STA 442 A3 Q2

Insu Yoon, 100318093

Introduction

We investigate the mortality in Montreal, Quebec after the COVID-19 pandemic started. In this paper, we will test the hypothesis that there is an increase in deaths in the under 50's in post-covid, whereas there is no change in deaths in the over 70's compared to pre-covid.

Methods

$$Y_i \sim \text{Poisson}(O_i \lambda_i)$$

$$log[\lambda_i] = X_i\beta + U(t_i) + V_i$$

$$[U_1 \dots U_T]^T \sim RW2(0, \sigma_U^2)$$

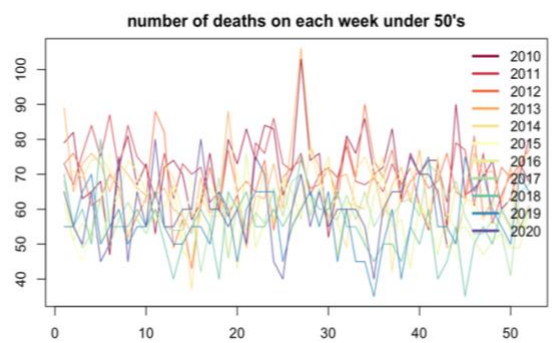
$$V_i \sim N(0, \sigma_V^2)$$

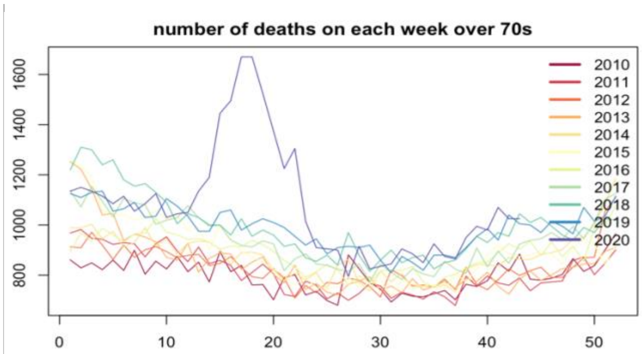
- X_{i0}= 1
- X_{i1}=$sin(2\pi t_i/365.25)$
- X_{i2}=$cos(2\pi t_i/365.25)$
- X_{i3}=$sin(2\pi t_i/182.625)$
- X_{i4}=$cos(2\pi t_i/182.625)$
- U(t) is a second-order random walk
- …second derivatives are $N(0, \sigma_U^2)$
- V_i covers independent variation or over-dispersion

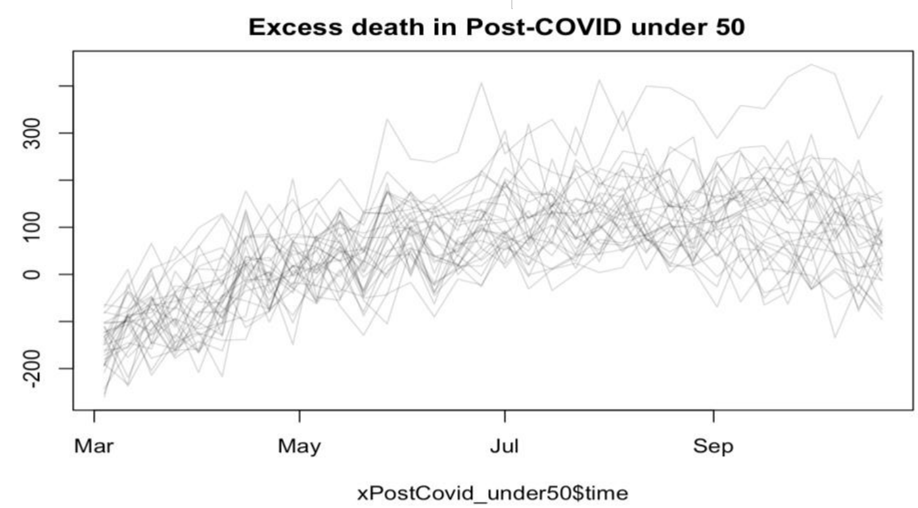
Result

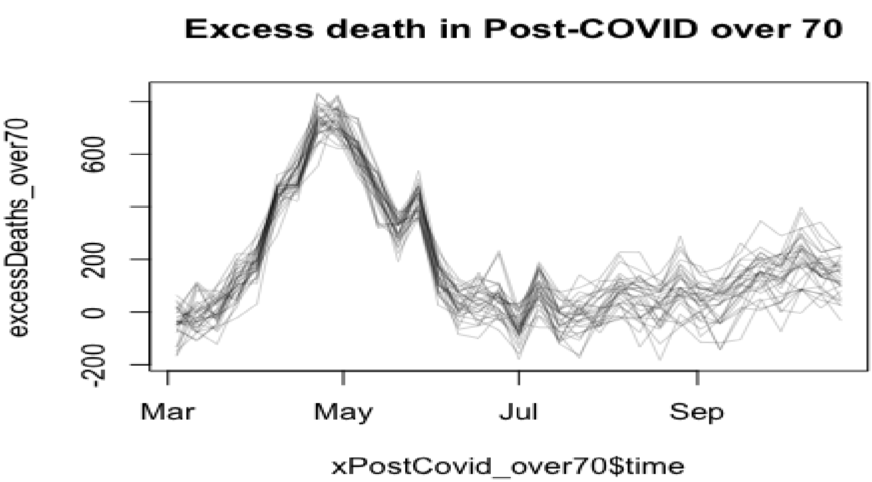
First, when we look at the number of deaths on each week for two age groups that the plot of under 50's has no outstanding number of deaths on 2020 even after COVID-19 which means that we do not have sufficient evidence that there is an impact of COVID-19. Moreover, we also calculate the excess deaths of under 50's after covid pandemic, which shows a slight increasing trend of number of deaths but there is no significant change.

However, over 70’s has huge increase in number of deaths in 2020 and the excess deaths is over 600 at the early covid stage. Therefore, we also reject our hypothesis that there is no impact on number of deaths for over 70’s.



