Lab 3 submission

Gaurav Kandlikar

October 13, 2016

To Dos from Lab 3

(1) Run jags to get 50000 samples. Get plots of the priors and posteriors, and get autocorrelation plots.

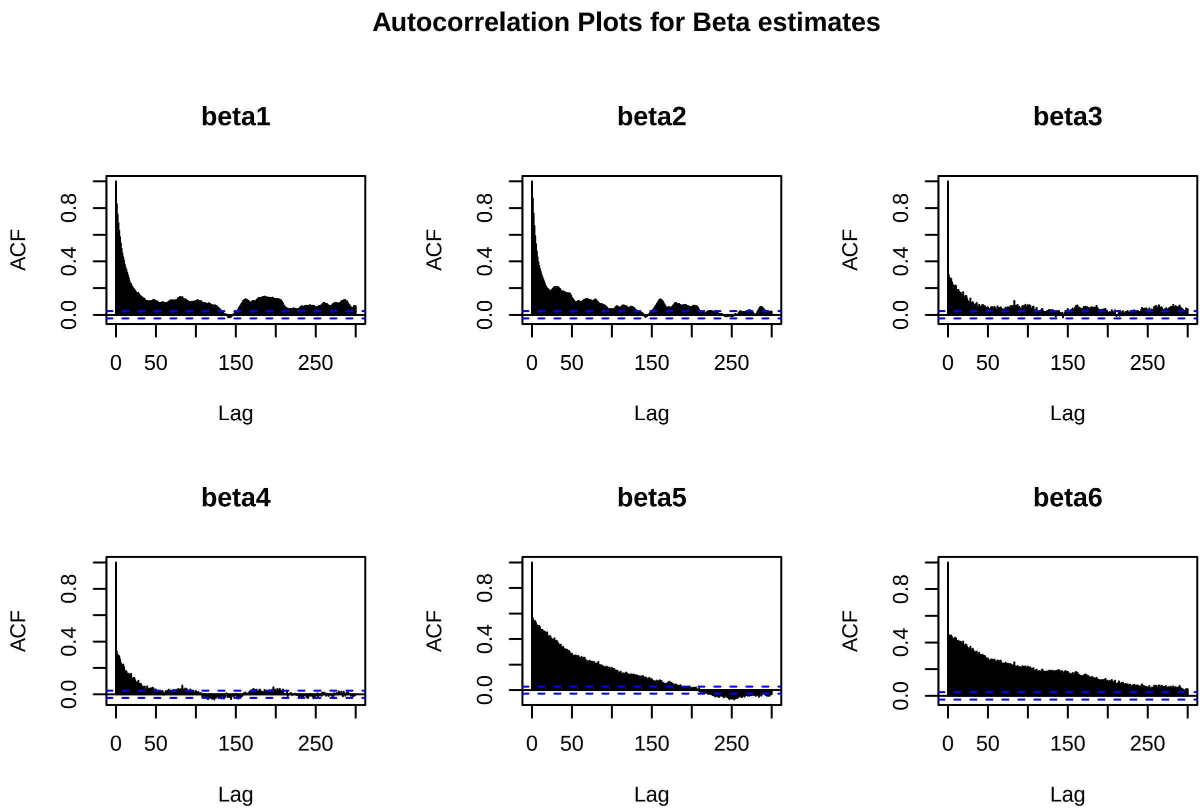
```
todo1.data=list(invXp=invXp)
todo1.inits=rep(list(list(pie=c(0.5,0.5,0.5,0.5,0.5,0.5))),5) #5 chains
todo1.parameters = c("betas", "pie[1:6]")

#Run the JAGS model
todo1.out = jags(todo1.data, todo1.inits, todo1.parameters, "LAB3.Priors.txt",
    n.chains=5, n.iter=51000, n.burnin=0, n.thin=2, DIC=F)
```

```
## module glm loaded

## module dic loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 6
##    Total graph size: 63
##
## Initializing model

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 300
##    Unobserved stochastic nodes: 6
##    Total graph size: 3026
##
## Initializing model

## [1] "model"              "BUGSoutput"         "parameters.to.save"
## [4] "model.file"         "n.iter"             "DIC"

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 100
##    Unobserved stochastic nodes: 6
##    Total graph size: 1062
##
## Initializing model
```

Assignment

Q1: 1. At what lags do the autocorrelations hit zero for the 6 regression coefficients? Are the beta autocorrelations better or worse than the 6 pi's?

