Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Forecast Verification Methods Implemented in the R package SpecsVerification: Ensemble Scores, Comparative Verification, and Uncertainty Quantification

Stefan Siegert
University of Exeter

Abstract

Keywords: keywords, comma-separated, not capitalized, Java.

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 The package documented in this paper is an updated, yet unpublished version of **SpecsVerification**, documenting new and rewritten functions that are not currently available.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

Ensemble forecasting general An ensemble forecast is a collection of forecasts for the same target. The ensemble members usually differ in the initial conditions, boundary conditions, model physics and background information. Ensemble forecasting is today operationally used to quantify forecast uncertainty. Since ensembles are composed of multiple runs of deterministic models, but are used to provide uncertainty information, they can be regarded as a hybrid of deterministic and probabilistic forecasts. To evaluate the quality of ensemble forecasts, the forecasts have to be compared to their verifying observations of the real world.

Forecast verification general 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 the collection $D = \{x_t, y_t\}_{t=1}^{N}$ of past forecasts x_t and verifying observations y_t. The collection D is also called a hindcast data set. A scoring rule is a function $s(x_t, y_t)$ that assigns a real number to a forecast-observation pair. A general convention that the best possible forecast achieves a score of zero, and lower scores indicate "better" forecasts. The empirical score $1/N \sum_t s(x_t, y_t)$ can be used as a verification measure. There verification measures that are not scoring rules; examples include the correlation coefficient, and the area under the ROC curve. There are different scoring rules and verification measures for deterministic forecasts and probabilistic forecasts. Since ensembles are interpreted as a hybrid, both types of verification measures can be applied to ensembles.

Finite size effect in ensemble verification It is known that verification measures can depend on the ensemble size. In general, large ensembles achieve better average verification scores than small ensembles, when both ensembles come from the same ensemble forecasting system (Buizza and Palmer 1998). In hindcast experiments it is often desirable to estimate the finite-ensemble effect on the verification score. For example, an experimental model configuration might be run at a small ensemble size to save computational resources. The skill calculated for the hindcast experiment is not representative of the skill that an operational ensemble with many members would achieve. A number of verification scores have been proposed in the past to estimate the finite ensemble effect, and correct it. These will be summarised in the present paper, and are implemented as R functions in the package **SpecsVerification**.

Comparative verification The value of a verification score is often of little interest by itself. If the value of the score is not zero, we know that the forecast was not perfect. But a score different from zero is often difficult to interpret without a reference point. For this reason, forecasts should be evaluated in a comparative way, by subjecting the forecast of interest to a simple reference forecast, such as the climatological average, or a simple time-series model. If the forecast is able to achieve a better score than the reference forecast, there is non-trivial information in the forecast. Forecast verification therefore often includes comparative measures of forecast performance. The package **SpecsVerification** includes functions to compare forecast verification scores.

Uncertainty quantification If the score was calculated from a finite number of forecast-observation pairs, the value of a verification score is of little use without a measure of uncertainty.

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

2. Ensemble-adjusted verification scores

2.1. Representation of ensemble and observation data

In **SpecsVerification**, archives of N instances of ensemble forecasts, each with R members, are represented by $N \times R$ matrices. The data shown in Figure 1 depicts 27 years or 24-member ensemble forecasts, i.e., a matrix with dimensions

```
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)
```

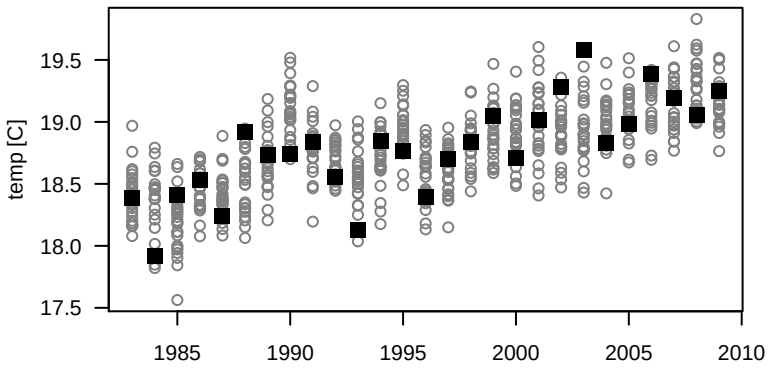

Figure 1: Seasonal European temperature forecasts by NCEP CFSv2, initialised in May, verified in JJA.

```
dim(ens)

## [1] 27 24
```

