Connecticut_County_Monthly_Spacio_Temporal

2022-04-25

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# https://cran.r-project.org/web/packages/surveillance/vignettes/hhh4_spacetime.pdf

# install.packages("surveillance")
# install.packages("spdep")
# install.packages("maps")
# install.packages("maptools")
# install.packages("classInt")
# install.packages("RColorBrewer")
#
# install.packages("rgdal")


library(surveillance)
```

```
## Loading required package: sp
```

```
## Loading required package: xtable
```

```
## This is surveillance 1.20.0. For overview type 'help(surveillance)'.
```

```
library(spdep)
```

```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1; sf_use_s2() is TRUE
```

```
library(maps)
library(maptools)
```

```
## Checking rgeos availability: FALSE
## Please note that 'maptools' will be retired by the end of 2023,
## plan transition at your earliest convenience;
## some functionality will be moved to 'sp'.
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpclib,
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
##
## Attaching package: 'maptools'
```

```
## The following object is masked from 'package:surveillance':
##
##     unionSpatialPolygons
```

```
## The following object is masked from 'package:xtable':
##
##     label
```

```
library(classInt)
library(RColorBrewer)
library(rgdal)
```

```
## Please note that rgdal will be retired by the end of 2023,
## plan transition to sf/stars/terra functions using GDAL and PROJ
## at your earliest convenience.
##
## rgdal: version: 1.5-31, (SVN revision 1171)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.0.4, released 2020/01/28
## Path to GDAL shared files: /usr/share/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ_VERSION: 631]
## Path to PROJ shared files: /usr/share/proj
## Linking to sp version:1.4-7
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
```

Including Plots

You can also embed plots, for example:

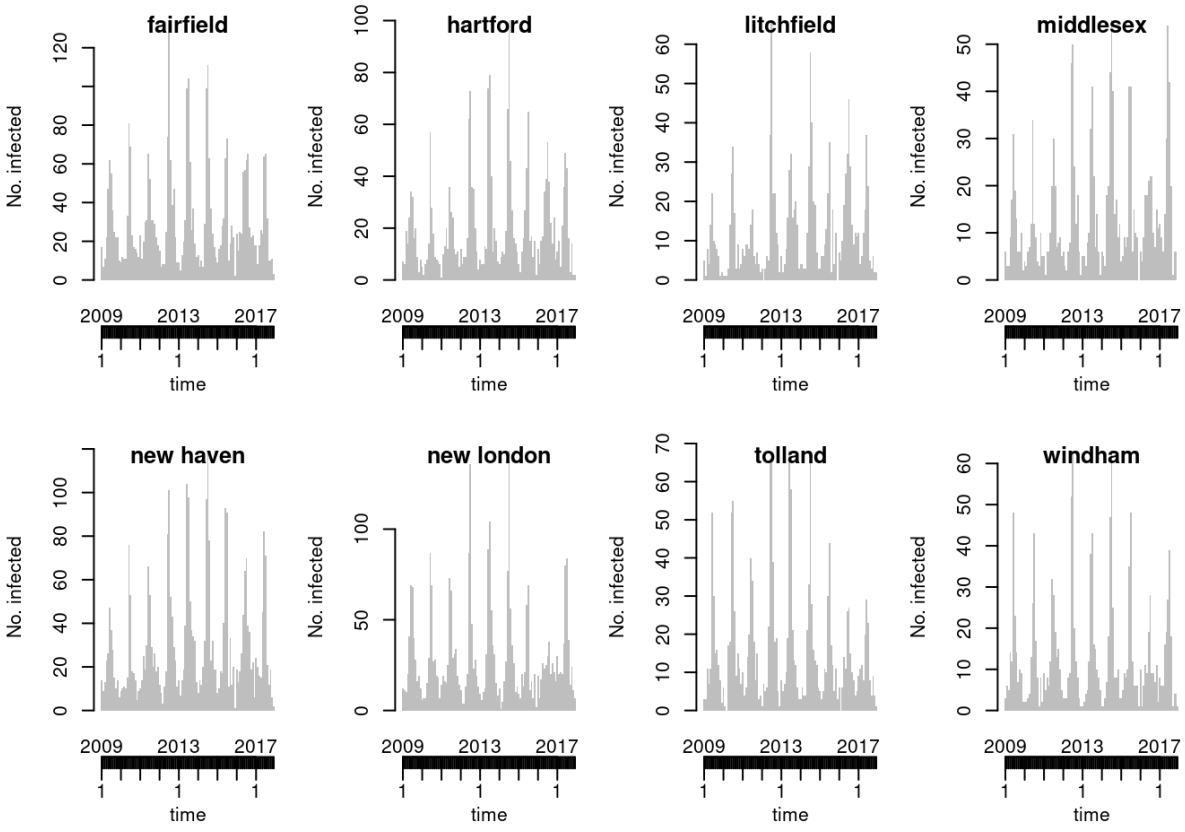
```
## [1] "connecticut,fairfield"  "connecticut,hartford"   "connecticut,litchfield"
## [4] "connecticut,middlesex"  "connecticut,new haven"  "connecticut,new london"
## [7] "connecticut,tolland"    "connecticut,windham"
```