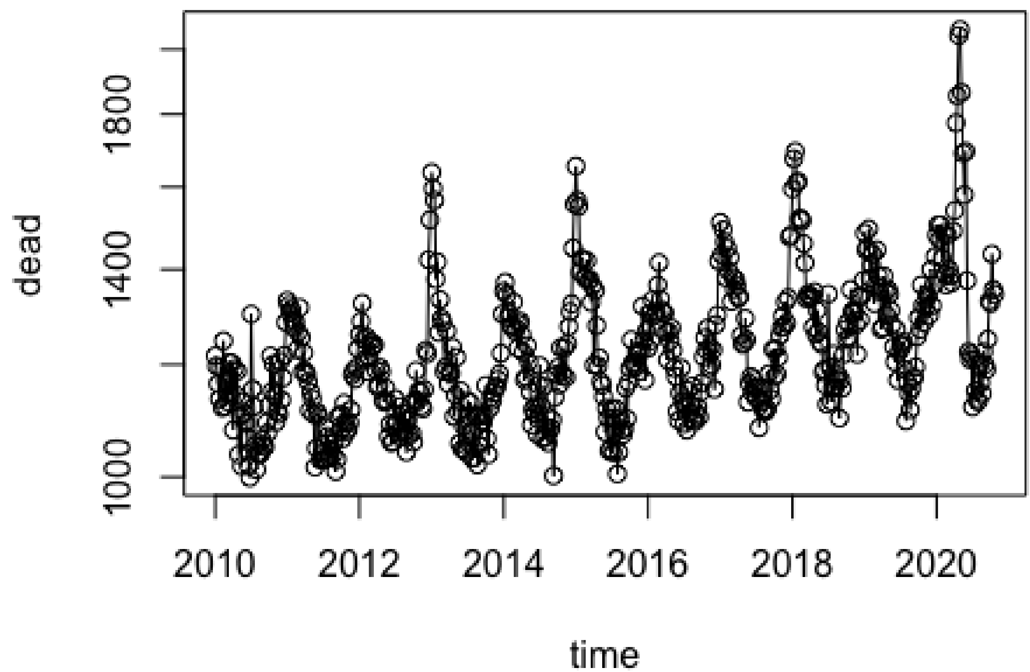




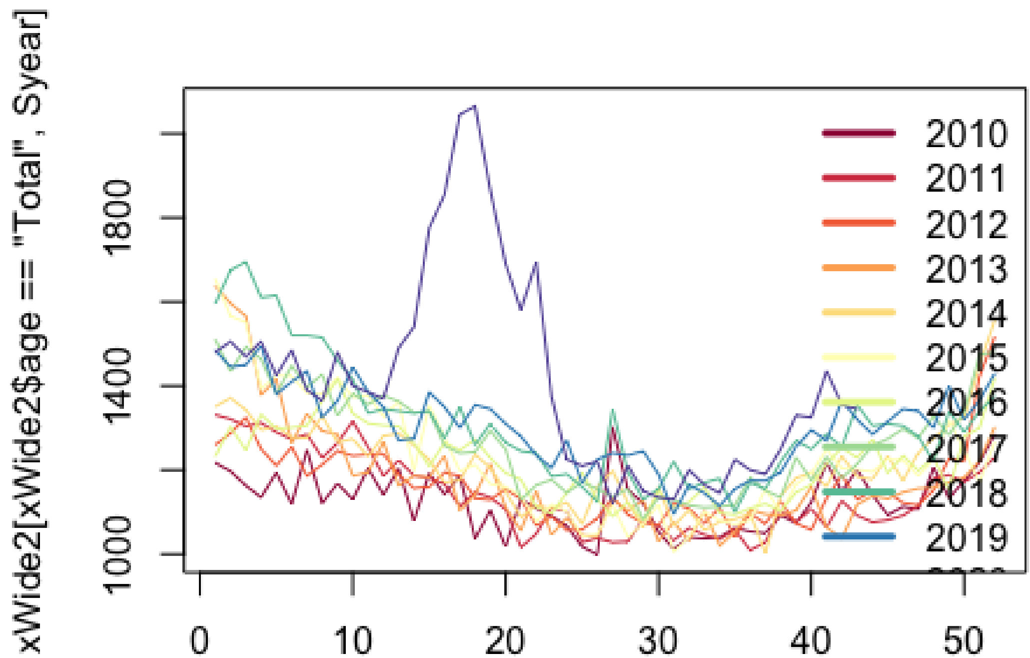


Appendix

```
xWide = read.table(paste0("https://www.stat.gouv.qc.ca/statistiques/",
"population-demographie/deces-mortalite/", "WeeklyDeaths_QC_2010-
2020_AgeGr.csv"), sep = ";", skip = 7, col.names = c("year", "junk",
"age", paste0("w", 1:53)))
xWide = xWide[grep("^[[:digit:]]+$", xWide$year), ]

## Warning in grep("^[[:digit:]]+$", xWide$year): input string 47 is invalid
in
## this locale

## Warning in grep("^[[:digit:]]+$", xWide$year): input string 50 is invalid
in
## this locale

## Warning in grep("^[[:digit:]]+$", xWide$year): input string 59 is invalid
in
## this locale

x = reshape2::melt(xWide, id.vars = c("year", "age"),
measure.vars = grep("^w[[:digit:]]+$", colnames(xWide)))
x$dead = as.numeric(gsub("[[:space:]]", "", x$value))
x$week = as.numeric(gsub("w", "", x$variable))
x$year = as.numeric(x$year)
x = x[order(x$year, x$week, x$age), ] #convert the ‘week’ variable to time
newYearsDay = as.Date(ISOdate(x$year, 1, 1))
x$time = newYearsDay + 7 * (x$week - 1)
x = x[!is.na(x$dead), ]
x = x[x$week < 53, ]

plot(x[x$age == "Total", c("time", "dead")], type = "o", log = "y") # we get
bigger spikes in the past with the flue vaccine. So these are often elderly
people who have all sorts of other health problems.
```




```
xWide2 = reshape2::dcast(x, week + age ~ year, value.var = "dead")
Syear = grep("[[:digit:]]", colnames(xWide2), value = TRUE)
Scol = RColorBrewer::brewer.pal(length(Syear), "Spectral")
matplot(xWide2[xWide2$age == "Total", Syear], type = "l",
lty = 1, col = Scol)
legend("topright", col = Scol, legend = Syear, bty = "n",
lty = 1, lwd = 3)
```




```
xWide3 = reshape2::dcast(x, week + age ~ year, value.var = "dead")
Syear = grep("[[:digit:]]", colnames(xWide3), value = TRUE)
Scol = RColorBrewer::brewer.pal(length(Syear), "Spectral")
matplot(xWide3[xWide3$age == "0-49 years old", Syear], type = "l",
lty = 1, col = Scol)
legend("topright", col = Scol, legend = Syear, bty = "n",
lty = 1, lwd = 3)
```

