ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY
ΜΕΤΑΠΤΥΧΙΑΚΟ ΣΤΑΤΙΣΤΙΚΗ
MSc IN STATISTICS
FULL TIME: 47A, Evelpidon Street & 33, Lefkados Street, Athens 113 62 Greece. E-mail: masterst@aueb.gr / www.masterst.aueb.gr

Course: Epidemic Models

Lecturer: N. Demiris

Student: Andre Ehrlich

Final Assignment

Submission/Exam/Presentation: 03 Apirl 2023

Each student will analyse the COVID-19 case data for the period 1 Sep 2020 – 20 June 2021 for 3 European countries.

(a) The analysis could take place with the chain binomial model we saw in the practical, applied to the Shanghai influenza data. Note that the COVID data are typically the number of new cases and not the total number of cases (active set).

(b) Fit a more realistic model where the infection rate (and therefore also Rt) is piecewise constant with 1 change point that you select.

(c) Extend the model to multiple changepoints.

(d) Fit the model with 1 changepoint by estimating the time point of change. The Stagnant example could be useful for this task.

(e) (Optional) Try a combination of © and (d) fitting a model with multiple unknown changepoints.

Alternatively, at https://imperialcollegelondon.github.io/epidemia/ is a package you may find user-friendly. You may analyse the data via this package if you prefer.

Abstract

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a virus, SARS-CoV-2, that killed an estimated 6.8 Million people world-wide over the course of 3 years, and caused major disruptions to the world's economies and socialization norms. In this paper, we analyze 3 countries' data over the 2020-2021 school year (Sep 1 2020 - June 20, 2021).

We fit Reed-Frost Chain Binomial changepoints models using computational inference software BUGS (Bayesian Inference Using Gibbs Sampling) to make statistical inference on the transmission rate parameters of the Chain Binomial that was fit to each COVID wave.

COVID-19 Pandemic

Viral Disease

Viruses are infectious agents that replicate inside the living cells of organisms, and can spread amongst a population via several known mechanisms such as sexual activity, infectious molecules shared via breathing, and other forms of contact. COVID-19 spreads via aerosol droplets in breath.

Biological components considered during statistical epidemic analysis: infectious period, severity of illness, basic reproduction number (R-naught), effect of age, etc. Viruses mutate when they replicate, varying characteristics of the replicant strands. This variation is overlooked for the purpose of this analysis.

Social Measures

Social considerations for viral pandemic: social contact norms (how people congregate, greet each other, share contact), interventions against transmission, etc.. To protect against COVID-19, societies and individuals modified their behavior in several ways in order to elude or minimize the spread and/or severe outcomes of the disease, such as working from home, closing businesses, restricting travel, socializing in open-air environments, and reducing social contact. Behavior varied by region.

Waves

There are typically 2 or 3 "waves" of COVID. These waves could be dictated by a mixture of biological and social factors. Socially, nations, states, cities enacted sporadic isolation measures; also seasonal factors such as cold weather reducing outdoor gathering and school schedules. Biologically, the COVID-19 had several identified mutated strains with distinct characteristics. COVID-19 infection-induced immunity wanes after a period of time, re-exposing the person susceptible to infection.

Methodology

Epidemic Modeling

Epidemiology, from Greek as roughly *“what befalls a population”,* is the study of public health and disease transmission. Epidemic modeling is complex and calls for the quantification of uncertainty and the simulation of changing conditions. In the context of COVID-19, we must summarize the behavior of ever-mutating viral interpersonal transmissions on national and global scales, in such a way that it is informative and/or actionable to political decision-makers and health-care professionals.

Discrete-time Reed-Frost Model

The Generalised Stochastic Epidemic (GSE), has a likelihood that is analytically and numerically intractable, and is mathematically complex and computationally difficult even for moderate populations.

The Reed-Frost Chain Binomial (RFCB) model is a modest alternative discrete-time process that assumes closed, homogenous, and homogeneously mixing process, in which at each time step, *“each infectious individual independently infects each susceptible individual with the same probability”* (Malmros et al, 2014). It is a chain of binomial random variables. Each state is dependent on the previous. Each day-to-day transition is a binomial. The likelihood is a product of binomials.

“Reed-Frost model is a special case of the standard SIR epidemic model presented in Chapter 2, in which the length of the infectious period is deterministic (implying that contacts between pairs of individuals occur independently).” (Anderson & Britton).

This distribution is complex, and lends itself to MCMC method to obtain the posterior probability density. Numerically, S(t) and I(t) are functions of a single parameter, which is the probability of infection from a given individual. We have to estimate this single parameter given the daily case count data.

Bayesian Inference Using Gibbs Sampling (BUGS)

We use free computational software (BUGS) to estimate the posterior distribution using a Bayesian model formulation. We specify prior belief on the parameters, and then update this belief based on the data. We consider the model parameters (including change-points) to be distributions.

Analysis

We look at at COVID data from the European Center for Disease Control (ECDC), interested to fit Reed-Frost Chain Binomial changepoints models using computational inference software BUGS to make statistical inference on the transmission rate parameters of the Chain Binomial that was fit to each COVID wave.

The nature of this work is exploratory, and makes simplistic assumptions about the nature of the spread. We consider the infectious period to be a constant, we do not consider the possibility of re-infection. Considering the large population size of the European countries under analysis, we may consider the results of our Reed-Frost Chain Binomial model to be the asymptotic result of a more complex stochastic model, according to Function Central Limit Theorem.

Seasonality

The reported cases have clear weekday seasonality. More cases are reported on Tuesday and Wednesday, and fewer on Sunday and Monday. This is likely an artifact of the data reporting mechanism, and not reflective of any underlying biological characteristic of the infectious spread of the virus. The periodicity is quite severe in both Global & Czech data (See figure 1), less so in Denmark.

Perhaps a social component of disease spread affects intraweek periodicity, but at this scale of analysis, it is overlooked. We “deseasonalize” the time series data, by removing the average case count by day-of-week from each observation, to have a smoother, less noisy curve with which to fit our model.

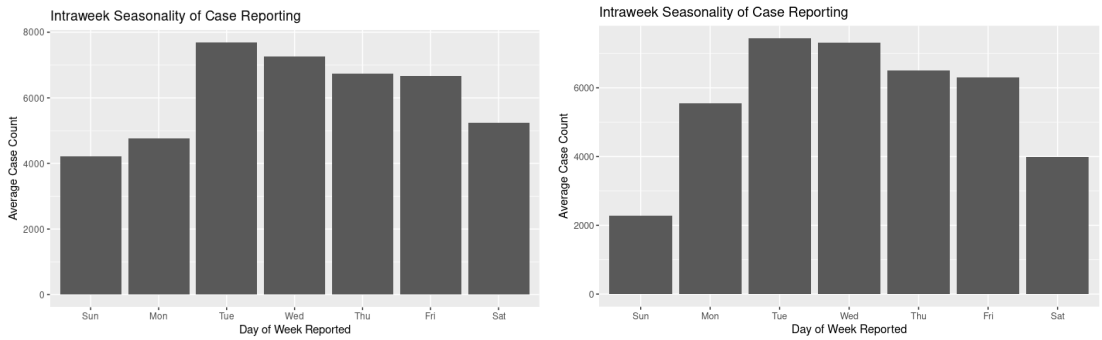
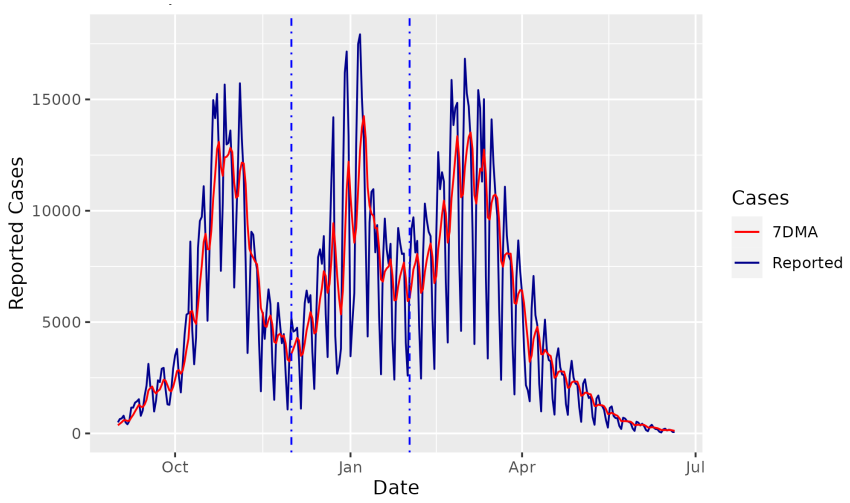


