

## Power for Log-linear Models with R

Test conditional independence of XY given Z, but true pattern is that X and Y are independent only for for Z = 2. Find sample size so that the power of LR test is 0.80.

```
> A = rbind(c(100,50),
+           c(50,100))
> B = rbind(c(60,60),
+           c(40,40))
> truth1 = numeric(8)
> dim(truth1) = c(2,2,2)
> truth1[, ,1] = A
> truth1[, ,2] = B
> truth1
, , 1
     [,1] [,2]
[1,] 100  50
[2,]  50 100
, , 2
     [,1] [,2]
[1,]  60  60
[2,]  40  40
> n0 = sum(truth1); n0
[1] 500
> # Effect is departure from (XZ)(YZ)
> mod1.1 = loglin(truth1,margin=list(c(1,3),c(2,3))); mod1.1
2 iterations: deviation 0
$lrt
[1] 33.97981
$pearson
[1] 33.33333
$df
[1] 2
```

```
$margin
$margin[[1]]
[1] 1 3
```

```
$margin[[2]]
[1] 2 3
```

```
> G2.1 = mod1.1$lrt
> effectsize1 = G2.1/n0
> wantpow = 0.80; power = 0 ; n = 40; crit = qchisq(0.95,2)
> while(power<wantpow)
+   {
+     n = n+1
+     lambda = n*effectsize1
+     power = 1-pchisq(crit,2,lambda)
+   } # End while power < wantpow
> n; power
[1] 142
[1] 0.8006787
>
> # Interesting to look at the truth1 table from different angles
> margin.table(truth1,c(1,2))
      [,1] [,2]
[1,]  160  110
[2,]   90  140
> margin.table(truth1,c(1,3))
      [,1] [,2]
[1,]  150  120
[2,]  150   80
> margin.table(truth1,c(2,3))
      [,1] [,2]
[1,]  150  100
[2,]  150  100
>
```

```

> # (XY)(XZ) is false, but I bet power to detect this is lower.
> mod1.2 = loglin(truth1,margin=list(c(1,2),c(1,3))); mod1.2
2 iterations: deviation 0
$lrt
[1] 13.71005

$pearson
[1] 13.75511

$df
[1] 2

$margin
$margin[[1]]
[1] 1 2

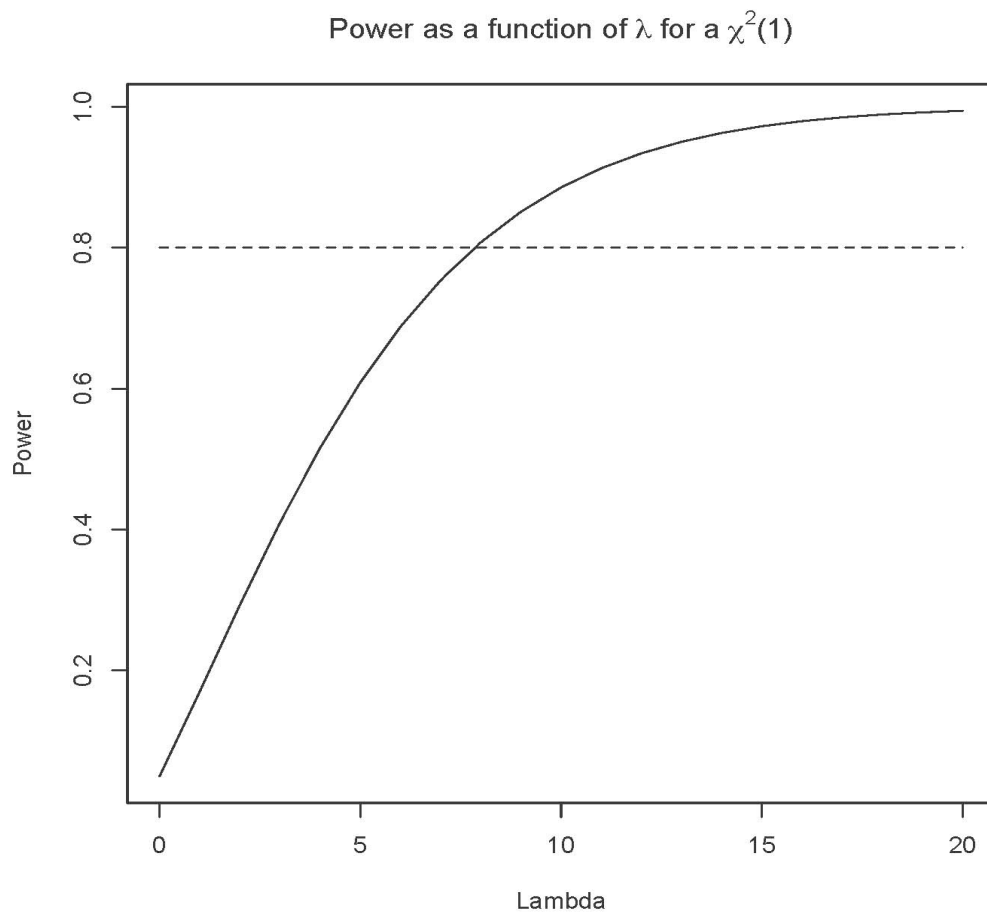
$margin[[2]]
[1] 1 3

> G2.2 = mod1.2$lrt
> effectsize2 = G2.2/n0; lambda = 142*effectsize2
> 1-pchisq(crit,2,lambda)
[1] 0.4056593
> # What sample size is required?
> wantpow = 0.80; power = 0 ; n = 40; crit = qchisq(0.95,2)
> while(power<wantpow)
+   {
+   n = n+1
+   lambda = n*effectsize2
+   power = 1-pchisq(crit,2,lambda)
+   } # End while power < wantpow
> n; power
[1] 352
[1] 0.8007486

