STA442 Homework 3

Kevin Dang

Due 13/11/2019

CO2

In this report we investigate the impact of historic events on Carbon Dioxide concentrations in the atmosphere. The data was collected by Mauna Loa Observatory located in Hawaii.

In order to fit a generalized linear mixed model this data we need to use cosine and sine transformations on the dates to account for annual (12 months) and semi-annual (6 months) cycles due to potential seasonal effects. This adds up to four independent variables in the model, with two cosine and two sine terms for each of the two cycles. The time variable will be used as a Random Walk 2 random effect as it is a smoother fit compared to Random Walk 1 and the CO2 data follows a smooth pattern as opposed to a jagged pattern. Last but not least, the response is CO2 levels in the atmosphere. The linear mixed model is:

$$\text{CO2} = \beta_0 + \beta_1 \cdot \sin 12 + \beta_2 \cdot \cos 12 + \beta_3 \cdot \sin 6 + \beta_4 \cdot \cos 6 + b_i + e_{ij}$$

where b_i is the Random Walk 2 random effect for time, and e_{ij} are the error terms.

In Figure 1 we investigate the CO2 concentration in ppm for each historic event. The events each have their own coloured line which represents the date when the event took place and they are: the OPEC oil embargo which began in October 1973 (red), the global economic recessions around 1980-1982 (orange), the fall of the Berlin wall in November 1991 (yellow), China joining the WTO on 11 December 2001 (green), the bankruptcy of Lehman Brothers on 15 September 2008 (blue), the signing of the Paris Agreement on 12 December 2015 (purple). All of the lines have positive slopes, however the plots corresponding to the red, orange, yellow, and purple lines all have slopes that decrease after the events took place. The plots with green and blue lines have increasing slopes after the events took place. This means that the growth rate for CO2 decreased after the following events: the OPEC oil embargo which began in October 1973, the global economic recessions around 1980-1982, the fall of the Berlin wall in November 1991 and the signing of the Paris Agreement on 12 December 2015. On the flip side, the growth rate for CO2 increased after China joined the WTO on 11 December 2001 and the bankruptcy of Lehman Brothers on 15 September 2008.

In Figure 2 we can see the derivative of log ppm versus time. The derivative decreases after the events have occurred for the red, orange, yellow and purple lines. The derivative increases after the events take place for the green and blue lines. This results agrees with our analysis for Figure 1.

