Supplement of Adv. Stat. Clim. Meteorol. Oceanogr., 4, 37–52, 2018
https://doi.org/10.5194/ascmo-4-37-2018-supplement

ASCMO
Open Access

Supplement of

Downscaling probability of long heatwaves based on seasonal mean daily maximum temperatures

Rasmus E. Benestad et al.

Correspondence to: Rasmus E. Benestad (rasmus.benestad@met.no)

Supporting Material for Probability of long heatwaves

Rasmus Benestad

February 7, 2017

The supporting material (SM) for the paper 'Downscaling probability for long heatwaves based on seasonal mean temperatures' by Benestad et al.

The paper has been submitted to Advances in Statistical Climatology, Meteorology and Oceanography (Copernicus) https://www.advances-statistical-climatology-meteorology-oceanography.net

About the CixPAG project

CiXPAG will investigate the complex interactions between climate extremes, air pollution and agricultural ecosystems. Climate extremes (e.g., droughts, floods, heatwaves) and air pollution events often co-occur causing substantial losses in agricultural productivity. We do not yet fully understand how these stresses interact and what the impacts of the combined climate - air pollution effects may be for agricultural ecosystems in some of the most vulnerable parts of the world. This lack of knowledge is particularly challenging considering the threats that climate change and food security pose to society.
The novel research proposed in CiXPAG will collect new experimental data and develop new modelling techniques to integrate knowledge on changes in climate extremes and air pollution to assess effects on agricultural productivity. Integration of farmers' knowledge will enable the results to be translated into agricultural adaptation options within the particular socio-economic and political context.

R Markdown

This script uses the R computing environment that runs on all platforms and is freely available from http://cran.r-project.org. You can also run it in the R-studio environment that also offers a free version from http://rstudio.com. You would need to install a few extra packages and libraries, e.g for developing code for reading/writing netCDF files https://www.unidata.ucar.edu/software/netcdf/.
This is an R Markdown document to assess the duration of warm spells in Indian maximum temperature (tmax) data. The objective of this R-markdown document is to provide an exact recipe for the analysis presented in the main paper: rerunning this script will replicate the exact steps taken. 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.

The following lines are of technical nature to do some magic: to extract the R-code from this script (this chunk with lines is usually not implemented - `eval=FALSE`):

```
## Extract just the R-code for the generation of the graphics used for the figures seperately.
## Check if you need to get the devtools-package:
install.knitr <- ("knitr" %in% rownames(installed.packages()) == FALSE)

if (install.knitr) {
  print('Need to install the knitr package')
  ## You need online access.
    install.packages('knitr')
}
library(knitr)
purl('~/git/esd_Rmarkdown/CixPAG/spell-statistics.Rmd', output='~/git/esd_Rmarkdown/CixPAG/spell-statis
```

And to knit the final document:

The 'esd' analysis tool

This analysis relies on the R-packages `esd` ('extreme simple data', formerly 'empirical-statistical downscaling') that is available from http://github.com/metno/esd. See its GitHub wikipage for more informationhttp://github.com/metno/esd/wiki. The following chunk of R code installs the 'esd-package automatically.

```
## Check if you need to get the devtools-package:
install.devtools <- ("devtools" %in% rownames(installed.packages()) == FALSE)

if (install.devtools) {
  print('Need to install the devtools package')
  ## You need online access.
    install.packages('devtools')
}

## Use the devtools-package for simple facilitation of installing.
library('devtools')
install_github('metno/esd')
```

```
## Skipping install of 'esd' from a github remote, the SHA1 (74e57b3a) has not changed since last insta
##   Use `force = TRUE` to force installation
```

```
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(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:esd':
##
##     select
```

Additional R-packages

The analysis also makes use of the LatticeKrig package for the interpolation of large spatial datasets. The following code will automatically install this package if it is missing.

```
## Check if you need to get the devtools-package:
install.latticekrig <- ("LatticeKrig" %in% rownames(installed.packages()) == FALSE)

if (install.latticekrig) {
  print('Need to install the LatticeKrig package')
  ## You need online access.
    install.packages('LatticeKrig')
}
```

If you passed this point, you sucessfully managed to install and load `esd` and additional packages. The analysis can now proceed.

Daily Indian temperatures

The following lines extracts daily temperature records for India from the Global Historic Climate Network (GHCND) through the R-package `esd` (open code and available from http://github.com/metno/esd).

