Power and Sample Size

Lavinia Carabet

Power and Sample Size

```
dat <- read.table('./agingStudy11FCortexAffy.txt', header=T, row.names=1)
dim(dat)
```

```
## [1] 12625    30
```

```
dat[1:5,]
```

```
##            GSM27015.26.M GSM27016.26.M GSM27018.29.M GSM27021.37.M
## 31307_at         179.8630      106.4950      265.5860      301.2430
## 31308_at         559.0780      411.4830      481.1760      570.7330
## 31309_r_at        20.7697       30.6415       50.2153       42.6892
## 31310_at         154.1910      224.4460      188.8230      177.8630
## 31311_at         956.7970      648.3100      933.6560     1016.4100
##            GSM27023.40.M GSM27024.42.M GSM27025.45.M GSM27027.52.M
## 31307_at         218.5090      224.6100      256.0590      137.9230
## 31308_at         333.5390      370.0790      558.0270      310.9570
## 31309_r_at        27.1059       21.5762       10.6286       47.6724
## 31310_at         233.4630      120.9080      217.8070       66.0645
## 31311_at         762.0130     1040.2900     1058.2000      695.9470
##            GSM27028.53.M GSM27031.66.M GSM27032.70.M GSM27034.73.M
## 31307_at          139.180       50.7749      260.5160       226.349
## 31308_at          404.723      426.2720      478.3090       447.678
## 31309_r_at         11.073       39.8862       15.7331        47.437
## 31310_at          205.010      126.1900      196.9490       157.376
## 31311_at         1185.670      658.6110      881.6030       658.250
##            GSM27035.77.M GSM27036.80.M GSM27038.85.M GSM27040.90.M
## 31307_at        173.98100      192.7170      141.1460      229.2120
## 31308_at        479.22900      445.5770      505.1160      379.0360
## 31309_r_at        7.34582       42.8014       26.7969       24.2459
## 31310_at        195.44200      339.7540      161.6910      157.2240
## 31311_at        870.22500      812.7340      849.2230      922.9160
##            GSM27042.91.M GSM27043.95.M GSM27017.27.F GSM27019.30.F
## 31307_at          251.251      378.7240      323.9430      193.1300
## 31308_at          437.748      339.7580      295.5310      457.5170
## 31309_r_at         48.923       53.0538       42.7983       33.8314
## 31310_at          273.100      345.5750      167.6910      223.6410
## 31311_at          767.145      516.2440      643.6290      821.7430
##            GSM27020.36.F GSM27022.38.F GSM27026.48.F GSM27029.56.F
## 31307_at         196.1630      183.3520      127.5160      154.1870
## 31308_at         446.2050      391.7210      238.4240      511.7930
```

```
## 31309_r_at       33.8327       32.3855       23.8202       18.0976
## 31310_at        164.8950      152.4320      183.8300      233.1270
## 31311_at        789.4560      831.7770     1703.3100      859.2930
##            GSM27030.61.F GSM27033.71.F GSM27037.81.F GSM27039.87.F
## 31307_at        188.3010      152.1210      139.1030       199.679
## 31308_at        300.5500      468.0530      463.4340       525.637
## 31309_r_at       27.0145       13.8967       14.9957        19.525
## 31310_at         98.7380      230.4060       68.8964       245.693
## 31311_at        820.1260     1128.4400      854.6600       755.398
##            GSM27041.90.F GSM27044.106.F
## 31307_at        114.8440       217.9670
## 31308_at        373.7830       385.1110
## 31309_r_at       43.4167        55.4229
## 31310_at        447.8360       246.6050
## 31311_at        899.7910       790.3100
```

```
dat <- as.data.frame(scale(dat))
dat[1:5,]
```

```
