GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Evaluation and Aggregation of Covid-19 Death Forecasts in the United States

Nikos Bosse

A master thesis in Applied Statistics

Submitted to the Faculty of Business and Economic Sciences at Göttingen University in September 2020

Ackowldedgements

I would like to thank my supervisors, Sebastian Funk and Thomas Kneib for their support and guidance. I thank Sebastian for the amazing opportunity to work in London and the very warm welcome in his group. To Professor Kneib I am thankful for his teachings as well as his exceptional support and kindness throughout the past years I studied in Göttingen.

I am very thankful to the working group at the London School of Hygiene and Tropical Medicine, to Sam, Joel, Kath, James and Sophie for the pleasure I had and have working with them. I especially thank Sam for his support and mentoring and his help with the `scoringutils` package.

I thank Malte, as well as Anne, Felix, Joel, and Sam for reading through parts of my work and giving me feedback.

Yuling Yao has contributed large and important parts to the `stackr` package. Sebastian and researchers at the Forecast Hub have inspired important aspects of the evaluation process discussed in this thesis. Thank you for that.

Thank you to my friends and family for their support. Thank you to my flatmates in Göttingen and London, and especially to Anne, who has become my second home in the UK.

Abbreviations

Abbreviation	Meaning
AD test	Anderson-Darling test for uniformity
CDC	Centers for Disease Control and Prevention
CDF	Cumulative Distribution Function
CRPS	Continuous Ranked Probability Score
LSHTM	London School of Hygiene and Tropical Medicine
MAE	Mean Absolute Error
MCMC	Markov Chain Monte Carlo
QRA	Quantile Regression Average
WIS	Weighted Interval Score

Contents

Ackowldedgements **3**

Abbreviations **5**

1 Introduction **9**

2 Forecasting and evaluation **11**
2.1 An overview of differenct forecast types . . . 11
2.2 The forecasting paradigm . . . 12
2.3 Assessing calibration . . . 13
2.3.1 Calibration and bias . . . 13
2.3.2 Calibration and empirical coverage . . . 13
2.3.3 Calibration and the probability integral transform . . . 15
2.4 Assessing sharpness . . . 15
2.5 Proper scoring rules . . . 17
2.5.1 Log Score . . . 17
2.5.2 (Continuous) Ranked Probability Score . . . 18
2.5.3 Interval score . . . 19
2.6 A proposed evaluation framework . . . 20
2.7 The scoringutils package . . . 20

3 Model aggregation **23**
3.1 Theoretical motivation . . . 23
3.2 The Quantile Regression Average ensemble . . . 24
3.3 The CRPS ensemble . . . 25

4 Data and forecasting models **27**
4.1 Introduction to the COVID-19 Forecast Hub and overview of the data . . . 27
4.2 An overview of the different forecast models . . . 28

5 Results - evaluation and aggregation of Covid-19 death forecasts **33**
5.1 Forecast visualisation . . . 33
5.2 Summarised scores and overall performance . . . 34
5.3 Examining the relationship between individual metrics . . . 36
5.4 Identifying main contributors to the WIS . . . 38
5.5 Identifying external drivers of differences WIS . . . 39
5.6 Understanding model characteristics that drive differences in WIS . . . 44
5.6.1 Bias . . . 44
5.6.2 Coverage . . . 45
5.6.3 PIT histograms . . . 49
5.6.4 Sharpness . . . 50
5.7 Specific analysis of ensemble models . . . 52
5.8 Sensitivity analysis . . . 56

6 Summary and discussion **59**

A Appendix **63**

Bibliography **63**

Chapter 1

Introduction

Policy makers have recently started to rely on forecasts to make decisions. Accurate knowledge of the future is immensely valuable in all sorts of areas from farming to economics to public health. With the rise of the novel coronavirus SARS-CoV-2, statistical forecasting has gathered renewed attention. As the virus has spread over the globe, more and more research teams began forecasting the trajectory of the pandemic to help inform public policy. Several countries like the United States, Germany and the United Kingdom have therefore started to aggregate forecasts from different teams. Among these efforts, the US Forecast Hub (UMass-Amherst Influenza Forecasting Center of Excellence, 2020) is the largest and most visible. Its goal is to collect forecasts, to aggregate them, and to make them available to policy makers and the general public in the best possible way. Two questions have been at the centre of these efforts: The first is "how can we best evaluate the performance of a model?" The second is "how can we combine and aggregate different models to get the best possible prediction?". These two questions will be our guiding questions as well throughout this work.

Objectives

This thesis has three main objectives: To obtain a deeper understanding of model evaluation, to explore ways to aggregate models to ensembles, and to facilitate model evaluation and model aggregation by creating appropriate tools. As a case study we analysed the predictions of eight models submitted to the US Forecast Hub between the 22th of June 2020 and the 3rd of August 2020, as well as three different ways of aggregating these models to ensembles. One of the models, 'epiforecasts-ensemble1', was the model submitted by the working group at the London School of Hygiene and Tropical Medicine (LSHTM) which co-supervised this thesis. While the comparison of different models is interesting in and of itself, the main goal of this analysis is to obtain a deeper understanding of the evaluation metrics and model aggregation techniques studied throughout this thesis. To that end we elucidate the concepts behind model evaluation and discuss a variety of possible evaluation metrics in detail. This theoretical discussion allows us then to thoroughly asses the performance of the Forecast Hub models as well as gain a better understanding of the metrics in an applied setting. In addition to this, we explore ways of aggregating models to ensembles. These two things, evaluation and model aggregation are closely connected, because the metrics used to score forecasts can also be used as a target to guide the formation of an optimal ensemble. Apart from a simple mean ensemble, we look into two approaches to combine individual model predictions. The first, Quantile Regression Averaging (Nowotarski and Weron, 2015), forms an ensemble by minimising the so called weighted interval score (Gneiting and Raftery, 2007). The second method is a novel stacking approach (Yao et al., 2018) that optimises the continuous ranked probability score (Matheson and Winkler, 1976; Gneiting and Raftery, 2007). We discuss these model aggregation techniques, apply them to the eight original forecast models and evaluate the ensemble performance alongside the other models.

Contributions

A number of novel contributions come out of this thesis. The first one is the structured model evaluation approach described and proposed in Chapter 2. Model evaluation has already been discussed extensively in the literature (see e.g. Gneiting and Raftery (2007), Gneiting (2010), Bracher et al. (2020), and Funk et al. (2019)). For most parts, however, this discussion is quite technical in nature. Chapter 2 summarises important aspects from the literature and proposes a structured evaluation approach that can easily be applied even by non-experts. Most of the metrics discussed there have been previously published (Bracher et al., 2020; Funk et al., 2019; Gneiting and Raftery, 2007), but some of them have been adapted or newly developed within the working group at LSHTM and are described here for the first time. The second contribution is the `scoringutils` package (Bosse et al., 2020a) that was developed in the context of this thesis and used to evaluate the predictions from the Forecast Hub. The package implements the metrics discussed in Chapter 2 and greatly facilitates structured model evaluation in a simple and unified workflow. The third contribution is the `stackr` package (Bosse et al., 2020b) that was developed in collaboration with Yuling Yao from the Columbia University in New York. It implements a novel stacking procedure based on the continuous ranked probability score. This technique is described in Chapter 3 and explored in practice in Chapter 5. The fourth contribution is the comprehensive evaluation of eight Forecast Hub models and three different ensembles. This evaluation allows us to obtain a better understanding of the models, as well as of the evaluation metrics themselves.

Structure

The remainder of this thesis is structued as follows: Chapter 2 gives a detailed introduction to forecast evaluation. It introduces and discusses different metrics and proper scoring rules in detail that form the basis for the coming chapters. Chapter 3 is dedicated to model ensembles. It provides an intuition for how different predictive distributions can be combined and describes two different model aggregation techniques that build upon the scoring rules described in Chapter 2. Chapter 4 provides some context for the later analysis in Chapter 5. It describes the Forecast Hub in greater detail, takes a first look at the observed data and gives an overview of the different forecasting models. Chapter 5 applies the tools described in Chapters 2 and 3 to the forecasts from eight different models submitted to the US Forecast Hub. It assesses the individual models and analyses the performance of the different model aggregation techniques. Chapter 5 also examines the sensitivity of the results to different choices of model aggregation and evaluation parameters. Chapter 6 discusses the results and concludes this thesis.

Code

The analysis for this thesis was conducted in R, version 4.0.2 (R Core Team, 2020). All code is publicly available. This includes the code for this thesis[1], for the `scoringutils`[2] and the `stackr`[3] package as well as the code used to create the epiforecasts-ensemble predictions[4] submitted to the Forecast Hub.

[1] github.com/nikosbosse/master_thesis
[2] github.com/epiforecasts/scoringutils
[3] github.com/epiforecasts/stackr
[4] github.com/epiforecasts/covid-us-forecasts

Chapter 2

Forecasting and evaluation

Model evaluation is an integral of the forecasting process that can provide us with valuable insights. It can help us to choose between different models, but also give us a better understanding of how a model works and how it can be improved. The evaluation metrics discussed here also form the basis to combine models into an ensemble that works better than all individual models. This chapter therefore provides the theoretical foundation for the discussion of model aggregation in Chapter 3 and for the analysis presented in Chapter 5. It first gives a brief overview of different types of forecasts to provide a background for the remainder of this chapter. It then reviews the forecasting paradigm as formulated by Gneiting et al. (2005) and Gneiting et al. (2007) that is at the core of forecast evaluation. The forecast paradigm states that a forecaster should aim to maximise the *sharpness* of their forecast subject to *calibration*. We therefore present different ways to assess these two properties, followed by a discussion of proper scoring rules that allow us to summarise the quality of a forecast in a single number. Finally, we propose a structured evaluation framework based on the notions discussed in this chapter and present the `scoringuitls` package in more detail that facilitates the evaluation process. We illustrate the concepts discussed throughout this chapter using examples from the COVIDhub-baseline model, one of the Forecast Hub models. The examples use data from a longer time frame than the one analysed in Chapter 5 and serve illustrative purposes only.

2.1 An overview of differenct forecast types

A forecast is the forecaster's stated belief about the future. In terms of quantitative forecasts we can distinguish point forecasts from probabilistic forecasts. A point forecast states a single number and is the simplest form of a forecast. It can, in essence, be understood as an estimate for the mean of the unknown true data-generating distribution. A point forecast is limited in its usefulness, as it does not state any uncertainty around the mean forecast. A very certain forecasts may warrant a very different course of actions than does a very uncertain ones. Providing uncertainty around a forecast therefore increases its usefulness. Ideally, however, predictions should be stated in terms of the entire predictive distributions (Gneiting and Raftery, 2007). Such a forecast is then called a probabilistic forecast. Providing the entire predictive distribution allows the forecaster to express their belief about all aspects of the underlying data-generating distribution (including e.g. skewness or the width of its tails).

These forecasts can also be reported in different formats that all require slightly different evaluation approaches. The predictive distribution can be expressed analytically, is oftentimes represented by a set of predictive samples from that distribution. This is especially useful as the forecaster can use methods like Markov Chain Monte Carlo (MCMC) algorithms to generate predictions if no analytical expression of the predictive distribution is available. The downside is that predictive samples take a lot of storage space. They also come with a loss of precision that is especially pronounced in the tails of the predictive distribution, where we need quite a lot of samples to characterise the distribution accurately. To circumvent these problems, often quantiles of the predictive distribution

are reported instead. Quantile forecasts can easily be obtained from the explicit analytical form of a probalistic forecasts as well as from predictive samples. A forecaster could also in principle state their forecasts in a binary way by defining an outcome and assigning a probability that the outcome will come true. This type of forecasting is common in many classification problems, but will not be discussed further here as it is not the focus in infectious disease modeling. If we think of the outcome of some number being larger or smaller than a certain value, however, we can see that binary predictions are naturally related to the concept of a cumulative distribution function (CDF).

2.2 The forecasting paradigm

Any forecaster should aim to minimise the difference between the predictive distribution and the unknown true data-generating distribution (Gneiting et al., 2007). For an ideal forecast, we therefore have

$$P_t = F_t,$$

where P_t is the the cumulative density function (CDF) of the predictive distribution at time t and F_t is the CDF of the true, unknown data-generating distribution. As we don't know the true data-generating distribution, we cannot assess the difference between the two distributions directly. Gneiting et al. (2005) and Gneiting et al. (2007) instead suggest to focus on two central aspects of the predictive distribution, *calibration* and *sharpness*. Calibration refers to the statistical consistency between the predictive distribution and the observations. There are different possible ways in which a model can be (mis-)calibrated (Gneiting et al., 2007), but for the remainder of this thesis it suffices to say that a well calibrated forecast does not systematically deviate from the observed values. Sharpness is a feature of the forecast only and describes how concentrated the predictive distribution is, i.e. how precise the forecasts are. The general forecasting paradigm states that we should *maximise sharpness of the predictive distribution subject to calibration*. Take for example the task of predicting rainfall in a city like London. A model that made very precise forecasts would not be useful if the forecasts were wrong most of the time. On the other hand, a model that predicts the same rainfall probability for every day can be correct on average[1], but is also less useful than a model that were able to accurately predict the weather every single day. Figure 2.1 illustrates the concepts of calibration and sharpness once again.

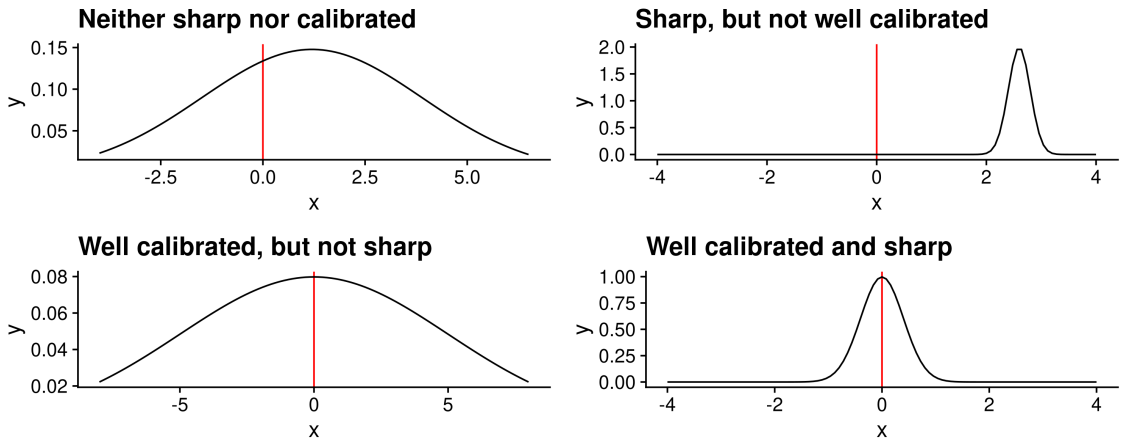

Figure 2.1: Illustration of the forecasting paradigm that states we should maximise sharpness subject to calibration. Shown are different predictive distributions that predict the true value at x = 0.

The following sections look at calibration and sharpness in more detail. We first discuss different ways to assess these two properties independently. Then, we introduce proper scoring rules that allow us to represent the quality of a forecast in one numeric value.

[1]To be precise, this model would be marginally calibrated according to Gneiting et al. (2007)

2.3 Assessing calibration

In absence of knowledge of the true data-generating distribution we can never prove calibration, but only look for absence of miscalibration. Several strategies have been proposed to detect systematic deviations of the predictive distributions from the observations (see e.g. Funk et al. (2019); Gneiting et al. (2007); Gneiting and Raftery (2007)). In order to get a clearer picture of the different ways in which a model can be miscalibrated, it makes sense to look at calibration from more than one angle. In the following we explore three different ways to approach calibration. The first one is bias, i.e. systematic over- or underprediction. The second one is empirical coverage. Coverage measures what proportion of the observed values is covered by different parts of the predictive distribution. The third is the probability integral transform (PIT), a transformation of the original observed values that allows us to assess calibration more easily.

2.3.1 Calibration and bias

Systematic over- or underprediction is a very common form of miscalibration. It therefore makes sense to dedicate separate attention to the detection of systematic biases. We present three different bias metrics, each slightly adapted for continuous, integer and quantile forecasts.

For continuous forecasts, assessing whether a predictive distribution has a tendency to over- or underpredict can be very easily achieved by simply evaluating the predictive distribution at the true observed value. This metric is a generalisation of the integer-valued one Funk et al. (2019) have proposed. It is also closely related to the probability integral transform (PIT) discussed later in this chapter. To improve the interpretability of the score we can transform it to a value between -1 (under-prediction) and 1 (over-prediction). Consequently, we measure bias as

$$B_t(P_t, x_t) = 1 - 2 \cdot (P_t(x_t)),$$

where P_t is the cumulative distribution function of the predictive distribution for the true value x_t. When using predictive samples, $P_t(x_t)$ is simply the fraction of predictive samples for x_t that are smaller than the true observed x_t.

For integer valued forecasts, we use the metric proposed by Funk et al. (2019):

$$B_t(P_t, x_t) = 1 - (P_t(x_t) + P_t(x_t + 1)).$$

Bias can again assume values between -1 (under-prediction) and 1 (over-prediction) and is 0 ideally.

For quantile forecasts, we propose the following metric to assess bias:

$$\begin{aligned} B_t = &(1 - 2 \cdot \max\{i | q_{t,i} \in Q_t \wedge q_{t,i} \leq x_t\}) \mathbb{1}(x_t \leq q_{t,0.5}) \\ &+ (1 - 2 \cdot \min\{i | q_{t,i} \in Q_t \wedge q_{t,i} \geq x_t\}) \mathbb{1}(x_t \geq q_{t,0.5}), \end{aligned}$$

where Q_t is the set of quantiles that form the predictive distribution at time t. They represent our belief about what the true value x_t will be. For consistency, we define Q_t such that it always includes the element $q_{t,0} = -\infty$ and $q_{t,1} = \infty$. $\mathbb{1}()$ is the indicator function that is 1 if the condition is satisfied and 0 otherwise. In clearer terms, B_t is defined as the maximum percentile rank for which the corresponding quantile is still below the true value, if the true value is smaller than the median of the predictive distribution. If the true value is above the median of the predictive distribution, then B_t is the minimum percentile rank for which the corresponding quantile is still larger than the true value. If the true value is exactly the median, both terms cancel out and B_t is zero. For a large enough number of quantiles, the percentile rank will equal the proportion of predictive samples below the observed true value, and this metric coincides with the one for continuous forecasts. Figure 2.2 exemplifies a possible visualisation of bias for one-week-ahead predictions made by the COVIDhub-baseline model.

2.3.2 Calibration and empirical coverage

Another way to look at calibration[2] is to compare the proportion of observed values covered by different parts of the predictive distribution with the nominal coverage implied by the CDF of the

[2]precisely: probabilistic calibration in Gneiting et al. (2007)

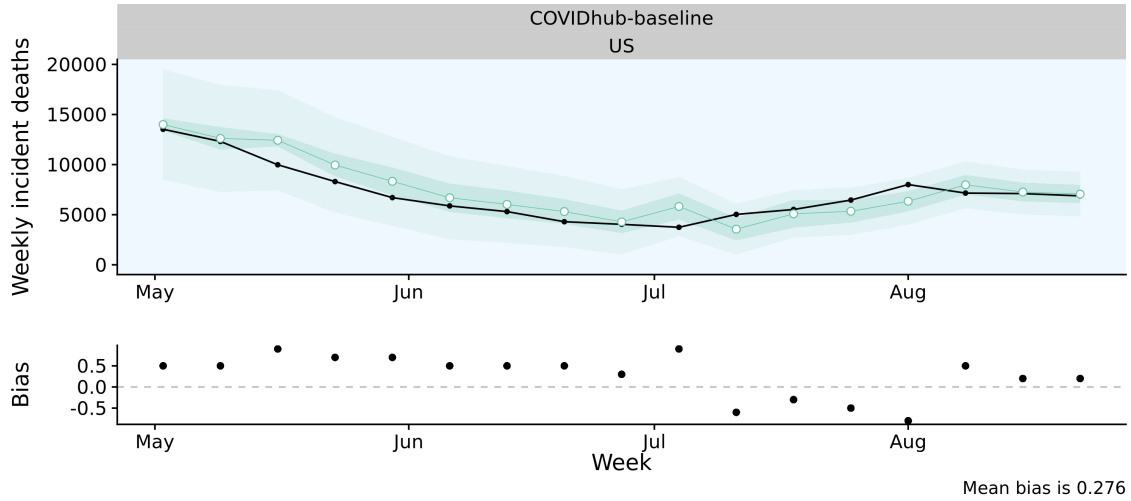

Figure 2.2: One week ahead forecasts from the COVIDhub-baseline model for the US (top) and corresponding bias values (bottom). Observations are shown in black, median predictions ar marked by white points, ribbons show the 50 percent and 90 percent prediction intervals.

distribution. This is most easily understood in the context of quantile forecasts, but can in principle be transferred to continuous and integer forecasts as well.

To assess empirical coverage at a certain interval range, we simply measure the proportion of true observed values that fall into corresponding range of the predictive distribution. If the 0.05, 0.25, 0.75, and 0.95 quantiles are given, then 50% of the true values should fall between the 0.25 and 0.75 quantiles and 90% should fall between the 0.05 and 0.95 quantiles. We can calculate and plot these values to inspect how well different parts of the forecast distribution are calibrated. This is illustrated in the left plot in Figure 2.3 where the empirical coverage of the US forecasts of the COVIDhub-baseline model is shown. We can seen that the interval coverage for the outer prediction intervals looks reasonable, but inner prediction intervals seem to miss too many observations.

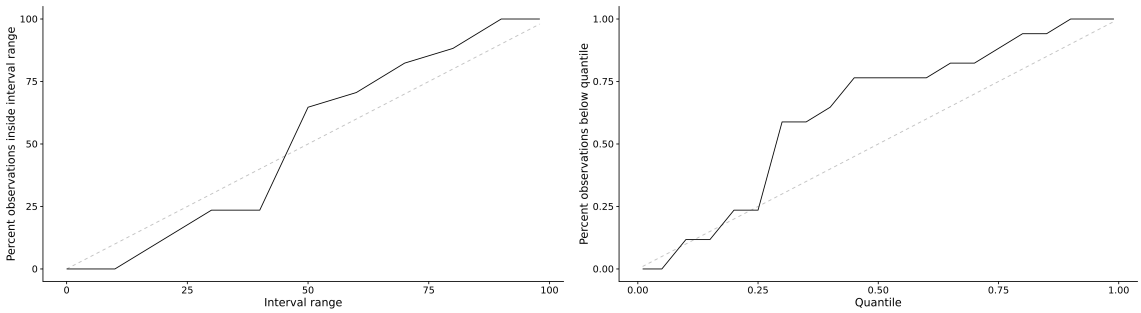

Figure 2.3: Empirical coverage of prediction intervals (left) and coverage of quantiles of the predictive distribution (right) for one week ahead forecasts from the COVIDhub-baseline model.

To get an even more precise picture, we can also look at the percentage of true values below every single quantile of the predictive distribution. This type of visualisation allows us to diagnose problematic aspects more accurately, as is illustrated on the right in Figure 2.3. This plot allows us to describe the problem more precisely as an upward bias of large parts of the predictive distribution, but not the lower tails.

2.3.3 Calibration and the probability integral transform

As explained previously, the CDF of predictice distribution P_t should ideally be equal to the CDF of the true unknown distribution F_t that generated the observed value x_t. In order to assess whether there are substantial deviations between the two, Dawid (1984) suggested to transform the observed values using the probability integral transform (PIT). Agreement between the forecasts and the observed values can then be examined by observing whether or not the transformed values follow a uniform distribution. The PIT is given by

$$u_t = P_t(x_t),$$

where u_t is the transformed variable and $P_t(x_t)$ is the predictive distribution evaluated at the true observed value x_t. If $P_t = F_t$ at all times t, then $u_t, t = 1 \dots T$ follows a uniform distribution (for a proof see e.g. Angus (1994)).

In the case of discrete outcomes, the PIT is no longer uniform even when forecasts are ideal. As Funk et al. (2019) suggest, we use a randomised PIT instead by redefining

$$u_t = P_t(x_t) + vt \cdot (P_t(x_t) - P_t(x_t - 1)),$$

where x_t is again the observed value at time t, $P_t()$ is the CDF of the predictive distribution function, $P_t(-1) = 0$ by definition, and v_t is a standard uniform variable independent of x_t. If P_t is equal to the true data-generating distribution function, then u_t is standard uniform. Czado et al. (2009) also propose a non-randomised version of the PIT for count data that could be used alternatively.

One can then plot a histogram of u_t values to look for deviations from uniformity. U-shaped histograms often result from predictions that are too narrow, while hump-shaped histograms indicate that predictions may be too wide. Biased predictions will usually result in a triangle-shaped histogram. Figure 2.4 shows four different simulated example PIT histograms that illustrate these characteristics.

In addition to the visual inspection, Funk et al. (2019) suggest to apply an Anderson-Darling (Anderson and Darling, 1952) test for uniformity to the transformed values. The test cannot prove uniformity, but only assess whether there is evidence against it. As a rule of thumb, Funk et al. suggest there is no evidence to call a forecasting model miscalibrated if the p-value found was greater than a threshold of $p \geq 0.1$, some evidence that it is miscalibrated if $0.01 < p < 0.1$, and good evidence that it is miscalibrated if $p \leq 0.01$. This approach, however, may be overly conservative[3] and should not be used as the sole criterion to judge calibration of a forecast.

Hamill (2001) discusses in length that uniformity of the PIT histogram is a necessary, but not a sufficient condition for calibration. Nevertheless, the PIT histogram can give us a good impression of the reliability of our forecasts. Figure 2.5 shows the pit histogram for one-week-ahead predictions made by the COVIDhub-baseline model across different states. We can see that the PIT histogram presents a pattern that suggests under-dispersion may be present, i.e. at least some of the confidence intervals may be too narrow. This corresponds well to the observation made in the coverage plot in Figure 2.3.

2.4 Assessing sharpness

Sharpness is the second property central to model evaluation. The ability to produce narrow forecasts is a quality of the forecasts only and does not depend on the observations. Sharpness

[3]To test this, we ran a small simulation study with i = 1 000 iterations. For every iteration, n = 1 000 true values were simulated from a standard normal distribution. Each of these true values was then 'predicted' using s = 10 000 samples from the same standard normal distribution. For every iteration i we therefore obtained 1 000 true values and 1 000 predictive distributions (with 10 000 samples each). These 1 000 true values very transformed using the probability integral transform, which again yielded 1 000 transformed values for every iteration. On these values, we then applied the Anderson-Darling test for uniformity and recorded the p-value. Out of 1000 iterations, there were 165 p-values ≤ 0.01, 88 p-values $0.01 < p < 0.1$ and 747 p-values ≤ 0.01. Note that 1000 true values is quite high for many applied settings. For n = 100 true_values and s = 2000 samples, the result was 108, 87 and 805. Based on this limited evidence can we can conclude that there is at least a chance that the AD test may be prone to reject a meaningful fraction of well calibrated predictions.

