

Supplemental Code Demo

Accessing Data from the US Flusight Hubverse Project

Consortium of Infectious Disease Modeling Hubs			Melissa Kerr
Rebecca Borchering	Alvaro Castro Rivadeneira		Lucie Contamin
Sebastian Funk	Harry Hochheiser	Emily Howerton	Anna Krystalli
Li Shandross	Nicholas G Reich		

2025-06-11

Software configuration

The following demonstration shows how to access and analyze data from a real hub using and several of the hubverse R packages.[The Consortium of Infectious Disease Modeling Hubs, 2025] The following packages are on CRAN and may be installed using the standard `install.packages()` function: `hubEnsembles`, `aws.s3`, `arrow`, `ggplot2` and `dplyr`.

The hubverse packages `hubData`, `hubVis`, and `hubEvals` must be installed through the R-universe package repository using the commands below.

```
install.packages(pkgs = c("hubData", "hubVis", "hubEvals"),
                 repos = c("https://hubverse-org.r-universe.dev",
                           "https://cloud.r-project.org"))
```

Once the required packages are installed, load them into your R session. (Session information including package versions are included at the end of this vignette.)

```
library(hubData)
library(hubEnsembles)
library(hubVis)
library(hubEvals)
```

Load forecast data

CDC's FluSight Forecast Hub exists as a GitHub repository.[US CDC, 2025] Using hubverse tools, it also has been mirrored to the cloud, so a copy of the data lives at a publicly-accessible AWS S3 bucket. This means that instead of needing to download a local copy of the repository, the data for this project can be directly accessed from the cloud using `hubData` tools.

We start by initializing a connection to the hub in the cloud. Below, we print out metadata about the resulting connection. This is by default not printed, but examining it shows how the hub configuration is passed to the various utilities.

```
(hub_con <- hubData::s3_bucket("cdcepi-flusight-forecast-hub/") |>
  hubData::connect_hub(skip_checks = TRUE))
#>
#> -- <hub_connection/FileSystemDataset> --
#>
#> * hub_name: "US CDC FluSight"
```

```

#> * hub_path: 'cdcepi-flusight-forecast-hub/'
#> * file_format: "parquet(2454/2454)"
#> * checks: FALSE
#> * file_system: "S3FileSystem"
#> * model_output_dir: "model-output/"
#> * config_admin: 'hub-config/admin.json'
#> * config_tasks: 'hub-config/tasks.json'
#>
#> -- Connection schema
#> hub_connection with 2454 Parquet files
#> 9 columns
#> reference_date: date32[day]
#> target: string
#> horizon: int32
#> location: string
#> target_end_date: date32[day]
#> output_type: string
#> output_type_id: string
#> value: double
#> model_id: string

```

We then collect a subset of model output that we will interact with further in this demonstration. Specifically, we will extract zero through three week-ahead quantile predictions of weekly incident flu hospitalizations made for the state of Texas on three dates in late 2023 and early 2024. For simplicity, we will restrict our query to predictions made for five arbitrarily chosen models, although predictions for over 30 models are available on each of these dates.

```

models <- c("FluSight-baseline",
            "CMU-TimeSeries",
            "UMass-flusion",
            "cfa-flumech",
            "UGA_flucast-INFLAenza")
reference_dates <- c("2023-12-16", "2024-01-13", "2024-02-10")

flusight_data <- hub_con |>
  dplyr::filter(target == "wk inc flu hosp",
                reference_date %in% reference_dates,
                model_id %in% models,
                output_type == "quantile",
                location == "48", ## FIPS code for Texas
                horizon > -1) |>
  hubData::collect_hub()

```

Table 1: Five rows of the `flusight_data` object collected above. The “target”, “horizon”, and “target_end_date” columns have been excluded for readability. The “output_type_id” and “value” columns show the different predicted quantiles for the number of hospital admissions due to influenza made by the CMU-TimeSeries model for location 48 (Texas). The predictions were made during the week of December 16, 2023 (reference date) and are making a prediction for the horizon of 1 week ahead, for the week ending December 23.

