



epinowcast: Hierarchical nowcasting of truncated count data

Hospital surge working group

Sam Abbott (@seabbs)¹, Felix Günther², Johannes Bracher³, Adrian Lison⁴,
Ryan Sebastian Funk¹

July 6, 2022

¹ London School of Hygiene Tropical Medicine,

² Department of Mathematical Statistics, Stockholm University

³ Karlsruhe Institute of Technology / Heidelberg Institute for Theoretical Studies

⁴ ETH Zurich

Table of contents

1. Introducing the nowcasting problem
2. The Germany nowcasting hub
3. The epinowcast model
4. The epinowcast R package
5. Extensions
6. Summary

Introducing the nowcasting problem

Introduction - General problem statement

- Infectious disease data is created by an underlying infection process.
- Infections are generally unobserved.
- We observe other related measures such as the onset of symptoms, test positivity, hospital admission, and death.
- All of these measures happen with some delay from the original date of infection and from each other.
- Observation itself generally has some delay.
- This means real-time observations are often truncated.
- This truncation is linked to the underlying growth rate of infections.

Introduction - Aims of nowcasting

Core aim

Estimate what will ultimately be reported for proxies of infection that we observe with truncation.

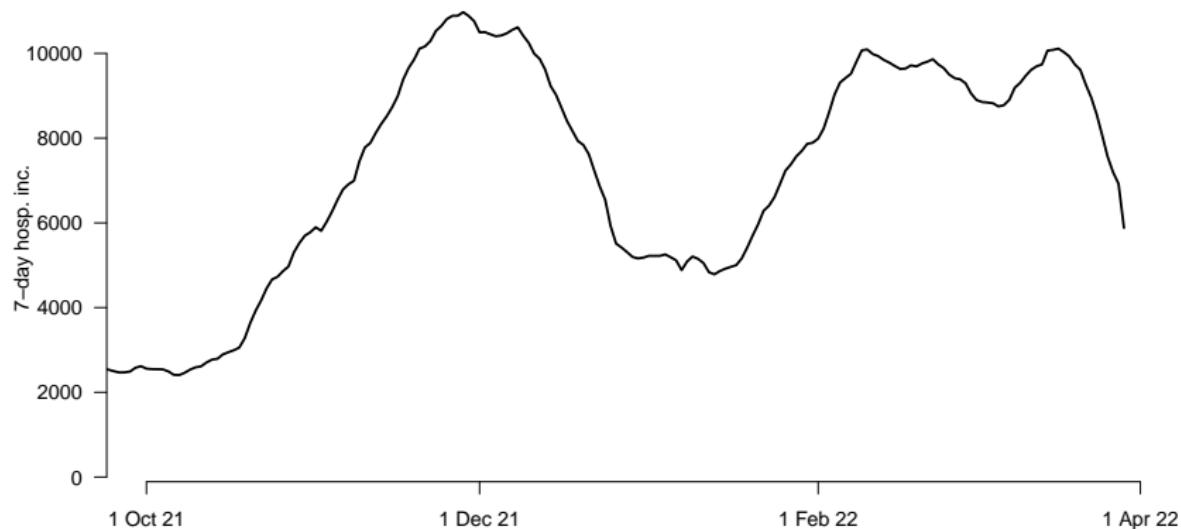
Secondary aims

- Provide improved situational awareness in real-time contexts.
- Estimate the underlying delay distributions for use in other contexts and to improve understanding of the disease system.
- Improve forecasts of the truncated observations.

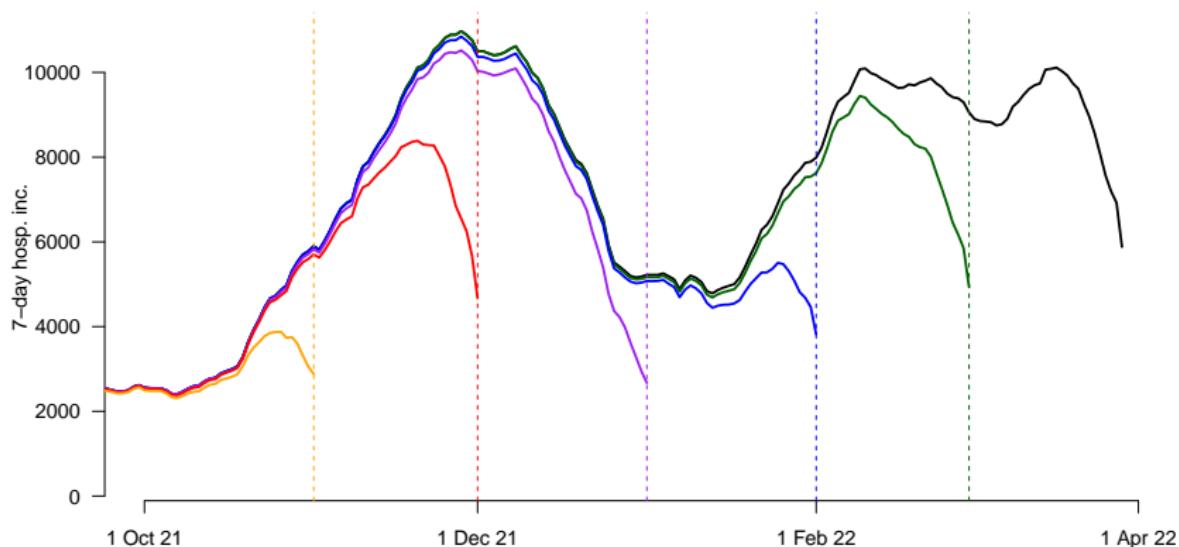
A German example - Seven day hospitalisation incidence

- **Definition:** The number of persons, who over a seven-day period
 - have been registered electronically as a COVID-19 case by a local health authority (*Meldedatum*).
 - and have been hospitalized (not necessarily during the seven-day period).
- Most recent values are biased downwards due to two types of delays:
 - delay between *Meldedatum* (\approx positive test) and hospitalization.
 - delay between hospitalization and appearance in RKI data.

A German example - What does the data look like?



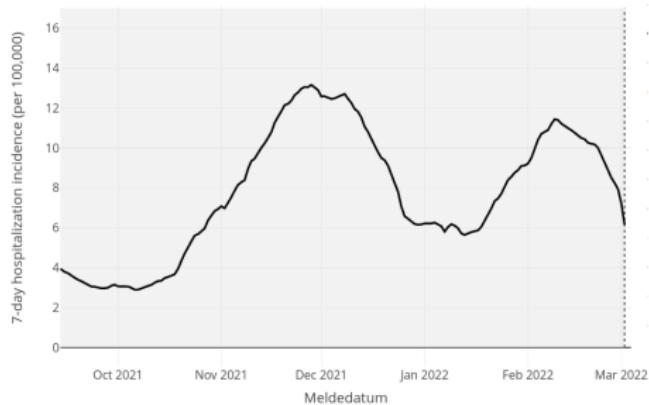
A German example - What does the data look like?



A German example - What are we trying to do?

Goal

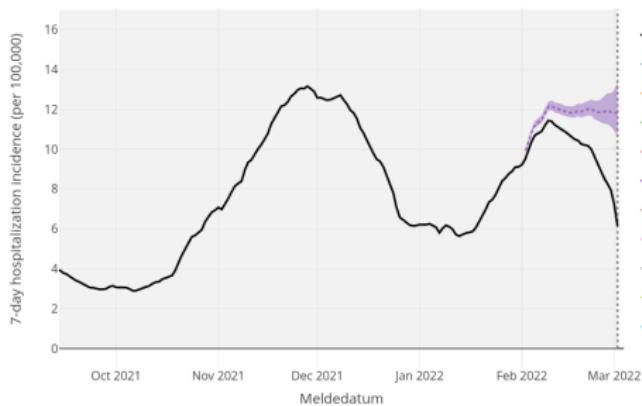
Estimate (predict) what preliminary/incomplete values will ultimately look like.



