

サンプルサイズ計算について

一宮西病院呼吸器内科 城下彰宏 (PGY6)

2021/01/11

0.1 総論

サンプルサイズ計算の要素

- **Sample size**

- **Minimum detectable difference** : H_0 と H_A それぞれの statistics の意味のある差。

- **Power** (検出力) : 実際に差があった場合に、「差がある」と判断する確率。 $1 - \beta$ 、つまり、alternative hypothesis (H_A) の下で critical value を越える確率 (rejection region)。

- **One-sided or two-sided**

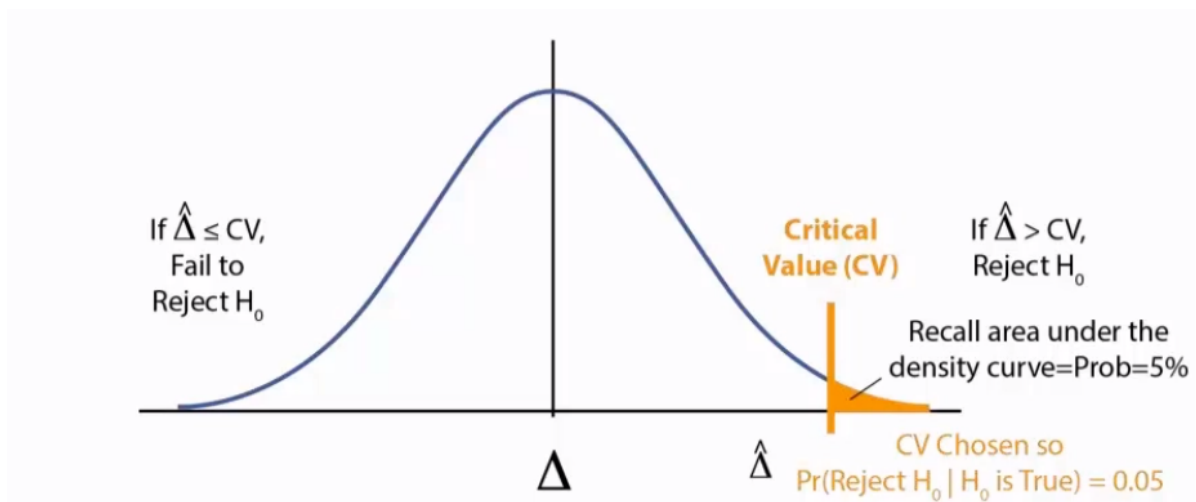
- **Allocation ratio** Target population からの random sampling が study sample であり、Null hypothesis (H_0) が正しいという仮定の下で sample から推定された statistic が「本当の (target population の) statistic」と違うのかを検討する。

Critical value よりも大きい場合にそれは起こりうる可能性の低い仮定となり、 H_0 は棄却され、 H_A が採択される。

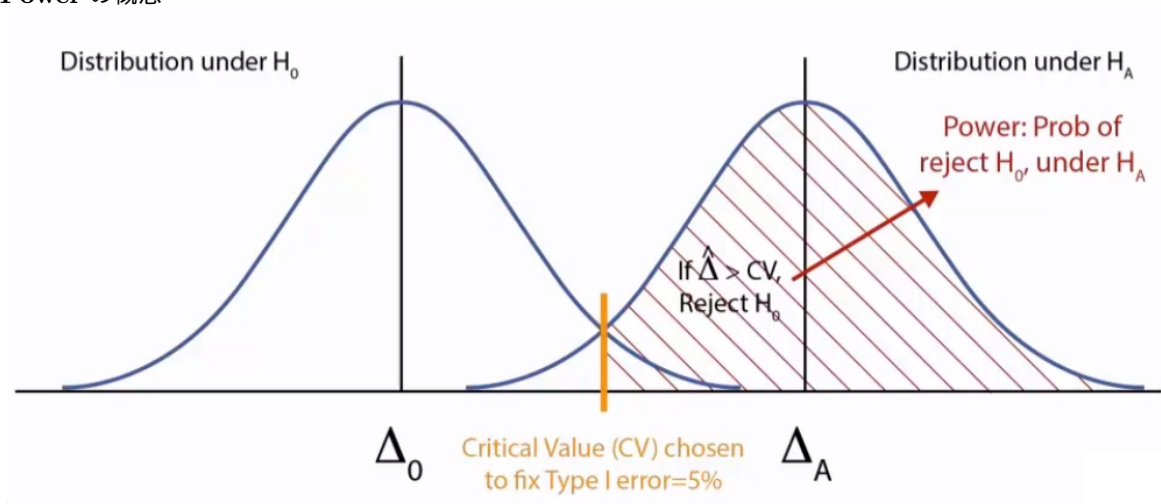
Type I error (α) : H_0 が正しいときに間違って H_0 を棄却してしまう確率

Type II error (β) : H_A が正しいときに間違って H_0 を棄却し損なう確率
仮説検定の場合は通常 Type I error のみを考慮しているため、サンプルサイズ計算が特に重要ということになる。それぞれの概念を図式化して確認してみる。

