MWSUG 2024– Paper AL-51

What if Proc Power Can't Help? Calculating Power via Simulation, Example with Count Regression

Brandy R. Sinco, MS, University of Michigan, Ann Arbor, MI
Joceline Vu, MD, University of Michigan, Ann Arbor, MI
Pasithorn Suwanabol, MD, University of Michigan, Ann Arbor, MI

ABSTRACT

Background: Reviewers for an article comparing opioid use after surgery questioned the amount of power for a Zero-Inflated Negative Binomial (ZINB) model. Power was estimated by simulation because Proc Power does not have a ZINB option.

Objectives. For a given dataset, show how to use Procs GenMod or CountReg to estimate the distribution parameters for Poisson, Negative Binomial, ZIP, and ZINB models. Then, use the streaminit() and RAND() functions to simulate distributions in a data stop. Run the outcomes model on the simulated data and compute the power.

Methods. Begin with one simulation. After calculating the ZINB parameters with Proc GenMod, compare the simulated distribution to the actual data with graphics and descriptive statistic from Procs Freq and Univariate.

Next, simulate 1,000 datasets. Put an index on _REPLICATION_ to maximize execution speed. Run the outcomes procedure 1,000 times and compute the simulation-based power as the percentage of times when the outcome is significant. I.E., compute power based on the definition of the probability of observing a significant difference between treatment and control, given that a significant difference is present in the data.

To calculate power for the model with covariates, use bootstrapping. Compare the actual data to the simulated data with Procs Freq, Means, T-Test, and NPar1Way.

Results. After outputting the results of Proc GenMod by _REPLICATION_ into an ODS dataset, the original dataset with 562 patients had 78% power to detect a significant difference in opioid use between surgical procedures without adjustment for additional covariates.

Outline

1)_Background – Reason for Power Request.

Two surgeons, Dr. Vu and Dr. Suwanabol, submitted a manuscript that compared post-operative opioid use between patients who had laparoscopic versus robotic colectomies.

The outcome, number of pills consumed, was a count variable, and was analyzed with a zero-inflated negative binomial model. Poisson and negative binomial regression models are two common models for count data. When the counts have more zeroes, than a Poisson or negative binomial distribution would have, zero-inflated models are a useful option.

Reviewers for the article comparing opioid use after surgery questioned the amount of power for a Zero-Inflated Negative Binomial (ZINB) model.

Brandy was asked to compute power via simulation to address the reviewers' concerns. Because SAS Proc Power can not compute power for ZINB models, Brandy turned to power via simulation.

2)_ Definition of Statistical Power.

Let μ = Parameter of interest, such as a mean, variance, rate, coefficient, or regression curve.

Power = Pr(Detecting difference in μ | Difference in μ is present). In plain English, statistical power is the ability to detect an effect, given that the effect is present. Power is a function of the sample size, the amount of variation in the data, and the magnitude of the effect that is being measured. Power increases with sample size and decreases with larger variance.

However, more power is needed to measure a smaller effect size. At first, this sounds counter-intuitive. The reason is that power is inversely proportional to the effect size that a person is trying to detect. For example, a larger sample size is needed to measure the mean temperature difference between Minneapolis and St. Paul, than between Alaska and Hawaii. I.E., more effort is required to detect a smaller difference.

For an example of hypotheses, for which to calculate power, consider post-operative opioid use between patients having laparoscopic versus robotic colectomies.

- H_0: Null hypothesis, no difference, $\mu_1 = \mu_2$.
- I.E., no impact of surgical technique on post-operative opioid use.

- H_A: Alternative hypothesis, $\mu_1 \neq \mu_2$ or $\mu_1 > \mu_2$. or $\mu_1 < \mu_2$.
- I.E., post-operative opioid use differs by surgical technique.

- Type 1 error, $\alpha = Pr(H_A \mid H_0 \text{ is true})$. α also called "level of significance".
- Type 1 error: Conclude that a difference is present, when difference is due to random variation. I.E., wrongfully conclude that opioid use differs by surgical technique, when there is really no difference.
- P-value = observed level of significance.

