Moxonidine Trial

Introduction

This report outlines a clinical trial which can be found here: https://pubmed.ncbi.nlm.nih.gov/17968979/.
The trial compared the drug moxonidine against a placebo in a double-blind trial.

Background

Myocardial ischaemia is the leading cause of morbidity and mortality after surgery on patients with coronary
artery disease. Patients were to be randomly assigned the drug moxonidine or a placebo to be taken the
morning before surgery and on days 1 to 4 after surgery. Study medication was administered by a member of
the team blinded to treatment allocation. Levels of cTnI were measured the day before surgery, immediately
after surgery and on days 1, 2, 3 and 7 after surgery. Based on historical data, an increase of cTnI of 2μg/l
was interpreted as being clinically significant (an increase of cTnI is related to a lower incidence of myocardial
ischaemia).

Trial Design

Based on an incidence of myocardial ischaemia of 45% found in a previous investigation, an estimated effect
size of 33% and the trial being performed at 80% power, the necessary sample size was calculated to be 180
patients per group. We can use equation 9.18 in Julious (2009) to show why the trial chose 180 patients in
each arm.

$$n_A = \frac{4}{(\pi_A - \pi_B)^2},$$

Here, we have $\pi_A = 0.45$ and $\pi_B = 0.5985$. If we substitute these values into the equation we obtain:

$$\begin{aligned}
n_A &= \frac{4}{(\pi_A - \pi_B)^2} \\
&= \frac{4}{(0.45 - 0.5985)^2} \\
&= \frac{4}{0.022} \\
&\approx 181.8,
\end{aligned}$$

so we can see why the value of 180 patients in each arm was chosen.

A blinded interim analysis was prospectively planned after one-third of the calculated study population had
been enrolled.

Results

At the interim analysis, 141 patients had been recruited into the trial. There were 78 patients in the
treatment group and 63 patients in the placebo group. The incidence of raised perioperative cTnI levels in
each groups were as follows: 31 in treatment, 23 in placebo. This corresponds to a proportion of 40% in the

treatment group and 37% in the placebo group. We can see that the levels of cTnI in the treatment group are slightly higher, but only 3%. This was analysed with a χ^2 test and the p-value calculated is 0.694, which clearly is not significant at all. Therefore, the trial was stopped due to futility (the treatment is clearly not clinically better so there is no advantage to continuing the trial until the end).

Assurance for this trial

The design of this trial is very traditional, very frequentist. We have a placebo (for which we have a point-estimate), a treatment (for which we have an estimate of its effects in comparison to the placebo), we have the type I error rate ($\alpha = 0.05$) and we have some power that we wish to perform the trial at (here, $1 - \beta = 0.8$). As a result, we saw how when we substitute these values into the equations, we obtain 180 patients in both of the two arms.

However, this method is sub-optimal for a number of reasons. The first reason is that this calculation assumes that the treatment is in fact better than the placebo - which we cannot possibly know as we would not be doing a trial otherwise. Therefore, the power of the test (80%) is the conditional probability that we will find an effect, if such an effect exists. In reality, there will be times where the treatment is no better than the placebo (as was the case in this trial) so the power is a misrepresentation of how effective a trial actually is.

The second reason is that we only have point-estimates for the placebo and the treatment effect. From historical data, we should be able to construct a distribution for the effectiveness of the placebo, and from possible phase II trials, pilot studies and in collaboration with clinical investigators, we should be able to construct a distribution for the effectiveness of the treatment. Even if we are not very confident about our estimates in the effectiveness of the treatment, these prior beliefs can be incorporated into the distribution.

Trial setup

The outcome is a binary variable, we have r_i successes out of n_i patients for both treatments ($i = 1$ for the placebo, $i = 2$ for the treatment). Therefore, the hypothesis test we conduct at the end of the trial is:

$$H_0 = \theta_1 = \theta_2$$
$$H_1 = \theta_1 \neq \theta_2$$

Therefore, to calculate assurance methods for this trial design, we require prior beliefs regarding θ_1 and θ_2. From the information in the trial, we know that $E[\theta_1] = 0.45$. The trial investigators will be able to calculate the SD of θ_1 accurately, but just for illustration, we will choose $\text{sd}[\theta_1] = 0.1$. We can find a Beta distribution which captures these beliefs by:

```
estBetaParams <- function(mu, var) {
  alpha <- ((1 - mu) / var - 1 / mu) * mu ^ 2
  beta <- alpha * (1 / mu - 1)
  return(params = list(alpha = alpha, beta = beta))
}

estBetaParams(0.45, 0.1^2)
```

```
$alpha
[1] 10.6875
##
$beta
[1] 13.0625
```

So we can see that these beliefs can be approximately captured by a Be(10.7, 13) distribution. We can repeat this process for $\theta_2$. From the trial setup, we know that the investigators believe $E[\theta_2] = 0.5985$. Again, the investigators will know more, but we can choose $\text{sd}[\theta_2] = 0.2$.

```
estBetaParams(0.5985, 0.2^2)
```

```
## $alpha
## [1] 2.996955
##
## $beta
## [1] 2.010489
```

As before, When we fit these parameters to a Beta distribution, we approximately obtain a Be(3, 2) distribution.

