Probabilistic seasonal forecast verification

Caio Coelho

Centro de Previsão de Tempo e Estudos Climáticos (CPTEC)

Instituto Nacional de Pesquisas Espaciais (INPE)

Plan of lecture
- **Introduction: Examples of forecasts**
- **Brier score and its decomposition: reliability, resolution and uncertainty**
- **Reliability diagram**
- **Exercise on Brier score, its decomposition and reliability diagram**
- **ROC: discrimination**
- **Exercise on ROC**

MedCOF Training Workshop on Verification of Operational Seasonal Forecasts in the Mediterranean region

Rome, Italy, 15-18 November 2016

Examples of forecasts

- Deterministic forecasts for Jakarta:

  Tomorrow´s max. temperature forecast: 32 Celsius

  Season (JJA) average temperature forecast: 26.5 Cesius

  Season (JJA) total precipitation forecast: 200 mm

  Verification: comparison of fsct and obs values using deterministic scores

- Probabilistic forecasts for Jakarta

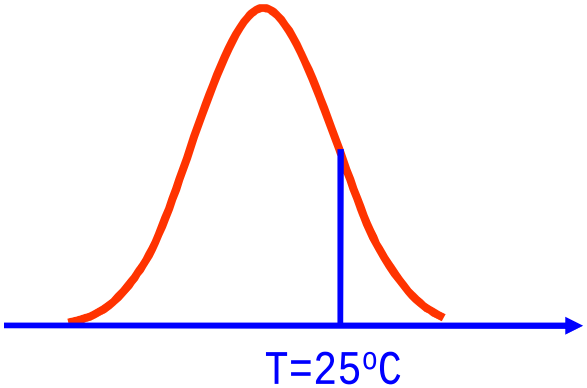
  Probability of tomorrow´s max. temperature to be above 30 Celsius is 90%

  Probability of next season (JJA) ave. temp. to be above 26.5 Cesius is 40%

  Probability of next season (JJA) total. precip. to be below 70 mm is 30%

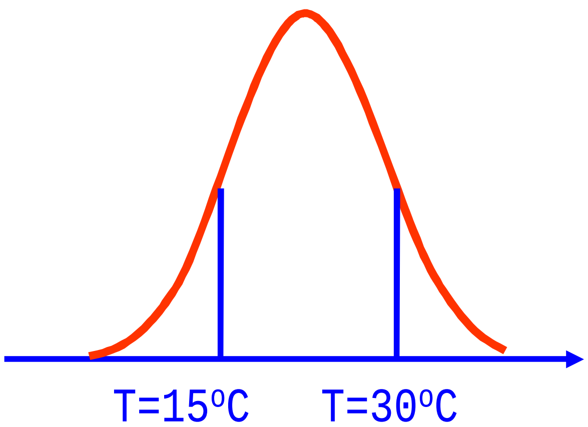
  Verification: comparing of fsct prob and occurrence (or non-occurrence) of event using probabilistic scores

Examples of probabilistic seasonal forecasts: JJA 2mT



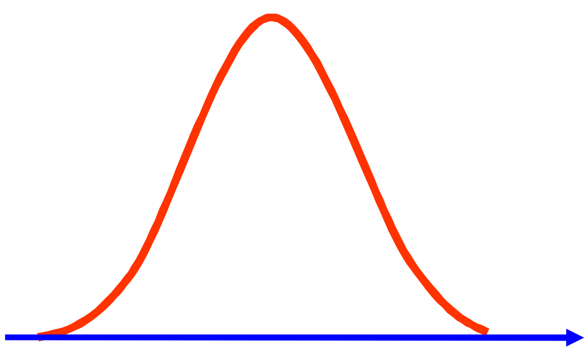
F is a set of probabilities
for the discrete values of O

F: 0.4, 0.3, 0.5, 0.1, 0.6, 0.2
O: 1 , 1 , 0 , 1 , 0 , 0



F is a probabilistic interval
of values for O (interval forecast)

F: 0.7, 0.6, 0.5, 0.8, 0.7, 0.5
O: 0 , 1 , 0 , 1 , 1 , 0



F is a full probability distribution
function for O

Probability scores

Imagine the following set of probability forecasts for a series k=1,2,...,n=6 of binary events:

k	1	2	3	4	5	6
o	0	1	1	0	0	0
p	0.7	0.6	0.2	0.8	0.9	0.3

The forecast skill can be measured using scores such as:

$$BS = \frac{1}{n}\sum_{k=1}^{n}(p_k - o_k)^2 \quad Brier\ \ score$$

$$A = \frac{1}{n}\sum_{k=1}^{n}| p_k - o_k | \quad Mean\ \ Absolute\ \ score$$

$$C = \frac{1}{n}\sum_{k=1}^{n}(-(1-o_k)\log(1-p_k) - o_k \log p_k) \quad Logarithmic\ \ score$$

Note: small values indicate good quality forecasts!
For a perfect forecast p=o and the score equals zero.

Skill Scores

The scores are often presented as skill scores by using the linear transformation:

$$SS = \frac{S - S_{ref}}{S_{best} - S_{ref}} = \frac{S - S_{ref}}{0 - S_{ref}} = 1 - \frac{S}{S_{ref}}$$

where S_{ref} is the value of the score for some *unskilful* reference forecast such as:

- Issuing the same constant probability each time
- Issuing random probabilities each time

Note: both of these can be thought of as sampling a probability from a distribution (constant probability is a special limit of zero width distribution)

Skill scores allow easy interpretation of forecasts:

0 → no skill forecast

1 → perfect forecast

$BSS=1-BS/BS_{clim}$

Forecast attributes assessed with the Brier score and reliability diagram

- Reliability: correspondence between forecast probabilities and observed relative frequency (e.g. an event must occur on 30% of the occasions that the 30% forecast probability was issued)

- Resolution: Conditioning of observed outcome on the forecasts
- Addresses the question: Does the frequency of occurrence of an event differs as the forecast probability changes?
- If the event occurs with the same relative frequency regardless of the forecast, the forecasts are said to have no resolution
- Forecasts with no resolution are useless because the outcome is the same regardless of what is forecast

Brier Score decomposition (Murphy, 1973)

$$BS = \frac{1}{n}\sum_{k=1}^{n}(p_k - o_k)^2 \qquad 0 \le BS \le 1$$

$$BS = \underbrace{\frac{1}{n}\sum_{i=1}^{l} N_i (p_i - \overline{o}_i)^2}_{\text{Reliability}} - \underbrace{\frac{1}{n}\sum_{i=1}^{l} N_i (\overline{o}_i - \overline{o})^2}_{\text{Resolution}} + \underbrace{\overline{o}(1-\overline{o})}_{\text{Uncert.}}$$

$$\overline{o}_i = p(o_1 \mid p_i) = \frac{1}{N_i}\sum_{k \in N_i} o_k \qquad \overline{o} = \frac{1}{n}\sum_{k=1}^{n} o_k$$

