Predicting seasonal influenza hospitalization using an ensemble super learner: a simulation study

**Short Title:** Predicting influenza hospitalizations with a super learner

Jason R. Gantenberg,(1,2) Kevin W. McConeghy,(2,3) Jon Steingrimsson,(4) Chanelle J. Howe,(5) Robertus van Aalst,(6,7), Ayman Chit,(6), Andrew R. Zullo(1,2,3)

**Corresponding author:** Jason R. Gantenberg ([jrgant@brown.edu](mailto:jrgant@brown.edu))

1. Department of Epidemiology, Brown University School of Public Health, 121 S. Main St., Providence, RI, 02912
2. Department of Health Services, Policy and Practice, 121 S. Main St., Providence, RI, 02912
3. Providence VA Medical Center, 830 Chalkstone Ave., Providence, RI, 02908
4. Department of Biostatistics, Brown University School of Public Health, 121 S. Main St., Providence, RI 02912
5. Center for Epidemiology and Environmental Health, Brown University School of Public Health, 121 S. Main St., Providence, RI, 02912
6. Sanofi Pasteur, Swiftwater, PA 18370
7. Department of Health Sciences, University of Groningen, University Medical Center Groningen, Groningen, the Netherlands

# Abstract

Seasonal influenza has a substantial impact on human health in the United States and worldwide. Accurate forecasts of key features of influenza epidemics and health care utilization can inform public health response to outbreaks. To date, predicting seasonal influenza-related hospitalizations has less frequently been the focus of forecasting efforts. We conducted a simulation study to measure the performance of a machine-learning algorithm (super learner) to predict three features of a given seasonal influenza epidemic: peak hospitalization rate, the week during which the hospitalization rate peaks, and the cumulative hospitalizations. We fit a set of statistical models and the super learner to 15,000 simulated influenza hospitalization curves to generate weekly predictions for the targets of interest. In each week, we compared the performance of the ensemble super learner (a weighted combination of component model predictions), the discrete super learner (best-performing individual statistical model), and a prediction based on a simple mean of a given outcome (e.g., mean peak hospitalization rate) across all simulated hospitalization curves. We found that the general super learner algorithm—which subsumes both the selection of the discrete super learner and estimation of the ensemble super learner—was consistently capable of choosing individual statistical models that improved predictions of the three seasonal prediction targets over those of a simple mean prediction. However, the ensemble super learner consistently exhibited higher prediction error than the discrete super learners and often performed comparably to a simple mean prediction. Given that national-level influenza hospitalization forecasts based on empirical may rely on the small number of past flu seasons recorded in surveillance data, the ensembles’ underperformance in a large sample of hypothetical influenza seasons suggests future work should determine the conditions under which ensemble models can be expected not to underperform the best-performing component model.

# Author Summary

Influenza prediction is a still-maturing science, but notable improvements have been made over the past decade in predicting influenza-like activity. In addition, increasing focus on machine learning approaches has provided hope that such approaches may further improve predictive accuracy. Less work, however, has focused on predicting influenza-related hospitalizations, and we sought to investigate the potential for machine learning approaches to predict several features of influenza hospitalization dynamics that could help inform the allocation of public health efforts and resources during a flu season. We used a machine learning approach, called *super learner*, which is capable of taking a large number of different statistical models and either combining them into a single prediction (the ensemble prediction) or choosing the best-performing model. Our results suggest that ensemble super learner predictions may not perform as well as the best-performing statistical model under certain conditions. Future work should focus on investigating when ensemble predictions based on machine learning would be expected to improve predictive accuracy.

# Meta

**Target journal:**

* *PLoS Computational Biology*

**Section:**

* Epidemiology and Clinical/Translational Studies

**Potential editors:**

* Benjamin Althouse
* Miles Davenport
* Matthew Ferrari
* Roger Kouyos
* James Lloyd-Smith

**Potential reviewers:**

* Ryan J. Tibshirani (co-author on paper we use for curve simulation)
* Logan C. Brooks (co-author on paper we use for curve simulation)
* Roni Rosenfeld (co-author on paper we use for curve simulation)
* Sherri Rose (expert in machine/super learning)
* David A. Osthus (flu modeler whose work came highly recommended by Nick Reich)
* Samrachana Adhikari (biostatistician and former lab member in the Carnegie Mellon group [with RJ Tibshirani)
* Oleg Sofrygin (sl3 package author)
* Nima Hejazi (sl3 package author)
* Jeremy Coyle (sl3 package author)

# Introduction

Between 2010 and 2017, approximately 140,000–570,000 individuals were hospitalized and 12,000–51,000 died annually due to seasonal influenza in the United States [1]. The ability to predict the future burden of influenza-related hospitalizations during a given influenza season can help policymakers, public health officials, physicians, and other stakeholders to allocate resources appropriately and prepare for expected changes in hospitalization rates [2]. For example, forecasts could provide hospitals with enough time to shift inpatient hospital beds available for patients admitted with influenza-related illnesses **[cite]**.

While influenza forecasting is a still-maturing science [3,4], researchers have made considerable progress over the past decade in improving the quality of and capacity for forecasting influenza-like illness (ILI) [3]. Many of these improvements have been spurred by the FluSight forecasting competitions sponsored by the Centers for Disease Control and Prevention (CDC) since the 2013–14 influenza season [3]. Many different types of models have been used to generate forecasts, including statistical time series models [3,5], Bayesian methods [4,6], and agent-based models [4], among others. Ensemble methods have emerged as a particularly promising approach to improving further the accuracy and stability of epidemic predictions due to ensembles’ ability to combine predictions from multiple estimators in a principled manner [2,7,8].

Ensemble methods combine predictions generated by a set of component models [7,9–11]. In some cases, ensembles aggreggate component model predictions by weighting better predictions more highly in the final ensemble prediction [7,8], though other weighting criteria can be applied [8]. One compelling rationale for using ensemble predictions rests in their ability to borrow the strengths and avoid the weaknesses of any single class of models used as inputs into the ensemble [12]. This feature tends to lead not only to more accurate predictions but to more stable ones that can be applied across a range of scenarios [8]. The CDC’s primary in-season outpatient ILI forecasts are now based on an ensemble forecast generated by aggregating predictions from a growing library of individual forecasts submitted by research teams around the United States [3].

To date, most work in influenza forecasting has focused on ILI [3,5,6,13,14], with considerably less work having been focused on predicting influenza-related hospitalization rates [15]. Because influenza-related hospitalization dynamics may evolve differently than those of influenza transmission itself during a given flu season—for instance, consequent hospitalizations usually lag influenza incidence by one to two weeks **[citation needed]**—and because hospitalization rates signal the severity of circulating flu strains [5], using ensemble methods to predict hospitalization rates could be a value complement to existing efforts in ILI forecasting.

One ensemble machine learning method, dubbed “super learner” [12,16,17], exhibits a number of desirable properties that suggest it may be a powerful tool for predicting flu hospitalizations. First, its developers have demonstrated that, asymptotically, the super learner is an oracle estimator, performing as well as the best-fitting component model [16,17], where the best-fitting component model performs as well as the true data-generating model [16] **[read and cite van der Laan & Dudoit, 2003?]**. Second, this oracle property has been found to apply approximately in finite samples [12,16,17]. Several packages have been developed to implement the super learner algorithm [18,19], providing researchers easy access to a large library of component models and a means to efficiently evaluate their predictive performance [19].

The super learner requires a sufficient, though not exorbitant, sample size in order to implement embedded procedures (e.g., cross-validation) [12]. However, the 15 empirical flu seasons available are not sufficient in number to develop the super learner.

In this study, we trained an ensemble super learner on 15,000 simulated influenza hospitalization curves to generate predictions for three national-level seasonal target parameters chosen based on the targets specified in CDC forecasting competitions [20]. We conducted the procedure independently at each of the 30 weeks in a typical flu season, and we compared the predictive performance of the ensemble super learner against those of the best-performing individual statistical model and a prediction based on the mean of the outcome across the simulated hospitalization curves.

# Methods

## Empirical data

We downloaded publicly available surveillance data on seasonal influenza-related hospitalizations from the CDC’s FluView Interactive dashboard [21]. Specifically, we included data from the Emerging Infections Program (EIP) beginning with the 2003–2004 season and ending with the 2018–2019 season, omitting the 2009-2010 pandemic influenza year. The EIP contains data on influenza-related hospitalizations in California, Colorado, Connecticut, Georgia, Maryland, Minnesota, New Mexico, New York, Oregon, and Tennessee [21]. Since the 2009–2010 season the FluSurv Network has included between 3 and 6 states in addition to those represented in the EIP data, depending on the year [21]. In order to maintain consistency within the empirical data, and to increase the number of flu seasons available to inform curve simulation, we did not include data from states outside the EIP.

Typically, the CDC releases data for epiweeks 40–53 and 1–17, corresponding approximately to October through April of the next year [21,22]. Epiweeks are simply integers assigned to each week of the calendar year, where epiweek 1 is the first week of the year and epiweeks 52/53 the last [22]. We renumbered the epiweeks 1–30, omitting epiweek 53, as only three seasons had influenza hospitalization data recorded in this week. When we refer subsequently to a *flu season*, we refer to this 30-week time period.

## Prediction targets

Following from the CDC’s Flu Sight challenge [20], we focused on three season-level prediction targets:

1. *Peak rate*, defined as the highest weekly rate of influenza-related hospitalizations throughout the course of a flu season (per 100,000 population);
2. *Peak week*, defined as the week during which this peak rate occurred; and
3. *Cumulative hospitalizations*, defined as the cumulative influenza-related hospitalization rate over the 30 weeks of the flu season (per 100,000 population).

Fifteen empirical observations were available for each prediction target, corresponding to the flu seasons recorded in the CDC’s surveillance data (Table 1).

| Table 1: Empirical distributions of peak hospitalization rate, peak week, and cumulative hospitalization rate in the United States, 2003--2019. | | | |
| --- | --- | --- | --- |
| **Season** | **Peak rate** | **Peak week** | **Cumulative hospitalizations** |
| 2003-04 | 5.4 | 12.0 | 30.6 |
| 2004-05 | 1.3 | 19.0 | 13.5 |
| 2005-06 | 1.1 | 23.0 | 10.9 |
| 2006-07 | 0.5 | 20.0 | 6.2 |
| 2007-08 | 2.3 | 21.0 | 18.3 |
| 2008-09 | 0.8 | 20.5 | 7.8 |
| 2009-10 | 3.8 | 3.0 | 29.3 |
| 2010-11 | 2.1 | 21.0 | 20.1 |
| 2011-12 | 1.1 | 24.0 | 8.6 |
| 2012-13 | 5.3 | 14.0 | 43.0 |
| 2013-14 | 3.9 | 14.5 | 35.2 |
| 2014-15 | 9.0 | 13.0 | 64.2 |
| 2015-16 | 4.2 | 23.0 | 31.5 |
| 2016-17 | 5.2 | 21.0 | 62.5 |
| 2017-18 | 10.3 | 14.0 | 102.5 |
| 2018-19 | 5.4 | 24.0 | 64.9 |
| Source: CDC FluView (Emerging Infections Program). Peak rate and cumulative rate expressed per 100,000 population. Pandemic influenza season 2009--2010 omitted. | | | |

## Hospitalization curve simulation

To simulate a distribution of seasonal influenza hospitalization curves, we adapted an approach by Brooks et al. originally used to predict influenza-like illness [6]. First, we fitted a linear trend filter [23,24] to the 15 observed influenza hospitalization curves from the EIP using the glmgen package in R [25]. The glmgen linear trend filter is a penalized method that fits a piecewise linear function to a time series, testing 50 values of the penalty () by default [25]. We used these fits as templates for the simulated influenza hospitalization curves. In all cases, we selected the fit that used the 25th value tested (S1 Fig) based on visual inspection of the overall distribution of simulated curves.

