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 well-

established 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.

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 in Section 5. A brief discussion of possible extensions is presented in Section 6. For efficiency, results are presented for a subset of the data considered in Allen *et al.* (2023), but the exact results therein can easily be generated by changing the parameters in Sections 4 and 5.

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.

2. Multivariate calibration

Suppose that our ensemble members and observations are now d-dimensional multivariate vectors, $\mathbf{Y}, \mathbf{X}_1, \ldots, \mathbf{X}_M \in \mathbb{R}^d$ 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 $\rho(\mathbf{Y}, \mathbf{X}_1, \ldots, \mathbf{X}_M)$ among $\rho(\mathbf{X}_1, \mathbf{Y}, \mathbf{X}_2, \ldots, \mathbf{X}_M), \ldots, \rho(\mathbf{X}_M, \mathbf{Y}, \mathbf{X}_1, \ldots, \mathbf{X}_{M-1})$ is uniformly distributed on the set $\{1, \ldots, M+1\}$ of possible ranks. This can be assessed by calculating the rank of several realisations of $\rho(\mathbf{Y}, \mathbf{X}_1, \ldots, \mathbf{X}_M)$ 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 observation; 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.

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) proposed 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) extended 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,

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

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

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

while adaptations of the variogram can quantify the dependence between different dimensions,

$$\rho_{dep}(\mathbf{Y};w) = -\frac{\sum_{i=1}^{d}\sum_{j=1}^{d} w_{i,j}|Y_i - Y_j|^2}{s^2_{\mathbf{Y}}},$$

for some non-negative weight matrix $w \in \mathbb{R}^{d\times d}$. The negative sign ensures that a larger value of ρ_{dep} indicates a larger dependence, in keeping with ρ_{mn} and ρ_{var} above. This simplifies the interpretation of the resulting multivariate rank histograms.

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. $\mathbf{Y}, \mathbf{X}_1, \ldots, \mathbf{X}_M \in \mathbb{R}^{p\times q}$. 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 a weight matrix w that is a function of spatial lags.

The 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.

Let $\mathbf{h} \in \{0, \dots, p-1\} \times \{0, \dots, q-1\}$ denote a multivariate lag, and let $\mathcal{I} = \{1, \dots, p\} \times \{1, \dots, q\}$ represent a set of grid points. We define the empirical variogram of $\mathbf{Y} \in \mathbb{R}^{p \times q}$ 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 $\mathcal{I}(\mathbf{h}) = \{\mathbf{j} \in \mathcal{I} : \mathbf{j} + \mathbf{h} \in \mathcal{I}\}$ and $\#\mathcal{I}$ is its cardinality, meaning the sum is over all grid points that are separated by the multivariate vector $\mathbf{h}$.

One example of a pre-rank function to quantify isotropy is

$$\rho_{iso}(\mathbf{x}; h) = -a(h) \left[\gamma_{\mathbf{x}}((h, 0)) - \gamma_{\mathbf{x}}((0, h))\right]^2,$$

where

$$a(h) = \left[\frac{2\gamma_{\mathbf{x}}((h,0))^2}{\#\mathcal{I}((h,0))} + \frac{2\gamma_{\mathbf{x}}((0,h))^2}{\#\mathcal{I}((0,h))}\right]^{-1}$$

