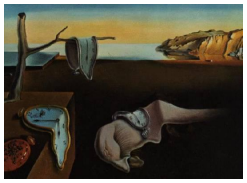

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

hubEnsembles: Ensembling Methods in R

Li Shandross
University of Massachusetts Amherst

Emily Howerton
The Pennsylvania State University

Lucie Contamin
University of Pittsburgh

Harry Hochheiser
University of Pittsburgh

Anna Krystalli
R-RSE SMPC

Consortium of
Infectious Disease Modeling Hubs
A list of authors and their affiliations appears at the end of the paper

Nicholas G. Reich
University of Massachusetts Amherst

Evan L. Ray
University of Massachusetts Amherst

Abstract

Combining predictions from multiple models into an ensemble is a widely used practice across many fields with demonstrated performance benefits. The R package **hubEnsembles** provides a flexible framework for ensembling various types of predictions, including point estimates and probabilistic predictions. A range of common methods for generating ensembles are supported, including weighted averages, quantile averages, and linear pools. The **hubEnsembles** package fits within a broader framework of open-source software and data tools called the "hubverse", which facilitates the development and management of collaborative modelling exercises.

Keywords: multiple models, aggregation, forecast, prediction.

1. Introduction

Predictions of future outcomes are essential to planning and decision making, yet generating reliable predictions of the future is challenging. One method for overcoming this challenge is combining predictions across multiple, independent models. These combination methods

(also called aggregation or ensembling) have been repeatedly shown to produce predictions that are more accurate (Clemen 1989; Timmermann 2006) and more consistent (Hibon and Evgeniou 2005) than individual models. Because of the clear performance benefits, multi-model ensembles are commonplace across fields, including weather (Alley *et al.* 2019), climate (Tebaldi and Knutti 2007), and economics (Aastveit *et al.* 2018). More recently, multi-model ensembles have been used to improve predictions of infectious disease outbreaks (Viboud *et al.* 2018; Johansson *et al.* 2019; McGowan *et al.* 2019; Reich *et al.* 2019; Cramer *et al.* 2022).

In the rapidly growing field of outbreak forecasting, there are many proposed methods for generating ensembles. Generally, these methods differ in at least one of two ways: (1) the function used to combine or "average" predictions, and (2) how predictions are weighted when performing the combination. No one method is universally "the best"; a simple average of predictions works surprisingly well across a range of settings (McGowan *et al.* 2019; Paireau *et al.* 2022; Ray *et al.* 2023) for established theoretical reasons (Winkler 2015). However, more complex approaches have also been shown to have benefits in some settings (Yamana *et al.* 2016; Ray and Reich 2018; Reich *et al.* 2019; Colón-González *et al.* 2021). Here, we present the **hubEnsembles** package, which provides a flexible framework for generating ensemble predictions from multiple models. Complementing other software for combining predictions from multiple models (e.g., (Pedregosa *et al.* 2011; Weiss *et al.* 2019; Bosse *et al.* 2023; Couch and Kuhn 2023)), **hubEnsembles** supports multiple types of predictions, including point estimates and different kinds of probabilistic predictions. Throughout, we will use the term "prediction" to refer to any kind of model output that may be combined including a forecast, a scenario projection, or a parameter estimate.

The **hubEnsembles** package is part of the "hubverse" collection of open-source software and data tools. The hubverse project facilitates the development and management of collaborative modelling exercises (Consortium of Infectious Disease Modeling Hubs 2024). The broader hubverse initiative is motivated by the demonstrated benefits of collaborative hubs (Reich *et al.* 2022; Borchering *et al.* 2023), including performance benefits of multi-model ensembles and the desire for standardization across such hubs. In this paper, we focus specifically on the functionality encompassed in **hubEnsembles**. We provide an overview of the methods implemented, including mathematical definitions and properties (Section 2) as well as implementation details (Section 3); we give simple examples to demonstrate the functionality (Section 5) and a more complex case study (Section 6) that motivates a discussion and comparison of the various methods (Section 7).

2. Mathematical definitions and properties of ensemble methods

The **hubEnsembles** package supports both point predictions and probabilistic predictions of different formats. A point prediction gives a single estimate of a future outcome while a probabilistic prediction provides an estimated probability distribution over a set of future outcomes. We use N to denote the total number of individual predictions that the ensemble will combine. For example, these predictions will often be produced by different statistical or mathematical models, and N is the total number of models that have provided predictions. Individual predictions will be indexed by the subscript i. Optionally, the package allows for calculating ensembles that use a weight w_i for each prediction; we define the set of model-specific weights as $\boldsymbol{w} = \{w_i | i \in 1, ..., N\}$. Informally, predictions with a larger weight have

a greater influence on the ensemble prediction, though the details of this depend on the ensemble method (described further below).

For a set of N point predictions, $\boldsymbol{p} = \{p_i | i \in 1, ..., N\}$, each from a distinct model i, the **hubEnsembles** package can compute an ensemble of these predictions

$$p_E = C(\boldsymbol{p}, \boldsymbol{w})$$

using any function C and a any set of model-specific weights $\boldsymbol{w}$. For example, an arithmetic average of predictions yields $p_E = \sum_{i=1}^{N} p_i w_i$, where the weights are non-negative and sum to 1. If $w_i = 1/N$ for all i, all predictions will be equally weighted. This framework can also support more complex functions for aggregation, such as a (weighted) median or geometric mean.

For probabilistic predictions, there are two commonly used classes of methods to average or ensemble multiple predictions: quantile averaging (also called a Vincent average (Vincent 1912)) and probability averaging (also called a distributional mixture or linear opinion pool (Stone 1961)) (Lichtendahl *et al.* 2013). To define these two classes of methods, let $F(x)$ be a cumulative density function (CDF) defined over values x of the target variable for the prediction, and $F^{-1}(\theta)$ be the corresponding quantile function defined over quantile levels $\theta \in [0, 1]$. Throughout this article, we may refer to x as either a 'value of the target variable' or a 'quantile' depending on the context, and similarly we may refer to θ as either a 'quantile level' or a '(cumulative) probability'. Additionally, we will use $f(x)$ to denote a probability mass function (PMF) for a prediction of a discrete variable or a discretization (such as binned values) of a continuous variable.

The quantile average combines a set of quantile functions, $\mathcal{Q} = \{F_i^{-1}(\theta) | i \in 1, ..., N\}$, with a given set of weights, $\boldsymbol{w}$, as

$$F_Q^{-1}(\theta) = C_Q(\mathcal{Q}, \boldsymbol{w}) = \sum_{i=1}^{N} w_i F_i^{-1}(\theta).$$

This computes the average value of predictions across different models for each fixed quantile level θ. It is also possible to use other combination functions, such as a weighted median, to combine quantile predictions.

The probability average or linear pool is calculated by averaging probabilities across predictions for a fixed value of the target variable, x. In other words, for a set $\mathcal{F} = \{F_i(x) | i \in 1, ..., N\}$ containing the values of CDFs at the point x and weights $\boldsymbol{w}$, the linear pool is calculated as

$$F_{LOP}(x) = C_{LOP}(\mathcal{F}, \boldsymbol{w}) = \sum_{i=1}^{N} w_i F_i(x).$$

For a set of PMF values, $\{f_i(x) | i \in 1, ..., N\}$, the linear pool can be equivalently calculated: $f_{LOP}(x) = \sum_{i=1}^{N} w_i f_i(x)$.

The different averaging methods for probabilistic predictions yield different properties of the resulting ensemble distribution. For example, the variance of the linear pool is $\sigma^2_{LOP} =$

$\sum_{i=1}^{N} w_i \sigma_i^2 + \sum_{i=1}^{N} w_i (\mu_i - \mu_{LOP})^2$, where μ_i is the mean and σ_i^2 is the variance of individual prediction i, and although there is no closed-form variance for the quantile average, the variance of the quantile average will always be less than or equal to that of the linear pool (Lichtendahl *et al.* 2013). Both methods generate distributions with the same mean, $\mu_Q = \mu_{LOP} = \sum_{i=1}^{N} w_i \mu_i$, which is the mean of individual model means (Lichtendahl *et al.* 2013). The linear pool method preserves variation between individual models, whereas the quantile average cancels away this variation under the assumption it constitutes sampling error (Howerton *et al.* 2023).

3. Model implementation details

To understand how these methods are implemented in **hubEnsembles**, we first must define the conventions employed by the hubverse and its packages for representing and working with model predictions. We begin with a short overview of concepts and conventions needed to utilize the **hubEnsembles** package, supplemented by example predictions provided by the hubverse, then explain the implementation of the two ensembling functions provided by the package, `simple_ensemble` and `linear_pool`.

3.1. Hubverse terminology and conventions

A central concept in the hubverse effort is "model output". Model output is a specially formatted tabular representation of predictions. Each row represents a single, unique prediction with each column providing information about what is being predicted, its scope, and its value. Per hubverse convention, each column serves one of three purposes: denote which model has produced the prediction (called the "model ID"), provide details about what is being predicted (called the "task IDs"), or specify how the prediction is represented (called the "model output representation") (Consortium of Infectious Disease Modeling Hubs 2024).

Predictions are assumed to be generated by distinct models, typically developed and run by a modeling team of one or more individuals. Each model should have a unique identifier that is stored in the `model_id` column. Then, the details of the outcome being predicted can be stored in a series of task ID columns. These task ID columns may also include additional information, such as any conditions or assumptions that were used to generate the predictions (Consortium of Infectious Disease Modeling Hubs 2024). For example, short-term forecasts of incident influenza hospitalizations in the US at different locations and amounts of time in the future might represent this information using a `target` column with the value "wk inc flu hosp", a `location` column identifying the location being predicted, a `reference_date` column with the "starting point" of the forecasts, and a `horizon` column with the number of steps ahead that the forecast is predicting relative to the `reference_date` (Table 1). All these variables make up the task ID columns.

model_id	target	horizon	output_type	output_type_id	value
MOBS-GLEAM_FLUH	wk inc flu hosp	0	quantile	0.25	514

model_id	target	horizon	output_type	output_type_id	value
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.25	563
MOBS-GLEAM_FLUH	wk inc flu hosp	2	quantile	0.25	469
MOBS-GLEAM_FLUH	wk inc flu hosp	3	quantile	0.25	324
MOBS-GLEAM_FLUH	wk inc flu hosp	0	quantile	0.5	596
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.5	664
MOBS-GLEAM_FLUH	wk inc flu hosp	2	quantile	0.5	575
MOBS-GLEAM_FLUH	wk inc flu hosp	3	quantile	0.5	408
MOBS-GLEAM_FLUH	wk inc flu hosp	0	quantile	0.75	713
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.75	803
MOBS-GLEAM_FLUH	wk inc flu hosp	2	quantile	0.75	705
MOBS-GLEAM_FLUH	wk inc flu hosp	3	quantile	0.75	512

Table 1: Example of forecasts for incident influenza hospitalizations, formatted according to hubverse standards. Quantile predictions for the median and 50% prediction intervals from a single model are shown for four distinct horizons. The `location` and `reference_date` columns have been omitted for brevity; all forecasts in this example were made on 2022-12-17 for Massachusetts. This example is a subset of the `forecast_outputs` data provided by the **hubExamples** package.

