# Bayesian Analysis of Crime Rate in Denver, Colorado

Shalini Mahanthege

STAT 8640, Fall 2021

A neighbourhood with a lower crime rate to live is a major factor considered in the property investment decision. Identifying high crime neighbourhoods is a useful factor for assigning police forces as well. In this project, we aim to categorize neighbours based on the crime rate as either a high crime rate or a lower crime rate compared to the average monthly crime rate. We study the crime rates in neighbourhoods of the city and county of Denver, Colorado.

Our objective is to model monthly crime rate category based on two different Bayesian models using Gibbs sampling.

## 1 Data Description

Two datasets were used in this study which were obtained from the Denver Open Data Catalog [1]. Crime Data - Criminal offenses in the city and county of Denver, Colorado from January 2nd, 2016 to November 2021. Crimes from 2016 to October 2021 were considered in the data modeling process while the last month, November 2021 were used for measuring model classification error and comparison of different models. Crime data is based on the National Incident Based Reporting System(NIBRS). American community survey neighborhoods - Five year average of neighborhood characteristics derived from the American Community Survey from years 2015-2019. This dataset includes different demographic characteristics of each neighborhood level. There are 78 neighborhoods in the city and county of Denver.

Crime data include many information about crimes such as the time, geo-location of the incident and the category of crime. Figure 1 shows the different crime catogories reported in Denver. Since we are interested in crimes that we suspect is influenced by the characteristics of the neighbourhood, crime selected the analysis are, Arson, Auto-theft, Burglary, Larceny, Public disorder, Robbery and Theft from motor vehicle.

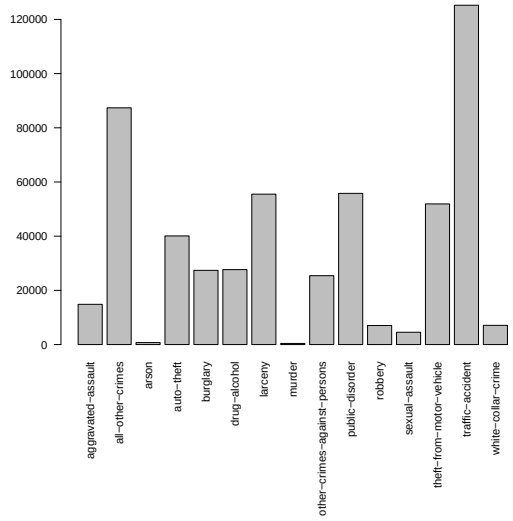
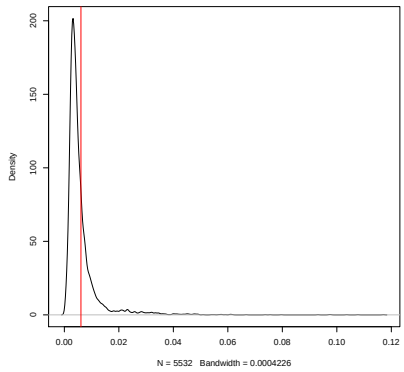
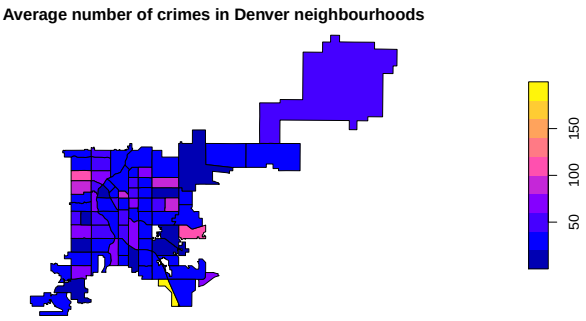



Figure 1: Crime categories in Denver from 2016-2021




(a) Crime rate density




(b) Spatial distribution of crime rates

Figure 2: Crime rates in Denver neighbourhoods

In figure 2 we show how the monthly crime rate varies in each neighbourhood. Figure 2 red line indicates the average monthly crime rate which is 0.00609 which is used as the cutoff point to identify high crime neighbourhoods and low crime neighbourhoods.

## 2 Data Model

Let $r_i(t)$ be the monthly crime rate for each neighborhood $i = 1, 2, ..., 78$ where $i$ is the neighborhood index and $t \in \{1, 2, ..., 71\}$ is the month in the time series. We compute $r_i(t)$ as,

$$r_i(t) = \frac{\text{Number of crimes on i neighborhood on month } t}{\text{Total population in the neighborhood i}}$$

where $0 < r_i(t) < 1$. Now we define N binary random variables $(Y_1(t), ..., Y_N(t))$ to represent the neighborhoods with high crime rate or lower crime rate compared to the average monthly crime rate(0.00609) computed based on crime from 2016-2021.

$$Y_i(t) = \begin{cases} Y_i(t) = 1 & \text{if } r_i(t) \geq \text{Average crime rate} \\ Y_i(t) = 0 & \text{if } r_i(t) < \text{Average crime rate} \end{cases}$$

Then $Y_i(t)$ is distributed Bernoulli with probability of success $p_i$. We define the Probit model for binary regression as,

$$P[Y_i(t) = 1] = p_i = \Phi(\mathbf{x}_i^T \beta)$$

where $i = 1, ...N$, $t = 1, 2, ..., 71$ and $\Phi$ is the cdf of the Standard Gaussian model. $\beta$ is a $10 \times 1$ vector of unknown parameters and $\mathbf{x}_i$ is a known vector of covariates such that,

$$\mathbf{x}_i^T \beta = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + ... + \beta_7 x_{i7} + \beta_8 t$$

$x_{1i} = 1, ..., 78$ neighborhood number as a categorical variable.
$x_{2i}$ =Average household income for the $i$th neighborhood.
$x_{3i}$ = number of people with less than high school diploma level education in neighborhood $i$.
$x_{4i}$ = Hispanic ethnicity percentage in neighborhood $i$.
$x_{5i}$ = White ethnicity percentage in neighborhood $i$.
$x_{6i}$ = Black ethnicity percentage in neighborhood $i$.
$x_{7i}$ = percentage of families in poverty in neighborhood $i$.
$t = 1, ..., 71$ month as a numerical value starting from January 2016 as the first month.

