CS 598: Advanced Bayesian Modeling Data Analysis Report

Shyam Shah

03/05/2021

Introduction

Police stop and searches are conducted at the discretion of police officers if they feel they have reasonable suspicion. [1]. However the idea of reasonable suspicion has come under scrutiny as different police officers may consider certain things in a specific situation as evidence for reasonable suspicion. With many cases of police brutality against people of color in the United States being publicized recently, it would be natural to think that there is a racial bias in US policing. However some researchers that looked at the data were getting interpretations with a lot of variation [2]. The purpose of this paper will be to look at data used by Rivera & Rosenbaum (2020) [3] to see if this it shows signs of racial bias.

Data

The data used for our analysis consists of 4 variables:

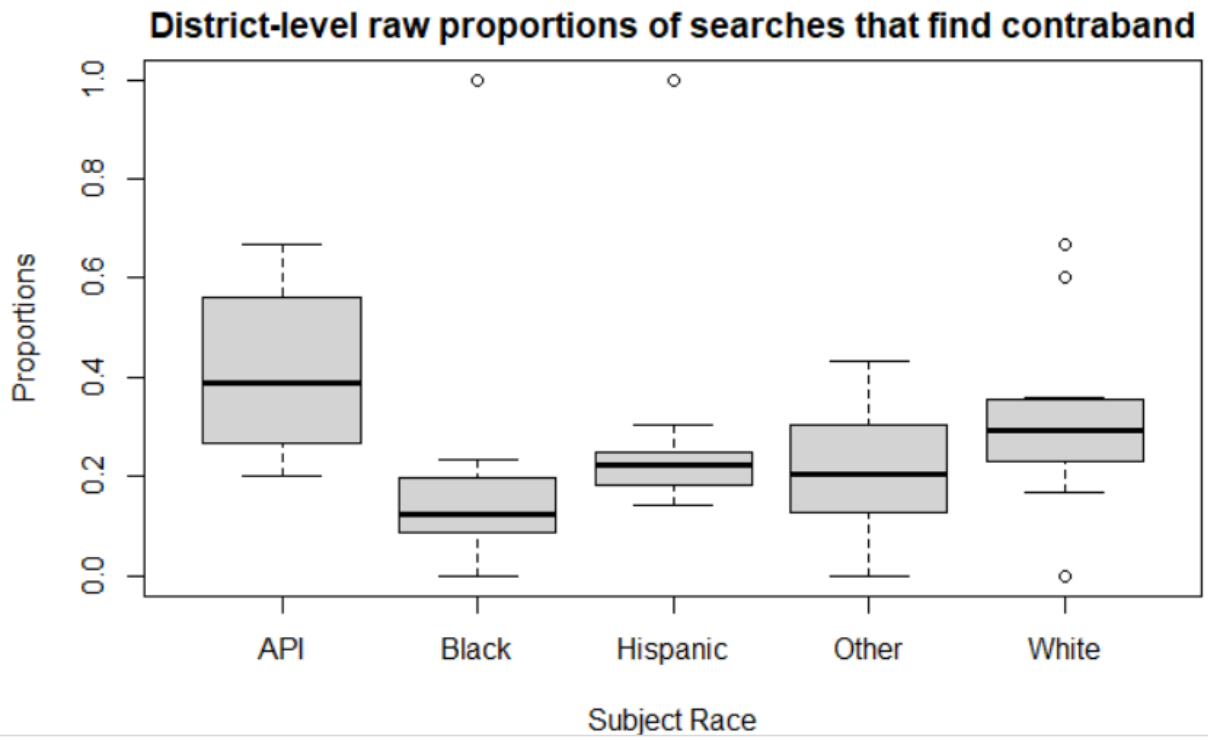
- SubjectRace: the racial grouping of the searched subject
- District: the designation for police district reporting the search
- ContrabandFound: the number of searches (out of the total) in which contraband was found
- TotalSearches: the total number of searched conducted (subsequent to police stops)

There are 5 different subject races observed in this data, and below are the boxplots for each race (over all districts):

[1] Rob Usry The Legal Basis for Warrantless Police Stops and Searches https://www.bhollandlawfirm.com/library/when-can-police-stop-and-search-you-without-a-warrant-.cfm

[2] Lynne Peeples What the data say about police shootings https://www.nature.com/articles/d41586-019-02601-9

[3] Rivera, R., & Rosenbaum, J. (2020, August). Racial disparities in police stops in US cities.



These boxplots aggregate over all districts, but there were race + district combinations for which there were no searches. In district S, there were no searches for API (Asian/Pacific Islander) or Hispanice people, and in district T there were no searches for API people. See Appendix 1 for the code to generate these plots.

Based on the boxplots above, it looks like the API group generally has the highest proportion of searches that find contraband, and the Black group has the smallest.

