

Crossover Trial Further Simulation and Plotting

Eamonn O'Brien

24 July, 2019

Contents

Function to simulate crossover trial dataset	3
Generate data and analyse	4
Simulation function for crossover design, preparing to vary an input parameter	5
Simulate crossover power and vary between person SD	6
Simulate crossover power and vary within person SD	7
Function to simulate crossover trial using nlme	11
Computing Environment	12

Contents

List of Figures

List of Tables

Function to simulate crossover trial dataset

```

ABBA <- function(n = 10, sdW = 4, sdB = 1, beta = c(8, 1, 0,
0), alpha = 0.05) {

  # generate data
  Patient <- as.factor(rep(1:(2 * n), rep(2, 2 * n)))
  Treatment <- c(rep(c("Treatment1", "Treatment2"), n), rep(c("Treatment2",
    "Treatment1"), n))
  Order <- rep(c("First", "Second"), 2 * n)
  Data <- data.frame(Patient, Treatment, Order)
  FMat <- model.matrix(~Treatment * Order, data = Data)
  RMat <- model.matrix(~0 + Patient, data = Data)
  Response <- FMat %*% beta + RMat %*% rnorm(2 * n, 0, sdB) +
    rnorm(4 * n, 0, sdW) # beta here
  Data$Response <- Response

  return(data.frame(Data))
  # analyse Fit <- lme(Response~ Treatment * Order, random=~1 |
  # Patient, data=Data, na.action='na.omit' ) Est <-
  # fixed.effects(Fit)[2] Ste <- sqrt(vcov(Fit)[2,2]) prod(Est
  # + c(-1,1) * qnorm(1-alpha/2) * Ste) > 0
}

```

Generate data and analyse

```
mydata <- ABBA(n = 10, sdW = 2, sdB = 2, beta = c(8, 1, 0, 0),
  alpha = 0.05)

require(nlme)
f <- lme(Response ~ Treatment * Order, random = ~1 | Patient,
  data = mydata, na.action = "na.omit")
anova(f)
```

	numDF	denDF	F-value	p-value
(Intercept)	1	19	421.0340	<.0001
Treatment	1	17	0.5352	0.4744
Order	1	17	0.9612	0.3406
Treatment:Order	1	17	0.4587	0.5073

```
summary(f)$tTable
```

	Value	Std.Error	DF
(Intercept)	8.023633	0.7468591	19
TreatmentTreatment2	1.030554	1.0562182	17
OrderSecond	1.182608	1.0562182	17
TreatmentTreatment2:OrderSecond	-1.166955	1.7229741	17

	t-value
(Intercept)	10.7431685
TreatmentTreatment2	0.9757021
OrderSecond	1.1196628
TreatmentTreatment2:OrderSecond	-0.6772910

	p-value
(Intercept)	0.000000001638637
TreatmentTreatment2	0.342892860343766
OrderSecond	0.278427969007660
TreatmentTreatment2:OrderSecond	0.507334457267561

Simulation function for crossover design, preparing to vary an input parameter

```

C.power <- function(a, b, c, d, e, f, g, n.sims = sims) {

  Treatment <- rep(NA, n.sims)
  Order <- rep(NA, n.sims)
  Interaction <- rep(NA, n.sims)

  for (s in 1:n.sims) {

    fake <- ABBA(n = a, sdW = b, sdB = c, beta = c(d, e,
      f, g), alpha = 0.05)

    possibleError <- tryCatch(f1 <- lme(Response ~ Treatment *
      Order, random = ~1 | Patient, data = fake, na.action = "na.omit"),
      error = function(e) e)

    ### http://stackoverflow.com/questions/8093914
    ### /skip-to-next-value-of-loop-upon-error-in-r-trycatch

    if (!inherits(possibleError, "error")) {

      modelint <- possibleError

      z <- as.matrix(summary(modelint)$tTable)
      Treatment[s] <- z[2, 5][[1]]
      Order[s] <- z[3, 5][[1]]
      Interaction[s] <- z[4, 5][[1]]
    }
  }

  trt <- mean(Treatment < 0.05)
  ord <- mean(Order < 0.05)
  int <- mean(Interaction < 0.05)

  c(trt, ord, int)
}

# stand alone, single example that shows power for main
# effect, order effect and interaction effect
C.power(a = 20, b = 1, c = 1, d = 8, e = 1, f = 0, g = 0, n.sims = 100)

