BDA - Project Work

Jacopo Losi, Nicola Saljoughi

Contents

Introduction

This project is based on a study carried out in 2015 by a group of researchers to estimate the incidence of serious suicide attempts in Shandong, China, and to examine the factors associated with fatality among the attempters.
We have chosen to examine a dataset on suicides because it is a really important but often underconsidered problem in today's society. Not only this problem reflects a larger problem in a country societal system but it can also be a burden for hospital resources. We think that by being able to talk about it more openly and by truly trying to estimate its size and impact we can start to understand where the causes are rooted and what can be done to fight it.
We invite the reader to check the source section to further read about the setting and results of the named paper.

Analysis Problem

The objective of the project is to use the bayesian approach to develop models to evaluate the most influential factors related to serious suicide attempts (SSAs, defined as suicide attempts resulting in either death or hospitalisation) and being able to make predictions for the years following the period where the study was set.

Data

Data from two independent health surveillance systems were linked, constituted by records of suicide deaths and hospitalisations that occured among residents in selected countries during 2009-2011.
The data set is constituted by 2571 observations of 11 variables:

- `Person_ID`: ID number, $1, ..., 2571$
- `Hospitalised`: *yes* or *no*
- `Died`: *yes* or *no*
- `Urban`: *yes*, *no* or *unknown*
- `Year`: 2009, 2010 or 2011
- `Month`: $1, ..., 12$
- `Sex`: *female* or *male*
- `Age`: years
- `Education`: *iliterate*, *primary*, *Secondary*, *Tertiary* or *unknown*
- `Occupation`: one of ten categories
- `method`: one of nine methods

It is important to notice that the population in the study is predominantly rura and that the limitation of the study is that the incidence estimates are likely to be underestimated due to underreporting in both surveillance systems.

Source

Sun J, Guo X, Zhang J, Wang M, Jia C, Xu A (2015) “Incidence and fatality of serious suicide attempts in a predominantly rural population in Shandong, China: a public health surveillance study,” BMJ Open 5(2): e006762. https://doi.org/10.1136/bmjopen-2014-006762

Data downloaded via Dryad Digital Repository. https://doi.org/10.5061/dryad.r0v35

Analysis

In this section we will carry out our analysis following the bayesian approach, first developing two different models to analyse the data, assessing their convergence, doing posterior predictive checking, comparing them to choose the one that performs best and eventually use the obtained model to select the most influencial factors and answering our analysis problem.

```r
random_index <- sample(mydata$Person_ID, size = 50, replace = TRUE)

data_reduced <- mydata[random_index, ]
data_reduced <- na.omit(data_reduced)
```

Model description

In order to evaluate the factors which influence the probability of SSA the most it is an obvious chioice to develop a multiple logistic regression model. Two different models have been implemented which will then be compared in the following analysis:

- **simple logistic regression model** with uniform priors and with no distinction between years and
- **hierarchical logistic regression model** where we divide our data into three groups (one for each year) and then develop our model defining priors in a hierarchical manner.

Simple Logistic Regression Model

Hierarchical Logistic Regression Model

Prior choices

Stan Code

Here we implement our models using `Stan`.

```r
## Create Stan data
dat <- list(N        = nrow(mydata),
            p        = ncol(mydata) - 2,
            died     = as.numeric(mydata$Died),
            urban    = as.numeric(mydata$Urban),
            year     = as.numeric(mydata$Year),
            season   = as.numeric(mydata$Season),
            sex      = as.numeric(mydata$Sex),
            age      = as.numeric(mydata$Age),
            edu      = as.numeric(mydata$Education),
            job      = as.numeric(mydata$Occupation),
            method   = as.numeric(mydata$method))
```

Then in this phase, in which we are working on testing different models, it is worth to take only some random samples from the data. As a matter of fact, the dataset that we have is big and thus the computation on the whole dataset will take a lot of time.

Therefore, we will proceed as follows: * we will generate a vector of 50 random number taken from our dataset; * we will test the models with this data, that are sufficient for not loosing in generality; * we will run the final model on the whole dataset.

```r
## Create Stan data
dat_red <- list(N    = nrow(data_reduced),
                p        = ncol(data_reduced) - 2,
                died     = as.numeric(data_reduced$Died),
                urban    = as.numeric(data_reduced$Urban),
                year     = as.numeric(data_reduced$Year),
                season   = as.numeric(data_reduced$Season),
                sex      = as.numeric(data_reduced$Sex),
                age      = as.numeric(data_reduced$Age),
                edu      = as.numeric(data_reduced$Education),
                job      = as.numeric(data_reduced$Occupation),
                method   = as.numeric(data_reduced$method))

## Load Stan file
fileName <- "./logistic_regression_model.stan"
stan_code <- readChar(fileName, file.info(fileName)$size)
cat(stan_code)
```

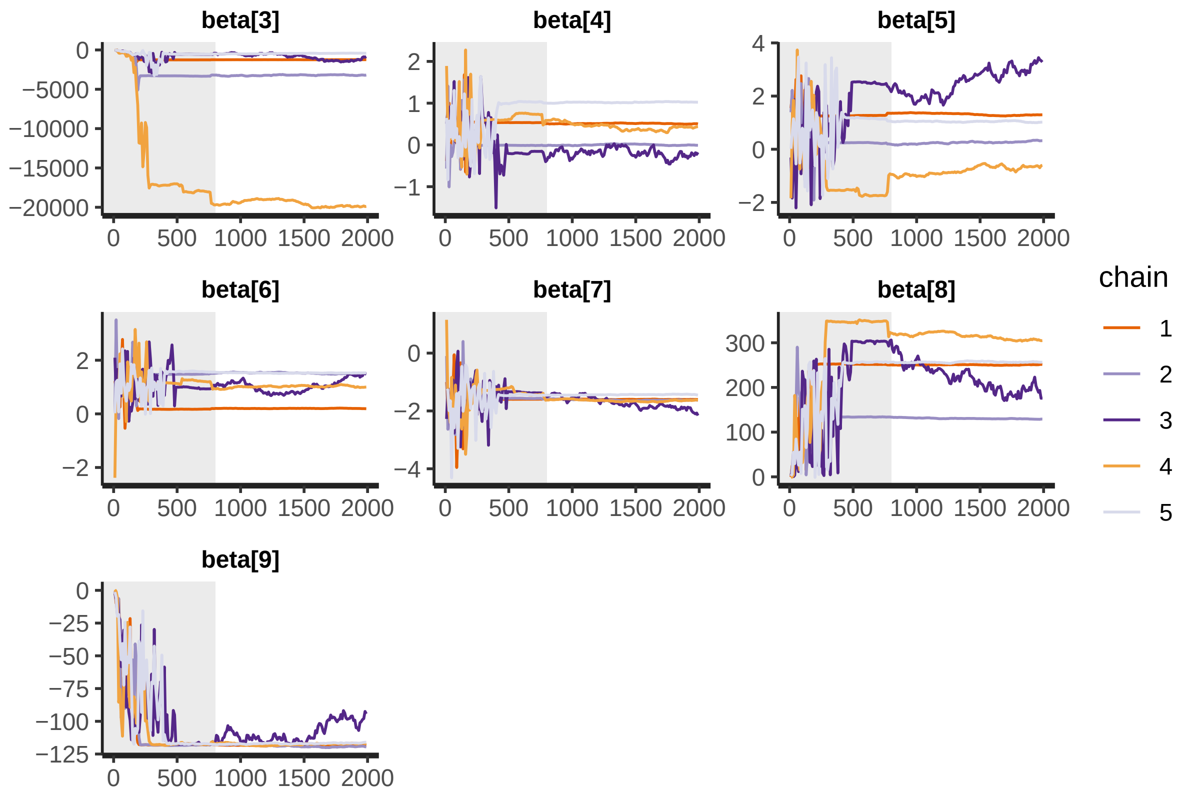
```