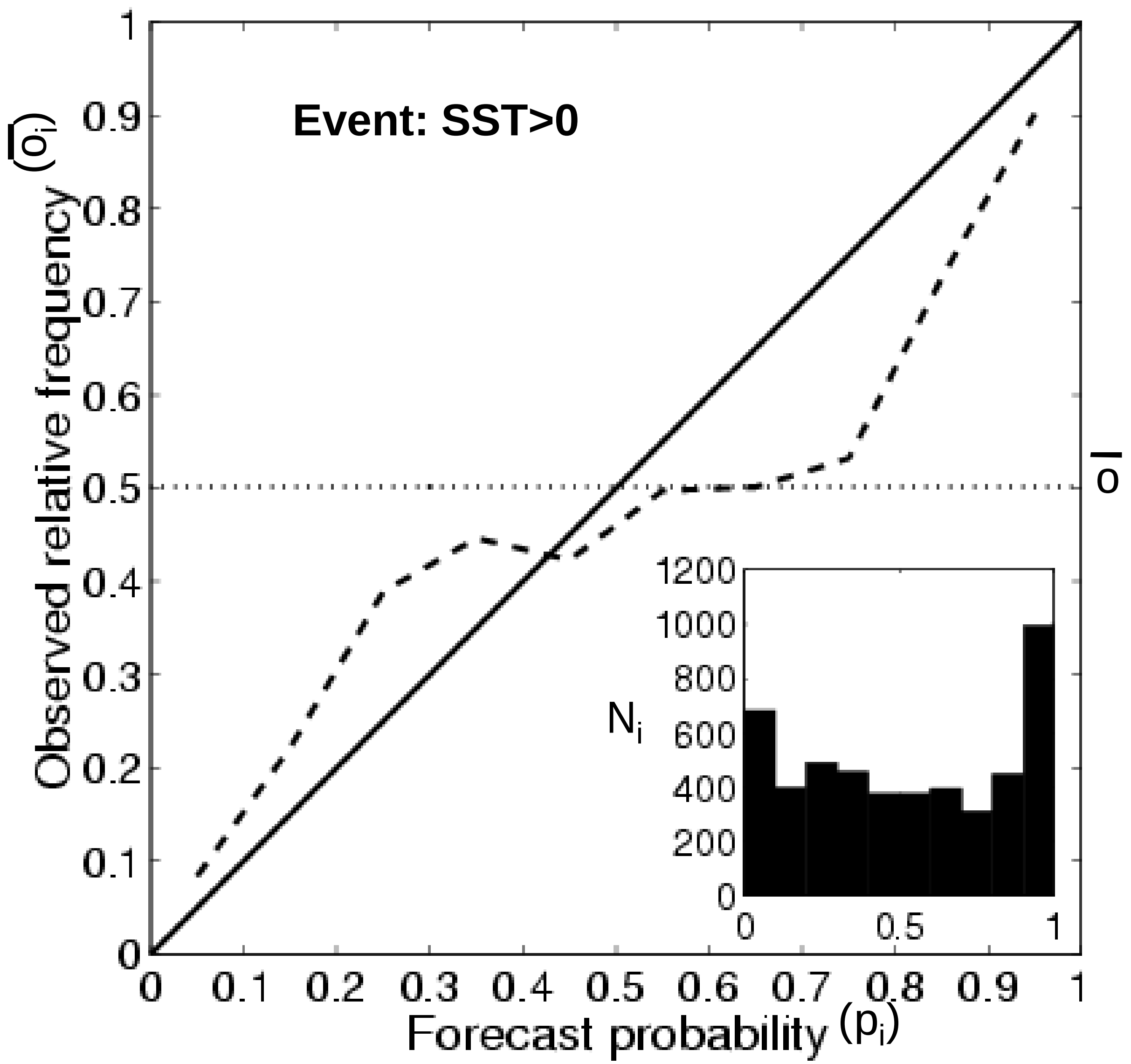
$$i = 1,\ldots,l = 11: p_1 = 0,\ p_2 = 0.1,\ p_3 = 0.2,\ldots,\ p_{10} = 0.9,\ p_{11} = 1$$

The Brier score can be improved (reduced):

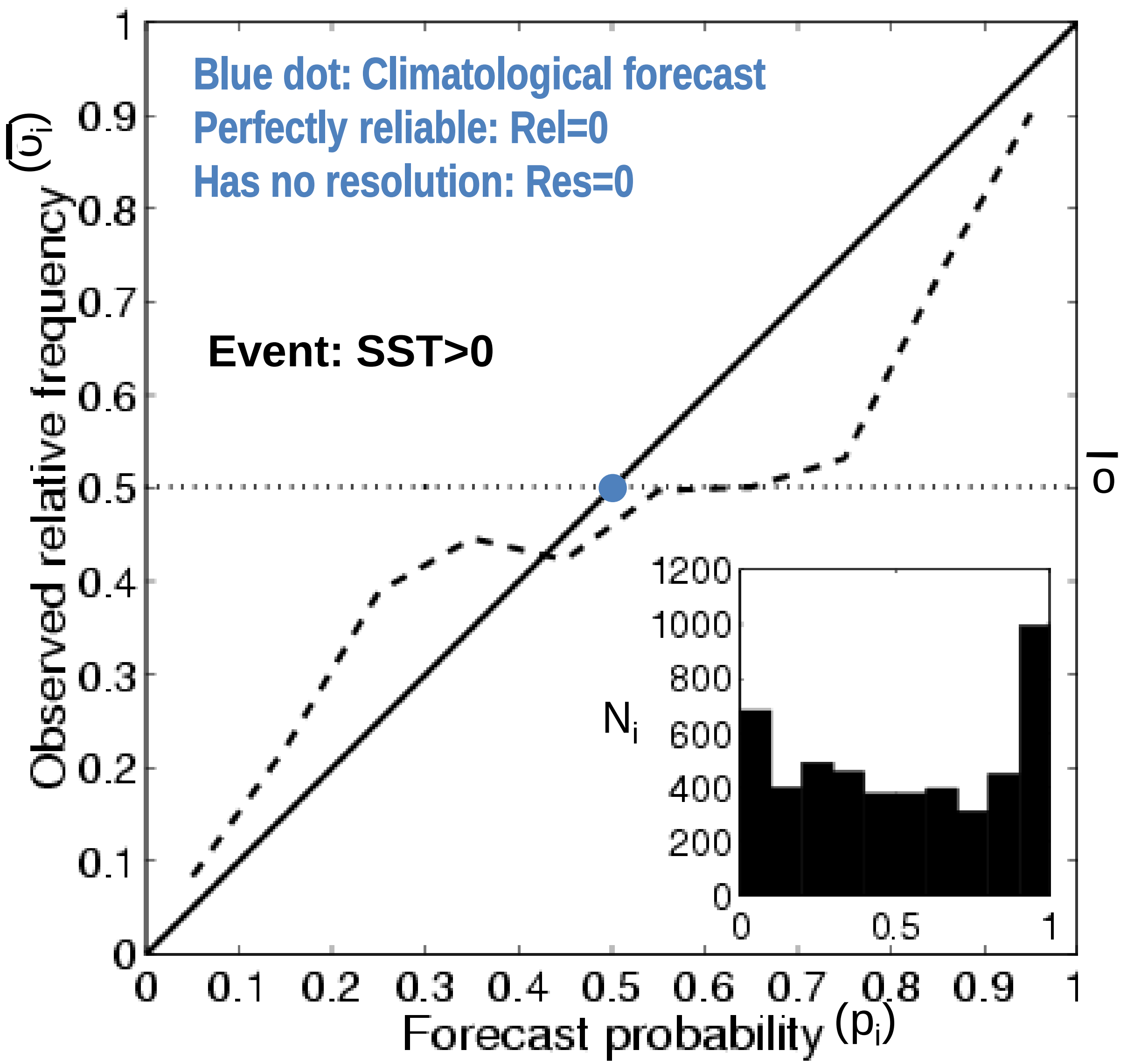
- forecasting events of small var(o)=$\overline{o}(1-\overline{o})$ (reduced uncertainty)
- increasing *resolution* (eg. combining forecasts)
- improving *reliability* (eg. calibrating forecasts)

