Climatrans results: Work package 3: Outline of the development trends and climate change impact up to 2050

title: “Extreme precip and heat waves: empirical-statistical downscaling and analysis” author: “Rasmus Benestad” date: “March 30, 2016” output: pdf_document fig_width: 8 fig_height: 8 —

Background The main objectives of the project are: 1) Assess climate change and environmental impacts in urban areas in India related to the transport sector. 2) Develop mitigation and adaption strategies related to the transport sector in urban areas in India. The objectives are set responding to the following objectives in the KLIMAFORK call: Improve knowledge about the impacts of climate change on the natural environment and society; and Enhance knowledge about how society can and should adapt to the challenges of climate change.

The CLIMATRANS project proposal responds to the research themes-Strategies for reduced human made climate change" and “Strategies for climate change adaption” of the KLMAFORSK call. Assessing risk, uncertainty and irreversibility as inherent to the climate problem, this project will provide knowledge that makes it possible to improve institutional capacity in climate decision-making strategies. The project is interdisciplinary within the realm of social sciences (economics, political science, sociology), and natural sciences (meteorology and civil engineering). It is also a collaboration between Norwegian (Institute of Transport Economics and Norway Meteorological Institute) and four Indian research environments, thereby responding to the INDNOR program.

India’s three largest cities are selected as case cities: New Delhi, Mumbai, and Bangalore.

WP3: Outline of the development trends and climate change impact up to 2050 WP leader: Dr. Fagerli, MET. Main contributors: TERI and TOI. Duration: Months 11-18. In WP3 we will attempt to outline the trends (of aspects named in WP2), and project the climate change impact in the case cities in 2050. Evaluation of the climate change impact by 2050 (base scenario) will assess the likely impact in the case cities given a base scenario (-business as usual‖) with no new policy interventions implemented. Task 1 through 6 will address the same respective issues as in WP2, but with focus on the situation in India in 2050. Partners responsible for the respective tasks in WP3 are the same as the corresponding tasks in WP2. In addition, the following impact assessments will be done: Task 7: Climate change impact on the transport infrastructure. (TERI, MET, TOI) Task 8: Social impact of climate change on various population groups, health and wellbeing effects, etc. In terms of health effects, the downscaled future climate scenario and the derived transport scenarios for the cities, combined with appropriate emission scenarios for India/Asia, will be used as basis for deriving health effects of ozone and PM (including climate change). The downscaled future climate scenario and the present day emission scenarios will be used to derive increases in health effects only due to climate change. (MET, TERI, TOI) 7 Task 9: Economic impact of climate change in the case cities, and for the population (e.g., household income, wealth distribution, poverty levels, etc.)

Analysis The downscaling of the wet-day mean precipitation used the surface temperature as predictor.

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com. To execute this type:

```
library(rmarkdown); render('climatrans.Rmd',pdf_document())
```

Load the esd and ncdf packages.

```
library(esd)
```

```
## Loading required package: ncdf4
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric


##
## Attaching package: 'esd'

## The following object is masked from 'package:base':
##
##     subset.matrix
```

```
library(ncdf)
## Information about the system and session
sessionInfo()
```

```
## R version 3.1.3 (2015-03-09)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu precise (12.04.5 LTS)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=nb_NO.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=nb_NO.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ncdf_1.8.6      esd_1.1         zoo_1.7-12      ncdf4_1.15
## [5] rmarkdown_0.9.5
##
## loaded via a namespace (and not attached):
##  [1] digest_0.6.9    evaluate_0.8.3  formatR_1.3     grid_3.1.3
##  [5] htmltools_0.3   knitr_1.12.3    lattice_0.20-33 magrittr_1.5
##  [9] stringi_1.0-1   stringr_1.0.0   tools_3.1.3
```

Functions

Define a function that converts the surface temperature into saturation vapour pressure:

```
tas2es <- function(x,mask=TRUE,land=TRUE,season=5:10,FUN='min') {
  if (mask) x <- mask(x,land=land)
  es <- subset(C.C.eq(x),it=month.abb[season])
```

```
  es <- aggregate(es,year,FUN=FUN)
  index(es) <- year(es)
  invisible(es)
}
```

Function to estimate probabilty for daily amount of exceeding a threshold x0 given that the distribution follows an exponential distribution and the records of the wet-day mean precipitation and frequency.

```
PrexpPr <- function(mu,fw,x0=10) {
  Pr <- zoo(coredata(fw)*exp(-x0/coredata(mu)),order.by=year(fw))

  Pr.mu <- zoo(mean(coredata(fw))*exp(-x0/coredata(mu)),order.by=year(fw))
  Pr.fw <- zoo(coredata(fw)*exp(-x0/mean(coredata(mu))),order.by=year(fw))
  attr(Pr,'prob.f(mu)') <- Pr.mu
  attr(Pr,'prob.f(fw)') <- Pr.fw
  Pr <- attrcp(mu,Pr)
  attr(Pr,'variable') <- 'Pr'
  attr(Pr,'unit') <- 'probability'
  attr(Pr,'longname') <- paste('Probability of exceeding',x0)
  class(Pr) <- class(mu)
  invisible(Pr)
}
```

Precipitation

Pre-processing: adapting the netCDF data to esd: set up station objects:

```
## Precipitation
## time origin: minutes since 1901-01-01 00:00
library(ncdf)
ncid <- open.ncdf('~/Dropbox/Public/ClimaTrans/delhi.nc')
delhi <- get.var.ncdf(ncid,varid='p')
time <- get.var.ncdf(ncid,varid='time')
close.ncdf(ncid)
delhi <- zoo(x=delhi,order.by=as.Date(time/(24*60),origin='1901-01-01'))
delhi <- as.station(delhi,loc='Delhi',param='precip',unit='mm/day',lon=77.2,lat=28.6,
                    cntr='India',longname='precipitation',
                    info='extracted rainfall data from IMD gridded daily data',
                    ref='Madhusoodanan M.S. ("Madhusoodanan M.S." <madhusoodanan@gmail.com>)')

ncid <- open.ncdf('~/Dropbox/Public/ClimaTrans/bombay.nc')
bombay <- get.var.ncdf(ncid,varid='p')
time <- get.var.ncdf(ncid,varid='time')
close.ncdf(ncid)
bombay <- zoo(x=bombay,order.by=as.Date(time/(24*60),origin='1901-01-01'))
bombay <- as.station(bombay,loc='Bombay',param='precip',unit='mm/day',lon=72.9,lat=19.0,
                cntr='India',longname='precipitation',
                info='extracted rainfall data from IMD gridded daily data',
                ref='Madhusoodanan M.S. ("Madhusoodanan M.S." <madhusoodanan@gmail.com>)')

ncid <- open.ncdf('~/Dropbox/Public/ClimaTrans/bangalore.nc')
bangalore <- get.var.ncdf(ncid,varid='p')
```

```
time <- get.var.ncdf(ncid,varid='time')
close.ncdf(ncid)

bangalore <- zoo(x=bangalore,order.by=as.Date(time/(24*60),origin='1901-01-01'))
bangalore <- as.station(bangalore,loc='Bangalore',param='precip',unit='mm/day',
                lon=77.6,lat=13.0,cntr='India',longname='precipitation',
                info='extracted rainfall data from IMD gridded daily data',
                ref='Madhusoodanan M.S. ("Madhusoodanan M.S." <madhusoodanan@gmail.com>)')

## Combine the individual stations into a group of stations
climatrans.pr <- combine(delhi,bombay,bangalore)
## Save the results for future use
save(file='climatrans.pr.rda',climatrans.pr)

## Maximum temperature
## time origin: minutes since 1969-01-01 00:00
ncid <- open.ncdf('~/Dropbox/Public/ClimaTrans/delhi_tmax.nc')
delhi <- get.var.ncdf(ncid,varid='temp')
time <- get.var.ncdf(ncid,varid='time')
close.ncdf(ncid)
delhi <- zoo(x=delhi,order.by=as.Date(time/(24*60),origin='1969-01-01'))
tx.delhi <- as.station(delhi,loc='Delhi',param='tmax',unit='degC',lon=77.2,lat=28.6,
               cntr='India',longname='daily maximum temperature',
               info='extracted from IMD gridded daily data',
               ref='Madhusoodanan M.S. ("Madhusoodanan M.S." <madhusoodanan@gmail.com>)')

ncid <- open.ncdf('~/Dropbox/Public/ClimaTrans/bombay_tmax.nc')
bombay <- get.var.ncdf(ncid,varid='temp')
time <- get.var.ncdf(ncid,varid='time')
close.ncdf(ncid)
bombay <- zoo(x=bombay,order.by=as.Date(time/(24*60),origin='1969-01-01'))
tx.bombay <- as.station(bombay,loc='Bombay',param='tmax',unit='degC',lon=72.9,lat=19.0,
                cntr='India',longname='daily maximum temperature',
                info='extracted from IMD gridded daily data',
                ref='Madhusoodanan M.S. ("Madhusoodanan M.S." <madhusoodanan@gmail.com>)')

ncid <- open.ncdf('~/Dropbox/Public/ClimaTrans/bangalore_tmax.nc')
bangalore <- get.var.ncdf(ncid,varid='temp')
time <- get.var.ncdf(ncid,varid='time')
close.ncdf(ncid)
bangalore <- zoo(x=bangalore,order.by=as.Date(time/(24*60),origin='1969-01-01'))
tx.bangalore <- as.station(bangalore,loc='Bangalore',param='tmax',unit='degC',
                lon=77.6,lat=13.0,cntr='India',longname='daily maximum temperature',
                info='extracted from IMD gridded daily data',
                ref='Madhusoodanan M.S. ("Madhusoodanan M.S." <madhusoodanan@gmail.com>)')

## Combine the individual stations into a group of stations
climatrans.tx <- combine(tx.delhi,tx.bombay,tx.bangalore)
## Save the results for future use
save(file='climatrans.tx.rda',climatrans.tx)
```

Analysis

The analysis and downscaling can be carried out once the data has been converted to esd-station object. First load the precipitation data:

Daily station data

```
## Load precipitation data extracted from the gridded IMD data set:
load('climatrans.pr.rda')
## Display the structure of the data as
str(climatrans.pr)
```

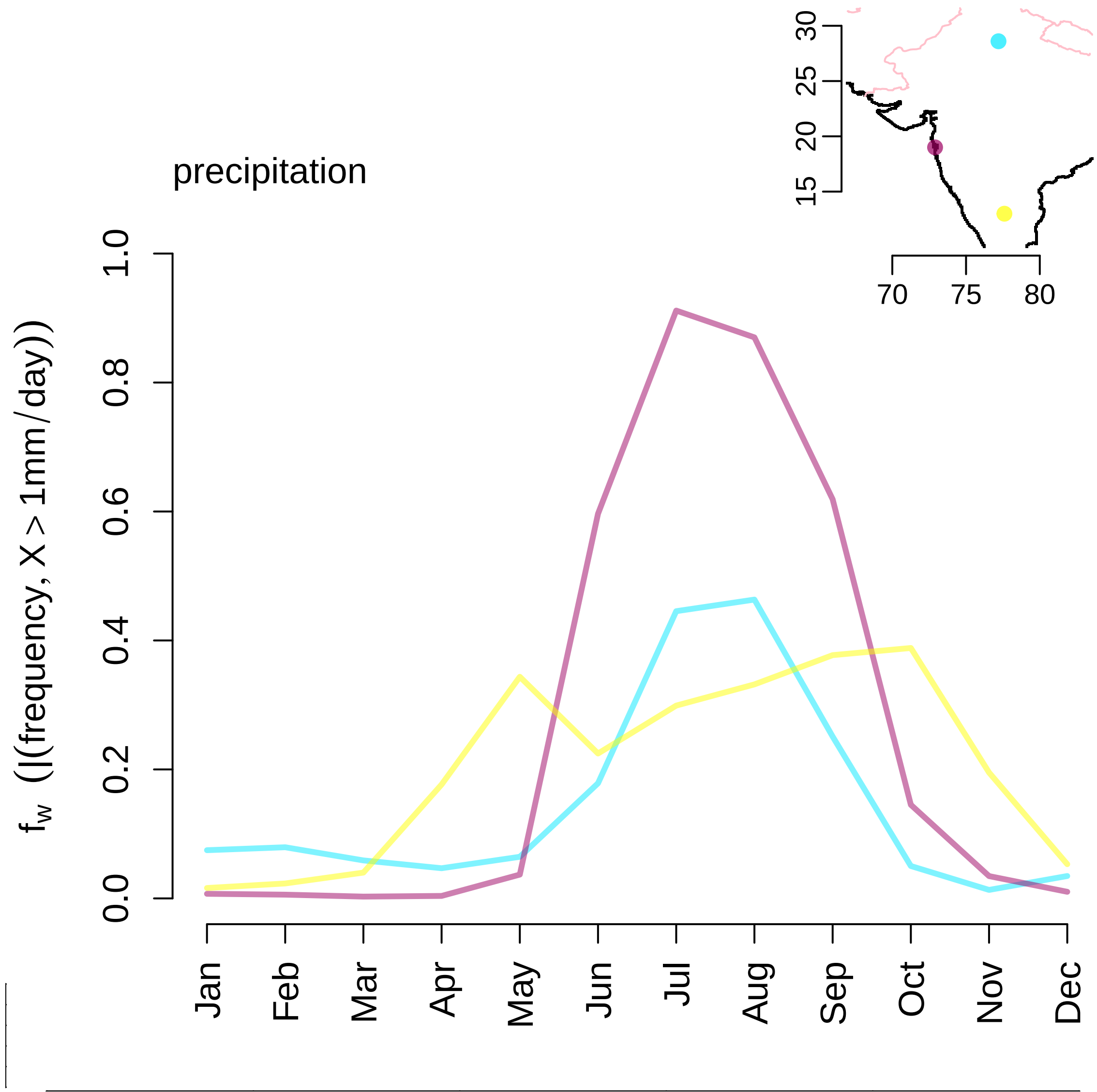
```
## 'zoo' series from 1901-01-01 to 2004-12-31
##   Data: num [1:37986, 1:3] 11.8 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "Delhi" "Bombay" "Bangalore"
##  - attr(*, "location")= chr [1:3] "Delhi" "Bombay" "Bangalore"
##  - attr(*, "country")= chr [1:3] "India" "India" "India"
##  - attr(*, "station_id")= logi [1:3] NA NA NA
##  - attr(*, "longitude")= num [1:3] 77.2 72.9 77.6
##  - attr(*, "latitude")= num [1:3] 28.6 19 13
##  - attr(*, "altitude")= logi [1:3] NA NA NA
##  - attr(*, "variable")= chr [1:3] "precip" "precip" "precip"
##  - attr(*, "longname")= chr [1:3] "precipitation" "precipitation" "precipitation"
##  - attr(*, "unit")= chr [1:3] "mm/day" "mm/day" "mm/day"
##  - attr(*, "aspect")= logi [1:3] NA NA NA
##  - attr(*, "source")= logi [1:3] NA NA NA
##  - attr(*, "quality")= logi [1:3] NA NA NA
##  - attr(*, "URL")= logi [1:3] NA NA NA
##  - attr(*, "history")=List of 3
##   ..$ call     :List of 1
##   .. ..$ : language combine.station(delhi, bombay, bangalore)
##   ..$ timestamp: chr "Mon Apr 11 15:47:39 2016"
##   ..$ session  :List of 3
##   .. ..$ R.version  : chr "R version 3.1.3 (2015-03-09)"
##   .. ..$ esd.version: chr "esd_1.2"
##   .. ..$ platform   : chr "x86_64-pc-linux-gnu (64-bit)"
##  - attr(*, "reference")= chr [1:3] "Madhusoodanan M.S. (\"Madhusoodanan M.S.\" <madhusoodanan@gmail.
##  - attr(*, "info")= chr [1:3] "extracted rainfall data from IMD gridded daily data" "extracted rainf
##   Index:  Date[1:37986], format: "1901-01-01" "1901-01-02" "1901-01-03" "1901-01-04" ...
```

