LONDON SCHOOL of HYGIENE &TROPICAL MEDICINE

KIT
Karlsruher Institut für Technologie

Universität Zürich UZH

FAU Friedrich-Alexander-Universitä
Erlangen-Nürnberg

Workshop on the endemic-epidemic framework for infectious disease modelling
R tutorial: using hhh4

Johannes, Maria, Sebastian (with on-site assistance at CMMID)

Wednesday 23rd March 2022 10:00–12:30 UK time

Outline

This part of the tutorial serve to introduce you to multivariate hhh4 with covariates ("taster"). The format of this part is:

- Introduction of exercise
- Time to do exercise (hands on)
- Brief discussion of results (time permitting)

If questions arise during the hands on part, we will utilise breakout rooms to assist you. Please use the chat function to flag a need for assistance. If persistent questions arise we may address them in plenary during the hands on part.

Please take breaks as needed during the hands on part.

Multivariate hhh4 models with covariates

Main objective

Create a multivariate endemic-epidemic model populated by our "own" data

Goals

- Create *simple* endemic-epidemic model for coronavirus disease 2019 (COVID-19) data
- Use `hhh4`with a case data set which is external to packages in the hhh4 package ecosystem
- Include a covariate in the endemic component of the model

Your task

- Go to coronavirus.data.gov.uk and download the “Cases by specimen date” data set by Nation and create a four dimensional surveillance time series (sts) object.

```
data <- read.csv("data_2022-Mar-15.csv")
```

```
tail(data)
```

```
##            England Northern Ireland Scotland Wales
## 2022-03-09   55106             2533    13913  1568
## 2022-03-10   53678             2211    12471  1495
## 2022-03-11   51731             1960    11319  1383
## 2022-03-12   48269             1687    10023  1119
## 2022-03-13   58013             2046    11969  1132
## 2022-03-14   42035             2134     8648     5
```

- Consider how you should format the data and what you need to include as `start` and `frequency` arguments when constructing the `sts` object

```
sts(data)
-- An object of class sts --
freq:  X
start:  Y Z
dim(observed):  N 4
```

- What other arguments from `sts` might you want to consider? (will be revisited)

- What kind of model is run if you call hhh4 on your sts object with no further inputs?
- Do visual inspection of your sts object. Based on the surveillance data, which (fixed or random) effect of country do you expect to have the largest effect estimate?
- Create a model with fixed effect (fe) of Nation in the endemic (end) component. Are the effect sizes as you expect based on the plot from before and the output of summary? How does this compare with the “default” code model?

- Look at a map of the UK and create the `matrix` of neighbourhood orders. Each country has adjacency 0 to itself so put zeros on the diagonal (G). Count countries next to each other. For example: Scotland borders England so they have adjacency 1 (Y) but you have to go through England to get between Scotland and Wales so they have adjacency 2 (B)

	Eng	NI	Sco	Wal
Eng	G		Y	
NI		G		
Sco	Y		G	B
Wal			B	G

- Something to consider in construction: How should you treat Northern Ireland? They share no land borders with the other Nations (no "obvious" solution)

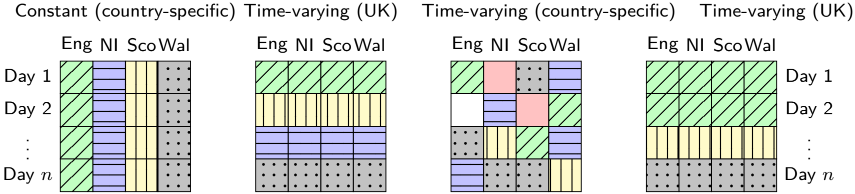
- Update your hhh4 model to include these neighbourhood (ne) weights and include also fixed effects in the epidemic component. Plot the model (plot). Does it have a good fit to the data?

```
neigh_mat <- matrix(c(0, X, 1, X,
                      X, 0, X, X,
                      1, X, 0, 2,
                      X, X, 2, 0),
                    ncol = 4, byrow = TRUE) + 1
```

- Locate (using the internet) population estimates for the four countries and incorporate this information in your sts object. Now include population(sts_covid) as an offset in your endemic (end) component. What effect does this have on the end.1.X estimates?

```
uk_pop <- c(X, Y, Z, W)
sts_covid <- sts(...)
hhh4(sts_covid, ...)
```

