Data Analysis Project

Marcel Vaz

November 26, 2016

Introduction

This study has the objective to test whether the new method of implanting stents (PCI) to fix patients with coronary obstructions outperforms the old one (CABG). To this date, there are at least eleven studies that compared mortality rates one year after the treatment for both types of surgical intervention. Here, I will use these studies to perform a meta-analysis, thus fulfilling this paper's objective.

Methods

The problem was modeled as below:

$$y_{ij}|\pi_{ij} \sim Binomial(n_{ij}, \pi_{ij}) \tag{1}$$

$$logit\pi_{ij} = \begin{cases} \mu_i - \delta_i/2 & \text{if } j = 0 \\ \mu_i + \delta_i/2 & \text{if } j = 1 \end{cases} \tag{2}$$

$$\mu_i|\mu_0, \sigma^2 \sim N(\mu_0, \sigma^2) \tag{3}$$

$$\delta_i|\delta_0, \tau^2 \sim N(\delta_0, \tau^2) \tag{4}$$

Where the observed casualties y in the i^{th} study in the j^{th} treatment are modeled as random draws from a binomial distribution (Eq.1) with sample size n and probability of death π. π_{ij} was modeled as a departure ($\delta_i/2$) from μ_i, the mean mortality rate for each study (Eq.2). $\delta_i/2$ was subtracted from μ_i for the PCI treatment ($j = 0$) and added for the CABG one ($j = 1$). Both μ_i and δ_i were modeled as draws from normal distributions (Eq.3,4). The parameters for these normal distributions were further modeled as follows:

$$\mu_0 \sim N(0, 3.36) \tag{5}$$

$$\sigma^2 \sim Uniform(0, 3.36) \tag{6}$$

$$\delta_0 \sim N(0, 2.17) \tag{7}$$

$$\tau^2 \sim Uniform(0, 2.17) \tag{8}$$

These priors were set in a way that would make them the least informative possible. I did this because I assumed there was no other source of information besides the eleven studies used here. In the case of μ_0 (Eq.5), I set it to be centered at zero because this is the logit value correspondent to a π_0 of 0.5, which is the least informative value for the mortality probability across studies and treatments. To get to the variance of this prior distribution I used the range method assuming the 2.5% and 97.5% quantiles of π_0 were the most extreme possible cases ($0.025 < \pi_0 < 0.975$). The logit of these boundaries translates into ± 3.66 ($3.66 < logit(\pi_0) < 3.66$), and the maximum expected variance given the range method in this case is 3.36. For this same reason, I established $sigma^2$ to be constrained between 0 and 3.36 (Eq.6), using a uniform distribution due to lack of further information. I applied the same reasoning to set the prior distributions of δ_0 (Eq.7) and tau^2 (Eq.8), setting δ_0 to be centered around zero, thus not biasing the posteriors in any direction (in favor of PCI or CABG). The limits of δ_0, if $0.025 < \pi_0 < 0.975$, would be 0.95 (2.94 in logit units); thus, by the range method, I set the variance to be 2.17. This is why tau^2 is set as a uniform distribution constrained between zero and 2.17. The whole analysis was run in JAGS through R; the code for the model used in JAGS is on page 6, and the script used in R to run the analyses and produce the graphics is on page 6. The model was run for 101000 times.

Results

Temporal auto-correlation between posterior samples vanished after a few time steps (Fig.1) and time series plots (Fig.2) show that MCMC had a rapid convergence (the first 1000 simulations were discarded). Main results are shown in Table 1. Posterior distributions of δ_i show that there is a considerable variation in the difference between the two treatments, PCI being the best method in some studies, and CABG in others (Fig.3). The posterior distribution of the hyper-parameter δ_0, however, shows that there is no significant difference between the treatments (Fig.4A), despite the fact that PCI has, on average, slightly lower mortality probabilities than CABG (Fig.4B). In order to evaluate the degree to which the choice of priors affected the results, I performed a sensitivity analysis (31000 simulations, excluding the first 1000 ones) by wiggling four of the eight constants used to define the prior distributions: the means of μ_0 and δ_0 and the maximum limit of $sigma^2$ and tau^2. Results showed that, as expected, the posterior distribution of δ_0 is only affected by the priors related to δ_i (Fig.5). As it could be expected, centering δ_0 around zero was fundamental to avoid biasing the results towards one of the treatments; changing the parameters that define the maximum limit of tau^2 only affects the uncertainty around this study conclusions (even dividing variation by half, though, was not enough to make one treatment significantly better than the other).