Get the daily maximum temperature and check the number of valid data points `nv`:

```
if (!file.exists("tmax.india.rda")) {
  ss <- select.station(param='tmax',src='GHCND',cntr='India',nmin=50)
  tmax <- station(ss)
  map(tmax,FUN='nv',new=FALSE)
  save(tmax,file='tmax.india.rda')
} else load("tmax.india.rda")
```

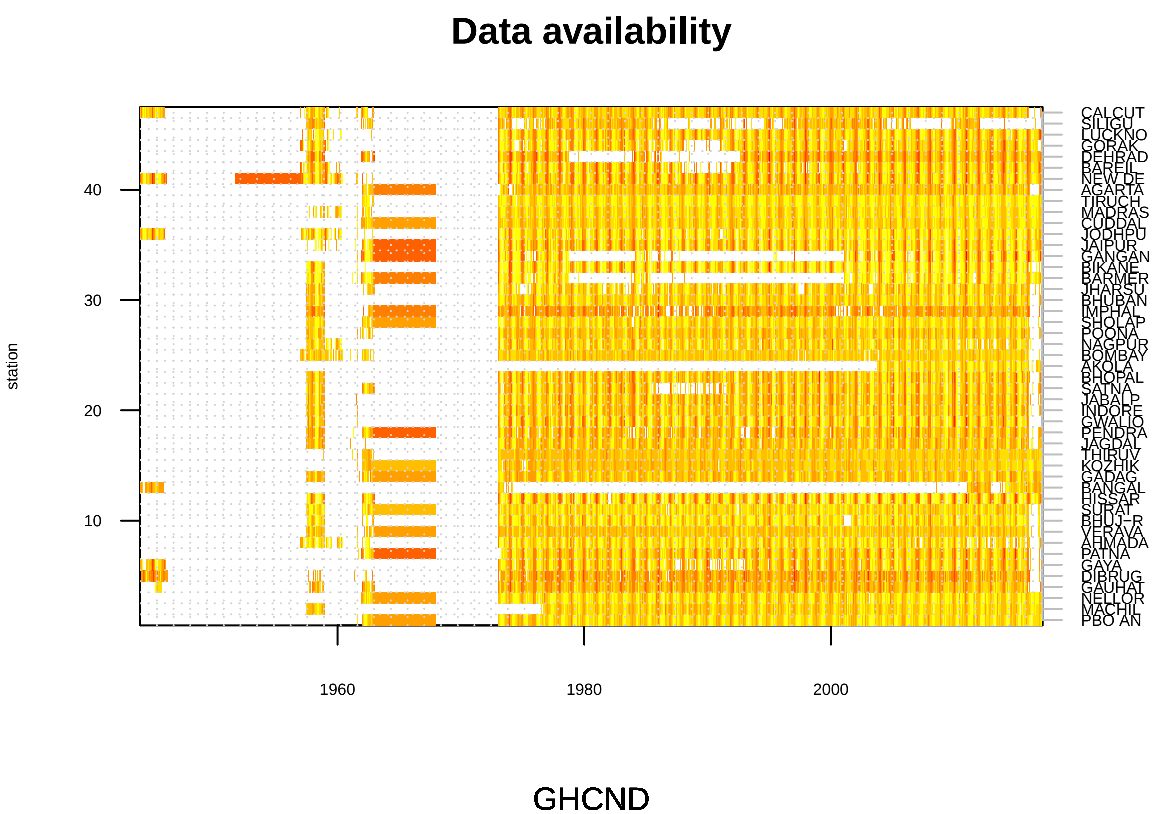
Get the daily minimum temperature:

```
library(esd)
if (!file.exists("tmin.india.rda")) {
  ss <- select.station(param='tmin',src='GHCND',cntr='India',nmin=50)
  tmin <- station(ss)
  map(tmin,FUN='nv',new=FALSE)
  save(tmin,file='tmin.india.rda')
} else load("tmin.india.rda")
```

Data quality

There is a need to check the data quality: show gaps of missing data and weed out stations with lots of missing values.

```
diagnose(tmax)
```




```r
## Remove periods with mostly missing data and stations with few valid data
Y <- subset(tmax,it=c(1970,2015))
nv <- apply(coredata(Y),2,FUN='nv')       ##nv = number of valid data points
Y <- subset(Y,is=nv > 10290)
```

There were some stations with little data or short records which have been omitted here. We have only kept records with plenty of valid data (more than 10290 data points) between 1970 and 2015.

