The **SpecsVerification** package: Verification of ensemble hindcasts and more

Stefan Siegert

January 2014

Contents

1 Install and load

```
install.packages("/home/stefan/folders/specs/r-package/specs-verification-git",
                 repo=NULL, lib="/tmp", type="source")
library("SpecsVerification", lib.loc="/tmp")
```

2 Load some example data

We will use seasonal ensemble forecast of surface temperature averaged over the `Atl3` region between 1993 and 2009, initialized twice a year (1 May and 1 November) and run 1 up to 4 months into the future. "Observations" are generated from the `ERAint` data set. Ensemble forecasts are generated by `ECMWF System4` (15 members) and an anomaly initialized ensemble generated at IC3. The data can be downloaded from the SPECS wiki: Atl3.ERAint.Rdata

```r
file <- "Atl3.ERAint.Rdata"
if (!file.exists(file)) {
  download.file(url="http://www.specs-fp7.eu/wiki/images/a/a8/Atl3.ERAint.Rdata",
                destfile=file)
}
load(file)

dimnames(obs) # the dimensions of obs are 1:initdate, 2:leadtime
```

```
## [[1]]
##  [1] "1993-05-01" "1993-11-01" "1994-05-01" "1994-11-01" "1995-05-01"
##  [6] "1995-11-01" "1996-05-01" "1996-11-01" "1997-05-01" "1997-11-01"
## [11] "1998-05-01" "1998-11-01" "1999-05-01" "1999-11-01" "2000-05-01"
## [16] "2000-11-01" "2001-05-01" "2001-11-01" "2002-05-01" "2002-11-01"
## [21] "2003-05-01" "2003-11-01" "2004-05-01" "2004-11-01" "2005-05-01"
## [26] "2005-11-01" "2006-05-01" "2006-11-01" "2007-05-01" "2007-11-01"
## [31] "2008-05-01" "2008-11-01" "2009-05-01" "2009-11-01"
##
## [[2]]
## [1] "1" "2" "3" "4"
```

```r
dimnames(ens) # the dimensions of ens are 1:initdate, 2:leadtime, 3:ensemble, 4:member
```

```
## [[1]]
##  [1] "1993-05-01" "1993-11-01" "1994-05-01" "1994-11-01" "1995-05-01"
##  [6] "1995-11-01" "1996-05-01" "1996-11-01" "1997-05-01" "1997-11-01"
## [11] "1998-05-01" "1998-11-01" "1999-05-01" "1999-11-01" "2000-05-01"
## [16] "2000-11-01" "2001-05-01" "2001-11-01" "2002-05-01" "2002-11-01"
## [21] "2003-05-01" "2003-11-01" "2004-05-01" "2004-11-01" "2005-05-01"
## [26] "2005-11-01" "2006-05-01" "2006-11-01" "2007-05-01" "2007-11-01"
## [31] "2008-05-01" "2008-11-01" "2009-05-01" "2009-11-01"
##
## [[2]]
## [1] "1" "2" "3" "4"
##
## [[3]]
## [1] "ecmwf" "l00w"
##
## [[4]]
##  [1] "member.1"  "member.2"  "member.3"  "member.4"  "member.5"
##  [6] "member.6"  "member.7"  "member.8"  "member.9"  "member.10"
## [11] "member.11" "member.12" "member.13" "member.14" "member.15"
```

```r
t <- paste(seq.Date(from=as.Date("1993-05-01"),
          to=as.Date("2009-05-01"), by="1 year"))
ecmwf <- ens[t,"1","ecmwf",]
```

```
l00w <- ens[t,"1","l00w",1:5]
ver <- obs[t,"1"]
plot(NULL, xlim=c(1, length(ver)), ylim=range(c(ver,l00w,ecmwf)),
     xlab="time", ylab="Atl3 temp. [C]")
points(ver, pch=15, cex=1.5, col=gray(.5))
for (i in 1:nrow(ecmwf)) points(rep(i-.2,15), ecmwf[i,], pch=15, cex=0.5)
for (i in 1:nrow(ecmwf)) lines(rep(i-.2,2), range(ecmwf[i,]))
for (i in 1:nrow(l00w)) points(rep(i+.2,5), l00w[i,], pch=16, cex=0.5)
for (i in 1:nrow(l00w)) lines(rep(i+.2,2), range(l00w[i,]))
```

