STA442 Assignment 3
Zhuojing Qiu
13/11/2019

Question1. CO2

Introduction

We analyzed the Carbon Dioxide concentrations from an observatory in Haiwaii, which can be made available by the Scripps CO2 Program at scrippsco2.ucsd.edu. Our main focus was to investigate whether the CO2 data appears to be impacted by some historical events. Specifically, we would like to find out the trend and growth rate of CO2 concentration since 1970.

Methods

We first fitted a simple Generalized Additive Model with covariates of *sine, cosine*, and *days.* Since the effect of time on the increase of CO2 can be linear or nonlinear, we used GAM model instead of LM model. As Figure 1 shown below, there is definitely an increasing trend, but there is also a strong seasonal effect.

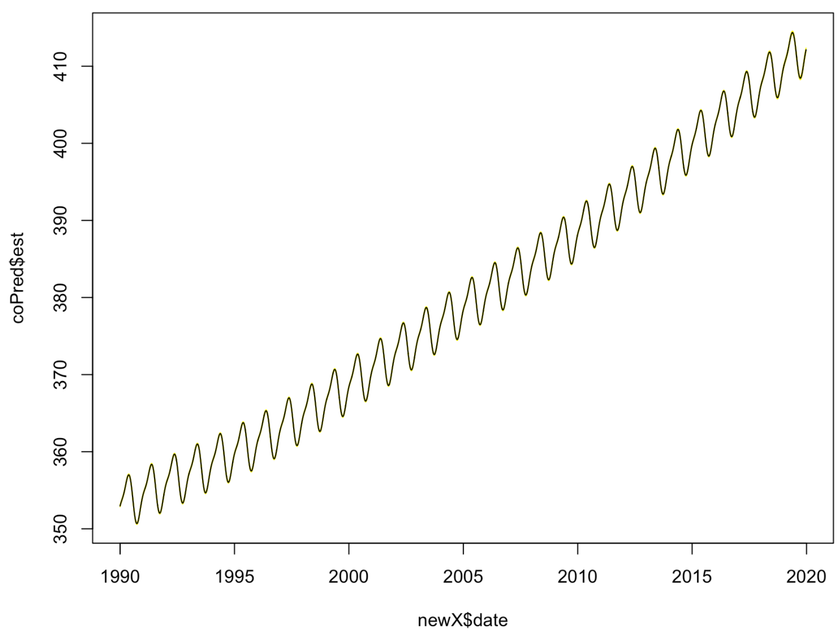

Figure 1. Predicted Carbon levels from 1990 to 2019

To visualize the trend more precisely, we fitted a Bayesian Generalized Additive Model from the Gamma family with a log link function to predict carbon concentration from days since 1970. Then we differentiated the smooths in a model with 95% CIs for derivatives. In this model, we set a PC prior to give 5% chance that the random slope standard deviation is greater than log(1.01)/26, which means that the slope changes by 1% from biweekly data over a year. Another prior is set to give 5% chance that the random intercept standard deviation of time is greater than 2. Below is our fitted model:

$$y_i \sim \Gamma(\theta)$$
$$log(E(Y)) = \sum_{j=1}^{4} \phi_j(x_i)\beta_j + U(t_i) + V_i$$

$$\phi_1(x_i) = cos(2\pi x_i),\ \phi_2(x_i) = sin(2\pi x_i),\ \phi_3(x_i) = cos(4\pi x_i),\ \phi_4(x_i) = sin(4\pi x_i)$$

where Y is concentration of carbon measured at year x_i since 1970

ϕ_1 and ϕ_2 represent yearly fluctuations and ϕ_3 and ϕ_4 represent biyearly fluctuations.

$U(t_i)$ is a second order random walk

V_i covers independent variation or over-dispersion.

