Question 1: CO2
This report discusses if the CO2 data from an observatory in Hawaii is impacted by several historical events. A generalized additive model is fitted to model the atmospheric Carbon Dioxide concentration. This is because although the atmospheric Carbon Dioxide concentration increases with time, the relationship between the two variables is highly nonlinear, as we can see from Figure 1.

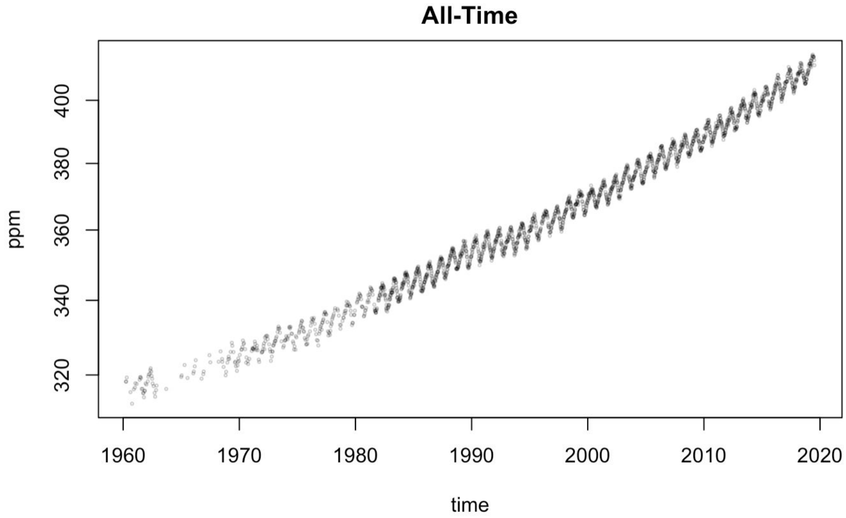


Figure 1. Sequence plot for ppm versus time

In this model, it is believed that each CO2 data record follows a gamma distribution, as we can infer from Figure 2. The response variable CO2 concentration is modelled as a function of some B-spline fits sin12,cos12,sin6, cos6, where sin12 and cos12 reflect yearly fluctuation and sin6 and cos3 reflect biyearly fluctuation, and the nonlinear effects variable timeRw2, which is a second-order random walk of time.

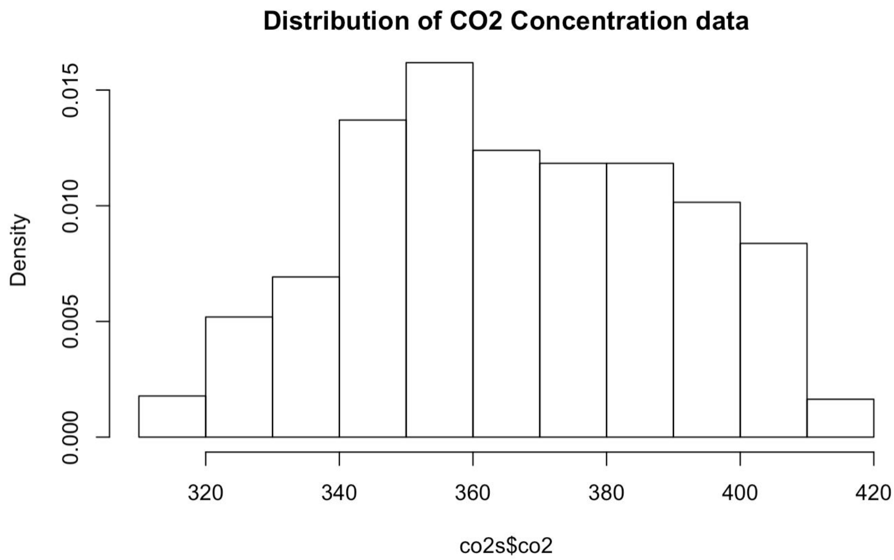


Figure 2. Histogram showing the distribution of CO2 concentration data

Using this model, the impact of the following events on the CO_2 data will be assessed:

- OPEC Oil Embargo (October 1973)

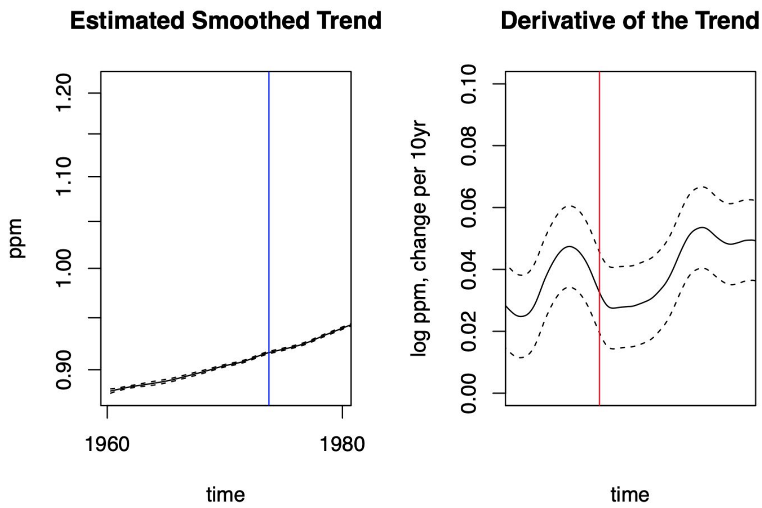


Figure 3. Estimated Smoothed Trend and Derivative of the Trend for Period 1960-80

In Figure 3, the estimated smoothed trend and derivative of the trend are shown. The blue and red lines indicate the time of October 1973. From the trends, it can be seen the CO2 concentrations increases but at a decreasing rate after the OPEC Oil Embargo, it appears that the trend becomes shallower. However, the rate did not sustain for a long period, hence we can say that this event might have reduced CO2 emission to a small degree for that period.

- Global economic recession around 1980-82

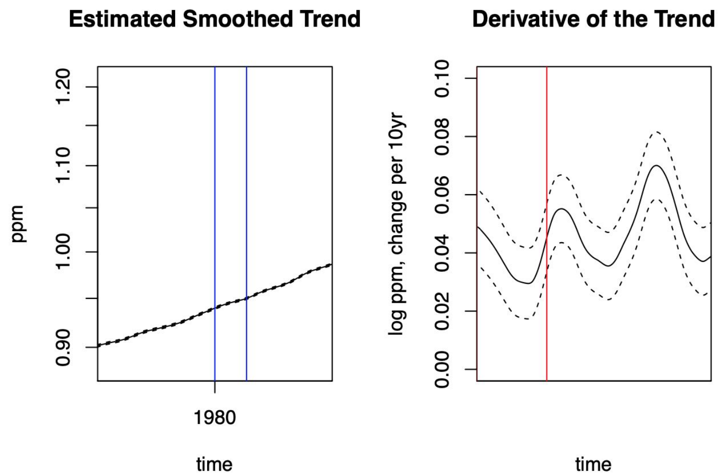


Figure 4. Estimated Smoothed Trend and Derivative of the Trend for Period 1970-90

In Figure 4, the blue and red lines indicate the period from 1980-82. From the trends, it can be seen the CO2 concentrations increases at a decreasing rate after 1980 and that rate sustains for a short period before eventually recovers to the same rate at around 1982, it appears that the trend has not changed drastically. Hence we can say that this event did have to reduce CO2 to a small degree for that period.