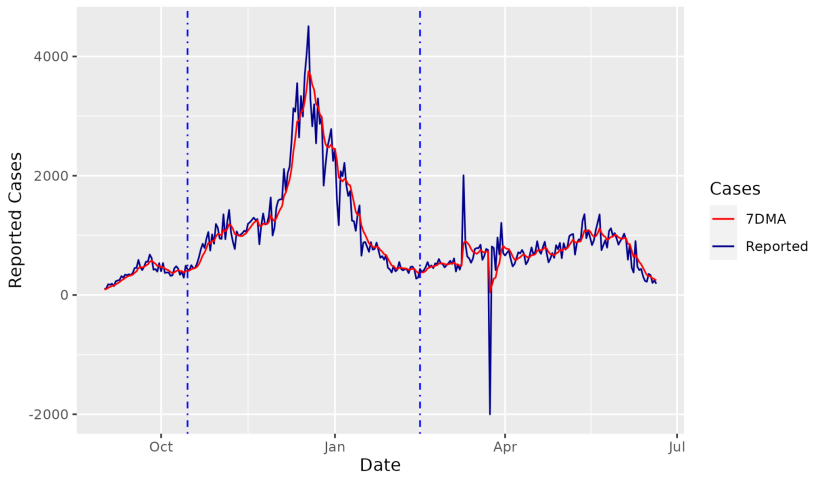
Figure 1: Intraweek Seasonality (a) Globally (b) Czechia

Country	**Population (Milions)**	**Max Cases**	**Waves**
CZ	10	14,251	2
DK	5.8	3,751	1 - 2
LU	0.6	748	2

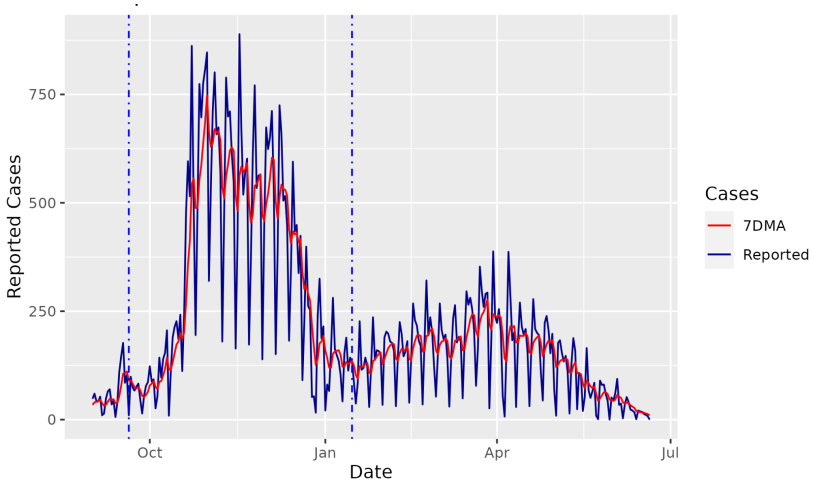
Table 1: A few Summary Statistics



(a) Czechia (CZ)



(b) Denmark (DK)



(c) Luxembourg (LU)

Figure 2: Country Cases: Reported (Dark-Blue), 7DMA (Red), Change Points (Blue-dashed)

Change Point Modeling

Our aim is to perform statistical inference on the spread of the COVID-19 disease, most importantly, the rate of transmission (the beta parameter) of the RFCB distribution fitted to the data. In the European countries, there are typically waves of viral spread, each of which should each fit its own RFCB model parameter.

We will follow an exploratory procedure to understand fitting such changepoint models. We naively fit a single beta parameter to the whole data set, then a second iteration where we specify exactly a single changepoint, and fit two beta parameters to the data, and then a third iteration with three changepoints. We follow this process regardless of the number of modes that appear in the case data of the given country. We specify changepoints with respect to the discrete time parameter.

We employ a second exploration to estimate the changepoint location(s) by treating them as a random variable and merely specifying prior distribution in our model, such that the Gibbs Sampling run via BUGS will learn the changepoint, and we can perform inference on the changepoint itself. We specify uniform prior and initial values within the 30-day range of our specified prior belief.

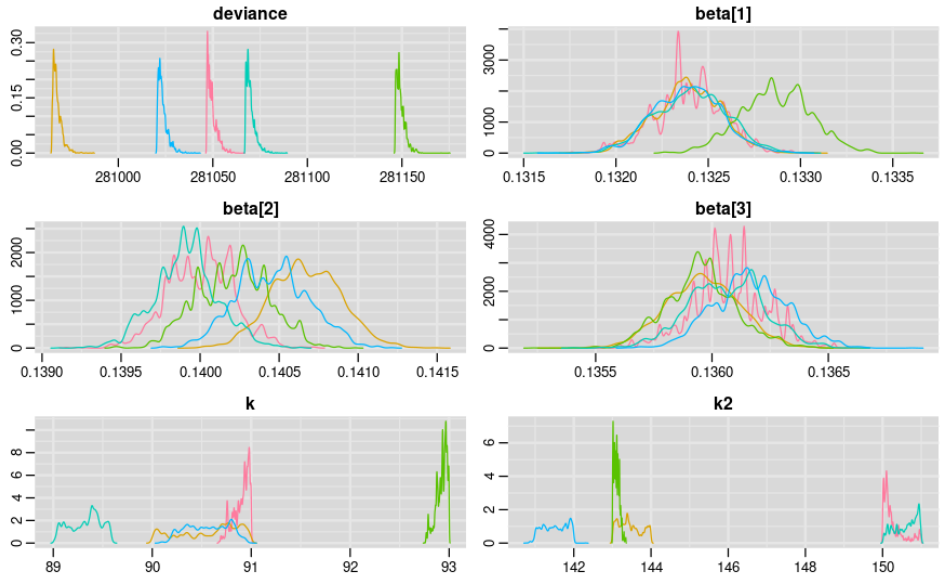
```
cp_margin <- 30
k1_lb <- k1 - cp_margin
k1_ub <- k1 + cp_margin

k2_lb <- k2 - cp_margin
k2_ub <- k2 + cp_margin

k ~ dunif(k1_lb,k1_ub
k2 ~ dunif(k2_lb,k2_ub)
```

We compare our models with Deviance Information Criterion. We also judge convergence by running several markov chains with random initial estimates, and by the autocorrelation of simulated values from the posterior distribution. See Table 1 for comparison. We typically performed our MCMC simulation with 5 chains, with 500 - 5000 burnin as model complexity increases, with n.thin = 50, and iterations = 250,000, up to as many as 500,000 iterations.