Note: It is common practice to decompose the Brier score in reliability and resolution for examining which component can be improved

Reliability diagram



Reliability diagram

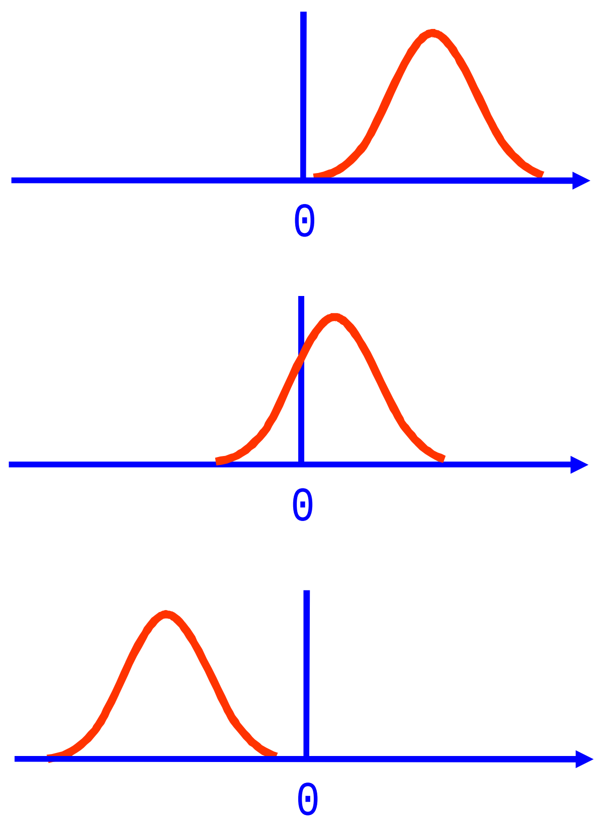


Example of how to construct a reliability diagram

Sample of probability forecasts:

22 years x 3000 grid points = 66000 forecasts

How many times the event (T>0) was forecast with probability p_i?



Forecast Prob.(p_i)	# Fcsts. N_i	"Perfect fcst." OBS-Freq.($\overline{o}_i$)	"Real fcst." OBS-Freq($\overline{o}_i$)
100%	8000	8000 (100%)	7200 (90%)
90%	5000	4500 (90%)	4000 (80%)
80%	4500	3600 (80%)	3000 (66%)
….	….	….	….
….	….	….	….
….	….	….	….
10%	5500	550 (10%)	800 (15%)
0%	7000	0 (0%)	700 (10%)

Courtesy: Francisco Doblas-Reyes

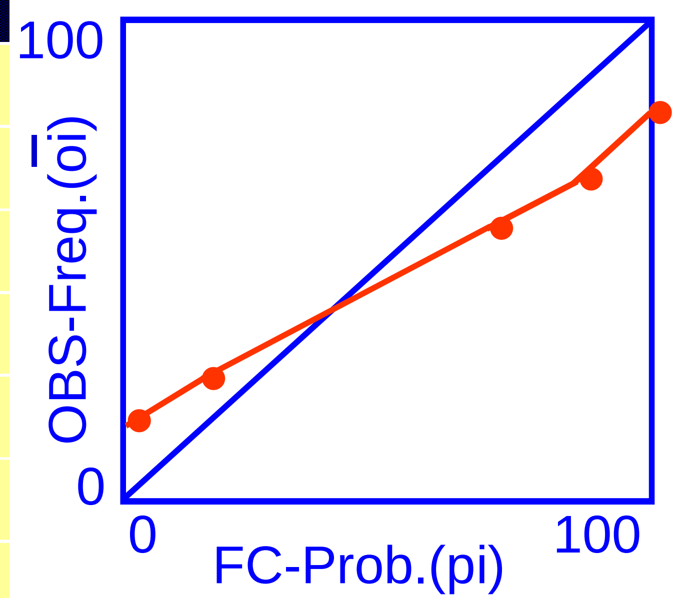
Example of how to construct a reliability diagram

Sample of probability forecasts:

22 years x 3000 grid points = 66000 forecasts

How many times the event (T>0) was forecast with probability p_i?

