STAT 578: Data Analysis Report

John Mohoang (mohoang2)

5/2/2020

1. Introduction

Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2).[1] The disease was first identified in December 2019 in Wuhan, the capital of China's Hubei province, and has since spread globally, resulting in a coronavirus pandemic.[2] As of 3 May 2020, more than 3.44 million cases have been reported across 187 countries and territories, resulting in more than 244,000 deaths.[3]

This virus is primarily spread between people during close contact, often via small droplets produced by coughing, sneezing, or talking.[4] The droplets usually fall to the ground or onto surfaces rather than remaining in the air over long distances. A such, people become infected by touching a contaminated surface and then touching their face. It is most contagious during the first three days after the onset of symptoms, although spread may be possible before symptoms appear and in later stages of the disease.[5]

European countries have been hit much harder than Asian nations and have spread the virus significantly more than other regions.[6]. A major reason behind this is because European countries were late to close their air links, consequently facilitating the spread through air travel.

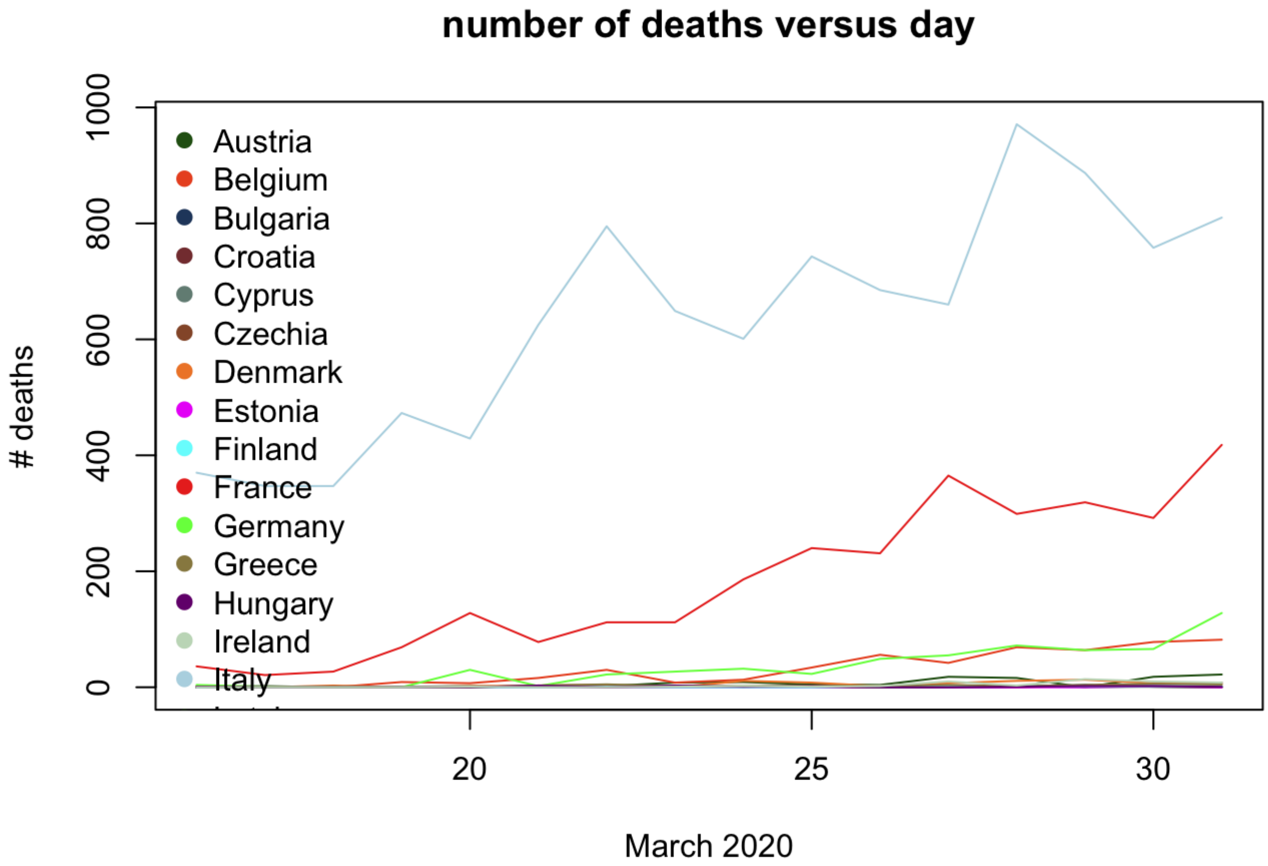
This study aims at examining the effects of COVID-19 in Europe, particularly the daily deaths caused by this disease.

