Student’s Name:

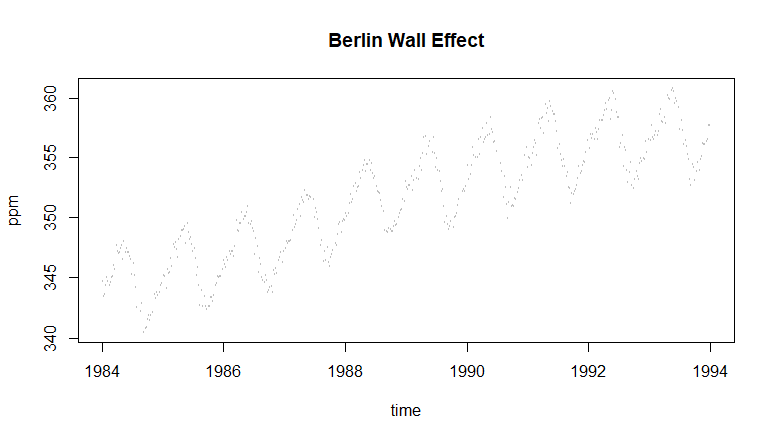
Instructor’s Name:

Course:

Date:

Carbon Dioxide

The quantity of carbon dioxide in the atmosphere is determined by a wide range of facilities which produce it. The main source atmospheric CO2 in the world is pollution from industries and other human activities. As such, one would expect that factors which affect these human activities directly affect the concentration of CO2 in the atmosphere. From a visual analysis of the trend in the carbon dioxide concentration between 1984 and 1994, one can detect the impact of the fall of the Berlin Wall on the production of this gas. The trend shows a shallower increase in the concentration of CO2 in the atmosphere after 1989 as compared to the half decade before. This most likely resulted from the reduced industrial production in eastern Europe and the Soviet Union. The graph of the carbon dioxide concentration in this decade is shown below.



This year also showed a reduced concentration of carbon dioxide. From comparing the sinusoid for 2020 with other years, the lower end dipped lower than the others and then rose with a smaller gradient towards the end of the year, unlike other years.

Further analysis using a Bayesian Semi-Parametric model confirmed these results. This model was used because it was greatly advantageous given its ease of fitting. It also took very little time to carry out the computation using the INLA() function in R. The model required that one make very explicit assumptions. These assumptions were made through forming a prior. With a neutral prior, the data was the primary influence on the model and the results it gave. Finally, the thorough process of drawing inferences using Bayesian methods provided very reliable results. The model is shown in the table below.

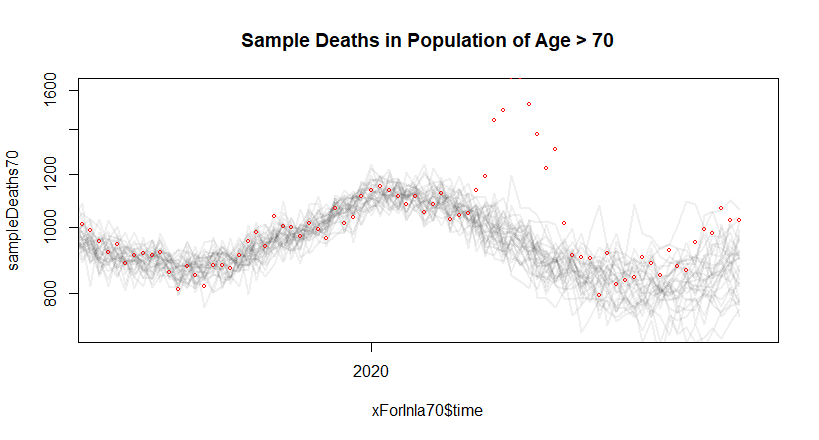
Fixed effects:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mean | sd | 0.025quant | 0.5quant | 0.975quant | mode |
| (Intercept) | 5.909 | 1.349 | 3.261 | 5.909 | 8.555 | 5.909 |
| sin12 | 0.008 | 0.125 | -0.237 | 0.008 | 0.254 | 0.008 |
| cos12 | -0.003 | 0.126 | -0.250 | -0.003 | 0.244 | -0.003 |
| sin6 | -0.002 | 0.032 | -0.065 | -0.002 | 0.062 | -0.002 |
| cos6 | 0.002 | 0.032 | -0.062 | 0.002 | 0.065 | 0.002 |