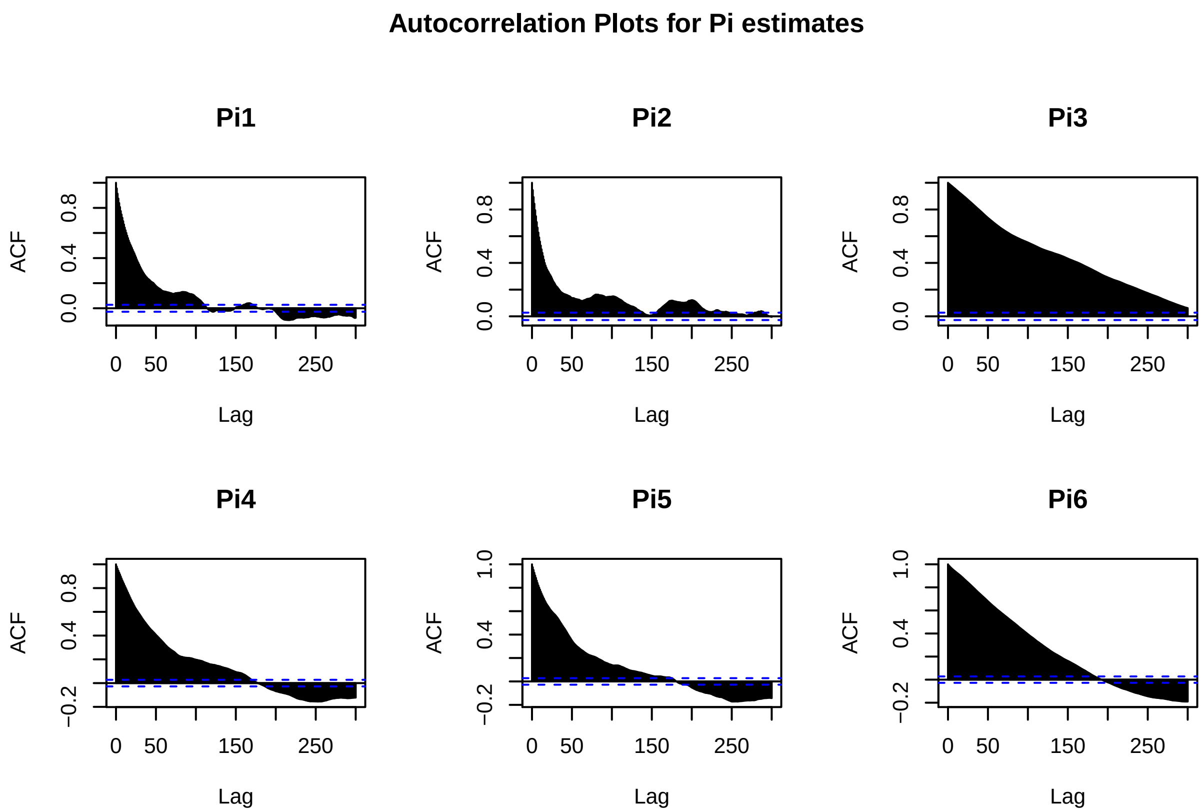
The following are autocorrelation plots for β coefficients.



The autocorrelation of the six βs hit zero at different times, and act differently after hitting zero for the first time.

Autocorrelation of β_1 hits 0 at Lag ≈ 70
Autocorrelation of β_2 hits 0 at Lag ≈ 180
Autocorrelation of β_3 hits 0 at Lag ≈ 100
Autocorrelation of β_4 hits 0 at Lag ≈ 70
Autocorrelation of β_5 hits 0 at Lag ≈ 70
Autocorrelation of β_6 hits 0 at Lag ≈ 300

The following are autocorrelation plots for *pi* from Full Dataset run:



Autocorrelation of π_1 hits 0 at Lag ≈ 100
Autocorrelation of π_2 hits 0 at Lag ≈ 150
Autocorrelation of π_3 hits 0 at Lag ≈ 300
Autocorrelation of π_4 hits 0 at Lag ≈ 150
Autocorrelation of π_5 hits 0 at Lag ≈ 150
Autocorrelation of π_6 hits 0 at Lag ≈ 150