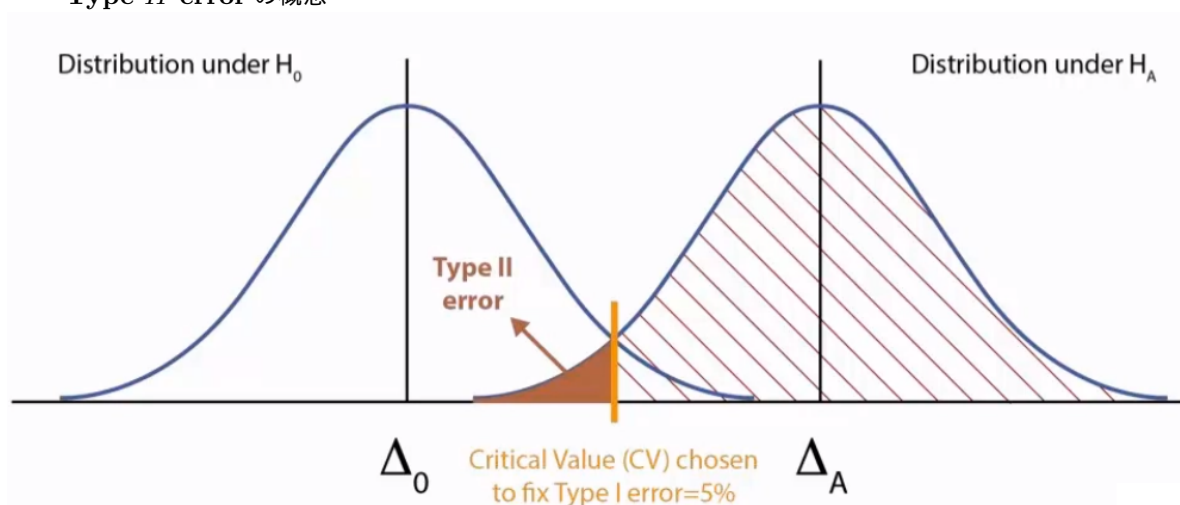
Type I error の概念



Power の概念



Type II error の概念



R では単純なサンプルサイズ計算は pwr package で行うことができる。

0.2 One sample mean (cross-sectional aspect)

$H_0 : \mu = \mu_0, H_A : \mu > \mu_1$ となる。Sample standard deviation = $s = \sqrt{\frac{1}{N-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ であり、population standard deviation = $\sigma = \frac{s}{\sqrt{N}}$ である。Statistics は $T = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$ (T は σ が既知の場合は正規分布に、 σ が s によって推定される場合は t 分布に従う) であるため、 $T > Z_\alpha$ (two-sided の場合は $|T| \geq Z_{\alpha/2}$) の場合に H_0 が棄却される。実際は $\alpha=0.05$ であり、one-sided の場合は $Z_\alpha = 1.64$ である。

```
library(pwr)
#TO CALCULATE SAMPLE SIZE
power.t.test(delta=3, sd=1, power=0.8, type="one.sample", alternative="one.sided")
```

```
##
##      One-sample t test power calculation
##
##              n = 2.552324
##              delta = 3
##              sd = 1
##              sig.level = 0.05
##              power = 0.8
##      alternative = one.sided
```

```
#TO CALCULATE DELTA
power.t.test(n=50, sd=1, power=0.8, type="one.sample", alternative="one.sided")
```

```
##
##      One-sample t test power calculation
##
##              n = 50
##              delta = 0.3566013
##              sd = 1
##              sig.level = 0.05
##              power = 0.8
##      alternative = one.sided
```

```
#TO CALCULATE POWER
power.t.test(n=50, delta=3, sd=1, type="one.sample", alternative="one.sided")
```

```
##
##      One-sample t test power calculation
```

```
##
##           n = 50
##         delta = 3
##           sd = 1
##       sig.level = 0.05
##           power = 1
##   alternative = one.sided
```

Package を使用しない方法も参考までに記載しておく。

```
#TO CALCULATE POWER
spone <- function(pnull,palt,n,alpha,sss)
{
  qnull <- 1-pnull
  qalt <- 1-palt
  sqv0 <- sqrt((pnull*qnull)/n)
  sqv1 <- sqrt((palt*qalt)/n)
  zalp <- qnorm(1-alpha/sss)
  int1 <- (palt - pnull - zalp*sqv0)/(sqv1)
  pnorm(int1)
}

##RO CALCULATE SAMPLE SIZE
ssone <- function(pnull,palt,pow,alpha,sss) {
  qnull <- 1-pnull
  qalt <- 1-palt
  sqv0 <- sqrt((pnull*qnull))
  sqv1 <- sqrt((palt*qalt))
  zalpha <- qnorm(1-alpha/sss)
  zbeta <- qnorm(pow)
  numerator <- zalpha*sqv0 + zbeta*sqv1
  denom <- palt - pnull
  (numerator/denom)**2
}
```

0.3 Two sample mean (cross-secrional aspect)

Population 1 と Population 2 からそれぞれの sampling population を考える。

$T = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{\bar{x}_1 - \bar{x}_2}}$ であるが、2 つの population の分散が等しく既知の場合と等しく未知の場合、等しくなく未知の場合に分かれる (数式は割愛)。考え方は one sample mean と同様である。

ちなみに paired two sample comparison の場合は one sample case と同様に扱うことができる。
以下は 2 つの population の分散が等しい場合の計算である。

```
#TO CALCULATE POWER  
d<-1/1.4  
pwr.t.test(d=d, n=30, sig.level=0.05, type="two.sample", alternative="two.sided")
```

```
##  
##      Two-sample t test power calculation  
##  
##              n = 30  
##              d = 0.7142857  
##      sig.level = 0.05  
##      power = 0.7764889  
##      alternative = two.sided  
##  
## NOTE: n is number in *each* group
```

```
#n is the total sample size. assume equal allocation between the two groups.  
#d is the standardized effect size. i.e. (mu1-mu0)/SD
```

```
#TO CALCULATE SAMPLE SIZE  
pwr.t.test(d=d, power=0.8, sig.level=0.05, type="two.sample", alternative="two.sided")
```

```
##  
##      Two-sample t test power calculation  
##  
##              n = 31.75708  
##              d = 0.7142857  
##      sig.level = 0.05  
##      power = 0.8  
##      alternative = two.sided  
##  
## NOTE: n is number in *each* group
```