Next, we used the resulting 15 trend filter fits into a modified version of the curve generation scheme described in Brooks et al. [6]. Specifically, the original scheme was developed to simulate potential ILI curves, and therefore, the curve-generating function included a term for season-onset, representing the threshold ILI at which the flu season would be considered to have started [6]. This onset is an additional prediction target in FluSight ILI forecasting because ILI surveillance is conducted year-round. On the contrary, such a threshold does not exist for influenza-related hospitalizations. Rather, hospitalization rates are recorded starting in epiweek 40 through the end of the flu season. Save for one season, hospitalization rates start at a rate of 0 per 100,000 [21]. We modified the curve-generating function to omit the seasonal onset term and added a transformation that constrains simulated hospitalization rates to be greater than or equal to 0. Otherwise, our notation borrows and follows closely from theirs.

Briefly, Brooks et al. conceptualize as seasonal influenza curve as some function plus noise [6]. Adapted to the hospitalization case, the hospitalization rate () in season and week is given by

where is a hospitalization rate and is normally distributed error term with mean 0 and variance .

For each empirical season , we use its linear trend filter fit and average the squared residuals over to estimate :

For each simulated curve, each of the following parameters is sampled randomly and independently of one another:

where denotes a randomly selected vector of estimated hospitalization rates based on a linear trend filter fit to empirical season , denotes the squared error of a linear trend filter fit to randomly sampled season and averaged across all weeks, denotes a random uniform draw from the range [0.75, 1.25], denotes a random draw from a uniform distribution bounded by the minimum and maximum peak hospitalization rates from the 15 trend filter fits, and denotes a random draw from a uniform distribution bounded by the minimum and maximum peak weeks based on the 15 trend filter fits (Table 2).

| Table 2: Input parameters to the influenza hospitalization curve generating function. | |
| --- | --- |
| **Parameter** | **Description** |
| Shape | $f \sim U \{ \hat{f} : \text{historical season } s \}$ |
| Noise | $\sigma \sim U \{\hat{\tau}^{s'} \text{ : historical season } s' \}$ |
| Pacing | $\nu \sim U[0.75, 1.25]$, stretches the curve around the peak week |
| Peak height | $\theta \sim U\left[\theta\_{min} , \theta\_{max}\right]$ |
| Peak week | $\mu \sim U[\mu\_{min}, \mu\_{max}]$ |
| The noise parameter for each simulation is drawn separately and may come from a season different than the season used as the shape. | |

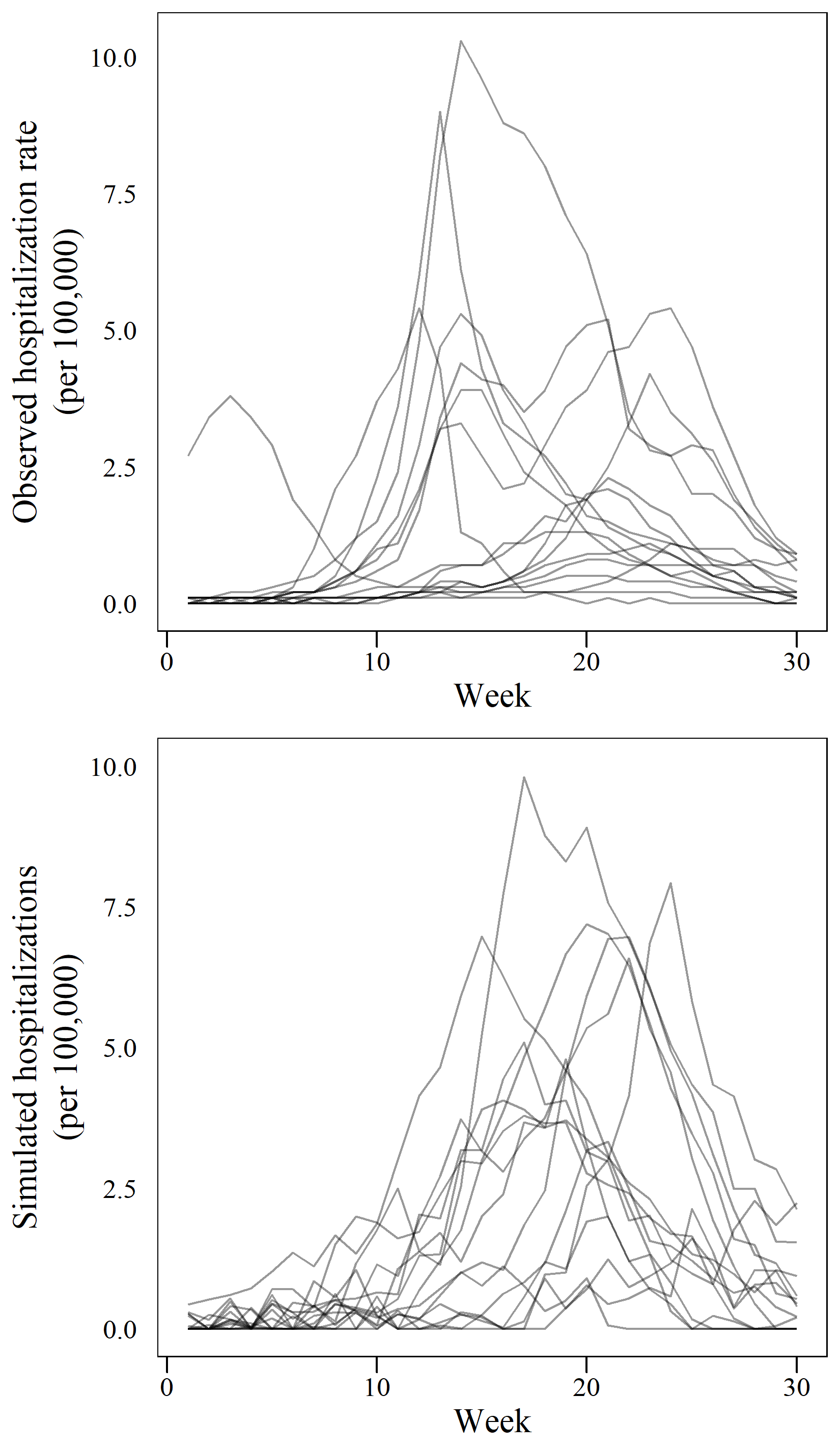
The generating function for hospitalization rate in week of simulated season therefore is given by:

where now denotes the vector of fitted hospitalization rates from the linear trend filter fit (randomly selected shape ) and the integer week for which we want to retrieve the prediction from this fit (equal to ). denotes the peak hospitalization rate from the selected shape, whereas denotes the week in which this peak occurred. The parameters , , and follow from their prior description. Finally, we introduce noise for each simulated weekly hospitalization rate based on the selected estimate of .

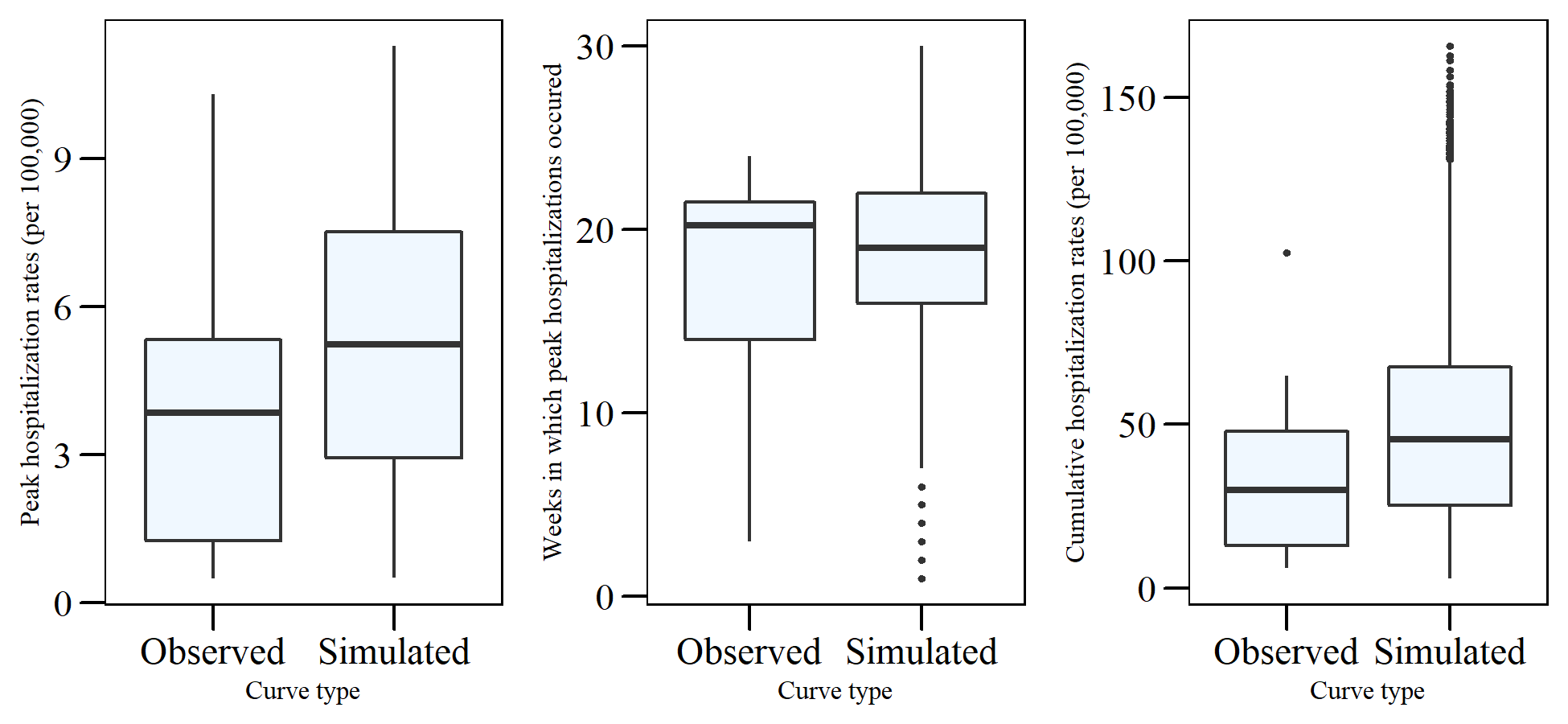
We impose a lower bound of 0 on the hospitalization rate via the following transformation of , denoted below as :

This transformation effectively preserves positive simulated hospitalization rates and sets negative hospitalization rates to 0. Negative hospitalization rates may be generated at the tails of a given season, when hospitalization rates are generally low, because the error is normally distributed in all weeks. This transformation was imposed on 20.5% of the simulated weekly hospitalization rates simulated.