# 3 Probit Model Gibbs Sampling

To model above probit model in Bayesian analysis, we follow the procedure proposed by [2]. We introduce $N$ latent variables $Z_1, ..., Z_N$ where the $Z_i$ are independent $N(\mathbf{x}_i^T\beta, 1)$ random variables for $N$ observations.

$$Y_i(t) = \begin{cases} 1 & \text{if } Z_i > 0 \\ 0 & \text{if otherwise} \end{cases}$$

Then the joint posterior density of $\beta$ and $\mathbf{Z}$ given the data $\mathbf{y} = (y_1, ..., y_N)$ is given by,

$$\pi(\beta, \mathbf{Z}|\mathbf{y}) = C\pi(\beta)\prod_{i=1}^{N}\{I(Z_i > 0)I(y_i = 1) + I(Z_i \leq 0)I(y_i = 0)\} \times \phi(Z_i; \mathbf{x}_i^T\beta, 1)$$

where $\phi(;\mu, \sigma^2)$ is the $N(\mu, \sigma^2)$ pdf and $I(X \in A)$ is the indicator function which is equal to 1 if the random variable $\mathbf{X}$ is contained in the set $A$. And $C$ is a proportionality constant. Given a flat non informative prior on $\beta$ we can write the posterior distributions as,

$$\beta|\mathbf{y}, \mathbf{Z} \sim N_k(\hat{\beta}_z, (\mathbf{X}^T\mathbf{X})^{-1})$$

where $\hat{\beta}_z = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Z})$ Next $Z_1, ..., Z_N$ are independent with

$$Z_i|\mathbf{y}, \beta \sim N(\mathbf{x}_i^T\beta, 1)\begin{cases} \text{truncated at the left by 0} & \text{if } y_i = 1 \\ \text{truncated at the right by 0} & \text{if } y_i = 0 \end{cases}$$

Initial value for $\beta$ is chosen as the MLE(Maximum likelihood estimator) obtained from GLM model with probit link function in R. 10000 simulations were run with a burn-in of 5000. In figure 3 we see that $\beta_3$ does not seem to converge after 10,000 simulations. But all the density plots indicate that the simulations are normally distributed. We can conclude that all $\beta$ coefficients other than $\beta_3$ converges. ACF plots in figure 4 indicates that auto correlation between draws are decreasing with increasing lag but there seems to also be high autocorrelation in $\beta_3$ which is the household income value. $\beta$ coefficient mean and their standard deviations are given in table 1.

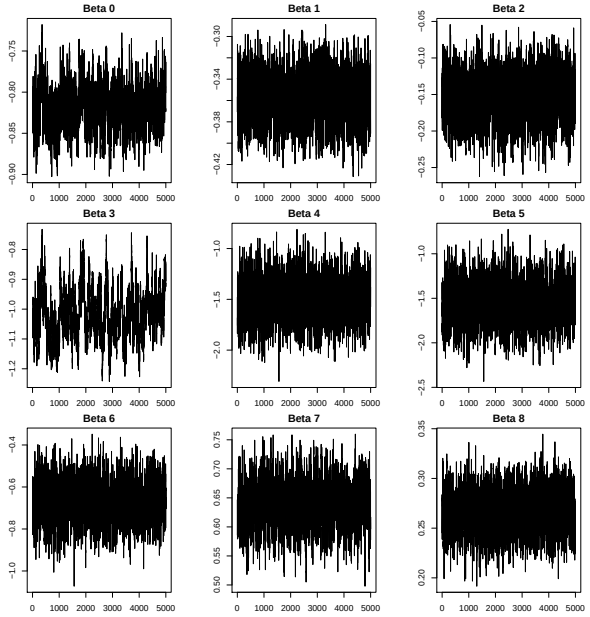
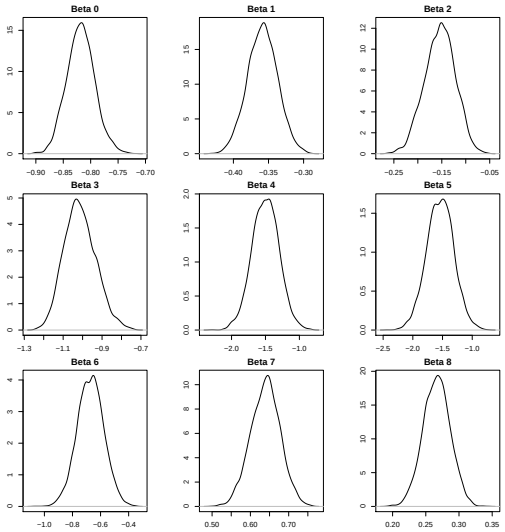
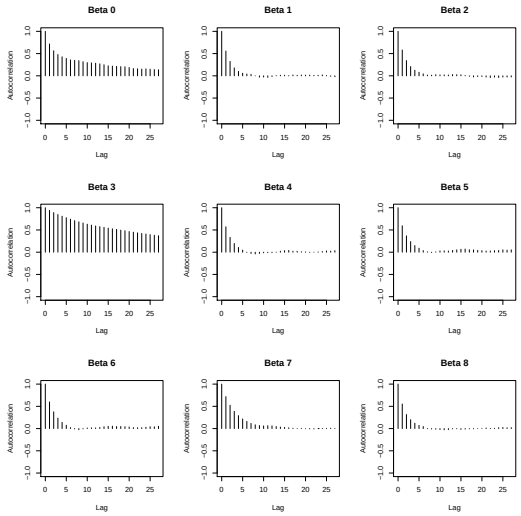



Figure 3: Trace plots of $\beta$ for probit model




(a) Density plots for simulated $\beta$




(b) ACF plots for simulated $\beta$

Figure 4: Simulation results for probit model

# 4 t link

Instead of a probit link function [2] suggests using the family of $t$ distributions. Since logit function which is one of the most popular link functions for binary response variables, we use a $t$ distribution with 8 degres of freedom which [2] states that have approximately the same quantiles as logit quantiles. In our analysis we try $t \in \{2, 4, 8, 16, 32\}$ to compare results and deviations.

We introduce $N$ latent variables $Z_1, ..., Z_N$ where the $Z_i$ are independent from $t$ distribution with $\nu$ degrees of freedom. Then we can write the scale mixture of normal distributions given $\lambda_i$ as $Z_i|\lambda_i \sim N(\mathbf{x}_i^T\beta, \lambda_i^{-1})$ and $\lambda_i$ is distributed Gamma$(\nu/2, 2/\nu)$ prior.