2.2. Binary forecasts

This section outlines the theory behind ensemble-adjusted verification scores, using probabilistic forecasts of binary events for illustration.

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 or non-occurrence of the event is coded as $y_t = 1$ or $y_t = 0$, respectively. The Brier score is given by the squared difference between forecast and observation:

$$s_B(p_t, y_t) = (p_t - y_t)^2 \tag{1}$$

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).

Assume next that instead of predicting the probability p_t, we make a prediction based on an ensemble forecast of size R, whose members were sampled identically and independently with probability p_t. That is, 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 given by the fraction i_t/R, where i_t is the number of successes, i.e. the number of ensemble members that predict the event $y_t = 1$. The Brier score of the estimated probability is equal to

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

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

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

That is, even though the fraction i_t/R is an unbiased estimator of the event probability p_t, the Brier score of i_t/R is not an unbiased estimator of the Brier score of p_t. The Brier score of i_t/R is in general larger 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 $R \to \infty$. The bias can thus 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) Brier score on average. 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. For example, if the hindcast ensemble has R members, and future operational forecasts will be made with $R^* > R$ ensemble members, the score calculated for the R-member hindcast ensemble will be a too pessimistic estimate of the score of the R^*-member forecast ensemble.

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

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

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

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

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 estimates the score of the underlying probability p_t. The ensemble-adjusted Brier score can be used to compare ensemble forecasting systems with different numbers of members. It further allows for the extrapolation of the average score of an ensemble forecast system to larger ensemble sizes.

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`.

We have transformed the continuous temperature data into binary forecasts by addressing the question "Will this year's summer be warmer than last year's"? To illustrate the finite ensemble effect and highlight the benefit of the finite-ensemble adjustment, we randomly split the 24-member forecast ensemble into a small 5-member ensemble and a larger 19-member ensemble. We then calculate their unadjusted Brier scores:

```
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)))
```

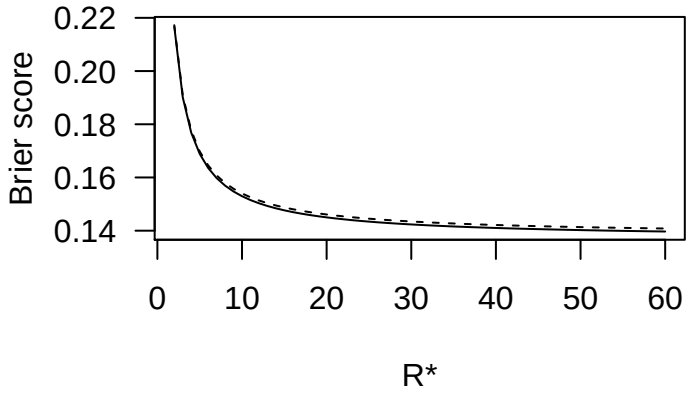

Figure 2: Ensemble-adjusted Brier score as a function of the adjusted ensemble size, calculated from a small 5-member ensemble (solid line) and a larger 19-member ensemble (dashed line), produced by the same forecasting system.

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

As expected, the large ensemble obtains a better score than the large 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 are able to estimate 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^*.

2.3. Categorical forecasts

Assume the ensemble forecasting system produces an ensemble of categorical rather than binary forecasts. That is, each ensemble members and the verifying observation falls into one of K classes. Two types of categorical forecasts can be distinguished: Disjoint categories and nested categories.

