hhh4: Endemic-epidemic modeling of areal count time series

Sebastian Meyer[*]
Friedrich-Alexander-Universität Erlangen-Nürnberg

Leonhard Held
University of Zurich

Michael Höhle
Stockholm University

Abstract

The availability of geocoded health data and the inherent temporal structure of communicable diseases have led to an increased interest in statistical models and software for spatio-temporal data with epidemic features. The R package **surveillance** can handle various levels of aggregation at which infective events have been recorded. This vignette illustrates the analysis of area-level time series of counts using the endemic-epidemic multivariate time-series model "hhh4" described in, e.g., Meyer and Held (2014, Section 3). See vignette("hhh4") for a more general introduction to hhh4 models, including the univariate and non-spatial bivariate case. We first describe the general modeling approach and then exemplify data handling, model fitting, visualization, and simulation methods for weekly counts of measles infections by district in the Weser-Ems region of Lower Saxony, Germany, 2001–2002.

Keywords: areal time series of counts, endemic-epidemic modeling, infectious disease epidemiology, branching process with immigration.

1. Model class: hhh4

An endemic-epidemic multivariate time-series model for infectious disease counts Y_{it} from units $i = 1, \dots, I$ during periods $t = 1, \dots, T$ was proposed by Held, Höhle, and Hofmann (2005) and was later extended in a series of papers (Paul, Held, and Toschke 2008; Paul and Held 2011; Held and Paul 2012; Meyer and Held 2014). In its most general formulation, this so-called "hhh4" model assumes that, conditional on past observations, Y_{it} has a negative binomial distribution with mean

$$\mu_{it} = e_{it}\,\nu_{it} + \lambda_{it}\,Y_{i,t-1} + \phi_{it} \sum_{j \neq i} w_{ji}\,Y_{j,t-1} \tag{1}$$

and overdispersion parameter $\psi_i > 0$ such that the conditional variance of Y_{it} is $\mu_{it}(1+\psi_i\mu_{it})$. Shared overdispersion parameters, e.g., $\psi_i \equiv \psi$, are supported as well as replacing the negative binomial by a Poisson distribution, which corresponds to the limit $\psi_i \equiv 0$.

Similar to the point process models in vignette("twinstim") and vignette("twinSIR"), the mean (1) decomposes additively into endemic and epidemic components. The endemic

---

[*]Author of correspondence: seb.meyer@fau.de

mean is usually modeled proportional to an offset of expected counts e_{it}. In spatial applications of the multivariate `hhh4` model as in this paper, the "unit" i refers to a geographical region and we typically use (the fraction of) the population living in region i as the endemic offset. The observation-driven epidemic component splits up into autoregressive effects, i.e., reproduction of the disease within region i, and neighborhood effects, i.e., transmission from other regions j. Overall, Equation 1 becomes a rich regression model by allowing for log-linear predictors in all three components:

$$\log(\nu_{it}) = \alpha_i^{(\nu)} + \boldsymbol{\beta}^{(\nu)\top} \boldsymbol{z}_{it}^{(\nu)} \,, \tag{2}$$

$$\log(\lambda_{it}) = \alpha_i^{(\lambda)} + \boldsymbol{\beta}^{(\lambda)\top} \boldsymbol{z}_{it}^{(\lambda)} \,, \tag{3}$$

$$\log(\phi_{it}) = \alpha_i^{(\phi)} + \boldsymbol{\beta}^{(\phi)\top} \boldsymbol{z}_{it}^{(\phi)} \,. \tag{4}$$

The intercepts of these predictors can be assumed identical across units, unit-specific, or random (and possibly correlated). The regression terms often involve sine-cosine effects of time to reflect seasonally varying incidence, but may, e.g., also capture heterogeneous vaccination coverage (Herzog, Paul, and Held 2011). Data on infections imported from outside the study region may enter the endemic component (Geilhufe, Held, Skrøvseth, Simonsen, and Godtliebsen 2014), which generally accounts for cases not directly linked to other observed cases, e.g., due to edge effects.

For a single time series of counts Y_t, `hhh4` can be regarded as an extension of `glm.nb` from package **MASS** (Ripley and Venables 2025) to account for autoregression. See the `vignette("hhh4")` for examples of modeling univariate and bivariate count time series using `hhh4`. With multiple regions, spatio-temporal dependence is adopted by the third component in Equation 1 with weights w_{ji} reflecting the flow of infections from region j to region i. These transmission weights may be informed by movement network data (Paul *et al.* 2008; Geilhufe *et al.* 2014), but may also be estimated parametrically. A suitable choice to reflect epidemiological coupling between regions (Keeling and Rohani 2008, Chapter 7) is a power-law distance decay $w_{ji} = o_{ji}^{-d}$ defined in terms of the adjacency order o_{ji} in the neighborhood graph of the regions (Meyer and Held 2014). Note that we usually normalize the transmission weights such that $\sum_i w_{ji} = 1$, i.e., the $Y_{j,t-1}$ cases are distributed among the regions proportionally to the jth row vector of the weight matrix (w_{ji}).

Likelihood inference for the above multivariate time-series model has been established by Paul and Held (2011) with extensions for parametric neighborhood weights by Meyer and Held (2014). Supplied with the analytical score function and Fisher information, the function `hhh4` by default uses the quasi-Newton algorithm available through the R function `nlminb` to maximize the log-likelihood. Convergence is usually fast even for a large number of parameters. If the model contains random effects, the penalized and marginal log-likelihoods are maximized alternately until convergence. Computation of the marginal Fisher information is accelerated using the **Matrix** package (Bates, Maechler, and Jagan 2025).

2. Data structure: `sts`

In public health surveillance, routine reports of infections to public health authorities give rise to spatio-temporal data, which are usually made available in the form of aggregated counts by region and period. The Robert Koch Institute (RKI) in Germany, for example, maintains

a database of cases of notifiable diseases, which can be queried via the *SurvStat@RKI* online service (https://survstat.rki.de). To exemplify area-level `hhh4` models in the remainder of this manuscript, we use weekly counts of measles infections by district in the Weser-Ems region of Lower Saxony, Germany, 2001–2002, downloaded from *SurvStat@RKI* (as of Annual Report 2005). These data are contained in **surveillance** as `data("measlesWeserEms")` – an object of the S4-class `sts` ("surveillance time series") used for data input in `hhh4` models and briefly introduced below. See Höhle and Mazick (2010) and Salmon, Schumacher, and Höhle (2016) for more detailed descriptions of this class, which is also used for the prospective aberration detection facilities of the **surveillance** package.

The epidemic modeling of multivariate count time series essentially involves three data matrices: a $T \times I$ matrix of the observed counts, a corresponding matrix with potentially time-varying population numbers (or fractions), and an $I \times I$ neighborhood matrix quantifying the coupling between the I units. In our example, the latter consists of the adjacency orders o_{ji} between the districts. A map of the districts in the form of a `SpatialPolygons` object (defined by the **sp** package of Pebesma and Bivand 2025) can be used to derive the matrix of adjacency orders automatically using the functions `poly2adjmat` and `nbOrder`, where the former wraps functionality of package **spdep** (Bivand 2022):

```
R> weserems_adjmat <- poly2adjmat(map)
R> weserems_nbOrder <- nbOrder(weserems_adjmat)
```

Visual inspection of the adjacencies identified by `poly2adjmat` is recommended, e.g., via labelling each district with the number of its neighbors, i.e., `rowSums(weserems_adjmat)`. If adjacencies are not detected, this is probably due to sliver polygons. In that case either increase the `snap` tolerance in `poly2adjmat` or use **rmapshaper** (Teucher and Russell 2020) to simplify and snap adjacent polygons in advance.

Given the aforementioned ingredients, the `sts` object `measlesWeserEms` has been constructed as follows:

```
R> measlesWeserEms <- sts(counts, start = c(2001, 1), frequency = 52,
+    population = populationFrac, neighbourhood = weserems_nbOrder, map = map)
```

Here, `start` and `frequency` have the same meaning as for classical time-series objects of class `ts`, i.e., (year, sample number) of the first observation and the number of observations per year. Note that `data("measlesWeserEms")` constitutes a corrected version of `data("measles.weser")` originally analyzed by Held *et al.* (2005, Section 3.2). Differences are documented on the associated help page.

