STAT431 Project - Revised

Chandler Nielsen

2023-04-16

Loading Libraries

```
library(tidyverse)
library(rjags)
library(readr)
```

Cleaning Data

```
### I'm scared. I pooped my pampers. Redoing Cleaning for Proper Binning, Mate.

### Reading in Data - Cleaning SDP - Changing SentenceTime to numeric

filthyData2 <- read_csv("filthyData2.csv")
```

```
## Rows: 29507 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): IDOC, Name, DOB, Sex, Race, Veteran, CurrentAdmissionDate, Admissi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
filthyData2 <- filthyData2[filthyData2$SentenceYears != "SDP", ]

sentenceYearVec <- ifelse(filthyData2$SentenceYears == "LIFE",
                          Inf,
                          as.numeric(filthyData2$SentenceYears))

sentenceMonthVec <- ifelse(is.na(filthyData2$SentenceMonths),
                           0,
                           as.numeric(filthyData2$SentenceMonths))

sentenceTimeVec <- sentenceYearVec + sentenceMonthVec / 12.0
filthyData2["SentenceTime"] <- sentenceTimeVec

###
```

```r
### Selecting Desired Columns

cleanedFilthyData <- filthyData2 %>%
  select("Sex", "Race", "Veteran", "CrimeClass", "SentencingCounty", "SentenceTime")

###

### Changing to Regions

regionVec = rep(0, length(cleanedFilthyData$SentencingCounty))
for (i in 1:length(cleanedFilthyData$SentencingCounty)) {

  if (cleanedFilthyData$SentencingCounty[i] == "Out of state") {
    regionVec[i] <- NA
    next
  }

  if (cleanedFilthyData$SentencingCounty[i] == 'Cook') {
    regionVec[i] <- 1
  }

  else if (cleanedFilthyData$SentencingCounty[i] %in% c("JoDaviess", "Stephenson", "Winnebago",
                                                        "Boone", "McHenry", "Lake", "Carroll",
                                                        "Ogle", "DeKalb", "Kane", "DuPage",
                                                        "Whiteside", "Lee", "Kendall",
                                                        "Grundy", "Will", "Kankakee")) {
    regionVec[i] <- 2
  }

  else if (cleanedFilthyData$SentencingCounty[i] %in% c("Rock Island", "Mercer", "Henry",
                                                        "Bureau", "LaSalle", "Henderson",
                                                        "Warren", "Knox", "Stark", "Putnam",
                                                        "Marshall", "Livingston", "Ford",
                                                        "Iroquois", "Vermillion", "Champaign",
                                                        "McLean", "Woodford", "Tazewell",
                                                        "Mason", "Peoria", "Fulton", "McDonough")) {
    regionVec[i] <- 3
  }

  else if (cleanedFilthyData$SentencingCounty[i] %in% c("Hancock", "Adams", "Schuyler",
                                                        "Brown", "Cass", "Menard", "Logan",
                                                        "Dewitt", "Piatt", "Douglas", "Edgar",
                                                        "Clark", "Coles", "Cumberland",
                                                        "Effingham", "Shelby", "Moultrie",
                                                        "Macon", "Christian", "Montgomery",
                                                        "Sangamon", "Morgan", "Macoupin",
                                                        "Green", "Jersey", "Calhoun", "Scott", "Pike"))
    regionVec[i] <- 4
  }

  else {
    regionVec[i] <- 5
  }
```

```
}

cleanedFilthyData["Region"] <- regionVec

###

### Changing Sex; Male = 1, Female = 2

cleanedFilthyDataSexVec <- ifelse(cleanedFilthyData$Sex == "Male", 1, 2)
cleanedFilthyData["SexNum"] <- cleanedFilthyDataSexVec

### Changing Race

## Races Key
## 1 - Black
## 2 - White
## 3 - Hispanic
## 4 - Asian
## 5 - American Indian
## 6 - Biracial
## 7 - Unknown

myRaceVec <- rep(0, length(cleanedFilthyData$Race))
for(i in 1:length(cleanedFilthyData$Race)) {

  if (cleanedFilthyData$Race[i] == "Black") {
    myRaceVec[i] <- 1
  }

  else if (cleanedFilthyData$Race[i] == "White") {
    myRaceVec[i] <- 2
  }

  else if (cleanedFilthyData$Race[i] == "Hispanic") {
    myRaceVec[i] <- 3
  }

  else if (cleanedFilthyData$Race[i] == "Asian") {
    myRaceVec[i] <- 4
  }

  else if (cleanedFilthyData$Race[i] == "American Indian") {
    myRaceVec[i] <- 5
  }

  else if (cleanedFilthyData$Race[i] == "Bi-Racial") {
    myRaceVec[i] <- 6
  }

  else {
    myRaceVec[i] <- 7
  }
```

```
}

cleanedFilthyData["RaceNum"] <- myRaceVec

###

### Changing Crime Class

## Crime Class Key
## 1 - Murder
## 2 - Class X
## 3 - Class I
## 4 - Class II
## 5 - Class III
## 6 - Class IV
## 7 - Unclassified

myCrimeClassVec <- rep(0, length(cleanedFilthyData$CrimeClass))
for(i in 1:length(cleanedFilthyData$CrimeClass)) {

  if (cleanedFilthyData$CrimeClass[i] == "Murder") {
    myCrimeClassVec[i] <- 1
  }

  else if (cleanedFilthyData$CrimeClass[i] == "Class X") {
    myCrimeClassVec[i] <- 2
  }

  else if (cleanedFilthyData$CrimeClass[i] == "Class 1") {
    myCrimeClassVec[i] <- 3
  }

  else if (cleanedFilthyData$CrimeClass[i] == "Class 2") {
    myCrimeClassVec[i] <- 4
  }

  else if (cleanedFilthyData$CrimeClass[i] == "Class 3") {
    myCrimeClassVec[i] <- 5
  }

  else if (cleanedFilthyData$CrimeClass[i] == "Class 4") {
    myCrimeClassVec[i] <- 6
  }

  else {
    myCrimeClassVec[i] <- 7
  }
}

cleanedFilthyData["CrimeNum"] <- myCrimeClassVec

###
```

```
### Veteran Status; 1 - Veteran, 2 - Not Veteran, 3 - Unknown

veteranVec <- rep(0, length(cleanedFilthyData$Veteran))
for (i in 1:length(cleanedFilthyData$Veteran)) {

  if (cleanedFilthyData$Veteran[i] == "Yes") {
    veteranVec[i] <- 1
  }

  else if (cleanedFilthyData$Veteran[i] == "No") {
    veteranVec[i] <- 2
  }

  else {
    veteranVec[i] <- 3
  }
}

cleanedFilthyData["VetNum"] <- veteranVec

###

### Completed cleaning, Rearranging Columns

cleanedData <- cleanedFilthyData %>%
  select("VetNum", "RaceNum", "SexNum", "CrimeNum", "Region", "SentenceTime")

cleanedData <- cleanedData[!is.na(cleanedData$Region), ]

###
```