```
## [1] "fairfield"  "hartford"   "litchfield" "middlesex"  "new haven"
## [6] "new london" "tolland"    "windham"
```

```
##    V1 V2 V3 V4  V5  V6 V7 V8
## 1  17  5  5  2  15  14  8  6
## 2  10  4  1  2  10  17  1  3
## 3  28  6  7 11  12  17  6  7
## 4  37  8 14 10  25  49 13 16
## 5  56 20 17 20  26  44 22 29
## 6 157 83 62 51 102 126 76 58
```

```
## -- An object of class sts --
## freq:         12
## start:        2009 1
## dim(observed):    108 8
##
## Head of observed:
##      fairfield hartford litchfield middlesex new haven new london tolland
## [1,]        17        7          5         6        14         12       3
##      windham
## [1,]       3
```

```
plot(cdcCon, type = observed ~ time | unit, same.scale = FALSE, col = "grey")
```




```
f_S1 <- addSeason2formula(f = ~ 1, S = 1, period = 10)
result0 <- hhh4(cdcCon, control = list(end = list(f = f_S1),
                                       family = "Poisson"))
summary(result0)
```

```
##
## Call:
## hhh4(stsObj = cdcCon, control = list(end = list(f = f_S1), family = "Poisson"))
##
## Coefficients:
##                         Estimate   Std. Error
## end.1                    3.029402   0.007524
## end.sin(2 * pi * t/10)  -0.047476   0.010603
## end.cos(2 * pi * t/10)  -0.039312   0.010662
##
## Log-likelihood:   -9226.53
## AIC:              18459.05
## BIC:              18473.31
##
## Number of units:        8
## Number of time points:  107
```

```
result1 <- update(result0, family = "NegBin1")
summary(result1)
```

```
##
## Call:
## hhh4(stsObj = object$stsObj, control = control)
##
## Coefficients:
##                         Estimate  Std. Error
## end.1                    3.02943   0.03106
## end.sin(2 * pi * t/10)  -0.04522   0.04349
## end.cos(2 * pi * t/10)  -0.03828   0.04449
## overdisp                 0.77655   0.03721
##
## Log-likelihood:   -3456.3
## AIC:              6920.6
## BIC:              6939.61
##
## Number of units:        8
## Number of time points:  107
```

```
result2 <- update(result1, ar = list(f = ~ 1))

