



Verification of Ensemble Forecasts: Ensemble-adjusted Scores, Comparative Verification, and Uncertainty Quantification Implemented in the R Package SpecsVerification

Stefan Siegert
University of Exeter

Abstract

Forecast verification, the comparison of retrospective forecasts to observations, is a common task at institutions that develop and issue forecasts, such as climate centers. To assess forecast uncertainty, ensembles of forecasts initialised from perturbed initial conditions are routinely issued. In recent years, advances have been made in statistical methodology for ensemble forecast verification, in particular to estimate the finite-ensemble effect on verification scores. This paper summarises statistical methodology to account for finite-ensemble effects in ensemble verification, to compare the quality of ensemble forecasts with different numbers of ensemble members, and to quantify uncertainty in forecast verification results. Implementations of the methods are freely available in the R package **SpecsVerification**.

Keywords: ensemble forecasting, forecast verification, finite-ensemble effect, comparative verification, uncertainty quantification, R.

Note to editor and reviewers

A package called **SpecsVerification** is currently available on CRAN. The package documented in this paper is an updated, yet unpublished version of **SpecsVerification**, documenting new and rewritten functions that are not currently available.

1. Introduction

An ensemble forecast is a collection of forecasts for the same target. The ensemble members

usually differ due to differences in initial conditions, boundary conditions, model physics and background information (Gneiting 2005; Leutbecher and Palmer 2008). Ensemble forecasting is today operationally used in weather and climate forecasting to explore the chaotic divergence of nonlinear systems, and to estimate forecast uncertainty. To evaluate the reliability and predictive skill of ensemble forecasts, the forecasts have to be compared to their verifying observations of the real world. The comparison of forecasts with their verifying observations is commonly referred to as forecast verification (Jolliffe and Stephenson 2012). A verification measure is thus a function that depends on an archive $D = \{x_t, y_t\}_{t=1}^N$ of past forecasts x_t and verifying observations y_t . The archive D is also called a hindcast data set. A variety of verification measures exists to assess the quality of different forecast products, such as deterministic forecasts, probabilistic forecasts, ensemble forecasts, univariate or multivariate forecasts, and gridded or unevenly spaced spatial forecasts.

The R package **SpecsVerification** documented in this paper focuses primarily on verification scores for ensemble forecasts. It is known that verification measures can depend systematically on the ensemble size; everything else being equal, large ensembles achieve better average scores than small ensembles (Buizza and Palmer 1998). Since hindcast experiments with state-of-the-art climate models are computationally expensive, new model configurations are often verified on small ensemble sizes. Large ensembles are only generated in forecast mode. But if verification measures depend on the ensemble size, the skill calculated for the hindcast experiment is not representative of the skill that an operational forecast ensemble with more members would achieve. A number of verification scores have been proposed in the past to estimate the finite ensemble effect. The main contribution of this paper is to summarise ensemble verification scores, and document their implementation in the R statistical programming environment. The package **SpecsVerification** contains functions to calculate verification scores, compare verification scores, and quantify uncertainty.

2. Forecast and observation data

SpecsVerification includes the data set `eurotempforecast` of seasonal temperature ensemble forecasts and verifying observations. The forecasts 24-member ensembles of near-surface air temperatures produced by the NCEP climate forecast system version 2 (Saha, Moorthi, Wu, Wang, Nadiga, Tripp, Behringer, Hou, Chuang, Iredell, and Coauthors 2014), initialised between 11 April and 6 May each year from 1983–2009 ($N = 27$), and averaged over the region limited by latitudes $30^\circ\text{N} - 75^\circ\text{N}$ and longitudes $12.5^\circ\text{W} - 42.5^\circ\text{E}$, and over the months June-July-August, i.e., the lead time is about 1–3 months. Data from the NCEP climate forecast system reanalysis (Saha, Moorthi, Pan, Wu, Wang, Nadiga, Tripp, Kistler, Woollen, Behringer, and Coauthors 2010) was taken as verifying observations. All data was downloaded through the ECOMS user data gateway R-interface (Santander Meteorology Group 2015). Ensemble members and observation data are plotted as time series in Figure 1.

```
data(eurotempforecast)
R    <- ncol(ens)
yrs  <- as.numeric(names(obs))
N    <- length(obs)
```

All functions in **SpecsVerification** that analyse ensemble forecast data, assume that archives

Table 1: Overview of verification functions implemented in **SpecsVerification**, and their application.

FUNCTION	LONG NAME	APPLIES TO
AbsErr	Absolute error Score*	Deterministic forecasts of continuous observations
Auc	Area under the ROC curve	Probability forecasts of binary observations
AucDiff	Difference between two areas under the ROC curve	Two competing probability forecasts for the same binary observations
Corr	Correlation coefficient	Deterministic forecasts of continuous observations
CorrDiff	Difference between two correlation coefficients	Two competing deterministic forecasts for the same continuous observations
DressCrps	Continuous ranked probability score for dressed ensembles*	Ensemble forecasts of continuous observations
DressIgn	Ignorance score for dressed ensembles*	Ensemble forecasts of continuous observations
EnsBrier	Ensemble-adjusted Brier score*	Ensemble forecast of binary observations
EnsCrps	Ensemble-adjusted continuous ranked probability score*	Ensemble forecasts of continuous observations
EnsRps	Ensemble-adjusted ranked probability score*	Ensemble forecasts of categorical observations
EnsQs	Ensemble-adjusted quadratic score*	Ensemble forecasts of categorical observations
GaussCrps	Continuous ranked probability score for Normal distributions*	Probability forecasts of continuous observations
PlotRankhist	Plot a rank histogram	see Rankhist
Rankhist	Calculate a rank histogram	Ensemble forecasts of continuous observations
ReliabilityDiagram	Calculate and plot a reliability diagram	Probability forecasts of binary observations
ScoreDiff	Calculate a score difference and assess uncertainty	All scores marked with a *
SkillScore	Calculate a skill score and assess uncertainty	All scores marked with a *
TestRankhist	Statistical tests of a rank histogram	see Rankhist
SqErr	Squared error score*	Deterministic forecasts of continuous observations

```
par(las=1, cex=0.7, mgp=c(3, 1, 0), mar=c(2,4,1,1))
matplot(yrs, ens, ylab="temp [C]", pch=1, col=gray(.5))
points(yrs, obs, pch=15, cex=1.5)
```

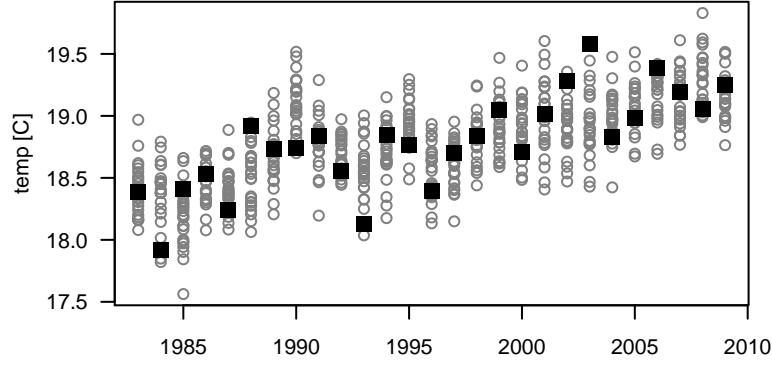


Figure 1: Seasonal European temperature ensemble forecasts (circles) and verifying observations (squares).

of N instances of R -member ensemble forecasts, are represented by $N \times R$ matrices. The data shown in Figure 1 depicts 27 years or 24-member ensemble forecasts, and is thus a R matrix with 27 rows and 24 columns.