2. Data

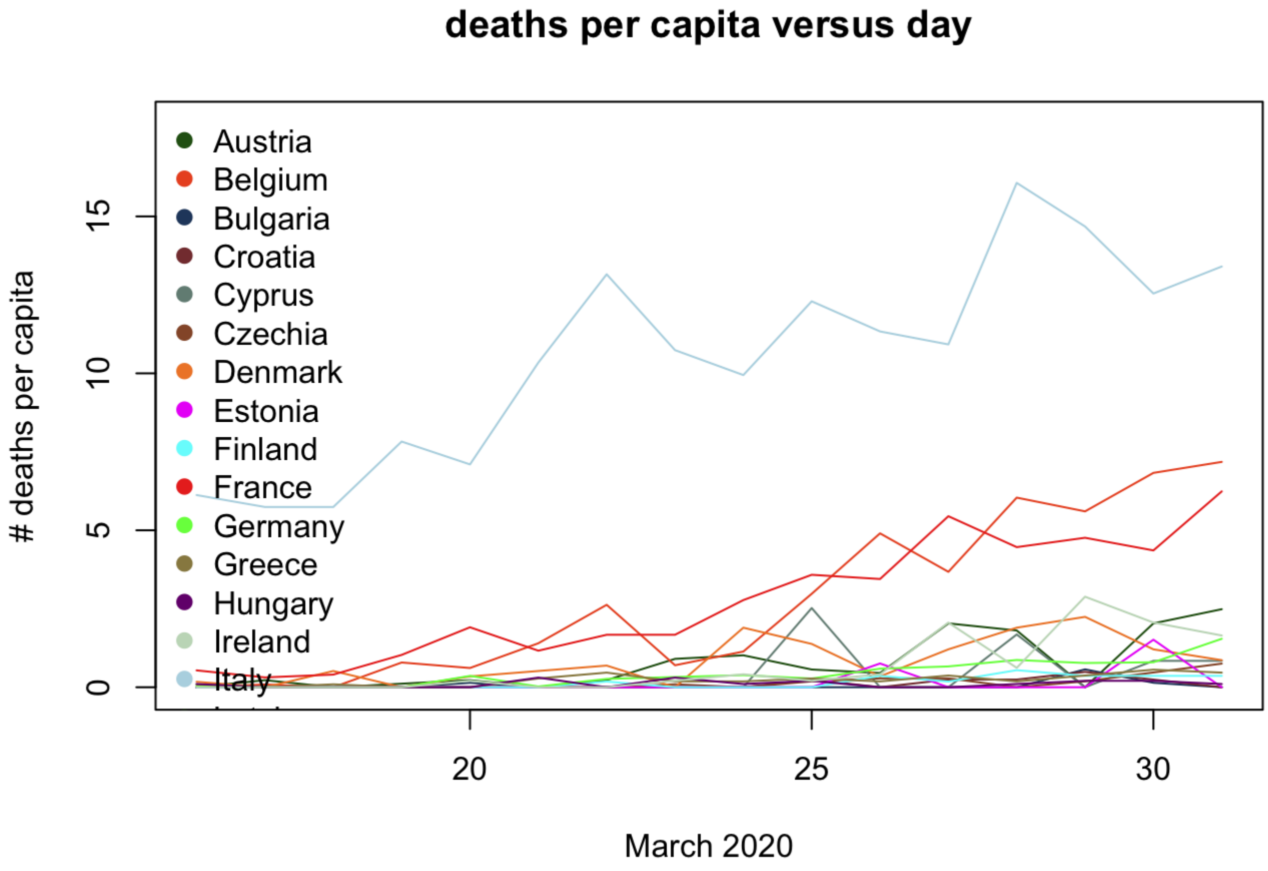
The data used in this study was obtained from the European Center for Disease Prevention and Control (ECDC) on daily deaths related to COVID-19 in the 27 European Union (EU) member states for the second half of March 2020.[7] Each row represents an EU country, and the columns are as follows:

- $Country$: 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

To visualize the data used in this study, there are two plots, the **number of deaths versus day** plot and the **number of deaths per capita versus day** plot.