- The fall of the Berlin wall (9 November 1989)

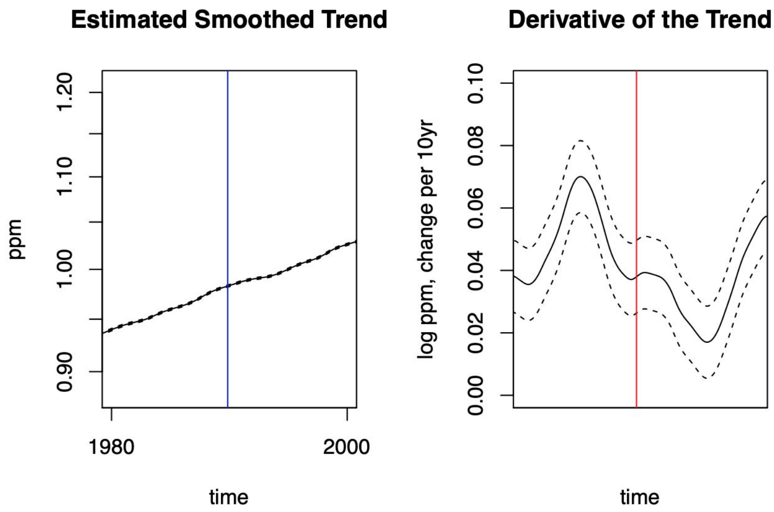


Figure 5. Estimated Smoothed Trend and Derivative of the Trend for Period 1980-2000

In Figure 5, the blue and red lines indicate the date of 9 November 1989. From the trends, it can be seen the CO2 concentrations increases at a decreasing rate after the event date, and the decreasing rate is sustained for a long period of time, as the trend becomes steeper. Hence we can say that this event did have a huge impact on the reduction of CO2 emission for that period.

- China joining the WTO (11 December 2001)

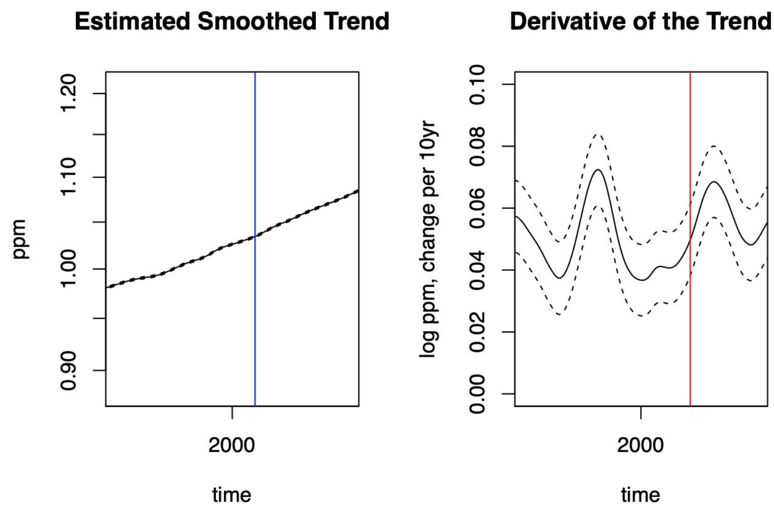


Figure 6. Estimated Smoothed Trend and Derivative of the Trend for Period 1990-2010

In Figure 6, the blue and red lines indicate the date of 11 December 2001. From the trends, it can be seen the CO2 concentrations increases at an increasing rate after the event, and the increasing rate is sustained for a short period of time before decreasing rate sets in again. Hence we can say that this event did have an impact on the increase in CO2 emission for that period.

- The bankruptcy of Lehman Brother (15 September 2008)

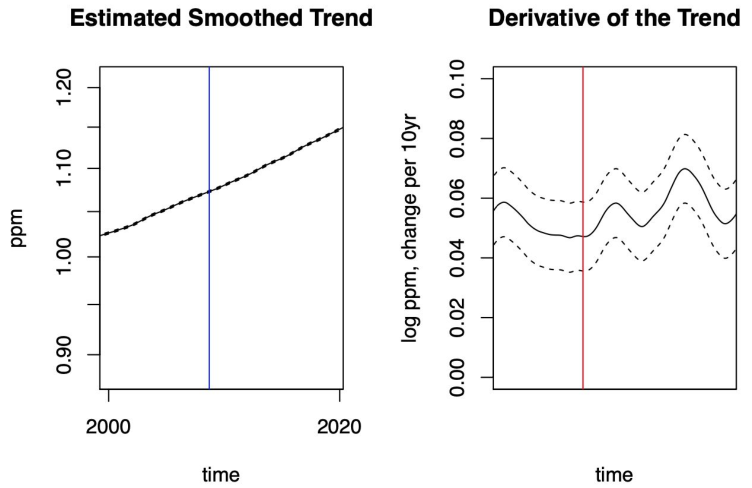


Figure 7. Estimated Smoothed Trend and Derivative of the Trend for Period 1990-2010

In Figure 7, the blue and red lines indicate the date of 15 September 2008. From the trends, it can be seen the CO2 concentrations increases at an increasing rate after the event, and the increasing rate is sustained for a long period of time, as the trend becomes steeper over time. Hence we can say that this event did have an impact on the increase in CO2 emission for that period.

- The signing of the Paris Agreement (12 December 2015)

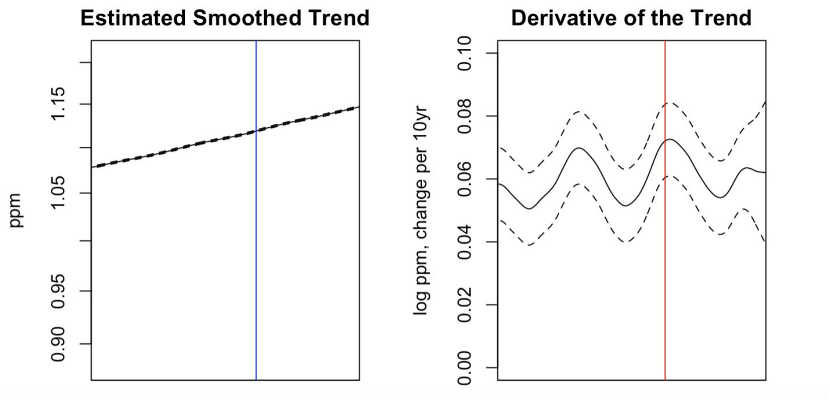


Figure 8. Estimated Smoothed Trend and Derivative of the Trend for Period 2010-30

In Figure 8, the blue and red lines indicate the date of 12 December 2015. From the trends, it can be seen the CO2 concentrations increases at a decreasing rate after the event, and the decreasing rate is sustained for a long period of time and it is even expected to sustain in the predicted range of data. Hence we can say that this event did have an impact on the reduction in CO2 emission for that period.

Question 2: Heat

This report discusses whether the data from Sable Island is broadly supportive of the statement from IPCC, stating 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. And global warming is likely to reach 1.5°C between 2030 and 2052 if it continues to increase at the current rate. A generalized additive model is fitted to model the daily maximum temperature data.

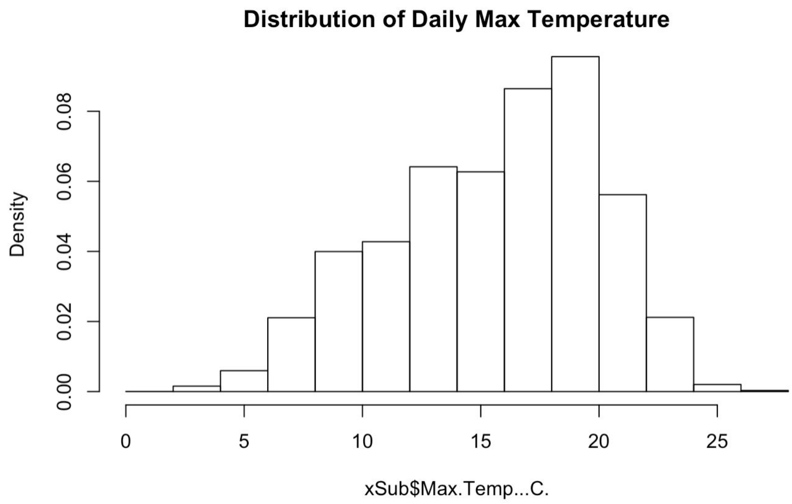


Figure 1. Histogram showing the distribution of daily maximum temperature

In this model, it is believed that each temperature data record follows a student T distribution, as we can infer from Figure 1, it is heavy-tailed. The response variable daily maximum temperature is modelled as a function of some B-spline fits sin12,cos12,sin6, cos6, where sin12 and cos12 reflect yearly fluctuation and sin6 and cos3 reflect biyearly fluctuation. It also includes “week” which is treated as a random slope, “weekIid” and “yearFac” which are treated as random intercepts. For the variable “week”, the PC prior to the precision has a parameter (1,0.5). For the variables “weekIid” and “yearFac”, the PC prior to the precision also has a parameter (1,0.5). Whereas for the entire model, the PC prior to the precision has a parameter (1,0.5). The dof-prior has a tabulated density and is the PC prior to the degree of freedom parameter in the Student-t distribution. The parameter to this prior in this model is (10,0.5).