First Model

The first model we will use is a logistic regression model. The response variable represents the number of searches in which contraband was found for each race/district combination and has a binomial distribution with the number of searches used as the n parameter, and some probability p_i modeled through the logit link function which is a linear combination of race and district variables with coefficients β. The parameters are the coefficients for the race and district. The hyperparameter σ is used in the distribution for the coefficient of district parameter. Because of this hyperparameter, the district parameter is treated as a random effect, while the race parameter is treated as a fixed effect.

This is the JAGS model used in this section:

```
model {

  for (i in 1:length(searches)) {
    found[i] ~ dbin(prob[i], searches[i])
    logit(prob[i]) <- betarace[race[i]] + betadistrict[district[i]]

    foundrep[i] ~ dbin(prob[i], searches[i])
  }

  for (j in 1:max(race)) {
    betarace[j] ~ dt(0, 0.01, 1)
```

```
  }

  for (k in 1:max(district)) {
    betadistrict[k] ~ dnorm(0, 1/sigmadistrict^2)
  }

  sigmadistrict ~ dunif(0,10)

}
```

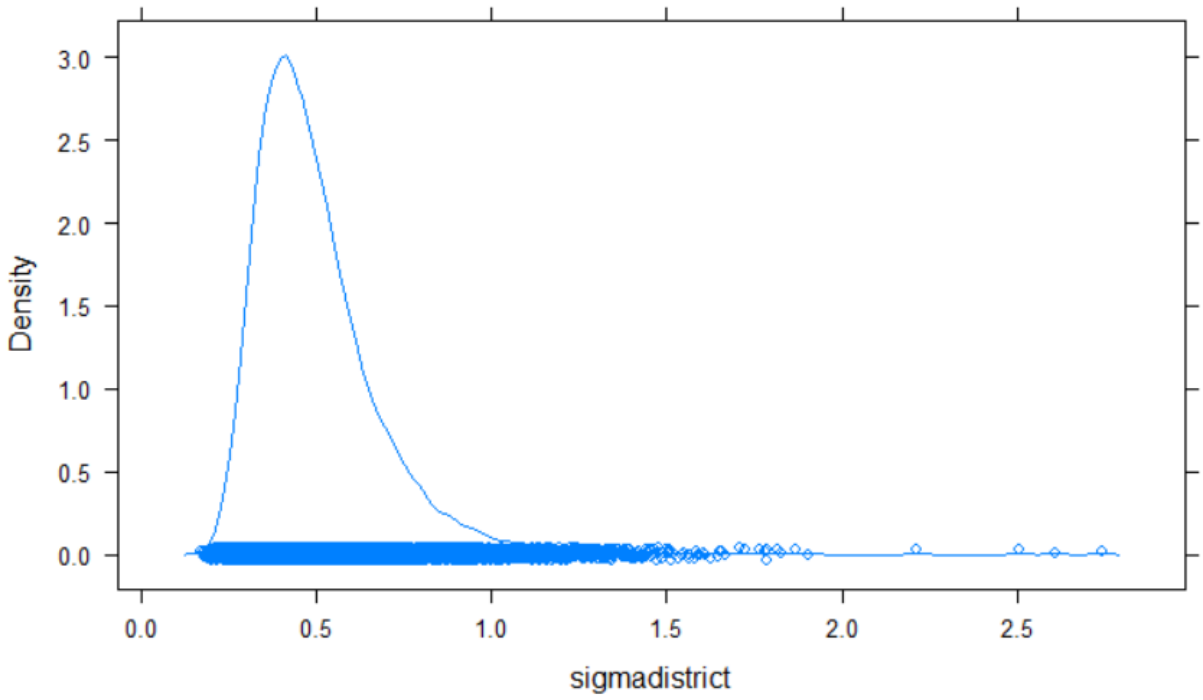
See Appendix 2 for the code used for all of the analysis done on this model.

I used four chains with dispersed initial values, 50000 iterations of burn in, 40000 iterations per chain with a thinning interval of 4. The effective sample sizes for the top level parameters are listed below:

Parameter	Sample Size
$\beta_{district1}$	3252.625
$\beta_{district2}$	2746.344
$\beta_{district3}$	2152.931
$\beta_{district4}$	2406.275
$\beta_{district5}$	2826.485
$\beta_{district6}$	3905.391
$\beta_{district7}$	3951.983
$\beta_{district8}$	2728.735
$\beta_{district9}$	2943.713
$\beta_{district10}$	2417.745
$\beta_{district11}$	21614.661
$\beta_{district12}$	21025.812
β_{race1}	3084.941
β_{race2}	2082.154
β_{race3}	2362.085
β_{race4}	3166.455
β_{race5}	2283.607
$\sigma_{district}$	9197.577

See Appendix 4 to see how these β's map to actual races and districts.

