The Analysis of CO2 Concentration

Xinqi Shen

1 Introduction

The dataset is from the Scripps program, and this dataset provides us with information about CO2 concentration from the Mauna Loa Observatory in Hawaii in past decades. In this report, we will discuss if the CO2 data appears to be impacted by some historical events.

2 Method

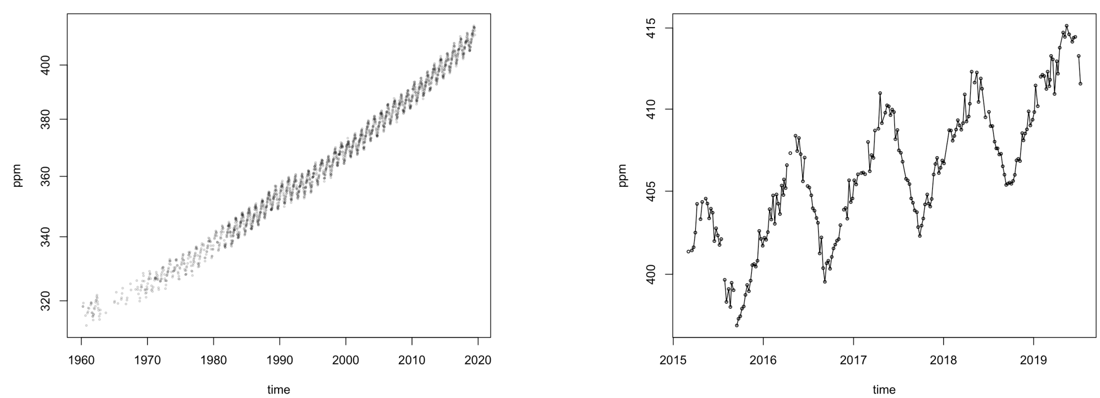

Figure 1: Plot for CO2 concentration

We fit a gamma generalized additive model to analyze the CO2 concentration. Since the concentration of CO2 are continuous and positive, so we assume the response variable follows a Gamma distribution. From Figure 1, the concentration of CO2 appears a seasonal effect with time, so we add annual and semi-annual seasonality predictors and treat them as fixed effects. Also, the concentration of CO2 shows an upward trend with time, then we use a second-order random walk in our model. Finally, in order to cover independent variation or over-dispersion, the model should also contain an iid component for time. Noted that, we use bi-weekly(14 days) as time break since it's more stable. Given the following model:

$$Y_i \sim Gamma(\theta, \frac{\lambda_i}{\theta})$$
$$log(\lambda_i) = \beta_0 + \beta_1 sin(4\pi x_i) + \beta_2 cos(4\pi x_i) + \beta_3 sin(2\pi x_i) + \beta_4 cos(2\pi x_i) + U(t_i) + V_i$$
$$[U_1 ... U_T]^T \sim RW2(\sigma_U^2)$$
$$V_i \sim N(0, \sigma_V^2)$$

Where Y_i is the CO2 concentration. $sin(4\pi x_i)$ and $cos(4\pi x_i)$ are annual seasonality predictors. $sin(2\pi x_i)$ and $cos(2\pi x_i)$ are semi-annual seasonality predictors. $U(t_i)$ is the time random effect representing the second-order random walk (timeRW2). V_i is an iid component (timeIid) for time covers independent variation or over-dispersion.

We use penalized complexity prior for the gamma shape θ, σ_U and σ_V. With $P(\frac{1}{\sqrt{\theta}} > 0.1) = 0.5$, $P(\sigma_U > 0.0004/26) = 0.5$ and $P(\sigma_V > log(1.001)/26) = 0.5$.

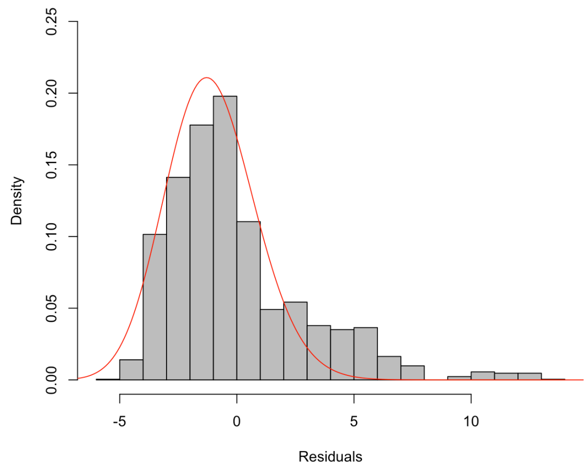

Figure 2: The histogram of residuals

Based on Figure 2, the Gamma distribution is a good fit in our model.

