Bayesian Statistical Methods

Partial solutions

Chapter 4: Linear models

Jump to probem: 1, 3, 5, 7, 9

(1) We assume the model $Y_i \sim \text{Normal}(\mu,\sigma^2)$ for placebo observations and $Y_i \sim \text{Normal}(\mu+\delta,\sigma^2)$ for treatment observations. The objective is to test whether $\delta = 0$ and thus the two groups have the same population mean. To do this we use the two-sample t-test with Jeffreys' prior in Equation (4.7). The results are

```
Y0    <- c(20,-31,-10,2,3,4)/10
Y1    <- c(-35,-16,-46,9,-51,1)/10
n0    <- n1 <- 6
xbar0 <- mean(Y0)
s20   <- var(Y0)
xbar1 <- mean(Y1)
s21   <- var(Y1)
sp    <- sqrt((s20/2+s21/2))

#Posterior of delta
post_mn <- xbar1-xbar0
post_sd <- sp*sqrt(1/n0+1/n1)
cred_set <- post_mn+post_sd*qt(c(0.025,0.975),df=n0+n1)

post_mn;post_sd;cred_set
```

```
## [1] -2.1
```

```
## [1] 1.234504
```

```
## [1] -4.789753  0.589753
```

The credible set includes zero and so there is not strong evidence that the mean differs by treatment group. To test for sensitivity to the prior we also fit the model using vague but proper priors using JAGS. The results are similar.

```
 library(rjags)
 data <- list(n=6,Y0=Y0,Y1=Y1)

 model_string <- textConnection("model{

  # Likelihood
  for(i in 1:n){
    Y0[i] ~ dnorm(mu,tau)
    Y1[i] ~ dnorm(mu+delta,tau)
  }

  # Priors
  mu    ~  dnorm(0, 0.0001)
  delta ~  dnorm(0, 0.0001)
  tau   ~  dgamma(0.1, 0.1)
  sigma <- 1/sqrt(tau)
}")

model <- jags.model(model_string,data = data, n.chains=2,quiet=TRUE)
update(model, 10000, progress.bar="none")
params  <- c("delta")
samples <- coda.samples(model,
           variable.names=params,
           n.iter=10000, progress.bar="none")
summary(samples)
```

```
##
## Iterations = 10001:20000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##      -2.126501       1.389859       0.009828       0.017040
##
## 2. Quantiles for each variable:
##
##    2.5%     25%     50%     75%   97.5%
## -4.9059 -3.0016 -2.1132 -1.2657  0.6476
```

(3a)

```
 load("election_2008_2016.RData")

 X     <- scale(X)   # standardize covariates
 X     <- cbind(1,X) # add intercept
 short <- c("Intercept", "Pop change", "65+","African American",
           "Hispanic","HS grad","Bachelor's",
           "Homeownership rate","Home value",
           "Median income", "Poverty")
 names <- c("Intercept", as.character(names[1:11,2]))
 colnames(X) <- short

 library(rjags)
 data <- list(n=length(Y),p=ncol(X),Y=Y,X=X)

 model_string <- textConnection("model{

  # Likelihood
  for(i in 1:n){
    Y[i] ~ dnorm(inprod(X[i,],beta[]),tau)
  }

  # Priors
  for(j in 1:p){beta[j] ~  dnorm(0, 0.0001)}
  tau ~ dgamma(0.01,0.01)
}")

model <- jags.model(model_string,data = data, n.chains=2,quiet=TRUE)
update(model, 10000, progress.bar="none")
params  <- c("beta")
samples <- coda.samples(model,
           variable.names=params,
           n.iter=10000, progress.bar="none")
out     <- summary(samples)$statistics
rownames(out)<-short
out
```

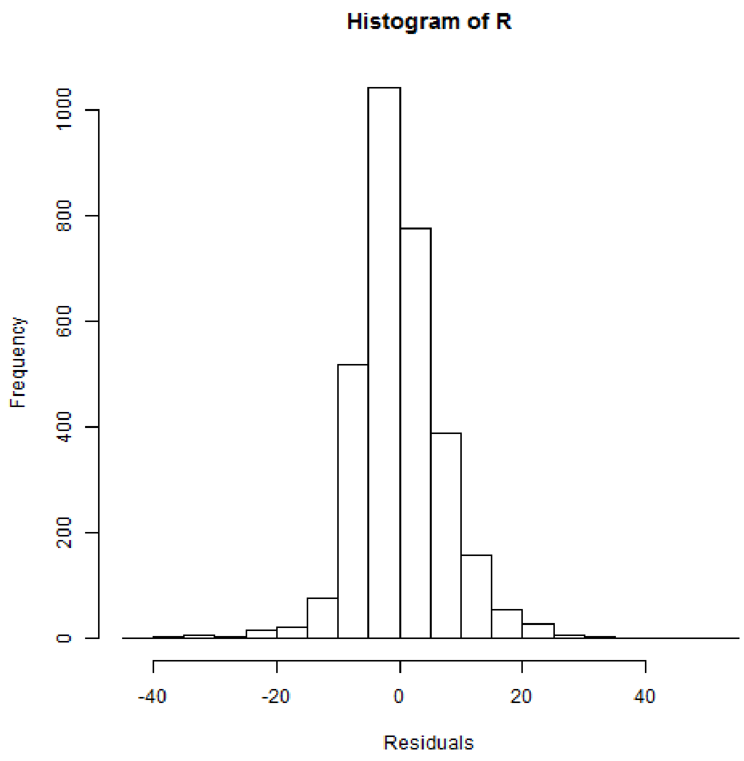
```
##                            Mean        SD     Naive SE Time-series SE
## Intercept           6.687222655 0.1348324 0.0009534088   0.0009534179
## Pop change         -1.125470520 0.1650844 0.0011673232   0.0016245785
## 65+                 0.926873901 0.1989963 0.0014071166   0.0031973178
## African American   -1.581957237 0.1683528 0.0011904339   0.0018478151
## Hispanic           -2.057189236 0.1718018 0.0012148220   0.0022195755
## HS grad             1.803845994 0.2544467 0.0017992097   0.0043317219
## Bachelor's         -6.336370753 0.2669881 0.0018878910   0.0046024483
## Homeownership rate -0.006793062 0.2014922 0.0014247653   0.0029854156
## Home value         -1.360743977 0.2314444 0.0016365590   0.0039646854
## Median income       1.850969079 0.3799842 0.0026868939   0.0091410401
## Poverty             1.486745549 0.2878377 0.0020353198   0.0058878744
```

```
beta_hat <- out[,1]
beta_hat
```

```
##          Intercept         Pop change                65+
##        6.687222655       -1.125470520        0.926873901
##   African American           Hispanic            HS grad
##       -1.581957237       -2.057189236        1.803845994
##         Bachelor's Homeownership rate         Home value
##       -6.336370753       -0.006793062       -1.360743977
##      Median income            Poverty
##        1.850969079        1.486745549
```

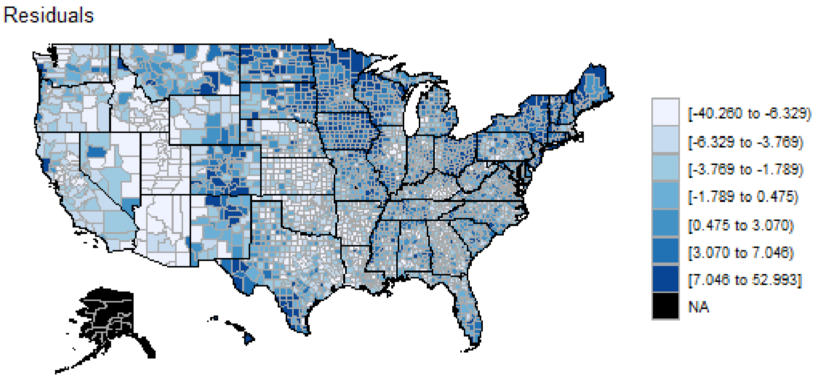
(3b)

```
R <- Y-X%*%beta_hat
hist(R,breaks=25,xlab="Residuals")
```



```
county_plot(fips,R,main="Residuals",units="")
```

```
## Warning in self$bind(): The following regions were missing and are being
## set to NA: 2050, 2105, 29105, 2122, 2150, 2164, 2180, 2188, 2240, 2090,
## 2198, 15005, 2100, 2170, 51515, 2016, 2060, 2290, 2282, 2070, 2110, 2130,
## 2185, 2195, 2220, 2230, 2020, 2068, 2013, 2261, 2270, 2275
```



```
smallest <- rank(R)<=10
largest  <- rank(-R)<=10
all_dat[smallest,2:3]
```

```
##              area_name state_abbreviation
## 586    Franklin County                 ID
## 598     Madison County                 ID
## 2825 Box Elder County                  UT
## 2826      Cache County                 UT
## 2829      Davis County                 UT
## 2835       Juab County                 UT
## 2841 Salt Lake County                  UT
## 2846     Tooele County                 UT
## 2848       Utah County                 UT
## 2852      Weber County                 UT
```

```
all_dat[largest,2:3]
```

```
##              area_name state_abbreviation
## 264    Costilla County                 CO
## 646   Henderson County                 IL
## 851      Howard County                 IA
## 1044    Elliott County                 KY
## 1879   Franklin County                 NY
## 2066    Rolette County                 ND
## 2634      Duval County                 TX
## 2782      Starr County                 TX
## 2822     Zavala County                 TX
## 3139 Menominee County                  WI
```