Alternatively, longer-term scenario projections may require different task ID columns. For example, projections of incident COVID-19 cases in the US at different locations, amounts of time in the future, and under different assumed conditions may use a `target` column of "cum death", a `location` column specifying the location being predicted, an `origin_date` column specifying the date on which the projections were made, a `horizon` column describing the number of steps ahead that the projection is predicting relative to the `origin_date`, and a `scenario_id` column denoting the future conditions that were modeled and are projected to result in the specified number of incident cases (Table 2). Different modeling efforts may use different sets of task ID columns and values to specify their prediction goals. Additional examples of task ID variables are available on the hubverse documentation website (Consortium of Infectious Disease Modeling Hubs 2024).

model_id	target	horizon	scenario_id	output_type	output_type_id	value
HUBuni-simexamp	inc case	26	A-2021-03-05	quantile	0.25	1147.00
HUBuni-simexamp	inc case	26	A-2021-03-05	quantile	0.50	1516.00
HUBuni-simexamp	inc case	26	A-2021-03-05	quantile	0.75	1929.00
HUBuni-simexamp	inc case	26	B-2021-03-05	quantile	0.25	4241.75
HUBuni-simexamp	inc case	26	B-2021-03-05	quantile	0.50	4952.50
HUBuni-simexamp	inc case	26	B-2021-03-05	quantile	0.75	6002.25
HUBuni-simexamp	inc case	26	C-2021-03-05	quantile	0.25	32478.75
HUBuni-simexamp	inc case	26	C-2021-03-05	quantile	0.50	38594.50
HUBuni-simexamp	inc case	26	C-2021-03-05	quantile	0.75	44975.50
HUBuni-simexamp	inc case	26	D-2021-03-05	quantile	0.25	85811.75
HUBuni-simexamp	inc case	26	D-2021-03-05	quantile	0.50	99841.50
HUBuni-simexamp	inc case	26	D-2021-03-05	quantile	0.75	113963.50

Table 2: Example of scenario projections for incident COVID-19 cases, formatted according to hubverse standards. Quantile predictions for the median and 50% prediction intervals from a single model are shown for four distinct scenarios. The `location` and `origin_date` columns have been omitted for brevity; all forecasts in this example were made on 2021-03-07 for the US. This example is a subset of the `scenario_outputs` data provided by the **hubExamples** package.

The third group of columns in model output specify the model predictions and details about how the predictions are represented. This "model output representation" includes the predicted values along with metadata that specifies how the predictions are conveyed and always consists of the same three columns: (1) `output_type`, (2) `output_type_id`, and (3) `value`. The `output_type` column defines how the prediction is represented and may be one of `"mean"` or `"median"` for point predictions, or `"quantile"`, `"cdf"`, `"pmf"`, or `"sample"` for probabilistic predictions (although the sample output type is not yet supported by the **hubEnsembles** package). The `output_type_id` provides additional identifying information for a prediction and is specific to the particular `output_type` (see Table 1). For quantile predictions, the `output_type_id` is a numeric value between 0 and 1 specifying the cumulative probability associated with the quantile prediction. In the notation we defined above, the `output_type_id` corresponds to θ and the `value` is the quantile prediction $F^{-1}(\theta)$. For CDF or PMF predictions, the `output_type_id` is the target variable value x at which the cumulative distribution

function or probability mass function for the predictive distribution should be evaluated, and the `value` column contains the predicted $F(x)$ or $f(x)$, respectively. Requirements for the values of the `output_type_id` and `value` columns associated with each valid output type are summarized in Table 3.

This representation of predictive model output is codified by the `model_out_tbl` S3 class in the **hubUtils** package, one of the foundational hubverse packages. Although this S3 class is required for all **hubEnsembles** functions, model predictions in other formats can easily be transformed using the `as_model_out_tbl()` function from **hubUtils**. An example of this transformation is provided in Section 6.

`output_type`	`output_type_id`	`value`
`mean`	NA (not used for mean predictions)	Numeric: The mean of the predictive distribution
`median`	NA (not used for median predictions)	Numeric: The median of the predictive distribution
`quantile`	Numeric between 0.0 and 1.0: A quantile level	Numeric: The quantile of the predictive distribution at the quantile level specified by the `output_type_id`
`cdf`	Numeric within the support of the outcome variable: a possible value of the target variable	Numeric between 0.0 and 1.0: The cumulative probability of the predictive distribution at the value of the outcome variable specified by the `output_type_id`
`pmf`	String naming a possible category of a discrete outcome variable	Numeric between 0.0 and 1.0: The probability mass of the predictive distribution when evaluated at a specified level of a discrete outcome variable
`sample`	String specifying sample index	Numeric: A sample from the predictive distribution

Table 3: A table summarizing how the model output representation columns are used for predictions of different output types. Adapted from (Consortium of Infectious Disease Modeling Hubs 2024)

4. Ensemble functions in hubEnsembles

The **hubEnsembles** package includes two functions that perform ensemble calculations: `simple_ensemble()`, which applies some function to each model prediction, and `linear_pool()`, which computes an ensemble using the linear opinion pool method. In the following sections, we outline the implementation details for each function and how these implementations correspond to the statistical ensembling methods described in Section 2. A short description of the calculation performed by each function is summarized by output type in Table 4.

Simple ensemble

The `simple_ensemble` function directly computes an ensemble from component model outputs by combining them via some function (C) within each unique combination of task ID variables, output types, and output type IDs. This function can be used to summarize predictions of output types mean, median, quantile, CDF, and PMF. The mechanics of the ensemble calculations are the same for each of the output types, though the resulting statistical ensembling method differs for different output types (Table 4).

By default, `simple_ensemble` uses the mean for the aggregation function C and equal weights for all models. For point predictions with a mean or median output type, the resulting ensemble prediction is an equally weighted average of the individual models' predictions. For probabilistic predictions in a quantile format, by default `simple_ensemble` calculates an equally weighted average of individual model target variable values at each quantile level, which is equivalent to a quantile average. For model outputs in a CDF or PMF format, by default `simple_ensemble` computes an equally weighted average of individual model (cumulative or bin) probabilities at each target variable value, which is equivalent to the linear pool method.

Any aggregation function C may be specified by the user. For example, a median ensemble may also be created by specifying "median" as the aggregation function, or a custom function may be passed to the `agg_fun` argument to create other ensemble types. Similarly, model weights can be specified to create a weighted ensemble.

`output_type`	`simple_ensemble(..., agg_fun="mean")`	`linear_pool()`
`mean`	mean of individual model means	mean of individual model means
`median`	mean of individual model medians	NA
`quantile`	mean of individual model target variable values at each quantile level, $F_Q^{-1}(\theta)$	quantile of the distribution obtained by computing the mean of estimated individual model cumulative probabilities at each target variable value, $F_{LOP}^{-1}(\theta)$
`cdf`	mean of individual model cumulative probabilities at each target variable value, $F_{LOP}(x)$	mean of individual model cumulative probabilities at each target variable value, $F_{LOP}(x)$
`pmf`	mean of individual model bin probabilities at each target variable value, $f_{LOP}(x)$	mean of individual model bin probabilities at each target variable value, $f_{LOP}(x)$

output_type	simple_ensemble(..., agg_fun="mean")	linear_pool()

Table 4: Summary of ensemble function calculations for each output type. The ensemble function (columns) determines the operation that is performed, and in the case of probabilistic output types (quantile, CDF, PMF), this also determines what ensemble distribution is generated (quantile average, $F_Q^{-1}(\theta)$, or linear pool, $F_{LOP}(x)$). The resulting ensemble will be returned in the same output type as the inputs. Thus, the output type (rows) determines how the resulting ensemble distribution is summarized (as a quantile function, $F^{-1}(\theta)$, cumulative distribution function, $F(x)$, or probability mass function $f(x)$). Estimating individual model cumulative probabilities is required to compute a `linear_pool()` for predictions of `quantile` output type; see Section 4.0.2 for details. In the case of `simple_ensemble()`, we report the calculations for the default case where `agg_fun = "mean"`; however, if another aggregation function is chosen (e.g., `agg_fun = "median"`), that calculation would be performed instead. For example, `simple_ensemble(..., agg_fun = "median")` applied to predictions of mean output type would return the median of individual model means.

Linear pool

The `linear_pool` function implements the linear opinion pool method for ensembling predictions. This function can be used to combine predictions with output types mean, quantile, CDF, and PMF. Unlike `simple_ensemble`, this function handles its computation differently based on the output type. For the CDF, PMF, and mean output types, the linear pool method is equivalent to calling `simple_ensemble` with a mean aggregation function (see Table 4), since `simple_ensemble` produces a linear pool prediction (an average of individual model cumulative or bin probabilities).

However, implementation of LOP is less straightforward for the quantile output type. This is because LOP averages cumulative probabilities at each value of the target variable, but the predictions are quantiles (on the scale of the target variable) for fixed quantile levels. The value for these quantile predictions will generally differ between models, and as a result we are typically not provided cumulative probabilities at the same values of the target variable for all component predictions. This lack of alignment between cumulative probabilities for the same target variable values impedes computation of LOP from quantile predictions and is illustrated in panel A of Figure 1.

Given that LOP cannot be directly calculated from quantile predictions, we must first obtain an estimate of the CDF for each component distribution using the provided quantiles, combine the CDFs, then calculate the quantiles from the ensemble's CDF. We perform this calculation in three main steps, assisted by the **distfromq** package (Ray and Gerding 2024) for the first two:

1. Interpolate and extrapolate from the provided quantiles for each component model to obtain an estimate of the CDF of that particular distribution.
2. Draw samples from each component model distribution. To reduce Monte Carlo variability, we use quasi-random samples corresponding to quantiles of the estimated distribution (Niederreiter 1992).

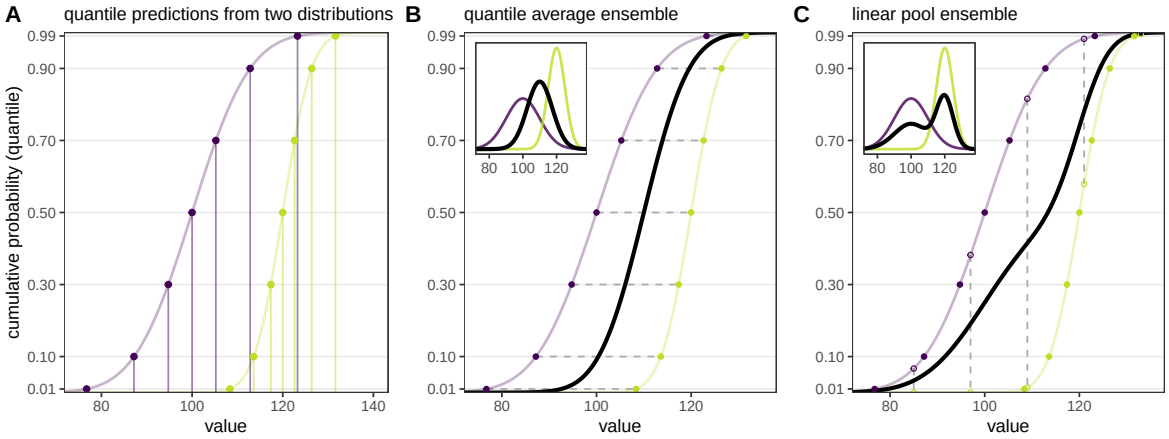

Figure 1: (Panel A) Example of quantile output type predictions. In this example, points show model output collected for seven fixed quantile levels ($\theta = 0.01$, 0.1, 0.3, 0.5, 0.7, 0.9, and 0.99) from two distributions ($N(100, 10)$ in purple and $N(120, 5)$ in green), with the underlying cumulative distribution functions (CDFs) shown with curves. The associated values for each fixed quantile level do not align across distributions (vertical lines). (Panel B) Quantile average ensemble, which is calculated by averaging values for each fixed quantile level (represented by horizontal dashed gray lines). The distributions and corresponding model outputs from panel A are re-plotted and the black line shows the resulting quantile average ensemble. Inset shows corresponding probability density functions (PDFs). (Panel C) Linear pool ensemble, which is calculated by averaging cumulative probabilities for each fixed value (represented by vertical dashed gray lines). The distributions and corresponding model outputs from panel A are re-plotted. To calculate the linear pool in this case, where model outputs are not defined for the same values, the model outputs are used to interpolate the full CDF for each distribution from which quantiles can be extracted for fixed values (shown with open circles). The black line shows the resulting linear pool average ensemble. Inset shows corresponding PDFs.

