# Analysis of the 2016 US Presidential Election

he data for this analysis come from Tony McGovern. The response variable, $Y_i$ , is the percentage change in Republican (GOP) support from 2012 to 2016, i.e.,

$$100\left(\frac{\%\text{ in }2016}{\%\text{ in }2012}-1\right),$$

in county $i = 1, \ldots, n$.

The $p = 10$ covariates $X_{ij}$ are county-level census variables obtained from Kaggle are:

Population, percent change - April 1, 2010 to July 1, 2014

Persons 65 years and over, percent, 2014

Black or African American alone, percent, 2014

Hispanic or Latino, percent, 2014

High school graduate or higher, percent of persons age 25+, 2009-2013

Bachelor's degree or higher, percent of persons age 25+, 2009-2013

Homeownership rate, 2009-2013

Median value of owner-occupied housing units, 2009-2013

Median household income, 2009-2013

Persons below poverty level, percent, 2009-2013

For a county in state $s$, we assume the linear model

$$Y_i = \beta_{0s} + \sum_{j=1}^{p} X_i\beta_{sj} + \varepsilon_i,$$

where $\beta_{js}$ is the effect of covariate $j$ in state $s$. We compare three models for the $\beta_{js}$.

1. Constant slopes: $\beta_{js} = \beta_j$ for all counties.
2. Varying slopes with uninformative priors: $\beta_{js} \sim \text{Normal}(0, 100)$
3. Varying slopes with informative priors: $\beta_{js} \sim \text{Normal}(\mu_j, \sigma_j^2)$.

In the third model, the means ($\mu_j$) and variances ($\sigma_j^2$) are assigned prior distributions and estimated based on the data, allowing for information sharing across states through the prior. The three methods are evaluated using the Deviance Information Criterion (DIC), and the final results are compared across the different models.

```r
# Load required libraries
library(choroplethr)
```

```
## Loading required package: acs
```

```
## Loading required package: stringr
```

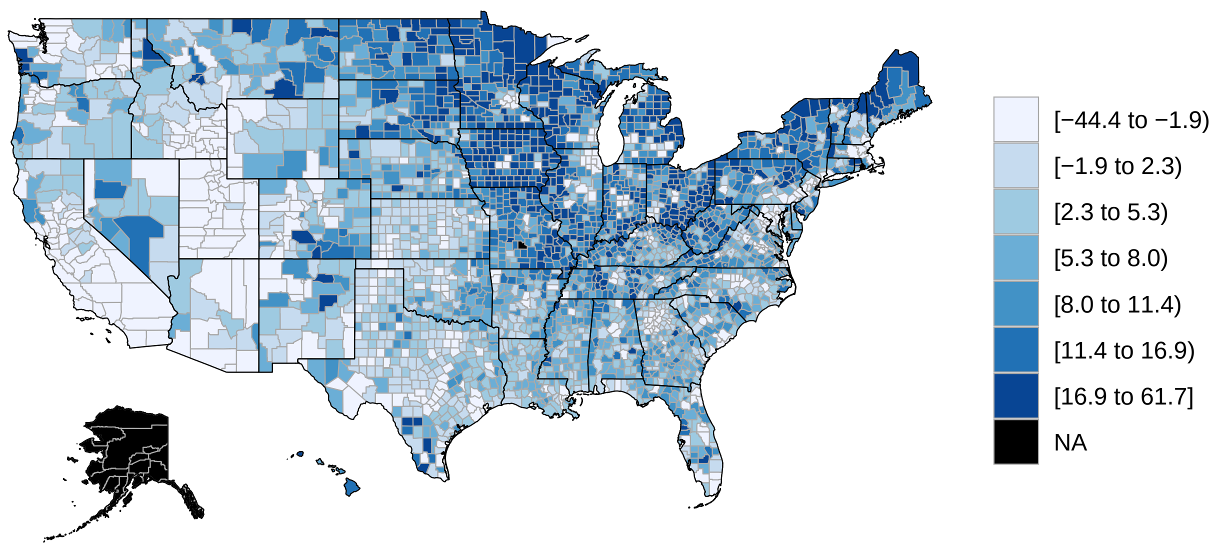
```
## Loading required package: XML
```

```
##
## Attaching package: 'acs'

## The following object is masked from 'package:base':
##
##     apply
```

```r
library(choroplethrMaps) # Required for county_choropleth
# Load the dataset
load("/Users/kamaladadashova/Documents/DoctoralCourses/Applied Bayesian Statistics/Lecture Notes with A
# Standardize the covariates and add an intercept
X = scale(X)
X = cbind(Intercept = 1, X)
# Define short names for the covariates
short_names = c("Intercept", "Pop change", "65+", "African American",
                "Hispanic", "HS grad", "Bachelor's",
                "Homeownership rate", "Home value",
                "Median income", "Poverty")
colnames(X) = short_names

# Define a function to create county maps
county_plot = function(fips, Y, main = "", units = "") {
  data = data.frame(region = fips, value = Y)
  county_choropleth(data, title = main, legend = units)
}
# Plot the map
county_plot(fips, Y, main = "Percent increase in GOP support", units = "")
```

