Homework #6

Dustin Leatherman

October 25, 2020

Load Parameters

Scale the covariates and set upfront settings

```
X <- scale(election[,-1])
Z <- election$Z

n <- length(Z)
p <- ncol(X)
names <- colnames(X)

data <- list(Z=Z,X=X,n=n,p=p)
params <- c("beta")

# Settings (automatically calculates the number of iterations needed based on inputs)
nBurn <- 10000
nChains <- 2
nSave <- 4000
nThin <- 10
nIter <- ceiling((nSave*nThin)/nChains)
```

1 & 2

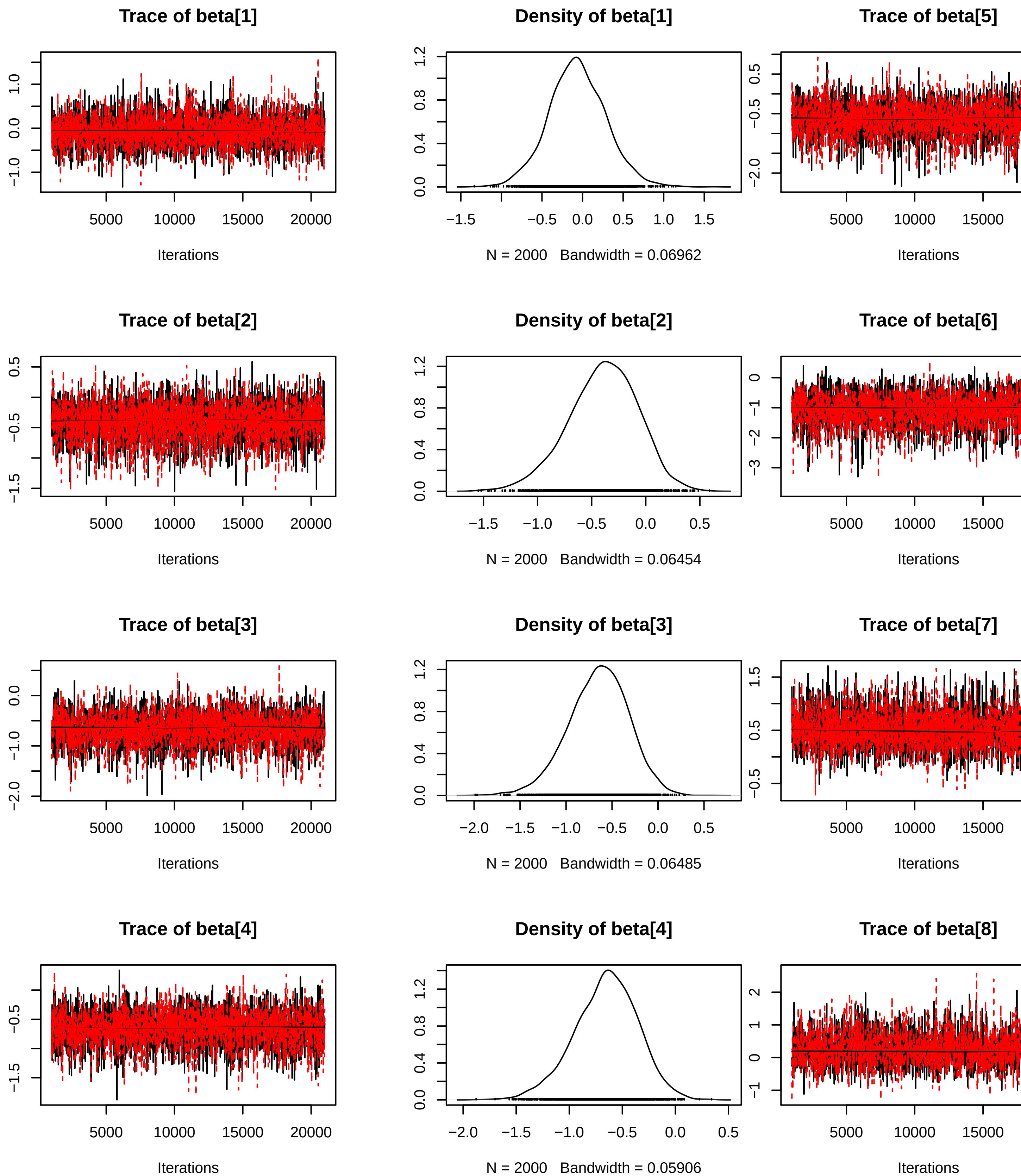
> Fit the model using $\tau = 1$ and $\tau = 100$. Assess convergence of samplers for each prior.