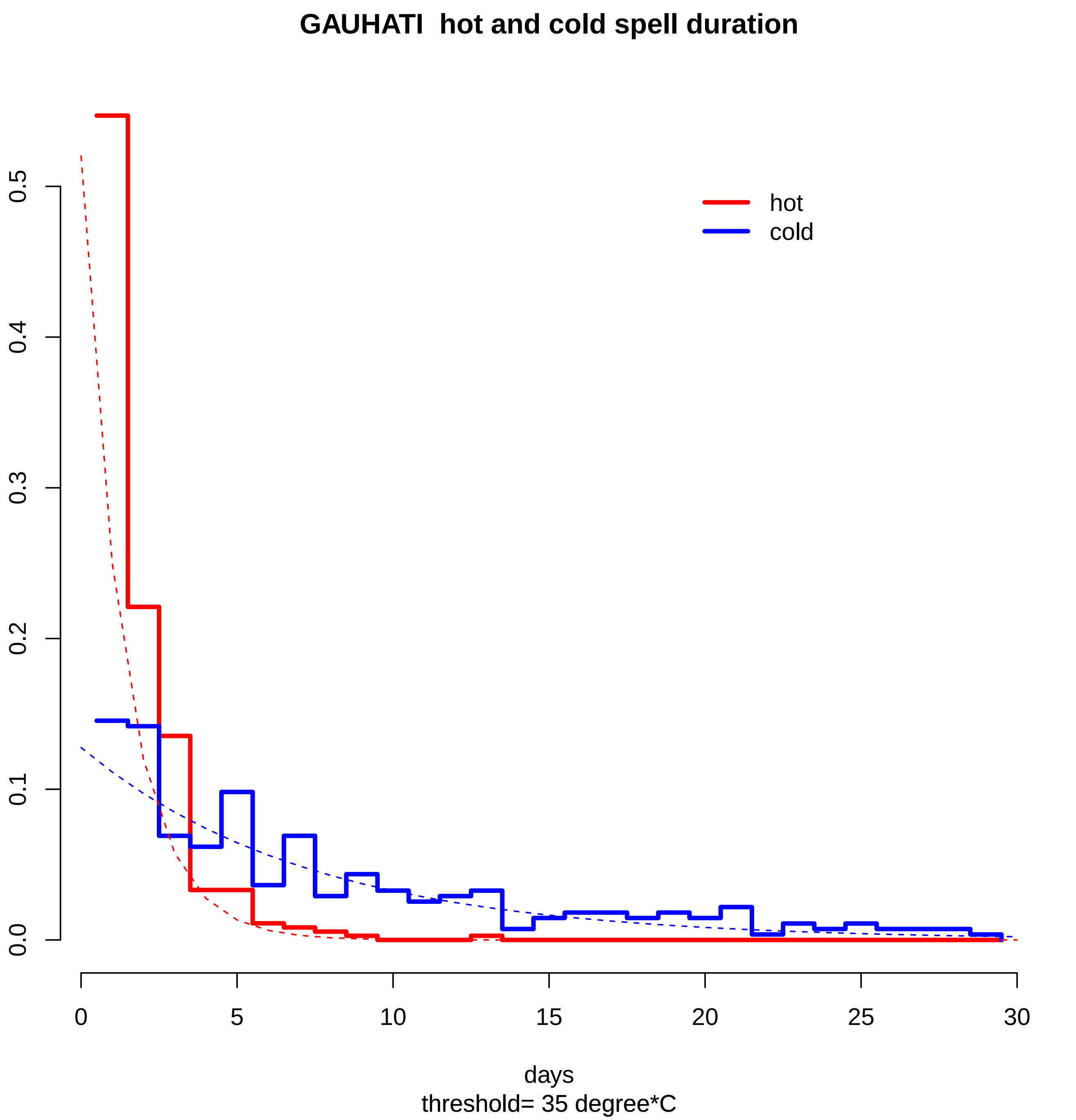
Spell statistics for heatwaves

For all Indian wheat varieties, the main challenge is the high temperatures in the final growing phase, late in the season from February to April. All temperatures above optimal decrease the yield. Here the upper temperature threshold was set to $T > 35°C$ for 5 days;

```r
d <- dim(Y)
n.yrs <- diff(range(year(Y)))+1
f.gt.5 <- rep(NA,d[2])        ## The portion of hot days with duration longer than five days
nf.gt.5 <- rep(NA,d[2])       ## The portion of seasons with a 5-day or longer 35C heatwave
Pr.Tmax.gt.35<- rep(NA,d[2])  ## The probability of temperatures  greater than 35C
for (i in 1:d[2]) {
  sds <- spell(subset(Y,is=i),threshold=35,upper=30)
  if (i==1) lws <- subset(sds,is=1) else
            lws <- combine.stations(lws,subset(sds,is=1))
  ## Example of the spell statistics
  if (i==4) hist(sds,new=FALSE)

  ## Heatwaves in February-March-April
  heatwave <- subset(sds,is=1,it=month.abb[2:4])
```

```r
  ## The fraction of events lasting more than 5 days
  f.gt.5[i] <- sum(heatwave > 5)/sum(is.finite(heatwave))
  #f.gt.5[i] <- sum(subset(sds,is=1) > 5)/sum(is.finite(subset(sds,is=1)))
  #f.gt.5[i] <- sum(subset(sds,is=1) > 5)/count(subset(Y,is=i),threshold=35)
  cc <- coredata(heatwave); cc[cc <= 5] <- NA; cc[is.finite(cc)] <- 1
  hw <- zoo(cc[is.finite(cc)],order.by=index(heatwave)[is.finite(cc)])
  ## The portion of seasons with a 5-day or longer 35C heatwave
  nf.gt.5[i] <- length(rownames(table(year(hw))))/n.yrs
  ## Probability of a 5-day or longer 35C heatwave in February-March-April
  Pr.Tmax.gt.35[i] <- 1-pnorm(35,mean=mean(subset(Y,it=month.abb[2:4],is=i),na.rm=TRUE),
                              sd=sd(subset(Y,it=month.abb[2:4],is=i),na.rm=TRUE))
  print(c(f.gt.5[i],Pr.Tmax.gt.35[i]))
}
```

```