is a weight that accounts for the uncertainty in the variogram. This pre-rank function quantifies the (standardised) squared distance between the variogram in the horizontal direction $\mathbf{h} = (h, 0)$ and the vertical direction $\mathbf{h} = (0, h)$ at a chosen lag h. Alternative lags and directions 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"`), 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
```

Some in-built pre-rank functions also require variable arguments. For example, to calculate the fraction of threshold exceedances, the user must specify a threshold `t`, a single numeric value. Similarly, the variogram pre-rank function requires a lag `h` or a weight matrix `w`. The lag can either be a single integer, or a vector of integers, in which case the variogram is calculated for all integers in the vector, and then summed to return a single value. The weight matrix must be symmetric and contain only non-negative values.

While `get_prerank()` assumes that `y` is a vector, pre-rank functions are also available when the observations and ensemble members are matrices. The **MultivCalibration** package 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"`).

In this case, the variogram pre-rank function requires an additional argument `h` that is a vector of length two. This denotes the spatial lag at which the variogram should be calculated. If `h` is a matrix with two columns, then the variogram is computed for all rows of this matrix, and these are again summed to return a single value. Alternatively, a 4-dimensional array with dimensions (p, q, p, q) can be specified, which contains the weights assigned to each pair of coordinates. The isotropy pre-rank function also depends on a parameter `h` that can be specified by the user. This must be a single integer, denoting the lag at which the variogram differences are calculated; by default, the isotropy pre-rank function uses `h = 1`.

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

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

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. 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 $\mu = (-0.25, \ldots, -0.25)$ (d times).

```
mu_x <- rep(-0.25, 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.

```
w_mat <- exp(-abs(outer(1:d, 1:d, FUN = "-"))) # weight matrix for variogram

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, , ], prerank = "variogram", w = w_mat)
  ranks <- c(mvr, avr, bdr, esr, loc, var, vgr)
  names(ranks) <- c("mvr", "avr", "bdr", "esr", "loc", "var", "vgr")
  return(ranks)
```

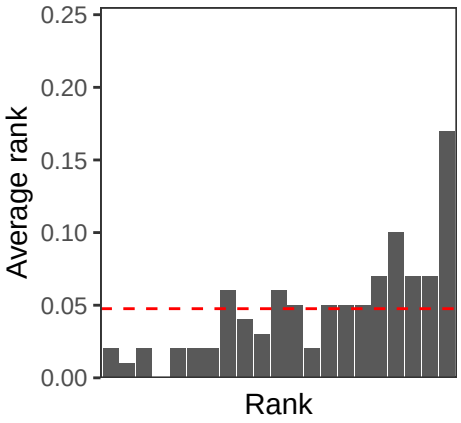

Figure 1: Multivariate rank histogram for the average rank pre-rank function in the multivariate normal simulation study.

```
})
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
## 1   21  21   3  18  21   3   8
## 2    4  16   2  19  16  20  18
## 3   14  16  15   8  16   7  12
## 4   18  10  21   1  13   1  18
## 5    5  15  16   5  15   3   1
## 6    4  17  14  17  19  19  18
## 7    4  14   7  13  18  14  21
## 8   13  20   3  19  20  16   7
## 9    8  17  19  17  18  11  13
## 10  18  19   5  15  16  13  21
```

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(0.25, 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 $\tau = 2$, as an example of when the dependence is too strong.

Having repeated this for all pre-rank functions and all six types of misspecification, Figure 2 displays the multivariate rank histograms corresponding to all pre-rank functions and misspecifications.

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 $p = q = 10$. 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

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

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.

```
p <- 10          # 30 is used in Allen et al. (2023)
q <- 10          # 30 is used in Allen et al. (2023)
d <- p*q
n <- 100         # 10000 is used in Allen et al. (2023)
M <- 20
sig2 <- 1
tau <- 1

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

Figure 2: Multivariate rank histogram for seven pre-rank functions in the multivariate normal distribution simulation study, when the forecast distributions: (a) under-estimate the mean, (b) over-estimate the mean, (c) under-estimate the variance, (d) over-estimate the variance, (e) under-estimate the correlation, (f) over-estimate the correlation.

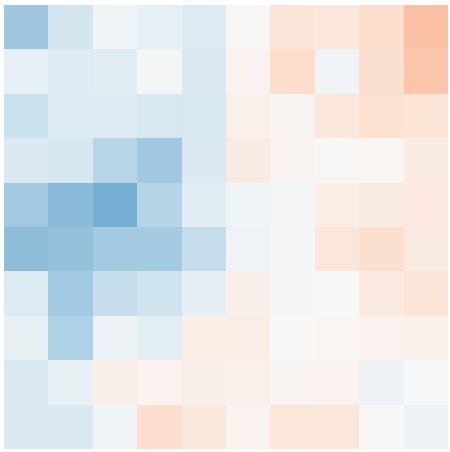

Figure 3: Example realisation of a Gaussian random field.

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(p, q, M, n)
```

