, with 1 observation for each of a set of uncorrelated parameter estimates.
- It outputs a new resultsset, with 1 observation for each by–group in the input resultsset, and data on a linear combination of the parameters in the by–group, with weightings defined by a Stata `aweight` or `iweight` expression.
- It can be used to estimate weighted arithmetic or geometric mean parameters for a meta–analysis, or differences (or ratios) between parameter pairs, or differences between differences (or ratios between ratios), sometimes called "interactions".
- We will demonstrate `metaparm` by calculating confidence intervals for mean differences between weights in car groups, using unequal–variance *t*–tests.

`metaparm` example: Unequal–variance *t*–tests in the `auto` data

- `metaparm` inputs an existing resultsset, with 1 observation for each of a set of uncorrelated parameter estimates.
- It outputs a new resultsset, with 1 observation for each by–group in the input resultsset, and data on a linear combination of the parameters in the by–group, with weightings defined by a Stata `aweight` or `iweight` expression.
- It can be used to estimate weighted arithmetic or geometric mean parameters for a meta–analysis, or differences (or ratios) between parameter pairs, or differences between differences (or ratios between ratios), sometimes called "interactions".
- We will demonstrate `metaparm` by calculating confidence intervals for mean differences between weights in car groups, using unequal–variance *t*–tests.

Creating a resultsset with 1 observation per car group

In the `auto` data, we define a new variable `odd`, indicating odd– or even–numbered cars. We then use `parmby` to create a resultsset with 1 observation for each of 4 car groups, defined by `odd` and `foreign`, and data on mean weights in these car groups.

```
. sysuse auto, clear;
(1978 Automobile Data)

. gene byte odd=mod(_n,2);

. lab def odd 0 "Even" 1 "Odd";

. lab val odd odd;

. lab var odd "Odd or even numbered car";

. parmby "mean weight", by(odd foreign) norestore;
```

`parmby` produces a lot of output (not shown), and also a resultsset in the memory.

Creating a `metaparm` resultsset with data on mean differences

We list the new resultsset, with data on 4 car group mean weights. Then we replace it with a new `metaparm` resultsset, with 1 observation per value of `odd`, and data on mean weight differences between non–US and US cars with that value of `odd`. Note the non–integer degrees of freedom for unequal–variance *t*–tests (Satterthwaite, 1946)[5], which `metaparm` uses as the default.

```
. list odd foreign estimate stderr dof min95 max95, clean noobs;

     odd    foreign    estimate      stderr   dof       min95       max95
    Even   Domestic   3200.7692    131.7234    25   2929.4798   3472.0586
    Even    Foreign   2321.8182   154.14762    10   1978.3559   2665.2805
     Odd   Domestic   3433.4615   139.65886    25   3145.8287   3721.0943
     Odd    Foreign        2310   109.66063    10   2065.6609   2554.3391

. metaparm [iwei=(foreign==1)-(foreign==0)], by(odd) norestore;

. list odd estimate stderr dof min95 max95 p, clean noobs;

     odd     estimate      stderr         dof        min95        max95           p
    Even   -878.95105   202.76228   24.673902   -1296.8278    -461.0743   .00021402
     Odd   -1123.4615   177.56704    33.49731   -1484.5207   -762.40233   3.466e-07
```

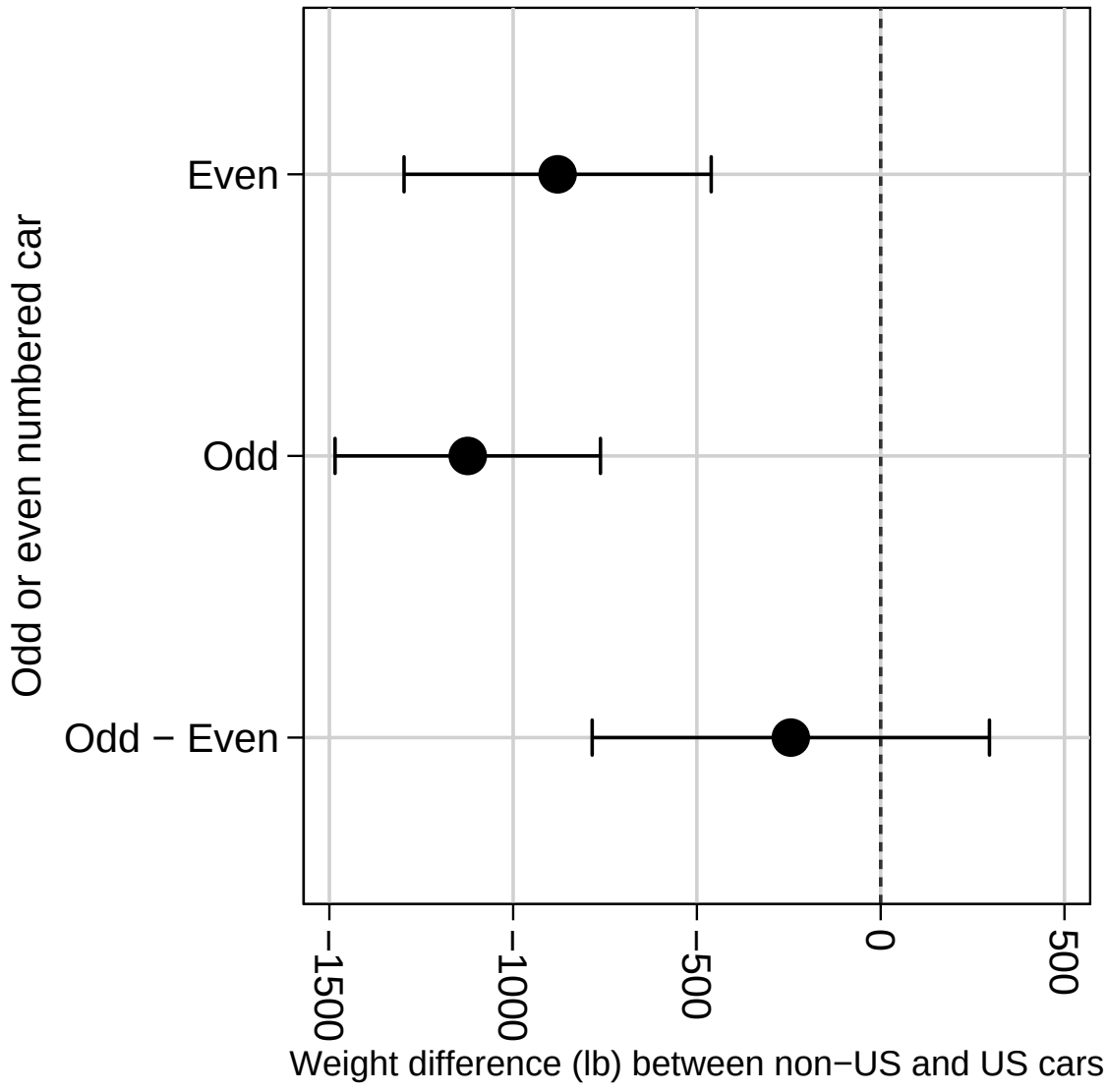
The odd–even difference between foreign–US differences