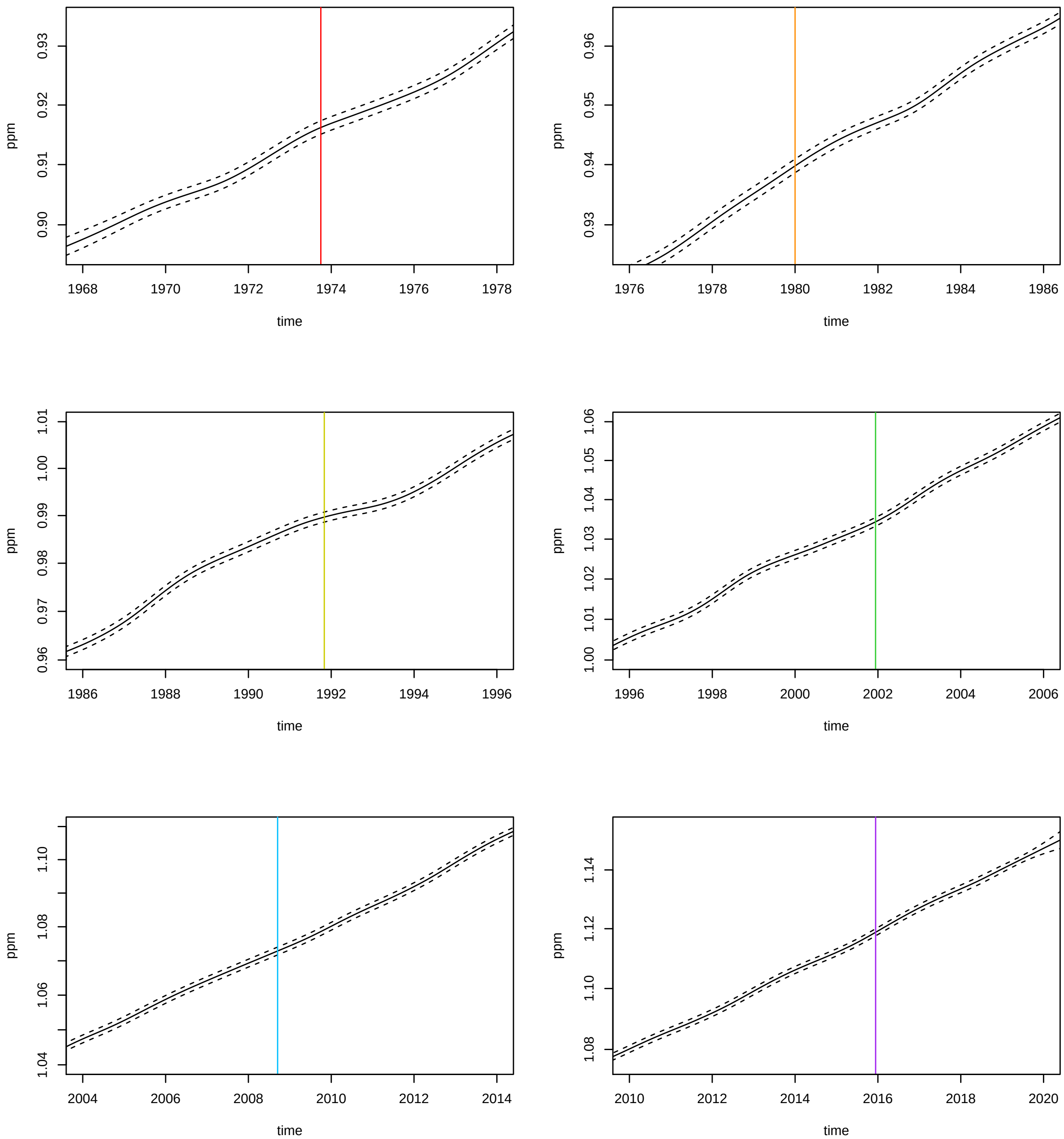


Figure 1: ppm over time for each historic event

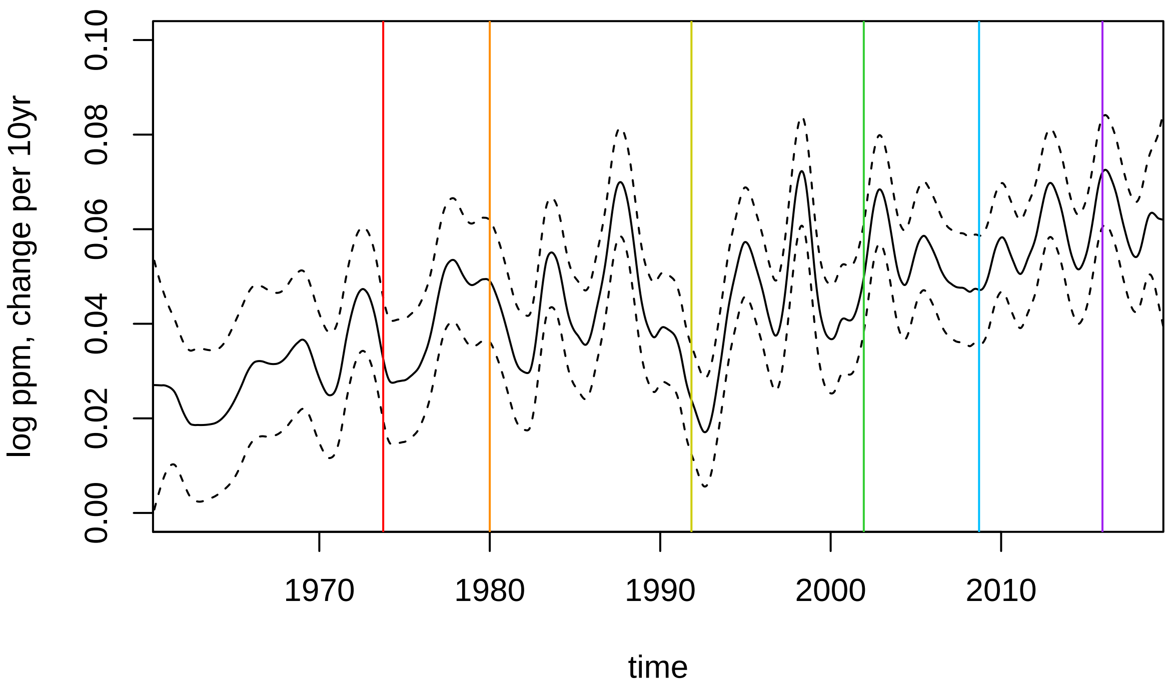


Figure 2: Time derivative of log ppm

Appendix A: CO2 Code

```r
library(INLA)

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$time), format = "%Y-%m-%d %H:%M", tz = "UTC")
# remove low-quality measurements
co2s[co2s$quality >= 1, "co2"] = NA
timeOrigin = ISOdate(1980, 1, 1, 0, 0, 0, tz = "UTC")
co2s$days = as.numeric(difftime(co2s$date, timeOrigin, units = "days"))
co2s$cos12 = cos(2 * pi * co2s$days/365.25)
co2s$sin12 = sin(2 * pi * co2s$days/365.25)
co2s$cos6 = cos(2 * 2 * pi * co2s$days/365.25)
co2s$sin6 = sin(2 * 2 * pi * co2s$days/365.25)
newX = data.frame(date = seq(ISOdate(1990, 1, 1, 0, 0, 0, tz = "UTC"), by = "1 days",
                             length.out = 365 * 30))
newX$days = as.numeric(difftime(newX$date, timeOrigin, units = "days"))
newX$cos12 = cos(2 * pi * newX$days/365.25)
newX$sin12 = sin(2 * pi * newX$days/365.25)
newX$cos6 = cos(2 * 2 * pi * newX$days/365.25)
newX$sin6 = sin(2 * 2 * pi * newX$days/365.25)

# time random effect
timeBreaks = seq(min(co2s$date), ISOdate(2025, 1, 1, tz = "UTC"), by = "14 days")
timePoints = timeBreaks[-1]
co2s$timeRw2 = as.numeric(cut(co2s$date, timeBreaks))
# derivatives of time random effect
D = Diagonal(length(timePoints)) - bandSparse(length(timePoints), k = -1)
derivLincomb = inla.make.lincombs(timeRw2 = D[-1, ])
names(derivLincomb) = gsub("^lc", "time", names(derivLincomb))
# seasonal effect
StimeSeason = seq(ISOdate(2009, 9, 1, tz = "UTC"),
                  ISOdate(2011, 3, 1, tz = "UTC"), len = 1001)
StimeYear = as.numeric(difftime(StimeSeason, timeOrigin, "days"))/365.35
seasonLincomb = inla.make.lincombs(sin12 = sin(2 * pi * StimeYear),
                                   cos12 = cos(2 * pi * StimeYear),
                                   sin6 = sin(2 * 2 * pi * StimeYear),
                                   cos6 = cos(2 * 2 * pi * StimeYear))
names(seasonLincomb) = gsub("^lc", "season", names(seasonLincomb)) # predictions
StimePred = as.numeric(difftime(timePoints, timeOrigin, units = "days"))/365.35
predLincomb = inla.make.lincombs(timeRw2 = Diagonal(length(timePoints)),
                                 `(Intercept)` = rep(1, length(timePoints)),
                                 sin12 = sin(2 *pi * StimePred),
                                 cos12 = cos(2 * pi * StimePred),
                                 sin6 = sin(2 * 2 * pi * StimePred),
                                 cos6 = cos(2 * 2 * pi * StimePred))
names(predLincomb) = gsub("^lc", "pred", names(predLincomb))
```

```r