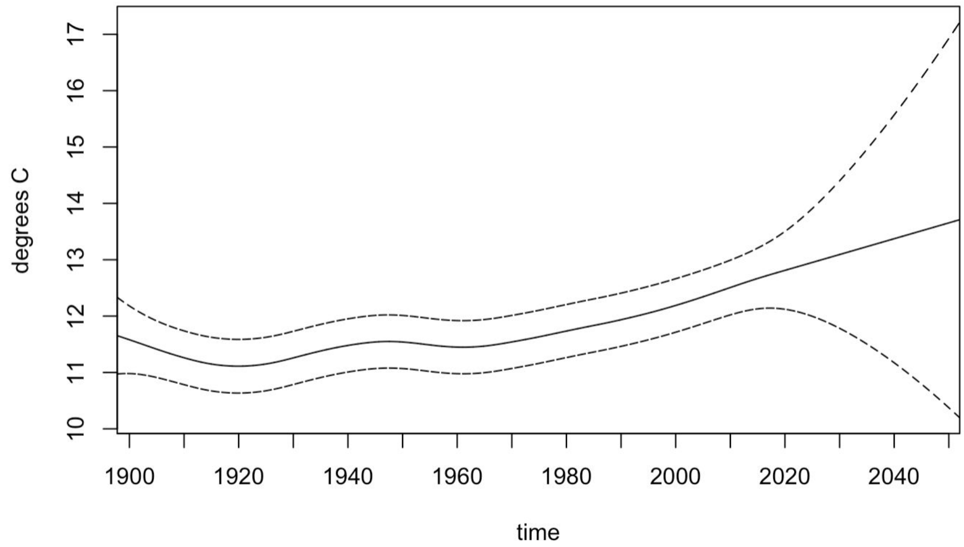


Figure 2. Estimated time trend of daily maximum temperature

Figure 2 is an estimated time trend of daily maximum temperature, from the period of 1880 to 2053. From the figure, there is an increasing trend in the temperature.

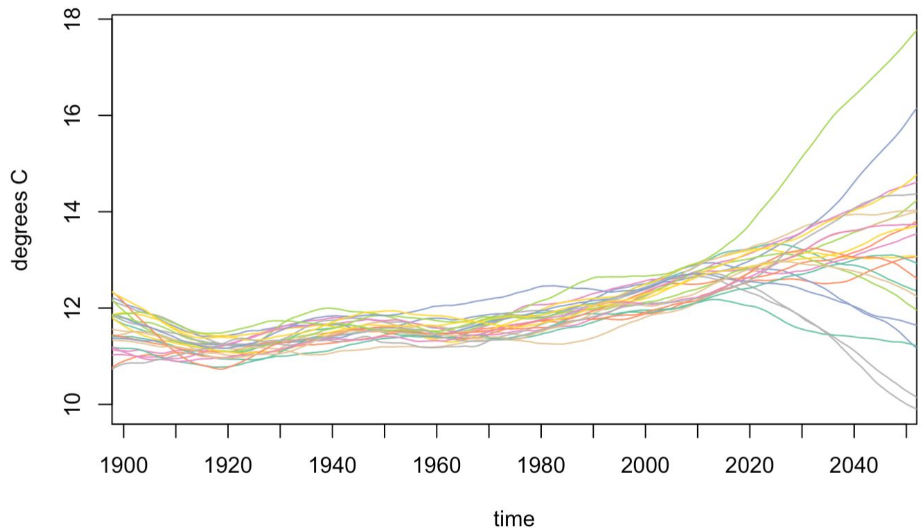


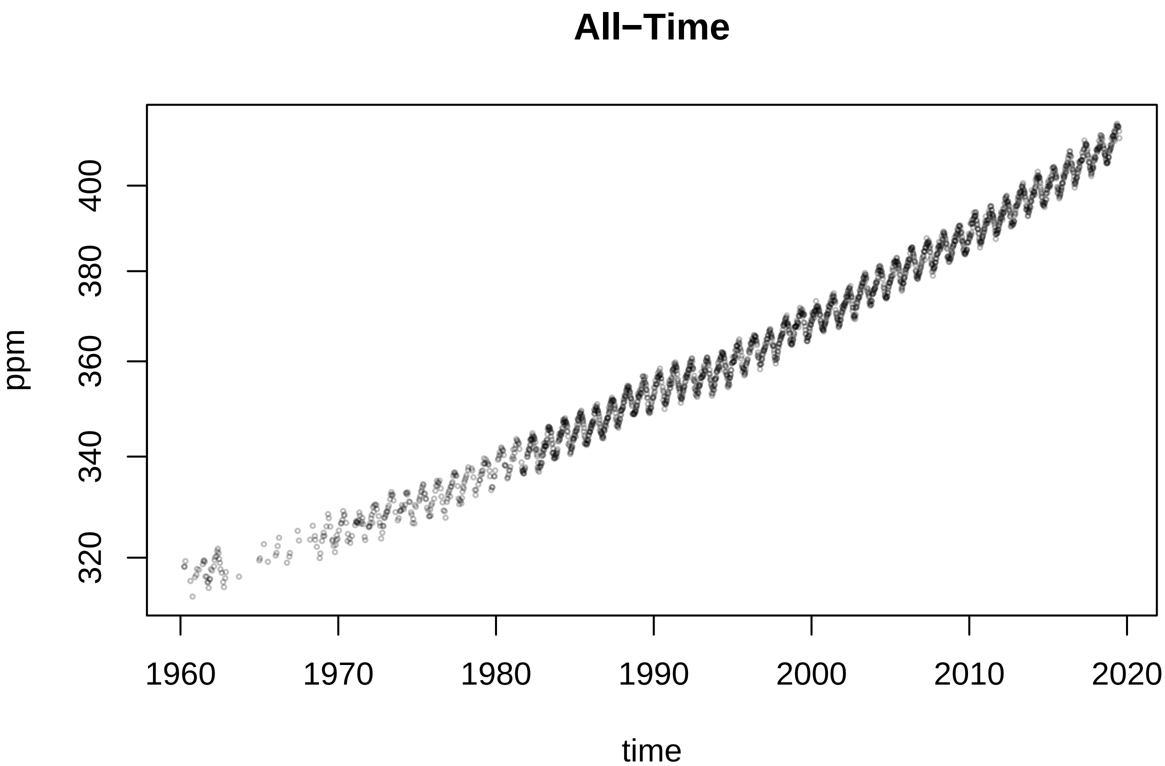
Figure 3. Posterior samples of the estimated time trend

To further prove this claim, looking at Figure 3, which consists of the posterior samples of the trend in Figure 2, most of the posterior samples have increasing trend. With such trends, it is impossible to deny that the global temperature is getting higher, hence it is highly possible that global warming is likely to reach 1.5°C between 2030 and 2052.