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 and fraction of threshold exceedances pre-rank functions require additional arguments corresponding to the spatial lag(s) or threshold.

```
t <- 1 # threshold to be used in FTE pre-rank function

# create array with weights that decrease exponentially with distance
w_mat <- array(NA, c(p, q, p, q))
for (i in 1:p) {
  for (j in 1:q) {
    w_mat[i, j, , ] <-
      outer(1:p, 1:q, FUN = function(k, l) exp(-sqrt((i - k)^2 + (j - l)^2)))
  }
}

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], prerank = "variogram", w = w_mat)
  fte <- get_prerank_gr(y[, , i], x[, , , i], prerank = "FTE", t = t)
  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,

```
tau_x <- 0.5
x <- grf(d, grid = "reg", cov.pars = c(sig2, tau_x), nsim = n*M, messages = F)
```

and the isotropy.

```
# rescale the fields vertically by a factor of 10%
x <- grf(d, grid = "reg", cov.pars = c(sig2, tau), aniso.pars = c(0, 1.1),
         nsim = n*M, messages = F)
```

The resulting multivariate rank histograms are displayed in Figure 4.

5. Case study

5.1. Data

Consider now 10m wind speed forecasts and observations from the European Meteorological Network's (EUMETNET) post-processing benchmark dataset (EUPPBench; Demaeyer *et al.* 2023). For every Monday and Thursday in 2017 and 2018, five years of reforecasts have been generated using the European Center for Medium-range Weather Forecasts' (ECMWF) Integrated Forecasting System (IFS). The ensemble forecasts are available at a lead time of five days and are comprised of $M = 11$ ensemble members.

The forecasts are compared to ERA5 reanalyses (Hersbach *et al.* 2020), which provide a best guess for the observed wind speed fields. The forecasts and observations are on a regular longitude-latitude grid that covers a small domain in central Europe (2.5-10.5E, 45.75-53.5N). The grid has a horizontal resolution of 0.25° and is comprised of 33 distinct longitudes and 32 latitudes.