3. Pool the samples from all component models and extract the desired quantiles.

For step 1, functionality in the **distfromq** package uses a monotonic cubic spline for interpolation on the interior of the provided quantiles. The user may choose one of several distributions to perform extrapolation of the CDF tails. These include normal, lognormal, and cauchy distributions, with "normal" set as the default. A location-scale parameterization is used, with separate location and scale parameters chosen in the lower and upper tails so as to match the two most extreme quantiles. The sampling process described in steps 2 and 3 approximates the linear pool calculation described in Section 2.

5. Basic demonstration of functionality

In this section, we provide a simple example to illustrate the two main functions in **hubEnsembles**, `simple_ensemble()` and `linear_pool()`.

5.1. Example data: a forecast hub

We will use an example hub provided by the hubverse to demonstrate the functionality of the **hubEnsembles** package (Consortium of Infectious Disease Modeling Hubs 2024). The example hub includes both example model output data and target data (sometimes known as "truth" data), which are included in the **hubExamples** package as data objects named `forecast_outputs` and `forecast_target_ts`. The toy dataset of model output contains predictions for only a small subset rows of select dates, locations, and output type IDs, far fewer than an actual modeling hub would typically collect.

The model output data includes quantile, mean and median forecasts of future incident influenza hospitalizations and PMF forecasts of hospitalization intensity. Each forecast is made for five task ID variables, including the location for which the forecast was made (`location`), the date on which the forecast was made (`reference_date`), the number of steps ahead (`horizon`), the date of the forecast prediction (a combination of the date the forecast was made and the forecast horizon, `target_end_date`), and the forecast target (`target`). Table 5 provides an example set of quantile forecasts included in this example model output. In Table 5, we show only the median, the 50%, and 90% prediction intervals, although other intervals and mean forecasts are included in the example model output data.

model_id	target	horizon	output_type	output_type_id	value
Flusight-baseline	wk inc flu hosp	1	quantile	0.05	496
Flusight-baseline	wk inc flu hosp	1	quantile	0.25	566
Flusight-baseline	wk inc flu hosp	1	quantile	0.75	598
Flusight-baseline	wk inc flu hosp	1	quantile	0.95	668
Flusight-baseline	wk inc flu hosp	1	median	NA	582
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.05	446
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.25	563

model_id	target	horizon	output_type	output_type_id	value
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.75	803
MOBS-GLEAM_FLUH	wk inc flu hosp	1	quantile	0.95	1097
MOBS-GLEAM_FLUH	wk inc flu hosp	1	median	NA	664
PSI-DICE	wk inc flu hosp	1	quantile	0.05	290
PSI-DICE	wk inc flu hosp	1	quantile	0.25	496
PSI-DICE	wk inc flu hosp	1	quantile	0.75	712
PSI-DICE	wk inc flu hosp	1	quantile	0.95	843
PSI-DICE	wk inc flu hosp	1	median	NA	613

Table 5: Example model output for forecasts of incident influenza hospitalizations. A subset of example model output is shown: 1-week ahead quantile forecasts made on 2022-12-17 for Massachusetts from three distinct models; only the median, 50% prediction intervals, and 90% prediction intervals are displayed. The `location`, `reference_date` and `target_end_date` columns have been omitted for brevity. This example data is provided in the **hubExamples** package.

We also have corresponding target data included in the **hubExamples** package (Table 6). The example target data provide observed incident influenza hospitalizations (`observation`) in a given week (`date`) and for a given location (`location`). This target data could be used as calibration data for generating forecasts or for evaluating the forecasts post hoc. The forecast-specific task ID variables `reference_date` and `horizon` are not relevant for the target data.

date	location	observation
2022-11-05	25	31
2022-11-12	25	43
2022-11-19	25	79
2022-11-26	25	221
2022-12-03	25	446
2022-12-10	25	578
2022-12-17	25	694
2022-12-24	25	769
2022-12-31	25	733
2023-01-07	25	466
2023-01-14	25	238
2023-01-21	25	122
2023-01-28	25	71

Table 6: Example target data for incident influenza hospitalizations. This table includes target data from 2022-11-01 and 2023-02-01. This target data is provided in the **hubExamples** package.

We can plot these forecasts and the target data using the `plot_step_ahead_model_output()` function from **hubVis**, another package for visualizing model outputs from the hubverse suite (Figure 2). We subset the model output data and the target data to the location and time horizons we are interested in.

```
R> model_outputs_plot <- hubExamples::forecast_outputs |>
+  hubUtils::as_model_out_tbl() |>
+  dplyr::filter(
+    location == "25",
+    output_type %in% c("median", "mean", "quantile"),
+    reference_date == "2022-12-17"
+  )
R> target_data_plot <- hubExamples::forecast_target_ts |>
+  dplyr::filter(
+    location == "25",
+    date >= "2022-11-01", date <= "2023-02-01"
+  )
R> hubVis::plot_step_ahead_model_output(
+  model_output_data = model_outputs_plot,
+  target_data = target_data_plot,
+  facet = "model_id",
+  facet_nrow = 1,
+  interactive = FALSE,
+  intervals = c(0.5, 0.9),
+  show_legend = FALSE,
+  use_median_as_point = TRUE,
+  x_col_name = "target_end_date",
+  x_target_col_name = "date"
+  ) +
+  theme_bw() +
+  labs(y = "incident hospitalizations")
```

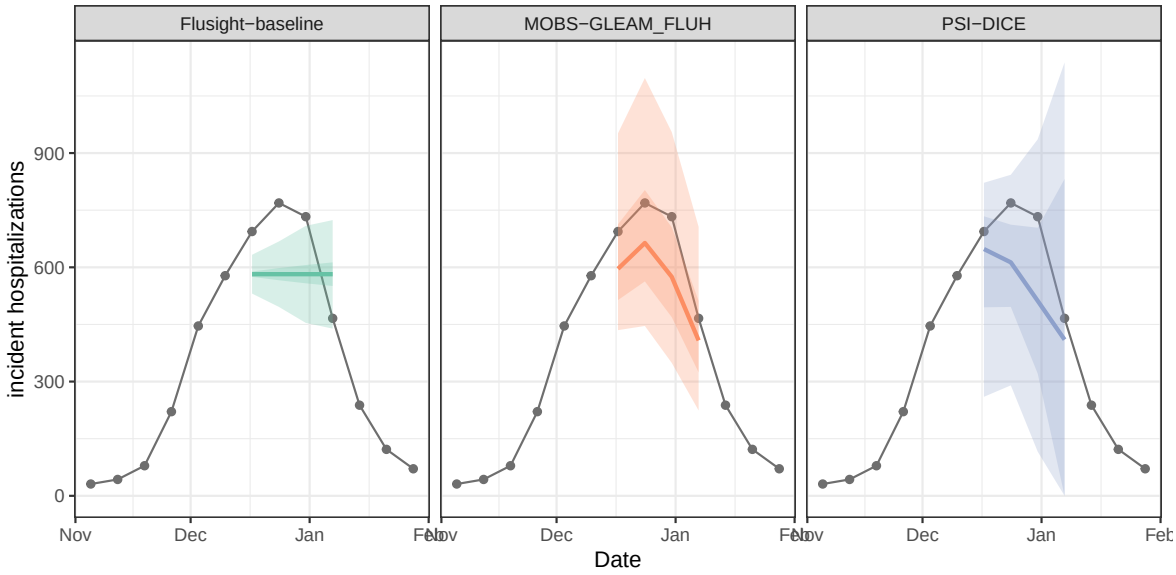

Figure 2: One example quantile forecast of weekly incident influenza hospitalizations in Massachusetts from each of three models (panels). Forecasts are represented by a median (line), 50% and 90% prediction intervals. Gray points represent observed incident hospitalizations.

Next, we examine the PMF target in the example model output data. For this target, teams forecasted the probability that hospitalization intensity will be "low", "moderate", "high", or "very high". These hospitalization intensity categories are determined by thresholds for weekly hospital admissions per 100,000 population. In other words, "low" hospitalization intensity in a given week means few incident influenza hospitalizations per 100,000 population are predicted, whereas "very high" hospitalization intensity means many hospitalizations per 100,000 population are predicted. These forecasts are made for the same task ID variables as the `quantile` forecasts of incident hospitalizations.

We show a representative example of the hospitalization intensity category forecasts in Table 7. Because these forecasts are PMF output type, the `output_type_id` column specifies the bin of hospitalization intensity and the `value` column provides the forecasted probability of hospitalization incidence being in that category. Values sum to 1 across bins. For the MOBS-GLEAM_FLUH and PSI-DICE models, incidence is forecasted to decrease over the horizon (Figure 2), and correspondingly, there is lower probability of "high" and "very high" hospitalization intensity for later horizons (Figure 3).

model_id	target	horizon	output_type	output_type_id	value
Flusight-baseline	wk flu hosp rate category	1	pmf	low	0.00
Flusight-baseline	wk flu hosp rate category	1	pmf	moderate	0.00
Flusight-baseline	wk flu hosp rate category	1	pmf	high	0.07

model_id	target	horizon	output_type	output_type_id	value
Flusight-baseline	wk flu hosp rate category	1	pmf	very high	0.92
MOBS-GLEAM_FLUH	wk flu hosp rate category	1	pmf	low	0.00
MOBS-GLEAM_FLUH	wk flu hosp rate category	1	pmf	moderate	0.00
MOBS-GLEAM_FLUH	wk flu hosp rate category	1	pmf	high	0.16
MOBS-GLEAM_FLUH	wk flu hosp rate category	1	pmf	very high	0.83
PSI-DICE	wk flu hosp rate category	1	pmf	low	0.01
PSI-DICE	wk flu hosp rate category	1	pmf	moderate	0.07
PSI-DICE	wk flu hosp rate category	1	pmf	high	0.22
PSI-DICE	wk flu hosp rate category	1	pmf	very high	0.70

Table 7: Example PMF model output for forecasts of incident influenza hospitalization intensity. A subset of example model output is shown: 1-week ahead PMF forecasts made on 2022-12-17 for Massachusetts from three distinct models. We round the forecasted probability (in the `value` column) to two digits. The `location`, `reference_date` and `target_end_date` columns have been omitted for brevity. This example data is provided in the **hubExamples** package.

5.2. Creating ensembles with simple_ensemble

Using the default options for `simple_ensemble()`, we can generate an equally weighted mean ensemble for each unique combination of values for the task ID variables, the `output_type` and the `output_type_id`. Recall that this means different ensemble methods will be used for different output types: for the `quantile` output type in our example data, the resulting ensemble is a quantile average, while for the PMF output type, the ensemble is a linear pool.

```
R> mean_ens <- hubEnsembles::simple_ensemble(
+  hubExamples::forecast_outputs,
+  model_id = "simple-ensemble-mean"
+)
```

The resulting model output has the same structure as the original model output data (Table 8), with columns for model ID, task ID variables, output type, output type ID, and value. We also use `model_id = "simple-ensemble-mean` to change the name of this ensemble in the resulting model output; if not specified, the default will be "hub-ensemble".

