PLANES: Plausibility Analysis of Epidemiological Signals

VP Nagraj Signature Science, LLC
Amy E. Benefield Signature Science, LLC
Desiree Williams Signature Science, LLC
Stephen D. Turner Signature Science, LLC

August 22, 2024

Abstract

Methods for reviewing epidemiological signals are necessary to building and maintaining data-driven public health capabilities. We have developed a novel approach for assessing the plausibility of infectious disease forecasts and surveillance data. The PLANES (**PL**ausibility **AN**alysis of **E**pidemiological **S**ignals) methodology is designed to be multi-dimensional and flexible, yielding an overall score based on individual component assessments that can be applied at various temporal and spatial granularities. Here we describe PLANES, provide a demonstration analysis, and discuss how to use the open-source rplanes R package. PLANES aims to enable modelers and public health end-users to evaluate forecast plausibility and surveillance data integrity, ultimately improving early warning systems and informing evidence-based decision-making.

1 Introduction

Near-term forecasts and long-term projections of infectious disease targets are critical to effective public health communication, decision-making, and resource allocation. Review of surveillance data and model output prior to dissemination is paramount for engendering trust in model-based policy and public health decision-making. Real-time efforts to model Ebola (King et al. 2015), COVID-19 (Cramer et al. 2022), and influenza (Mathis et al. 2024) have shown that even well-calibrated methods may yield implausible trajectories for certain targets. For consortia that openly solicit forecast contributions, implausible submissions could bias ensembles so as to misguide policymaking or erode confidence in public health communication. Likewise, some surveillance systems may suffer from delayed or temporarily faulty reporting mechanisms. Assessing plausibility of surveillance data in real-time could mitigate the impact of data integrity issues on forecasting efforts and identify systematic problems to fix, both of which would contribute to more effective early warning systems. However, there are currently no codified plausibility heuristics, and if any plausibility assessment of forecasts or surveillance signals is performed at all, it is typically *ad hoc* and results are not broadly disseminated (Nagraj et al. 2021, 2022).

To address the gap in methodology for reviewing epidemiological signals, we developed a novel approach, PLANES (Plausibility Analysis of Epidemiological Signals), for flexible, multi-dimensional assessment of near-term forecasts and surveillance data integrity. We envisioned the PLANES algorithm yielding an overall score based on individual component assessments. The PLANES scoring mechanism was designed to be agnostic to temporal and spatial granularity. We delivered the PLANES methodology via an open-source software package such that the approach could be readily operationalized by modelers and public health end-users working across federal, state, territorial, local, and tribal jurisdictions.

2 Methods

2.1 PLANES algorithm

Assessing whether a reported or forecasted value is plausible requires a baseline against which the signal can be compared. We designed the PLANES algorithm to use historical observed data deemed trustworthy to create initial "seed" characteristics to assess plausibility downstream. Each seed characteristic is location-specific such that plausibility analyses independently assess multiple locations in the forecast or surveillance data. One of the priorities in designing the PLANES methodology was the inclusion of multiple components. There are existing approaches for one-dimensional anomaly detection (Dancho and Vaughan 2023), and we therefore sought to develop an approach to examine multiple features of the data. The algorithm uses the relevant seed characteristics for assessments of individual components, each of which deliver a binary determination of whether the data appear implausible. Collectively, the binary assessments can be summarized to deliver an ordinal score.

It is important to note that the scoring system we envisioned would work for both forecasts and observed signals. However, we anticipated that certain components may include characteristics that would only be applicable to forecasted signals. We also expected that the presence of flags raised for certain components may be more or less of interest for specific use-cases, and therefore aimed to accommodate a weighting scheme to allow users to modulate the impact of components in the overall score. Figure A1 demonstrates the concept behind the algorithm, with illustrations of notional components and impacts.

2.2 R package

To deliver the PLANES algorithm, we developed an open-source R package (R Core Team 2023). The package, `rplanes`, was scoped to include functions to prepare and format the data for analysis, create the background seed characteristics, and run the plausibility analysis. We aimed to make the package as user-friendly as possible, with helpers to intuitively prepare data and a wrapper function to run all individual components. As part of the package, we also planned to deliver an interface to translate the programmatic API to point-and-click features via Shiny (Chang et al. 2021).

2.3 Demonstration analysis

After implementing PLANES in the `rplanes` package, we demonstrated how the approach can be used operationally by applying the plausibility scoring on real-world data. For this effort we used the 2022-23 FluSight forecasts[1]. We retrospectively masked the available weekly observed data to generate the background seed characteristics. We then iteratively ran the plausibility scoring to create a distribution of PLANES scores across the time span and signals analyzed. All weekly plausibility analyses were stratified by location and submitting forecaster.

Beyond a demonstration of the package features, our analysis aimed to 1) verify that the `rplanes` functionally worked when using operational data, 2) assess the sensitivity of the components in the algorithm (i.e., distribution of how many flags were raised), and 3) evaluate the correlation between PLANES and forecast performance. For the functional testing, we set out to confirm that the `rplanes` functions could be used without any errors during processing. When assessing the sensitivity of the PLANES algorithm, we aggregated all scores generated and counted how many times each score was observed. To correlate the PLANES scores with forecast performance, we used the weighted interval score (WIS) to estimate the accuracy of predictions (Bracher et al. 2021). As with the PLANES scores, the WIS was calculated across strata of forecast week, location, and forecaster. The WIS was further stratified by forecast horizon, but for this analysis we computed a mean WIS across all horizons to align with the resolution of the PLANES score.

2.4 Data

2.4.1 FluSight forecasts

To assess the plausibility of a forecast signal, we analyzed forecast submissions to the FluSight hub from the 2022-23 season[2]. During this season, the FluSight coordinators solicited weekly forecasts of incident flu hospitalizations in the United States. Participating teams were allowed to submit forecasts using multiple

[1] `https://www.cdc.gov/flu-forecasting/about/index.html`
[2] `https://github.com/cdcepi/FluSight-forecast-data`

models. Here we refer to each combination of team and model as a forecaster. Submissions included required metadata for the team name, contact information, model designation (i.e., "primary", "secondary", "proposed", or "other"), and licenses for forecasts. For this analysis we excluded forecasters that were designated as proposed or other, as well as any forecasters that had an ambiguous or non-permissive license. All forecasts were retrieved from the public GitHub repository for the challenge.

3 Results

3.1 Components

We developed a set of components to assess plausibility of epidemiological signals. Individually, each component provides a binary assessment (i.e., yes/no is the signal implausible). All evaluated components are then combined into an ordinal score. By default, each component is equally weighted in the overall PLANES score. When delivered in the `rplanes` R package, the user can optionally weight components higher or lower in the scoring scheme. What follows is a description of the characteristics assessed and methods used for each of the seven PLANES components we implemented. For each component we describe the internal logic and basic motivation. Examples of each component are visually depicted and detailed in equations where applicable (see Appendix).

3.1.1 Difference

The difference component checks the magnitude of point-to-point differences for all time steps of the evaluated data. This component can be used on either forecasts or observed signals. If an evaluated signal departs from the prior observation more dramatically than has been seen previously in the time series, then it is flagged as implausible. The function internally computes the maximum observed difference (based on absolute value) to set a threshold, which if exceeded will trigger a flag to be raised by the algorithm. While large and unexpected point-to-point changes may naturally occur in epidemiological signals, this component provides a means to draw attention to the most extreme cases.

3.1.2 Cover

The coverage component compares the prediction interval for the first horizon of the evaluated signal to the most recent value in the seed. If the interval does not cover the most recent data point, then the flag is raised as implausible. The width of the interval used for this evaluation can be customized by the user when preparing the signal data (see the `rplanes` section below for more details on the API). The narrower the width of the prediction interval, the more sensitive this component will be. This component is motivated by an expectation that the prediction interval for a poorly calibrated forecast may exhibit immediate departure from the most recent historical signal (i.e., in the first horizon), and that such implausibility would manifest in the prediction intervals as well. Note that because this component requires a prediction interval, it can only be used to assess plausibility of forecast signals.

3.1.3 Taper

The taper component checks whether the prediction interval for the evaluated signal decreases in width (i.e., certainty increases) as horizons progress. Because this component requires a prediction interval, it can only be used to assess plausibility of forecast signals. The width of the prediction interval at every horizon is assessed against the previous horizon and if any of the intervals for the earlier horizon is wider a flag is raised. One would expect that there would be more variability in signals forecasted further out in time, and therefore the prediction interval would be wider in later horizons. The goal of this component is to assess this phenomenon and to flag situations where signals exhibit behavior that counters this expectation.