## data {
##   // Define variables in data
##   // Number of observations (an integer)
##   int<lower=0> N;
##
##   // Number of parameters
##   int<lower=0> p;
##
##   // Variables
##   int  died[N];
##   int<lower=0>  year[N];
##   int<lower=0>  urban[N];
##   int<lower=0>  season[N];
##   int<lower=0>  sex[N];
##   int<lower=0>  age[N];
##   int<lower=0>  edu[N];
##   int<lower=0>  job[N];
##   int<lower=0>  method[N];
## }
##
## parameters {
##   // Define parameters to estimate
##   real beta[p];
## }
##
## transformed parameters  {
##   // Probability trasformation from linear predictor
##   real<lower=0> odds[N];
##   real<lower=0, upper=1> prob[N];
##   for (i in 1:N) {
##     odds[i] = exp(beta[1] + beta[2]*year[i]   + beta[3]*urban[i] +
##                             beta[4]*season[i] + beta[5]*sex[i]   +
##                             beta[6]*age[i]    + beta[7]*edu[i]   +
##                             beta[8]*job[i]    + beta[9]*method[i] );
##     prob[i] = odds[i] / (odds[i] + 1);
##   }
## }
##
## model {
##   // Prior part of Bayesian inference (flat if unspecified)
##
##   // Likelihood part of Bayesian inference
##     died ~ bernoulli(prob);
## }
```

