Simulation-based power analysis for linear and generalized linear models

Joerg Luedicke

Yale University & University of Florida

Stata Conference, New Orleans, LA – July 18-19, 2013

Outline

Significance testing and statistical power

- Point null hypothesis significance testing
- Type I & Type II error
  - Type I: Reject H_0 when it is true
  - Type II: Failure to reject H_0 when it is false
  - Type I & Type II trade-off
- Statistical power
  - $\beta \Rightarrow$ probability of not rejecting H_0 when it is false
  - Power $\Rightarrow 1 - \beta$
  - i.e., the probability of rejecting H_0, given it is indeed false
- Importance of power analysis
  - Study planning
  - Reasonable resource allocation
  - Saving time and money

Analytical vs. simulation-based approaches

- Analytical approach
  - A number of formulas have been derived for some standard situations (e.g., difference in means between two groups).
  - Usually, these formulas are fairly restrictive with respect to the underlying assumptions,
  - and are not very flexible with regard to a user's potential needs.
- Simulation-based method
  - A simulation-based approach is most flexible,
  - since it allows to perform power analyses for complex and/or highly specific scenarios.
  - Downside: computation time

The simulation procedure

Simulation procedure

- 1. Generate synthetic data, based on an assumed model, model parameters, and covariate distributions
- 2. Fit a model to the synthetic data
- 3. Do the significance test of interest and record the p-value
- 4. Repeat 1.-3. many times
- 5. The statistical power is the proportion of p-values that are lower than a specified α-level

The **powersim** command

- Flexible power analysis for linear and generalized linear models
- Automated simulations, based on user input via command options
- **powersim** creates a do-file that is used for generating predictor data
- The do-file can be modified for more complex synthetic datasets and/or user defined link functions
- The analysis model can be specified using Stata's **regress** or **glm** commands
- A summary of results is shown in the results pane
- Simulation results from each replication are stored in a dataset
- Power curves can be plotted using **powersimplot**

Specification of a data generating model

- Users can choose a distributional family,
- a link function,
- covariates with specified distributions,
- effect sizes for the respective regression parameters,
- correlated predictor variables (for Gaussian variables),
- interaction effects

Available distributional families

Family

- Gaussian
- Inverse Gaussian
- Gamma
- Poisson
- Binomial
- Negative binomial

Available link functions

Link function

- identity
- log
- logit
- probit
- complementary log-log
- odds power
- power
- negative binomial
- log-log
- log-complement

Available covariate distributions

Covariate distribution

- normal
- Poisson
- uniform
- binomial
- χ^2
- Student's t
- beta
- gamma
- negative binomial
- equally sized groups
- 2x2 block design

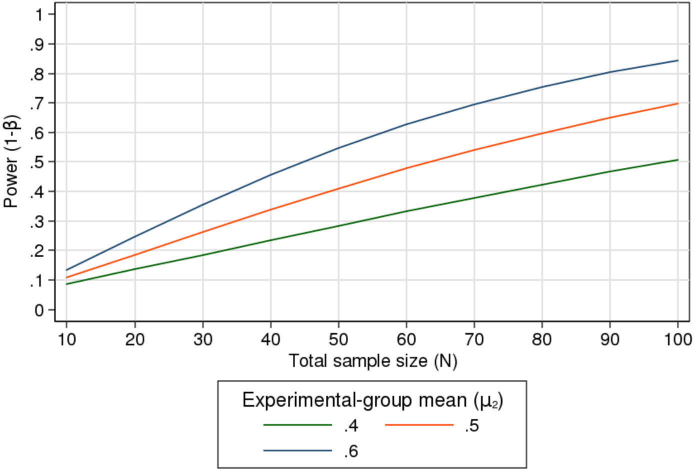
Example 1: Simple comparison of means in a linear model

- Suppose we would like to compare two independent means and calculate power for varying mean differences, measured in standard deviation units, and a varying number of sample sizes.
- In Stata, we can calculate the statistical power for the different effect and sample size combinations with the **power** command:

Stata's power command:

```
power twomeans 0 (0.4 0.5 0.6), n(10(10)100) ///
graph(ylabel(0(.1)1) title("") subtitle("") ///
xval recast(line))
```

Example 1: mean differences (with Stata's **power** command)



