# Package 'hubEnsembles'

October 2, 2024

**Title** Ensemble Methods for Combining Hub Model Outputs

**Version** 0.1.9

**Description** Functions for combining model outputs (e.g. predictions or
estimates) from multiple models into an aggregated ensemble model
output.

**License** MIT + file LICENSE

**URL** <https://github.com/hubverse-org/hubEnsembles>,
<https://hubverse-org.github.io/hubEnsembles/>

**BugReports** <https://github.com/hubverse-org/hubEnsembles/issues>

**Depends** R (>= 4.1)

**Imports** cli, distfromq (>= 1.0.2), dplyr, hubUtils (>= 0.0.1),
matrixStats, purrr, rlang, stats, tidyr, tidyselect

**Suggests** cowplot, ggplot2, hubExamples, hubVis (>= 0.0.0.9100), knitr,
rmarkdown, testthat (>= 3.0.0)

**Additional_repositories** <https://hubverse-org.r-universe.dev/>

**Config/Needs/website** hubverse-org/hubStyle

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Evan L Ray [aut],
Li Shandross [aut, cre] (<<https://orcid.org/0009-0008-1348-1954>>),
Emily Howerton [aut] (<<https://orcid.org/0000-0002-0639-3728>>),
Anna Krystalli [ctb] (<<https://orcid.org/0000-0002-2378-4915>>),
Zhian N. Kamvar [ctb] (<<https://orcid.org/0000-0003-1458-7108>>),
Nicholas G. Reich [ctb] (<<https://orcid.org/0000-0003-3503-9899>>),
Consortium of Infectious Disease Modeling Hubs [cph]

**Maintainer** Li Shandross <lshandross@umass.edu>

**Repository** CRAN

**Date/Publication** 2024-10-02 20:20:09 UTC

# Contents

---

component_outputs          *Example model output data for* linear_pool()

---

## Description

Toy model output data formatted according to hubverse standards to be used in the examples for linear_pool(). The predictions included are taken from three normal distributions with means -3, 0, 3 and all standard deviations 1.

## Usage

```
component_outputs
```

## Format

component_outputs:

A data frame with 123 rows and 5 columns:

**model_id** model ID

**target** forecast target

**output_type** type of forecast

**output_type_id** output type ID

**value** forecast value

---

fweights          *Example weights data for* simple_ensemble()

---

## Description

Toy weights data formatted according to hubverse standards to be used in the examples for simple_ensemble()

## Usage

```
fweights
```

## Format

fweights**:**

A data frame with 8 rows and 3 columns:

**model_id**  model ID
**location**  FIPS codes
**weight**  weight

---

| linear_pool | *Compute ensemble model outputs as a linear pool, otherwise known as a distributional mixture, of component model outputs for each combination of model task, output type, and output type id. Supported output types include* mean*,* quantile*,* cdf*, and* pmf*.* |
|---|---|

---

## Description

Compute ensemble model outputs as a linear pool, otherwise known as a distributional mixture, of component model outputs for each combination of model task, output type, and output type id. Supported output types include mean, quantile, cdf, and pmf.

## Usage

```
linear_pool(
  model_out_tbl,
  weights = NULL,
  weights_col_name = "weight",
  model_id = "hub-ensemble",
  task_id_cols = NULL,
  n_samples = 10000,
  ...
)
```

## Arguments

| | |
|---|---|
| model_out_tbl | an object of class model_out_tbl with component model outputs (e.g., predictions). |
| weights | an optional data.frame with component model weights. If provided, it should have a column named model_id and a column containing model weights. Optionally, it may contain additional columns corresponding to task id variables, output_type, or output_type_id, if weights are specific to values of those variables. The default is NULL, in which case an equally-weighted ensemble is calculated. Should be prevalidated. |
| weights_col_name | character string naming the column in weights with model weights. Defaults to "weight" |
| model_id | character string with the identifier to use for the ensemble model. |

| task_id_cols | character vector with names of columns in model_out_tbl that specify modeling tasks. Defaults to NULL, in which case all columns in model_out_tbl other than "model_id", "output_type", "output_type_id", and "value" are used as task ids. |
| --- | --- |
| n_samples | numeric that specifies the number of samples to use when calculating quantiles from an estimated quantile function. Defaults to 1e4. |
| ... | parameters that are passed to distfromq::make_q_fn, specifying details of how to estimate a quantile function from provided quantile levels and quantile values for output_type "quantile". |

### Details

The underlying mechanism for the computations varies for different output_types. When the output_type is cdf, pmf, or mean, this function simply calls simple_ensemble to calculate a (weighted) mean of the component model outputs. This is the definitional calculation for the CDF or PMF of a linear pool. For the mean output type, this is justified by the fact that the (weighted) mean of the linear pool is the (weighted) mean of the means of the component distributions.

