

A SAS® Module/Macro Program for Computing Statistical Power Curves for Detecting Treatment Effect

David Olaleye, SAS Institute Inc., Cary, NC

ABSTRACT

PowerReg is a user-written SAS® macro program for computing statistical power and sample sizes necessary to detect a treatment effect given a variety of designs and sampling plans. This program adopts a computer-intensive approach for deriving statistical power curves for different ranges of sample size choice. Although the primary focus of the program is to obtain adequate sample sizes for testing the hypothesis of a given treatment effect difference, emphasis is also placed on the method to use to generate such statistical computations under special circumstances. We illustrate the use of the program with a data set collected from a clinical trial testing a new nutritional formula for low birth weight babies. Its functionality also covers carrying out a sensitivity analysis for a different magnitude of treatment effect difference and repeated analysis of the same data set after elimination of outliers. Results are presented in a simple-to-read tabular format. An optional graphical plot (using SAS/GRAPH®) of power versus sample size can also be requested.

INTRODUCTION

In planning the next phase of a clinical trial, one of the many challenges faced by statisticians is the type and validity of the statistical method to use to power a study to detect the treatment effect of a new drug over a placebo. The challenges may involve a situation where prior information from the last phase of the trial suggests non-normal distributions of the efficacy endpoint data, or that the sources (given multi-center clinical studies) of the samples may not have the same level of variation. Different approaches to solve this problem have been advanced in the statistical literature. Further challenges are raised by the use of these approaches in terms of computational resources, the selection of appropriate statistics for testing and how to validate their results. Given the complex nature of some clinical trial data, the current practice is to choose a method that does not violate the statistical properties of the estimator to be used for hypothesis testing, based on what is known about the properties of the estimator and the data set itself.

Here I present a SAS macro program that employs a randomization method for deriving statistical power curves and sample size estimates needed to detect a clinically meaningful treatment difference for a Phase-3 clinical trial. In the next two sections, I give a brief description of the randomization method and its implementation using the PowerReg SAS macro program, and describe the program features and other functions that it performs.

METHODOLOGY

Randomization test is a method of statistical inference used for comparing an observed test statistic with a distribution that is generated by re-ordering the data values in some sense (Manly, 1997). The test has the advantages of being valid with non-random samples and allowing the user to choose a test statistic that is appropriate for the particular situation under consideration. However, the disadvantage of this method, other than being a computationally intense procedure, is that it can only be used to test hypotheses involving two or more groups. Also, it is not applicable to testing hypotheses regarding absolute values of parameters of populations (Manly 1997).

The PowerReg SAS macro program utilizes the powerful SAS/IML® language and a series of SAS data steps to fit a univariate or

multivariate linear regression model of the primary efficacy endpoint on a given set of independent variables. The implementation of the randomization technique using the PowerReg SAS macro is described below.

Random samples of varying equal sizes per treatment group will be generated from the available observed data set. The simulated sample size per group will vary from a specified minimum sample size (e.g., 25) needed to detect a treatment effect to a specified maximum size. For each generated sample, the primary endpoint linear model will be executed using the linear regression procedure in SAS/STAT® (the REG Procedure), and the treatment effect in the model tested for statistical significance at the type I error rate of 5% using a two-sided testing strategy. The statistical power is estimated by the percentage of rejections of the null hypothesis of no treatment effect. 1000 independent samples will be generated for each sample size.

The PowerReg macro program also performs sensitivity analyses given different clinically meaningful treatment effect differences. That is, several statistical power curves can be generated by plotting estimated power against sample size for different magnitudes of treatment effect sizes. Further, it has the capability to check for any outliers or influential observations (via another SAS macro, %Outlier) in the sample data by fitting a linear regression model to the observed data. If such observations exist, they are subsequently deleted, and the analysis is re-performed on the remaining sample data. An observation will be classified as an outlier if the absolute value of the DFBETAS (obtained from the influence option in the REG procedure) for the treatment effect in the model is greater than 2 times the reciprocal of the square root of the sample size (Belsley, Kuh, and Welsch 1980).

Features of PowerReg

The PowerReg SAS macro program is composed of three parts: (1) the input and output module (%PowerReg), (2) the simulation module (%Check), and (3) the outlier module (%Outlier). Each module is described in detail below.

