Mixture distributions in collaborative probabilistic forecasting of disease outbreaks

by

Spencer Gordon Wadsworth

A Creative Component submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Statistics

Program of Study Committee:
Jarad Niemi, Major Professor
Karin Dorman
Kori Khan

Iowa State University

Ames, Iowa

2022

DEDICATION

I want to dedicate this work to my uncle Bryce. At his recommendation I considered writing this about his life and accomplishments but then decided I wanted to graduate.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I want to thank Dr. Jarad Niemi for all he did to teach and mentor me while writing this creative component. His patience, scientific insights, and attention to detail have greatly enhanced my vision and appreciation for statistics and academic research. I also want to thank Dr. Karin Dorman and Dr. Kori Khan for serving on my committee.

ABSTRACT

Collaboration among multiple teams has played a major role in probabilistic forecasting events of influenza outbreaks, the COVID-19 pandemic, other disease outbreaks, and in many other fields. When collecting forecasts from individual teams, ensuring that each team's model represents forecast uncertainty according to the same format allows for direct comparison of forecasts as well as methods of constructing multi-model ensemble forecasts. This paper outlines several common probabilistic forecast representation formats including parametric distributions, sample distributions, bin distributions, and quantiles and compares their use in the context of collaborative projects. We propose the use of a discrete mixture distribution format in collaborative forecasting in place of other formats. The flexibility in distribution shape, the ease for scoring and building ensemble models, and the reasonably low level of computer storage required to store such a forecast make the discrete mixture distribution an attractive alternative to the other representation formats.

1. Introduction

Predicting the outcomes of prospective events is the object of much scientific inquiry and the basis for many decisions both public and private. Because predictions of the future can never be precise, it is usually desirable that a level of uncertainty be attached to any prediction. In recent years, it has become increasingly desirable that forecasts be probabilistic in order to account for uncertainty in predicted quantities or events (Gneiting and Katzfuss, 2014). Weather forecasting (Baran and Lerch, 2018), economics (Groen et al., 2013), and disease outbreaks (Yamana et al., 2016) are some of the areas where probabilistic forecasting is used.

A probabilistic forecast is a forecast in which possible outcomes are assigned probabilities. There are a number of ways whereby probabilities or uncertainty may be represented. A common representation is either a continuous or discrete parametric distribution, given as a probability density/mass function. Much of the literature on calibration, sharpness, and scoring of a forecast pertains to parametric distribution forecasts (Gneiting et al., 2007; Gneiting and Ranjan, 2013; Baran and Lerch, 2018). Other common representations include samples (Krueger et al., 2016), discretized bin distributions (McGowan et al., 2019), and quantile forecasts (Taylor, 2021; Bracher et al., 2021). Each representation may be more or less appropriate than the others for a given problem, but knowing how to interpret, score, and construct ensemble forecasts for a selected representation is essential when multiple teams collaborate in the same forecasting project.

Two collaborative projects on forecasting disease outbreaks for which many separate forecasts are used include the United States Centers for Disease Control (CDC) annual competition for forecasting the influenza outbreak (CDC) and the COVID-19 Forecast Hub which has continuously operated since the start of the COVID-19 pandemic in the US in early 2020 (Cramer et al., 2021a).

1.1 CDC flu competition

Since the 2013-14 flu season, the CDC has hosted an annual competition for forecasting the timing, peak, and intensity of the year's flu season. The specific events to be forecast are known as *targets*. Forecasts for these different targets also include forecasts for one, two, three, and four weeks in the future. National flu data is provided weekly to academic teams not directly affiliated with the CDC who use that data to construct forecasts using whatever methods they choose. Historically, the forecasts have been submitted in a discretized bin distribution or a bin distribution format. A *bin distribution* is a probability distribution represented by breaking the numeric range of an outcome into intervals or *bins* and directly assigning to each bin the probability that the event falls within the bin. During previous flu seasons the *binning scheme* or the assignment of bin values was on a numeric scale with a bounded range, and the prediction of a specific target was a set of probabilities assigned to each bin (McGowan et al., 2019). These forecasts were then evaluated against actual flu activity, and at the end of the season a winning team was declared (CDC).

This competition has provided the CDC, competing teams, and other interested parties a chance to collaborate and improve their forecasting from season to season. One proposed way to enhance prediction has been to aggregate the various teams' forecasts into a *multi-model ensemble forecast* (McGowan et al., 2019; McAndrew and Reich, 2019; Reich et al., 2019b), or an ensemble forecast. An *ensemble forecast* is a combination of several component forecast models into one model which often yields better predicting power than the individual models (Cramer et al., 2021b). Such an ensemble made from multiple influenza competition forecasts did in fact outperform the individual component models (Reich et al., 2019b).

1.2 COVID-19 Forecast Hub

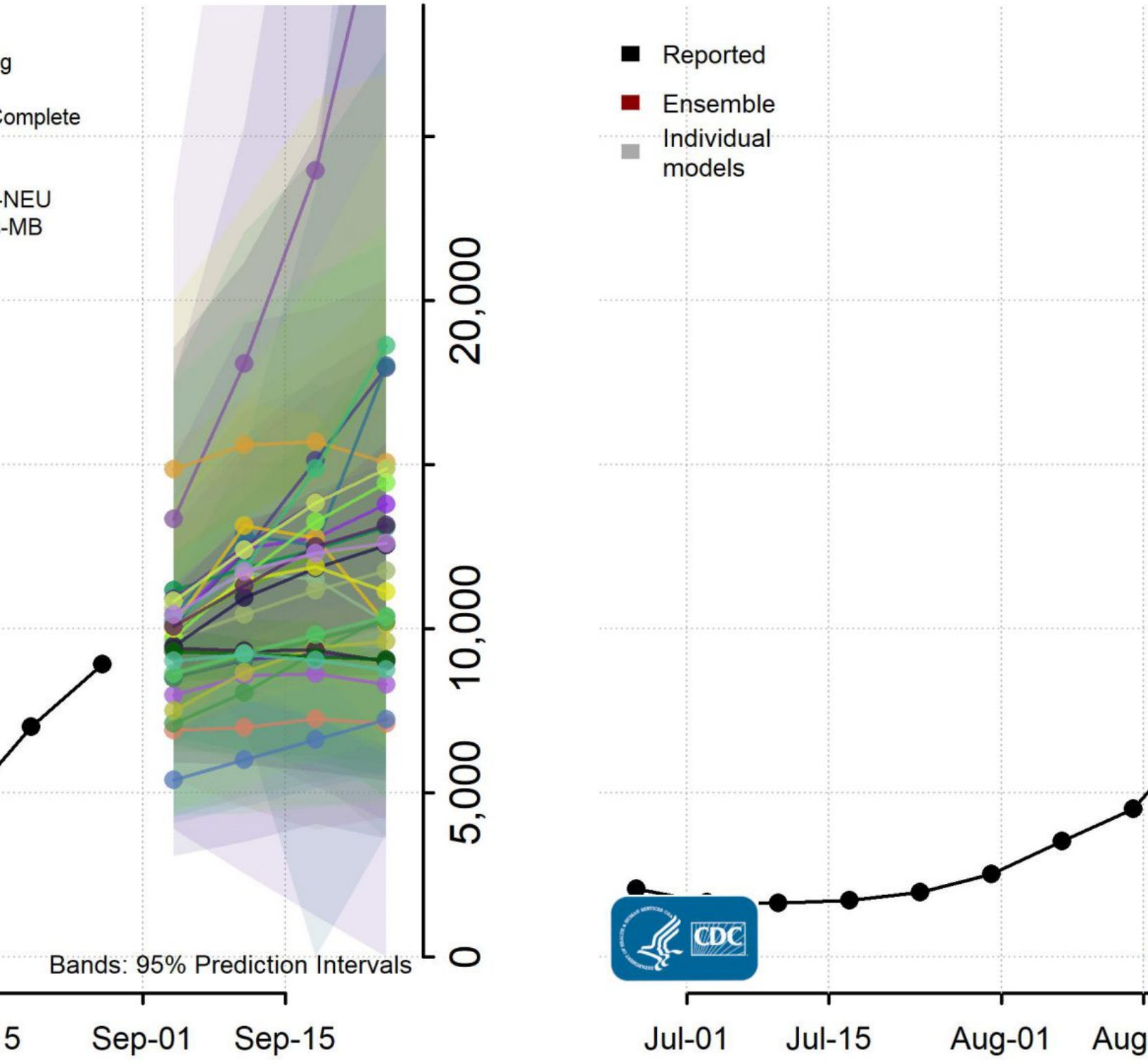
In March 2020, at the onset of the COVID-19 pandemic, the COVID-19 Forecast Hub was founded. Borrowing from the work done in the CDC flu competition, the COVID-19 Forecast Hub is a central site in which dozens of academic teams collaborate to forecast the ongoing COVID-19

pandemic. Every week relevant pandemic data is provided to these teams who construct forecast models to predict the target cases, hospitalizations, and deaths due to COVID-19. Forecasts are made on the US county, state, and national levels and for days, weeks, and months ahead. These forecasts are aggregated into a single ensemble forecast. The model data, forecasts, and the ensemble forecast are passed along to the CDC for its use in official communication (Cramer et al., 2021a). Figure 1.1 is from an official CDC report from August 2021. It shows forecasts from the COVID-19 Forecast Hub of increment deaths and cumulative deaths due to COVID-19.

Though similar to the forecasting in the CDC flu competition, the format of the COVID-19 Forecast Hub has some key distinctions. First, this project has been operating continuously since it began, so forecasts have been made for over 100 straight weeks. Second, rather than bin distributions the forecasts are requested as the predictive median and predictive intervals for various nominal levels depending on the target to be predicted (Bracher et al., 2021). Each value in a predictive interval is a value for a quantile at a specified nominal level. This makes a set of predictive intervals a *quantile forecast* or a forecast made up of a set of quantiles and corresponding values. Collecting forecasts as quantile forecasts instead of bin forecasts brings with it differences in how to score the forecasts, construct an ensemble forecast, and store the forecasts among other differences. Ray et. al show that ensemble forecasts in the COVID-19 Forecast Hub provide precise short-term forecasts which decline in accuracy in longer term forecasts approaching four weeks (Ray et al., 2020).

1.3 Outline

In the context of collaborative forecasting like that of the CDC flu competition or the COVID-19 Forecast Hub, bin forecasts and quantile forecasts have become important representations. Yet both representation types come with their drawbacks. Computer storage for instance might be a concern if many bin distributions are used for forecasting, and scoring methods are limited if forecasts are quantile forecasts. In this paper, we propose the use of finite mixture distributions as a means of forecasting for collaborative projects similar to the CDC flu


g
Complete
-NEU
-MB
Reported
Ensemble
Individual models
20,000
10,000
5,000
0
Bands: 95% Prediction Intervals
5
Sep-01
Sep-15
Jul-01
Jul-15
Aug-01
Aug
CDC

competition or the COVID-19 Forecast Hub. A finite mixture distribution –which we will refer to as a *mixture distribution*– is a distribution constructed by combining a finite collection of other distributions. In this paper, we focus on the case where the collection of distributions are parametric distributions. In Section 2, popular probabilistic forecast representation types are defined and reviewed. For each representation type, we review methods of scoring, storing, constructing ensembles, and other aspects. Section 3 presents using mixture distributions in a collaborative forecast project and discusses tools for scoring and constructing an ensemble forecast. Section 4 is a retrospective study of the CDC flu competition and COVID-19 Forecast Hub forecasts and an attempt to assess whether forecast models may be approximated by one component mixture distributions.

2. Probabilistic forecast representations

In this section we review four forecast representations already commonplace in forecasting. In a collaborative setting, certain aspects of each representation should be considered including scoring, computer storage, and how to construct an ensemble forecast. For each representation presented, applications of each of those aspects are also discussed.

2.1 Representation aspects to consider in collaborative projects

2.1.1 Scoring

Scoring rules are used to numerically evaluate or *score* a probability forecast. The score is a measure of the accuracy of the forecast and where multiple forecasts exist the score for each may be used to compare forecasts. If a scoring rule is *proper*, then the best possible score is obtained by reporting the true distribution. The rule is strictly proper if that value is unique. Under proper scoring rules, a forecaster has no incentive to be dishonest in their submission (Gneiting and Raftery, 2007). This makes proper scoring rules ideal for evaluating forecasts. We will limit our review of scoring methods to rules which are proper.

2.1.2 Storage

For a collaborative forecast project where many researchers are involved and many predictions are collected, computer storage may need to be addressed. As an example of required computer storage, the repository for the COVID-19 Forecast Hub contained 85 million forecasts as of April 4, 2022 which required more than 11.7 gigabytes of storage. (GitHub). When determining the goals of a forecast project, there should be consideration of the storage required for different forecast representation types.

2.1.3 Ensemble construction

An ensemble model is a statistical model made by combining information from two or more individual statistical models. Private and public decisions are regularly made after combining information from multiple sources. For a given problem, information from one source may provide insight on a subject which other sources fail to capture. Likewise one statistical model may provide insight that another model does not so that when they are combined into an ensemble the ensemble outperforms the individual component models.

As probabilistic forecasting becomes more commonplace, so too does ensemble modeling. Ensembles have been used extensively in weather and climate modeling (Baran and Lerch, 2018), and they have been used increasingly in modeling infectious disease outbreaks (Yamana et al., 2016). Ensembles allow for an incorporation of multiple signals –often from differing data sources– and sometimes individual model biases are canceled out or reduced by biases from other models (Reich et al., 2019b, see references therein). In several disease outbreak studies, ensemble forecasts have been shown to outperform individual model forecasts (Ray et al., 2020; Cramer et al., 2021b, see references therein). Construction of an ensemble may be done by combining individual forecast models using weighted averages. This has been called stacking (Wolpert, 1992) or weighted density ensembles (Ray and Reich, 2018).

2.2 Probabilistic forecast representation types

2.2.1 Parametric distributions

A parametric distribution is a discrete or continuous probability distribution described by a known function $p(x) := p(x|\theta)$. The function $p(x)$ is called a probability mass function (pmf) if the distribution is discrete and a probability density function (pdf) if the distribution is continuous. Here θ is a vector of parameters contained in the parameter space of the distribution.

The value of $p(\cdot)$ evaluated at x is defined as $p(x) = P(X = x)$ or the probability that the random variable X takes on x in the range of the random variable. In the continuous case

$P(X = x) = 0$ for all x, but the probability that X falls within an interval (a, b) is calculated by (2.1).

$$P(a < X < b) = \int_a^b p(x)dx \tag{2.1}$$

Other functions relative a parametric distribution include the cumulative distribution function (CDF) and the inverse CDF or quantile function. The CDF in the continuous case is defined in (2.2) and in the discrete case in (2.3). The quantile function is defined in (2.4) which returns a quantile value where p is a given probability such that $0 \leq p \leq 1$.

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(t)dt \tag{2.2}$$

$$F(x) = P(X \leq x) = \sum_{x \leq X} p(x) \tag{2.3}$$

$$Q(p) = \inf\{y \in \mathbb{R} : p \leq F(x)\} \tag{2.4}$$

For a forecast represented as a parametric distribution with pmf/pdf $p_m(\cdot)$, the accuracy of the forecast may be measured by how likely the realized value x^* is to occur. Commonly used proper scoring rules for parametric distributions include the logarithmic score (LogS), the continuous rank probability score (CRPS) (Hersbach, 2000) (Alves et al., 2013), and the interval/Brier score (IS) (Gneiting and Raftery, 2007) among others. See also (Gneiting and Katzfuss, 2014) Section 3 for more on proper scoring functions. The definitions (2.5), (2.6), and (2.15) are found in the review by Krueger. For a forecast with pdf/pmf $p(\cdot)$, (2.5) evaluates the probability of the observed value x^*.

$$\text{LogS}(p, x^*) = -\log p(x^*) \tag{2.5}$$

The goal for a forecaster is to minimize the LogS, so a forecast $p^{'}(x^*)$ is considered superior to $p(x^*)$ if $\text{LogS}(p^{'}, x^*) < \text{LogS}(p, x^*)$. The LogS is limited to scoring forecasts with density functions

and evaluating those densities only at the point x^*. The CRPS is a function of a CDF F and so it may be used more extensively than the LogS. For the CDF F, the CRPS is defined in (2.6). Here too a smaller score indicates a more accurate forecast.

$$\text{CRPS}(F, x^*) = \int_{-\infty}^{\infty} (F(x) - 1_{\{x* \leq x\}})^2 dx \tag{2.6}$$

Besides evaluating forecasts, the CRPS may also be used for optimizing weights used to build ensembles under model averaging (MA) as elaborated below. Considered the state-of-the-art techniques for combining component distributions into an ensemble distribution are nonhomogeneous regression and ensemble MA, both of which are defined by Gneiting and Katzfuss (Gneiting and Katzfuss, 2014).

In the context of an ensemble made from component models submitted from various sources, MA may be preferable because it does not require that modeling methods for individual components be the same. In MA, the final model does not have to be specified beforehand and the resulting forecast will be a mixture distribution of all component forecasts. The general form for an MA ensemble distribution p^E is defined (2.7), where p_m is the m^{th} component forecast distribution and $0 \leq w_m \leq 1$ is a weight assigned to that component where $\sum w_m = 1$. Methods for estimating weights include maximum likelihood estimation (Raftery et al., 2005), MCMC sampling (Vrugt et al., 2008), and minimizing the CRPS of the ensemble (Baran and Lerch, 2018).

$$p^E(x) = \sum_{m=1}^{M} w_m p_m(x) \tag{2.7}$$

For selecting distribution weights, minimizing the CRPS has some nice properties, but it can also be difficult to compute. For example, when the forecast is a mixture of a truncated normal distribution (TN) and a truncated lognormal (TL), the CRPS is not available in closed form (Baran and Lerch, 2018).

Generally computation and evaluation of parametric distribution functions is not hard. For most commonly used parametric distributions –normal, lognormal, Poisson, gamma, etc.– there is software readily available to compute density, distribution, and quantile values. A completely

defined continuous parametric distribution may be evaluated at a continuously infinite number of values which we call an *infinite resolution*. Also, requirements for storage are low compared to other representation types that will be discussed since the most common parametric distributions can be fully defined with three or four pieces of data including the distribution family and the corresponding parameters. Table 2.1 contains enough information to completely define a Lognormal(1,0.4) truncated on $[0, 8]$ distribution. The truncation is done here so as to make a direct comparison with the distributions shown later in Tables 2.2 and 2.3.

family	param1	param2	lowerlim	upperlim
lnorm	1	0.4	0	8

Table 2.1: This is an example of data required for a lognormal distribution with parameters $\mu = 1$ and $\sigma = 0.4$

A drawback of representing a forecast in a parametric distribution is the lack of flexibility in the model selection. Easy computation and evaluation of these models is limited to what is available in software, so certain distributional shapes may unattainable. Requiring a parametric forecast also bars the use of some statistical methods which might be used to create a forecast model including some Bayesian methods where a posterior distribution cannot be computed in closed form.

2.2.2 Sample distributions

Forecasters may want more flexibility in modeling than a parametric distribution can provide. Methods that require sampling from a posterior distribution or bootstrap sampling to obtain a sample distribution are examples where parametric distributions may not be appropriate for modeling because of the lack of flexibility in distribution shape.

A sample distribution is made of a sample of random variables $(X_1, ..., X_n)$ where $X_i \sim D$ and D is some distribution. From this sample, statistics such as mean, median, variance, and quantiles may be calculated. An empirical cumulative distribution function (ECDF) may also be calculated as in (2.8).

$$\text{ECDF} = F_n(x) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(X_i \leq x) \tag{2.8}$$

According to the Strong Law of Large numbers, the sample mean will converge to the expected value of the distribution as $n \to \infty$ as long as the expected value of that distribution exists. Likewise by the same law, the ECDF will converge to the true CDF as $n \to \infty$. Thus, if a sufficiently large sample is generated from a distribution for which an expectation exists, the sample will closely approximate the true distribution.

For common distribution families it is easy to generate large samples using existing functions in `R` and other programming platforms. For some distributions for which the mathematical formula is unknown or is not in closed form, more sophisticated methods may be required to generate samples. Bayesian analyses may require a Gibbs sampling or a Metropolis-Hastings algorithm, among other sampling methods, to generate a sample. Such samples are useful in that the true distribution may be closely approximated without knowing the true mathematical form.

Under the sample distribution representation, the options that a researcher has for constructing a forecast are more than if they are asked to submit a parametric distribution, and the range of possible shapes for a distribution is larger. In the last few decades, increased computing power and improvements in MCMC sampling have greatly contributed to growth in the use of sample distributions for forecasting (Gneiting and Raftery, 2007) (Krueger et al., 2016, see examples listed therein).

To properly score a forecast represented by a sample distribution, both the CRPS and LogS may be used. The CRPS has the advantage here of scoring the sample distribution directly since the CDF in (2.6) may be replaced with the ECDF in (2.8). To use the LogS to score a forecast, a density function for the sample may be approximated. Common approximations include a kernel density (KD) or Gaussian approximation (GA) (Krueger et al., 2016, for example).

The KD in (2.9) is defined by Krueger et. al where K is a kernel function, and h_n is a suitable bandwidth. The GA is defined in (2.10) where Φ is the standard normal CDF and $\hat{\mu}_n$ and $\hat{\sigma}_n$ are

the empirical mean and standard deviation of the sample (X_i) (Krueger et al., 2016, see also for a comparison of scoring MCMC drawn forecasts between the CRPS and the LogS).

$$\hat{p}_n(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right) \tag{2.9}$$

$$\hat{F}_n(x) = \Phi\left(\frac{x - \hat{\mu}_n}{\hat{\sigma}_n}\right) \tag{2.10}$$

To build an ensemble model from sample distribution forecasts, the MA construction from (2.7) may be used only replacing p_m with the approximate KD or GA pdf functions -$\hat{p}^{KD}_{n_m}$ or $\hat{p}^{GA}_{n_m}$ respectively. The optimal weights w_m may be estimated by maximizing the likelihood or minimizing the CRPS. If the desire is that the ensemble still be a sample distribution, after selecting weights, a sample may be selected by randomly selecting a sample from $(X_n)_m$ with probability w_m.

A potentially large issue with using sample distributions is the amount of storage it may require. For example when making MCMC draws from a posterior distribution, the final sample distribution can have a sample size of thousands or tens of thousands. Maybe not all distributions would require such a large sample size, but sizes of at least dozens or hundreds would be required for each forecast prediction. For any project the size of the CDC flu competition or the COVID-19 Forecast Hub, the storage required would be large and potentially expensive.

2.2.3 Bin distributions

An alternative to parametric distributions and sample distributions, which allows for higher flexibility in distribution shape than a parametric distribution and will usually require less storage space than samples is the bin distribution. A bin distribution may be constructed over a set $A = [a, b)$ by partitioning A into a set of K bins $\{B_i\}_{i=1}^{K}$ where $B_i = [b_{i-1}, b_i)$ and $\cup_{i=1}^{K} B_i = A$. Based on the problem to be forecast, researchers will determine the possible range A and select the number of bins and the sizes for each bin. It may be the case that a collaborative project will set the widths of all bins to be equal so that $\Delta = b_i - b_{i-1}$ is the same width for all i (McGowan

et al., 2019). To complete the construction, a probability p_i is assigned to each B_i where $\sum p_i = 1$. These probabilities are determined by the forecasters. This representation with a given bin and assigned probability may be treated like a discrete distribution with a pmf in that the calculation of the cumulative distribution is done similarly to that of a discrete parametric distribution. The cumulative distribution may be calculated as in (2.11). Here p_i is the probability for the bin B_i where $x \in B_i$.

$$P(X \leq x) = \sum_{i=1}^{n:x \in B_n} p_i \tag{2.11}$$

If the value to be forecast takes on discrete values, a common discrete distribution, such as a Bernoulli or Poisson distribution, may sometimes be used to assign probabilities to each of the bins. When the values to be forecast are continuous, a forecaster may need to employ a method of discretization to a forecast distribution. There are a number of possible ways to do this including those outlined by Chakraborty and Subrata (Chakraborty, 2015).

For several flu seasons now, the CDC flu competition has used a bin distribution as the forecast representation. The CDC has also used it for other disease outbreak forecast projects. In that context it has become the standard representation (Brooks et al., 2020). Much work has been done in evaluating and constructing ensemble models on influenza forecasts represented by discretized bins (McGowan et al., 2019; McAndrew and Reich, 2019; Reich et al., 2019a).

Because a bin distribution can be viewed as a pmf, methods for proper scoring already discussed –LogS and CRPS– are useable and MA is a valid method for ensemble construction. Reich et. al used MA to combine multiple forecasts from the flu competition. They constructed and compared ensemble models with different weighting schemes including equally weighted components, $w_m = 1/M$, and estimating weights according to the model specification. To estimate weights they used the expectation maximization (EM) algorithm (Reich et al., 2019b, see supplementary material within for details).

The exact amount of information required for a bin forecast will vary depending on the permitted range of the forecast and the desired resolution. In the CDC flu contest, a forecast

might have 131 bins between 0% and 13% –bins having increments of 0.1 or 0.01%– with corresponding probabilities in each. This makes 262 pieces of information per prediction. For any binning scheme of more than two or three bins, the information requirement for a bin distribution will be higher than for a parametric distribution. Table 2.2 illustrates what the bin distribution discretized from a Lognormal(1,0.4) distribution truncated over $[0, 8]$ looks like in 41 equally spaced bins. The discretization was done such that the probabilities p_i are calculated as in (2.12) where p^{TL} is the pdf of a truncated Lognormal(1,0.4). This is similar to Methodology-IV from Chakraborty (Chakraborty, 2015; Kemp, 2004). The truncation here is done because in practice a bin forecast will generally have a finite support.

$$p_i = \int_{b_{i-1}}^{b_i} p^{TL}(x)dx \tag{2.12}$$

Submitted as a forecast prediction, the distribution illustrated in Table 2.2 includes 82 values. For parts of the CDC influenza competition some forecasts included up to 262. This is far less storage than the possible thousands of draws from a sample distribution but is still much larger than the three or five pieces of information required to report a lognormal or truncated lognormal distribution.

bin	prob
...	...
[1.4, 1.6)	0.04414
[1.6, 1.8)	0.05896
[1.8, 2.0)	0.07032
[2.0, 2.2)	0.07172
[2.2, 2.4)	0.07955
...	...

Table 2.2: This is a storage example of a discretized lognormal with $\mu = 1$ and $\sigma = 0.4$ truncated over $[0, 8]$.

Besides the potentially large amount of information required per forecast, creating the right binning scheme may be a challenge. Because there must be a finite number of bins, forecast distributions often have finite support. And where the range of possible outcomes to a problem is

not well known, the right binning scheme may be hard to produce. This may depend on the details of the event to be forecast, but in the case of the COVID-19 outbreak, choosing the right set of bins posed a few problems.

2.2.4 Quantiles

When deciding how forecasts should be represented in the COVID-19 Forecast Hub, the time pressure of generating forecasts and the large range for possible outcomes both contributed to the COVID-19 Forecast Hub decision to forego trying to create the right binning scheme and use quantile forecasts to forecast the COVID-19 pandemic (Bracher et al., 2021). The COVID-19 Forecast Hub requires predictions to be submitted as 11 or three nominal intervals –depending on the specific target and unit to be forecast– and a median. Using this quantile representation prevents the use of certain methods for scoring and constructing ensemble forecasts.

A quantile forecast is constructed as in (2.13). Here for N given quantile levels $\alpha_1, ..., \alpha_N$; $q_1, ..., q_N$ are the values such that we have (2.13). When the quantiles are reported as prediction intervals we have (2.14).

$$P(Y \leq q_1) = \alpha_1, P(Y \leq q_2) = \alpha_2, ..., P(Y \leq q_N) = \alpha_N \tag{2.13}$$

$$P(Y \leq q_1) = \alpha_1, P(Y \leq q_2) = \alpha_2, ..., P(Y \leq q_{N-1}) = 1 - \alpha_2, P(Y \leq q_N) = 1 - \alpha_1 \tag{2.14}$$

Besides being limited scoring quantile forecasts and constructing ensemble forecasts, the shape of a distribution is also not known, and in fact nothing is known about the tails or the uncertainty beyond the most extreme reported quantile values. In the COVID-19 Forecast Hub forecasts, nothing is reported about the range below the 1^{st} quantile or above the 99^{th}. Yet the quantile representation has its advantages. Quantile forecasts allow for forecasters to submit fairly detailed forecasts without restricting the range of possible values. Since quantiles are easily calculated from any regular distribution type –using the quantile function for parametric

functions or calculating sample quantiles– we consider quantile forecasts to be highly flexible in terms of what methods forecasters may employ in modeling.

To score a quantile forecast, neither the LogS nor the CRPS may be used, but another proper scoring rule the IS may be used. For an observed outcome x^* and a prediction interval (l, r) where l and r are the $\alpha/2$ and $(1-\alpha/2)$ quantiles that bound the central $(1-\alpha)$ prediction interval, the IS is defined as in (2.15). This is a sum –weighted by α– of the width of the interval and the distance between x^* and the interval (if x^* is not captured in the interval) (Gneiting and Katzfuss, 2014). The IS requires only a single central $(1-\alpha)\times 100$ prediction interval.

$$IS_\alpha(l, r; x^*) = (r-l) + \frac{2}{\alpha}(l-x^*)1\{x^* < l\} + \frac{2}{\alpha}(x^*-r)1\{x^* > r\} \tag{2.15}$$

When a quantile forecast is made up of multiple intervals each with different α levels, the weighted interval score (WIS) may be used. Bracher et. al use the WIS to score COVID-19 quantile forecasts (Bracher et al., 2021). There are multiple versions of the WIS, some of which are described in Bracher et. al, but the version used by the COVID-19 Forecast Hub for a forecast of K intervals is defined in (2.16). Here *median* refers to the predictive median and $w_k = \alpha_k/2$ is the weight on the k^{th} interval. With that selection of weights, it may be shown that the WIS approximates the CRPS (Bracher et al., 2021, see S1 Text therein).

$$WIS_{0,K}(F_m, x^*) = \frac{1}{K+1/2}\left(w_0|x^* - median| + \sum_{k=1}^{K}\{w_k IS_{\alpha_k}(F_m, x^*)\}\right) \tag{2.16}$$

Bogner, Liechti, and Zappa compared scoring forecasts of quantiles with the Quantile Score (QS) similar to the interval score and scoring distribution functions fit to those quantiles using the CRPS (Bogner et al., 2017). The CRPS corresponds to the integral of the QS over all possible thresholds rather than just specific quantiles, so it more effectively reveals deficiencies in parts of the distribution and especially in the tails past the end points of quantiles used in QS or IS. Thus there may be something lost in terms of scoring when the WIS is used since it also is constructed from the IS.

Like the CRPS, not only does the WIS provide an easily interpretable proper score for interval forecasts, but it may also be useful when building an ensemble forecast. The ensemble

forecast constructed by the COVID-19 Forecast Hub is made as an equally-weighted average of forecasts from the component models. More specifically, each quantile value of the ensemble is the average of values from all models corresponding to the same quantile (Ray et al., 2020). For M models each with K quantiles, the k^{th} ensemble quantile q_k^E is calculated as in (2.17) where w_m is the weight assigned to each forecast and $\sum w_m = 1$. In the COVID-19 Forecast Hub model, $w_m = w = 1/M$. Where the overall mean or a weighted mean may be used for averaging, the median may also be used. Brooks et. al compare performance of the COVID-19 ensemble using equally-weighted means, weighted means, and median value constructions (Brooks et al., 2020). In their report, they show that weighted means and median constructions tend to outperform an equally-weighted mean construction. To come up with optimal weights, they select values w_m from (2.17) which minimize the WIS of the ensemble forecast.

$$q_k^E = \sum_{m=1}^{M} w_m q_k^m \tag{2.17}$$

More generally, quantile averaging or Vincentization for a complete distribution is defined as in (2.18) where $F_m^{-1}(\alpha) = \inf\{y : F_m(y) \geq \alpha\}$ for $0 < \alpha \leq 1$. Some notable characteristics are that it is more likely for the ensemble distribution to be unimodal than it is under linear averaging of densities like MA (Busetti, 2017). Under some circumstances, such as when member distributions are from an exponential, Weibull, or logistic family the aggregated distribution is of the same family (Ratcliff, 1979). Quantile averaging produces smoother distributions than MA according to Schepen and Wang (Schepen and Wang, 2015). And Lichtendahl et. al conclude that quantile averaging produces sharper forecasts and tends to perform better in scoring than probability averaging like MA (Lichtendahl Jr et al., 2013).

$$F_v^{-1}(\alpha) = \sum_{m=1}^{M} w_m F_m^{-1}(\alpha) \tag{2.18}$$

As in sample distributions and bin distributions, data storage for interval forecasts will depend on the desired clarity of resolution. For the COVID-19 forecasts submitted to the COVID-19 Forecast Hub, 23 quantile values are requested for quantiles (0.01, 0.025, 0.05, 0.10,

..., 0.95, 0.975, 0.99). This includes a median along with 11 predictive intervals (Bracher et al., 2021). Forecasters are thus required to submit 46 values in each short-term forecast (some of the longer term forecasts only include seven quantiles). In terms of storage, this is an improvement over requirements for the CDC flu competition. Table 2.3 shows how a submission of 23 quantiles from a Lognormal(1,0.4) truncated on $[0, 8]$ might look.

quantile	0.01	0.025	0.05	...	0.95	0.975	0.99
value	1.07137	1.2404	1.40689	...	5.18328	5.82391	6.58783

Table 2.3: This shows six quantiles and values from a lognormal distribution with $\mu = 1$ and $\sigma = 0.4$

Figure 2.1 illustrates how the densities and CDFs compare between parametric distributions, sample distributions, bin distributions, and quantiles.

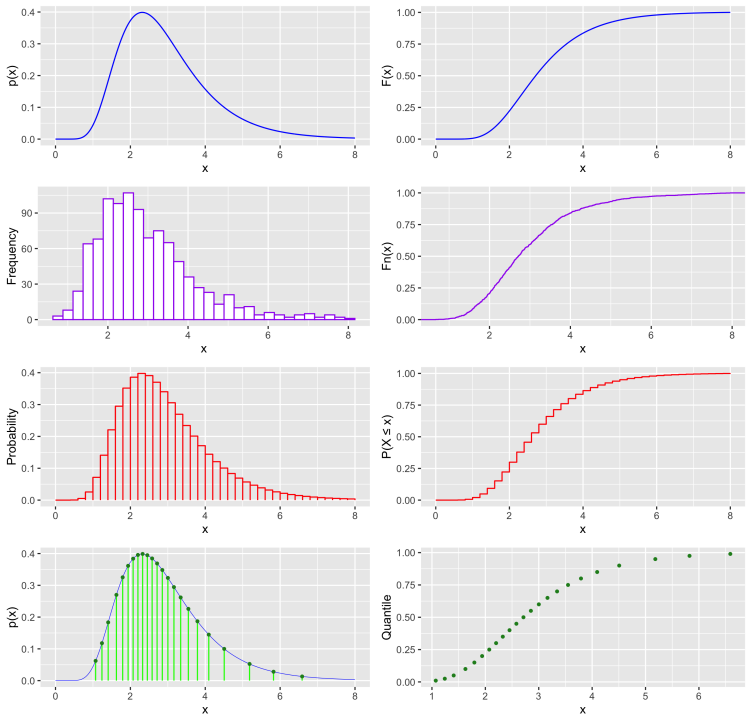


Figure 2.1: This figure compares the densities and CDFs of forecast representation types discussed in the left and right columns respectively. Each is generated from a Lognorma(1,0.4) distribution truncated on $[0, 8]$. Blue shows the density and CDF functions. Purple shows a histogram and ECDF of 1,000 samples. Red shows bin probabilities and the CDF function for a bin distribution. Green shows quantiles with corresponding values.

2.3 Mixture distributions

A mixture distribution forecast representation is an attractive alternative to the four representations already discussed. A mixture distribution forecast would allow for a large range of distribution shapes, a high resolution, storage comparable to that of bin and quantile forecasts, and ensemble construction using MA. A mixture distribution may be constructed in the same way as the ensemble described in section 2.2.1 (2.7) where for C distributions with pdfs $p_c(x)$ and $w_c > 0$ and $\sum w_c = 1$ we have (2.19).

$$p^M(x) = \sum_{c=1}^{C} w_c p_c(x) \tag{2.19}$$

Like a parametric distribution, a mixture distribution may be evaluated using existing software like the `distr` package in `R` (Camphausen et al., 2007). And scoring may be done using the LogS, CRPS, and IS. A mixture distribution, like its parametric distribution components, has an infinite resolution. A mixture distribution may be more flexible than a single component parametric distribution in terms of distribution shape. According to McLachlan and Peel, a mixture of normal densities with common variance may be used to approximate arbitrarily well any continuous distribution (Peel and MacLahlan, 2000) (see also (Nguyen and McLachlan, 2019)). Thus, for an unconventional probability distribution –such as an MCMC posterior sample– it may be reasonable to approximate the distribution by fitting those samples to a mixture of normal distributions. Depending on the number of components a forecaster includes in a mixture forecast, the amount of storage per forecast might be as little as for a parametric forecast or as much as is permitted in the specific collaborative forecast project.

An ensemble model may be constructed by using (2.7) only replacing p_m with p_m^M from (2.19). Solving for weights may also be done by maximizing the likelihood of the forecast or minimizing the CRPS. However with the added complexity of component models being mixture distributions the computation is likely to be more expensive. An example where this is true is when minimizing the CRPS when the exact mixture distribution does not produce a closed form CRPS (Baran and Lerch, 2018). In large projects like the COVID-19 Forecast Hub, if an equal weight is not

assigned to each component, it may be determined that models not reaching a certain standard of predictive performance are assigned an ensemble weight of 0. This would simplify an ensemble model to include only the best performing forecasts.

Table 2.4 shows how a mixture distribution forecast compares with the other formats discussed in terms of methods for scoring, information and resolution provided, methods for ensemble building, and computer storage requirement. To summarize, a continuous mixture distribution has the infinite resolution of a parametric distribution with the flexibility of a bin distribution, a sample distribution, and a set of quantiles. The common proper scoring rules LogS and CRPS may be used to score a mixture forecast. The storage requirement is comparable to that of a bin distribution or a set of quantiles. And MA may be used for building an ensemble. In Section 3 we show how a mixture distribution may be constructed, scored, and used to construct an ensemble using software available in `R`.

Representation	**Scoring**	**Flexibility/Information**	**Ensemble**	**Storage Requirement**
Parametric distribution	LogS, CRPS, IS	Limited to common distribution families. Infinite resolution	MA	Low 3-6 values per prediction
Sample distribution	CRPS and IS. LogS after smoothing	Any shape. Resolution may be very high but depends on sample size	MA after smoothing. Resampling otherwise	Hundreds or thousands of values per prediction
Bin distribution	LogS, CRPS, IS	Any shape allowed, but limited by binning scheme. Range also may be limited	MA	Depends on binning scheme but dozens to hundreds of values
Quantiles	IS, WIS	Shape unknown but with enough quantiles there is still a decent amount of information. No tail information	Quantile averaging	Depends on quantiles requested, but dozens of values
Mixture distribution	LogS, CRPS, IS. May be more limited by computation	With a sufficient number of component distributions, it may approximate any distribution shape. Infinite resolution	MA	3 values to dozens per forecast depending on number of components

Table 2.4: This table compares scoring, information, ensemble building, and storage requirements for the different forecast representations discussed. To summarize a continuous mixture distribution has the infinite resolution of a parametric distribution with the flexibility of a bin distribution, a sample distribution, and a set of quantiles. The common proper scoring rules LogS and CRPS may be used to score a mixture forecast. The storage requirement is comparable to that of a bin distribution or a set of quantiles. And MA may be used for building an ensemble.

3. Mixture distributions in a collaborative forecast project

The CDC flu competition and the COVID-19 Forecast Hub as well as other collaborative projects have their own established systems for receiving, scoring, and constructing ensemble forecasts. A transition from using bin or quantile forecasts to using mixture distribution forecasts would require a few adjustments to those systems. In this section we outline how some of these adjustments may be implemented. We also present tools which may be used to build forecasts from submissions, score those forecasts, and construct ensembles from them.

3.1 Submission format

For a collaborative forecast project to run smoothly, forecast submissions from all forecasters should follow the same format. For both the CDC flu competition and the COVID-19 Forecast Hub, teams provide a `.csv` spreadsheet which contains the distributional information for one or multiple forecasts. Tables 3.1 and 3.2 show what variables are included in those submissions and a couple rows to illustrate possible values. The column variables include `location`, `target`, `type`, `unit`, `bin` or `quantile`, and `value`. Here `location` defines the specific county, state, or country of the forecast. The `target` variable defines what is forecast with levels: season onset, deaths, hospitalizations, etc. The `type` variable defines the type for the `value` variable with levels of point, bin, or quantile. The `unit` variable defines the time frame of the forecast with levels of one week, two weeks, four weeks, etc. The variables `bin` and `quantile` give a specific bin or a specific quantile. The `value` variable is a number that either gives the probability associated with a bin or the value associated with a quantile.

A single submission may include many forecasts aimed at forecasting different combinations of `location`, `target`, and `unit`. A set of rows which share the same specific combination of `location`, `target`, and `unit` constitute a single forecast. One forecast for the CDC flu

competition may require up to 131 rows whereas in the COVID-19 Forecast Hub one forecast may require up to 23 rows.

location	target	type	unit	bin	value
us national	season onset	bin	week	0.0	.
us national	season onset	bin	week	0.1	.
...	...	...	...	...	...

Table 3.1: This table shows a few rows of a submission file for a bin forecast like those in the CDC flu competition. A set of rows which share the same combination of `location`, `target`, and `unit` make up a single forecast. One submission may include many forecasts specified by differing combinations of those three columns.

location	target	type	unit	quantile	value
us national	season onset	quantile	week	0.01	.
us national	season onset	quantile	week	0.025	.
...	...	...	...	...	...

Table 3.2: This shows a few rows of a submission file for a quantile forecast like those in the COVID-19 Forecast Hub. A set of rows which share the same combination of `location`, `target`, and `unit` make up a single forecast. One submission may include many forecasts specified by differing combinations of those three columns.

Table 3.3 illustrates adjustments made to the submission formats from Tables 3.1 and 3.2 which make a usable submission format for mixture distribution forecasts. In such a format, each row represents one component distribution used in a mixture distribution. The variables `bin` or `quantile` and `value` are removed and replaced with `family`, `param1`, `param2`, and `weight` where `family` is the distribution family of the component, `param1` and `param2` are the parameters for the component distribution, and `weight` is the weight w_i for the i^{th} component.

For reasons of storage and computation, a forecast project may have a limit to the number of components allowed per forecast. For reference, a mixture distribution forecast following the format in table 3.1 with 17 components would require $17 \times 8 = 136$ pieces of information submitted per forecast. A submission to the COVID-19 Forecast Hub forecast with 23 quantiles according to the format in Table 3.1 requires $23 \times 6 = 138$ cells. Thus if the COVID-19 Forecast

location	target	type	unit	family	param1	param2	weight
us national	season onset	dist	week	norm	a_n	b_n	w_1
us national	season onset	dist	week	lnorm	a_l	b_l	w_2
...	...	...	...	...	...	...	...

Table 3.3: This is an example of a submission file for a disease outbreak forecast using a mixture distribution representation. Each row represents a component distribution of a mixture distribution. The variables location, target, and unit specify what is being forecast. The variable type specifies that the row represents a parametric distribution. The variables family, param1, and param2 specify the exact component distribution. And the variable weight specifies the weight w_i that the i^{th} component distribution is assigned in the mixture distribution. Here two components are shown with distributions Normal(a_n, b_n), Lognormal(a_l, b_l) and weights w_1 and w_2.

Hub were to change the forecast representation from quantile forecasts to mixture forecasts but continue allowing the same amount of data per forecast, a mixture distribution with 17 components could be used in one forecast. That many components could allow for a large range of distribution shapes and flexible forecasts.

In the remainder of this section, explanations of how to work with mixture distributions submitted according to Table 3.3 are given. Also given is `R` code which demonstrates constructing a mixture distribution from a forecast submission, scoring the forecast, and building an ensemble from two separate submissions.

3.2 Mixture construction and scoring tools

A single `.csv` submission file of the format in Table 3.3 may contain multiple forecasts forecasting different combinations of location, target, and unit. Selecting only rows which share a specific combination of location, target, and unit will produce a table representing a single forecast. That table may look like Tables 3.4 and 3.5. If the table is saved as a standard `data.frame` in `R`, then tools based on the `distr` package (Camphausen et al., 2007) may be used for evaluating a mixture distribution with the component distributions in the table.

The `distr` package contains a function `UnivarMixingDistribution()` which takes as arguments a list of distributions and a vector of weights for each distribution and an object of class `AbscontDistribution` is returned. An `AbscontDistribution` class is a mother class which defines a random number generator, pdf, CDF, and quantile function for continuous distributions

from common families contained in the `distr` package and for mixture distributions with component distributions from those families. We wrote a function `MakeDist()` (see APPENDIX) which takes on a `data.frame` with variables family, param1, param2, param3, and weight and where each row represents a component distribution in a mixture distribution. The function `MakeDist()` calls on the `UnivarMixingDistribution()` function and returns a mixture distribution object of class `AbscontDistribution`.

If a forecast such as in Table 3.4 is taken as an argument in `MakeDist()`, the resulting mixture distribution may then be evaluated with functions for the pdf, CDF, quantile function, and random samples from the mixture distribution may be drawn. The distribution may then be scored using the LogS or CRPS.

family	param1	param2	param3	weight
Lnorm	2	1	NA	0.3
Norm	2.1	1	NA	0.7

Table 3.4: This is an illustrative example of a mixture distribution forecast where the distribution is described in a data frame. The first component is a Lognormal$(2, 1)$ with a weight of 0.3 in the mixture and the second component is a Normal$(2.1, 1)$ with a weight of 0.7. The distribution family abbreviations are capitalized here because that is how they will be requested in the `MakeDist()` function. Refer to Table 1 in the APPENDIX for more details.

family	param1	param2	param3	weight
Norm	1.5	1	NA	0.4
Norm	4	2	NA	0.6

Table 3.5: This is a second illustrative example of a mixture distribution forecast where the distribution is described in a data frame. The first component is a Normal$(1.5, 1)$ with a weight of 0.4 in the mixture and the second component is a Normal$(4, 2)$ with a weight of 0.6. The distribution family abbreviations are capitalized here because that is how they will be requested in the `MakeDist()` function. Refer to Table 1 in the APPENDIX for more details.

Here we include code to illustrate the process of constructing and scoring two separate forecasts. We suppose that Table 3.4 represents a submitted forecast from one forecaster and Table 3.5 represents a forecast of the same event from a second forecaster. Table 3.1 shows plots of the pdfs for both mixture forecasts. Note the additional `param3` variable in Tables 3.4 and 3.4. This variable is included in the table because of the functionality of the `MakeDist()` function

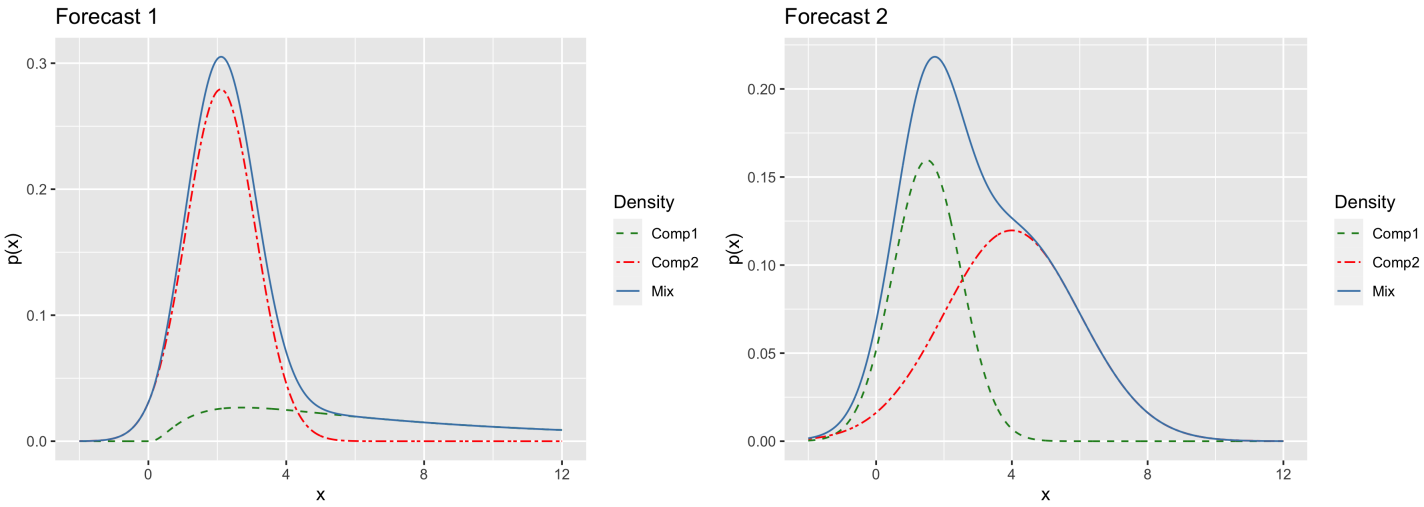


Figure 3.1: These plots show the density functions of the two example mixture distribution forecasts along with the component density functions scaled by the corresponding weights

which allows for component distributions of up to three parameters. The code here shows these two forecasts as `data.frame`s and how the `MakeDist()` function is used to create the distributions in `R`. Once the distributions are created as `AbscontDistribution` objects, then functions for evaluating a pdf and a CDF for each are created.

```
preddf1

##   family param1 param2 param3 weights
## 1  Lnorm    2.0      1     NA     0.3
## 2   Norm    2.1      1     NA     0.7

preddf2

##   family param1 param2 param3 weights
## 1   Norm    1.5      1     NA     0.4
## 2   Norm    4.0      2     NA     0.6

#make mixture distributions from prediction submissions
```

```
mdist1 <- MakeDist(preddf1)
mdist2 <- MakeDist(preddf2)
```

```
#make pdfs for mixture predictions
dmdist1 <- function(x) {distr::d(mdist1)(x)}
dmdist2 <- function(x) {distr::d(mdist2)(x)}


#make cdfs for mixture predictions
pmdist1 <- function(x) {distr::p(mdist1)(x)}
pmdist2 <- function(x) {distr::p(mdist2)(x)}
```

The LogS or the CRPS may then be calculated for each forecast using the pdf and CDF functions respectively. Here will will assume that the true value which both forecasts attempted to predict was `3`. The `CRPS()` function here is one that we wrote and is included in the APPENDIX. It is seen in the code below that under the LogS, forecast 1 from Table 3.4 outperforms forecast 2 from Table 3.5 with scores of 1.547 and 1.849 respectively. However, under the CRPS, forecast 2 outperforms forecast 1 with scores 0.635 and 0.635 respectively. We continue to use these same forecasts in Section 3.3 used in constructing an ensemble forecast.

```
#realized observation
xstar <- 3


#LogS for predictions at the realized observation
-log(dmdist1(xstar))

## [1] 1.547238

-log(dmdist2(xstar))

## [1] 1.848796
```

```
#CRPS for predictions at the realized observation
CRPS(pmdist1,y=xstar)

## [1] 0.6348212

CRPS(pmdist2,y=xstar)

## [1] 0.5306083
```

3.3 Ensemble construction

To construct an ensemble distribution from multiple mixture distributions, the `UnivarMixingDistribution()` function may be used. The function takes two or more `AbscontDistribution` distribution objects, including mixture distribution objects, and a vector of weights corresponding to each object. A new `AbscontDistribution` object is returned as an ensemble of mixture distributions as in (2.7). Since they are `AbscontDistribution` objects, `mdist1` and `mdist2` created in the code in Section 3.2 may be input as arguments into the function `UnivarMixingDistribution()`, but weights for each object also need to be determined.

At the onset of a collaborative forecast before there are true event observations which the forecasts may be scored on, it may make sense to assign an equal weight to each component distribution in an ensemble. As a project progresses, however, assigning weights based on past performance may be desired. As mentioned in section 2.2.1, weights may be selected by maximizing the likelihood of (2.7) or by minimizing the CRPS. Another method of selecting weights is to use the posterior model probability.

If we have T models (M_t) the posterior model probability of M_t is defined as in (3.1) where $p(\cdot|M_t) := p_t(\cdot)$ is the pdf of the model distribution and $p(M_t)$ is the prior probability assigned to the model. A common approach is to assume the prior probabilities for each model are equal or $p(M_t) = 1/T$ for all t in which case (3.1) is reduced to (3.2). In this case the posterior model probability for the t^{th} model is equal to the exponential of its negative LogS or

$p(M_t|x) = e^{-\text{LogS}(p(M_t|x))}$, so the performance of a forecast based on the LogS is directly related to its posterior model probability and may be used as an ensemble weight. For an observed event x^*, ensemble weights (w_t) from (2.7) may be defined as $w_t := p(M_t|x^*)$.

$$p(M_t|x) = \frac{p(x|M_t)p(M_t)}{p(x)} = \frac{p(x|M_t)p(M_t)}{\sum_{k=1}^{T} p(x|M_k)p(M_k)} \tag{3.1}$$

$$p(M_t|x) = \frac{p(x|M_t)}{\sum_{k=1}^{T} p(x|M_k)} \tag{3.2}$$

Using the illustrative example from Section 3.2, the following code shows how to use the posterior model probability to select weights, construct an ensemble distribution, and score the ensemble forecast. Here again we take the true event value to be 3. The ensemble distribution along with component distributions is shown in Figure 3.2.

```
#posterior model probability for calculating weights
w1 <- pmdist1(xstar)/(pmdist1(xstar) + pmdist2(xstar))
w2 <- 1-w1
w1

## [1] 0.5286434

w2

## [1] 0.4713566

#build ensemble with calculated weights
ensdist <- distr::UnivarMixingDistribution(mdist1,
                                           mdist2,
                                           mixCoeff = c(w1,w2))


#pdf and cdf for ensemble
```

```r
densdist <- function(x) {(distr::d(ensdist)(x))}

pensdist <- function(x) {(distr::p(ensdist)(x))}


#LogS for predictions at the realized observation

-log(densdist(xstar))

## [1] 1.678156

#CRPS for predictions at the realized observation

CRPS(pensdist,y=xstar)

## [1] 0.5486368
```

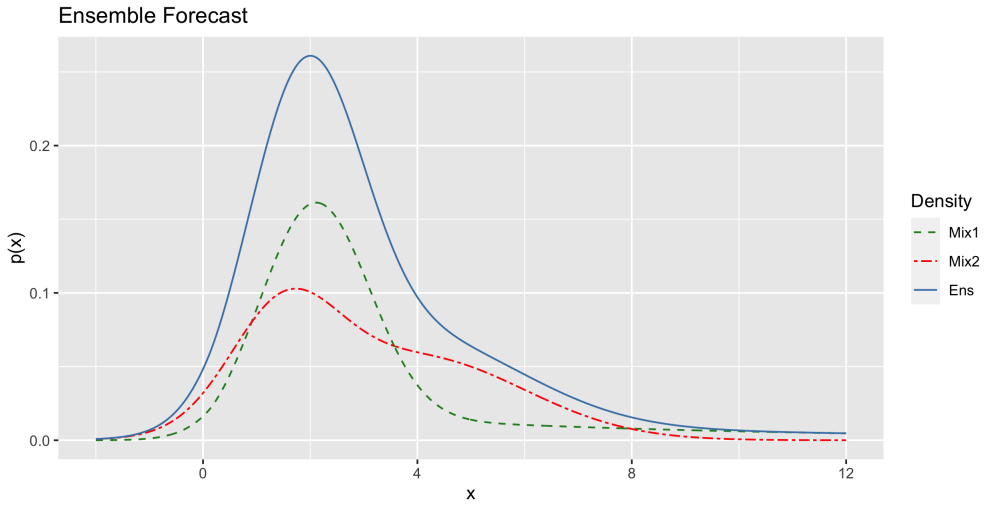


Figure 3.2: Ensemble forecast made from forecast 1 and forecast 2 from Section 3.2. The green line is the density component of mixture forecast 1 with weight 0.529. The red line is the density component of mixture forecast 2 with weight 0.471. The blue line is the density of the ensemble forecast.

4. Retrospective analysis

For large collaborative forecast projects having already established the representation formats for forecasting, it may be difficult for individual forecast teams to adjust to a mixture distribution format. There may be several reasons for this, including that not all forecast modeling methods will produce forecasts which may conveniently be represented by a mixture distribution. In this section we attempt to assess whether or not any forecasts from the CDC flu competition and the COVID-19 Forecast Hub were generated from one component mixture distribution models. We do this by fitting one component mixture distributions to the forecasts and assessing how well the fit models represent the forecasts.

Forecasters in both the CDC flu competition and the COVID-19 Forecast Hub do not include with their forecast submissions information about modeling methods or distributional assumptions. Thus the only information we have for fitting distributions are bin forecasts and quantile forecasts. We are unaware of formal statistical methods for fitting parametric distributions or mixture distributions to bin distributions. Methods of fitting a distribution to quantiles include Bayesian Quantile Matching (Nirwan and Bertschinger, 2020), step interpolation with exponential tails (Quinonero-Candela et al., 2005), and the Method of Simulated Quantiles (Dominicy and Veredas, 2013). These studies, however, lack claims that the methods for fitting are statistically formal. Nirwan and Bertshinger state that minimizing the mean square error between quantile values and a CDF function has been the most common way to fit a distribution to a set of quantiles. This is the method we will use in Section 4.2. Because of the lack of statistically formal methods for fitting a parametric distribution to a bin distribution or a set of quantiles, it should be noted that any conclusions made in this section may not be stated in terms of statistical certainty.

4.1 CDC flu competition

The CDC Retrospective Forecasts project on zoltrdata.com (Zoltar, a) contains over 869,638 probabilistic influenza-like illness forecasts for all combinations of 11 regions in the United States and seven targets from 27 different modeling teams. These include forecasts made during all flu seasons between October 2010 and December 2018. All forecasts are represented by bin distributions. We wanted to determine whether or not any of these forecasts were possibly generated from one component mixture distributions of continuous uniform, truncated normal (TN), truncated lognormal (TL), or truncated gamma (TG) distribution families. The process we followed for determining if a forecast was possibly generated by one of these common families was to first fit a distribution of that family to the bin forecast and then measure the closeness of that fit. The decision to fit truncated distributions was due to all bin forecasts being assessed having finite support.

In fitting a distribution to a forecast, we want minimize (4.1). We call (4.1) the mean square difference (MSD). Here $F(\cdot|\hat{\theta})$ is a CDF, p_i is the reported probability for the i^{th} bin $B_i := [b_{i-1}, b_i)$, and K is the number of bins. The fitted parameter vector $\hat{\theta}$ is the solution to (4.2). If a distribution was fit to the submitted bin forecast and the MSD was less than a specified cutoff value, we concluded that the bin forecast may have been generated from the distribution.

$$\text{MSD} = \frac{1}{K}\sum_{i=1}^{K}(p_i - [F(b_i|\hat{\theta}) - F(b_{i-1}|\hat{\theta})])^2 \tag{4.1}$$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{K}\sum_{i=1}^{K}(p_i - [F(b_i;\theta) - F(b_{i-1};\theta)])^2 \tag{4.2}$$

To determine what values to use as cutoffs we conducted a study where bin distributions were discretized from known distributions, parameters were fit to the bin distributions, and the MSDs were calculated. This was done 1,000 times for each of the four distribution families considered and the cutoff value for each family was set as the value under which 95% of MSD values fell. The exact process is as follows.

For each of uniform, TN, TL, and TG distributions the following was done 1,000 times. A set of 131 bins with interval widths of 0.1 between 0 and 13 was constructed. A value u was selected from a Unif(0,13) distribution and v from a Unif(.05,1.6) distribution. The drawn values u and v were taken as respectively the center and scale parameters of a TN distribution. For uniform, TL, and TG family distributions, model parameters were computed such that u and v were roughly the mean and standard deviation. With a known distribution with $\theta = (u, v)$, probabilities $p_i = F(b_i|\theta) - F(b_{i-1}|\theta)$ were calculated for each bin. A distribution from the same family was fit to the bins by minimizing (4.1) with the resulting distribution function $F(\cdot|\hat{\theta})$. The minimization was done using the `optim` function in the `R stats` package. From the fit distribution $\hat{p}_i = F(b_i|\hat{\theta}) - F(b_{i-1}|\hat{\theta})$ was computed for each of the 131 bins. Finally the MSD was calculated. Table 4.1 below shows MSD value for which 95% of all 1,000 MSDs fell below. Those values were selected as the cutoff values for declaring whether or not a bin distribution was discretized from from a one component mixture distribution from that family.

Distribution	MSD 95% Cutoff
Uniform	8.809460e-05
Truncated Normal	1.126683e-07
Truncated Lognormal	3.877829e-06
Truncated Gamma	1.500152e-06

Table 4.1: If a binned one component mixture distribution is fit to a bin forecast and the MSD is smaller than the corresponding cutoff value listed here, we consider that bin distribution to have been discretized from the fit mixture distribution.

Of the 869,638 forecasts from the CDC Retrospective Forecasts project, we fit each of uniform, TN, TL, and TG distributions to 11,715 of those forecasts. From each of the 27 teams represented, there is a list of forecast submissions from which we randomly selected 12. From each submission, we we randomly selected six forecasts. In many cases the `optim` function used to fit the distributions failed to converge, so of all forecasts we were only able to successfully fit distributions to 11,715. We calculated the MSD for a distribution fit from each of the four families, and for each forecast the fit distribution with the lowest of the four MSDs was

considered the best fit. If the MSD from the best fit fell below the corresponding cutoff value listed in 4.1 we concluded that the forecast was possibly generated from the fit distribution.

Table 4.1 shows the results of this analysis. Of the 11,715 binned distributions fit to the mixture distributions, 2,502 of those fits had MSD values below the cutoff values listed in Table 4.1. Thus we conclude that a proportion of 0.214 of those forecasts were possibly generated from a one component mixture distribution. The results for the 11,715 fit distributions, which families the bin distributions were best fit to, and whether or not the fits produce an MSD below the cutoff value, are seen in Table 4.2. Figure 4.1 is an image showing a distribution from each of the four distribution families having been fit to the same bin distribution forecast.

Distribution Family	Total	Total below MSD cutoff value	Proportion from mixture distribution
Uniform	896	669	0.747
Truncated Normal	3,804	198	0.052
Truncated Lognormal	4,501	1,340	0.289
Truncated Gamma	2,514	295	0.117
Total	**11,715**	**2,502**	**0.214**

Table 4.2: Of the 11,715 forecasts 864 were best fit by a uniform distribution, 3,804 by a TN, 4,501 by a TL, and 2,514 by a TG. Of those forecasts, 669 appear to have come from a uniform distribution, 198 from a TN, 1,340 from a TL, and 295 from a TG. Overall, 2,502 of 11,715 forecasts could have been generated from a distribution from one of the four families.

4.2 COVID-19 Forecast Hub

As of April 4, 2022 there were over 90 million forecasts from 117 different modeling teams submitted to the COVID-19 Forecast Hub (Zoltar, b). These forecasts covered all combinations of 3,202 municipalities (mostly counties) in the United States with 441 target/unit combinations. The first of these forecasts was submitted in March of 2020 shortly after the initial outbreak of the COVID-19 virus in the US, and forecasts have been received weekly since then. The forecasts are all quantile forecasts made up of three or 11 predictive intervals –depending on the specific unit for the forecast– and a median. Thus each forecast includes seven or 23 quantiles.

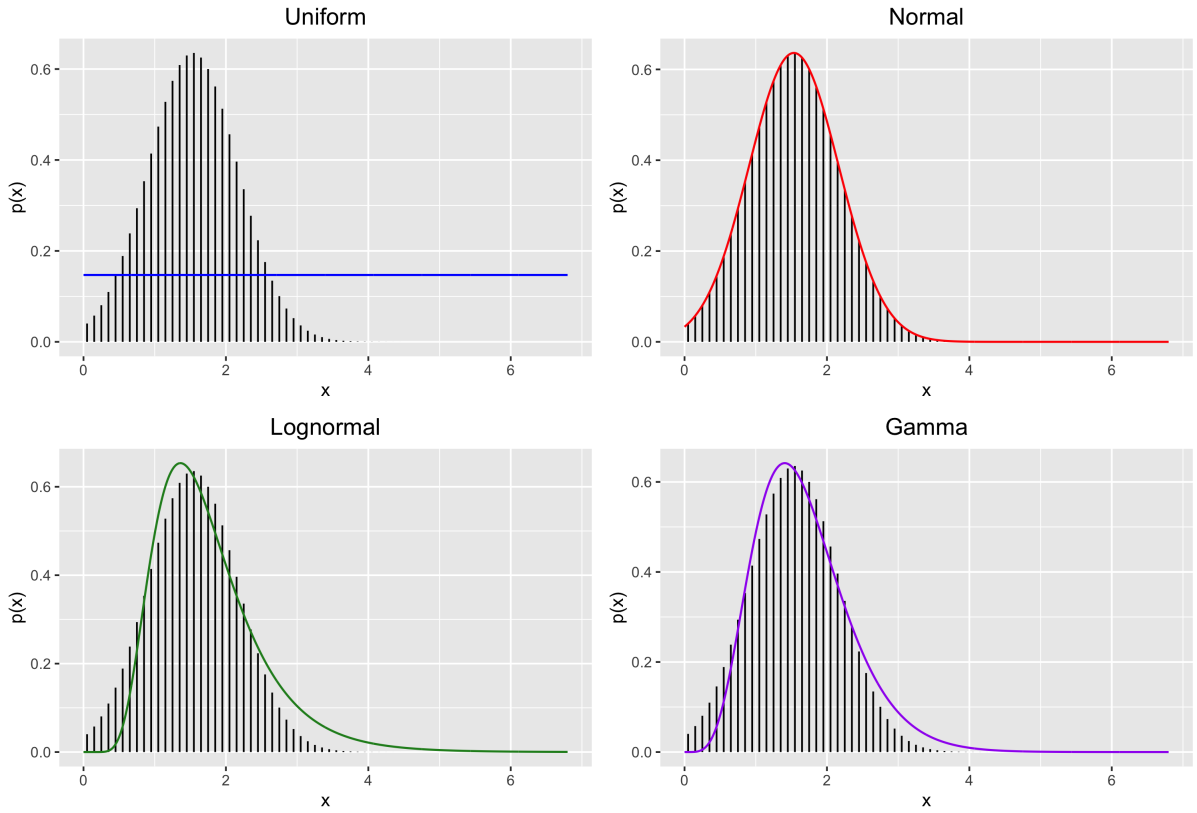


Figure 4.1: This figure shows the plots of fits from each a uniform, TN, TL, and TG distributions to the same bin distribution. Each black vertical line is a bin probability $p_i \times 10$. The probabilities are multiplied by 10 to better illustrate fitting a distribution to the probabilities. The title of this forecast given by the submission team is Bayesian Model Averaging. This forecast was submitted to the CDC on May 15, 2017 and is a three week ahead forecast of percent of hospital patients due to an influenza like illness for a specific region of the United States. In this case, the fit to the normal distribution produced an MSD below the TN cutoff value, so we conclude that the bin forecast may have been discretized from a TN. Proportions of distributions with MSDs below the cutoff values are also included.

To assess whether or not a quantile forecast was calculated from a well known continuous distribution, we minimized the mean square error (MSE) in (4.3) where α_i is the i^{th} quantile level (out of N levels) from a forecast and $F(q_i|\hat{\theta})$ is a fit CDF evaluated at the quantile value q_i with parameter $\hat{\theta}$. The parameter $\hat{\theta}$ is the solution to (4.4).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{m}(\alpha_i - F(q_i|\hat{\theta}))^2 \tag{4.3}$$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{m}(\alpha_i - F(q_i|\theta))^2 \tag{4.4}$$

For each forecast assessed, we fit the quantiles to a uniform, normal, lognormal, gamma, and location-scale t CDF. Unlike in the previous section, we felt no need to truncate the distributions because there is no range limit to these forecasts. We also included the t distribution to account for any symmetric forecasts with heavier tails than in a normal distribution. If the MSE between a given forecast and a fit CDF fell below a certain cutoff value, we considered the quantiles as approximately matching those from the fit CDF. To determine the cutoff values, for each of the five distribution families, the same procedure was followed 1,000 times.

A decision was randomly made on creating a quantile distribution with seven quantiles or 23 quantiles with probabilities 1/3 and 2/3 respectively. This was done because in the COVID-19 Forecast Hub certain target/unit combinations require seven quantiles whereas others require 23. The quantiles used matched those used by the COVID-19 Forecast Hub. After that decision was made, a random value u was drawn from a Unif(2,000, 25,000) distribution and another value v was drawn from a Unif(3, 200) distribution. These values were taken as the mean and standard deviation for each of the five families, and for each the proper transformations were computed to find model parameters corresponding to that distribution family. Quantile values were then calculated for each given quantile. When using a t distribution, a value for degrees of freedom was drawn from a Unif(2,35) distribution. A CDF was fit by minimizing (4.4) over the parameter vector θ and the MSE value from (4.3) was calculated. The MSE value for which 95% of the 1,000 simulated distributions fell below was considered the cutoff for that family and is seen in table

4.3. It is immediately noticeable that the cutoff value for the location-scale t family is several orders of magnitude higher than for the other distribution families. We believe this is related to the fact that the t distribution is a three parameter family where as the other four families are two parameter families, thus influencing the optimization methods used enough to make such a difference in the cutoff value.

Distribution	MSE 95% Cutoff
Uniform	7.220667e-07
Normal	3.097645e-05
Lognormal	1.166775e-07
Gamma	2.953370e-05
Location-scale t	0.4894471

Table 4.3: If a one component mixture distribution of a certain distribution family is fit to a quantile forecast and the MSE is smaller than the corresponding cutoff value listed here, we conclude the quantiles to have been calculated from the fit distribution.

With these cutoff values selected, distributions were then fit to quantile forecasts from the COVID-19 forecasts on zoltardata.com (Zoltar, b). From 115 modeling teams we randomly selected four forecast submissions. From those submissions we randomly selected three units and attempted to assess forecasts for all targets under than unit. This was a computationally more difficult problem than fitting distributions to the bin forecasts from the CDC flu competition, so the number of forecasts where convergence for fitting was met was much smaller. In total, distributions from all five families of interest were fit to 2,504 forecasts. The MSE was calculated for each, and the fit with the lowest MSE was selected as the best fit. If the MSE fell below the specified cutoff, we concluded that the quantile forecast was approximately from the same distribution as the fit distribution.

Table 4.4 shows the results of the analysis by distribution family. Of the 2,504 fits, 99 of them had an MSE value below the corresponding cutoff listed in Table 4.3. Thus we conclude that a proportion of 0.04 of the forecasts assessed could have come from distributions fit to them. Results for fits by distribution are seen in Table 4.4. Figures 4.2 and 4.3 show the

quantile-quantile (QQ) plots and the CDF plots for four of the different distribution families fit to the same set of quantiles.

Distribution Family	Total	Total below MSE cutoff	Proportion from a one component mixture
Uniform	239	0	0
Normal	609	74	0.122
Lognormal	694	0	0
Gamma	597	0	0
Location-scale t	365	25	0.068
Total	**2,504**	**99**	**0.04**

Table 4.4: Results for COVID-19 Forecast Hub retro analysis. Here it is seen that 239 of the forecasts were best fit by a uniform, 609 by a normal, 694 by a lognormal, 597 by a gamma, and 365 by a location-scale t. Of those fits 0, 74, 0, 0, and 25 respectively had an MSE below the cutoff values give in Table 4.3.

The results from these two studies suggest that in some cases forecasters are using methods which may produce one component mixture distribution forecasts. Many likely are not. To expand upon these studies, one could attempt to fit the bin forecasts or the quantile forecasts to mixture distributions with multiple components. Forecasts that are very close to a mixture distribution may suggest that the forecasters already have models that make transition to a mixture distribution forecast format easy or straightforward.

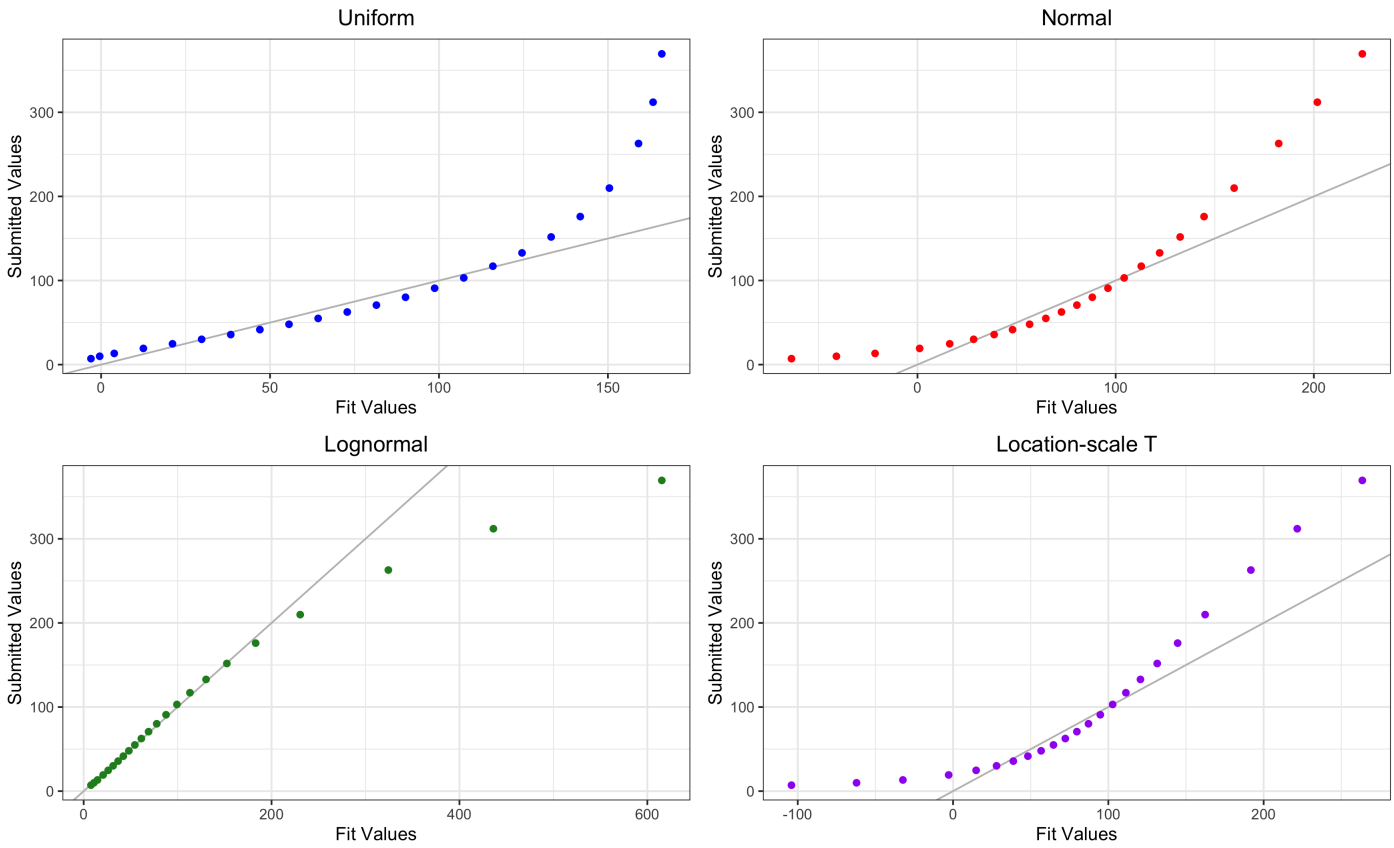


Figure 4.2: This figure shows QQ plots for a set of submitted quantiles against the quantiles the continuous distribution to which it was fit. The name of this forecast given by the team who submitted it is Gleam COVID-19. This forecast was submitted to the C0VID-19 Forecast Hub on November 1, 2021 and is forecasting deaths one in the next week in a certain US state due to the virus. In no case here was the MSE below the cutoff value of the corresponding distribution.

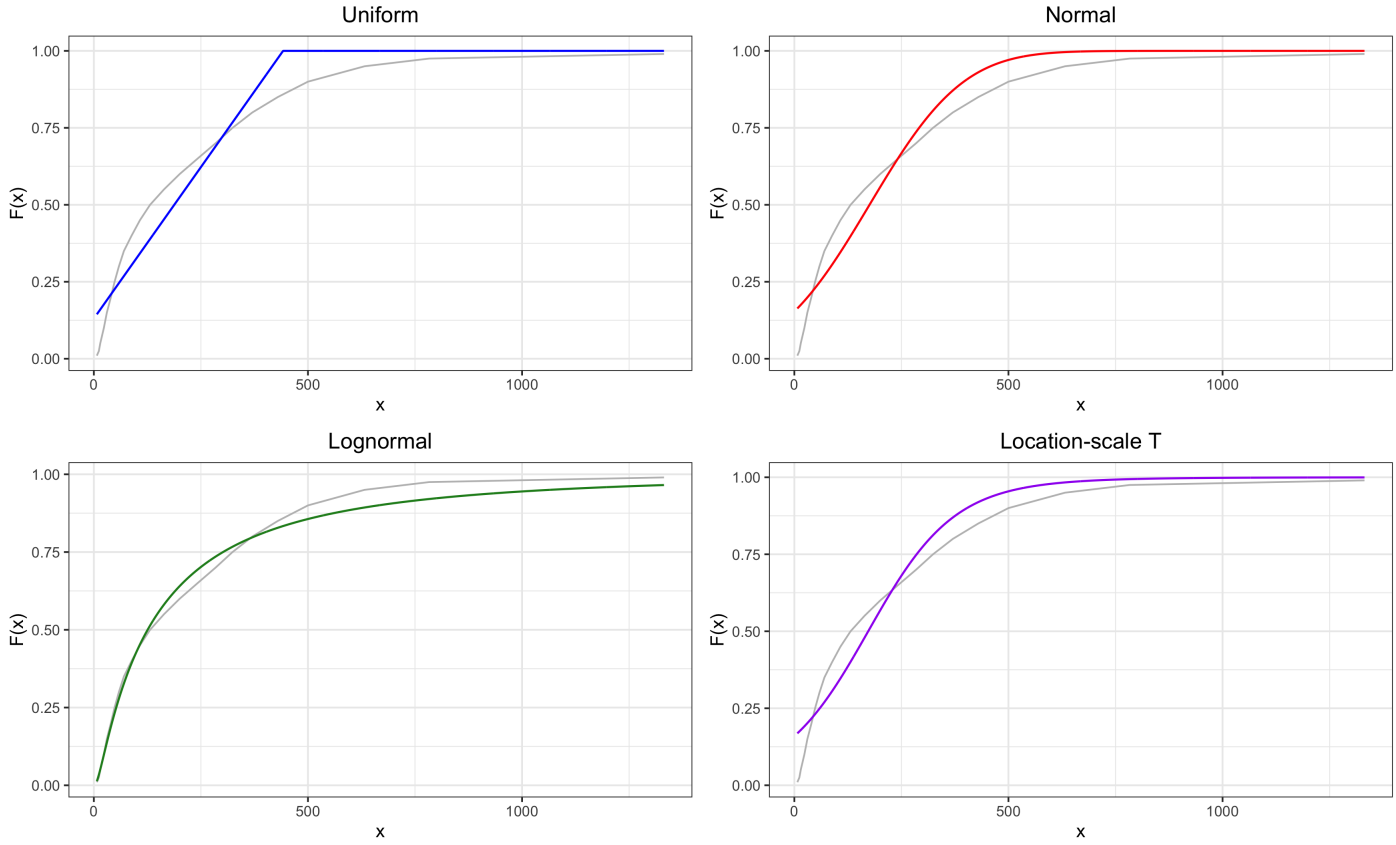


Figure 4.3: This figure shows the CDF plots for distributions fit to a set of quantiles. The grey line is a line of connected dots from the submitted quantiles and values and is the same in all four plots. This is the same set of forecasts as is 4.2. In no case here was the MSE below the cutoff value of the corresponding distribution.

5. Discussion

In this paper we have reviewed four representation types commonly used in probabilistic forecasting and discussed proper scoring, data storage, and ensemble model construction for each type. We presented the mixture distribution representation and argue that its use in collaborative probabilistic forecasting is preferable to the other representations. In terms of model flexibility, storage, and ensemble construction it is comparable to bin and quantile forecasts but also provides a forecast with a infinite nominal resolution. Based on a retrospective analysis, we argue that among the teams participating in the CDC flu competition and the COVID-19 Forecast Hub, there are already some which produce forecasts resembling single component mixture distributions, making the transition from past and current formats to a mixture distribution format straightforward. We thus advocate for the use of mixture distributions in future forecasting projects like those done in the CDC flu competition or in the COVID-19 Forecast Hub.

For a number of reasons, some forecasters may prefer not to the adopt mixture distributions as a format in collaborative forecasting. A collaborative forecast center, along with forecasters, using a different representation format may simply not want to break from tradition. There may be some concern that a mixture distribution does not represent well certain models. And the implementation of new scoring and ensemble construction methods may also be a barrier. Development of tools beyond what was used in Section 3.2 would assist in making a transition to using mixture distributions more straightforward. One aspect of ensemble construction which received little attention in this paper is the selection of weights for components of an ensemble where each of the components is a mixture distribution. Computing requirements could be a concern in such a problem, and further research on this may provide ideas of best methods for weight selection.

Another area of recommended research is the use of joint mixture distributions for forecasting. We have only considered here probabilistic forecasting of one event at a time, or example, the number of new infections in one week at one specific location. This forecast is presented as a marginal distribution for that specific target, time, and location. A joint distribution for forecasting multiple targets, times, or locations may sometimes be desirable and may require further consideration on how joint mixture distributions could be used as a format in collaboration.

BIBLIOGRAPHY

Alves, J.-H. G., Wittmann, P., Sestak, M., Schauer, J., Stripling, S., Bernier, N. B., McLean, J., Chao, Y., Chawla, A., Tolman, H., et al. (2013). The ncep–fnmoc combined wave ensemble product: Expanding benefits of interagency probabilistic forecasts to the oceanic environment. *Bulletin of the American Meteorological Society*, 94(12):1893–1905.

Baran, S. and Lerch, S. (2016). Mixture emos model for calibrating ensemble forecasts of wind speed. *Environmetrics*, 27(2):116–130.

Baran, S. and Lerch, S. (2018). Combining predictive distributions for the statistical post-processing of ensemble forecasts. *International Journal of Forecasting*, 34(3):477–496.

Bogner, K., Liechti, K., and Zappa, M. (2017). Combining quantile forecasts and predictive distributions of streamflows. *Hydrology and Earth System Sciences*, 21(11):5493–5502.

Bracher, J., Ray, E. L., Gneiting, T., and Reich, N. G. (2021). Evaluating epidemic forecasts in an interval format. *PLoS Computational Biology*, 17(2):e1008618.

Brooks, L. C., Ray, E. L., Bien, J., Bracher, J., Rumack, A., Tibshirani, R. J., and Reich, N. G. (2020). Comparing ensemble approaches for short-term probabilistic covid-19 forecasts in the us. *International Institute of Forecasters*.

Busetti, F. (2017). Quantile aggregation of density forecasts. *Oxford Bulletin of Economics and Statistics*, 79(4):495–512.

Camphausen, F., Kohl, M., Ruckdeschel, P., Stabla, T., and Ruckdeschel, M. P. (2007). The distr package.

CDC. Flusight: Flu forecasting. https://www.cdc.gov/flu/weekly/flusight/index.html. Accessed: 2022-02-03.

Chakraborty, S. (2015). Generating discrete analogues of continuous probability distributions -a survey of methods and constructions. *Journal of Statistical Distributions and Applications*, 2(1):1–30.

Cramer, E. Y., Huang, Y., Wang, Y., Ray, E. L., Cornell, M., Bracher, J., Brennen, A., Castro Rivadeneira, A. J., Gerding, A., House, K., Jayawardena, D., Kanji, A. H., Khandelwal, A., Le, K., Niemi, J., Stark, A., Shah, A., Wattanachit, N., Zorn, M. W., Reich, N. G., and Consortium, U. C.-. F. H. (2021a). The united states covid-19 forecast hub dataset. *medRxiv*.

Cramer, E. Y., Lopez, V. K., Niemi, J., George, G. E., Cegan, J. C., Dettwiller, I. D., England, W. P., Farthing, M. W., Hunter, R. H., Lafferty, B., et al. (2021b). Evaluation of individual and ensemble probabilistic forecasts of covid-19 mortality in the us. *medRxiv*.

Dominicy, Y. and Veredas, D. (2013). The method of simulated quantiles. *Journal of Econometrics*, 172(2):235–247.

GitHub. Github: Covid-19 forecasts. https://api.github.com/repos/reichlab/covid19-forecast-hub. Accessed: 2022-04-04.

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 Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151.

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. and Ranjan, R. (2013). Combining predictive distributions. *Electronic Journal of Statistics*, 7:1747–1782.

Groen, J. J., Paap, R., and Ravazzolo, F. (2013). Real-time inflation forecasting in a changing world. *Journal of Business & Economic Statistics*, 31(1):29–44.

Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5):559–570.

Kemp, A. W. (2004). Classes of discrete lifetime distributions. *Communications in Statistics – Theory and Methods*, 33(12):3069–3093.

Krueger, F., Lerch, S., Thorarinsdottir, T. L., and Gneiting, T. (2016). Probabilistic forecasting and comparative model assessment based on markov chain monte carlo output. *arXiv preprint arXiv:1608.06802*, 12.

Leutbecher, M. and Palmer, T. N. (2008). Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539.

Lewis, J. M. (2005). Roots of ensemble forecasting. *Monthly Weather Review*, 133(7):1865–1885.

Lichtendahl Jr, K. C., Grushka-Cockayne, Y., and Winkler, R. L. (2013). Is it better to average probabilities or quantiles? *Management Science*, 59(7):1594–1611.

McAndrew, T. and Reich, N. G. (2019). Adaptively stacking ensembles for influenza forecasting with incomplete data. *arXiv preprint arXiv:1908.01675*.

McGowan, C. J., Biggerstaff, M., Johansson, M., Apfeldorf, K. M., Ben-Nun, M., Brooks, L., Convertino, M., Erraguntla, M., Farrow, D. C., Freeze, J., et al. (2019). Collaborative efforts to forecast seasonal influenza in the united states, 2015–2016. *Scientific Reports*, 9(1):1–13.

Nguyen, H. D. and McLachlan, G. (2019). On approximations via convolution-defined mixture models. *Communications in Statistics-Theory and Methods*, 48(16):3945–3955.

Nirwan, R.-S. and Bertschinger, N. (2020). Bayesian quantile matching estimation. *arXiv preprint arXiv:2008.06423*.

Peel, D. and MacLahlan, G. (2000). Finite mixture models. *John & Sons*.

Quinonero-Candela, J., Rasmussen, C. E., Sinz, F., Bousquet, O., and Schölkopf, B. (2005). Evaluating predictive uncertainty challenge. In *Machine Learning Challenges Workshop*, pages 1–27. Springer.

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.

Ratcliff, R. (1979). Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin*, 86(3):446.

Ray, E. L. and Reich, N. G. (2018). Prediction of infectious disease epidemics via weighted density ensembles. *PLoS computational biology*, 14(2):e1005910.

Ray, E. L., Wattanachit, N., Niemi, J., Kanji, A. H., House, K., Cramer, E. Y., Bracher, J., Zheng, A., Yamana, T. K., Xiong, X., et al. (2020). Ensemble forecasts of coronavirus disease 2019 (covid-19) in the us. *MedRXiv*.

Reich, N. G., Brooks, L. C., Fox, S. J., Kandula, S., McGowan, C. J., Moore, E., Osthus, D., Ray, E. L., Tushar, A., Yamana, T. K., et al. (2019a). A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the united states. *Proceedings of the National Academy of Sciences*, 116(8):3146–3154.

Reich, N. G., McGowan, C. J., Yamana, T. K., Tushar, A., Ray, E. L., Osthus, D., Kandula, S., Brooks, L. C., Crawford-Crudell, W., Gibson, G. C., et al. (2019b). Accuracy of real-time multi-model ensemble forecasts for seasonal influenza in the us. *PLoS Computational Biology*, 15(11):e1007486.

Schepen, A. and Wang, Q. (2015). Model averaging methods to merge operational statistical and dynamic seasonal streamflow forecasts in australia. *Water Resources Research*, 51(3):1797–1812.

Taylor, J. W. (2021). Evaluating quantile-bounded and expectile-bounded interval forecasts. *International Journal of Forecasting*, 37(2):800–811.

Vrugt, J. A., Diks, C. G., and Clark, M. P. (2008). Ensemble bayesian model averaging using markov chain monte carlo sampling. *Environmental Fluid Mechanics*, 8(5):579–595.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Yamana, T. K., Kandula, S., and Shaman, J. (2016). Superensemble forecasts of dengue outbreaks. *Journal of The Royal Society Interface*, 13(123):20160410.

Zoltar. Zoltardata: Cdc retrospective forecasts. https://zoltardata.com/project/6. Accessed: 2022-04-04.

Zoltar. Zoltardata: Covid-19 forecasts. https://zoltardata.com/project/44. Accessed: 2022-04-04.

APPENDIX

The following is the code to make the `MakeDist()` function introduced in 3.2. Tables 1 and 2 show how distribution family arguments are to be written and what parameters are to be used for the distributions in the `MakeDist()` function.

```
MakeDist <- function(distsdf){


  distdf <- distsdf[distsdf[,1] != 'Lst',]
  tdist <- distsdf[distsdf[,1] == 'Lst',]


  fun_dist <-
    apply(distdf, FUN=function(x) {
      paste('distr::',x[1], '(',
            ifelse(!is.na(x[2]) & (!is.na(x[3]) | !is.na(x[4])),
                   paste(x[2],',',sep=''),
                   ifelse(!is.na(x[2]) & is.na(x[3]) & is.na(x[4]),
                          x[2], '')),
            ifelse(!is.na(x[3]) & !is.na(x[4]),
                   paste(x[3],',',sep=''),
                   ifelse(!is.na(x[3]) & is.na(x[4]), x[3],'')),
            ifelse(!is.na(x[4]),x[4],''), ')',sep='')
    }, MARGIN = 1
    )
```

```
  fun_tdist <- apply(tdist, FUN=function(x) {
    paste0('distr::Td(',x[4],')*',x[3], '+', x[2])
  }, MARGIN = 1
  )

  dist_args <- paste(fun_dist, collapse=',',sep='')
  tdist_args <- paste0(fun_tdist,collapse=',')
  args <- ifelse(tdist_args!='',paste(dist_args,tdist_args,sep=','),dist_args)

  weights <- c(distdf[,5],tdist[,5])
  mixString <- paste('distr::UnivarMixingDistribution(',
                     args,',mixCoeff=weights)',sep='')
  mixDist <- eval(parse(text=mixString))

  return(mixDist)
}
```

The following is the code used to make the function `CRPS()` used in section 3.2.

```
crps_integrand <- function(x,dist,y) {(dist(x) - as.numeric(y <= x))^2}

CRPS <- function(y,dist) {
  int <- integrate(crps_integrand,-Inf,Inf,y,dist=dist)
  return(int$value)
}
```

Argument	**Summary**	**Options**
dist	A string specifying a distribution family.	Beta, Cauchy, Lnorm, Logis, "Unif, Lst (location scale t distribution), Weibull, Fd, Norm, Chisq, Gammad, Exp Binom, Dirac, Pois, Hyper, Nbinom, Geom
param1	A real number specifying the first parameter value of the distribution.	Beta: shape1; Cauchy: location; Lnorm: meanlog; Logis: location; Unif: min; Lst: location; Weibull: shape; Fd: df1; Norm: mean; Chisq: df; Gammad: scale; Exp: rate; Binom: size; Dirac: location; Pois: lambda; Hyper: m; Nbinom: n; Geom: prob
param2	A real number specifying the second parameter value of the distribution.	Beta: shape2; Cauchy: scale; Lnorm: slog; Logis: scale; Unif: Max; Lst: scale; Weibull: scale; Fd: df2; Norm: sd; Chisq: ncp; Gammad: shape; Binom: prob; Hyper: n; Nbinom: p
param3	A real number specifying the third parameter value of the distribution.	Lst: df; Hyper: k
weight	A real number between 0 and 1 specifying the weight given to the distribution in the overall mixture distribution. The sum of the weight column should equal 1.	...

Table 1: This table describes how each argument may be entered into a data frame entered into the `MakeDist()` function, including exactly how the distribution families should appear and what paramaters should be included for each distribution family.

Distribution	**dist**	**param1**	**param2**	**param3**
Beta	Beta	shape1	shape2	
Cauch	Cauchy	location	scale	
Log-normal	Lnorm	meanlog	slog	
Logistic	Logis	location	scale	
Uniform	Unif	min	max	
Location Scale T	Lst	location	scale	df
Weibull	Weibull	shape	scale	
F	Fd	df1	df2	
Normal	Norm	mean	sd	
Chisqure	Chisq	df		
Gamma	Gammad	scale	shape	
Exponential	Exp	rate		
Binomial	Binom	size	prob	
Dirac	Dirac	location		
Poisson	Pois	lambda		
Hypergeometric	Hyper	m	n	k
Negative binomial	Nbinom	n	p	
Geometric	Geom	prob		

Table 2: This table shows what parameters may be used for each distribution family used in a data frame for the `MakeDist()` function or in the `UnivarMixingDistribution()` function.