##            GSM27015.26.M GSM27016.26.M GSM27018.29.M GSM27021.37.M
## 31307_at      -0.4012821   -0.50182923    -0.2325196    -0.1863199
## 31308_at       0.1374789   -0.07244514     0.1065680     0.2243745
## 31309_r_at    -0.6273102   -0.60862126    -0.5712622    -0.5803478
## 31310_at      -0.4377550   -0.33576931    -0.3532551    -0.3743472
## 31311_at       0.7025289    0.26097698     0.8182444     0.9035724
##            GSM27023.40.M GSM27024.42.M GSM27025.45.M GSM27027.52.M
## 31307_at      -0.2942040   -0.29475825    -0.2973134    -0.4244374
## 31308_at      -0.1003360   -0.05489769     0.1586570    -0.1813824
## 31309_r_at    -0.6167887   -0.62953610    -0.6679122    -0.5512093
## 31310_at      -0.2690010   -0.46575013    -0.3550737    -0.5253745
## 31311_at       0.6218006    1.05019811     0.9139160     0.3594000
##            GSM27028.53.M GSM27031.66.M GSM27032.70.M GSM27034.73.M
## 31307_at     -0.47606206   -0.56102101    -0.2509553   -0.29881592
## 31308_at     -0.05100599   -0.03518737     0.1092388    0.01943368
## 31309_r_at   -0.68112361   -0.57626918    -0.6557863   -0.55607400
## 31310_at     -0.37068763   -0.45541221    -0.3560847   -0.39799240
## 31311_at      1.19905997    0.29017243     0.7762214    0.32221576
##            GSM27035.77.M GSM27036.80.M GSM27038.85.M GSM27040.90.M
## 31307_at     -0.43386907   -0.32555686   -0.46049716    -0.3215833
## 31308_at      0.03307096    0.06179213    0.08188765    -0.1041302
## 31309_r_at   -0.68877209   -0.55520827   -0.63089915    -0.6190690
## 31310_at     -0.40104003   -0.10031509   -0.42988118    -0.4260659
## 31311_at      0.63118032    0.62422941    0.59467279     0.6852517
##            GSM27042.91.M GSM27043.95.M GSM27017.27.F GSM27019.30.F
## 31307_at     -0.22773312   0.007847766    -0.1113452   -0.32685357
## 31308_at      0.05699974  -0.057870928    -0.1545169    0.08126045
## 31309_r_at   -0.53663585  -0.541416210    -0.5385411   -0.57275065
## 31310_at     -0.19437533  -0.048060179    -0.3487682   -0.27975607
## 31311_at      0.55990409   0.239784191     0.3744137    0.64348830
##            GSM27020.36.F GSM27022.38.F GSM27026.48.F GSM27029.56.F
## 31307_at     -0.35655101   -0.37259694    -0.4811573    -0.4466009
## 31308_at      0.02118438   -0.05428717    -0.3104594     0.1221680
## 31309_r_at   -0.60178141   -0.60321720    -0.6407551    -0.6630498
## 31310_at     -0.40378720   -0.41983112    -0.3944847    -0.3210476
```

```
## 31311_at      0.53972947    0.61795355     1.9441397     0.6748634
##            GSM27030.61.F GSM27033.71.F GSM27037.81.F GSM27039.87.F
## 31307_at      -0.3529005   -0.44957797   -0.47356369    -0.3618413
## 31308_at      -0.1762272    0.04808927   -0.03693362     0.1225193
## 31309_r_at    -0.6067558   -0.66731377   -0.64064293    -0.6295429
## 31310_at      -0.4938673   -0.32626068   -0.56807921    -0.2934663
## 31311_at       0.6415545    1.08835418    0.48975373     0.4639349
##            GSM27041.90.F GSM27044.106.F
## 31307_at     -0.48660260     -0.3019465
## 31308_at     -0.05919343     -0.0181946
## 31309_r_at   -0.60450173     -0.5778895
## 31310_at      0.06303973     -0.2533293
## 31311_at      0.80904443      0.6696913
```

```
ann <- read.table('./agingStudy1FCortexAffyAnn.txt', header=T, row.names=1)
dim(ann)
```

```
## [1] 30  2
```

```
ann
```

```
##          Gender Age
## GSM27015      M  26
## GSM27016      M  26
## GSM27018      M  29
## GSM27021      M  37
## GSM27023      M  40
## GSM27024      M  42
## GSM27025      M  45
## GSM27027      M  52
## GSM27028      M  53
## GSM27031      M  66
## GSM27032      M  70
## GSM27034      M  73
## GSM27035      M  77
## GSM27036      M  80
## GSM27038      M  85
## GSM27040      M  90
## GSM27042      M  91
## GSM27043      M  95
## GSM27017      F  27
## GSM27019      F  30
## GSM27020      F  36
## GSM27022      F  38
## GSM27026      F  48
## GSM27029      F  56
## GSM27030      F  61
## GSM27033      F  71
## GSM27037      F  81
## GSM27039      F  87
## GSM27041      F  90
## GSM27044      F 106
```

```
names(dat) <- substr(names(dat),1,8)
```

```
o <- row.names(ann[ann$Age >=50,]); y <- row.names(ann[ann$Age < 50,])
o; y
```

```
##  [1] "GSM27027" "GSM27028" "GSM27031" "GSM27032" "GSM27034" "GSM27035"
##  [7] "GSM27036" "GSM27038" "GSM27040" "GSM27042" "GSM27043" "GSM27029"
## [13] "GSM27030" "GSM27033" "GSM27037" "GSM27039" "GSM27041" "GSM27044"
```

```
##  [1] "GSM27015" "GSM27016" "GSM27018" "GSM27021" "GSM27023" "GSM27024"
##  [7] "GSM27025" "GSM27017" "GSM27019" "GSM27020" "GSM27022" "GSM27026"
```

