Outbreaks

Mladen Cucak

5/4/2020

## Libraries

list.of.packages <-  
 c(  
 "tidyverse",  
 "readxl",  
 "broom",   
 "data.table",  
 "minpack.lm",  
 "knitr",  
 "zoo",  
 "egg",  
 "ggthemes",  
 "here",  
 "stringr",  
 "lubridate",  
 "nlme",  
 "pander",  
 "maps",  
 "ggrepel",  
 "sf",  
 "ggspatial",  
 "factoextra",   
 "NbClust"  
 )  
  
new.packages <-  
 list.of.packages[!(list.of.packages %in% installed.packages()[, "Package"])]  
  
#Download packages that are not already present in the library  
if (length(new.packages))  
 install.packages(new.packages)  
  
if (length(new.packages))  
 install.packages(new.packages, repos = c(CRAN="https://cran.r-project.org/"))  
  
  
if ("reconPlots" %in% installed.packages() == FALSE) devtools::install\_github("andrewheiss/reconPlots")  
  
packages\_load <-  
 lapply(list.of.packages, require, character.only = TRUE)  
  
#Print warning if there is a problem with installing/loading some of packages  
if (any(as.numeric(packages\_load) == 0)) {  
 warning(paste("Package/s", paste(list.of.packages[packages\_load != TRUE]), "not loaded!"))  
} else {  
 print("All packages were successfully loaded.")  
}

## [1] "All packages were successfully loaded."

rm(packages\_load, list.of.packages, new.packages)

## Background

I guess the whole idea about this is related to some of my PhD projects that I could not finish. I had a pile of yearbooks with initial disease outbreaks and follow-up assessments from early 1950s to late 1990s. The thickness of these yearbooks was slimmer and slimmer towards end of the last century, hence, beside the historical and theoretical value they would not be of much use for my current project. I have given up on digitising them, because both myself and few of my students started hating me, because they are really messy.  
(Very relevant paper by Zwankhuizen & Zadoks)[<https://bsppjournals.onlinelibrary.wiley.com/doi/full/10.1046/j.1365-3059.2002.00738.x>], which I think I have mentioned to Neil.  
The main idea related to the current data set is that it is a bit more relevant to current agrosystem. It might be interesting to see how the initial outbreaks, their distribution, rate of increase, etc., is changes over years. This is a observational “study” and comes with all its caveats. These are reports by blight scouts, and it is hard to know if some scouts are more active than others, and similar variables. However, this data is collected by Louise Cooke, and I am sure she would be happy to answer any queries, if it comes to that.

## Data

The disease outbreak data consisted of the date and coordinates of 352 late blight outbreaks from across Northern Ireland over an 11- year period (2005-2014). The NI data were collected every year as part of the Agriculture and Horticulture Development Board (AHDB) Potatoes ‘Fight Against Blight’ campaign (<https://potatoes.ahdb.org.uk/>). The coordinates of blight outbreak locations were obtained using the ‘geocoding’ function from the ‘ggmap’ package (Kahle and Wickham 2013) and confirmed manually. The same data set was used in revision of the Hutton Critera, previously known as Smith Periods (some details [here](https://euroblight.net/fileadmin/euroblight/Workshops/AArhus/Proceedings/5._Siobhan_Dancy-p53-58.pdf)).

outbreaks <- read\_csv(file = here::here("dat", "outbreaks\_fin.csv"))  
  
outbreaks$date <- lubridate::mdy(outbreaks$date)  
  
outbreaks$jday <- as.Date(outbreaks$date,format="%m/%d/%Y") %>% yday()  
  
  
#Remove outbreaks from   
outbreaks <-   
 outbreaks %>%  
 filter(lat>53.9)   
   
outbreaks <-   
 outbreaks %>%   
   
 mutate(source = tolower(source)) %>%   
 mutate(source = ifelse(source == "Crop", "crop",   
 ifelse(source == "crops", "crop",source))) %>%   
 mutate(source = ifelse(source == "groundkeepers", "groundkeeper", source)) %>%   
 mutate(source = ifelse(source == "allotments", "allotment", source))

A look at data set.

