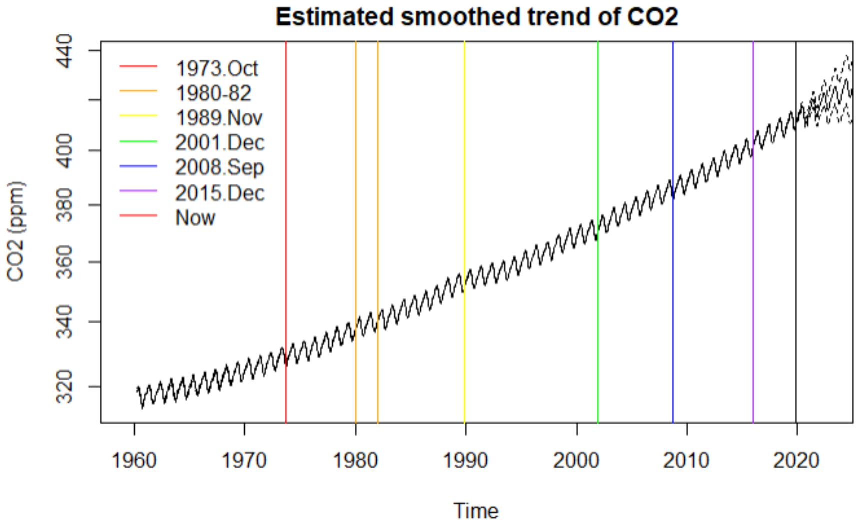
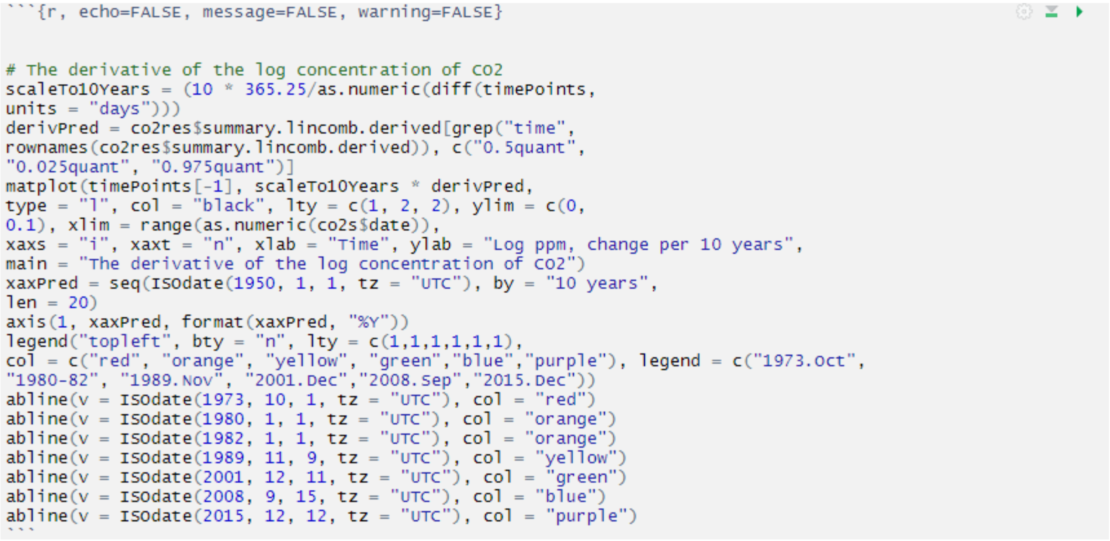