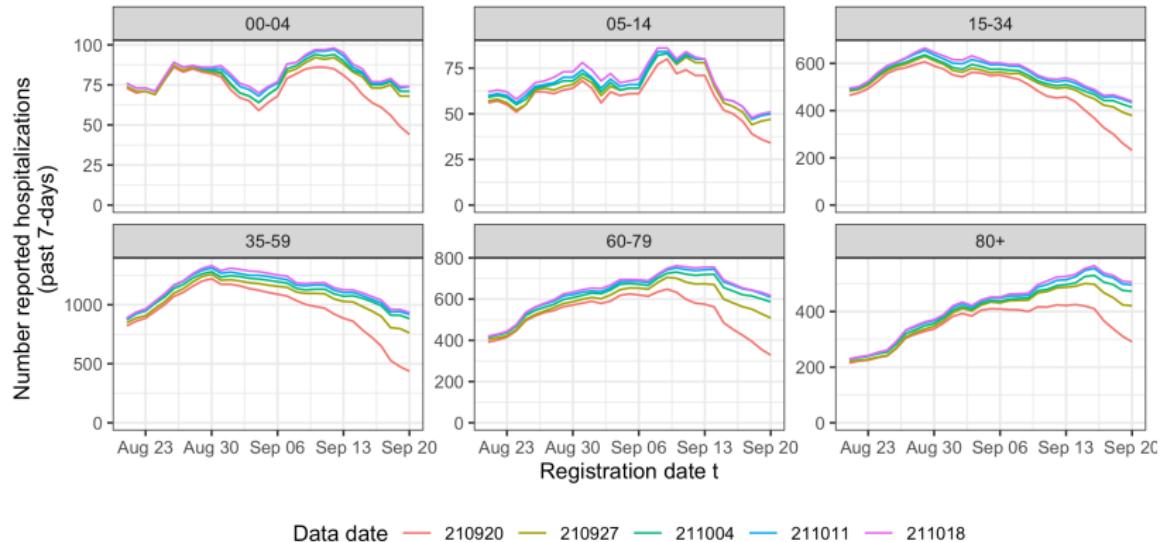
A German example - What are we trying to do?

Goal

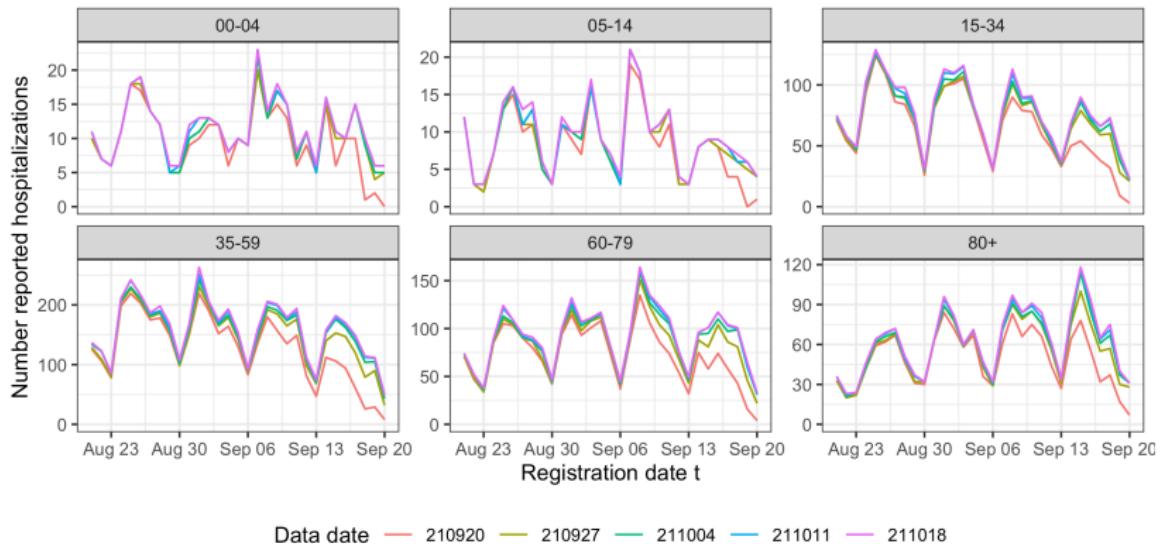
Estimate (predict) what preliminary/incomplete values will ultimately look like.



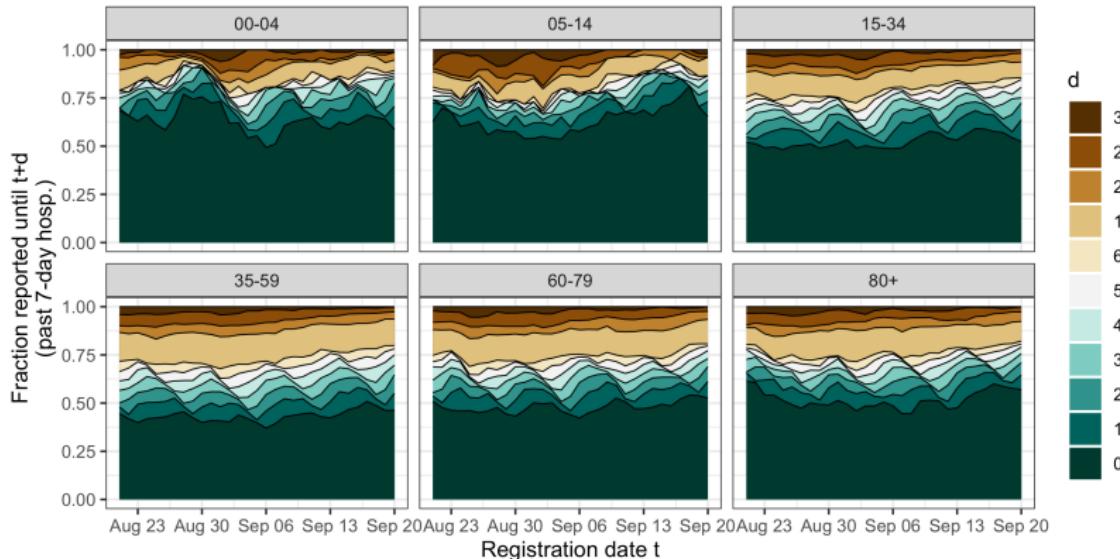
A German example - Age and region stratified



A German example - underlying data has a strong day of week signal



A German example - What about reporting delays?



A German example - Summary

- Seven day hospitalisation incidence by date of positive test is used as a key indicator in Germany.
- These data are truncated and ignoring this may lead to biased surveillance measures and flawed disease making.
- The data is age and location stratified.
- Both incidence and reporting has a strong weekly structure.
- Reporting delays appear to vary over time and by strata.

The nowcasting aim in this context

Estimate hospitalisations for registration days from the number of already reported hospitalisations and the date of these reports.