Based on the above graph, Italy has the most deaths over this period.



Based on the above graph, Italy has the most deaths per capita over this period.

The countries that have no deaths reported over this period are Latvia, Malta, Slovakia.

3. First Model

a.

The first model I used the analyze the data is a Poisson model. The response variable is the number of deaths. This is a rate model. The rate in this model is the population, more accurately, the natural logarithm of the population. The log of population is used as the rate parameter because we presume that population size plays a major factor in the death count.

The Parameters Used:

- deaths - number of Coronavirus deaths. Approximated via a poisson distribution based off of the population (of each country) and the day (centered).
- lambda - prior used in the deaths poisson distribution. It is a linear combination of the log population, intercept, and product of the slope and centered days.
- intercept - intercept of the lambda parameter. Approximated via a normal distribution with intercept mean and intercept variance, both explained in the hyper-priors section below.

The Hyper-priors Used:

- intercept mean - normal distribution with mean 0 and variance 100
- intercept variance - uniform distribution (flat) 0 to 100
- slope - normal distribution with mean 0 and variance 100

For the linear portion (lambda), the intercept differs by country, whereas the slope is the same for all countries.

b.

```
####################################################################################
# Q3.b
# JAGS Model for the posterior distributions
####################################################################################
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)

}
```

c.

Posterior distributions of the first model parameters were approximated using Markov Chain Monte Carlo (MCMC) methods implemented in JAGS using the rjags package in the R programming language. There were four chains that were used in the analysis with overdispersed starting values. 2,000 iterations were discarded for burn-in. Another 2,000 iterations were used to measure convergence. Convergence was confirmed using both the Gelman-Rubin Statistic and visually using the Trace plots. Sample sizes of over 4,000 were collected for all parameters.

d.

Based off of the analysis of the first model, the analysis of the top-level parameters is as follows:

- The mean of the slope is 0.1086179
- The standard deviation of the slope is 0.0015479
- The 95% central posterior interval of the slope is (0.1055486, 0.1116178)

- The mean of the intercept mean is -1.2381987
- The standard deviation of the intercept mean is 0.3980256
- The 95% central posterior interval of the intercept mean is (-2.0475534, -0.4731147)

- The mean of the intercept standard deviation is 1.9679022
- The standard deviation of the intercept standard deviation is 0.3346898
- The 95% central posterior interval of the intercept standard deviation is (1.4405456, 2.7288921)

e.

Based off of the analysis of the first model, the country that has the highest posterior median per capita value for its intercept is Italy. And the country that has the lowest posterior median per capita value for its intercept is Slovakia.

f.

The effective number of parameters is about 27. This is just 1 less than the actual number of parameters. The approximate value of (Plummer's) DIC is 3742.