Results

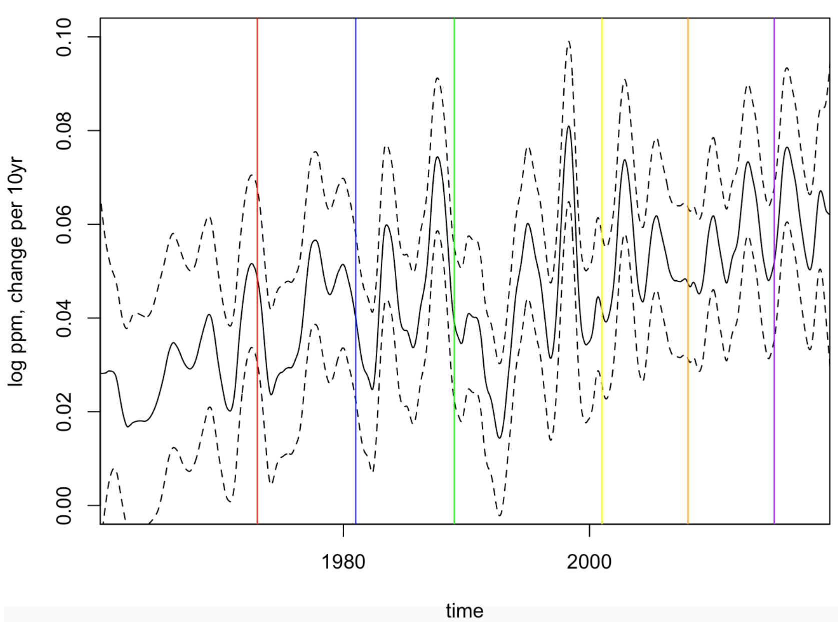

Figure 2. An approximation to the derivative of the GAM Model

As shown in Figure 2, we obtained the first order derivative of CO2 trend with a 95% confidence intervals for every estimate. The confidence intervals were illustrated in dotted lines. Additionally, we labelled 1973, 1981, 1989, 2001, 2008 and 2015 in red, blue, green, yellow, orange and purple.

We noticed that the derivative is at its smallest at the beginning of 1990, which coincides with the fall of the Berlin wall and the preceding dramatic fall in industrial production. This indicates the CO2 concentration increased at a slower rate than ever, so there was a big effect of the event.

At the end of 1973, the OPEC decided to stop exporting oil to the U.S, we can see that although the CO2 concentration is still going up, the occurrence of this event dropped the growth rate of CO2 from CI:0.03-0.07 to CI:0.03-0.06, however, it soon picked up later in 1974. Similarly, around 1980 to 1982, the global economic recessions made CO2 growth rate drop constantly in that period but soon picked up at a even faster rate. After China joining the WTO at the end of 2001, the CO2 concentration grew rapidly towards 0.07, which shows that the growth in industrial production correspond to a positive growth in CO2. In 2008, the bankruptcy of Lehman Brothers did not affect the CO2 growth rate too much, as it remains around 0.05 for 1 to 2 years.

Lastly, after the Paris Agreement signed at the end of 2015, we see a decreasing trend of the growth rate of CO2, so this event definitely helped to decrease CO2 emissions.

Summary

We have analyzed the trend and growth rate of CO2 concentration since 1970. From the estimated smooth trend of CO2, we saw that there is definitely an increasing trend over the decades. Though slight reduction in the growth rate of CO2 happens regularly, it always soon recovers to grow. From the growth rate of CO2 concentration, we found that the economy development and industrial development can both have a huge impact on CO2 emissions. So while making development of an economy or an industry is important, we should always remember to consider the consequences that can be brought to our environment.

Question2. Heat

Introduction

We analyzed the temperature data recorded on Sable Island, off the coast of Nova Scotia, using an R version of the dataset available at pbrown.ca. Our main focus was to investigate whether human activities have caused a global warming over time, Specifically, we would like to find out the estimated time trend and temperature and determine by how much the temperature has changed since pre-industrial period.