3.1.4 Repeat

The repeat component checks whether consecutive values in an observed or forecasted signal are repeated more than the tolerated number of times (k). When the seed is created, it stores the maximum number of consecutive repeats for each location and uses this as the default value for k. If the evaluated data exceeds k, then the signal is considered implausible and a flag is raised. Here a repeat is defined as two or more consecutive values that are exactly equivalent. This definition means that the component may be most informative for signals that communicate counts (e.g., number of positive influenza cases) instead of other measures (e.g., percentage of influenza-like illness visits). By default the length of the values used to check

repeats is the number of evaluated horizons plus a prepend length, which is set to the defined maximum number of repeats. The tolerance and prepend parameters can be overridden by the user in the `rplanes` package.

3.1.5 Trend

The trend component assesses if there is a significant change in the magnitude or direction of the slope for the evaluated signal compared to the most recent data in the seed. Each "change point" in the signal is identified using a hierarchical divisive algorithm originally implemented in the `ecp` R package (Nicholas A. James, Wenyu Zhang, and David S. Matteson 2019). The input for the algorithm is the lagged difference of all points in a combined time series that concatenates every value in the observed and evaluated signals. All change points in the time series are identified, but a flag is only raised if there is a change point in any of the evaluated horizons or the most recent observed value. The analysis requires at least four times as many observed values as there are evaluated values. The trend component is only available for forecast signals.

As noted above, the change point analysis uses the `ecp` package, and specifically the `e.divisive()` function. The methods for the change point detection have been previously published (Nicholas A. James and David S. Matteson 2014). In brief, change points are identified based on distances between segments, with larger distances indicating higher likelihood of a change point. Internally, the function uses a permutation test to calculate an approximate p-value that is used in hypothesis testing. Only statistically significant change points are flagged, and the significance level (α) can be customized to the use-case. A higher value for α will decrease the sensitivity of the change point detection and therefore reduce the number of trend flags raised.

3.1.6 Shape

While the trend component scans the time series for an inflection point, the shape component assesses the time series for unusual shapes across multiple points. To arrive at the shape assessment, the algorithm first divides the observed seed data into sliding windows to form trajectories. The trajectories are summarized as a set of shapes against which the forecasted trajectory is compared. If the shape of the forecasted trajectory does not match any shapes in the seed data, then the forecast is considered implausible per this component. The core intuition underlying this component is that the shape of future data is more likely to reflect patterns that have previously been observed and less likely to be a novel trajectory. Therefore, it may be useful to flag any novel shapes for review.

We developed two methods for summarizing the shapes of signal trajectories. The first method uses a dynamic time warping (DTW) technique to return the Euclidean distance between sets of consecutive values. Each set is constructed by sliding across the time series of observed values for the given location in the seed data. The size of each window (i.e., length of sliding time series) is fixed to equal the number of horizons in the forecast to be evaluated. DTW methods and applications have been described in detail previously (Tormene et al. 2009). We used DTW as implemented in the `dtw` R package and the `dtwDist()` function with default parameters (Giorgino 2009). Our algorithm finds the minimum distances for each window, and the maximum of the minimum distances serves as a threshold. The algorithm then calculates the DTW distances between the forecast signal and every observed sliding window. Note that as part of this procedure, the algorithm builds trajectories for the forecast point estimates as well as upper and lower bounds of the prediction interval. If the distance between the forecast trajectories and any observed sliding window is less than or equal to the threshold defined above, then this shape is not considered novel.

The distance calculations in the DTW approach can be time consuming, especially as the number of observations in the seed data increases. To mitigate the computational expense, we developed an alternative approach for identifying shapes. This method uses differences of consecutive observations to construct trajectories. Each point-to-point difference is computed and then centered and scaled by standard deviation around the mean of all differences. As with DTW, we define the trajectories in sliding windows. For scaled differences, we further categorize each difference as an "increase", "decrease", or "stable" change. The threshold for increase or decrease is a difference of one standard deviation in the respective direction. Collectively, the categorical summaries of the differences within the given window form a shape (e.g., "increase;stable;stable;decrease"). We then assess categorical changes for forecasts and compare to the set of observed shapes. If the shape is novel, then a flag is raised for implausibility. Given its computational efficiency compared to DTW, the scaled difference method is set as the default in `rplanes`.

3.1.7 Zero

The zero component was designed to check if there are any "sudden" zeros in the evaluated signal. Whether it is a broken surveillance instrument or miscalibrated forecast, we expect it would be unlikely to observe a zero if it has never been reported in the seed data. This algorithm first identifies whether any values in the evaluated signal are equal to zero. If zeros are found, it examines the seed for the presence of any zeros. If zeroes exist in the seed, the function determines that the evaluated zero is plausible. However, if no zeroes are present in the seed, the function deems the evaluated zero implausible.

3.2 rplanes

The PLANES algorithm is implemented in the `rplanes` R package. The package is published under an open-source license and is available on GitHub[3] and the Comprehensive R Archive Network (CRAN)[4]. The package changelog, function documentation, and narrative user guides are delivered in a publicly available website[5]. The package API is designed to provide an intuitive and efficient set of functions to prepare and run the PLANES analysis. The entire PLANES analysis workflow is depicted in Figure 1.

At a high level, there are three steps to the PLANES analysis with `rplanes`. First, the user determines if the analysis will be assessing an observed or forecasted signal. Both options are available in the `to_signal()` constructor function, which creates an S3 object with a primary class "signal" and a secondary class corresponding to the type of signal specified (i.e., "observed" or "forecast"). Inputs for `to_signal()` must be provided as a data frame, with data at daily, weekly, or monthly resolution. The input data frame requires that specific features are present depending on whether it contains observed or forecast data. For observed signals, it should include columns for location (geographic unit such as FIPS code) and date (date of reported value), along with an outcome column. For forecast signals, the data frame should include columns for location, date (corresponding to the forecast horizon), horizon, lower and upper limits of the prediction interval, and point estimates. Note that `rplanes` includes a helper function called `read_forecast()` to convert the quantile format that has been standardized in disease forecasting hubs directly from a file. Downstream analysis functions have select data validation checks in place and will issue warnings or errors for incomplete or incompatible data. However, data prepared as a signal object should be cleaned (e.g., location names disambiguated) and complete (e.g., free of large gaps) prior to using `to_signal()`.

Once the signal to be evaluated has been prepared, the user then needs to retrieve and prepare the observed data that will be used to generate baseline seed characteristics. This observed data should also be prepared with `to_signal()` before being passed to `plane_seed()`, which returns a named list with summarized characteristics for each of the locations in the dataset. If the user is evaluating an observed signal, then the same object for evaluation can be used for seeding, so long as the user specifies a cut date in `plane_seed()`.

With the signal to be evaluated and the seed prepared, the user can run a wrapper function to generate PLANES results. For convenience, the package includes a single function (`plane_score()`) that wraps individual plausibility analysis functions (e.g., `plane_shape()`, `plane_diff()`, etc.) and runs all components with equal weights by default. Users can optionally customize this behavior to exclude certain components or adjust the weight that individual components receive in the overall score. As described above, certain components only apply to forecast evaluation, and as such, those are automatically excluded if an observed signal is assessed with `plane_score()`.

Beyond the programmatic API, the `rplanes` package also provides a user interface (UI) to conduct PLANES analyses. The UI is delivered as a Shiny app within the package, which can be launched via the `rplanes_explorer()` function. Steps for preparing signal objects and seeding baseline characteristics are translated to point-and-click equivalents. The app also includes an option to use built-in example data to facilitate demonstrations and exploration of the analysis outputs. PLANES results are available in tabular format and as a series of data visualizations, including plots of specific components to visually investigate cases when plausibility flags are raised.

3.3 Demonstration analysis

We conducted a retrospective analysis of FluSight forecasts to demonstrate the PLANES method as delivered in the `rplanes` package with default arguments. After retrieving all FluSight submissions for the 2022-23

[3] `https://github.com/signaturescience/rplanes`
[4] `https://CRAN.R-project.org/package=rplanes`
[5] `https://signaturescience.github.io/rplanes/`

season, we found that 44 forecasters submitted at least one forecast between the first submission date (October 17, 2022) and final date (May 15, 2023). Of these, we evaluated 32 forecasters that were designated as either "primary" or "secondary" methods and had unambiguous, permissive licensing in the available metadata. When aggregated across all forecasters, forecast weeks, horizons, and locations, there were 120,539 forecasts analyzed. There were 53 combinations of locations, forecasters, and forecast weeks that did not include a forecast for the 4 week-ahead horizon. Additionally, there were 2,640 forecasts for Puerto Rico and Virgin Islands that were excluded prior to PLANES and WIS analysis. After aggregating across horizons, we had combined PLANES and WIS results for 29,488 forecasts.