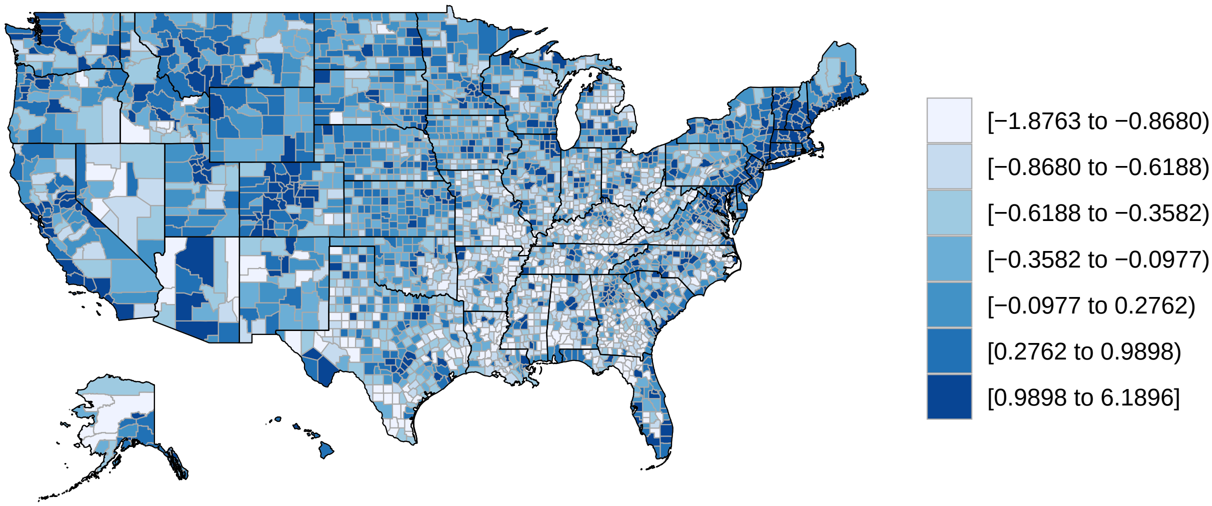
```
## Warning in self$bind(): The following regions were missing and are being set to
## NA: 2050, 2105, 29105, 2122, 2150, 2164, 2180, 2188, 2240, 2090, 2198, 15005,
## 2100, 2170, 51515, 2016, 2060, 2290, 2282, 2070, 2110, 2130, 2185, 2195, 2220,
## 2230, 2020, 2068, 2013, 2261, 2270, 2275
```

Percent increase in GOP support




```r
# Plot the map for the Bachelor's covariate (X[,7])
county_plot(fips, X[,7], main = "Bachelor's", units = "")
```