Package を使用しない方法も参考までに記載しておく。

```
#CALCULATE POWER  
sptwo <- function(p1,p2,n1,n2,alpha) {  
pbar <- (p1+p2)/2
```

```

delta <- abs(p2-p1)
zalpha <- qnorm(1-alpha/2)
intn <- (1/n1) + (1/n2)
int1 <- (delta/sqrt(pbar*(1-pbar)*intn))
pnorm(int1 - zalpha)
}

#CALCULATE SAMPLE SIZE
sstwo <- function(p1,p2,r,alpha,power) {
pbar <- (p1+r*p2)/(r+1)
qbar <- 1-pbar
zalpha <- qnorm(1-alpha/2)
zbeta <- qnorm(power)
int1 <- zalpha*sqrt((r+1)*pbar*qbar)
+zbeta*sqrt(r*p1*(1-p1)+p2*(1-p2))
mprime <- int1**2 / (r*(p2-p1)**2)
# below is continuity correction equation 4.18
ss1 <- (mprime/4)*(1+sqrt(1+(2*(r+1))/
(mprime*r*abs(p1-p2))))**2
ss2 <- ss1*r
list(ss1,ss2)
}

```

2 つの population の分散が未知の等しくない場合も参考までに掲載する。

```

#TO CALCULATE POWER
powerfunc<-function(alpha, mu1, mu2, sigma1, sigma2, n1, n2){
  delta<-mu1-mu2
  sat.df<-(sigma1^2/n1+sigma2^2/n2)^2/((1/(n1-1))*(sigma1^2/n1)^2+(1/(n2-1))*(sigma2^2/n2)^2)
  #Satterwhith df.
  non.cent<-delta/(sqrt(sigma1^2/n1+sigma2^2/n2)) #Non-centrality parameter
  #Two sided alternative
  power<-pt(qt(p=(1-alpha/2),
              df=sat.df, lower.tail=TRUE),df=sat.df, ncp=non.cent, lower.tail=F)+
    pt(qt(p=(alpha/2), df=sat.df, lower.tail=TRUE),df=sat.df, ncp=non.cent, lower.tail=T)
  return(power)
}

powerfunc(alpha=0.05, mu1=5, mu2=3, sigma1=1, sigma2=2, n1=40,n2=50)

```

```
## [1] 0.999982
```

```
#TO CALCULATE SAMPLE SIZE
```

```
n<-seq(50,80,1)
n1<-n
n2<-n
power.result<-c()
for (i in 1:length(n)){
  power.result[i]<-powerfunc(alpha=0.05, mu1=5, mu2=3,
                             sigma1=1, sigma2=2, n1=n1[i],n2=n2[i])
}
power.result
```

```
## [1] 0.9999906 0.9999930 0.9999947 0.9999961 0.9999971 0.9999978 0.9999984
## [8] 0.9999988 0.9999991 0.9999993 0.9999995 0.9999996 0.9999997 0.9999998
## [15] 0.9999998 0.9999999 0.9999999 0.9999999 1.0000000 1.0000000 1.0000000
## [22] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [29] 1.0000000 1.0000000 1.0000000
```

0.4 Correlation coefficient, simple linear regression, and multiple linear regression

Correlation coefficient である r は skewed data なので、Fisher's Z transformation を行わないといけない (r : $F(r) = \frac{1}{2} \ln(\frac{1+r}{1-r})$)。 $F(r)$ は正規分布に従うことが知られている。そこで $T = \frac{F(r) - F(\rho_0)}{\sigma_{F(r)}}$ となり、 $\sigma_{F(r)} = \frac{1}{\sqrt{N-3}}$ からこれまでと同様に検定について考えることができる。

ただし、R では simple linear regression $Y = \beta_0 + \beta_1 X + \epsilon$ 、つまり、 $\hat{\beta}_1 = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{SS_x}$ ($SS_x = \sum_{i=1}^N (x_i - \bar{x})^2$) において、 $H_0 : \beta_1 = 0$ とすることで計算する。 $T = \frac{\hat{\beta}_1 - 0}{\sigma_{\hat{\beta}_1}}$ が F 分布 (df=1, N-2) に従うことから計算するということである。そこで、 SS_x と residual variance が必要になる。

R では multiple linear regression も対応している pwr.f2.test function を使用する。他のパッケージもあるので、色々試してみると良いかもしれない。

```
sd.y<-1
sd.x<-6
b<-0.1
sd.e<-sqrt(sd.y^2-0.1^2*sd.x^2) #This is the standard deviation of residuals.
r.sq<-1-(sd.e^2/sd.y^2) #This is the R_square.
r.sq
```

```
## [1] 0.36
```

```
pwr.f2.test(u=1, f2=r.sq/(1-r.sq), power=0.8)
```

```
##
##      Multiple regression power calculation
##
##          u = 1
##          v = 14.12059
##          f2 = 0.5625
##      sig.level = 0.05
##          power = 0.8
```

```
#u is the DF of numerator,
#since this is simple linear regression testing for one variable, u=1.
#v is left unspecified but power= is specified, so this is requesting output for n.
#v:the DF of the denominator
#v=N-2 -> N=v+2.
```