In Figure 2, we present the number of times each component was flagged in the FluSight forecasts that were analyzed. Taper and shape were the most commonly flagged components, while difference and zero were the least common. Figure 3 shows the results of the plausibility analysis and forecast performance using the PLANES and WIS metrics respectively. The violin plots of WIS distribution are aggregated by number of PLANES components flagged in the given forecasts. The maximum number of flags raised for any given forecast was five. Roughly 40% (n=11,738) of forecasts had no flags raised, while about another 40% (n=11,750) had one flag raised. These forecasts had similar log-transformed median WIS values. As the number of PLANES flags increased from two through five, the median WIS appears to clearly step up in a linear pattern. After filtering for results with at least two PLANES components flagged, we found that PLANES and WIS were significantly correlated ($r=0.267$;$p<0.0001$).

4 Discussion

Epidemiological forecasting and reporting instruments are subject to occasionally implausible signals. Regardless of the underlying causes, having an efficient and interpretable method to review forecasts and surveillance data can help public health stakeholders address these data integrity issues in a timely manner. PLANES is a data-driven, multi-dimensional approach that was designed to fill this need. We have developed and delivered PLANES via the `rplanes` R package. Several groups have released methods and tools to identify data integrity issues via anomaly detection (Brooks et al. 2024; Eze et al. 2023). However, existing approaches are generally limited to investigating individual features of the time series. With PLANES we have aspired to incorporate multiple components that check for explicit plausibility expectations tailored to epidemiological signals. To our knowledge, this provides a novel method, which can readily be evaluated and incorporated into operational workflows using `rplanes`.

One of the key features of PLANES is that the algorithm is not fully automated. We have intentionally designed an approach that stops short of automatically intervening in data flagged as implausible (e.g., automatic censoring). Ultimately, the decision for action on a plausibility assessment will be context-specific, and may necessitate human review of flagged components. The goal of the PLANES tooling is to dramatically reduce the burden of this review, and to codify metrics for more standardized interpretation and communication around signal data integrity. The system we have developed is intentionally modular. We have created `rplanes` such that the existing components can be re-weighted or ignored altogether depending on the use-case. Furthermore, the API is built flexibly to potentially support new components in the future, with very minimal change to the user experience or interpretation of the PLANES scoring output.

In our analysis of FluSight forecasts we found that the PLANES score was correlated with forecast performance, with forecasts scored as more implausible being more inaccurate on average. This result demonstrates practical utility of the PLANES approach in forecasting endeavors. For consortia efforts, PLANES could be operationalized to add a weighting scheme for contributed forecasts. In this use-case, forecasts submitted with higher PLANES scores may be given a lower weight when creating the ensemble. Forecasters could also use plausibility scoring to guide review and potential censoring of their model output prior to submitting. PLANES is designed to improve forecasting performance before forecasts are even created. The integrity of surveillance data used to train models can contribute to the quality of downstream forecasts. The flexibility of the PLANES methods to analyze observed signals enables public health stakeholders to review surveillance data and improve instruments.

During the 2023-24 FluSight season we piloted `rplanes` for the forecast and observed signal use-cases. We established a threshold for PLANES scores based on the 2022-23 FluSight analysis. Any forecasts with at least two the PLANES components flagged were set aside for further human review. If our team deemed it necessary based on a combination of the scores and visual inspection, we removed forecasts for the given location from the weekly submission. Our forecasts were generated using models trained on flu admission data reported at the state level via the National Healthcare Safety Network (NHSN). During the 2023-24

season, the NHSN data were aggregated weekly at the state level. Using `rplanes` as a dependency, we built a custom API that queried the NHSN instrument and scored the data as an observed signal. Our API was automated such that it ran on a schedule and delivered an email with any flags that were raised for the weekly hospitalization data in each state. If the state-level reporting had at least one PLANES component flagged, then we manually reviewed the reported signal before fitting our models to confirm trust in the training data.

We see the PLANES methodology and tooling as filling an immediate need in the disease modeling and epidemiological surveillance domains. However, we anticipate that there are multiple ways in which the approach may evolve and improve over time. As noted above, the modularity of PLANES invites the addition of new components. While we expect they will apply to percentages and proportions, many of the current components were developed with count data in mind. We also acknowledge that while conceiving of PLANES, we were primarily focused on respiratory disease signals. While we have used the approach to explore select vector-borne disease signals, additional applications for this kind of data may reveal areas of improvement to the underlying algorithms. Likewise, we acknowledge that PLANES fundamentally depends on having trustworthy historical reporting from which you can create seed characteristics. This may limit use-cases for PLANES where historical data is unavailable or sparse. Our goal in codifying the PLANES methods and building the `rplanes` R package was to invite users to adopt this approach as they see fit. We expect that community input will identify limitations and new applications, and therefore we openly invite feedback.

5 Conclusion

We have developed a novel approach to conduct plausibility analysis of epidemiological signals. PLANES is delivered in `rplanes`, which is an open-source R package. In demonstration analyses, we have shown that the PLANES scoring system correlates with forecast performance. We anticipate that the methods and tooling described here will be useful to public health stakeholders, including disease modelers as well as other users and maintainers of epidemiological surveillance data systems.

6 Acknowledgements

The work described in this manuscript would not have been possible without open and collaborative efforts from multiple entities. We acknowledge the following groups: the Armed Forces Health Surveillance Division - Integrated Biosurveillance Branch (AFHSD-IB) for their willingness to provide continued feedback on the plausibility approach while piloting the rplanes R package; the CDC for coordinating FluSight and providing guidance, interpretation, and dissemination of forecast data; the Council of State and Territorial Epidemiologists (CSTE) for establishing collaborative networks through which forecasting groups can interact with one another and public health stakeholders; all participating teams in the FluSight network for their sustained contributions, innovative techniques, and commitment to openness through operational forecasting activities.

This work was supported in part by a subaward to Signature Science, LLC from CSTE via CDC Cooperative Agreement No. NU38OT000297.

7 Figures

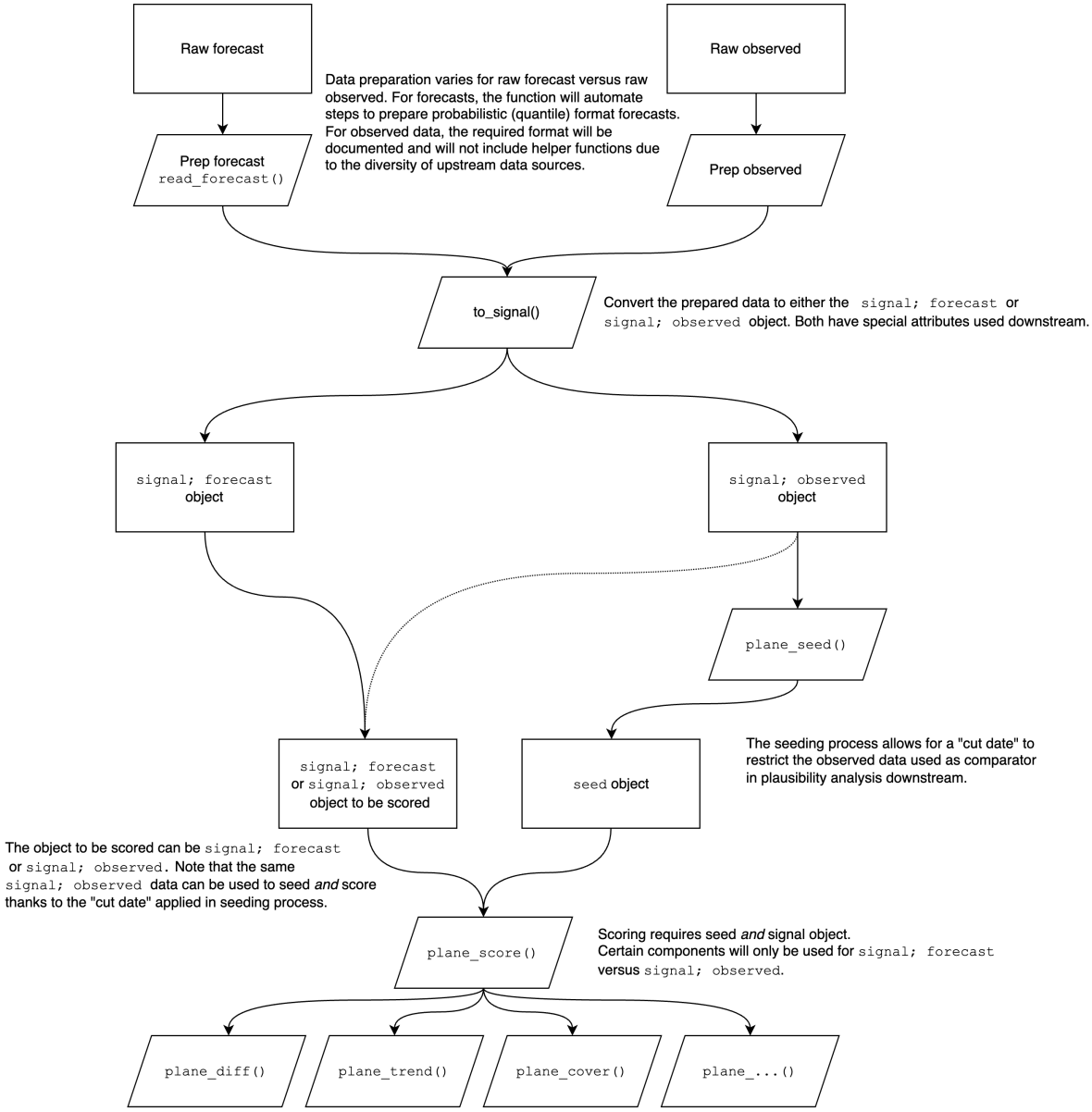

Figure 1: Workflow for the rplanes API. The diagram depicts the process for preparing and analyzing data with the available functions in the package. Users begin by preparing data to evaluate as well as data to seed background characteristics. These datasets begin as data frames and are coerced to 'signal' objects using rplanes helper functions. If the signal to be evaluated is observed data, then the same object used for the seed can be used in downstream analysis. If the signal is a forecast, then the user must prepare both an observed and forecast signal object for eventual scoring. The package also includes a function to build the seed using an observed signal. Given a seed and a signal to evaluate, the user can assess all components independently across all locations in the signal using a built-in wrapper function.

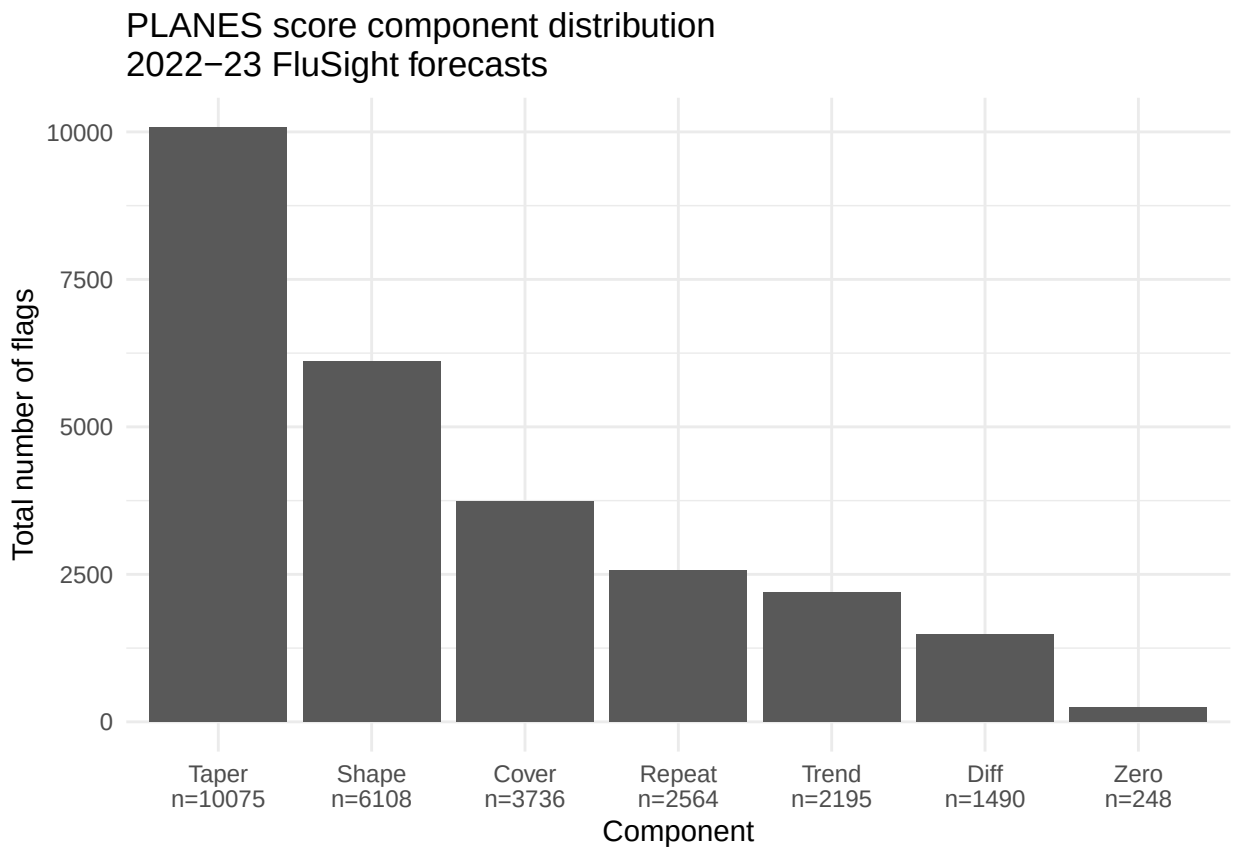

Figure 2: Distribution of components flagged in analzyed forecasts. The barplot shows the number of times each component was assessed as implausible in the 2022-23 FluSight forecast data. The most common components flagged were for the prediction interval tapering and novel shape, while the least common were for point-to-point difference and sudden zeros.

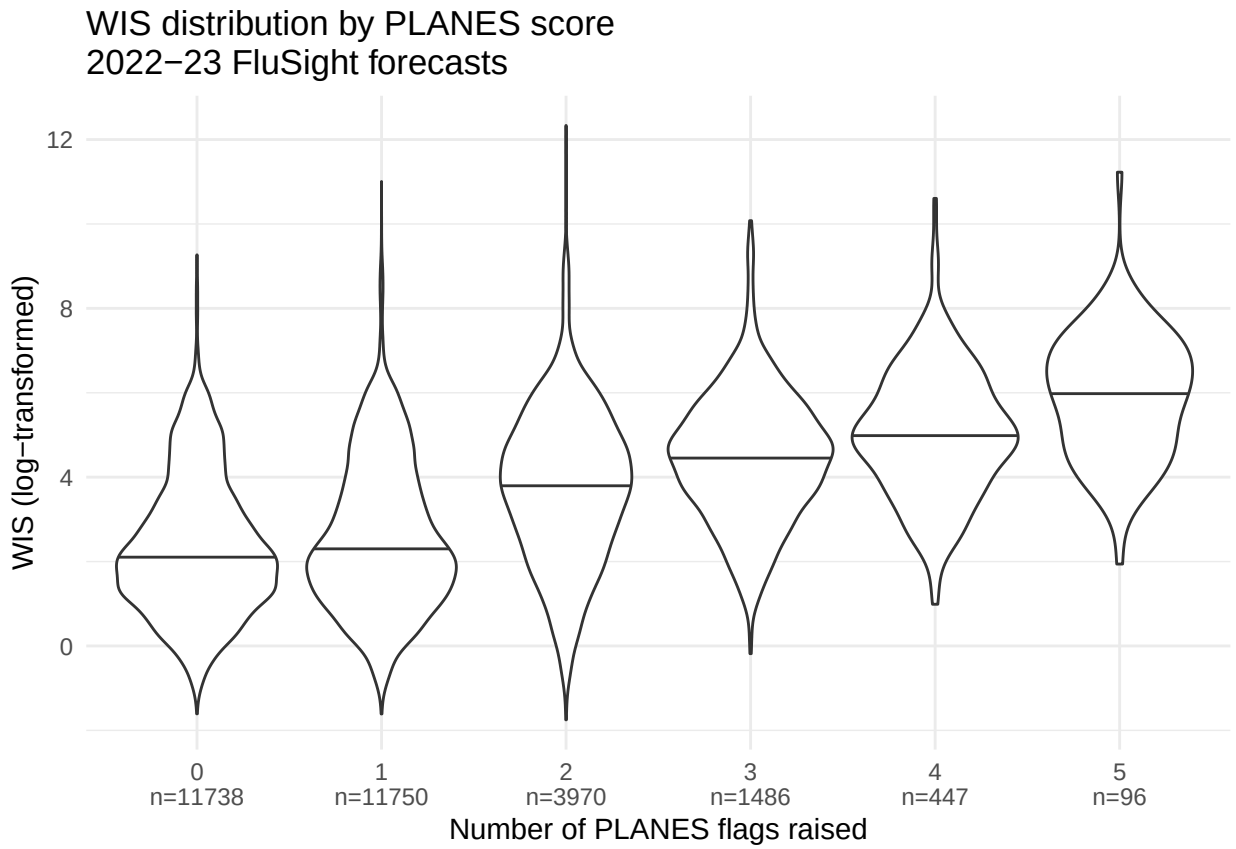

Figure 3: Relationship between WIS and PLANES scoring. The violin plot shows distribution of the WIS stratified by number of PLANES flags raised for the given forecast. Horizontal lines indicate median of the log-transformed WIS. The median performance of forecasts with zero or one component flagged was about equal. When forecasts had at least two PLANES flags raised, the median WIS increased (i.e., performance degraded) in a linear fashion.

8 References

Bracher, Johannes, Evan L. Ray, Tilmann Gneiting, and Nicholas G. Reich. 2021. "Evaluating Epidemic Forecasts in an Interval Format." Edited by Virginia E. Pitzer. *PLOS Computational Biology* 17 (2): e1008618. https://doi.org/10.1371/journal.pcbi.1008618.

Brooks, Logan, Daniel McDonald, Evan Ray, and Ryan Tibshirani. 2024. *Epiprocess: Tools for Basic Signal Processing in Epidemiology*. https://cmu-delphi.github.io/epiprocess/.

Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2021. *Shiny: Web Application Framework for r*. https://CRAN.R-project.org/package=shiny.

Cramer, Estee Y, Evan L Ray, Velma K Lopez, Johannes Bracher, Andrea Brennen, Alvaro J Castro Rivadeneira, Aaron Gerding, et al. 2022. "Evaluation of Individual and Ensemble Probabilistic Forecasts of COVID-19 Mortality in the United States." *Proc. Natl. Acad. Sci. U. S. A.* 119 (15): e2113561119.

Dancho, Matt, and Davis Vaughan. 2023. *Anomalize: Tidy Anomaly Detection*. https://CRAN.R-project.org/package=anomalize.

Eze, Peter U., Nicholas Geard, Ivo Mueller, and Iadine Chades. 2023. "Anomaly Detection in Endemic Disease Surveillance Data Using Machine Learning Techniques." *Healthcare (Basel, Switzerland)* 11 (13): 1896. https://doi.org/10.3390/healthcare11131896.

Giorgino, Toni. 2009. "Computing and Visualizing Dynamic Time Warping Alignments in r : The **Dtw** Package." *Journal of Statistical Software* 31 (7). https://doi.org/10.18637/jss.v031.i07.

King, Aaron A., Matthieu Domenech de Cellès, Felicia M. G. Magpantay, and Pejman Rohani. 2015. "Avoidable Errors in the Modelling of Outbreaks of Emerging Pathogens, with Special Reference to Ebola." *Proceedings. Biological Sciences* 282 (1806): 20150347. https://doi.org/10.1098/rspb.2015.0347.

Mathis, Sarabeth M., Alexander E. Webber, Tomás M. León, Erin L. Murray, Monica Sun, Lauren A. White, Logan C. Brooks, et al. 2024. "Title Evaluation of FluSight Influenza Forecasting in the 2021-22 and 2022-23 Seasons with a New Target Laboratory-Confirmed Influenza Hospitalizations." *Nature Communications* 15 (1): 6289. https://doi.org/10.1038/s41467-024-50601-9.

Nagraj, VP, Chris Hulme-Lowe, Stephanie L. Guertin, and Stephen D. Turner. 2021. "FOCUS: Forecasting COVID-19 in the United States." *medRxiv*. https://doi.org/10.1101/2021.05.18.21257386.

Nagraj, VP, Chris Hulme-Lowe, Shakeel Jessa, and Stephen D. Turner. 2022. "Automated Infectious Disease Forecasting: Use-Cases and Practical Considerations for Pipeline Implementation." https://arxiv.org/abs/2208.05019.

Nicholas A. James, and David S. Matteson. 2014. "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data." *Journal of Statistical Software* 62 (7): 1–25. https://www.jstatsoft.org/v62/i07/.

Nicholas A. James, Wenyu Zhang, and David S. Matteson. 2019. "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data_. R Package Version 3.1.4." https://cran.r-project.org/package=ecp.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Tormene, Paolo, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. 2009. "Matching Incomplete Time Series with Dynamic Time Warping: An Algorithm and an Application to Post-Stroke Rehabilitation." *Artificial Intelligence in Medicine* 45 (1): 11–34. https://doi.org/10.1016/j.artmed.2008.11.007.

9 Appendix

9.1 A1: PLANES concept

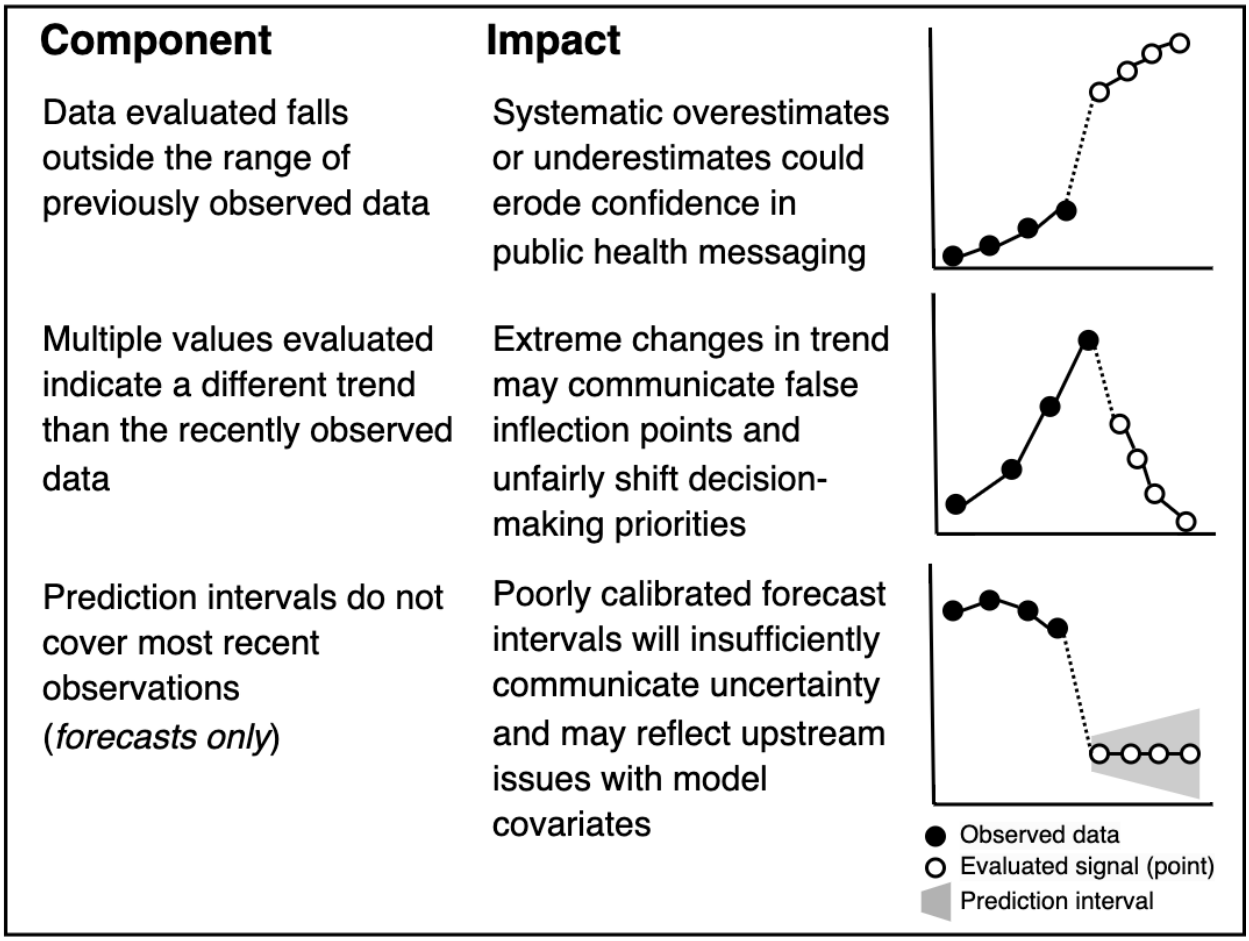

Figure A1: Conceptual motivation for developing the PLANES approach. Examples of plausibility components and their impacts are described. Each component is illustrated to show relationship between observed data and evaluated signal.

9.2 A2: Difference component

The formula in Equation 1 demonstrates how the maximum difference (μ) is computed, with X as the observed signal used to create the seed, t being the time step, and i representing the the number of steps in the time series.

$$\mu = max(|X_{t=1} - X_{t=0}|, |X_{t=2} - X_{t=1}|, ..., |X_{t=i} - X_{t=i-1}|) \tag{1}$$

For each horizon (h) in the time steps (j) for the evaluated signal (Y), μ is compared to the computed difference between h and $h-1$. As we specify in Equation 2, the algorithm checks if any of the differences exceed μ.

$$any(|Y_{h=1} - X_{t=i}| > \mu, |Y_{h=2} - Y_{h=1}| > \mu, ..., |Y_{h=j} - Y_{h=j-1}| > \mu) \tag{2}$$

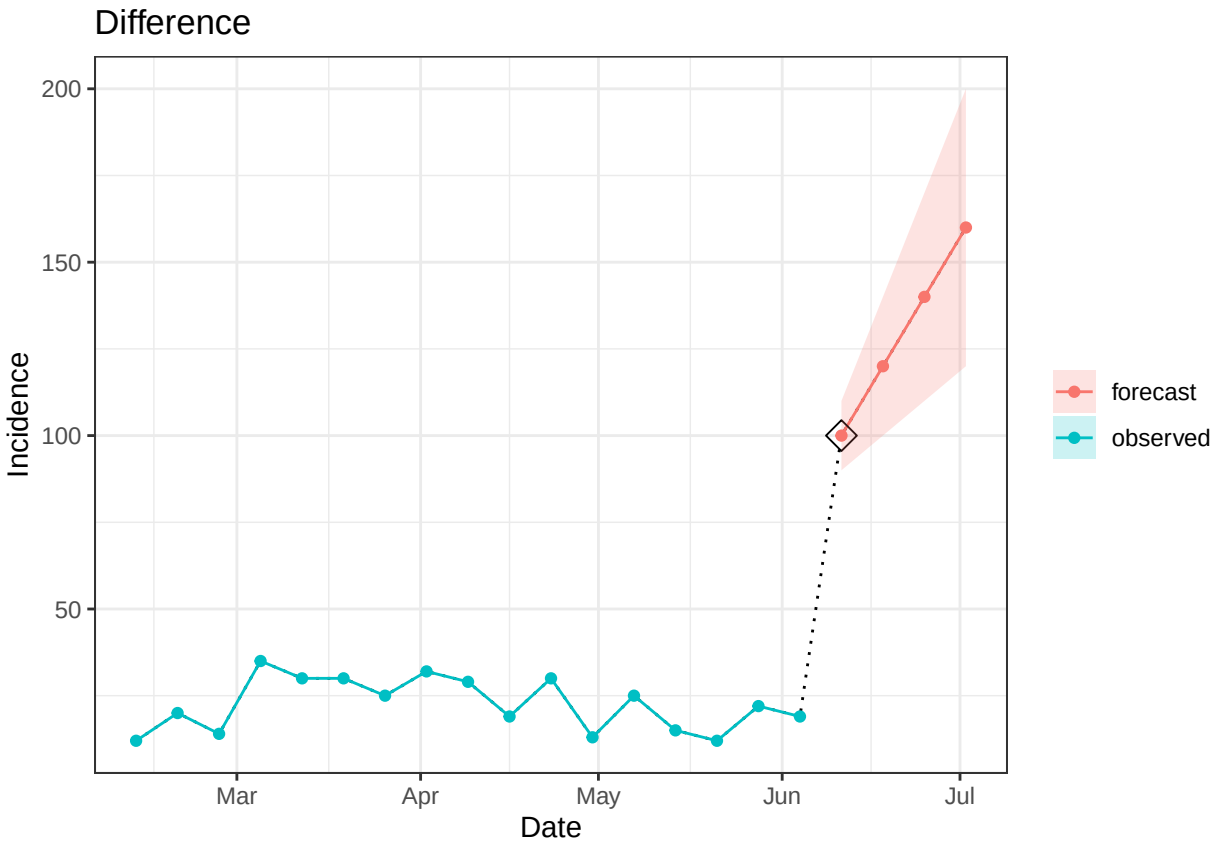

Figure A2: Depiction of a flag raised with the difference component. The difference component checks the point-to-point differences for evaluated signal. This component can be used on either forecasts or observed signals. The function internally computes the maximum observed difference (using absolute value) and checks to see if any of the point-to-point differences for the evaluated data exceed that threshold.

9.3 A3: Coverage component

Equation 3 specifies the formula for the coverage component, with the minimum and maximum of the forecasted prediction interval (ϕ) at the first horizon assessed against the last observation of the observed signal.

$$X_{t=i} < min(\phi_{h=1}) \vee X_{t=i} > max(\phi_{h=1}) \tag{3}$$

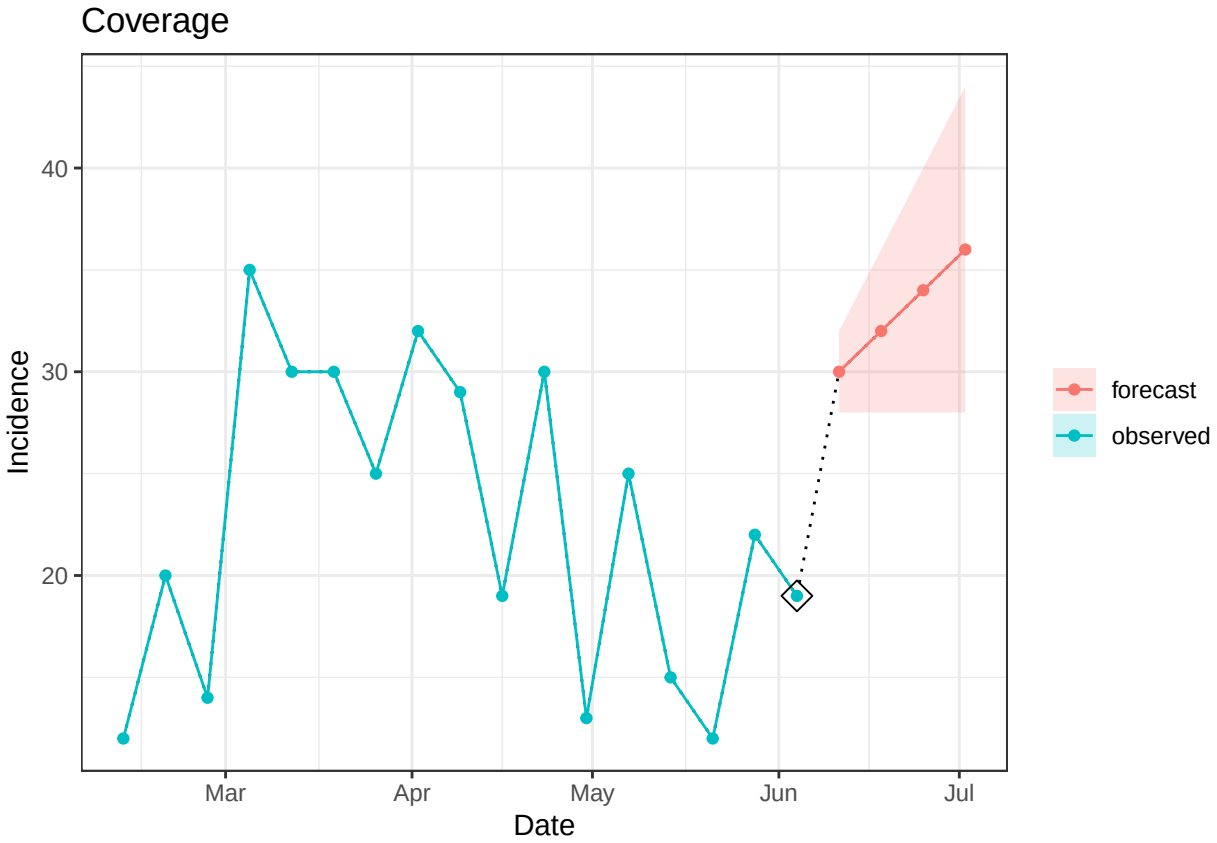

Figure A3: Depiction of a flag raised with the coverage component. The coverage component compares the prediction interval for the first horizon of the evaluated signal to the most recent value in the seed. If the interval does not cover the most recent data point, then the flag is raised as implausible. Because this component requires a prediction interval, it can only be used to assess plausibility of forecast signals.

9.4 A4: Taper component

Equation 4 describes the algorithm for comparing the width of prediction interval (ϕ) at each horizon (h) to the corresponding width at every consecutive time step (j).

$$any(max(\phi_{h=2}) - min(\phi_{h=2}) < max(\phi_{h=1}) - min(\phi_{h=1}), ..., max(\phi_{h=j}) - min(\phi_{h=j}) < max(\phi_{h=j-1}) - min(\phi_{h=j-1})) \quad (4)$$