Assume the observation assumes on of K possible values, or classes, and a probabilistic forecast $\mathbf{p}_t = (p_{t,1}, \cdots, p_{t,K})$, is issued. 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 $y_{t,k} = 0$ otherwise. The quadratic score for such a probability forecast is given by

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

The quadratic score is simply the sum of Brier scores for the individual categories. Or stated differently, the Brier score is one-half the quadratic score of a 2-class categorical forecast.

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 $k = 1, \cdots, K$. Using results obtained for the ensemble-adjusted Brier score, the ensemble-adjusted quadratic score is seen to be

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

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

The quadratic score is insensitive to relabelling the K categories. Sometimes this is undesired in categorical forecasting problems. The forecast categories might have a natural ordering, such as low/medium/high intensity of precipitation. In that case, an order-sensitive score might be desired that penalises forecasts that assign most probability to the "high" category more than it penalises forecasts that assign most probability to the "medium" category, if the low category verifies. A lot of probability in the "medium" category can be considered closer to the observation than a lot of probability in the "high" category, such that the former forecast can be considered "better". An order-sensitive score is also desired if the forecast categories are nested within one another, e.g., for precipitation amounts categorised into $<$1mm/$<$5mm/$<$20mm/$<\infty$. In that case, a high probability assigned to the $<$5mm category should receive less penalty than a high probability assigned to the $<$20mm category, if the actual amount is $<$1mm.

The ranked probability score (RPS) is an order-sensitive quadratic score for categorical probability forecasts. 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 RPS is the quadratic score achieved by the cumulative forecast $\mathbf{j}_t$ for the cumulative observation $\mathbf{z}_t$. 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 ensures order-sensitivity of the score. Using results from the previous section, we get the ensemble-adjusted RPS

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

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.

We have transformed the continuous ensemble forecasts into categorical forecasts by addressing the question "Will this year's summer temperature be similar to last year's temperature (within a 0.5K range), colder, or warmer?" For example, the categorical prediction and observation for the year 2000 is

```
matrix(c(obs.cat["2000"], ens.cat["2000",]), ncol=1)

##       [,1]
##  [1,]    1
##  [2,]    2
##  [3,]    2
##  [4,]    2
##  [5,]    3
##  [6,]    2
##  [7,]    2
##  [8,]    2
##  [9,]    2
## [10,]    2
## [11,]    1
## [12,]    1
## [13,]    2
## [14,]    2
## [15,]    2
## [16,]    1
## [17,]    2
## [18,]    1
## [19,]    2
## [20,]    1
## [21,]    2
## [22,]    2
## [23,]    2
## [24,]    1
## [25,]    2
```

This transforms the forecast problem into a 3-category problem, which we first evaluate by the ensemble-adjusted quadratic score:

```
mean(EnsQs(ens.cat, obs.cat))

## [1] 0.5782
```

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

##             [,1]
## small.ens 0.6519
## large.ens 0.5856
```

```
rbind(
small.ens=mean(EnsRps(ens.cat[, i.small], obs.cat)),
large.ens=mean(EnsRps(ens.cat[, i.large], obs.cat)))

##             [,1]
## small.ens 0.3719
## large.ens 0.3386
```

2.4. Continuous 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

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

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 $x_t = \{x_{t,1}, x_{t,2} \ldots, x_{t,R}\}$ 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

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

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

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

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

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

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. The ensemble-adjusted CRPS is implemented in the **SpecsVerification** function `EnsCrps`.

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

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

The ensemble adjusted Ignorance score has recently been proposed (**?**).

3. Comparative verification and uncertainty quantification

3.1. Reference forecast

The value of a verification score by itself is meaningless. In order to evaluate the skill of a forecast, its verification score has to be compared to the score achieved by a reference forecast. For example, if the skill of a state-of-the-art high resolution climate model is evaluated, it is reasonable to compare its verification score to the score achieved by an older climate model, possibly with lower resolution and less physical detail.

In the absence of a dynamical climate model to which the score can be compared, simple statistical benchmark predictions can be used. A popular simple reference forecast is the climatological forecast, which is only based on the known record of observations, without reference to any numerical forecast model. **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)
```

The new data set of climatological ensembles can be used as a reference ensemble to which the numerical forecast ensemble can be compared. We recommend also considering statistical reference forecasts such as a linear trend or an auto-regressive model, which might be more suitable than the climatological forecast.

