SCHOLASTIC SWEDISH ARSON

Socioeconomic Predictors of Swedish Arson in Schools

Desmond Cole • drcole@umich.edu
Andrei Kopelevich • andreisk@umich.edu
Mark Kurzeja • mtkurzej@umich.edu
Teerth Patel • pateltj@umich.edu

Professor Yang Chen - Stats 551
april 24th, 2018

Abstract

The nation of Sweden has an unusual social problem. On average, between one and two school fires occur every day somewhere in the country, usually the product of arson. Drawing from a rich dataset of the frequency of Swedish school fires over time by municipality, alongside a large collection of economic and demographic variables, we analyze predictors of the incidence of man-made fires across 290 municipalities, between the years 1998 and 2014. Our models allow us to predict the number of fires set to occur in a year given municipal characteristics as well as allowing us to make inferences on the correlates of fire incidence. We find that weaker local economic conditions seem to be linked to these fires as well as *(surprisingly)* suburban, rather than urban or rural, settings.

Group Member	Task Description
Desmond	Data Scrubbing, Translations, and Preliminary Data Analysis
Andrei	Preliminary PCA Analysis, Visualizations, & Document Preparation
Mark	Stan Modeling, Modeling Iterations, and Posterior Visual Analysis
Teerth	Spatial Visualizations & Document Preparation

CONTENTS

1 INTRODUCTION

1.1 *Data Description*

The data is obtained through Kaggle, where it compiled from the Swedish Civil Contingencies Agency. The original dataset is quite large (121 MB across four zipped files), including such minutae as books on loan from municipal libraries weighted by share of population. To reduce computational burden while maintaining interpretability, we manually selected 25 predictors, from the original 2672, on the basis of relevance to our research question. A description of these variables is provided below. Note that, due to cross-referencing issues with translation, there may be some slight errors in interpretation.

The data is structured as panel data covering 290 municipalities over a 17-year period from 1998 to 2014, with municipality-year as the unit of observation. Before it could be usefully analyzed, the data required a) translation from Swedish to English, and b) some imputation of missing values. For the former issue, we used R functionality to run the variable names through Google Translate and generate English descriptors. In the case of the latter issue, certain years in the 17-year period of study lack values for variables of interest. Gini coefficients, for example, are only provided for the last few years of the time window. To mitigate missing data issues, we avoided exceptionally sparse variables, and used chained multivariate imputation to fill in missing values for chosen predictors.

Variable Name	Description
municipality_name	Municipality name
municipality_id	Municipality ID
Foreign_Born_Share	Percentage of population who are foreign-born
Gini_Coefficient	Gini Coefficient (an index of economic inequality)
Median_Income	Median income
Population Share_65+	Percentage of population over 65 years old
Share_Of_Voters_Who_Voted_Local	Percentage of population who vote in local elections
Share_Of_Voters_Who_Voted_National	Percentage of population who vote in national elections
Unemployment	Unemployment rate
Youth_Unemployment	Youth unemployment rate
foretagsklimatRanking	Företagsklimat ranking (a municipal business environment index)
Year	Year
Year_id	Year ID
urbanDegree	Percentage of the municipality defined as urban
asylumCosts	Per person cost of caring for asylum-seekers
municipalityType	Specific municipality type (commuter town, suburb, etc.)
municipalityType_id	Municipality type ID
municipalityTypeBroad	General municipality type (City, town, rural)
municipalityTypeBroad_id	Broad municipality type ID
governing	Municipal government (liberal/conservative)
governing_id	Municipal government ID
refugees	Population of refugees, in hundreds
rentalApartments	Monthly cost of an apartment (in USD)
snowmobiles	Population of snowmobiles
cars	Population of cars
tractors	Population of tractors
motorcycles	Population of motorcycles
fokusRanking	Fokus ranking (a municipal quality of life index)
Fires	Number of fires

2 PRELIMINARY DATA ANALYSIS

2.1 *Why Bayesian?*

We note that an important feature of our data is the presence of multiple observations per municipality and per year which indicates that the errors across observations are not independent, but are actually correlated. We will employ a Bayesian model to account for this feature of this data. This, combined with correlations for the predictors, mean that traditional inference would suffer from bouncing betas, heteroskedasticity, and correlation in the errors - something that we hope to remedy with our model.

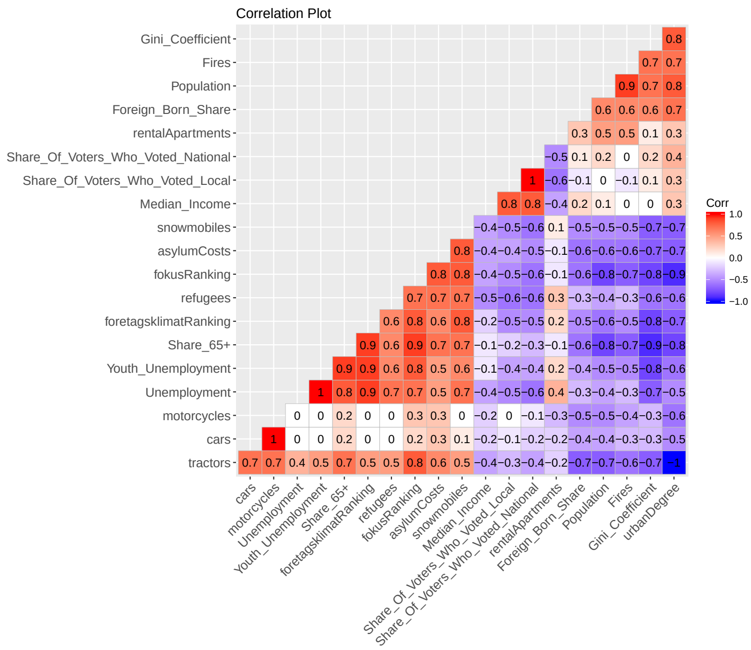


Figure 1: Correlations between the 25 predictors can be also be quite high. For example, the correlation between the population of automobiles and that of motorcycles is over 0.89. Even less obviously related metrics like the Fokus rating and youth unemployment have a correlation of 0.44.

The Bayesian priors we employ will perform regularizations to control for this, and aid us in selecting the most important of our larger number of predictors. Additionally, they can allow us to incorporate prior knowledge about the values of the parameters and intercepts- potentially useful given the economic and demographic domain.

2.2 *Data Exploration*

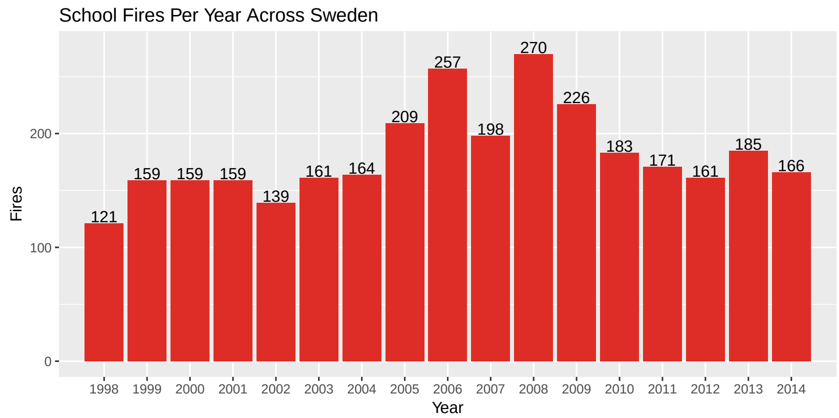


Figure 2: Fires per year. We note that the years 2006 and 2008 were apparent outliers. Allowing our intercepts to vary will hopefully account for this effect and not bias our results.