Step 1: Input and Output Module

The macro parameters DSET1 and DSET2 receive names of SAS data sets containing data points for the treatment and placebo (or control) groups, respectively. The name can be a one-level, temporary SAS data set name (&dset1) or a two-level SAS data set name (POWER.&dset1). The parameters OUTCOME and PREDICT hold the variable names for the observed and computed efficacy endpoints, respectively. The latter variable is obtained by multiplying the observed primary endpoint (OUTCOME) by the treatment effect difference (SCALE). Here the magnitude of the treatment effect difference denotes a scientifically significant increase. The parameter INDEPVAR contains the names of the independent (regressor) variables, separated by blanks. [Note: to avoid macro resolution errors, I suggest enclosing the names of the regressor variables in parentheses using %STR() macro function.] The parameters ALPHA and SEED contain the test level of rejection (for example, $\alpha=0.05$) and the seed for the uniform random generator. SSRANGE is the macro parameter that contains the range of desired sample sizes for which power is sought.

The NLOOP parameter asks for the number of simulations for each sample size. The parameter ALLBASE receives input for the name of the SAS data set to store the final results of power and sample size computations. The last three parameters- PLOTIT, TITLE2, and VERIFY request for plot of estimated power versus sample size, a subtitle for the graphical plot, and the name of the SAS data set that contains the intermediate regression results performed for each run. Depending on the number of runs (NLOOP) requested, the size of this data set could be very large. However, the data set is useful should there be a need to validate the results obtained from the PowerReg macro program or to perform additional analysis on the statistical properties of the test statistic.

```
options ls=90 ps=54 fmtsearch=(power)
        nosymbolgen nomprint nosource cleanup;
```

```
libname power 'c:\powerreg'
```

```
%macro powerreg(
    dset1=, /*Treatment Data Set*/
    dset2=, /*Placebo or Control Data Set*/
    predict=, /*Predicted Outcome variable*/
    outcome=, /*Initial outcome variable*/
    indepvar=, /*Indept vars., separated by*/
                /*blanks: e.g. %str(x1 x2 x3)*/
    alpha=, /*alpha: 0.05 for 2-tailed*/
    scale=, /*Effect size*/
    seed=, /*Specify random seed*/
    ssrange=, /*Range of sample sizes for desired*/
                /*power e.g.ssrange=20 300 25; */
                /*start_stop_byvalue */
    nloop=, /*Number of simulations: e.g.1000 */
    allbase=, /*Output data set for report and*/
                /*graphic plots*/
    plotit=, /*Request plot (Y=Yes or N=No)*/
    title2=, /*Add optional graph title*/
    verify=); /*Store intermediate results for*/
                /*validation test*/
```

```
FILENAME logfile 'c:\sas\saswork\tracklog.txt';
proc printto log=logfile; run;
```

```
proc format;
    value treat 1="&dset1" 0="&dset2";
run;
```

```
%let min_size=%scan(&ssrange,1,%str( ));
%let max_size=%scan(&ssrange,2,%str( ));
%let by_value=%scan(&ssrange,3,%str( ));
```

```
/*Limited error checking*/
%if &min_size= %then %do;