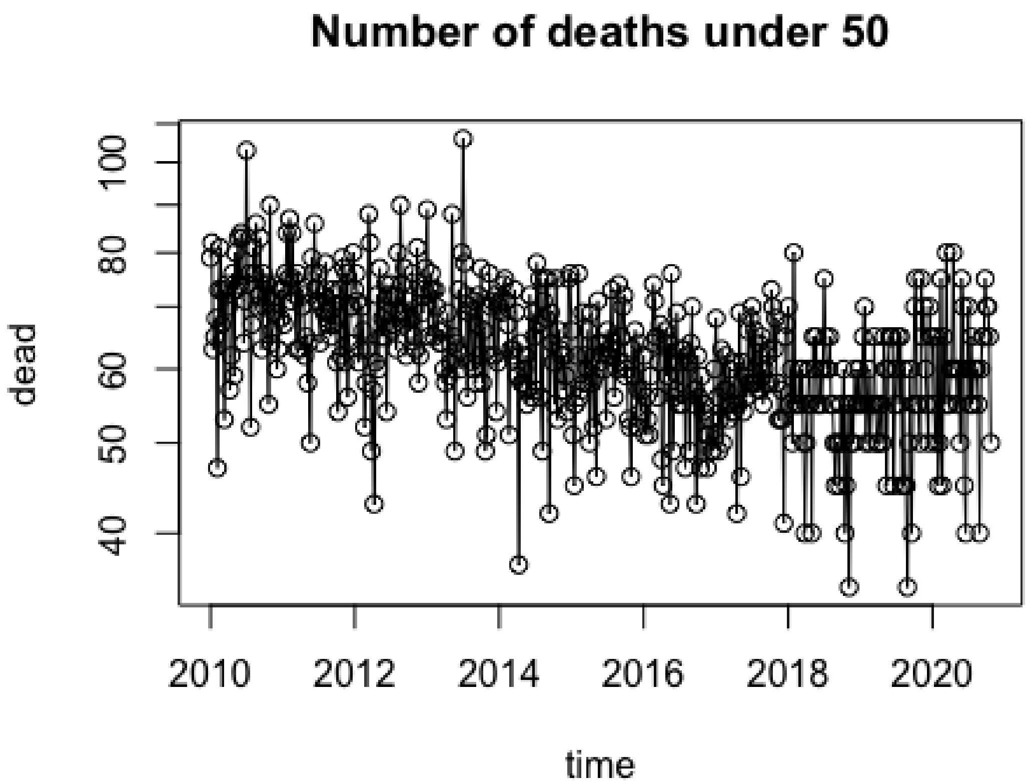
```
xWide4 = reshape2::dcast(x, week + age ~ year, value.var = "dead")
Syear = grep("[[:digit:]]", colnames(xWide4), value = TRUE)
Scol = RColorBrewer::brewer.pal(length(Syear), "Spectral")
matplot(xWide3[xWide4$age == "70 years old and over", Syear], type = "l",
lty = 1, col = Scol)
legend("topright", col = Scol, legend = Syear, bty = "n",
lty = 1, lwd = 3)
```

Divide the data into pre and post covid, add extra dates to data so that INLA will create forecasts.

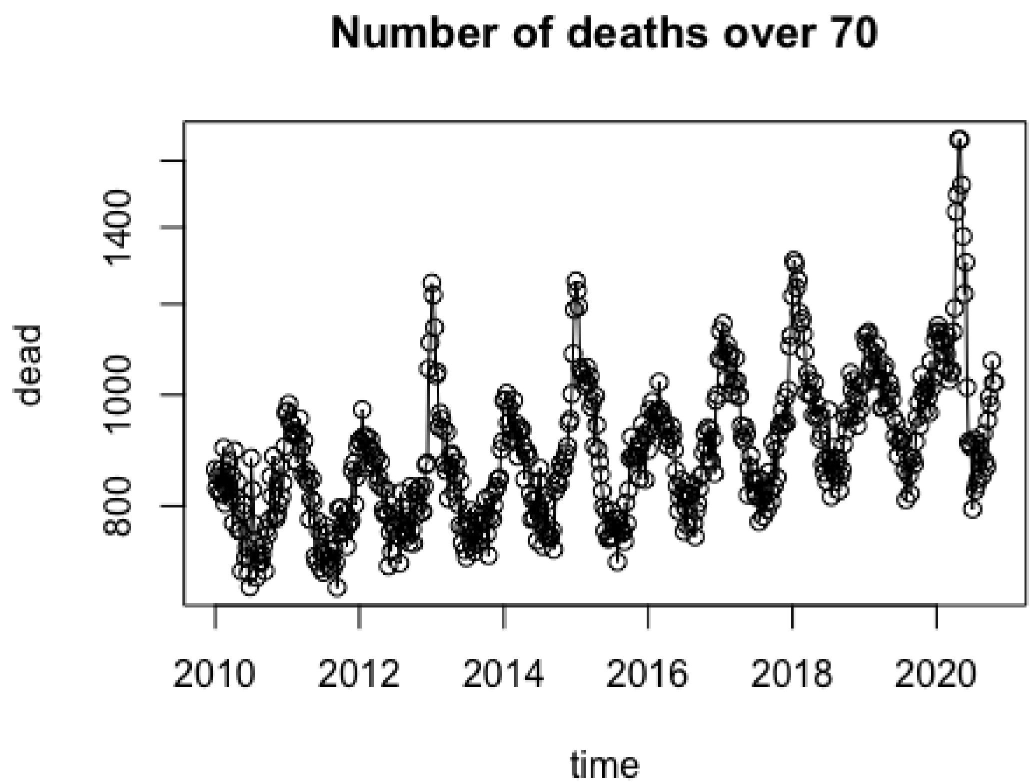
```r
dateCutoff = as.Date("2020/3/1")
xPreCovid = x[x$time < dateCutoff, ]
xPostCovid = x[x$time >= dateCutoff, ]
toForecast = expand.grid(age = unique(x$age), time = unique(xPostCovid$time),
dead = NA)
xForInla = rbind(xPreCovid[, colnames(toForecast)],toForecast)
xForInla = xForInla[order(xForInla$time, xForInla$age),]

xForInla$timeNumeric = as.numeric(xForInla$time)
xForInla$timeForInla = (xForInla$timeNumeric -
as.numeric(as.Date("2015/1/1")))/365.25
xForInla$timeIid = xForInla$timeNumeric
xForInla$sin12 = sin(2 * pi * xForInla$timeNumeric/365.25)
xForInla$sin6 = sin(2 * pi * xForInla$timeNumeric *
2/365.25)
xForInla$cos12 = cos(2 * pi * xForInla$timeNumeric/365.25)
xForInla$cos6 = cos(2 * pi * xForInla$timeNumeric *2/365.25)
xForInlaTotal= xForInla[xForInla$age == 'Total', ]
xForInla_under50 = xForInla[xForInla$age == '0-49 years old',]
xForInla_over70 = xForInla[xForInla$age == '70 years old and over',]
library(INLA, verbose=FALSE)

## Loading required package: Matrix

## Loading required package: sp

## Loading required package: parallel

## Loading required package: foreach

## This is INLA_20.03.17 built 2020-10-27 02:19:26 UTC.
## See www.r-inla.org/contact-us for how to get help.
## To enable PARDISO sparse library; see inla.pardiso()

res = inla(dead ~ sin12 + sin6 + cos12 + cos6 +
f(timeIid, prior='pc.prec', param= c(log(1.2), 0.5)) + f(timeForInla, model =
'rw2', prior='pc.prec', param= c(0.01, 0.5)),
data=xForInla_over70,
control.predictor = list(compute=TRUE, link=1), control.compute =
list(config=TRUE),
# control.inla = list(fast=FALSE, strategy='laplace'),
family='poisson')
qCols = paste0(c(0.5, 0.025, 0.975), "quant")
rbind(res$summary.fixed[, qCols], Pmisc::priorPostSd(res)$summary[,qCols])

##                      0.5quant  0.025quant 0.975quant
## (Intercept)        6.78520154 6.777584946 6.79251442
## sin12              0.06367217 0.056174012 0.07121581
## sin6               0.01138859 0.004902564 0.01786384
## cos12              0.11701052 0.109481282 0.12460255
## cos6               0.01182356 0.005329866 0.01829595
```

```
## SD for timeIid     0.04050370 0.036336243 0.04504260
## SD for timeForInla 0.16066285 0.106910730 0.23982926
```

