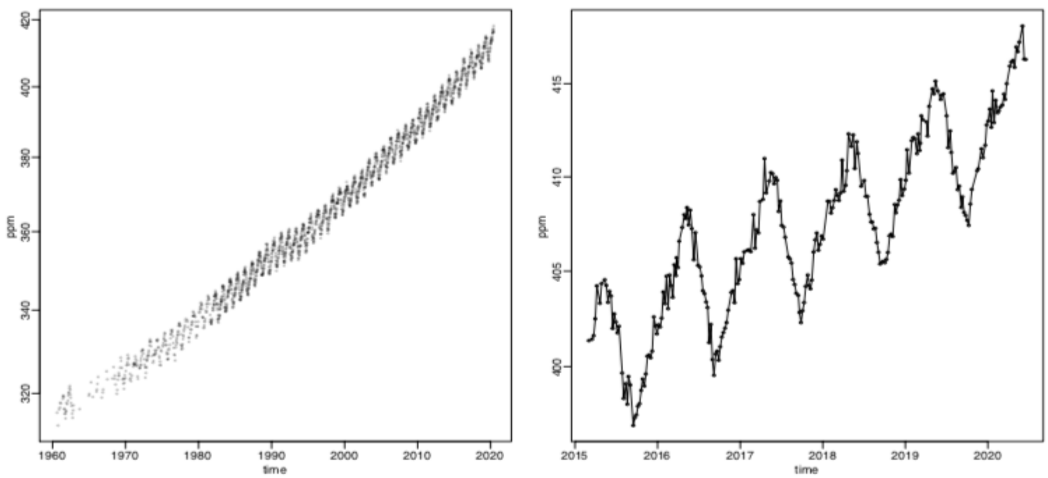
STA442 A3 - Luo Manyi - 1003799419

Question 1 – CO2

Introduction

This report is constructed for investigating whether the CO2 level appears to be impacted by two historical events: 1. the fall of the Berlin wall in November 1989 years ago; 2. the global lockdown during the COVID-19 pandemic starting in February 2020. The general trend for CO2 fluctuations within different time frame can be illustrated below, and further analysis about the estimated smoothed trend of CO2 before and after the events are demonstrated though plots.



(a) all (b) recent

Plots for the general trend in CO2 fluctuations at Mauna Loa Observatory, Hawaii (in 2 different scales)

Statistical Method

Given the fact that the CO2 data is considered as continuous and positive, the Gamma distribution should be followed in this scenario and a statistical model corresponding to the data can be expressed as below. Note that, since we observe periodical fluctuations of CO2 general trend, we adopt sinusoidal basis functions toward the model, including twelve months and six months, in order to obtain the widespread seasonal fluctuations in a precise manner.

$$Y_i \sim Gamma(\theta, \lambda_i/\theta)$$
$$log(\lambda_i) = X_i * \beta + f(t_i)$$
$$X_{i1} = \cos(2\pi t_i/365.25)$$
$$X_{i2} = \sin(2\pi t_i/365.25)$$
$$X_{i3} = \cos(2\pi t_i/182.6)$$
$$X_{i4} = \sin(2\pi t_i/182.6)$$

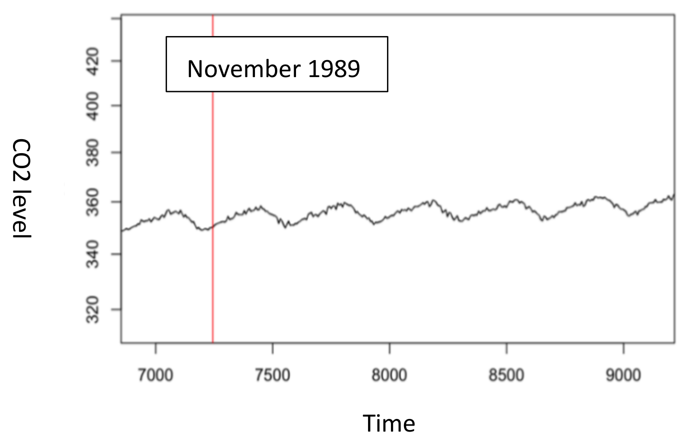
Here, Y_i represents each individual level (ith) of the CO2 data, which has a distribution showed in the plots above; whereas λ_i and θ represents the rate and mean of CO2 level in this data set. Moreover, $f(t_i)$ indicates the random variable for time, and $1/365.25$ as well as $1/182.6$ represent frequencies for either twelve months or six months cycle. Considering prior distributions, the prior for random variation indicates changes by log scale 0.1; whereas the prior for random walk indicates changes of log rate slope by 0.1 from year to year. The penalized complexity prior is selected, as the intended research questions are to compare the variability of CO2 before and after the events, such that:

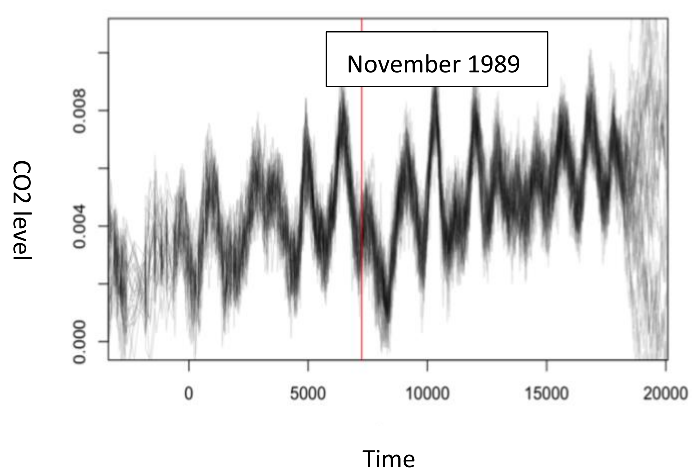
$$Prob(\delta^2 > 0.1) = 0.5$$

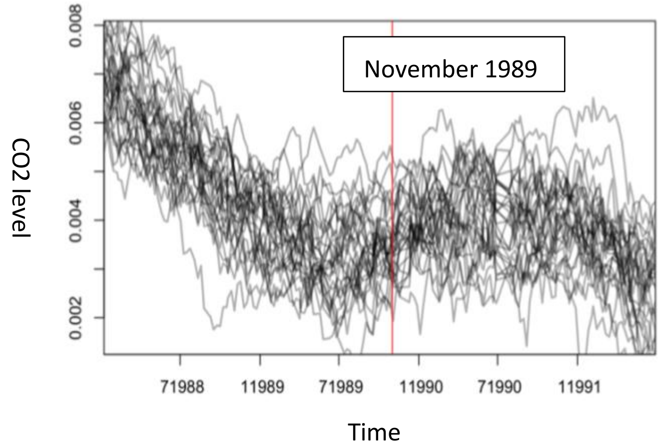
<u>Results</u>

NOTE: Since R versions and system updates transformed the time scale (x axis), I'll relabel the desired time period using red line to clarify three different time scales.

The trend for CO2 fluctuations during the fall of the Berlin wall in November 1989 are represented by plots below, which indicate the relationship between time (X axis) and CO2 level (Y axis). Through analyzing the smooth trend of CO2 from all three plots, we can't observe a steeper change in data variations after the desired time point, meaning that the fall in industrial production in the Soviet Union and Eastern Europe didn't place a huge influence on CO2 level. Specifically, we can observe cyclical trends of CO2 from the first and the second graphs, and the variability of CO2 level increases drastically through observing the peaks and troughs before and after the fall of the Berlin wall specifically in the second graph, as the overall amplitude of increasing in CO2 level was shut down.