summary(result2)
```

```
##
## Call:
## hhh4(stsObj = object$stsObj, control = control)
##
## Coefficients:
##                         Estimate  Std. Error
## ar.1                    -0.20659   0.04513
## end.1                    1.47202   0.09508
## end.sin(2 * pi * t/10)   0.04226   0.09385
## end.cos(2 * pi * t/10)  -0.16454   0.09532
## overdisp                 0.38344   0.02137
##
## Log-likelihood:   -3161.52
## AIC:              6333.04
## BIC:              6356.8
##
## Number of units:        8
## Number of time points:  107
```

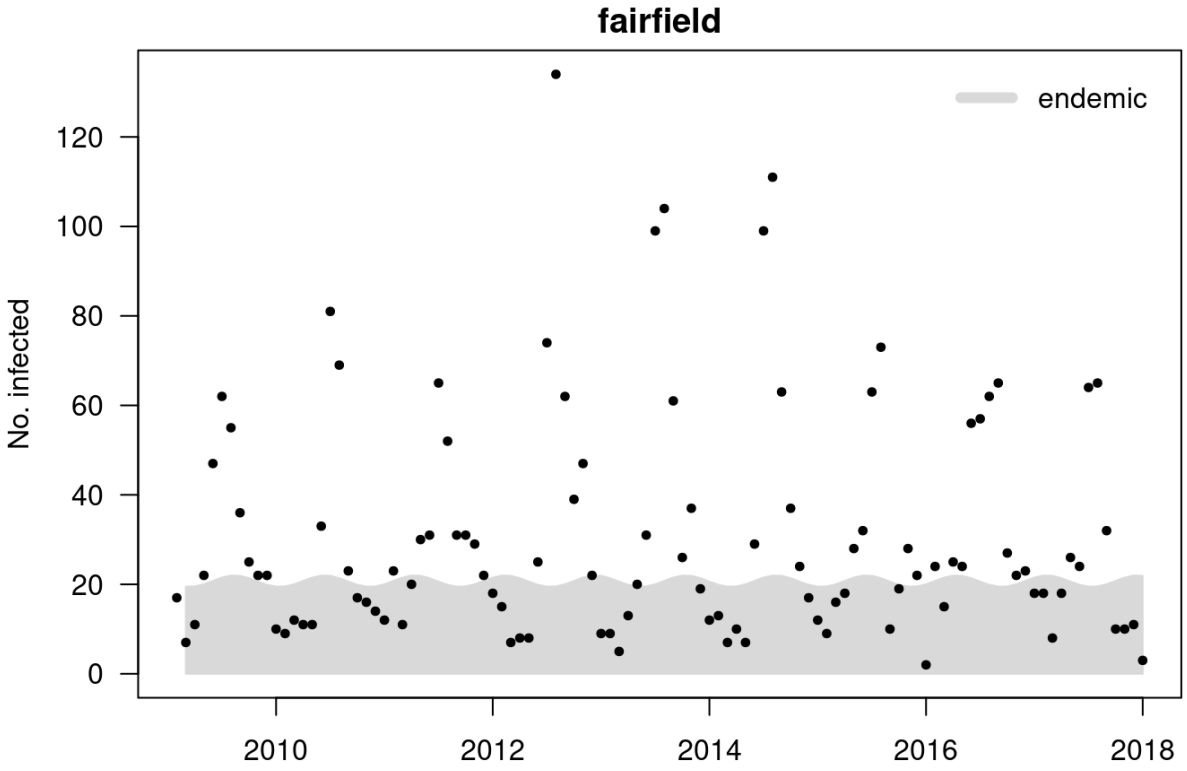
```
AIC(result0, result1, result2)
```