0.5 One sample proportion (cross-sectional aspect)

Binary outcome つまり、2*2 表の書ける outcome について。Y=Y₁ + Y₂ +Y_n で、Y は二項分布 binomial(n,p) に従う (p=prevalence of characteristics)。ここで p を推定したいので、実際のデータから求めら $\hat{p} = y/n$ であるので、 $H_0 : p = p_0$ を仮定することで検定ができる。サンプルサイズが「十分に」大きくなると sampling distribution は正規分布に近似できることを用いて、 $p\hat{p}$ は平均 \hat{p} で、分散 $n * \hat{p} * p$ の正規分布に従うことになる。T= $\frac{\hat{p}-p_0}{\sqrt{p_0(1-p_0)/n}}$ が statistics になり、これまで通りの検定ができる。ちなみに「十分に」というのは $p_0 * (1 - p_0) * n > 5$ を目安にしている。近似できない (サンプルサイズが小さい) 場合は arc-sin transformation を行なうという選択肢がある。 $2\text{srcsin}(\sqrt{\hat{p}}) \sim \text{Normal}(2\text{arcsin}\sqrt{p}, 1/n)$ を用いて検定する。R の pwr package は arcsine transformation を用いていることに注意が必要である。

```
#TO CALCULATE POWER
h <- ES.h(0.02, 0.03)
pwr.p.test(h=h, n=300, sig.level=0.05, power=NULL, alternative="two.sided")

##
##      proportion power calculation for binomial distribution (arcsine transformation)
##
##          h = 0.06437191
##          n = 300
##      sig.level = 0.05
##          power = 0.2001055
```



```
##      alternative = two.sided
```

```
#To CALCULATE SAMPLE SIZE
```

```
h <- ES.h(0.02, 0.03)
```

```
pwr.p.test(h=h, n=NULL, sig.level=0.05, power=.8, alternative="two.sided")
```

```
##
```

```
##      proportion power calculation for binomial distribution (arcsine transformation)
```

```
##
```

```
##          h = 0.06437191
```

```
##          n = 1894.147
```

```
##      sig.level = 0.05
```

```
##          power = 0.8
```

```
##      alternative = two.sided
```

0.6 Two sample proportion (cross-sectional aspect)

2つの population を想定し、Fisher's exact test で検定する。Cumulative incidence の場合も 2*2 表を作成できるので同様である（ただし、loss to follow-up を考慮しなくてはならない）。

2つの sample size が等しい場面を想定した R のコードを記載した。ここでは両群のサンプルサイズが異なっても使用できる pwr.2p2n.test function を使用したが、pwr.p.test でも計算できる。

```
#TO CALCULATE POWER
```

```
h <- ES.h(0.4, 0.5)
```

```
pwr.2p2n.test(h, 200, 300)
```

```
##
```

```
##      difference of proportion power calculation for binomial distribution (arcsine transformation)
```

```
##
```

```
##          h = 0.2013579
```

```
##          n1 = 200
```

```
##          n2 = 300
```

```
##      sig.level = 0.05
```

```
##          power = 0.5970976
```

```
##      alternative = two.sided
```

```
##
```

```
## NOTE: different sample sizes
```

```
#To CALCULATE SAMPLE SIZE
```

```
h <- ES.h(0.4, 0.5)
```

```
pwr.2p.test(h, power=0.8)
```

```
##
```

```
##      Difference of proportion power calculation for binomial distribution (arcsine transformation
```

```
##
```

```
##              h = 0.2013579
```

```
##              n = 387.1677
```

```
##      sig.level = 0.05
```

```
##      power = 0.8
```

```
##      alternative = two.sided
```

```
##
```

```
## NOTE: same sample sizes
```

```
pwr.2p2n.test(h, n1=1000, n2=NULL, sig.level=0.05, power=0.8, alternative="two.sided")
```

```
##
```

```
##      difference of proportion power calculation for binomial distribution (arcsine transformation
```

```
##
```

```
##              h = 0.2013579
```

```
##              n1 = 1000
```

```
##              n2 = 240.0546
```

```
##      sig.level = 0.05
```

```
##      power = 0.8
```

```
##      alternative = two.sided
```

```
##
```

```
## NOTE: different sample sizes
```

0.7 Two sample difference, ratios, and odds ratios (cross-sectional aspect)

Two sample proportion とほとんど同じである。Statistics が difference は $Z_{\delta} = \frac{\hat{\delta}}{SE(\hat{\delta})}$ 、ratios は $Z_{\rho} = \frac{\ln \hat{\rho}}{SE(\ln \hat{\rho})}$ 、odds-ratio は $Z_{\theta} = \frac{\ln \hat{\theta}}{SE(\ln \hat{\theta})}$ であるのが違いとなる。Statistics を正規分布に近似していることも proportion の場合と同じである。

0.8 Incidence rate (longitudinal aspect)

Incidence rate (mortality) は at-risk time を考慮しないといけない ($I=y/T$ =Total number of events/Total person-time at risk)。まず、仮定として、ある時点で event の起こる可能性はどの患者でも等しいとする。Y はポアソン分布に従い ($\text{Poisson}(\lambda, T)$ に従う)、T は研究デザインから固定されているとすると、 $y/T = T\lambda/T = \hat{\lambda}$ が incidence rate となる。 $H_0: \lambda = \lambda_0$ 、 $H_1: \lambda \neq \lambda_0$ (two-sided) となる。 H_0 を仮定し、 $T\lambda > 30$ の場合はポアソン分布は正規分布に近似することができる $Y \sim N(T\lambda, T\lambda)$ ことから、分散が既知の場合と同様の検定ができる。ちなみに、 $W = \sqrt{Y}$ とし、 $W \sim N(\sqrt{T\lambda}, 1/4)$ や、Event 数が少ないときは exact methods も使用される。