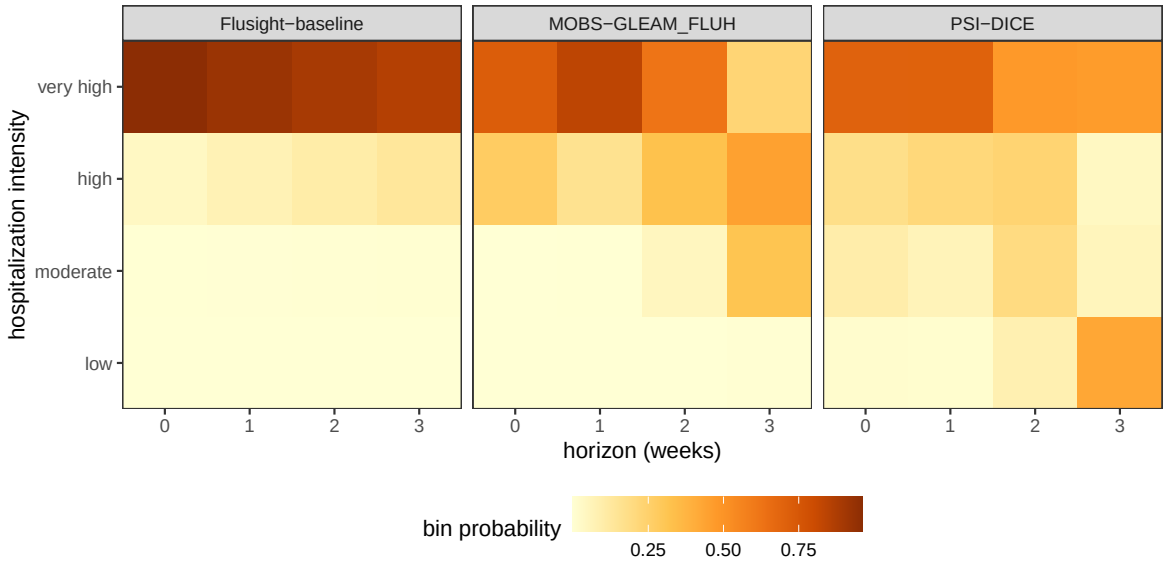

Figure 3: One example PMF forecast of incident influenza hospitalization intensity is shown for each of three models (panels). Each cell shows the forecasted probability of a given hospitalization intensity bin (low, moderate, high, and very high) for each forecast horizon (0-3 weeks ahead). Darker colors indicate higher forecasted probability.

`model_id`	`target`	`horizon`	`output_type`	`output_type_id`	`value`
simple-ensemble-mean	wk flu hosp rate category	1	pmf	high	0.15
simple-ensemble-mean	wk flu hosp rate category	1	pmf	low	0.00
simple-ensemble-mean	wk flu hosp rate category	1	pmf	moderate	0.02
simple-ensemble-mean	wk flu hosp rate category	1	pmf	very high	0.82
simple-ensemble-mean	wk inc flu hosp	1	median	NA	619.67
simple-ensemble-mean	wk inc flu hosp	1	quantile	0.25	541.67
simple-ensemble-mean	wk inc flu hosp	1	quantile	0.75	704.33

Table 8: Mean ensemble model output. The values in the `model_id` column are determined by the argument `simple_ensemble(..., model_id = )`. A subset of ensemble model output is shown: 1-week ahead PMF forecasts made on 2022-12-17 for Massachusetts. Results are generated for all output types. Here, we show only the median and 50% prediction interval for the quantile output type and all bins for the PMF output type. The `location`, `reference_date` and `target_end_date` columns have been omitted for brevity, and the `value` column is rounded to two digits.

Changing the aggregation function

We can change the function that is used to aggregate model outputs. For example, we may want to calculate a median of the component models' submitted values for each quantile. We do so by specifying `agg_fun = median`.

```
R> median_ens <- hubEnsembles::simple_ensemble(hubExamples::forecast_outputs,
+  agg_fun = median,
+  model_id = "simple-ensemble-median"
+)
```

Custom functions can also be passed into the `agg_fun` argument. We illustrate this by defining a custom function to compute the ensemble prediction as a geometric mean of the component model predictions. Any custom function to be used must have an argument `x` for the vector of numeric values to summarize, and if relevant, an argument `w` of numeric weights.

```
R> geometric_mean <- function(x) {
+  n <- length(x)
+  return(prod(x)^(1 / n))
+}
R> geometric_mean_ens <- hubExamples::forecast_outputs |>
+  hubEnsembles::simple_ensemble(
+    agg_fun = geometric_mean,
+    model_id = "simple-ensemble-geometric"
+  )
```

As expected, the mean, median, and geometric mean each give us slightly different resulting ensembles. The median point estimates, 50% prediction intervals, and 90% prediction intervals in Figure 4 demonstrate this.

Weighting model contributions

We can weight the contributions of each model in the ensemble using the `weights` argument of `simple_ensemble`. This argument takes a `data.frame` that should include a `model_id` column containing each unique model ID and a `weight` column. In the following example, we include the baseline model in the ensemble, but give it less weight than the other forecasts.

```
R> model_weights <- data.frame(
+  model_id = c("MOBS-GLEAM_FLUH", "PSI-DICE", "simple_hub-baseline"),
+  weight = c(0.4, 0.4, 0.2)
+)
R> weighted_mean_ens <- hubExamples::forecast_outputs |>
+  hubEnsembles::simple_ensemble(
+    weights = model_weights,
+    model_id = "simple-ensemble-weighted-mean"
+  )
```

5.3. Creating ensembles with linear_pool

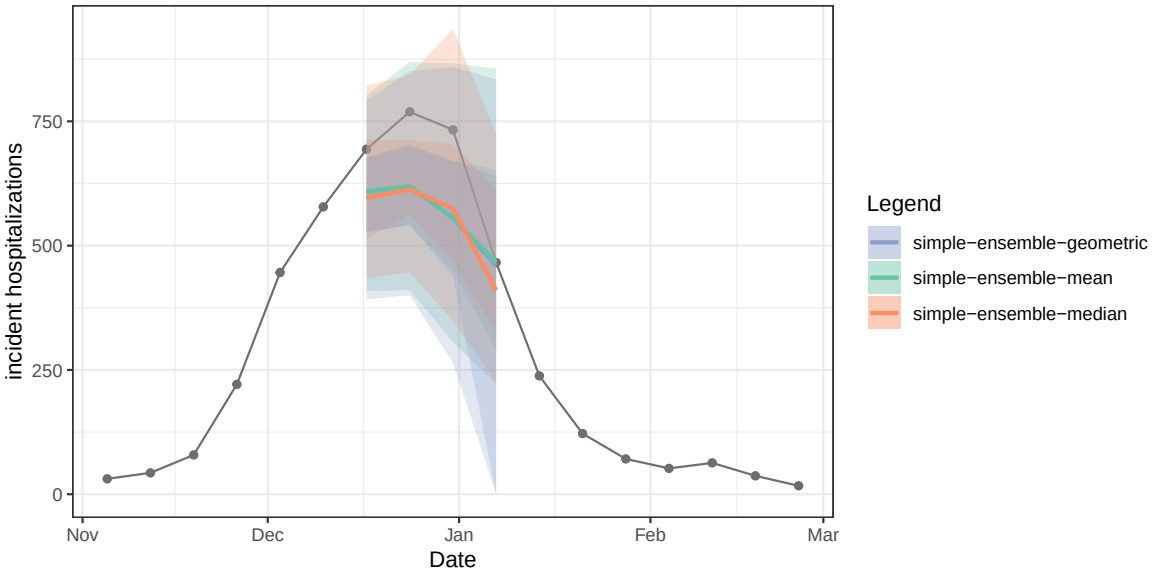

Figure 4: Three different ensembles for weekly incident influenza hospitalizations in Massachusetts. Each ensemble combines individual predictions from the example hub (Figure 2) using a different method: arithmetic mean, geometric mean, or median. All methods correspond to variations of the quantile average approach.

We can also generate a linear pool ensemble, or distributional mixture, using the `linear_pool()` function; this function can be applied to predictions with an `output_type` of mean, quantile, CDF, or PMF. Our example hub includes median output type, so we exclude it from the calculation.

```
R> linear_pool_ens <- hubExamples::forecast_outputs |>
+  dplyr::filter(output_type != "median") |>
+  hubEnsembles::linear_pool(model_id = "linear-pool")
```

As described above, for `quantile` model outputs, the `linear_pool` function approximates the full probability distribution for each component prediction using the value-quantile pairs provided by that model, and then obtains quasi-random samples from that distributional estimate. The number of samples drawn from the distribution of each component model defaults to `1e4`, but this can be changed using the `n_samples` argument.

In Figure 5, we compare ensemble results generated by `simple_ensemble` and `linear_pool` for model outputs of output types PMF and quantile. As expected, the results from the two functions are equivalent for the PMF output type: for this output type, the `simple_ensemble` method averages the predicted probability of each category across the component models, which is the definition of the linear pool ensemble method. This is not the case for the quantile output type, because the `simple_ensemble` is computing a quantile average.

6. Example: in-depth analysis of forecast data

To further demonstrate the utility of the **hubEnsembles** package and the differences between

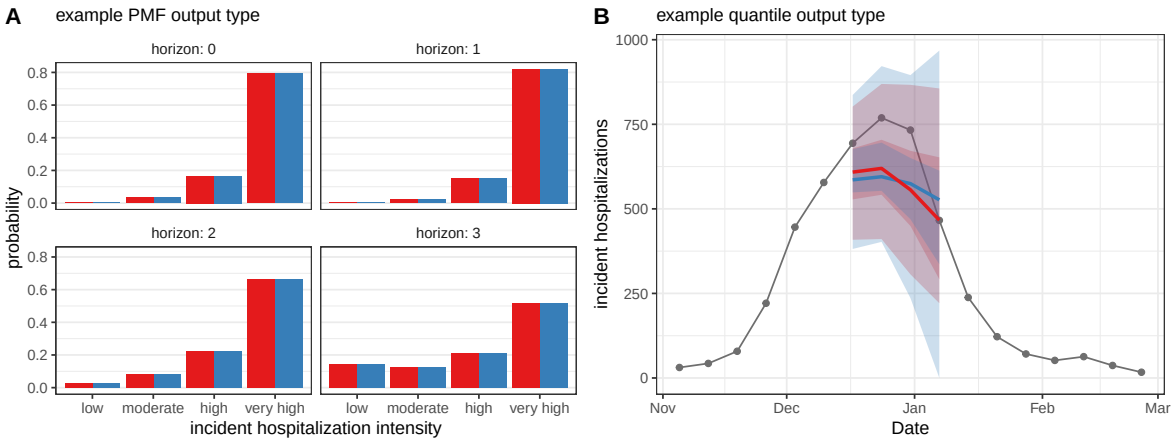

Figure 5: Comparison of results from `simple_ensemble` (blue) and `linear pool` (red). (Panel A) Ensemble predictions of Massachusetts incident influenza hospitalization intensity (classified as low, moderate, high, or very high), which provide an example of PMF output type. (Panel B) Ensemble predictions of weekly incident influenza hospitalizations in Massachusetts, which provide an example of quantile output type. Note, for quantile output type, `simple_ensemble` corresponds to a quantile average. Ensembles combine individual models from the example hub (Figure 2).

the two ensembling functions, we examine a more complex example. Unlike the previous section's basic showcase of functionality, we use this case study to provide a more complete analysis that compares and evaluates ensemble model performance using real forecasts collected by a modeling hub, with an overarching goal of choosing a single best ensembling approach for the application.