Select a gene and plot the values for both samples

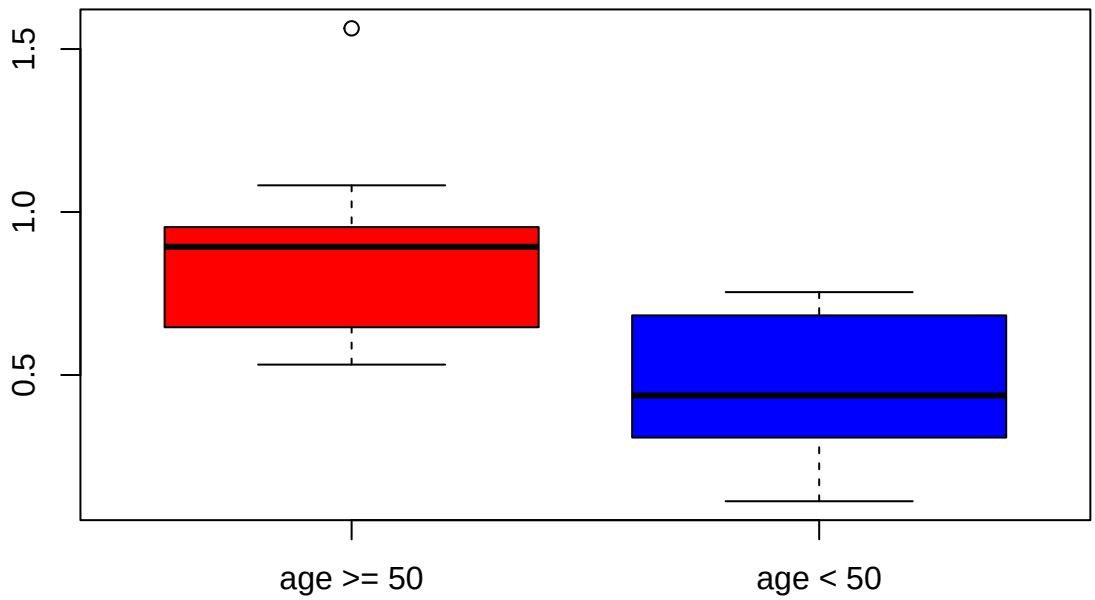
```
o.1 <- as.numeric(dat[8822, o])
y.1 <- as.numeric(dat[8822, y])

o.1; y.1
```

```
##  [1] 0.9537886 0.9453162 0.7109490 0.9167000 1.0443018 0.6465263 0.8970588
##  [8] 0.9855901 0.5370561 0.6973506 0.8896835 0.6031323 0.5710672 1.0818800
## [15] 0.5318590 0.7795261 0.9069673 1.5634529
```

```
##  [1] 0.4720785 0.3543752 0.4021776 0.6712631 0.7382950 0.1127189 0.7542355
##  [8] 0.3700816 0.2617471 0.4817855 0.6945991 0.1525636
```

```
# box plot
oy1.list <- list(o.1, y.1)
boxplot(oy1.list, col=c('red','blue'), names = c('age >= 50','age < 50'), main='Gene #1')
```



Print summary statistics on which the box plots are based on

```
boxplot(oy1.list, col=c('red','blue'), names = c('age >= 50','age < 50'), main='Gene #1', plot = FALSE)
```