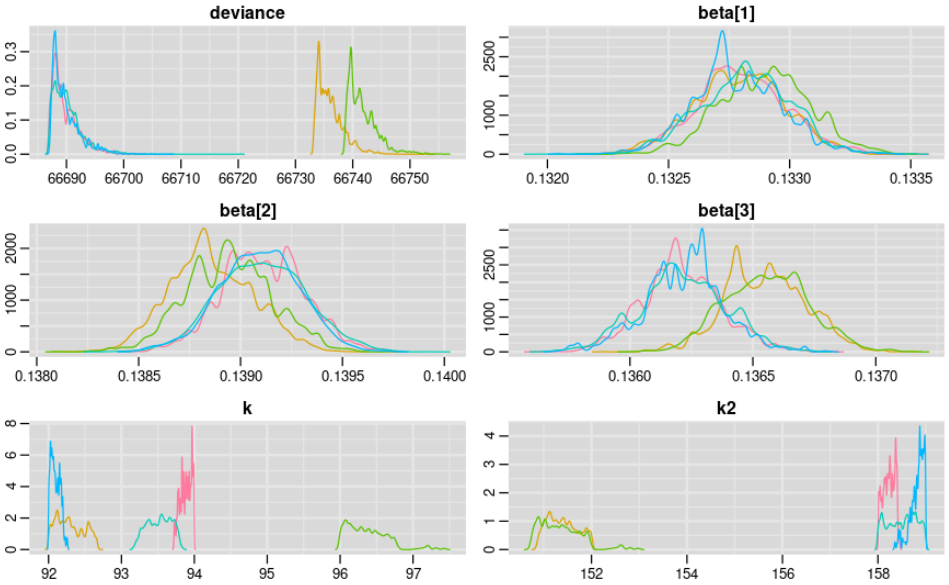
Czechia

For Czechia, there are clearly 2 waves in the data, but prevalent seasonal intraweek seasonal components of the data-reporting may interfere with fitting to spread of the virus. Seasonality is not so severe in the other two countries, especially not in Luxemburg.

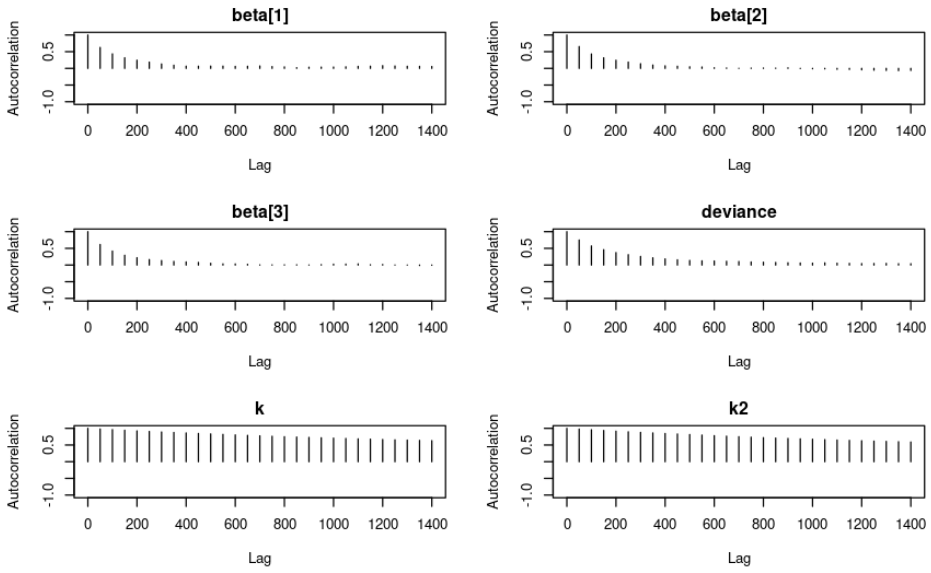
We compared the multi-change point estimation model fitted on the raw reported cases against the same model fitted on the 7DMA of reported cases (see figure 2). The autocorrelation is heavily-present in the change-point parameters, and the chains do not indicate convergence. The chains for the 7DMA model seem to have two clusters, maybe this indicates partial convergence. There is high auto-correlation still after completing 500,000 iterations, further iterations may be required, but likely something else is missing.



(a) CZ: Density Plots Raw Reported Cases



(b) CZ: Density Plots 7DMA



(c) CZ: Auto-Correlation plots are similar for both data/models)

Figure 2: CZ: Estimate 2 changepoints (a) Density Raw Reported Cases (b) Density 7DMA (c) AutoCorrelation

Luxemburg

The Luxemburg 2-changepoint estimation has much lower autocorrelation, and the chains do not wholly converge, but they seem to get confused between the two parameters. The posterior point estimates have a high standard deviation, of about 2-3 weeks. These chain convergence diagnostics indicate a lack of stability in the computational simulation mechanism. It is possible that there is only one true changepoint, just after the new year.

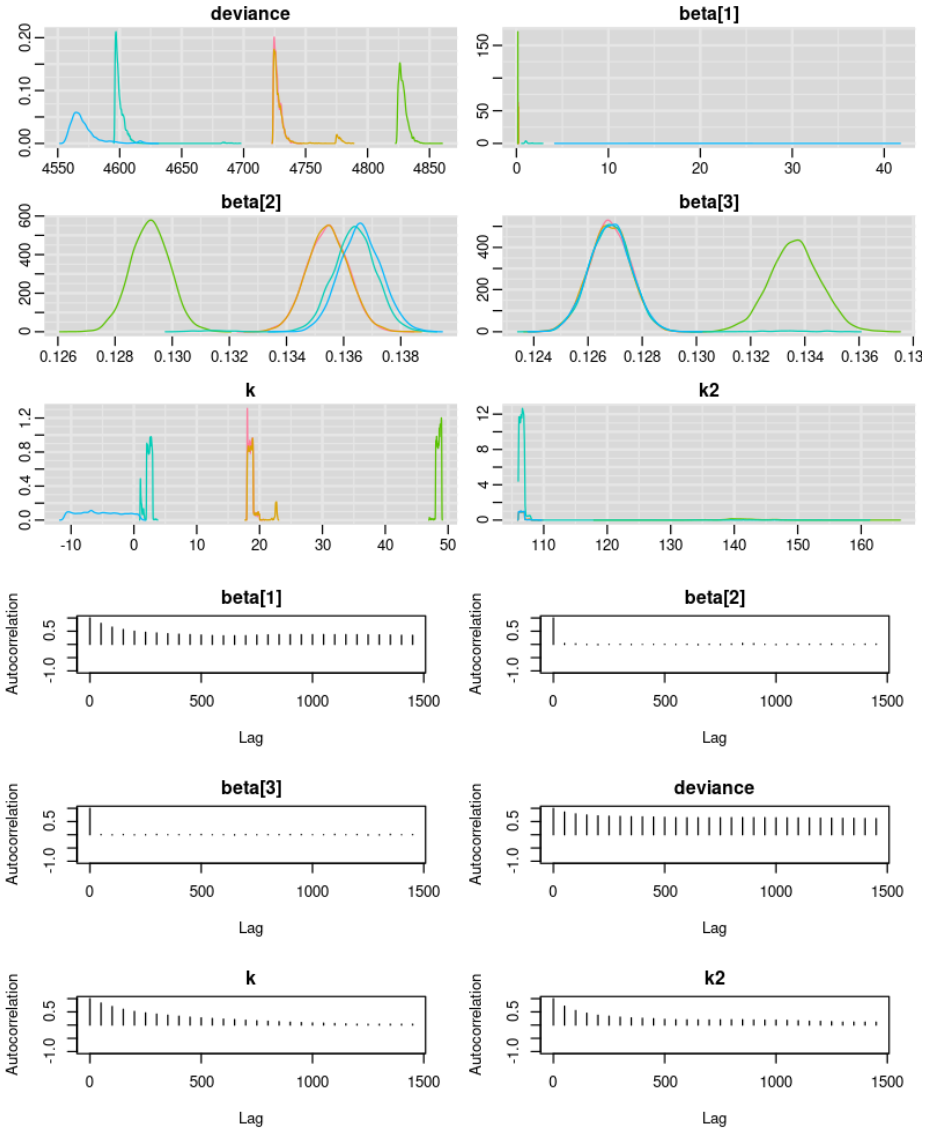


Figure 3: LU 2 estimated changepoints (a) Density Plots (b) Autocorrelation Plots