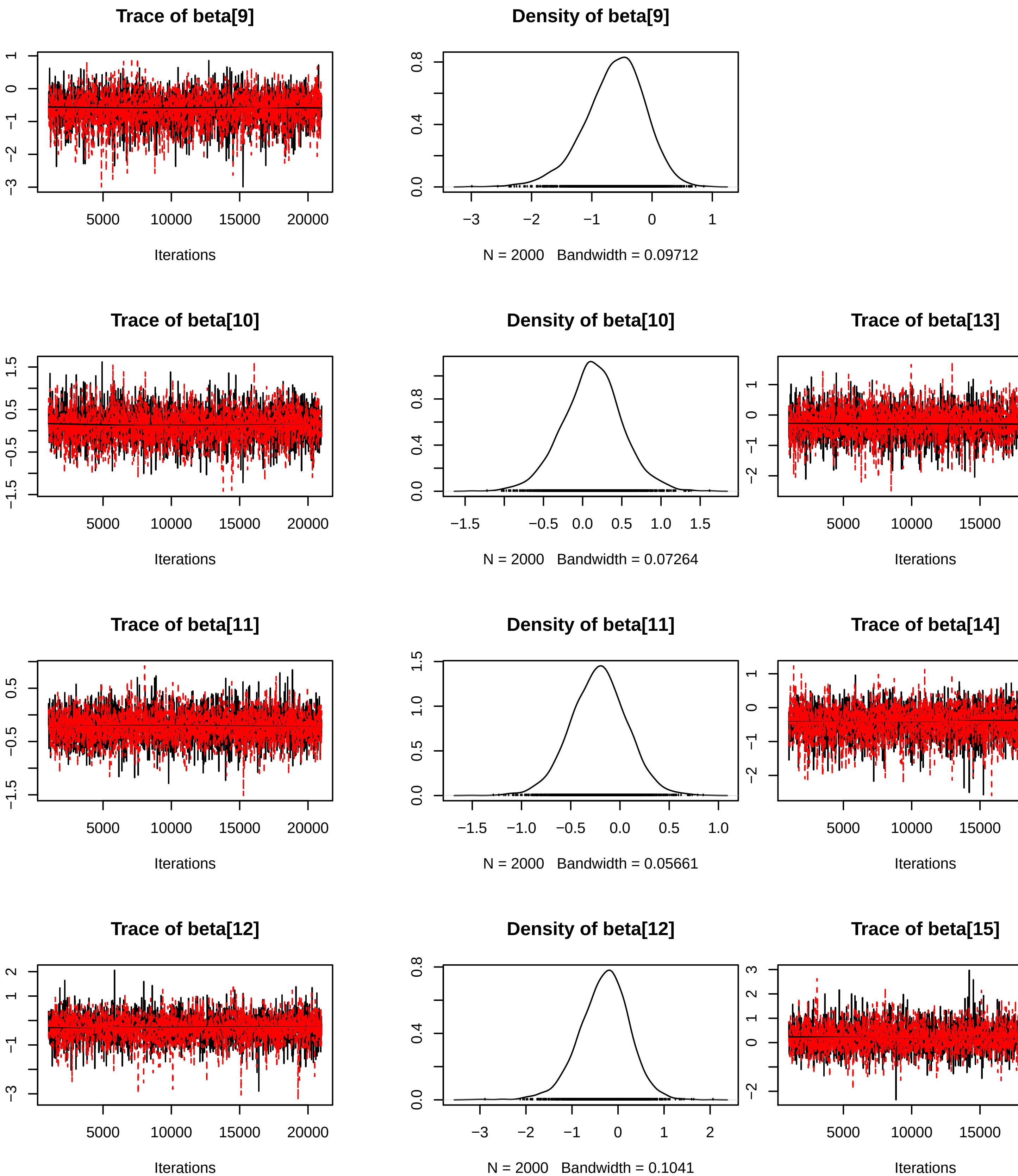
Let's fit some models using JAGS.