model_id	reference_date	location	output_type	output_type_id	value
CMU-TimeSeries	2024-01-13	48	quantile	0.01	305.56
CMU-TimeSeries	2024-01-13	48	quantile	0.025	516.40
CMU-TimeSeries	2024-01-13	48	quantile	0.05	752.16
CMU-TimeSeries	2024-01-13	48	quantile	0.1	1096.37
CMU-TimeSeries	2024-01-13	48	quantile	0.15	1346.90

Loading target data

We can use the `aws.s3` package to read the contents of the most up-to-date target data file stored in the cloud.

```
target_data <-
  aws.s3::s3read_using(
    readr::read_csv,
    object = "s3://cdcepi-flusight-forecast-hub/target-data/target-hospital-admissions.csv"
  ) |>
  dplyr::mutate(target = "wk inc flu hosp") |>
  dplyr::select(date, target, location, value) |> # keep only required columns
  dplyr::filter(location == "48",
    date >= as.Date("2023-09-23"), date <= as.Date("2024-05-01")) |>
  dplyr::rename(observation = value) |>
  dplyr::arrange(date)
```

Table 2: Five rows of the `target_data` object read in above. The observation column shows the number of hospital admissions due to influenza reported in Texas (location 48) on the week ending on each date.

date	target	location	observation
2023-09-23	wk inc flu hosp	48	102
2023-09-30	wk inc flu hosp	48	118
2023-10-07	wk inc flu hosp	48	119
2023-10-14	wk inc flu hosp	48	167
2023-10-21	wk inc flu hosp	48	242

Visualize forecasts

We can use the `hubVis` package to create a single graphic that overlays the forecasts from the five models and three forecast dates with the observed data.

```
flusight_data |>
  hubVis::plot_step_ahead_model_output(
    target_data,
    x_col_name = "target_end_date",
    use_median_as_point = TRUE,
    group = "reference_date",
    interactive = FALSE,
    fill_transparency = .1,
    intervals = 0.8
  ) +
  ggplot2::theme(legend.position = "bottom") +
  ggplot2::ylab("hospital admissions due to influenza")
```

