<u>**Question 1 -- CO2**</u>

We analyzed the data set of CO2 concentrations of Hawaii from scrippscxo2.ucsd.edu. The main problem was to discuss how the CO2 data appears to be impacted by historical events. Since the linear predictor in this question depends linearly on unknown smooth functions of some predictor variables. The nonparametric approach is more flexible. We used **Generalized Additive Model from gamma family with log link functions** to predict the CO2 concentration of Hawaii since JAN 1, 1897.

$$\mathrm{Y_i} \sim \mathrm{Gamma}(\lambda_i) \quad \log(\lambda_i) = X_i\beta + f(W_i) + V_i$$

$$Y_i \textit{ are the concentration of carbon that measured on the specific days}$$

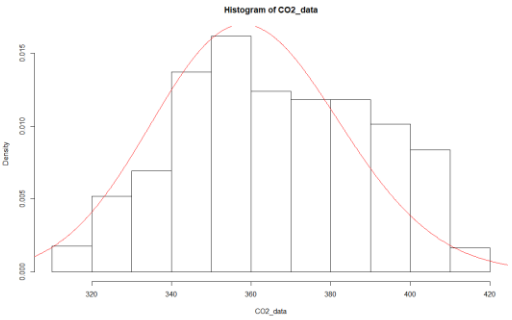
$$\textit{yearly fluctuations}: X_1 = \cos(2 * \pi * x_i) \ \& \ X_2 = \sin(2 * \pi * x_i);$$

$$\textit{biyearly fluctuations}: X_3 = \cos(4 * \pi * x_i) \ \& \ X_4 = \sin(4 * \pi * x_i)$$

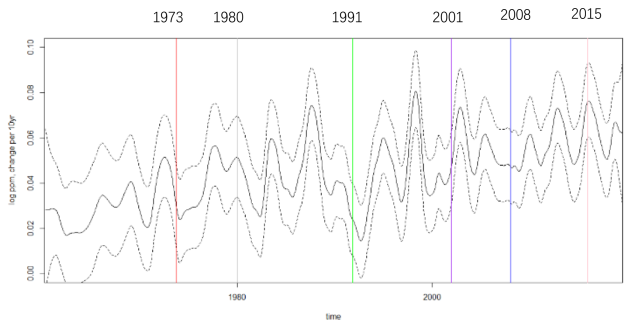
$$\textit{We set } 1980 \textit{ as time origin}, x_i \textit{ are the year's gap from } 1980 \textit{ to specific days's year}$$

$$f(w) \textit{ a smoothly} - \textit{varing function with roughness parameter } v$$
$$- \textit{caused random walk of order } 2 \textit{ with variance } \sigma^2$$

$$\mathrm{V_i} \sim N(0, \sigma_v^2) \ V_i \textit{ covers independent variation or over} - \textit{dispersion}$$



Graph1



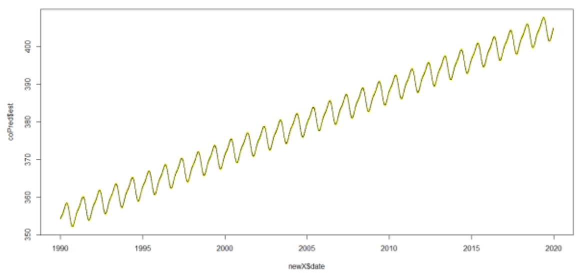
Graph2

We used histogram of data to check if the data fits the model well. The data and fitted line followed the same pattern (looks gamma) which was good. Because we used Bayesian here, we need a prior for all unknown quantities. Since we don't know $\sigma^2_{f(w)}$, we chose **penalized complexity prior** (PC Prior) which put an exponential prior on the standard $\sigma^2_{f(w)}$. We set a penalized complexity prior for f(w) by $\mathrm{P}(\sigma > \mathrm{a}) = \mathrm{b}$. Param = c (log (1.01)/26, 0.5) indicated that $\mathrm{P}\left(\sigma_{f(w)} > \log(1.01)/26\right) = 0.5$. For interpretation, there was a 50% chance that between subject variability $\sigma_{f(w)} > \log(1.01)/26$.

