STA442 Assignment 3: Smoothing

Hongbo Du
1003568089
hongbo.du@mail.utoronto.ca

Contents

Question 1: CO_2

Introduction

This report contains the graphical analysis of the atmospheric Carbon Dioxide concentrations from an observatory in Hawaii. The goal is to determine whether the CO_2 data appears to be impacted by some historical events.

Method

Using Integrated Nested Laplace Approximation (INLA) to fit a RW(2) model is an efficient way to analysis the data. Mathematically, the model for the data follows:

$$Y_i \sim \text{Gamma}(O_i\lambda_i)$$

$$\log(\lambda_i) = \sin(2\pi X_i)\beta_1 + \cos(2\pi X_i)\beta_2 + \sin(2 * 2\pi X_i)\beta_3 + \cos(2 * 2\pi X_i)\beta_4 + U(t_i) + V_i$$
$$[U_1 \ldots U_T]^\top \sim \text{RW2}(0, \sigma_U^2)$$
$$V_i \sim \mathcal{N}(0, \sigma_V^2)$$

According to the goal, we obtain the derivative plot as Figure 1.1.

Conclusion

First of all, the OPEC oil embargo which began in October 1973 can be thought as a turning point of the CO_2 concentration, because the change of the CO_2 concentration was going down before the policy was implemented. In Figure 1.1, the vertical red line on the left indicates the time when the change turns upwards.
Secondly, the global economic recessions around 1980-1982 can be thought as the time interval of the change of CO_2 concentration dropped, because the change of the CO_2 concentration was going down during these years. This probably is due to less industrial production during economic recession. In Figure 1.1, the downward slope between two blue vertical line indicates the time when the the economic recessions happened which resulted in a decreased change in CO_2 concentration. After the economic recession, the CO_2 concentration increased.
Thirdly, according to green vertical line in the Figure 1.1, it is obvious that, the fall of the Berlin wall almost exactly 30 years ago, preceding a dramatic fall in industrial production in the Soviet Union and Eastern Europe. This event directly led a fall in the change in CO_2 concentration, since the slope of change in CO_2 concentration after the green line appeals a dramatic fall.
At the beginning of the 21$^{\text{th}}$ century, China joining the WTO on 11 December 2001, which was followed by rapid growth in industrial production. This can be seen from the purple line in the Figure 1.1. Unlike the green line, the purple line indicates a rapid growth in the change of the CO_2 concentration, which shows the rapid growth in industrial production.
Next, the bankruptcy of Lehman Brothers on 15 September 2008, regarded as the symbolic start of the most recent global financial crisis, does not appeal to have impact on the CO_2 data. The yellow line in the Figure 1.1 does not mark the fall of CO_2 concentration. Instead,

it marks a turning point of the growth CO_2 concentration. Maybe it is because that the financial crisis does not have strong negative impact in industrial production.
Lastly, the signing of the Paris Agreement on 12 December 2015, intended to limit CO_2 emissions can be thought as a remarkable contribution on limiting CO_2 emission. The reason is that the orange in Figure 1.1 indicates a beginning of downward slope of the change of the CO_2 concentration.