- Type 2 error, $\beta = Pr(H_0 \mid H_A)$. Fail to detect a difference when a difference is present.
- **Power = $1 - \beta = Pr(H_A \mid H_A)$. Probability of accurately detecting a difference.**

3)_Simple Example of Power by Simulation with T-Test for Two Means.

Calculating power by simulation is done by Creating a large number of datasets, 1000 or more with simulated distributions of the parameter of interest. Then, Power = % of datasets with statistically significant results.

For example, suppose group 1 has a mean of 2, standard deviation of 2.5, and $N_1 = 10$; group 2 has a mean of 1, standard deviation of 2.25, and $N_2 = 20$. How much power do we have to detect a difference in means between groups 1 and 2? For this question, SAS Proc Power can help. Before diving into zero-inflated negative binomial regression, let's focus on this simpler example.

The formula for power to detect a difference of d between two populations that have variances, σ_1^2 and σ_2^2, is $N = (\sigma_1^2 + \sigma_2^2)(Z_{\alpha/2} + Z_\beta)^2/d^2$; d = minimum detectable difference.

- $F = \sigma_1^2/\sigma_2^2 = (2.5/2.25)^2 = 1.235$, p = .332; use pooled T-test.

```
/* SAS code to find p-value for difference in variance, */
Data _NULL_;
p_value=1-probf(1.235, 9, 19);
Put "F_05_9_19=" F_05_9_19 "P_Value=" P_Value;
Run;

/* Unequal Variances */
proc power;
twosamplemeans test=diff_satt
groupmeans = 2 | 1
GROUPSTDDEVS = 2.5 | 2.25
groupns = (10 20)
power = .;
run;

/* Equal Variances */
```

- `Test = diff instead of diff_satt.`
- `StdDev = 2.333 instead of GroupStdDevs.`

Table of Proc Power Output

Distribution	Normal
Method	Exact
Group 1 Mean	2
Group 2 Mean	1
Group 1 Standard Deviation	2.5
Group 2 Standard Deviation	2.25
Group 1 Sample Size	10

Group 2 Sample Size	20
Number of Sides	2
Null Difference	0
Nominal Alpha	0.05

Computed Power	
Actual Alpha	**Power**
0.0505	**0.172**

Interpretation of the Proc Power output. Assuming unequal variances, power = .172. If a simulated dataset is created for 1,000 repetitions, the means should be significantly different 17.2% of the time.

Assuming equal variances, power = .188. So, simulated dataset should have differing means ~18.8% of the time. If the dataset has a mixture of equal and unequal variances, the power to detect a significant difference in means should be ~ 16% to 20%.

SAS Program to Create 1,000 Datasets of Simulated Means

```
Data MeansTest1000;
N1=10; N2=20;
do Rep=1 to 1000; Call streaminit(Rep);
   do JI=1 to N1;
      X = Rand('Normal', 2, 2.5); Group=1;  output;
   end; /* J1 loop */

   do J2=1 to N2;
      X = Rand('Normal', 1, 2.25); Group=2;output;
   end;/* J2 loop */
end; /* Rep Loop */
Run;

/* T-Tests on Simulated Data */
Proc TTest Data=MeansTest1000;
class Group;
var X;
by Rep;

ods output Equality=EqualVar1000 TTests=TTests1000;
run;
```

Most SAS statistical procedures can run with a “By Rep;” statement.
Run the procedure “by Rep” to calculate power by simulation.