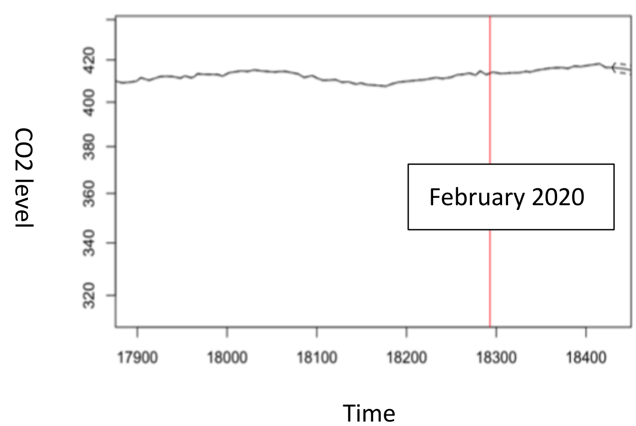


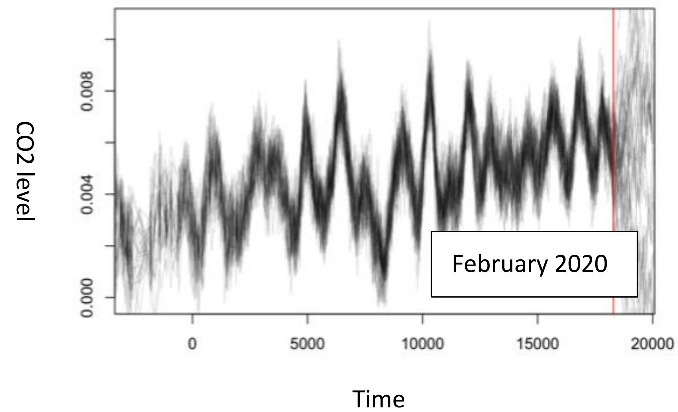


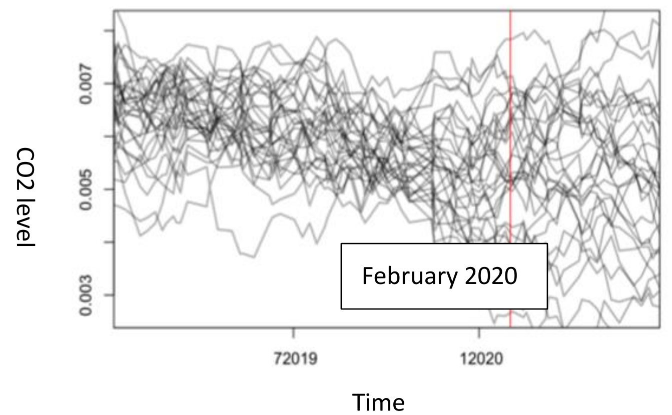


Plots for CO2 fluctuations during the fall of the Berlin wall in November 1989 (in 3 different scales)

The trend for CO2 fluctuations during the global lockdown during the COVID-19 pandemic starting in February 2020 are represented by plots below as well, which indicate the relationship between time (X axis) and CO2 level (Y axis). Through observing the first plot, we can't observe any significant change in data variations after the desired time point, even from the aspect of entire time frame. This means that the shutting down much of the global economy also didn't place a huge influence on CO2 level, which might be caused from the fact that productions and consumptions weren't stagnated completely, but rather transformed to other formats, for example: the option for remote work and store or restaurant pick up. Also, from the second and third graphs, it crucial to note that there is a huge disturbance in CO2 level after the February 2020 shut down, which might be resulted from the fact that the complete data set for 2020 was not updated yet, which potentially leads to this variation.







Plots for CO2 fluctuations during the COVID-19 pandemic starting in February 2020 (in 3 different scales)

Therefore, we can conclude that the CO2 level doesn't appear to be impacted by two historical events: 1. the fall of the Berlin wall in November 1989 years ago; 2. the global lockdown during the COVID-19 pandemic starting in February 2020, since both of the events didn't lead to significant drop in the global economic condition.

Question 2 – COVID Death

Introduction

This report is constructed in order to find out whether real-life mortality data during COVID supported two hypotheses claimed by government official: 1. Deaths amongst the elderly in the spring (March, April and May) were well above the historical averages, whereas the under 50's had deaths in line with previous years; 2. In the most recent death data, there is an increase in deaths in the under 50's whereas the over 70's have no more deaths than would be expected pre-COVID.

Statistical Method

Since the number of deaths associated with different age groups follow the Poisson distribution, which is a discrete and countable distribution, a statistical model corresponding to the data set can be expressed as below. Similarly, in order to accurately display the periodical fluctuations of death trend, we adopt sinusoidal basis functions toward the model, including twelve months and six months, in order to obtain the widespread seasonal fluctuations in a precise manner.

$$Y_i \sim Poisson(O_i\ \lambda_i)$$
$$log(\lambda_i) = X_i * \beta + U(t_i) + V_i$$
$$U(t_i) \sim RW2(0, \delta_U^2)$$
$$V_i \sim N(0, \delta_V^2)$$
$$X_{i1} = \cos\ (2\pi t_i/365.25)$$
$$X_{i2} = \sin\ (2\pi t_i/365.25)$$
$$X_{i3} = \cos\ (2\pi t_i/182.6)$$
$$X_{i4} = \sin\ (2\pi t_i/182.6)$$

