

BDA Project: Influence of Unemployment Rate on Suicide Rate

Duong Le, Guting Huang

December 2022

Contents

Introduction	1
Problem Description	2
Data Description	2
Main modeling idea	2
Separate model	4
Model description	4
Stan code and running option	4
Convergence diagnostics	5
Posterior predictive check	7
Predictive performance assessment	8
Prior sensitivity analysis	11
Hierarchical model	12
Model description	12
Stan code and running options	13
Convergence diagnostics	14
Posterior predictive check	15
Predictive performance assessment	18
Prior sensitivity analysis	18
Model comparision	19
Disscussion	20
Conclusion of the results	20
Reflection	21
References	21

Introduction

Globally around 800,000 people die from suicide every year. Drop in unemployment at the start and during the COVID-19 pandemic is unprecedented, and there is a noticeable increase in the number of suicides. After this observation, we decided to explore the relationship between unemployment rate and suicide rate, trying to make predictions based on unemployment rate for each country. If suicide rates can be predicted by unemployment rate, it can become a useful factor to consider when planning for preventative interventions.

Problem Description

Our goal is to investigate in the relationship between unemployment rate and suicide rate, trying to make predictions based on unemployment rate for each country.

Data Description

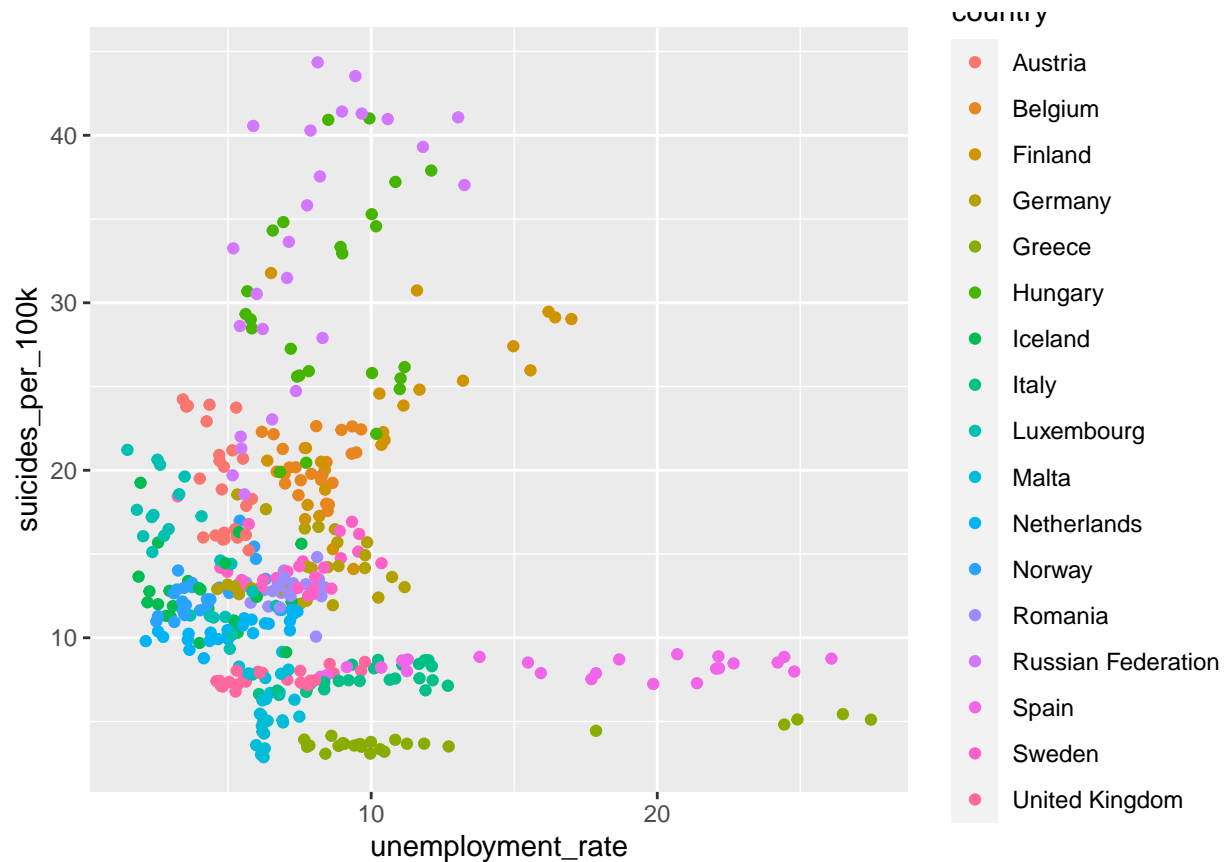
Our data consists of two datasets retrieved from kaggle.com. The first dataset contains suicide information by year and country from 1985-2016 with 12 variables (country, year, sex, age group, count of suicides, population, suicide rate, country-year composite key, HDI for year, gdp for year \$, gdp per capita, generation). The other dataset includes the unemployment rates by country from 1991-2021.

Since the two datasets have a different time range, we first took their intersection by year and kept only rows whose year are common to both, from 1991-2015. Furthermore, some of the countries do not have complete data for every years in the selected time range, therefore we decided to focus on those European countries that have complete suicide information from 1991-2015. Because of the limited data, we also decided to ignore the year attribute. Finally, the remaining variables that are of interest are country, suicide rate (suicide per 100k), and unemployment rate. In addition, we fit the model using data until 2014, and leave 2015 for predictive performance assessment.

##	year	country	gdp_per_capita	suicides_per_100k
## 1	1991	Austria	23808	24.233780
## 2	1991	Belgium	22523	19.209076
## 3	1991	Finland	27214	31.776099
## 4	1991	Germany	24662	18.557299
## 5	1991	Greece	10816	3.919292
## 6	1991	Hungary	3561	40.921252

Main modeling idea

Before going into the models, let us take a look at the data:



A quick look at the data suggests that there are no overall trend between the unemployment rate and the suicide rate. However, upon a closer look, we can see that the unemployment rate and the suicide rate of individual countries follows a linear trend. Thus, we decide that we would implement two models: a linear separate model and a linear hierarchical model.

In more detail, denote the suicide rate of country j as y_j , and the corresponding unemployment rate as x_j . We want to fit the following model:

$$y_j = \beta_{0_j} + x_j \cdot \beta_{1_j}$$

The details of the model and how we choose the priors will be presented in each model section below.

