

STA442 Assignment 3

Xin Wei

12/11/2019

CO2 Concentration Change After Events

In our analysis of CO2 concentration at Mauna Loa Observatory in Hawaii, we discovered that out of the six events, only the event of China joining the WTO caused an higher growth rate of CO2 emission. And all the other five events, including the OPEC oil embargo, the global economic recessions, the fall of the Berlin wall, the bankruptcy of Lehman Brothers, and the signing of the Paris Agreement, were followed by a lower growth rate of CO2 emission.

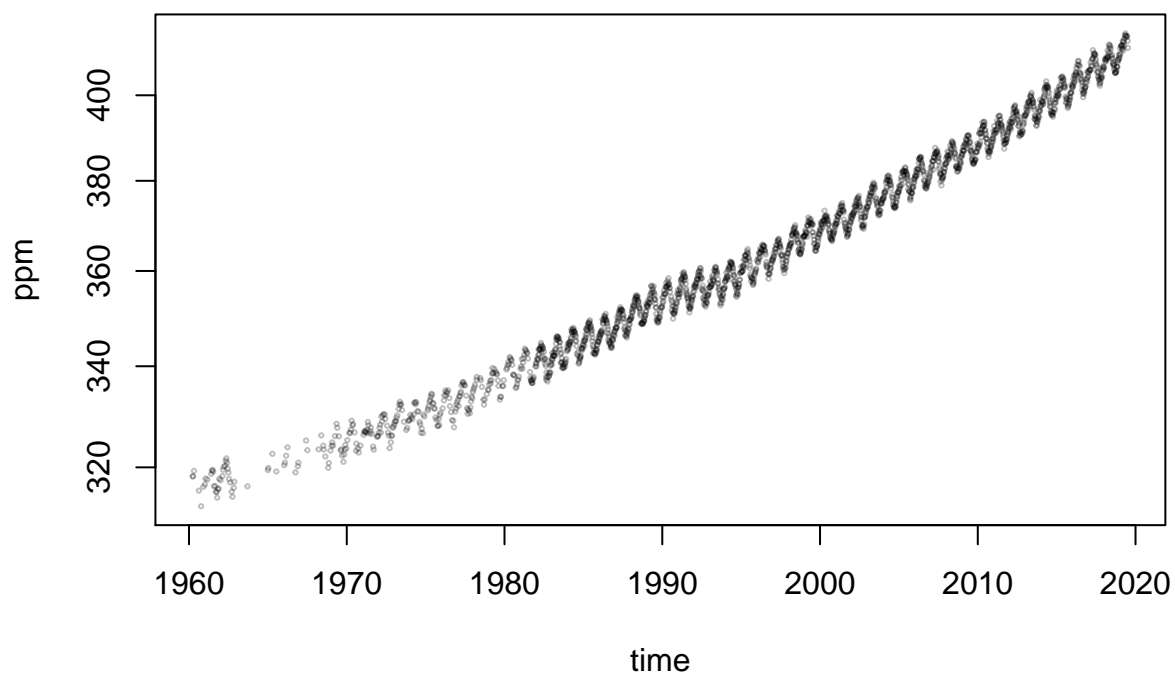
It has been suspected that the concentration of atmospheric CO2 might be increasing in the atmosphere due to fossil fuel combustion, which has become a serious environmental concern. We want to discover whether CO2 data was impacted by some big events in human history. We used the In-situ CO2 data from Hawaii station, provided by the Scripps CO2 Program.

The measurement of CO2 concentration in the Scripps CO2 Program began in 1958 at Mauna Loa Observatory, which located in 19.5 °N, 155.6 °W, and the elevation was 3397m. Monthly values of CO2 concentration are included in the data set from year 1958 up to 2019. As an explanatory tool, based on historical data, we assume penalized complexity prior distribution for our random effect, the random walk of bi-weekly variation and then assume that $P(\sigma_U^2 > \log(1.01)/26) = 0.5$. After obtaining the data, we use INLA to perform the approximate Bayesian inference for the posterior distribution. Then we use a semi-parametric generalized additive model from gamma family as such: $Y_i \sim \Gamma(\alpha, \beta)$, and $\log(E(Y_i)) = \sin(2\pi x_i)\beta_1 + \cos(2\pi x_i)\beta_2 + \sin(4\pi x_i)\beta_3 + \cos(2\pi x_i)\beta_4 + U(t_i)$, where Y_i is the CO2 concentration measured on each day, starting from Jan 1st, 1970. $E(Y_i) = \frac{\alpha_i}{\beta_i}$. Yearly and biyearly fluctuations(fixed effects) are measured by sine and cosine functions. $U(t) - 2U(t-1) + U(t-2) \sim N(0, \sigma_U^2)$ is a random walk of order 2, we use this random effect because there is a constant slope in the time series plot, which shows non-stationarity.