In all, we simulated 15,000 curves (S2 Fig). These simulated curves may be thought of as a plausible distribution of hypothetical flu seasons that could be observed in principle but perhaps have not yet been realized [6].



Empirical (top) and 15 randomly selected simulated (bottom) hospitalization curves. Empirical source: CDC, Emerging Infections Program (omitting 2009–2010 pandemic influenza season).



Empirical (N = 15) vs. simulated (N = 15,000) target distributions.

## Super learner

The super learner is a loss-based estimation algorithm, meaning predictive performance is estimated by defining a loss function that targets a specific prediction target of interest [12,16,17]. For instance, setting the loss function to be the squared error loss means that model accuracy is optimized to minimize the squared error, and hence, the implicit prediction target is the conditional mean. The super learner takes as inputs training data, a *library* of component prediction algorithms (called *learners* but which we refer to as component models), and a desired loss function. See Naimi & Balzer for an accessible introduction [27].

For each prediction target (peak rate, peak week, and cumulative hospitalizations), we used the same library of component models and the absolute error loss function, which targets the conditional median of the prediction target [17]. The absolute error loss is defined as the absolute difference between the predicted and true observations [17].

We trained the super learner on the simulated hospitalization curves using the sl3 package in R [19]. We conducted the entire super learner procedure sequentially at each week *i*, proposing the following covariates (i.e., independent predictors) to each learner:

* hospitalization rate per 100,000 population in week *i*,
* cumulative hospitalizations per 100,000 population through week *i*,
* hospitalization rates from all weeks prior to week *i*,
* cumulative hospitalizations from all weeks prior to week *i*,
* the difference between the hospitalization rate in week *i* and the hospitalization rate in each prior week up to 5 weeks in the past,
* the difference between the cumulative hospitalization rate in week *i* and the cumulative hospitalization rate in each prior week up to 5 weeks in the past
* product terms between the hospitalization rate in week *i* and each of the cumulative hospitalization differences, and
* product terms between the cumulative hospitalization rate in week *i* and each of the hospitalization rate differences.

The predictive accuracy for the component models and the ensemble learner using 15-fold cross-validation [28]. Briefly, *V*-fold cross-validation is an iterative sample-splitting procedure in which each observation is assigned to a group, called a fold, and where each fold serves as the validation set for a model trained on the rest of the sample [27,28]. This procedure maximizes the amount of data available for model-fitting, by avoiding the need for a hold-out validation sample, while at the same time avoiding overfitting [12,27]. The predictive accuracy of each learning algorithm is evaluated as a *risk*, in our case defined as the average absolute error between a learner’s prediction and the true observation. These risks are estimated in each fold when it is assigned to be the validation set as part of the cross-validation procedure. Therefore, in the current context, the *cross-validated risk* refers to the average risk across all 15 folds [12].

We repeated the following procedure for each prediction target:

1. Create 30 data sets, one for each week of the flu season, containing the outcome variables (peak rate, peak week, and cumulative hospitalizations) and the covariates described previously.
2. Assign each observation to a fold for use in cross-validation.
   1. We assigned all simulated curves based on a common observed season shape to the same fold, in order to account for the dependence between these simulated curves. We did not account for potential dependencies due to repeated sampling of the error term, as we believed the season shape was more consequential with respect to inducing dependence between simulated curves.
3. Run the super learner algorithm:
   1. Train each learner on the data set using 15-fold cross-validation and calculate its cross-validated risk as the average risk across the 15 validation sets.
   2. Generate predictions with each component learner on the full data set.
   3. Regress the current prediction target on each learner’s predictions in the full sample (*N* = 15,000) using a non-negative least squares model constrained to produce coefficients in the range [0,1] and which sum to 1 [27]. These constraints are not necessary but are theoretically expected to improve the stability of the ensemble estimator [12]. This model is referred to as the *metalearner* [19].
   4. Calculate the cross-validated risk for the ensemble super learner as in step 4a.

In all, we trained **[62]** component models in for each prediction target and week of the season, including variations on tuning and input parameters for the various learners (Table 2).

## Component models

We used the same library of component models to build the super learners for all three prediction targets (Table 3). The component models included a standard linear regression model, generalized additive models [29], loss-based regression models (e.g., lasso, ridge) [30], random forests [31], support vector regression [9], neural networks [9], loess [28], and polynomial multivariate adaptive regression splines (polyMARS) [32]. Theory and finite-sample demonstrations support the idea that the super learner should benefit from being provided with a large variety of models, even if some of those models perform poorly in the given data [12]. Therefore, we specified the component learners and tuning parameter sets to attempt to probe a large portion of the component model space.

In all cases the learning algorithms incorporating these various component models were implemented using functions provided in the sl3 [19] and SuperLearner [18] packages in R.

| Table 3: Component models and tuning parameters. | | |
| --- | --- | --- |
| **Model** | **Tuning parameters** | **R package** |
| Linear regression | Variables screened for inclusion first using a cross-validated lasso procedure to omit variables with zero-valued coefficients. | base |
| Random forest (regression trees) | number of trees = [50, 100, 200, 500], terminal node sizes = [3, 5, 10] | randomForest |
| Generalized additive model | gamma penalty = [1, 2, 3, 4, 5] | gam |
| Support vector regression | kernel = [radial, polynomial], degree = [1, 2, 3] applied only to polynomial | svm |
| Loss-based regression | penalty = [lasso, ridge] | glmnet |
| Elastic net | alpha penalty = [0.25, 0.5, 0.75] | glmnet |
| Neural network | number of nodes in hidden layer = [5, 10, 25, 50, 75, 100], decay = [0, 0.005, 0.1, 0.2, 0.4] | nnet |
| Loess | span = [0.25, 0.5, 0.75, 1] | base |
| Polynomial multivariate adaptive regression spline | gcv penalty = [2, 4, 6, 8, 10] | polspline |
| For the random forest and neural network learners, all combinations of the tuning parameters shown were proposed. | | |

### Linear regression

For each week of the flu season, we specified a linear regression model that included covariates chosen by a lasso-based screening algorithm [18,33] in which covariates with non-zero coefficients were identified using 10-fold cross validation and then entered into the linear regression. The linear model was specified using the glm() function in R with a Gaussian error distribution and identity link.

### Loss-based regression

We implemented several loss-based regression learners using the glmnet package in R [33]. These learners included a learner with the standard lasso penalty [30], a learner with the ridge penalty [30], and three versions of the elastic net model [cite] using different penalties (Table 3). The elastic net model combines lasso and ridge penalties within the same model [citation].

### Random forests

We implemented random forest algorithms [31] using the randomForest package [34], tuning parameters governing the number of regression trees and minimum observations allowed in terminal nodes (Table 3).

### Support vector regression

We implemented several support vector regressions [9] using the e1071 package [35]—one using a radial kernel and three others using a polynomial kernel of differing degrees (Table 3).

### Neural networks

We implemented several neural networks [9] using the nnet package [36]. All learners proposed contained a single hidden layer, altering tuning parameters governing the number of nodes in this hidden layer and the magnitude of decay applied to node weights (Table 3).

### Loess

We submitted several loess-based learners as implemented in base R, varying the smoothing kernel span (Table 3).

### Polynomial multivariate adaptive regression splines

We submitted several polyMARS models using the polspline package [32], varying the generalized cross-validation value, larger values of which produce smaller models [32].

## Technical details

The super learner algorithm was implemented on a high-performance computing cluster housed at the Center for Computation and Visualization at Brown University. Each run was submitted to a large-memory node and parallelized across 32 central processing units based on the native functionality provided in sl3 [19] via the delayed [37] and future [38] packages in R.

# Results

## Peak rate

For the peak rate prediction target, the ensemble super learner produced risk estimates similar to the simple mean prediction and generally worse than the discrete super learner (Table 4). Throughout the flu season, the discrete super learner (best-performing component model) consistently exhibited a lower average risk than a prediction based on the mean, indicating better predictive ability. Furthermore, the discrete super learner’s cross-validated risk generally decreased as the season progressed. In each week a different discrete super learner was selected, with neural networks and penalized regressions being selected in the first several weeks, support vector regressions being selected toward the middle of the flu season, and polyMARS being selected more frequently toward the end of the season.

| Table 4: Cross-validated risks for the ensemble super learner, discrete super learner, and simple mean predictions of peak hospitalization rate, by week of influenza season. Estimates presented as mean risk (standard error). | | | | |
| --- | --- | --- | --- | --- |
| **Week** | **Super Learner** | **Discrete SL Risk** | **Mean** | **Discrete SL** |
| 1 | 2.83 (0.013) | 2.68 (0.013) | 2.75 (0.012) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 2 | 2.97 (0.022) | 2.20 (0.015) | 2.75 (0.012) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_10\_0.005 |
| 3 | 2.93 (0.025) | 1.77 (0.014) | 2.75 (0.012) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_5\_0.005 |
| 4 | 2.57 (0.020) | 1.85 (0.019) | 2.75 (0.012) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_50\_0.2 |
| 5 | 2.97 (0.032) | 2.50 (0.028) | 2.75 (0.012) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_5\_0.005 |
| 6 | 2.59 (0.018) | 2.36 (0.019) | 2.75 (0.012) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_10\_0.2 |
| 7 | 2.74 (0.026) | 2.19 (0.014) | 2.75 (0.012) | Lrnr\_glmnet\_NULL\_deviance\_10\_0.5\_100\_TRUE |
| 8 | 2.97 (0.017) | 2.46 (0.021) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 9 | 2.98 (0.033) | 2.40 (0.014) | 2.75 (0.012) | Lrnr\_randomForest\_100\_TRUE\_10\_NULL\_FALSE |
| 10 | 3.00 (0.022) | 2.20 (0.022) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 11 | 3.26 (0.015) | 2.27 (0.017) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 12 | 3.11 (0.018) | 2.32 (0.024) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 13 | 2.91 (0.017) | 1.99 (0.026) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 14 | 3.51 (0.016) | 1.97 (0.025) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 15 | 3.29 (0.012) | 2.11 (0.025) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 16 | 3.24 (0.019) | 2.16 (0.025) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 17 | 3.36 (0.015) | 2.11 (0.021) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 18 | 3.14 (0.029) | 2.09 (0.030) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 19 | 2.71 (0.030) | 1.33 (0.025) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 20 | 2.42 (0.019) | 1.22 (0.022) | 2.75 (0.012) | Lrnr\_polspline\_5\_2, Lrnr\_polspline\_5\_4 |
| 21 | 2.73 (0.018) | 1.06 (0.024) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 22 | 2.64 (0.021) | 0.86 (0.025) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 23 | 2.56 (0.020) | 0.44 (0.020) | 2.75 (0.012) | Lrnr\_polspline\_5\_2 |
| 24 | 2.75 (0.022) | 0.68 (0.021) | 2.75 (0.012) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 25 | 2.64 (0.017) | 0.74 (0.020) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 26 | 2.73 (0.017) | 0.78 (0.021) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 27 | 2.69 (0.015) | 0.67 (0.022) | 2.75 (0.012) | Lrnr\_polspline\_5\_2 |
| 28 | 2.63 (0.017) | 0.63 (0.022) | 2.75 (0.012) | Lrnr\_polspline\_5\_2 |
| 29 | 2.41 (0.016) | 0.63 (0.022) | 2.75 (0.012) | Lrnr\_polspline\_5\_2 |
| 30 | 2.52 (0.018) | 0.72 (0.021) | 2.75 (0.012) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| The ensemble super learner (SuperLearner) is the prediction generated as a weighted combination of component model predictions. The discrete super learner (Discrete SL) is the best-performing component model. The mean (Mean) is the average peak hospitalization rate per 100,000 population across the simulated hospitalization curves. The absolute error loss function used to optimized the super learner targets the median peak hospitalization rate. | | | | |