```

```
[1] 0.60 0.08 0.10
```

Simulate crossover power and vary between person SD

```
## execute the functions numerous times assess changing
## between person SD

sims <- 200

J <- c(2, 2.5, 3, 4) # here we vary

k <- length(J)

dnam = list(betweenSD = J, Estimate = c("trt effect power", "order power",
    "interaction power"))

pwpr <- array(NA, dim = sapply(dnam, length), dimnames = dnam)

system.time(for (i in 1:k) {
  pwpr[i, ] <- C.power(a = 20, b = 2, c = J[i], d = 8, e = 4,
    f = 0, g = 0, n.sims = sims)
})

  user  system elapsed
14.40    0.00   14.43

print(pwpr, digits = 4)
```

	Estimate			
betweenSD	trt effect power	order power	interaction power	
2	0.990	0.050	0.055	
2.5	0.975	0.035	0.030	
3	0.920	0.035	0.030	
4	0.785	0.030	0.040	

Simulate crossover power and vary within person SD

```

sims <- 200

J <- c(2, 2.5, 3, 4) # here we vary

k <- length(J)

dnam = list(betweenSD = J, Estimate = c("trt effect power", "order power",
    "interaction power"))

pwpr <- array(NA, dim = sapply(dnam, length), dimnames = dnam)

system.time(for (i in 1:k) {
  pwpr[i, ] <- C.power(a = 20, b = J[i], c = 4, d = 8, e = 4,
    f = 1, g = 0, n.sims = sims)
})

```

```

      user  system elapsed
13.78    0.00    13.81

```

```
print(pwpr, digits = 4)
```

	Estimate			
betweenSD	trt effect power	order power	interaction power	
2		0.820	0.110	0.035
2.5		0.795	0.070	0.030
3		0.615	0.065	0.055
4		0.515	0.075	0.045

```

mydata <- ABBA(n=10, sdW=2, sdB=2, beta=c(8, 1, 0, 0), alpha=0.05 )
par(mfrow=c(3,1))
ggpubr::ggpaired(mydata , x="Treatment", y = "Response",
                  color = "Treatment", line.color = "gray", line.size = 0.4,
                  palette = "npg", title="Pooled across period comparison")+
stat_compare_means(paired = TRUE)

ggpubr::ggpaired(mydata[mydata$Order=="First",], x="Treatment", y = "Response",
                  color = "Treatment", line.color = "gray", line.size = 0.4,
                  palette = "npg", , title="First period comparison")+
stat_compare_means(paired = TRUE)

ggpubr::ggpaired(mydata[mydata$Order=="Second",], x="Treatment", y = "Response",
                  color = "Treatment", line.color = "gray", line.size = 0.4,
                  palette = "npg" , title="Second period comparison")+
stat_compare_means(paired = TRUE)
par(mfrow=c(1,1))

```

<http://www.surefoss.org/visualisation/visualizing-small-scale-paired-data-combining-boxplots-stripchart>

```

co.plot<-function (mydata){

  par(mfrow=c(2,2))

  mydata$temp <- paste(mydata$Treatment, mydata$Order, sep=".")

  pre<-as.vector(mydata[grepl("Treatment1.First", mydata$temp), 4 ])
  post<-as.vector(mydata[grepl("Treatment1.Second", mydata$temp), 4 ])
  s<-seq(length(pre))
  par(bty="l")

  boxplot(pre,post,main="Treatment1->Treatment2",xlab="Order",ylab="Sig Score",
           names=c("Treatment1", "Treatment2"),col=c("lightgreen","lightblue"))
  stripchart(list(pre,post),vertical=T,pch=16,method="jitter",cex=0.5,add=T)
  segments(rep(0.95,length(pre))[s],pre[s],rep(2,length(pre))[s],post[s],col=1,lwd=0.5)

  res<-wilcox.test(post,pre,paired=T,conf.int=T)
  stripchart(post-pre,vertical=T,pch=16,method="jitter",main="Difference",
             ylab="Difference:Treatment2-Treatment1",xlab="Median+/-95%CI")
  points(1,res$estimate,col="red",pch=16,cex=2)
  arrows(1,res$conf.int[1],1,res$conf.int[2],col="red",code=3,lwd=3,angle=90)
  abline(h=0,lty=2)#Zero-effectline

  pre<-as.vector(mydata[grepl("Treatment2.First", mydata$temp), 4 ])
  post<-as.vector(mydata[grepl("Treatment2.Second", mydata$temp), 4 ])
  s<-seq(length(pre))
  par(bty="l")

  boxplot(pre,post,main="Treatment2->Treatment1",xlab="Order",ylab="Sig Score",
           names=c("Treatment2", "Treatment1"),col=c("lightblue","lightgreen"))
  stripchart(list(pre,post),vertical=T,pch=16,method="jitter",cex=0.5,add=T)
  segments(rep(0.95,length(pre))[s],pre[s],rep(2,length(pre))[s],post[s],col=1,lwd=0.5)
}