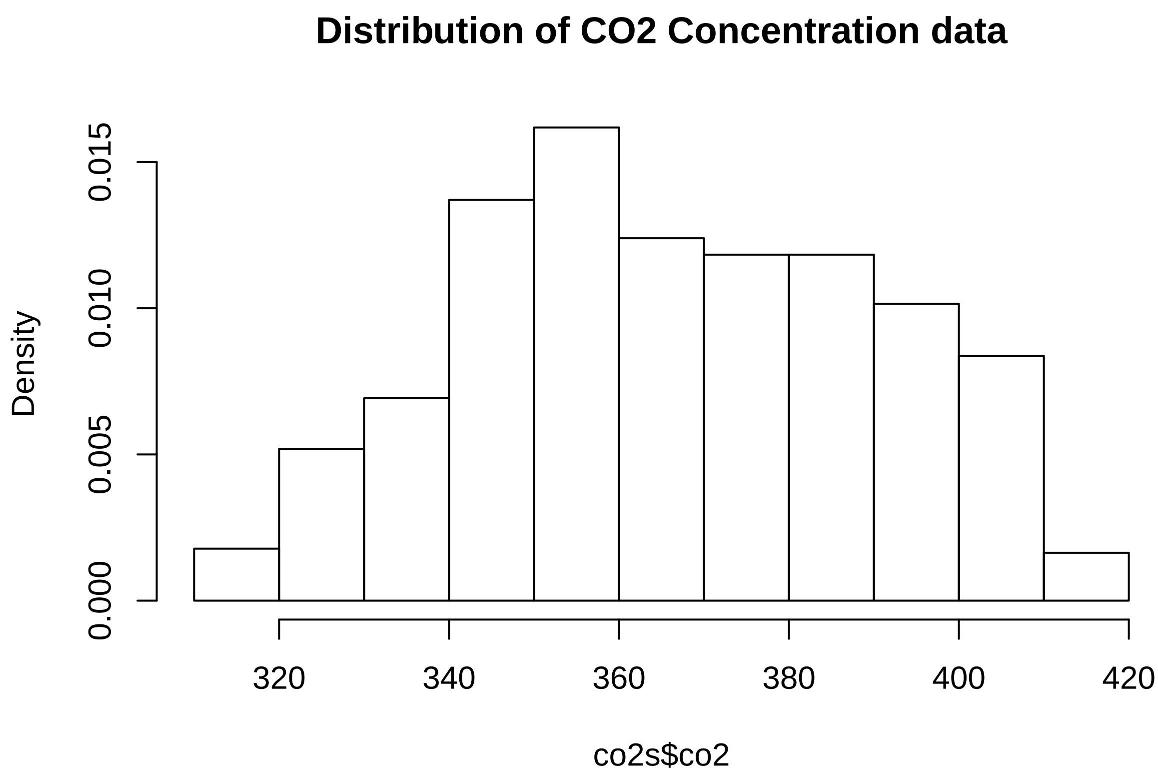
STA442 HW3 Question 1

CO2

```r
co2s = read.csv("co2.csv")
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

# Figure 1: Sequence plot for all-time
plot(co2s$date, co2s$co2, log='y', cex=0.3, col='#00000040', xlab='time', ylab='ppm',main ="All-Time")
```




```r
# Figure 2: Histogram
hist(co2s$co2,prob=TRUE,main="Distribution of CO2 Concentration data")
```




```r
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)

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)
```

```r
library("INLA")
```

```
## Loading required package: Matrix

## Loading required package: sp

## Loading required package: parallel

## This is INLA_19.09.03 built 2019-09-03 09:07:31 UTC.
```

```
## See www.r-inla.org/contact-us for how to get help.
## To enable PARDISO sparse library; see inla.pardiso()
```

```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)))
```

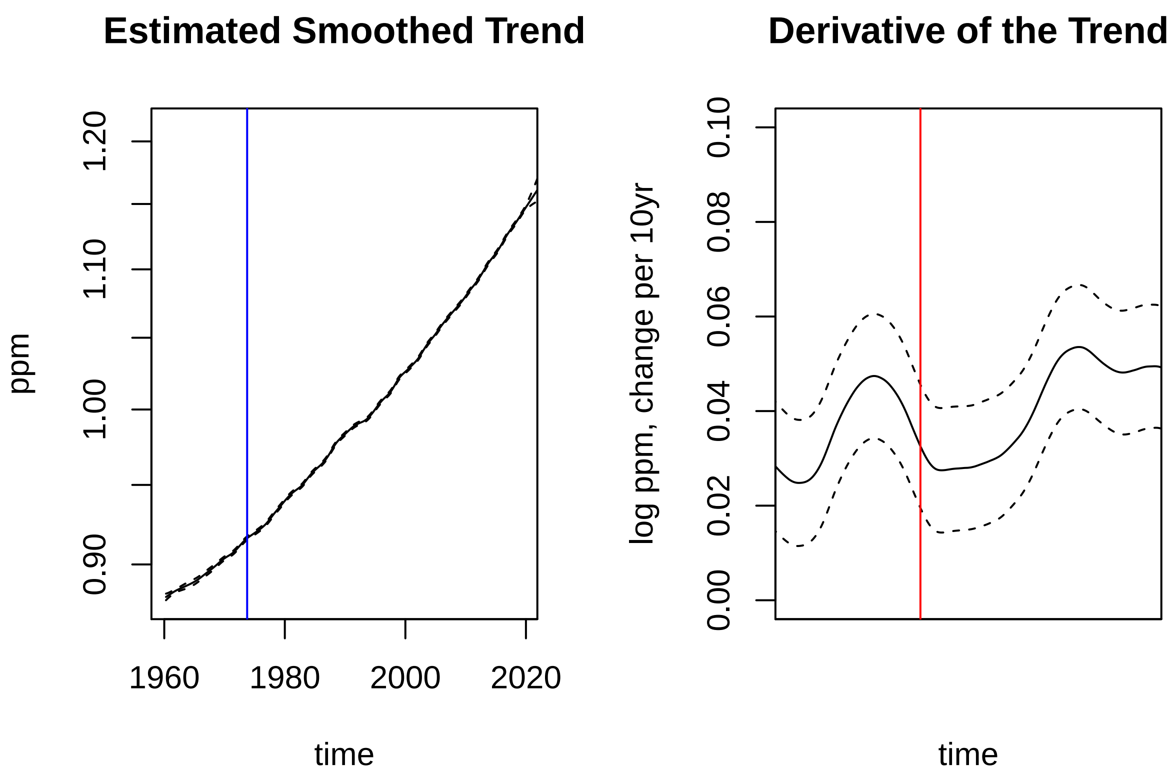
```r
# disable some error checking in INLA
library("INLA")
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())

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)))), verbose=TRUE)
```

```r
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")))
xaxSeason = seq(ISOdate(2009, 9, 1, tz = "UTC"), by = "2 months", len = 20)
```

```
timePred = co2res$summary.lincomb.derived[grep("pred",rownames(co2res$summary.lincomb.derived)),
                                           c("0.5quant","0.025quant", "0.975quant")]
```

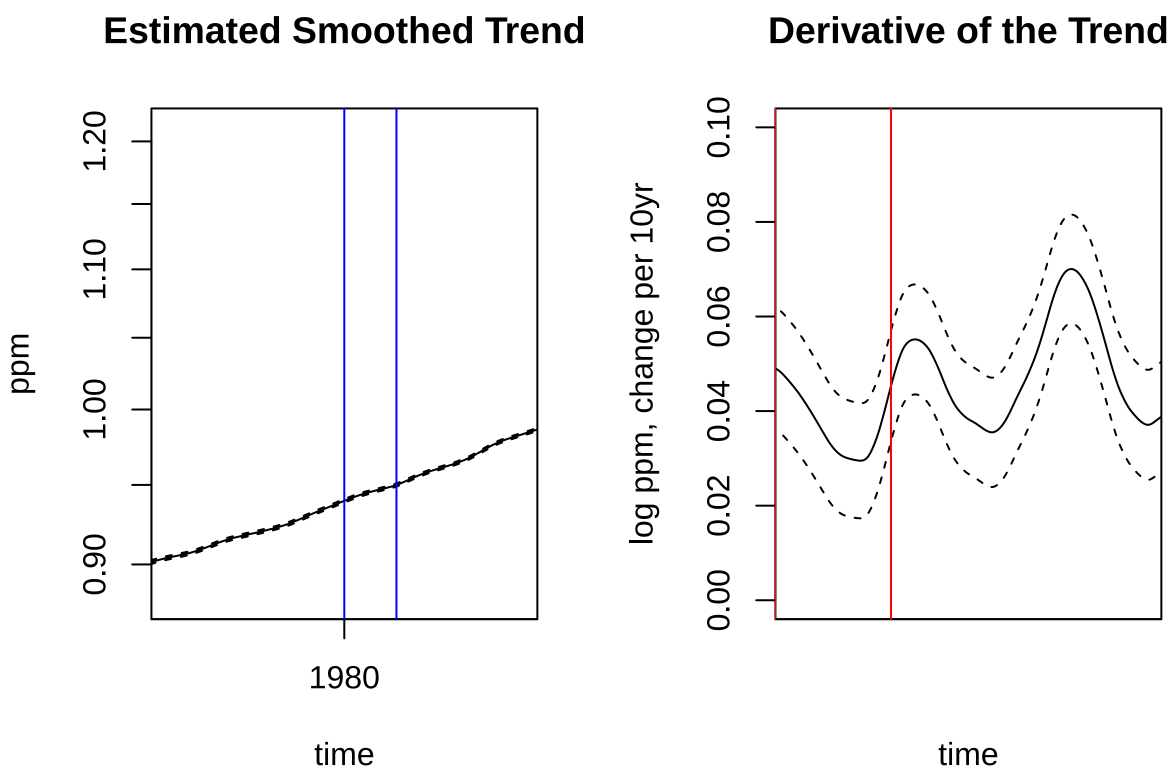
OPEC oil embargo which began in October 1973

```
par(mfrow = c(1,2))
# Sequence
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",
        xlim = range(as.numeric(co2s[co2s$date > ISOdate(1960,1,1, tz='UTC')
                               & co2s$date < ISOdate(2080,1,1, tz='UTC'),
                               c('date')])),main="Estimated Smoothed Trend")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "blue")

# Derivatives
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[co2s$date > ISOdate(1970,1,1, tz='UTC')
                               & co2s$date < ISOdate(1980,1,1, tz='UTC'),
                               c('date')])),
        xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Derivative of the Trend")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
```



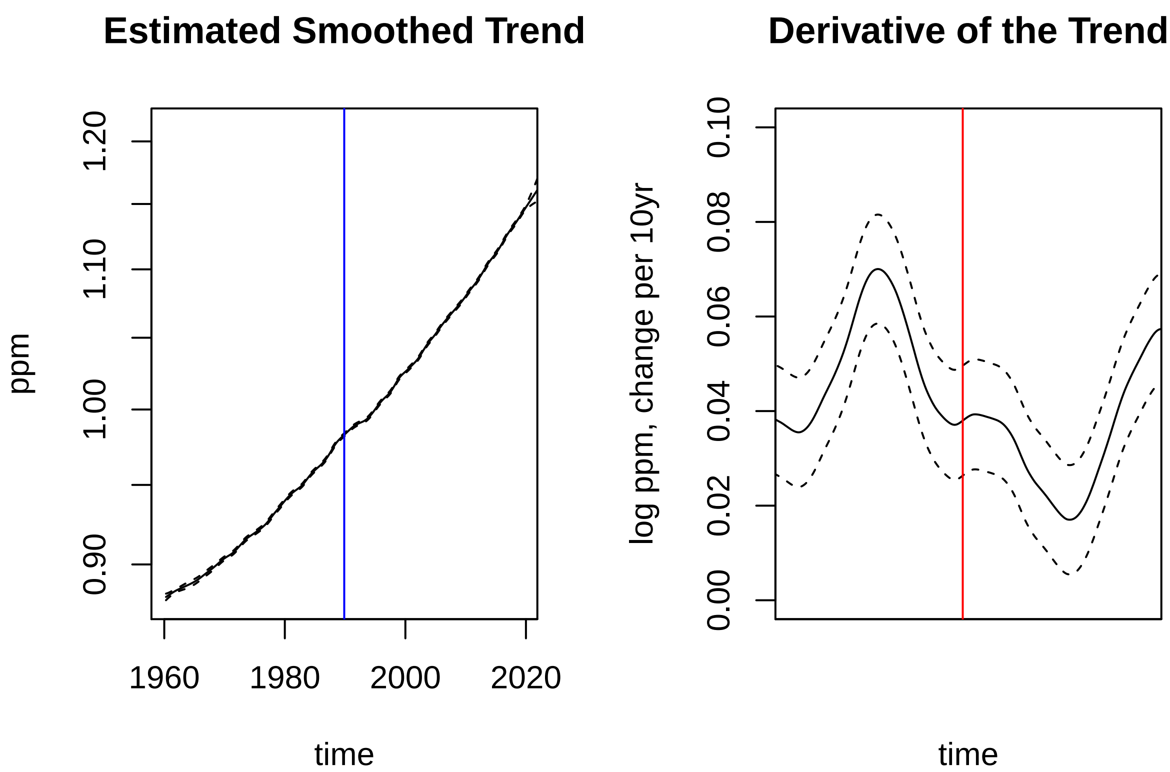
Global economic recessions around 1980-1982

```r
par(mfrow = c(1,2))
# Sequence
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",
        xlim = range(as.numeric(co2s[co2s$date > ISOdate(1970,1,1, tz='UTC')
                                     & co2s$date < ISOdate(1990,1,1, tz='UTC'),
                                     c('date')])),main="Estimated Smoothed Trend")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(1982, 12, 1, tz = "UTC"), col = "blue")

# Derivatives
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[co2s$date > ISOdate(1980,1,1, tz='UTC')
                                     & co2s$date < ISOdate(1990,1,1, tz='UTC'),
                                     c('date')])),
        xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Derivative of the Trend")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(1982, 12, 31, tz = "UTC"), col = "red")
```



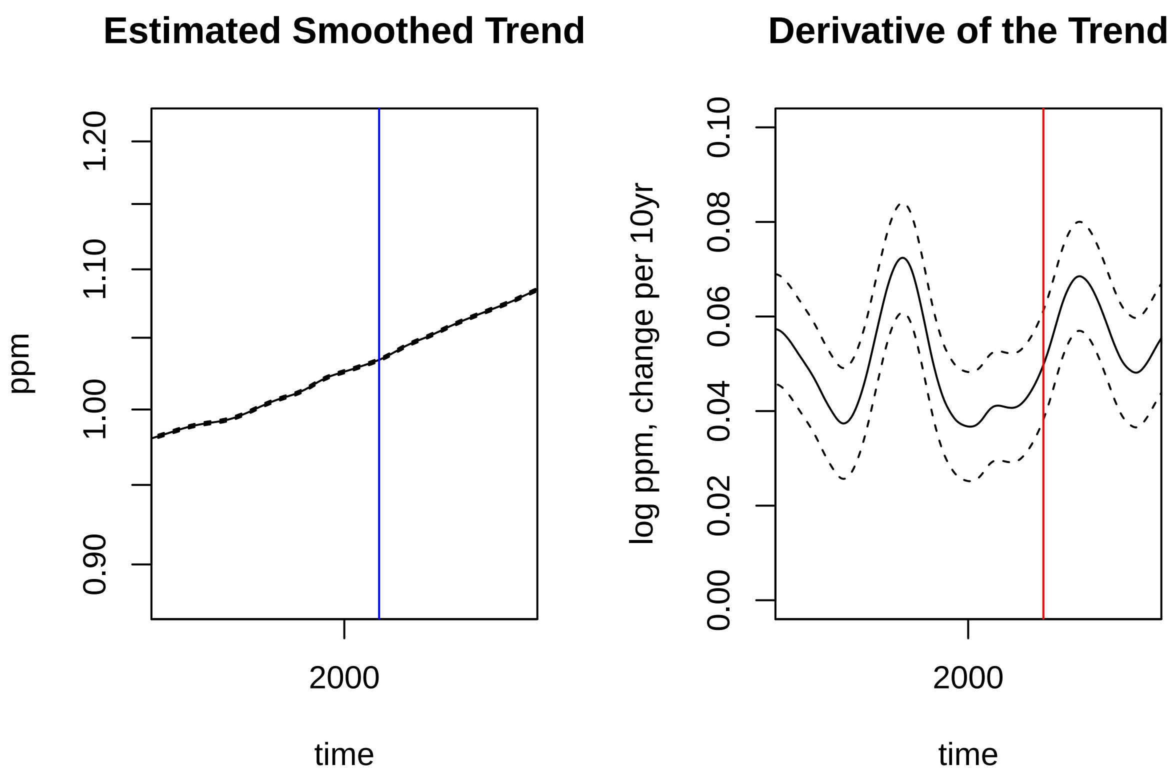
The fall of the Berlin on 9 November 1989

```
par(mfrow = c(1,2))
# Sequence
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",
        xlim = range(as.numeric(co2s[co2s$date > ISOdate(1960,1,1, tz='UTC')
                             & co2s$date < ISOdate(2020,1,1, tz='UTC'),
                             c('date')])),main="Estimated Smoothed Trend")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1989, 11, 9, tz = "UTC"), col = "blue")

# Derivatives
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[co2s$date > ISOdate(1985,1,1, tz='UTC')
                             & co2s$date < ISOdate(1995,1,1, tz='UTC'),
                             c('date')])),
        xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Derivative of the Trend")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(1989, 11, 9, tz = "UTC"), col = "red")
```



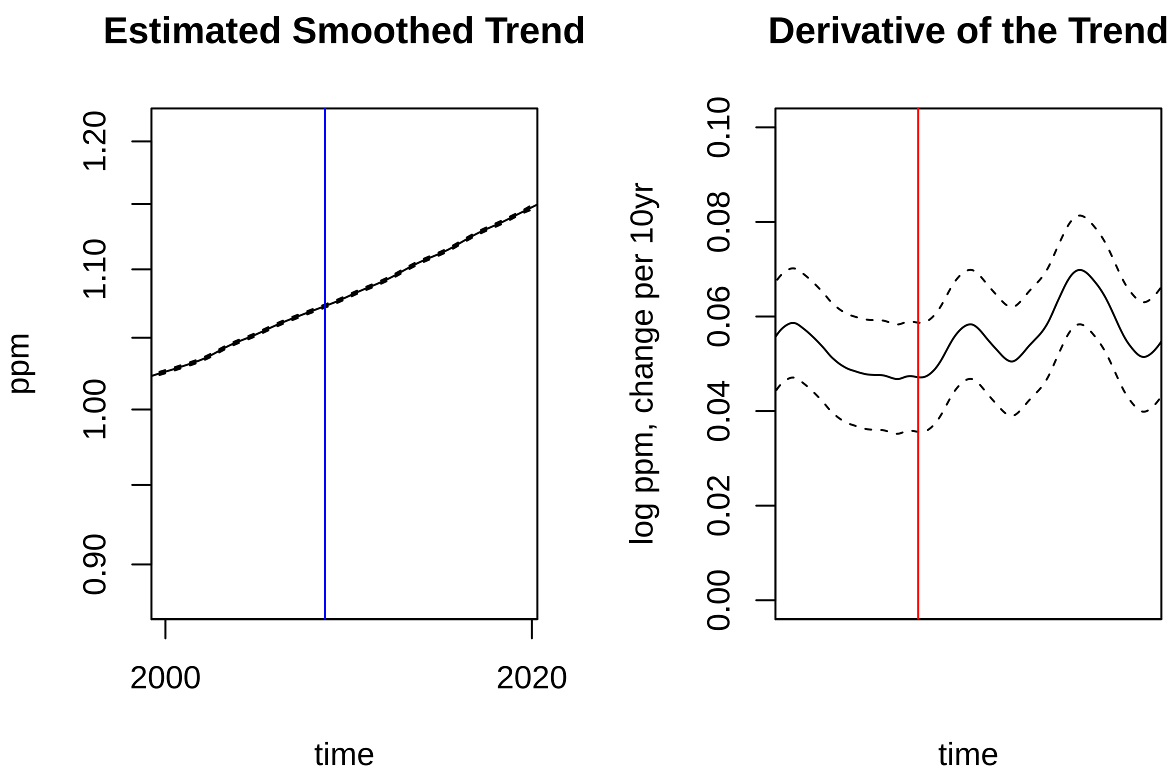
China joining the WTO on 11 December 2001

```
par(mfrow = c(1,2))
# Sequence
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",
        xlim = range(as.numeric(co2s[co2s$date > ISOdate(1990,1,1, tz='UTC')
                                     & co2s$date < ISOdate(2010,1,1, tz='UTC'),
                                     c('date')])),main="Estimated Smoothed Trend")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "blue")

# Derivatives
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[co2s$date > ISOdate(1995,1,1, tz='UTC')
                                     & co2s$date < ISOdate(2005,1,1, tz='UTC'),
                                     c('date')])),
        xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Derivative of the Trend")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "red")
```



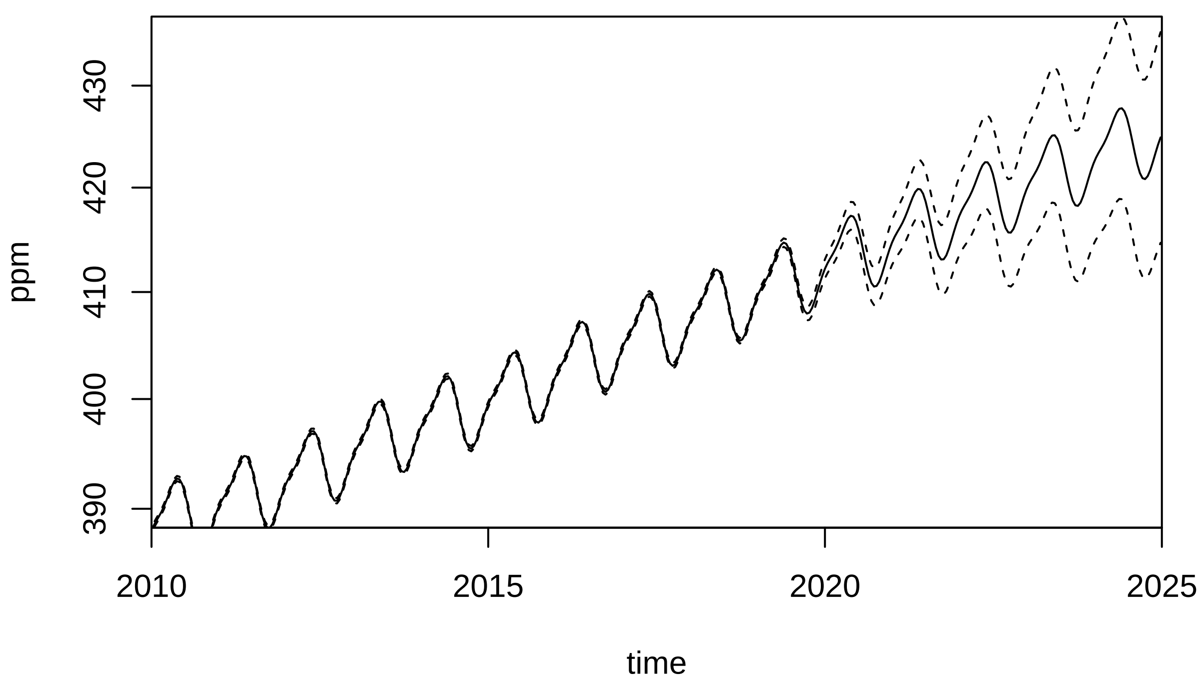
The bankruptcy of Lehman Brothers on 15 September 2008

```
par(mfrow = c(1,2))
# Sequence
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",
        xlim = range(as.numeric(co2s[co2s$date > ISOdate(2000,1,1, tz='UTC')
                                    & co2s$date < ISOdate(2020,1,1, tz='UTC'),
                                    c('date')])),main="Estimated Smoothed Trend")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "blue")

# Derivatives
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[co2s$date > ISOdate(2005,1,1, tz='UTC')
                                    & co2s$date < ISOdate(2015,1,1, tz='UTC'),
                                    c('date')])),
        xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Derivative of the Trend")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "red")
```

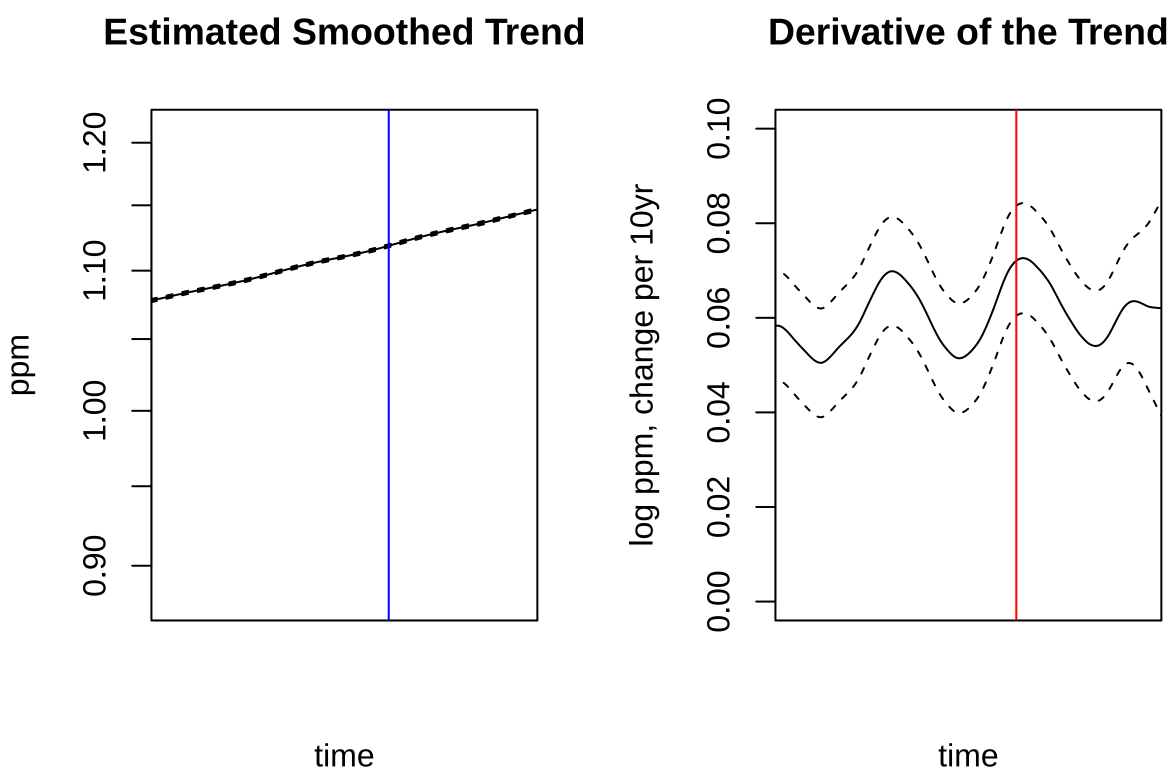


The signing of the Paris Agreement on 12 December 2015

```
# Prediction
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"))
```



```r
par(mfrow = c(1,2))
# Sequence
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",
        xlim = range(as.numeric(co2s[co2s$date > ISOdate(2010,1,1, tz='UTC')
                                     & co2s$date < ISOdate(2030,1,1, tz='UTC'),
                                     c('date')])),main="Estimated Smoothed Trend")
xax = pretty(timePoints)
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "blue")

# Derivatives
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[co2s$date > ISOdate(2010,1,1, tz='UTC')
                                     & co2s$date < ISOdate(2020,1,1, tz='UTC'),
                                     c('date')])),
        xaxs = "i", xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Derivative of the Trend")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "red")
```

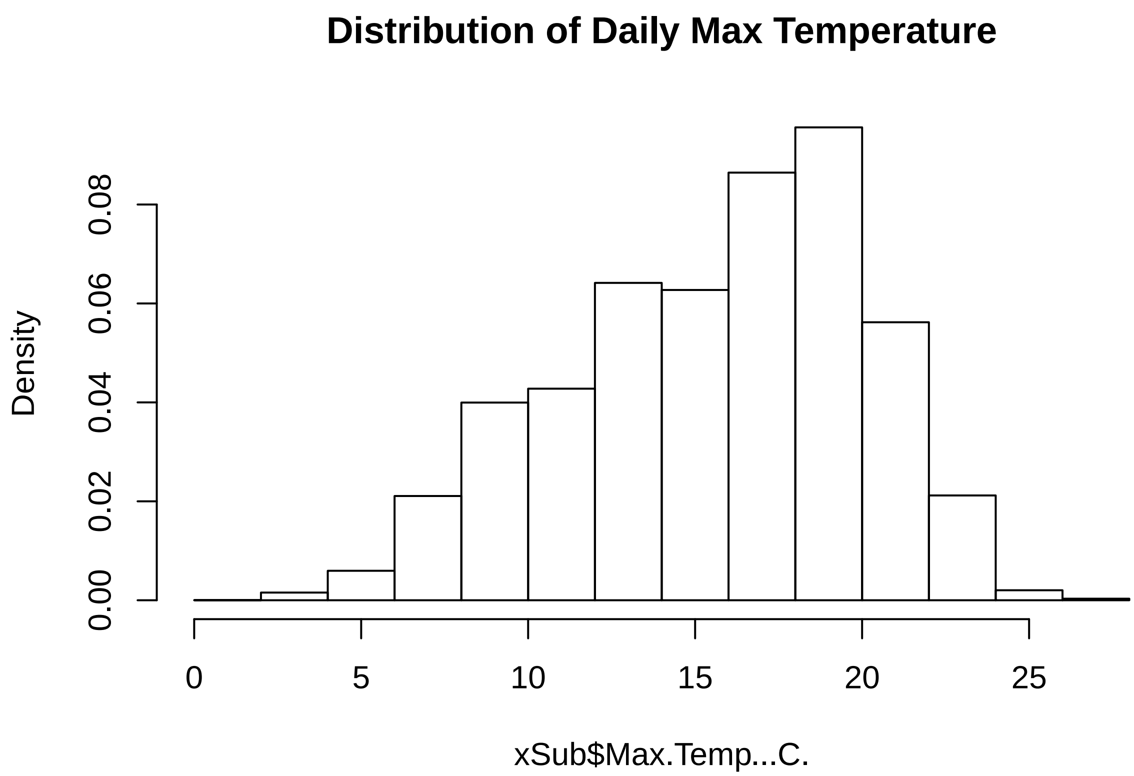


STA442 HW3 Q2

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$weekIid = xSub$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(xSub$Max.Temp...C.,prob=TRUE,main = "Distribution of Daily Max Temperature")
```




```
library("INLA")
```

```
## Loading required package: Matrix
```

```
## Loading required package: sp
```

```
## Loading required package: parallel
```

```
## This is INLA_19.09.03 built 2019-09-03 09:07:31 UTC.
## See www.r-inla.org/contact-us for how to get help.
## To enable PARDISO sparse library; see inla.pardiso()
```

```
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',
```

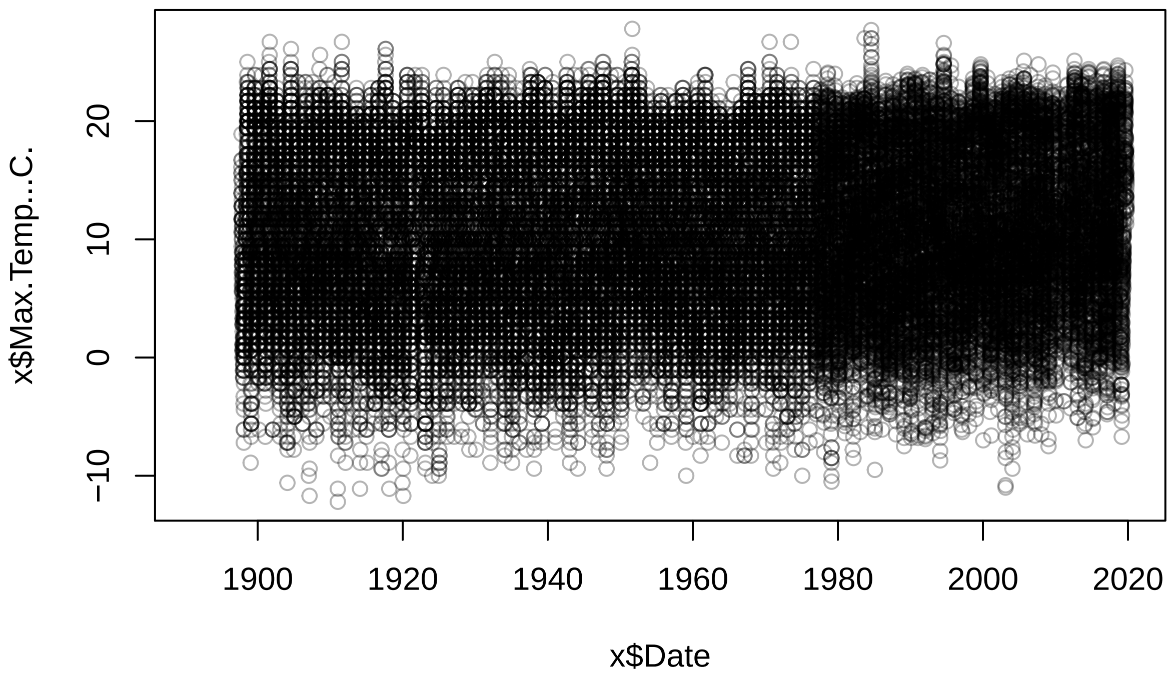
```
                                 param=c(10, 0.5)))),
                   control.mode = list(theta = c(-1,2,20,0,1),
                                       x = startingValues, restart=TRUE),
                   control.compute=list(config = TRUE),
                   data = xSub, verbose=TRUE)
```

```
library(Pmisc)

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))
```

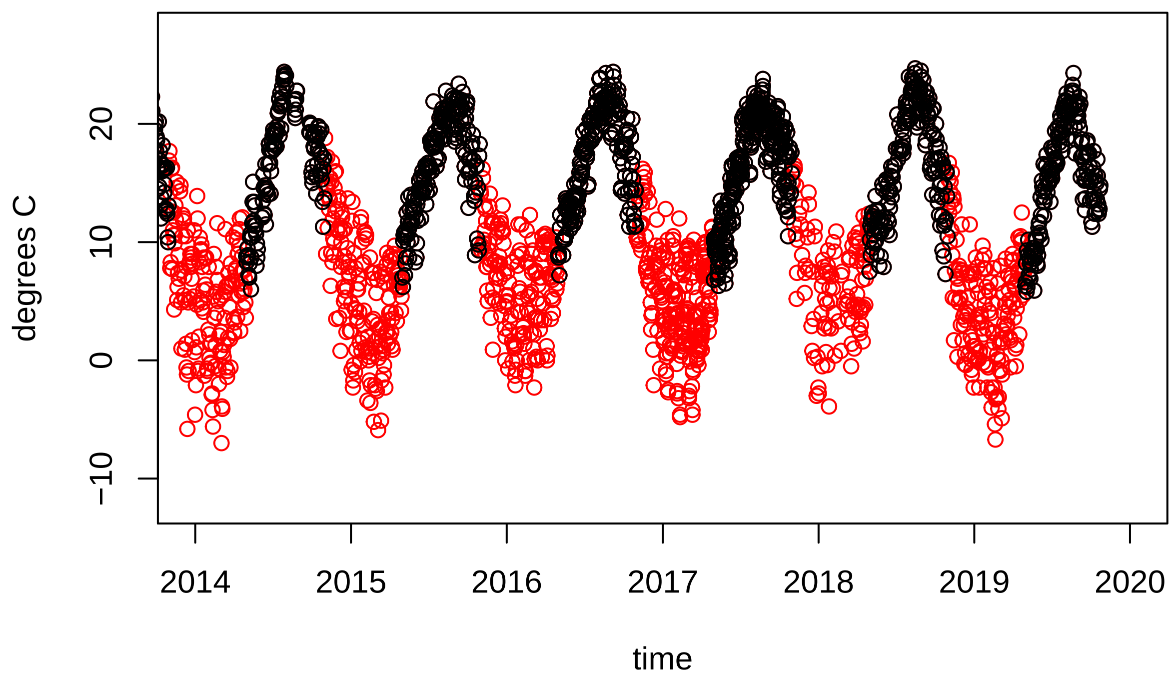
```
library(mapmisc)
```

```
## Loading required package: raster

## map images will be cached in  /var/folders/d6/69h871p92l59n9sx36t_bz000000gn/T//RtmpScbBe1/mapmiscCa
```

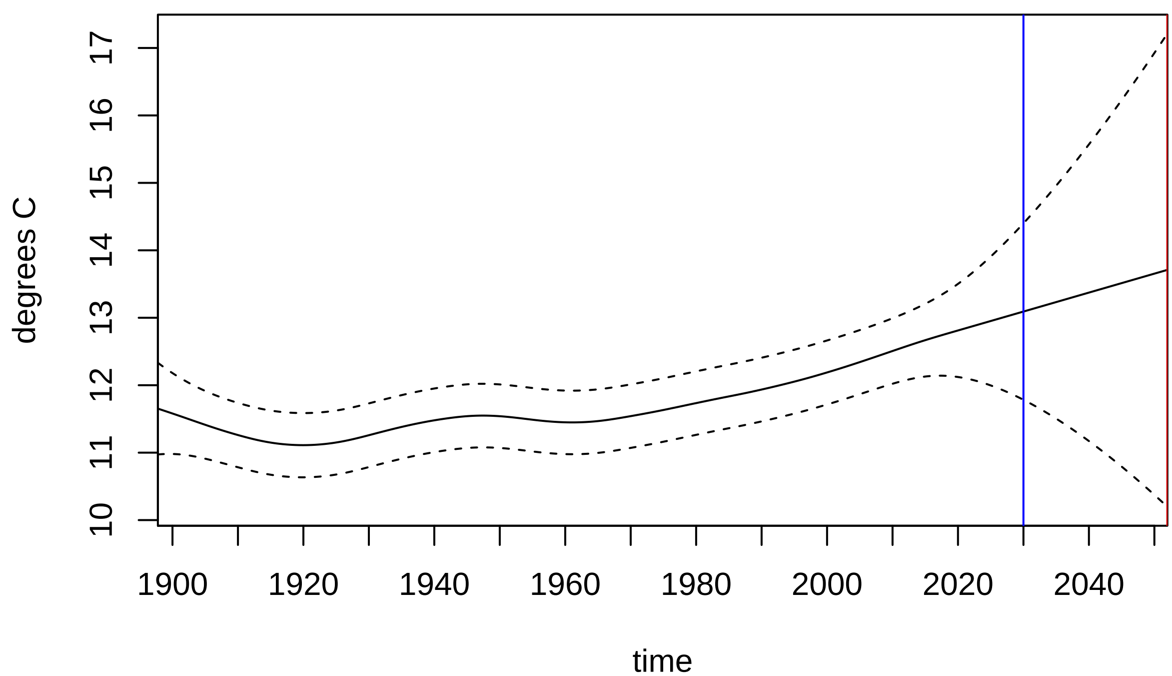
```
plot(x$Date, x$Max.Temp...C., col = mapmisc::col2html("black", 0.3))
```




```
forAxis = ISOdate(2014:2020, 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"
forXaxis2 = ISOdate(seq(1880, 2052, by = 10), 1, 1, tz = "UTC")
axis(1, forXaxis2, format(forXaxis2, "%Y"))
abline(v = ISOdate(2030, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(2052, 1, 1, tz = "UTC"), col = "red")
```




```
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"))
```