## [1] "Warning for PBO ANANTAPUR - 2524 missing values ( 16.1  %) filled by interpolation"
## [1] 0.3162393 0.7856550
## [1] "Warning for MACHILIPATNAM  - 2585 missing values ( 17  %) filled by interpolation"
## [1] 0.2142857 0.2291169
## [1] "Warning for NELLORE - 2711 missing values ( 17.3  %) filled by interpolation"
## [1] 0.3367876 0.5429190
## [1] "Warning for GAUHATI  - 2437 missing values ( 15.5  %) filled by interpolation"
```




```
## [1] 0.04597701 0.06079823
## [1] "Warning for DIBRUGARH/MOHANBAR - 5344 missing values ( 34  %) filled by interpolation"
## [1] 0.14285714 0.01487847
## [1] "Warning for PATNA  - 3179 missing values ( 20.2  %) filled by interpolation"
## [1] 0.4207317 0.2984817
## [1] "Warning for AHMADABAD  - 3828 missing values ( 24.4  %) filled by interpolation"
## [1] 0.2983871 0.5469897
## [1] "Warning for VERAVAL  - 3054 missing values ( 19.5  %) filled by interpolation"
## [1] 0.03255814 0.11909073
## [1] "Warning for BHUJ-RUDRAMATA - 3090 missing values ( 19.7  %) filled by interpolation"
## [1] 0.3043478 0.5085943
```

```
## [1] "Warning for SURAT  - 3421 missing values ( 21.8  %) filled by interpolation"
## [1] 0.3630769 0.4975182
## [1] "Warning for HISSAR - 2589 missing values ( 16.5  %) filled by interpolation"
## [1] 0.3181818 0.2803953
## [1] "Warning for GADAG  - 3076 missing values ( 19.6  %) filled by interpolation"
## [1] 0.450000 0.545762
## [1] "Warning for KOZHIKODE  - 3682 missing values ( 23.5  %) filled by interpolation"
## [1] 0.12765957 0.09627756
## [1] "Warning for THIRUVANANTHAPURAM - 2244 missing values ( 14.3  %) filled by interpolation"
## [1] 0.008064516 0.088270267
## [1] "Warning for JAGDALPUR  - 3980 missing values ( 25.3  %) filled by interpolation"
## [1] 0.4453125 0.4746028
## [1] "Warning for PENDRA ROAD  - 4596 missing values ( 29.3  %) filled by interpolation"
## [1] 0.3185185 0.3048203
## [1] "Warning for GWALIOR  - 3282 missing values ( 20.9  %) filled by interpolation"
## [1] 0.2301587 0.3414621
## [1] "Warning for INDORE - 2941 missing values ( 18.7  %) filled by interpolation"
## [1] 0.2746479 0.4446849
## [1] "Warning