Appendix: Graphs and Tables

```
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), ]
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), ]

newYearsDay = as.Date(ISOdate(x$year + 2008, 1, 1))
x$time = newYearsDay + 7 * (x$week - 1)
x = x[!is.na(x$dead), ]
x = x[x$week < 53, ]
# fit the model using pre covid data
dateCutoff = as.Date("2020/3/1")
xPreCovid = x[x$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)

xForInlaUnder50= xForInla[xForInla$age == '0-49 years old', ]
library(INLA, verbose=FALSE)

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=xForInlaUnder50,
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")

xForInlaOver70 = xForInla[xForInla$age == '70 years old and over', ]
```

```r
res70 = 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=xForInlaOver70,
control.predictor = list(compute=TRUE, link=1),
control.compute = list(config=TRUE),
# control.inla = list(fast=FALSE, strategy='laplace'),
family='poisson')
```

Graphs

Overviews of sample data

```r
plot(x[x$age == "0-49 years old", c("time", "dead")], type = "o", log = "y",
     ylab = "death counts")
```

```r
plot(x[x$age == "70 years old and over", c("time", "dead")], type = "o", log = "y",
     ylab = "death counts")
```

Graphs of Predicted Death Intensity

```r
# predicted values
forX = as.Date(c("2010/1/1", "2020/1/1"))
forX = seq(forX[1], forX[2], by = "2 years")

matplot(xForInlaUnder50$time, res$summary.fitted.values[, qCols], type = "l",
        ylim = c(30, 110), lty = c(1, 2, 2), col = c("black", "darkgrey", "darkgrey"),
        log = "y", xaxt = "n", ylab = "number of deaths", xlab = "time",
        lwd = c(1, 2, 2))
points(x[x$age == "0-49 years old", c("time", "dead")], cex = 0.4, col = "red")
axis(1, as.numeric(forX), format(forX, "%Y"))
legend(x = as.numeric(forX[1], format = "%Y"), y = 50,
       c("real data", "predicted value", "confidence interval"),
       lty = c(NA, 1, 2), pch = c(20, NA, NA), col=c("red", "black", "darkgrey"),
       lwd = c(NA, 1, 2))

covid1 = as.numeric(as.Date("2020/03/01"), format="%b%Y")
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
text(covid1+200, 100, "after \n covid", col="blue")
```

```r
# predicted values
forX = as.Date(c("2010/1/1", "2020/1/1"))
forX = seq(forX[1], forX[2], by = "2 years")

matplot(xForInlaOver70$time, res70$summary.fitted.values[, qCols], type = "l",
        ylim = c(600, 1700), lty = c(1, 2, 2), col = c("black", "darkgrey", "darkgrey"),
        log = "y", xaxt = "n", ylab = "number of deaths", xlab = "time",
        lwd = c(1, 2, 2))
```

```
points(x[x$age == "70 years old and over", c("time", "dead")], cex = 0.4, col = "red")
axis(1, as.numeric(forX), format(forX, "%Y"))
legend(x = as.numeric(forX[1], format = "%Y"), y = 1700,
       c("real data", "predicted value", "confidence interval"),
       lty = c(NA, 1, 2), pch = c(20, NA, NA), col=c("red", "black", "darkgrey"),
       lwd = c(NA, 1, 2))

covid1 = as.numeric(as.Date("2020/03/01"), format="%b%Y")
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
text(covid1+200, 640, "after \n covid", col="blue")
```

Graphs of Predicted Random Effects

```
# predicted random effects
matplot(xForInlaUnder50$time, res$summary.random$timeForInla[,
c("0.5quant", "0.975quant", "0.025quant")], type = "l",
lty = c(1, 2, 2), col = "black", ylim = c(-2, 2) *
0.1, xaxt = "n", ylab = "predicted random effects", xlab = "time")
axis(1, as.numeric(forX), format(forX, "%Y"))
legend(x = as.numeric(forX[1], format = "%Y"), y = -0.08,
       c("predicted value", "confidence interval"), lty = c(1, 2))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
text(covid1+200, 0.1, "after \n covid", col="blue")
```

```
# predicted random effects
matplot(xForInlaOver70$time, res70$summary.random$timeForInla[,
c("0.5quant", "0.975quant", "0.025quant")], type = "l",
lty = c(1, 2, 2), col = "black", ylim = c(-3, 2) *
0.1, xaxt = "n", ylab = "predicted random effects", xlab = "time")
axis(1, as.numeric(forX), format(forX, "%Y"))
legend(x = as.numeric(forX[1], format = "%Y"), y = 0.2,
       c("predicted value", "confidence interval"), lty = c(1, 2))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
text(covid1+200, -0.25, "after \n covid", col="blue")
```

```
sampleList = INLA::inla.posterior.sample(30, res, selection = list(Predictor = 0))
sampleIntensity = exp(do.call(cbind, Biobase::subListExtract(sampleList,
"latent")))
sampleDeaths = matrix(rpois(length(sampleIntensity),
sampleIntensity), nrow(sampleIntensity), ncol(sampleIntensity))
sampleList70 = INLA::inla.posterior.sample(30, res70, selection = list(Predictor = 0))
sampleIntensity70 = exp(do.call(cbind, Biobase::subListExtract(sampleList70,
"latent")))
sampleDeaths70 = matrix(rpois(length(sampleIntensity70),
sampleIntensity70), nrow(sampleIntensity70), ncol(sampleIntensity70))
```