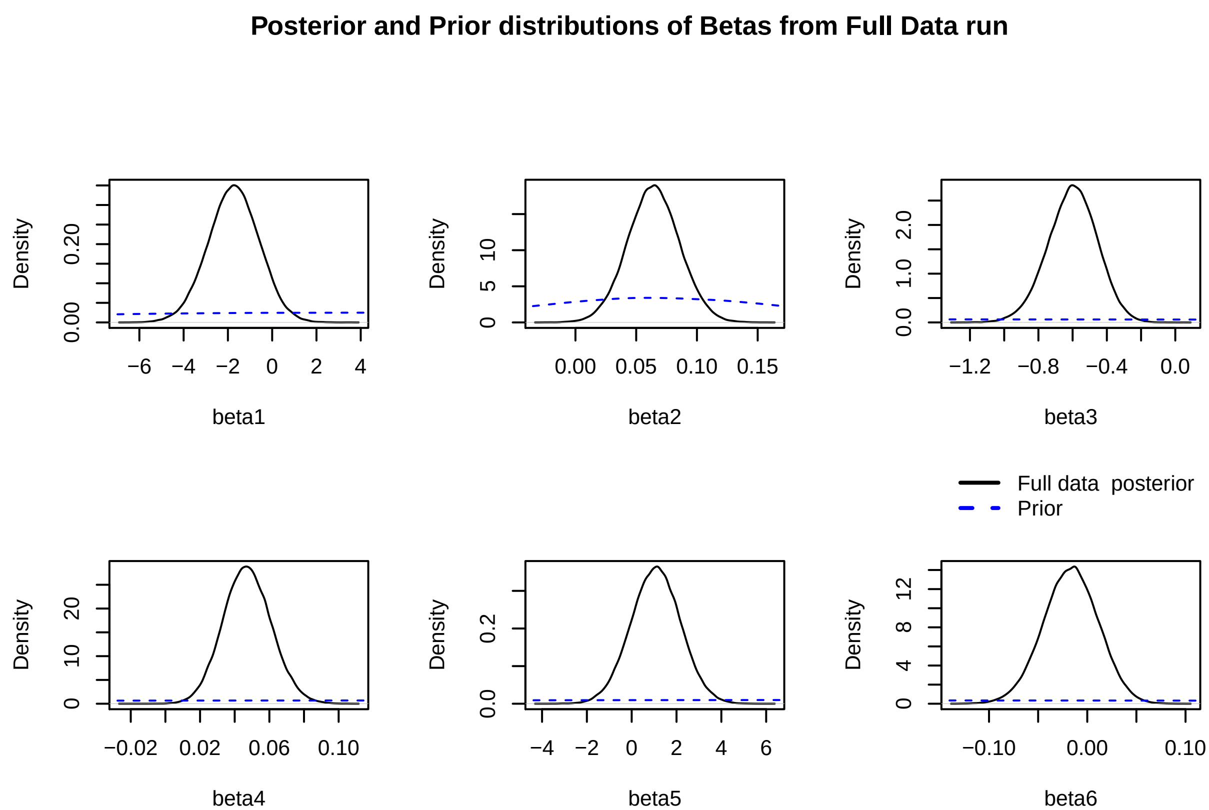
On the whole, the autocorrelations for the βs reach zero at a **lower lag value** than that of the pis. There are a few pi estimates (pi1, pi5, pi6) that reach zero autocorrelation very late, though that lag varies by run.

Q2: Turn in your properly formatted table of output for the full data set, and turn in a set of the 6 plots of the prior and posterior for the betas.

The following table summarizes the posterior distribution of Betas from the full-dataset model, and the following graphs show the posterior and prior distributions.