Finally, we use `metaparm` again to create a new resultsset, containing the odd–even difference between the foreign–US differences, and stored in a temporary file. This resultsset is `append`ed to the original resultsset, and given a third value of `odd`, and the combined resultsset of mean weight differences is listed.

```
. tempfile tf1;

. metaparm [iwei=(odd==1)-(odd==0)], saving(`"`tf1'"', replace);
(note: file C:\DOCUME~1\rnewson\LOCALS~1\Temp\ST_0700000e.tmp not found)
file C:\DOCUME~1\rnewson\LOCALS~1\Temp\ST_0700000e.tmp saved

. append using `"`tf1'"';

. replace odd=2 if missing(odd);
(1 real change made)

. lab def odd 2 "Odd - Even", modify;

. list odd estimate stderr dof min95 max95 p, clean noobs;

          odd     estimate      stderr         dof        min95        max95            p
         Even   -878.95105   202.76228   24.673902   -1296.8278    -461.0743   .00021402
          Odd   -1123.4615   177.56704    33.49731   -1484.5207   -762.40233   3.466e-07
    Odd - Even   -244.51049    269.5229   53.746839   -784.92967    295.90869   .36835197
```

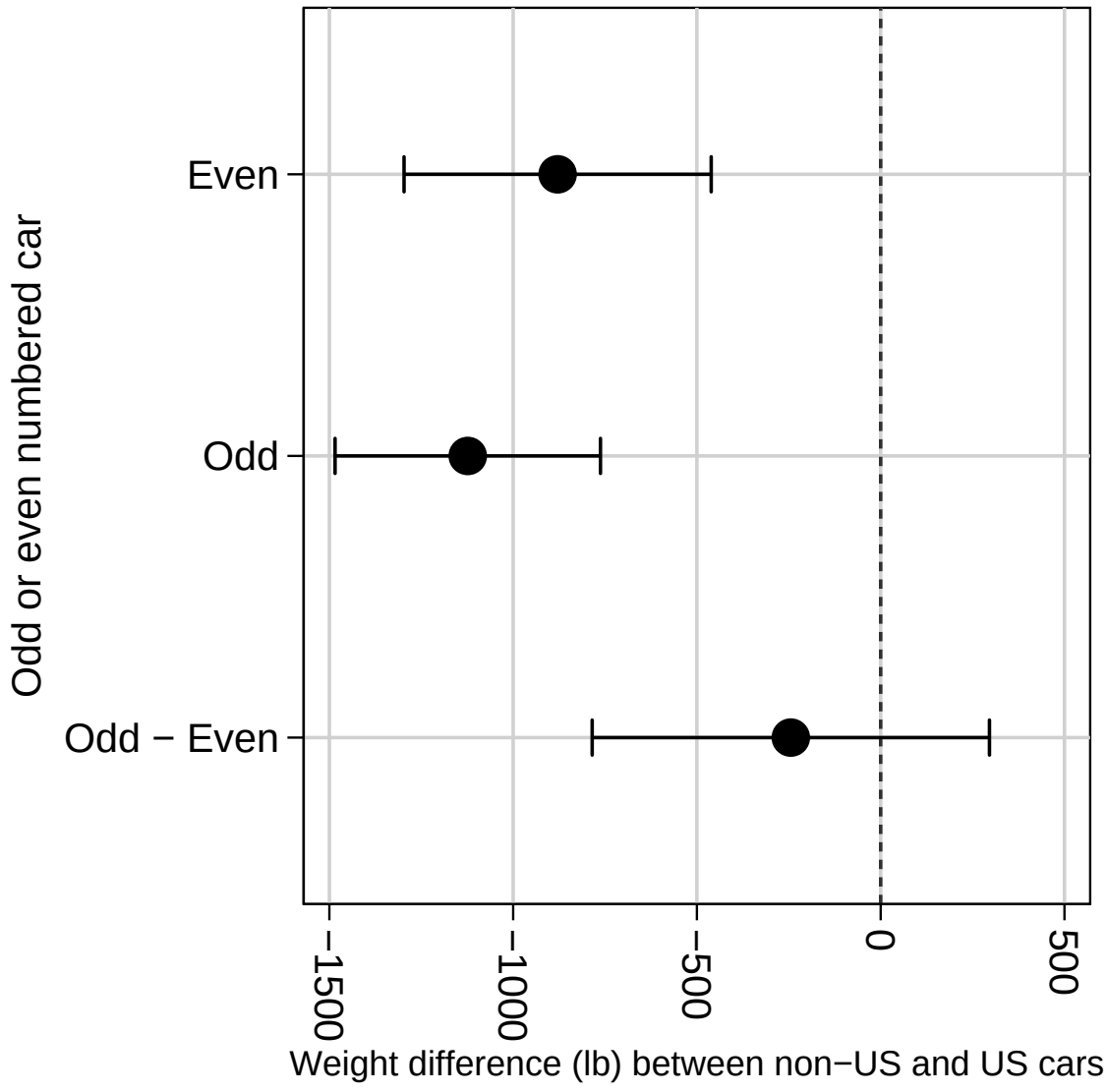
Satterthwaite confidence interval plot of foreign–US weight differences

- This plot was produced using the SSC package `eclplot`.
- Unsurprisingly, non–US cars are lighter (on average) than US cars, whether they are odd– or even–numbered.
- *However*, the population odd–even difference between foreign–US differences may be zero.



