STA442_HW3

Zeyang Zhang

12/11/2019

CO2 Report

Introduction

We investigate the changes in atmospheric Carbon Dioxide concentrations from an observatory in Hawaii. The oldest data is collected on 1960-03-30 and the newest is collected on 2019-07-09. The data point is sparser when the data is collected earlier. We hope to discuss if the CO2 data appears to be impacted by a series of events.

Methods

We used a Generalized Additive Model from the Gamma family with a log link function to predict carbon concentration. We use time as random walk 2 random effect. We also eliminate the effect of seasonal changes by considering yearly fluctuations and biyearly fluctuations.

$$
\begin{aligned}
&Y_i \sim \Gamma(\lambda_i, \theta_i) \\
&log(\lambda_i \theta_i) = \beta_0 + \beta_1 sin(2\pi x_i) + \beta_2 cos(2\pi x_i) + \beta_3 sin(4\pi x_i) + \beta_4 cos(4\pi x_i) + U(t_i) \\
&U(t) - 2U(t-1) + U(t-2) \sim N(0, \sigma_U^2)
\end{aligned}
$$

We note that Y_i is the response variable, represents the ppm of CO2, following a Gamma distribution with $Gamma(\lambda_i, \theta_i)$. $sin(2\pi x_i)$ and $cos(2\pi x_i)$ represent yearly fluctuations and $sin(4\pi x_i)$ and $cos(4\pi x_i)$ represent biyearly fluctuations. θ_i is the scalar for the gamma distribution. We did not add other random effect beacuse the current model already gives good estimation.

With prior:

- λ_i follows $pr(1/\sqrt{\lambda} > 2) = 0.5$.
- σ_U follows an exponential distribution with $pr(\sigma_U > log(1.01)/26) = 0.5$.

Result

From the CO2 concentration plot (see Appendix), we can say that CO2 concentration increases despite the seasonal changes. Then it is more meaningful to look at the increasing speed of the CO2 concentration.

We approximated the first derivative of the time trend using a finite difference approximation. The approximation is also graphed in figure 1. I add several timelines to indicate the changes during the time indicated.

- The red line corresponds to 1973-10-17, when the OPEC oil embargo began. It seems CO2 concentration increases faster afterwards.
- The orange line and the yellow line corresponds to 1980-1982, global economic recessions. CO2 concentration increases slower during this recession comparing with the time before and afterwards.
- The green line corresponds to 1991-11-1, when the Berlin wall fell. The industrial production is stimulated, and CO2 concentration increases faster after it.
- The blue line corresponds to 2001-12-11, when China joined the WTO and grow rapidly in industrial production afterwards. It seems increasing speed of the CO2 concentration is not influenced significantly by this event.