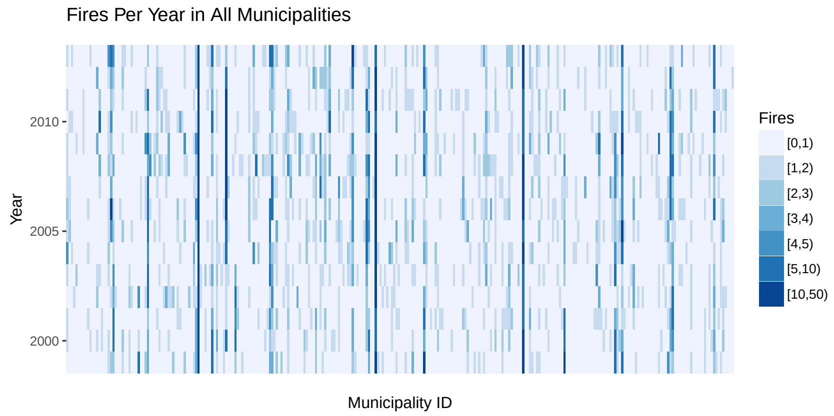


Figure 3: Fires per municipality over time. Some municipalities, like those in the center, have a frequent number of fires per year, and others have almost none (lighter colors)

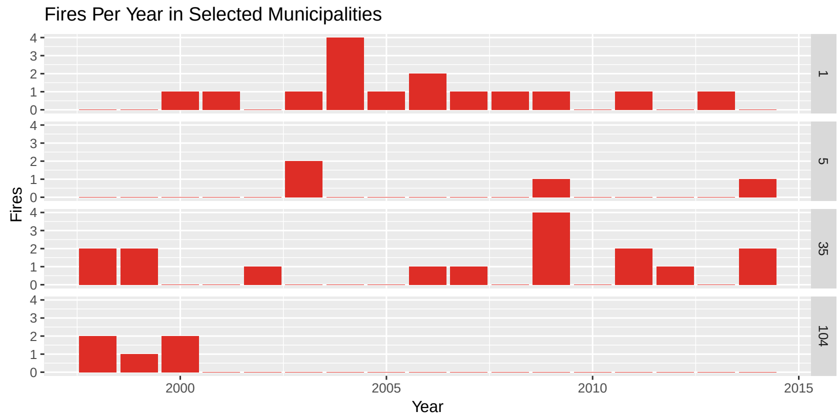


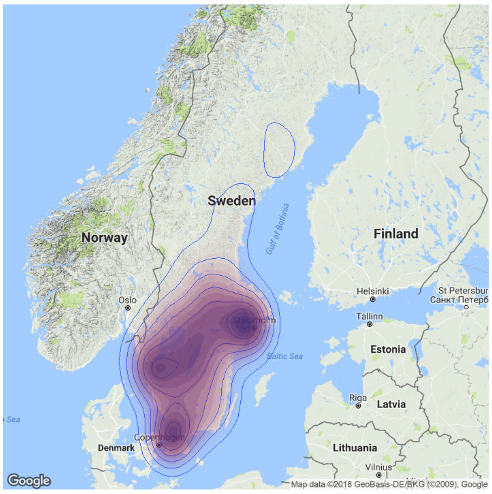
Figure 4: Fires over time for select municipalities. The variability in fires for each municipality is quite noticeable

While fires over time show a relatively stable trend (see Figure 2, 3, and 4), among the municipalities, there is a great amount of variation. While nearly every municipality has at least one fire between 1998 and 2014, a smaller number of municipalities account for the majority of the fires. For example, in 2006, one municipality had 48 school fires. This will pose a challenge when modeling to account for the high amount of between- and within- municipality variance. The data is also very heavily biased for low numbers of fires per year. Over 97% of the data has less than or equal to three fires, and so our model will need to work at both extremes - no fires and lots of fires.

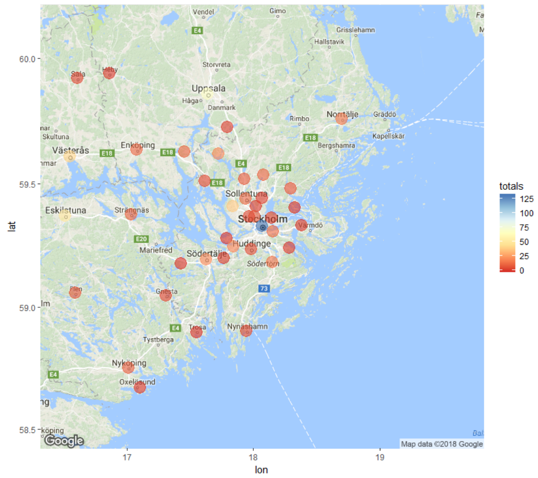
2.3 *Geographic Analysis*

Many of our predictors relate to measures of city scale and density, and anecdotally there does seem to be some connection between how rural or urban a municipality is and the incidence of school fires.

Accordingly, we began our analysis by making a heat map of school fires across Sweden.



(a) Heatmap of Sweden



(b) Heatmap of Stockholm and Surrounding Areas

At first glance, it appears that school fires are more predominant around the more populated coastal areas in the south of the peninsula, particularly around the largest cities of Stockholm, Gothenburg, and Malmö. Examining the area around capital of Stockholm more closely, we find interestingly that the municipalities more strongly impacted by school fires appear to, paradoxically, be in the less populated suburbs.

2.4 *Preliminary Regression and Key Predictor Interpretations*

For a preliminary assessment of predictor-outcome relationships, we ran some initial Bayesian regressions within a specific slice of time, prior to the full model incorporating year-to-year differences. Due to the relative sparsity of the outcome, we selected aggregate school fires in the 2010-2014 period as an outcome.[1] After comparing a handful of models for accuracy, run time, and interpretability, we settled on a fairly basic model specification below:

$$Fires_{muni} \sim Poisson(\lambda)$$
$$log(\hat{\lambda}) = \beta_0 + \sum_{i=1}^{V} \beta_i v_i$$
$$\beta_1, ..., \beta_V \sim N(0,1)$$

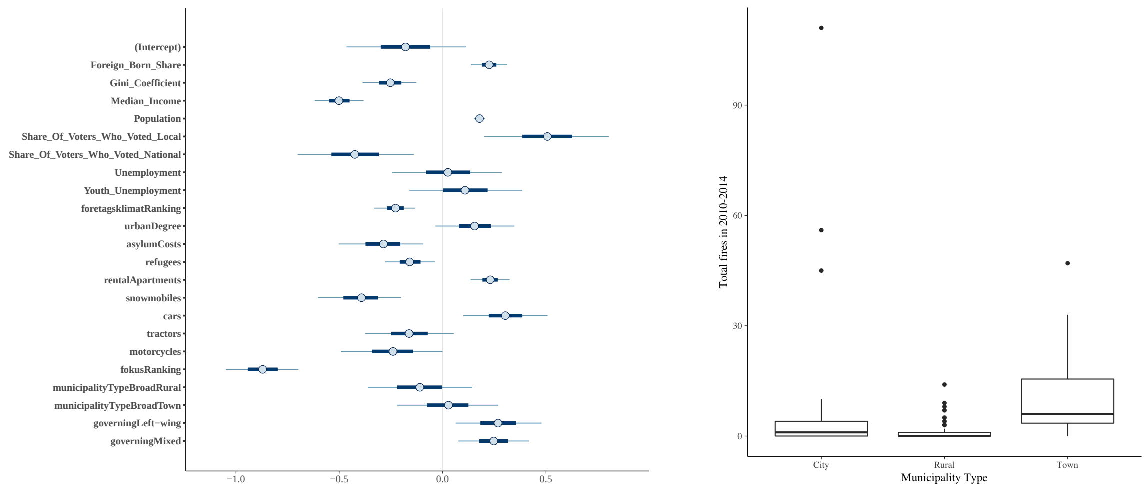
Where subscript *muni* refers to a given municipality, and V is the total number of parameters.

Preliminary model plots are shown in Figure 6a and 6b. Our results and plots provide moderate evidence that suburban areas, with higher populations but not peak population densities see the highest occurrence of school fires. More tractors and snowmobiles, proxies for rurality, correlate with fewer school arsons.

Higher levels of urbanicity correlate with more school arsons, but this tails off near higher levels of density. More directly, the box-plot in Figure 6b shows that municipalities classified specifically as towns show the highest incidence of arson in the 2010-2014 period.

Additionally, we have evidence that poor economic conditions or a lack of a vibrant economy has a positive relationship with school arson. Unemployment and youth unemployment are both correlated (albeit weakly) with school arson. We also see that median income and the Gini Coefficient (which, as an inequality measure, will often *increase* for cities with growing wealth), are both significantly negatively correlated with school arson.

[1] This period was selected due to a) the relative richness of predictor data in this period and b) the fact that Sweden's overall economic conditions were fairly consistent across this period, without much volatility.