Here, Y_i represents the number of deaths of the COVID data, which has a Poisson distribution; whereas O_i represents the time variable (years), $U(t_i)$ indicates a second order random walk, and V_i indicates the overdispersion for this model. Similarly, $1/365.25$ as well as $1/182.6$ represent frequencies for either twelve months or six months cycle. Considering prior distributions, the prior for random variation indicates changes by log scale 1.1; whereas the prior for random walk indicates changes of log rate slope by 0.01 from year to year. The penalized complexity prior is selected, as the intended research questions are to compare the variability of deaths before and after the events, such that:

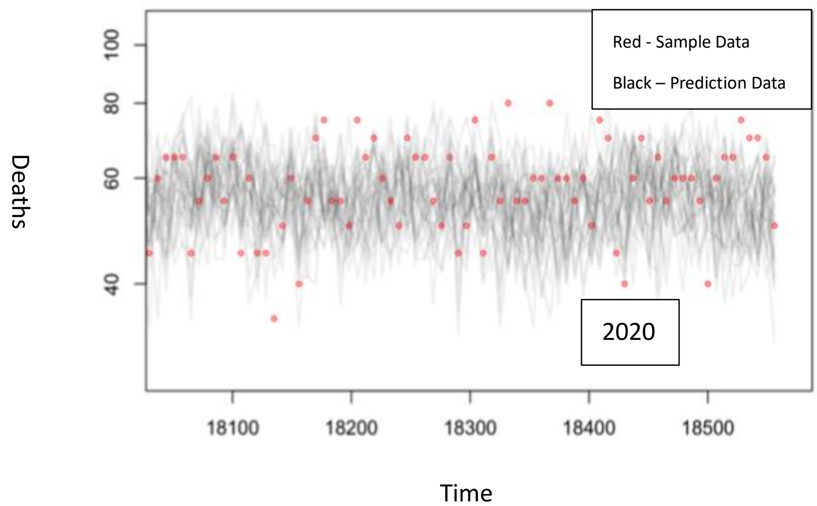
$$Prob(\delta_U^2 > 0.01) = 0.5, Prob(\delta_V^2 > \log\ (1.1)) = 0.5$$

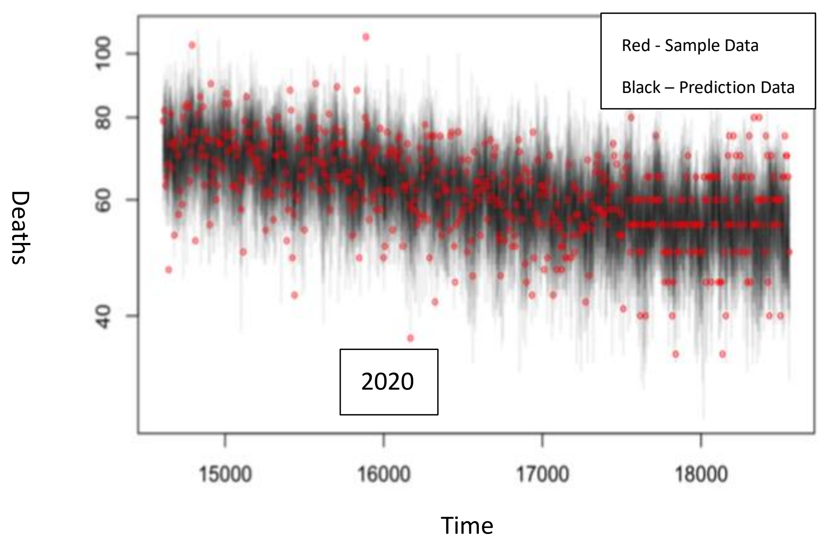
<u>Results</u>

NOTE: Since R versions and system updates transformed the time scale (x axis), I'll relabel the desired time periods to clarify three different time scales.

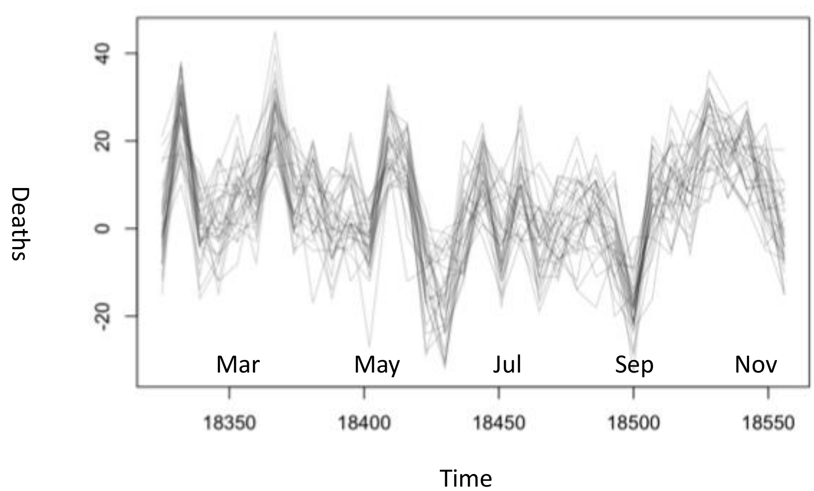
In order to test out whether the under 50's had deaths in line with previous years, and in the most recent death data, there is an increase in deaths in the under 50's, we first create plots for sampled real-life data and prediction data for those deaths under 50's with different scales, which indicate the relationship between time (X axis) and the number of death (Y axis). We can then plot the excess deaths for those under 50's to present a clearer illustration for the second hypothesis.