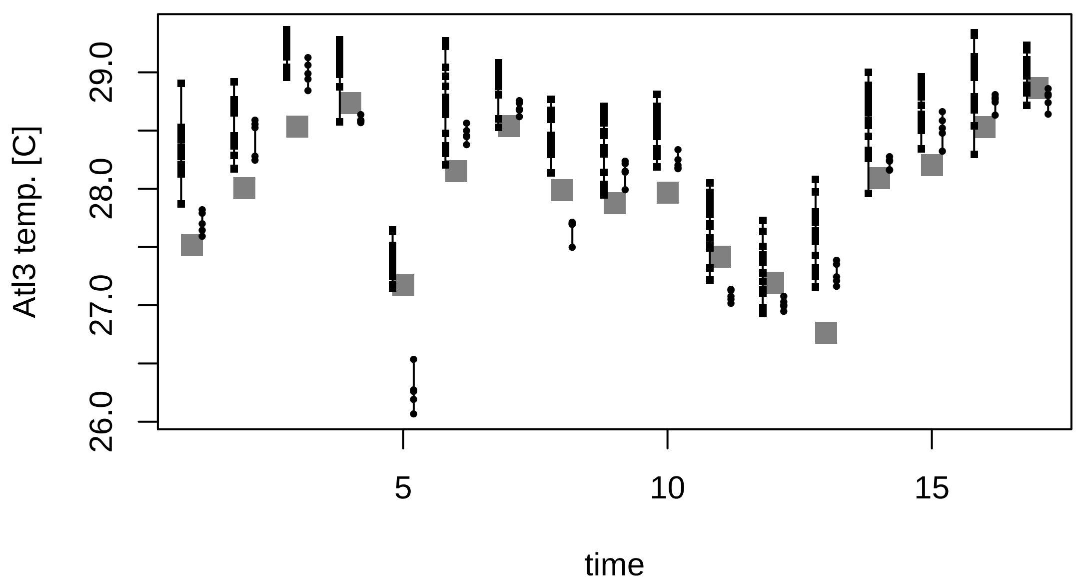

Figure 1: Time series of the ensemble data and observations

3 Fair Brier Score analysis

The function `FairBrier` returns individual values of fair Brier scores of ensembles and their verifications. The argument `tau` is the threshold to whose exceedance defines the binary event:

```
fbr.ecmwf <- FairBrier(ens=ecmwf, obs=ver, tau=28.5)
fbr.l00w <- FairBrier(ens=l00w, obs=ver, tau=28.5)
plot(fbr.ecmwf, type="b", pch=15, col=gray(.5))
lines(fbr.l00w, type="b", pch=16, col="black")
```

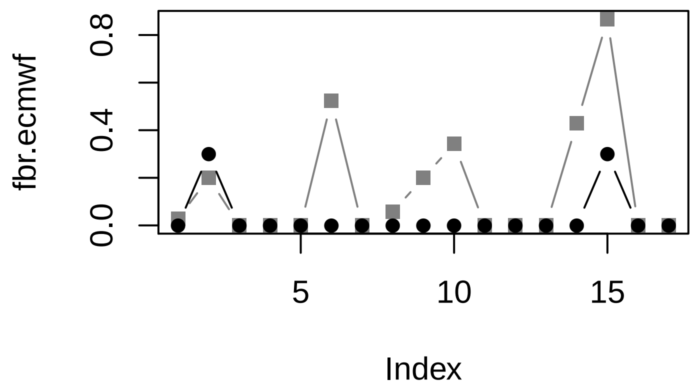

Figure 2: Plot of fair Brier Scores of ECMWF (gray) and l00w (black).

```
print(c(mean(fbr.ecmwf), mean(fbr.l00w)))
```

```
## [1] 0.15574 0.03529
```

The function `AnalyzeFairBrier` returns the mean fair Brier score difference and optional estimated quantiles of the sampling distribution of the mean difference:

```
FairBrierDiff(ens=l00w, ens.ref=ecmwf, obs=ver,
                 tau=28.5, probs=c(0.05, 0.95))
```

```
## $br.diff
## [1] 0.1204
##
## $sampling.quantiles
##    0.05    0.95
## 0.03162 0.20928
```