Conclusion

This paper's conclusion is that, although the PCI treatment seem to be an improvement over CABG, there is not enough evidence from the studies done so far that would allow us to state that. On the other hand, given that PCI is a procedure with less complications and with shorter recovery time, knowing it is at least as secure as the CABG may be enough for doctors to prescribe it preferentially.

Appendix

Figures

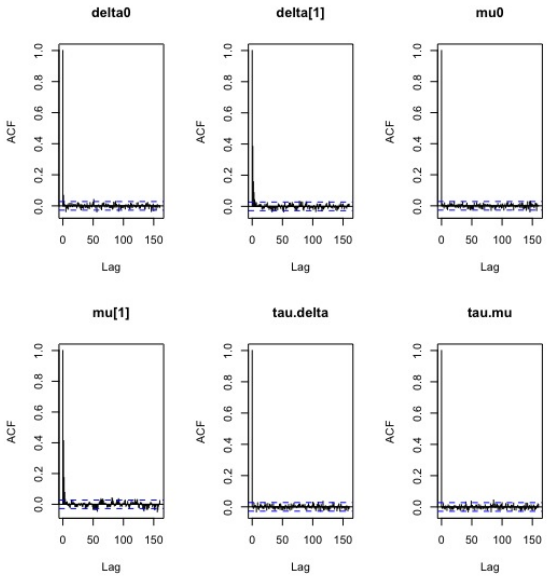

Figure 1: Posterior distributions of δ_i.

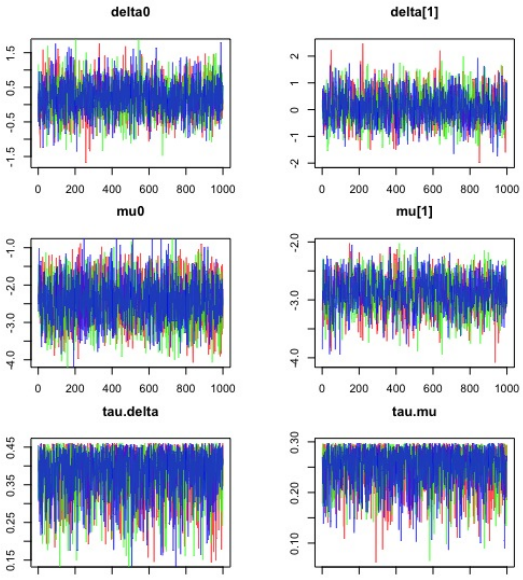

Figure 2: Posterior distributions of δ_i.

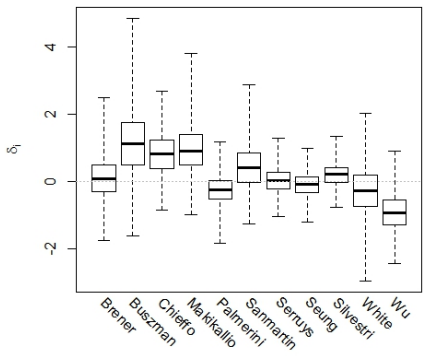

Figure 3: Posterior distributions of δ_i.

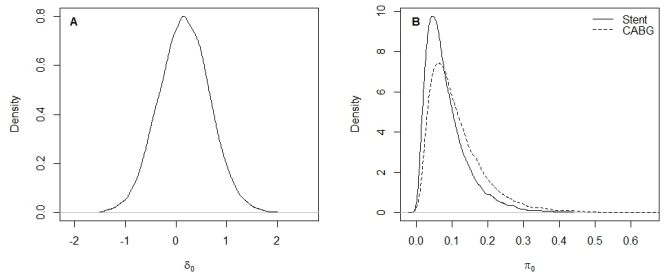

Figure 4: Posterior distributions of δ_0 (A) and π_0 for both treatment types (B).

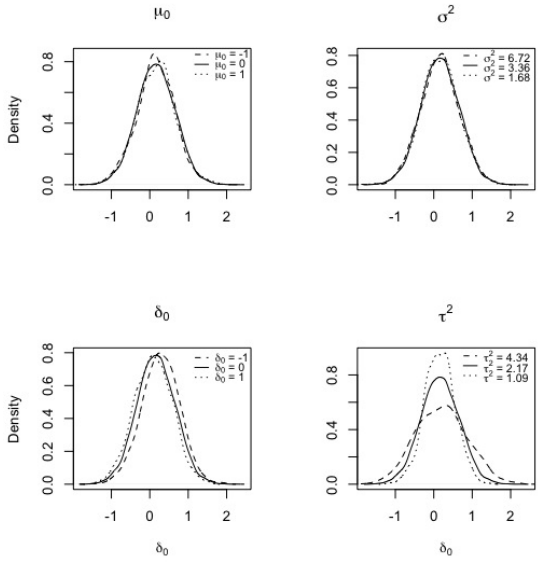

Figure 5: Posterior distribution of δ_0 as a function of wiggling one of the four parameters set for the priors: μ_0, σ^2, δ_0, and τ^2.

