STA442 HW3

Depeng Ye 1002079500

27/10/2019

CO2

Consulting Report

At first, we tried to fit a linear model at first:

$$Y_i = X_i + sin12 + sin6 + cos12 + cos6$$

where Y_i is CO_2, X_i is dates, sines and cosines are seasonal. In the four seasonals, cosine and sine that end with 12 are seasonal parameters for one year, while funcitons end with 6 are seasonals for half year. However, linear model was not a good fit as we could see from the plots provided in the appendix.

Hence, we use GAM for this question to find the proper fit, our Y_i follows Gamma distribution:

$$Y_i \sim Gamma(k, \theta)$$

$$\log(Y_i) = X_i\beta + U(t_i) + V_i$$
$$[U_1 \dots U_T]^T \sim RW2(0, \sigma_U^2)$$
$$V_i \sim N(0, \sigma_V^2)$$

where $U(t)$ follows second-order random walk and V_i is random noise.

We could see from the histogram overlay that using a gamma distribution is a good choice for such a data set.

The plot given for the prediction of CO_2 within a year, we could see that using Sine and Cosine funciton are proper for this data set.

Event 1 The OPEC oil embargo which began in October 1973. Indicated as RED cut-off line in the Derivative plot.

We can see from the plot that after the event has taken place, the CO_2 level has significantly decreased. This could also be explained based on our knowledge that less gasoline sold on the market will lead to less CO_2 comsumption which results in a large decrease in CO_2 level. Hence, it proves that our prediction model is reliable based on this part of this historical event.

Event 2 The Global Economic recessions around 1980-1982. Indicated as Orange cut-off line in the Derivative plot.

As we could see from the plot, during the period of 1980-1982, the CO_2 level has decreased significantly. We could interpret from our experience that along with the recession on global economics, companies and countries will have less money to spend on gasoline and industrial progression. Many factories will shut down, and many companies will go on bankcruptcy. Both production and consumption will decrease within the recession period. Hence, the CO_2 level will decrease by a large mount.
Our interpretation coincide with our model prediction in this historical event.

Event 3 The fall of Berlin wall happened on November 1991. Indicated as Green cut-off line in the Derivative plot.

What we see from the plot: the CO_2 level has significantly decreased around the neighbourhood of the green cut-off line. Our interpretation from the historical event is that during the time period of the fall of Berlin wall, both the industrial production in Soviet Union and Eastern Europe has decreased dramatically. Such curtailment in the instudtrial production leads to the incremental decrease in CO_2 level. Once again, our model's prediction coincides with our rational interpretation.

Event 4 China joining WTO on 11 December 2001. Indicated as Cyan cut-off line in the Derivative plot.

As our model predicted, after the entrance on China into WTO, the CO_2 level increases dramatically. China enters WTO causes more trades happening in Chinese Market, which leads a notable increase in the industrial production in China. The growth in production will, of course, lead to the growth in CO_2 level. Our interpretation is consistent with our model prediction.

Event 5 The Bankcruptcy of Lehman Brothers on 15 September 2008. Labeled as Blue cut-off line in the plot.

We could see from the plot that the $\mathrm{CO_2}$ emission level is getting higher and higher after the financial crisis in 2008. According to the research paper on this most recent financial crisis, we know that the energy price decreased significantly during that period resulting in minor changes in global energy structure. Large government investment in many countries to achieve a quick recovery of the economic, and the high GDP growth rate in the developing world caused the $\mathrm{CO_2}$ level to increase instead of decrease, as it has happened in the 1980s. One more time, our model prediciton is consistent with what we have been understanding, and what we have observed from the historical event.

source of the research paper mentioned:https://www.globalcarbonproject.org/global/pdf/pep/Peters_2011_Budget2010.pdf

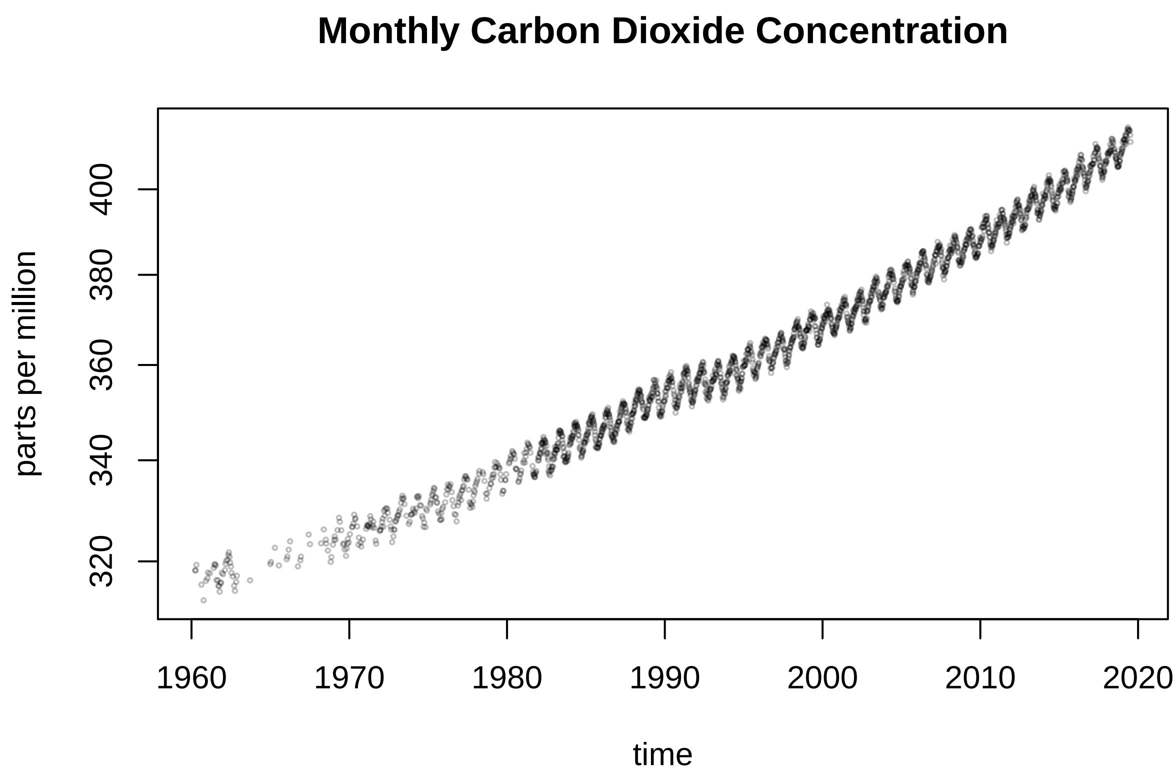
Event 6 The signing of Paris Agreement on 12 December 2015. Indicated in Purple cut-off line in the plot.

After the signing of the Paris Agreement, the $\mathrm{CO_2}$ emission has been decreasing in the following couple of years according to our model prediction, which is consistent with the historical fact that the $\mathrm{CO_2}$ emission level should be lower.