(a) Uncertainty Intervals for Regression Parameters

(b) Fires by municipality type

Figure 6: Preliminary Findings from Analysis of 2010-2014 Fires

Comparing this model structure with alternatives yielded little deviation in performance. The table in Table 2 shows a quick comparison of two models in terms of their estimated expected log predictive density (ELPD), a measure of predictive accuracy. Estimating it enables comparison of model performance – a higher estimated ELPD suggesting a higher out-of-sample predictive performance.

$$ELPD = \sum_{i=1}^{n} E_f \left[\log p_{post}(\tilde{y}_i)\right] = \sum_{i=1}^{n} \int \log p_{post}(\tilde{y}_i)\, d\tilde{y}$$

Comparing the 5-fold cross-validated model without random effects (summarized above) against a model with random effects (on different governing structures and municipality types) shows little meaningful difference in performance. The random effects model, additionally, has a higher run time.

No.RE	With.RE
-465.92	-579.52

Table 2: Comparison of Models with and without Random Effects

3 BAYESIAN MODELING FRAMEWORK

3.1 *Early Modeling Attempts*

3.1.1 *Improving the Fit*

We took a step-wise approach to avoid building too complicated a model that would potentially overfit the data. We aimed to keep our model as simple as possible for reasons of both computation and interpretation, and consequently updated it over several iterations.

We began with a fixed effects, fixed intercepts model which was easily implemented in Rstanarm. We then realized that the variability in municipalities required a different model - one with varying intercepts for the municipalities as so we moved to STAN to implement this model. The outlier years in 2006 and 2008 were problematic in that the observations in these years were not being captured in our posterior predictive distribution, and so we incorporated a third term (the year intercepts) to account for the fact that some years had more fires than others.

Correlation in the predictors, and our ignorance as to what features should be implemented, allowed regularization priors to be implemented in the next model, which greatly reduced the bouncing betas that we were seeing in the predictors. However, divergence issues and weakly informative data forced us to re-parameterize our model to move the correlation from the parameters (the estimation of the slopes) into the hyperparameters - this is talked about more in the next section.

We decided, at last, to settle on a mixed-intercepts fixed effects model for both reasons of interpretation, computation, and it performed very well at recovering the original data in the model checking phase. We used regularization priors, and we will discuss this model in the sections to come.

We also attempted to fit a zero-inflated Poisson mixture model to account for the fact that many of the municipalities had zero fires in a given year. After looking at the improvement in fit, and weighed against the very large increase in both computation time and implementation difficulty, however, we decided against using the more complicated mixture model, and instead prefer the mixed-intercepts model for ease of computation and interpretation[2].

3.1.2 *Divergence, Hierarchical Funnels, and "Matt's Trick"*

As a consequence of hierarchical modeling, and the design of the NUTS algorithm (No U-Turn Sampler: reliant on the gradient and the Hessian of posterior space), hierarchical models are prone to divergence issues when the data is not strongly informative or when the sample size is small. Such issues are commonly known as *Neal's Funnel*. As is the case with most economic data, the amount of noise present in our data from correlated predictors and observations, inherent census sampling errors, and our economic predictors being lagging indicators, we had divergence issues plague our initial models - especially those that used centered parameterizations.

As a consequence of the geometry of the posterior space, we had to implement "Matt's Trick" (a non-centered parameterization method) to tame the posterior and move the model correlations in the parameters over to the hyper-parameters. We used the non-centered parametrization of the double exponential distribution to regularize most variables, along with the fact that it is a location-scale family, to rework our parameterization as suggested in [6] and [2]. This solved all of our divergence issues, and ultimately allowed the sampler to work much more efficiently.

3.2 *Description of the model*

Our final model was a mixed-intercepts, fixed slopes model with shrinkage priors[3].

$$
\begin{aligned}
\beta_0 &\sim N(\mu_{(Intercept)}, \sigma_{(Intercept)}) \quad \text{[Intercept Prior]} \\
\beta_{(variable_i)} &\sim Laplace(0, \sigma_{variable_i}) \quad \text{[Regularization Prior]} \\
\beta_{(year)} &\sim Laplace(0, \sigma_{year}) \quad \text{[Regularization Prior]} \\
\beta_{(muni)} &\sim Laplace(0, \sigma_{muni}) \quad \text{[Regularization Prior]} \\
\lambda_{muni,year} &= \beta_0 + \beta_{(muni)} + \beta_{(year)} + \textstyle\sum_{i=1}^{V} \beta_{(variable_i)} variable_{(i,year)} \\
Fires_{muni,year} &\sim Pois\left(\exp\left[\lambda_{muni,year}\right]\right)
\end{aligned}
$$

The final model contains several priors in order to account for the variables in the data. Now adjusting for different mean fires across 17 year and over 250 municipality intercepts, alongside the 25 socioeconomic predictors, our model estimates over 4000 means. While our preliminary model treated time as 'flat', ignoring that dimension, this model takes temporal changes into account. It also aims to manage the natural propensity of some municipalities to have a disproportionate number of fires outside of what we predict. The mixed-intercepts account for these two sources of bias. Additionally, by using regularizing priors, we can limit the dramatic, sensitive changes in predictor coefficients that would occur in a standard regression framework, hopefully providing for more stable and accurate predictions. The greatest challenge of this model is going to be picking up the outliers in the data - over 97% of the data has less than or equal to three fires a year and so we need to have a model sensitive enough to work in the tails as well.

---

[2] Please refer to our posterior predictive summaries in the model checking phase to see that the model is flexible enough to account for the zero-inflated data at the municipality level.

[3] See [7] for a discussion on mixed effects models and Page 527-528 of [6] for a non-centered parameterizations for a Double Exponential distribution to avoid divergence issues.

4 MODELING RESULTS

4.1 *Convergence and Model Checking*

After running the model, we ran ShinyStan to perform most of the model checking steps. We had effective sample sizes all greater than 40% of the total sample size, and we averaged about 90% efficiency on an *effective samples / total samples* basis. There were no messages for divergence of the chains (see Matt's Trick above), and the tree-depth was not exceeded on any of the runs. R-hat was never more than a rounding error above 1.00. The energy for each of the chains was consistent as well, indicating that each chain was exploring a similar geometry. We have elected to omit figures for the graphical model checking for the purposes of brevity, but they are available on request and we saved our model object if you would like to investigate further.

We, now, need to check that the posterior predictions cover our response variable in most cases. If the posterior predictive distribution fails to capture our data, our model may not be rich enough to express the nuances in the data.

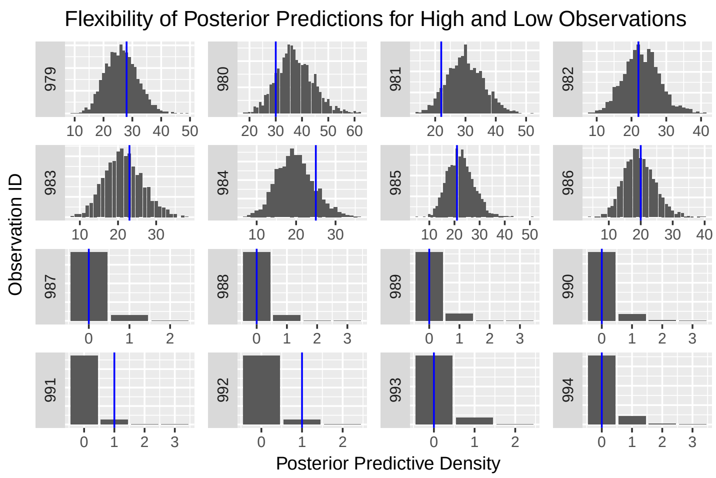


Figure 7: Histogram of fire counts. The first eight observations (on top) are from a municipality with a lot of fires, and the bottom eight observations are from a municipality with a small amount of fires.

Looking at two different municipalities, in Figure 7, one with a high number of fires (top eight), and one with a low number of fires (lower eight), we can see that the posterior predictive distribution (histogram) covers the true number of fires (verticle blue line) in every case.