pander(head(outbreaks))

Table continues below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| date | yr | source | variety | varietyII | Location |
| 2003-05-21 | 2003 | crop | Lady Felicia | NA | Ards |
| 2003-05-31 | 2003 | crop | Home Guard | NA | BallyLagen,BClare |
| 2003-06-05 | 2003 | crop | Kerrs Pink | NA | Benagh, Kilkeel |
| 2003-06-06 | 2003 | crop | Kerrs Pink | NA | Glenkeen Colraine |
| 2003-06-06 | 2003 | crop | Home Guard | NA | Pollee B.Mena |
| 2003-06-07 | 2003 | crop | Kerrs Pink | NA | Dunamoy B.Clare |

Table continues below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| County | Comments | loc\_full | lon | lat |
| Down | NA | Ards, Down, UK | -5.664 | 54.66 |
| Antrim | NA | BallyLagen,BClare, Antrim, UK | -5.922 | 54.76 |
| Down | protected crop | Benagh, Kilkeel, Down, UK | -6.078 | 54.05 |
| Londonderry | NA | Glenkeen Colraine, Londonderry, UK | -7.305 | 54.99 |
| Antrim | NA | Pollee B.Mena, Antrim, UK | -6.121 | 54.55 |
| Antrim | NA | Dunamoy B.Clare, Antrim, UK | -6.041 | 54.78 |

|  |
| --- |
| jday |
| 141 |
| 151 |
| 156 |
| 157 |
| 157 |
| 158 |

Varieties present in the data. I know most of them, and all are traditional varieties with not partial resistance.

unique(outbreaks$ variety)

## [1] "Lady Felicia" "Home Guard"   
## [3] "Kerrs Pink" "British Queen"   
## [5] "Desiree" "Saxon"   
## [7] NA "Up to Date"   
## [9] "Dunbar Standard" "D.Standard"   
## [11] "King Edward" "Dundrod"   
## [13] "Sante" "Nicola"   
## [15] "Red Duke of York" "Navan"   
## [17] "Rooster" "Pentland Squire"   
## [19] "Arran Victory" "Milagro"   
## [21] "Sharpes Express" "Arran Banner"   
## [23] "Caberet" "Dunluce"   
## [25] "Maris Piper" "Desiree & Nicola"   
## [27] "Lady Claire" "Golden Wonder"   
## [29] "Pentland Ivory" "Marfona"   
## [31] "Pentland Javelin" "Pink Fir Apple"   
## [33] "Avalanche" "UpToDate"   
## [35] "Santee" "Romano"   
## [37] "Piccolo Star" "various"   
## [39] "Charlotte" "Lady Rosetta"   
## [41] "Paramount" "R1"   
## [43] "Up To Date" "Cara"   
## [45] "Sarpo Mira" "Hermes"   
## [47] "Cabaret" "Duke of York"   
## [49] "Tomato Money Maker" "Tomato Gardeners Delight"  
## [51] "Up-to-Date" "AFLO1\_1"   
## [53] "VR808" "Golden Nugget"   
## [55] "Aphrodite" "Kifli"   
## [57] "Ramos" "Up-To-Date"

This is the data summary for different sources.

outbreaks %>%   
 group\_by(yr, source) %>%   
 summarise(counts = n()) %>%   
 spread(yr, counts)%>%   
 replace(is.na(.), 0) %>%   
 pander()