Figure

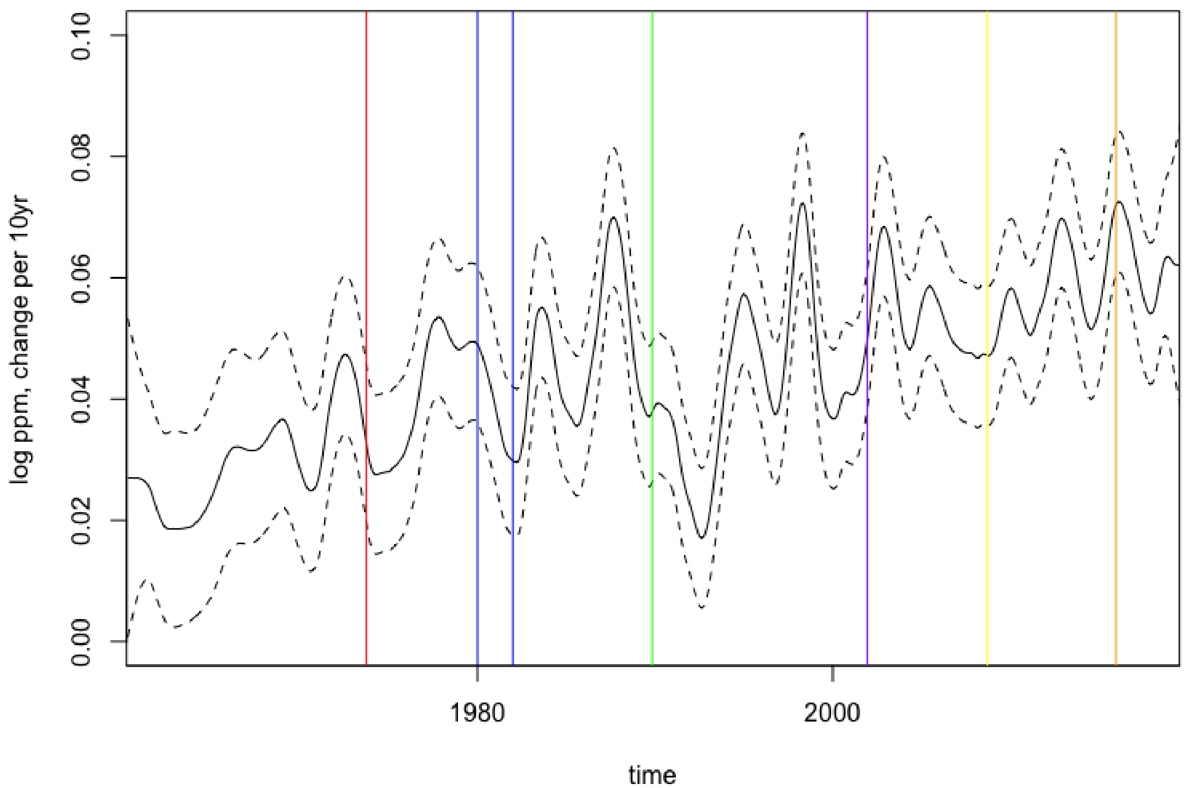


Figure 1.1: the derivative of the trend of the CO2 concentration.

Question 2: Heat

To: Maxim Burningier
From: Hongbo Du

Dear Maxim Burningier,

I am writing to reply you the letter that you wrote me earlier in regards to write a consulting report using the Sable Island data ti refute the IPCC's statements about global temperatures rises. Kindly read the following statistical report.

Introduction

This report contains the statistical results of the daily maximum temperature data recorded on Sable Island. The goal is to determine if the IPCC's statement is true according to statistical analysis. Firstly, IPCC states that "Human activities are estimated to have caused approximately 1.0°C of global warming above pre- industrial levels, with a likely range of 0.8°C to 1.2°C". Secondly, IPCC states that "Global warming is likely to reach 1.5°C between 2030 and 2052 if it continues to increase at the current rate. (high confidence)".

Method

In order to determine the IPCC's statements, fitting a INLA model is sufficient to predicate the data and make forecasting about the temperature. In this report, the method is using INLA model to fit a RW(2) since we want the random slope to be a straight line. The reason of choosing INLA model are: 1) it is fast enough using INLA, 2) explicit specification of assumptions, 3) and rigorous inference methodology. Mathematically, the model is as follows:

$$\sqrt{s\tau}(Y_i - \eta) \sim T_\nu$$

where for the response Y_i, τ is the precision parameter, $s > 0$ is a fixed scaling, η is the linear predictor, and T_ν is a reparameterized standard Student-t with $\nu > 2$ degrees of freedom with unit variance for all values of ν.

$$\nu = \sin(2\pi X_i)\beta_1 + \cos(2\pi X_i)\beta_2 + \sin(2 * 2\pi X_i)\beta_3 + \cos(2 * 2\pi X_i)\beta_4 + U(t_i) + V_i$$

$$[U_1 \ldots U_T]^\top \sim \text{RW2}(0, \sigma_U^2)$$

$$V_i \sim \mathcal{N}(0, \sigma_V^2)$$

Model assumptions:

- $U(t)$ is a second-order random walk and the second derivatives are $\mathcal{N}(0, \sigma_U^2)$
- V_i covers independent variation or over-dispersion

The prior is defined on $\theta = (\theta_1, \theta_2)$ where $\theta_1 = \log(\tau)$ and $\theta_2 = \log(\nu - 2)$ are hyperparameters. In this model, the prior is (0.1/(52*100), 0.05), that is, $\mathrm{P}(\nu < 0.1/(52 * 100)) = 0.05$. There is 0.05 chance of the slope change smaller than 0.1 per week over 100 years.

Results

According to Figure 2.1, the horizontal blue line marks the estimated initial temperature which is defined as the pre-industrial baseline in this report (11.56872°C). The vertical blue line indicates the time before industrial level and the vertical green line indicates the time of November in 2019. The horizontal green solid line marks the estimated temperature if the temperature has raised by 1°C with a range of 0.8°C to 1.2°C(the two horizontal green dashed lines). The intersection of the horizontal and vertical green lines is the estimated increased temperature by now. Meanwhile, the model estimated temperature in November 2019 is above and is very close to in the interval of two green dashed lines (in the interval of 12.56872±0.2°C).
According to Figure 2.1 again, the two red vertical line indicates the time interval between 2030 and 2052, and the red horizontal line represents the estimated temperature if the temperature raises by 1.5°C. Notice that the model estimated temperature by January 1^{st}, 2030, is above the horizontal red line which marks the 1.5°C increasing, and the temperature is estimated to be positively increasing between 2030 and 2052.

Conclusion

By interpreting the results, there is strong evidence that human activities are estimated to have caused approximately 1.0°C of global warming above pre-industrial levels, with a likely range of 0.8°C to 1.2°C. In addition, the forecasting shows that there is high confidence that global warming is likely to reach 1.5°C between 2030 and 2052 if it continues to increase at the current rate.