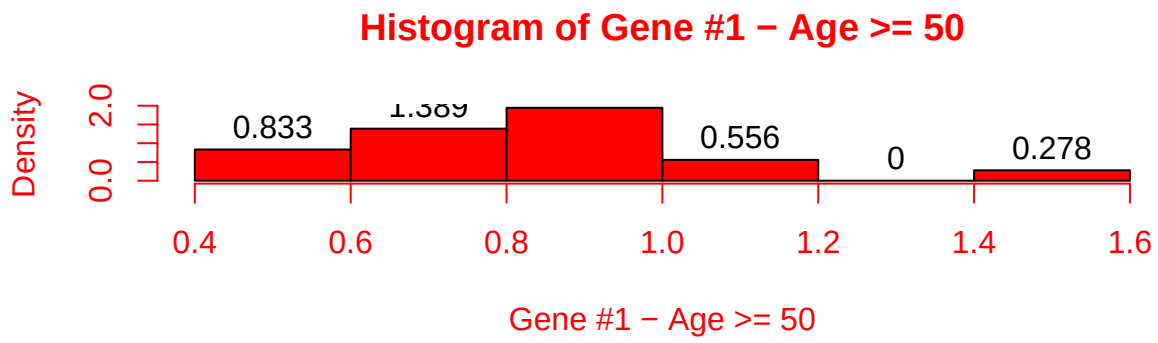
```
## $stats
##           [,1]      [,2]
## [1,] 0.5318590 0.1127189
## [2,] 0.6465263 0.3080612
## [3,] 0.8933711 0.4371281
## [4,] 0.9537886 0.6829311
## [5,] 1.0818800 0.7542355
##
## $n
## [1] 18 12
##
## $conf
##           [,1]      [,2]
## [1,] 0.7789437 0.2661474
## [2,] 1.0077985 0.6081088
##
## $out
## [1] 1.563453
##
## $group
## [1] 1
##
```

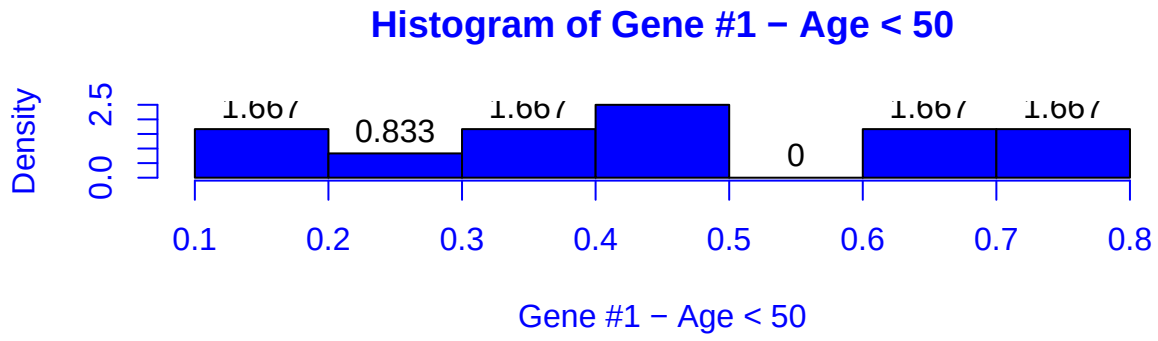
```
## $names
## [1] "age >= 50" "age < 50"
```

```
par(mfrow=c(2, 1))

xname<-"Gene #1 - Age >= 50"
hist(o.1,col='red', main=paste("Histogram of" , xname), xlab=xname,
     labels = TRUE, freq = FALSE, axes = FALSE, col.lab='red', col.main='red')
axis(side=1, col='red', col.axis='red')
axis(side=2, col='red', col.axis='red')

xname<-"Gene #1 - Age < 50"
hist(y.1,col='blue', main=paste("Histogram of" , xname), xlab=xname,
     labels = TRUE, freq = FALSE, axes = FALSE, col.lab='blue', col.main='blue')
axis(side=1, col='blue', col.axis='blue')
axis(side=2, col='blue', col.axis='blue')
```