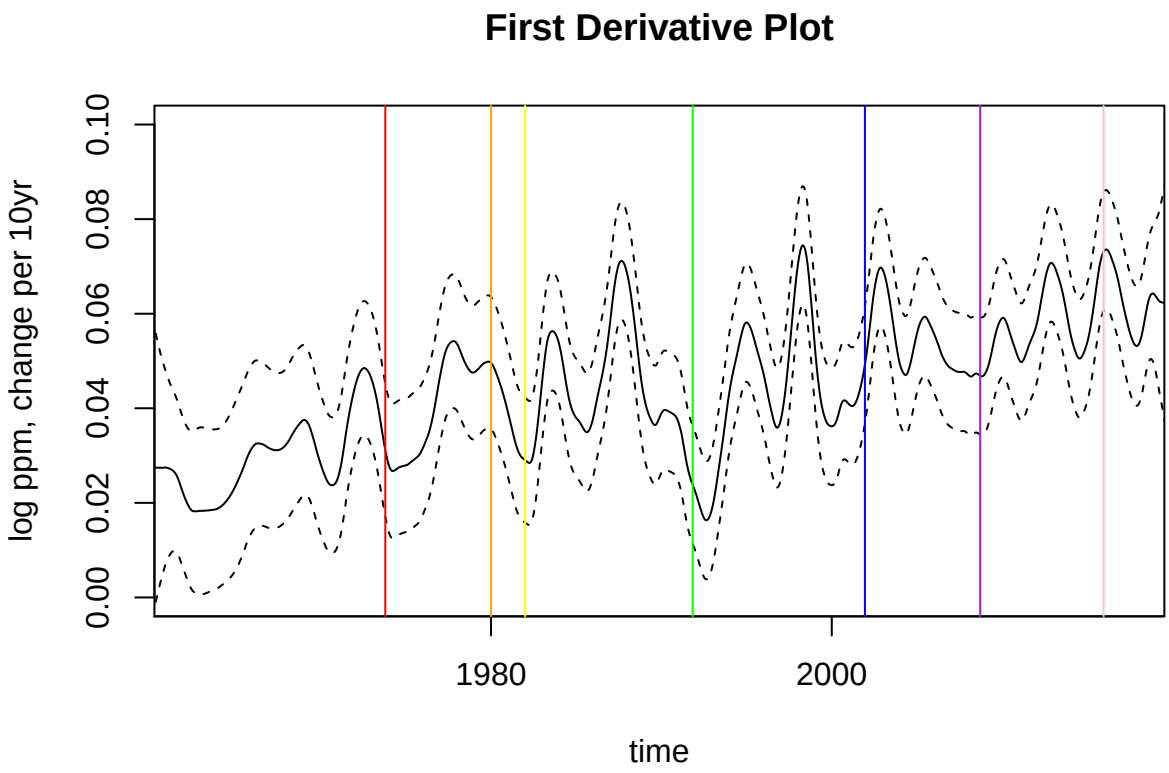


Figure 1: CO2 concentration first derivative plot

- The purple line corresponds to 2008-9-15, when the Lehman Brothers bankrupted, and the most recent global financial crisis started. It seems CO2 concentration increases faster afterwards.
- The pink line corresponds to 2015-12-15, when the Paris Agreement is signed and intended to limit CO2 emissions. It shows that growth rate of CO2 in the atmosphere is lower now that it has been in the recent past.

Heat Report

Introduction

We use daily maximum temperature data recorded on Sable Island from 1900 to now to test whether IPCC statement about the global warming is true or not. The The IPCC states that "Human activities are estimated to have caused approximately 1.0°C of global warming above preindustrial levels, with a likely range of 0.8°C to 1.2°C. Global warming is likely to reach 1.5°C between 2030 and 2052 if it continues to increase at the current rate."

Method

Putting all the data point in one plot does not show any pattern for time series data. Instead, we should used a student-t Bayesian Model with suitable piror to predict the temperature. Only the maximum temperature from summer (month 5-10) are taken to be the response variables as we are advised. We use each week's data as random walk 2 random effect. We should also allow some fluctuations between each week and each year so we take **year** and **week** as iid random effect. Fitting yearly fluctuations and biyearly fluctuations into the model eliminates the effect of seasonal changes.

$$
\begin{aligned}
&\sqrt{s\tau}(y_i - \eta_i) \sim T_\nu \\
&\eta_i = \beta_0 + \beta_1 sin(2\pi x_i) + \beta_2 cos(2\pi x_i) + \beta_3 sin(4\pi x_i) + \beta_4 cos(4\pi x_i) + U(t_i) + U_{i,week} + U_{i,year} \\
&U(t) - 2U(t-1) + U(t-2) \sim N(0, \sigma_U^2) \\
&U_{i,week} \overset{ind}{\sim} N(0, \sigma_{week}^2) \\
&U_{i,year} \overset{ind}{\sim} N(0, \sigma_{year}^2)
\end{aligned}
$$

We note that $sin(2\pi x_i)$ and $cos(2\pi x_i)$ represent yearly fluctuations and $sin(4\pi x_i)$ and $cos(4\pi x_i)$ represent biyearly fluctuations. s is a fixed scaling and $s > 0$, y is the response variable and η is the linear predictor. T_ν is a reparameterized standard Student-t with $\nu > 2$ degrees of freedom with unit variance for all values of ν. U_{week} and U_{year} represent weekly and yearly random effect.

Prior:

- Precision parameter τ follows $pr(1/\sqrt{\tau} > 1) = 0.5$. Note if using student-t change that hyperparameter it will have $pr(log(\tau) > 1) = 0.5$
- The degree of freedom ν has $pr(\nu > 10) = 0.5$. Note if using student-t change that hyperparameter it will have $pr(log(\nu - 2) > 10) = 0.5$
- σ_U follows $pr(\sigma_U > 0.1/(52 * 100)) = 0.05$.
- σ_{week} follows $pr(\sigma_{week} > 10) = 0.5$. We allow a relative large difference between weeks.
- σ_{year} follows $pr(\sigma_{week} > 1) = 0.5$. We allow a relative small difference between weeks.

Result

The Table 1 shows that all fixed effect is significant under 95% CI. The Table 2 shows the standard deviation for random effect. In this table, "sd for week" means the standard deviation of random walk 2 weekly fluctuation and is 0. The total standard deviation is contributed by iid week and year random effect.

Table 1: Estimation of fixed effects

	0.5quant	0.025quant	0.975quant
sin12	-4.688	-5.204	-4.172
cos12	4.770	4.479	5.060

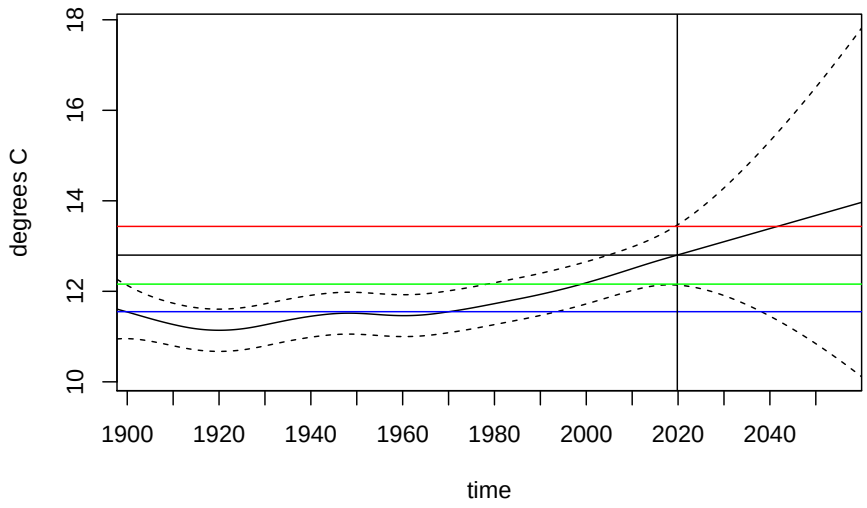


Figure 2: Heat plot prediction under 95 percent credible interval, current prediction