The Germany nowcasting hub

The Germany nowcasting hub - Multi-model nowcasting

- Experience from weather and infectious disease forecasting (i.e the CDC and ECDC forecasting hubs) shows that combining different models can improve predictions.
- The hub collects and combines probabilistic nowcasts from 8 independently run models.

hospitalization-nowcast-hub / data-processed / KIT-simple_nowcast / 2022-03-29-KIT-simple_nowcast.csv

dwolfram Update Baseline ✓ Latest comm

At 1 contributor

5337 lines (5337 sloc) | 472 KB

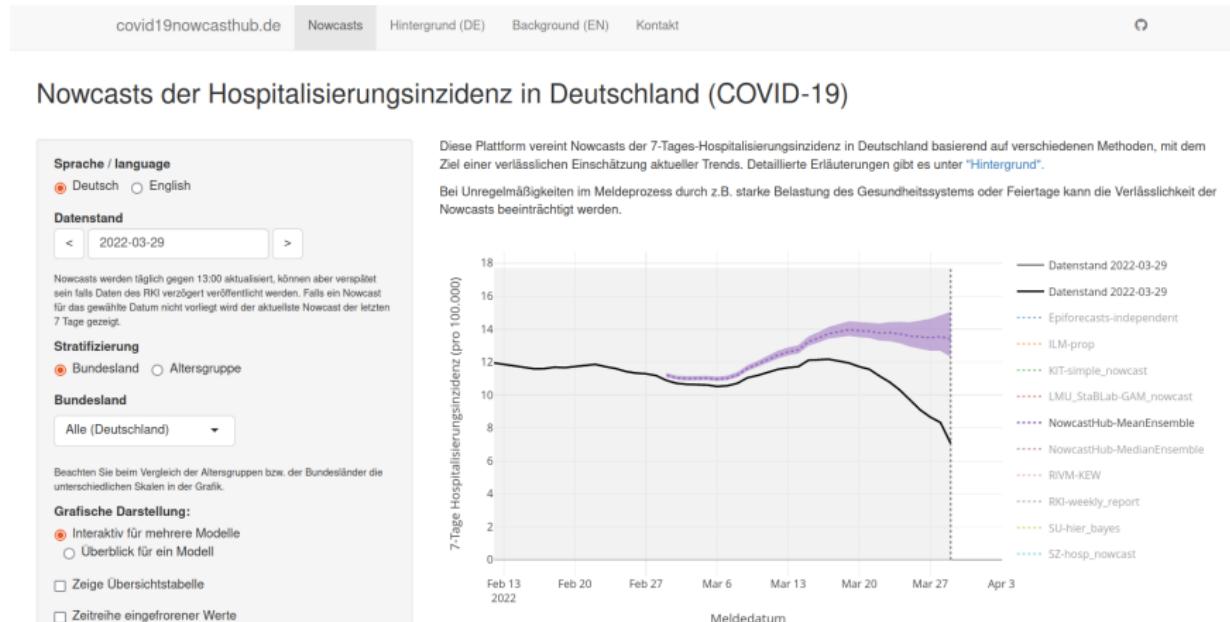
Search this file...

1	location	age_group	forecast_date	target_end_date	target	type	quantile	value	pathogen
2	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	mean	NA	10642	COVID-19
3	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.025	8521	COVID-19
4	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.1	9131	COVID-19
5	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.25	9753	COVID-19
6	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.5	10534	COVID-19
7	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.75	11413	COVID-19
8	DE	00+	2022-03-29	2022-03-29	0 day ahead inc hosp	quantile	0.9	12292	COVID-19

<https://github.com/KITmetricslab/hospitalization-nowcast-hub/tree/main/data-truth/COVID-19>

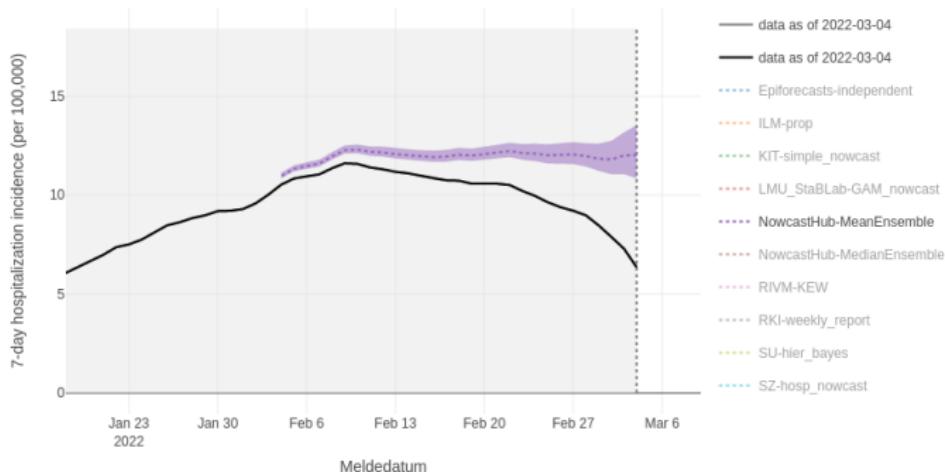
The Germany nowcasting hub - Interactive online platform

<https://covid19nowcasthub.de/>

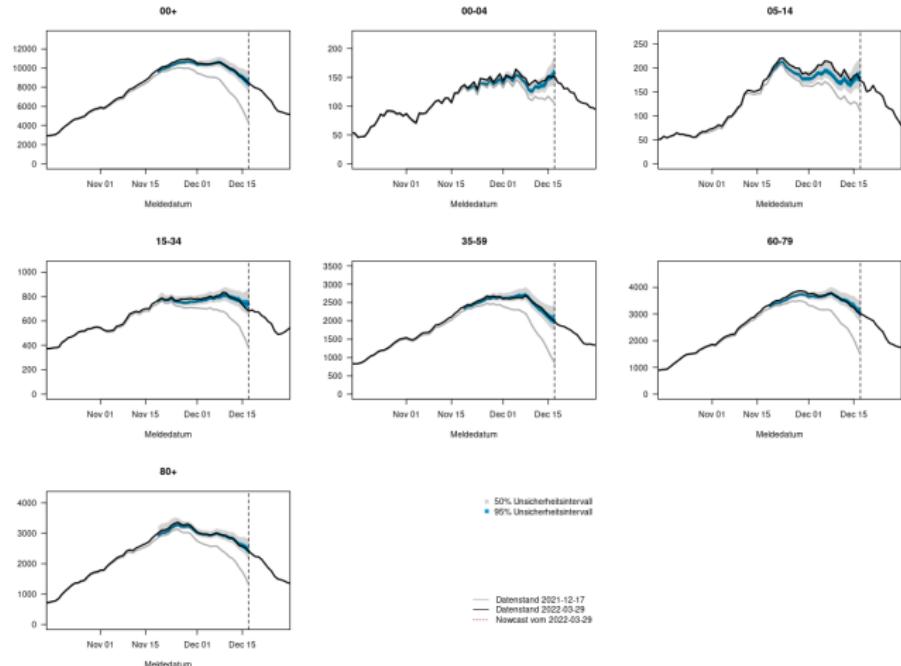


The Germany nowcasting hub - the ensemble

- The main output of the platform is an **ensemble nowcast**, i.e. combination of all available models.

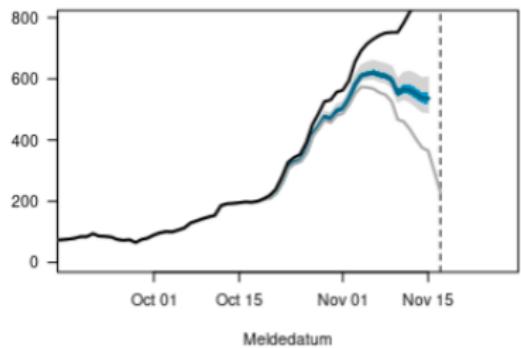


The Germany nowcasting hub - The ensemble is pretty good

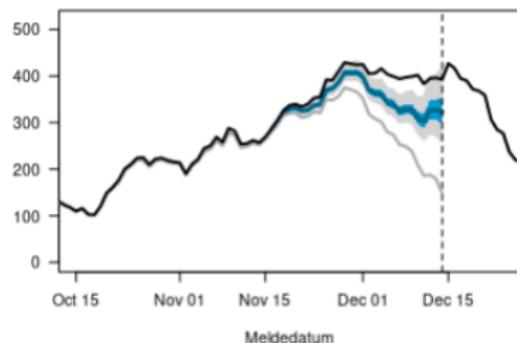


The Germany nowcasting hub - ... except when it isn't.