Denmark

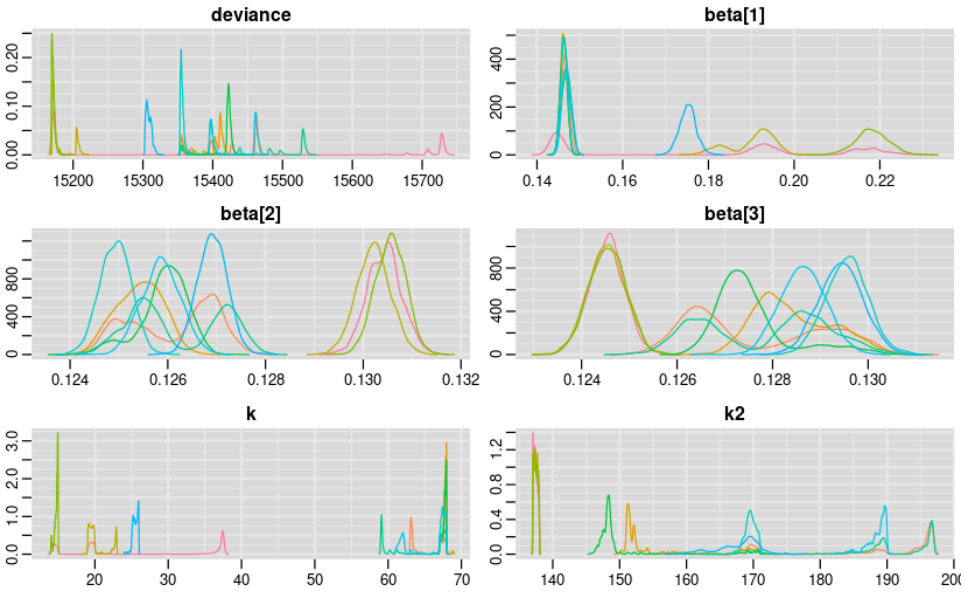


Figure 4: DK 2 Estimated Changepoints

Denmark may not have any changepoints. The posterior estimates (for 10 chains) sprawled over 40 days for each changepoint. See appendix for autocorrelation plot.

Learn Number of Points

Say we want to estimate the number of change points and their respective locations. Is this a tractable problem ?

One idea we have is to recursively estimate the variable number of change points based on the MCMC chain divergences of an initial run. Although it is not clear whether the posterior density of a particular changepoint is due to computational instability or existence of real changepoint?

Pseudo Algorithm: Find the first change point. Fit many models with different change-points and compare DIC. Can take two approaches: (a) Greedy search (b) Find local maxima/minima numerically. If there are several real change points in the data generating mechanism, more than the specified number of parameters in the model, then the simulation will get confused and not converge, it will jump around to the several eligible changepoints, as seen previously in this analysis. Can we use this to make a more informed second guess ? Or will we get fooled by unsophisticated model and simulation instability ?

A more sophisticated model might be an Expectation Maximization algorithm to find the missing change-point parameters (Paquette). Or a change-point estimation via Gaussian Processes.

Conclusion

Convergence for estimated changepoints seems tenuous at best. Across all three data sets, the estimated changepoints models do not converge under the current model specification. Also, the DIC generally does not decrease as significantly as we expected when introducing changepoints into the models. The nearness of the marginal posterior distribution of changepoint parameters to the human-observed changepoints seem to be due to strong prior placement more than robust model dynamics. But, this probably indicates that something is structurally missing from our model to be well-fitted to our data with informative posterior estimates of the changepoints. More sophisticated models are worth exploring with this problem.

Appendix

Country	# Change points	pD	DIC	Beta μ (σ)	K
CZ 7DMA	0	1	67226	0.136 (0.0)	NA
	1	2	66979	0.133 (0.0) 0.137 (0.0)	k = 90
	2	3	66930	0.133 (0.0) 0.137 (0.0) 0.138 (0.0)	k = 90 K2 = 145
	2 (est)	-67.9	66640	0.132 (0.0) 0.136 (0.0) 0.139 (0.0)	$\widehat{k}$ = 93.6(1.5) k_2=155.6(3.5)
DK 7DMA	0	1	16043	0.129 (0.0)	NA
	1	2	15666	0.126 (0.0) 0.138 (0.0)	k = 44
	2	3	15595	0.125 (0.0) 0.129 (0.0) 0.139 (0.0)	k = 44 k_2 = 167
	2 (est)	-506	15210	0.123 (0.001) 0.128 (0.002) 0.140 (0.001)	$\widehat{k}$ = 95.2 (6.8) $\widehat{k_2}$=155.3(7.5)
LU 7DMA	0	1	4953	0.132(0.001)	NA
	1	2	4844	0.132 (0.001) 0.194 (0.007)	k = 19
	2	3	4788	0.130 (0.006) 0.133 (0.001) 0.203 (0.001)	k = 19 k_2 = 136
	2 (est)	-244	4708	2.337 (3.8) 0.135 (0.002) 0.138 (0.001)	$\widehat{k}$ = 16.3(19.7) $\widehat{k_2}$=113(12.1)

Table 2: Summary of fitted models

Country	# Change points	pD	DIC	Beta μ (σ)	K	Note
CZ	0	1	281796	0.136 (0.0)	NA	
	1	2	281455	0.133 (0.0) 0.137 (0.0)	k = 90	
	2	3	281200	0.133 (0.0) 0.137 (0.0) 0.139 (0.0)	k = 90 k_2 = 145	Negligible difference in DIC
	2 (est)		280700	0.132 (0.0) 0.136 (0.0) 0.140 (0.0)	$\widehat{k}$ = 90.85 $\widehat{k_2}$ = 145.785	Convergence ? DIC 1000 less than 0-cp model

Table 3: Czechia Raw Reported Cases

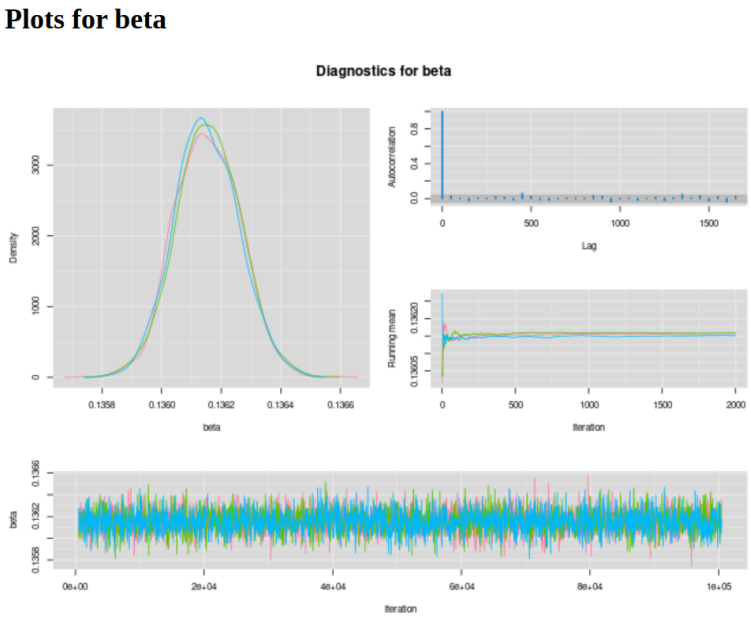
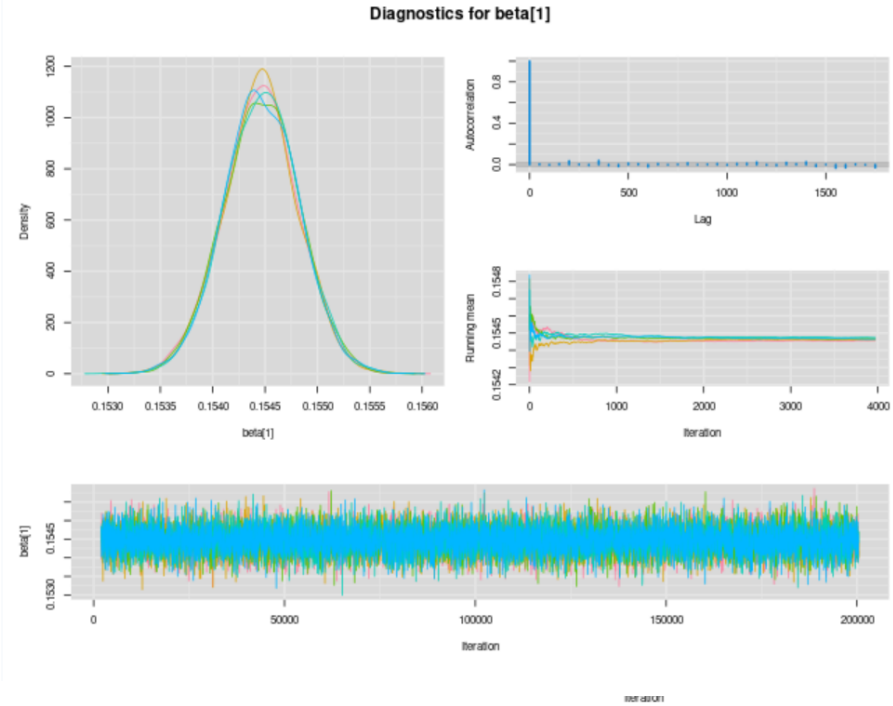
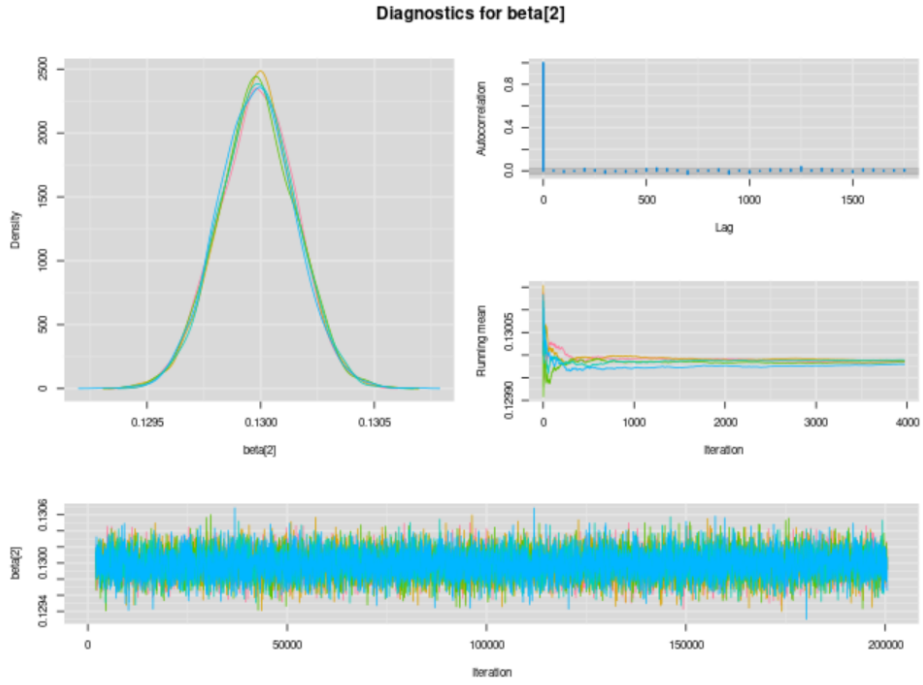