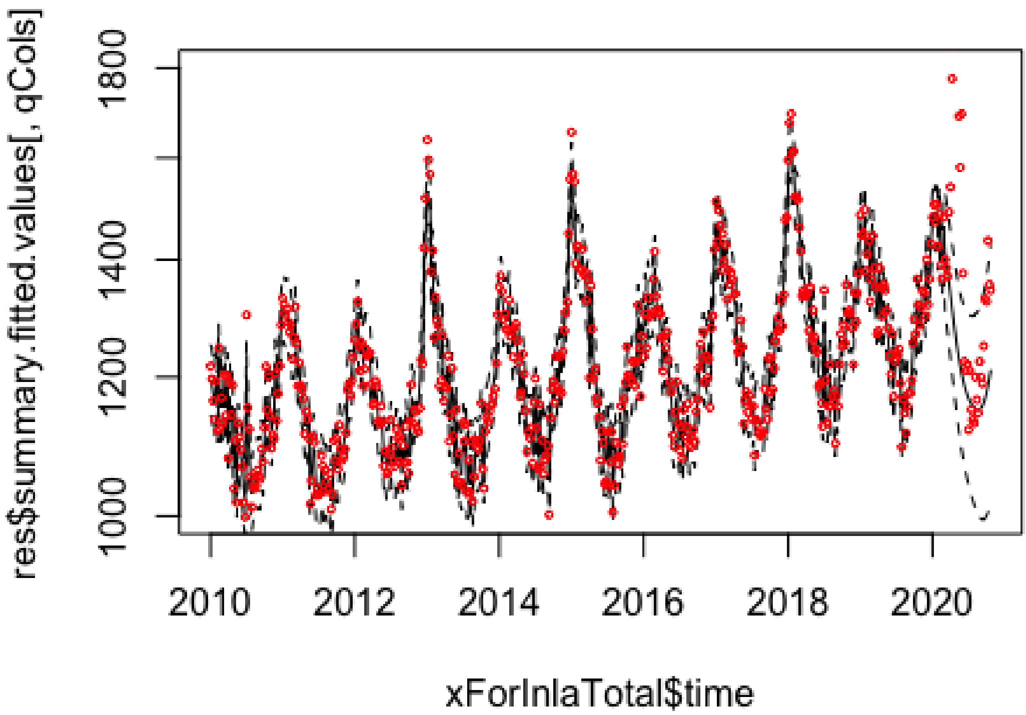
```
plot(x[x$age == "0-49 years old", c("time", "dead")], type = "o", log=
"y",main = "Number of deaths under 50")
```