Sensitivity tests

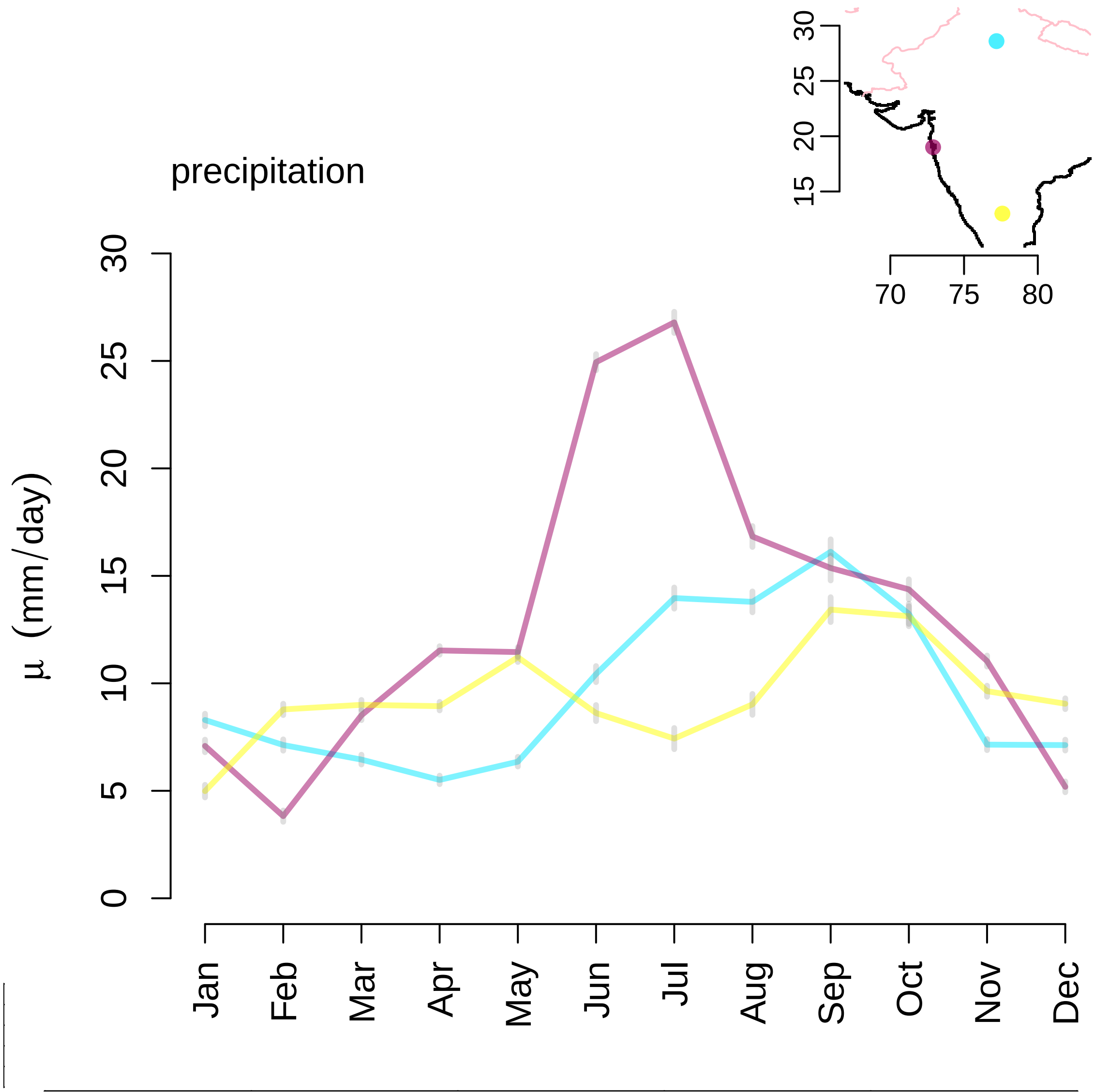
How do the precipitation respond to varying forcings/conditions? Such iformation may provide useful clues as to how the precipitation is expected to change in the future.

Examine the mean seasonal cycle to see if it responds to systematic forcings - the simples and most obvious being the seasonal variation in the solar inclination and the monsoon season:

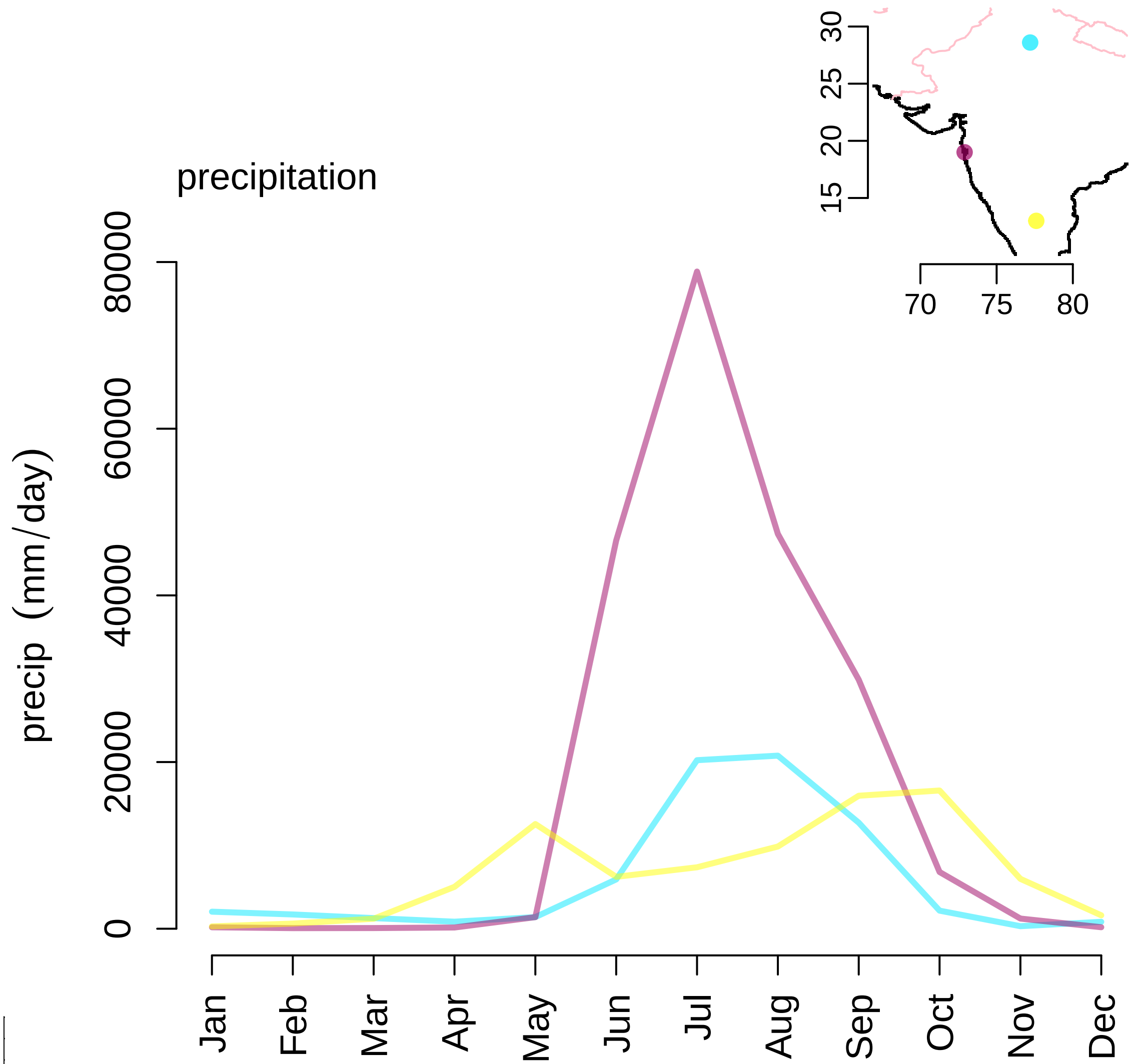
```r
## Wet-day frequency
plot(aggregate(climatrans.pr,month,'wetfreq'),new=FALSE)
```




```r
## Wet-day mean
plot(aggregate(climatrans.pr,month,'wetmean'),new=FALSE)
```




```
## Monthly precipitation totals
plot(aggregate(climatrans.pr,month,'sum'),new=FALSE)
```

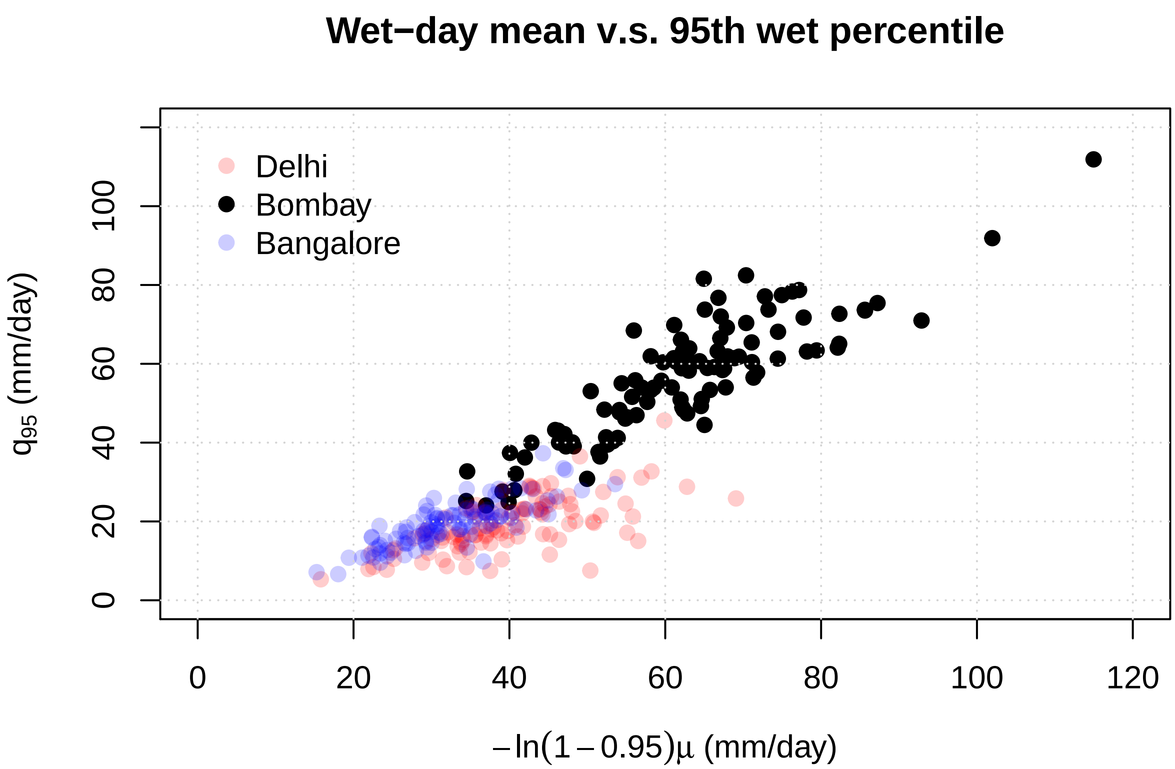


Estimate annual aggregated statistics for the months May-October (the wet/Monsoon season):

```
season<-5:10
fw <- aggregate(subset(climatrans.pr,it=month.abb[season]),year,'wetfreq')
mu <- aggregate(subset(climatrans.pr,it=month.abb[season]),year,'wetmean')
q95 <- aggregate(subset(climatrans.pr,it=month.abb[season]),year,'quantile',probs=0.95)
```