The histogram shows that the results are reasonably well approximated by a normal distribution but with a few large residuals in both tails. Counties with small (large) residuals suggest that there is some unobserved factor that explains why these counties had a smaller (larger) swing towards the GOP in 2016 than expected by the model.

(3c) Adding random effects might be needed because the residuals cluster by state and so observations within a state are correlated.

```
 state <- as.character(all_dat[,3])
 AKHI  <- state=="AK" | state=="HI" | state=="DC"
 fips  <- fips[!AKHI]
 Y     <- Y[!AKHI]
 X     <- X[!AKHI,]
 state <- state[!AKHI]

 # Assign a numeric id to the counties in each state
 st    <- unique(state)
 id    <- rep(NA,length(Y))
 for(j in 1:48){
   id[state==st[j]]<-j
 }
 data  <- list(n=length(Y),p=ncol(X),Y=Y,X=X,id=id,ns=48)

 model_string <- textConnection("model{

  # Likelihood
  for(i in 1:n){
    Y[i] ~ dnorm(inprod(X[i,],beta[]) + RE[id[i]],tau1)
  }

  # Priors
  for(j in 1:p){beta[j] ~  dnorm(0, 0.0001)}
  for(j in 1:ns){RE[j] ~  dnorm(0, tau2)}
  tau1 ~ dgamma(0.01,0.01)
  tau2 ~ dgamma(0.01,0.01)
}")

init  <- list(beta=beta_hat,RE=rep(0,48),tau2=100,tau1=0.0001)
model <- jags.model(model_string,data = data, inits=init,n.chains=2,quiet=TRUE)
update(model, 10000, progress.bar="none")
params  <- c("beta","RE")
samples <- coda.samples(model,
           variable.names=params,
           n.iter=10000, progress.bar="none")

out                    <- summary(samples)$statistics
rownames(out)[1:48]    <- st
rownames(out)[1:11+48] <- short
out
```

```
##                           Mean        SD     Naive SE Time-series SE
## AL                  -2.5381511 1.2743309 0.0090108799    0.110904056
## AZ                 -12.4493666 1.7509584 0.0123811453    0.102631871
## AR                  -6.3848909 1.2364472 0.0087430021    0.106031478
## CA                  -7.2021255 1.3577861 0.0096009976    0.100651382
## CO                   0.9590420 1.2677279 0.0089641903    0.101151502
## CT                  10.4550614 2.1321338 0.0150764629    0.086233617
## DE                   3.1985464 2.9952264 0.0211794490    0.071983918
## FL                  -3.0890836 1.2595784 0.0089065640    0.111783211
## GA                  -4.2754932 1.1655488 0.0082416743    0.109428435
## ID                 -11.7541567 1.3412377 0.0094839826    0.109955283
## IL                   2.6720245 1.1856482 0.0083837991    0.109053089
## IN                   2.7544112 1.2035987 0.0085107284    0.104618921
## IA                  10.5370108 1.1949956 0.0084498951    0.108195136
## KS                  -5.8007093 1.1932709 0.0084376997    0.111860668
## KY                  -1.0444404 1.1862762 0.0083882391    0.112640964
## LA                  -4.8517074 1.2916793 0.0091335519    0.103523267
## ME                   7.4894848 1.6932298 0.0119729425    0.100826884
## MD                   3.2199695 1.5387222 0.0108804088    0.097956564
## MA                   1.6287385 1.8022288 0.0127436821    0.093138298
## MI                   1.5720252 1.2235019 0.0086514651    0.110432781
## MN                   7.0158118 1.2002773 0.0084872419    0.106467596
## MS                  -2.7388707 1.2786247 0.0090412419    0.102093870
## MO                   2.5958474 1.1780829 0.0083303042    0.111662616
## MT                  -0.1989164 1.2844631 0.0090825259    0.108614906
## NE                  -2.3311245 1.2167744 0.0086038943    0.109919514
## NV                  -3.9214415 1.6717156 0.0118208142    0.096078759
## NH                   7.4671302 1.9576104 0.0138423960    0.096112055
## NJ                   8.9651218 1.6010711 0.0113212824    0.102565729
## NM                  -6.3684831 1.4928678 0.0105561691    0.103258491
## NY                   9.4245590 1.2635537 0.0089346736    0.109887998
## NC                  -1.7326954 1.1983510 0.0084736215    0.103373061
## ND                   6.0626363 1.3117506 0.0092754777    0.101749142
## OH                   7.3768663 1.2061752 0.0085289463    0.101534226
## OK                  -4.8948496 1.2372011 0.0087483330    0.112427897
## OR                  -4.3593108 1.3855098 0.0097970337    0.107839352
## PA                   3.0851966 1.2497493 0.0088370620    0.111352968
## RI                  12.5778386 2.5381247 0.0179472518    0.082685986
## SC                  -1.7629445 1.3634230 0.0096408563    0.107107642
## SD                   4.4638590 1.2575194 0.0088920052    0.103763292
## TN                  -0.1320741 1.2086381 0.0085463622    0.104727595
## TX                  -5.1988844 1.1462767 0.0081054001    0.117074102
## UT                 -25.5622541 1.4623855 0.0103406270    0.103652185
## VT                  10.3841205 1.7584378 0.0124340329    0.094236396
## VA                   0.3784467 1.1709193 0.0082796500    0.111975330
## WA                  -4.8958555 1.3801111 0.0097588589    0.107715722
## WV                   1.3058359 1.2871777 0.0091017211    0.106346024
## WI                   7.2193501 1.2422961 0.0087843600    0.105822482
## WY                  -2.9684350 1.5417470 0.0109017978    0.104624438
## Intercept            7.0874662 1.0581376 0.0074821627    0.110332157
## Pop change          -0.3161565 0.1356184 0.0009589666    0.001683968
## 65+                  0.7046091 0.1661628 0.0011749487    0.003260125
## African American    -0.6137066 0.1655547 0.0011706485    0.002536466
## Hispanic            -0.1314370 0.1875373 0.0013260887    0.003527172
## HS grad              1.9374214 0.2264082 0.0016009481    0.004988527
## Bachelor's          -6.3073453 0.2105940 0.0014891241    0.004193702
## Homeownership rate   0.7231877 0.1676885 0.0011857365    0.002996933
## Home value          -0.6096993 0.2290344 0.0016195178    0.004899355
## Median income       -0.1700349 0.3072504 0.0021725881    0.008700188
## Poverty              1.5699952 0.2180461 0.0015418191    0.004671809
```

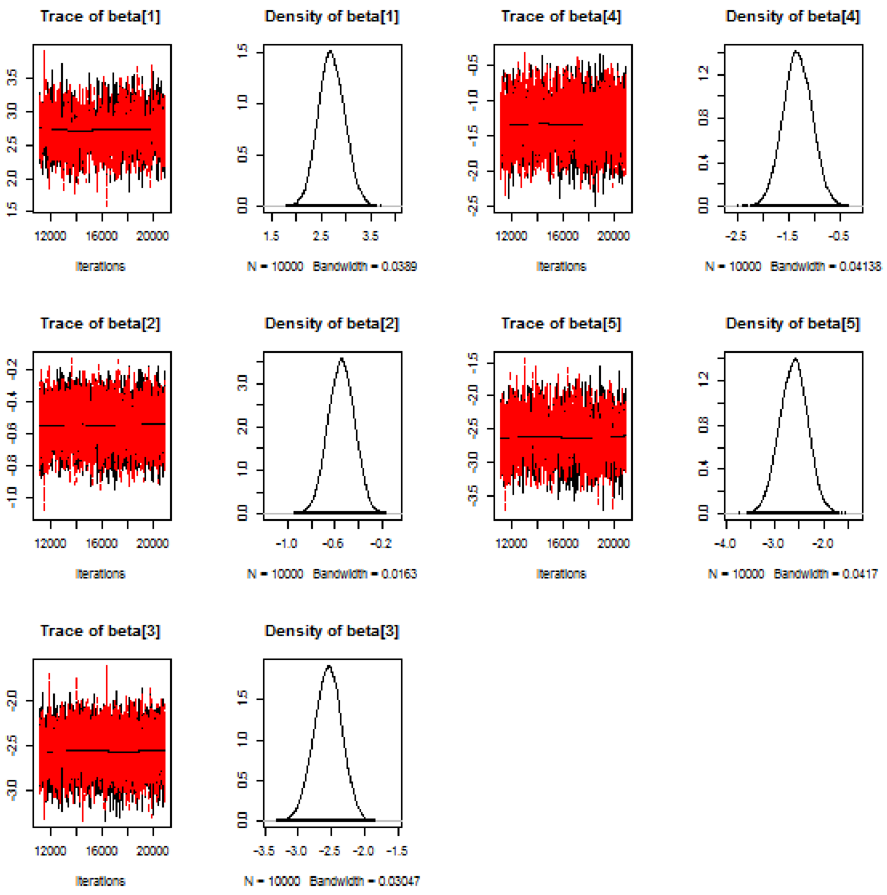
The states with small (large) random effects had a smaller (larger) swing towards the GOP than expected by our model. The state with smallest posterior mean random effect is Utah; the state with largest posterior mean random effect is Rhode Island.

(5) We fit the logistic regression model

$$\text{logit}[\text{Prob}(Y_i = 1)] = \sum_{j=1}^{p} X_{ij}\beta_j$$

with uninformative priors $\beta_j \sim \text{Normal}(0, 1000)$.

```
 library("titanic")
 dat    <- titanic_train
 Y      <- dat[,2]
 age    <- dat[,6]
 gender <- dat[,5]
 class  <- dat[,3]
 X      <- cbind(1,scale(age),
           ifelse(gender=="male",1,0),
           ifelse(class==2,1,0),
           ifelse(class==3,1,0))
 colnames(X) <- c("Intercept","Age","Gender","Class=2","Class=3")
 miss <- is.na(rowSums(X))
 X    <- X[!miss,]
 Y    <- Y[!miss]


 library(rjags)
 data <- list(n=nrow(X),p=ncol(X),Y=Y,X=X)

 model_string <- textConnection("model{

  # Likelihood
  for(i in 1:n){
    Y[i] ~ dbern(prob[i])
    logit(prob[i]) =  inprod(X[i,],beta[])
  }

  # Priors
  for(j in 1:p){beta[j] ~  dnorm(0, 0.01)}
}")

model <- jags.model(model_string,data = data, n.chains=2,quiet=TRUE)
update(model, 10000, progress.bar="none")
params  <- c("beta")
samples <- coda.samples(model,
           variable.names=params,
           n.iter=10000, progress.bar="none")
out      <- summary(samples)$quantiles
plot(samples)
```



```
rownames(out)<-colnames(X)
round(out,2)
```

```
##              2.5%   25%   50%   75% 97.5%
## Intercept  2.21  2.53  2.71  2.89  3.25
## Age       -0.76 -0.62 -0.55 -0.47 -0.33
## Gender    -2.96 -2.69 -2.55 -2.41 -2.15
## Class=2   -1.88 -1.52 -1.33 -1.14 -0.77
## Class=3   -3.18 -2.81 -2.61 -2.42 -2.06
```

The posterior medians are negative and 95% intervals exclude zero for all of the covariates. Therefore, the profile of the passenger with highest probability of survival is a young women in first class.

(7) Gibbs sampling is a good choice because all of the full conditional distributions are conjugate. For initial values one might set α_j to the group mean $\bar{Y}_j = \sum_{i=1}^{n} Y_{ij}/n$, τ^2 to the sample variance of the $\bar{Y}_j$, and σ^2 to the variance of the $Y_{ij} - \bar{Y}_j$. At iteration t the Gibbs sampler would update the parameters as

$$\alpha_j \,|\, \text{rest} \sim \text{Normal}\left(\frac{\sum_{i=1}^{n} Y_{ij}}{n + \sigma^2/\tau^2}, \frac{\sigma^2}{n + \sigma^2/\tau^2}\right)$$

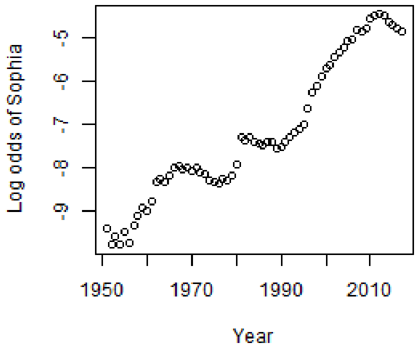
$$\sigma^2 \,|\, \text{rest} \sim \text{InvGamma}\left(nm/2 + a, \sum_{i=1}^{n}\sum_{j=1}^{m}(Y_{ij} - \alpha_j)^2/2 + b\right)$$

$$\tau^2 \,|\, \text{rest} \sim \text{InvGamma}\left(m/2 + a, \sum_{j=1}^{m}\alpha_j^2/2 + b\right)$$

(9a)

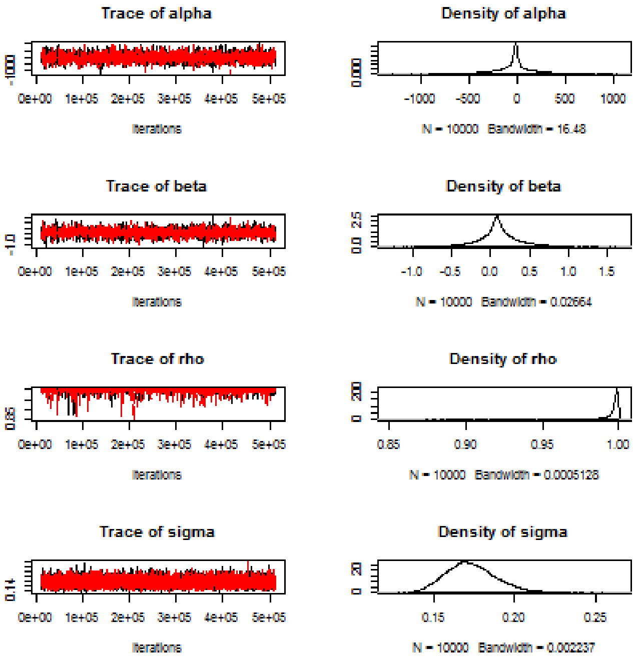
The mean trend has intercept α and slope β, so the average increase in log odds per year is β. The parameter ρ controls autocorrelation with $\rho = 0$ giving indepedence across years and large ρ giving strong dependence. Finally, σ^2 controls the variance of the process.

```
library(babynames)
dat <- babynames
dat <- dat[dat$name=="Sophia" &
dat$sex=="F" &
dat$year>1950,]
yr <- dat$year
p <- dat$prop
t <- dat$year - 1950
Y <- log(p/(1-p))

plot(t+1950,Y,xlab="Year",ylab="Log odds of Sophia")
```



(9b)

```
 library(rjags)
 data <- list(n=length(Y),Y=Y)

 model_string <- textConnection("model{

  # Likelihood
   for(t in 2:n){
     Y[t]      ~ dnorm(meanY[t],tau)
     meanY[t] = alpha + beta*t +
                rho*(Y[t-1] - alpha - beta*(t-1))
   }

  # Priors
   alpha   ~ dnorm(0,0.00001)
   beta    ~ dnorm(0,0.00001)
   rho     ~ dbeta(1,1)
   tau     ~ dgamma(0.01,0.01)
   sigma  <- 1/sqrt(tau)
}")

model <- jags.model(model_string,data = data, n.chains=2,quiet=TRUE)
update(model, 10000, progress.bar="none")
params  <- c("alpha","beta","rho","sigma")
samples <- coda.samples(model,
           variable.names=params,
           n.iter=500000, thin=50,progress.bar="none")
plot(samples)
```



```
summary(samples)
```

```
##
## Iterations = 11050:511000
## Thinning interval = 50
## Number of chains = 2
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean        SD  Naive SE Time-series SE
## alpha -19.6937 210.41735 1.4878754      3.6719609
## beta     0.1008   0.26276 0.0018580      0.0053501
## rho      0.9929   0.01457 0.0001030      0.0006751
## sigma    0.1732   0.01561 0.0001104      0.0001104
##
## 2. Quantiles for each variable:
##
##              2.5%        25%       50%     75%    97.5%
## alpha -478.4752 -100.62187 -12.39686 50.3556 465.8114
## beta     -0.4593   -0.02144   0.08826  0.2226   0.6873
## rho       0.9493    0.99428   0.99773  0.9990   0.9999
## sigma     0.1459    0.16225   0.17188  0.1828   0.2070
```

```
effectiveSize(samples)
```

```
##      alpha       beta        rho      sigma
##  3318.6469  2432.6914   503.3292 20000.0000
```

```
gelman.diag(samples)
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## alpha       1.00       1.00
## beta        1.00       1.01
## rho         1.05       1.09
## sigma       1.00       1.00
##
## Multivariate psrf
##
## 1.01
```

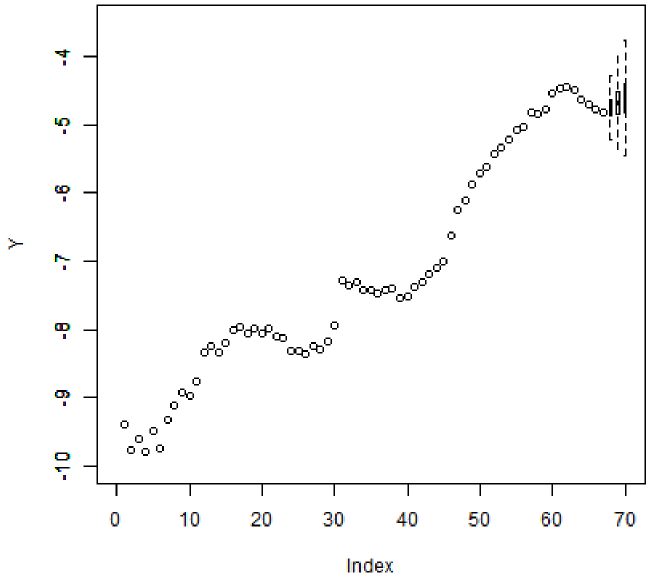
Convergence is slow (because $\rho \approx 1$ and there is strong correlation between observations) and requires extremely long chains.

(9c) The prediction for 2020 depends on the values in 2018 and 2019. So we first sample 2018, then 2019, and then 2020.

```
# Extract the posterior samples

samps  <- rbind(samples[[1]],samples[[2]])
samps[1:2,]
```

```
##          alpha      beta       rho     sigma
## [1,] -337.1472 0.3442191 0.9990959 0.1745659
## [2,] -209.2539 0.4841456 0.9976049 0.1776975
```

```
S      <- nrow(samps)
alpha  <- samps[,1]
beta   <- samps[,2]
rho    <- samps[,3]
sigma  <- samps[,4]

# Make predictions

e1     <- rnorm(S,0,sigma)
e2     <- rnorm(S,0,sigma)
e3     <- rnorm(S,0,sigma)
Y_2018 <- alpha + beta*68 + rho*( Y[67]-alpha - beta*67) + e1
Y_2019 <- alpha + beta*69 + rho*(Y_2018-alpha - beta*68) + e2
Y_2020 <- alpha + beta*70 + rho*(Y_2019-alpha - beta*69) + e3

# Plot the results
plot(Y,xlim=c(0,70),ylim=c(-10,-3.5))
boxplot(Y_2018,add=TRUE,at=68,outline=FALSE)
boxplot(Y_2019,add=TRUE,at=69,outline=FALSE)
boxplot(Y_2020,add=TRUE,at=70,outline=FALSE)
```



The prediction intervals seem reasonable.