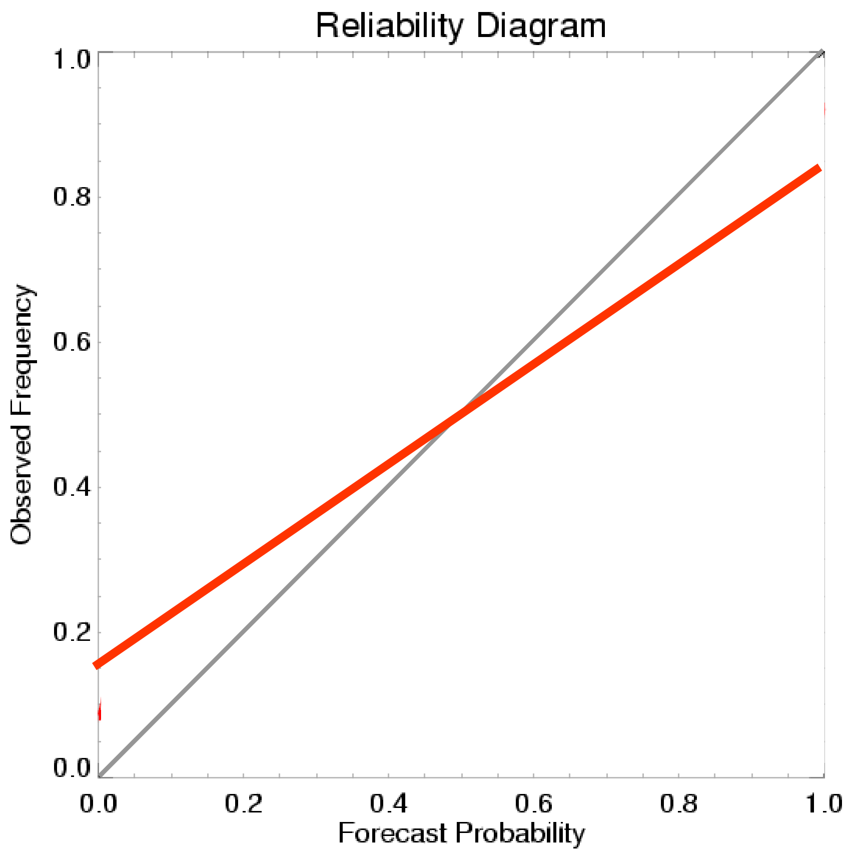
Forecast Prob.(p_i)	# Fcsts. N_i	“Perfect fcst.” OBS-Freq.($\bar{o}_i$)	“Real fcst.” OBS-Freq($\bar{o}_i$)
100%	8000	8000 (100%)	7200 (90%)
90%	5000	4500 (90%)	4000 (80%)
80%	4500	3600 (80%)	3000 (66%)
….	….	….	….
….	….	….	….
….	….	….	….
10%	5500	550 (10%)	800 (15%)
0%	7000	0 (0%)	700 (10%)