	0.5quant	0.025quant	0.975quant
sin6	-2.043	-2.267	-1.820
cos6	-0.149	-0.321	0.022

Table 2: Estimation of standard deviation

	mean	0.025quant	0.975quant
sd for t	1.770	1.751	1.790
sd for week	0.000	0.000	0.000
sd for weekIid	1.093	1.053	1.138
sd for yearFac	0.694	0.603	0.806

Based on our model, we generate a 95% credible interval prediction plot (Figure 2).

We use the blue line represents the base line, which means basically the preindustrial temperature. In the beginning of 20th century, when human activities have not influenced the temperature, the median temperature is 11.55. From the plot, we can see that the 95% CI contain that temperature for a rather long period.

Nowadays (2019-11-1), the median temperature is approximately 12.80 degrees (marked in the black line). The lower bound is 12.15 (green line) and the upper bound is 13.45 (red line). Our result shows that there is approximately 1.25 °C of global warming above preindustrial levels, with a likely range of 0.6°C to 1.9°C by now. This is similar to the IPCC's statement: "Human activities are estimated to have caused approximately 1.0°C of global warming above preindustrial levels, with a likely range of 0.8°C to 1.2°C".

In another plot (Figure 3), we predict that by 2030, the median temperature will be 13.10 (black vertical line) and by 2052, the median temperature will be 13.74 (black vertical line). The upper and lower bound become larger beacuse the longer period makes the prediction less precise. The median temperature has increased 1.55°C by 2030 and 2.19°C by 2052. There is clear tendency that median line continue to increase. So the result was slightly different from the IPCC statement that "Global warming is likely to reach 1.5°C between 2030 and 2052.". The number 1.5°C was more conservative for the temperature changes between 2030 and 2052.

In Figure 4, we generate a larger sample illustrating some possible prediction curves based on the model. Though it is possible that the temperature can go down again, the trend is clearly upward.

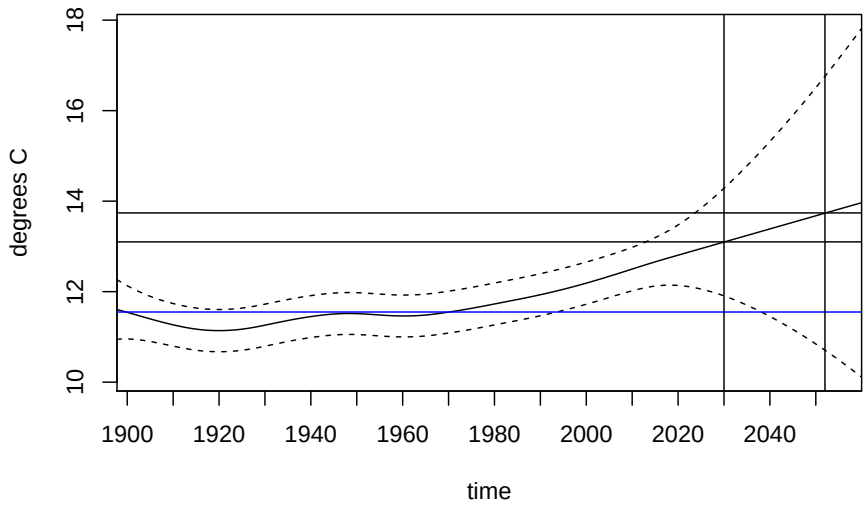


Figure 3: Heat plot prediction under 95 percent credible interval, future prediction

