Usage:

Power Calculation for Within-Center Randomization Designs with Correlated Binary Data

Yi LU 11/1/2011

Statement

The statistical program was developed by Yi Lu[1] under supervision of Drs. Mei-Cheng Wang[1] and Larry Wissow[2].

Sampling Design Background

The simulation and the power calculation are based on the sampling design below:

1. Suppose there are "N" clusters (e.g. hospitals/health centers), and within each cluster there are "2m" patients. In each cluster, "m" patients are randomized into the treatment group and the other "m" patients are in the control group.

2. Assume the "N" clusters are independent of each other. **Within** each cluster:

   (a) Suppose there is a binary output for each patient (0/1, e.g. disease/non-disease). Assume that the expected value (of the binary output data) for the control group is "p1" ($0 \le p1 \le 1$), e.g. the incidence of the disease is p, while the expect value for the treatment group is "p2" ($0 \le p2 \le 1$). Note that the expected value for the treatment group is no greater than that for the control group ($\mathbf{p1 \ge p2}$).
   (b) The 2m patients (no matter group they are in) are correlated, and the correlation (coefficient) between any two patients is equal to "r" ($0 \le r \le 1$).

Subject to treatment-effect-size (i.e., treatment difference) "$d = p1 - p2$" (e.g. the expected decrease in the incidence of the disease), which is determined by p1 and p2, the type I error "α" and correlation "r", the power is to be calculated for the design above:

$$\mathbf{Power(N, m, p1, p2, r, \alpha) = Pr(\hat{d} \in R | d),}$$

where $d \neq 0$, $\hat{d}$ is the difference of the mean value of the $N \times m$ outputs from the patients of treatment groups and the mean value of the other $N \times m$ outputs from those in control groups, R is the rejection area determined by $\alpha = Pr(\hat{d} \in R | d = 0)$.

[1] Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health
[2] Department of Health, Behavior and Society, Johns Hopkins Bloomberg School of Public Health

[Remark]: The power calculation in R involves complicated simulations with the sample size as a necessary input parameter. Therefore, to estimate minimum sample size given a fixed power, it's suggested that the program be applied repeatedly with different combination of parameter settings (e.g. different sample sizes) so that the minimum sample size required could be approximated.

The .R Function: "power_binary".

Based on the sampling design above, we have the following parameters to set:

N	Number of clusters.
m	Number of subjects (patients) in each cluster (within each group: treatment/control)
p1	The incidence probability for group 1 (expected value for control groups).
p2	The incidence probability for group 2 (expected value for treatment groups).
r	The correlation between subjects (patients) in the same cluster.
alpha	Type I error (default value = 0.05).
Rep	Number of repeated sampling in simulation (default value = 1000).

We have the requirements for the parameter setting above:

1. $\mathbf{0 \le p1, p2, r \le 1}$
2. $\mathbf{p1 \ge p2}$
3. $\mathbf{0 \le \frac{r \times (1-p2)}{\frac{p2}{p1} \times (1-p1)} \le 1}$

Note that the above requirements are necessary due to the natural structure of the discrete data model (i.e. correlated binary data). If these requirements are not satisfied, R will report errors without power calculations conducted. For technique details involving the requirements above in simulating correlated binary data, see Appendix.

We may follow the example below to illustrate the usage of the R function "power_binary".

[Example 1]: Suppose we have the "power_binary.R" file downloaded at the directory: C:/Users/Documents. (Note that this directory can be changed up to your choice. If you do so, please make corresponding change in the "source" command below.)

Step 1: Download and install R.
Step 2: Open R. In the "R Console" window, type the following command:

```
source("C:/Users/Documents/power_binary.R")
```

Note that the directory part above is subject to change if you save the "power_binary.R" file

at other directory. Each time you open R software, you may have to type the "source" command above to load the "power_binary.R" function in your R working space.

Step 3: Use the power_binary.R function to calculate power by setting the parameters in the following command (the 1st line below):

```
power_binary(N=10,m=25,p1=0.8,p2=0.8,r=0.04)
The power is:
[1] 0.057[3]
```

The 2-3rd lines are the output from R. Note that "alpha" and "Rep" have default values as 0.05 and 1000, respectively, which can be omitted in the command. If needed, they can be set as:

```
power_binary(N=10,m=25,p1=0.8,p2=0.8,r=0.04,alpha=0.1)
power_binary(N=10,m=25,p1=0.8,p2=0.8,r=0.04,Rep=5000)
power_binary(N=10,m=25,p1=0.8,p2=0.8,r=0.04,alpha=0.01,Rep=500)
```

[Remark]: As noted by the footnote 3, the power calculation is an "estimate" (not the exact theoretical number) based on the simulation whose preciseness could be adjusted by the parameter "Rep" in the function. Generally speaking, by setting "Rep" large could make the estimate more precise and stable (i.e. converge to the true theoretical value). Therefore, it's suggested that the user start with a relatively small number of replications (e.g. the default number Rep = 1000) to see how fast it runs, and then increase the number of replications for stable results.

