

Computing aspects of power for multiple regression

WILLIAM P. DUNLAP, XUE XIN, and LEANN MYERS
Tulane University, New Orleans, Louisiana

Rules of thumb for power in multiple regression research abound. Most such rules dictate the necessary sample size, but they are based only upon the number of predictor variables, usually ignoring other critical factors necessary to compute power accurately. Other guides to power in multiple regression typically use approximate rather than precise equations for the underlying distribution; entail complex preparatory computations; require interpolation with tabular presentation formats; run only under software such as Mathematica or SAS that may not be immediately available to the user; or are sold to the user as parts of power computation packages. In contrast, the program we offer herein is immediately downloadable at no charge, runs under Windows, is interactive, self-explanatory, flexible to fit the user's own regression problems, and is as accurate as single precision computation ordinarily permits.

Computational shortcuts regarding matters as serious as estimating aspects of power in research are useful only if they involve all of the critical factors in a power determination. Consider, for example, the rule of thumb suggested by Tabachnick and Fidell (1996) that the sample size, N , should equal or exceed $50 + 8p$, where p equals the number of predictor variables. Reading the text, we find that this rule assumes a fixed alpha equaling .05 and a "medium-sized" relationship. Clearly, the probability of a Type I error (alpha) and the effect size are important factors, but with simple rules these critical variables are either fixed or are loosely specified. For a more complete discussion of such rules of thumb, see Green (1991) or Maxwell (2000).

Early attempts to provide more specific information about power were ambiguous graphical depictions in the form of power curves or functions (e.g., Myers & Well, 1995), which required the computation of noncentrality parameters as well as skill at graph reading and interpolation. The use of power tables, such as those of Cohen (1988), provided more accessible and better accepted approaches to power analysis, but these graphs still required the preliminary calculation of noncentrality parameters, as well as tabular interpolation. Certainly, with the surge in available computing hardware, in particular

the almost ubiquitous presence of personal computers, better interactive algorithms should provide more precise and individualized results, while sparing the user from any of the complex mathematical computations and decisions necessary for the answers to important power questions. Providing such a program is the purpose of this article.

Effect Size

At the beginning of his article, Maxwell (2000) emphasizes that determination of the appropriate "effect size" is a serious impediment to power analysis. This is the relevant factor most often left unspecified by more simplistic rules of thumb in power for regression analysis. For a meaningful power analysis, however, one must specify this value with respect to one's own research. This can be done on the basis of previous research in the area, estimating from past regression findings from experiments most like the one envisioned. Alternatively, one can approach the problem in terms of utility analysis, asking what advantage, in terms of a process of selection or prediction, an equation might have, usually framed in monetary amounts (e.g., Cascio, 2000).

Estimation of effect size in regression and correlation is complicated by the fact that many authors argue that squared correlations are more interpretable than the correlation coefficient itself because the square is related to the proportion of the variance explained or predicted. In the case of multiple regression, this is the proportion of criterion variance accounted for by an optimally selected weighted average of the predictor variables. But what exactly does the criterion variance mean? We know that for fairly symmetric data, about four to five standard deviations will encompass the vast majority of data points, but the variance has no such simple interpretation in terms of raw data. In fact, the variance may be either larger or

William P. Dunlap of the Department of Psychology at Tulane University died on February 28, 2002, during the revision of this manuscript. His Web site will be maintained, as well as the program available from it as described in the text. Numerous other programs that Bill wrote and published over the years are also available in executable form from his Web site. Copies of the FORTRAN statements upon which the executable version of this multiple regression power program are based can be obtained from L. Myers, Department of Biostatistics, School of Public Health and Tropical Medicine, Tulane University, 1440 Canal St., Suite 2001, New Orleans, LA 70112 (e-mail: myersl@tulane.edu).

smaller than the standard deviation, depending upon the scale of measurement.

Effect size can be expressed in terms of the multiple R^2 or some equivalent coefficient such as a noncentrality parameter, λ , or the measure Cohen (1988) prefers, $f^2 = R^2/(1-R^2)$. Following the rule "when in doubt, keep things simple," our program asks for effect size in terms of the multiple R . If one translates the f^2 values Cohen gives for what he terms small, medium, and large effects to the form of R , one finds that they are $R = .14, .36$, and $.51$, respectively, so users unwilling to directly confront the issue of effect size might plan their research to detect a medium effect size, or population R of $.36$.