Since 2013, the US Centers for Disease Control and Prevention (CDC) has been soliciting forecasts of seasonal influenza from modeling teams through a collaborative challenge called FluSight (CDC 2023). We use a subset of these predictions to create four equally-weighted ensembles with `simple_ensemble()` and `linear_pool()` and compare the resulting ensembles' performance. The ensembling methods chosen for this case study consist of a quantile (arithmetic) mean, a quantile median, a linear pool with normal tails, and a linear pool with lognormal tails. Note that only a select portion of the code is shown in this manuscript for brevity, although all the functions and scripts used to generate the case study results can be found in the associated GitHub repository (https://github.com/Infectious-Disease-Modeling-Hubs/hubEnsemblesManuscript).

6.1. Data and Methods

We begin by querying the component forecasts used to generate the four ensembles from Zoltar (Reich *et al.* 2021), a repository designed to archive forecasts created by the Reich Lab at UMass Amherst. Here, we only consider Flusight predictions in a quantile format from the 2021-2022 and 2022-2023 seasons. These forecasts were stored in two data objects, split by season, called `flu_forecasts-raw_21-22.rds` and `flu_forecasts-raw_22-23.rds`. Forecasts must conform to hubverse standards to be fed into either of the ensembling functions,

so we first transform the raw forecasts using `as_model_out_tbl` function from the **hubUtils** package. Here, we specify the task ID variables `forecast date` (when the forecast was made), `location`, `horizon`, and `target`.

```
R> flu_forecasts_raw_21_22 <- readr::read_rds(
+  here::here("analysis/data/raw_data/flu_forecasts-raw_21-22.rds")
+)
R> flu_forecasts_raw_22_23 <- readr::read_rds(
+  here::here("analysis/data/raw_data/flu_forecasts-raw_22-23.rds")
+)
R> flu_forecasts_raw <- rbind(flu_forecasts_raw_21_22,
+                           dplyr::select(flu_forecasts_raw_22_23, -season))
R> flu_forecasts_hubverse <- flu_forecasts_raw |>
+  dplyr::rename(target = target_long) |>
+  as_model_out_tbl(
+    model_id_col = "model_id",
+    output_type_col = "output_type",
+    output_type_id_col = "output_type_id",
+    value_col = "value",
+    sep = "-",
+    trim_to_task_ids = FALSE,
+    hub_con = NULL,
+    task_id_cols = c("forecast_date", "location", "horizon", "target"),
+    remove_empty = TRUE
+  )
```

We excluded component forecasts from the ensemble calculations if any prediction (defined by a unique combination of task ID variables) did not include all 23 quantiles specified by FluSight ($\theta \in \{.010, 0.025, .050, .100, ..., .900, .950, .990\}$). We also excluded the FluSight baseline and median ensemble models that were previously generated by the FluSight hub.

With these inclusion criteria, the final data set of component forecasts consists of predictions from 25 modeling teams and 42 distinct models, 53 forecast dates (one per week), 54 US locations, 4 horizons, 1 target, and 23 quantiles. In the 2021-2022 season, 25 models made predictions for 22 weeks spanning from late January 2022 to late June 2022, and in the 2022-2023 season, there were 31 models making predictions for 31 weeks spanning mid-October 2022 to mid-May 2023. Fourteen of the 42 total models made forecasts for both seasons.

In both seasons, forecasts were made for the same locations (the 50 US states, Washington DC, Puerto Rico, the Virgin Islands, and the US as a whole), horizons (1 to 4 weeks ahead), quantiles (the 23 described above), and target (week ahead incident flu hospitalization). The values for the forecasts are always non-negative. In Table 9, we provide an example of these predictions, showing select quantiles from a single model, forecast date, horizon, and location.

model_id	target	horizon	output_type	output_type_id	value
UMass-trends_ensemble	wk ahead inc flu hosp	1	quantile	0.025	12

model_id	target	horizon	output_type	output_type_id	value
UMass-trends_ensemble	wk ahead inc flu hosp	1	quantile	0.100	17
UMass-trends_ensemble	wk ahead inc flu hosp	1	quantile	0.250	25
UMass-trends_ensemble	wk ahead inc flu hosp	1	quantile	0.750	46
UMass-trends_ensemble	wk ahead inc flu hosp	1	quantile	0.900	56
UMass-trends_ensemble	wk ahead inc flu hosp	1	quantile	0.975	68

Table 9: An example prediction of weekly incident influenza hospitalizations. The example model output was made on May 15, 2023 for California at the 1 week ahead horizon. The forecast was generated during the FluSight forecasting challenge, then formatted according to hubverse standards post hoc. The `location`, `forecast_date`, and `season` columns have been omitted for brevity.

Next, we combine the component model outputs to generate predictions from each ensemble model. We begin by excluding the baseline model from the set of predictions that will be combined. Then, we create one object to store the ensemble results generated from each method we are interested in comparing.

```
R> flu_forecasts_hubverse <- dplyr::filter(
+  flu_forecasts_hubverse,
+  model_id != "Flusight-baseline"
+)
R> mean_ensemble <- flu_forecasts_hubverse |>
+  hubEnsembles::simple_ensemble(
+  weights = NULL,
+  agg_fun = "mean",
+  model_id = "mean-ensemble"
+)
R> median_ensemble <- flu_forecasts_hubverse |>
+  hubEnsembles::simple_ensemble(
+  weights = NULL,
+  agg_fun = "median",
+  model_id = "median-ensemble"
+)
R> lp_normal <- flu_forecasts_hubverse |>
+  hubEnsembles::linear_pool(
+  weights = NULL,
+  n_samples = 1e5, model_id = "lp-normal",
+  tail_dist = "norm"
+)
R> lp_lognormal <- flu_forecasts_hubverse |>
```

```
+  hubEnsembles::linear_pool(
+  weights = NULL,
+  n_samples = 1e5,
+  model_id = "lp-lognormal",
+  tail_dist = "lnorm"
+)
```

We evaluate the performance of these ensembles using scoring metrics that measure the accuracy and calibration of their forecasts. Here, we use several common metrics in forecast evaluation, including mean absolute error (MAE), weighted interval score (WIS) (Bracher *et al.* 2021), 50% prediction interval (PI) coverage, and 95% PI coverage. MAE measures the average absolute error of a set of point forecasts; smaller values of MAE indicate better forecast accuracy. WIS is a generalization of MAE for probabilistic forecasts and is an alternative to other common proper scoring rules which cannot be evaluated directly for quantile forecasts (Bracher *et al.* 2021). WIS is made up of three component penalties: (1) for over-prediction, (2) for under-prediction, and (3) for the spread of each interval (where an interval is defined by a symmetric set of two quantiles). This metric weights these penalties across all prediction intervals provided. A lower WIS value indicates a more accurate forecast (Bracher *et al.* 2021). PI coverage provides information about whether a forecast has accurately characterized its uncertainty about future observations. The 50% PI coverage rate measures the proportion of the time that 50% prediction intervals at that nominal level included the observed value; the 95% PI coverage rate is defined similarly. Achieving approximately nominal (50% or 95%) coverage indicates a well-calibrated forecast.

We also use relative versions of WIS and MAE (rWIS and rMAE, respectively) to understand how the ensemble performance compares to that of the FluSight baseline model. These metrics are calculated as

$$\text{rWIS} = \frac{\text{WIS}_{\text{model } m}}{\text{WIS}_{\text{baseline}}} \qquad \text{rMAE} = \frac{\text{MAE}_{\text{model } m}}{\text{MAE}_{\text{baseline}}},$$

where model m is any given model being compared against the baseline. For both of these metrics, a value less than one indicates better performance compared to the baseline while a value greater than one indicates worse performance. By definition, the FluSight baseline itself will always have a value of one for both of these metrics.

Here, we score each unique prediction from an ensemble model, i.e., combination of task ID variables, against target data with the `score_forecasts()` function from the **covidHubUtils** package. This function outputs each of the metrics described above. We use median forecasts taken from the 0.5 quantile for the MAE evaluation.

6.2. Performance results across ensembles

The quantile median ensemble had the best overall performance in terms of WIS and MAE (and the relative versions of these metrics), and had coverage rates that were close to the nominal levels (Table 10). The two linear opinion pools had very similar performance to each other. These methods had the second-best performance as measured by WIS and MAE, but they had the highest 50% and 95% coverage rates, with empirical coverage that was well

above the nominal coverage rate. The quantile mean performed the worst of the ensembles with the highest MAE, which was substantially different from that of the other ensembles.

model	wis	rwis	mae	rmae	cov50	cov95
median-ensemble	**18.158**	**0.794**	**27.36**	**0.933**	0.597	**0.922**
lp-normal	19.745	0.863	27.932	0.953	0.709	0.99
lp-lognormal	19.747	0.863	27.933	0.953	0.708	0.99
mean-ensemble	20.18	0.882	29.582	1.009	**0.595**	0.889
Flusight-baseline	22.876	1	29.315	1	0.604	0.881

Table 10: Summary of overall model performance across both seasons, averaged over all locations except the US national location. The best values for each metric is bolded, though the metric values are often quite similar among the models.

Plots of the models' forecasts can aid our understanding about the origin of these accuracy differences. For example, the linear opinion pools consistently had some of the widest prediction intervals, and consequently the highest coverage rates. The median ensemble, which had the best WIS, balanced interval width with calibration best overall, with narrower intervals than the linear pools that still achieved near-nominal coverage on average across all time points. The quantile mean's interval widths varied, though it usually had narrower intervals than the linear pools. However, this model's point forecasts had a larger error margin compared to the other ensembles, especially at longer horizons. This pattern is demonstrated in Figure 6 for the 4-week ahead forecast in California following the 2022-23 season peak on December 5, 2022. Here the quantile mean predicted a continued increase in hospitalizations, at a steeper slope than the other ensemble methods.

Generally, the median ensemble outperforms the other models regardless of scoring metric. The median model has lower MAE and WIS values than the mean model during similar periods of time, which are localized times of change. The median model also had better coverage rates than the mean model in the tells the distribution (95% intervals) but similar coverage rates in the center of the distribution (50% intervals). All of the ensemble models tend to have similar MAE values during the entire evaluation time period, with slight divergence in MAE values for certain weeks at the four week ahead horizon (Figure 7). However, the models show greater differences for the other two metrics, WIS and coverage, particularly during times of rapid change in the observed incident hospitalizations (Figure 8 and Figure 9). The linear pools have a lower WIS than the median ensemble at the one week ahead forecast horizon for over a third of forecast dates (11 weeks) during the 2022-2023 season: from October 17, 2022 to December 12, 2022; January 2, 2023; and January 9, 2023 (Figure 8). These dates span the rapid rise and fall of incident flu hospitalizations surrounding the season's peak, with the largest differences in WIS occurring on November 28, December 5, December 12, January 2, and January 9. Additionally, the PI coverage rates for the linear pools were at least as large as the coverage rates of the other models throughout the entire period of analysis at both the 1 and 4 week ahead forecast horizons (see Figure 9).