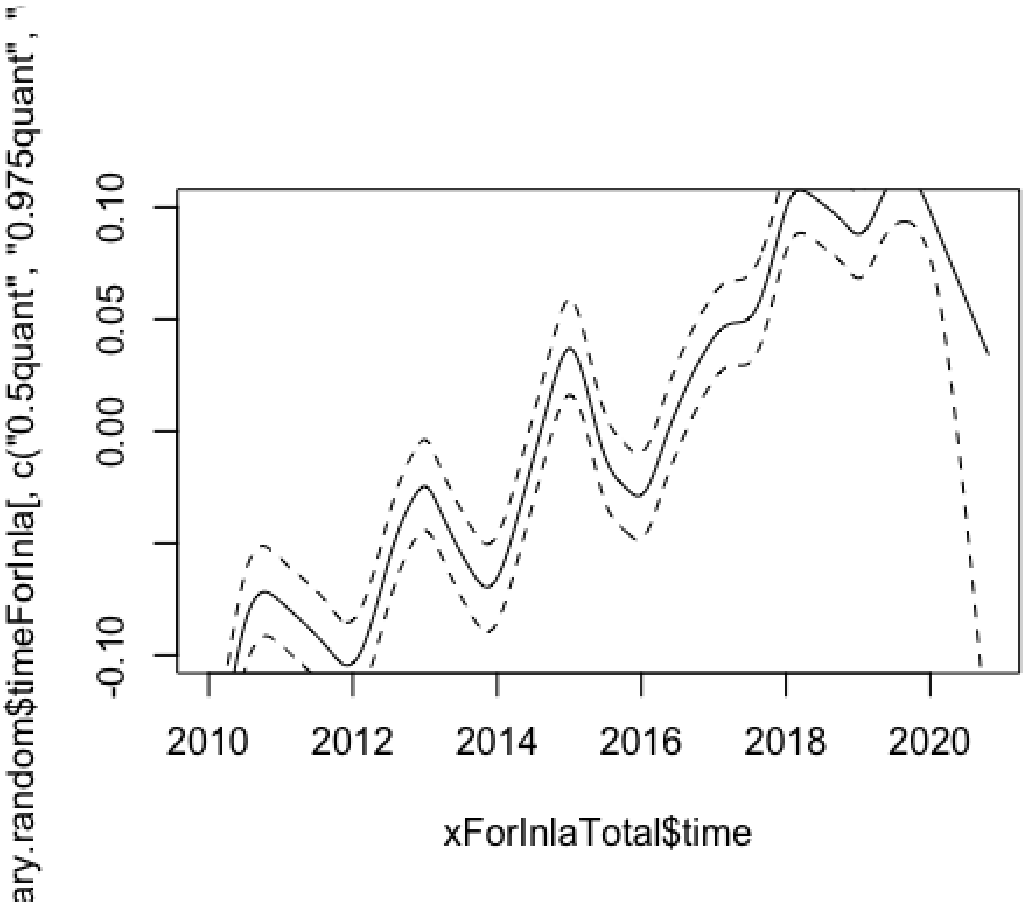

```
plot(x[x$age == "70 years old and over", c("time", "dead")], type = "o", log=
"y",main = "Number of deaths over 70")
```




```r
matplot(xForInlaTotal$time, res$summary.fitted.values[,
qCols], type= "l", ylim = c(1000, 1800), lty = c(1,2, 2), col =  "black", log
= "y")
points(x[x$age == "Total", c("time", "dead")], cex = 0.4,col = "red")
```




```
matplot(xForInlaTotal$time, res$summary.random$timeForInla[, c("0.5quant",
"0.975quant", "0.025quant")], type = "l", lty = c(1, 2, 2), col = "black",
ylim = c(-1, 1) *
0.1) # this is after taking out seasonality
```

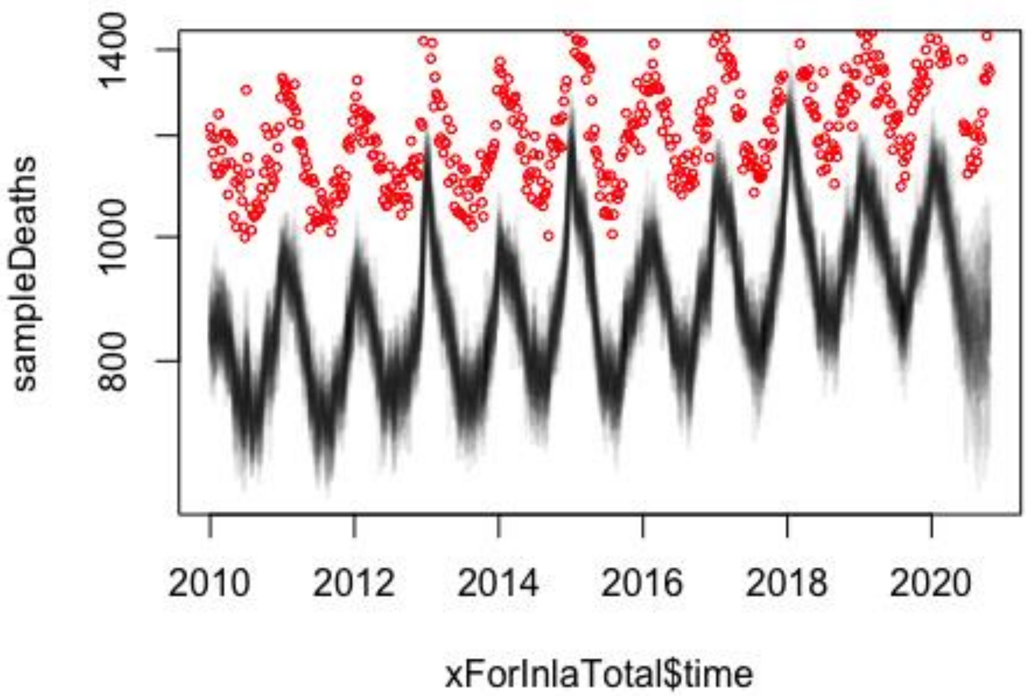


Take posterior samples of the intensity

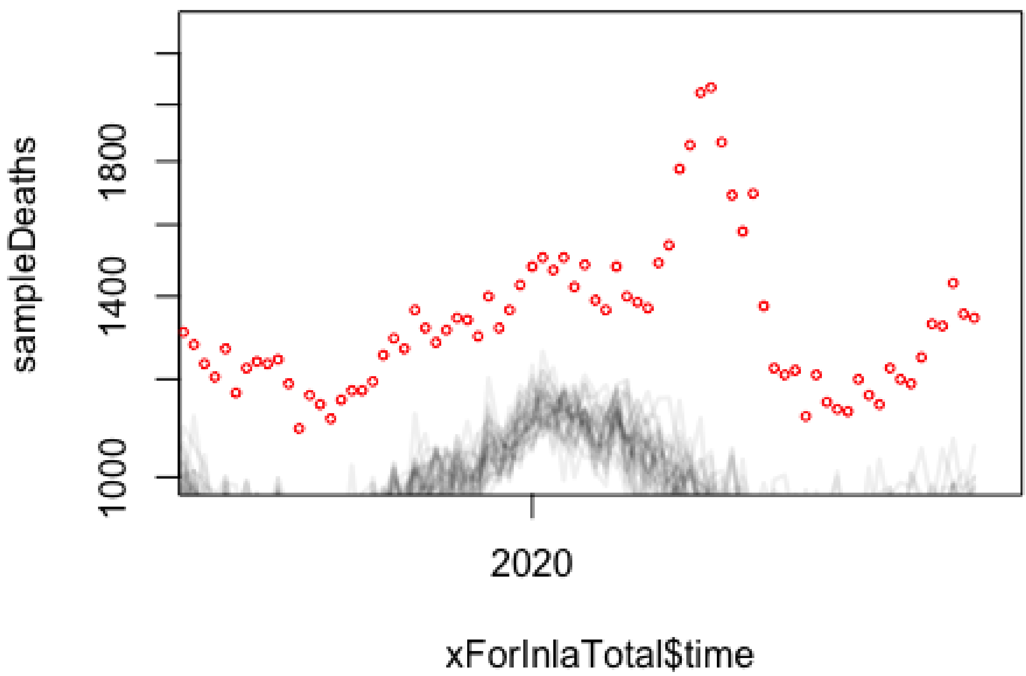
```
# get posterior samples
sampleList = INLA::inla.posterior.sample(30, res, selection = list(Predictor
= 0)) # prdictors = 0 is samples of log(\lmabda(t))  turning that into a
matrix
sampleIntensity = exp(do.call(cbind,
Biobase::subListExtract(sampleList,"latent"))) # and exponentiate it them to
put them into natural sclae
sampleDeaths = matrix(rpois(length(sampleIntensity), #generating deaths from
poisson distribution
sampleIntensity), nrow(sampleIntensity), ncol(sampleIntensity))
```