```

## A Table of Target Lambda Values

```
> Lambda = 0:20
> Power = numeric(21)
> critval = qchisq(0.95,1)
> for(j in 1:21) Power[j] = 1-pchisq(critval,df=1,ncp=Lambda[j])
> plot(Lambda,Power,type='l')
> lines(c(0,20),c(0.8,0.8),lty=2)
> title(expression(paste("Power as a function of ",lambda," for a
",chi^2,"(1)")))
```



Where does power hit 0.80? Only have to find it once.

```

> # What lambda gives power equal 0.80 for a chisq(1)?
> critval = qchisq(0.95,df=1); critval
[1] 3.841459
> # help(uniroot) Says: "The function uniroot searches the interval from
lower
> #                               to upper for a root (i.e., zero) of the function f
with
> #                               respect to its first argument.
>
> f = function(lambda) # Specific to power of 0.80 for chi-square(1)
+   { f = 1-pchisq(critval,df=1,ncp=lambda) - 0.80; f}
>
> Lambda = uniroot(f,lower=0,upper=20)$root; Lambda
[1] 7.848869
> 1-pchisq(critval,df=1,ncp=Lambda)
[1] 0.8000004

```

Remember this?

```

> # R calculation for the recidivism example
> rm(list=ls())
> crit = qchisq(0.95,df=1); crit
[1] 3.841459
> dummy = rbind(c(40,60),
+              c(30,70))
> X2 = loglin(dummy,margin=list(1,2))$pearson # $
2 iterations: deviation 0
> effectsize = X2/200; effectsize
[1] 0.01098901
> wantpow = 0.80; power = 0 ; n = 50; crit = qchisq(0.95,1)
> while(power<wantpow)
+   {
+     n = n+2 # Keeping equal sample sizes
+     lambda = n*effectsize
+     power = 1-pchisq(crit,1,lambda)
+   } # End while power < wantpow
> n; power
[1] 716
[1] 0.8009609

```

Pearson chi-squared test of independence with  $df = 1$ , want power = 0.80

$$\lambda = n \sum_{i=1}^I \sum_{j=1}^J \frac{(\pi_{ij} - \pi_{i+} \pi_{+j})^2}{\pi_{i+} \pi_{+j}}$$

```
> effectsize = X2/200; effectsize  
[1] 0.01098901
```

```
n * effectsize = 7.848869 => n = 7.848869/effectsize  
> 7.848869/0.01098901  
[1] 714.2471
```

Before we got  $n=116$ , but we were increasing  $n$  by 2 each time to keep sample sizes equal. This is a lot easier. We need a table of these target lambda values. They are not in any book I have seen.

```
> # Make the table of target lambda values  
>  
> df = 1:8  
> power = c(0.50, 0.70, 0.80, 0.90)  
> target = matrix(0,8,4)  
> # More general function: Want root of this.  
> f = function(lambda,DF,WANTPOW) # alpha=0.05  
+   {  
+   critval = qchisq(0.95,df=DF)  
+   f = 1-pchisq(critval,DF,ncp=lambda) - WANTPOW  
+   f  
+   } # End function f  
>  
> for(i in 1:length(df))  
+   {  
+   for(j in 1:length(power))  
+     {  
+     target[i,j] = uniroot(f,lower=0,upper=100,  
+                           DF = df[i], WANTPOW = power[j])$root  
+     } # Next power value  
+   } # Next df value  
>
```

```

> # Label the matrix
> labls = list() # Empty list
> labls$df = as.character(df)
> labls$Power = as.character(power)
> dimnames(target) = labls
> target
  Power
df    0.5    0.7    0.8    0.9
 1 3.841022 6.172007 7.848883 10.50739
 2 4.956736 7.701774 9.634685 12.65394
 3 5.760482 8.792389 10.902570 14.17149
 4 6.419476 9.682473 11.935286 15.40503
 5 6.991270 10.452523 12.827607 16.46946
 6 7.503313 11.140677 13.624286 17.41883
 7 7.971192 11.768443 14.350527 18.28355
 8 8.404641 12.349332 15.022138 19.08270

```

```

> Upper left entry looks suspicious, but it's okay.
> qchisq(0.95,1)
[1] 3.841459
> 1-pchisq(qchisq(0.95,1),df=1,ncp=3.841022)
[1] 0.4999999

```

Example: Testing a proportion  $H_0: \pi=0.50$ , True  $\pi=0.60$ . Want power of 0.90.

$$Z_2 = \frac{\sqrt{n}(p - \pi_0)}{\sqrt{p(1-p)}}$$

$$Z_2^2 \sim \chi^2(1, \lambda), \text{ with } \lambda = n \frac{(\pi - \pi_0)^2}{\pi(1-\pi)}$$

```

> effect = (.6-.5)^2/(.6*.4)
> 10.50739/effect
[1] 252.1774
> # Need n=253

```

Equal numbers of students are to be assigned to one of three teaching methods, and the response variable is whether they pass at the end. Suppose methods A and B result in a true pass rate of 50%, while method C results in a true pass rate of 70%. We plan to test whether training programme is independent of outcome, and want to select the smallest sample size needed for a power of 0.80.

a) We could use either a Pearson or Likelihood Ratio chi-squared test. Which is more powerful at  $\alpha = 0.05$ ? Is it more powerful for ANY sample size?

```
> Training = rbind(c(50,50),
+                 c(50,50),
+                 c(70,30))
> rownames(Training) = c("A","B","C")
> colnames(Training) = c("Pass","Fail")
> Training
  Pass Fail
A   50   50
B   50   50
C   70   30
> # If T is the chi-squared test statistic on the made up data,
> # Then T/300 is the effect size,
> # And ncp is lambda = n * effect size
> cheapway = loglin(Training,margin=list(1,2))
2 iterations: deviation 2.842171e-14
> g2_effsize = cheapway$lrt/300; g2_effsize
[1] 0.03702442
> x2_effsize= cheapway$pearson/300; x2_effsize
[1] 0.0361991
>
```

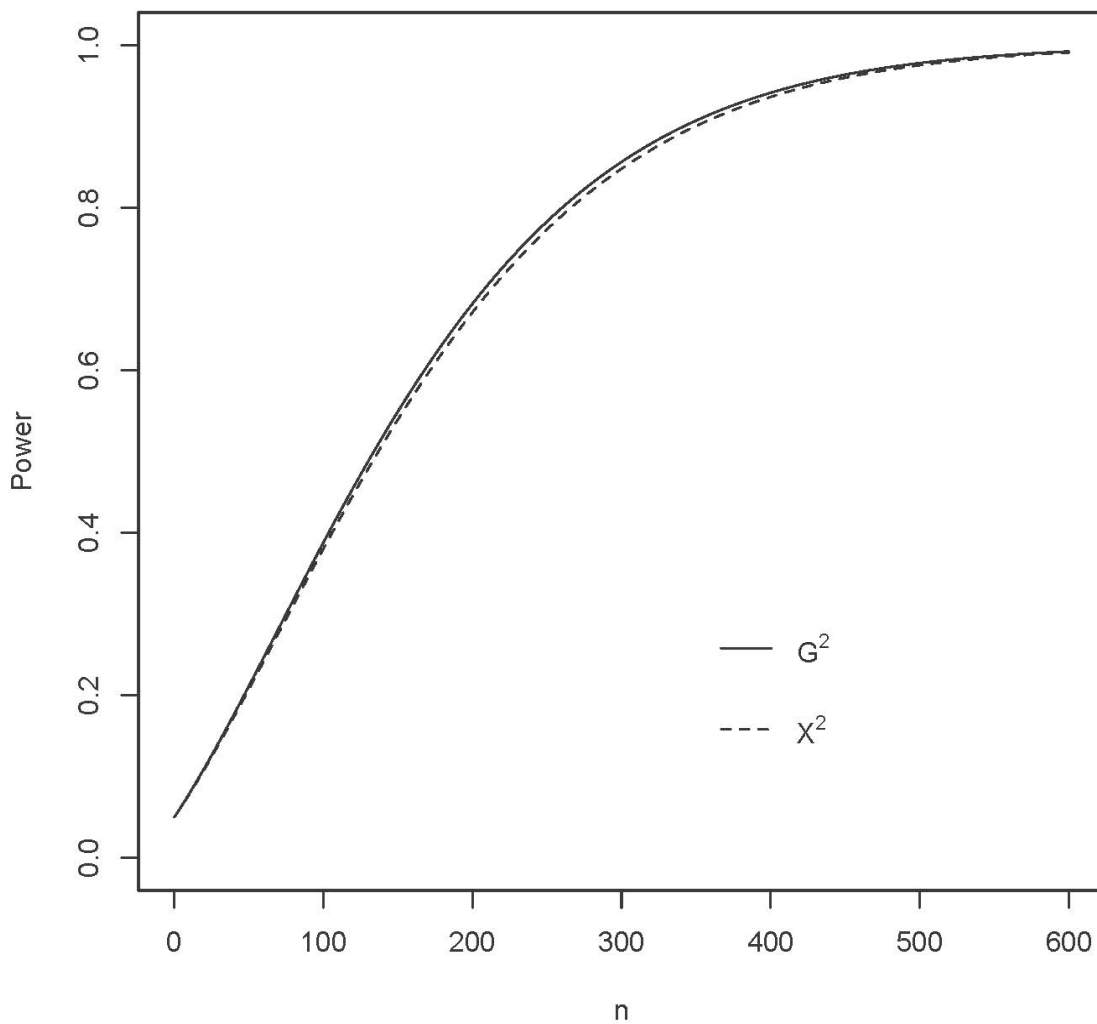


```

> # Plot power as a function of sample size for the two effect sizes
> n = seq(from=0,to=600,by=3)
> g2lambda = n * g2_effsize
> x2lambda = n * x2_effsize
> crit = qchisq(0.95,2)
> Power = 1 - pchisq(crit,df=2,ncp=g2lambda)
> x2pow = 1 - pchisq(crit,df=2,ncp=x2lambda)
> plot(n,Power,type='l',ylim=c(0,1))
> lines(n,x2pow,lty=2)
> legend(x=350,y=0.3,lty=1,bty="n",legend=expression(G^2))
> legend(x=350,y=0.2,lty=2,bty="n",legend=expression(X^2))
> title("Power: Likelihood Ratio vs. Pearson Chi-square Tests")

```

**Power: Likelihood Ratio vs. Pearson Chi-square Tests**



b) What is the required sample size for the LRT test? The Pearson test?

```
> target
```

```
Power
df      0.5      0.7      0.8      0.9
1 3.841022 6.172007 7.848883 10.50739
2 4.956736 7.701774 9.634685 12.65394
3 5.760482 8.792389 10.902570 14.17149
4 6.419476 9.682473 11.935286 15.40503
5 6.991270 10.452523 12.827607 16.46946
6 7.503313 11.140677 13.624286 17.41883
7 7.971192 11.768443 14.350527 18.28355
8 8.404641 12.349332 15.022138 19.08270
```

```
> # Required sample size = required lambda / effect size
```

```
> # LRT
```

```
> 9.634685 / g2_effsize
```

```
[1] 260.2251
```

```
> # Pearson
```

```
> 9.634685 / x2_effsize
```

```
[1] 266.1582
```

```
> # Equal sample sizes per condition:
```

```
> 261/3
```

```
[1] 87
```

```
> 267/3
```

```
[1] 89
```

# Power for a retrospective design

Middle aged Canadians are classified as either above or below the median in the percent of calories in their diet that come from fresh fruits and vegetables. Suppose that the probability of cancer is 1% for those eating an above average amount of fruits and vegetables (the "healthy eaters"), and 2% for the other group.

a) For a prospective design, what sample size is required to reject the null hypothesis of independence with a probability of 0.80 at  $\alpha = 0.05$  using a likelihood ratio test?

```
> pro = rbind(c(99,1),
+           c(98,2) )
> pro_effsize = loglik(pro,margin=list(1,2))$lrt/200
2 iterations: deviation 0
> pro_effsize
[1] 0.001724371
> # Required sample size = required lambda / effect size
> 7.848883/pro_effsize # Ouch.
[1] 4551.736
```

b) Now we will consider a retrospective design, in which equal numbers with and without cancer are sampled, and we check whether or not they are healthy eaters (by this definition). Here are the numbers we need:

- i) Given cancer, what is the probability of being in the above average group?
- ii) Given No cancer, what is the probability of being in the above average group?

H = Healthy eater

C = Cancer

i) Given cancer, what is the probability of being in the above average group?

H = Healthy eater

C = Cancer

$$\begin{aligned}P(H|C) &= \frac{P(H \cap C)}{P(C)} \\&= \frac{P(C|H)P(H)}{P(C|H)P(H) + P(C|H^c)P(H^c)} \\&= \frac{P(C|H)}{P(C|H) + P(C|H^c)} \\&= \frac{1/100}{1/100 + 2/100} \\&= \frac{1}{3}\end{aligned}$$

ii) Given No cancer, what is the probability of being in the above average group?

$$\begin{aligned}P(H|C^c) &= \frac{P(H \cap C^c)}{P(C^c)} \\&= \frac{P(C^c|H)P(H)}{P(C^c|H)P(H) + P(C^c|H^c)P(H^c)} \\&= \frac{P(C^c|H)}{P(C^c|H) + P(C|H^c)} \\&= \frac{99/100}{99/100 + 98/100} \\&= \frac{99}{197} \approx 0.5025381\end{aligned}$$

```

> # Want to make this come out even to make a dummy table
> nn = 197*3
> nn * c(1/3,2/3)
[1] 197 394
> 99*3; (197-99)*3
[1] 297
[1] 294
>
> retro = rbind(c(197, 394),
+              c(297, 294) )
> rownames(retro) = c("Cancer","No Cancer")
> colnames(retro) = c("Healthy","Unhealthy")
> retro
      Healthy Unhealthy
Cancer      197      394
No Cancer   297      294
> # Here are the true pi_ij numbers
> prop.table(retro,1)
      Healthy Unhealthy
Cancer  0.3333333 0.6666667
No Cancer 0.5025381 0.4974619
>
> retro_effsize = loglin(retro,margin=list(1,2))$lrt/(2*nn)
2 iterations: deviation 0
> retro_effsize
[1] 0.02958543
> # Required sample size = required lambda / effect size
> 7.848883/retro_effsize
[1] 265.2955

```

So for a prospective design, we need  $n = 4,552$  while for a retrospective design we need  $n = 266$  to have the same power.