example1.R

mascj

2026-01-13

```r
# Example 1: Normal responses
#
# Two-arm trial, new treatment vs control
#
# Observations distributed as
#
#   X~N(mu_c,sigma^2)  on the control arm
#
#   X~N(mu_t,sigma^2)  on the new treatment arm
#
# Treatment effect theta = mu_t - mu_c
#
# To test H_0: theta =< 0 vs theta > 0 with one-sided type I error
# probability alpha=0.025 and power 1-beta=0.9 at
#
#   theta = delta = 0.4
#
# when sigma^2 = 0.64

# Load functions to be called later

source("~/My files/deming_dec_2025_demos/standards.R")
source("~/My files/deming_dec_2025_demos/ersptest.R")
source("~/My files/deming_dec_2025_demos/findroot.R")

alpha=0.025
beta=0.1
sigma=0.8
delta=0.4

# Design a fixed sample size trial

ifix=(qnorm(1-alpha)+qnorm(1-beta))^2/delta^2
ifix
```

```
## [1] 65.67139
```

```r
# nfix= number of observations per treatment

nfix=ifix*(2*sigma^2)
nfix
```

```
## [1] 84.05938
```

```r
# ifix=65.67
# nfix=84.06, which rounds up to 85 observations per treatment

# Design a group sequential trial with 5 analyses
#
# Error spending design, non-binding futility boundary, rho-family
# error spending function, rho=2 for efficacy and futility boundaries
#
# Call function find_rfac_ersp1b to find the inflation factor

r=16          #  Gives high accuracy for numerical computations
na=5          #  Number of analyses
alpha=0.025   #  Type I error rate
beta=0.1      #  Type II error rate
rho=2         #  Determines rate of error spending

rfac=find_rfac_ersp1b(r,na,alpha,beta,rho)