In this analysis, all of the ensemble variations outperformed the baseline model; yet, different ensembling methods performed best under different circumstances. While the quantile median

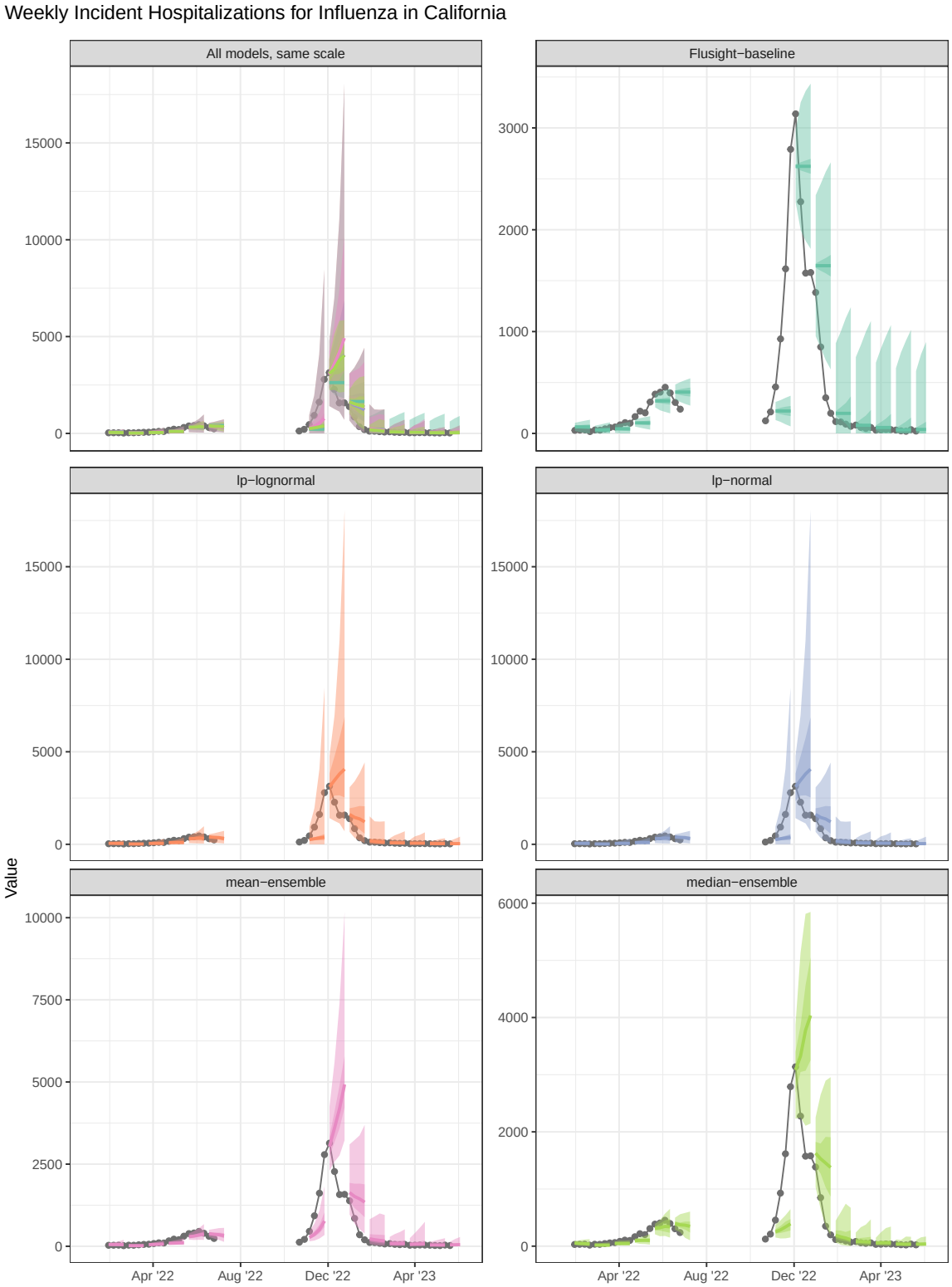

Figure 6: One to four week ahead forecasts for select dates plotted against target data for California. The first panel shows all models on the same scale. All other panels show forecasts for each individual model, with varying y-axis scales, and their prediction accuracy as compared to observed influenza hospitalizations.

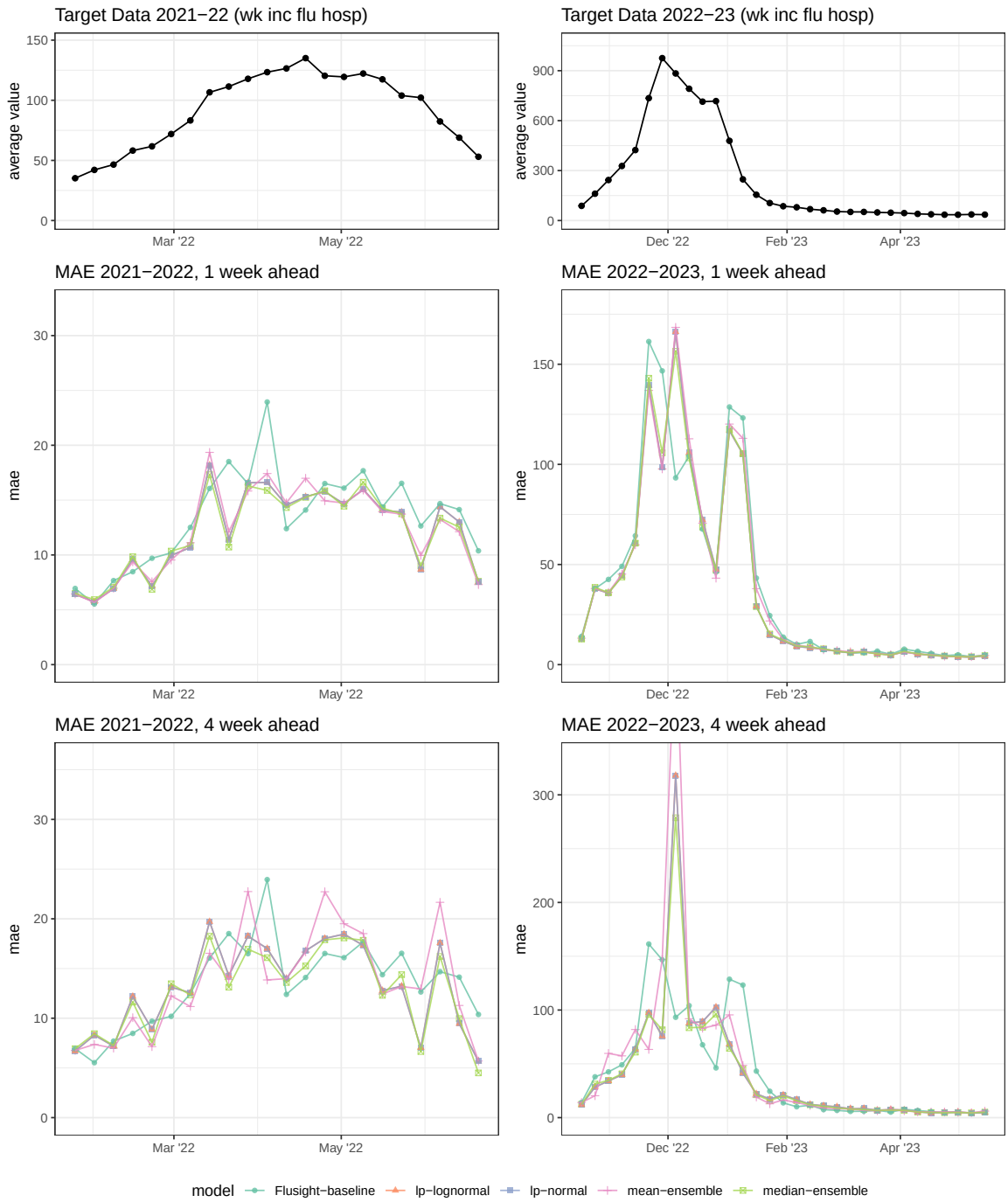

Figure 7: Mean absolute error (MAE) averaged across all locations. Average MAE is shown for each season (columns) and for 1-week and 4-week ahead forecasts (rows). Results are plotted for each ensemble model (colored points) across the entire season. Average target data across all locations is plotted in black.

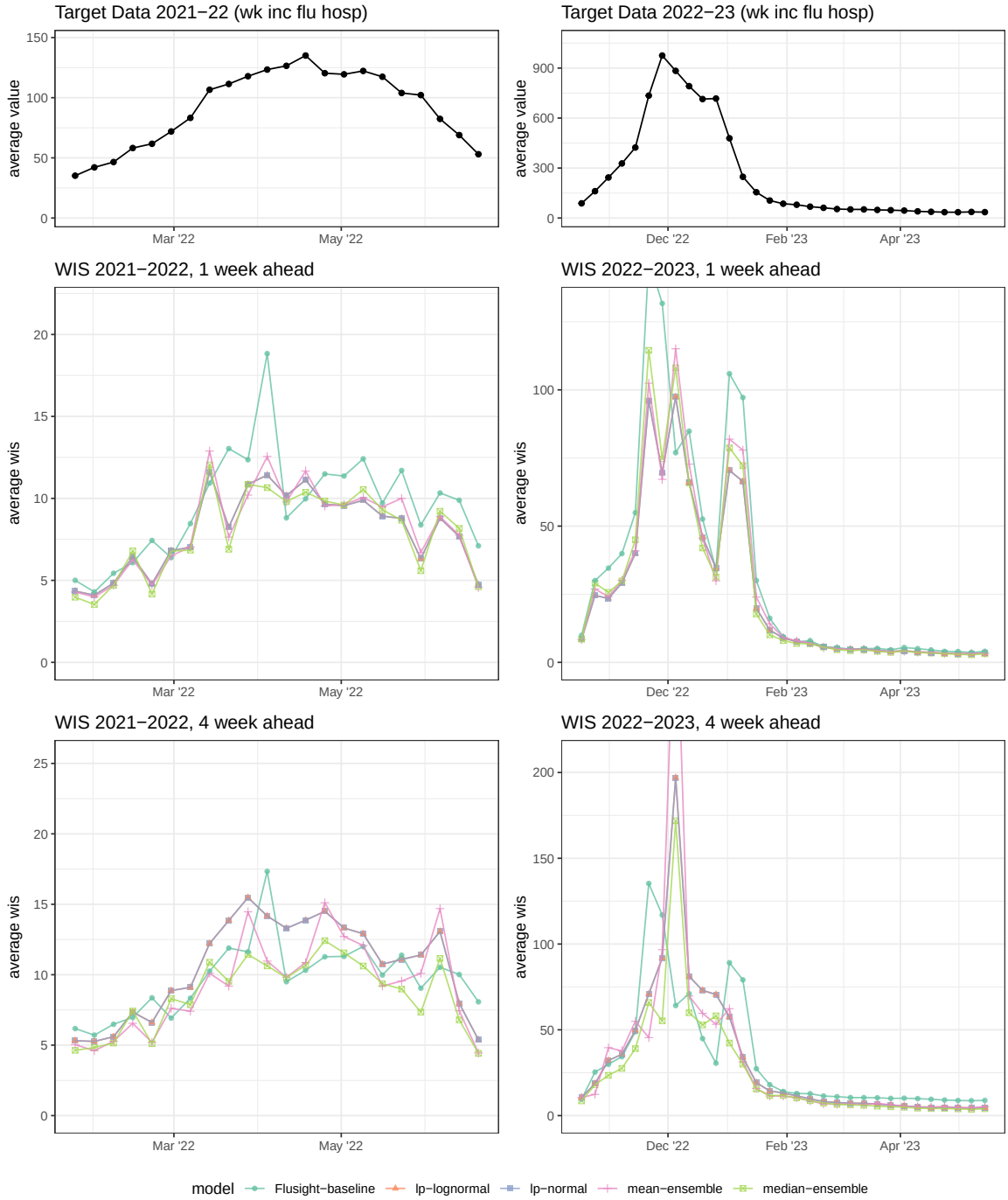

Figure 8: Weighted interval score (WIS) averaged across all locations. Average WIS is shown for each season (columns) and for 1-week and 4-week ahead forecasts (rows). Results are plotted for each ensemble model (colored points) across the entire season. Average target data across all locations is plotted in black.