We can visualize such `sts` data in four ways: individual time series, overall time series, map of accumulated counts by district, or animated maps. For instance, the two plots in Figure 1 have been generated by the following code:

```
R> plot(measlesWeserEms, type = observed ~ time)
R> plot(measlesWeserEms, type = observed ~ unit,
+    population = measlesWeserEms@map$POPULATION / 100000,
+    labels = list(font = 2), colorkey = list(space = "right"),
+    sp.layout = layout.scalebar(measlesWeserEms@map, corner = c(0.05, 0.05),
+      scale = 50, labels = c("0", "50 km"), height = 0.03))
```
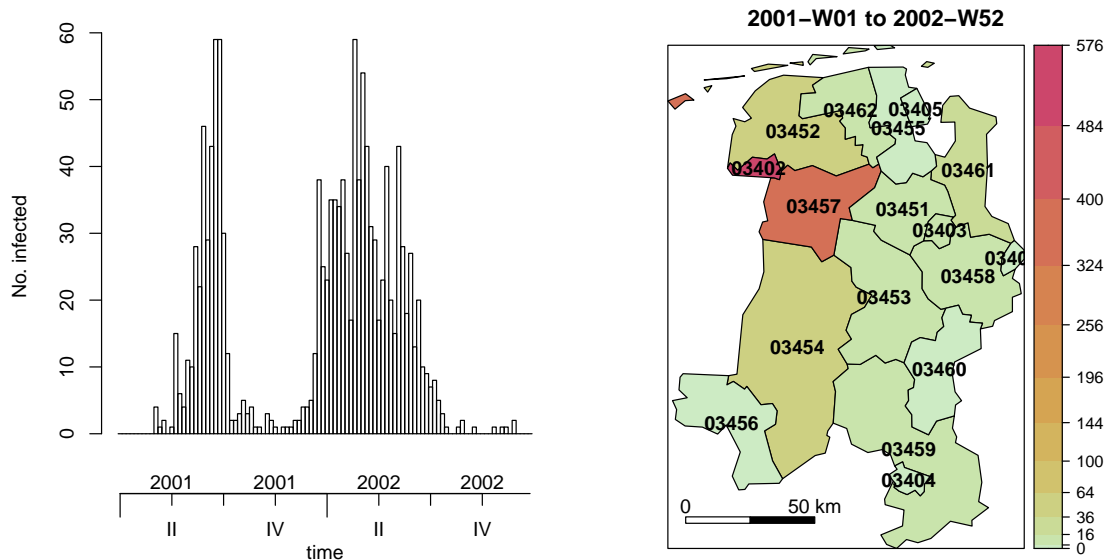
(a) Time series of weekly counts. (b) Disease incidence (per 100 000 inhabitants).

Figure 1: Measles infections in the Weser-Ems region, 2001–2002.

The overall time-series plot in Figure 1a reveals strong seasonality in the data with slightly different patterns in the two years. The spatial plot in Figure 1b is a tweaked `spplot` (package **sp**) with colors from **colorspace** (Ihaka, Murrell, Hornik, Fisher, Stauffer, Wilke, McWhite, and Zeileis 2024) using $\sqrt{}$-equidistant cut points.

The default plot `type` is `observed ~ time | unit` and displays the district-specific time series. Here we show the output of the equivalent `autoplot`-method (Figure 2), which is based on and requires **ggplot2** (Wickham 2016):

```
R> if (require("ggplot2")) {
+     autoplot(measlesWeserEms, units = which(colSums(observed(measlesWeserEms)) > 0))
+ } else plot(measlesWeserEms, units = which(colSums(observed(measlesWeserEms)) > 0))
```

The districts 03401 (SK Delmenhorst) and 03405 (SK Wilhelmshaven) without any reported cases are excluded in Figure 2. Obviously, the districts have been affected by measles to a very heterogeneous extent during these two years.

An animation of the data can be easily produced as well. We recommend to use converters of the **animation** package (Xie 2013), e.g., to watch the series of plots in a web browser. The following code will generate weekly disease maps during the year 2001 with the respective total number of cases shown in a legend and – if package **gridExtra** (Auguie 2017) is available – an evolving time-series plot at the bottom:

```
R> animation::saveHTML(
+     animate(measlesWeserEms, tps = 1:52, total.args = list()),
+     title = "Evolution of the measles epidemic in the Weser-Ems region, 2001",
+     ani.width = 500, ani.height = 600)
```
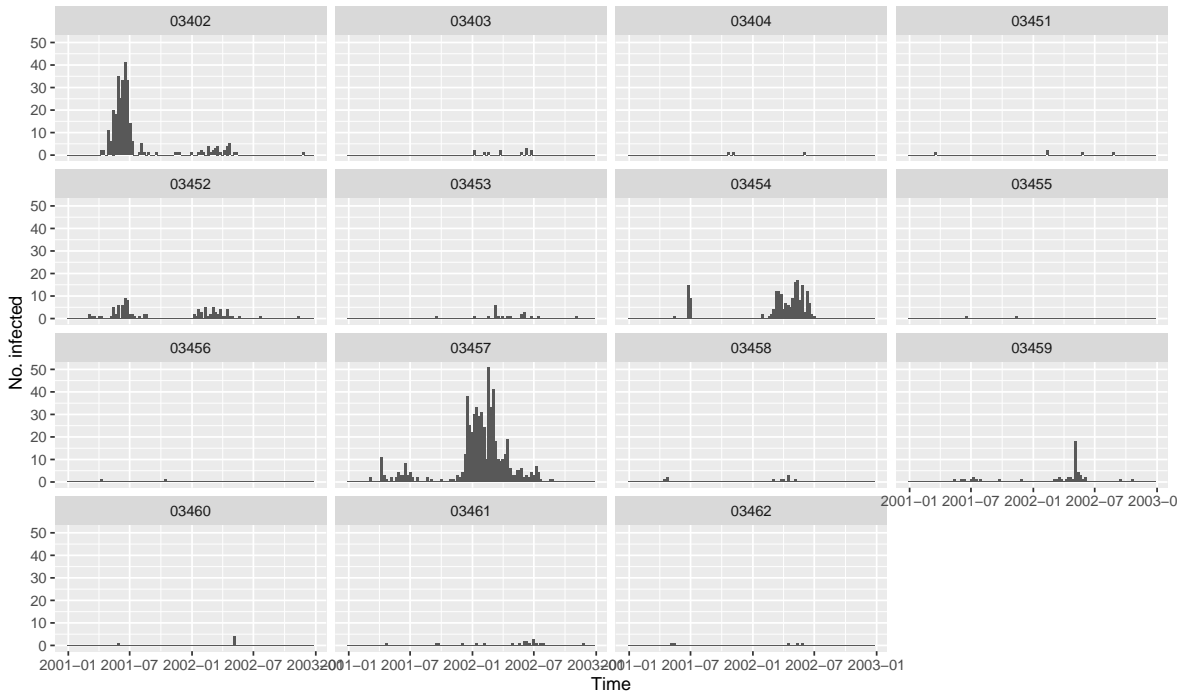
Figure 2: Count time series of the 15 affected districts.

3. Modeling and inference

For multivariate surveillance time series of counts such as the `measlesWeserEms` data, the function `hhh4` fits models of the form (1) via (penalized) maximum likelihood. We start by modeling the measles counts in the Weser-Ems region by a slightly simplified version of the original negative binomial model used by Held *et al.* (2005). Instead of district-specific intercepts $\alpha_i^{(\nu)}$ in the endemic component, we first assume a common intercept $\alpha^{(\nu)}$ in order to not be forced to exclude the two districts without any reported cases of measles. After the estimation and illustration of this basic model, we will discuss the following sequential extensions: covariates (district-specific vaccination coverage), estimated transmission weights, and random effects to eventually account for unobserved heterogeneity of the districts.

3.1. Basic model

Our initial model has the following mean structure:

$$\mu_{it} = e_i\,\nu_t + \lambda\,Y_{i,t-1} + \phi \sum_{j\neq i} w_{ji}Y_{j,t-1} \,, \tag{5}$$

$$\log(\nu_t) = \alpha^{(\nu)} + \beta_t t + \gamma\sin(\omega t) + \delta\cos(\omega t) \,. \tag{6}$$

To account for temporal variation of disease incidence, the endemic log-linear predictor ν_t incorporates an overall trend and a sinusoidal wave of frequency $\omega = 2\pi/52$. As a basic district-specific measure of disease incidence, the population fraction e_i is included as a multiplicative offset. The epidemic parameters $\lambda = \exp(\alpha^{(\lambda)})$ and $\phi = \exp(\alpha^{(\phi)})$ are assumed homogeneous across districts and constant over time. Furthermore, we define $w_{ji} = \mathbb{I}(j \sim i) = \mathbb{I}(o_{ji} = 1)$

for the time being, which means that the epidemic can only arrive from directly adjacent districts. This `hhh4` model transforms into the following list of `control` arguments:

```
R> measlesModel_basic <- list(
+    end = list(f = addSeason2formula(~1 + t, period = frequency(measlesWeserEms)),
+               offset = population(measlesWeserEms)),
+    ar = list(f = ~1),
+    ne = list(f = ~1, weights = neighbourhood(measlesWeserEms) == 1),
+    family = "NegBin1")
```