Figure A1: Czechia No Changepoint model: MCMC Diagnostic plots. We observe convergence from all of the chains, they are overlapping in the posterior density plot, running mean. No autocorrelation is observed from the sample after burn-in period.


Diagnostics for beta[1]
Density
beta[1]
Autocorrelation
Lag
Running mean
Iteration
beta[1]
Iteration


Diagnostics for beta[2]
Density
beta[2]
Autocorrelation
Lag
Running mean
Iteration
beta[2]
Iteration

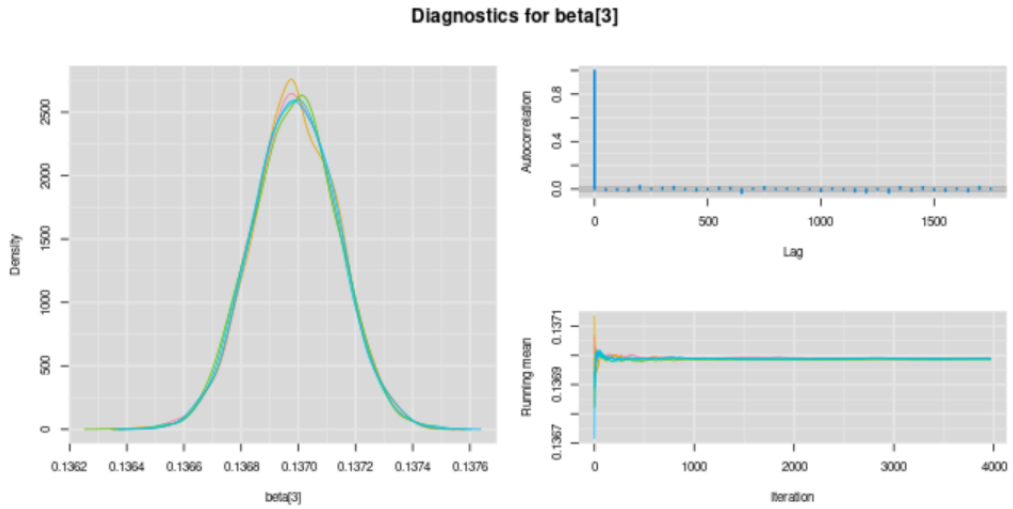


Figure A2: RFCB Diagnostic Plots for Model with 2 Exact ChangePoints

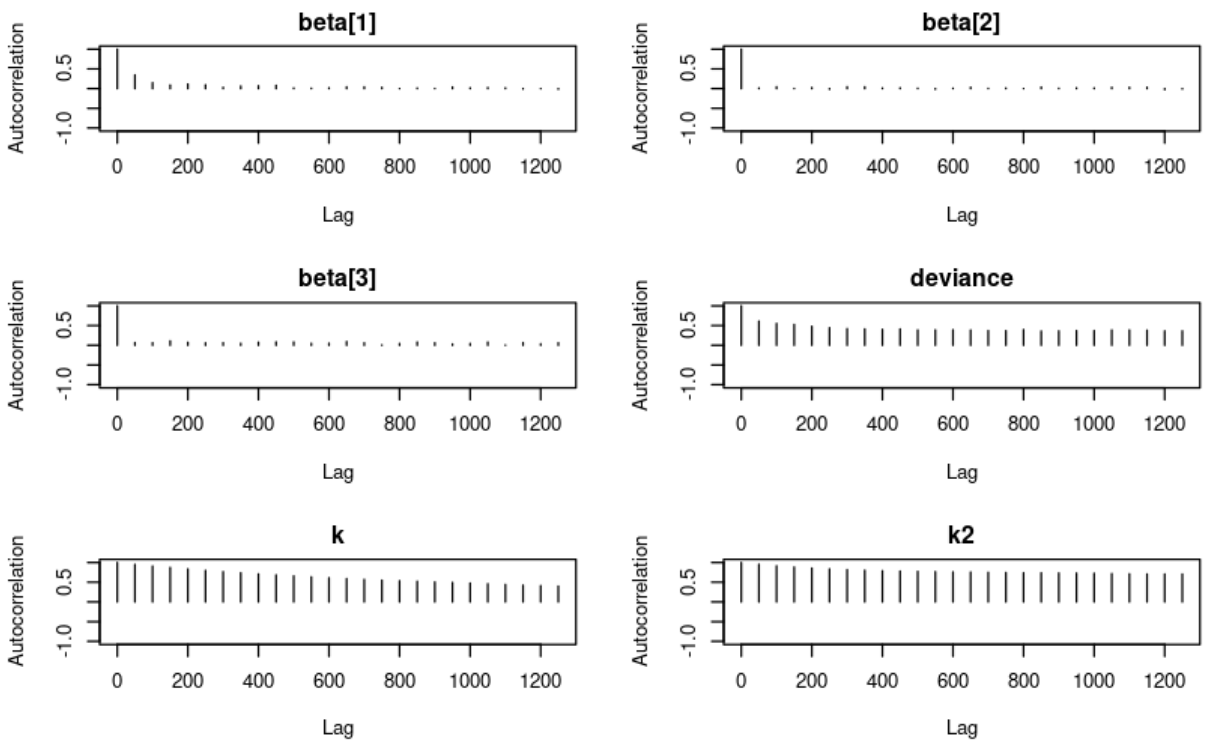


Figure: Denmark Autocorrelation Plots

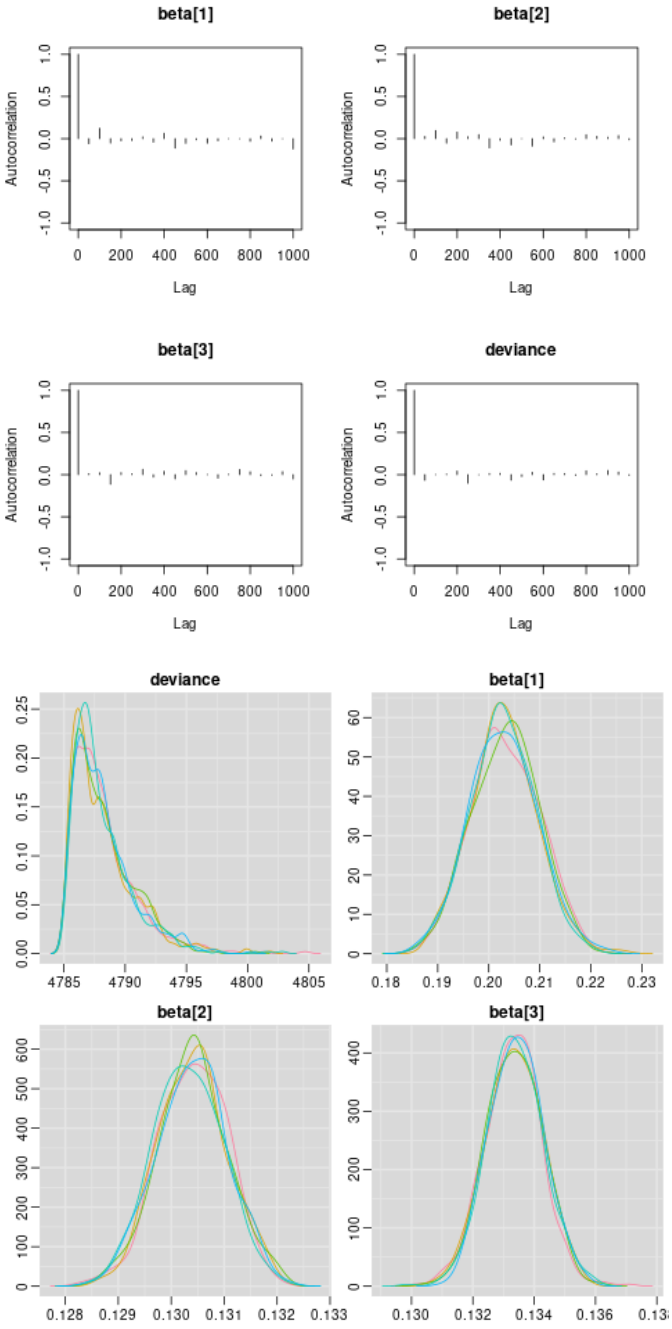


Figure: Luxemburg Model 3, Multiple Exact changepoints

Bibliography

1 Demiris Lecture Slides
2 Britton & Anderson, Stochastic Epidemic Models
3 Malmros et al, Random Walks on Directed Networks: Inference and Respondent-Driven Sampling
4 Carlin et Al, Hierarchical Bayesian Analysis of Changepoint Problems
5 Ulrich Paquet, Empirical Bayesian Change Point Detection

Code

Unset

````
---
title: "Chain Binomial COVID Changepoints"
author: "Andre Ehrlich"
date: "25/3/2023"
output:
  pdf_document: default
  html_document: default
---

```{r setup, include=FALSE}
library(R2WinBUGS)
library(coda)
library(outbreaks)
library(BRugs)
library(incidence)
library(bayesplot)
library(mcmcplots)


library(zoo)
library(TTR)
library(data.table)
library(lubridate)
library(ggplot2)

knitr::opts_chunk$set(cache = TRUE, echo = TRUE, message = FALSE,
warning = FALSE)
```


# Overview of Analysis
Each student will analyse the COVID-19 case data for the period 1
Sep 2020 – 20 June 2021 for 3 European countries.
````

A) Chain binomial model
- we saw in the practical, applied to the Shanghai influenza data.
- Note that the COVID data are typically the number of new cases and not the total number of cases (active set).
B) Peicewise constant with 1 change point
- Fit a more realistic model where the infection rate (and therefore also Rt) is piece wise constant with 1 change point that you select.
C) Extend the model to multiple changepoints.
D) Estimate change-point
- Fit the model with 1 changepoint by estimating the time point of change.
- The Stagnant example could be useful for this task.
E) Estimate multiple unknown changepoints (Optional)
- Try a combination of © and (d) fitting a model with multiple unknown change points.

```{r}
#
# # WHO COVID GLOBAL DATA
# who_data <- read.table("~/aueb/Semester 2A/Epidemic Models (Demiris)/WHO-COVID-19-global-data.csv", header = T, sep = ",")
#
# who_data$Date_reported <- as.Date(who_data$Date_reported)
#
# # Starting Population for selected countries
# # population <- data.frame("country_code"=c(), "pop"=c())
# # population <- rbind(population, c("CZ", 10708981))
# # population <- rbind(population, c("UK", 12))
# # population <- rbind(population, c("BG", 3))
# # colnames(population) <- c("country_code", "pop")
# # population
#
# ccountry <- "CZ"
```

```
# # country_data <- who_data[who_data$Country_code == ccountry,]
# # country_data
# n_pop <- 10708981#population[population$country_code ==
ccountry]
# n_pop#
#
#
# who_data_subset <- who_data[
#   which(who_data$Country_code == ccountry &
who_data$Date_reported >= start_date & who_data$Date_reported <=
stop_date),