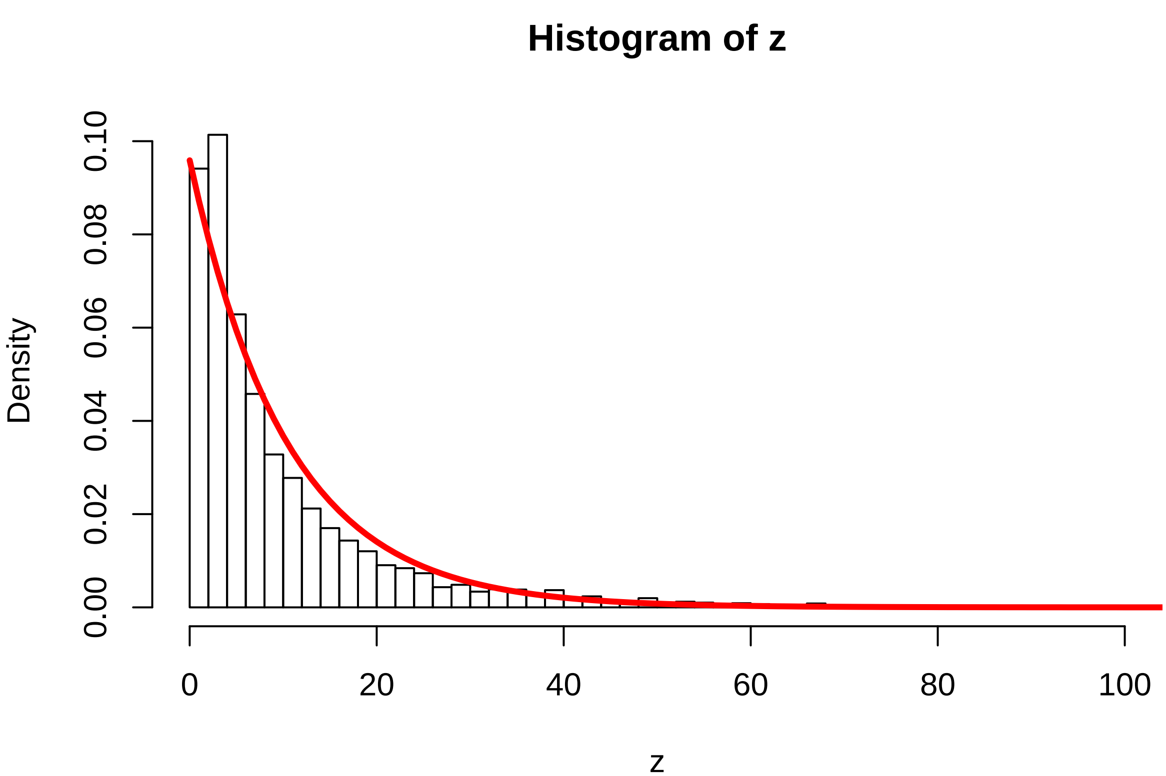
Compare the data to the exponential distribution - if the data is exponentially distributed, then the wet-day mean precipitation μ and the 95-percentile q_{95} are propportional by $q_p = -ln(1-p)\mu$.

```
## check the relationship between wet-day 95-percentile and wet-day mean
plot(-log(0.05)*zoo(mu[,2]),zoo(q95[,2]),pch=19,xlim=c(0,120),ylim=c(0,120),
     main='Wet-day mean v.s. 95th wet percentile',
     xlab=expression(paste(-ln(1-0.95)*mu,' (mm/day)')), ylab=expression(paste(q[95],' (mm/day)')))
points(-log(0.05)*zoo(mu[,1]),zoo(q95[,1]),pch=19,col=rgb(1,0,0,0.2))
points(-log(0.05)*zoo(mu[,3]),zoo(q95[,3]),pch=19,col=rgb(0,0,1,0.2))
grid()
legend(0,120,loc(climatrans.pr),pch=19,col=c(rgb(1,0,0,0.2),'black',rgb(0,0,1,0.2)),bty='n')
```



Compare the histogram of the preciptation data with an exponential distribution

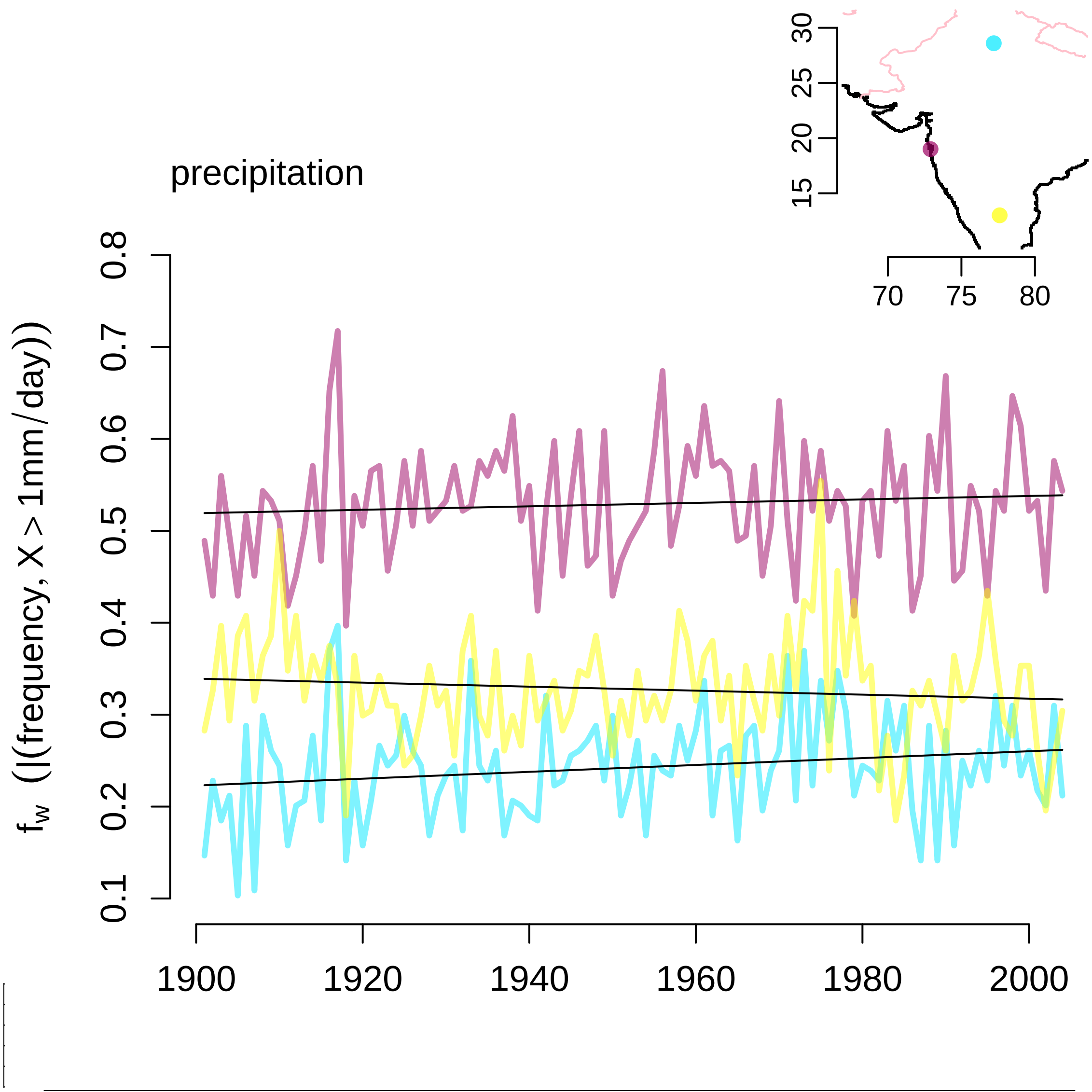
```
z <- climatrans.pr[,3]; z <- z[z > 1]
hist(z,breaks=seq(0,500,by=2),xlim=c(0,100),freq=FALSE)
lines(seq(0,250,by=1),dexp(seq(0,250,by=1),rate=1/mean(z)),lwd=3,col='red')
```



Historical long-term trends

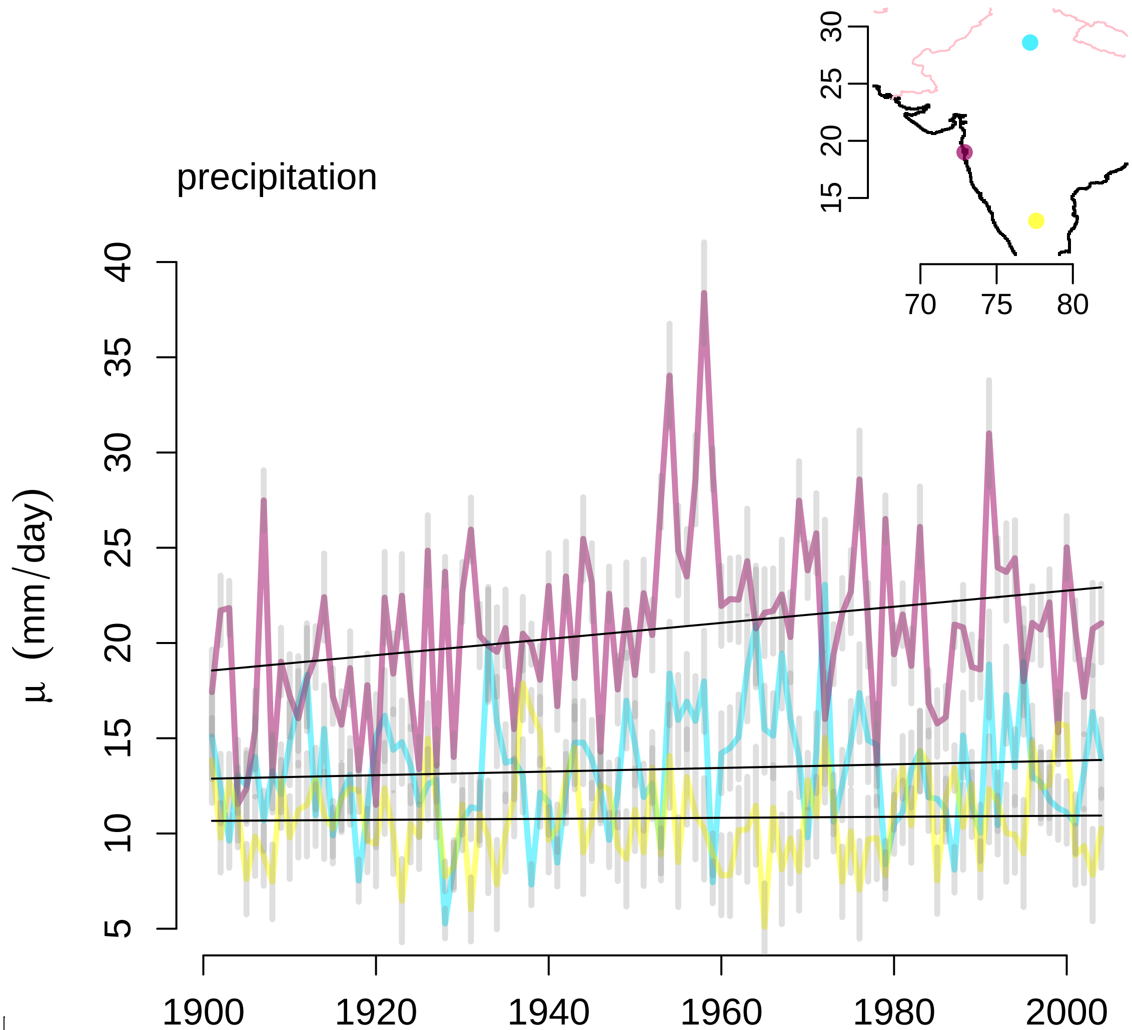
Estimate the long-term trends in the precipitation statistics at the three megacities: first the wet-day frequency f_w - does the number of rainy days change over time?

```
## Wet-day frequency
plot(fw,new=FALSE)
for(i in 1:3) lines(trend(subset(fw,is=i)))
```



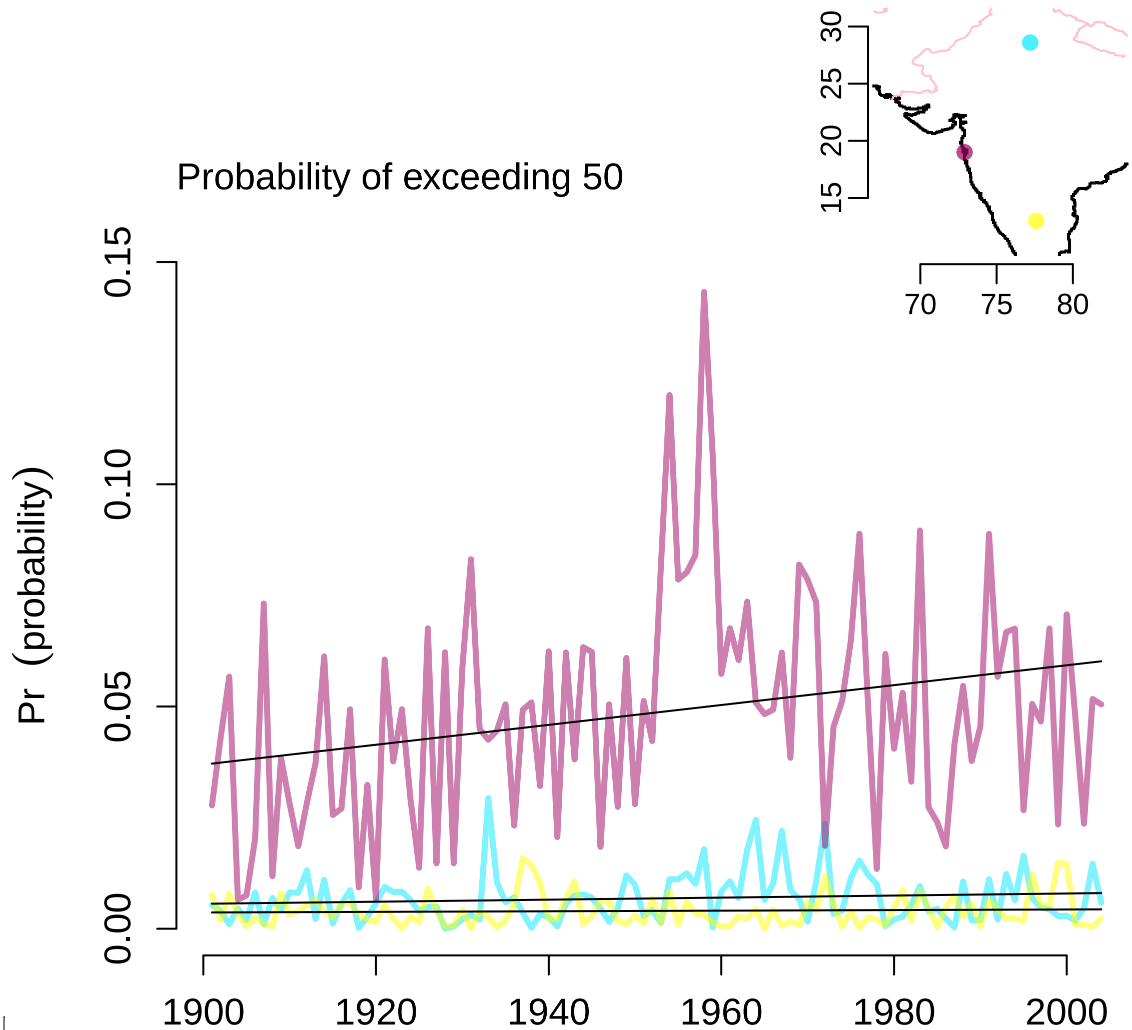
Does the wet-day mean precipitation μ change over time?

```
## Wet-day mean
plot(mu,new=FALSE)
for(i in 1:3) lines(trend(subset(mu,is=i)))
```