4 Fair CRPS analysis

The fair continuously ranked probability score for ensemble forecasts has similar routines as the fair Brier score:

```
fcrps.ecmwf <- FairCrps(ens=ecmwf, obs=ver)
fcrps.l00w <- FairCrps(ens=l00w, obs=ver)
plot(fcrps.ecmwf, type="b", pch=15, col=gray(.5))
lines(fcrps.l00w, type="b", pch=16, col="black")
```

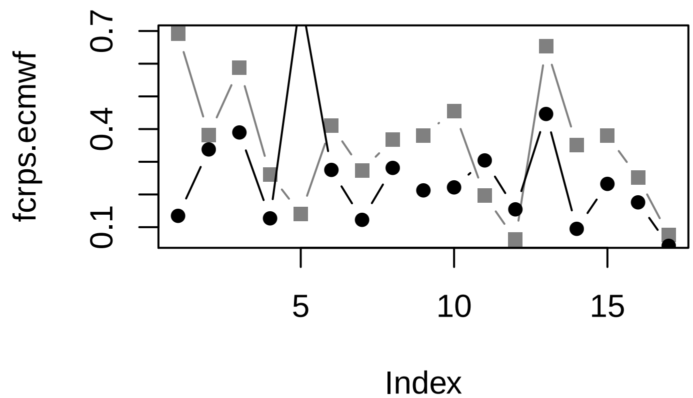

Figure 3: Plot of fair CRPS of ECMWF (gray) and l00w (black).

```
print(c(mean(fcrps.ecmwf), mean(fcrps.l00w)))
```

```
## [1] 0.3485 0.2566
```

```
print(FairCrpsDiff(ens=l00w, ens.ref=ecmwf, obs=ver, probs=c(0.05, 0.95)))
```

```
## $crps.diff
## [1] 0.0919
##
## $sampling.quantiles
##     0.05     0.95
## -0.01167  0.19547
```

5 Rank histogram analysis

fix the margin issue!!!

The ECMWF ensemble and the observations are transformed to anomalies by centering them around zero. Then the rank histogram is drawn in the "raw" version and on probability paper:

```
ecmwf <- ens[,1,"ecmwf",]
ecmwf <- ecmwf - mean(ecmwf)
ver <- obs[,1]
ver <- ver - mean(ver)
rh <- Rankhist(ens=ecmwf, obs=ver)
PlotRankhist(rh, mode="raw")
```