## Peak week

For the peak week prediction target, the ensemble superlearner generally produced risk estimates close to those of the peak week predictions based on a simple mean (Table 5). The discrete super learner consistently exhibited lower mean risk than both the ensemble super learner and predictions based on the simple mean. As with peak rate, the discrete super learner tended to perform better as the flu season progressed. Over the first 6 weeks, some form of neural network was chosen as the discrete super learner in 5 of those weeks. Throughout the rest of the season, various versions of the random forest algorithms predominated the list of discrete super learners.

| Table 5: Cross-validated risks for the ensemble super learner, discrete super learner, and mean predictions of the week in which the peak hospitalization rate occurs, by week of influenza season. Estimates presented as mean risk (standard error). | | | | |
| --- | --- | --- | --- | --- |
| **Week** | **Super Learner** | **Discrete SL Risk** | **Mean** | **Discrete SL** |
| 1 | 4.02 (0.027) | 3.34 (0.037) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_50\_0.4 |
| 2 | 4.49 (0.032) | 3.50 (0.031) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_25\_0.2 |
| 3 | 4.59 (0.024) | 2.58 (0.027) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_5\_0.005 |
| 4 | 4.63 (0.025) | 2.80 (0.021) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_100\_0.005 |
| 5 | 4.45 (0.030) | 2.43 (0.026) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_10\_0.005 |
| 6 | 3.98 (0.022) | 2.82 (0.018) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_5\_0.005 |
| 7 | 4.45 (0.033) | 2.81 (0.024) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_3\_NULL\_FALSE |
| 8 | 4.94 (0.042) | 3.23 (0.024) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_5\_NULL\_FALSE |
| 9 | 3.80 (0.022) | 2.99 (0.026) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_10\_NULL\_FALSE |
| 10 | 4.15 (0.023) | 3.04 (0.026) | 3.92 (0.027) | Lrnr\_randomForest\_500\_TRUE\_10\_NULL\_FALSE |
| 11 | 4.10 (0.038) | 2.60 (0.023) | 3.92 (0.027) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 12 | 3.97 (0.024) | 2.18 (0.034) | 3.92 (0.027) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_10\_0.4 |
| 13 | 3.72 (0.020) | 2.23 (0.025) | 3.92 (0.027) | Lrnr\_randomForest\_100\_TRUE\_10\_NULL\_FALSE |
| 14 | 3.47 (0.021) | 2.29 (0.024) | 3.92 (0.027) | Lrnr\_polspline\_5\_2 |
| 15 | 3.66 (0.020) | 1.95 (0.028) | 3.92 (0.027) | Lrnr\_randomForest\_200\_TRUE\_10\_NULL\_FALSE |
| 16 | 3.97 (0.026) | 1.89 (0.027) | 3.92 (0.027) | Lrnr\_randomForest\_200\_TRUE\_5\_NULL\_FALSE |
| 17 | 4.43 (0.036) | 1.85 (0.022) | 3.92 (0.027) | Lrnr\_randomForest\_100\_TRUE\_5\_NULL\_FALSE |
| 18 | 3.90 (0.028) | 1.79 (0.023) | 3.92 (0.027) | Lrnr\_randomForest\_100\_TRUE\_3\_NULL\_FALSE |
| 19 | 4.26 (0.032) | 1.71 (0.024) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_3\_NULL\_FALSE |
| 20 | 3.97 (0.027) | 2.15 (0.023) | 3.92 (0.027) | Lrnr\_randomForest\_500\_TRUE\_10\_NULL\_FALSE |
| 21 | 4.14 (0.026) | 1.94 (0.023) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_10\_NULL\_FALSE |
| 22 | 3.78 (0.022) | 1.78 (0.021) | 3.92 (0.027) | Lrnr\_randomForest\_100\_TRUE\_10\_NULL\_FALSE |
| 23 | 3.94 (0.020) | 1.92 (0.022) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_10\_NULL\_FALSE |
| 24 | 4.08 (0.032) | 1.84 (0.025) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_3\_NULL\_FALSE |
| 25 | 3.91 (0.020) | 1.98 (0.022) | 3.92 (0.027) | Lrnr\_randomForest\_200\_TRUE\_3\_NULL\_FALSE |
| 26 | 3.91 (0.023) | 1.98 (0.022) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_3\_NULL\_FALSE |
| 27 | 4.12 (0.025) | 2.06 (0.022) | 3.92 (0.027) | Lrnr\_randomForest\_100\_TRUE\_3\_NULL\_FALSE |
| 28 | 3.60 (0.021) | 2.14 (0.019) | 3.92 (0.027) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 29 | 4.27 (0.023) | 2.00 (0.023) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_10\_NULL\_FALSE |
| 30 | 3.83 (0.022) | 1.86 (0.023) | 3.92 (0.027) | Lrnr\_randomForest\_50\_TRUE\_3\_NULL\_FALSE |
| The ensemble super learner (SuperLearner) is the prediction generated as a weighted combination of component model predictions. The discrete super learner (Discrete SL) is the best-performing component model. The mean (Mean) is the average week during which the peak hospitalization rate occurred across the simulated hospitalization curves. The absolute error loss function used to optimized the super learner targets the median peak hospitalization rate. | | | | |

## Cumulative hospitalizations

For the cumulative hospitalizations prediction target, the ensemble super learner produced risk estimates generally near to those based on a prediction based on the simple mean (Table 6). Again, the discrete super learner consistently outperformed both the ensemble super learner and the simple mean prediction, improving substantially as the flu season progressed. Some form of support vector regression was chosen most frequently as the discrete super learner, with penalized regressions, polyMARS, and neural networks among the other models chosen in at least one of the weeks.

| Table 6: Cross-validated risks for the ensemble super learner, discrete super learner, and mean predictions of the cumulative hospitalization rate, by week of influenza season. Estimates presented as mean risk (standard error). | | | | |
| --- | --- | --- | --- | --- |
| **Week** | **Super Learner** | **Discrete SL Risk** | **Mean** | **Discrete SL** |
| 1 | 33.4 (1.01) | 28.4 (1.17) | 33.4 (1.01) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_75\_0.2 |
| 2 | 36.3 (1.65) | 23.2 (1.62) | 33.4 (1.01) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 3 | 30.6 (1.411) | 24.6 (0.797) | 33.4 (1.010) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_10\_0.2 |
| 4 | 28.6 (1.708) | 22.7 (0.861) | 33.4 (1.010) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_5\_0 |
| 5 | 38.4 (1.035) | 24.4 (0.966) | 33.4 (1.010) | Lrnr\_polspline\_5\_4 |
| 6 | 32.5 (1.339) | 26.3 (0.584) | 33.4 (1.010) | Lrnr\_pkg\_SuperLearner\_SL.nnet\_100\_0.005 |
| 7 | 32.0 (1.88) | 25.0 (1.20) | 33.4 (1.01) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 8 | 32.6 (0.979) | 25.7 (1.627) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 9 | 32.6 (1.25) | 24.1 (1.02) | 33.4 (1.01) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 10 | 32.4 (0.869) | 22.4 (1.416) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 11 | 36.1 (0.751) | 23.8 (1.333) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 12 | 32.0 (0.993) | 22.9 (1.393) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 13 | 30.6 (1.32) | 24.4 (1.54) | 33.4 (1.01) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 14 | 36.8 (0.902) | 23.7 (1.535) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 15 | 36.4 (0.925) | 21.7 (1.523) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_radial\_TRUE\_FALSE |
| 16 | 35.8 (1.01) | 17.2 (1.31) | 33.4 (1.01) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 17 | 31.0 (0.993) | 20.3 (1.238) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 18 | 35.3 (0.949) | 20.6 (1.448) | 33.4 (1.010) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 19 | 33.6 (1.09) | 15.8 (1.30) | 33.4 (1.01) | Lrnr\_glmnet\_NULL\_deviance\_10\_0.75\_100\_TRUE |
| 20 | 31.2 (1.28) | 15.9 (1.35) | 33.4 (1.01) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 21 | 36.34 (0.782) | 9.82 (1.031) | 33.37 (1.010) | Lrnr\_svm\_TRUE\_NULL\_polynomial\_TRUE\_FALSE\_1 |
| 22 | 33.49 (0.915) | 4.44 (0.862) | 33.37 (1.010) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 23 | 29.29 (0.924) | 2.89 (0.865) | 33.37 (1.010) | Lrnr\_polspline\_5\_2 |
| 24 | 30.50 (1.091) | 2.87 (0.886) | 33.37 (1.010) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 25 | 32.80 (1.027) | 2.46 (0.874) | 33.37 (1.010) | Lrnr\_glmnet\_NULL\_deviance\_10\_1\_100\_TRUE\_TRUE |
| 26 | 36.57 (0.828) | 1.93 (0.851) | 33.37 (1.010) | Lrnr\_polspline\_5\_4 |
| 27 | 31.29 (1.10) | 0.83 (0.85) | 33.37 (1.01) | Lrnr\_polspline\_5\_4 |
| 28 | 34.10 (0.942) | 0.85 (0.855) | 33.37 (1.010) | Lrnr\_polspline\_5\_4 |
| 29 | 34.97 (0.909) | 0.15 (0.842) | 33.37 (1.010) | Lrnr\_polspline\_5\_4 |
| 30 | 32.8 (1.001) | 0.0 (0.842) | 33.4 (1.010) | Pipeline(Lrnr\_pkg\_SuperLearner\_screener\_screen.glmnet->Lrnr\_glm\_TRUE) |
| The ensemble super learner (SuperLearner) is the prediction generated as a weighted combination of component model predictions. The discrete super learner (Discrete SL) is the best-performing component model. The mean (Mean) is the average cumulative hospitalization rate per 100,000 across the simulated hospitalization curves. The absolute error loss function used to optimized the super learner targets the median peak hospitalization rate. | | | | |

# Discussion

In our study, we found that discrete super learners were consistently better at predicting median peak rate, peak week, and cumulative hospitalizations compared to a simple mean prediction when fit to a distribution of hypothetical influenza hospitalization curves. On the contrary, the ensemble super learner generally tended to perform comparably to a simple mean prediction and consistently worse than the discrete super learners with respect to cross-validated risk. In fact, the discrete super learner routinely provided the best predictions (lowest mean cross-validated risk) across all prediction targets. The discrete super learner also appeared to provide greater improvements over the simple mean prediction as a season wore on, a result that comports with similar findings in the domain of influenza-like illness [8].