3.2. Mean scores and mean score differences

Suppose we have calculated two time series $\{s_{1,1}, s_{1,2}, \ldots, s_{1,N}\}$ and $\{s_{2,1}, s_{2,2}, \ldots, s_{2,N}\}$ of verification scores for two competing forecast systems for the same observation. Diebold and

Mariano (1995) suggest to test the null-hypothesis of equal forecast accuracy using the time series $d_1, \ldots, d_N$ of loss differentials $d_t = s_{1,t} - s_{2,t}$. Define $\bar{d}$ to be the empirical average over $d_1, d_2, \cdots$. Under the assumption of temporal independence of d_t, and zero mean of the loss-differential, the test statistic

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

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     ci.L   ci.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
```

How do the score differences change if adjust the ensemble size to infinity?

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

##       score.diff score.diff.sd    p.value     ci.L   ci.U
## brier    0.11907       0.04206 0.00232081  0.03663 0.2015
## qs       0.11198       0.08959 0.10568147 -0.06362 0.2876
## rps      0.09529       0.06438 0.06942387 -0.03089 0.2215
## crps     0.09050       0.02399 0.00008094  0.04348 0.1375
```

3.3. 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. The skill score is then given by the score difference between the reference forecast and the evaluated forecast, normalised by the difference between the reference forecast and the perfect forecast:

$$SS = \frac{S_{ref} - S}{S_{ref} - S_{perf}} \tag{14}$$

The variance of the skill score can be estimated by error propagation as follows

$$var(SS) \approx \frac{1}{(S_{ref} - S_{perf})^2} var(S) + \frac{(S - S_{perf})^2}{(S_{ref} - S_{perf})^2} var(S_{ref}) - 2\frac{S - S_{perf}}{(S_{ref} - S_{perf})^3} cov(S, S_{ref}) \tag{15}$$

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. The skill score is implemented in **SpecsVerification** in the function `SkillScore`, which takes as inputs two vectors of verification scores of the evaluated and the reference forecast, the constant score achieved by a perfect forecaster, 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
```

The skill scores increase slightly if we adjust the ensemble size to $R^* \to \infty$ for the ensemble forecasts and the climatological reference forecast:

```
rbind(
  brier = SkillScore(EnsBrier(ens.bin, obs.bin, R.new=Inf),
                     EnsBrier(ens.bin.ref, obs.bin, R.new=Inf)),
  qs    = SkillScore(EnsQs(ens.cat, obs.cat, R.new=Inf),
```

```
                    EnsQs(ens.cat.ref, obs.cat, R.new=Inf)),
  rps   = SkillScore(EnsRps(ens.cat, obs.cat, R.new=Inf),
                    EnsRps(ens.cat.ref, obs.cat, R.new=Inf)),
  crps  = SkillScore(EnsCrps(ens, obs, R.new=Inf),
                    EnsCrps(ens.ref, obs, R.new=Inf))
)