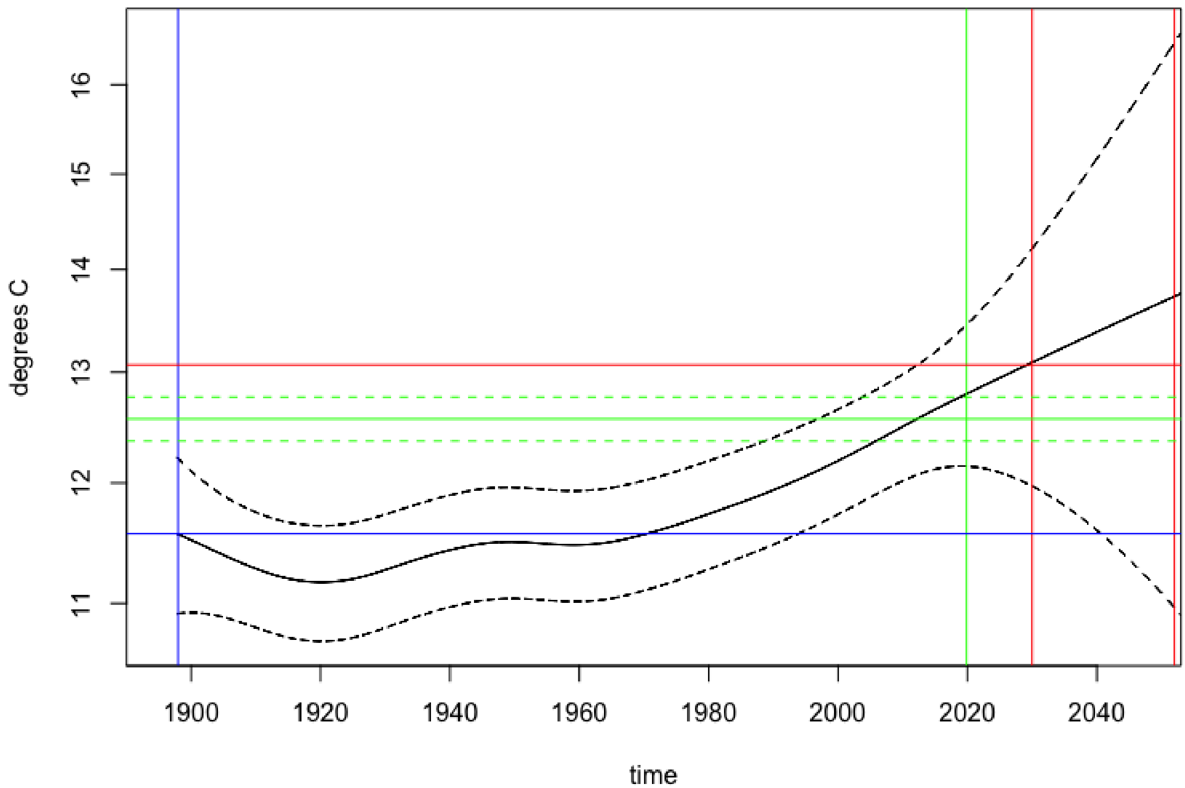


Figure 2.1: the estimated degree of the trend.

At the end, my statistical of the Sable Island data results fail to refute the IPCC's statements and it indicates that global warming is happening.
Save our planet.

Hongbo Du

Appendix

```
library(tidyverse)
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",
```

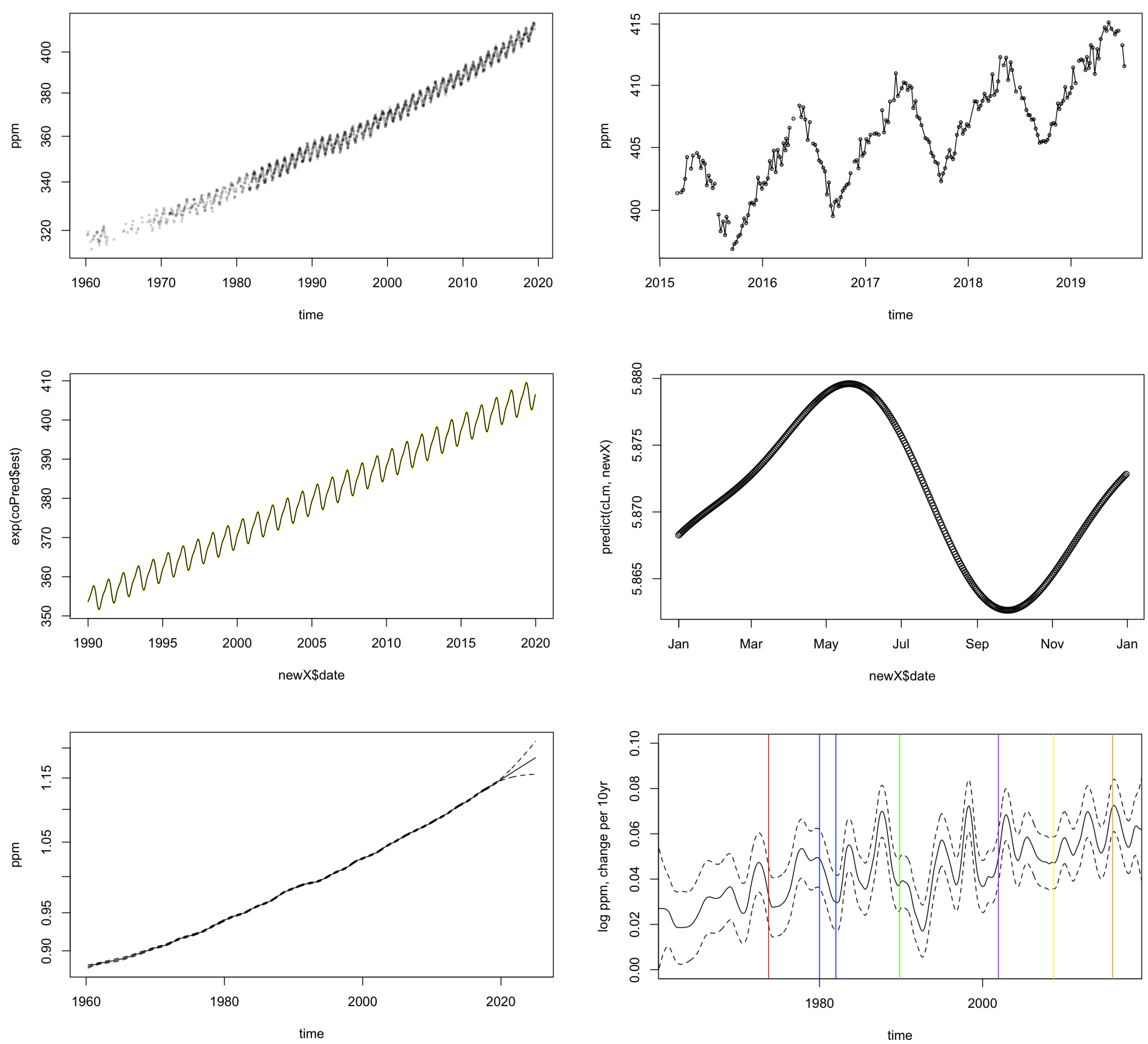
```
                             "junk2", "Nflasks", "quality", "co2"))
co2s$date = strptime(paste(co2s$day, co2s$time),
                   format = "%Y-%m-%d␣%H:%M", tz = "UTC")
# remove low-quality measurements
co2s[co2s$quality >= 1, "co2"] = NA
plot(co2s$date, co2s$co2, log = "y", cex = 0.3, col = "#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")
# reset dates as numeric
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)
library('mgcv')
cLm = gam(co2 ~ days + cos12 + sin12 + cos6 + sin6, data = co2s, family = '
   Gamma'(link = "log"))

knitr::kable(summary(cLm)$p.table[,1:2], digits = 3)
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(cLm, newX, se.fit = TRUE)
coPred = data.frame(est = coPred$fit,
                  lower = coPred$fit - 2 * coPred$se.fit,
                  upper = coPred$fit + 2 * coPred$se.fit)

# forecast = predicted
plot(newX$date, exp(coPred$est), type = "l")
matlines(as.numeric(newX$date), exp(coPred[, c("lower", "upper", "est")]),
        lty = 1, col = c("yellow", "yellow", "black"))
newX = newX[1:365, ]
```