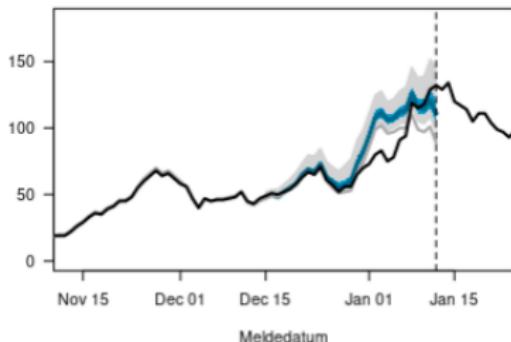
Sachsen



Rheinland-Pfalz



Bremen



The Germany nowcasting hub - Pre-registered evaluation study

- The hub team are conducting a systematic evaluation study of real-time nowcasts from different methods.
- This study has been pre-registered (<https://osf.io/mru75/>) and runs from Nov 2021 through Apr 2022.

The screenshot shows a project page on the OSF Registry platform. The top navigation bar includes links for 'Add New', 'My Registrations', 'Help', 'Donate', 'Join', and 'Login'. A message at the top states: 'have increased our measures to flag spam content on OSF. Contact support@osf.io if you believe your content has been flagged in error.' The main title of the project is 'Comparison and combination of real-time COVID19 forecasts in Germany and Poland'. Below the title, there's a 'Public registration' button and some social sharing icons. On the left, a sidebar menu lists 'Overview' (selected), 'Files', 'Wiki', 'Components' (0), 'Links' (0), 'Analytics', and 'Comments' (0). The 'Summary' section contains a detailed description of the project's purpose and methodology. It mentions that the registration serves to ensure transparency and provides rules and criteria for the study. An attached PDF is mentioned. The 'Contributors' section lists 'Johannes Bracher'. The 'Description' section provides a detailed overview of the project's goal: to improve situational awareness and provide an additional element to inform public health decision making during the COVID19 pandemic. It notes that while few prediction models were available early in the pandemic, there is now a growing number of forecasts based on diverse methods and data streams.

OSF REGISTRIES ▾

Add New My Registrations Help Donate Join Login

have increased our measures to flag spam content on OSF. Contact support@osf.io if you believe your content has been flagged in error.

Comparison and combination of real-time COVID19 forecasts in Germany and Poland

Public registration ▾

Overview

Files

Wiki

Components 0

Links 0

Analytics

Comments 0

Summary

Provide a narrative summary of what is contained in this registration or how it differs from prior registrations. If this project contains documents for a preregistration, please note that here.

This registration serves to ensure a transparent set of rules and criteria to guide the study. Details are provided in the attached PDF.

Add supplemental files or additional information

- Preregistration.pdf

Contributors

Johannes Bracher

Description

Short-term forecasts of cases, deaths and hospitalizations can improve situational awareness and provide an additional element to inform public health decision making during the COVID19 pandemic. While early in the pandemic only few prediction models were available, there is now a growing number of forecasts based on diverse methods and data streams. This project

A quick tangent into forecast evaluation

Proper scoring rules

The highest expected reward is given if the true probability distribution is supplied as the forecast.

Here the continuous ranked probability score (CRPS) and its approximate cousin the weighted interval score (WIS) are used to evaluate forecasts.

The CRPS is defined as,

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(x) - \mathcal{H}(x \geq y))^2 dx$$

F is the CDF, \mathcal{H} is a step function, y is the true value, and x is the forecast.

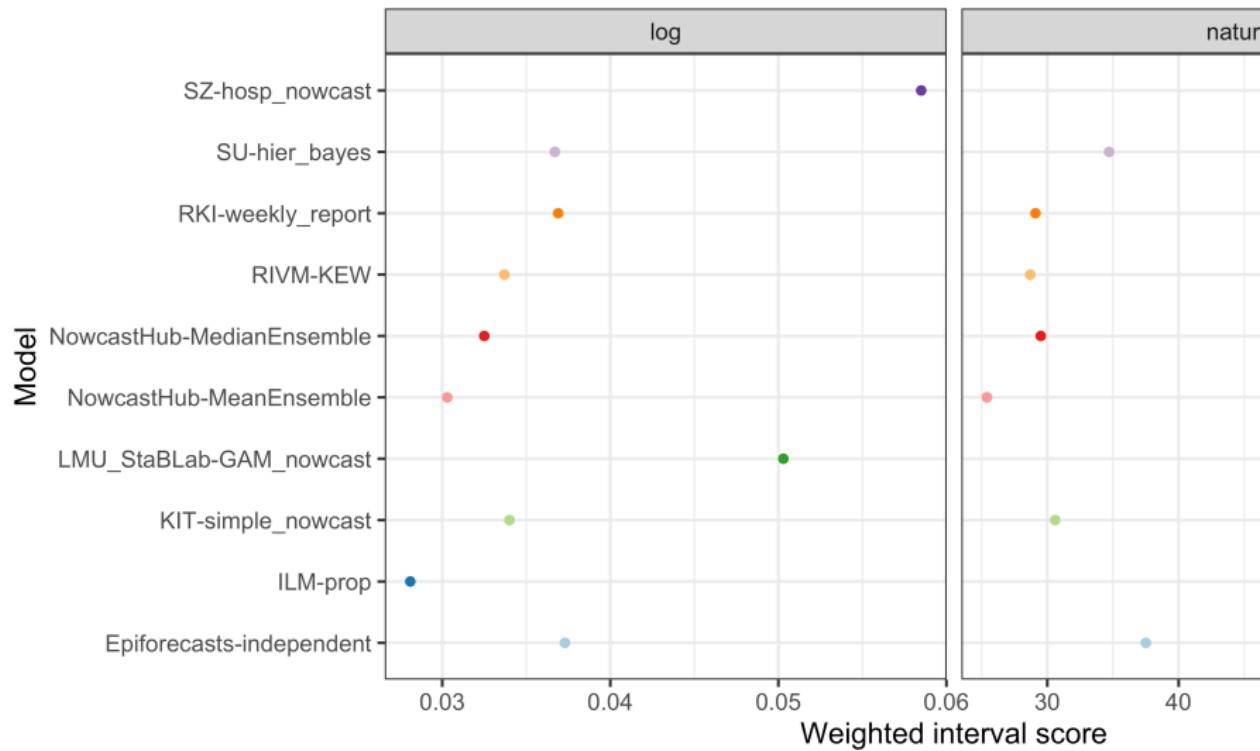
Proper scoring rules

The highest expected reward is given if the true probability distribution is supplied as the forecast.

- A generalisation of absolute error to a probabilistic setting.
- If we take the log of observations and forecasts and calculate the CRPS it becomes an approximate generalisation of the relative error.