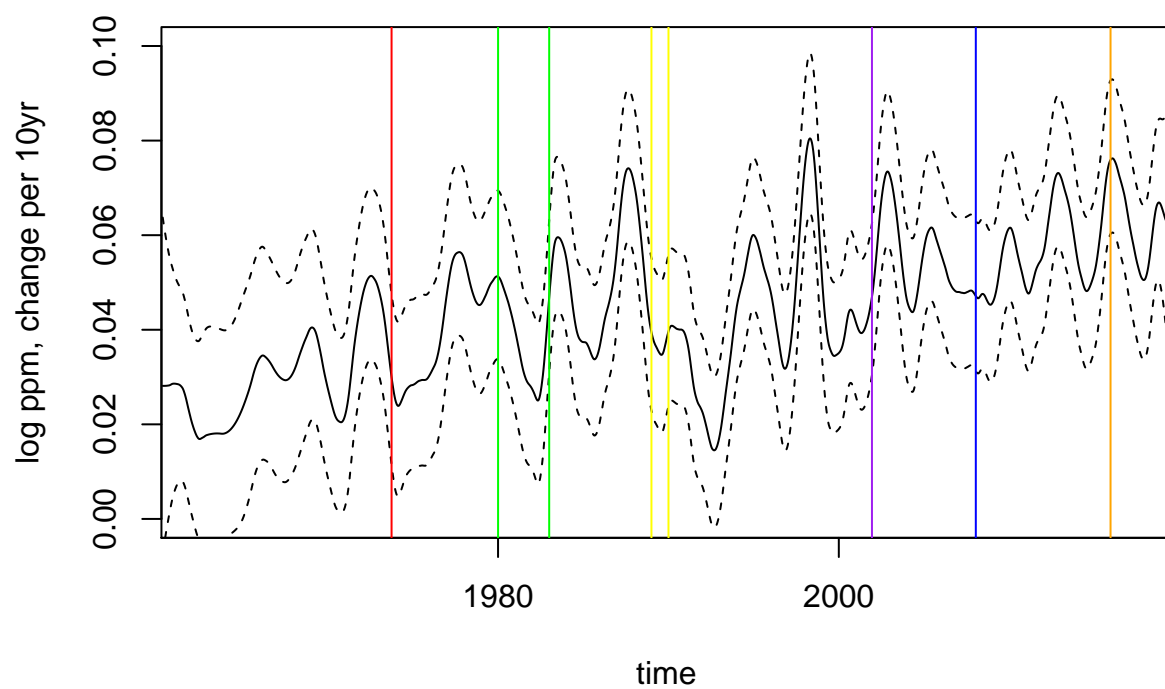
As seen in the graph below, the GAM is roughly a good fit from the data. The data of the CO2 concentration is not stationary time series as it shows an increasing mean and seasonal trend over time.

In the derivative plot, we see that after the OPEC oil embargo in 1973, the rate of concentration growth dropped a bit in the following few months but then started to grow again. During the global economic recessions around 1980-1982, the rate of growth dramatically dropped roughly a half and rose to a local peak during the recovery. In the third event that the fall of Berlin wall around 1989, the growth rate showed a fluctuation in a short time, but then a significant decrease may due to the fall in industrial production in the Soviet Union and Eastern Europe. After China joined the WTO on Dec 11th 2001, CO2 concentration rose rapidly because of the development in industrial production in China. After the most recent global financial crisis began on 15 September 2008, the CO2 emission did not seem to be affected a lot, despite a short fall in growth rate. The signing of the Paris Agreement on 12 December 2015, impacted on the increase of CO2 emission, and we can observe a great fall in the rate. Although the growth rate has been roughly doubled since 1973, it is lower now than it has been in the recent past.