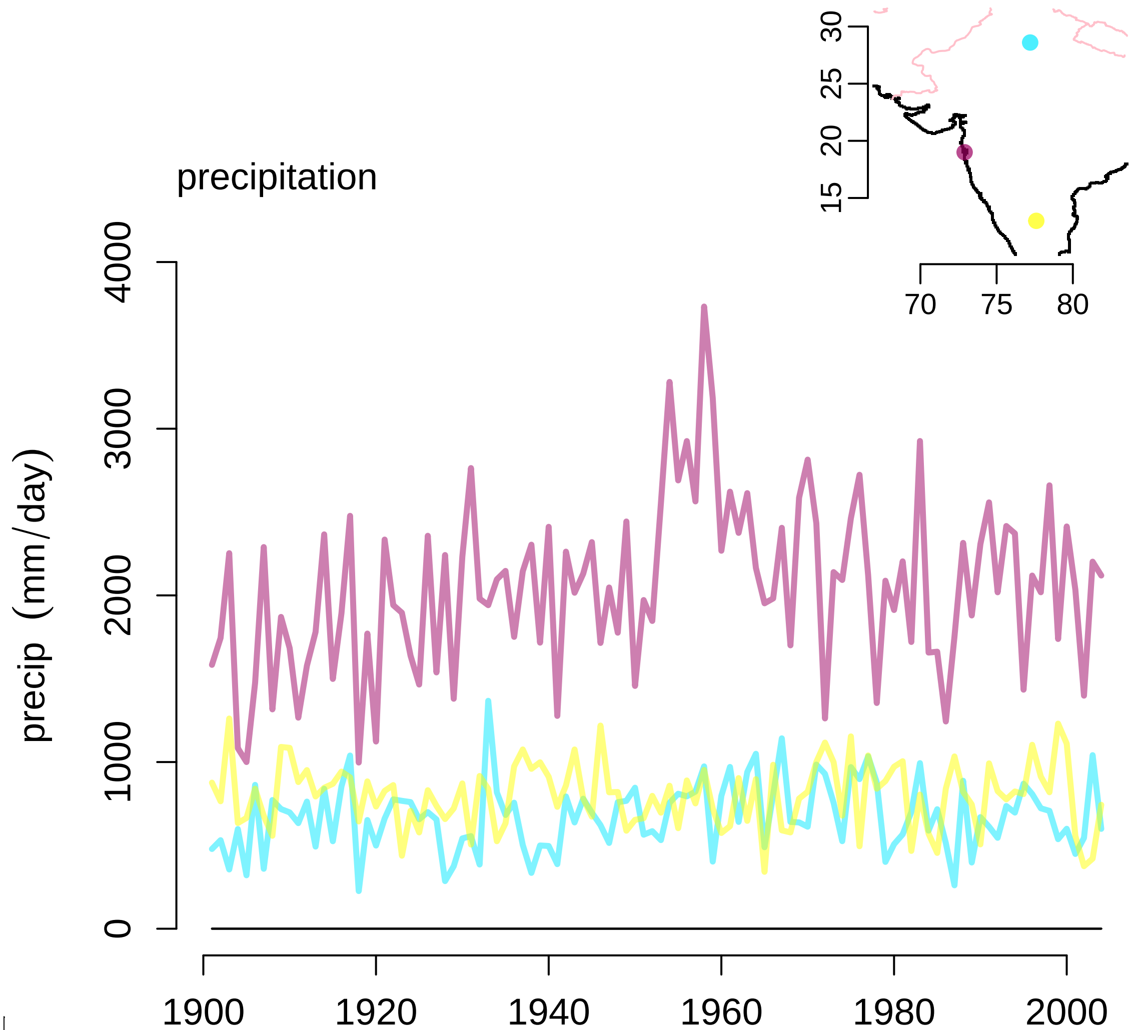
If the 24-hr preciptation amounts is exponential (when it rains), then the probability of exceeding a threshold is $Pr(X > x_0) = f_w e^{-x_0/\mu}$. Since we know both

```
## Wet-day mean
Pr <- PrexpPr(mu,fw,x0=50)
plot(Pr,new=FALSE,errorbar=FALSE)
for(i in 1:3) lines(trend(subset(Pr,is=i)))
```



Also, it's interesting to see how the annual total precipitation amount changes over time:

```
## Aggregate to mean seasonal cycle and annual mean values
plot(annual(climatrans.pr,'sum'),new=FALSE)
for(i in 1:3) lines(trend(subset(annual(climatrans.pr,'wetfreq'),is=i)))
```



Predictors

Now prepare for the downscaling: Retrieve and process the predictors. For the wet-day mean we try temperature (or sea surface temperature, SST). There are different possibilities. We can estimate the vapour saturation pressure based on the Clausius-Clapeyron equation (`C.C.eq()`) or we can use the temperature directly (the former translates more directly into moisture mass). The aggregation of the temperature over the wet-season may involve the average, minimum or the maximum values. Trial anderror to see what works and what doesn't:

```
## Aggregate sub-season values to annual mean values
if (!exists("es")) {
  t2m <- retrieve('air.mon.mean.nc',lon=c(60,100),lat=c(0,23))
  es <- aggregate(tas2es(t2m,season=season),year,'mean')
}
```

```
## [1] "Warning : Calendar attribute has not been found in the meta data and will be set automatically.
```

```
## Warning in if (deparse(substitute(by)) == "year") {: the condition has
## length > 1 and only the first element will be used

## Warning in if (deparse(substitute(by)) == "year") {: the condition has
## length > 1 and only the first element will be used
```

The mean seal-level pressure (SLP) is usually a promising predictor for the wet-day frequency.

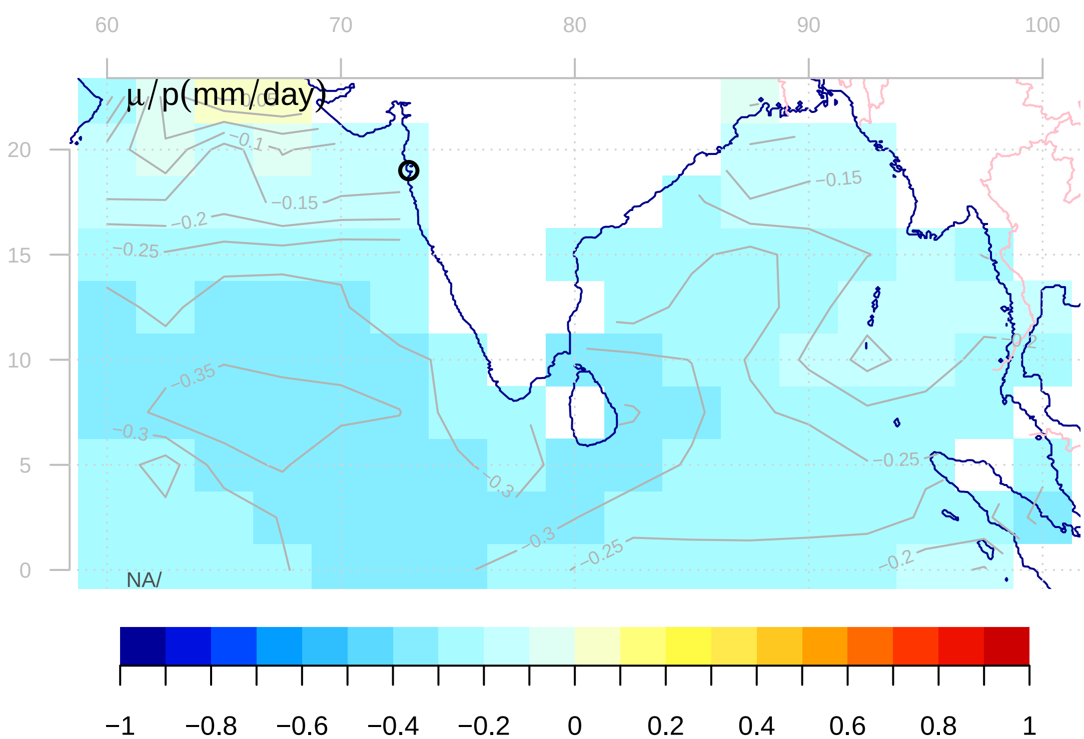
```
if (!exists("slp")) {
  slp <- retrieve('slp.mon.mean.nc',lon=c(60,100),lat=c(0,23))
  slp <- aggregate(subset(slp,it=month.abb[season]),year,'mean')
}
```

```
## [1] "Warning : Calendar attribute has not been found in the meta data and will be set automatically.

## Warning in if (deparse(substitute(by)) == "year") {: the condition has
## length > 1 and only the first element will be used
```

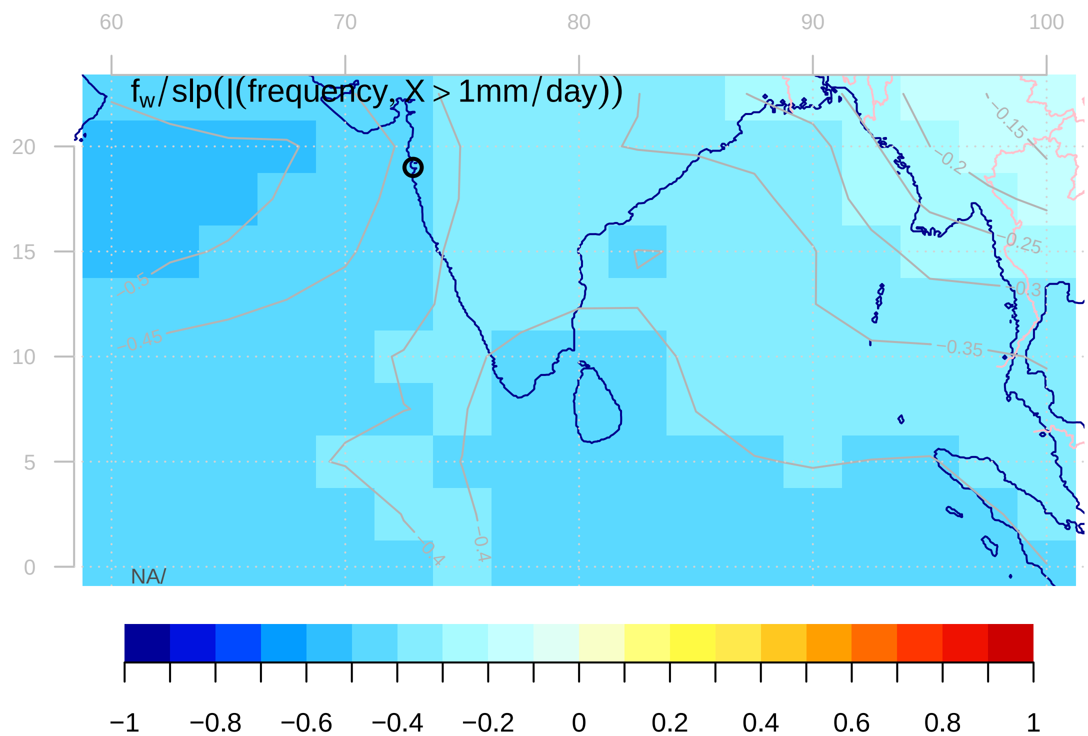
Examine potential large-scale tele-connections: correlation fields

```
## Correlation between the wet-day mean and the temperature-based predictor
y <- subset(mu,is=2)
corfield(y,es,colbar=list(pal='t2m',breaks=seq(-1,1,by=0.1)))
```

```
## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf
```



```
## Correlation between the wet-day frequency and the SLP
z <- subset(fw,is=2)
corfield(z,slp,colbar=list(pal='t2m',breaks=seq(-1,1,by=0.1)))
```



Principal component analysis (PCA) can improve the quality of empirical-statistical downscaling (ESD) by reorganising the data so that the large-scale variability is enhanced (http://dx.doi.org/10.3402/tellusa.v67.28326).

Estimate PCA for the predicands

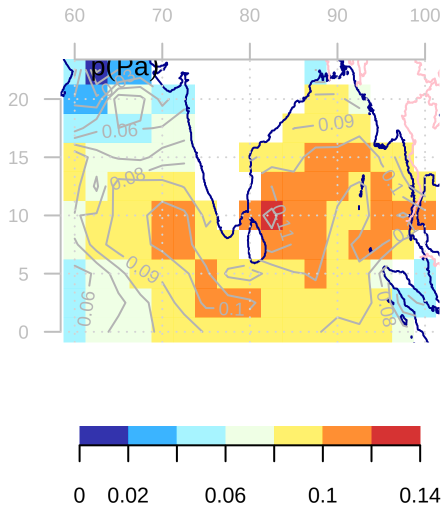
```
pca.mu <- PCA(mu)
pca.fw <-PCA(fw)
```

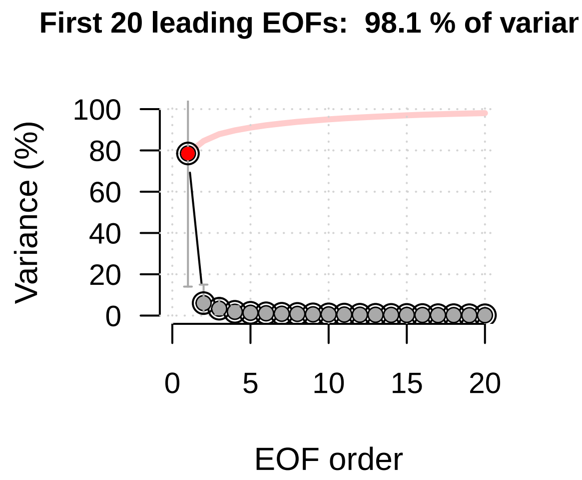
Estimate EOFs for the predictands

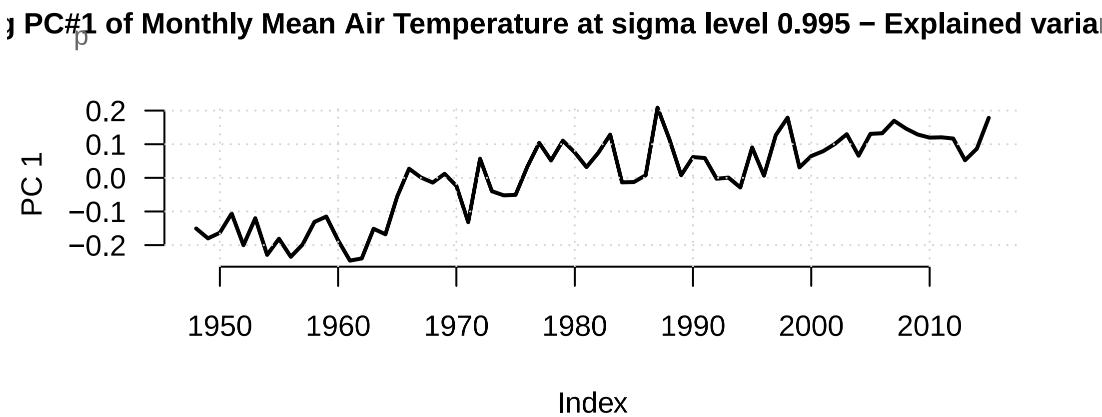
```
eof.es <- EOF(es)
eof.slp <- EOF(slp)
index(eof.es) <- year(eof.es)
index(eof.slp) <- year(eof.slp)
```