Table of Proc TTest Output

Obs	Rep	Variable	Method	Variances	tValue	DF	Probt
1	1	X	Pooled	Equal	2.80	28	0.0091
2	1	X	Satterthwaite	Unequal	3.08	23.212	0.0052
3	2	X	Pooled	Equal	0.79	28	0.4353
4	2	X	Satterthwaite	Unequal	0.87	23.307	0.3927
5	3	X	Pooled	Equal	0.26	28	0.7944
6	3	X	Satterthwaite	Unequal	0.27	18.671	0.7930
7	4	X	Pooled	Equal	2.07	28	0.0483
8	4	X	Satterthwaite	Unequal	1.86	14.091	0.0832
9	5	X	Pooled	Equal	0.82	28	0.4171
10	5	X	Satterthwaite	Unequal	0.88	21.3	0.3912

```
/* Calculate the Power from the T-tests on 1000 datasets */
Data TTests1000Sig;
set TTests1000;
Sig=(ProbT<.05);  run;

Title 'Power if variance pooled';
/* Power = 20.2% simulation, compared to  18.8% Proc Power */
proc freq data=TTests1000Sig;
tables Sig;
where  Method='Pooled';run;

Title 'Power if Satterthwaite';
/* Power = 17.5% simulation, compared to 17.2% Proc Power */
proc freq data=TTests1000Sig;
tables Sig;
where  Method='Satterthwaite';  run;
```

4)_Count Regression: Poisson, Negative Binomial, Zero-Inflated Models.

Poisson. Let Y = number of occurrences of an event, such as the number of a driver's traffic accidents. In a Poisson distribution, the number of occurrences increases with the area of measure. For example, Pr(accident) increases with distance of commute to work.

- $f(Y=y; \mu) = e^{-\mu}\mu^{Y}/y!$, μ = mean = variance, y = 0, 1, 2, …, ∞.
- Poisson Regresion ln(y) = Xβ.
-

Negative Binomial. Negative binomial regression is an alternative to Poisson regression when variance>mean, also called over-dispersion.

- Negative Binomial Regression uses the log link, same as Poisson regression.
- Scale parameter = Φ, Φ > 0.
- Mean = μ.
- Variance = $\mu + \Phi\mu^2$.

When Φ = 0, the Negative Binomial variance ➔ μ, the variance for the Poisson model.

Zero-inflated count models are for distributions with excess zeroes, where y could be modeled as Poisson or Negative Binomial, if the distribution didn't contain excessive 0's. (Lambert, 1992 and Greene, 1994).

Zero-Inflated Poisson. Mixture of a Bernoulli distribution and a Poisson distribution. Model for distribution with excess zeroes.

- When y=0, p*Logistic + (1-p)Poisson.
- When y>0, (1-p)Poisson.

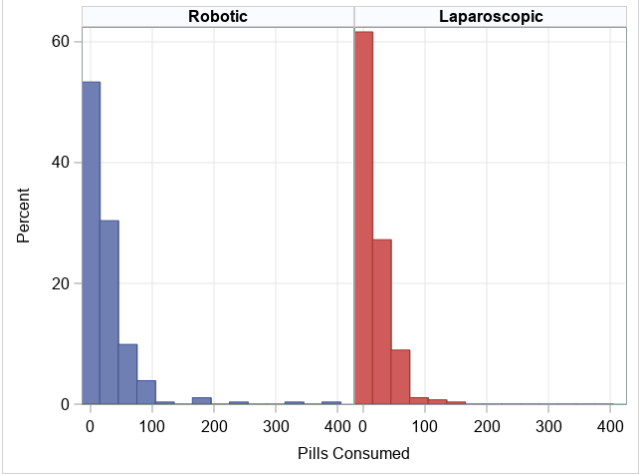
Zero-Inflated Negative Binomial. Mixture of a Bernoulli distribution and a Negative Binomial distribution.

5)_Simulating the Zero Inflated Negative Binomial in SAS.

Opioids consumed had the distribution in the table and graph below.

Pills Consumed	Robotic (N = 283)	Laparoscopic (N = 279)
0	67 (23.7%)	94 (33.7%)
>0 to <5	41 (14.5%)	38 (13.6%)
≥5	175 (61.8%)	147 (52.7%)

The unadjusted opioid consumption trend was significantly different between laparoscopic and robotic surgery patients, p = .010, Cochran-Armitage trend test. Use "trend" option in Proc Freq Tables statement to get the Cochran-Armitage trend test.



Robotic. N=283, mean=23.5, std=42.0.