Real-valued forecasts and real-valued observations are saved as R matrix `ens` and R vector `obs`. The continuous data in `ens` and `obs` was used to derive binary and categorical forecasts and observations (matrices `ens.bin` and `ens.cat`, and vectors `obs.bin` and `obs.cat`). The binary forecasts, were generated by asking "Will this year's summer be warmer than last year's"? The binary observation at time index t is equal to one if the temperature in the corresponding year exceeds the temperature of the previous year, and zero otherwise. The individual ensemble members were transformed equivalently, by comparing the r th ensemble member in year t to the observation in year $t - 1$. Categorical forecasts and observations were generated by asking "Will this year's temperature be similar to last year's temperature (within a $0.5K$ range), colder, or warmer?" Ensemble members and the observation thus fall into one of 3 categories: Category 1 if the temperature that is less than $0.25K$ colder than the previous year, category 2 if the temperature is within $\pm 0.25K$ of the previous year, and category 3 if it is more than $0.25K$ warmer. In addition, the vector `obs.lag` is provided, containing 1-year lagged observed temperatures. For example, continuous, binary and categorical observations and forecasts for the year 2001 are

```
rbind(
  continuous = c(obs=obs["2001"], ens["2001", 1:5]),
  binary      = c(obs=obs.bin["2001"], ens.bin["2001", 1:5]),
  categorical = c(obs=obs.cat["2001"], ens.cat["2001", 1:5]))

##          obs.2001 Member_1 Member_2 Member_3 Member_4 Member_5
## continuous    19.01    18.65    18.96    19.23    18.41    18.81
## binary         1.00     0.00     1.00     1.00     0.00     1.00
```

##	categorical	3.00	2.00	3.00	3.00	1.00	2.00
----	-------------	------	------	------	------	------	------

3. Ensemble-adjusted verification scores

3.1. Binary ensemble forecasts

This subsection outlines the theory behind ensemble-adjusted verification scores, using probabilistic forecasts of binary events. One of the most common verification measures for probabilistic forecasts of binary events is the Brier score ([Brier 1950](#)). Suppose a probability forecast $p_t \in [0, 1]$ is issued at time t for a binary (yes/no) event. The occurrence (non-occurrence) of the event is coded as $y_t = 1$ ($y_t = 0$). The Brier score is given by the squared difference between forecast and observation:

$$s_B(p_t, y_t) = (p_t - y_t)^2 \quad (1)$$

The Brier score is negatively oriented: Lower scores indicate better forecasts. The Brier score is a strictly proper verification score, meaning that the expected score obtains its minimum value if and only if the observation y_t is a random draw from p_t ([Gneiting and Raftery 2007](#)).

Suppose that, instead of predicting the probability p_t , we make a prediction based on an ensemble forecast of size R . Each of the R ensemble members is an independent Bernoulli trial with success probability p_t . An unbiased estimator of the success probability p_t is the fraction i_t/R , where i_t is the number of ensemble members that predict the event $y_t = 1$. The Brier score of the probability forecast i_t/R is equal to

$$s_B\left(\frac{i_t}{R}, y_t\right) = \left(\frac{i_t}{R} - y_t\right)^2 \quad (2)$$

Taking expectation over the random variable $i_t \sim \text{Binomial}(p_t, R)$, it is shown that ([Ferro, Richardson, and Weigel 2008](#))

$$E\left[s_B\left(\frac{i_t}{R}, y_t\right)\right] = s_B(p_t, y_t) + \frac{p_t(1 - p_t)}{R} \quad (3)$$

That is, the Brier score of i_t/R is not an unbiased estimator of the Brier score of p_t . The expected Brier score of i_t/R is larger (i.e. worse) than the Brier score of p_t . The bias, given by the additional positive term on the rhs of Equation 3, depends on the ensemble size and vanishes for $R \rightarrow \infty$. The bias can be interpreted as a finite-ensemble penalty: If two ensembles sample their members from the same probability p_t , the one with the larger ensemble size obtains the lower (i.e. better) expected Brier score. This is reasonable since more ensemble members allow for more robust estimation of the true probability p_t . But in the analysis of ensemble hindcasts it is sometimes desirable to correct the finite-ensemble bias. Suppose, for example, a hindcast ensemble has R members, but future operational forecasts will be made with $R^* > R$ ensemble members, using the same ensemble system. Due to the finite-ensemble bias, the score calculated for the R -member hindcast ensemble will be a too pessimistic estimate of the expected score of the R^* -member forecast ensemble. An adjustment is therefore desirable.

The ensemble-adjusted Brier score, given by (Ferro *et al.* 2008)

$$s_B^*(i_t, R, R^*, y_t) = \left(\frac{i_t}{R} - y_t \right)^2 - \frac{i_t(R - i_t)}{R(R - 1)} \left(\frac{1}{R} - \frac{1}{R^*} \right) \quad (4)$$

includes a correction of the finite-ensemble bias. The ensemble-adjusted Brier score is, in expectation, equal to the Brier score that would be achieved by an ensemble with R^* members, whose members are sampled from the same probability p_t , i.e.,

$$E[s_B^*(i_t, R, R^*, y_t)] = (p_t - y_t)^2 + \frac{p_t(1 - p_t)}{R^*}. \quad (5)$$

Note that, trivially, $s_B^*(i_t, R, R, y_t) = s_B(i_t/R, y_t)$. Note further that setting $R^* = \infty$ yields the fair Brier score (Ferro 2013) which is an unbiased estimator of the score of the underlying (unknown) probability p_t . The ensemble-adjusted Brier score can be used to compare ensemble forecasting systems (e.g. from different climate centers) that use different ensemble sizes. The score further allows for the extrapolation of the average score of an ensemble forecast system to larger ensemble sizes, e.g., to estimate forecast skill (with R^* members) from hindcast skill (with $R < R^*$ members).

The **SpecsVerification** function **EnsBrier** calculates the ensemble-adjusted Brier scores of a collection of N ensemble forecasts and their corresponding binary observations. The argument **R.new** allows for estimation of the score of an arbitrary ensemble size, including **R.new=Inf**. To illustrate the finite ensemble effect we randomly split the 24-member forecast ensemble **ens** into a small 5-member ensemble and a larger 19-member ensemble, and calculate their unadjusted Brier scores (Equation 2):

```
i.small <- sample(1:R, 5)
i.large <- setdiff(1:R, i.small)
c(small.ens=mean(EnsBrier(ens.bin[, i.small], obs.bin)),
  large.ens=mean(EnsBrier(ens.bin[, i.large], obs.bin)))

## small.ens large.ens
##      0.1689      0.1465
```

The large ensemble obtains a better score than the small ensemble. We next adjust the Brier score for the finite ensemble size, by calculating s_B^* using $R^* = 20$ for both ensembles:

```
c(small.ens=mean(EnsBrier(ens.bin[, i.small], obs.bin, R.new=19)),
  large.ens=mean(EnsBrier(ens.bin[, i.large], obs.bin, R.new=19)))

## small.ens large.ens
##      0.1454      0.1465
```

The scores are very similar. We have correctly estimated the score of the 19-member ensemble, using only a 5-member ensemble. How big is the estimated improvement if we could further increase the ensemble size? We show this in Figure 2, where we plot the ensemble adjusted Brier score over the adjusted ensemble size R^* . The fair Brier score, i.e. the ensemble-adjusted Brier score using $R^* = \infty$, is

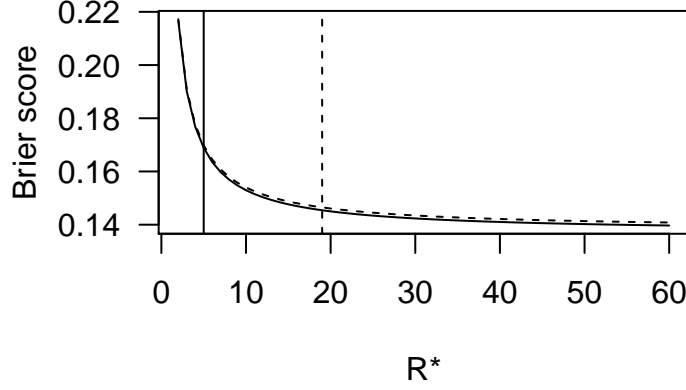


Figure 2: Ensemble-adjusted Brier score as a function of the adjusted ensemble size, calculated from a small 5-member ensemble (solid line) and a larger 19-member ensemble (dashed line), produced by the same forecasting system. Vertical lines indicate actual ensemble sizes.

```
mean(EnsBrier(ens.bin[, i.large], obs.bin, R.new=Inf))

## [1] 0.1382
```