Parameters: α = .05, μ_1 = 0, σ = 1

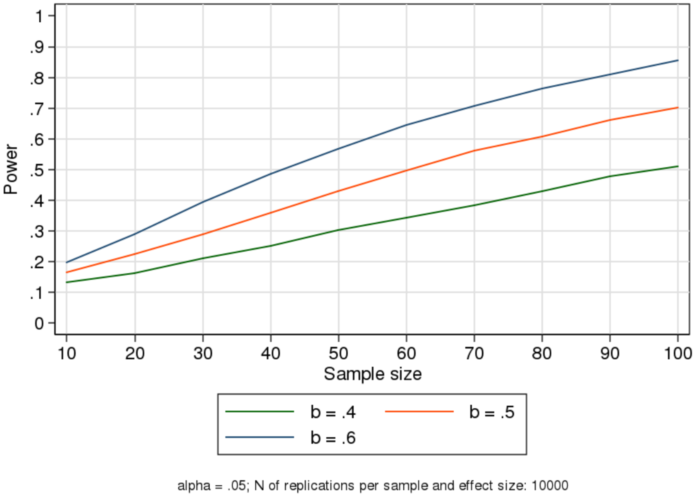
Example 1: Simple comparison of means in a linear model

Now we can replicate these results using simulations (assuming a linear model with Gaussian error and two equally sized (fixed) groups):

powersim code:

```
powersim , ///
b(0.4 0.5 0.6) ///
alpha(0.05) ///
pos(1) ///
sample(10(10)100) ///
nreps(10000) ///
family(gaussian 1) ///
link(identity) ///
cov1(x1 _bp block 2) ///
dofile(ex1_dofile, replace) : reg y x1
```

Example 1: mean differences (**powersim** command)



Example 2: Poisson regression with an interaction effect and correlated predictors

- Now suppose that we would like to simulate the power for the test of an interaction effect of two correlated predictor variables in a Poisson model.
- The assumed model can be expressed as: $y \sim Poisson(exp(0.5 - 0.25 * x1 + 0.4 * x2 + \_bp * x1 * x2))$,
- where $\_bp$ is a placeholder for the various effect sizes for which we simulate the power,
- and x1,x2 $\sim N(\mu, \Sigma)$ with zero means, unit variances, and $\rho = 0.5$
- Now, before we fire up the simulations we create a single synthetic dataset (using **powersim**'s `gendata()` option) in order to check whether the assumed model is consistent with our hypotheses:

Example 2: Poisson regression with an interaction effect and correlated predictors

powersim command:

```
powersim , b(0.1) alpha(0.05) pos(3) ///
sample(300) nreps(500) ///
family(poisson) link(log) ///
cov1(x1 -0.25 normal 0 1) ///
cov2(x2 0.4 normal 0 1) ///
inter1(_bp x1*x2) ///
cons(0.5) ///
corr12(0.5) ///
inside ///
gendata /// // <-- creating a single realization
dofile(ex2_dofile, replace) :  ///
glm y c.x1##c.x2, family(poisson) link(log)
```

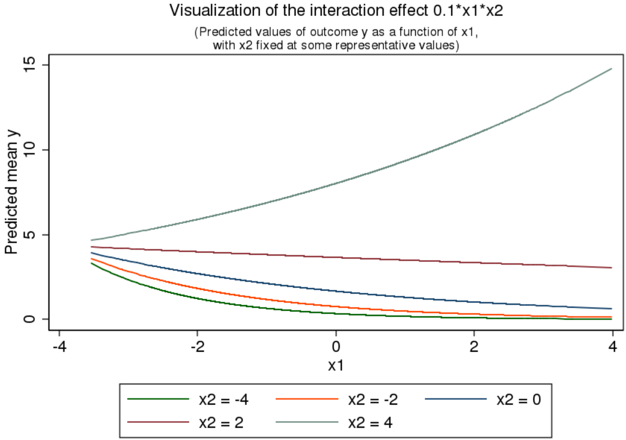
Example 2: Poisson regression with an interaction effect and correlated predictors

Now we could fit the analysis model to the fabricated data:

```
. glm y c.x1##c.x2, family(poisson) link(log) nolog

Generalized linear models                         No. of obs      =      10000
Optimization     : ML                             Residual df     =       9996
                                                  Scale parameter =          1
Deviance         =   11461.6822                   (1/df) Deviance =   1.146627
Pearson          =  10017.90901                   (1/df) Pearson  =   1.002192

Variance function: V(u) = u                       [Poisson]
Link function    : g(u) = ln(u)                   [Log]

                                                  AIC             =    3.25342
Log likelihood   = -16263.10185                   BIC             =  -80604.88

------------------------------------------------------------------------------
             |                 OIM
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |  -.2475788   .0087262   -28.37   0.000    -.2646817   -.2304758
          x2 |   .3971381   .0085716    46.33   0.000      .380338    .4139383
             |
   c.x1#c.x2 |   .0994309   .0056088    17.73   0.000     .0884378     .110424
             |
       _cons |   .5009491    .008613    58.16   0.000      .484068    .5178303
------------------------------------------------------------------------------
```