But before going in detailed about the models, we should prepare the data to pass into the Stan program.

```
# Add a group indicator column to the data
countries <- unique(data$country)
for (i in 1:length(countries)) {
  data$group[data$country == countries[i]] <- i
}

data_train <- data[data$year < 2015,] # We leave out data in 2015 to use in prediction
data_valid <- data[data$year == 2015,]

# Data for the stan models
stan_data <- list(
  N = nrow(data_train),
  J = length(countries),
```

```

id = data_train$group,
x = data_train$unemployment_rate,
y = data_train$suicides_per_100k
)

```

Separate model

Model description

Based on Fountoulakis et al's paper Fountoulakis et al. (2014), the yearly suicide rate are concentrated around 12.83. We also see that the estimated coefficient for the National unemployment rate is 0.005. Thus, it is sensible to say that $\beta_1 \sim N(0, \sigma_{\beta_1})$. We also believe that it is reasonable that an increase in 1% of unemployment rate will not change the suicide rate by more than 50%. Thus, the following probability should hold for β_1 :

$$Pr(-6.415 < \beta_1 < 6.415) = 0.99$$

Solve for σ_{β_1} , we obtain:

$$\beta_1 \sim N(0, 2.5)$$

In the same paper, the authors also find out that the intercept for the suicide rate is 1.89. And we believe that the intercept should not have a standard deviation more than 20. Thus, we have the following priors for the intercept:

$$\beta_0 \sim N(1.89, 20)$$

For the variance parameter, we do not have any prior knowledge about this. For this reason, we decide to go with a weakly informative prior, more explicitly, a heavy-tailed prior distribution. Because of the heavier tail, it would be safer if the parameters that we choose do not accomodate the true value of the variance parameter. The distribution of choice for this part is Half-Cauchy distribution:

$$\sigma \sim Half - Cauchy(0, 10)$$

Thus, we have the following separate model:

$$\begin{aligned}
\beta_{0_j} &\sim N(1.89, 20) \\
\beta_{1_j} &\sim N(0, 2.5) \\
\sigma_j &\sim Half - Cauchy(0, 10) \\
y_j &\sim N(\beta_{0_j} + x_j \cdot \beta_{1_j}, \sigma_j)
\end{aligned}$$

Stan code and running option

Below is the Stan code for the separate model

```

## data {
##   int<lower=0> N;           // number of observations
##   int<lower=0> J;           // number of countries
##   int<lower=0> id[N];       // group indicator
##   vector[N] y;
##   vector[N] x;             // Unemployment rate

```

```

## }
##
## parameters {
##   vector[J] beta0;
##   vector[J] beta1;
##   vector<lower=0>[J] sigma;
##
## }
##
## model {
##   // priors
##   for (j in 1:J){
##     beta0[j] ~ normal(1.89, 20);
##     beta1[j] ~ normal(0, 2.5);
##     sigma[j] ~ cauchy(0, 10);
##   }
##
##   // likelihood
##   for (n in 1:N) {
##     y[n] ~ normal(x[n]*beta1[id[n]] + beta0[id[n]], sigma[id[n]]);
##   }
## }
##
## generated quantities {
##   vector[N] log_lik;
##   for (n in 1:N) {
##     log_lik[n] = normal_lpdf(y[n] | x[n] * beta1[id[n]] + beta0[id[n]], sigma[id[n]]);
##   }
## }

```

This is the running command and its options:

```
sm_sep <- stan(file='stan_model/separate.stan', data=stan_data, iter=2000, refresh=0)
```

```

##
## Divergences:
## 0 of 4000 iterations ended with a divergence.
##
## Tree depth:
## 0 of 4000 iterations saturated the maximum tree depth of 10.
##
## Energy:
## E-BFMI indicated no pathological behavior.

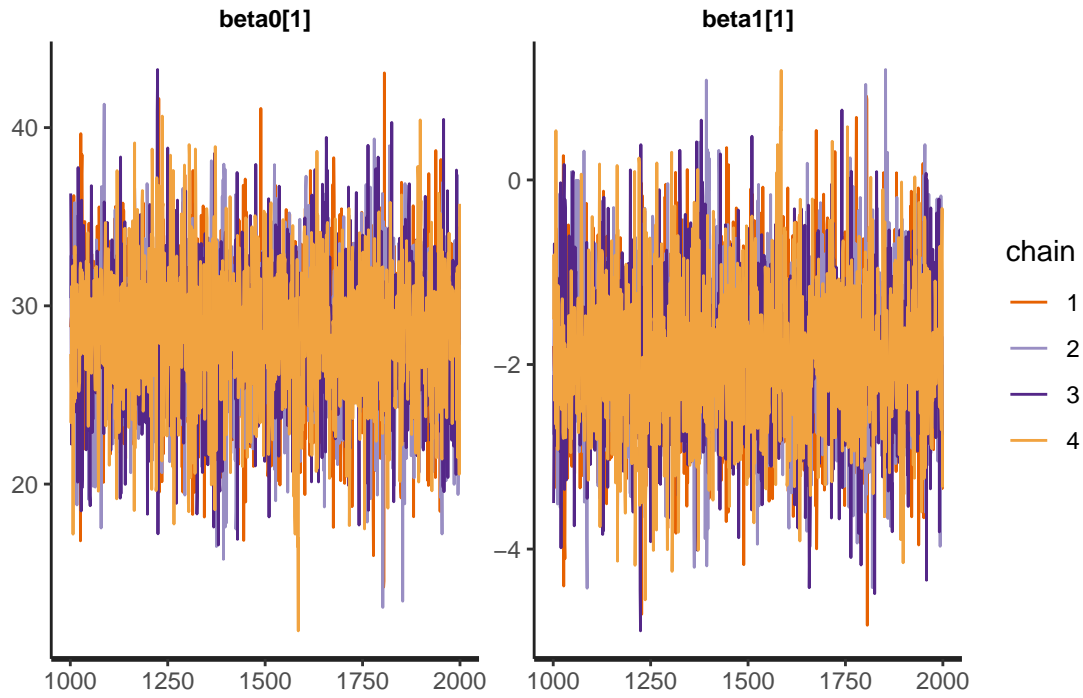
```

According to the HMC diagnostics, none of the iterations resulted in divergence, nor saturated the maximum tree depth. In other words, the sampling behaves well.

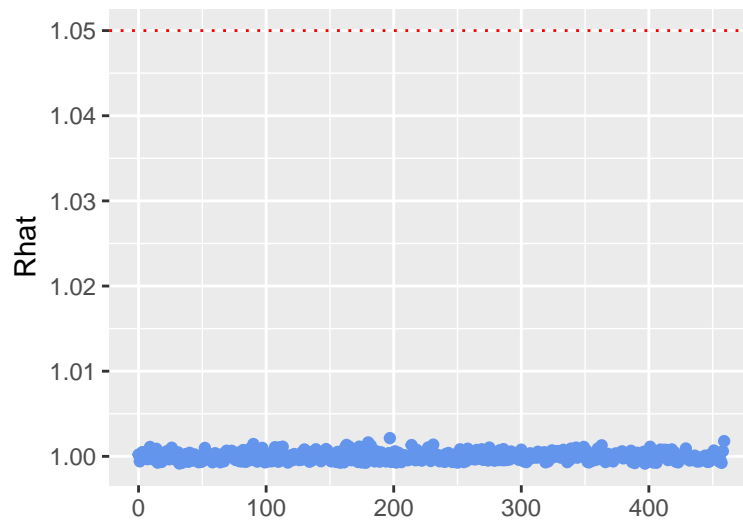
Convergence diagnostics

It seems that there were no warning on convergence for this one, so it is a good sign that the chains are not diverge. But just to be sure, let's visualize the chains for the parameters β_0 and β_1 of Austria.