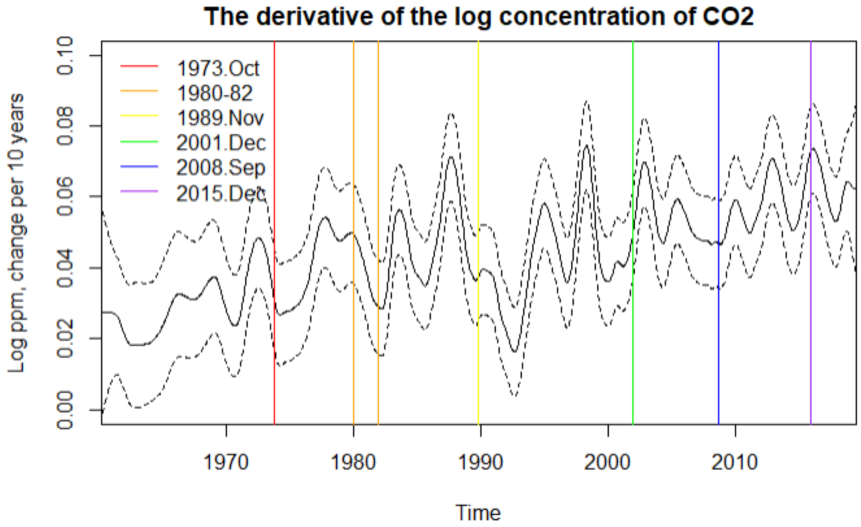
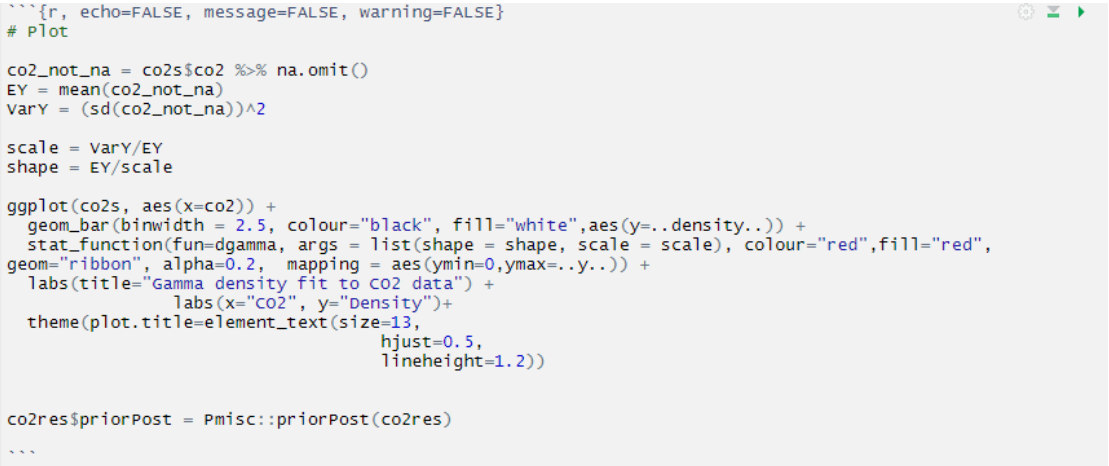


