example2.R

mascj

2026-01-13

```r
# Example 2: Survival data
#
# Two-arm trial, new treatment vs control
#
# Observations follow a proportional hazards model with
# log hazard ratio theta, where
#
#    hazard ratio > 1 and theta > 0
#
# imply the new treatment is superior
#
# To test H_0: theta =< 0 vs theta > 0 with one-sided type I error
# probability alpha=0.025 and power 1-beta=0.8 at
#
#   theta = delta = 0.5 = log(1.65)

# 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=0.2
delta=0.5

# Design a fixed sample size trial

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

```
## [1] 31.39552
```

```r
# ifix=31.40

# 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
```

```r
na=5          #  Number of analyses
alpha=0.025   #  Type I error rate
beta=0.2      #  Type II error rate
rho=2         #  Determines rate of error spending

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

```
## [1] 1.133308
```

```r
# Required inflation factor is rfac=1.133 (the same to 3dp as for beta=0.1)

imax=rfac*ifix
imax
```

```
## [1] 35.58081
```

```r
# imax=35.58

# Inspect this design with cumulative information levels as planned

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

```
## [1]  7.116161 14.232323 21.348484 28.464645 35.580807
```

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

```
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] 0.008 0.032 0.072 0.128 0.200
## [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
```

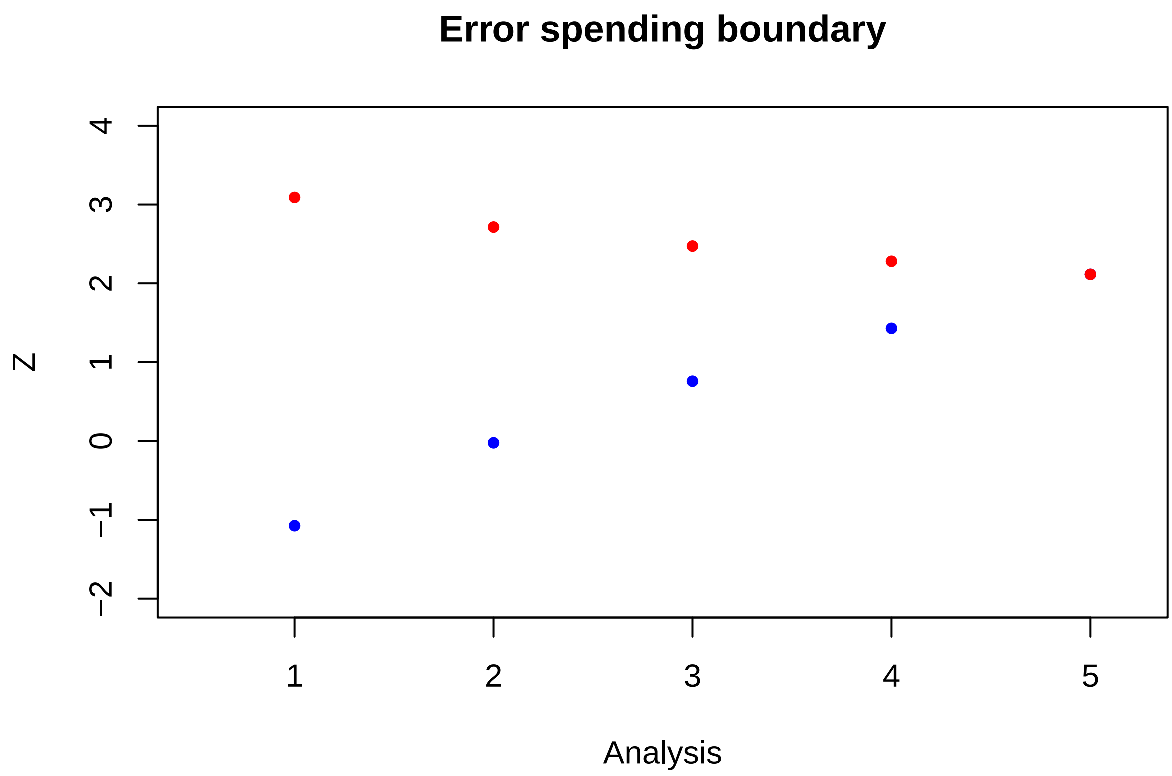
```
## [[1]]
##           [,1]        [,2]      [,3]     [,4]     [,5]
## [1,] -1.075109 -0.02317822 0.7579646 1.429153 2.114028
## [2,]  3.090233  2.71411137 2.4727772 2.279863 2.114028
##
## [[2]]
## [1] 0.025
```