The formulae of the three predictors $\log \nu_t$, $\log \lambda$ and $\log \phi$ are specified as element `f` of the `end`, `ar`, and `ne` lists, respectively. For the endemic formula we use the convenient function `addSeason2formula` to generate the sine-cosine terms, and we take the multiplicative `offset` of population fractions e_i from the `measlesWeserEms` object. The autoregressive part only consists of the intercept $\alpha^{(\lambda)}$, whereas the neighborhood component specifies the intercept $\alpha^{(\phi)}$ and also the matrix of transmission `weights` (w_{ji}) to use – here a simple indicator of first-order adjacency. The chosen `family` corresponds to a negative binomial model with a common overdispersion parameter ψ for all districts. Alternatives are `"Poisson"`, `"NegBinM"` (ψ_i), or a factor determining which groups of districts share a common overdispersion parameter. Together with the data, the complete list of control arguments is then fed into the `hhh4` function to estimate the model:

```
R> measlesFit_basic <- hhh4(stsObj = measlesWeserEms, control = measlesModel_basic)
```

The fitted model is summarized below:

```
R> summary(measlesFit_basic, idx2Exp = TRUE, amplitudeShift = TRUE, maxEV = TRUE)
```

```
Call:
hhh4(stsObj = measlesWeserEms, control = measlesModel_basic)

Coefficients:
                      Estimate  Std. Error
exp(ar.1)              0.64540    0.07927
exp(ne.1)              0.01581    0.00420
exp(end.1)             1.08025    0.27884
exp(end.t)             1.00119    0.00426
end.A(2 * pi * t/52)   1.16423    0.19212
end.s(2 * pi * t/52)  -0.63436    0.13350
overdisp               2.01384    0.28544

Epidemic dominant eigenvalue:  0.72

Log-likelihood:    -971.7
AIC:                1957
BIC:                1996

Number of units:         17
Number of time points:  103
```

The `idx2Exp` argument of the `summary` method requests the estimates for λ, ϕ, $\alpha^{(\nu)}$ and $\exp(\beta_t)$ instead of their respective internal log-values. For instance, `exp(end.t)` represents the seasonality-adjusted factor by which the basic endemic incidence increases per week. The `amplitudeShift` argument transforms the internal coefficients γ and δ of the sine-cosine terms to the amplitude A and phase shift φ of the corresponding sinusoidal wave $A\sin(\omega t + \varphi)$ in $\log \nu_t$ (Paul *et al.* 2008). The resulting multiplicative effect of seasonality on ν_t is shown in Figure 3 produced by:

```R
R> plot(measlesFit_basic, type = "season", components = "end", main = "")
```
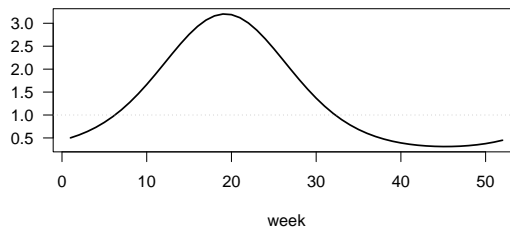

Figure 3: Estimated multiplicative effect of seasonality on the endemic mean.

The epidemic potential of the process as determined by the parameters λ and ϕ is best investigated by a combined measure: the dominant eigenvalue (`maxEV`) of the matrix $\boldsymbol{\Lambda}$ which has the entries $(\Lambda)_{ii} = \lambda$ on the diagonal and $(\Lambda)_{ij} = \phi w_{ji}$ for $j \neq i$ (Paul *et al.* 2008). If the dominant eigenvalue is smaller than 1, it can be interpreted as the epidemic proportion of disease incidence. In the above model, the estimate is 72%.

Another way to judge the relative importance of the three model components is via a plot of the fitted mean components along with the observed counts. Figure 4 shows this for the five districts with more than 50 cases as well as for the sum over all districts:

```R
R> districts2plot <- which(colSums(observed(measlesWeserEms)) > 50)
R> par(mfrow = c(2,3), mar = c(3, 5, 2, 1), las = 1)
R> plot(measlesFit_basic, type = "fitted", units = districts2plot,
+      hide0s = TRUE, par.settings = NULL, legend = 1)
R> plot(measlesFit_basic, type = "fitted", total = TRUE,
+      hide0s = TRUE, par.settings = NULL, legend = FALSE) -> fitted_components
```

We can see from the plots that the largest portion of the fitted mean indeed results from the within-district autoregressive component with very little contribution of cases from adjacent districts and a rather small endemic incidence.

The `plot` method invisibly returns the component values in a list of matrices (one by unit). In the above code, we have assigned the result from plotting the overall fit (via `total = TRUE`) to the object `fitted_components`. Here we show the values for the weeks 20 to 22 (corresponding to the weeks 21 to 23 of the measles time series):

```R
R> fitted_components$Overall[20:22,]
```

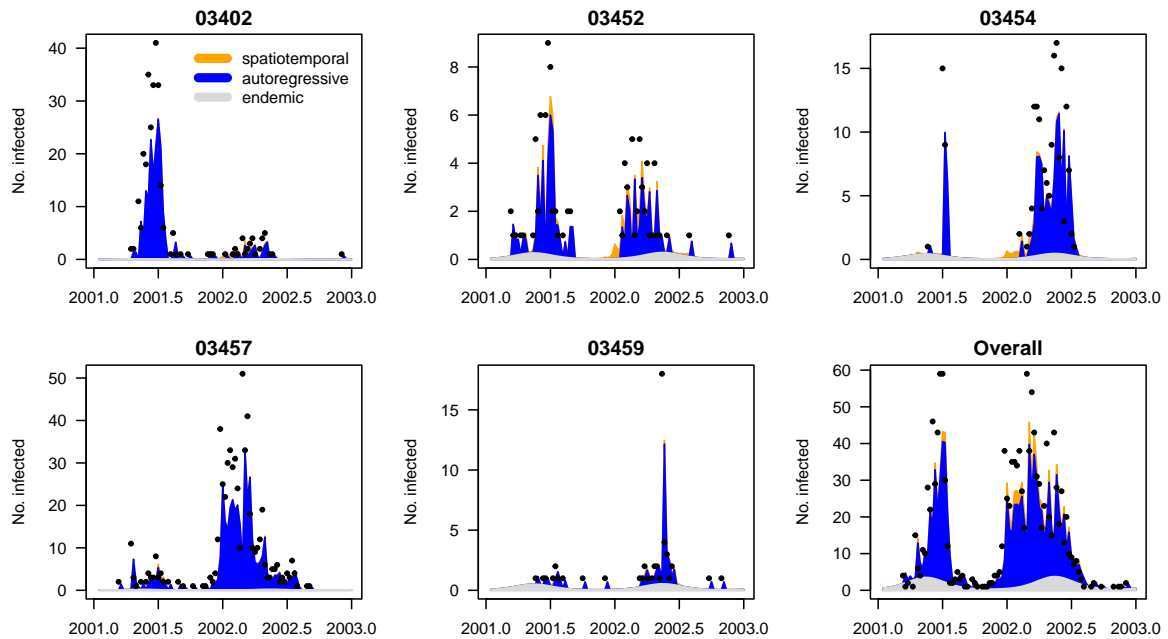|	mean	epidemic	endemic	epi.own	epi.neighbours	ar.exppred	ne.exppred	end.exppred
[1,]	22.57	19.11	3.453	18.07	1.0431	10.97	0.2687	58.70
[2,]	18.41	15.08	3.329	14.20	0.8851	10.97	0.2687	56.59
[3,]	34.73	31.57	3.158	29.69	1.8808	10.97	0.2687	53.69

Figure 4: Fitted components in the initial model `measlesFit_basic` for the five districts with more than 50 cases as well as summed over all districts (bottom right). Dots are only drawn for positive weekly counts.

The first column of this matrix refers to the fitted mean (epidemic + endemic). The four following columns refer to the epidemic (own + neighbours), endemic, autoregressive ("own"), and neighbourhood components of the mean. The last three columns refer to the point estimates of λ, ϕ, and ν_t, respectively. These values allow us to calculate the (time-averaged) proportions of the mean explained by the different components:

```
R> colSums(fitted_components$Overall)[3:5] / sum(fitted_components$Overall[,1])
```

```
        endemic         epi.own epi.neighbours
        0.14876         0.76060        0.09063
```

Note that the "epidemic proportion" obtained here (85%) is a function of the observed time series (so could be called "empirical"), whereas the dominant eigenvalue calculated further above is a theoretical property derived from the autoregressive parameters alone.

Finally, the `overdisp` parameter from the model summary and its 95% confidence interval

```
R> confint(measlesFit_basic, parm = "overdisp")
```

```
          2.5 % 97.5 %
overdisp 1.454  2.573
```

suggest that a negative binomial distribution with overdispersion is more adequate than a Poisson model corresponding to $\psi = 0$. We can underpin this finding by an AIC comparison, taking advantage of the convenient `update` method for `hhh4` fits:

```
R> AIC(measlesFit_basic, update(measlesFit_basic, family = "Poisson"))

                                           df  AIC
measlesFit_basic                            7 1957
update(measlesFit_basic, family = "Poisson")  6 2479
```

Other plot `types` and methods for fitted `hhh4` models as listed in Table 1 will be applied in the course of the following model extensions.