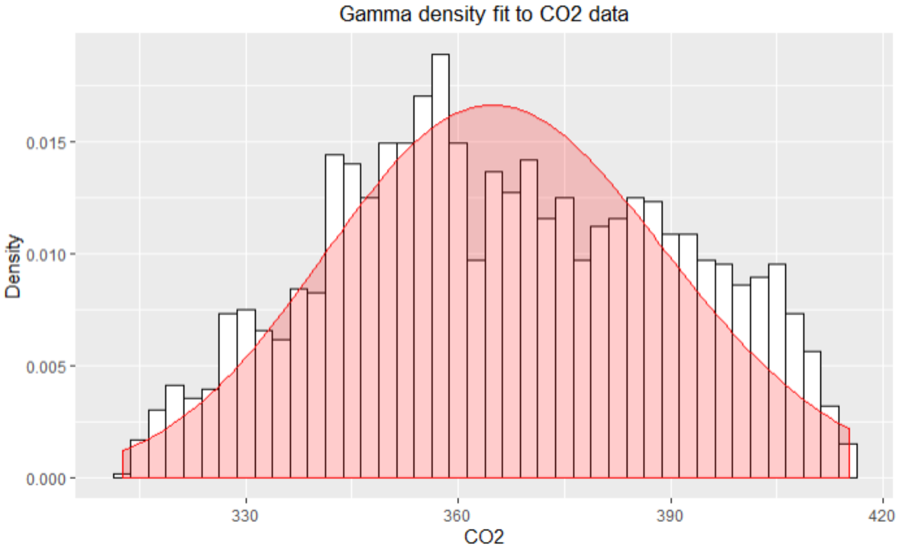


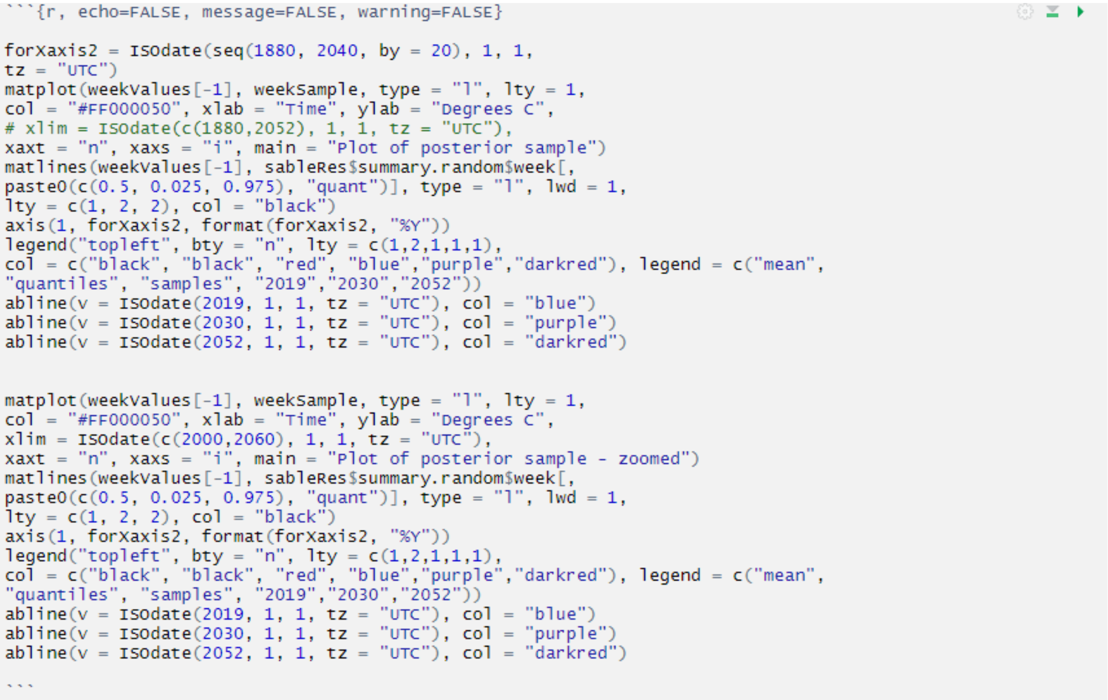
~~~
```{r, echo=FALSE, message=FALSE, warning=FALSE}


# The derivative of the log concentration of CO2
scaleTo10Years = (10 * 365.25/as.numeric(diff(timePoints,
units = "days")))
derivPred = co2res$summary.lincomb.derived[grep("time",
rownames(co2res$summary.lincomb.derived)), c("0.5quant",
"0.025quant", "0.975quant")]
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 10 years",
main = "The derivative of the log concentration of CO2")
xaxPred = seq(ISOdate(1950, 1, 1, tz = "UTC"), by = "10 years",
len = 20)
axis(1, xaxPred, format(xaxPred, "%Y"))
legend("topleft", bty = "n", lty = c(1,1,1,1,1,1),
col = c("red", "orange", "yellow", "green","blue","purple"), legend = c("1973.Oct",
"1980-82", "1989.Nov", "2001.Dec","2008.Sep","2015.Dec"))
abline(v = ISOdate(1973, 10, 1, tz = "UTC"), col = "red")
abline(v = ISOdate(1980, 1, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(1982, 1, 1, tz = "UTC"), col = "orange")
abline(v = ISOdate(1989, 11, 9, tz = "UTC"), col = "yellow")
abline(v = ISOdate(2001, 12, 11, tz = "UTC"), col = "green")
abline(v = ISOdate(2008, 9, 15, tz = "UTC"), col = "blue")
abline(v = ISOdate(2015, 12, 12, tz = "UTC"), col = "purple")
```
~~~