##       skillscore skillscore.sd
## brier     0.4749       0.15475
## qs        0.1665       0.13519
## rps       0.2234       0.14806
## crps      0.4051       0.07602
```

3.4. Correlation and correlation difference

The Pearson (moment-product) 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, **SpecsVerification** provides the function `Corr`, which returns the sample correlation coefficient r_{xy}, a one-sided p-value using the test statistic

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

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

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

where $z_{xy} = atanh(r_{xy})$ 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)

##         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_{by} 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 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:

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

The test statistic

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

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 based on Zou (2007) are calculated. First estimate the correlation between the correlation coefficients r_{ay} and r_{by} by

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

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 eq. 17. An approximate $(1 - \alpha) \times 100\%$ confidence interval (L, U) for the correlation difference $r_{by} - r_{ay}$ is then given by

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

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)

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

3.5. 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 $t = 1, \ldots, N$. (The forecasts are allowed to take values on the real line, and need not be probabilitities.) 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 rth 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 forecasts 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 $X_i^{(r)}$, $i = 1, \cdots, m$ the elements of $C_1^{(r)}$ and by $Y_i^{(r)}$, $i = 1, \cdots, n$ the members of $C_0^{(r)}$. Define the function Ψ as

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

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

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

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

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

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

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

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

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

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

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

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

Note that the AUC is asymptotically Normally distributed. The estimated variance can therefore be used to construct a confidence interval, i.e., $\hat{\theta} \pm 1.96\sqrt{var(\hat{\theta})}$ 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. We expect ensemble size to effect AUC, but we are not aware of published results to adjust the AUC.

4. 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.

```
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 histgram can be assessed by a Pearson χ^2-test (Pearson 1900). Suppose rank i was observed r_i times for $i = 1, \ldots, R+1$, and define $e_i = N/(R+1)\ \forall\ i$ the expected number of counts if each verification rank were equally likely. Define further

