MRC Clinical Trials Unit

UCL

Coding robust simulation studies in Stata

Ella Marley-Zagar, Tim Morris
MRC Clinical Trials Unit at UCL

e.marley-zagar@ucl.ac.uk
tim.morris@ucl.ac.uk @tmorris_mrc

Stata Biostatistics and Epidemiology Virtual Symposium 23 Feb 2023

Smarter Studies
Global Impact
Better Health

Plan of talk

Introduce simulation studies and **ADEMP**

Some tips when something has gone wrong

Analysis of simulation studies using `siman`

MRC Clinical Trials Unit

Introducing simulation studies

Simulation studies are used in a variety of disciplines.

They are an important and useful tool which enables researchers to compare different statistical methods and understand their properties.

What can simulation studies be used for?

- Check that code does the intended analysis
- Check robustness of programs
- Better understand statistical concepts
- Understand new commands
- Check algebra
- **Evaluate a new statistical method**
- **Compare statistical methods head-to-head**
- **Calculate sample size / power**
- …and more

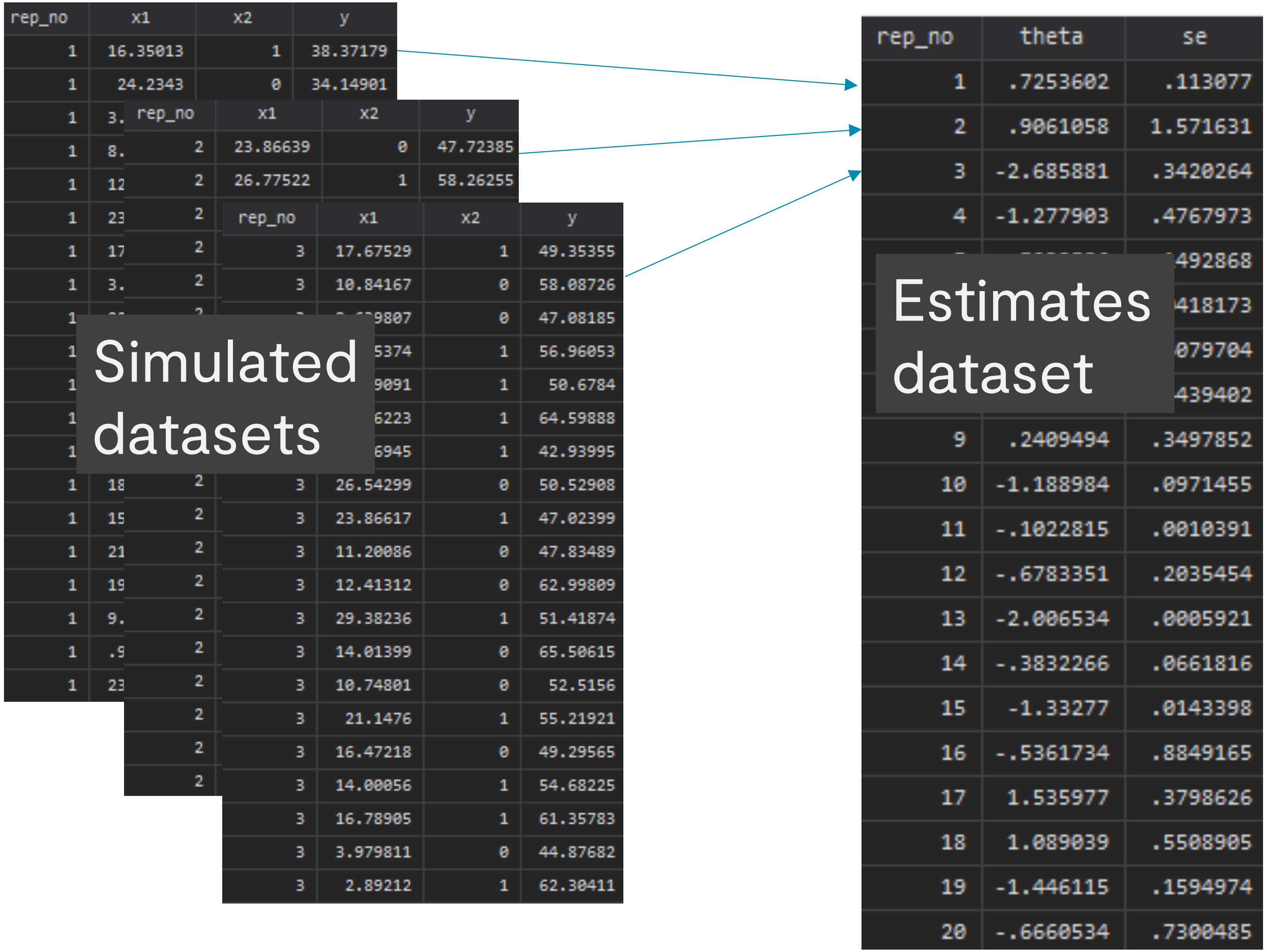
Simulation study set up: ADEMP

In a simulation study, data is often generated from some distribution (so often we know the truth), the data is analysed and compared with the known truth. Convenient to plan using 'ADEMP' structure.

Aims: What question(s) the simulation study addresses
Data-generating mechanisms (DGMs): How the simulated datasets are to be generated
Estimands/targets: quantities to be estimated by the analysis
Methods of analysis: How a given simulated dataset is to be analysed
Performance measures: How the performance of the methods of analysis is to be summarised
(Also worth thinking about implementation and reporting)

MRC Clinical Trials Unit

Two types of dataset



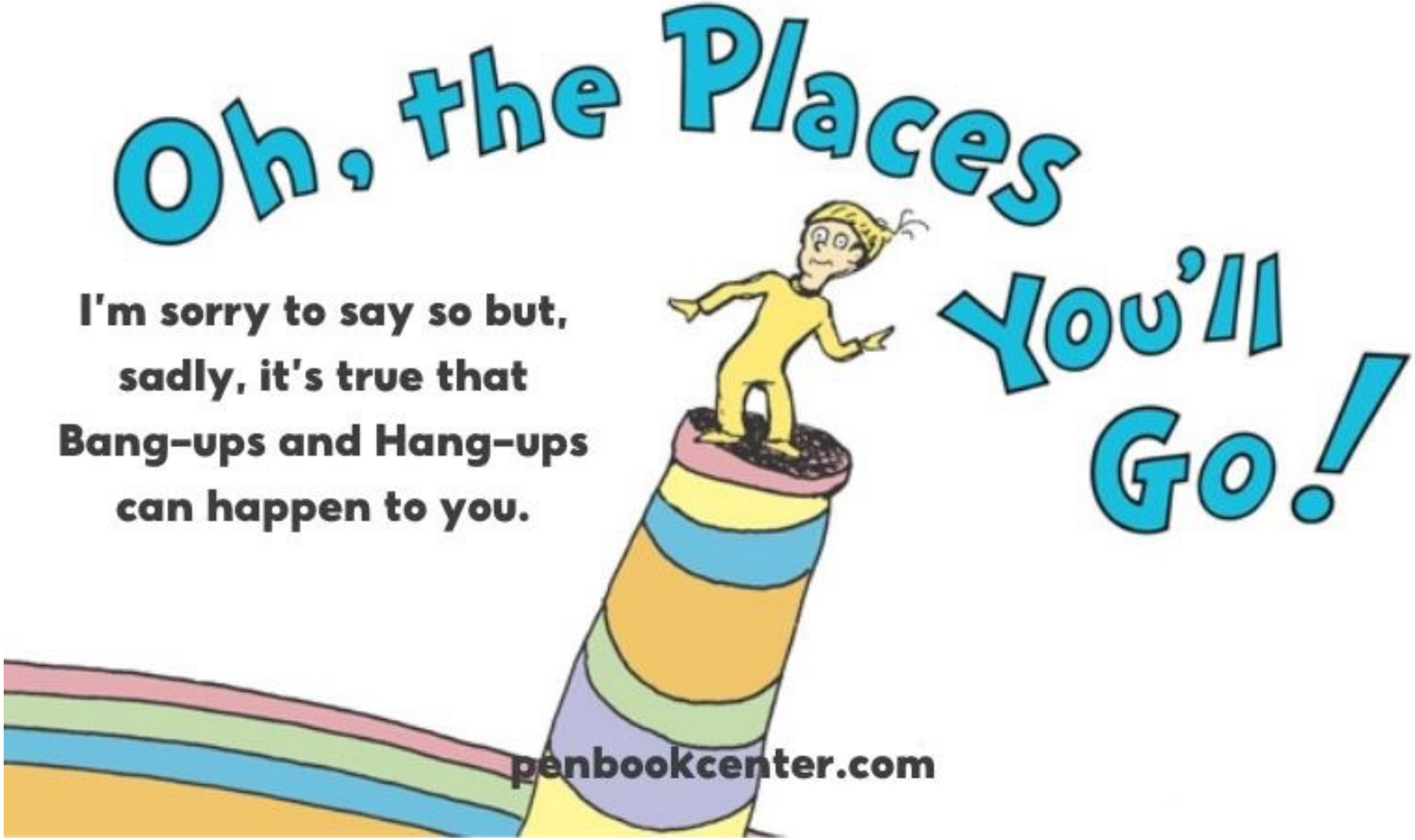
I don’t believe it!