3.2. Categorical ensemble forecasts

Suppose a probabilistic forecast $\mathbf{p}_t = (p_{t,1}, \dots, p_{t,K})$ is issued, whose k -th element equals the forecast probability that the observation will fall into class k . The verifying observation is vector-valued \mathbf{y}_t , where the k -th element of \mathbf{y}_t is $y_{t,k} = 1$ if the k -th class is observed, and $y_{t,j} = 0$ for all $j \neq k$. The quadratic score for such a probability forecast is given by

$$s_Q(\mathbf{p}_t, \mathbf{y}_t) = \sum_{k=1}^K (p_{t,k} - y_{t,k})^2 \quad (6)$$

Now assume an R -member categorical ensemble forecast \mathbf{i}_t is issued at time t , indicating that $i_{t,k}$ out of R ensemble members have predicted the k -th category, for $k = 1, \dots, K$. Using results obtained for the ensemble-adjusted Brier score, (see also [Ferro et al. 2008](#)), the ensemble-adjusted quadratic score is seen to be

$$s_Q^*(\mathbf{i}_t, R, R^*, \mathbf{y}_t) = \sum_{k=1}^K \left\{ \left(\frac{i_{t,k}}{R} - y_{t,k} \right)^2 - \left(\frac{1}{R} - \frac{1}{R^*} \right) \frac{i_{t,k}(R - i_{t,k})}{R(R-1)} \right\} \quad (7)$$

The ensemble adjusted quadratic score is implemented as the function `EnsQs` in **SpecsVerification**.

The quadratic score is insensitive to relabelling the K categories. An order-sensitive score can be constructed by nesting the forecast categories within each other. The forecast vector \mathbf{i}_t is transformed to the K -element cumulated forecast vector \mathbf{j}_t , with k -th element equal to $j_{t,k} = \sum_{l=1}^k i_{t,l}$. Likewise, the cumulated observation vector \mathbf{z}_t has its k -th element equal to $z_{t,k} = \sum_{l=1}^k y_{t,l}$. The quadratic score of the cumulated forecast \mathbf{j}_t evaluated on the cumulated observation \mathbf{z}_t , is called the ranked probability score (RPS). Accumulating the elements of \mathbf{i}_t and \mathbf{y}_t nests the K forecast categories within each other. The forecast is transformed from “ $i_{t,k}$ out of R ensemble members predict category k ” to the forecast “ $j_{t,k}$ out of R ensemble members forecast category k or less”. The nesting of forecast categories enables order-sensitivity of the score. Using results from the previous section, we get the ensemble-adjusted RPS

$$s_R^*(\mathbf{i}_t, R, R^*, \mathbf{y}_t) = \sum_{k=1}^K \left\{ \left(\frac{j_{t,k}}{R} - z_{t,k} \right)^2 - \left(\frac{1}{R} - \frac{1}{R^*} \right) \frac{j_{t,k}(R - j_{t,k})}{R(R - 1)} \right\} \quad (8)$$

The ensemble adjusted RPS is implemented as the function `EnsRps` in **SpecsVerification**. Note that the ensemble forecast vector \mathbf{i}_t in the above equations is assumed to be a histogram, indicating the number of ensemble members that forecast each class. The functions `EnsQs` and `EnsRps` assume as inputs true categorical ensemble forecasts, i.e. vectors of length R (ensemble size), with entries indicating which class label the ensemble predicts.

To illustrate the finite-ensemble effect and its adjustment, we once more split the ensemble randomly into a small 5-member ensemble and a larger 19-member ensemble. We then evaluate the unadjusted quadratic score of the full 24-member ensemble, the unadjusted scores of the small and large sub-ensembles, as well as the quadratic scores of the two sub-ensembles adjusted for the size $R^* = 24$ of the full ensemble:

```
i.small <- sample(1:R, 5)
i.large <- setdiff(1:R, i.small)
rbind(
  ens      = mean(EnsQs(ens.cat,      obs.cat)),
  small.ens = mean(EnsQs(ens.cat[, i.small], obs.cat)),
  large.ens = mean(EnsQs(ens.cat[, i.large], obs.cat)),
  small.ens.adj = mean(EnsQs(ens.cat[, i.small], obs.cat, R.new=24)),
  large.ens.adj = mean(EnsQs(ens.cat[, i.large], obs.cat, R.new=24)))

##           [,1]
## ens          0.5782
## small.ens     0.6519
## large.ens     0.5856
## small.ens.adj 0.5815
## large.ens.adj 0.5810
```

The full-ensemble obtains a better score than either of the sub-ensembles. The large sub-ensemble performs better than the small sub-ensemble. When adjusting scores of the sub-ensembles to $R^* = 24$ the scores of the sub-ensembles agree, and are close to the score of the full ensemble.

Repeating the evaluation with the ranked probability score, we obtain


```

rbind(
ens      = mean(EnsRps(ens.cat,          obs.cat)),
small.ens = mean(EnsRps(ens.cat[, i.small], obs.cat)),
large.ens = mean(EnsRps(ens.cat[, i.large], obs.cat)),
small.ens.adj = mean(EnsRps(ens.cat[, i.small], obs.cat, R.new=24)),
large.ens.adj = mean(EnsRps(ens.cat[, i.large], obs.cat, R.new=24)))

##           [,1]
## ens      0.3344
## small.ens 0.3719
## large.ens 0.3386
## small.ens.adj 0.3597
## large.ens.adj 0.3378

```

Again, the full ensemble outperforms both sub-ensembles, and the large sub-ensemble outperforms the small sub-ensemble. In this analysis, the scores do not agree that well after adjusting to $R^* = 24$. Recall that the equality of ensemble-adjusted scores is only in expectation, and empirical averages over finite hindcast data sets should be expected to differ.

3.3. Continuous ensemble forecasts

If the forecast target is a continuous variable, such as temperature or pressure, the continuous ranked probability score (Matheson and Winkler 1976) can be used for forecast verification. If the forecast for the continuous target y_t is given as a cumulative distribution function $F_t(x)$, the CRPS is given by

$$s_C(F_t, y_t) = \int_{-\infty}^{\infty} dz |F_t(z) - H(z - y_t)|^2 \quad (9)$$

where $H(x)$ is the Heaviside step-function, satisfying $H(x) = 1$ for all $x \geq 0$ and $H(x) = 0$ otherwise. Suppose an ensemble forecast x_t with R real-valued members $x_t = \{x_{t,1}, x_{t,2} \dots, x_{t,R}\}$ is issued for the real-valued verifying observation y_t . The ensemble can be transformed into a cdf by taking the empirical distribution function given by

$$\hat{F}_t(z) = \frac{1}{R} \sum_{r=1}^R H(z - x_{t,r}). \quad (10)$$

Using properties of the Heaviside function, it is possible to show that the CRPS of the empirical distribution \hat{F} is given by

$$s_C(\hat{F}_t, y_t) = \frac{1}{R} |x_{t,r} - y_t| - \frac{1}{2R^2} \sum_{r=1}^R \sum_{r'=1}^R |x_{t,r} - x_{t,r'}|. \quad (11)$$

Fricker, Ferro, and Stephenson (2013) show that the CRPS is sensitive to the ensemble size, and propose the ensemble-adjusted CRPS

$$s_C^*(x_t, R, R^*, y_t) = \frac{1}{R} \sum_{r=1}^R |x_{t,r} - y_t| - \frac{1}{2R(R-1)} \left(1 - \frac{1}{R^*}\right) \sum_{r=1}^R \sum_{r'=1}^R |x_{t,r} - x_{t,r'}|. \quad (12)$$

The ensemble-adjusted CRPS is, in expectation, equal to the CRPS that the empirical distribution function calculated from an ensemble of size R^* would achieve. This includes the case $R^* = \infty$, for which the fair CRPS is obtained (Fricker *et al.* 2013). The ensemble-adjusted CRPS is implemented in the **SpecsVerification** function `EnsCrps`.

```

rbind(
  unadjusted = mean(EnsCrps(ens, obs)),
  fair       = mean(EnsCrps(ens, obs, R.new=Inf))
)

##           [,1]
## unadjusted 0.1381
## fair       0.1329

```

4. Comparative verification and uncertainty quantification

4.1. Reference forecast

The value of a verification score by itself is often not easily interpretable. In order to evaluate the skill of a forecast, its verification score should be compared to the score achieved by a suitable reference forecast. For example, if a state-of-the-art high resolution climate model is evaluated, it is reasonable to compare its score to the score achieved by an older climate model version with lower resolution and less physical detail.