From the first two plots below, which corresponds to the first hypothesis, we can clearly observe that the red samples only slightly deviate from the black general death trend, meaning that real-life sampled deaths from those under 50's during COVID aligns with the prediction of usual deaths occur during this time period. Therefore, we can conclude that the under 50's had deaths in line with previous years. Also, through observing the plot for excess deaths, given the knowledge that the peaks in excess deaths start from spring (March, April and May), we can observe consecutive months follow almost align trends with previous months, even showing a slightly decreasing trend in recent peaks. As a result, we can also conclude toward the hypothesis that, the age group under 50's is less impacted by COIVD after spring, which has less to do with the fact that young people, primarily university undergraduates, acting irresponsibly.





Plots for sampled real-life data and prediction data for those deaths under 50's (in 2 different scales)

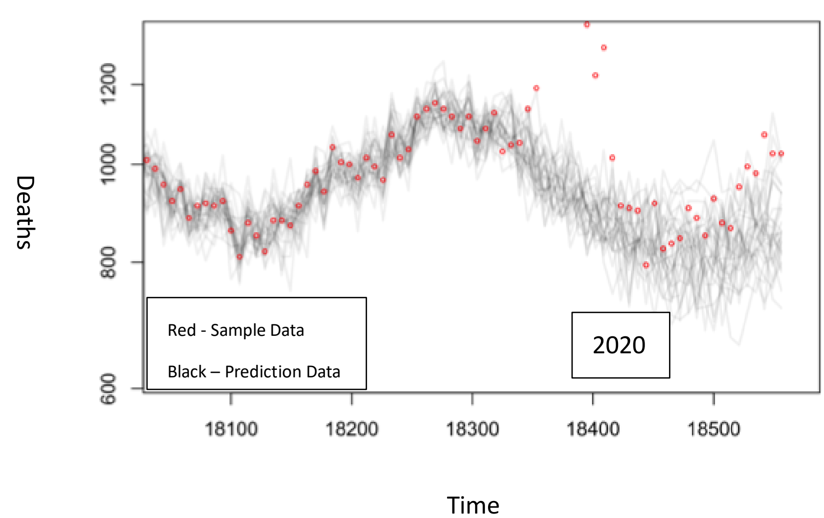


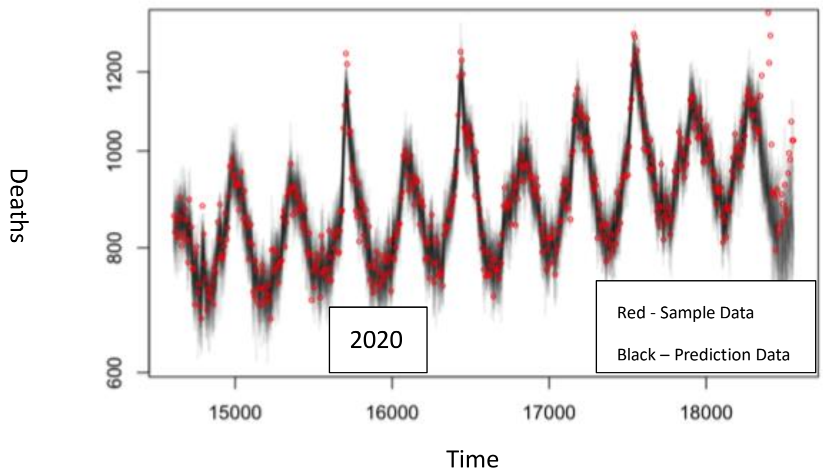
Plots for excess deaths data for those under 50's during months in 2020

In order to test out whether the deaths amongst the elderly in the spring (March, April and May) were well above the historical averages, and in the most recent death data, the over 70's have no more deaths than would be expected pre-COVID, we first create plots for sampled real-life data and prediction data for those deaths over 70's with different scales, which indicate the relationship between time (X axis) and the number of death (Y axis). We can then plot the excess deaths for those over 70's to present a clearer illustration for the second hypothesis.