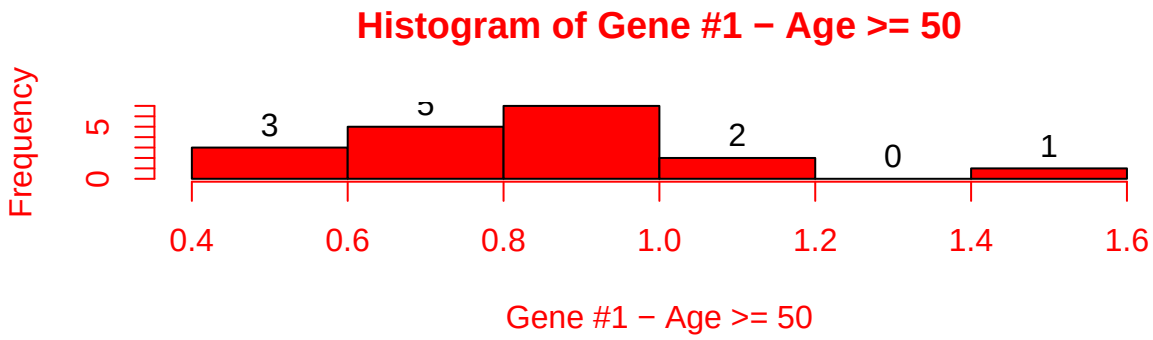


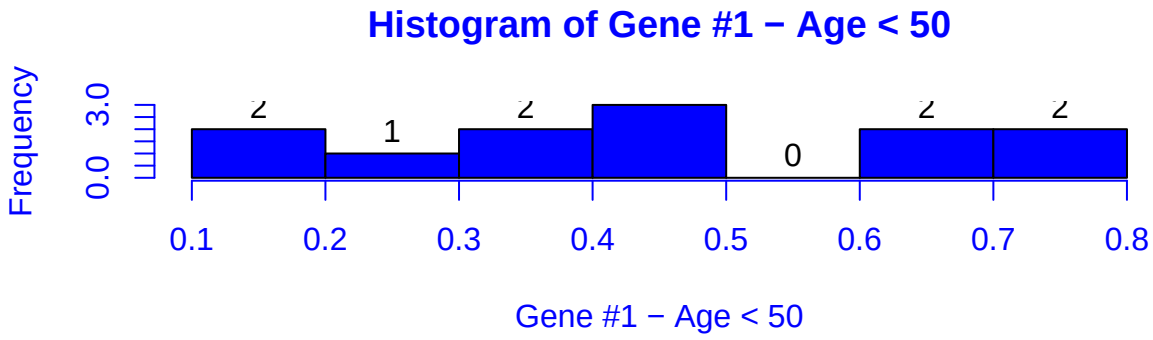



```
par(mfrow=c(2, 1))

xname<-"Gene #1 - Age >= 50"
hist(o.1,col='red', main=paste("Histogram of" , xname), xlab=xname,
     labels = TRUE, axes = FALSE, col.lab='red', col.main='red')
axis(side=1, col='red', col.axis='red')
axis(side=2, col='red', col.axis='red')
```

```
xname<-"Gene #1 - Age < 50"
hist(y.1,col='blue', main=paste("Histogram of" , xname), xlab=xname,
     labels = TRUE, axes = FALSE, col.lab='blue', col.main='blue')
axis(side=1, col='blue', col.axis='blue')
axis(side=2, col='blue', col.axis='blue')
```





Summary stats 'old' class for histogram

```
xname<-"Gene #1 - Age >= 50"
hist(o.1, plot = FALSE)
```

```
## $breaks
## [1] 0.4 0.6 0.8 1.0 1.2 1.4 1.6
##
## $counts
## [1] 3 5 7 2 0 1
##
## $density
## [1] 0.8333333 1.3888889 1.9444444 0.5555556 0.0000000 0.2777778
##
## $mids
## [1] 0.5 0.7 0.9 1.1 1.3 1.5
##
## $xname
## [1] "o.1"
```

```
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

Summary stats 'young' class for histogram

```
xname<-"Gene #1 - Age < 50"
hist(y.1, plot = FALSE)
```

```
## $breaks
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
##
## $counts
## [1] 2 1 2 3 0 2 2
##
## $density
## [1] 1.6666667 0.8333333 1.6666667 2.5000000 0.0000000 1.6666667 1.6666667
##
## $mids
## [1] 0.15 0.25 0.35 0.45 0.55 0.65 0.75
##
## $xname
## [1] "y.1"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

Calculate the minimum sample size necessary to detect a 1.5 fold difference inn the gene expression between the two groups at 80% power and with 99% confidence. Determine standard deviation of each group and choose max

```
o.1.sd <- sd(o.1)
y.1.sd <- sd(y.1)
max <- max(o.1.sd, y.1.sd)
o.1.sd; y.1.sd; max
```

```
## [1] 0.2517097
```

```
## [1] 0.2217859
```

```
## [1] 0.2517097
```

```
min.ssize <- ceiling(power.t.test(delta=log2(1.5),
                                  sd=max,
                                  sig.level=1-0.99,
                                  power=.8)$n)
min.ssize
```

Calculate number of replicates to detect 1.5 fold change at 80% power and 99% confidence

```
## [1] 7
```

```
power.t.test(delta=log2(1.5), sd=max, sig.level=1-0.99,power=.8)
```

```
##
##      Two-sample t test power calculation
##
##               n = 6.182428
##           delta = 0.5849625
##              sd = 0.2517097
##       sig.level = 0.01
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
n <- min(length(o.1),length(y.1))
n
```

```
## [1] 12
```

```
power <- round(
  power.t.test(n=n, delta=log2(2), sd=max, sig.level=1-0.99)$power*100, 2)
power
```

```
## [1] 100
```