When the output_type is quantile, we obtain the quantiles of a linear pool in three steps:

1. Interpolate and extrapolate from the provided quantiles for each component model to obtain an estimate of the CDF of that distribution.

2. Draw samples from the distribution for each component model. To reduce Monte Carlo variability, we use quasi-random samples corresponding to quantiles of the estimated distribution.

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

Steps 1 and 2 in this process are performed by distfromq::make_q_fn.

### Value

a model_out_tbl object of ensemble predictions. Note that any additional columns in the input model_out_tbl are dropped.

### Examples

```
# We illustrate the calculation of a linear pool when we have quantiles from the
# component models. We take the components to be normal distributions with
# means -3, 0, and 3, all standard deviations 1, and weights 0.25, 0.5, and 0.25.
data(component_outputs)
data(weights)

expected_quantiles <- seq(from = -5, to = 5, by = 0.25)
lp_from_component_qs <- linear_pool(component_outputs, weights)

head(lp_from_component_qs)
all.equal(lp_from_component_qs$value, expected_quantiles, tolerance = 1e-2,
          check.attributes = FALSE)
```

---

model_outputs      *Example model output data for* simple_ensemble()

---

## Description

Toy model output data formatted according to hubverse standards to be used in the examples for simple_ensemble()

## Usage

```
model_outputs
```

## Format

model_outputs:

A data frame with 24 rows and 8 columns:

**model_id** model ID

**location** FIPS codes

**horizon** forecast horizon

**target** forecast target

**target_date** date that the forecast is for

**output_type** type of forecast

**output_type_id** output type ID

**value** forecast value

---

simple_ensemble    *Compute ensemble model outputs by summarizing component model outputs for each combination of model task, output type, and output type id. Supported output types include* mean, median, quantile, cdf, *and* pmf.

---

## Description

Compute ensemble model outputs by summarizing component model outputs for each combination of model task, output type, and output type id. Supported output types include mean, median, quantile, cdf, and pmf.

**Usage**

```
simple_ensemble(
  model_out_tbl,
  weights = NULL,
  weights_col_name = "weight",
  agg_fun = mean,
  agg_args = list(),
  model_id = "hub-ensemble",
  task_id_cols = NULL
)
```

**Arguments**

model_out_tbl    an object of class `model_out_tbl` with component model outputs (e.g., predictions).

weights          an optional `data.frame` with component model weights. If provided, it should have a column named `model_id` and a column containing model weights. Optionally, it may contain additional columns corresponding to task id variables, `output_type`, or `output_type_id`, if weights are specific to values of those variables. The default is `NULL`, in which case an equally-weighted ensemble is calculated. Should be prevalidated.

weights_col_name
                 `character` string naming the column in `weights` with model weights. Defaults to `"weight"`

agg_fun          a function or character string name of a function to use for aggregating component model outputs into the ensemble outputs. See the details for more information.

agg_args         a named list of any additional arguments that will be passed to `agg_fun`.

model_id         `character` string with the identifier to use for the ensemble model.

task_id_cols     `character` vector with names of columns in `model_out_tbl` that specify modeling tasks. Defaults to `NULL`, in which case all columns in `model_out_tbl` other than `"model_id"`, `"output_type"`, `"output_type_id"`, and `"value"` are used as task ids.

**Details**

The default for `agg_fun` is `"mean"`, in which case the ensemble's output is the average of the component model outputs within each group defined by a combination of values in the task id columns, output type, and output type id. The provided `agg_fun` should have an argument `x` for the vector of numeric values to summarize, and for weighted methods, an argument `w` with a numeric vector of weights. If it desired to use an aggregation function that does not accept these arguments, a wrapper would need to be written. For weighted methods, `agg_fun = "mean"` and `agg_fun = "median"` are translated to use `matrixStats::weightedMean` and `matrixStats::weightedMedian` respectively. For `matrixStats::weightedMedian`, the argument `interpolate` is automatically set to FALSE to circumvent a calculation issue that results in invalid distributions.

## Value

a `model_out_tbl` object of ensemble predictions. Note that any additional columns in the input `model_out_tbl` are dropped.

## Examples

```
# Calculate a weighted median in two ways
data(model_outputs)
data(fweights)

weighted_median1 <- simple_ensemble(model_outputs, weights = fweights,
                                    agg_fun = stats::median)
weighted_median2 <- simple_ensemble(model_outputs, weights = fweights,
                                    agg_fun = matrixStats::weightedMedian)
all.equal(weighted_median1, weighted_median2)
```

---

| weights | *Example weights data for* `linear_pool()` |
| --- | --- |

---

## Description

Toy weights data formatted according to hubverse standards to be used in the examples for `linear_pool()`. Weights are 0.25, 0.5, 0.25.

## Usage

```
weights
```

## Format

`weights`:
A data frame with 3 rows and 2 columns:

**model_id** model ID
**location** FIPS codes
**weight** weight

# Index