    proc printto log=log; run;
    %put ERROR: Range of samples size desired
        not specified.;
    %goto exit;
%end;
```

```
%if &max_size= %then %do;
    %let max_size=&min_size;
    %put NOTE: Desired maximum sample size
        range not specified. Will use
        &min_size instead.;
%end;
```

```
%if &by_value eq %then %let by_value=1;
```

```
%macro gen_data;
    %global keep_;
    %local i;
```

```
%do i=1 %to 2;
    %if &i=1 %then
        %do; %let seed_=&seed;
            %let _dset_=_treat;
        %end;
    %else %if &i=2 %then
        %do; %let seed_=%eval(&seed+1000);
            %let _dset_=_placebo;
        %end;
```

```
%let keep_=&outcome &indepvar;
%if %bquote(&keep_) ne %then
%do;
```

```
data &dset_;
    set &&dset&i (keep=&keep_); run;
%end;
```

```
data &dset_;
    length trtgroup $20;
    trtgroup="&&dset&i";
    do sample=&min_size to &max_size by
        &by_value;
```

```
/*indicate N(simulations) eg.nloop=1000*/
    do loop=1 to &nloop;
        do i=1 to sample;
            ran_seed=ranuni(&seed_);
            select=int(ran_seed*n)+1;
            set &dset_point=select nobs=n;
            if _error_ then abort;
            if sample then output;
        end;
    end;
end;
stop;
run;
%end;
%mend gen_data;
%gen_data /*macro to generate data sets*/
```

```
/*Execute regression module for each sample
size run.*/
```

```
%do j=&min_size %to &max_size %by &by_value;
    %check(ssize=&j,_dset1=T_&j,_dset2=P_&j)
%end;
```

```
proc printto log=log; run;
proc sort data=&allbase; by factor label;
run;
```

```
data &allbase ;
    length trtgroup $30;
    set &allbase; by factor label;
    _effect=put(factor,4.1);
    trtgroup="Effect="||_effect;
run;
```

```
options center nonumber nodate ps=54 ls=80;
proc tabulate data=&allbase order=data
    format=9.;
    class label predict ssize trtgroup;
    var factor power totalsum count ;
    table (label='Treatment Comparison= '),
        (ssize='Sample Size'),
        (max=' '*f=best9.) *
        (trtgroup='Treatment Effect') *
        (power='Power'
        count='Number of Runs'
        totalsum='Total H0 Rejection')
        / misstext='0' rts=20 condense box=_page_;
```

```
title1 "Power and Sample Size Calculations
        for Detecting Treatment Effect";
title2 "&title2";
run;
```

```
%if %upcase(&plotit) eq Y %then
%do;
```

```
goptions reset=all border device=win
    rotate=landscape vorigin=1 horigin=1
    gunit=pct htitle=4.0 htext=3.0
    vsize=6.0 hsize=8.5;
```

```
symbol1 width=2.0 height=2.0 i=rc
    color=black v=*;
symbol2 width=2.0 height=2.0 i=rc
    color=black v=+;
symbol3 width=2.0 height=2.0 i=rc
    color=black v=diamond;
```

```
AXIS1 label=(height=2.5 'Power')
    ORDER=(0 TO 100 BY 10) COLOR=BLACK
    width=2.0;
AXIS2 label=(height=2.5 'Sample Size')
    ORDER=(&min_size TO &max_size BY
    &by_value) COLOR=BLACK width=2.0;
```

```

legend1 value=(height=2.0) frame;

PROC GPLOT DATA=&allbase;
PLOT power*ssize=TRTGROU / HAXIS=axis2
      VAXIS=AXIS1 legend=legend1;
TITLE1 h=2.5 j=c "Plot of Estimated Power
Versus Sample Size to Detect
Treatment Effect";
TITLE2 h=2.0 j=c "&title2";
RUN; QUIT;
%end;

%exit;
proc printto log=log; run;
%mend powerreg;

```

The output of the PowerReg macro program is a SAS data set containing the following: the estimated power (POWER) needed to detect a treatment effect difference (FACTOR) for each sample size, sample size (SSIZE), the number of simulations (COUNT), and the total number of rejections (TOTALSUM).

Step 2: The Simulation Module

The %CHECK macro performs a linear regression of the efficacy endpoint using the SAS/IML regression module, and this process is replicated NLOOP times. That is, for each sample size, NLOOP regressions are performed and the output stored in a SAS data set. This data set is then used to compute the power, estimated by the percentage of rejections of the null hypothesis of no treatment effect.

```

%macro check(ssize=, _dset1=, _dset2=);
  data &_dset1; set _treat;
    if sample=&ssize;
  run;

  data &_dset2; set _placebo;
    if sample=&ssize;
  run;

  %do _loop=1 %to &nloop;
    data _reg;
      retain intercept 1; /*intercept term*/
      set &_dset1 (in=in1) &_dset2;
      where sample=&ssize and loop=&_loop;
      if in1 then treat=1; else treat=0;

      /*Create new response variable*/
      if in1 then &predict=&outcome*&scale;
      else &predict= &outcome;
    run;

    proc iml;
      reset noprint ;
      reset fw=7;

      start regress;
      use _reg;
      read all var {intercpt treat
        &indepvar} into xvars;
      read all var {&predict} into yvar;
      xvarname={Intercept treat &indepvar};
      xvarname=xvarname`;

      n=nrow(xvars);
      k=ncol(xvars);
      xx=xvars` * xvars;
      xy=xvars` * yvar;
      xxi=inv(xx);
      parmest=xxi * xy;
      yhat=xvars * parmest;
      resid=yvar - yhat;
      _sse_=resid` * resid;
      df=n-k;
      _mse_=_sse_ / df;
      covb=xxi#_mse_;
      std_err=sqrt(vecdiag(covb));
      t_for_H0=parmest / std_err;
      p_value=1-probf(t_for_H0#t_for_H0,1,df);
      print xvarname parmest std_err
        t_for_H0 p_value;

      create _est_ var{xvarname parmest
        std_err t_for_H0 p_value};
      append;
      close _est_;
      finish regress;
      run regress;
    quit;

    data _work1;
      length label $20 predict $10;
      set _est_;
      where left(xvarname)='TREAT';
      if p_value<=0.05 then reject=1;
      else reject=0;
      if parmest<0 and p_value<=&alpha then
        placebo=1; else placebo=0;
      if parmest>=0 and p_value<=&alpha then
        trtmnt=1; else trtmnt=0;
      loop=&_loop;
      ssize=&ssize;
      factor=&scale;
      predict=symget('predict');
      label="&dset1 vs &dset2";
    run;

    proc append base=base data=_work1 force;
    run;
  %end;

  proc sort data=base; by ssize loop; run;
  data _temp(keep=label predict parmest
    factor ssize power totalsum count
    prop_plc prop_trt);
  set base end=eof;
  by ssize loop;
  if first.ssize then
    do;
      count=0; totalsum=0; tot_plc=0;
      tot_trt=0;
    end;
    count+1;
    totalsum+reject;
    tot_plc+placebo;
    tot_trt+trtmnt;
    if last.ssize=1 then
      do;
        power=100*(totalsum/count);
        prop_plc=100*(tot_plc/count);
        prop_trt=100*(tot_trt/count);
      end;
    if last.ssize=1 then output;
  run;

  proc append base=&allbase data=_temp
    force;
  run;

  %if %upcase(&verify) eq Y %then
  %do;
    proc append base=_base_ data=base
      force;
    proc datasets library=work
      memtype=data;
      save _treat _placebo &allbase _base_;
    run; quit;
  %end;
  %else %do;
    proc datasets library=work
      memtype=data;
      save _treat _placebo &allbase;
    run; quit;
  %end;
  %mend check;

```

Step 3: The OUTLIER Module

The %OUTLIER macro checks for the presence of outliers in the pooled data set containing both treatment and placebo data points. This code is executed if there is a reason to believe that the observed results are being driven by some influential observations present in the data set. The macro first concatenates the treatment and placebo data sets, and the resulting data set is passed as an input data set to the

SAS/IML module for regression analysis.

[Here, we utilized the SAS Institute supplied IML modules to perform the regression. The SAS/IML program is included with the SAS System software and can be found under the SAS Sample Library. For this presentation, the code has been slightly modified and expanded to save the DFBETAS for the covariates in the regression model to a SAS data set.]

The SAS data set containing the DFBETAS is merged with the original pooled data set for the purpose of identifying outliers. If influential observations are present (based on the specified criteria), they are eliminated. The resulting data set is partitioned into treatment and placebo data sets. These two data sets serve as the input data sets in the %PowerReg macro, and the code is then re-executed.

```
%macro outlier(
  dset1=, /*Treatment Data Set*/
  dset2=, /*Placebo or Control Data Set*/
  outcome=, /*Initial outcome variable*/
  indepvar=, /*Indept vars., separated by*/
            /*blanks: e.g. %str(x1 x2 x3)*/
  scale=, /*Effect size*/
  seed=, /*Random seed*/
  allbase=); /*Output data set name*/

proc format;
  value treat 1="&dset1" 0="&dset2";
run;

/*Retrieve iml_addin.sas from macro library*/

%include 'c:\powerreg\iml_addin.sas';

data outlier;
  retain intercept 1;
  set &dset1 (in=in1) &dset2;
  keep intercept treat &outcome &indepvar;
  if in1 then treat=1; else treat=0;
run;

proc iml ;
  reset nolog noprint linesize=90 fw=8;
  reset storage='reg';
  load module=(regest regtest regcol regresid
  regplotr regplotp);

  use outlier;
  read all var {intercpt treat &indepvar}
    into x;
  read all var {&outcome} into y;
  read all var {treat} into t;
  names={Intercept treat &indepvar};
  names=names';

  run regest; /*estimate the model*/
  run regresid; /*analyze residuals and*/
  /*create _dfbeta_ data set*/

quit;

data _work2(keep=treat_);
  set _dfbeta_ nobs=n;
  treat_=treat;
  call symput('n',n);
run;

data _trt _plac;
  merge outlier _work2;
  if abs(treat_) > 2/sqrt(&n) then delete;
  if treat=1 then output _trt;
  else output _plac;
run;

/*Execute the PowerReg macro*/
%powerreg(dset1=_trt,
  dset2=_plac,
  predict=_resp_,
  outcome=&outcome,
  indepvar=&indepvar,
  alpha=0.05,
  scale=&scale,
  seed=&seed,
```

```
ssrange=25 300 25,
  nloop=1000,
  allbase=&allbase,
  plotit=Y,
  title2=%str(With Elimination of
  Outliers from Pooled Data Set),
  verify=N)
```

```
%mend outlier;
```

