A Bayesian Approach to Analyze Drug Overdose Deaths Across the United States in 2018

Russell Land

Georgia Southern University

December 9, 2019

Abstract

The goal of this project is to show my understanding of the course STAT 7090: Introduction to Bayesian Statistics. I found a data set online regarding drug overdose deaths, a current epidemic across the United States. The data set includes over 25,000 observations, but I slim it down considerably for this project. After organizing the data I use prior knowledge found online and the data to obtain a posterior density for the data.

The second half of the project involves running a Bayesian multiple linear regression and finding statistics on the data to better understand how the response variable interacts with the covariates. The entirety of the project is done using R and OpenBUGS. The presentation is made using the LaTeX typesetting language and the BibTex bibliography software tool.

Source of Data

The data used was found on *www.data.CDC.gov*[1]. The data set includes the following variables:

- State
- Year
- Period
- Indicator
- Data.Value
- Percent.Complete
- Percent.Pending.Investigation
- State.Name
- Footnote
- Footnote.Symbol
- Predicted.Value

Source of Data (cont.)

I am interested in the variables: **Data.Value**, **State**, **Year**, **Month**, and **Indicator**. The **indicator** I am interested in is "Number of Drug Overdose Deaths."

The following code will look at observations with **Year** of '2018' and an **indicator** of 'Number of Drug Overdose Deaths.' It will then add the number of deaths per month per state.

```
data1 <- read.csv("Drug_Overdose_Deaths.csv",header =
    TRUE,sep = ",")
a <- data1[data1$Year=="2018" & data1$Indicator=="Number
    of Drug Overdose Deaths",]
b <- as.matrix(a)
c <- b[,-c(2:5,7:12)]
d <- transform(c, Data.Value = as.numeric(Data.Value))
e <- aggregate(Data.Value~State,d,sum)
f <- e[-c(45,53),]              ## Finally done sorting the
    large data set
```

Source of Data (cont.)

It takes this...

State	Month	Data.Value
AK	January	46
AK	February	30
AK	March	26
⋮	⋮	⋮
AK	December	25

and turns it into...

State	Data.Value
AK	412

The next step is to size this data to something easier to work with and something applicable. Next, I add the population of each state[2], the region of every state[3], the annual income per capita per state[4], and the health care expenditures per capita by state residence[5].

So why did I choose these variables to add to the data?

Population of Each State

I start with the number of deaths for each state (n_i) and the population of each state (N_i). Dividing the two will give the ratio (r_i) of deaths per state:

$$r_i = \frac{n_i}{N_i}.$$

After this I multiply the ratio by 100 to give the percentage, which makes the data easier to read. With this following transformation, the data now looks like...

Number	State	Number of Deaths (n_i)	Population (N_i)	Ratio (%)
1	AK	412	737,438	0.055869104
2	AL	2632	4,887,871	0.053847575
3	AR	1718	3,013,825	0.057003973
⋮	⋮	⋮	⋮	⋮
51	WY	103	577,737	0.017828181

Regions

I classify each state into a region. Just a brief glance at the data I notice a higher trend of deaths in the eastern United States; they may have a linear relationship. Here is how I classified each state. Note that I classify Alaska in region 1 and Hawaii in region 2.

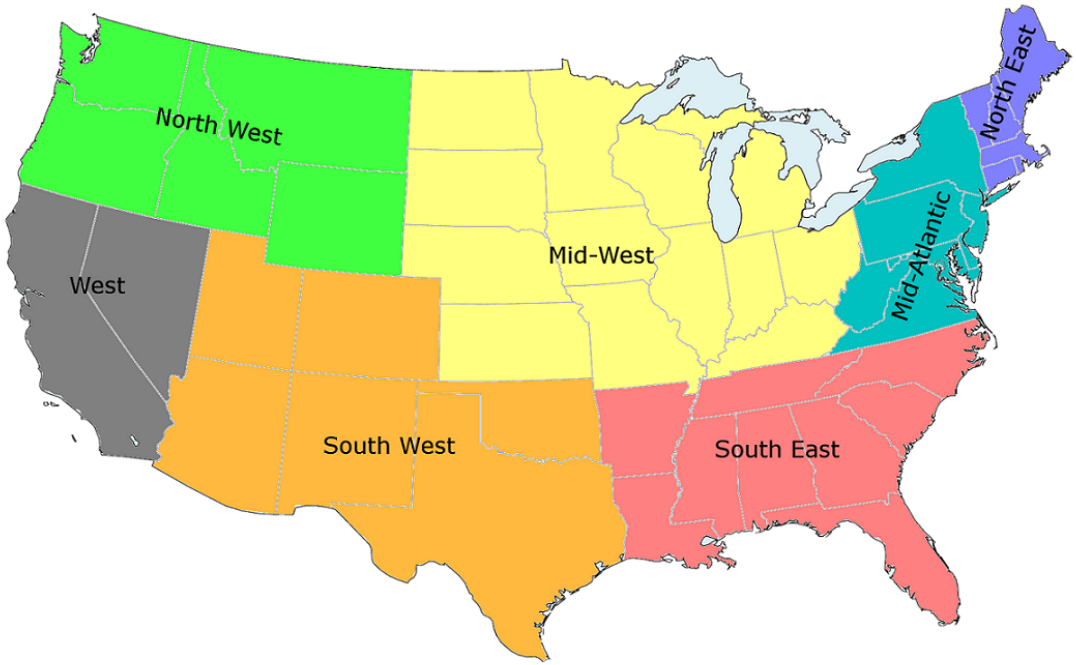


Figure 1: Regions of the USA[3]

Income

I include the variable 'household median incomes per capita per state'[4]. My reasoning behind this, is the idea that the level of an individuals income could be related to their potential to abuse illegal or prescription drugs. Here are a few of the observations:

State	Income ($)
AK	73,181
AL	48,123
AR	45,869
⋮	⋮
WY	60,434

Health Care

The last variable I add is health care expenditures per capita of each state. The more a state spends on health care should be related to how many prescription drugs are purchased, right? We will test this.

With all this being said, here is a summary of the data I will be working with.

