MRC Biostatistics Unit

UNIVERSITY OF CAMBRIDGE

Bayesian Methods for Clinical Trials

by Libby Daniells & Pavel Mozgunov & Thomas Jaki

February 25, 2025

Practical 2: Fixed-sample designs & JAGS

You are asked to design a Phase II singe-agent PoC clinical trial of a new anti-cancer agent. The team would like to evaluate a particular dose of the new agent in terms of the objective response rate (ORR) at Week 12 after the start of the treatment. The uninteresting response rate that would not suggest a promising activity is $p_0 = 0.30$ and an interesting treatment effect that would trigger the further development of the compound is $p_1 = 0.50$.

> Let Y be the number of objective responses. $H_0 : p \leq p_0$ is tested against $H_1 : p > p_0$. The treatment is deemed efficacious if $Y \geq u$ for a suitable u. Then n and u are chosen such that:
>
> $$\mathbb{P}(Y \geq u; p_0) \leq \alpha \quad \text{and} \quad \mathbb{P}(Y \geq u; p_1) \geq 1 - \beta.$$

(a) Given that the trial consists of $n = 40$ patients and that $u = 16$, write out the expressions for the type I error and power for the above trial in terms of the binomial distribution. Then write a function in R that takes in the values of n, u, p_0 and p_1 and returns the type I error and power. *(Hint: you will need the pbinom() function)*

Using the constraints in the box above and the fact that responses are binary, the probabilities can be expressed as a sum over the binomial distributions.

$$\Pr(Y \geq u; p_0 = 0.30) = \sum_{i=u}^{n} \binom{n}{i} p_0^i (1 - p_0)^{n-i}, \tag{1}$$

$$\Pr(Y \geq u; p_1 = 0.50) = \sum_{i=u}^{n} \binom{n}{i} p_1^i (1 - p_1)^{n-i}. \tag{2}$$

```
error_power <- function(n,u,p0,p1){
    Error <- pbinom(u-1,size=n,prob=p0,lower.tail=F)
    Power <- pbinom(u-1,size=n,prob=p1,lower.tail=F)
return(data.frame(Error,Power))
}

error_power(40,16,0.3,0.5)
```

```
   error   power
> 0.1151467 0.92307
```

(b) We will now compute the sample size n (and an associate critical value, u) that achieves a statistical power of 80% of finding this treatment efficacious at the one-sided 5% significance level.

(i) Use a for loop and your function from (a) to do an exhaustive search over a number of plausible values of n and u and find the values that satisfy the power and error constraints. What is the smallest sample size required?
Hint: to define the combinations of u and n to search over you can use the following code

```
combo <- data.frame(nb=rep(1:maxN,1:maxN),ub=sequence(1:maxN))
```

where maxN *is a maximum sample size to be defined.*

```
alpha <- 0.05
power <- 0.8
maxN <- 50
combo <- data.frame(nb=rep(1:maxN,1:maxN),ub=sequence(1:maxN))
Myprobs <- array(0,c(nrow(combo),2))

for(i in 1:nrow(combo)){
  Myprobs[i,1] <- error_power(combo[i,1],combo[i,2],0.3,0.5)$Error
  Myprobs[i,2] <- error_power(combo[i,1],combo[i,2],0.3,0.5)$Power
}

data.frame(
  combo[which(Myprobs[,1] < alpha & Myprobs[,2] > power),],
  pprob = Myprobs[which(Myprobs[,1] < alpha & Myprobs[,2] > power),])

      n  u     prob.1    prob.2
758  39 17 0.04998419 0.8316082
838  41 18 0.04135975 0.8255556
922  43 19 0.03423413 0.8198112
965  44 19 0.04371645 0.8543924
1010 45 20 0.02834511 0.8143510
```

Thus, the smallest sample size that satisfies the power and alpha constraints is 39, with H_0 being rejected if $Y \geq 17$

(ii) Give a brief explanation of why these values are not exactly 5% and 80% as targeted?

As our data is discrete, it is not possible to achieve the exact probabilities. For example, we saw previously that $n = 39$ and $u = 17$ gives a type I error of 0.04998, which is slightly conservative. If we were to lower $u = 16$ to be less conservative, this probability is 0.09441. There is no middle ground between the two values so using $u = 17$ is as close as we'll get to 5%.

(c) We will now consider a Bayesian design wherein efficacy decisions are made based on posterior probabilities, comparing the posterior response rate to the null and target response rates.

The treatment is deemed efficacious if $\mathbb{P}(p > p_0 | Y = u) \geq \eta$, where $\eta = 0.95$ is chosen to be close to 1. Any values of Y greater than u will also be deemed efficacious.

The treatment is abandoned if $\mathbb{P}(p < p_1 | Y = u - 1) \geq \zeta$, where $\zeta = 0.8$ is chosen to be close to 1. Any values of Y less than u will also be abandoned.

Thus, as before, the treatment is deemed efficacious if $Y \geq u$.

(i) Write a function in R that computes the two probabilities: $\mathbb{P}(p > p_0 | Y = u)$ and $\mathbb{P}(p < p_1 | Y = u - 1)$ given the prior $p \sim \text{Beta}(a, b)$.

```
BayesianConstraints <- function(n,u,p0,p1,a,b){
  C1 <- pbeta(p0,shape1=a+u,shape2=b+n-u,lower.tail=F)
  C2 <- pbeta(p1,shape1=a+u-1,shape2=b+n-(u-1))
  return(data.frame(C1,C2))
}
```

(ii) Using a for loop and your function from (c)(i), do an exhaustive search over a number of plausible values of n and u and find the values that satisfy the two probability constraints, implementing the following priors:

$$p \sim \text{Beta}(1,1); \quad p \sim \text{Beta}(10,10); \quad p \sim \text{Beta}(15,15); \quad p \sim \text{Beta}(5,15).$$

What is the smallest sample size required under each prior?

```
eta <- 0.95
zeta <- 0.8

Myprobs11 <- array(0,c(nrow(combo),2))
for(i in 1:nrow(combo)){
  Myprobs11[i,1] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=1,b=1)$C1
  Myprobs11[i,2] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=1,b=1)$C2
}
data.frame(
  combo[which(Myprobs11[,1] >= eta & Myprobs11[,2] >= zeta),],
  pprob = Myprobs11[which(Myprobs11[,1] >= eta & Myprobs11[,2] >= zeta),])

     nb ub   pprob.1   pprob.2
363  27 12 0.9508962 0.8275358
```

```
419  29 13 0.9599475 0.8192027
479  31 14 0.9672763 0.8114572
510  32 14 0.9562608 0.8518968
543  33 15 0.9732247 0.8042358
...

Myprobs1010 <- array(0,c(nrow(combo),2))
for(i in 1:nrow(combo)){
  Myprobs1010[i,1] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=10,b=10)$C1
  Myprobs1010[i,2] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=10,b=10)$C2
}
data.frame(
  combo[which(Myprobs1010[,1] >= eta & Myprobs1010[,2] >= zeta),],
  pprob = Myprobs1010[which(Myprobs1010[,1] >= eta & Myprobs1010[,2] >= zeta),])

     nb ub   pprob.1   pprob.2
39    9  3 0.9508962 0.8275358
59   11  4 0.9599475 0.8192027
83   13  5 0.9672763 0.8114572
96   14  5 0.9562608 0.8518968
111  15  6 0.9732247 0.8042358
...

Myprobs1515 <- array(0,c(nrow(combo),2))
for(i in 1:nrow(combo)){
  Myprobs1515[i,1] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=15,b=15)$C1
  Myprobs1515[i,2] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=15,b=15)$C2
}
data.frame(
  combo[which(Myprobs1515[,1] >= eta & Myprobs1515[,2] >= zeta),],
  pprob = Myprobs1515[which(Myprobs1515[,1] >= eta & Myprobs1515[,2] >= zeta),])

     nb ub   pprob.1   pprob.2
11    5  1 0.9732247 0.8042358
16    6  1 0.9641178 0.8447477
22    7  1 0.9529618 0.8785075
30    8  2 0.9705325 0.8379957
38    9  2 0.9612451 0.8720625
...

Myprobs515 <- array(0,c(nrow(combo),2))
```

```
for(i in 1:nrow(combo)){
  Myprobs515[i,1] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=5,b=15)$C1
  Myprobs515[i,2] <- BayesianConstraints(combo[i,1],combo[i,2],0.3,0.5,a=5,b=15)$C2
}
data.frame(
  combo[which(Myprobs515[,1] >= eta & Myprobs515[,2] >= zeta),],
  pprob = Myprobs515[which(Myprobs515[,1] >= eta & Myprobs515[,2] >= zeta),])

       nb ub   pprob.1   pprob.2
44    9  8 0.9508962 0.8275358
64   11  9 0.9599475 0.8192027
88   13 10 0.9672763 0.8114572
101  14 10 0.9562608 0.8518968
116  15 11 0.9732247 0.8042358
...
```

(iii) Describe how the prior affects sample size determination. Is it the more informative the better?

From the reported n and u above, we observe that a smaller sample size is required to give the same power and error when using a more informative prior distribution. Use of an informative prior should be carefully justified, especially that the information represented in the prior is consistent with the new data. Otherwise, one could be 'precisely wrong' and possibly need more new data to 'correct' the prior understanding. Comparing the Beta(10,10) prior and Beta(5,15) prior, both give the same n, but the prior that puts more mass around the null value, Beta(5,15), requires all but one patient to respond to the treatment for it to be deemed efficacious, whereas the Beta(10,10) prior requires just 3 responses out of the 9 patients.

(d) Simulate a dataset using the code below.

```
    set.seed(100)
    MyDat1 <- rbinom(n = 49, size = 1, prob = 0.50)
```

(i) Write a JAGS model for this data with a vague prior $p \sim \text{Uniform}(0, 1)$. Run the model and plot the prior and posterior distributions.

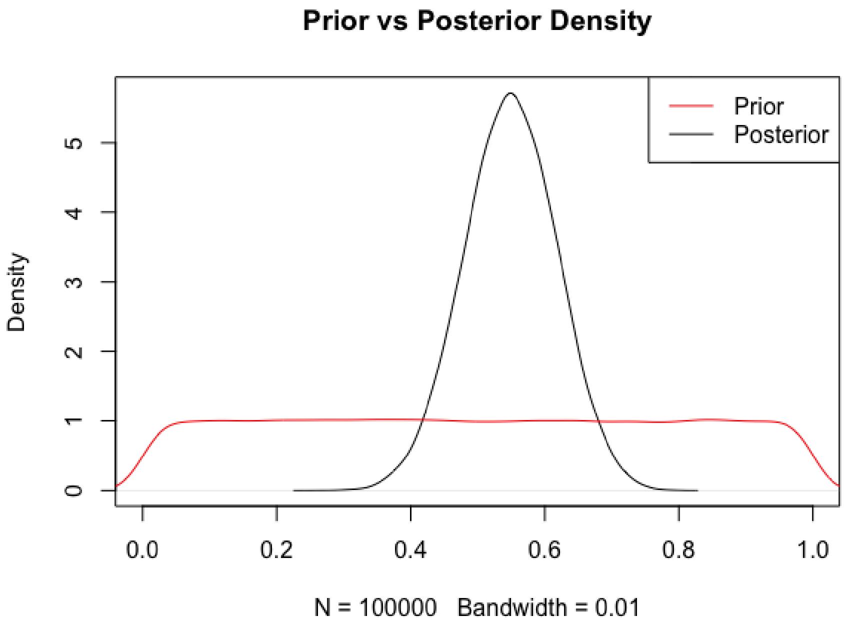
```
library(rjags)

model1 <- "model{
  for(i in 1:length(Y)){
      Y[i] ~ dbern(p)
  }
  # prior
  p ~ dunif(0, 1)
}"


data_list <- list(Y = MyDat1)
iter <- 100000
model1.spec <- textConnection(model1)
jags <- jags.model(model1.spec, data = data_list,
                   n.chains = 1,  n.adapt = iter)
update(jags, iter, progress.bar = "none")
mcmc.sampling <- jags.samples(jags, c("p"), iter, progress.bar = "none")
samples.p <- mcmc.sampling$p[1,,]

data_list_prior <- list(Y = NA)
model1.spec <- textConnection(model1)
jags_prior <- jags.model(model1.spec, data = data_list_prior,
                   n.chains = 1,  n.adapt = iter)
update(jags_prior, iter, progress.bar = "none")
mcmc.sampling_prior <- jags.samples(jags_prior, c("p"), iter, progress.bar = "none")
samples.p_prior <- mcmc.sampling_prior$p[1,,]

plot(density(samples.p,bw=0.01),xlim=c(0,1),main='Prior vs Posterior Density')
lines(density(samples.p_prior),col='red')
legend('topright',legend=c('Prior','Posterior'),col=c('red','black'),lty=rep(1,2))
```



(ii) Given that we establish treatment benefit if $\Pr(p > 0.3 \mid \boldsymbol{x}) \geq 95\%$, otherwise claim the treatment as not efficacious. What conclusion would you draw following your analysis of `MyDat1`? *Hint: You can include the probability computation in the jags model using step(x) which computes the probability that $x > 0$. You can then extract this probability from the jags output.*

```
library(rjags)

model1 <- "model{
  for(i in 1:length(Y)){
      Y[i] ~ dbern(p)
  }

  pCat <- step(p - p0)

  # prior
  p ~ dunif(0, 1)
}"

data_list <- list(
  Y = MyDat1,
  p0 = 0.3
)

iter <- 10000
model1.spec <- textConnection(model1)
jags <- jags.model(model1.spec, data = data_list,
                   n.chains = 1,  n.adapt = iter)

update(jags, iter, progress.bar = "none")
mcmc.sampling <- jags.samples(jags, c("p", "pCat"), iter, progress.bar = "none")

samples.p <- mcmc.sampling$p[1,,]
samples.pCat <- mcmc.sampling$pCat[1,,]

> mean(samples.p); sd(samples.p)
[1] 0.5487675
[1] 0.06932363
> mean(samples.pCat)
[1] 0.9999
```

The probability is 0.9999 which is greater than 0.95 and hence the treatment is efficacious.

