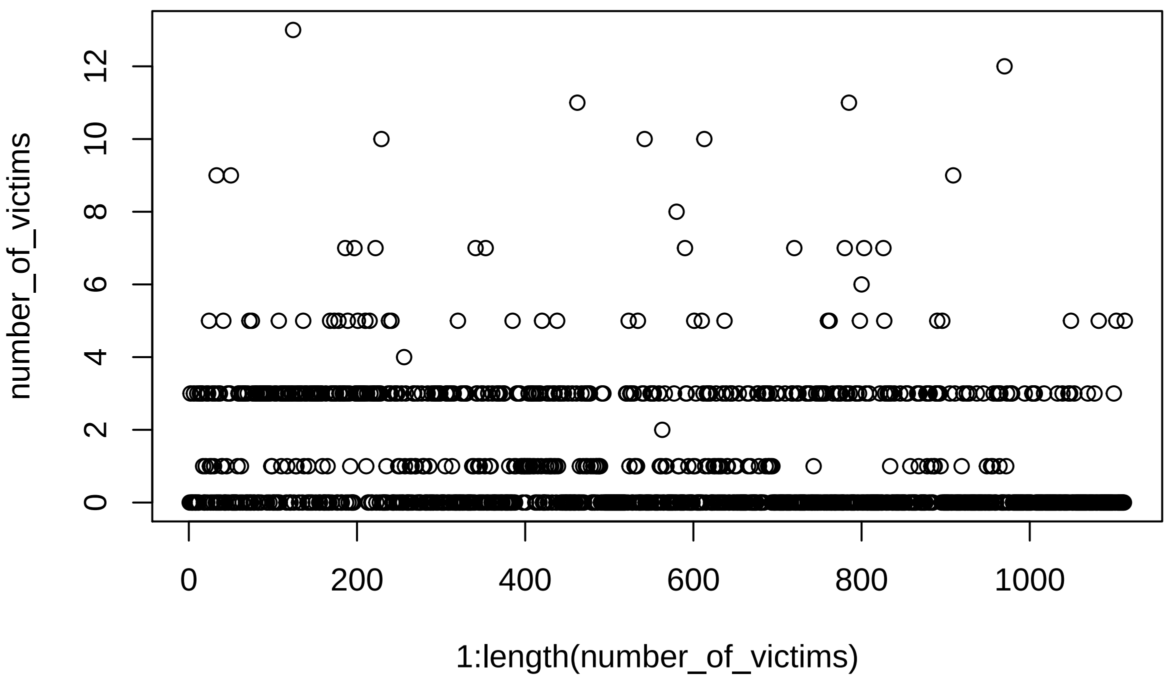
Bayes Project (Casey's Part)

Bianca, Heather, Casey

3/5/2018

Data



We first attempt a simple poisson regression.

$$Y_t = \text{number of victims of the t'th school shooting}$$

$$\begin{aligned} X_{1:4} = & \text{ gun type encoded as} \\ & 0 = \text{handgun} \\ & 1 = \text{rifle} \\ & 2 = \text{shotgun} \\ & 3 = \text{unknown/other} \end{aligned}$$

$$Y_t \sim Poisson(\lambda)$$
$$log(\lambda) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta 3 x_3$$