Regression Model

Sampson (1974) and Gatsonis and Sampson (1989) point out that the regression model used has implications for how one computes power in regression research, whether the predictors are considered fixed (the conditional model) or random (the unconditional model), although it has no bearing upon the test of significance (the F ratio and corresponding alpha). Because one has fixed variables only after data collection, the authors above conclude that the best model for planning power is the random, or unconditional, model. Although Cohen (1988) uses the fixed, or conditional, model, we have chosen to use the random model, on the basis of the recommendation of the authors cited above.

Also, our power calculations have to do with the significance test of the overall multiple regression coefficient, rather than tests of a single predictor within the context of a larger regression model (e.g., Maxwell, 2000). Although testing the partial contribution of single predictors in the context of other predictors is a frequently used procedure subsequent to a significant multiple R , it is not without interpretive problems (see, e.g., Dunlap & Landis, 1998).

THE PROGRAM

The program was written in FORTRAN, supplemented by subroutines from the IMSL (1982) package. The focal routine for the computations was an algorithm by Ding and Bargmann (1991) that computes integrals from the distribution of R^2 , assuming random multivariate normal predictors and criterion. The functions suggested by Ding and Bargmann for computing the natural logarithm of the gamma function and for the incomplete beta function were replaced by their IMSL equivalents to simplify the resulting program. Other IMSL functions used in the final program were FIN and CHIIN from the IMSL package to get critical values for F and χ^2 , respectively. The program allows the user to compute power, necessary sample size, or detectable population R . The input is interactive and, we hope, self-explanatory. We have compiled the program on both the Tulane University computer, a RISC 6000, and for PCs running under the Windows operating system (using Visual FORTRAN, Version 6.0,

professional edition). Computational delays between input and computed results are negligible on both platforms. When computing necessary sample size, the present program always rounds up, so that a sample of 33.2 was rounded to 34, which we believed was a reasonable resolution of fractional subjects.

DISCUSSION

Comparisons With Other Programs

When compared with the results of Gatsonis and Sampson (1989) in terms of necessary sample size, our program reproduced the exact values given in their Table 1. When the output of the present program was compared with the tables in Cohen (1988), the greatest discrepancies were seen for smaller sample sizes and fewer numbers of predictors. We believe that these differences are the result of either the power approximations that Cohen used or the fixed model approach used by Cohen, as opposed to the random model fundamental to the present solution.

We compared our solutions with those of a commercially available program by Borenstein, Rothstein, and Cohen (1997) (see also Rothstein, Borenstein, Cohen, & Pollack, 1990). Monte Carlo simulation was used to help decide whether our values were more accurate. We sampled repeatedly (100,000 times) from a population with

Table 1
Comparisons for the Power Solutions Using the Exact Method (Current Program), the Approximation Method (Borenstein, Rothstein, & Cohen, 1997) and the Empirical Monte Carlo Simulation ($\alpha = .05$, Number of Iterations = 100,000)

Number of Predictors	Population R^2	Present Program	Borenstein, Rothstein, & Cohen (1997)	Empirical
<i>(n = 15)</i>				
3	.85	.9983	1.0000	.9654
	.50	.7091	.7819	.7104
	.25	.3008	.3211	.3009
	.10	.1249	.1300	.1256
8	.85	.9285	.9795	.9292
	.50	.3188	.3416	.3199
	.25	.1279	.1330	.1289
	.10	.0732	.0748	.0720
<i>(n = 50)</i>				
3	.85	1.0000	1.0000	1.0000
	.50	.9997	1.0000	.9996
	.25	.8999	.9225	.9003
	.10	.4385	.4490	.4372
8	.85	1.0000	1.0000	1.0000
	.50	.9963	.9993	.9966
	.25	.7368	.7620	.7368
	.10	.2712	.2759	.2722
<i>(n = 100)</i>				
3	.85	1.0000	1.0000	1.0000
	.50	1.0000	1.0000	1.0000
	.25	.9979	.9991	.9980
	.10	.7784	.7904	.7807
8	.85	1.0000	1.0000	1.0000
	.50	1.0000	1.0000	1.0000
	.25	.9853	.9906	.9844
	.10	.5903	.5994	.5920

known population R for given sample sizes, and counted the number of sample values that were significant at α , which is the empirical power under those conditions. The results are shown in Table 1. Monte Carlo simulation of the power under identical conditions clearly favored the accuracy of our computations, which is probably a consequence of the reasons offered above for differences from the tables in Cohen's textbook.

