STAT 578 – Advanced Bayesian Modeling – Spring 2020

Data Analysis Report | 5/4/2020 | Tao Li (taol4)

Introduction

COVID-19 (2019 Novel Coronavirus) [1] is a coronavirus identified as the cause of an outbreak of respiratory illness first detected in Wuhan, China. As of May 5th, 2020, COVID-19 pandemic has been confirmed in **3 595 662** cases and **247 652** deaths worldwide of **215** Countries and Territories, **1,171,185** cases and **62698** deaths in United States [2].

COVID-19 start to spread in US at the middle of March (2234 cases on March 15), the pandemic spread very fast reached 163,539 cases by end of March [3]. The outbreak has since spread to all the states, slowed down the spreading since middle of April following by the Social Distancing restrictions and Stay-at-home order [4].

This analysis is working on the Bayesian Modeling of COVID-19 deaths using the daily deaths data from the European Center for Disease Prevention and Control (ECDC)[5].

Data

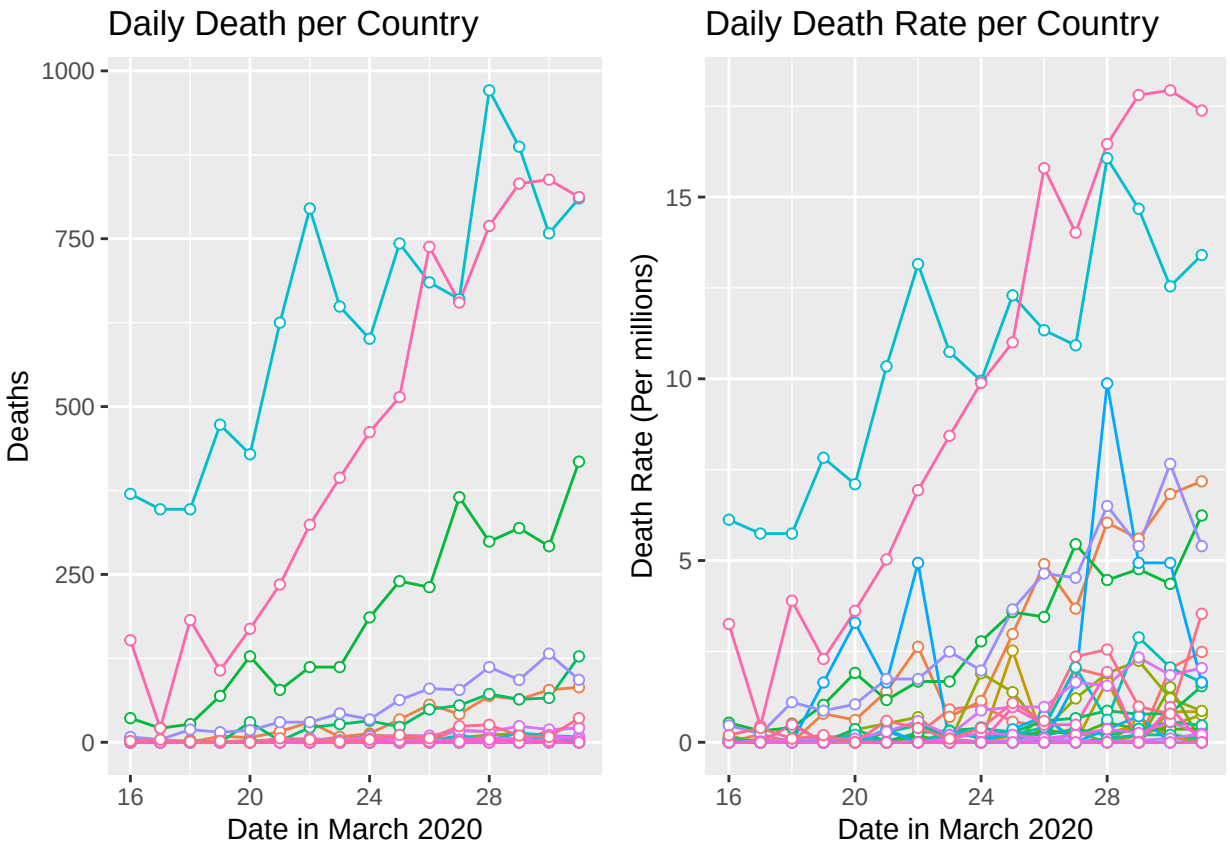
The data file **EUCOVIDdeaths.csv** contains data from the European Center for Disease Prevention and Control (ECDC) on daily deaths related to COVID-19 in the 27 European Union (EU) member states for the second half of March 2020. Each row represents an EU country, and the 18 columns are as follows:

- **Country** name of the country (member state)
- **PopulationM** 2018 population of the country, in millions
- Mar16–Mar31 number of COVID-19 deaths recorded in the country for that date

Country	PopulationM	Mar16	Mar17	Mar18	Mar19	Mar20	Mar21	Mar22	Mar23
Austria	8.847	0	2	0	1	2	0	2	8
Belgium	11.422	0	1	0	9	7	16	30	8
Bulgaria	7.024	0	0	0	0	1	0	0	0
Croatia	4.089	0	0	0	0	0	0	0	0
Cyprus	1.189	0	0	0	0	0	0	0	0

Let's analyze and compare the daily deaths and death rate per country and come out some insights.

- Which country has the most deaths over this period? **Italy**
- Which country has the highest death rate per capita over this period? **Italy**
- Which countries have no deaths over this period? **Latvia, Malta, and Slovakia**



First Model

(a) Let response variable $y_{i,j}$ be number of COVID-19 deaths in coutry $i (i = 1, ..., 27)$ on day j ($j = 1$ is March 16, $j = 2$ is March 17, etc.). The 1st model is a **Poisson loglinear regression** on population of coutry and day.

$$y_{i,j}|r_{i,j} \sim indep.Poisson(r_{i,j})$$
$$log(r_{i,j}) = p_i + \beta_i^{intercept} + \beta^{slope} * d_j$$

Where:

- p_i = natura logarithm of the 2018 population of country i
- d_j = day index centered so that this variable has an average of zero

Consider the prior: $\beta^{slope} \sim iid \quad N(0, 100^2) \quad \beta_i^{intercept} \sim iid \quad N(\mu_{intercept}, \sigma^2_{intercept})$

With: $\mu_{intercept} \sim iid \quad N(0, 100^2) \qquad \sigma_{intercept} \sim \quad U(0, 100)$

- The model is a rate model, the rate is the p_i.
- The parameters are: $\beta_i^{intercept}$ and β^{slope}
- The hyperparameters are: $\mu_{intercept}$ and $\sigma_{intercept}$
- The intercept $\beta_i^{intercept}$ varies by country
- The slope β^{slope} is consistant for all countries.

(b) Here's the JAGS model in the first model.

```
model {
  for (i in 1:length(logpopulation)) {
    for (j in 1:length(daycent)) {
      deaths[i,j] ~ dpois(lambda[i,j])
      log(lambda[i,j]) <- logpopulation[i] + intercept[i] + slope*daycent[j]
    }

    intercept[i] ~ dnorm(mu.intercept, 1/sigma.intercept^2)
  }

  slope ~ dnorm(0, 1/100^2)

  mu.intercept ~ dnorm(0, 1/100^2)
  sigma.intercept ~ dunif(0, 100)
}
```

And the data-related nodes are as follows:

- ($deaths[i, j]$ is the $y_{i,j}$: number of COVID-19 deaths recorded in country $i(i = 1, ..., 27)$ on day j.
- $logpopulation[i]$ is p_i: natural logarithm of the 2018 population of country i, where population is expressed in millions
- $daycent[j]$ is d_i. : day index centered so that this variable has an average of zero (but is not standardized); the difference between $daycent[j + 1]$ and $daycent[j]$ should equal 1

(c) The model used 4 chains, with 4000 iterations burn-in and sampled 2000 iteration after checking the convergence.

And got enough effective sample size of 6944 for mu.intercept, 2168 for sigma.intercept, and 3794 for slope which is significant enough for the sampling.

