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)
#rm(list=ls())
## Information about the system and session
```

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)
}
```

Below is a listing of the function used to extract the raw precipitation results averaged over the Indian sub-continent.

```r
RMSE <- function(x,y) sqrt(sum((x-y)^2))/length(x)

precipcmip <- function(predictor='data/ERAINT/era-int-precip-mon.nc',
                       path="CMIP5.monthly/",rcp='rcp45',pattern="pr_Amon_ens_",
                       type='ncdf',it=c(1950,2015),plot=TRUE,verbose=FALSE,
                       lon=c(70,90),lat=c(7,30),select=NULL,xlim=NULL,fname=NULL) {

  if (is.null(fname)) fname <- paste('precipcmip.',paste(lon,lat,sep='.',collapse='-'),'.rda',sep='')

  print(paste('precipcmip: results (will be) stored in',fname))

  if (file.exists(fname)) {
    load(fname)
    return(results)
  }

  ## Diagnose the common EOFs and effect of bias-adjustment

  if (verbose) print(paste('getcmip, it=',it))

  if (verbose) print('predictor')
  if (is.character(predictor))
    ## Unit: "mm"
    rea <- retrieve(ncfile=predictor,lon=lon,lat=lat,
                    type=type,verbose=verbose) else
  if (inherits(predictor,'field'))
    rea <- subset(predictor,is=list(lon=lon,lat=lat))
  if (!is.null(it)) {
    rea <- subset(rea,it=it)
    it <- c(min(year(rea)),max(year(rea)))
  }

  ## The reanalysis
  if (verbose) print('aggregate the reanalysis')
  Rea <- aggregate.area(rea,FUN='mean')
  X <- aggregate(Rea,month,FUN='mean')
  Xt <- annual(Rea,FUN='sum')

  if (plot) plot(X,lwd=3,col='black',main='Climatology',ylim=c(0,2*round(max(X))),
                 ylab=ylab(X),xlab='Calendar month',map.type='rectangle')

  # Ensemble GCMs
  path <- file.path(path,rcp,fsep = .Platform$file.sep)
  ncfiles <- list.files(path=path,pattern=pattern,full.name=TRUE)
  N <- length(ncfiles)

  if (is.null(select)) select <- 1:N else
                       N <- length(select)
  if (verbose) {print('GCMs:'); print(path); print(ncfiles[select])}

  Z <- matrix(rep(NA,N*12),N,12)
  Zt <- matrix(rep(NA,N*201),N,201)
  gcmnm <- rep("",N); rmse <- rep(NA,N); r <- rep(NA,N)
```

```
  ## Set up a list environment to keep all the results
  if (verbose) print("loop...")
  for (i in 1:N) {
    if (verbose) print(ncfiles[select[i]])

    ## unit: "kg m-2 s-1"
    gcm <- retrieve(ncfile = ncfiles[select[i]],type=type,
                          lon=range(lon(rea)),
                          lat=range(lat(rea)),verbose=verbose)
    GCM <- aggregate.area(gcm,FUN='mean')
    ## Annual total precipitation
    zt <- annual(GCM,FUN='sum')
    i1 <- is.element(year(zt),1900:2100)
    i2 <- is.element(1900:2100,year(zt))
    Zt[i,i2] <- coredata(zt)[i1]

    if (!is.null(it)) {
      if (verbose) print('Extract some months ot a time period')
      if (verbose) print(it)
      gcm <- subset(gcm,it=it)
    }
    ## The mean annual cycle
    ## The units of the GCMs are mm/day - multiply by 30
    z <- 30*aggregate(GCM,month,FUN='mean')
    Z[i,] <- coredata(z)
    #gcmnm[i] <- attr(gcm,'model_id')
    gcmnm.i <- paste(attr(gcm,'model_id'),attr(gcm,'realization'),sep="-r")
    gcmnm.i <- gsub(' ','',gcmnm.i)
    gcmnm.i <- gsub('-','.',gcmnm.i)
    rmse[i] <- round(RMSE(coredata(X),coredata(z)),2)
    r[i] <- round(cor(coredata(X),coredata(z)),2)
    gcmnm[i] <- gcmnm.i
    print(paste(gcmnm.i,unit(GCM),unit(Rea),rmse[i],r[i]))
    if (plot) lines(z,col=rgb(i/N,0,0.5+0.5*r[i],0.1),lwd=3)
  }
  attr(Z,'GCMs') <- gcmnm
   if (plot) {
     lines(X,col='black',lwd=3)
     grid()
     figlab(paste(start(Rea),end(Rea)))
   }
  results <- list(reanalysis=X,GCMs=z,
                  obs=Xt,gcm=zoo(t(Zt),order.by=1900:2100),rmse=rmse,cor=r)
  dev.copy2pdf(file=paste('precipcmip.',
                 paste(lon,lat,sep='.',collapse='-'),'.pdf',sep=''))
  save(file=fname,results)
  return(results)
}
```

The function `hotsummerdays` is provided in `esd`, and the listing of the code is provided below:

```
hotsummerdays <- function (x, y = NULL, dse = NULL, it = "jja", threshold = 30,
    verbose = FALSE, plot = TRUE, nmin = 90, new = TRUE, ...)
{
    if (verbose)
        print("mildwinterdays")
    stopifnot(inherits(x, "station"))
    if (is.null(y))
        y <- x
    djf <- subset(x, it = it)
    djfy <- subset(y, it = it)
    nwd1 <- annual(djfy, FUN = "count", threshold = threshold,
        nmin = nmin)
    mwd1 <- annual(djf, FUN = "mean", nmin = nmin)
    cal <- data.frame(x = c(coredata(mwd1)), y = c(coredata(nwd1)))
    dfit <- glm(y ~ x, family = "poisson", data = cal)
    if (plot) {
        if (new)
            dev.new()
        par(bty = "n")
        plot(cal, pch = 19, ylim = c(0, 90), xlab = expression(paste("mean temperature ",
            (degree * C))), ylab = "number of hot days", main = loc(x))
        pre <- data.frame(x = seq(min(cal$x, na.rm = TRUE) -
            1, max(cal$x, na.rm = TRUE) + 5, by = 0.1))
        lines(pre$x, exp(predict(dfit, newdata = pre)), col = rgb(1,
            0, 0, 0.3), lwd = 3)
        if (new)
            dev.new()
        djf.sd <- sd(coredata(djf), na.rm = TRUE)
        qqnorm(coredata(djf))
        qqline(coredata(coredata(djf)), col = "red")
        grid()
    }
    if (is.null(dse))
        dse <- DSensemble.t2m(x, biascorrect = TRUE, verbose = verbose,
            plot = plot)
    djf.dse <- subset(dse, it = "djf")
    index(djf.dse) <- year(djf.dse)
    ovl <- window(djf.dse, start = year(start(x)), end = year(end(x)))
    djf.dse <- djf.dse - mean(coredata(ovl), na.rm = TRUE) +
        mean(coredata(mwd1), na.rm = TRUE)
    q1 <- data.frame(x = apply(coredata(djf.dse), 1, quantile,
        probs = 0.05, na.rm = TRUE))
    q2 <- data.frame(x = apply(coredata(djf.dse), 1, quantile,
        probs = 0.95, na.rm = TRUE))
    qm <- data.frame(x = apply(coredata(djf.dse), 1, mean, na.rm = TRUE))
    obs <- data.frame(x = coredata(mwd1))
    t <- year(index(djf.dse))
    preq1 <- exp(predict(dfit, newdata = q1))
    preq1[preq1 > 90] <- NA
    tr1 <- predict(lm(preq1 ~ t + I(t^2) + I(t^3) + I(t^4) +
        I(t^5)))
    tr1[!is.finite(preq1)] <- NA
    preq2 <- exp(predict(dfit, newdata = q2))
```

```
    preq2[preq2 > 90] <- NA
    tr2 <- predict(lm(preq2 ~ t + I(t^2) + I(t^3) + I(t^4) +
        I(t^5)))
    tr2[!is.finite(preq2)] <- NA
    prem <- exp(predict(dfit, newdata = qm))
    prem[prem > 90] <- NA
    tr3 <- predict(lm(prem ~ t + I(t^2) + I(t^3) + I(t^4) + I(t^5)))
    tr3[!is.finite(prem)] <- NA
    Nwd <- zoo(cbind(preq1, preq2, prem, tr1, tr2, tr3), order.by = t)
    nwd.pre <- zoo(exp(predict(dfit, newdata = obs)), order.by = year(mwd1))
    if (plot) {
        if (new)
            dev.new()
        par(bty = "n")
        plot(zoo(djf.dse, order.by = year(djf.dse)), plot.type = "single",
            col = rgb(0.5, 0.5, 0.5, 0.2), ylab = expression(paste("mean temperature",
                (degree * C))), xlab = "", main = loc(x))
        points(mwd1, pch = 19)
        grid()
        if (new)
            dev.new()
        par(bty = "n")
        plot(Nwd, plot.type = "single", lwd = 5, main = loc(x),
            ylim = c(0, 90), xlab = "", ylab = paste("number of hot days: T(2m) > ",
                threshold, unit(x)), col = c(rgb(0.5, 0.5, 0.7,
                0.5), rgb(0.8, 0.5, 0.5, 0.5), rgb(0.8, 0.5,
                0.8, 0.5), rgb(0.3, 0.3, 0.6, 0.5), rgb(0.6,
                0.3, 0.3, 0.5), rgb(0.6, 0.3, 0.6, 0.5)), ...)
        grid()
        points(nwd1, pch = 19)
        lines(nwd.pre, col = rgb(0.5, 0.5, 0.5, 0.5))
    }
    Nwd <- attrcp(x, Nwd)
    attr(Nwd, "unit") <- "days"
    attr(Nwd, "info") <- paste("number of hot days: t2m > ",
        threshold)
    attr(Nwd, "observation") <- nwd1
    attr(Nwd, "nwd.pre") <- nwd.pre
    index(Nwd) <- t
    class(Nwd) <- c("nevents", "zoo")
    invisible(Nwd)
}
```

Precipitation

We look at various aspects of the precipitation in order to get an idea about what is happening over the sub-continent of India and at the three Indian megacities. First, we present features from the past, such as the mean annual cycle, historical long-term trends, and statistical distribution for different aspects connected to precipitation: wet-day frequency and the wet-day mean (precipitation intencity).

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

```r
## 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 of rain gauge data

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

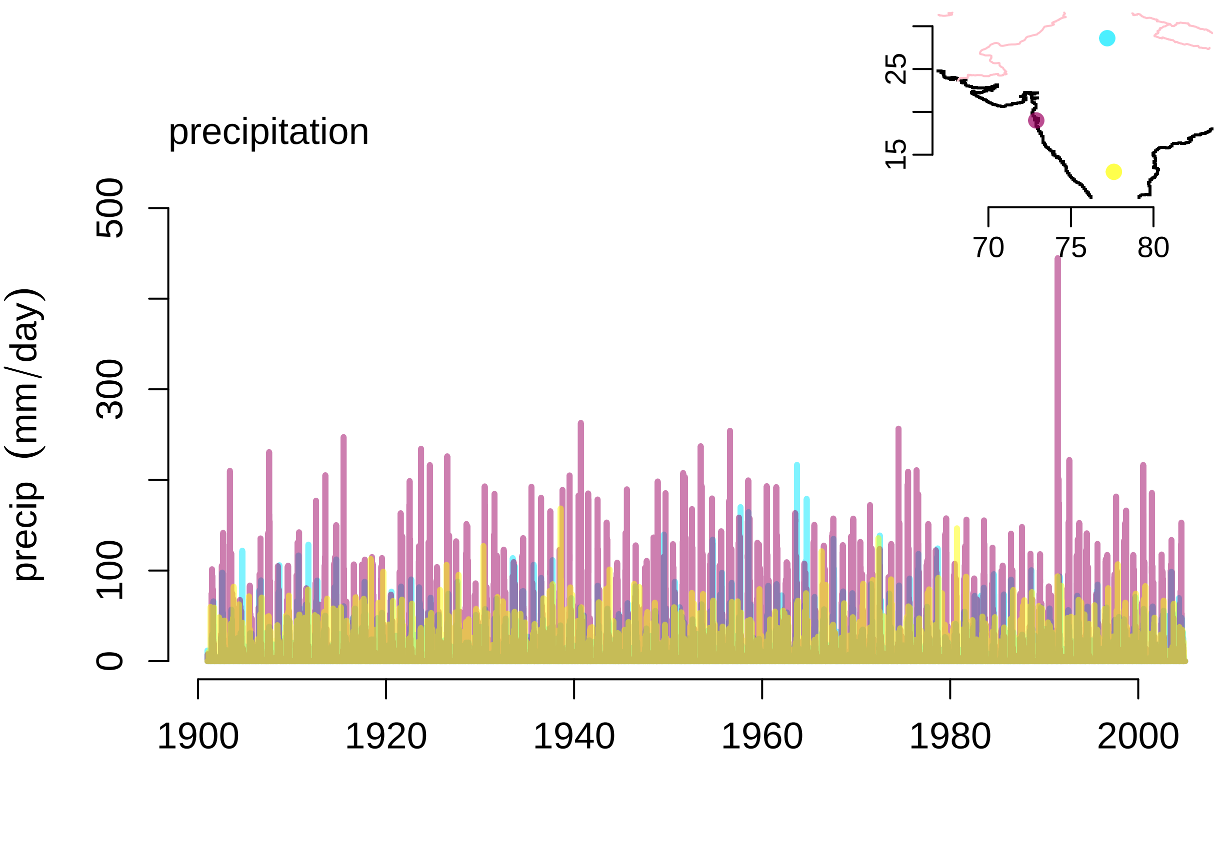
Daily station data

```
## 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.c
##   - attr(*, "info")= chr [1:3] "extracted rainfall data from IMD gridded daily data" "extracted rainfa
##    Index:  Date[1:37986], format: "1901-01-01" "1901-01-02" "1901-01-03" "1901-01-04" ...
```

```
plot(climatrans.pr,new=FALSE)
```

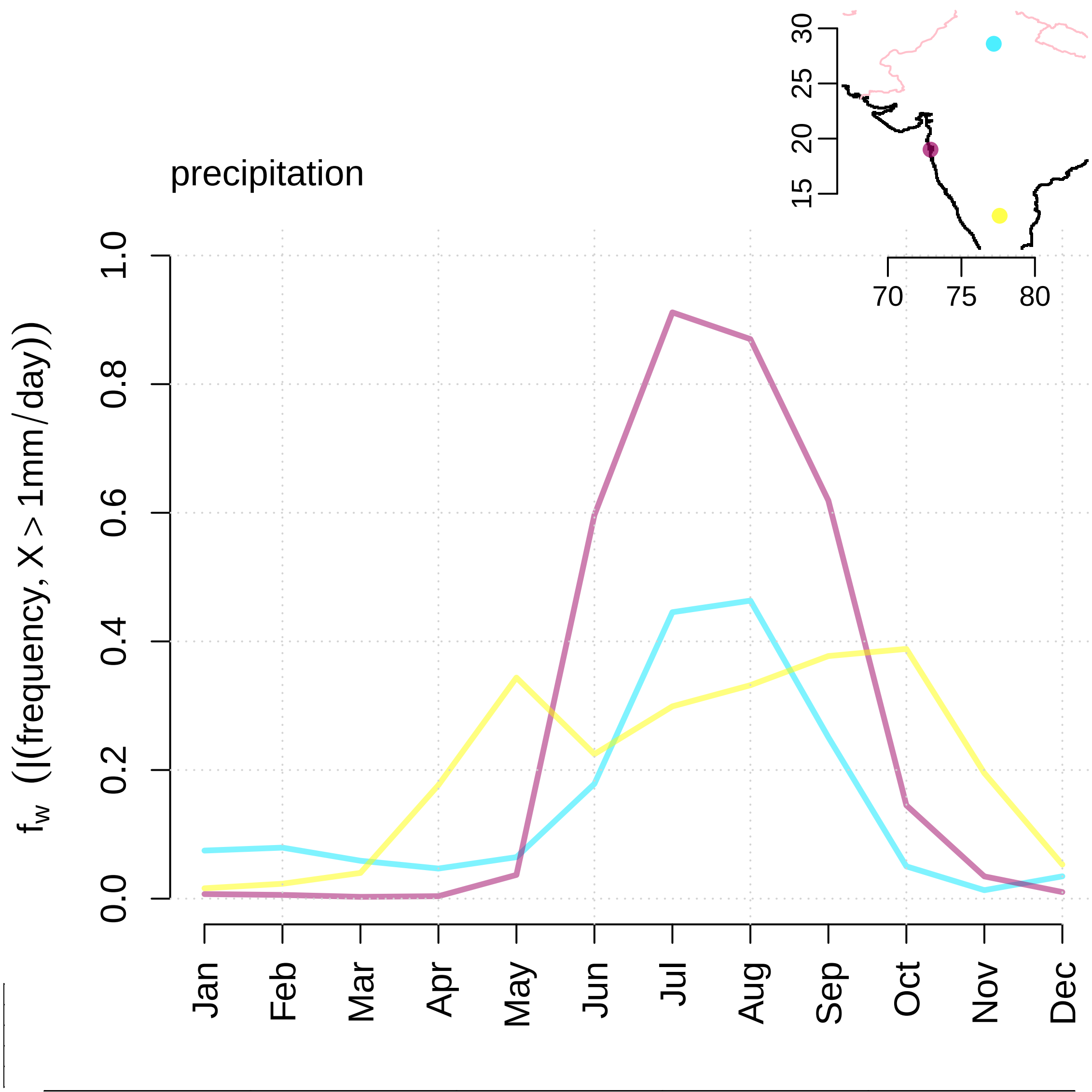


Sensitivity tests

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.

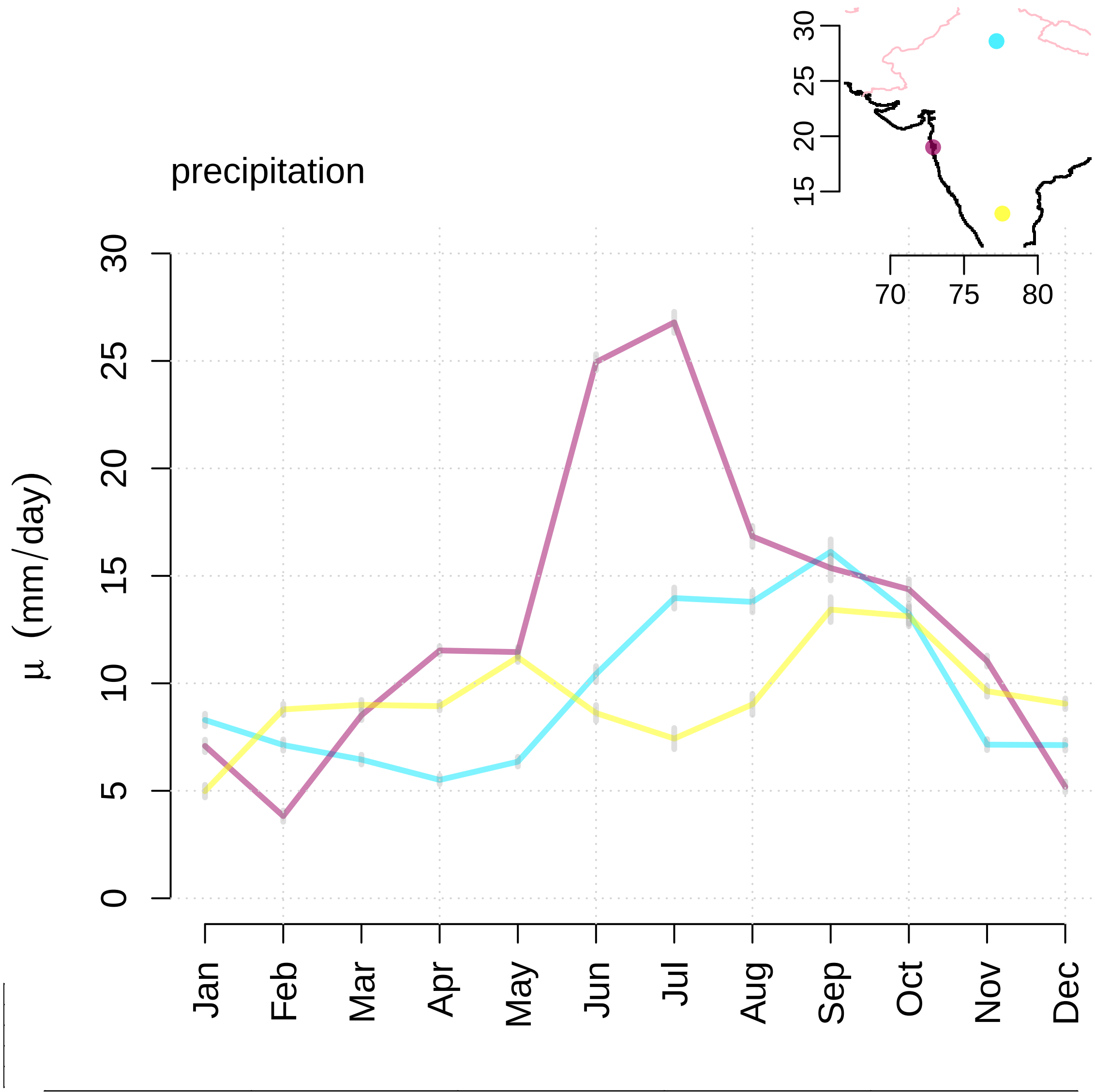
Examine the mean seasonal cycle to see if it responds to systematic forcings - the simples and most obvious being the seasonal variation in the solar inclination and the monsoon season:

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



The monsoon season is clearly visible in the wet-day frequency f_w, being a period when it rains almost every day in Mumbay (Bombay) to somewhat less than every other day in the two other cities New Delhi and Bangalore. The duration of the Monsoon season is longest in Bangalore (~Apr-Oct) and shortest in New Delhi (~Jun-Sep).

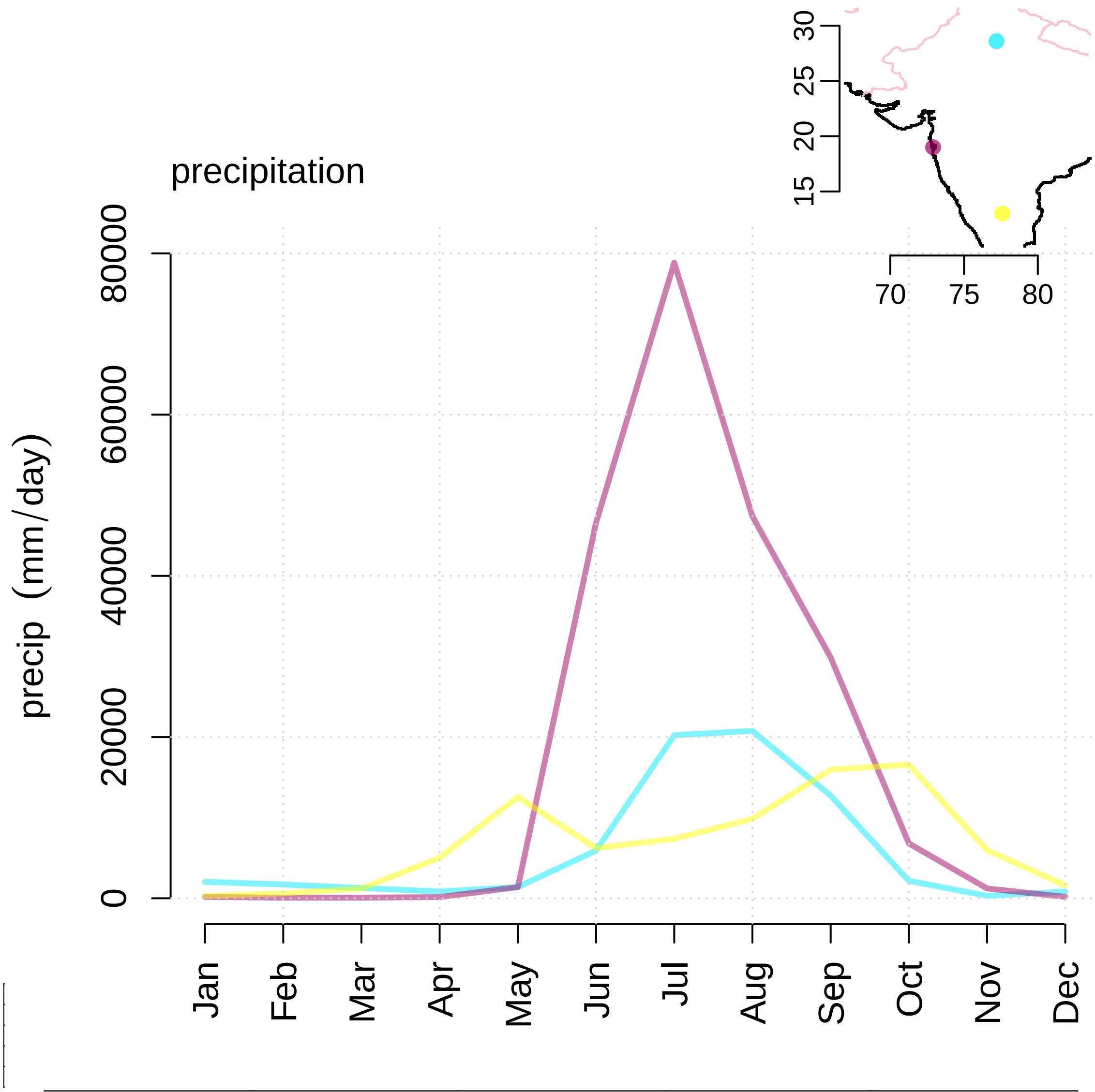
```
## Wet-day mean
plot(aggregate(climatrans.pr,month,'wetmean'),new=FALSE)
grid()
```



There are less pronounced difference in the precipitation intencity μ between the Monsoon season and the other Calendar months, although the most intense rains tend to take place in Mumbay during June and later in the season in New Delhi (Sep) and Bangalore (Sep-Oct).

Both the frequency and intensity of the rains are typically higher in Mumbay than in New Delhi and Bangalore, and Mumbay has at outset a more challenging situation in terms of coping with extreme rains than the two other megacities. This difference is also clearly visible in the typical monthly precipitation totals, which are much higher for Mumbay and peaks in July when both f_w and μ tend to be high.

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



Testing the precipitation statistics.

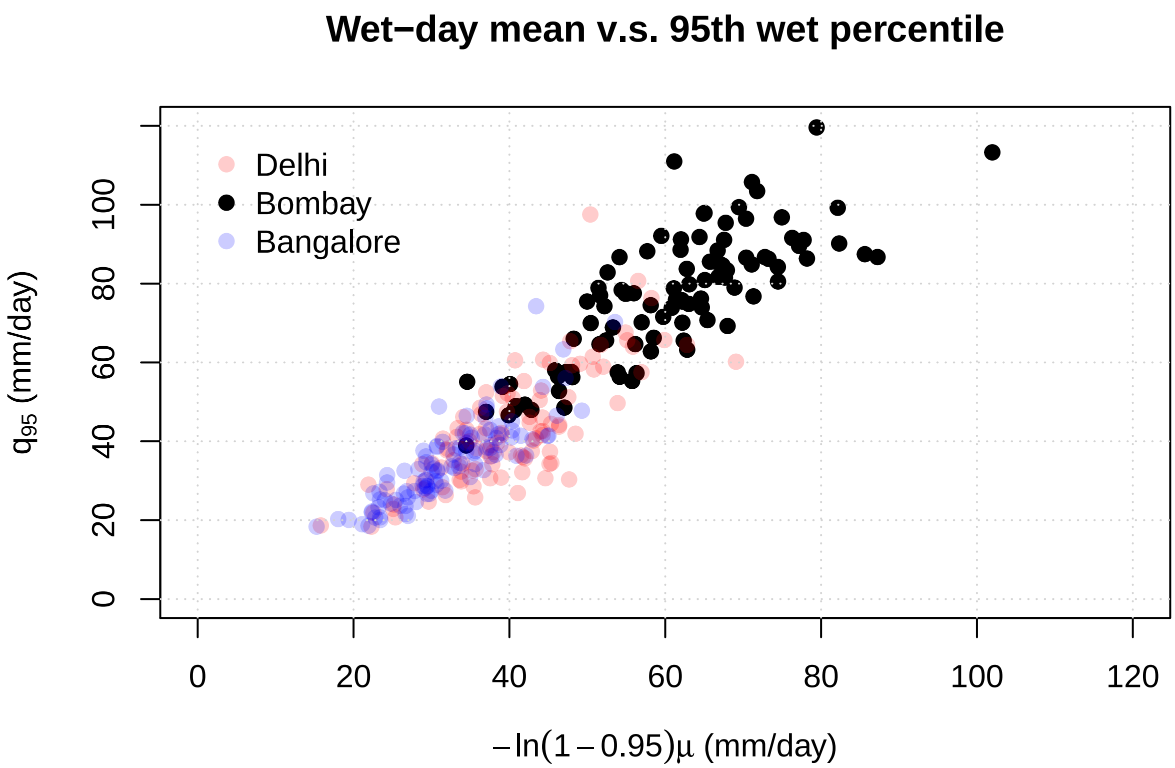
The idea that the wet-day precipitation approcimately follows the exponential distribution enbles a simplified analysis and interpreation of the reainfall statistics and the prediction/projection of heavy precipitation events.

We concentrate on the monsoon season and estimate annual aggregated statistics for the months May-October:

```
wq95 <- function(x) {x <- x[is.finite(x)]; x <- x[x >= 1]; wq95 <- quantile(x,probs=0.95); wq95}
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')
nd <- aggregate(subset(climatrans.pr,it=month.abb[season]),year,'nv')
ndhr <- aggregate(subset(climatrans.pr,it=month.abb[season]),year,'count',threshold=100)
q95 <- aggregate(subset(climatrans.pr,it=month.abb[season]),year,'wq95')
```

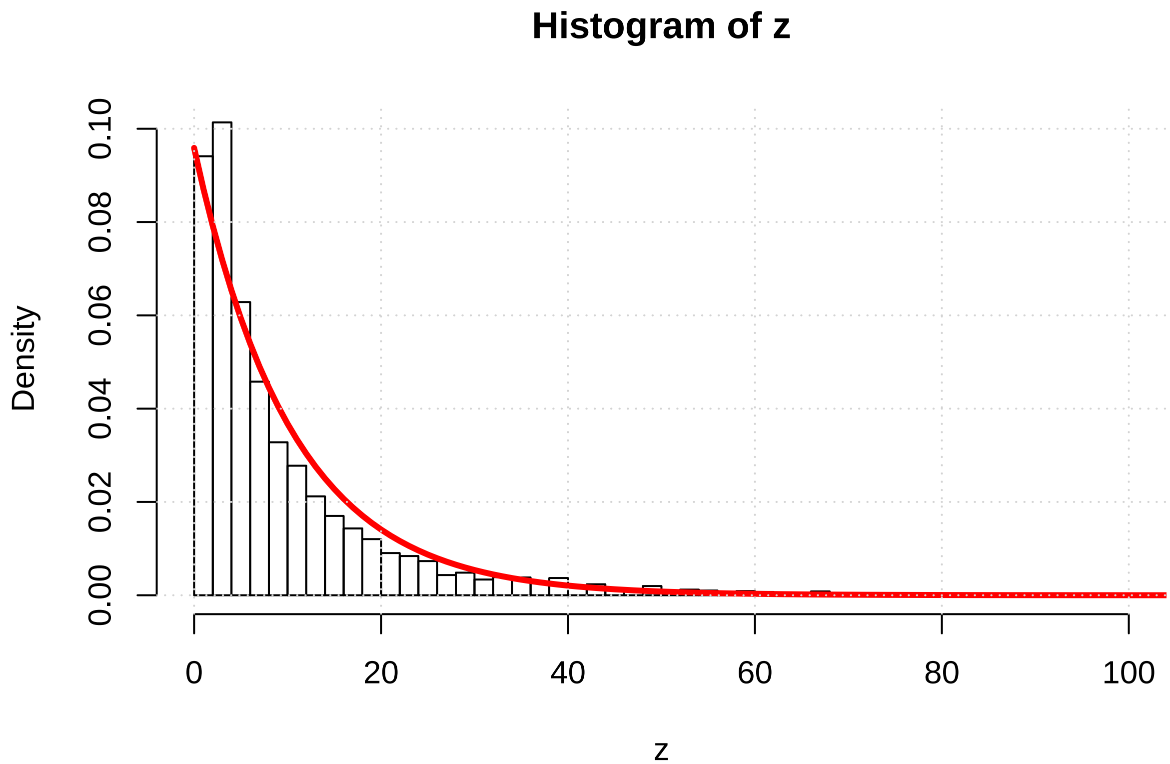
A simple test to see if the data follows the exponential distribution is to compare the data to the exponential distribution - if the data is exponentially distributed, then the wet-day mean precipitation μ and the 95-percentile q_{95} are propportional by $q_p = -ln(1-p)\mu$.

```
## 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')
```



another way to visualise this is to compare the histogram of the preciptation data with an exponential distribution

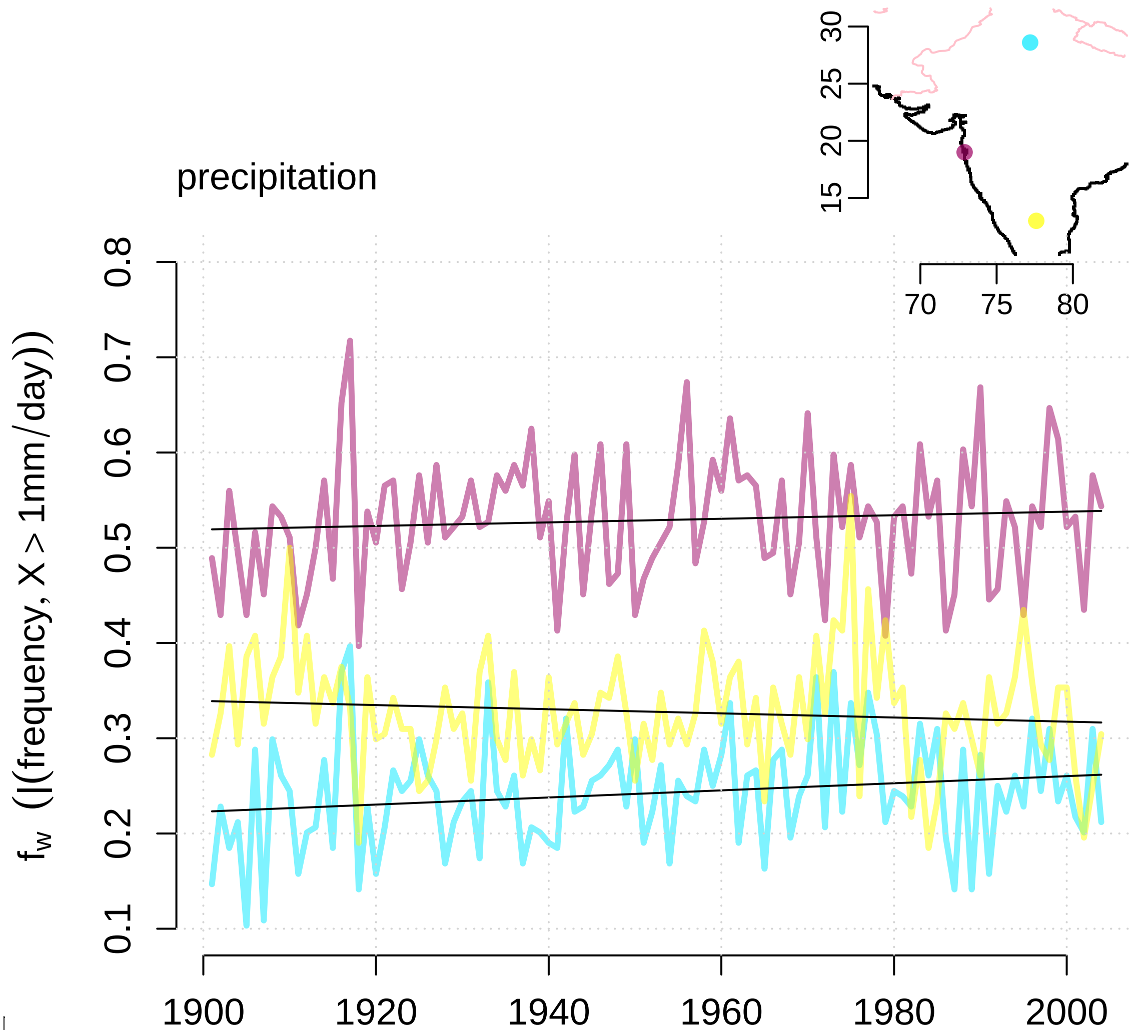
```
z <- climatrans.pr[,3]; z <- z[z > 1]
hist(z,breaks=seq(0,500,by=2),xlim=c(0,100),freq=FALSE)
lines(seq(0,250,by=1),dexp(seq(0,250,by=1),rate=1/mean(z)),lwd=3,col='red')
grid()
```



Historical long-term trends in precipitation

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)))
grid()
```




```r
print(trend(fw,result='pval'))
```

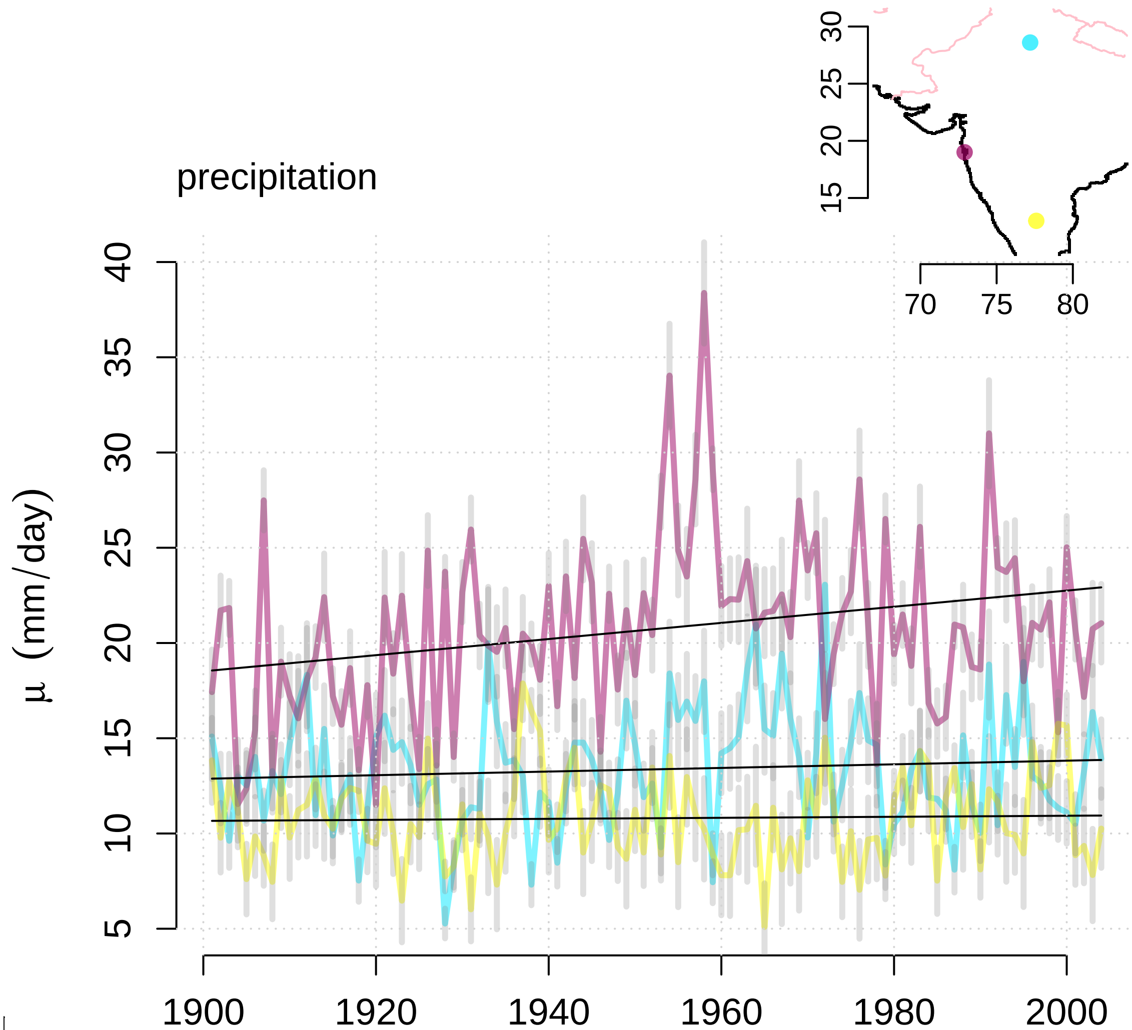
```
## [1] 0.04878119 0.38033384 0.27764397
## attr(,"location")
## [1] "Delhi"     "Bombay"    "Bangalore"
## attr(,"longitude")
## [1] 77.2 72.9 77.6
## attr(,"latitude")
## [1] 28.6 19.0 13.0
## attr(,"altitude")
## [1] NA NA NA
## attr(,"cntr")
## [1] "India" "India" "India"
## attr(,"stid")
## [1] NA NA NA
## attr(,"history")
## attr(,"history")$call
```

```
## attr(,"history")$call[[1]]
## trend.station(fw, result = "pval")
##
##
## attr(,"history")$timestamp
## [1] "Wed Aug  3 14:13:37 2016"
##
## attr(,"history")$session
## attr(,"history")$session$R.version
## [1] "R version 3.1.3 (2015-03-09)"
##
## attr(,"history")$session$esd.version
## [1] "esd_1.2"
##
## attr(,"history")$session$platform
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

The long-term trends only hint to potentially slight changes, and only the trend in New Delhi appears to be statistically significant at the 5% level.

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)))
grid()
```




```
print(trend(mu,result='pval'))
```

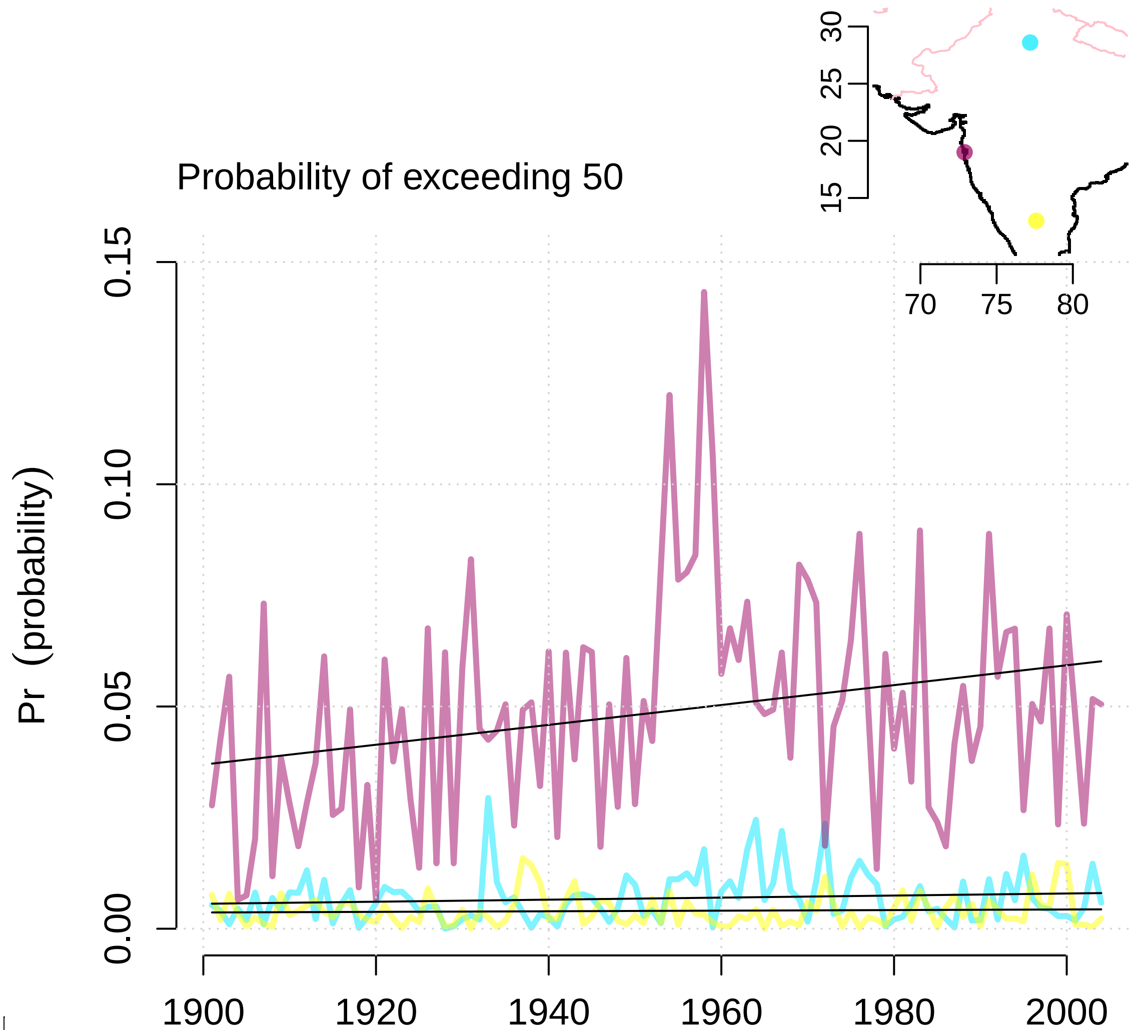
```
## [1] 0.357517060 0.004662834 0.736730369
## attr(,"location")
## [1] "Delhi"     "Bombay"    "Bangalore"
## attr(,"longitude")
## [1] 77.2 72.9 77.6
## attr(,"latitude")
## [1] 28.6 19.0 13.0
## attr(,"altitude")
## [1] NA NA NA
## attr(,"cntr")
## [1] "India" "India" "India"
## attr(,"stid")
## [1] NA NA NA
## attr(,"history")
## attr(,"history")$call
```

```
## attr(,"history")$call[[1]]
## trend.station(mu, result = "pval")
##
##
## attr(,"history")$timestamp
## [1] "Wed Aug  3 14:13:40 2016"
##
## attr(,"history")$session
## attr(,"history")$session$R.version
## [1] "R version 3.1.3 (2015-03-09)"
##
## attr(,"history")$session$esd.version
## [1] "esd_1.2"
##
## attr(,"history")$session$platform
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

For μ, only the record for Mumbay suggest a significant long-term trend at the 5%-level. There were a couple of years during the 1950s which stand out as unique seasons with particularly intense rainfall in Mumbay.

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, we can approximately quantify the probability for days receiving more than 50 mm of rain.

