MultivCalibration: Assessing the calibration of multivariate probabilistic forecasts in R

Sam Allen
University of Bern
Oeschger Centre for Climate Change Research

Abstract

When predicting future events, it is common to issue forecasts that are probabilistic. For probabilistic forecasts to be useful, they must be calibrated, in the sense that they align statistically with the corresponding outcomes. Allen *et al.* (2023) introduce simple and interpretable checks for multivariate calibration that facilitate a more comprehensive understanding of how multivariate forecasts perform. This vignette reproduces the results in this paper, and demonstrates how the accompanying R package **MultivCalibration** allows these checks for multivariate calibration to be implemented in practice.

Keywords: forecast evaluation, probabilistic forecasting, calibration, multivariate forecasting, R.

1. Introduction

Suppose we issue an ensemble forecast $\mathbf{X} = (X_1, \dots, X_M)$ for a real-valued quantity Y. The forecast is said to be probabilistically calibrated if the ensemble members and the outcome are exchangeable. Calibration is a fundamental property that probabilistic forecasts must satisfy to be considered trustworthy. To assess the probabilistic calibration of a forecast, we can calculate the rank of the outcome among the ensemble members,

$$\text{rank}(Y; \mathbf{X}) = 1 + \sum_{i=1}^{M} \mathbf{1}\{X_i < Y\},$$

and check whether this rank is uniformly distributed on the set $\{1, \dots, M+1\}$ of possible ranks. Here, $\mathbf{1}\{\cdot\}$ is the indicator function, which is equal to one if the statement inside the brackets is true, and zero otherwise. In practice, we observe several realisations of Y and $\mathbf{X}$, and can calculate the rank for each forecast-observation pair. It is then common to display the observed ranks in a histogram. The ensemble forecast is probabilistically calibrated if this so-called rank histogram is flat. Otherwise, the shape of the histogram can be used to identify systematic errors in the forecasts: a $\cup$-shaped histogram suggests the observations are frequently either above or below all ensemble members, implying the forecasts are under-dispersed; a $\cap$-shaped histogram implies that the forecasts are over-dispersed; and a triangular histogram suggests that the forecasts tend to either over- or under-predict the outcome, indicative of a systematic forecast bias.

Due to the information they provide about forecast performance, rank histograms are wellestablished when evaluating ensemble forecasts. Multivariate adaptations have also been proposed to assess the calibration of multivariate ensemble forecasts, where we are interested in predicting several variables, time points, or locations simultaneously (e.g. Gneiting *et al.* 2008; Thorarinsdottir *et al.* 2016). Multivariate rank histograms introduce a so-called *pre-rank function* that transforms the ensemble members and observations to univariate values. A univariate rank histogram can then be constructed by calculating the ranks of the transformed observations among the transformed ensemble members. Proposed approaches differ in the choice of transformation. Allen *et al.* (2023) argue that any function $\rho : \mathbb{R}^d \to \mathbb{R}$ can be used as a pre-rank function (where d is the dimension of the multivariate forecasts and observations), and by choosing ρ to measure a particular univariate summary statistic of the multivariate vectors, the resulting histograms will be easy to interpret and provide useful information regarding the systematic errors that occur in the forecasts.

While we discuss calibration here in the context of forecast evaluation, the methods are also applicable when validating (multivariate) probabilistic models. In this case, the output of the model can be interpreted as a probabilistic forecast for the target variable, and we can then check the model fit using the methods discussed herein.

In this vignette, we document the **MultivCalibration** R package, which allows multivariate rank histograms to be applied in practice. The following section introduces multivariate probabilistic calibration. Section 3 then demonstrates how multivariate calibration can be assessed using **MultivCalibration**, before the package is used to reproduce the results in Allen *et al.* (2023). A simulation study is presented in Section 4, and these multivariate rank histograms are then used to evaluate probabilistic forecasts of gridded wind speed fields over Europe. A brief discussion of possible extensions is presented in Section 5.

2. Multivariate calibration

Suppose that our ensemble members and observations are now d-dimensional multivariate vectors, $\mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_M \in \mathbb{R}^d$ for $d > 1$. While univariate probabilistic calibration can be assessed using the rank of the observation among the ensemble members, the notion of a rank is not well-defined in higher dimensions. Instead, it is common to introduce a pre-rank function

$$\rho : \mathbb{R}^d \times \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{M \text{ times}} \to \mathbb{R}$$

that transforms the observation and the M ensemble members to univariate values. A multivariate forecast is said to be probabilistically calibrated with respect to a pre-rank function ρ if the rank of $\rho(\mathbf{Y})$ among $\rho(\mathbf{X}_1), \dots, \rho(\mathbf{X}_M)$ is uniformly distributed on the set $\{1, \dots, M+1\}$ of possible ranks. This can be assessed by calculating the rank of several realisations of $\rho(\mathbf{Y})$ among the corresponding transformed ensemble members, and plotting these ranks in a histogram. This histogram of the ranks of the transformed observations is typically called a multivariate rank histogram.