for JABALPUR - 3214 missing values ( 20.5  %) filled by interpolation"
## [1] 0.2727273 0.3818472
## [1] "Warning for BHOPAL/BAIRAGARH - 2744 missing values ( 17.5  %) filled by interpolation"
## [1] 0.2880000 0.3952677
## [1] "Warning for BOMBAY/SANTACRUZ - 2892 missing values ( 18.4  %) filled by interpolation"
## [1] 0.04444444 0.15441221
## [1] "Warning for NAGPUR SONEGA - 3191 missing values ( 20.3  %) filled by interpolation"
## [1] 0.3679245 0.5817169
## [1] "Warning for POONA  - 1540 missing values ( 9.8  %) filled by interpolation"
## [1] 0.4035088 0.5366835
## [1] "Warning for SHOLAPUR - 2038 missing values ( 13  %) filled by interpolation"
## [1] 0.3516484 0.7910898
## [1] "Warning for BHUBANE - 2424 missing values ( 15.4  %) filled by interpolation"
## [1] 0.3830645 0.5133757
## [1] "Warning for BIKANER  - 1781 missing values ( 11.3  %) filled by interpolation"
## [1] 0.3846154 0.3424543
## [1] "Warning for JAIPUR/SA - 1530 missing values ( 9.7  %) filled by interpolation"
## [1] 0.3773585 0.2965087
## [1] "Warning for JODHPUR - 3264 missing values ( 20.8  %) filled by interpolation"
## [1] 0.3761468 0.3976173
## [1] "Warning for CUDDALO - 3009 missing values ( 19.2  %) filled by interpolation"
## [1] 0.2155172 0.1157803
## [1] "Warning for MADRAS/MINAMBAKKAM - 2113 missing values ( 13.5  %) filled by interpolation"
## [1] 0.3031674 0.3039122
## [1] "Warning for TIRUCHCHIRAPALLI - 3859 missing values ( 24.6  %) filled by interpolation"
## [1] 0.3972603 0.5874839
## [1] "Warning for AGARTALA - 3527 missing values ( 22.6  %) filled by interpolation"
## [1] 0.2231405 0.1416561
## [1] "Warning for NEW DELHI/S - 1135 missing values ( 7.2  %) filled by interpolation"
## [1] 0.3252033 0.2166436
## [1] "Warning for LUCKNOW/AMAUSI - 2242 missing values ( 14.3  %) filled by interpolation"
## [1] 0.4054054 0.3303515
## [1] "Warning for CALCUTTA/DUM DUM - 2379 missing values ( 15.2  %) filled by interpolation"
## [1] 0.3445378 0.2964698
```