Number	State	Deaths	Percent.Pop	Region	Income ($)	Health.Care ($)
1	AK	412	0.055869104	1	73,181	11,064
2	AL	2,632	0.053847575	5	48,123	7,281
3	AR	1,718	0.057003973	5	45,869	7,408
⋮	⋮	⋮	⋮	⋮	⋮	⋮
51	WY	103	0.017828181	1	60,434	8,320

Frequentist Approach

I know this is a Bayesian project, but I will run a frequentist analysis to get a feel for the data. Then we can compare the frequentist approach and Bayesian approach at the end. I ran the following code and got the output below.

```
linreg <- lm(data2[,3]~data2[,4]+data2[,5]+data2[,6])
summary(linreg)
```

```
Call:
lm(formula = data2[,3]   data2[,4] + data2[,5] + data2[,6])

Coefficients:
                Estimate  Std.  Error  t value  Pr(>|t|)
(Intercept)  -1.931e-02    3.178e-02   -0.608    0.5463
data2[,4]     6.682-e03    2.633e-03    2.538    0.0145
data2[,5]     -2074e-07    4.545e-07   -0.456    0.6503
data2[,6]     7.711e-06    4.141e-06    1.862    0.0688

Multiple R-squared:  0.2841,   Adjusted R-squared:  0.2384
```

Frequentist Approach (cont.)

We will now only include the first covariate (region), because it is the only significant one.

```
linreg1 <- lm(data2[,3]~data2[,4])
summary(linreg1)
```

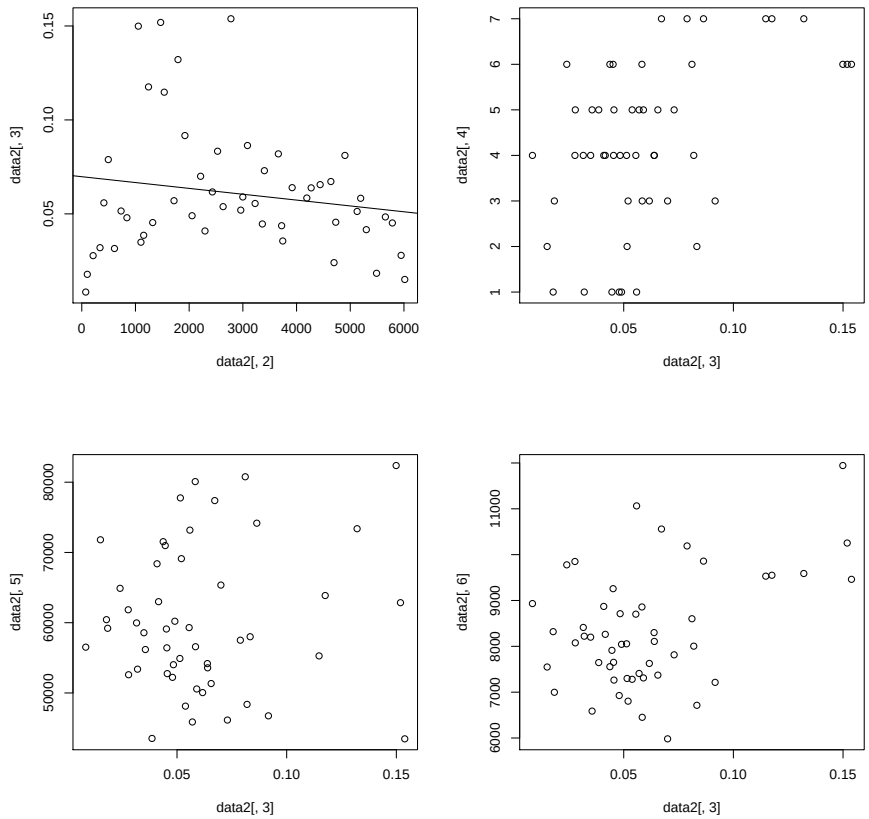
```
Call:
lm(formula = data2[,3]   data2[,4])

Coefficients:
             Estimate  Std.  Error  t value  Pr(>|t|)
(Intercept)  0.022503     0.010907    2.063  0.044416
data2[,4]    0.009023     0.002362    3.819  0.000377

Multiple R-squared:  0.2294,  Adjusted R-squared:  0.2137
```

Frequentist Approach (cont.)

Here are scatterplots for (deaths vs. ratios), (ratios vs. region), (ratio vs. income), and (ratio vs. health care).



Posterior Density

Before we begin a posterior density calculation we need to make some assumptions about the data.

- We will assume the data is distributed normally with mean θ and known standard deviation σ.
- We are going to assume prior knowledge about this data. In an article about the predictions of 2018 data, there was an estimate that every 58 out of 100,000 people in the United States will die from a drug overdose in 2018[6]. We will choose a normal prior with mean 0.058 and standard deviation of 0.5. These were picked by analyzing data from 2017.
- Unrelated to our study, about 7 out of 10 of these overdose deaths come from opioids such as fentanyl, heroin, and prescription opioids[6].

Posterior Density Derivation

This distribution is one that we derived in class. I will include the derivation below. We know that a posterior density can be obtain with the following:

$$\pi(\theta|y) = \frac{f(y|\theta) \cdot \pi(\theta)}{m(y)}$$

where

$$m(y) = \int_{\theta} f(y|\theta) \cdot \pi(\theta) d\theta$$

Let's also mention that both the likelihood function and prior belong to the exponential family, and thus are conjugate distributions.

Posterior Density Derivation (cont.)

The likelihood function is

$$f(y|\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y-\theta)^2\right\}$$

and the prior in terms of θ is

$$\pi(\theta) = \frac{1}{\tau_0\sqrt{2\pi}} \exp\left\{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2\right\}$$

where μ_0 are τ_0 are hyper-parameters.