Table continues below

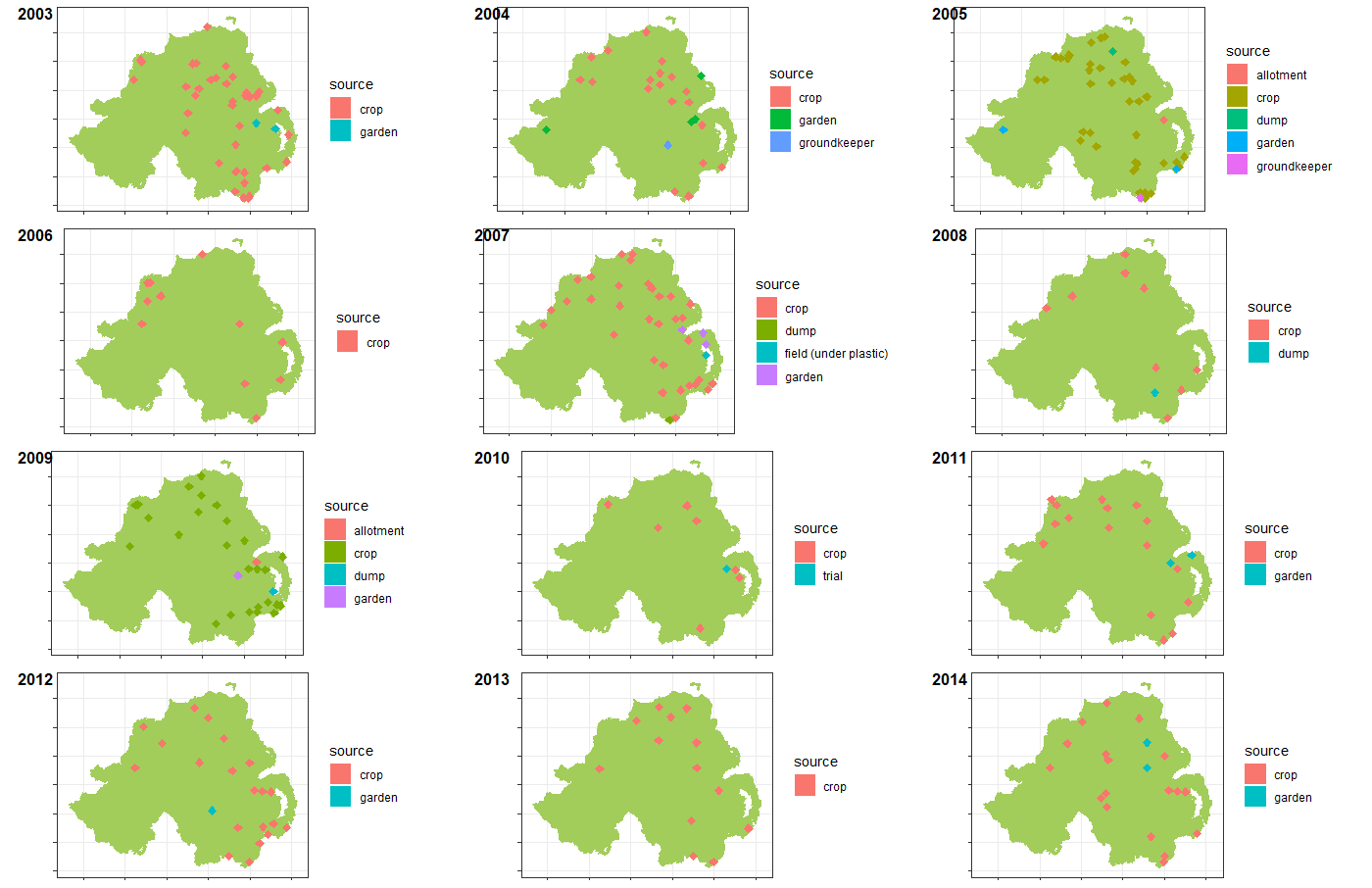
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| source | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 |
| allotment | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 |
| crop | 43 | 21 | 46 | 15 | 43 | 9 | 36 | 9 |
| dump | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| field (under plastic) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| garden | 2 | 4 | 2 | 0 | 3 | 0 | 1 | 0 |
| groundkeeper | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| trial | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2011 | 2012 | 2013 | 2014 |
| 0 | 0 | 0 | 0 |
| 25 | 35 | 16 | 32 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 2 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

The entire data set can be seen [here](https://github.com/mladencucak/Outbreaks/blob/master/dat/outbreaks_fin.csv)