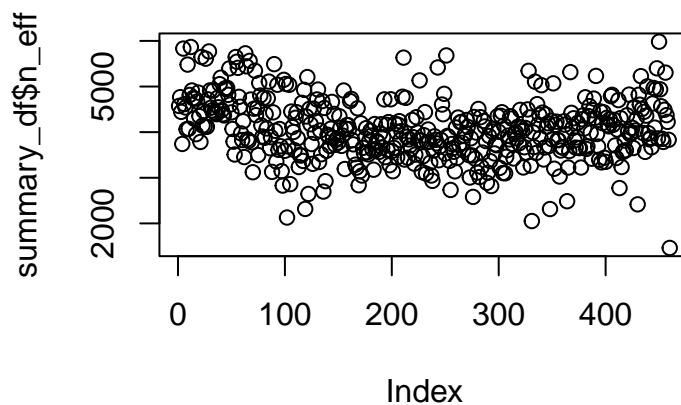
```
traceplot(sm_sep, pars=c('beta0[1]', 'beta1[1]'))
```



Although there are some chains that behave quite differently, visually the chains seem to be converged. Let us also look at the \hat{R} values



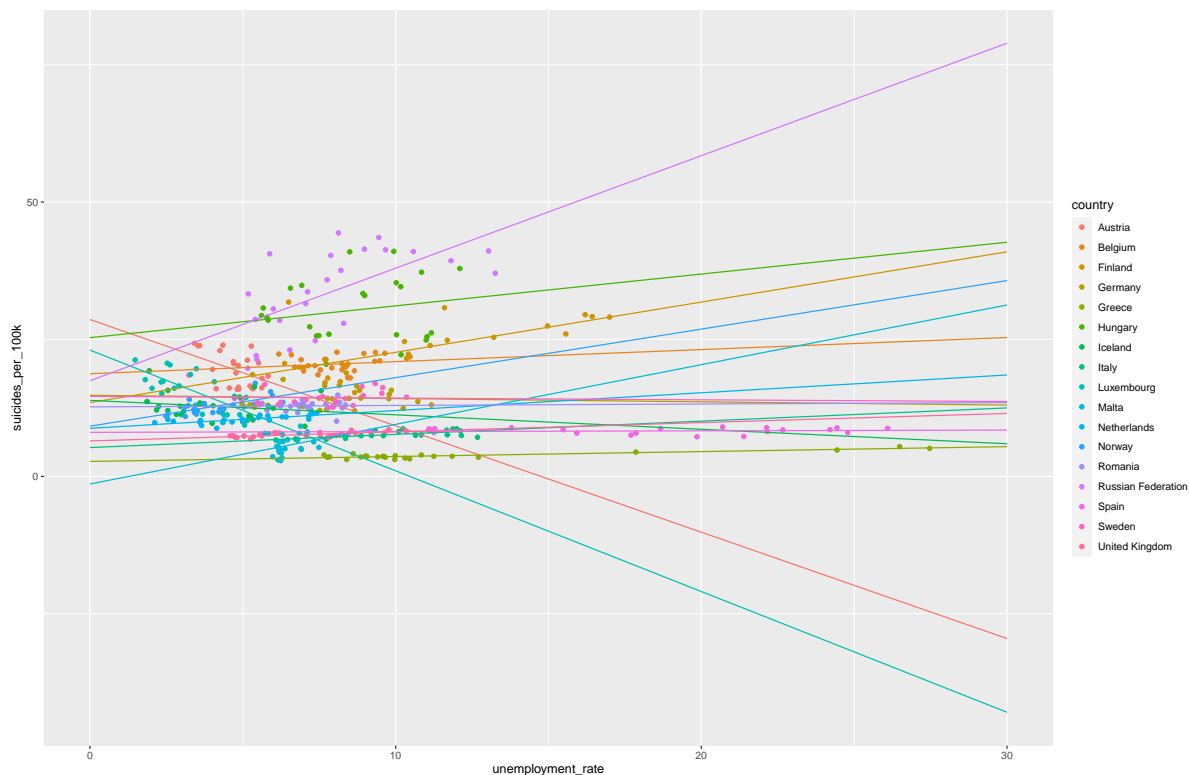
With the \hat{R} values are close to 1 and below the recommended value of 1.05, we can say that the chains are indeed well converged.



The effective sample size (ESS) measures the sample size needed to achieve the same level of precision for a similarly sampled set. In other words, it measures the quality of our estimation for parameters. Many of the ESS are larger than the actual number of samples (4000), which means that it is likely there are negative estimated autocorrelation in the samples.

Posterior predictive check

We can draw the fitted line onto the data points to assess the posterior predictive of the model. In this part, we only choose the median value of each β_0 and β_1 to make the figure looks a bit nicer.



The plot can be a bit confusing to look at, so Figure 1 shown the plot that are splitted into smaller subplots

It can be seen from Figure ?? and Figure 1 that the line seems to be a good fit for most of the countries.

Another method to assess the predictive performance of the model is to perform leave-one-out cross-validation (CV-LOO). A visualization of the \hat{k} values can be seen in Figure 2,

As can be seen in the PSIS diagnostic figure, some of the \hat{k} values are higher than 0.7. This indicate that our model is likely to be optimistic.

Predictive performance assessment

Next, let us predict the suicide rate in 2015. To get the value for β_1 and β_0 , we simply take the mean of the corresponding sample group.

