STA442 HW3

Xinyue Jiang

Question 1

Introduction

We analyzed the atmoshperic Carbon Dioxide concentrations from an observatory in Haiwaii, made available by the Scripps CO2 Program at scrippsco2.ucsd.edu. We wanted to figure out if the CO2 data appears to be impacted by the following events:

1. the OPEC oil embargo which began in October 1973;
2. the global economic recessions around 1980-1982;
3. the fall of the Berlin wall almost exactly 30 years ago, preceding a dramatic fall inindustrial production in the Soviet Union and Eastern Europe;
4. China joining the WTO on 11 December 2001, which was followed by rapid growth in industrial production;
5. the bankruptcy of Lehman Brothers on 15 September 2008, regarded as the symbolic start of the most recent global financial crisis
6. the signing of the Paris Agreement on 12 December 2015, intended to limit CO2 emissions.

Model

By ploting the concentration of CO2 based on time, we saw cycles along a upward trend (Figure1 and Figure2),so we chose $cos(2\pi x_i) + sin(2\pi x_i) + cos(4\pi x_i) + sin(4\pi x_i)$ as part of the model, where $cos(2\pi x_i) + sin(2\pi x_i)$ represents the annually fluctuations and $cos(4\pi x_i) + sin(4\pi x_i)$ represents the semiannually fluctuations. We tried the linear model first,which was $E(CO_2) = X_i\beta$, but it did not fit the data well by looking at Figure3 and Figure4. Then we tried the semi-parametric model with the response variable CO2 follows Gamma distribution (by checking Figure5). The fixed part $X_i\beta$ is $cos(2\pi x_i) + sin(2\pi x_i) + cos(4\pi x_i) + sin(4\pi x_i)$, the smoothing part is U(t1)= f(days). The model is :

$Y_i \sim \tau(\theta)$
$log(E(Y)) = X_i\beta + U(t_i) + V_i$
$V_i \sim N(0, \sigma_v^2)$
Y_i are responses
$U(t)$ is a second-order random work
V_i is independent variation