Display	Extract	Modify	Other
print	nobs	update	predict
summary	coef		simulate
plot	fixef		pit
	ranef		scores
	vcov		calibrationTest
	confint		all.equal
	coeflist		*oneStepAhead*
	logLik		
	residuals		
	terms		
	formula		
	getNEweights		

Table 1: Generic and *non-generic* functions applicable to `hhh4` objects.

3.2. Covariates

The `hhh4` model framework allows for covariate effects on the endemic or epidemic contributions to disease incidence. Covariates may vary over both regions and time and thus obey the same $T \times I$ matrix structure as the observed counts. For infectious disease models, the regional vaccination coverage is an important example of such a covariate, since it reflects the (remaining) susceptible population. In a thorough analysis of measles occurrence in the German federal states, Herzog *et al.* (2011) found vaccination coverage to be associated with outbreak size. We follow their approach of using the district-specific proportion $1 - v_i$ of unvaccinated children just starting school as a proxy for the susceptible population. As v_i we use the proportion of children vaccinated with at least one dose among the ones presenting their vaccination card at school entry in district i in the year 2004.[1] This time-constant covariate needs to be transformed to the common matrix structure for incorporation in `hhh4`:

```
R> Sprop <- matrix(1 - measlesWeserEms@map@data$vacc1.2004,
+    nrow = nrow(measlesWeserEms), ncol = ncol(measlesWeserEms), byrow = TRUE)
R> summary(Sprop[1, ])
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0306  0.0481  0.0581  0.0675  0.0830  0.1398
```

There are several ways to account for the susceptible proportion in our model, among which the simplest is to update the endemic population offset e_i by multiplication with $(1 - v_i)$.

---

[1]First year with data for all districts; more recent data is available from the public health department of Lower Saxony (https://www.nlga.niedersachsen.de/impfreport/).

Herzog *et al.* (2011) found that the susceptible proportion is best added as a covariate in the autoregressive component in the form

$$\lambda_i\, Y_{i,t-1} = \exp\left(\alpha^{(\lambda)} + \beta_s \log(1 - v_i)\right) Y_{i,t-1} = \exp\left(\alpha^{(\lambda)}\right)(1 - v_i)^{\beta_s}\, Y_{i,t-1}$$

according to the mass action principle (Keeling and Rohani 2008). A higher proportion of susceptibles in district i is expected to boost the generation of new infections, i.e., $\beta_s > 0$. Alternatively, this effect could be assumed as an offset, i.e., $\beta_s \equiv 1$. To choose between endemic and/or autoregressive effects, and multiplicative offset vs. covariate modeling, we perform AIC-based model selection. First, we set up a grid of possible component updates:

```
R> Soptions <- c("unchanged", "Soffset", "Scovar")
R> SmodelGrid <- expand.grid(end = Soptions, ar = Soptions)
R> row.names(SmodelGrid) <- do.call("paste", c(SmodelGrid, list(sep = "|")))
```

Then we update the initial model `measlesFit_basic` according to each row of `SmodelGrid`:

```
R> measlesFits_vacc <- apply(X = SmodelGrid, MARGIN = 1, FUN = function (options) {
+    updatecomp <- function (comp, option) switch(option, "unchanged" = list(),
+      "Soffset" = list(offset = comp$offset * Sprop),
+      "Scovar" = list(f = update(comp$f, ~. + log(Sprop))))
+    update(measlesFit_basic,
+      end = updatecomp(measlesFit_basic$control$end, options[1]),
+      ar = updatecomp(measlesFit_basic$control$ar, options[2]),
+      data = list(Sprop = Sprop))
+    })
```

The resulting object `measlesFits_vacc` is a list of 9 `hhh4` fits, which are named according to the corresponding `Soptions` used for the endemic and autoregressive components. We construct a call of the function `AIC` taking all list elements as arguments:

```
R> aics_vacc <- do.call(AIC, lapply(names(measlesFits_vacc), as.name),
+    envir = as.environment(measlesFits_vacc))

R> aics_vacc[order(aics_vacc[, "AIC"]), ]
```

```
                      df  AIC
`Scovar|unchanged`     8 1917
`Scovar|Scovar`        9 1919
`Soffset|unchanged`    7 1922
`Soffset|Scovar`       8 1924
`Scovar|Soffset`       8 1934
`Soffset|Soffset`      7 1937
unchanged|unchanged    7 1957
`unchanged|Scovar`     8 1959
`unchanged|Soffset`    7 1967
```

Hence, AIC increases if the susceptible proportion is only added to the autoregressive component, but we see a remarkable improvement when adding it to the endemic component. The best model is obtained by leaving the autoregressive component unchanged (λ) and adding the term $\beta_s \log(1 - v_i)$ to the endemic predictor in Equation 6.

```
R> measlesFit_vacc <- update(measlesFit_basic,
+    end = list(f = update(formula(measlesFit_basic)$end, ~. + log(Sprop))),
+    data = list(Sprop = Sprop))
R> coef(measlesFit_vacc, se = TRUE)["end.log(Sprop)", ]
```

```
  Estimate Std. Error
    1.7181     0.2877
```

The estimated exponent $\hat{\beta}_s$ is both clearly positive and different from the offset assumption. In other words, if a district's fraction of susceptibles is doubled, the endemic measles incidence is estimated to multiply by $2^{\hat{\beta}_s}$:

```
R> 2^cbind("Estimate" = coef(measlesFit_vacc),
+    confint(measlesFit_vacc))["end.log(Sprop)",]
```

```
Estimate    2.5 %   97.5 %
   3.290    2.226    4.864
```

3.3. Spatial interaction

Up to now, the model assumed that the epidemic can only arrive from directly adjacent districts ($w_{ji} = \mathbb{I}(j \sim i)$), and that all districts have the same ability ϕ to import cases from neighboring regions. Given that humans travel further and preferably to metropolitan areas, both assumptions seem overly simplistic and should be tuned toward a "gravity" model for human interaction. First, to reflect commuter-driven spread in our model, we scale the district's susceptibility with respect to its population fraction by multiplying ϕ with $e_i^{\beta_{pop}}$:

```
R> measlesFit_nepop <- update(measlesFit_vacc,
+    ne = list(f = ~log(pop)), data = list(pop = population(measlesWeserEms)))
```

As in a similar analyses of influenza (Geilhufe *et al.* 2014; Meyer and Held 2014), we find strong evidence for such an agglomeration effect: AIC decreases from 1917 to 1887 and the estimated exponent $\hat{\beta}_{pop}$ is

```
R> cbind("Estimate" = coef(measlesFit_nepop),
+    confint(measlesFit_nepop))["ne.log(pop)",]
```

```
Estimate    2.5 %   97.5 %
   2.852    1.831    3.873
```

Second, to account for long-range transmission of cases, Meyer and Held (2014) proposed to estimate the weights w_{ji} as a function of the adjacency order o_{ji} between the districts. For instance, a power-law model assumes the form $w_{ji} = o_{ji}^{-d}$, for $j \neq i$ and $w_{jj} = 0$, where the decay parameter d is to be estimated. Normalization to $w_{ji}/\sum_k w_{jk}$ is recommended and applied by default when choosing `W_powerlaw` as weights in the neighborhood component:

```
R> measlesFit_powerlaw <- update(measlesFit_nepop,
+    ne = list(weights = W_powerlaw(maxlag = 5)))
```

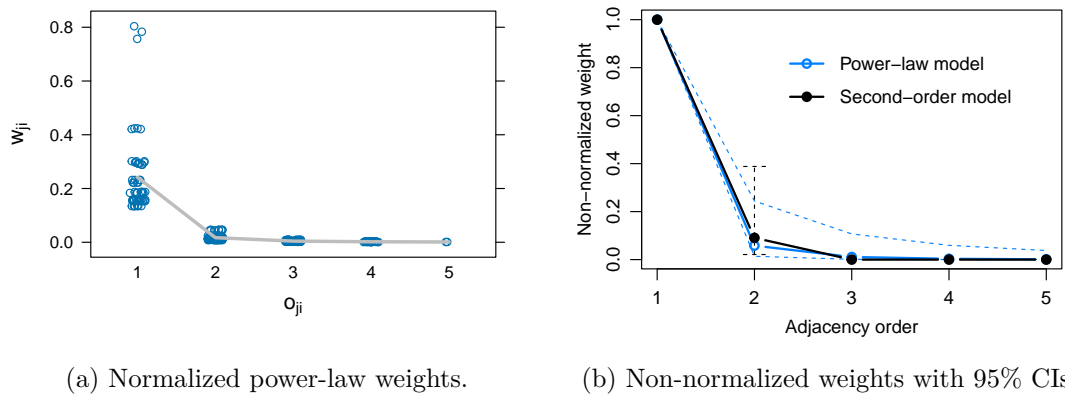(a) Normalized power-law weights.   (b) Non-normalized weights with 95% CIs.