4. Second Model

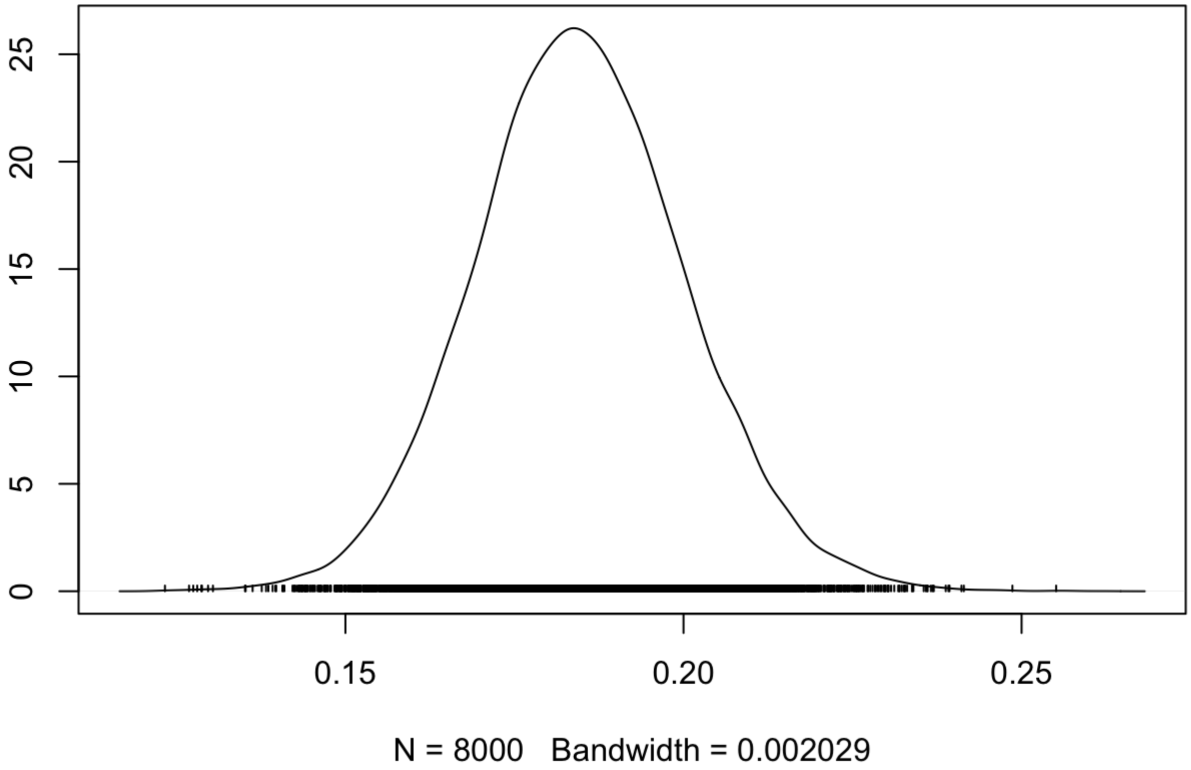
a.

```
###################################################################################
# Q4.a
# JAGS Extended Model for the posterior distributions
###################################################################################
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.slope ~ dnorm(0, 1/100^2)
  sigma.slope ~ dunif(0, 100)

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

}
```

b.

Posterior distributions of the second model parameters were approximated using Markov Chain Monte Carlo (MCMC) methods implemented in JAGS using the rjags package in the R programming language. Similar to the first model, there were four chains that were used in the analysis with overdispersed starting values. 2,000 iterations were discarded for burn-in. Another 2,000 iterations were used to measure convergence. Convergence was confirmed using both the Gelman-Rubin Statistic and visually using the Trace plots. Sample sizes of over 4,000 were collected for all parameters.