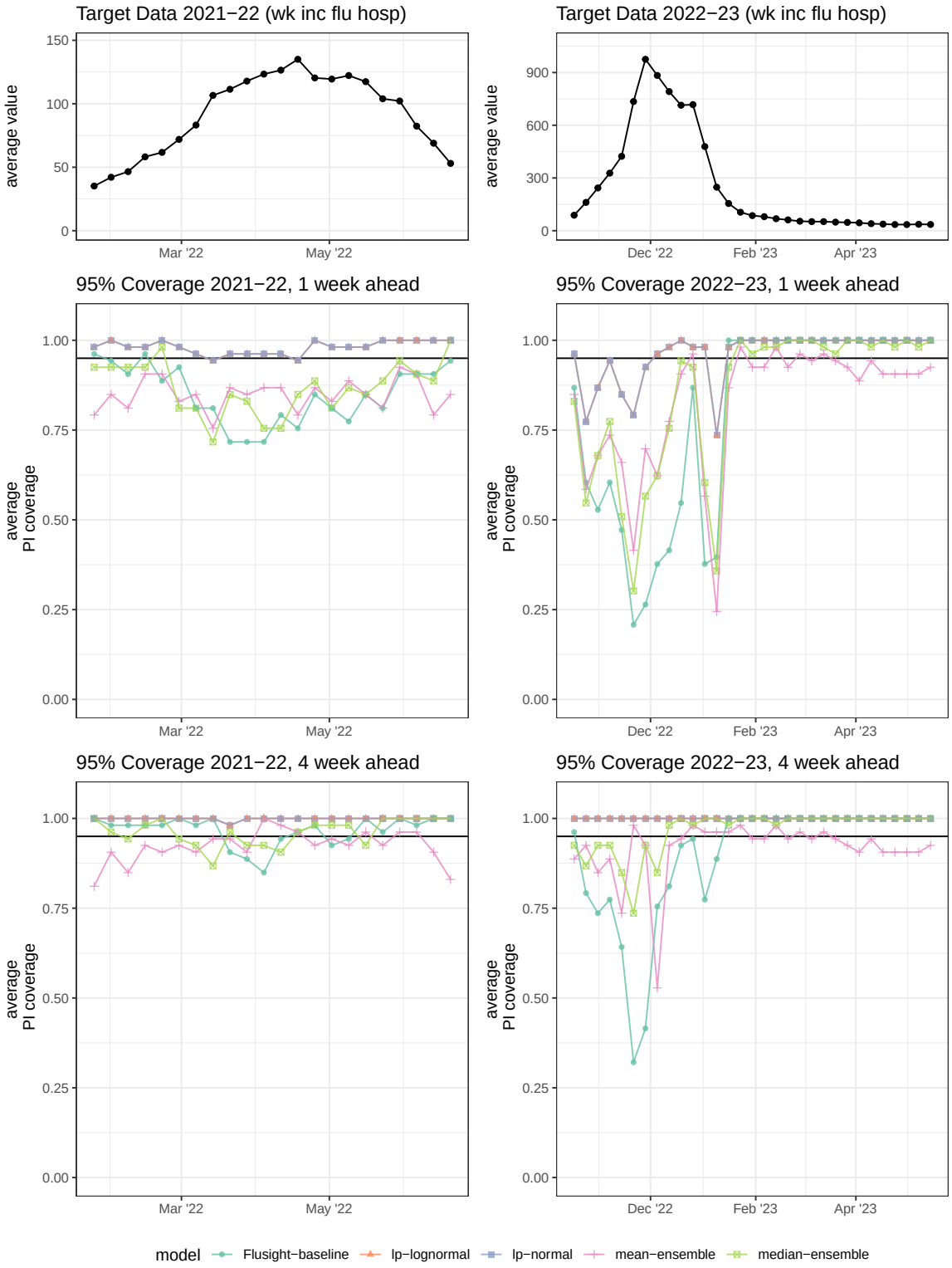

Figure 9: 95% prediction interval (PI) coverage averaged across all locations. Average coverage is shown for each season (columns) and for 1-week and 4-week ahead forecasts (rows). Results are plotted for each ensemble model (colored points) across the entire season. Average target data across all locations is plotted in black.

had the best overall results for WIS, MAE, 50% PI coverage, and 95% PI coverage, other models may perform better from week-to-week for each metric. Around the 2022-2023 season's peak in early December, the remaining four models (including the baseline) each had instances in which they achieved the lowest WIS, like the linear pool ensembles for the one week ahead horizon over several weeks of this period.

The choice of an appropriate ensemble aggregation method may depend on the forecast target, the goal of forecasting, and the behavior of the individual models contributing to an ensemble. One case may call for prioritizing high coverage rates while another may prioritize accurate point forecasts. The `simple_ensemble` and `linear_pool` functions and the ability to specify component model weights and an aggregation function for `simple_ensemble` allow users to implement a variety of ensemble methods.

7. Summary and discussion

Ensembles of independent models are a powerful tool to generate more accurate and more reliable predictions of future outcomes than a single model alone. Here, we have demonstrated how to utilize **hubEnsembles**, a simple and flexible framework to combine individual model predictions into an ensemble.

The **hubEnsembles** package is situated within the larger hubverse collection of open-source software and data tools to support collaborative modeling exercises. Collaborative hubs offer many benefits, including serving as a centralized entity to guide and elicit predictions from multiple independent models (Reich *et al.* 2022). Given the increasing popularity of multi-model ensembles and collaborative hubs, there is a clear need for generalized data standards and software infrastructure to support these hubs. By addressing this need, the hubverse suite of tools can reduce duplicative efforts across existing hubs, support other communities engaged in collaborative efforts, and enable the adoption of multi-model approaches in new domains.

When using **hubEnsembles**, it is important to carefully choose an ensemble method that is well suited for the situation. Although there may not be a universal "best" method, matching the properties of a given ensemble method with the features of the component models will likely yield best results (Howerton *et al.* 2023). Our case study on seasonal influenza forecasts in the US demonstrates this point. The quantile median ensemble performed best overall for a range of metrics, including weighted interval score, mean absolute error, and prediction interval coverage. Yet, the linear pool method, which generates an ensemble with wider prediction intervals, demonstrated performance advantages during periods of rapid change, when outlying component forecasts were likely more important. Notably, all ensemble methods outperformed the baseline model. The performance improvements from ensemble models motivate the use of a "hub-based" approach to prediction for infectious diseases and in other fields.

Ongoing development of the **hubEnsembles** package and the larger suite of hubverse tools will continue to support multi-model predictions in new ways, including for example supporting additional types of predictions and enabling cloud-based data storage. All such infrastructure will ultimately provide a comprehensive suite of open-source software tools for leveraging the power of collaborative hubs and multi-model ensembles.

Acknowledgements

The authors thank all members of the hubverse community; the broader hubverse software infrastructure made this package possible. E. Howerton was supported by the Eberly College of Science Barbara McClintock Science Achievement Graduate Scholarship in Biology at the Pennsylvania State University. L. Contamin and H. Hochheiser were supported by NIGMS grant U24GM132013. PLEASE ADD FUNDING HERE.

Consortium of Infectious Disease Modeling Hubs

Consortium of Infectious Disease Modeling Hubs authors include Alvaro J. Castro Rivadeneira (University of Massachusetts Amherst), Lucie Contamin (University of Pittsburgh), Sebastian Funk (London School of Hygene and Tropical Medicine), Aaron Gerding (University of Massachusetts Amherst), Hugo Gruson (ADD AFFILIATION), Harry Hochheiser (University of Pittsburgh), Emily Howerton (The Pennsylvania State University), Melissa Kerr (University of Massachusetts Amherst), Anna Krystalli (R-RSE SMPC), Sara L. Loo (Johns Hopkins University), Evan L. Ray (University of Massachusetts Amherst), Nicholas G. Reich (University of Massachusetts Amherst), Koji Sato, Li Shandross (University of Massachusetts Amherst), Katharine Sherratt (London School of Hygene and Tropical Medicine), Shaun Truelove (Johns Hopkins University), Martha Zorn (University of Massachusetts Amherst)

References

Aastveit KA, Mitchell J, Ravazzolo F, van Dijk HK (2018). "The Evolution of Forecast Density Combinations in Economics." *Tinbergen Institute Discussion Papers*. URL https://hdl.handle.net/10419/185588.

Alley RB, Emanuel KA, Zhang F (2019). "Advances in weather prediction." *Science*, **363**(6425), 342–344. doi:10.1126/science.aav7274.

Borchering RK, Healy JM, Cadwell BL, Johansson MA, Slayton RB, Wallace M, Biggerstaff M (2023). "Public health impact of the U.S. Scenario Modeling Hub." *Epidemics*, **44**, 100705. doi:10.1016/j.epidem.2023.100705.

Bosse N, Yao Y, Abbott S, Funk S (2023). *stackr: Create Mixture Models From Predictive Samples*. R package version 0.1.0, https://github.com/epiforecasts/stackr, Accessed: 2024-01-05.

Bracher J, Ray EL, Gneiting T, Reich NG (2021). "Evaluating epidemic forecasts in an interval format." *PLOS Computational Biology*, **17**(2), e1008618. doi:10.1371/journal.pcbi.1008618.

CDC (2023). "About Flu Forecasting." Accessed: 2024-05-02, URL https://www.cdc.gov/flu/weekly/flusight/how-flu-forecasting.htm.

Clemen RT (1989). "Combining forecasts: A review and annotated bibliography." *International Journal of Forecasting*, **5**(4), 559–583. doi:10.1016/0169-2070(89)90012-5.

Colón-González FJ, Bastos LS, Hofmann B, Hopkin A, Harpham Q, Crocker T, Amato R, Ferrario I, Moschini F, James S, Malde S, Ainscoe E, Nam VS, Tan DQ, Khoa ND, Harrison M, Tsarouchi G, Lumbroso D, Brady OJ, Lowe R (2021). "Probabilistic seasonal dengue forecasting in Vietnam: A modelling study using superensembles." *PLOS Medicine*, **18**(3), e1003542. doi:10.1371/journal.pmed.1003542.

Consortium of Infectious Disease Modeling Hubs (2024). "The hubverse: open tools for collaborative forecasting." Accessed: 2024-05-02, URL https://hubdocs.readthedocs.io/en/latest/index.html.

Couch S, Kuhn M (2023). *stacks: Tidy Model Stacking*. R package version 1.0.3, https://github.com/tidymodels/stacks, Accessed: 2024-01-05.