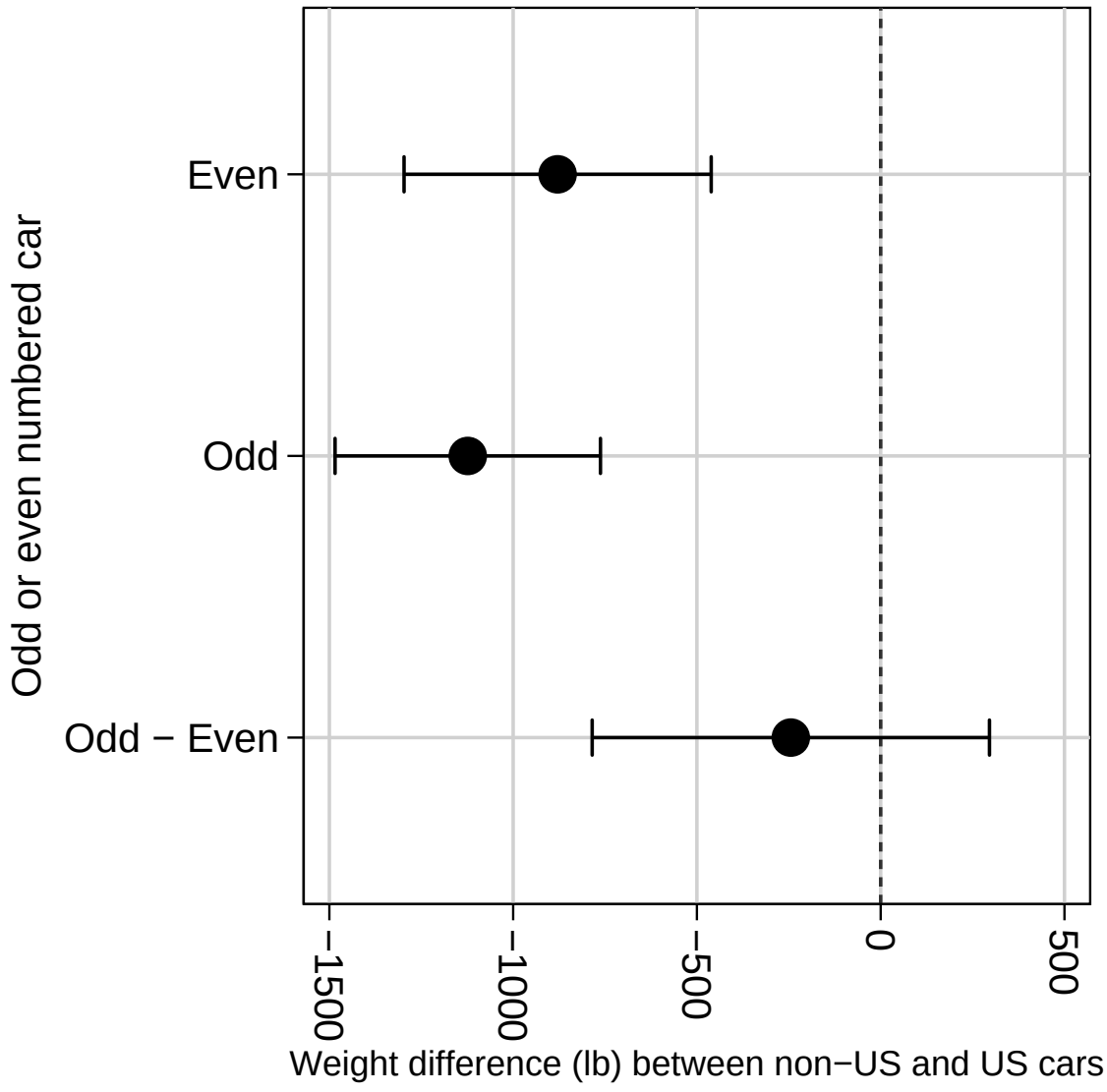
Satterthwaite confidence interval plot of foreign–US weight differences

- This plot was produced using the SSC package `eclplot`.
- Unsurprisingly, non–US cars are lighter (on average) than US cars, whether they are odd– or even–numbered.
- *However*, the population odd–even difference between foreign–US differences may be zero.



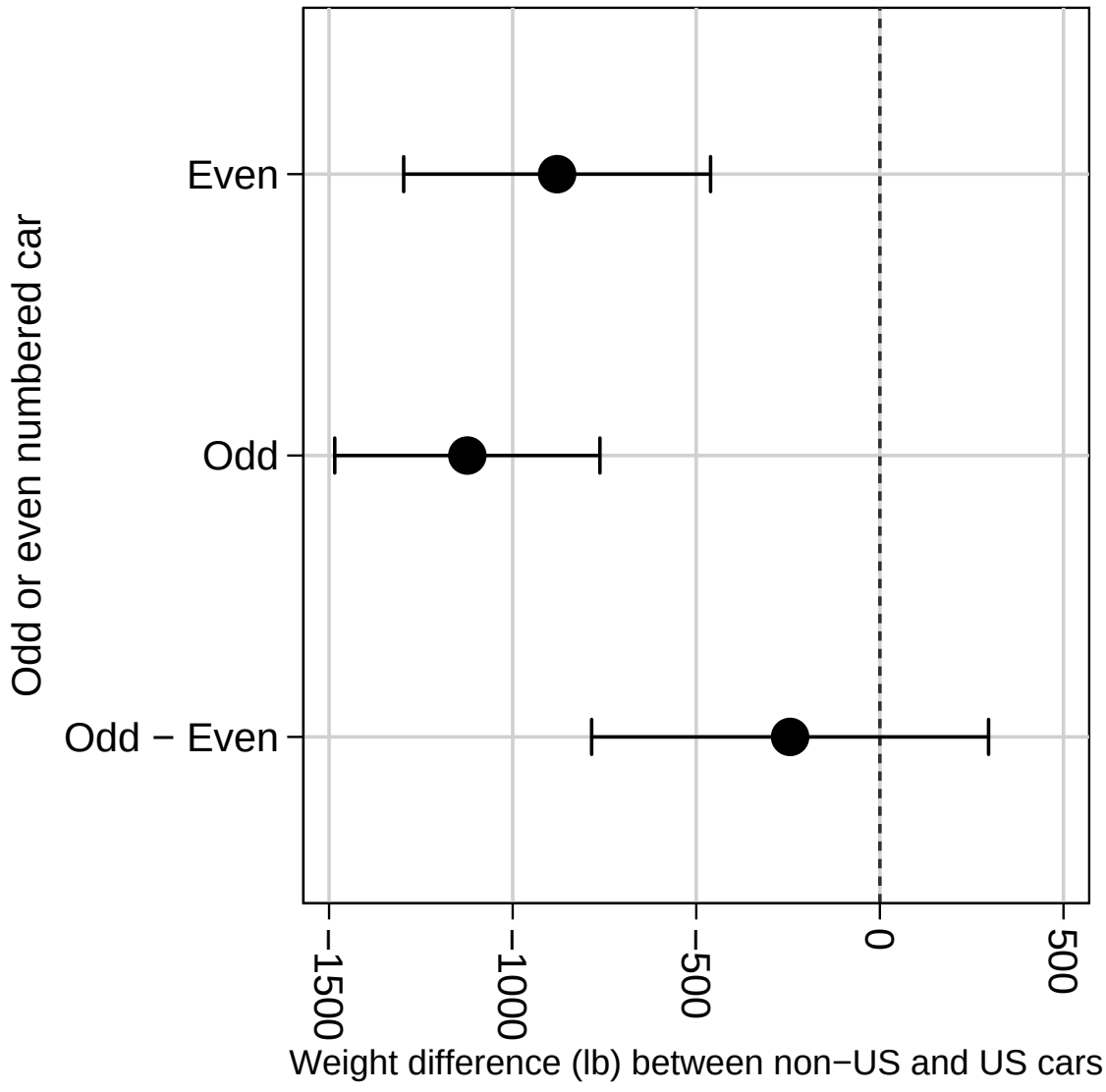
Satterthwaite confidence interval plot of foreign–US weight differences

- This plot was produced using the SSC package `eclplot`.
- Unsurprisingly, non–US cars are lighter (on average) than US cars, whether they are odd– or even–numbered.
- *However*, the population odd–even difference between foreign–US differences may be zero.