Table 1: Posterior of Betas from Full Dataset Run

	Parameter	Mean	SD	2.5%	97.5%	P>0
betas[1]	Intercept	-1.7379	1.1518	-4.0229	0.4979	0.0633
betas[2]	ISS	0.0644	0.0214	0.0224	0.1067	0.9984
betas[3]	RTS	-0.6011	0.1444	-0.8948	-0.327	0
betas[4]	Age	0.0473	0.0139	0.0208	0.0752	0.9999
betas[5]	ti	1.072	1.103	-1.0986	3.2351	0.8348
betas[6]	Age * ti	-0.017	0.028	-0.0728	0.0369	0.2737



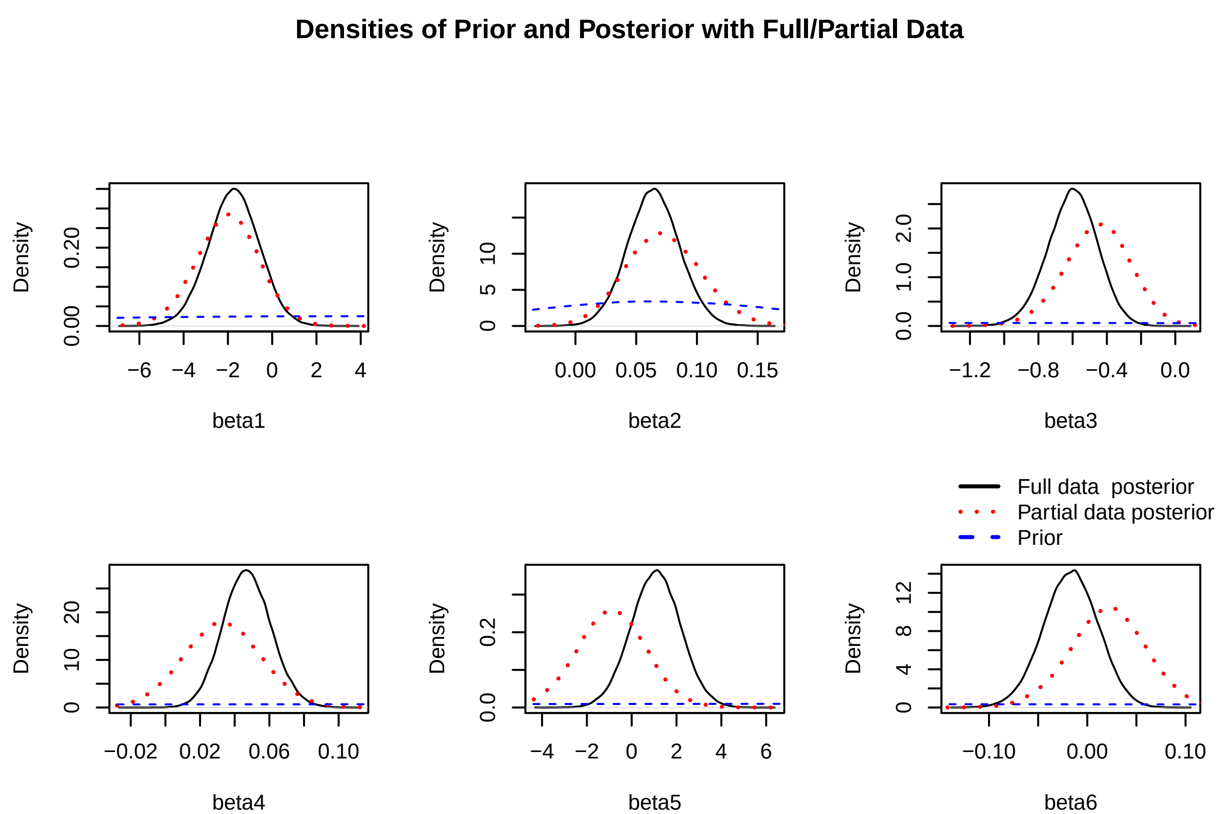
Q3: Turn in the results of the TODO step 2 properly formatted and your figures nicely annotated.

The following table summarizes estimates of βs from the partial-data model only:

Table 2: Posterior of Betas from Partial Dataset model

	Parameter	mu.vect	sd.vect	2.5%	97.5%	P>0
betas[1]	Intercept	-2.0138	1.4223	-4.8522	0.7346	0.0752
betas[2]	ISS	0.0728	0.0315	0.0136	0.1369	0.9921
betas[3]	RTS	-0.4558	0.1935	-0.8523	-0.0922	0.007
betas[4]	Age	0.033	0.0225	-0.0106	0.0777	0.9294
betas[5]	ti	-0.8976	1.563	-4.0059	2.1675	0.2808
betas[6]	Age * ti	0.021	0.0389	-0.0573	0.0959	0.712

The following density plots show the prior distribution for βs as well as the posterior distributions estimated from the full- and partial-data runs.