We believe that the results of our interactive software are accurate within the bounds of single precision computations and are sufficiently precise that we did not choose to use double precision. Our program at present does not compute confidence intervals on sample multiple regression coefficients, because that is really an accuracy as opposed to a power issue, and as such is perhaps deserving of a future program of its own.

The primary advantage of power programs over tables is flexibility, which our program allows. The number of predictors, alpha levels, effect size, and sample size can all be specified by the researcher. In addition to flexibil-

ity and accuracy, our program has the distinct advantage of being free.

Program Availability

This program runs only on PCs under Windows, and will not run on other personal computers, such as Macintosh. The executable program and FORTRAN source code are offered at no charge to potential users (cf. Borenstein et al., 1997, whose program costs individual users \$595). The PC-executable program can be downloaded directly from the home page of the first author (<http://www.tulane.edu/~dunlap/dunlap.html>). Click on the button labeled "Statistical Programs"; then click on the program name "powmr.exe." This will permit you to download the program to your PC into the location you select. To run the program, simply double-click the program name on your own computer. To end the program operation, you may either enter Ctrl-C at any time, or follow the instructions to enter zeros in response to program queries. If the latter program termination is selected, the

Interactive program (powmr.exe)

```

This program helps design mutiple regression research
in terms of power. You may compute power, needed sample
size, or the detectable population multiple R. The
computations are accurate to limits of single precision

What is your alpha level (<0 to stop)? .05

You may compute either 1=Power, 2=Sample Size,
or 3=Detectable Population R. 0=New alpha.
(enter 0, 1, 2, or 3. ) 2

Enter no. of predictors, population R (<0,0 for new problem) 4,.12
Enter desired power (<0 for new problem) .8
With 4. predictors, and rho = 0.1200
a sample of N = 825 gives power = 0.80000
Enter desired power (<0 for new problem) 0
Enter no. of predictors, population R (<0,0 for new problem) 0,0
You may compute either 1=Power, 2=Sample Size,
or 3=Detectable Population R. 0=New alpha.
(enter 0, 1, 2, or 3. ) 0
What is your alpha level (<0 to stop)? 0
A summary of power calculations is in "powmr.txt"
<press "enter" to stop>

```

Output (powmr.txt)

Alpha = 0.0500

With 4. predictors, and rho = 0.1200
a sample of N = 825 gives power = 0.80000

Figure 1. Sample input and output for running the program.

user will be informed that an output listing of power computations has been written to file "powmr.txt." Figure 1 shows an example of running the program from the PC.

The FORTRAN program that corresponds to the executable program appears in the Appendix. To obtain the FORTRAN file itself, simply e-mail that request to either Leann Myers (myersl@tulane.edu). This FORTRAN source program can be compiled on other computers with very little modification, but it must have access to functions from the IMSL (1982) program package or to equivalent functions.

REFERENCES

- BORENSTEIN, M., ROTHSTEIN, H., & COHEN, J. (1997). *Power and precision*. Teaneck, NJ: Biostat.
- CASCIO, W. F. (2000). *Costing human resources: The financial impact of behavior in organizations*. Cincinnati, OH: South-Western College Publishing.
- COHEN, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.
- DING, C. G., & BARGMANN, R. E. (1991). Algorithm AS 260: Evaluation of the distribution of the square of the sample multiple-correlation coefficient. *Applied Statistics*, **40**, 195-198.
- DUNLAP, W. P., & LANDIS, R. S. (1998). Interpretations of multiple regression borrowed from factor analysis and canonical correlation. *Journal of General Psychology*, **125**, 397-407.
- GATSONIS, C., & SAMPSON, A. R. (1989). Multiple correlation: Exact power and sample size calculations. *Psychological Bulletin*, **106**, 516-524.
- GREEN, S. B. (1991). How many subjects does it take to do a regression analysis? *Multivariate Behavioral Research*, **26**, 499-510.
- INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARIES (1982). *International mathematical and statistical libraries reference manual* (9th ed.). Houston, TX: Author.
- MAXWELL, S. E. (2000). Sample size and multiple regression analysis. *Psychological Methods*, **5**, 434-458.
- MYERS, J. L., & WELL, A. D. (1995). *Research design and statistical analysis*. Hillsdale, NJ: Erlbaum.
- ROTHSTEIN, H. R., BORENSTEIN, M., COHEN, J., & POLLACK, S. (1990). Statistical power analysis for multiple regression/correlation: A computer program. *Educational & Psychological Measurement*, **50**, 819-830.
- SAMPSON, A. R. (1974). A tale of two regressions. *Journal of the American Statistical Association*, **69**, 682-689.
- TABACHNICK, B. G., & FIDELL, L. S. (1996). *Using multivariate statistics* (3rd ed.). New York: HarperCollins.

ARCHIVED MATERIALS

The following materials associated with this article are retrievable from the Psychonomic Society's Norms, Stimuli, and Data archive at <http://www.psychonomic.org/archive/>.

To access the above files or links, search the archive for this article using the journal (Behavior Research Methods, Instruments, & Computers), the first author's name (Dunlap) and the publication year (2004).

FILE: Dunlap-BRMIC-2004.zip

DESCRIPTION: Includes the file "power.f" which contains the FORTRAN source code for the power for multiple regression program described in this article. The program was written using Visual FORTRAN Professional Edition 6.0A, and should compile on any FORTRAN system that has access to the IMSL subroutine package.

AUTHOR'S WEB SITE: <http://www.tulane.edu/~dunlap/dunlap.html>.

APPENDIX
FORTRAN Program Listing

```

c Computes aspects of power for multiple R using powr2 & sqmcor
c if running with Visual Numerics PC Fortran, uncomment the next line
c use numerical_libraries
  write(*,1)
1  format(' This program helps design multiple regression research'/
* ' in terms of power. You may compute power, needed sample'/
* ' size, or the detectable population multiple R. The'/
* ' computations are accurate to limits of single precision')
2  write(*,3)
3  format('/' What is your alpha level (0 to stop)? ', $)
  read(*,*)alpha
  if(alpha.le.0.0.or.alpha.ge.1.0)stop
4  write(*,5)
5  format('/' You may compute either 1=Power, 2=Sample Size, '/
* ' or 3=Detectable Population R. 0=New alpha. '/
* ' (enter 0, 1, 2, or 3. ) ', $)
  read(*,*)kk
  if(kk.le.0.or.kk.gt.3)goto 2
  goto (6,9,17),kk

c
c Calculate Power
c
6  write(*,7)
7  format('/'Enter no. of predictors, N, population R (0,0,0 for new
* problem) ', $)
  read(*,*)pv,xn,rho
  if(pv.le.0.0.or.xn.le.0.0.or.rho.lt.0.0.or.rho.gt.1.0)goto 4
  rho2=rho*rho
  power=powr2(alpha,pv,xn,rho2)
  write(*,8)power
8  format('/' Power is ',F10.5)
  goto 6

c
c Calculate Sample Size
c
9  write(*,10)
10 format('/'Enter no. of predictors, population R
* (0,0 for new problem) ', $)
  read(*,*)pv,rho
  if(pv.le.0.0.or.rho.le.0.0.or.rho.ge.1.0)goto 4
  rho2=rho*rho
11  write(*,12)
12 format('/'Enter desired power (0 for new problem) ', $)
  read(*,*)powr
  if(powr.le.0.0.or.powr.ge.1.0)goto 9
  xn1=chiin(1.0-alpha,pv)/rho2
  p1=powr2(alpha,pv,xn1,rho2)
  xn2=powr/p1*xn1
  if(xn2.le.pv+2.0)xn2=pv+2.0
  p2=powr2(alpha,pv,xn2,rho2)
  iter=0
13  continue
  if(p1.eq.p2)goto 11
  xnn=xn1+(powr-p1)/(p2-p1)*(xn2-xn1)
  iter=iter+1
  if(xnn.lt.pv+2.0)xnn=pv+2.0
  pn=powr2(alpha,pv,xnn,rho2)
  if(abs(powr-pn).le.0.00001.or.iter.gt.300)goto 15
  if(abs(power-p1).gt.abs(power-p2))goto 14
  if(xn2.eq.xnn)goto 11
  xn2=xnn
  p2=pn