!!TASK plot outbreak maps for each year

# ni\_sf <-  
# all\_counties.sf[all\_counties.sf$CountyName == c("Tyrone","Antrim","Armagh", "Fermanagh","Londonderry","Down"),]  
# basemap <-   
# ggplot() +  
# # Plot borders (shapefile)  
# geom\_sf(  
# data = ni\_sf  
# ,  
# color= "darkolivegreen3",  
# fill = "darkolivegreen3"  
# ) +  
# #Set the theme  
# theme\_bw(  
#   
# ) +  
# #limit the plotting area  
# coord\_sf(xlim = c(-8.4, -4.9), ylim = c(54, 55.4), expand = FALSE) +  
# theme(axis.text.x=element\_blank(),  
# axis.text.y=element\_blank())  
# # Define names for labs  
# labs(x = "Longitude", y = "Latitude")+  
# #add fancy anotation  
# annotation\_north\_arrow(location = "br", which\_north = "true",   
# pad\_x = unit(0.35, "in"),  
# pad\_y = unit(0.25, "in"),  
# style = north\_arrow\_fancy\_orienteering) +  
#   
# annotation\_scale(location = "br", width\_hint = 0.4)   
#   
# plist <- list()  
# for(i in seq(unique(df\_loc\_sf$yr))){  
# yrs <- unique(df\_loc\_sf$yr)[i]  
#   
# plist[[i]] <-   
# basemap+  
# geom\_sf(  
# data = df\_loc\_sf[df\_loc\_sf$yr == unique(df\_loc\_sf$yr)[i],],  
# aes(fill = source, color = source),  
# shape = 23,  
# size = 2  
# ) +  
# geom\_point( size = .5)  
# }  
#

 `

## Model fitting

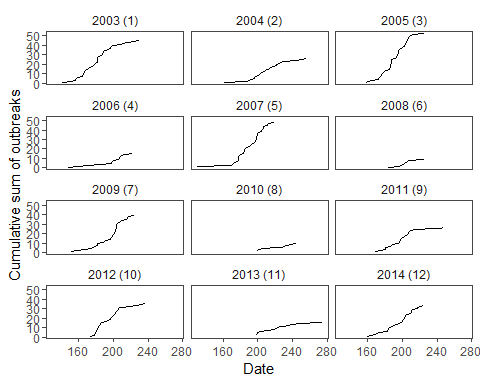
### Prepare the data

Initially, a simple cumulative sum per year was calcualted with julian day and year as covariates.

cdfdt <-   
outbreaks[outbreaks$Location!= "Oak Park",] %>%  
 mutate(mon = month(date),  
 day = day(date)) %>%  
 group\_by(mon, day) %>%  
 unite(., date, mon, day, sep = "-", remove = FALSE) %>%  
 mutate(d = as.Date(date, format = "%m-%d")) %>%  
 ungroup() %>%   
 select(yr,jday) %>%   
 group\_by(yr,jday) %>%  
 summarise(count = n()) %>%   
 ungroup() %>%   
 group\_by(yr) %>%   
 mutate(cums = cumsum(count)) %>%   
 dplyr::select(.,yr, jday, cums) %>%   
 ungroup() %>%   
 mutate(year = yr,  
 yr = as.factor(yr-min(yr-1)))  
  
pander::pander(cdfdt[1:20,])

|  |  |  |  |
| --- | --- | --- | --- |
| yr | jday | cums | year |
| 1 | 141 | 1 | 2003 |
| 1 | 151 | 2 | 2003 |
| 1 | 156 | 3 | 2003 |
| 1 | 157 | 5 | 2003 |
| 1 | 158 | 6 | 2003 |
| 1 | 164 | 7 | 2003 |
| 1 | 168 | 13 | 2003 |
| 1 | 170 | 14 | 2003 |
| 1 | 171 | 16 | 2003 |
| 1 | 177 | 19 | 2003 |
| 1 | 178 | 20 | 2003 |
| 1 | 179 | 21 | 2003 |
| 1 | 181 | 22 | 2003 |
| 1 | 182 | 27 | 2003 |
| 1 | 184 | 28 | 2003 |
| 1 | 186 | 29 | 2003 |
| 1 | 188 | 32 | 2003 |
| 1 | 189 | 33 | 2003 |
| 1 | 191 | 34 | 2003 |
| 1 | 197 | 37 | 2003 |

cdfdt %>%   
mutate(lab =paste0(year," (",yr,")")) %>%   
ggplot() +  
 geom\_line(aes(jday, cums))+  
 facet\_wrap(~ lab, ncol = 3)+  
 labs(y = "Cumulative sum of outbreaks",  
 x = "Date") +  
 theme\_article()+  
ggsave(here::here("out", "Cumsum of outbreaks per year.png"),  
 width = 6,  
 height = 4.5,  
 dpi = 620)