Tables

Table 1: Descriptive statistics of the posterior samples taken for all estimated parameters.

	mean	sd	2.5%	97.5%
δ_0	0.16	0.50	-0.82	1.13
δ_1	0.08	0.58	-1.01	1.27
δ_2	1.10	0.91	-0.59	3.01
δ_3	0.82	0.63	-0.37	2.12
δ_4	1.02	0.69	-0.20	2.54
δ_5	-0.27	0.44	-1.13	0.59
δ_6	0.40	0.62	-0.72	1.69
δ_7	0.03	0.36	-0.69	0.74
δ_8	-0.10	0.34	-0.76	0.56
δ_9	0.20	0.33	-0.42	0.84
δ_{10}	-0.32	0.69	-1.71	1.02
δ_{11}	-0.93	0.54	-2.02	0.09
μ_0	-2.43	0.58	-3.57	-1.27
μ_1	-2.88	0.31	-3.52	-2.32
μ_2	-3.25	0.50	-4.34	-2.37
μ_3	-3.17	0.34	-3.88	-2.57
μ_4	-2.68	0.36	-3.46	-2.06
μ_5	-2.02	0.23	-2.49	-1.59
μ_6	-2.63	0.32	-3.31	-2.06
μ_7	-3.14	0.19	-3.52	-2.78
μ_8	-3.34	0.17	-3.68	-3.01
μ_9	-2.16	0.16	-2.50	-1.85
μ_{10}	-2.14	0.37	-2.93	-1.47
μ_{11}	-2.23	0.28	-2.81	-1.70
π_{1s}	0.06	0.03	0.02	0.12
π_{2s}	0.03	0.02	0.00	0.08
π_{3s}	0.03	0.02	0.01	0.07
π_{4s}	0.05	0.03	0.01	0.12
π_{5s}	0.14	0.04	0.07	0.21
π_{6s}	0.06	0.03	0.02	0.14
π_{7s}	0.04	0.01	0.02	0.06
π_{8s}	0.04	0.01	0.02	0.05
π_{9s}	0.10	0.02	0.06	0.14
π_{10s}	0.13	0.05	0.05	0.25
π_{11s}	0.15	0.04	0.08	0.24
π_{1c}	0.06	0.02	0.03	0.10
π_{2c}	0.07	0.03	0.02	0.15
π_{3c}	0.06	0.02	0.03	0.11
π_{4c}	0.10	0.02	0.07	0.14
π_{5c}	0.11	0.03	0.06	0.17
π_{6c}	0.08	0.02	0.05	0.12
π_{7c}	0.04	0.01	0.02	0.07
π_{8c}	0.03	0.01	0.02	0.05
π_{9c}	0.11	0.02	0.08	0.16
π_{10c}	0.10	0.04	0.03	0.20
π_{11c}	0.07	0.03	0.03	0.13
τ^2	0.38	0.06	0.22	0.46
σ^2	0.25	0.04	0.16	0.30

s = Stent, c = CABG

JAGS model

```
model{
  for(i in 1:N){
    for(j in 1:2){
      y[i,j] ~ dbin(pie[i,j],n[i,j])
      logit(pie[i,j]) <- mu[i] + (2*j - 3) * delta[i]/2
      }

    mu[i] ~ dnorm(mu0, tau.mu)
    delta[i] ~ dnorm(delta0, tau.delta)
    }

  mu0 ~ dnorm(mm,mt)
  tau.mu ~ dunif(tm1,tm2)
  delta0 ~ dnorm(dm,dt)
  tau.delta ~ dunif(td1,td2)
  }
```