Now we can define the $t$-link model for binary regression as,

$$P[Y_i(t) = 1] = p_i = \Phi(\lambda_i^{1/2}\mathbf{x}_i^T\beta)$$

Now the posterior densities for **Z,X**, $\beta$ and $\nu$ is given by,

$$\beta|\mathbf{y}, \mathbf{Z}, \lambda, \nu \sim N_k(\hat{\beta}_{z,\lambda}, (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}) \text{ where,}$$
$$\hat{\beta}_z = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Z}) \text{ and } \mathbf{W} = \text{diag}(\lambda_i)$$

$$Z_i|\mathbf{y}, \beta, \lambda, \nu \sim N(\mathbf{x}_i^T\beta, \lambda_i^{-1}) \begin{cases} \text{truncated at the left by 0} & \text{if } y_i = 1 \\ \text{truncated at the right by 0} & \text{if } y_i = 0 \end{cases}$$

$$\lambda_i|\mathbf{y}, \beta, \lambda, \nu \sim \text{Gamma}\left(\frac{\nu+1}{2}, \frac{2}{\nu + (Z_i - \mathbf{x}_i^T\beta)^2}\right)$$

$$p(\nu|\mathbf{y}, \mathbf{Z}, \beta, \lambda) \approx \pi(\nu)\prod_{i=1}^{N}(c(\nu)\lambda_i^{\nu/2-1}e^{-\nu\lambda_i/2})$$

This represents a hierarchical Bayesian model which we can simulate using Gibbs sampling. Initial values for $\beta$ was selected at the maximum likelihood estimators from the GLM function in R with a probit link function. Since this simulations takes longer than the probit bayesian gibbs sampling approach 5000 simulations were run with 1000 burn-ins. Initial values for $\lambda$ was taken to be 1 and the simulation was done for different $\nu \in \{2, 4, 8, 16, 32\}$.

Figure 5 shows the trace plot for $t$ link simulation with 8 degrees of freedom. All $\beta$ coefficients seems to converge except of $\beta_3$ which again is the household income. Autocorrelation plot of $\beta_3$ is also indicating high autocorrelation. But all coefficients are normally distributed as indicated in the density plots. Similar simulations were done for other degrees of freedom for which, traceplots are given in appendix A. The reason for household income coefficient not converging might be that, it was not divided by the neighbourhood population before considering in the model while all other variables were scaled based on the neighbourhood population.

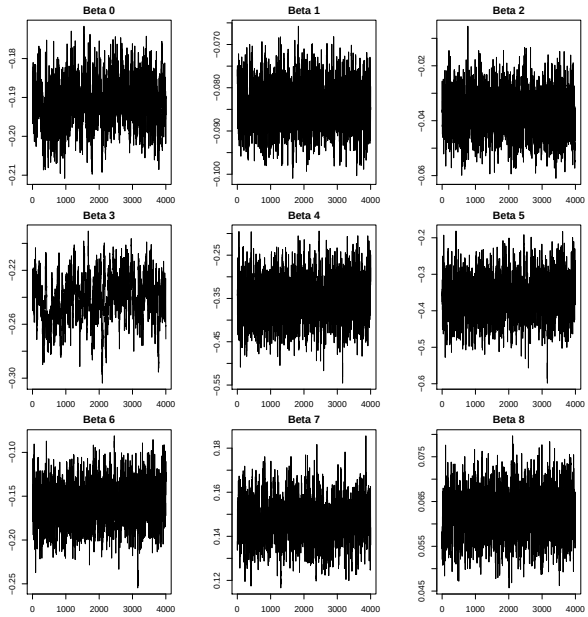


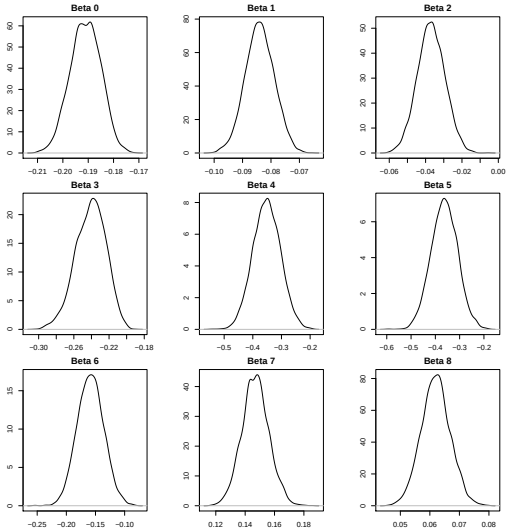

Figure 5: Trace plots of $\beta$ for $t$-link model df = 8

# 5 Model Comparison

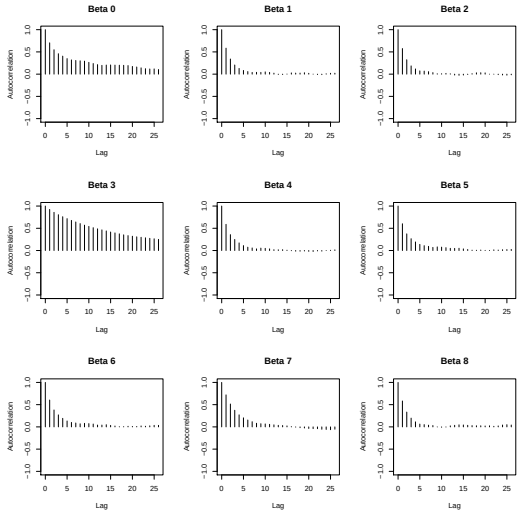
As the first step of model comparison we can denote the mean coefficient values from each simulation model and its standard deviation which is given in table 1. A direct comparison between coefficients in the GLM model and Probit model can be done as both models follow the same linear model. GLM model and probit model give similar mean and standard deviations for all $\beta$ coefficients. $\beta$ coefficients of GLM and $t$-link model cannot be directly compared due to the involvment of the vector $\lambda$ in the $t$-link model. But if we closely observe GLM $\beta$ coefficient with the $\lambda$ multiplier times the $t$-link beta coefficient we see that both models have close coefficients. With the increase of degrees of freedom in the $t$-link, value of $\lambda$ increases by a factor of 2 and $\beta$ coefficients decreases by a factor of 2.