The relative underperformance of the ensemble super learner, a feature observed across all prediction targets, raises several questions regarding the likely usefulness of the super learner algorithm in the context of predicting influenza hospitalizations. It may be that the ensemble super learner should not be relied upon if used to generate in-season forecasts given the small sample size of observed hospitalization curves [21] and the fact that we did not observe improvements in prediction in a sample size of 15,000. As discussed in the introduction, the ensemble super learner has desirable asymptotic oracle properties, theoretically guaranteed to preform as well as the discrete super learner [12]. Furthermore, the ensemble super learner is generally assumed not to be affected adversely by the inclusion of poor algorithms or overfitting, as the cross-validation procedure should result in a final weighted ensemble that assigns low or no weight to poor-performing predictions [17]. It is possible that adding additional algorithms such as gradient boosted models or exploring a larger range of tuning parameters for the models incorporated in our analysis would have improved the performance of the ensemble learner.

Our findings should be interpreted in light of several limitations.

First, while our simulating 15,000 hypothetical hospitalization curves provided an adequate sample size with which to train the super learner—an analysis that would not have been possible using only the 15 observed flu seasons—the validity of our findings rely on the assumption that the distribution of simulated curves accurately characterizes the hypothetical distribution of plausible seasonal influenza hospitalization trajectories. We did not attempt to rigidly match the simulated distributions to their empirical counterparts so as not to restrict the possibility of atypical influenza seasons that could be, but perhaps have not yet been, observed. In other words, we did not assume the 15 observed seasons captured the full range of possible values for the selected prediction targets. A related issue concerns the selection of the value for each trend filter fit to an empirical season, as these fits were used as templates for the simulated hospitalization curves. This choice of penalty was somewhat arbitrary. Selecting the penalty that minimized the root mean squared error of the trend filter predictions with respect to the empirical data may have resulted in a distribution of simulated curves with different properties. However, while methods for selecting the penalty for the trend filter have been proposed [39], to our knowledge, no consensus currently exists regarding the best procedure to do so.

Second, we did not include any mechanistic models in the component learner library, such as compartmental or agent-based epidemic models. Future work could incorporate such models or, alternatively, the predictions generated by an ensemble super learner could be incorporated into a wider ensemble model that incorporates predictions from such models.

Third, our approach to simulating hypothetical influenza hospitalization curves incorporated only information on hospitalization rates, without relying on other mechanistic factors to generate these curves. Therefore, we were unable to examine whether incorporating other influenza-related surveillance data (e.g., ILI, viral activity [21]) or climate data may have improved the super learner’s predictions.

Fourth, we did not model hospitalization rate time series explicitly, in order to increase the number of component models available for inclusion [19]. Nonetheless, we did attempt to incorporate information regarding the trajectory of influenza hospitalization rates up to a given week by incorporating lagged hospitalization rates, differences between the hospitalization rates in the current week and prior weeks, and selected interactions. Future work should explore the potential for explicit modeling of time series, perhaps using an “online” ensemble super learner, which continuously updates the super learner’s predictions based on its predictions in prior weeks [40].

Finally, because we focused on predicting hospitalizations due to seasonal influenza, these findings do not have implications for predicting hospitalizations during years in which a pandemic influenza strain is circulating.

The super learner remains a promising algorithm for predicting influenza-related hospitalizations, but future work should determine the proper breadth of component learners such that the ensemble learner can be relied upon to perform as well as the discrete super learner. Adapting this approach to incorporate empirical surveillance data as well as other data relevant to influenza dynamics (e.g., viral activity, weather patterns), and to focus on hospitalizations from specific settings (e.g., skilled nursing facilities), may also be fruitful avenues for future research.

# Software and code

The code necessary to reproduce these results and the output from the super learner procedures are provided in a repository at Open Science Framework **(doi: XXXXXXXXX)**. These materials are also housed permanently at the Brown Digital Repository **(doi/url: XXXXXXXXXXX)**, and the package can be installed directly via the Github repository (<https://www.github.com/jrgant/FluHospPrediction>).

# Declarations

## Acknowledgment

We thank Ashley Naimi, Laura Balzer, and Nicholas Reich for their comments on the study aims and the super learner approach. The super learner algorithm was fit using computational resources and services at the Center for Computation and Visualization, Brown University.

## Funding statement

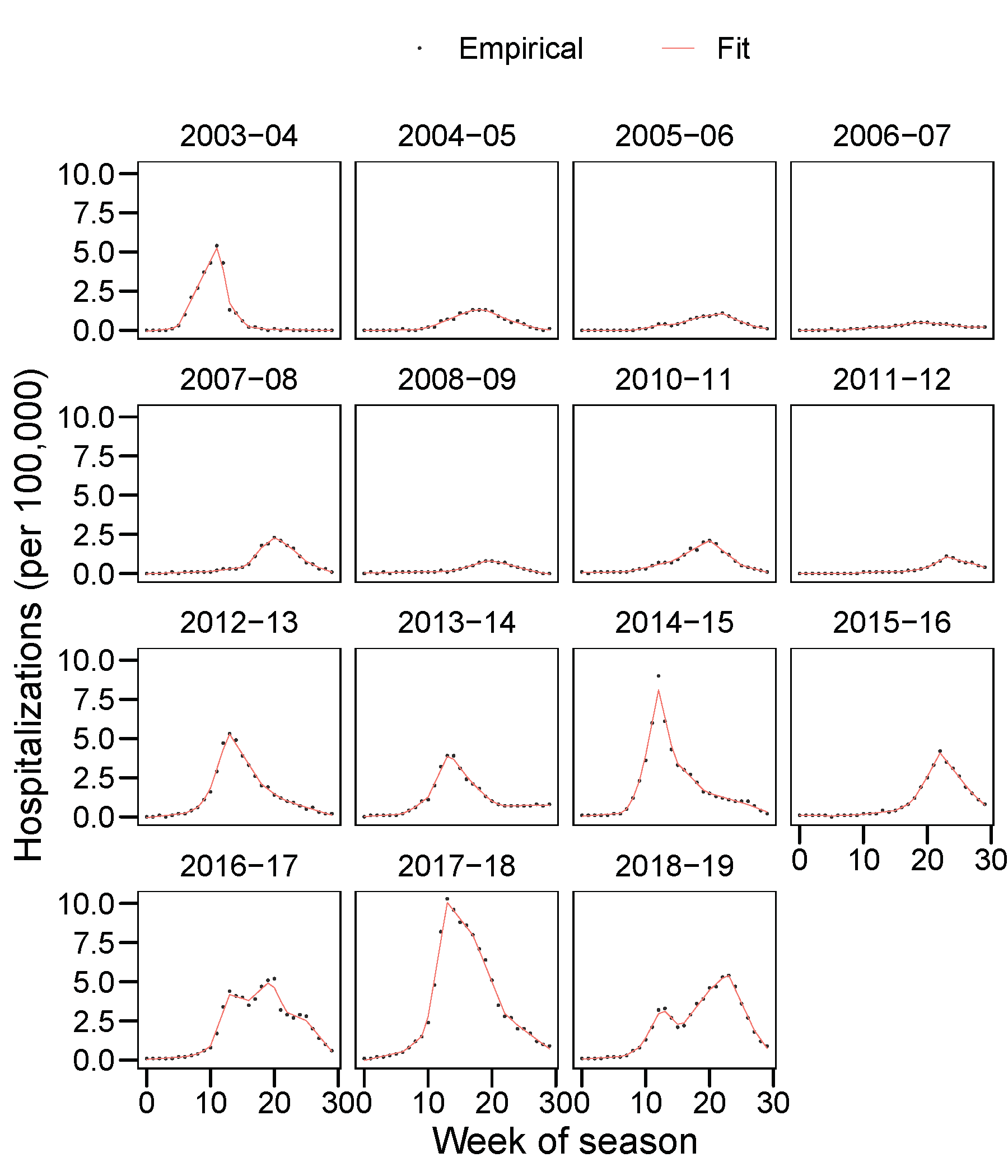
This work was funded by an unrestricted grant from Sanofi (PI: Andrew Zullo). The funders did not assist in the statistical analysis nor did they have a say in the final decision to submit the manuscript for publication.

## Competing interests

[solicit competing interests from co-authors]

# Supporting information

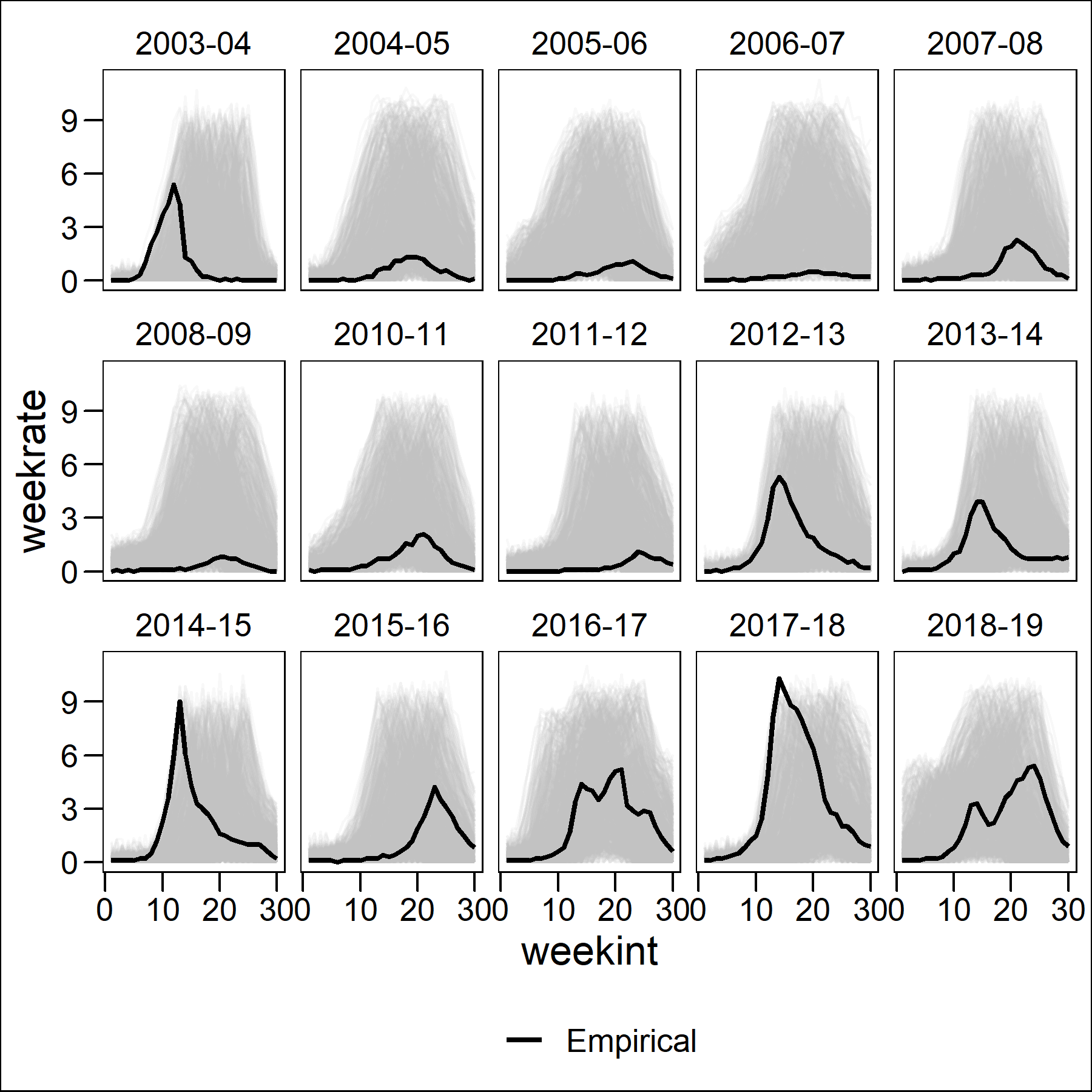
**S1 Fig. Linear trend filter fits to observed influenza hospitalization curves.**