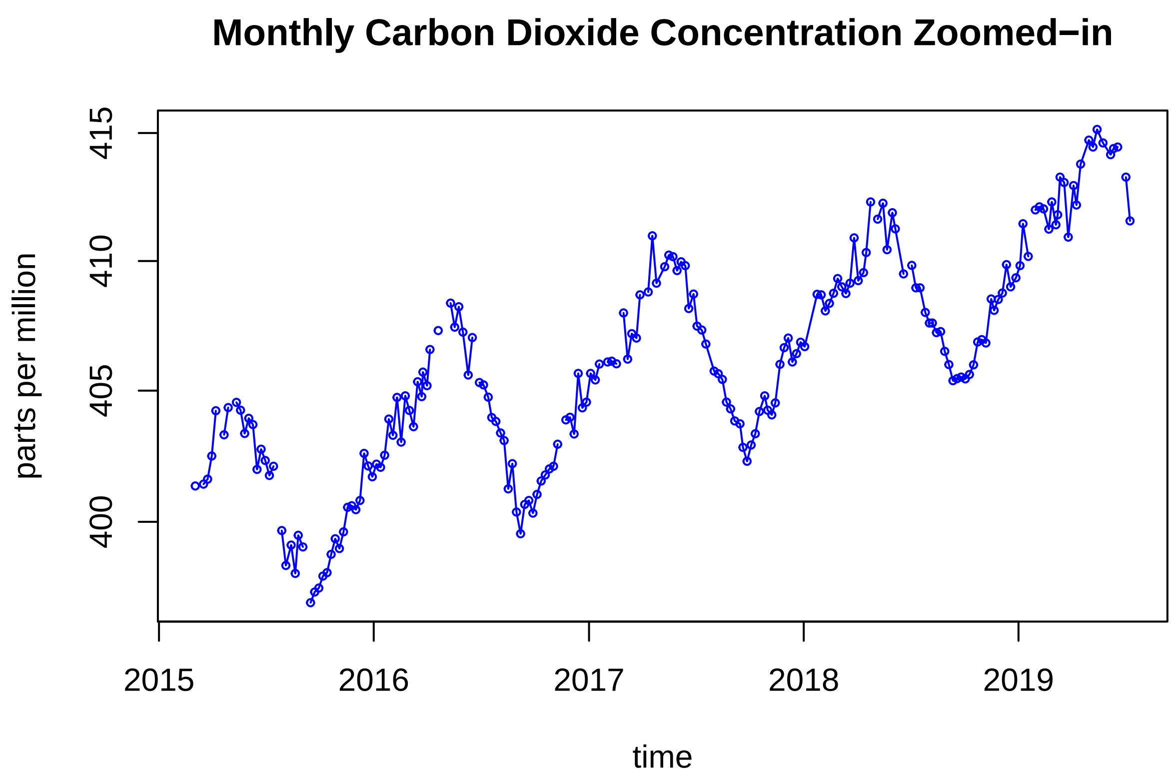
Conclusion With all the above analysis on the 6 given events, we could safe draw a conclusion that our predictive model can properly and accuratly predict the $\mathrm{CO_2}$ level of a certain time period.

Appendix

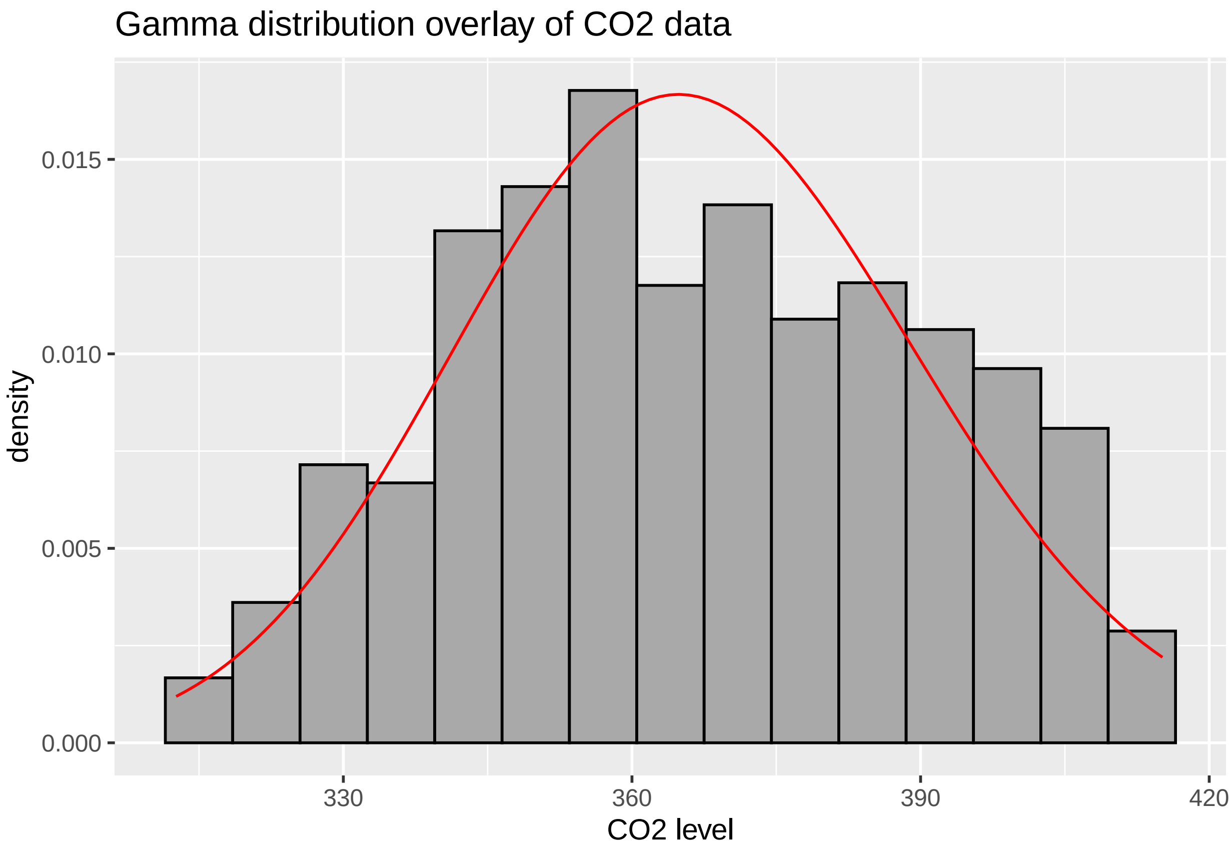
```r
cUrl = paste0("http://scrippsco2.ucsd.edu/assets/data/atmospheric/",
"stations/flask_co2/daily/daily_flask_co2_mlo.csv")
cFile = basename(cUrl)
if (!file.exists(cFile)) download.file(cUrl, cFile)
co2s = read.table(cFile, header = FALSE, sep = ",",
skip = 69, stringsAsFactors = FALSE, col.names = 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 = "parts per million",
main = "Monthly Carbon Dioxide Concentration")
```




```r
plot(co2s[co2s$date > ISOdate(2015, 3, 1, tz = "UTC"),
c("date", "co2")], log = "y", type = "o", xlab = "time",
ylab = "parts per million", cex = 0.5,col = "blue",
main = "Monthly Carbon Dioxide Concentration Zoomed-in")
```



```r
#see how does gamma distribution fit the data
co2s_noNA = na.omit(co2s)
va = var(co2s_noNA$co2)
me = mean(co2s_noNA$co2)
theta = va / me
k = me / theta
ggplot(co2s_noNA, aes(x = co2)) +
  geom_histogram(aes(y = ..density..), binwidth = 7, fill = "darkgrey", col = "black") +
  stat_function(fun = dgamma, args = list(shape = k, scale = theta), col = "red") +
  labs(x = "CO2 level", title = "Gamma distribution overlay of CO2 data")
```