We often find ourselves in the position of seeing some simulation results – our own, a colleague’s or a student’s – and thinking:

‘I’M SCEPTICAL OF THESE RESULTS’

or

‘I DON’T BELIEVE THIS’


Oh, the Places You'll Go!
I'm sorry to say so but,
sadly, it's true that
Bang-ups and Hang-ups
can happen to you.
penbookcenter.com

But is it wrong?

The fact that we are sceptical does not necessarily mean the result is wrong

That being said, in many cases it is.

We need to make sure we have exhausted possible errors to the extent possible.

MRC Clinical Trials Unit

Three ‘phases’ of simulation study

Design

Getting ADEMP right!

Conduct

Writing code to generate data, analyse, store results

Analysis

Computing performance measures from an estimates dataset

A series of tips

Based on ‘How to check a simulation study’ osf.io/cbr72/ (pre-print).

Problems often arise when we have many DGMs, estimands and/or methods of analysis.

Issues that involve Stata can occur in any of the three stages.

Running example: White & Carlin

Loosely based on:
White IR, Carlin JB. Bias and efficiency of multiple imputation compared with complete-case analysis for missing covariate values. *Statistics in Medicine* 2010; 29: 2920–2931.

MRC Clinical Trials Unit

Running example: White & Carlin

Aim	To compare multiple imputation with complete case analysis.
Data-generating mechanism (rough)	Quantitative confounder C is drawn from a standard Normal distribution. Binary exposure E and binary outcome D are drawn from logistic models depending on C (so E does not cause D). Data are made missing on C, initially using a missing completely at random model. Parameters to be varied are the marginal probabilities of E and D, the strength of the dependence of E and D on C, and the missing data mechanism. The sample size of 500 is fixed.

Running example: White & Carlin

Estimand	The log odds ratio between E and D, conditional on C.
Methods of analysis	1. Analysis of full data before data deletion in C. 2. Analysis of complete cases (excluding cases with missing C). 3. Multiple imputation of missing values of C (various imputation models may be used).
Performance measures	Bias Empirical standard error Relative error in model-based standard error Coverage
Implementation?	1,000 repetitions

MRC Clinical Trials Unit

Two tips

TIP: If possible, include a ‘benchmark’ setting with known properties – either known theoretically or ‘known’ from someone else’s simulation study. In our missing data example, we have included ‘full data’ as this should give better performance than any analysis with incomplete data.

TIP: Write well-structured code *from the start!*

Well-structured code?

Two criteria for ‘well structured code’:

1. For a single repetition, to the extent possible, separate data generation, data analysis and ‘posting’/storage of result/s
2. Produce a well-structured estimates dataset

MRC Clinical Trials Unit

Well structured code: a suggestion

Write:

- A program that generates data
- A program that analyses data and returns (or directly posts) results of interest
- A program that repeatedly calls these and structures the results sensibly an estimates dataset

DGM program

```
program define gendata
    version 17
    syntax, obs(int) logite(string) logitd(string) pmiss(string)
    clear
    set obs `obs'
    drawnorm Ctrue
    gen E = runiform() < invlogit(`logite')
    gen D = runiform() < invlogit(`logitd')
    gen Cobs = Ctrue if runiform()>=`pmiss'
end
```

Analysis program

```
program define anadata
    version 14
    syntax, rep(int 0) [ post(string) ]

    * Method 1: full data before data deletion
    logit D E Ctrue
    if !mi("`post'") post `post' (`rep') ("Full") (_b[E]) (_se[E]) (e(N)) (.)
    * we are posting:            rep     method   est     se         N      df
    …

    mi impute regress Cobs D##E, add(5)
    mi estimate, post: logit D E Cobs
    if !mi("`post'") post `post' (`rep') ("MI") (_b[E]) (_se[E]) (e(N)) ///
      (e(df_mi)[1,"D:E"])
end
```

MRC Clinical Trials Unit

Tip: study a single very large dataset

Ideally, 'fit back' the data-generating model and check that number of observations is correct, parameters are close to inputs, true values of parameters are trapped by confidence intervals, etc.

```
gendata , obs(1000000)
```

MRC Clinical Trials Unit

Tip: run with three repetitions

1. Verify that your estimates dataset is 'well-structured'
2. Verify that the second and third repetitions produce different data and results

Why? Sometimes simulation code wrongly sets the seed within a program starting the first repetition. This is sometimes done in some program called by `gendata` or `anadata`.

If (for example) this is done at the end of a repetition, the second and third repetitions will produce identical data and results.

MRC Clinical Trials Unit

Tip: run with three repetitions

```
. set rngstream 1
. set seed 576819506
. local nreps 3

. tempname est
. postfile `est' int(rep) str3(method) float(b se) int(N) float(df)
> using estimates, replace

. forvalues i=1/`nreps' {
.     gendata, obs(500) logite(-1+C) logitd(-1+C) pmiss(.3)
.     anadata, rep(`i') post(`est')
. }
. postclose `est'
```

MRC Clinical Trials Unit

Tip: run with three repetitions

```
. list, sepby(rep)

     +---------------------------------------------------------+
     | rep   method           b          se     N          df |
     |---------------------------------------------------------|
  1. |   1     Full   -.1081978    .2457296   500           . |
  2. |   1      CCA   -.3064098     .314233   337           . |
  3. |   1       MI    -.207532    .2880667   500    71.65478 |
     |---------------------------------------------------------|
  4. |   2     Full    .0867077    .2260111   500           . |
  5. |   2      CCA    .1381473    .2631429   349           . |
  6. |   2       MI    .1110958    .2330245   500    1092.247 |
     |---------------------------------------------------------|
  7. |   3     Full    .1232898    .2289063   500           . |
  8. |   3      CCA   -.1608068    .2822033   348           . |
  9. |   3       MI    .0335785    .2445695   500    508.5607 |
     +---------------------------------------------------------+
```

MRC Clinical Trials Unit

UCL

Tip: anticipate analysis failures

Code should be written to capture the error so the simulation does not halt.