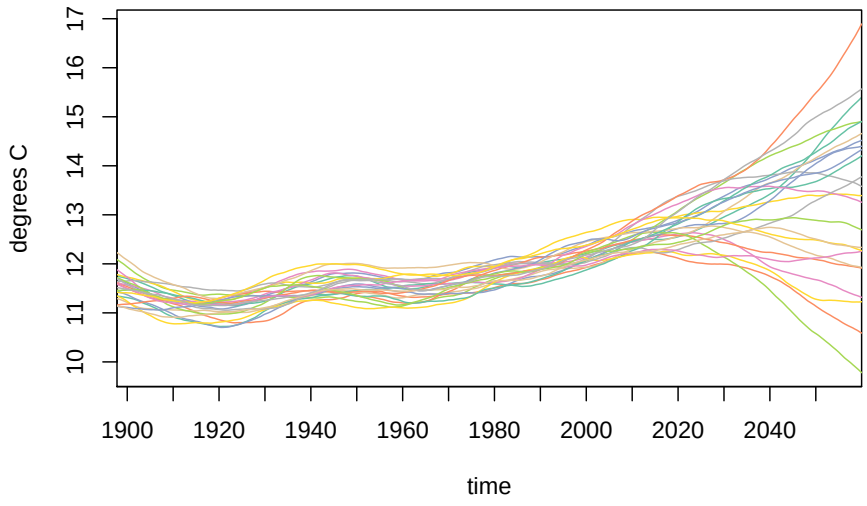


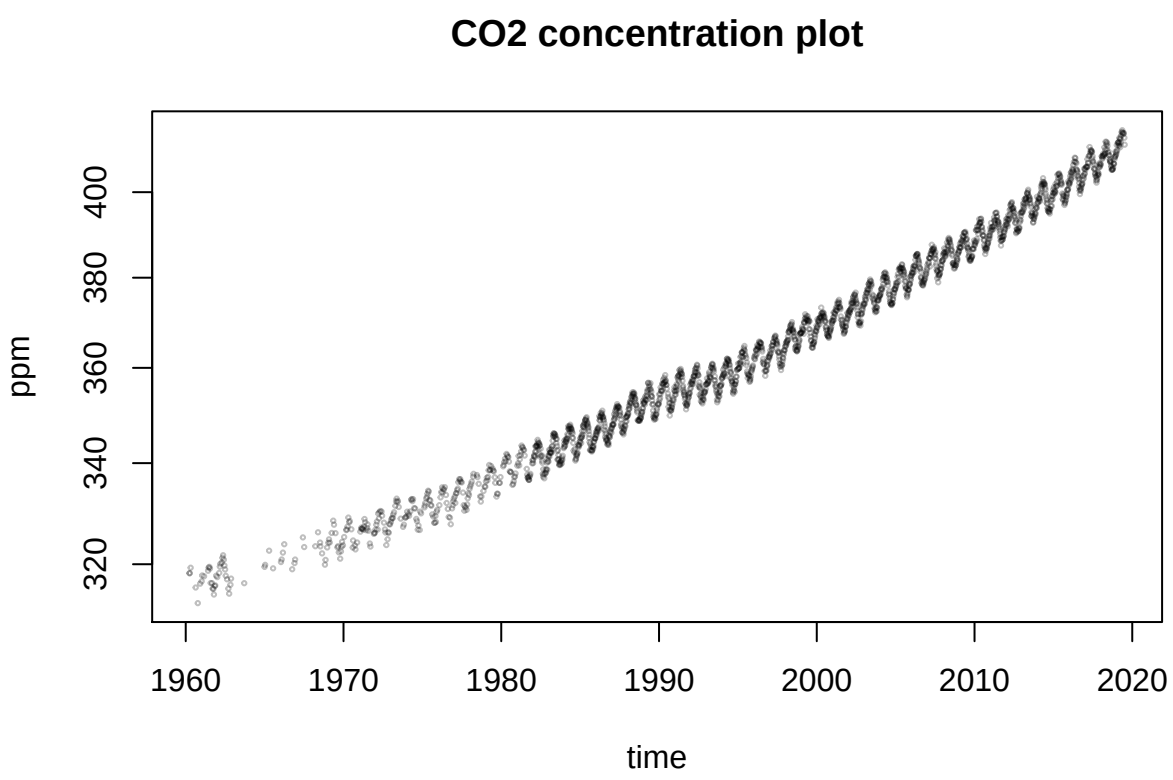
Figure 4: Sample predictions form the model

Conclusion

In conclusion, the IPCC statement is rather reliable. Though the number are slightly different, the trend is predicted well. Some numbers the IPCC use is even more conservative. Considering data was recorded on Sable Island, off the coast of Nova Scotia, it is understandable that the climate change is more significant for places near the Arctic.