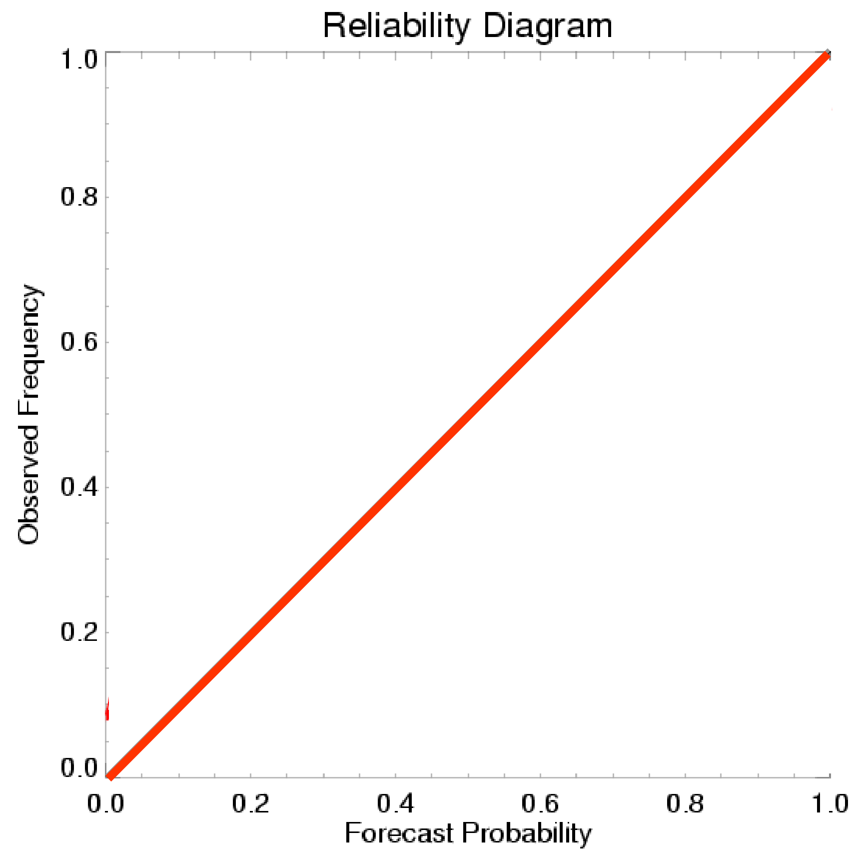
Courtesy: Francisco Doblas-Reyes

Reliability diagram

Over-confident forecasts



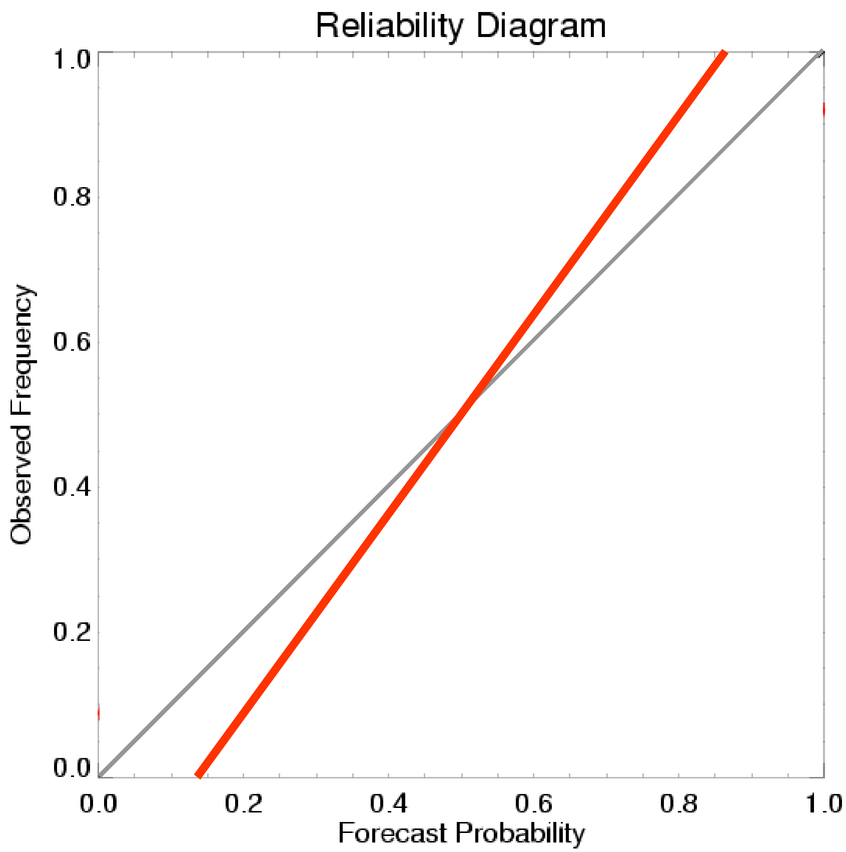
Perfect forecasts



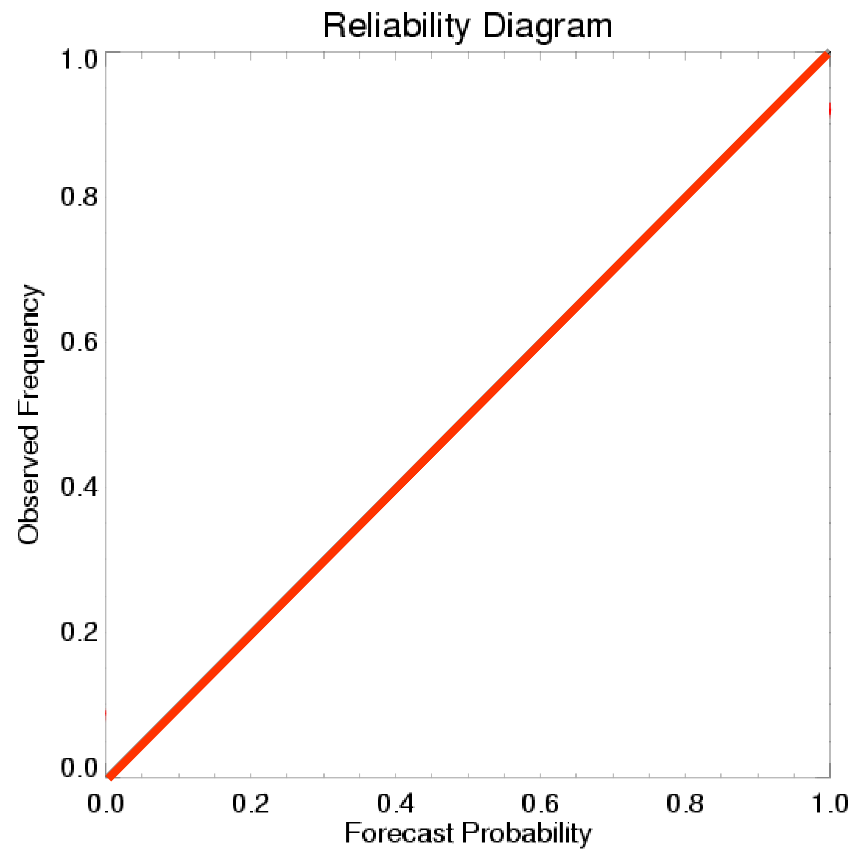
Courtesy: Francisco Doblas-Reyes

Reliability diagram

Under-confident forecasts



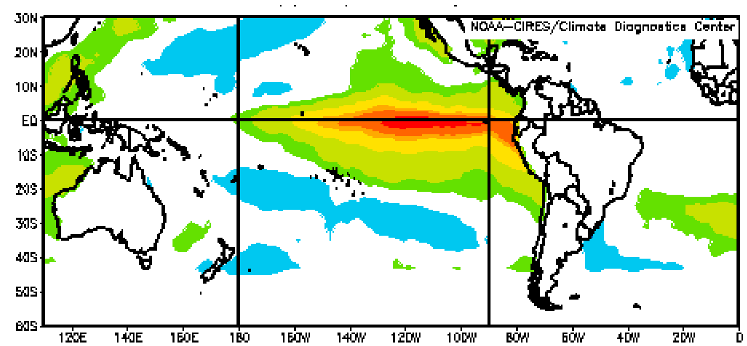
Perfect forecasts



Courtesy: Francisco Doblas-Reyes

Example:Equatorial Pacific SST

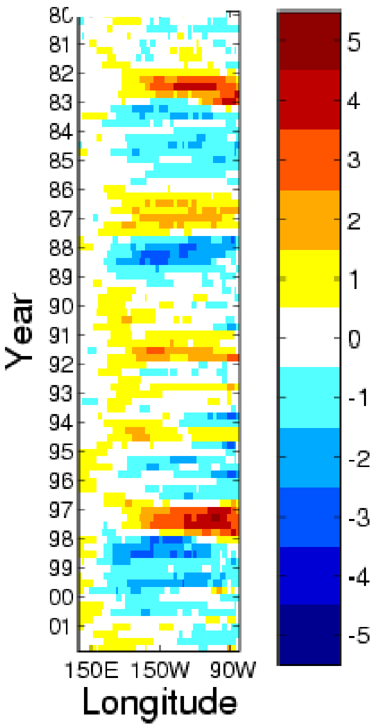
88 seasonal probability forecasts of binary SST anomalies at 56 grid points along the equatorial Pacific. Total of 4928 forecasts.
6-month lead forecasts for 4 start dates (F,M,A,N) valid for (Jul,Oct,Jan,Aug)



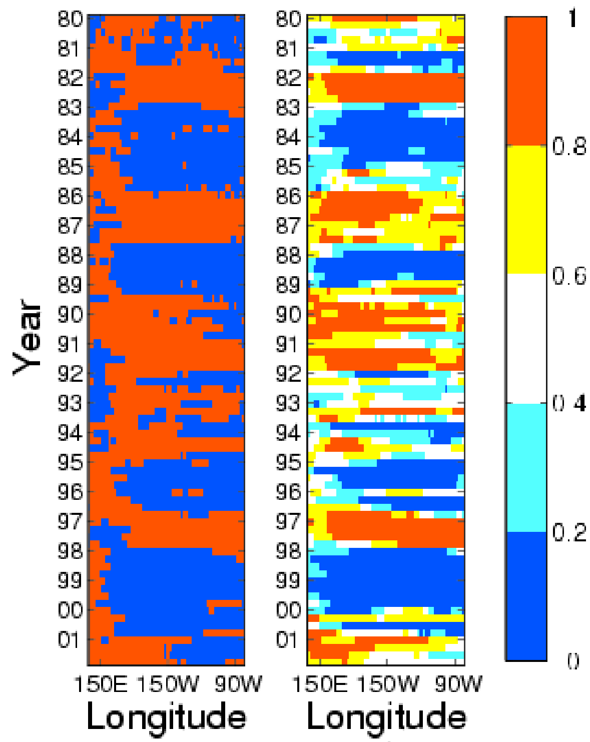
SST

$o = (SST > 0)$ $\quad f = \Pr(\hat{o})$

OBS



OBS ENS



The probability forecasts were constructed by fitting Normal distributions to the ensemble mean forecasts from the 7 DEMETER coupled models, and then calculating the area under the normal density for SST anomalies greater than zero.

SST anomalies (°C)

Forecast probabilities: f

Exercise 1:

Read data file equatorialpacificsst.txt which contains forecast probabilities for the event Eq. Pac. SST>0 and the corresponding binary observations

```
data<-read.table("equatorialpacificsst.txt")
```

```
#1st column contains forecast probabilities
probfcsts<-data[,1]
```

```
#2nd column contains binary observation
binobs<-data[,2]
```

```
#Compute the climatological frequency of the event
 obar<-mean(binobs)

#Compute the Brier score for the climatological frequency
#(i.e. the climatological forecast)
bsclim<-mean((obar-binobs)^2)

#Compute the variance of binary observation
var(binobs) *(length(binobs)-1)/length(binobs)

#Compute the uncertainty component of the Brier score
obar*(1-obar)
#How does this compare with the Brier score computed
#above? What can you conclude about the reliabilty and
#resolution components of the Brier score for the
#climatological forecast?
```

```
#Compute the Brier score for the SST prob. forecasts
#for the event SST>0
bs<-mean((probfcsts-binobs)^2)

#How does this compare with the Brier score for the
#climatological forecast? What can you conclude about the
#skill of these forecasts (i.e. which of the two are more
#skillfull by looking at their Brier score values)?

#Compute the Brier skill score
bss <- 1-(bs/bsclim)

#How do you interpret the Brier skill score obtained
#above? I.e. what can you conclude about the skill of the SST
#prob. forecasts when compared to the climatological
#forecast?
```

```
#Use the verification package to compute the Brier score and
#its decomposition for the SST prob. forecasts for
#the event SST>0
library(verification)
A<-verify(binobs,probfcsts, frcst.type="prob",obs.type="binary'
summary(A)

#Note: Brier score – Baseline is the Brier score for the
#reference climatological forecast
#Skill Score is the Brier skill score
#Reliability, resolution and uncertainty are the three
#components of the Brier score decomposition

#What can be conclude about the quality of these forecasts
#when compared with the climatological forecasts?
```

```
#Construct the reliability diagram for these forecasts using
#10 bins
nbins<-10
bk<-seq(0,1,1/nbins)
h<-hist(probfcsts,breaks=bk,plot=F)$counts
g<-hist(probfcsts[binobs==1],breaks=bk,plot=F)$counts
obari <- g/h
yi <- seq((1/nbins)/2,1,1/nbins)

par(pty='s',las=1)
reliability.plot(yi,obari,h,titl="10 bins",legend.names="")
abline(h=obar)

#What can you conclude about these forecasts by examining
#the feature of the reliability diagram curve?
```

```
# Compute reliability, resolution and uncertainty components
# of the Brier score
n<-length(probfcsts)
reliab <- sum(h*((yi-obari)^2), na.rm=TRUE)/n
resol <- sum(h*((obari-obar)^2), na.rm=TRUE)/n
uncert<-obar*(1-obar)
bs<-reliab-resol+uncert

#How does the results above compare with those obtained
#with the verify function?
```

Discrimination

- Conditioning of forecasts on observed outcomes
- Addresses the question: Does the forecast (probabilities) differ given different observed outcomes? Or, can the forecasts distinguish (discriminate or detect) an event from a non-event?

  Example: Event (Positive SST anom. observed)
  Non-event (Positive SST anom. not obs)
- If the forecast is the same regardless of the outcome, the forecasts cannot discriminate an *event* from a *non-event*
- Forecasts with no discrimination ability are useless because the forecasts are the same regardless of what happens

Important notes about events and non-events

- Example: event (precip. obs. in upper tercile)
  non-event (precip. not obs. in upper tercile)
- Events and non-events are complementary
- Events can happen (occur) or not happen (not occur)
- If fcst probability for an event to happen is 80% this indicates high confidence for the event to happen
- If fcst probability for an event to happen is 20% this indicates high confidence for the event not to happen
- Will see that in ROC curve (used to assess discrimination or distinction btw events and non-events):
  a) high confidence that an event will happen will appear in points located at the bottom left of ROC curve;
  b) high confidence that an event will not happen will appear in points located at the top right of ROC curve

Important notes about events and non-events

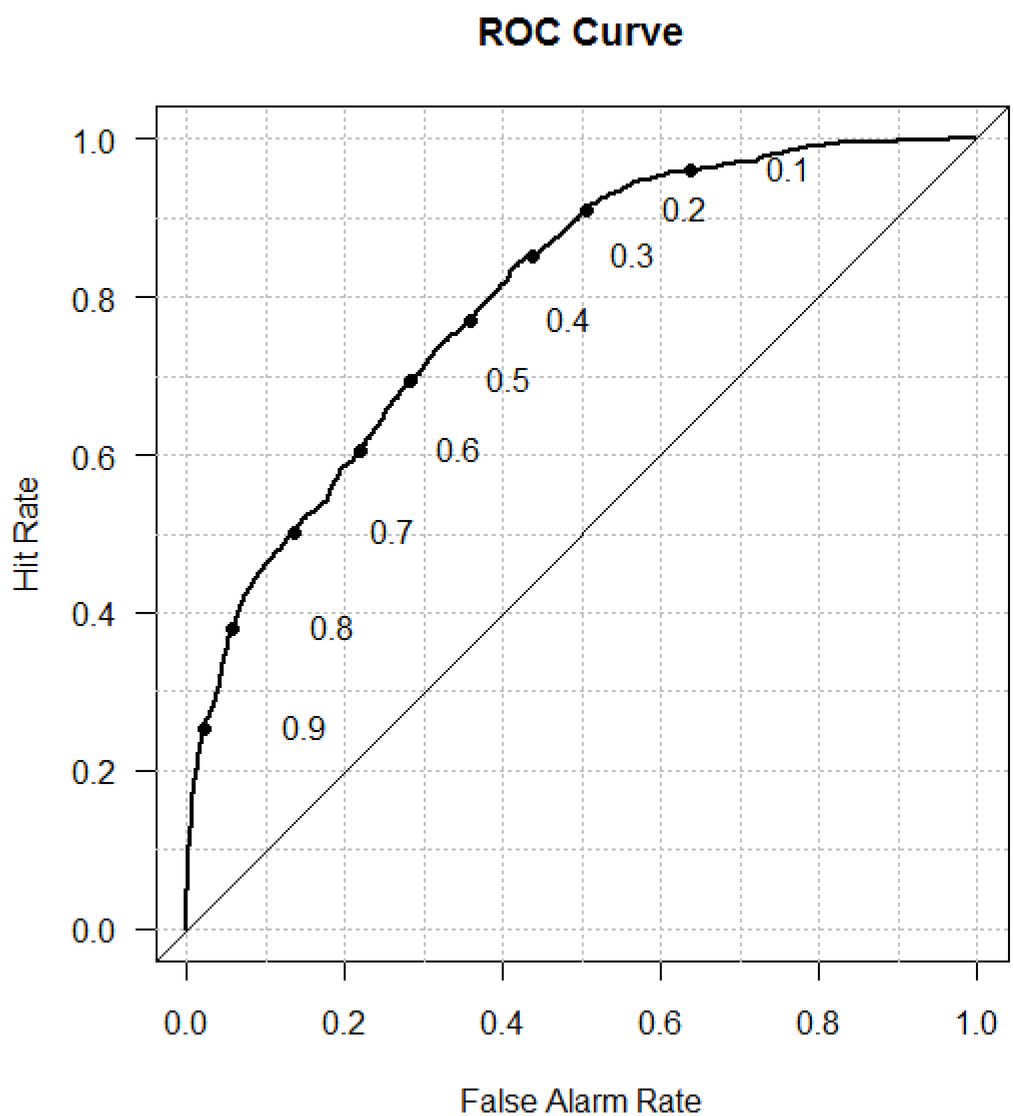
- As events and non-events are binary (i.e have 2 possible outcomes) the probability of correctly discriminating (distinguishing) and event from a non-event is 50%
- Example:
  - Lets say we have two years: 1990 and 1999
  - We know in one year (1990) precip in upper tercile was observed
  - We also know that in the other year (1999) precip in upper tercile was not observed
  - if in 1990 the fcst prob for precip in upper tercile was p=80% and in 1999 the fcst prob for precip in upper tercile was p=10% then we successfully discriminated btw the event and the non-event
- The ROC area will tell us the probability of successfully discriminating an event from a non event. (How different fcst probilities are for events and non events)