# ]
# who_data_subset
#
# plot(who_data_subset$Date_reported, who_data_subset$New_cases,
type="l")
# # This Data has some strange periodicity likely due to uneven
case reporting
# # WMA ??
#
# # Reported Cases each day between start and stop date.
# new_cases <- who_data_subset$New_cases
# n_obs <- length(new_cases)
#
# # Deterministic Assumption: After 8 days of infection, subject
is removed
# avg_infectious_period <- 8
# new_removals <- c(rep(0, avg_infectious_period), new_cases)
# new_removals <- new_removals[0:n_obs]
#
# # Modify data into a form suitable for BUGS
# data <- list("n_obs", "n_pop", "new_cases", "new_removals")
```

ECDC Data

```{r ecdc_data, echo = TRUE}
```

```
# Time period
start_date <- as.Date("2020-09-01", format = "%Y-%m-%d" )
stop_date <- as.Date("2021-06-20", format = "%Y-%m-%d")

# Deterministic Assumption: After 8 days of infection, subject is
removed
avg_infectious_period <- 8

# smoothing
x_day_moving_average <- 7

# Load Data
ecdc <- read.table("~/aueb/Semester 2A/Epidemic Models
(Demiris)/ecdc_covid_data.csv", header = T, sep = ",")
ecdc$dateRep <- as.Date(ecdc$dateRep, format="%d/%m/%Y")
ecdc <- setDT(ecdc)
```


```{r, country_data}

prepare_country_data <- function(ccountry, changepoints){

  # ccountry = 'LU'
  # changepoints = my_changepoints$LU

  # Select country
  ecdc_subset <- ecdc[geoId == ccountry]

  # Chronological order
  ecdc_subset <- ecdc_subset[order(dateRep)]

  #### MA
  # interpolate na's
  ecdc_subset$cases_narm = na.approx(ecdc_subset$cases)
```

```
  # 7-day average, round to integer
  ecdc_subset$cases_ma <- round(WMA(ecdc_subset$cases_narm, n=7))

  # Time Period cutoff
  ecdc_subset <- ecdc_subset[dateRep >= start_date & dateRep <=
stop_date, ]


  # file.path(getwd(), "cz_cases.png")
  {
    plot_path <- file.path(getwd(), paste0(ccountry, "_cases"))
    # png(file=plot_path, width=1000, height=600)
    par(mfrow=c(1,1))

    ggplot() +
      # Time Series
      geom_line(data=ecdc_subset, mapping=aes(x=as.Date(dateRep),
y=cases,  color = "Reported")) +
      geom_line(data=ecdc_subset, mapping=aes(as.Date(dateRep),
cases_ma, color="7DMA")) +
      # geom_line(data=ecdc_subset, mapping=aes(y=cases,  color =
"Reported")) +
      # geom_line(data=ecdc_subset, mapping=aes(y=cases_ma,
color="7DMA")) +

      scale_color_manual(name = "Cases", values = c("Reported" =
"darkblue", "7DMA" = "red")) +

      # Changepoints
      # geom_vline(aes(xintercept=as.numeric(cz_cp1),
color="Changepoint1")) +
      # geom_vline(aes(xintercept=as.numeric(cz_cp2),
color="Changepoint2")) +
      geom_vline(xintercept=as.numeric(changepoints[1]),
color="blue", linetype=4) +
```

```
      geom_vline(xintercept=as.numeric(changepoints[2]),
color="blue", linetype=4) +
      #

      # Labels
      ylab('Reported Cases') +
      xlab('Date') +
      ggtitle(paste(ccountry, " Reported Cases"))

    ggsave( paste0(ccountry, "_cases.png"),dpi = 300)


  }

  # Specify Model Parameters & Data
  n_pop <- ecdc[ecdc$geoId == ccountry, popData2020][1]
  new_cases <- ecdc_subset$cases_ma
  n_obs <- length(new_cases)
  new_removals <- c(rep(0, avg_infectious_period),
new_cases)[0:n_obs]
  model_data <- list("n_obs"=n_obs, "n_pop" = n_pop,
"new_cases"=new_cases, "new_removals"=new_removals)
  # return(list(model_data, ecdc_subset))

  return(model_data)
}


```



```{r, data_config}


country_codes <- c("CZ", "DK", "LU")
```

```
# Czechia
my_changepoints <- data.frame(
  "CZ" = c(
  as.Date("2020-12-01", format = "%Y-%m-%d" ), # 90
  as.Date("2021-02-01", format = "%Y-%m-%d" )  # 155
),
  "DK" = c(
  as.Date("2020-10-15", format = "%Y-%m-%d" ),
  as.Date("2021-02-15", format = "%Y-%m-%d" )
),
  "LU" = c(
  as.Date("2020-09-20", format = "%Y-%m-%d" ),
  as.Date("2021-01-15", format = "%Y-%m-%d" )
)
)


date_to_cp <- function(mydate) {
  as.numeric(mydate - as.Date("2020-09-01", "%Y-%m-%d"))
}

cz_cp1 <- date_to_cp(my_changepoints$CZ[1])
cz_cp2 <- date_to_cp(my_changepoints$CZ[2])

dk_cp1 <- date_to_cp(my_changepoints$DK[1])
dk_cp2 <- date_to_cp(my_changepoints$DK[2])

lu_cp1 <- date_to_cp(my_changepoints$LU[1])
lu_cp2 <- date_to_cp(my_changepoints$LU[2])

cz_data <- prepare_country_data("CZ", my_changepoints$CZ)
# cz_data


# Denmark
```

```
dk_data <- prepare_country_data("DK", my_changepoints$DK)
# dk_data


lu_data <- prepare_country_data("LU", my_changepoints$LU)
# lu_data



```


The following chain binomial model belongs to the broader class of stochastic discrete-time SIR models.

$$\begin{array}{rcl}
new_cases_{t} & \sim & Bin \left (S_{t-1}, 1-e^{-\beta \frac{I_{t-1}}{N}} \right)\\
S_{t} & = & S_{t-1} - new_cases_{t} \\
I_{t} & = & I_{t-1} + new_cases_{t} - removals_t
\end{array}$$

where $$\beta$$ is the probability that a susceptible individual has infectious contact with an infected individual and becomes infected.

Note that, if we write $$q = e^{\frac{-\beta}{N}}$$ and assume that infectious period is fixed and constant, we have a Reed-Frost model. A susceptible at time t-1 can remain susceptible by avoiding being infected by all infectives I_{t-1}, and the probability of avoiding being infected by one infective is q.

```{r no_change, echo=TRUE}
```

```
# Fit Chain Binomial with no change points

rfcb_nocp <- function(my_data){

  CBmodel <- function(){
    S0 <- n_pop - 1
    I0 <- 1

    p[1] <- 1-exp(-(beta*I0/n_pop))
    new_cases[1] ~ dbin(p[1],S0)
    S[1] <- S0 - new_cases[1]
    I[1] <- I0 + new_cases[1] - new_removals[1]

    for (t in 2:n_obs){
      p[t] <- 1-exp(-(beta*I[t-1]/n_pop))
      new_cases[t] ~ dbin(p[t],S[t-1])
      S[t] <- S[t-1]- new_cases[t]
      I[t] <- I[t-1] + new_cases[t] - new_removals[t]
    }

    # prior
    beta ~ dlnorm(0,5)
  }

  filename<- file.path(getwd(), "CBmodel.bug")
  write.model(CBmodel, filename)

  n_chains=3
  n_burnin=500
  n_iter=25000
  n_thin=50
  set.seed(1234)

  # Specify parameters to monitor:
  params <- c("beta")
```

```
  # Generate initial values for the parameters:
  inits = function(){
    list(beta=runif(1,0.05,0.07))
  }

  time.start_mcmc <- Sys.time()

  # Run MCMC in openBUGS
  mcmc_fit <- openbugs(
      my_data,
      inits,
      model.file = filename,
      parameters.to.save = params,
      # program="OpenBUGS",
      n.chains = n_chains,
      n.iter = n_iter,
      n.burnin = n_burnin,
      n.thin = n_thin
  )

  time.end_mcmc <- Sys.time()
  duration_mcmc <- time.end_mcmc - time.start_mcmc

  mcmc_fit
}

```


```{r one_changepoint,  echo=FALSE}

# 1 changepoint
rfcb_1cp <- function(my_data, cp1){
  my_data$k <- cp1
  CBmodel2 <- function(){
    S0 <- n_pop - 1
```

```
  I0 <- 1
  # k <- cp1


  p[1] <- 1-exp(-(beta[1]*I0/n_pop))
  new_cases[1] ~ dbin(p[1], S0)
  S[1] <- S0 - new_cases[1]
  I[1] <- I0 + new_cases[1] - new_removals[1]

  for (t in 2:n_obs){

    # select beta vector depending on change point
    param[t] <- 1 + step(k - t - 1)
    p[t] <- 1 - exp(-(beta[param[t]]*I[t-1]/n_pop))
    new_cases[t] ~ dbin(p[t],S[t-1])
    S[t] <- S[t-1]- new_cases[t]
    I[t] <- I[t-1] + new_cases[t] - new_removals[t]
  }

  # prior
  for (j in 1:2){
    beta[j] ~ dlnorm(0, 5)
  }
}

paste(getwd())

filename2 <- file.path(getwd(), "CBmodel2.bug")
write.model(CBmodel2, filename2)

n_chains=3
n_burnin=500
n_iter=25000
n_thin=50
set.seed(1234)
```

```
  # Specify parameters to monitor:
  params <- c("beta")

  # Generate initial values for the parameters:
  inits = function(){
    list(beta=c(runif(1,0.05,0.07), runif(1,0.15,0.17)))
  }

  time.start_mcmc <- Sys.time()
  #this function will call bugs through R, the user must set the
correct file, where the bugs #executable is inside
  #If you choose debug == TRUE the Winbugs will remain open, even
after the sampling is finished
  #There is a possible error Error in file(con, "wb") : cannot
open the connection which can be #ignored

  mcmc_fit2 <- openbugs(
      my_data,
      inits,
      model.file = filename2,
      parameters.to.save = params,
      # program="OpenBUGS",
      n.chains = n_chains,
      n.iter = n_iter,
      n.burnin = n_burnin,
      n.thin = n_thin
  )

  time.end_mcmc <- Sys.time()

  duration_mcmc <- time.end_mcmc - time.start_mcmc

  mcmc_fit2
}