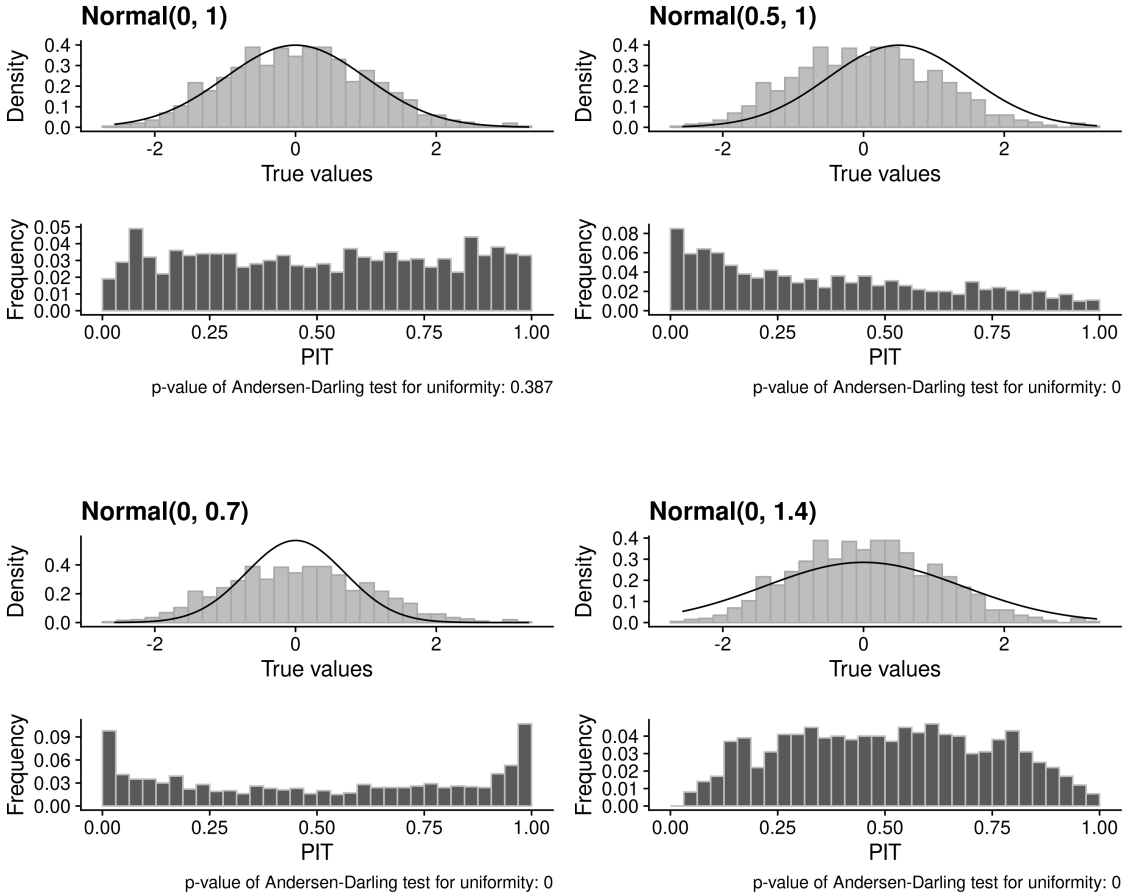

Figure 2.4: Four examples of different predictive distributions and the correspoding PIT histograms below. The data always follows the same Normal(0,1) distribution. Preditive distributions are indicated by the title of the subplot.

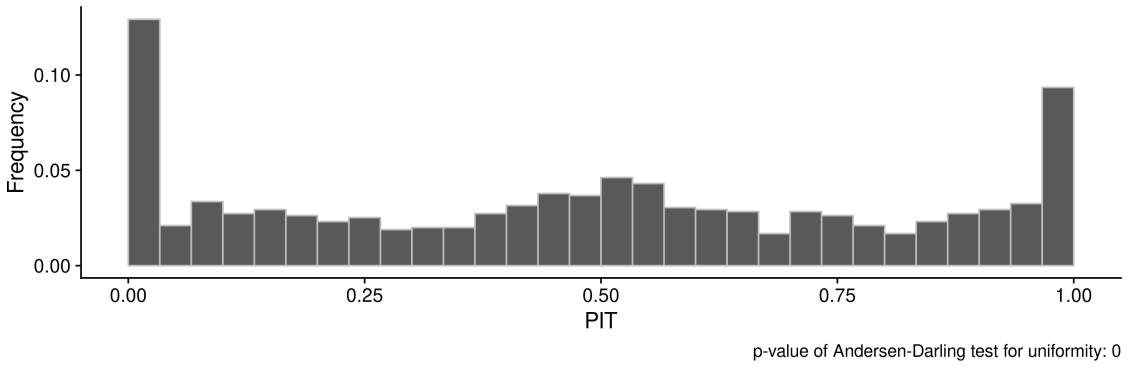

Figure 2.5: PIT histogram for one-week-ahead forecasts from the COVIDhub-baseline model. In order to obtain samples from the quantiles provided, a separate gamma distribution was first fit to every set of quantiles provided to the Forecast Hub using the `nloptr` package (Ypma and Johnson, 2020). Samples were then drawn from these distributions. We can see a pattern commonly found in underdispersed predictions.

is therefore only of interest conditional on calibration: a very precise forecast is not useful if it is clearly wrong. Again we need to take slightly different approaches for continuous, integer and quantile forecasts. For continuous and integer forecasts we follow the suggestion from Funk et al. (2019). For quantile forecasts we propose a novel metric.

For continuous and integer forecasts, Funk et al. (2019) suggest to measure sharpness as the normalised median absolute deviation about the median (MADN), i.e.

$$S_t(P_t) = \frac{1}{0.675} \cdot \text{median}(|y - \text{median(y)}|),$$

where y is the vector of all predictive samples and $\frac{1}{0.675}$ is a normalising constant that ensures that sharpness will equal the standard deviation of the predictive distribution if P_t is the CDF of a normal distribution.

For quantile forecasts, we propose to measure sharpness as a weighted mean of the width of the interval ranges. Let Q_t be a set of predicted quantiles for a true x_t at time t. This set of quantiles is assumed to be symmetric, such that there exist K corresponding pairs of elements $q_{t,\frac{\alpha}{2}}$ and $q_{t,1-\frac{\alpha}{2}}$. These K corresponding pairs of quantiles cover a $(1-\alpha)\cdot 100$ prediction interval. We can, accordingly, also denote $q_{t,\frac{\alpha}{2}}$ as l_t the lower bound of the prediction interval at time t and $q_{t,1-\frac{\alpha}{2}}$ as u_t, the upper bound. We measure the sharpness of a quantile forecast at time t as

$$\begin{aligned}\text{sharpness}_t &= \frac{1}{K}\sum_{\alpha}\frac{\alpha}{2}(q_{t,1-\frac{\alpha}{2}} - q_{t,\frac{\alpha}{2}}) \\ &= \frac{1}{K}\sum_{\alpha}\frac{\alpha}{2}(u_t - l_t).\end{aligned}$$

Weighting the width of different intervals with $\frac{\alpha}{2}$ ensures that the score does not grow indefinitely for very large prediction intervals, and correspondingly, very small α. We also argue that this sharpness metric for quantile forecasts is a natural choice for quantile forecasts as it corresponds to the sharpness component of the Weighted Interval Score described in the following section.

2.5 Proper scoring rules

Instead of assessing calibration and sharpness independently, we can make use of proper scoring rules to express the quality of our forecast in a single number. Propriety is a feature of a score that guarantees that the ideal forecast will always on average receive the lowest score (Gneiting and Raftery, 2007). A forecaster judged by a proper scoring rule is therefore always incentivised to make forecasts as close to the true data-generating distribution as possible. Proper scoring rules are closely related to the forecasting paradigm, as any proper scoring rule for finite-valued targets can be decomposed into a component that evaluates sharpness and one that scores calibration (Bröcker, 2009; Hersbach, 2000). Different scoring rules, however, may weigh sharpness and calibration differently and therefore yield different results. The following sections present three different proper scoring rules: the Log Score, the (Continuous) Ranked Probability Score and the (Weighted) Interval Score.

2.5.1 Log Score

The Log Score is one of the oldest proper scoring rules and can be traced back to Shannon (1948) and his work on communication and information theory and to Good (1952) who first proposed a log score for binary predictions. The Log Score is now widely used in many fields, especially in Bayesian inference (Gelman et al., 2014). The log score is simply the log density of the predictive distribution at time t evaluated at the true observed value:

$$\text{log score}_t = \log p_t(x_t),$$

where p_t is the predictive density function at time t. This is illustrated in Figure 2.6. One problem with the log score is that it becomes numerically unstable for values of $p_t(x_t)$ close to zero. The log

score will therefore not play a large role in this thesis, but is mentioned for completeness sake as it of great importance to many applications. It also quite nicely illustrates the concept of looking at observations in terms of the predictive distribution that we already saw in the PIT.

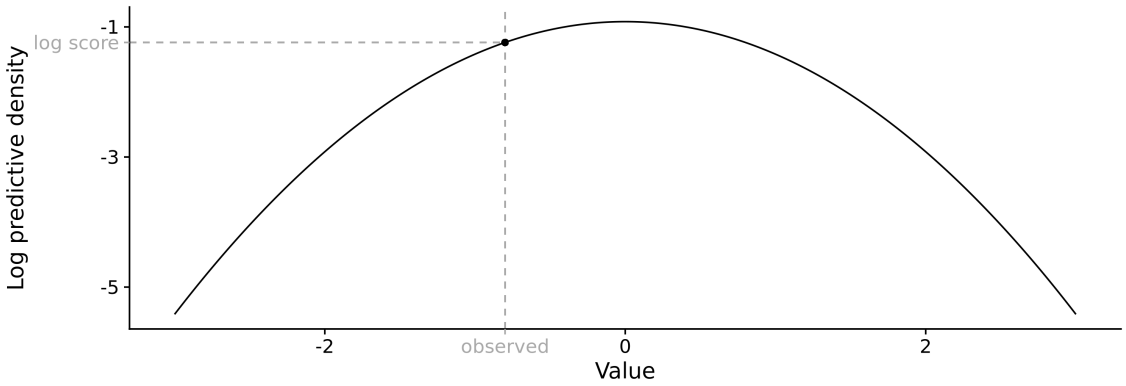

Figure 2.6: Illustration of the Log score as the log predictive density evaluated at the true observed value.

2.5.2 (Continuous) Ranked Probability Score

The Continuous Ranked Probability Score (CRPS) (Matheson and Winkler, 1976; Gneiting and Raftery, 2007) is a proper scoring rule that is considered more stable and therefore better suited for our purposes. Intuitively, we can think of the CRPS as a measure of the distance between the CDFs of the predictive distribution and the data-generating distribution. Smaller values are therfore preferable. The CRPS is defined as

$$\text{CRPS}(P_t, x_t) = \int_{-\infty}^{\infty} \left(P(y) - \mathbb{1}(y \geq x_t)\right)^2 dy, \tag{2.1}$$

where P_t is again the CDF of the predictive distribution and x_t is the true observed value.

The CRPS can also be expressed as

$$\text{CRPS}(P_t, x_t) = \frac{1}{2}\mathbb{E}_{P_t}|X - X'| - \mathbb{E}_P|X - x_t|,$$

where X and X' are independent realisations from the predictive distributions P_t with finite first moment (Gneiting and Raftery (2007)). This formulation is convenient as we can simply replace X and X' with predictive samples and sum over all possible combinations to obtain the desired sample CRPS.

For integer counts, we can use the Ranked Probability Score (RPS) as proposed by Epstein (1969) and Murphy (1969), and discussed e.g. by Czado et al. (2009). The RPS is defined as

$$\text{RPS}(P_t, x_t) = \sum_{y=0}^{\infty} (P_t(y) - \mathbb{1}(y \geq x_t))^2.$$

Figure 2.7 gives an intuitive illustration of the CRPS. For the case of a point prediction, as shown in the top half, the CRPS is equal to the Mean Absolute Error (MAE) of the point forecast. In this case the predictive distribution degenerates to a distribution with its entire mass on the single predicted point and the CDF of the predictive distribution becomes a step function. The CRPS then equals the area between the true observed value and the predicted value (as $1^2 = 1$ and the height of the rectangle is again 1). For other distributions, as shown in the bottom half, this does not hold exactly, as $P_t()$ is squared in Equation (2.1). Intuitively, we can nevertheless understand the CRPS as a measure related to the vertical distance between the predictive CDF and the true data-generating distribution.

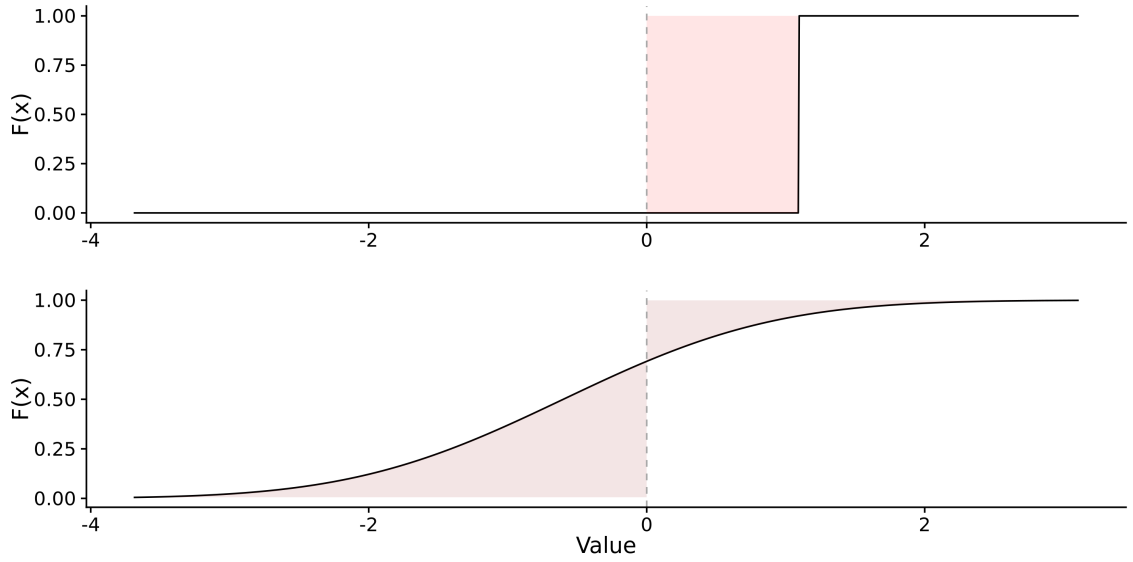

Figure 2.7: Illustration of the CRPS. Top: CRPS for a predictive distribution with its entire mass on the predicted value, 1. The CRPS corresponds to the mean absolute error of the point prediction. This is the absolute difference between the predicted value, 1, and the true observed value, 0, as the height of the shaded square is one. Bottom: llustration that gives an intuition of the CRPS as a measure related to the vertical distance between the CDF and the true value. Note that the CRPS does not in fact equal the shaded area, as the term in Equation (2.1) is squared.

2.5.3 Interval score

The Interval Score is a proper scoring rule to evaluate predictions in a quantile format (Bracher et al., 2020; Gneiting and Raftery, 2007). Let us consider a pair of predictive quantiles $q_{\frac{\alpha}{2}}, q_{1-\frac{\alpha}{2}}$ that form a $(1-\alpha)*100$ prediction interval. Let us denote $q_{\frac{\alpha}{2}}$, the lower bound, as l, and $q_{1-\frac{\alpha}{2}}$, the upper bound, as u. Then the Interval score is given as

$$\text{IS}_\alpha = (u-l) + \frac{2}{\alpha} \cdot (l-x) \cdot \mathbb{1}(x \leq l) + \frac{2}{\alpha} \cdot (x-u) \cdot \mathbb{1}(x \geq u).$$

This score can be separated into three parts: $(u-l)$ measures the sharpness of the predictive distribution. $\frac{2}{\alpha} \cdot (l-x) \cdot \mathbb{1}(x \leq l)$ and $\frac{2}{\alpha} \cdot (x-u) \cdot \mathbb{1}(x \geq u)$ are penalties that occur if the observed value falls below the lower or above the upper end of the interval range. These over- and underprediction penalties give us a second way to characterise the bias of a forecast. Whereas the bias metric presented previously looked for the innermost quantile that included the observation, the Interval Score captures the actual difference between the forecast and the observed value whenever a forecast falls outside the prediction interval. For the median prediction[4], forecast has a sharpness of zero and the Interval Score is proportional to the MAE.

Usually, more than one predictive interval is reported at once. For a set of K quantile pairs, we can obtain the overall Interval Score as a (weighted) average of individual Interval Score contributions:

$$\text{IS} = \frac{1}{K} \sum_\alpha w_\alpha \cdot \text{IS}_\alpha.$$

This Interval Score is proper for any non-negative choice of w_α. For $w_\alpha = \frac{\alpha}{2}$, one can show that the Interval Score converges to the CRPS for an increasing set of equally spaced prediction intervals (Bracher et al., 2020). It is this particular version of the Interval Score with $w_\alpha = \frac{\alpha}{2}$ that we refer to when we mention the 'Weighted Interval Score' (WIS) or simply 'Interval Score' throughout the remainder of this thesis.

[4]To simplify notation, we treat the the median as a 0% interval range represented by the pair of quantiles $(q_{\frac{1}{2}}, q_{\frac{1}{2}})$.

2.6 A proposed evaluation framework

The previous sections have introduced a variety of different metrics that can help to highlight different important aspects of model performance. Based on these individual building blocks we propose the following structured approach to model evaluation:

1. Visualise the observed data as well as the predictions to get a feeling for how individual models perform. Revisit these plots at each step of the evaluation process.
2. Obtain an overall model ranking and a first indication of where potential problems with individual models lie. To that end, look at summarised scores for all metrics and proper scoring rules. Apply a mixed-effects regression with the Interval Score s as dependent variable and with models and other appropriate variables as predictors. To our knowledge, Jacob Bien and Evan Ray were the first to propose this framework in an informal setting as part of their work for the Forecast Hub. The regression can be very helpful in terms of model selection and differentiation, but also as a first step to identify the main drivers of performance differences, e.g. between different locations.
3. Analyse the main contributors and factors that drive differences in scores. It makes sense to divide this analysis into external drivers, like different locations or time points that cause difficulties to all models, and innate properties of the models themselves.
    a) Look at external drivers by identifying settings (e.g. locations, forecast dates) that induce models to be more or less calibrated or seem to be harder to forecast.
    b) Identify model characteristics that lead to better or worse performance. To that end, examine plots for bias, coverage, and sharpness as well as PIT histograms in detail. This analysis can provide important feedback for model improvement.
4. Look again at visualisations of predictions versus observations to sense check the evaluation results and analysis aspects in detail that are left unclear.

This structured evaluation approach will guide the evaluation of the Forecast Hub models and the ensembles in Chapter 5.

2.7 The scoringutils package

The structured model evaluation approach outlined in the last section is greatly facilitated through the `scoringutils` package (Bosse et al., 2020a). The package makes all metrics described in this chapter easily available to the user who can automatically apply the appropriate metrics to a forecast. It also allows for simple aggregation over arbitrary subgroups which makes summarising, plotting, and fitting a regression very convenient.

The stable version of the package is available on CRAN, the development version can be found on github[5]. Internally, `scoringutils` uses `data.table` (Dowle and Srinivasan, 2019) to allow for an efficient handling of large data.frames. The package is extensively documented, has example data, and a vignette that walks the user through all relevant steps.

Evaluation metrics in the package can be accessed in two different ways. They can either be used independently from each other in a format built around vectors and matrices. Alternatively, users can decide to have forecasts automatically scored in a `data.frame` format through the function `eval_forecast`.

The function `eval_forecast` takes in a `data.frame` with predictions and true observed values. Users then specify the unit of a single observation with the `by` argument that takes in a character vector with different column names. If predictions are for example made on different forecast dates by several models for several locations over different horizons, then the user should specify `by = c("model", "forecast_date", "location", "horizon")`. Scores can be aggregated over different groups using `summarise_by`. If we were only interested in the score per model, we would specify `summarise_by = c("model")`. The `quantiles` argument allows us to summarise the aggregated scores by a set of quantiles. This is especially useful for plotting.

[5] github.com/epiforecasts/scoringutils

The following snippet shows an example evaluation that uses toy data from the package:

```
quantile_example <- data.table::setDT(scoringutils::quantile_example_data_long)
print(quantile_example, 3, 3)
```

```
##      true_values id  model predictions boundary range horizon
##   1:    2.659261  1 model1  -0.6448536    lower    90       1
##   2:    2.659261  1 model1   0.3255102    lower    50       1
##   3:    2.659261  1 model1   1.0000000    lower     0       1
##  ---
## 718:   30.189608 30 model2  31.3873685    upper    90       2
## 719:   30.189608 30 model2  30.6399809    upper    50       2
## 720:   30.189608 30 model2  31.2576984    upper     0       2
```

```
eval <- scoringutils::eval_forecasts(quantile_example,
                                     by = c("model", "id", "horizon"),
                                     summarise_by = c("model", "range"))
print(eval)
```

```
##     model range interval_score    sharpness is_underprediction
## 1: model1     0      0.8879926  0.065132463        0.300326488
## 2: model1    50      0.7760589  0.304421447        0.179681560
## 3: model1    90      0.2658170  0.152391127        0.024935181
## 4: model2     0      0.9215835 -0.003467508        0.298074288
## 5: model2    50      0.6787509  0.356631485        0.072721303
## 6: model2    90      0.2721723  0.160614315        0.008071852
##    is_overprediction calibration coverage_deviation      bias
## 1:        0.52253362  0.11666667         0.11666667 0.1912281
## 2:        0.29195591  0.40000000        -0.10000000 0.1912281
## 3:        0.08849074  0.81666667        -0.08333333 0.1912281
## 4:        0.62697674  0.08333333         0.08333333 0.2771930
## 5:        0.24939811  0.53333333         0.03333333 0.2771930
## 6:        0.10348616  0.85000000        -0.05000000 0.2771930
```

The example data has forecasts from two different models for two different forecast horizons (e.g. one and two weeks ahead into future) and thirty days (denoted by the column 'id') as well as the corresponding observed values. The unit of a single observation is therefore `c("model", "id", "horizon")`. With the `summarise_by` argument we can specify that we want to average over horizons and time points. We then obtain one average score per model, separated for the different interval ranges. In addition to the mean score, we could have obtained arbitrary quantiles and the standard deviation using the `quantile` argument or the `sd` argument.

The following metrics are returned: 'interval_score' refers to the Weighted Interval Score, 'sharpness', 'is_underprediction', and 'is_underprediction' are its three components that together sum up to the WIS. 'Calibration' refers to the coverage achieved by the respective interval range. The column 'coverage_deviation' is calculated as empirical coverage - desired interval coverage and denotes the deviation of empirical interval coverage from the nominal interval coverage. In Chapter 5 we report coverage deviation instead of empirical coverage, as it does not really make sense to average over the empirical coverage for different ranges.

Chapter 3

Model aggregation

Following the discussion of proper scoring rules in Chapter 2 we go one step further in this chapter and explore how we can combine individual models to optimal ensembles using these proper scoring rules. This chapter presents two approaches, Quantile Regression Averaging (QRA) that builds upon the weighted interval score and a novel stacking approach that forms an ensemble that minimises the continuous ranked probability score. Prior to discussing these model aggregation approaches, this Chapter provides an intuition for why model ensembles can improve predictive performance and for how different predictive distributions can be combined.

3.1 Theoretical motivation

Oftentimes, a single forecasting model is not able to capture the nuances and full complexity of the true data-generating process. Different models have different strengths and weaknesses and often can represent some aspects well that other models may miss. This allows us to increase predictive performance by aggregating individual models.

As an illustrative example, let us consider a number of models that each try to predict one single unknown true value. All models are correct on expectation (i.e. $\mathbb{E}(\hat{y}_k) = y$), but all exhibit an independent prediction error (i.e. $\hat{y}_k = y + \epsilon_k$). Every single model will be biased (ϵ_k has zero probability of being exactly zero, even if $\mathbb{E}(\epsilon_k) = 0$). The average of all these model predictions, however, will converge to the true value for an increasing number of models. This is, in very simplified terms, the idea behind the mean ensemble. Note that in practice, prediction errors of different models are seldom completely independent, but are instead often correlated, which may reduce the effectiveness of model aggregation. Adding a model to the ensemble of course only makes sense if we indeed believe that its predictions do not deviate systematically from the true values. However, in many circumstances this decision will not be clear cut. Instead, we could decide to give less weight to models that have performed poorly in the past and more weight towards those that performed well. This is the idea behind weighted ensembles.

In the following, we discuss two different ways of combining predictive distributions, the quantile average and the mixture distribution. The quantile average aligns all corresponding quantiles of the predictive distributions and determines the quantiles of the ensemble distribution as a weighted average of the corresponding ensemble member quantiles. We can understand this as a horizontal combination of the cumulative distribution functions (CDF) of the individual predictive distributions. A mixture distribution, on the other hand, can be understood as a vertical combination of the CDFs of the individual ensemble member distribution. It may, however, be more intuitive to think of it as the result of random sampling from the individual ensemble member distributions. The mixture distribution can again be weighted by drawing from the individual distributions with different probabilities. These ideas are illustrated in Figure 3.1. In principle, the average seems more appropriate if we there is one single future scenario and we want to optimally predict it. The mixture may be better suited if we believe that our models reflect different possible future scenarios

and are uncertain, which of these scenarios will occur.

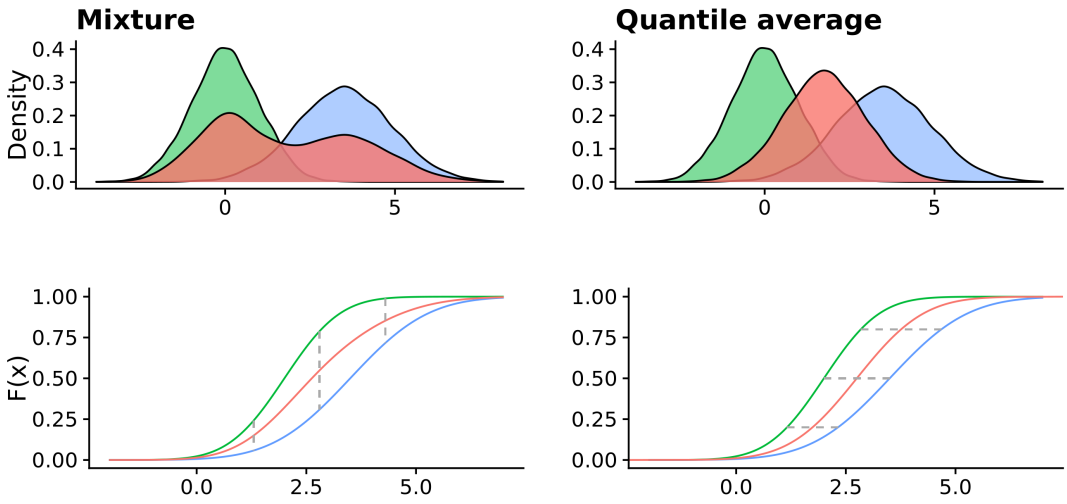

Figure 3.1: Two different ways of combining two predictive distributions (based on predictive samples). The left is a mixture distribution (red) generated by taking random samples with equal probability from the two original distributions (green and blue). The right is a quantile average that was generated by taking the pairwise mean of the sorted vectors of the predictive samples from the two distributions. The lower two plots show the same combinations in terms of averages of the cumulative distribution functions. The mixture can be thought of as a vertical combination of CDFs, while the quantile average is a vertical combination.

In the following, we present two different ensemble formation strategies. The Quantile Regression Average (QRA) is an ensemble strategy suited for quantile forecasts that determines optimal weights for a weighted quantile average. The CRPS ensemble works with predictive samples and determines optimal weights for a mixture distribution.

3.2 The Quantile Regression Average ensemble

The Quantile Regression Average (Nowotarski and Weron, 2015) is an ensemble strategy build upon the weighted interval score presented in Chapter 2. Consider a forecast made for observation $i, i = 1, \ldots, n$ by model $k, k = 1, \ldots, K$ at different quantile levels $\tau \in (0, 1)$. The corresponding quantile prediction for observation i from model k at quantile level τ is denoted $q_{ik\tau}$. The ensemble prediction at quantile level τ, $q_{i,\text{ensemble},\tau}$ is then a weighted average of the corresponding predictive quantiles of the individual models:

$$q_{i,\text{ensemble},\tau} = \sum_{k=1}^{K} w_k \cdot q_{ik\tau},$$

where w_k is the weight given to model k. The weights are usually constrained to be non-negative and to sum up to one. To get an optimal ensemble, we are looking for the combination of weights that, across all quantile levels τ, produce an ensemble which minimises the weighted interval score over past observations. The optimisation problem can be denoted as follows (Tibshirani, Ryan, 2020):

$$\arg\min_{w} \sum_{i=1}^{n} \sum_{\tau} \psi_{\tau} \left(y_i - \sum_{k=1}^{K} w_k q_{ik\tau} \right),$$

where $\psi_\tau()$ denotes the so-called pinball loss at quantile level τ. The pinball loss is defined as

$$\psi_\tau(x) = \max(\tau \cdot x, (\tau - 1) \cdot x).$$

The solution to this minimisation problem yields the ensemble that minimises past weighted interval scores. This optimisation problem can be extended in a number of ways. For example, one can estimate different weights for different quantile levels or one can incorporate additional constraints, e.g. that quantiles not cross. The minimisation problem (including the additional constraints) can conveniently be solved using the `quantgen` package (Tibshirani, 2020).