Laparoscopic. N=279, mean=15.0, std= 21.0

Wilcoxon Test p = 0.007

```
/* SAS Code for Histograms */
Proc SGPanel data=Col_Opioid NOAUTOLEGEND;
Panelby lap / rows=1 columns=2 HEADERATTRS=(Color=Black Size=12
Weight=Bold) NoVarName;
Histogram pills_consumed / group=lap;
rowaxis grid LABELATTRS=(Size=12) VALUEATTRS=(Size=12);
colaxis grid LABELATTRS=(Size=12) VALUEATTRS=(Size=12) values=(0 to
400 by 50);
run;
```

The first step in simulating a ZINB distribution is to estimate p and k separately by surgery type, laparoscopic or robotic.

```
proc genmod data=Col_opioid; *** lap=0, (robotic colectomy) ***;
model pills_consumed= / dist=zinb;
zeromodel;
ods output ParameterEstimates=Parm_GenMod_lap0;
where lap=0; run;

proc genmod data=Col_opioid; *** lap=1, (laparoscopic) ***;
model pills_consumed= / dist=zinb;
zeromodel;
ods output ParameterEstimates=Parm_GenMod_lap1;
where lap=1; run;
```

Proc GenMod output for negative binomial part of ZINB model, lap=0.

Parameter	DF	Estimate	Standard Error	Wald 95%		Wald Chi-Square	Pr > ChiSq
Intercept	1	3.3307	0.0902	3.1539	3.5075	1363.65	<.0001
Dispersion	1	1.6097	0.2309	1.2151	2.1324		

Proc GenMod output for zero part of zero model, lap=0.

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	-1.6673	0.2738	-2.2040	-1.1307	37.08	<.0001

Use ZINB estimation method in SAS Global Forum paper, (Erdman, Jackson, Sinko, 2008).

```
/* Initialize parameters for lap=0 */
data colect_param0;  set Parm_GenMod_lap0;
Dispersion=1.6097;
B0_ZINB=3.3307; /* intercept ZINB model */
B0_ZeroLogistic =-1.6673; /* B0 logistic model for count=0 */
/* Computed parameters */
p_Zerologistic=exp(B0_ZeroLogistic)/(1+exp(B0_ZeroLogistic));
NB_k = 1/Dispersion; /* Number of successes, negative binomial */
```

```
NB_p = 1/(1 + (Dispersion*exp(B0_ZINB)));
/* p=Prob success, negative binomial */
run;

*** First, simulate 1 dataset of N=283, lap=0 ***;
data SimLap0_1rep;   set colect_param0;
Call streaminit(11262);
/* Initialize stream of random number with a seed */

do i=1 to 283;
p_Uniform = Rand('Uniform',0, 1);
if (p_Uniform > p_Zerologistic) then pills_consumed =
    rand('NEGB', NB_p, NB_k);
      else pills_consumed=0;
Simulate=1; lap=0;

/* Simulate other variables in the dataset */
/* Continuous, such as length of stay (los, normal */
/* Binary, such as age group, Bernoulli distribution */
val_los=rand('Normal', 7.76, 5.19); /* unequal lap 1 & 0 */
Age4564=rand('BERNOULLI', .431); /* equal lap 1 & 0 */
Output; end;
```

Compare Pills Consumed, Actual and Simulated, 1 Rep

```
/* Example with Robotic Colectomy (lap=0) */
Data Both_lap0;
set Col_Opioid SimLap0_1rep;
keep lap simulate pills_consumed ConsumeCat3;
where lap=0;  run;
```

Simulate	N	Mean	Std Dev
0	283	23.5	42.0
1	283	23.7	32.8

```
/* Wilcoxon p-value = 0.734 for pills_consumed between simiuated &
actual data */
Proc NPar1Way Wilcoxon Data = Both_lap0;
Class simulate;
Var pills_consumed;
run;
```

For other variables in the dataset, combine simulated and original data. Compare the categorical variables with Proc Freq and the numerical variables with Proc TTest (if normal) or Proc NPar1Way (if non-normal).