Plot the EOFs for the temperature-based predictor

```
plot(eof.es,new=FALSE)
```

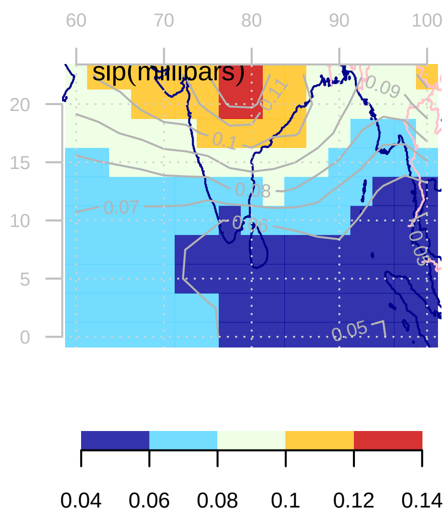


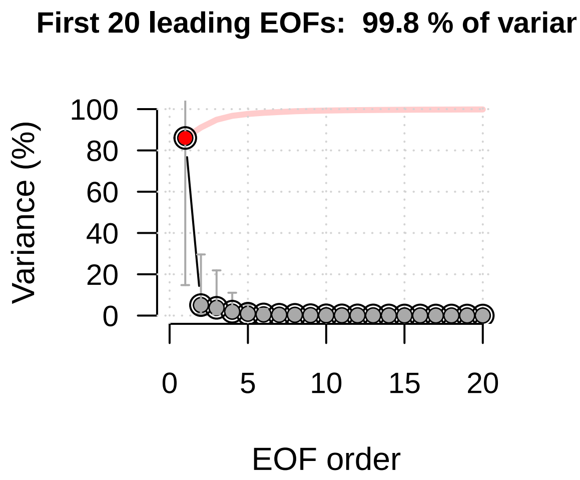


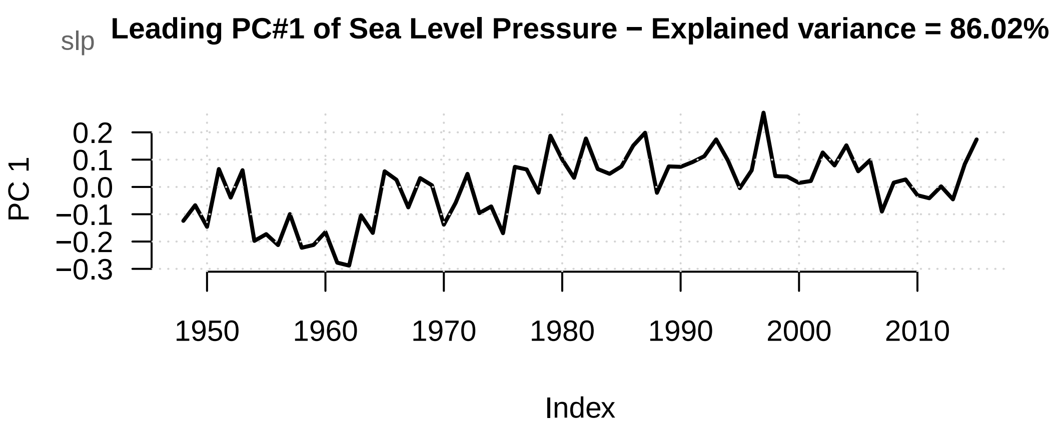


Plot the EOFs for the SLP

```
plot(eof.slp,new=FALSE)
```





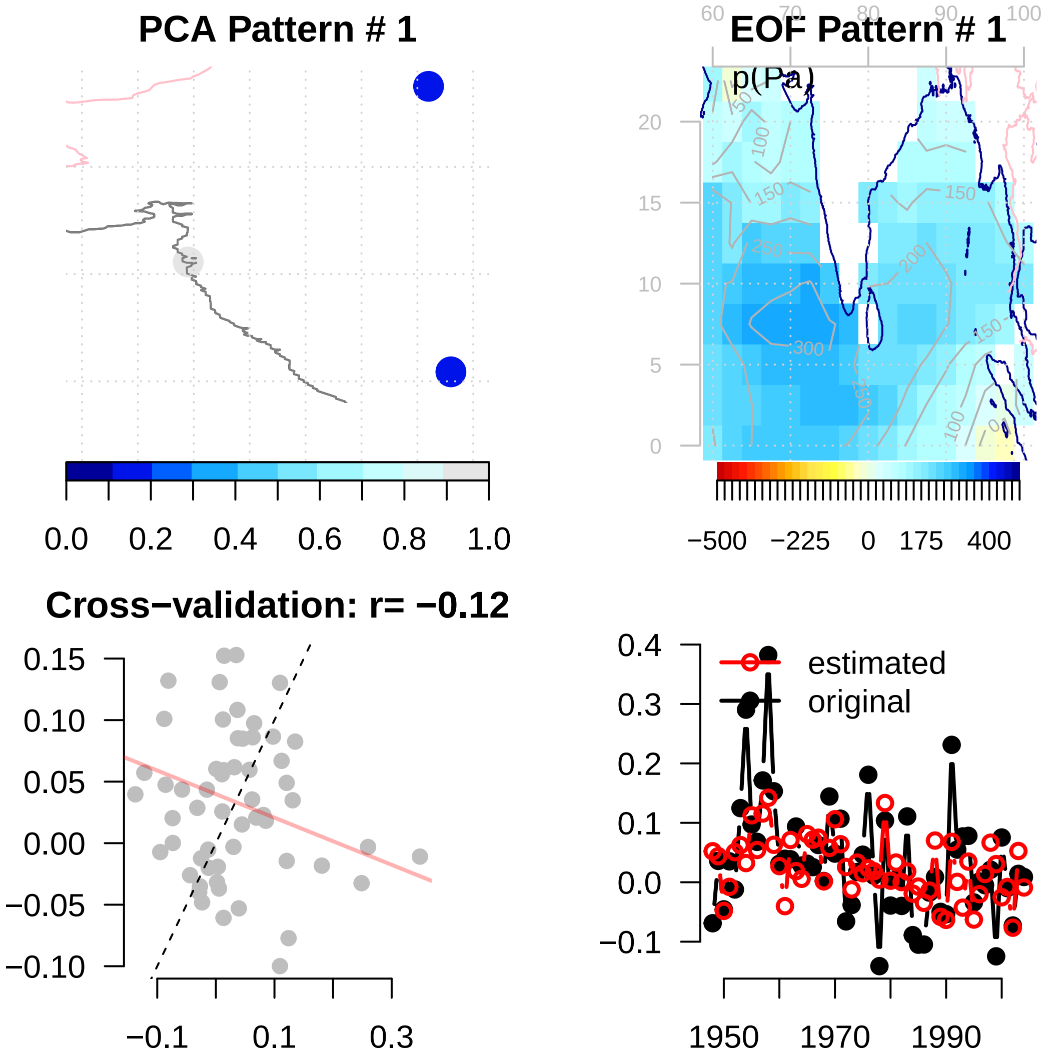


Do the downscaling

```
##
  |
  |                                                                      |   0%
  |
  |=======================                                               |  33%
  |
  |===============================================                       |  67%
  |
  |======================================================================| 100%

##
  |
  |                                                                      |   0%
  |
  |=======================                                               |  33%
  |
```

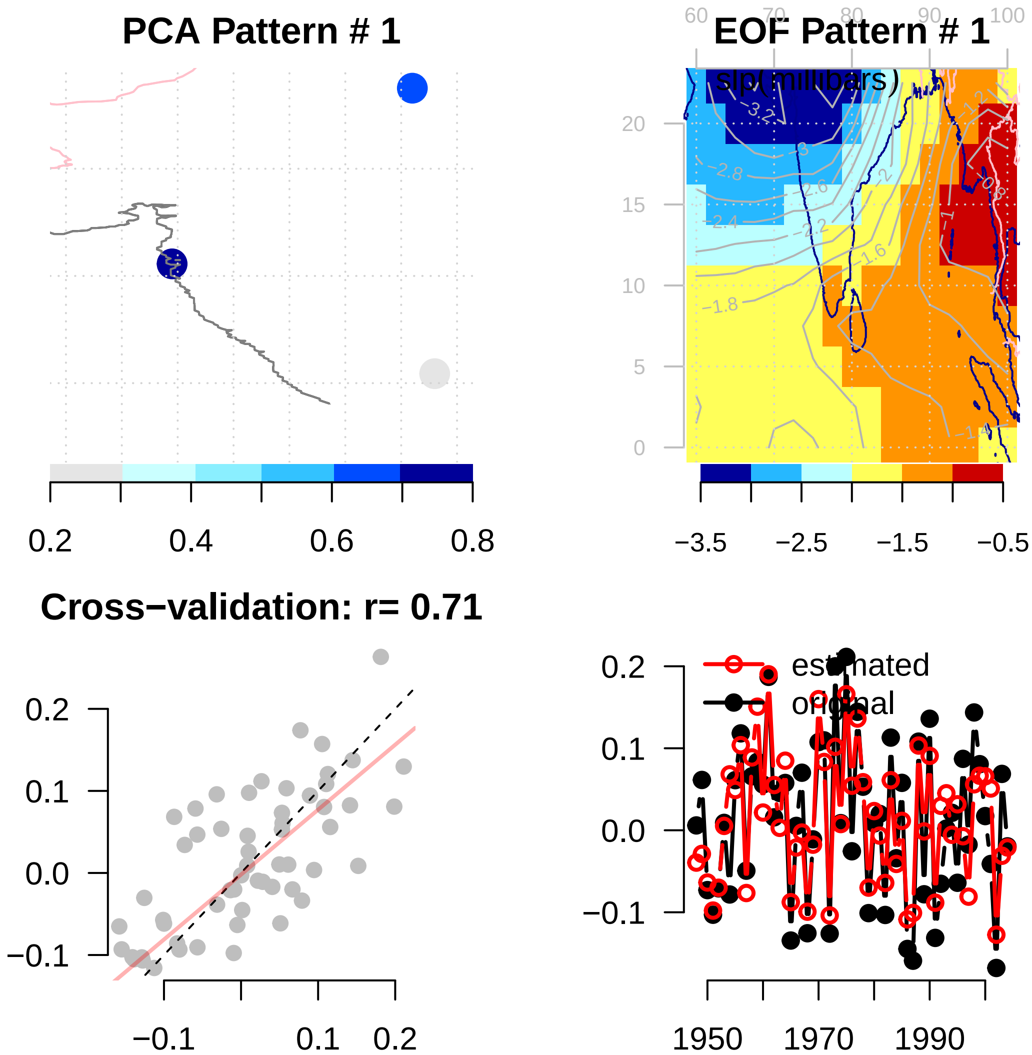
```
  |===========================================                      |  67%
  |
  |=================================================================| 100%
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "plot" is not a
## graphical parameter
```




```
## NULL
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "plot" is not a
## graphical parameter
```




```
## NULL
```

Examine the residuals from the downscaling

First check the residuals for the wet-day mean precipitation:

```
## Examine the PCA of residuals for the cold monthsto see if there are any structures left
  mu.2 <- as.residual(as.station(ds.mu))
  mu.2 <- attrcp(mu,mu.2); class(mu.2) <- class(mu)
  pca.mu.2 <- PCA(mu.2)
  plot(pca.mu.2)
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "plot" is not a
## graphical parameter
```

The residuals for the wet-day frequency:

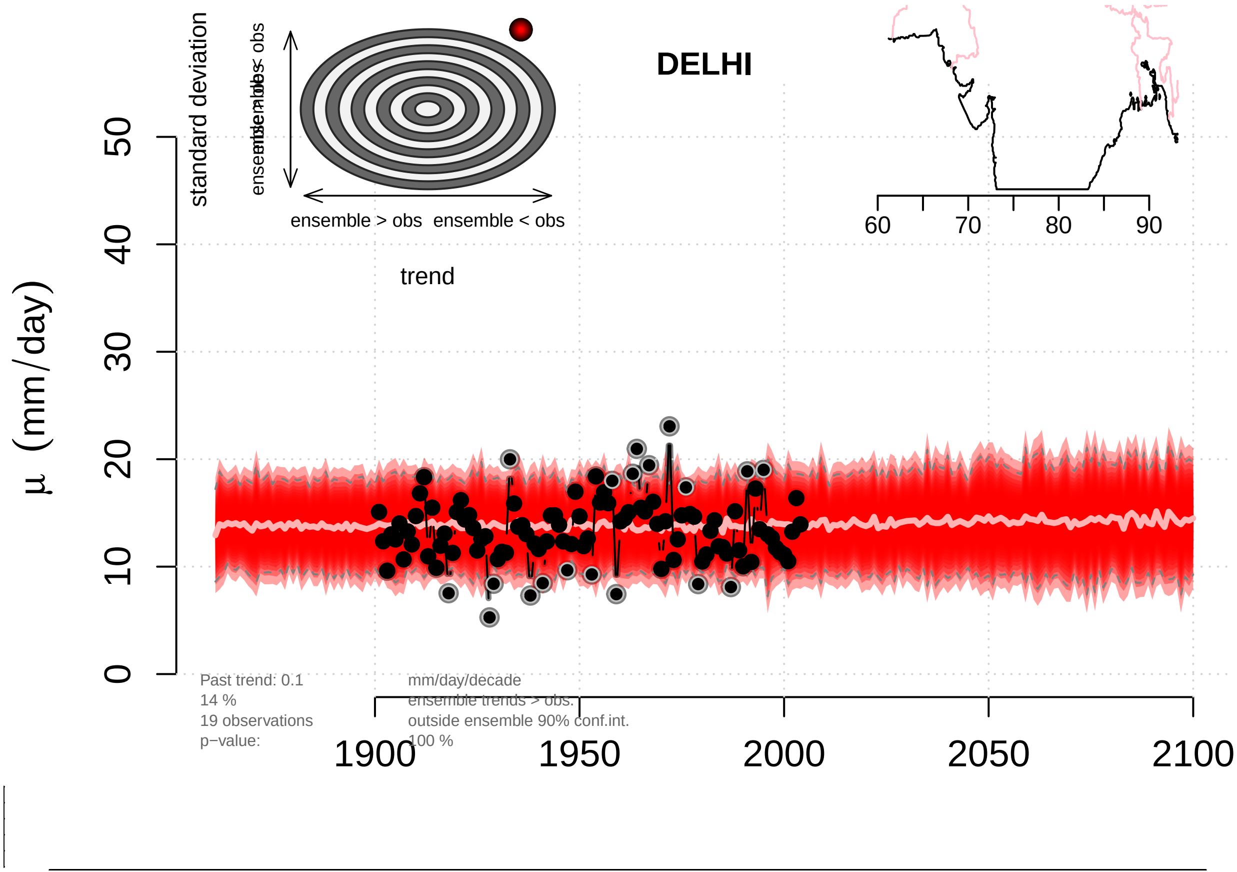
```r
## Examine the PCA of residuals for the cold monthsto see if there are any structures left
  fw.2 <- as.residual(as.station(ds.fw))
  fw.2 <- attrcp(fw,fw.2); class(fw.2) <- class(fw)
  pca.fw.2 <- PCA(fw.2)
  plot(pca.fw.2)
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "plot" is not a
## graphical parameter
```