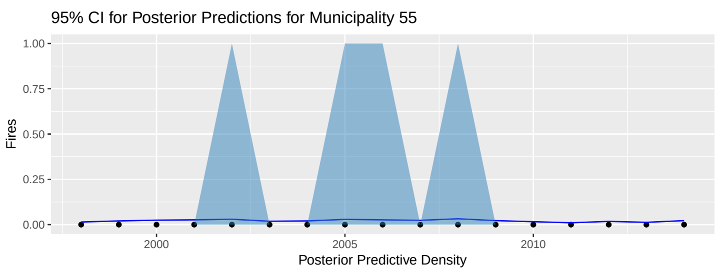


Figure 8: Credible interval, municipality with few fires

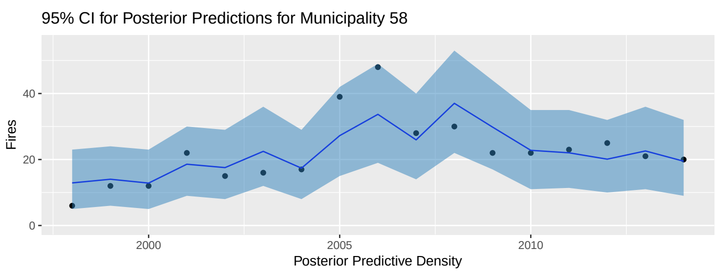


Figure 9: Credible interval, municipality with many fires

Looking at Figure 8 and 9, for municipalities with a small amount of fires, the credible interval hovers between zero and one fire- this covers 100% of the points in the first municipality (Figure 8). For municipalities with a large amount of fires, we can see that the credible interval also does a good job at capturing the number of fires. The width of the confidence intervals is largely a function of the over-dispersion that we have in our Poisson estimates as a function of the noise in our predictor variables and thus noise in our computation of λ.

4.2 *Graphical Analysis*

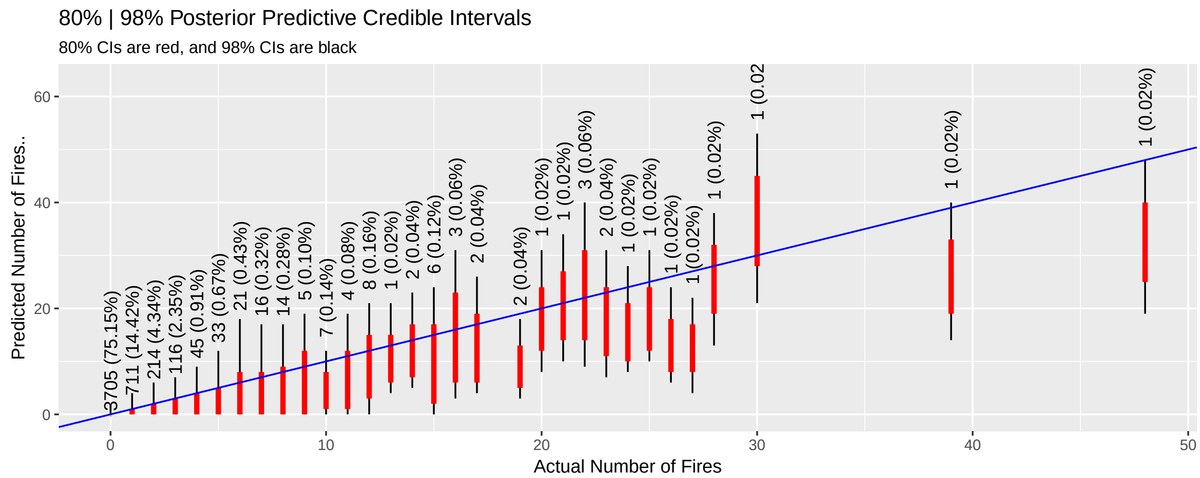


Figure 10: Credible intervals, posterior predictive distribution. The first number is the number of observations with that fire count, and the second is this number as a percentage of the total observations. Despite the incredibly rare occurrence of over 20 fires per year, our posterior intervals, in almost every case, manage to cover the true number of fires (blue line). This is impressive considering just how bias the data is (97% of our data points have less than or equal to three fires in a given year.)

We can see that most of the 98% CI's contain the actual number of fires - indicating that the model was able to capture the nuances of the data, except for a handful of cases that were outliers (such as the two observations with 19 fires). When implementing the zero-inflated model, we noticed that the posterior intervals were slightly better at capturing the true observations of number of fires, but the computational concerns, and the fact that every CI now overweighted zero, meant that the CI's had a larger standard error because of the way that the CI was constructed. In future analysis, we would like to construct bi-interval CI's to account for the bimodal nature of a zero-inflated Poisson to see if this helps better the prediction.

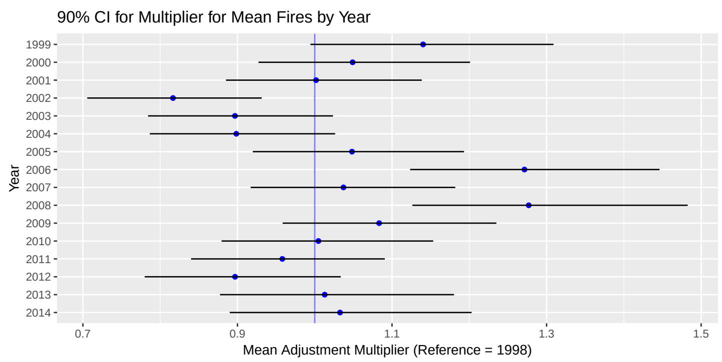


Figure 11: Comparing Confidence Intervals for Mean Fires by Year, with reference year 1998

When examining the year multiplier with reference year 1998, we see that very few years appear to be significantly different from the reference in terms of effect size. In 2002, we bias the fires down 0.70x-0.90x relative to the 1998, and we correct them upwards in 2006 and 2008, both which were years with a relatively large number of fires. This would confirm some of our earlier intuition that the number of fires does not swing as dramatically from year to year as it does from municipality to municipality.

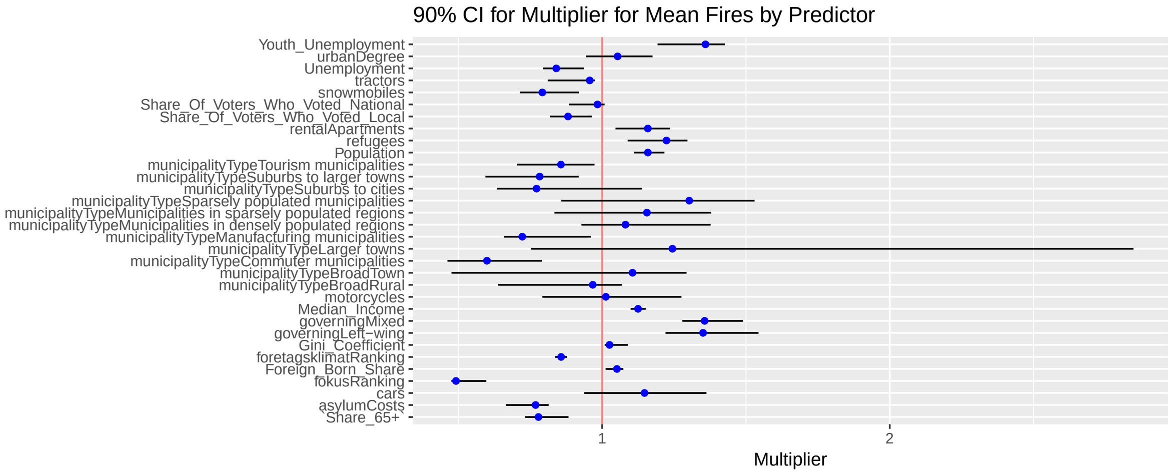


Figure 12: Mean multiplicative factor for each of the predictors

In the parameter intervals shown in Figure 12, now considering the full year-by-year analysis, we see some similarities (and some differences) from our preliminary model results. By controlling for the multiple observations on a given municipality, and controlling for the bias for fires to occur more or less frequently as a function of the year, we notice that the betas on the predictors have changed as a result. Importantly, youth unemployment now emerges as a major positive predictor, while general unemployment actually becomes a negative predictor. We also no longer see as clear a pattern in which suburban-level town types predict fire incidence, although municipalities classified as larger towns still show a boost in arson incidence. Note the significant uncertainty inherent in many of the municipality classification parameter estimates, which suggest that specific demographic and economic factors are the most relevant predictors of arson.

5 CONCLUSION

Sweden's school fires are a problem both unusual and unfortunate. Like many cultural phenomena, they resist easy quantification, prediction, and understanding. However, through Bayesian regression and hierarchical modeling, we have been able to isolate some correlates of schoolyard arson that make intuitive sense. In both our preliminary and final models, measures of low economic conditions showed significance. In particular, after expanding our model to include a time component (near-essential for economic data), youth unemployment specifically stood out. Collectively, this suggests that limited opportunities for young people are associated with more frequent man-made fires in schools.

Additionally, increasing urbanicity and wealth (as measured by improved business rankings, median income, inequality, and increasing inflows of foreign-born residents) were also a notable beta. This appeared especially true in areas of median population. Together with the economic predictors, this suggests that Swedish school fires may be more comprehensible than we first suspected. In the United States, as well as in many other countries, semi-urban environments with lower levels of economic activity are anecdotally associated with general, potentially short-lived, increases in low-level criminal behavior (e.g. vandalism, graffiti, etc). It would appear that, in Sweden, this international practice extends to criminal arson in schools.

In addition to our conclusions regarding the problem in question, some learnings were also made regarding the model itself. Very quickly, we found that the general rarity of school arson introduces estimate uncertainty. The usual regression assumptions of uncorrelated errors were disrupted by year-to-year serial correlation and the multicollinearity of otherwise very useful economic data. Hierarchical modeling and regularization priors proved valuable tools in adjusting for this.

Lastly, the data leaves plenty of room for follow up. We only scratched the surface of a deep well of predictors offered by the Swedish government, totaling over 2600 variables and offering many options to the interested analyst. While this significant dimensionality poses considerable feature selection challenges, data mining approaches, along with model selection methods like the use of Bayes information criterion, could be useful in addressing these.

6 REFERENCES

[1] Azur, Melissa, Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf. "Multiple Imputation by Chained Equations: What is it and how does it work?" *Int J Methods Psychiatr Res.* Mar. 2011.

[2] Betancourt, M. J.; Girolami, Mark. "Hamiltonian Monte Carlo for Hierarchical Models". December 2013.

[3] Data available here: `https://www.kaggle.com/mikaelhuss/swedish-school-fires`

[4] Gelman, Andrew, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. *Bayesian Data Analysis: Third Edition*. CRC Press. 2014.

[5] Johannson, Nils, Patrick van Hees, Margaret Simonson McNamee, Michael Strömgren and Robert Jansson. "Façade fires in Swedish school buildings." *EDP Sciences*. 2013.

[6] Stan Modeling Language User's Guide and Reference Manual, Version 2.17.0. `https://github.com/stan-dev/stan/releases/download/v2.17.0/stan-reference-2.17.0.pdf`

[7] Vasishth, Shravan. "Bayesian Linear Mixed Models using Stan: A tutorial for psychologists, linguists, and cognitive scientists". `http://www.ling.uni-potsdam.de/~vasishth/statistics/BayesLMMs.html`. Stan Tutorials.

7 CODE APPENDIX

7.1 *Stan Model*

The following code is the script file for Stan. This is the sixth implemented model.

```
/*
This is the model that uses vectorization, non-centered parameterization
"Matts Method" as well as regularization to estimate the coefficients for the
years, municipalities, and the betas.
*/
data {
  int<lower = 1> nobs; // Number of observations / rows
  int<lower = 1> n_muni; // Number of municipalities
  matrix[nobs, n_muni] munis; // factor matrix of the municipalities
  int<lower = 1> n_preds; // Number of the predictors
  matrix[nobs, n_preds] preds; // Predictors themselves
  int<lower = 1> n_years; // The number of years we have collected
  matrix[nobs, n_years] years; // Year factor matrix
  int fires[nobs]; // Get a matrix of the fire outputs
}
parameters {
  real beta_null; // model intercept
  vector[n_preds] betas; // model slopes

  vector[n_muni] beta_muni_tilde; // muni mixed-intercepts temp param
  real<lower = 0> beta_muni_sd; // muni mixed-sd

  vector[n_years] beta_year_tilde; // year mixed-intercepts temp param
  real<lower=0> beta_year_sd; // year mixed-sd
}
transformed parameters {
  vector[n_years] beta_year; // year mixed-intercepts
  vector[n_muni] beta_muni; // muni mixed-intercepts

  // Use matts trick (non-centered reparam) to get the results that we want
  // http://mc-stan.org/users/documentation/case-studies/divergences_and_bias.html
  // Now beta_year is basically a double exponential
  // See page 528 of Stan-reference 2.17.0 to see this parameterization derivation
  beta_year = 0 + beta_year_sd * beta_year_tilde;
  beta_muni = 0 + beta_muni_sd * beta_muni_tilde;

}
model {
  // Means for each muni-year combination
  vector[nobs] lambda;

  // Prior on the global intercept
  beta_null ~ normal(0, 1);

  // Prior on the predictor terms
  betas ~ double_exponential(0, 0.5);

  // Non-centered double exponential for beta_muni
  beta_muni_sd ~ exponential(0.5);
  beta_muni_tilde ~ normal(0,1);

  // Non-centered double exponential for beta_year
  beta_year_sd ~ exponential(0.5);
  beta_year_tilde ~ normal(0,1); // This is a divergence hack....

  // Update the mean for each of the predictors
  lambda = rep_vector(beta_null, nobs) +
    munis * beta_muni +
    years * beta_year +
    preds * betas;

  // Update the posterior
  fires ~ poisson_log(to_array_1d(lambda));
```

```
}
```

../code/swed_fires_model_six_new_data_doubleexp.stan

7.2 *Stan Call Script and Visualizations*

The following code is responsible for running the stan model and producing the visualizations that are used in model checking, posterior predictions, and inference analysis.

```
###############################################################################
#                                                                             #
# Purpose:        Stan Implementation File  - Sweden School Fires             #
#                                                                             #
# Author:         Mark Kurzeja                                                #
# Contact:        mtkurzej@umich.edu                                          #
# Client:         Mark Kurzeja                                                #
#                                                                             #
# Code created:   2018-04-04                                                  #
# Last updated:   2018-04-24                                                  #
#                                                                             #
# Comment:        A file responsible for getting the stan models up and running #
#                                                                             #
###############################################################################

rm(list = ls())
setwd("C:/Users/Mark k/Dropbox/Graduate School/05) Courses/Stats 551/551FinalProject/data")
library(magrittr)
library(stringr)
library(dplyr)
library(tidyr)
library(readxl)
library(ggcorrplot)
library(GGally)
library(rstan)
options(mc.cores = 8)
rstan_options(auto_write = TRUE)

###############################################################################
#                                                                             #
#                               Data Prep                                     #
#                                                                             #
###############################################################################

# Read in the data
mdat <- readxl::read_xlsx("./yearly_joint_data_clean.xlsx")
mdat_full <- mdat

# Remove some of the other columns
varsToRemove <- c("municipality_name", "Year_id", "municipalityType_id",
          "municipalityTypeBroad_id", "governing_id")
mdat[,varsToRemove] <- NULL

# Convert Chars to Factors:
mdat$municipalityType %<>% as.factor()
mdat$municipalityTypeBroad %<>% as.factor()
mdat$governing %<>% as.factor()
mdat$municipality_id %<>% as.factor()
mdat$Year %<>% as.factor()

# Scale some of the factors that we are going to use - specify the ones we do not want to
    scale
mdat[, colnames(mdat) %>% setdiff(c("municipality_id", "Year",
                                "municipalityType", "municipalityTypeBroad",
                  "governing", "Fires"))] %<>% scale()

# FOR DEV ONLY - Downsample data to make model run faster
# mdat <- mdat %>% sample_n(200)