In figure 7a we plot the predicted probabilities from each Bayesian model with the prediction probabilities from the GLM model. We see that probit model as well as $t$-link model for higher degrees of freedom predictions are very similar to the GLM model predictions. To see how much of difference exist between these models, figure 7b shows the density plot of the difference between the probabilities from the compared models. For most bayesian models expect for the model with 2 degrees of freedom, probability differences are in the order the 4th decimal place.

Now we can compare the performance of all these models. I have considered two statistics to compare model performance. In-sample classification error and out of sample classification error. In-sample classification error is computed as the number of observations correctly classified by the fitted model divided by the total number of observations in the dataset used in the model building process. For out-of-sample

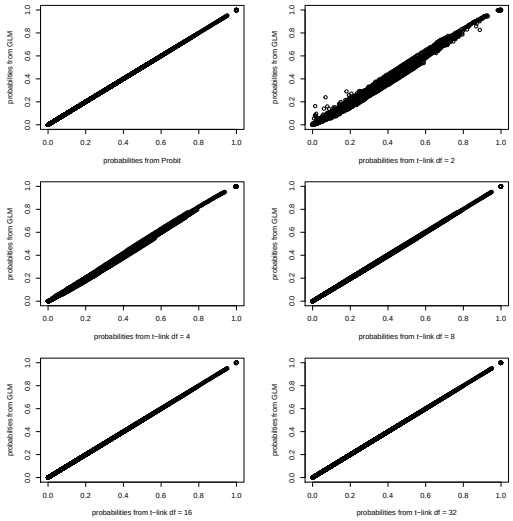



(a) Density plots for simulated $\beta$

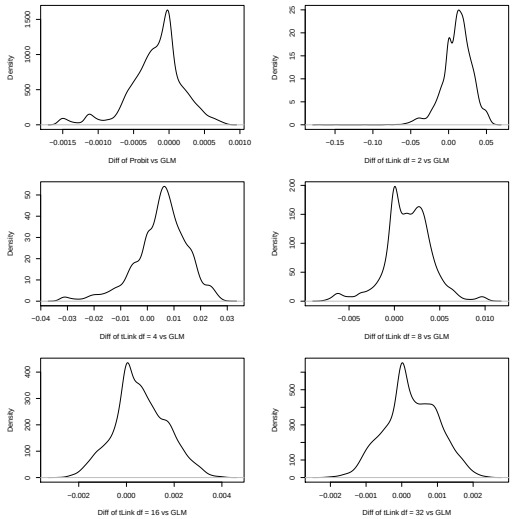



(b) ACF plots for simulated $\beta$

Figure 6: Simulation results for $t$-link with 8 degrees of freedom




(a) Comparison of predicted probabilities




(b) Density of difference in probabilities

Figure 7: Comparing predicted probabilities with GLM model

Table 1: Mean $\beta$ coefficients and standard deviation

| Beta index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| GLM | -0.8172 | -0.3580 | -0.1537 | -1.0136 | -1.5063 | -1.5475 | -0.6740 | 0.6387 | 0.2657 |
| | (0.0250) | (0.0215) | (0.0316) | (0.0801) | (0.1989) | (0.2274) | (0.0938) | (0.0419) | (0.0205) |
| Probit | -0.8181 | -0.3585 | -0.1539 | -1.0133 | -1.4996 | -1.5394 | -0.6717 | 0.6391 | 0.2663 |
| | (0.0256) | (0.0213) | (0.0319) | (0.0817) | (0.1989) | (0.2289) | (0.0949) | (0.0384) | (0.0203) |
| $t$ link df=2 | -0.477 | -0.2127 | -0.1117 | -0.6083 | -0.8491 | -0.8888 | -0.3998 | 0.3287 | 0.1536 |
| | (0.0172) | (0.0161) | (0.0232) | (0.0475) | (0.1449) | (0.1630) | (0.0681) | (0.0267) | (0.0152) |
| $t$ link df=4 | -0.3400 | -0.1506 | -0.0720 | -0.4187 | -0.6147 | -0.6348 | -0.2816 | 0.2472 | 0.1099 |
| | (0.0105) | (0.0095) | (0.0140) | (0.0330) | (0.0885) | (0.1020) | (0.0424) | (0.0159) | (0.0088) |
| $t$ link df=8 | -0.1911 | -0.0839 | -0.0369 | -0.2410 | -0.3516 | -0.3632 | -0.1586 | 0.1470 | 0.0620 |
| | (0.0060) | (0.0051) | (0.0076) | (0.0172) | (0.0480) | (0.0544) | (0.0227) | (0.0092) | (0.0049) |
| $t$ link df=16 | -0.0987 | -0.0435 | -0.0191 | -0.1221 | -0.1811 | -0.1857 | -0.0811 | 0.0769 | 0.0323 |
| | (0.0031) | (0.0026) | (0.0039) | (0.0098) | (0.0244) | (0.0279) | (0.0116) | (0.0047) | (0.0025) |
| $t$ link df=32 | -0.0502 | -0.0220 | -0.0096 | -0.0623 | -0.0926 | -0.0949 | -0.0415 | 0.0392 | 0.0164 |
| | (0.0015) | (0.0014) | (0.0020) | (0.0049) | (0.0124) | (0.0141) | (0.0058) | (0.0024) | (0.0012) |

Table 2: Summary statistics for $\lambda$

| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|---|
| $t$ link df=2 | 1.473 | 1.555 | 1.592 | 1.610 | 1.641 | 4.083 |
| $t$ link df=4 | 2.263 | 2.300 | 2.311 | 2.318 | 2.325 | 2.835 |
| $t$ link df=8 | 4.206 | 4.249 | 4.260 | 4.261 | 4.272 | 4.456 |
| $t$ link df=16 | 8.160 | 8.235 | 8.251 | 8.251 | 8.266 | 8.330 |
| $t$ link df=32 | 16.13 | 16.23 | 16.25 | 16.25 | 16.27 | 16.36 |

classification error, a new dataset, crime categories for the month of November 2021 were used which was not used in the data modeling process. Classification error was computed similar to that of the in-sample classification error. Table 3 gives the classification errors for each model. We see that all models give the same classification error correct to the second decimal point. Hence we can conclude that there is no difference in the classification error based on the model used for modeling crime category.

# 6 Conclusion

In conclusion, we do not see a significance difference between the performance of the Probit model and $t$-link model compared to the GLM model. But when degrees of freedom is very low (df=2) we see that performance suffers slightly which is expected because $t$ distribution deviates from normal distribution as

Table 3: Classification Errors from each model

| | in-sample | out-of sample |
|---|---|---|
| GLM | 0.2214 | 0.2564 |
| Probit | 0.2216 | 0.2564 |
| $t$ link df=2 | 0.2237 | 0.2692 |
| $t$ link df=4 | 0.2230 | 0.2564 |
| $t$ link df=8 | 0.2212 | 0.2564 |
| $t$ link df=16 | 0.2214 | 0.2564 |
| $t$ link df=32 | 0.2216 | 0.2564 |

the degrees of freedom gets smaller. There is significance difference in the model convergence time for the three models with $t$-link taking the longest time. We did not compare the performance time in this study but if it were to be considered then, $t$-link model will not be chosen as the best model. Finally we can say the all three models in this study can be used to model crime categories in Denver, Colorado. For future analysis either household income should be removed from the model or it should be scaled similar to the other response variables in the model.

# References

[1] Denver open data catelog.

[2] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.

# 7 R code

```
crime<-read.csv("G:/My␣Drive/Bayes/Final␣Project/Data/crime.csv", stringsAsFactors = TRUE)
#ACS<-read.csv("G:/My Drive/Bayes/Final Project/Data/american_community_survey_nbrhd_2013_2017.c
ACS<-read.csv("G:/My␣Drive/Bayes/Final␣Project/Data/american_community_survey_nbrhd_2015_2019.cs


crime$FIRST_OCCURRENCE_DATE<-as.POSIXct(crime$FIRST_OCCURRENCE_DATE,format="%m/%d/%Y␣%I:%M:%S␣%p
crime$FIRST_OCCURRENCE_DATE<-as.POSIXct(crime$FIRST_OCCURRENCE_DATE,format="%m/%d/%Y␣%I:%M:%S␣%p
crime$month <- months.POSIXt(crime$FIRST_OCCURRENCE_DATE)
crime$year <- format(crime$FIRST_OCCURRENCE_DATE, format = "%Y")
min(crime$FIRST_OCCURRENCE_DATE, na.rm = TRUE)
max(crime$FIRST_OCCURRENCE_DATE, na.rm = TRUE)
ACS$AVG_HH_INCOME <- as.numeric(ACS$AVG_HH_INCOME)

#summary of different crime categories in the dataset
pdf(paste0("crimecat.pdf"),width=8, height=8)
par(mfrow = c(1,1),mar=c(13,4,1.5,1.5))
plot(crime$OFFENSE_CATEGORY_ID,las=2 )
dev.off()

#extract the crime categories we want
crime.type <- c("arson", "auto-theft", "burglary",
                "larceny","public-disorder","robbery",
                 "theft-from-motor-vehicle")
crime.new <- subset(crime, subset = OFFENSE_CATEGORY_ID %in% crime.type )
rm(crime.type, crime)

#average crimes per month in each neighbourhood
agg.crime.month <- aggregate(crime.new$INCIDENT_ID, by = list(crime.new$NEIGHBORHOOD_ID, crime.n
colnames(agg.crime.month) <- c("NBHD_NAME", "year", "month", 'crime')
agg.crime.month.avg <- aggregate(agg.crime.month$crime, by = list(agg.crime.month$NBHD_NAME), FU
colnames(agg.crime.month.avg) <- c("NBHD_NAME", 'crime.per.month')
plot(density(agg.crime.month.avg$crime.per.month), main = "")

rm(crime.new)

crimedata <- merge(agg.crime.month, ACS, by = 'NBHD_NAME')
crimedata$crime.rate <- crimedata$crime/crimedata$TTL_POPULATION_ALL
#crimedata$crime.rate <- crimedata$crime/crimedata$TTL_HOUSEHOLDS
#summary(crimedata$crime.rate)

#creating crime category
crimedata$crime.cat <- 0
crimedata[which(crimedata$crime.rate>= mean(crimedata$crime.rate)),"crime.cat"] <- 1
summary(as.factor(crimedata$crime.cat))

pdf(paste0("crimemonthratedensity.pdf"),width=8, height=8)
plot(density(crimedata$crime.rate), main = "")
abline(v=0.006087045,col = "red")
dev.off()


pdf(paste0("crimemonthdensiyt2in1.pdf"),width=8, height=4)
par(mfrow = c(1,2))
plot(density(agg.crime.month$crime), main = "Number␣of␣Crimes")
plot(density(crimedata$crime.rate), main = "Crime␣rate")
dev.off()
```

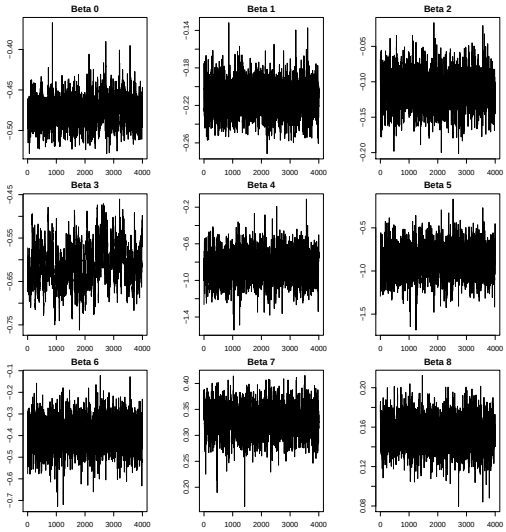
```r
library(stringr)
crimedata$time <- str_c(crimedata$year,"-",crimedata$month,"-01")
crimedata$time <- as.Date(crimedata$time, "%Y-%B-%d")
min(crimedata$time)
crimedata$timenew <- round(difftime(crimedata$time, "2015-12-01", units = "days")/30)
crimedata <- crimedata[order(crimedata$timenew),]
#lets remove the last month november 2021 because it does not have a full year of data
length(which(crimedata$timenew==72)) #75 data points 5532-75=5457
crimedata <- crimedata[-which(crimedata$timenew==72),]
crimedata$constant <- 1
crimedata$NBHD_NO <- as.numeric(crimedata$NBHD_NAME)

#method 1
library(truncnorm)
library(MASS)
crimedat <- crimedata[, c("constant","NBHD_NO", "AVG_HH_INCOME","LESS_THAN_HS_DIPLOMA_EDU","PCT_
crimedata[,155] <- as.numeric(crimedata[,155])
myprobit <- glm(crime.cat ~ NBHD_NO + AVG_HH_INCOME  + LESS_THAN_HS_DIPLOMA_EDU + PCT_HISPANIC +
                data = crimedata)
summary(myprobit)

colnames(X) <- c("constant","NBHD_NO", "AVG_HH_INCOME","LESS_THAN_HS_DIPLOMA_EDU","PCT_HISPANIC"
test.dat <- data.frame(X)
myprobit <- glm(crimedata[,"crime.cat"] ~ NBHD_NO + AVG_HH_INCOME  + LESS_THAN_HS_DIPLOMA_EDU +
                data = test.dat)
summary(myprobit)


X <- unname(data.matrix(crimedat))
X <- scale(X)
X[,1] <- 1
y <- crimedata[,"crime.cat"]

#Gibbs sampling
set.seed(1919)
n <- 10000 #number of iterations
N <- nrow(X) #5395
A <- solve(t(X) %*% X)
beta_mc_P <- matrix(NA,9,n)
Z_mc <- matrix(NA,N,n)
Z <- matrix(NA,N,1)

#initial values
beta <- myprobit$coefficients

#Gibbs sampling
for (i in 1:n) {
  for (j in 1:N) {
    m <- t(X[j,]) %*% beta
    if(y[j] == 1){
      Z_mc[j,i] <- Z[j] <- rtruncnorm(1,a = 0, mean = m, sd = 1)
    }else{
      Z_mc[j,i] <- Z[j] <- rtruncnorm(1,b = 0, mean = m, sd = 1)
    }
  }
  B <- t(X) %*% Z
  mvn_mean <- A %*% B
  beta_mc_P[,i] <- beta <- mvrnorm(1, mvn_mean, A)
```

```
}

rm(Z,Z_mc,A,B)

pdf(paste0("beta_mc_P.pdf"),width=8, height=8)
par(mfrow = c(3,3),mar=c(2,3,2,3))
for (i in 1:9) {
  plot(beta_mc_P[i,5000:10000], type = "l", ylab = "", main= paste0("Beta␣",i-1))
}
dev.off()

pdf(paste0("beta_mc_P_density.pdf"), width=8, height=8)
par(mfrow = c(3,3),mar=c(2,3,2,3))
for (i in 1:9) {
  plot(density(beta_mc_P[i,5000:10000]), ylab = "", main= paste0("Beta␣",i-1))
}
dev.off()

beta_P <- array(NA, dim = c(9,2))
for (i in 1:9) {
  beta_P[i,1] <- mean(beta_mc_P[i,5000:10000])
  beta_P[i,2] <- sd(beta_mc_P[i,5000:10000])
}

#probability of crime for each observation in sample
p_beta_P <- array(NA, dim = c(N,2))
for (i in 1:N) {
  p_beta_P[i,1] <- pnorm(t(X[i,]) %*% beta_P[,1])
}

summary(p_beta_P)

#classification based on predicted probability
p_beta_P[,2] <- 1
p_beta_P[which(p_beta_P[,1]<0.5),2] <- 0
class_errorbeta_P <- length(which(p_beta_P[,2]!=y))/length(y)


#t link method
library(MASS) #for mvn
library(truncnorm)
set.seed(1919)
n <- 5000 #number of iterations
N <- nrow(X) #5457
A <- solve(t(X) %*% X)
beta_mc_2 <- matrix(NA,9,n)
Z_mc <- matrix(NA,N,n)
lambda_mc <- matrix(NA,N,n)
Z <- matrix(NA,N,1)

#df_prob <- matrix(NA,n,2)
#df_prob_sum <- 0
df <- 2
c <- (gamma(df/2)*(df/2)^(df/2))^(-1)
#initial values

beta <- myprobit$coefficients
lambda <- rgamma(N,df/2,2/df)
W <- matrix(0,N,N)
```

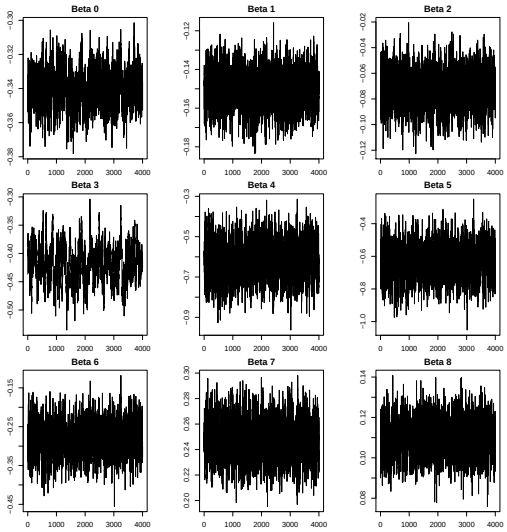
```
for (i in 1:n) {
  for (j in 1:N) {
    m <- t(X[j,]) %*% beta
    if(y[j] == 1){
      Z_mc[j,i] <- Z[j] <- rtruncnorm(1,a = 0, mean = m, sd = sqrt(1/lambda[j]))
    }else{
      Z_mc[j,i] <- Z[j] <- rtruncnorm(1,b = 0, mean = m, sd = sqrt(1/lambda[j]))
    }
  }
  diag(W) <- lambda
  K <- solve(t(X) %*% W %*% X)
  beta_hat <- K %*% t(X) %*% W %*% Z
  beta_mc_2[,i] <- beta <- mvrnorm(1,mu = beta_hat, Sigma = K)
  for (k in 1:N) {
    sig <- df + (Z[k]-t(X[k,]) %*% beta)^2
    lambda_mc[k,i] <- lambda[k] <- rgamma(1,(df+1)/2,2/sig)
  }
if(i%%100==0){
  print(i)
}
}

rm(Z_mc,Z,W,K)


pdf(paste0("beta_mc_2.pdf"),width=8, height=8)
par(mfrow = c(3,3),mar=c(2,3,2,3))
for (i in 1:9) {
  plot(beta_mc_2[i,1000:5000], type = "l", ylab = "", main= paste0("Beta␣",i-1))
}
dev.off()

pdf(paste0("beta_mc_2_density.pdf"),width=8, height=8)
par(mfrow = c(3,3),mar=c(2,3,2,3))
for (i in 1:9) {
  plot(density(beta_mc_2[i,1000:5000]), ylab = "", main= paste0("Beta␣",i-1))
}
dev.off()

beta_2 <- array(NA, dim = c(9,2))
for (i in 1:9) {
  beta_2[i,1] <- mean(beta_mc_2[i,1000:5000])
  beta_2[i,2] <- sd(beta_mc_2[i,1000:5000])
}

lambda_post_2 <- array(NA,dim = c(N,1))
for (i in 1:N) {
  lambda_post_2[i] <- mean(lambda_mc[i,1000:5000])
}

lambda_post_2 <- sqrt(lambda_post_2)

p_beta_2 <- array(NA, dim = c(N,2))
for (i in 1:N) {
  p_beta_2[i,1] <- pnorm(lambda_post_2[i] %*% t(X[i,]) %*% beta_2[,1])
}

summary(p_beta_2)
```

```r
#classification based on predicted probability
p_beta_2[,2] <- 1
p_beta_2[which(p_beta_2[,1]<0.5),2] <- 0
class_errorbeta_2 <- length(which(p_beta_2[,2]!=y))/length(y)

summary(p_beta_2)
summary(p_beta_P)
plot(p_beta_P[,1],p_beta_2[,1], xlab = "p from Probit", ylab = "p from tLink")

### model comparison

nov.crime <- read.csv("G:/My Drive/Bayes/Final Project/Data/November crimes.csv", stringsAsFacto
nov.crime$FIRST_OCCURRENCE_DATE<-as.POSIXct(nov.crime$FIRST_OCCURRENCE_DATE,format="%m/%d/%Y %I:
nov.crime$FIRST_OCCURRENCE_DATE<-as.POSIXct(nov.crime$FIRST_OCCURRENCE_DATE,format="%m/%d/%Y %I:
nov.crime$month <- months.POSIXt(nov.crime$FIRST_OCCURRENCE_DATE)
nov.crime$year <- format(nov.crime$FIRST_OCCURRENCE_DATE, format = "%Y")
crime.type <- c("arson", "auto-theft", "burglary",
                "larceny","public-disorder","robbery",
                "theft-from-motor-vehicle")
nov.crime <- subset(nov.crime, subset = OFFENSE_CATEGORY_ID %in% crime.type )

nov.agg.crime.month <- aggregate(nov.crime$INCIDENT_ID, by = list(nov.crime$NEIGHBORHOOD_ID), FU
colnames(nov.agg.crime.month) <- c("NBHD_NAME", 'crime')

nov.crimedata <- merge(nov.agg.crime.month, ACS, by = 'NBHD_NAME')
nov.crimedata$crime.rate <- nov.crimedata$crime/nov.crimedata$TTL_POPULATION_ALL

nov.crimedata$crime.cat <- 0
nov.crimedata[which(nov.crimedata$crime.rate>= 0.0064475),"crime.cat"] <- 1
summary(as.factor(nov.crimedata$crime.cat))


library(stringr)
nov.crimedata$timenew <- 72
nov.crimedata$constant <- 1
nov.crimedata$NBHD_NO <- as.numeric(nov.crimedata$NBHD_NAME)

nov.crimedat <- nov.crimedata[, c("constant","NBHD_NO", "AVG_HH_INCOME","LESS_THAN_HS_DIPLOMA_ED
X.pred <- unname(data.matrix(nov.crimedat))
X.old <- unname(data.matrix(crimedat))
x.stat <- array(NA, dim = c(ncol(X.old),2))
for (i in 1:ncol(X.old)) {
  x.stat[i,1] <- mean(X.old[,i])
  x.stat[i,2] <- sd(X.old[,i])
}
X.pred <- scale(X.pred, center = x.stat[,1], scale = x.stat[,2] )
X.pred[,1] <- 1
y.pred <- nov.crimedata[,"crime.cat"]


p.pred <- array(NA, dim = c(length(y.pred),6))
for (i in 1:length(y.pred)) {
  p.pred[i,1] <- pnorm(t(X.pred[i,]) %*% beta_P[,1])
}

for (i in 1:length(y.pred)) {
  p.pred[i,2] <- pnorm(lambda_post_2[i] %*% t(X.pred[i,]) %*% beta_2[,1])
  p.pred[i,3] <- pnorm(lambda_post_4[i] %*% t(X.pred[i,]) %*% beta_4[,1])
```

```
  p.pred[i,4] <- pnorm(lambda_post_8[i] %*% t(X.pred[i,]) %*% beta_8[,1])
  p.pred[i,5] <- pnorm(lambda_post_16[i] %*% t(X.pred[i,]) %*% beta_16[,1])
  p.pred[i,6] <- pnorm(lambda_post_32[i] %*% t(X.pred[i,]) %*% beta_32[,1])
}

cat.pred <- array(1, dim = c(length(y.pred),6))
for (k in 1:6) {
  cat.pred[which(p.pred[,k]<0.5),k] <- 0
}

class.error <- array(NA, dim = c(1,6))
for (k in 1:6) {
  class.error[1,k] <- length(which(cat.pred[,k]!=y.pred))/length(y.pred)
}

### classification errors for glm
pred.insampl <- predict(myprobit,test.dat[-1] , type = "response")
sample.pred <- array(1,dim = length(y))
sample.pred[which(pred.insampl<0.5)] <- 0
class.error.glm.insample <- length(which(sample.pred!=y))/length(y)

colnames(X.pred) <- c("constant","NBHD_NO", "AVG_HH_INCOME","LESS_THAN_HS_DIPLOMA_EDU","PCT_HISP
test.dat.pred <- data.frame(X.pred)
pred.out <- predict(myprobit,test.dat.pred[,-1] , type = "response")
sample.pred <- array(1,dim = length(y.pred))
sample.pred[which(pred.out<0.5)] <- 0
class.error.glm.outsample <- length(which(sample.pred!=y.pred))/length(y.pred)


### comparison plots
pdf(paste0("probability␣comparison.pdf"),width=8, height=8)
par(mfrow = c(3,2),mar=c(4,5,2,2))
plot(p_beta_P[,1],pred.insampl, ylab = "probabilties␣from␣GLM", xlab = "probabilities␣from␣Probi
plot(p_beta_2[,1],pred.insampl, ylab = "probabilties␣from␣GLM", xlab = "probabilities␣from␣t-lin
plot(p_beta_4[,1],pred.insampl, ylab = "probabilties␣from␣GLM", xlab = "probabilities␣from␣t-lin
plot(p_beta_8[,1],pred.insampl, ylab = "probabilties␣from␣GLM", xlab = "probabilities␣from␣t-lin
plot(p_beta_16[,1],pred.insampl,ylab = "probabilties␣from␣GLM", xlab = "probabilities␣from␣t-lin
plot(p_beta_32[,1],pred.insampl,ylab = "probabilties␣from␣GLM", xlab = "probabilities␣from␣t-lin
dev.off()

pdf(paste0("probability␣density␣comparison.pdf"),width=8, height=8)
par(mfrow = c(3,2),mar=c(4,5,2,2))
diff.prob <- p_beta_P[,1] - pred.insampl
plot(density(diff.prob), xlab = "Diff␣of␣Probit␣vs␣GLM",main = "")
diff.prob <- p_beta_2[,1] - pred.insampl
plot(density(diff.prob),xlab = "Diff␣of␣tLink␣df␣=␣2␣vs␣GLM",main = "")
diff.prob <- p_beta_4[,1] - pred.insampl
plot(density(diff.prob),xlab = "Diff␣of␣tLink␣df␣=␣4␣vs␣GLM",main = "")
diff.prob <- p_beta_8[,1] - pred.insampl
plot(density(diff.prob),xlab = "Diff␣of␣tLink␣df␣=␣8␣vs␣GLM",main = "")
diff.prob <- p_beta_16[,1] - pred.insampl
plot(density(diff.prob),xlab = "Diff␣of␣tLink␣df␣=␣16␣vs␣GLM",main = "")
diff.prob <- p_beta_32[,1] - pred.insampl
plot(density(diff.prob),xlab = "Diff␣of␣tLink␣df␣=␣32␣vs␣GLM",main = "")
dev.off()



diff.prob <- p_beta_P[,1]-pred.insampl
```