The printout from the analysis of spell duration (consecutive days with more than 35 degrees) reveals that a number of stations still had gaps of missing data. To deal with this in an ad hoc manner (it's difficult to estimate the heatwave durations with missing data), a linear interpolation in time was used to fill those gaps. This is a step that will introduce errors and an additional layer of uncertainties that will affect the whole analysis.

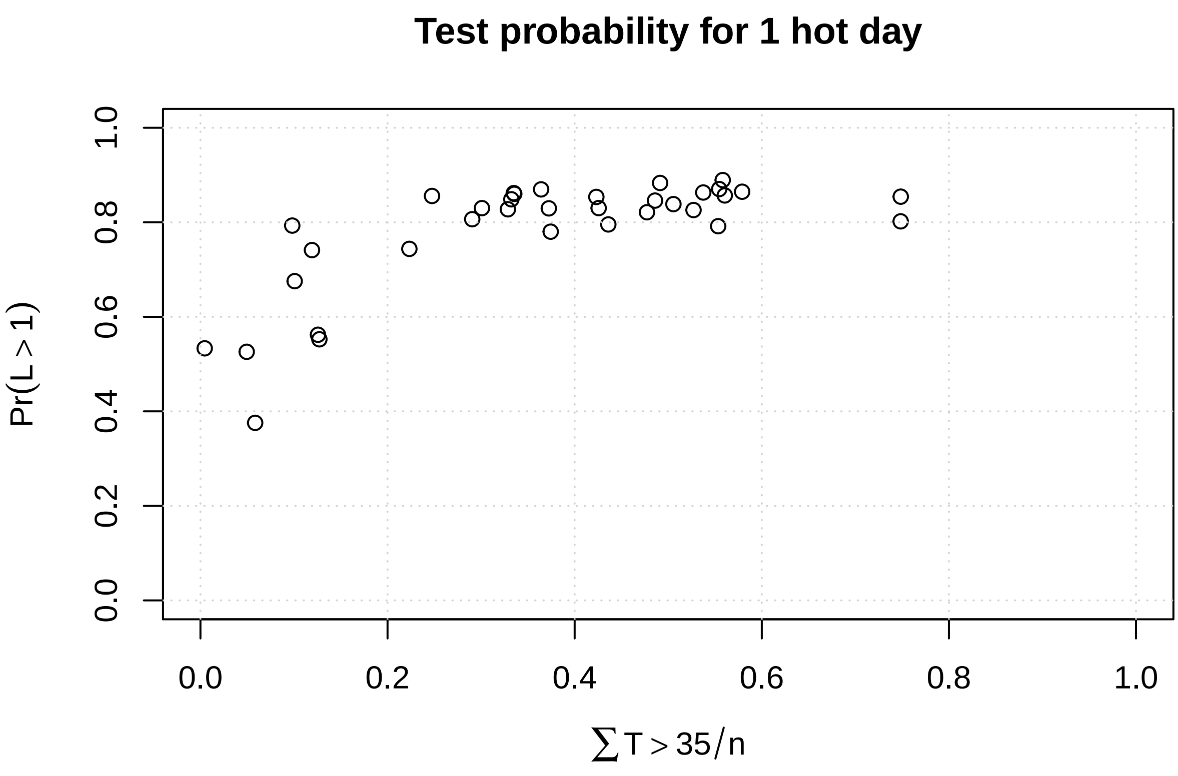
```
## The mean statistics for the winter-spring, February-March-April
tmax.fma <- aggregate(subset(Y,it=month.abb[2:4]),year,FUN='mean')
```

```
## Warning in sqrt(coredata(n) - 1): NaNs produced
```

```
lws.fma <- aggregate(subset(lws,it=month.abb[2:4]),year,FUN='mean')
```

```
## Warning in sqrt(coredata(n) - 1): NaNs produced
```

```
rng.tmax.fma <- range(c(tmax.fma),na.rm=TRUE) ## Sanity check
rng.lws.fma <- range(c(lws.fma),na.rm=TRUE) ## Sanity check
## Test geometric distribution for 1 day - same as the frequency of hot days?
y <- 1-pgeom(0,1/apply(lws.fma,2,'mean',na.rm=TRUE))
x <- apply(coredata(subset(Y,it=month.abb[2:4])),2,function(x) sum(x > 35,na.rm=TRUE)/sum(is.finite(x))
plot(x,y,xlim=c(0,1),ylim=c(0,1),
     xlab=expression(sum(T > 35)/n),ylab=expression(Pr(L>1)),main='Test probability for 1 hot day')
grid()
```



The results of the test for the probability for one day or more with maximum daily temperature exceeding $35°C$ against the observed frequency of such hot days suggests that the estimated probability assuming a geometric distribution gives higher values that are less sensitiveto the location differences. The probabilities here are the probability of heatwaves per season. The two estimates differ because the former takes heatwaves greater than 1 day as one case where the length is greater than zero, and therefore the two are not entirely comparable. However, based on this, the observed frequency on the x-axis is expected to be higher than the probability on the y-axis.