The following table summarizes the differences in estimates of βs depending on the dataset provided to calculate the posterior:

Table 3: Parameter estimates from Prior, Partial Data, and Full Data models

	model	mean	sd
Beta1	Prior	4.6488	16.7234
	Partial	-2.0138	1.4223
	Full	-1.7379	1.1518
Beta2	Prior	0.0644	0.1227
	Partial	0.0728	0.0315
	Full	0.0644	0.0214
Beta3	Prior	-3.062	6.5361
	Partial	-0.4558	0.1935
	Full	-0.6011	0.1444
Beta4	Prior	0.2541	0.5942
	Partial	0.033	0.0225
	Full	0.0473	0.0139
Beta5	Prior	15.9558	41.7255
	Partial	-0.8976	1.563
	Full	1.072	1.103
Beta6	Prior	-0.4464	1.1837
	Partial	0.021	0.0389
	Full	-0.017	0.028

Q4: The model tracks the parameters pi1 to pi6, what is the interpretation of these parameters once the data has been incorporated?

The posterior distribution of pi1 to pi6 represents the posterior probability that the six pseudo-cases will die. The posterior of these pi estimates incorporate the expert opinion from the doctor as well as the 300 datapoints to yield a posterior.

Q5: Extra credit: you may (but don't need to) Turn in your answer to TODO step 3.

We add the following lines of code to our BUGS model to use the posterior predictive to make predictions about future cases:

```
futurefit1 <- ilogit(inprod(people[1,],betas[]))
futurefit2 <- ilogit(inprod(people[2,],betas[]))
futurefit3 <- ilogit(inprod(people[3,],betas[]))
```

With this setup, we need to ensure that the data list that we provide JAGS include a matrix called `people` that has the same column headings as the dataset of 300, and 3 rows corresponding to the new observations. In the parameter list, we need to add `futurefit1`, `futurefit2`, and `futurefit3` so that we can retrieve these parameters after the model runs and explore their distribution.

We can plot the posterior predictive probability that each of the three patients will die and make a table of our predictions:

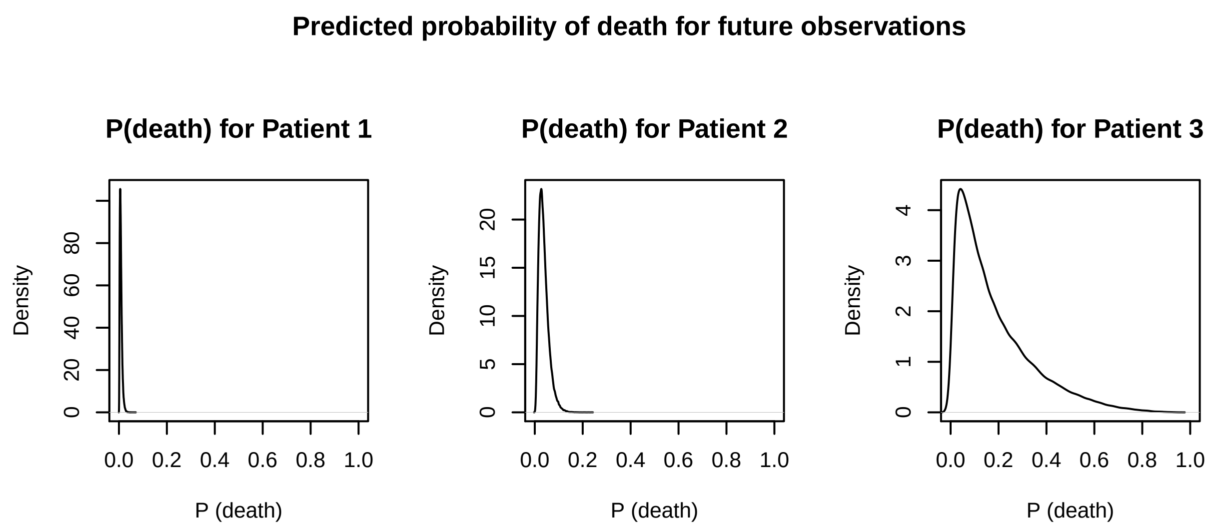


Table 4: Predicted probabilities of survival for hypothetical patients

	Mean	SD	2.5%	97.5%	P>0
Patient 1	0.0082	0.0049	0.0019	0.0206	1
Patient 2	0.0379	0.0211	0.0096	0.0904	1
Patient 3	0.1838	0.1578	0.0124	0.5963	1

Code Appendix

Q1

```
# Betas
par(mfrow = c(2,3), oma = c(0,0,4,0))
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,1], main="beta1", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,2], main="beta2", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,3], main="beta3", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,4], main="beta4", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,5], main="beta5", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,6], main="beta6", lag.max = 300)
title(outer = T, "Autocorrelation Plots for Beta estimates")

# Pis
par(mfrow=c(2,3), oma = c(0,0,4,0))
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,7], main="Pi1", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,8], main="Pi2", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,9], main="Pi3", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,10], main="Pi4", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,11], main="Pi5", lag.max = 300)
acf(todo1a.out$BUGSoutput$sims.array[1:5000,1,12], main="Pi6", lag.max = 300)
title(outer = T, "Autocorrelation Plots for Pi estimates")
```

Q2

```
names <- paste0("betas[", 1:6, "]")
to_print <- round(Output2$Burnin.Summary[names, ], 4)
colnames(to_print) = c("Mean", "SD", "2.5%", "97.5%", "P>0")
Parameter <- c("Intercept", "ISS", "RTS", "Age", "ti", "Age * ti")
to_print <- cbind(Parameter, to_print)
knitr::kable(to_print, digits = 4, caption = "Posterior of Betas from Full Dataset Run")
par(mfrow = c(2,3), oma = c(0,0,4,0))
plot(density(Output2$Burnin.sims.matrix[,1]), xlab = "beta1", main = "")
lines(density(Output1$Burnin.sims.matrix[,1]), xlim = c(-4, 4), col = "blue", lty = 2)

plot(density(Output2$Burnin.sims.matrix[,2]), xlab = "beta2", main = " ")
lines(density(Output1$Burnin.sims.matrix[,2]), col = "blue", lty = 2, xlim = c(0, .15))

plot(density(Output2$Burnin.sims.matrix[,3]), xlab = "beta3", main = "")
lines(density(Output1$Burnin.sims.matrix[,3]), col = "blue", lty = 2, xlim = c(-1.2, .6))

plot(density(Output2$Burnin.sims.matrix[,4]), xlab = "beta4", main = "")
lines(density(Output1$Burnin.sims.matrix[,4]), col = "blue", lty = 2, xlim = c(0, .13))

plot(density(Output2$Burnin.sims.matrix[,5]), xlab = "beta5", main = "")
lines(density(Output1$Burnin.sims.matrix[,5]), col = "blue", lty = 2, xlim = c(-4, 5))

plot(density(Output2$Burnin.sims.matrix[,6]), xlab = "beta6", main = "")
lines(density(Output1$Burnin.sims.matrix[,6]), col = "blue", lty = 2, xlim = c(.1, .1))
par(xpd = T)
```

```r
legend("topleft", lty = c(1,2), col = c("black", "blue"), lwd = 2,
       legend = c("Full data  posterior", "Prior"), inset=c(0,-.7), bty = "n")

title("Posterior and Prior distributions of Betas from Full Data run", outer = T)
```

Q3

```r
names <- paste0("betas[", 1:6, "]")
to_print <- round(Output3$Burnin.Summary[names, ], 4)
to_print <- cbind(Parameter, to_print)

knitr::kable(to_print, digits = 4, caption = "Posterior of Betas from Partial Dataset model")
## Make Plots
#par(mfrow = c(3,2))
par(mfrow = c(2,3), oma = c(0,0,4,0))
plot(density(Output2$Burnin.sims.matrix[,1]), xlab = "beta1", main = "")
lines(density(Output3$Burnin.sims.matrix[,1]), xlab = "beta1", col = "red", lty = 3, lwd = 2)
lines(density(Output1$Burnin.sims.matrix[,1]), xlim = c(-4, 4), col = "blue", lty = 2)

plot(density(Output2$Burnin.sims.matrix[,2]), xlab = "beta2", main = "")
lines(density(Output3$Burnin.sims.matrix[,2]), col = "red", lty = 3, lwd = 2, xlim = c(0, .15))
lines(density(Output1$Burnin.sims.matrix[,2]), col = "blue", lty = 2, xlim = c(0, .15))

plot(density(Output2$Burnin.sims.matrix[,3]), xlab = "beta3", main = "")
lines(density(Output3$Burnin.sims.matrix[,3]), col = "red", lty = 3, lwd = 2, xlim = c(-1.2, .6))
lines(density(Output1$Burnin.sims.matrix[,3]), col = "blue", lty = 2, xlim = c(-1.2, .6))

plot(density(Output2$Burnin.sims.matrix[,4]), xlab = "beta4", main = "")
lines(density(Output3$Burnin.sims.matrix[,4]), col = "red", lty = 3, lwd = 2, xlim = c(0, .13))
lines(density(Output1$Burnin.sims.matrix[,4]), col = "blue", lty = 2, xlim = c(0, .13))

plot(density(Output2$Burnin.sims.matrix[,5]), xlab = "beta5", main = "")
lines(density(Output3$Burnin.sims.matrix[,5]), col = "red", lty = 3, lwd = 2,
      xlab = "beta5", main = "")
lines(density(Output1$Burnin.sims.matrix[,5]), col = "blue", lty = 2, xlim = c(-4, 5))

plot(density(Output2$Burnin.sims.matrix[,6]), xlab = "beta6", main = "")
lines(density(Output3$Burnin.sims.matrix[,6]), col = "red", lty = 3, lwd = 2,
      xlab = "beta6", main = "")
lines(density(Output1$Burnin.sims.matrix[,6]), col = "blue", lty = 2, xlim = c(.1, .1))
par(xpd = T)
legend("topleft", lty = c(1,3,2), col = c("black", "red", "blue"), lwd = c(2,2,2),
       legend = c("Full data  posterior", "Partial data posterior", "Prior"),
       inset=c(0,-.7), bty = "n")
title("Densities of Prior and Posterior with Full/Partial Data", outer = T)

beta1_means <- c(Output1$Burnin.Summary["betas[1]","mu.vect"],
                 Output3$Burnin.Summary["betas[1]","mu.vect"],
                 Output2$Burnin.Summary["betas[1]","mu.vect"])
beta1_sds <- c(Output1$Burnin.Summary["betas[1]","sd.vect"],
               Output3$Burnin.Summary["betas[1]","sd.vect"],
               Output2$Burnin.Summary["betas[1]","sd.vect"])
```

```r
beta2_means <- c(Output1$Burnin.Summary["betas[2]","mu.vect"],
                 Output3$Burnin.Summary["betas[2]","mu.vect"],
                 Output2$Burnin.Summary["betas[2]","mu.vect"])
beta2_sds <- c(Output1$Burnin.Summary["betas[2]","sd.vect"],
               Output3$Burnin.Summary["betas[2]","sd.vect"],
               Output2$Burnin.Summary["betas[2]","sd.vect"])


beta3_means <- c(Output1$Burnin.Summary["betas[3]","mu.vect"],
                 Output3$Burnin.Summary["betas[3]","mu.vect"],
                 Output2$Burnin.Summary["betas[3]","mu.vect"])
beta3_sds <- c(Output1$Burnin.Summary["betas[3]","sd.vect"],
               Output3$Burnin.Summary["betas[3]","sd.vect"],
               Output2$Burnin.Summary["betas[3]","sd.vect"])

beta4_means <- c(Output1$Burnin.Summary["betas[4]","mu.vect"],
                 Output3$Burnin.Summary["betas[4]","mu.vect"],
                 Output2$Burnin.Summary["betas[4]","mu.vect"])
beta4_sds <- c(Output1$Burnin.Summary["betas[4]","sd.vect"],
               Output3$Burnin.Summary["betas[4]","sd.vect"],
               Output2$Burnin.Summary["betas[4]","sd.vect"])

beta5_means <- c(Output1$Burnin.Summary["betas[5]","mu.vect"],
                 Output3$Burnin.Summary["betas[5]","mu.vect"],
                 Output2$Burnin.Summary["betas[5]","mu.vect"])
beta5_sds <- c(Output1$Burnin.Summary["betas[5]","sd.vect"],
               Output3$Burnin.Summary["betas[5]","sd.vect"],
               Output2$Burnin.Summary["betas[5]","sd.vect"])

beta6_means <- c(Output1$Burnin.Summary["betas[6]","mu.vect"],
                 Output3$Burnin.Summary["betas[6]","mu.vect"],
                 Output2$Burnin.Summary["betas[6]","mu.vect"])
beta6_sds <- c(Output1$Burnin.Summary["betas[6]","sd.vect"],
               Output3$Burnin.Summary["betas[6]","sd.vect"],
               Output2$Burnin.Summary["betas[6]","sd.vect"])

b1 <- cbind(beta1_means, beta1_sds)
b2 <- cbind(beta2_means, beta2_sds)
b3 <- cbind(beta3_means, beta3_sds)
b4 <- cbind(beta4_means, beta4_sds)
b5 <- cbind(beta5_means, beta5_sds)
b6 <- cbind(beta6_means, beta6_sds)

to_print <- round(rbind(b1, b2, b3, b4, b5, b6), 4)
to_print = cbind(rep(c("Prior", "Partial", "Full"), 3), to_print)
colnames(to_print) = c("model","mean", "sd")
rownames(to_print) = c("Beta1", " ", " ", "Beta2", " ", " ",
                       "Beta3", " ", " ", "Beta4", " ", " ",
                       "Beta5", " ", " ", "Beta6", " ", " ")
knitr::kable(to_print, caption = "Parameter estimates from Prior, Partial Data, and Full Data models")
```

Q5

```
# sink("Lab3.Posteriors.Future.txt")
cat("
model{

  # Restate the priors
    pie[1]~dbeta(1.1,8.5)
    pie[2]~dbeta(3.0,11.0)
    pie[3]~dbeta(5.9,1.7)
    pie[4]~dbeta(1.3,12.9)
    pie[5]~dbeta(1.1,4.9)
    pie[6]~dbeta(1.5,5.5)

  # Convert from log odds to odds, or whatever
    for(j in 1:6){
        logitp[j]<-logit(pie[j])
    }

    betas <- invXp %*% logitp[]

      # NOTE: This is where the data is modeled!:

        for(i in 1:T){
        y[i] ~ dbern(p[i])
        p[i]<-ilogit(inprod(x[i,],betas[]))
        }

    # Using the posterior predictive to estimate survivial prob:
    futurefit1 <- ilogit(inprod(people[1,],betas[]))
    futurefit2 <- ilogit(inprod(people[2,],betas[]))
    futurefit3 <- ilogit(inprod(people[3,],betas[]))


}
  ",fill = TRUE)
# sink()

p1 = c(1,2,7.55,25,0,0)
p2 = c(1, 11, 7.8408, 42, 1, 42)
p3 = c(1, 16, 7.8408, 80, 1, 80)
people <- rbind(p1, p2, p3)
colnames(people) = colnames(Xobs)
prediction.data = list(x=Xobs, y=Yobs, T = 300, invXp=invXp, people = people)
prediction.inits = rep(list(list(pie=c(0.5,0.5,0.5,0.5,0.5,0.5))),5)
prediction.parameters = c("futurefit1", "futurefit2", "futurefit3" )

future.out2 = jags(prediction.data, prediction.inits, prediction.parameters, "Lab3.Posteriors.Future.tx

FutureOut2 = AddBurnin(future.out2$BUGSoutput$sims.array,burnin=1000,n.thin=2)
par(mfrow = c(1,3), oma = c(0, 0, 4, 0))
plot(density(FutureOut2$Burnin.sims.array[,,"futurefit1"]),
     main = "P(death) for Patient 1", xlab = "P (death)", xlim = c(0,1))
```

```r
plot(density(FutureOut2$Burnin.sims.array[,,"futurefit2"]),
     main = "P(death) for Patient 2", xlab = "P (death)", xlim = c(0,1))
plot(density(FutureOut2$Burnin.sims.array[,,"futurefit3"]),
     main = "P(death) for Patient 3", xlab = "P (death)", xlim = c(0,1))
title("Predicted probability of death for future observations", outer = T)

to_print <- FutureOut2$Burnin.Summary
rownames(to_print) = c("Patient 1", "Patient 2", "Patient 3")
colnames(to_print) = c("Mean", "SD", "2.5%", "97.5%", "P>0")
knitr::kable(to_print, digits = 4,
             caption = "Predicted probabilities of survival for hypothetical patients")
```