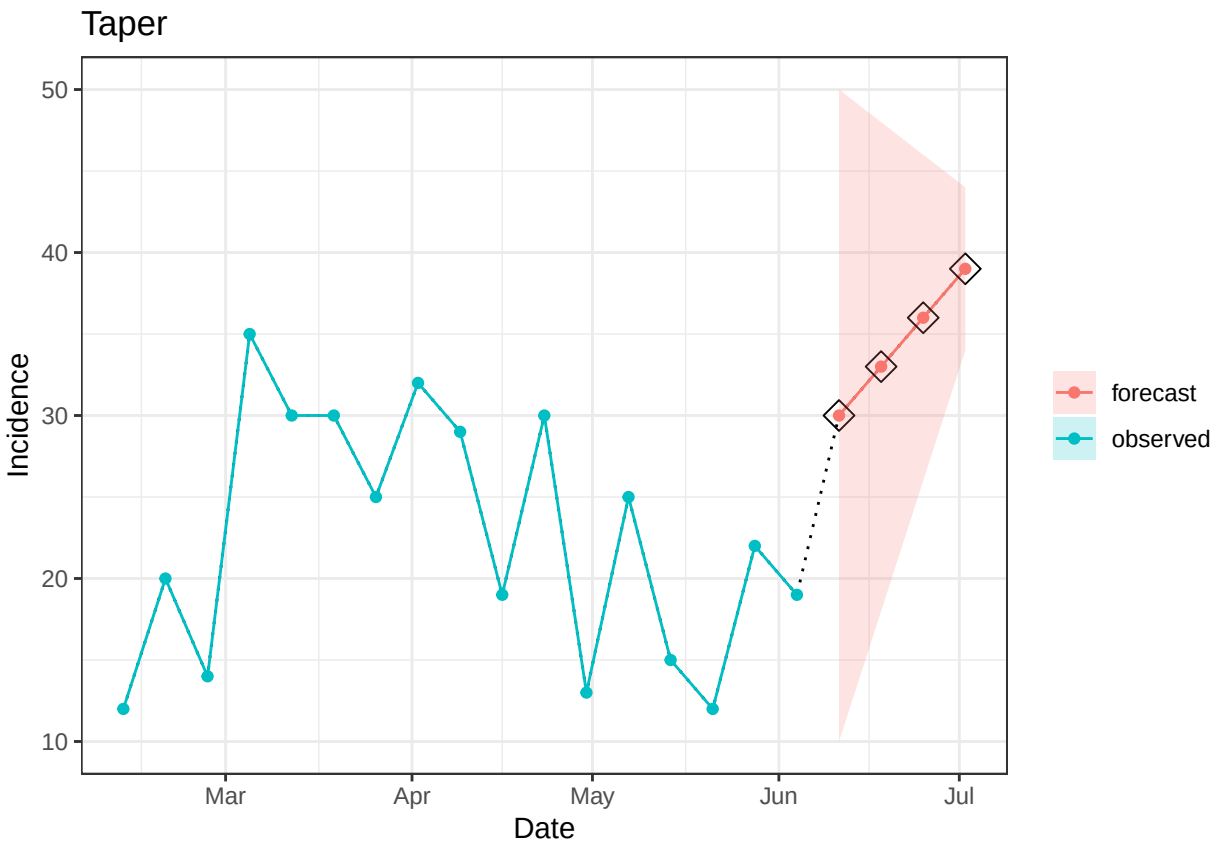

Figure A4: Depiction of a flag raised with the taper component. The taper component checks whether the prediction interval for the evaluated signal decreases in width (i.e., certainty increases) as horizons progress. Because this component requires a prediction interval, it can only be used to assess plausibility of forecast signals.

9.5 A5: Repeat component

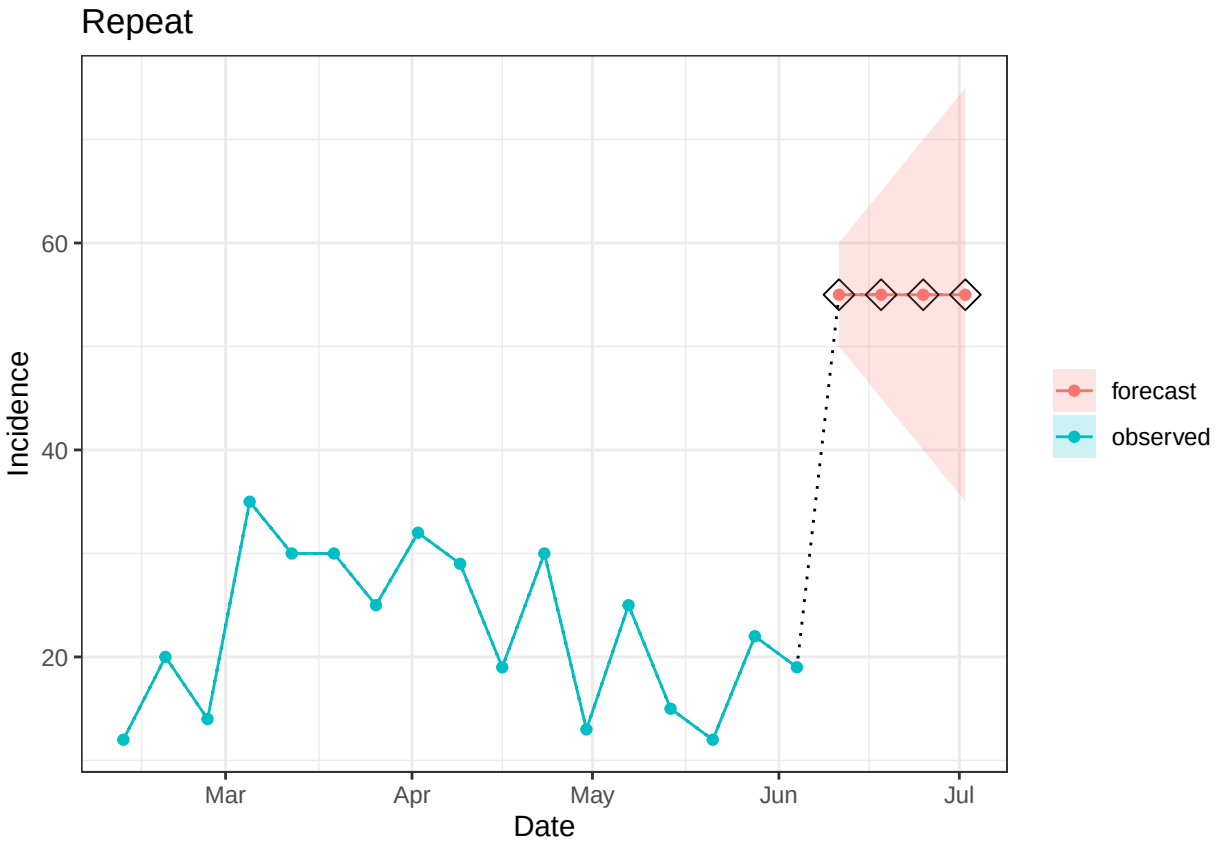

Figure A5: Depiction of a flag raised with the repeat component. The repeat component checks whether consecutive values in an observed or forecasted signal are repeated k times. When the seed is created, it stores the maximum number of consecutive repeats for each location and uses this as the default value for k. If the evaluated data exceeds k then the signal is considered implausible and a flag is raised.

9.6 A6: Trend component

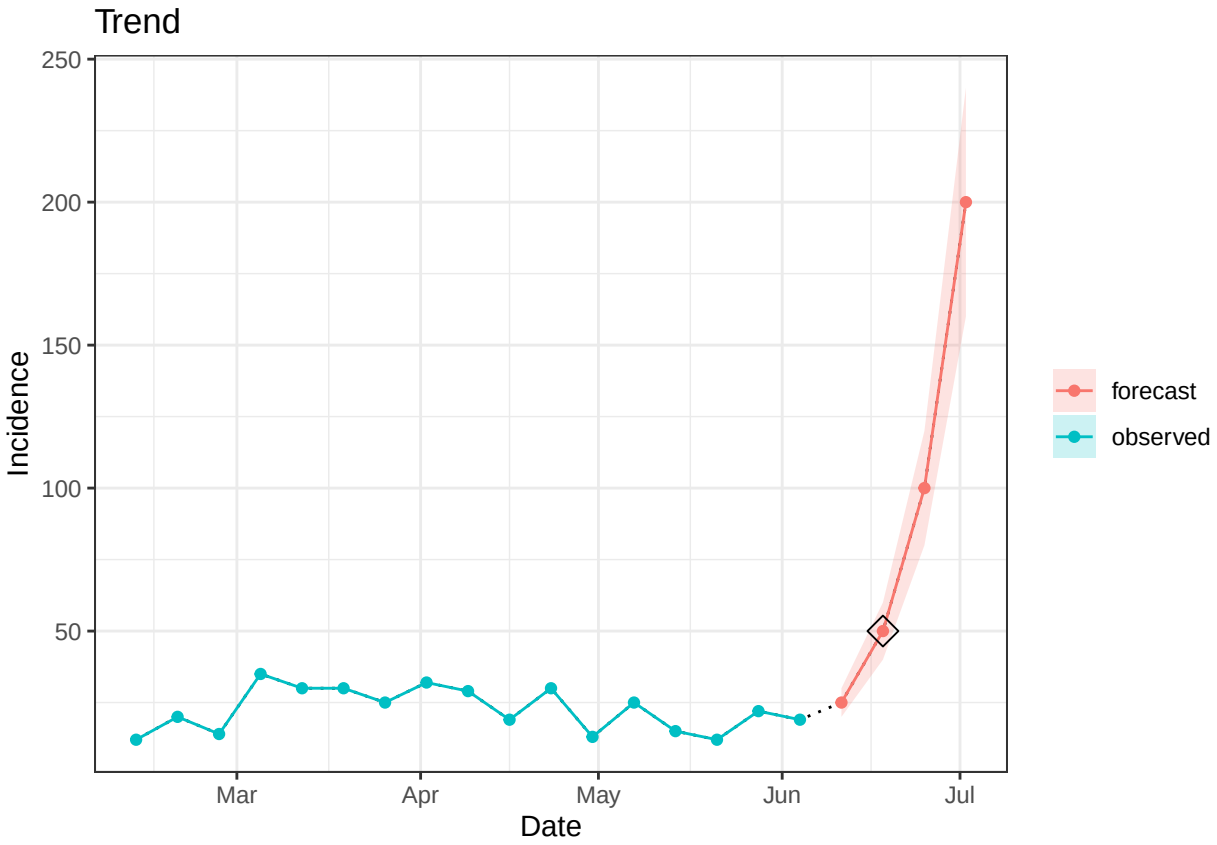

Figure A6: Depiction of a flag raised with the trend component. The trend component assesses whether there is a significant change in the magnitude or direction of the slope for the evaluated signal compared to the most recent data in the seed. If a "change point" is identified in any of the forecasted horizons and/or the most recent seed value, then the flag is raised for implausibility.

9.7 A7: Shape component

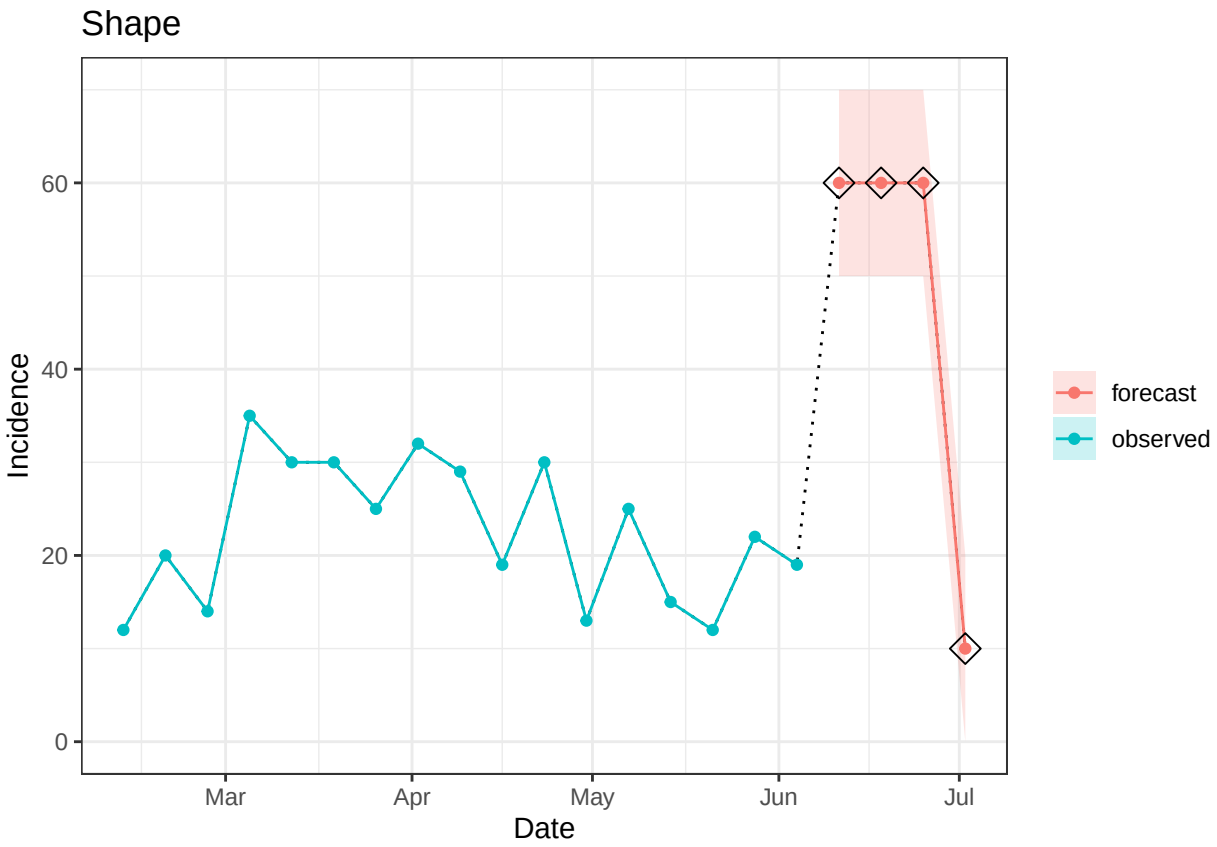

Figure A7: Depiction of a flag raised with the shape component. The shape component evaluates the shape of the trajectory of the forecast signal and compares that shape to existing shapes in the observed seed data. If the shape is identified as novel, a flag is raised, and the signal is considered implausible.

9.8 A8: Zero component

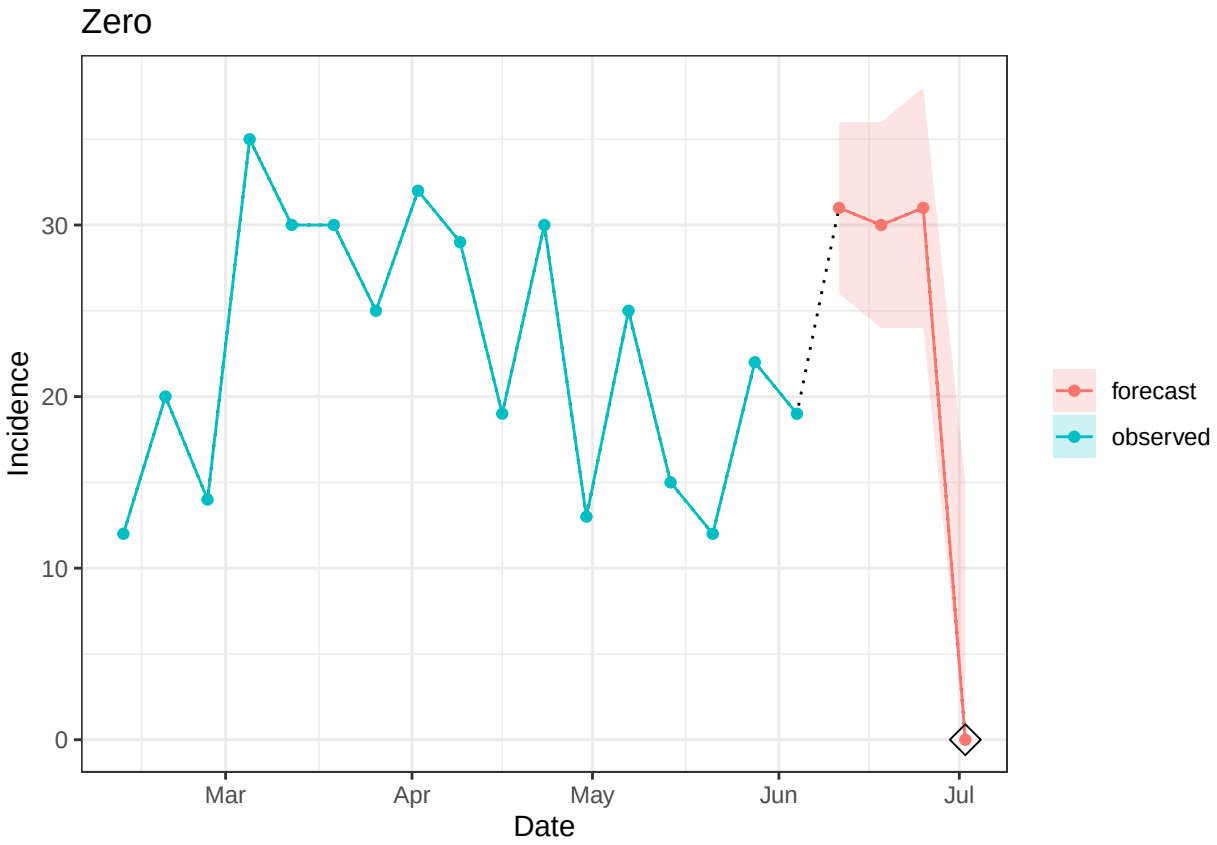

Figure A8: Depiction of a flag raised with the zero component. This component checks for the presence of any value equal to zero in the evaluated signal. If there are any zeros found, then the component will look in the seed to see if there are zeros anywhere else in the time series. If so, the component will consider the evaluated zero plausible and no flags will be raised. If not, the component will consider the evaluated zero implausible and a flag will be raised.