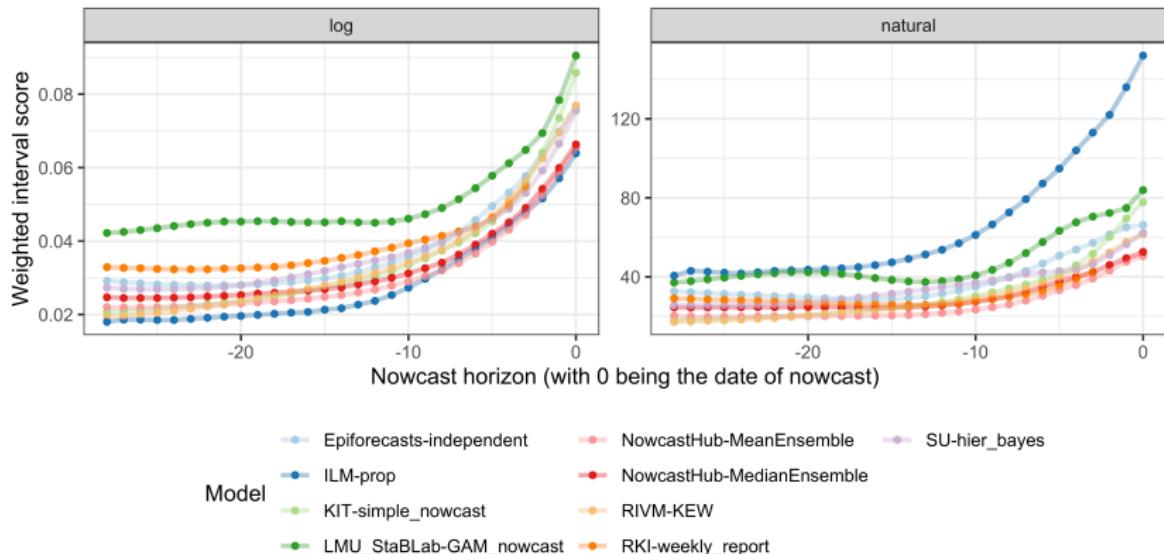
The Germany nowcasting hub - Preliminary Evaluation

Overall:



The Germany nowcasting hub - Preliminary Evaluation

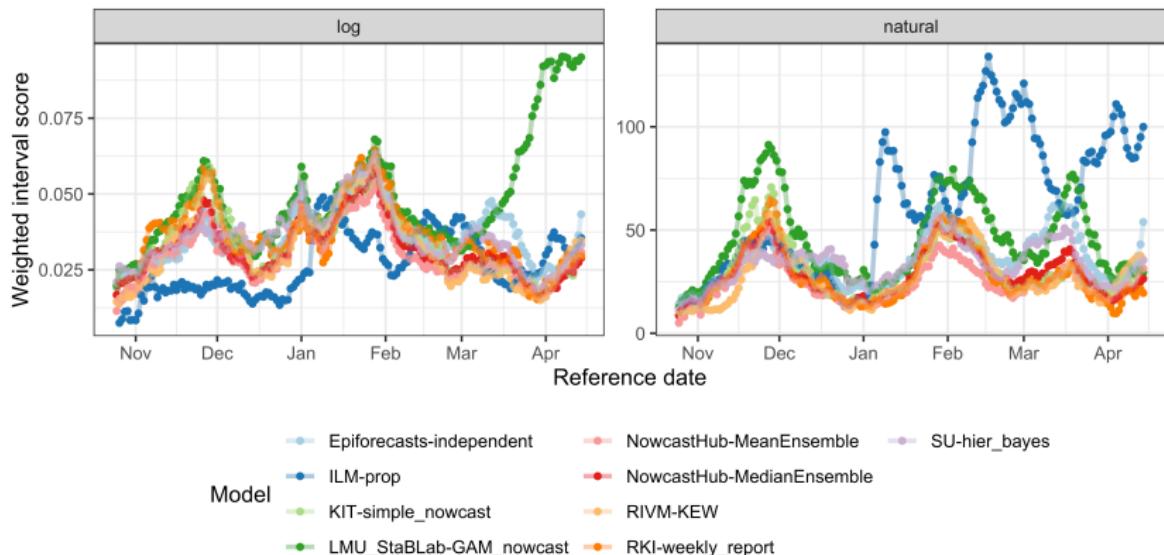
By horizon:



See more: <https://epiforecasts.io/eval-germany-sp-nowcasting/>

The Germany nowcasting hub - Preliminary Evaluation

By date of test positivity:



See more : <https://epiforecasts.io/eval-germany-sp-nowcasting/>

The Germany nowcasting hub - Summary

- In most cases, nowcasts have conveyed a good picture of actual trends.
- Most methods are a bit confident
- Sometimes even the ensemble of all models is very clearly wrong.
- The hub ensemble is generally somewhat better than any individual model
- Collaboratively comparing models allows us to learn about which methods work best.

The epinowcast model

Follow along at home

epinowcast 0.0.6-2000 Reference Articles • Changelog

Search for

Hierarchical nowcasting of right censored epidemiological counts

Published 0.0.6-2000 License MIT Dependencies 3
0.0.6 (0.55 kB) news/51737500

This package contains tools to enable flexible and efficient hierarchical nowcasting of right censored epidemiological counts using a semi-mechanistic Bayesian method with support for both day of reference and day of report effects. Nowcasting in this context is the estimation of the total notifications (for example hospitalisations or deaths) that will be reported for a given date based on those currently reported and the pattern of reporting for previous days. This can be useful when tracking the spread of infectious disease in real-time as otherwise changes in trends can be obfuscated by partial reporting or their detection may be delayed due to the use of simpler methods like truncation.

Installation

Installing the package

Install the stable development version of the package with:

```
install.packages("epinowcast", repos = "https://epiforecasts.r-universe.dev")
```

Install the unstable development from GitHub using the following,

```
remotes::install_github("epiforecasts/epinowcast", dependencies = TRUE)
```

Installing CmdStan

If you don't already have CmdStan installed then, in addition to installing `epinowcast`, it is also necessary to install CmdStan using CmdStan's `install_CmdStan()` function to enable model fitting in `epinowcast`. A suitable C++ toolchain is also required. Instructions are provided in the [Getting started with CmdStan](#) vignette. See the [CmdStanR documentation](#) for further details and support.

Links

[Browse source code](#)
[Report a bug](#)

License

[Full license](#)
[MIT + file LICENSE](#)

Community

[Code of conduct](#)

Citation

[Citing `epinowcast`](#)

Developers

[Sam Abbott](#) Author, maintainer
[Adrian Lison](#) Author
[More about authors...](#)

Dev status

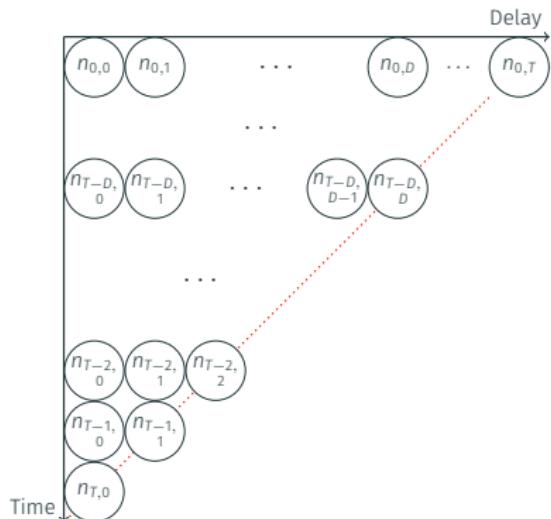
[RStudio connector](#)
 [X GitHub project](#)
 [Crossref DOI](#)

<https://epiforecasts.io/epinowcast/dev/>

The statistical problem - Completing the reporting triangle

Available data at Day T ('now'), per strata $s = 1 \dots, S$:

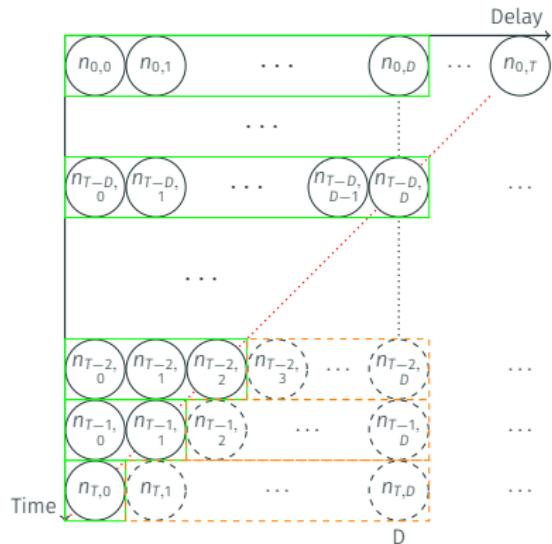
- $n_{t,d,s}$: Count on the reference day t , reported d days after (i.e. at Day $t+d$) for all $t+d \leq T$
- $N_s(t, T) = \sum_{d=0}^{T-t} n_{t,d,s}$: Overall count for reference day t until day T (now)



The statistical problem - Completing the reporting triangle

Aim of nowcasting:

- Predict $N_s(t, \infty) = \sum_{d=0}^{\infty} n_{t,d,s}$ for all days $t \leq T$ based on available information at day T
- Corresponds to the prediction of (so far unseen) $N_s(t, \infty) - N_s(t, T)$
- In practice, one defines a maximum reporting delay D , e.g. $D = 35$ days,
 $N_s(t, \infty) = \sum_{d=0}^D n_{t,d,s}$



The epinowcast model - the basic idea

- Building up on Höhle [1], McGough [2], and Günther [3].
- General idea: separate nowcasting problem into two “sub-models”
 1. Model for the expected number of final notifications
 2. Model for *delay distribution* of the reporting process. This model is further split into 3 sub-models.
 - 2.1 The baseline hazard model.
 - 2.2 The report day hazard model.
 - 2.3 The reference day hazard model.
- When groups of observations (i.e age or location) are present we can either choose to model jointly or independently.

The epinowcast model - expected final notifications sub-model

Here we follow Günther [3] and specify a group specific daily random walk on the log scale. This will soon be generalisable to arbitrary models.

$$\log(\lambda_{gt}) \sim \text{Normal}(\log(\lambda_{gt-1}), \sigma_g^\lambda)$$

$$\log(\lambda_{g0}) \sim \text{Normal}(\log(N_{g0}), 1)$$

$$\sigma_g^\lambda \sim \text{Half-Normal}(0, 1)$$

Notation

λ_{gt} : expected number of hospitalizations in group g with a reference date at day $t = 0, \dots, T$

The epinowcast model - delay distribution model

We define the delay distribution (p_{gtd}) as a discrete time hazard model:

$$h_{gtd} = P(\text{delay} = d | \text{delay} \geq d, W_{gtd})$$

We extend this model to decompose W_{gtd} into 3 components:

1. Hazard derived from a parametric delay distribution (γ_{gtd}) dependent on covariates at the date of occurrence.
2. Hazard not derived from a parametric distribution (δ_{gtd}) dependent on covariates at the date of occurrence.
3. Hazard dependent on covariates referenced to the date of report (ϵ_{gtd}).

The epinowcast model - baseline hazard model

We assume that the probability of reporting p'_{gtd} on a given date follows a parametric distribution with the summary parameters defined using reference date indexed fixed (α_i) and random (β_i) coefficients,

$$p'_{gtd} \sim \text{LogNormal}(\mu_{gt}, v_{gt})$$

$$\mu_{gt} = \mu_0 + \alpha_\mu X_\gamma + \beta_\mu Z_\gamma$$

$$v_{gt} = \exp(v_0 + \alpha_v X_\gamma + \beta_v Z_\gamma)$$

The parametric logit hazard for this component of the model is then,

$$\gamma_{gtd} = \text{logit} \left(\frac{p'_{gtd}}{\left(1 - \sum_{d'=0}^{d-1} p'_{gtd'} \right)} \right)$$

If we defined this directly using daily hazard terms we would have defined the Cox model.

The epinowcast model - proportional hazard models

We then define our two sub-models that assume proportional hazards.

These act based on the reference date and report date respectively (i.e the first assumes all reports from a given reference day are impacted and the second assumes all reports that occur on a given day are impacted).

Similar these are specified with fixed (α_i) and random (β_i) coefficients.

$$\delta_{gtd} = \mu_0 + \alpha_\delta X_\delta + \beta_\delta Z_\delta \quad (1)$$

$$\epsilon_{gtd} = \epsilon_0 + \alpha_\epsilon X_\epsilon + \beta_\epsilon Z_\epsilon \quad (2)$$

The epinowcast model - Overall hazard and probability of report

The overall hazard for each group, occurrence time, and delay is then,

$$\text{logit}(h_{gtd}) = \gamma_{gtd} + \delta_{gtd} + v_{gtd}, \quad h_{gtD} = 1$$

The probability of report for a given delay, occurrence date, and group is then as follows,

$$p_{gt0} = h_{gt0}, \quad p_{gtd} = \left(1 - \sum_{d'=0}^{d-1} p_{gtd'}\right) \times h_{gtd}$$

The epinowcast model - Observation model

Expected notifications by time of occurrence (t) and reporting delay can now be found by multiplying expected final notifications for each t with the probability of reporting for each day of delay (p_{gtd}).

$$n_{gtd} \mid \lambda_{gt}, p_{gtd} \sim \text{NB}(\lambda_{gt} \times p_{gtd}, \phi), \quad t = 1, \dots, T.$$

We produce a nowcast of final observed notifications at each occurrence time by summing posterior estimates for each observed notification for that occurrence time.

$$N_{gt} = \sum_{d=0}^D n_{gtd}$$

The epinowcast model - Summary

- Phew that was a lot. Can you see why we need a nice and friendly package!
- This is all really a complex regression.
- We can also think of it as a decomposed regression and survival model (i.e Cox and friends).
- The flexible structure outlined here allows us to define a range of models including day of the week effects, random walks by week etc.

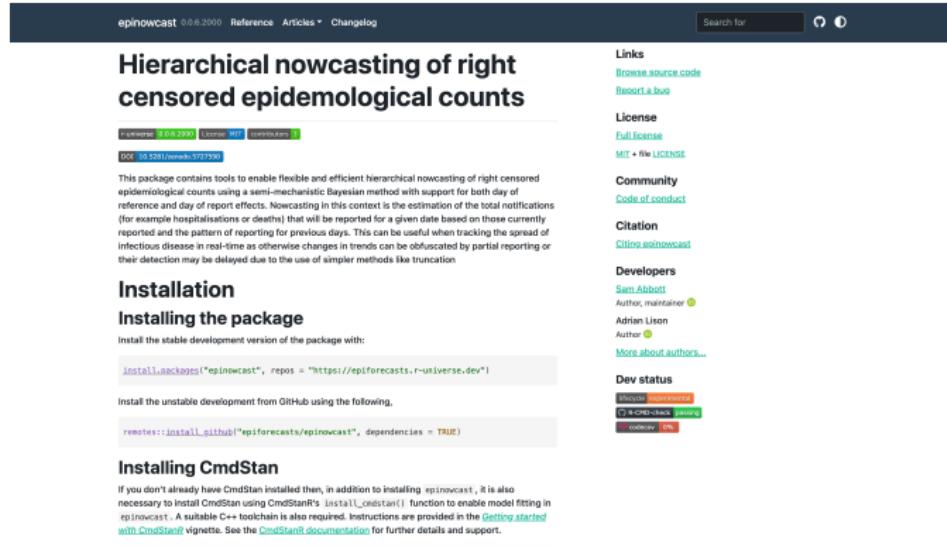
The epinowcast R package

The epinowcast R package - Why?