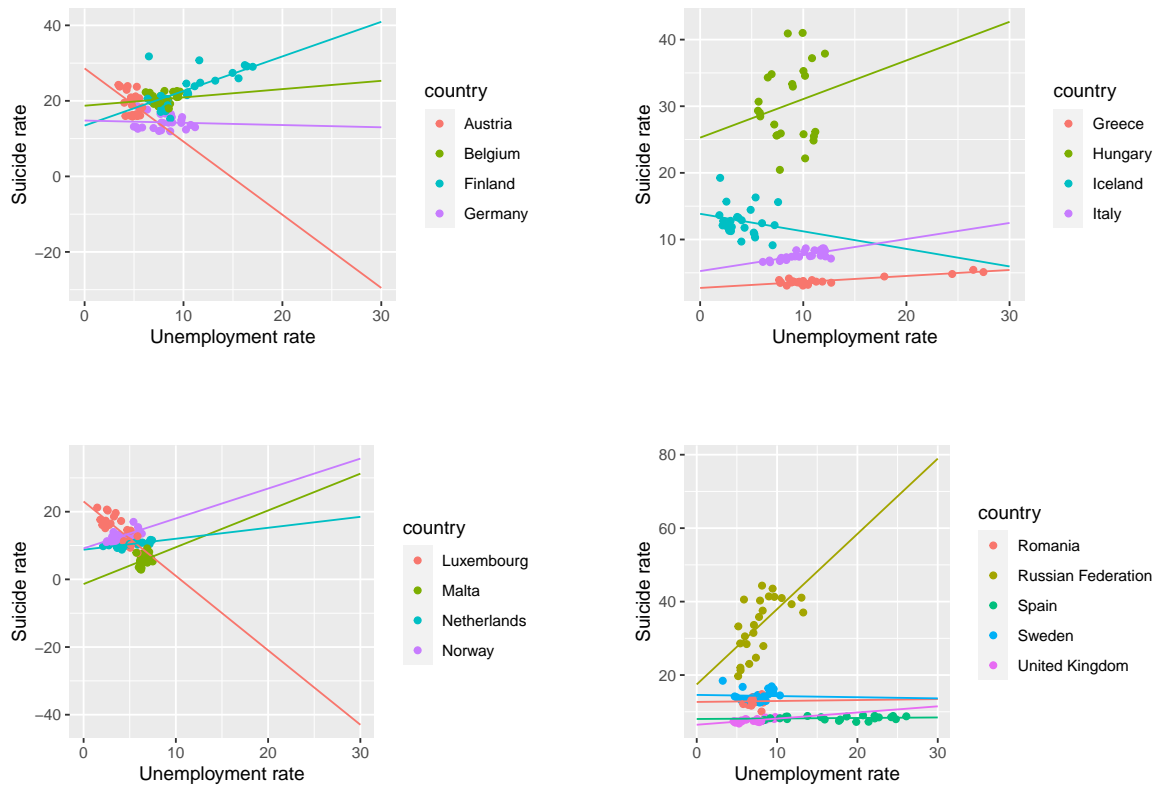


Figure 1: Subplots of the fitted posterior

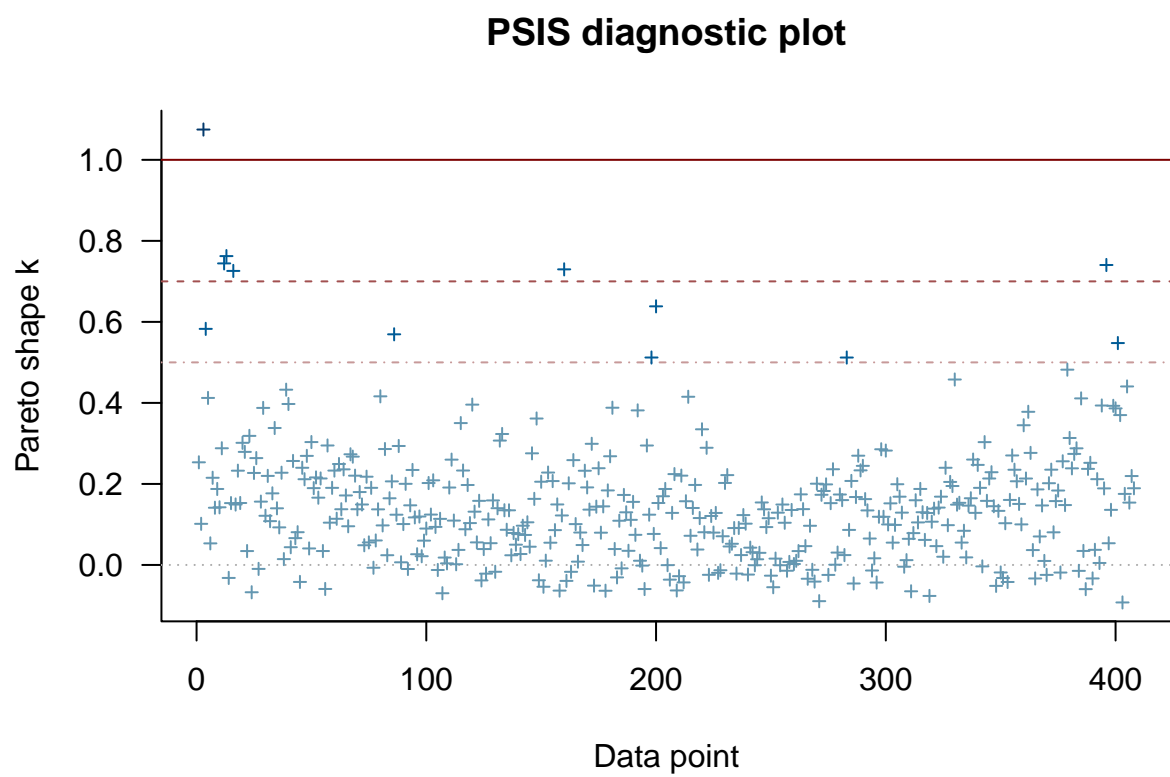


Figure 2: Visualization of the Pareto k values



From the plot, we can see that many of the predicted data points are far from their true value. To get a better picture, let us calculate the RSME of the prediction:

```
rmse(data_valid$suicides_per_100k, data_valid$pred_sep)
```

```
## [1] 4.249983
```

Given the scale of the data, an RMSE of 4 is quite large. One reason we can think of is the number of validation data. For each of the country, we only have one validation data point. Thus, this error might not convey the complete message.

Prior sensitivity analysis

To wrap up the analysis of the separate model, let us do a sensitivity analysis. We can make a new separate model by fitting a very weak prior to the parameters. In more details, we will change the priors for β_0 and β_1 to $N(0, 200)$, and the σ parameters to $Half - Cauchy(0, 50)$. Below is the Stan code for the new model:

```
## data {
##   int<lower=0> N;          // number of observations
##   int<lower=0> J;          // number of countries
##   int<lower=0> id[N];      // group indicator
##   vector[N] y;
##   vector[N] x;            // Unemployment rate
## }
##
## parameters {
##   vector[J] beta0;
##   vector[J] beta1;
##   vector<lower=0>[J] sigma;
## }
##
## model {
##   // priors
##   for (j in 1:J){
##     beta0[j] ~ normal(0, 200);
##     beta1[j] ~ normal(0, 200);
```

```
##     sigma[j] ~ cauchy(0, 50);
##   }
##
##   // likelihood
##   for (n in 1:N) {
##     y[n] ~ normal(x[n]*beta1[id[n]] + beta0[id[n]], sigma[id[n]]);
##   }
## }
##
## generated quantities {
##   vector[N] log_lik;
##   for (n in 1:N) {
##     log_lik[n] = normal_lpdf(y[n] | x[n] * beta1[id[n]] + beta0[id[n]], sigma[id[n]]);
##   }
## }

sm_sep_2 <- stan(file='stan_model/separate_2.stan', data=stan_data, iter=2000, refresh=0)
```

We compare the two separate models using `loo_compare`:

```
##           elpd_diff se_diff
## separate_1  0.0         0.0
## separate_2 -1.1         1.2
```

Since the `elpd_diff` and `se_diff` between the original and second separate model are very small, we conclude that our model is relatively not sensitive to prior choices.

Hierarchical model

Model description

We want to go a bit further with our model. It is quite natural to think that the slopes and intercepts of different countries can come from a common population distribution. For that reason, we decided to fit a hierarchical model to our data set.