Seasonal aggregates

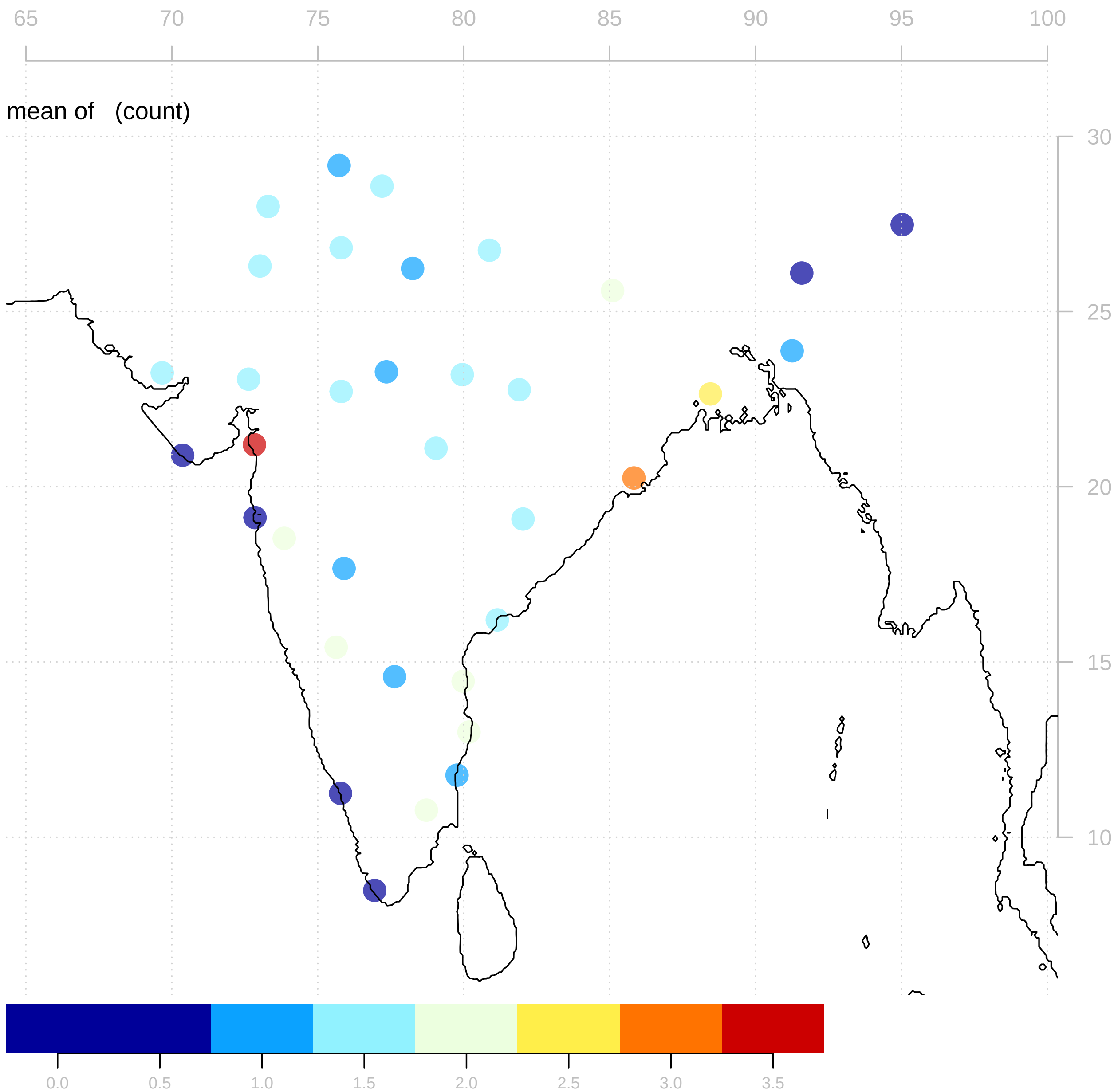
We want to use seasonal statistics of the temperature and duration of hot spells for downscaling the hot spell statistics. Here `lws.fma` (also referred to as $\overline{L_H}$ in the main paper) refers to the mean length of warm spells for February-March-April, and the probability of five-day spells are estimated from this using the geometric distribution. Aggregates over seasons are expected to be more resistant to errors in the data and caused by the interpolation than for the individual events.

```
## There are some missing data which will cause some technical problems in the analysis
## but these are only a few data points, and we can interpolate their values in order to
## get around the stumbling blocks that the missing values cause. The function 'pcafill' makes
## use of a spatio-temporal covariance matrix for filling in the gaps.
tmax.fma <- pcafill(tmax.fma)
lws.fma <- pcafill(lws.fma)
## Fix suspect data caused by interpolation
coredata(tmax.fma)[coredata(tmax.fma) > max(rng.tmax.fma)] <-  max(rng.tmax.fma)
coredata(tmax.fma)[coredata(tmax.fma) < min(rng.tmax.fma)] <-  min(rng.tmax.fma)
coredata(lws.fma)[coredata(lws.fma) > max(rng.lws.fma)] <-  max(rng.lws.fma)
coredata(lws.fma)[coredata(lws.fma) < min(rng.lws.fma)] <-  min(rng.lws.fma)
```

There were gaps of missing data also for the aggregated data, and the PCA required complete datasets with no missing data. We used a PCA-based strategy to fill in the gaps through the use of PCA applied to subsets (blocks of the data matrix) of the data with complete coverage, and regression to fill in the gaps. This is explained in Benestad et al (2015) 'On using principal components to represent stations in empirical–statistical downscaling' https://www.tandfonline.com/doi/full/10.3402/tellusa.v67.28326.