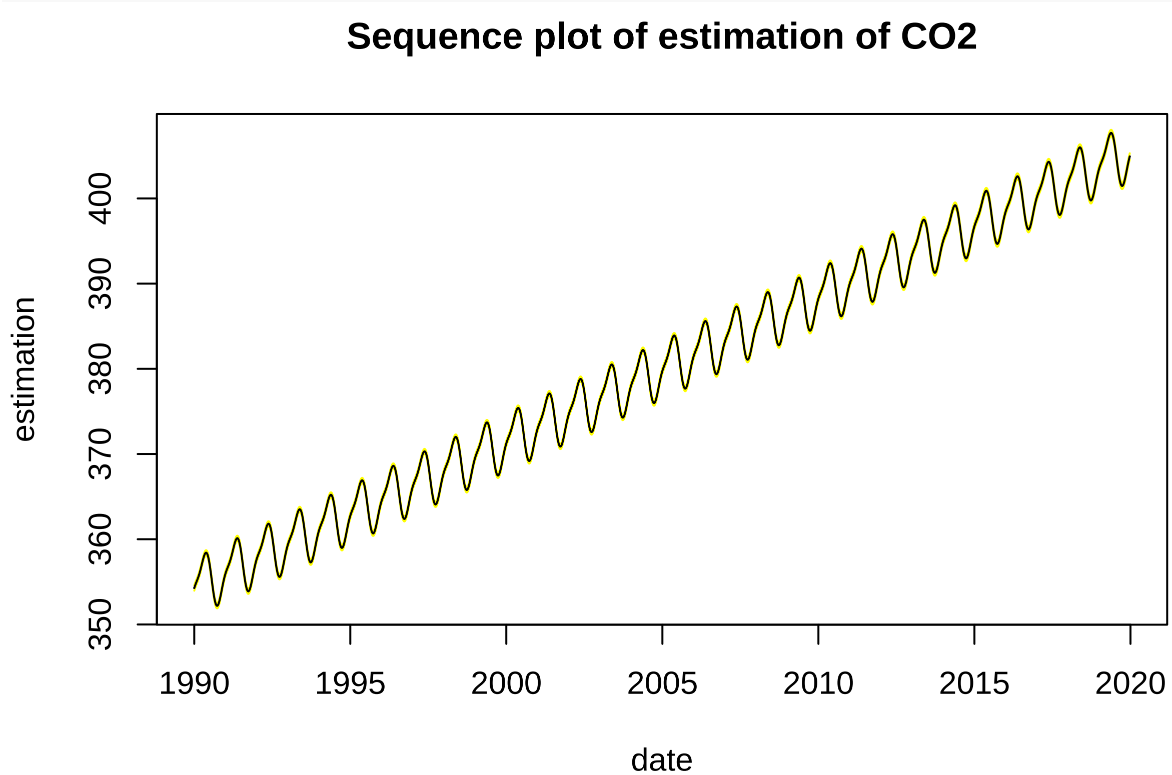



```r
timeOrigin = ISOdate(1980, 1, 1, 0, 0, 0, tz = "UTC")
co2s$days = as.numeric(difftime(co2s$date, timeOrigin,
units = "days"))
co2s$cos12 = cos(2 * pi * co2s$days/365.25)
co2s$sin12 = sin(2 * pi * co2s$days/365.25)
co2s$cos6 = cos(2 * 2 * pi * co2s$days/365.25)
co2s$sin6 = sin(2 * 2 * pi * co2s$days/365.25)

# try linear model first.
cLm = lm(co2 ~ days + cos12 + sin12 + cos6 + sin6,
data = co2s)
knitr::kable(summary(cLm)$coef[, 1:2], digits = 3,
             caption = "Summary of LM fit")
```

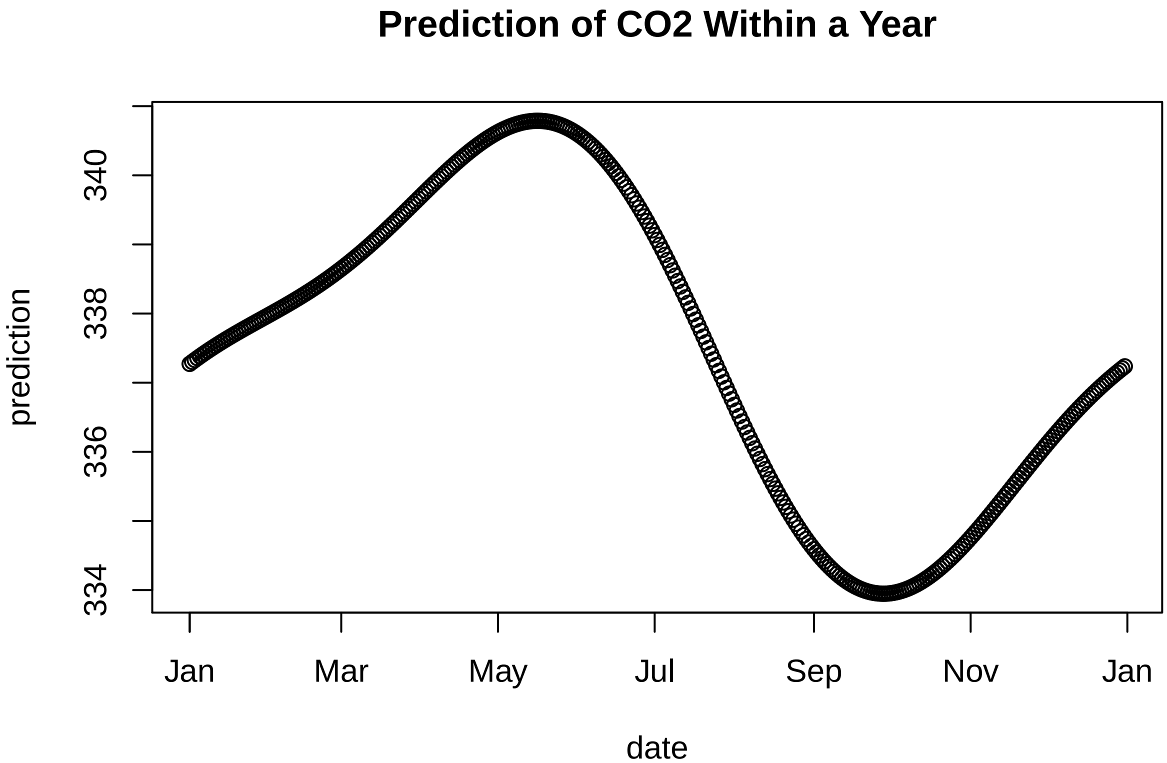
Table 1: Summary of LM fit

	Estimate	Std. Error
(Intercept)	337.499	0.103
days	0.005	0.000
cos12	-0.899	0.093
sin12	2.884	0.092
cos6	0.658	0.092
sin6	-0.613	0.092

```
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", xlab = "date", ylab = "estimation",
     main = "Sequence plot of estimation of CO2")
matlines(as.numeric(newX$date), coPred[, c("lower",
"upper", "est")], lty = 1, col = c("yellow", "yellow", "black"))
```