Cramer EY, Ray EL, Lopez VK, Bracher J, Brennen A, Castro Rivadeneira AJ, Gerding A, Gneiting T, House KH, Huang Y, Jayawardena D, Kanji AH, Khandelwal A, Le K, Mühlemann A, Niemi J, Shah A, Stark A, Wang Y, Wattanachit N, Zorn MW, Gu Y, Jain S, Bannur N, Deva A, Kulkarni M, Merugu S, Raval A, Shingi S, Tiwari A, White J, Abernethy NF, Woody S, Dahan M, Fox S, Gaither K, Lachmann M, Meyers LA, Scott JG, Tec M, Srivastava A, George GE, Cegan JC, Dettwiller ID, England WP, Farthing MW, Hunter RH, Lafferty B, Linkov I, Mayo ML, Parno MD, Rowland MA, Trump BD, Zhang-James Y, Chen S, Faraone SV, Hess J, Morley CP, Salekin A, Wang D, Corsetti SM, Baer TM, Eisenberg MC, Falb K, Huang Y, Martin ET, McCauley E, Myers RL, Schwarz T, Sheldon D, Gibson GC, Yu R, Gao L, Ma Y, Wu D, Yan X, Jin X, Wang YX, Chen Y, Guo L, Zhao Y, Gu Q, Chen J, Wang L, Xu P, Zhang W, Zou D, Biegel H, Lega J, McConnell S, Nagraj VP, Guertin SL, Hulme-Lowe C, Turner SD, Shi Y, Ban X, Walraven R, Hong QJ, Kong S, van de Walle A, Turtle JA, Ben-Nun M, Riley S, Riley P, Koyluoglu U, DesRoches D, Forli P, Hamory B, Kyriakides C, Leis H, Milliken J, Moloney M, Morgan J, Nirgudkar N, Ozcan G, Piwonka N, Ravi M, Schrader C, Shakhnovich E, Siegel D, Spatz R, Stiefeling C, Wilkinson B, Wong A, Cavany S, España G, Moore S, Oidtman R, Perkins A, Kraus D, Kraus A, Gao Z, Bian J, Cao W, Lavista Ferres J, Li C, Liu TY, Xie X, Zhang S, Zheng S, Vespignani A, Chinazzi M, Davis JT, Mu K, Pastore y Piontti A, Xiong X, Zheng A, Baek J, Farias V, Georgescu A, Levi R, Sinha D, Wilde J, Perakis G, Bennouna MA, Nze-Ndong D, Singhvi D, Spantidakis I, Thayaparan L, Tsiourvas A, Sarker A, Jadbabaie A, Shah D, Della Penna N, Celi LA, Sundar S, Wolfinger R, Osthus D, Castro L, Fairchild G, Michaud I, Karlen D, Kinsey M, Mullany LC, Rainwater-Lovett K, Shin L, Tallaksen K, Wilson S, Lee EC, Dent J, Grantz KH, Hill AL, Kaminsky J, Kaminsky K, Keegan LT, Lauer SA, Lemaitre JC, Lessler J, Meredith HR, Perez-Saez J, Shah S, Smith CP, Truelove SA, Wills J, Marshall M, Gardner L, Nixon K, Burant JC, Wang L, Gao L, Gu Z, Kim M, Li X, Wang G, Wang Y, Yu S, Reiner RC, Barber R, Gakidou E, Hay SI, Lim S, Murray C, Pigott D, Gurung HL, Baccam P, Stage SA, Suchoski BT, Prakash BA, Adhikari B, Cui J, Rodríguez A, Tabassum A, Xie J, Keskinocak P, Asplund J, Baxter A, Oruc BE, Serban N, Arik SO, Dusenberry M, Epshteyn A, Kanal E, Le LT, Li CL, Pfister T, Sava D, Sinha R, Tsai T, Yoder N, Yoon J, Zhang L, Abbott S, Bosse NI, Funk S, Hellewell J, Meakin SR, Sherratt K, Zhou M, Kalantari R, Yamana TK, Pei S, Shaman J, Li ML, Bertsimas D, Skali Lami O, Soni S, Tazi Bouardi H, Ayer T, Adee M, Chhatwal J, Dalgic OO, Ladd MA, Linas BP, Mueller P, Xiao J, Wang Y, Wang Q, Xie S, Zeng D, Green A, Bien J, Brooks L, Hu AJ, Jahja M, McDonald D, Narasimhan B, Politsch C, Rajanala S, Rumack A, Simon N, Tibshirani RJ, Tibshirani R, Ventura V, Wasserman L, O'Dea EB, Drake JM, Pagano R,

Tran QT, Ho LST, Huynh H, Walker JW, Slayton RB, Johansson MA, Biggerstaff M, Reich NG (2022). "Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States." *Proceedings of the National Academy of Sciences*, **119**(15), e2113561119. doi:10.1073/pnas.2113561119.

Hibon M, Evgeniou T (2005). "To combine or not to combine: selecting among forecasts and their combinations." *International Journal of Forecasting*, **21**(1), 15–24. doi:10.1016/j.ijforecast.2004.05.002.

Howerton E, Runge MC, Bogich TL, Borchering RK, Inamine H, Lessler J, Mullany LC, Probert WJM, Smith CP, Truelove S, Viboud C, Shea K (2023). "Context-dependent representation of within- and between-model uncertainty: aggregating probabilistic predictions in infectious disease epidemiology." *Journal of The Royal Society Interface*, **20**(198), 20220659. doi:10.1098/rsif.2022.0659.

Johansson MA, Apfeldorf KM, Dobson S, Devita J, Buczak AL, Baugher B, Moniz LJ, Bagley T, Babin SM, Guven E, Yamana TK, Shaman J, Moschou T, Lothian N, Lane A, Osborne G, Jiang G, Brooks LC, Farrow DC, Hyun S, Tibshirani RJ, Rosenfeld R, Lessler J, Reich NG, Cummings DAT, Lauer SA, Moore SM, Clapham HE, Lowe R, Bailey TC, García-Díez M, Carvalho M, Rodó X, Sardar T, Paul R, Ray EL, Sakrejda K, Brown AC, Meng X, Osoba O, Vardavas R, Manheim D, Moore M, Rao DM, Porco TC, Ackley S, Liu F, Worden L, Convertino M, Liu Y, Reddy A, Ortiz E, Rivero J, Brito H, Juarrero A, Johnson LR, Gramacy RB, Cohen JM, Mordecai EA, Murdock CC, Rohr JR, Ryan SJ, Stewart-Ibarra AM, Weikel DP, Jutla A, Khan R, Poultney M, Colwell RR, Rivera-García B, Barker CM, Bell JE, Biggerstaff M, Swerdlow D, Mier-Y-Teran-Romero L, Forshey BM, Trtanj J, Asher J, Clay M, Margolis HS, Hebbeler AM, George D, Chretien JP (2019). "An open challenge to advance probabilistic forecasting for dengue epidemics." *Proceedings of the National Academy of Sciences*, **116**(48), 24268–24274. doi:10.1073/pnas.1909865116.

Lichtendahl KC, Grushka-Cockayne Y, Winkler RL (2013). "Is It Better to Average Probabilities or Quantiles?" *Management Science*, **59**(7), 1594–1611. doi:10.1287/mnsc.1120.1667.

McGowan CJ, Biggerstaff M, Johansson M, Apfeldorf KM, Ben-Nun M, Brooks L, Convertino M, Erraguntla M, Farrow DC, Freeze J, Ghosh S, Hyun S, Kandula S, Lega J, Liu Y, Michaud N, Morita H, Niemi J, Ramakrishnan N, Ray EL, Reich NG, Riley P, Shaman J, Tibshirani R, Vespignani A, Zhang Q, Reed C (2019). "Collaborative efforts to forecast seasonal influenza in the United States, 2015–2016." *Scientific Reports*, **9**(1), 683. doi:10.1038/s41598-018-36361-9.

Niederreiter H (1992). *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics, Philadelphia PA.

Paireau J, Andronico A, Hozé N, Layan M, Crépey P, Roumagnac A, Lavielle M, Boëlle PY, Cauchemez S (2022). "An ensemble model based on early predictors to forecast COVID-19 health care demand in France." *Proceedings of the National Academy of Sciences*, **119**(18), e2103302119. doi:10.1073/pnas.2103302119.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot

M, Duchesnay (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, **12**(85), 2825–2830. doi:10.5555/1953048.2078195.

Ray EL, Brooks LC, Bien J, Biggerstaff M, Bosse NI, Bracher J, Cramer EY, Funk S, Gerding A, Johansson MA, Rumack A, Wang Y, Zorn M, Tibshirani RJ, Reich NG (2023). "Comparing trained and untrained probabilistic ensemble forecasts of COVID-19 cases and deaths in the United States." **39**(3), 1366–1383. doi:10.1016/j.ijforecast.2022.06.005.

Ray EL, Gerding A (2024). *distfromq: Reconstruct a Distribution from a Collection of Quantiles*. R package version 1.0.3, http://github.com/reichlab/distfromq, Accessed: 2024-01-08.

Ray EL, Reich NG (2018). "Prediction of infectious disease epidemics via weighted density ensembles." *PLOS computational biology*, **14**(2), e1005910. doi:10.1371/journal.pcbi.1005910.

Reich NG, Cornell M, Ray EL, House K, Le K (2021). "The Zoltar forecast archive, a tool to standardize and store interdisciplinary prediction research." *Scientific Data*, **8**(1), 59. doi:10.1038/s41597-021-00839-5.

Reich NG, Lessler J, Funk S, Viboud C, Vespignani A, Tibshirani RJ, Shea K, Schienle M, Runge MC, Rosenfeld R, Ray EL, Niehus R, Johnson HC, Johansson MA, Hochheiser H, Gardner L, Bracher J, Borchering RK, Biggerstaff M (2022). "Collaborative Hubs: Making the Most of Predictive Epidemic Modeling." *American Journal of Public Health*, **112**(6), 839–842. doi:10.2105/AJPH.2022.306831.

Reich NG, McGowan CJ, Yamana TK, Tushar A, Ray EL, Osthus D, Kandula S, Brooks LC, Crawford-Crudell W, Gibson GC, Moore E, Silva R, Biggerstaff M, Johansson MA, Rosenfeld R, Shaman J (2019). "Accuracy of real-time multi-model ensemble forecasts for seasonal influenza in the U.S." *PLOS computational biology*, **15**(11), e1007486. doi:10.1371/journal.pcbi.1007486.

Stone M (1961). "The Opinion Pool." *The Annals of Mathematical Statistics*, **32**(4), 1339–1342.

Tebaldi C, Knutti R (2007). "The Use of the Multi-Model Ensemble in Probabilistic Climate Projections." *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, **365**(1857), 2053–2075. doi:10.1098/rsta.2007.2076.

Timmermann A (2006). *Chapter 4 Forecast Combinations*, volume 1, pp. 135–196. Elsevier. doi:10.1016/S1574-0706(05)01004-9.

Viboud C, Sun K, Gaffey R, Ajelli M, Fumanelli L, Merler S, Zhang Q, Chowell G, Simonsen L, Vespignani A (2018). "The RAPIDD ebola forecasting challenge: Synthesis and lessons learnt." *Epidemics*, **22**, 13–21. doi:10.1016/j.epidem.2017.08.002.

Vincent SB (1912). *The function of the vibrissae in the behavior of the white rat.* Ph.D. thesis, University of Chicago, Cambridge MA. OCLC: 17104960.

Weiss C, Raviv E, Roetzer G (2019). "Forecast Combinations in R using the ForecastComb Package." *The R Journal*, **10**(2), 262. doi:10.32614/RJ-2018-052.

Winkler RL (2015). "Equal Versus Differential Weighting in Combining Forecasts." *Risk Analysis*, **35**(1), 16–18. doi:10.1111/risa.12302.

Yamana TK, Kandula S, Shaman J (2016). "Superensemble forecasts of dengue outbreaks." *Journal of The Royal Society Interface*, **13**(123), 20160410. doi:10.1098/rsif.2016.0410.

Affiliation:

Li Shandross, contributed equally
E-mail: lshandross@umass.edu

Emily Howerton, contributed equally

Lucie Contamin

Harry Hochheiser

Anna Krystalli

Consortium of
Infectious Disease Modeling Hubs

Nicholas G. Reich

Evan L. Ray

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd