REPLICATION CODE

In this section, we provide the code necessary R to replicate the above results, and offer further explanation where necessary. We begin by loading the required packages, reading in the Non-specific Lag Bayesian Model Averaging functions, and finally reading in the data for the first example, the litigant signal model.

```
rm(list=ls())
library(foreign)
library(MCMCpack)

source("albma.R")

data <- read.dta("CourtsData.dta", convert.factors=FALSE)
```

Our first model is the baseline litigant signal model, with the index of Policy Priority (`LIPP`) and Legislative Attention (`LLEGIS`) each lagged one time period (year, in this case).

```
# Reduced Form litigant signal model.
formula<-CASES~ LIPP + LLEGIS + LCASES + LMQV + ABMCH + warren + dumcivrights
   + dumcriminal + dumunions + dumecon + dumdueproc + dumjudicial + dumfed
   + dumenvironment + dumtax
rfj.out <- MCMCregress(formula, b0=0, B0=0.1, burnin = 1000, mcmc = 10000,
   marginal.likelihood= "Chib95", data=data)
# Plots of Posterior Distributions.
par(mfrow=c(1,2))
plot(density(rfj.out[,2]), main="Policy Priority t-1", xlab="Posterior Distribution",
   lwd=2)
abline(v=0,lty=2)
plot(density(rfj.out[,3]), main="Legislative Attention t-1",
   xlab="Posterior Distribution", lwd=2)
abline(v=0,lty=2)
```

Next, we estimate the distributed lag model, including six lags each for the index of Policy Priority and Legislative Attention. Note the proliferation of regressors in the model. We then plot the posterior distributions for each of the six lags of each of the lagged regressors.

```
# Distributed Lag litigant signal model.
formula<-CASES~ LIPP + L2IPP + L3IPP + L4IPP + L5IPP + L6IPP
   + LLEGIS + L2LEGIS + L3LEGIS + L4LEGIS + L5LEGIS + L6LEGIS
   + LCASES + LMQV + ABMCH + warren + dumcivrights + dumcriminal
   + dumunions + dumecon + dumdueproc + dumjudicial + dumfed
   + dumenvironment + dumtax
dlj.out <- MCMCregress(formula, b0=0, B0=0.1, burnin = 1000, mcmc = 10000,
   marginal.likelihood= "Chib95", data=data)
# Plots of Posterior Distributions for Policy Priority
par(mfrow=c(2,3))
plot(density(dlj.out[,2]), main="Policy Priority, t-1", xlim=c(-6,8), lwd=2, xlab="")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,3]), main="Policy Priority, t-2", xlim=c(-6,8), lwd=2, xlab="")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,4]), main="Policy Priority, t-3", xlim=c(-6,8), lwd=2, xlab="")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,5]), main="Policy Priority, t-4", xlim=c(-6,8), lwd=2,
   xlab="Posterior Distribution")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,6]), main="Policy Priority, t-5", xlim=c(-6,8), lwd=2,
   xlab="Posterior Distribution")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,7]), main="Policy Priority, t-6", xlim=c(-6,8), lwd=2,
   xlab="Posterior Distribution")
par(new=TRUE)
```

```
abline(v=0, lty=2)
# Plots of Posterior Distributions for Legislative Attention.
par(mfrow=c(2,3))
plot(density(dlj.out[,8]), main="Legislative Attention, t-1", xlim=c(-6,8), lwd=2, xlab="")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,9]), main="Legislative Attention, t-2", xlim=c(-6,8), lwd=2, xlab="")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,10]), main="Legislative Attention, t-3", xlim=c(-6,8), lwd=2, xlab="")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,11]), main="Legislative Attention, t-4", xlim=c(-6,8), lwd=2,
     xlab="Posterior Distribution")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,12]), main="Legislative Attention, t-5", xlim=c(-6,8), lwd=2,
     xlab="Posterior Distribution")
par(new=TRUE)
abline(v=0, lty=2)
plot(density(dlj.out[,13]), main="Legislative Attention, t-6", xlim=c(-6,8), lwd=2,
     xlab="Posterior Distribution")
par(new=TRUE)
abline(v=0, lty=2)
```

Finally, we turn to the non-specific lag model. Here, we specify the formula by identifying only the *unlagged* versions of the variable. The `alreg` function generates the appropriate lags once the user specifies the feasible set. Note that variables with non-specific lags should be listed first in the formula.

```
formula <- CASES ~ IPP + LEGIS + LCASES + MQV + ABMCH + warren
   + dumcivrights + dumcriminal + dumunions + dumecon + dumdueproc
   + dumjudicial + dumfed + dumenvironment + dumtax
```

```
judicial.out<-alreg(formula=formula,
data=data, model="ls",
method="pooled", lags=c(1,2),
by="svalue", time.var=data$year,
min.lag=1, max.lag=8,
ml.type = "Chib95",
b0 = 0, B0 = 0.1,
mod.prob.method = "BF")
```

We then pass the formula and the data to the `alreg` function. A brief description of the function is warranted. In addition to passing the formula and data to the function, the user specifies the model to be estimated. The function currently supports linear regression ("ls"), logit ("logit"), probit ("probit") and poisson ("poisson") models. The only option for method is currently "pooled" though future iterations of the function may include support for additional analyses of time-specific dynamics. Through `lags` the user then specifies which of the variables have non-specific lags. Here, it is `IPP` and `LEGIS`, therefore `lags=c(1,2)`. Panel variables are specified through `by`, while the time variable is specified by `time.var`. These are included in order to generate the lagged versions of the regressors. The feasible set is determined by setting a minimum lag length (`min.lag`) and a maximum lag length (`max.lag`). The function currently supports lags up to 15 periods. The next three items correspond to counterparts in `MCMCpack`. `ml.type` corresponds to `marginal.likelihood` in `MCMCpack`, and can be specified as "Laplace" or "Chib95" at this time. `b0` indicates the prior mean of β, while `B0` indicates the prior precision of β. Finally, `mod.prob.method` identifies the method by which posterior model probabilities will be calculated. The function presently supports Bayes Factors ("BF") and Bayesian Information Criterior ("BIC").

```
# Gather posterior means across specifications for LEGIS.
ipp.triples <- matrix(NA,8,3)
ipp.triples[1,] <- c(summary(rfj.out)$quantiles[2,3], summary(rfj.out)$quantiles[2,1],
   summary(rfj.out)$quantiles[2,5])
ipp.triples[2:7,] <- c(summary(dlj.out)$quantiles[2:7,3],
   summary(dlj.out)$quantiles[2:7,1], summary(dlj.out)$quantiles[2:7,5])
```

```
ipp.triples[8,] <- c(judicial.out$bma[2,1], judicial.out$bma[2,3],
   judicial.out$bma[2,4])
# Plot the posterior means across specifications.
plot(c(1:8), ipp.triples[,1], ylim=c(min(ipp.triples), max(ipp.triples)),
   pch=19, xaxt="n", ylab="Posterior Mean", xlab="Measure", main="Policy Priority")
axis(1, at=c(1:8), labels=c("Base", "DL t-1", "DL t-2", "DL t-3", "DL t-4",
   "DL t-5", "DL t-6", "BMA"))
segments(c(1:8), ipp.triples[,2], c(1:8), ipp.triples[,3], lwd=2)
abline(h=0, lty=2)
abline(v=1.5)
abline(v=7.5)

# Gather posterior means across specifications for LEGIS.
ipp.triples <- matrix(NA,8,3)
ipp.triples[1,] <- c(summary(rfj.out)$quantiles[3,3], summary(rfj.out)$quantiles[3,1],
     summary(rfj.out)$quantiles[3,5])
ipp.triples[2:7,] <- c(summary(dlj.out)$quantiles[8:13,3],
   summary(dlj.out)$quantiles[8:13,1], summary(dlj.out)$quantiles[8:13,5])
ipp.triples[8,] <- c(judicial.out$bma[3,1], judicial.out$bma[3,3],
   judicial.out$bma[3,4])
# Plot the posterior means across specifications.
plot(c(1:8), ipp.triples[,1], ylim=c(min(ipp.triples), max(ipp.triples)), pch=19,
   xaxt="n", ylab="Posterior Mean", xlab="Measure", main="Legislative Attention")
axis(1, at=c(1:8), labels=c("Base", "DL t-1", "DL t-2", "DL t-3", "DL t-4",
   "DL t-5", "DL t-6", "BMA"))
segments(c(1:8), ipp.triples[,2], c(1:8), ipp.triples[,3], lwd=2)
abline(h=0, lty=2)
abline(v=1.5)
abline(v=7.5)
```

The posterior model probabilities are saved in the `alreg` object. They can be plotted simply as follows:

```
# Plot posterior model probabilities.
plot(judicial.out$postprob, type="b", pch=4, lwd=2, xlab = "Lag",
```

```
     ylab = "Posterior Model Probability")
```

For the litigant signal model application, we estimate a number of sensitivity checks. The first examines posterior model probabilities across different specifications of the maximum lag in the feasible set. Here, we specify different values for the maximum lag, then loop through those values, saving the output to `plot.output`. Plotting is then straightforward.

```
# Choose a group of maximum lags.
mxlg <- c(4,6,8,10)
# Set up matrix for output.
plot.output <- matrix(NA, 4, 10)
# Loop through models with different feasible sets.
for (i in 1:length(mxlg)){
  judicial.out <- alreg(formula=formula,
                        data=data, model="ls", method="pooled", lags=c(1,2),
                        time.var = data$year, by="svalue", min.lag = 1,
                        max.lag = mxlg[i], ml.type ="Chib95", b0 = 0, B0 = .1,
                        mod.prob.method = "BF")
  plot.output[i,1:mxlg[i]] <- judicial.out$postprob
}
# Plot posterior model probabilities.
plot(1:10, plot.output[1,], type="b", pch=19, col=rgb(0,0,0,alpha=.3), lwd=2,
   xlab = "", ylab = "", xlim=c(1,10), ylim= c(0,1))
par(new=TRUE)
plot(1:10, plot.output[2,], type="b", pch=19, col=rgb(0,0,0,alpha=.3), lwd=2,
   xlab = "", ylab = "", xlim=c(1,10), ylim= c(0,1))
par(new=TRUE)
plot(1:10, plot.output[3,], type="b", pch=19, col=rgb(0,0,0,alpha=.3), lwd=2,
   xlab = "", ylab = "", xlim=c(1,10), ylim= c(0,1))
par(new=TRUE)
plot(1:10, plot.output[4,], type="b", pch=19, col=rgb(0,0,0,alpha=.3), lwd=2,
   xlab = "Lag", ylab = "Posterior Model Probability", xlim=c(1,10), ylim= c(0,1))
```

Our second sensitivity check again examines changes in the maximum lag in the feasible set, but now looks at the robustness of the average posterior mean for the regressor of interest,

Policy Priority. The code is thus similar to above, but now pulls the posterior mean and 95% credible interval for policy priority, then plots these as a ropeladder plot across the choices of the maximum lag.

```
# Choose a group of maximum lags.
mxlg <- c(3,4,5,6,7,8,9,10)
# Set up matrix for output.
plot.output <- matrix(NA, 8, 3)
# Loop through models with different feasible sets.
for (i in 1:length(mxlg)){
  judicial.out <- alreg(formula=formula,
                        data=data, model="ls", method="pooled", lags=c(1,2),
                        time.var = data$year, by="svalue", min.lag = 1,
                        max.lag = mxlg[i], ml.type ="Chib95", b0 = 0, B0 = .1,
                        mod.prob.method = "BF")
# Pull BMA estimate on Policy Priority.
  plot.output[i,1] <- judicial.out$bma[2,1]
# Pull lower bound on BMA estimate on Policy Priority.
  plot.output[i,2] <- judicial.out$bma[2,3]
# Pull upper bound on BMA estimate on Policy Priority.
  plot.output[i,3] <- judicial.out$bma[2,4]

}
# Plot average posterior means.
plot(mxlg, plot.output[,1], pch=19, cex=1.5, xlab = "Feasible Set",
   ylab = "Posterior Mean", xlim=c(min(mxlg),max(mxlg)), ylim= c(0,11))
# Add 95% credible intervals.
segments(mxlg, plot.output[,2], mxlg, plot.output[,3], col="black", lwd=2)
abline(h=0, lty=2, lwd=2, col="gray24")
par(new=TRUE)
```

Finally, we examine the robustness of the results to different specifications of the prior mean and prior precision. As in the sensitivity analyses above, we specify a set of estimates for each, estimate the model across the set, extract the relevant information from each model,

then plot the results.

```
# Identify the set of prior means.
b0 <- c(-5,-2,-1,-.5,-.1, -.01,0,.01,.1,.5, 1, 2, 5)
# Set up a matrix to store output.
output <- matrix(NA,length(b0),6)
# Loop through the models.
for (i in 1:length(b0)){
  judicial.out <- alreg(formula=formula,
                        data=data, model="ls", method="pooled", lags=c(1,2),
                        time.var = data$year, by="svalue", min.lag = 1, max.lag = 6,
                        ml.type ="Chib95", b0 = b0[i], B0 = .1,
                        mod.prob.method = "BF")
# Extract posterior model probabilities.
  output[i,] <- judicial.out$postprob
}
# Plot posterior model probabilities. Each line represents a particular lag,
  as indicated by pch.
plot(b0, output[,1], xlim=c(-5,5), ylim=c(0,1), type = "b", pch="1",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(b0, output[,2], xlim=c(-5,5), ylim=c(0,1), type = "b", pch="2",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(b0, output[,3], xlim=c(-5,5), ylim=c(0,1), type = "b", pch="3",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(b0, output[,4], xlim=c(-5,5), ylim=c(0,1), type = "b", pch="4",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(b0, output[,5], xlim=c(-5,5), ylim=c(0,1), type = "b", pch="5",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(b0, output[,6], xlim=c(-5,5), ylim=c(0,1), type = "b", pch="6",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="Posterior Probability", xlab="Values of b0")
```

```
# Identify the set of prior precisions
B0 <- c(.001,.02,.05,.1,.2,.5,.75, 1, 2, 5)
# Set up a matrix to store output.
output <- matrix(NA,length(B0),6)
# Loop through the models.
for (i in 1:length(B0)){
  judicial.out <- alreg(formula=formula,
                        data=data, model="ls", method="pooled", lags=c(1,2),
                        time.var = data$year, by="svalue", min.lag = 1, max.lag = 6,
                        ml.type ="Chib95", B0 = B0[i], b0 = .1,
                        mod.prob.method = "BF")

  output[i,] <- judicial.out$postprob
}
# Plot posterior model probabilities. Each line represents a particular lag,
  as indicated by pch.
plot(B0, output[,1], xlim=c(0,5), ylim=c(0,1), type = "b", pch="1",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(B0, output[,2], xlim=c(0,5), ylim=c(0,1), type = "b", pch="2",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(B0, output[,3], xlim=c(0,5), ylim=c(0,1), type = "b", pch="3",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(B0, output[,4], xlim=c(0,5), ylim=c(0,1), type = "b", pch="4",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(B0, output[,5], xlim=c(0,5), ylim=c(0,1), type = "b", pch="5",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="", xlab="")
par(new=TRUE)
plot(B0, output[,6], xlim=c(0,5), ylim=c(0,1), type = "b", pch="6",
   col=rgb(0,0,0,alpha=.3), lwd=2, ylab="Posterior Probability", xlab="Values of B0")
```

We next turn to the modernization theory example. We begin by clearing the workspace and reading in the modernization data. Then, we estimate the baseline modernization theory model, then move to the distributed lag modernization theory model.

```
rm(list=ls())
data <- read.dta("ModernizationData.dta")
source("albma.R")

# Specify baseline formula.
formula <- demd ~ demd_L1 + income_L1 + lpop_L1 + ethfrac_L1 + relfrac_L1
   + instab_L1 + oil + ncontig
# Estimate baseline model.
rfp.out <- MCMCprobit(formula, b0=0, B0=.01, burnin = 1000, mcmc = 10000,
   marginal.likelihood= "Chib95", data=data)

# Specify distributed lag formula.
formula <- demd ~ income_L1 + income_L2 + income_L3 + income_L4 + income_L5
   + income_L6 + income_L7 + income_L8 + income_L9 + income_L10 + lpop_L1
   + lpop_L2 + lpop_L3 + lpop_L4 + lpop_L5 + lpop_L6 + lpop_L7 + lpop_L8
   + lpop_L9 + lpop_L10 + ethfrac_L1 + ethfrac_L2 + ethfrac_L3 + ethfrac_L4
   + ethfrac_L5 + ethfrac_L6 + ethfrac_L7 + ethfrac_L8 + ethfrac_L9
   + instab_L1 + instab_L2 + instab_L3 + instab_L4 + instab_L5 + instab_L6
   + instab_L7 + instab_L8 + instab_L9 + instab_L10 + demd_L1 + oil + ncontig
# Estimate distributed lag model.
dlp.out <- MCMCprobit(formula, b0=0, B0=.01, burnin = 1000, mcmc = 10000,
   marginal.likelihood= "Chib95", data=data)

# Compose results table.
su.out <- matrix(NA,nrow(summary(dlp.out)$statistics),4)
su.out[,1] <- summary(dlp.out)$statistics[,1]
su.out[,2] <- summary(dlp.out)$statistics[,2]
su.out[,3] <- summary(dlp.out)$quantiles[,1]
su.out[,4] <- summary(dlp.out)$quantiles[,5]
rownames(su.out) <- rownames(summary(dlp.out)$quantiles)
```

```
xtable(su.out, digits=3)
```

For the distributed lag model, we plot the posterior means and 95% credible intervals for each of the lagged variables. To do so, we create vectors for each lagged regressor (i.e., income) of the posterior means for each lag. To plot the credible intervals, we create matrices of the lag value and lower and upper bounds.

```
# Create vectors of posterior means.
income.vec <- su.out[2:11,1]
lpop.vec <- su.out[12:21,1]
ethfrac.vec <- su.out[22:30,1]
instab.vec <- su.out[31:40,1]
lag.val <- 1:10
# Create matrices for the 95% credible interval.
income.mat <- su.out[2:11,3:4]
income.mat <- cbind(lag.val, income.mat[,1], lag.val, income.mat[,2])
lpop.mat <- su.out[12:21,3:4]
lpop.mat <- cbind(lag.val, lpop.mat[,1], lag.val, lpop.mat[,2])
lag.val <- 1:9
ethfrac.mat <- su.out[22:30,3:4]
ethfrac.mat <- cbind(lag.val, ethfrac.mat[,1], lag.val, ethfrac.mat[,2])
lag.val <- 1:10
instab.mat <- su.out[31:40,3:4]
instab.mat <- cbind(lag.val, instab.mat[,1], lag.val, instab.mat[,2])
lag.val <- c(1:10)
# Plot all four in one figure.
par(mfrow=c(2,2))
# Income.
par(mar=c(5,5,1,1))
plot(lag.val, income.vec, type="b", pch=19, lwd=3, lty=3, xlab="Lag Value",
   ylab="Income Posterior Means", ylim=c(-1,1))
abline(h=0, lwd=2, col="grey")
for(i in 1:nrow(income.mat)){
  segments(income.mat[i,1],income.mat[i,2],income.mat[i,3],income.mat[i,4],
```

```
     lwd=2)
}
# Population.
par(mar=c(5,5,1,1))
plot(lag.val, lpop.vec, type="b", pch=19, lwd=3, lty=3, xlab="Lag Value",
   ylab="Log Population Posterior Means", ylim=c(min(lpop.mat[,2]),
   max(lpop.mat[,4])))
abline(h=0, lwd=2, col="grey")
for(i in 1:nrow(lpop.mat)){
  segments(lpop.mat[i,1],lpop.mat[i,2],lpop.mat[i,3],lpop.mat[i,4], lwd=2)
}
# Ethnic fractionalization.
par(mar=c(5,5,1,1))
lag.val <- c(1:9)
plot(lag.val, ethfrac.vec, type="b", pch=19, lwd=3, lty=3, xlab="Lag Value",
   ylab="Eth. Fract. Posterior Means", ylim=c(min(ethfrac.mat[,2]),
   max(ethfrac.mat[,4])))
abline(h=0, lwd=2, col="grey")
for(i in 1:nrow(ethfrac.mat)){
  segments(ethfrac.mat[i,1],ethfrac.mat[i,2],ethfrac.mat[i,3],ethfrac.mat[i,4],
     lwd=2)
}
# Political instability.
lag.val <- 1:10
par(mar=c(5,5,1,1))
plot(lag.val, instab.vec, type="b", pch=19, lwd=3, lty=3, xlab="Lag Value",
   ylab="Instability  Posterior Means", ylim=c(min(instab.mat[,2]),
   max(instab.mat[,4])))
abline(h=0, lwd=2, col="grey")
for(i in 1:nrow(instab.mat)){
  segments(instab.mat[i,1], instab.mat[i,2], instab.mat[i,3], instab.mat[i,4],
     lwd=2)
}
```

To compute the Bayesian model average across the feasible set of lags, we again specify our formula using only the non-lagged versions of the variables of interest. Here, we choose a maximum lag of 15 for the feasible set, and identify the first five variables in the formula as having lags. The function therefore estimates 15 models, with the first five variables lagged 1 through 15 times.

```
formula <- demd ~ income + lpop + ethfrac + relfrac + instab + demd_L1 + oil
   + ncontig
polity.out <- alreg(formula=formula,
data=data, model="probit", method="pooled", lags=c(1,2,3,4,5),
time.var = "year", by="ccode", min.lag = 1, max.lag = 15,
ml.type ="Chib95", b0 = 0, B0 = .01,
mod.prob.method = "BF")
```

With three approaches completed, we create plots to compare the posterior means for each of the lagged variables across different approaches. To do so, we reshape the output of the distributed lag specification, then create vectors of posterior means and credible intervals for each of the lagged variables from the baseline model and the Bayesian model averaging approach. The posterior means and credible intervals for all three approaches are then stored in matrices for plotting.

```
# Reshape the distributed lag modernization example into matrices for plotting.
income.mat[,1] <- income.vec
lpop.mat[,1] <- lpop.vec
ethfrac.mat[,1] <- ethfrac.vec
instab.mat[,1] <- instab.vec
income.mat <- income.mat[,-3]
lpop.mat <- lpop.mat[,-3]
ethfrac.mat <- ethfrac.mat[,-3]
instab.mat <- instab.mat[,-3]

# Generate vectors of baseline and BMA posterior means and credible intervals
# for plotting, then store all three in a matrix for plotting.
```

```
income.rf <- c(summary(rfp.out)$statistics[3,1], summary(rfp.out)$quantiles[3,1],
   summary(rfp.out)$quantiles[3,5])
income.bma <- c(polity.out$bma[2,1], polity.out$bma[2,3], polity.out$bma[2,4])
income.mat <- rbind(income.rf, income.mat, income.bma)
lpop.rf <- c(summary(rfp.out)$statistics[4,1], summary(rfp.out)$quantiles[4,1],
   summary(rfp.out)$quantiles[4,5])
lpop.bma <- c(polity.out$bma[3,1], polity.out$bma[3,3], polity.out$bma[3,4])
lpop.mat <- rbind(lpop.rf, lpop.mat, lpop.bma)
ethfrac.rf <- c(summary(rfp.out)$statistics[5,1], summary(rfp.out)$quantiles[5,1],
   summary(rfp.out)$quantiles[5,5])
ethfrac.bma <- c(polity.out$bma[4,1], polity.out$bma[4,3], polity.out$bma[4,4])
ethfrac.mat <- rbind(ethfrac.rf, ethfrac.mat, ethfrac.bma)
instab.rf <- c(summary(rfp.out)$statistics[7,1], summary(rfp.out)$quantiles[7,1],
   summary(rfp.out)$quantiles[7,5])
instab.bma <- c(polity.out$bma[6,1], polity.out$bma[6,3], polity.out$bma[6,4])
instab.mat <- rbind(instab.rf, instab.mat, instab.bma)

# Plot Income.
par(mfrow=c(1,1))
plot(c(1:(nrow(income.mat))), income.mat[,1], ylim=c(min(income.mat),
   max(income.mat)), pch=19, xaxt="n", ylab="Posterior Mean",
   xlab="Measure", main="Income")
axis(1, c(1:(nrow(income.mat))), labels=c("RF", "DL t-1", "DL t-2", "DL t-3",
   "DL t-4", "DL t-5", "DL t-6", "DL t-7", "DL t-8", "DL t-9", "DL t-10",  "BMA"))
segments(c(1:(nrow(income.mat))), income.mat[,2], c(1:(nrow(income.mat))),
   income.mat[,3], lwd=2)
abline(h=0, lty=2)
abline(v=1.5)
abline(v=11.5)

# Plot Population.
par(mfrow=c(1,1))
plot(c(1:(nrow(lpop.mat))), lpop.mat[,1], ylim=c(min(lpop.mat), max(lpop.mat)),
   pch=19, xaxt="n", ylab="Posterior Mean", xlab="Measure",
```

```
   main="Population (Logged)")
axis(1, c(1:(nrow(lpop.mat))), labels=c("RF", "DL t-1", "DL t-2", "DL t-3",
   "DL t-4", "DL t-5", "DL t-6", "DL t-7", "DL t-8", "DL t-9", "DL t-10",  "BMA"))
segments(c(1:(nrow(lpop.mat))), lpop.mat[,2], c(1:(nrow(lpop.mat))),
   lpop.mat[,3], lwd=2)
abline(h=0, lty=2)
abline(v=1.5)
abline(v=11.5)

# Plot Ethnic Fractionalization.
par(mfrow=c(1,1))
plot(c(1:(nrow(ethfrac.mat))), ethfrac.mat[,1], ylim=c(min(ethfrac.mat),
   max(ethfrac.mat)), pch=19, xaxt="n", ylab="Posterior Mean", xlab="Measure",
   main = "Ethnic Fract.")
axis(1, c(1:(nrow(ethfrac.mat))), labels=c("RF", "DL t-1", "DL t-2", "DL t-3",
   "DL t-4", "DL t-5", "DL t-6", "DL t-7", "DL t-8", "DL t-9",  "BMA"))
segments(c(1:(nrow(ethfrac.mat))), ethfrac.mat[,2], c(1:(nrow(ethfrac.mat))),
   ethfrac.mat[,3], lwd=2)
abline(h=0, lty=2)
abline(v=1.5)
abline(v=10.5)

# Plot Political Instability.
par(mfrow=c(1,1))
plot(c(1:(nrow(instab.mat))), instab.mat[,1], ylim=c(min(instab.mat),
   max(instab.mat)), pch=19, xaxt="n", ylab="Posterior Mean", xlab="Measure",
   main="Instability")
axis(1, c(1:(nrow(instab.mat))), labels=c("RF", "DL t-1", "DL t-2", "DL t-3",
   "DL t-4", "DL t-5", "DL t-6", "DL t-7", "DL t-8", "DL t-9", "DL t-10",  "BMA"))
segments(c(1:(nrow(instab.mat))), instab.mat[,2], c(1:(nrow(instab.mat))),
   instab.mat[,3], lwd=2)
abline(h=0, lty=2)
abline(v=1.5)
abline(v=11.5)
```

Finally, we again plot the posterior model probabilities across the feasible set of lags. The posterior model probabilities are save in the `alreg` object, therefore plotting is straightforward.

```
# Then plot the posterior model probabilities
plot(polity.out$postprob, type="b", pch=4, lwd=2, xlab = "Feasible Set",
   ylab = "Posterior Model Probability")
```