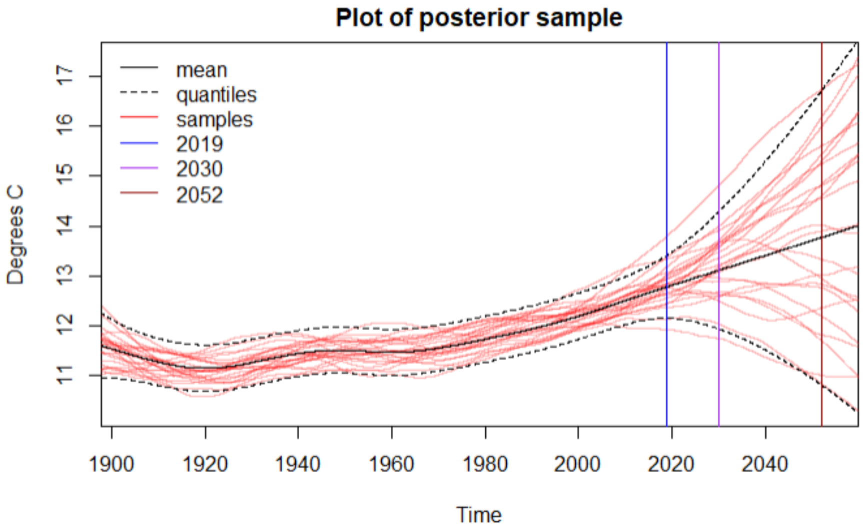


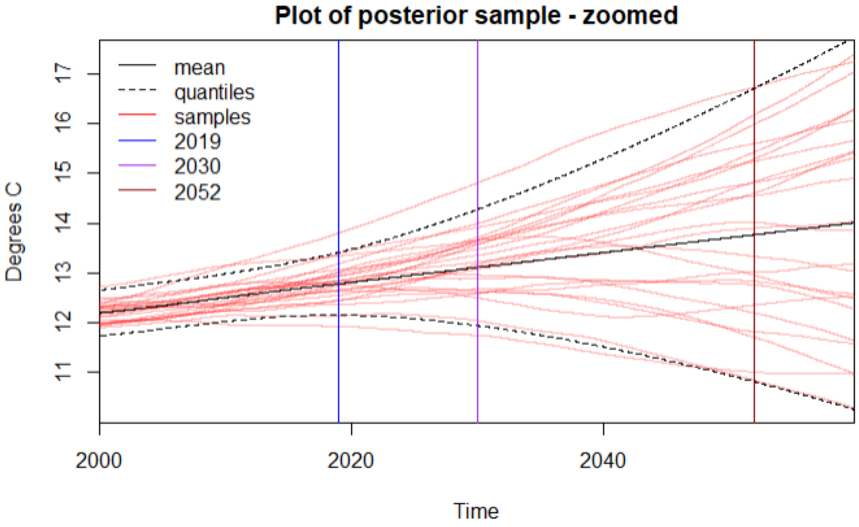
~~~
```{r, echo=FALSE, message=FALSE, warning=FALSE}
# Plot

co2_not_na = co2s$co2 %>% na.omit()
EY = mean(co2_not_na)
VarY = (sd(co2_not_na))^2

scale = VarY/EY
shape = EY/scale

ggplot(co2s, aes(x=co2)) +
  geom_bar(binwidth = 2.5, colour="black", fill="white",aes(y=..density..)) +
  stat_function(fun=dgamma, args = list(shape = shape, scale = scale), colour="red",fill="red",
geom="ribbon", alpha=0.2,  mapping = aes(ymin=0,ymax=..y..)) +
  labs(title="Gamma density fit to CO2 data") +
                labs(x="CO2", y="Density")+
  theme(plot.title=element_text(size=13,
                                    hjust=0.5,
                                    lineheight=1.2))