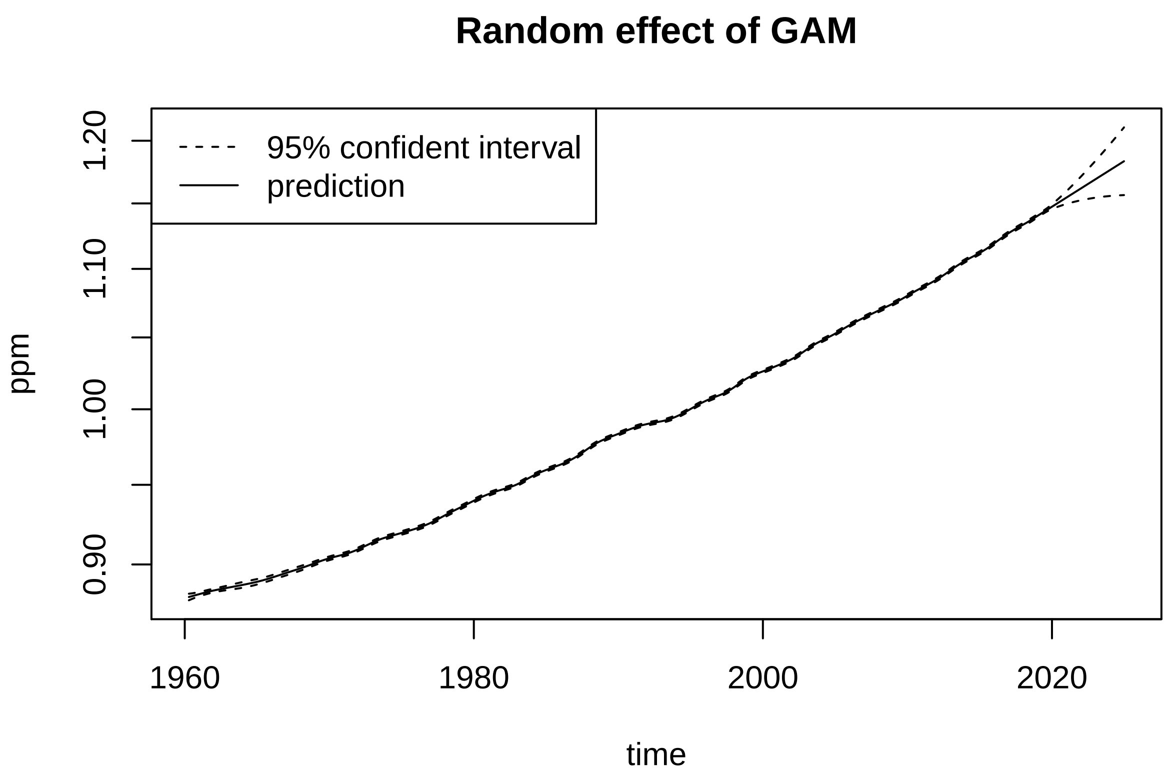
```r
newX = newX[1:365, ]
newX$days = 0
plot(newX$date, predict(cLm, newX), xlab = "date", ylab = "prediction",
     main = "Prediction of CO2 Within a Year")
```



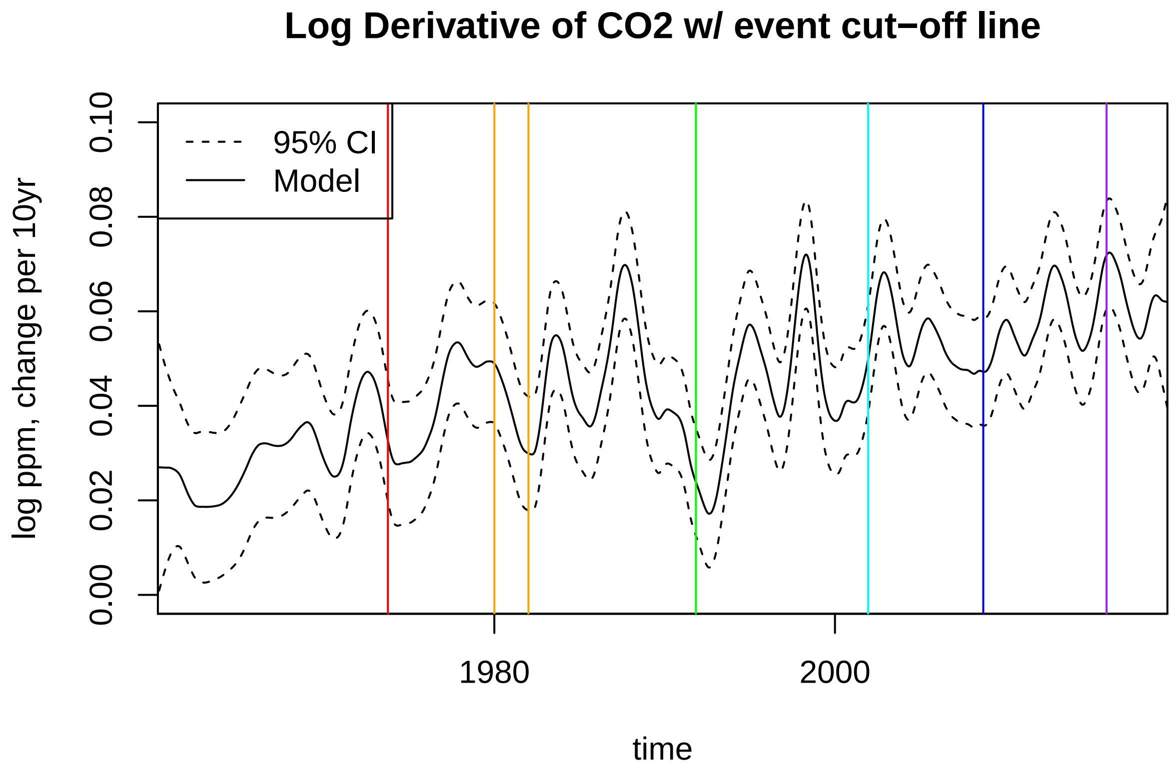

```r
# time random effect
timeBreaks = seq(min(co2s$date), ISOdate(2025, 1, 1,
                                          tz = "UTC"), by = "14 days")
timePoints = timeBreaks[-1]
co2s$timeRw2 = as.numeric(cut(co2s$date, timeBreaks))
# derivatives of time random effect
D = Diagonal(length(timePoints)) - 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)` =
```

```r
                                    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)))
```