Covariates in the model need to be formatted to be the same dimension as the `sts` object



e.g. monthly covariate with daily counts
or yearly with weekly counts
(different temporal resolution)

...many different options exist! For example you could information which is gathered for England and Wales (combined) or only available for Scotland

- Return to coronavirus.data.gov.uk and include some information from “Testing capacity by Pillar” in your model

```
pillar <- read.csv("pillar_2022-Mar-15.csv")
```

Your task (recap)

1. Load COVID-19 case data from `coronavirus.data.gov.uk` and create a four-dimensional `sts` object
2. Write down the expression of the model if `hhh4` is run on this object with no further arguments/default arguments
3. Based on visual inspection of the `sts` object which country do you expect will have the largest effect estimate as a fixed effect in the endemic component?
4. Construct the matrix of neighbourhood order between Nations and add 1 (+1) before including it in your model and add fixed effects in the epidemic component
5. Include population size estimates in your `sts` object and offset the endemic component with them
6. Include testing capacity by pillar in your model

If you finish early, feel free to continue exploring the modelling framework and `hhh4` options

Go to coronavirus.data.gov.uk and download the "Cases by specimen date" data set by Nation and create a four dimensional surveillance time series (sts) object.

```
data <- read.csv("data_2022-Mar-15.csv")
dates <- unique(data$date) # For use with covariate later
data <- t(with(data, tapply(newCasesBySpecimenDate, list(areaName, date),
                                 function(x){x})))
tail(data)


##            England Northern Ireland Scotland Wales
## 2022-03-09   55106             2533    13913  1568
## 2022-03-10   53678             2211    12471  1495
## 2022-03-11   51731             1960    11319  1383
## 2022-03-12   48269             1687    10023  1119
## 2022-03-13   58013             2046    11969  1132
## 2022-03-14   42035             2134     8648     5
```

We need to transfer the data from having an areaName column to instead have columns for each of the four nations

We construct the sts object

```
library(surveillance)
sts_covid <- sts(data,
                 start = c(2020, as.Date(rownames(head(data, 1))) -
                             as.Date("2019-12-31")),
                 frequency = 365)
```

We have a frequency of 365 since we are considering daily case counts and a start value with year 2020 and sample number 30 for the first observation (for 30th January, the date of the first case)

```
sts_covid@freq

## [1] 365

sts_covid@start

## [1] 2020   30
```

We might also want to consider including neighbourhood and population in our sts construction

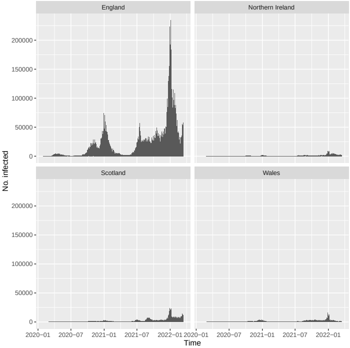
```
mod0 <- hhh4(sts_covid)
summary(mod0)

##
## Call:
## hhh4(stsObj = sts_covid)
##
## Coefficients:
##        Estimate   Std. Error
## end.1  8.7908561  0.0002246
##
## Log-likelihood:   -29307431
## AIC:              58614864
## BIC:              58614870
##
## Number of units:        4
## Number of time points:  774
##   (81 observations excluded due to missingness)
```

A call of hhh4 on this sts object with no further inputs yields the model with

$$\log(\nu_{\text{nation }i,t}) = \alpha, \quad \hat{\alpha} = \exp(8.79) = 6573.86 \tag{1}$$

(This is not super informative)

```
autoplot.sts(sts_covid)
```



Most cases seem to be reported in England so we could expect a coefficient for England (when included in the model) to have a large effect estimate (all other things equal)

We include a fixed effect of Nation in the endemic component

```
mod1 <- hhh4(sts_covid,
             control = list(end = list(f = ~ fe(1, unitSpecific = TRUE) - 1)))
summary(mod1)


##
## Call:
## hhh4(stsObj = sts_covid, control = list(end = list(f = ~fe(1,
##     unitSpecific = TRUE) - 1)))
##
## Coefficients:
##                           Estimate   Std. Error
## end.1.England             9.9798865  0.0002446
## end.1.Northern Ireland    6.7783361  0.0012335
## end.1.Scotland            7.6846576  0.0007851
## end.1.Wales               7.0150801  0.0010966
##
## Log-likelihood:   -14033963
## AIC:              28067934
## BIC:              28067958
##
```

```
## Number of units:        4
## Number of time points:  774
##   (81 observations excluded due to missingness)
```

Now the model has

$$log(\nu_{\text{nation } i,t}) = \alpha_{\text{nation } i} \tag{2}$$

Based on AIC we see an improvement

```
AIC(mod0, mod1)


##      df      AIC
## mod0  1 58614864
## mod1  4 28067934
```

Before we had

$$\hat{\alpha} = \exp(8.79) = 6573.86 \tag{3}$$

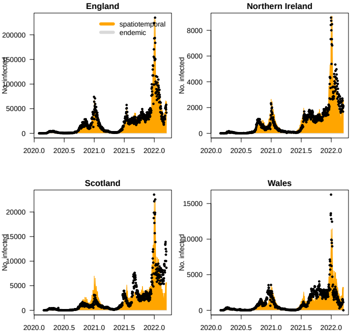
Now we have

$$log(\nu_{\text{nation } i,t}) = \begin{cases} \exp(9.98) = 21587.86 & i = \text{England} \\ \exp(6.78) = 878.61 & i = \text{Northern Ireland} \\ \exp(7.68) = 2174.73 & i = \text{Scotland} \\ \exp(7.02) = 1113.3 & i = \text{Wales} \end{cases} \tag{4}$$

As expected, England has a larger effect than the other three nations. This seems slightly more informative than the "default" code model as the epi curves for the four countries looked different

We look at a map of the UK and create the `matrix` of neighbourhood orders. We assumed here that Northern Ireland is a neighbour to Scotland but not Wales. We include this as a transmission `weight` matrix between Nations in the model.

```r
# Here assuming only NI and Scotland connected
neigh_mat <- matrix(c(0, 0, 1, 1,
                      0, 0, 1, 0,
                      1, 1, 0, 2,
                      1, 0, 2, 0),
                    ncol = 4, byrow = TRUE) + 1
colnames(neigh_mat) <- rownames(neigh_mat) <-
  c("England", "Northern Ireland", "Scotland", "Wales")
mod2 <- hhh4(sts_covid,
             control = list(end = list(f = ~ fe(1, unitSpecific = TRUE) - 1),
                            ne = list(f = ~ fe(1, unitSpecific = TRUE) - 1,
                                      weights = neigh_mat)))
plot(mod2, units = 1 : 4)
```



Overall a good fit. Seems like a couple of the peaks for Scotland not fully captured

We include population estimates for the four Nations in the `sts` object.

```r
#Population estimates for the UK, England and Wales,
#Scotland and Northern Ireland: mid-2020
uk_pop <- c(56550000, 1896000, 5466000, 3170000)
names(uk_pop) <- c("England", "Northern Ireland", "Scotland", "Wales")
# Implicit assumption of population not changing over time

# Update sts_covid to contain this information
sts_covid <- sts(data,
                 start = c(2020, as.Date(rownames(head(data, 1))) -
                             as.Date("2019-12-31")),
                 frequency = 365,
                 population = uk_pop)
```

NB rather than using `neigh_mat` we could have included the neighbourhood order adjacencies in the `sts` object and called it from the object when creating the `control` list

We include the populations as offsets in the endemic (end) and examine the effect this has on on the end.1.X estimates:

```
mod3 <- hhh4(sts_covid,
             control = list(end = list(f = ~ fe(1, unitSpecific = TRUE) - 1,
                                      offset = population(sts_covid)),
                            ne = list(f = ~ fe(1, unitSpecific = TRUE) - 1,
                                      weights = neigh_mat)))
```

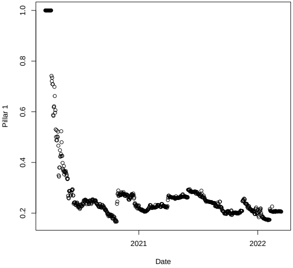
```
summary(mod2)$fixef

##                              Estimate
## ne.1.England               -0.2901628
## ne.1.Northern Ireland      -3.4822757
## ne.1.Scotland              -3.1733242
## ne.1.Wales                 -3.8722989
## end.1.England               4.8024605
## end.1.Northern Ireland    -15.0290624 1
## end.1.Scotland            -14.2320065 2
## end.1.Wales                 2.1275934
```

```
summary(mod3)$fixef

##                              Estimate    S
## ne.1.England               -0.2901628  0.0
## ne.1.Northern Ireland      -3.4822757  0.0
## ne.1.Scotland              -3.1733242  0.0
## ne.1.Wales                 -3.8722989  0.0
## end.1.England             -13.0481752  0.0
## end.1.Northern Ireland    -29.9529791 242.7
## end.1.Scotland            -29.0169901 194.7
## end.1.Wales               -12.8416488  0.0
```

Estimates in the endemic component change (as expected) as they are now offset by population

We include information from “Testing capacity by Pillar” in the model. We consider the proportion of planned capacity made up by pillar 1 testing (just an example)

```
pillar <- read.csv("pillar_2022-Mar-15.csv")
pillar <- pillar[pillar$date %in% dates, ] # Match to case data
pillar$prop <- pillar$capacityPillarOne / pillar$plannedCapacityByPublishDate
plot(as.Date(pillar$date), pillar$prop, xlab = "Date", ylab = "Pillar 1")
```

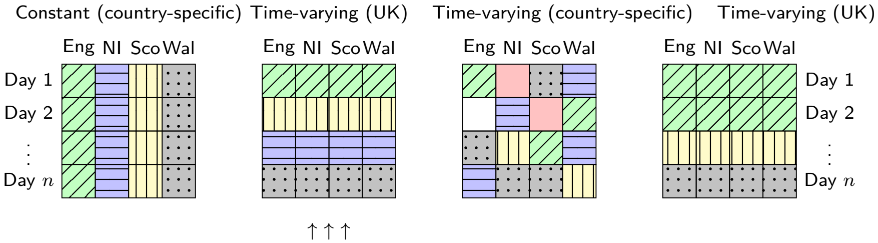


This seems to have the opposite pattern than our case data; larger values at the beginning compared to the end

```r
head(pillar[, c("areaName", "date", "plannedCapacityByPublishDate", "prop")])
```

```
##         areaName       date plannedCapacityByPublishDate      prop
## 1 United Kingdom 2022-03-14                       975032 0.2056302
## 2 United Kingdom 2022-03-13                       970890 0.2065074
## 3 United Kingdom 2022-03-12                       966896 0.2073605
## 4 United Kingdom 2022-03-11                       966896 0.2073605
## 5 United Kingdom 2022-03-10                       969222 0.2068628
## 6 United Kingdom 2022-03-09                       966896 0.2073605
```

Note the values of `areaName` – this is the second type of covariate shown in the earlier slide



Constant (country-specific) Time-varying (UK) Time-varying (country-specific) Time-varying (UK)

It is important to ensure the covariates have the correct dimensions and format

```
# Keep what we are using
# Sanity check -- are dimensions the same
isTRUE(dim(data)[1] == dim(pillar)[1])

## [1] FALSE

# Need to add zeros for the following dates
pillar <- rbind(pillar[, c("date", "prop")],
      cbind(date = dates[which(!(dates %in% pillar$date))],
            prop = rep(0, length(dates[which(!(dates %in% pillar$date))]))))

# Sanity check -- is covariate numeric?
isTRUE(is.numeric(pillar$prop))

## [1] FALSE

test <- as.numeric(pillar$prop)
```

This does not seem to yield an improvement – other options should be considered (homework for you!)

```
mod4 <- hhh4(sts_covid,
             control = list(end = list(f = ~ fe(1, unitSpecific = TRUE) - 1,
                                       offset = population(sts_covid)),
                            ne = list(f = ~ fe(1, unitSpecific = TRUE) - 1 +
                                        test,
                                      weights = neigh_mat)))

AIC(mod3, mod4)
```

```
##      df     AIC
## mod3  8 1718460
## mod4  9 1718462
```