$\tau = 100$

```
model_string <- textConnection("model{
  for(i in 1:n) {
    # Likelihood
    Z[i] ~ dbern(prob[i])
    prob[i] <- 1 / (1 + exp(-a[i]))
    a[i] <- alpha + inprod(X[i,],beta[])
  }

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

  alpha ~ dnorm(0,0.01)
  tau ~ dgamma(0.01, 0.01)
```

```
}")

model <- jags.model(model_string,data=data,n.chains=nChains,quiet=TRUE)
update(model,burn=nBurn,progress.bar="none")
samples1 <- coda.samples(model,variable.names=params,thin=nThin,n.iter=nIter,
                         progress.bar="none")

plot(samples1)
```

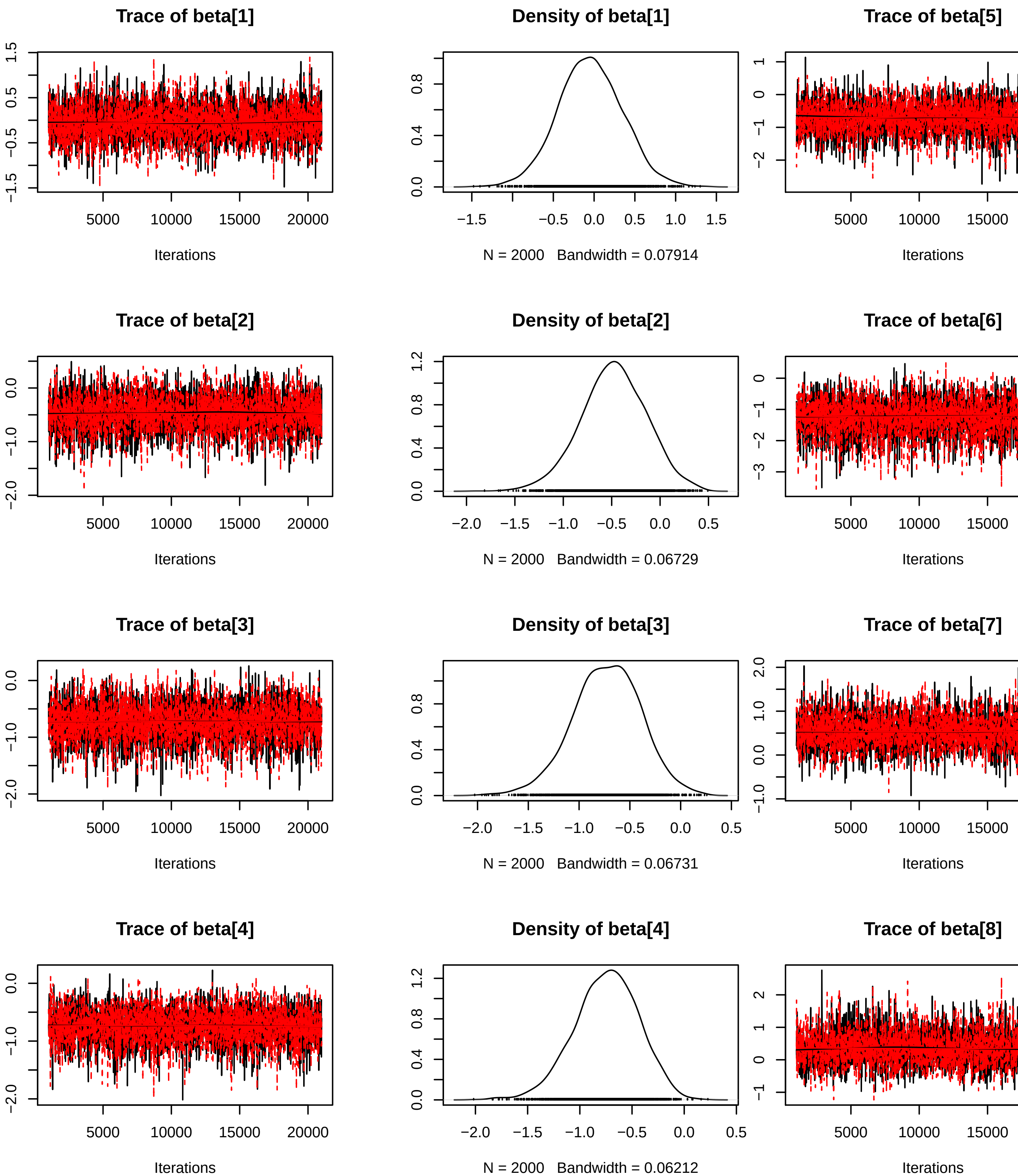

Trace of beta[1]
Iterations
Density of beta[1]
N = 2000 Bandwidth = 0.06962
Trace of beta[5]
Iterations
Trace of beta[2]
Iterations
Density of beta[2]
N = 2000 Bandwidth = 0.06454
Trace of beta[6]
Iterations
Trace of beta[3]
Iterations
Density of beta[3]
N = 2000 Bandwidth = 0.06485
Trace of beta[7]
Iterations
Trace of beta[4]
Iterations
Density of beta[4]
N = 2000 Bandwidth = 0.05906
Trace of beta[8]
Iterations

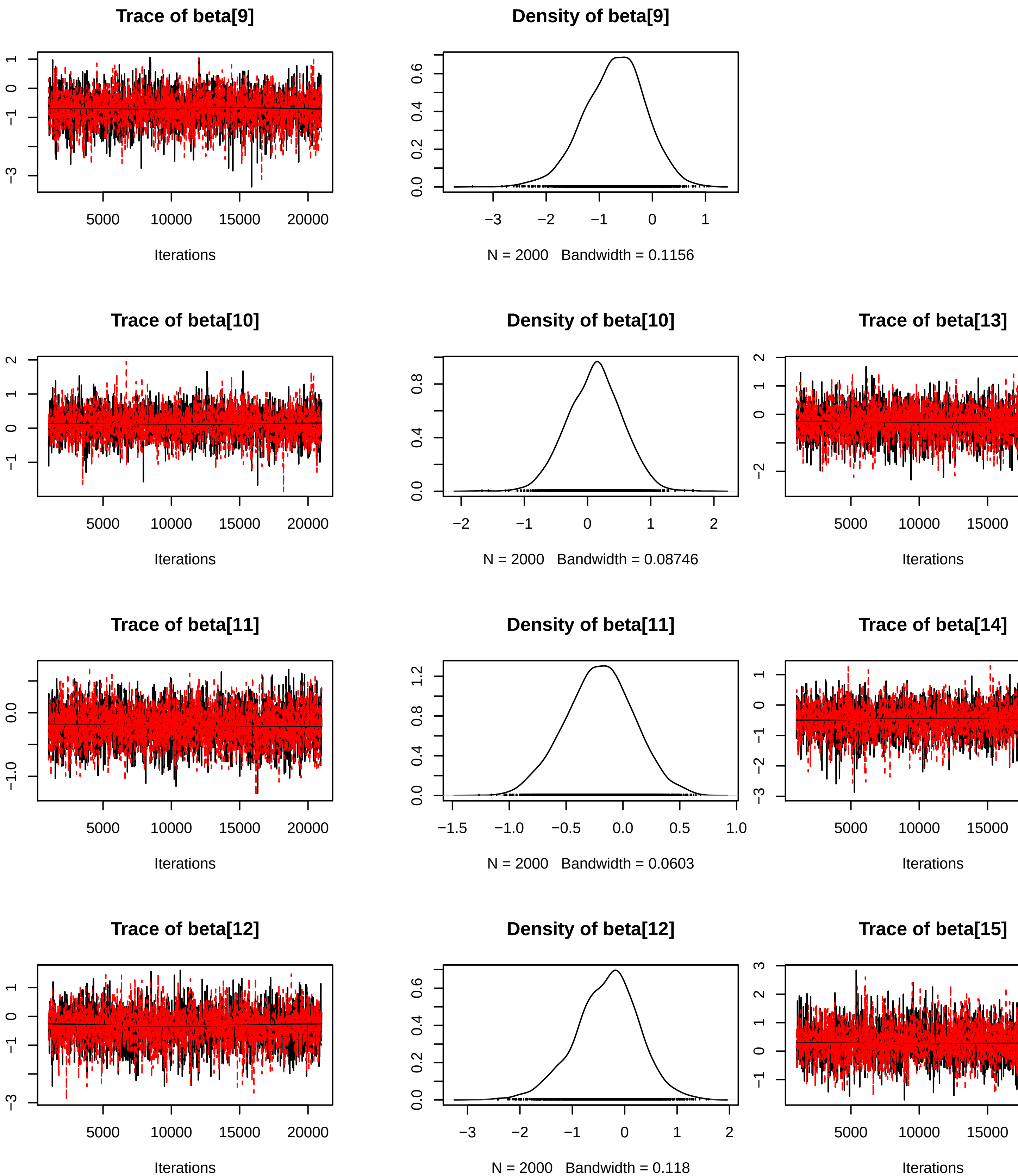


The trace plots don't have any noticable patterns and are roughly caterpillar-shaped which provide a good indication that the plots have converged.

$\tau = 1$

```
model_string <- textConnection("model{
  for(i in 1:n) {
    # Likelihood
    Z[i] ~ dbern(prob[i])
    prob[i] <- 1 / (1 + exp(-a[i]))
    a[i] <- alpha + inprod(X[i,],beta[])
  }

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

  alpha ~ dnorm(0, 1)
  tau ~ dgamma(1, 1)
}")

model <- jags.model(model_string,data=data,n.chains=nChains,quiet=TRUE)
update(model,burn=nBurn,progress.bar="none")
samples2 <- coda.samples(model,variable.names=params,thin=nThin,n.iter=nIter,
                         progress.bar="none")

plot(samples2)
```


Trace of beta[1]
Density of beta[1]
Trace of beta[5]
Iterations
N = 2000 Bandwidth = 0.07914
Trace of beta[2]
Density of beta[2]
Trace of beta[6]
Iterations
N = 2000 Bandwidth = 0.06729
Trace of beta[3]
Density of beta[3]
Trace of beta[7]
Iterations
N = 2000 Bandwidth = 0.06731
Trace of beta[4]
Density of beta[4]
Trace of beta[8]
Iterations
N = 2000 Bandwidth = 0.06212



The trace plots don't have any noticable patterns and are roughly caterpillar-shaped which provide a good indication that the plots have converged.

Summary Statistics

```
round(effectiveSize(samples1),1) %>%
  as_tibble(rownames = NA) %>%
  rownames_to_column() %>%
  kable(
    col.names = c("Variable", "ESS"),
    caption = "Effective Sample Size for Tau = 100"
  ) %>%
  kable_styling(full_width = T, bootstrap_options = "striped", latex_options = "hold_position")
```

Table 1: Effective Sample Size for Tau = 100

Variable	ESS
beta[1]	3855.0
beta[2]	3316.4
beta[3]	3471.3
beta[4]	3401.7
beta[5]	3584.2
beta[6]	3031.5
beta[7]	3806.5
beta[8]	2794.5
beta[9]	3007.7
beta[10]	3571.2
beta[11]	4393.0
beta[12]	4000.0
beta[13]	3609.7
beta[14]	3697.1
beta[15]	3577.0

```
round(effectiveSize(samples2),1) %>%
  as_tibble(rownames = NA) %>%
  rownames_to_column() %>%
  kable(
    col.names = c("Variable", "ESS"),
    caption = "Effective Sample Size for Tau = 1"
  ) %>%
  kable_styling(full_width = T, bootstrap_options = "striped", latex_options = "hold_position")
```

The Effective Sample Sizes are large enough given the number of iterations that we can feel confident that the chains appropriately fit the underlying distributions.

3

Compare the distributions of β_j under these two priors. Are the results sensitive to the prior?

```
# Format the model summary
sum1 <- summary(samples1)
rownames(sum1$statistics) <- names
```

Table 2: Effective Sample Size for Tau = 1

Variable	ESS
beta[1]	3836.2
beta[2]	3562.5
beta[3]	4000.0
beta[4]	3753.0
beta[5]	3687.3
beta[6]	3486.4
beta[7]	5035.4
beta[8]	3152.6
beta[9]	3090.0
beta[10]	3288.1
beta[11]	4000.0
beta[12]	4004.2
beta[13]	3908.6
beta[14]	3723.0
beta[15]	3781.9

```r
rownames(sum1$quantiles) <- names
sum1$statistics <- round(sum1$statistics,3)
sum1$quantiles <- round(sum1$quantiles,3)
sum1
```

```
##
## Iterations = 1011:21001
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean    SD Naive SE Time-series SE
## PopPctChange        -0.064 0.352    0.006          0.006
## Age65PlusPct        -0.395 0.320    0.005          0.006
## BlackPct            -0.650 0.326    0.005          0.006
## HispanicPct         -0.654 0.296    0.005          0.005
## HSgradPct           -0.643 0.435    0.007          0.007
## BachGradPct         -1.060 0.563    0.009          0.010
## OwnHomeRate          0.504 0.354    0.006          0.006
## MedianHomeValue      0.244 0.503    0.008          0.010
## MedianHHInc         -0.619 0.504    0.008          0.009
## PovertyPct           0.125 0.379    0.006          0.006
## RetailSalesPerCap   -0.214 0.286    0.005          0.004
## PopPerSqMile        -0.302 0.544    0.009          0.009
## Vets                -0.267 0.503    0.008          0.008
## Manufacture         -0.431 0.479    0.008          0.008
## MerchantSales        0.211 0.553    0.009          0.009
##
## 2. Quantiles for each variable:
##
##                       2.5%    25%    50%    75%  97.5%
```

```
## PopPctChange      -0.755 -0.297 -0.070  0.165  0.650
## Age65PlusPct      -1.074 -0.600 -0.380 -0.170  0.180
## BlackPct          -1.341 -0.856 -0.635 -0.425 -0.042
## HispanicPct       -1.276 -0.843 -0.639 -0.451 -0.112
## HSgradPct         -1.595 -0.911 -0.629 -0.349  0.162
## BachGradPct       -2.300 -1.395 -1.004 -0.657 -0.101
## OwnHomeRate       -0.155  0.266  0.482  0.725  1.266
## MedianHomeValue   -0.646 -0.093  0.205  0.531  1.371
## MedianHHInc       -1.738 -0.923 -0.578 -0.278  0.262
## PovertyPct        -0.629 -0.117  0.125  0.365  0.893
## RetailSalesPerCap -0.780 -0.402 -0.211 -0.026  0.343
## PopPerSqMile      -1.428 -0.636 -0.273  0.056  0.714
## Vets              -1.301 -0.575 -0.257  0.051  0.722
## Manufacture       -1.472 -0.718 -0.386 -0.107  0.404
## MerchantSales     -0.846 -0.148  0.194  0.546  1.334
```