rfac
```

```
## [1] 1.132736
```

```r
# Required inflation factor is rfac=1.133

imax=rfac*ifix
imax
```

```
## [1] 74.38838
```

```r
nmax=imax*(2*sigma^2)
nmax
```

```
## [1] 95.21712
```

```r
# imax=0.4761
# Maximum sample size is just over 95 per treatment

# Inspect this design with cumulative information levels as planned

iobs=imax*c(1:5)/5
iobs
```

```
## [1] 14.87768 29.75535 44.63303 59.51070 74.38838
```

```r
# Define cumulative error probabilities cume
#
# cume[2;1:na]  cumulative type 1 error probabilities under theta=0
# cume[1;1:na]  cumulative type 2 error probabilities under theta=delta
#
# If we observe data with iobs[na] > imax, we cap type I error spent at
# alpha and the type II error spent at beta
#
# If we observe data with iobs[na] < imax, we set the final type I error
# to be alpha, so cume[2,na=alpha]

rho=2
cume=array(0,c(2,na))
cume[2,1:na]=pmin(alpha*(iobs/imax)^rho,alpha)
```

```r
cume[1,1:na]=pmin(beta*(iobs/imax)^rho,beta)
cume
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] 0.004 0.016 0.036 0.064 0.100
## [2,] 0.001 0.004 0.009 0.016 0.025
```

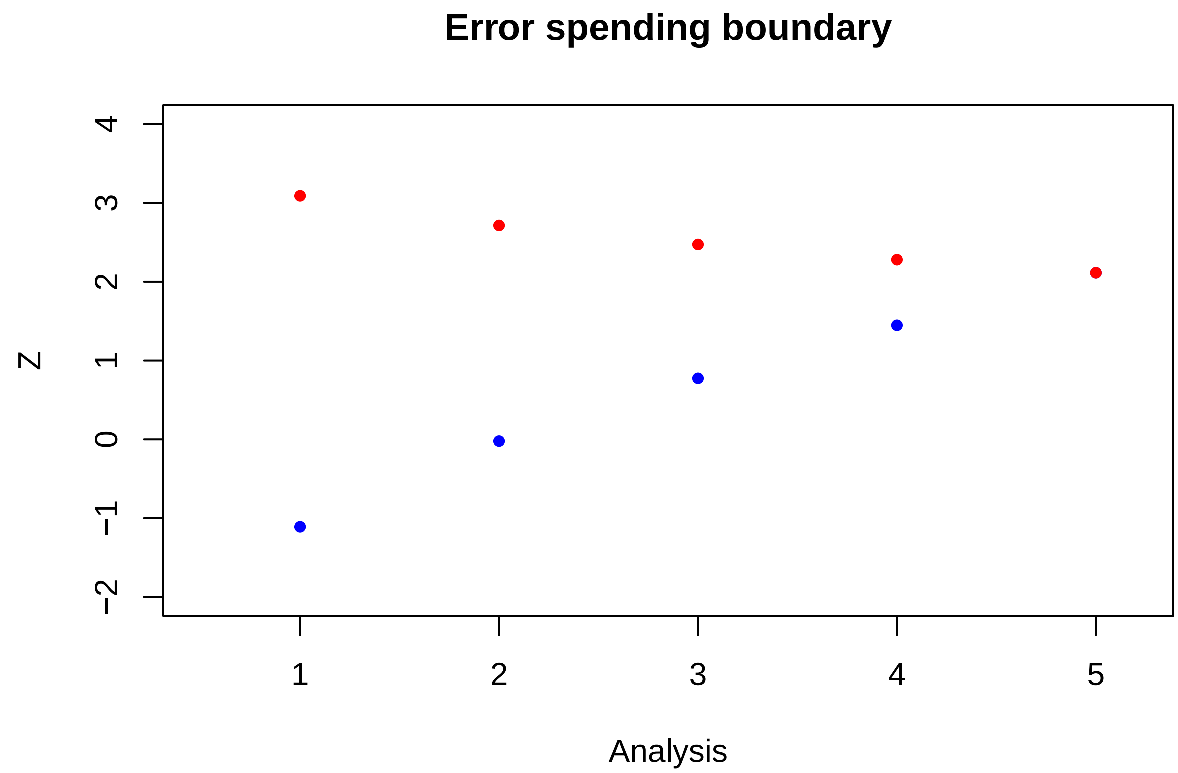
```r
out=ersp1b(r,na,iobs,delta,cume)
out  # Boundary, type I error probability, type II error probability
```

```
## [[1]]
##           [,1]        [,2]      [,3]     [,4]     [,5]
## [1,] -1.109206 -0.02230967 0.7743041 1.447198 2.114028
## [2,]  3.090233  2.71411137 2.4727772 2.279863 2.114028
##
## [[2]]
## [1] 0.025
##
## [[3]]
## [1] 0.09999999
```