```
## Probability of exceedin 50 mm/day
Pr <- PrexpPr(mu,fw,x0=50)
plot(Pr,new=FALSE,errorbar=FALSE)
for(i in 1:3) lines(trend(subset(Pr,is=i)))
grid()
```



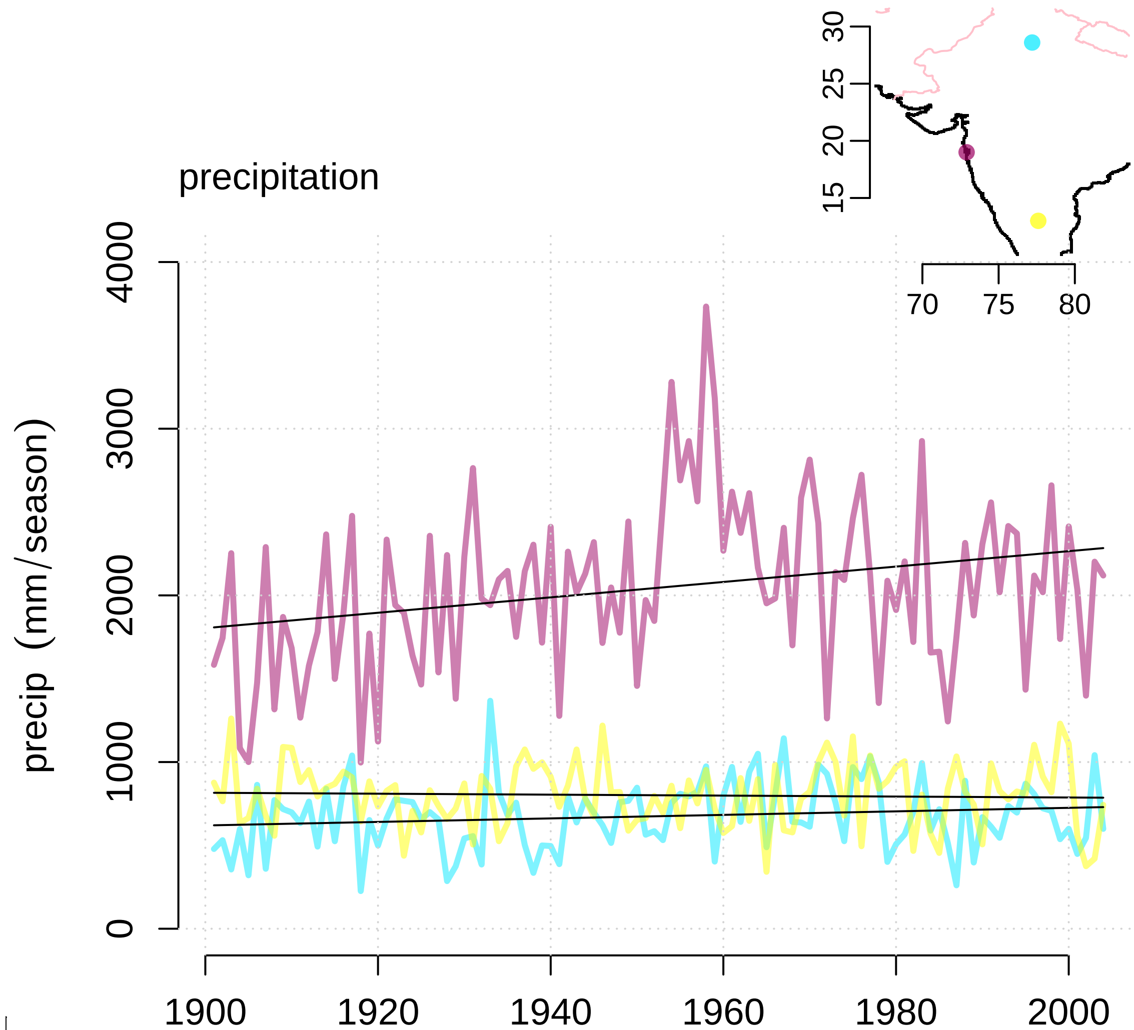

```
print(trend(Pr,result='pval'))
```

```
## [1] 0.204772035 0.005204967 0.564496266
## attr(,"location")
## [1] "Delhi"     "Bombay"    "Bangalore"
## attr(,"longitude")
## [1] 77.2 72.9 77.6
## attr(,"latitude")
## [1] 28.6 19.0 13.0
## attr(,"altitude")
## [1] NA NA NA
## attr(,"cntr")
## [1] "India" "India" "India"
## attr(,"stid")
## [1] NA NA NA
## attr(,"history")
## attr(,"history")$call
```

```
## attr(,"history")$call[[1]]
## trend.station(Pr, result = "pval")
##
##
## attr(,"history")$timestamp
## [1] "Wed Aug  3 14:13:42 2016"
##
## attr(,"history")$session
## attr(,"history")$session$R.version
## [1] "R version 3.1.3 (2015-03-09)"
##
## attr(,"history")$session$esd.version
## [1] "esd_1.2"
##
## attr(,"history")$session$platform
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

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

```
## Aggregate to annual totals
pt <- annual(climatrans.pr,'sum')
attr(pt,'unit') <- 'mm/season'
plot(pt,new=FALSE)
for(i in 1:3) lines(trend(subset(pt,is=i)))
grid()
```



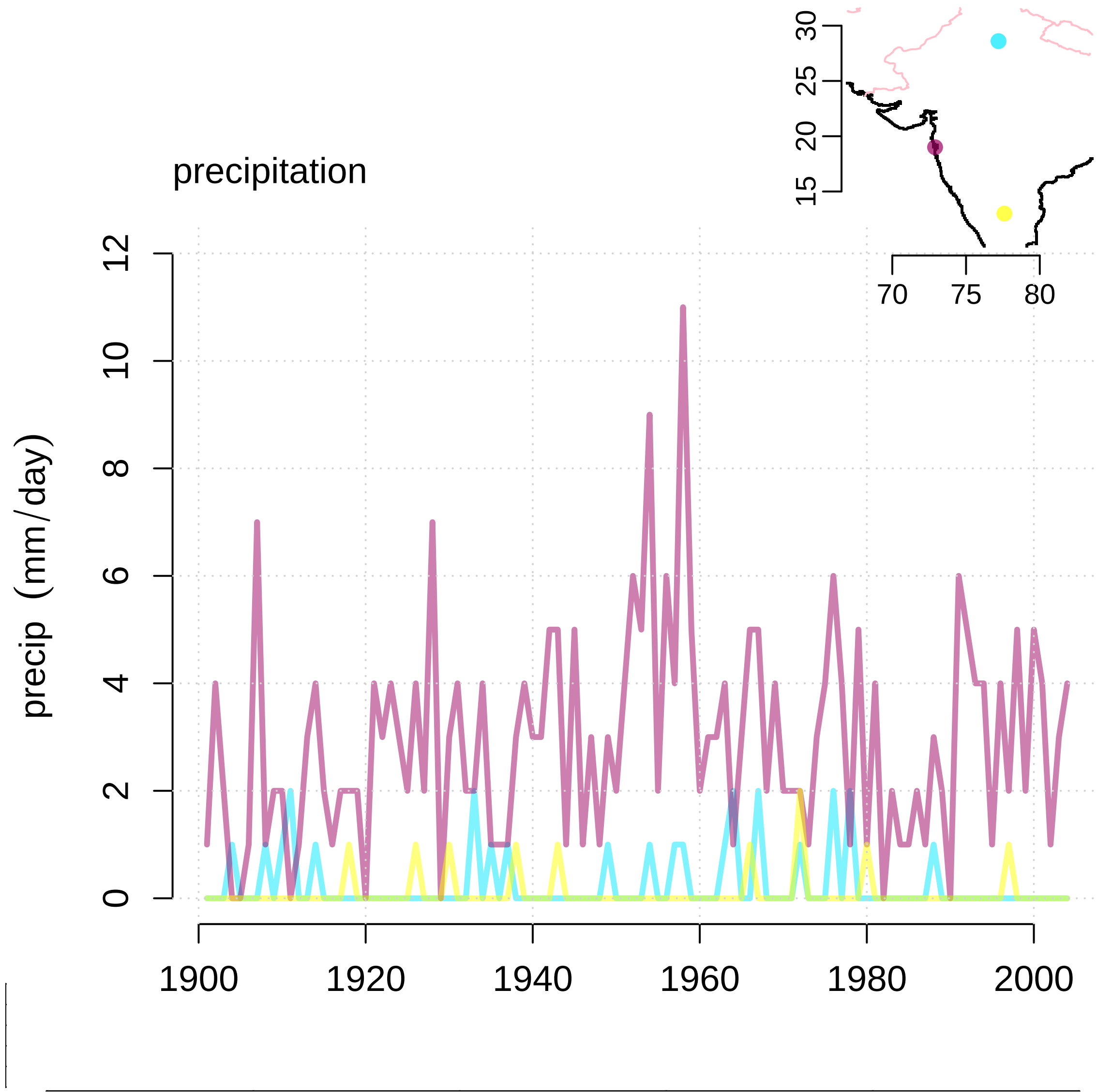

```r
print(trend(pt,result='pval'))
```

```
## [1] 0.11892811 0.00452211 0.65915133
## attr(,"location")
## [1] "Delhi"     "Bombay"    "Bangalore"
## attr(,"longitude")
## [1] 77.2 72.9 77.6
## attr(,"latitude")
## [1] 28.6 19.0 13.0
## attr(,"altitude")
## [1] NA NA NA
## attr(,"cntr")
## [1] "India" "India" "India"
## attr(,"stid")
## [1] NA NA NA
## attr(,"history")
## attr(,"history")$call
```

```
## attr(,"history")$call[[1]]
## trend.station(pt, result = "pval")
##
##
## attr(,"history")$timestamp
## [1] "Wed Aug  3 14:13:46 2016"
##
## attr(,"history")$session
## attr(,"history")$session$R.version
## [1] "R version 3.1.3 (2015-03-09)"
##
## attr(,"history")$session$esd.version
## [1] "esd_1.2"
##
## attr(,"history")$session$platform
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

Also, it's interesting to see how the number of days with heave rainfall changes over time:

```
plot(ndhr,new=FALSE)
grid()
```




```r
print(summary(coredata(pt)))
```

```
##        V1               V2               V3
##  Min.   : 226.0   Min.   : 997.1   Min.   : 341.4
##  1st Qu.: 529.8   1st Qu.:1716.0   1st Qu.: 662.0
##  Median : 665.2   Median :2067.7   Median : 820.8
##  Mean   : 675.1   Mean   :2045.8   Mean   : 800.8
##  3rd Qu.: 799.2   3rd Qu.:2367.5   3rd Qu.: 913.9
##  Max.   :1367.5   Max.   :3733.0   Max.   :1261.4
```

```r
print(summary(coredata(fw*nd)))
```

```
##      Delhi           Bombay         Bangalore
##  Min.   :19.00   Min.   : 73.00   Min.   : 34.00
##  1st Qu.:38.00   1st Qu.: 90.00   1st Qu.: 54.00
##  Median :45.00   Median : 97.00   Median : 60.00
```

```
## Mean   :44.62   Mean   : 97.34   Mean   : 60.31
## 3rd Qu.:51.00   3rd Qu.:105.00   3rd Qu.: 67.00
## Max.   :73.00   Max.   :132.00   Max.   :102.00
```

```
print(summary(coredata(mu)))
```

```
##      Delhi           Bombay        Bangalore
## Min.   : 5.275   Min.   :11.50   Min.   : 5.094
## 1st Qu.:11.327   1st Qu.:17.73   1st Qu.: 9.004
## Median :13.163   Median :20.77   Median :10.384
## Mean   :13.370   Mean   :20.74   Mean   :10.802
## 3rd Qu.:15.086   3rd Qu.:23.04   3rd Qu.:12.389
## Max.   :23.059   Max.   :38.38   Max.   :17.876
```

Predictors for downscaling

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 and error to see what works and what doesn't: the results suggest a reasonable choise was May-October season mean.

```
## Retrieve the reanalysis from FTP
if (!file.exists("air.mon.mean.nc"))
  download.file('ftp://ftp.cdc.noaa.gov/Datasets/ncep.reanalysis.derived/surface/air.mon.mean.nc',
                destfile = 'air.mon.mean.nc')
if (!file.exists("slp.mon.mean.nc"))
  download.file('ftp://ftp.cdc.noaa.gov/Datasets/ncep.reanalysis.derived/surface/slp.mon.mean.nc',
                destfile = 'slp.mon.mean.nc')
```

```
## 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 for the May-October season.

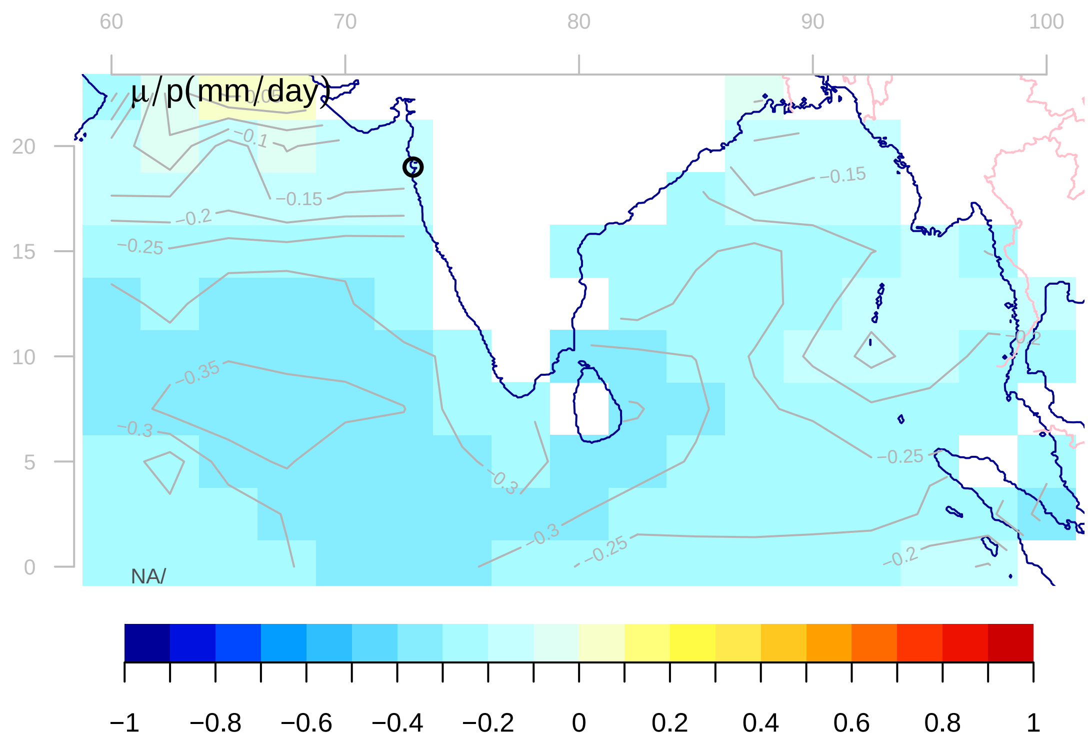
```
if (!exists("slp")) {
  slp <- retrieve('slp.mon.mean.nc',lon=c(60,100),lat=c(0,23))
  slp <- aggregate(subset(slp,it=month.abb[season]),year,'mean')
}
```

```
## [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
```

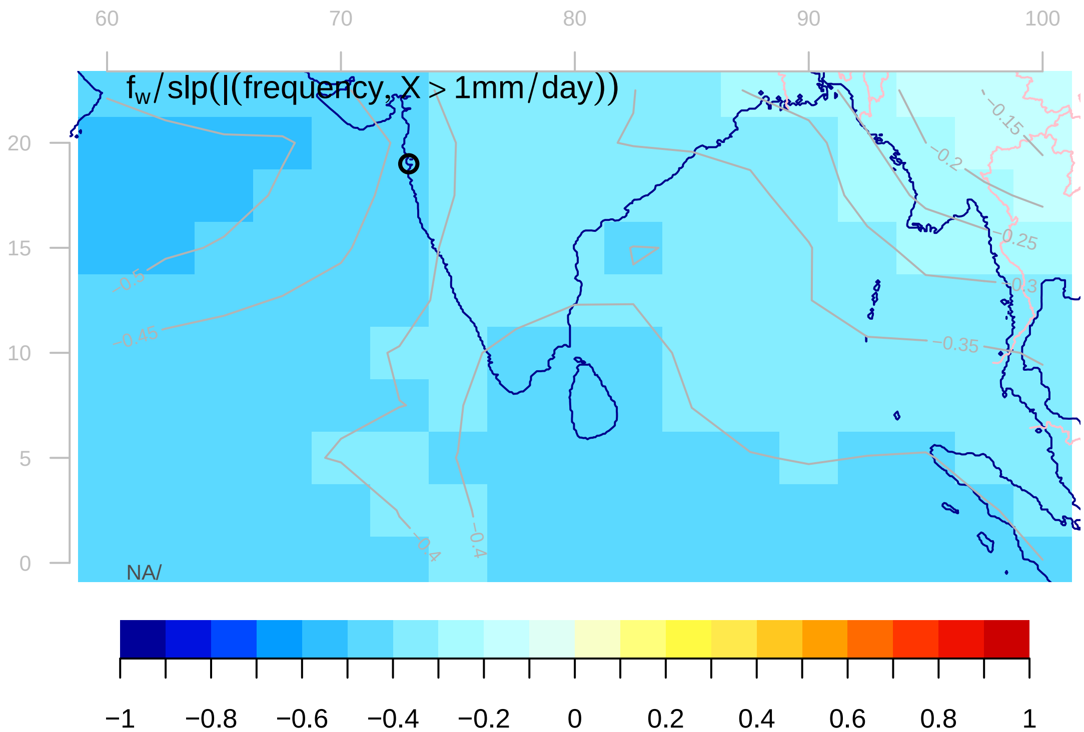
Examine potential large-scale tele-connections: correlation fields

```
## 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)),new=FALSE)
```

```
## 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)),new=FALSE)
```

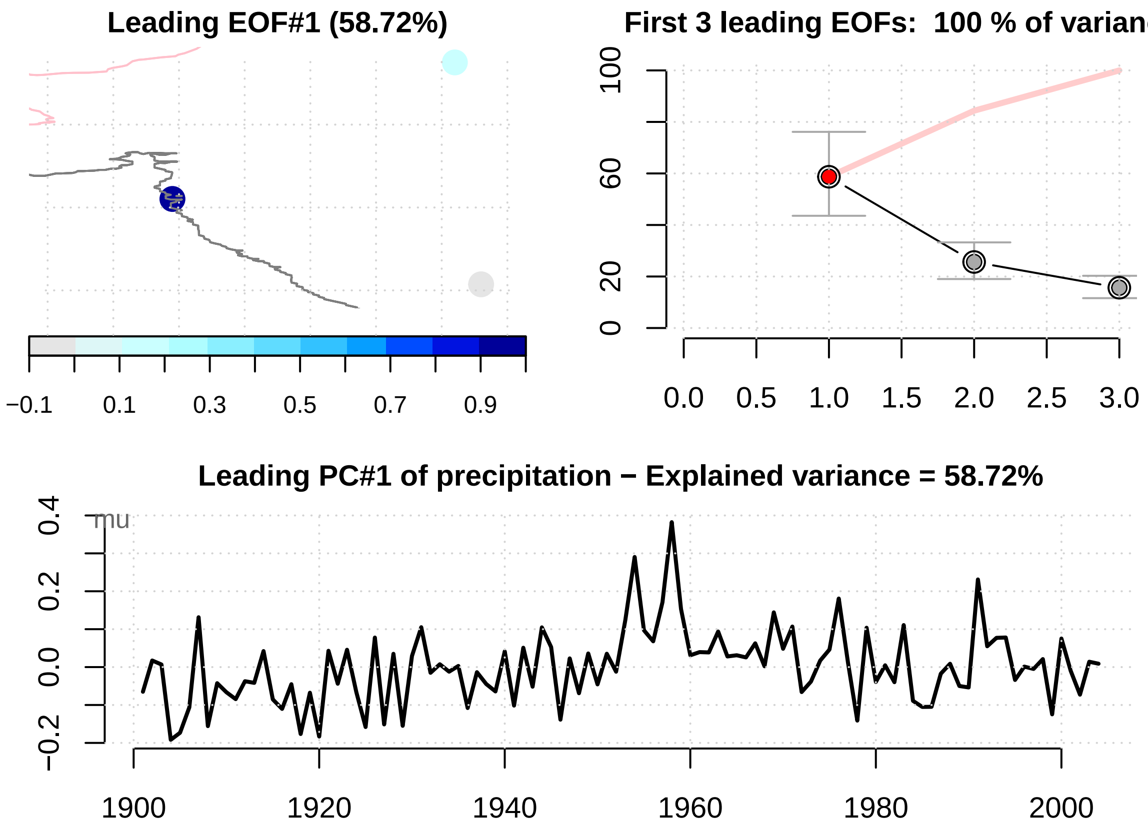


Principal component analysis (PCA) can improve the quality of empirical-statistical downscaling (ESD) by reorganising the data so that the large-scale variability is enhanced (http://dx.doi.org/10.3402/tellusa.v67.28326).

Estimate PCA for the predicands

```
pca.mu <- PCA(mu)
pca.fw <-PCA(fw)
plot(pca.mu,new=FALSE)
```

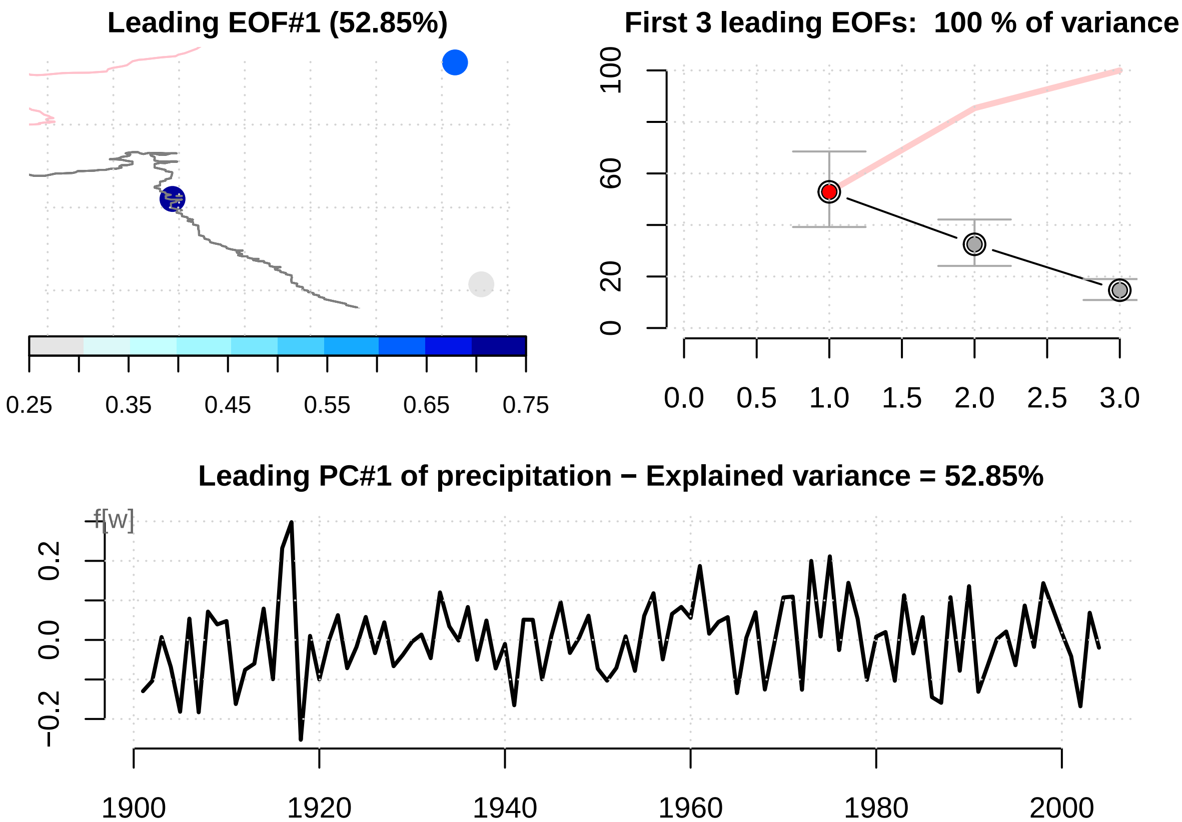
```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "plot" is not a
## graphical parameter
```



The leading mode for the seasonal wet-day mean precipitation is dominated by Mumbay with almost zero covariance with Bangalore.

```
plot(pca.fw,new=FALSE)
```

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



The PCA pattern has more similar spatial weights for f_w over the three megacities, even though there is a more equal share of variance between the modes.

Estimate EOFs for the predictors

```
eof.es <- EOF(es)
eof.slp <- EOF(slp)
eof.t2m <- EOF(aggregate(subset(t2m,is=list(lon=c(65,80),lat=c(10,20))),year,'mean'))
```

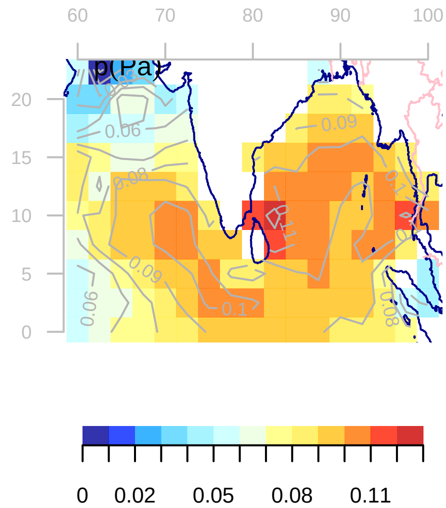
```
## Warning in if (deparse(substitute(by)) == "year") {: the condition has
## length > 1 and only the first element will be used
```

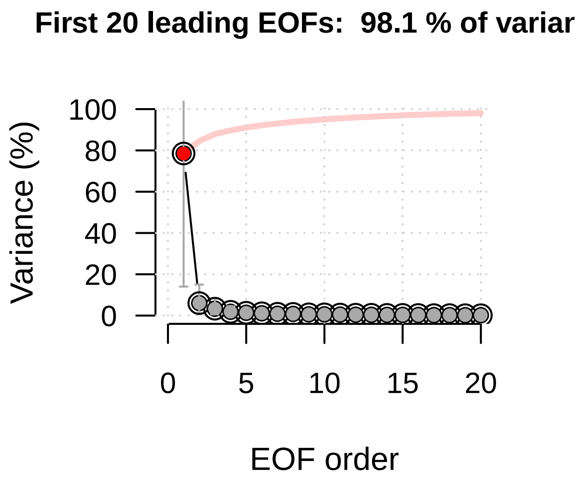
```
index(eof.es) <- year(eof.es)
index(eof.slp) <- year(eof.slp)
index(eof.t2m) <- year(eof.t2m)
```

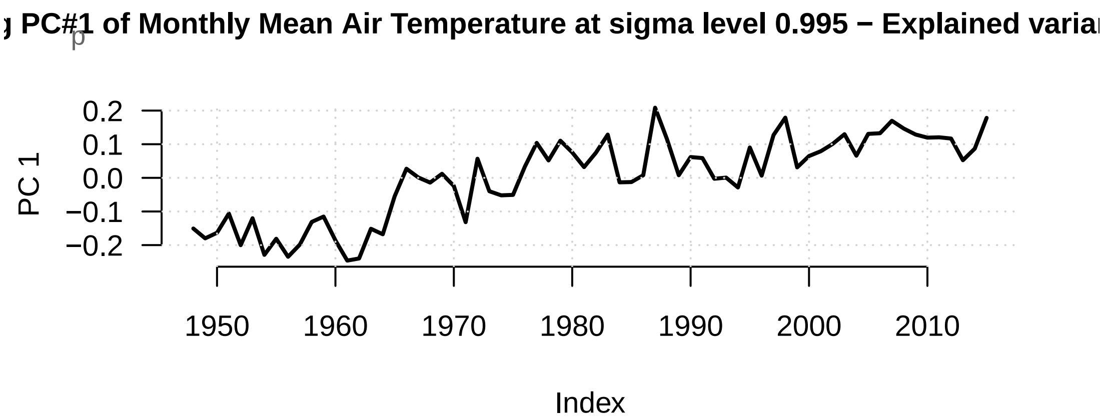
The dominant mode of variability in the May-October mean e_s featured largest magnitudes east of south Indian coast.

Plot the EOFs for the temperature-based predictor

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



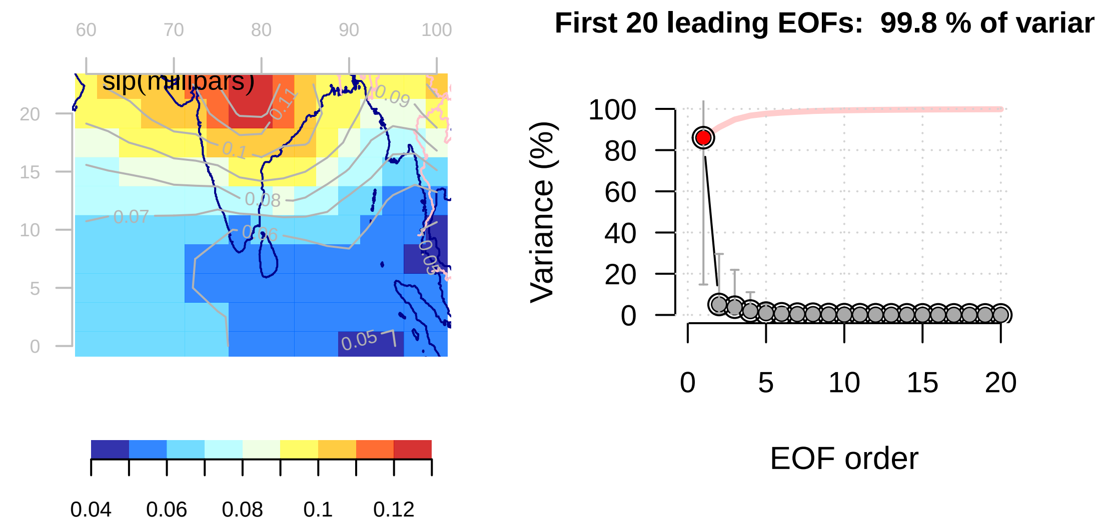


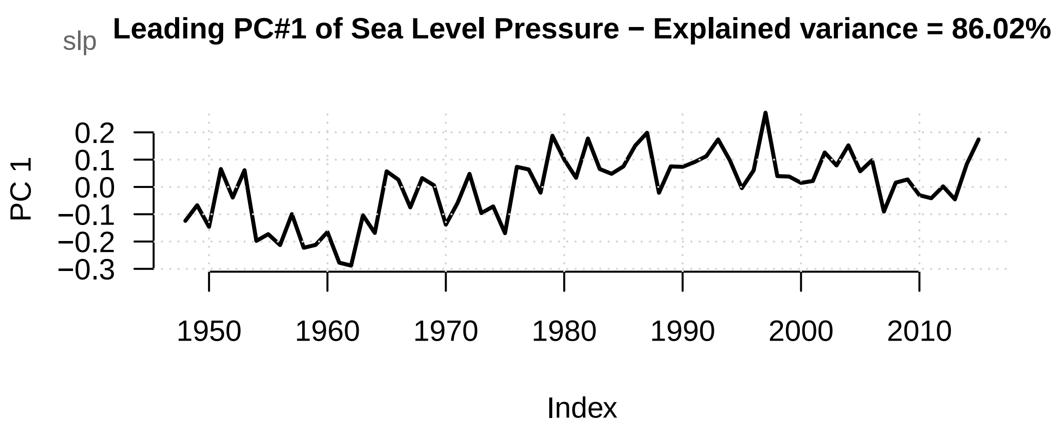


The main May-October SLP mode was associated with the strength of the Monsoon winds.

Plot the EOFs for the SLP

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

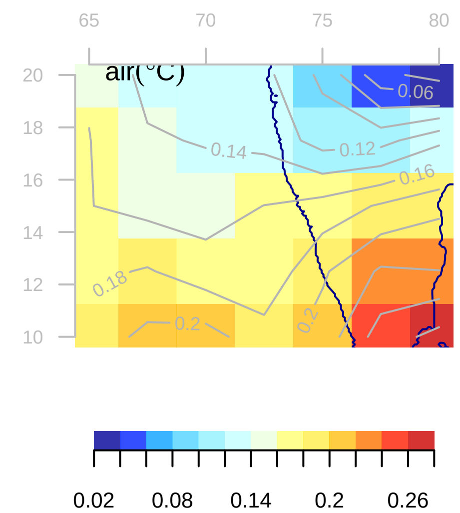


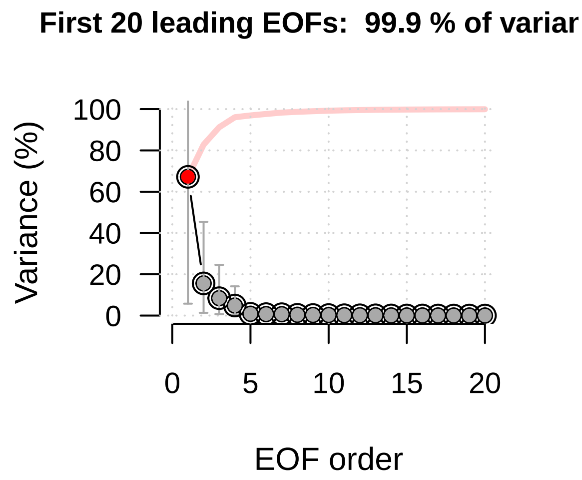


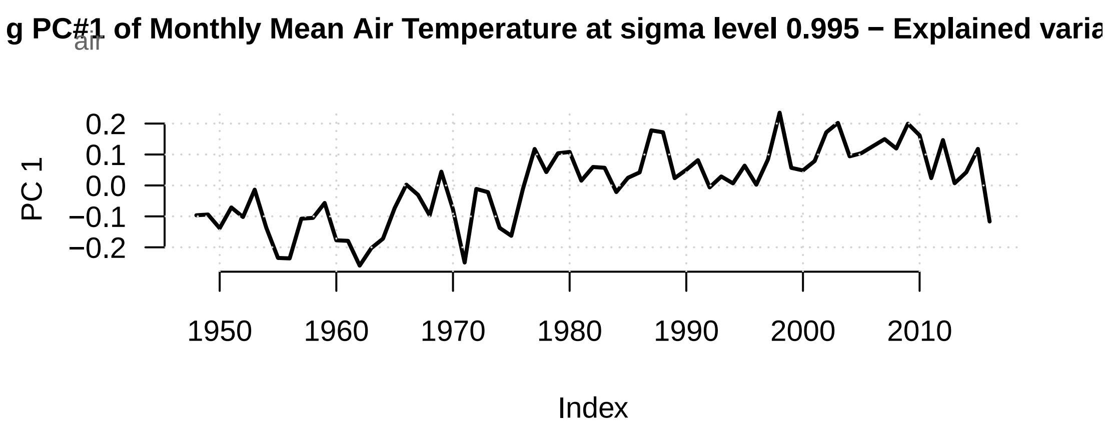
The main May-October T(2m) mode was associated with the strength of the Monsoon winds.

Plot the EOFs for the SLP

```
plot(eof.t2m,new=FALSE)
```







Do the downscaling

The wet-day mean precipitation

The predictor for the wet-day mean precipitation was taken to be the saturation vapour pressure e_s estimated from maritime surface air taken from the NCEP/NCAR reanalysis 1. This choice was motivated by physics, expecting the intensity to be influenced by the amount of moisture in the air - which again has ita main source from the evaopration over the oceans.

```
## The function DS uses multiple regression by default
ds.mu <- DS(pca.mu,eof.es,eofs=1:20)
```

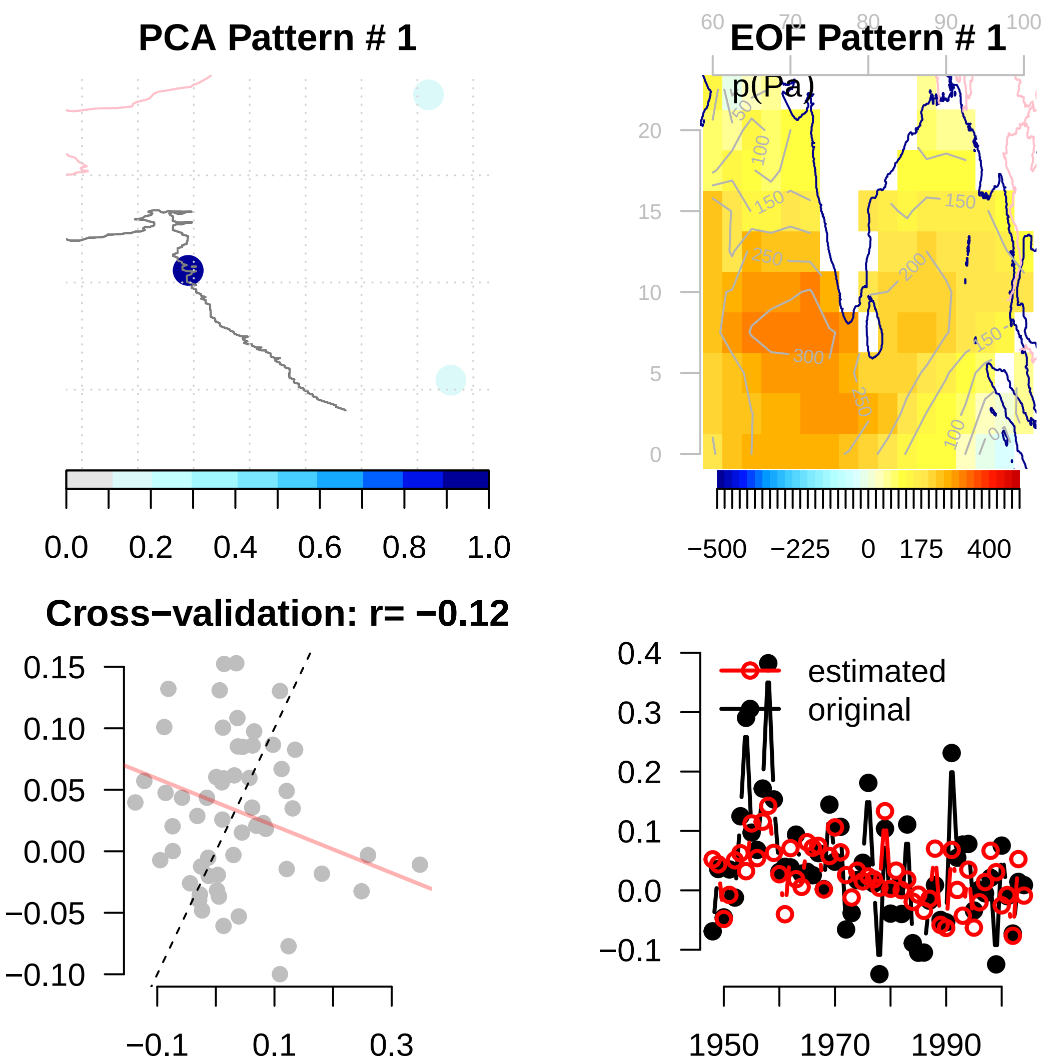
```
##
  |
  |                                                                  |   0%
  |
  |======================                                            |  33%
```

```
## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric
```

```
##
  |
  |===========================================                      |  67%

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric

##
  |
  |=================================================================| 100%

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric
```

```
plot(ds.mu,colbar1=list(breaks=seq(0,1,by=0.1)),
           colbar2=list(breaks=seq(-500,500,by=25)),new=FALSE)
```

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



```
## NULL
```

The results for the wet-daty mean μ were problematic in the sense that the evaluation of the downscaling model (cross-validation) revealed low scores, in particular associated with a small number of extreme seasons with exceptionally high μ.

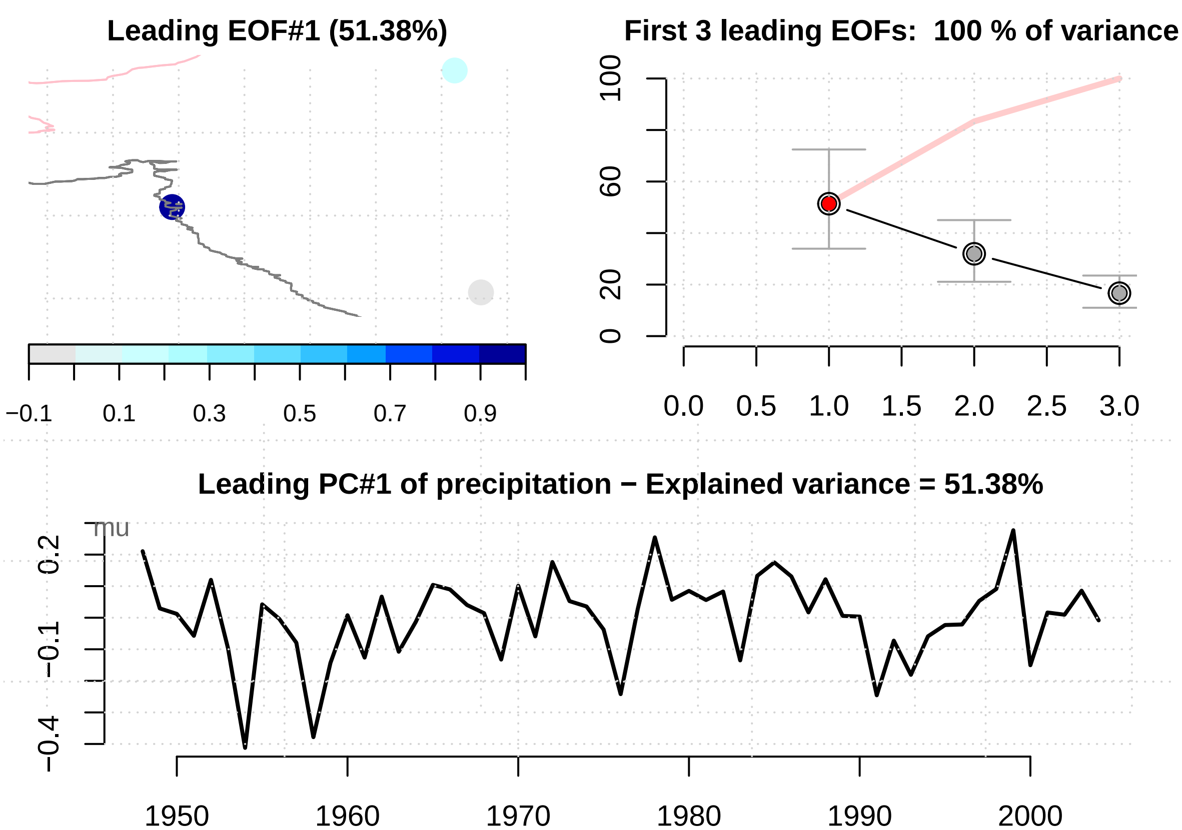
Examine the residuals from the downscaling for the wet-day mean

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

```
## Examine the PCA of residuals for the cold months to see if there are any structures left
  mu.2 <- as.station(as.residual(ds.mu))
  pca.mu.2 <- PCA(mu.2)
  plot(pca.mu.2,new=FALSE)
```

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

```
  grid()
```




```
  str(mu.2)
```

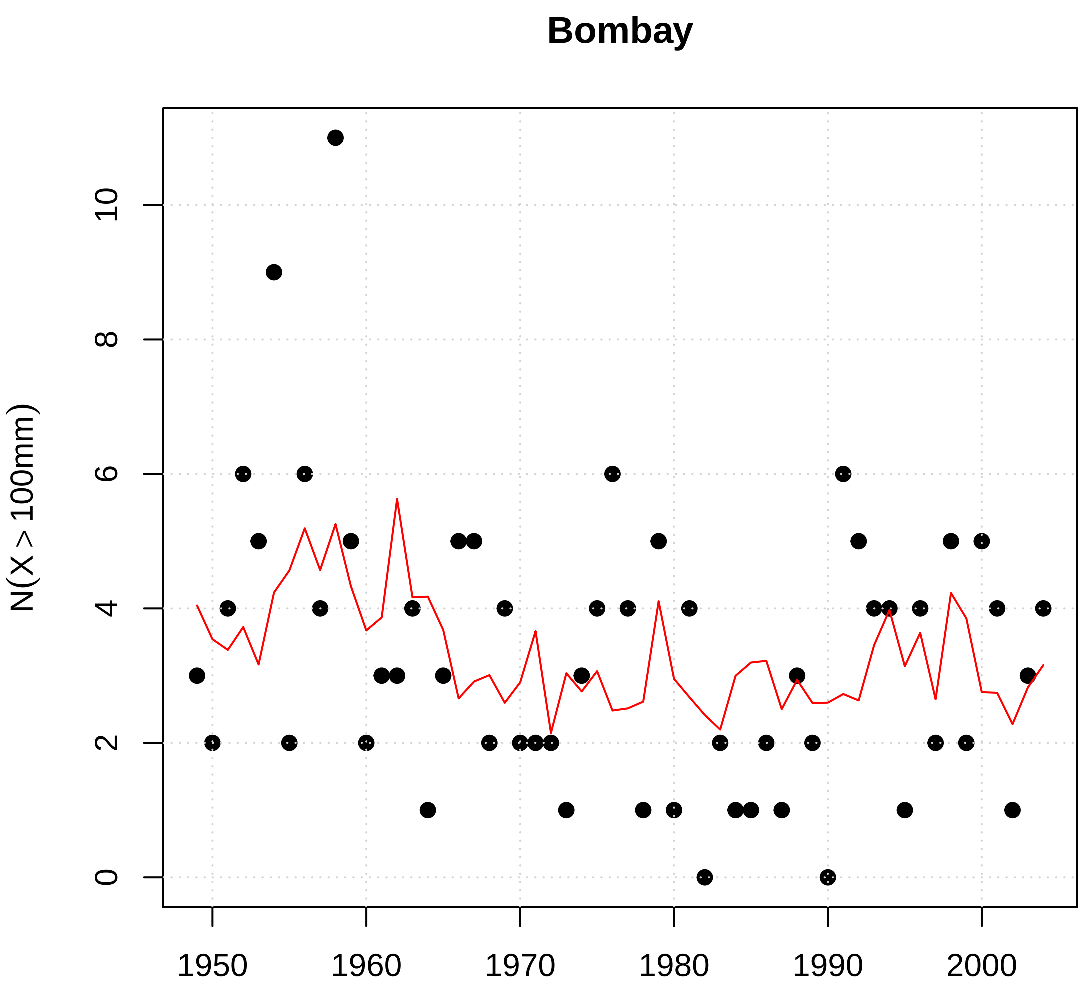
```
## 'zoo' series from 1948-01-01 to 2004-01-01
##   Data: num [1:57, 1:3] 4.484 0.672 0.398 1.663 0.992 ...
##  - attr(*, "dimnames")=List of 2
```

```
##    ..$ : chr [1:57] "1948" "1949" "1950" "1951" ...
##    ..$ : chr [1:3] "Delhi" "Bombay" "Bangalore"
##  - attr(*, "location")= chr [1:3] "Delhi" "Bombay" "Bangalore"
##  - attr(*, "variable")= chr "mu"
##  - attr(*, "unit")= chr "mm/day"
##  - 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(*, "country")= chr [1:3] "India" "India" "India"
##  - attr(*, "longname")= chr [1:3] "precipitation" "precipitation" "precipitation"
##  - attr(*, "station_id")= logi [1:3] NA NA NA
##  - attr(*, "calendar")= chr "gregorian"
##  - attr(*, "source")= logi [1:3] NA NA NA
##  - attr(*, "URL")= logi [1:3] NA NA NA
##  - attr(*, "type")= logi NA
##  - attr(*, "aspect")= chr "residual"
##  - 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
##  - attr(*, "method")= logi NA
##  - attr(*, "history")=List of 3
##    ..$ call     :List of 1
##    .. ..$ : language as.station.zoo(as.residual(ds.mu))
##    ..$ timestamp: chr "Wed Aug  3 14:14:25 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)"
##    Index:  Date[1:57], format: "1948-01-01" "1949-01-01" "1950-01-01" "1951-01-01" ...
```

Alternative approach: downscale the number of days with heavy rainfall

Model the number with heavy rainfall (more than 100 mm/day) in Mumbay.

```r
ndhr.cal <- data.frame(y = coredata(subset(subset(ndhr,is=2),it=c(1949,2004))),
                       X = coredata(subset(eof.es,it=c(1949,2004))))
ds.ndhr <- glm(y ~ X.1 + X.2 + X.3 + X.4 + X.5 + X.6 + X.7, data=ndhr.cal,family="poisson")
plot(1949:2004,ndhr.cal$y,pch=19,ylab=expression(N(X > 100*mm)),xlab='',main=loc(ndhr)[2])
grid()
lines(1949:2004,exp(predict(ds.ndhr)),col='red')
```

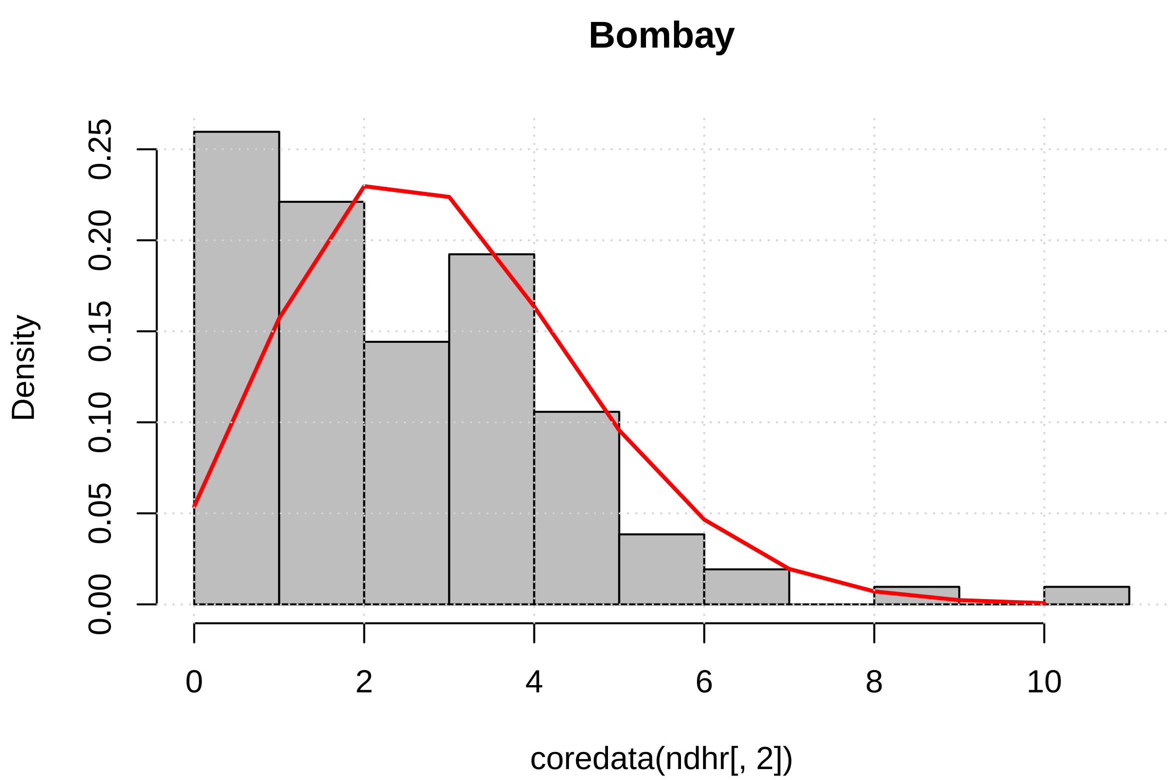


Check if the number of days with heavy precipitation is Poisson-distributed

```
print(paste('Mean number=',mean(ndhr[,2],na.rm=TRUE),'Variance=',var(ndhr[,2],na.rm=TRUE),
            'should equal',mean(ndhr[,2],na.rm=TRUE),'for Poisson process'))
```

```
## [1] "Mean number= 2.92307692307692 Variance= 3.85810306198656 should equal 2.92307692307692 for Poiss
```

```
h <- hist(coredata(ndhr[,2]),freq=FALSE,col='grey',
     main=loc(ndhr)[2])
lines(h$mids-0.5,dpois(x=h$mids-0.5,lambda=mean(ndhr[,2],na.rm=TRUE)),lwd=2,col='red')
grid()
```



The wet-day frequency

Mean sea level pressure was used as a predictor to downscale the wet-day frequency f_w, which too is motivated from physical reasoning: the wind speed/direction and transport of moisture influence the probability for precipitation.

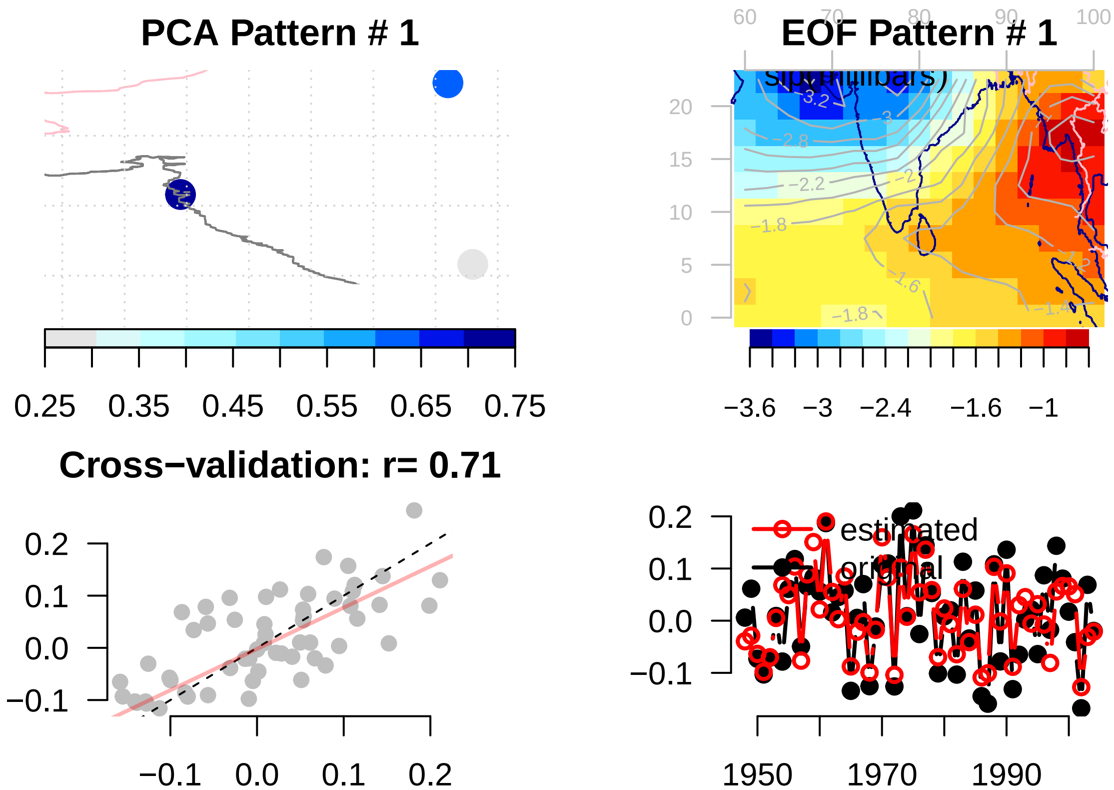
```
ds.fw <- DS(pca.fw,eof.slp,eofs=1:20)
```

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

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric

##
  |
  |===========================================                      |  67%

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric

##
  |
  |=================================================================| 100%
```

```
## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric
```

```
plot(ds.fw,new=FALSE)
```

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




```
## NULL
```

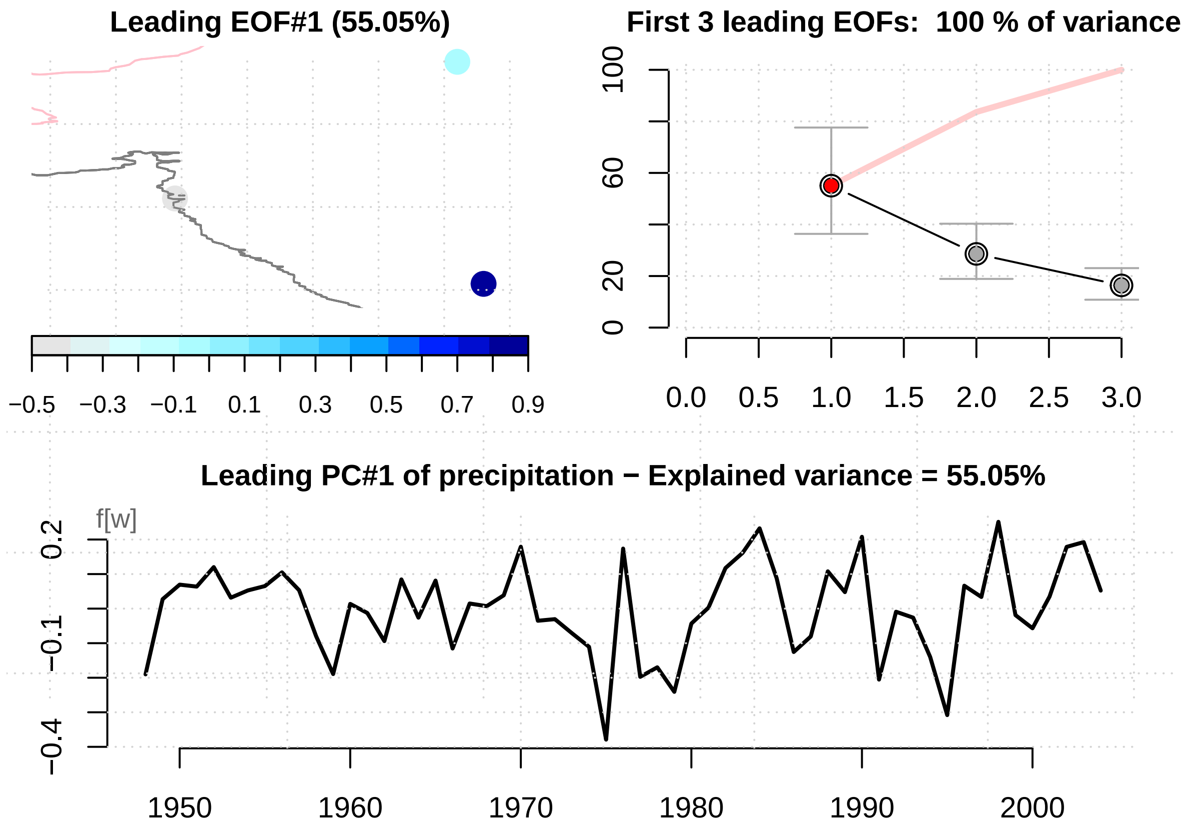
The downscaled results for f_w were associated with high skill (cross-validation = 0.71), and the associated predictor pattern implies that the rain occurrence is associated with large-scale southwesterly wind field, typically associated with the monsoon.

The residuals for the wet-day frequency:

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

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

```
grid()
```



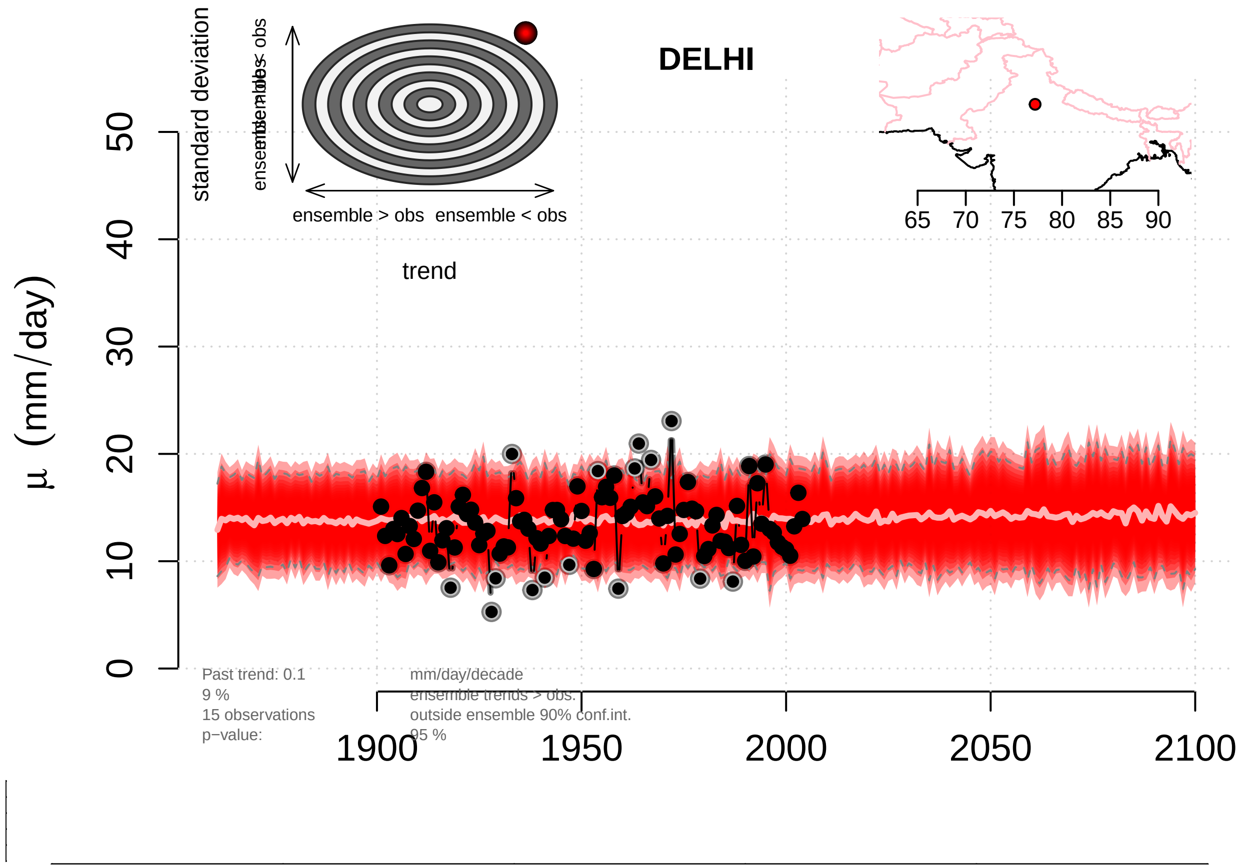
Do the downscaling based on the CMIP5 RCP4.5 ensemble

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, we apply the downscaling to the mean sea level pressure (MSLP).

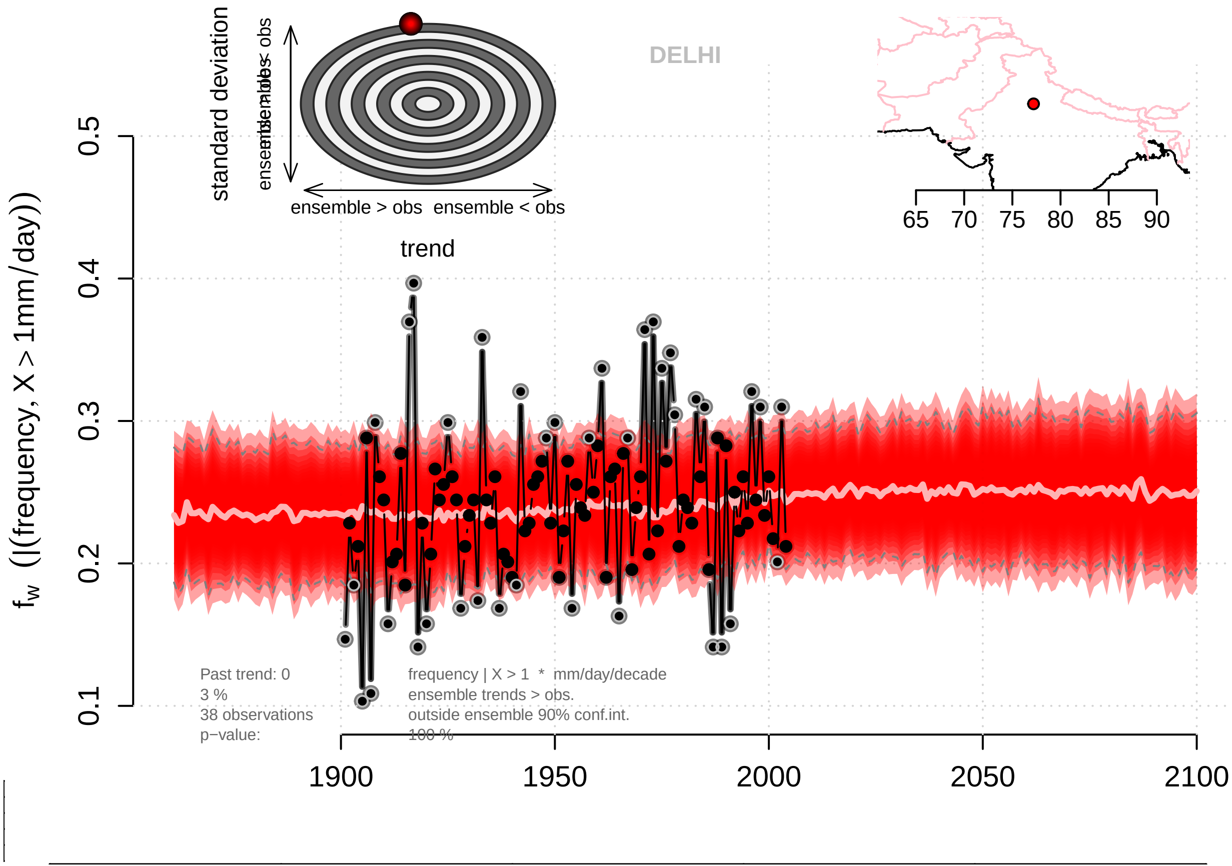
```
if (!file.exists('dse.mu.climatrans.rda')) {
  index(pca.mu) <- as.Date(paste(year(pca.mu),'01-01',sep='-'))
  index(pca.fw) <- as.Date(paste(year(pca.fw),'01-01',sep='-'))
  dse.mu <- DSensemble(pca.mu,predictor=es,FUNX='tas2es',xfuns='tas2es',plot=TRUE)
  dse.fw <- DSensemble(pca.fw,predictor=slp,pattern="psl_Amon_ens_",plot=TRUE)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

```
## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length
```

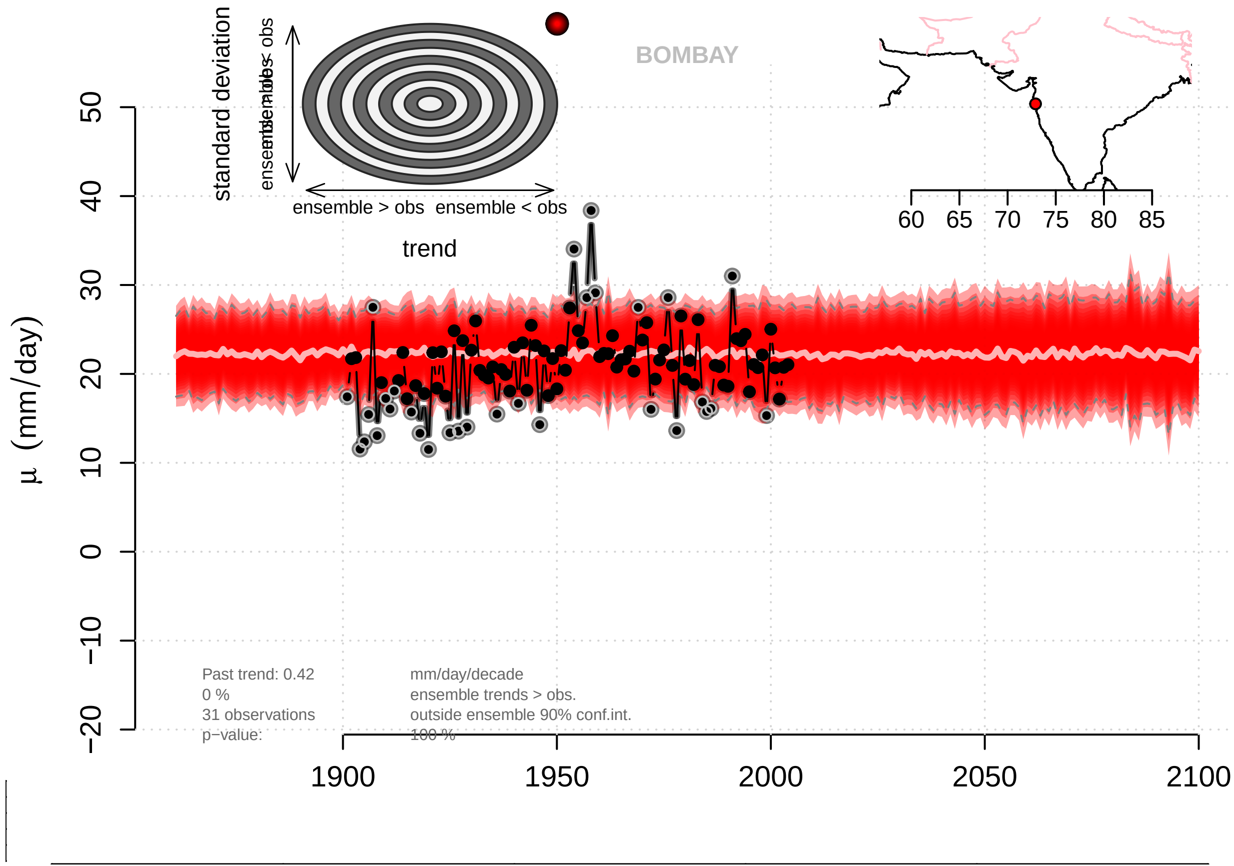
```
## Warning in `<.default`(yz[, 1], q05): longer object length is not a
## multiple of shorter object length

## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length

## Warning in `<.default`(yz[, 1], q05): longer object length is not a
## multiple of shorter object length
```



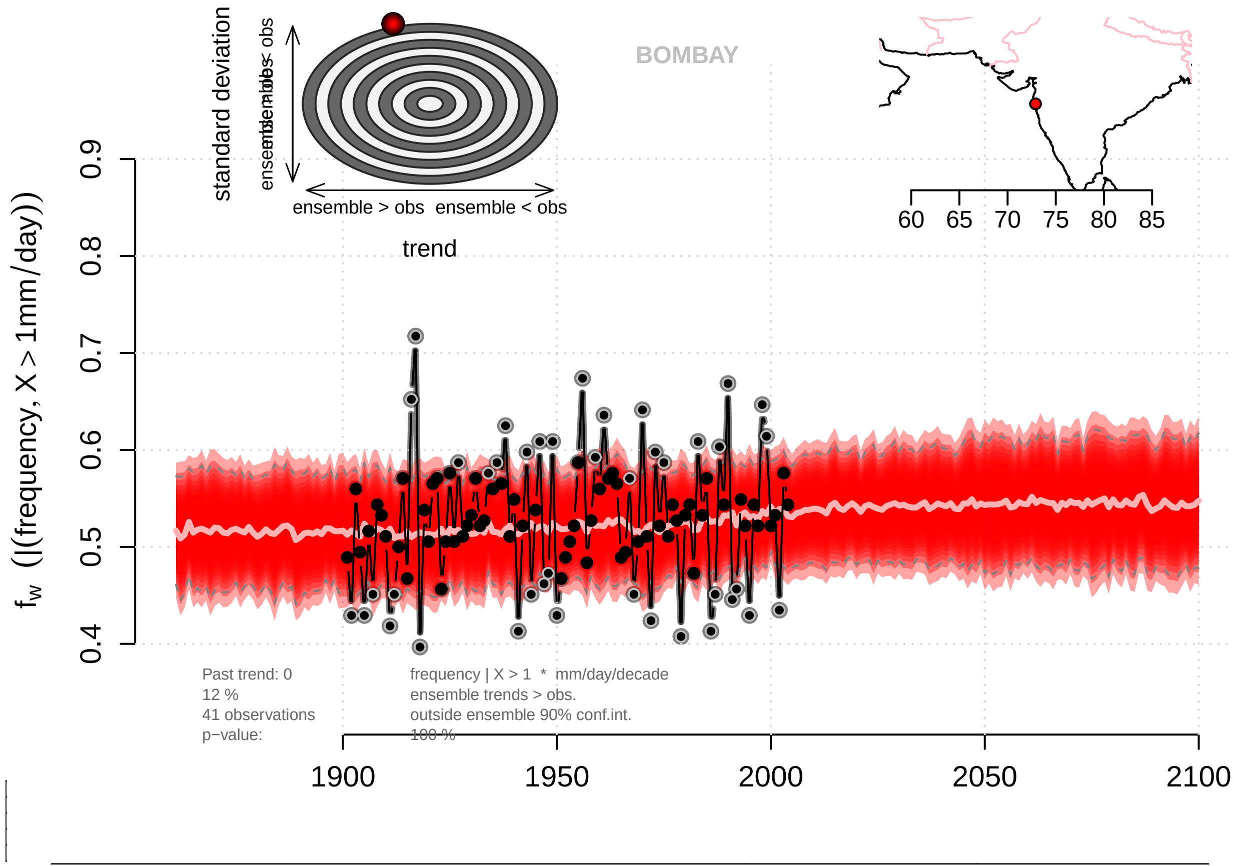

```
## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length

## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length
```



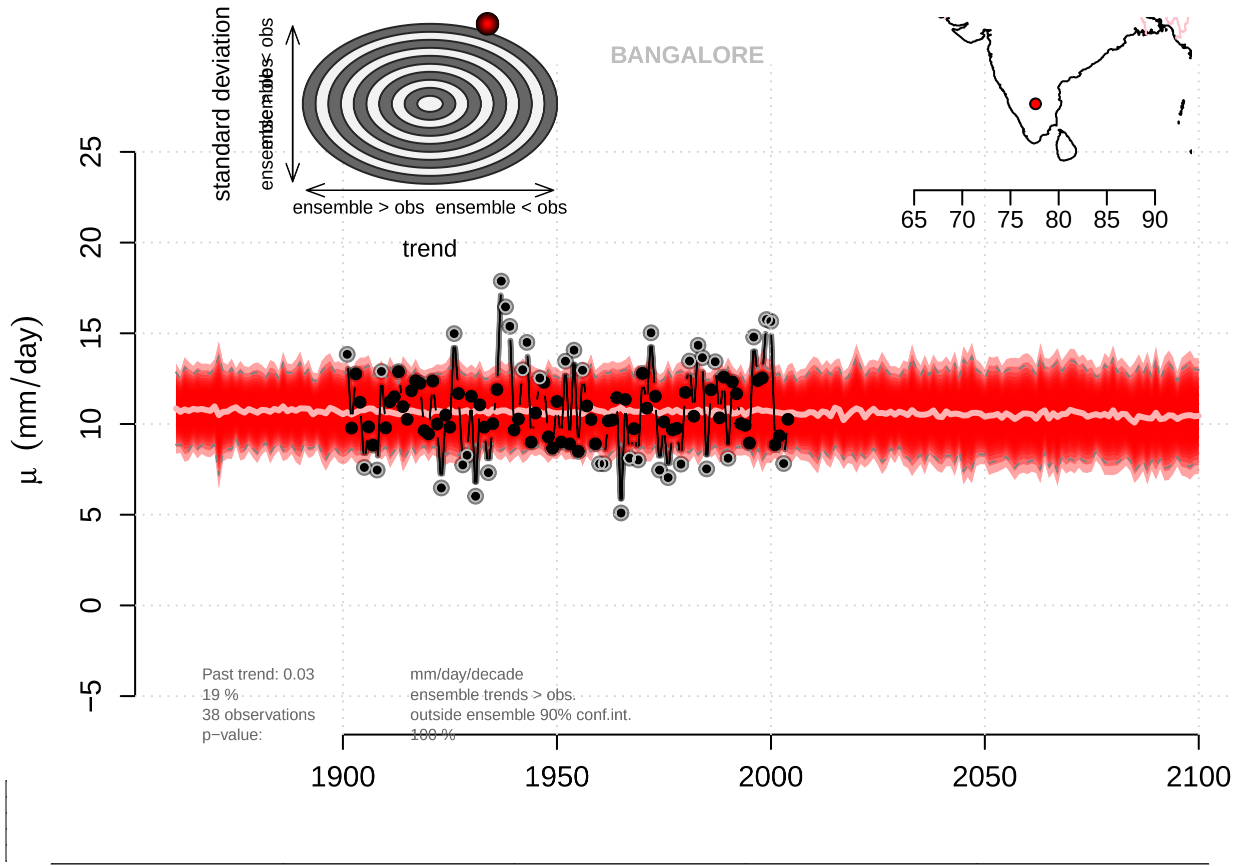

```
## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length

## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length
```

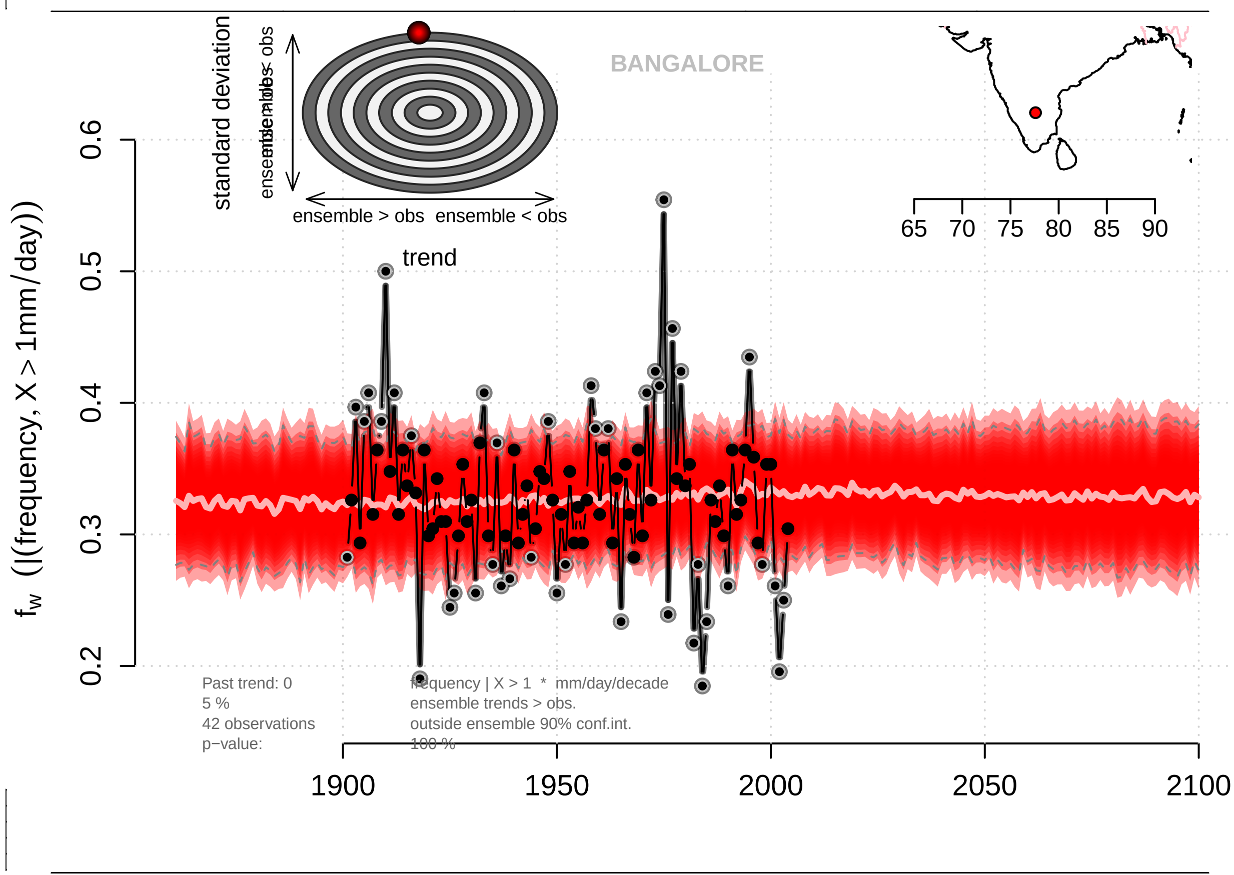



```
## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length

## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length
```




```
## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length

## Warning in `>.default`(yz[, 1], q95): longer object length is not a
## multiple of shorter object length
```


standard deviation
ensemble < obs
ensemble > obs
ensemble > obs ensemble < obs
BANGALORE
65 70 75 80 85 90
trend
μ (mm/day)
25
20
15
10
5
0
−5
Past trend: 0.03 mm/day/decade
19 % ensemble trends > obs.
38 observations outside ensemble 90% conf.int.
p−value: 100 %
1900 1950 2000 2050 2100


standard deviation
ensemble < obs
ensemble > obs
ensemble > obs ensemble < obs
BANGALORE
65 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 frequency | X > 1 * mm/day/decade
5 % ensemble trends > obs.
42 observations outside ensemble 90% conf.int.
p−value: 100 %
1900 1950 2000 2050 2100

Delhi Validation of the downscaled ensembles underestimated both the observed interannual variations and trend in the May-October mean μ. The projections do not indicate much change in the future, however, the analysis does not capture well all factors that may influence the precipitation intencity. It may be more sensitive to small-scale conditions and processes (e.g. mesoscale convection).

The donscaled ensemble provides an inadequate description of f_w for Delhi, although both models and observations agree on near-zero trends. There are uncounted sources for interannual variability, which implies and underestimation of dry and wet seasons.

Mumbay The downscaled results provide an inadequate description of both μ and f_w in terms of interannual variability and trend (μ). The results do not suggest future change in μ, however, there are some indication of a slight increase in f_w.

Bangalore The downscaled results provide an inadequate description of both μ and f_w in terms of interannual variability and trend, and suggest future no change in μ or f_w.

GCM raw results for precipitation over the Indian sub-continent:

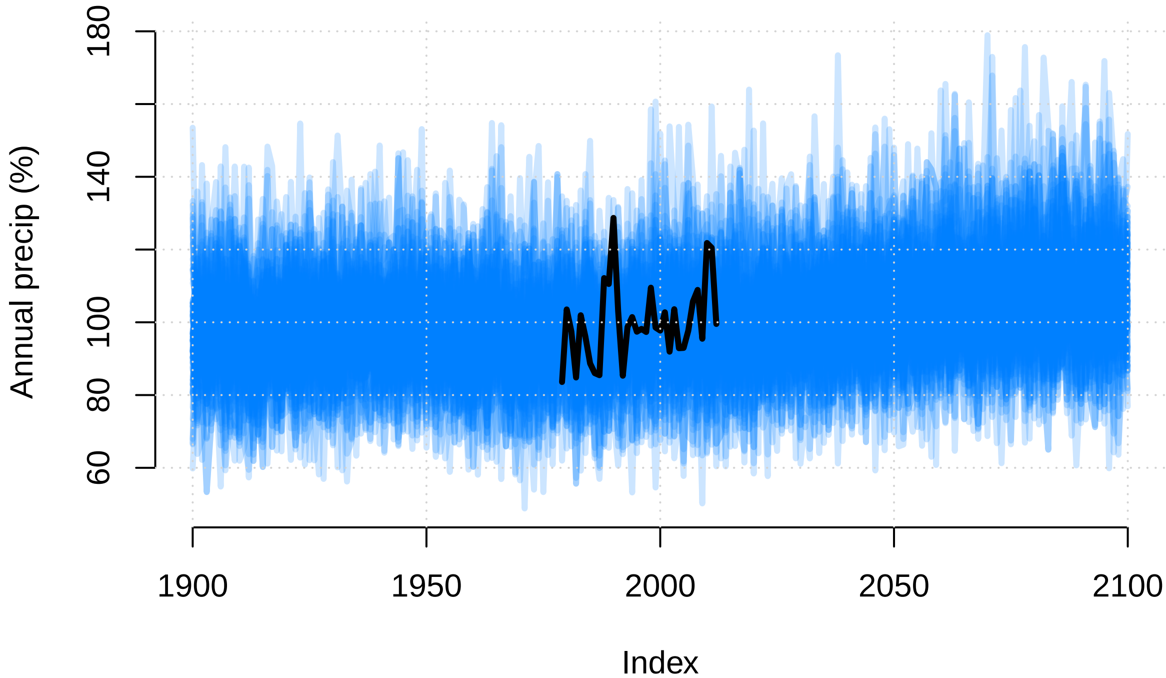
As an extra check, a plot is made of the raw CMIP5 RCP4.5 GCM results for the precipitation averaged over the Indian continent (70-90E/7-30N). Here, the annual mean precipitation $\overline{x}$ is shown rather than May-October f_w and μ, however, $\overline{x} = f_w \mu$ and most of the rainfall typically falls within the May-October period. Hence, any change in the annual mean precipitation is mostly due to the rainy monsoon season.

The results suggest a modest increase simulated for precipitation. The corresponding estimate from the NCAR/NCEP reanalyses suggests an increasing trend, however, such results should be viewed with caution doe to potential inhomogeneities (https://climatedataguide.ucar.edu/climate-data/atmospheric-reanalysis-overview-comparison-tables)

```
## India:
Z2 <- precipcmip(lon=c(70,90),lat=c(7,30))
```

```
## [1] "precipcmip: results (will be) stored in precipcmip.70.7-90.30.rda"
```

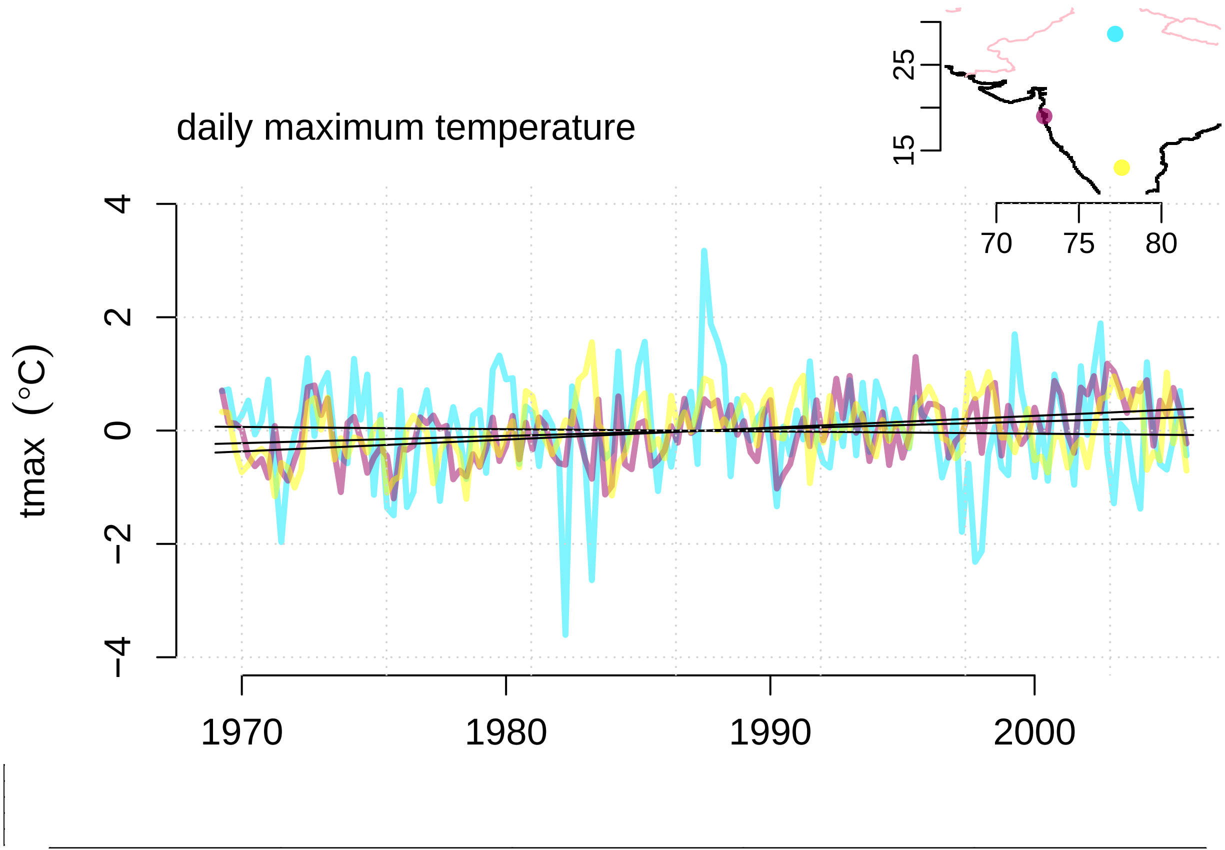
```
par(bty='n')
X <- zoo(x=apply(coredata(Z2$gcm),2,function(x) 100*(x/mean(x,na.rm=TRUE))),order.by=index(Z2$gcm))
plot(X,lwd=3,col=rgb(0,0.5,1,0.2),plot.type='single',ylab='Annual precip (%)')
lines(100*Z2$obs/mean(Z2$obs,na.rm=TRUE),lwd=3)
grid()
```



* Analysis of 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)),new=FALSE)
for(i in 1:3) lines(trend(subset(as.4seasons(anomaly(climatrans.tx)),is=i)))
grid()
```




```
print(summary(coredata(climatrans.tx)))
```

```
##       Delhi             Bombay          Bangalore
##  Min.    :11.64   Min.    :23.85   Min.    :20.91
##  1st Qu.:26.29   1st Qu.:29.00   1st Qu.:28.83
##  Median :32.75   Median :30.85   Median :30.38
##  Mean    :31.57   Mean    :30.94   Mean    :30.89
##  3rd Qu.:36.14   3rd Qu.:32.91   3rd Qu.:32.95
##  Max.    :46.96   Max.    :38.11   Max.    :38.87
```

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)
```

Organise the predictand

```
## 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)
```

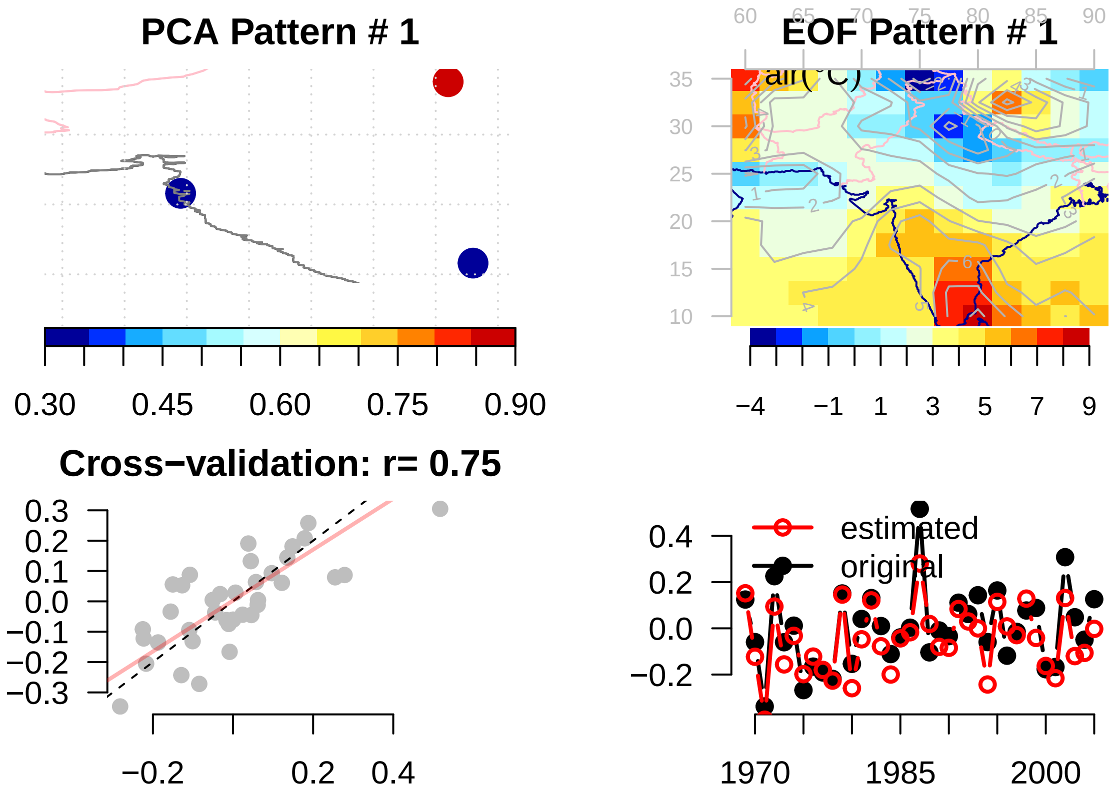
Apply the downscaling to maximum temperature and check the residuals.

```
  ds.tx <- DS(pca.tx,eof.t2m)
```

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

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric

##
  |
  |============================================                      |  67%

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric

##
  |
  |==================================================================| 100%

## Warning in DS.station(ys, X, biascorrect = biascorrect, m = m, eofs =
## eofs, : DS.station: different indices: Date numeric
```

```
  plot(ds.tx,new=FALSE)
```

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

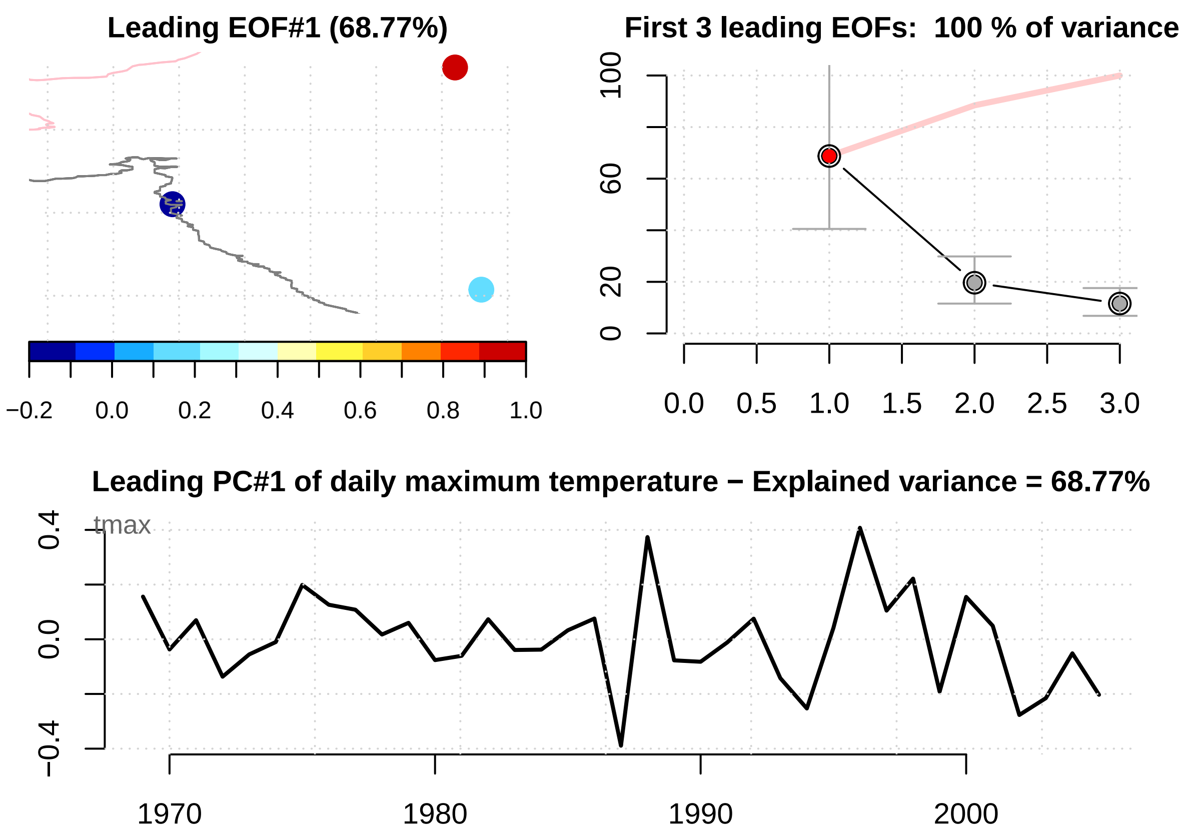



```
## NULL
```

Check the residuals of the regression analysis

```
## Check the residuals
  tx.2 <- as.station(as.residual(ds.tx))
  pca.tx.2 <- PCA(tx.2)
  plot(pca.tx.2,new=FALSE)
```

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

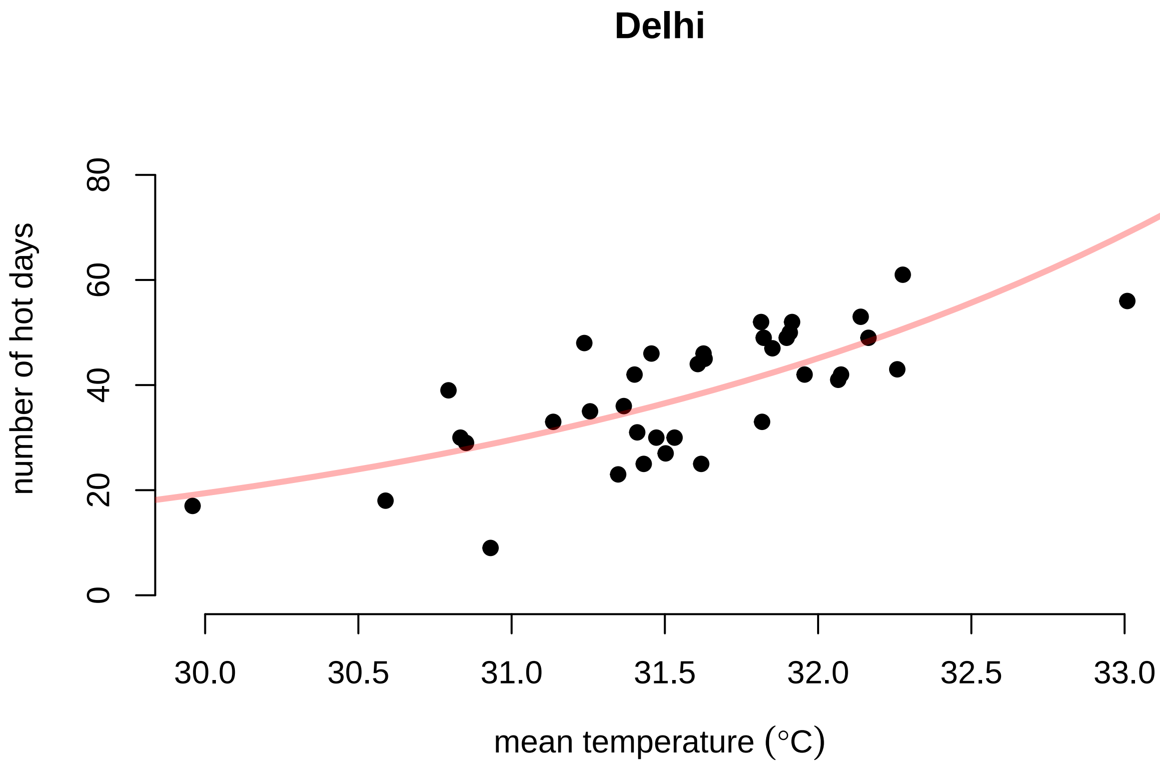


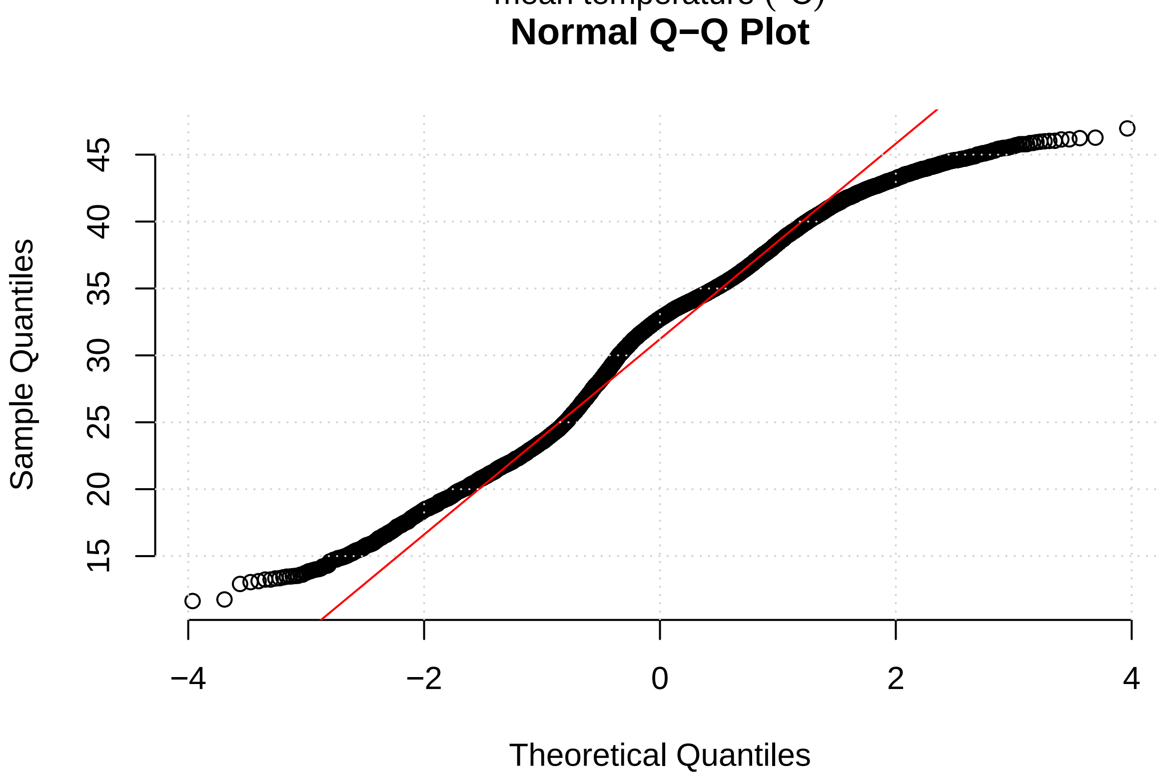
The eigenvalue spectrum is not flat, and the time series hints of some temporal structure, however, the spatial weights suggests little spatial covariance.

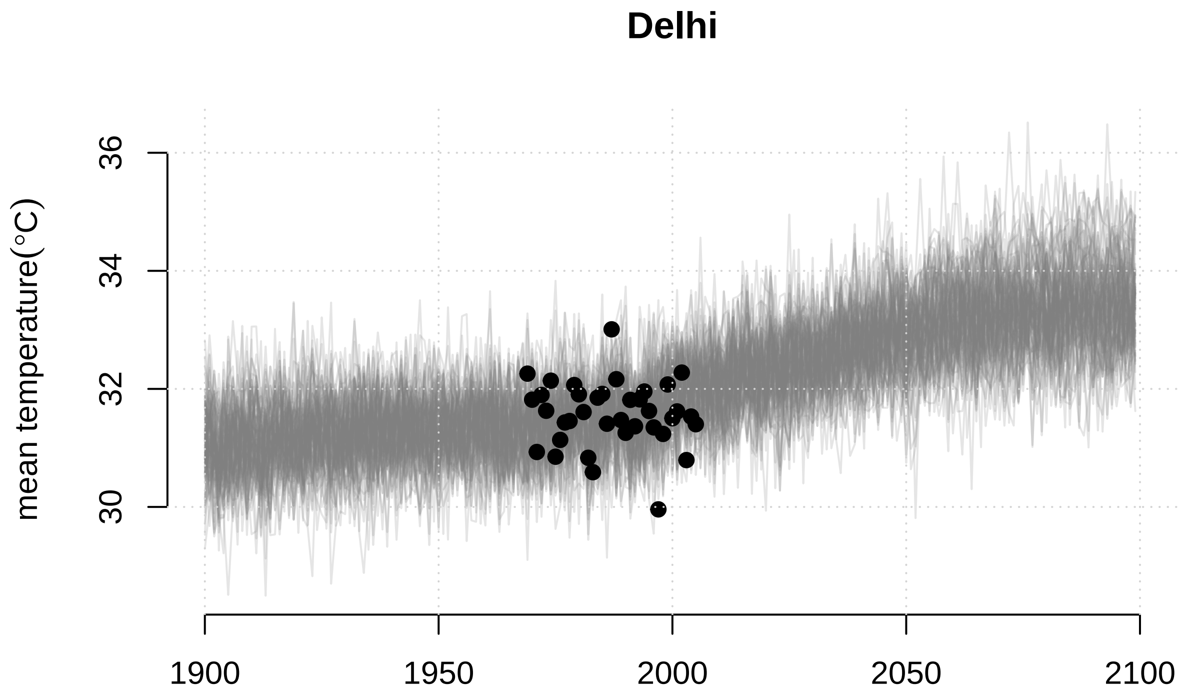
Hot day statistics

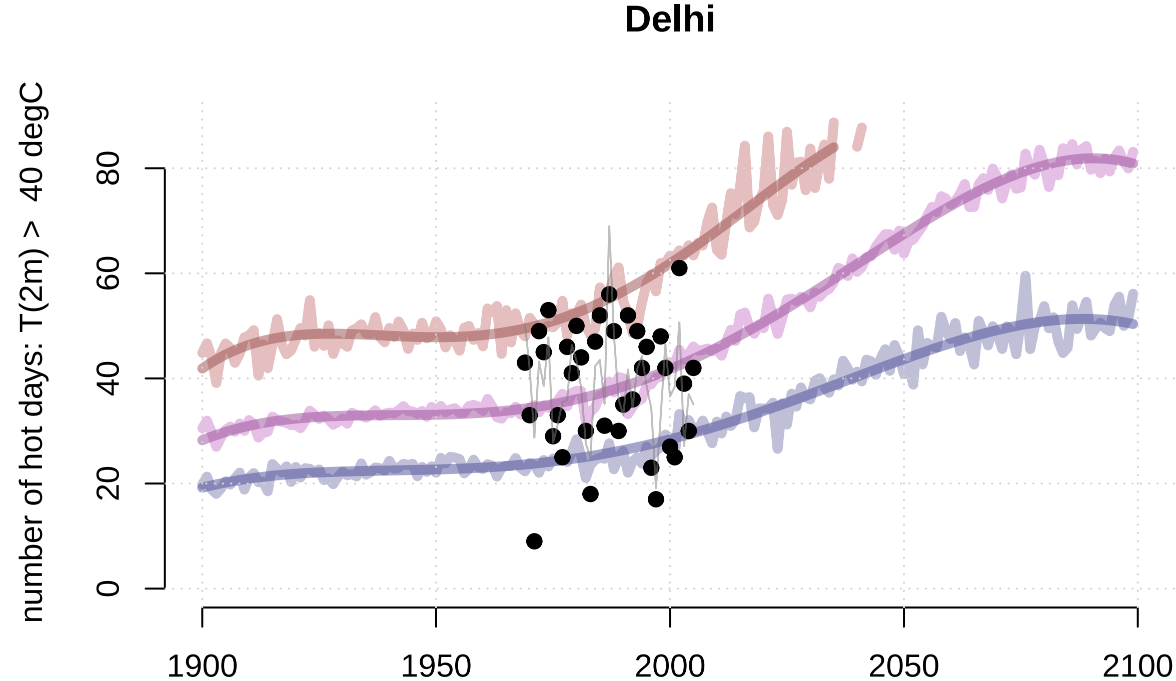
Apply the analysis of number of hot days

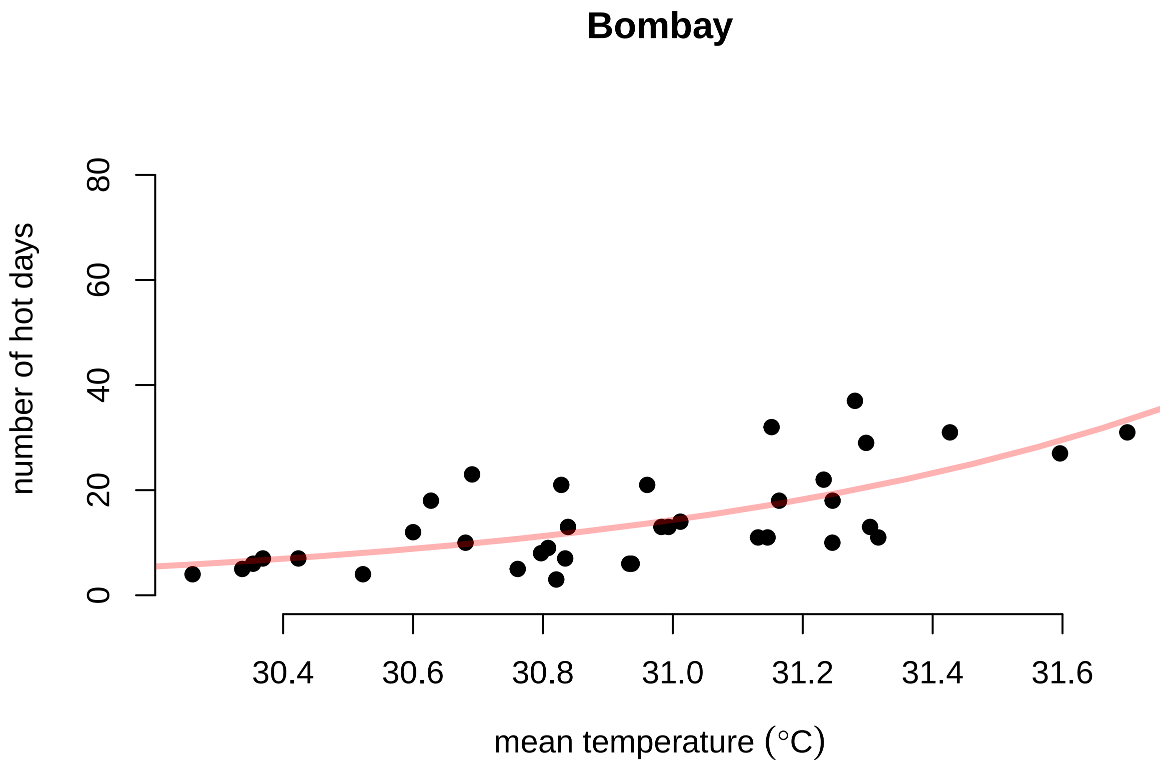
```
trh <- c(40,35,35)
for (i in 1:3) {
  y <- subset(climatrans.tx,is=i)
  if (!file.exists(paste('dse.tx.climatrans.',loc(y),'.rda',sep=''))) {
    dse.tx <- DSensemble.t2m(y,biascorrect=TRUE,type='ncdf4',
                        predictor=predictor,nmin=60,verbose=FALSE)
    save(file=paste('dse.tx.climatrans.',loc(y),'.rda',sep=''),dse.tx)
  } 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
  dse.tx[is.element(year(dse.tx),2100),] <- NA   # some of the results for 2100 are suspect
  hw <- hotsummerdays(x=y,dse=dse.tx,threshold=trh[i],it=NULL,plot=TRUE,new=FALSE)
  #plot(hw)
}
```

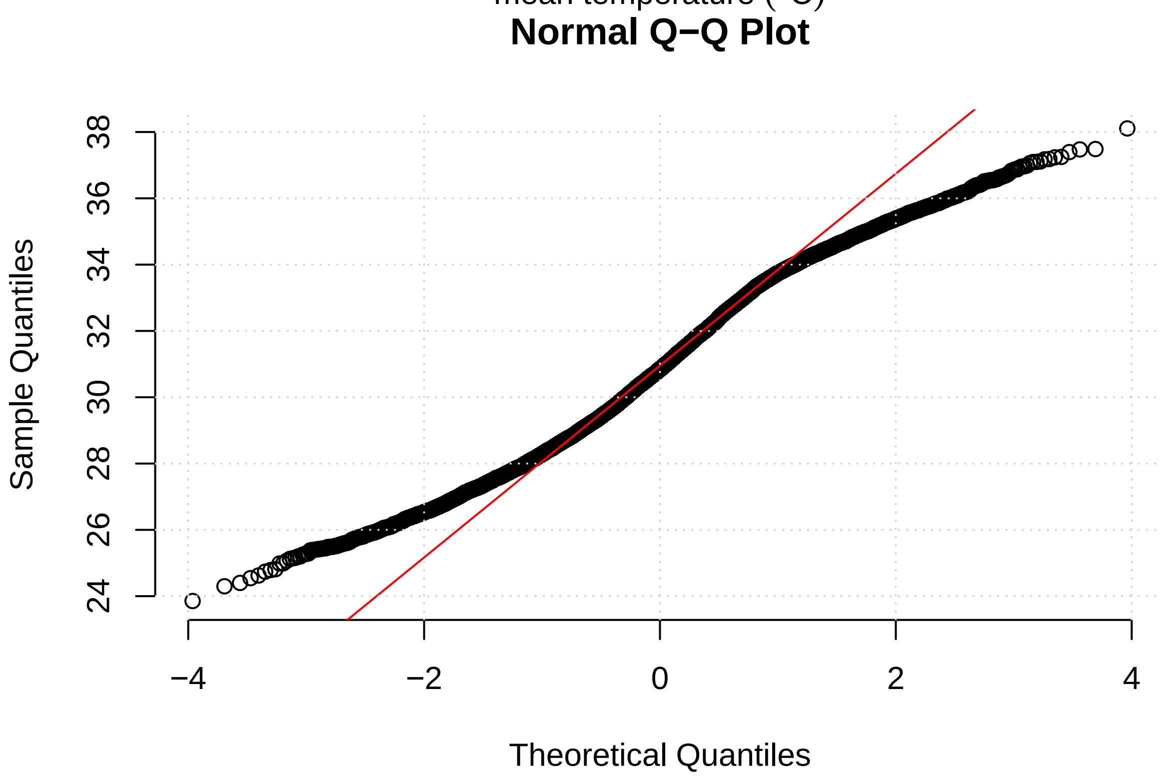

Delhi
number of hot days
0
20
40
60
80
30.0
30.5
31.0
31.5
32.0
32.5
33.0
mean temperature (°C)

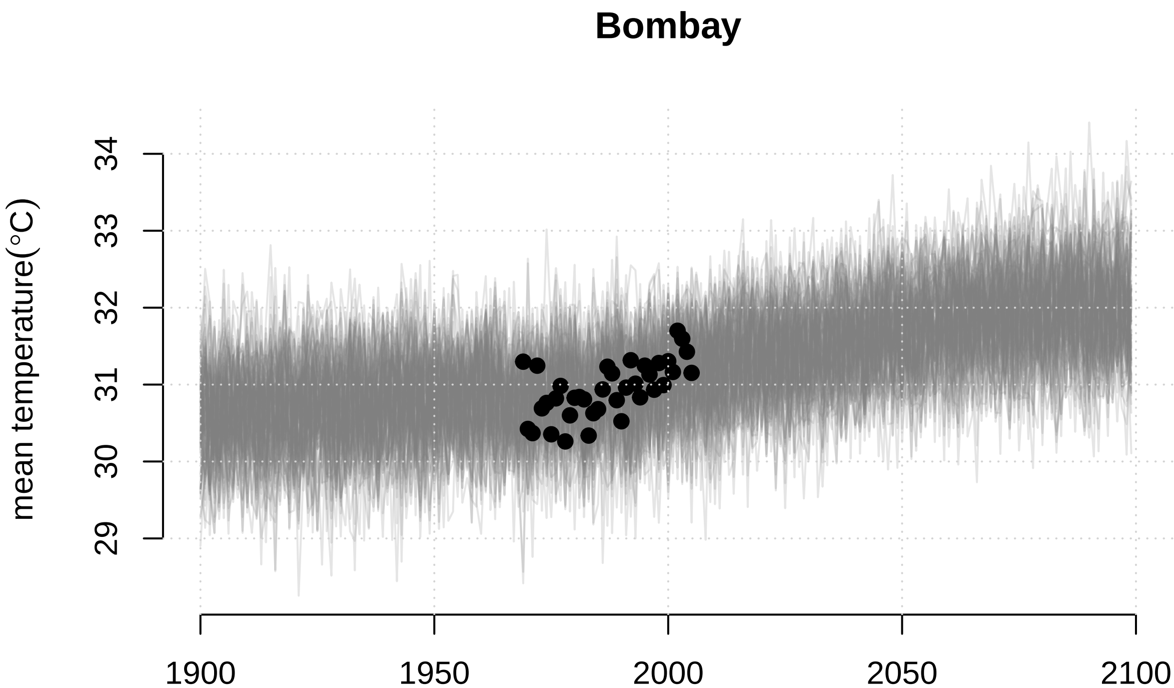

Normal Q−Q Plot
Sample Quantiles
15
20
25
30
35
40
45
−4
−2
0
2
4
Theoretical Quantiles

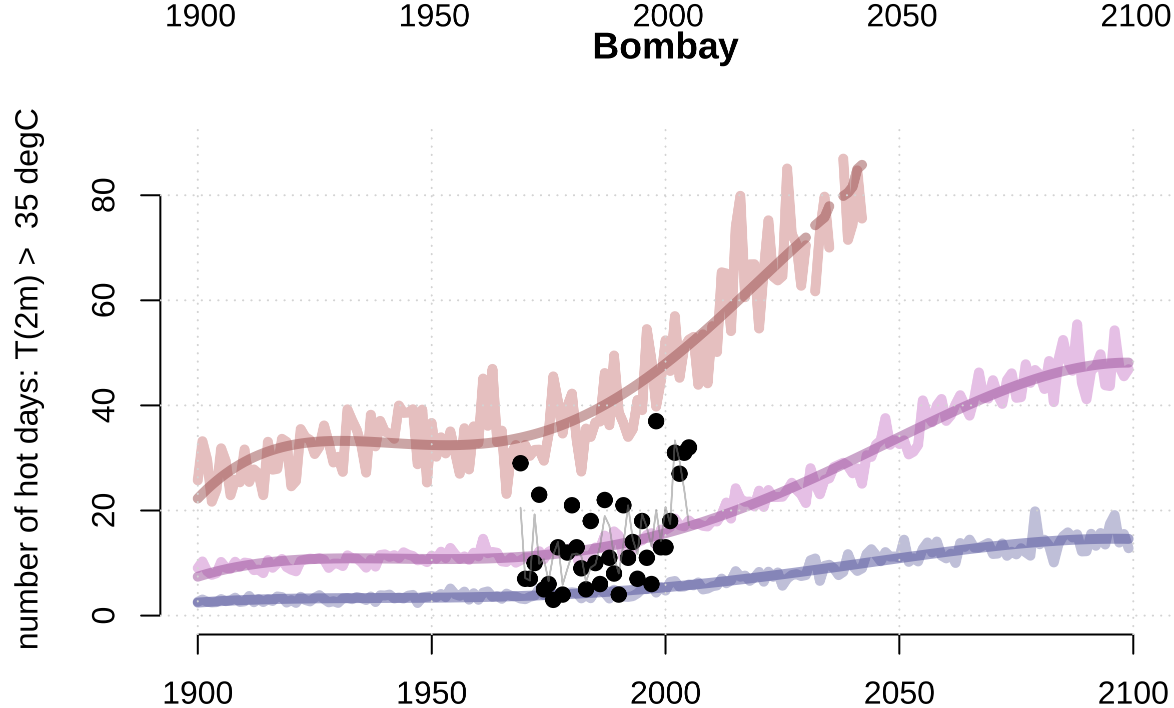

Delhi
mean temperature(°C)
30
32
34
36
1900
1950
2000
2050
2100

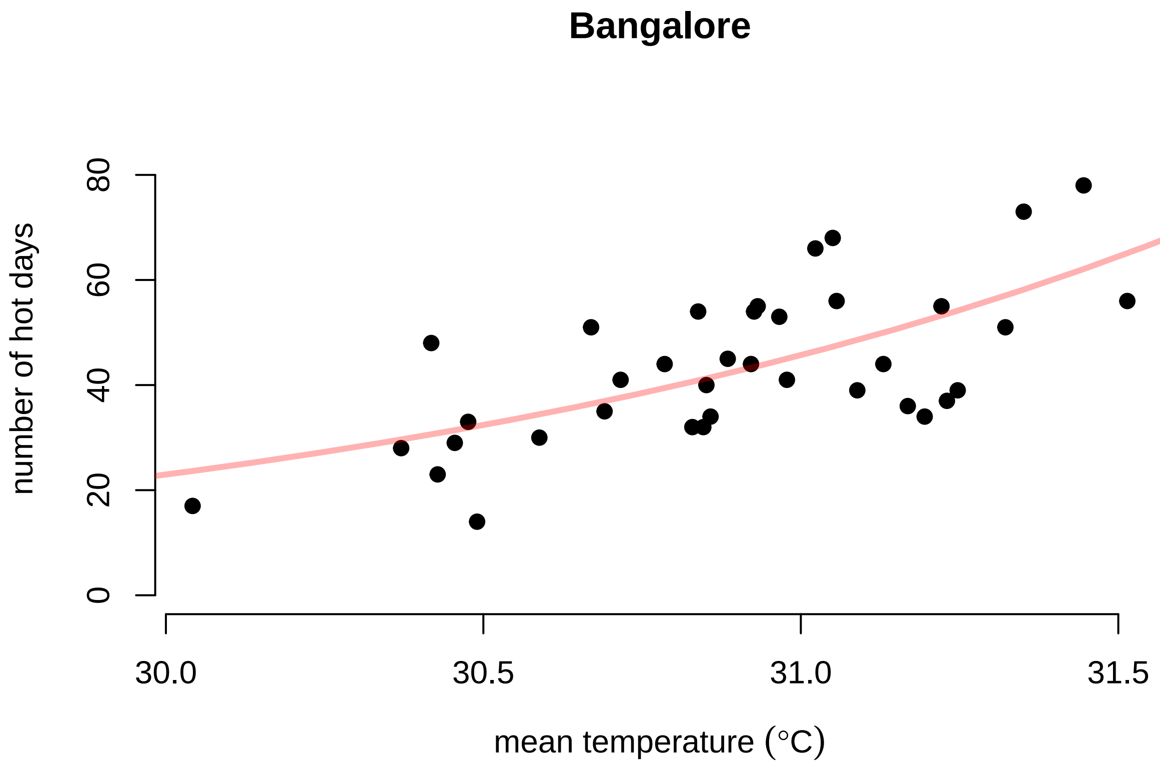

Delhi
number of hot days: T(2m) > 40 degC
0
20
40
60
80
1900
1950
2000
2050
2100

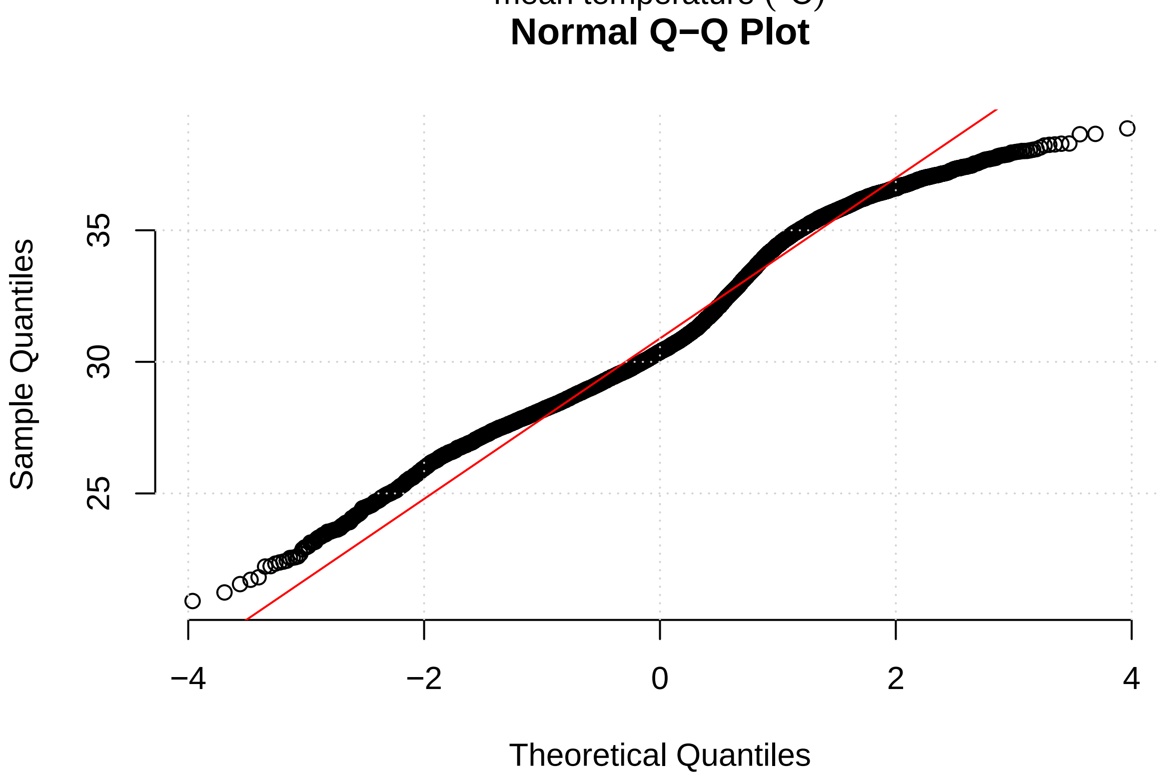

Bombay
number of hot days
0
20
40
60
80
30.4
30.6
30.8
31.0
31.2
31.4
31.6
mean temperature (°C)

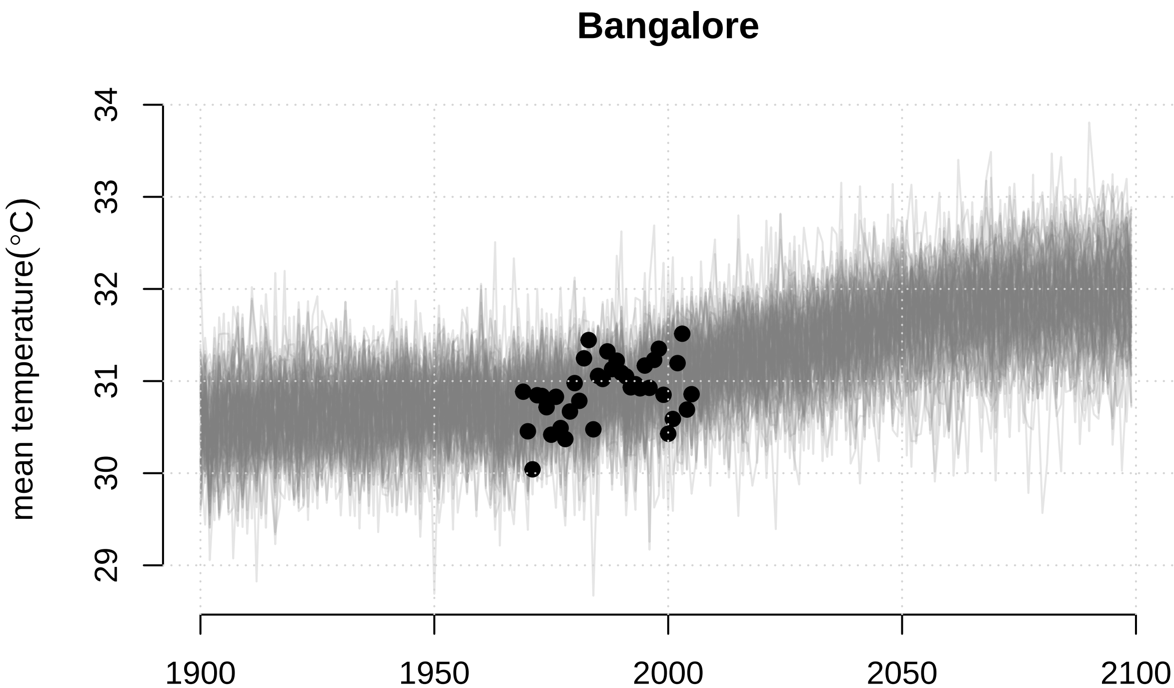

Normal Q−Q Plot
Sample Quantiles
24
26
28
30
32
34
36
38
−4
−2
0
2
4
Theoretical Quantiles

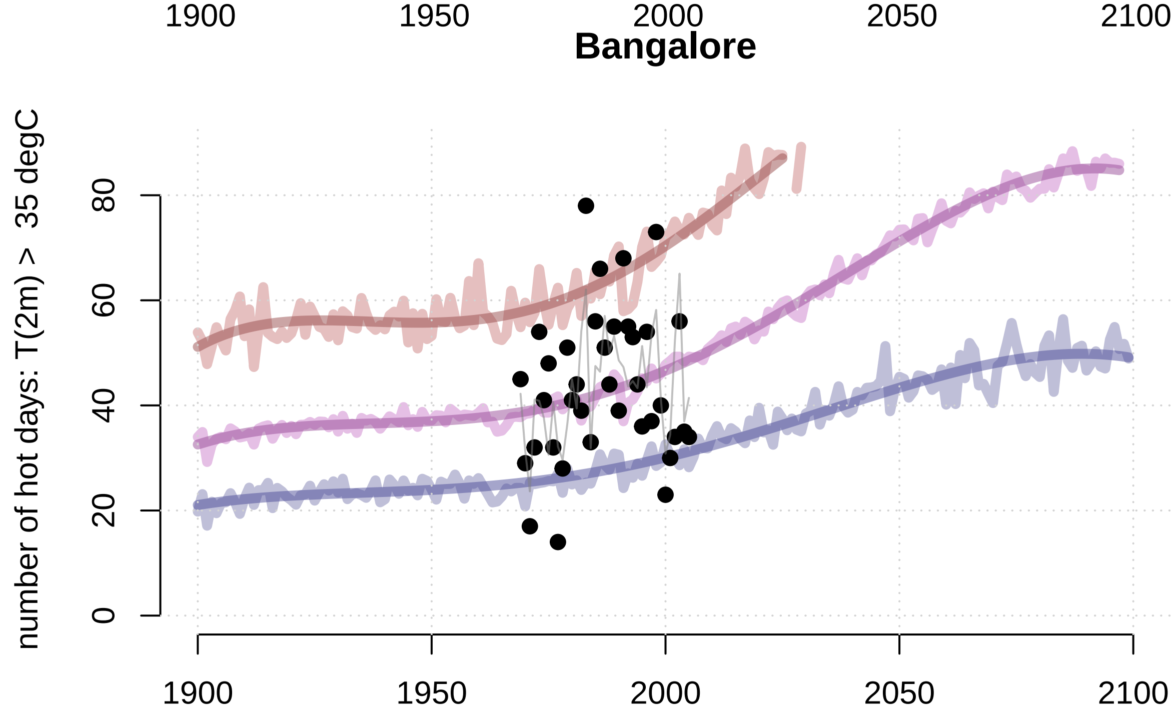

Bombay
mean temperature(°C)
29
30
31
32
33
34
1900
1950
2000
2050
2100


Bombay
number of hot days: T(2m) > 35 degC
0
20
40
60
80
1900
1950
2000
2050
2100


Bangalore
number of hot days
0
20
40
60
80
30.0
30.5
31.0
31.5
mean temperature (°C)


Normal Q−Q Plot
Sample Quantiles
25
30
35
−4
−2
0
2
4
Theoretical Quantiles





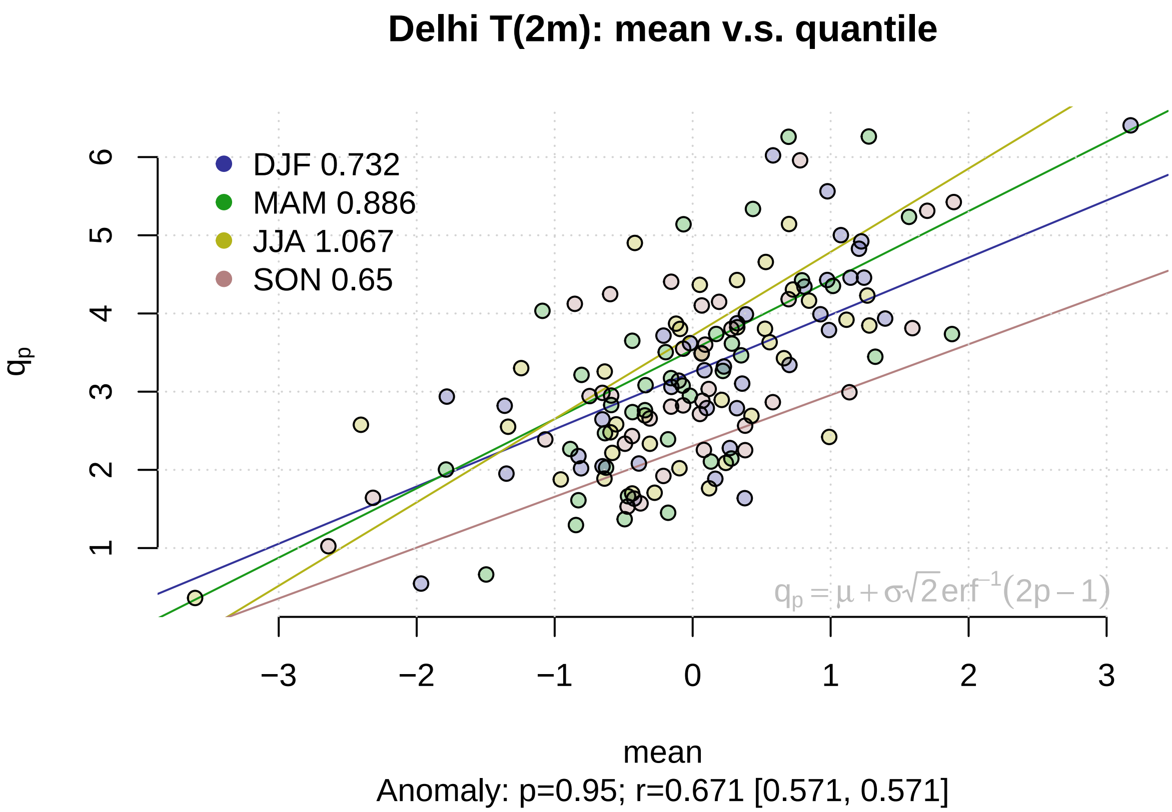
The general linear models provide a reasonable description of the number of hot events given the summer mean temperature. The tails of the distribution do not quite conform to the normal distribution, however. The downscaled model summer mean temperatures are comparable with observed summer mean temperatures.

Check link between the mean and 95-percentile

More traditional approaches of downscaling typically involve either downscaling daily data and estimate a probability density function (pdf) from these results or downscale the pdf directly in terms of the parameters describing the shape (http://climatescience.oxfordre.com/view/10.1093/acrefore/9780190228620.001.0001/acrefore-9780190228620-e-27). The probability of hot days may be estimated based on the pdf, and it is important to check the consistency between the data and the relevant parametric distribution.

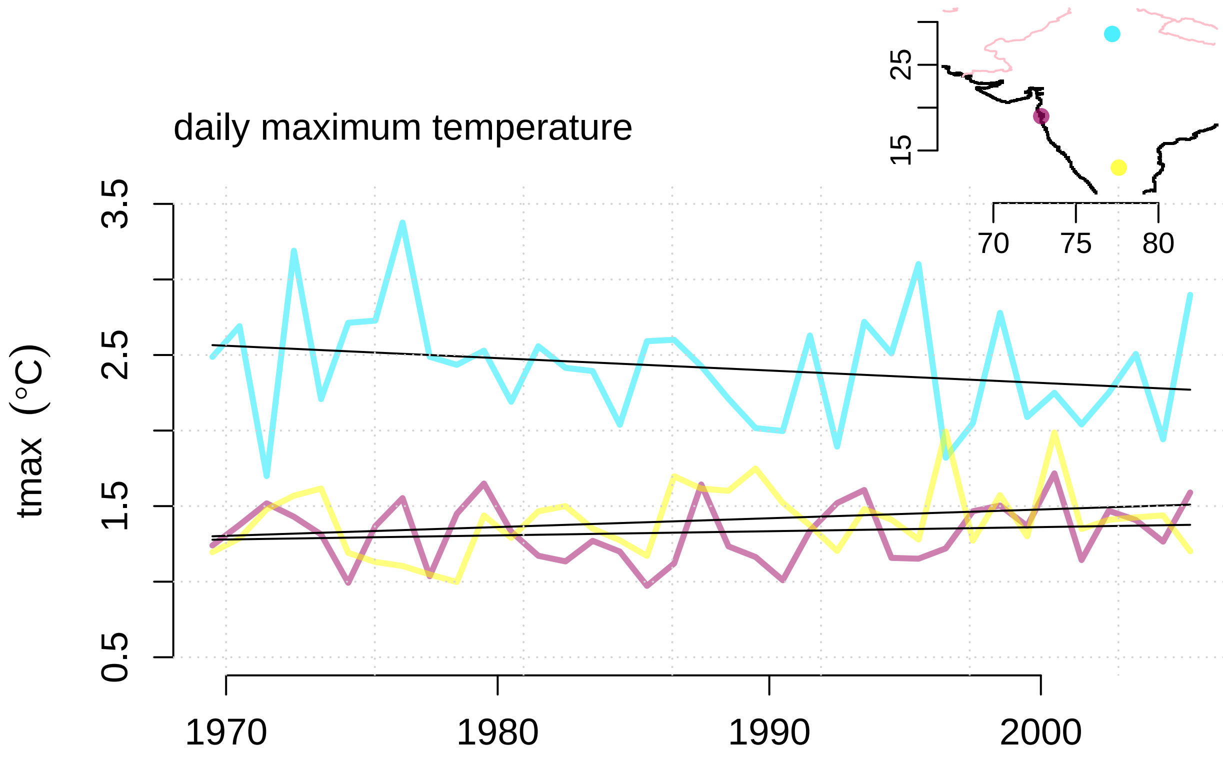
Check the correlation between the seasonal percentiles and the seasonal mean temperature, assuming the temperature is normally distributed.

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



Check 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.

```
tx.sd <- as.4seasons(anomaly(climatrans.tx),FUN='sd')
plot(subset(tx.sd,it='jja'),new=FALSE)
for(i in 1:3) lines(trend(subset(subset(tx.sd,it='jja'),is=i)))
grid()
```




```
print(trend(subset(tx.sd,it='jja'),result='pval'))
```

```
## [1] 0.1684444 0.3809063 0.1005206
## attr(,"location")
## [1] "Delhi"     "Bombay"    "Bangalore"
## attr(,"longitude")
## [1] 77.2 72.9 77.6
## attr(,"latitude")
## [1] 28.6 19.0 13.0
## attr(,"altitude")
## [1] NA NA NA
## attr(,"cntr")
## [1] "India" "India" "India"
## attr(,"stid")
## [1] NA NA NA
## attr(,"history")
## attr(,"history")$call
## attr(,"history")$call[[1]]
## trend.station(subset(tx.sd, it = "jja"), result = "pval")
##
##
## attr(,"history")$timestamp
## [1] "Wed Aug  3 14:16:21 2016"
##
## attr(,"history")$session
## attr(,"history")$session$R.version
```

```
## [1] "R version 3.1.3 (2015-03-09)"
##
## attr(,"history")$session$esd.version
## [1] "esd_1.2"
##
## attr(,"history")$session$platform
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

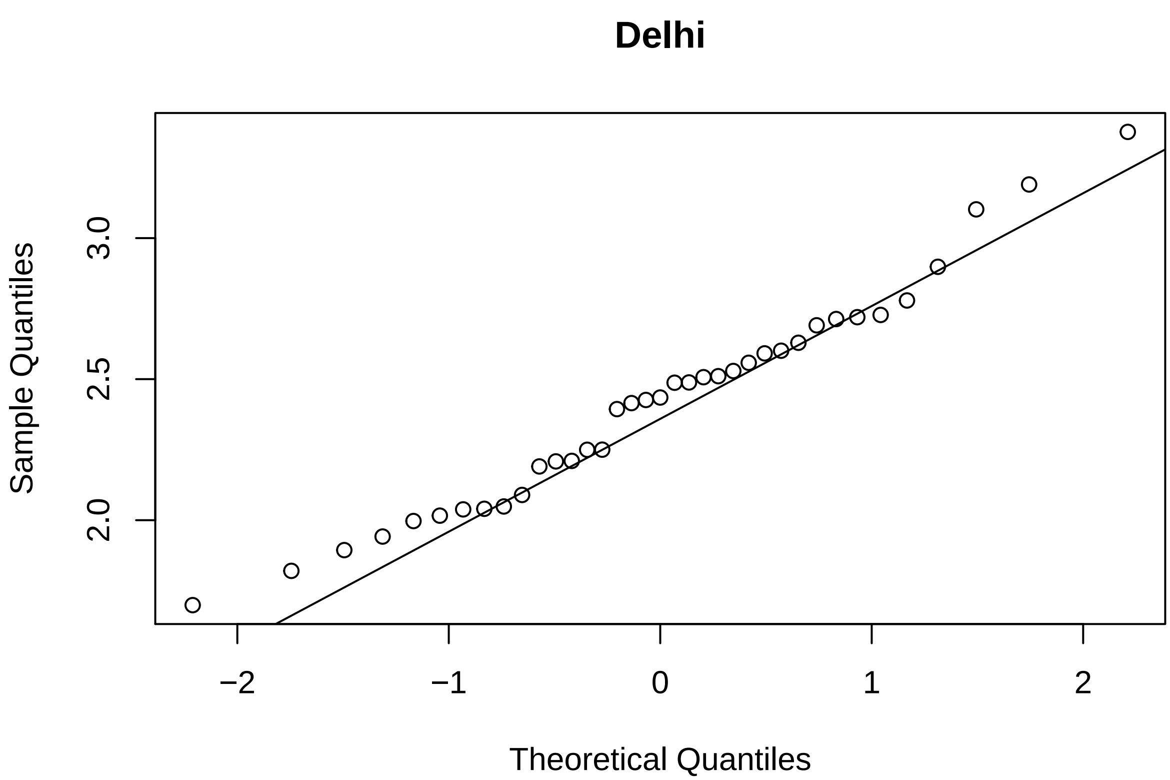
```
print(colMeans(climatrans.tx))
```

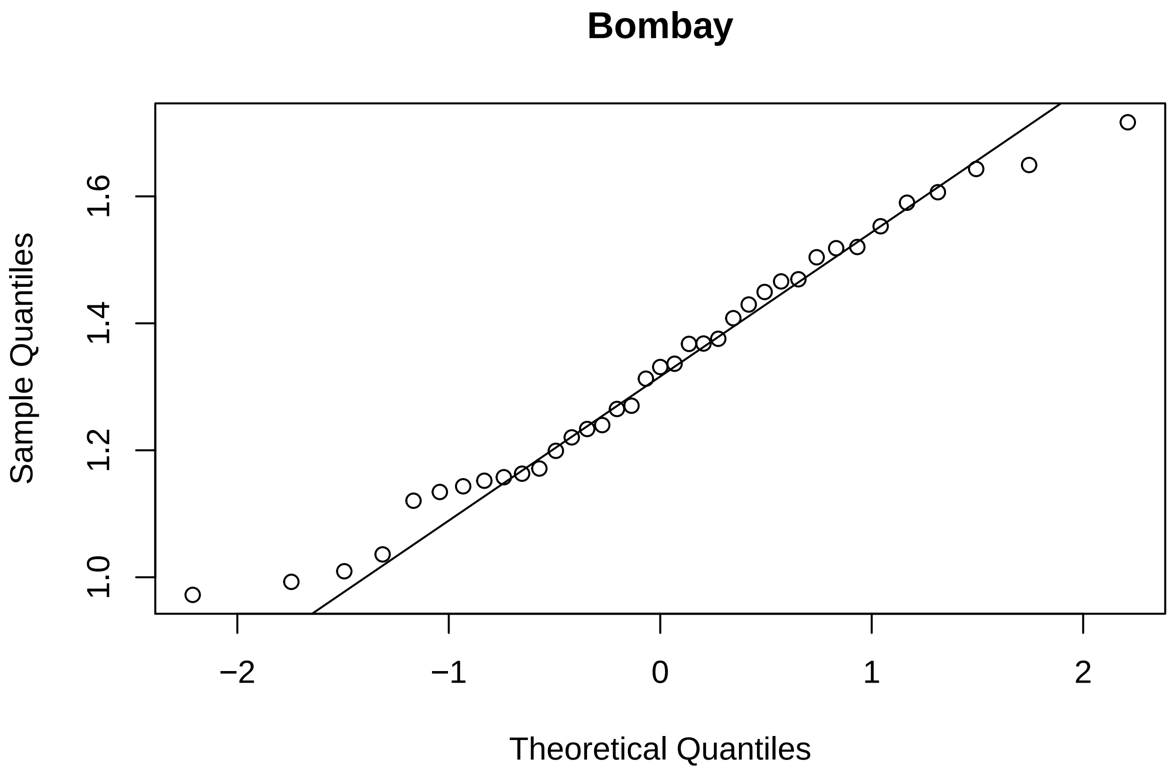
```
##      Delhi    Bombay Bangalore
##   31.56637  30.93555  30.89198
```

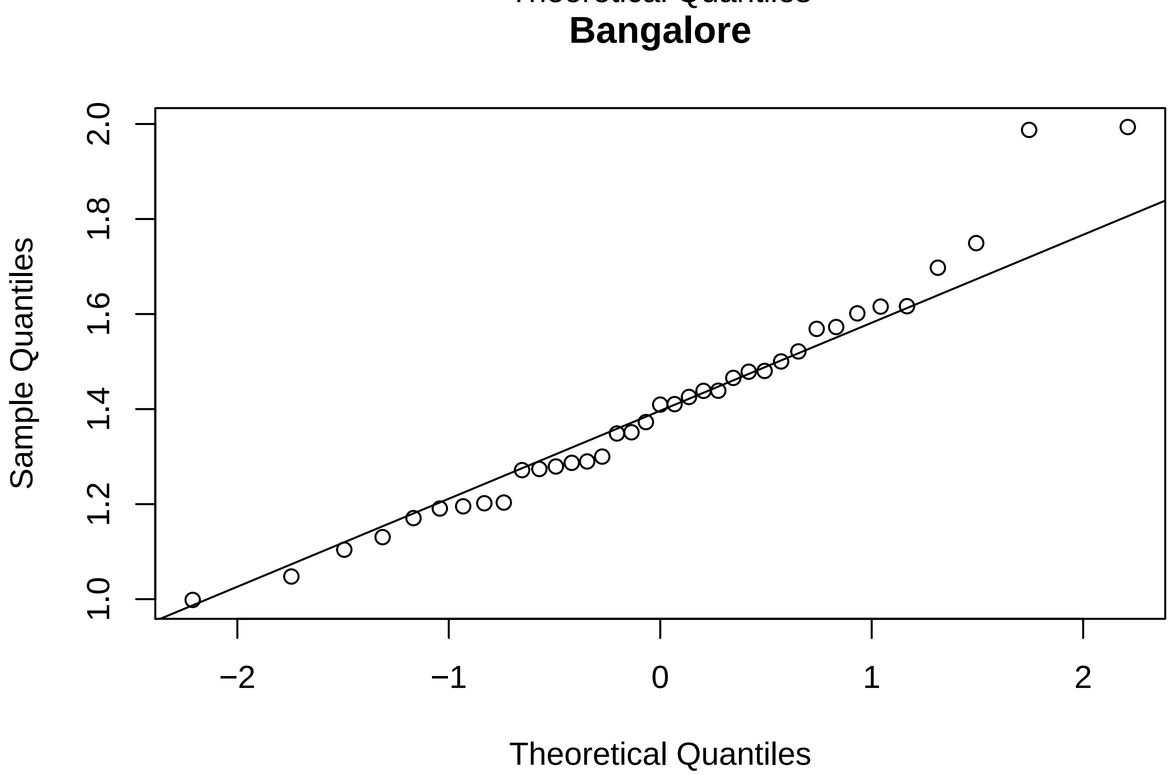
The interannual variabilty in the seasonal daily max temperature variance of ~2 degrees is small compared to the temperature level (especially in degrees Kelvin).

Check if the variance in standard deviation conforms to randomness: normally distributed

```
for (i in 1:3) {
  qqnorm(coredata(subset(subset(tx.sd,it='jja'),is=i)),main=loc(tx.sd)[i])
  qqline(coredata(subset(subset(tx.sd,it='jja'),is=i)),main=loc(tx.sd)[i])
}
```



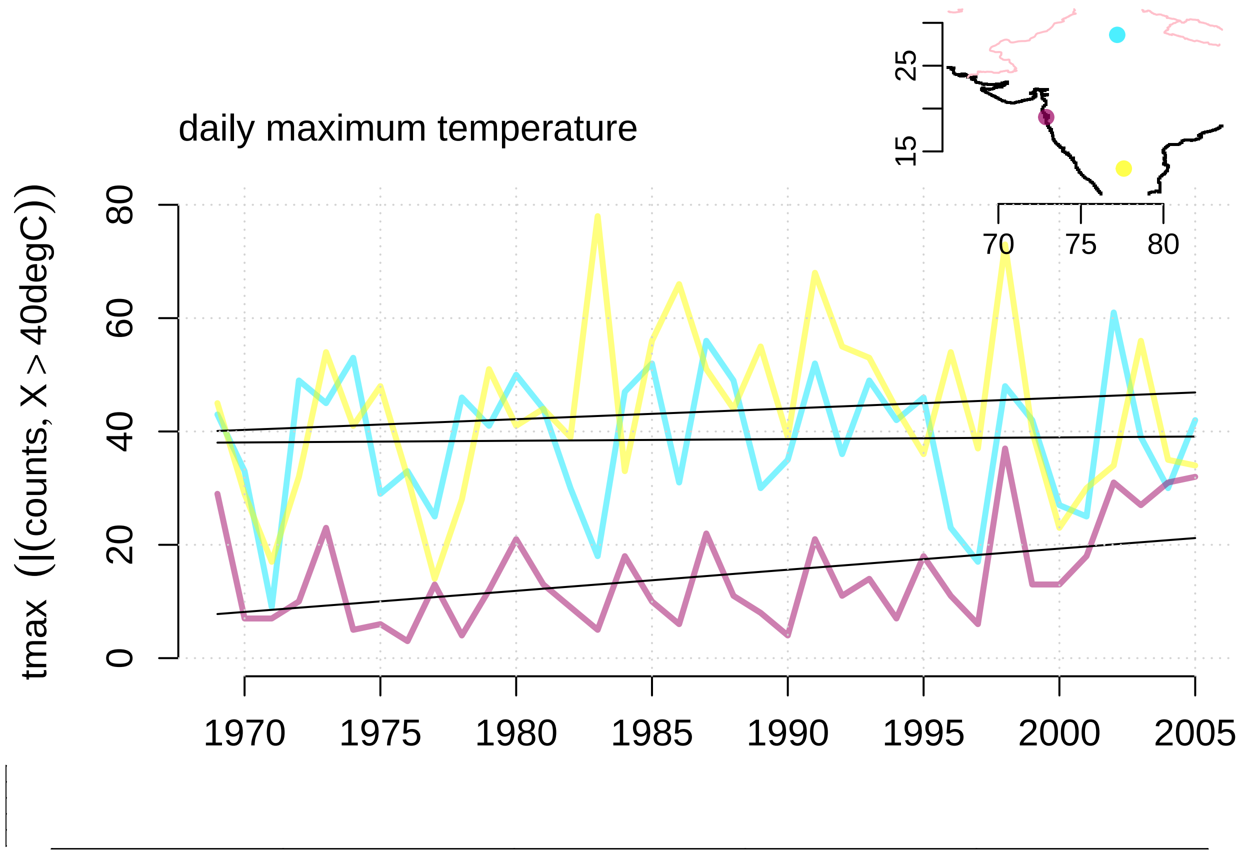




Are the hot events behaving like Poisson distribution?

Take hot events to be any day warmer than 40C:

```r
for(i in 1:3) {
  nhot1 <- annual(subset(climatrans.tx,is=i),FUN='count',threshold=trh[i])
  if (i==1) nhot <- nhot1 else nhot <- combine(nhot,nhot1)
}
plot(nhot,new=FALSE)
for(i in 1:3) lines(trend(subset(nhot,is=i)))
grid()
```




```r
print(trend(nhot,result='pval'))
```

```
## [1] 0.875314768 0.006774845 0.406790080
## attr(,"location")
## [1] "Delhi"     "Bombay"    "Bangalore"
## attr(,"longitude")
## [1] 77.2 72.9 77.6
## attr(,"latitude")
## [1] 28.6 19.0 13.0
## attr(,"altitude")
## [1] NA NA NA
## attr(,"cntr")
## [1] "India" "India" "India"
## attr(,"stid")
## [1] NA NA NA
## attr(,"history")
## attr(,"history")$call
```

```
## attr(,"history")$call[[1]]
## trend.station(nhot, result = "pval")
##
##
## attr(,"history")$timestamp
## [1] "Wed Aug  3 14:16:31 2016"
##
## attr(,"history")$session
## attr(,"history")$session$R.version
## [1] "R version 3.1.3 (2015-03-09)"
##
## attr(,"history")$session$esd.version
## [1] "esd_1.2"
##
## attr(,"history")$session$platform
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

```
print(paste('Mean number=',mean(nhot[,1],na.rm=TRUE),'Variance=',var(nhot[,1],na.rm=TRUE),
            'should equal',mean(nhot[,1],na.rm=TRUE),'for Poisson process'))
```

```
## [1] "Mean number= 38.5675675675676 Variance= 144.252252252252 should equal 38.5675675675676 for Pois
```

```
hist(coredata(nhot[,1]),freq=FALSE,col='grey',
     main=loc(nhot)[1])
lines(dpois(x=seq(0,max(nhot[,1]),by=1),lambda=mean(nhot[,1],na.rm=TRUE)),lwd=2,col='red')
grid()
```