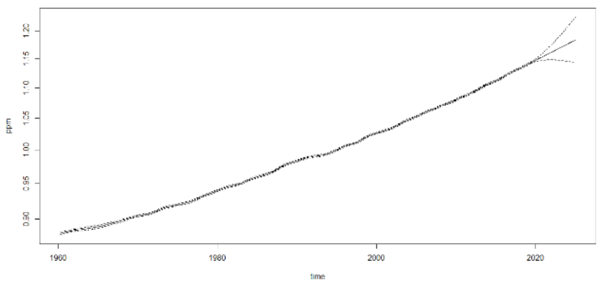
We analyzed the derivative of the trend over time from graph 2. From **October 1973**, the beginning of the OPEC oil embargo led to a small decrease in the growth rate of CO2 concentrations. In the time of **1980 to 1982**, since the global economic recessions led to factory closure, the growth rate of CO2 levels had a sharp decline and had a significant fluctuation until 1990. In **1991**, the fall of the Berlin wall coincided with a dramatic fall in industrial production in the Soviet Union and Eastern Europe; thus, the growth rate of CO2 levels experienced a downward trend.

Furthermore, China joining the WTO on **11 December 2001**, due to rapid growth in industrial production, the growth rate of CO_2 concentrations increased dramatically and reached a peak in 2003. On **15 September 2008**, the Lehman Brothers' bankruptcy symbolized the most recent financial crisis, which led to a moderate decrease in the growth rate of CO_2. In **2015**, the growth rate of CO_2 emissions reached a peak and started a dramatic decline since the signing of the Paris Agreement limited CO_2 emissions.

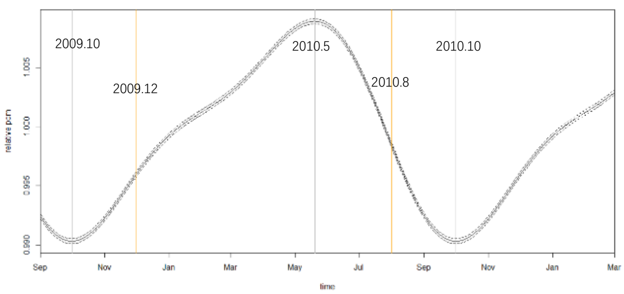
Graph3: Observed vs Predicted



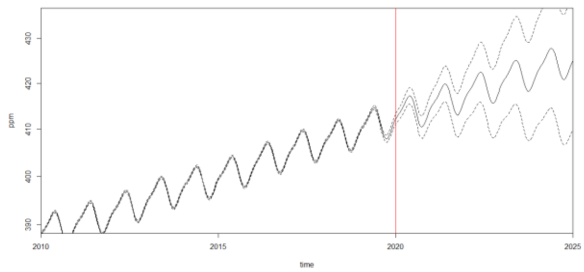
Graph4



Graph5: Seasonal Variation in Carbon Levels from 2009.9,1 to 2011.3.1



Graph6



From graph 4, we noticed that the **CO_2 concentration continuously increasing with an upward tendency through the time**; in other words, f(w) is increasing. From graph 5, we looked at the CO_2 level's variation from 1 September 2009 to 1 Mar 2011. During this period, we had the smallest credible interval at Dec 2009 and Aug 2010; also, the largest credible interval on May 2010 (peak), Oct 2009(nadir) and Oct 2010(nadir). We also could conclude that on Oct 2009 and May 2010 had significantly different CO_2 levels.

From graph 6, we plot the predicted CO_2 levels for the next ten years. In 2020, the CO_2 concentrations are expected to be between 410 ppm to 414 ppm, which has a small variation. The variation is increasing. In 2025, the CO_2 levels' credible interval will be 408 ppm to 440 ppm.

Question 2 – Heat

We used R to analyze the temperature data recorded on Sable Island in detail. The data was available from http://pbrown.ca/teaching/appliedstats/data/sableIsland.rds . We researched how many degrees of global warming affected by human activities. We used **Generalized Additive Model from "T" family with identity link functions** to predict the temperature change since OCT 1, 1897.

$$Y_i \sim t(\lambda_i, \theta) \quad \log(\lambda_i) = X_i\beta + f(W_1) + f(W_2) + f(W_3) + V_i$$

$$Y_i \textit{ are the temperature that measured on the specific days}$$

$$yearly\ fluctuations: X_1 = \cos(2 * \pi * x_i)\ \&\ X_2 = sin(2 * \pi * x_i);$$

$$biyearly\ fluctuations: X_3 = \cos(4 * \pi * x_i)\ \&\ X_4 = sin(4 * \pi * x_i)$$

$$We\ set\ OCT1, 1897\ as\ time\ origin, x_i\ are\ the\ year's\ gap\ from\ OCT1, 1897\ to\ specific\ days's\ year$$