3 Result

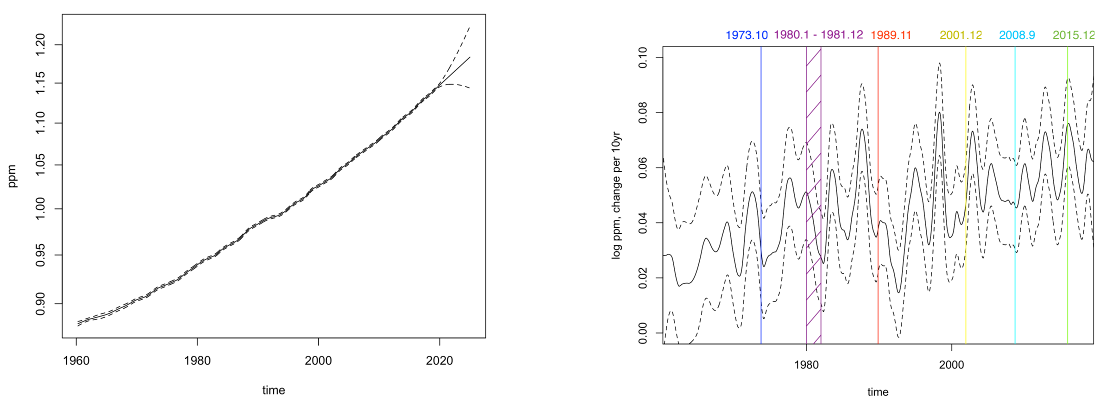

Figure 3: (a) random effect; (b) derivative

The Figure 3(a) provides the estimated smoothed trend of CO2 concentration. However, we cannot clearly see the change of CO2 concentration during specific events. Instead, we can analyze the impact of those events to CO2 concentration based on the first derivative of estimated smoothed trend shown in Figure 3(b). Both plots demonstrate an overall increasing of CO2 concentration over time, we will mainly focus on the change of increasing rate of CO2 concentration impacted by the following events.

- **OPEC oil embargo which began in October 1973**: After this event, we find that the increasing rate of CO2 concentration maintained the previous decreasing trend in the following months. However, after it reached a local minimum, the increasing rate restarted growth.
- **Global economic recessions around 1980-1982**: During this period, the increasing rate showed an obvious decreasing trend started from a local maximum to a local minimum.

- **Fall of the Berlin wall on November 1989**: After this event, the increasing rate of CO2 concentration slightly increased first, then it stayed decreasing until reaching the lowest point since 1960.
- **China joining the WTO on 11 December 2001**: After this event, the increasing rate of CO2 concentration rapidly increased in the following time, which may indicate the development of industrialization in China.
- **Bankruptcy of Lehman Brothers on 15 September 2008**: After this event, the increasing rate slightly declined in a very short time followed by rapid growth in the following year.
- **Signing of the Paris Agreement on 12 December 2015**: After this event, the increasing rate immediately appeared to fall, so the Paris Agreement has achieved results.

4 Conclusion

Based on the analysis from this dataset, we found that the CO2 concentration maintained an increasing trend in the past few decades. However, not surprisingly, some historical events have a short-term or long-term effect on the concentration of CO2. More specific, some events caused the increasing rate of CO2 concentration immediately decreased, such as Global economic recessions, and Signing of the Paris Agreement. Conversely, some events led to rapid growth in the increasing rate of CO2 concentration, such as China joining the WTO. Besides, in terms of Bankruptcy of Lehman Brothers and OPEC oil embargo, both events caused the increasing rate of CO2 declined in a short time followed by the rapid growth. Finally, Fall of the Berlin wall led to the increasing rate of CO2 concentration reached the lowest point since 1960 after a brief rise.