Posterior Density Derivation (cont.)

$$\pi(\theta|y) = \frac{\overbrace{f(y|\theta)\cdot\pi(\theta)}^{n}}{\int_\theta f(y|\theta)\cdot\pi(\theta)d\theta}$$

$$n = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{1}{2\sigma^2}(y-\theta)^2\right\}\cdot\frac{1}{\tau_0\sqrt{2\pi}}\exp\left\{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2\right\}$$

$$= \frac{1}{2\pi\sigma\tau_0}\exp\left\{-\frac{1}{2\sigma^2}(y-\theta)^2 - \frac{1}{2\tau_0^2}(\theta-\mu_0)^2\right\}$$

$$= \frac{1}{2\pi\sigma\tau_0}\exp\left\{-\frac{1}{2\sigma^2}\left(y^2-2y\theta+\theta^2\right) - \frac{1}{2\tau_0^2}\left(\theta^2-2\theta\mu_0+\mu_0^2\right)\right\}$$

$$= \underbrace{\frac{1}{2\pi\sigma\tau_0}}_{C}\exp\left\{\frac{2\theta y\tau_0^2+2\theta\mu_0\sigma^2-\theta^2\left(\tau_0^2+\sigma^2\right)}{2\sigma^2\tau_0^2} - \underbrace{\frac{y^2\tau_0^2+\mu_0^2\sigma^2}{2\sigma^2\tau_0^2}}_{C_1}\right\}$$

Posterior Density Derivation (cont.)

$$= C \cdot \exp\left\{ \frac{-\left(\tau_0^2 + \sigma^2\right)\left(\theta^2 - 2\theta \overbrace{\left(y\tau_0^2 + \mu_0\sigma^2\right)\left(\tau_0^2 + \sigma^2\right)^{-1}}^{B} + B^2 - B^2\right)}{2\sigma^2\tau_0^2} - C_1 \right\}$$

$$= C \cdot \exp\left\{ \frac{\left(\tau_0^2 + \sigma^2\right)B^2}{2\sigma^2\tau_0^2} - \frac{\left(\tau_0 + \sigma^2\right)(\theta - B)^2}{2\sigma^2\tau_0^2} - C_1 \right\}$$

$$= C_2 \cdot \exp\left\{ -\frac{\left(\tau_0 + \sigma^2\right)(\theta - B)^2}{2\sigma^2\tau_0^2} \right\} \tag{1}$$

where

$$C_2 = \frac{1}{2\pi\sigma\tau_0} \exp\left\{ \frac{\left(\tau_0^2 + \sigma^2\right)B^2 - y^2\tau_0^2 - \mu_0^2\sigma^2}{2\sigma^2\tau_0^2} \right\}$$

Posterior Density Derivation (cont.)

Re-expressing equation (1) slightly will give:

$$\pi(\theta|y) = C_2 \cdot \exp\left\{ -\frac{\left(\frac{1}{\tau_0^2} + \frac{1}{\sigma^2}\right)(\theta - B)^2}{2} \right\}$$

which looks like the kernel of a normal distribution will mean B and precision $\left(\frac{1}{\tau_0^2} + \frac{1}{\sigma^2}\right)$, leaving us with the constant C_2. Now, let's consider the denominator $m(y)$:

$$\begin{aligned} m(y) &= \int_\theta f(y|\theta) \cdot \pi(\theta) d\theta \\ &= \int_\theta C_2 \cdot \exp\left\{ -\frac{\left(\frac{1}{\tau_0^2} + \frac{1}{\sigma^2}\right)(\theta - B)^2}{2} \right\} d\theta \qquad (2) \\ &= C_2 \cdot \sqrt{2\pi \left(\frac{1}{\tau_0^2} + \frac{1}{\sigma^2}\right)^{-1}} \end{aligned}$$

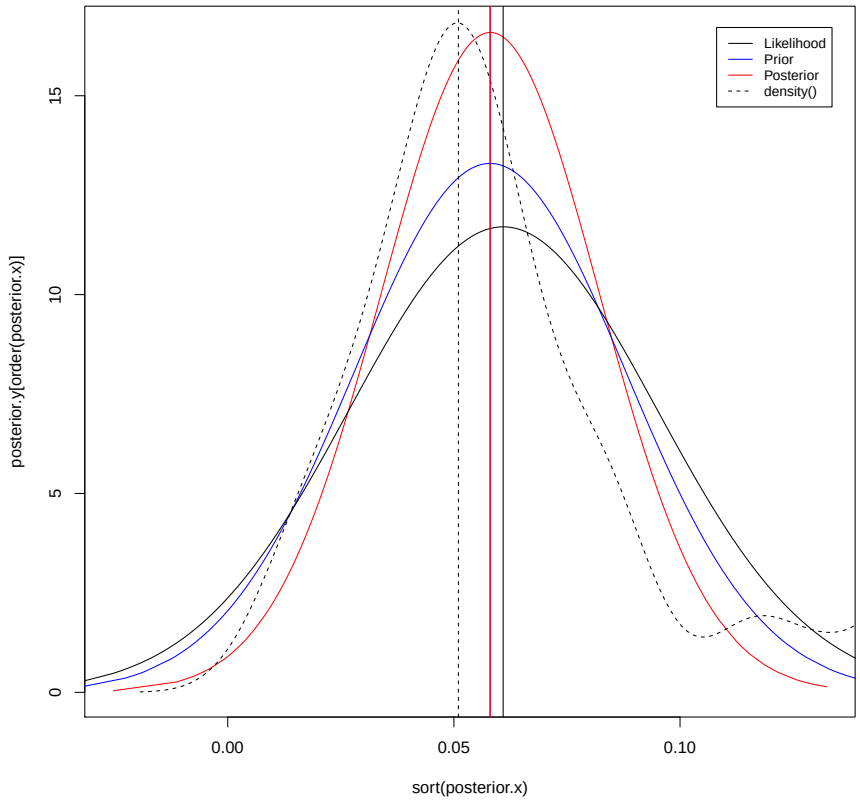
Posterior Density Derivation (cont.)

we arrive at equation (2) using the same exact steps as before. From here we have obtained the posterior distribution. We can conclude

$$\pi(\theta|y) \sim N\left(\frac{\frac{y}{\sigma^2}+\frac{\mu_0}{\tau_0^2}}{\frac{1}{\sigma^2}+\frac{1}{\tau_0^2}}\ ,\ \left(\frac{1}{\sigma^2}+\frac{1}{\tau_0^2}\right)^{-1}\right)$$

Visualization

With the posterior we just found, we can now make plots using the data. Below is a plot containing the likelihood, prior, and posterior. I also include a density estimation using a built-in R function. A vertical line is placed at the max of every curve to help see the differences.