```r
zbdy=out[[1]]
zbdy
```

```
##           [,1]        [,2]      [,3]     [,4]     [,5]
## [1,] -1.109206 -0.02230967 0.7743041 1.447198 2.114028
## [2,]  3.090233  2.71411137 2.4727772 2.279863 2.114028
```

```r
plot(c(1:na),zbdy[1,],col="blue",ylim=c(-2,4),xlim=c(0.5,5.2),pch=20,
     main="Error spending boundary",ylab="Z",xlab="Analysis")
points(c(1:na),zbdy[2,],col="red",pch=20)
```




```r
# Under theta=0

theta=0
out=gst1(r,na,iobs,zbdy,theta)
out
```

```
## [[1]]
##              [,1]       [,2]        [,3]        [,4]        [,5]
## [1,] 0.1336705967 0.36689448 0.292061004 0.139177482 0.045086409
## [2,] 0.0009999964 0.00299999 0.004996126 0.006890495 0.007223388
## [3,] 0.1346705932 0.36989447 0.297057130 0.146067977 0.052309797
##
## [[2]]
## [1] 0.02311
##
## [[3]]
## [1] 0.97689
##
## [[4]]
## [1] 38.85233
```

```r
# Output: pstop(1:3,1:na)  pstop(1,k)=P(Trial stops by crossing the lower
#                                      boundary at analysis k)
#                          pstop(2,k)=P(Trial stops by crossing the upper
#                                      boundary at analysis k)
#                          pstop(3,k)=pstop(1,k)+pstop(2,k)
#         pu               P(Exit by upper boundary)
```

```r
#         pl                P(Exit by lower boundary)
#         einf              E(information on termination)

einf=out[[4]]
100*einf/ifix # Expected sample size as a percentage of the fixed sample size
```

```
## [1] 59.16173
```

```r
# Under theta=delta

theta=delta