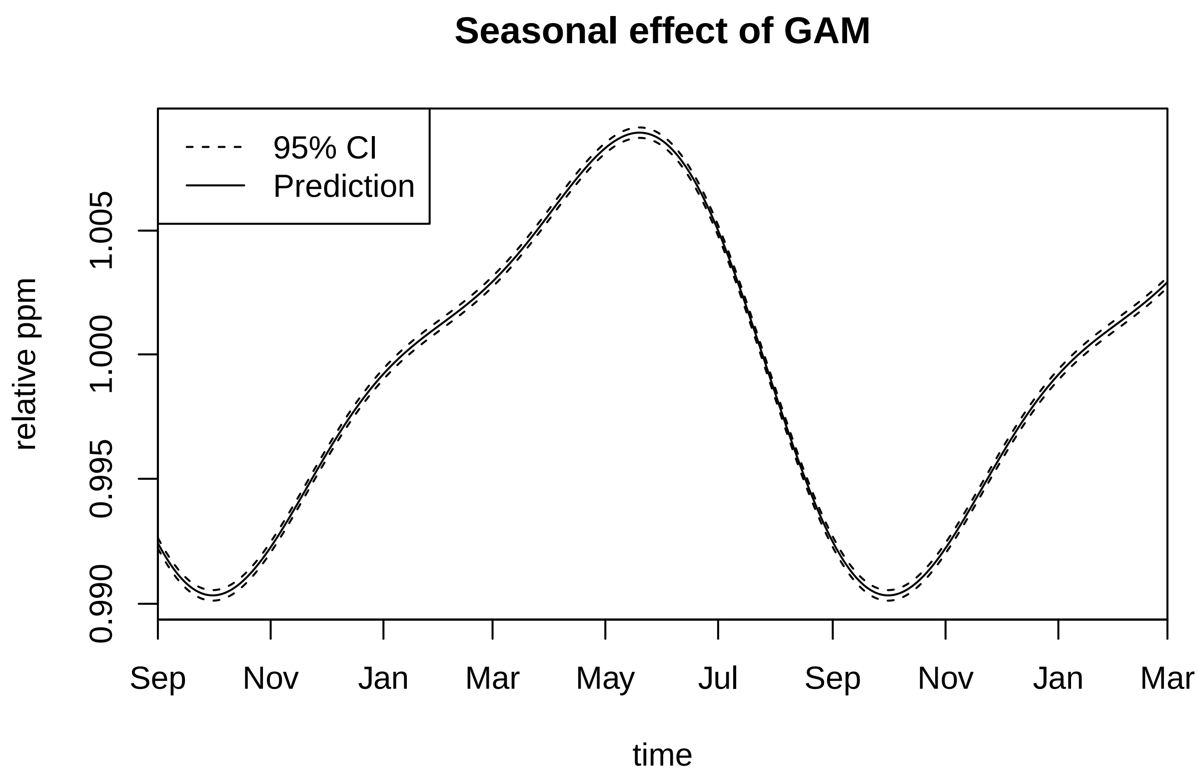
```r
# disable some error checking in INLA
library("INLA")
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())
co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 +
f(timeRw2, model = 'rw2',
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)))),
# 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", main = "Random effect of GAM")
xax = pretty(timePoints)
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")))
legend("topleft", legend = c("95% confident interval", "prediction"),
       col = c("black", "black"), lty = 2:1)
```



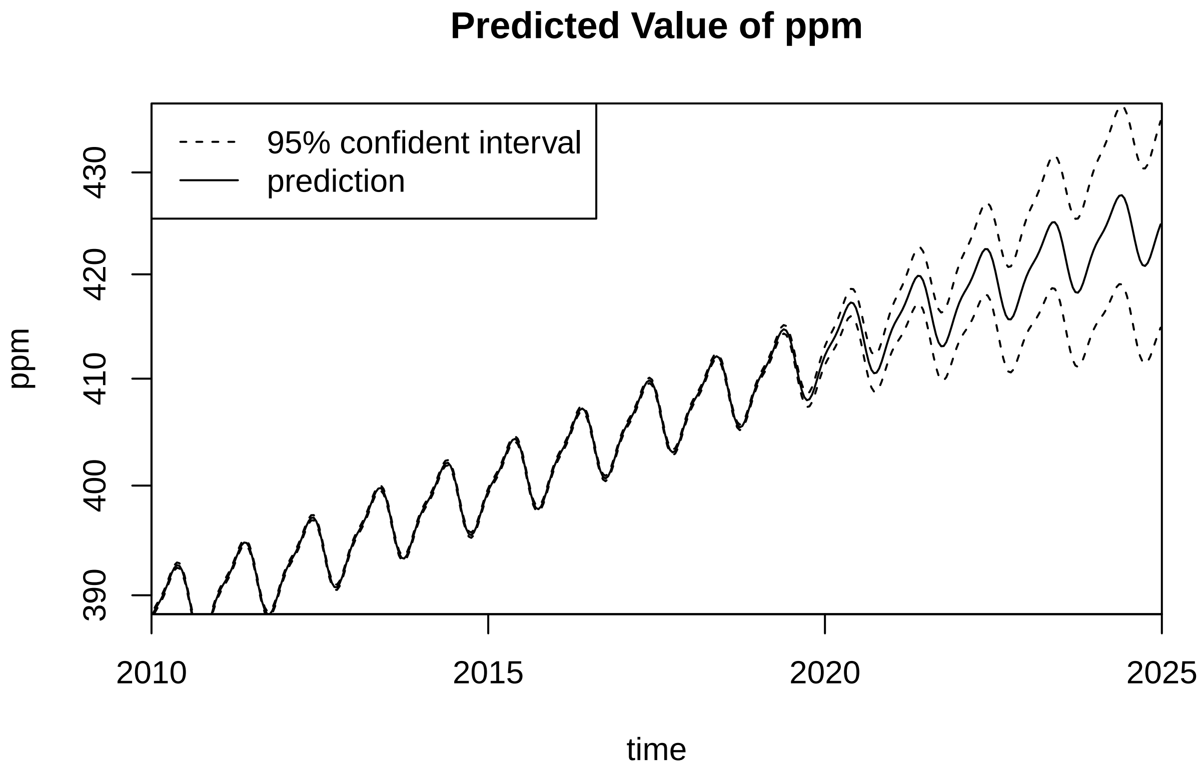

```r
matplot(timePoints[-1], scaleTo10Years * derivPred, type = "l",
        col = "black", lty = c(1, 2, 2), ylim = c(0,0.1),
        xlim = range(as.numeric(co2s$date)), xaxs = "i",
        xaxt = "n", xlab = "time", ylab = "log ppm, change per 10yr",
        main = "Log Derivative of CO2 w/ event cut-off line")
axis(1, xax, format(xax, "%Y"))
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "purple")
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "blue")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "cyan")
abline(v = ISOdate(1991, 11, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(1982, 1, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
legend("topleft", legend = c("95% CI", "Model"),
       col = c("black", "black"), lty = 2:1)
```




```
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 of GAM")
xaxSeason = seq(ISOdate(2009, 9, 1, tz = "UTC"), by = "2 months",
len = 20)
axis(1, xaxSeason, format(xaxSeason, "%b"))
legend("topleft", legend = c("95% CI", "Prediction"),
       col = c("black", "black"), lty = 2:1)
```




```r
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 Value of ppm")
xaxPred = seq(ISOdate(2010, 1, 1, tz = "UTC"), by = "5 years", len = 20)
axis(1, xaxPred, format(xaxPred, "%Y"))
legend("topleft", legend = c("95% confident interval", "prediction"),
       col = c("black", "black"), lty = 2:1)
```


Predicted Value of ppm
95% confident interval
prediction
ppm
390
400
410
420
430
2010
2015
2020
2025
time

Heart

TO: Maxim Burningier

FROM: DEPENG YE

Dear Maxim Burningier,

I am afraid I have to appologies that I can not agree with your opinion. Even though in the first scatter plot, we can not see any relationship or whatsoever in between the temperature, after my detailed investigation, there is a significant relationship between time and temperature. The prediction by IPCC is reliable and accurate I my point of view. The reasons are as follows:

First of all, considering the second plot (the Period Temperature plot from 2016 to the present), we could notice that the variability of winter temperature are dramatically higher than summer temperature. I have consulted a reliable environmental scientist, who told me that only the summer temperature are valuable for modeling a historical temperature time series becaues the winter temperature are governed by many complex physical process. Therefore only the summer temperature were used to evaluate whether the statement of IPCC is flawled or not.

Since we are not quite sure what model the Sable Island data follows, we prefer to use Generalized Additive Model with integrated smothness estimation to evaluate the trend of our data. Our data follows reparameterized standard t-distribution:

$$Y_i \sim T(\nu)$$
$$\log(\nu_i) = X_i\beta + U(t_i) + V_i$$
$$[U_1 \dots U_T]^T \sim RW2(0, \sigma_U^2)$$
$$V_i \sim N(0, \sigma_V^2)$$

where $U(t)$ follows second-order random walk and V_i is random noise.

To show that the summer data follows reparameterized standard t-distribution, we plotted the overlayed t-distribution in the appendix. We could see from the overlay plot that the reparameterized t-distribution fitted the data quite well. We are confident to use such a distribution to fit our model. We reparametrized t-distribution using the formula:

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

The model we have fitted are shown in the third plot. We could see that in a high level of confidence, the temperature is increasing constantly and steadily. In the period of 2016 to present compared with the pre-industrial levels (1750-1850, that is before 1990s), the temperature indeed has increased by 1°C. In the future, according to our model prediction, we could see that the temperature will increase by at least 1.5°C in the future after 2020.

Moreover, same result can be drawn the posterior plot. We could see a similar pattern as we have discussed in the previous part that the maximum temperature has increased by 1°C compared with the pre-industrial period and the temperature will increase by 1.5°C more until 2030-2050.

As a conclusion, the Sable Island data you have provided me actually is supportive to the IPCC statement in a very high confidence level based on my statistical analysis. Hence, it is very unfortunate to tell you that I can not agree with you opinion that the temperature has not changed over the past decades. We need to seriously take care of our environment as soon as possible to prevent or to resist a further and deeper level of global warming.

Sincerely,

Depeng Ye

Appendix

```
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$Date), ISOdate(2030, 1, 1,
0, 0, 0, tz = "UTC"), by = "7 days")
xSub$week = cut(xSub$Date, weekValues)
xSub$weekIid = xSub$week
xSub$day = as.numeric(difftime(xSub$Date, min(weekValues),
units = "days"))
xSub$cos12 = cos(xSub$day * 2 * pi/365.25)
xSub$sin12 = sin(xSub$day * 2 * pi/365.25)
xSub$cos6 = cos(xSub$day * 2 * 2 * pi/365.25)
xSub$sin6 = sin(xSub$day * 2 * 2 * pi/365.25)
xSub$yearFac = factor(format(xSub$Date, "%Y"))
lmStart = lm(Max.Temp...C. ~ sin12 + cos12 + sin6 +
               cos6, data = xSub)
startingValues = c(lmStart$fitted.values, rep(lmStart$coef[1],
nlevels(xSub$week)), rep(0, nlevels(xSub$weekIid) +
nlevels(xSub$yearFac)), lmStart$coef[-1])
INLA::inla.doc('^t$')

library("INLA")
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if(class(mm) == 'function') mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())
sableRes = INLA::inla(
Max.Temp...C. ~ 0 + sin12 + cos12 + sin6 + cos6 +
  f(week, model='rw2', constr=FALSE, prior='pc.prec',
    param = c(0.1/(52*100), 0.05)) + f(weekIid, model='iid',
                                   prior='pc.prec',
                                   param = c(1, 0.5)) +
  f(yearFac, model='iid', prior='pc.prec', param = c(1, 0.5)),
family='T', control.family = list(hyper = list(prec = list(prior='pc.prec',
                                                     param=c(1, 0.5)),
                                               dof = list(prior='pc.dof',
                                                     param=c(10, 0.5)))),
control.mode = list(theta = c(-1,2,20,0,1), x = startingValues, restart=TRUE),
control.compute=list(config = TRUE),
control.inla = list(strategy='gaussian', int.strategy='eb'),
data = xSub, verbose=TRUE)
knitr::kable(sableRes$summary.hyper[, c(4, 3, 5)], digit = 3, caption = "Summary of Prediciton")
```

Table 2: Summary of Prediciton

	0.5quant	0.025quant	0.975quant
precision for the student-t observations	3.190000e-01	3.130000e-01	3.270000e-01

	0.5quant	0.025quant	0.975quant
degrees of freedom for student-t	1.382700e+01	1.128800e+01	1.666800e+01
Precision for week	1.449157e+09	9.732429e+08	2.069335e+09
Precision for weekIid	8.410000e-01	7.830000e-01	9.010000e-01
Precision for yearFac	2.097000e+00	1.603000e+00	2.799000e+00

```
knitr::kable(sableRes$summary.fixed[, c(4, 3, 5)], digit = 3, caption = "Summary of Seasonal effect")
```

Table 3: Summary of Seasonal effect

	0.5quant	0.025quant	0.975quant
sin12	-4.686	-5.202	-4.170
cos12	4.768	4.478	5.058
sin6	-2.045	-2.268	-1.822
cos6	-0.150	-0.321	0.021

```
knitr::kable(Pmisc::priorPostSd(sableRes)$summary[, c(1, 3, 5)], digit = 3, caption = "Summary of poste
```

Table 4: Summary of posteriors

	mean	0.025quant	0.975quant
SD for week	0.000	0.000	0.00
SD for weekIid	1.091	1.054	1.13
SD for yearFac	0.691	0.598	0.79

```
mySample = inla.posterior.sample(n = 24, result = sableRes,
num.threads = 8, selection = list(week = seq(1,
nrow(sableRes$summary.random$week))))
names(mySample[[1]])
```

```
## [1] "hyperpar" "latent"   "logdens"
```

```
weekSample = do.call(cbind, lapply(mySample, function(xx) xx$latent))
dim(weekSample)
```

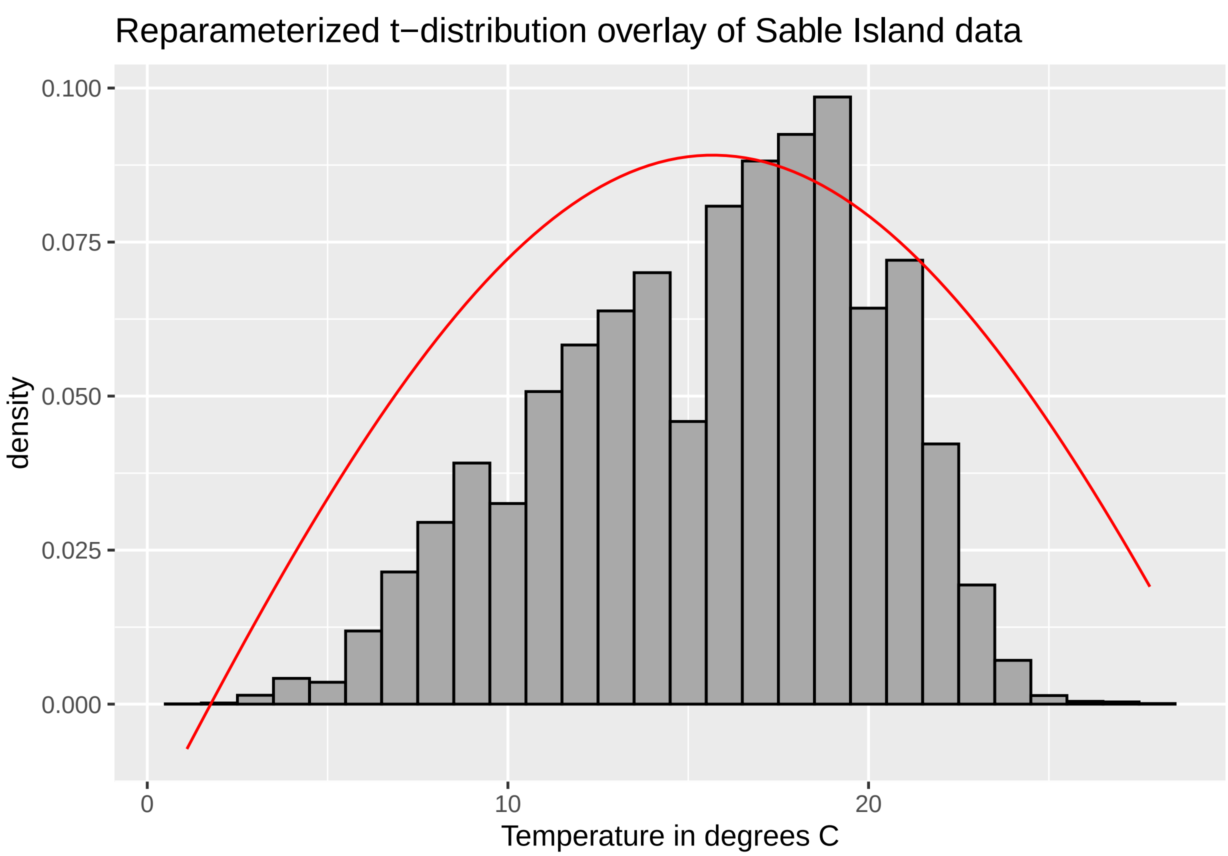
```
## [1] 6900   24
```

```
head(weekSample)
```