5 Appendix

```
cUrl = paste0("http://scrippsco2.ucsd.edu/assets/data/atmospheric/",
              "stations/flask_co2/daily/daily_flask_co2_mlo.csv")
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$time), format = "%Y-%m-%d %H:%M",
                     tz = "UTC")
co2s[co2s$quality >= 1, "co2"] = NA
plot(co2s$date, co2s$co2, log = "y", cex = 0.3, col = "#00000040",
     xlab = "time", ylab = "ppm")
plot(co2s[co2s$date > ISOdate(2015, 3, 1, tz = "UTC"),
          c("date", "co2")], log = "y", type = "o", xlab = "time",
     ylab = "ppm", cex = 0.5)
timeOrigin = ISOdate(1970, 1, 1, 0, 0, 0, tz = "UTC")
co2s$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$days/365.25)
cLm = lm(co2 ~ days + cos12 + sin12 + cos6 + sin6,
         data = co2s)
summary(cLm)$coef[, 1:2]
hist(cLm$residuals, prob=TRUE,col="grey",
     border="black",main="",xlab = "Residuals",ylim=c(0,0.25))
xSeq = seq(-20,20, by=0.1)
lines(xSeq, 10*dgamma(1:401, shape=100, scale=1.9), col = 'red')
library("INLA")
```

```r
# 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())
co2s$timeIid = co2s$timeRw2
co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 +
        f(timeRw2, model = 'rw2',
        values = StimeIndex,
        prior='pc.prec', param = c(0.0004/26, 0.5)) +
        f(timeIid, model = 'iid',
        prior = 'pc.prec',param = c(log(1.001)/26, 0.5)),
        data = co2s, family='gamma',
        lincomb = c(derivLincomb, seasonLincomb, predLincomb),
        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', 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"))
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")))
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")
axis(1, xax, format(xax, "%Y"))
rangeY <- range(scaleTo10Years * derivPred)
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "blue")
rect(xleft=ISOdate(1980, 1, 1, tz = "UTC"),
    xright = ISOdate(1981, 12, 31, tz = "UTC"),
    ybottom=rangeY[1],ytop=rangeY[2], density=10, col = "#8B008B")
abline(v = ISOdate(1989, 11, 9, tz = "UTC"), col = "red")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "yellow")
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "#00FFFF")
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "#7FFF00")
```

The Analysis of Global Warming

Xinqi Shen

1 Introduction

Nowadays, Climate warming has become an issue of increasing concern. The Intergovernmental Panel on Climate Change(IPCC) states that "Human activities are estimated to have caused approximately 1.0℃ of global warming above pre-industrial levels, with a likely range of 0.8℃ to 1.2℃. Global warming is likely to reach 1.5℃ between 2030 and 2052 if it continues to increase at the current rate." In this report, we will analyze the temperature data recorded on Sable Island, then discussing whether the IPCC statement is reliable.

2 Method

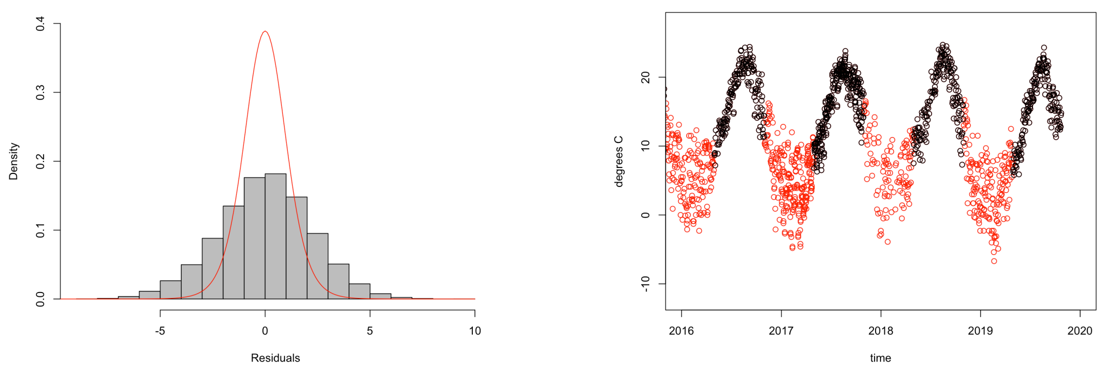

Figure 1: (a) the histogram of residuals; (b) the period of daily maximum temperature from 2016 to present

We fit a student-t generalized additive model to analyze the temperature. From Figure 1(a), the student-t distribution is approximately a good fit in our model. Also, from Figure 1(b) , it appears a seasonal effect with time, so we add annual and semi-annual seasonality predictors and treat them as fixed effects. In addition, we consider there is an increasing trend for temperature with time, so we use a second-order random walk in our model. Finally, in order to cover independent variation or over-dispersion, we also add two iid components for time(week and year) and treat them as random effects. Given the following model:

$$\sqrt{s\tau}(Y_{ij} - \eta_{ij}) \sim T_\nu$$
$$\eta_{ij} = \beta_0 + \beta_1 sin(4\pi x_i) + \beta_2 cos(4\pi x_i) + \beta_3 sin(2\pi x_i) + \beta_4 cos(2\pi x_i) + U(t_{ij}) + V_{ij} + W_i$$
$$[U_1 ... U_T]^T \sim RW2(\sigma_U^2)$$
$$V_{ij} \sim N(0, \sigma_V^2)$$
$$W_i \sim N(0, \sigma_W^2)$$