(iii) Sticking with the Uniform prior, write a function in R (using JAGS) that computes the two probabilities: $\mathbb{P}(p > p_0 | Y = u)$ and $\mathbb{P}(p < p_1 | Y = u - 1)$.

(Hint: You can either run two jags models: one to compute the first probability and one to compute the second or do this in a single model. For a single model you will need to specify the two values of Y (i.e. u and $u-1$) and their distributions, with separate p parameters and priors in order to avoid a hierarchy structure.)

```
        model1 <- "model{
Y1 ~ dbinom(p_eta,n)
Y2 ~ dbinom(p_zeta,n)

pCateta <- step(p_eta - p0)
pCatzeta <- step(p1-p_zeta)

# prior
p_eta ~ dunif(c, d)
p_zeta ~ dunif(c , d)
}"

JAGSssUNIF <- function(n,u,p0,p1,c,d){
  iter <- 100000
  data_list1 <- list(Y1 = u, Y2=u-1, n=n, p0 = p0, p1 = p1, c=c, d=d)

  model1.spec <- textConnection(model1)
  jags <- jags.model(model1.spec, data = data_list1,
                     n.chains = 1, n.adapt = iter)
  update(jags, iter,progress.bar='none')
  mcmc.sampling <- jags.samples(jags, c("pCateta","pCatzeta"),
                        iter, progress.bar = "none")

  samples.pCateta <- mcmc.sampling$pCateta[1,,]
  P1 <- mean(samples.pCateta)
  samples.pCatzeta <- mcmc.sampling$pCatzeta[1,,]
  P2 <- mean(samples.pCatzeta)
  return(list('P1'=P1,'P2'=P2))
}
```

(iv) Using a for loop and your function from (d)(iii), do an exhaustive search over a number of plausible values of n and u and find the values that satisfy the two probability constraints $\mathbb{P}(p > p_0|Y = u) \geq \eta$, where $\eta = 0.95$ and $\mathbb{P}(p < p_1|Y = u-1) \geq \zeta$, where $\zeta = 0.8$. Implement the following two priors:

$$p \text{ Uniform}(0,1); \quad p \sim \text{Uniform}(0.25, 0.75).$$

```
maxN <- 50
```

```
eta <- 0.95
zeta <- 0.8
combo <- data.frame(nb=rep(1:maxN,1:maxN),ub=sequence(1:maxN))

Myprobs01 <- array(0,c(nrow(combo),2))
for(i in 1:nrow(combo)){
  Myprobs01[i,1] <- JAGSssUNIF(combo[i,1],combo[i,2],0.3,0.5,c=0,d=1)$P1
  Myprobs01[i,2] <- JAGSssUNIF(combo[i,1],combo[i,2],0.3,0.5,c=0,d=1)$P2
}

data.frame(
  combo[which(Myprobs01[,1] >= eta & Myprobs01[,2] >= zeta),],
  pprob = Myprobs01[which(Myprobs01[,1] >= eta & Myprobs01[,2] >= zeta),])

     nb ub pprob.1 pprob.2
363  27 12 0.95231 0.82618
419  29 13 0.95932 0.81679
479  31 14 0.96676 0.81227
510  32 14 0.95637 0.85173
543  33 15 0.97386 0.80266

Myprobs025075 <- array(0,c(nrow(combo),2))
for(i in 1:nrow(combo)){
  Myprobs025075[i,1] <- JAGSssUNIF(combo[i,1],combo[i,2],0.3,0.5,c=0.25,d=0.75)$P1
  Myprobs025075[i,2] <- JAGSssUNIF(combo[i,1],combo[i,2],0.3,0.5,c=0.25,d=0.75)$P2
}

data.frame(
  combo[which(Myprobs025075[,1] >= eta & Myprobs025075[,2] >= zeta),],
  pprob = Myprobs025075[which(Myprobs025075[,1] >= eta & Myprobs025075[,2] >= zeta),])

    nb ub pprob.1 pprob.2
311 25 11 0.95507 0.82949
363 27 12 0.96255 0.82282
419 29 13 0.96750 0.81806
448 30 13 0.95808 0.85386
479 31 14 0.97182 0.80765
```

The values of n and u selected using the JAGS simulation and a Uniform(0,1) prior matches those from (c) using a $p \sim \text{Beta}(1, 1)$ prior.