### Fit the model

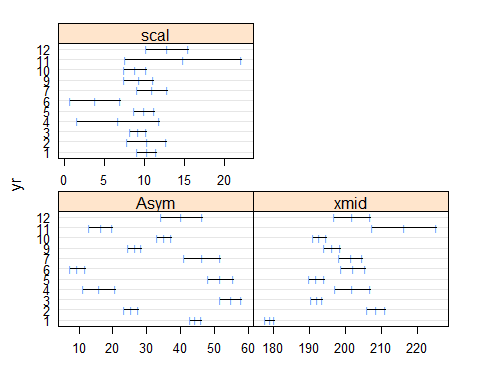
Get the starting values.

start\_vals <-   
 cdfdt %>%  
 filter(yr != "8") %>%  
 nlsList(cums ~ SSlogis(jday, Asym, xmid, scal) | yr,  
 data = .)  
# broom::tidy(start\_vals)

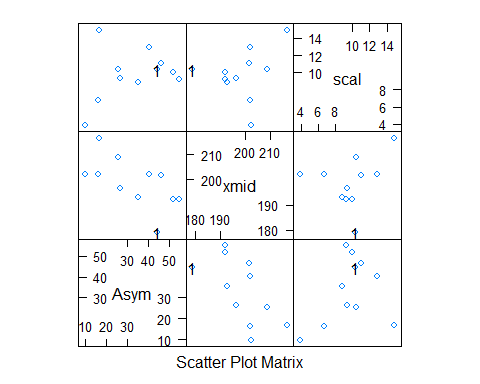
start\_vals

## Call:  
## Model: cums ~ SSlogis(jday, Asym, xmid, scal) | yr   
## Data: .   
##   
## Coefficients:  
## Asym xmid scal  
## 1 44.49956 179.0494 10.346504  
## 2 25.51055 208.6816 10.396048  
## 3 54.93293 192.1397 9.256966  
## 4 16.08021 201.9996 6.748833  
## 5 51.85104 192.0237 9.991017  
## 6 9.60443 202.2748 3.909670  
## 7 46.46492 201.5679 11.036716  
## 9 26.58916 196.4398 9.358434  
## 10 35.30460 192.8631 8.859651  
## 11 16.55426 216.4380 14.897322  
## 12 40.34877 201.9031 12.913451  
##   
## Degrees of freedom: 201 total; 168 residual  
## Residual standard error: 1.56226

start\_vals %>%  
 intervals %>%  
 plot



pairs(start\_vals, id = 0.1)



Logistic fit with julian day as predictor and year as random grouping variable.

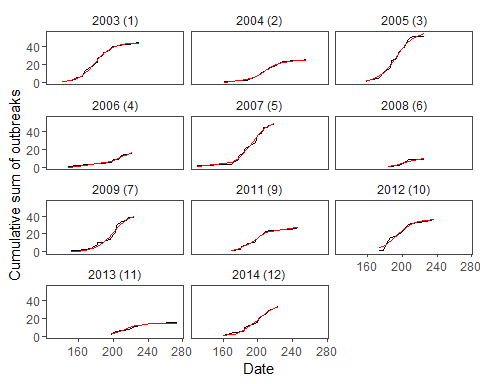
# Fit on lnear model with logistic fuction  
w1 <-   
 cdfdt %>%  
 filter(yr != "8") %>%  
 nlme(cums ~ SSlogis(jday, Asym, xmid, scal),# model evaluates the logistic function and its gradient  
 fixed = list(Asym ~ 1,  
 xmid ~ 1,  
 scal ~ 1),#Fixed effects of the   
 #random parameters of the model grouped by year  
 #to reduce the number of parameters, the scale and it was excluded because it does not impact the model fit  
 random = Asym + xmid ~ 1 | yr,  
 data = .,  
 start = fixef(start\_vals)#Start parameters for the optimisation alghorithm  
 )