```

```
# BUGS step()
# y <- step(x)
# y = 0 if step(x) < 0
# y = 1 if x >= 0

```{r many_changepoint,  echo=FALSE}

rfcb_2cp <- function(my_data, cp1, cp2){

  my_data$k <- cp1
  my_data$k2 <- cp2

  CBmodel3 <- function(){
    S0 <- n_pop - 1
    I0 <- 1

    # k <- cp1
    # k2 <- cp2

    p[1] <- 1-exp(-(beta[1]*I0/n_pop))
    new_cases[1] ~ dbin(p[1], S0)
    S[1] <- S0 - new_cases[1]
    I[1] <- I0 + new_cases[1] - new_removals[1]

    for (t in 2:n_obs){

      param[t] <- 1 + step(t - k - 1) + step(t - k2 -1)
      p[t] <- 1 - exp(-(beta[param[t]]*I[t-1]/n_pop))
      new_cases[t] ~ dbin(p[t],S[t-1])
      S[t] <- S[t-1]- new_cases[t]
      I[t] <- I[t-1] + new_cases[t] - new_removals[t]
    }

    # prior
```
```

```
    for (j in 1:3){
      beta[j] ~ dlnorm(0, 5)
    }
  }

  paste(getwd())

  filename3 <- file.path(getwd(), "CBmodel3.bug")
  write.model(CBmodel3, filename3)

  n_chains=5
  n_burnin=2000
  n_iter=25000
  n_thin=50
  set.seed(1234)

  # Specify parameters to monitor:
  params <- c("beta")

  # Generate initial values for the parameters:
  inits = function(){
    list(beta=c(runif(1,0.05,0.07), runif(1,0.05,0.07),
runif(1,0.05,0.07)))
  }

  time.start_mcmc <- Sys.time()
  #this function will call bugs through R, the user must set the
correct file, where the bugs #executable is inside
  #If you choose debug == TRUE the Winbugs will remain open, even
after the sampling is finished
  #There is a possible error Error in file(con, "wb") : cannot
open the connection which can be #ignored

  mcmc_fit3 <- openbugs(
      my_data,
      inits,
```

```
      model.file = filename3,
      parameters.to.save = params,
      # program="OpenBUGS",
      n.chains = n_chains,
      n.iter = n_iter,
      n.burnin = n_burnin,
      n.thin = n_thin
  )

  time.end_mcmc <- Sys.time()

  duration_mcmc <- time.end_mcmc - time.start_mcmc

  mcmc_fit3
}

```


```{r learn_many_changepoint,  echo=FALSE}

rfcb_2cp_learn <- function(my_data, cp1, cp2, cp_margin=30){

  # Express confidence in change point via the margin parameter
  my_data$cp1_lb <- cp1 - cp_margin
  my_data$cp1_ub <- cp1 + cp_margin

  my_data$cp2_lb <- cp2 - cp_margin
  my_data$cp2_ub <- cp2 + cp_margin


  CBmodel4 <- function(){
    S0 <- n_pop - 1
    I0 <- 1

    p[1] <- 1-exp(-(beta[1]*I0/n_pop))
```

```
  new_cases[1] ~ dbin(p[1], S0)
  S[1] <- S0 - new_cases[1]
  I[1] <- I0 + new_cases[1] - new_removals[1]

  for (t in 2:n_obs){

    param[t] <- 1 + step(t - k - 1) + step(t - k2 - 1)
    p[t] <- 1 - exp(-(beta[param[t]]*I[t-1]/n_pop))
    new_cases[t] ~ dbin(p[t],S[t-1])
    S[t] <- S[t-1]- new_cases[t]
    I[t] <- I[t-1] + new_cases[t] - new_removals[t]
  }

  # prior

  # Ro parameteres
  for (j in 1:3){
    beta[j] ~ dlnorm(0, 5)
  }


  k ~ dunif(cp1_lb,cp1_ub)
  k2 ~ dunif(cp2_lb,cp2_ub)

  # prior on number of change points
  # n_points ~ dunif(3,8)

  # changepoint priors
  # for (k in 1:n_points){
  #   change_points[k] ~ dunif((k-1)*90, (k-1)*90+20)
  # }

}

paste(getwd())
```

```
  filename4 <- file.path(getwd(), "CBmodel4.bug")
  paste(filename4)
  write.model(CBmodel4, filename4)

  n_chains=5
  n_burnin=5000
  n_iter=100000
  n_thin=50
  set.seed(1234)

  # Specify parameters to monitor:
  params <- c("beta", "k", "k2")

  # Generate initial values for the parameters:
  inits = function(){
    list(beta=c(runif(1,0.05,0.07), runif(1,0.05,0.07),
runif(1,0.05,0.07)), k=runif(1,my_data$cp1_lb,my_data$cp1_ub),
k2=runif(1,my_data$cp2_lb,my_data$cp2_ub))
  }

  time.start_mcmc <- Sys.time()
  #this function will call bugs through R, the user must set the
correct file, where the bugs #executable is inside
  #If you choose debug == TRUE the Winbugs will remain open, even
after the sampling is finished
  #There is a possible error Error in file(con, "wb") : cannot
open the connection which can be #ignored

  mcmc_fit4 <- openbugs(
      my_data,
      inits,
      model.file = filename4,
      parameters.to.save = params,
      # program="OpenBUGS",
      n.chains = n_chains,
      n.iter = n_iter,
```

```
      n.burnin = n_burnin,
      n.thin = n_thin
  )

  time.end_mcmc <- Sys.time()

  duration_mcmc <- time.end_mcmc - time.start_mcmc

 mcmc_fit4
}

```

```{r, fit-models}

fit_all <- function(my_data, cp1, cp2, country){

  # Fit all the Models
  print(country)

  print("Model 1")
  mcmc_fit <- rfcb_nocp(my_data)
  print(mcmc_fit, digits = 3)
  diagnostic_plots(mcmc_fit, title=country, model_num="1")

  print("Model 2")
  mcmc_fit2 <- rfcb_1cp(my_data,cp1)
  print(mcmc_fit2, digits = 3)
    diagnostic_plots(mcmc_fit2, title=country, model_num="2")

  print("Model 3")
  mcmc_fit3 <- rfcb_2cp(my_data, cp1, cp2)
  print(mcmc_fit3, digits = 3)
    diagnostic_plots(mcmc_fit3, title=country, model_num="3")

  print("Model 4")
```

```
  mcmc_fit4 <- rfcb_2cp_learn(my_data, cp1, cp2, cp_margin=30)
  print(mcmc_fit4, digits = 3)
  diagnostic_plots(mcmc_fit4, title=country, model_num="4")

  return(c(mcmc_fit, mcmc_fit2, mcmc_fit3, mcmc_fit4))
}

```


```{r diagnostic-plots}

#### DIAGNOSTICS
# # Produce html file with trace, density, and autocorrelation
plots. The files are displayed in the default internet browser
# mcmcplot(mcmc_results4)
# diagnostic_plots(mcmc_fit4, country="country",
model_num="lala")
diagnostic_plots <- function(mcmc_obj, country="country",
model_num="lala"){
  plot(mcmc_obj, display.parallel = TRUE) # gives a summary plot
of parameters and credible intervals

  mcmc_obj_list <- as.mcmc.list(mcmc_obj)


  fname <- file.path(getwd(), paste0(country, "_model",
model_num, "_ac"))
  png(fname)
  autocorr.plot(mcmc_obj_list)
  dev.off()

  fname <- file.path(getwd(), paste0(country, "_model",
model_num, "_density"))
  png(fname)
  denplot(mcmc_obj_list, parms = c("deviance","beta", "k", "k2"))
```

```
  dev.off()

  fname <- file.path(getwd(), paste0(country, "_model",
model_num, "_trace"))
  png(fname)
  traplot(mcmc_obj_list, parms = c("deviance","beta", "k", "k2"))
  dev.off()

}
```

```{r, run-all}

# RUN EVERYTHING
lu_models <- fit_all(lu_data, lu_cp1, lu_cp2, "LU")
cz_models <- fit_all(cz_data, cz_cp1, cz_cp2, "CZ")
dk_models <- fit_all(dk_data, dk_cp1, dk_cp2, "DK")

save.image(file='learn_many_changepoint.RData')
```