ROC: Relative operating characteristics

Measures discrimination (ability of forecasting system to detect the event of interest)

Forecast	Observed		
	Yes	No	Total
Yes	a (Hit)	b (False alarm)	a+b
No	c (Miss)	d (Correct rejection)	c+d
Total	a+c	b+d	a+b+c+d=n

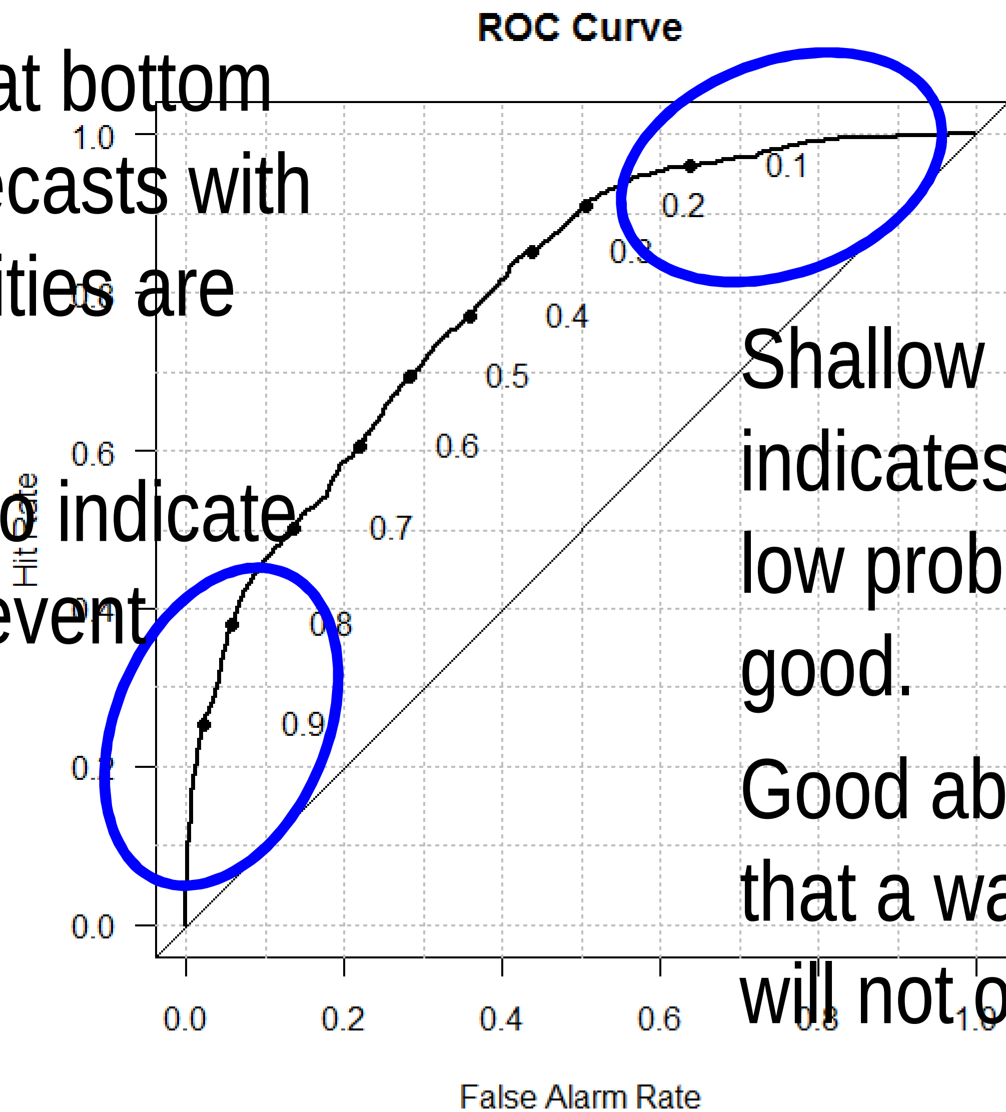
Hit rate=a/(a+c)
False alarm rate=b/(b+d)
ROC curve: plot of hit versus false-alarm rates for decreasing prob. thresholds



- The ROC curve is constructed by calculating the hit and false-alarm rates for decreasing probability thresholds
- Area under ROC curve (A) is a measure of discrimination: A = 0.79 (prob. of successfully discriminating a warm (SST>0) from a cold (SST<0) event)

Steep curve at bottom indicates forecasts with high probabilities are good.

Good ability to indicate that a warm event will occur.



Shallow curve at top indicates forecasts with low probabilities are good.

Good ability to indicate that a warm event will not occur.

- The ROC curve is constructed by calculating the hit and false-alarm rates for decreasing probability thresholds
- Area under ROC curve (A) is a measure of discrimination: A = 0.79 (prob. of successfully discriminating a warm (SST>0) from a cold (SST<0) event)

Exercise 2:

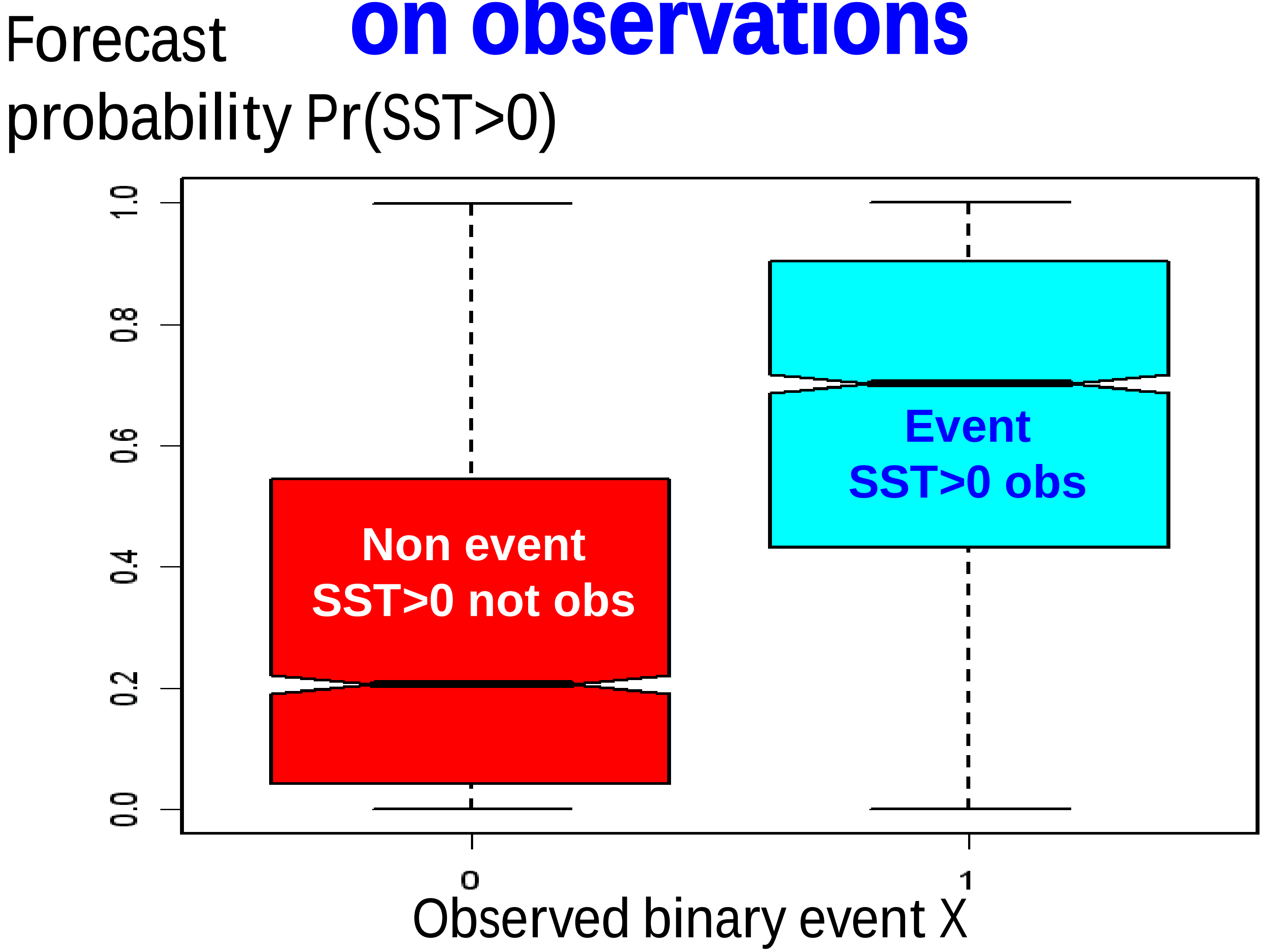
Read data file equatorialpacificsst.txt which contains forecast probabilities for the event Eq. Pac. SST>0 and the corresponding binary observations

data<-read.table("equatorialpacificsst.txt")

#1st column contains forecast probabilities

#2nd column contains binary observations

Prob. forecasts conditioned/stratified on observations



→ Forecasts do differ given different outcomes

→ Forecast system has discrimination (distinguish event from non-event)

Reproducing the previous plot

1) Stratify forecast probabilities p (1st column of data) on observed (1) and not observed (0) binary events (2nd column od data)

d1 #object containing strat of p on not observed event

```
> d1<-data[data[,2]==0,1]
```

d2 #object containing strat of p on observed event

```
> d2<-data[data[,2]==1,1]
```

2) Produce a boxplot using the command

```
> boxplot(d1,d2,col=c(2,5),notch=T,names=c(0,1))
```

```
# extract only forecast/obs pairs with p >=0.9
p<-0.9
# forecast events
f<-data[data[,1]>=p,]
a<-sum(f[,2]==1) #forecast and observed (hit)
b<-sum(f[,2]==0) #forecast and not observed (false alarm)
# not forecast events
g<-data[data[,1]<p,]
c<-sum(g[,2]==1) #not forecast and observed (miss)
d<-sum(g[,2]==0) #not fcst and not obs (correct rejection)
n<-a+b+c+d
hr<-a/(a+c)
far<-b/(b+d)
```

```
#Plot first point of the ROC curve
par(pty='s',las=1)
plot(far,hr,type="p",pch=16,xlim=c(0,1),ylim=c(0,1),xlab="False alarm rate",ylab="Hit rate")
abline(0,1)
```

```
#repeat the same procedure for p>=0.8

#extract only forecast/obs pairs with p >=0.8
p<-0.8
# forecast events
f<-data[data[,1]>=p,]
a<-sum(f[,2]==1) #forecast and observed (hit)
b<-sum(f[,2]==0) #forecast and not observed (false alarm)
# not forecast events
g<-data[data[,1]<p,]
c<-sum(g[,2]==1) #not forecast and observed (miss)
d<-sum(g[,2]==0) #not fcst and not obs (correct rejection)
n<-a+b+c+d
hr<-a/(a+c)
far<-b/(b+d)
```

```
#Plot new point in the ROC curve
points(far,hr,pch=16)

#repeat the same procedure for p>=0.7, p>=0.6, p>=0.5,
#p>=0.4, p>=0.3, p>=0.2 and p>=0.1 adding the new points
#in the ROC curve. Try later to do this using a for loop.

#The area below the curve that joins all points (the ROC
#area) is a forecast skill measure of discrimination.
#ROC area values equal 0.5 indicate no skill.
#ROC area values equal to 1 indicate perfect discrimination.
#ROC area values equal to 0 indicate perfectly bad
#discrimination.
```

```
#Constructing the empirical ROC curve

#find unique forecast probability values
p<-unique(data[,1])
#sort unique fcst prob values from largest to smallest
p<-rev(sort(p))
#define vectors to store hit and false-alarm rates
hr<-rep(NA,length(p)+2)
far<-rep(NA,length(p)+2)
#set first and last point in the ROC curve to (0,0) and (1,1)
hr[1]<-0
far[1]<-0
hr[length(p)+2]<-1
far[length(p)+2]<-1
```

```
#compute hit and false alarm rates for all fcst prob thresholds
for (i in 1:length(p)){
f<-data[data[,1]>=p[i],] #forecast events
a<-sum(f[,2]==1) #hit
b<-sum(f[,2]==0) #false alarm
g<-data[data[,1]<p[i],] # not forecast events
c<-sum(g[,2]==1) #miss
d<-sum(g[,2]==0) #correct rejection
hr[i+1]<-a/(a+c)
far[i+1]<-b/(b+d)
}
#plot empirical ROC curve
par(pty='s',las=1)
plot(far,hr,type="l",xlim=c(0,1),ylim=c(0,1),xlab="False alarm
rate",ylab="Hit rate")
abline(0,1)
```

```
#plot roc curve with verification package for comparison
x11()
roc.plot(data[,2],data[,1])

#compute area under empirical ROC curve
dif<-diff(far)
area<-sum(0.5*(hr[1:((length(hr)-1))]+hr[2:length(hr)])*dif)

#compute ROC area using the verification package
roc.area(data[,2],data[,1])

#The ROC skill score is defined as (2*ROC area)-1
#so that positive values indicate good discrimination skill
#and negative values indicate bad discrimination skill
rss<-2*area-1
```