plot samples and real data

```
matplot(xForInlaTotal$time, sampleDeaths, col = "#00000010", lwd = 2, lty =
1, type = "l", log = "y") # graph of posterior samples
points(x[x$age == "Total", c("time", "dead")], col = "red", cex = 0.5) #
actual datas
```




```
matplot(xForInlaTotal$time, sampleDeaths, col = "#00000010",
lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1",
"2020/11/1")), ylim = c(1, 2.3) * 1000)
points(x[x$age == "Total", c("time", "dead")], col = "red", cex = 0.5) #
these red dots are real deaths after the covid.
```



calculate excess deaths

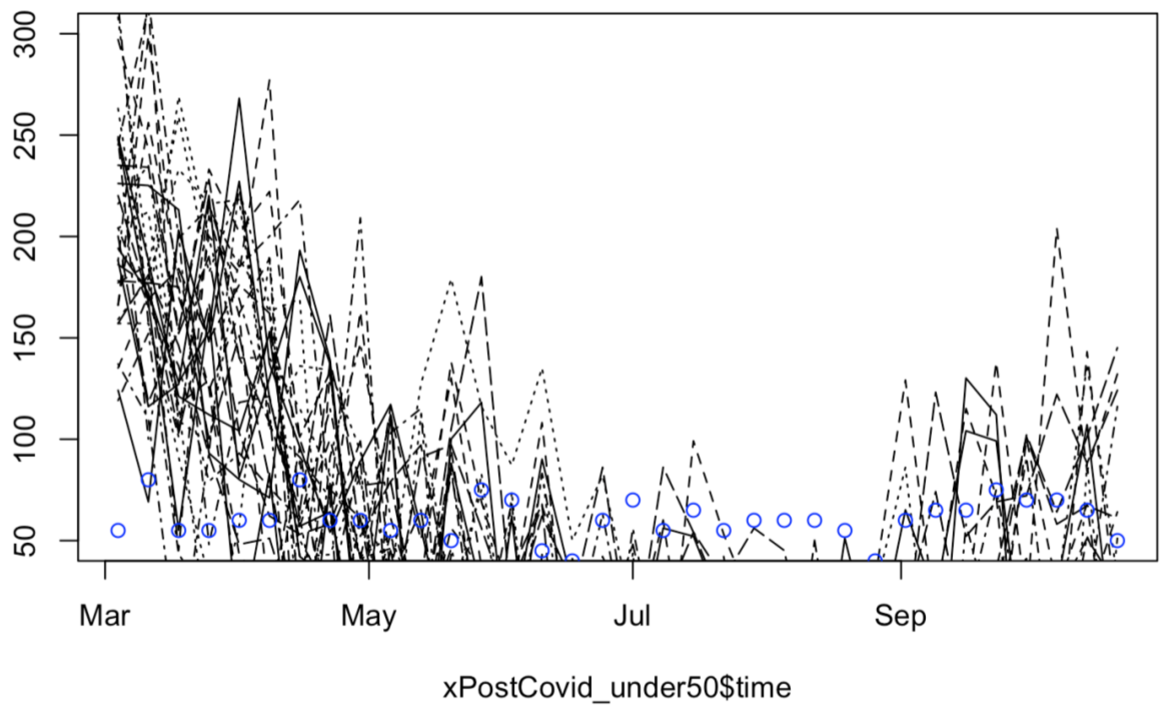
```
xPostCovid_under50 = xPostCovid[xPostCovid$age == '0-49 years old', ] # this
is number of deaths each week
xPostCovidForecast_under50 = sampleDeaths[match(xPostCovid_under50$time,
xForInla_under50$time), ]#5 posterior samples of what we would have expected.
excessDeaths_under50 = xPostCovid_under50$dead - xPostCovidForecast_under50
xPostCovid_over70 = xPostCovid[xPostCovid$age == '70 years old and over', ] #
this is number of deaths each week
xPostCovidForecast_over70 = sampleDeaths[match(xPostCovid_over70$time,
xForInlaTotal$time), ]#5 posterior samples of what we would have expected.
excessDeaths_over70 = xPostCovid_over70$dead - xPostCovidForecast_over70
xPostCovidTotal = xPostCovid[xPostCovid$age == 'Total', ] # this is number of
deaths each week
xPostCovidForecast = sampleDeaths[match(xPostCovidTotal$time,
xForInlaTotal$time), ]#5 posterior samples of what we would have expected.
excessDeaths = xPostCovidTotal$dead - xPostCovidForecast
```