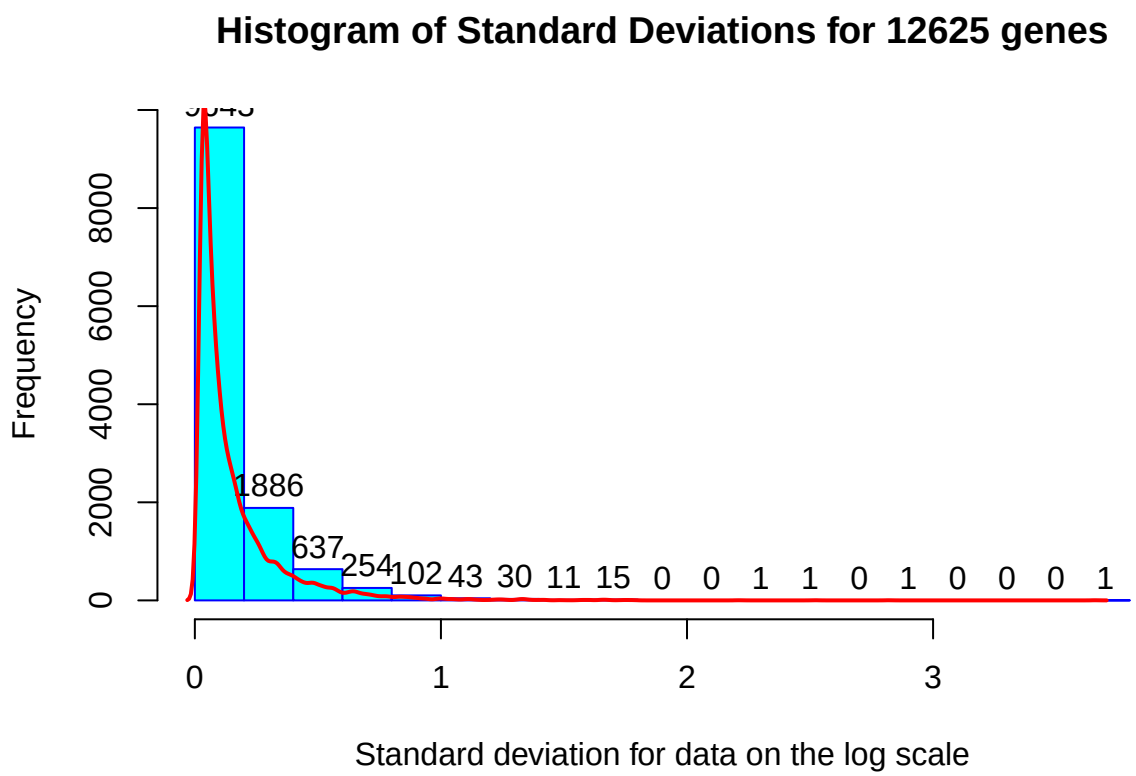
```
library(ssize)
library(gdata)
```

```
dat.sd <- apply(dat, 1, sd)
genes.no <- length(dat.sd)
hist(dat.sd, breaks=20, col="cyan", border="blue", main="",
     xlab=" Standard deviation for data on the log scale ", labels=TRUE)
dens <- density(dat.sd)
dens
```

```
##
## Call:
##  density.default(x = dat.sd)
##
```

```
## Data: dat.sd (12625 obs.);   Bandwidth 'bw' = 0.01512
##
##        x                  y
##  Min.   :-0.03069   Min.   :0.000000
##  1st Qu.: 0.90333   1st Qu.:0.000000
##  Median : 1.83735   Median :0.001893
##  Mean   : 1.83735   Mean   :0.267246
##  3rd Qu.: 2.77137   3rd Qu.:0.033400
##  Max.   : 3.70539   Max.   :8.004980
```

```
lines(dens$x, dens$y*par("usr")[4]/max(dens$y),col="red",lwd=2)
title(main = paste("Histogram of Standard Deviations for",genes.no, "genes"))
```