Data processing

```
## Warning: There were 599 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 4.87, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess

## Inference for Stan model: 44efd1e4898e49d7c3da763fa46eaad0.
## 5 chains, each with iter=2000; warmup=800; thin=10;
## post-warmup draws per chain=120, total post-warmup draws=600.
##
##              mean se_mean      sd      2.5%      25%      50%     75%
## beta[1]    227.35   38.84   62.75    148.71   195.16   213.05  230.61
## beta[2]     -0.38    0.29    0.52     -1.49    -0.66    -0.60    0.01
## beta[3] -5053.75 4604.02 7299.14 -19971.53 -3258.26 -1242.56 -519.84
## beta[4]      0.36    0.27    0.43     -0.30    -0.01     0.43    0.52
## beta[5]      0.87    0.70    1.14     -1.00     0.22     1.04    1.36
## beta[6]      1.06    0.30    0.50      0.20     0.83     1.07    1.52
## beta[7]     -1.60    0.05    0.13     -1.92    -1.64    -1.61   -1.55
## beta[8]    234.63   37.85   61.91    129.68   195.70   250.72  258.15
## beta[9]   -116.05    2.25    5.11   -119.57  -118.41  -117.55 -116.65
##            97.5% n_eff  Rhat
## beta[1]   345.55     3  6.17
## beta[2]     0.40     3  3.82
## beta[3]  -410.70     3 50.95
## beta[4]     1.03     3  8.61
## beta[5]     3.14     3  9.65
## beta[6]     1.55     3  5.28
## beta[7]    -1.42     6  2.54
## beta[8]   323.96     3  7.85
## beta[9]  -98.05     5  1.91
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 05 23:31:56 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

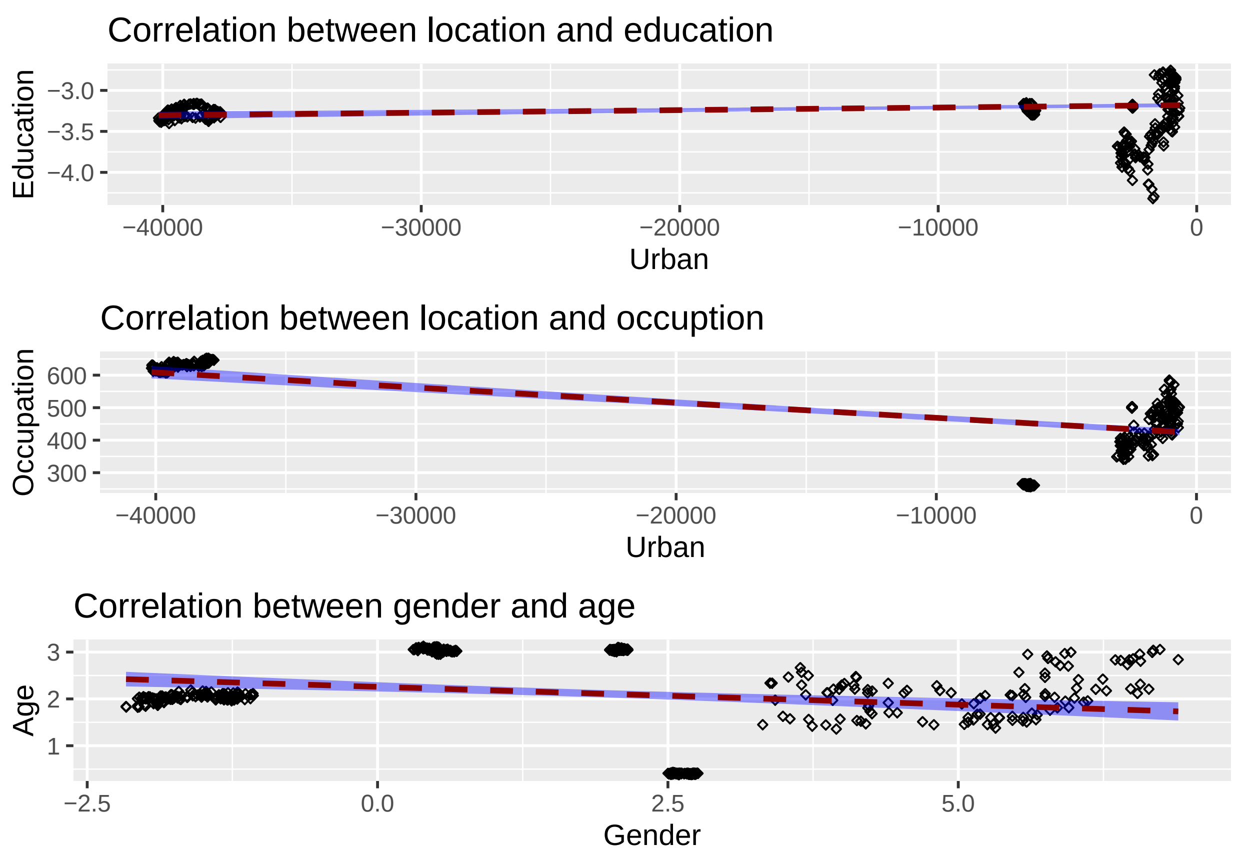
```
# Transform fitting over beta in a dataframe for the plots