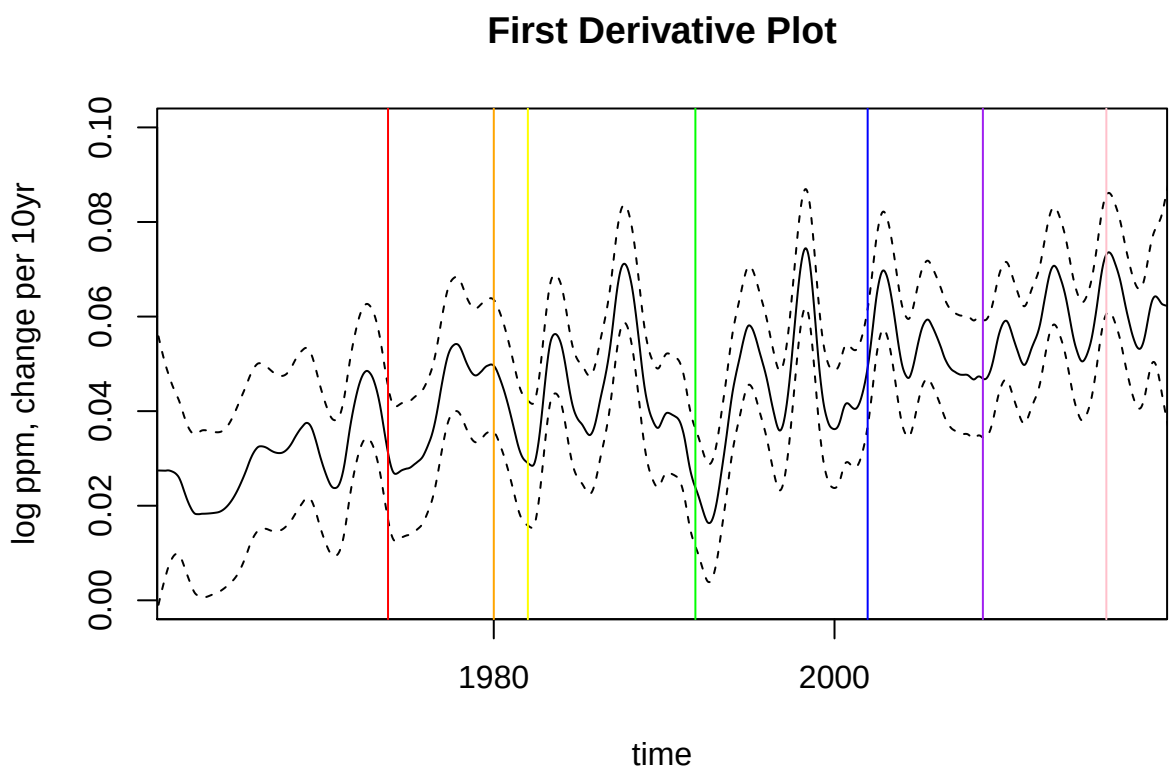
Appendix

```
# Co2 report
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",main = "CO2 concentration plot")
```

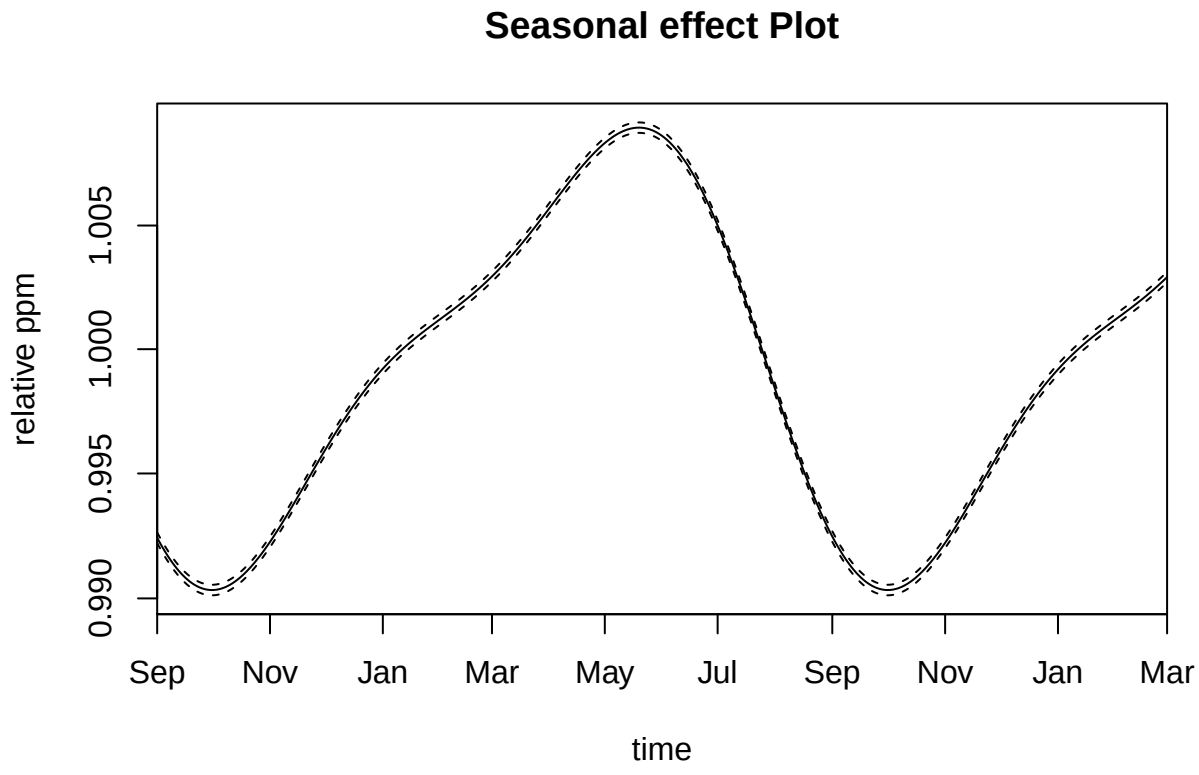



```
timeOrigin = ISOdate(1980, 1, 1, 0, 0, 0, tz = "UTC")
co2s$days = as.numeric(difftime(co2s$date, timeOrigin, units = "days"))
co2s$cos12 = cos(2 * pi * co2s$days/365.25)
co2s$sin12 = sin(2 * pi * co2s$days/365.25)
co2s$cos6 = cos(2 * 2 * pi * co2s$days/365.25)
co2s$sin6 = sin(2 * 2 * pi * co2s$days/365.25)

library("INLA")
# time random effect
timeBreaks = seq(min(co2s$date), ISOdate(2025, 1, 1, tz = "UTC"), by = "14 days")
timePoints = timeBreaks[-1]
```