The failure of a method should be stored – and investigated – along with its error code.

```
. capture noisily logit D E Cobs

. if _rc==0 & !mi("`post'") post `post' (`rep') ("CCA")
> (_b[E]) (_se[E]) (e(N)) (.) (_rc)

. if _rc>0 & !mi("`post'") post `post' (`rep') ("CCA") (.)
> (.) (.) (.) (_rc)
```

MRC Clinical Trials Unit

Tip: make it easy to recreate any simulated dataset

Your estimates dataset needs a unique identifier for each result (I have a repetition number and a variable giving the method of analysis used)

If we can recreate a specific result, we can explore method failures, outliers, etc. Two ways:

1. Store the RNG state at the start of a repetition: see github.com/tpmorris/TheRightWay/
2. Save every simulated dataset (if the analysis has a stochastic element, such as multiple imputation or bootstrap, you still need the RNG state)

MRC Clinical Trials Unit

Analysis tips

The pre-print gives some tips for understanding issues with analysis, which we won't recap here (more focused on understanding results than on 'coding robust simulation studies').

If you have a well-structured estimates dataset, Ella is now going to talk through a suite of commands that analyse a simulation study. In particular, it automates unpleasant data wrangling, analysis and several graphics.

MRC Clinical Trials Unit

ELLA’S SLIDES

MRC
Clinical
Trials Unit

Introducing `siman`

We introduce the `siman` suite which has been created to assist the analysis of simulation results.

A set of Stata programs that offer data manipulation, analysis and graphics to process, explore and visualise the results of simulation studies.

The new siman program described here is available at *https://github.com/UCL/siman*

Work with Ian White and Tim Morris, MRC Clinical Trials Unit at UCL

MRC Clinical Trials Unit

Types of dataset: Estimates data

- Estimates data set: results from analysing multiple simulated data sets, with data relating to different statistics (e.g. point estimate (est), se) for each simulated data set.
- repetition (rep): simulation number

	rep	dgm	estimand	method	est	se	true
1	1	1	beta	A	-.1433122	.0773843	0
2	1	1	beta	B	-.23375	.1103629	0
3	1	1	gamma	A	-.051712	.0809558	0
4	1	1	gamma	B	-.1375358	.1166905	0
5	1	2	beta	A	-.1134563	.0945678	0
6	1	2	beta	B	-.1543437	.1399313	0
7	1	2	gamma	A	-.0597119	.0934798	0
8	1	2	gamma	B	-.1588126	.1347426	0
9	2	1	beta	A	-.1508732	.0768133	0
10	2	1	beta	B	-.0784156	.1087276	0
11	2	1	gamma	A	-.0296873	.0737869	0
12	2	1	gamma	B	.1310411	.1115952	0
13	2	2	beta	A	-.1337332	.0927938	0
14	2	2	beta	B	-.1540722	.1323587	0
15	2	2	gamma	A	-.0342799	.0852017	0
16	2	2	gamma	B	.1513133	.128859	0

MRC Clinical Trials Unit

Types of dataset: long and wide

For the input estimates data, there are 3 formats permitted by the `siman` suite:

1. Long for both targets and methods: longlong
2. Wide for both target and methods : widewide
3. Long for targets, wide for methods : longwide

	r	seBgamma	true
1	1	.1166905	0
2	1	.1347426	0
3	2	.1115952	0
4	2	.128859	0

	rep	dgm	estimand	estA	seA	estB	seB	true
1	1	1	beta	-.1433122	.0773843	-.23375	.1103629	0
2	1	1	gamma	-.051712	.0809558	-.1375358	.1166905	0
3	1	2	beta	-.1134563	.0945678	-.1543437	.1399313	0
4	1	2	gamma	-.0597119	.0934798	-.1588126	.1347426	0
5	2	1	beta	-.1508732	.0768133	-.0784156	.1087276	0
6	2	1	gamma	-.0296873	.0737869	.1310411	.1115952	0
7	2	2	beta	-.1337332	.0927938	-.1540722	.1323587	0
8	2	2	gamma	-.0342799	.0852017	.1513133	.128859	0

	rep	dgm	estimand	method	est	se	true
9	2	1	beta	A	-.1508732	.0768133	0
10	2	1	beta	B	-.0784156	.1087276	0
11	2	1	gamma	A	-.0296873	.0737869	0
12	2	1	gamma	B	.1310411	.1115952	0
13	2	2	beta	A	-.1337332	.0927938	0
14	2	2	beta	B	-.1540722	.1323587	0
15	2	2	gamma	A	-.0342799	.0852017	0
16	2	2	gamma	B	.1513133	.128859	0

MRC Clinical Trials Unit

Performance Measures

Typically a statistical method outputs an estimate $\hat{\theta}$, its standard error $\widehat{se\ (\hat{\theta})}$and a confidence interval $(\hat{\theta}\ low, \hat{\theta}\ upp)$

The following performance measures may be of interest:

- Properties of estimate $\hat{\theta}$
  - Bias
  - Empirical SE
  - MSE
- Properties of SE
  - Average model-based SE
- Properties of confidence interval
  - Coverage
  - Power

MRC Clinical Trials Unit

Types of Data Sets: OUTPUT Performance Measures

Performance measures data set: results from analysing an estimates data set, with data relating to different performance measures (e.g. bias, coverage) summarised over estimates data sets for different data generating mechanisms.

	rep	dgm	estimand	method	est	se	true	_perfmeascode	_dataset
1	Non-missing point estimates	1	beta	A	1000	.	.	bsims	Performance
2	Non-missing point estimates	1	beta	B	1000	.	.	bsims	Performance
3	Non-missing standard errors	1	beta	A	1000	.	.	sesims	Performance
4	Non-missing standard errors	1	beta	B	1000	.	.	sesims	Performance
5	Bias in point estimate	1	beta	A	-.0043991	.0024993	.	bias	Performance
6	Bias in point estimate	1	beta	B	-.0025973	.0035644	.	bias	Performance
7	Mean of point estimate	1	beta	A	-.0043991	.0024993	.	mean	Performance
8	Mean of point estimate	1	beta	B	-.0025973	.0035644	.	mean	Performance
9	Empirical standard error	1	beta	A	.0790336	.0017681	.	empse	Performance
10	Empirical standard error	1	beta	B	.1127159	.0025217	.	empse	Performance
11	% precision gain relative to method A	1	beta	A	.	.	.	relprec	Performance
12	% precision gain relative to method A	1	beta	B	-50.8353	2.28676	.	relprec	Performance
13	Mean squared error	1	beta	A	.0062594	.0002881	.	mse	Performance
14	Mean squared error	1	beta	B	.0126989	.0006187	.	mse	Performance
15	Root mean squared error	1	beta	A	.0791165	.0230148	.	rmse	Performance
16	Root mean squared error	1	beta	B	.1126895	.0243622	.	rmse	Performance
17	RMS model-based standard error	1	beta	A	.0787452	.0001572	.	modelse	Performance
18	RMS model-based standard error	1	beta	B	.1136343	.0003405	.	modelse	Performance
19	Mean conf. interval width	1	beta	A	.3080681	.0006123	.	ciwidth	Performance
20	Mean conf. interval width	1	beta	B	.4434751	.0013217	.	ciwidth	Performance
21	Relative % error in standard error	1	beta	A	-.3649609	2.237882	.	relerror	Performance
22	Relative % error in standard error	1	beta	B	.8147949	2.275551	.	relerror	Performance
23	% coverage of nominal 95% conf. interval	1	beta	A	94.7	.7084563	.	cover	Performance
24	% coverage of nominal 95% conf. interval	1	beta	B	95.7	.6414907	.	cover	Performance

MRC Clinical Trials Unit

Setting up `siman`: long-long format

`siman setup` takes the estimates data set held in memory, checks the data, reformats it if necessary and attaches characteristics to the data set available for use across multiple sessions.