out=gst1(r,na,iobs,zbdy,theta)
out
```

```
## [[1]]
##             [,1]     [,2]      [,3]      [,4]       [,5]
## [1,] 0.003999999 0.012000 0.0200000 0.0280000 0.03599999
## [2,] 0.060887053 0.244007 0.2898351 0.2074266 0.09784419
## [3,] 0.064887052 0.256007 0.3098351 0.2354266 0.13384419
##
## [[2]]
## [1] 0.9
##
## [[3]]
## [1] 0.09999999
##
## [[4]]
## [1] 46.37868
```

```r
einf=out[[4]]
100*einf/ifix
```

```
## [1] 70.62235
```

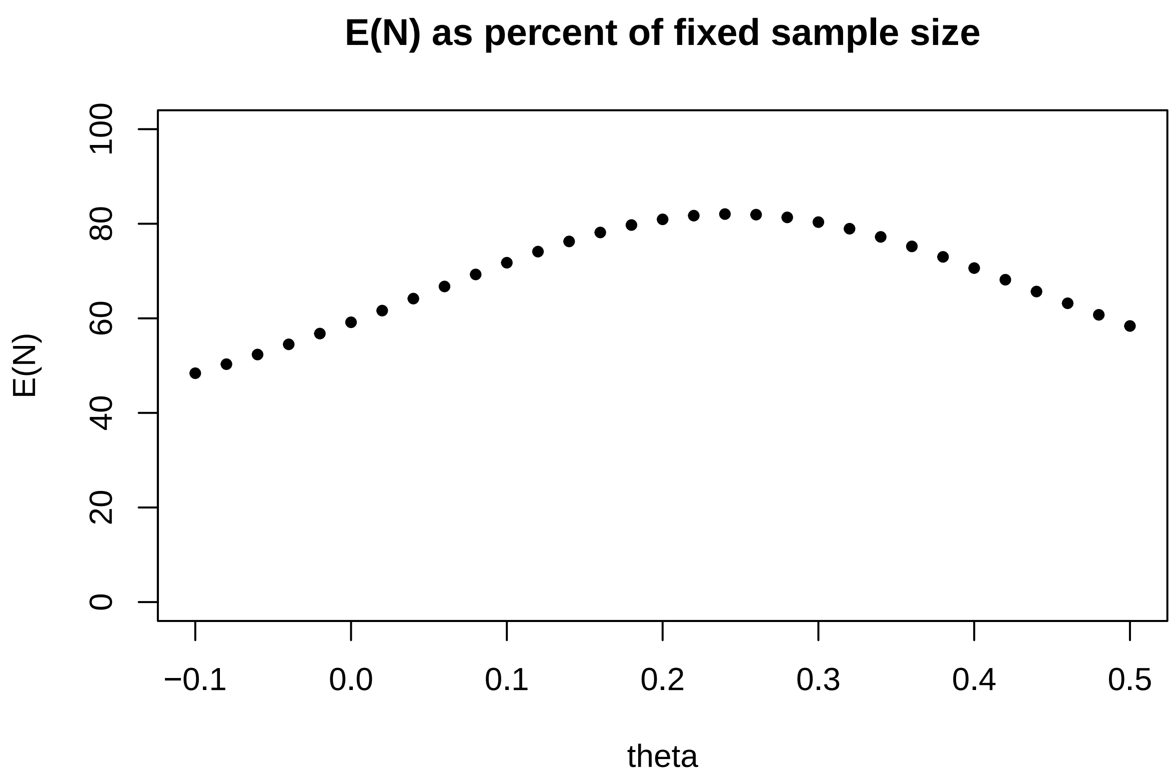
```r
# Under theta=delta/2

theta=delta/2
out=gst1(r,na,iobs,zbdy,theta)
out
```

```
## [[1]]
##            [,1]       [,2]       [,3]      [,4]      [,5]
## [1,] 0.03001058 0.11050374 0.16730454 0.1811001 0.1531262
## [2,] 0.01020290 0.04581434 0.08496303 0.1121673 0.1048074
## [3,] 0.04021348 0.15631808 0.25226757 0.2932674 0.2579336
##
## [[2]]
## [1] 0.357955
##
## [[3]]
## [1] 0.6420451
##
## [[4]]
## [1] 53.14885
```

```r
einf=out[[4]]
100*einf/ifix
```

```
## [1] 80.93151
```

```
theta.vec=seq(-0.1,0.5,0.02)
ntheta=length(theta.vec)
einf.percent.vec=rep(NA,ntheta)
for(ii in c(1:ntheta))
{
 theta=theta.vec[ii]
 einf.percent.vec[ii]=100*gst1(r,na,iobs,zbdy,theta)[[4]]/ifix
}
plot(theta.vec,einf.percent.vec,ylim=c(0,100),pch=20,
     main="E(N) as percent of fixed sample size",ylab="E(N)",xlab="theta")
```




```
# Running the trial with observed group sizes

# Suppose we have
#
# Cumulative sample sizes on treatment and control

nt=c(20,40,60,82,95)
nc=c(20,40,60,82,95)