With visual assessment of the data, it seems that all of the countries share the same intercept. That is why we only impose a hierarchical structure on the slope parameter.

We use the same reasoning as above to obtain a prior $N(1.89, 20)$ for intercept β_0 and a *Half-Cauchy*(0, 10) prior for the variance σ .

For the slope β_1 , this time, we want to have a prior as follow:

$$\beta_{1_j} \sim N(\mu_{1_j}, \sigma_{1_j})$$

Similar to before, we believe the mean of the β_1 should be centered around 0. Thus, the parameter μ_{1_j} should also be centered around 0. We also select the standard deviation value of 30 for μ_{1_j} to reflect that we do not have much information. For σ_{1_j} , we select a *Half-Cauchy*(0, 10) distribution as a weakly informative distribution.

The complete hierarchical model:

$$\begin{aligned}\mu_{1_j} &\sim N(0, 10) \\ \sigma_{1_j} &\sim \text{Half-Cauchy}(0, 10) \\ \beta_{1_j} &\sim N(\mu_{1_j}, \sigma_{1_j}) \\ \beta_0 &\sim N(1.89, 20) \\ \sigma &\sim \text{Half-Cauchy}(0, 10) \\ y_{ij} &\sim N(\beta_0 + \beta_{1_j} \cdot x_{ij}, \sigma)\end{aligned}$$

Stan code and running options

Below is the Stan code for the hierarchical model:

```
## data {
##   int<lower=0> N;           // number of observations
##   int<lower=0> J;           // number of groups
##   int<lower=0> id[N];       // group index
##   vector[N] x;             // explanatory variables
##   vector[N] y;             // response variables
## }
##
## parameters {
##   vector[J] beta1;         // The slope
##   real mu1;                // Mean hyperprior for beta1
##   real<lower=0> sigma1;     // Variance hyperprior for beta1
##
##   real beta0;              // The common intercept
##   real<lower=0> sigma;      // The common variance
## }
##
## model {
##
##   beta0 ~ normal(1.89, 20);
##   mu1 ~ normal(0, 10);
##   sigma1 ~ cauchy(0, 10);
##
##   sigma ~ cauchy(0, 10);
##
##   for(j in 1:J) {
##     beta1[j] ~ normal(mu1, sigma1);
##   }
##
##   for (n in 1:N) {
##     y[n] ~ normal(x[n]*beta1[id[n]] + beta0, sigma);
##   }
##
## }
##
## generated quantities{
##   vector[N] log_lik;
##   for (n in 1:N) {
##     log_lik[n] = normal_lpdf(y[n] | x[n] * beta1[id[n]] + beta0, sigma);
##   }
## }
```

Our setting to run the file was as below:

```
sm_hie <- stan(file='stan_model/linear_hierachical.stan', data=stan_data, iter=2000, refresh=0)
```

```
##  
## Divergences:  
## 0 of 4000 iterations ended with a divergence.  
##  
## Tree depth:  
## 0 of 4000 iterations saturated the maximum tree depth of 10.  
##  
## Energy:  
## E-BFMI indicated no pathological behavior.
```

According to the HMC diagnostics, none of the iterations resulted in divergence, nor saturated the maximum tree depth. In other words, the sampling behaves well.

Convergence diagnostics

It seems that there were no warnings of convergence for this model also. But just to be sure, we once again visualize the chains for β_0 and β_1 of Austria

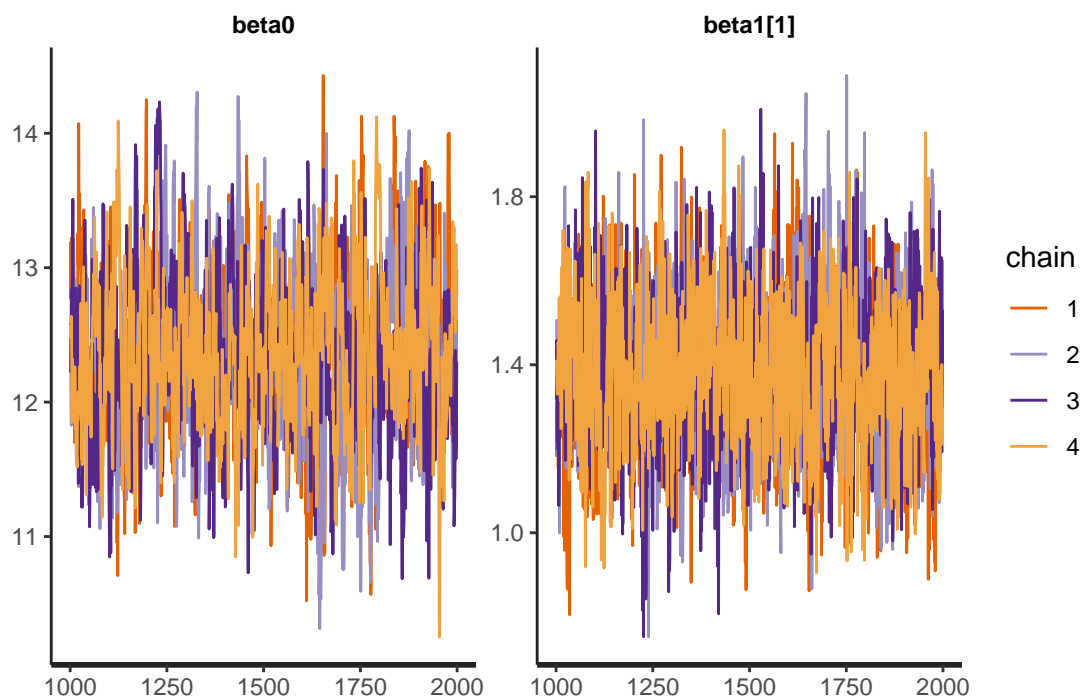
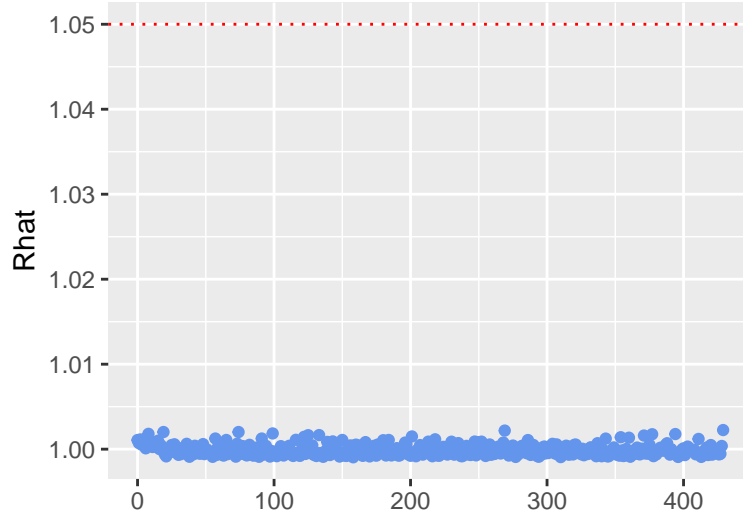
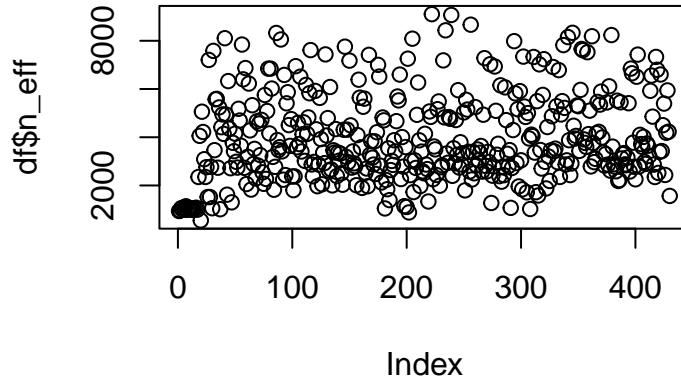


Figure 3: The chains of the intercept and the slope of Austria group

With visual inspection, it seems that the chains converged quite well. In some parts they behave differently, so the \hat{R} should be checked to detect any potential divergence



The \hat{R} values are all close to 1 and below the recommended value of 1.05. With these information, we can safely say that our model is converged well. To conclude our diagnostics, let's look at the effective sample size



The effective sample size (ESS) measures the sample size needed to achieve the same level of precision for a similarly sampled set. In other words, it measures the quality of our estimation for parameters. Many of the ESS are larger than the actual number of samples (4000), which means that it is likely there are negative estimated autocorrelation in the samples.

Posterior predictive check

We can draw the fitted line onto the data points to assess the posterior predictive of the model. Similar with the separate model, we only choose the median value β_0 and each β_1 to make the figure looks a bit nicer.

Again, let's break down the plot into four subplots of 4-5 countries:

It can be seen from Figure 4 and Figure 5 that the line seems to be a good fit for most of the countries.

Let's also perform leave-one-out cross-validation (CV-LOO). Below is the \hat{k} of the hierarchical model:

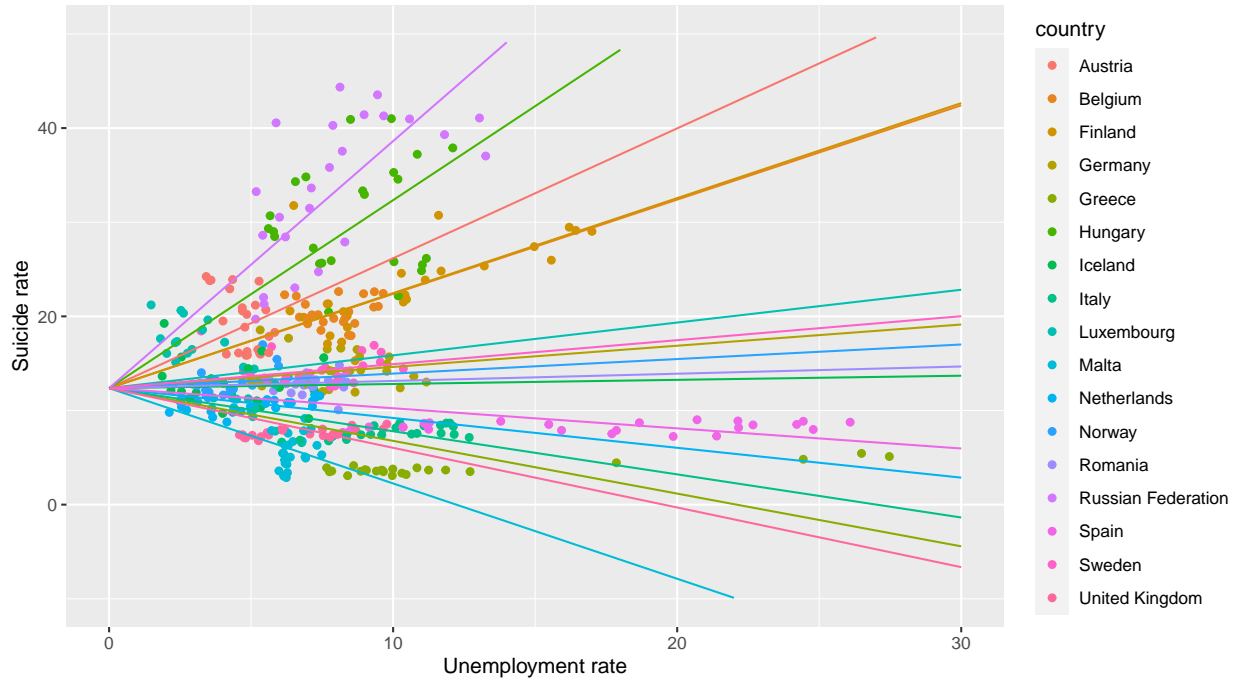


Figure 4: The fitting of the linear model to the data points

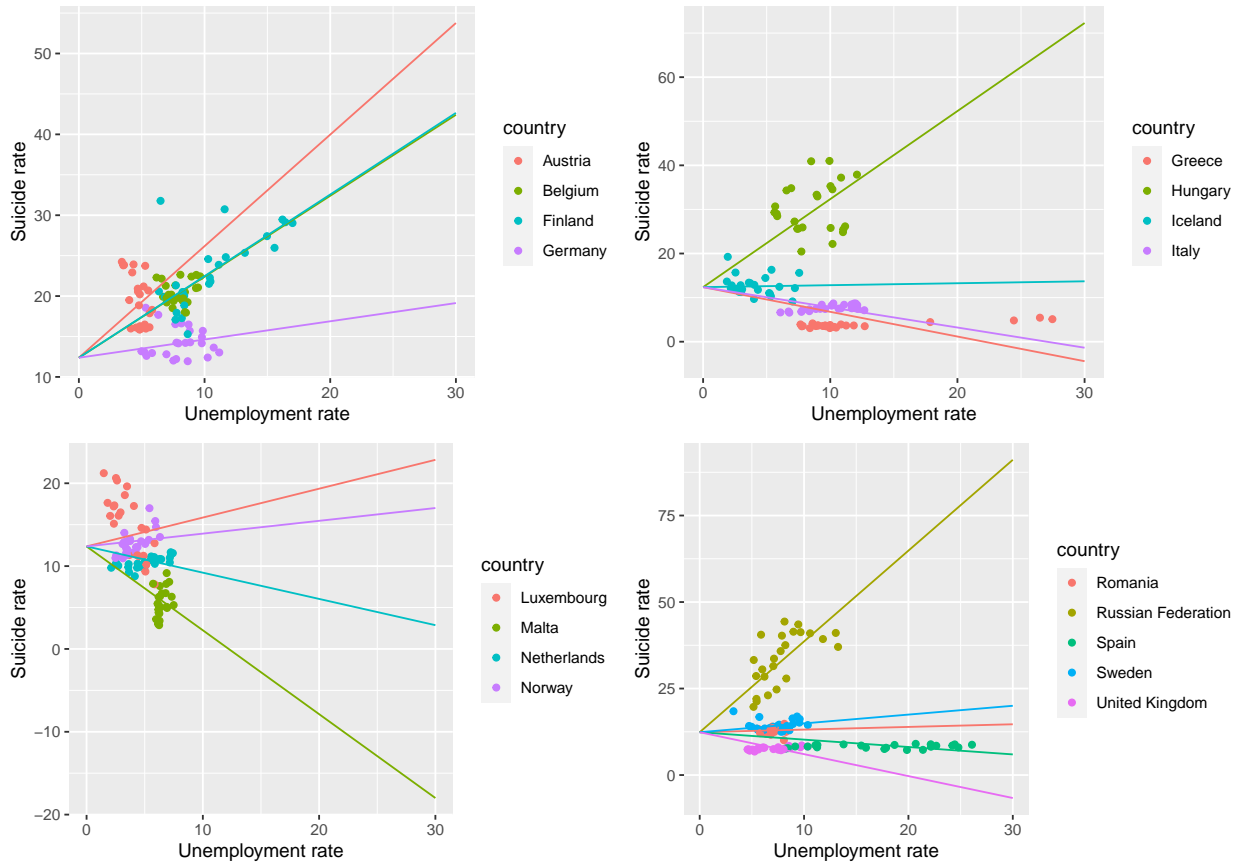


Figure 5: Subplots of the fitted posterior

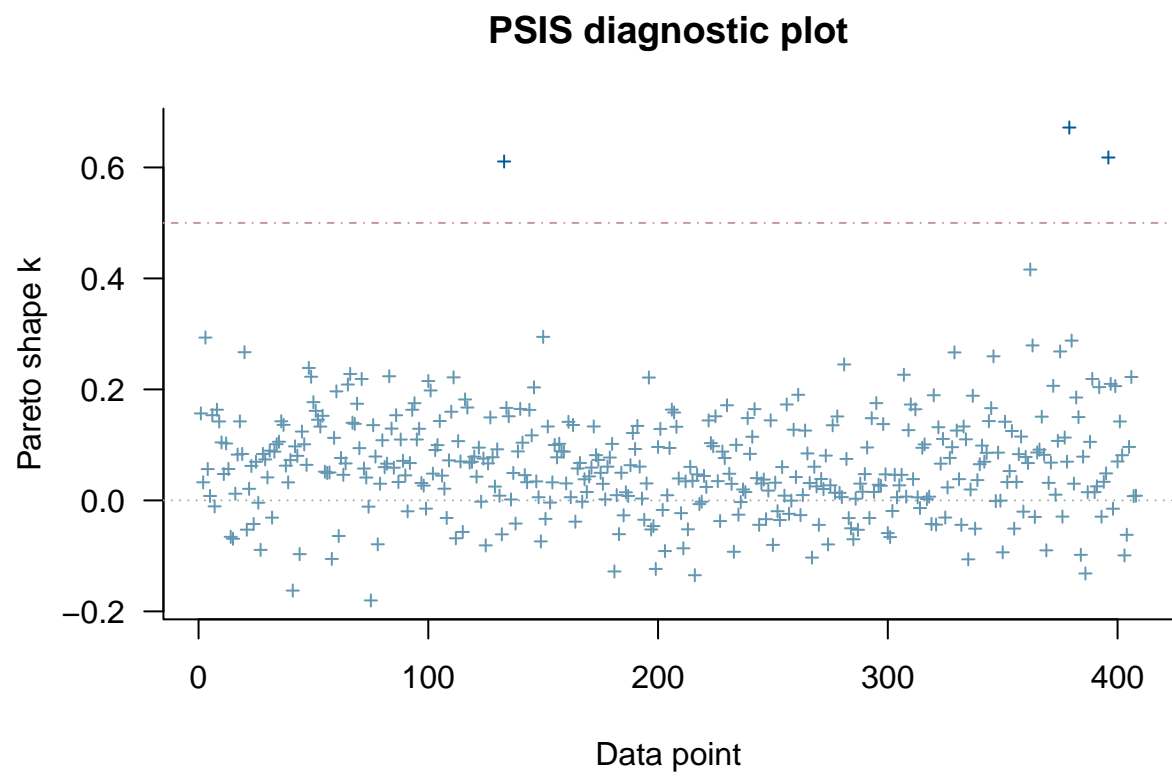
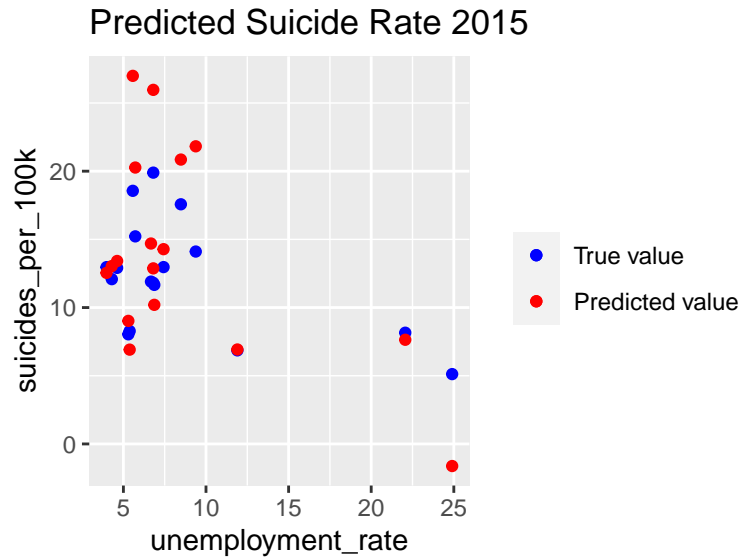


Figure 6: Visualization of the Pareto k values

As shown in the Figure 6, all \hat{k} value are under 0.7. This indicates that the model is reliable.

Predictive performance assessment

Let's try to visualize the prediction for year 2015. To get the value for β_1 and β_0 , we will just simply take the mean of the corresponding sample group



From the plot, we can see that for most of the data point, the predicted value seems a bit far away from the true value. But we can check a bit closer using RMSE:

```
rmse(data_valid$suicides_per_100k, data_valid$pred_hie)
```

```
## [1] 3.957917
```

We can see that the RMSE here is still large, but it is a bit smaller than the separate model. Again, we hypothesized that one reason for this is due to the lack of validating samples.

