# Using simulation studies to evaluate statistical methods in Stata: A tutorial

**Tim Morris**, MRC Clinical Trials Unit at UCL
**Ian White**, MRC Biostatistics Unit
**Michael Crowther**, University of Leicester


# Tutorial outline

Introduction

Planning with ADMEP

Coding

Pseudo-randomness, seeds and states

Analysis of simulation studies

Conclusions

# Introduction

# Introduction

We regularly run a course on using simulation to evaluate statistical methods

This talk will go through some of the key points of the course, with a focus on concepts, Stata issues and avoiding trip-ups

I will do this through a running example

# ‘*Not*’s

This talk is **not**:

- A condensed version of the course (if you want the course, come to the next one or invite us!)
- About how to generate specific types of data
- Delving into ‘realistic’ or ‘unrealistic’ data structures

# ‘*Is*’s

This talk **is** about:

- Treating a simulation study as an proper experiment, not just something dashed-off
- A structured approach to planning, based on ADMEP (an awkward initialism for the elements)
- Presenting measures of uncertainty
- Exploring how we might present simulation results

# Uses of simulation

Simulation can be used for all sorts of things in statistical research:

- Check that code does the intended analysis
- Check robustness of our programs
- Understand concepts or commands
- Check algebra (esp. approximations)
- **Evaluation of a method**
- **Comparison of methods**
- Sizing studies

# Example: meta-analysis of crossover trials

A primer for those unfamiliar with crossover designs:

- Trial design suitable for patients with chronic, stable conditions who undergo repeated treatment
- Patients are randomised to a *sequence* of treatments
- Describes a very general class of designs but the most common is the 'AB/BA' design: half assigned to A-then-B; half assigned to B-then-A
- Main advantage is balance $\rightarrow$ efficient estimate of treatment effect
- Seminal books are by Jones and Kenward (2003) and Senn (2002)

# Example: meta-analysis of crossover trials



*International Journal of Epidemiology* 2002;**31**:140–149

## Meta-analyses involving cross-over trials: methodological issues

Diana R Elbourne,[a] Douglas G Altman,[b] Julian PT Higgins,[c] Francois Curtin,[a] Helen V Worthington[d] and Andy Vail[e]

# Example: meta-analysis of crossover trials

For today we will consider linear models only.

Authors describe and rank three possible ways to include crossover data in (two-stage) meta-analysis:

1. Include results from paired analysis
2. Include results using data from first period only
3. Include results based on all data but ignoring pairing

Note that (1) is not always possible in meta-analysis when using published results rather than individual-level data

# Example: meta-analysis of crossover trials

1. Paired analysis
2. **Period-1 only**
3. Unpaired analysis of all data

Rationale for (2): '. . . in a randomize cross-over trial the first period is, in effect, a parallel group trial.'

Ok to throw away [up to] half of the data?

# Example: meta-analysis of crossover trials

1. Paired analysis
2. Period-1 only
3. **Unpaired analysis of all data**

Why is (2) $>$ (3), supposedly?
'At best, it [method (3)] is conservative as it ignores the within-patient correlation and so does not make use of the design advantages of a cross-over trial. More importantly, this approach ignores the fact that the same patients appear in both arms of the study and so they are not independent of each other, as required in standard statistical methods.'

# Example: meta-analysis of crossover trials

1. Paired analysis
2. Period-1 only
3. Unpaired analysis of all data

By the authors' own arguments, (3) $>$ (2). I will demonstrate why with a simulation study

For simplicity, I will focus on analysis of a single crossover trial, rather that meta-analysis (results similar either way)

# Does it matter?

PLOS ONE

RESEARCH ARTICLE

## The Use and Reporting of the Cross-Over Study Design in Clinical Trials and Systematic Reviews: A Systematic Assessment

Sarah Jane Nolan[1,2]*, Ian Hambleton[2,3], Kerry Dwan[4]

# Planning with ADMEP

# Planning with ADMEP

Based on the example, I will plan a simulation study using the following structured approach:

- A – Aims
- D – Data-generating mechanisms
- M – Methods
- E – Estimands
- P – Performance measures

# ADMEP: Aims

Before starting, need to work out what we want to learn so we can decide on the best way to learn it

**To determine which of the unpaired analyses (2) and (3) is preferable**

**Aim to investigate whether (3) is conservative (*compared to what?*) and the power/precision of the various methods.**

# ADMEP: Data-generating mechanisms

We're going to consider an AB/BA design and assume a crossover trial is appropriate (main effects of period may exist but no carryover of any sort)

Generate $(Y_1, Y_2) \sim$ BVN for $n = 200$ patients

- Mean is 0 for control arm, $\theta$ for research arm (chosen so that power for method (3)=80% )
- Variance = 1 in both periods
- Correlations between $(Y_1, Y_2)$ of 0 and 0.3

Trivial to do using `drawnorm` and `reshape`

# ADMEP: Data-generating mechanisms

Here, we are not looking for something realistic and have used something simple which is sufficient to make the point

More generally, choosing data-generating mechanisms can be very hard, especially when the mechanism/s impact on how misspecified the methods are.

# ADMEP: Methods to evaluate

1. Paired analysis of crossover trial (comparator/benchmark)

   ```
   . regress y trt period i.id
   ```

2. First period only

   ```
   . regress y trt if period==1
   ```

3. Unpaired analysis of all data

   ```
   . regress y trt period
   ```

# ADMEP: Estimands

(Estimand = the quantity we wish to estimate)

We are interested in estimation of the treatment effect $\theta$

This is the mean of $(Y_A - Y_B)$ and is the estimand of primary interest in crossover trials – the design is predicated on minimising Var($\theta$)

# ADMEP: Estimands

For our example the estimand is obvious. This is not always true.

- Marginal vs. conditional estimands can be subtle
- For prognostic models may need many estimands for the many quantities people are interested – need to cover these
- Methods for modelling nonlinear effects: parameters themselves may not be comparable, for example comparing categorisation vs. splines vs. fractional polynomials

# ADMEP: Performance measures

No issue of **bias** in $\hat{\theta}$ for any analysis

Elbourne *et al*. claim that method (3) is 'conservative'. They mean that $\widehat{\text{Var}}(\hat{\theta})$ is positively biased, leading to confidence intervals that are too wide / over-coverage, so these must be evaluated.
Our performance measures are:

- **Coverage of 95%** confidence intervals
- **Empirical SE** of each method, and relative SE of (2) & (3) vs. (1)
- **Model SE** for each method and relative error
- **Power** of each method

# Choosing the number of repetitions $n^{\text{sim}}$

A very common question. Performance measures will dictate the number of repetitions required: the issue is **Monte Carlo error** (representation of uncertainty due to using finite $n^{\text{sim}}$)

- Could just try something and see if MC error is suitably low, then decide whether more are needed $\rightarrow$ a bit *ad hoc*
- Prefer to start by selecting performance measures of central interest and work out uncertainty we would be prepared to accept (can always increase if needed)

# Choosing the number of repetitions $n^{\text{sim}}$

For example, say key performance measures are coverage and power. Monte Carlo SE is

$$\sqrt{\frac{\pi(1-\pi)}{n^{\text{sim}}}}$$

We expect coverage $\geq 95\%$ and chose $\theta$ s.t. power $\geq 80\%$ for analysis (3). Say we are willing to accept MC error ($\text{SE}^{\text{req}}$) of 0.4%. Then plug into

$$n^{\text{sim}} = \frac{\pi(1-\pi)}{\left(\text{SE}^{\text{req}}\right)^2}$$

Then, for coverage, $n^{\text{sim}} \approx 2,969$
For power, $n^{\text{sim}} = 10,000$

# Coding

# Code for the DGM

```
mat def sd = (1,1)
mat def corr = (1, .3 \.3, 1)
drawnorm y1 y2 , sds(sd) corr(corr) n(200) clear
gen int id = _n
gen byte tperiod = 1 in 1/100
   replace tperiod = 2 in 101/200
reshape long y , i(id) j(period)
gen byte trt = period==tperiod
drop tperiod
replace y = y + ‘trteff’ if trt
```

# Code for the DGM

```
mat def sd = (1,1)
mat def corr = (1, .3 \.3, 1)
drawnorm y1 y2 , sds(sd) corr(corr) n(200) clear
gen int id = _n
gen byte tperiod = 1 in 1/100
   replace tperiod = 2 in 101/200
reshape long y , i(id) j(period)
gen byte trt = period==tperiod
drop tperiod
replace y = y + ‘trteff’ if trt
```

Henceforth this chunk = -dgm-

# Generating, analysing, posting (post)

```
local nsim 10000
local sigma 1 ...
tempname tim
postfile ‘tim’ int(rep) str7(method) float(corr)
> double(theta se) int(df) using estimates, replace
forval r = 1/‘nsim’ {
  foreach c of numlist 0 .3 {
    -dgm-
    -analysis 1-
    post ‘tim’ (‘r’) ("Paired") (‘c’) (_b[trt])
    > (_se[trt]) (e(df_r))
    -analysis 2- ...
  }
}
postclose ‘tim’
```

# Generating, analysing, posting (`simulate`)

The `simulate` command is an alternative to `post`.