```

```
################################################################################
#                                                                              #
#                           Build the Stan Object                              #
#                                                                              #
################################################################################

# Grab the fires variable - it disappears after we make the model matrix
Fires <- mdat$Fires

# Make the model matrix to extract the data from:
mdat <- model.matrix(Fires ~., data = mdat)

# Ensure that it is a dataframe so that we can work with the dplyr functions
mdat %<>% data.frame(check.names = F)

# Get the factor matrix of the munipalities
muni_matrix <- mdat %>% dplyr::select(`(Intercept)`, dplyr::contains("municipality_id"))
mdat_temp <- mdat[,setdiff(colnames(mdat), colnames(muni_matrix))]

# Get the factor matrix of the years
year_matrix <- mdat %>% dplyr::select(contains("Year"))
mdat_temp <- mdat_temp[,setdiff(colnames(mdat_temp), colnames(year_matrix))]

# Get the stan object as we need it...
stan_pass <- list(
  nobs = nrow(mdat),
  n_muni = ncol(muni_matrix),
  munis = muni_matrix,
  n_preds = ncol(mdat_temp),
  preds = mdat_temp,
  n_years = ncol(year_matrix),
  years = year_matrix,
  fires = Fires
)

################################################################################
#                                                                              #
#                            Run the Stan Object                               #
#                                                                              #
################################################################################

if(FALSE) {
# Run and save the model
mmod_doublexp <- stan(file = "../code/swed_fires_model_six_new_data_doubleexp.stan", data =
    stan_pass,
                      chains = 4, iter = 1200,
                      warmup = 600, thin = 1, refresh = 1200,
                      # control = list(max_treedepth = 15),
                      verbose = F, pars = c("betas", "beta_muni", "beta_year",
                 "beta_year_sd", "beta_muni_sd", "beta_null"))
  save(mmod_doublexp, file = "../code/full_model_inference_exp.model")
} else {
  load("../code/full_model_inference_exp.model")
}

############################
# shinystan::launch_shinystan(mmod_doublexp)


################################################################################
#                                                                              #
#                               Data Export                                    #
#                                                                              #
################################################################################
if (FALSE) {
  dd <- extract(mmod_doublexp) %>% data.frame()

  result <- list()

  # Get the muni_vars
```

```
  result[["munis"]] <- dd %>% dplyr::select(contains("beta_muni."))
  colnames(result[["munis"]]) <- c("Intercept", sprintf("Muni.%i", 2:290))

  # Get the years vars
  result[["years"]] <- dd %>% dplyr::select(contains("beta_year."))
  colnames(result[["years"]]) <- as.character(1999:2014)

  # Get the coefficients
  result[["betas"]] <- dd %>% dplyr::select(contains("betas"))
  colnames(result[["betas"]]) <- colnames(stan_pass$preds)

  # Get the last of the parameters
  result[["last"]] <- dd[,c("beta_year_sd", "beta_muni_sd", "beta_null")]

  dplyr::bind_cols(result) %>%
    write.csv("./stan_output_table_exp.csv", row.names = F)
}

################################################################################
#                                                                              #
#                        Simulate the Posterior Predictive                     #
#                                                                              #
################################################################################
dd <- extract(mmod_doublexp) %>% data.frame()

# Matrix multiply two dataframes
mmult <- function(x,y) {
  as.matrix(x) %*% t(as.matrix(y))
}

# This is the function that generates posterior predictions for each of the parameters
post_pred_values <- function(dat_row, n_samp = 200) {
  # Get the data that we are working with
  y = stan_pass$years[dat_row, ]
  m = stan_pass$munis[dat_row, ]
  p = stan_pass$preds[dat_row, ]

  year_mat <- dd[, grep(x = colnames(dd), pattern = "beta_year.", fixed = T)]
  muni_mat <- dd[, grep(x = colnames(dd), pattern = "beta_muni.", fixed = T)]
  pred_mat <- dd[, grep(x = colnames(dd), pattern = "betas", fixed = T)]
  beta_null_mat <- dd[, grep(x = colnames(dd), pattern = "beta_null", fixed = T)]

  row_choices <- sample(1:nrow(dd), n_samp, replace = T)

  k <- mmult(year_mat[row_choices,], y) +
    mmult(muni_mat[row_choices,], m) +
    mmult(pred_mat[row_choices,], p) + as.matrix(beta_null_mat[row_choices], ncol = 1)
  data.frame(yhat = k %>% as.numeric %>% exp %>% rpois(n = length(.) * 10, lambda = .))
}

################################################################################
#                                                                              #
#                                 Visualizations                               #
#                                                                              #
################################################################################

# -------------------- Plotting the posterior predictive --------------------
myggsave <- function(name, w = 6, h = 4) {
  ggsave(filename = sprintf("../fig/%s.pdf", name), device = "pdf", width = w, height = h)
}

plotss <- plyr::ldply(seq_len(stan_pass$nobs), function(i) {
  v = post_pred_values(i)
  data.frame(observation_id = i, actual = Fires[i], yhat = v)
}, .progress = plyr::progress_win())

# First save down this dataframe for future :)
if (FALSE) {
  # plotss %>% write.csv("./post_pred_exp.csv", row.names = F)
  save(plotss, file = "../code/plotss_exp.data")
```

```
}
load("../code/plotss_exp.data")

# ---------- Plot the confidence intervals for each of the parameters ----------
library(HDInterval)
plotss %>% group_by(actual) %>%
  summarise(lower = hdi(yhat, credMass = 0.98)[1],
            upper = hdi(yhat, credMass = 0.98)[2],
            lowermid = hdi(yhat, credMass = 0.80)[1],
            uppermid = hdi(yhat, credMass = 0.80)[2],
            count = n() / 2000) %>%
  mutate(pers = count / sum(count), lable = sprintf("%i (%.2f%%)", count, pers * 100)) %>%
  ungroup() %>%
  ggplot(.) +
  geom_segment(aes(x = actual, xend = actual, y = lower, yend = upper)) +
  geom_segment(aes(x = actual, xend = actual, y = lowermid, yend = uppermid), size = 1.5,
    color = "red") +
  geom_abline(intercept = 0, slope= 1, color = "blue") +
  geom_text(aes(actual, upper, label = lable, angle = 90), nudge_y = 10) +
  labs(x = "Actual Number of Fires", y = "Predicted Number of Fires..") +
  ggtitle("80% | 98% Posterior Predictive Credible Intervals", "80% CIs are red, and 98% CIs
    are black")
myggsave("CI_post_pred_intervals", w = 10, h = 4)

# ---------------- Plot the posterior coverage for First 20 Obs ----------------
plotss %>% filter(observation_id %in% seq(979,length.out = 16)) %>%
  ggplot(.) +
  geom_bar(aes(yhat)) +
  facet_wrap(~observation_id, scales = "free", strip.position = "left") +
  geom_vline(aes(xintercept = actual), color = "blue") +
  theme(axis.text.y = element_blank(), axis.ticks.y = element_blank()) +
  labs(x = "Posterior Predictive Density", y = "Observation ID") +
  ggtitle("Flexibility of Posterior Predictions for High and Low Observations")
myggsave("post_pred_samples", w = 6, h = 4)

# ----------------------- Predictions for an active muni -----------------------
obsnum = 58
plotss2 <- plyr::ldply(which(mdat_full$municipality_id == obsnum), function(i) {
  v = post_pred_values(i, 5000)
  data.frame(observation_id = i, actual = Fires[i], yhat = v)
}, .progress = plyr::progress_win())

ys <- data.frame(observation_id = which(mdat_full$municipality_id == obsnum), years =
    1998:2014 )
plotss2 %>%
  filter(observation_id == which(mdat_full$municipality_id == obsnum)) %>%
  left_join(., ys) %>% group_by(years) %>%
  summarize(lower = quantile(yhat, probs = 0.01),
            upper = quantile(yhat, probs = 0.99),
            actual = mean(actual),
            mean = mean(yhat)) %>%
  ggplot(.) +
  scale_y_continuous(limits = c(0,55)) +
  geom_point(aes(years, actual)) +
  geom_line(aes(years, mean), color = "blue") +
  geom_ribbon(aes(x = years, ymin = lower, ymax = upper),
              alpha = 0.5, fill = RColorBrewer::brewer.pal(3, "Blues")[3]) +
  labs(x = "Posterior Predictive Density", y = "Fires") +
  ggtitle(sprintf("95%% CI for Posterior Predictions for Municipality %i", obsnum))
myggsave("post_pred_samples_ribbon_high", w = 8, h = 3)

# ------------------------ Predictions for a dead muni ------------------------
obsnum = 55
plotss2 <- plyr::ldply(which(mdat_full$municipality_id == obsnum), function(i) {
  v = post_pred_values(i, 2000)
  data.frame(observation_id = i, actual = Fires[i], yhat = v)
}, .progress = plyr::progress_win())

ys <- data.frame(observation_id = which(mdat_full$municipality_id == obsnum), years =
    1998:2014 )
```

```
plotss2 %>%
  filter(observation_id == which(mdat_full$municipality_id == obsnum)) %>%
  left_join(., ys) %>% group_by(years) %>%
  summarize(lower = quantile(yhat, probs = 0.025, type = 2),
            upper = quantile(yhat, probs = 0.975, type = 2),
            actual = mean(actual),
            mean = mean(yhat)) %>%
  ggplot(.) +
  geom_point(aes(years, actual)) +
  geom_line(aes(years, mean), color = "blue") +
  geom_ribbon(aes(x = years, ymin = lower, ymax = upper),
              alpha = 0.5, fill = RColorBrewer::brewer.pal(3, "Blues")[3]) +
  labs(x = "Posterior Predictive Density", y = "Fires") +
  ggtitle(sprintf("95%% CI for Posterior Predictions for Municipality %i", obsnum))
myggsave("post_pred_samples_ribbon_low", w = 8, h = 3)

# ---------------------- Plotting the Year Multipliers ----------------------
yy <- dd[,c("beta_year.1", "beta_year.2",
            "beta_year.3", "beta_year.4", "beta_year.5", "beta_year.6", "beta_year.7",
            "beta_year.8", "beta_year.9", "beta_year.10", "beta_year.11",
            "beta_year.12", "beta_year.13", "beta_year.14", "beta_year.15",
            "beta_year.16")]

colnames(yy) <- c(1999:2014)

yy %>%
  tidyr::gather(factor_key = T) %>%
  mutate(value = exp(value)) %>% group_by(key) %>%
  summarize(lower = quantile(value, probs = 0.05),
            upper = quantile(value, probs = 0.95),
            median = median(value)) %>%
  mutate(key = factor(key, levels = rev(levels(key)))) %>%
  ggplot(.) +
  geom_point(aes(median, key), color = "blue") +
  geom_vline(xintercept = 1, color = "blue", alpha = 0.5) +
  geom_segment(aes(x = lower, xend = upper, y = key, yend = key)) +
  labs(x = "Mean Adjustment Multiplier (Reference = 1998)", y = "Year") +
  ggtitle("90% CI for Multiplier for Mean Fires by Year")
myggsave("year_multiplier", w = 8, h = 4)

# ---------------------- Plotting the Beta Multipliers ----------------------
dd <- extract(mmod_doublexp) %>% data.frame()

# Get the coefficients
bb <- dd %>% dplyr::select(contains("betas"))
colnames(bb) <- colnames(stan_pass$preds)

# Plot
bb %>%
  head() %>%
  tidyr::gather() %>%
  mutate(value = exp(value)) %>%
  group_by(key) %>%
  summarize(lower = quantile(value, probs = 0.05),
            upper = quantile(value, probs = 0.95),
            median = median(value)) %>%
  ggplot(.) +
  geom_vline(xintercept = 1, col = "red", alpha = 0.5) +
  geom_segment(aes(x = lower, xend = upper, y = key, yend = key)) +
  geom_point(aes(median, key), color = "blue") +
  labs(x = "Multiplier", y = "") +
  ggtitle("90% CI for Multiplier for Mean Fires by Predictor")
  myggsave("beta_multiplier", w = 10, h = 4)

# ------------------------ Plot Some Munis Fire Rates ------------------------
mdat_full %>% dplyr::filter(municipality_id %in% c(1,5, 35, 104)) %>%
    dplyr::select(municipality_id, Fires, Year) %>%
  ggplot(data = ., aes(x = Year, y = Fires)) +
  geom_bar(stat = "identity", fill = RColorBrewer::brewer.pal(3, "Reds")[3]) +
  facet_grid(municipality_id~.) +
```

```
  theme(legend.position = "none") +
  ggtitle("Fires Per Year in Selected Municipalities")
myggsave("FiresPerMuni",8)

# ———————————————— Fires Per Year Plot ————————————————
mdat_full %>%
  dplyr::select(Fires, Year) %>%
  group_by(Year) %>%
  summarize(Fires = sum(Fires)) %>%
  ggplot(data = ., aes(x = Year, y = Fires)) +
  geom_bar(stat = "identity", fill = RColorBrewer::brewer.pal(3, "Reds")[3]) +
  geom_text(aes(Year, Fires, label = Fires),nudge_y = 6) +
  scale_x_discrete(limits = 1998:2014) +
  theme(legend.position = "none") +
  ggtitle("School Fires Per Year Across Sweden")
myggsave("FiresPerYear", 8)

# ————————— Plot the color tiles for the number of fires —————————
mdat_full %>%
  dplyr::select(municipality_id, Fires, Year) %>%
  mutate(Fires = as.factor(cut(Fires, breaks = c(0,1,2,3, 4, 5,10, 50), right = F))) %>%
  ggplot(data = ., aes(x = municipality_id, y = Year)) +
  geom_tile(aes(fill = Fires)) +
  scale_fill_brewer(palette = "Blues") +
  theme_bw() +
  theme(panel.grid = element_blank(), panel.border = element_blank()) +
  theme(axis.ticks.x  = element_blank()) +
  theme(axis.text.x  = element_blank()) +
  scale_y_continuous(limits = c(1998,2014), expand = c(0, 0)) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("Fires Per Year in All Municipalities") +
  labs(x = "Municipality ID", y = "Year")
myggsave("FiresPerMuni_tile", 8)

######################### Get the correlation matrix #########################

mc <- mdat[,c("Foreign_Born_Share", "Gini_Coefficient",
        "Median_Income", "Population", "Share_65+", "Share_Of_Voters_Who_Voted_Local",
        "Share_Of_Voters_Who_Voted_National", "Unemployment", "Youth_Unemployment",
        "foretagsklimatRanking",  "urbanDegree", "asylumCosts",
         "refugees",
        "rentalApartments", "snowmobiles", "cars", "tractors", "motorcycles",
        "fokusRanking", "Fires")] %>% cor()


cormat <- round(cor(mc),1)
ggcorrplot(cormat, hc.order = TRUE, type = "lower",
           lab = TRUE, ggtheme = ggplot2::theme_gray, title = "Correlation Plot")
myggsave("corr_plot", 10, 10)
```