Example

We illustrate the use of PowerReg macro program with a data set comprising birth weight measurements of infants enrolled in a clinical trial testing a new nutritional formula for low birth weight. This was a two-arm study with the objective of determining the superiority of the new nutritional formula over the old formula (referenced in this paper as a placebo treatment). The data used for the power and sample size computations come from four-month data. The exercise is to use the information derived from this trial to run a bigger trial with evaluation of the endpoint at six months.

The primary efficacy endpoint is the percent change in birth weight from baseline to six months. The statistical analysis model for efficacy is the linear regression model of percent change in weight with the independent variables being treatment, baseline weight (measured in grams), site, and gestational age at birth. The study was conducted in four sites, so the site variable is treated as a dummy categorical variable with site 1 as the reference site. To account for the increase in birth weight between the four-month and six-month evaluation, the percent change in weight in the treatment group is multiplied by a factor of 1.1 (and for the sensitivity analysis, a factor of 1.3).

The following statements invoke the PowerReg macro and specify the parameter values required for calculating the power and sample size when the treatment effect sizes are set at 1.1 and 1.3.

```
%powerreg(dset1 =power.trtdata,
  dset2 =power.placebo,
  predict =_resp_,
  outcome =pc_wt,
  indepvar=%str(weight0 gestage
  site2 site3 site4),
  alpha =0.05,
  scale =1.1,
  seed =564781,
  ssrange =25 300 25,
  nloop =1000,
  allbase =iudata_a,
  plotit =Y,
  title2 =%str(),
  verify =N)
```

The power and sample size calculations are displayed in Table 1 and the plot of power versus sample size is shown in Figure 1. A cubic spline is used to plot the estimated power versus the sample size per treatment group. The results suggest that with such a design, a 5% level two-sided test has power of approximately 90% of detecting a treatment effect difference (that is, a factor of 1.1 increase in percent change in birth weight) when the sample size is 225 infants in each treatment group. Similarly, when the treatment effect size is set at 1.3, then the power is 91% for a sample size of 225 infants per group.

The optional %OUTLIER macro can also be executed to check if there are outliers or unusual observations in the data set. If any are found, they are deleted, and %PowerReg macro is re-executed. The following statements invoke the OUTLIER macro required for calculating the power and sample size when the treatment effect sizes are set at 1.1 and 1.3, respectively.

```
%outlier(dset1=%str(power.trtdata),
dset2=%str(power.control),
outcome=pc_wt,
indepvar=%str(weight0 gestage
site2 site3 site4),
scale=1.1,
seed=674781
allbase=iudata_e)
```

The results are displayed in Table 2, and the plot of power versus sample size is shown in Figure 2. If we assume an effect size of 1.1 increase in percent change in birth weight of infants in the treatment group relative to the control group, and with two-tailed type I error at $\alpha=0.05$, a sample of 125 infants in each treatment group would be required to achieve 90% power. For an effect size of 1.3, a sample of 125 infants in each group would yield an approximate 91% power.