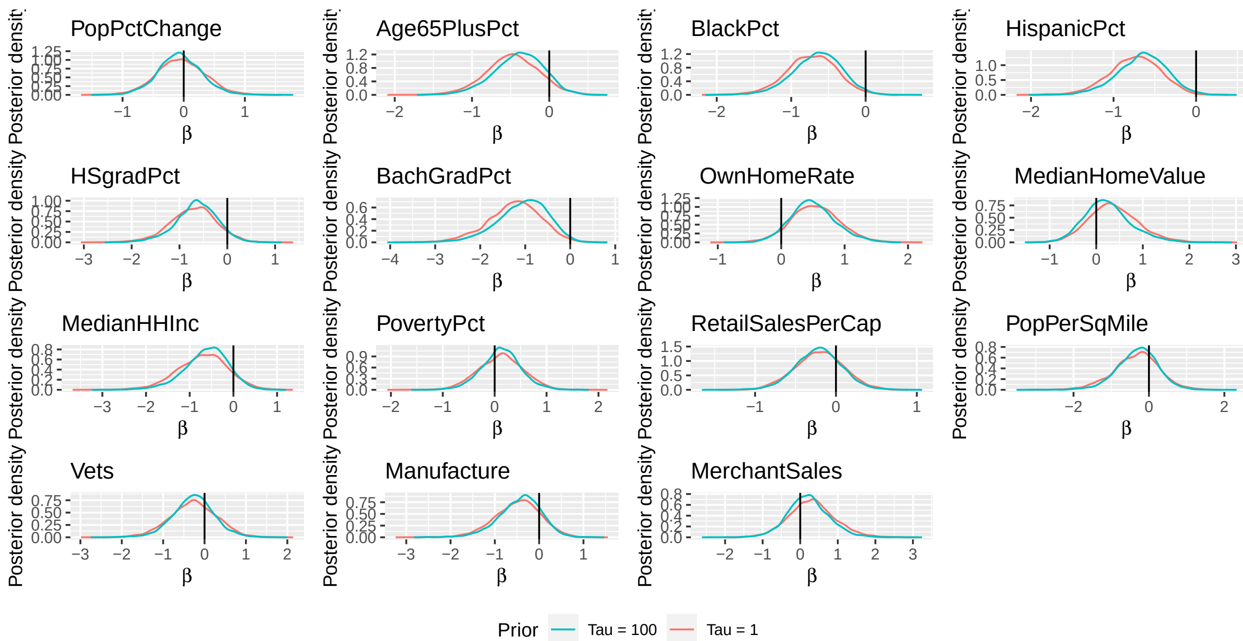
```
# Format the model summary
sum2 <- summary(samples2)
rownames(sum2$statistics) <- names
rownames(sum2$quantiles) <- names
sum2$statistics <- round(sum2$statistics,3)
sum2$quantiles <- round(sum2$quantiles,3)
sum2
```

```
##
## Iterations = 1011:21001
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                     Mean    SD Naive SE Time-series SE
## PopPctChange      -0.047 0.394    0.006          0.006
## Age65PlusPct      -0.479 0.340    0.005          0.006
## BlackPct          -0.729 0.337    0.005          0.005
## HispanicPct       -0.745 0.308    0.005          0.005
## HSgradPct         -0.714 0.485    0.008          0.008
## BachGradPct       -1.241 0.587    0.009          0.010
## OwnHomeRate        0.539 0.382    0.006          0.006
## MedianHomeValue    0.373 0.530    0.008          0.009
## MedianHHInc       -0.708 0.579    0.009          0.010
## PovertyPct         0.119 0.434    0.007          0.008
## RetailSalesPerCap -0.205 0.299    0.005          0.005
## PopPerSqMile      -0.340 0.610    0.010          0.010
## Vets              -0.272 0.567    0.009          0.009
## Manufacture       -0.487 0.521    0.008          0.009
## MerchantSales      0.313 0.612    0.010          0.010
##
## 2. Quantiles for each variable:
##
##                     2.5%    25%    50%    75%  97.5%
## PopPctChange      -0.812 -0.311 -0.050  0.215  0.733
## Age65PlusPct      -1.168 -0.696 -0.474 -0.249  0.177
```

```
## BlackPct         -1.423 -0.946 -0.720 -0.499 -0.092
## HispanicPct      -1.380 -0.945 -0.732 -0.530 -0.186
## HSgradPct        -1.719 -1.022 -0.697 -0.395  0.204
## BachGradPct      -2.487 -1.608 -1.206 -0.843 -0.178
## OwnHomeRate      -0.199  0.280  0.532  0.788  1.305
## MedianHomeValue  -0.589  0.013  0.341  0.704  1.511
## MedianHHInc      -1.904 -1.087 -0.684 -0.320  0.360
## PovertyPct       -0.730 -0.175  0.127  0.406  0.957
## RetailSalesPerCap -0.802 -0.405 -0.199 -0.003  0.365
## PopPerSqMile     -1.629 -0.722 -0.299  0.061  0.820
## Vets             -1.404 -0.638 -0.262  0.111  0.805
## Manufacture      -1.578 -0.822 -0.453 -0.136  0.460
## MerchantSales    -0.854 -0.083  0.304  0.692  1.572
```

Compare the Fits

```r
library(cowplot)
plot_list <- list()

for(j in 1:p){

  # Collect the MCMC iteration from both chains for the three priors
  s1 <- c(samples1[[1]][,j],samples1[[2]][,j])
  s2 <- c(samples2[[1]][,j],samples2[[2]][,j])

  # Get smooth density estimates for each prior
  d1 <- density(s1)
  d2 <- density(s2)

  Prior <- c(rep("Tau = 100",length(d1$x)),
             rep("Tau = 1",length(d2$x)))
  x <- c(d1$x,d2$x)
  y <- c(d1$y,d2$y)
  d.data <- data.frame(x=x,y=y,Prior=Prior)

  # Plot the density estimates
  max.y <- max(y)
  plot.title <- names[j]
  g <- ggplot(d.data,aes(x=x,y=y,color=Prior))+geom_line()+
    labs(x=expression(beta),y="Posterior density")+ggtitle(plot.title)+
    ylim(c(0,max.y))+geom_vline(xintercept=0)
  plot_list[[j]] <- g+theme(legend.position="none")
}
prow <- plot_grid(plotlist=plot_list,nrow=4)
legend <- get_legend(g+guides(color=guide_legend(reverse=TRUE,nrow=1))+
                     theme(legend.position="bottom"))
plot_grid(prow,legend,nrow=5,rel_heights = c(1,0.1))
```



The fits are close but still provide a potentially noticable difference for some covariates. e.g. BachGradPct, HispanicPct, MerchantSales. This shows that there is *some* sensitivity to the prior for some covariates.