Frequentist Analysis

```
withoutLife <- cleanedData[cleanedData$SentenceTime < Inf, ]

withoutLife$VetNum <- factor(withoutLife$VetNum)
withoutLife$RaceNum <- factor(withoutLife$RaceNum)
withoutLife$SexNum <- factor(withoutLife$SexNum)
withoutLife$CrimeNum <- factor(withoutLife$CrimeNum)
withoutLife$Region <- factor(withoutLife$Region)

myModel <- lm(SentenceTime ~ SexNum + RaceNum + VetNum +
                CrimeNum + Region, data = withoutLife)

summary(myModel)
```

```
##
## Call:
## lm(formula = SentenceTime ~ SexNum + RaceNum + VetNum + CrimeNum +
##     Region, data = withoutLife)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.73  -4.84  -1.06   2.50 556.47
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  47.77494    0.55693   85.782  < 2e-16 ***
## SexNum2      -2.90508    0.38676   -7.511 6.03e-14 ***
## RaceNum2     -0.60208    0.20260   -2.972 0.002963 **
## RaceNum3     -2.19877    0.25225   -8.717  < 2e-16 ***
## RaceNum4     -2.48652    1.36017   -1.828 0.067546 .
## RaceNum5      0.35601    2.24700    0.158 0.874113
## RaceNum6     -2.22139    1.76577   -1.258 0.208393
## RaceNum7     -4.82057    1.65085   -2.920 0.003503 **
## VetNum2      -2.04890    0.52754   -3.884 0.000103 ***
## VetNum3      -2.15304    0.53982   -3.988 6.67e-05 ***
## CrimeNum2   -27.11846    0.23630 -114.762  < 2e-16 ***
## CrimeNum3   -35.44837    0.30055 -117.946  < 2e-16 ***
## CrimeNum4   -39.21650    0.26813 -146.257  < 2e-16 ***
## CrimeNum5   -41.44308    0.33834 -122.489  < 2e-16 ***
## CrimeNum6   -42.65738    0.36575 -116.628  < 2e-16 ***
## CrimeNum7   -44.09684   13.27816   -3.321 0.000898 ***
## Region2      -0.24846    0.22995   -1.080 0.279931
## Region3       1.37081    0.26751    5.124 3.01e-07 ***
## Region4      -0.15861    0.30056   -0.528 0.597698
## Region5      -0.05627    0.26407   -0.213 0.831272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.27 on 27856 degrees of freedom
## Multiple R-squared:  0.5353, Adjusted R-squared:  0.535
## F-statistic:  1689 on 19 and 27856 DF,  p-value: < 2.2e-16
```