Poisson

```
library(rjags)
library(R2jags)
model <- "model {
    ## Likelihood
```

```
for(i in 1:N){
      y[i] ~ dpois(lambda[i])
      log(lambda[i]) <- mu[i]
      mu[i] <- beta4 + beta1*x1[i] +beta2*x2[i] + beta3*x3[i]
      }
    ## Priors

    beta1 ~ dnorm(mu.beta,tau.beta)
    beta2 ~ dnorm(mu.beta,tau.beta)
    beta3 ~ dnorm(mu.beta,tau.beta)
    beta4 ~ dnorm(mu.beta,tau.beta)
}"

dat <- data.frame(x=type_of_gun,y=number_of_victims)

forJags <- list(x1=dat$x.1,
                x2=dat$x.2,
                x3=dat$x.3,# predictors
                y=dat$y,  # DV
                N=1113,  # sample size
                mu.beta=0,  # priors centered on 0
                tau.beta=1)  # diffuse priors

parnames <- c( "beta1","beta2","beta3","beta4")
mod <- jags(data = forJags,
                    parameters.to.save=parnames,
                    n.chains = 3, n.burnin = 1500, n.iter =1500 + 1000, n.thin = 10, model.file = textC
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1113
##    Unobserved stochastic nodes: 4
##    Total graph size: 4473
##
## Initializing model
```

```
mcmc.array <- mod$BUGSoutput$sims.array
#hist(c(mcmc.array[,,"beta[1]"]), freq = F, main = "", xlab ="Intercept")
#hist(c(mcmc.array[,,"beta[2]"]), freq = F, main = "", xlab ="Slope")
print ("Effect of Handgun")
```

```
## [1] "Effect of Handgun"
```

```
print (quantile(mcmc.array[,,"beta1"],c(.025,.975)))
```

```
##        2.5%       97.5%
## -0.19611121  0.01754014
```

```
print ("Effect of Rifle")
```

```
## [1] "Effect of Rifle"
```

```r
print (quantile(mcmc.array[,,"beta2"],c(.025,.975)))
```

```
##        2.5%       97.5%
## -0.08744894  0.37333149
```

We see that neither handgun nor rifle has a significant effect on the number of victims.

What if we control for race and age?

$$log(\lambda) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta 3 x_3 + \beta_5 r + \beta_6 a$$

Poisson w Covariates

```r
library(rjags)
library(R2jags)
model <- "model {
    ## Likelihood
for(i in 1:N){
      y[i] ~ dpois(lambda[i])
      log(lambda[i]) <- mu[i]
      mu[i] <- beta1*x1[i] +beta2*x2[i] + beta3*x3[i] + beta4  + beta5*race[i] + beta6*age[i]
      }
    ## Priors

    beta1 ~ dnorm(mu.beta,tau.beta)
    beta2 ~ dnorm(mu.beta,tau.beta)
    beta3 ~ dnorm(mu.beta,tau.beta)
    beta4 ~ dnorm(mu.beta,tau.beta)
    beta5 ~ dnorm(mu.beta,tau.beta)
    beta6 ~ dnorm(mu.beta,tau.beta)
}"

dat <- data.frame(x=type_of_gun,y=number_of_victims)

forJags <- list(x1=dat$x.1,
                x2=dat$x.2,
                x3=dat$x.3,
                x4=dat$x.4,
                age = age,
                race = race,# predictors
                y=dat$y,  # DV
                N=1113,  # sample size
                mu.beta=0,  # priors centered on 0
                tau.beta=1)  # diffuse priors

parnames <- c( "beta1","beta2","beta3","beta4","beta5","beta6")
mod <- jags(data = forJags,
                    parameters.to.save=parnames,
                    n.chains = 3, n.burnin = 1500, n.iter =1500 + 1000, n.thin = 10, model.file = textC
```

```
## Warning in jags.model(model.file, data = data, inits = init.values,
## n.chains = n.chains, : Unused variable "x4" in data
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 1113
##    Unobserved stochastic nodes: 6
##    Total graph size: 7746
##
## Initializing model
```

```r
mcmc.array <- mod$BUGSoutput$sims.array
#hist(c(mcmc.array[,,"beta[1]"]), freq = F, main = "", xlab ="Intercept")
#hist(c(mcmc.array[,,"beta[2]"]), freq = F, main = "", xlab ="Slope")
print ("Effect of Handgun")
```

```
## [1] "Effect of Handgun"
```

```r
print (quantile(mcmc.array[,,"beta1"],c(.025,.975)))
```

```
##        2.5%       97.5%
## -0.28255326 -0.07280948
```

```r
print ("Effect of Rifle")
```

```
## [1] "Effect of Rifle"
```

```r
print (quantile(mcmc.array[,,"beta2"],c(.025,.975)))
```

```
##       2.5%      97.5%
## -0.2116564  0.2327253
```

We still don't see any effect of rifle on number of victims, but we see a negative effect of a handgun. That is, we can't say that rifles kill more people than other weapons categories (what I hoped we would find), but we can say that handguns kill fewer people than other weapons. From a policy perspective it makes sense to limit weapons to handguns.