StimeIndex = seq(1, length(timePoints))
timeOriginIndex = which.min(abs(difftime(timePoints, timeOrigin)))
# 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(timeRw2, model = 'rw2',
#                                                     values = StimeIndex,
#                                                     prior='pc.prec',
#                                                     param = c(log(1.01)/26, 0.5)),
#               data = co2s, family='gamma', lincomb = c(derivLincomb, seasonLincomb, predLincomb),
#               control.family = list(hyper=list(prec=list(prior='pc.prec', param=c(2, 0.5)))),
#               control.inla = list(strategy='gaussian', int.strategy='eb'),
#               verbose=TRUE)

# Can't run model in reasonable period of time so need to save and load model
#save(co2res, file="co2res.Rdata")
load("co2res.Rdata")

# Derivative
derivPred = co2res$summary.lincomb.derived[grep("time", rownames(co2res$summary.lincomb.derived)),
                                           c("0.5quant", "0.025quant", "0.975quant")]
scaleTo10Years = (10 * 365.25/as.numeric(diff(timePoints, units = "days")))

# Figure 1
par(mfrow=c(3,2))

# OPEC oil embargo, Oct 1973
matplot(timePoints, exp(co2res$summary.random$timeRw2[, c("0.5quant", "0.025quant", "0.975quant")]),
        type = "l", col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n", xlab = "time",
        ylab = "ppm", xlim = ISOdate(c(1968, 1978), 1, 1, tz = "UTC"), ylim = c(0.895, 0.935))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
xax = ISOdate(c(1968,1970, 1972, 1974, 1976, 1978), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))

# Global Economic Recessions, 1980-1982
matplot(timePoints, exp(co2res$summary.random$timeRw2[, c("0.5quant", "0.025quant", "0.975quant")]),
        type = "l", col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n", xlab = "time",
        ylab = "ppm", xlim = ISOdate(c(1976, 1986), 1, 1, tz = "UTC"), ylim = c(0.925, 0.965))
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "darkorange")
xax = ISOdate(c(1976, 1978, 1980, 1982, 1984, 1986), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))

# Fall of Berlin Wall, Nov 1991
matplot(timePoints, exp(co2res$summary.random$timeRw2[, c("0.5quant", "0.025quant", "0.975quant")]),
        type = "l", col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n", xlab = "time",
        ylab = "ppm", xlim = ISOdate(c(1986, 1996), 1, 1, tz = "UTC"), ylim = c(0.96, 1.01))
abline(v = ISOdate(1991, 11, 1, tz = "UTC"), col = "yellow3")
xax = ISOdate(c(1986, 1988, 1990, 1992, 1994, 1996), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))
```