co2res$priorPost = Pmisc::priorPost(co2res)

```
~~~

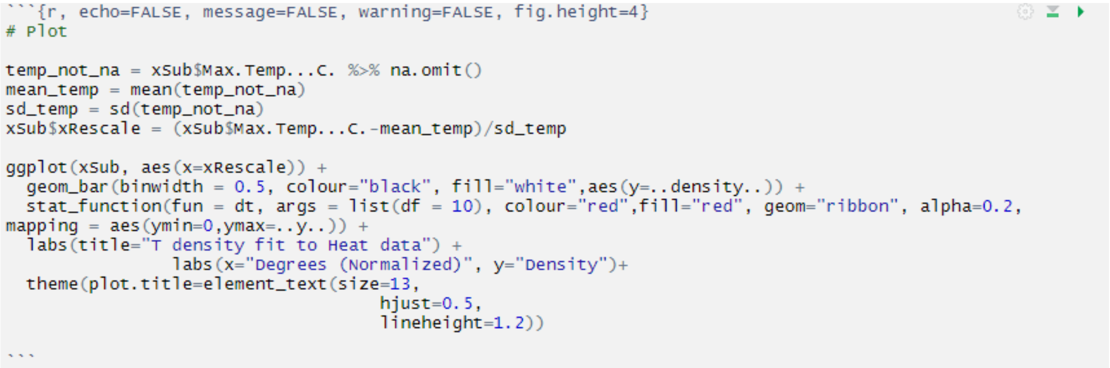


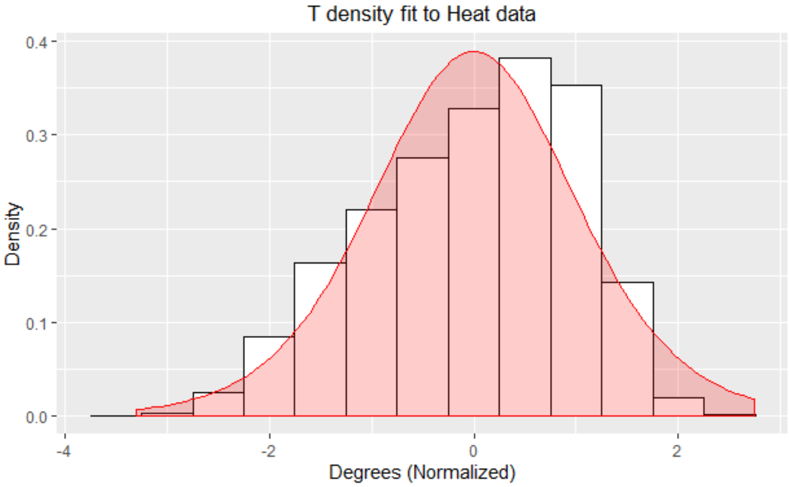


~~~
```{r, echo=FALSE, message=FALSE, warning=FALSE}

forxaxis2 = ISOdate(seq(1880, 2040, by = 20), 1, 1,
tz = "UTC")
matplot(weekValues[-1], weekSample, type = "l", lty = 1,
col = "#FF000050", xlab = "Time", ylab = "Degrees C",
# xlim = ISOdate(c(1880,2052), 1, 1, tz = "UTC"),
xaxt = "n", xaxs = "i", main = "Plot of posterior sample")
matlines(weekValues[-1], sableRes$summary.random$week[,
paste0(c(0.5, 0.025, 0.975), "quant")], type = "l", lwd = 1,
lty = c(1, 2, 2), col = "black")
axis(1, forxaxis2, format(forxaxis2, "%Y"))
legend("topleft", bty = "n", lty = c(1,2,1,1,1,1),
col = c("black", "black", "red", "blue","purple","darkred"), legend = c("mean",
"quantiles", "samples", "2019","2030","2052"))
abline(v = ISOdate(2019, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(2030, 1, 1, tz = "UTC"), col = "purple")
abline(v = ISOdate(2052, 1, 1, tz = "UTC"), col = "darkred")


matplot(weekValues[-1], weekSample, type = "l", lty = 1,
col = "#FF000050", xlab = "Time", ylab = "Degrees C",
xlim = ISOdate(c(2000,2060), 1, 1, tz = "UTC"),
xaxt = "n", xaxs = "i", main = "Plot of posterior sample - zoomed")
matlines(weekValues[-1], sableRes$summary.random$week[,
paste0(c(0.5, 0.025, 0.975), "quant")], type = "l", lwd = 1,
lty = c(1, 2, 2), col = "black")
axis(1, forxaxis2, format(forxaxis2, "%Y"))
legend("topleft", bty = "n", lty = c(1,2,1,1,1,1),
col = c("black", "black", "red", "blue","purple","darkred"), legend = c("mean",
"quantiles", "samples", "2019","2030","2052"))
abline(v = ISOdate(2019, 1, 1, tz = "UTC"), col = "blue")
abline(v = ISOdate(2030, 1, 1, tz = "UTC"), col = "purple")
abline(v = ISOdate(2052, 1, 1, tz = "UTC"), col = "darkred")

```
~~~






~~~
```{r, echo=FALSE, message=FALSE, warning=FALSE, fig.height=4}
# Plot

temp_not_na = xSub$Max.Temp...C. %>% na.omit()
mean_temp = mean(temp_not_na)
sd_temp = sd(temp_not_na)
xSub$xRescale = (xSub$Max.Temp...C.-mean_temp)/sd_temp

ggplot(xSub, aes(x=xRescale)) +
  geom_bar(binwidth = 0.5, colour="black", fill="white",aes(y=..density..)) +
  stat_function(fun = dt, args = list(df = 10), colour="red",fill="red", geom="ribbon", alpha=0.2,
mapping = aes(ymin=0,ymax=..y..)) +
  labs(title="T density fit to Heat data") +
              labs(x="Degrees (Normalized)", y="Density")+
  theme(plot.title=element_text(size=13,
                                hjust=0.5,
                                lineheight=1.2))

```
~~~


T density fit to Heat data
Density
Degrees (Normalized)