plot samples of excess deaths

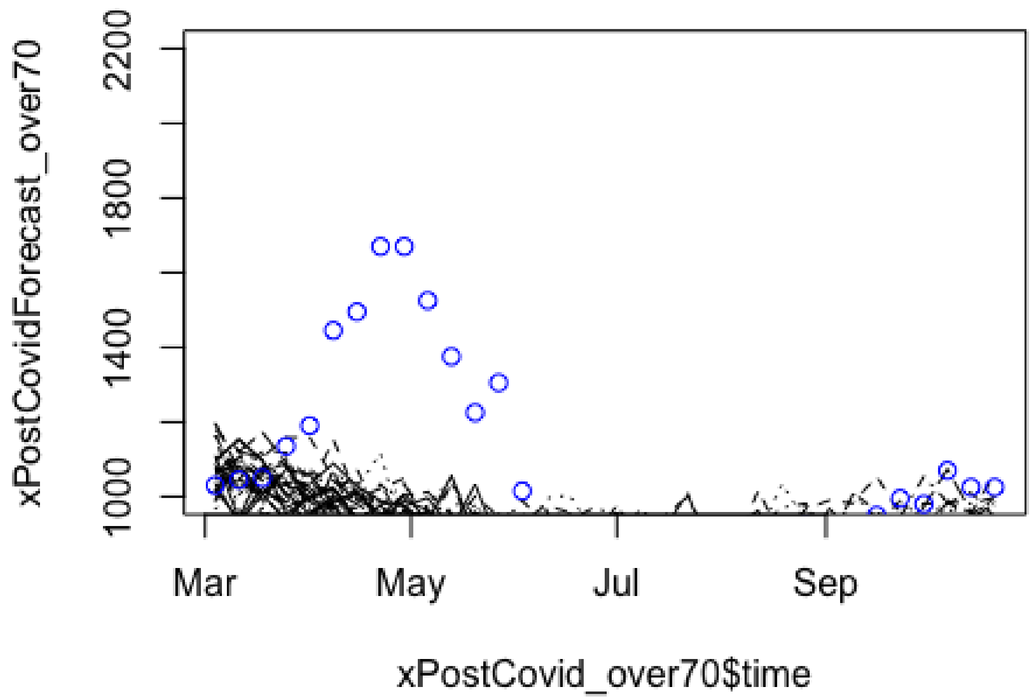
```
matplot(xPostCovid_under50$time, xPostCovidForecast_under50, type = "l", ylim
= c(1000, 2200), col = "black")
points(xPostCovid_under50[, c("time", "dead")], col = "blue")  # so what is
```

```
the difference between red and the black that is the next graph # the black
line represents the covid free world  and red is actual death
```

```
matplot(xPostCovid_under50$time, excessDeaths_under50, type = "l",lty = 1,
col = "#00000030", main =" Excess death in Post-COVID under 50")
```

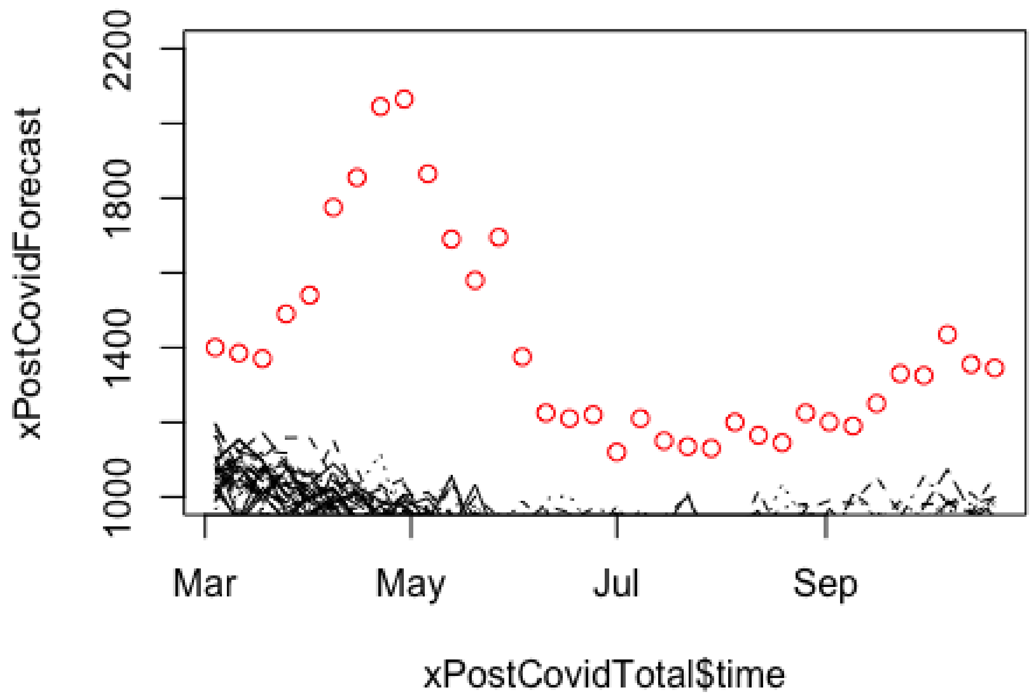



```
matplot(xPostCovid_over70$time, xPostCovidForecast_over70, type = "l", ylim =
c(1000, 2200), col = "black")
points(xPostCovid_over70[, c("time", "dead")], col = "blue")  # so what is
the difference between red and the black that is the next graph # the black
line represents the covid free world  and red is actual death
```

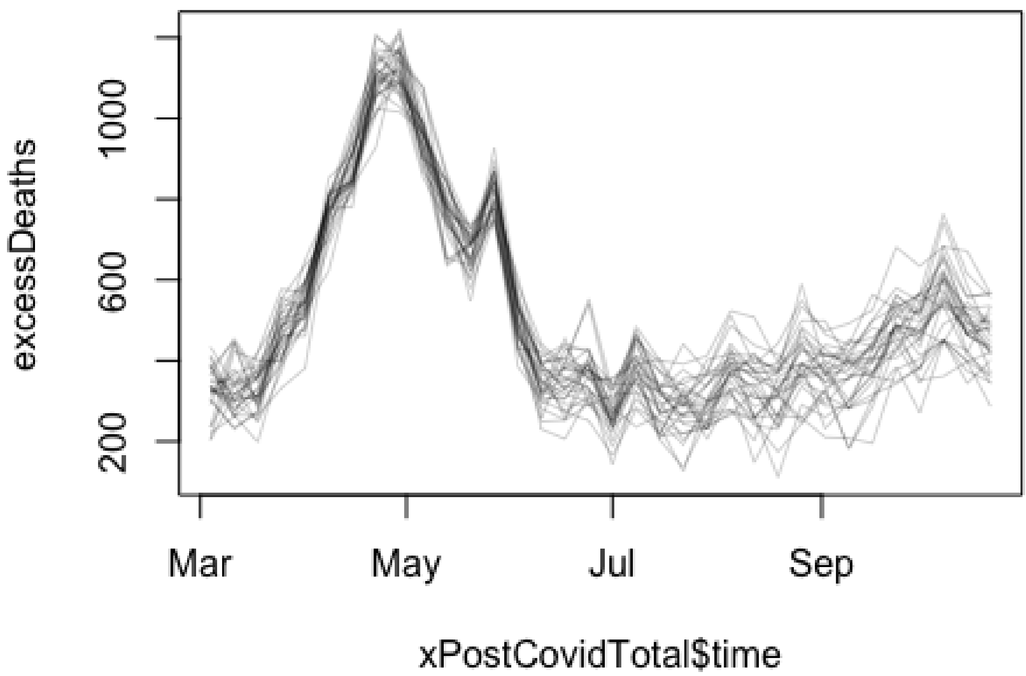



```r
matplot(xPostCovid_over70$time, excessDeaths_over70, type = "l",lty = 1, col
= "#00000030", main =" Excess death in Post-COVID over 70")
```