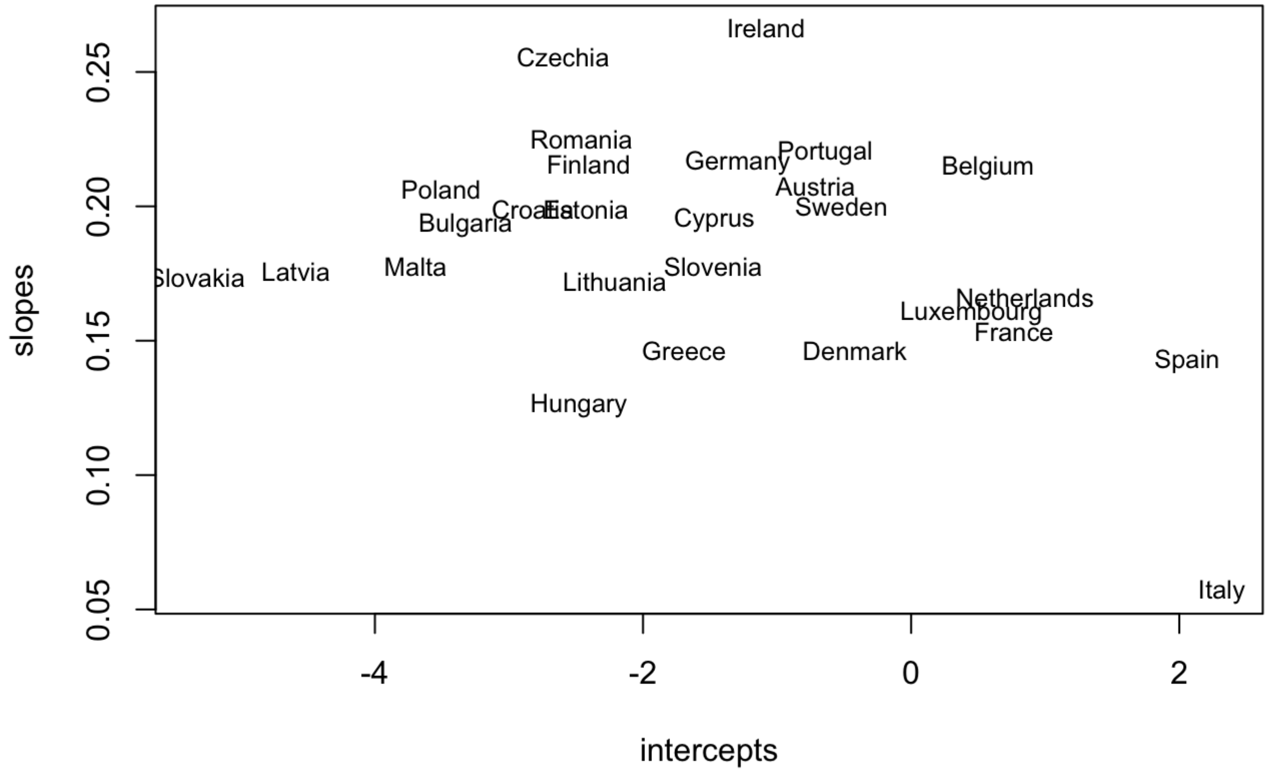
c.



The graph above is the approximate posterior density for the σ_{slope}. Considering most of the density is above 0, this suggests that there are actual differences among the slopes of different countries.

d.

For each country, the approximate posterior expected intercept and an approximate posterior expected slope have been computed and plotted on the graph below. The vertical axis is the approximate posterior expected slope and the horizontal axis is the approximate posterior expected intercept.



e.

The effective number of parameters in the second model is 42. This is considerably higher than in the first model because each country has its own slope parameter.
The approximate value of (Plummer's) DIC is 2631. Since the DIC value is smaller in the second model, that is proof that the second model is better than the first one.

5. Conclusions

In the first model, it can be seen that the intercept is significant considering its 95% central posterior interval does not include 0. This implies that the number of deaths in a country within a period is influenced by the number of deaths that have already occurred previously.

Also, the first model, the slope parameter is positive. This was expected considering the pandemic was ongoing during the period examined.

The second model, however, sheds light into the the significance of separate slope parameters for each country. The approximate posterior density for the σ_{slope} parameter was above 0, thus proving its significance. The approximate value of (Plummer's) DIC further showed that the second model was better that the first.

In conclusion, results from this study show that the number of deaths from COVID-19 within a period is influenced by the number of deaths that have already occurred, and the rate of deaths is different depending on the country.

6. Appendix

```
###################################################################################
# Libraries
###################################################################################
library(rjags)
library(lattice)


###################################################################################
# Q2
# Number of deaths versus day plot
###################################################################################

eucoviddeaths <- read.csv('EUCOVIDdeaths.csv', header = TRUE)

colors <- c("#285F17", "#EB5528", "#27466C", "#853B3D", "#748D84",
            "#955735", "#EF8632", "#EA33F7", "#75FBFD", "#EA3323",
            "#75FB4C", "#988950", "#75147C", "#C5DBC2", "#B6D7E4")

plot(16:31, eucoviddeaths[1, 3:18], type='l', col=colors[1], main = "number of deaths versus day", xlab='March 2020', ylab='# deaths', ylim=c(0, max(eucoviddeaths[, 3:18])))
for(i in 2:nrow(eucoviddeaths)) {
  lines(16:31, eucoviddeaths[i, 3:18], col=colors[i])

}
legend("topleft", legend = paste(eucoviddeaths$Country), col = colors[1:15], pch = 19, bty = "n")


###################################################################################
# Q2
# Number of deaths per capita versus day plot
###################################################################################

plot(16:31, eucoviddeaths[1, 3:18]/eucoviddeaths[1, 2], type='l', col=colors[1], main = "deaths per capita versus day", xlab='March 2020', ylab='# deaths per capita', ylim=c(0, max(eucoviddeaths[, 3:18]/eucoviddeaths[, 2])))
for(i in 2:nrow(eucoviddeaths)) {
  lines(16:31, eucoviddeaths[i, 3:18]/eucoviddeaths[i, 2], col=colors[i])

}
legend("topleft", legend = paste(eucoviddeaths$Country), col = colors[1:15], pch = 19, bty = "n")

###################################################################################
# Q2
# Countries that have no deaths reported
###################################################################################

eucoviddeaths$total <- 0

for(i in 1:nrow(eucoviddeaths)) {
  eucoviddeaths[i, 'total'] <- sum(eucoviddeaths[i, 3:18])
}
```

```
eucoviddeaths[which(eucoviddeaths$total==0), ]$Country


#####################################################################################
# Q3.b
# JAGS Model for the posterior distributions
#####################################################################################
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)

}


#####################################################################################
# Q3.c
# JAGS Model for the posterior distributions: First Model Computations
#####################################################################################

# Data
d1 <- list(logpopulation = log(eucoviddeaths$PopulationM),
           deaths = eucoviddeaths[, 3:18],
           daycent = as.vector(scale(16:31, scale = FALSE)))

# Initialization
inits1 <- list(list(sigma.intercept=0.01, mu.intercept=100, slope=10),
               list(sigma.intercept=100, mu.intercept=-100, slope=10),
               list(sigma.intercept=0.01, mu.intercept=100, slope=-10),
               list(sigma.intercept=100, mu.intercept=-100, slope=-10))

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

# Burn in of 2000
update(m1, 2000)

# 2000 iterations for convergence
x1 <- coda.samples(m1, c("sigma.intercept", "mu.intercept", "slope"), n.iter=2000)
```

```r
gelman.diag(x1, autoburnin = FALSE, multivariate = FALSE)

# 4000 iterations for analysis
x1 <- coda.samples(m1, c("sigma.intercept", "mu.intercept", "slope", "intercept"), n.iter=4000)

# Trace Plots
plot(x1[, c("sigma.intercept", "mu.intercept", "slope")], smooth=FALSE)

# Effective Size
effectiveSize(x1[, c("sigma.intercept", "mu.intercept", "slope")])


###################################################################################
# Q3.d
# Approximation of hyperparameters: Slope
###################################################################################

# Mean
summary(x1)$statistics["slope", "Mean"]

# Standard Deviation
summary(x1)$statistics["slope", "SD"]

# 95% central posterior interval
summary(x1)$quantiles["slope", "2.5%"]
summary(x1)$quantiles["slope", "97.5%"]


###################################################################################
# Q3.d
# Approximation of hyperparameters: Intercept Mean
###################################################################################

# Mean
summary(x1)$statistics["mu.intercept", "Mean"]

# Standard Deviation
summary(x1)$statistics["mu.intercept", "SD"]

# 95% central posterior interval
summary(x1)$quantiles["mu.intercept", "2.5%"]
summary(x1)$quantiles["mu.intercept", "97.5%"]


###################################################################################
# Q3.d
# Approximation of hyperparameters: Standard Deviation
###################################################################################

# Mean
summary(x1)$statistics["sigma.intercept", "Mean"]

# Standard Deviation
```

```
summary(x1)$statistics["sigma.intercept", "SD"]

# 95% central posterior interval
summary(x1)$quantiles["sigma.intercept", "2.5%"]
summary(x1)$quantiles["sigma.intercept", "97.5%"]


###################################################################################
# Q3.e
# Country with the highest/lowest posterior median value for its intercept
###################################################################################
post.samp1 <- as.matrix(x1)[, 1:27]

# highest posterior median
medians.intercept <- apply(post.samp1, 2, FUN = median)
maxmedianintercept <- max(medians.intercept)
eucoviddeaths$Country[which(medians.intercept == maxmedianintercept)]

# lowest posterior median
minmedianintercept <- min(medians.intercept)
eucoviddeaths$Country[which(medians.intercept == minmedianintercept)]


##############################################################################################
# Q3.f
# Approximate the value of (Plummer's) DIC and the associated effective number of parameters
##############################################################################################
dic.samples(m1, 20000)


###################################################################################
# Q4.a
# JAGS Extended Model for the posterior distribution
###################################################################################
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.slope ~ dnorm(0, 1/100^2)
  sigma.slope ~ dunif(0, 100)
  mu.intercept ~ dnorm(0, 1/100^2)
  sigma.intercept ~ dunif(0, 100)

}


###################################################################################
# Q4.b
# JAGS Model for the posterior distributions: Second Model Computations
###################################################################################
# Data
d2 <- list(logpopulation = log(eucoviddeaths$PopulationM),
           deaths = eucoviddeaths[, 3:18],
           daycent = as.vector(scale(16:31, scale = FALSE)))

# Initialization
inits2 <- list(list(sigma.intercept=0.01, mu.intercept=100, sigma.slope=0.01, mu.slope=1
0),
               list(sigma.intercept=100, mu.intercept=-100, sigma.slope=100, mu.slope=10
),
               list(sigma.intercept=0.01, mu.intercept=100, sigma.slope=100, mu.slope=-1
0),
               list(sigma.intercept=100, mu.intercept=-100, sigma.slope=0.01, mu.slope=-
10))

# JAGS model
m2 <- jags.model("secondmodel.bug", d2, inits2, n.chains=4, n.adapt=1000)

# Burn in of 2000
update(m2, 2000)

# 2000 iterations for convergence
x2 <- coda.samples(m2, c("sigma.intercept", "mu.intercept", "sigma.slope", "mu.slope"),
 n.iter=2000)

gelman.diag(x2, autoburnin = FALSE, multivariate = FALSE)

# 8000 iterations for analysis
x2 <- coda.samples(m2, c("sigma.intercept", "mu.intercept", "sigma.slope", "mu.slope",
"intercept", "slope"), n.iter=8000)

# Trace Plots
plot(x2[, c("sigma.intercept", "mu.intercept", "sigma.slope", "mu.slope")], smooth=FALSE
)

# Effective Size
effectiveSize(x2[, c("sigma.intercept", "mu.intercept", "sigma.slope", "mu.slope")])


###################################################################################
# Q4.c
# Approximate posterior density
###################################################################################
```

```
densplot(x2[, "mu.slope"])


##################################################################################
# Q4.d
# Posterior expected intercept
##################################################################################
intercepts <- as.vector(summary(x2)$statistics[paste("intercept[", 1:nrow(eucoviddeath
s),"]", sep=""), "Mean"])
slopes <- as.vector(summary(x2)$statistics[paste("slope[", 1:nrow(eucoviddeaths),"]", se
p=""), "Mean"])

plot(x = intercepts, y = slopes, type="n")
text(intercepts, slopes, eucoviddeaths$Country, cex=0.8)


#########################################################################################
#####
# Q4.e
# Approximate the value of (Plummer's) DIC and the associated effective number of parame
ters
#########################################################################################
#####
dic.samples(m2, 20000)
```

[1] World Health Organization (WHO) - Naming the coronavirus disease (https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it)↩

[2] HWO - Novel Coronavirus – China (https://www.who.int/csr/don/12-january-2020-novel-coronavirus-china/en/)↩

[3] WHO - Coronavirus dashboard (https://covid19.who.int/)↩

[4] USNews - How Does Coronavirus Spread? (https://health.usnews.com/conditions/articles/how-does-coronavirus-spread)↩

[5] WHO - Situation Report 73 (https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200402-sitrep-73-covid-19.pdf?sfvrsn=5ae25bc7_4)↩

[6] The Intercept - Europe became the hub for Coronavirus (https://theintercept.com/2020/04/02/coronavirus-europe-travel/)↩

[7] European Centre for Disease Prevention and Control (https://www.ecdc.europa.eu/en/publications-data/%20download-todays-data-geographic-distribution-covid-19-cases-worldwide) Accessed April 22, 2020↩