Calculate the predictions.

cdfdt\_pred <-   
 cdfdt %>%  
 filter(yr != "8") %>%  
 mutate(pred = predict(w1))

Plot predicted values.

cdfdt\_pred %>%   
 mutate(lab =paste0(year," (",yr,")")) %>%   
ggplot() +  
 geom\_line(aes(jday, cums))+  
 facet\_wrap(~ lab, ncol = 3)+  
 labs(y = "Cumulative sum of outbreaks",  
 x = "Date") +  
 theme\_article() +  
 geom\_line(aes(jday, pred), col = 2)



Extract parameter values.

params <-   
cdfdt\_pred %>%   
 mutate(lab =paste0(year," (",yr,")")) %>%   
 group\_by(lab) %>%  
 summarise(year = unique(lab)) %>%   
 ungroup() %>%   
 select(year) %>%   
 bind\_cols(coef(w1)) %>%   
 write\_csv(here::here("out" , "paramter\_est\_per\_year.csv"))

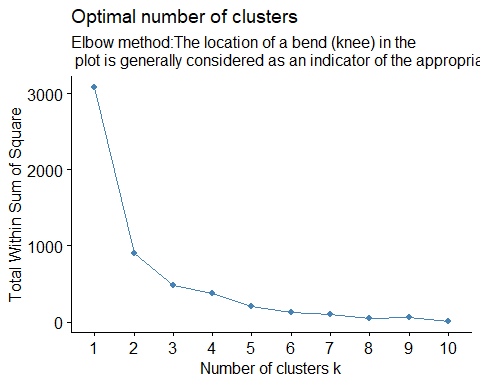
## Clustering

Aplication of any method is limited due to low number of data points.

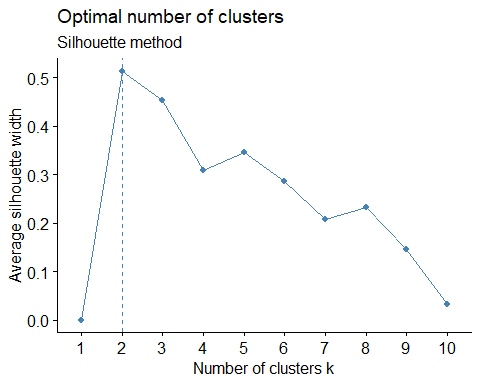
### Estimate optimum number of clusters

althoug there are numerous methods to hel determining the number of clusters, in the end it is a subjective decision.

# Elbow method  
#Minimising the toal sum of squares  
# choose a number of clusters so that adding another cluster doesn’t improve much better the total WSS  
fviz\_nbclust(params[, 2:4], kmeans, method = "wss") +  
 labs(subtitle = "Elbow method:The location of a bend (knee) in the\n plot is generally considered as an indicator of the appropriate number of clusters.")



# Silhouette method  
# it measures the quality of a clustering.  
# That is, it determines how well each object lies within its cluster.   
# A high average silhouette width indicates a good clustering.  
fviz\_nbclust(params[, 2:4], kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



According to these results and our subjective feeling, it is possible to define k = 3 as the optimum number of clusters in the data.

#Seeds allow you to create a starting point for randomly generated numbers,   
#so that each time your code is run, the same answer is generated  
set.seed(20)  
clusters <-  
kmeans(params[, 2:4],3, nstart = 20)  
  
#number of points in each cluster  
clusters$size

## [1] 4 3 4

#vector of within-cluster sum of squares, one component per cluster.  
clusters$withinss

## [1] 128.1499 191.9003 163.7779

clusters$tot.withinss # total within-cluster sum of squares

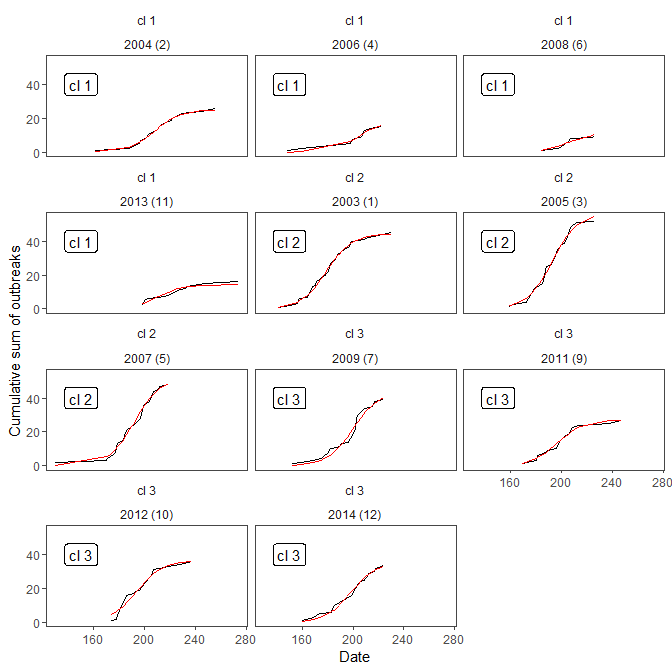
## [1] 483.8282

params$group <- as.factor(clusters$cluster)

Plot the observed and predicted curves and the cluster they are assigned to.

cdfdt\_pred %>%   
 mutate(year =paste0(year," (",yr,")")) %>%   
 left\_join(., params[, c("group", "year")]) %>%   
 mutate(group = paste0("cl ",as.character(group) )) %>%   
ggplot() +  
 geom\_line(aes(jday, cums))+  
 geom\_label(aes(x = 150, y=40,label = group))+  
 facet\_wrap(group ~ year, ncol = 3)+  
 labs(y = "Cumulative sum of outbreaks",  
 x = "Date") +  
 theme\_article() +  
 geom\_line(aes(jday, pred), col = 2)

## Joining, by = "year"



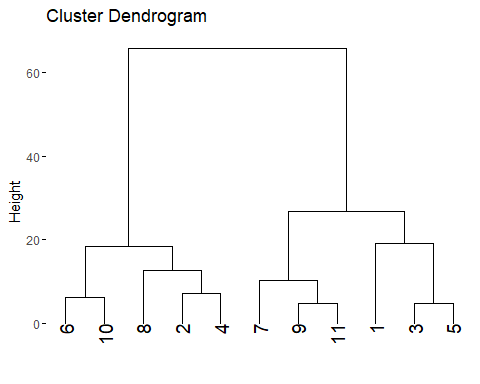
### Dendograms

Create dendogram.

res.dist <- dist(params[, 2:4], method = "euclidean")  
  
as.matrix(res.dist)[1:6, 1:6]

## 1 2 3 4 5 6  
## 1 0.00000 34.920068 18.266488 36.125756 14.793138 42.187996  
## 2 34.92007 0.000000 34.826069 7.146055 31.101560 13.613077  
## 3 18.26649 34.826069 0.000000 39.302608 4.816588 46.490611  
## 4 36.12576 7.146055 39.302608 0.000000 35.128568 7.260463  
## 5 14.79314 31.101560 4.816588 35.128568 0.000000 42.245644  
## 6 42.18800 13.613077 46.490611 7.260463 42.245644 0.000000

res.hc <- hclust(d = res.dist, method = "ward.D2")  
  
# cex: label size  
library("factoextra")  
fviz\_dend(res.hc, cex = 1)



# Compute cophentic distance  
res.coph <- cophenetic(res.hc)  
  
# Correlation between cophenetic distance and  
# the original distance  
cor(res.dist, res.coph)

## [1] 0.718701

res.hc2 <- hclust(res.dist, method = "average")  
  
cor(res.dist, cophenetic(res.hc2))

## [1] 0.7257661

# Cut tree into 4 groups  
grp <- cutree(res.hc, k = 4)  
head(grp, n = 4)

## [1] 1 2 3 2

# Number of members in each cluster  
table(grp)

## grp  
## 1 2 3 4   
## 1 5 2 3