beta_matrix <- zeros(length(extract(resStan)$beta[,1]), ncol(data_reduced) - 2)

for (i in 1:ncol(data_reduced) - 2)
  beta_matrix[,i] = beta_matrix[,i] + extract(resStan)$beta[,i]

beta_df <- as.data.frame(beta_matrix)
```

```
# Show traceplot
traceplot(resStan, pars = c('beta[3]','beta[4]', 'beta[5]',
                            'beta[6]', 'beta[7]', 'beta[8]',
                            'beta[9]'), inc_warmup = TRUE)
```

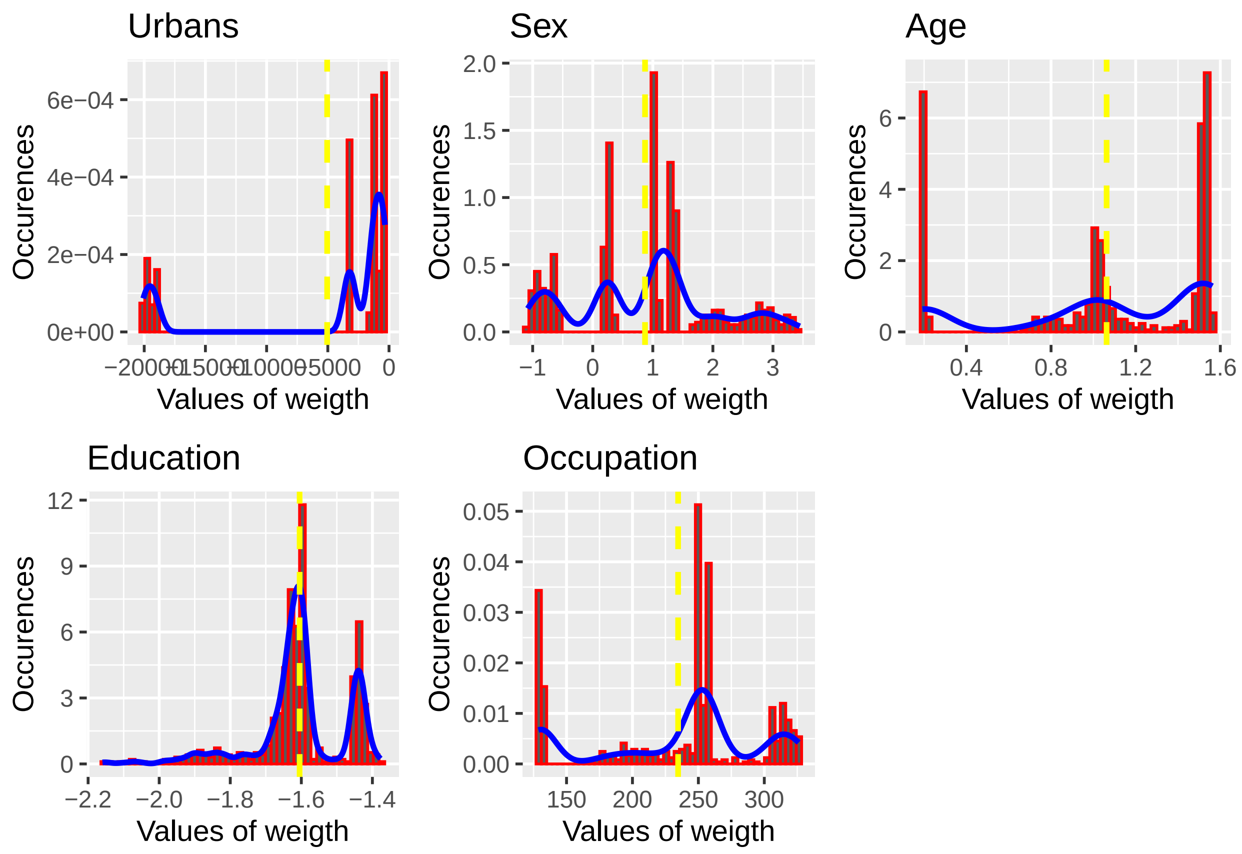


```
# Generate some scatter plots in order to see the correlations between parameters
scatter_1 <- ggplot(beta_df, aes(x=V3, y=V7)) +
                    ggtitle("Correlation between location and education") +
                    xlab("Urban") + ylab("Education") +
```