Method

We fitted an INLA model using T distribution as our response variable, since T distribution works well with heavy-tails data set. Our model had a random effect of a random walk defined every week, which was given a prior such that there is a 5% chance for the random slope standard deviation to be greater than 0.1/(52*100). This prior means that the slope changes by 10% from weekly over 100 years. Since every week has an independent effect, we also had a random effect on week. Additionally, for large scale in climatic change, we tend to have warmer summers and colder summers, so we put years as a random effect as well. Both of week effect and year effect were given a prior such that there is a 5% chance that the random intercept standard deviation is greater than 1 degree.

Besides, we have two parameters in such a T-distribution model. One is precision parameter and the other is degree of freedom parameter. For precision, the standard deviation of an individual daily observation has a 50% probability that it is above 1 degree. For degrees of freedom, the prior medium is 10 degrees of freedom for a T distribution. Below is our fitted model:

$$y_i \sim \Gamma(\theta)$$

$$E(Y) = \sum_{j=1}^{4} \phi_j(x_i)\beta_j + U(t_i) + W_i + T_i + V_i$$

$$\phi_1(x_i) = cos(2\pi x_i),\ \phi_2(x_i) = sin(2\pi x_i),\ \phi_3(x_i) = cos(4\pi x_i),\ \phi_4(x_i) = sin(4\pi x_i)$$

where Y is the temperature measured on Sable Island at week x_i since 2016 May

ϕ_1 and ϕ_2 represent yearly fluctuations and ϕ_3 and ϕ_4 represent biyearly fluctuations.

W_i is the random intercepts weekIid

T_i is the random intercepts yearFac

$U(t_i)$ is a second order random walk

V_i covers independent variation or over-dispersion.