まずは one sample incidence のコードから掲載する。

```
#TO CALCULATE POWER
rate.power.one <- function(rnull,ralt,ttt,alpha,sss) {
  xnull <-sqrt(rnull*ttt) #rnull: null rate, ttt:person time
  xalt <- sqrt(ralt*ttt) #anull: alternative rate
  int1 <-(xnull-xalt)/sqrt(0.25)
  zalp <- qnorm(1-alpha/sss) #sss=1, 2 sided
  int2 <- 1-pnorm(zalp+int1)
  int2}

#TO CALCULATE SAMPLE SIZE
rate.ss.one <- function(rnull,ralt,pow,alpha,sss) {
  xnull <-sqrt(rnull)
  xalt <- sqrt(ralt)
  zalp <- qnorm(1-alpha/sss)
  zbeta <- qnorm(pow)
  numerator <- (zalp + zbeta)
  denom <- xalt - xnull
  final <- 0.25 * (numerator/denom)**2
  final
}
```

次に two sample incidence の比較についてのコードを記載する。Incidence rate ratio や incidence rate difference を statistics とし、 $H_0: \text{IRR}=1$, $\text{IRD}=0$ とすることで、それぞれ正規分布に従う $\ln(\text{IRR}) = \ln(\frac{Y_A}{T_A}/\frac{Y_B}{T_B})$ 、 $\sqrt{Y_A/T_A} - \sqrt{Y_B/T_B} = W_A - W_B$ から検定を行う。。IRR と IRD のどちらも使用しても大差はないだろう。

```
#TO CALCULATE POWER
rate.power.twosqrt <- function(r1,r2,n1,n2,alpha,sss) {
  num <- sqrt(r1) - sqrt(r2)
```

```

den <- .5 * sqrt(1/n1 + 1/n2)
zalpha <- qnorm(1 - alpha/sss)
int1 <- num / den
pnorm(int1 - zalpha)
}

#TO CALCULATE SAMPLE SIZE
rate.ss.twosqrt <- function(r1,r2,r,alpha,power,sss) {
  #r = ratio of sample sizes t2/t1
  term1 <- 1+r
  zalpha <- qnorm(1 - alpha/sss)
  zbeta <- qnorm(power)
  term2 <- 0.25 * (zalpha+zbeta)**2
  term3 <- (sqrt(r1) - sqrt(r2))**2
  ss1 <- term1 * term2 / term3
  ss2 <- ss1*r
  list(ss1,ss2)
}

```

0.9 Continuous outcome (longitudinal aspect)

Continuous outcome の 2 時点の変化について paired t-test (one sample の場合) や変化量についての two-sample t-test (two sample の場合) を行うことを想定する。注意点は変化量についての two-sample t-test において SD は変化量の SD であり、 $\sigma_{change}^2 = \sigma_{time1}^2 + \sigma_{time2}^2 - 2\rho\sigma_{time1}\sigma_{time2}$ つまり、それぞれの時点の outcome の SD だけではなく、時点同士の SD の correlation も指定しないといけない。

```

#TO CALCULATE SAMPLE SIZE (one sample)
pwr.t.test(d=(0-5)/5, power=0.8, sig.level=0.05, type="paired", alternative="two.sided")

##
##      Paired t test power calculation
##
##              n = 9.93785
##              d = 1
##      sig.level = 0.05
##      power = 0.8
##      alternative = two.sided
##
## NOTE: n is number of *pairs*

```

```

#TO CALCULATE SAMPLE SIZE (two sample)
cohensD<-(6-5)/sqrt(6^2+5^2/2)
#cohensD <- (mean(aM)-mean(aP))/(sqrt((sd(aM)^2+sd(aP)^2)/2))
pwr.t.test(d = cohensD, power=0.8, type = "paired", alternative = "two.sided", sig.level= 0.05)

##
##      Paired t test power calculation
##
##              n = 382.5949
##              d = 0.1435916
##      sig.level = 0.05
##      power = 0.8
##      alternative = two.sided
##
## NOTE: n is number of *pairs*

```

0.10 Binary outcome (longitudinal aspect)

Binary outcome の 2 時点の変化について、よくある状況は同じ群で prevalence を 2 時点で変化を比較するときや関連のある二群で outcome の変化の比較を行うときなどである。McNemar's test が使用される。McNemar's test は例えば STD についてある sampling patients で付き合いのある男女のそれぞれの疾患割合を比較する場合を例にしてみる。2*2 表を 4 つの couple である N11(P11), N10(P10), N01(P01), N00(P00) とし、discordant couple の N01(P01) と N00(P00) について $P_M - P_F = P10 - P01$ であることからこれら discordant couple のみに注目できる。ここで、 $H_0: P_F = P_M$ ($P_F = P11 + P01$ と $P_M = P11 + P10$) を仮定すると、statistics $T = \frac{(N_{10} - N_{01})^2}{(N_{10} + N_{01})}$ は $\chi^2(df = 1)$ に従うことから検定できる。 H_A における sampling distribution は conditional distribution と unconditional distribution の 2 つの仮定がある。R では TrialSize package を使用する。

```

library(TrialSize)
N<-McNemar.Test(alpha=0.05,beta=0.2,psai=0.2/0.5,paid=.7)
#psai: the ratio of p01/p10, paid: the sum p10+p01
N

