Evaluating Distributional Forecasts

Leonidas Tsaprounis

PyData London meet-up
04 July 2023

About me

- Senior Data Scientist at HALEON
- Core Developer at 

- A little obsessed with time series forecasting metrics
- My pet peeve is carrying an umbrella for no reason

in/leonidas-tsaprounis-302ba882

https://github.com/ltsaprounis

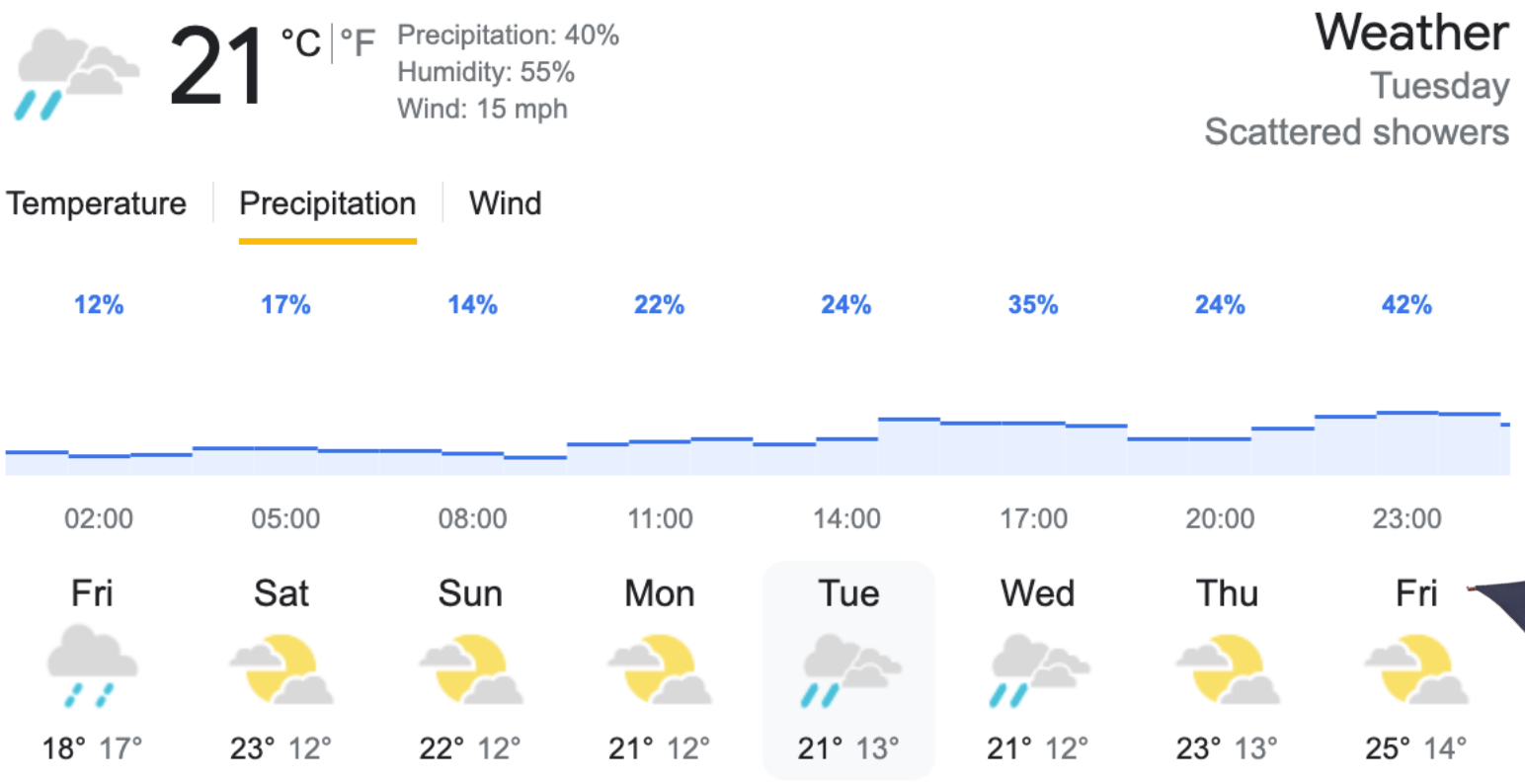
https://medium.com/@leonidas.tsap

What's this talk about?

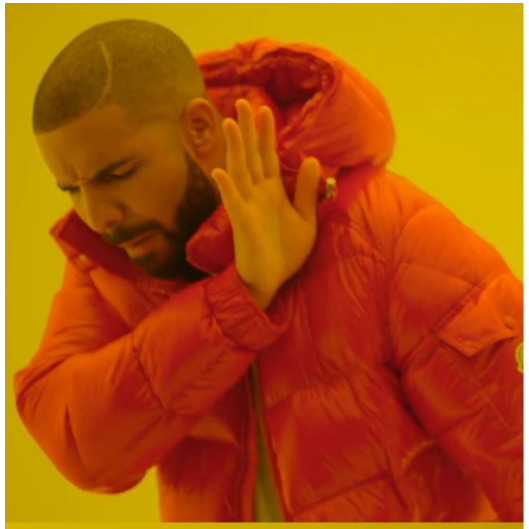
By the end of this talk, I want to:

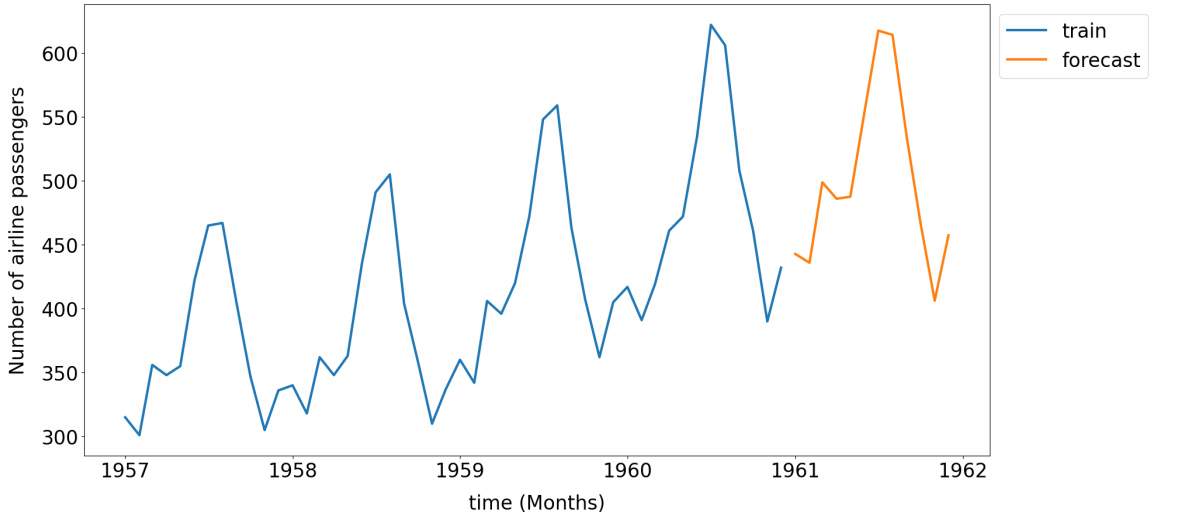
- Convince you that Distributional Forecasts are more useful than Point Forecasts
- Show you how to evaluate them
- Make you aware of the properties of the different metrics used for distributional forecasts

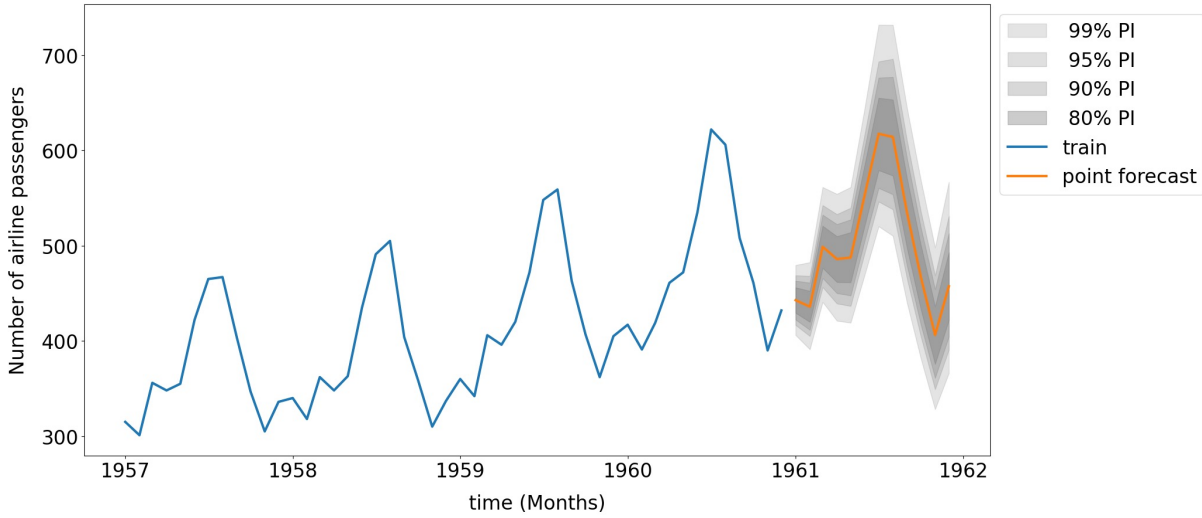
A little thought experiment



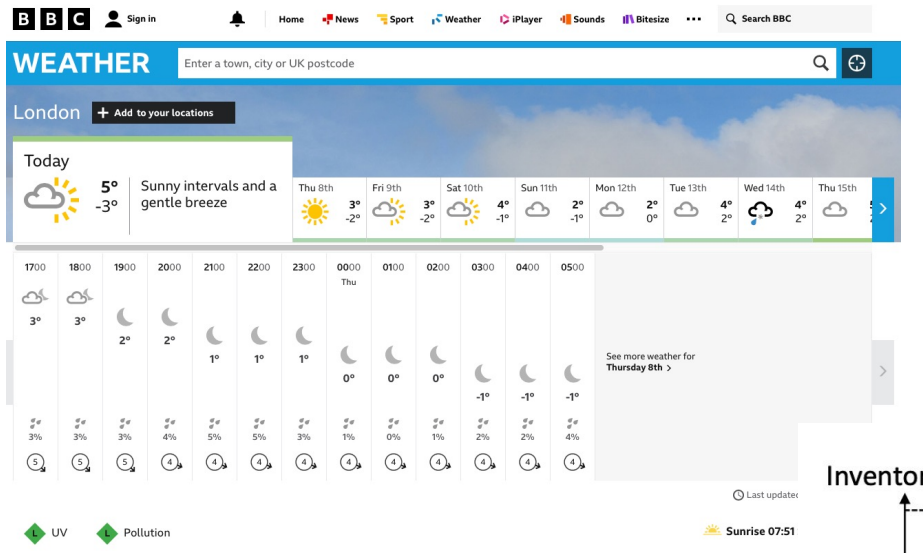
Point Forecasts VS Distributional Forecasts

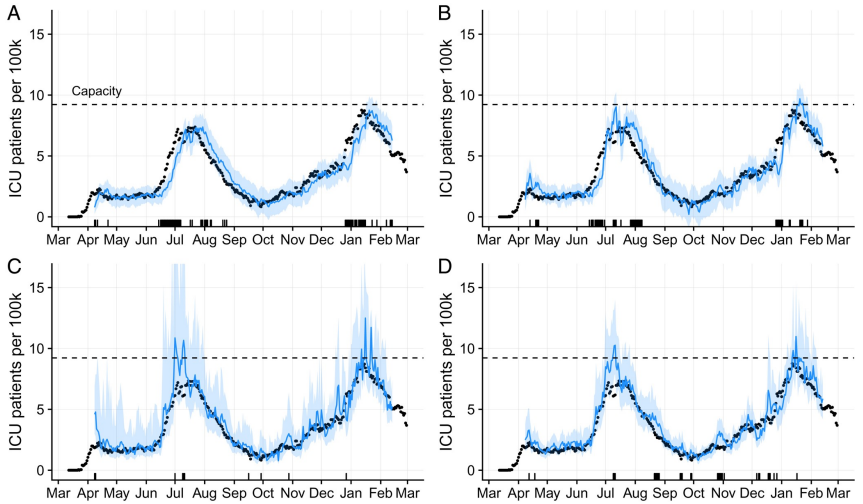


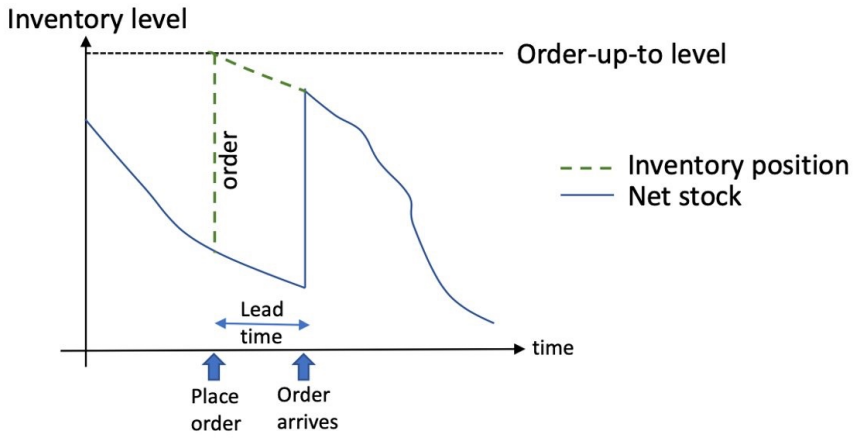




What are distributional forecasts useful for?





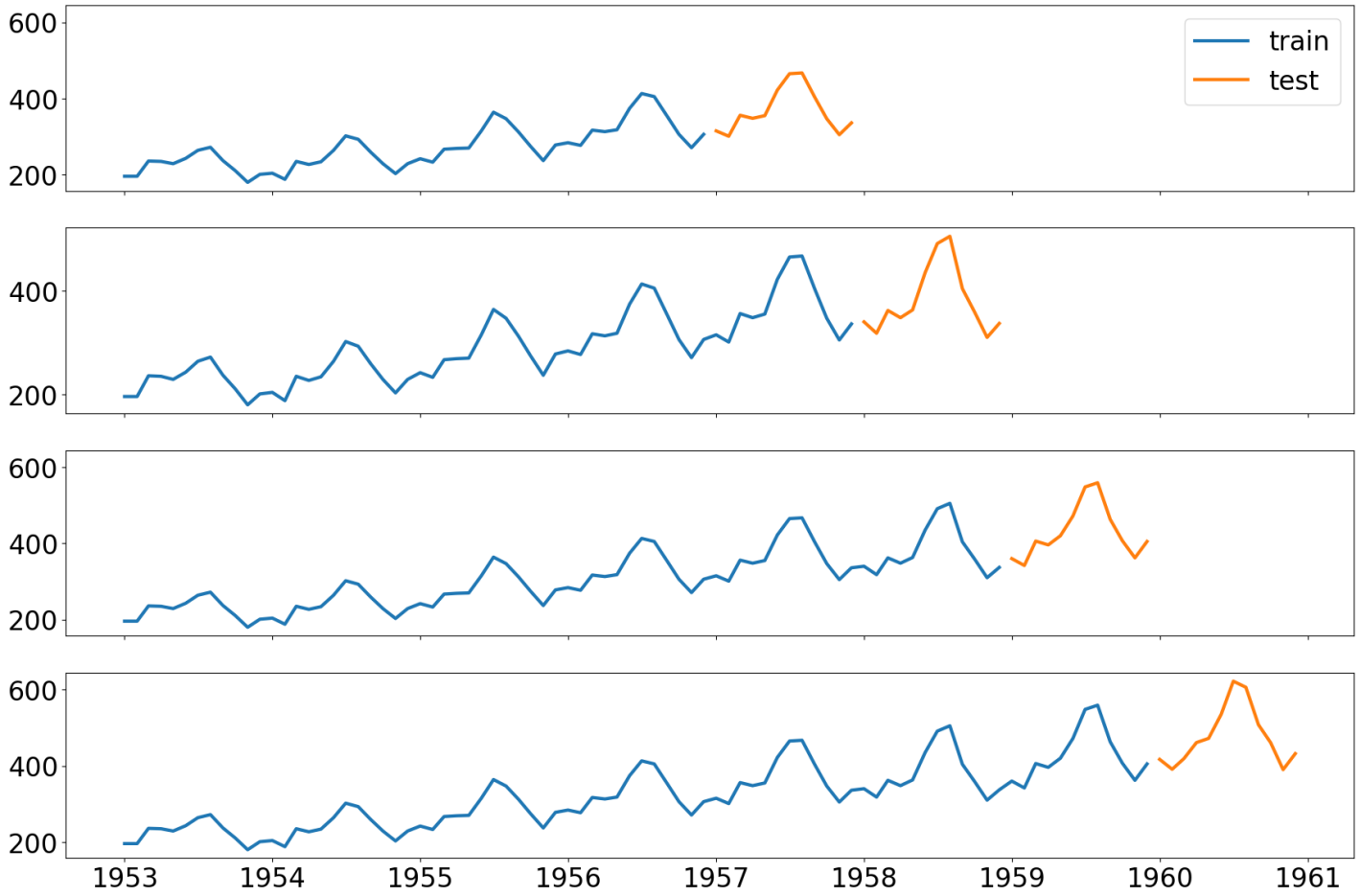


Decision-making in the time of a pandemic

Probabilistic reasoning has a role if not the final word

https://www.ft.com/content/c614480f-9a04-49fa-9ad2-bd9481493eb0

How do we evaluate a forecasting model?



Quick summary of point forecasting metrics

Point Forecasting metrics

Scale Dependent Error Metrics

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{h}\Sigma_{t=T+1}^{T+h}(y_t - \hat{y}_t)^2}$$

Mean Absolute Error:

$$MAE = \frac{1}{h}\Sigma_{t=T+1}^{T+h}|y_t - \hat{y}_t|$$

Percentage Error Metrics

Mean Absolute Percentage Error:

$$MAPE = \frac{1}{h}\Sigma_{t=T+1}^{T+h}\frac{|y_t - \hat{y}_t|}{y_t}$$

Symmetric Mean Absolute Percentage Error:

$$sMAPE = \frac{200}{h}\Sigma_{t=T+1}^{T+h}\frac{|y_t - \hat{y}_t|}{|y_t + \hat{y}_t|}$$

Scaled Error Metrics

Root Mean Squared Scaled Error:

$$RMSSE = \sqrt{\frac{\frac{1}{h}\Sigma_{i=T+1}^{T+h}(y_t - \hat{y}_t)^2}{\frac{1}{T-1}\Sigma_{i=2}^{T}(y_t - y_{t-1})^2}}$$

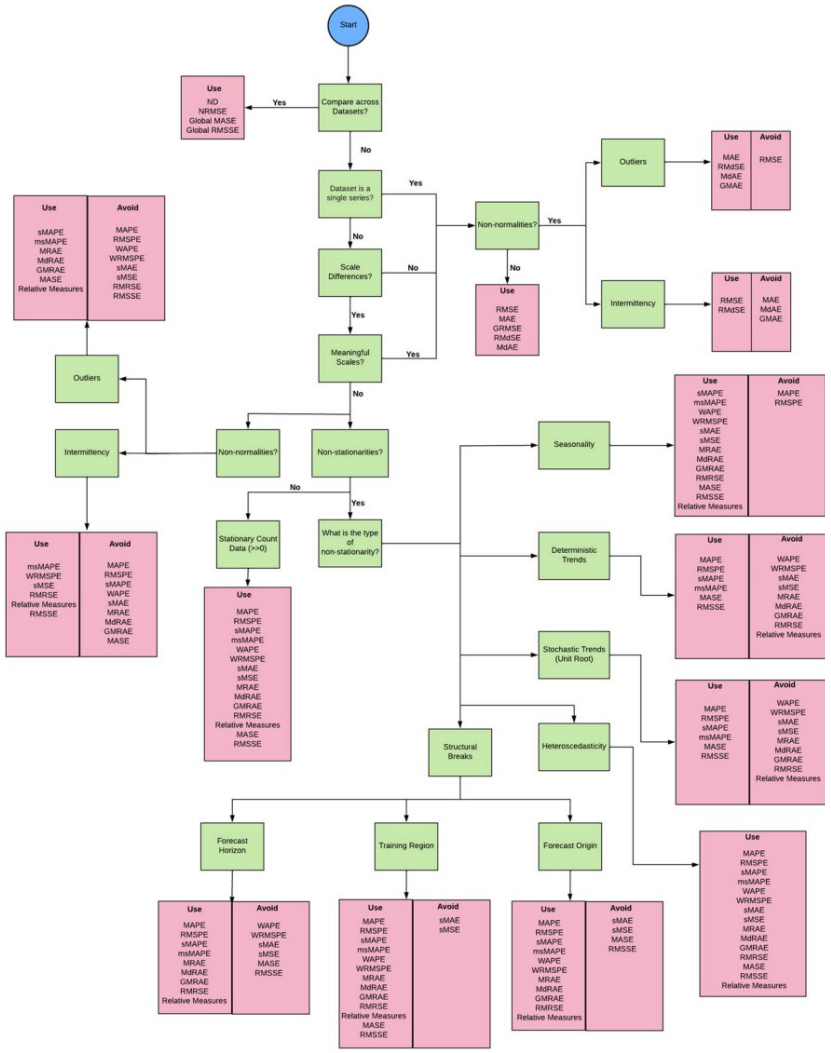
Mean Absolute Scaled Error:

$$MASE = \frac{\frac{1}{h}\Sigma_{t=T+1}^{T+h}|y_t - \hat{y}_t|}{\frac{1}{T-1}\Sigma_{t=2}^{T}|y_t - y_{t-1}|}$$

Point Forecasting metrics

Table 1: Checklist for selecting error measures for final forecast evaluation based on different time series characteristics

Stationary Count Data (>>0)	Seasonality	Trend (Linear/Exp.)	Unit Roots	Heteroscedasticity	Structural Breaks (With Scale Differences)			Intermittence	Outliers	Error Measures	Scaling	
					Forecast Horizon	Training Region	Forecast Origin					
✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	RMSE	None	
✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	MAE		
✓	✗	✓	✓†	✓†	✓	✓	✓	✗	✗	MAPE		
✓	✗	✓	✓†	✓†	✓	✓	✓	✗	✗	RMSPE	OOS Per Step	
✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	sMAPE		
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	msMAPE		
✓	✓	✗	✗	✓	✗	✓	✓	✗	✗	WAPE	OOS Per Series	Actual Values
✓	✓	✗	✗	✓	✗	✓	✓	✓†	✗	WRMSPE		
✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	sMAE	In-Sample Per Series	
✓	✓	✗	✗	✓	✗	✗	✗	✓†	✗	sMSE		
✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	ND	OOS All Series	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	NRMSE		
✓†	✓†	✗	✗	✓	✓	✓†	✓	✗	✓†	MRAE	OOS Per Step	Benchmark Errors
✓†	✓†	✗	✗	✓	✓	✓†	✓	✗	✓†	MdRAE		
✓†	✓†	✗	✗	✓	✓	✓†	✓	✗	✓†	GMRAE		
✓†	✓†	✗	✗	✓	✓	✓†	✓	✓†	✗	RMRSE		
✓†	✓†	✗	✗	✓	✓	✓†	✓	✓†	✓†	Relative Measures	OOS Per Series	
✓†	✓†	✓	✓	✓	✗	✓†	✗	✗	✓	MASE	In-Sample Per Series	
✓†	✓†	✓	✓	✓	✗	✓†	✗	✓†	✗	RMSSE		
✓†	✓†	✓	✓	✓	✓	✓	✓	✓	✓		In-Sample All Series	
✓	✓	✓†	✓	✓†	✓	✓	✓	✓	✓	Measures with Transformations	None	

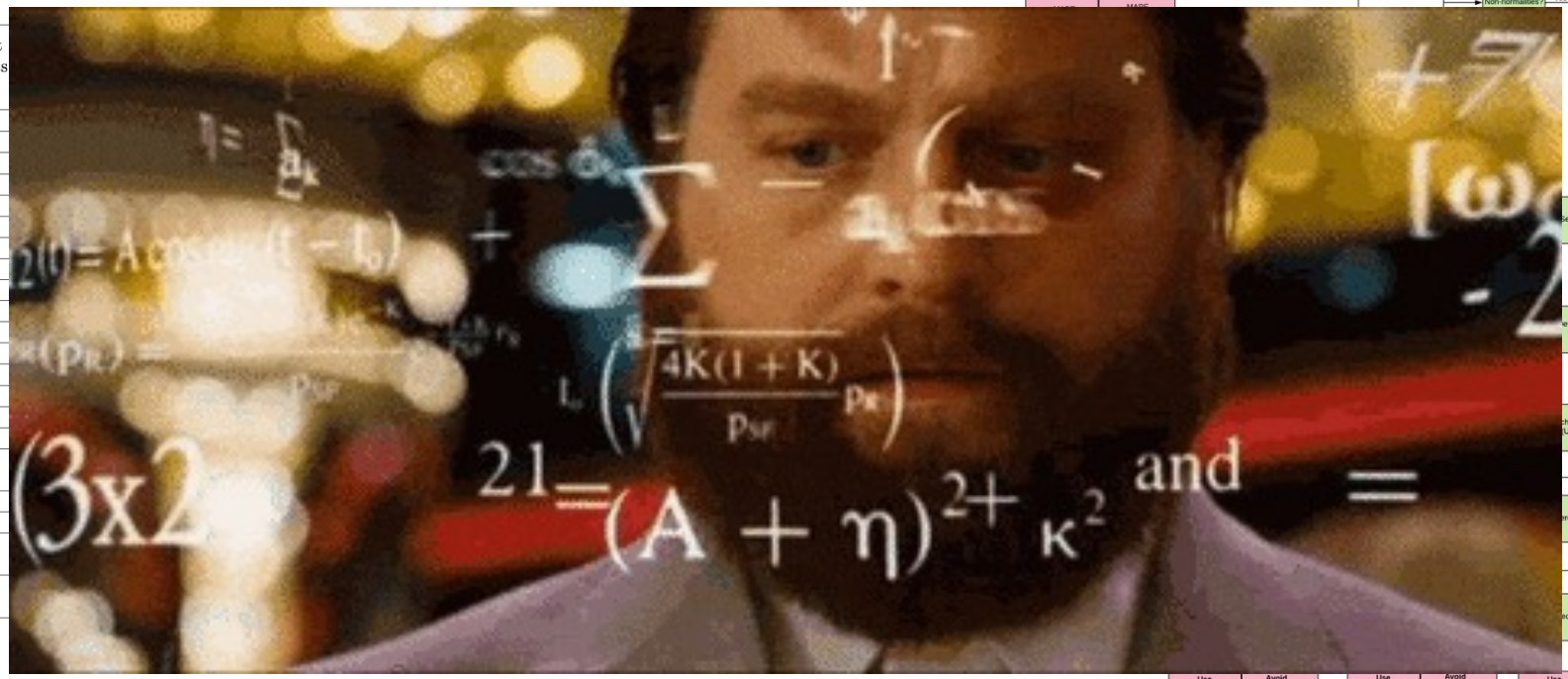


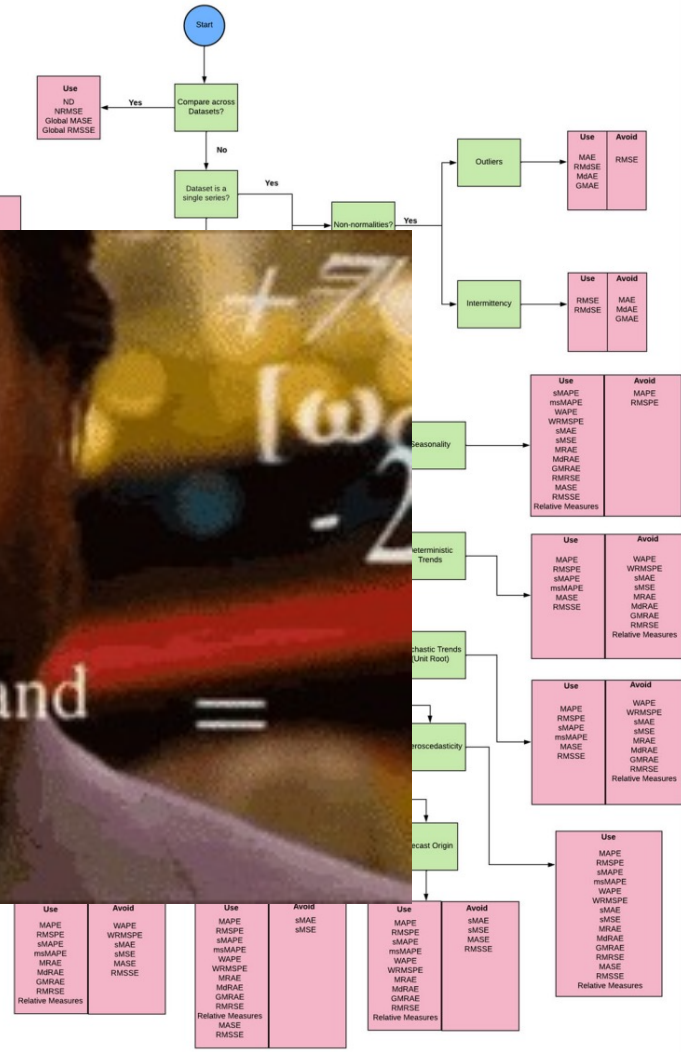
Hewamalage, H., Ackermann, K. & Bergmeir, C. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Min Knowl Disc* (2022). https://doi.org/10.1007/s10618-022-00894-5

Point Forecasting metrics

Table 1: Checklist for selecting error measures for final forecast evaluation based on different time series characteristics

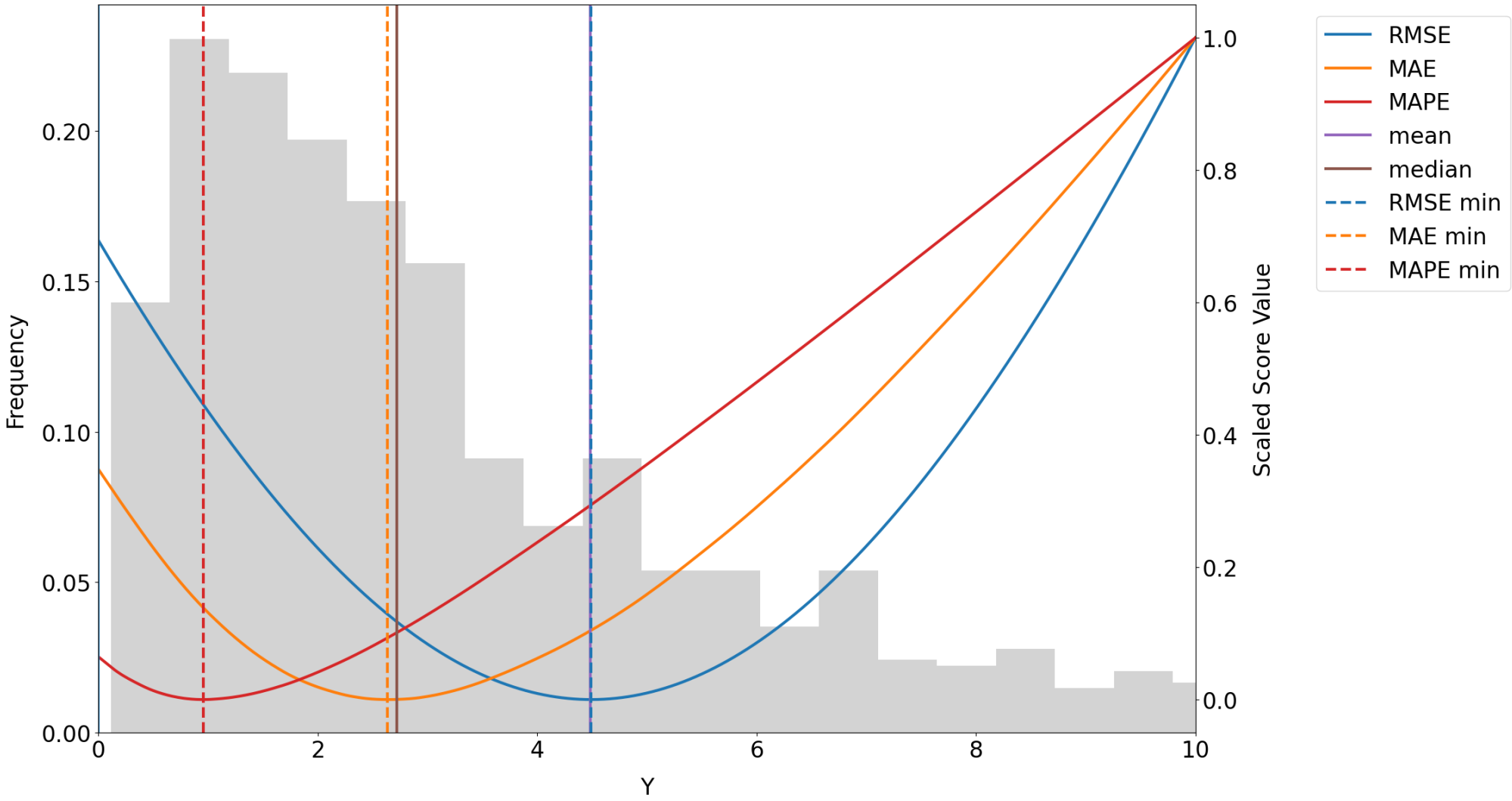
Stationary Count Data (>>0)	Seasonality	Trend (Linear/Exp.)	Unit Roots
✓	✓	✓	✓
✓	✓	✓	✓
✓	✗	✓	✓†
✓	✗	✓	✓†
✓	✓	✓	✓
✓	✓	✓	✓
✓	✓	✗	✗
✓	✓	✗	✗
✓	✓	✗	✗
✓	✓	✗	✗
✓	✓	✓	✓
✓	✓	✓	✓
✓†	✓†	✗	✗
✓†	✓†	✗	✗
✓†	✓†	✗	✗
✓†	✓†	✗	✗
✓†	✓†	✗	✗
✓†	✓†	✓	✓
✓†	✓†	✓	✓
✓†	✓†	✓	✓
✓	✓	✓†	✓





Hewamalage, H., Ackermann, K. & Bergmeir, C. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Min Knowl Disc* (2022). https://doi.org/10.1007/s10618-022-00894-5

Most metrics are “geared” towards a statistical functional



Tilmann Gneiting (2011) Making and Evaluating Point Forecasts, Journal of the American Statistical Association, 106:494, 746-762, DOI: 10.1198/jasa.2011.r10138

Horses for Courses

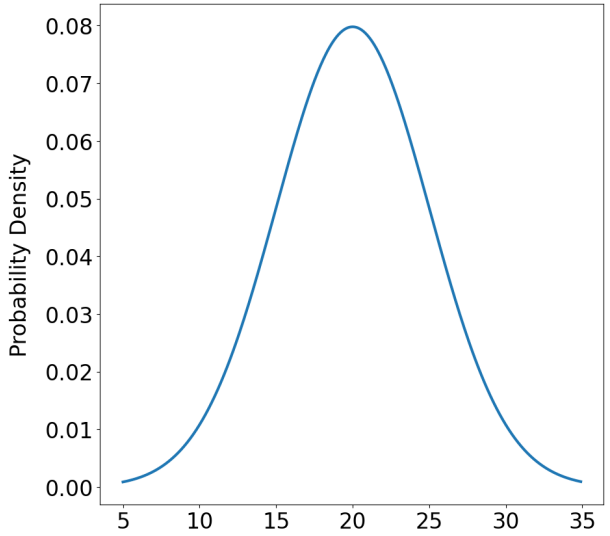
Inventory Optimization

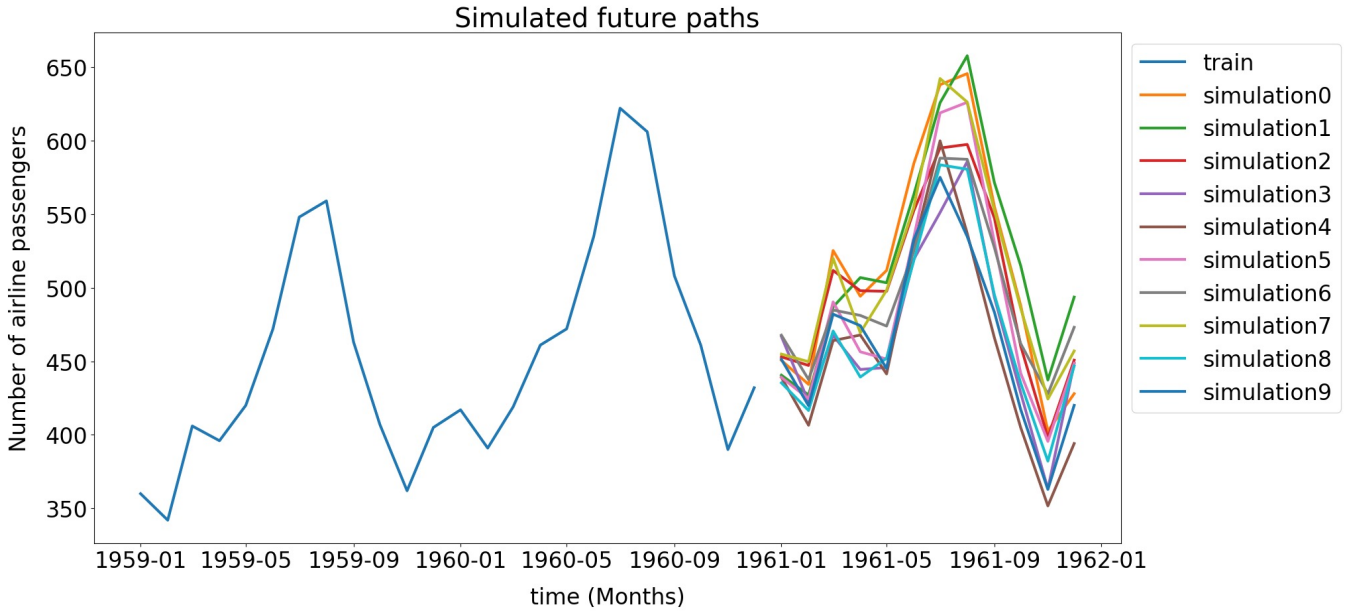
Best MAPE model

Back to Distributional Forecasts

Distributional Forecasts

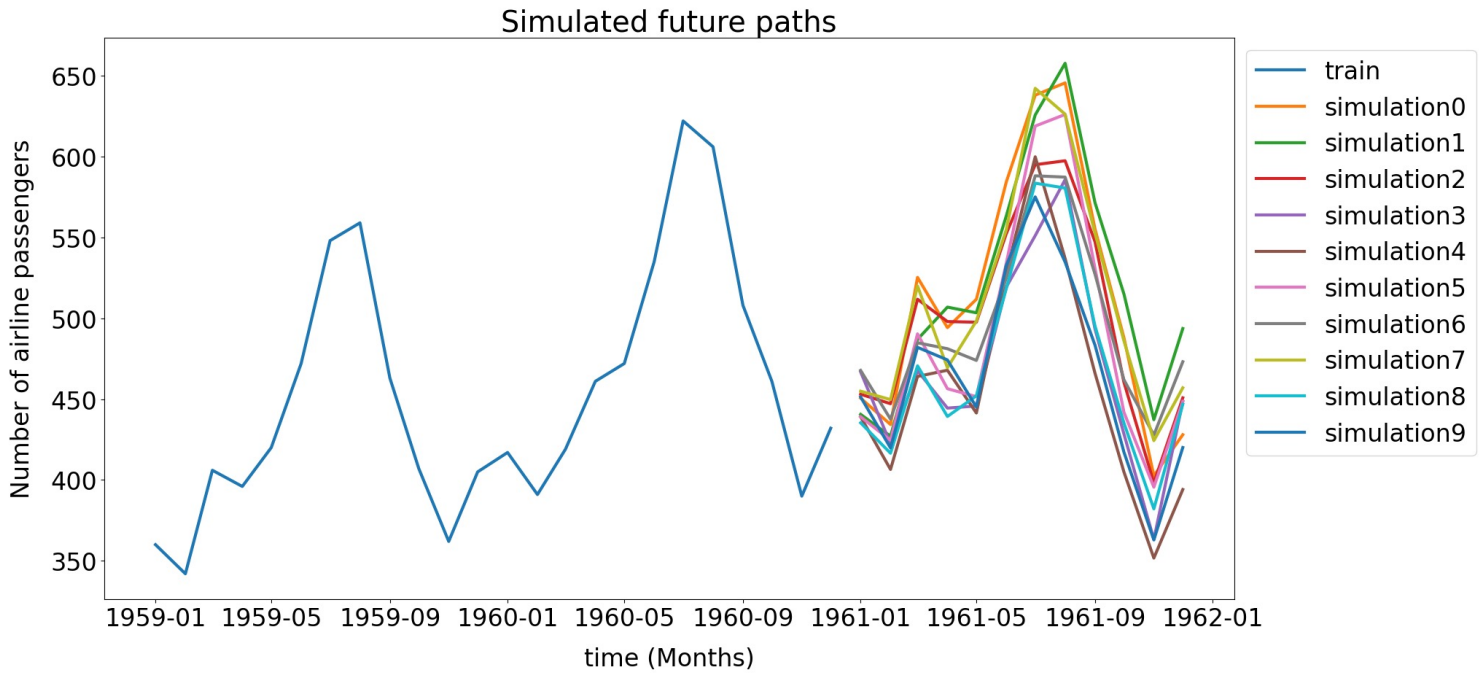
Distributional forecasts can be in the form of analytical distributions or in the form of sampled future paths.





Sampled Future paths

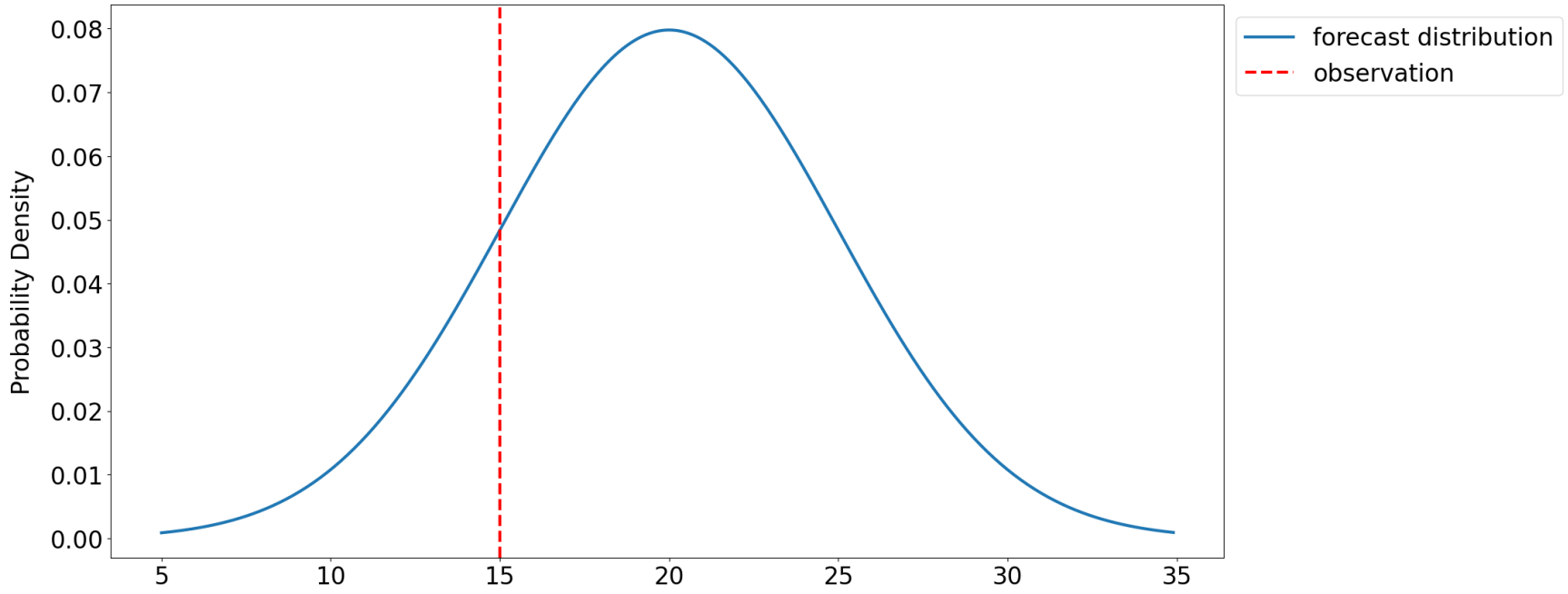
We usually represent sampled future paths in the form of a matrix, with the time dimension on the rows and each simulation as a column.



	Simulated Future Paths		
Forecasted Step	1	…	N
0			
1			
2			
3			
.			
.			
.			
10			
11			
12			

What is a good metric for Distributional Forecasts?

Although we produce distributional forecasts, we will never be able to observe the true data generating distribution. As a matter of fact, it's a fictional concept. We will only observe point values, that we assume are the results of sampling from that data generating distribution. So, what is a good metric that allows us to compare our predicted distribution to a point value?



What is a good distributional forecast?

"... maximizing the sharpness of the predictive distributions subject to calibration. Calibration refers to the statistical consistency between the distributional forecasts and the observations and is a joint property of the predictions and the events that materialize. Sharpness refers to the concentration of the predictive distributions and is a property of the forecasts only."

- Calibration: On average, events occur at a frequency that matches the forecasted distributions
- Sharpness: The probability is "concentrated" making the forecast informative.

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: 243-268. https://doi.org/10.1111/j.1467-9868.2007.00587.x

Proper Scoring Rules

"A scoring rule is proper if the forecaster maximises the expected score for an observation drawn from the distribution F if they issue the probabilistic forecast F, rather than G≠F. It is strictly proper if the maximum is unique."

So, if a forecaster wants to achieve the best score possible, if the scoring rule is strictly proper, they can only do so by providing the true data generating distribution as their forecast. Therefore, a scoring rule that is strictly proper encourages honest distributional forecasts.

Today we will go through the properties and the computational methods for 2 proper scoring rules:

- Log Score
- Continuous Ranked Probability Score (CRPS)

We'll calculate these scores for forecasts in the form of scipy distributions and sampled paths represented by numpy matrices

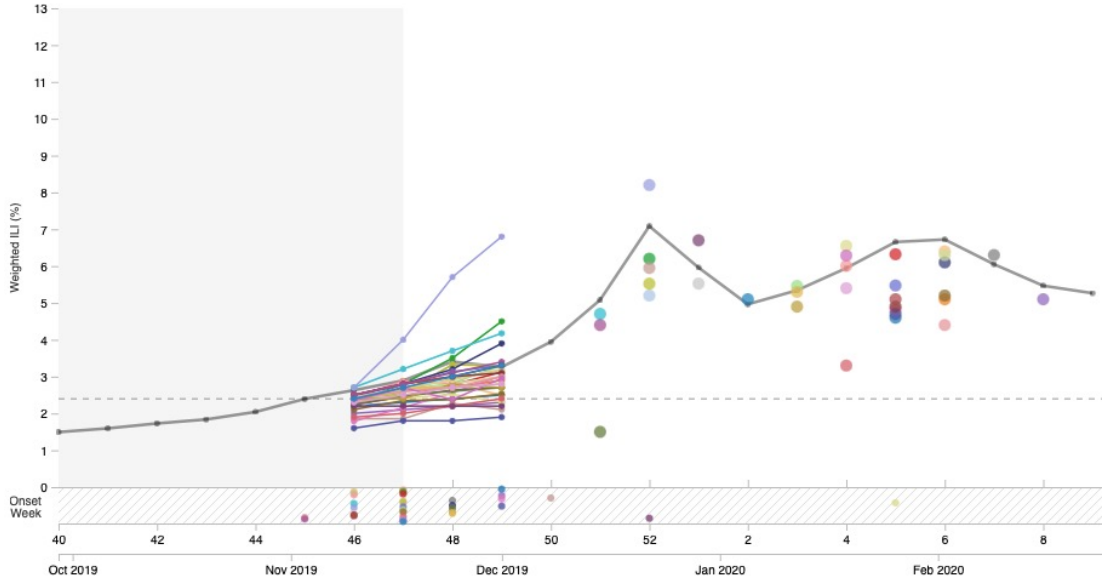
Tilmann Gneiting & Adrian E Raftery (2007) Strictly Proper Scoring Rules, Prediction, and Estimation, Journal of the American Statistical Association, 102:477, 359–378,
DOI: 10.1198/016214506000001437

Log Score

The Log score (LogS), introduced by I. J. Good in 1952, is the log of the probability density function (PDF) of the predicted distribution for the actual value.

LogS(F, y) = log(F(y)), where F is the PDF of the predicted distribution and y is the value of the observation. Because we use the convention that a smaller value is better for all scores, we will use -LogS.

The Log Score was the metric used in the CDC's flusight competition that has gained wide recognition in the epidemiological forecasting community.

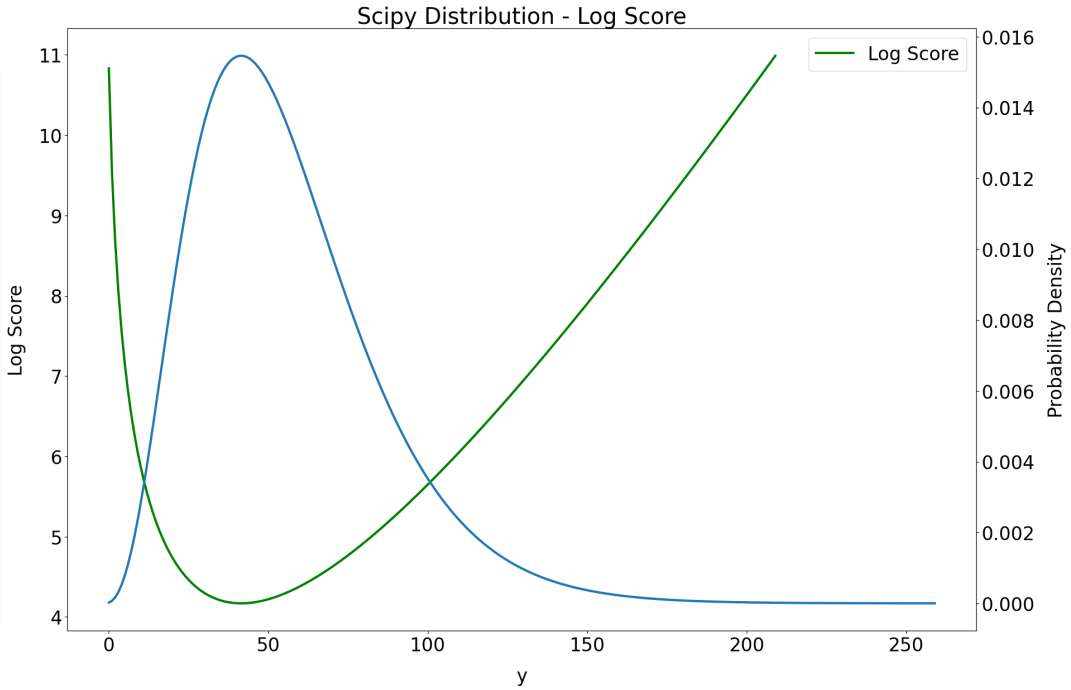


Good, I.J. (1952), Rational Decisions. Journal of the Royal Statistical Society: Series B (Methodological), 14: 107–114. https://doi.org/10.1111/j.2517-6161.1952.tb00104.x

Image from: https://predict.cdc.gov/post/5d8257befba2091084d47b4c

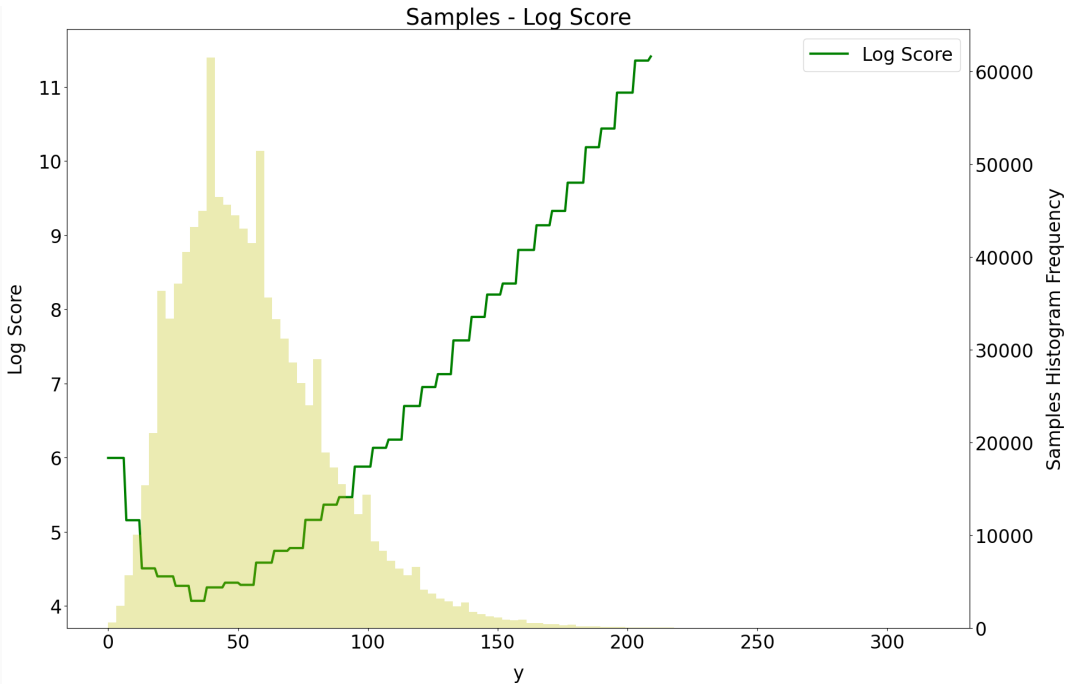
Log Score – distributions

```python
def scipy_dist_log_score(
    dist_list: List[rv_frozen],
    actuals: np.array,
) -> np.array:
    probs = [
        dist_list[i].pmf(actuals[i])
        if hasattr(dist_list[i], "pmf")
        else dist_list[i].pdf(actuals[i])
        for i in range(len(dist_list))
    ]
    return -np.log(probs)
```



Log Score - samples

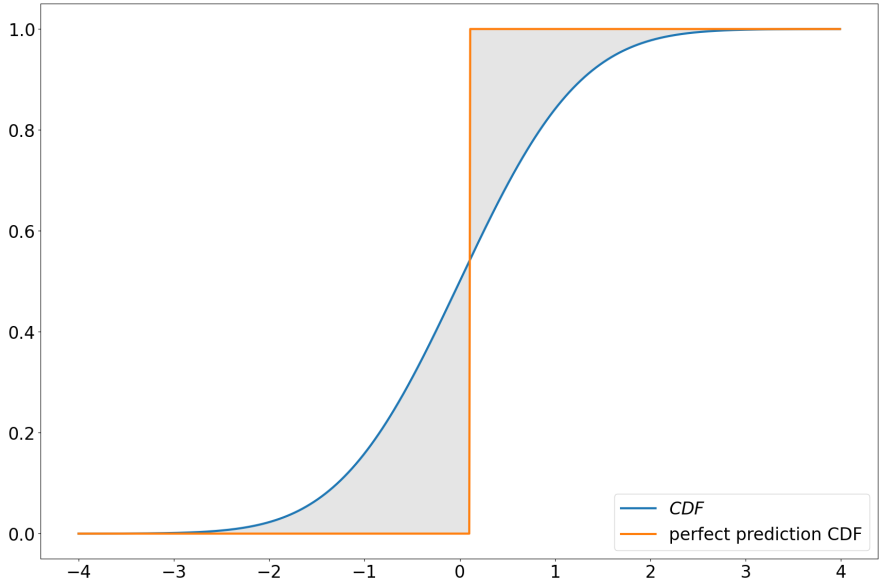
```python
def sample_log_score(
    samples: np.array, actuals: np.array, n_bins: int = 50
) -> np.array:
    hist = [
        np.histogram(samples[i], bins=n_bins, density=True)
        for i in range(samples.shape[0])
    ]

    probs = [
      hist[i][0][np.digitize(actuals[i], hist[i][1])]
      for i in range(len(hist))
    ]

    return -np.log(probs)
```



CRPS – Continuous Ranked Probability Score

CRPS is based on the cumulative distribution function (CDF) as opposed to the probability density function (PDF) used in the log score. This makes it more suitable for cases where the forecast is in the form of samples because we can use the empirical CDF. CRPS was introduced as a proper score for continuous distributions back in 1976 by James E. Matheson and Robert L. Winkler.

Conceptually, CRPS is quantifying the difference of the predicted distribution CDF to a “perfect” prediction CDF. A perfect distributional forecast for a given value of an actual observation is essentially a point forecast that is equal to the actual value. The CDF of a perfect prediction is a step function going from 0 to 1, with the step being on the actual value.



Matheson, J. E., & Winkler, R. L. (1976). Scoring Rules for Continuous Probability Distributions. Management Science, 22(10), 1087–1096. http://www.jstor.org/stable/2629907

CRPS – Continuous Ranked Probability Score

More formally:

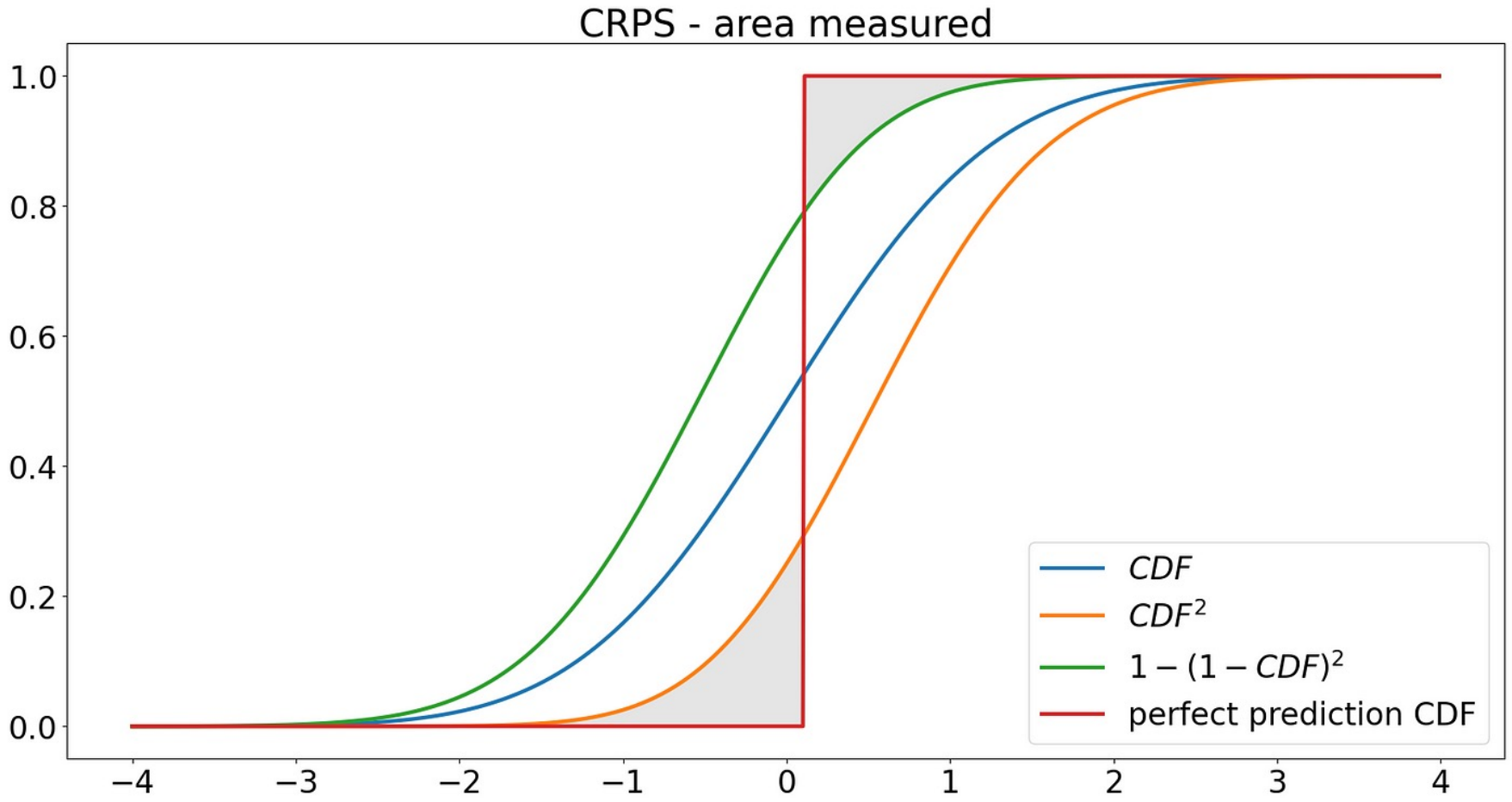
$$CRPS = \int_{-\infty}^{\infty} (F(x) - \mathbb{1}(x \geq y))^2 dx$$

where $F(x)$ is the cdf of the predicted distribution and $\mathbb{1}(x \geq y)$ is a function that is 1 if the inequaltiy $x \geq y$ is true and zero otherwise.

Therefore it can also be expressed as:

$$CRPS = \int_{-\infty}^{y} F(x)^2 dx + \int_{y}^{\infty} (F(x) - 1)^2 dx$$

CRPS – Area measured



Bracher J, Ray EL, Gneiting T, Reich NG (2021) Evaluating epidemic forecasts in an interval format. PLOS Computational Biology 17(2): e1008618. https://doi.org/10.1371/journal.pcbi.1008618

For point forecasts CRPS is equal to the Mean Absolute Error

Let $\hat{y}$ be the prediction for a timestep and y the actual value.
Then the predicted distribution CDF is $\mathbb{1}(x \geq \hat{y})$.

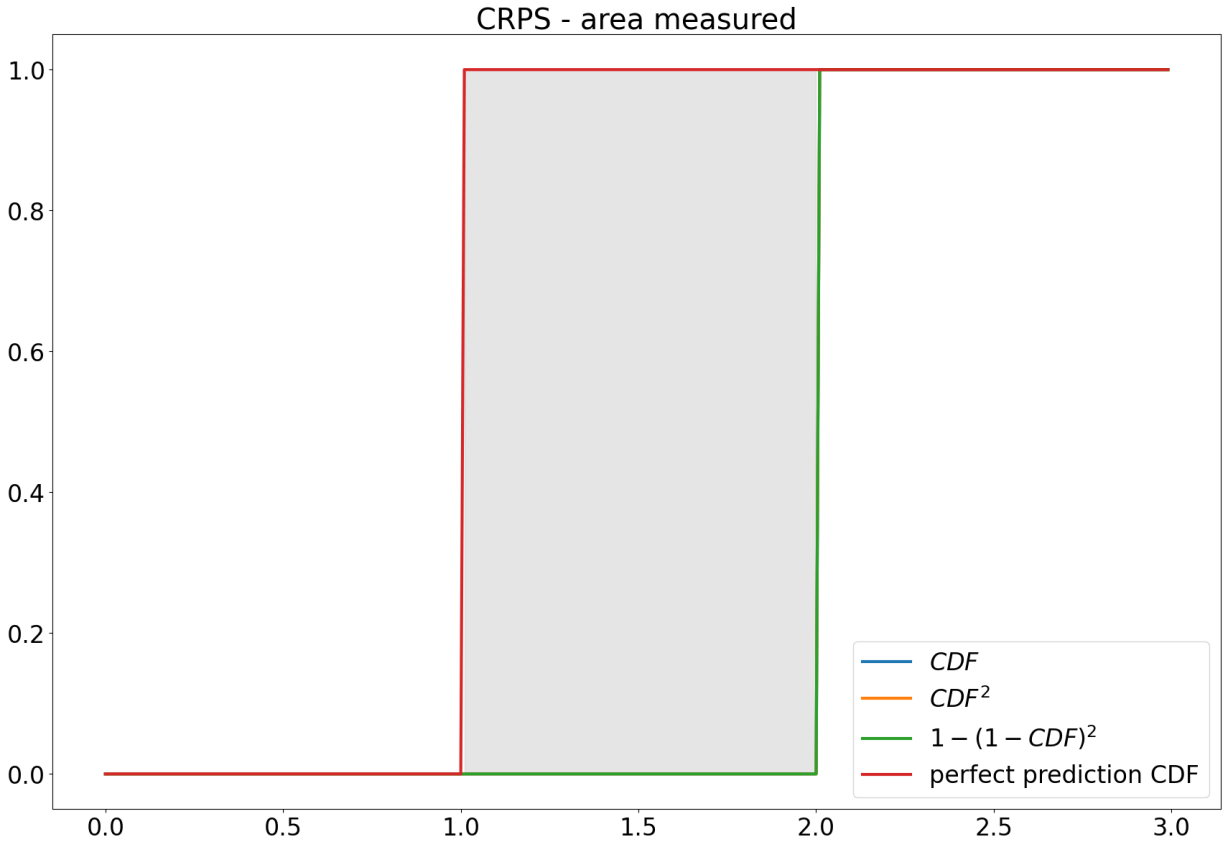
$$CRPS = \int_{-\infty}^{\infty} (\mathbb{1}(x \geq \hat{y}) - \mathbb{1}(x \geq y))^2 dx$$

if $y \geq \hat{y}$:

$$\begin{aligned} CRPS &= \int_{-\infty}^{\hat{y}} (\mathbb{1}(x \geq \hat{y}) - \mathbb{1}(x \geq y))^2 dx \\ &\quad + \int_{\hat{y}}^{y} (\mathbb{1}(x \geq \hat{y}) - \mathbb{1}(x \geq y))^2 dx \\ &\quad + \int_{y}^{\infty} (\mathbb{1}(x \geq \hat{y}) - \mathbb{1}(x \geq y))^2 dx \\ &= \int_{\hat{y}}^{y} (\mathbb{1}(x \geq \hat{y}) - \mathbb{1}(x \geq y))^2 dx \\ &= y - \hat{y} \end{aligned}$$

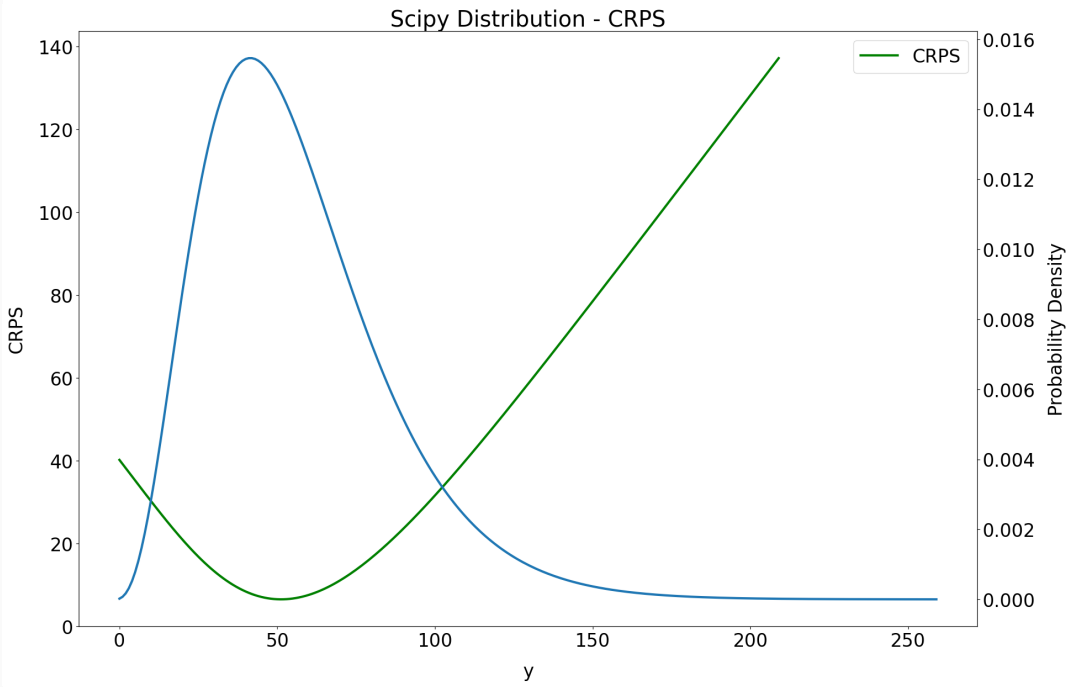
Similarly if $\hat{y} > y$, it can be shown that $CRPS = \hat{y} - y$.

Therefore $CRPS = |y - \hat{y}| = MAE$



CRPS – distributions

```python
def scipy_dist_crps(
    dist_list: List[rv_frozen],
    actuals: np.array,
    integration_lower_lim: float = -10000,
    integration_upper_lim: float = 10000,
    dx: float = 1,
) -> np.array:
    total_min = min(integration_lower_lim, np.min(actuals))
    total_max = max(integration_upper_lim, np.max(actuals))

    x_axis = np.arange(start=total_min, stop=total_max, step=dx)

    cdfs = [dist.cdf(x_axis) for dist in dist_list]

    # Perfect prediction cdf
    repeated_actuals = np.expand_dims(actuals, axis=1)
    perfect_prediction = 1 * x_axis >= repeated_actuals

    # Numerical integral calculation
    crps_vector = np.trapz(
      (cdfs - perfect_prediction) ** 2, x_axis, axis=1
    )
    return crps_vector
```

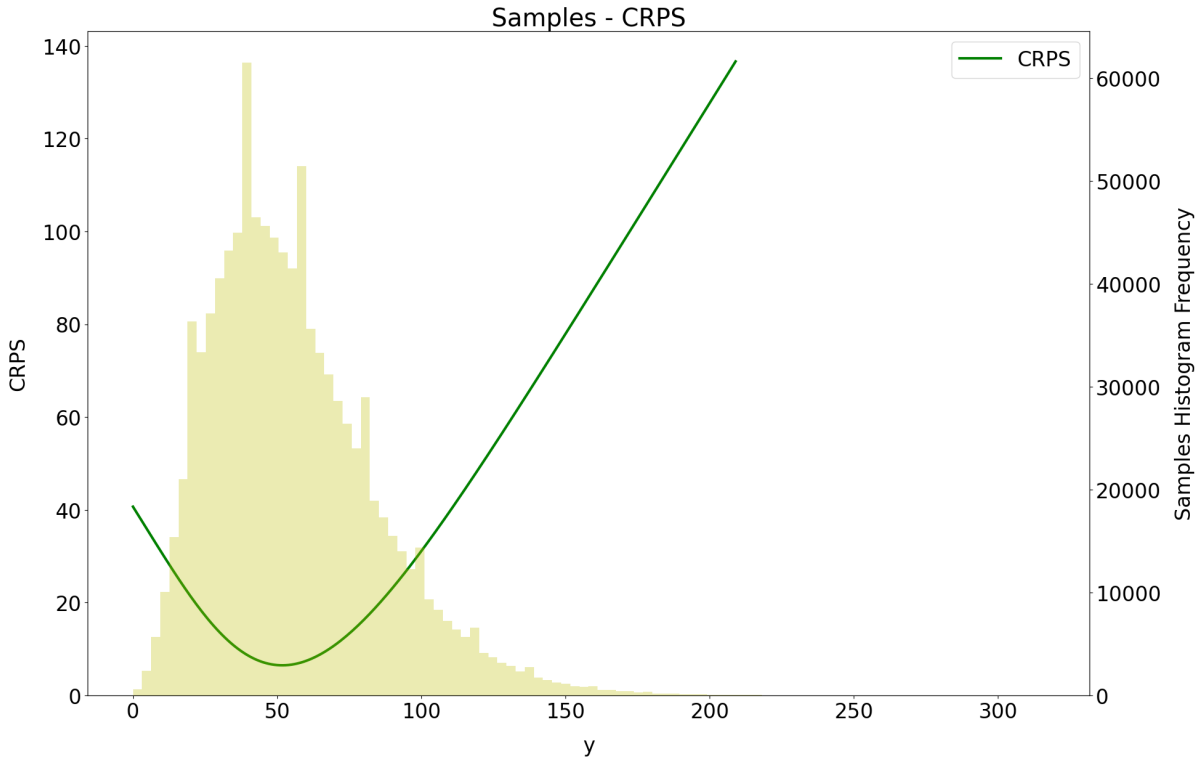


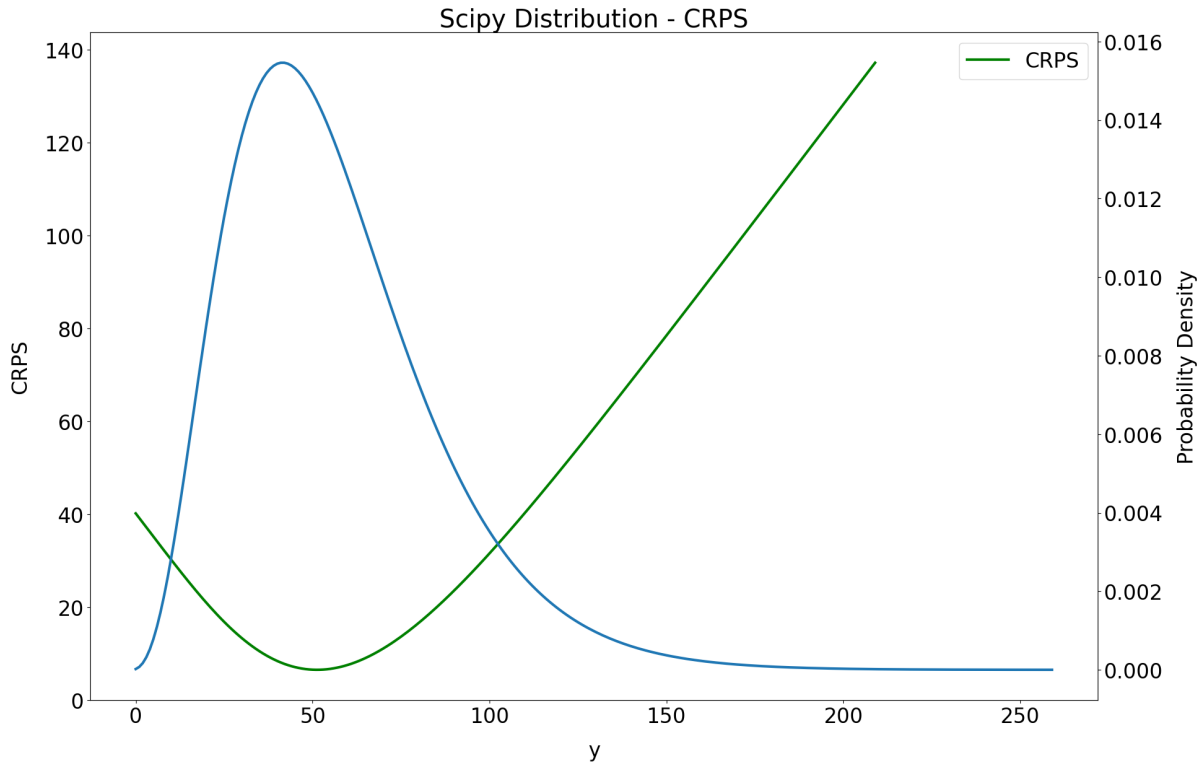
CRPS - samples

```python
def sample_crps(
    samples: np.array,
    actuals: np.array,
    integration_lower_lim: float = -10000,
    integration_upper_lim: float = 10000,
    dx: float = 1,
) -> np.array:
    sample_min = np.min(samples)
    sample_max = np.max(samples)

    total_min = min(
      sample_min, integration_lower_lim, np.min(actuals)
    )
    total_max = max(
      sample_max, integration_upper_lim, np.max(actuals)
    )

    pad_width_tuple = (
        int(np.abs(sample_min - total_min) / dx),
        int(np.abs(total_max - sample_max) / dx),
    )

    # T: number of time steps
    # N: number of samples in time step
    T, N = samples.shape

    # Sorted observations and rank (for empirical cdf)
    sorted_samples = np.sort(samples, axis=1)
    sorted_samples_rank_single = np.arange(N) / N
    sorted_samples_rank = np.tile(
      sorted_samples_rank_single, (T, 1)
    )
```

```python
    # Padding - necessary for the numerical integration
    padded_sorted = np.pad(
        sorted_samples,
        pad_width=[(0, 0), pad_width_tuple],
        mode="linear_ramp",
        end_values=(total_min, total_max),
    )

    padded_sorted_rank = np.pad(
        sorted_samples_rank,
        pad_width=[(0, 0), pad_width_tuple],
        mode="constant",
        constant_values=(0, 1),
    )

    # Perfect prediction cdf
    repeated_actuals = np.expand_dims(actuals, axis=1)
    perfect_prediction = 1 * padded_sorted >= repeated_actuals

    # Numerical integral calculation
    crps_vector = np.trapz(
      (padded_sorted_rank - perfect_prediction) ** 2,
      padded_sorted,
      axis=1
    )
    return crps_vector
```

CRPS - samples





But wait there is more!

The CRPS can be alternatively expressed as the quantile score integrated over all possible quantiles. Same holds for the Weighted Interval Score for the appropriate weights. From this definition it follows that with just a few intervals we can get a decent approximation of the CRPS using the Weighted Interval Score.

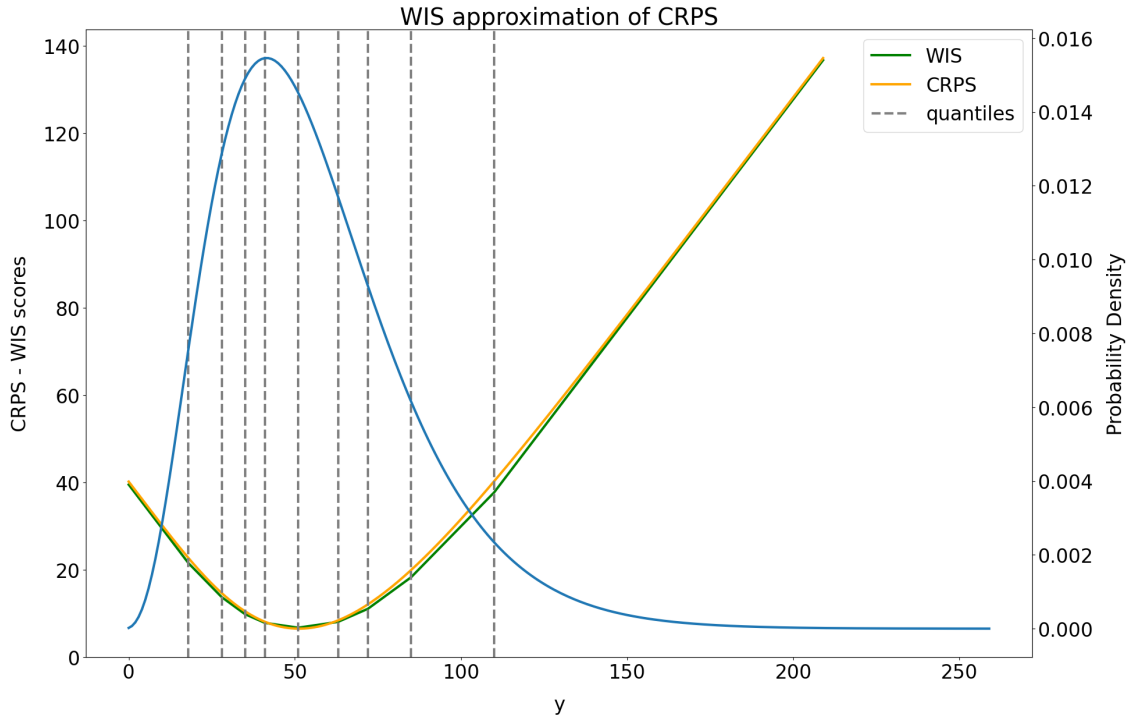
The Quantile Score is defined as

$$QS_\tau(F, y) = 2\{\mathbb{1}(y \leq q_\tau) - \tau\}(q_\tau - y)$$

where q_τ is the τ quantile of the forecast F and y is the observed outcome.

And CRPS can be expressed as

$$CRPS(F, y) = \int_0^1 QS_\tau(F, y) d\tau$$



- Bracher J, Ray EL, Gneiting T, Reich NG (2021) Evaluating epidemic forecasts in an interval format. PLOS Computational Biology 17(2): e1008618. https://doi.org/10.1371/journal.pcbi.1008618
- Gneiting T, Ranjan R. Comparing density forecasts using threshold- and quantile- weighted scoring rules. Journal of Business and Economic Statistics. 2011;29(3):411–422. doi:10.1198/jbes.2010.08110
- Gasthaus, J., Benidis, K., Wang, Y., Rangapuram, S.S., Salinas, D., Flunkert, V. & Januschowski, T.. (2019). Probabilistic Forecasting with Spline Quantile Function RNNs. Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research 89:1901-1910 Available from https://proceedings.mlr.press/v89/gasthaus19a.html.

Scaling the CRPS

As MAE and RMSE, CRPS is dependent on the scale of the time series. If you want to meaningfully aggregate and compare the CRPS for multiple time series, you will need to scale it somehow. A good strategy is to use a skill score utilizing the CRPS of a baseline forecast.

$$Skill\ Score = \frac{CRPS_{baseline} - CRPS_{candidate\ model}}{CRPS_{baseline}}$$

For a less computationally expensive scaling option I would use the same scaling factor as in MASE (Mean Absolute Scaled Error), namely the MAE of a one step Naive forecast in the training set but I would say that is this not as robust as the former scaling method.

- Jose, V.R., Nau, R.F., & Winkler, R.L. (2009). Sensitivity to Distance and Baseline Distributions in Forecast Evaluation. Manag. Sci., 55, 582–590.
- Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on July 2nd 2022.

Scaling the CRPS

As MAE and RMSE, CRPS is dependent on the scale of the time series. If you want to meaningfully aggregate and compare the CRPS for multiple time series, you will need to scale it somehow. A good strategy is to use a skill score utilizing the CRPS of a baseline forecast.

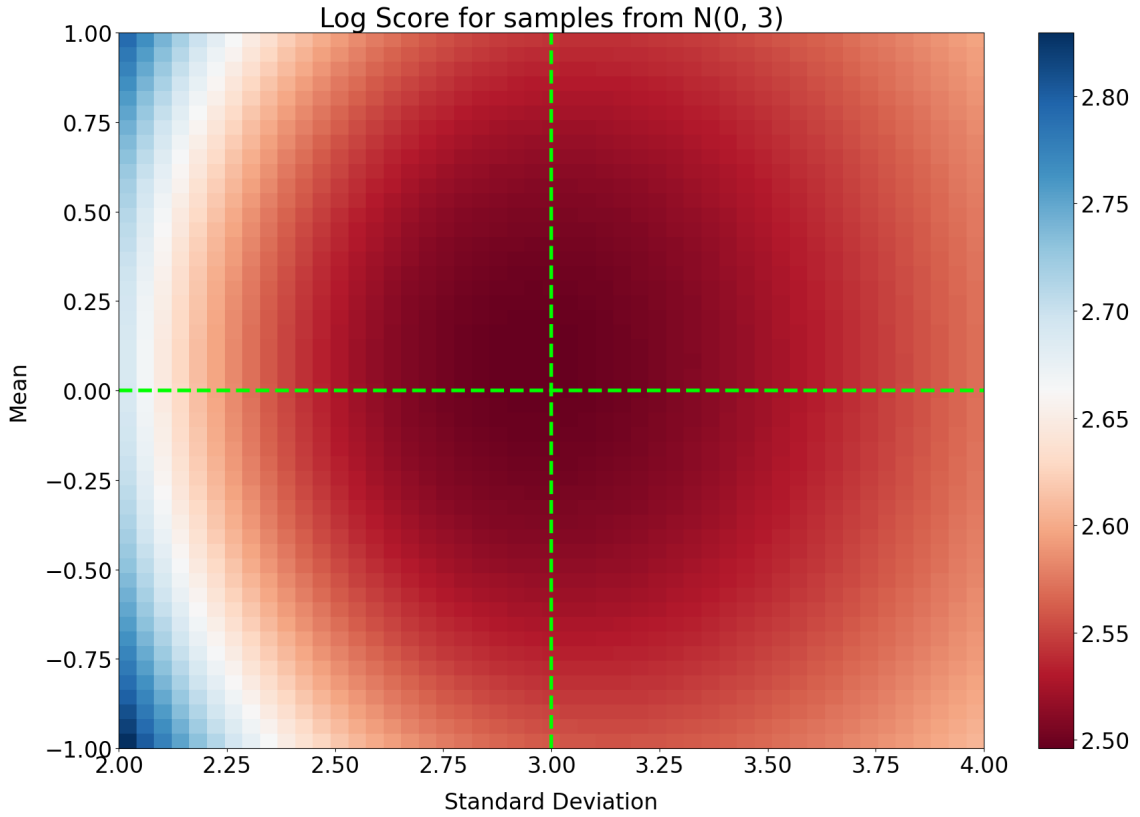
$$Skill\ Score = \frac{CRPS_{baseline} - CRPS_{candidate\ model}}{CRPS_{baseline}}$$

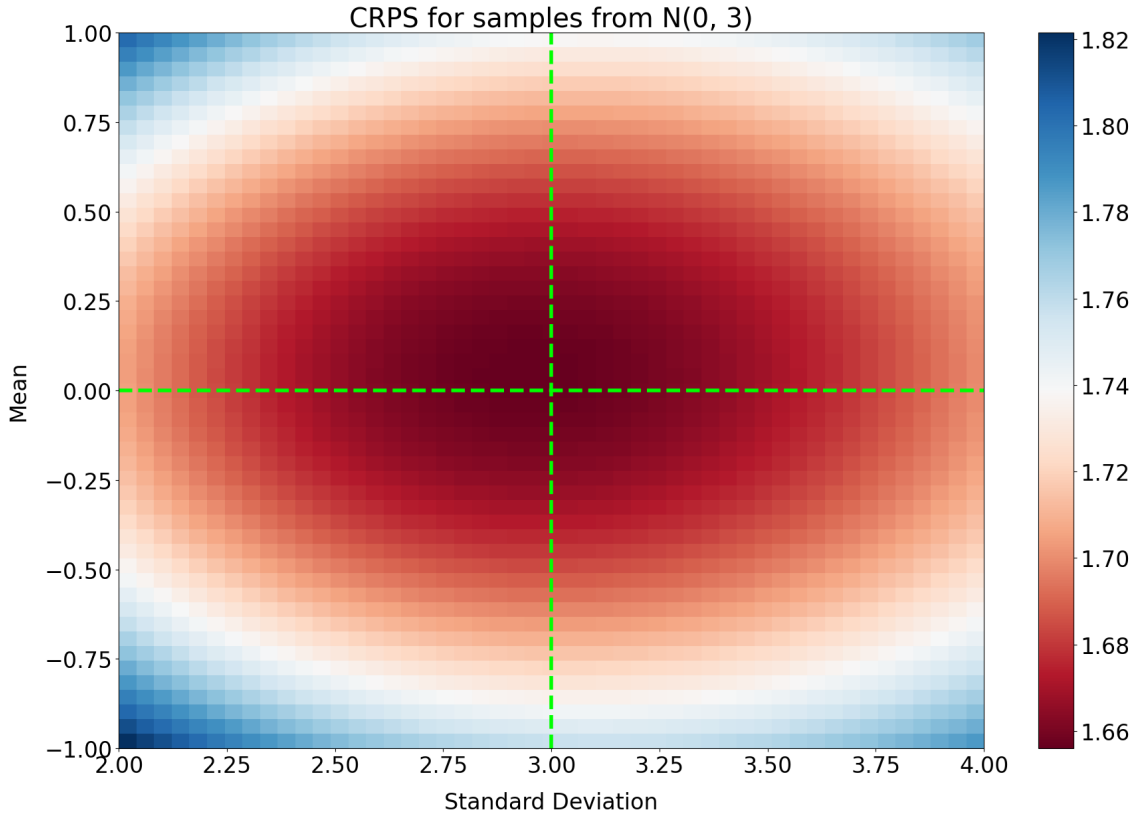
For a less com[illegible] I would use the same scaling factor as in MASE (Mean Absolute Scaled Error), namely [illegible]st in the training set but I would say that is this not as robust as the former scaling

- Jose, V.R., Nau, R.F., & Winkler, R.L. (2009). Sensitivity to Distance and Baseline Distributions in Forecast Evaluation. Manag. Sci., 55, 582–590.
- Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on July 2nd 2022.

Proper scoring property shown through simulations

Assume can sample from the **data generating distribution**, as many times we like. What will the average score be (approximating the expected value) for different forecasted distributions?





Local VS Distance Sensitive Scoring Rules

The Log Score and CRPS belong to 2 different classes of scoring rules.

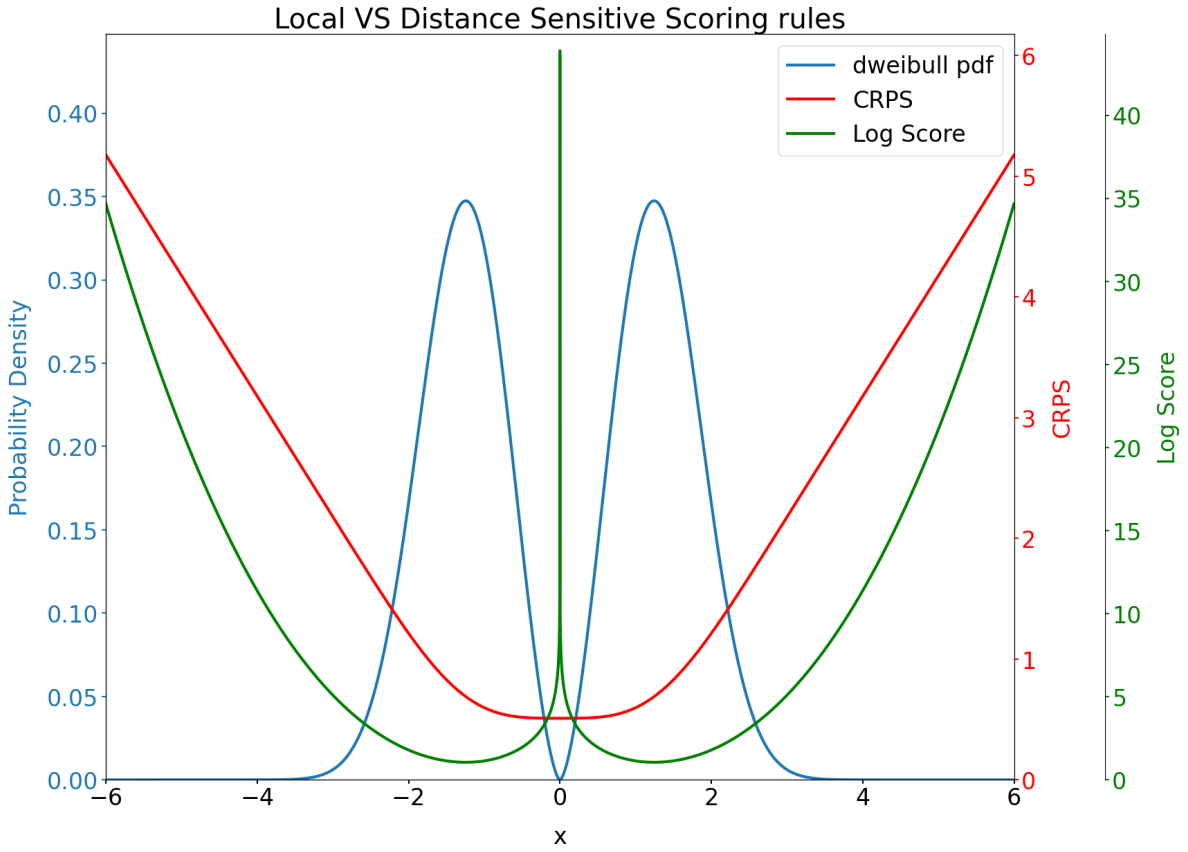
The Log Score is a **local scoring rule**, meaning that the value of the score for a particular observation depends only on the probability density function value for the value of that observation. This is clear from the definition of the log score, LogS(f, y) = - log(f(y)), where f the is PDF of the predicted distribution and f(y) is the value of the PDF for the observation.

CRPS is a **distance sensitive score**. In cases where the values have a meaningful ranking (e.g. toothpaste demand), we would usually prefer that our scoring rule takes into account the "distance" of the predicted distribution from the observed value and not just the assigned probability.

- Jose, V.R., Nau, R.F., & Winkler, R.L. (2009). Sensitivity to Distance and Baseline Distributions in Forecast Evaluation. Manag. Sci., 55, 582–590.
- Bernardo, J. M. (1979). Expected Information as Expected Utility. The Annals of Statistics, 7(3), 686–690. http://www.jstor.org/stable/2958753

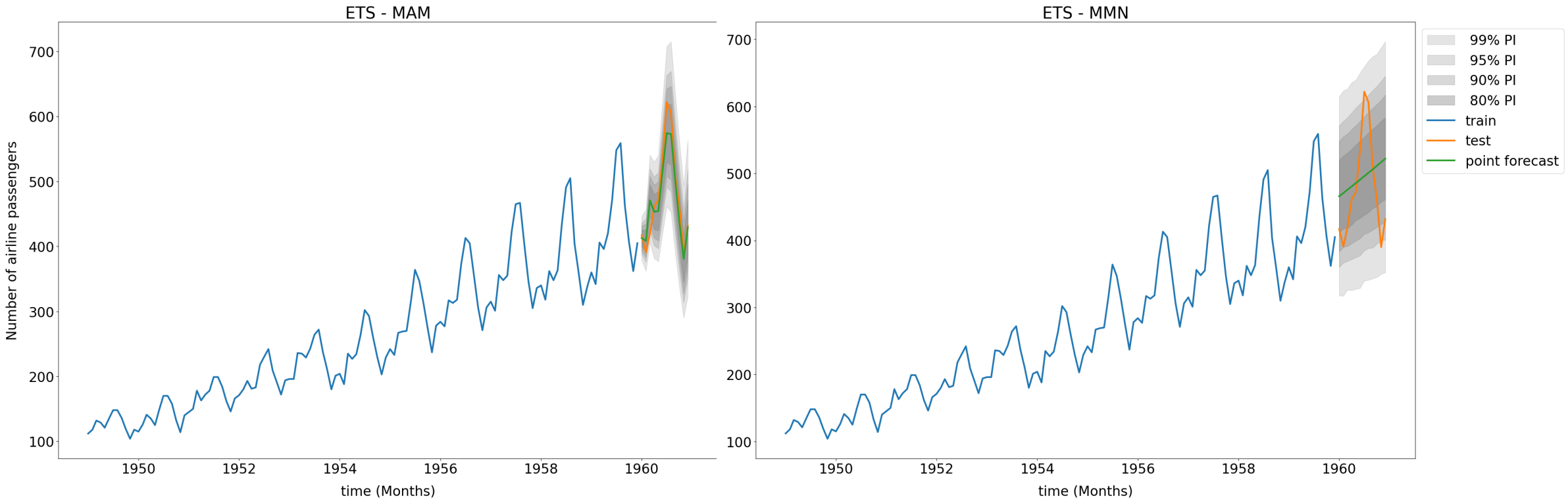
Local VS Distance Sensitive Scoring Rules

The differences between the 2 classes of scores are also very clear in the example below, where we're calculating the score for a symmetric bi-modal distribution (double Weibull). In this case the PDF on the median of the distribution is going to be very close to zero. As a result, **CRPS is minimised if the observation is equal to the median** of this predicted distribution and **the Log Score is maximised**, meaning that the 2 scores are in a big disagreement close to the median.



Using the LogScore and CRPS to evaluate forecasts

The statsmodels.tsa API has a simulate method that we can use to generate future sample paths from fitted models. Darts and gluonts can also serve probabilistic forecasts in the form of sampled paths.



The results!

	Point Forecast		Simulations	
	ETS - MAM	**ETS - MMN**	**ETS - MAM**	**ETS - MMN**
MAE	20.10	63.61	N/A	N/A
Mean CRPS	20.12	63.64	15.66	43.6
Mean LogS	N/A	N/A	4.83	5.74

Software Implementations of Scoring Rules

- properscoring - A python library with efficient implementations for CRPS and the Brier Score, leveraging numba for a significant speed-up. Unfortunately, the last release was in 2015.
- Pyro - probabilistic programming library built on Pytorch, has a neat implementation of empirical CRPS.
- gluonts - A popular time-series forecasting library by Amazon. Has very good (in some cases the original) implementations of probabilistic forecasting algorithms and has torch implementations of CRPS for a lot of torch distributions.
- xskillscore - an open-source project and Python package that provides verification metrics of deterministic (and probabilistic from properscoring) forecasts with xarray.
- nixtla - the new kids on the block! An ecosystem of packages with efficient implementations of the classic forecasting algorithms, deep-learning models and many more. They have an implementation of the multi-quantile loss, where it's limit over all possible quantiles approximates the CRPS.
- scoringutils - R library with a wide range of scoring rules.
- scoringRules - R library with scoring rules implementations in rcpp for better performance.

Recap

- We saw examples of how distributional forecasts can help us take decisions under uncertainty
- We had a quick overview of point forecasting metrics
- We introduced the proper scoring rules and their properties
- We saw how to compute the CRPS and the Log Score for both scipy distributions and future sampled paths
- Finally, we used the CPRS and the Log Score to select the best distributional forecast for the famous air passengers time series.

Thank you!

“It’s tough to make predictions, especially about the future”

Yogi Berra

References

1. Hewamalage, H., Ackermann, K. & Bergmeir, C. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Min Knowl Disc* (2022). https://doi.org/10.1007/s10618-022-00894-5

2. Tilmann Gneiting (2011) Making and Evaluating Point Forecasts, Journal of the American Statistical Association, 106:494, 746-762, DOI: 10.1198/jasa.2011.r10138

3. 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: 243-268. https://doi.org/10.1111/j.1467-9868.2007.00587.x

4. Tilmann Gneiting & Adrian E Raftery (2007) Strictly Proper Scoring Rules, Prediction, and Estimation, Journal of the American Statistical Association, 102:477, 359–378, DOI: 10.1198/016214506000001437

5. Bracher J, Ray EL, Gneiting T, Reich NG (2021) Evaluating epidemic forecasts in an interval format. PLOS Computational Biology 17(2): e1008618. https://doi.org/10.1371/journal.pcbi.1008618

6. Good, I.J. (1952), Rational Decisions. Journal of the Royal Statistical Society: Series B (Methodological), 14: 107–114. https://doi.org/10.1111/j.2517-6161.1952.tb00104.x

7. Matheson, J. E., & Winkler, R. L. (1976). Scoring Rules for Continuous Probability Distributions. Management Science, 22(10), 1087–1096. http://www.jstor.org/stable/2629907

8. Gneiting T, Ranjan R. Comparing density forecasts using threshold- and quantile- weighted scoring rules. Journal of Business and Economic Statistics. 2011;29(3):411–422. doi:10.1198/jbes.2010.08110

9. Gasthaus, J., Benidis, K., Wang, Y., Rangapuram, S.S., Salinas, D., Flunkert, V. & Januschowski, T.. (2019). Probabilistic Forecasting with Spline Quantile Function RNNs. Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research 89:1901-1910 Available from https://proceedings.mlr.press/v89/gasthaus19a.html.

10. Jose, V.R., Nau, R.F., & Winkler, R.L. (2009). Sensitivity to Distance and Baseline Distributions in Forecast Evaluation. Manag. Sci., 55, 582–590.

11. Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on October 10th 2022.

12. Bernardo, J. M. (1979). Expected Information as Expected Utility. The Annals of Statistics, 7(3), 686–690. http://www.jstor.org/stable/2958753