First Bayesian Model - LIFE removed

```
initsSet1 <- list(list(Race = c(0, -0.6, -2.2, -2.4, 0.4, -2.2, -4.8),
                       CrimeType = c(0, -27.1, -35.4, -39.2, -41.4, -42.7, -44.1),
                       SexType = c(0, 2.9),
                       RegionType = c(0, -0.2, 1.4, -0.1, -0.04)),

                  list(Race = c(10, -10, 10, -10, 10, -10, 10),
                       CrimeType = c(20, 17.5, 15, 0, -15, -17.5, -20),
                       SexType = c(2.9, 0),
                       RegionType = c(-1, 1, -1, 1, -1)),

                  list(Race = c(-10, 10, -10, 10, -10, 10, -10),
                       CrimeType = c(-20, -17.5, -15, 0, 15, 17.5, 20),
                       SexType = c(-10, 10),
                       RegionType = c(10, -10, 10, -10, 10)))

cat('
model {
```

```
  for(i in 1:length(SentenceTime)) {
    SentenceTime[i] ~ dnorm(mu[i], tausq)
    mu[i] <- CrimeType[CrimeNum[i]] + SexType[SexNum[i]] + Race[RaceNum[i]] + RegionType[Region[i]]
  }

  tausq ~ dgamma(0.0001,0.0001)

  for (j in 1:5) {
    RegionType[j] ~ dnorm(0, 0.01)
  }

  for (k in 1:2) {
    SexType[k] ~ dnorm(0, 0.01)
  }

  for (l in 1:7) {
    Race[l] ~ dnorm(0, 0.01)
  }

  for (w in 1:7) {
    CrimeType[w] ~ dnorm(0, 0.01)
  }

}
', file = {revisedBayes1 = tempfile()})

revisedBayes1 <- jags.model(revisedBayes1, withoutLife, initsSet1, n.chains=3)
revised1sample1 <- coda.samples(revisedBayes1,
                                c("CrimeType", "SexType", "Race", "RegionType"),
                                n.iter=50000)

gelman.plot(revised1sample1, autoburnin=FALSE)
gelman.diag(revised1sample1, autoburnin=FALSE)

summary(window(revised1sample1, 48000, 50000))
```