(a) Trace plots of $\beta$ for tLink with df = 2

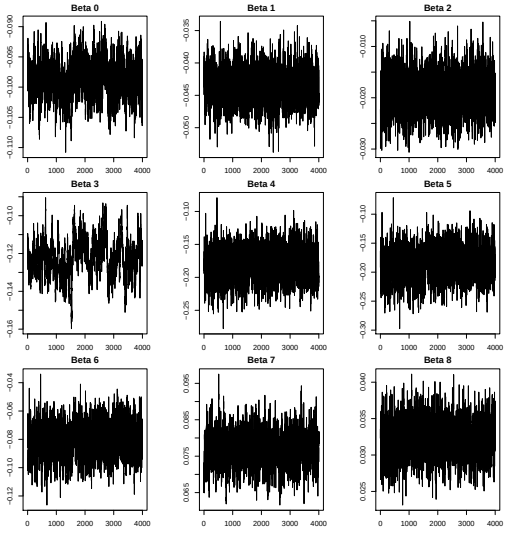



(b) Trace plots of $\beta$ for tLink with df = 4

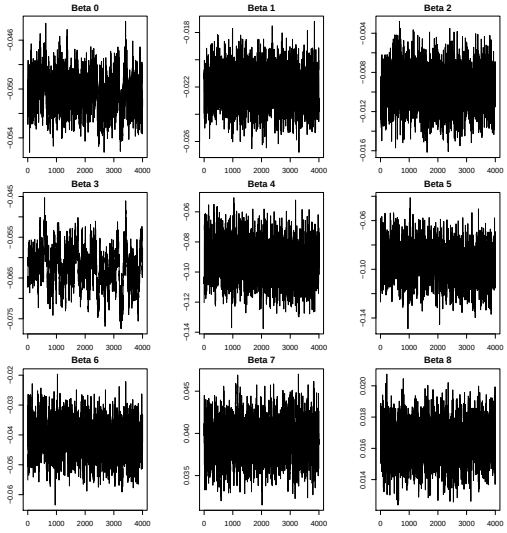
Figure 8: Simulation results for tLink

```
plot(density(diff.prob))

#model convergence
### Convergence
library(coda)

rownames(beta_mc_P) <- c("Beta␣0", "Beta␣1", "Beta␣2", "Beta␣3", "Beta␣4", "Beta␣5", "Beta␣6", "
beta_mc_P_draws <- mcmc(t(beta_mc_P)[5000:10000,])
pdf(paste0("beta_mc_P_ACF.pdf"),width=8, height=8)
autocorr.plot(beta_mc_P_draws)
dev.off()

rownames(beta_mc_8) <- c("Beta␣0", "Beta␣1", "Beta␣2", "Beta␣3", "Beta␣4", "Beta␣5", "Beta␣6", "
beta_mc_8_draws <- mcmc(t(beta_mc_8)[1000:5000,])
pdf(paste0("beta_mc_8_ACF.pdf"),width=8, height=8)
autocorr.plot(beta_mc_8_draws)
dev.off()

summary(myprobit)
summary(beta_mc_P_draws)

rownames(beta_mc_32) <- c("Beta␣0", "Beta␣1", "Beta␣2", "Beta␣3", "Beta␣4", "Beta␣5", "Beta␣6",
beta_mc_32_draws <- mcmc(t(beta_mc_32)[1000:5000,])
pdf(paste0("beta_mc_32_ACF.pdf"),width=8, height=8)
autocorr.plot(beta_mc_32_draws)
dev.off()
summary(beta_mc_32_draws)

###comparing lambda values
summary(lambda_post_32[,1])
```

# 8 Appendix A




(a) Trace plots of $\beta$ for tLink with df = 16




(b) Trace plots of $\beta$ for tLink with df = 32

Figure 9: Simulation results for tLink