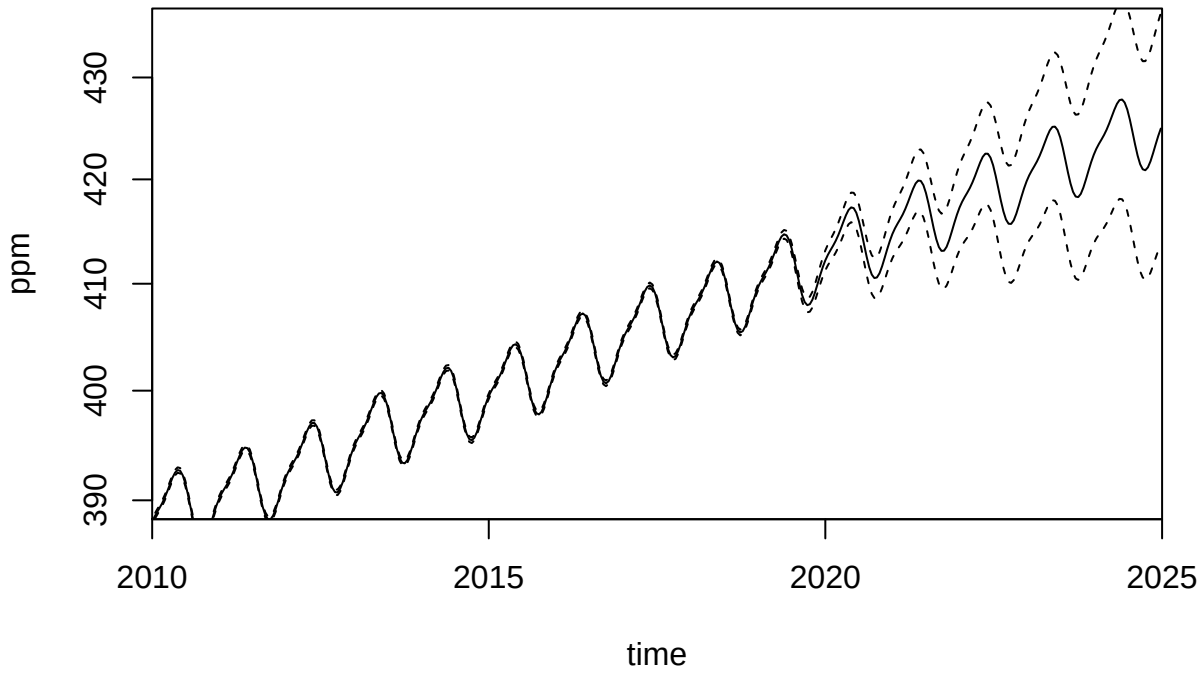
```
co2s$timeRw2 = as.numeric(cut(co2s$date, timeBreaks))
# derivatives of time random effect
D = Diagonal(length(timePoints)) - bandSparse(length(timePoints), k = -1)
derivLincomb = inla.make.lincombs(timeRw2 = D[-1, ])
names(derivLincomb) = gsub("^lc", "time", names(derivLincomb))
# seasonal effect
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)))),
  control.inla = list(strategy='gaussian', int.strategy='eb'),
  verbose=TRUE)

xax = pretty(timePoints)

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", main = "First Derivative Plot")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1973, 10, 17, tz = "UTC"), col = "red")
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(1982, 1, 1, tz = "UTC"), col = "yellow")
abline(v = ISOdate(1991, 11, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "blue")
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "purple")
abline(v = ISOdate(2015, 12, 15, tz = "UTC"), col = "pink")
abline(v = ISOdate(2019, 11, 1, tz = "UTC"), col = "orchid")
```




```
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", main = "Seasonal effect Plot")
xaxSeason = seq(ISOdate(2009, 9, 1, tz = "UTC"), by = "2 months", len = 20)
axis(1, xaxSeason, format(xaxSeason, "%b"))
```




```
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(2010, 2025), 1, 1, tz = "UTC"),
        ylim = c(390, 435),xaxs = "i", xaxt = "n",
        xlab = "time", ylab = "ppm",main = "Prediction Plot")
xaxPred = seq(ISOdate(2010, 1, 1, tz = "UTC"), by = "5 years", len = 20)
axis(1, xaxPred, format(xaxPred, "%Y"))
```




```
## heat report
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$Date, "%Y"))
lmStart = lm(Max.Temp...C. ~ sin12 + cos12 + sin6 + cos6, data = xSub)
startingValues = c(lmStart$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::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(10, 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)
knitr::kable(sableRes$summary.fixed[, c(4, 3, 5)], digits=3, caption = "Estimation of fixed effects",)
```