Satterthwaite confidence interval plot of foreign–US weight differences

- This plot was produced using the SSC package `eclplot`.
- Unsurprisingly, non–US cars are lighter (on average) than US cars, whether they are odd– or even–numbered.
- *However*, the population odd–even difference between foreign–US differences may be zero.



`parmcip`: the low–level utility behind the `parmest` package

- The `parmcip` module inputs existing variables containing estimates, standard errors, and (optionally) degrees of freedom.
- It produces new variables in the same dataset, containing confidence intervals, t– or z–test statistics, and P–values.
- The `parmest`, `parmby` and `metaparm` modules all call `parmcip` to calculate their confidence intervals, test statistics and P–values.
- *However*, `parmcip` can also be used alone, as we will show.

`parmcip`: the low–level utility behind the `parmest` package

- The `parmcip` module inputs existing variables containing estimates, standard errors, and (optionally) degrees of freedom.
- It produces new variables in the same dataset, containing confidence intervals, t– or z–test statistics, and P–values.
- The `parmest`, `parmby` and `metaparm` modules all call `parmcip` to calculate their confidence intervals, test statistics and P–values.
- *However*, `parmcip` can also be used alone, as we will show.

`parmcip`: the low–level utility behind the `parmest` package

- The `parmcip` module inputs existing variables containing estimates, standard errors, and (optionally) degrees of freedom.
- It produces new variables in the same dataset, containing confidence intervals, *t*– or *z*–test statistics, and *P*–values.
- The `parmest`, `parmby` and `metaparm` modules all call `parmcip` to calculate their confidence intervals, test statistics and *P*–values.
- *However*, `parmcip` can also be used alone, as we will show.

`parmcip`: the low–level utility behind the `parmest` package

- The `parmcip` module inputs existing variables containing estimates, standard errors, and (optionally) degrees of freedom.
- It produces new variables in the same dataset, containing confidence intervals, t– or z–test statistics, and P–values.
- The `parmest`, `parmby` and `metaparm` modules all call `parmcip` to calculate their confidence intervals, test statistics and P–values.
- *However*, `parmcip` can also be used alone, as we will show.

`parmcip`: the low–level utility behind the `parmest` package

- The `parmcip` module inputs existing variables containing estimates, standard errors, and (optionally) degrees of freedom.
- It produces new variables in the same dataset, containing confidence intervals, t– or z–test statistics, and P–values.
- The `parmest`, `parmby` and `metaparm` modules all call `parmcip` to calculate their confidence intervals, test statistics and P–values.
- *However*, `parmcip` can also be used alone, as we will show.

`parmcip` example: Hardy–Weinberg equilibrium

- A typical 2–allele polymorphism has a commoner allele A, a rarer allele a, and possible genotypes AA, Aa and aa, with population prevalences P_{AA}, P_{Aa} and P_{aa}, respectively.
- An important parameter is the **geometric mean (GM) homozygote/heterozygote ratio** (Lindley, 1988)[1], defined as

$$H = \sqrt{P_{AA}P_{aa}}/P_{Aa}$$

  and is equal to 0.5 if the maternal and paternal alleles of a population member are statistically independent, and greater than (less than) 0.5 if the maternal and paternal alleles are positively (negatively) correlated.
- If $H = 0.5$ (as with random mating), then the 2–allele polymorphism is said to be in **Hardy–Weinberg equilibrium**.

`parmcip` example: Hardy–Weinberg equilibrium

- A typical 2–allele polymorphism has a commoner allele A, a rarer allele a, and possible genotypes AA, Aa and aa, with population prevalences P_{AA}, P_{Aa} and P_{aa}, respectively.
- An important parameter is the **geometric mean (GM) homozygote/heterozygote ratio** (Lindley, 1988)[1], defined as

$$H = \sqrt{P_{AA}P_{aa}}/P_{Aa}$$

  and is equal to 0.5 if the maternal and paternal alleles of a population member are statistically independent, and greater than (less than) 0.5 if the maternal and paternal alleles are positively (negatively) correlated.
- If $H = 0.5$ (as with random mating), then the 2–allele polymorphism is said to be in **Hardy–Weinberg equilibrium**.

`parmcip` example: Hardy–Weinberg equilibrium

- A typical 2–allele polymorphism has a commoner allele A, a rarer allele a, and possible genotypes AA, Aa and aa, with population prevalences P_{AA}, P_{Aa} and P_{aa}, respectively.
- An important parameter is the **geometric mean (GM) homozygote/heterozygote ratio** (Lindley, 1988)[1], defined as

$$H = \sqrt{P_{AA}P_{aa}}/P_{Aa}$$

  and is equal to 0.5 if the maternal and paternal alleles of a population member are statistically independent, and greater than (less than) 0.5 if the maternal and paternal alleles are positively (negatively) correlated.
- If $H = 0.5$ (as with random mating), then the 2–allele polymorphism is said to be in **Hardy–Weinberg equilibrium**.

`parmcip` example: Hardy–Weinberg equilibrium

- A typical 2–allele polymorphism has a commoner allele A, a rarer allele a, and possible genotypes AA, Aa and aa, with population prevalences P_{AA}, P_{Aa} and P_{aa}, respectively.
- An important parameter is the **geometric mean (GM) homozygote/heterozygote ratio** (Lindley, 1988)[1], defined as

$$H = \sqrt{P_{AA}P_{aa}}/P_{Aa}$$

  and is equal to 0.5 if the maternal and paternal alleles of a population member are statistically independent, and greater than (less than) 0.5 if the maternal and paternal alleles are positively (negatively) correlated.
- If $H = 0.5$ (as with random mating), then the 2–allele polymorphism is said to be in **Hardy–Weinberg equilibrium**.