The IFS ensemble forecasts are compared to forecasts obtained from two distinct statistical post-processing methods. The goal of post-processing is to remove systematic errors that manifest in the output of numerical weather models. Both post-processing methods assume that the future wind speed at each grid point follows a logistic distribution that is truncated below at zero, with location and scale parameters that depend linearly on the IFS ensemble

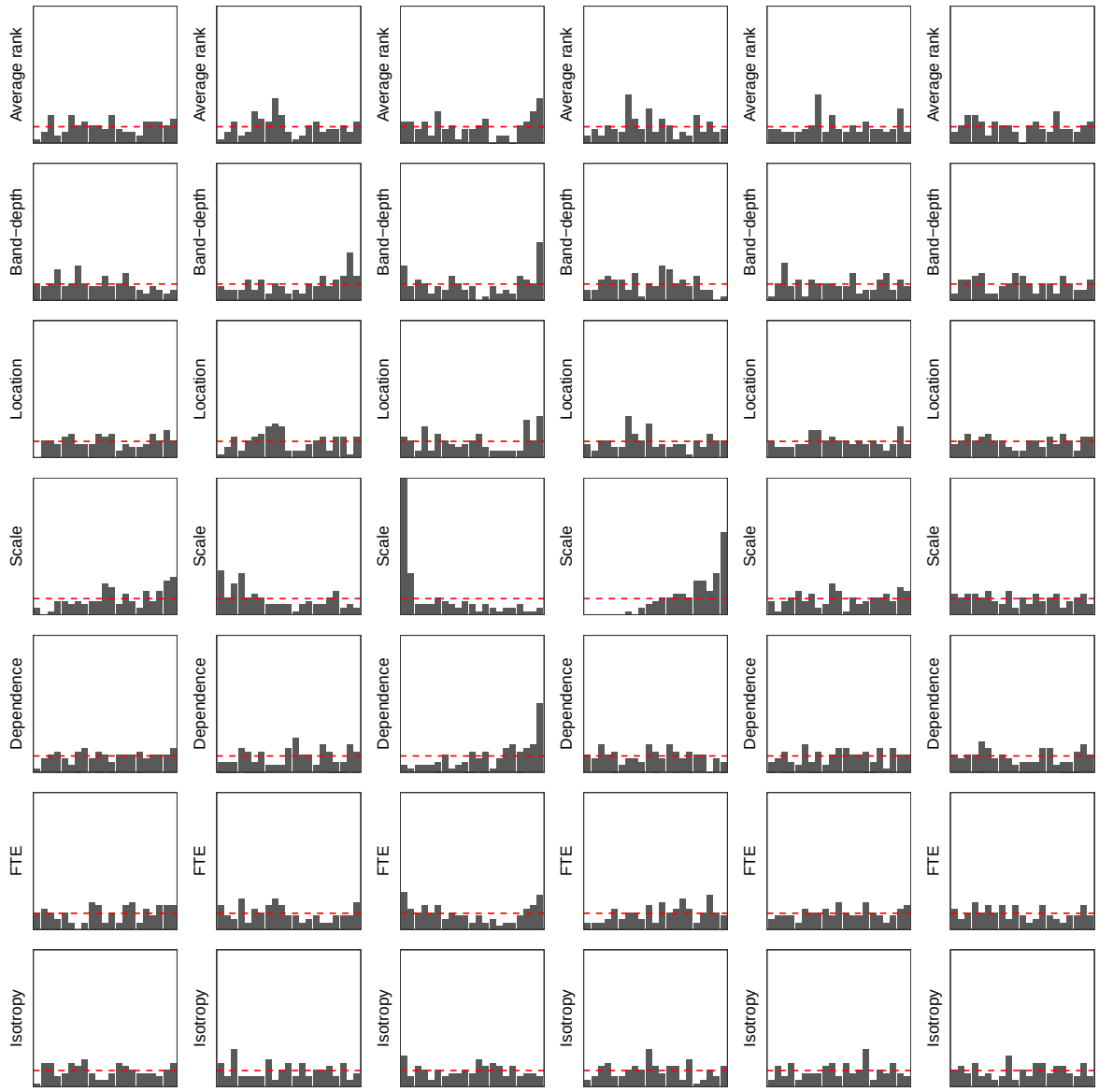

Figure 4: Multivariate rank histogram for seven pre-rank functions in the Gaussian random field simulation study, when the forecast distributions: (a) under-estimate the variance, (b) over-estimate the variance, (c) under-estimate the correlation, (d) over-estimate the correlation, (e) under-estimate the isotropy, (f) over-estimate the isotropy.

mean and standard deviation, respectively. These parameters are estimated from fifteen years of reforecasts prior to the time period under consideration.

To obtain forecast distributions that have a realistic spatial dependence structure, evenly-spaced quantiles are extracted from the univariate post-processed distributions, and then reordered according to a relevant dependence template. Two approaches are considered here: ensemble copula coupling (ECC), which uses the raw IFS ensemble forecasts as a template to reorder the post-processed forecast distributions (Schefzik *et al.* 2013); and the Schaake Shuffle, which instead uses a random selection of past multivariate observations to construct the dependence template (Clark *et al.* 2004). The calibration of the forecast fields obtained from these two post-processing methods is compared to the calibration of the raw numerical model output.

The forecast and observation fields can be obtained from the **MultivCalibration** package using

```
data("wind_dat")
list2env(wind_dat, globalenv()) # convert list elements to global environment

## <environment: R_GlobalEnv>

rm(wind_dat)
```

To save time when building the vignettes, attention here is restricted to a subset of the domain containing 10 longitudes and 10 latitudes.