Bachelor's




```
# Remove AK, HI and DC due to missing data
set.seed(5656)
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
}
n = length(Y) # number of counties
N = 48        # number of states
p = ncol(X)   # number of features
iters = 50000
burn  = 10000
```

## Model 1: Constant slopes

```
model1_string = "model{
```

```
   # Likelihood
   for(i in 1:n){
      Y[i]   ~ dnorm(mu[i],taue)
      mu[i] <- inprod(X[i,],beta[])
   }
   # Priors
   for(j in 1:p){beta[j] ~ dnorm(0,0.01)}
   taue ~ dgamma(0.1,0.1)
   sig <- 1/sqrt(taue)

   # WAIC calculations
   for(i in 1:n){
     like[i]    <- dnorm(Y[i],mu[i],taue)
   }
 }"

library(rjags)
```

```
## Loading required package: coda

## Linked to JAGS 4.3.1

## Loaded modules: basemod,bugs
```

```
# Load the model
dat    = list(Y=Y,n=n,X=X,p=p)
init   = list(beta=rep(0,p))
model1 = jags.model(textConnection(model1_string),n.chains=2,
                    inits=init,data = dat,quiet=TRUE)
# Generate samples
update(model1, burn, progress.bar="none")
samp1    = coda.samples(model1,
          variable.names="beta",
          n.iter=iters, progress.bar="none")

# Compile results
ESS1     = effectiveSize(samp1)
out1     = summary(samp1)$quantiles
rownames(out1)=short_names

# Compute DIC
dic1     = dic.samples(model1,n.iter=iters,progress.bar="none")

# Compute WAIC
waic1    = coda.samples(model1,
          variable.names=c("like"),
          n.iter=iters, progress.bar="none")
like1    = waic1[[1]]
fbar1    = colMeans(like1)
P1       = sum(base::apply(log(like1),2,var))
WAIC1    = -2*sum(log(fbar1))+2*P1
```

## Model 2: Slopes as fixed effects

```
model2_string = "model{

   # Likelihood
   for(i in 1:n){
      Y[i]    ~ dnorm(mnY[i],taue)
      mnY[i] <- inprod(X[i,],beta[id[i],])
   }

   # Slopes
   for(j in 1:p){for(i in 1:N){
       beta[i,j] ~ dnorm(0,0.01)
   }}

   # Priors
   taue ~ dgamma(0.1,0.1)


   # WAIC calculations
   for(i in 1:n){
     like[i]    <- dnorm(Y[i],mnY[i],taue)
   }
  }"

# Load the model
dat    = list(Y=Y,n=n,N=N,X=X,p=p,id=id)
init   = list(beta=matrix(0,N,p))
model2 = jags.model(textConnection(model2_string),n.chains=2,
                     inits=init,data = dat,quiet=TRUE)

# Generate samples
update(model2, burn, progress.bar="none")
samp2   = coda.samples(model2,
          variable.names="beta",
          n.iter=iters, progress.bar="none")

# Compile results
ESS2     = effectiveSize(samp2)
sum      = summary(samp2)$stat
post_mn2 = matrix(sum[,1],N,p)
post_sd2 = matrix(sum[,2],N,p)

# Compute DIC
dic2   = dic.samples(model2,n.iter=iters,progress.bar="none")

# Compute WAIC
waic2    = coda.samples(model2,
           variable.names=c("like"),
           n.iter=iters, progress.bar="none")
like2    = waic2[[1]]
fbar2    = colMeans(like2)
P2       = sum(base::apply(log(like2),2,var))
```

```
WAIC2   = -2*sum(log(fbar2))+2*P2
```