# and treatment effect estimates

theta.hat=c(0.10,0.06,0.21,0.31)

# Then observed information levels are
```

```r
iobs=1/(sigma^2/nt+sigma^2/nc)
iobs
```

```
## [1] 15.62500 31.25000 46.87500 64.06250 74.21875
```

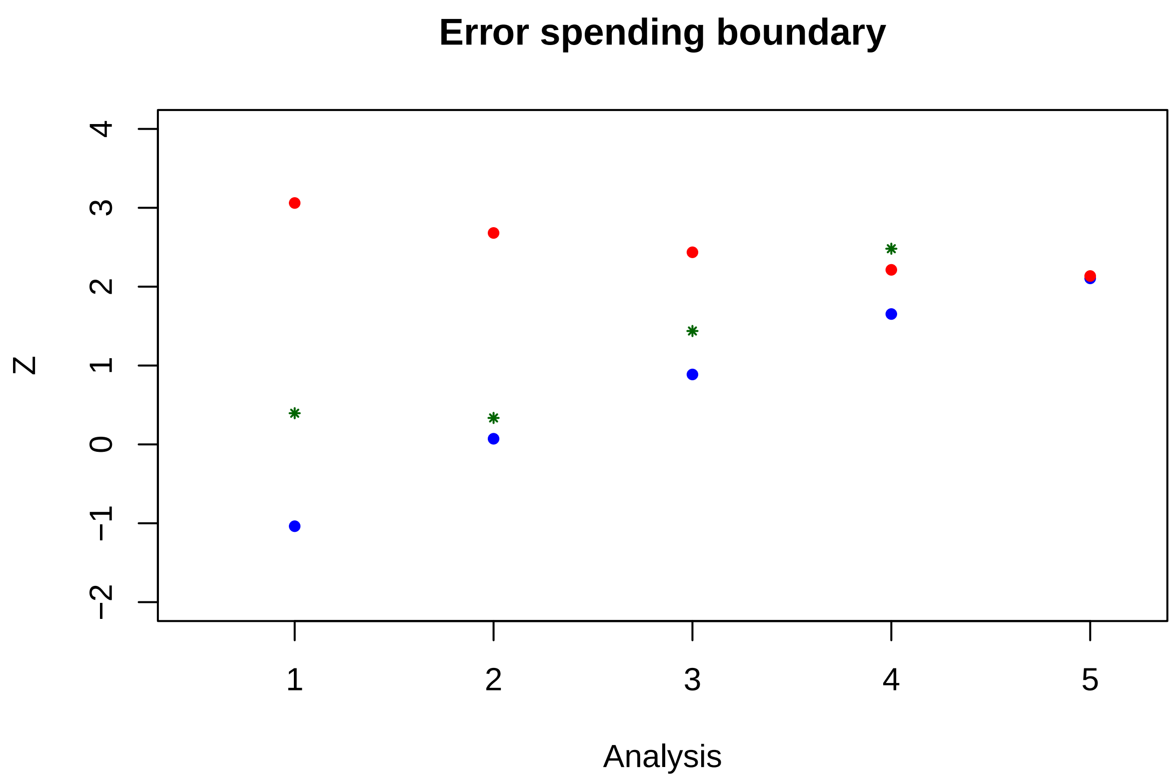
```r
# and Z-values are

zobs=theta.hat*sqrt(iobs[1:4])
zobs
```

```
## [1] 0.3952847 0.3354102 1.4377717 2.4812106
```

```r
# Error is spent according to the observed information iobs

cume[2,1:na]=pmin(alpha*(iobs/imax)^rho,alpha)
cume[2,na]=alpha
cume[1,1:na]=pmin(beta*(iobs/imax)^rho,beta)

# Giving the boundary

out=ersp1b(r,na,iobs,delta,cume)
zbdy=out[[1]]
zbdy
```

```
##            [,1]       [,2]      [,3]     [,4]     [,5]
## [1,] -1.037664 0.07156572 0.8866303 1.653104 2.105261
## [2,]  3.061004 2.68077772 2.4359035 2.213349 2.135177
```

```r
plot(c(1:na),zbdy[1,],col="blue",ylim=c(-2,4),xlim=c(0.5,5.2),pch=20,
     main="Error spending boundary",ylab="Z",xlab="Analysis")
points(c(1:na),zbdy[2,],col="red",pch=20)

# Comment in ersp1b:
#
# With too little information: The lower and upper boundaries do not
# meet up. To use these in practice, take zbdy[na,2] as the final
# critical value, and ignore zbdy[1,na].

# Add the observed data to the plot

points(c(1:4),zobs,col="dark green",pch=8,cex=0.5)
```



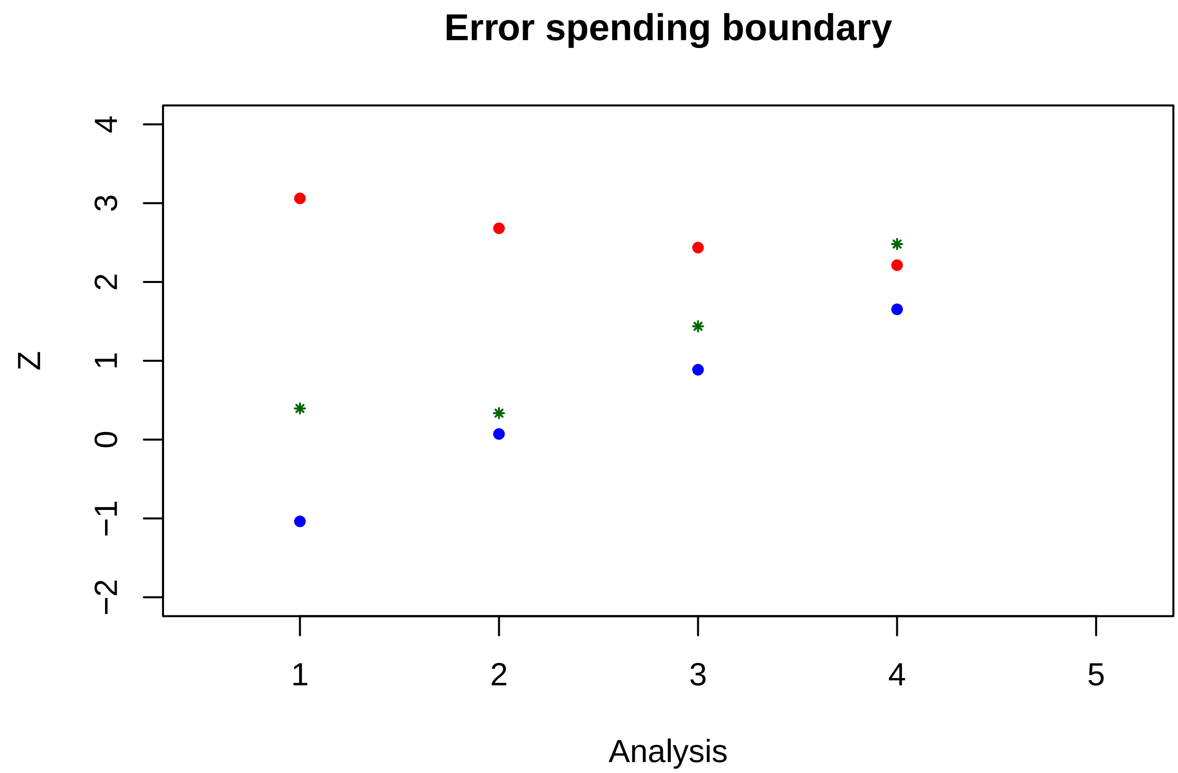

```
# The efficacy boundary would have been crossed at analysis 4

# At that point the boundary would have been calculated as

rho=2
cume4=array(0,c(2,4))
cume4[2,1:4]=pmin(alpha*(iobs[1:4]/imax)^rho,alpha)
cume4[1,1:4]=pmin(beta*(iobs[1:4]/imax)^rho,beta)
cume4
```

```
##             [,1]        [,2]        [,3]       [,4]
## [1,] 0.004411943 0.017647772 0.039707486 0.07416476
## [2,] 0.001102986 0.004411943 0.009926872 0.01854119
```

```
iobs4=iobs[1:4]
out=ersp1b(r,4,iobs4,delta,cume4)
zbdy4=out[[1]]
zbdy4
```

```
##           [,1]       [,2]      [,3]     [,4]
## [1,] -1.037664 0.07156572 0.8866303 1.653104
## [2,]  3.061004 2.68077772 2.4359035 2.213349
```

```
plot(c(1:4),zbdy4[1,],col="blue",ylim=c(-2,4),xlim=c(0.5,5.2),pch=20,
     main="Error spending boundary",ylab="Z",xlab="Analysis")
points(c(1:4),zbdy4[2,],col="red",pch=20)
points(c(1:4),zobs[1:4],col="dark green",pch=8,cex=0.5)
```




```
# With a binding futility boundary
#
# Still with imax=74.39,alpha=0.025, beta=0.1, delta=0.4
#
# Cumulative error rates are as before:

rho=2
cume=array(0,c(2,na))
cume[2,1:na]=pmin(alpha*(iobs/imax)^rho,alpha)
cume[1,1:na]=pmin(beta*(iobs/imax)^rho,beta)
cume[2,na]=alpha

out=ersp1c(r,na,iobs,delta,cume)
zbdy_binding_futility=out[[1]]
zbdy_binding_futility
```

```
##           [,1]       [,2]      [,3]     [,4]     [,5]
## [1,] -1.037664 0.07156572 0.8866303 1.653113 2.043569
## [2,]  3.061004 2.68077531 2.4355293 2.202682 2.043569
```

```
# Compare with the non-binding boundary

zbdy
```

```
##           [,1]       [,2]      [,3]     [,4]     [,5]
## [1,] -1.037664 0.07156572 0.8866303 1.653104 2.105261
## [2,]  3.061004 2.68077772 2.4359035 2.213349 2.135177
```