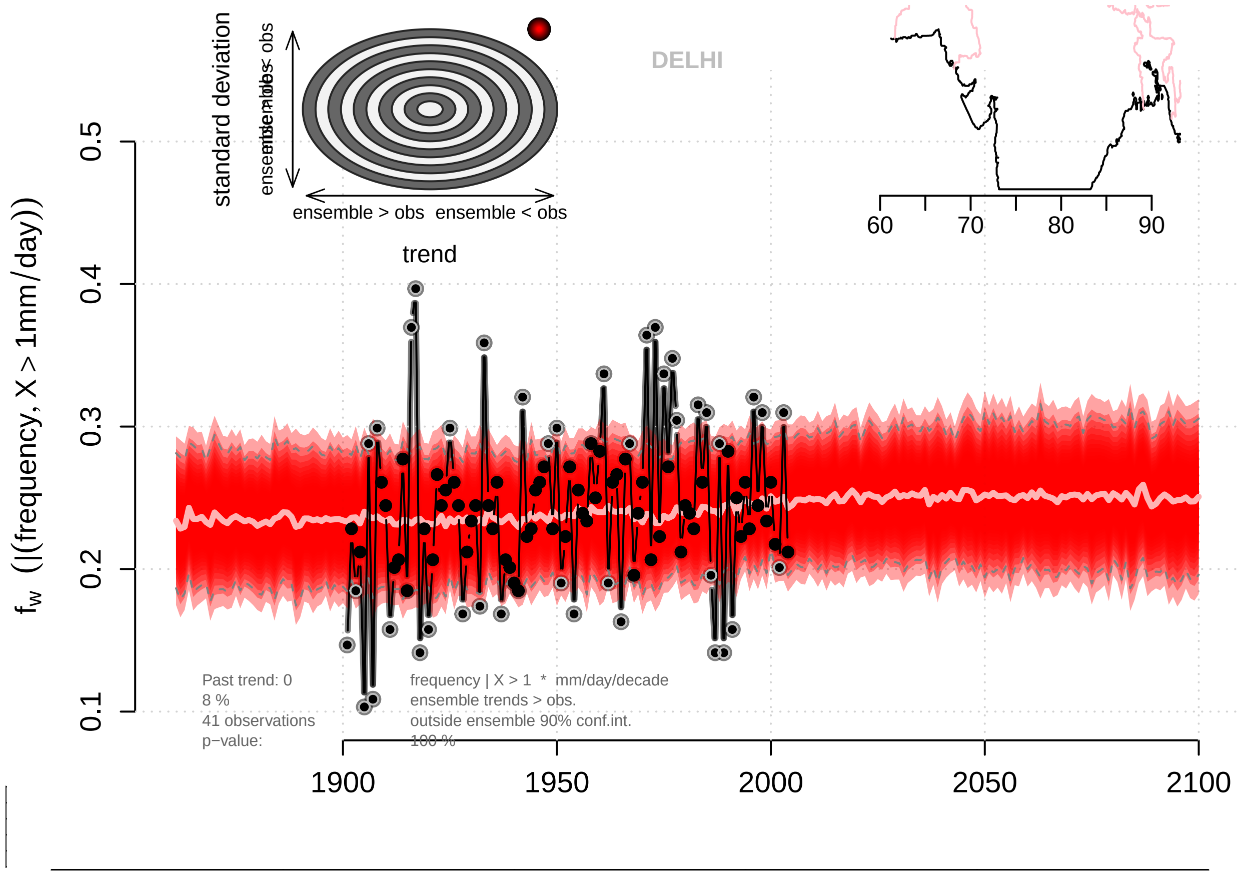
Do the downscaling based on the CMIP5 RCP4.5 ensemble

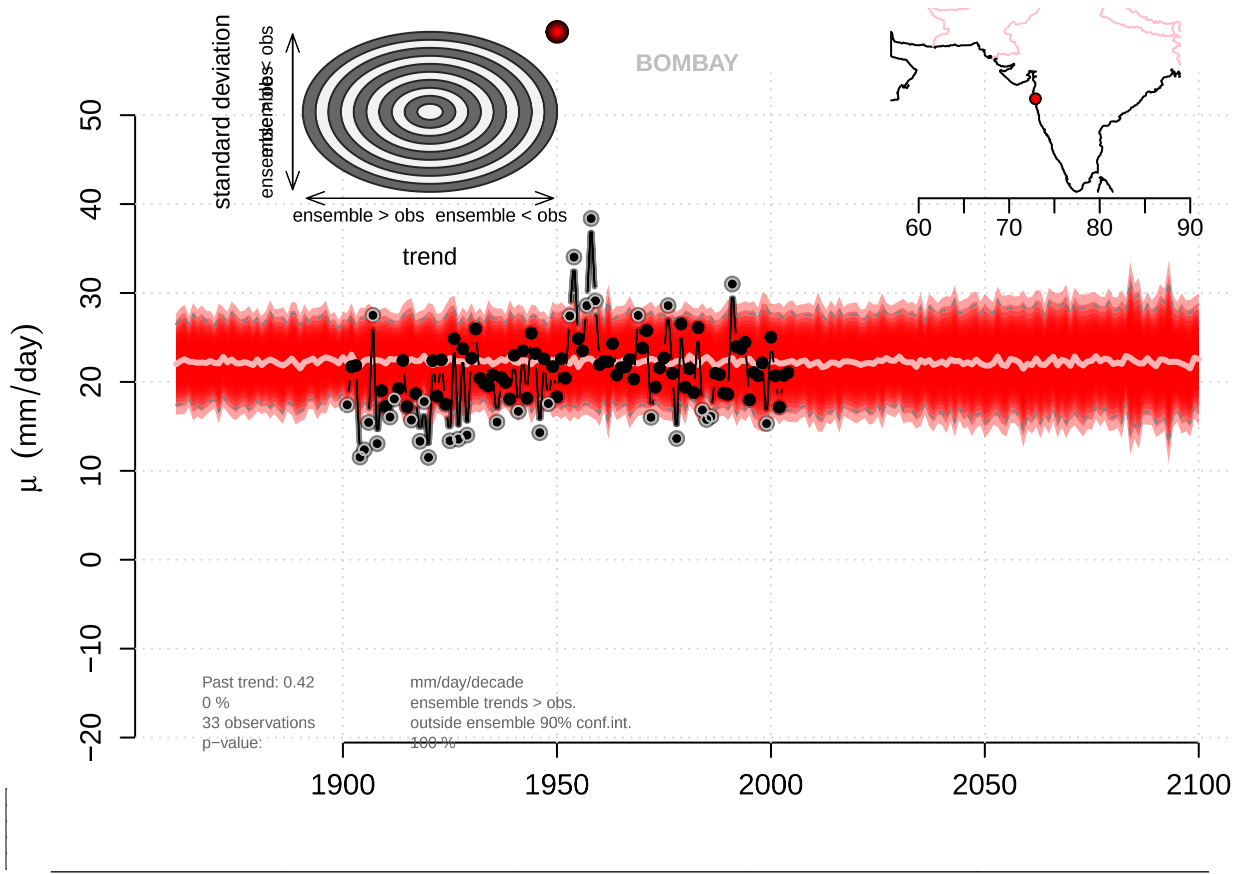
Here include the argument xfuns='tas2es' to use the option annual(tas2es(gcm)) rather than the option annual(gcm,FUN='tas2es') which will fail. Also

```r
if (!file.exists('dse.mu.climatrans.rda')) {
  index(pca.mu) <- as.Date(paste(year(pca.mu),'01-01',sep='-'))
  index(pca.fw) <- as.Date(paste(year(pca.fw),'01-01',sep='-'))
  dse.mu <- DSensemble(pca.mu,predictor=es,FUNX='tas2es',xfuns='tas2es',plot=TRUE)
  dse.fw <- DSensemble(pca.fw,predictor=slp,pattern="psl_Amon_ens_",plot=TRUE)
  save(file='dse.mu.climatrans.rda',dse.mu,dse.fw)
} else load('dse.mu.climatrans.rda')
mu.stations<-as.station(dse.mu)
fw.stations<-as.station(dse.fw)

## Plot the downscaled results for the three megacities:
for (i in 1:3) {
  plot(mu.stations[[i]],new=FALSE)
  plot(fw.stations[[i]],new=FALSE)
}
```

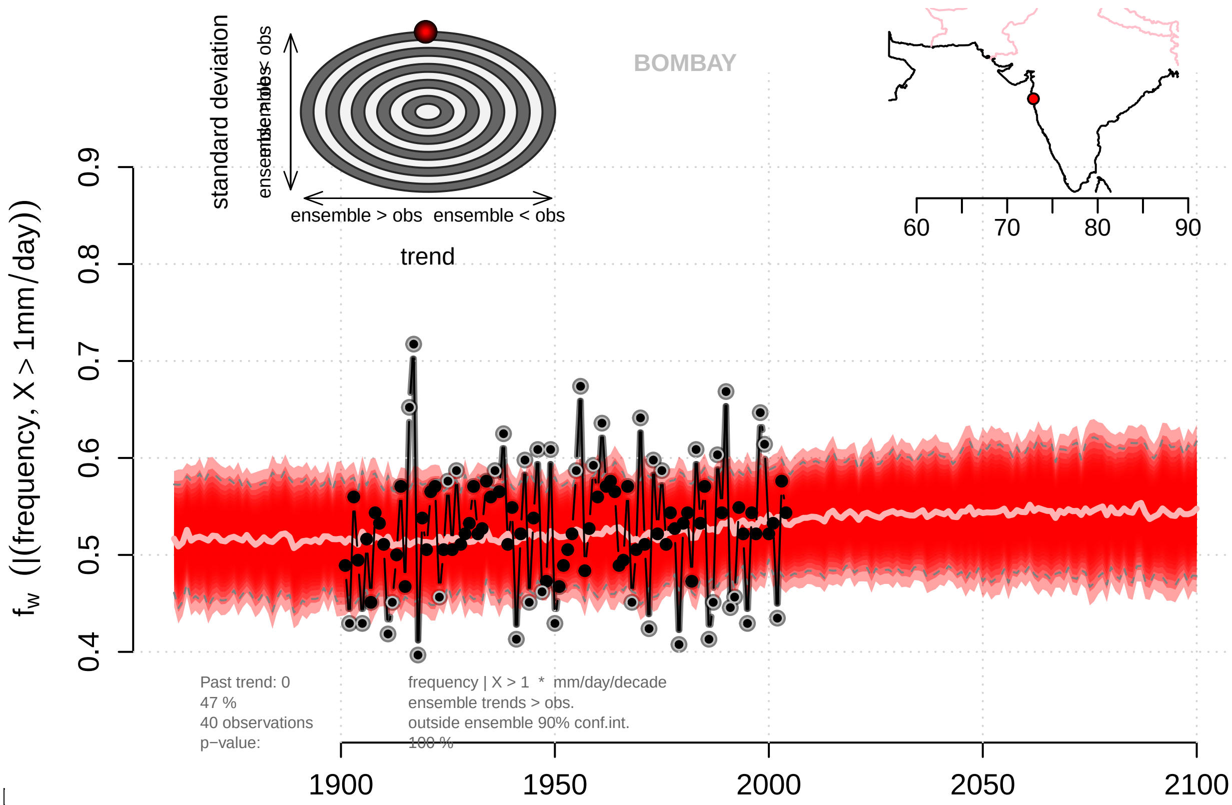


DELHI
standard deviation
ensemble < obs
ensemble > obs ensemble < obs
trend
60
70
80
90
μ (mm/day)
0
10
20
30
40
50
Past trend: 0.1
14 %
19 observations
p−value:
mm/day/decade
ensemble trends > obs.
outside ensemble 90% conf.int.
100 %
1900
1950
2000
2050
2100