## Model 3: Slopes as random effects

```
 model3_string = "model{

   # Likelihood
   for(i in 1:n){
      Y[i]    ~ dnorm(mnY[i],taue)
      mnY[i] <- inprod(X[i,],beta[id[i],])
   }

   # Random slopes
   for(j in 1:p){
     for(i in 1:N){
       beta[i,j] ~ dnorm(mu[j],taub[j])
     }
     mu[j]    ~ dnorm(0,0.01)
     taub[j]  ~ dgamma(0.1,0.1)
   }

   # Priors
   taue ~ dgamma(0.1,0.1)


   # WAIC calculations
   for(i in 1:n){
     like[i]    <- dnorm(Y[i],mnY[i],taue)
   }
  }"


# Load the model
dat    = list(Y=Y,n=n,N=N,X=X,p=p,id=id)
init   = list(beta=matrix(0,N,p))
model3 = jags.model(textConnection(model3_string),n.chains=2,
                     inits=init,data = dat,quiet=TRUE)

# Generate samples
update(model3, burn, progress.bar="none")
samp3  = coda.samples(model3,
          variable.names="beta",
          n.iter=iters, progress.bar="none")

# Compile results
ESS3     = effectiveSize(samp3)
sum      = summary(samp3)$stat
post_mn3 = matrix(sum[,1],N,p)
post_sd3 = matrix(sum[,2],N,p)

# Compute DIC
```

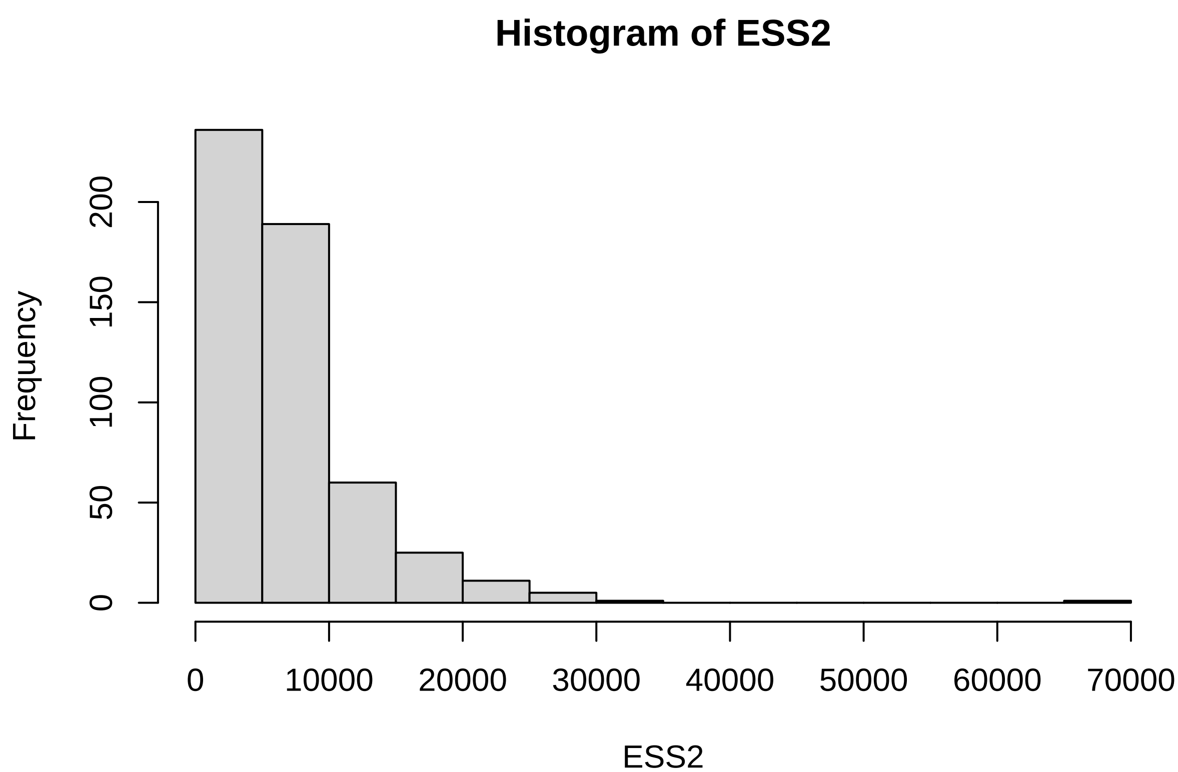
```r
dic3     = dic.samples(model3,n.iter=iters,progress.bar="none")

# Compute WAIC
waic3    = coda.samples(model3,
            variable.names=c("like"),
            n.iter=iters, progress.bar="none")
like3    = waic3[[1]]
fbar3    = colMeans(like3)
P3       = sum(base::apply(log(like3),2,var))
WAIC3    = -2*sum(log(fbar3))+2*P3
```