In the absence of a reference forecast generated by a dynamical climate model, a simple statistical benchmark prediction can be used. A popular statistical reference forecast is the climatological forecast, which is only based on the known record of observations. To benchmark ensemble forecasts, a climatological reference can be generated by sampling randomly from the record of known observations, or by treating all previously available observations as an ensemble forecast. **SpecsVerification** includes the function `ClimEns` which transforms a vector of observations into a matrix of climatological ensemble forecasts, including the possibility to leave out the t -th observation in the t -th climatological ensemble:

```

ens.ref      <- ClimEns(obs,      leave.one.out=TRUE)
ens.cat.ref  <- ClimEns(obs.cat, leave.one.out=TRUE)
ens.bin.ref  <- ClimEns(obs.bin, leave.one.out=TRUE)

```

In addition to the climatological forecast, it is advisable to also consider statistical reference forecasts such as a linear trend or an auto-regressive model, which are often more suitable than the climatological forecast.

4.2. Score differences

Suppose we have calculated two sets of N verification scores, $\{s_1^{(1)}, s_2^{(1)}, \dots, s_N^{(1)}\}$ for forecast 1, and $\{s_1^{(2)}, s_2^{(2)}, \dots, s_N^{(2)}\}$ for forecast 2, using the same set of observation. Diebold and Mariano (1995) suggest to test the null-hypothesis of equal forecast accuracy using the time

series d_1, \dots, d_N of loss differentials, $d_t = s_t^{(1)} - s_t^{(2)}$. Define \bar{d} to be the empirical average over d_1, d_2, \dots . Under the assumption of temporal independence of d_t , and zero mean of the loss-differential, the test statistic

$$T = \bar{d} \sqrt{\frac{N}{\text{var}(d_t)}} \quad (13)$$

is asymptotically Normally distributed with mean zero and variance one. This test is implemented in **SpecsVerification** in the function **ScoreDiff**. The function includes the option to account for autocorrelation of the loss-differential by specifying an effective sample size **N.eff**.

```

rbind(
  brier = ScoreDiff(EnsBrier(ens.bin,      obs.bin),
                    EnsBrier(ens.bin.ref, obs.bin)),
  qs    = ScoreDiff(EnsQs(  ens.cat,      obs.cat),
                    EnsQs(  ens.cat.ref, obs.cat)),
  rps   = ScoreDiff(EnsRps(  ens.cat,      obs.cat),
                    EnsRps(  ens.cat.ref, obs.cat)),
  crps  = ScoreDiff(EnsCrps( ens,          obs),
                    EnsCrps( ens.ref,      obs))
)

##      score.diff score.diff.sd   p.value      L      U
## brier    0.12185    0.04217 0.00192772  0.03921 0.2045
## qs       0.12004    0.08972 0.09046758 -0.05581 0.2959
## rps      0.09753    0.06438 0.06491317 -0.02866 0.2237
## crps     0.09391    0.02400 0.00004543  0.04688 0.1409

```

4.3. Skill scores

It is common practice to compare scores of competing forecasts by a so-called skill score, which is a normalised mean score difference (Wilks 2011). Denote by S the mean score of the forecast under evaluation, by S_{ref} the mean score of a reference forecast, and by S_{perf} the mean score that would be achieved by the perfect forecaster (often we have $S_{perf} = 0$). The skill score is then given by the average score difference between the reference forecast and the evaluated forecast, normalised by the average score difference between the reference forecast and the perfect forecast:

$$SS = \frac{S_{ref} - S}{S_{ref} - S_{perf}} \quad (14)$$

The variance of the skill score can be estimated by error propagation (also known as the delta-method) as follows:

$$\begin{aligned}
\text{var}(SS) \approx & \frac{1}{(S_{ref} - S_{perf})^2} \text{var}(S) + \frac{(S - S_{perf})^2}{(S_{ref} - S_{perf})^2} \text{var}(S_{ref}) \\
& - 2 \frac{S - S_{perf}}{(S_{ref} - S_{perf})^3} \text{cov}(S, S_{ref})
\end{aligned} \quad (15)$$

where the variances and covariances of the mean scores S and S_{ref} are approximated by the variances and covariances calculated for the individual scores, divided by the sample size. Calculation of skill scores is implemented in **SpecsVerification** in the function **SkillScore**, which takes as inputs two vectors of verification scores of the evaluated and the reference forecast, as well as a possibly user-defined effective sample size.

```

rbind(
  brier = SkillScore(EnsBrier(ens.bin, obs.bin),
                     EnsBrier(ens.bin.ref, obs.bin)),
  qs    = SkillScore(EnsQs(ens.cat, obs.cat),
                     EnsQs(ens.cat.ref, obs.cat)),
  rps   = SkillScore(EnsRps(ens.cat, obs.cat),
                     EnsRps(ens.cat.ref, obs.cat)),
  crps  = SkillScore(EnsCrps(ens, obs),
                     EnsCrps(ens.ref, obs))
)

##      skillscore skillscore.sd
## brier      0.4680      0.14973
## qs        0.1719      0.13042
## rps       0.2258      0.14624
## crps      0.4048      0.07343

```

4.4. Correlation and correlation difference

The Pearson (product-moment) correlation coefficient is one of the most popular verification criteria, and can be calculated with the built-in R function **cor**. Since uncertainty quantification is often of interest in forecast verification, **SpecsVerification** provides the function **Corr**, which returns the sample correlation coefficient r_{xy} , a one-sided p-value, and a confidence interval. The p-value is calculated based on the test statistic

$$T_{cor} = \sqrt{(N-2) \frac{r_{xy}^2}{1-r_{xy}^2}} \quad (16)$$

which has a Student's t-distribution with $N-2$ degrees of freedom under the null-hypothesis of zero correlation in the population. The confidence interval $[L, U]$ with a user-defined confidence coefficient $1-\alpha$, calculated by

$$[L, U] = \left[\tanh \left(z_{xy} + \frac{Z_{\alpha/2}}{\sqrt{N-3}} \right), \tanh \left(z_{xy} + \frac{Z_{1-\alpha/2}}{\sqrt{N-3}} \right) \right] \quad (17)$$

where $z_{xy} = \operatorname{atanh}(r_{xy})$ is the Fisher transformation of the correlation coefficient, and Z_p is the p -quantile of the standard Normal distribution. An effective sample size to account for possible auto-correlation can be specified.

```

ens.mean <- rowMeans(ens)
Corr(ens.mean, obs, conf.level=0.95)

##          corr          p.value          L          U
## 0.757095576 0.000002427 0.529391069 0.883049977

```

It is often of interest to compare the correlation coefficients between two forecasts for the same observation by calculating the correlation difference. **SpecsVerification** implements the function `CorrDiff` that returns the difference between the correlation r_{by} of the forecast B and the correlation r_{ay} of a reference forecast A, both of which were issued for the same observation Y. The function calculates a one-sided p-value, using the test for differences between overlapping correlation coefficients by [Steiger \(1980\)](#): Denote by r_{ab} the correlation between forecast A and forecast B. The determinant of the sample correlation matrix of the two forecasts and the observation is calculated:

$$R = (1 - r_{ay}^2 - r_{by}^2 - r_{ab}^2) + (2r_{ay}r_{by}r_{ab}) \quad (18)$$

The test statistic

$$T_{\text{cordiff}} = (r_{by} - r_{ay}) \sqrt{\frac{(N-1)(1+r_{ab})}{2 \left(\frac{N-1}{N-3}\right) R + \frac{1}{4}(r_{ay} + r_{by})^2(1-r_{ab})^3}} \quad (19)$$

has a Student's t -distribution with $N-3$ degrees of freedom under the null-hypothesis that A and B have equal correlations. Furthermore, a confidence interval for the correlation difference $r_{by} - r_{ay}$ is calculated, based on [Zou \(2007\)](#). First estimate the correlation between the correlation coefficients r_{ay} and r_{by} by

$$c_{ab} = \frac{(r_{ab} - \frac{1}{2}r_{ay}r_{by}) \left(1 - r_{ay}^2 - r_{by}^2 - r_{ab}^2\right) + r_{ab}^3}{(1 - r_{ay}^2)(1 - r_{by}^2)}, \quad (20)$$