What happens if we control for the large number of zeros present in the data?

$$Y_t \sim \begin{cases} \pi_t + (1-\pi_t) \cdot e^{\mu_t} & \text{if } \ y_t = 0 \\ (1-\pi_t) \cdot Poisson(\mu_t) & \text{if } \ y_t > 0 \end{cases}$$

$$logit(\pi_t) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha 3 x_3 + \alpha_5 r + \alpha_6 a$$

$$log(\lambda_t) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta 3 x_3 + \beta_5 r + \beta_6 a$$

Zero Inflated Poisson w Covariates

```r
library(rjags)
library(R2jags)
model <- "model {
    ## Likelihood
    for(i in 1:N){
      y[i] ~ dpois(lambda.hacked[i])
      lambda.hacked[i] <- lambda[i]*(1-zero[i]) + 1e-10*zero[i]
      lambda[i] <- exp(mu.count[i])
      mu.count[i] <- beta1*x1[i] +beta2*x2[i] + beta3*x3[i] + beta4  + beta5*race[i] + beta6*age[i]

      ## Zero-Inflation
      zero[i] ~ dbern(pi[i])
      pi[i] <- ilogit(mu.binary[i])
      mu.binary[i] <- alpha1*x1[i] +alpha2*x2[i] + alpha3*x3[i] + alpha4  + alpha5*race[i] + alpha6*age
    }
```

```r
    for(i in 1:N){
      pp[i] ~ dpois(plambda.hacked[i])
      plambda.hacked[i] <- plambda[i]*(1-zero[i]) + 1e-10*zero[i]
      plambda[i] <- exp(pmu.count[i])
      pmu.count[i] <- beta1*x1[i] +beta2*x2[i] + beta3*x3[i] + beta4  + beta5*race[i] + beta6*age[i]

      ## Zero-Inflation
      pzero[i] ~ dbern(ppi[i])
      ppi[i] <- ilogit(pmu.binary[i])
      pmu.binary[i] <- alpha1*x1[i] +alpha2*x2[i] + alpha3*x3[i] + alpha4  + alpha5*race[i] + alpha6*age
    }
    ## Priors

    beta1 ~ dnorm(mu.beta,tau.beta)
    beta2 ~ dnorm(mu.beta,tau.beta)
    beta3 ~ dnorm(mu.beta,tau.beta)
    beta4 ~ dnorm(mu.beta,tau.beta)
    beta5 ~ dnorm(mu.beta,tau.beta)
    beta6 ~ dnorm(mu.beta,tau.beta)

    alpha1 ~ dnorm(mu.beta,tau.beta)
    alpha2 ~ dnorm(mu.beta,tau.beta)
    alpha3 ~ dnorm(mu.beta,tau.beta)
    alpha4 ~ dnorm(mu.beta,tau.beta)
    alpha5 ~ dnorm(mu.beta,tau.beta)
    alpha6 ~ dnorm(mu.beta,tau.beta)

}"

dat <- data.frame(x=type_of_gun,y=number_of_victims)

forJags <- list(x1=dat$x.1,
                x2=dat$x.2,
                x3=dat$x.3,
                x4=dat$x.4,
                age = age,
                race = race,# predictors
                y=dat$y,
                N=1113,  # sample size
                mu.beta=0,  # priors centered on 0
                tau.beta=1)  # diffuse priors

parnames <- c( "beta1","beta2","beta3","beta4","beta5","beta6","alpha1","alpha2","alpha3","alpha4","alp
mod <- jags(data = forJags,
                    parameters.to.save=parnames,
                    n.chains = 3, n.burnin = 1500, n.iter =1500 + 1000, n.thin = 10, model.file = textC
```

```
## Warning in jags.model(model.file, data = data, inits = init.values,
## n.chains = n.chains, : Unused variable "x4" in data
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
```

```
##    Observed stochastic nodes: 1113
##    Unobserved stochastic nodes: 3351
##    Total graph size: 16604
##
## Initializing model
```

```
mcmc.array <- mod$BUGSoutput$sims.array
#hist(c(mcmc.array[,,"beta[1]"]), freq = F, main = "", xlab ="Intercept")
#hist(c(mcmc.array[,,"beta[2]"]), freq = F, main = "", xlab ="Slope")
print ("Effect of Handgun")
```

```
## [1] "Effect of Handgun"
```

```
print (quantile(mcmc.array[,,"beta1"],c(.025,.975)))
```

```
##       2.5%      97.5%
## -0.1187452  0.1317035
```

```
print ("Effect of Rifle")
```

```
## [1] "Effect of Rifle"
```

```
print (quantile(mcmc.array[,,"beta2"],c(.025,.975)))
```

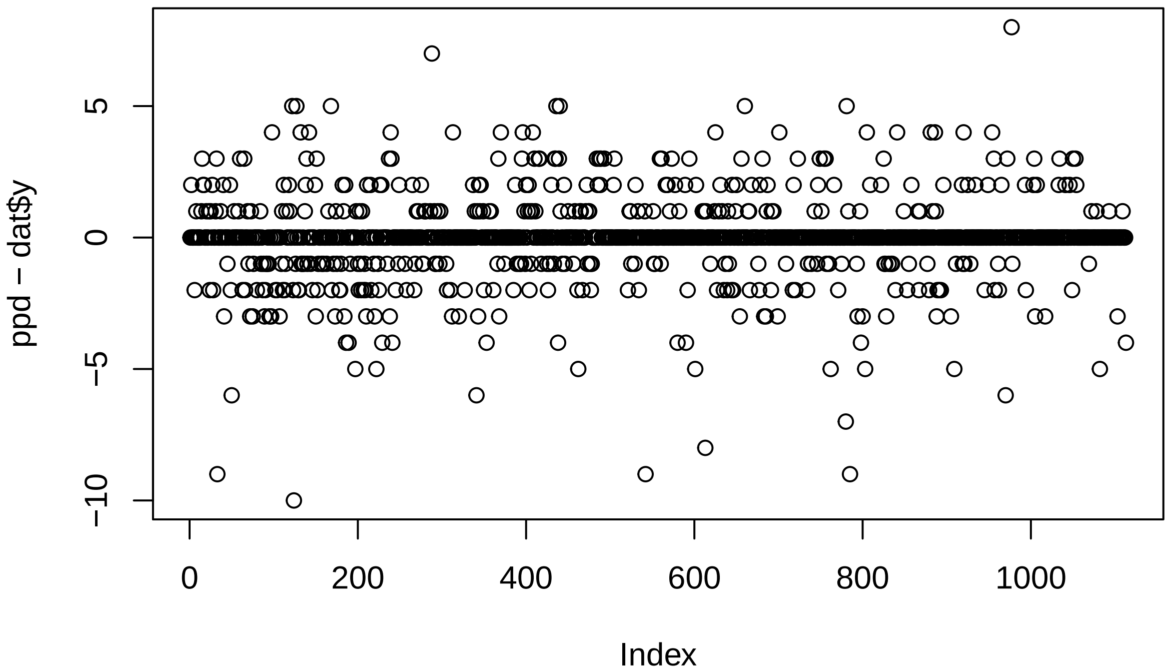
```
##       2.5%      97.5%
## 0.05290727 0.56661337
```

Ah-ha! if we use the Zero-inflated model we see that rifles do have a positive association with a higher number of victims. This makes sense because the effect of the weapon only matters if the shooter is able to use it (which we use 0 victims as a proxy).

Because we are handling time-series data, errors may be correlated across time. In order to check this, we examine the residuals of the ZIP model fit.

```
ppd <- c()
for (i in 1:length(dat$x.1)){
    ppd <- c(ppd,mcmc.array[,,paste(paste("pp[",i,sep=""),"]",sep="")][1])
}

plot(ppd-dat$y)
```



The residuals do look nice and centered around 0, so I'm not sure it is really necessary to to use an AR error? We can fit an AR(1) process to the residuals and see if it gives a reasonable fit.

Poisson + Ar with Covariates

```
library(rjags)
library(R2jags)
model <- "model {
    ## Likelihood
mu.count[1] <-  beta1*x1[1] +beta2*x2[1] + beta3*x3[1] + beta4  + beta5*race[1] + beta6*age[1]
for(i in 2:N){
      y[i] ~ dpois(lambda[i])
      lambda[i] <- exp(mu[i])
      mu[i] <- mu.count[i] + ar1 * ( y[i-1] - mu.count[i-1] )
      mu.count[i] <- beta1*x1[i] +beta2*x2[i] + beta3*x3[i] + beta4  + beta5*race[i] + beta6*age[i]


    }
    ## Priors
    ar1 ~ dunif(-1.1,1.1)
    beta1 ~ dnorm(mu.beta,tau.beta)
    beta2 ~ dnorm(mu.beta,tau.beta)
    beta3 ~ dnorm(mu.beta,tau.beta)
    beta4 ~ dnorm(mu.beta,tau.beta)
    beta5 ~ dnorm(mu.beta,tau.beta)
    beta6 ~ dnorm(mu.beta,tau.beta)




}"
```

```r
dat <- data.frame(x=type_of_gun,y=number_of_victims)

forJags <- list(x1=dat$x.1,
                x2=dat$x.2,
                x3=dat$x.3,
                x4=dat$x.4,
                age = age,
                race = race,# predictors
                y=dat$y,  # DV
                N=1113,  # sample size
                mu.beta=0,  # priors centered on 0
                tau.beta=1)  # diffuse priors

parnames <- c( "beta1","beta2","beta3","beta4","beta5","beta6","alpha1","alpha2","alpha3","alpha4","alph
mod <- jags(data = forJags,
                    parameters.to.save=parnames,
                    n.chains = 3, n.burnin = 1500, n.iter =1500 + 1000, n.thin = 10, model.file = textC
```

```
## Warning in jags.model(model.file, data = data, inits = init.values,
## n.chains = n.chains, : Unused variable "x4" in data

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1112
##    Unobserved stochastic nodes: 7
##    Total graph size: 10556
##
## Initializing model

## Warning in jags.samples(model, variable.names, n.iter, thin, type = "trace", : Failed to set trace mo
## Variable alpha1 not found

## Warning in jags.samples(model, variable.names, n.iter, thin, type = "trace", : Failed to set trace mo
## Variable alpha2 not found

## Warning in jags.samples(model, variable.names, n.iter, thin, type = "trace", : Failed to set trace mo
## Variable alpha3 not found

## Warning in jags.samples(model, variable.names, n.iter, thin, type = "trace", : Failed to set trace mo
## Variable alpha4 not found

## Warning in jags.samples(model, variable.names, n.iter, thin, type = "trace", : Failed to set trace mo
## Variable alpha5 not found

## Warning in jags.samples(model, variable.names, n.iter, thin, type = "trace", : Failed to set trace mo
## Variable alpha6 not found
```

```r
mcmc.array <- mod$BUGSoutput$sims.array
#hist(c(mcmc.array[,,"beta[1]"]), freq = F, main = "", xlab ="Intercept")
#hist(c(mcmc.array[,,"beta[2]"]), freq = F, main = "", xlab ="Slope")
print ("Effect of Handgun")
```

```
## [1] "Effect of Handgun"
```

```r
print (quantile(mcmc.array[,,"beta1"],c(.025,.975)))
```

```
##        2.5%       97.5%
```

```
## -0.31180731 -0.08811719
```

```r
print ("Effect of Rifle")
```

```
## [1] "Effect of Rifle"
```

```r
print (quantile(mcmc.array[,,"beta2"],c(.025,.975)))
```

```
##       2.5%      97.5%
## -0.2434123  0.1981317
```