6)_Simulating Multiple Reps and Computing Power.

```
/* Simulate 1000 Reps for Robotic and Laparoscopic. */
/* Estimate model parameters fro Proc GenMod for lap=0 */
proc genmod data=Col_opioid;
```

```
model pills_consumed= / dist=zinb;
zeromodel;
ods output ParameterEstimates=Parm_GenMod_lap0;
where lap=0;
run;

/* Process Output from Proc GenMod */
data colect_param0;

/* Initialize parameters for lap=0 */
Dispersion=1.6097;
B0_ZINB=3.3307; /* intercept ZINB model */
B0_ZeroLogistic =-1.6673; /* B0 logistic model for count=0 */

/* Computed parameters */
p_Zerologistic=exp(B0_ZeroLogistic)/(1+exp(B0_ZeroLogistic));
NB_k = 1/Dispersion; /* Number of successes, negative binomial */
NB_p = 1/(1 + (Dispersion*exp(B0_ZINB)));  /* p=Prob success, negative
binomial */
run;

data SimLap0_1000rep; /* Robotic; lap=0 */
set colect_param0;
Call streaminit(11263);  /* Initialize stream of random number with a
seed */

do rep=1 to 1000;
do i=1 to 283;
p_Uniform = Rand('Uniform',0, 1);
  if (p_Uniform > p_Zerologistic) then
  pills_consumed = rand('NEGB', NB_p, NB_k);
  else pills_consumed=0;
pills_prescribed=rand('Normal', 43.46, 45.86);
lap=0;  output;
End; end; run;

ods html path="c:\temp";
proc genmod data=Col_opioid;
model pills_consumed= / dist=zinb;
zeromodel;
ods output ParameterEstimates=Parm_GenMod_lap1;
where lap=1;
run;
ods html close;

*** Repeat same procedure for lap=1 (laparoscopic surgery) ***;
data colect_param1;
/* Initialize parameters for lap=1 */
Dispersion=1.0668;
B0_ZINB=3.0682; /* intercept ZINB model */
B0_ZeroLogistic =-0.8411; /* B0 logistic model for count=0 */
```

```
/* Computed parameters */
p_Zerologistic=exp(B0_ZeroLogistic)/(1+exp(B0_ZeroLogistic));
NB_k = 1/Dispersion; /* Number of successes, negative binomial */
NB_p = 1/(1 + (Dispersion*exp(B0_ZINB)));  /* p=Prob success, negative
binomial */
run;

data SimLap1_1000rep;
set colect_param1;
Call streaminit(20194);  /* Initialize stream of random number with a
seed */

do rep=1 to 1000;
do i=1 to 279;
  p_Uniform = Rand('Uniform',0, 1);
  if (p_Uniform > p_Zerologistic) then
  pills_consumed = rand('NEGB', NB_p, NB_k);
  else pills_consumed=0;
Simulate=1;
lap=1;
pills_prescribed=rand('Normal', 36.4, 23.3); /* unequal lap 1 & 0 */
output;
end; /* end do 1 to 279 */
end; /* end do rep=1 to 1000 */
run;
```

```
/* Combine simulated datasets for lap=0 and lap=1 */
Data SimLap_1000rep;
Set SimLap0_1000rep SimLap1_1000rep;
run;

proc sort data=SimLap_1000rep;
by Rep Lap;
run;

proc datasets library=work;
modify SimLap_1000rep;
index create Rep;
run;
quit;
```