Gneiting *et al.* (2008) demonstrate that any pre-rank function can be chosen that is invariant to permutations of the final M arguments. Allen *et al.* (2023) remark that any function that does not depend on the final M elements is trivially invariant to permutations of them, and is thus a valid pre-rank function. These are termed *simple pre-rank functions.* If multivariate forecasts are auto-calibrated (see Tsyplakov 2013; Gneiting and Resin 2022, for details) then they are also probabilistically calibrated with respect to any simple pre-rank function. These simple pre-rank functions essentially choose a univariate summary statistic that quantifies some relevant characteristic of the multivariate observations, and then assess to what extent the forecasts are probabilistically calibrated when predicting this summary statistic.

2.1. Pre-rank functions

The choice of pre-rank function will determine how to interpret the resulting histogram. Any pre-rank function can be chosen depending on what information is of interest to the forecast users. If we are interested in predicting extreme events, we can choose a pre-rank function that quantifies the extremity of the multivariate forecast; if we are interested in the dependence between different dimensions, we can introduce a pre-rank function that quantifies this dependence. These can readily be employed in the framework above, allowing us to assess the calibration of predictions for these univariate quantities. Of course, information is lost when converting the multivariate forecasts and observations to univariate values. It is therefore generally recommended that several pre-rank functions are employed.

Different pre-rank functions have been proposed in the literature. Smith and Hansen (2004) and Wilks (2004) proposed using the inverse length of the minimum spanning tree of the set $\mathbf{X}_1, \dots, \mathbf{X}_M$ as a pre-rank function for $\mathbf{Y}$. The pre-rank of $\mathbf{X}_1$ is the inverse length of the minimum spanning tree of the set $\mathbf{Y}, \mathbf{X}_2, \dots, \mathbf{X}_M$, with the ensemble member of interest replaced by the observation, and similarly for the other ensemble members. This is also the case for the pre-rank functions introduced below.

Gneiting *et al.* (2008) introduced the *multivariate rank* as a pre-rank function,

$$\rho_{mv}(\mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_M) = 1 + \sum_{m=1}^{M} \mathbf{1}\{\mathbf{X}_m \preceq \mathbf{Y}\},$$

where $\mathbf{X}_m \preceq \mathbf{Y}$ signifies that $X_{m,j} \leq Y_j$ for all $j = 1, \dots, d$ with $\mathbf{X}_m = (X_{m,1}, \dots, X_{m,d}) \in \mathbb{R}^d$ for $m = 1, \dots, M$.

Thorarinsdottir *et al.* (2016) proposed the *average rank* along each dimension,

$$\rho_{av}(\mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_M) = \frac{1}{d} \sum_{j=1}^{d} \text{rank}(Y_j; X_{1,j}, \dots, X_{M,j}),$$

and the *band-depth rank*,

$$\rho_{bd}(\mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_M) = \frac{1}{d} \sum_{j=1}^{d} \left[M + 1 - \text{rank}(Y_j; X_{1,j}, \dots, X_{M,j})\right] \left[\text{rank}(Y_j; X_{1,j}, \dots, X_{M,j}) - 1\right].$$

This representation of the band-depth rank pre-rank function assumes that there are no ties between $Y_j, X_{1,j}, \dots, X_{M,j}$ for $j = 1, \dots, d$, though a more general formula exists for when this is not the case.

Knüppel *et al.* (2022) suggested using multivariate proper scoring rules as pre-rank functions, such as the energy score,

$$\rho_{es}(\mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_M) = \frac{1}{M} \sum_{m=1}^{M} \|\mathbf{X}_m - \mathbf{Y}\| - \frac{1}{2M^2} \sum_{m=1}^{M} \sum_{k=1}^{M} \|\mathbf{X}_m - \mathbf{X}_k\|,$$

where $\|\cdot\|$ denotes the Euclidean distance in $\mathbb{R}^d$.

Scheuerer and Hamill (2018) propose using the fraction of threshold exceedances,

$$\rho_{FTE}(\mathbf{Y};t) = \frac{1}{d}\sum_{j=1}^{d} \mathbf{1}\{Y_j > t\}$$

for some threshold $t \in \mathbb{R}$. In contrast to the pre-rank functions listed above, the fraction of threshold exceedances does not depend on all ensemble members, and is therefore a simple pre-rank function. In this case, we omit the ensemble members from the function arguments.

Allen *et al.* (2023) extend this to introduce several simple pre-rank functions. For example, the mean of the multivariate vectors could be used as a measure of the average behaviour across all dimensions,

$$\rho_{loc}(\mathbf{Y}) = \bar{Y} = \frac{1}{d}\sum_{j=1}^{d} Y_j.$$

Similarly, the spread of the multivariate vector can be quantified using the sample variance,

$$\rho_{sc}(\mathbf{Y}) = s^2_{\mathbf{Y}} = \frac{1}{d}\sum_{j=1}^{d} \left(Y_j - \bar{Y}\right)^2,$$

while the variogram quantifies the dependence between different dimensions,

$$\rho_{dep}(\mathbf{Y};h) = -\frac{\gamma_{\mathbf{Y}}(h)}{s^2_{\mathbf{Y}}},$$

for some lag $h \in \mathbb{N}$, where

$$\gamma_{\mathbf{Y}}(h) = \frac{1}{2(d-h)}\sum_{j=1}^{d-h} |Y_j - Y_{j+h}|^2$$

is an empirical variogram at lag h. The negative sign ensures that a larger value of ρ_{dep} indicates a larger dependence, in keeping with ρ_{loc} and ρ_{sc} above. This simplifies the interpretation of the resulting multivariate rank histograms.