(Convergence with Plot Trace and Gelman And Rubin's Convergence Diagnostic are included in te Appendix.)

(d) Here's the approximate posterior mean, posterior standard deviation, and 95% central posterior interval for each top-level (hyper)parameter.

hyperparameters	mean	standard_deviation	quantile_0.025	quantile_0.975
mu.intercept	-1.2299	0.3912	-2.0225	-0.4736
sigma.intercept	1.9603	0.3271	1.4449	2.6954
slope	0.1086	0.0016	0.1056	0.1117

(e) Based on the posterior median analysis.

- **Italy** has the highest posterior median itercept.
- **Slovakia** has the lowest median itercept.

(f) I approximated the value of (Plummer's) DIC and the associated effective number of parameters of **27** which is close to the actual number of 28 parameters (27 for countries, 1 for slope).

Second Model

The second model is based on the first model but allows each country to have a separate slope:

$$log(r_{i,j}) \quad = \quad p_i + \beta_i^{intercept} + \beta_i^{slope} * d_j$$

Consider the prior: $\beta_i^{intercept} \sim iid \quad N(\mu_{intercept}, \sigma^2_{intercept}) \qquad \beta_i^{slope} \sim iid \quad N(\mu_{slope}, \sigma^2_{slope})$ With:

- $\mu_{intercept} \sim iid \quad N(0, 100^2) \quad \sigma_{intercept} \sim \quad U(0, 100)$
- $\mu_{slope} \sim iid \quad N(0, 100^2) \quad \sigma_{slope} \sim \quad U(0, 100)$

Here's the summary to the second model:

- The parameters are: $\beta_i^{intercept}$ and β_i^{slope}
- The hyperparameters are: $\mu_{intercept}$, $\sigma_{intercept}$, μ_{slope}, σ_{slope}
- The intercept $\beta_i^{intercept}$ varies by country
- The slope β_i^{slope} varies by country, which is different to the first model.

(a) Here's the JAGS model in the second model.

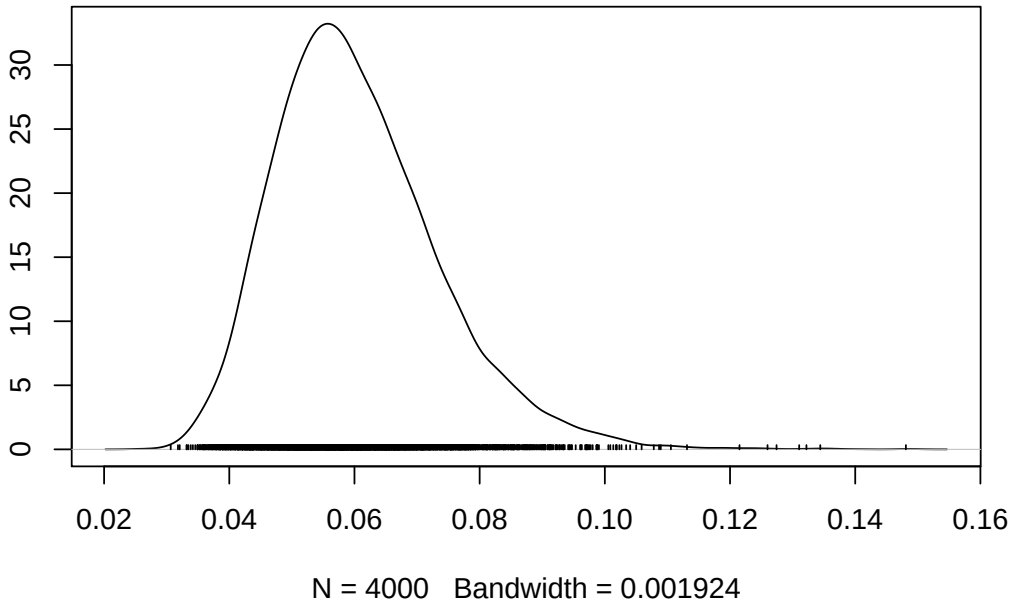
```
model {

  for (i in 1:length(logpopulation)) {
    for (j in 1:length(daycent)) {
      deaths[i,j] ~ dpois(lambda[i,j])
      log(lambda[i,j]) <- logpopulation[i] + intercept[i] + slope[i]*daycent[j]
    }

    intercept[i] ~ dnorm(mu.intercept, 1/sigma.intercept^2)
    slope[i] ~ dnorm(mu.slope, 1/sigma.slope^2)
  }

  mu.intercept ~ dnorm(0, 1/100^2)
  sigma.intercept ~ dunif(0, 100)

  mu.slope ~ dnorm(0, 1/100^2)
  sigma.slope ~ dunif(0, 100)
}
```