```r
             geom_point(size=1, shape=23) +
             geom_smooth(method=lm, linetype="dashed", color="darkred", fill="blue")

scatter_2 <- ggplot(beta_df, aes(x=V3, y=V8)) +
                    ggtitle("Correlation between location and occuption") +
                    xlab("Urban") + ylab("Occupation") +
             geom_point(size=1, shape=23) +
             geom_smooth(method=lm, linetype="dashed", color="darkred", fill="blue")

scatter_3 <- ggplot(beta_df, aes(x=V5, y=V6)) +
                    ggtitle("Correlation between gender and age") +
                    xlab("Gender") + ylab("Age") +
             geom_point(size=1, shape=23) +
             geom_smooth(method=lm, linetype="dashed", color="darkred", fill="blue")


ggplot2.multiplot(scatter_1,scatter_2,scatter_3, cols=1)
```



```r
# overlay histogram, density and show the mean value
# The plots of the most interesting parameters are presented.
# Using the mean value it could be interesting to understand
# which weaknly informative priors can be designed

plot_1 <- qplot(extract(resStan)$beta[,3], geom = 'blank', xlab = 'Values of weigth', ylab = 'Occurences
  geom_histogram(aes(y = ..density..),col = I('red'), bins = 50) +
  geom_line(aes(y = ..density..), size = 1, col = I('blue'), stat = 'density', ) +
  geom_vline(aes(xintercept=mean(extract(resStan)$beta[,3])), col=I('yellow'), linetype="dashed", size=

plot_2 <- qplot(extract(resStan)$beta[,5], geom = 'blank', xlab = 'Values of weigth', ylab = 'Occurences
  geom_histogram(aes(y = ..density..),col = I('red'), bins = 50) +
  geom_line(aes(y = ..density..), size = 1, col = I('blue'), stat = 'density', ) +
  geom_vline(aes(xintercept=mean(extract(resStan)$beta[,5])), col=I('yellow'), linetype="dashed", size=

plot_3 <- qplot(extract(resStan)$beta[,6], geom = 'blank', xlab = 'Values of weigth', ylab = 'Occurence
  geom_histogram(aes(y = ..density..),col = I('red'), bins = 50) +
  geom_line(aes(y = ..density..), size = 1, col = I('blue'), stat = 'density', ) +
  geom_vline(aes(xintercept=mean(extract(resStan)$beta[,6])), col=I('yellow'), linetype="dashed", size=

plot_4 <- qplot(extract(resStan)$beta[,7], geom = 'blank', xlab = 'Values of weigth', ylab = 'Occurence
  geom_histogram(aes(y = ..density..),col = I('red'), bins = 50) +
  geom_line(aes(y = ..density..), size = 1, col = I('blue'), stat = 'density', ) +
  geom_vline(aes(xintercept=mean(extract(resStan)$beta[,7])), col=I('yellow'), linetype="dashed", size=

plot_5 <- qplot(extract(resStan)$beta[,8], geom = 'blank', xlab = 'Values of weigth', ylab = 'Occurences
  geom_histogram(aes(y = ..density..),col = I('red'), bins = 50) +
  geom_line(aes(y = ..density..), size = 1, col = I('blue'), stat = 'density', ) +
  geom_vline(aes(xintercept=mean(extract(resStan)$beta[,8])), col=I('yellow'), linetype="dashed", size=

ggplot2.multiplot(plot_1,plot_2,plot_3,plot_4, plot_5, cols=3)
```



From the analysis done above, and especially looking at the histogram, it is clear that the most important parameters that count in our analysis are: the fact that the people come from urban or rural areas, then their education, occupation and partially if they are man or woman. As a matter of fact, the mean and the maximum values of the coeffcient related to those paramters have the bigger magnitude. This means that those parameters are weighted more in the multi regression function in the model.

Therefore, for further analysis, it will be good to develop specific analysis using only these parameters, in order to have a more precise evalution considering only the most relevant parameters.

Frequentist approach

```r
outcomeModel <- glm(as.numeric(Died) ~ as.numeric(Urban) +
                                       as.numeric(Year) +
                                       as.numeric(Season) +
                                       as.numeric(Sex) +
                                       as.numeric(Age) +
                                       as.numeric(Education) +
                                       as.numeric(Occupation) +
                                       as.numeric(method), data = mydata,
                    family = binomial(link = "logit"))
summary(outcomeModel)
```

```
##
## Call:
## glm(formula = as.numeric(Died) ~ as.numeric(Urban) + as.numeric(Year) +
##     as.numeric(Season) + as.numeric(Sex) + as.numeric(Age) +
##     as.numeric(Education) + as.numeric(Occupation) + as.numeric(method),
##     family = binomial(link = "logit"), data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3828  -0.8351   0.3501   0.8233   2.5409
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -0.239151   0.325168  -0.735   0.4621
## as.numeric(Urban)       0.296072   0.161150   1.837   0.0662 .
## as.numeric(Year)        0.291617   0.061971   4.706 2.53e-06 ***
## as.numeric(Season)      0.008641   0.044876   0.193   0.8473
## as.numeric(Sex)         0.424288   0.099000   4.286 1.82e-05 ***
## as.numeric(Age)         0.331736   0.052953   6.265 3.73e-10 ***
## as.numeric(Education)  -1.248306   0.080832 -15.443  < 2e-16 ***
## as.numeric(Occupation)  0.518808   0.131602   3.942 8.07e-05 ***
## as.numeric(method)     -0.051068   0.045090  -1.133   0.2574
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3344.4  on 2413  degrees of freedom
## Residual deviance: 2560.3  on 2405  degrees of freedom
## AIC: 2578.3
##
## Number of Fisher Scoring iterations: 4
```

Comparison

```r
## Bayesian
print(resStan, pars = c("beta"))
```

```
## Inference for Stan model: 44efd1e4898e49d7c3da763fa46eaad0.
## 5 chains, each with iter=2000; warmup=800; thin=10;
## post-warmup draws per chain=120, total post-warmup draws=600.
##
##             mean se_mean      sd      2.5%      25%      50%     75%
## beta[1]   227.35   38.84   62.75    148.71   195.16   213.05  230.61
## beta[2]    -0.38    0.29    0.52     -1.49    -0.66    -0.60    0.01
## beta[3] -5053.75 4604.02 7299.14 -19971.53 -3258.26 -1242.56 -519.84
## beta[4]     0.36    0.27    0.43     -0.30    -0.01     0.43    0.52
## beta[5]     0.87    0.70    1.14     -1.00     0.22     1.04    1.36
## beta[6]     1.06    0.30    0.50      0.20     0.83     1.07    1.52
## beta[7]    -1.60    0.05    0.13     -1.92    -1.64    -1.61   -1.55
## beta[8]   234.63   37.85   61.91    129.68   195.70   250.72  258.15
## beta[9]  -116.05    2.25    5.11   -119.57  -118.41  -117.55 -116.65
##           97.5% n_eff  Rhat
## beta[1]  345.55     3  6.17
## beta[2]    0.40     3  3.82
## beta[3] -410.70     3 50.95
## beta[4]    1.03     3  8.61
## beta[5]    3.14     3  9.65
## beta[6]    1.55     3  5.28
## beta[7]   -1.42     6  2.54
## beta[8]  323.96     3  7.85
## beta[9]  -98.05     5  1.91
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 05 23:31:56 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```r
## Frequentist
tableone::ShowRegTable(outcomeModel, exp = FALSE)
```

```
##                          coef  [confint]       p
## (Intercept)              -0.24 [-0.88, 0.40]   0.462
## as.numeric(Urban)         0.30 [-0.02, 0.61]   0.066
## as.numeric(Year)          0.29 [0.17, 0.41]   <0.001
## as.numeric(Season)        0.01 [-0.08, 0.10]   0.847
## as.numeric(Sex)           0.42 [0.23, 0.62]   <0.001
## as.numeric(Age)           0.33 [0.23, 0.44]   <0.001
## as.numeric(Education)    -1.25 [-1.41, -1.09] <0.001
## as.numeric(Occupation)    0.52 [0.26, 0.78]   <0.001
## as.numeric(method)       -0.05 [-0.14, 0.04]   0.257
```

Same clustering on the data

Let us try to cluster the data using the specific year in order to do a prediction on the following year

```r
indexYear2009 <- which(mydata$Year == 2009)
data_year_2009 <- mydata[indexYear2009,]

indexYear2010 <- which(mydata$Year == 2010)
data_year_2010 <- mydata[indexYear2010,]

indexYear2011 <- which(mydata$Year == 2011)
data_year_2011 <- mydata[indexYear2011,]
```

Simple model

```r
## SIMPLE LOGISTIC REGRESSION MODEL

## Load Stan Model
fileNameOne <- "./logistic_regression_model.stan"
stan_code_simple <- readChar(fileNameOne, file.info(fileNameOne)$size)
cat(stan_code_simple)
```

```
## data {
##   // Define variables in data
##   // Number of observations (an integer)
##   int<lower=0> N;
##
##   // Number of parameters
##   int<lower=0> p;
##
##   // Variables
##   int  died[N];
##   int<lower=0>  year[N];
##   int<lower=0>  urban[N];
##   int<lower=0>  season[N];
##   int<lower=0>  sex[N];
##   int<lower=0>  age[N];
##   int<lower=0>  edu[N];
##   int<lower=0>  job[N];
##   int<lower=0>  method[N];
## }
##
## parameters {
##   // Define parameters to estimate
##   real beta[p];
## }
##
## transformed parameters  {
##   // Probability trasformation from linear predictor
##   real<lower=0> odds[N];
##   real<lower=0, upper=1> prob[N];
##   for (i in 1:N) {
##     odds[i] = exp(beta[1] + beta[2]*year[i]  + beta[3]*urban[i] +
##                             beta[4]*season[i] + beta[5]*sex[i]  +
##                             beta[6]*age[i]    + beta[7]*edu[i]  +
##                              beta[8]*job[i]    + beta[9]*method[i] );
##     prob[i] = odds[i] / (odds[i] + 1);
##   }
## }
##
## model {
##   // Prior part of Bayesian inference (flat if unspecified)
##
##   // Likelihood part of Bayesian inference
##     died ~ bernoulli(prob);
## }
```

Hierarchical model

```r
## HIERARCHICAL LOGISTIC REGRESSION MODEL

## Load Stan Model
fileNameTwo <- "./logistic_regression_model.stan"
stan_code_hier <- readChar(fileNameTwo, file.info(fileNameTwo)$size)
cat(stan_code_hier)
```

```
## data {
##   // Define variables in data
##   // Number of observations (an integer)
##   int<lower=0> N;
##
##   // Number of parameters
##   int<lower=0> p;
##
##   // Variables
##   int  died[N];
##   int<lower=0>  year[N];
##   int<lower=0>  urban[N];
##   int<lower=0>  season[N];
##   int<lower=0>  sex[N];
##   int<lower=0>  age[N];
##   int<lower=0>  edu[N];
##   int<lower=0>  job[N];
##   int<lower=0>  method[N];
## }
##
## parameters {
##   // Define parameters to estimate
##   real beta[p];
## }
##
## transformed parameters  {
##   // Probability trasformation from linear predictor
##   real<lower=0> odds[N];
##   real<lower=0, upper=1> prob[N];
##   for (i in 1:N) {
##     odds[i] = exp(beta[1] + beta[2]*year[i]  + beta[3]*urban[i] +
##                             beta[4]*season[i] + beta[5]*sex[i]  +
##                             beta[6]*age[i]    + beta[7]*edu[i]  +
##                             beta[8]*job[i]    + beta[9]*method[i] );
##     prob[i] = odds[i] / (odds[i] + 1);
##   }
## }
##
## model {
##   // Prior part of Bayesian inference (flat if unspecified)
##
##   // Likelihood part of Bayesian inference
##     died ~ bernoulli(prob);
## }
```

Convergence Analysis

Posterior Predictive Checking

Model Comparison

Sensitivity Analysis

Conclusions

Problems encountered

Potential improvements