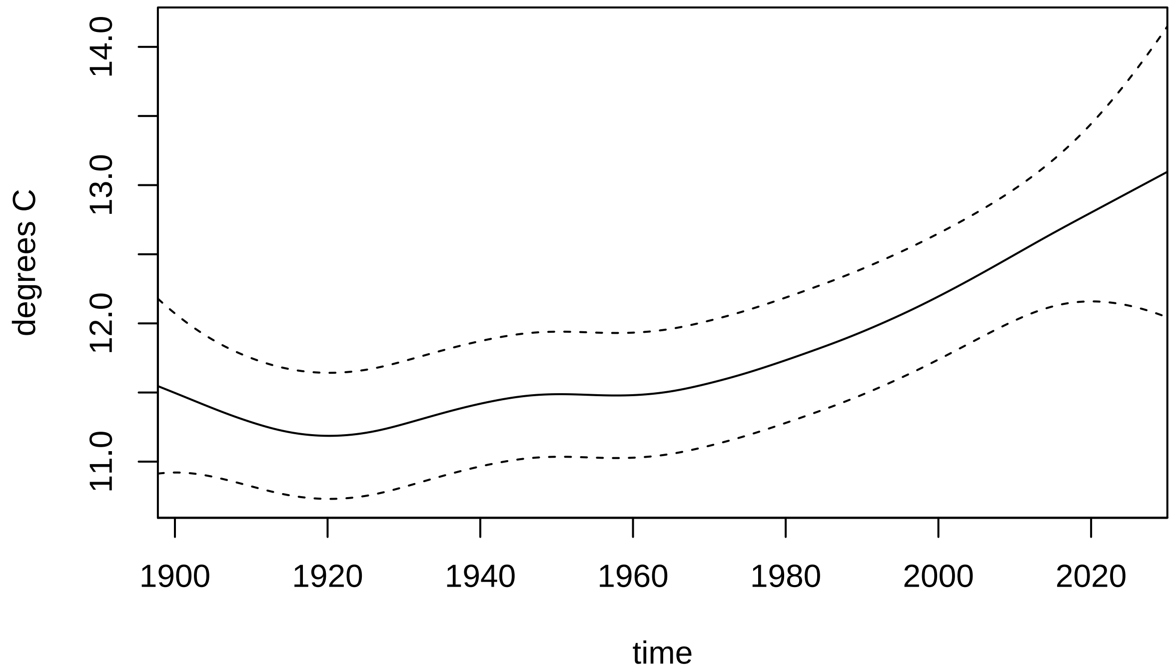
```r
# China joining WTO, 11 Dec 2001
matplot(timePoints, exp(co2res$summary.random$timeRw2[, c("0.5quant", "0.025quant", "0.975quant")]),
        type = "l", col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n", xlab = "time",
        ylab = "ppm", xlim = ISOdate(c(1996, 2006), 1, 1, tz = "UTC"), ylim = c(1, 1.06))
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "limegreen")
xax = ISOdate(c(1996, 1998, 2000, 2002, 2004, 2006), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))

# Global Recession, 15 Sep 2008
matplot(timePoints, exp(co2res$summary.random$timeRw2[, c("0.5quant", "0.025quant", "0.975quant")]),
        type = "l", col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n", xlab = "time",
        ylab = "ppm", xlim = ISOdate(c(2004, 2014), 1, 1, tz = "UTC"), ylim = c(1.04, 1.11))
abline(v = ISOdate(2008, 09, 15, tz = "UTC"), col = "deepskyblue")
xax = ISOdate(c(2004, 2006, 2008, 2010, 2012, 2014), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))

# Paris Agreement, 12 Dec 2015
matplot(timePoints, exp(co2res$summary.random$timeRw2[, c("0.5quant", "0.025quant", "0.975quant")]),
        type = "l", col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n", xlab = "time",
        ylab = "ppm", xlim = ISOdate(c(2010, 2020), 1, 1, tz = "UTC"), ylim = c(1.075, 1.155))
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "purple")
xax = ISOdate(c(2010, 2012, 2014, 2016, 2018, 2020), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))

# Figure 2
matplot(timePoints[-1], scaleTo10Years * derivPred,
        type = "l", col = "black", lty = c(1, 2, 2), ylim = c(0, 0.1),
        xlim = range(as.numeric(co2s$date)), xaxs = "i", xaxt = "n",
        xlab = "time", ylab = "log ppm, change per 10yr")
xax = ISOdate(c(1960, 1970, 1980, 1990, 2000, 2010), 1, 1, tz = "UTC")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "darkorange")
abline(v = ISOdate(1991, 11, 1, tz = "UTC"), col = "yellow3")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "limegreen")
abline(v = ISOdate(2008, 09, 15, tz = "UTC"), col = "deepskyblue")
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "purple")
```

Heat



degrees C
14.0
13.0
12.0
11.0
1900
1920
1940
1960
1980
2000
2020
time

Appendix B: Heat Code

```
heatUrl = "http://pbrown.ca/teaching/appliedstats/data/sableIsland.rds"
heatFile = tempfile(basename(heatUrl))
download.file(heatUrl, heatFile)
x = readRDS(heatFile)
x$month = as.numeric(format(x$Date, "%m"))
xSub = x[x$month %in% 5:10 & !is.na(x$Max.Temp...C.), ]
weekValues = seq(min(xSub$Date), ISOdate(2030, 1, 1, 0, 0, 0, tz = "UTC"), by = "7 days")
xSub$week = cut(xSub$Date, weekValues)
xSub$weekIid = xSub$week
xSub$day = as.numeric(difftime(xSub$Date, min(weekValues), units = "days"))
xSub$cos12 = cos(xSub$day * 2 * pi/365.25)
xSub$sin12 = sin(xSub$day * 2 * pi/365.25)
xSub$cos6 = cos(xSub$day * 2 * 2 * pi/365.25)
xSub$sin6 = sin(xSub$day * 2 * 2 * pi/365.25)
xSub$yearFac = factor(format(xSub$Date, "%Y"))

lmStart = lm(Max.Temp...C. ~ sin12 + cos12 + sin6 + cos6, data = xSub)
startingValues = c(lmStart$fitted.values,
                   rep(lmStart$coef[1], nlevels(xSub$week)),
                   rep(0, nlevels(xSub$weekIid) + nlevels(xSub$yearFac)),
                   lmStart$coef[-1])

library(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())

#sableRes = inla(Max.Temp...C. ~ 0 + sin12 + cos12 + sin6 + cos6 +
#                  f(week, model='rw2', constr=FALSE, prior='pc.prec', param = c(0.1/(52*100),0.05)) +
#                  f(weekIid, model='iid', prior='pc.prec', param = c(1, 0.5)) +
#                  f(yearFac, model='iid', prior='pc.prec', param = c(1, 0.5)), family='T',
#                control.family = list(hyper = list(prec = list(prior='pc.prec', param=c(1, 0.5)),
#                                                   dof = list(prior='pc.dof', param=c(10, 0.5)))),
#                control.mode = list(theta = c(-1,2,20,0,1), x = startingValues, restart=TRUE),
#                control.compute=list(config = TRUE),
#                control.inla = list(strategy='gaussian', int.strategy='eb'),
#                data = xSub, verbose=TRUE)

#save(sableRes, file = "sableRes.Rdata")
load("sableRes.Rdata")

mySample = inla.posterior.sample(n = 24, result = sableRes, num.threads = 8,
                                 selection = list(week = seq(1, nrow(sableRes$summary.random$week))))
weekSample = do.call(cbind, lapply(mySample, function(xx) xx$latent))


# Fit
matplot(weekValues[-1], sableRes$summary.random$week[, paste0(c(0.5, 0.025, 0.975), "quant")],
        type = "l", lty = c(1, 2, 2), xlab = "time", ylab = "degrees C", xaxt = "n",
        col = "black", xaxs = "i")
```

```
forXaxis2 = ISOdate(seq(1880, 2040, by = 20), 1, 1, tz = "UTC")
axis(1, forXaxis2, format(forXaxis2, "%Y"))
```