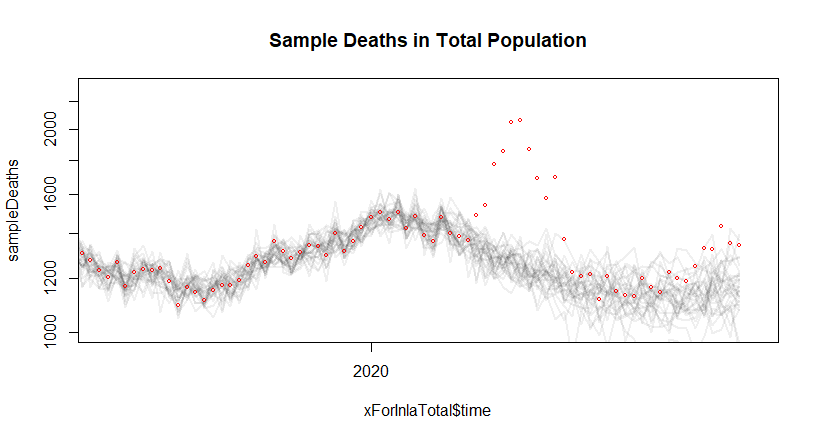
Coronavirus Deaths

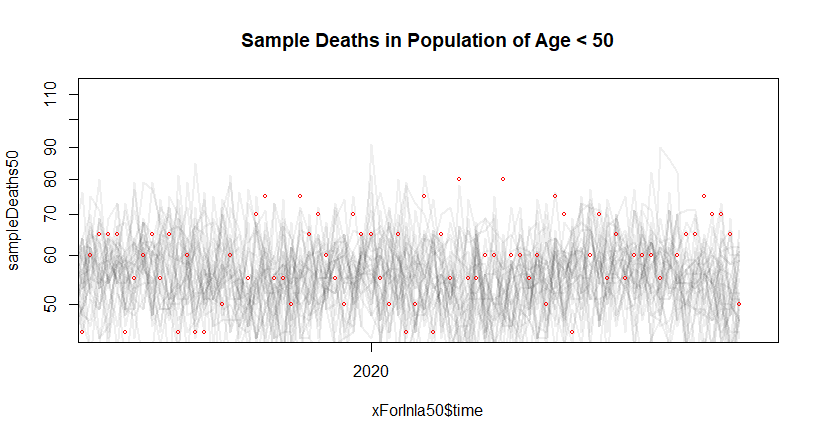
The coronavirus disease 2019 (COVID-19) is estimated to affect older people and people with pre-existing illnesses more severely than younger people and healthy individuals. As such, one would expect that more casualties of people over the age of 70 than the rest of the population. True to the fact, under the circumstances of the first wave of the disease, this demographic was the most severely affected. One would hypothesize that since the second wave came after the lifting of lockdowns and opening of the economy, the younger population would be more affected since they are not as keen as they were in following the COVID-19 control measures.

This study used a Bayesian Semi-Parametric model to assess whether the data available backed this hypothesis. The data showed that the first wave of COVID-19 did affect the older generation, with a substantial increase in the deaths of people in this demographic. This increase can be seen between the months of March and May when the pandemic was the most severe. The graph below is a plot of the number of deaths against time. It illustrates that the increase in deaths in the second quarter of 2020



However, the second wave of COVID-19 which started in September has not had significant impacts on the population under the age of 50. This population experienced no significant increase in deaths since the beginning of 2020. It maintained its original trend, showing that although the deaths in the demographic are increasing, the increase is not primarily caused by COVID-19. The graph below illustrates the trend of deaths of people from the ‘total’ group, followed by the graph for this information.