../code/Stan_call_script_new_data_lasso_model.R

7.3 *Preliminary Model*

This is the code that runs the preliminary analysis for our models

```
#This section of code runs a preliminary model on a set of simplified KPIs, using fires in the
    2010-2014 period as the Poisson outcome.

#Final Code
library(magrittr)
library(stringr)
library(dplyr)
library(tidyr)
library(data.table)
library(rstanarm)
library(splines)
library(readxl)
```

```r
library(magrittr)
library(BMA)
library(xtable)

#Load Data and Swap Muni Names
setwd('~/Documents/GitHub/551FinalProject/code')
joint_data = data.table(read_xlsx("../data/yearly_joint_data_clean.xlsx"))
getnames = fread("../data/yearly_joint_data.csv")
muni = unique(getnames$municipality_name)

#Function to force numeric scale output
numscale = function(x) {
  as.numeric(scale(x))
}

#Subset and Aggregate Years
TotalFires = joint_data[Year>= 2010,.(Fires = sum(Fires)),by=.(municipality_name)]$Fires

FactorVars = joint_data[Year >= 2010, lapply(.SD,min),
  by= municipality_name,
  .SDcols = c("municipalityType","municipalityTypeBroad","governing")] %>%
  .[,-c("municipality_name"),with=FALSE]

NumVars = joint_data[Year >= 2010,] %>%
  .[,lapply(.SD,mean,na.rm=TRUE),
    by=.(municipality_name),
    .SDcols = -c("municipalityType","municipalityTypeBroad","governing","Fires")] %>%
  .[,-c("municipality_name"),with=FALSE] %>%
  .[,lapply(.SD,numscale),.SDcols = -c("municipality_id")]

Prelim_Data = cbind(muni,NumVars,FactorVars,TotalFires) %>%
  .[,Share_65_Plus := `Share_65+`] %>%
  .[,-c("Year","Year_id","municipalityType_id","municipalityTypeBroad_id",
        "governing_id","Share_65+")]

Prelim_Data$municipalityTypeBroad = as.factor(Prelim_Data$municipalityTypeBroad)
Prelim_Data$governing = as.factor(Prelim_Data$governing)

#Generate formula for model
formnames = paste(names(Prelim_Data)[2:19],sep="")
modelform = as.formula(paste("TotalFires ~ ",
                             paste(formnames, collapse="+"),
                             "+ municipalityTypeBroad + governing"))

#Run model - stanreg
options(mc.cores=4)
set.seed(1)
prelimmodel = stan_glm(modelform,data=Prelim_Data,family=poisson(link="log"),prior=normal(),
     chains=4)
kprelim = kfold(prelimmodel,K=5,save_fits=TRUE)

posterior = as.array(prelimmodel)

#Run model - glmer
modelform = as.formula(paste("TotalFires ~ ",
                             paste(formnames, collapse="+"),
                             "+ (1 | municipalityTypeBroad) + (1 | governing)"))
prelim_lmer = stan_glmer(modelform,data=Prelim_Data,family=poisson(link='log'),
                         prior=normal(),adapt_delta=.99)
kprelim_lmer = kfold(prelim_lmer,K=5)

#Model comparison
lmer_lpd = kprelim_lmer$elpd_kfold
glm_lpd = kprelim$elpd_kfold
Comparison = data.frame(`No RE` = glm_lpd,`With RE` = lmer_lpd)
row.names(Comparison) = "ELPD"
xtable(Comparison,caption = "Comparison of Models with and without Random Effects")


#Visualizations/Model Checking
```

```
library(bayesplot)
library(ggplot2)

#Parameter Uncertainty
paramints = plot(prelimmodel) +
  theme(axis.text = element_text(size=12))
paramints
ggsave("../fig/parameterintervals.pdf",plot=paramints,
       width=10.7,height=7.97,units='in')

#Model Fit
y = Prelim_Data$TotalFires
ydraws = t(replicate(6,sample(prelimmodel$fitted.values,length(y),replace=TRUE)))
errorhist = ppc_error_hist(y,ydraws,freq=FALSE,binwidth=20)
ggsave("../fig/errorhist.pdf",plot=errorhist)

#Boxplot
suburbplot = ggplot(data=Prelim_Data,aes(municipalityTypeBroad,TotalFires)) +
  geom_boxplot() +
  labs(x="Municipality Type",y="Total fires in 2010-2014")
ggsave("../fig/muni_type_fires.png",plot=suburbplot,
       width=6.26,height=7.04,units='in')
```