## Convergence Test

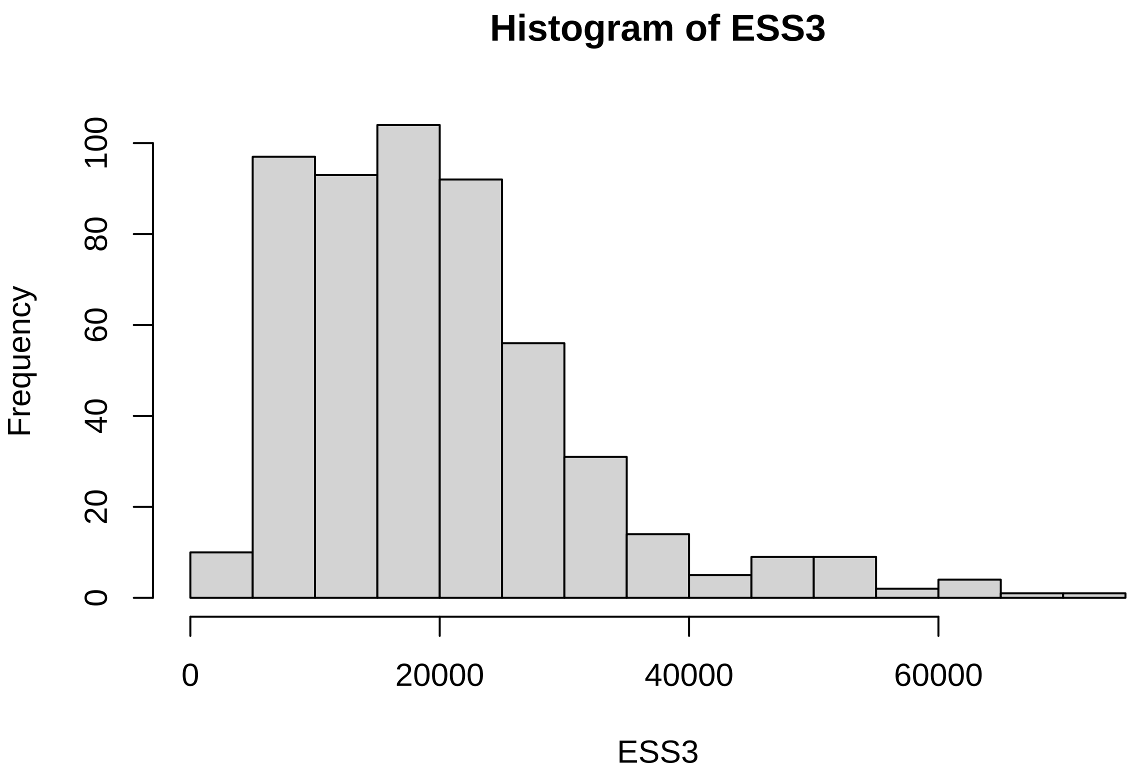
```r
ESS1
```

```
##     beta[1]     beta[2]     beta[3]     beta[4]     beta[5]     beta[6]     beta[7]
## 100000.000   56354.219   18388.417   41303.479   29727.008   16823.142   16571.993
##     beta[8]     beta[9]    beta[10]    beta[11]
##  23836.951   17087.014    8342.817   11594.937
```

```r
hist(ESS2)
```



```r
hist(ESS3)
```




Summary: The effective sample size is substantial for all parameters across all models, indicating that the MCMC algorithm appears to have converged successfully.

## Summarize the non-spatial model

```r
library(kableExtra)
 kbl(round(out1,2))
```

| | 2.5% | 25% | 50% | 75% | 97.5% |
|---|---|---|---|---|---|
| Intercept | 6.40 | 6.58 | 6.67 | 6.76 | 6.93 |
| Pop change | -1.46 | -1.25 | -1.14 | -1.03 | -0.81 |
| 65+ | 0.54 | 0.79 | 0.93 | 1.06 | 1.33 |
| African American | -1.89 | -1.67 | -1.56 | -1.44 | -1.23 |
| Hispanic | -2.40 | -2.18 | -2.06 | -1.95 | -1.72 |
| HS grad | 1.25 | 1.58 | 1.75 | 1.92 | 2.25 |
| Bachelor's | -6.72 | -6.37 | -6.19 | -6.01 | -5.67 |
| Homeownership rate | -0.38 | -0.13 | 0.01 | 0.15 | 0.41 |
| Home value | -1.98 | -1.68 | -1.52 | -1.36 | -1.05 |
| Median income | 1.14 | 1.62 | 1.87 | 2.13 | 2.62 |
| Poverty | 0.91 | 1.28 | 1.48 | 1.67 | 2.04 |

Summary: Except for the homeownership rate, all covariates have 95% confidence intervals that do not include zero. GOP support generally increased in counties with a declining population, a high proportion of seniors and high school graduates, a low proportion of African Americans and Hispanics, high income, low home value, and a high poverty rate.

# Compare models with DIC

```
dic1
```

```
## Mean deviance:  21300
## penalty 12.01
## Penalized deviance: 21312
```

```
dic2
```

```
## Mean deviance:  18483
## penalty 455.2
## Penalized deviance: 18939
```

```
dic3
```

```
## Mean deviance:  18604
## penalty 238.1
## Penalized deviance: 18842
```

Summary: The first model, which has constant slopes, is the simplest but fits the data poorly, resulting in the highest DIC. The second model, featuring different slopes for each state, offers the best fit (smallest mean deviance) but is overly complex and has a large $p_D$. The final model achieves a balance between fit and complexity, with a relatively small mean deviance and $p_D$, yielding the lowest DIC.

# Compare models with WAIC

```
WAIC1; P1
```

```
## [1] 21334.97
```

```
## [1] 20.08412
```

```
WAIC2; P2
```

```
## [1] 18972.99
```

```
## [1] 406.2257
```

```
WAIC3; P3
```

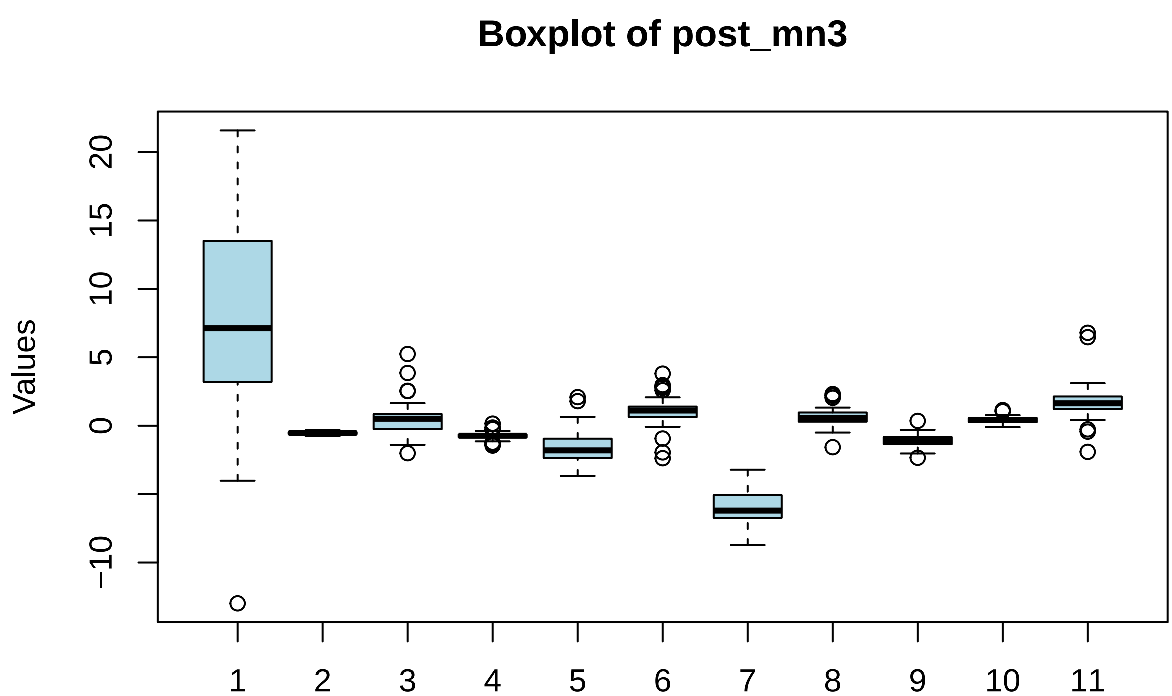
```
## [1] 18909.68
```

```
## [1] 258.7669
```

Summary: WAIC overlaps with DIC. Both prefer Model 3 with the regression coefficients treated as random effects.

## Explore the results of the final model

```
boxplot(post_mn3, main = "Boxplot of post_mn3", ylab = "Values", col = "lightblue")
```

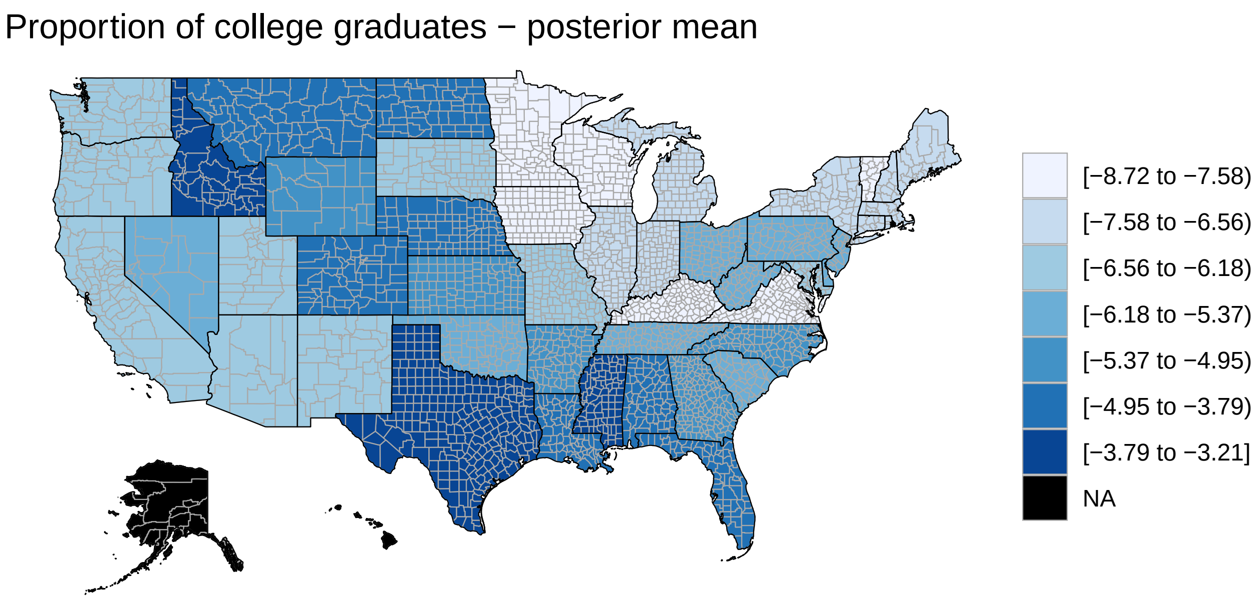



Summary: The effect of the proportion of college graduates varies the most across states

## Explore the three estimate of the effects of college graduates

```
# Posterior mean
 county_plot(fips,post_mn3[id,7],
            main="Proportion of college graduates - posterior mean")
```

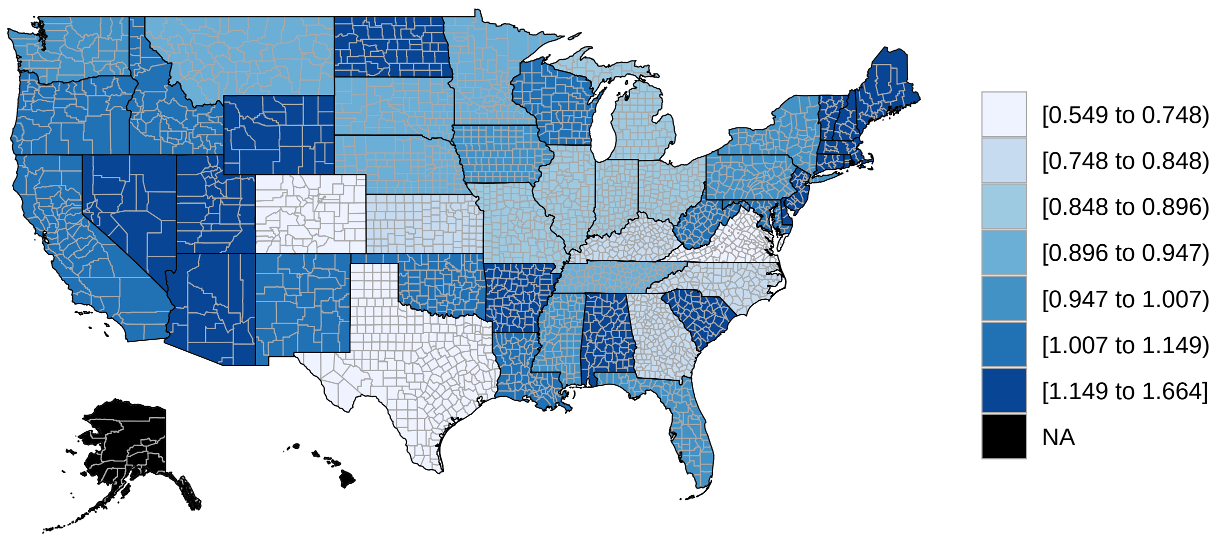
```
## Warning in self$bind(): The following regions were missing and are being set to
## NA: 2050, 2105, 2122, 2150, 2164, 2180, 2188, 2240, 2090, 2198, 15005, 2100,
## 2170, 2016, 2060, 2290, 2282, 15003, 2070, 2110, 2130, 2185, 2195, 2220, 2230,
## 2020, 2068, 2013, 2261, 2270, 11001, 2275, 15001, 15007, 15009
```




```r
# Posterior sd
county_plot(fips,post_sd3[id,7],
main="Proportion of college graduates - posterior SD")
```

```
## Warning in self$bind(): The following regions were missing and are being set to
## NA: 2050, 2105, 2122, 2150, 2164, 2180, 2188, 2240, 2090, 2198, 15005, 2100,
## 2170, 2016, 2060, 2290, 2282, 15003, 2070, 2110, 2130, 2185, 2195, 2220, 2230,
## 2020, 2068, 2013, 2261, 2270, 11001, 2275, 15001, 15007, 15009
```

Proportion of college graduates − posterior SD




Summary: The proportion of college graduates has the most significant and negative effect in the Midwest.