- Previous nowcasting implementations have either been question specific or rigidly defined.
- The nowcasting hub has highlighted the potential complexity of models.
- It has also highlighted issues with comparison as there is no easy way for one researcher to run all the models.
- Nowcasting is at the core of many real-time analysis questions but is often not the focus. We want to improve this step for everyone.

The epinowcast R package - What?

An in development R package:



The screenshot shows the R-Forge page for the epinowcast package. At the top, there's a navigation bar with links for 'epinowcast 0.0.6.2000', 'Reference', 'Articles', and 'Changelog'. A search bar and a 'Logout' button are also present. Below the header, the title 'Hierarchical nowcasting of right censored epidemiological counts' is displayed in large bold letters. To the left of the main content, there are download links for 'Source [0.0.6.2000]' (gzipped tarball), 'License MIT', and 'Contributors'. There's also a link to 'DOI 10.5281/zenodo.5777590'. The main text describes the package's purpose: enabling flexible and efficient hierarchical nowcasting of right censored epidemiological counts using a semi-mechanistic Bayesian method. It mentions the estimation of total notifications from hospitalizations or deaths, and how it can track infectious disease spread in real-time by accounting for reporting delays. Below this, there's a section titled 'Installation' with a sub-section 'Installing the package'. It provides instructions for installing the stable version via 'install.packages()' and the unstable version from GitHub using 'remotes::install_github()'. There's also a note about installing CmdStan. On the right side of the page, there are sections for 'Links' (with 'Browse source code' and 'Report a bug'), 'License' (with 'Full license' and 'MIT + file LICENSE'), 'Community' (with 'Code of conduct'), 'Citation' (with 'Citing epinowcast'), 'Developers' (listing 'Sam Abbott' as author and maintainer, and 'Adrian Leon' as author), and 'Dev status' (showing 'Wrecker' as the CI status, 'CI in CI+CI+ check passed', and 'Continuous' with a green status bar).

Hierarchical nowcasting of right censored epidemiological counts

Source [0.0.6.2000] License MIT Contributors DOI 10.5281/zenodo.5777590

This package contains tools to enable flexible and efficient hierarchical nowcasting of right censored epidemiological counts using a semi-mechanistic Bayesian method with support for both day of reference and day of report effects. Nowcasting in this context is the estimation of the total notifications (for example hospitalisations or deaths) that will be reported for a given date based on those currently reported and the pattern of reporting for previous days. This can be useful when tracking the spread of infectious disease in real-time as otherwise changes in trends can be obfuscated by partial reporting or their detection may be delayed due to the use of simpler methods like truncation

Installation

Installing the package

Install the stable development version of the package with:

```
install.packages("epinowcast", repos = "https://epiforecasts.r-universe.dev")
```

Install the unstable development from GitHub using the following,

```
remotes::install_github("epiforecasts/epinowcast", dependencies = TRUE)
```

Installing CmdStan

If you don't already have CmdStan installed then, in addition to installing `epinowcast`, it is also necessary to install CmdStan using CmdStan's `install_CmdStan()` function to enable model fitting in `epinowcast`. A suitable C++ toolchain is also required. Instructions are provided in the [Getting started with CmdStan vignette](#). See the [CmdStan documentation](#) for further details and support.

See more: <https://epiforecasts.io/epinowcast/dev/>

The epinowcast R package - What?

Highly optimised stan implementation:

```
139 model {
140   profile("model_priors") {
141     // priors for unobserved expected reported cases
142     leobs_init ~ normal(eobs_init, 1);
143     eobs_lsd ~ zero_truncated_normal(eobs_lsd_p[1], eobs_lsd_p[2]);
144     for (i in 1:g) {
145       leobs_resids[i] ~ std_normal();
146     }
147     // priors for the intercept of the log normal truncation distribution
148     logmean_int ~ normal(logmean_int_p[1], logmean_int_p[2]);
149     logsd_int ~ normal(logsd_int_p[1], logsd_int_p[2]);
150     // priors and scaling for date of reference effects
151     if (neffs) {
152       logmean_eff ~ std_normal();
153       logsd_eff ~ std_normal();
154       if (neff_sds) {
155         logmean_sd ~ zero_truncated_normal(logmean_sd_p[1], logmean_sd_p[2]);
156         logsd_sd ~ zero_truncated_normal(logsd_sd_p[1], logsd_sd_p[2]);
157       }
158     }
159     // priors and scaling for date of report effects
160     if (nrd_effs) {
161       rd_eff ~ std_normal();
162       if (nrd_eff_sds) {
163         rd_eff_sd ~ zero_truncated_normal(rd_eff_sd_p[1], rd_eff_sd_p[2]);
164       }
165     }
166     // reporting overdispersion (1/sqrt)
167     sqrt_phi ~ normal(sqrt_phi_p[1], sqrt_phi_p[2]) T@0,];
168   }
169   // log density: observed vs model
170   if (likelihood) {
171     profile("model_likelihood") {
172       target += reduce_sum(obs_lupmf, st, 1, flat_obs, sl, csl, imp_obs, sg, st,
173                           rdlurd, srldlh, ref_lh, dpmfs, ref_p, phi);
174     }
175   }
176 }
```

See more: <https://github.com/epiforecasts/epinowcast/blob/main/inst/stan/epinowcast.stan>

The epinowcast R package - What?

Developed in the open on GitHub:

	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
3 Open · 25 Closed							
Feature optional profiling include paths × enhancement	#64 opened 3 days ago by adrian-lison · Review required						2
Vectorise hazard_to_prob × enhancement	#65 opened 14 days ago by seabbs · Review required						7
Feature vectorised truncated dists ✓	#61 by seabbs was merged 29 days ago · Review required						4
Refactor functions pmfs ✓ enhancement	#60 by adrian-lison was merged 29 days ago · Approved						3
Revert "Refactor functions/pmfs"	#49 by adrian-lison was merged 29 days ago						
Fix bug causing pkgdown to fail bug	#48 by seabbs was merged on 13 Apr · Approved			1			3
Bugfix optional profiling ✓ bug	#46 by adrian-lison was merged on 12 Apr · Approved			1			3
Update CmdStan version ✓	#45 by adrian-lison was merged on 12 Apr · Approved						2
Refactor functions/pmfs ✓ enhancement	#42 by adrian-lison was merged 29 days ago · Approved						18
Optional profiling ✓ enhancement	#41 by adrian-lison was merged on 12 Apr · Approved			1			20
Feature vectorise likelihood × enhancement	#40 by seabbs was merged on 12 Apr · Review required						24
Vectorised truncation distributions ✓ enhancement help wanted	#38 by seabbs was merged 29 days ago · Approved						22
Feature update docs ✘	#37 by seabbs was merged on 24 Mar · Review required						
Drop logit and inv_logit × enhancement	#36 by seabbs was closed on 24 Mar · Review required						2

See more: <https://github.com/epiforecasts/epinowcast/pulls>

The epinowcast R package - What?

An active slack and monthly meeting:

The screenshot shows the Slack interface for the `epinowcast` channel. The left sidebar lists various channels and apps. The main pane displays two messages from different users:

Sam Abbott 7:46 PM
Borrowing an idea from [@Adrian.Lison](#) <https://github.com/epiforecasts/epinowcast/pull/53>
#53 Vectorise hazard_to_prob
This PR makes use of ideas from [@adrian-lison](#) and implemented [here](#) to vectorise `hazard_to_prob`. It differs from this implementation in two ways. Firstly it does not assume the complete hazard is available (due to this not being the case in the likelihood call) and therefore does not optimise the probability at the maximum delay. Secondly, it makes use of `logsum` for a slight performance boost and numerical stability.
It also adds slight optimisations to `prob_to_hazard` so that only required cumulative probabilities are calculated.
Initial testing suggests this gives a small speedup at the cost of slightly less clear code.