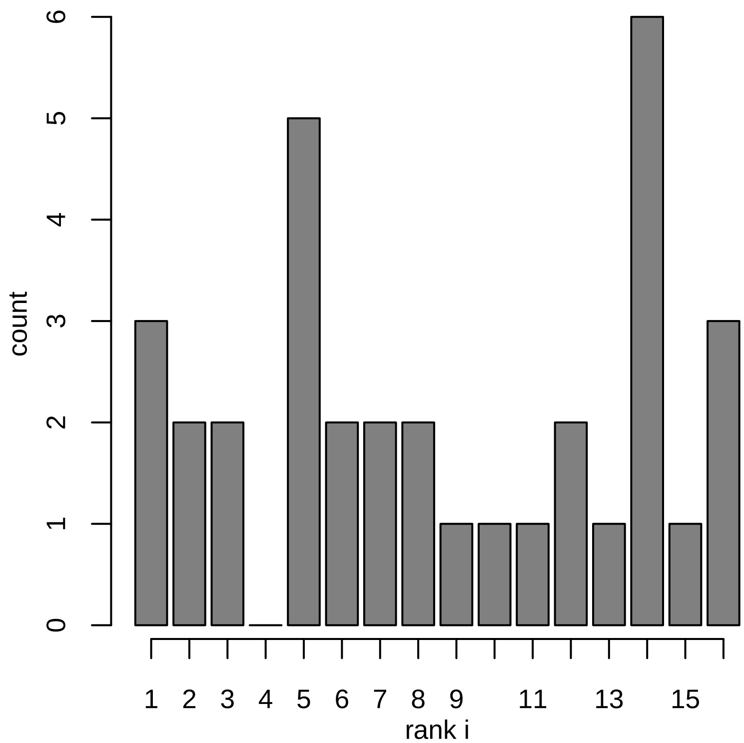

Figure 4: plot of chunk rankhist

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

6 Reliability diagram analysis

The experimental l00w ensemble and the observations are transformed to anomalies. The event of interest is the exceedance of a value of 1 and probabilities are generated for this event by counting ensemble members:

```
l00w <- ens[,4,"l00w",]
l00w <- l00w - mean(l00w, na.rm=TRUE)
p <- rowMeans(l00w > 1, na.rm=TRUE)
ver <- obs[,4]
ver <- ver - mean(ver)
y <- 1 * (ver > 1)
rd <- ReliabilityDiagram(probs=p, obs=y, bins=c(0,1/3,2/3,1), plot=TRUE, nboot=10000, mc.cores=
```

```
print(rd)
```

```
##    p.avgs cond.probs cbar.lo cbar.hi
```

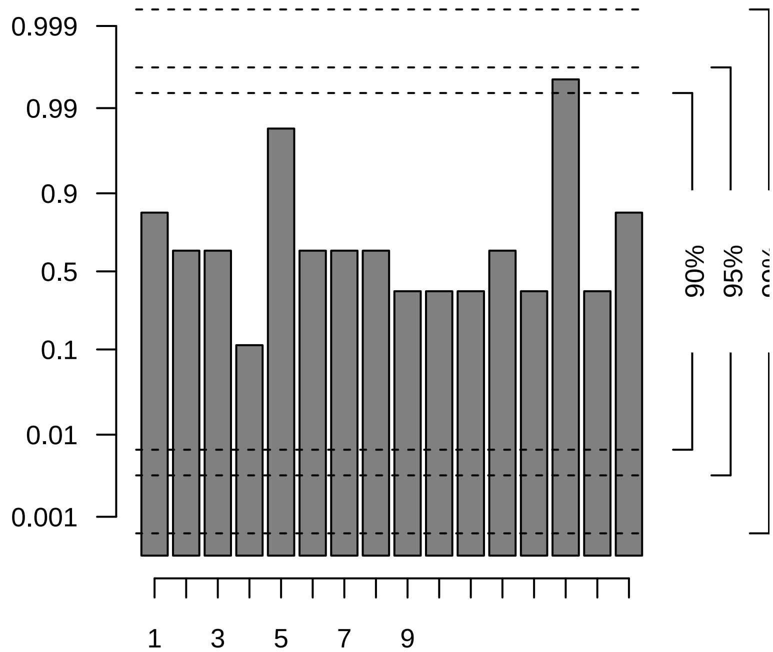

Figure 5: plot of chunk rankhist

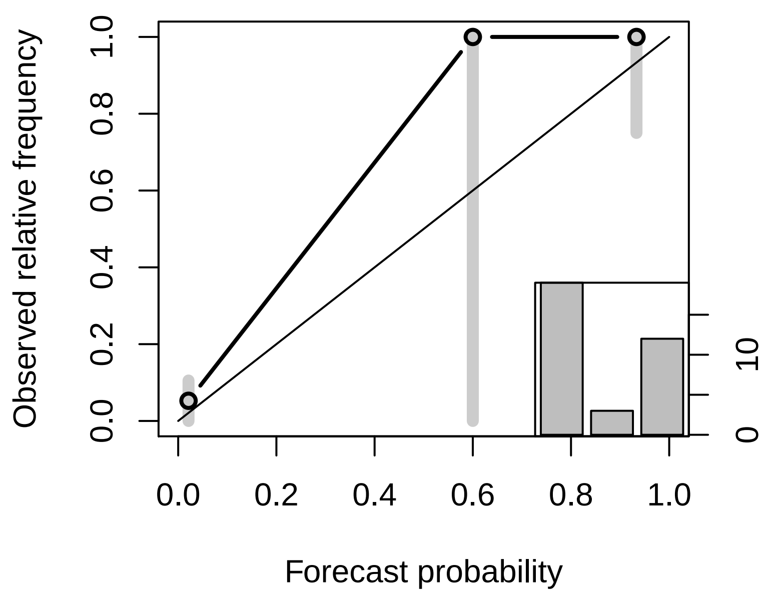

Figure 6: plot of chunk reldiag

```
## 1 0.02105    0.05263    0.00  0.1053
## 2 0.60000    1.00000    0.00  1.0000
## 3 0.93333    1.00000    0.75  1.0000
```