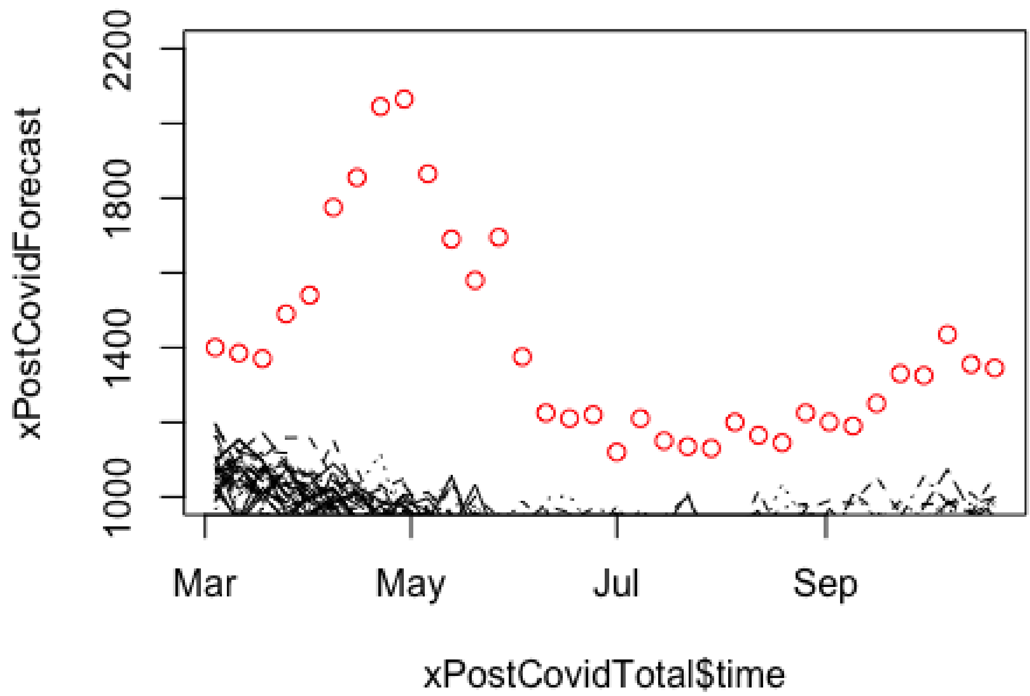
```r
matplot(xPostCovidTotal$time, xPostCovidForecast, type = "l", ylim = c(1000,
2200), col = "black")
points(xPostCovidTotal[, c("time", "dead")], col = "red")  # so what is the
difference between red and the black that is the next graph # the black line
represents the covid free world  and red is actual death
```



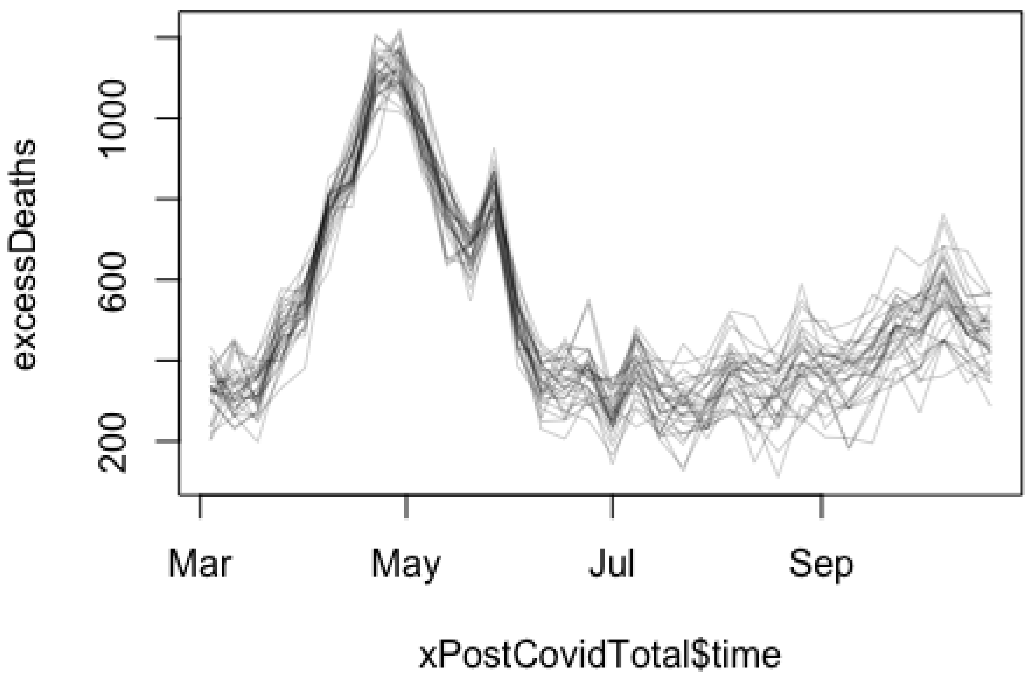

```
matplot(xPostCovidTotal$time, excessDeaths, type = "l",lty = 1, col =
"#00000030")
```




```
matplot(xPostCovidTotal$time, xPostCovidForecast, type = "l", ylim = c(1000,
2200), col = "black")
points(xPostCovidTotal[, c("time", "dead")], col = "red")  # so what is the
difference between red and the black that is the next graph # the black line
represents the covid free world  and red is actual death
```




```
matplot(xPostCovidTotal$time, excessDeaths, type = "l",lty = 1, col =
"#00000030")
```



excessDeaths
200
600
1000
Mar
May
Jul
Sep
xPostCovidTotal$time