```

APPENDIX (Continued)

```

        goto 13
14  continue
    if(xn1.eq.xnn)goto 11
    xn1=xnn
    pl=pn
        goto 13
15  nnn=xnn+1.0
    write(*,16)nnn,powr
16  format(/'Sample Size =',i10,' for power =',f10.4)
    goto 11
c
c Calculate Detectable Rho
c
17  write(*,18)
18  format(/'Enter no. of predictors, N (0,0 for new problem) ',,$)
    read(*,*)pv,xn
        if(pv.le.0.0.or.xn.lt.pv+2.0)goto 4
19  write(*,20)
20  format(/'Enter desired power (0 for new problem) ',,$)
    read(*,*)powr
        if(powr.le.0.0.or.powr.ge.1.0)goto 17
    v1=.1
    p1=powr2(alpha,pv,xn,v1)
    v2=.3
    p2=powr2(alpha,pv,xn,v2)
21  continue
    vn=v1+(powr-p1)/(p2-p1)*(v2-v1)
    if(vn.ge.1.0)vn=.999
        if(vn.lt.0.0)vn=0.0
        pn=powr2(alpha,pv,xn,vn)
    if(abs(powr-pn).le.0.00001)goto 23
        if(abs(powr-p1).gt.abs(powr-p2))goto 22
        if(v2.eq.vn)goto 19
        v2=vn
        p2=pn
        goto 21
22  continue
    if(v1.eq.vn)goto 19
    v1=vn
    p1=pn
        goto 21
23  rho=sqrt(vn)
    write(*,24)rho,powr
24  format(/'Population Multiple R =',f10.4,' has power =',f10.4)
    goto 19
end
c
function powr2(alpha,pv,xn,rho2)
c computes power using sqrcor & fin (IMSL)
    cvf=fin(1.0-alpha,pv,xn-pv-1.0)
    r2=cvf/(cvf+(xn-pv-1.0)/pv)
    powr2=1.0-sqmcor(r2,pv+1.0,xn,rho2,ifault)
    return
end
c
function sqmcor(x,p,xn,rho2,ifault)
c
c Function sqmcor computes the C.D.F. for the distribution
c of multiple R squared (R2) (Ding & Bargmann, 1991, Algorithm AS 260,
c Appl. Stat., 40, 195-198)
c Input: value of multiple R squared (x), total number of
c variables (p), sample size (xn), & population multiple R
c squared (rho2)

```

APPENDIX (Continued)

```

c Output: integral from 0 to R2 of R2 distribution; probability
c   of values of R2 less than or equal to input R2.
c Uses: alngam & betdf from IMSL
c
c   use numerical_libraries
c     errmax=1.0e-6
c       itrmax=100
c
c     sqmcor = x
c     ifault = 2
c     if(rho2.lt.0.0.or.rho2.gt.1.0)return
c       if(p.lt.2.0.or.xn.le.p)return
c     ifault = 3
c     if (x.lt.0.0.or.x.gt.1.0)return
c     ifault = 0
c     if (x.eq.0.0.or.x.eq.1.0)return
c
c     a = 0.5*(p-1.0)
c     b = 0.5*(xn-p)
c
c       Initialize the series
c
c     bet = exp(alngam(a)+alngam(b)-
c *       alngam(a+b))
c     temp = betdf(x,a,b)
c     gx = exp(a*log(x)+b*log(1.0-x)-log(a))/bet
c     q = (1.0-rho2)**(a+b)
c     xj = 0.0
c     term = q*temp
c     sumq = 1.0-q
c     sqmcor = term
c
c     perform recurrence until convergence is achieved
c
c     xj = xj+1.0
c     temp = temp-gx
c     gx = gx*(a+b+xj-1.0)*x/(a+xj)
c     q = q*(a+b+xj-1.0)*rho2/xj
c     sumq =sumq-q
c     term = temp*q
c     sqmcor =sqmcor+term
c
c     Check for convergence and act accordingly
c
c     errbd = (temp-gx)*sumq
c     if ((int(xj).lt.itrmax).and.(errbd.gt.errmax))goto 1
c     if (errbd.gt.errmax)ifault=1
c     return
c     end

```