MCMC Sampling

We were able to find the posterior density, and OpenBUGS can handle our model, but I will run an algorithm in R anyway. I will use the Metropolis-Hastings algorithm. This algorithm can be broken down into the following steps:

1. Pick an arbitrary density, say $q(\theta|\phi)$, to suggest transition probabilities.
2. Next, we will initialize a θ value, say θ_0.
3. At time i, we will generate θ^* from the density $q\left(\theta^*|\theta_{i-1}\right)$.
4. Next, the acceptance probability can be calculated by

$$\alpha = \min\left(1, \frac{P\left(\theta^*\right) \cdot q\left(\theta_{i-1}|\theta^*\right)}{P\left(\theta_{i-1}\right) \cdot q\left(\theta^*|\theta_{i-1}\right)}\right) \qquad \text{where } P(\cdot) = \pi(\theta|y)$$

5. Generate U from a uniform distribution $(0, 1)$.
6. If $U \leq \alpha$, then let $\theta_{i-1} = \theta^*$. Otherwise let $\theta_{i+1} = \theta_i$.
7. Repeat this process for N iterations.

Metropolis-Hastings Algorithm


```r
## Establish initials
mu <- 0.05
sigma <- 1
delta <- 1
prior.mean <- 0.058
prior.sd <- 0.03
theta.initial <- rnorm(1,mu,1)
N <- 20000
theta.post <- rep(NA, N)
theta.post[1] <- theta.initial
```

Metropolis-Hastings Algorithm (cont.)


```
## Metropolis-Hastings Algorithm
for (i in 2:N) {
  theta = rnorm(1,theta.post[i-1],delta)
  num = prod(dnorm(data,theta,sigma))*dnorm(theta,prior.
      mean,prior.sd)*dnorm(theta.post[i-1],theta,delta)
  denom = prod(dnorm(data,theta.post[i-1],sigma))*dnorm(
      theta.post[i-1],prior.mean,prior.sd)*dnorm(theta,
      theta.post[i-1],delta)
  W = num/denom
  alpha = min(1,W)
  U = runif(1,0,1)

  if (U <= alpha) {
    theta.post[i] = theta
    print("advance")
  }
  else {
    theta.post[i] <- theta.post[i-1]
    print("maintain")
  }
  print(theta.post[i])
}
```

Metropolis-Hastings Algorithm (cont.)

Burn-in step not included.

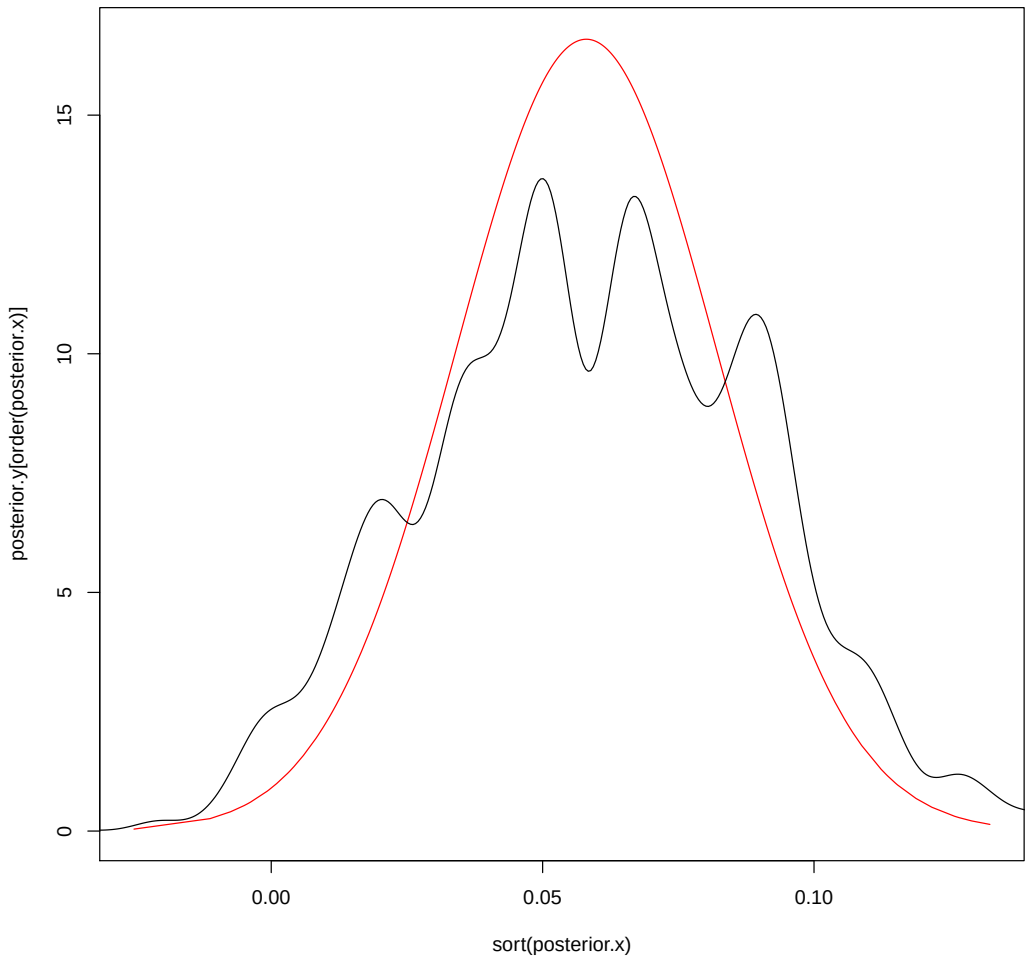
```
## Posterior Density vs Algorithm Density
pdf(file = "metropolis.pdf",width = 9,height=9)
plot(sort(posterior.x),posterior.y[order(posterior.x)],
     type = "l",col="red")
lines(density(post.sample))
dev.off()

## Posterior Summary
mean(post.sample)
sd(post.sample)

## History Plot
pdf(file = "history.pdf",width = 9,height=9)
par(mfrow=c(2,1))
plot(c(1:N), theta.post, type="l")                    # before burn-in
plot(c(1:length(post.sample)), post.sample, type="l")  # after burn-in
dev.off()
```

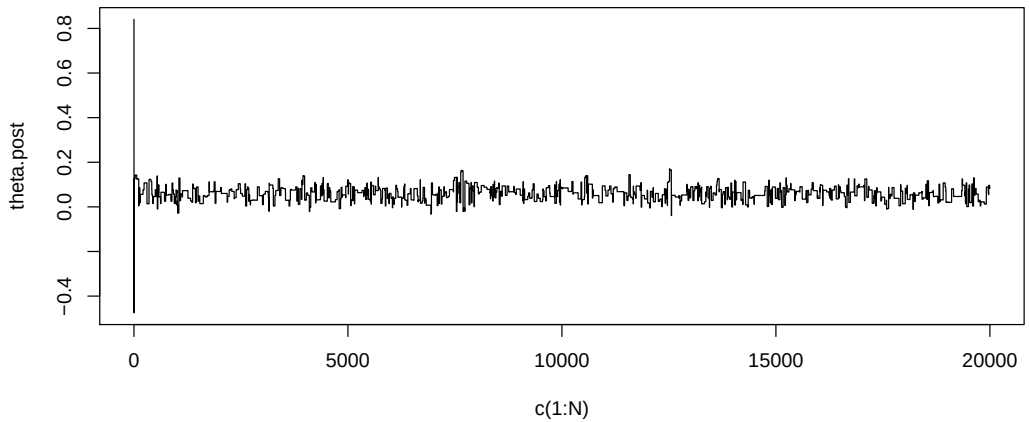
Metropolis-Hastings Algorithm Output

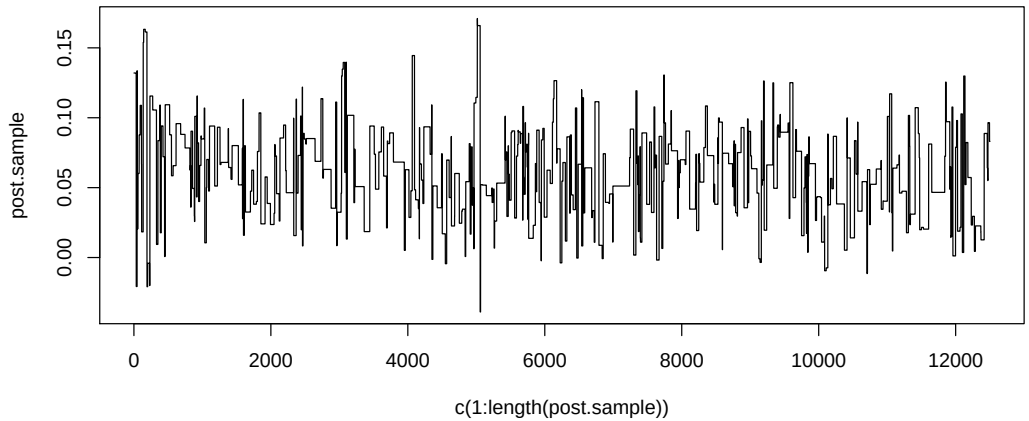
Density Plot



Metropolis-Hastings Algorithm Output

History Plots





OpenBUGS Posterior Density Estimation

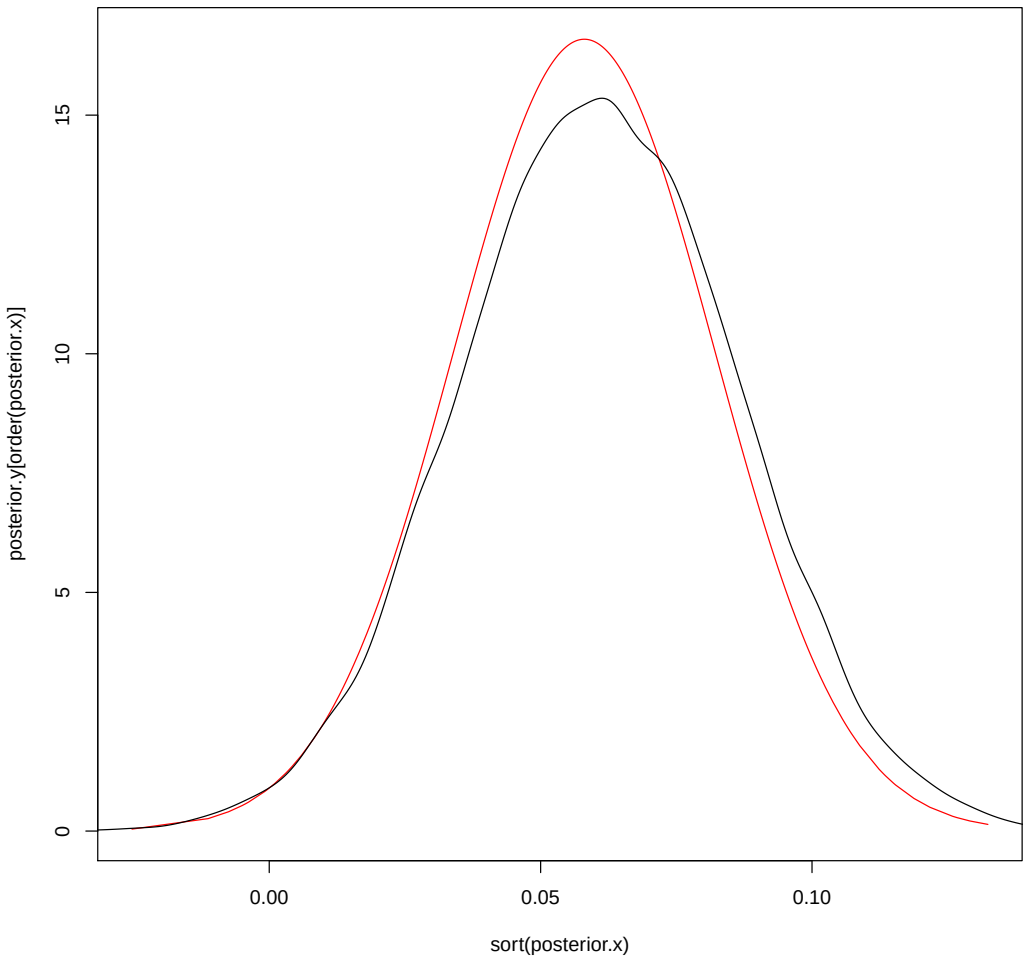
We can use OpenBUGS to estimate a posterior density. Here is the model I constructed:

```
model1 <- function(){
  for (i in 1:51){
    y[i] ~ dnorm(mu,29.33735)
  }

  mu ~ dnorm(0.058,33.3333333)
  y.pred ~ dnorm(mu,29.33735)
  prob.y <- step(y.pred-0.1)
}
```

OpenBUGS Posterior Density Estimation Output

Posterior Density vs. OpenBUGS Estimation



Some Statistics on OpenBUGS Density Estimation

With the density that OpenBUGS created, we can see it is a pretty close match to our calculated posterior. Let's now show the 95% credible interval and the highest posterior density (HPD) and interpret them. The 95% credible interval can be constructed with the following code:

```
## 95% Credible Interval for mu
credible.interval <- function(sample,alpha){
  c(sort(sample)[length(sample)*(alpha/2)],sort(sample)[
      length(sample)*(1-(alpha/2))])
}

credible.interval(samples1.mu,0.05)
```

So, 95% of a future annual drug overdose ratio will be between the interval (0.01081,0.11110). This is a fairly large interval.

Some Statistics on OpenBUGS Density Estimation

The HPD interval can be constructed with a built-in function, but here is code that I made:

```
## Function for the estimated CDF(a,b)
integrate.sample <- function(sample,a,b){
  if (missing(b)){
    sum(sample < a)/length(sample)
  }
  else {
    if (missing(a)) {
      sum(sample > b)/length(sample)
    }
    else {
      sum(sample < b)/length(sample) - sum(sample < a)/
          length(sample)
    }
  }
}
# If a is left blank, it will evalute upper tail area.
# If b is left blank, it will evaulate lower tail area.
integrate.sample(samples1.mu,0.00,0.05)
```

Some Statistics on OpenBUGS Density Estimation


```
# Returns the approximate HPD interval for any given
    sample and alpha, for unimodal posterior samples.
interval.HPD <- function(sample,alpha){

  pdf <- density(sample,n=1000)
  a <- 0
  q <- (which(pdf$y==max(pdf$y))-5)
  Q <- length(pdf$y)

  for (i in q:1){
    for (j in (i+1):Q){
      if (pdf$y[j] <= pdf$y[i]){
        a <- integrate.sample(sample,pdf$x[i],pdf$x[j])
      }
      if (a > (1-alpha)){
        break
      }
      if (pdf$y[j] <= pdf$y[i]){
        break
      }
    }
```

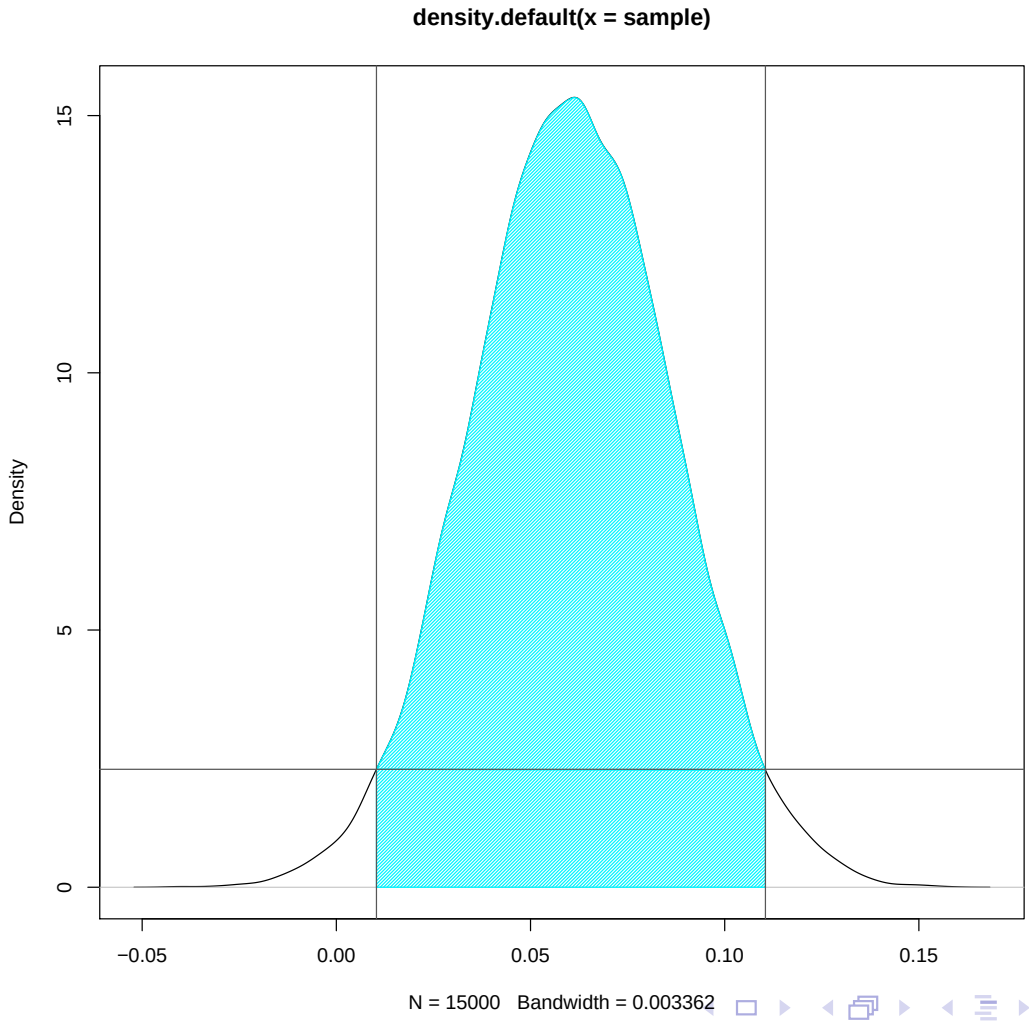
Some Statistics on OpenBUGS Density Estimation


```
      if (a > (1-alpha)){
        print("The HDP Interval is:")
        print(pdf$x[i])
        print(pdf$x[j])
        pdf(file = "hdi.pdf",width = 9,height=9)
        plot <- plot(density(sample))
        x <- pdf$x[i:j]
        y <- pdf$y[i:j]
        polygon(x,y,density=50,col='turquoise1')
        z <- c(pdf$x[i],pdf$x[i],pdf$x[j],pdf$x[j])
        w <- c(0,pdf$y[i],pdf$y[j],0)
        polygon(z,w,density=50,col='turquoise1')
        abline(h=pdf$y[i],col="gray34")
        abline(v=pdf$x[i],col="gray34")
        abline(v=pdf$x[j],col="gray34")
        dev.off()
        break
      }
    }
}

interval.HPD(samples1.mu,0.05)
```

Some Statistics on OpenBUGS Density Estimation

The highest density interval is the narrowest interval that includes the highest probability possible at 95%. Below is a visual of this.



Bayesian Linear Regression

Next in our analysis I will run a Bayesian multiple linear regression on the response variable 'number of deaths' with covariates 'region', 'income per capita per state', and 'health care expenditures per capita per state'.

The goal of this multiple linear regression is to answer the following questions:

1. How does the response variable vary as as function of the covariates?
2. Which X_j's have an effect?
3. Can we predict Y as a function of X?

These are questions I will answer. Because we are taking a Bayesian approach we will test three different priors, two shrinking priors and one flat prior. The shrinking priors will tell us the significant covariates. I will start with Zellner's g-prior.

Zellner's G-Prior

We first need to initialize our response variable and covariates. This prior involves a variance-covariance matrix, so we need to calculate that as well. Here is the code for that:

```
## Now let's run a linear regression on the data.
## Let's start by using a shrinking prior, to find
    important factors.
## Zellner's g-prior
Y <- data2[,3]
X <- matrix(c(rep(1,length=length(Y)),data2[,4],data2
    [,5],data2[,6]),ncol = 4)
colnames(X) <- c("Intercept","X1","X2","X3")
taumatrix <- t(X) * X
mean <- c(0,0,0,0)
m <- length(Y)
```

The next slide will contain the model.

Zellner's G-Prior (cont.)


```
model2 <- function(){
  for (i in 1:m){
    Y[i] ~ dnorm(mu[i],tau)
    mu[i] <- inprod(beta[],X[i,])
  }

  g <- 2 * m
  beta[1:4] ~ dmnorm(mean[],prec[,])

  for(i in 1:4){
    for(j in 1:4){
      prec[i,j] <- (1/g)*tau*taumatrix[i,j]
    }
  }

  for (i in 1:m) {
    y.pred[i] ~ dnorm(mu[i],tau)
  }
```

Zellner's G-Prior (cont.)


```
1   for (i in 1:m) {
2     log_cpo[i] <- -0.5*log(tau/6.283)-0.5*pow((Y[i]-mu[i
          ]),2)
3   }
4   LPML <- sum(log_cpo[1:m])
5
6   for (i in 1:m) {
7     y[i] ~ dnorm(mu[i],tau)
8   }
9
10  tau ~ dgamma(0.01,0.01)
11 }
```

Here is the regression equation for this model:

$$\hat{Y} = -0.019 + 0.006X_1 - 0.000X_2 + 0.000X_3$$

The last two coefficients are very small. This is a trend that continues for all of the models I test.

Other Prior Choices

I ran the following priors using all three covariates and then again with the same priors, but with only the first covariate, 'regions'.

- Zellner's G-Prior
- Double Exponential Prior
- Flat Prior

I am not going include all of the models, but I will list the regression equation for each.

Regression Equations

- Zellner's G-Prior (three covariates)

$$\hat{Y} = -0.019 + 0.006X_1 - 0.000X_2 + 0.000X_3$$

- Double Exponential Prior (three covariates)

$$\hat{Y} = 298.126 + 49.401X_1 + 0.016X_2 - 0.179X_3$$

- Flat Prior (three covariates)

$$\hat{Y} = -0.012 + 0.007X_1 - 0.000X_2 + 0.000X_3$$

- Zellner's G-Prior (one covariate)

$$\hat{Y} = 0.022 + 0.008X_1$$

- Double Exponential (one covariate)

$$\hat{Y} = 0.228 + 0.008X_1$$

- Flat Prior (one covariate)

$$\hat{Y} = 0.023 + 0.008X_1$$

Model Comparison

After running multiple models, how can we decide which one is superior?

We have a few ways for model comparison. We can use LPML, or log-psuedo-marginal-likelihood. This is a type of 'leave-one-out' cross-validation. This process esentially works by running the model and leaving out an observation each time. After running the model n times, calculate the conditional predictive ordinate, usually abbreviated by cpo. We denote it as

$$\text{cpo}_j = P\left(y_1|y_{(-1)}\right) = \int_\theta f\left(y_1|\theta_1 y_{(-1)}\right)\pi\left(\theta|y_{(-1)}\right)d\theta$$

Then,

$$\text{LPML} = \sum_{i=1}^{n} \ln(\text{cpo}_i)$$

Larger LPML will indicate a better model, because they are dependent on posterior density probabilities.

Model Comparison (cont.)

Another form of model comparison is called DIC, or deviance information criteria, and is defined as

$$DIC = -2 \log p\left(y|\hat{\theta}_{\text{bayes}}\right) + 2p_{\text{DIC}}$$

where

$$p_{\text{DIC}} = \left(\log p\left(y|\hat{\theta}_{\text{bayes}}\right) - E_{\text{post}}\left(\log p(y|\theta)\right)\right).$$

Whereas the last model selection method indicates better models with higher values, the DIC indicate superior models with smaller values. This is because the DIC is taking the difference of "expectations". The smaller a DIC value is, the better it is at predicting responses.

There is another model selection method called Bayes Factor that directly compares two or more models, which we will discuss later.

Model Comparison (cont.)

By implementing the LPML and DIC methods, we can run a model selection on the six models we have chosen. Luckily, OpenBUGS has the power to calculate these for us! Below are the results from all six models:

	Model	LPML	DIC
Three Covariates	Zellner's G-Prior	-123.1792	-203.9
	Double Exponential Prior	-1258577	1140
	Flat Prior	-121.8963	-202.9
One Covariate	Zellner's G-Prior	-121.8831	-204.6
	Double Exponential Prior	5.136655	14.37
	Flat Prior	-121.5848	-204.6

From the above results, we can see that the last row has the largest LPML value (excluding the double exponential) and the smallest DIC value. This indicates it is a superior model. The double exponential prior gives me strange results, so I will not compare them with the rest and I will conclude they are not good models for this particular data.

Bayes Factor

Bayes Factor is a model comparison method. We can consider two models. For instance, the flat prior with three covariates (M_1) and the flat prior with one covariate (M_2). We define Bayes Factor between the first and second model as

$$\mathrm{BF}_{12} = \frac{\pi\left(M_1|x\right)/P\left(M_1\right)}{\pi\left(M_2|x\right)/P\left(M_2\right)}$$

and if we assume $P(M_1) = P(M_2) = \frac{1}{2}$, we can reduce this to

$$\mathrm{BF}_{12} = \frac{\pi\left(M_1|x\right)}{\pi\left(M_2|x\right)}$$

Bayes Factor (cont.)

Constructing the following model will allow for a Bayes Factor calculation:

```
model8 <- function(){
  M ~ dcat(p[1:2])

  for (i in 1:m){
    Y[i] ~ dnorm(mu[M,i],tau[M])
    mu[1,i] <- inprod(alpha[],X[i,])
    mu[2,i] <- inprod(beta[],X1[i,])
  }

  p[1] <- 0.5
  p[2] <- 0.5

  for (i in 1:4) {
    alpha[i] ~ dnorm(0,1e-6)
  }

  for (i in 1:2) {
    beta[i] ~ dnorm(0,1e-6)
  }
```

Bayes Factor (cont.)


```
  tau[1] ~ dgamma(0.01,0.01)
  tau[2] ~ dgamma(0.01,0.01)
}
data.sales8 <- list("X","X1","m","Y")
inits.sales8 <- function(){list(tau=0.01,alpha=c
    (0.1,0.1,0.1,0.1),beta=c(0.1,0.1))}
params.sales8 <- c("mu","M")
out.sales8 <- bugs(data=data.sales8, inits=inits.sales8,
    parameters.to.save=params.sales8,
                    model.file=model.sales8,n.iter=20000,
                        n.burnin=5000, debug=F,n.chains
                       =1)
samples8.M <- out.sales8$sims.list$M
BF12 = sum(samples8.M==1)/sum(samples8.M==2)
BF12
```

After running this, we get a BF_{12} of 0. This means that for every iteration that OpenBUGS ran, it favored Model 2 almost every time. This indicates that Model 2 is a superior model.

Making Predictions

From all the tests and results we have just seen, let's conclude that the model with the flat prior and one covariate is the superior model out of the six we started with. This tell us that the region of the Unites States that you live in has impact on the the number of people that will die of a drug overdose.

Now, let's predict what percent of the population will die next year of a drug overdose in all seven regions of the United States defined early. We can use the regression equation for this prediction or we can use OpenBUGS. For simplicity, we will use the regression equation. The results are below:

Region	1	2	3	4	5	6	7
Prediction (%)	0.032	0.041	0.050	0.059	0.068	0.076	0.085

Conclusion

I believe that the data was a good fit for our model. The posterior density we constructed for the percent of drug overdoses for each state matched closely to the OpenBUGS results. We were able to show graphically how each density compared to each other; including the likelihood, prior, posterior, kernel estimation. We also were able to show graphically how the credible intervals and the HPD intervals work.

Next, we fit the data with a linear regression model. I do not believe this data has strong linear correlation with any of its covariates. We were able to construct a model and choose a superior model, but in practice, I would use another model, or even another Bayesian analysis technique other than regression.

Regarding a comparison with the frequentist approach, both concluded that the variable 'region' has stronger correlation than the others. They also both conclude that even though the first covariate had the best fit, it still was not a great fit for the data. The frequentist approach had a multiple R-squared value of 0.2294.

Conclusion (cont.)

In conclusion, I have learned a lot while taking Introduction to Bayesian Statistics. Not only have I learned a new approach to statistics, but I have also learned that there is not only one way to interpret statistics. Another thing I am glad we were exposed to was R and OpenBUGS. Although it was frustrating, it helped me learn to be patient and also taught me a lot of coding that I can apply to my future studies and other classes.

Thank You

References I


F. Ahmad, L. Rossen, M. Spencer, M. Warner, and P. Sutton, *VSRR Provisional Drug Overdose Death Counts*.
Department of Health and Human Services, 2019.


United States Census Bureau, *Annual Estimates of the Resident Population for the United States, Regions, States, and Puerto Rico: April 1, 2010 to July 1, 2018*, 2018.
XLSX file.


"Regions of the usa." website, December 2018.
image.


"Median household income (in 2018 inflation-adjusted dollars) - united states – states; and puerto rico." website, December 2018.
table.

References II


Kaiser Family Foundation, *Health Care Expenditures per Capita by State of Residence*, July 2017.
sourced from U.S. Bureau of the Census.


"Monthly estimates on drug overdose deaths in the united states." transcript, April 2019.