```

```
## [1] 58.63224
```

0.11 Survival outcomes (longitudinal aspect)

Log-rank test を想定した time-to-event outcome のサンプルサイズ計算については、censoring について（特に right censoring）について考慮しないといけない。ここでは Schoenfeld method を紹介する。Survival function $S(t)=\Pr(T>t)=1-\Pr(T\leq t)$ は nonparametric method の Kaplan-Meier curve で推定できる。一方、hazard function は $\lambda(t) = P(T = t|T \geq t) = \frac{P(T=t)}{P(T\geq t)}$ (T:discrete)、つまり instantaneous failure rate (t まで生存した患者が t で event を発生する risk) を表す。Log-rank test は $H_0: \lambda_A(t) = \lambda_B(t)$ (または $S_A(t) = S_b(t)$)、 $H_A: \lambda_A(t) \neq \lambda_B(t)$ (two-sided) と仮定する検定であり、それぞれの failure time で 2*2 表を作成してまとめるイメージである。そこで、 $H_0: \theta = 0$ ($\theta = \log(\frac{\lambda_1}{\lambda_0})$) と変換することができ (hazard ratio)、statistics を $\chi^2(df = 1)$ を考えることができる。ここから、必要な event 数は $m = \frac{Z_{\alpha/2} + Z_{\beta}}{\theta^2 \pi(1-\pi)} (\pi \text{ allocation proportion to first group event})$ $\$$: enrollment period、f: フォロアップ期間、proportional hazards model を仮定したもとの survival function の情報を合わせて計算する。それぞれの群の survival function から平均を算出して ($\bar{S}(t) = \pi * \bar{S}_0(t) + (1-\pi) * \bar{S}_1(t)$)、試験終了までに event の起こる確率を計算する ($\bar{F}(a+f) = 1 - 1/6[\bar{S}(f) + 4\bar{S}(0.5a+f) + \bar{S}(a+f)]$)。最終的に $N = \frac{m}{\bar{F}(a+f)}$ からサンプルサイズが求められる。R では Hmisc package を使用する。

```
#TO CALCULATE POWER
library(Hmisc)
tref<-2.5      #Specify a time about the survival probability
n<-4000       #total sample size
mc<-0.45      #Probability of having a event by tref, i.e. 1-Survial prob at tref.
hr<-1.13      #Hazard ratio.
r<-(1-((1-(1-mc)^hr)/mc))*100 #Reduction of risk
#1- the ratio of event probability in intervention group compared to control group
#at tref. Can be specified directly without useing
accrual<-2     #Length of accrual period
tmin<-3        #Minimum length of follow-up
pwr.res<-cpower(tref=tref, n=n, mc=mc, r=r, accrual=accrual, tmin=tmin)

##
## Accrual duration: 2 y  Minimum follow-up: 3 y
##
## Total sample size: 4000
##
## Alpha= 0.05
##
## 2.5-year Mortalities
##      Control Intervention
```

```
##      0.4500000      0.4911265
##
## Hazard Rates
##      Control Intervention
##      0.2391348      0.2702223
##
## Probabilities of an Event During Study
##      Control Intervention
##      0.6121072      0.6565620
##
## Expected Number of Events
##      Control Intervention
##      1224.2      1313.1
##
## Hazard ratio: 1.13
## Standard deviation of log hazard ratio: 0.03972899
```

```
pwr.res
```

```
##      Power
## 0.8678575
```

```
#TO CALCULATE SAMPLESIZE
Nsim<-seq(1000,3000, 100)
pwr.res<-c()
for (i in 1:length(Nsim) ){
  pwr.tmp<-cpower(tref=tref, n=Nsim[i], mc=mc, r=r, accrual=accrual, tmin=tmin,pr=FALSE)
  pwr.res[i]<-pwr.tmp["Power"]
}
pwr.res.all<-cbind(Nsim, pwr.res)
pwr.res.all
```

```
##      Nsim  pwr.res
## [1,] 1000 0.3368116
## [2,] 1100 0.3645668
## [3,] 1200 0.3917863
## [4,] 1300 0.4184145
## [5,] 1400 0.4444043
## [6,] 1500 0.4697172
## [7,] 1600 0.4943217
```

```
## [8,] 1700 0.5181931
## [9,] 1800 0.5413128
## [10,] 1900 0.5636678
## [11,] 2000 0.5852497
## [12,] 2100 0.6060549
## [13,] 2200 0.6260835
## [14,] 2300 0.6453390
## [15,] 2400 0.6638283
## [16,] 2500 0.6815605
## [17,] 2600 0.6985475
## [18,] 2700 0.7148029
## [19,] 2800 0.7303421
## [20,] 2900 0.7451818
## [21,] 3000 0.7593401
```

0.12 Simulation

複雑な解析では test statistics が未知であったりモデルの仮定を満たさない場合がある。その際はサンプルサイズ計算はシミュレーションで行わなくてはならない。

流れとしては H_A のもとで sample size を定めて random sampling を行い、検定を行う。これを複数回 B 繰り返し、 $\text{power} = (\text{Number of p-value} < 0.05) / B$ を計算する。Power < 0.8 であれば sample size を増やしてみる、という流れで行う。

まずは one sample mean についてシミュレーションしてみる。まずは H_1 が指数分布に従う場合についてみる。

```
#TO CALCULATE POWER
p.res<-c() #This is to save the p-values from each simulation
B<-1000 #No. of simulations
alpha<-0.05
mu0<-5.5
mu1<-6.0
sd<-1.4
N<-30 #Sample size
for (i in 1:B){
  set.seed(12300+123*i) #Setting the seed number for random number generation, this is to make
  sim.data<-rexp(n=N, rate=1/mu1)
  p.res[i]<-t.test(x=sim.data, alternative = c( "greater"), mu = mu0, conf.level = 1-alpha)$"p.v
}
tmp<-ifelse(p.res<alpha,1,0)
```



```
power.res<-sum(tmp)/B
power.res
```

```
## [1] 0.062
```