Table 3: Estimation of fixed effects

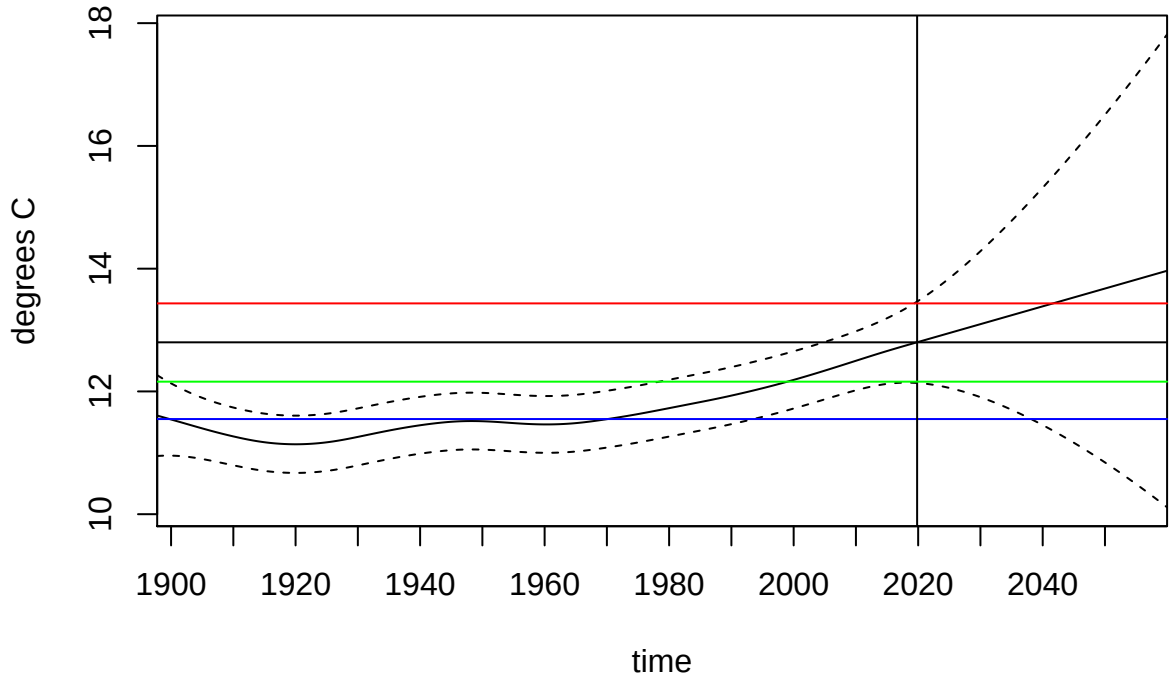
	0.5quant	0.025quant	0.975quant
sin12	-4.688	-5.204	-4.172
cos12	4.770	4.479	5.060
sin6	-2.043	-2.267	-1.820
cos6	-0.149	-0.321	0.022

```
knitr::kable(Pmisc::priorPost(sableRes)$summary[, c(1, 3, 5)], digits=3,
             caption = "Estimation of standard deviation",)
```

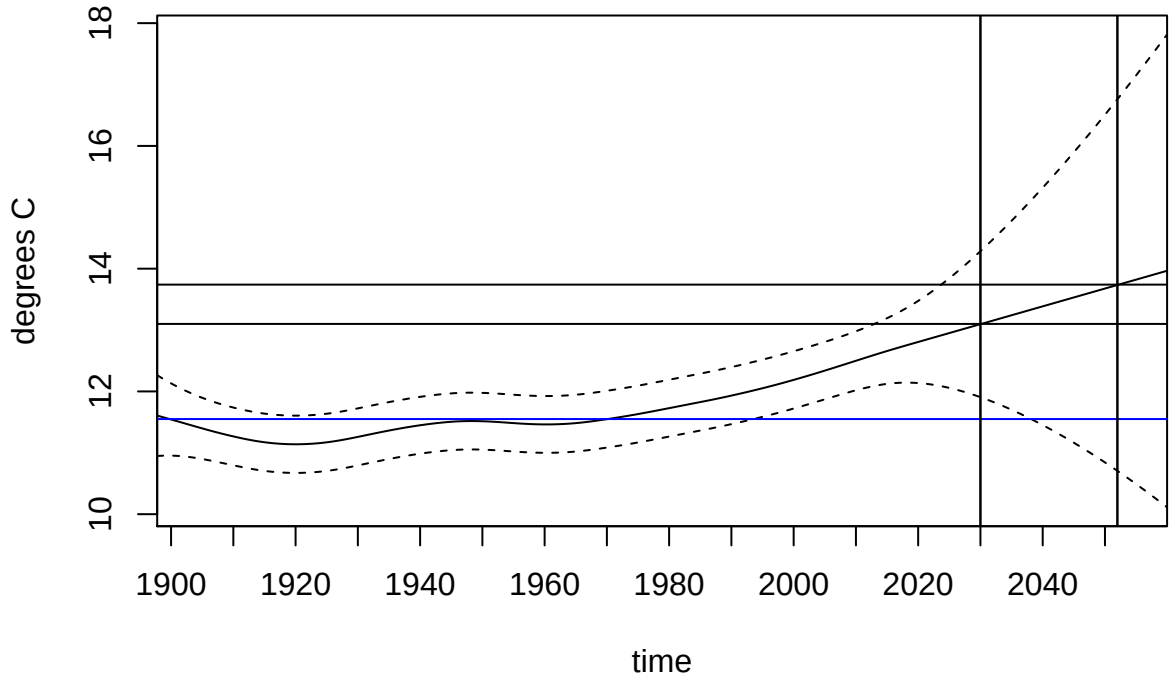
Table 4: Estimation of standard deviation