Results

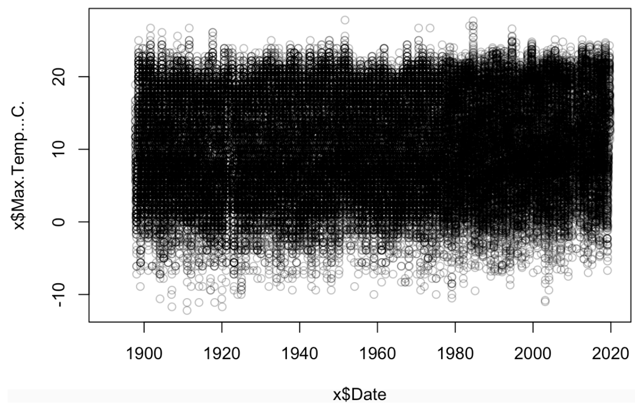

Figure1. Daily maximum temperature from 1897 October to present

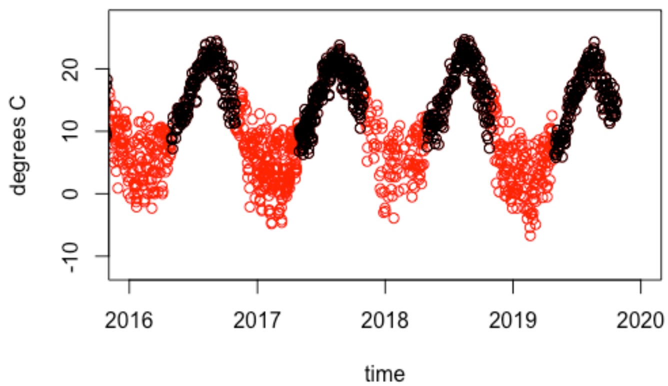

Figure2. Temperature with summer months from 2016 to present

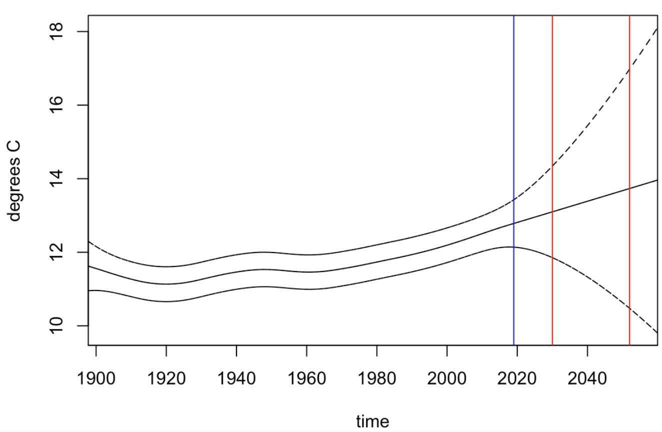

Figure3. Estimated time trend and temperature

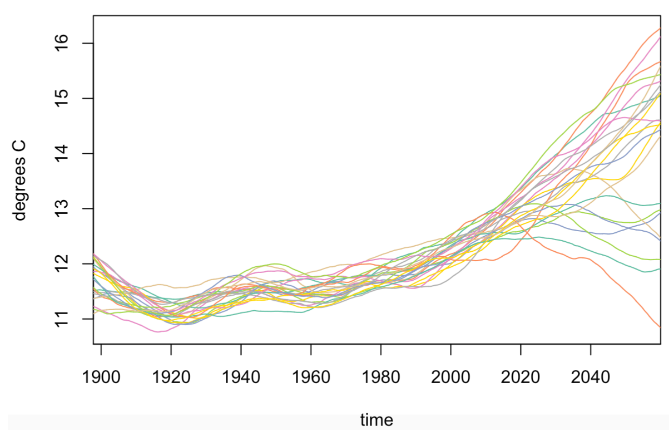

Figure 4. Posterior samples of the estimated time trend and temperature

Based on our fitted model, we obtained 4 plots. Figure 1 gives us the daily maximum temperature from 1897 October to present. However, there does not appear to be any trend at all. Figure 2 shows a period from 2016 to the present, with summer months in black and winter in red. There is clearly a seasonal effect shown in the plot. However, we cannot see the overall trend of temperature.

Figure 3 shows the random walk. The intercept centres at approximately 12 instead of 0, since our random walk was not constrained while fitting the model. We can read off the overall time trend from this plot, where temperature rises consistently with the degrees. We labelled the year 2019 as present in blue line and the year 2030 ad 2052 in red lines. Even though it is clearly that there is an increasing smooth trend, the lower bound of the confidence interval starts decreasing since 2020. As indicated by the blue line, the temperature in current year is approximately 12.5 degrees, with a confidence interval between 12 and 13. The temperature during pre-industrial time is approximately 11.5 degrees, with a confidence interval between 11 and 12. So temperature on Sable Island has been increased by approximately 1 degree at present compared to the pre-industrial period. Also, as indicated by the two red lines, we can approximate the posterior medium from 2030 to 2052 to be 13 to 14 degrees. So global warming is likely to reach 1.5 degrees between 2030 and 2052.

Figure 4 illustrates 20 different posterior samples, which gives us a similar trend with Figure 3, but a rougher trend than the estimation in Figure 3.

Conclusion

We have analyzed the estimated time trend and temperature by fitting an INLA model. From the estimated time trend, we concluded that the temperature has been increased by approximately 1 degree compared to pre-industrial period. Moreover, global warming is likely to reach 1.5 degree between 2030 and 2052. So our analysis is broadly supportive to IPCC's statement.