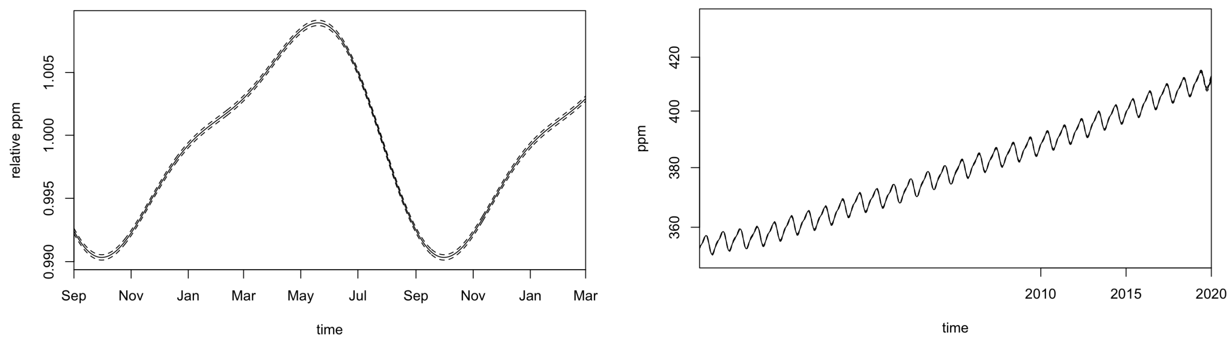
```
# cycle
plot(newX$date, predict(cLm, newX))

library("INLA")
range(co2s$date)
# time random effect
timeBreaks = seq(min(co2s$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)

# plot of random effect
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"))

# plot of 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")))
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"))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(1982, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(1989, 11, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "purple")
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "yellow")
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "orange")

# plot of seasonal
matplot(
  StimeSeason,
  exp(co2res$summary.lincomb.derived[
    grep("season", rownames(co2res$summary.lincomb.derived)),
    c("0.5quant", "0.025quant", "0.975quant")]),
  type = "l",
  col = "black",
  lty = c(1, 2, 2),
  log = "y",
  xaxs = "i",
  xaxt = "n",
  xlab = "time",
  ylab = "relative␣ppm")
xaxSeason = seq(ISOdate(2009, 9, 1, tz = "UTC"), by = "2␣months", len = 20)
axis(1, xaxSeason, format(xaxSeason, "%b"))

# plot of predicted
timePred = co2res$summary.lincomb.derived[
  grep("pred", rownames(co2res$summary.lincomb.derived)),
  c("0.5quant", "0.025quant", "0.975quant")]
matplot(timePoints,
       exp(timePred),
       type = "l",
       col = "black",
       lty = c(1, 2, 2),
       log = "y",
       xlim = ISOdate(c(1990, 2020), 1, 1, tz = "UTC"),
       ylim = c(350, 435),
       xaxs = "i",
       xaxt = "n",
       xlab = "time",
       ylab = "ppm")
xaxPred = seq(ISOdate(2010, 1, 1, tz = "UTC"), by = "5␣years", len = 20)
```

```
axis(1, xaxPred, format(xaxPred, "%Y"))

| | Estimate| Std. Error|
|:-----------|--------:|----------:|
|(Intercept) | 5.822| 0|
|days | 0.000| 0|
|cos12 | -0.002| 0|
|sin12 | 0.008| 0|
|cos6 | 0.002| 0|
|sin6 | -0.002| 0|
```



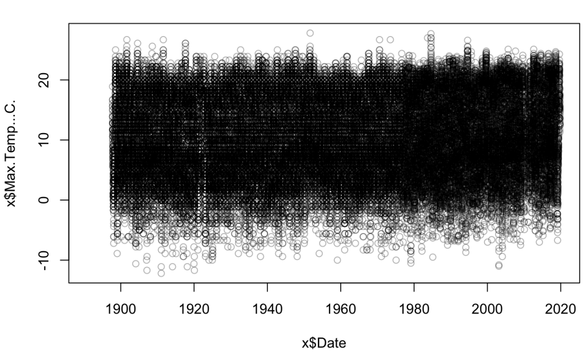


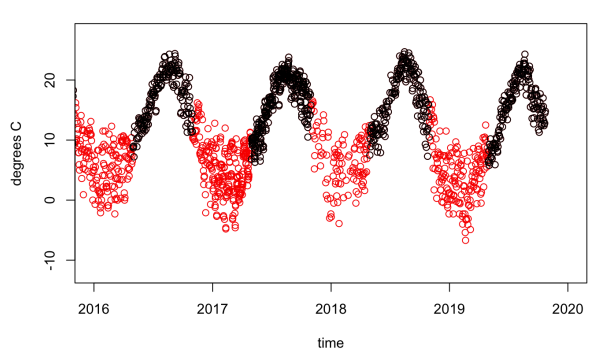

```
# Question 2: Heat
heatUrl = "http://pbrown.ca/teaching/appliedstats/data/sableIsland.rds"
heatFile = tempfile(basename(heatUrl))
download.file(heatUrl, heatFile)
x = readRDS(heatFile)

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$Date),
                 ISOdate(2053, 1, 1, 0, 0, 0, tz = "UTC"),
                 by = "7␣days")
xSub$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$coef[-1])
INLA::inla.doc('^t$')
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())
```