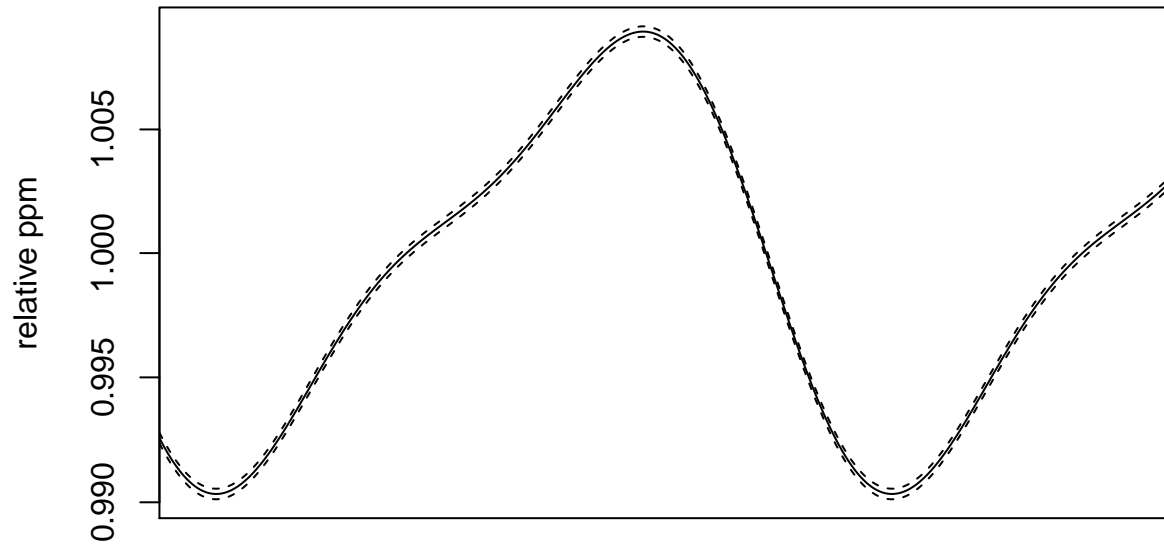
CO2 Concentration Time Series Since 1960



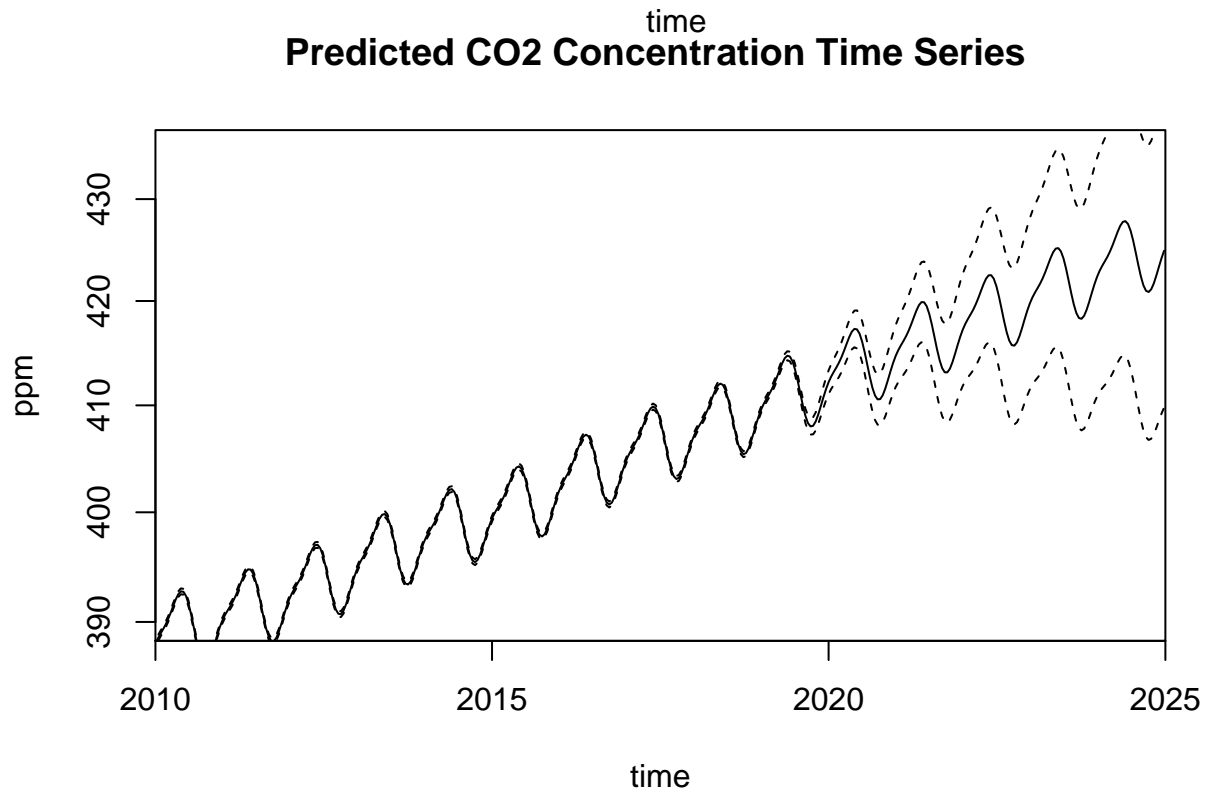
Derivative of the CO2 Concentration Estimates



Seasonal Effect on DEC 12th 2015



Predicted CO2 Concentration Time Series



Will the Global Temperature Rise by 1.5 Degrees?

Summary

In our analysis of daily temperature measured at Sable Island over time starting from year 1900, we found there is an increasing trend in the temperature. Specifically, we are very confident that the rise will be 1.5 celsius degrees by the year of 2053. This conclusion is consistent with the statements from IPCC.

Introduction

After the signing of the Paris Agreement in 2015, the Intergovernmental Panel on Climate Change (IPCC) was invited to show a special report on the prediction of global warming of 1.5 degrees above pre-industrial levels. However, the People's Party questions the statement because they think the model IPCC used was flawed and it incorrectly predicted the future temperatures. They also state that higher levels of CO2 concentration in the atmosphere is beneficial for agriculture. We used the records collected daily on Sable Island to explore whether the prediction of IPCC was supported by our data.

Methods

As suggested by IPCC's results, we make some assumptions that the temperature time series is not constant over time, but it has a certain slope. As an explanatory tool, based on historical data, we assume penalized complexity prior distribution for our random effects, weeks and years. Specifically, for both weeks and years, we assume $P(\sigma^2 > 1) = 0.5$. Since the degrees of freedom is a prior parameter of the shape of T-distribution, we plot a T -distribution with degree of freedom of 10 to fit the histogram of our data, and we see this is roughly a good fit. So we want to use this distribution and its prior is assumed that $P(u > 10) = 0.5$. After obtaining the data, we use INLA to perform the approximate Bayesian inference for the posterior distribution, which is proportional to prior distribution. We want to fit a generalized additive model with T_{10} family as follows:

$$y_i \sim \eta + T_{10}/\sqrt{s\tau}$$

$$E(y_i) = \sin(2\pi x_i)\beta_1 + \cos(2\pi x_i)\beta_2 + \sin(4\pi x_i)\beta_3 + \cos(4\pi x_i)\beta_4 + U(t_i) + W_i + F_i$$

where:

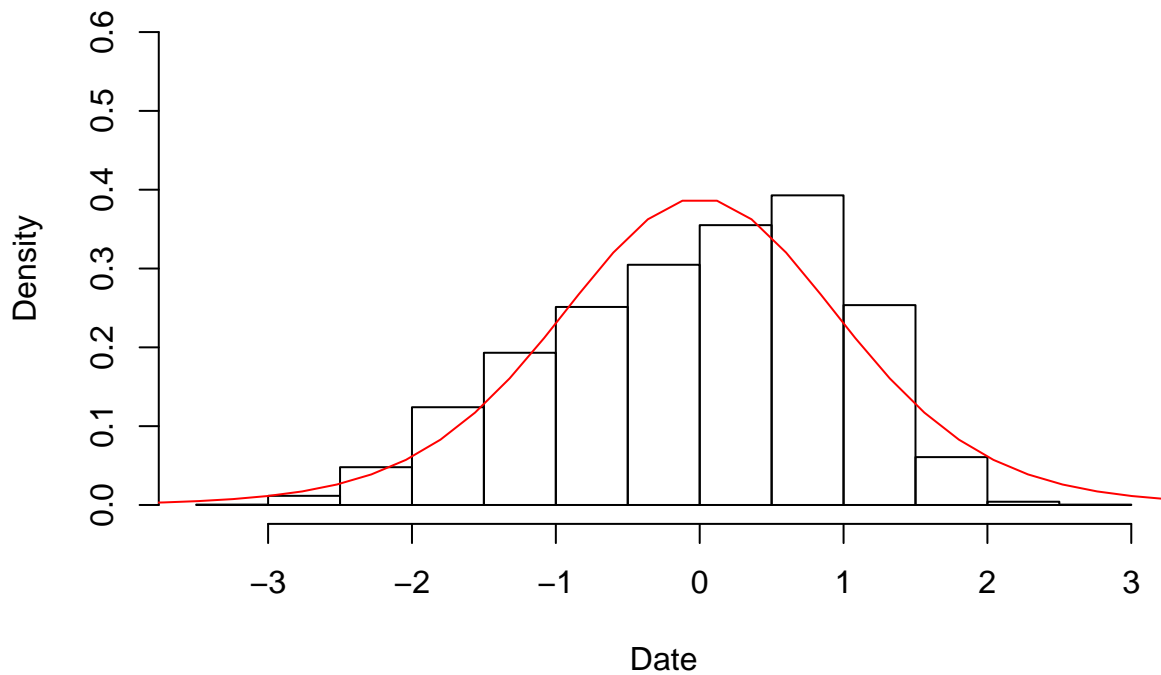
Y_{ij} is the daily maximum temperature, sine and cosine functions predict the yearly and biyearly fluctuations of the data, which are fixed effects. We used $U(t_i)$, which is a random walk of order 2, i.e., $U(t) - 2U(t-1) + U(t-2) \sim N(0, \sigma_U^2)$, that is because we observed the time series of our data is non-stationary, and there is a constant slope. Besides, W_i and F_j are random intercepts, which represent weekly and yearly variations, and follow iid Normal distributions.

Results

When we take a look at the table of standard deviation of posterior of random effects, we see that the yearly variation is the largest(6.9). As in the full scatter plot of daily maximum temperatures, we observe that over the period 1900-2020, the values span consistently from roughly -10 celsius degrees to roughly 24 degrees. Then we zoom in to the period from 2016 to the present. Since winter temperatures are more variable, we only consider the records in summer. As seen in the graph, the summer records are roughly stationary, that is, we don't see much change in the temperature over the time period. However, in the estimated time trend plot, from the starting point estimate at roughly 11.6 degrees in 1900 with a 95% credible interval (10.9, 12.2), the point estimate is predicted to reach 13.1 around year 2030, with a 95% credible interval (11.6, 14). That is to say, we are 95% confident that the global temperature will increase by 1.5 degrees with a 95% credible interval (0.7, 1.8). And by the year of 2052, the point estimate is expected to reach 13.5 with a 95%

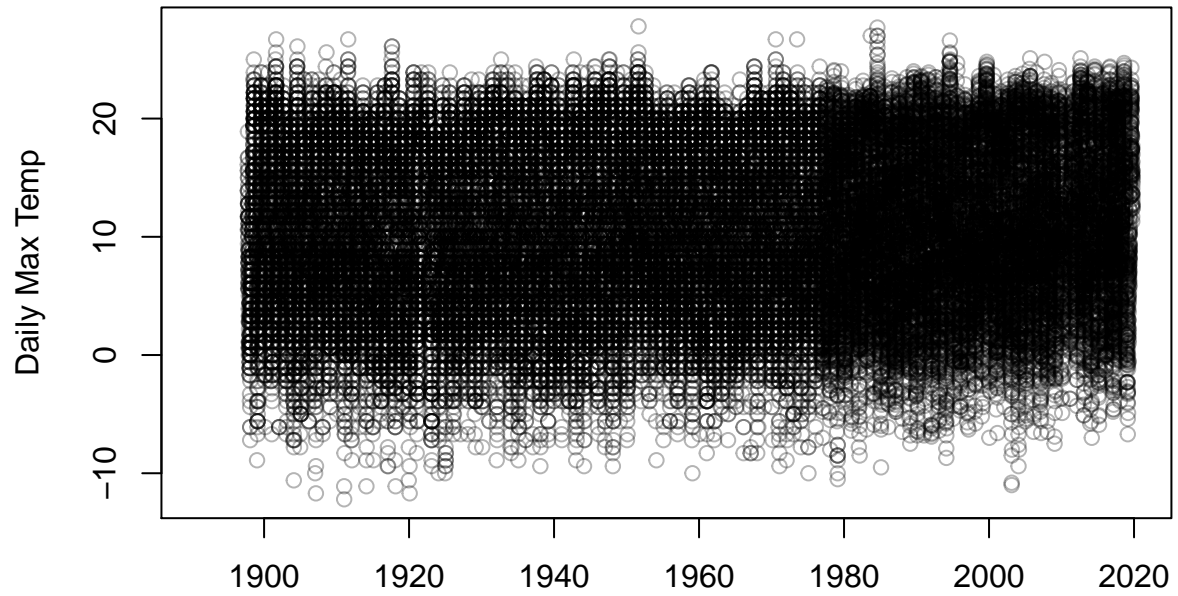
credible interval (11.4, 16), which indicates an increase of 1.9 degrees in temperature. Moreover, in the last plot, the posterior samples have different intercepts due to their weekly and yearly variations, and they have positive slopes because there exists a random effect, which is a random walk of order 2. We observe that the overall trend of different posterior samples are roughly the same, and the average increase is approximately 1.5 degrees from 1900 to 2052. Hence, we conclude our result is consistent with the statements from IPCC that the temperature increase is predicted to reach 1.5 degrees between year 2030 and 2052.