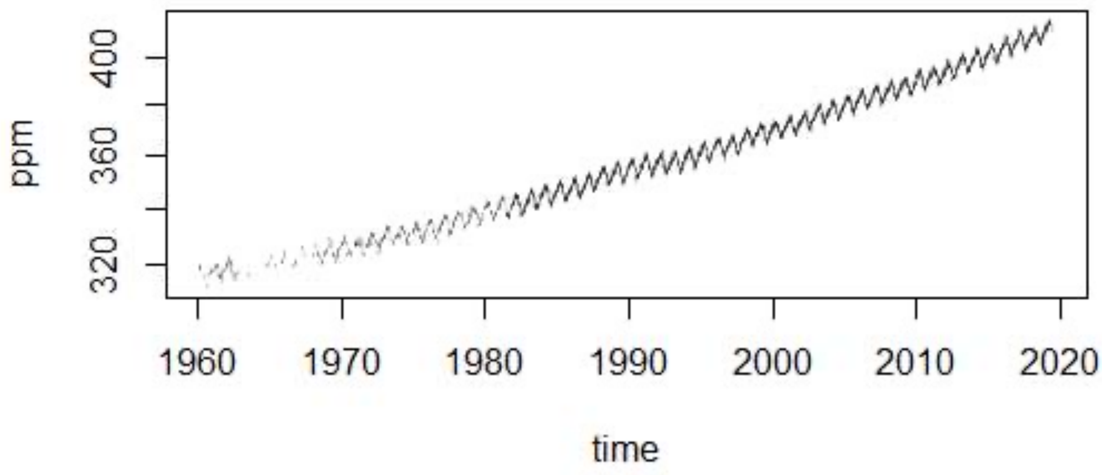


Figure 1

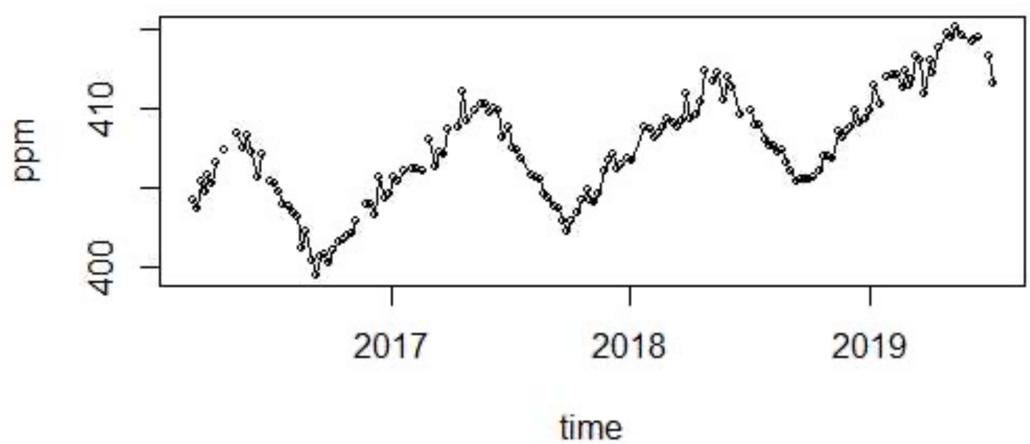


Figure 2

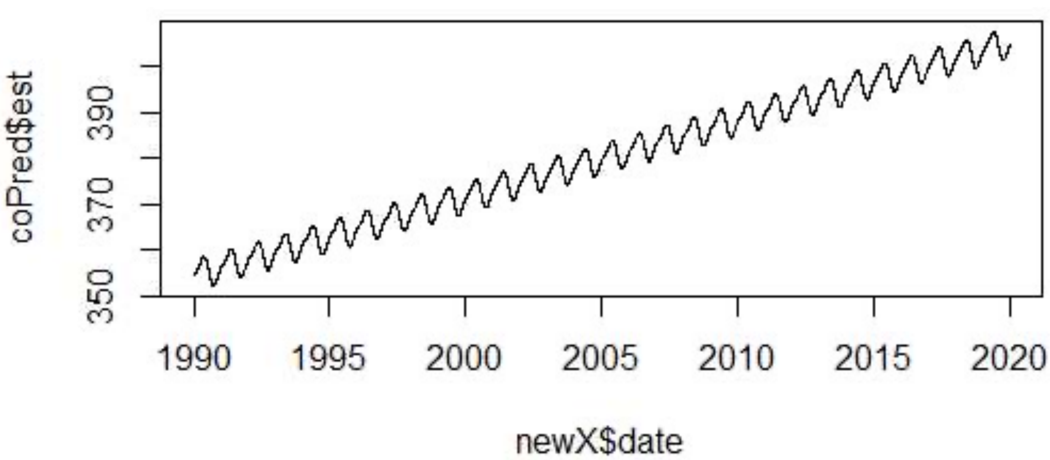


Figure 3

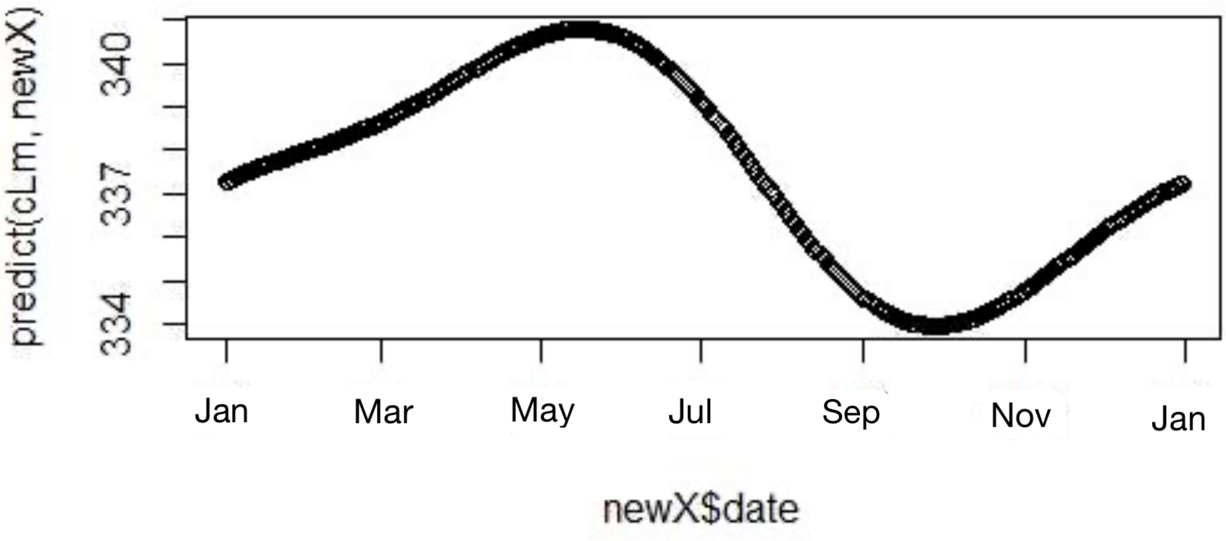


Figure 4

Jan

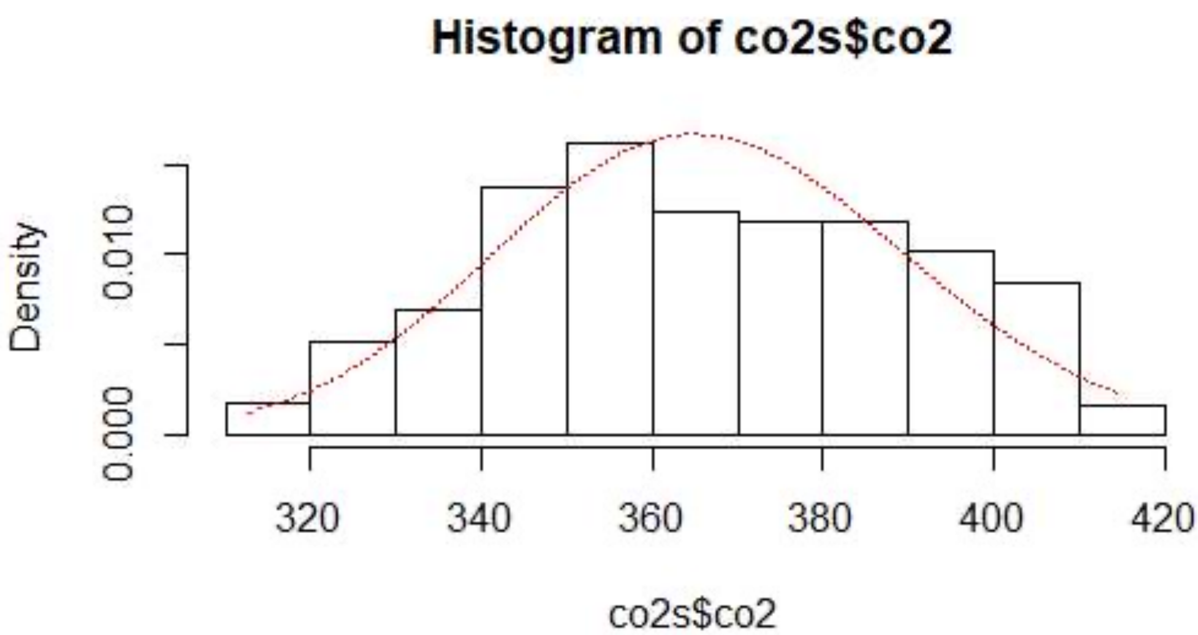


Figure 5

We plotted the slope of the trend (FIGURE6) and the derivative(Figure 7) of the fitted semi-parametric model and found that the general trend of the density of CO_2 is upward and with an increasing rate. And the events had influences on the changing rate of the density of CO_2.We've marked the events on FIGURE7 by different colours.

Analysis

We plotted the slope of the trend (FIGURE6) and the derivative(Figure 7) of the fitted semi-parametric model and found that the general trend of the density of CO2 is upward and with an increasing rate. And the events had influences on the changing rate of the density of CO2.We've marked the events on FIGURE7 by different colours.

event1: The OPEC oil embargo which began in October 1973 caused the growth rate of CO2 to decrease

event2:The growth of the density of CO2 decelearated immediately after the global economic recessions starated in 1980 and gradually accelerated after 1982.

event3: The growth of the density of CO2 decelerated for more than two years after the fall of the Berlin wall, due to the dramatic fall in industrial production in the Soviet Union and Eastern Europe.

event4: The growth rate of the density of the CO2 increased immediately after China joining the WTO on 11 December 2001, which was followed by rapid growth in industrial production, and the accleration of the growth of the density of CO2 lasted for around one year.

event5: The deceleration of the growth of the density of CO2 only lasted for a very short time after the bankruptcy of Lehman Brothers on 15 September 2008, then the growth of the density of CO2 accelerated back to the level before the economic ression.

event6: the signing of the Paris Agreement on 12 December 2015, intended to limit CO2 emissions, had a large influence on the control of CO2 emissions. The growth rate of the density of CO2 decreased immediately. The decelaration of the growth rate lasted for around two years and the rebound did not reach the original level.