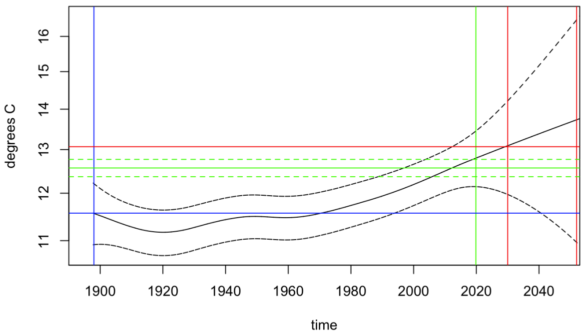
```
## danger zone ##
sableRes = INLA::inla(
  Max.Temp...C. ~ 0 + sin12 + cos12 + sin6 + cos6 +
    f(week, model='rw2', constr=FALSE, prior='pc.prec',
      param = c(0.1/(52*100), 0.05)) +
    f(weekIid, model='iid', prior='pc.prec', param = c(1, 0.5)) +
    f(yearFac, model='iid', prior='pc.prec', 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::priorPostSd(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)

# historical daily max data plot
plot(x$Date, x$Max.Temp...C., col = mapmisc::col2html("black", 0.3))
forAxis = ISOdate(2016:2020, 1, 1, tz = "UTC")

# 2016 - 2020 time series
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"))

# est time trend
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",
       log = "y",
       xlim = ISOdate(c(1890, 2053), 1, 1, tz = "UTC"))
forXaxis2 = ISOdate(seq(1880, 2040, by = 20), 1, 1, tz = "UTC")
axis(1, forXaxis2, format(forXaxis2, "%Y"))
abline(v = ISOdate(2030, 1, 1, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(2052, 1, 1, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(2019, 11, 11, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(1898, 1, 1, 1, tz = "UTC"), col = "blue")
abline(h = first(sableRes$summary.random$week[,paste0(c(0.5, 0.025, 0.975),
                                                "quant")]$"0.5"),
      col = "blue")
abline(h = 1 + first(sableRes$summary.random$week[,paste0(c(0.5, 0.025,
   0.975),
                                                    "quant")]$"0.5"),
      col = "green")
abline(h = 0.8 + first(sableRes$summary.random$week[,paste0(c(0.5, 0.025,
   0.975),
                                                      "quant")]$"0.5"),
      col = "green", lty=2)
abline(h = 1.2 + first(sableRes$summary.random$week[,paste0(c(0.5, 0.025,
   0.975),
                                                      "quant")]$"0.5"),
      col = "green", lty=2)
abline(h = 1.5 + first(sableRes$summary.random$week[,paste0(c(0.5, 0.025,
   0.975),
                                                      "quant")]$"0.5"),
      col = "red", lty=1)
```

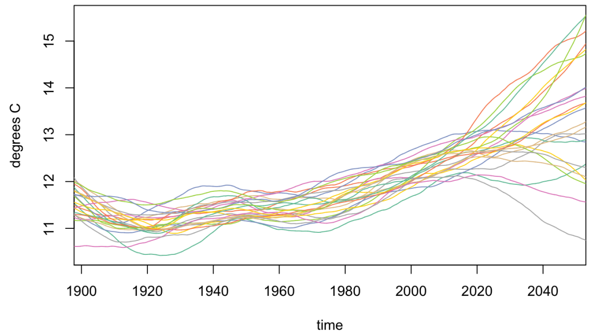
```
myCol = mapmisc::colourScale(NA, breaks = 1:8,
                             style = "unique", col = "Set2", opacity = 0.3)$
                                col

# posterior samples
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"))

> print("initial␣temperature:␣" )
[1] "initial␣temperature:␣"
> first(sableRes$summary.random$week[,paste0(c(0.5, 0.025, 0.975), "quant")]
   $"0.5")
[1] 11.56872
> # upper bound
> print("upper␣bound:␣" )
[1] "upper␣bound:␣"
> first(sableRes$summary.random$week[,paste0(c(0.5, 0.025, 0.975), "quant")]
   $"0.5") + 1.2
[1] 12.76872
> # lower bound
> print("lower␣bound:␣")
[1] "lower␣bound:␣"
> first(sableRes$summary.random$week[,paste0(c(0.5, 0.025, 0.975), "quant")]
   $"0.5") + 0.8
[1] 12.36872
> # if raised by 1.5 degrees
> print("temperature␣if␣raised␣by␣1.5␣degrees␣" )
[1] "temperature␣if␣raised␣by␣1.5␣degrees␣"
> first(sableRes$summary.random$week[,paste0(c(0.5, 0.025, 0.975), "quant")]
   $"0.5") + 1.5
[1] 13.06872
```







degrees C
time



degrees C
time