Example 2: Poisson regression, inspecting synthetic data

... and can do some checking WRT to our hypotheses, for example:



Example 2: Running the simulations

Now we run the simulations by removing the `gendata()` option. We also add a few more sample sizes and add an additional effect size:

powersim command:

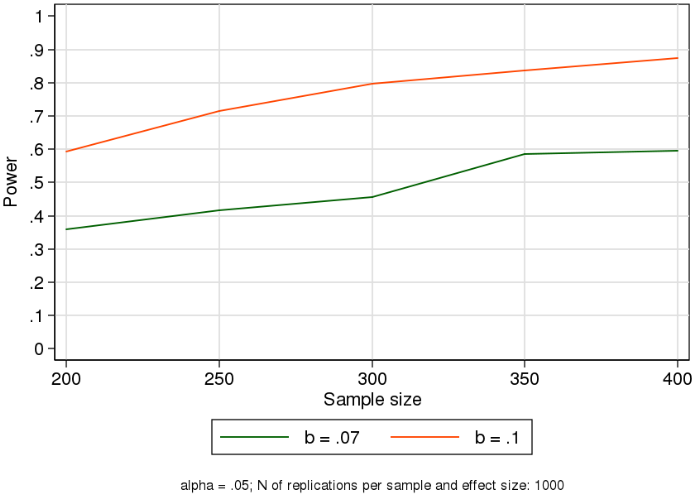
```
powersim , ///
b(0.07 0.1) alpha(0.05) pos(3) ///
sample(200(50)400) nreps(1000) ///
family(poisson) link(log) ///
cov1(x1 -0.25 normal 0 1) ///
cov2(x2 0.4 normal 0 1) ///
inter1(_bp x1*x2) ///
cons(0.5) corr12(0.5) inside ///
dofile(example2_dofile, replace) :  ///
glm y c.x1##c.x2, family(poisson) link(log)
```

Example 2: Output


```
(output omitted)
--------------------------------------------------------------------------

Power analysis simulations

Effect sizes b:        .07 .1
H0:                    b = 0
Sample sizes:          200 250 300 350 400
alpha:                 .05
N of simulations:      1000

do-file used for data generation: example2_dofile
Model command:                    glm y c.x1##c.x2, family(poisson) link(log)

Power by sample and effect sizes:

------------------------
Sample    | Effect size
size      |   .07     .1
----------+-------------
      200 | 0.363  0.608
      250 | 0.416  0.733
      300 | 0.479  0.792
      350 | 0.537  0.853
      400 | 0.607  0.888
------------------------

--------------------------------------------------------------------------
```

Example 2: Power curves

Now we can simply type: `powersimplot`



Example 2: Post-simulation - results dataset


```
. des

Contains data from /tmp/St01559.00000d
  obs:        10,000
 vars:             9                          7 Jul 2013 14:41
 size:       510,000
-------------------------------------------------------------------------------
              storage  display    value
variable name   type   format     label      variable label
-------------------------------------------------------------------------------
nd              double  %10.0g               Iteration ID
b               double  %10.0g               Effect b
se              double  %10.0g               Standard error of b
p               double  %10.0g               p-value
n               double  %10.0g               Sample size
c95             byte    %8.0g                95% coverage (1=covered)
power           byte    %8.0g                1 = p < .05
esize           double  %10.0g               Effect size
esize_id        byte    %8.0g      eid       Effect size ID
-------------------------------------------------------------------------------
Sorted by:  n  esize_id
```

Example 2: Post-simulation - inspecting simulation results

- Example: 95% CI coverage

```
. tabstat c95 if esize_id==2, by(n)

Summary for variables: c95
     by categories of: n (Sample size)

       n |      mean
---------+----------
     200 |       .95
     250 |      .959
     300 |      .949
     350 |      .945
     400 |      .951
---------+----------
   Total |     .9508
--------------------
```

- User-written commands for analyzing simulation results:
  - simsum from Ian White (SSC)
  - simpplot from Maarten Buis (SSC)

Outlook

- Implementing additional features:
  - More models:
    - (un)ordered categorical
    - zero-inflated count models
    - beta regression
    - random effects models
    - **meglm**
  - Correlated predictor data:
    - binary-binary
    - binary-normal
  - Dialog box (?)

Thank you!

Contact:

joerg.luedicke@ufl.edu