R script

```
### Data Analysis Project

# Loading packages
library(R2jags)
library(lattice)
library(ggplot2)
library(xtable)

# Useful functions
source("AddBurnin.R")
logit = function(x) log(x/(1-x)) # creates logit function
ilogit = function(x) exp(x)/(1+exp(x)) # creates inverse logit function

# Data
namesd = list(c("Brener","Buszman","Chieffo","Makikallio","Palmerini",
        "Sanmartin","Serruys","Seung","Silvestri","White","Wu"),c("Stent","CABG")) # labels for data table
nn = matrix(c(71,174,52,53,107,142,49,238,86,103,49,241,357,348,516,512,186,218,36,41,70,80),
        nrow=11, ncol=2, byrow=T) # number of people in each treatment group in each study
yy = matrix(c(4,1,3,2,12,3,15,19,18,5,11,10,4,9,25,11,20,15,17,25,4,5),
        nrow=11, ncol=2, byrow=F) # idem, but for the number of deaths after one year of the treatment
dimnames(nn) = dimnames(yy) = namesd
pp = yy/nn # mortality rates
round(pp,2)
round(apply(pp,2,mean),2)

# Jags model (see above)

# Jags input info
data = list(n=nn, y=yy, N=nrow(nn), # raw data
        mm=0, mt=1/3.36, tm1=0, tm2=1/3.36, dm=0, dt=1/2.17, td1=0, td2=1/2.17) # parameters that define the prior distributio
inits = rep(list(list(
       pie=matrix(rep(0.5,22), ncol=2, nrow=11, byrow=TRUE), # death probabilities per study per treatment
       mu=rep(0,11), delta=rep(0,11), # study-level parameters
       mu0=0, tau.mu=0.01, delta0=0, tau.delta=0.01 # hyper-parameters to be estimated
       )),3) # number of chains
params = c("pie[1:11,1:2]","mu[1:11]","delta[1:11]","mu0","delta0","tau.mu","tau.delta")

# Jags output
model.out = jags(data=data, inits=NULL, parameter=params, "model.txt",
        n.chains=3, n.iter=101000, n.burnin=0, n.thin=20, DIC=F)
outparams = dimnames(model.out$BUGSoutput$sims.array)[[3]] # all parameter names estimated by jags

# Take the first 1000 runs as burnin
Output = AddBurnin(model.out$BUGSoutput$sims.array, burnin=1000,n.thin=1)

# Checking for MCMC convergence
pdf("acf.pdf",paper="letter")
par(mfrow=c(2,3))
for(i in outparams){
  acf(model.out$BUGSoutput$sims.array[1:5000, 1, i], lag.max= 160, main=i)
  }
dev.off()
par(mfrow=c(1,1))

par(mfrow=c(2,3))
for(i in c("delta0","delta[1]","mu0","mu[1]","tau.delta","tau.mu")){
  acf(model.out$BUGSoutput$sims.array[1:5000, 1, i], lag.max= 160, main=i)
  }
par(mfrow=c(1,1))

# Time series
cols = rainbow(3,alpha=0.7)
pdf("timeSeries.pdf",paper="letter")
par(mfrow=c(4,1))
for(i in outparams){
  plot(model.out$BUGSoutput$sims.array[1:1000, 1, i], type="l", col=cols[1], main=i, ylab="", xlab="Iteration")
  lines(model.out$BUGSoutput$sims.array[1:1000, 2, i], type="l", col=cols[2])
  lines(model.out$BUGSoutput$sims.array[1:1000, 3, i], type="l", col=cols[3])
  }
dev.off()
par(mfrow=c(1,1))
```

```
par(mfrow=c(3,2),mar=c(1, 4, 4, 2) + 0.1)
for(i in c("delta0","delta[1]","mu0","mu[1]","tau.delta","tau.mu")){
  plot(model.out$BUGSoutput$sims.array[1:1000, 1, i], type="l", col=cols[1], main=i, ylab="", xlab="Iteration")
  lines(model.out$BUGSoutput$sims.array[1:1000, 2, i], type="l", col=cols[2])
  lines(model.out$BUGSoutput$sims.array[1:1000, 3, i], type="l", col=cols[3])
}
par(mfrow=c(1,1),mar=c(5, 4, 4, 2) + 0.1)

# Main results
Output$Burnin.Summary
xtable(Output$Burnin.Summary[,1:4])

# Stent vs. CABG (pie[i])
posts = Output$Burnin.sims.matrix
boxplot(posts[1:1000,paste("delta[",1:11,"]",sep="")],range=0,xaxt="n",ylab=expression(paste(delta[i])))