`parmcip` example: Hardy–Weinberg equilibrium (continued)

- The sample estimate of the population GM homozygote/heterozygote ratio is the sample GM homozygote/heterozygote ratio, defined as

$$\widehat{H} = \sqrt{N_{AA}N_{aa}}/N_{Aa}$$

  where N_{AA}, N_{Aa} and N_{aa} are the sample frequencies of genotypes AA, Aa and aa.
- The sample standard error of $\log \widehat{H}$ is

$$\widehat{SE}\left(\log \widehat{H}\right) = \sqrt{1/(4N_{AA}) + 1/(4N_{aa}) + 1/N_{Aa}}$$

  and can be used to define confidence intervals for $\log H$, and therefore for H.
- This can be done using `parmcip` in a dataset with one observation for each of a set of polymorphisms, and data on the 3 sample genotype frequencies.

`parmcip` example: Hardy–Weinberg equilibrium (continued)

- The sample estimate of the population GM homozygote/heterozygote ratio is the sample GM homozygote/heterozygote ratio, defined as

$$\widehat{H} = \sqrt{N_{AA}N_{aa}}/N_{Aa}$$

where N_{AA}, N_{Aa} and N_{aa} are the sample frequencies of genotypes AA, Aa and aa.

- The sample standard error of $\log \widehat{H}$ is

$$\widehat{SE}\left(\log \widehat{H}\right) = \sqrt{1/(4N_{AA}) + 1/(4N_{aa}) + 1/N_{Aa}}$$

and can be used to define confidence intervals for $\log H$, and therefore for H.

- This can be done using `parmcip` in a dataset with one observation for each of a set of polymorphisms, and data on the 3 sample genotype frequencies.

`parmcip` example: Hardy–Weinberg equilibrium (continued)

- The sample estimate of the population GM homozygote/heterozygote ratio is the sample GM homozygote/heterozygote ratio, defined as

$$\widehat{H} = \sqrt{N_{AA}N_{aa}}/N_{Aa}$$

  where N_{AA}, N_{Aa} and N_{aa} are the sample frequencies of genotypes AA, Aa and aa.
- The sample standard error of $\log \widehat{H}$ is

$$\widehat{SE}\left(\log \widehat{H}\right) = \sqrt{1/(4N_{AA}) + 1/(4N_{aa}) + 1/N_{Aa}}$$

  and can be used to define confidence intervals for $\log H$, and therefore for H.
- This can be done using `parmcip` in a dataset with one observation for each of a set of polymorphisms, and data on the 3 sample genotype frequencies.

`parmcip` example: Hardy–Weinberg equilibrium (continued)

- The sample estimate of the population GM homozygote/heterozygote ratio is the sample GM homozygote/heterozygote ratio, defined as

$$\widehat{H} = \sqrt{N_{AA}N_{aa}}/N_{Aa}$$

  where N_{AA}, N_{Aa} and N_{aa} are the sample frequencies of genotypes AA, Aa and aa.
- The sample standard error of $\log \widehat{H}$ is

$$\widehat{SE}\left(\log \widehat{H}\right) = \sqrt{1/(4N_{AA}) + 1/(4N_{aa}) + 1/N_{Aa}}$$

  and can be used to define confidence intervals for $\log H$, and therefore for H.
- This can be done using `parmcip` in a dataset with one observation for each of a set of polymorphisms, and data on the 3 sample genotype frequencies.

Hardy–Weinberg equilibrium in the ALSPAC cohort

- In the ALSPAC cohort (14060 children), subsets were genotyped for 18 two-allele polymorphisms.
- We checked these polymorphisms for Hardy–Weinberg equilibrium, using a dataset with 1 observation per polymorphism, and data on frequencies of genotypes with 0, 1 and 2 copies of the rarer allele.
- We calculated the homozygote/heterozygote ratio H, the log of $H/0.5$, and the standard error of this log.
- We then used `parmcip` to calculate confidence limits and P–values for the log of $H/0.5$.
- Finally, we back-transformed the confidence limits, and plotted the confidence intervals for H.

Hardy–Weinberg equilibrium in the ALSPAC cohort

- In the ALSPAC cohort (14060 children), subsets were genotyped for 18 two-allele polymorphisms.
- We checked these polymorphisms for Hardy–Weinberg equilibrium, using a dataset with 1 observation per polymorphism, and data on frequencies of genotypes with 0, 1 and 2 copies of the rarer allele.
- We calculated the homozygote/heterozygote ratio H, the log of $H/0.5$, and the standard error of this log.
- We then used `parmcip` to calculate confidence limits and P–values for the log of $H/0.5$.
- Finally, we back-transformed the confidence limits, and plotted the confidence intervals for H.

Hardy–Weinberg equilibrium in the ALSPAC cohort

- In the ALSPAC cohort (14060 children), subsets were genotyped for 18 two-allele polymorphisms.
- We checked these polymorphisms for Hardy–Weinberg equilibrium, using a dataset with 1 observation per polymorphism, and data on frequencies of genotypes with 0, 1 and 2 copies of the rarer allele.
- We calculated the homozygote/heterozygote ratio H, the log of $H/0.5$, and the standard error of this log.
- We then used `parmcip` to calculate confidence limits and P–values for the log of $H/0.5$.
- Finally, we back-transformed the confidence limits, and plotted the confidence intervals for H.

Hardy–Weinberg equilibrium in the ALSPAC cohort

- In the ALSPAC cohort (14060 children), subsets were genotyped for 18 two-allele polymorphisms.
- We checked these polymorphisms for Hardy–Weinberg equilibrium, using a dataset with 1 observation per polymorphism, and data on frequencies of genotypes with 0, 1 and 2 copies of the rarer allele.
- We calculated the homozygote/heterozygote ratio H, the log of $H/0.5$, and the standard error of this log.
- We then used `parmcip` to calculate confidence limits and P–values for the log of $H/0.5$.
- Finally, we back-transformed the confidence limits, and plotted the confidence intervals for H.

Hardy–Weinberg equilibrium in the ALSPAC cohort

- In the ALSPAC cohort (14060 children), subsets were genotyped for 18 two-allele polymorphisms.
- We checked these polymorphisms for Hardy–Weinberg equilibrium, using a dataset with 1 observation per polymorphism, and data on frequencies of genotypes with 0, 1 and 2 copies of the rarer allele.
- We calculated the homozygote/heterozygote ratio H, the log of $H/0.5$, and the standard error of this log.
- We then used `parmcip` to calculate confidence limits and P–values for the log of $H/0.5$.
- Finally, we back-transformed the confidence limits, and plotted the confidence intervals for H.

Hardy–Weinberg equilibrium in the ALSPAC cohort

- In the ALSPAC cohort (14060 children), subsets were genotyped for 18 two-allele polymorphisms.
- We checked these polymorphisms for Hardy–Weinberg equilibrium, using a dataset with 1 observation per polymorphism, and data on frequencies of genotypes with 0, 1 and 2 copies of the rarer allele.
- We calculated the homozygote/heterozygote ratio H, the log of $H/0.5$, and the standard error of this log.
- We then used `parmcip` to calculate confidence limits and P–values for the log of $H/0.5$.
- Finally, we back-transformed the confidence limits, and plotted the confidence intervals for H.

A dataset with 1 observation per polymorphism

Our input dataset had 1 observation per polymorphism, and data on total numbers of subjects genotyped, and on sample frequencies of subjects with 0, 1 and 2 copies of the rarer allele. The first few lines are listed here.

```
. list poly N _freq*, clean noobs;
```

poly	N	_freq0	_freq1	_freq2
rs10183914	8671	3617	3975	1079
rs1806649	8606	4889	3199	518
rs1962142	8763	7154	1528	81
rs2364723	8725	4047	3760	918
rs6706649	8731	6696	1920	115
rs6726395	8692	2580	4323	1789

(This dataset was produced using the SSC packages `xcontract`, `dsconcat` and `factmerg`, together with `reshape`.)

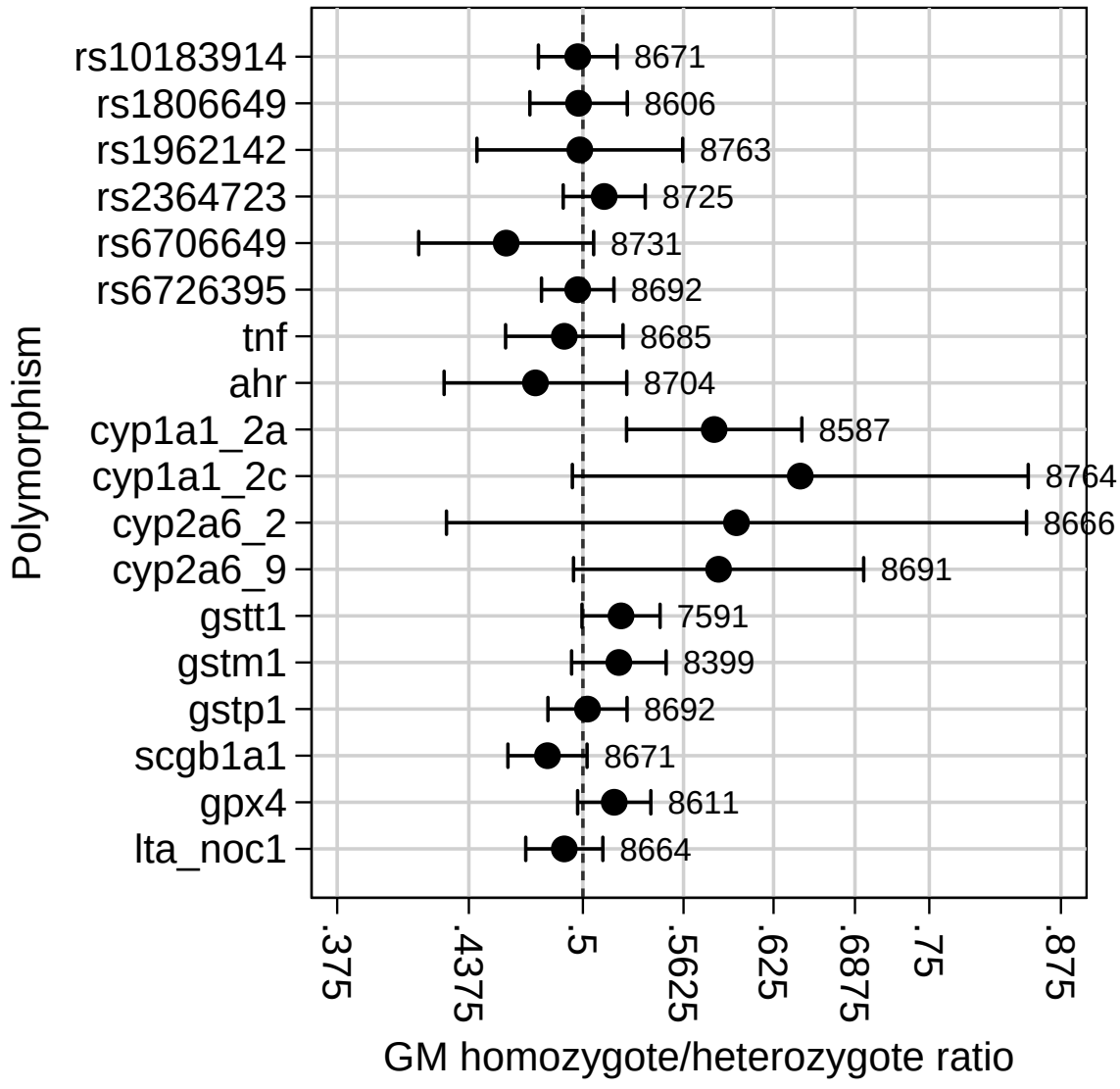
The code to produce confidence intervals for H

This code produced the variables `homhet`, containing H, and `min95` and `max95`, containing its lower and upper 95% confidence limits.

```
. gene double homhet=sqrt(_freq0*_freq2)/_freq1;

. gene double estimate=log(homhet)-log(0.5);

. gene stderr=sqrt( 1/(4*_freq0) + 1/(4*_freq2) + 1/_freq1 );

. lab var homhet "GM homozygote/heterozygote ratio";

. lab var estimate "Log (GM homozygote/heterozygote ratio/0.5)";

. lab var stderr "SE Log (GM homozygote/heterozygote ratio/0.5)";

. parmcip;
Note: variable dof not found, normal distribution assumed

. foreach Y of var min95 max95 {;
  2. replace `Y'=0.5*exp(`Y');
  3. };
(18 real changes made)
(18 real changes made)
```

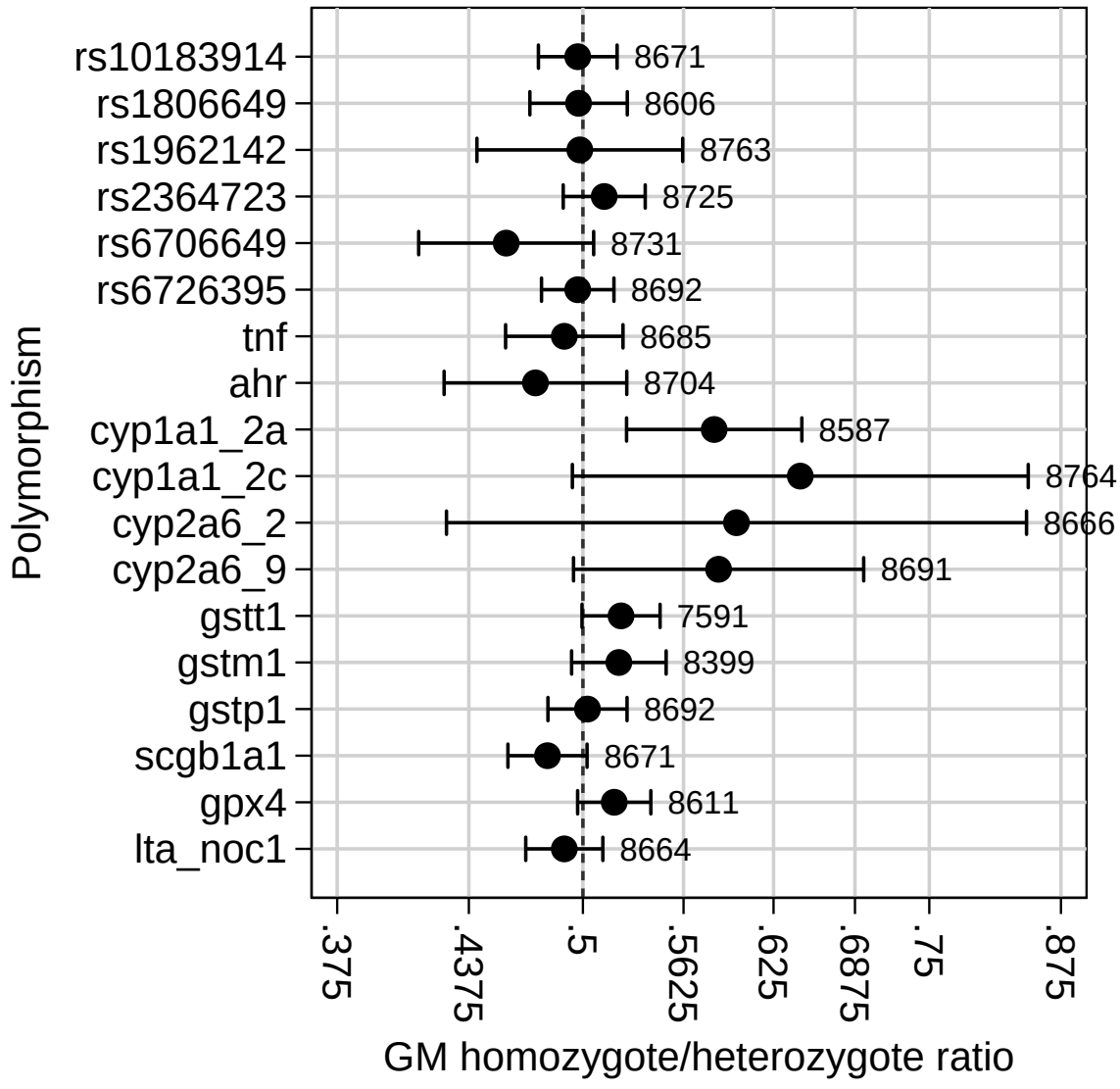
Confidence interval plot of homozygote/heterozygote ratios

- Confidence intervals for H, calculated using `parmcip`, are plotted for the 18 polymorphisms, with their sample numbers.
- Most of these confidence intervals contain the Hardy–Weinberg value of 0.5.
- *However*, there is a hint of *possible* "inbreeding" ($H > 0.5$) for a few polymorphisms.



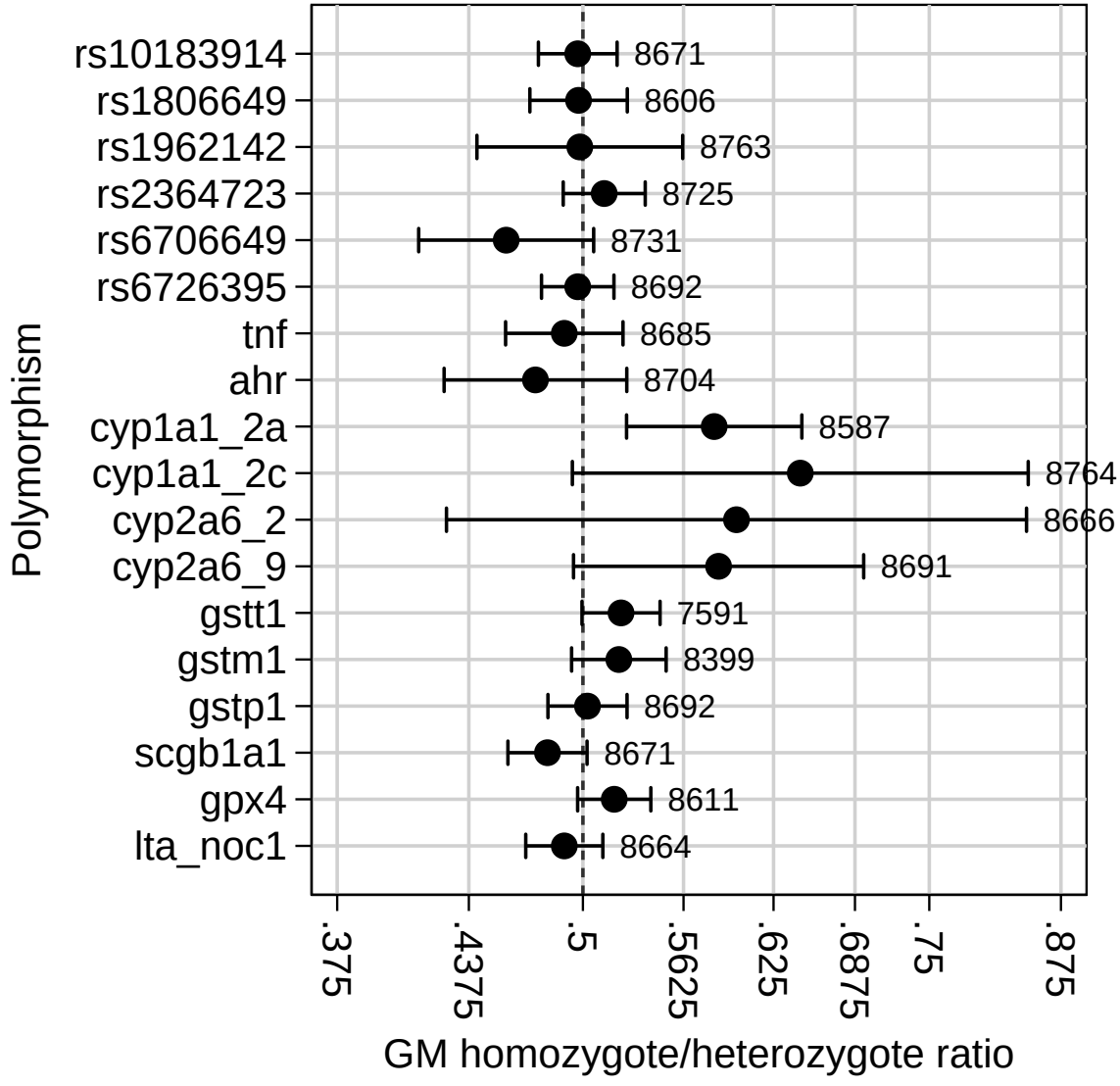
Confidence interval plot of homozygote/heterozygote ratios

- Confidence intervals for H, calculated using `parmcip`, are plotted for the 18 polymorphisms, with their sample numbers.
- Most of these confidence intervals contain the Hardy–Weinberg value of 0.5.
- *However*, there is a hint of *possible* "inbreeding" ($H > 0.5$) for a few polymorphisms.



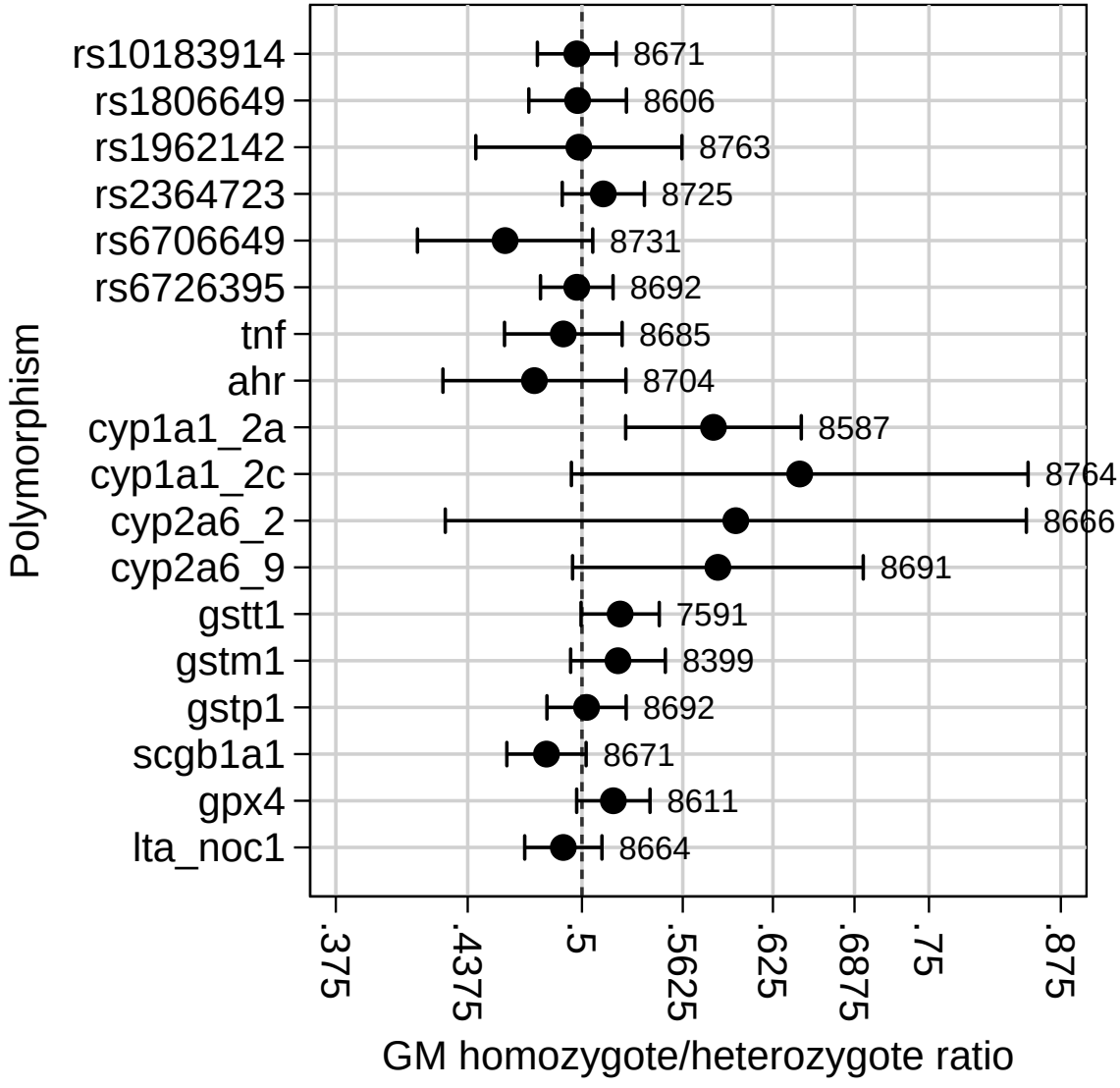
Confidence interval plot of homozygote/heterozygote ratios

- Confidence intervals for H, calculated using `parmcip`, are plotted for the 18 polymorphisms, with their sample numbers.
- Most of these confidence intervals contain the Hardy–Weinberg value of 0.5.
- *However*, there is a hint of *possible* "inbreeding" ($H > 0.5$) for a few polymorphisms.



Confidence interval plot of homozygote/heterozygote ratios

- Confidence intervals for H, calculated using `parmcip`, are plotted for the 18 polymorphisms, with their sample numbers.
- Most of these confidence intervals contain the Hardy–Weinberg value of 0.5.
- *However*, there is a hint of *possible* "inbreeding" ($H > 0.5$) for a few polymorphisms.



References

[1] Lindley D. 1988. Statistical inference concerning Hardy–Weinberg equilibrium. *Bayesian Statistics* **3**: 307–326.

[2] Newson R. and the ALSPAC Study Team. 2003. Multiple–test procedures and smile plots. *The Stata Journal* **3(2)**: 109–132.

[3] Newson R. 2003. Confidence intervals and *P*–values for delivery to the end user. *The Stata Journal* **3(3)**: 245–269.

[4] Northstone K., Emmett P. and Rogers I. 2008. Dietary patterns in pregnancy and associations with socio-demographic and lifestyle factors. *European Journal of Clinical Nutrition* **62**: 471–479.

[5] Satterthwaite F. E. 1946. An approximate distribution of estimates of variance components. *Biometrics Bulletin* **2(6)**: 110–114.

This presentation can be downloaded from the conference website at *http://ideas.repec.org/s/boc/usug08.html*

The `parmest`, `descsave`, `listtex`, `smileplot`, `eclplot`, `xcontract`, `dsconcat` and `factmerg` packages can be downloaded from SSC, using the `ssc` command.