Labels enhancement
Assignees [@seabbs](#)

Tuesday, May 10th

Adrian Lison 12:18 PM
Optional profiling also for included .stan files <https://github.com/epiforecasts/epinowcast/pull/54>
#54 Feature optional profiling include paths
This PR extends the optional profiling also to .stan files in the include paths, and moves the whole functionality to a separate function which is called by `ens_mcmc`, if required. All manipulated stan files with profiling statements removed are stored in the same temporary directory.
Implementation details: By using the same folder structure in the temporary directory as in the original include paths, the relative include paths still apply and the model code is not further touched aside from removing the profiling. This should make the approach very robust. Also, the names of the main model and the included file paths are now directly mirrored, so that warnings or errors from stan c stating a... [Show more](#)

Labels enhancement
Comments 1

Tuesday, May 10th

Send a message to `#features`

Icons for message, file, and other functions are visible at the bottom.

The epinowcast R package - What?

Case studies:

The screenshot shows a web page for the `epinowcast` R package version 0.0.6.2000. The top navigation bar includes links for Reference, Articles (with a dropdown menu), and Changelog. A search bar and a user icon are also present. The main content area features a title "Hierarchical nowcasting of age stratified COVID-19 hospitalisations in Germany" by Sam Abbott, with a source link to a vignette. Below the title is a detailed text about the vignette's purpose and methodology. A "Packages" section lists dependencies: `epinowcast`, `data.table`, `purrr`, `knitr`, `gridExtra`, `grid`, `ggridges`, `loo`, `scoringutils`, and `knitr`. A note at the bottom discusses thread usage for model fitting.

Hierarchical nowcasting of age stratified COVID-19 hospitalisations in Germany

Sam Abbott

Source: [vignettes/germany-age-stratified-nowcasting.Rmd](#)

In this vignette we explore using `epinowcast` to estimate COVID-19 hospitalisations by date of positive test in Germany stratified by age using several model specifications with different degrees of flexibility. We then evaluate the resulting nowcasts using visual checks, approximate leave-one-out (LOO) cross-validation using Pareto smoothed importance sampling, and out of sample scoring using the weighted interval score and other scoring measures for the single report date considered here. Before working through this vignette reading the model definition is advised (`vignette("model-definition")`)

Packages

We use the `epinowcast` package, `data.table` and `purrr` for data manipulation, `ggplot2` for plotting, `knitr` to produce tables of output, `loo` to approximately evaluate out of sample performance and `scoringutils` to evaluate out of sample forecast performance.

```
library(epinowcast)
library(data.table)
library(purrr)
library(ggplot2)
library(loo)
library(scoringutils)
library(knitr)
```

See more: <https://epiforecasts.io/epinowcast/articles/germany-age-stratified-nowcasting.html>

The epinowcast R package - Summary

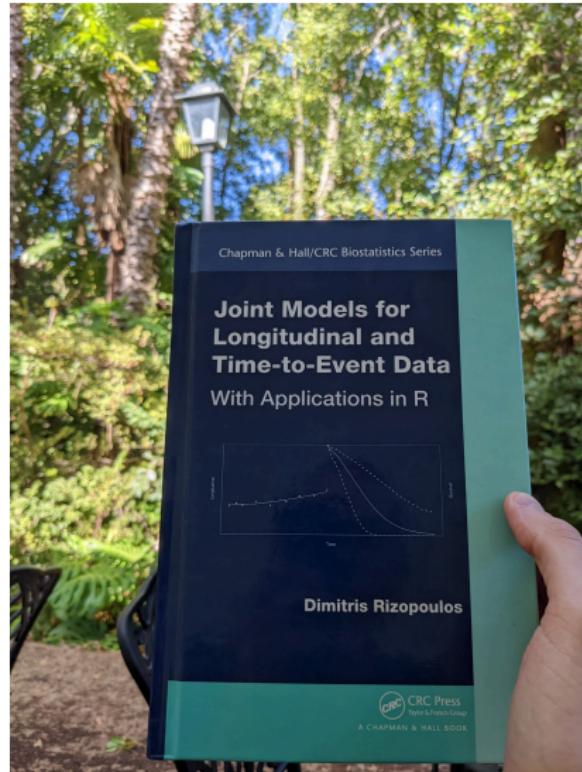
- A flexible nowcasting framework that can fit a range of nowcasting models.
- A community driven package with regular discussions.
- A focus on optimisation and new methodology development.
- Built using software development best practices .
- Evaluated in real-time as part of the Germany nowcasting hub

Extensions

Extensions - Coming soon

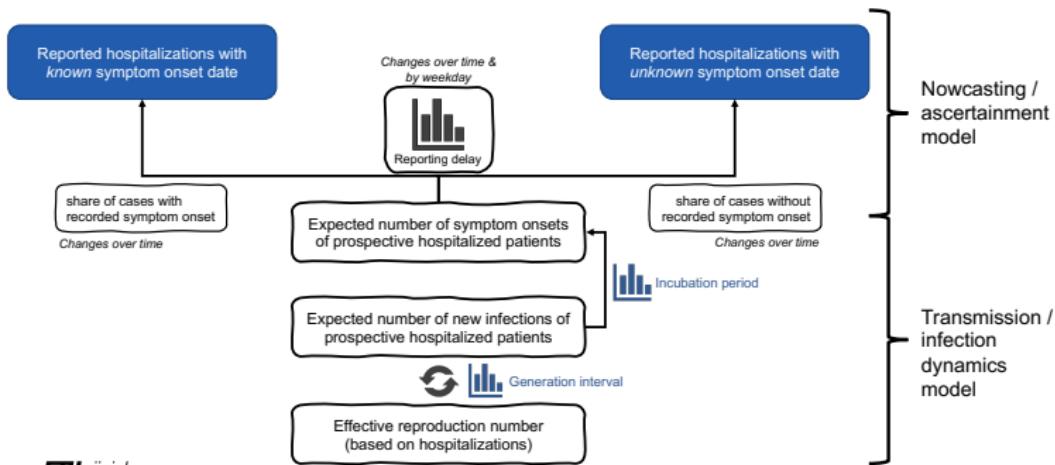
- A full featured formula interface.
- A flexible expectation model.
- The ability to forecast into the future.
- An extension to model missing data from Adrian Lison.
- More software development.

Extensions - Connection to survival models



Extensions - Latent infection modelling using renewal equations

Nowcasting R_t from hospitalization linelist data Bayesian hierarchical model



Summary

Summary

- Because of delays between infections and observation truncation is everywhere when studying infectious disease models.
- Multiple models exist to account for this and most perform quite well.
- In general though these models are not used as part of wider practice.
- The epinowcast R package aims to change that.
- There is lots of interesting development to be done and lots of exciting use cases and extensions. Please reach out if interested!

References i

-  Michael Höhle and Matthias an der Heiden.
Bayesian nowcasting during the STEC O104: H4 outbreak in Germany, 2011.
Biometrics, 70(4):993–1002, 2014.
-  Sarah F McGough, Michael A Johansson, Marc Lipsitch, and Nicolas A Menzies.
Nowcasting by Bayesian Smoothing: A flexible, generalizable model for real-time epidemic tracking.
PLoS computational biology, 16(4):e1007735, 2020.
-  Felix Günther, Andreas Bender, Katharina Katz, Helmut Küchenhoff, and Michael Höhle.
Nowcasting the COVID-19 pandemic in Bavaria.
Biometrical Journal, 63(3):490–502, 2021.