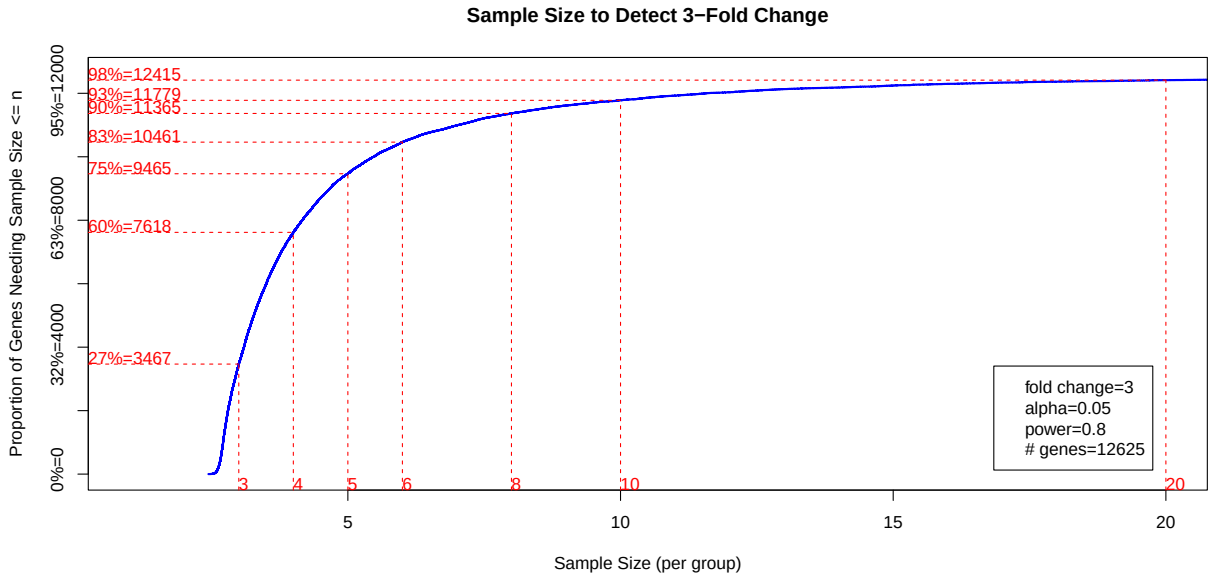

```
fold.change=3.0; power=0.8; sig.level=0.05;

all.size <- ssize(sd=dat.sd, delta=log2(fold.change),
                  sig.level=sig.level, power=power)
```

```
## ..........................................................................................
```

```
ssize.plot(all.size, lwd=2, col="blue", xlim=c(1,20))
xmax <- par("usr")[2]-1;
ymin <- par("usr")[3] + 0.05
legend(x=xmax, y=ymin,
```

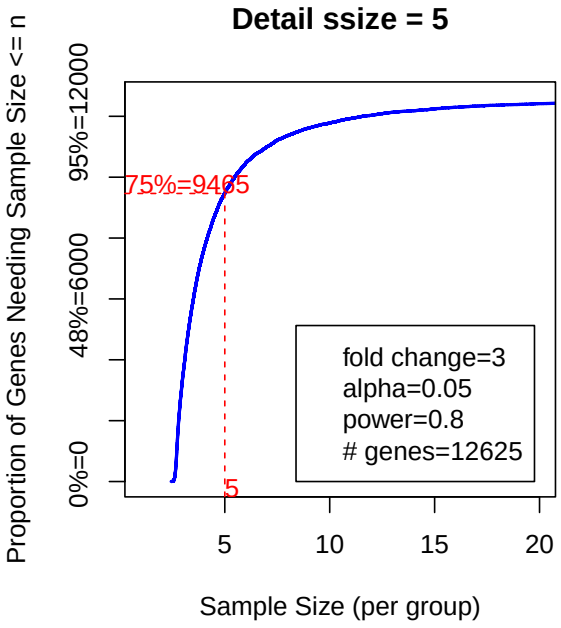
```
       legend= strsplit(paste("fold change=",fold.change,",",
                                              "alpha=", sig.level, ",",
                                              "power=",power,",", "# genes=",
                                 length(dat.sd), sep=''), "," )[[1]],
       xjust=1, yjust=0, cex=1.0)
title("Sample Size to Detect 3-Fold Change")
```