Prior sensitivity analysis

Again, to wrap up the analysis of the hierarchical model, let us do a prior sensitivity analysis. Again, let us fit a very weak priors to all of the parameters. For μ_1 and β_0 , we will replace the old prior with a $N(0, 200)$ distribution, and for all of the σ_1 and σ , I will replace them with a $Half - Cauchy(0, 50)$. Below is the Stan code.

```
## data {
##   int<lower=0> N;           // number of observations
##   int<lower=0> J;           // number of groups
##   int<lower=0> id[N];       // group index
##   vector[N] x;             // explanatory variables
##   vector[N] y;             // response variables
## }
##
## parameters {
##   vector[J] beta1;         // The slope
##   real mu1;                // Mean hyperprior for beta1
##   real<lower=0> sigma1;     // Variance hyperprior for beta1
##
##   real beta0;              // The common intercept
```

```

##   real<lower=0> sigma;           // The common variance
## }
##
## model {
##   beta0 ~ normal(0, 200);
##   mu1 ~ normal(0, 200);
##   sigma1 ~ cauchy(0, 50);
##   sigma ~ cauchy(0, 50);
##
##   for(j in 1:J) {
##     beta1[j] ~ normal(mu1, sigma1);
##   }
##
##   for (n in 1:N) {
##     y[n] ~ normal(x[n]*beta1[id[n]] + beta0, sigma);
##   }
## }
##
## generated quantities{
##   vector[N] log_lik;
##   for (n in 1:N) {
##     log_lik[n] = normal_lpdf(y[n] | x[n] * beta1[id[n]] + beta0, sigma);
##   }
## }

sm_hie_2 <- stan(file='stan_model/linear_hierachical_2.stan', data=stan_data, iter=2000, refresh=0)

```

We compare the two hierarchical models using `loo_compare`:

```

log_lik_hie2 <- extract_log_lik(sm_hie_2, merge_chains = FALSE)
r_eff_hie2 <- relative_eff(exp(log_lik_hie2), cores=4)
loo_hie2 <- loo(log_lik_hie2, r_eff=r_eff_hie2, cores=4)
loo_compare(list('hierarchical_1'=loo_hie, 'hierarchical_2'=loo_hie2))

```

```

##               elpd_diff se_diff
## hierarchical_2  0.0         0.0
## hierarchical_1 -0.4         0.2

```

Since the `elpd_diff` and `se_diff` between the original and second separate model, this suggests that our model is not sensitive to changes in priors.

Model comparison

From previous parts, it can be seen that the hierarchical model seems to be a better choice. It has a slightly smaller RMSE when predicting new values, and also less susceptible to optimistic due to PSIS analysis. But to be sure, we will perform a model comparison with `loo_compare`

```

loo_compare(list('separate'=loo_sep, 'hierachical'=loo_hie))

##               elpd_diff se_diff
## separate       0.0         0.0
## hierachical -322.4       23.3

```

Pairwise comparisons between each model and the model with the largest elpd, in this case the separate model,

is made. Since for the hierarchical and separate model $|\text{elpd_diff}/\text{se_diff}| > 5$, the models are well-specified. This is a surprise for us and contradicts with our expectation that the hierarchical model is a better fit. It seems that according to PSIS-LOO, the best model is the separate model. However, one thing to be noted is that some of the \hat{k} values of the separate model are less than 0.7. Thus, this can suggest that our error estimate is optimistic according to Vehtari et al. (2015).

Discussion

During the course of this project, we noticed that there were some issues regarding our analysis. The most prominent problem is that our data does not cover all of Europe. Since the model we built are either separate (the first model) or a hierarchical structure (the second model), this makes the countries that are not included in the dataset cannot be predicted by our models. In addition, the dataset we used is only ranging from 1991 to 2015, and it is very natural to expect that the suicide pattern can change after these period. This is especially true considering the COVID-19 pandemic, which leads to an unprecedented unemployment rate and suicide rate.

Going deeper into the models. Some of the prior parameters for the separate model are based on educated guess, instead of conducting formal research. This goes the same for the distribution used. We mostly used Normal and Cauchy distribution because they are simple to work with. This might also be one of the causes that lead to some of the \hat{k} values fall into the **bad** and **very bad** category.

The same issues can be said about the hierarchical model. We only expand on what we have known in the separate priors. This might also leads to the surprising results observed in the Model Comparison section. While it is true that some of the \hat{k} values of the separate model are very large, the percentage of them are only about 1% (At the time the script is ran). There is also the issue in which we choose a common β_0 values in the hierarchical model. On retrospective, it might be more logical if we also imposed a hierarchical structure onto β_0 . In our best interest, we ran a very quick test with a model that has hierarchical structures on both β_0 and β_1 and compared the `elpd` with the separate model. Although the `elpd_diff` was still very large, it was noticeably smaller compared to the hierarchical model with common β_0 . Finally, it is also worth to note that both models were not very good when predicting new data, though this can be due to the lack of validation data for each group.

For future improvement of the project, we can collect more recent data and data that cover more countries than the current dataset. It is also worthwhile to conduct more literature review so we might find potentially better priors. Last but not least, we can incorporate more covariates to the model. The current model only predict using unemployment rate, but we can add gdp per capita or HDI value of the countries to the analysis.

Conclusion of the results

To recap the progress and results of the project, we have fitted a linear regression model with the aim to predict suicide rates using unemployment rates. We have tried two different approaches: first we treat each countries as separated groups, and then we tried to add a hierarchical structure to the countries. The results of the analysis showed that with the chosen priors, the models can well converged, and it seems that the posterior fit were quite reasonable. However, when doing more analysis, it was revealed to us that both model were not very good when given new data to predict. The hierarchical model performed slightly better, but this difference can be minor. We also noticed that in term of PSIS-LOO performace the separate model has a significant difference in term of `elpd` compared to the hierarchical model. But it is also worth to note that about 1% of the \hat{k} values of the separate model are greater than 1. So if we only choose the model based on `elpd`, the separate model seems to be the better model, although it is likely to be optimistic. Explain in words, the results of this analysis tell us that when predicting the suicide rates using unemployment rates, it would be better if we treat the countries as separated groups.

Reflection

Throughout this project we were able to combine concepts taught in the course and apply them to real world data, which enables us to get a hang of the workflow of Bayesian data analysis. One of the main challenges encountered at the start of the project is choosing a suitable dataset. Complete information is not always possible and pre-processing can be the most laborous part if the dataset is not chosen carefully. Under the scope of this project, we chose the dataset that seems to require the minimum amount of preprocessing so that we could focus more on the Bayesian analysis part.

The other main challenge we encountered is choosing a set of models to apply on our data. The amount of practice from the course assignment for defining suitable model for the data at hand is limited. Through this project and help from our dedicated teaching assistants, we were able to experience various ways to look at our data and new methods not covered in depth in the course, such as centered parameterization. There remains a lot to be explored in Stan and in Bayesian analysis. Furthermore, we were able to practice a lot with using visualizations, especially on ggplot2, to make posterior and predictive checks.

References

- Fountoulakis, Konstantinos N, Wolfram Kawohl, Pavlos N Theodorakis, Ad JFM Kerkhof, Alvydas Navickas, Cyril Höschl, Dusica Lecic-Tosevski, et al. 2014. “Relationship of Suicide Rates to Economic Variables in Europe: 2000–2011.” *The British Journal of Psychiatry* 205 (6): 486–96.
- Vehtari, Aki, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. 2015. “Pareto Smoothed Importance Sampling.” *arXiv Preprint arXiv:1507.02646*.