CONCLUSION

The PowerReg macro program described in this paper was developed to perform power and sample size calculations necessary to detect a treatment effect difference of a specified size using a randomization test method. The algorithm used here is fairly straightforward but may be computationally intense depending on the number of runs needed to estimate power for each sample size. The next challenge is to find a more efficient algorithm in terms of reduction in code, computer time and resources that can be used to address this type of problem. We would appreciate knowing if you encounter difficulties or have suggestions for improvements.

REFERENCES

Belsley, D.A., Kuh, E., and Welsch, R.E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons, Inc.

Manly, B.F. (1997), *Randomization, Bootstrap and Monte Carlo Methods in Biology, Second Edition*, London: Chapman and Hall, Inc.

SAS Institute Inc. (1990), *SAS/IML Software: Usage and Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/Graph Software: Usage and Reference, Version 6, Sixth Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/STAT User's Guide, Version 6, Fourth Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/IML Software: Usage and Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

SAS, SAS/STAT, SAS/IML, and SAS/GRAPH software are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

CONTACT INFORMATION

Your comments on and improvements to the SAS macro program presented in this paper are welcomed. The author can be contacted at:

David Olaleye
 SAS Institute Inc.
 SAS Campus Drive
 Cary, NC 27513
 Work Phone: 919-677-8000 x2931
 Email: daolal@wnt.sas.com

N	No. of Runs	Treatment Effect			
		Effect=1.1		Effect=1.3	
		Power	Total H0 Reject	Power	Total H0 Reject
25	1000	17.7	177	19.4	194
50	1000	33.0	330	32.7	327
75	1000	46.1	461	45.7	457
100	1000	58.7	587	63.4	634
125	1000	69.3	693	70.2	702
150	1000	74.7	747	78.0	780
175	1000	81.5	815	82.4	824
200	1000	87.2	872	87.9	879
225	1000	89.7	897	91.4	914
250	1000	93.4	934	94.0	940
275	1000	94.7	947	95.5	955
300	1000	96.1	961	97.0	970

Table 2: Power and Sample Size Calculations for Detecting Treatment Effect (With elimination of outliers)

N	No. of Runs	Treatment Effect			
		Effect=1.1		Effect=1.3	
		Power	Total H0 Reject	Power	Total H0 Reject
25	1000	28.5	285	26.8	268
50	1000	51.2	512	54.8	548
75	1000	69.9	699	72.7	727
100	1000	81.6	816	85.9	859
125	1000	89.6	896	91.2	912
150	1000	94.8	948	94.4	944
175	1000	96.7	967	97.5	975
200	1000	98.2	982	98.6	986
225	1000	98.3	983	99.3	993
250	1000	99.4	994	99.6	996
275	1000	99.9	999	99.9	999
300	1000	100.0	1000	99.9	999

Table 1: Power and Sample Size Calculations for Detecting Treatment Effect

Figure 1: Plot of Power versus Sample Size

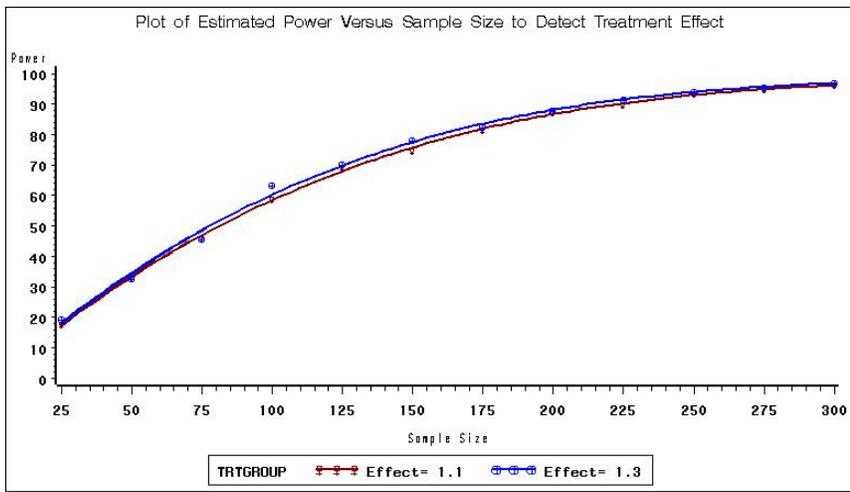


Figure 2: Plot of Power versus Sample Size (With elimination of outliers)