3.3 The CRPS ensemble

Instead of the weighted interval score, we can also use the CRPS as a basis for an ensemble formation approach. The major conceptual advantage of using CRPS and predictive samples is that we can create a mixture distribution instead of a quantile average. The approach described in the following is a form of stacking (see Yao et al. (2018)) that is in theory optimal even if the true data-generating distribution is not among the individual ensemble distributions. While many other strategies like Bayesian Model Averaging eventually (Raftery et al., 1997; Hoeting et al., 1999; Raftery et al., 2005) converge to putting all their weights to the single model that is closest to the true data-generating distribution, stacking is able to combine information from all models to form an optimal ensemble.

This CRPS ensembling approach was developed in collaboration with Yuling Yao from the Columbia University in New York and is implemented in the R package `stackr` (Bosse et al., 2020b). The following method overview is based on work written by Yuling Yao and can also be found in the `stackr` vignette.

As stated in Equation (2.1) in Chapter 2, the CRPS for a predictive distribution with finite first moment and the corresponding true value y is given by

$$crps(F, y) = \mathbb{E}_X|X - y| - \frac{1}{2}\mathbb{E}_{X,X'}|X - X'|.$$

The notation is slightly altered in comparison to Chapter 2 to keep consistency with the `stackr` vignette and also to avoid having x denote samples as well as observations. The predictive distribution is denoted by F and true observed values are donated by y. Let us assume we have data from T time points $t = 1, \dots, T$ in R regions $r = 1, \dots, R$. Observations are denoted y_{tr}. Predictive samples are generated from K different models $k = 1, \dots, K$. For every observation y_{tr} the S predictive samples $s = 1, \dots, S$ are denoted $x_{1ktr}, \dots, x_{Sktr}$.

Let us first look at the CRPS for one observation and one predictive model before deriving the CRPS of a mixture of all models. Based on the predictive samples, we can compute the CRPS of the k-th model for the observation y_{tr} at time t in region r as

$$\begin{aligned}\widehat{\text{crps}}_{ktr} &= \widehat{\text{crps}}(x_{1ktr}, \dots, x_{Sktr}, y_{tr}) \\ &= \frac{1}{S}\sum_{s=1}^{S}|x_{sktr} - y_{tr}| - \frac{1}{2S^2}\sum_{s,j=1}^{S}|x_{sktr} - x_{jktr}|.\end{aligned}$$

Now we want to aggregate predictions from these K models. When the prediction is a mixture of the K models with weights $w_1, \dots, w_s$, the CRPS can be expressed as

$$\begin{aligned}\widehat{\text{crps}}_{\text{ensemble},tr}(w_1, \dots, w_K) =& \frac{1}{S}\sum_{k=1}^{K} w_k \sum_{s=1}^{S}|x_{skt} - y_t| \\ &- \frac{1}{2S^2}(\sum_{k=1}^{K}\sum_{k,k'=1}^{K} w_k w_{k'} \sum_{s,j=1}^{S}|x_{skt} - x_{jk't}|).\end{aligned}$$

The overall CRPS for the mixture of all models for all observations can then simply be obtained by summing up the individual CRPS contributions from the different pairs of observations and predictions over all regions and time points. We can extend this framework by assigning different weights to different time points and regions. This makes sense for example if we want to assign less weight to older observations because we believe they are less characteristic of the current and future

dynamics. Similarly, we might want to give more or less weight to certain regions. Mathematically we can introduce a time-varying weight $\lambda_1, \dots, \lambda_T$, e.g. $\lambda_t = 2 - (1 - t/T)^2$ to penalize earlier estimates. Likewise we can introduce a region-specific weight τ_r.

To obtain the optimal CRPS weights we finally solve a quadratic optimisation:

$$\min_{w_1,\dots,w_K} \sum_{t=1}^{T} \sum_{r=1}^{R} \lambda_t \tau_r \widehat{crps}_{\text{ensemble},tr}(w),$$

$$s.t.\ 0 \leq w_1, \dots, w_K \leq 1, \sum_{k=1}^{K} w_k = 1.$$

In `stackr`, this is implemented using the `optimizing` function from the `rstan` (Guo et al., 2020) package. To speed up computation, the terms $\sum_{s=1}^{S} |x_{skt} - y_{tr}|$, $\sum_{s,j=1}^{S} |x_{sktr} - x_{jktr}|$, and $\sum_{s,j=1}^{S} |x_{sktr} - x_{jk'tr}|$ are only computed once for all k, k' pairs. Currently, `stackr` does not yet support different forecast horizons, but instead one horizon has to be chosen to optimise for.

After having obtained the mixture weights we can now obtain the final mixture by drawing samples from the individual member distribution distributions with probability equal to the weight asssigned to the corresponding model. This is implemented in the function `mixture_from_samples` in `stackr`. The following code snippet illustrates how a CRPS ensemble can be obtained using `stackr`:

```
splitdate <- as.Date("2020-03-28")
data <- data.table::setDT(stackr::example_data)
print(data, 3, 3)
```

```
##          geography model sample_nr       date   y_pred    y_obs
##      1:  Tatooine ARIMA         1 2020-03-14 1.719445 1.655068
##      2:  Tatooine ARIMA         2 2020-03-14 1.896555 1.655068
##      3:  Tatooine ARIMA         3 2020-03-14 1.766821 1.655068
##     ---
## 103998: Coruscant Naive       498 2020-04-08 1.433936 1.543976
## 103999: Coruscant Naive       499 2020-04-08 1.719357 1.543976
## 104000: Coruscant Naive       500 2020-04-08 0.781818 1.543976
```

```
traindata <- data[date <= splitdate]
testdata <- data[date > splitdate]

# Obtain weights based on training data
weights <- stackr::crps_weights(traindata)

# create mixture based on predictive samples in the testing data.
test_mixture <- stackr::mixture_from_samples(testdata, weights = weights)
print(test_mixture, 3, 3)
```

```
##        geography       date    y_pred sample_nr        model
##     1:  Tatooine 2020-03-29 0.7764884         1 crps_mixture
##     2:  Tatooine 2020-03-29 1.0045886         2 crps_mixture
##     3:  Tatooine 2020-03-29 0.8462946         3 crps_mixture
##    ---
## 10998: Coruscant 2020-04-08 0.1022286       498 crps_mixture
## 10999: Coruscant 2020-04-08 0.4481191       499 crps_mixture
## 11000: Coruscant 2020-04-08 0.3028524       500 crps_mixture
```

Chapter 4

Data and forecasting models

After Chapters 2 and 3 have described the theoretical foundations of model evaluation and model aggregation, this chapter provides some context and background for the model evaluation in Chapter 5. We first describe the Forecasting Hub in more detail and offer a first look at the data. We then give a brief overview of the different forecasting models.

4.1 Introduction to the COVID-19 Forecast Hub and overview of the data

The COVID-19 Forecast Hub (UMass-Amherst Influenza Forecasting Center of Excellence, 2020) is a collaboration between the U.S. Centers for Disease Control and Prevention (CDC), academic research groups led by Professor Nicholas Reich at the University of Massachussets Amherst, and different industry partners. Starting on April 13, the consortium collected weekly forecasts for all US states and territories from research teams around the world. Forecasts were submitted every Monday in a probabilistic format. The predictive distribution was represented by the median and eleven prediction intervals ranging from a 10% prediction interval to a 98% prediction interval[1]. Forecasts were made for one to (at least) four week ahead horizons of targets like daily or weekly deaths and case numbers. Out of these targets, we focus only on weekly death incidences.

Eight models were chosen from all Forecast Hub models in a way that attempts to reflect the variety of models submitted to the Hub. Among these models also was the 'epiforecasts-ensemble1' model that was submitted by the working group at the London School of Hygiene and Tropical Medicine that co-supervised this work. The forecasts we analysed were made between June 22nd 2020 and August 3rd 2020 in twelve different US states as well as nationwide. As the ensemble models need at least one week of past data to estimate weights, only forecasts between June 29th and August 3rd were evaluated. Dates and locations were chosen to obtain a complete set of predictions with no missing forecasts for any location or time point. While this isn't strictly necessary, it avoids downstream complications with model aggregation and evaluation that are beyond the scope of this thesis. The ground truth data is provided by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (Dong et al., 2020). Table 4.1 at the end of this chapter gives an overview of the dates, locations and models included. Figure 4.1 shows observed deaths in all thirteen locations. The time period analysed is highlighted in green. We can see that the evolution of deaths exhibits quite different dynamics across different locations. In states like Illinois, Maryland or Massachusetts, deaths were mostly constant or falling. In others, like Arizona, Florida or Texas, deaths showed a strong upwards trend. Data issues have been corrected for the state of New Jersey. On June 25th, New Jersey started counting probable deaths as well and increased their death count by 1 854 probable cases. These cases were then later redistributed to previous days. We can see the effects of this reporting issue in the forecasts of the COVIDhub-baseline model,

[1]The following 23 quantiles were recorded: `c(0.01, 0.025, seq(0.05, 0.95, by = 0.05), 0.975, 0.99)`.

where we observe spikes in the forecasts for New Jersey and the US nationwide in Figures A.1 in the Appendix.

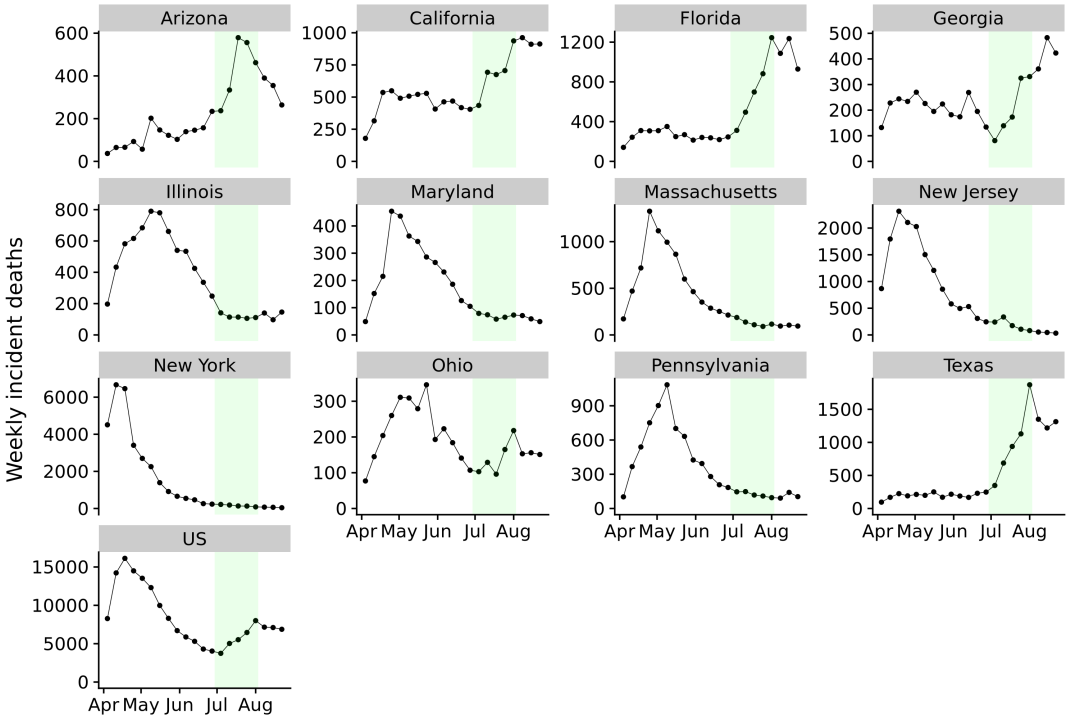

Figure 4.1: Observed deaths in all thirteen locations. Weeks for which predictions were evaluated are highlighted in green.

4.2 An overview of the different forecast models

The models analysed in-depth in the next chapter vary substantially in the way they generate predictions. This section therefore gives a quick overview of the different model types. This overview should not be thought of as an exhaustive discussion, but is merely intended as a short primer that allows us to mentally place the COVID-19 Forecast Hub models in broad categories. The information used to inform this overview is taken from the descriptions provided by the research teams themselves[2].

Among the most widely used models in epidemiology are so-called compartmental models[3]. These divide the overall population into different compartments and model the flow between them. The basic compartments are *Susceptible* (S), *Infectious* (I) and *Recovered* (R), giving these models the name SIR models. The flow from one compartment to the other is usually modeled using a set of differential equations. Compartmental models help to model specific characterisitcs of people in different compartments. For example, People in the *Susceptible* compartment can be infected, while those who are in the *Recovered* compartment are assumed to be immune against further infection. Compartmental models are therefore able to model the depletion of susceptibles as the epidemic progresses. Usually, a compartment called *Exposed but not infectious* (E) is added to model incubation periods, giving these models the name SEIR models. Additional compartments can be included ad libitum to model other aspects of the disease dynamic.

[2]Model descriptions were uploaded by the teams on github.com/reichlab/covid19-forecast-hub/tree/master/data-processed. The descriptions were accessed on July 8th 2020. As descriptions on github were updated whenever the model changes, this overview may therefore not be entirely accurate for all models over the entire history of past submissions included in the analysis.

[3]See e.g. Brauer (2008) for an extensive overview.

Most models submitted to the Forecast Hub were SEIR compartmental models. Three models analysed here belong to that category: UMass-MechBayes, YYG-ParamSearch and CU-select. UMass-MechBayes[4] is a Bayesian model build from a classical SEIR compartmental model. The model is fit independently to each state and allows its parameters to vary over time. The YYG-ParamSearch model[5] adds an additional machine learning layer to a classical SEIR model to learn optimal hyperparameters for the model. Some of the parameters like the infectious period are shared across locations, but most parameters like the mortality from Covid-19 or the effective reproduction number R_t are determined per state. The effective reproduction number R_t is a common parameter in epidemiological models[6] and denotes the average number of people each infected person will infect in turn[7]. This parameter varies over time, for example depending on changes in behaviour or official counter measures put in place. The CU-select model[8] is a SEIR model that is augmented by human insight. The Columbia University regularly submits a range of projections under different scenarios. For the CU-select model they always hand-pick the one they believe is most plausible.

Another common approach is to model the evolution of the pandemic using a time-varying growth rate. Based on the assumption that the spread of a disease is exponential in nature it is intuitive to estimate future case numbers by modeling the growth rate. The LANL-GrowthRate model, and to a certain extent the epiforecasts-ensemble1, follow this approach. The LANL-GrowthRate model is a two component model. The first component models the number of infections using a time-varying growth parameter that connects present (or future) infections to earlier infections and the number of susceptibles. In a second step, this infections get mapped to reported deaths by assuming a fraction of infections likely to die. The epiforecasts-ensemble combines the growth rate approach with classical time series modeling[9]. It consists of three submodels that were aggregated using first an equally weighted quantile average and later a quantile regression average. The three submodels were two time series models and an R_t-based prediction model. The two times eries models were generated using the forecastHybrid package (Shaub and Ellis, 2020). The package automatically selects an Autoregressive Integrated Moving Average (ARIMA) model or a State Space model with appropriate error, trend and seasonality (ETS model). One of the epiforecasts-ensemble1 time series models included current cases as a lagged predictor, while the other was based on deaths only. The third model was generated using the R packages `EpiSoon` (Sam Abbott et al., 2020a), `EpiNow` (Sam Abbott et al., 2020b), and later `EpiNow2` (Sam Abbott et al., 2020c). `EpiSoon` takes in reported cases to estimate the trajectory of R_t. This trajectory is then forecasted into the future using `forecastHybrid` and transformed back to incidences using the renewal equation Cori et al. (2013) that models futures cases as a weighted sum of past cases times R_t. These three models are fit independently to each location.

A third possibility is to model deaths more directly in terms of a regression framework. This the approach chosen by the UT-Mobliity model (Woody et al., 2020)) that employs a Bayesian negative binomial regression model. One of the major predictors used is GPS mobility data provided by a company called SafeGraph. This data is used to compute measures of social distancing that are ultimately fed into the regression model.

In order to provide a sensible baseline to compare models against, the Forecast Hub created the COVIDhub-baseline model. This model basically assumes that incidences will be the same as in the past and models the uncertainty around this estimate according to the distribution of past changes in incidences. More precisely, forecasts for incidences are generated in the following way: The time series of all past incidences is taken and first differences are formed. These first differences, as well as the negative of these incidences are then used further along. From this vector of past changes, samples are drawn to get a predictive distribution of future changes. A predictive distribution for

[4]See github.com/dsheldon/covid.

[5]See covid19-projections.com and github.com/youyanggu/covid19_projections.

[6]To be exact, R_t is usually not an explicit parameter in SEIR models, but can easily be computed from estimated parameters. The information provided by the YYG-ParamSearch model team, however, mentioned R_t explicitly.

[7]See e.g. Nishiura and Chowell (2009) or Cori et al. (2013) for an in-depth discussion

[8]See blogs.cuit.columbia.edu/jls106/publications/covid-19-findings-simulations/.

[9]See for example Hyndman, Rob J and Athanasopoulos, George (2019) for an in-depth overview of different models and approaches.

future incidences is then obtained by adding these samples to the last observed incidence. The predictive distribution is then shifted to enforce that the mean of the distribution equals the last observed value. Incidences below zero are truncated, and quantiles are obtained from the samples.

In addition to the models described above, four model ensembles were analysed. One of these ensembles, the COVIDhub-ensemble was an ensemble created by the Forecast Hub itself using a quantile average approach. This ensemble was formed by taking the arithmetic mean of the corresponding quantiles from all eligible models. Models were deemed eligible if they passed some general sense checks, for example that cumulative forecasts were not decreasing over time. The number of models included varies from state to state as not all teams submitted forecasts for all locations. The other three ensembles were formed from the above described eight original models.

The mean-ensemble is a simple equally weighted quantile average of all original models. All models included in the mean-ensemble are therefore also included in the COVIDhub-ensemble. The mean-ensemble therefore serves as an important control. Any performance difference between the two ensembles can be attributed to the effect of the selection of models included for the analysis.

The qra-ensemble is formed based on the two last weeks of data using the methodology outlined in Chapter 3. The qra-ensemble takes all past forecasts for which we have observations into account. It is therefore created using one and two-week-ahead forecasts. For the first evaluation date, June 29th, only one week of past data was used to inform the ensemble weights. Two weeks of past data were chosen as a sensible default to avoid overfitting. Chapter 5, however, also provides a sensitivity analysis that explores other choices.

The crps-ensemble was also formed using two weeks of past data. But in order to make model aggregation possible, forecasts had to be transformed first. While the CRPSensemble approach described in Chapter 3 is based on predictive samples, the Forecast Hub only stores predictive distributions in a quantile forecast. We therefore fit a separate gamma distribution to every set of predictive quantiles. The gamma distribution was chosen for its simplicity and reasonably good fit. The family of Metalog distributions described by Keelin (2016) where tested as an alternative in the context of a sensitivity analysis described in Chapter 5. The fitted distributions were subsequently used to obtain 1 000 predictive samples per forecast. We then estimated weights based on the predictive samples and created a mixture distribution by random sampling from the individual model samples with probability equal to the estimated weights. The samples drawn for the mixture model were then again used to create quantiles. This approach is not ideal as it is bound to lose information, but overall showed acceptable results. Figure 4.2 shows the difference between quantiles from the actual predictive distributions submitted to the Forecast Hub and sample quantiles obtained by random sampling from a gamma distribution fitted to the same forecasts. This figure includes all forecasts for all locations, forecast dates and quantiles at once. It therefore does not show the full picture, but suggests that the approach works reasonably for most quantiles of the predictive distribution. Only for higher quantiles can we see substantial deviations. As we argue in Chapter 5, these outer quantiles exert only a minor influence on overall performance as measured by the Weighted Interval Score. As `stackr` currently does not support multiple forecast horizons, we chose a two week forecast horizon to optimise for. This effectively meant that the crps-ensemble could not use as much data as the qra-ensemble, since forecasts from the previous week did not yet have matching observations and therefore only the two-week-ahead forecasts from two weeks ago could be used. Again, other possible choices for these parameters are explored in Chapter 5.

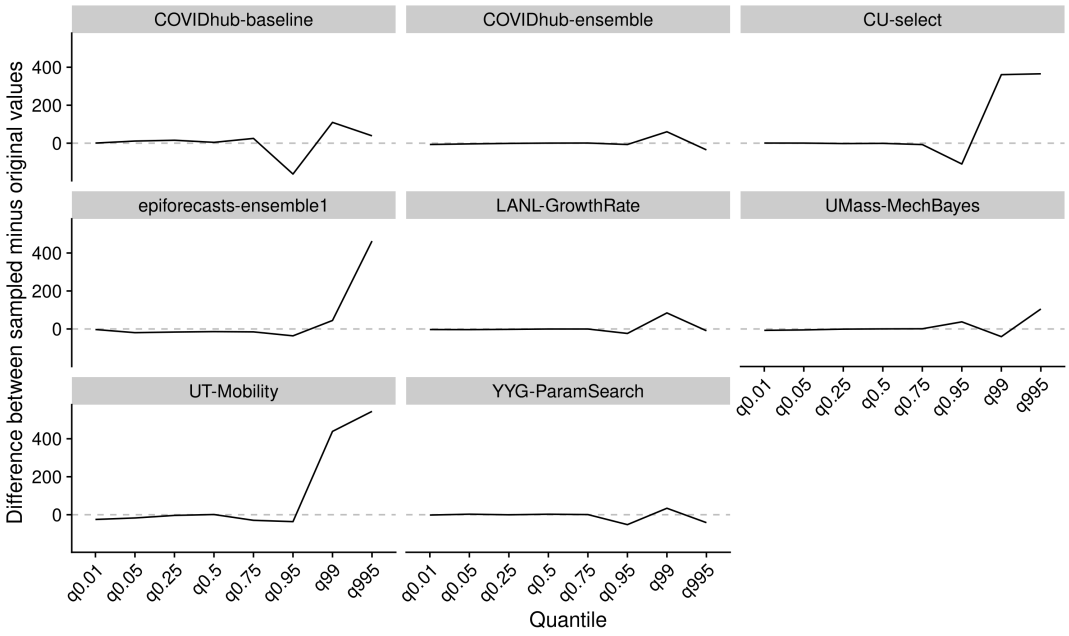

Figure 4.2: Comparison of actual quantiles and the quantiles that were recovered by first fitting a gamma distribution to the quantiles and obtaining samples from that distribution.

Table 4.1: An overview of the dates (left), models (middle) and dates (right) included in the analysis.

dates	models	model summary	locations
2020-06-29	COVIDhub-baseline	Baseline prediction model	Arizona
2020-07-06	COVIDhub-ensemble	Official quantile average ensemble	California
2020-07-13	epiforecasts-ensemble1	time series / growth rate model	Florida
2020-07-20	UMass-MechBayes	Bayesian SEIR model	Georgia
2020-07-27	YYG-ParamSearch	Machine Learning / SEIR model	Illinois
2020-08-03	CU-select	SEIR / human selection model	Maryland
	UT-Mobility	Regression model	Massachusetts
	LANL-GrowthRate	Growth rate model	New Jersey
			New York
	mean-ensemble	Quantile average ensemble	Ohio
	qra-ensemble	QRA ensemble	Pennsylvania
	crps-ensemble	CRPS ensemble	Texas
			US

Chapter 5

Results - evaluation and aggregation of Covid-19 death forecasts

This chapter applies the tools presented in Chapters 2 and 3 to the data introduced in Chapter 4. We analyse the performance of the Forecast Hub models as well as the ensembles in detail to find out which models perform well and why. The structure of this chapter largely follows the general structure of the evaluation process proposed in Chapter 2, at times digressing a bit further to discuss additional aspects of interest. The starting point of the analysis forms the visualisation of the forecasts in Section 5.1. In Section 5.2 we then assess overall model performance to determine which models perform well and which do not. To that end we examine summarised scores, as well as take a look at scores in terms of a mixed effects regression. We subsequently analyse how the metrics relate to each other, what contributes to the performance measures and what drives differences in performance. We first examine correlations between the individual metrics in Section 5.3. While this is not strictly necessary for the model evaluation, it provides us with a better understanding of the metrics themselves. Afterwards, in Section 5.4, we explore the main contributors to the various scores. This analysis focuses mainly on the Weighted Interval Score (WIS) and its components, but we also look at how different ranges of the predictive distributions contribute to our measures of sharpness, calibration and overall performance. In Section 5.5 we then look at the main external factors and especially characteristics of the locations that drive performance differences. After having analysed the properties of the metrics and external factors that drive divergences in scores, we examine the models themselves more closely. In Section 5.6 we look at various aspects of calibration and sharpness in detail that help us to explain performance differences as well as hint to ways in which the models could be improved. The evaluation is followed by a discussion of the ensemble models in Section 5.7 where we discuss some aspects specific to the ensembles in more detail. These include a look at ensemble weights over time as well as analysis of different ensemble alternatives. Finally, we present a small sensitivity analysis in Section 5.8 that serves to check the plausibility of the inferences made throughout the chapter.

5.1 Forecast visualisation

A natural first starting point for the evaluation process is to visualise the forecasts and the observations to get a sense of the data. Figure 5.1 shows one and four-week-ahead forecasts for the United States as a whole[1]. From a brief, look we can see that most models generally predicted death numbers adequately in the short term. For four-week-ahead predictions, performance seems to have deteriorated significantly. We can already identify differences between the candidates. The

[1]Plots for other locations can be seen in the Appendix, where visualisations for one-week-ahead predictions of all forecasts are provided.

mean-ensemble, crps-ensemble and UMass-MechBayes model for example made consistently good forecasts at one and four week ahead predictions. The UT-Mobility model looks good for one week ahead (except for the last time point), but performed poorly with regards to longer-term forecasts. Predictions from the LANL-GrowthRate model seem rather off regardless of the horizon.

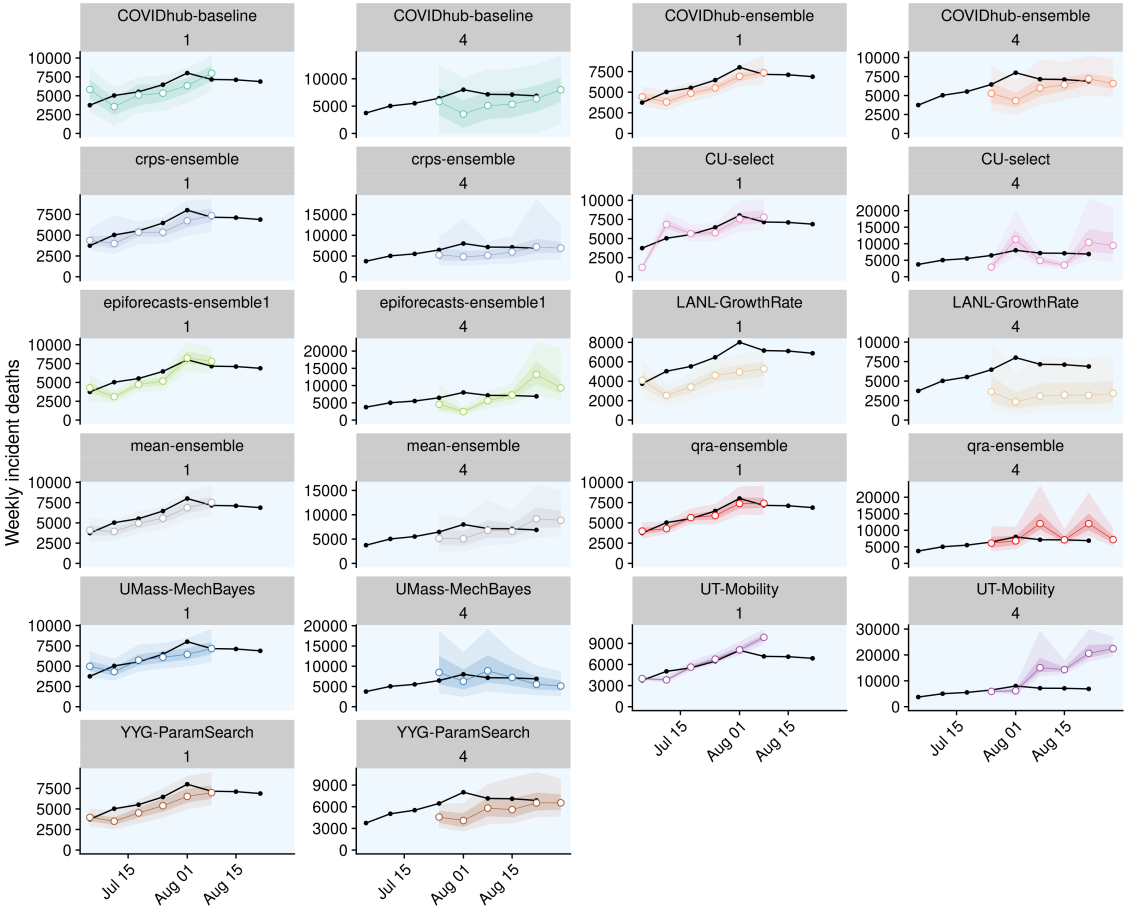

Figure 5.1: One week ahead forecasts for the US from all models. Observations are shown in black, median predictions ar marked by white points, ribbons show the 50 percent and 90 percent prediction intervals.

5.2 Summarised scores and overall performance

We start our formal model evaluation by assessing overall model performance. To that end, we first look at aggregated scores from the different metrics that help us to summarise all the complexity and nuances in a few numbers. Afterwards, we present the results of mixed-effects regression used to determine differences between models.

Figure 5.2 shows the summarised scores for all eleven models from the metrics presented in Chapter 2. Models are ordered according to the Weighted Interval Score (WIS). Note that the model ranking was slightly different for log transformed values of the WIS. This suggests that the average Weighted Interval Score was substantially influenced by extreme values. This overview gives us a very concise summary of overall model performance. We can see a rather clear divide between two groups in terms of performance: the four ensembles, UMass-MechBayes and YYG-ParamSeach performed similarly well, whereas the second group was less favourably ranked. This largely confirms the first visual impression obtained from Figure 5.1. We also see already that worse performing models tended to be more biased and to be either less calibrated or less sharp.

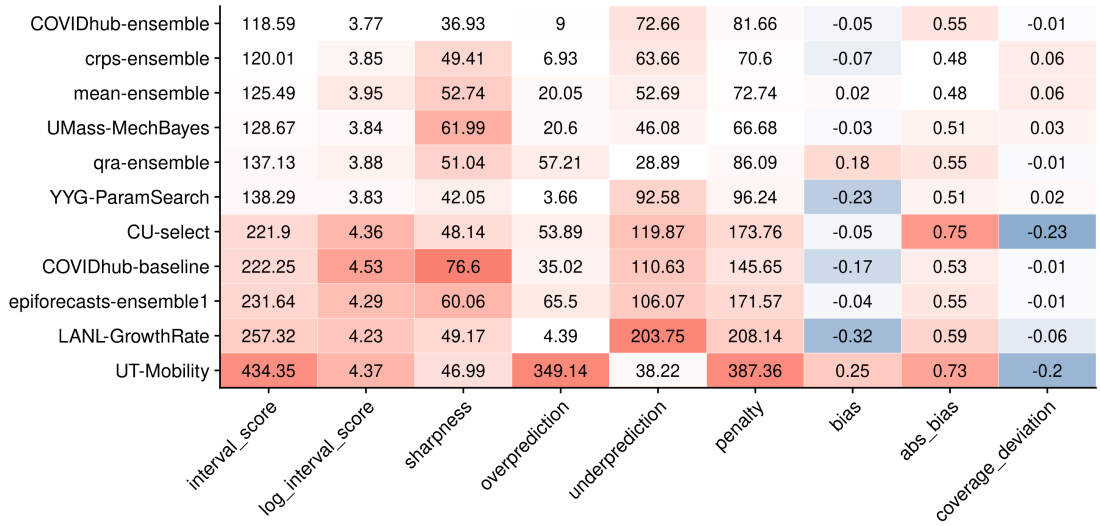

Figure 5.2: Colour coded summary of scores. Neutral / optimal values are shown in white, too low values in blue and too high values in red. Overprediction and underprediction refer to the over- and underprediction penalty parts of the WIS. Summed together, they form the column 'penalty' which again together with the 'sharpness' column sums up to the WIS. The absolute bias was included as well, as the original bias included both positive and negative values and the average can therefore be misleading.

To examine whether models were actually really significantly different in their performance, we employed a mixed-effects model with fixed effects for models and horizons, and random effects for states and forecast dates. The model formula looks as follows:

`lmer(log_scores ~ model + horizon + (1|state) + (1|forecast_date).`

Log Weighted Interval scores were used instead of the WIS to mitigate issues with heavy tails of the original distribution. As a baseline we took the COVIDhub-ensemble model (the top performer). This helped us to discern whether models at the top could actually be distinguished and show significant performance differences.

Table 5.1 shows the results from that regression. We can see that the regression confirms general tendencies observed before. The overall ranking of model effects corresponds to the model ranking by log Interval Score presented in 5.2. The regression output suggests again a clear split between two groups of models in terms of performance. Models in the top group were roughly comparable and not significantly different from the COVIDhub-ensemble model, except for the mean ensemble and maybe the qra-ensemble which fared a bit worse. As expected, the horizon had a highly significant effect on the Weighted Interval Score.

This regression framework is not only helpful in terms of model selection, but also provides a starting point to determine the key factors that drive differences in overall model scores. This will be discussed later, when we examine the random effects from the regression output more closely. In applications beyond this thesis, the regression framework is also very helpful in case of an incomplete set of predictions. If a certain research group for example misses a submission or does not submit forecasts in all states, then we can better mitigate this in a regression framework than by merely averaging over all available data. The specific regression we used can of course be adapted. We could for example include an interaction between model and horizon to control for the fact that some models cope better or worse with increasing uncertainty. One could also model horizon as a factor instead of a metric variable. If we were, for example, especially interested in performance over a four-week-ahead horizon, we could then estimate a separate effect for every model at horizon four by looking at the combination of the model effect and the interaction.

Table 5.1: Mixed model regression of the log Weighted Interval Score on model, horizon (both fixed), state, and forecast date (both random)

	Estimate	Std. Error	df	t value	Pr(>|t|)
(Intercept)	3.175	0.332	12.889	9.576	0.000
modelCOVIDhub-baseline	0.758	0.059	3259.890	12.830	0.000
modelUT-Mobility	0.597	0.059	3259.890	10.105	0.000
modelCU-select	0.584	0.059	3259.890	9.891	0.000
modelepiforecasts-ensemble1	0.518	0.059	3259.890	8.764	0.000
modelLANL-GrowthRate	0.462	0.059	3259.890	7.818	0.000
horizon	0.242	0.012	3262.136	20.865	0.000
modelmean-ensemble	0.179	0.059	3259.890	3.027	0.002
modelqra-ensemble	0.110	0.059	3259.890	1.856	0.064
modelcrps-ensemble	0.079	0.059	3259.890	1.340	0.180
modelUMass-MechBayes	0.065	0.059	3259.890	1.093	0.275
modelYYG-ParamSearch	0.056	0.059	3259.890	0.955	0.340

5.3 Examining the relationship between individual metrics

Before we analyse the performance differences that became evident in the previous section, we take a closer look at how the various evaluation metrics relate. Figure 5.3 shows the correlation between all metrics. We can see that, as expected, the penalty and sharpness terms correlated very strongly with the Weighted Interval Score. Penalty and WIS have the strongest correlation, and the overall penalty was again most strongly correlated with overprediction. Interestingly, penalty and sharpness were also positvely correlated. We should expect an increase in sharpness to lead to a decrease in penalties (as penalties only apply whenever an observation is outside the prediction interval). The fact that we see instead a positive correlation highlights that this analysis does not control for the performance level. Worse performing models evidently were less sharpn (i.e. had higher sharpness values) and also incurred more penalties at the same time. Bias and coverage deviation, as measured by the metrics described in Chapter 2, had a weaker correlation with the Interval Score. This is an indication that coverage deviation and bias measure aspects that the WIS does not capture directly. Even though we do rely heavily on the WIS to measure performance, it is important to remind ourselves that the WIS does not in fact equal predictive performance, but is instead only one of many possible ways to measure it. As long as we follow the forecasting paradigm of maximising sharpness subject to calibration it does make sense to also take other measures into account. It is also interesting to see that absolute bias and coverage deviation were more strongly correlated with the log Interval Score than the original WIS. This makes sense if we think of the log WIS as less influenced by outliers, since coverage deviation and our bias metric also tend to be rather robust. Even if none of the prediction intervals covers the true value, coverage deviation and bias scores are bounded. The WIS, on the other hand, can take infinitely large values depending on how much the predicitve intervals have missed.

Figure 5.4 goes into even more detail and shows a full correlation plot with all univariate and bivariate distributions. We can clearly see on the diagonal that the distributions of the WIS, as well as its components, had heavy tails. We can also make another interesting observations: A positive coverage deviation (i.e. covering too much by stating too wide prediction intervals) was still associated with a lower WIS in the models analysed, even though a positive coverage deviation is in principle also a form of miscalibration. This suggests that better models also tended to exhibit positive coverage deviation. It also implies that these could have presumably performed even better had they had slightly narrower prediction intervals.

To examine the main determinants of the Weighted Interval Score, we ran a regression of the log Weighted Interval Score on absolute bias, coverage deviation and sharpness and the penalty term:

```
lm(log_scores ~ abs(bias) + coverage_deviation + sharpness + penalty).
```

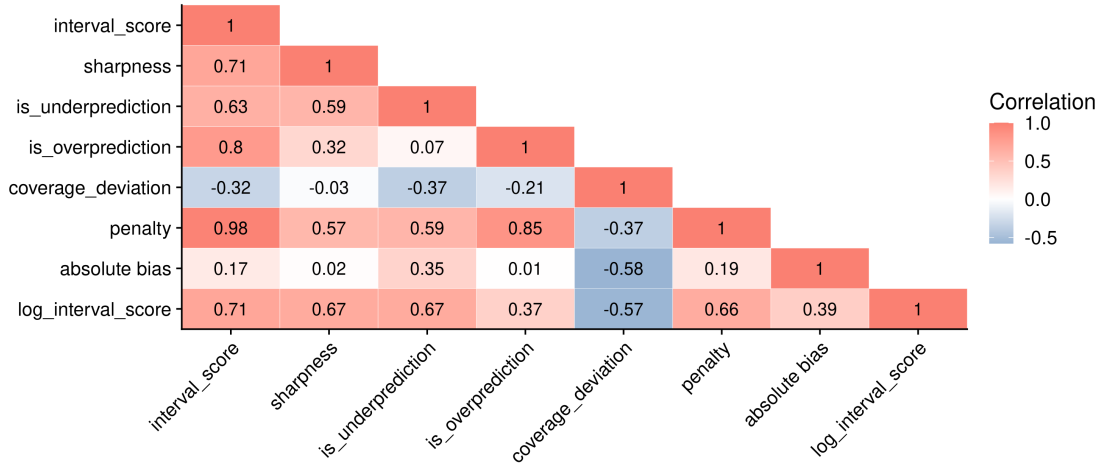

Figure 5.3: Correlation between the different metrics

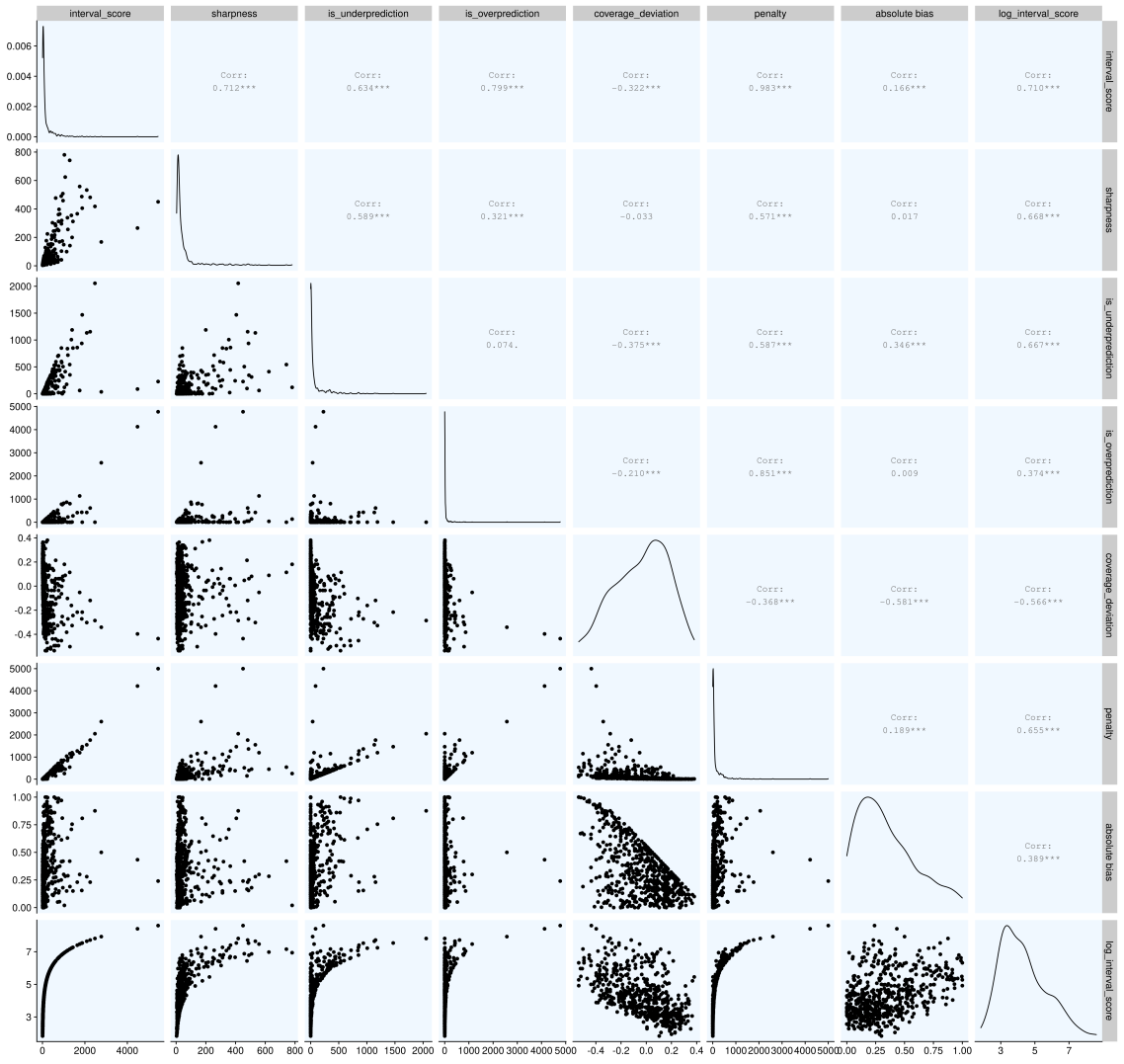

Figure 5.4: Correlation plot that shows bivariate scatter plots for all evaluation metrics.

Table 5.2: Regression of the log Weighted Interval Score on the (standardised) absolute bias, coverage deviation and penalty and sharpness.

term	estimate	std.error	statistic	p.value
(Intercept)	4.2897169	0.0265816	161.378966	0.0000000
abs_bias_std	0.1277333	0.0327084	3.905210	0.0001055
coverage_deviation_std	-0.5612346	0.0352656	-15.914487	0.0000000
sharpness_std	0.7305022	0.0333011	21.936272	0.0000000
penalty_std	0.2037627	0.0358050	5.690896	0.0000000

The result can be seen in Table 5.2. All regressors have been standardised, estimates should therefore be interpreted in terms of standard deviations away from the average values. We can see that all metrics had a strong and significant influence on the log WIS. This supports the idea to take multiple different metrics into account. The independent effects of 'sharpness' and 'coverage deviation' seem to have been especially pronounced. This makes sense in the context of the forecasting paradigm, but may as well just reflect the specific data set analysed here.

5.4 Identifying main contributors to the WIS

The last section has given us a better sense of how different metrics relate and has offered a first look at what aspects of the predictive distribution most strongly determine the Interval Scores. This section continues the analysis into the main contributors of the various scores. We first split the WIS into its components: sharpness and the penalties for over- and underprediction. Afterwards, we look into how different ranges of the predictive distributions contribute to the different metrics and the WIS in particular. This becomes even more evident when we plot relative contributions, as is shown in Figure A.11 in the Appendix. Figure 5.5 shows the WIS for all eleven models over different horizons, separated into its components sharpness, overprediction and underprediction. We immediately see that forecasts further ahead into the future received the largest WIS and therefore also had the largest impact on average WIS. We can also see that, especially for worse performing models, over- and underprediction penalties constituted a bigger share of the overall WIS than the sharpness component. For better performing models, the share of the sharpness component relative to the overall WIS increased. This makes sense in the context of the forecasting paradigm, where sharpness should become increasingly important once calibration is satisfied.

We can also observe that most models incurred the majority of their penalties from underprediction. This is even true for models that according to Figure 5.2 exhibited no bias or even a slight upward bias, like the mean-ensemble. This underlines again that the bias metric behaves slightly different from the under- and overprediction component of the WIS. The former measures the innermost quantile that covered the true value. The latter measures the actual difference between predictions and observations and is therefore more sensitive to extreme values. It is therefore possible that inaccuracies in one direction get more severely punished than inaccuracies in the other, even if both types of inaccuracies appear equally often.

To understand the the observed scores better we also looked at the contributions from different interval ranges to the various metrics. Figure 5.6 shows these contributions to WIS, sharpness, penalty, and coverage deviation. We can observe that inner prediction intervals had the strongest influence on the WIS, while outer intervals were less important. This can be explained by the weighting scheme of the Interval Score, as well as the fact that outer prediction intervals naturally incur less penalties from over- or underprediction. In terms of sharpness it seems that intervals around the 50% interval range have the strongest impact. Outer prediction intervals get weighted down, while inner intervals are too narrow to have a strong influence. If we look at coverage deviation, however, we see that problems in forecasts tend to be even more pronounced in the tails of the distribution. These tails, however, matter less for the WIS as they receive only a small weight.

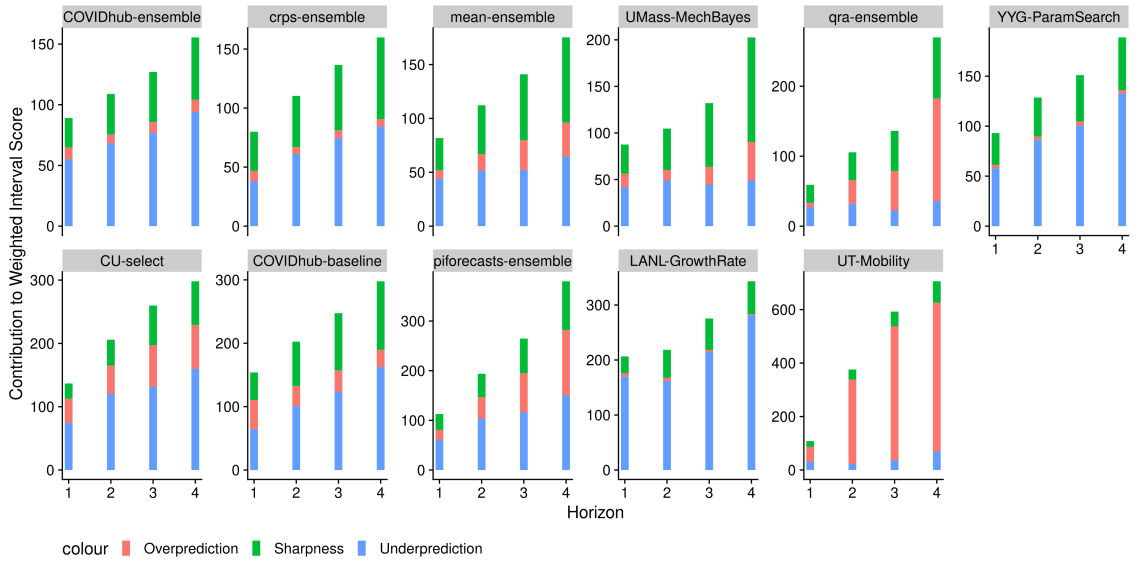

Figure 5.5: Contributions to the Weighted Interval Score from its underprediction (blue), overprediction (red) and sharpness (green) components.

5.5 Identifying external drivers of differences WIS

So far we have analysed the behaviours of the evaluation metrics and looked at the most important contributors to the scores. In this section and the following one we want to examine what drives differences in WIS. This section identifies determining factors other than the models themselves, while the next section examines individual model characteristics more closely.

As a starting point, we look again at the mixed-effects regression shown in Table 5.1. From the fixed effects estimates, we already know that the forecast horizon made a large difference in average scores. We can now go one step further and also analyse the estimated random effects from the same regression model. These are shown in Figure 5.7. We can see that forecast dates did not have a very strong effect, but that states did. This difference between the states could have been completely driven by differences in death numbers or it could be due to variation in how difficult it was to forecast certain states. The remainder of this section examines this more closely.

Figure 5.8 shows deaths on the x-axis versus Weighted Interval Scores per model on the y-axis. The plot shows substantial variation in scores, but overall a very strong relationship between average WIS and the number of deaths is visible[2]. It therefore seems that most of the variation in observed Interval Scores between states can be explained by the overall level of death numbers. We nevertheless continued and tried to identify states that were particularly hard to forecast.

One natural way to estimate state difficulty is to measure it as the difference between the Interval Scores actually observed in a state and the ones that would be expected from this relationship. To that end we fit the following regression to the log Interval Scores:

`lm(log(wis) ~ log(deaths)).`

Difficulty was then estimated as

$$\text{difficulty}_{\text{location}} = \log(\text{WIS}) - (\beta_0 + \beta_1 \cdot \log(\text{deaths}_{\text{location}}).$$

The results are shown in Figure 5.9. We can see that indeed models consistently struggled with some states, whereas others seem to have been easier to predict than expected by the number of

[2]One could argue whether or not to include the US as a whole in this plot. On the one hand the overall US death numbers are closely related to the sum of death numbers from the twelve states analysed. On the other hand numbers were not identical as the majority of states was missing from the analysis. We could therefore also think of the US as just another very large state. We tried both options and found that the difference was very small in terms of the regression fit and the overall relationship observed.

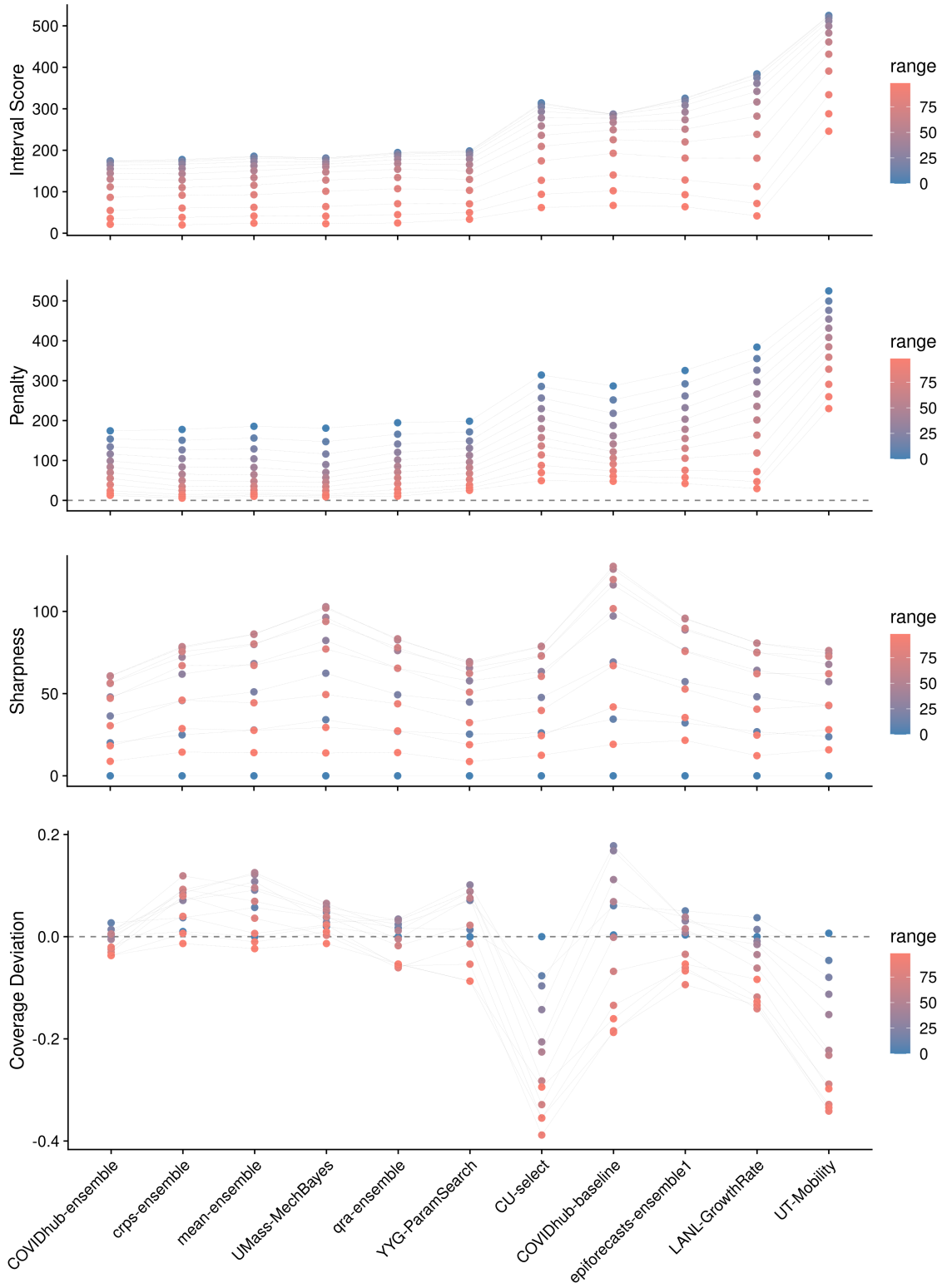

Figure 5.6: Weighted Interval Score (top), added penalties from over- and underprediction (second from top), sharpness component of WIS (third from top) and coverage deviatin (bottom) for all eleven models and different interval ranges. We can see that overall outer intervals contributed less to the total WIS.

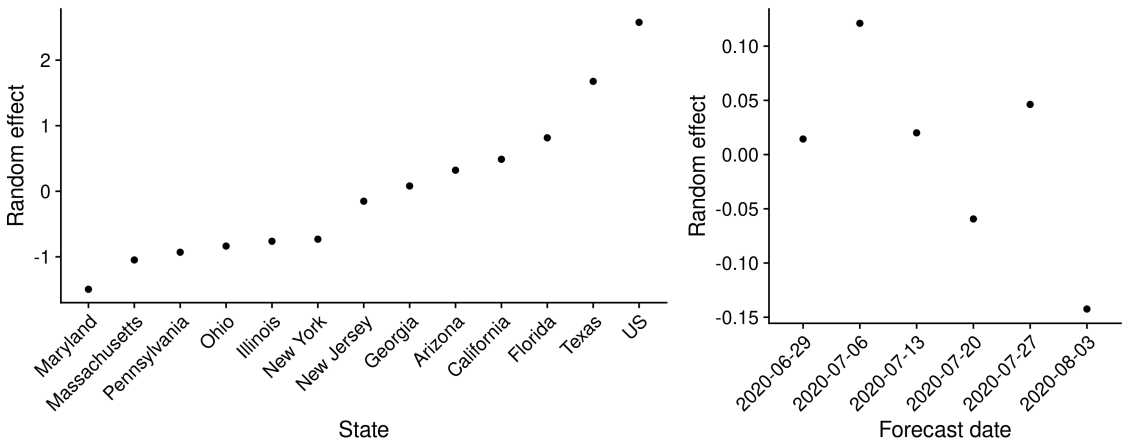

Figure 5.7: Random effect estimates for the different locations (left) and forecast dates (right)

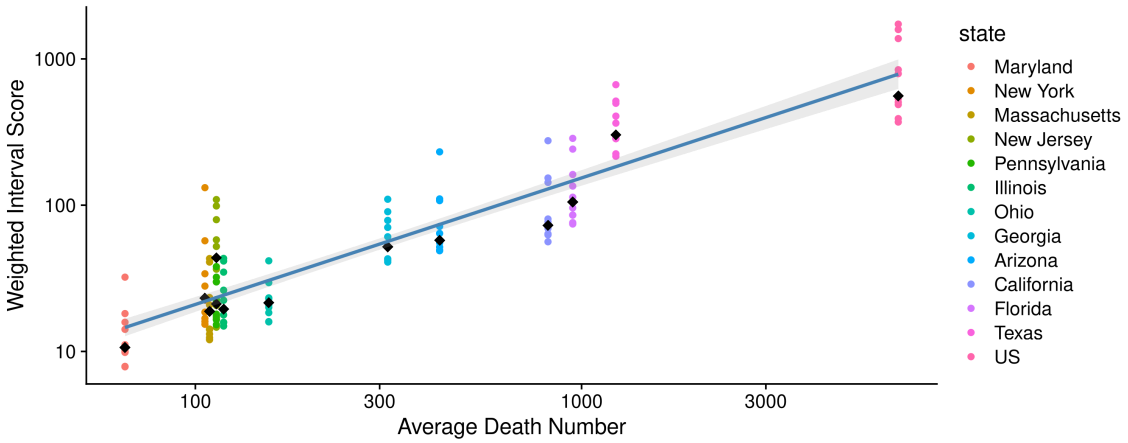

Figure 5.8: Plot of the Weighted Interval Score achieved by the models per location versus the average death number in that location. The black diamonds mark the median WIS for that location.

their deaths. Comparing this to the visualisation of the observations from all locations in Figure 4.1, however, we do not immediately see a clear picture emerge from this analysis. At least visually, for example, California looks harder to forecast than New Jersey[3], New York or Pennsylvania. States with the hardest difficulty estimates still include states with very diverse patterns and trends. Some experienced falling death numbers (New Jersey, Pennsylvania) or very little change at all (New York[4]), while others saw their death numbers rapidly increase (Texas).

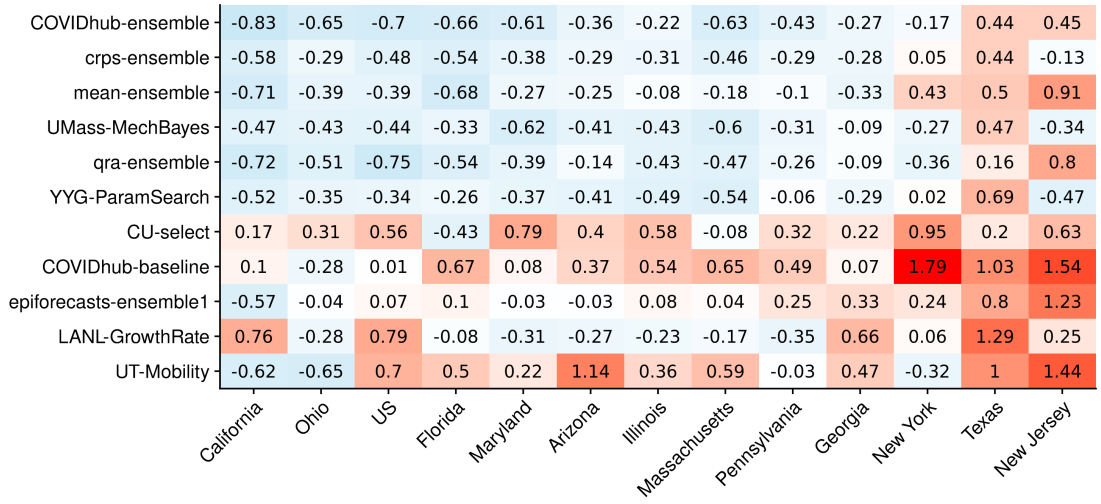

Figure 5.9: Estimated difficulty per state. Difficulty was estimated as the difference between actual log WIS and log WIS expected based on the relationship between log WIS and the log of the average death number.

We also looked at coverage deviation as well as bias by state to get a feeling for how much models were off in terms of calibration across states. Figure 5.10 shows coverage deviation and Figure 5.11 shows 'bias' for all models and locations. These analyses corresponded a bit better to our intuition of how hard it is to forecast states. Most of the states on the left in Figure 5.10 were states with slightly falling death numbers and relatively smooth curves. If we compare this to Figure 5.11, we see that a substantial part of what made states hard to predict was associated by a tendency among all models to over- or underpredict. A similar way to look at locations in terms of bias and calibration is to visualise the relative share of sharpness versus misprediction penalties in different locations. This is shown in Figure A.12 in the Appendix.

To understand what drives the dynamic it makes sense to look again at the visualisation of observed deaths in Figure 4.1. We see that models exhibited the strongest downward bias in the locations that saw their numbers increase (Texas, Georgia, California, US, Florida, Ohio, Arizona) and the strongest upward bias in the states with falling death numbers (New Jersey, Maryland, New York, Pennsylvania). While this is unfeasible for all states, Figure 5.12 shows the one-week-ahead predictions and observed values in Texas as a case study. We can observe that Texas experienced a rapidly changing trend that all models were unable to keep up with. One particular feature that might have made Texas even harder to forecast than for example Georgia, California or Ohio, is that death numbers were very stable in the weeks following the stark increase. On the other hand, a similar observation could be made for death numbers in Florida, but bias was less pronounced there than in Texas. One could also argue that we should have expected bias to be even higher in states like Ohio, Georgia or California where deaths were falling before the change in trend. While we can make plausible hypotheses, we ultimately have to acknowledge that is hard to draw definitive conclusions with limited knowledge of how predictions were generated.

[3]New Jersey may be a special case due to the data issues mentioned in chapter 4

[4]New York may be potentially considered an outlier as well, as overall performance in New York was strongly influenced by the high uncertainty and correspondingly high WIS of the COVIDhub-baseline model.

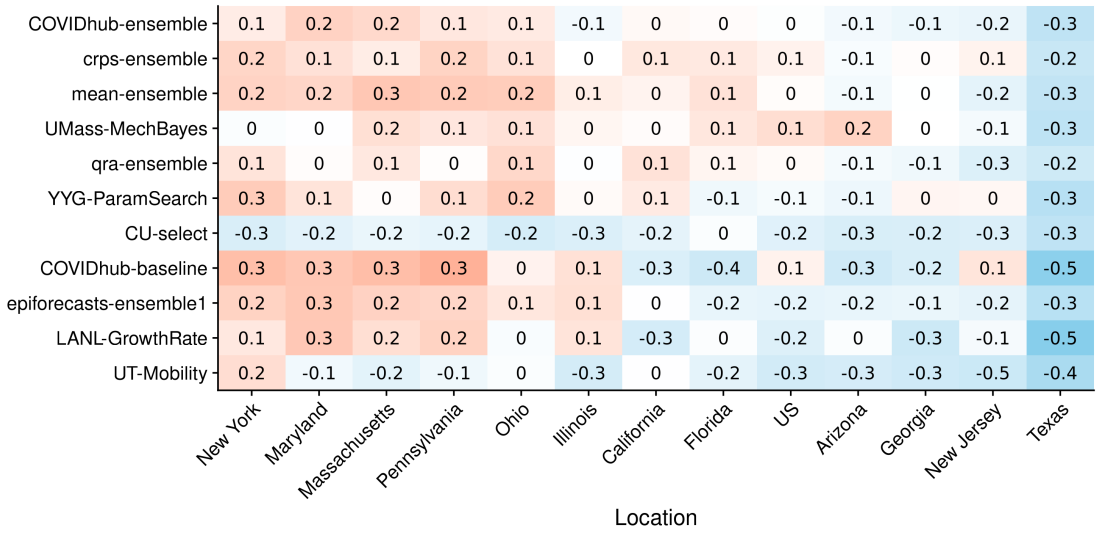

Figure 5.10: Average coverage deviation for all models and all thirteen locations.

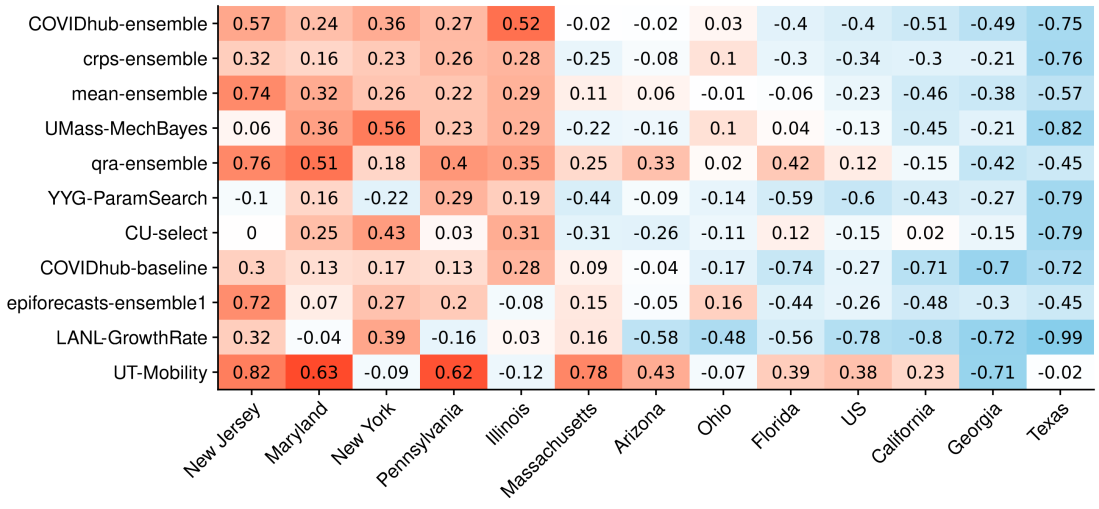

Figure 5.11: Average bias for all models and all thirteen locations.

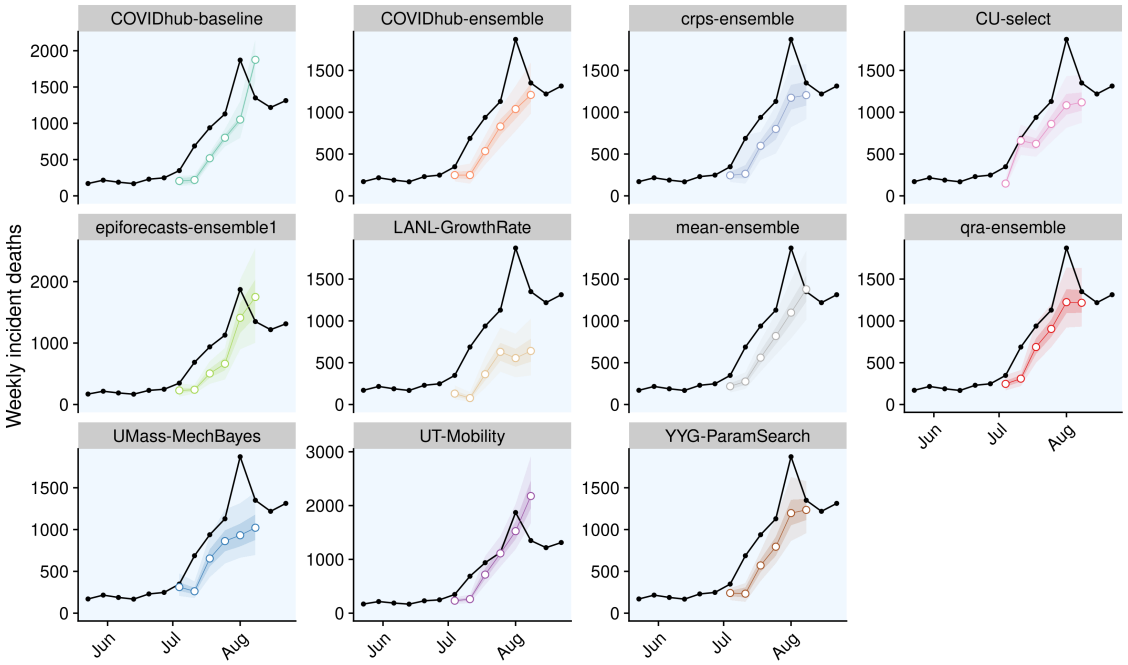

Figure 5.12: One week ahead predictions and observed values in Texas

5.6 Understanding model characteristics that drive differences in WIS

In the last section we have looked into the main external drivers of variation in WIS. This section analyses characteristics inherent to the models themselves that explain divergences in performance. As a starting point we examine whether models performed substantially different across locations, or whether models tended to perform conistently well or badly. We then look in detail at patterns in Bias, Coverage, PIT histograms, Sharpness that serve to explain performance differences.

Figure 5.13 shows the WIS of all eleven models in all thirteen locations. The shading indicates how much larger a score is than the lowest score achieved in that state. We see that models tended to perform rather consistently across locations. This may be an indicator that models tended to fail in very similar ways. At least, we can conclude that any relative strengths that models may have had were dwarfed by divergences in overall performance. The qra-ensemble, for example, with its tendency to overpredict did well in states like Texas or California that most models underpredicted. On the other hand, it did not perform exceptionally well in the US and in Florida, two locations which were also prone to underprediction. The YYG-ParamSearch model, which is downwards bias on average, does very well in some states with falling death numbers (New York, New Jersey, Illinois), but mediocre in others (Massachusetts, maybe Maryland). It also is the best performing model in Georgia, one of the states that showed consistent underprediction by all models. We can hypothesise that models like the qra-ensemble or the YYG-ParamSearch might have had a relative advantage in locations with certain characteristics, but it is hard to identify general patterns, as overall performance clearly dominates the picture. To investigate further, the following subsections look more closely into differences in calibration and sharpness between the mdoels.

5.6.1 Bias

Just as we did in Chapter 2, we started our analysis of model calibration by examining systematic over- or underprediciton. Where we previously looked at bias with the goal to explain variation across states, this section investigates bias in order to explore differences in the models.

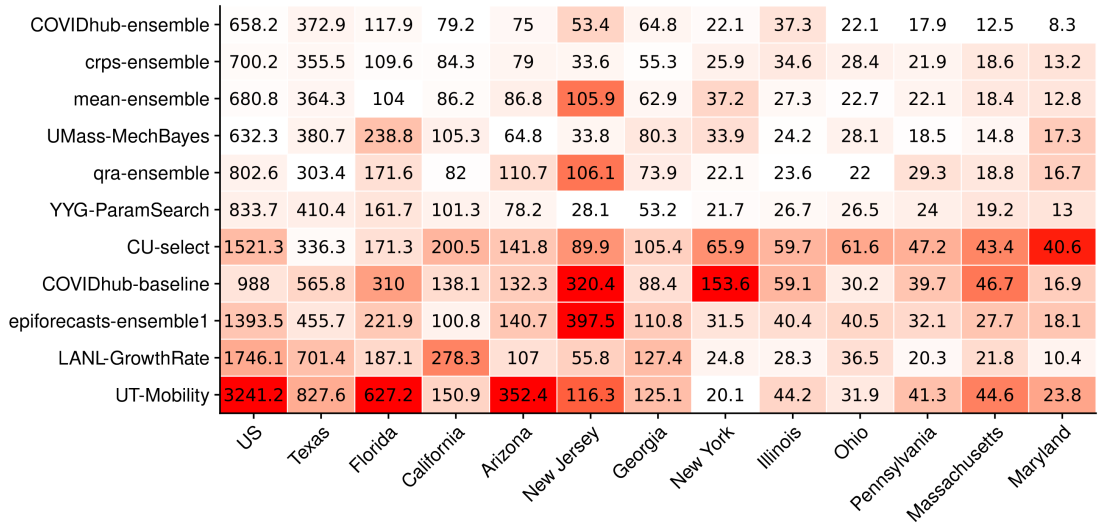

Figure 5.13: Average Weighted Interval Score for all eleven models and all thirteen locations. The colouring indicates how much larger a score is than the lowest score achieved in that state.

Figure 5.14 shows our bias metric as well as penalties from over- and underprediction for all models. Looking at the bias metric in the top panel of the figure, we see that models with lower overall WIS were also rather less biased. We also see a slight increase in bias for increasing foreacast horizons. If we also look at bias in terms of the over- and underprediction penalties incurred by the individual models, we clearly see that better models tended to avoid large penalties. In comparison with summarised scores and the contributions to overall WIS showin with Figures 5.2 and 5.5 we can clearly see the connection and trade-off between sharpness and misprediction penalties among the top performers. The UMass-MechBayes model, for example, largely avoided these penalties, but was also not very sharp. This trade-off between sharpness and prediction penalties is a general challenge forecasters face. The COVIDhub-ensemble was much sharper, but also suffered more misprediction penalties. We also see a bit of a tension between both ways to measure bias. From the first plot, we would for example conclude that the COVIDhub-ensemble did very well in terms of 'bias'. From the second, we might draw that the model suffered from underprediction and that we could try and improve it by skewing it a bit upwards.

For the purpose of individual model improvement it seems most useful to compare the evolution of bias over time with the actual predictions and observations. With the help of this comparison we can obtain insights regarding the particular situations that cause models to be biased or not. While this is of course unfeasible to do for all eleven models, Figure 5.15 shows one-week-ahead predictions and bias for the three ensemble models in the six locations with the highest average WIS. We see that all models made rather similar predictions. Even though differences were small, the qra-ensemble model looks slightly ahead in many (Arizona, California, maybe Texas) of the locations displayed with an upwards trend, but not all. Since these locations saw severe underprediction across all models, this might point towards a small relative advantage that the qra-ensemble enjoys in such locations. All models seem to have had similar difficulties with picking up rapid changes in trends. This was especially pronounced in Texas, as was discussed previously, but is also true for the other locations. We can also again see that models tended to overpredict when deaths were increasing (Arizona, Texas) and tended to underpredict when cases were decreasing (New Jersey, last two weeks in Arizona).

5.6.2 Coverage

We next turn to examine coverage. Figure 5.16 shows the empirical interval and quantile coverage for all eleven models. The green shading indicates the areas of the plot that correspond to excessive

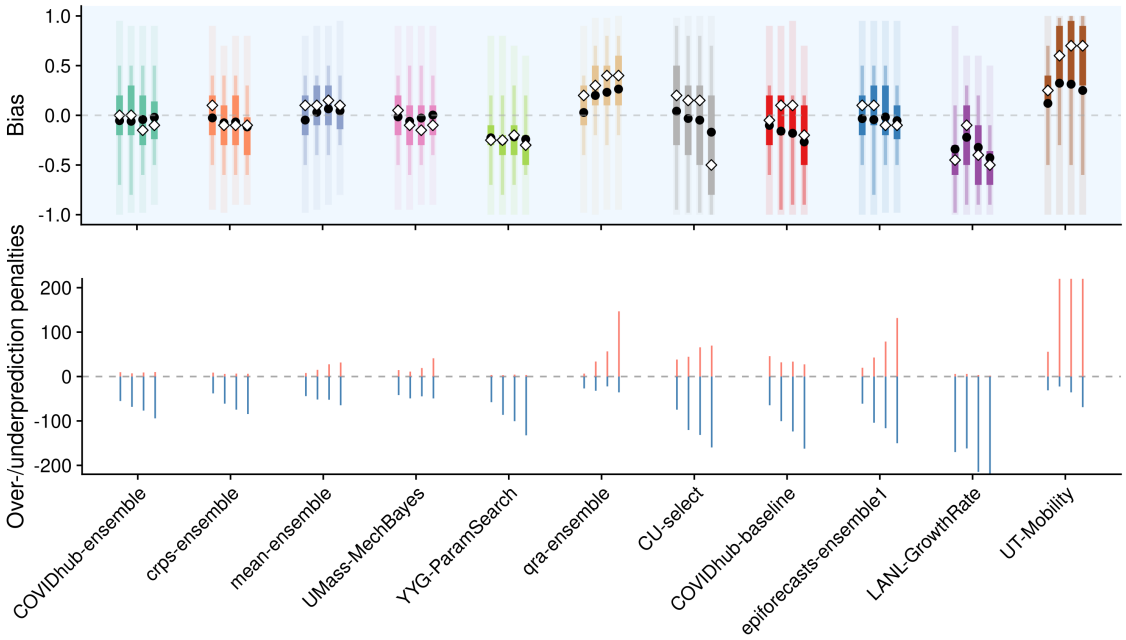

Figure 5.14: Top: Bias for all models and different horizons. The black dot denotes the median bias, the black square the mean bias and different colour shadings show the 20, 40, and 90 percent intervals of all observed quantile values. Models are again ordered according to their overall performance by WIS. Bottom: Over- and underprediction penalties incurred by the different models. Underprediction penalties were shown as negative to make the illustration more intuitive.

coverage, or correspondingly in the case of the quantile coverage to predictive quantiles which tend to be more extreme than they should. The interval coverage plots are especially good at giving us a quick impression of overall model calibration. In this regard, the COVIDhub-baseline model is an interesting example. While the aggregated coverage deviation score in Figure 5.2 looked very good, we can now immediately conclude that the COVIDhub-baseline model was not well calibrated. Interval coverage provide a very good first impression, but cannot tell us where a lack of coverage comes from exactly. Quantile coverage plots are a bit harder to interpret, but they convey more information and can show us whether an issue is located in the lower or upper tails of the predictive distributions. With regards to the crps-ensemble for example, we see from the interval coverage plot that it was consistently covering too much by its prediction intervals. The quantile coverage plot allows us to be more precise and say that the issue arose from the lower tails of the predictive distributions that tended to be too wide.

The quantile coverage plots also allow us to reexamine the bias component of calibration again. We can for example see that the UT-Mobility and qra-ensemble, which exhibited an upward bias (compare again the summarised scores in Figure 5.2), are moved to the left of the diagonal, while e.g. the YYG-ParamSearch or the LANL-GrowthRate model, which were downward biased, are moved to the right. The COVIDhub-baseline model is again an interesting example, as it had a downward bias tendency, even though its median predictions were on average spot on[5]. The line in the quantile coverage plot almost exactly crosses (0.5, 0.5), but Figure 5.14 shows that the model was nevertheless biased downwards on average and incurred more under- than overprediction penalties.

We can, unsurprisingly, observe a general tendency that better performing models also showed better results in terms of interval and quantile coverage. Especially the COVIDhub-ensemble and the UMass-MechBayes models stand out in Figure 5.16, while models in the lower performance bracket

[5]Note again that this finding was heavily influenced by the selection of states for this analysis. Had we analysed a different set of locations, we might have come to different conclusions

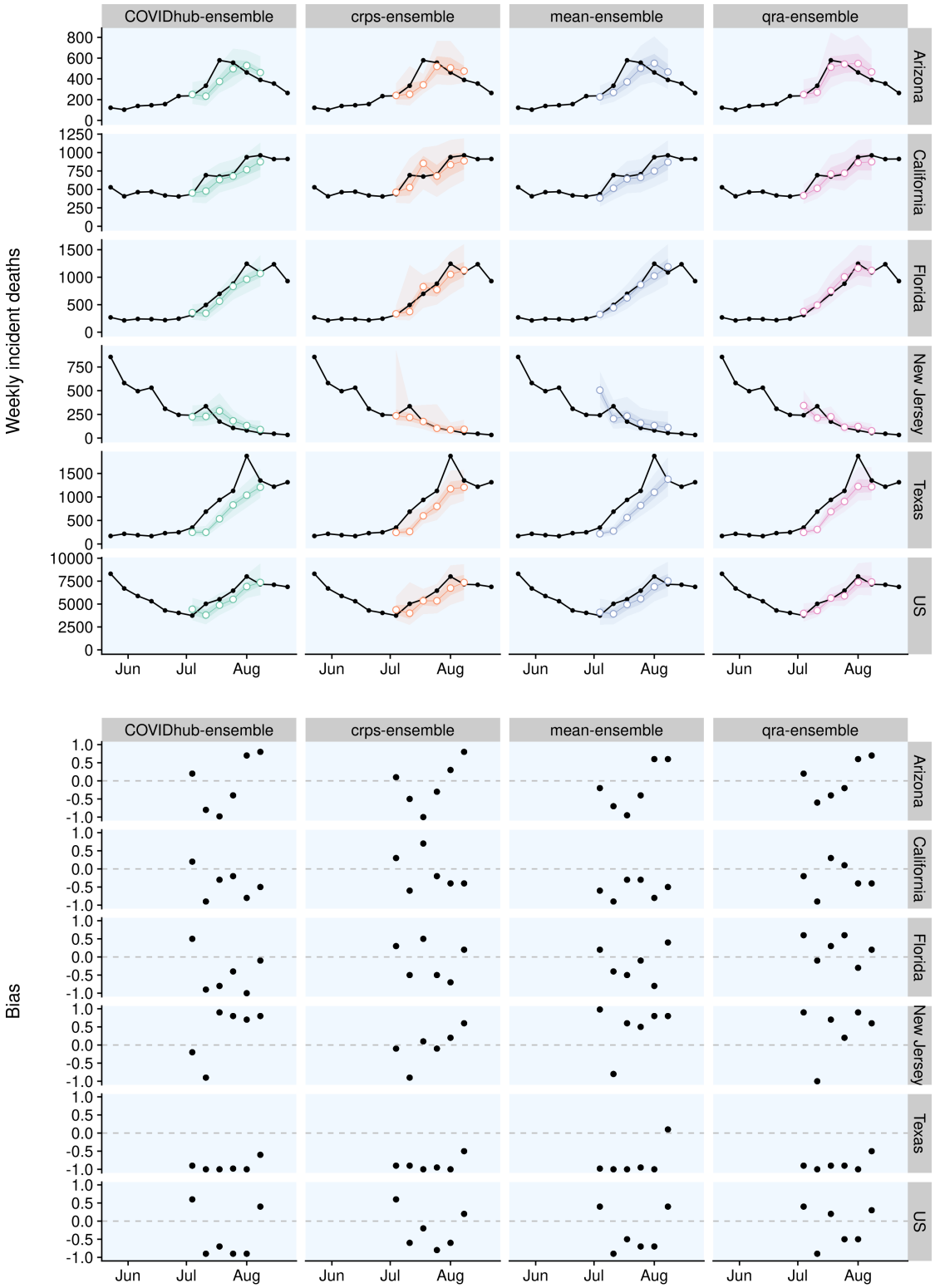

Figure 5.15: Observations and predictions (top) as well as bias (bottom) for the four ensemble models in the six states that exhibited the largest absolute bias.

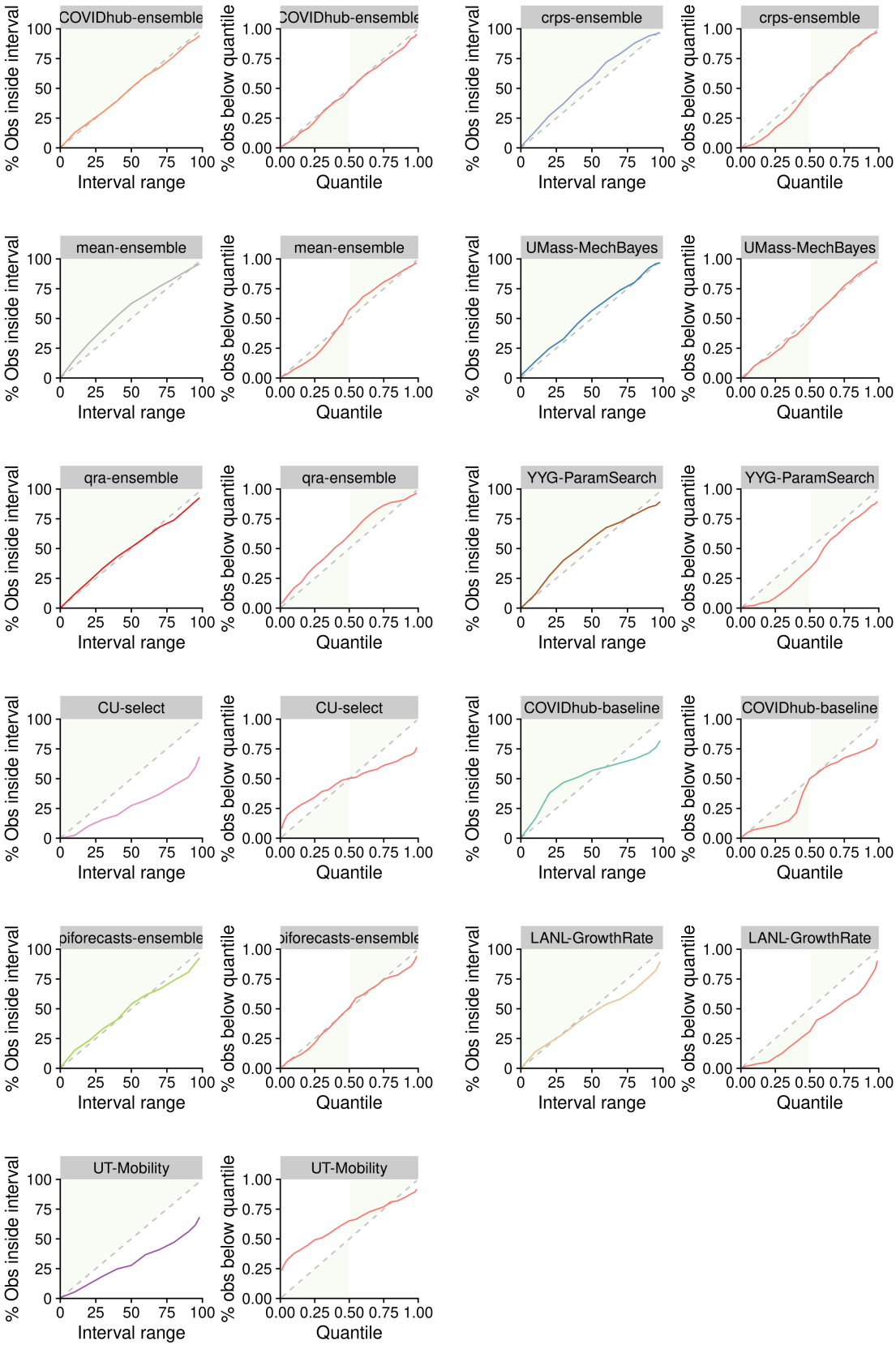

Figure 5.16: Interval and quantile coverage for all models across all locations and forecast dates

look much less favourably. The only real exception to that pattern is the epiforecasts-ensemble1. It does look much better in terms of coverage than other models that performed similarly well. To investigate further, one could examine calibration plots for the epiforecasts-ensemble1 model for every single location, but we omit this additional analysis here and instead show a plot of predictions versus observations in Figure 5.17. In the plot, we see that the model also exhibits a pattern of overprediction whenever death numbers were falling and underprediction whenever they were rising that results in the impression of a well calibrated model on an aggregate level. We also see that the model tended to sometimes be wrong in a very unpredictable way. Especially forecasts for July 27th seem affected, where predictions as well as the uncertainty made a sudden unpredictable jump in many locations. This again underlines the importance of breaking down the evaluation in smaller subcategories as well as looking on the aggregate level.

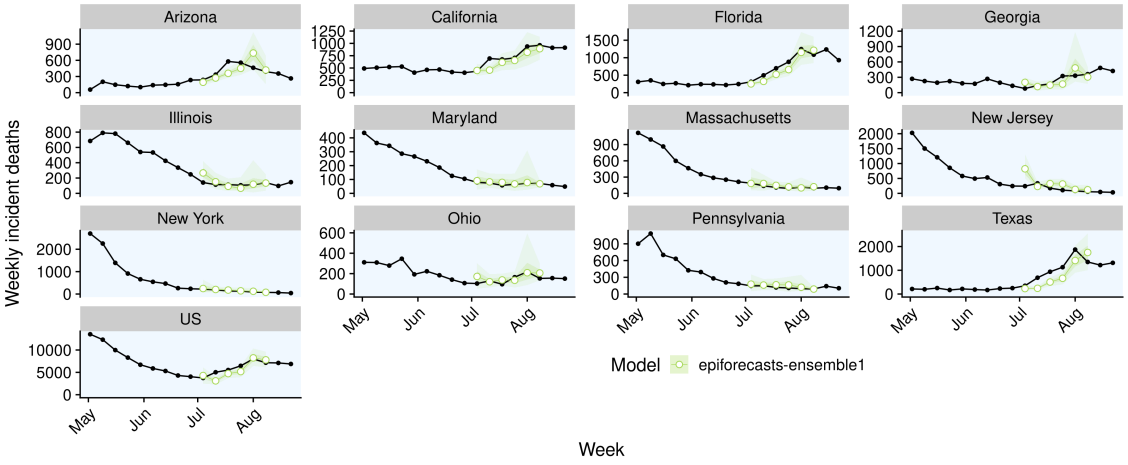

Figure 5.17: One-week ahead forecasts for the epiforecasts-ensemble1 model.

One major advantage this type of visualisation has over PIT histograms is that we can easily compare different models in a single plot. Figure 5.18 exemplifies this for the ensemble models. In terms of interval coverage (left in the plot), the COVIDhub-ensemble and the qra-ensemble did best, while the crps-ensemble and the mean-ensemble had a slightly too high interval coverage. These observations are in line with the summarised scores displayed in Figure 5.2. Looking on the right, we can clearly see the COVIDhub-ensemble also did best in terms of quantile coverage. Given its good interval coverage, it may be surprising to see that the qra-ensemble model was the most biased of all ensemble models. We can explain this discrepancy by taking sharpness into account: In Figures 5.2 and 5.5 we could see that the COVIDhub-ensemble was much sharper than the other three ensembles which were about equally sharp. For the mean-ensemble and the crps-ensemble the increased width of the prediction intervals translated into a positive coverage deviation, while for the qra-ensemble it only mitigated the effect the increased bias would have had on interval coverage. This slight discrepancy between quantile and interval coverage, highlights again that interval coverage only shows one kind of calibration. Good interval coverage is a necessary condition, but not sufficient to prove good calibration.

In some cases it can be interesting to also visualise calibration over states or the evolution of coverage over horizons. This can, for example, be an indicator of how far ahead into the future we can forecast. As these plots tend to be somewhat hard to interpret, we have omitted this analysis here, but included a plot of coverage over different horizons in the Appendix in Figure A.13.

5.6.3 PIT histograms

In addition to looking at coverage plots, we can also approach calibration through PIT histograms. Figure 5.19 shows the PIT histograms for all eleven models. These were created by first fitting a metalog distribution to predictive quantiles in order to obtain samples. The metalog distribution was chosen here to retain the information from the quantiles as accurately as possible. Instead,

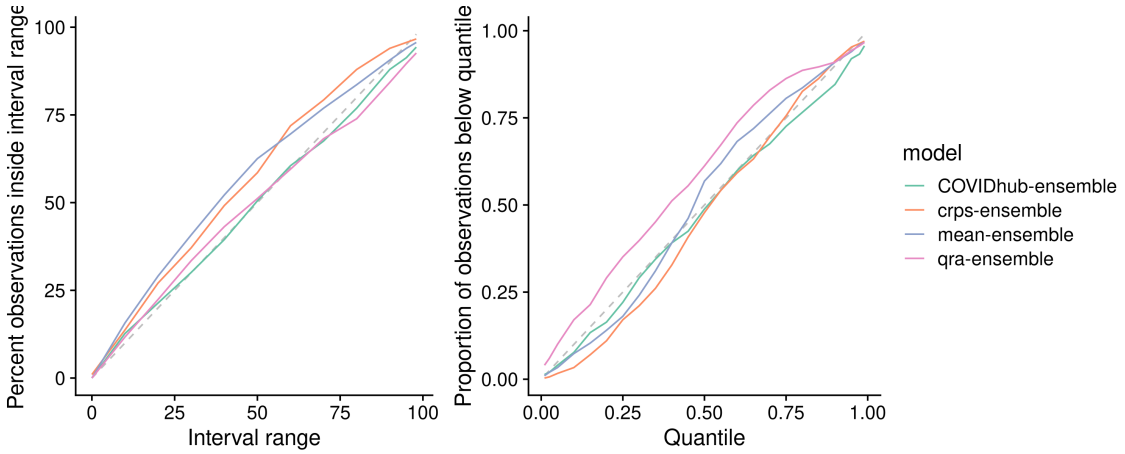

Figure 5.18: Aggregated interval coverage (left) and quantile coverage (right) for all four ensemble models.

one could have created PIT histograms with exactly 23 bars directly from the quantiles submitted to the Forecast Hub. This, unfortunately is not yet implemented in the `scoringutils` package. Fitting samples also allows us to make use of the Anderson-Darling test for uniformity. We can see that the AD test is rejected for all models. This suggests that all models were miscalibrated, but we cannot rule out effects from fitting a distribution to the samples. As discussed in Chapter 2, the AD test may also sometimes be overly conservative. For many of the models we indeed saw signs of severe miscalibration in the coverage plots in Figure 5.16. For others, like the COVIDhub-ensemble model, we may not want to reject calibration outright. Just on the basis of the AD test we can also not quantify miscalibration here. As all p-values are indistinguishably close to zero, the AD test is only of limited value here for the purpose of comparing models and differentiating betwen them.

PIT histograms provide a very good way to succinctly summarise different aspects of calibration in one plot. The information they provide is very similar to the one included in quantile coverage plots. For forecasts in a continuous or integer forecast, PIT histograms can convey more information than coverage plots, if we choose an appropriately large bin size. For forecasts in a quantile format, we only have enough resolution to inform as many bins as we have available quantiles anyway. In our case, preferring PIT histograms or coverage plots is therefore purely a matter of taste. PIT histograms do have the advantage to combine the information included in interval coverage as well as quantile coverage plots. For example, they make it easy to visually diagnose over- or underdispersion in the forecasts by simply identifying a U-shape or hump-shape. We can see this hump-shape in the crps-ensemble or the mean-ensemble, and can observe a U-shape in the PIT histogram for the UT-Mobility model. We can also see again how upward and downward bias, respectively, are visible in the histograms for the qra-ensemble and the YYG-ParamSearch model. But as with coverage plots, we only see an aggregate picture and should also examine PIT histograms for individual locations or forecast dates.

5.6.4 Sharpness

We already discussed sharpness briefly when we looked at the overall composition of the WIS. We concluded that sharpness plays an increasing role in differentiating top performing models, while it is only a small part of overall WIS for those models that struggle with calibration. We now look at sharpness again more closely, both in terms of model comparison as well as in terms of model improvement.

Figure 5.20 shows sharpness over different horizons for all eleven models. As expected and intended, prediction intervals tended to grow with increasing forecast horizons, and therefore accounted for increasing uncertainty. Only CU-select and LANL-GrowthRate slightly failed this basic sanity check as their median sharpness did not increase over all horizons. For the better performing

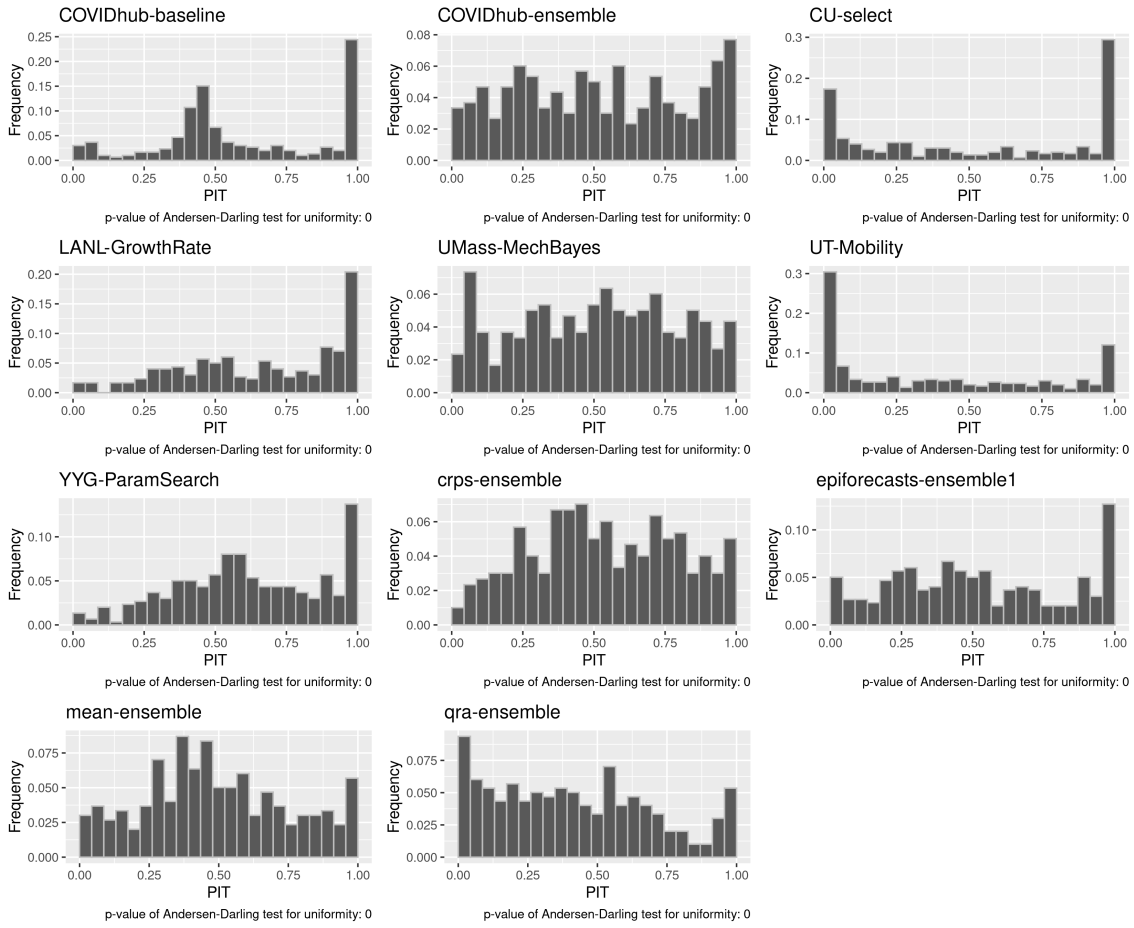

Figure 5.19: PIT histograms for all models. Samples were obtained by first fitting a metalog distribution to the set of quantiles. Note that the PIT plots shown here don't have the same scale on the y-axis, which make them easier to read on their own, but a bit harder to compare.

models we mostly observe a moderate increase[6], while worse performing models tend to see slightly sharper increases or more erratic behaviour. Given that we should maximise sharpness subject to calibration, it makes more sense to split the eleven models in two groups in order to compare sharpness between models that do equally well. One group comprises the four ensemble models as well as the UMAss-MechBayes and the YYG-ParamSearch, the other group includes the rest of the models. Within each group we indeed see a tendency that better performing models also were less sharp. We should, however, be careful not to interpret these findings causally. Given the high correlation between sharpness and penalty observed in the correlation map in Figure 5.3, this may just reflect that better performing models also may have been sharper for unrelated reasons.

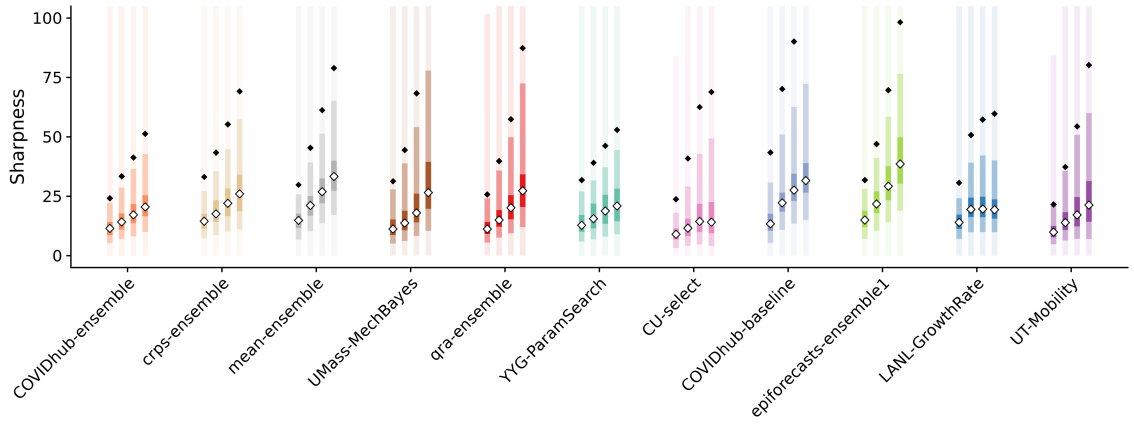

Figure 5.20: Sharpness for all models over different horizons. The black dot denotes the median sharpness, the black square the mean sharpness and different colour shadings show the 20, 40, and 90 percent intervals of all observed quantile values. Models are again ordered according to their overall performance by WIS. Values were capped at 110 to make models comparable.

In order to understand and improve the individual model, it is again most helpful to plot sharpness next to predictions. Figure 5.21 shows this for the crps-ensemble model for one-week-ahead predictions. We can see that sharpness of the crps-ensemble model does not really follow a clearly identifiable pattern. As expected, it was much larger in locations like the US that saw more deaths on average. What we cannot see is the ensemble model consistently adapting in reaction to past mistakes. Ideally, we would want a model to make wider predictions whenever predictions and observations have not matched previously (for example, when the trend changes) and narrower ones when past predictions and observations have matched. Unfortunately, none of the original models really seems to have exhibited that kind of behaviour (compare the visualisations for one-week-ahead forecasts of all models in the Appendix) and the ensemble was not able to mitigate this shortcoming. Instead we see rather random looking changes in sharpness, e.g. in Arizona or in the national forecasts for the US that do not seem to be informed by past errors.

5.7 Specific analysis of ensemble models

The past sections have provided a comprehensive evaluation of all eleven models, but have not devoted special attention to the analysis of the ensemble models. This section explores aspects specific to the ensembles. It focuses on the QRA and the CRPS ensemble, as these are the ones of greatest interest for the purpose of this thesis. The first part examines how ensemble member weights evolve over time for the two model aggregation approaches. The second part then looks at different variants of the QRA and CRPS ensemble varying the number of past observations to take into account or the horizon to optimise for. A regression framework is used again to determine differences between the different ensemble variants.

[6]For the UMass-MechBayes model we also observe substantial outliers in sharpness. Looking back at the breakdown of WIS into its components in Figure 5.5, we see that this has indeed hurt the UMass-MechBayes model in terms of overall WIS performance.

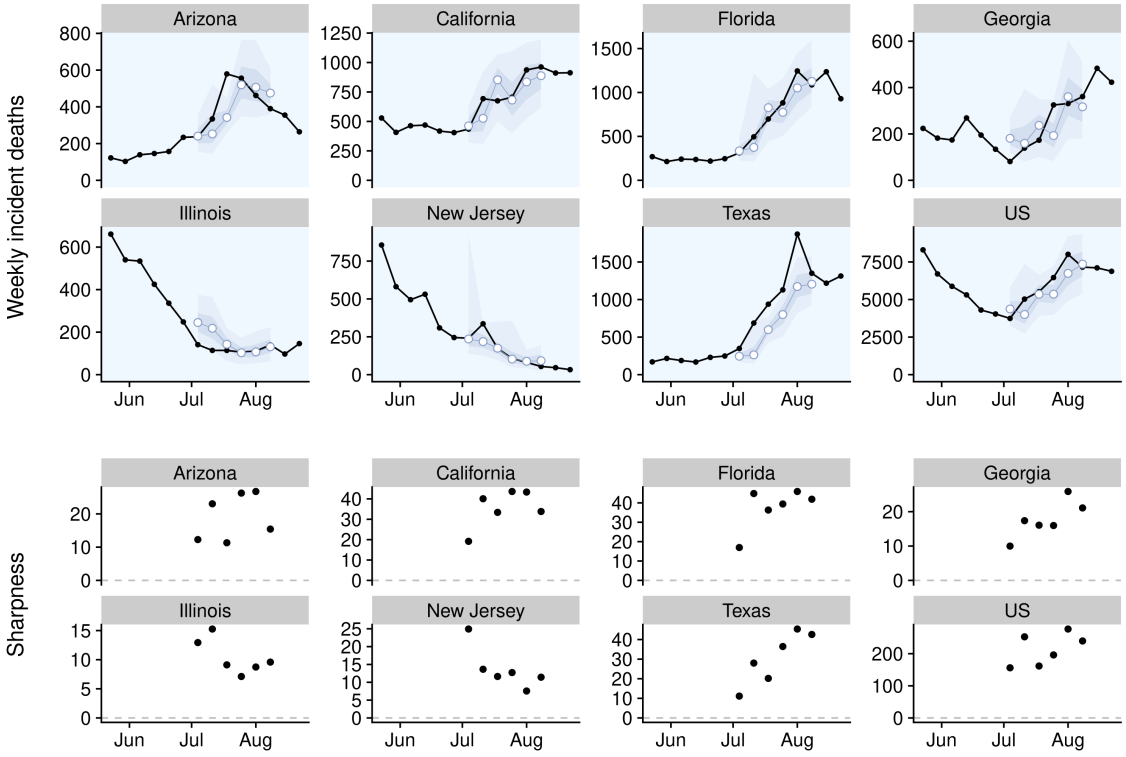

Figure 5.21: Predictions (top) and sharpness (bottom) of the crps-ensemble model in different locations.

The COVIDhub-ensemble and the mean-ensemble serve as a benchmark and a control for this analysis. The mean-ensemble is a control in the sense that all models in the mean-ensemble are also included in the COVIDhub-ensemble. If the mean-esemble had performed better than the COVIDhub-ensemble, then this would have indiciated that we had selected models that performed better than the average model submitted to the Forecast Hub. In this case, the crps- and qra-ensemble could have beaten the COVIDhub-ensemble by pure chance, by just randomly selecting models that performed better than the COVIDhub-ensemble. We should, on the other hand, hope for any ensemble not to perform significantly worse than the COVIDhub-ensemble, as it could always have improved its performance by giving a weight of one to the COVIDhub-ensemble.

Looking at the how ensemble weights evolved over time can give us insights about how the ensembles work. Figure 5.22 visualises this evolution. We can see that both model aggregation approaches chose similar models for their ensembles and that QRA weights seem to have stayed slightly more stable over time. It is interesting to see that both aggregation techniques included models like CU-select and UT-Mobility in the ensemble, even though those were not among the top performers.Instead, it seems they were able to add something of value to the ensembles even in spite of their problems.

Figure 5.23 allows us to examine this in more detail and shows the weights over time against the WIS of the models. Performance is only shown for the first two horizons as these were the only ones that influenced the ensembles. Note also that performance and weights are shown on the day of the forecast, so we should see a two-week delay between a forecast and the evaluation of the forecast for the ensemble formation. We see this delay for example on July 13th when the weight given to the UT-Mobility by the crps-ensemble suddenly jumped up two weeks after the UT-Mobility model showed good performance. We can, however, also see that the weaker models, like the CU-select model, were not only included because of evaluation delays. Rather, Figure 5.23 seems to confirm that even suboptimal models can play an important role in an ensemble. This strengthens the case

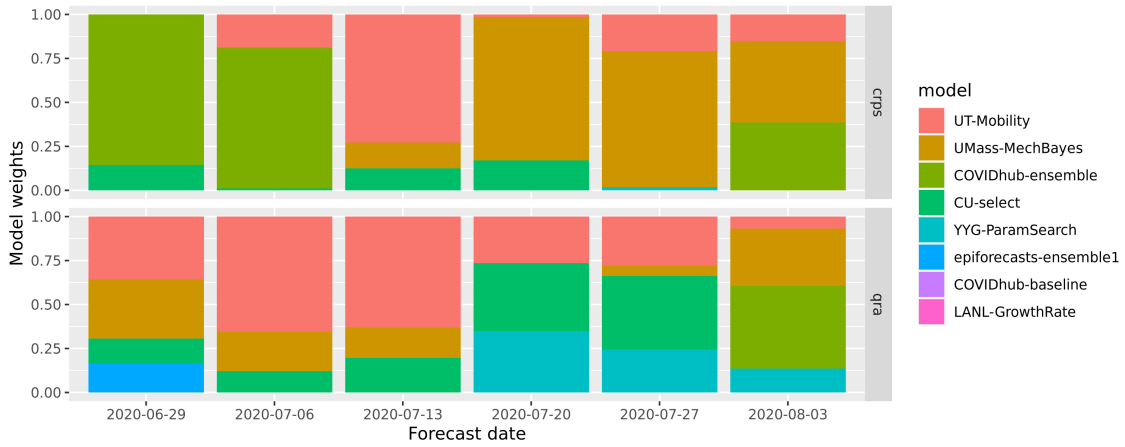

Figure 5.22: Weights given to the different models in the two ensembles over time

Table 5.3: Overview of the ensemble variants tested

Model name	Type	Weeks of past data included	Horizon that was optimised for
crps-ensemble-1-1	CRPS	1	1
crps-ensemble-2-1	CRPS	2	1
crps-ensemble-2-2	CRPS	3	2
crps-ensemble-3-1	CRPS	4	1
crps-ensemble-3-2	CRPS	4	2
crps-ensemble-3-3	CRPS	4	3
crps-ensemble-4-1	CRPS	4	1
crps-ensemble-4-2	CRPS	4	2
crps-ensemble-4-3	CRPS	4	3
crps-ensemble-4-4	CRPS	4	4
crps-ensemble-metalog-2-2	CRPS / metalog	2	2
qra-ensemble-1	QRA	1	
qra-ensemble-2	QRA	2	
qra-ensemble-3	QRA	3	
qra-ensemble-4	QRA	4	

for including diverse models into an overall ensemble.

So far we have looked at only one particular version of the crps- and the qra-ensemble with very specific parameter choices. For the purpose of a fair evaluation it makes sense to choose a sensible default instead of optimising the parameter settings in order to avoid overfitting. It is nevertheless interesting to see how other ensemble variants would have performed. This section therefore explores multiple ensemble variations with different parameter settings. An overview of the different variants explored is shown in Table 5.3

For this analysis we ran the QRA ensembles with one to four weeks of past forecasts (called qra-ensemble-1 to qra-ensemble-4). The qra-ensemble-2 corresponds to the default qra-ensemble used throughout this chapter. For the crps-ensemble we also explored variants with one to four weeeks of past data included. As `stackr` currently does not support multiple horizons, we also varied the horizon we optimised four. The second number in the crps-ensemble name therefore indicates the horizon for which the CRPS was optimised. The default model corresponds to crps-ensemble-2-2. Results slightly differed from the ones shown in Figure 5.2 as the model changes every run due to random sampling. In addition to these variants, we also reran the crps-ensemble-2-2 model with a metalog distribution (Keelin, 2016) fitted to samples instead of a gamma distribution. The metalog distribution should in theory be more flexible and better able fit the distribution. Figure 5.24 shows aggregate model performance for the different ensemble variants. No obvious picture emerged regarding the superiority of either QRA or CRPS ensembles. There were, however, a

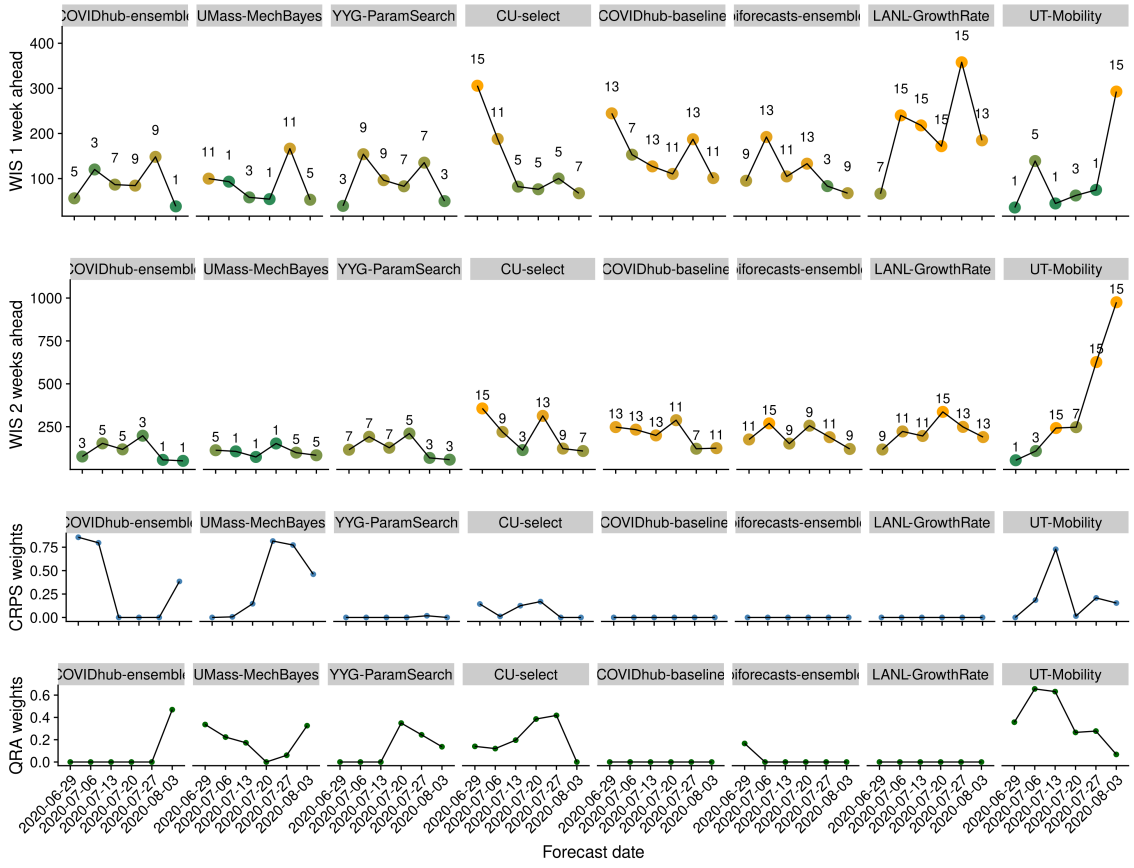

Figure 5.23: Performance (top) and weights in the two ensembles (bottom) of the eight orginal models over time. Colouring and the numbers in the top represent the model rank in terms of WIS among all the predictions from the original models made at that forecast date.

couple of interesting patterns to observe. Firstly, crps-ensembles optimised only on one-week-ahead forecast horizon tended to do worst, while those optimised on three and especially two weeks did best. For the CRPS ensemble it seems that the forecast horizon mattered more than the number of past forecasts included in the weighting. This is somewhat surprising given that qra-ensemble-1 (implying optimisation on one week ahead forecasts) is among the top performers. We would therefore expect the qra-ensemble-1 to perform similarly to crps-ensemble-1-1. We can also see that the qra-ensemble-4 and qra-ensemble-1 were top performers, while qra-ensemble-3 and qra-ensemble-2 were not. This casts some doubt whether there is a clear best choice of the number of past observations to include. We can see that the metalog distribution did not perform better than the gamma distribution.

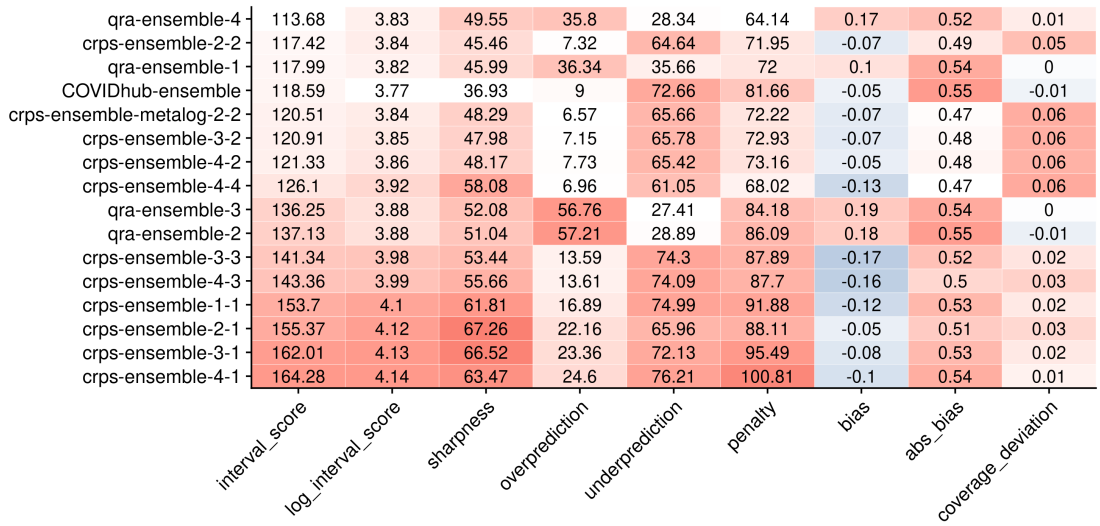

Figure 5.24: Summarised scores for all ensemble variants explored.

Table 5.4 shows the results of a regression of the WIS on the different ensemble variants with the COVIDhub-ensemble model as the baseline. Given that estimates were not significant, we could distinguish the top performing qra-ensembles (one and four weeks of past data included) as well as the crps-ensembles optimised for two-week-ahead predictions from the COVIDhub-ensemble model. But again, we also see that none of the models was able to beat the COVIDhub-ensemble model. The qra-variants with two and three weeks of data and all other crps-ensembles did significantly worse, but we should not treat this as definitive evidence, given the limitied number of observations. Especially for the optimal amount of data to include in QRA ensembles, the observed pattern does not warrant strong conclusions.

5.8 Sensitivity analysis

In order to test the validity and robustness of the results obtained, this section presents a small sensitivity analysis. We looked at three different alternative scenarios. Scenario 1 simply removed the last time point from the evaluation. Scenario 2 removed the state New Jersey that has known data issues. Scenario 3 removed the last two evaluation time points as well as the US and New Jersey. Figure 5.25 shows the summarised scores for the baseline as well as the other three scenarios. We can see that model rankings remained relatively stable across the different scenarios, which should increase the confidence in our findings. We can see that removing the US as the region with the largest average WIS had a more pronounced impact on overall model ranking. One can argue whether or not we should give a lot of weight to states with high death numbers. One the one hand is average WIS heavily influenced by these locations. Model rankings may therefore not reflect 'typical' performance. On the other hand do we most likely also care most about locations with large death numbers and should therefore be willing to accept a higher weight for these locations.

Table 5.4: Mixed model regression of the log Weighted Interval Score on model (fixed), state, and forecast date (both random)

	Estimate	Std. Error	df	t value	Pr(>|t|)
modelqra-ensemble-1	0.0493050	0.0527252	4750.98757	0.9351314	0.3497681
modelqra-ensemble-4	0.0545820	0.0527252	4750.98757	1.0352166	0.3006204
modelcrps-ensemble-2-2	0.0645933	0.0527252	4750.98757	1.2250932	0.2206008
modelcrps-ensemble-metalog-2-2	0.0704788	0.0527252	4750.98757	1.3367206	0.1813779
modelcrps-ensemble-3-2	0.0811276	0.0527252	4750.98757	1.5386871	0.1239473
modelcrps-ensemble-4-2	0.0889305	0.0527252	4750.98757	1.6866806	0.0917304
modelqra-ensemble-3	0.1092535	0.0527252	4750.98757	2.0721306	0.0383071
modelqra-ensemble-2	0.1096244	0.0527252	4750.98757	2.0791658	0.0376556
modelcrps-ensemble-4-4	0.1459303	0.0527252	4750.98757	2.7677525	0.0056663
modelcrps-ensemble-3-3	0.2095602	0.0527252	4750.98757	3.9745758	0.0000716
modelcrps-ensemble-4-3	0.2202571	0.0527252	4750.98757	4.1774545	0.0000300
modelcrps-ensemble-1-1	0.3251869	0.0527252	4750.98757	6.1675825	0.0000000
modelcrps-ensemble-2-1	0.3456434	0.0527252	4750.98757	6.5555651	0.0000000
modelcrps-ensemble-3-1	0.3635872	0.0527252	4750.98757	6.8958924	0.0000000
modelcrps-ensemble-4-1	0.3673354	0.0527252	4750.98757	6.9669818	0.0000000
(Intercept)	3.7580580	0.3196030	13.91018	11.7585199	0.0000000

Overall, however, we did not see very large differences between the different scenarios.

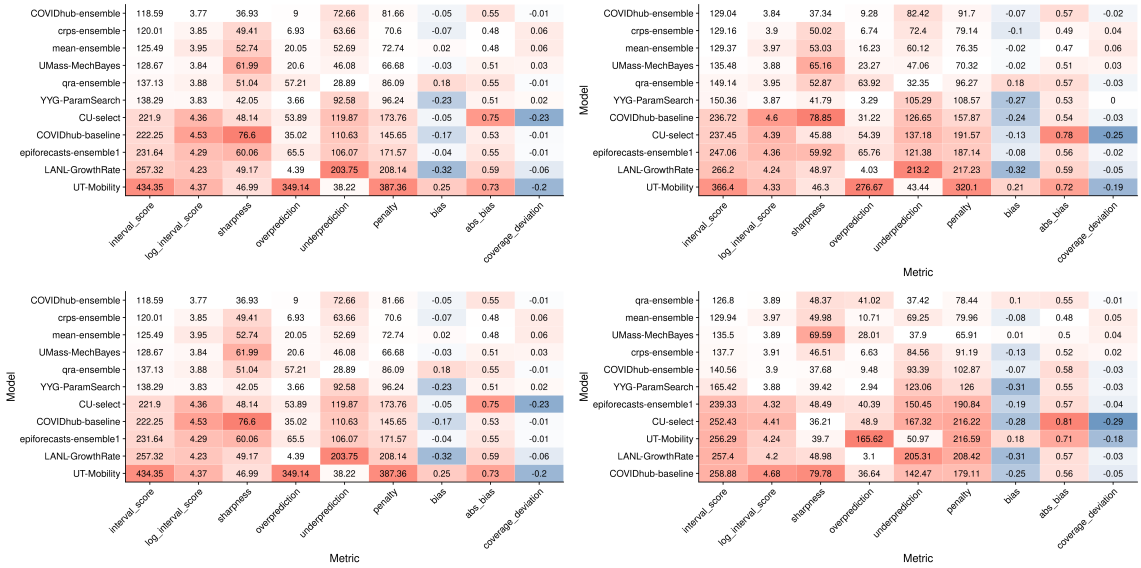

Figure 5.25: Summarised scores for all eleven models in the four different scenarios, 'baseline' (top left), '1' (top right), '2' (bottom left) and '3' (bottom right).

We also conducted a small sensitivity analysis for the ensemble variants. Figure 5.26 shows aggregated scores for all ensemble variants for the four different scenarios. Results also seem relatively stable for the ensemble models. We can see that the qra-ensemble-4 consistently stayed at the top in terms of the Weighted Interval Score. The qra-ensemble-1 fared almost equally well, while the qra-ensemble-3 and qra-ensemble-4 lagged in performance. Only in scenario 3 were all QRA ensemble at the top. We should nevertheless be careful to conclude that one approach was superior to the other, given that many of the CRPS ensembles performed almost equally well. In terms of log WIS, differences were even smaller. The results weakly suggest that the QRA ensembles tended to exhibit slightly less variance in their performance across different parameters. But since we included more CRPS ensembles we should also expect more variation in their performance. The

one observation that again stands out the most is the strong difference between the qra-ensemble-1 and the crps-ensemble-1-1.

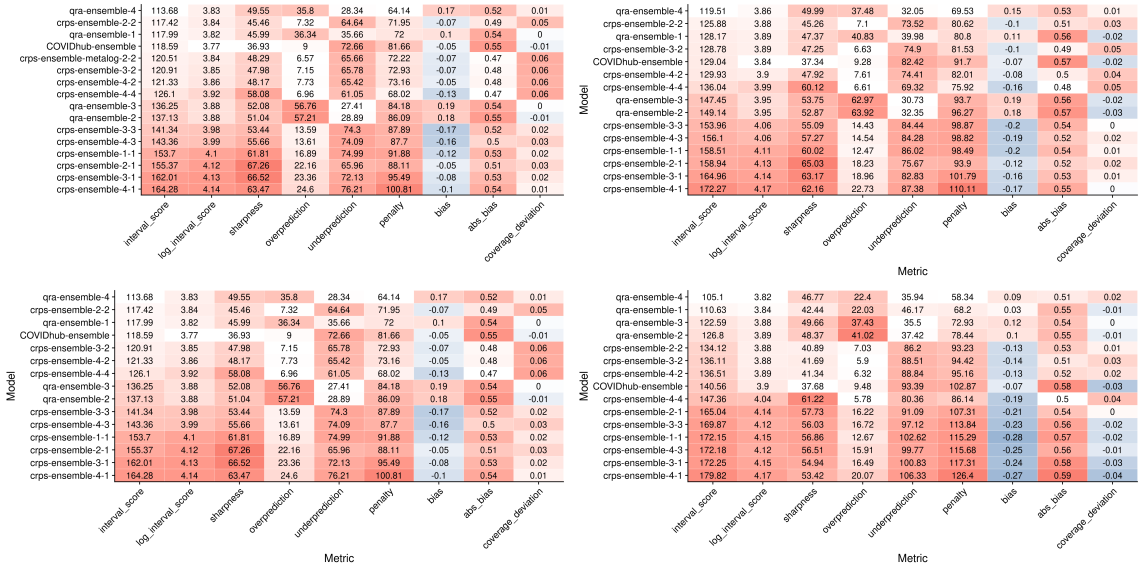

Figure 5.26: Summarised scores for the ensemble variants in the four different scenarios, 'baseline' (top left), '1' (top right), '2' (bottom left) and '3' (bottom right)

Chapter 6

Summary and discussion

Thesis summary

In this thesis we have given an extensive overview of different theoretical aspects of model evaluation and model aggregation and have applied this theory to the case study of eight models from the US Forecast Hub. We have discussed the forecasting paradigm and the notions of calibration and sharpness, and have presented several well-studied evaluation metrics as well as some previously unpublished. Based on these theoretical considerations we proposed a structured evaluation approach that was subsequently applied to data from the Forecast Hub. We also presented the `scoringutils` package that facilitates the evaluation process. Building on the discussion of proper scoring rules, we introduced two different model aggregation techniques. One of them constructs a quantile average ensemble to minimise the weighted interval score, while the other forms a mixture distribution based on the Continuous Ranked Probability Score. In order to implement the CRPS ensemble approach, we have introduced the `stackr` package. After the theoretical discussion of model evaluation and model aggregation, we provided background on the Forecast Hub and presented the data and models analysed in the ensuing case study. Applying the tools presented previously, we evaluated the performance of the eight Forecast Hub models and the three ensembles in detail.

This thesis had three main objectives: To obtain a deeper understanding of model evaluation, to explore ways to aggregate models to ensembles, and to facilitate model evaluation and model aggregation by creating appropriate tools. The following paragraphs address these objectives in the order in which they were discussed in this thesis, followed by a discussion of limitations and suggestions for future research.

Results summary

In order to facilitate model evaluation and model aggregation, we have successfully developed the `scoringutils` and `stackr` packages and demonstrated their value throughout this thesis. Both packages are still under active development and we hope to use them in future applications beyond this thesis.

By conducting a detailed evaluation of the Forecast Hub models and the ensembles, we obtained a deeper understanding of the models analysed as well of the evaluation process itself and the metrics involved. The structured evaluation approach proposed in Chapter 2 provided a useful guideline and framework for this evaluation process. We found that the metrics discussed measured different aspects of performance, but all had a significant influence on the Weighted Interval Score. This suggests that they were all able to contribute something useful in terms of model evaluation and therefore merit their own attention.

The examination of relationships between the metrics should be interpreted in view of trends in the actual data. While we would have expected a negative correlation between sharpness and penalties for over- and underprediction, we saw the opposite. This suggests that even though we should see

a trade-off, better performing models tended to do better on both accounts. We also observed that in the data positive coverage deviation was associated with smaller WIS. While unnecessarily high interval coverage should increase Interval Scores, the affected models still performed better on average than models with lower interval coverage.

For better performing models, the sharpness component of the WIS played a larger relative role for overall scores than for models that struggled with calibration. For those, WIS performance was dominated by the penalties for over- and underprediction. This finding is plausible in light of our goal to maximisie sharpness subject to calibration. Overall Interval Scores were dominated by predictions around the median of the predictive distributions, while tails mattered less. This can be attributed to the construction of the WIS that gives less weights to outer prediction intervals, as well as to the fact that outer prediction intervals incur fewer misprediction penalties. Ranges around the 50% prediction interval contributed most to sharpness. We explored two slightly different ways of assessing systematic over- or underprediction.

The bias metric presented in Chapter 2 proved to be more reflective of the general tendency of a model to over- or underpredict. The misprediction penalty component of the WIS was more heavily influenced by extreme values. Unfortunately, it remained unclear which of these two measures should take precedence in evaluation or model improvement. We observed that better performing models were also noticeably better calibrated. We were mostly able to diagnose and locate calibration issues very precisely using coverage plots and PIT histograms. However, we also saw that the aggregate picture may be misleading or at least incomplete. This became most apparent with the epiforecasts-ensemble1 model that showed good calibration on the aggregate level, but sometimes exhibited miscalibration in individual locations. The models analysed showed a lot of variation in terms of sharpness. Comparing sharpness between models therefore only made sense after splitting the models in two groups of about equal performance. Within these groups, we could observe a tendency for better models to also be sharper. We could, however, not see models consistently increase or decrease their sharpness in response to past performance.

We explored the QRA and CRPS ensembles alongside the mean-ensemble and the COVIDhub-ensemble that served as control and benchmark. We found that all ensemble models tended to perform very well, but could not clearly identify one ensemble that was superior to the others. For the qra-ensemble, taking four weeks of past observations into account was optimal consistently. We also observed that the qra-ensemble with only one week of past data performed almost equally well, while the ones with two and three weeks did not. This finding should therefore be treated with caution. For the crps-ensemble, we found that optimising for two-weeks-ahead forecasts consistently led to the best performance. This should again be regarded as limited evidence, given the small numbers of models, time points and locations analysed here. The crps-ensemble was able perform similarly well as the qra-ensemble, even though the model aggregation process involved fitting a distribution to quantiles. For most crps-ensemble variants we used a gamma distribution, but also fitted a metalog-distribution to examine the sensitivity of the results. We found that this did not improve performance significantly. This suggests that the imprecision introduced by fitting a gamma distribution did not have a large effect. Inaccuracies mostly affected the tails of the predictive distributions, which only had a small weight in terms of overall WIS. As we did not thoroughly test the metalog-distribution ensemble, research is needed to come to more definitive conclusions. We observed a tendency for the qra-ensemble to overpredict, while the crps-ensemble exhibited a slight downwards bias. We were, however, not able to provide an explanation for this result. When examining ensemble weights we observed that even models not among the top performers contributed to both ensembles. This suggests that including a greater number of diverse models in the ensemble may help improve performance.

Limitations

Limited knowledge of the model details made it hard to point to specific model characteristics that could explain better or worse performance and as well to make suggestions for improvement. We found that two out of three SEIR models were among the top performers, but this should not be considered as strong evidence in favour of SEIR models given the small number of models and observations. Models tended to perform consistently well or badly and we therefore could

not unambiguously identify relative model advantages in specific situations. Relative advantages between models were dwarfed by general performance differences. As we were not able to adequately handle missing predictions, the choice to include the epiforecasts-ensemble1 model made it necessary to restrict the set of locations and forecast dates to a very limited set. This makes it hard to generalise patterns observed throughout the model evaluation process. All results should therefore be treated with caution. On the other hand, including even more locations and models would have made it even harder to devote appropriate attention to individual models. Evaluating eleven models at the same time already meant that the majority of the analysis was conducted on an aggregate level. The analysis was also limited by the the need to develop appropriate software. We were, for example, not able incorporate more than one forecast horizon in the CRPS ensemble or could not create PIT histograms without fitting samples to the quantiles first. Not having predictive samples available also limited comparability between the CRPS and QRA ensemble models.

Outlook and future research

A variety of interesting research questions warrant further investigation. Most notably, the evaluation process demonstrated here could be expanded to all models and observations from the Forecast Hub. While the evaluation results obtained here are only of limited value due to the small number of models and observations included, a full evaluation would be of great interest to researchers and policy makers. Results could then be compared with other analyses that are currently being conducted. A full evaluation would also allow us to determine whether the CRPS and QRA ensembles could have outperformed the COVIDhub-ensemble, had they had access to the identical candidate models to inform their ensemble distributions. Ideally, we would also like to see an equally weighted mixture distribution alongside the COVIDhub-ensemble (an equally weighted quantile average ensemble). This would make it possible to determine whether one of the two strategies to combine predictive distributions is superior to the other, irrespective of the optimisation strategy.

Further research could also be done with respect to improving the two model aggregation techniques. The CRPS ensembles could potentially be improved by allowing for more than one forecast horizon to inform weights. As we observed a tendency of the QRA to ensemble to overpredict, we suggest to investigate whether this is a consistent property of the QRA that could be leveraged to further improve the model aggregation approach. In addition, weights for both models could be varied by state instead of having one weight per location. This would make it possible for the ensemble to leverage relative advantages that models may enjoy in specific locations or circumstances. Ideally, one would probably like to approach this in a hierarchical framework, where weights would be shared across locations unless there is strong evidence that weights should be different in a specific location. Analogously, different ranges can be weighted differently for QRA ensembles, but we do not currently have good knowledge of situations in which this might be advantageous. To improve how models adapt their sharpness in response to past performance, we would also like to investigate post-processing of forecasts based on accordance with past observations.

With regards to the tools presented in this thesis, we plan to add the plots presented throughout this thesis to the `scoringutils` package. Ideally, we would also like to add an automated report that users could generate to obtain feedback for their model. We also intend to increase the flexibility of the `stackr` package by allowing for multiple forecast horizons at a time. Overall, we hope this thesis has provided an important first step that motivates and enables future research.

Appendix A

Appendix

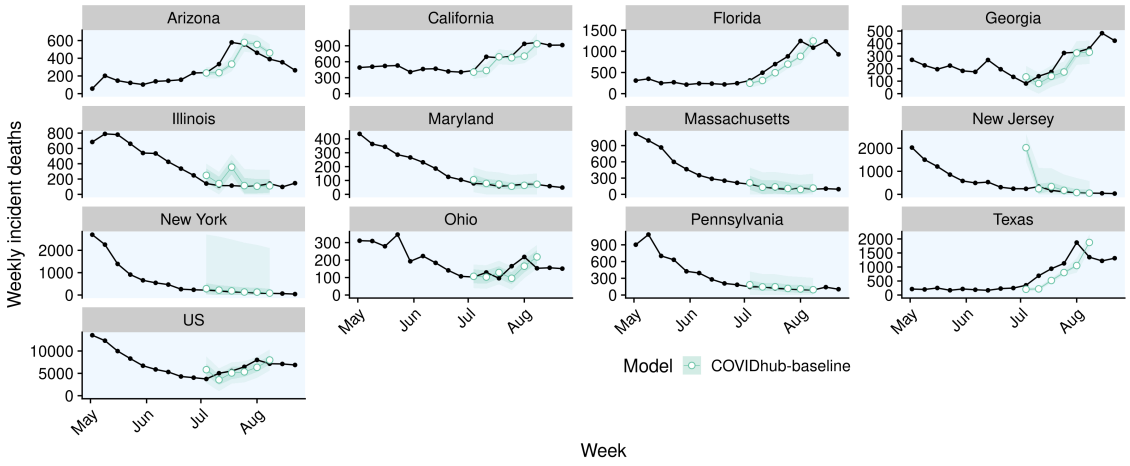

Figure A.1: One week ahead forecasts for all locations for the COVIDhub-baseline model.

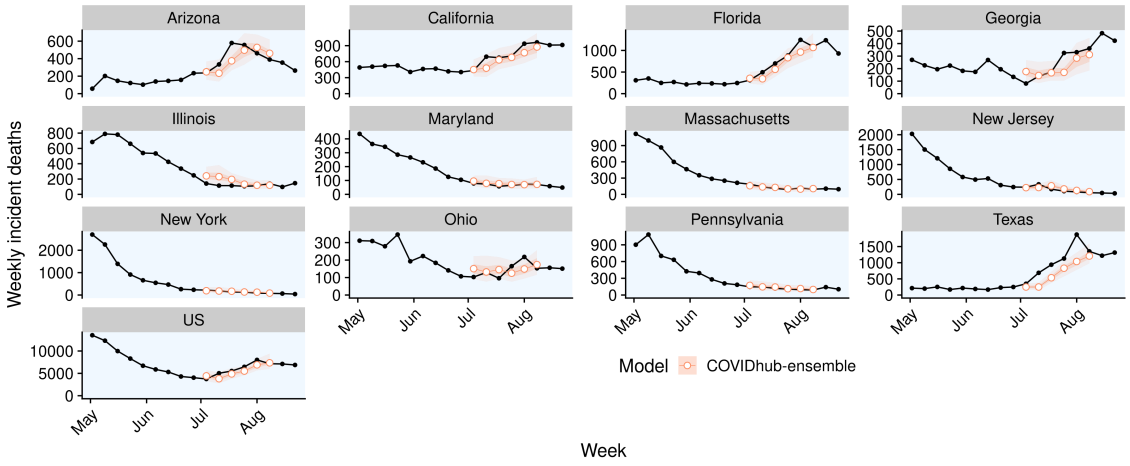

Figure A.2: One week ahead forecasts for all locations for the COVIDhub-ensemble model.

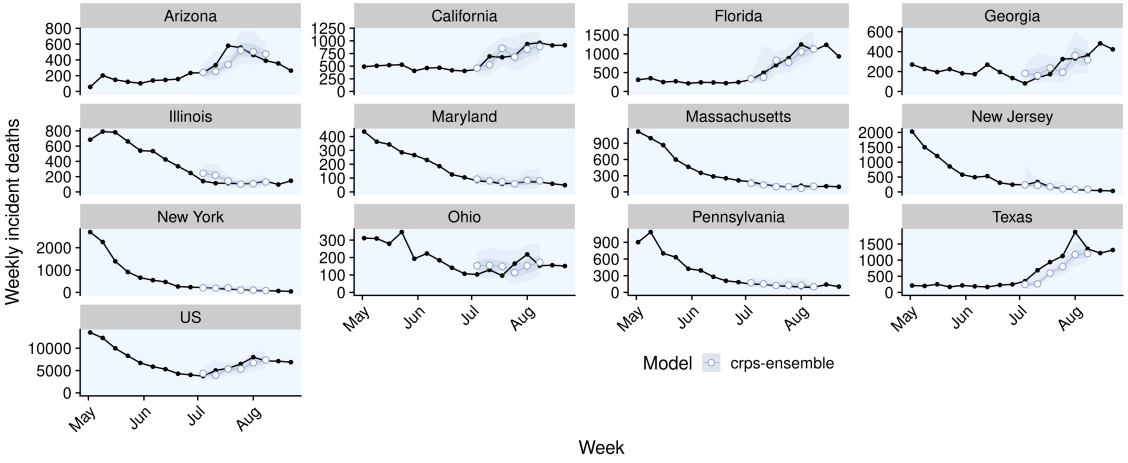

Figure A.3: One week ahead forecasts for all locations for the crps-ensemble model.

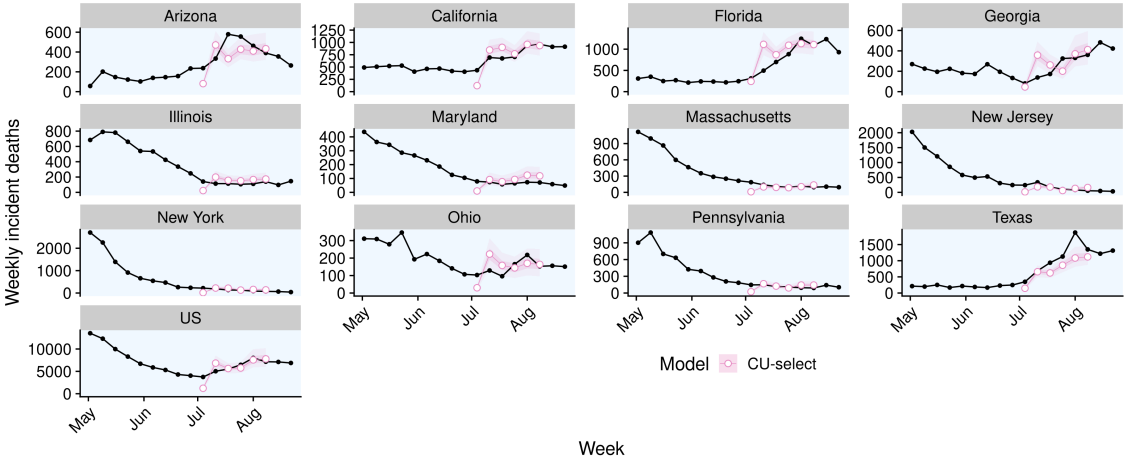

Figure A.4: One week ahead forecasts for all locations for the CU-select model.

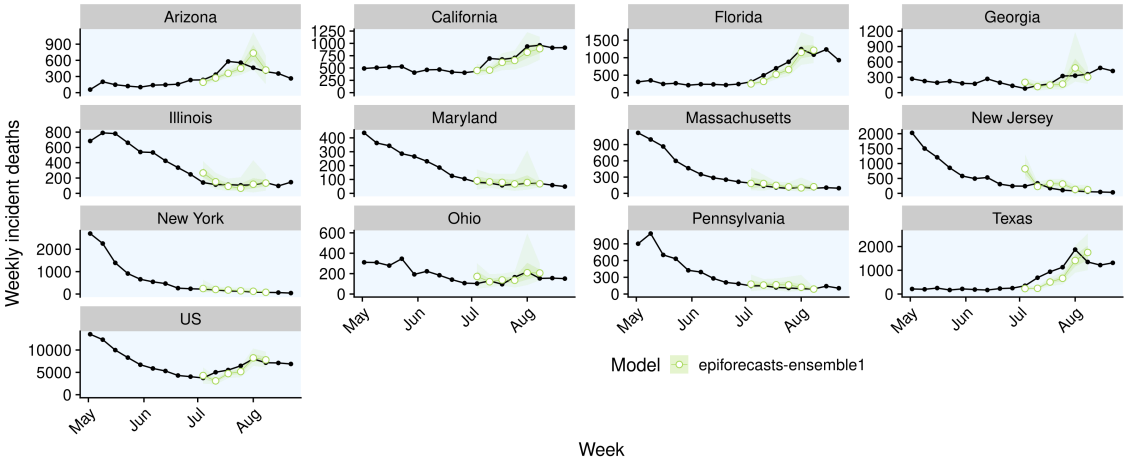

Figure A.5: One week ahead forecasts for all locations for the epiforecasts-ensemble1 model.

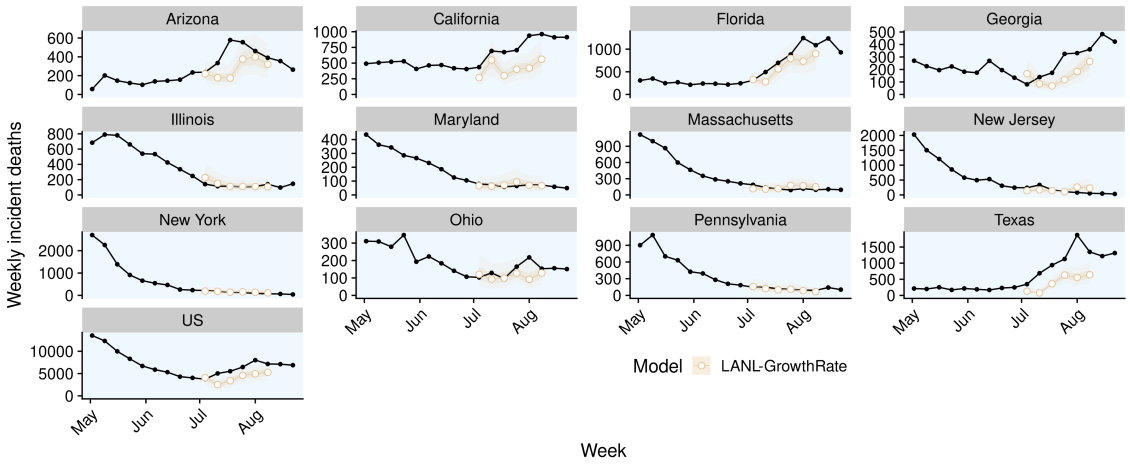

Figure A.6: One week ahead forecasts for all locations for the LANL-GrowthRate model.

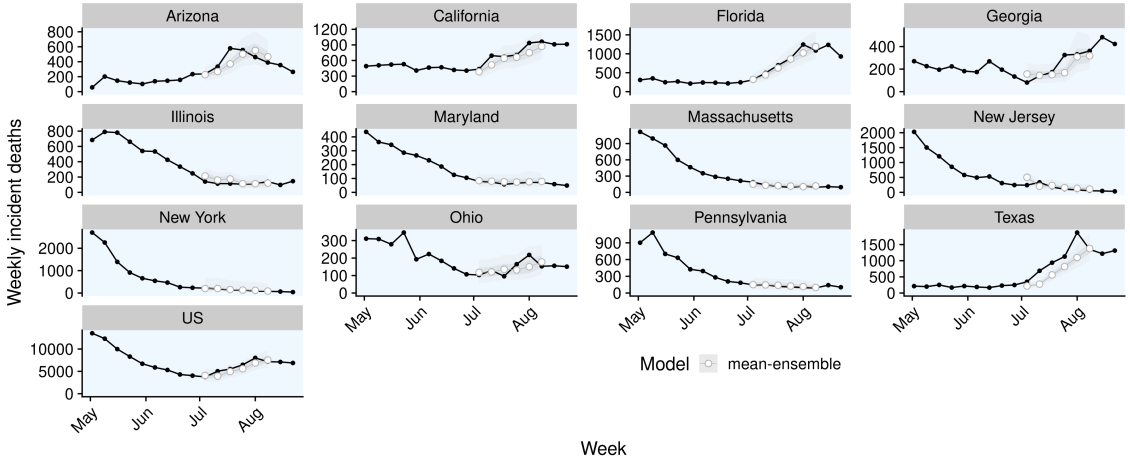

Figure A.7: One week ahead forecasts for all locations for the mean-ensemble model.

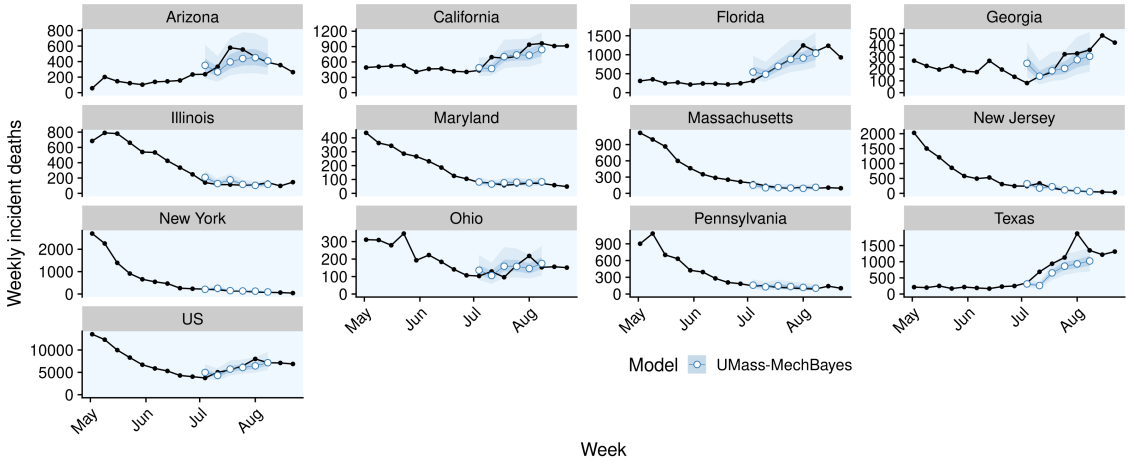

Figure A.8: One week ahead forecasts for all locations for the UMass-MechBayes-ensemble model.

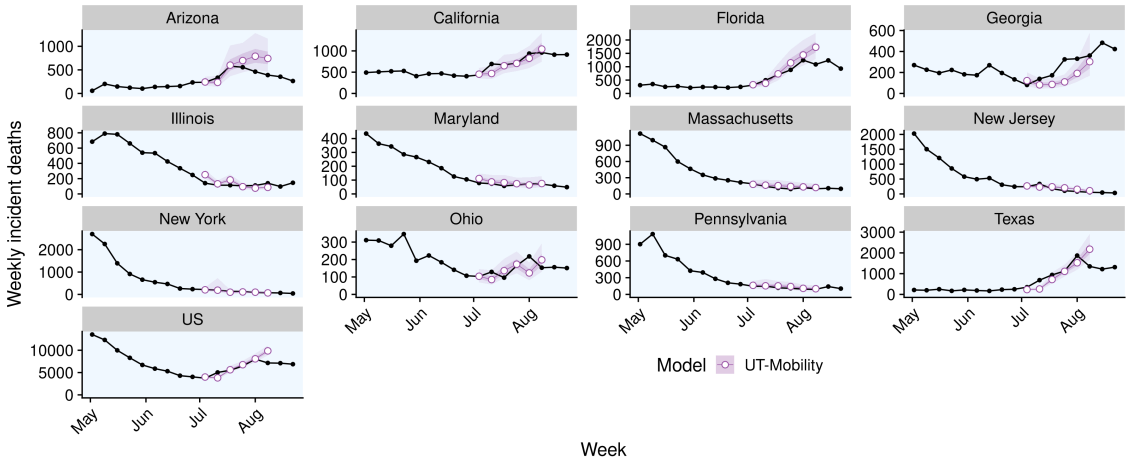

Figure A.9: One week ahead forecasts for all locations for the UT-mobility model.

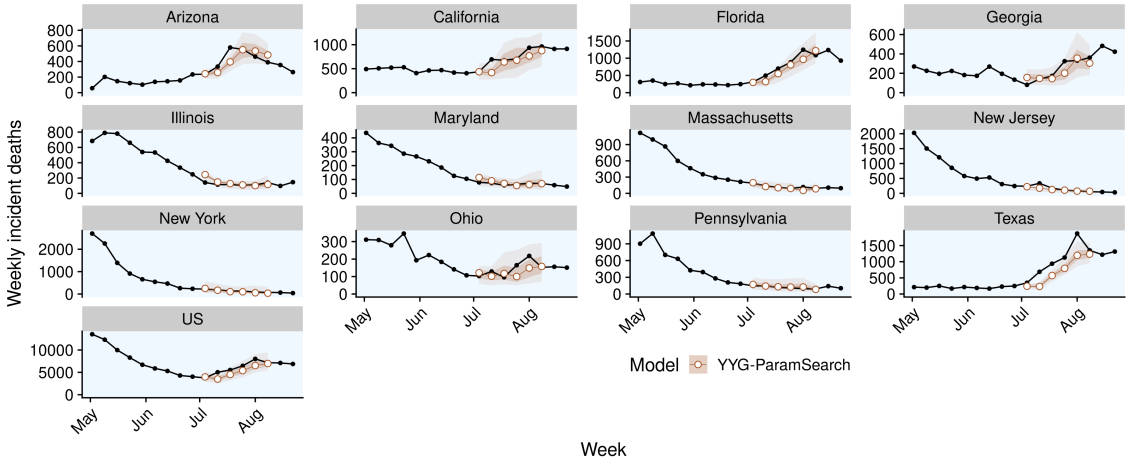

Figure A.10: One week ahead forecasts for all locations for the YYG-ParamSearch model.

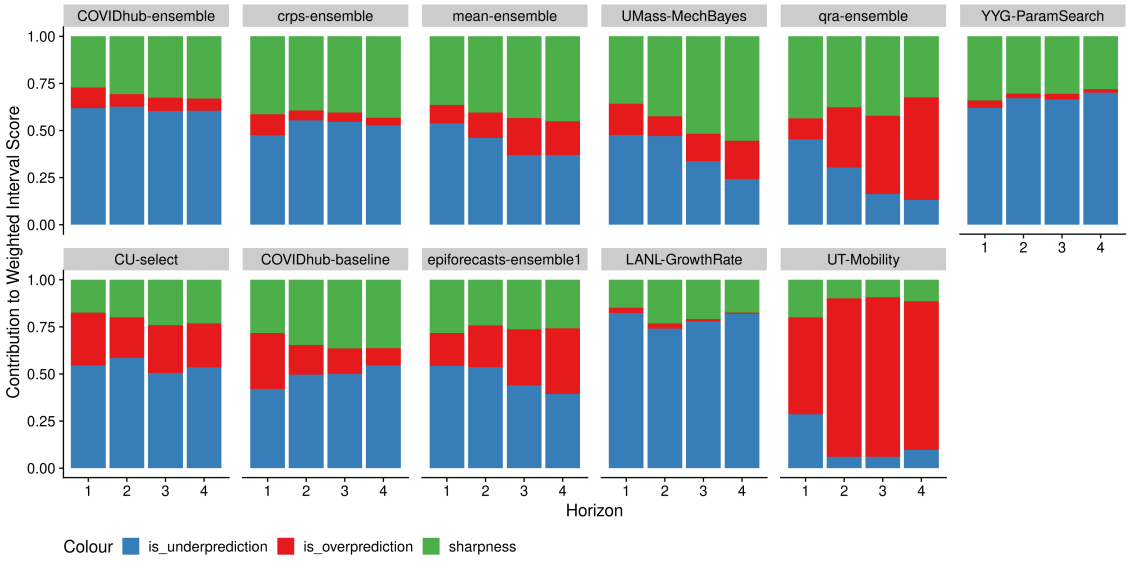

Figure A.11: Relative contributions to the Weighted Interval Score from its underprediction (blue), overprediction (red) and sharpness (green) components.

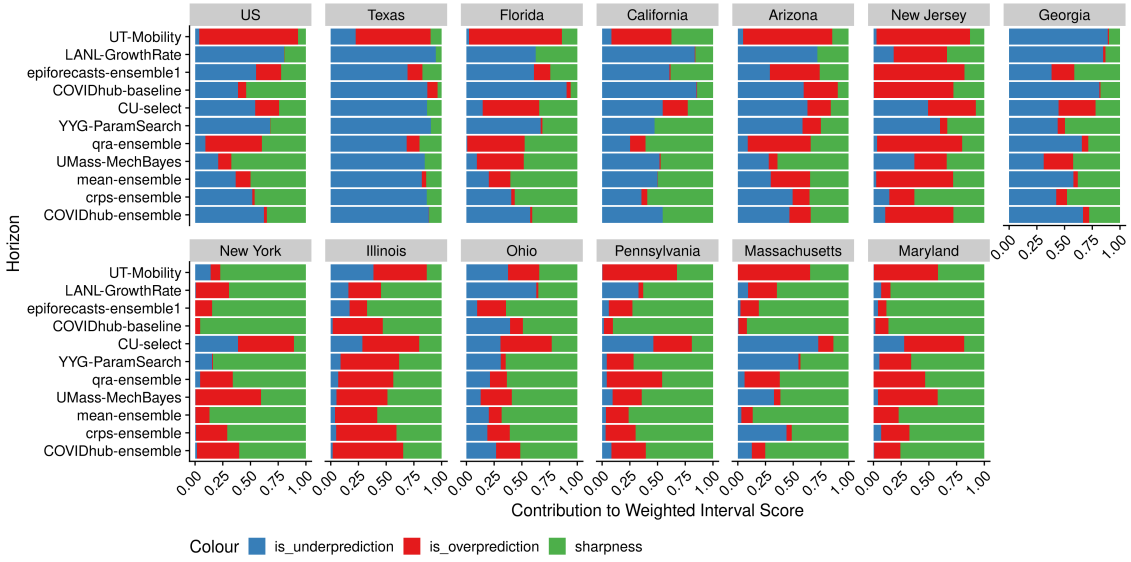

Figure A.12: Relative contributions to the Weighted Interval Score from its underprediction (blue), overprediction (red) and sharpness (green) components in different locations.

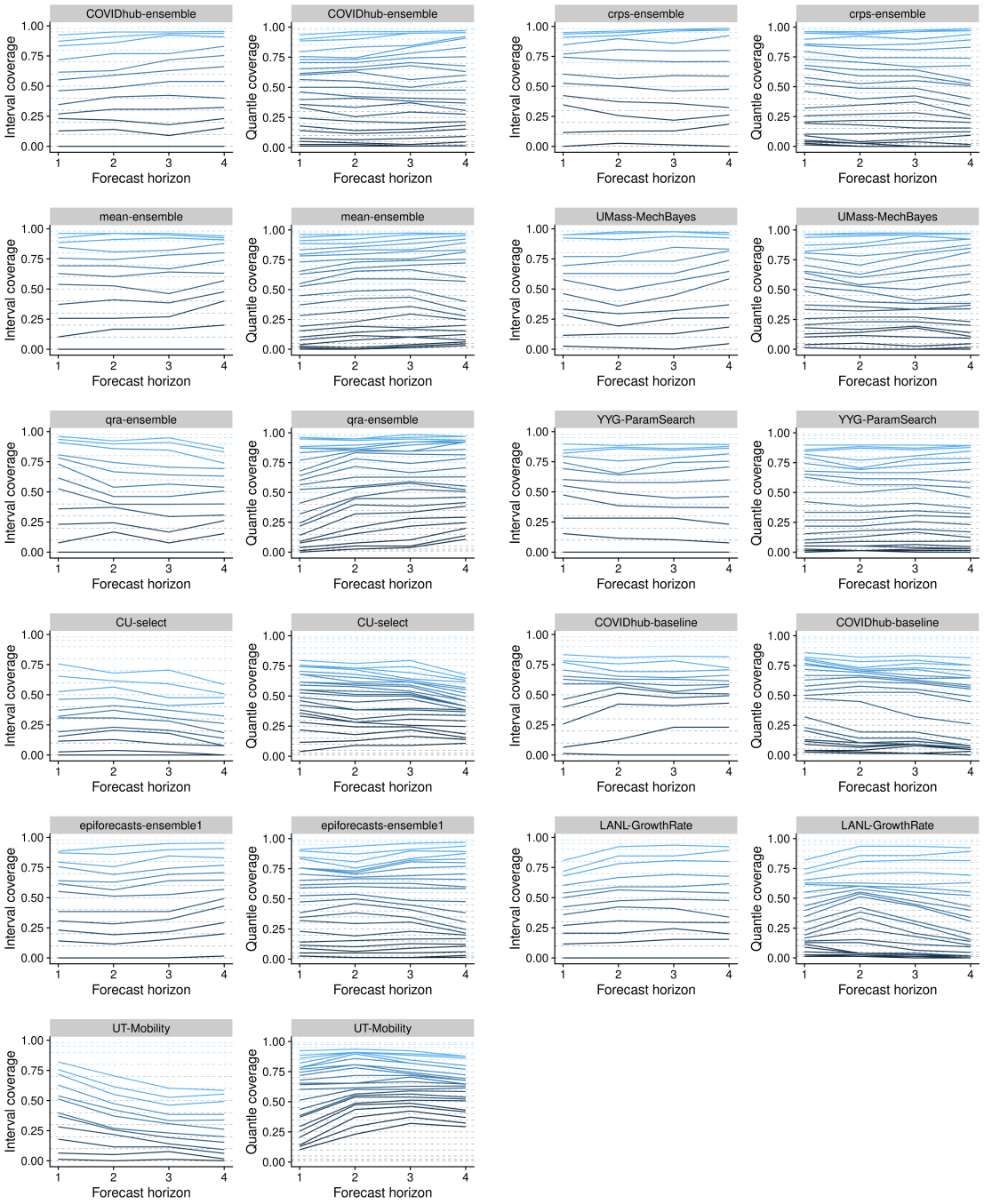

Figure A.13: Evolution of interval and quantile coverage over multiple horizons..

Bibliography

Anderson, T. W. and Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193–212.

Angus, J. E. (1994). The Probability Integral Transform and Related Results. *SIAM Review*, 36(4):652–654. Publisher: Society for Industrial and Applied Mathematics.

Bosse, N., Sam Abbott, and Funk, S. (2020a). *scoringutils: Utilities for Scoring and Assessing Predictions*. R package version 0.1.2.

Bosse, N., Yao, Y., Abbott, S., and Funk, S. (2020b). *stackr: Create Mixture Models From Predictive Samples*. R package version 0.1.0.

Bracher, J., Ray, E. L., Gneiting, T., and Reich, N. G. (2020). Evaluating epidemic forecasts in an interval format. *arXiv:2005.12881 [q-bio, stat]*. arXiv: 2005.12881.

Brauer, F. (2008). Compartmental Models in Epidemiology. In Brauer, F., van den Driessche, P., and Wu, J., editors, *Mathematical Epidemiology*, Lecture Notes in Mathematics, pages 19–79. Springer, Berlin, Heidelberg.

Bröcker, J. (2009). Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519.

Cori, A., Ferguson, N. M., Fraser, C., and Cauchemez, S. (2013). A New Framework and Software to Estimate Time-Varying Reproduction Numbers During Epidemics. *American Journal of Epidemiology*, 178(9):1505–1512.

Czado, C., Gneiting, T., and Held, L. (2009). Predictive model assessment for count data. *Biometrics*, 65(4):1254–1261.

Dawid, A. P. (1984). Present Position and Potential Developments: Some Personal Views Statistical Theory the Prequential Approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290. _eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.2307/2981683.

Dong, E., Du, H., and Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases*, 20(5):533–534. Publisher: Elsevier.

Dowle, M. and Srinivasan, A. (2019). *data.table: Extension of 'data.frame'*. R package version 1.12.8.

Epstein, E. S. (1969). A Scoring System for Probability Forecasts of Ranked Categories. *Journal of Applied Meteorology*, 8(6):985–987. Publisher: American Meteorological Society.

Funk, S., Camacho, A., Kucharski, A. J., Lowe, R., Eggo, R. M., and Edmunds, W. J. (2019). Assessing the performance of real-time epidemic forecasts: A case study of Ebola in the Western Area region of Sierra Leone, 2014-15. *PLOS Computational Biology*, 15(2):e1006785.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016.

Gneiting, T. (2010). Making and Evaluating Point Forecasts. *arXiv:0912.0902 [math, stat]*. arXiv: 0912.0902.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268.

Gneiting, T. and Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Gneiting, T., Raftery, A. E., Westveld, A. H., and Goldman, T. (2005). Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation. *Monthly Weather Review*, 133(5):1098–1118. Publisher: American Meteorological Society.

Good, I. J. (1952). Rational Decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(1):107–114. Publisher: [Royal Statistical Society, Wiley].

Guo, J., Gabry, J., and Goodrich, B. (2020). *rstan: R Interface to Stan*. R package version 2.19.3.

Hamill, T. M. (2001). Interpretation of Rank Histograms for Verifying Ensemble Forecasts. *Monthly Weather Review*, 129(3):550–560. Publisher: American Meteorological Society.

Hersbach, H. (2000). Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems. *Weather and Forecasting*, 15(5):559–570. Publisher: American Meteorological Society.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian Model Averaging: A Tutorial. *Statistical Science*, 14(4):382–401.

Hyndman, Rob J and Athanasopoulos, George (2019). *Forecasting: Principles and Practice*.

Keelin, T. W. (2016). The Metalog Distributions. *Decision Analysis*, 13(4):243–277.

Matheson, J. E. and Winkler, R. L. (1976). Scoring Rules for Continuous Probability Distributions. *Management Science*, 22(10):1087–1096. Publisher: INFORMS.

Murphy, A. H. (1969). On the "Ranked Probability Score". *Journal of Applied Meteorology*, 8(6):988–989. Publisher: American Meteorological Society.

Nishiura, H. and Chowell, G. (2009). The Effective Reproduction Number as a Prelude to Statistical Estimation of Time-Dependent Epidemic Trends. *Mathematical and Statistical Estimation Approaches in Epidemiology*, pages 103–121.

Nowotarski, J. and Weron, R. (2015). Computing electricity spot price prediction intervals using quantile regression and forecast averaging. *Computational Statistics*, 30(3):791–803.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raftery, A. E., Gneiting, T., Balabdaoui, F., and Polakowski, M. (2005). Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *Monthly Weather Review*, 133(5):1155–1174. Publisher: American Meteorological Society.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). Bayesian Model Averaging for Linear Regression Models. *Journal of the American Statistical Association*, 92(437):179–191.

Sam Abbott, Bosse, N., Hellewell, J., Sherratt, K., Munday, J., Thompson, R., Chateigner, A., Mareschal, S., Rau, A., Vialaneix, N., DeWitt, M., and Funk, S. (2020a). *EpiSoon: Forecast Cases Using Reproduction Numbers*. R package version 0.3.0.

Sam Abbott, Hellewell, J., Munday, J., Thompson, R., and Funk, S. (2020b). *EpiNow: Estimate Realtime Case Counts and Time-varying Epidemiological Parameters*. R package version 0.3.0.

Sam Abbott, Hellewell, J., Munday, J., Thompson, R., and Funk, S. (2020c). *EpiNow2: Estimate Realtime Case Counts and Time-varying Epidemiological Parameters*. R package version 0.3.0.

Shannon, C. E. (1948). A Mathematical Theory of Communication. page 55.

Shaub, D. and Ellis, P. (2020). *forecastHybrid: Convenient Functions for Ensemble Time Series Forecasts*. R package version 5.0.18.

Tibshirani, R. (2020). *quantgen: Tools for generalized quantile modeling*. R package version 1.0.0.

Tibshirani, Ryan (2020). Quantile Stacking. https://ryantibs.github.io/quantgen/stacking_example.html.

UMass-Amherst Influenza Forecasting Center of Excellence (2020). Covid19forecasthub.org. https://covid19forecasthub.org/.

Woody, S., Garcia Tec, M., Dahan, M., Gaither, K., Lachmann, M., Fox, S., Meyers, L. A., and Scott, J. G. (2020). Projections for first-wave COVID-19 deaths across the US using social-distancing measures derived from mobile phones. Preprint, Infectious Diseases (except HIV/AIDS).

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). Using stacking to average Bayesian predictive distributions. *Bayesian Analysis*, 13(3):917–1007. arXiv: 1704.02030.

Ypma, J. and Johnson, S. G. (2020). *nloptr: R Interface to NLopt*. R package version 1.2.2.2.