A More Versatile Sample Size Calculator

Richard Hooper

Senior Lecturer in Medical Statistics

Barts and The London
School of Medicine and Dentistry

Queen Mary
University of London

Why worry about sample size?

Why worry about sample size?

1336 BRITISH MEDICAL JOURNAL VOLUME 281 15 NOVEMBER 1980

Medicine and Mathematics

Statistics and ethics in medical research

III How large a sample?

DOUGLAS G ALTMAN

Whatever type of statistical design is used for a study, the problem of sample size must be faced. This aspect, which causes considerable difficulty for researchers, is perhaps the most common reason for consulting a statistician. There are also, however, many who give little thought to sample size, choosing the most convenient number (20, 50, 100, etc) or time period (one month, one year, etc) for their study. They, and those who approve such studies, should realise that there are important statistical and ethical implications in the choice of sample size for a study.

Why worry about sample size?

"The number of subjects in a clinical trial should always be large enough to provide a reliable answer to the questions addressed. This number is usually determined by the primary objective of the trial."

ICH Guidelines for Clinical Trials

Why worry about sample size?

"The number of subjects in a clinical trial should always be large enough to provide a reliable answer to the questions addressed. This number is usually determined by the primary objective of the trial."

ICH Guidelines for Clinical Trials

"For scientific and ethical reasons, the sample size for a trial needs to be planned carefully, with a balance between medical and statistical considerations."

CONSORT statement on reporting clinical trials

Why worry about sample size?

"The number of subjects in a clinical trial should always be large enough to provide a reliable answer to the questions addressed. This number is usually determined by the primary objective of the trial."

ICH Guidelines for Clinical Trials

"For scientific and ethical reasons, the sample size for a trial needs to be planned carefully, with a balance between medical and statistical considerations."

CONSORT statement on reporting clinical trials

"This [sample size calculation] is frequently one of the least credible components of a trial [funding] application."

NIHR/MRC Efficacy & Mechanisms Evaluation funding programme

Statistical power

Power is the probability that a research study will find evidence for an effect.

Statistical power

Power is the probability that a research study will find evidence for an effect.

Power depends on:

- the effect hypothesised
- what counts as evidence
- how the study is designed

Statistical power

Power is the probability that a research study will find evidence for an effect.

Power depends on:

- the effect hypothesised
- what counts as evidence
- how the study is designed

But given these things, power depends on size: the bigger the study, the greater the power.

Statistical power

Power is the probability that a research study will find evidence for an effect.

Power depends on:

- the effect hypothesised
- what counts as evidence
- how the study is designed

But given these things, power depends on size: the bigger the study, the greater the power.

A study should have at least 80% power at the 5% significance level to detect a clinically important effect.

A simple example

Two arm clinical trial:

- single measurement of systolic blood pressure in people given an experimental drug, compared with people given a placebo
- assume blood pressure is normally distributed, with s.d. 20mmHg in each group
- suppose we want to detect a mean difference of 4mmHg between treated and placebo groups

A simple example

Two arm clinical trial:

- single measurement of systolic blood pressure in people given an experimental drug, compared with people given a placebo
- assume blood pressure is normally distributed, with s.d. 20mmHg in each group
- suppose we want to detect a mean difference of 4mmHg between treated and placebo groups

We can use the **`sampsi`** command

```
. sampsi 0 4, sd(20) power(0.8) alpha(0.05)

Estimated sample size for two-sample comparison of
means

Test Ho: m1 = m2, where m1 is the mean in population 1
                    and m2 is the mean in population 2
Assumptions:

         alpha =   0.0500  (two-sided)
         power =   0.8000
            m1 =        0
            m2 =        4
           sd1 =       20
           sd2 =       20
         n2/n1 =     1.00

Estimated required sample sizes:

            n1 =      393
            n2 =      393
```

A more versatile sample size calculator

What if there was a sample size calculator that could work out the required sample size for *any* statistical method under *any* statistical model that we can program?

```
. program define s_mcnemar, rclass
      syntax , OR(real) DISC(real) NPAIRS(integer)
      drop _all
      set obs `npairs’
      scalar p01=`disc'*`or'/(1+`or')
      gen r=runiform()
      gen y1=(r<p01)
      gen y2=((r<`disc')&(r>p01))
      capture noisily mcc y1 y2
      return scalar p_exact=r(p_exact)
      return scalar p_chi2=2*(1-normal(sqrt(r(chi2))))
   end
```

```
. program define s_mcnemar, rclass
      syntax , OR(real) DISC(real) NPAIRS(integer)
      drop _all
      set obs `npairs’
      scalar p01=`disc'*`or'/(1+`or')
      gen r=runiform()
      gen y1=(r<p01)
      gen y2=((r<`disc')&(r>p01))
      capture noisily mcc y1 y2
      return scalar p_exact=r(p_exact)
      return scalar p_chi2=2*(1-normal(sqrt(r(chi2))))
  end
```