Graphs of 30 Posterior Samples

```r
# posterior samples vs real data points
matplot(xForInlaUnder50$time, sampleDeaths, col = "#00000010",
lwd = 2, lty = 1, type = "l", log = "y", xaxt = "n", xlab = "time",
ylab = "number of deaths", ylim = c(20, 120))
points(x[x$age == "0-49 years old", c("time", "dead")], col = "red",
cex = 0.5)
axis(1, as.numeric(forX), format(forX, "%Y"))
legend(x = as.numeric(forX[1], format = "%Y"), y = 35,
       c("real data point", "posterior sample"), lty = c(NA, 1),
       pch = c(1, NA), col=c("red", "#00000050"))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
text(covid1+180, 110, "after \n covid", col="blue")

lines(res$summary.fitted.values[, qCols], type = "l", lty = 2, col = "yellow")
```

```r
# posterior samples vs real data points
matplot(xForInlaOver70$time, sampleDeaths70, col = "#00000010",
lwd = 2, lty = 1, type = "l", log = "y", xaxt = "n", xlab = "time",
ylab = "number of deaths", ylim = c(600, 1700))
points(x[x$age == "70 years old and over", c("time", "dead")], col = "red",
cex = 0.5)
axis(1, as.numeric(forX), format(forX, "%Y"))
legend(x = as.numeric(forX[1], format = "%Y"), y = 1700,
       c("real data point", "posterior sample"), lty = c(NA, 1),
       pch = c(1, NA), col=c("red", "black"))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
text(covid1+200, 650, "after \n covid", col="blue")
```

```r
forX = as.Date(c("2019/6/1", "2020/11/1"))
forX = seq(forX[1], forX[2], by = "2 months")
matplot(xForInlaUnder50$time, sampleDeaths, col = "#00000010",
lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1",
"2020/11/1")), ylim = c(20, 120), xaxt = "n", xlab = "time", ylab="number of deaths")
axis(1, as.numeric(forX), format(forX, "%b%Y"))
points(x[x$age == "0-49 years old", c("time", "dead")], col = "red",
cex = 0.5)
legend(x = as.numeric(forX[1], format = "%Y"), y = 36,
       c("real data point", "posterior sample", "posterior quantitles"),
       lty = c(NA, 1, 2), pch = c(1, NA, NA), col=c("red", "#00000050", "black"))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
covid2 = as.numeric(as.Date("2020/06/01"), format="%b%Y")
abline(v = covid2, col = "blue", lwd = 1, lty = 2)
text((covid1 + covid2)/2, 25, "first wave", col = "blue")
covid3 = as.numeric(as.Date("2020/09/01"), format="%b%Y")
abline(v = covid3, col = "blue", lwd = 1, lty = 2)
text(covid3 + 35, 25, "second \n wave", col = "blue")
lines(xForInlaUnder50$time, res$summary.fitted.values[,"0.975quant"], type = "l", lty = 2)
lines(xForInlaUnder50$time, res$summary.fitted.values[,"0.025quant"], type = "l", lty = 2)
```

```r
matplot(xForInlaOver70$time, sampleDeaths70, col = "#00000010",
lwd = 2, lty = 1, type = "l", log = "y", xlim = as.Date(c("2019/6/1",
"2020/11/1")), xaxt = "n", xlab = "time", ylab="number of deaths", ylim=c(500, 1700))
```

```r
axis(1, as.numeric(forX), format(forX, "%b%Y"))
points(x[x$age == "70 years old and over", c("time", "dead")], col = "red",
cex = 0.5)
legend(x = as.numeric(forX[1], format = "%Y"), y = 740,
       c("real data point", "posterior sample", "posterior quantitles"),
       lty = c(NA, 1, 2), pch = c(1, NA, NA), col=c("red","#00000050", "black"))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
covid2 = as.numeric(as.Date("2020/06/01"), format="%b%Y")
abline(v = covid2, col = "blue", lwd = 1, lty = 2)
text((covid1 + covid2)/2, 700, "first wave", col = "blue")
covid3 = as.numeric(as.Date("2020/09/01"), format="%b%Y")
abline(v = covid3, col = "blue", lwd = 1, lty = 2)
text(covid3 + 40, 1400, "second \n wave", col = "blue")
lines(xForInlaOver70$time, res70$summary.fitted.values[,"0.975quant"],
      type = "l", lty = 2)
lines(xForInlaOver70$time, res70$summary.fitted.values[,"0.025quant"],
      type = "l", lty = 2)
```

```r
xPostCovid50 = xPostCovid[xPostCovid$age == "0-49 years old",]
xPostCovidForecast = sampleDeaths[match(xPostCovid50$time,
xForInlaUnder50$time), ]
excessDeaths = xPostCovid50$dead - xPostCovidForecast
xPostCovid70 = xPostCovid[xPostCovid$age == "70 years old and over",]
xPostCovidForecast70 = sampleDeaths70[match(xPostCovid70$time,
xForInlaOver70$time), ]
excessDeaths70 = xPostCovid70$dead - xPostCovidForecast70
```

```r
par(mar = c(4,4,1, 12), xpd = TRUE)
# detailed plot of posterior samples vs real data points
forX = as.Date(c("2020/3/1", "2020/09/1"))
forX = seq(forX[1], forX[2], by = "2 months")

matplot(xPostCovid50$time, xPostCovidForecast, type = "l",
ylim = c(20,90), col = "#00000050", xaxt = "n", xlab="time",
ylab = "number of deaths")
points(xPostCovid50[, c("time", "dead")], col = "red")
axis(1, as.numeric(forX), format(forX, "%b"))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
covid2 = as.numeric(as.Date("2020/06/01"), format="%b%Y")
abline(v = covid2, col = "blue", lwd = 1, lty = 2)
text((covid1 + covid2)/2, 30, "first wave", col = "blue")
covid3 = as.numeric(as.Date("2020/09/01"), format="%b%Y")
abline(v = covid3, col = "blue", lwd = 1, lty = 2)
text(covid3 + 25, 30, "second wave", col = "blue")
legend(x = as.numeric(forX[length(forX)], format = "%Y") + 70, y = 40,
       c("real data point", "posterior sample", "posterior quantitles"),
       lty = c(NA, 1, 2), pch = c(1, NA, NA), col=c("red", "#00000050", "black"),
       lwd = c(NA, 1, 2))
lines(xPostCovid50$time,
      res$summary.fitted.values[match(xPostCovid50$time, xForInlaUnder50$time),"0.975quant"],
      type = "l", lty = 2, lwd = 2)
lines(xPostCovid50$time,
      res$summary.fitted.values[match(xPostCovid50$time, xForInlaUnder50$time),"0.025quant"],
```

```
      type = "l", lty = 2, lwd = 2)
```

```
par(mar = c(4,4,1, 12), xpd = TRUE)
matplot(xPostCovid70$time, xPostCovidForecast70, type = "l", col = "#00000050",
        xaxt = "n", xlab="time", ylab = "number of deaths", ylim=c(600, 1700))
points(xPostCovid70[, c("time", "dead")], col = "red")
axis(1, as.numeric(forX), format(forX, "%b"))
abline(v = covid1, col = "blue", lwd = 1, lty = 2)
covid2 = as.numeric(as.Date("2020/06/01"), format="%b%Y")
abline(v = covid2, col = "blue", lwd = 1, lty = 2)
text((covid1 + covid2)/2, 700, "first wave", col = "blue")
covid3 = as.numeric(as.Date("2020/09/01"), format="%b%Y")
abline(v = covid3, col = "blue", lwd = 1, lty = 2)
text(covid3 + 25, 1400, "second wave", col = "blue")
lines(xPostCovid70$time,
      res70$summary.fitted.values[match(xPostCovid70$time, xForInlaOver70$time),"0.975quant"],
      type = "l", lty = 2, lwd = 2)
lines(xPostCovid70$time,
      res70$summary.fitted.values[match(xPostCovid70$time, xForInlaOver70$time),"0.025quant"],
      type = "l", lty = 2, lwd = 2)
legend(x = as.numeric(forX[length(forX)], format = "%Y") + 70, y = 900,
       c("real data point", "posterior sample", "posterior quantitles"),
       lty = c(NA, 1, 2), pch = c(1, NA, NA), col=c("red", "#00000050", "black"),
       lwd = c(NA, 1, 2))
```

```
# plot excess death (difference between red and black)
par(mar = c(4,4,1, 12), xpd = TRUE)
matplot(xPostCovid50$time, excessDeaths, type = "l",
lty = 1, col = "#00000030", xaxt = "n", xlab = "time",
ylab = "number of deaths", ylim = c(-40, 45))
axis(1, as.numeric(forX), format(forX, "%b"))
abline(v = covid2, col = "blue", lwd = 1, lty = 2)
text((covid1 + covid2)/2, -30, "first wave", col = "blue")
abline(v = covid3, col = "blue", lwd = 1, lty = 2)
text(covid3 + 25, -30, "second wave", col = "blue")
lines(xPostCovid50$time, apply(excessDeaths, 1, quantile, 0.975),
      type = "l", lty = 2, lwd = 2)
lines(xPostCovid50$time, apply(excessDeaths, 1, quantile, 0.025),
      type = "l", lty = 2, lwd = 2)
legend(x = as.numeric(forX[length(forX)], format = "%Y") + 70, y = -20,
       c( "excess deaths", "quantiles"), lty = c( 1, 2),
       col=c( "#00000050", "black"), lwd = c( 1, 2))
```

```
par(mar = c(4,4,1, 12), xpd = TRUE)
# plot excess death (difference between red and black)
matplot(xPostCovid70$time, excessDeaths70, type = "l",
lty = 1, col = "#00000030", xaxt = "n", xlab = "time", ylab = "excess deaths",
ylim = c(-250, 950))
axis(1, as.numeric(forX), format(forX, "%b"))
abline(v = covid2, col = "blue", lwd = 1, lty = 2)
text((covid1 + covid2)/2, 0, "first wave", col = "blue")
abline(v = covid3, col = "blue", lwd = 1, lty = 2)
text(covid3 + 25, 600, "second wave", col = "blue")
```

```
lines(xPostCovid70$time, apply(excessDeaths70, 1, quantile, 0.975),
      type = "l", lty = 2, lwd = 2)
lines(xPostCovid70$time, apply(excessDeaths70, 1, quantile, 0.025),
      type = "l", lty = 2, lwd = 2)
legend(x = as.numeric(forX[length(forX)], format = "%Y") + 70, y = 40,
       c( "excess deaths", "quantiles"), lty = c( 1, 2),
       col=c( "#00000050", "black"), lwd = c( 1, 2))
```

Tables

Tables of posterior fixed effects and random effects

```
qCols = paste0(c(0.5, 0.025, 0.975), "quant")
knitr::kable(rbind(res$summary.fixed[, qCols],
Pmisc::priorPostSd(res)$summary[,qCols]), digits = 3,
caption = "Posterior quantiles of fixed effects and random effects (under 50 years old)",
label = 'tab:atable')
knitr::kable(rbind(res70$summary.fixed[, qCols],
Pmisc::priorPostSd(res70)$summary[,qCols]), digits = 3,
caption = "Posterior quantiles of fixed effects and random effects (over 70 years old)",
label = 'tab:atable')
```

Tables of excess deaths (real data - predicted samples)

```
# excess deaths from March to May inclusively
excessDeathsSub = excessDeaths[xPostCovid50$time >
as.Date("2020/03/01") & xPostCovid50$time <
as.Date("2020/06/01"), ]
excessDeathsInPeriod = apply(excessDeathsSub, 2, sum)
excessDeaths70Sub = excessDeaths70[xPostCovid70$time >
as.Date("2020/03/01") & xPostCovid70$time <
as.Date("2020/06/01"), ]
excessDeaths70InPeriod = apply(excessDeaths70Sub, 2, sum)
df = data.frame("under 50 years old" = quantile(excessDeathsInPeriod,
                                                c(0.025, 0.5, 0.975)),
                "over 70 years old" = quantile(excessDeaths70InPeriod,
                                               c(0.025, 0.5, 0.975)))
knitr::kable(df, digits = 0,
caption = "Sample Quantiles of Excess Deaths from March to May Inclusively",
label = 'tab:atable')
```

```
# excess deaths after September
excessDeathsSub = excessDeaths[xPostCovid50$time >
as.Date("2020/09/01"), ]
excessDeathsInPeriod = apply(excessDeathsSub, 2, sum)

excessDeaths70Sub = excessDeaths70[xPostCovid70$time >
as.Date("2020/09/01"), ]
excessDeaths70InPeriod = apply(excessDeaths70Sub, 2, sum)
```

```r
df = data.frame("under 50 years old" = quantile(excessDeathsInPeriod,
                                                c(0.025, 0.5, 0.975)),
                "over 70 years old" = quantile(excessDeaths70InPeriod,
                                               c(0.025, 0.5, 0.975)))
knitr::kable(df, digits = 0,
caption = "Sample Quantiles of Excess Deaths from Sept 1 to Oct 21",
label = 'tab:atable')
```