# Appendix 1

# Code for Question 1

library('INLA', verbose=FALSE)

library('Biobase')

library('Pmisc')

cUrl = paste0('https://scrippsco2.ucsd.edu/assets/data/atmospheric/',

'stations/flask\_co2/daily/daily\_flask\_co2\_mlo.csv')

cFile = basename(cUrl)

if(!file.exists(cFile)) download.file(cUrl, cFile)

co2s = read.table(cFile,header = FALSE, sep = ",",skip = 69, stringsAsFactors = FALSE,

col.names = c("day", "time", "junk1", "junk2", "Nflasks", "quality",

"co2"))

co2s$date = strptime(paste(co2s$day, co2s$time), format = "%Y-%m-%d %H:%M", tz = "UTC")

# remove low-quality measurements

co2s = co2s[co2s$quality == 0, ]

plot(co2s$date, co2s$co2, log = "y", cex = 0.3,

col = "#00000040", xlab = "time", ylab = "ppm")

plot(co2s[co2s$date > ISOdate(2015, 3, 1, tz = "UTC"),

c("date", "co2")], log = "y", type = "o", xlab = "time",

ylab = "ppm", cex = 0.5)

co2s$day = as.Date(co2s$date)

toAdd = data.frame(day = seq(max(co2s$day) + 3,

as.Date("2025/1/1"),by = "10 days"), co2 = NA)

co2ext = rbind(co2s[, colnames(toAdd)], toAdd)

timeOrigin = as.Date("2000/1/1")

co2ext$timeInla = round(as.numeric(co2ext$day - timeOrigin)/365.25,2)

co2ext$cos12 = cos(2 \* pi \* co2ext$timeInla)

co2ext$sin12 = sin(2 \* pi \* co2ext$timeInla)

co2ext$cos6 = cos(2 \* 2 \* pi \* co2ext$timeInla)

co2ext$sin6 = sin(2 \* 2 \* pi \* co2ext$timeInla)

# disable some error checking in INLA

mm = get("inla.models", INLA:::inla.get.inlaEnv())

if(class(mm) == 'function') mm = mm()

mm$latent$rw2$min.diff = NULL

assign("inla.models", mm, INLA:::inla.get.inlaEnv())

co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 + f(timeInla, model = 'rw2',

prior='pc.prec',

param = c(0.1, 0.5)), data = co2ext,

family='gamma', control.family = list(hyper=list(prec=list(

prior='pc.prec',

param=c(0.1, 0.5)))),

# add this line if your computer has trouble

control.inla = list(strategy='gaussian'),

control.predictor = list(compute=TRUE, link=1),

control.compute = list(config=TRUE), verbose=FALSE)

qCols = c('0.5quant','0.025quant','0.975quant')

Pmisc::priorPost(co2res)$summary[,qCols]

sampleList = INLA::inla.posterior.sample(30, co2res, selection = list(timeInla = 0))

sampleMean = do.call(cbind, Biobase::subListExtract(sampleList, "latent"))

sampleDeriv = apply(sampleMean, 2, diff)/diff(co2res$summary.random$timeInla$ID)

matplot(co2ext$day, co2res$summary.fitted.values[, qCols], type = "l", col = "black",

lty = c(1, 2, 2), log = "y", xlab = "time", ylab = "ppm")

Stime = timeOrigin + round(365.25 \* co2res$summary.random$timeInla$ID)

matplot(Stime, co2res$summary.random$timeInla[, qCols],type = "l", col = "black",

lty = c(1, 2, 2), xlab = "time", ylab = "y")

matplot(Stime[-1], sampleDeriv, type = "l", lty= 1, xaxs = "i",

col = "#00000020", xlab = "time", ylim = quantile(sampleDeriv, c(0.01, 0.995)))

forX = as.Date(c("2018/1/1", "2021/1/1"))

forX = seq(forX[1], forX[2], by = "6 months")

toPlot = which(Stime > min(forX) & Stime < max(forX))

matplot(Stime[toPlot], sampleDeriv[toPlot, ], type = "l",

lty = 1, lwd = 2, xaxs = "i", col = "#00000050",

xlab = "time", ylab = "deriv", xaxt = "n",

ylim = quantile(sampleDeriv[toPlot,], c(0.01, 0.995)))

axis(1, as.numeric(forX), format(forX, "%b%Y")