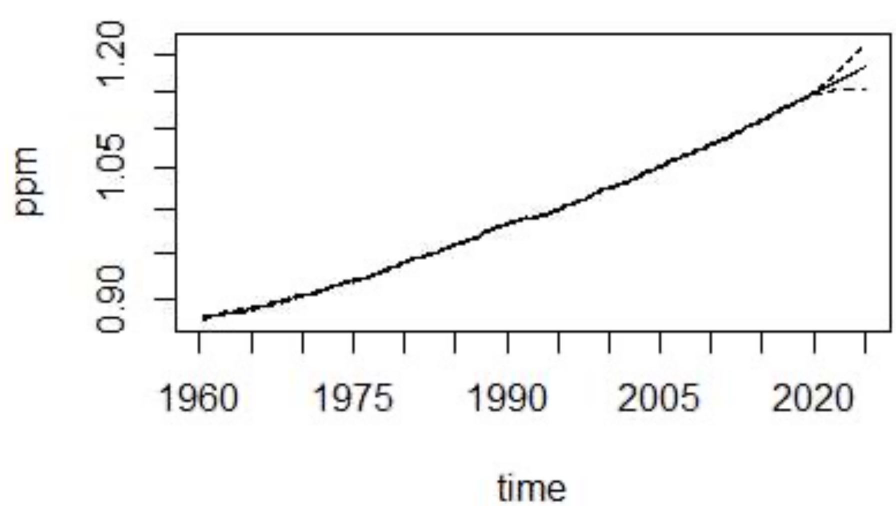
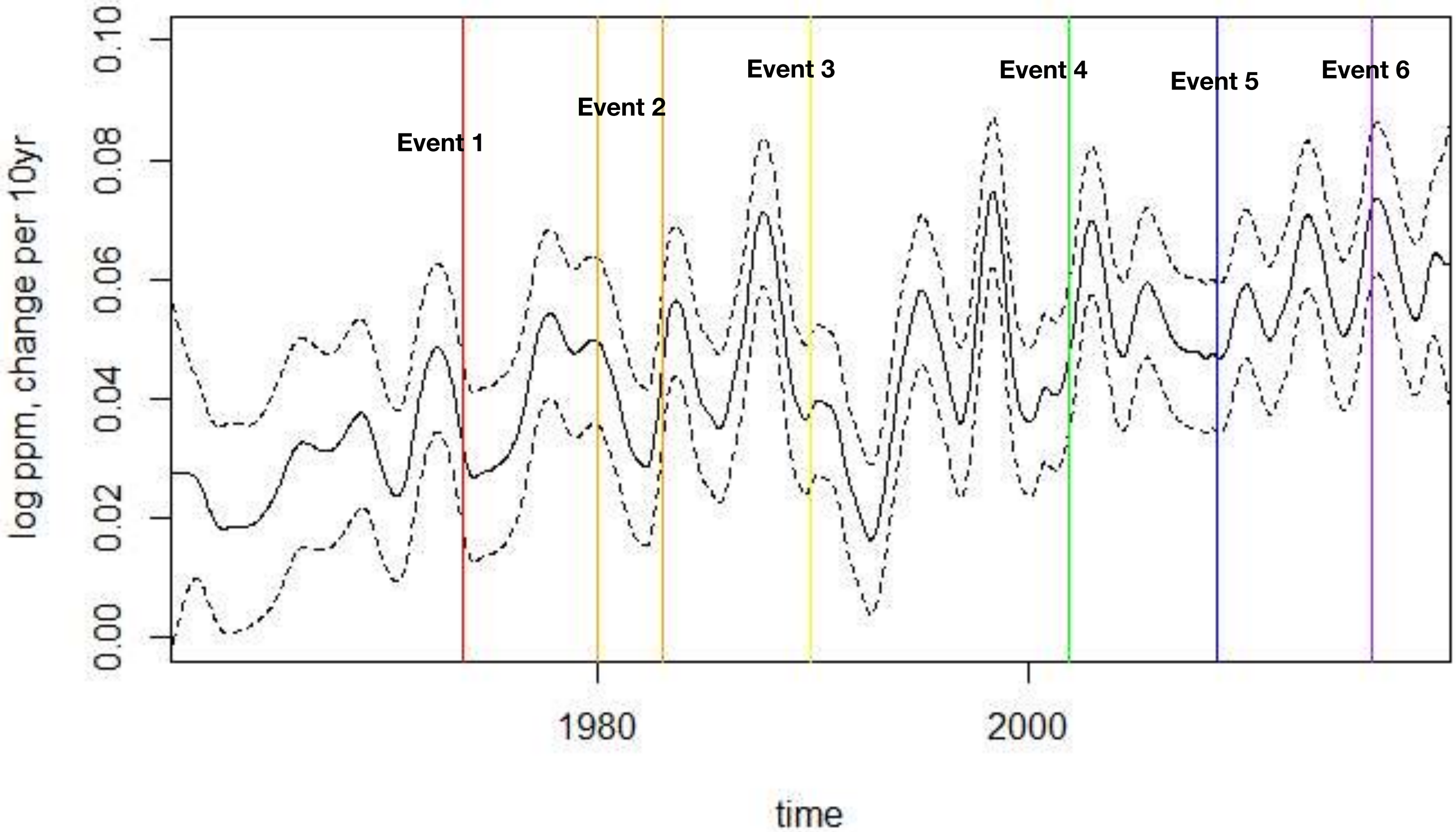


Figure 6



Event 1
Event 2
Event 3
Event 4
Event 5
Event 6
log ppm, change per 10yr
0.00
0.02
0.04
0.06
0.08
0.10
1980
2000
time

Figure 7

Prediction

By checking FIGURE6 and FIGURE8, we saw the the density of the CO2 will continue increasing in the future, and the seasonal fluctuations will continue to happen.

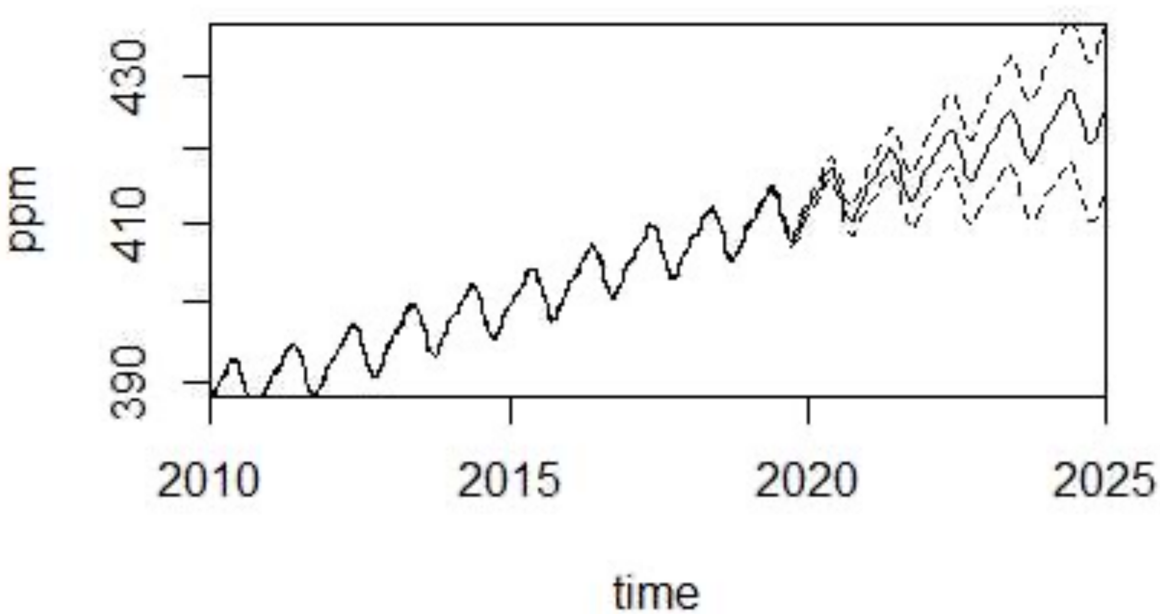


Figure 8

Question 2

Introduction

We analyzed the daily maximun temperature data recorded on Sable Island and tried to figure out whether the data from Sable island is broadly suppotive of the statement from the IPCC - "Human activities are estimated to have cause dapproximately 1.0°Cof global warming above pre industrial 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. (high confidence)"

Method

Figure1 plots the daily maximun temperature data according to time. Figure2 shows the data in the period from 2016 to the present, with summer months (May to October inclusive) in black and winter in red. Because the winter temperatures are more variable than summer temperatures, we considered only summer temperatures when modelling.We found the response variable - daily maximun temperature followed T distribution by plotting the histogram (Figure 3). We used semi-parametric model to fit the data. The parametric part $X_i\beta$ is $cos(2\pi x_i) + sin(2\pi x_i) + cos(4\pi x_i) + sin(4\pi x_i)$. The nonparametric part is U(t1) + U(t2) + U(t3) = f(week) + f(weekIid) + f(yearFac) where f(week)is the random slope, f(weekIid) and f(yearFac) are the random intercepts.V_i represent the noise
We used the Bayes inferance to smooth the nonparametric part. We used penalized complexity prior as our prior distribution for $\sigma_{U(t1_i)}$, and we chose the parameters (0.1/(52*100), 0.05) because we thought there was 5% probability that the deviation of this second-order random walk > 1.9×10^{-5} $(P(\sigma_{U(t1_i)} > 1.9 \times 10^{-5}) = 0.05)$. We also used penalized complexity prior as our prior distribution for $\sigma_{U(t2_i)}$ and $\sigma_{U(t3_i)}$, then adjusted both by using parameters (1, 0.5) because we thought there was 50% probability the deviation of these two RW(0)s > 1. We used the data Y, to adjusted the priors of $\sigma_{U(t1_i)}$,$\sigma_{U(t2_i)}$ and $\sigma_{U(t3_i)}$, and got the posteriors , which are $[\sigma_{U(t1_i)}|Y]$,$[\sigma_{U(t2_i)}|Y]$, and $[\sigma_{U(t3_i)}|Y]$
We also used the posteriors of the deviations of degree of freedom and V_i to estimate the σ_{V_i} and σ_{dof}. We used penalized complexity prior as prior distributions and with parameters (10, 0.5) and (1, 0.5)respectively, which means we thought $P(\sigma_{dof} > 10) = 0.5$ and $P(\sigma_{V_i} > 1) = 0.5$, and we used data Y to adjusted the priors to get the posteriors.

Model: $\sqrt{s\tau}(y-\eta) \sim T_v$
$E(Y_i) = X_i\beta + U(t1_i) + U(t2_i) + U(t3_i) + V_i$
$V_i \sim N(0, \sigma_v^2)$ Y_i are responses
τ:is the precision parameter
s: is a fixed scaling s>0
η: is the linear predictor
T_v: is a reparameterized standard Student-t with v > 2 degrees of freedom with unit variance for all values of v.
V_i is the independent variation

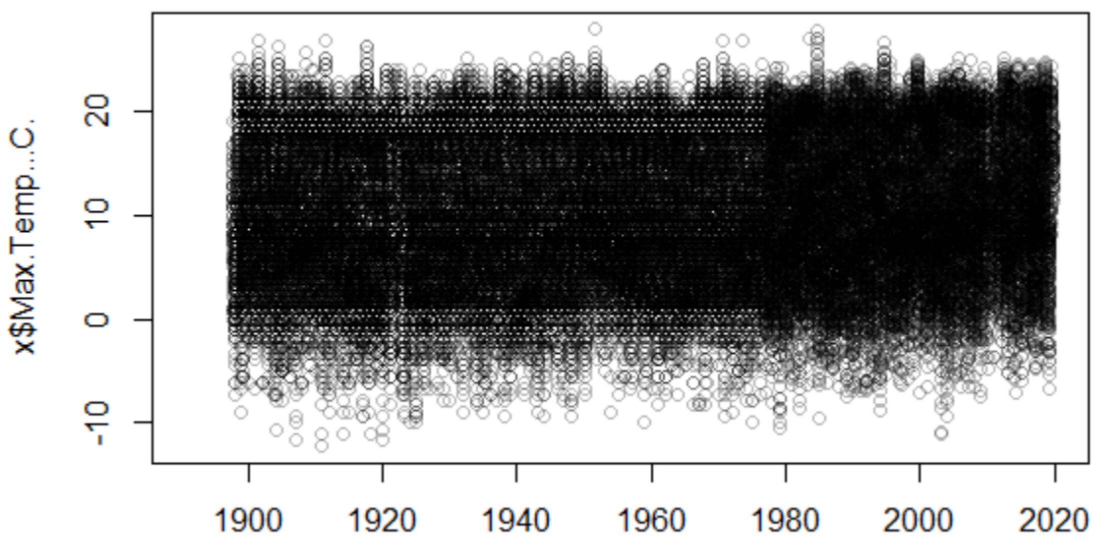


Figure 1

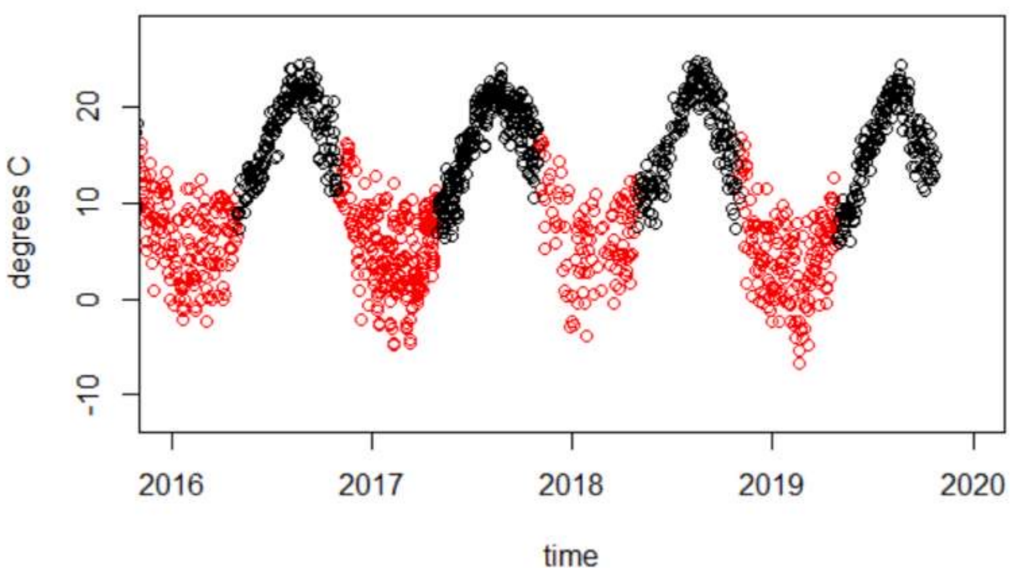
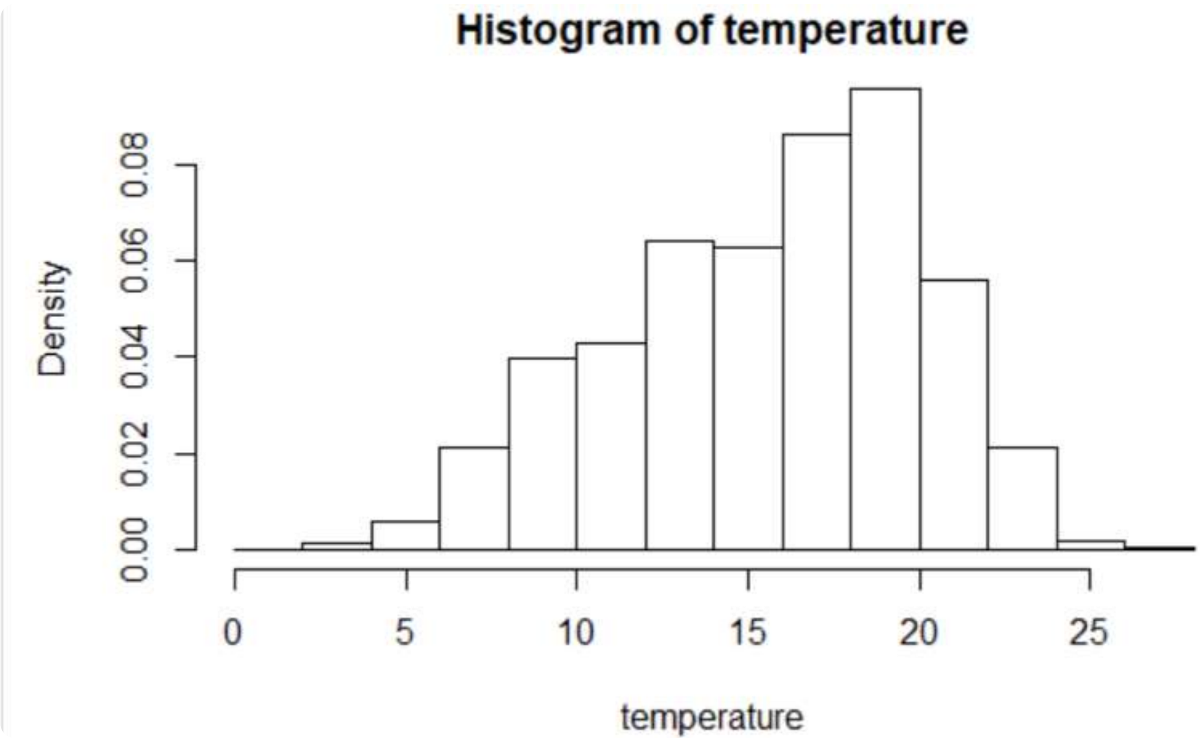


Figure 2


Histogram of temperature
Density
0.00
0.02
0.04
0.06
0.08
0
5
10
15
20
25
temperature
Figure 3

Analysis

We plotted the week random effect according to time to checked the trend of the temperature by FIGURE4 and found the global temperature has increased approximately 1.0˚C from 1900 till now with a likely range of 0.8˚C to 1.2˚C by checking the 95% credible interval of the week random effect. The credible interval becomes wider as we look at the future becuse the furthur a day from today, the less the tempeture in that day is correlated with today's. We could not make good predictions of what would happen in 2030 and 2052. The expect of the temperature keeps increasing, but we were uncertain about whether or not the global warming is likely to reach 1.5˚C between 2032 and 2052 since the 95% credible interval is very wide. Figure5 shows the posterior samples of the week random effect, and we could see from this figure that the temperature may increase or decrease in the future. In conclusion, the data from Sable island is suppotive of the statement of "Human activities are estimated to have cause dapproximately 1.0°Cof global warming above pre industrial levels, with a likely range of 0.8°C to 1.2°C.", but not supportive of the statement" Global warming is likely to reach 1.5°C between 2030 and 2052 if it continues to increase at the current rate. (high confidence)"

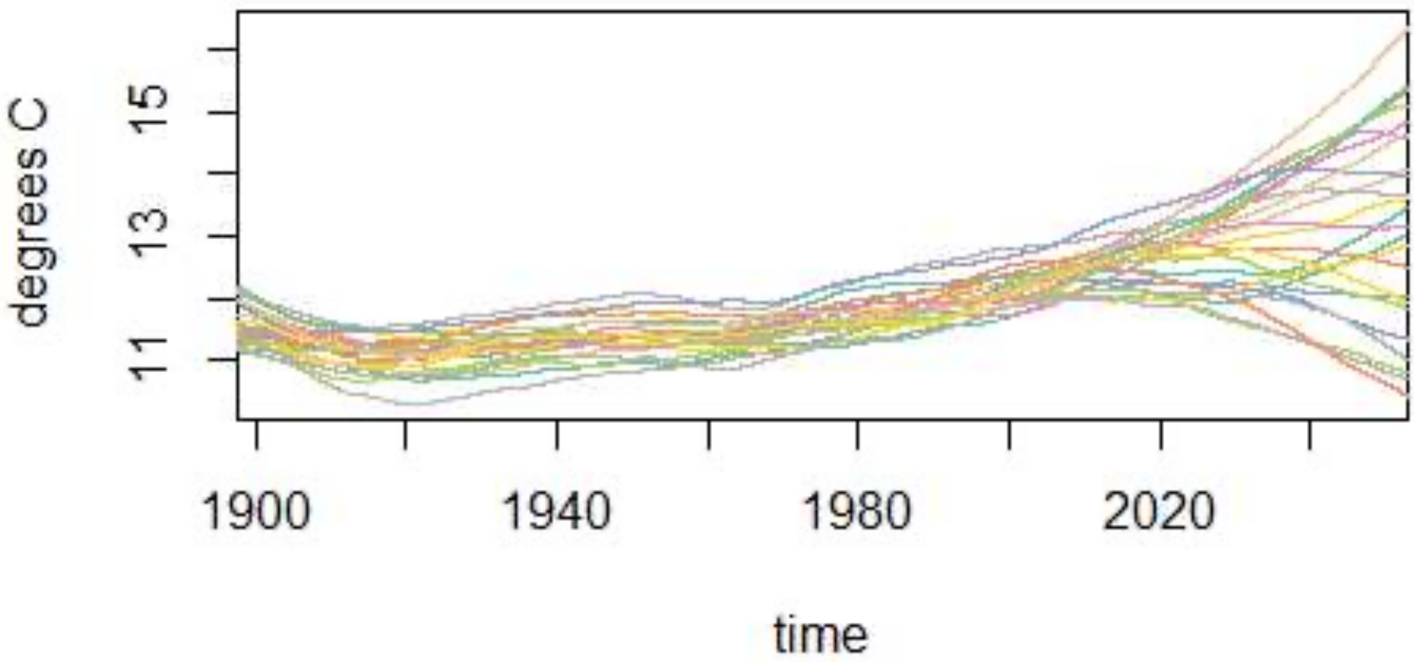


Figure 5

Appendix

```
co2s = read.csv("C:/Users/Jessica Jiang/Downloads/daily_flask_co2_mlo.csv",header = FALSE, sep = ",",
                  skip = 69, stringsAsFactors = FALSE, col.names = c("day","time", "junk1","junk2","Nfla
#if (!file.exists(cFile)) download.file(cUrl, cFile)

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
#qqnorm(co2s$co2)
plot(co2s$date, co2s$co2, log = "y", cex = 0.3, col = "#00000040",
     xlab = "time", ylab = "ppm")
plot(co2s[co2s$date > ISOdate(2016, 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)
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$date), coPred[, c("lower",
                                           "upper", "est")], lty = 1, col = c("yellow", "yellow",
                                                                            "black"))
newX = newX[1:365, ]
newX$days = 0
plot(newX$date, predict(cLm, newX))

m <- na.omit(co2s$co2)
param <- MASS::fitdistr(m, "gamma")
```

```r
x <- seq(min(m), max(m), length.out = 100)
hist(co2s$co2,probability = TRUE)
lines(sort(co2s$co2), dgamma(sort(m), shape = param$estimate[1],rate = param$estimate[2]),
      col = "red", lty = "dotted")

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 * StimePre

                                 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())

# dont run this code
co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 +
                f(timeRw2, model = 'rw2',
                  values = StimeIndex,
                  prior='pc.prec', param = c(log(1.01)/26, 0.5)),
```

```r
              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
        col = "black", lty = c(1, 2, 2), log = "y", xaxt = "n",
        xlab = "time", ylab = "ppm")
xax = pretty(timePoints,10)
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$date)),
        xaxs = "i", xaxt = "n",
        xlab = "time",
        ylab = "log ppm, change per 10yr")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2008, 1, 1, tz = "UTC"), col = "blue")
matplot(StimeSeason, exp(co2res$summary.lincomb.derived
                         [grep("season",rownames(co2res$summary.lincomb.derived)),
                          c("0.5quant","0.025quant", "0.975quant")]),
        type = "l", col = "black",
        lty = c(1, 2, 2), log = "y", xaxs = "i", xaxt = "n",
        xlab = "time", ylab = "relative ppm")
xaxSeason = seq(ISOdate(2009, 9, 1, tz = "UTC"), by = "2 months",
               len = 20)
axis(1, xaxSeason, format(xaxSeason, "%b"))
timePred = co2res$summary.lincomb.derived[grep("pred",
                                        rownames(co2res$summary.lincomb.derived)), c("0.5quant",
                                                                                   "0.025quant
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")
xaxPred = seq(ISOdate(2010, 1, 1, tz = "UTC"), by = "5 years",
              len = 20)
axis(1, xaxPred, format(xaxPred, "%Y"))



###########################

## Question2 Heat

heatUrl = "http://pbrown.ca/teaching/appliedstats/data/sableIsland.rds"
```

```r
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$')
library("Matrix")
library("sp")
library("parallel")
library("INLA")
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())

### dont run this line
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),
```

```r
                      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$Max.Temp...C., col = mapmisc::col2html("black",
                                                       0.3))
forAxis = ISOdate(2016:2020, 1, 1, tz = "UTC")
plot(x$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"))
```