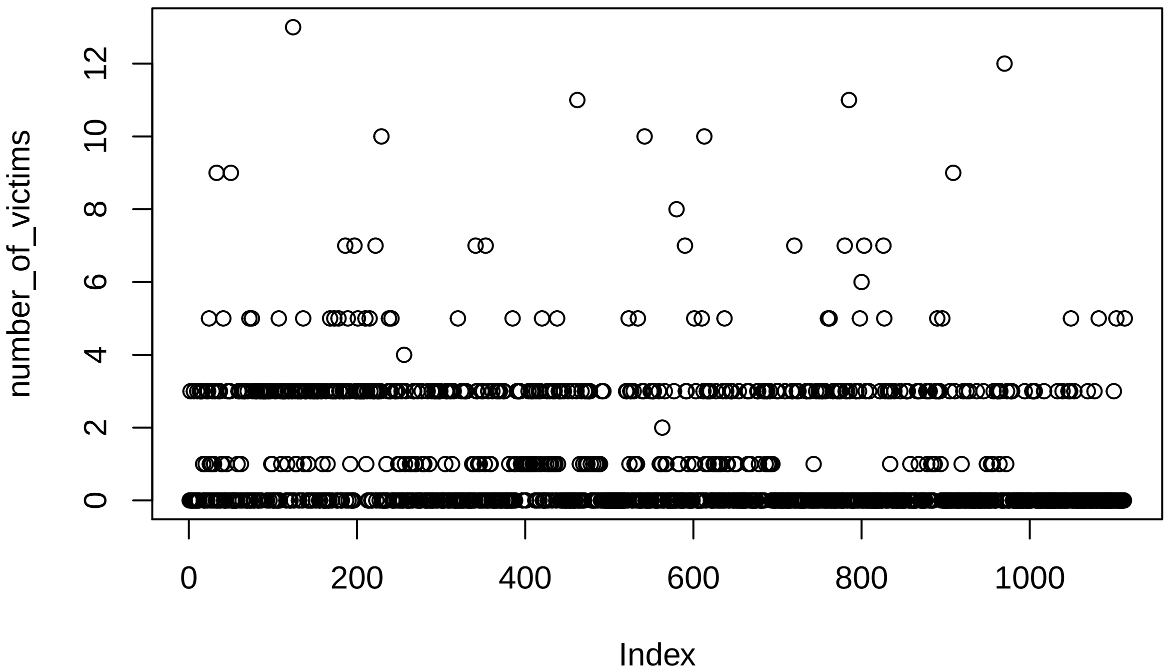
Bayes Project (Casey's Part)

Bianca, Heather, Casey

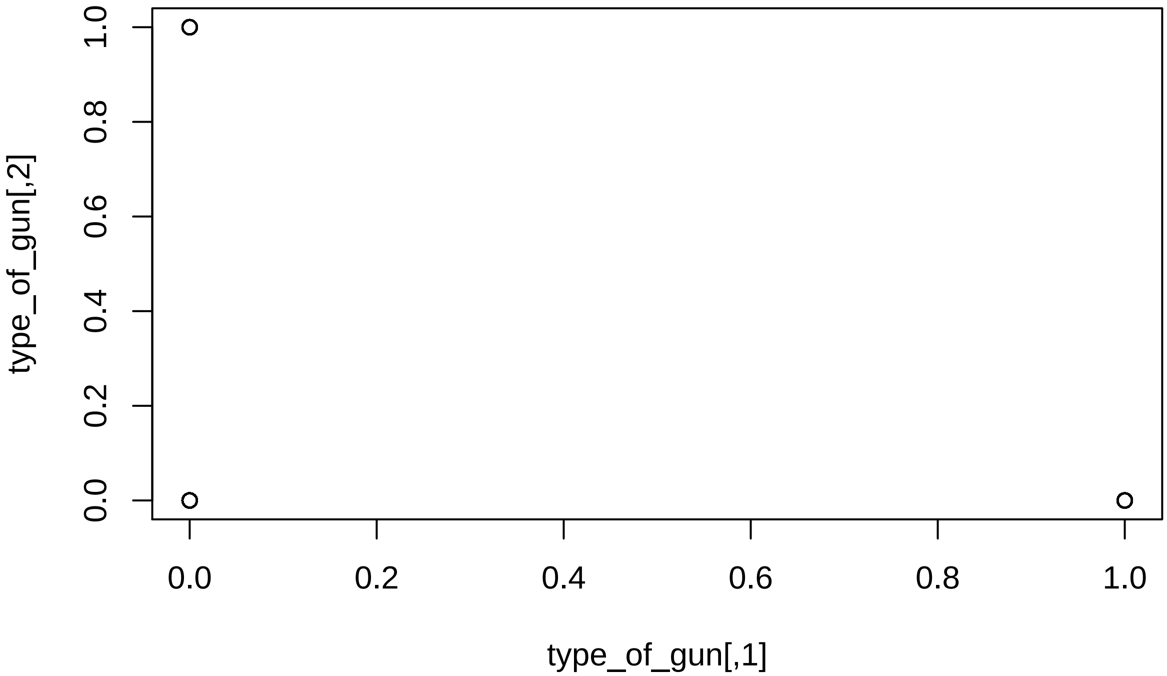
3/5/2018

Data

```r
#setwd("/Users/gcgibson/BayesProject/")

shootingsafter <- read.csv("/Users/gcgibson/BayesProject/ShootingsAfter1991.csv")
moreshootings <- read.csv("/Users/gcgibson/BayesProject/Shootings2016.csv")

write.csv(rbind(shootingsafter,moreshootings), "AllShootings.csv")

allshootings <- read.csv("/Users/gcgibson/BayesProject/AllShootings.csv")

ames.data <- allshootings$Victim.s..Deceased..at.school.

number_of_victims <- c()
for (i in 1:length(ames.data)){
  if (is.na(ames.data[i])){
    #print ("hello")
    number_of_victims <- c(number_of_victims,0)
  } else if (ames.data[i] == "None"){
    number_of_victims <- c(number_of_victims,0)
  } else{
    number_of_victims <- c(number_of_victims,ames.data[i])
  }
}

plot(number_of_victims)
```




```
type_of_gun <- matrix(0,nrow=length(number_of_victims),ncol=4)
tmp <- allshootings$Weapon.s..Categories

for (i in 1:length(tmp)){
  if (tmp[i] == "Handgun"){
    type_of_gun[i,1] =  1
  } else if (tmp[i] == "Rifle"){
    type_of_gun[i,2] =  1
  }  else if (tmp[i] == "Shotgun"){
    type_of_gun[i,3] =  1
  }  else {
    type_of_gun[i,4] =  1
  }
}
plot(type_of_gun)
```



We first attempt a simple poisson regression.

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
```

```r
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
```

```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%
## -0.199757955  0.006614321
```

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

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

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

```
##       2.5%      97.5%
## -0.1038271  0.3885998
```

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

What if we control for race and age?

Poisson w Covariates

```r
race <- allshootings$Shooter.s..or.Attacker.s..Race
race_clean <- c()
for (i in 1:length(race)){
  if (race[i] == "African American"){
    race_clean <- c(race_clean,0)
  } else if (race[i] == "Caucasian"){
    race_clean <- c(race_clean,1)
  }  else if (race[i] == "Hispanic"){
    race_clean <- c(race_clean,2)
  }   else{
    race_clean <- c(race_clean,4)
  }
}
```

```r
age <- allshootings$Shooter.s..or.Attacker.s..Age
age_clean <- c()
for (i in 1:length(age)){
  if ( 0 < as.numeric(age[i]) & as.numeric(age[i]) <10 ){
    age_clean <- c(age_clean,0)
  } else if (10 < as.numeric(age[i]) & as.numeric(age[i]) < 20){
    age_clean <- c(age_clean,1)
  }   else{
    age_clean <- c(age_clean,2)
  }
}
```

```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
```

```
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%
## -0.29082282 -0.05859833
```

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

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

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

```
##       2.5%      97.5%
## -0.1960608  0.2442154
```

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?

Zero Inflated Poisson w Covariates

```
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
    }
    ## 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,  # 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 = textCo
```

```
## 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: 1125
##    Total graph size: 14378
##
## 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%
## -0.1135156  0.1319552
```

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

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

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

```
##       2.5%      97.5%
## 0.05131746 0.53763073
```

Ah-ha! if we use the Zero-inflated model we see that rifles do have a positive association with a higher number of victims.

Just for fun we throw in an $AR(1)$ error process since our data is time-series data.

Poisson + Ar with Covariates

```r
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)




}"

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","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: 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 m
## Variable alpha1 not found

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

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

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

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

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

```
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%
## -0.3250752 -0.0839551
```

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

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

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

```
##       2.5%      97.5%
## -0.2671755  0.2145988
```