$$x_i = \frac{r_i - e_i}{\sqrt{e_i}}. \tag{29}$$

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

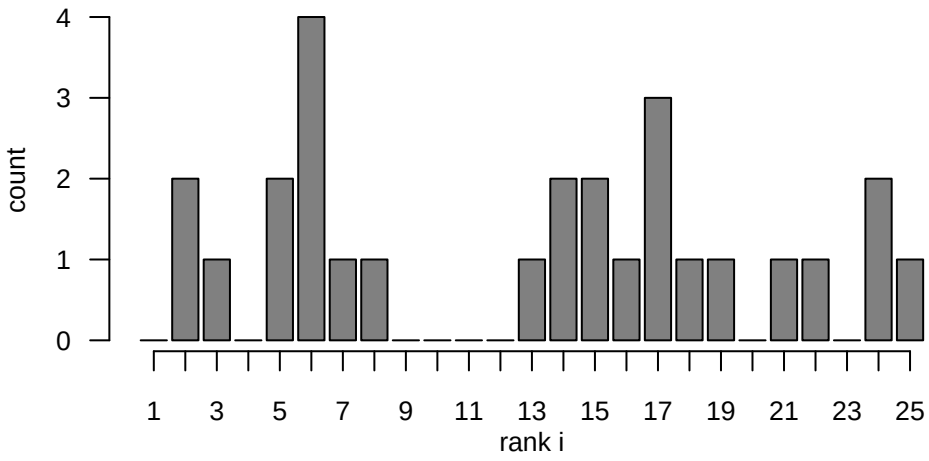

Figure 3: Rank histogram

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

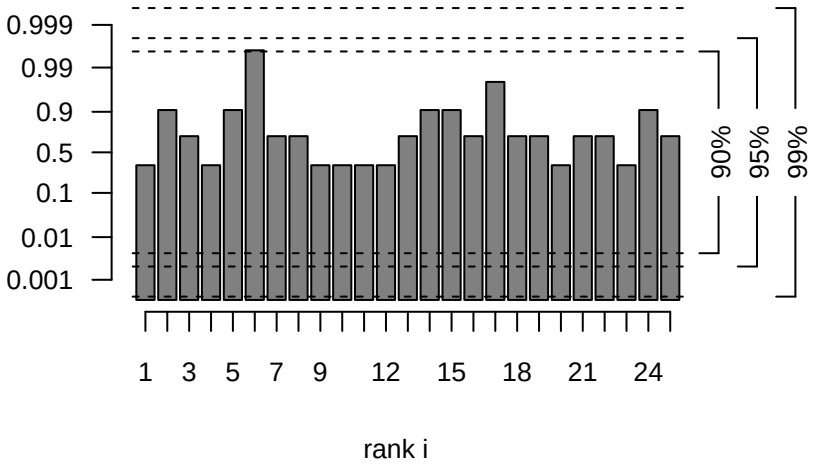

Figure 4: Rank histogram on probability paper

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

$$\chi^2 = \sum_{i=1}^{R+1} x_i^2 \tag{30}$$

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

Hamill (2001) showed that certain types of violation of ensemble reliability are visible as different patterns in the rank histogram. In particular, a constant bias of the mean produces sloped rank histograms, and ensembles with insufficient (excessive) ensemble spread produce $\cup$-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 $\mathbf{c}^{(1)}$, $\mathbf{c}^{(2)}$, ..., the test statistics $(\sum_i c_i^{(k)} x_i)^2$ are independently χ^2 distributed with one d.o.f. The function `TestRankhist` uses 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 $i = 1, \cdots, J$ are given by

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

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

The χ^2 test using the linear contrast is sensitive to sloped rank histgrams, 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.

5. 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 foredcasts 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.avgs 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 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).

6. 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 additional functions for recalibration and verification of forecasts. 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, and can be analysed by the functions `ScoreDiff` and `SkillScore`.

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

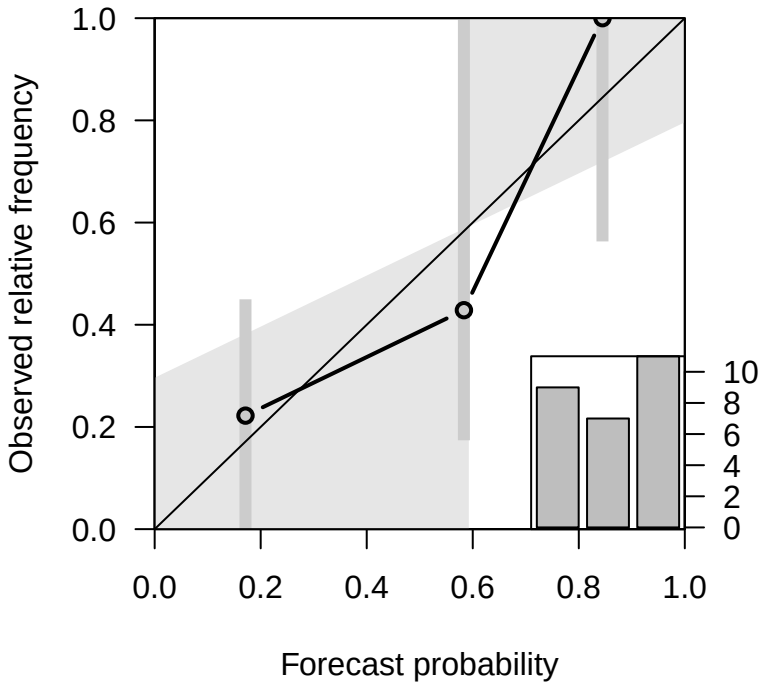

Figure 5: Reliability diagram

7. Conclusion

Acknowledgments

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. 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. URL http://dx.doi.org/10.1175/WAF993.1.

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. URL 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. 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.

Ferro CAT (2013). "Fair scores for ensemble forecasts." *Quarterly Journal of the Royal Meteorological Society*, **140**(683), 1917–1923. doi: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.

Fricker TE, Ferro CAT, Stephenson DB (2013). "Three recommendations for evaluating climate predictions." *Meteorological Applications*, **20**(2), 246–255. doi:10.1002/met.1409.

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.

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. URL 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. URL 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. 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.

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. 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.

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. URL http://dx.doi.org/10.1080/14786440009463897.

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

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd