powerlog: power calculations for logistic regressions

Yezi “Zora” Yang

```r
# install.packages("../powerlog_0.1.0.tar.gz", repos = NULL, type = "source")
# library(powerlog)
source("R/power.R")
source("R/sampsize.R")
source("R/effectsize.R")
```

Example 1: S-B Table 1

```r
calc_pwr(b0 = -1, b1 = 0.33, n = 500, dist = "normal", mu = 0, sigma = 1)
```

```
## [1] 0.8901227
```

```r
calc_samp_size(b0 = -1, b1 = 0.33, targetpwr = 0.9, alpha = 0.05,
               dist = "normal", mu = 0, sigma = 1)
```

```
## [1] 517.2049
```

```r
approx_pwr(b0 = -1, b1 = 0.33, b2 = 0.9, rho = 0.5, n = 500,
           dist = "normal", mu = 0, sigma = 1)
```

```
## [1] 0.7473011
```

```r
approx_samp_size(b0 = -1, b1 = 0.33, b2 = 0.9, rho = 0.5, targetpwr = 0.9,
                 alpha = 0.05, dist = "normal", mu = 0, sigma = 1)
```

```
## [1] 761.8756
```

```r
# TODO: replace with data URL
nhanes = read.csv("Data/nhaneslarge.csv")
```

Example 2: NHANES The data set

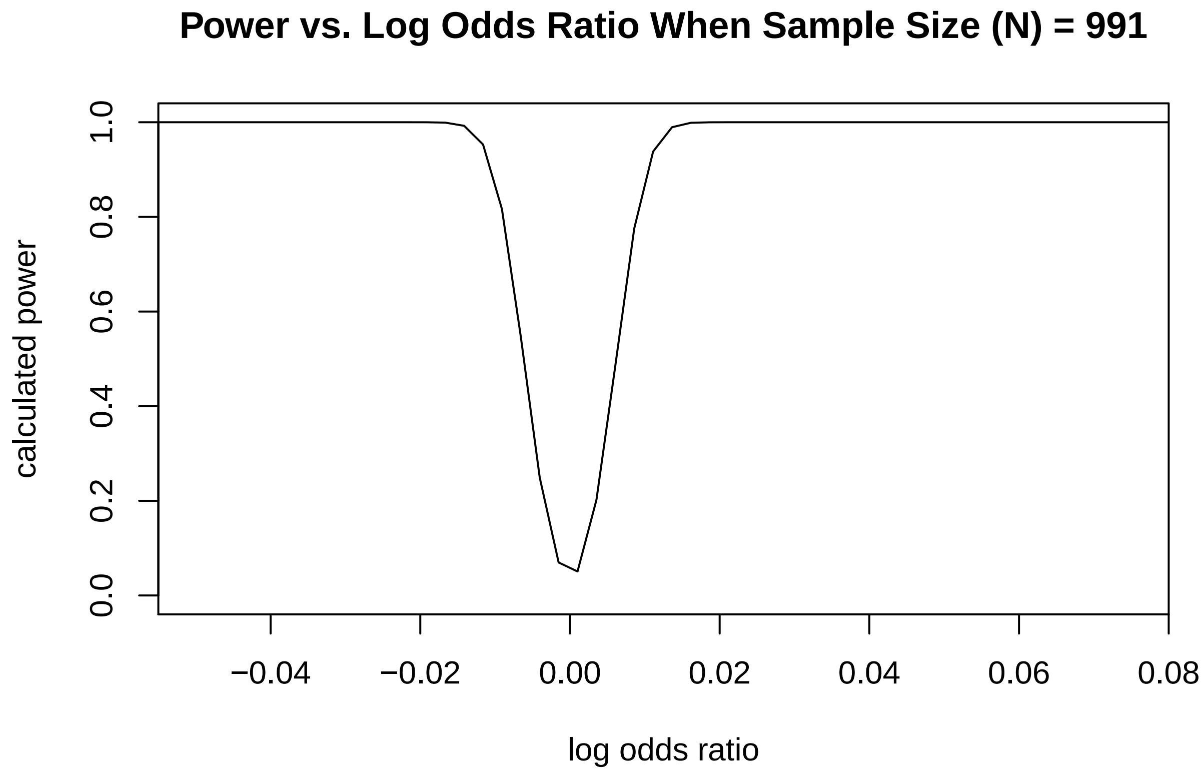
```r
names(nhanes)
```

```
##  [1] "BPXSAR"    "BPXDAR"    "BPXDI1"    "BPXDI2"    "race_ethc"
##  [6] "gender"    "DR1TFOLA"  "RIAGENDR"  "BMXBMI"    "RIDAGEYR"
```

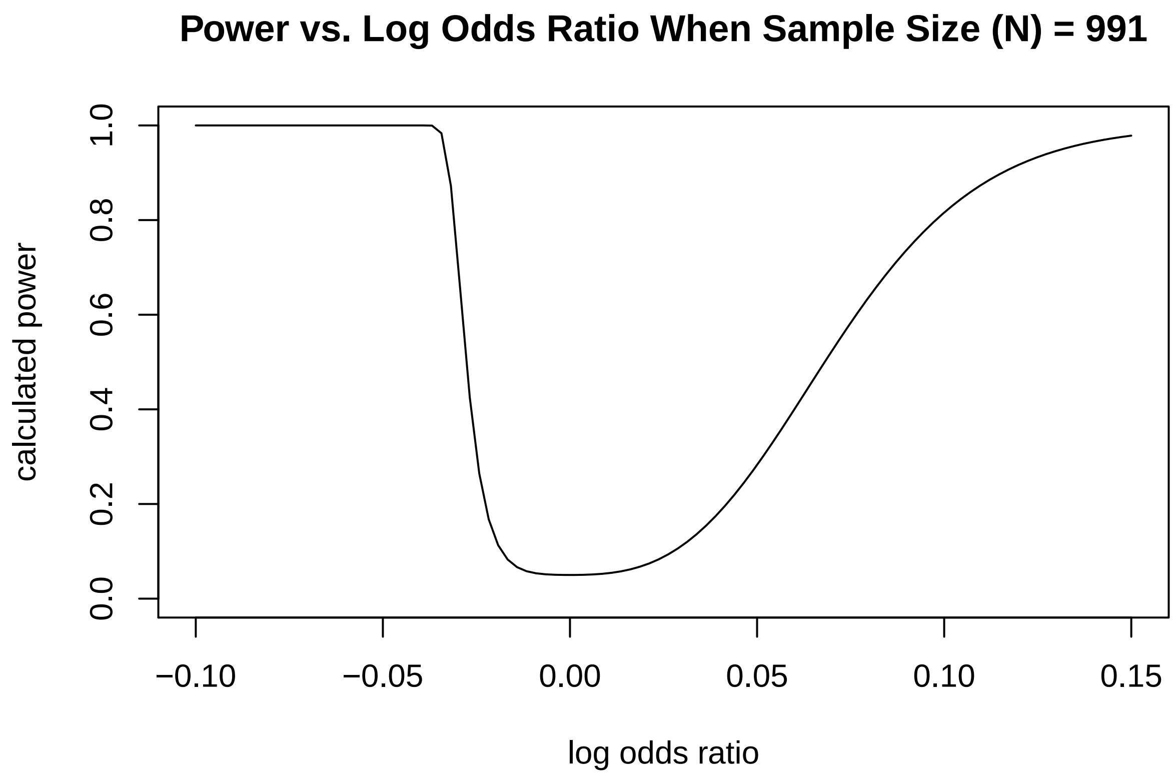
```r
nhanes <- nhanes[complete.cases(nhanes[,c("BMXBMI","RIDAGEYR","BPXDAR")]),]
nhanes$HighDBP = ifelse(nhanes$BPXDAR>70, 1, 0)
nhanes$BMXBMI = as.numeric(nhanes$BMXBMI)
nhanes$RIDAGEYR = as.numeric(nhanes$RIDAGEYR)

mod = glm(HighDBP ~ RIDAGEYR, data = nhanes)
mod.adj = glm(HighDBP ~ RIDAGEYR + BMXBMI, data = nhanes)

prop_nhanes = sum(nhanes$HighDBP==1)/length(nhanes$HighDBP)
```

Power curves

```r
granu = 100
lowerb = -0.1
upperb= 0.15
b1_hypos = seq(lowerb, upperb, length.out = granu)
xmat_nhanes_univ = cbind(rep(1, nrow(nhanes)), nhanes$RIDAGEYR)
b0_hats = sapply(X = b1_hypos, FUN = findb0emp, prop=prop_nhanes,
                 xmat = xmat_nhanes_univ, lower = -1, upper = 1)
# univariate
powers_univ = vector(length = granu)
for (i in 1:length(powers_univ)){
  pwr_i = calc_pwr_emp(b0 = b0_hats[i], b1 = b1_hypos[i], xmat = xmat_nhanes_univ,
                   alpha =.05, reg = "uni")
  powers_univ[i] = pwr_i
}
plot(x = b1_hypos, y = powers_univ, type = "l", ylim = c(0, 1), xlim = c(-.05, .075),
     xlab = "log odds ratio", ylab = "calculated power",
     main = paste0("Power vs. Log Odds Ratio When Sample Size (N) = ", nrow(nhanes)))
```



```
# adjusted
b2_hypo = 0.013
b2_hypos = rep(b2_hypo, granu)
xmat_nhanes_biv = cbind(rep(1, nrow(nhanes)), nhanes$RIDAGEYR, nhanes$BMXBMI)

powers_biv = vector(length = granu)
for (i in 1:length(powers_biv)){
  b0_hat = findb0emp(b1 = b1_hypos[i], b2 = b2_hypos[i], prop=prop_nhanes,
                     xmat = xmat_nhanes_biv, reg = "multi", lower = -1, upper = 1)
  pwr_i = calc_pwr_emp(b0 = b0_hat, b1 = b1_hypos[i], b2 = b2_hypos[i],
                       xmat = xmat_nhanes_biv, alpha =.05, reg = "multi")
  powers_biv[i] = pwr_i
}

plot(x = b1_hypos, y = powers_biv, type = "l", ylim = c(0, 1),
     xlab = "log odds ratio", ylab = "calculated power",
     main = paste0("Power vs. Log Odds Ratio When Sample Size (N) = ", nrow(nhanes)))
```



```
mod.adj = glm(HighDBP ~ RIDAGEYR + BMXBMI, data = nhanes); summary(mod.adj)
```

```
##
## Call:
## glm(formula = HighDBP ~ RIDAGEYR + BMXBMI, data = nhanes)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8099  -0.4444  -0.2384   0.4991   0.7953
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0314634  0.0666723  -0.472    0.637
## RIDAGEYR     0.0032497  0.0007862   4.134 3.88e-05 ***
## BMXBMI       0.0131364  0.0024380   5.388 8.90e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2337317)
##
##     Null deviance: 245.66  on 990  degrees of freedom
## Residual deviance: 230.93  on 988  degrees of freedom
## AIC: 1376.8
##
## Number of Fisher Scoring iterations: 2
```

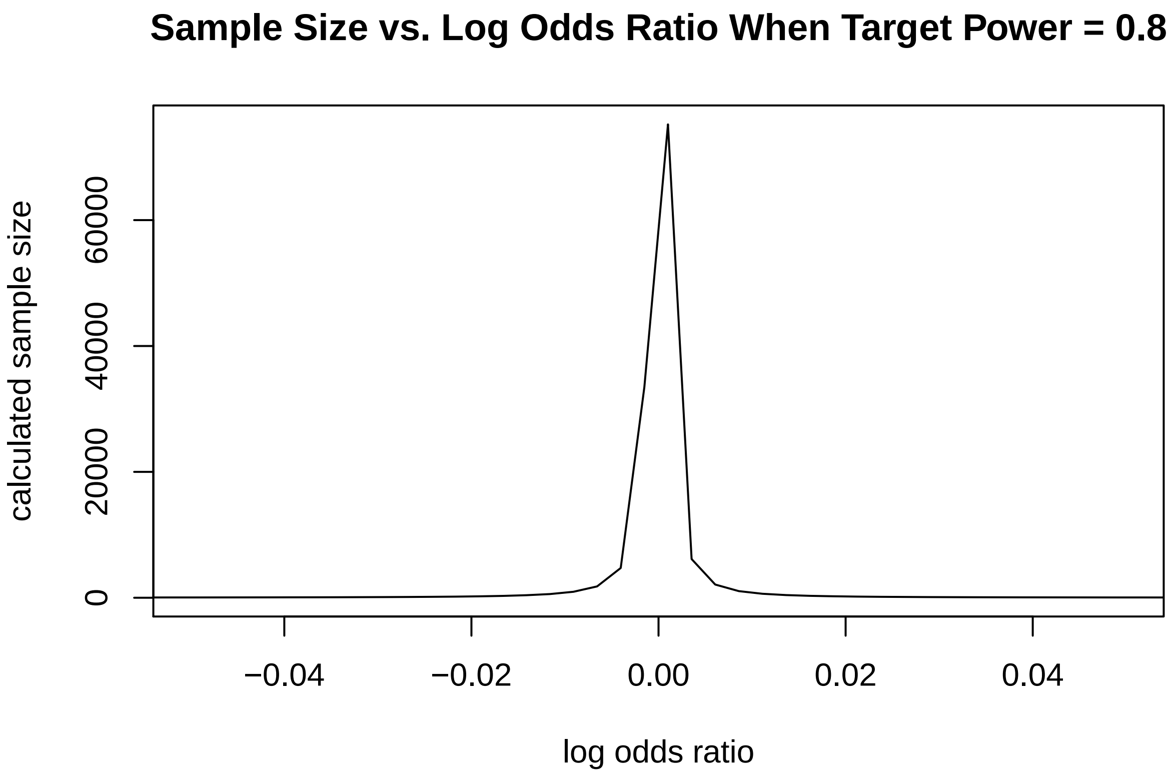
```
b0_MLE = -0.0314634; b1_MLE = 0.0032497; b2_MLE = 0.0131364
fitted_p = expit(xmat_nhanes_biv%*%c(b0_MLE, b1_MLE, b2_MLE))

p_vals = vector()
```

```
for (i in 1:1000){
  fitted_Y = rbinom(nrow(nhanes), 1, prob = fitted_p)
  p_vals[i] = summary(glm(fitted_Y ~ RIDAGEYR + BMXBMI, data = nhanes))$coeff[2,4]
}

sum(p_vals<0.05)/length(p_vals)
```

```
## [1] 0.181
```

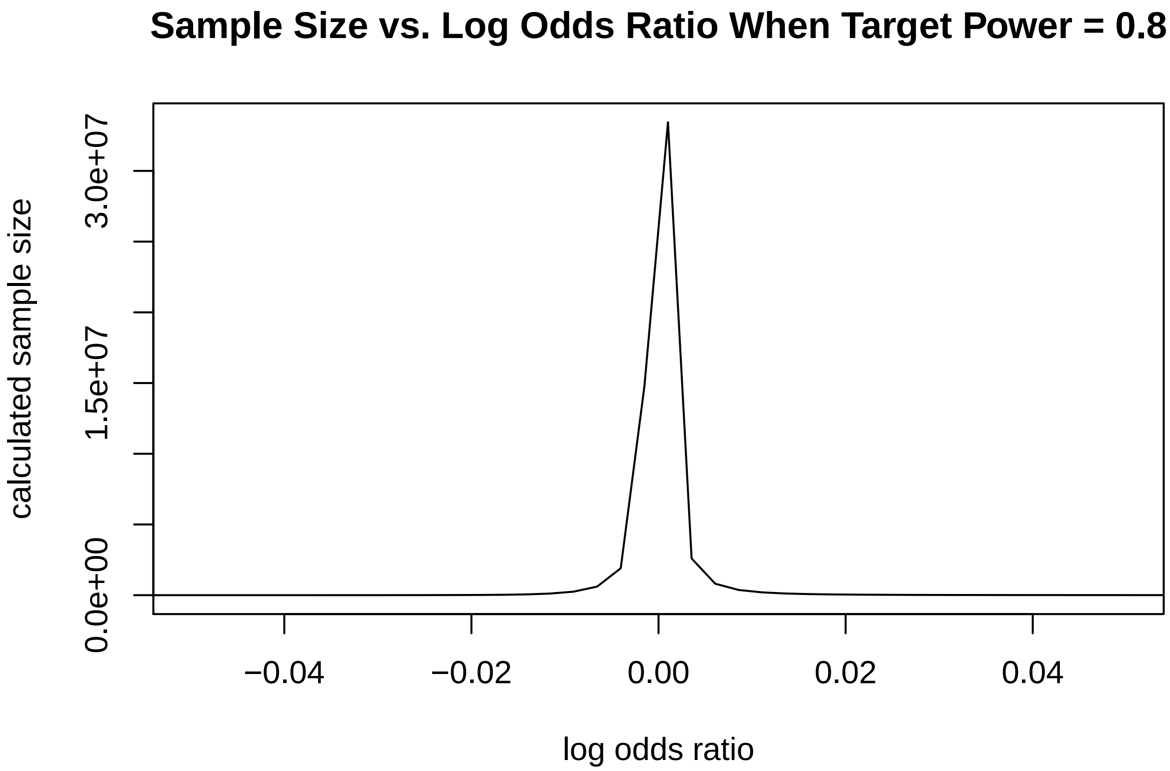
Sample size curve univariate

```
target_power = .80

# HighDBP ~ RIDAGEYR
ss_univ = vector(length = granu)
for (i in 1:length(powers_univ)){
  ss_i = calc_samp_size_emp(b0 = b0_hats[i], b1 = b1_hypos[i], xmat = xmat_nhanes_univ,
                  targetpwr = target_power, reg = "uni")
  ss_univ[i] = ss_i
}
plot(x = b1_hypos, y = ss_univ, type = "l", xlim = c(-0.05, 0.05),
     xlab = "log odds ratio", ylab = "calculated sample size",
     main = paste0("Sample Size vs. Log Odds Ratio When Target Power = ", target_power))
```



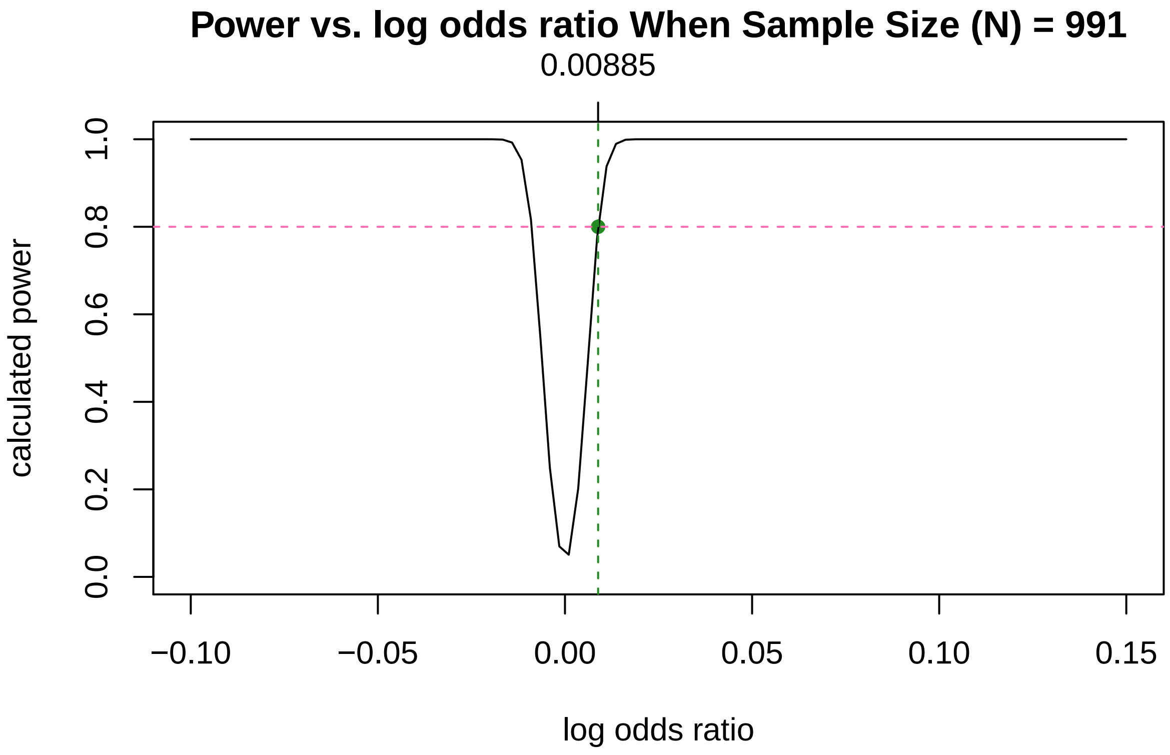
Sample size curve bivariate

```
# HighDBP ~ RIDAGEYR + BMXBMI
ss_biv = vector(length = granu)
for (i in 1:length(ss_biv)){
  b0_hat = findb0emp(b1 = b1_hypos[i], b2 = b2_hypos[i], prop = prop_nhanes, xmat = xmat_nhanes_biv, re
  ss_i = calc_samp_size_emp(b0 = b0_hat, b1 = b1_hypos[i], b2 = b2_hypos[i], xmat = xmat_nhanes_biv, ta
  ss_biv[i] = ss_i
}

plot(x = b1_hypos, y = ss_biv, type = "l", xlim = c(-0.05, 0.05),
     xlab = "log odds ratio", ylab = "calculated sample size",
     main = paste0("Sample Size vs. Log Odds Ratio When Target Power = ", target_power))
```

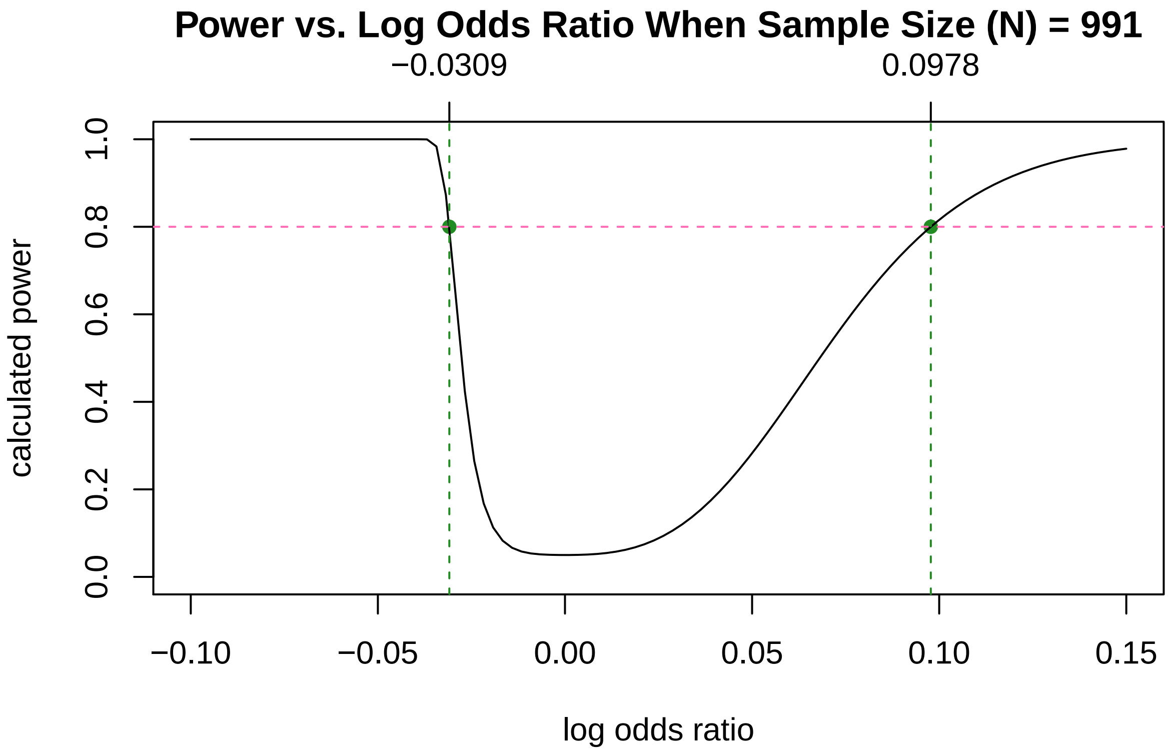


log odds ratio

```
# HighDBP ~ RIDAGEYR
b1hat1 = calc_effect_size_emp(xmat=xmat_nhanes_univ, prop=prop_nhanes, targetpwr=target_power, alpha=0.(
par(mar = c(5, 4, 5, 2) + 0.1)
plot(x = b1hat1, y = target_power, type = "p", col = "forestgreen", pch = 16,
     xlab = "log odds ratio", ylab = "calculated power",
     ylim = c(0, 1), xlim = c(lowerb, max(upperb, b1hat1*1.1)),
     main = paste0("Power vs. log odds ratio When Sample Size (N) = ", nrow(nhanes)))
lines(x = b1_hypos, y = powers_univ)
abline(h = target_power, col = "hotpink", lty = 2)
axis(side = 3, at = b1hat1, labels = signif(b1hat1, 3))
abline(v = b1hat1, col = "forestgreen", lty = 2)
```



```
# HighDBP ~ RIDAGEYR + BMI
b1hat2u = calc_effect_size_emp(xmat=xmat_nhanes_biv, prop=prop_nhanes,
                               b2=b2_hypo, targetpwr=target_power,
                               alpha=0.05, reg = "multi", lower=0, upper=0.15)
b1hat2l =  calc_effect_size_emp(xmat=xmat_nhanes_biv, prop=prop_nhanes,
                               b2=b2_hypo, targetpwr=target_power,
                               alpha=0.05, reg = "multi", lower=-.10, upper=0)
par(mar = c(5, 4, 5, 2) + 0.1)
plot(x = b1hat2u, y = target_power, type = "p", col = "forestgreen", pch = 16,
     xlab = "log odds ratio", ylab = "calculated power",
     ylim = c(0, 1), xlim = c(lowerb, max(upperb, b1hat2u*1.1)),
     main = paste0("Power vs. Log Odds Ratio When Sample Size (N) = ", nrow(nhanes)))
points(x = b1hat2l,y = target_power, type = "p", col = "forestgreen", pch = 16 )
lines(x = b1_hypos, y = powers_biv)
abline(h = target_power, col = "hotpink", lty = 2)
axis(side = 3, at = b1hat2u, labels = signif(b1hat2u, 3))
axis(side = 3, at = b1hat2l, labels = signif(b1hat2l, 3))
abline(v = b1hat2u, col = "forestgreen", lty = 2)
abline(v = b1hat2l, col = "forestgreen", lty = 2)
```



Genetic data example:

```
fms = read.delim("http://stat-gen.org/book.e1/data/FMS_data.txt", header=T, sep="\t")
#FAMuSS - functional SNPs associated with muscle size and strength

# trait is NDRM.CH - percentage change in muscle strength in non-dom arm

# before and after exercise training

# SNP is r577x, more at https://www.snpedia.com/index.php/Rs1815739

fms$Geno = as.numeric(fms$actn3_r577x)-1
fms$Male = as.numeric(fms$Gender=="Male") # a covariate to adjust for, if you need one.

table(fms$Male)
```

```
##
##   0   1
## 607 426
```

```
table(fms$Geno)
```

```
##
##   0   1   2
## 216 318 201
```

```
fms$Trait = fms$NDRM.CH
summary(fms$Trait)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   29.40   45.50   52.57   66.70  250.00     386
```

```
fms$binaryTrait = as.numeric(fms$Trait>=60)

fms2 = na.omit(fms[c("binaryTrait", "Geno")])
summary(glm(binaryTrait ~ Geno, data=fms2))$coef
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 0.30401277 0.03198963 9.503480 4.770272e-20
## Geno        0.05200051 0.02595494 2.003492 4.557299e-02
```

```
fms3 = na.omit(fms[c("binaryTrait", "Geno", "Male")])
summary(glm(binaryTrait ~ Geno + Male, data=fms3))$coef
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)  0.43968357 0.03355002 13.105313 1.093962e-34
## Geno         0.05327588 0.02436242  2.186806 2.914211e-02
## Male        -0.33698760 0.03717703 -9.064403 1.762057e-18
```

Same as before, parameters and etc.

```
mod.b = glm(binaryTrait ~ Geno, data=fms2)
prop.fms2 = sum(fms2$binaryTrait==1)/length(fms2$binaryTrait); prop.fms2
```

```
## [1] 0.3548922
```

```
mod.b.adj = glm(binaryTrait ~ Geno + Male, data=fms3)
mean(fms3$Geno, na.rm=T); mean(fms3$Male)
```

```
## [1] 0.9784411
```

```
## [1] 0.4063018
```

```
sd(fms3$Geno, na.rm=T); sd(fms3$Male)
```

```
## [1] 0.7501049
```

```
## [1] 0.4915499
```

```
prop.fms3 = sum(fms3$binaryTrait==1)/length(fms3$binaryTrait); prop.fms3
```

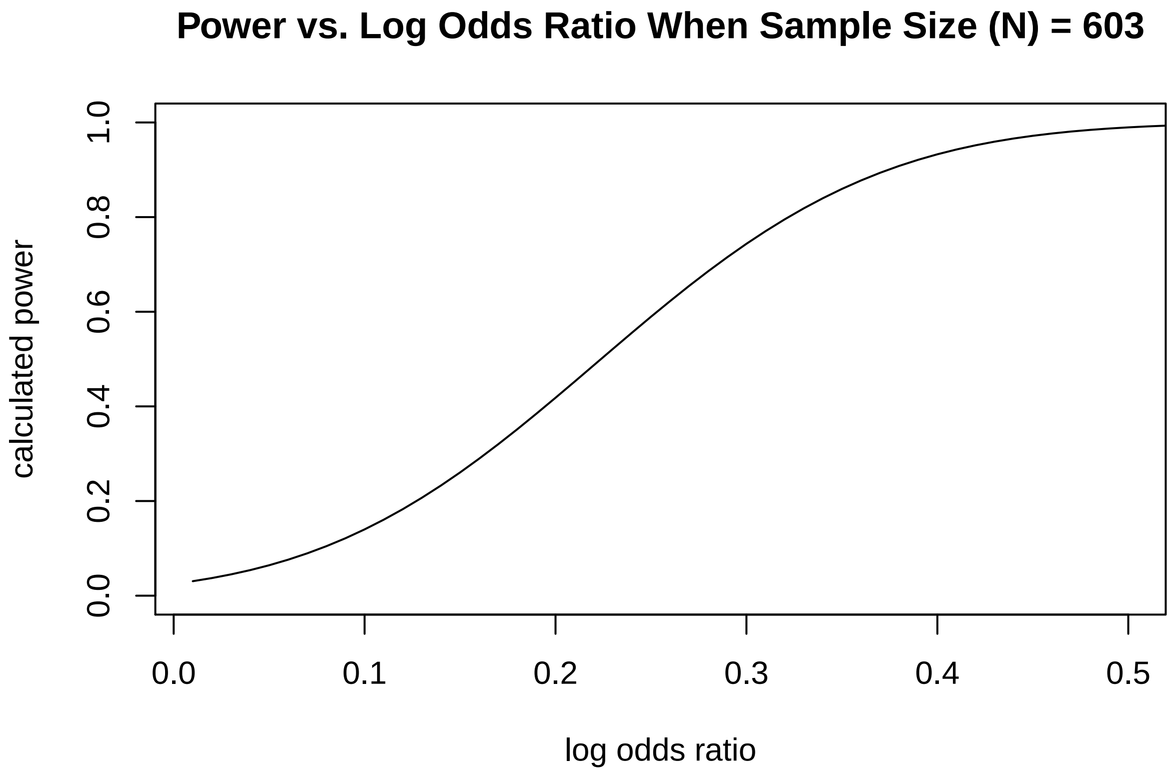
```
## [1] 0.3548922
```

Setup

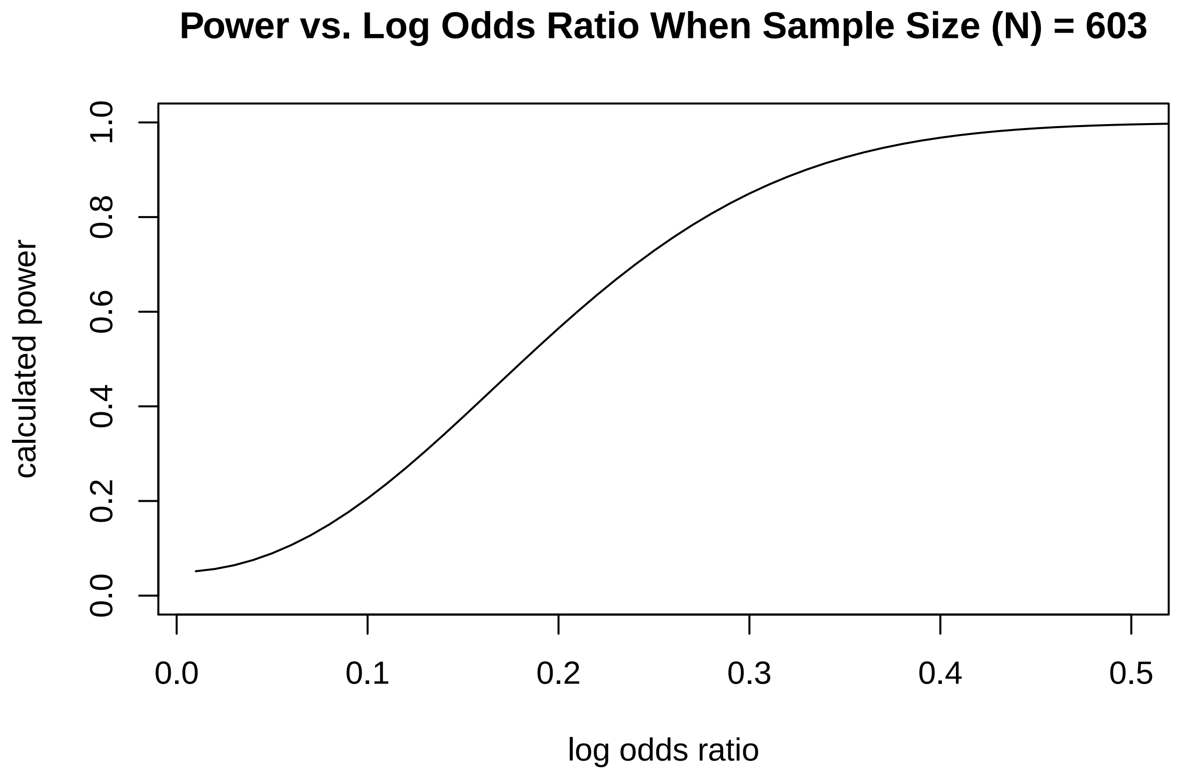
```
#library(powerlog)
granu.b = 100
lowerb.b = 0.01
upperb.b = 1
b1.hypos.b = seq(lowerb.b, upperb.b, length.out = granu.b)
xmat.fms.univ = cbind(rep(1, nrow(fms2)), fms2$Geno)
b0.hats.b = sapply(X = b1.hypos.b, FUN = findb0emp, prop=prop.fms2,
                   xmat = xmat.fms.univ, lower = -1, upper = 1)
b2.hypo.b = -0.33
xmat.fms.biv = cbind(rep(1, nrow(fms3)), fms3$Geno, fms3$Male)
```

Power curves

```
# BinaryTrait ~ Genotype
powers.univ.b = vector(length = granu.b)
for (i in 1:length(powers.univ.b)){
  pwr.i = calc_pwr_emp(b0 = b0.hats.b[i], b1 = b1.hypos.b[i], xmat = xmat.fms.univ,
                   alpha =.05, reg = "uni")
  powers.univ.b[i] = pwr.i
}
plot(x = b1.hypos.b, y = powers.univ.b, type = "l", ylim = c(0, 1), xlim=c(0.01, .5),
     xlab = "log odds ratio", ylab = "calculated power",
     main = paste0("Power vs. Log Odds Ratio When Sample Size (N) = ", nrow(fms2)))
```

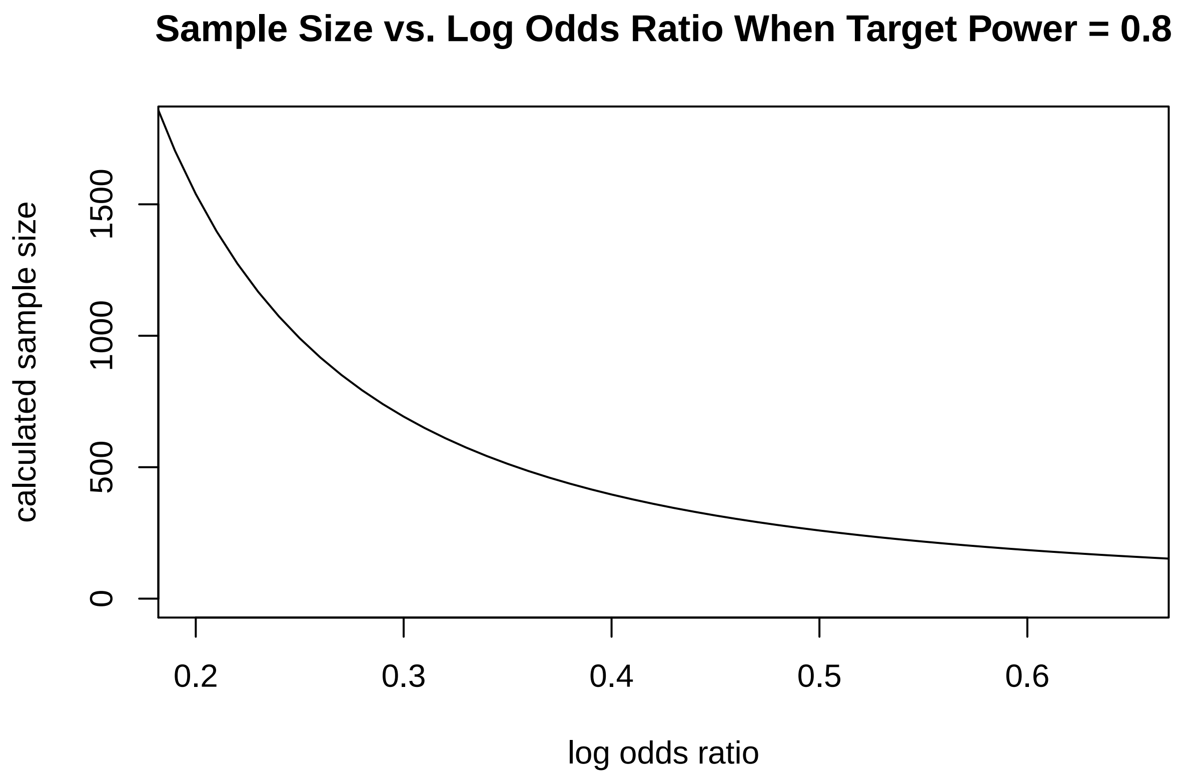


```
# BinaryTrait ~ Genotype + Male
powers.biv.b = vector(length = granu.b)
for (i in 1:length(powers.biv.b)){
  b0.hat = findb0emp(b1 = b1.hypos.b[i], b2 = b2.hypo.b, prop=prop.fms3, xmat = xmat.fms.biv, reg = "mul
  pwr.i = calc_pwr_emp(b0 = b0.hat, b1 = b1.hypos.b[i], b2 = b2.hypo.b, xmat = xmat.fms.biv, alpha =.05
  powers.biv.b[i] = pwr.i
}

plot(x = b1.hypos.b, y = powers.biv.b, type = "l", ylim = c(0, 1), xlim = c(0.01, 0.5),
     xlab = "log odds ratio", ylab = "calculated power",
     main = paste0("Power vs. Log Odds Ratio When Sample Size (N) = ", nrow(fms3)))
```

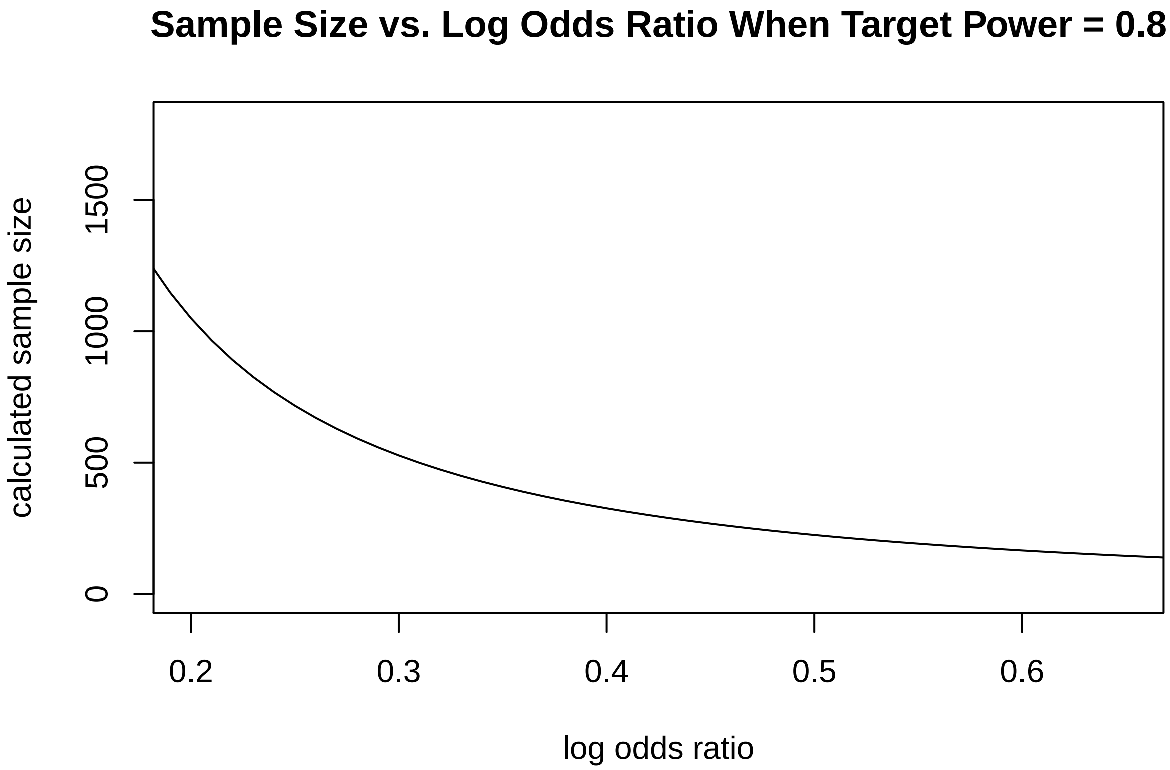


Sample size curves

```r
target.power.b = .80
# BinaryTrait ~ Genotype
ss.univ.b = vector(length = granu.b)
for (i in 1:length(powers.univ.b)){
  ss.i = calc_samp_size_emp(b0 = b0.hats.b[i], b1 = b1.hypos.b[i], xmat = xmat.fms.univ,
                 targetpwr = target.power.b, reg = "uni")
  ss.univ.b[i] = ss.i
}
plot(x = b1.hypos.b, y = ss.univ.b, type = "l", xlim = c(0.2, 0.65), ylim = c(0, 1800),
     xlab = "log odds ratio", ylab = "calculated sample size",
     main = paste0("Sample Size vs. Log Odds Ratio When Target Power = ", target.power.b))
```

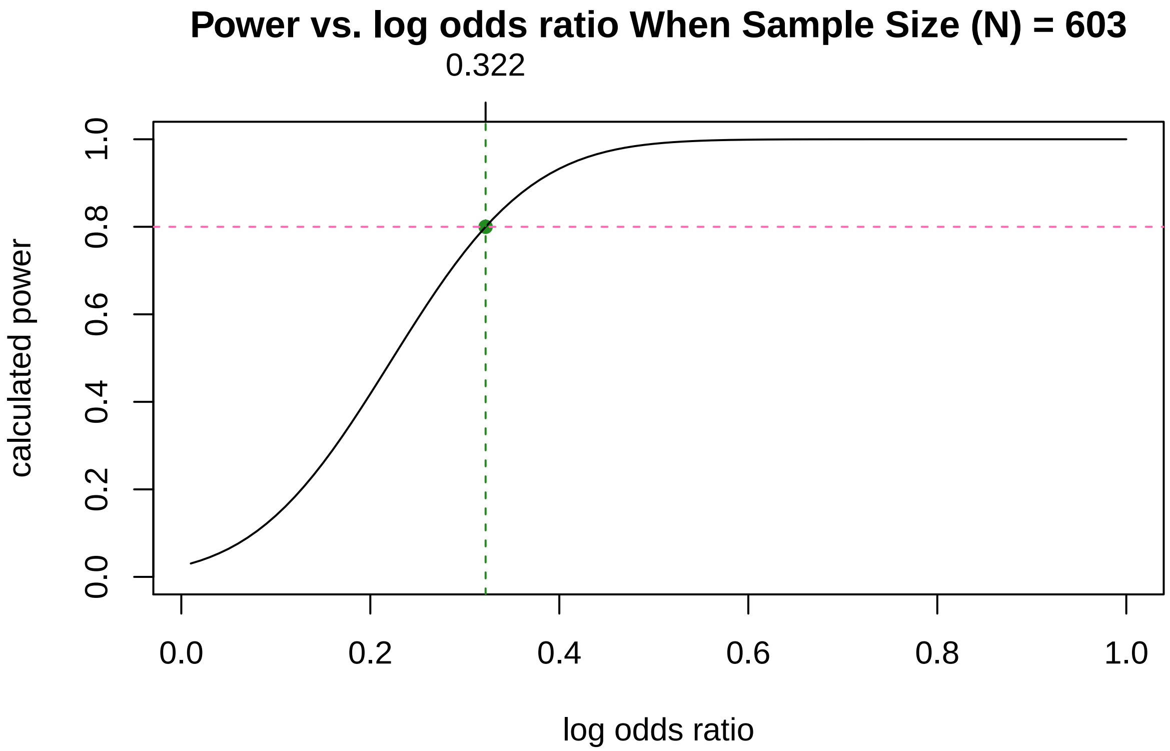


```
# BinaryTrait ~ Genotype + Male
ss.biv.b = vector(length = granu.b)
for (i in 1:length(ss.biv.b)){
  b0.hat = findb0emp(b1 = b1.hypos.b[i], b2 = b2.hypo.b, prop = prop.fms3, xmat = xmat.fms.biv, reg = "r
  ss.i = calc_samp_size_emp(b0 = b0.hat, b1 = b1.hypos.b[i],  b2 = b2.hypo.b, xmat = xmat.fms.biv, targe
  ss.biv.b[i] = ss.i
}

plot(x = b1.hypos.b, y = ss.biv.b, type = "l", xlim = c(0.2, 0.65), ylim = c(0, 1800),
     xlab = "log odds ratio", ylab = "calculated sample size",
     main = paste0("Sample Size vs. Log Odds Ratio When Target Power = ", target.power.b))
```

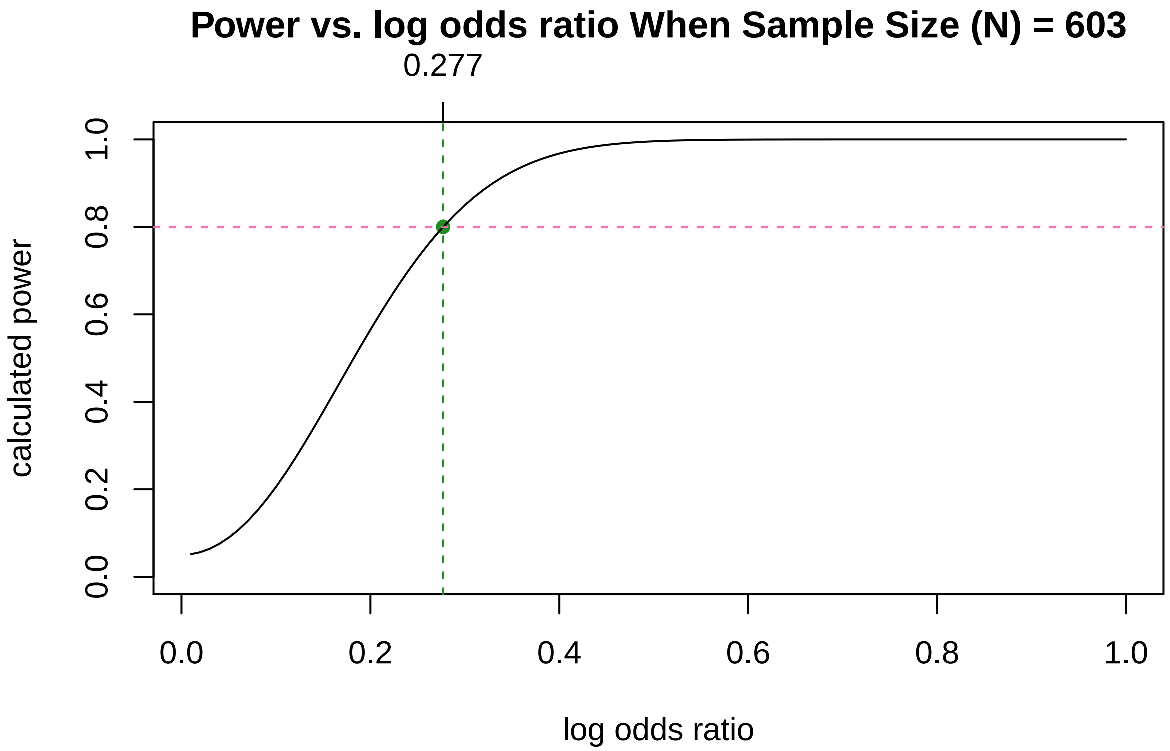


Estimated log odds ratio at the power target

```
b1hat1.b2 = calc_effect_size_emp(xmat=xmat.fms.univ, prop=prop.fms2, targetpwr = target.power.b, alpha=

par(mar = c(5, 4, 5, 2) + 0.1)
plot(x = b1hat1.b2, y = target.power.b, type = "p", col = "forestgreen", pch = 16,
     xlab = "log odds ratio", ylab = "calculated power",
     ylim = c(0, 1), xlim = c(lowerb.b, max(upperb.b, b1hat1.b2*1.1)),
     main = paste0("Power vs. log odds ratio When Sample Size (N) = ", nrow(fms2)))
lines(x = b1.hypos.b, y = powers.univ.b)
abline(h = target.power.b, col = "hotpink", lty = 2)
axis(side = 3, at = b1hat1.b2, labels = signif(b1hat1.b2, 3))
abline(v = b1hat1.b2, col = "forestgreen", lty = 2)
```



```
b1hat1.b3 = calc_effect_size_emp(xmat=xmat.fms.biv, prop=prop.fms3, b2=b2.hypo.b, targetpwr = target.pow

plot(x = b1hat1.b3, y = target.power.b, type = "p", col = "forestgreen", pch = 16,
     xlab = "log odds ratio", ylab = "calculated power",
     ylim = c(0, 1), xlim = c(lowerb.b, max(upperb.b, b1hat1.b3*1.1)),
     main = paste0("Power vs. log odds ratio When Sample Size (N) = ", nrow(fms3)))
lines(x = b1.hypos.b, y = powers.biv.b)
abline(h = target.power.b, col = "hotpink", lty = 2)
axis(side = 3, at = b1hat1.b3, labels = signif(b1hat1.b3, 3))
abline(v = b1hat1.b3, col = "forestgreen", lty = 2)
```


Power vs. log odds ratio When Sample Size (N) = 603
0.277
calculated power
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
log odds ratio