Where Y_{ij} is the maximum temperature in year i week j, η_{ij} is the corresponding linear predictor, τ is the precision parameter, s is a fixed scaling variable($s > 0$) and ν is the degree of freedom. $sin(4\pi x_i)$ and $cos(4\pi x_i)$ are annual seasonality predictors. $sin(2\pi x_i)$ and $cos(2\pi x_i)$ are semi-annual seasonality predictors. $U(t_{ij})$ is the second-order random walk. V_{ij} is an iid component for week random effect. W_i is an iid component for year random effect.

We use penalized complexity prior for τ, ν, σ_U, σ_V, and σ_W. With $P(\frac{1}{\sqrt{\tau}} > 1) = 0.5$, $P(\nu < 10) = 0.5$, $P(\sigma_U > 0.1/(52 * 100)) = 0.05$, $P(\sigma_V > 1) = 0.5$, and $P(\sigma_W > 1) = 0.5$.

3 Result

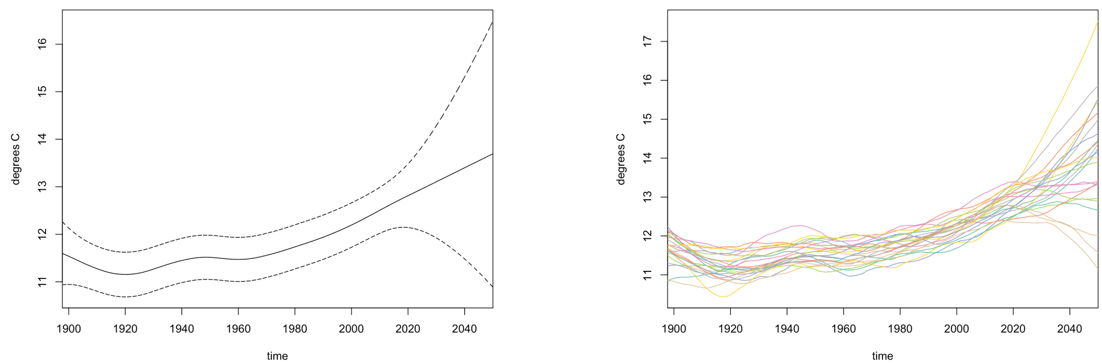

Figure 2: (a) estimated time trend; (b) posterior samples of this trend

Based on Figure 2(a), the estimated time trend shows that the temperature approximately increases 1.0℃(with a likely range of 0.8℃ to 1.2℃) from pre-industrial level to the present(2019), and it is likely to reach 1.5℃ until 2040 if it continues to increase at the current rate. Not surprisingly, the Figure 2(b) provides the similar results. The posterior samples visually are all within 95%CI shown in Figure 2(a). Noted that, even though it is possible for the model generating extreme samples against IPCC statement due to large CI in the future(after 2019), this possibility is quite low. Thus, we still have high confidence that the result does support the statement from the IPCC in general.

4 Conclusion

Based on the analysis from the data recorded on Sable Island, we found that global temperature keeps increasing since 1880 due to human activities. More specific, the global temperature has already increased around 1℃(with a likely range of 0.8℃ to 1.2℃) above pre-industrial level, and if it continues at the current rate, it is likely to reach 1.5℃(high confidence) between 2030 and 2052. Thus, the result is consistent with the statement from IPCC, and we can conclude that its statement tends to be reliable.

5 Appendix

```
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$Date), ISOdate(2050, 1, 1,
                                           0, 0, 0, tz = "UTC"), by = "7 days")
xSub$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])
hist(lmStart$residuals, prob=TRUE, col="grey",border="black",xlab = 'Residuals',
main='',ylim=c(0,0.4))
xSeq = seq(-10, 10, by= 0.1)
lines(xSeq, dt(xSeq, df=10), col = 'red')
library("INLA")
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$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)
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$Max.Temp...C., col = mapmisc::col2html("black",
                                                     0.3))
forAxis = ISOdate(2016:2020, 1, 1, tz = "UTC")
plot(x$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, 2040, 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"))
```