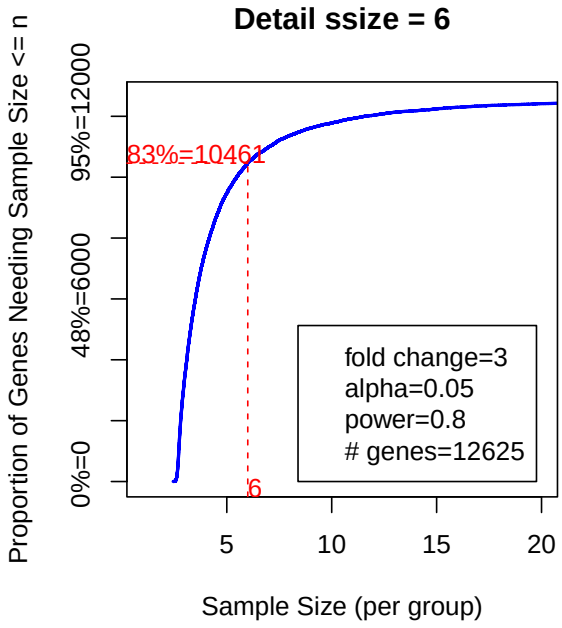


Sample Size to Detect 3-Fold Change

```
par(mfrow=c(1,2))
ssize.plot(all.size, lwd=2, col="blue", xlim=c(1,20),marks=5)
xmax <- par("usr")[2]-1;
legend(x=xmax, y=0,
       legend= strsplit( paste("fold change=",fold.change,",",
                               "alpha=", sig.level, ",", "power=",power,",",
                               "# genes=", length(dat.sd), sep=''), "," )[[1]],
       xjust=1, yjust=0, cex=1.0)
title("Detail ssize = 5")

ssize.plot(all.size, lwd=2, col="blue", xlim=c(1,20), marks=6)
xmax <- par("usr")[2]-1;
legend(x=xmax, y=0,
       legend= strsplit( paste("fold change=",fold.change,",",
                               "alpha=", sig.level, ",", "power=",power,",",
                               "# genes=", length(dat.sd), sep=''), "," )[[1]],
       xjust=1, yjust=0, cex=1.0)
title("Detail ssize = 6")
```





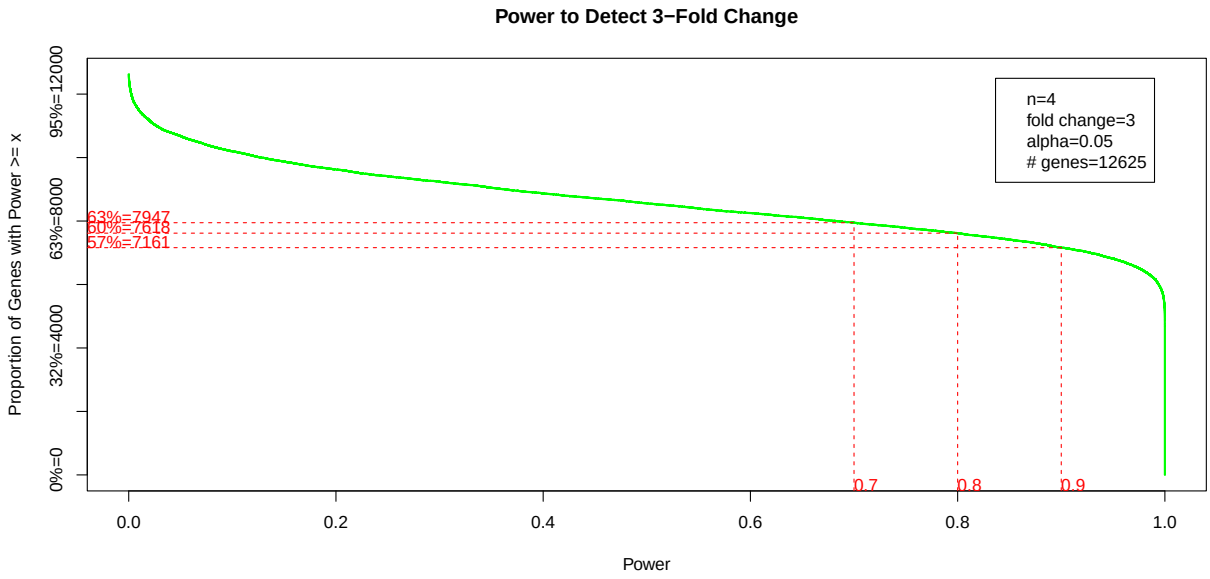
Power to Detect 3-Fold Change

How many genes are powered at the specified effect size (fold change), sample size, and confidence level (1-alpha)

```
fold.change=3.0; power=0.8; sig.level=0.05; n=4
all.power <- pow(sd=dat.sd, n=n, delta=log2(fold.change), sig.level=sig.level)
```

```
## ..........................................................................................
```

```
power.plot(all.power, lwd=2, col="green")
xmax <- par("usr")[2] -0.05;
ymin <- par("usr")[4] -0.05
legend(x=xmax, y=ymin,
       legend= strsplit( paste("n=",n,",","fold change=",fold.change,",",
                               "alpha=", sig.level, ",",
                               "# genes=", length(dat.sd), sep=''), "," )[[1]],
       xjust=1, yjust=1, cex=1.0)
title("Power to Detect 3-Fold Change")
```


Power to Detect 3−Fold Change
Proportion of Genes with Power >= x
0%=0
32%=4000
63%=8000
95%=12000
63%=7947
60%=7618
57%=7161
n=4
fold change=3
alpha=0.05
# genes=12625
0.7
0.8
0.9
0.0
0.2
0.4
0.6
0.8
1.0
Power