# Appendix 2

# Code for Question 2

library(INLA, verbose=FALSE)

library(reshape2)

library(RColorBrewer)

xWide = read.table(paste0("https://www.stat.gouv.qc.ca/statistiques/",

"population-demographie/deces-mortalite/",

"WeeklyDeaths\_QC\_2010-2020\_AgeGr.csv"),

sep = ";", skip = 7, col.names = c("year", "junk",

"age", paste0("w", 1:53)))

xWide = xWide[grep("^[[:digit:]]+$", xWide$year), ]

x = reshape2::melt(xWide, id.vars = c("year", "age"),

measure.vars = grep("^w[[:digit:]]+$", colnames(xWide)))

x$dead = as.numeric(gsub("[[:space:]]", "", x$value))

x$week = as.numeric(gsub("w", "", x$variable))

x$year = as.numeric(x$year)

x = x[order(x$year, x$week, x$age), ]

newYearsDay = as.Date(ISOdate(x$year, 1, 1))

x$time = newYearsDay + 7 \* (x$week - 1)

x = x[!is.na(x$dead), ]

x = x[x$week < 53, ]

plot(x[x$age == "Total", c("time", "dead")], type = "o", log = "y")

xWide2 = reshape2::dcast(x, week + age ~ year, value.var = "dead")

Syear = grep("[[:digit:]]", colnames(xWide2), value = TRUE)

Scol = RColorBrewer::brewer.pal(length(Syear), "Spectral")

matplot(xWide2[xWide2$age == "Total", Syear], type = "l",lty = 1, col = Scol)

legend("topright", col = Scol, legend = Syear, bty = "n", lty = 1, lwd = 3)

dateCutoff = as.Date("2020/3/1")

xPreCovid = x[x$time < dateCutoff, ]

xPostCovid = x[x$time >= dateCutoff, ]

toForecast = expand.grid(age = unique(x$age), time = unique(xPostCovid$time),

dead = NA)

xForInla = rbind(xPreCovid[, colnames(toForecast)],

toForecast)

xForInla = xForInla[order(xForInla$time, xForInla$age),]

xForInla$timeNumeric = as.numeric(xForInla$time)

xForInla$timeForInla = (xForInla$timeNumeric - as.numeric(as.Date("2015/1/1")))/365.25

xForInla$timeIid = xForInla$timeNumeric

xForInla$sin12 = sin(2 \* pi \* xForInla$timeNumeric/365.25)

xForInla$sin6 = sin(2 \* pi \* xForInla$timeNumeric \*

2/365.25)

xForInla$cos12 = cos(2 \* pi \* xForInla$timeNumeric/365.25)

xForInla$cos6 = cos(2 \* pi \* xForInla$timeNumeric \*2/365.25)

xForInlaTotal= xForInla[xForInla$age == 'Total', ]

res = inla(dead ~ sin12 + sin6 + cos12 + cos6 +

f(timeIid, prior='pc.prec', param= c(log(1.2), 0.5)) + f(timeForInla,

model = 'rw2',

prior='pc.prec',

param= c(0.01, 0.5)),

data=xForInlaTotal,

control.predictor = list(compute=TRUE, link=1),

control.compute = list(config=TRUE),

# control.inla = list(fast=FALSE, strategy='laplace'),

family='poisson')

qCols = paste0(c(0.5, 0.025, 0.975), "quant")

rbind(res$summary.fixed[, qCols], Pmisc::priorPostSd(res)$summary[,qCols])

matplot(xForInlaTotal$time, res$summary.fitted.values[,qCols], type= "l",

ylim = c(1000, 1800), lty = c(1,2,2),col="black",log="y")

points(x[x$age == "Total", c("time", "dead")], col = "red", cex = 0.4)

matplot(xForInlaTotal$time, res$summary.random$timeForInla[,

c("0.5quant", "0.975quant",

"0.025quant")], type = "l",

lty = c(1, 2, 2), col = "black", ylim = c(-1, 1) \*0.1)

sampleList = INLA::inla.posterior.sample(30, res, selection = list(Predictor = 0))

sampleIntensity = exp(do.call(cbind, Biobase::subListExtract(sampleList,"latent")))