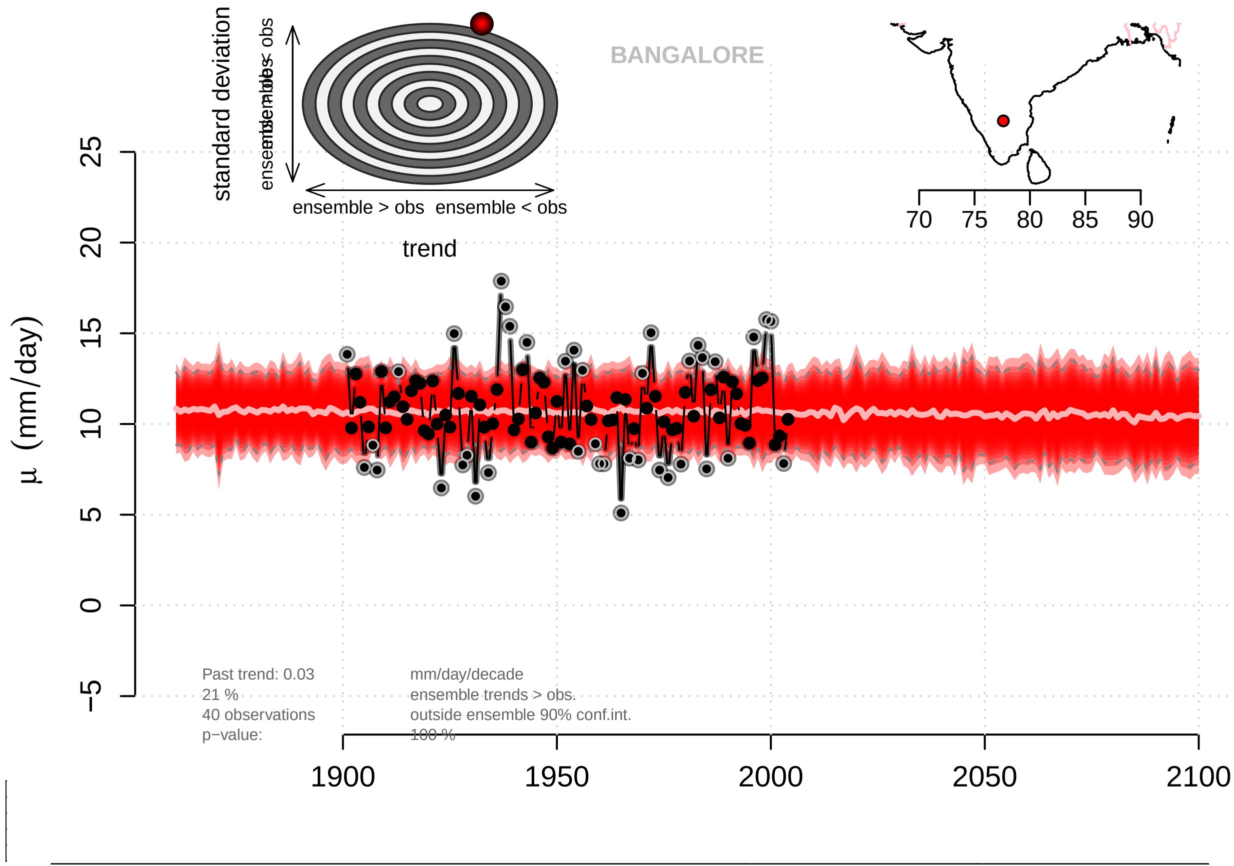

DELHI
standard deviation
ensemble > obs
ensemble < obs
trend
60
70
80
90
0.5
0.4
0.3
0.2
0.1
f_w (|(frequency, X > 1mm/day))
Past trend: 0
8 %
41 observations
p−value:
frequency | X > 1 * mm/day/decade
ensemble trends > obs.
outside ensemble 90% conf.int.
100 %
1900
1950
2000
2050
2100



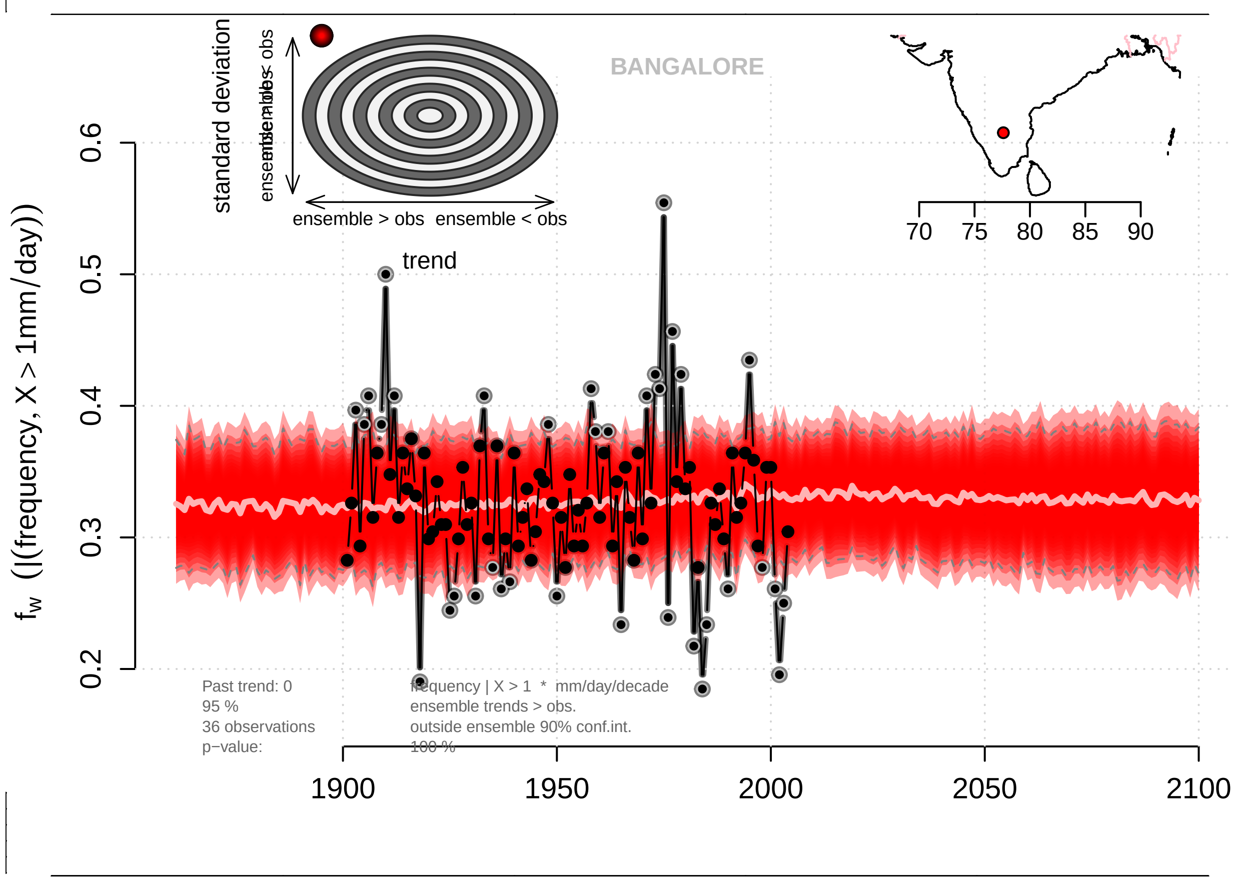
BOMBAY
standard deviation
ensemble < obs
ensemble > obs
ensemble > obs
ensemble < obs
trend
μ (mm/day)
Past trend: 0.42
mm/day/decade
0 %
ensemble trends > obs.
33 observations
outside ensemble 90% conf.int.
p−value:
100 %



standard deviation
ensemble > obs
ensemble < obs
ensemble > obs
ensemble < obs
trend
BOMBAY
60
70
80
90
0.4
0.5
0.6
0.7
0.8
0.9
f_w (|(frequency, X > 1mm/day))
Past trend: 0
47 %
40 observations
p−value:
frequency | X > 1 * mm/day/decade
ensemble trends > obs.
outside ensemble 90% conf.int.
100 %
1900
1950
2000
2050
2100



standard deviation
ensemble > obs
ensemble < obs
ensemble > obs ensemble < obs
BANGALORE
70 75 80 85 90
trend
μ (mm/day)
25
20
15
10
5
0
−5
Past trend: 0.03
21 %
40 observations
p−value:
mm/day/decade
ensemble trends > obs.
outside ensemble 90% conf.int.
100 %
1900 1950 2000 2050 2100



standard deviation
ensemble > obs
ensemble < obs
ensemble > obs ensemble < obs
BANGALORE
70 75 80 85 90
trend
f_w (|(frequency, X > 1mm/day))
0.6
0.5
0.4
0.3
0.2
Past trend: 0
95 %
36 observations
p−value:
frequency | X > 1 * mm/day/decade
ensemble trends > obs.
outside ensemble 90% conf.int.
100 %
1900 1950 2000 2050 2100

Maximum temperature

Check the long-term change: historic trend in the mean - include all seasons, but to get a clearer ipression, plot the anomalies (i.e. exclude the mean seasonal cycle).

```
  load('climatrans.tx.rda')
  plot(as.4seasons(anomaly(climatrans.tx)))
  for(i in 1:3) lines(trend(subset(as.4seasons(anomaly(climatrans.tx)),is=i)))
  grid()
```

Downscale the seasonal seasonal mean maximum temperature and then use the mean temperature to estimate the number of hot days assuming that the daily distribution is approximately Gaussian for a given season.

```
## The predictor
  predictor <- retrieve('air.mon.mean.nc',param='air',lon=c(60,90),lat=c(10,35))
```

```
## [1] "Warning : Calendar attribute has not been found in the meta data and will be set automatically.
```

```
  predictor <- aggregate(subset(predictor,it='jja'),year,FUN='mean')
```

```
## Warning in if (deparse(substitute(by)) == "year") {: the condition has
## length > 1 and only the first element will be used
```

```
  index(predictor) <- year(predictor)
  eof.t2m <- EOF(predictor)
## The predictand
  Tx <- subset(as.4seasons(climatrans.tx),it='jja')
  pca.tx <- PCA(Tx)
  class(pca.tx) <- c("pca","station", "annual", "zoo")
  index(pca.tx) <- year(pca.tx)
  ds.tx <- DS(pca.tx,eof.t2m)
```

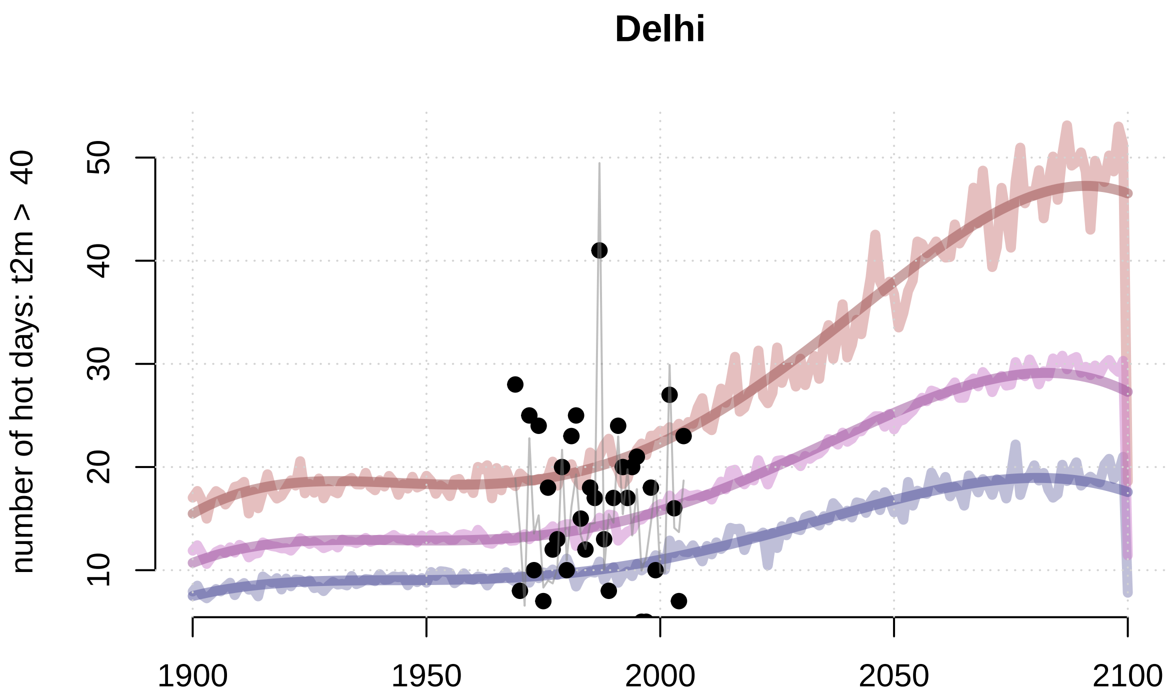
```
##
  |
  |                                                                 |   0%
  |
  |======================                                           |  33%
  |
  |===========================================                      |  67%
  |
  |=================================================================| 100%
```

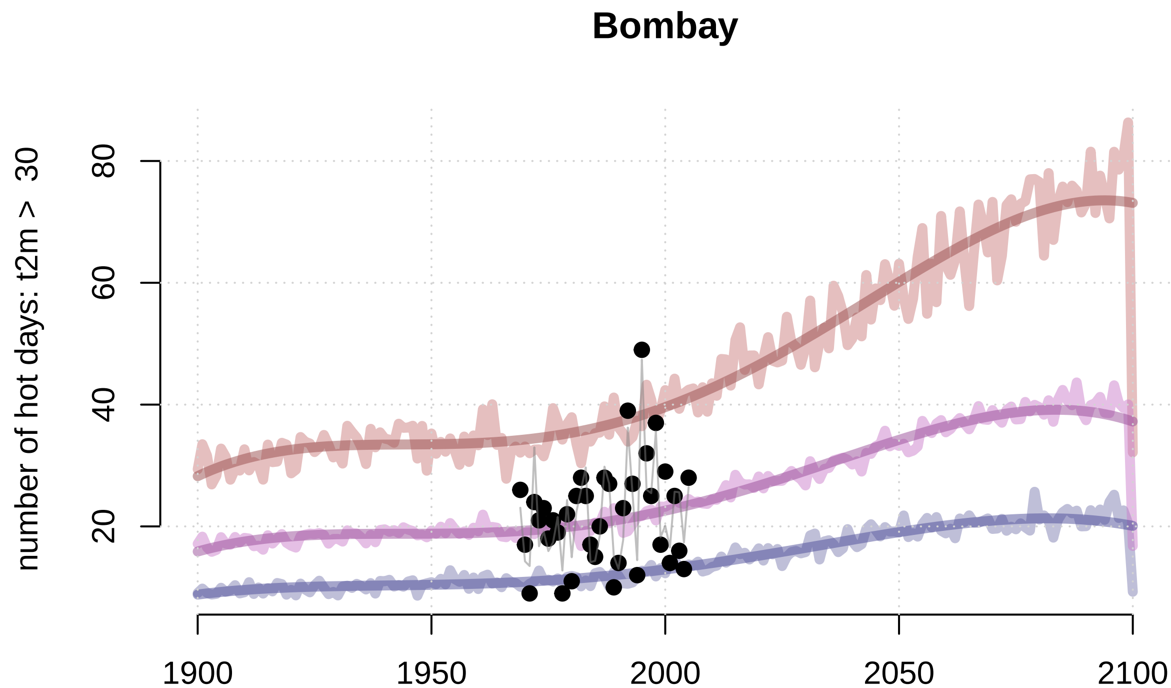
```
## Check the residuals
  tx.2 <- as.residual(as.station(ds.tx))
  tx.2 <- attrcp(Tx,tx.2); class(tx.2) <- class(Tx)
  pca.tx.2 <- PCA(tx.2)
  plot(pca.tx.2)
```

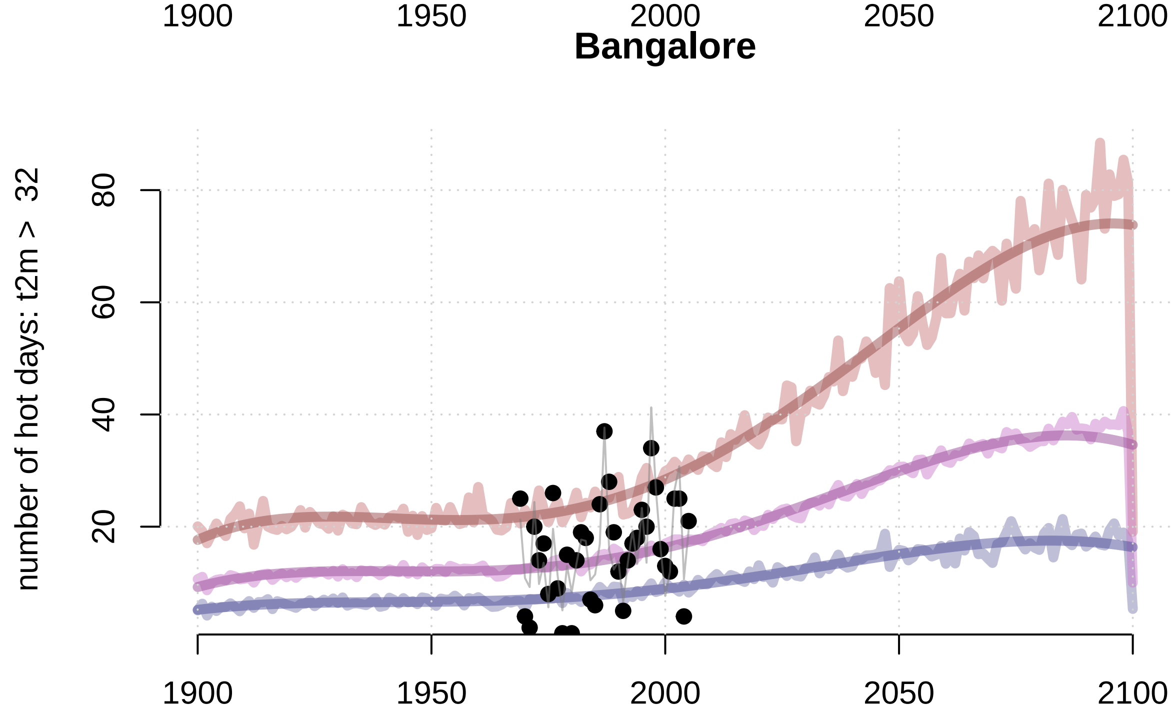
```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "plot" is not a
## graphical parameter
```

Apply the analysis of number of hot days

```
trh <- c(40,30,32)
for (i in 1:3) {
  y <- subset(climatrans.tx,is=i)
  if (!file.exists(paste('dse.tx.climatrans.',loc(y),'.rda',sep=''))) {
    dse.tx <- DSensemble.t2m(y,biascorrect=TRUE,type='ncdf4',
                      predictor=predictor,nmin=60,verbose=FALSE)
    save(file=paste('dse.tx.climatrans.',loc(y),'.rda',sep=''),dse.tx)
  } else load(paste('dse.tx.climatrans.',loc(y),'.rda',sep=''))

  ## The function hotsummerdays estimates the number of hot days based on
  ## the assumption that the temperature is close to being normally distributed
  hw <- hotsummerdays(x=y,dse=dse.tx,threshold=trh[i],plot=FALSE)
  plot(hw,new=FALSE)
}
```

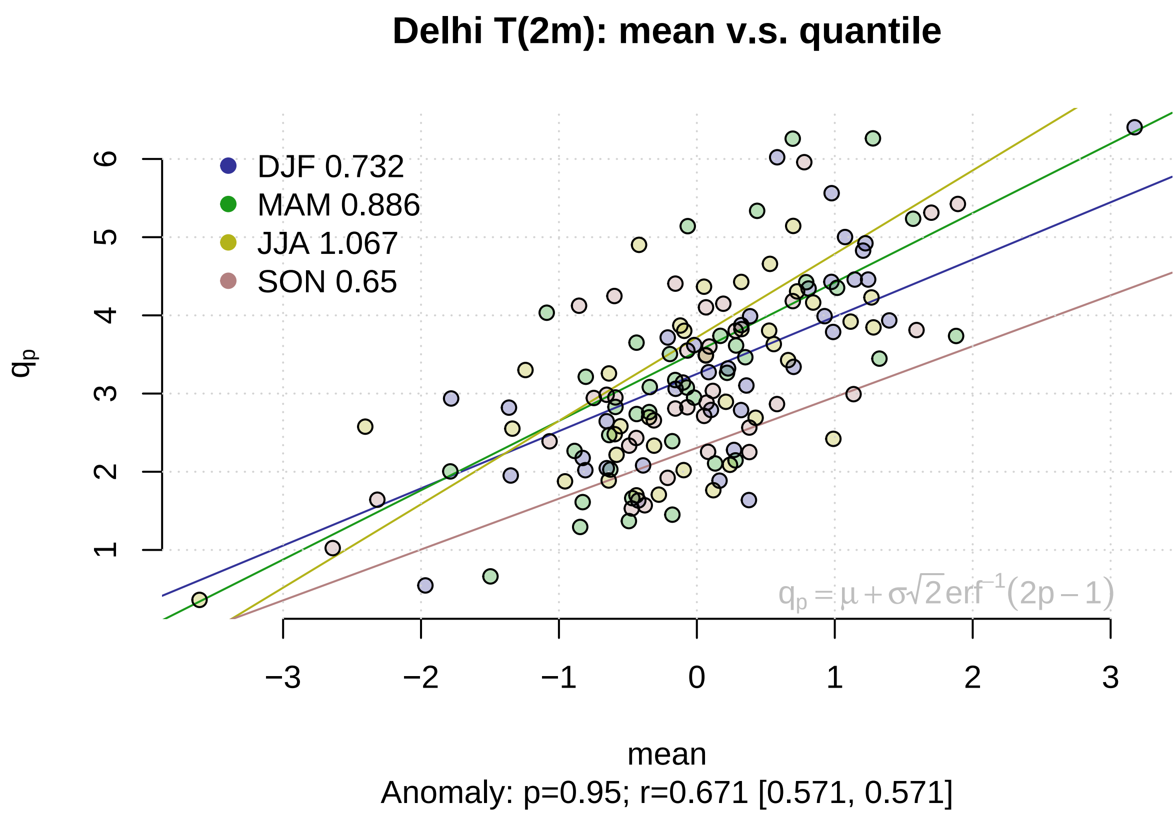






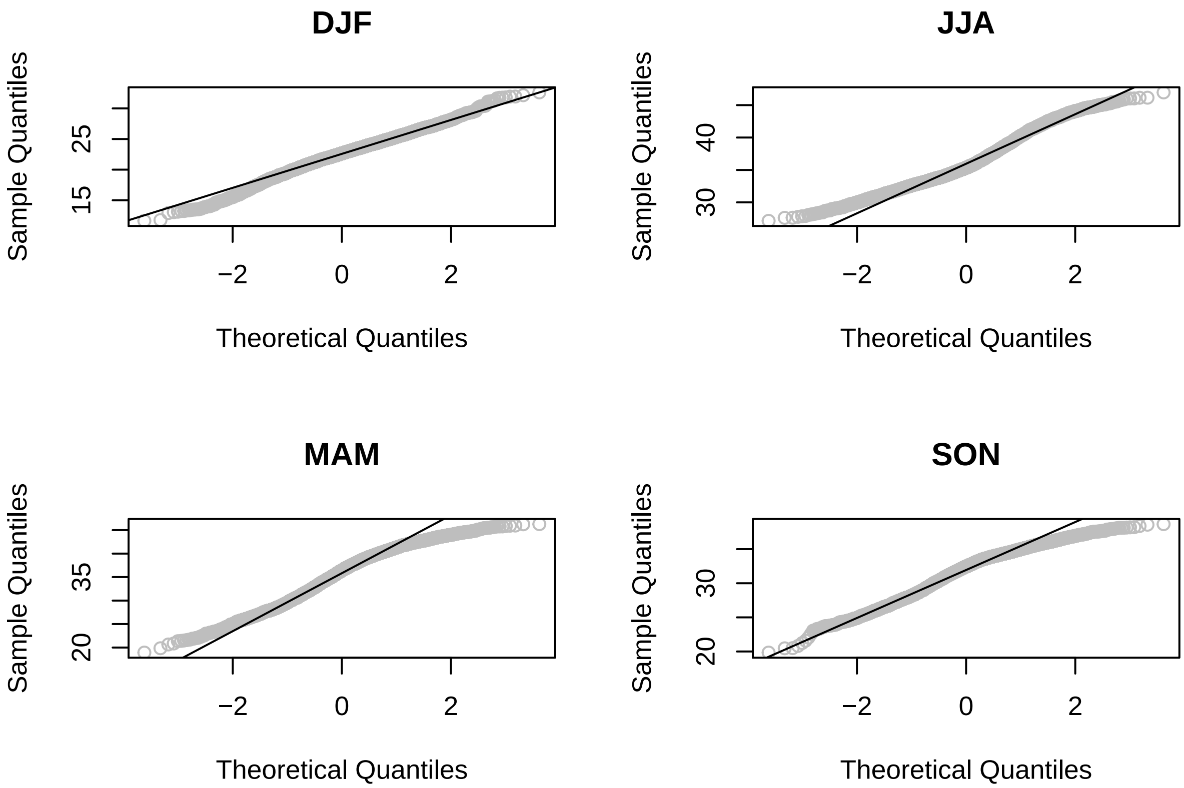
Check the correlation between the seasonal percentiles and the seasonal mean temperature, assuming the temperature is normally distributed.

```
txq95 <- diagnose(subset(climatrans.tx,is=1))
```



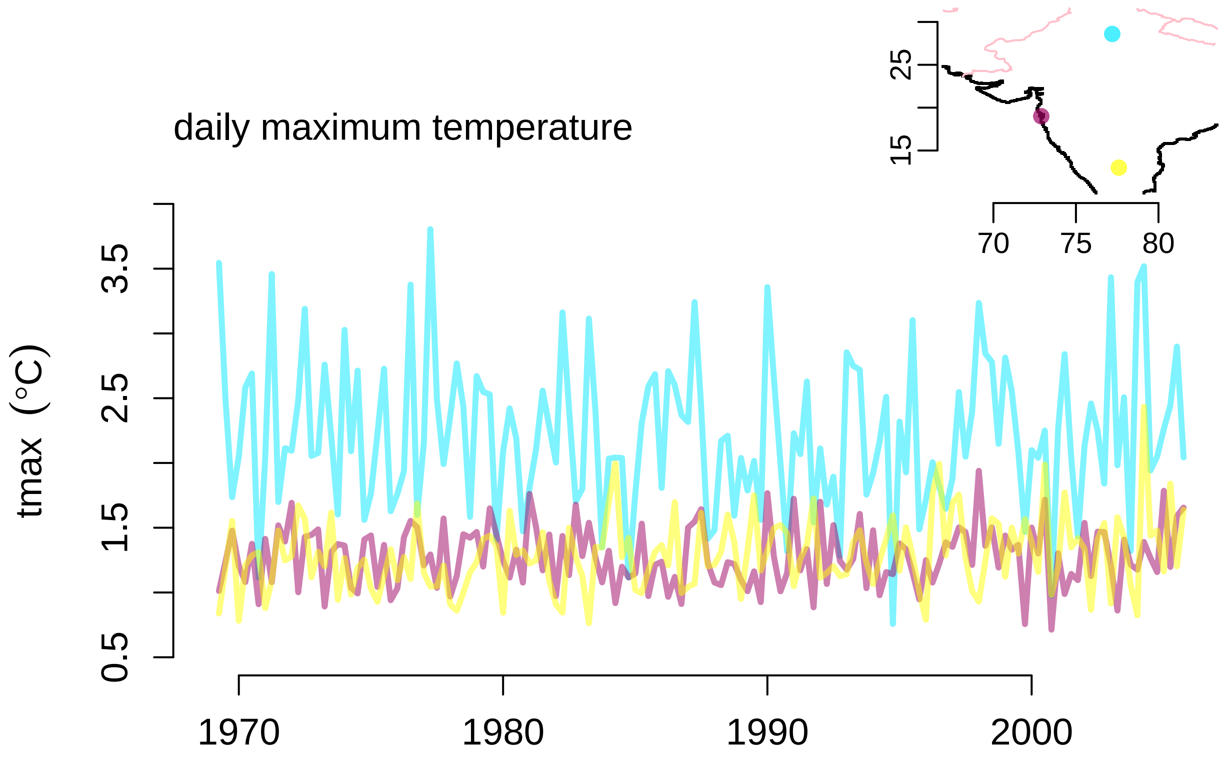
Check whether the maximum temperature is close to being normally distributed

```
## New Delhi:
  new.delhi.tx <- subset(climatrans.tx,is=1)
## Bangalore:
  bangalore.tx <- subset(climatrans.tx,is=3)
## Bombay:
  bombay.tx <- subset(climatrans.tx,is=2)

  y <- new.delhi.tx
  par(mfcol=c(2,2))
  qqnorm(coredata(subset(y,it='djf')),col='grey',main='DJF')
  qqline(coredata(subset(y,it='djf')))
  qqnorm(coredata(subset(y,it='mam')),col='grey',main='MAM')
  qqline(coredata(subset(y,it='mam')))
  qqnorm(coredata(subset(y,it='jja')),col='grey',main='JJA')
  qqline(coredata(subset(y,it='jja')))
  qqnorm(coredata(subset(y,it='son')),col='grey',main='SON')
  qqline(coredata(subset(y,it='son')))
```



Check the long-term change: historic trend in the seasonal spread (standard deviation) - the above estimates assumed that the spread is constant. The number of hot days will be even higher if the spread increases over time.

```
plot(as.4seasons(anomaly(climatrans.tx),FUN='sd'),new=FALSE)
```



daily maximum temperature
tmax (°C)
0.5
1.5
2.5
3.5
1970
1980
1990
2000
15
25
70
75
80