Moreover, the first 5 parameters are required to be set for the function to run. If you set them as the standard order of (N,m,p1,p2,r), you can omit these "symbols". Otherwise, if you change this order, you have to keep these "symbols". For example, the following commands are equivalent:

```
power_binary(N=10,m=25,p1=0.8,p2=0.6,r=0.04)
power_binary(10,25,0.8,0.6,0.04)
power_binary(r=0.04,p1=0.8,m=25,N=10,p2=0.6)
```

Finally, if your parameter setting misfit the 3 parameter requirements mentioned above, you may obtain the error message like:

```
> power_binary(N=10,m=25,p1=0.6,p2=0.8,r=0.04)
[1] "Errors in parameter setting! 'p2' should be no greater than 'p1'."
The power is:
[1] NaN
```

[3] The calculation of power is through replications of simulation (with the replication number set by the parameter "Rep") and therefore the calculated results **might (should) be different** each time the programs run even with the same parameter settings, although the difference is expected to be small when the number of replications ("Rep") is large.

```
> power_binary(N=10,m=25,p1=1.2,p2=0.8,r=0.04)
[1] "Errors in parameter setting! 'p1' should be between 0 and 1."
The power is:
[1] NaN

> power_binary(N=10,m=25,p1=0.2,p2=-1,r=0.04)
[1] "Errors in parameter setting! 'p2' should be between 0 and 1."
The power is:
[1] NaN

> power_binary(N=10,m=25,p1=0.8,p2=0.2,r=1.5)
[1] "Errors in parameter setting! 'r' should be between 0 and 1."
The power is:
[1] NaN

> power_binary(N=10,m=25,p1=0.8,p2=0.08,r=0.04)
[1] "Errors in parameter setting! 'r*(1-p2)/(p2/p1*(1-p1))' should be
between 0 and 1."
The power is:
[1] NaN
```

Appendix: Correlated Binary Data Simulation

To generate correlated binary data for each cluster. The key point is that unequal Bernoulli probabilities exist within each cluster for treatment and control group, respectively. Fortunately, within a fixed cluster, there are only 2 different values of probabilities: one for treatment group, and the other for control group, and the difference of which is the so-called "treatment effect/difference ('d')". Therefore, we may combine the method indicated in Section 2.1 and the "multiplicative factor on the correlation" mentioned in Section 2.4 in the paper by Lunn and Davies (1998) to generate desired binary data with the exchangeable correlation structures within clusters.

To be specific, consider clustered binary values $\{X_{ijk} : 1 \le i \le N, 1 \le j \le m, k=1,2\}$, where "i" for cluster 1 to N, "j" for the j^{th} subject within each cluster for a certain group (treatment/control), and "k" =1 for control; =2 for treatment group. We want X_{ijk} satisfying:

1. $\Pr(X_{ijk}=1) = p_k$ (k = 1, 2. $p1 \ge p2$). Thus, the treatment difference would be $d = p1 - p2$ (e.g. a decrease in probability for obtaining disease $X_{ijk}=1$).
2. $\mathrm{Cor}(X_{ijk}, X_{imk'}) = r \ge 0$ for $j \ne m$.

Denote $p = p1$; $a = {}^{p2}/_{p1}$ $(0 \le a \le 1)$. Require that $0 \le \dfrac{r(1-ap)}{a(1-p)} = \dfrac{r(1-p_2)}{\frac{p_2}{p_1}(1-p_1)} \le 1$. In light of these settings, within a fixed cluster "i", generate the following 5m+1 independent Bernoulli variables:

$Y_{ijk} \sim Bi(1, p)$; j = 1,…,m. k = 1,2.

$U_{ij1} \sim Bi(1, \sqrt{r})$; $U_{ij2} \sim Bi(1, \sqrt{\dfrac{r(1-ap)}{a(1-p)}})$; j = 1,…m.

$A_{ij} \sim Bi(1, a)$; j = 1,…,m.

$Z_i \sim Bi(1, p)$.

Next, define $W_{ijk} = (1-U_{ijk})Y_{ijk} + U_{ijk}Z_i$. Now W_{ijk} are Bernoulli random variables with

$E(W_{ijk}) = p$, and it follows that $Var(W_{ijk}) = p(1-p)$. Moreover, we have

$$E(W_{ijk}W_{imk'}) = r_k r_{k'} p(1-p) + p^2 \quad (j \ne m)$$

$$\text{and} \quad cov(W_{ijk}, W_{imk'}) = r_k r_{k'} p(1-p) \quad (j \ne m),$$

where $r_1 = \sqrt{r}$ and $r_2 = \sqrt{\dfrac{r(1-ap)}{a(1-p)}}$ (note $U_{ijk} \sim Bi(1, r_k)$).

Finally, set $X_{ijk} = \begin{cases} W_{ij1} & k=1 \\ A_{ij}W_{ij2} & k=2 \end{cases}$. Then X_{ijk} are Bernoulli random variables with

$E(X_{ijk}) = \begin{cases} p & k=1 \\ ap & k=2 \end{cases}$, and it follows that $Var(X_{ijk}) = \begin{cases} p(1-p) & k=1 \\ ap(1-ap) & k=2 \end{cases}$. Based on the results above, we have further for $j \neq m$:

$$\text{cov}(X_{ijk}, X_{imk'}) = \begin{cases} r_1^2 p(1-p) = rp(1-p) & \text{if } k = k' = 1 \\ ar_1 r_2 p(1-p) & \\ = r\sqrt{p(1-p)}\sqrt{ap(1-ap)} & \text{if } k = 1; k' = 2 \\ a^2 r_2^2 p(1-p) = rap(1-ap) & \text{if } k = k' = 2 \end{cases}.$$

Therefore, we always have $\text{Cor}(X_{ijk}, X_{imk'}) = r \geq 0$ for $j \neq m$.

Reference

Lunn, A. D. and Davies, S.J., 1998, A note on generating correlated binary variables, *Biometrika* **85**, 487–490.