```



```

res<-wilcox.test(post,pre,paired=T,conf.int=T)
stripchart(post~pre,vertical=T,pch=16,method="jitter",main="Difference",
           ylab="Difference:Treatment1-Treatment2",xlab="Median+/-95%CI")
points(1,res$estimate,col="red",pch=16,cex=2)
arrows(1,res$conf.int[1],1,res$conf.int[2],col="red",code=3,lwd=3,angle=90)
abline(h=0,lty=2)#Zero-effectline
#detach(mydata)
par(mfrow=c(1,1))

}

x<- ABBA(n=50, sdW=2, sdB=2, beta=c(8, 4, 1, 0), alpha=0.05 )
co.plot( x)

#####
#####PLOT 1#####
#####

mydata<- ABBA(n=50, sdW=2, sdB=2, beta=c(8, 4, 1, 0), alpha=0.05 )

p4 <- function(x) {formatC(x, format="f", digits=4)}

d1<-mydata
require(reshape)
d1$grp<-paste(d1$Treatment, d1$Order, sep=" ")
d1 <- rename(d1, c(Response="count"))
d1 <- rename(d1, c(grp="spray"))
d1 <- d1[order(d1$spray),]

attach(d1)
sprayTypes <- unique(spray)

y<-as.numeric(as.character(d1[,4]))

plot(y, as.factor(d1[,5]) , ylim=c(min(y),max(y)) , xlim=c(1, 5) , xaxt="n",
     main=paste("\nTruth: Intercept (TRT2 mean) ",beta[1]," , TRT (main) effect ",beta[2]," , \nSecond scan effect ",beta[3],
               ", \n Interaction (TRT only second scan effect)",beta[4],
               ", \n Between subjects SD",sb," , within subjects SD", p4(sw) ),
     xlab="", ylab="Outcome measure", frame.plot =F , col="white")
##if not white I get a nasty boxplot

axis(1, at=1:4, labels=F )

text(x=1:length(sprayTypes)*1.05, par("usr")[3]-0.08 , labels = sprayTypes,
     srt = 45, pos = 2, xpd = TRUE,
     cex.axis=.5 )

##make colours
chk<-as.character(d1$spray)
x<-as.data.frame(table(chk))
freq<-x[,2]

```

```

value<-max(dput(1:dim(x)[1]))
IR<-value+1
clr<-rainbow(IR)
clr<-c("blue","red")
wR<-(2*IR-1:IR)/(2*IR)
##colour made
for (i in 1 : length(sprayTypes)){
  y <- count[spray == sprayTypes[i]]
  n <- sum(spray == sprayTypes[i])
  points(jitter(rep(i, n), amount = .1), y, pch = 16, cex = .5,

        col=ifelse(i<3,clr[1],clr[2]))

)

col=ifelse(Treatment=="Treatment1",clr[1],clr[2])

lines(i + c(.12, .28), rep(mean(y), 2), lwd = 1, col="black")
lines(rep(i + .2, 2),
      mean(y) + c(-1.96, 1.96) * sd(y) / sqrt(n), lwd = 1, col="black"

)
}

detach(d1)
#####
#####
#####

```

Function to simulate crossover trial using nlme

Computing Environment

```
sessionInfo()
```

```
R version 3.6.1 (2019-07-05)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 17134)
```

```
Matrix products: default
```

```
locale:
[1] LC_COLLATE=English_United Kingdom.1252
[2] LC_CTYPE=English_United Kingdom.1252
[3] LC_MONETARY=English_United Kingdom.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.1252
```

```
attached base packages:
[1] stats      graphics  grDevices  utils      datasets
[6] methods    base
```

```
other attached packages:
[1] nlme_3.1-140 knitr_1.23
```

```
loaded via a namespace (and not attached):
[1] compiler_3.6.1 magrittr_1.5   formatR_1.7
[4] tools_3.6.1     htmltools_0.3.6 yaml_2.2.0
[7] Rcpp_1.0.1      stringi_1.4.3 rmarkdown_1.14
[10] grid_3.6.1      stringr_1.4.0 xfun_0.8
[13] digest_0.6.20   lattice_0.20-38 evaluate_0.14
```

```
This took 30.36 seconds to execute.
```