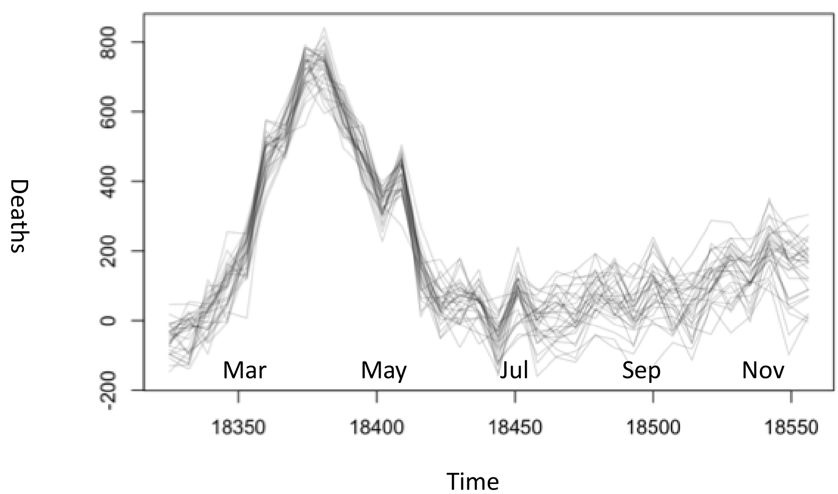
From the first plot below, which corresponds to the first hypothesis, we can clearly observe that the red posterior samples deviate a lot from the black general death trend, meaning that real-life sampled deaths from those over 70's during COVID dramatically surpasses the prediction of usual deaths occur during this time period. Therefore, we can conclude that the over 70's had deaths well above the historical averages during March, April and May. Moreover, from the second plot, we can observe that even during later months of 2020, red samples for over 70's still indicate some deviations toward the black general death trend, meaning that deaths amongst the elderly reach an all-time high level.

Also, through observing the plot for excess deaths, there exists a significant peak for those over 70's during spring (March, April and May). Even though we can observe a drastic decrease in number of deaths follow the initial peak, there is a gradual increasing trend above predictions in consecutive months, specifically after the spring of COVID first occurrence, which might be related with the fact that young people, primarily university undergraduates, acting irresponsibly.





Plots for sampled real-life data and prediction data for those deaths over 70's (in 2 different scales)



Plots for excess deaths data for those over 70's during months in 2020

Through investigating these two sets of plots, we can conclude that deaths amongst the elderly in the spring (March, April and May) were indeed well above the historical averages, whereas the under 50's had deaths in line with previous years as well. However, we can't make a conclusion toward the second hypothesis, as we fail to observe a drastic increase in deaths in the under 50's, and for those over 70's still has more deaths than would be expected comparing to pre-COVID.

```
```Appendix
#Question 1 - CO2
cUrl = paste0("http://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$time), format = "%Y-%m-%d %H:%M", tz = "UTC")
co2s = co2s[co2s$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)
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())
co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 + f(timeInla, model = 'rw2', prior='pc.prec', param = c(0.1, 0.5)),
data = co2ext, family='gamma', control.family = list(hyper=list(prec=list(prior='pc.prec', param=c(0.1, 0.5)))),
control.inla = list(strategy='gaussian'), control.predictor = list(compute=TRUE, link=1), control.compute =
list(config=TRUE), verbose=TRUE)
Pmisc::priorPost(co2res)$summary[,qCols]
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("Biobase")
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$timeInla[, qCols], type = "l", col = "black", lty = c(1, 2, 2), xlab = "time",
ylab = "y")
matplot(Stime[-1], sampleDeriv, type = "l", lty = 1, xaxs = "i", col = "#00000020", xlab = "time", ylab = "deriv",
ylim = quantile(sampleDeriv, c(0.01, 0.995)))
forX = as.Date(c("2018/1/1", "2021/1/1"))
forX = seq(forX[1], forX[2], by = "6 months")
toPlot = which(Stime > min(forX) & Stime < max(forX))
matplot(Stime[toPlot], sampleDeriv[toPlot, ], type = "l", lty = 1, lwd = 2, xaxs = "i", col = "#00000050",
xlab = "time", ylab = "deriv", xaxt = "n", ylim = quantile(sampleDeriv[toPlot,], c(0.01, 0.995)))
axis(1, as.numeric(forX), format(forX, "%b%Y"))
matplot(co2ext$day, co2res$summary.fitted.values[,qCols], type = "l", col = "black", lty = c(1, 2, 2), log = "y",
xlab = "time", ylab = "ppm", xlim = c(as.Date("1989-01-01"), as.Date("1994-12-31")))
abline(v = as.Date("1989-11-01"), col = "red")
Stime = timeOrigin + round(365.25 * co2res$summary.random$timeInla$ID)
matplot(Stime[-1], sampleDeriv, type = "l", lty = 1, xaxs = "i", col = "#00000020", xlab = "time", ylab = "deriv",
ylim = quantile(sampleDeriv, c(0.01, 0.995)))
abline(v = as.Date("1989-11-01"), col = "red")
forX = as.Date(c("1988/1/1", "1991/12/31"))
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"))
abline(v = as.Date("1989-11-01"), col = "red")
matplot(co2ext$day, co2res$summary.fitted.values[,qCols], type = "l", col = "black", lty = c(1, 2,2), log = "y",
xlab = "time", ylab = "ppm", xlim=c(as.Date("2019-01-01"), as.Date("2020-06-16")))
abline(v = as.Date("2020-02-01"), col = "red")
Stime = timeOrigin + round(365.25 * co2res$summary.random$timeInla$ID)
matplot(Stime[-1], sampleDeriv, type = "l", lty = 1, xaxs = "i", col = "#00000020", xlab = "time", ylab = "deriv",
ylim = quantile(sampleDeriv, c(0.01, 0.995)))
abline(v = as.Date("2020-02-01"), col = "red")
forX = as.Date(c("2019/1/1", "2020/11/22"))
forX = seq(forX[1], forX[2], by = "6 months")
toPlot = which(Stime > min(forX) & Stime < max(forX))
```
```

```r
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"))
abline(v = as.Date("2020-02-01"), col = "red")

#Question 2 - COVID Death
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(as.character(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$week < 53, ]
dateCutoff = as.Date("2020/3/1")
xPreCovid = x[x$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$timeNumeric - as.numeric(as.Date("2015/1/1")))/365.25
xForInla$timeId = 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 == '0-49 years old', ]
library(INLA, verbose=FALSE)
res = inla(dead ~ sin12 + sin6 + cos12 + cos6 + f(timeId, 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')

rbind(res$summary.fixed[, qCols], Pmisc::priorPostSd(res)$summary[, qCols])
matplot(xForInlaTotal$time, res$summary.fitted.values[, qCols], type = "l", lty = c(1, 2, 2), col = "black", log = "y")
points(x[x$age == "0-49 years old", c("time", "dead")], cex = 0.4, col = "red")
matplot(xForInlaTotal$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 == "0-49 years old", c("time", "dead")], col = "red", cex = 0.5)
matplot(xForInlaTotal$time, sampleDeaths, col = "#00000010", lwd = 2, lty = 1, type = "l", log = "y",
xlim = as.Date(c("2019/6/1", "2020/11/1")))
points(x[x$age == "0-49 years old", c("time", "dead")], col = "red", cex = 0.5)
xPostCovid50 = xPostCovid[xPostCovid$age == "0-49 years old",]
xPostCovidForecast = sampleDeaths[match(xPostCovid50$time, xForInlaTotal$time), ]
excessDeaths = xPostCovid50$dead - xPostCovidForecast
matplot(xPostCovidTotal$time, xPostCovidForecast, type = "l", col = "black")
points(xPostCovidTotal[, c("time", "dead")], col = "red")
cset2 = GET::create_curve_set(list(r=as.numeric(xPostCovid50$time), obs=xPostCovidForecast))
myEnv2 = GET::central_region(cset2, coverage = 0.75)
matplot(xPostCovid50$time, xPostCovidForecast, type = "l", lty = 1, col ="#00000030", xlab = 'time',
ylab = 'number of death', main = "Forecasted deaths and actual deaths for under 50's")
matlines(xPostCovid50$time, as.data.frame(myEnv2)[,c("lo","hi","central")], type = "l", lty = c(2,2,1),
col = "black", xlab='time', ylab='number of deaths')
points(xPostCovid50[, c("time", "dead")], col = "red")
cset = GET::create_curve_set(list(r=as.numeric(xPostCovidTotal$time), obs=excessDeaths))
myEnv = GET::central_region(cset, coverage = 0.75)
matplot(xPostCovidTotal$time, excessDeaths, type = "l", lty = 1, col ="#00000030", xlab = 'time',
ylab = 'number of death', main = "Excess deaths post COVID-19 for under 50's")
matlines(xPostCovidTotal$time, as.data.frame(myEnv)[,c("lo","hi","central")], type = "l", lty = c(2,2,1),
col = "black", xlab='time', ylab='number of deaths')
excessDeathsSub = excessDeaths[xPostCovidTotal$time > as.Date("2020/03/01")
& xPostCovidTotal$time < as.Date("2020/06/01"), ]
excessDeathsInPeriod = apply(excessDeathsSub, 2, sum)
round(quantile(excessDeathsInPeriod))
round(quantile(excessDeaths[nrow(excessDeaths), ]))
newYearsDay = as.Date(ISOdate(x$year, 1, 1))
```

```
x$time = newYearsDay + 7 * (x$week - 1)
x = x[!is.na(x$dead), ]
x = x[x$week < 53, ]
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$age == '70 years old and over', ]
library(INLA, verbose=FALSE)
res = inla(dead ~ sin12 + sin6 + cos12 + cos6 + f(timeIid, prior='pc.prec', param= c(log(1.2), 0.5)) +
f(timeForInla, model = 'rw2', prior='pc.prec', param= c(0.01, 0.5)), data=xForInlaTotal, control.predictor =
list(compute=TRUE, link=1), control.compute = list(config=TRUE), control.inla = list(fast=FALSE, strategy='laplace'),
family='poisson')
qCols = paste0(c(0.5, 0.025, 0.975), "quant")
rbind(res$summary.fixed[, qCols], Pmisc::priorPostSd(res)$summary[, qCols])
matplot(xForInlaTotal$time, res$summary.fitted.values[, qCols], type = "l", lty = c(1, 2, 2), col = "black", log = "y")
points(x[x$age == "70 years old and over", c("time", "dead")], cex = 0.5, col = "red")
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$age == "70 years old and over", c("time", "dead")], col = "red", cex = 0.5)
matplot(xForInlaTotal$time, sampleDeaths, col = "#00000010",
lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1", "2020/11/1")))
points(x[x$age == "70 years old and over", c("time", "dead")], col = "red", cex = 0.5)
xPostCovid70 = xPostCovid[xPostCovid$age == "70 years old and over",]


xPostCovidForecast2 = sampleDeaths[match(xPostCovid70$time, xForInlaTotal$time), ]
excessDeaths2 = xPostCovid70$dead - xPostCovidForecast2
matplot(xPostCovidTotal$time, xPostCovidForecast, type = "l", ylim =c(1000,2000), col = "black")
points(xPostCovidTotal[, c("time", "dead")], col = "red")
cset4 = GET::create_curve_set(list(r=as.numeric(xPostCovid70$time), obs=xPostCovidForecast2))
myEnv4 = GET::central_region(cset4, coverage = 0.75)
matplot(xPostCovid70$time, xPostCovidForecast2, type = "l", lty = 1, col ="#00000030", xlab = 'time',
ylab = 'number of death', main = "Forecasted deaths and actual deaths for over 70s")
matlines(xPostCovid70$time, as.data.frame(myEnv4)[,c("lo","hi","central")], type = "l", lty = c(2,2,1),
col = "black", xlab='time', ylab='number of death')
points(xPostCovid70[, c("time", "dead")], col = "red")
matplot(xPostCovidTotal$time, excessDeaths, type = "l", lty = 1, col = "#00000030")
cset3 = GET::create_curve_set(list(r=as.numeric(xPostCovid70$time), obs=excessDeaths2))
myEnv3 = GET::central_region(cset3, coverage = 0.75)
matplot(xPostCovid70$time, excessDeaths2, type = "l", lty = 1, col ="#00000030", xlab = 'time',
ylab = 'number of death', main = " Excess deaths post COVID-19 for over 70's ")
matlines(xPostCovid70$time, as.data.frame(myEnv)[,c("lo","hi","central")], type = "l", lty = c(2,2,1), col = "black",
xlab='time', ylab='number of death')
excessDeathsSub = excessDeaths[xPostCovidTotal$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), ])
```