1 Method Comparison

This chapter aims to compare the effectiveness of all Post-Processing methods that were introduced in the previous chapters. In particular, we investigate if some methods consistently outperform other procedures across a wide range of scenarios, i.e. different data sets and different covariate combinations.

Further, it will be interesting to observe the *types* of adjustments to the original forecasts: Some methods might improve the Weighted Interval Score by *extending* the interval width and thus increasing coverage whereas others might yield a similar final score by *shrinking* the prediction intervals leading to a higher precision. One can imagine even more variations: Moving the interval bounds farther apart or closer together can happen in *symmetric* or *asymmetric* manner and the interval's midpoint might stay *fixed* or get *shifted* throughout the post-processing process.

Before jumping into the analysis, we propose one additional model that, in contrast to those we have covered so far, does not add any new information to the equation. Instead, it *combines* the predictions from existing post-processing methods to build an *ensemble* prediction. The idea is that leveraging information from multiple independent algorithms can stabilize estimation since the ensemble learns to focus on a model with a strong performance for one particular covariate set while the same model might perform worse for a different covariate set and, thus, make little contributions to the ensemble in that case.

Next, we explain the mathematical reasoning behind the ensemble model in more detail.

1.1 Ensemble Model

There exist various options how to combine multiple building blocks into one ensemble. We chose an approach that can be efficiently computed by well-understood algorithms on the one hand and is highly interpretable on the other hand. Each quantile prediction of our ensemble model is a *convex combination* of the individual methods, i.e. a linear combination where all weights are contained in the unit interval and sum up to one. Hence, the resulting value lives on the same scale as the original predictions and each weight can be interpreted as the *fractional contribution* of each building block method

Consider one particular feature combination of `model`, `location`, `horizon`, `target_type` and `quantile`. Let n specify the number of observations in the training set within this combination, $\mathbf{y} \in \mathbb{R}^n$ the vector of true values, $\mathbf{l}_1, \dots, \mathbf{l}_k \in \mathbb{R}^n$ vectors of original lower quantile predictions and $\mathbf{u}_1, \dots, \mathbf{u}_k \in \mathbb{R}^n$ vectors of original upper quantile predictions from k different post-processing procedures.

Then, for each such combination, the ensemble model computes weights $\mathbf{w}^* \in [0,1]^k$ by solving the following nonlinear constrained optimization problem:

$$\begin{aligned}
\mathbf{w}^* = \underset{\mathbf{w}\in[0,1]^k}{\arg\min}\, IS_\alpha(\mathbf{y}) = \underset{\mathbf{w}\in[0,1]^k}{\arg\min} (\mathbf{u}-\mathbf{l}) + \frac{2}{\alpha}\cdot(\mathbf{l}-\mathbf{y})\cdot\mathbb{1}(\mathbf{y}\le\mathbf{l}) + \frac{2}{\alpha}\cdot(\mathbf{y}-\mathbf{u})\cdot\mathbb{1}(\mathbf{y}\ge\mathbf{u}),\\
\text{with} \qquad \mathbf{l} = \sum_{j=1}^k w_j\mathbf{l}_j, \quad \mathbf{u} = \sum_{j=1}^k w_j\mathbf{u}_j\\
\text{s.t.} \qquad \|\mathbf{w}\|_1 = \sum_{j=1}^k w_j = 1,
\end{aligned}$$

where all operations for vector inputs $\mathbf{l}$, $\mathbf{u}$ and $\mathbf{y}$ are understood elementwise and the *same* weights w_j, $j = 1, \dots, k$ are chosen for lower and upper quantiles.

Hence, we choose the (nonlinear) Interval Score (**??**) as our objective function that we minimize subject to linear constraints. The optimization step is implemented with the nloptr[1] package (Ypma and Johnson 2022), which describes itself as "an R interface to NLopt, a free/open-source library for nonlinear optimization".

[1] https://cran.r-project.org/web/packages/nloptr/index.html

Table 1: WIS of all Post-Processing Methods on Training and Validation Set

method	validation score	training score	dispersion
ensemble	57.69	18.22	21.73
qsa_uniform	60.00	20.88	26.84
qsa_flexible	60.47	19.48	25.31
qsa_flexible_symmetric	60.92	20.49	33.22
cqr	62.15	20.82	24.10
cqr_asymmetric	63.97	14.46	17.99
original	65.74	23.62	12.00

Note that, technically, the weight vector has to be denoted by $\mathbf{w}^*_{m,l,h,t,q}$ since the computed weights are generally different for each feature combination. We omit the subscripts at this point to keep the notation clean.

The Interval Score always considers *pairs* of quantiles α and $1-\alpha$ as outer bounds of a $(1-2\alpha)\cdot 100\%$ prediction interval. The best results are achieved when a separate weight vector for each quantile pair is computed. Since our data sets contain 11 quantile pairs, 2 target types and 4 horizons and we consider 6 different forecasters, the ensemble model requires solving $11 \cdot 2 \cdot 4 \cdot 6 = 528$ nonlinear optimization problems for each location, which amounts to $18 \cdot 528 = 9504$ optimization problems for the European Hub Data Set.

Due to this high computational cost the *maximum number of iterations* within each optimization is an important hyperparameter that balances the trade-off between computational feasibilty and sufficient convergence of the iterative optimization algorithm. Here, we ultimately settled with 10.000 maximum steps which could ensure convergence with respect to a *tolerance level* of 10^{-8} in the vast majority of cases.

Finally, it is worth noting that the weight vector of the ensemble model $\mathbf{w}^*$ is learned on a *training set* such that a fair comparison with all individual post-processing methods on a separate *validation set* is possible.

1.2 Comparison of CQR, QSA & Ensemble

Now that we have introduced *Conformalized Quantile Regression* in **??**, *Quantile Spread Averaging* in **??** and the *Ensemble* Model in Section 1.1, the obvious question is which of the methods performs best. Thus, this section is dedicated to a detailed comparison across various covariate combinations as well as both the UK Forecasting Challenge data and the European Forecast Hub data.

Except for some minor modifications for computational efficiency, the results that constitute the starting point of the analysis in this chapter can be generated with the following commands, where `df` represents either the UK or the European Forecast Hub data set:

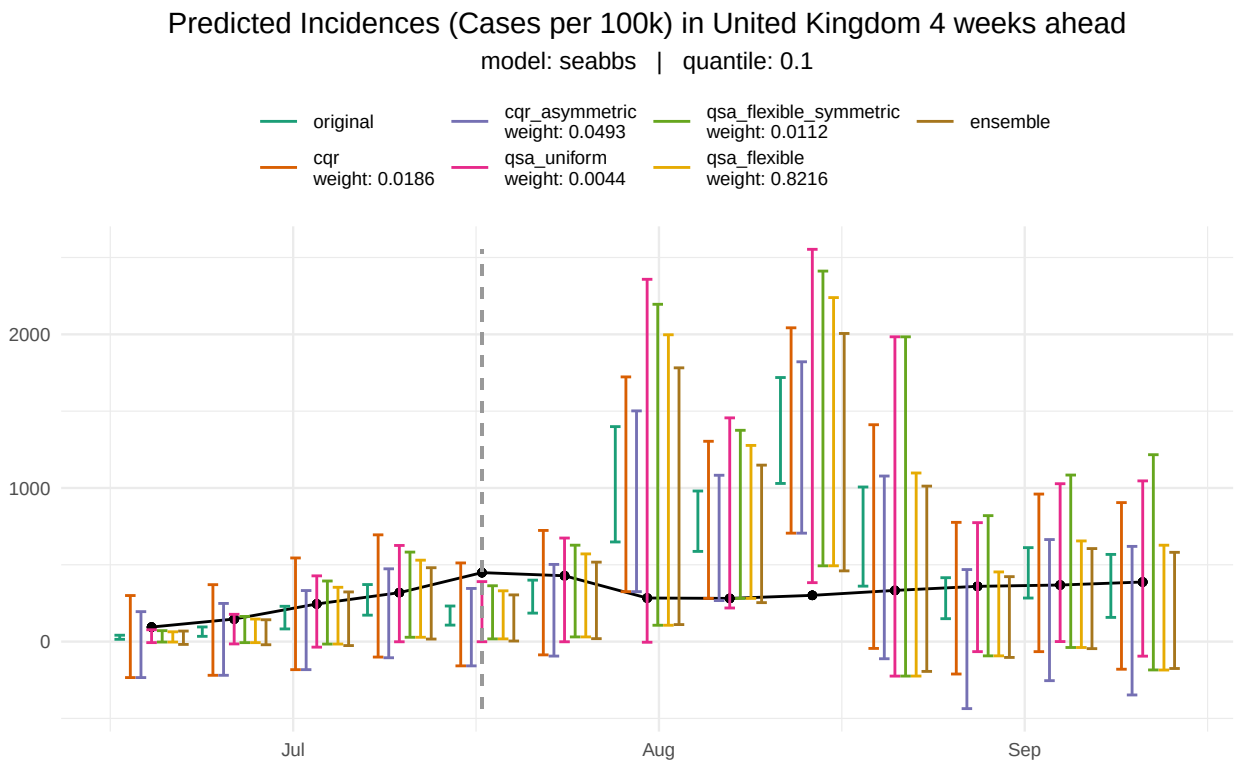
```
library(postforecasts)

df_updated <- df |>
  update_predictions(
    methods = c(
      "cqr", "cqr_asymmetric", "qsa_uniform", "qsa_flexible", "qsa_flexible_symmetric"
    ),
    cv_init_training = 0.5
  ) |>
  collect_predictions() |>
  add_ensemble()
```

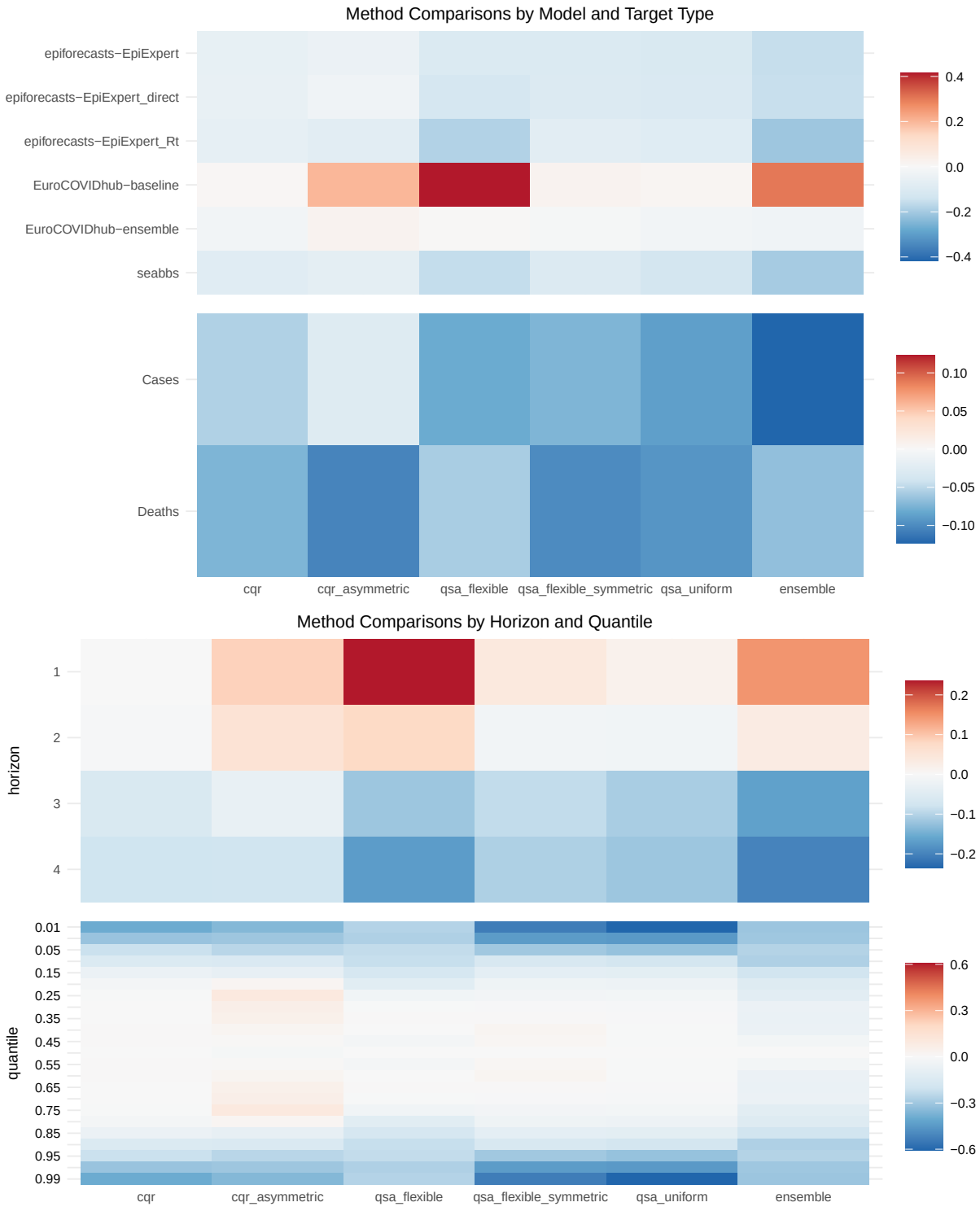
Table 2: Fraction of Feature Combinations where largest Ensemble Weight exceeds Threshold

> 0.5	> 0.9	> 0.99
0.93	0.69	0.53

Table 3: Number (Row 1) and Fraction (Row 2) of largest Ensemble Weights for each Method

cqr	cqr_asymmetric	qsa_uniform	qsa_flexible_symmetric	qsa_flexible
72.00	450.00	2	50.00	530.00
0.07	0.41	0	0.05	0.48





Ypma, Jelmer, and Steven G. Johnson. 2022. *Nloptr: R Interface to NLopt.* https://CRAN.R-project.org/package=nloptr.