	rep	dgm	estimand	method	est	se	true
1	1	1	beta	A	-.1433122	.0773843	0
2	1	1	beta	B	-.23375	.1103629	0
3	1	1	gamma	A	-.051712	.0809558	0
4	1	1	gamma	B	-.1375358	.1166905	0
5	1	2	beta	A	-.1134563	.0945678	0
6	1	2	beta	B	-.1543437	.1399313	0
7	1	2	gamma	A	-.0597119	.0934798	0
8	1	2	gamma	B	-.1588126	.1347426	0
9	2	1	beta	A	-.1508732	.0768133	0
10	2	1	beta	B	-.0784156	.1087276	0
11	2	1	gamma	A	-.0296873	.0737869	0
12	2	1	gamma	B	.1310411	.1115952	0
13	2	2	beta	A	-.1337332	.0927938	0
14	2	2	beta	B	-.1540722	.1323587	0
15	2	2	gamma	A	-.0342799	.0852017	0
16	2	2	gamma	B	.1513133	.128859	0

```
siman setup, rep(rep) dgm(dgm) target(estimand)
            method(method) est(est) se(se) true(true)
```

Setting up `siman`: wide-wide format

	rep	dgm	estAbeta	seAbeta	estBbeta	seBbeta	estAgamma	seAgamma	estBgamma	seBgamma	true
1	1	1	-.1433122	.0773843	-.23375	.1103629	-.051712	.0809558	-.1375358	.1166905	0
2	1	2	-.1134563	.0945678	-.1543437	.1399313	-.0597119	.0934798	-.1588126	.1347426	0
3	2	1	-.1508732	.0768133	-.0784156	.1087276	-.0296873	.0737869	.1310411	.1115952	0
4	2	2	-.1337332	.0927938	-.1540722	.1323587	-.0342799	.0852017	.1513133	.128859	0

```
siman setup, rep(rep) dgm(dgm) target(beta gamma)
    method(A B) est(est) se(se) true(true) order(method)
```

MRC Clinical Trials Unit

UCL

Setting up siman: `siman describe`

```
                    SUMMARY OF DATA
_____________________________________________________

The siman format is:               format 1: long-long
The format for targets is:         long
The format for methods is:         long
The number of targets is:          2
The target values are:             beta gamma

The number of methods is:          2
The method values are:             A B

Data generating mechanism (dgm)
The total number of dgms is:       2
The dgm variables (# levels):      dgm (2)

Estimates are contained in the dataset

The estimates variable is:         est
The se variable is:                se
The df variable is:                N/A
The ci variables are:              N/A
The p variable is:                 N/A
The true variable is:              true
_____________________________________________________
```

MRC
Clinical
Trials Unit

Exploring the estimates data: `siman reshape`

	rep	dgm	estimand	estA	seA	estB	seB	true
1	1	1	beta	-.1433122	.0773843	-.23375	.1103629	0
2	1	1	gamma	-.051712	.0809558	-.1375358	.1166905	0
3	1	2	beta	-.1134563	.0945678	-.1543437	.1399313	0
4	1	2	gamma	-.0597119	.0934798	-.1588126	.1347426	0
5	2	1	beta	-.1508732	.0768133	-.0784156	.1087276	0
6	2	1	gamma	-.0296873	.0737869	.1310411	.1115952	0
7	2	2	beta	-.1337332	.0927938	-.1540722	.1323587	0
8	2	2	gamma	-.0342799	.0852017	.1513133	.128859	0

```
siman reshape, longlong
siman reshape, longwide
```

	rep	dgm	estimand	method	est	se	true
1	1	1	beta	A	-.1433122	.0773843	0
2	1	1	beta	B	-.23375	.1103629	0
3	1	1	gamma	A	-.051712	.0809558	0
4	1	1	gamma	B	-.1375358	.1166905	0
5	1	2	beta	A	-.1134563	.0945678	0
6	1	2	beta	B	-.1543437	.1399313	0
7	1	2	gamma	A	-.0597119	.0934798	0
8	1	2	gamma	B	-.1588126	.1347426	0
9	2	1	beta	A	-.1508732	.0768133	0
10	2	1	beta	B	-.0784156	.1087276	0
11	2	1	gamma	A	-.0296873	.0737869	0
12	2	1	gamma	B	.1310411	.1115952	0
13	2	2	beta	A	-.1337332	.0927938	0
14	2	2	beta	B	-.1540722	.1323587	0
15	2	2	gamma	A	-.0342799	.0852017	0
16	2	2	gamma	B	.1513133	.128859	0

MRC Clinical Trials Unit

Exploring the estimates data: example data from earlier

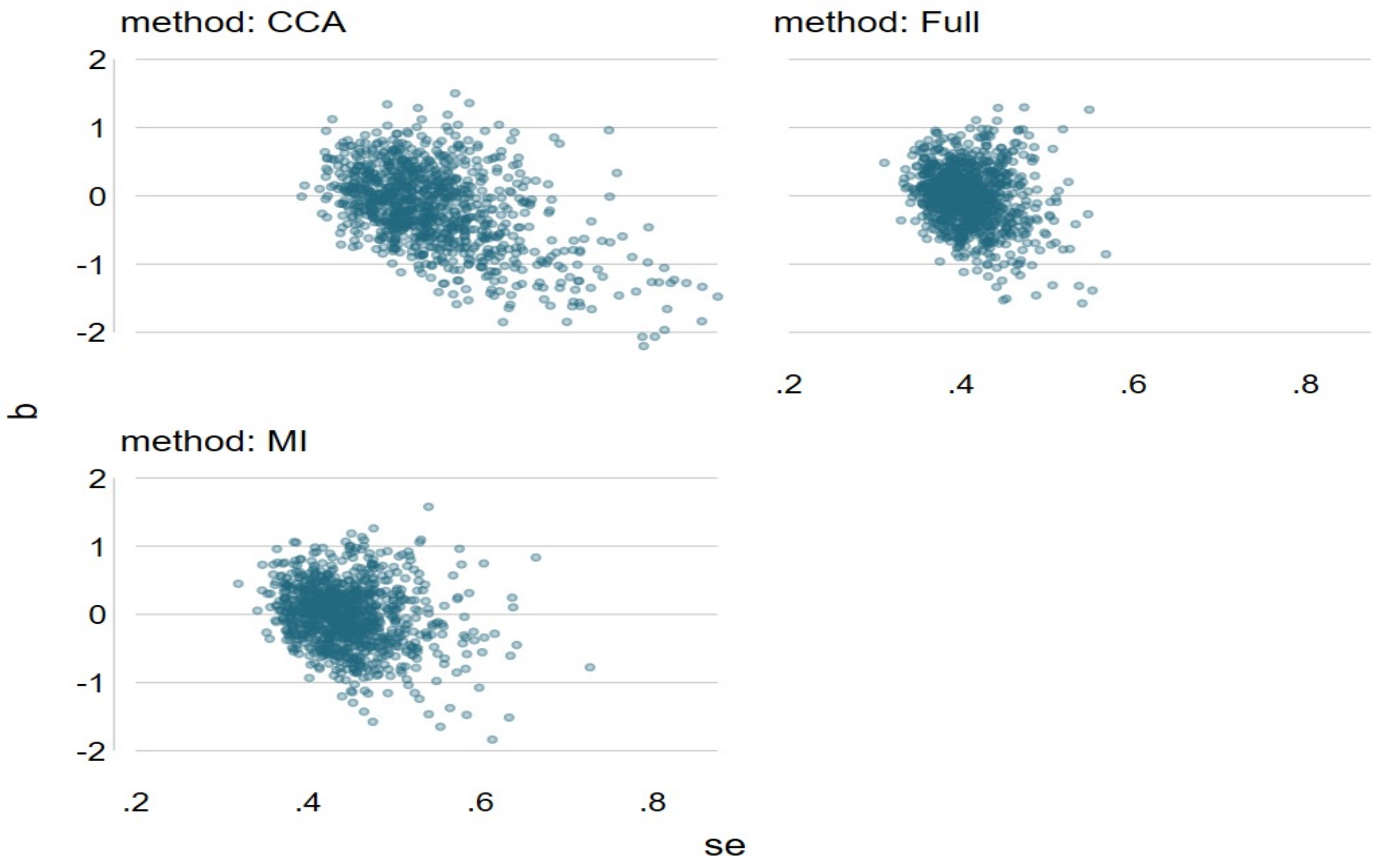
simcheck paper, White et al. (preprint https://osf.io/cbr72/)

	rep	method	b	se	df	true	dgm
1	1	Full	-.3664537	.4296166	.	0	2
2	1	CCA	-.6404278	.5808458	.	0	2
3	1	MI	-.7200662	.4606918	1350.781	0	2
4	2	Full	.4978587	.4003829	.	0	2
5	2	CCA	.7493323	.4796979	.	0	2
6	2	MI	.5480893	.4190439	527.496	0	2
7	3	Full	-.0834949	.335249	.	0	2
8	3	CCA	-.4181158	.4036536	.	0	2
9	3	MI	-.2691882	.3600377	362.7664	0	2
10	4	Full	-.4460969	.370079	.	0	2
11	4	CCA	-.371215	.3978041	.	0	2
12	4	MI	-.3969929	.3777045	1664.085	0	2
13	5	Full	-.4702572	.3891214	.	0	2
14	5	CCA	-.3799451	.4851103	.	0	2
15	5	MI	-.4747566	.4262364	174.7589	0	2
16	6	Full	.0816109	.385957	.	0	2
17	6	CCA	-.1413643	.4341343	.	0	2
18	6	MI	.0940165	.4007708	841.9497	0	2

MRC Clinical Trials Unit

Exploring the estimates data: `siman scatter`

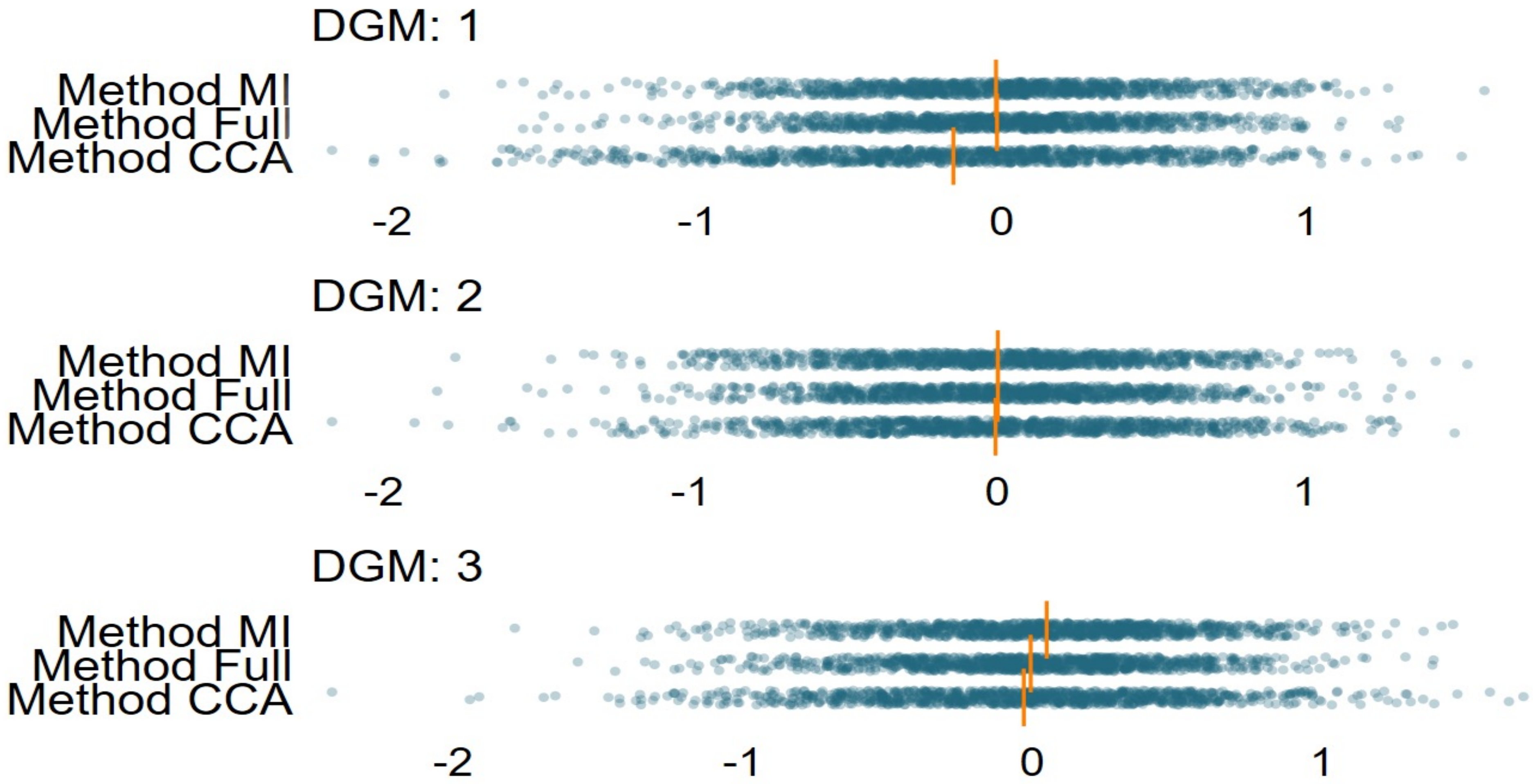
```
siman scatter [if] [in] [, options]

siman scatter if dgm == 1
```



MRC Clinical Trials Unit

Exploring the estimates data: `siman swarm`

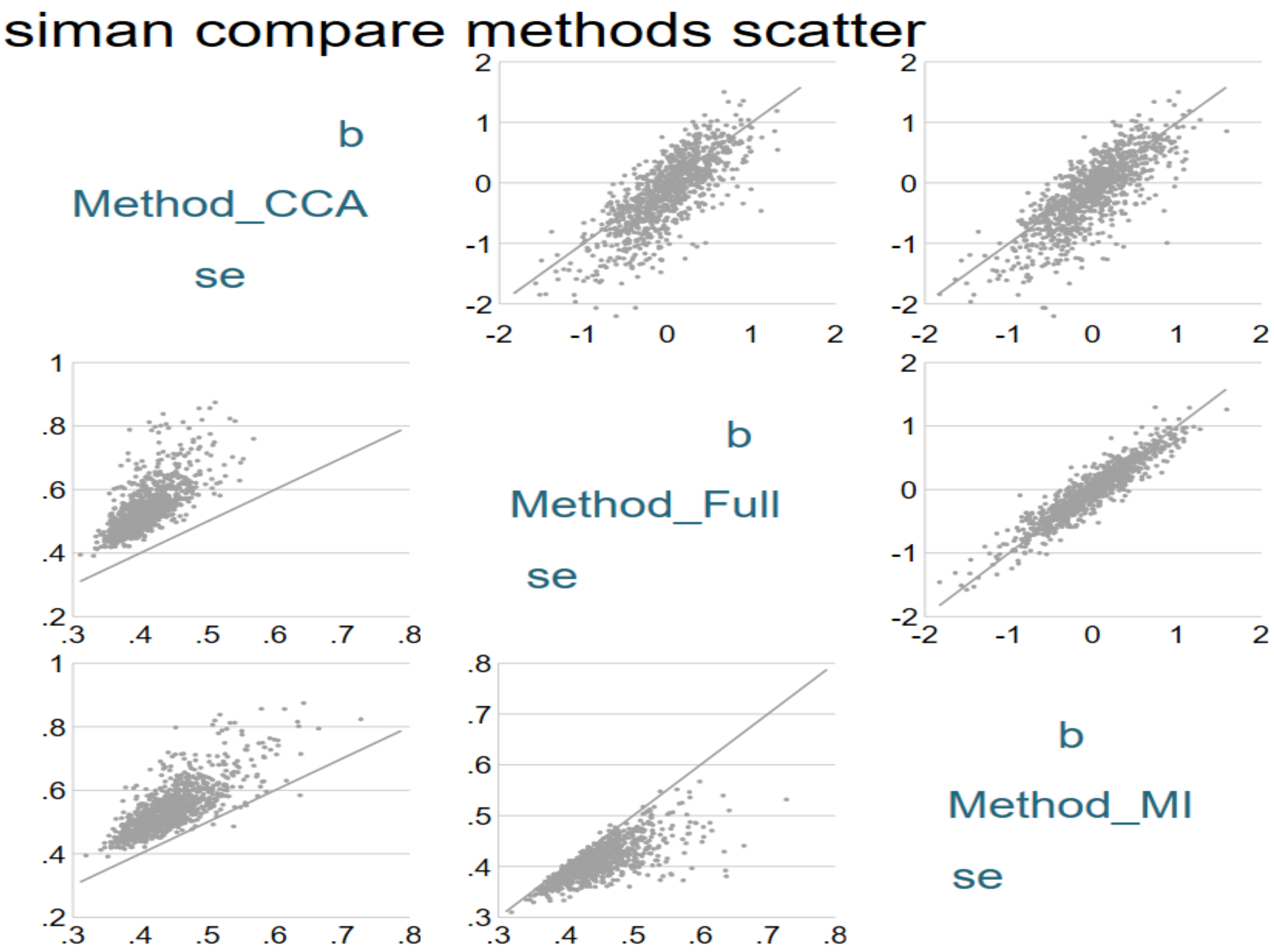
```
siman swarm [if] [in] [, options]
```



MRC Clinical Trials Unit

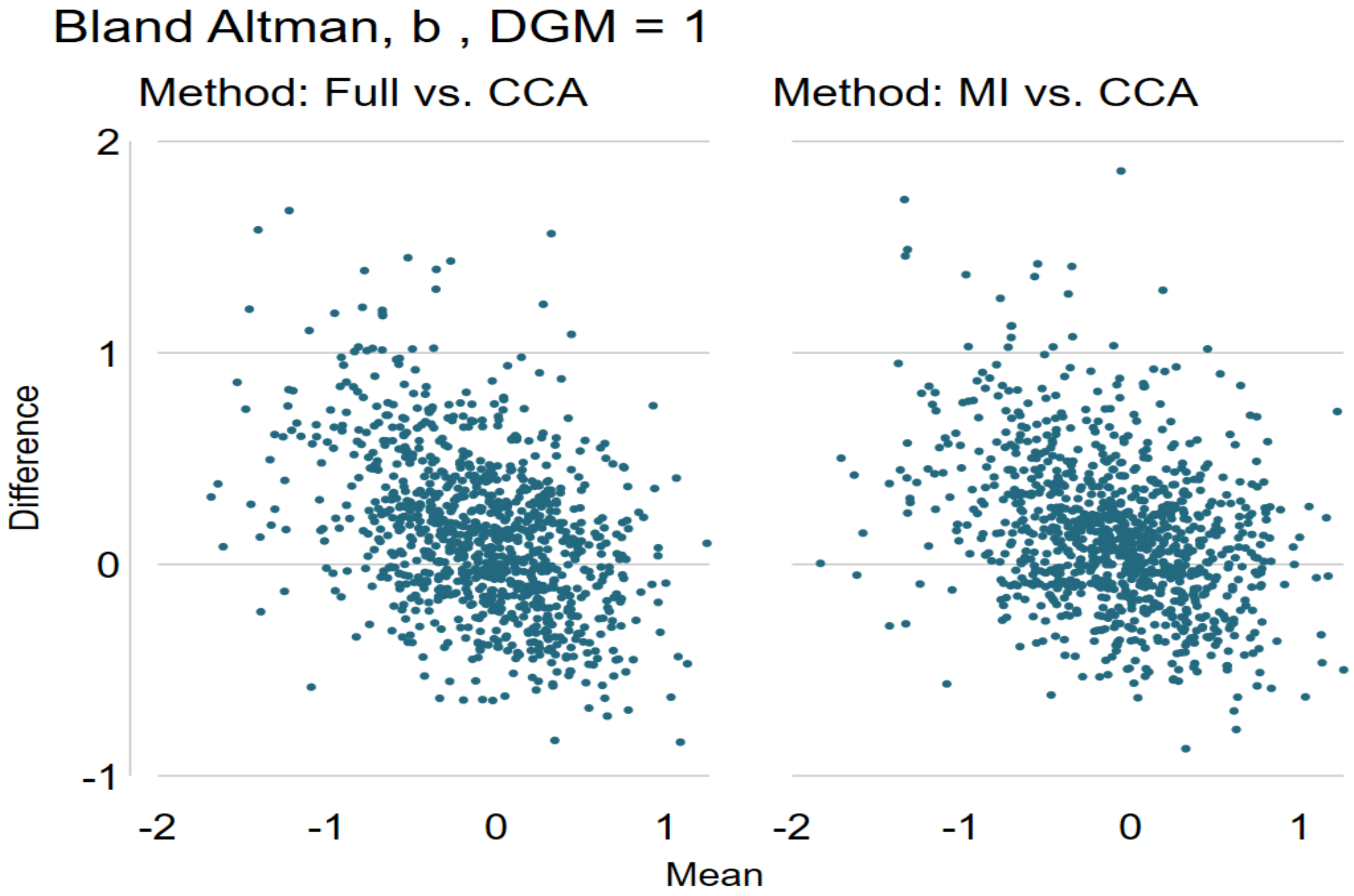
Exploring the estimates data: `siman comparemethodsscatter`

```
siman comparemethodsscatter [estimate|se] [if] [in] [, options]
```



Exploring the estimates data: `siman blandaltman`

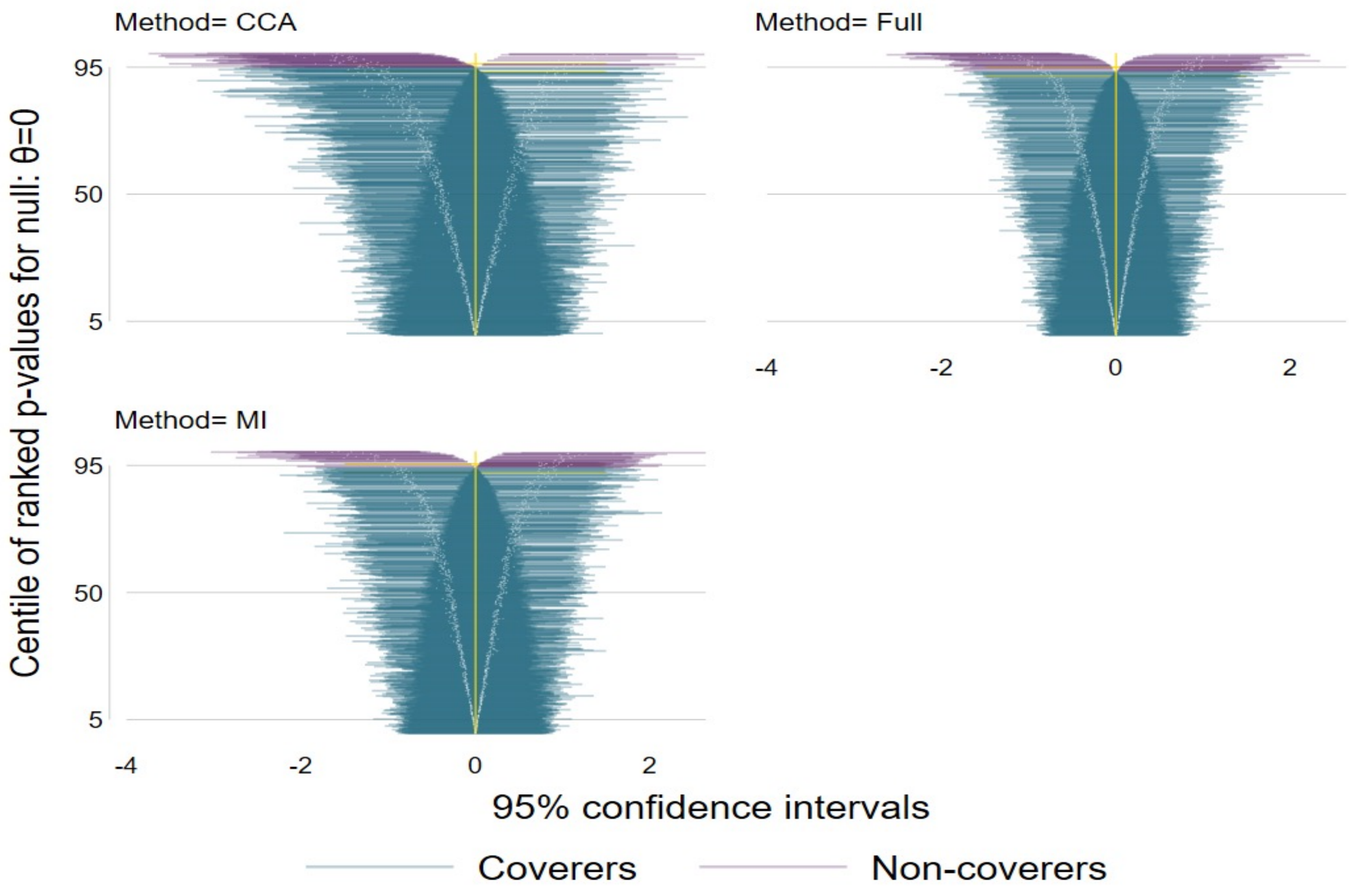
```
siman blandaltman [if] [in] [, options]
```



MRC Clinical Trials Unit

Exploring the estimates data: `siman zipplot`

```
siman zipplot [if] [in] [, options]
```



MRC Clinical Trials Unit

Creating performance measures

Once `siman setup` has been run, performance measures can be created using the command `siman analyse`

```
siman analyse [if], [performancemeasures perfonly
replace]
```

`performancemeasures` as per `simsum`. If none of the following options are specified, then all available performance measures are computed.....

[illegible] standard error of the reference method. This calculation is slow: omitting it can reduce run time by up to 90%.

Creating performance measures cont.

`perfonly` the program will automatically append the performance measures data to the estimates data, unless the user specifies `perfonly` for performance measures only.

`replace` if `siman analyse` has already been run and the user specifies it again then they must use the replace option, to replace the existing performance measures in the data set.

MRC Clinical Trials Unit

Creating performance measures: `siman table`

```
siman table [performancemeasures] [if], [column(varname)]
siman table bias
```

```
. siman table bias
```

dgm and performance measure	method CCA	Full	MI
1			
bias	-.160113	-.0168955	-.0203649
	.0183634	.0137363	.0146205
2			
bias	-.0075347	-.0000126	.0000898
	.0163235	.0134727	.0143525
3			
bias	-.0301532	-.0063853	.0486318
	.0170389	.0136254	.0145317

NOTE: Where there are 2 entries in the table, the first entry is the performance measure and the second entry is its Monte Carlo error.

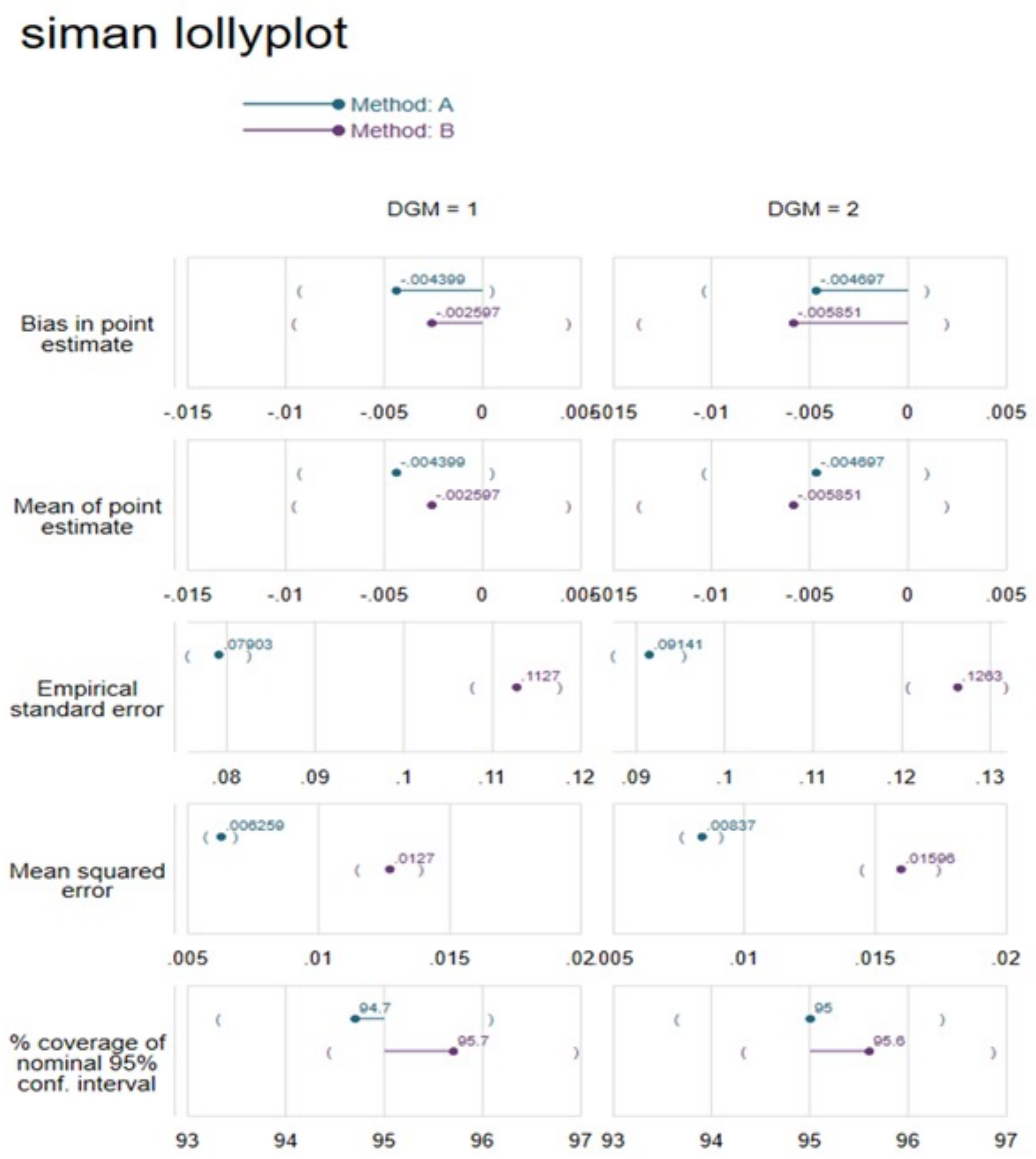
MRC Clinical Trials Unit

Creatin

lollyp

hs: siman

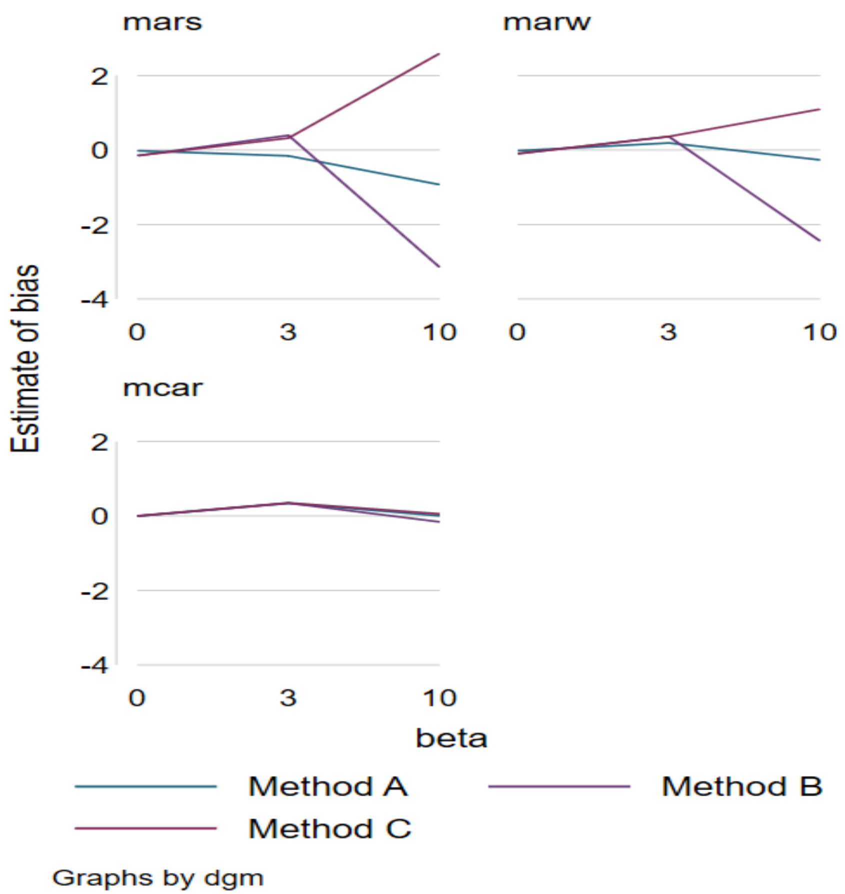
siman lolly



MRC Clinical Trials Unit

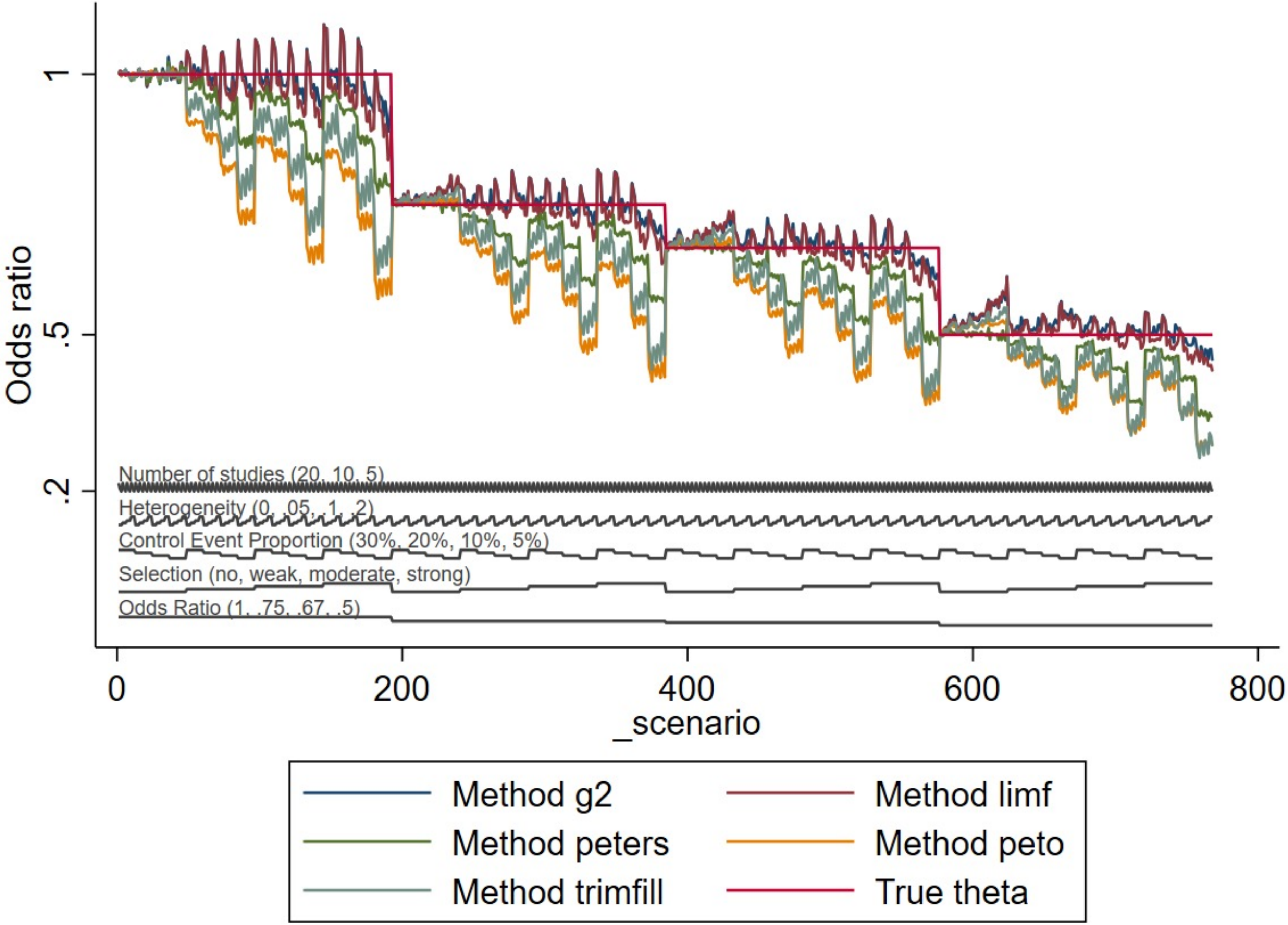
Creating performance measures graphs: `siman trellis`

```
siman trellis [performancemeasures] [if] [, options]
```



Creati n nestl

siman nest



MRC Clinical Trials Unit

Software testing

We have a program of testing our unit's software.

The siman suite is being tested on numerous data set formats by EMZ, IW, TM: long-long, long-wide, wide-wide, numeric and string, multiple methods, targets and dgms.

Error checking to make sure it fails when it is meant to (with a sensible error message).

Roundup 1

We've given a collection of tips that help to find/avoid errors in simulation studies.

The `siman` suite automates the data wrangling and analysis that often leads to problems. People often, for example:

1. Tangle up DGMs with methods
2. Compute average of model SEs (instead of root-mean variance)
3. Forget to separate out different estimands, DGMs or methods

Roundup 2

Doing all these things may simply be a matter of reassuring yourself that a particular result is ‘real’ and not goofing the code.

One remaining (unpopular?) tip if you don’t believe your results: write at least some of your code in another software package. In the pre-print, there is code for both Stata and R, and the different approaches to handling near-separation are very interesting.

Acknowledgements

Ian White

Tra My Pham

Jingyi Xuan

Michael Crowther

Matteo Quartagno

MRC Clinical Trials Unit

References

I. R. White, T. M. Pham, M. Quartagno, T. P. Morris. 2023. How to check a simulation study. Pre-print [under review]: doi.org/10.31219/osf.io/cbr72

T. P. Morris, I. R. White, M. J. Crowther. 2019. Using simulation studies to evaluate statistical methods. *Statistics in Medicine* 38(11): 2074–2102.

G. Rücker, G. Schwarzer. 2014. Presenting simulation results in a nested loop plot. *BMC Medical Research Methodology* 14(1): 1–8.

I. R. White 2010. `simsum`: Analyses of simulation studies including Monte Carlo error. *The Stata Journal* 10(3): 369–385.

I. R. White, J. B. Carlin. Bias and efficiency of multiple imputation compared with complete-case analysis for missing covariate values. *Statistics in Medicine* 2010; 29: 2920–2931.