sampleDeaths = matrix(rpois(length(sampleIntensity),

sampleIntensity), nrow(sampleIntensity),

ncol(sampleIntensity))

matplot(xForInlaTotal$time, sampleDeaths, col = "#00000010", lwd = 2, lty = 1,

type = "l", log = "y")

points(x[x$age == "Total", c("time", "dead")], col = "red", cex = 0.5)

matplot(xForInlaTotal$time, sampleDeaths, col = "#00000010",

lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1",

"2020/11/1")),

ylim = c(1, 2.3) \* 1000)

points(x[x$age == "Total", c("time", "dead")], col = "red", cex = 0.5)

title(main = "Sample Deaths in Total Population")

xPostCovidTotal = xPostCovid[xPostCovid$age == "Total", ]

xPostCovidForecast = sampleDeaths[match(xPostCovidTotal$time, xForInlaTotal$time), ]

excessDeaths = xPostCovidTotal$dead - xPostCovidForecast

matplot(xPostCovidTotal$time, xPostCovidForecast, type = "l", ylim = c(1000, 2200),

col = "black")

points(xPostCovidTotal[, c("time", "dead")], col = "red")

matplot(xPostCovidTotal$time, excessDeaths, type = "l",

lty = 1, col = "#00000030")

excessDeathsSub = excessDeaths[xPostCovidTotal$time > as.Date("2020/03/01") &

xPostCovidTotal$time < as.Date("2020/06/01"), ]

excessDeathsInPeriod = apply(excessDeathsSub, 2, sum)

round(quantile(excessDeathsInPeriod))

round(quantile(excessDeaths[nrow(excessDeaths), ]))

head(x)

#FIT MODEL FOR DEATHS OF AGE >70

xForInla70= xForInla[xForInla$age == '70 years old and over', ]

res70 = inla(dead ~ sin12 + sin6 + cos12 + cos6 +

f(timeIid, prior='pc.prec', param= c(log(1.2), 0.5))

+ f(timeForInla, model = 'rw2', prior='pc.prec', param= c(0.01, 0.5)),

data=xForInla70,

control.predictor = list(compute=TRUE, link=1),

control.compute = list(config=TRUE),

# control.inla = list(fast=FALSE, strategy='laplace'),

family='gamma')

qCols70 = paste0(c(0.5, 0.025, 0.975), "quant")

rbind(res$summary.fixed[, qCols], Pmisc::priorPostSd(res)$summary[,qCols])

matplot(xForInla70$time, res70$summary.fitted.values[,qCols70], type= "l",

ylim = c(500, 1800), lty = c(1,2,2),col="black",log="y")

points(x[x$age == "70 years old and over", c("time", "dead")], col = "red", cex = 0.4)

matplot(xForInla70$time, res70$summary.random$timeForInla[,

c("0.5quant", "0.975quant",

"0.025quant")], type = "l",

lty = c(1, 2, 2), col = "black", ylim = c(-1, 1) \*0.1)

sampleList70 = INLA::inla.posterior.sample(30, res70, selection = list(Predictor = 0))

sampleIntensity70 = exp(do.call(cbind, Biobase::subListExtract(sampleList70,

"latent")))

sampleDeaths70 = matrix(rpois(length(sampleIntensity70),

sampleIntensity70), nrow(sampleIntensity70),

ncol(sampleIntensity70))

matplot(xForInla70$time, sampleDeaths70, col = "#00000010", lwd = 2, lty = 1,

type = "l", log = "y")

points(x[x$age == "70 years old and over", c("time", "dead")], col = "red", cex = 0.5)

title(main = "Deaths in Population of Age > 70")

matplot(xForInla70$time, sampleDeaths70, col = "#00000010",

lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1",

"2020/11/1")),

ylim = c(1, 2.3) \* 700)

points(x[x$age == "70 years old and over", c("time", "dead")], col = "red", cex = 0.5)

title(main = "Sample Deaths in Population of Age > 70")

xPostCovid70 = xPostCovid[xPostCovid$age == "70 years old and over", ]

xPostCovidForecast70 = sampleDeaths70[match(xPostCovid70$time, xForInla70$time), ]