Figure 5: Estimated weights as a function of adjacency order.

The argument `maxlag` sets an upper bound for spatial interaction in terms of adjacency order. Here we set no limit since `max(neighbourhood(measlesWeserEms))` is 5. The decay parameter d is estimated to be

```
R> cbind("Estimate" = coef(measlesFit_powerlaw),
+    confint(measlesFit_powerlaw))["neweights.d",]


Estimate    2.5 %    97.5 %
   4.102    2.034     6.170
```

which represents a strong decay of spatial interaction for higher-order neighbors. As an alternative to the parametric power law, unconstrained weights up to `maxlag` can be estimated by using `W_np` instead of `W_powerlaw`. For instance, `W_np(maxlag = 2)` corresponds to a second-order model, i.e., $w_{ji} = 1 \cdot \mathbb{I}(o_{ji} = 1) + e^{\omega_2} \cdot \mathbb{I}(o_{ji} = 2)$, which is also row-normalized by default:

```
R> measlesFit_np2 <- update(measlesFit_nepop,
+    ne = list(weights = W_np(maxlag = 2)))
```

Figure 5b shows both the power-law model $o^{-\hat{d}}$ and the second-order model. Alternatively, the plot `type = "neweights"` for `hhh4` fits can produce a `stripplot` (Sarkar 2023) of w_{ji} against o_{ji} as shown in Figure 5a for the power-law model:

```
R> library("lattice")
R> plot(measlesFit_powerlaw, type = "neweights", plotter = stripplot,
+    panel = function (...) {panel.stripplot(...); panel.average(...)},
+    jitter.data = TRUE, xlab = expression(o[ji]), ylab = expression(w[ji]))
```

Note that only horizontal jitter is added in this case. Because of normalization, the weight w_{ji} for transmission from district j to district i is determined not only by the districts' neighborhood o_{ji} but also by the total amount of neighborhood of district j in the form of $\sum_{k \neq j} o_{jk}^{-d}$, which causes some variation of the weights for a specific order of adjacency. The function `getNEweights` can be used to extract the estimated weight matrix (w_{ji}).

An AIC comparison of the different models for the transmission weights yields:

```
R> AIC(measlesFit_nepop, measlesFit_powerlaw, measlesFit_np2)
```

```
                      df  AIC
measlesFit_nepop       9 1887
measlesFit_powerlaw   10 1882
measlesFit_np2        10 1881
```

AIC improves when accounting for transmission from higher-order neighbors by a power law or a second-order model. In spite of the latter resulting in a slightly better fit, we will use the power-law model as a basis for further model extensions since the stand-alone second-order effect is not always identifiable in more complex models and is scientifically implausible.

3.4. Random effects

Paul and Held (2011) introduced random effects for `hhh4` models, which are useful if the districts exhibit heterogeneous incidence levels not explained by observed covariates, and especially if the number of districts is large. For infectious disease surveillance data, a typical example of unobserved heterogeneity is underreporting. Our measles data even contain two districts without any reported cases, while the district with the smallest population (03402, SK Emden) had the second-largest number of cases reported and the highest overall incidence (see Figures 1b and 2). Hence, allowing for district-specific intercepts in the endemic or epidemic components is expected to improve the model fit. For independent random effects $\alpha_i^{(\nu)} \overset{iid}{\sim} N(\alpha^{(\nu)}, \sigma_\nu^2)$, $\alpha_i^{(\lambda)} \overset{iid}{\sim} N(\alpha^{(\lambda)}, \sigma_\lambda^2)$, and $\alpha_i^{(\phi)} \overset{iid}{\sim} N(\alpha^{(\phi)}, \sigma_\phi^2)$ in all three components, we update the corresponding formulae as follows:

```
R> measlesFit_ri <- update(measlesFit_powerlaw,
+    end = list(f = update(formula(measlesFit_powerlaw)$end, ~. + ri() - 1)),
+    ar  = list(f = update(formula(measlesFit_powerlaw)$ar,  ~. + ri() - 1)),
+    ne  = list(f = update(formula(measlesFit_powerlaw)$ne,  ~. + ri() - 1)))
```

```
R> summary(measlesFit_ri, amplitudeShift = TRUE, maxEV = TRUE)
```

```
Call:
hhh4(stsObj = object$stsObj, control = control)

Random effects:
            Var  Corr
ar.ri(iid)  1.08
ne.ri(iid)  1.29 0
end.ri(iid) 1.31 0     0

Fixed effects:
                    Estimate  Std. Error
ar.ri(iid)          -1.61389  0.38197
ne.log(pop)          3.42406  1.07722
ne.ri(iid)           6.62429  2.81553
end.t                0.00578  0.00480
end.A(2 * pi * t/52) 1.20359  0.20149
end.s(2 * pi * t/52) -0.47916 0.14205
end.log(Sprop)       1.79350  0.69159
```

```
end.ri(iid)            4.42260    1.94605
neweights.d            3.60640    0.77602
overdisp               0.97723    0.15132


Epidemic dominant eigenvalue:  0.84

Penalized log-likelihood:  -868.6
Marginal log-likelihood:   -54.2


Number of units:          17
Number of time points:  103
```

The summary now contains an extra section with the estimated variance components σ_λ^2, σ_ϕ^2, and σ_ν^2. We did not assume correlation between the three random effects, but this is possible by specifying `ri(corr = "all")` in the component formulae. The implementation also supports a conditional autoregressive formulation for spatially correlated intercepts via `ri(type = "car")`.

The estimated district-specific deviations $\alpha_i^{(\cdot)} - \alpha^{(\cdot)}$ can be extracted by the `ranef`-method:

```
R> head(ranef(measlesFit_ri, tomatrix = TRUE), n = 3)


      ar.ri(iid) ne.ri(iid) end.ri(iid)
03401     0.0000   -0.05673     -1.0045
03402     1.2235    0.04312      1.5264
03403    -0.8273    1.55878     -0.6199
```
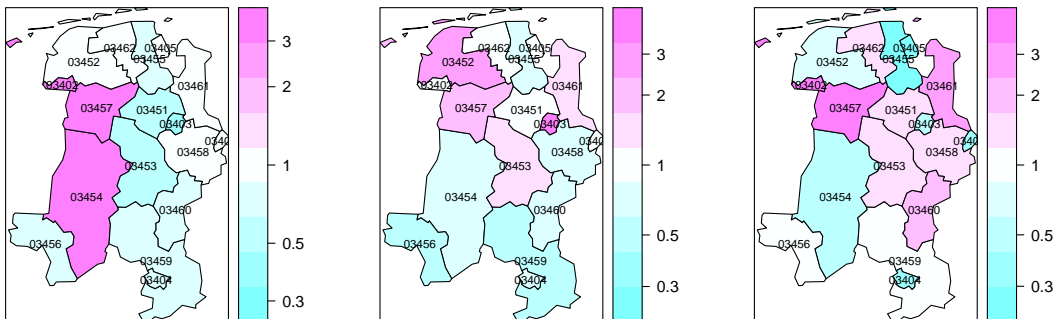
The `exp`-transformed deviations correspond to district-specific multiplicative effects on the model components, which can be visualized via the plot `type = "ri"` as follows (Figure 6):

```
R> for (comp in c("ar", "ne", "end")) {
+   print(plot(measlesFit_ri, type = "ri", component = comp, exp = TRUE,
+     labels = list(cex = 0.6)))
+ }
```


| (a) Autoregressive | (b) Spatio-temporal | (c) Endemic |

Figure 6: Estimated multiplicative effects on the three components.

For the autoregressive component in Figure 6a, we see a pronounced heterogeneity between the three western districts in pink and the remaining districts. These three districts have been affected by large local outbreaks and are also the ones with the highest overall numbers of cases. In contrast, the city of Oldenburg (03403) is estimated with a relatively low autoregressive coefficient: $\lambda_i = \exp(\alpha_i^{(\lambda)})$ can be extracted using the `intercept` argument as

```
R> exp(ranef(measlesFit_ri, intercept = TRUE)["03403", "ar.ri(iid)"])
```

```
[1] 0.08706
```

However, this district seems to import more cases from other districts than explained by its population (Figure 6b). In Figure 6c, the two districts without any reported measles cases (03401 and 03405) appear in cyan, which means that they exhibit a relatively low endemic incidence after adjusting for the population and susceptible proportion. Such districts could be suspected of a larger amount of underreporting.

