k-diagram: Technical Report (Derivations and Details)

Kouao Laurent Kouadio

***Abstract*—This technical report outlines the mathematical foundations of the `k-diagram` Python package. It explains the core principles of mapping statistical metrics onto a polar coordinate system, which underpins the package's innovative visualizations. Indeed, the formal definitions and equations are essential for generating key diagnostic plots, including those for the coverage of the prediction interval, the analysis of the error distribution, and the evaluation of the forecast horizon. The goal is to serve as a concise reference for users and developers who seek a deeper understanding of the theoretical foundations behind `k-diagram`.**

***Index Terms*—Probabilistic Forecasting, Uncertainty Quantification, Data Visualization, Polar Coordinates, Model Diagnostics, Technical Report.**

I. Introduction

THE `k-diagram` package introduces a suite of visualization tools, each designed to diagnose the performance of probabilistic forecasts with a level of granularity and interpretability that goes beyond conventional methods [1]. Traditional evaluation metrics frequently condense information into single summary scores; however, this approach can obscure critical structures within forecast uncertainty and error. In contrast, 'k-diagram' employs a polar coordinate system, thereby enabling the detection and interpretation of nuanced patterns that might otherwise remain hidden.

This document serves as a supplementary resource to the primary JOSS paper. Its primary aim is to outline the mathematical formulations that transform the raw forecast data and observations into the visual diagnostics provided by the package. Accordingly, I begin by laying out the general mapping framework, then proceed to a detailed explanation of the specific equations and methods that underpin the key diagnostic plots. This structured approach is intended to guide the reader from foundational concepts through to practical application, fostering both a rigorous and accessible understanding of `k-diagram`'s methodology.

II. Core Principles

The core advancement of `k-diagram` lies in mapping one-dimensional statistical properties onto a two-dimensional polar space, where each data point is defined by a radius (r) and an angle (θ). This approach provides a concise way to visualize forecast behavior that extends beyond what linear methods offer.

To clarify the setup, consider N observations $Y = \{y_1, y_2, \ldots, y_N\}$. For each y_i, the probabilistic forecast delivers a predictive distribution, often summarized through quantiles. Here, $y_i \in \mathbb{R}$ denotes the true value, while the point forecast $\hat{y}_i$ typically corresponds to the median (0.5-quantile). The prediction interval (PI) at nominal coverage $(1-\alpha) \times 100\%$ is $[L_i, U_i]$, given by the $\alpha/2$ and $1-\alpha/2$ quantiles. The forecast error is $e_i = y_i - \hat{y}_i$.

The methodology proceeds in two steps. First, for each instance i, a score S_i is computed to quantify error, interval width, or coverage. Next, S_i is mapped to the radius r_i, while either the instance index or a categorical variable determines the angle θ_i.

Typically, angles are spread uniformly over the chosen coverage $A_{cov} \in (0, 2\pi]$:

$$\theta_i = A_{cov} \cdot \frac{i-1}{N-1}, \quad \text{for } i = 1, \ldots, N \tag{1}$$

The radius r_i usually reflects the score S_i directly or in normalized form. Hence, this structure enables the visualization of complex forecast properties in a clear, interpretable manner.

III. Formulations for Key Diagnostic Plots

A. Coverage Evaluation

Coverage diagrams offer a direct assessment of whether each observation lies within its predicted interval. For each observation y_i and its associated prediction interval $[L_i, U_i]$, the coverage indicator c_i is defined using the indicator function $\mathbf{1}(\cdot)$ as follows:

$$c_i = \mathbf{1}\{L_i \le y_i \le U_i\} = \begin{cases} 1 & \text{if } y_i \in [L_i, U_i] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Consequently, each instance i is assigned an angle θ_i, and the value of c_i is visualized at that angle, typically with color to distinguish coverage status. The radius is commonly fixed for all instances in this plot, allowing the focus to be on coverage rather than magnitude. Furthermore, the empirical Prediction Interval Coverage Probability (PICP) is calculated as the mean of the c_i values as

$$\text{PICP} = \frac{1}{N} \sum_{i=1}^{N} c_i \tag{3}$$

This aggregate provides a concise measure of overall coverage accuracy.

K. L. Kouadio is with the School of Geosciences and Infophysics, Central South University, Changsha 410083, China, and the Hunan Key Laboratory of Exploration of Nonferrous Resources and Geological Hazards, Changsha, Hunan, 410083, China (e-mail: lkouao@csu.edu.cn).

This document serves as a technical appendix to the JOSS paper "k-diagram: Rethinking Forecasting Uncertainty via Polar-based Visualization."

Version v1.3.0, September 26, 2025.

B. Model Error Distribution (Polar Violin Plot)

The polar violin plot visualizes the full error distribution for each model within a designated angular sector. For a given model, the raw errors $\{e_i\}_{i=1}^N$ are mapped such that the radial axis represents error values, while the angular sector uniquely identifies the model. The density of errors is estimated non-parametrically using a kernel density estimator (KDE) [2]:

$$\hat{f}(e) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{e - e_i}{h}\right) \tag{4}$$

where K denotes the kernel function (typically Gaussian) and h is the bandwidth. The resulting width of the violin at radius $r = e$ is proportional to $\hat{f}(e)$, thus visually conveying the error distribution. A violin centered around $r = 0$ indicates an unbiased model, while systematic deviations reveal model bias and dispersion.

C. Forecast Horizon Drift

To explore how forecast uncertainty changes with increasing lead time, the forecast horizon drift diagram summarizes the mean interval width at each horizon. For each horizon j, we compute the mean interval width $\bar{w}_j$ as follows: let $w_{i,j} = U_{i,j} - L_{i,j}$ denote the interval width for instance i at horizon j, and N_j be the number of instances for that horizon. The mean width is then

$$\bar{w}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} w_{i,j} \tag{5}$$

Each forecast horizon j is represented by a unique angle θ_j in the polar plot, and a bar is drawn at this angle with a length proportional to $\bar{w}_j$. In this way, the diagram facilitates intuitive comparison of uncertainty evolution across multiple horizons.

D. Probabilistic Calibration (Polar PIT Histogram)

The calibration of probabilistic forecasts is evaluated using the Probability Integral Transform (PIT) [3], [2], which checks whether the PIT values are uniformly distributed. For each observation y_i, the PIT value p_i is calculated from the cumulative distribution function (CDF) F_i as

$$p_i = F_i(y_i) \tag{6}$$

A set of perfectly calibrated forecasts yields PIT values $\{p_i\}_{i=1}^N$ that are uniformly distributed on $[0, 1]$. To visualize this, the unit interval is divided into K bins, with each bin corresponding to an angular sector in the polar coordinate system. The score for a sector k is the frequency of PIT values in the k-th bin:

$$\text{Freq}(k) = \sum_{i=1}^{N} \mathbf{1}\left\{p_i \in \left[\frac{k-1}{K}, \frac{k}{K}\right)\right\} \tag{7}$$

The radius of the bar in each sector is set according to $\text{Freq}(k)$. Hence, a well-calibrated forecast yields bars of similar length, forming a circle, while deviations from uniformity indicate miscalibration.

IV. Relationship and Error Visualizations

This section introduces visual tools for directly comparing model predictions with observed values and for analyzing the structure of forecast errors in relation to other variables.

A. Actual vs. Predicted

The actual versus predicted plot provides a clear comparison between true values and the corresponding point predictions, such as the median forecast, within the polar coordinate system. Suppose we have N observations $Y = \{y_1, \ldots, y_N\}$ and matching predictions $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_N\}$. Each point i is assigned an angle θ_i according to its index and the angular coverage A_{cov}, as previously defined in Eq. 1. The radius is then set to either y_i or $\hat{y}_i$, so that both the actuals (y_i, θ_i) and predictions $(\hat{y}_i, \theta_i)$ are plotted as separate series on the same polar axis. To directly visualize the error, a radial line segment connects $\hat{y}_i$ and y_i at each angle θ_i, making the magnitude and direction of the forecast error $e_i = y_i - \hat{y}_i$ immediately apparent. Hence, this plot offers an intuitive and granular depiction of prediction accuracy for each instance.

B. Conditional Quantile

To examine how forecast uncertainty varies with the magnitude of the observed value, the conditional quantile plot visualizes the sequence of predicted quantiles as a function of the sorted true values. Let $\{y_i\}_{i=1}^N$ denote the true values and $\{\hat{y}_i^{(\tau_k)}\}_{k=1}^K$ the matrix of quantile predictions for each i. First, all points are sorted by y_i, yielding sorted indices j and corresponding pairs $\{y_{(j)}, \hat{y}_{(j)}^{(\tau_k)}\}$. The angle θ_j then maps the range of sorted true values onto the angular span A_{cov} as

$$\theta_j = \text{map}(y_{(j)}, [\min(y), \max(y)], [0, A_{cov}]) \tag{8}$$

At each θ_j, the radii are given by the predicted quantiles $\{\hat{y}_{(j)}^{(\tau_k)}\}$. Prediction intervals (for example, between $\tau = 0.05$ and $\tau = 0.95$) are visualized as shaded bands, and by nesting several such bands, the plot reveals how the predictive distribution's shape and width evolve as a function of the observed value. This approach is particularly effective for diagnosing heteroscedasticity and other systematic variations in forecast uncertainty.

C. Error and Residual Relationship

These diagnostic plots are instrumental in detecting systematic model deficiencies by exposing patterns within the forecast errors, or residuals. The central objective is to determine whether errors exhibit dependence on other variables, as such relationships highlight potential areas for model improvement. Consider the data set $\{y_i\}$ of true values, $\{\hat{y}_i\}$ of predictions, and the resulting errors $e_i = y_i - \hat{y}_i$. Because errors may be negative while polar radii must be non-negative, errors are shifted by an offset R_{offset}, defined as $R_{\text{offset}} = |\min(\{e_i\})|$ if $\min(\{e_i\}) < 0$, and zero otherwise. Each radius is computed as

$$r_i = e_i + R_{\text{offset}} \tag{9}$$

This transformation ensures that the zero-error reference is represented by a circle at radius R_{offset}. The angular coordinate is given by the variable of interest, allowing for two primary approaches. If the error is plotted against the true value y_i, the angle θ_i is mapped from y_i, revealing any dependence of error magnitude on actual outcomes and thus highlighting heteroscedasticity. Conversely, if the error is examined as a function of the predicted value $\hat{y}_i$, the angle is derived from $\hat{y}_i$, which may uncover model bias or misspecification. Sorting data by the chosen angular variable before plotting creates a continuous sweep, and an ideal model yields points scattered randomly about the zero-error circle.

D. Error Bands

To further diagnose systematic bias, error band visualizations plot forecast error as a function of another variable, such as time of day or seasonality, and separate the mean error (bias) from its variability (uncertainty). Given a set of errors $\{e_i\}$ and a corresponding angular variable $\{\alpha_i\}$, the data are grouped into K angular bins according to α_i. For each bin B_k, the mean error and standard deviation are calculated as

$$\mu_{e,k} = \frac{1}{|B_k|} \sum_{i \in B_k} e_i \tag{10}$$

$$\sigma_{e,k} = \sqrt{\frac{1}{|B_k| - 1} \sum_{i \in B_k} (e_i - \mu_{e,k})^2} \tag{11}$$

The mean error $\mu_{e,k}$ is then plotted as a line at the central angle of each bin, while a shaded band reflects the range $[\mu_{e,k} - n \cdot \sigma_{e,k},\ \mu_{e,k} + n \cdot \sigma_{e,k}]$, where n is a chosen multiplier. This representation makes both systematic bias and error volatility visually accessible; a reference circle at $r = 0$ indicates perfect calibration, so deviations from this baseline reveal departures from ideal model behavior.

V. Formulations for 2D and Dynamic Visualizations

Beyond one-dimensional diagnostics, `k-diagram` supports the visualization of two-dimensional distributions and dynamic processes, including vector fields and temporal evolution. These advanced methods rely on binning, kernel density estimation, and vector calculus, each adapted for the polar coordinate system to extract richer structural information from the data.

A. Polar Heatmap (2D Density)

The polar heatmap enables the exploration of joint distributions by visualizing the density of two variables mapped into polar coordinates. Suppose we have N data points, each described by a radial coordinate r_i and an angular value α_i. The raw angular data α_i is first mapped to the plotting angle θ_i over the desired angular coverage A_{cov}:

$$\theta_i = \text{map}(\alpha_i, [\min(\alpha), \max(\alpha)], [0, A_{\text{cov}}]) \tag{12}$$

The polar plane is then partitioned into $N_r \times N_\theta$ bins, where each bin (j, k) spans a radial interval $[R_j, R_{j+1}]$ and an angular interval $[\Theta_k, \Theta_{k+1}]$. For each bin, the score C_{jk} is defined as the count of points falling within its boundaries:

$$C_{jk} = \sum_{i=1}^{N} \mathbf{1}\{r_i \in [R_j, R_{j+1}] \wedge \theta_i \in [\Theta_k, \Theta_{k+1}]\} \tag{13}$$

where $\mathbf{1}(\cdot)$ is the indicator function. The matrix of bin counts C forms the basis of the heatmap, with color intensity reflecting the local density of observations. Hence, this approach highlights zones of concentration and dispersion in a manner that is visually intuitive and mathematically precise.

B. Polar Quiver (Vector Fields)

To represent vector fields in polar coordinates, the polar quiver plot depicts both the magnitude and direction of change at each point. For each anchor (r_i, θ_i), a vector is defined by its radial component Δr_i and angular component $\Delta\theta_i$. Each arrow in the plot is placed at (r_i, θ_i), with orientation and length determined by the vector $(\Delta r_i, \Delta\theta_i)$. The magnitude of each vector is given by

$$M_i = \sqrt{(\Delta r_i)^2 + (r_i \Delta\theta_i)^2} \tag{14}$$

where multiplication by r_i converts the angular change from radians to a proper length scale. Colors can be used to encode either the magnitude M_i or an additional scalar variable. This visualization, implemented in `matplotlib` [4], provides a dynamic view of gradients and flows, allowing users to discern both local and global patterns in the underlying process.

C. Velocity Diagram

The velocity diagram visualizes the temporal rate of change of a forecasted quantity for multiple locations or entities. For each of N locations, consider a time series of median forecasts $Q_{i,t}$ across M time steps, where i indexes location and t indexes time. The average velocity v_i for each location is calculated as the mean of the first differences:

$$v_i = \frac{1}{M-1} \sum_{t=1}^{M-1} (Q_{i,t+1} - Q_{i,t}) \tag{15}$$

In the resulting polar scatter plot, each location i is mapped to an angle $\theta_i = A_{\text{cov}} \cdot (i-1)/(N-1)$, ensuring that all locations are distributed evenly around the circle. The radius r_i encodes the normalized velocity value, which is computed as

$$r_i = \frac{v_i - \min(\{v_k\})}{\max(\{v_k\}) - \min(\{v_k\})} \tag{16}$$

for stable visualization. The color assigned to each point may represent an auxiliary metric, such as the mean absolute magnitude of the forecasted quantity over time, $\bar{Q}_i = \frac{1}{M} \sum_{t=1}^{M} |Q_{i,t}|$. Consequently, the diagram provides an integrated view of both the direction and magnitude of temporal change across locations.

D. Radial Density Ring

The radial density ring adapts the one-dimensional kernel density estimate (KDE) [5], [3] for polar visualization, offering an intuitive depiction of a variable's distribution. Let $X = \{x_1, \ldots, x_N\}$ be the data, which might represent direct measurements or derived statistics such as interval widths. The probability density $\hat{f}(x)$ is estimated via KDE, as previously defined. The radial axis corresponds to the value x, while the angular coordinate spans the full range A_{cov}. Since the density is constant with respect to angle, it is replicated to form a continuous ring. At each radius r, color encodes the normalized density $\hat{f}_{\text{norm}}(r) = \hat{f}(r)/\max(\hat{f})$, resulting in brightly colored concentric circles in high-density regions. This format enables at-a-glance assessment of distributional features, such as modality and skewness, in a compact polar representation.

E. Error Ellipses (2D Uncertainty)

The error ellipses plot conveys two-dimensional uncertainty, particularly when the magnitudes of uncertainty differ between the radial and angular directions. This visualization is especially useful for representing anisotropic uncertainty, such as in tracking or geospatial forecasting tasks. For each data point i, suppose the mean position is given in polar coordinates as $(\mu_{r,i}, \mu_{\theta,i})$, and the standard deviations of uncertainty are $\sigma_{r,i}$ in the radial and $\sigma_{\theta,i}$ (in radians) in the angular direction. An uncertainty ellipse is constructed for each location in a local Cartesian frame, with semi-axes scaled by a chosen multiplier n: the radial semi-axis is $a_i = n \cdot \sigma_{r,i}$, and the tangential semi-axis is $b_i = n \cdot (\mu_{r,i} \tan(\sigma_{\theta,i}))$, which is approximately $n \cdot \mu_{r,i}\sigma_{\theta,i}$ for small $\sigma_{\theta,i}$. The ellipse is then mapped to the polar axes by calculating its parametric path, rotating it by the mean angle $\mu_{\theta,i}$, and translating the center to $(\mu_{r,i}, \mu_{\theta,i})$ in global coordinates. This ensures that each ellipse accurately depicts both the magnitude and orientation of local 2D uncertainty.

VI. Comprehensive Model Evaluation

The package also provides tools for holistic model comparison, enabling the simultaneous assessment of multiple models across several performance metrics within a single, compact visualization.

A. Regression Performance

The regression performance plot, also known as a "Polar Performance Chart," provides a multidimensional comparison of regression models by visualizing their results across various performance metrics in a polar grouped bar chart. Consider J models evaluated according to K metrics, with S_{jk} denoting the raw score of model j on metric k. All metrics are transformed so that higher values consistently reflect better performance. For error-based metrics, such as MAE or RMSE, negative values are used, while standard metrics like R^2 retain their original sign. To ensure comparability, each metric is normalized independently to the range $[0, 1]$:

$$r_{jk} = \frac{S_{jk} - \min_{j'}(S_{j'k})}{\max_{j'}(S_{j'k}) - \min_{j'}(S_{j'k})} \tag{17}$$

This normalization assigns a score of 1 to the top-performing model and 0 to the lowest on each metric. For visualization, the polar axis is split into K angular sectors, each representing a metric with central angle $\Theta_k = A_{\text{cov}} \cdot (k-1)/K$. Within each sector, J grouped bars are drawn, each bar's height (radius) corresponding to r_{jk} and its offset marking the associated model. The chart is bounded by an inner "Worst Performance" ring at a radius of 0 and an outer "Best Performance" ring at a radius of 1, providing clear visual cues for model comparison. This structure enables the rapid identification of each model's strengths and weaknesses across the selected set of metrics [6].

B. Pinball Loss Plot

The pinball loss plot offers a detailed view of quantile forecast performance by visualizing the average pinball loss across the set of predicted quantiles [6], [2], [3]. This scoring rule is particularly well-suited for evaluating both the calibration and sharpness of probabilistic models. For a true value y and a quantile forecast $\hat{y}^{(\tau)}$ at level τ, the pinball loss L_τ is defined as

$$L_\tau(y, \hat{y}^{(\tau)}) = \begin{cases} (y - \hat{y}^{(\tau)})\tau & \text{if } y \geq \hat{y}^{(\tau)} \\ (\hat{y}^{(\tau)} - y)(1 - \tau) & \text{if } y < \hat{y}^{(\tau)} \end{cases} \tag{18}$$

This asymmetric penalty structure ensures that over- and under-predictions are treated according to the target quantile. For a dataset of N observations and K quantile levels $\{\tau_k\}_{k=1}^K$, the average pinball loss for each quantile is

$$\bar{L}_{\tau_k} = \frac{1}{N} \sum_{i=1}^{N} L_{\tau_k}(y_i, \hat{y}_i^{(\tau_k)}) \tag{19}$$

Each quantile level τ_k is mapped to angle $\theta_k = A_{\text{cov}} \cdot \tau_k$ in the polar plot, while the radius r_k is set to the corresponding average loss $\bar{L}_{\tau_k}$. This produces a characteristic U-shaped curve, where central quantiles typically have lower loss values, and tail quantiles reflect higher losses, thus highlighting the calibration profile and predictive sharpness of the model across the entire forecast distribution.

VII. Conclusion

This report has presented the mathematical formulations underlying the diagnostic visualizations in the `k-diagram` package. By systematically mapping statistical scores to polar coordinates, the package provides a comprehensive framework for dissecting and interpreting the complex behaviors exhibited by forecasting models. For details on implementation and practical usage, please consult the package documentation at https://k-diagram.readthedocs.io/ and the main JOSS publication [1].

REFERENCES

[1] K. L. Kouadio, "k-diagram: Rethinking forecasting uncertainty via polar-based visualization," 2025, pre-publication draft. [Online]. Available: https://github.com/earthai-tech/k-diagram

[2] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1986.

[3] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 69, no. 2, pp. 243–268, 2007.

[4] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[5] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Olsczewski, M. Blondel, F. Prejbal, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.