excessDeaths70 = xPostCovid70$dead - xPostCovidForecast70

matplot(xPostCovid70$time, xPostCovidForecast70, type = "l", ylim = c(700, 2200),

col = "black")

points(xPostCovid70[, c("time", "dead")], col = "red")

title(main = "Projected Deaths in Population of Age > 70")

matplot(xPostCovid70$time, excessDeaths70, type = "l",

lty = 1, col = "#00000030")

excessDeathsSub70 = excessDeaths70[xPostCovid70$time > as.Date("2020/03/01") &

xPostCovid70$time < as.Date("2020/06/01"), ]

excessDeathsInPeriod70 = apply(excessDeathsSub70, 2, sum)

round(quantile(excessDeathsInPeriod70))

round(quantile(excessDeaths70[nrow(excessDeaths70), ]))

#FIT MODEL FOR DEATHS OF AGE <50

xForInla50= xForInla[xForInla$age == '0-49 years old', ]

res50 = inla(dead ~ sin12 + sin6 + cos12 + cos6 +

f(timeIid, prior='pc.prec', param= c(log(1.2), 0.5))

+ f(timeForInla, model = 'rw2', prior='pc.prec', param= c(0.01, 0.5)),

data=xForInla50,

control.predictor = list(compute=TRUE, link=1),

control.compute = list(config=TRUE),

# control.inla = list(fast=FALSE, strategy='laplace'),

family='gamma')

qCols50 = paste0(c(0.5, 0.025, 0.975), "quant")

rbind(res$summary.fixed[, qCols], Pmisc::priorPostSd(res)$summary[,qCols])

matplot(xForInla50$time, res50$summary.fitted.values[,qCols50], type= "l",

ylim = c(50, 110), lty = c(1,2,2),col="black",log="y")

points(x[x$age == "0-49 years old", c("time", "dead")], col = "red", cex = 0.4)

matplot(xForInla50$time, res50$summary.random$timeForInla[,

c("0.5quant", "0.975quant",

"0.025quant")], type = "l",

lty = c(1, 2, 2), col = "black", ylim = c(-1, 1) \*0.1)

sampleList50 = INLA::inla.posterior.sample(30, res50, selection = list(Predictor = 0))

sampleIntensity50 = exp(do.call(cbind, Biobase::subListExtract(sampleList50,

"latent")))

sampleDeaths50 = matrix(rpois(length(sampleIntensity50),

sampleIntensity50), nrow(sampleIntensity50),

ncol(sampleIntensity50))

matplot(xForInla50$time, sampleDeaths50, col = "#00000010", lwd = 2, lty = 1,

type = "l", log = "y")

points(x[x$age == "0-49 years old", c("time", "dead")], col = "red", cex = 0.5)

title(main = "Deaths in Population of Age < 50")

matplot(xForInla50$time, sampleDeaths50, col = "#00000010",

lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1",

"2020/11/1")),

ylim = c(1, 2.5) \* 45)

points(x[x$age == "0-49 years old", c("time", "dead")], col = "red", cex = 0.5)

title(main = "Sample Deaths in Population of Age < 50")

xPostCovid50 = xPostCovid[xPostCovid$age == "0-49 years old", ]

xPostCovidForecast50 = sampleDeaths50[match(xPostCovid50$time, xForInla50$time), ]

excessDeaths50 = xPostCovid50$dead - xPostCovidForecast50

matplot(xPostCovid50$time, xPostCovidForecast50, type = "l", ylim = c(30, 100),

col = "black")

points(xPostCovid50[, c("time", "dead")], col = "red")

title(main = "Projected Deaths in Population of Age < 50")

matplot(xPostCovid50$time, excessDeaths50, type = "l",

lty = 1, col = "#00000030")

excessDeathsSub50 = excessDeaths50[xPostCovid50$time > as.Date("2020/03/01") &

xPostCovid50$time < as.Date("2020/06/01"), ]

excessDeathsInPeriod50 = apply(excessDeathsSub50, 2, sum)

round(quantile(excessDeathsInPeriod50))

round(quantile(excessDeaths50[nrow(excessDeaths50), ]))