Further details regarding these pre-rank functions can be found in Allen *et al.* (2023), where additional pre-rank functions are introduced when evaluating probabilistic spatial field forecasts.

2.2. Pre-rank functions for gridded objects

A particular example of a multivariate observation is a gridded object, such as a spatial field. In this case, the ensemble members and observations are matrices, rather than vectors, i.e. $\mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_M \in \mathbb{R}^{p\times q}$. These matrices can be unravelled to obtain vectors of length $d = p \times q$, but they additionally have some spatial structure that should be considered during evaluation.

The calibration of these gridded forecasts can similarly be assessed by introducing a pre-rank function

$$\rho : \mathbb{R}^{p\times q} \times \underbrace{\mathbb{R}^{p\times q} \times \cdots \times \mathbb{R}^{p\times q}}_{M \text{ times}} \to \mathbb{R}$$

that converts the matrices to univariate values.

While the pre-rank functions listed above can be applied to unravelled matrices, additional pre-rank functions can also be designed that incorporate the spatial structure. For example, the variogram pre-rank function can be extended to use spatial lags $\mathbf{h} \in \{0, \dots, p-1\} \times \{0, \dots, q-1\}$. Let $\mathcal{I} = \{1, \dots, p\} \times \{1, \dots, q\}$ represent a set of grid points, and define the empirical variogram of a field $\mathbf{Y} \in \mathbb{R}^{p \times q}$ at multivariate lag $\mathbf{h}$ as

$$\gamma_{\mathbf{Y}}(\mathbf{h}) = \frac{1}{2|\mathcal{I}(\mathbf{h})|} \sum_{\mathbf{j} \in \mathcal{I}(\mathbf{h})} |Y_{\mathbf{j}} - Y_{\mathbf{j}+\mathbf{h}}|^2,$$

where $\mathcal{I}(\mathbf{h}) = \{\mathbf{j} \in \mathcal{I} : \mathbf{j} + \mathbf{h} \in \mathcal{I}\}$, meaning the sum is over all grid points that are separated by the multivariate vector $\mathbf{h}$. This can be employed within the definition of $\rho_{dep}(\mathbf{Y}; h)$ to construct a gridded variogram pre-rank function.

This spatial variogram can also be used to generate a pre-rank function that quantifies the isotropy of the variogram. A variogram is said to be isotropic if it depends only on the distance between elements of the multivariate vector, and not on the direction between them. By introducing a pre-rank function that measures the isotropy of an empirical variogram, we can assess to what extent the multivariate ensemble forecasts reproduce the degree of (an)isotropy present in the observed outcomes.

One example of such a pre-rank function is

$$\rho_{iso}(\mathbf{Y}; h) = -\left\{ \left[\frac{\gamma_{\mathbf{Y}}((h,0)) - \gamma_{\mathbf{Y}}((0,h))}{\gamma_{\mathbf{Y}}((h,0)) + \gamma_{\mathbf{Y}}((0,h))} \right]^2 + \left[\frac{\gamma_{\mathbf{Y}}((h,h)) - \gamma_{\mathbf{Y}}((-h,h))}{\gamma_{\mathbf{Y}}((h,h)) + \gamma_{\mathbf{Y}}((-h,h))} \right]^2 \right\}.$$

This pre-rank function quantifies the squared distance between the variogram in the horizontal direction $\mathbf{h} = (h, 0)$ and the vertical direction $\mathbf{h} = (0, h)$, plus the squared distance between the variogram in the two diagonal directions $\mathbf{h} = (h, h)$ and $\mathbf{h} = (-h, h)$. Alternative pairs of lags could also be employed.

3. Examples

The **MultivCalibration** package has the functionality to compute the pre-rank functions listed above when evaluating multivariate forecasts in R. Pre-ranks can be obtained using the `get_prerank()` function

```
get_prerank(y, x, prerank, return_rank = TRUE, ...)
```

which takes as inputs an observation vector `y` of length d, and a matrix of ensemble members `x` with d rows and M columns; as above d is the dimension of the multivariate objects and M is the number of ensemble members. While `get_prerank()` therefore calculates the pre-rank corresponding to one multivariate forecast and observation, the `apply()` functions or `for` loops can be used to sequentially apply `get_prerank()` to multiple forecasts.

The argument `prerank` specifies which pre-rank function should be applied to the multivariate forecasts and ensemble members. This can either be a string corresponding to one of several in-built options, or it can be a user-specified function. The in-built pre-rank functions currently available are the multivariate rank (`prerank = "multivariate_rank"`), the average rank (prerank = "average_rank"), the band-depth rank (prerank = "band_depth"), the mean (prerank = "mean"), the variance (prerank = "variance"), the energy score (prerank = "energy_score"), the fraction of threshold exceedances (prerank = "fte_rank"), and the variogram (prerank = "variogram").

The argument return_rank is a logical that specifies whether the rank of the (pre-rank transformed) observation should be returned, rather than the vector of pre-ranks; the default is to return the rank, otherwise a named vector is returned containing the pre-ranks corresponding to the observation and each ensemble member.

For example, to calculate the average rank pre-rank for an observation vector and ensemble forecast, get_prerank() could be used as follows

```
d <- 5
M <- 7

# generate data from a standard multivariate normal distribution
y <- as.vector(mvrnorm(1, rep(0, d), diag(d)))
x <- t(mvrnorm(M, rep(0, d), diag(d)))

# return pre-ranks
get_prerank(y, x, prerank = "average_rank", return_rank = FALSE)

##  obs ens1 ens2 ens3 ens4 ens5 ens6 ens7
##  3.6  5.0  5.0  5.0  4.2  5.4  3.0  4.8

# return rank of the observation pre-rank
get_prerank(y, x, prerank = "average_rank")

## [1] 2
```

If prerank is a function, it should convert a vector of dimension d to a single numeric value. Checks are in place to ensure this is satisfied. The prerank function could also take additional inputs, in which case these inputs should be included as variable arguments in get_prerank().

For example, while the mean and variance of the multivariate vector are provided as in-built pre-rank functions, we may also want to assess the skewness of the forecasts and observations. To do so, we can define a custom pre-rank function to measure the skewness, and pass this as an input to get_prerank().

```
prerank <- function(z) mean((z - mean(z))^3)
get_prerank(y, x, prerank = prerank, return_rank = FALSE)

##     obs    ens1    ens2    ens3    ens4    ens5    ens6    ens7
## -0.1413 -1.8909  0.1753 -0.0952  0.0601 -0.3156  0.0311  0.5449
```

This can be generalised to the k-th central moment of the vector, in which case the pre-rank function depends on an additional parameter k. This can be included in get_prerank() as an additional argument.

```
prerank <- function(z, k) mean((z - mean(z))^k)
get_prerank(y, x, prerank = prerank, return_rank = FALSE, k = 3)

##     obs    ens1    ens2    ens3    ens4    ens5    ens6    ens7
## -0.1413 -1.8909  0.1753 -0.0952  0.0601 -0.3156  0.0311  0.5449
```

While `get_prerank()` assumes that `y` is a vector, pre-ranks are also available when the observations and ensemble members are matrices. The **MultivCalibration** additionally exports a function `get_prerank_gr()` that calculates pre-ranks corresponding to gridded objects.

```
get_prerank_gr(y, x, prerank, return_rank = TRUE, ...)
```

The input `y` is a numeric matrix with p rows and q columns, while the ensemble forecast `x` is an array of dimension (p, q, M).

The `prerank` argument can again be either a string corresponding to a list of in-built options for the pre-rank function, or a user-specified function. In addition to the pre-rank functions available for `get_prerank()`, the isotropy pre-rank function is also available (`prerank = "isotropy"`).

4. Simulation study

4.1. Multivariate Gaussian

Suppose observations are drawn from a multivariate normal distribution with mean vector $\mu = \mathbf{0}$ and covariance matrix Σ for which

$$\Sigma_{i,j} = \sigma^2 \exp\left(-\frac{|i-j|}{\tau}\right), \quad i, j = 1, \dots, d.$$

The parameter $\sigma^2 > 0$ controls the variance of the observations along each dimension, while $\tau > 0$ determines how quickly the correlation decays as the distance between the dimensions increases. In this sense, there is assumed to be an ordering of the variables, as is typically the case in a time series or spatial setting. We set $d = 10$, $\sigma^2 = 1$, and $\tau = 1$. Analogous conclusions are also drawn from other configurations.

For each observation, $M = 20$ ensemble members are drawn at random from a mis-specified multivariate normal distribution. We consider six possible mis-specifications, corresponding to under- and over-estimation of the mean vector μ, scale parameter σ^2, and correlation parameter τ.

```
d <- 10         # dimensions
n <- 100        # number of iterations (10000 is used in Allen et al. (2023))
M <- 20         # number of samples from the forecast distribution

sig2 <- 1       # variance parameter
tau <- 1        # correlation parameter
```

The observations are drawn from a multivariate normal distribution with the following mean vector (mu_y) and covariance matrix (Sig_y)

```
mu_y <- rep(0, d)
Sig_y <- outer(1:d, 1:d, function(i, j) sig2*exp(-abs(i - j)/tau))
y <- mvrnorm(n, mu = mu_y, Sigma = Sig_y)
```

Firstly, consider errors in the mean. Suppose that the forecasts are obtained from a multivariate normal distribution with the correct covariance matrix, but with mean vector $\mu = (-0.5, \dots, -0.5)$ (of length d).

```
mu_x <- rep(-0.5, d)
x <- replicate(M, mvrnorm(n, mu = mu_x, Sigma = Sig_y))
```

The resulting x is an array of dimension (n, d, M). We can use the get_prerank() function to extract the multivariate rank for different pre-rank functions. For example, consider the average rank pre-rank function applied to the first forecast-observation pair

```
get_prerank(y[1, ], x[1, , ], prerank = "average_rank")

## [1] 21
```

We can use sapply() to loop over all n observations. We repeat this for all pre-rank functions, and store the result in a data frame.

```
# specify weight matrix for variogram pre-rank function
w_mat <- matrix(as.numeric(abs(outer(1:d, 1:d, FUN = "-")) <= 1), nrow = d)

rank_df <- sapply(1:n, function(i) {
  mvr <- get_prerank(y[i, ], x[i, , ], prerank = "multivariate_rank")
  avr <- get_prerank(y[i, ], x[i, , ], prerank = "average_rank")
  bdr <- get_prerank(y[i, ], x[i, , ], prerank = "band_depth")
  esr <- get_prerank(y[i, ], x[i, , ], prerank = "energy_score")
  loc <- get_prerank(y[i, ], x[i, , ], prerank = "mean")
  var <- get_prerank(y[i, ], x[i, , ], prerank = "variance")
  vgr <- get_prerank(y[i, ], x[i, , ], w = w_mat, prerank = "variogram")
  ranks <- c(mvr, avr, bdr, esr, loc, var, vgr)
  names(ranks) <- c("mvr", "avr", "bdr", "esr", "loc", "var", "vgr")
  return(ranks)
})
rank_df <- data.frame(t(rank_df))

# display the observation pre-ranks for the first 10 forecast cases
head(rank_df, 10)

##    mvr avr bdr esr loc var vgr
```

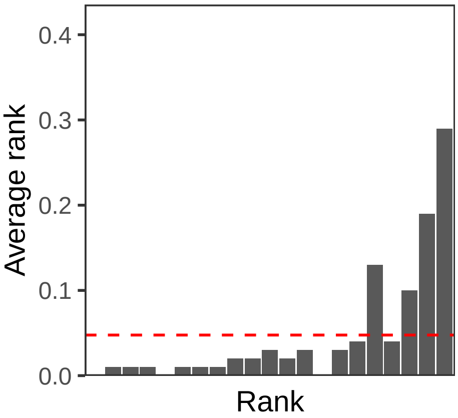

Figure 1: Multivariate rank histogram for the average rank pre-rank function in the multivariate normal simulation study.

```
## 1    21  21   1  20  21   6  10
## 2     4  17   1  20  17  20   2
## 3    14  20  15  12  20   6   5
## 4    19  19  21   1  17   1   1
## 5     5  18  17   7  17   3  20
## 6     4  21   8  18  21  19   7
## 7    19  19   7  15  19  14   8
## 8    21  20  10  21  21  15  13
## 9     8  19   5  19  21  10  13
## 10  20  19  12  19  20  13   3
```

To display the multivariate rank histograms corresponding to each pre-rank function, we first define a function `pit_hist`. This function is also available from the **WeightedForecastVerification** package on GitHub, which can be installed using **devtools**

```
devtools::install_github("sallen12/WeightedForecastVerification")
```

Consider the average rank as an example again. The multivariate rank histogram corresponding to this pre-rank function is displayed in Figure 1.

```
pit_hist(rank_df$avr, ylab = "Average rank", xticks = FALSE)
```

This can be repeated for all pre-rank functions, and for different types of errors. For example, when there is a positive bias in the forecasts, rather than a negative one: `mu_x = rep(0.5, d)`.

We can additionally see the behaviour of the multivariate rank histograms when we change the scale of the multivariate normal distribution.

```
sig2_x <- 0.85
Sig_x <- Sig_y*sig2_x
x <- replicate(M, mvrnorm(n, mu = mu_y, Sigma = Sig_x))
```

This can similarly be repeated using `sig2_x = 1.25` to analyse over-dispersion in the multivariate forecast distributions.

We can also make changes to the dependence structure. For example, consider $\tau = 0.5$ rather than 1.

```
tau_x <- 0.5
Sig_x <- outer(1:d, 1:d, function(i, j) sig2*exp(-abs(i - j)/tau_x))
x <- replicate(M, mvrnorm(n, mu = mu_y, Sigma = Sig_x))
```

This is also repeated using $\tau = 1.5$, as an example of when the dependence is too strong.

Repeating this for all pre-rank functions and all six types of misspecification, we can plot the multivariate rank histograms together.

4.2. Gaussian random fields

Now consider a second simulation study in which the forecasts and observations are gridded fields rather than multivariate vectors, with $p = q = 30$. This extends the previous example to a higher dimensional setting in which there is additionally spatial structure present in the data. The observations are drawn from a zero-mean Gaussian random field with an exponential covariance function such that the covariance between two locations $\mathbf{i}$ and $\mathbf{j}$ on the grid is

$$\sigma^2 \exp\left(-\frac{||\mathbf{i}-\mathbf{j}||}{\tau}\right), \quad \mathbf{i},\mathbf{j} \in \{1,\dots,30\} \times \{1,\dots,30\}.$$

We can use the **geoR** package to obtain realisations of a Gaussian random field with these parameters. An example field is shown in Figure 3.

```
d <- 30^2
n <- 10
M <- 20
sig2 <- 1
tau <- 1

y <- grf(d, grid = "reg", cov.pars = c(sig2, tau), nsim = n, messages = F)
```

Forecasts are then generated from mis-specified Gaussian random fields. We again consider six different types of mis-specification, corresponding to the scale, correlation, and isotropy of the random fields. For example, we can obtain ensemble members that under-estimate the scale in the observation fields using

```
sig2_x <- 0.85
x <- grf(d, grid = "reg", cov.pars = c(sig2_x, tau), nsim = n*M, messages = F)
x <- x$data
dim(x) <- c(sqrt(d), sqrt(d), M, n)
```

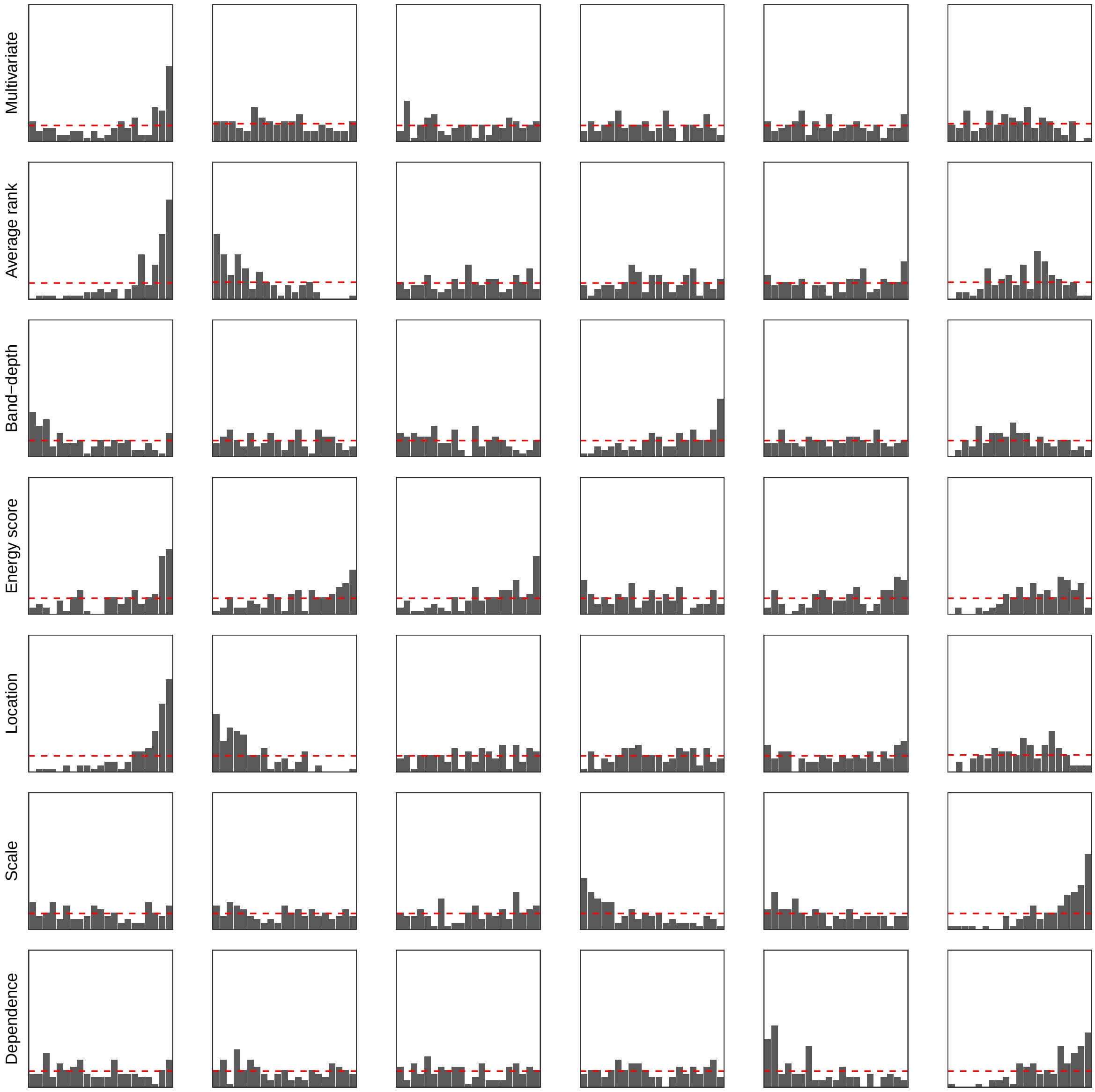

Figure 2: Multivariate rank histogram for seven pre-rank functions and the six types of mis-specification in the multivariate forecast distributions.

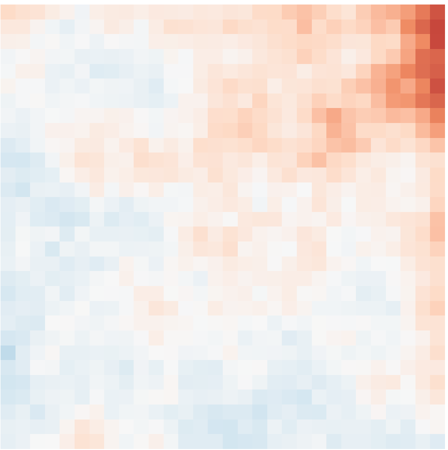

Figure 3: Example realisation of a Gaussian random field.

We can use the `get_prerank_gr()` function to extract the multivariate rank for different pre-rank functions, and use `sapply()` to loop over all n observations. Note that the variogram, fraction of threshold exceedances, and isotropy pre-rank functions require additional arguments corresponding to the threshold or spatial lag(s).

```
t <- 1
h <- rbind(c(0, 1), c(1, 0), c(1, 1), c(1, -1))

rank_df <- sapply(1:n, function(i) {
  avr <- get_prerank_gr(y[, , i], x[, , , i], prerank = "average_rank")
  bdr <- get_prerank_gr(y[, , i], x[, , , i], prerank = "band_depth")
  loc <- get_prerank_gr(y[, , i], x[, , , i], prerank = "mean")
  var <- get_prerank_gr(y[, , i], x[, , , i], prerank = "variance")
  vgr <- get_prerank_gr(y[, , i], x[, , , i], h = h, prerank = "variogram")
  fte <- get_prerank_gr(y[, , i], x[, , , i], t = t, prerank = "FTE")
  iso <- get_prerank_gr(y[, , i], x[, , , i], prerank = "isotropy")
  ranks <- c(avr, bdr, loc, var, vgr, fte, iso)
  names(ranks) <- c("avr", "bdr", "loc", "var", "vgr", "fte", "iso")
  return(ranks)
})
rank_df <- data.frame(t(rank_df))
```

We can similarly calculate the pre-ranks when the ensemble forecasts over-estimate the scale of the observed fields, and when there are errors in the correlation structure,

and the isotropy.

```
# rescale the fields vertically by a factor of 5/4
x <- grf(d, grid = "reg", cov.pars = c(sig2, tau), aniso.pars = c(0, 5/4),
         nsim = n*M, messages = F)
```

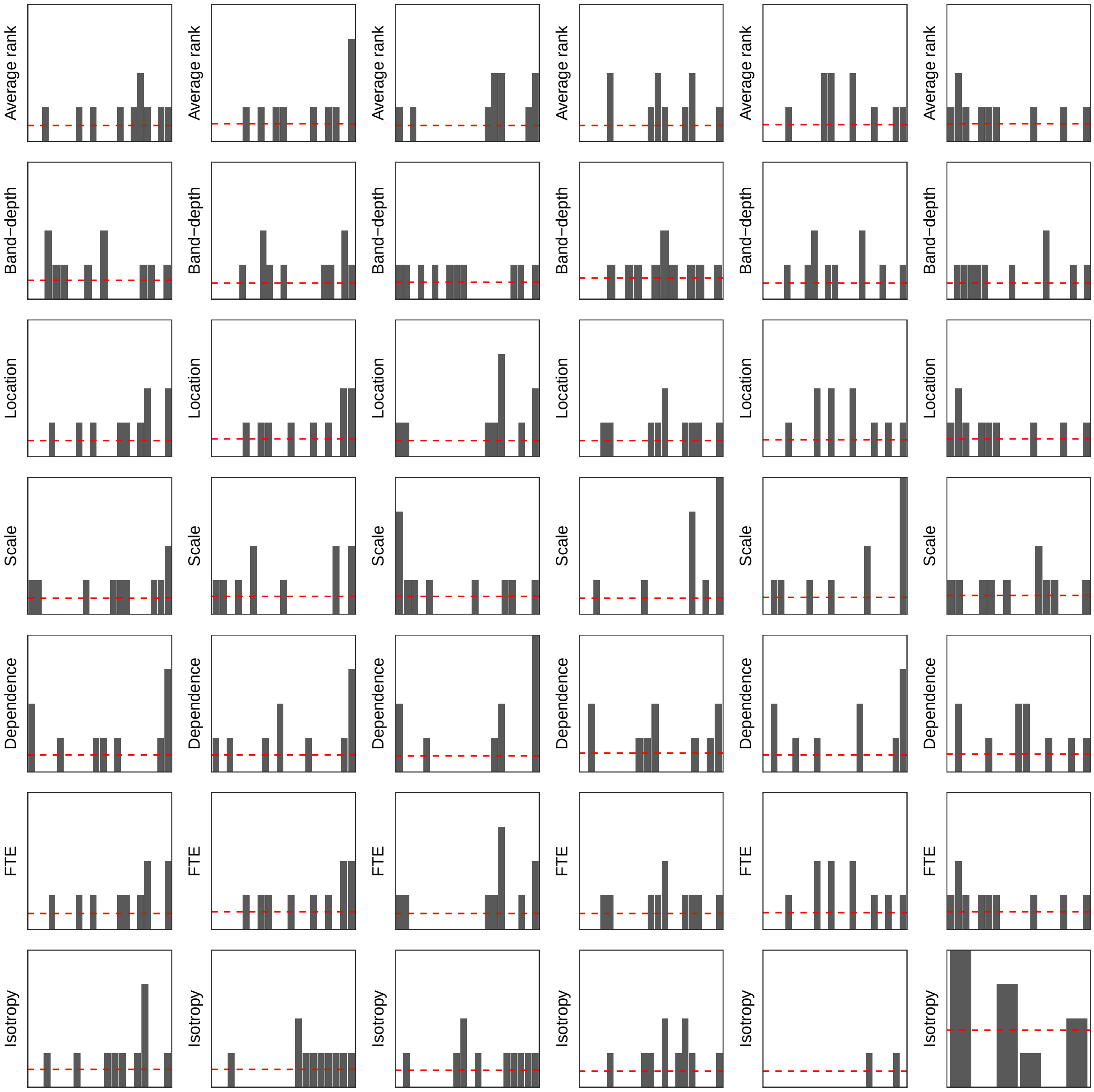

Figure 4: Multivariate rank histogram for seven pre-rank functions and the six types of mis-specification in the forecast fields.

The resulting multivariate rank histograms are displayed in Figure 4.

5. Discussion

This vignette discusses the R package **MultivCalibration**, which facilitates the assessment of multivariate probabilistic forecasts. The package consists of several pre-rank functions that can be used to construct multivariate rank histograms, allowing users to visualise the calibration of multivariate forecasts. To demonstrate the usage of the package, it is used to reproduce the results in Allen *et al.* (2023).

The package contains pre-rank functions previously proposed in the literature, including the multivariate rank of Gneiting *et al.* (2008), the average rank and band-depth rank of Thorarinsdottir *et al.* (2016), and a collection of simple pre-rank functions listed in Allen *et al.* (2023). There is also the option for users to employ custom pre-rank functions that can extract user-specific information about multivariate forecast performance. Additional pre-rank functions could additionally be made available in the future, including the minimum spanning tree-based pre-rank function proposed by Smith and Hansen (2004) and Wilks (2004).

The package is still in development, and several other extensions could also be included. The package currently contains pre-rank functions suitable for multivariate forecasts and observations, though the same framework can readily be applied when assessing the calibration of forecasts for other objects, such as networks or graphs. Pre-rank functions could therefore be introduced for forecasts in this form. Furthermore, while the package allows for user-specified pre-rank functions, these custom pre-rank functions must be simple. There is currently not the functionality to employ custom pre-rank functions that are not simple.

References

Allen S, Ziegel J, Ginsbourger D (2023). "Assessing the calibration of multivariate probabilistic forecasts." *arXiv preprint arXiv:2307.05846.*

Gneiting T, Resin J (2022). "Regression diagnostics meets forecast evaluation: Conditional calibration, reliability diagrams, and coefficient of determination." *arXiv preprint arXiv:2108.03210.*

Gneiting T, Stanberry LI, Grimit EP, Held L, Johnson NA (2008). "Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds." *Test*, **17**, 211–235.

Knüppel M, Krüger F, Pohle MO (2022). "Score-based calibration testing for multivariate forecast distributions." *arXiv preprint arXiv:2211.16362.*

Scheuerer M, Hamill TM (2018). "Generating calibrated ensembles of physically realistic, high-resolution precipitation forecast fields based on GEFS model output." *Journal of Hydrometeorology*, **19**, 1651–1670.

Smith LA, Hansen JA (2004). "Extending the limits of ensemble forecast verification with the minimum spanning tree." *Monthly Weather Review*, **132**, 1522–1528.

Thorarinsdottir TL, Scheuerer M, Heinz C (2016). "Assessing the calibration of high-dimensional ensemble forecasts using rank histograms." *Journal of Computational and Graphical Statistics*, **25**, 105–122.

Tsyplakov A (2013). "Evaluating density forecasts: a comment." *Available at SSRN 1907799.*

Wilks DS (2004). "The minimum spanning tree histogram as a verification tool for multidimensional ensemble forecasts." *Monthly Weather Review*, **132**, 1329–1340.

Affiliation:

Sam Allen
University of Bern
Institute of Mathematical Statistics and Actuarial Science
Alpeneggstrasse 22
3012 Bern, Switzerland
E-Mail: sam.allen@unibe.ch
and
Oeschger Centre for Climate Change Research