**S2 Table. Number of simulated curves based on each observed flu season (Emerging Infections Program).**

| **Template season** | **N** |
| --- | --- |
| 2003-04 | 1,032 |
| 2004-05 | 979 |
| 2005-06 | 972 |
| 2006-07 | 1,021 |
| 2007-08 | 994 |
| 2008-09 | 1,033 |
| 2010-11 | 946 |
| 2011-12 | 956 |
| 2012-13 | 999 |
| 2013-14 | 1,028 |
| 2014-15 | 1,013 |
| 2015-16 | 995 |
| 2016-17 | 1,040 |
| 2017-18 | 986 |
| 2018-19 | 1,006 |

**S3 Figure. Simulated hospitalization curves generated based on each empirical shape template.** Note that because each parameter used in the curve-generating function was drawn independently, simulated hospitalization curves based on an empirical shape template should have a similar shape (i.e., unimodal, bimodal) but may have very different peak and/or cumulative hospitalization rates compared to the empirical template.



**S4 Table. Weekly cross-validated risks across the component learners used to predict peak hospitalization rate per 100,000 population.**

| **Week** | **Mean** | **SD** | **Median** | **Minimum** | **Maximum** |
| --- | --- | --- | --- | --- | --- |
| 1 | 10.09 | 20.41 | 3.16 | 2.68 | 82.75 |
| 2 | 8.98 | 19.82 | 2.80 | 2.20 | 82.75 |
| 3 | 8.19 | 19.77 | 2.75 | 1.77 | 82.75 |
| 4 | 10.74 | 21.72 | 2.65 | 1.85 | 82.75 |
| 5 | 21.91 | 80.05 | 4.54 | 2.50 | 621.47 |
| 6 | 9.12 | 19.60 | 3.31 | 2.36 | 82.75 |
| 7 | 9.02 | 19.61 | 3.36 | 2.19 | 82.75 |
| 8 | 8.65 | 19.65 | 3.30 | 2.46 | 82.75 |
| 9 | 9.02 | 19.63 | 3.28 | 2.40 | 82.75 |
| 10 | 8.41 | 19.70 | 2.92 | 2.20 | 82.75 |
| 11 | 8.47 | 19.68 | 3.46 | 2.27 | 82.75 |
| 12 | 8.46 | 19.69 | 3.19 | 2.32 | 82.75 |
| 13 | 8.18 | 19.75 | 3.00 | 1.99 | 82.75 |
| 14 | 8.34 | 19.74 | 3.07 | 1.97 | 82.75 |
| 15 | 8.32 | 19.72 | 3.09 | 2.11 | 82.75 |
| 16 | 8.25 | 19.74 | 2.98 | 2.16 | 82.75 |
| 17 | 8.16 | 19.75 | 3.02 | 2.11 | 82.75 |
| 18 | 8.10 | 19.77 | 2.89 | 2.09 | 82.75 |
| 19 | 7.77 | 19.86 | 2.68 | 1.33 | 82.75 |
| 20 | 7.58 | 19.91 | 2.59 | 1.22 | 82.75 |
| 21 | 7.48 | 19.94 | 2.60 | 1.06 | 82.75 |
| 22 | 7.38 | 19.97 | 2.66 | 0.86 | 82.75 |
| 23 | 7.39 | 19.97 | 2.67 | 0.44 | 82.75 |
| 24 | 7.38 | 19.98 | 2.66 | 0.68 | 82.75 |
| 25 | 7.39 | 19.97 | 2.75 | 0.74 | 82.75 |
| 26 | 7.36 | 19.98 | 2.70 | 0.78 | 82.75 |
| 27 | 7.32 | 19.99 | 2.63 | 0.67 | 82.75 |
| 28 | 7.32 | 19.99 | 2.64 | 0.63 | 82.75 |
| 29 | 7.32 | 19.99 | 2.65 | 0.63 | 82.75 |
| 30 | 7.32 | 19.99 | 2.69 | 0.72 | 82.75 |

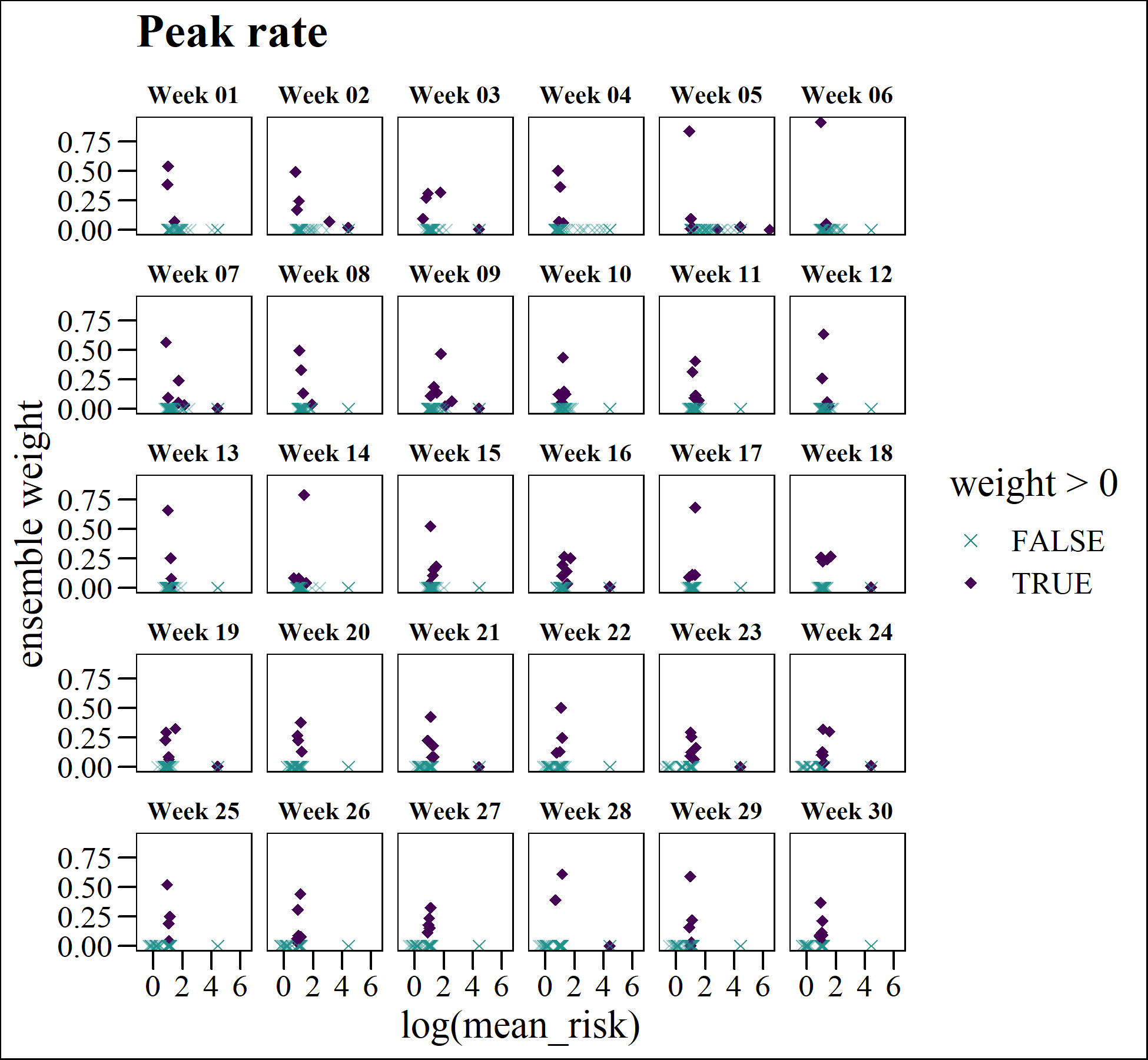
**S5 Table. Weekly cross-validated risks across the component learners used to predict peak week.**

| **Week** | **Mean** | **SD** | **Median** | **Minimum** | **Maximum** |
| --- | --- | --- | --- | --- | --- |
| 1 | 12.55 | 29.68 | 3.92 | 3.34 | 122.43 |
| 2 | 17.44 | 36.86 | 4.33 | 3.50 | 164.76 |
| 3 | 11.88 | 29.30 | 3.92 | 2.58 | 122.43 |
| 4 | 11.89 | 29.32 | 3.82 | 2.80 | 122.43 |
| 5 | 60.17 | 368.34 | 3.76 | 2.43 | 2,902.90 |
| 6 | 11.97 | 29.32 | 3.91 | 2.82 | 122.43 |
| 7 | 12.12 | 29.28 | 4.07 | 2.81 | 122.43 |
| 8 | 12.62 | 29.23 | 4.42 | 3.23 | 122.43 |
| 9 | 12.39 | 29.44 | 4.16 | 2.99 | 122.43 |
| 10 | 12.39 | 29.18 | 4.60 | 3.04 | 122.43 |
| 11 | 11.77 | 29.32 | 4.09 | 2.60 | 122.43 |
| 12 | 11.45 | 29.43 | 3.65 | 2.18 | 122.43 |
| 13 | 11.20 | 29.47 | 3.54 | 2.23 | 122.43 |
| 14 | 11.21 | 29.46 | 3.62 | 2.29 | 122.43 |
| 15 | 11.08 | 29.50 | 3.60 | 1.95 | 122.43 |
| 16 | 11.30 | 29.44 | 3.84 | 1.89 | 122.43 |
| 17 | 11.21 | 29.47 | 3.53 | 1.85 | 122.43 |
| 18 | 11.14 | 29.49 | 3.52 | 1.79 | 122.43 |
| 19 | 10.94 | 29.53 | 3.36 | 1.71 | 122.43 |
| 20 | 10.93 | 29.53 | 3.44 | 2.15 | 122.43 |
| 21 | 11.24 | 29.48 | 3.47 | 1.94 | 122.43 |
| 22 | 10.92 | 29.54 | 3.49 | 1.78 | 122.43 |
| 23 | 10.96 | 29.53 | 3.43 | 1.92 | 122.43 |
| 24 | 10.98 | 29.53 | 3.50 | 1.84 | 122.43 |
| 25 | 11.03 | 29.51 | 3.67 | 1.98 | 122.43 |
| 26 | 11.07 | 29.51 | 3.61 | 1.98 | 122.43 |
| 27 | 11.14 | 29.48 | 3.78 | 2.06 | 122.43 |
| 28 | 11.07 | 29.49 | 3.69 | 2.14 | 122.43 |
| 29 | 11.13 | 29.48 | 3.67 | 2.00 | 122.43 |
| 30 | 11.03 | 29.51 | 3.67 | 1.86 | 122.43 |