Then calculate $(1-\alpha) \times 100\%$ confidence intervals (l_a, u_a) for r_{ay} and (l_b, u_b) for r_{by} , using the Fisher transformation as in Equation 17. An approximate $(1-\alpha) \times 100\%$ confidence interval (L, U) for the correlation difference $r_{by} - r_{ay}$ is then given by

$$\begin{aligned} L &= (r_{by} - r_{ay}) - \sqrt{(r_{by} - l_b)^2 + (u_a - r_{ay})^2 - 2c_{ab}(r_{by} - l_b)(u_a - r_{ay})}, \\ U &= (r_{by} - r_{ay}) + \sqrt{(u_b - r_{by})^2 + (r_{ay} - l_a)^2 - 2c_{ab}(u_b - r_{by})(r_{ay} - l_a)}. \end{aligned} \quad (21)$$

For illustration, we evaluate the difference in correlation between the ensemble mean forecast and the persistence forecast:

```

CorrDiff(fcst=ens.mean, fcst.ref=obs.lag, obs=obs, conf.level=0.95)

## corr.diff    p.value          L          U
## 0.179021    0.029082 -0.005417 0.440518

```

The one-sided p-value is small and the value of zero correlation is close to the boundary of the 95% confidence interval, which provides ample evidence that the seasonal forecast has higher correlation skill than the persistence forecast.

4.5. Area under the curve (AUC) and AUC differences

Relative operating characteristics (ROC, [Mason and Graham 2002](#), and references therein) analysis is a method from signal detection theory to evaluate the quality of forecasts for binary events. Consider the two competing forecasts $x_t^{(1)}$ and $x_t^{(2)}$ for the same binary observations $y_t \in \{0, 1\}$ for $t = 1, \dots, N$. (The forecasts are allowed to take values on the real line, and need not be probabilities.) For ROC analysis, the forecasts are grouped into two sets: $C_0^{(r)}$ contains all forecasts $x^{(r)}$ for which an event did not happen ($y_t = 0$), and $C_1^{(r)}$ contains all forecasts for which an event did happen ($y_t = 1$). The area under the ROC curve (AUC) for the r th forecast ($r = 1, 2$) is equal to the probability that a randomly drawn forecast from $C_1^{(r)}$ is larger than a randomly drawn forecast from $C_0^{(r)}$. The AUC is thus a measure of the ability of the forecast system to distinguish events from non-events.

[DeLong, DeLong, and Clarke-Pearson \(1988\)](#) suggest a nonparametric method to estimate the variance of AUC, and of differences in AUC. Denote by $X_i^{(r)}$, $i = 1, \dots, m$ the elements of $C_1^{(r)}$ and by $Y_i^{(r)}$, $i = 1, \dots, n$ the members of $C_0^{(r)}$. Define the function Ψ as

$$\Psi(x, y) = \mathbb{1}(x > y) + \frac{1}{2}\mathbb{1}(x = y) \quad (22)$$

where $\mathbb{1}(\cdot)$ is the indicator function which equals one if its argument is true, and zero otherwise. The AUC of the r th forecast is estimated by

$$\hat{\theta}^{(r)} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \Psi(X_i^{(r)}, Y_j^{(r)}) \quad (23)$$

For variance estimation, first define the quantities $V_i^{(r)}$ and $W_i^{(r)}$ by

$$V_i^{(r)} = \frac{1}{n} \sum_{j=1}^n \Psi(X_i^{(r)}, Y_j^{(r)}) \quad \text{and} \quad W_j^{(r)} = \frac{1}{m} \sum_{i=1}^m \Psi(X_i^{(r)}, Y_j^{(r)}) \quad (24)$$

and $v_{r,s}$ and $w_{r,s}$ by

$$v_{r,s} = \frac{1}{m-1} \sum_{i=1}^m [V_i^{(r)} - \hat{\theta}^{(r)}] [V_i^{(s)} - \hat{\theta}^{(s)}] \quad (25)$$

$$w_{r,s} = \frac{1}{n-1} \sum_{j=1}^n [W_j^{(r)} - \hat{\theta}^{(r)}] [W_j^{(s)} - \hat{\theta}^{(s)}] \quad (26)$$

where $r = 1, 2$ and $s = 1, 2$. Finally, the estimated variance of the r th AUC estimate $\hat{\theta}^{(r)}$ is given by

$$\text{var}(\hat{\theta}^{(r)}) = \frac{1}{m} v_{r,r} + \frac{1}{n} w_{r,r} \quad (27)$$

and the variance of the AUC difference $\theta^{(2)} - \theta^{(1)}$ is approximated by

$$\text{var}(\hat{\theta}^{(2)} - \hat{\theta}^{(1)}) = \frac{1}{m}(v_{1,1} + v_{2,2} - 2v_{1,2}) + \frac{1}{n}(w_{1,1} + w_{2,2} - 2w_{1,2}). \quad (28)$$

Note that the AUC is asymptotically Normally distributed. The estimated variance can therefore be used to construct a confidence interval, i.e., $\hat{\theta} \pm 1.96\sqrt{\text{var}(\hat{\theta})}$ is a central 95% confidence interval.

SpecsVerification provides the functions **Auc** and **AucDiff** that implement calculation of AUC and AUC differences and the corresponding variance estimates. The following calculates AUCs for the ensemble mean of the binary ensemble, using a large and a small subensemble:

```
rbind(
  large.ens = Auc(rowMeans(ens.bin[, i.large]), obs.bin),
  small.ens = Auc(rowMeans(ens.bin[, i.small]), obs.bin)
)

##           auc  auc.sd
## large.ens 0.8892 0.06944
## small.ens 0.8523 0.07179
```

The AUC of both ensembles is significantly larger than 0.5, which is a sign of forecast skill. The large subensemble has a slightly higher AUC than the small subensemble. The following evaluates the AUC difference between the large and a small ensemble:

```
AucDiff(rowMeans(ens.bin[, i.large]), rowMeans(ens.bin[, i.small]), obs.bin)

##      auc.diff auc.diff.sd
##      0.03693      0.06346
```

The AUC difference is within sampling variability, and we remain uncertain about whether the larger ensemble improves the AUC. In general, we expect ensemble size to effect AUC, but we are not aware of any published ensemble-adjustments for the AUC.

5. Rank histogram analysis for ensemble forecasts

The verification rank histogram (Hamill 2001) is a non-parametric graphical tool to assess the reliability of an ensemble forecasting system. For each pair of ensemble forecast and verifying observation, the rank of the observation among the ordered ensemble members is calculated. In a R -member ensemble, the rank is between 1 and $R + 1$. If the ensemble is a reliable representation of the uncertainty in the observation, the observation should statistically behave like “just another ensemble member”. Each verification rank should therefore be equally likely on average, and the histogram over verification ranks should be flat. **SpecsVerification** contains the function **Rankhist** to calculate the verification rank counts for an archive of ensembles and observations.

```
PlotRankhist(rh, mode="raw")
```

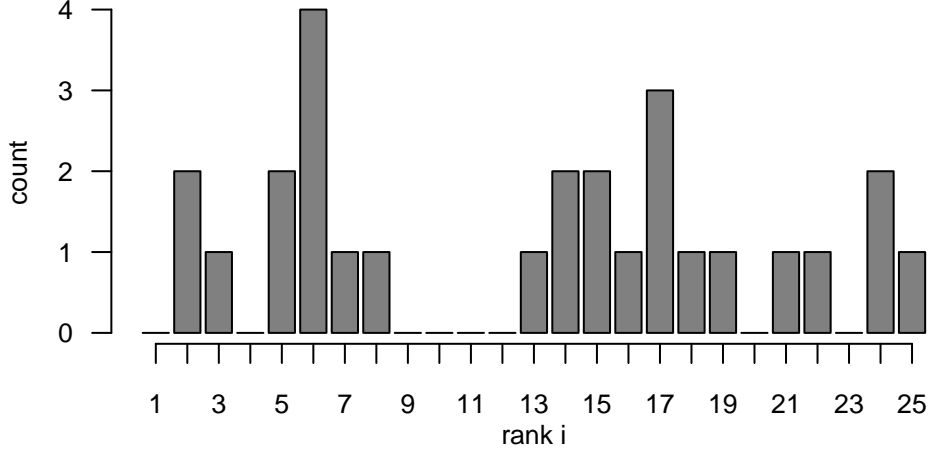


Figure 3: Rank histogram

```
rh <- Rankhist(ens, obs)
rh

## [1] 0 2 1 0 2 4 1 1 0 0 0 0 1 2 2 1 3 1 1 0 1 1 0 2 1
```

The function `PlotRankhist` plots the rank histogram. Two plotting modes are available: `mode="raw"` simply plots the rank counts as a bar plot histogram. `mode="prob.paper"` plots the rank counts on probability paper following Bröcker (2008). Assuming that each rank count has a binomial distribution with success probability $1/(R+1)$ and sample size N . The observed rank count c_i is transformed to the cumulative probability ν_i under the Binomial distribution. To test the null-hypothesis of a flat rank histogram, 90-, 95-, and 99-percent prediction intervals are included, corrected for multiple testing. The interpretation is, given that the null-hypothesis is true, on average 9 out of 10 rank histograms should lie completely inside the 90% prediction interval.

The function `TestRankhist` implements different statistical tests of the null-hypothesis of flat rank histogram. Flatness of the rank histogram can be assessed by a Pearson χ^2 -test (Pearson 1900). Suppose rank i was observed r_i times for $i = 1, \dots, R+1$, and define $e_i = N/(R+1) \forall i$ the expected number of counts if each verification rank were equally likely. Define further

$$x_i = \frac{r_i - e_i}{\sqrt{e_i}}. \quad (29)$$

Under the null-hypothesis of equally likely verification ranks, the test statistic

$$\chi^2 = \sum_{i=1}^{R+1} x_i^2 \quad (30)$$

has a χ^2 -distribution with R degrees of freedom.


```
PlotRankhist(rh, mode="prob.paper")
```

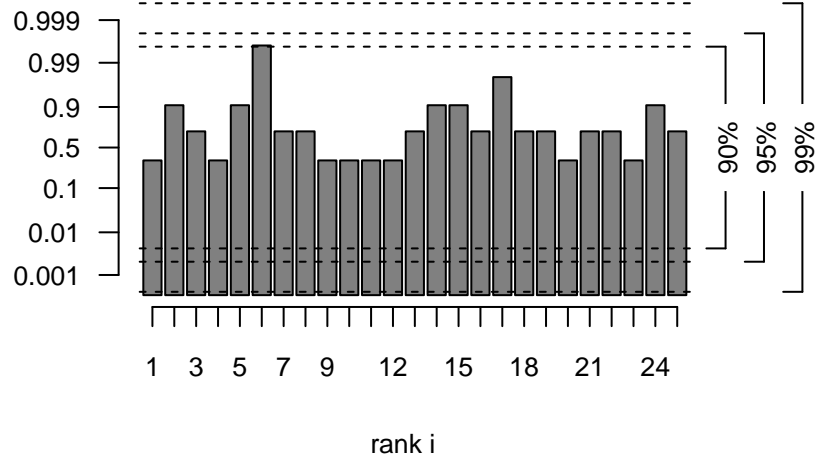


Figure 4: Rank histogram on probability paper

Hamill (2001) showed that certain types of violation of ensemble reliability are visible as different patterns in the rank histogram. In particular, a systematic bias of the ensemble mean produces sloped rank histograms, and ensembles with insufficient (excessive) ensemble spread produce U-shaped (\cap -shaped) rank histogram. Jolliffe and Primo (2008) showed that the χ^2 -test statistic can be decomposed to test for sloped and convex rank histograms specifically, thus increasing the power of the test. The test requires the definition of suitable contrast vectors \mathbf{c} of length $R + 1$, that satisfy $\sum_i c_i = 0$, $\sum_i c_i^2 = 1$, and $\sum_i c_i c'_i = 0$ for every pair of contrasts \mathbf{c} and \mathbf{c}' . Assuming a number of up to R contrast vectors $\mathbf{c}^{(1)}$, $\mathbf{c}^{(2)}$, \dots , the test statistics $(\sum_i c_i^{(k)} x_i)^2$ are independently χ^2 distributed with one d.o.f. The function **TestRankhist** applies this test, using a linear and a squared contrast. Defining $J = R + 1$, the i -th element of the contrast vectors $\mathbf{c}^{(lin)}$ and $\mathbf{c}^{(sq)}$, for $i = 1, \dots, J$ are given by

$$c_i^{(lin)} = -\sqrt{\frac{3(J+1)}{J(J-1)}} + i\sqrt{\frac{12}{J^3 - J}}, \text{ and} \quad (31)$$

$$c_i^{(sq)} = -\frac{\sqrt{5}J^2 - \sqrt{5}}{\sqrt{4(J-2)(J-1)J(J+1)(J+2)}} + \left(i - \frac{J+1}{2}\right)^2 \sqrt{\frac{180}{J^5 - 5J^3 + 4J}}. \quad (32)$$

The χ^2 test using the linear contrast is sensitive to sloped rank histograms, i.e. biased ensembles, while the χ^2 based on the squared contrast is sensitive to convex rank histograms, i.e. over- or under-dispersed ensembles. **TestRankhist** returns the test-statistics and one-sided p-values of the Pearson χ^2 test, and of the two tests based on the contrasts $\mathbf{c}^{(lin)}$ and $\mathbf{c}^{(sq)}$:

```
TestRankhist(rh)

##                pearson.chi2  jp.slope  jp.convex
## test.statistic      23.9259    0.0114   0.005574
## p.value             0.4658    0.9150   0.940485
```

The rank histogram of the temperature ensemble forecast provides no evidence against the null-hypothesis of a reliable ensemble.

6. Reliability diagrams for probability forecasts

The reliability diagram is a classical tool to compare probability forecasts of binary events to the verifying binary observations (Jolliffe and Stephenson 2012). The reliability diagram compares the forecast probability to the conditional frequency of the observation, given the forecast. A forecast is reliable if the forecast probability and conditional event frequency coincide. Forecast reliability is a reasonable criterion that probability forecasts should satisfy; over all instances that the forecast issued a probability p , the event should happen $p \times 100\%$ of the time.

If the forecast issues probabilities that take any value on the unit interval, most forecast probabilities will be issued only once. To estimate the conditional event frequency in this case the forecasts can be grouped into a finite number of non-overlapping bins. The average event frequency taken over all instances where the forecast is in a given bin is then taken as an average of the conditional event frequency. The reliability diagram is a plot of the conditional event frequency over the in-bin average of the forecast probabilities. **SpecsVerification** provides the function `ReliabilityDiagram` that takes as inputs a collection of probability forecasts and binary verifying observations, and calculates the reliability diagram for a specified number of equidistant bins, or a user-defined non-equidistant binning. The consistency resampling method proposed by Bröcker and Smith (2007) is used to estimate the likely spread of the reliability diagrams around the diagonal if the given forecast were, in fact, reliable.