abline(h=0,lty=3,col="grey")
text(x=1:11,y=-3.5,namesd[[1]],xpd=T,srt=-45,offset=0,adj=0)

# Stent vs. CABG (pie0)
par(mfrow=c(1,2))
plot(density(posts[,"delta0"]),xlab=expression(paste(delta[0])),main="")
mtext("A",font=2,side=3,at=-2,line=-1.5)

pp0Stent = ilogit(posts[,"mu0"]-posts[,"delta0"]) # posterior distribution of pie 0 for Stent
pp0CABG = ilogit(posts[,"mu0"]+posts[,"delta0"]) # posterior distribution of pie 0 for CABG
plot(density(pp0Stent),xlab=expression(paste(pi[0])),main="")
lines(density(pp0CABG),lty=2)
legend("topright",legend=c("Stent","CABG"),lty=1:2,bty="n")
mtext("B",font=2,side=3,at=0,line=-1.5)
par(mfrow=c(1,1))

# Sensitivity analysis
senan = function(ni=31000, mm=0, mt=1/3.36, tm1=0, tm2=1/3.36, dm=0, dt=1/2.17, td1=0, td2=1/2.17){
  data = list(n=nn, y=yy, N=nrow(nn), mm=mm, mt=mt, tm1=tm1, tm2=tm2, dm=dm, dt=dt, td1=td1, td2=td2)
  model.out = jags(data=data, inits=NULL, parameter=params,
        "model.txt", n.chains=3, n.iter=ni, n.burnin=0, n.thin=20, DIC=F)
  Output = AddBurnin(model.out$BUGSoutput$sims.array, burnin=1000,n.thin=1)
  return(list(Output$Burnin.Summary,Output$Burnin.sims.matrix[,"delta0"]))
  }

# Set delta0 prior to be centered at 1 and -1
D1 = senan(dm=1)
Dn1 = senan(dm=-1)

# Set mu0 prior to be centered around 1 and -1
M1 = senan(mm=1)
Mn1 = senan(mm=-1)

# Set sigma2 prior to no larger than 1/(3.36*2) and 2/3.36
Sd2 = senan(tm2=1/3.36/2)
Sm2 = senan(tm2=1/3.36*2)

# Set tau2 prior to no larger than 1/(2.17*2) and 2/2.17
Td2 = senan(td2=1/2.17/2)
Tm2 = senan(td2=1/2.17*2)

# Plot sensitivities of priors parameters (mu0, sigma2, delta0, and tau2) on delta0
par(mfrow=c(2,2))
rr = range(density(posts[,"delta0"])$y,density(M1[[2]])$y,density(Mn1[[2]])$y)
plot(density(posts[,"delta0"]),xlab="",main=expression(bold(paste(mu[0])))),ylim=rr)
lines(density(M1[[2]]),lty=2)
lines(density(Mn1[[2]]),lty=3)
legend(x=1,y=.9,lty=c(2,1,3),c(expression(paste(mu[0],"~=~-1")),expression(paste(mu[0],"~=~0")),
       expression(paste(mu[0],"~=~1"))),bty="n",y.intersp=1.5,cex=.8)

rr = range(density(posts[,"delta0"])$y,density(Sd2[[2]])$y,density(Sm2[[2]])$y)
plot(density(posts[,"delta0"]),xlab="",main=expression(bold(paste(sigma^2))),ylim=rr,ylab="")
lines(density(Sd2[[2]]),lty=2)
lines(density(Sm2[[2]]),lty=3)
legend(x=.6,y=.85,lty=c(2,1,3),c(expression(paste(sigma^2,"~=~6.72")),expression(paste(sigma^2,"~=~3.36")),
       expression(paste(sigma^2,"~=~1.68"))),bty="n",y.intersp=1.5,cex=.8)

rr = range(density(posts[,"delta0"])$y,density(D1[[2]])$y,density(Dn1[[2]])$y)
plot(density(posts[,"delta0"]),xlab=expression(paste(delta[0])),
       main=expression(bold(paste(delta[0])))),ylim=rr)
lines(density(D1[[2]]),lty=2)
lines(density(Dn1[[2]]),lty=3)
legend(x=1,y=.82,lty=c(2,1,3),c(expression(paste(delta[0],"~=~-1")),expression(paste(delta[0],"~=~0")),
       expression(paste(delta[0],"~=~1"))),bty="n",y.intersp=1.5,cex=.8)

rr = range(density(posts[,"delta0"])$y,density(Td2[[2]])$y,density(Tm2[[2]])$y)
plot(density(posts[,"delta0"]),xlab=expression(paste(delta[0])),
       main=expression(bold(paste(tau^2)))),ylim=rr,ylab="")
lines(density(Td2[[2]]),lty=2)
lines(density(Tm2[[2]]),lty=3)
legend(x=.6,y=1,lty=c(2,1,3),c(expression(paste(tau^2,"~=~4.34")),expression(paste(tau^2,"~=~2.17")),
       expression(paste(tau^2,"~=~1.09"))),bty="n",y.intersp=1.5,cex=.8)
par(mfrow=c(1,1))
```