Adding an index to Rep will help the power calculation to run faster. When Proc GenMod (or other SAS procedures) are run with a “by” statement and the variable in the “by” statement is indexed, the procedure will run faster.(Raithel, Creating and Exploiting SAS® Indexes, 2000).

```
/* Combine Simulated Datasets and Run Model */
Proc GenMod data=SimLap_1000Rep;
Model pills_consumed=lap pills_prescribed/dist=zinb;
```

```
Zeromodel lap pills_prescribed;
by Rep;
Ods Output ParameterEstimates=Parm_NB_LapOnly
ZeroParameterEstimates=Parm_ZeroModel_LapOnly;
Run;

data Parm_NB_LapOnly;
set Parm_NB_LapOnly(keep=Rep Parameter Estimate ProbChiSq);
Where parameter='lap';
if ProbChiSq<.05 then lapSignificant=1;
if ProbChiSq>=.05 then lapSignificant=0;
run;

Data Parm_ZeroModel_LapOnly;
set  Parm_ZeroModel_LapOnly(keep=Rep Parameter Estimate ProbChiSq);
Where parameter='lap';
if ProbChiSq<.05 then lapSignificant=1;
if ProbChiSq>=.05 then lapSignificant=0;
run;

/* Negative Binomial Portion of Model */
ods htm path="c:\temp";
proc Freq Data=Parm_NB_LapOnly;
tables lapSignificant;
run;
ods html;

/* Zero Model */
ods htm path="c:\temp";
proc Freq Data=Parm_ZeroModel_LapOnly;
tables lapSignificant;
run;
ods html;
```

- Only the negative binomial portion of the model was of interest to the doctors. Based on this simulation the power was 78% to detect a significant difference between laparoscopic and robotic colectectomies, when the only other covariate was pills_prescribed. Because 78% was close to the 80% common threshold for power, 78% was satisfactory to the doctors. The 78% power was based on the table of lapSignificant from SAS Proc Freq.
- To investigate the effect on power by increasing the sample size to 500 or larger, change the N's of 279 and 283 to 500 or larger, and re-run Procs GenMod and Freq to get the resulting power.

7)_Bootstrapping.

*** LEFT OFF HERE ***

Conclusions.

- ✓ Monte Carlo simulation is an effective method of adjusting for multiple comparisons and is available in SAS for comparisons of percentages and means, the Wilcoxon and Kruskal-Wallis tests, linear and logistic regression, and mixed models.
- ✓ When comparing percentages, use PROC FREQ with the Exact statement and the MC option for Monte Carlo simulation.
- ✓ For the T-Test, use PROC MULTTEST with the “test mean” option and remember the “plots=PByTest” option on the PROC MULTTEST statement.
- ✓ For linear regression, use PROC GLM and then PROC PLM for post-processing of estimates.
- ✓ For linear mixed models, use PROC MIXED, followed by PROC PLM.
- ✓ For generalized linear mixed models with binary or categorical outcomes, PROC GLIMMIX already has an “Adjust=Simulate” option on the estimate statement.

REFERENCES

1. Herek GM. Documenting prejudice against lesbians and gay men on campus: The yale sexual orientation survey. *Journal of Homosexuality*. 1993.

2. Sue DW. *Microaggressions in everyday life: Race, gender, and sexual orientation.* NJ: John Wiley & Sons; 2010.

3. Sawilowsky SS. Target article: You think you've got trivials? *Journal of modern applied statistical methods*. 2003;2(1):218-225.

4. Efron B. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*. 1979;7(1):1-26.

5. Efron B, Tibshirani R. *An introduction to the bootstrap.* New York: Chapman & Hall; 1993:xvi, 436 p.

6. Smith E, Osborne JW. Bootstrapping confidence intervals for effect sizes (and other weird things). . 2015;Conference on Statistical Practice, New Orleans.

7. Agresti A. A survey of exact inference for contingency tables. *Statistical Science*. 1992;7(1):131-153.

8. Mehta CR, Patel NR. A network algorithm for performing fisher's exact test in r × c contingency tables. *Journal of the American Statistical Association*. 1983;78(382):427-434.

9. Westfall PH, Young SS. *Resampling-based multiple testing: Examples and methods for P-value adjustment.* New York: Wiley; 1993:xvii, 340 p.

10. Mehta CR, Patel NR, Senchaudhuri P. Efficient monte carlo methods for conditional logistic regression. *Journal of the American Statistical Association*. 2000;95(449):99-108.

CONTACT INFORMATION

Your comments and questions are welcome.

Brandy R. Sinco, Statistician Senior
University of Michigan Medical School
Center for Health Outcomes and Policy (CHOP)
2800 Plymouth Rd., 16-100N-006
Ann Arbor, MI 48109-2800

Phone: 734-232-5244

E-Mail: brsinco@umich.edu