```
n <- 1045
p <- 10          # 33 is used in Allen et al. (2023)
q <- 10          # 32 is used in Allen et al.
d <- p*q
M <- 11

# consider first 100 forecast cases to save time
obs <- obs[1:n, 1:p, 1:q]
fc_ifs <- fc_ifs[1:n, 1:p, 1:q, 1:M]
fc_ecc <- fc_ecc[1:n, 1:p, 1:q, 1:M]
fc_ss <- fc_ss[1:n, 1:p, 1:q, 1:M]
```

5.2. Results

Firstly, consider the calibration of the forecasts at each grid point. Rank histograms for the IFS and post-processed forecasts are displayed in Figure 5, with the ranks aggregated over all times and grid points; the ECC and Schaake shuffle differ only in how they reorder the post-processed distributions, so result in the same rank histograms. The IFS forecasts are slightly under-dispersed, resulting in a $\cup$-shaped histogram, whereas the post-processed forecasts appear better calibrated.

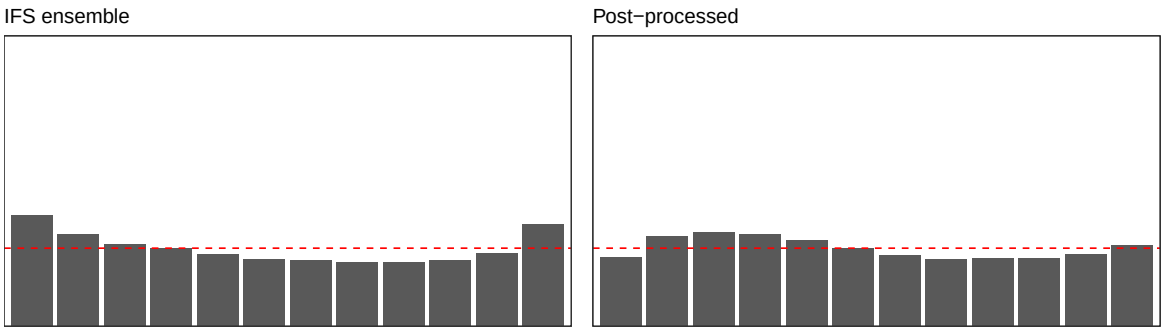

Figure 5: Univariate rank histograms for the IFS and post-processed ensemble forecasts. A dotted red line indicates a flat histogram. Ranks have been aggregated across all grid points.

```
# function to get univariate ranks
get_univ_ranks <- function(y, x) {
  ranks <- sapply(1:nrow(y), function(i) sapply(1:ncol(x), function(j)
    get_prerank_gr(y, x, prerank = function(z) z[i, j])))
  return(ranks)
}

univ_ranks_ifs <- sapply(1:n, function(i)
  get_univ_ranks(obs[i, , ], fc_ifs[i, , , ]))

univ_ranks_ecc <- sapply(1:n, function(i)
  get_univ_ranks(obs[i, , ], fc_ecc[i, , , ]))


plot_ifs <- pit_hist(as.vector(univ_ranks_ifs),
                     ymax = 0.31, title = "IFS ensemble",
                     xticks = F, yticks = F, xlab = NULL, ylab = NULL)

plot_ecc <- pit_hist(as.vector(univ_ranks_ecc),
                     ymax = 0.31, title = "Post-processed",
                     xticks = F, yticks = F, xlab = NULL, ylab = NULL)
```

Multivariate rank histograms can be obtained using the **MultivCalibration** functionality.

```
t <- 6 # threshold to be used in the FTE pre-rank function

# create array with weights that decrease exponentially with distance
w_mat <- array(NA, c(p, q, p, q))
for (i in 1:p) {
  for (j in 1:q) {
    w_mat[i, j, , ] <-
      outer(1:p, 1:q, FUN = function(k, l) exp(-sqrt((i - k)^2 + (j - l)^2)))
  }
```