../code/PrelimModel.R

7.4 *Code for Translating the KPI to English*

We had to translate the KPIs from Swedish to English. This code is responsible for this translation using the Google Translate API.

```
#This section of code uses Google Translate to translate each of the 2600 KPIs from Swedish
    into English.

library(data.table)
library(magrittr)
library(googleLanguageR)

setwd("/Users/Desmond/Desktop/Work/551 data/Swedish")
temp = list.files(pattern="*.csv")
datalist = lapply(temp,fread)
FullData = datalist[[1]]
for(i in 2:5){
FullData = rbind(FullData,datalist[[i]])
}
TestData = FullData[!duplicated(FullData[,1])] %>%
  .[,c(1,6)]

#Use gl_translate to connect to Google through API and generate translations.
#In order to successfully run this code, you must obtain an API key from Google, which was
    obtained
#for this analysis using a free trial.
translate_KPIs = gl_translate(TestData$kpi_desc,target='en',
                             source='sv')

#Generate dataset of translated KPIs.
EnglishKPIs = data.table(kpi = TestData$kpi,
                         KPI_desc = translate_KPIs$translatedText)

#Merge translated KPIs with original data and export file.
EnglishData = FullData[EnglishKPIs,on="kpi"]
write.csv(EnglishData,file='../DataWithEnglishKPIs.csv')
zip("../DataWithEnglishKPIs_ZIP","../DataWithEnglishKPIs.csv")
```

../code/Translations.R

7.5 *Refactoring Code*

This takes the Translated data and makes it into the panel data.

```
#This code draws on the translated dataset to format and export a panel data file for selected
     predictors of interest. We focused specifically on a set of manually chosen variables of
    interest. For future work, data mining techniques could be explored to uncover more
    significant patterns.



library(data.table)
library(magrittr)
library(readxl)
library(dplyr)
FullData = fread("/Users/Desmond/Desktop/Work/551 data/DataWithEnglishKPIs.csv")
listvars = unique(FullData$KPI_desc)
checkunem = grepl("foreign-born",listvars)
listvars[checkunem]


#Generate list of KPI names and IDs in English
KPI_English_Names = unique(FullData[,c("kpi","KPI_desc")])
write.csv(KPI_English_Names, file="./Documents/GitHub/551FinalProject/data/KPI_English_Names.
    csv")

otherdata = fread("/Users/Desmond/Documents/GitHub/551FinalProject/data/simplified_
    municipality_indicators.csv")

KPIs = c("Number of persons aged 16-24 in the municipality who are open unemployed or in
    programs with activity support,
divided by the number of residents 16-24 years in the municipality on 31/12 years T-1.
    Unemployment refers to statistics
from March month T. Source: Employment and Statistics Sweden.",
        "Number of unemployed unemployed and persons in programs with activity support
    between the ages of 18 and 64 divided
by the number of residents 18-64 years. Refers to statistics from March month T. Source:
    Employment Service and Statistics Sweden.",
        "Number of inhabitants 65-79 years divided by number of inhabitants total 31/12.
    Source: SCB.",
        "Total number of inhabitants on 31/12. Source: SCB.",
        "Number of votes cast in the last municipal elections (valid and invalid) divided by
    the number of eligible
voters multiplied by 100. Source: Valuation and SCB.",
        "Number of votes cast in the last parliamentary elections (valid and invalid) divided
     by the number of eligible
voters multiplied by 100. Source: Valuation and SCB.",
        "Total income earned between 20-64 years municipality (median), kr. Total earned
    income is the sum of income from
employment and income from business activities. The accumulated acquisition income consists of
     the total current taxable
income, which refers to income from employment, entrepreneurship, retirement, sickness benefit
     and other taxable transfers.
Total earned income does not include income from capital. Source: SCB.",
        "This is a development key figure, see questions and answers to kolada.se for more
    information. Municipal ranking
(1-290) of the complex business environment. The rankings contain a total of 18 factors
    weighted differently heavily.
The heaviest weighting in the ranking is the company's assessment of% u201DThe summary
    assessment of the business environment
in the municipality% u201D. Source: Swedish Enterprise",
        "The gin coefficient has a value between zero (0) and one hundred percent (1). 0
    means that all individuals have
exactly equal assets (ie total equality) while 1 means total inequality. Based on total earned
     income. Source: SCB.",
        "Number of foreign-born members in the municipality divided by the total number of
    members in the municipality
multiplied by 100. The statistics are only published every four years, but in Kolada it is
    published in addition to the
year T, T + 1, T + 2 and T + 3. Source: SCB.")
```

```r
KPIs = gsub("\r?\n|\r"," ", KPIs)

KPINames = c("Youth_Unemployment",
             "Unemployment",
             "Share_65+",
             "Population",
             "Share_Of_Voters_Who_Voted_Local",
             "Share_Of_Voters_Who_Voted_National",
             "Median_Income",
             "foretagsklimatRanking",
             "Gini_Coefficient",
             "Foreign_Born_Share")


KPINameData = data.table(KPI_desc = KPIs, Vars = KPINames)


#Subset to Year-by-Year Data of interest
YearlyData = FullData[KPI_desc %in% KPIs,-c("kpi_desc","V1"),with=FALSE] %>%
  .[KPINameData,on="KPI_desc"] %>%
  .[,-c("KPI_desc","kpi"),with=FALSE] %>%
  dcast(municipality_name + municipality_id + period ~ Vars,value.var='value') %>%
  .[order(municipality_name)] %>%
  .[,Year := as.numeric(period)] %>%
  .[,-c("period")]


#Load joint table of simplified indicators
other_data = fread("../data/untouched data/simplified_municipality_indicators.csv") %>%
  .[order(name)]

simpl_data = other_data
for(i in 1:16){
  simpl_data = rbind(simpl_data,other_data)
}
Year = rep(1998:2014,each=length(unique(YearlyData$municipality_name)))

simpl_data = cbind(simpl_data,Year) %>%
  .[order(name)] %>%
  .[,municipality_name := name] %>%
  .[,c("municipality_name","Year","urbanDegree","asylumCosts","municipalityType",
       "municipalityTypeBroad","governing","refugees","rentalApartments",
       "snowmobiles","cars","tractors","motorcycles","fokusRanking"),with=FALSE]


#Load table of fires
firedata = fread("../data/untouched data/school_fire_cases_1998_2014.csv") %>%
  .[,-c("Population"),with=FALSE]

missingdata = data.table(Municipality = c(rep("Knivsta",5),"Nykvarn"),Cases = rep(0,6),
                         Year = c(1998:2002,1998))

firedata = rbind(firedata,missingdata) %>%
  .[order(Municipality)]


# Join yearly data with simplified
agg_data = YearlyData[simpl_data,on=c("municipality_name","Year")] %>%
  cbind(Cases = firedata$Cases)

#Variable type conversions
agg_data[,3:12] = agg_data[,3:12] %>% mutate_if(is.character,as.numeric)
agg_data[,3:ncol(agg_data)] = agg_data[,3:ncol(agg_data)] %>% mutate_if(is.character,as.factor)


#Imputation of missing data
library(mice)

impdata = mice(agg_data[,3:26],m=5,maxit=20,method='pmm',ridge=.0001)
```

```r
finaldata = complete(impdata)

finaldata = cbind(agg_data[,1:2],finaldata)

#Export final dataset as csv
write.csv(finaldata,"./Documents/GitHub/551FinalProject/data/yearly_joint_data.csv")
```

../code/ExtractYearByYearVars.R