This is a density plot for $\sigma_{district}$:



The skew suggests that there are actual differences among the districts.

$\beta_B < \beta_W$ indicates that contraband is less likely to be found on a black subject because it gives the binomial distribution used to sample how much contraband is found in our model a lower probability for black subjects. The approximate posterior probability that $\beta_B < \beta_W$ is 0.999725, meaning that the model is almost certain that contraband is less likely to be found on a black subject.

The posterior predictive p-value based on the chi-squre discrepancy is 0.023675 which is near 0, indicating that there is over dispersion.

The approximate value of Plummer's DIC is 312.5 and it's associated effective number of parameters is 14.13. The actual number of parameters is 18 (1 for the 5 β_{race}, 1 for the 12 $\beta_{district}$, and then $\sigma_{district}$). So the effective number of parameters is less than the actual number of parameters.

Second Model

Since we noticed overdispersion in the first model, we'll use a slightly modified one to address this. This is the JAGS model used for this section:

```
model {

  for (i in 1:length(searches)) {
    found[i] ~ dbin(prob[i], searches[i])
    logit(prob[i]) <- betarace[race[i]] + betadistrict[district[i]] + epsilon[i]

    epsilon[i] ~ dnorm(0, 1/sigmaepsilon^2)
    foundrep[i] ~ dbin(prob[i], searches[i])
  }

  for (j in 1:max(race)) {
    betarace[j] ~ dt(0, 0.01, 1)
```

```
  }

  for (k in 1:max(district)) {
    betadistrict[k] ~ dnorm(0, 1/sigmadistrict^2)
  }

  sigmadistrict ~ dunif(0,10)
  sigmaepsilon ~ dunif(0,10)

}
```

See Appendix 3 for the code used for all of the analysis done on this model.

I used four chains with dispersed initial values, 50000 iterations of burn in, 40000 iterations per chain with a thinning interval of 4. The effective sample sizes for the top level parameters are listed below:

Parameter	Sample Size
$\beta_{district1}$	5383.711
$\beta_{district2}$	4528.651
$\beta_{district3}$	3265.661
$\beta_{district4}$	3754.679
$\beta_{district5}$	4151.657
$\beta_{district6}$	7183.531
$\beta_{district7}$	6813.523
$\beta_{district8}$	3957.404
$\beta_{district9}$	4840.803
$\beta_{district10}$	4065.975
$\beta_{district11}$	20641.211
$\beta_{district12}$	21596.440
β_{race1}	4681.498
β_{race2}	2739.354
β_{race3}	3281.008
β_{race4}	4564.866
β_{race5}	3353.906
$\sigma_{district}$	9195.450
σ_{ϵ}	3475.036

See Appendix 4 to see how these β's map to actual races and districts.

The approximate posterior probability that $\beta_B < \beta_W$ is 0.99795. There is almost no change in this value between the two volumes, so the same conclusion still applies.

The approximate value of Plummer's DIC is 302.8 (lower than the DIC for the first model) and it's associated effective number of parameters is 28.93. Since this model's value is lower, it can be considered as better than the first one.

Conclusion

The initial analysis done on the provided data showed that there was an imbalance in the percentage of stop and searches that resulted in contraband being found between races and over all districts. Our first model showed that there potentially were differences across districts but both models ultimately showed that contraband is less likely to be found in a search of Black subjects over White subjects. This suggests that

there does seem to be racial bias influencing police decisions to conduct stop and searches on people in the two cities that the data was taken from.

Appendix 1

This is the code used to generate the box plots in the introduction.

```
data <- read.csv("./policesearchSanFransisco.csv")
  boxplot(ContrabandFound/TotalSearches~SubjectRace,
  data=data,
  main="District-level raw proportions of searches that find contraband",
  xlab="Subject Race",
  ylab="Proportions")
```

Appendix 2

The initial analysis of our data

```
# prepare the data separately first
found <- data$ContrabandFound
searches <- data$TotalSearches
race <- unclass(factor(data$SubjectRace))
district <- unclass(factor(data$District))

# prepare the data for the model
d1 <- list(found=found, searches=searches, race=race, district=district)

# prepare initial values
inits1 <- list(list(betarace=c(10, 10, 10, 10, 10),
                    betadistrict=c(10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10),
                    sigmadistrict=0.01,
                    .RNG.name="base::Wichmann-Hill", .RNG.seed=123456),
               list(betarace=c(10, 10, 10, -10, -10),
                    betadistrict=c(10, 10, 10, 10, 10, 10, 10, 10, 10, -10, -10, -10),
                    sigmadistrict=9,
                    .RNG.name="base::Wichmann-Hill", .RNG.seed=123455),
               list(betarace=c(10, 10, -10, -10, -10),
                    betadistrict=c(-10, -10, -10, 10, 10, 10, 10, 10, 10, 10, -10, -10),
                    sigmadistrict=0.01,
                    .RNG.name="base::Wichmann-Hill", .RNG.seed=123454),
               list(betarace=c(10, -10, -10, -10, -10),
                    betadistrict=c(-10, -10, -10, 10, -10, 10, -10, 10, -10, -10, -10, -10),
                    sigmadistrict=9,
                    .RNG.name="base::Wichmann-Hill", .RNG.seed=123453))

library (rjags)


#create model
m1 <- jags.model("./firstmodel.bug", d1, inits1, n.chains=4, n.adapt=1000)

#burn-in
```

```
update(m1, 50000)

#get samples
x1 <- coda.samples(m1, c("betarace", "betadistrict", "sigmadistrict", "prob", "foundrep"), n.iter=40000

#check for convergence
gelman.diag(x1, autoburnin=FALSE, multivariate=FALSE) #shows convergence if < 1.1

#check effective sample sizes are > 2000
effectiveSize(x1)
```

This is the code used to generate the density plot for $\sigma_{district}$

```
require(lattice)

densityplot(as.matrix(x1)[,138], xlab="sigmadistrict")]
```

This is the code used to compute the posterior predictive probability that $\beta_B < \beta_W$.

```
mean(as.matrix(x1[,c("betarace[2]")])<as.matrix(x1[,c("betarace[4]")]))
```

This is the code used to compute the predictive p-value based on the chi square discrepancy

```
probs <- as.matrix(x1)[, paste("prob[",1:60,"]", sep="")]
foundrep <- as.matrix(x1)[, paste("foundrep[",1:60,"]", sep="")]

Tchi <- numeric(nrow(foundrep))
Tchirep <- numeric(nrow(foundrep))

for (s in 1:nrow(foundrep)){
  Tchi[s] <- sum((data$ContrabandFound - data$TotalSearches*probs[s,])^2 /
                   (data$TotalSearches*probs[s,]*(1-probs[s,])), na.rm=T)
  Tchirep[s] <- sum((foundrep[s,] - data$TotalSearches*probs[s,])^2 /
                   (data$TotalSearches*probs[s,]*(1-probs[s,])), na.rm=T)
}

mean(Tchirep>=Tchi)
```

This is the code used to calculate the value of Plummer's DIC and the associated effective sample size.

```
dic.samples(m1, 100000)
```

Appendix 3

This is the code used for setting up and running the second model.

```
# (b)

# prepare the data for the model
d2 <- list(found=found, searches=searches, race=race, district=district)
```

```
# prepare initial values
inits2 <- list(list(betarace=c(10, 10, 10, 10, 10),
                    betadistrict=c(10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10),
                    sigmadistrict=0.01,
                    sigmaepsilon=0.01),
               list(betarace=c(10, 10, 10, -10, -10),
                    betadistrict=c(10, 10, 10, 10, 10, 10, 10, 10, 10, -10, -10, -10),
                    sigmadistrict=9,
                    sigmaepsilon=0.01),
               list(betarace=c(10, 10, -10, -10, -10),
                    betadistrict=c(-10, -10, -10, 10, 10, 10, 10, 10, 10, 10, -10, -10),
                    sigmadistrict=0.01,
                    sigmaepsilon=9),
               list(betarace=c(10, -10, -10, -10, -10),
                    betadistrict=c(-10, -10, -10, 10, -10, 10, -10, 10, -10, -10, -10, -10),
                    sigmadistrict=9,
                    sigmaepsilon=9))

#create model
m2 <- jags.model("./secondmodel.bug", d2, inits2, n.chains=4, n.adapt=1000)

#burn-in
update(m2, 50000)

#get samples
x2 <- coda.samples(m2, c("betarace", "betadistrict", "sigmadistrict", "prob", "foundrep", "sigmaepsilon

#check for convergence
gelman.diag(x2, autoburnin=FALSE, multivariate=FALSE)

#check effective sample sizes are > 2000
effectiveSize(x2)
```

This is the code used to compute the posterior predictive probability that $\beta_B < \beta_W$.

```
mean(as.matrix(x2[,c("betarace[2]")])<as.matrix(x2[,c("betarace[4]")]))
```

This is the code used to calculate the value of Plummer's DIC and the associated effective sample size.

```
dic.samples(m2, 100000)
```

Appendix 4

This is the mapping of variable names used to their actual values.

Variable	Actual Value
district1	A
district2	B
district3	C
district4	D
district5	E

Variable	Actual Value
district6	F
district7	G
district8	H
district9	I
district10	J
district11	S
district12	T
race1	API
race2	Black
race3	Hispanic
race4	White
race5	Other