Histogram for Daily Max Temp

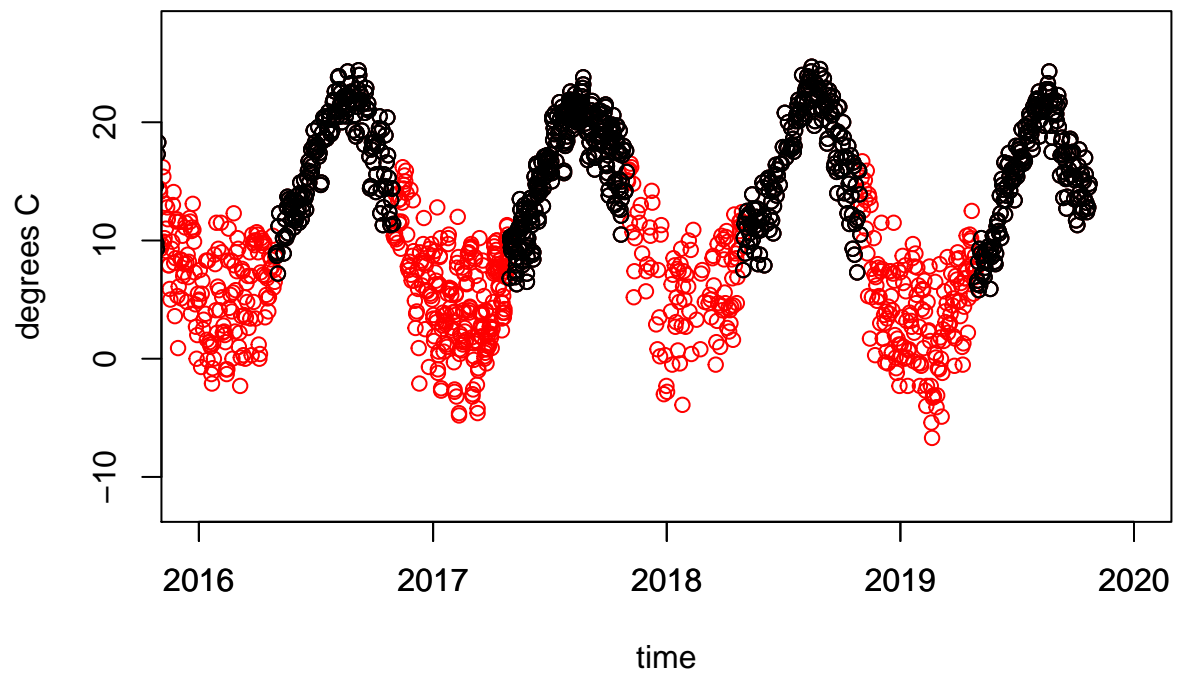


```
##               mean    0.025quant    0.975quant
## SD for week      1.885757e-05 1.455963e-05 2.391035e-05
## SD for weekIid 1.089871e+00 1.056139e+00 1.123806e+00
## SD for yearFac 6.906039e-01 6.108434e-01 7.859561e-01
## [1] 24
## [1] "hyperpar" "latent"  "logdens"
```

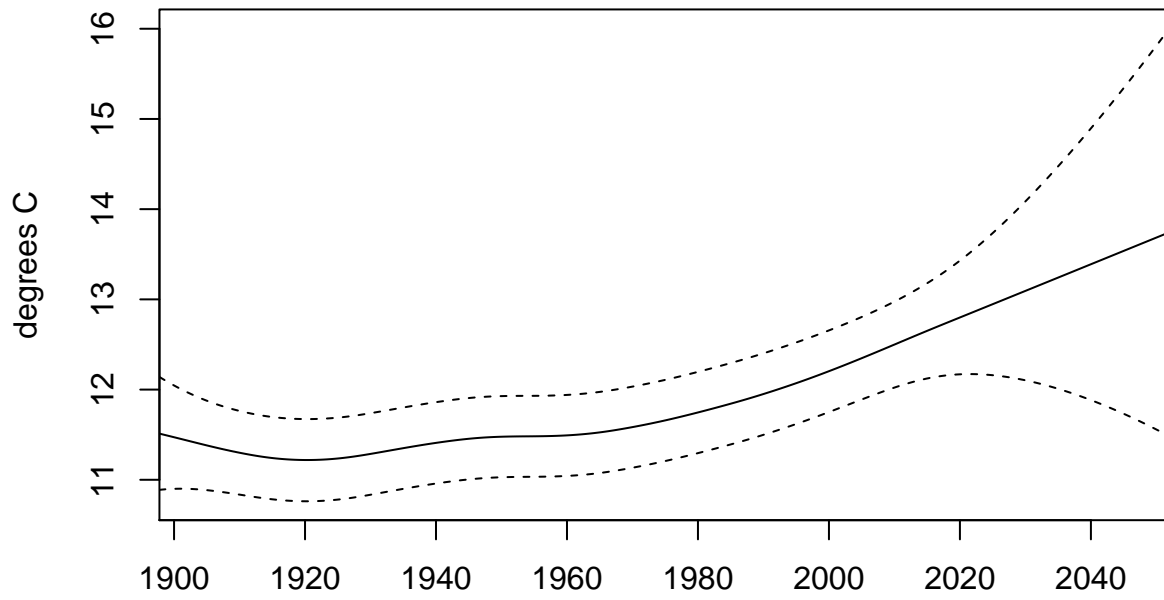
Daily Maximum Temperatures



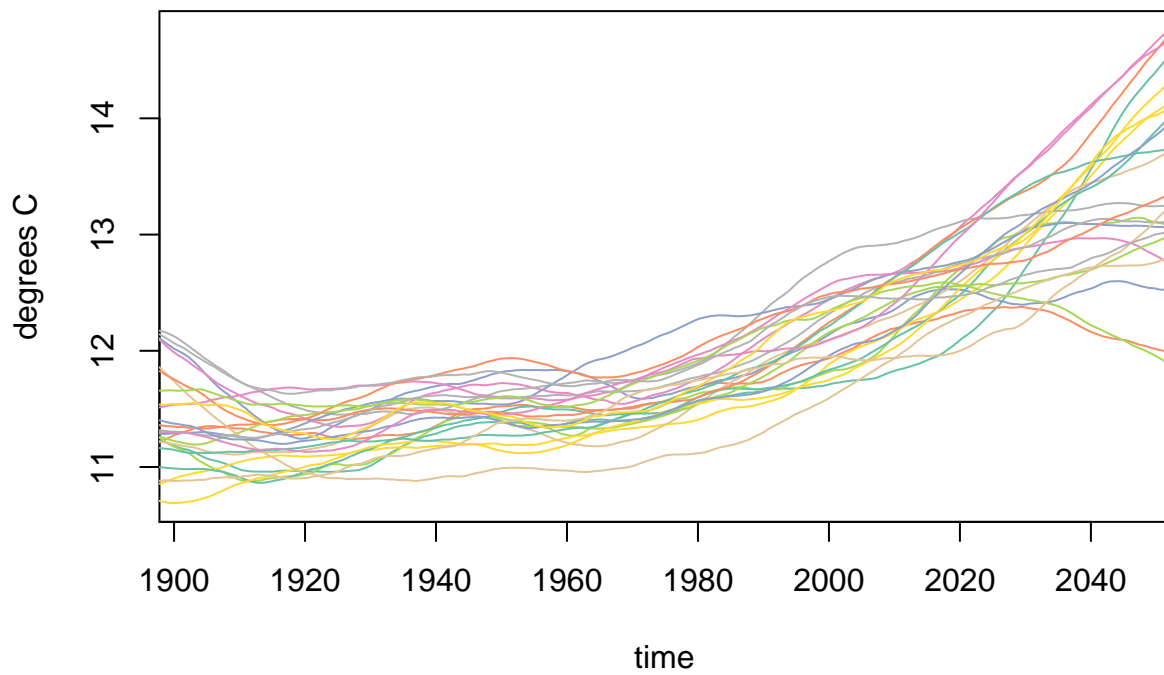
Seasonal Records of Daily Max Temp From 2016



Estimated Time Trend



Posterior Samples



Appendix

```
#CO2
cUrl = paste0("http://pbrown.ca/teaching/appliedstats/data/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 Time Series Since 1960")
#plot(co2s[co2s$date > ISOdate(2015, 3, 1, tz = "UTC"),c("date", "co2")], log = "y", type = "o", xlab =
timeOrigin = ISOdate(1970, 1, 1, 0, 0, 0, tz = "UTC")
co2s$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]
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$se.fit)
#plot(newX$date, coPred$est, type = "l")
#matlines(as.numeric(newX$date), coPred[, c("lower", "upper", "est")], lty = 1, col = c("yellow", "yellow", "yellow"))
newX = newX[1:365, ]
newX$days = 0
#plot(newX$date, predict(cLm, newX))
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, yax, 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$date)),
xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr", main="Derivative of the CO2 C
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
#the global economic recessions
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(1983, 1, 1, tz = "UTC"), col = "green")
#the fall of the Berlin wall
abline(v = ISOdate(1989, 1, 1, tz = "UTC"), col = "yellow")
abline(v = ISOdate(1990, 1, 1, tz = "UTC"), col = "yellow")
#China joining the WTO
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "purple")
#the bankruptcy of Lehman Brothers
abline(v = ISOdate(2008, 1, 15, tz = "UTC"), col = "blue")
#the signing of the Paris Agreement
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "orange")
#seasonal effect plot
matplot(StimeSeason, exp(co2res$summary.lincomb.derived[grep("season", rownames(co2res$summary.lincomb.d
type = "l", col = "black",
lty = c(1, 2, 2), log = "y", xaxs = "i", xaxt = "n",
xlab = "time", ylab = "relative ppm", main="Seasonal Effect on DEC 12th 2015")
xaxSeason = seq(ISOdate(2015, 12, 12, tz = "UTC"), by = "2 months", len = 20)
axis(1, xaxSeason, format(xaxSeason, "%b"))

#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(2010, 2025), 1, 1, tz = "UTC"),
ylim = c(390, 435),
xaxs = "i", xaxt = "n",
xlab = "time", ylab = "ppm",
main="Predicted CO2 Concentration Time Series ")
#abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "green")

xaxPred = seq(ISOdate(2010, 1, 1, tz = "UTC"), by = "5 years",
len = 20)
axis(1, xaxPred, format(xaxPred, "%Y"))

#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$Date), ISOdate(2052, 1, 1,
0, 0, 0, tz = "UTC"), by = "7 days")
xSub$week = cut(xSub$Date, weekValues)
xSub$weekId = 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$week)), rep(0, nlevels(xSub$weekId) +
nlevels(xSub$yearFac)), lmStart$coef[-1])

maxTemp <- xSub$Max.Temp...C.
stdMax <- (maxTemp - mean(maxTemp))/sd(maxTemp)
hist(stdMax, prob = TRUE, main = "Histogram for Daily Max Temp", xlab = "Date", ylim=c(0,0.6))
xSeq = seq(-120, 120, len = 1000)
lines(xSeq, dt(xSeq, 10), col = "red")

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(weekId, 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))

plot(x$Date, x$Max.Temp...C., col = mapmisc::col2html("black",0.3), xlab="Date", ylab="Daily Max Temp",
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", main="Seasonal Records of Daily Max Temp From 2016")
points(xSub$Date, xSub$Max.Temp...C.)
axis(1, forAxis, format(forAxis, "%Y"))

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",
        main="Estimated Time Trend")
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", main="Posterior Samples")
axis(1, forXaxis2, format(forXaxis2, "%Y"))

```