simsum: Analyses of simulation studies including Monte Carlo error

Ian White
MRC Biostatistics Unit
Cambridge
ian.white@mrc-bsu.cam.ac.uk

UK Stata Users' Group, London, 10th September 2009

Introduction

- Simulation studies are often badly reported.
- Monte Carlo (MC) error – the standard deviation of an estimated quantity over repeated simulation studies – is often not reported.
- `simsum` provides routine analysis of simulation studies
- I've worked out appropriate formulae for MC error and implemented them
- `simsum` allows each simulated data set to have been analysed by multiple methods
- This talk describes the scope and syntax of `simsum`, the formulae used, and gives examples.

Statistics computed by `simsum`

Using point estimates:

- Number of non-missing point estimates
- Bias in point estimate
- Empirical standard error
- % gain in precision for one method compared with another *

Using point estimates and standard errors (SEs):

- Number of non-missing SEs
- Root mean squared model-based SE
- Relative % error in SE *
- Coverage of nominal confidence interval
- Power of test

* I've derived (new?) expressions for MC error for these statistics

Example: simulation study comparing different ways to impute missing covariates when fitting a Cox model.

(White & Royston, SiM 2009;28:1982-1998)
Big question: how should we include the outcome in the imputation model?

- 1000 simulated data sets of size 200
- Covariates: standard Normal x and z
- Outcome: Cox model with linear predictor $0.5x + 0.5z$
- 20% of covariate values deleted
- Each simulated data set analysed in three ways:
  - Complete cases (CC)
  - Multiple imputation using `ice`, outcome included as survival time and event indicator (MI_T)
  - As MI_T, but survival time logged (MI_LOGT)
- Focus on coefficient of x

Some data

dataset	method	b	se
1	CC	.7067682	.14651
1	MI_T	.6841882	.1255043
1	MI_LOGT	.7124795	.1410814
2	CC	.3485008	.1599879
2	MI_T	.4060082	.1409831
2	MI_LOGT	.4287003	.1358589
3	CC	.6495075	.1521568
3	MI_T	.5028701	.130078
3	MI_LOGT	.5604051	.1168512

(1000 datasets; true beta = 0.5)

Output from simsum

```
. simsum b, se(se) long(method) id(dataset) true(0.5) mcse format(%7.0g)
Reshaping data to wide format ...
Starting to process results ...
```

Statistic	CC	(MCse)	MI_LOGT	(MCse)	MI_T	(MCse)
Non-missing point estimates	1000	.	1000	.	1000	.
Non-missing standard errors	1000	.	1000	.	1000	.
Bias in point estimate	.01677	.00478	.00092	.00417	-.00119	.00425
Empirical standard error	.15112	.00338	.13201	.00295	.13443	.00301
% gain in precision relative to method CC	.	.	31.046	3.9394	26.368	3.8443
RMS model-based standard error	.1471	.00053	.13494	.0006	.13383	.00059
Relative % error in standard error	-2.6594	2.2055	2.2233	2.3323	-.44122	2.2695
Coverage of nominal 95% confidence interval	94.3	.73315	94.9	.69569	94.3	.73315
Power of 5% level test	94.6	.71473	96.9	.54808	96.3	.59692

Some comments follow...

Output from simsum

```
. simsum b, se(se) long(method) id(dataset) true(0.5) mcse format(%7.0g)
Reshaping data to wide format ...
Starting to process results ...
```

Statistic	CC	(MCse)	MI_LOGT	(MCse)	MI_T	(MCse)
Non-missing point estimates	1000	.	1000	.	1000	.
Non-missing standard errors	1000	.	1000	.	1000	.
Bias in point estimate	.01677	.00478	.00092	.00417	-.00119	.00425
Empirical standard error	.15112	.00338	.13201	.00295	.13443	.00301
% gain in precision relative to method CC	.	.	31.046	3.9394	26.368	3.8443
RMS model-based standard error	.1471	.00053	.13494	.0006	.13383	.00059
Relative % error in standard error	-2.6594	2.2055	2.2233	2.3323	-.44122	2.2695
Coverage of nominal 95% confidence interval	94.3	.73315	94.9	.69569	94.3	.73315
Power of 5% level test	94.6	.71473	96.9	.54808	96.3	.59692

CC has (small-sample) bias away from the null.

Output from simsum

```
. simsum b, se(se) long(method) id(dataset) true(0.5) mcse format(%7.0g)
Reshaping data to wide format ...
Starting to process results ...
```

Statistic	CC	(MCse)	MI_LOGT	(MCse)	MI_T	(MCse)
Non-missing point estimates	1000	.	1000	.	1000	.
Non-missing standard errors	1000	.	1000	.	1000	.
Bias in point estimate	.01677	.00478	.00092	.00417	-.00119	.00425
Empirical standard error	.15112	.00338	.13201	.00295	.13443	.00301
% gain in precision relative to method CC	.	.	31.046	3.9394	26.368	3.8443
RMS model-based standard error	.1471	.00053	.13494	.0006	.13383	.00059
Relative % error in standard error	-2.6594	2.2055	2.2233	2.3323	-.44122	2.2695
Coverage of nominal 95% confidence interval	94.3	.73315	94.9	.69569	94.3	.73315
Power of 5% level test	94.6	.71473	96.9	.54808	96.3	.59692

CC is inefficient compared with MI_LOGT and MI_T.

Output from simsum

```
. simsum b, se(se) long(method) id(dataset) true(0.5) mcse format(%7.0g)
Reshaping data to wide format ...
Starting to process results ...
```

Statistic	CC	(MCse)	MI_LOGT	(MCse)	MI_T	(MCse)
Non-missing point estimates	1000	.	1000	.	1000	.
Non-missing standard errors	1000	.	1000	.	1000	.
Bias in point estimate	.01677	.00478	.00092	.00417	-.00119	.00425
Empirical standard error	.15112	.00338	.13201	.00295	.13443	.00301
% gain in precision relative to method CC	.	.	31.046	3.9394	26.368	3.8443
RMS model-based standard error	.1471	.00053	.13494	.0006	.13383	.00059
Relative % error in standard error	-2.6594	2.2055	2.2233	2.3323	-.44122	2.2695
Coverage of nominal 95% confidence interval	94.3	.73315	94.9	.69569	94.3	.73315
Power of 5% level test	94.6	.71473	96.9	.54808	96.3	.59692

Model-based standard errors are close to the empirical values.

Output from simsum

```
. simsum b, se(se) long(method) id(dataset) true(0.5) mcse format(%7.0g)
Reshaping data to wide format ...
Starting to process results ...
```

Statistic	CC	(MCse)	MI_LOGT	(MCse)	MI_T	(MCse)
Non-missing point estimates	1000	.	1000	.	1000	.
Non-missing standard errors	1000	.	1000	.	1000	.
Bias in point estimate	.01677	.00478	.00092	.00417	-.00119	.00425
Empirical standard error	.15112	.00338	.13201	.00295	.13443	.00301
% gain in precision relative to method CC	.	.	31.046	3.9394	26.368	3.8443
RMS model-based standard error	.1471	.00053	.13494	.0006	.13383	.00059
Relative % error in standard error	-2.6594	2.2055	2.2233	2.3323	-.44122	2.2695
Coverage of nominal 95% confidence interval	94.3	.73315	94.9	.69569	94.3	.73315
Power of 5% level test	94.6	.71473	96.9	.54808	96.3	.59692

Coverage is fine.

Output from simsum

```
. simsum b, se(se) long(method) id(dataset) true(0.5) mcse format(%7.0g)
Reshaping data to wide format ...
Starting to process results ...
```

Statistic	CC	(MCse)	MI_LOGT	(MCse)	MI_T	(MCse)
Non-missing point estimates	1000	.	1000	.	1000	.
Non-missing standard errors	1000	.	1000	.	1000	.
Bias in point estimate	.01677	.00478	.00092	.00417	-.00119	.00425
Empirical standard error	.15112	.00338	.13201	.00295	.13443	.00301
% gain in precision relative to method CC	.	.	31.046	3.9394	26.368	3.8443
RMS model-based standard error	.1471	.00053	.13494	.0006	.13383	.00059
Relative % error in standard error	-2.6594	2.2055	2.2233	2.3323	-.44122	2.2695
Coverage of nominal 95% confidence interval	94.3	.73315	94.9	.69569	94.3	.73315
Power of 5% level test	94.6	.71473	96.9	.54808	96.3	.59692

CC lacks power – not surprising in view of its inefficiency.

Notes

- Can start with the data “long” or “wide”
- Can save results to file or load them into memory in order to create better-formatted output
- Can handle various simulation settings using `by(`*varlist*`)` option
- Program checks for extreme point estimates and SEs

Formulae

Assume that the true parameter is β and that the ith simulated data set ($i = 1$ to n) yields a point estimate $\hat{\beta}_i$ with standard error s_i. Write

$$
\begin{aligned}
\bar{\beta} &= \frac{1}{n}\sum_i \hat{\beta}_i \\
V_{\hat{\beta}} &= \frac{1}{n-1}\sum_i (\hat{\beta}_i - \bar{\beta})^2 \\
\overline{s^2} &= \frac{1}{n}\sum_i s_i^2 \\
V_{s^2} &= \frac{1}{n-1}\sum_i (s_i^2 - \overline{s^2})^2
\end{aligned}
$$

Performance of $\hat{\beta}$

Bias is defined as $\mathrm{E}\left[\hat{\beta}_i\right] - \beta$:

$$
\begin{aligned}
\text{estimated bias} &= \bar{\beta} - \beta \\
\text{MC error} &= \sqrt{V_{\hat{\beta}}/n}
\end{aligned}
$$

Precision is measured by the "empirical standard deviation" of $\hat{\beta}_i$:

$$
\begin{aligned}
\text{empirical SD} &= \sqrt{V_{\hat{\beta}}} \\
\text{MC error} &= \sqrt{V_{\hat{\beta}}/2(n-1)}
\end{aligned}
$$

- Assumes $\hat{\beta}_i$ is Normally distributed.

Method comparison

Relative precision of $\hat{\beta}_2$ compared with $\hat{\beta}_1$ is

$$\begin{aligned} \text{relative precision} &= V_{\hat{\beta}_1} / V_{\hat{\beta}_2} \\ \text{MC error} &\approx \frac{2V_{\hat{\beta}_1}}{V_{\hat{\beta}_2}} \sqrt{\frac{1-\rho_{12}^2}{n-1}} \end{aligned}$$

where ρ_{12} is the correlation of $\hat{\beta}_1$ with $\hat{\beta}_2$.

- Assumes $(\hat{\beta}_1, \hat{\beta}_2)$ are bivariate Normal, and uses a Taylor series approximation.

Performance of model-based standard error s_i

Average model-based standard error is

$$\bar{s} = \sqrt{\overline{s^2}}$$
$$\text{MC error} \approx \sqrt{V_{s^2}/4n\overline{s^2}}$$

- uses a Taylor series approximation

Relative error in the model-based standard error is

$$\text{relative error} = \bar{s}/\sqrt{V_{\hat{\beta}}} - 1$$
$$\text{MC error} \approx (\bar{s}/\sqrt{V_{\hat{\beta}}})\sqrt{V_{s^2}/(4n\bar{s}^4) + 1/2(n-1)}$$

- Assumes $\bar{s}$ and $V_{\hat{\beta}}$ are uncorrelated; uses a further Taylor approximation

Joint performance of $\hat{\beta}$ and s_i

Coverage and power are easy (means of binary variables).

Evaluations

- I've evaluated the MC errors by running 250 simulation studies and running `simsum` on its own results (!).
- All seems to work well, except in one case where the joint distribution of two parameter estimates was very far from Normal, and the MC error for the relative precision of the two methods was 3 times too small.

Syntax with main options

Wide format data: `simsum` *betavars* [`if`] [`in`] , [*options*]
Long format data: `simsum` *betavar* [`if`] [`in`] ,
`long(`*methodvar*`)` `id(`*idvar*`)` [*options*]

- *betavars*: point estimates to be evaluated
- `long(`*methodvar*`)` `id(`*idvar*`)`: identifies the method
- Standard errors can be identified by `se(`*varlist*`)`, `seprefix(`*string*`)` or `sesuffix(`*string*`)`
- `true(`*exp*`)`: specifies true value – needed for bias & coverage
- `mcse`
- `by(`*varlist*`)`
- `clear`: loads results into memory

Other options

- Identifying implausible values: `dropbig max(#) semax(#)`
- Specifying degree of freedom for CIs & tests: `df(string)`
- Reference method for method comparisons: `ref(varname)`
- Statistics required (default all): `bsims sesims bias empse relprec modelse relerror cover power`
- Output format: `format(`*format_expression*`)` `sepby(varlist) sep(num)`

2nd example: multiple parameter settings

dataset	truebeta	method	b	se
1	0	MI_NA	.1457046	.1191742
1	.5	MI_NA	.5891385	.164332
1	1	MI_NA	.9476269	.1791874
2	0	MI_NA	.0080439	.1491085
2	.5	MI_NA	.279084	.106511
2	1	MI_NA	.9994065	.1615081
3	0	MI_NA	.0843964	.1257705
3	.5	MI_NA	.6068171	.1359049
3	1	MI_NA	1.310416	.1761329

(1000 datasets)

2nd example: results

```
. simsum b, se(se) true(truebeta) mcse by(truebeta) bias empse modelse relerror
Starting to process results ...

  +-----------------------------------------------------------------------+
  |                     Statistic   truebeta            1         (MCse) |
  |-----------------------------------------------------------------------|
  |          Bias in point estimate          0   -.0004306    .0040768 |
  |          Bias in point estimate         .5     -.00047    .0040689 |
  |          Bias in point estimate          1   -.0299813    .0045115 |
  |-----------------------------------------------------------------------|
  |        Empirical standard error          0    .1289194    .0028842 |
  |        Empirical standard error         .5    .1286701    .0028786 |
  |        Empirical standard error          1    .1426651    .0031917 |
  |-----------------------------------------------------------------------|
  |  RMS model-based standard error          0    .1318145    .0005536 |
  |  RMS model-based standard error         .5    .1343978    .0005811 |
  |  RMS model-based standard error          1    .1525643    .0008317 |
  |-----------------------------------------------------------------------|
  | Relative % error in standard error       0     2.24561     2.32739 |
  | Relative % error in standard error      .5    4.451456    2.380013 |
  | Relative % error in standard error       1    6.938776     2.46243 |
  +-----------------------------------------------------------------------+
```

Conclusion / discussion

- Alternative formulae?

Available from

- `net from http://www.mrc-bsu.cam.ac.uk/BSUsite/Software/pub/software/stata/`
- Will be on `ssc` before long