We plot the new model fit (Figure 7) for comparison with the initial fit shown in Figure 4:

```
R> par(mfrow = c(2,3), mar = c(3, 5, 2, 1), las = 1)
R> plot(measlesFit_ri, type = "fitted", units = districts2plot,
+       hide0s = TRUE, par.settings = NULL, legend = 1)
R> plot(measlesFit_ri, type = "fitted", total = TRUE,
+       hide0s = TRUE, par.settings = NULL, legend = FALSE)
```
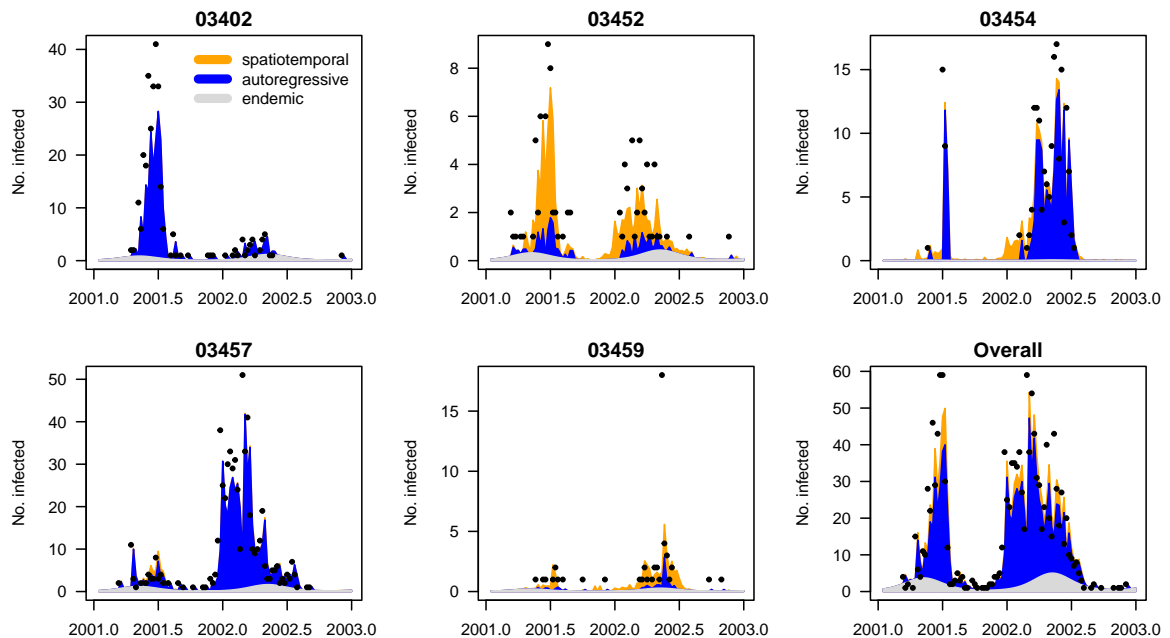

Figure 7: Fitted components in the random effects model `measlesFit_ri` for the five districts with more than 50 cases as well as summed over all districts. Compare to Figure 4.

For some of these districts, a great amount of cases is now explained via transmission from neighboring regions while others are mainly influenced by the local autoregression.

The decomposition of the estimated mean by district can also be seen from the related plot
`type = "maps"` (Figure 8):

```
R> plot(measlesFit_ri, type = "maps",
+    which = c("epi.own", "epi.neighbours", "endemic"),
+    prop = TRUE, labels = list(cex = 0.6))
```
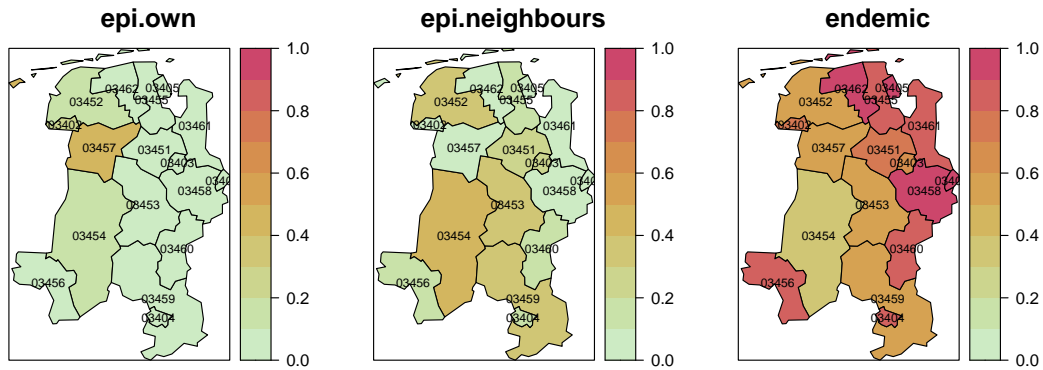

Figure 8: Maps of the fitted component proportions averaged over all weeks.

The extra flexibility of the random effects model comes at a price. First, the runtime of
the estimation increases considerably from 0.1 seconds for the previous power-law model
`measlesFit_powerlaw` to 1.3 seconds with random effects. Furthermore, we no longer obtain
AIC values, since random effects invalidate simple AIC-based model comparisons. For quan-
titative comparisons of model performance we have to resort to more sophisticated techniques
presented in the next section.

3.5. Predictive model assessment

Paul and Held (2011) suggest to evaluate one-step-ahead forecasts from competing models
using proper scoring rules for count data (Czado, Gneiting, and Held 2009). These scores
measure the discrepancy between the predictive distribution P from a fitted model and the
later observed value y. A well-known example is the squared error score ("ses") $(y - \mu_P)^2$,
which is usually averaged over a set of forecasts to obtain the mean squared error. The Dawid-
Sebastiani score ("dss") additionally evaluates sharpness. The logarithmic score ("logs") and
the ranked probability score ("rps") assess the whole predictive distribution with respect to
calibration and sharpness. Lower scores correspond to better predictions.

In the `hhh4` framework, predictive model assessment is made available by the functions
`oneStepAhead`, `scores`, `pit`, and `calibrationTest`. We will use the second quarter of 2002
as the test period, and compare the basic model, the power-law model, and the random effects
model. First, we use the `"final"` fits on the complete time series to compute the predictions,
which then simply correspond to the fitted values during the test period:

```
R> tp <- c(65, 77)
R> models2compare <- paste0("measlesFit_", c("basic", "powerlaw", "ri"))
R> measlesPreds1 <- lapply(mget(models2compare), oneStepAhead,
+    tp = tp, type = "final")
```

Note that in this case, the log-score for a model's prediction in district i in week t equals the associated negative log-likelihood contribution. Comparing the mean scores from different models is thus essentially a goodness-of-fit assessment:

```
R> SCORES <- c("logs", "rps", "dss", "ses")
R> measlesScores1 <- lapply(measlesPreds1, scores, which = SCORES, individual = TRUE)
R> t(sapply(measlesScores1, colMeans, dims = 2))
```

```
                     logs    rps    dss   ses
measlesFit_basic    1.089 0.7358 1.2911 5.289
measlesFit_powerlaw 1.101 0.7307 2.2223 5.394
measlesFit_ri       1.007 0.6381 0.9656 4.823
```

All scoring rules claim that the random effects model gives the best fit during the second quarter of 2002. Now we turn to true one-week-ahead predictions of `type = "rolling"`, which means that we always refit the model up to week t to get predictions for week $t + 1$:

```
R> measlesPreds2 <- lapply(mget(models2compare), oneStepAhead,
+    tp = tp, type = "rolling", which.start = "final")
```

Figure 9 shows **fanplot**s (Abel 2021) of the sequential one-week-ahead forecasts from the random effects models for the same districts as in Figure 7:

```
R> for (unit in names(districts2plot))
+    plot(measlesPreds2[["measlesFit_ri"]], unit = unit, main = unit,
+      key.args = if (unit == tail(names(districts2plot),1)) list())
```

The `plot`-method for `oneStepAhead` predictions is based on the associated `quantile`-method (a `confint`-method is also available). Note that the sum of these negative binomial distributed forecasts over all districts is not negative binomial distributed. The package **distr** (Ruckdeschel and Kohl 2014) could be used to approximate the distribution of the aggregated one-step-ahead forecasts (not shown here).

Looking at the average scores of these forecasts over all weeks and districts, the most parsimonious initial model `measlesFit_basic` actually turns out best:

```
R> measlesScores2 <- lapply(measlesPreds2, scores, which = SCORES, individual = TRUE)
R> t(sapply(measlesScores2, colMeans, dims = 2))
```

```
                     logs    rps   dss   ses
measlesFit_basic    1.102 0.7478 1.339 5.404
measlesFit_powerlaw 1.136 0.7654 2.929 5.865
measlesFit_ri       1.110 0.7632 2.349 7.080
```

Statistical significance of the differences in mean scores can be investigated by a `permutationTest` for paired data or a paired t-test:

```
R> set.seed(321)
R> sapply(SCORES, function (score) permutationTest(
+    measlesScores2$measlesFit_ri[, , score],
+    measlesScores2$measlesFit_basic[, , score],
+    nPermutation = 999))
```
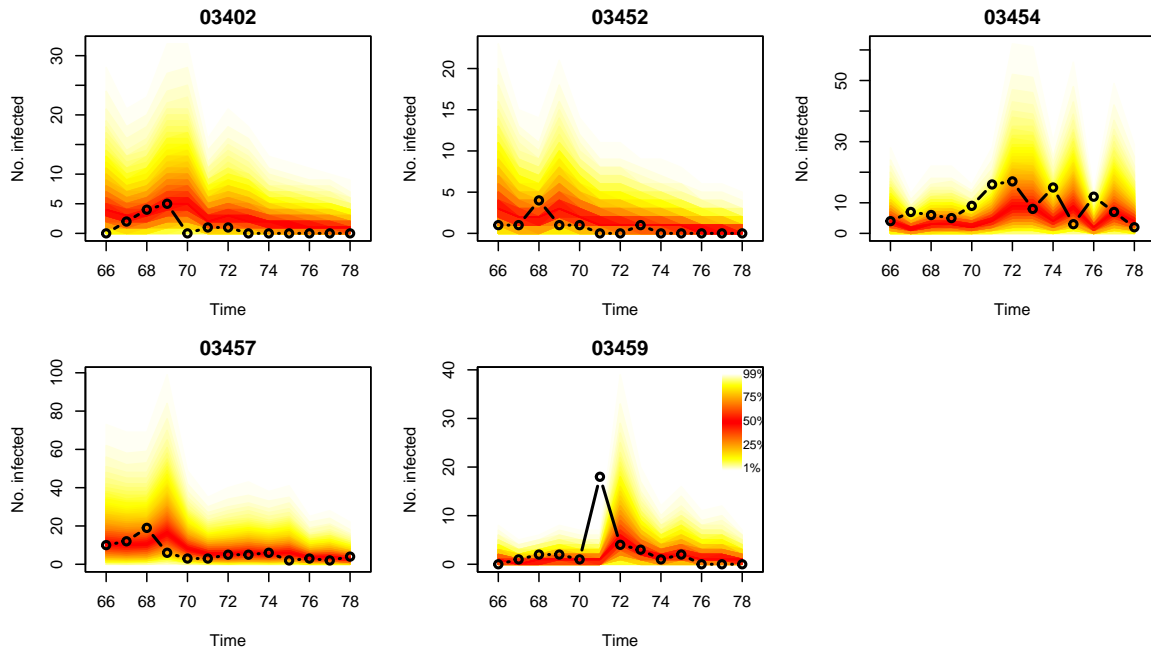
Figure 9: Fan charts of rolling one-week-ahead forecasts during the second quarter of 2002, as produced by the random effects model `measlesFit_ri`, for the five most affected districts.

```
               logs     rps     dss    ses
diffObs        0.007822 0.01541 1.01   1.677
pVal.permut    0.86     0.734   0.51   0.202
pVal.t         0.8541   0.7165  0.3737 0.1711
```

Hence, there is no clear evidence for a difference between the basic and the random effects model with regard to predictive performance during the test period.

Whether predictions of a particular model are well calibrated can be formally investigated by `calibrationTest`s for count data as recently proposed by Wei and Held (2014). For example:

```
R> calibrationTest(measlesPreds2[["measlesFit_ri"]], which = "rps")


        Calibration Test for Count Data (based on RPS)

data:  measlesPreds2[["measlesFit_ri"]]
z = 0.80671, n = 221, p-value = 0.4198
```

Thus, there is no evidence of miscalibrated predictions from the random effects model. Czado *et al.* (2009) describe an alternative informal approach to assess calibration: probability integral transform (PIT) histograms for count data (Figure 10).

```
R> for (m in models2compare)
+     pit(measlesPreds2[[m]], plot = list(ylim = c(0, 1.25), main = m))
```

Under the hypothesis of calibration, i.e., $y_{it} \sim P_{it}$ for all predictive distributions P_{it} in the test period, the PIT histogram is uniform. Underdispersed predictions lead to U-shaped
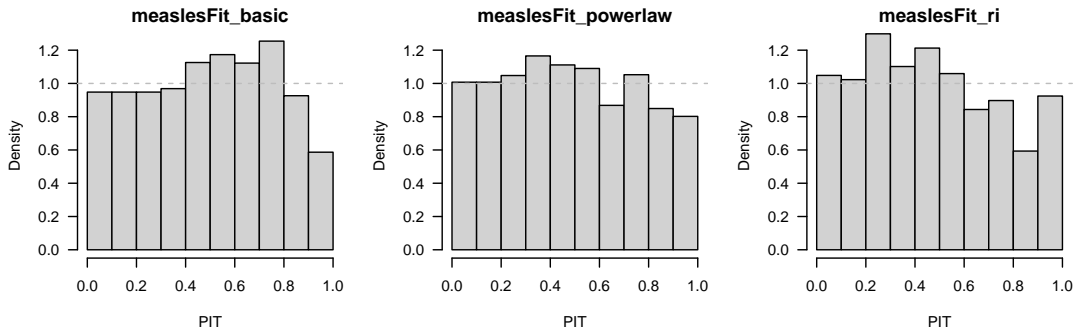
Figure 10: PIT histograms of competing models to check calibration of the one-week-ahead predictions during the second quarter of 2002.

histograms, and bias causes skewness. In this aggregate view of the predictions over all districts and weeks of the test period, predictive performance is comparable between the models, and there is no evidence of badly dispersed predictions. However, the right-hand decay in all histograms suggests that all models tend to predict higher counts than observed. This is most likely related to the seasonal shift between the years 2001 and 2002. In 2001, the peak of the epidemic was in the second quarter, while it already occurred in the first quarter in 2002 (cp. Figure 1a).

3.6. Further modeling options

In the previous sections we extended our model for measles in the Weser-Ems region with respect to spatial variation of the counts and their interaction. Temporal variation was only accounted for in the endemic component, which included a long-term trend and a sinusoidal wave on the log-scale. Held and Paul (2012) suggest to also allow seasonal variation of the epidemic force by adding a superposition of S harmonic waves of fundamental frequency ω, $\sum_{s=1}^{S} \left\{ \gamma_s \sin(s\,\omega t) + \delta_s \cos(s\,\omega t) \right\}$, to the log-linear predictors of the autoregressive and/or neighborhood component – just like for $\log \nu_t$ in Equation 6 with $S = 1$. However, given only two years of measles surveillance and the apparent shift of seasonality with regard to the start of the outbreak in 2002 compared to 2001, more complex seasonal models are likely to overfit the data. Concerning the coding in R, sine-cosine terms can be added to the epidemic components without difficulties by again using the convenient function `addSeason2formula`. Updating a previous model for different numbers of harmonics is even simpler, since the `update`-method has a corresponding argument S. The plots of `type = "season"` and `type = "maxEV"` for `hhh4` fits can visualize the estimated component seasonality.

Performing model selection and interpreting seasonality or other covariate effects across *three* different model components may become quite complicated. Power-law weights actually enable a more parsimonious model formulation, where the autoregressive and neighbourhood components are merged into a single epidemic component:

$$\mu_{it} = e_{it}\,\nu_{it} + \phi_{it} \sum_j (o_{ji} + 1)^{-d} Y_{j,t-1} \,. \tag{7}$$

With only two predictors left, model selection and interpretation is simpler, and model extensions are more straightforward, for example stratification by age group (Meyer and Held

2017) as mentioned further below. To fit such a two-component model, the autoregressive component has to be excluded (`ar = list(f = ~ -1)`) and power-law weights have to be modified to start from adjacency order 0 (via `W_powerlaw(..., from0 = TRUE)`).

All of our models for the measles surveillance data incorporated an epidemic effect of the counts from the local district and its neighbors. Without further notice, we thereby assumed a lag equal to the observation interval of one week. However, the generation time of measles is around 10 days, which is why Herzog *et al.* (2011) aggregated their weekly measles surveillance data into biweekly intervals. We can perform a sensitivity analysis by running the whole code of the current section based on `aggregate(measlesWeserEms, nfreq = 26)`. Doing so, the parameter estimates of the various models retain their order of magnitude and conclusions remain the same. However, with the number of time points halved, the complex random effects model would not always be identifiable when calculating one-week-ahead predictions during the test period.

We have shown several options to account for the spatio-temporal dynamics of infectious disease spread. However, for directly transmitted human diseases, the social phenomenon of "like seeks like" results in contact patterns between subgroups of a population, which extend the pure distance decay of interaction. Especially for school children, social contacts are highly age-dependent. A useful epidemic model should therefore be additionally stratified by age group and take the inherent contact structure into account. How this extension can be incorporated in the spatio-temporal endemic-epidemic modeling framework `hhh4` has recently been investigated by Meyer and Held (2017). The associated **hhh4contacts** package (Meyer 2024) contains a demo script to exemplify this modeling approach with surveillance data on norovirus gastroenteritis and an age-structured contact matrix.

4. Simulation

Simulation from fitted `hhh4` models is enabled by an associated `simulate`-method. Compared to the point process models described in `vignette("twinstim")` and `vignette("twinSIR")`, simulation is less complex since it essentially consists of sequential calls of `rnbinom` (or `rpois`). At each time point t, the mean μ_{it} is determined by plugging in the parameter estimates and the counts $Y_{i,t-1}$ simulated at the previous time point. In addition to a model fit, we thus need to specify an initial vector of counts `y.start`. As an example, we simulate 100 realizations of the evolution of measles during the year 2002 based on the fitted random effects model and the counts of the last week of the year 2001 in the 17 districts:

```
R> (y.start <- observed(measlesWeserEms)[52, ])

03401 03402 03403 03404 03405 03451 03452 03453 03454 03455 03456 03457 03458 03459
    0     0     0     0     0     0     0     0     0     0     0    25     0     0
03460 03461 03462
    0     0     0

R> measlesSim <- simulate(measlesFit_ri,
+     nsim = 100, seed = 1, subset = 53:104, y.start = y.start)
```

The simulated counts are returned as a $52 \times 17 \times 100$ array instead of a list of 100 `sts` objects. We can, e.g., look at the final size distribution of the simulations:

```
R> summary(colSums(measlesSim, dims = 2))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    223     326     424     550     582    3971
```

A few large outbreaks have been simulated, but the mean size is below the observed number of `sum(observed(measlesWeserEms)[53:104, ]) = 779` cases in the year 2002. Using the `plot`-method associated with such `hhh4` simulations, Figure 11 shows the weekly number of observed cases compared to the long-term forecast via a fan chart:

```
R> plot(measlesSim, "fan", means.args = list(), key.args = list())
```
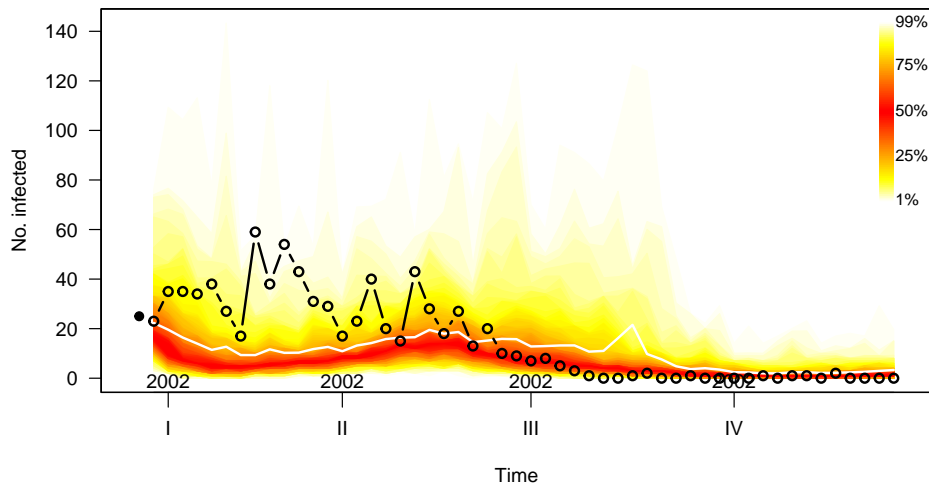

Figure 11: Simulation-based long-term forecast starting from the last week in 2001 (left-hand dot). The plot shows the weekly counts aggregated over all districts. The fan chart represents the 1% to 99% quantiles of the simulations in each week; their mean is displayed as a white line. The circles correspond to the observed counts.

We refer to `help("simulate.hhh4")` and `help("plot.hhh4sims")` for further examples.

References

Abel GJ (2021). *fanplot: Visualisation of Sequential Probability Distributions Using Fan Charts*. R package version 4.0.0, URL http://guyabel.github.io/fanplot/.

Auguie B (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. doi:10.32614/CRAN.package.gridExtra. R package version 2.3, URL https://CRAN.R-project.org/package=gridExtra.

Bates D, Maechler M, Jagan M (2025). *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.7-3, URL https://Matrix.R-forge.R-project.org.

Bivand R (2022). "R Packages for Analyzing Spatial Data: A Comparative Case Study with Areal Data." *Geographical Analysis*, **54**(3), 488–518. `doi:10.1111/gean.12319`.

Czado C, Gneiting T, Held L (2009). "Predictive model assessment for count data." *Biometrics*, **65**(4), 1254–1261. `doi:10.1111/j.1541-0420.2009.01191.x`.

Geilhufe M, Held L, Skrøvseth SO, Simonsen GS, Godtliebsen F (2014). "Power law approximations of movement network data for modeling infectious disease spread." *Biometrical Journal*, **56**(3), 363–382. `doi:10.1002/bimj.201200262`.

Held L, Höhle M, Hofmann M (2005). "A statistical framework for the analysis of multivariate infectious disease surveillance counts." *Statistical Modelling*, **5**(3), 187–199. `doi:10.1191/1471082X05st098oa`.

Held L, Paul M (2012). "Modeling seasonality in space-time infectious disease surveillance data." *Biometrical Journal*, **54**(6), 824–843. `doi:10.1002/bimj.201200037`.

Herzog SA, Paul M, Held L (2011). "Heterogeneity in vaccination coverage explains the size and occurrence of measles epidemics in German surveillance data." *Epidemiology and Infection*, **139**(4), 505–515. `doi:10.1017/S0950268810001664`.

Höhle M, Mazick A (2010). "Aberration detection in R illustrated by Danish mortality monitoring." In T Kass-Hout, X Zhang (eds.), *Biosurveillance: Methods and Case Studies*, chapter 12, pp. 215–238. Chapman & Hall/CRC.

Ihaka R, Murrell P, Hornik K, Fisher JC, Stauffer R, Wilke CO, McWhite CD, Zeileis A (2024). *colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes*. R package version 2.1-1, URL `https://colorspace.R-Forge.R-project.org/`.

Keeling MJ, Rohani P (2008). *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press.

Meyer S (2024). *hhh4contacts: Age-Structured Spatio-Temporal Models for Infectious Disease Counts*. R package version 0.13.4, URL `https://codeberg.org/EE-hub/hhh4contacts`.

Meyer S, Held L (2014). "Power-law models for infectious disease spread." *Annals of Applied Statistics*, **8**(3), 1612–1639. `doi:10.1214/14-AOAS743`.

Meyer S, Held L (2017). "Incorporating social contact data in spatio-temporal models for infectious disease spread." *Biostatistics*, **18**(2), 338–351. `doi:10.1093/biostatistics/kxw051`.

Meyer S, Held L, Höhle M (2017). "Spatio-temporal analysis of epidemic phenomena using the R package `surveillance`." *Journal of Statistical Software*, **77**(11), 1–55. `doi:10.18637/jss.v077.i11`.

Paul M, Held L (2011). "Predictive assessment of a non-linear random effects model for multivariate time series of infectious disease counts." *Statistics in Medicine*, **30**(10), 1118–1136. `doi:10.1002/sim.4177`.

Paul M, Held L, Toschke AM (2008). "Multivariate modelling of infectious disease surveillance data." *Statistics in Medicine*, **27**(29), 6250–6267. `doi:10.1002/sim.3440`.

Pebesma E, Bivand R (2025). *sp: Classes and Methods for Spatial Data.* R package version 2.2-0, URL https://github.com/edzer/sp/.

Ripley B, Venables B (2025). *MASS: Support Functions and Datasets for Venables and Ripley's MASS.* R package version 7.3-65, URL http://www.stats.ox.ac.uk/pub/MASS4/.

Ruckdeschel P, Kohl M (2014). "General purpose convolution algorithm in S4 classes by means of FFT." *Journal of Statistical Software*, **59**(4), 1–25. doi:10.18637/jss.v059.i04.

Salmon M, Schumacher D, Höhle M (2016). "Monitoring count time series in R: Aberration detection in public health surveillance." *Journal of Statistical Software*, **70**(10), 1–35. doi:10.18637/jss.v070.i10.

Sarkar D (2023). *lattice: Trellis Graphics for R.* R package version 0.22-5, URL https://lattice.r-forge.r-project.org/.

Teucher A, Russell K (2020). *rmapshaper: Client for 'mapshaper' for 'Geospatial' Operations.* URL https://CRAN.R-project.org/package=rmapshaper.

Wei W, Held L (2014). "Calibration tests for count data." *Test*, **23**(4), 787–805. doi:10.1007/s11749-014-0380-8.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag, New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Xie Y (2013). "animation: An R Package for Creating Animations and Demonstrating Statistical Methods." *Journal of Statistical Software*, **53**(1), 1–27. doi:10.18637/jss.v053.i01.