```
##         df       AIC
## result0  3 18459.054
## result1  4  6920.603
## result2  5  6333.038
```

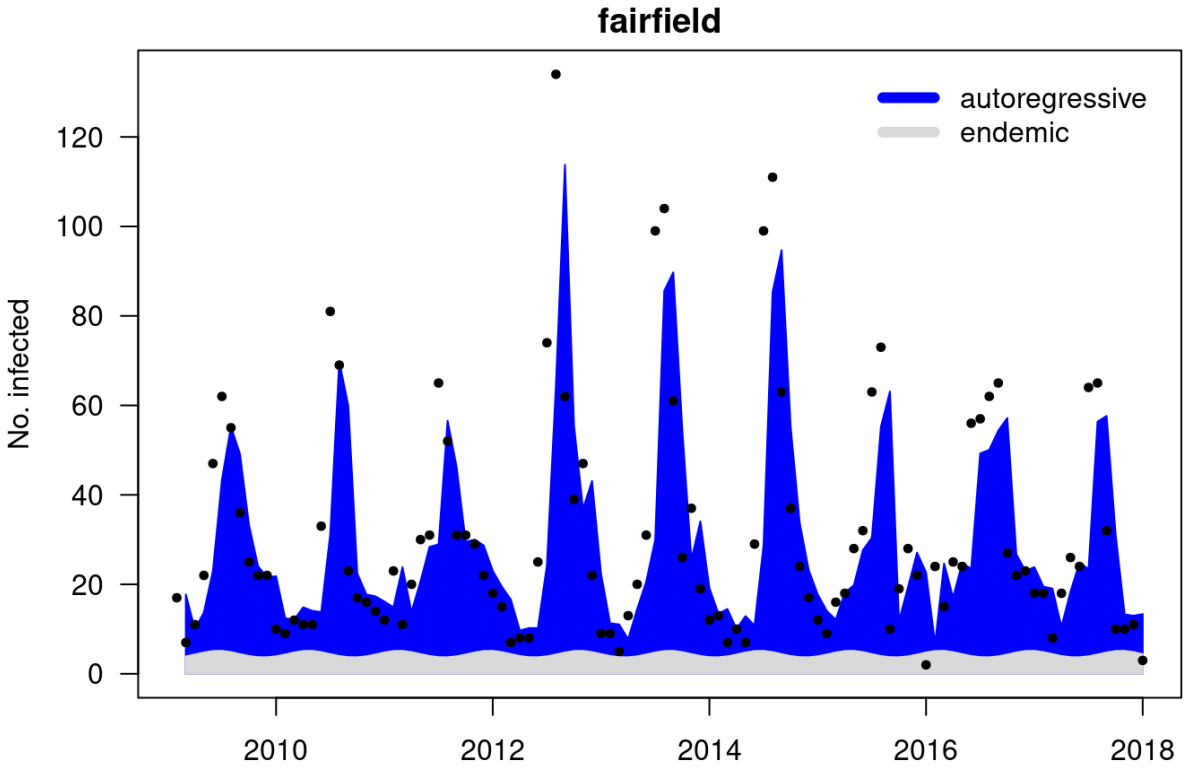
```
coef(result2, se = TRUE,    # also return standard errors
     amplitudeShift = TRUE, # transform sine/cosine coefficients
     # to amplitude/shift parameters
     idx2Exp = TRUE)
```

```
##                        Estimate Std. Error
## exp(ar.1)             0.8133561 0.03670402
## exp(end.1)            4.3580197 0.41437646
## end.A(2 * pi * t/10)  0.1698831 0.09555021
## end.s(2 * pi * t/10) -1.3193975 0.07036980
## overdisp              0.3834365 0.02136550
```

```
plot(result0)
```



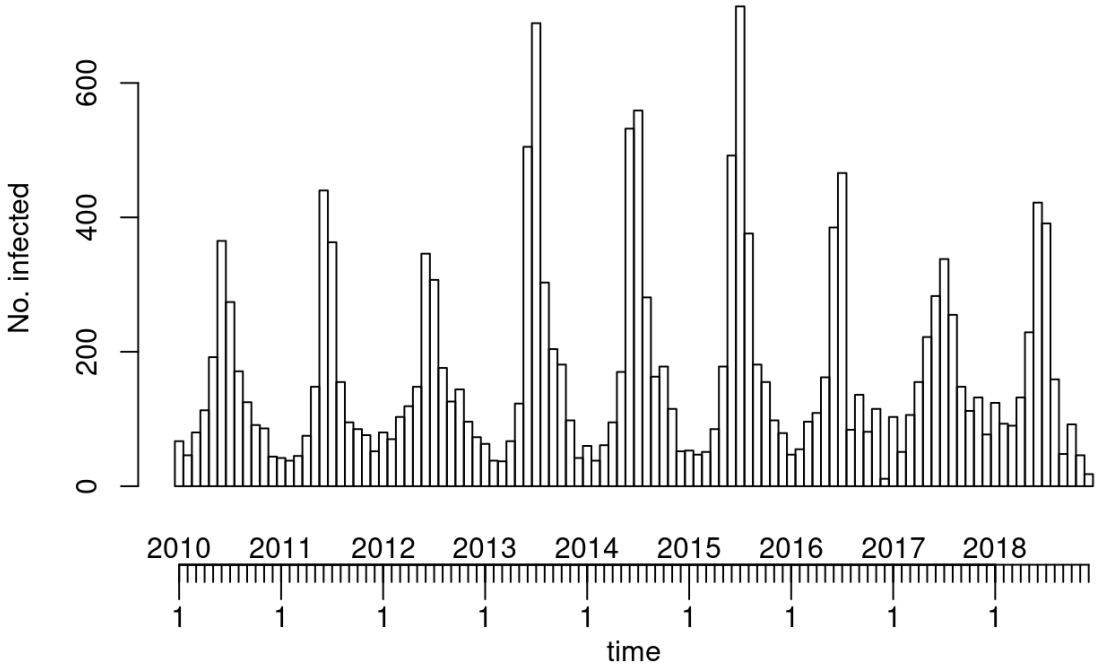

```
plot(result2)
```



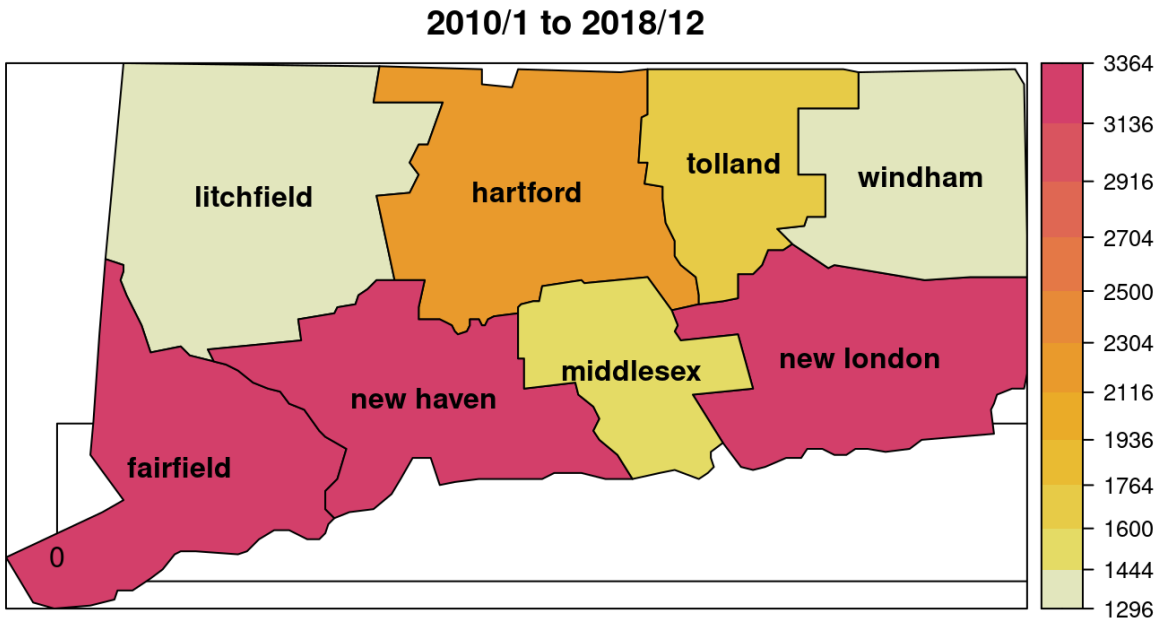
```
#dfDisprogNb <- sts(month,observed, start = c(2009, 1), frequency = 12, neighbourhood = county_state_nbOrde
r, map = mn.poly)
#working
# Add population to do spatio-temporal analysis and see the effect of surrounding counties on the current co
unty
#dfDisprogNb <- sts(observed, start = c(2009, 1), frequency = 12, neighbourhood = county_state_nbOrder, map
 = mn.poly)


#census_connecticut_2010_2018.csv


popConn = read.csv("census_connecticut_county_2010_2018.csv")

popConn = popConn[ -c(1) ]

popFrac = popConn/100000

colnames(popFrac) = colnames

dfDisprogNb <- sts(observed, start = c(2010, 1), frequency = 12, population = data.matrix(popFrac),
                   neighbourhood = county_state_nbOrder, map = mn.poly)


dfDisprogNb
```

```
## -- An object of class sts --
## freq:         12
## start:        2010 1
## dim(observed):    108 8
##
## Head of observed:
##      fairfield hartford litchfield middlesex new haven new london tolland
## [1,]        17        7          5         6        14         12       3
##      windham
## [1,]       3
##
## map:
## Object of class SpatialPolygons
## Coordinates:
##         min       max
## x -73.72248 -71.78015
## y  41.01232  42.04937
## Is projected: NA
## proj4string : [NA]
## Features    : 8
##
## Head of neighbourhood:
##           fairfield hartford litchfield middlesex new haven new london tolland
## fairfield         0        0          1         0         1          0       0
##           windham
## fairfield       0
```

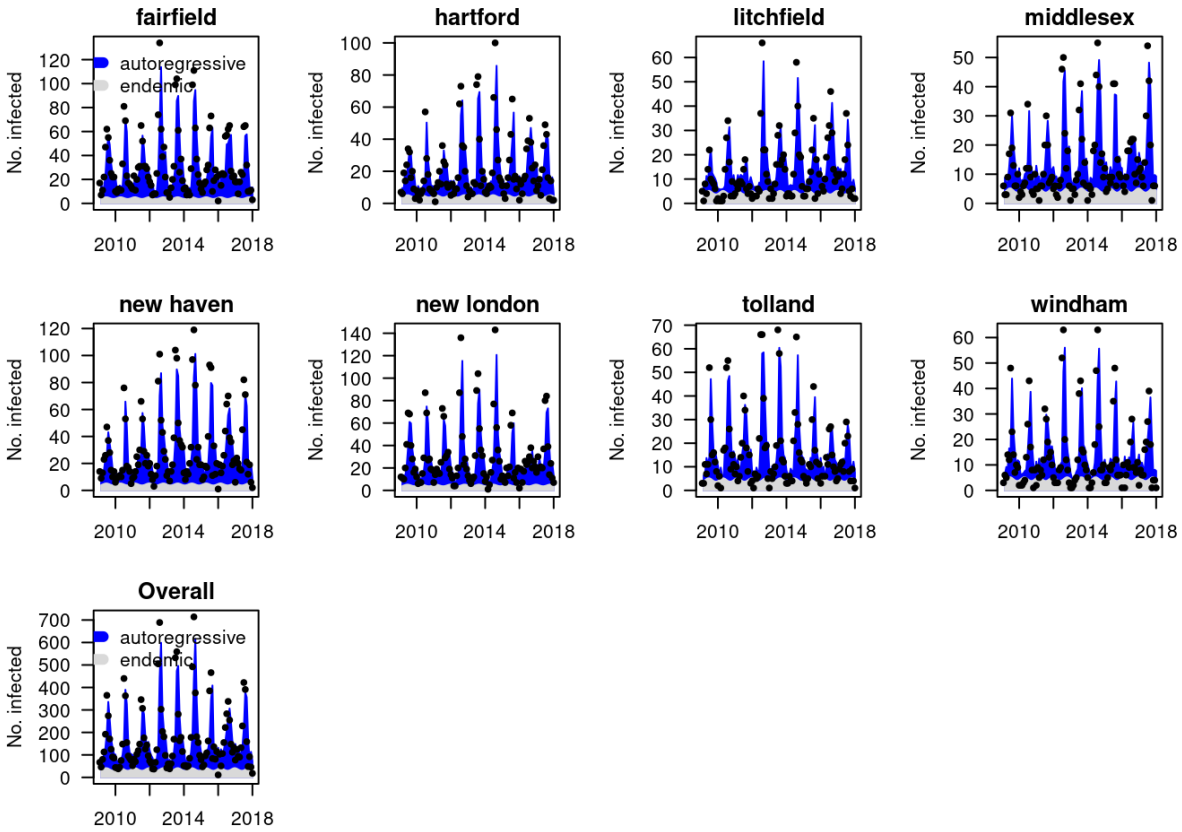
```
plot(dfDisprogNb, type = observed ~ time)
```




```
plot(dfDisprogNb, type = observed ~ unit,
     labels = list(font = 2), colorkey = list(space = "right"),
     sp.layout = layout.scalebar(dfDisprogNb@map, corner = c(0.05, 0.05),
                                 scale = 10, labels = c("0", "50 km"), height = 0.03))
```



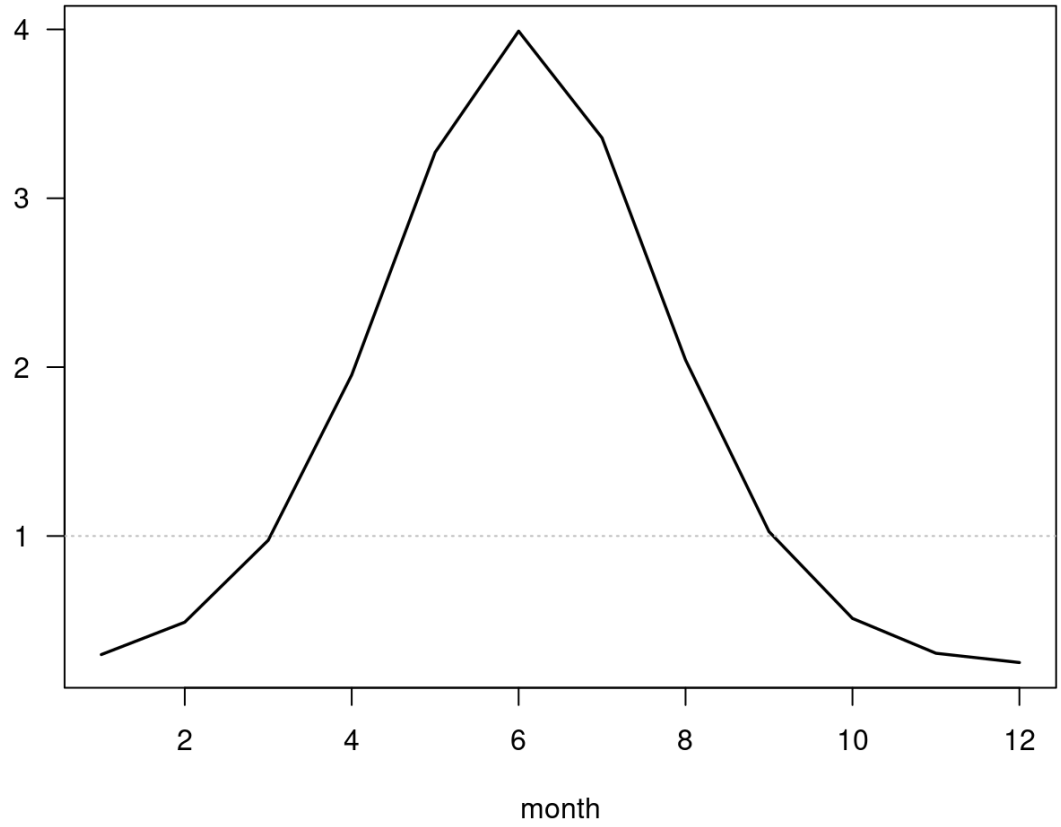
```
districts2plot <- which(colSums(observed) > 50)
par(mfrow = c(3,4), mar = c(3, 5, 2, 1), las = 1)
plot(result2, type = "fitted", units = districts2plot, hide0s = TRUE, par.settings = NULL, legend = 1)
plot(result2, type = "fitted", total = TRUE, hide0s = TRUE, par.settings = NULL, legend = TRUE) -> fitted_co
mponents
```



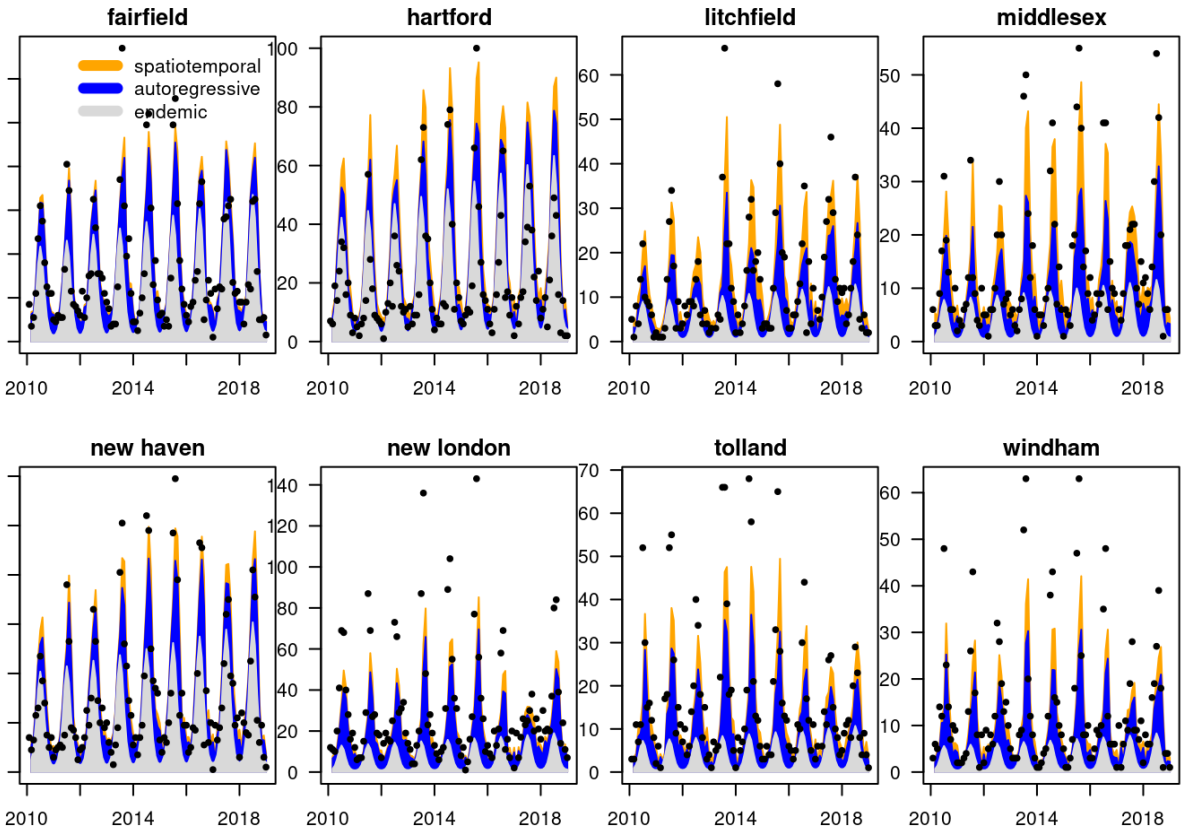

```
lymeModel_basic <- list(end = list(f = addSeason2formula(~1 + t, period = 12), offset = data.matrix(popFra
c)),
ar = list(f = ~1), ne = list(f = ~1, weights = county_state_nbOrder == 1), family = "NegBin1")

lymeFit_basic <- hhh4(stsObj = dfDisprogNb, control = lymeModel_basic)
summary(lymeModel_basic, idx2Exp = TRUE, amplitudeShift = TRUE, maxEV = TRUE)
```

```
##        Length Class  Mode
## end    2      -none- list
## ar     1      -none- list
## ne     2      -none- list
## family 1      -none- character
```

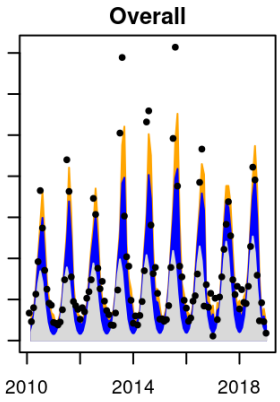
```
plot(lymeFit_basic, type = "season", components = "end", main = "")
```




```
districts2plot <- which(colSums(observed) > 50)

par(mfrow = c(2,4), mar = c(3, 1, 2, 1), las = 1)

plot(lymeFit_basic, type = "fitted", units = districts2plot, hide0s = TRUE, par.settings = NULL, legend = 1)
```




```
plot(lymeFit_basic, type = "fitted", total = TRUE, hide0s = TRUE, par.settings = NULL, legend = FALSE) -> fi
tted_components
```



```
# data("measlesWeserEms")
#
# head(measlesWeserEms)
#
# measlesWeserEms@populationFrac
#
# measlesModel_basic <- list(end = list(f = addSeason2formula(~1 + t, period = measlesWeserEms@freq), offset
= population(measlesWeserEms)),
# ar = list(f = ~1), ne = list(f = ~1, weights = neighbourhood(measlesWeserEms) == 1), family = "NegBin1")
#
# measlesFit_basic <- hhh4(stsObj = measlesWeserEms, control = measlesModel_basic)
# summary(measlesFit_basic, idx2Exp = TRUE, amplitudeShift = TRUE, maxEV = TRUE)
# plot(measlesFit_basic, type = "season", components = "end", main = "")
#
#
# districts2plot <- which(colSums(observed(measlesWeserEms)) > 50)
# par(mfrow = c(2,3), mar = c(3, 5, 2, 1), las = 1)
# plot(measlesFit_basic, type = "fitted", units = districts2plot, hide0s = TRUE, par.settings = NULL, legend
= 1)
# plot(measlesFit_basic, type = "fitted", total = TRUE, hide0s = TRUE, par.settings = NULL, legend = FALSE)
 -> fitted_components
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.