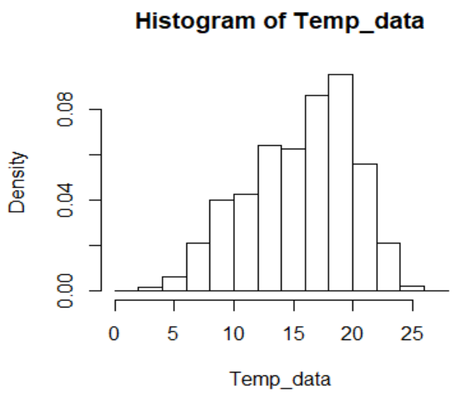
$$f(w)\ a\ smoothly - varing\ function\ with\ roughness\ parameter$$

$$\mathrm{f(W_1): week - Random\ Walk\ of\ order\ 2\ ;\ \ f(W_2): weekIid\ - Random\ intercept;}$$

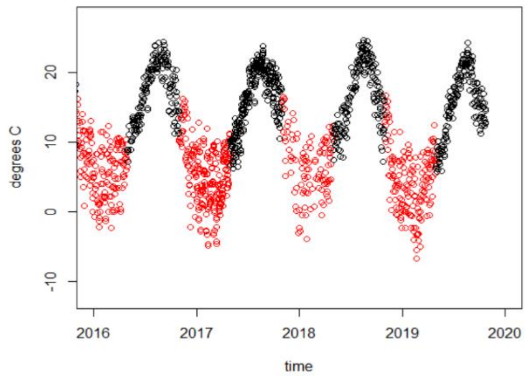
$$\mathrm{f(W_3): yearFac - Random\ intercept}$$

$$\mathrm{note: a\ random\ walk\ time\ series\ is\ not\ satationary}$$

$$V_i \sim N(0, \sigma_v^2)\ \ V_i\ covers\ independent\ variation\ or\ over - dispersion$$



Graph 1

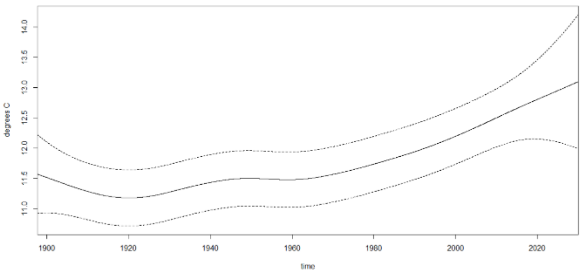


Graph 2

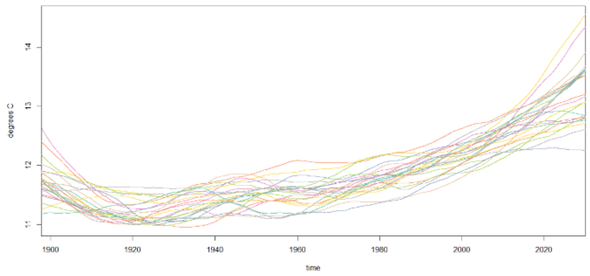
We used the histogram of data to check if the data fit the model well. The data and fitted line followed the same pattern (looks like T-distribution), which was good. Graph 2 illustrated that the summer(black) and winter(red) months' temperature from 2016 to the present. We have noticed that winter's temperature was much lower than in summers. It had a repeating pattern like sine and cosine.

We need a prior for all unknown quantities. Since we don't know the distribution of $\sigma^2_{f(w)}$ and σ^2_v, we chose **penalized complexity prior** (PC Prior) which put an exponential prior on the standard $\sigma^2_{f(w)}$ *and* σ^2_v. We set a penalized complexity prior for f(w) and V by $\mathrm{P}(\sigma > a) = b$. For $f(W_1) - week$, we set param = c (0.1/ (52*100), 0.05)) which indicated that $\mathrm{P}\big(\sigma_{f(w_1)} > 0.1/(52 * 100)\big) = 0.05$. For interpretation, there was a 5% chance that between subject variability $\sigma_{f(w)} > 0.1/(52 * 100)$. Furthermore, for $f(W_2)$ *and* $f(W_3)$, we both set param = c(1, 0.5) which indicated that $\mathrm{P}\big(\sigma_{f(w_2)} > 1\big) = 0.5$ and $\mathrm{P}\big(\sigma_{f(w_3)} > 1\big) = 0.5$. For interpretation, there was a 50% chance that between subject variability $\sigma_{f(w_2)} > 1$ and $\sigma_{f(w_3)} > 1$. Moreover, we also need to set a prior for independent variation V. Param=c (1, 0.5) illustrated that there was a 50% chance that between subject variability $\sigma_v > 1$.

Furthermore, the dof-prior was the PC Prior for degrees of freedom parameter in the Student-t distribution. The parameters to this prior was (a, b); for interpretation was P(v < a)= b where v was degree of freedom. In this case, Prior='pc.dof', param=c (10, 0.5)
indicated that P (v < 10) =0.5; in other words, there was a 50% chance that $v < 10$.



Graph 3



Graph 4

The data from Sable island is broadly supportive of the statement from the IPCC. Based on a random week sample, the estimated degrees increased from 11.6 to 13.1 degrees from 1880 to 2040. The 95% credible interval in 1880 was (10.9°C, 12.1°C) with a range of 1.2°C. For interpretation, there was a 95% chance that the parameter is in the interval (10.9°C, 12.1°C). The degrees reached a nadir in 1920 with a range of 1.1°C and it reached a small peak in 1950 with range of 0.9°C. Thus, model results strongly supported the IPPC's statement that **human activities were 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**.

The global warming has continuous upward tendency based on the current increasing rate. The credible interval becomes wider after 2020. In 2030, we predict the credible interval is (11.9°C, 14.1°C) with a range of 2.2°C. In other words, there is a 95% chance that the parameter is in the (11.9°C, 14.1°C). The IPCC's idea is that global warming is likely to reach 1.5°C between 2030 and 2052 if it continues to increase at the current rate. We support this statement partially. In fact, the global warming will reach **more than 1.5°C** (likely 2.2°C) between 2030 and 2052 based on current trend.

In graph 4, we show posterior samples of the temperature' trend. It clearly shows the global temperature's tendency is similar to the pattern in one sample graph (graph 3). Human activities have significant impacts on global warming and lead to higher temperatures. For further research, due to winter temperatures' considerable variation, we may consider using only summer temperature data for modeling.

Appendix for Code:

Q1:

```
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$time), format = "%Y-%m-%d %H:%M",
                                    tz = "UTC")
co2s$date
# 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")

plot(co2s[co2s$date > ISOdate(2015, 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$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))

library("INLA")
# time random effect
timeBreaks = seq(min(co2s$date), ISOdate(2025, 1, 1,
                                          tz = "UTC"), by = "14 days")
timeBreaks
timePoints = timeBreaks[-1]
timePoints
co2s$timeRw2 = as.numeric(cut(co2s$date, timeBreaks))
co2s$timeRw2

# 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)))

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)
summary(co2res)

# histogram
CO2_data=na.omit(co2s$co2)
mean(CO2_data)
var(CO2_data)

shape= (mean(CO2_data))^2/var(CO2_data)
shape

scale <- exp(5.885)/shape
scale
hist(CO2_data,prob=TRUE)
xseq = seq(280,450, len=1000)
dgamma(xseq,shape=shape, scale= scale)
lines(xseq,dgamma(xseq,shape=shape,scale=scale),col="red")
```

```
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)
xax
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")
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "grey")
abline(v = ISOdate(1991, 11, 1, tz = "UTC"), col = "green")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "purple")
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "pink")
abline(v = ISOdate(2003, 1, 1, tz = "UTC"), col = "orange")


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"))
abline(v = ISOdate(2010, 8, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(2009, 12, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(2009, 10, 1, tz = "UTC"), col = "grey")
abline(v = ISOdate(2010, 5, 20, tz = "UTC"), col = "grey")
```

```
abline(v = ISOdate(2010, 10, 1, tz = "UTC"), col = "grey")

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

abline(v = ISOdate(2020, 1, 1, tz = "UTC"), col = "red")
```

Q2:

```
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.),
         ]
xSub$Date
weekValues = seq(min(xSub$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$day
xSub$Date

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 +
    #random slope
    f(week, model='rw2',
      constr=FALSE,
      prior='pc.prec',
      param = c(0.1/(52*100), 0.05)) +
    #random intercept
```

```
        f(weekIid, model='iid',
            prior='pc.prec',
            param = c(1, 0.5)) +
        #random intercept
        f(yearFac, model='iid', prior='pc.prec',
            param = c(1, 0.5)),
    family='T',
    # Vi
    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)


#histogram
Temp_data = na.omit(xSub$Max.Temp...C.)
hist(Temp_data,prob=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"))

sableRes$summary.random$week
weekValues[-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 = "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"))
weekSample
```