We can plot θ_1 and θ_2 to ensure the distributions capture our beliefs correctly.
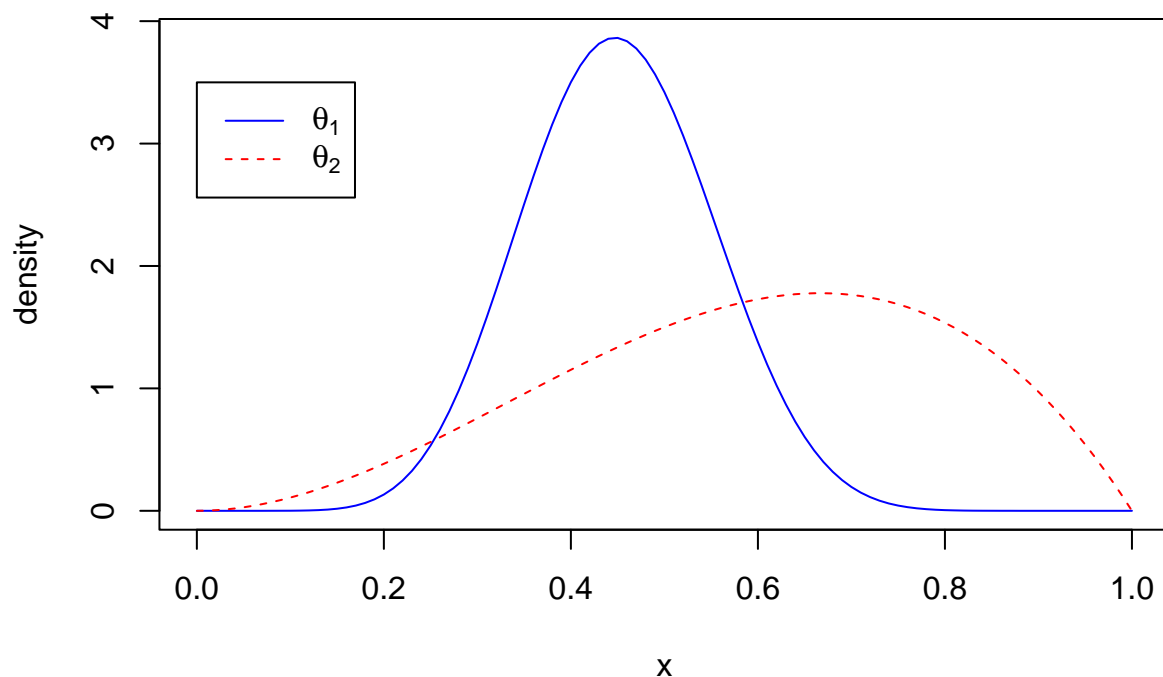
From the plot we can see that the distributions seem to give a good fit to our intuition. We are more sure in our beliefs of θ_1, this is reflected in more of the density being centred around 0.45 - which is what we believe to be our most likely value. For θ_2, we are more unsure in our beliefs and this is reflected in the density. A much wider range of values are likely, but the median/mode is higher than θ_1 which is what we require, as we believe that the treatment has some positive effect on outcomes.

##An Aside

We have looked at Example 4 in O'Hagan (2005): https://onlinelibrary.wiley.com/doi/epdf/10.1002/pst.175 and worked it through, using the Appendix. In the Appendix there is code given for the example, the code calculates assurance for the example. The code is as follows:

```
library(rjags)
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
data = list(n1=200, n2=400, zsig=1.96)
modelstring="
model{
  theta1~dbeta(5,20)
  theta2 <- (eff * theff)+((1-eff)*thneff) # These lines set out
  eff~dbern(0.85) # the prior as in Example 4.
  theff~dbeta(3,4.5) # They should be modified
  thneff~dbeta(2, 23) # as appropriate for any real example.
  r1~dbin(theta1, n1)
  r2~dbin(theta2, n2)
  p1 <-  r1/n1
  p2 <- r2/n2
  se <-  sqrt((p1*(1-p1)/n1)+(p2*(1-p2)/n2))
  z <-  (p2-p1)/se
  ass <- step(z-zsig) # The mean of ass is the assurance
}
"
model = jags.model(textConnection(modelstring), data = data)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 6
##    Total graph size: 35
##
## Initializing model
```

```
update(model, n.iter = 1000)
output = coda.samples(model=model, variable.names = c("ass"), n.iter = 1000000)
mean(output[[1]])
```

```
## [1] 0.632947
```

The authors calculated assurance to be $\gamma = 0.635$ and we can see that our code replicates this well (with MCMC error).

#Code to calculate assurance

As a result of the aside, we can modify the code above to be applicable in our trial. We have changed the sample sizes in both arms and the prior distributions on the θ_i's. The code is as follows:

```r
library(rjags)
data = list(n1=180, n2=180, zsig=1.96)
modelstring="
model{
  theta1~dbeta(10.7, 13)
  theta2~dbeta(3, 2)
  r1~dbin(theta1, n1)
  r2~dbin(theta2, n2)
  p1 <-  r1/n1
  p2 <- r2/n2
  se <-  sqrt((p1*(1-p1)/n1)+(p2*(1-p2)/n2))
  z <-  (p2-p1)/se
  ass <- step(z-zsig) # The mean of ass is the assurance
}
"
model = jags.model(textConnection(modelstring), data = data)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 4
##    Total graph size: 26
##
## Initializing model
```

```r
update(model, n.iter = 1000)
output = coda.samples(model=model, variable.names = c("ass"), n.iter = 1000000)
mean(output[[1]])
```

```
## [1] 0.592791
```

We can see that assurance is 0.59 or 59% in this case. This seems fairly high for an assurance calculation, one reason for this could be the overestimation of how well the treatment actually works (the elicitation for θ_2). We also chose the SD of the priors without any prior knowledge, whereas in practice the investigatory team would be able to give more accurate predictions for the SD's.