If the `plot` argument is set to `FALSE`, the `ReliabilityDiagram` function returns the quantities necessary to plot the reliability diagram.

```
p.bin <- rowMeans(ens.bin)
ReliabilityDiagram(p.bin, obs.bin, plot=FALSE, bins=3)

##   p.avg  cond.probs  cbar.lo  cbar.hi  p.counts  bin.lower  bin.upper
## 1 0.1713    0.2222  0.0000  0.4784         9    0.0000    0.3333
## 2 0.5833    0.4286  0.1667  1.0000         7    0.3333    0.6667
## 3 0.8447    1.0000  0.6000  1.0000        11    0.6667    1.0000
```

If the argument `plot=TRUE` the reliability diagram is plotted, as shown in Figure 5. The logical argument `plot.refin` controls the refinement diagram, i.e. the histogram over the forecast probabilities. The logical argument `attributes` controls plotting of the polygon defined by the vertical no-resolution line at $\bar{y} = 1/n \sum_t y_t$, where y_t is the binary observation at time t , and the no-skill line defined by the linear equation $f(x) = (x + \bar{y})/2$, to produce

```
rd <- ReliabilityDiagram(p.bin, obs.bin, plot=TRUE, bins=3, attributes=TRUE)
```

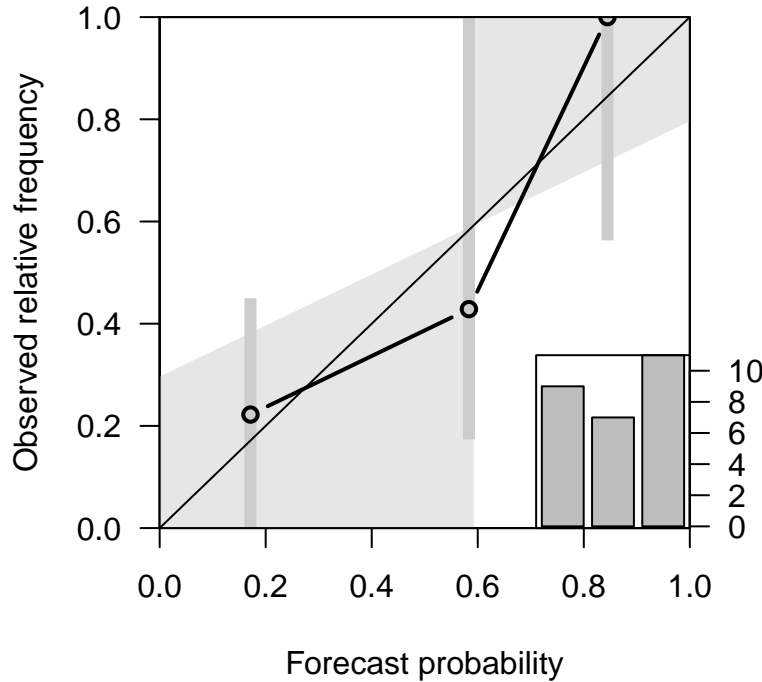


Figure 5: Reliability diagram

the attributes diagram (Hsu and Murphy 1986). Points that fall into the shaded area of the attributes diagram contribute positively to forecast skill (defined by the Brier skill score).

7. Additional functions

The focus of **SpecsVerification**, and of the present paper, is on verification functions for ensemble and probability forecasts, uncertainty quantification, and graphical display. **SpecsVerification** includes a number of functions for post-processing and verification of forecasts that were included in the package following user requests.

For verification of deterministic forecasts x_t , e.g. the ensemble mean forecast, the squared error score $(x_t - y_t)^2$ (function **SqErr**), and the Absolute Error score $|x_t - y_t|$ (function **AbsErr**) have been implemented. The function **GaussCrps** calculates the CRPS (eq. 9) where the forecast cdf $F_t(x)$ is a Normal distribution with mean μ_t and standard deviation σ_t^2 . The CRPS integral can then be solved analytically (Gneiting, Raftery, Westveld, and Goldman 2005) thus eliminating the need for expensive numerical integration. These score functions be analysed by the functions **ScoreDiff** and **SkillScore**.

Ensemble forecasts produced by numerical climate models often contain systematic errors due to numerical approximations, missing physical mechanisms in the model, or coding errors. These biases include a constant bias of the mean, or ensemble dispersion errors. Ensemble dressing is a statistical post-processing method based on kernel density estimation to transform an raw forecast ensemble into a smooth forecast probability function. In particular, affine kernel dressing (AKD; Bröcker and Smith 2008) is a method that corrects systematic model errors by an affine transformation of the ensemble, and produces a smooth forecast distribution by dressing the transformed ensemble members with Gaussian kernels. AKD is implemented in the function `DressEnsemble`, and the AKD parameters can be fitted with the function `FitAkdParameters`. The dressed ensemble can be evaluated by the CRPS using the function `DressCrps` which uses results by Grit, Gneiting, Berrocal, and Johnson (2006), and by the Ignorance Score (Roulston and Smith 2002), also known as the Logarithmic score, using the function `DressIgn`. The dressed ensemble can be further analysed by the functions `GetDensity` and `PlotDressedEns`.

Auxiliary functions were added to the package to, namely the `Detrend` to remove a common linear trend from an ensemble of forecasts, and the function `GenerateToyData` that simulates artificial ensemble and observation data using a statistical signal-plus-noise model (Siegert, Stephenson, Sansom, Scaife, Eade, and Arribas 2016).

8. Conclusion

The package **SpecsVerification** for the R statistical programming environment implements a variety of new forecast verification functions. The focus of **SpecsVerification** is on comparative verification of ensemble forecasts, and uncertainty quantification by statistical testing and confidence intervals. Continuous, categorical and binary ensemble forecasts for univariate quantities can be evaluated and compared by a number of verification scores. Additional functions for data transformation and statistical post-processing simplify a number of common verification tasks.

Acknowledgements

This paper benefitted from numerous discussions with the members of the statistical science group at the University of Exeter, in particular Christopher Ferro and David Stephenson. This work was supported by the European Union Programme FP7/2007-13 under grant agreement 3038378 (SPECS). The views expressed herein are those of the author and do not necessarily reflect the views of funding bodies or any of their subagencies.

References

- Brier GW (1950). “Verification of forecasts expressed in terms of probability.” *Monthly Weather Review*, **78**(1), 1–3. doi:10.1175/1520-0493(1950)078<0001:vofeit>2.0.co;2.
- Bröcker J (2008). “On reliability analysis of multi-categorical forecasts.” *Nonlinear Processes*

- in *Geophysics*, **15**(4), 661–673. ISSN 1607-7946. doi:[10.5194/npg-15-661-2008](https://doi.org/10.5194/npg-15-661-2008). URL <http://dx.doi.org/10.5194/npg-15-661-2008>.
- Bröcker J, Smith LA (2007). “Increasing the Reliability of Reliability Diagrams.” *Weather and Forecasting*, **22**(3), 651–661. ISSN 1520-0434. doi:[10.1175/waf993.1](https://doi.org/10.1175/waf993.1). URL <http://dx.doi.org/10.1175/WAF993.1>.
- Bröcker J, Smith LA (2008). “From ensemble forecasts to predictive distribution functions.” *Tellus A*, **60**(4), 663–678. ISSN 1600-0870. doi:[10.1111/j.1600-0870.2008.00333.x](https://doi.org/10.1111/j.1600-0870.2008.00333.x). URL <http://dx.doi.org/10.1111/j.1600-0870.2008.00333.x>.
- Buizza R, Palmer TN (1998). “Impact of Ensemble Size on Ensemble Prediction.” *Monthly Weather Review*, **126**(9), 2503–2518. ISSN 1520-0493. doi:[10.1175/1520-0493\(1998\)126<2503:ioesoe>2.0.co;2](https://doi.org/10.1175/1520-0493(1998)126<2503:ioesoe>2.0.co;2). URL [http://dx.doi.org/10.1175/1520-0493\(1998\)126<2503:IOESOE>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1998)126<2503:IOESOE>2.0.CO;2).
- DeLong ER, DeLong DM, Clarke-Pearson DL (1988). “Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach.” *Biometrics*, **44**(3), 837–845. ISSN 0006-341X. doi:[10.2307/2531595](https://doi.org/10.2307/2531595). URL <http://dx.doi.org/10.2307/2531595>.
- Diebold FX, Mariano RS (1995). “Comparing Predictive Accuracy.” *Journal of Business & Economic Statistics*, **13**(3), 253. doi:[10.2307/1392185](https://doi.org/10.2307/1392185).
- Ferro CAT (2013). “Fair scores for ensemble forecasts.” *Quarterly Journal of the Royal Meteorological Society*, **140**(683), 1917–1923. doi:[10.1002/qj.2270](https://doi.org/10.1002/qj.2270).
- Ferro CAT, Richardson DS, Weigel AP (2008). “On the effect of ensemble size on the discrete and continuous ranked probability scores.” *Meteorological Applications*, **15**(1), 19–24. doi:[10.1002/met.45](https://doi.org/10.1002/met.45).
- Fricker TE, Ferro CAT, Stephenson DB (2013). “Three recommendations for evaluating climate predictions.” *Meteorological Applications*, **20**(2), 246–255. doi:[10.1002/met.1409](https://doi.org/10.1002/met.1409).
- Gneiting T (2005). “ATMOSPHERIC SCIENCE: Weather Forecasting with Ensemble Methods.” *Science*, **310**(5746), 248–249. ISSN 1095-9203. doi:[10.1126/science.1115255](https://doi.org/10.1126/science.1115255). URL <http://dx.doi.org/10.1126/science.1115255>.
- Gneiting T, Raftery AE (2007). “Strictly Proper Scoring Rules, Prediction, and Estimation.” *Journal of the American Statistical Association*, **102**(477), 359–378. doi:[10.1198/016214506000001437](https://doi.org/10.1198/016214506000001437).
- Gneiting T, Raftery AE, Westveld AH, Goldman T (2005). “Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation.” *Monthly Weather Review*, **133**(5), 1098–1118. ISSN 1520-0493. doi:[10.1175/mwr2904.1](https://doi.org/10.1175/mwr2904.1). URL <http://dx.doi.org/10.1175/MWR2904.1>.
- Grimmett EP, Gneiting T, Berrocal VJ, Johnson NA (2006). “The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification.” *Q.J.R. Meteorol. Soc.*, **132**(621C), 2925–2942. ISSN 1477-870X. doi:[10.1256/qj.05.235](https://doi.org/10.1256/qj.05.235). URL <http://dx.doi.org/10.1256/qj.05.235>.

- Hamill TM (2001). “Interpretation of Rank Histograms for Verifying Ensemble Forecasts.” *Monthly Weather Review*, **129**(3), 550–560. ISSN 1520-0493. doi: [10.1175/1520-0493\(2001\)129<0550:iorhfv>2.0.co;2](https://doi.org/10.1175/1520-0493(2001)129<0550:iorhfv>2.0.co;2). URL [http://dx.doi.org/10.1175/1520-0493\(2001\)129<0550:IORHfV>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(2001)129<0550:IORHfV>2.0.CO;2).
- Hsu Wr, Murphy AH (1986). “The attributes diagram A geometrical framework for assessing the quality of probability forecasts.” *International Journal of Forecasting*, **2**(3), 285–293. ISSN 0169-2070. doi: [10.1016/0169-2070\(86\)90048-8](https://doi.org/10.1016/0169-2070(86)90048-8). URL [http://dx.doi.org/10.1016/0169-2070\(86\)90048-8](http://dx.doi.org/10.1016/0169-2070(86)90048-8).
- Jolliffe IT, Primo C (2008). “Evaluating Rank Histograms Using Decompositions of the Chi-Square Test Statistic.” *Monthly Weather Review*, **136**(6), 2133–2139. ISSN 1520-0493. doi: [10.1175/2007mwr2219.1](https://doi.org/10.1175/2007mwr2219.1). URL <http://dx.doi.org/10.1175/2007MWR2219.1>.
- Jolliffe IT, Stephenson DB (2012). *Forecast verification: a practitioner’s guide in atmospheric science*. John Wiley & Sons.
- Leutbecher M, Palmer T (2008). “Ensemble forecasting.” *Journal of Computational Physics*, **227**(7), 3515–3539. ISSN 0021-9991. doi: [10.1016/j.jcp.2007.02.014](https://doi.org/10.1016/j.jcp.2007.02.014). URL <http://dx.doi.org/10.1016/j.jcp.2007.02.014>.
- Mason SJ, Graham NE (2002). “Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation.” *Q. J. R. Meteorol. Soc.*, **128**(584), 2145–2166. ISSN 0035-9009. doi: [10.1256/003590002320603584](https://doi.org/10.1256/003590002320603584). URL <http://dx.doi.org/10.1256/003590002320603584>.
- Matheson JE, Winkler RL (1976). “Scoring Rules for Continuous Probability Distributions.” *Management Science*, **22**(10), 1087–1096. doi: [10.1287/mnsc.22.10.1087](https://doi.org/10.1287/mnsc.22.10.1087).
- Pearson K (1900). “X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.” *Philosophical Magazine Series 5*, **50**(302), 157–175. ISSN 1941-5990. doi: [10.1080/14786440009463897](https://doi.org/10.1080/14786440009463897). URL <http://dx.doi.org/10.1080/14786440009463897>.
- Roulston MS, Smith LA (2002). “Evaluating Probabilistic Forecasts Using Information Theory.” *Monthly Weather Review*, **130**(6), 1653–1660. ISSN 1520-0493. doi: [10.1175/1520-0493\(2002\)130<1653:epfuit>2.0.co;2](https://doi.org/10.1175/1520-0493(2002)130<1653:epfuit>2.0.co;2). URL [http://dx.doi.org/10.1175/1520-0493\(2002\)130<1653:EPFUIT>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(2002)130<1653:EPFUIT>2.0.CO;2).
- Saha S, Moorthi S, Pan HL, Wu X, Wang J, Nadiga S, Tripp P, Kistler R, Woollen J, Behringer D, Coauthors (2010). “The NCEP Climate Forecast System Reanalysis.” *Bulletin of the American Meteorological Society*, **91**(8), 1015–1057. ISSN 1520-0477. doi: [10.1175/2010bams3001.1](https://doi.org/10.1175/2010bams3001.1). URL <http://dx.doi.org/10.1175/2010BAMS3001.1>.
- Saha S, Moorthi S, Wu X, Wang J, Nadiga S, Tripp P, Behringer D, Hou YT, Chuang Hy, Iredell M, Coauthors (2014). “The NCEP Climate Forecast System Version 2.” *J. Climate*, **27**(6), 2185–2208. ISSN 1520-0442. doi: [10.1175/jcli-d-12-00823.1](https://doi.org/10.1175/jcli-d-12-00823.1). URL <http://dx.doi.org/10.1175/JCLI-D-12-00823.1>.

- Santander Meteorology Group (2015). *ecomUDG.Raccess: R interface to the ECOMS User Data Gateway*. R package version 4.2-0, URL <http://meteo.unican.es/trac/wiki/udg/ecomS>.
- Siegert S, Stephenson DB, Sansom PG, Scaife AA, Eade R, Arribas A (2016). “A Bayesian Framework for Verification and Recalibration of Ensemble Forecasts: How Uncertain is NAO Predictability?” *J. Climate*, **29**(3), 995–1012. ISSN 1520-0442. doi:10.1175/JCLI-D-15-0196.1. URL <http://dx.doi.org/10.1175/JCLI-D-15-0196.1>.
- Steiger JH (1980). “Tests for comparing elements of a correlation matrix.” *Psychological Bulletin*, **87**(2), 245–251. ISSN 0033-2909. doi:10.1037/0033-2909.87.2.245. URL <http://dx.doi.org/10.1037/0033-2909.87.2.245>.
- Wilks DS (2011). *Statistical methods in the atmospheric sciences*, volume 100. Academic press.
- Zou GY (2007). “Toward using confidence intervals to compare correlations.” *Psychological Methods*, **12**(4), 399–413. ISSN 1082-989X. doi:10.1037/1082-989X.12.4.399. URL <http://dx.doi.org/10.1037/1082-989X.12.4.399>.

Affiliation:

Stefan Siegert
Exeter Climate Systems
College for Engineering, Mathematics, and Physical Sciences
University of Exeter
Exeter, EX4 4QF, United Kingdom
E-mail: Stefan.Siegert@exeter.ac.uk
URL: <http://emps.exeter.ac.uk/mathematics/staff/ss610>