You write an rclass program that does one repetition and returns what you would have posted.

It has some serious drawbacks so I avoid it.

# Generating, analysing, posting (`simulate`)

The `simulate` command is an alternative to `post`.

You write an rclass program that does one repetition and returns what you would have posted.

It has some serious drawbacks so I avoid it.

Ok fine, I'll show you then. Here's how we would code our simulation study with `simulate`...

# Generating, analysing, posting (simulate)

```
program define crosssim, rclass
  syntax [ , n(integer 200) sd(real 1) corr(real .3) ]
  return scalar corr = ‘corr’
  -dgm-
  -analysis 1-
  regress y trt period i.id
    return scalar theta1 = _b[trt]
    return scalar se1 = _se[trt]
    return scalar df1 = ‘e(df_r)’
  -analysis 2-
  ...
end
crosssim // test with a single run
return list // check what it returns
```

# Running the reps (simulate)

```
program define crosssim, rclass
  syntax [ , n(integer 200) sd(real 1) corr(real .3) ]
  return scalar corr = ‘corr’
  -dgm-
  regress y trt i.id
    return scalar theta1 = _b[trt]
    return scalar se1 = _se[trt]
    return scalar df1 = ‘e(df_r)’
  -analysis 2-
    return scalar theta2 = _b[trt]
  ...
end
simulate corr=r(corr) theta1=r(theta1) se1=r(se1) df1=r(df1)
> theta2=r(theta2) se2=r(se2) df2=r(df2) ... , reps(10000):
> crosssim
```

# So what's wrong with `simulate`?

1. **Doesn't post a repetition number. I want a unique identifier and sort order as I don't trust it not to change.**
   Could say 'Who cares! They are independent repetitions so the order doesn't matter.' Perhaps.
2. **You can't post strings**
   For what you have just seen, you could argue that it's more efficient to store 'method' as byte (1, 2 and 3) and label the resulting values afterwards. True.

Hold those thoughts...

# Pseudo-randomness, seeds and states

# Coding simulation studies and the RNG

All simulation files will loosely follow:

- Generate data involving some element of random sampling
- Apply some method to data
- Store results of method

# Pseudo, seeds and states

At the core, simulation studies involve using (pseudo-)random sampling from probability distributions. This means it is actually deterministic.

This shouldn't concern us. A coin-toss or die-roll can be viewed as equally deterministic (albeit as a result of unknown factors that act in a completely unpredictable fashion, rendering reproduction of a long sequence difficult).

# Pseudo, seeds and states

The ‘pseudo’ element is sometimes characterised negatively. The main negative arises if the random number generator (RNG) is misused. Can lead to the state ‘jumping in’ on a previous state, resulting in lack of independence across simulated datasets (implications for parallelisation).

Positive:

- Having planted a seed, the state evolves in a completely deterministic manner, with each random number generated moving the position of the state forwards by one
- If there is a failure for repetition 4,870 you can reproduce the exact data and explore to understand better

# Pseudo, seeds and states

Advice:

1. **Set the seed at the beginning, once only**
2. **Store the state as often as necessary**

This avoids ‘jumping in’ through manipulation of the seed but facilitates reproducibility

# Set the seed once

With `post`, set the seed before any random numbers are generated and outside of the main loop.

```
set seed 827271
local nsim 10000
local sigma 1
...
```

# Set the seed once

With `simulate`, it's an option:

```
simulate ... , reps(10000) seed(827271):
> crosssim
```

# Storing the state

In Stata 13, it was called c(seed) and looked like this:

```
X075bcd151f123bb5159a55e5002286570053e56
```

In Stata 14 (Mersenne Twister), it's called `c(rngstate)` and is much longer

```
. di strlen("`c(rngstate)'") > 5011
```

Storing the state requires `strL` format

# Storing the state with post (v14)

When using post, you can post up to str2025. So, to store the state in Stata 14:

```
tempname poststate
postfile ‘poststate’ int(repno) str2000(state1 state2)
> str1100(state3) using statefile.dta
...
post ‘poststate’ (‘r’) (substr(c(rngstate),1,2000))
> (substr(c(rngstate),2001,2000))
> (substr(c(rngstate),4001,.))
...
postclose ‘poststate’
```

Resulting file size for 1,000 reps: >5mb

# Storing the state with post (v13)

Previously, in Stata 13:

```
tempname poststate
postfile ‘poststate’ int(repno) str41(state) using
statefile.dta
...
post ‘poststate’ (‘r’) (c(rngstate))
...
postclose ‘poststate’
```

Resulting file size for 1,000 reps: $\sim$43kb

# Storing the state with `simulate`

When using simulate, you can't post strings, so you have to work out another way to reproduce simulated datasets (there are ways)

# Store the state often

Not just important if you anticipate errors (you should)

Necessary if you want to increase $n^{\text{sim}}$ without jumping in

Useful if you want to add a method (which you may wish to in future), but wish to avoid repeating the entire simulation study and potentially getting slightly different results for original methods

(Note: point only applies if there is a stochastic element in the analysis)

# Analysis of simulation studies

# Analysis of estimates data

Have a dataset of estimates in long or wide format. Here are results for first rep. in long format:

| rep | method | corr | theta | se | df |
|---|---|---|---|---|---|
| 1 | 1 | 0 | .59832 | .09165 | 199 |
| 1 | 2 | 0 | .59832 | .09397 | 398 |
| 1 | 3 | 0 | .62074 | .1328 | 198 |
| 1 | 1 | .3 | .47513 | .08136 | 199 |
| 1 | 2 | .3 | .47513 | .09920 | 398 |
| 1 | 3 | .3 | .53624 | .14550 | 198 |

# Analysis of estimates data

Results for first rep. in wide format:

| rep | corr | theta1 | se1 | df1 | theta2 | se2 | df2 | ... |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | .59832 | .09165 | 199 | .59832 | .09397 | 398 | ... |
| 1 | .3 | .47513 | .08136 | 199 | .47513 | .09920 | 398 | ... |

# Analysis of estimates data

Computing performance measures across DGMs and methods is often done without Monte Carlo error and presented in a table.

Stuff like this…

# Jackson, White & Thompson, *Stat. Med.* (2009)

**Table I.** Properties of estimates $\hat{\mu}_X$ from the simulation study with $n=10$, using 1000 simulations for each run.

| | | | | | $E(\hat{\mu}_X)$ | | $\mathrm{Var}(\hat{\mu}_X)$ | | $Z$ length | | $Z$ coverage | | $t$ coverage | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Run | $I^2_X$ | $I^2_Y$ | $\kappa$ | $\rho_i$ | DL | RE | DL | RE | DL | RE | DL | RE | DL | RE |
| 1 | 0 | 0 | 0 | 0 | 0.002 | 0.001 | 0.007 | 0.007 | 0.370 | 0.355 | 0.961 | 0.960 | 0.987 | 0.977 |
| 2 | 0 | 0.3 | 0 | 0 | −0.001 | −0.001 | 0.008 | 0.008 | 0.372 | 0.357 | 0.960 | 0.952 | 0.980 | 0.977 |
| 3 | 0 | 0.75 | 0 | 0 | 0.000 | 0.000 | 0.008 | 0.008 | 0.366 | 0.356 | 0.962 | 0.965 | 0.988 | 0.983 |
| 4 | 0.3 | 0 | 0 | 0 | −0.005 | −0.004 | 0.012 | 0.011 | 0.418 | 0.408 | 0.936 | 0.928 | 0.966 | 0.960 |
| 5 | 0.3 | 0.3 | 0 | 0 | 0.007 | 0.007 | 0.011 | 0.011 | 0.413 | 0.401 | 0.941 | 0.925 | 0.963 | 0.950 |
| 6 | 0.3 | 0.75 | 0 | 0 | −0.003 | −0.002 | 0.012 | 0.012 | 0.420 | 0.414 | 0.929 | 0.919 | 0.957 | 0.947 |
| 7 | 0.75 | 0 | 0 | 0 | 0.006 | 0.006 | 0.029 | 0.028 | 0.630 | 0.631 | 0.912 | 0.913 | 0.950 | 0.945 |
| 8 | 0.75 | 0.3 | 0 | 0 | −0.010 | −0.010 | 0.029 | 0.029 | 0.631 | 0.630 | 0.895 | 0.892 | 0.930 | 0.924 |
| 9 | 0.75 | 0.75 | 0 | 0 | −0.004 | −0.004 | 0.028 | 0.028 | 0.631 | 0.628 | 0.916 | 0.915 | 0.942 | 0.936 |
| 10 | 0.3 | 0.3 | 0.7 | 0.7 | 0.006 | 0.006 | 0.011 | 0.011 | 0.406 | 0.408 | 0.927 | 0.925 | 0.958 | 0.959 |
| 11 | 0.3 | 0.75 | 0.7 | 0.7 | −0.004 | −0.004 | 0.012 | 0.012 | 0.403 | 0.402 | 0.919 | 0.925 | 0.953 | 0.957 |
| 12 | 0.75 | 0.3 | 0.7 | 0.7 | 0.005 | 0.005 | 0.030 | 0.030 | 0.636 | 0.642 | 0.908 | 0.914 | 0.942 | 0.945 |
| 13 | 0.75 | 0.75 | 0.7 | 0.7 | −0.005 | −0.004 | 0.030 | 0.030 | 0.626 | 0.629 | 0.892 | 0.885 | 0.932 | 0.934 |
| 14 | 0.3 | 0.3 | 0.95 | 0.95 | −0.001 | −0.001 | 0.010 | 0.010 | 0.384 | 0.381 | 0.909 | 0.911 | 0.948 | 0.953 |
| 15 | 0.3 | 0.75 | 0.95 | 0.95 | 0.002 | 0.002 | 0.011 | 0.010 | 0.392 | 0.392 | 0.938 | 0.949 | 0.963 | 0.970 |
| 16 | 0.75 | 0.3 | 0.95 | 0.95 | 0.001 | 0.001 | 0.030 | 0.030 | 0.603 | 0.623 | 0.889 | 0.905 | 0.918 | 0.933 |
| 17 | 0.75 | 0.75 | 0.95 | 0.95 | −0.001 | −0.001 | 0.028 | 0.028 | 0.613 | 0.619 | 0.890 | 0.893 | 0.931 | 0.929 |
| 18 | 0.3 | 0.3 | 0.7 | 0 | 0.003 | 0.003 | 0.012 | 0.012 | 0.416 | 0.410 | 0.922 | 0.915 | 0.956 | 0.948 |
| 19 | 0.3 | 0.75 | 0.7 | 0 | 0.001 | 0.001 | 0.011 | 0.011 | 0.413 | 0.409 | 0.932 | 0.922 | 0.961 | 0.959 |
| 20 | 0.75 | 0.3 | 0.7 | 0 | 0.008 | 0.008 | 0.029 | 0.029 | 0.645 | 0.647 | 0.908 | 0.910 | 0.938 | 0.939 |
| 21 | 0.75 | 0.75 | 0.7 | 0 | −0.001 | −0.001 | 0.028 | 0.028 | 0.616 | 0.622 | 0.903 | 0.906 | 0.938 | 0.935 |
| 22 | 0.3 | 0.3 | 0.95 | 0 | 0.003 | 0.003 | 0.011 | 0.011 | 0.415 | 0.410 | 0.930 | 0.927 | 0.963 | 0.956 |
| 23 | 0.3 | 0.75 | 0.95 | 0 | −0.002 | −0.002 | 0.011 | 0.011 | 0.416 | 0.410 | 0.939 | 0.935 | 0.958 | 0.962 |
| 24 | 0.75 | 0.3 | 0.95 | 0 | 0.004 | 0.004 | 0.031 | 0.031 | 0.638 | 0.640 | 0.901 | 0.902 | 0.935 | 0.938 |
| 25 | 0.75 | 0.75 | 0.95 | 0 | 0.005 | 0.006 | 0.029 | 0.029 | 0.639 | 0.640 | 0.909 | 0.920 | 0.952 | 0.958 |

In each case, DL and RE denote values using the multivariate DerSimonian and Laird and REML procedures, respectively. $E(\hat{\mu}_X)$ denotes the average estimated first treatment effect and $\mathrm{Var}(\hat{\mu}_X)$ denotes the Monte Carlo variance of these estimates. '$Z$ length' is the average length of a nominal 95 per cent confidence interval for the first treatment effect using the standard normal quantile and '$Z$ coverage' and '$t$ coverage' denote the proportion of nominal 95 per cent confidence intervals that cover the true value, using the standard normal and $t$ quantiles, respectively.

# Analysis of estimates data

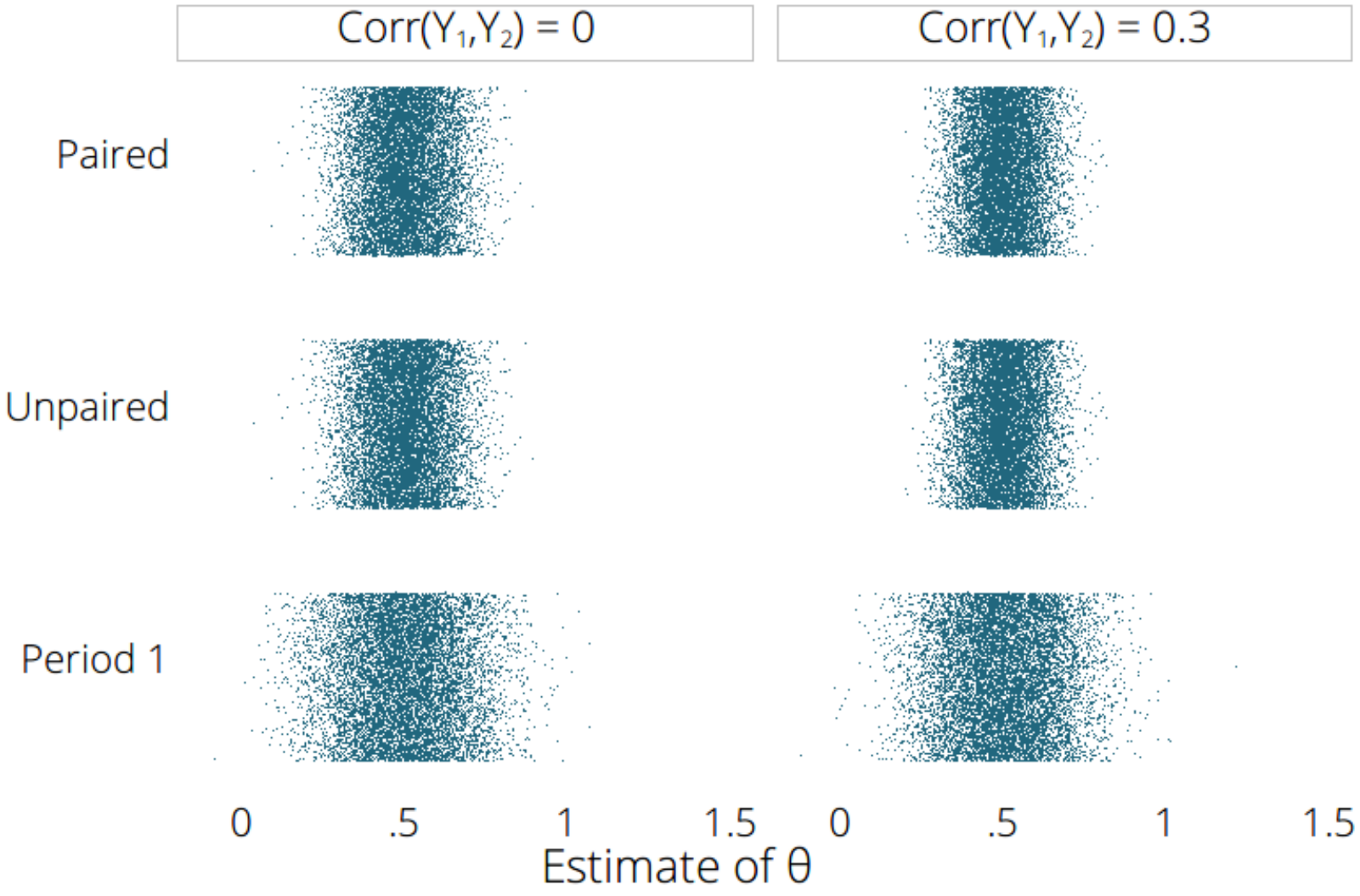
Please, for the sake of the children, avoid this sort of table (except in an appendix)

Instead, suggest:

1. Plots of estimates data
2. Plots of performance measures

We do not claim to have perfect answers, but some ideas follow.

# Plot $\hat{\theta}_i$ by DGM and method

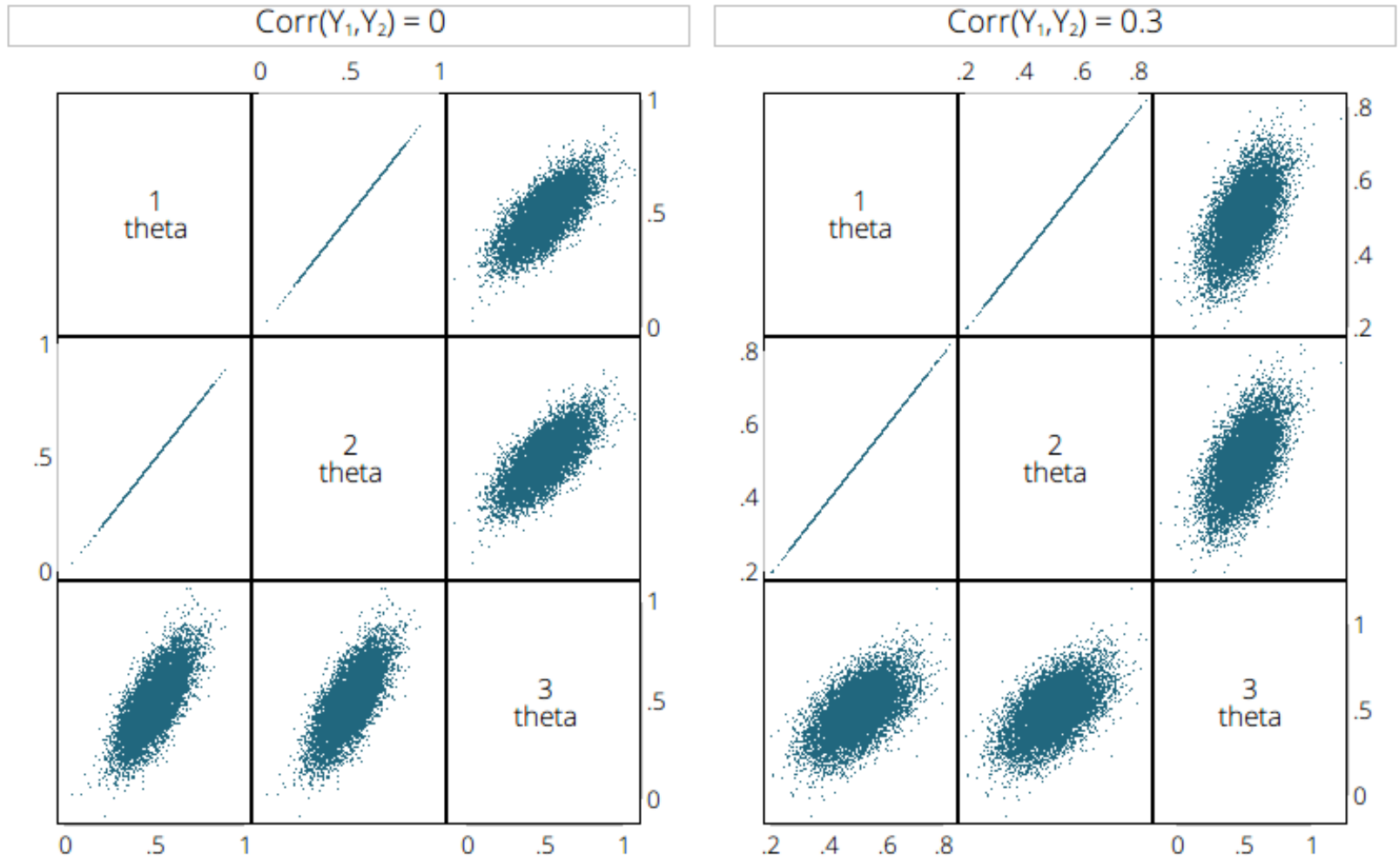


# Plot $\hat{\theta}_i$ by DGM and method

What does this tell us?

1. All methods are unbiased
2. Paired and unpaired analysis of all data closer on average to true $\theta$ than analysis of period 1 data only: they have lower empirical SE
3. Paired and unpaired analysis almost indistinguishable

# Compare $\hat{\theta}_i$ across methods

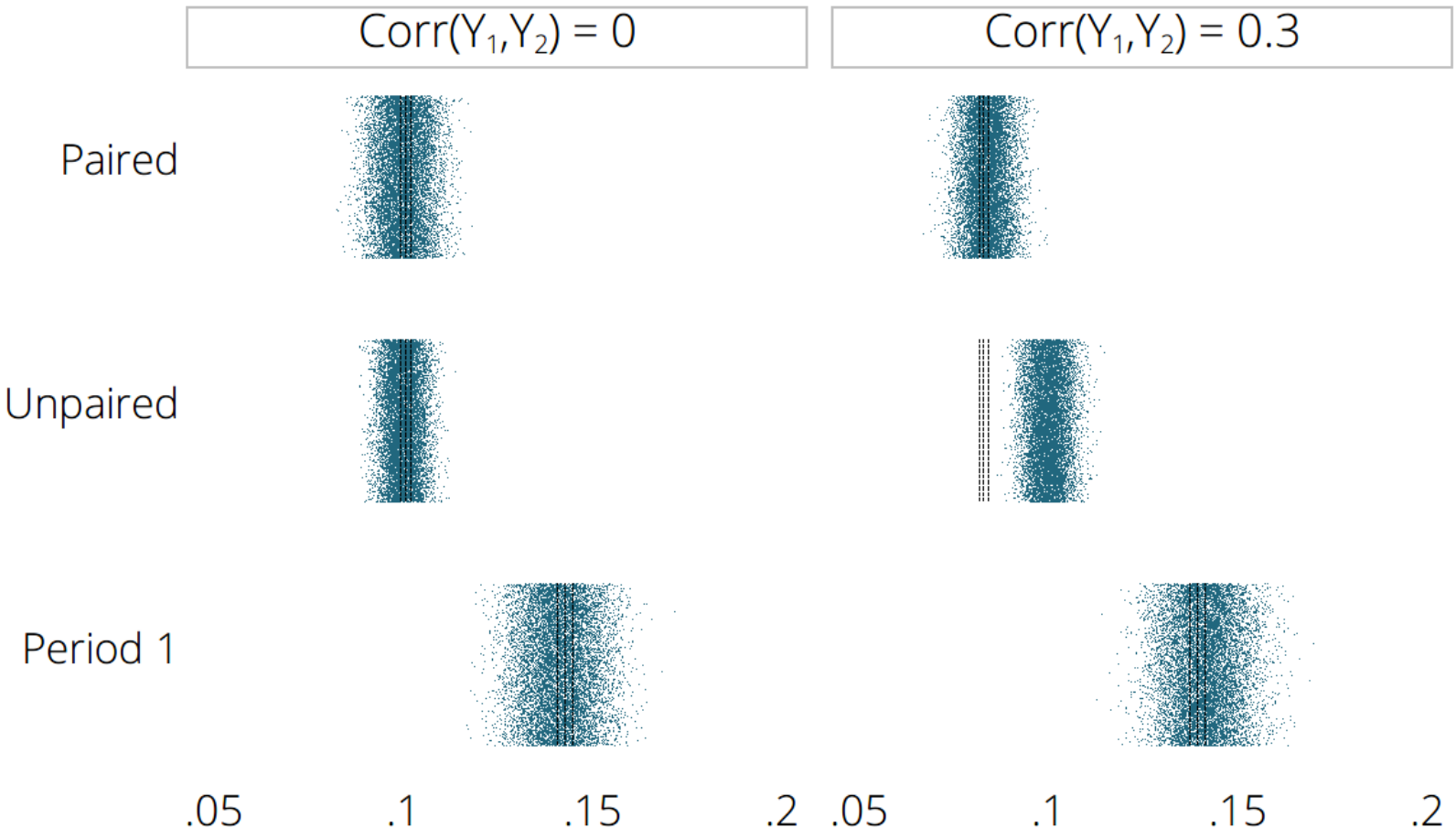


# Plot $\hat{\theta}_i$ by DGM and method

What does this tell us?

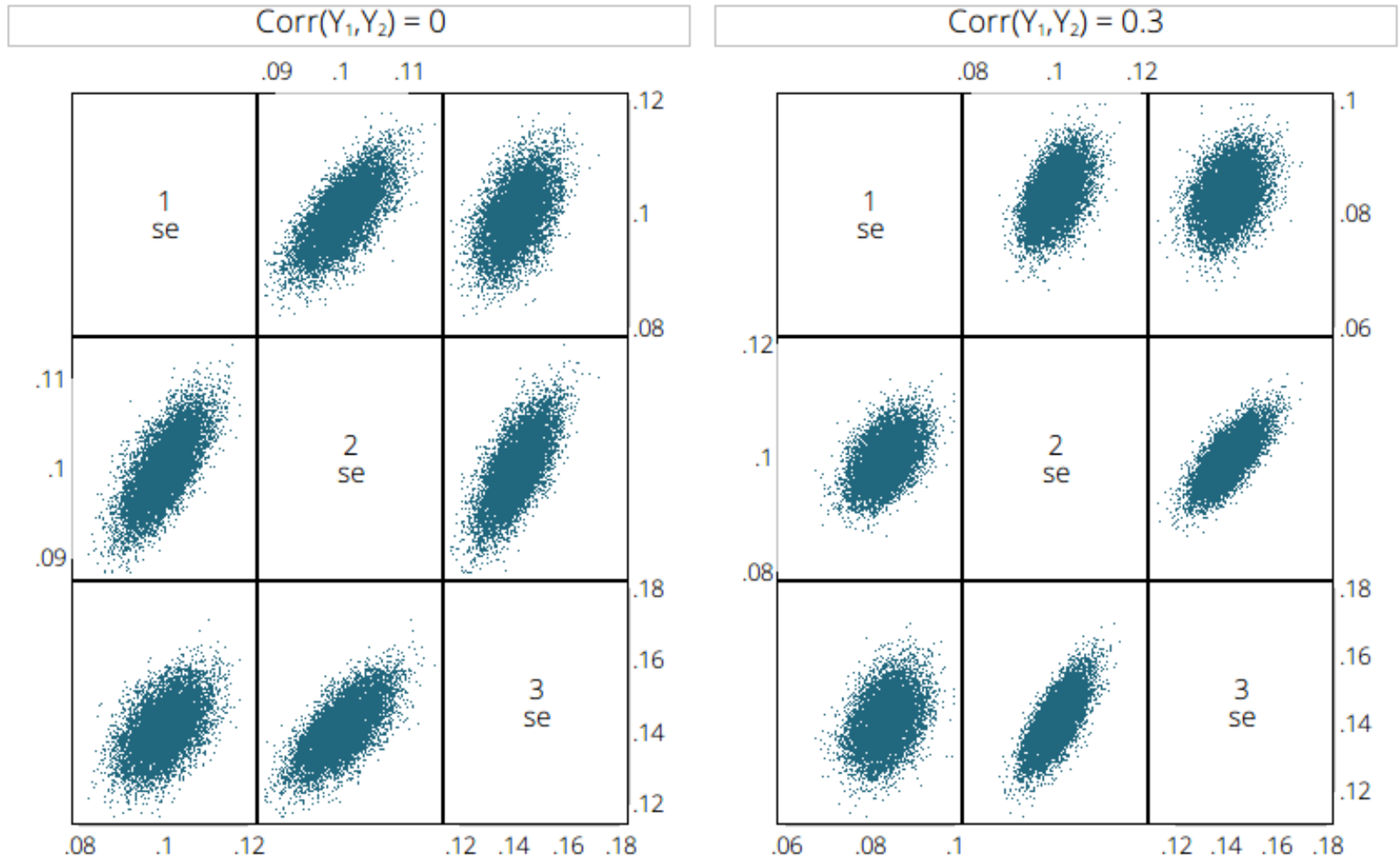
1. Paired and unpaired analysis return *identical* point estimates (this is obvious on hindsight because of full balance)

# Plot $\widehat{\text{SE}}(\hat{\theta}_i)$ by DGM and method

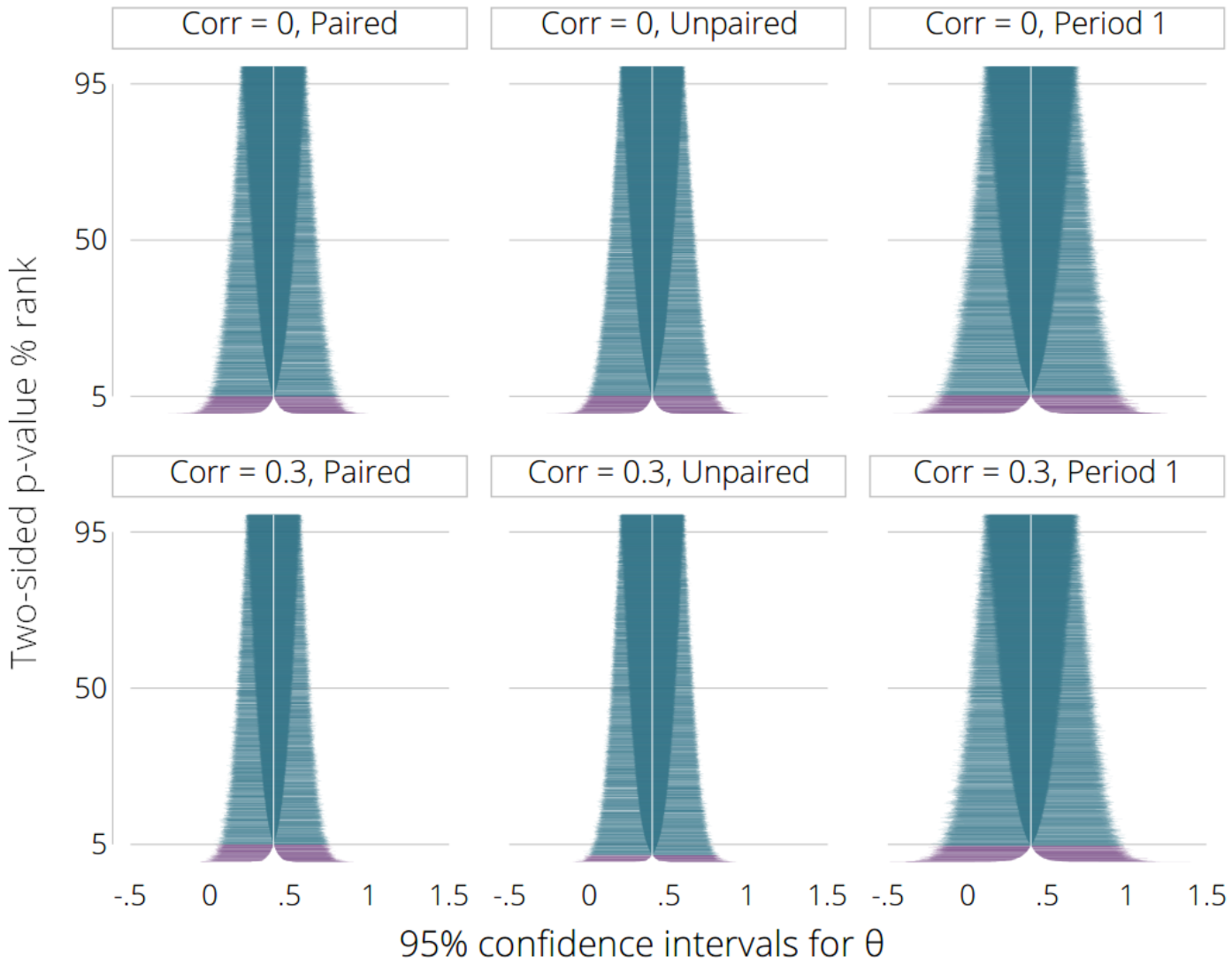



(lines are EmpSE and 95% conf. int.)

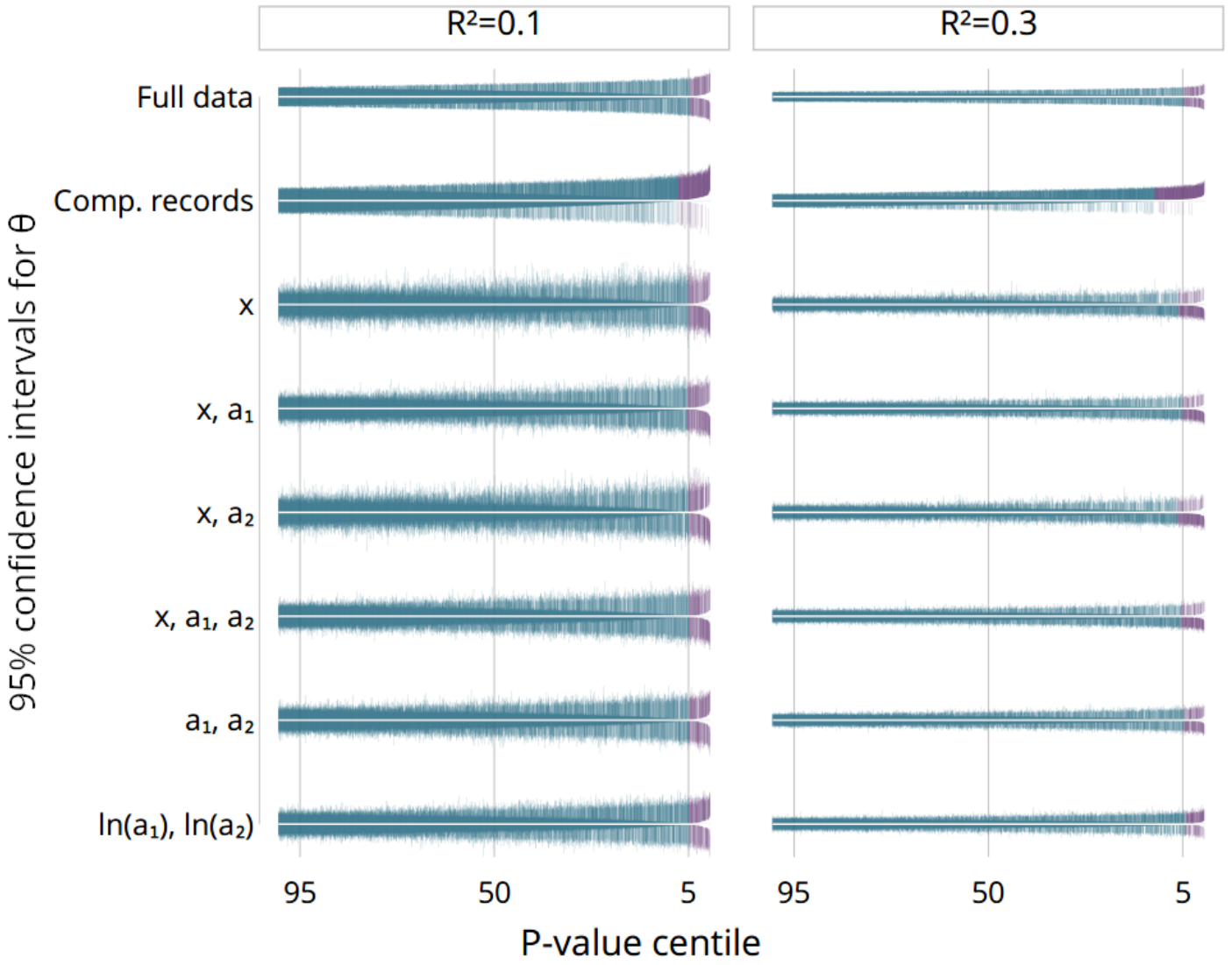
# Compare $\hat{\theta}_i$ across methods



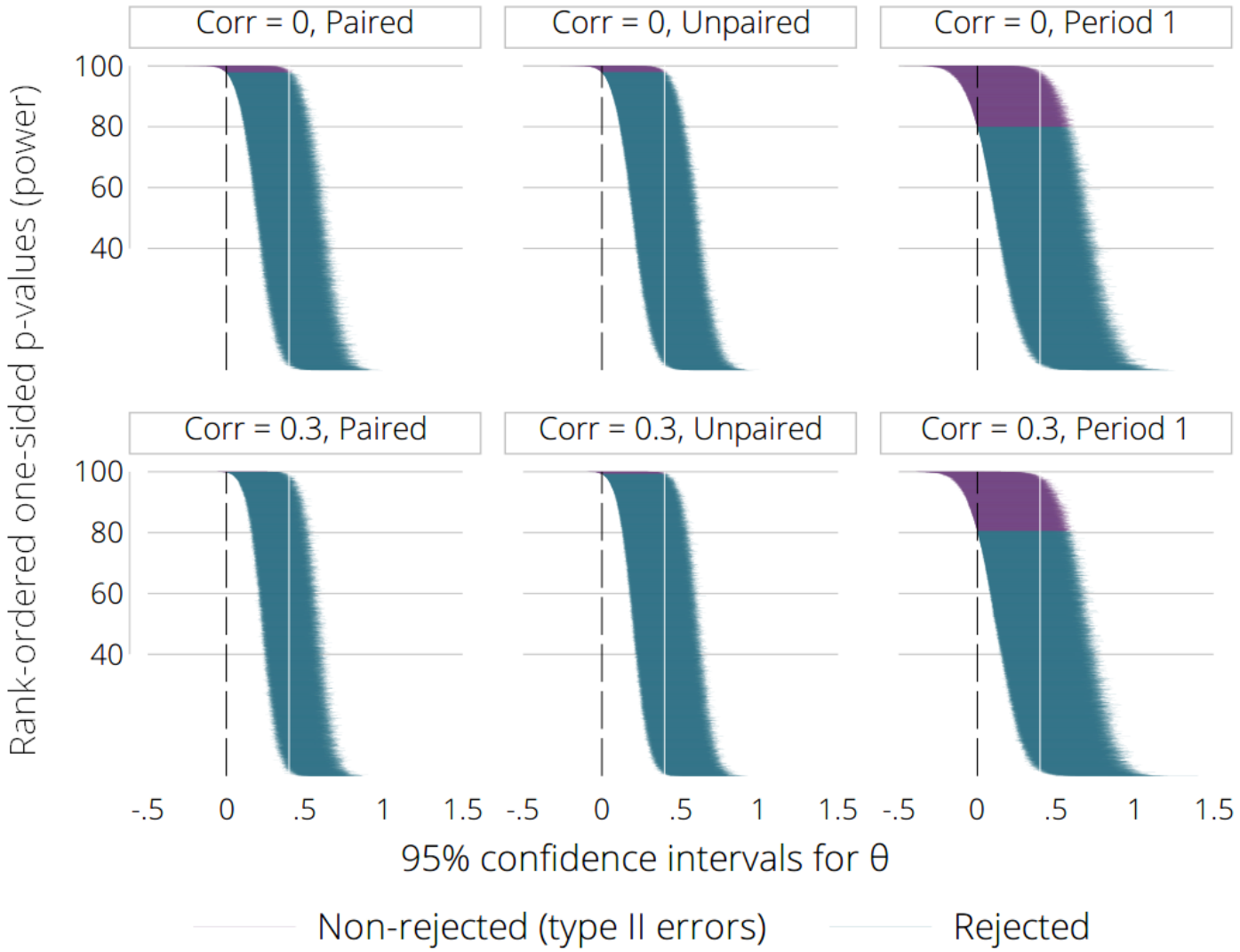
# Plot of CIs: coverage



# More interesting alternative



# Plot of CIs: power



# Analysis of estimates data

For information on estimation of performance measures and Monte Carlo error, see:

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

Koehler E, Brown E, Haneuse. On the assessment of Monte Carlo error in simulation-based statistical analyses. *The American Statistician* 2009; **63**:155–162

# Estimate and tabulate performance measures

| **Performance meas.** relating to $\hat{\theta}$ | **Corr** | **Paired** | | **Unpaired** | | **Period 1** | |
|---|---|---|---|---|---|---|---|
| Coverage % | 0 | 94.9 | 0.22 | 94.8 | 0.22 | 94.6 | 0.23 |
| | 0.3 | 95.1 | 0.22 | **98.0** | 0.14 | 95.3 | 0.21 |

# Estimate and tabulate performance measures

| **Performance meas.** relating to $\hat{\theta}$ | **Corr** | **Paired** | | **Unpaired** | | **Period 1** | |
|---|---|---|---|---|---|---|---|
| Coverage % | 0 | 94.9 | 0.22 | 94.8 | 0.22 | 94.6 | 0.23 |
| | 0.3 | 95.1 | 0.22 | **98.0** | 0.14 | 95.3 | 0.21 |
| Empir. SE ($\times$100) | 0 | 10 | 0.07 | 10 | 0.07 | **14.2** | 0.1 |
| | 0.3 | 8.2 | 0.05 | 8.2 | 0.05 | **13.9** | 0.09 |

# Estimate and tabulate performance measures

| **Performance meas.** relating to $\hat{\theta}$ | **Corr** | **Paired** | | **Unpaired** | | **Period 1** | |
|---|---|---|---|---|---|---|---|
| Coverage % | 0 | 94.9 | 0.22 | 94.8 | 0.22 | 94.6 | 0.23 |
| | 0.3 | 95.1 | 0.22 | **98.0** | 0.14 | 95.3 | 0.21 |
| Empir. SE ($\times$100) | 0 | 10 | 0.07 | 10 | 0.07 | **14.2** | 0.1 |
| | 0.3 | 8.2 | 0.05 | 8.2 | 0.05 | **13.9** | 0.09 |
| % gain in precision | 0 | 0 | . | 0 | . | **−50.4** | 0.69 |
| (vs. paired) | 0.3 | 0 | . | 0 | . | **−64.7** | 0.58 |

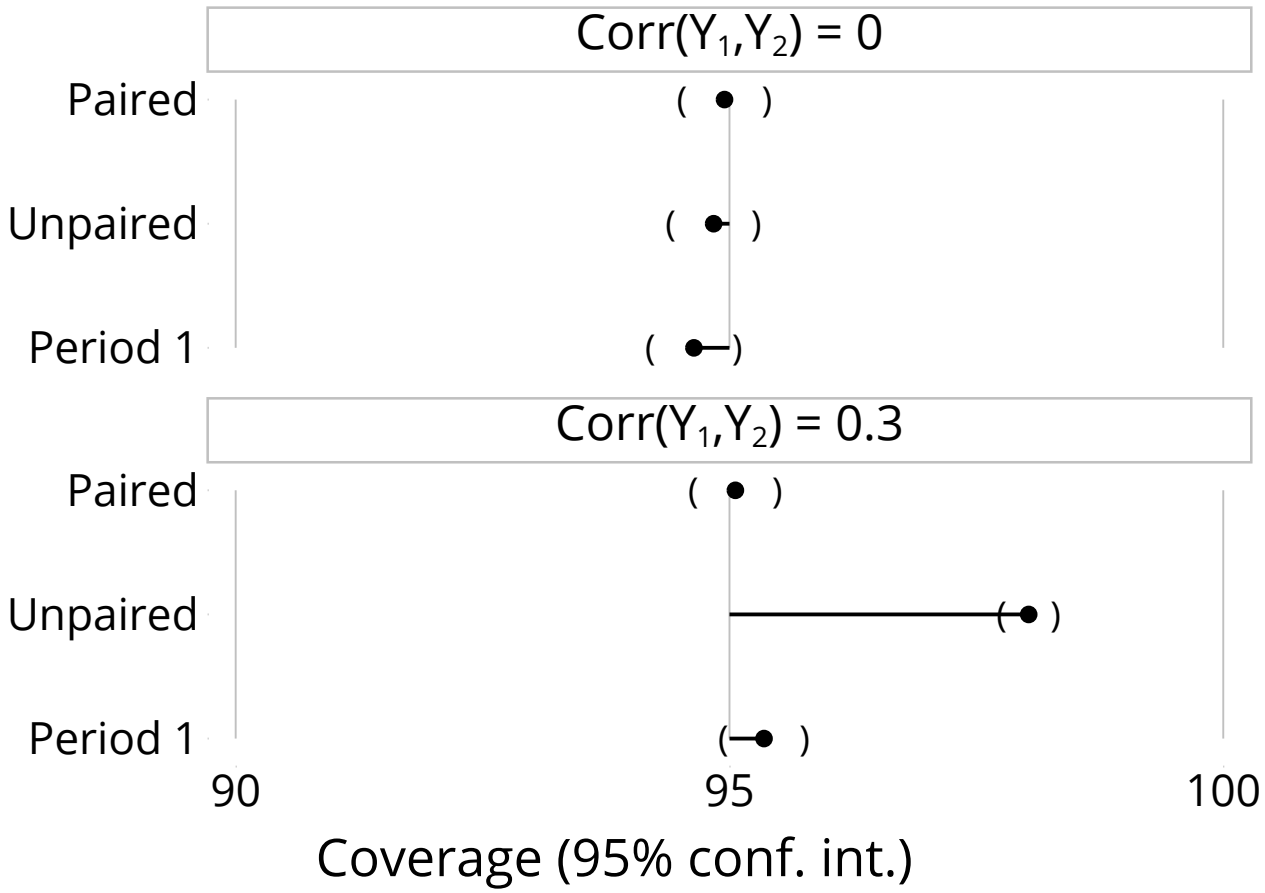
# Estimate and tabulate performance measures

| **Performance meas.** relating to $\hat{\theta}$ | **Corr** | **Paired** | | **Unpaired** | | **Period 1** | |
|---|---|---|---|---|---|---|---|
| Coverage % | 0 | 94.9 | 0.22 | 94.8 | 0.22 | 94.6 | 0.23 |
| | 0.3 | 95.1 | 0.22 | **98.0** | 0.14 | 95.3 | 0.21 |
| Empir. SE ($\times$100) | 0 | 10 | 0.07 | 10 | 0.07 | **14.2** | 0.1 |
| | 0.3 | 8.2 | 0.05 | 8.2 | 0.05 | **13.9** | 0.09 |
| % gain in precision | 0 | 0 | . | 0 | . | **−50.4** | 0.69 |
| (vs. paired) | 0.3 | 0 | . | 0 | . | **−64.7** | 0.58 |
| Model SE ($\times$100) | 0 | 10.0 | 0.005 | 10.0 | 0.004 | 14.1 | 0.007 |
| | 0.3 | 8.4 | 0.004 | 10.0 | 0.004 | 14.1 | 0.007 |

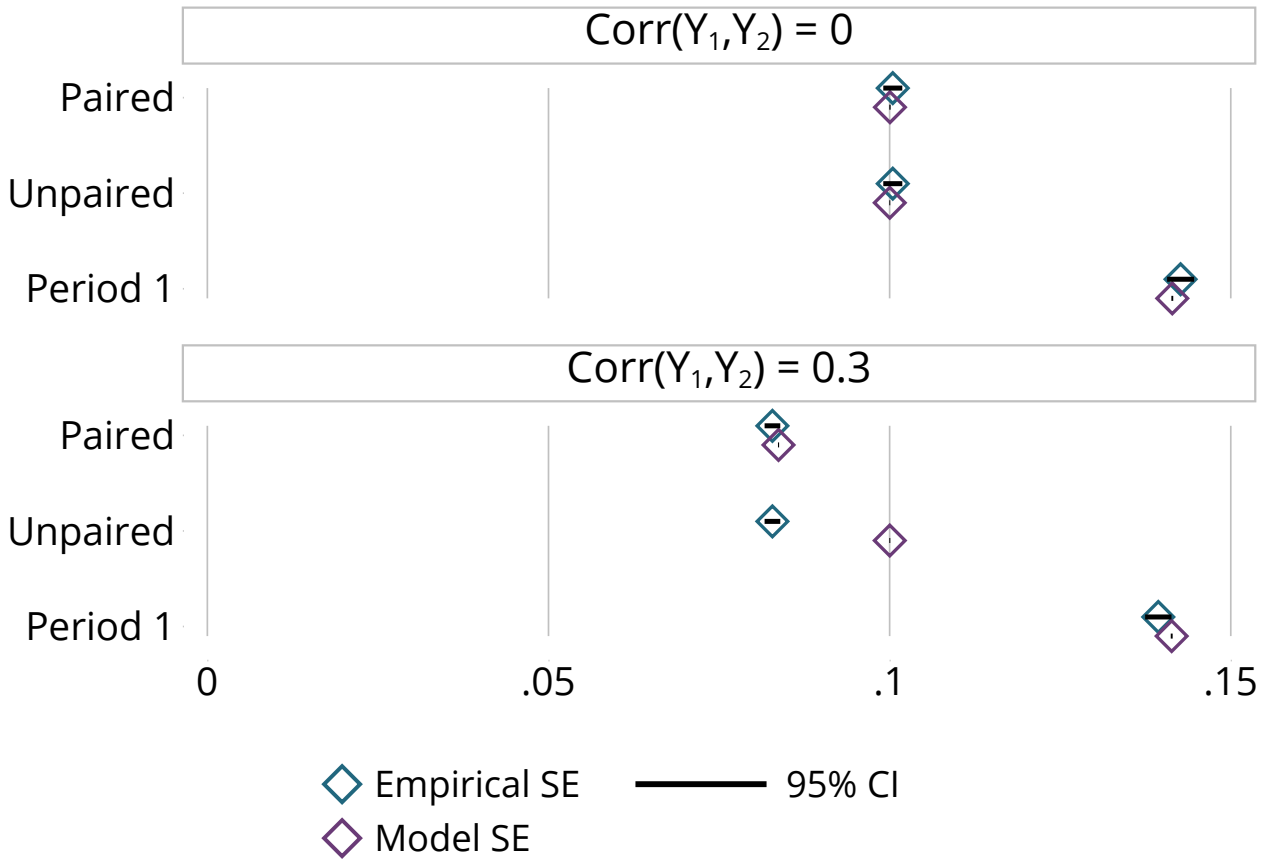
# Estimate and tabulate performance measures

| **Performance meas.** relating to $\hat{\theta}$ | **Corr** | **Paired** | | **Unpaired** | | **Period 1** | |
|---|---|---|---|---|---|---|---|
| Coverage % | 0 | 94.9 | 0.22 | 94.8 | 0.22 | 94.6 | 0.23 |
| | 0.3 | 95.1 | 0.22 | **98.0** | 0.14 | 95.3 | 0.21 |
| Empir. SE ($\times$100) | 0 | 10 | 0.07 | 10 | 0.07 | **14.2** | 0.1 |
| | 0.3 | 8.2 | 0.05 | 8.2 | 0.05 | **13.9** | 0.09 |
| % gain in precision | 0 | 0 | . | 0 | . | **−50.4** | 0.69 |
| (vs. paired) | 0.3 | 0 | . | 0 | . | **−64.7** | 0.58 |
| Model SE ($\times$100) | 0 | 10.0 | 0.005 | 10.0 | 0.004 | 14.1 | 0.007 |
| | 0.3 | 8.4 | 0.004 | 10.0 | 0.004 | 14.1 | 0.007 |
| Power % | 0 | 97.8 | 0.1 | 97.8 | 0.1 | **79.8** | 0.4 |
| | 0.3 | 99.8 | 0.04 | 99.3 | 0.1 | **80.4** | 0.4 |

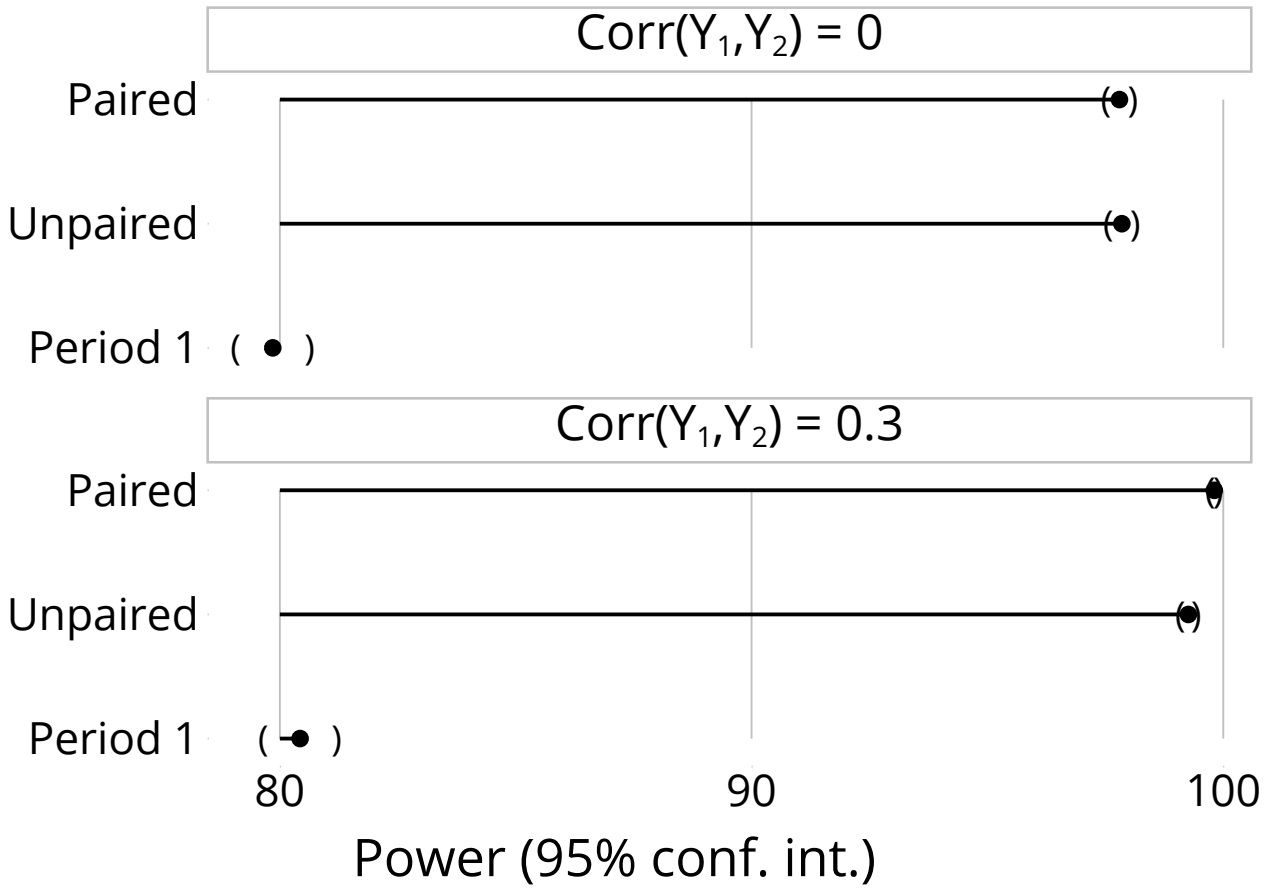
# Estimate and plot performance measures



# Estimate and plot performance measures



# Estimate and plot performance measures



# Sample-conditional performance

A cautionary principle: do not estimate performance measures conditional on sample statistics.

# Sample-conditional performance

A cautionary principle: do not estimate performance measures conditional on sample statistics. An example:

1. Draw samples from $N(0, 1)$
2. In each, compute mean and $t$-based CI
3. Estimate coverage overall and in tertile-based groups of SE

| | $n^{\text{sim}}$ | Coverage | SE |
|---|---|---|---|
| Overall | 30,000 | 95.0 | 0.1 |

# Sample-conditional performance

A cautionary principle: do not estimate performance measures conditional on sample statistics. An example:

1. Draw samples from $N(0, 1)$
2. In each, compute mean and $t$-based CI
3. Estimate coverage overall and in tertile-based groups of SE

| | $n^{\text{sim}}$ | Coverage | SE |
|---|---|---|---|
| Overall | 30,000 | 95.0 | 0.1 |
| Lowest third of SEs | 10,000 | 91.5 | 0.2 |
| Middle third of SEs | 10,000 | 95.5 | 0.2 |
| Highest third of SEs | 10,000 | 98.0 | 0.1 |

# Preparing to fail?

For some simulation studies it is inevitable that some repetitions will result in failure – even after efforts to robustify (user-written) commands.

Implications:

- Need to ensure results get stored as failures (*i.e.* avoid accidentally posting leftovers from previous reps)
- Need an approach to analysis (could regard results for a method as 'encouraging conditional on a method converging' – but see previous slide)
- May need to consider incomplete-data methods

# Conclusions

# Meta-analysis of crossover trials

1. Paired analysis
2. **Period-1 only**
3. Unpaired analysis of all data

Why is (2) supposedly superior to (3)?
'At best, it [method (3)] is conservative as it ignores the within-patient correlation and so does not make use of the design advantages of a cross-over trial. More importantly, this approach ignores the fact that the same patients appear in both arms of the study and so they are not independent of each other, as required in standard statistical methods.'

# Meta-analysis of crossover trials

1. Paired analysis
2. Period-1 only
3. **Unpaired analysis of all data**

Why is (3) actually superior to (2)?
At best, it [method (3)] is as good as method (1) – despite the fact that it ignores the within-patient correlation and does not make use of the design advantages of a cross-over trial. More importantly, compared with method (2), this approach is at less risk of making either a type I or a type II error, and so should be preferred.

# Meta-analysis of crossover trials

I have a strong view on this, but one could argue that in fact method (2) > method (3) if:

1. You care that coverage must be as advertised and don't want extra for free ('I don't take charity!')
2. You have an obsession with model SE $\approx$ empirical SE ('randomisation validity')

# Simulation recommendations

- Plan simulation studies on paper before coding
- Structure is key and ADMEP is a useful general approach (you may go back and forth. . . lots), which is a ready-written methods section
- Use theoretical knowledge of the problem (e.g. we did not need to evaluate bias)
- Build code up slowly
- In general, use `postfile` and not `simulate` (until it's fixed!)
- Report Monte Carlo SEs (`simsum`)
- Make the elements of ADMEP coherent in tabulations or plots of performance measures
- Do all you can to make results reproducible

# References relating to example

Jones B, Kenward MG. *Design and analysis of cross-over trials*. Chapman & Hall/CRC: Florida, 2003

Senn S. *Cross-over trials in clinical research*. Wiley: Chichester, 2002

Elbourne DR, Altman DG, Higgins JPT, Curtin F, Worthington HV, Vail A. Meta-analyses involving cross-over trials: methodological issues. *International Journal of Epidemiology* 2002; **31**:140–149

Nolan S, Hambleton I, Dwan K. The use and reporting of the cross-over study design in clinical trials and systematic reviews: a systematic assessment. *PLoS ONE* 2016; **11**(7): e0159014

# References relating to simulation

Burton A, Altman DG, Royston P, Holder RL. The design of simulation studies in medical statistics. *Statistics in Medicine*, 2006; **25**:4279–4292.

White IR. simsum: Analyses of simulation studies including monte carlo error. *Stata Journal*, 2010; **10**(3):369–385.

Koehler E, Brown E, Haneuse. On the assessment of Monte Carlo error in simulation-based statistical analyses. *The American Statistician* 2009; **63**:155–162.

# Using simulation studies to evaluate statistical methods in Stata: A tutorial

**Tim Morris**, MRC Clinical Trials Unit at UCL
**Ian White**, MRC Biostatistics Unit
**Michael Crowther**, University of Leicester