Appendix

CO2

```
library(tidyverse)
library(readr)
library('mgcv')
co2s <- read_csv("Desktop/untitled folder/co2.csv")
#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
#plot(co2s$date, co2s$co2, log = "y", cex = 0.3, col = "#00000040", xlab = "time", ylab =
"ppm")
#plot(co2s[co2s$date > ISOdate(2015, 3, 1, tz = "UTC"), c("date", "co2")], log = "y", type = "o",
xlab = "time", ylab = "ppm", cex = 0.5)
timeOrigin = ISOdate(1980, 1, 1, 0, 0, 0, tz = "UTC")
co2s$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$days/365.25)
cGam = gam(co2 ~ s(days) + cos12 + sin12 + cos6 + sin6, data = co2s)
newX = data.frame(date = seq(ISOdate(1990, 1, 1, 0, 0, 0, tz = "UTC"), by = "1 days", length.out
= 365 * 30))
newX$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$days/365.25)
coPred = predict(cGam, newX, se.fit = TRUE)
coPred = data.frame(est = coPred$fit, lower = coPred$fit - 2 * coPred$se.fit, upper = coPred$fit
+ 2 * coPred$se.fit)
plot(newX$date, coPred$est, type = "l")
matlines(as.numeric(newX$date), coPred[, c("lower","upper", "est")], lty = 1, col = c("yellow",
"yellow","black"))
#newX = newX[1:365, ]
#newX$days = 0
```

```
#plot(newX$date, predict(cGam, newX)) # this is a forecast
library("INLA")
# time random effect
timeBreaks = seq(min(co2s$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))
StimeIndex = seq(1, length(timePoints))
timeOriginIndex = which.min(abs(difftime(timePoints, timeOrigin)))
# disable some error checking in INLA
library("INLA")
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())
co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 + f(timeRw2, model = 'rw2',
                                values = StimeIndex,
                                 prior='pc.prec', param = c(log(1.01)/26, 0.5)),
        data = co2s, family='gamma', lincomb = c(derivLincomb, seasonLincomb,
predLincomb),
        control.family = list(hyper=list(prec=list(prior='pc.prec', param=c(2, 0.5)))),
        # add this line if your computer has trouble
        #  control.inla = list(strategy='gaussian', int.strategy='eb'),
        verbose=TRUE)
matplot(timePoints, exp(co2res$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")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1973, 1, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(1981, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(1989, 1, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(2001, 1, 1, tz = "UTC"), col = "yellow")
abline(v = ISOdate(2008, 1, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(2015, 1, 1, tz = "UTC"), col = "purple")
```

Heat

```
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$Date), ISOdate(2060, 1, 1, 0, 0, 0, tz = "UTC"), by = "7 days")
xSub$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"))

matplot(xSub[1:1000, c('sin12','cos12')], type='l')

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])

INLA::inla.doc('^t$')
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::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)

sableRes$summary.hyper[, c(4, 3, 5)]
sableRes$summary.fixed[, c(4, 3, 5)]
Pmisc::priorPost(sableRes)$summary[, c(1, 3, 5)]

mySample = inla.posterior.sample(n = 24, result = sableRes, num.threads = 8, selection =
list(week = seq(1,nrow(sableRes$summary.random$week))))
length(mySample)
names(mySample[[1]])
weekSample = do.call(cbind, lapply(mySample, function(xx) xx$latent))
dim(weekSample)
head(weekSample)
plot(x$Date, x$Max.Temp...C., col = mapmisc::col2html("black", 0.3))
forAxis = ISOdate(2016:2040, 1, 1, tz = "UTC")
plot(x$Date, x$Max.Temp...C., xlim = range(forAxis),
     xlab = "time", ylab = "degrees C", col = "red",
     xaxt = "n")
points(xSub$Date, xSub$Max.Temp...C.)
axis(1, forAxis, format(forAxis, "%Y"))
matplot(weekValues[-1], sableRes$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, 2060, by = 20), 1, 1, tz = "UTC")
```

```
axis(1, forXaxis2, format(forXaxis2, "%Y"))
myCol = mapmisc::colourScale(NA, breaks = 1:8, style = "unique",
                        col = "Set2", opacity = 0.3)$col
matplot(weekValues[-1], weekSample, type = "l", lty = 1,col = myCol, xlab = "time", ylab =
"degrees C", xaxt = "n", xaxs = "i")
axis(1, forXaxis2, format(forXaxis2, "%Y"))
```