次に H_1 が正規分布に従う場合のサンプルサイズ計算についてみる。サンプルサイズ計算を網羅的に行う場合はこちらのほうがよりスムーズに行えるだろう。

```
#TO CALCULATE SAMPLE SIZE
B.sim<-1000 #No. of simulations
alpha.sim<-0.05
mu0.sim<-5.5
mu1.sim<-6.0
sd.sim<-1.4
targ.power.sim<-0.8
N.sim<-30:60 #Sample size to try.
pwr.func<-function(N, B=B.sim, alpha=alpha.sim,
                    mu0=mu0.sim, mu1=mu1.sim, sd=sd.sim)
{
  p.res<-c() #This is to save the p-values from each simulation
  for (j in 1:B){
    set.seed(12300+123*N*j)
    sim.data<-rnorm(n=N, mean=mu1, sd=sd)
    p.res[j]<-t.test(x=sim.data, alternative = c( "greater"),
                    mu = mu0, conf.level = 1-alpha)$"p.value"
  }
  tmp<-ifelse(p.res<alpha,1,0)
  power.all<-sum(tmp)/B
  return(power.all)
}
res.pwr<-c()
for (i in 1:length(N.sim)){
  res.pwr[i]<-pwr.func(N=N.sim[i])
}

res.pwr1<-cbind(N.sim, res.pwr)
res.pwr1
```

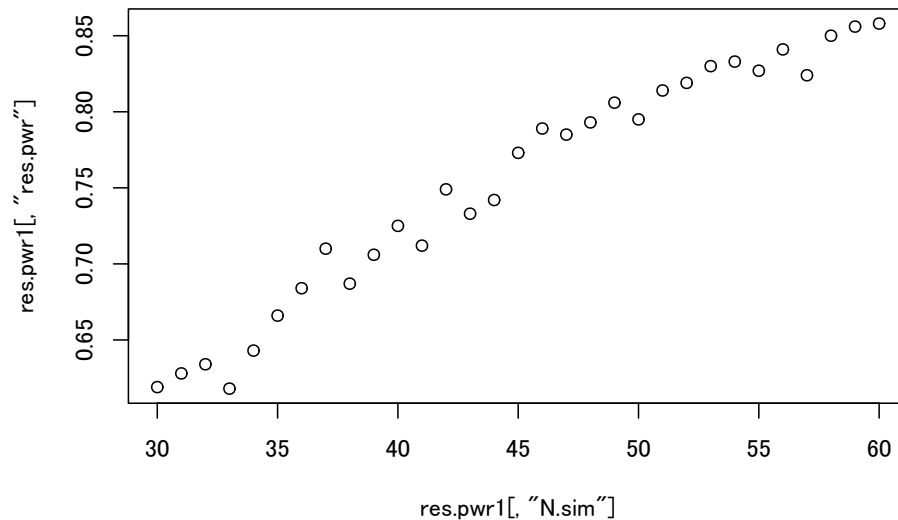
```
##      N.sim res.pwr
```

```
## [1,] 30 0.619
## [2,] 31 0.628
## [3,] 32 0.634
## [4,] 33 0.618
## [5,] 34 0.643
## [6,] 35 0.666
## [7,] 36 0.684
## [8,] 37 0.710
## [9,] 38 0.687
## [10,] 39 0.706
## [11,] 40 0.725
## [12,] 41 0.712
## [13,] 42 0.749
## [14,] 43 0.733
## [15,] 44 0.742
## [16,] 45 0.773
## [17,] 46 0.789
## [18,] 47 0.785
## [19,] 48 0.793
## [20,] 49 0.806
## [21,] 50 0.795
## [22,] 51 0.814
## [23,] 52 0.819
## [24,] 53 0.830
## [25,] 54 0.833
## [26,] 55 0.827
## [27,] 56 0.841
## [28,] 57 0.824
## [29,] 58 0.850
## [30,] 59 0.856
## [31,] 60 0.858
```

```
tmp<-ifelse((res.pwr1[, "res.pwr"]-targ.power.sim)>0,
            (res.pwr1[, "res.pwr"]-targ.power.sim),100)
#This is to identify the N with power closest to target power.
N.need.pwr1<-res.pwr1[which(tmp==min(tmp)),]
N.need.pwr1
```

```
## N.sim res.pwr
## 49.000 0.806
```

```
plot(x=res.pwr1[, "N.sim"], y=res.pwr1[, "res.pwr"]) #Power curve
```



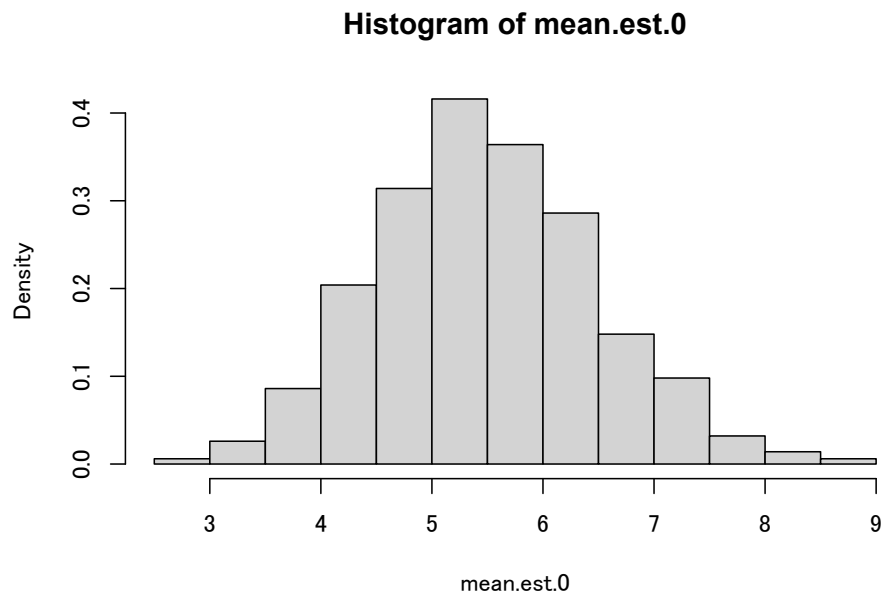
さらに、 H_0 と H_A のどちらの distribution も未知の場合について考えてみる。この場合は、parameter と test statistics を仮定し、 H_0 のもとで random sampling を行い、それぞれの sampling で statistics を計算する。それぞれの sampling から計算された statistics の distribution をみて、percentile から critical value を定める (central limit theorem)。 H_1 のもとで random sampling を来ない、 H_0 が棄却されるかを検討する、という流れで行う。

```
p.res<-c()
#This is to save the p-values from each simulation
B<-1000 #No. of simulations
alpha<-0.05
mu0<-5.5
mu1<-6.0
sd<-1.4
N<-30 #Sample size
#FIRST GENERATE DATA UNDER H0
#TO DETERMINE CV AND THUS THE REJECTION REGION
mean.est.0<-c()
for (i in 1:B){
  set.seed(12300+123*i)
  #Setting the seed number for random number generation
  #this is to make sure the result will be reproducible.
```

```

sim.data<-rexp(n=N, rate=1/mu0)
mean.est.0[i]<-mean(sim.data)
}
hist(mean.est.0, freq =FALSE)

```



```

crt.value<-quantile(mean.est.0, probs=1-alpha)
#GENERATE DATA UNDER H1.
mean.est.1<-c()
for (i in 1:B){
  set.seed(568+1234*i)
  sim.data<-rexp(n=N, rate=1/mu1)
  mean.est.1[i]<-mean(sim.data)
}
pwr.est<-sum(ifelse(mean.est.1>=crt.value, 1,0))/B
pwr.est

```

```
## [1] 0.158
```

最後に random effects model についてのシミュレーションを行っておく。 $y = u + B_0 + B_1x + \epsilon$ (random intercept : $u \sim N(0, \sigma_u^2)$, residual : $\epsilon \sim N(0, \sigma_\epsilon^2)$) (y : continuous variable)。 $\beta_1=0.35$ 、一つ上の階層の correlation on $y = 0.4$ 、 y の SD = 0.16 と仮定し、

```

#TO CALCULATE POWER
library(MASS)
library(nlme)
#####Initialize parameters
alpha<-0.05
Bsim<-500
effect<-0.1
corr<-0.4
beta0<-0.6
sd<-0.40 #For y
Nsim<-300 #Sample size (i.e total no. of clusters).
fam.dist<-c(1/3, 1/2, 1/6)
#Probability of a cluster having 1, 2, or 3 patients
#carrying the risk, respectively.
rmat<-matrix(c(1, corr, corr, corr, 1, corr, corr, corr, 1), nrow=3, ncol=3, byrow=TRUE)
#correlation coefficient matrix
p.value<-c() #Initialize p.value
for (i in 1:Bsim) {
  set.seed<-12300+i*11+Nsim*10
  n.fam<-rmultinom(n=1, size=Nsim, prob=fam.dist)
  #Numbers of families with 1, 2, or 3 patients
  #carrying the risk, respectively.
  #Generate X the exposure variable
  x.fam1<-rep(c(1,0,0), n.fam[1])
  #the x vector for clusters with 1 patient carrying the risk.
  id.fam1<-rep(1:n.fam[1], each=3)
  #generate the ids for clusters with 1 patient carrying the risk.
  #every 3 items
  x.fam2<-rep(c(1,1,0), n.fam[2])
  #the x vector for clusters with 2 patients carrying the risk.
  id.fam2<-rep(1:n.fam[2], each=3)+n.fam[1]
  #generate the ids for clusters with 1 patient carrying the risk.
  x.fam3<-rep(c(1,1,1), n.fam[3])
  #the x vector for clusters with 3 patients carrying the risk.
  id.fam3<-rep(1:n.fam[3], each=3)+n.fam[1]+n.fam[2]
  #generate the ids for clusters with 1 patient carrying the risk.
  x.vec<-c(x.fam1, x.fam2, x.fam3)
  id.vec<-c(id.fam1, id.fam2, id.fam3)
  v.mat<-sd*rmat
}

```

```

#For clusters with 1 patient carrying the risk
set.seed<-12300+i*100+Nsim*1
mean.fam1<-beta0+effect*c(1,0,0)
y.fam1<-mvrnorm(n.fam[1], mean.fam1, v.mat)
#For clusters with 2 patients carrying the trait
set.seed<-12300+2*i*100+Nsim*2
mean.fam2<-beta0+effect*c(1,1,0)
y.fam2<-mvrnorm(n.fam[2], mean.fam2, v.mat)
#For clusters with 3 patients carrying the trait
set.seed<-12300+3*i*100+Nsim*3
mean.fam3<-beta0+effect*c(1,1,1)
y.fam3<-mvrnorm(n.fam[3], mean.fam3, v.mat)
###Run the LME
y.vec<-c(c(t(y.fam1)), c(t(y.fam2)), c(t(y.fam3)))
dataset<-data.frame(y.vec, x.vec, id.vec) #Create the dataset
data.formu<-groupedData(y.vec~x.vec|id.vec, data=dataset) #Attacheh the formula to the dat
res<-lme(data.formu, random=~1)
p.value[i]<-summary(res)$tTable["x.vec", "p-value"]
}

pwr.func.res<-sum(ifelse(p.value<alpha,1,0))/Bsim
pwr.func.res

## [1] 0.774

```