```
}

# wrapper to get data frame of ranks corresponding to each pre-rank function
get_ranks <- function(y, x, w_mat, t = 1, h = 1) {
  n <- nrow(y)

  get_ranks_i <- function(y, x, w_mat, t, h) {
    ranks_i <- c(get_prerank_gr(y, x, prerank = "average_rank"),
                 get_prerank_gr(y, x, prerank = "band_depth"),
                 get_prerank_gr(y, x, prerank = "mean"),
                 get_prerank_gr(y, x, prerank = "variance"),
                 get_prerank_gr(y, x, prerank = "variogram", w = w_mat),
                 get_prerank_gr(y, x, prerank = "FTE", t = t),
                 get_prerank_gr(y, x, prerank = "isotropy", h = h))
    names(ranks_i) <- c("avr", "bdr", "mea", "var", "dep", "fte", "iso")
    return(ranks_i)
  }

  output <- sapply(1:n, function(i) {
    get_ranks_i(y[i, , ], x[i, , , ], w_mat, t, h)})
  rank_df <- data.frame(t(output))


  return(rank_df)
}


rank_df_ifs <- get_ranks(obs, fc_ifs, w_mat, t = 6)
rank_df_ecc <- get_ranks(obs, fc_ecc, w_mat, t = 6)
rank_df_ss <- get_ranks(obs, fc_ss, w_mat, t = 6)
```

While only in-built pre-rank functions are considered here, alternative custom functions could also have been studied. Multivariate rank histograms for all three methods are displayed in Figure 6. The IFS forecasts appear calibrated with respect to the average rank, variance, and fraction of threshold exceedance pre-rank functions, but miscalibrated when assessed using the band-depth, dependence, and isotropy pre-rank functions. The two post-processing methods perform similarly: post-processing removes the errors in the band-depth pre-rank function, resulting in a more uniform histogram; however, the forecasts still exhibit large errors in the isotropy and the dependence between the wind speed at neighbouring grid points. This is slightly weaker for the Schaake shuffle forecasts.

While rank histograms provide a graphical visualisation of multivariate forecast calibration, a formal test for calibration can be derived by checking whether the histogram is flat. This can be achieved sequentially over time using e-values. Technical details about e-values can be found in Arnold *et al.* (2021) and Allen *et al.* (2023), and they can be applied in practice using the `e_rank_histogram()` and `evalue_combine_h()` functions in the **epit** package. This package is not available on CRAN, but can be installed using **devtools**

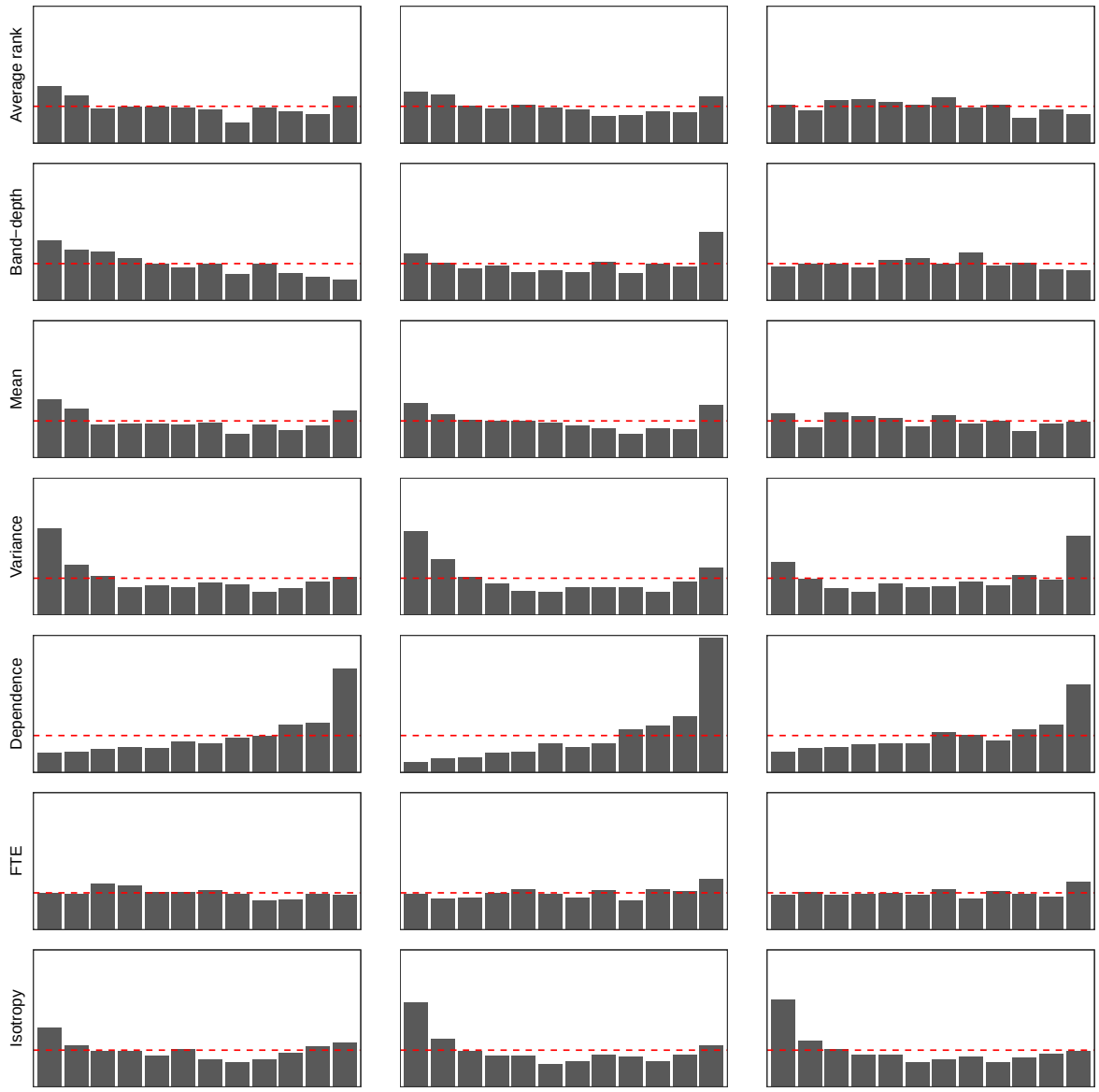

Figure 6: Multivariate rank histograms corresponding to several pre-rank functions for the IFS (left), ECC (centre), and Schaake shuffle (right) forecast fields.

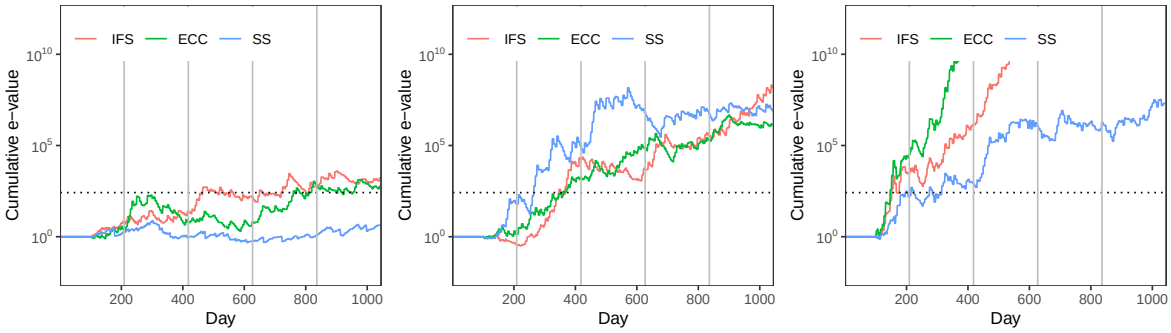

Figure 7: E-values for the three forecasting methods as a function of time, corresponding to band-depth (left), variance (middle), and dependence (right) pre-rank functions. When the e-value exceeds the horizontal dotted line, there is significant evidence to suggest that the forecast fields are miscalibrated with respect to the associated pre-rank function at the 5% level.

```
devtools::install_github("AlexanderHenzi/epit")
```

We can write a wrapper to apply these functions to extract a time series of e-values for each pre-rank function and forecasting method. For example, consider the band-depth rank

```
alpha <- 0.05

get_evals <- function(r, lag, m, n0 = 1, strategy = "betabinom") {
  evals <- epit::e_rank_histogram(r = r, h = lag, m = m,
                                  options = list(n0 = n0),
                                  strategy = strategy)$evalues_h
  evals <- epit:::evalue_combine_h(lapply(evals, function(x) x$e))
  return(evals)
}
```

This can be repeated for other pre-rank functions. Figure 7 displays the e-values corresponding to the band-depth, variance, and variogram pre-rank functions. An increasing e-value suggests the forecasts are becoming increasingly miscalibrated, and a horizontal line has been added to the plots that corresponds to the critical value of a hypothesis test for calibration at the 5% level.

The e-values reinforce the conclusions drawn from the multivariate rank histograms. The IFS and ECC forecasts are significantly miscalibrated with respect to the band-depth pre-rank function, while the Schaake shuffle forecasts appear calibrated. All three forecast methods are severely miscalibrated when interest is on the dependence between wind speeds at neighbouring locations.

Finally, since multivariate forecast calibration is assessed using multiple pre-rank functions, one might ask what constitutes a good set of pre-rank functions? A collection of pre-rank functions will be most useful when the individual pre-rank functions provide complementary information. To assess this for this application, we can calculate the correlation between the ranks obtained using different pre-rank functions.

```
get_srcc <- function(rank_df) {
  k <- ncol(rank_df)
  srcc <- sapply(1:k, function(i) sapply(1:k, function(j)
    cor(rank_df[[i]], rank_df[[j]])))
  rownames(srcc) <- colnames(srcc) <- colnames(rank_df)
  return(srcc)
}

round(get_srcc(rank_df_ifs), 2)

##       avr   bdr   mea   var   dep   fte   iso
## avr  1.00 -0.01  0.98  0.39  0.26  0.23 -0.08
## bdr -0.01  1.00 -0.01 -0.17 -0.02 -0.12  0.00
## mea  0.98 -0.01  1.00  0.41  0.28  0.25 -0.08
## var  0.39 -0.17  0.41  1.00  0.58  0.19 -0.14
## dep  0.26 -0.02  0.28  0.58  1.00  0.07 -0.03
## fte  0.23 -0.12  0.25  0.19  0.07  1.00  0.00
## iso -0.08  0.00 -0.08 -0.14 -0.03  0.00  1.00
```

Results are shown for the raw IFS forecasts, though this could easily be repeated for the two post-processing methods. There is a strong positive correlation between the average rank and location pre-rank functions, which both assess the mean behaviour of the spatial fields, and also the FTE pre-rank function. There is also strong positive correlation between the variance and dependence pre-rank functions. The band-depth and isotropy pre-rank functions, on the other hand, exhibit relatively low correlations with the other pre-rank functions, making them particularly useful in this application.

6. 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*.

Arnold S, Henzi A, Ziegel JF (2021). "Sequentially valid tests for forecast calibration." *arXiv preprint arXiv:2109.11761*.

Clark M, Gangopadhyay S, Hay L, Rajagopalan B, Wilby R (2004). "The Schaake shuffle: A method for reconstructing space–time variability in forecasted precipitation and temperature fields." *Journal of Hydrometeorology*, **5**, 243–262.

Demaeyer J, Lerch S, Primo C, Van Schaeybroeck B, Atencia A, Ben Bouallègue Z, Chen J, Dabernig M, Evans G, Faganeli Pucer J, *et al.* (2023). "The EUPPBench postprocessing benchmark dataset v1.0." *Earth System Science Data Discussions*, pp. 1–25.

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.

Hersbach H, Bell B, Berrisford P, Hirahara S, Horányi A, Muñoz-Sabater J, Nicolas J, Peubey C, Radu R, Schepers D, *et al.* (2020). "The ERA5 global reanalysis." *Quarterly Journal of the Royal Meteorological Society*, **146**, 1999–2049.

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

Schefzik R, Thorarinsdottir TL, Gneiting T (2013). "Uncertainty Quantification in Complex Simulation Models Using Ensemble Copula Coupling." *Statistical Science*, pp. 616–640.

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