(b) The model used 4 chains, with 2000 iterations burn-in and sampled 4000 iteration after checking the convergence.

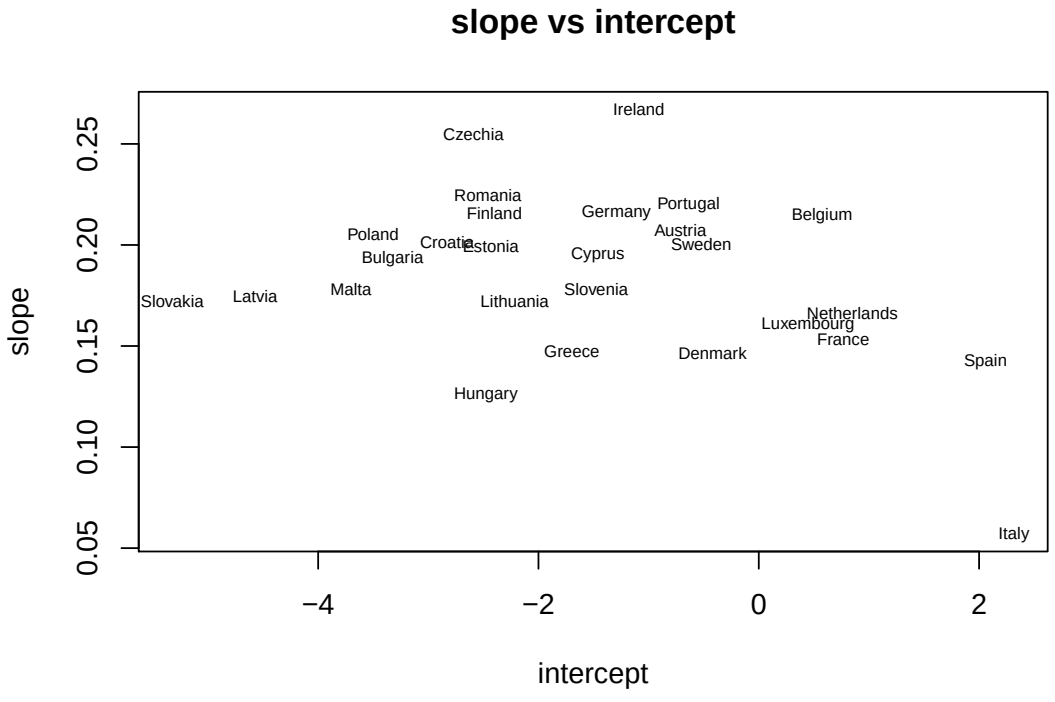
And got enough effective sample size of 12302 for mu.intercept, 4911 for sigma.intercept, 4309 for mu.slope, 2221 for sigma.slope which is significant enough for the sampling.

(Convergence with Plot Trace and Gelman And Rubin's Convergence Diagnostic are included in te Appendix.)

(c) Below is the (estimated) posterior density of σ_1 for slope. This shows that the slopes of different countries are different.



(d) For each country, let's compute an approximate posterior expected intercept and an approximate posterior expected slope, then plot these pairs on a scatterplot of slope versus intercept.



It's interesting to see that **Italy** has the lowest slope compared to other countries. But that also make sense

as COVID-19 started in Italy earlier, and already slowing down the death rate in this time range. However, for most of other countries, the COVID-19 just started which has a higher slope but also lower intercept.