```
##
## [[3]]
## [1] 0.2
```

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

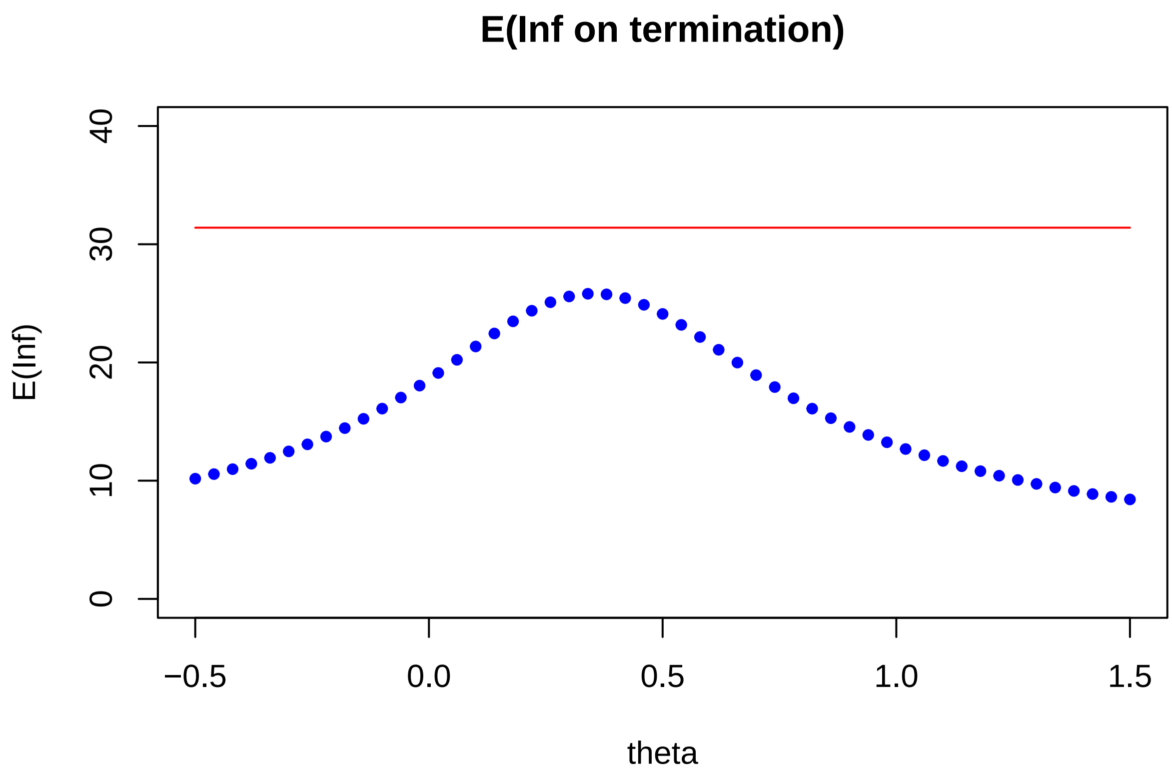
```
##           [,1]        [,2]      [,3]     [,4]     [,5]
## [1,] -1.075109 -0.02317822 0.7579646 1.429153 2.114028
## [2,]  3.090233  2.71411137 2.4727772 2.279863 2.114028
```

```
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)
```




```
# Expected information on termination

theta.vec=seq(-0.5,1.5,0.04)
ntheta=length(theta.vec)
einf.vec=rep(NA,ntheta)
for(ii in c(1:ntheta))
{
  theta=theta.vec[ii]
  einf.vec[ii]=gst1(r,na,iobs,zbdy,theta)[[4]]
}
plot(theta.vec,einf.vec,ylim=c(0,40),pch=20,col="blue",
     main="E(Inf on termination)",ylab="E(Inf)",xlab="theta")
lines(theta.vec,rep(31.4,ntheta),col="red")
```




```r
# Running the trial with observed information

iobs=c(5.43,12.58,21.11,30.55,33.28)
rho=2
cume=array(0,c(2,na))
cume[2,1:na]=pmin(alpha*(iobs/imax)^rho,alpha)
cume[2,na]=alpha
cume[1,1:na]=pmin(beta*(iobs/imax)^rho,beta)
cume
```

```
##              [,1]        [,2]        [,3]       [,4]      [,5]
## [1,] 0.0046579849 0.025001133 0.070400360 0.14744187 0.1749706
## [2,] 0.0005822481 0.003125142 0.008800045 0.01843023 0.0250000
```

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

```
##           [,1]       [,2]     [,3]     [,4]     [,5]
## [1,] -1.435118 -0.2322139 0.749690 1.638194 1.835698
## [2,]  3.247437  2.7822063 2.460957 2.201277 2.087693
```

```r
# With a binding futility boundary

rfac=find_rfac_ersp1c(r,na,alpha,beta,rho)
rfac
```

```
## [1] 1.09822
```

```
imax=rfac*ifix
imax
```

```
## [1] 34.47919
```

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

```
##                [,1]        [,2]        [,3]       [,4]      [,5]
## [1,] 0.0049603854 0.026624229 0.074970814 0.15701393 0.1863298
## [2,] 0.0006200482 0.003328029 0.009371352 0.01962674 0.0250000
```

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

```
##           [,1]       [,2]      [,3]     [,4]     [,5]
## [1,] -1.413461 -0.2049963 0.7835031 1.682853 1.898330
## [2,]  3.229497  2.7614338 2.4371611 2.163331 2.059422
```

```
# Back to a non-binding futility boundary

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

```
## [1] 1.133308
```

```
imax=rfac*ifix
imax
```

```
## [1] 35.58081
```

```
# Analysis of Cox model output

iobs=c(4.11,10.89,19.23,28.10,30.96)
rho=2
cume=array(0,c(2,na))
cume[2,1:na]=pmin(alpha*(iobs/imax)^rho,alpha)
cume[2,na]=alpha
cume[1,1:na]=pmin(beta*(iobs/imax)^rho,beta)
cume
```

```
##                [,1]        [,2]        [,3]       [,4]     [,5]
## [1,] 0.0026685912 0.018735020 0.058419385 0.12474153 0.151426
## [2,] 0.0003335739 0.002341878 0.007302423 0.01559269 0.025000
```

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

```
##           [,1]       [,2]      [,3]     [,4]     [,5]
## [1,] -1.772289 -0.4675028 0.5546372 1.413680 1.613849
## [2,]  3.402755  2.8654671 2.5213026 2.266767 2.048875
```