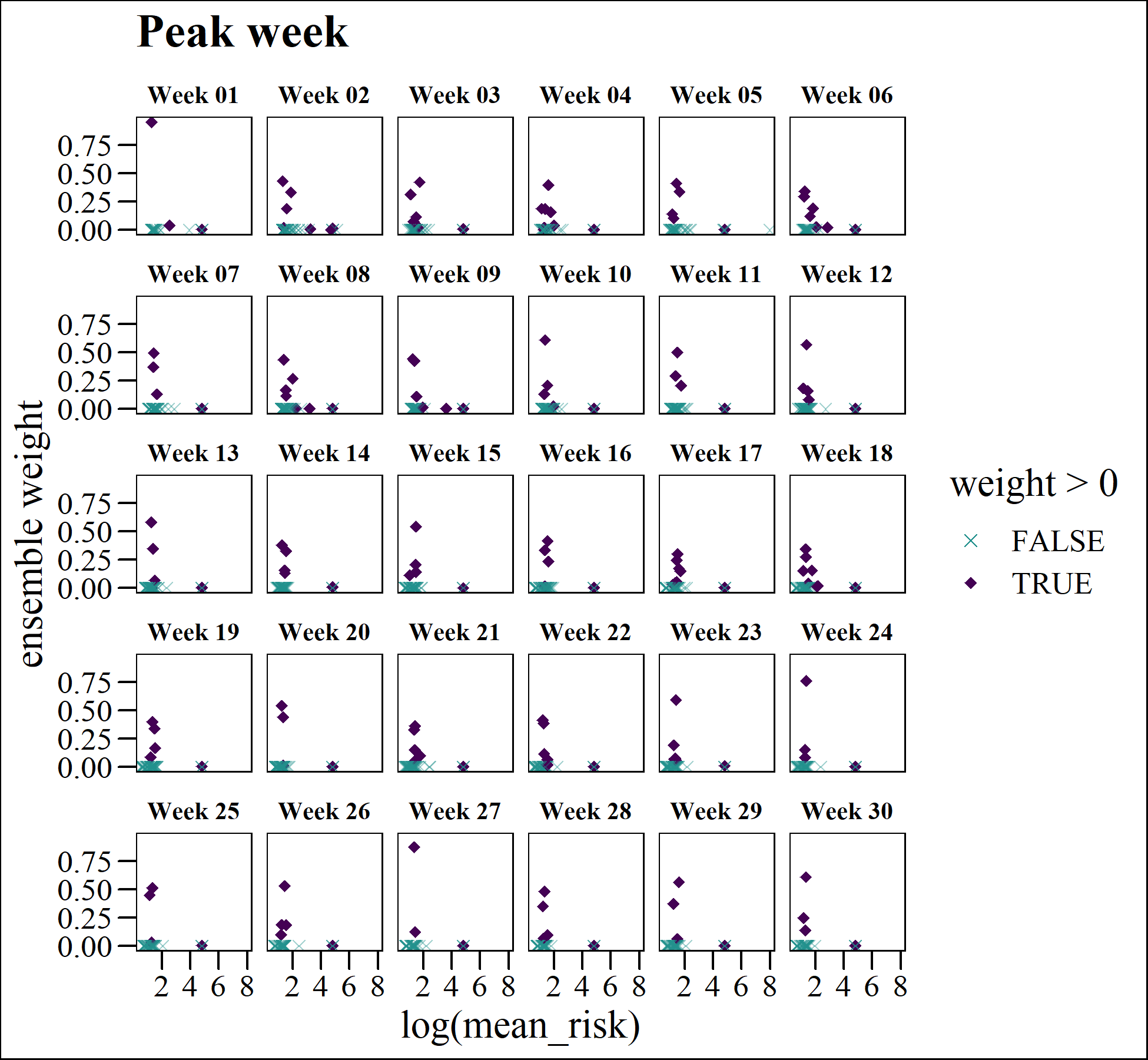
**S6 Table. Weekly cross-validated risks across the component learners used to predict cumulative hospitalization rate per 100,000 population.**

| **Week** | **Mean** | **SD** | **Median** | **Minimum** | **Maximum** |
| --- | --- | --- | --- | --- | --- |
| 1 | 117.34 | 266.81 | 34.81 | 28.41 | 1,026.83 |
| 2 | 100.78 | 246.65 | 32.08 | 23.19 | 1,026.83 |
| 3 | 97.52 | 246.66 | 30.42 | 24.61 | 1,026.83 |
| 4 | 95.06 | 247.04 | 28.60 | 22.74 | 1,026.83 |
| 5 | 326.71 | 1,647.27 | 34.07 | 24.44 | 12,918.25 |
| 6 | 98.59 | 245.86 | 33.26 | 26.26 | 1,026.83 |
| 7 | 96.82 | 246.28 | 31.85 | 25.00 | 1,026.83 |
| 8 | 99.44 | 245.84 | 32.86 | 25.73 | 1,026.83 |
| 9 | 98.57 | 246.50 | 31.58 | 24.05 | 1,026.83 |
| 10 | 96.72 | 246.30 | 32.71 | 22.36 | 1,026.83 |
| 11 | 97.63 | 246.06 | 32.87 | 23.78 | 1,026.83 |
| 12 | 96.59 | 246.32 | 32.89 | 22.90 | 1,026.83 |
| 13 | 96.32 | 246.41 | 32.87 | 24.35 | 1,026.83 |
| 14 | 97.37 | 246.24 | 32.61 | 23.71 | 1,026.83 |
| 15 | 97.47 | 246.13 | 33.03 | 21.71 | 1,026.83 |
| 16 | 95.84 | 246.54 | 32.41 | 17.21 | 1,026.83 |
| 17 | 96.78 | 246.27 | 33.14 | 20.29 | 1,026.83 |
| 18 | 96.59 | 246.33 | 33.01 | 20.55 | 1,026.83 |
| 19 | 93.65 | 247.14 | 32.62 | 15.76 | 1,026.83 |
| 20 | 92.52 | 247.45 | 32.03 | 15.90 | 1,026.83 |
| 21 | 91.71 | 247.74 | 32.52 | 9.82 | 1,026.83 |
| 22 | 89.59 | 248.37 | 32.49 | 4.44 | 1,026.83 |
| 23 | 90.31 | 248.20 | 32.84 | 2.89 | 1,026.83 |
| 24 | 89.44 | 248.45 | 32.87 | 2.87 | 1,026.83 |
| 25 | 88.91 | 248.60 | 32.51 | 2.46 | 1,026.83 |
| 26 | 88.60 | 248.70 | 32.87 | 1.93 | 1,026.83 |
| 27 | 88.55 | 248.74 | 32.01 | 0.83 | 1,026.83 |
| 28 | 88.16 | 248.85 | 31.90 | 0.85 | 1,026.83 |
| 29 | 87.74 | 248.95 | 32.47 | 0.15 | 1,026.83 |
| 30 | 87.94 | 248.92 | 32.57 | 0.00 | 1,026.83 |

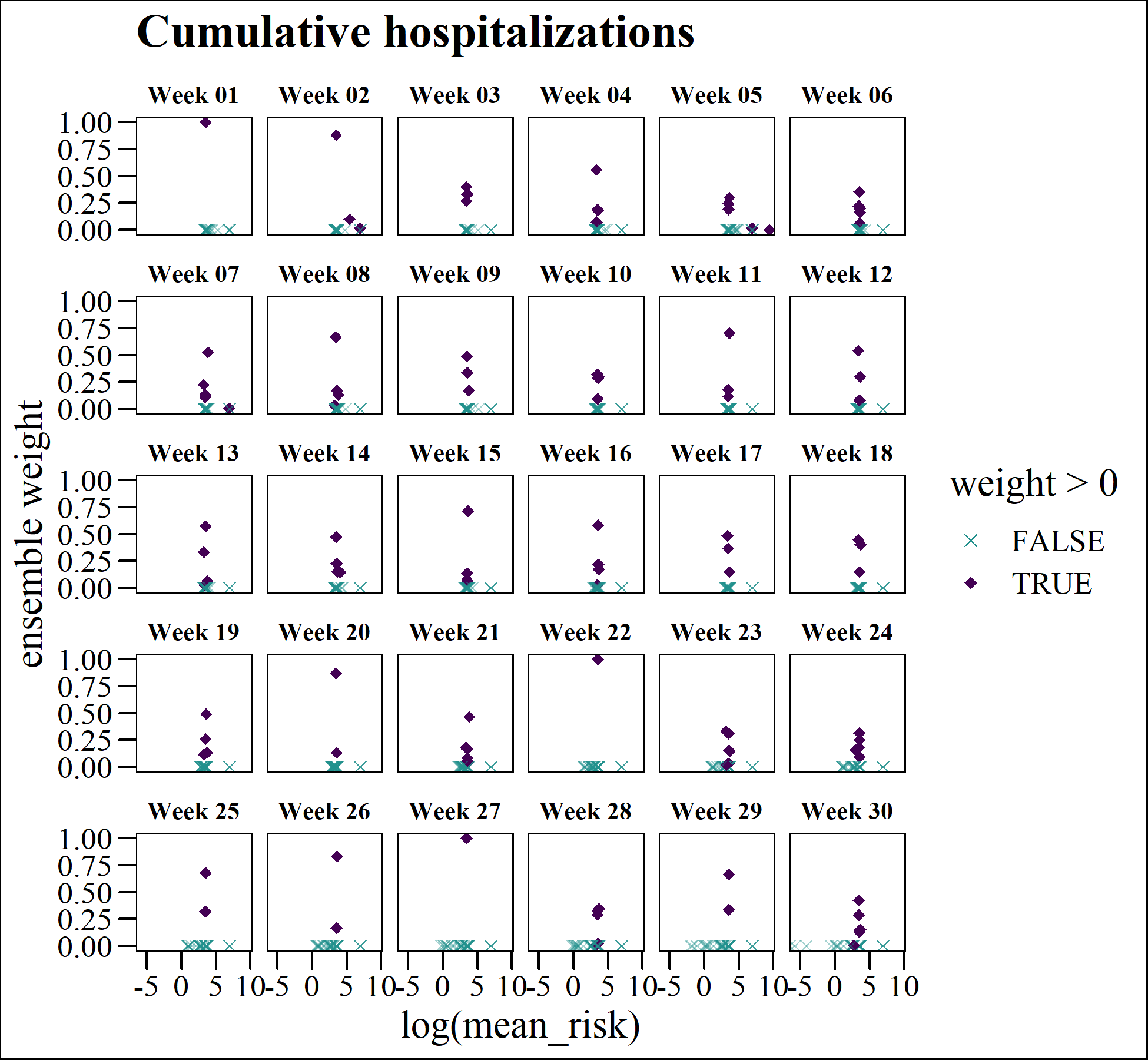
**S7 Figure. Peak rate, ensemble learner weights as a function of log mean cross-validated risk.** Ensemble learner weights are assigned by regressing the prediction target on corresponding predictions made by each component learner. The coefficients estimated for each independent predictor (i.e., component learner) represent the weight given to the learner’s predictions in the final ensemble super learner prediction.



**S8 Figure. Peak week, ensemble learner weights as a function of log mean cross-validated risk.** Ensemble learner weights are assigned by regressing the prediction target on corresponding predictions made by each component learner. The coefficients estimated for each independent predictor (i.e., component learner) represent the weight given to the learner’s predictions in the final ensemble super learner prediction.



**S9 Figure. Cumulative hospitalizations, ensemble learner weights as a function of log mean cross-validated risk.** Ensemble learner weights are assigned by regressing the prediction target on corresponding predictions made by each component learner. The coefficients estimated for each independent predictor (i.e., component learner) represent the weight given to the learner’s predictions in the final ensemble super learner prediction.



# References

1. Centers for Disease Control and Prevention. Burden of influenza. <https://www.cdc.gov/flu/about/burden/index.html>; 2020.

2. Lutz CS, Huynh MP, Schroeder M, Anyatonwu S, Dahlgren FS, Danyluk G, et al. Applying infectious disease forecasting to public health: A path forward using influenza forecasting examples. BMC Public Health. 2019;19: 1659. doi:[10.1186/s12889-019-7966-8](https://doi.org/10.1186/s12889-019-7966-8)

3. Reich NG, Brooks LC, Fox SJ, Kandula S, McGowan CJ, Moore E, et al. A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the united states. Proc Natl Acad Sci U S A. 2019;116: 3146–3154. doi:[10.1073/pnas.1812594116](https://doi.org/10.1073/pnas.1812594116)

4. Chretien J-P, George D, Shaman J, Chitale RA, McKenzie FE. Influenza forecasting in human populations: A scoping review. PLoS One. 2014;9: e94130. doi:[10.1371/journal.pone.0094130](https://doi.org/10.1371/journal.pone.0094130)

5. Biggerstaff M, Kniss K, Jernigan DB, Brammer L, Bresee J, Garg S, et al. Systematic assessment of multiple routine and near Real-Time indicators to classify the severity of influenza seasons and pandemics in the united states, 2003-2004 through 2015-2016. Am J Epidemiol. 2018;187: 1040–1050. doi:[10.1093/aje/kwx334](https://doi.org/10.1093/aje/kwx334)

6. Brooks LC, Farrow DC, Hyun S, Tibshirani RJ, Rosenfeld R. Flexible modeling of epidemics with an empirical bayes framework. PLoS Comput Biol. 2015;11: e1004382. doi:[10.1371/journal.pcbi.1004382](https://doi.org/10.1371/journal.pcbi.1004382)

7. Reich NG, McGowan CJ, Yamana TK, Tushar A, Ray EL, Osthus D, et al. Accuracy of real-time multi-model ensemble forecasts for seasonal influenza in the U.S. PLoS Comput Biol. 2019;15: e1007486. doi:[10.1371/journal.pcbi.1007486](https://doi.org/10.1371/journal.pcbi.1007486)

8. Ray EL, Reich NG. Prediction of infectious disease epidemics via weighted density ensembles. PLoS Comput Biol. 2018;14: e1005910. doi:[10.1371/journal.pcbi.1005910](https://doi.org/10.1371/journal.pcbi.1005910)

9. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: Data mining, inference, and prediction. Springer, New York, NY; 2009. doi:[10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7)

10. Wolpert DH. Stacked generalization. Neural Netw. 1992;5: 241–259. doi:[10.1016/S0893-6080(05)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)

11. Breiman L. Stacked regressions. Mach Learn. 1996;24: 49–64. doi:[10.1007/BF00117832](https://doi.org/10.1007/BF00117832)

12. Laan MJ van der, Polley EC, Hubbard AE. Super learner. Stat Appl Genet Mol Biol. De Gruyter; 2007;6: Article25. doi:[10.2202/1544-6115.1309](https://doi.org/10.2202/1544-6115.1309)

13. McGowan CJ, Biggerstaff M, Johansson M, Apfeldorf KM, Ben-Nun M, Brooks L, et al. Collaborative efforts to forecast seasonal influenza in the united states, 2015-2016. Sci Rep. nature.com; 2019;9: 683. doi:[10.1038/s41598-018-36361-9](https://doi.org/10.1038/s41598-018-36361-9)

14. Kandula S, Yamana T, Pei S, Yang W, Morita H, Shaman J. Evaluation of mechanistic and statistical methods in forecasting influenza-like illness. J R Soc Interface. 2018;15. doi:[10.1098/rsif.2018.0174](https://doi.org/10.1098/rsif.2018.0174)

15. Kandula S, Pei S, Shaman J. Improved forecasts of influenza-associated hospitalization rates with google search trends. J R Soc Interface. 2019;16: 20190080. doi:[10.1098/rsif.2019.0080](https://doi.org/10.1098/rsif.2019.0080)

16. Polley EC, Laan MJ van der. Super learner in prediction. University of California, Berkeley; 2010.

17. Polley EC, Rose S, Laan MJ van der. Super learning. In: Laan MJ van der, Rose S, editors. Targeted learning: Causal inference for observational and experimental data. New York, NY: Springer New York; 2011. pp. 43–66. doi:[10.1007/978-1-4419-9782-1\\_3](https://doi.org/10.1007/978-1-4419-9782-1\_3)

18. Polley E, LeDell E, Kennedy C, van der Laan M. SuperLearner: Super learner prediction [Internet]. 2019. Available: <https://CRAN.R-project.org/package=SuperLearner>

19. Coyle JR, Hejazi NS, Malenica I, Sofrygin O. Sl3: Pipelines for machine learning and super learning. 2020. doi:[10.5281/zenodo.1342293](https://doi.org/10.5281/zenodo.1342293)

20. Centers for Disease Control and Prevention. Epidemic prediction initiative. <https://predict.cdc.gov/post/59973fe26f7559750d84a843>;

21. Centers for Disease Control and Prevention. Flu view phase 3 quick reference guide [Internet]. Available: <https://gis.cdc.gov/GRASP/Fluview/FluHospRates.html>

22. Centers for Disease Control and Prevention. MMWR week overview. <https://wwwn.cdc.gov/nndss/document/MMWR_Week_overview.pdf>;

23. Kim S, Koh K, Boyd S, Gorinevsky D. 1 trend filtering. SIAM Rev. Society for Industrial; Applied Mathematics; 2009;51: 339–360. doi:[10.1137/070690274](https://doi.org/10.1137/070690274)

24. Tibshirani RJ. Adaptive piecewise polynomial estimation via trend filtering. Ann Stat. Institute of Mathematical Statistics; 2014;42: 285–323. doi:[10.1214/13-AOS1189](https://doi.org/10.1214/13-AOS1189)

25. Arnold T, Sadhanala V, Tibshirani R. Glmgen. 2015.

26. R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2020. Available: <https://www.R-project.org/>

27. Naimi AI, Balzer LB. Stacked generalization: An introduction to super learning. Eur J Epidemiol. 2018;33: 459–464. doi:[10.1007/s10654-018-0390-z](https://doi.org/10.1007/s10654-018-0390-z)

28. Harrell FE Jr. Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis. Springer International Publishing; 2015. doi:[10.1007/978-3-319-19425-7](https://doi.org/10.1007/978-3-319-19425-7)

29. Wood S. Mgcv: Mixed GAM computation vehicle with automatic smoothness estimation [Internet]. 2019. Available: <https://cran.r-project.org/web/packages/mgcv/index.html>

30. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological). 1996. pp. 267–288. doi:[10.1111/j.2517-6161.1996.tb02080.x](https://doi.org/10.1111/j.2517-6161.1996.tb02080.x)

31. Breiman L. Random forests. Mach Learn. 2001;45: 5–32. doi:[10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324)

32. Kooperberg C, Moler C, Dongarra J. Polspline: Polynomial spline routines [Internet]. 2019. Available: <https://cran.r-project.org/web/packages/polspline/index.html>

33. Friedman J, Hastie T, Tibshirani R, Narasimhan B, Simon N, Qian J. Glmnet: Lasso and Elastic-Net regularized generalized linear models [Internet]. 2019. Available: <https://cran.r-project.org/web/packages/glmnet/index.html>

34. Liaw A, Wiener M. Breiman and cutler’s random forests for classification and regression [Internet]. 2018. Available: <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>

35. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F, Chang C-C, et al. E1071 [Internet]. 2019. Available: <https://cran.r-project.org/web/packages/e1071/e1071.pdf>

36. Ripley B, Venables W. Nnet [Internet]. 2020. Available: <https://cran.r-project.org/web/packages/nnet/nnet.pdf>

37. Coyle J, Hejazi N. Delayed: A framework for parallelizing dependent tasks [Internet]. 2020. Available: <https://cran.r-project.org/web/packages/delayed/index.html>

38. Bengtsson H. Future: Unified parallel and distributed processing in R for everyone [Internet]. 2020. Available: <https://cran.r-project.org/web/packages/future/index.html>

39. Yamada H, Yoon G. Selecting the tuning parameter of the trend filter. Stud Nonlinear Dyn Econom. 2016;20: 1. doi:[10.1515/snde-2014-0089](https://doi.org/10.1515/snde-2014-0089)

40. Laan MJ van der, Benkeser D. Online super learning. In: Laan MJ van der, Rose S, editors. Targeted learning in data science: Causal inference for complex longitudinal studies. Cham: Springer International Publishing; 2018. pp. 303–315. doi:[10.1007/978-3-319-65304-4\\_18](https://doi.org/10.1007/978-3-319-65304-4\_18)