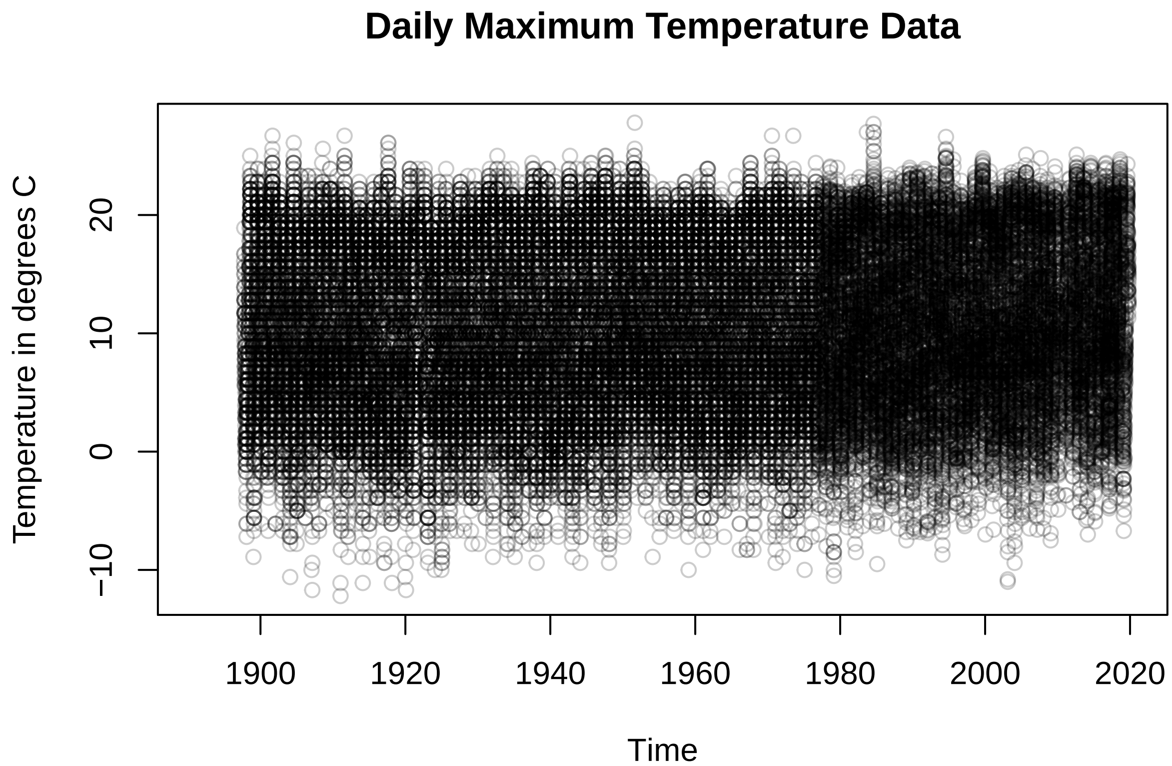
```
##         sample1  sample2  sample3  sample4  sample5  sample6  sample7
## week:1 11.77305 11.80115 11.00769 11.65335 10.95556 12.10672 11.27889
## week:2 11.77136 11.80024 11.00736 11.65289 10.95536 12.10602 11.27938
## week:3 11.76969 11.79932 11.00704 11.65245 10.95517 12.10528 11.27986
## week:4 11.76800 11.79845 11.00675 11.65199 10.95500 12.10452 11.28035
## week:5 11.76640 11.79761 11.00643 11.65152 10.95481 12.10377 11.28083
## week:6 11.76475 11.79678 11.00613 11.65105 10.95459 12.10300 11.28128
##         sample8  sample9 sample10 sample11 sample12 sample13 sample14
## week:1 11.66154 11.90764 11.87724 11.12048 11.47056 11.65580 11.46721
## week:2 11.66165 11.90695 11.87632 11.12109 11.47029 11.65467 11.46570
## week:3 11.66172 11.90625 11.87543 11.12169 11.47001 11.65354 11.46420
## week:4 11.66176 11.90553 11.87457 11.12232 11.46970 11.65238 11.46270
```

```
## week:5 11.66176 11.90479 11.87368 11.12292 11.46939 11.65123 11.46118
## week:6 11.66174 11.90405 11.87277 11.12349 11.46909 11.65007 11.45968
##        sample15 sample16 sample17 sample18 sample19 sample20 sample21
## week:1 11.70082 11.90628 11.62300 11.87247 12.01875 11.74739 10.99463
## week:2 11.69914 11.90621 11.62252 11.87149 12.01771 11.74710 10.99456
## week:3 11.69750 11.90609 11.62208 11.87049 12.01671 11.74680 10.99451
## week:4 11.69586 11.90595 11.62162 11.86951 12.01570 11.74653 10.99444
## week:5 11.69423 11.90581 11.62116 11.86851 12.01465 11.74630 10.99437
## week:6 11.69258 11.90565 11.62075 11.86749 12.01359 11.74609 10.99427
##        sample22 sample23 sample24
## week:1 11.45214 12.02151 11.26819
## week:2 11.45173 12.01999 11.26891
## week:3 11.45131 12.01842 11.26961
## week:4 11.45091 12.01686 11.27033
## week:5 11.45049 12.01527 11.27100
## week:6 11.45008 12.01366 11.27167
```

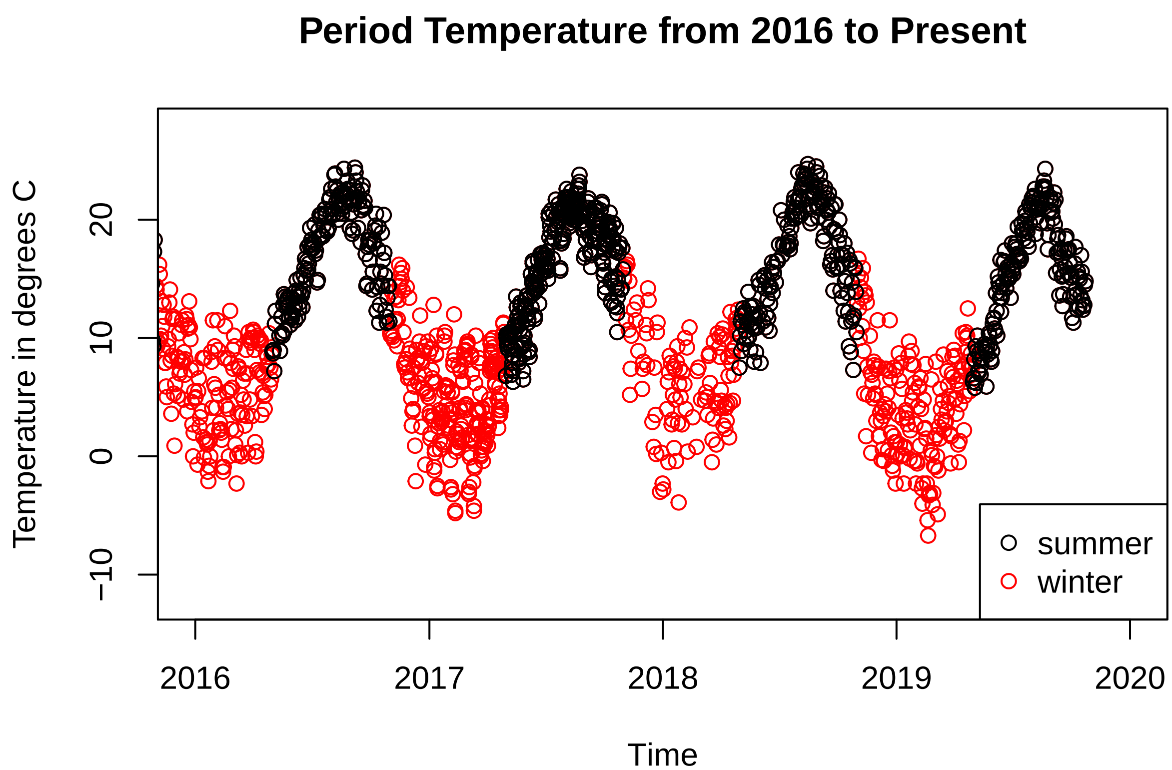
```r
#plot the reparametrized t-distribution overlay
eta = mean(xSub$Max.Temp...C.)
tau = var(xSub$Max.Temp...C.)
custom <- function(x) {dt(0.05 * (x - eta), 10) - 0.3
}
ggplot(xSub, aes(x = Max.Temp...C.)) +
  geom_histogram(aes(y = ..density..), binwidth = 1, fill = "darkgrey", col = "black") +
  stat_function(fun = custom, col = "red") +
  labs(x = "Temperature in degrees C",
       title = "Reparameterized t-distribution overlay of Sable Island data")
```



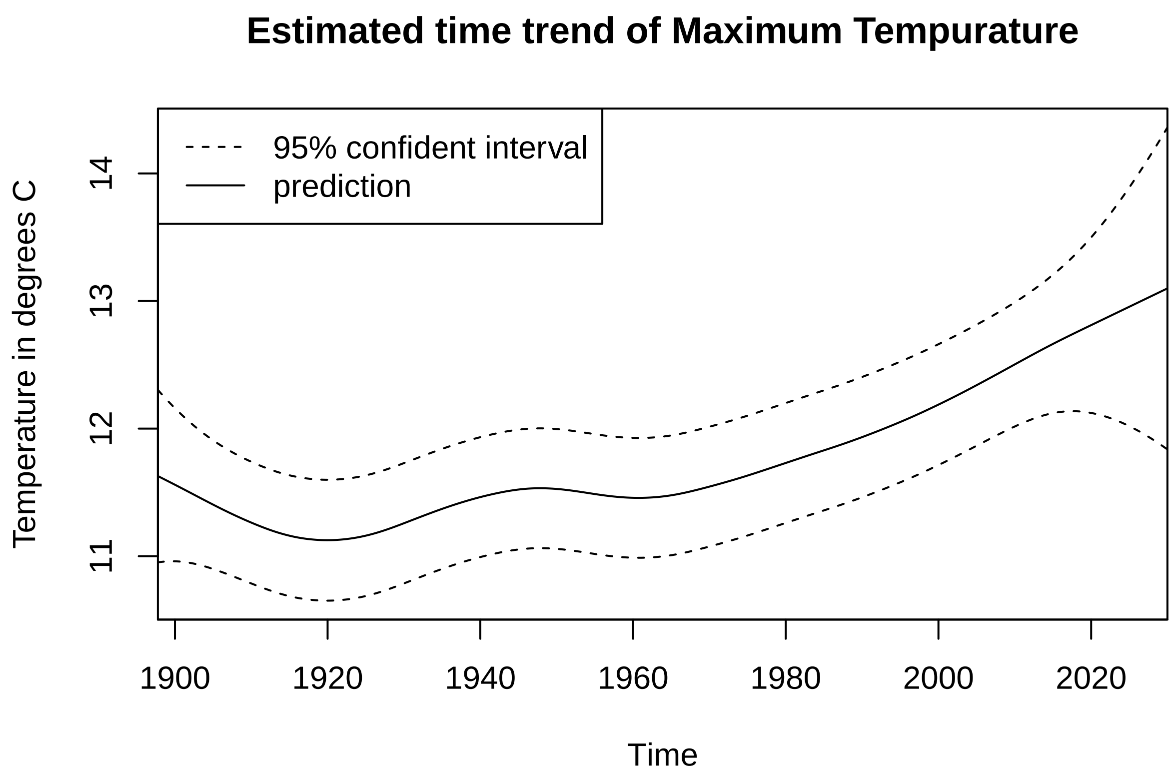
```r
# hist(xSub$Max.Temp...C., prob = T)
# lines(custom(0:25), type = "l")


plot(x$Date, x$Max.Temp...C., col = mapmisc::col2html("black", 0.2),
     ylab = "Temperature in degrees C", xlab = "Time",
     main = "Daily Maximum Temperature Data")
```




```r
forAxis = ISOdate(2016:2020, 1, 1, tz = "UTC")
plot(x$Date, x$Max.Temp...C., xlim = range(forAxis),
     xlab = "Time", ylab = "Temperature in degrees C", col = "red",
     xaxt = "n", main = "Period Temperature from 2016 to Present")
points(xSub$Date, xSub$Max.Temp...C.)
axis(1, forAxis, format(forAxis, "%Y"))
legend("bottomright", legend = c("summer", "winter"),
       col = c("black", "red"), pch = 1:1)
```




```
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 = "Temperature in degrees C",
        xaxt = "n", col = "black", xaxs = "i",
        main = "Estimated time trend of Maximum Tempurature")
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
legend("topleft", legend = c("95% confident interval", "prediction"),
       col = c("black", "black"), lty = 2:1)
```




```r
matplot(weekValues[-1], weekSample,
        type = "l", lty = 1, col = myCol,
        xlab = "Time", ylab = "Temperature in degrees C", xaxt = "n",
        xaxs = "i", main = "Posterior Sample of Time Trend")
axis(1, forXaxis2, format(forXaxis2, "%Y"))
```