	mean	0.025quant	0.975quant
sd for t	1.770	1.751	1.790
sd for week	0.000	0.000	0.000
sd for weekIid	1.093	1.053	1.138
sd for yearFac	0.694	0.603	0.806

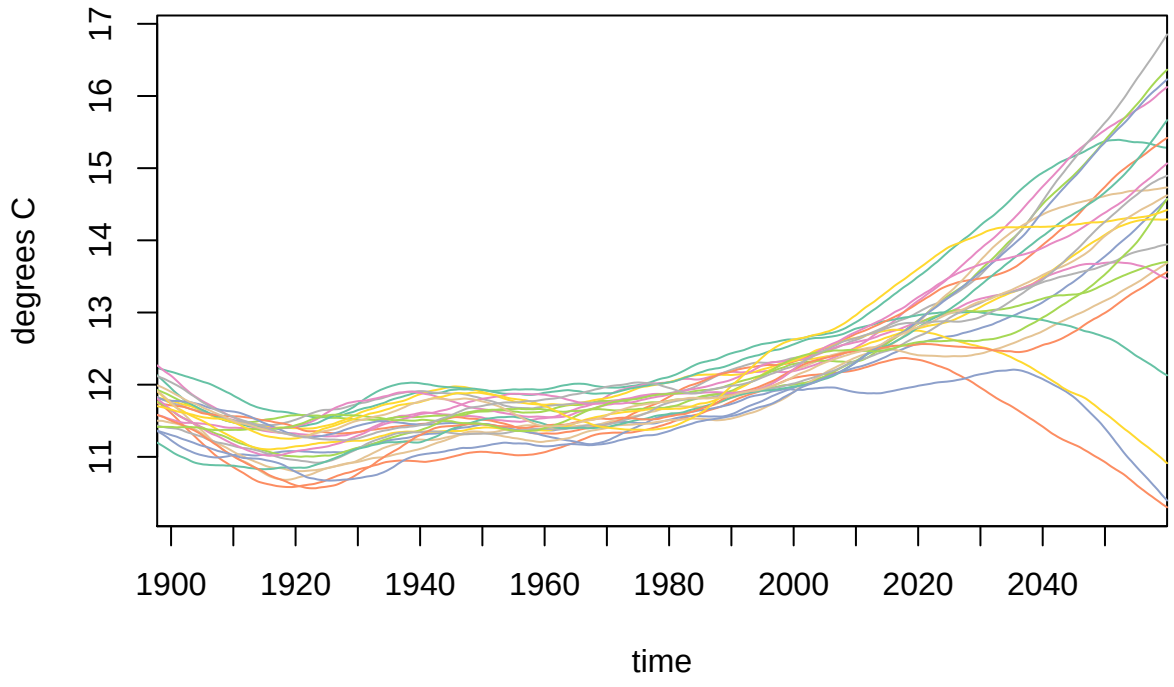
```
mySample = inla.posterior.sample(n = 24,
  result = sableRes, num.threads = 8,
  selection = list(week = seq(1,
  nrow(sableRes$summary.random$week))))
weekSample = do.call(cbind, lapply(mySample, function(xx) xx$latent))
matplot(weekValues[-1], sableRes$summary.random$week[,
  paste0(c(0.5, 0.025, 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 = 10), 1, 1, tz = "UTC")
axis(1, forXaxis2, format(forXaxis2, "%Y"))

abline(h=11.55 , col="blue")
abline(v = ISOdate(2019, 11, 1, tz = "UTC"), col = "black")
abline(h=12.8, col="black")
abline(h=12.15899 , col="green")
abline(h=13.43446 , col="red")
```




```
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 = 10), 1, 1, tz = "UTC")
axis(1, forXaxis2, format(forXaxis2, "%Y"))
abline(h=11.55 , col="blue")
abline(h=13.10 , col="black")
abline(h=13.74 , col="black")

abline(v = ISOdate(2030, 1, 1, tz = "UTC"), col = "black")
abline(v = ISOdate(2052, 1, 1, tz = "UTC"), col = "black")
abline(v = ISOdate(2030, 1, 1, tz = "UTC"), col = "black")
abline(v = ISOdate(2052, 1, 1, tz = "UTC"), col = "black")
```




```
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"))
```


degrees C
11
12
13
14
15
16
17
1900
1920
1940
1960
1980
2000
2020
2040
time