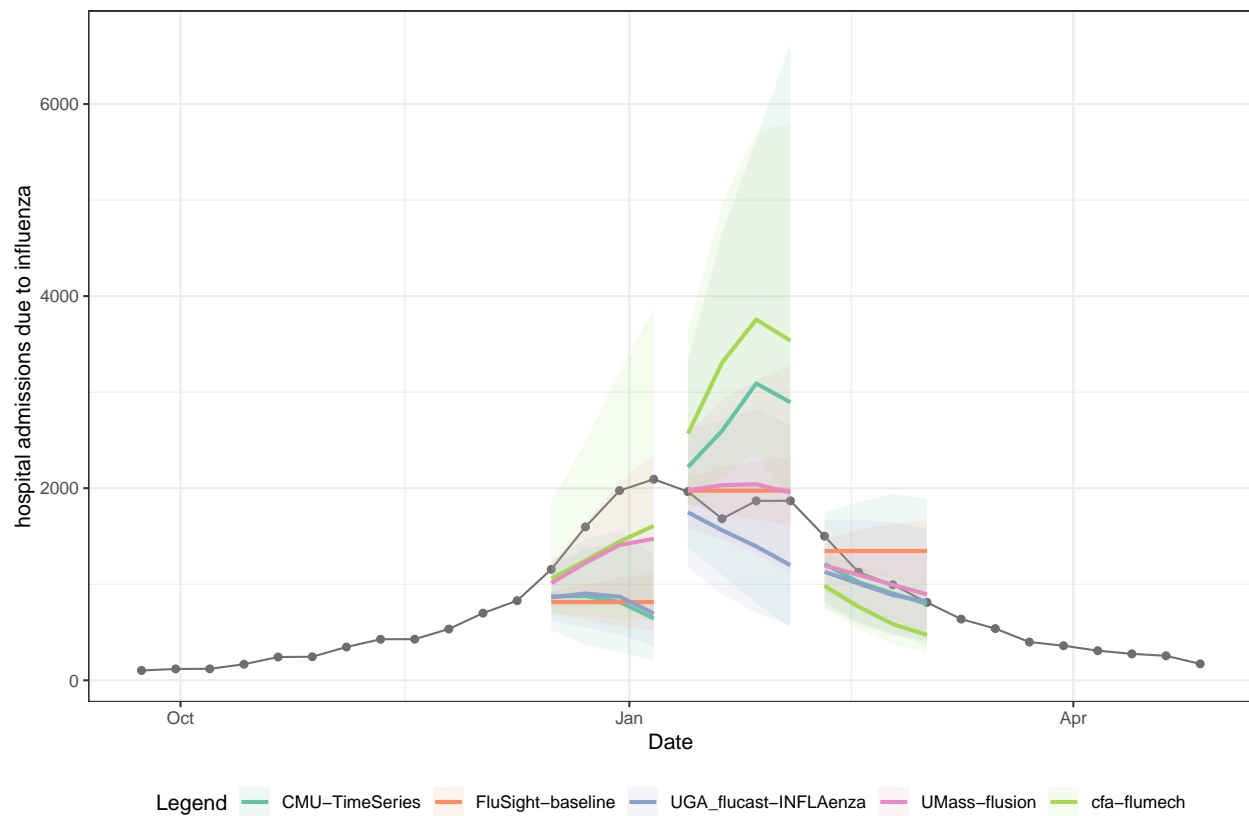


Figure 1: Forecasts of hospital admissions in Texas due to influenza in the 2023-2024 respiratory virus season. Predictions from five models are shown along with observed data (line with dots for specific observations). Each model is shown in a different color, with the 80% prediction intervals shown as a shaded region.

Build ensemble and visualize

Using the model output data collected above, we can pass these data to the `hubEnsembles::simple_ensemble()` function to build an ensemble that takes the mean of predicted values from each model at each quantile level. This new forecast output is then added to the existing forecasts and visualized all together. The `hubEnsembles` package supports a range of aggregation approaches including a linear opinion pool (probability density averaging) or using the mean or median of quantiles.[Shandross et al., 2025] Model weights can be specified for the various approaches, allowing for some models to influence the output more strongly.

```
ensemble_forecast <- hubEnsembles::simple_ensemble(flusight_data)

all_forecasts <- flusight_data |>
  dplyr::bind_rows(ensemble_forecast)

all_forecasts |>
  hubVis::plot_step_ahead_model_output(
    target_data,
    x_col_name = "target_end_date",
    use_median_as_point = TRUE,
    group = "reference_date",
    interactive = FALSE,
    fill_transparency = .2,
    intervals = 0.90,
    one_color = "darkgray",
    ens_name = "hub-ensemble",
    ens_color = "orange",
    pal_color = NULL) +
  ggplot2::theme(legend.position = "bottom") +
  ggplot2::ylab("hospital admissions due to influenza")
```

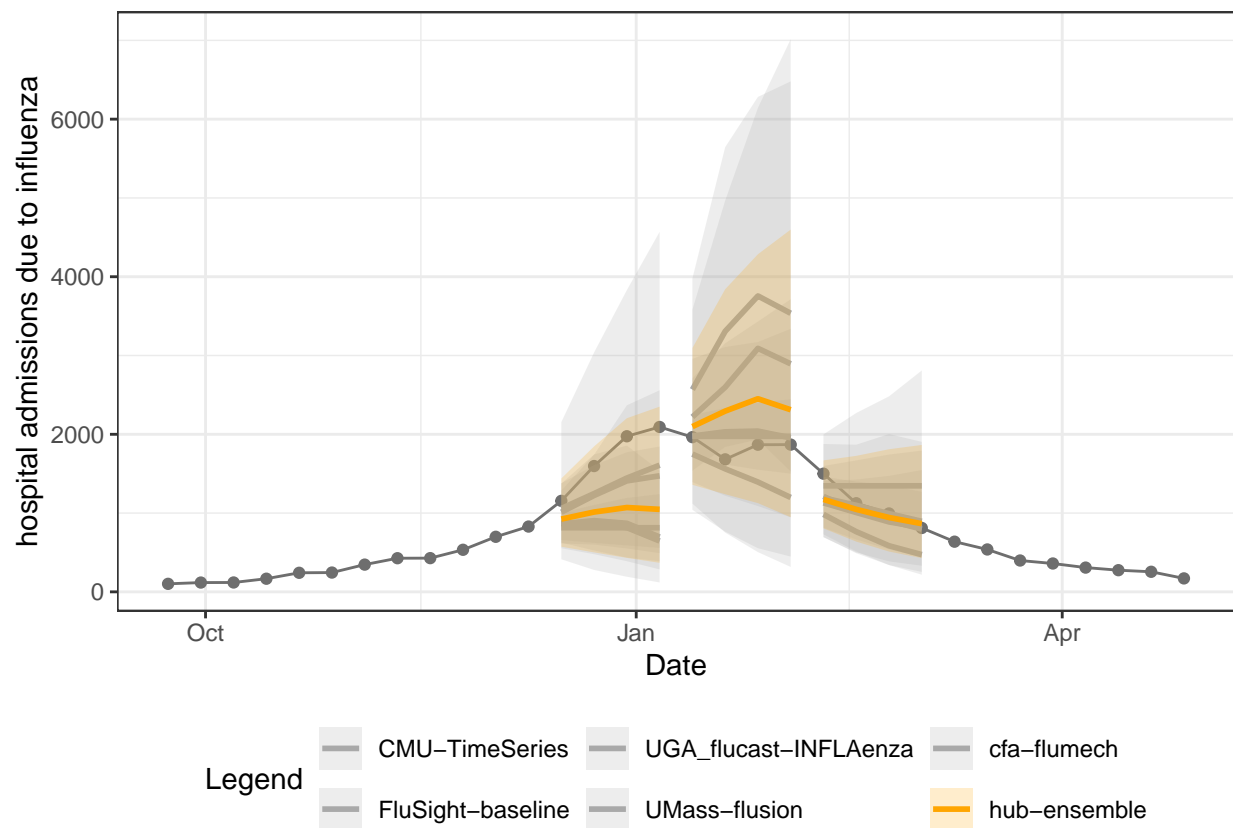


Figure 2: Forecasts of hospital admissions in Texas due to influenza in the 2023-2024 respiratory virus season. Predictions from five submitted models and an ensemble are shown along with observed data. The observed data are shown as a line with dots for specific weekly observations. Each individual model is shown in orange and the mean ensemble is shown in black, with the 80% prediction intervals shown as shaded regions.

Evaluate forecasts

The package `hubEvals` can be used to evaluate probabilistic predictions against observed data. The `hubEvals` package creates an interface for `hubverse`-style model output and target data to the `scoringutils` package, which is an engine for computing a number of common scores for probabilistic predictions.[Bosse et al., 2024]

The code below starts by making oracle output data, which is a dataset formatted like `hubverse`-style model output data but with a point prediction representing the eventual observation, as if observed by an oracle model. The `FluSight` hub stores target data in a time-series format but not in an oracle output format. The oracle output data format is convenient to use for scoring, as it can be joined with model output data and then the observation and the prediction reside in a single row of stored data. In this example, we compute and show results for the weighted interval score (WIS) with its associated decomposition into overprediction, underprediction, and dispersion penalties, the 50% and 90% prediction interval coverage rates, and the absolute error computed on the median prediction.[Bracher et al., 2021]

```
## make oracle output
oracle_output <- target_data |>
  dplyr::select(date, location, observation) |>
  dplyr::rename(
    target_end_date = date,
    oracle_value = observation) |>
  tidyr::crossing(horizon = 0:3) |>
  dplyr::mutate(reference_date = as.Date(target_end_date) - horizon * 7L) |>
  dplyr::filter(reference_date %in% reference_dates)

scores <- hubEvals::score_model_out(
  all_forecasts,
  oracle_output,
  baseline = "FluSight-baseline") |>
  dplyr::arrange(wis)

scoringutils::plot_wis(scores, x="model_id")
```

Table 3: A print-out of the `scores` object computed above, containing the summary metrics for the five component models and the ensemble model. Columns including other metrics, including bias, underprediction, overprediction and dispersion were omitted for space.

model_id	wis	interval_coverage_50	interval_coverage_90	ae_median
UMass-flusion	164.00	0.50	0.92	228.71
hub-ensemble	255.90	0.50	1.00	420.65
UGA_flucast-INFLAenza	301.79	0.50	0.83	464.08
FluSight-baseline	344.22	0.25	0.58	444.83
CMU-TimeSeries	398.85	0.50	0.83	627.16
cfa-flumech	444.97	0.33	0.75	739.35

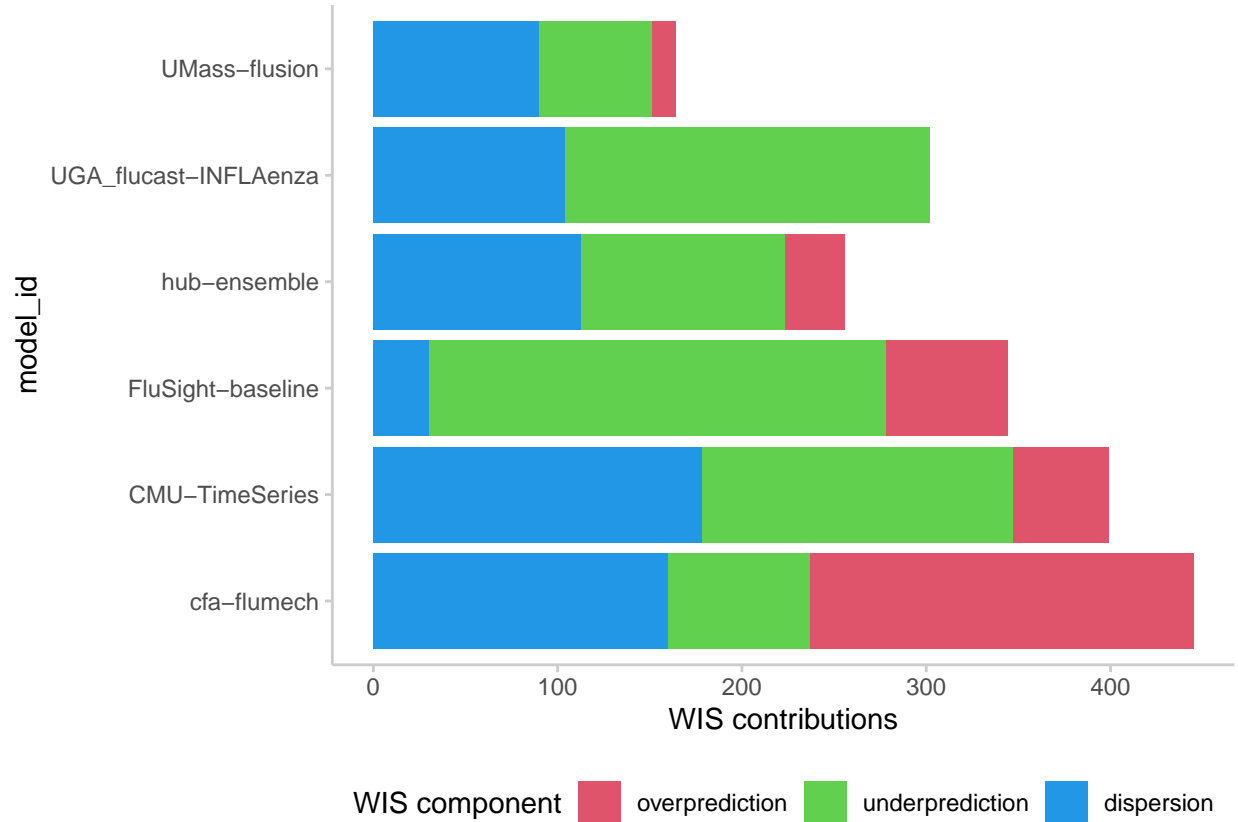


Figure 3: Average weighted interval score (WIS) computed by model (y-axis). The total length of each bar is the average WIS across all predictions made. WIS can be decomposed into penalties for overprediction, underprediction and dispersion, the contributions of which are shown by the segments of the bar. The UMass-flusion model has the lowest (best) average WIS, and it received a greater penalty for underprediction relative to the penalty for overprediction.


```
sessionInfo()
#> R version 4.4.3 (2025-02-28)
#> Platform: aarch64-apple-darwin20
#> Running under: macOS Sequoia 15.5
#>
#> Matrix products: default
#> BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/Libraries/libBLAS.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> time zone: America/New_York
#> tzcode source: internal
#>
#> attached base packages:
#> [1] stats      graphics  grDevices  utils      datasets  methods    base
#>
#> other attached packages:
#> [1] hubEvals_0.0.0.9001      hubVis_0.1.2.9000      hubEnsembles_1.0.0.9000 hubData_1.3.0.9000
#>
#> loaded via a namespace (and not attached):
#> [1] gtable_0.3.6             xfun_0.52              ggplot2_3.5.2          htmlwidgets_0.14.1
#> [8] tools_4.4.3             generics_0.1.4         curl_6.3.0             parallel_4.0.3
#> [15] checkmate_2.3.2         RColorBrewer_1.1-3     assertthat_0.2.1       lifecycle_1.0.3
#> [22] tinytex_0.57            htmltools_0.5.8.1     yaml_2.3.10            lazyeval_0.0.2
#> [29] tidyr_1.3.1            MASS_7.3-64           aws.s3_0.3.21          cachem_1.1.0
#> [36] stringi_1.8.7          zoltr_1.0.2           dplyr_1.1.4           purrr_1.0.2
#> [43] grid_4.4.3             cli_3.6.5             magrittr_2.0.3         hubUtils_0.1.0
#> [50] readr_2.1.5            withr_3.0.2           scales_1.4.0           backports_1.4.0
#> [57] matrixStats_1.5.0      bit_4.6.0             scoringutils_2.1.0.9000 hms_1.1.3
#> [64] viridisLite_0.4.2      scoringRules_1.1.3     rlang_1.1.6           Rcpp_1.0.11
#> [71] vroom_1.6.5            jsonlite_2.0.0        R6_2.6.1              fs_1.6.6
```

Nikos I. Bosse, Hugo Gruson, Anne Cori, Edwin van Leeuwen, Sebastian Funk, and Sam Abbott. Evaluating forecasts with scoringutils in R. *arXiv*, 2024. doi: 10.48550/ARXIV.2205.07090. URL <https://arxiv.org/abs/2205.07090>.

Johannes Bracher, Evan L. Ray, Tilmann Gneiting, and Nicholas G. Reich. Evaluating epidemic forecasts in an interval format. *PLoS Computational Biology*, 17(2):e1008618, February 2021. doi: 10.1371/journal.pcbi.1008618. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7880475/>.

Li Shandross, Emily Howerton, Lucie Contamin, Harry Hochheiser, Anna Krystalli, Consortium of Infectious Disease Modeling Hubs, Nicholas G. Reich, and Evan L. Ray. Multi-model ensembles in infectious disease and public health: Methods, interpretation, and implementation in R. May 2025. doi: 10.1101/2024.06.24.24309416. URL <https://www.medrxiv.org/content/10.1101/2024.06.24.24309416v2>.

The Consortium of Infectious Disease Modeling Hubs. *The hubverse: open tools for collaborative modeling.*, 2025. URL <https://hubverse.io>. Accessed: 2025-06-10.

US CDC. *cdcepi/FluSight-forecast-hub*, June 2025. URL <https://github.com/cdcepi/FluSight-forecast-hub>.
Accessed: 2025-06-10.