		y1	
		0	1
y2	0	1 - disc	p01
	1	disc - p01	0

```
. program define s_mcnemar, rclass
    syntax , OR(real) DISC(real) NPAIRS(integer)
    drop _all
    set obs `npairs’
    scalar p01=`disc'*`or'/(1+`or')
    gen r=runiform()
    gen y1=(r<p01)
    gen y2=((r<`disc')&(r>p01))
    capture noisily mcc y1 y2
    return scalar p_exact=r(p_exact)
    return scalar p_chi2=2*(1-normal(sqrt(r(chi2))))
end
```

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4)) inc(10) prec(0.005)
> pvalue(p_exact) notable

        npairs = 190
        achieves 80.57% power (99% CI 80.07, 81.06)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.0e-03
```

How does **`simsam`** work?

`simsam` uses simulation to estimate power at a number of different sample sizes to find the smallest sample size that achieves the required power

See Feiveson (2009)
"How can I use Stata to calculate power by simulation?"
www.stata.com/support/faqs/statistics/power-by-simulation/

– but note that **`simsam`** uses a faster, more efficient, and more fully automated search than is described in this FAQ

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4)) inc(10) prec(0.005)
> pvalue(p_exact) notable

        npairs = 190
        achieves 80.57% power (99% CI 80.07, 81.06)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.0e-03
```

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4)) inc(10) prec(0.005)
> pvalue(p_exact)

------------------------------------------------------
iteration   npairs              power (99% CI)
------------------------------------------------------
        1      100 ........... 0.6100 (0.4765, 0.7327)
        2      160 ........... 0.7400 (0.7027, 0.7750)
        3      190 ........... 0.8098 (0.7995, 0.8198)
        4      190 ........... 0.8057 (0.8007, 0.8106)
        5      180 ........... 0.7845 (0.7793, 0.7896)
------------------------------------------------------

        npairs = 190
        achieves 80.57% power (99% CI 80.07, 81.06)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.0e-03
```

Continuing **`simsam`**

`simsam` stops if

- it has converged on a solution
- it has completed a specified number of iterations (default 10)
- the sample size cannot be reliably determined to within one increment
- the estimated power is unnaturally low
- increasing the sample size doesn't seem to be controlling the power

Continuing **`simsam`**

`simsam` stops if

- it has converged on a solution
- it has completed a specified number of iterations (default 10)
- the sample size cannot be reliably determined to within one increment
- the estimated power is unnaturally low
- increasing the sample size doesn't seem to be controlling the power

In each case, you can attempt to continue using the command **`simsam continue`** *e.g.* continuing after a fixed number of iterations

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4)) inc(10) prec(0.005)
> pvalue(p_exact) iter(2)

-----------------------------------------------------
iteration   npairs              power (99% CI)
-----------------------------------------------------
        1      100 ........... 0.4900 (0.3594, 0.6216)
        2      210 ........... 0.8570 (0.8263, 0.8843)
-----------------------------------------------------

Warning: did not converge within 2 iterations
```

```
. simsam continue

-----------------------------------------------------
iteration   npairs              power (99% CI)
-----------------------------------------------------
        1      180 ........... 0.7872 (0.7765, 0.7977)
        2      190 ........... 0.8107 (0.8058, 0.8156)
        3      180 ........... 0.7890 (0.7838, 0.7941)
-----------------------------------------------------

        npairs = 190
        achieves 81.07% power (99% CI 80.58, 81.56)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.0e-03
```

e.g. continuing to obtain a higher-precision solution

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4)) inc(10) prec(0.005)
> pvalue(p_exact)

------------------------------------------------------
iteration   npairs              power (99% CI)
------------------------------------------------------
        1      100 ........... 0.4000 (0.2763, 0.5335)
        2      270 ........... 0.9180 (0.8931, 0.9388)
        3      190 ........... 0.8111 (0.8008, 0.8211)
        4      190 ........... 0.8118 (0.8068, 0.8166)
        5      180 ........... 0.7907 (0.7856, 0.7958)
------------------------------------------------------

        npairs = 190
        achieves 81.18% power (99% CI 80.68, 81.66)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.0e-03
```

```
. simsam continue, inc(1) prec(0.0005)

------------------------------------------------------
iteration   npairs              power (99% CI)
------------------------------------------------------
        1      190 ........... 0.8092 (0.8082, 0.8102)
        2      186 ........... 0.8004 (0.7999, 0.8009)
        3      185 ........... 0.7980 (0.7975, 0.7985)
------------------------------------------------------

        npairs = 186
        achieves 80.04% power (99% CI 79.99, 80.09)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.1e-03
```

e.g. correcting the precision or increment to ensure convergence

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4)) inc(1) prec(0.005)
> pvalue(p_exact)

------------------------------------------------------
iteration   npairs              power (99% CI)
------------------------------------------------------
        1      100 ........... 0.5500 (0.4170, 0.6781)
------------------------------------------------------

Warning: npairs not reliably determined to within one
increment

If continuing, use prec/inc < 1.1e-03
```

```
. simsam continue, inc(10) prec(0.005)

---------------------------------------------------------
iteration   npairs              power (99% CI)
---------------------------------------------------------
        1      190 ........... 0.8010 (0.7666, 0.8325)
        2      190 ........... 0.8015 (0.7910, 0.8117)
        3      190 ........... 0.8061 (0.8012, 0.8111)
        4      180 ........... 0.7902 (0.7850, 0.7952)
---------------------------------------------------------

        npairs = 190
        achieves 80.61% power (99% CI 80.12, 81.11)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

If continuing, use prec/inc < 1.0e-03
```

Estimating the “power” under the null

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) null(or(1)) assuming(disc(0.4))
> inc(10) prec(0.005) pvalue(p_exact) notable

        npairs = 190
        achieves 81.13% power (99% CI 80.64, 81.62)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

     under null:  3.74% power (99% CI  3.32,  4.19)

If continuing, use prec/inc < 1.0e-03
```

Using a different returned P-value

```
. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) null(or(1)) assuming(disc(0.4))
> inc(10) prec(0.005) pvalue(p_chi2) notable

        npairs = 180
        achieves 82.13% power (99% CI 81.64, 82.60)
          at the 5% significance level
    to detect
            or = 2
     assuming
          disc = 0.4

     under null:  4.99% power (99% CI  4.51,  5.51)

If continuing, use prec/inc < 1.1e-03
```

Returning a non-significant indicator instead of a P-value

```
. program define s_mcnemar, rclass
      syntax , OR(real) DISC(real) NPAIRS(integer) A(real)
      drop _all
      set obs `npairs’
      scalar p01=`disc'*`or'/(1+`or')
      gen r=runiform()
      gen y1=(r<p01)
      gen y2=((r<`disc')&(r>p01))
      capture noisily mcc y1 y2
      return scalar nonsig=(r(p_exact)>`a')
   end
```

```
. program define s_mcnemar, rclass
      syntax , OR(real) DISC(real) NPAIRS(integer) A(real)
      drop _all
      set obs `npairs’
      scalar p01=`disc'*`or'/(1+`or')
      gen r=runiform()
      gen y1=(r<p01)
      gen y2=((r<`disc')&(r>p01))
      capture noisily mcc y1 y2
      return scalar nonsig=(r(p_exact)>`a')
   end


. simsam s_mcnemar npairs, power(0.8) alpha(0.05)
> detect(or(2)) assuming(disc(0.4) a(0.05))
> inc(10) prec(0.005) pvalue(nonsig)
```

Better example: group sequential methods

e.g. 2-stage O'Brien-Fleming procedure:

After stage 1, assume there is a standard normal test statistic Z_1:

if $\lvert Z_1 \rvert \geq 2.795$	stop, reject H_0;
otherwise	continue to Stage 2.

After stage 2, assume there is a standard normal test statistic Z_2:

if $\lvert Z_2 \rvert \geq 1.977$	stop, reject H_0;
otherwise	stop, accept H_0.

Then the overall significance level is 5%

```
. simsam s_groupseq2 npergrpperstage, power(0.8)
> alpha(0.05) detect(d(4)) null(d(0))
> assuming(sd(20) crit1(2.795) crit2(1.977))
> inc(10) prec(0.005) pvalue(nonsig) notable

  npergrpper~e = 200
        achieves 80.62% power (99% CI 80.12, 81.11)
          at the 5% significance level
    to detect
             d = 4
     assuming
            sd = 20
         crit1 = 2.795
         crit2 = 1.977

     under null:  4.95% power (99% CI  4.47,  5.47)

If continuing, use prec/inc < 9.8e-04
```

```
. simulate npergrp=r(npergrp), reps(10000) :
> s_groupseq2, d(4) sd(20) npergrpperstage(200)
> crit1(2.795) crit2(1.977)
```

[output omitted]

```
. summ npergrp

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     npergrp |     10000      356.98    82.18245        200        400
```

Closing remarks

- **`simsam`** is an extremely versatile sample size calculator
- It is remarkably robust, finding the required sample size whenever it can, and giving up when it has no hope
- It gives an answer that is repeatable to within the specified sample size increment

Hooper R. Versatile sample size calculation using simulation. *Stata Journal* (in press)

r.l.hooper@qmul.ac.uk

Thank you