(e) I approximated the value of (Plummer's) DIC and the associated effective number of parameters of **42** (The actual parameters should be 54: 27 for slope and 27 for intercept of each country.)

The new model has a much smaller DIC (2590) compared to the first model (3715), so the new model is much better, which also make sense as the new model allows different slope per market for accurate modeling.

Conclusions

The rapid spread of a novel coronavirus, declared a pandemic by the World Health Organization (WHO), continues to spread and force countries to take drastic action — including closing borders — to prevent the spread of the deadly virus. In this paper, the Bayesian modeling is being used to analyze one important task of dealths using data from ECDC.

In the Bayesian modeling, we usually formulate linear regression using probability distributions rather than point estimates. The aim of Bayesian Linear Regression is not to find the single "best" model parameters, but rather to determine the reasonible posterior distribution for the model parameters. Not only is the response generated from a probability distribution, but the model parameters are assumed to come from a distribution as well.

In this analysis using the COVID-19 death data from ECDC is a very good exercise to modeling the COVID deaths using the features of countries and dates. I first started the data analysis on the deaths and death rate per coutries. And experimented 2 models with the deaths per country and date as response variable on features of natual logarithms of population of country and centered day.

- First Model: A model allowing each country to have different intercept and share a same slope.
- Second Model: A improved model allowing each country to have different intercept and slope.

Upon these two models, more detailed analysis was continued to analyze the parameter and hyperparameters behind the model. Overall, the second model perform better to this case which lower DIC and a more flexible model allowing the different slope per country.

In the next stpes, more analysis and modeling will be applied on other important factors or features. I also hope this analysis and modeling can be used to predict in other countries and United States.

Appendix

```
options(scipen = 1, digits = 4, width = 80, fig.align = "center")
############################## Code Chunk: Data ##############################
# Data: Read Data
# Read Data from EUCOVIDdeaths.csv, and slow some examples.

library("knitr")
eucoviddeaths <- read.table("EUCOVIDdeaths.csv",quote = "\"'", sep = ",",
                            check.names=FALSE,header = TRUE, stringsAsFactors = TRUE)
kable(eucoviddeaths[1:5, 1: 10])
# Data: Plot analysis
# Plot the daily deaths and death rate for comparision per country.
# Use the package of ggplot2 and reshape2.
```

```r
library("ggplot2")
library(reshape2)
require(gridExtra)
mdf <- melt(eucoviddeaths, id.vars=c("Country","PopulationM"), variable_name = "Year",
            value.name="Deaths", variable.name="Date")
mdf[,3] <- as.numeric(sub("^Mar", "", mdf[,3]))
mdf$DeathRate = mdf$Deaths/mdf$PopulationM

plot1 <- ggplot(data=mdf, aes(x=Date, y=Deaths, group = Country, color = Country))+
         geom_line() + geom_point(size=1.5, shape=21, fill="white")+
         xlab("Date in March 2020") + ylab("Deaths")+
         ggtitle("Daily Death per Country") +
         theme(legend.position="none")

plot2 <- ggplot(data=mdf, aes(x=Date, y=DeathRate, group = Country, color = Country))+
         geom_line() + geom_point(size=1.5, shape=21, fill="white")+
         xlab("Date in March 2020") + ylab("Death Rate (Per millions)")+
         ggtitle("Daily Death Rate per Country") +
         theme(legend.position="none")

grid.arrange(plot1, plot2, ncol=2)
# Data: Analysis
# Data analysis to get the country with max deaths, min deaths and max death rate.
aggdata<- aggregate(Deaths ~ Country, data = mdf,mean)
head(aggdata[order(aggdata$Deaths),])
head(aggdata[order(-aggdata$Deaths),])

aggdata<- aggregate(DeathRate ~ Country, data = mdf,mean)
head(aggdata[order(-aggdata$DeathRate),])


############################## Code Chunk: First Model ##############################


model {
  for (i in 1:length(logpopulation)) {
    for (j in 1:length(daycent)) {
      deaths[i,j] ~ dpois(lambda[i,j])
      log(lambda[i,j]) <- logpopulation[i] + intercept[i] + slope*daycent[j]
    }

    intercept[i] ~ dnorm(mu.intercept, 1/sigma.intercept^2)
  }

  slope ~ dnorm(0, 1/100^2)

  mu.intercept ~ dnorm(0, 1/100^2)
  sigma.intercept ~ dunif(0, 100)
}
# First Model:
# Init the data
#  logpopulation | daycent | deaths
d1 <- list(logpopulation = log(eucoviddeaths$PopulationM),
```

```
           daycent = c(1:16) - 8.5,
           deaths = eucoviddeaths[,3:18])

inits1 <- list(list(mu.intercept = 20, sigma.intercept = 0.01, slope = 20),
               list(mu.intercept = 20, sigma.intercept = 90, slope = -20),
               list(mu.intercept = -20, sigma.intercept = 0.01, slope = 20),
               list(mu.intercept = -20, sigma.intercept = 90, slope = -20))

# First Model:
# Now, let's load the model and burn in with 2000 iterations

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

# burn-in
update(m1, 4000)

# Sample data
x1 <- coda.samples(m1, c("lambda","intercept","slope","mu.intercept","sigma.intercept"),
                   n.iter=2000)
# Convergence Check: Trace Plot

plot(x1[,c("mu.intercept","sigma.intercept","slope")],
     smooth=FALSE)
# Convergence Check: Gelman And Rubin's Convergence Diagnostic

gelman.diag(x1[,c("mu.intercept","sigma.intercept","slope")],
            autoburnin=FALSE)
#And effective size is:

effectiveSize(x1[,c("mu.intercept","sigma.intercept","slope")])
# Here's the coda summary of the results for the monitored regression coefficients
summary(x1[,c("mu.intercept","sigma.intercept","slope")])
# Analysis of approximate posterior mean, posterior standard deviation,
# and 95% central posterior interval for each top-level (hyper)parameter
col1 = c("mu.intercept","sigma.intercept","slope")
col2 = c(mean(as.matrix(x1)[,"mu.intercept"]),
         mean(as.matrix(x1)[,"sigma.intercept"]),
         mean(as.matrix(x1)[,"slope"]))

col3 = c(sd(as.matrix(x1)[,"mu.intercept"]),
         sd(as.matrix(x1)[,"sigma.intercept"]),
         sd(as.matrix(x1)[,"slope"]))

col4 = c(quantile(as.matrix(x1)[,"mu.intercept"], c(0.025)),
         quantile(as.matrix(x1)[,"sigma.intercept"], c(0.025)),
         quantile(as.matrix(x1)[,"slope"], c(0.025)))

col5 = c(quantile(as.matrix(x1)[,"mu.intercept"], c(0.975)),
         quantile(as.matrix(x1)[,"sigma.intercept"], c(0.975)),
         quantile(as.matrix(x1)[,"slope"], c(0.975)))

statdf = data.frame(hyperparameters = col1,
```

```
                    mean=col2,
                    standard_deviation=col3,
                    quantile_0.025=col4,
                    quantile_0.975=col5)
kable(statdf)
# posterior median analysis

intercept_median <- numeric(nrow(eucoviddeaths))

for(s in 1:nrow(eucoviddeaths)){
  intercept_median[s] = median(as.matrix(x1)[,paste("intercept[",s,"]", sep="")])
}

country_intercept = data.frame(Country = eucoviddeaths$Country,
                               intercept_median =intercept_median)
head(country_intercept[order(country_intercept$intercept_median),])

head(country_intercept[order(-country_intercept$intercept_median),])
# DIC Analysis
dic.samples(m1,10000)


############################## Code Chunk: Second Model ##############################


model {

  for (i in 1:length(logpopulation)) {
    for (j in 1:length(daycent)) {
      deaths[i,j] ~ dpois(lambda[i,j])
      log(lambda[i,j]) <- logpopulation[i] + intercept[i] + slope[i]*daycent[j]
    }

    intercept[i] ~ dnorm(mu.intercept, 1/sigma.intercept^2)
    slope[i] ~ dnorm(mu.slope, 1/sigma.slope^2)
  }

  mu.intercept ~ dnorm(0, 1/100^2)
  sigma.intercept ~ dunif(0, 100)

  mu.slope ~ dnorm(0, 1/100^2)
  sigma.slope ~ dunif(0, 100)
}
# Second Model:
# Init the data
d2 <- list(logpopulation = log(eucoviddeaths$PopulationM),
           daycent = c(1:16) - 8.5,
           deaths = eucoviddeaths[,3:18])

inits2 <- list(list(mu.intercept = 10, sigma.intercept = 0.01,
                    mu.slope = 10, sigma.slope = 90),
               list(mu.intercept = 10, sigma.intercept = 90,
                    mu.slope = -10, sigma.slope = 0.01),
```

```
              list(mu.intercept = -10, sigma.intercept = 0.01,
                   mu.slope = 10, sigma.slope = 90),
              list(mu.intercept = -10, sigma.intercept = 90,
                   mu.slope = -10, sigma.slope = 0.01))

# Second Model:
# Now, let's load the model and burn in with 2000 iterations

library(rjags)
m2 <- jags.model("secondmodel.bug", d2, inits2, n.chains=4, n.adapt=1000)
update(m2, 2000) # burn-in
x2 <- coda.samples(m2, c("intercept","slope","mu.intercept",
                         "sigma.intercept", "mu.slope","sigma.slope"),
                   n.iter=4000)
# Convergence Check: Trace Plot

plot(x2[,c("mu.intercept","sigma.intercept","mu.slope","sigma.slope")],
     smooth=FALSE)
# Convergence Check: Gelman And Rubin's Convergence Diagnostic
gelman.diag(x2[,c("mu.intercept","sigma.intercept","mu.slope","sigma.slope")],
            autoburnin=FALSE)
#And effective size is:

effectiveSize(x2[,c("mu.intercept","sigma.intercept", "mu.slope","sigma.slope")])
# Here's the coda summary of the results for the monitored regression coefficients
summary(x2[,c("mu.intercept","sigma.intercept", "mu.slope","sigma.slope")])
densplot(x2[,c("sigma.slope")])

# posterior mean analysis of intercept and slope

intercept_mean <- numeric(nrow(eucoviddeaths))
slope_mean <- numeric(nrow(eucoviddeaths))

for(s in 1:nrow(eucoviddeaths)){
  intercept_mean[s] = mean(as.matrix(x2)[,paste("intercept[",s,"]", sep="")])
  slope_mean[s] = mean(as.matrix(x2)[,paste("slope[",s,"]", sep="")])
}

country_stat = data.frame(Country = eucoviddeaths$Country,
                          intercept =intercept_mean,
                          slope= slope_mean)

#head(country_intercept[order(country_intercept$intercept_median),])
#head(country_intercept[order(-country_intercept$intercept_median),])
#country_stat


# For each country, let's compute an approximate posterior expected
# intercept and an approximate posterior expected slope,
# then plot these pairs on a scatterplot of slope (vertical axis) versus
# intercept (horizontal axis).

plot(x = intercept_mean,
```

```
     y = slope_mean,
     xlab = "intercept",
     ylab = "slope",
     main = "slope vs intercept",
     type ="n"
)

text(intercept_mean, slope_mean, eucoviddeaths$Country, cex=0.6)
# DIC Analysis
dic.samples(m2,10000)
# this R markdown chunk generates a code appendix
```

Application written in the R programming language [6].

REFERENCES

1. Coronavirus Disease 2019 (COVID-19). https://www.cdc.gov/coronavirus/2019-ncov/index.html
2. "Coronavirus Diseases 2019." 2020. https://www.who.int/emergencies/diseases/novel-coronavirus-2019.
3. Cases in Hte U.S.|cdc. 2020. https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/previouscases. html.
4. "Stay-at-Home Order." 2020. https://en.wikipedia.org/wiki/Stay-at-home_order.
5. "Geographic distribution of COVID-19 cases worldwide". https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide
6. "R: A Language and Environment for Statistical Computing." https://coronavirus-data. com/.