We need to assess the frequency of 'heatwaves' - defined events with $T_{max} > 35°$C.

```
## Use without pcafill - the number of hot events
lws5d <- subset(lws,it=month.abb[2:4]); x <- coredata(lws5d); x[x < 5] <- NA; x -> coredata(lws5d)
nh <- aggregate(lws5d,year,FUN='count')
attr(nh,'variable') <- 'heatwave-frequency'
attr(nh,'unit') <- 'count'
map(nh,FUN='mean',new=FALSE)
```




```
mnh <- round(apply(coredata(nh),2,'mean',na.rm=TRUE),2)
mTx <- round(apply(coredata(tmax.fma),2,'mean',na.rm=TRUE),2)
write.table(cbind(loc(lws),alt(lws),mTx,mnh),sep='  & ',eol='  \\\\  \n',quote=FALSE,row.names=FALSE)
```

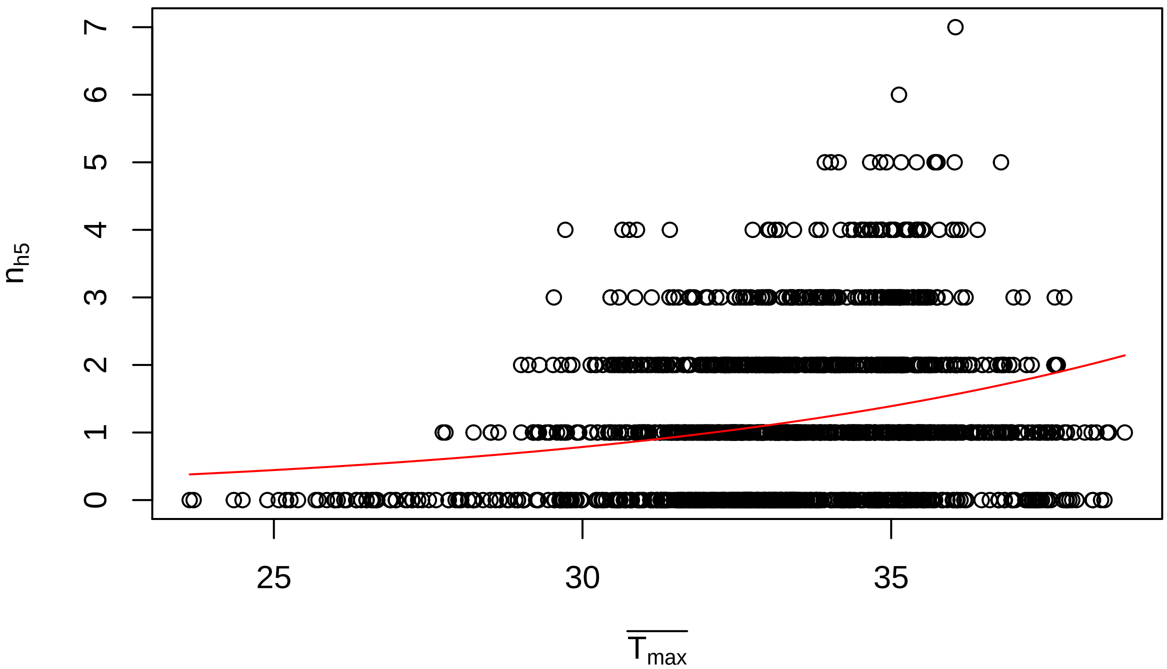
```
##   &   & mTx  & mnh  \\
## PBO ANANTAPUR  & 364  & 37.23  & 0.95  \\
## MACHILIPATNAM  & 3  & 33.26  & 1.16  \\
## NELLORE  & 20  & 35.33  & 1.88  \\
## GAUHATI  & 54  & 29.38  & 0.19  \\
## DIBRUGARH/MOHANBAR  & 111  & 26.41  & 0.02  \\
## PATNA  & 60  & 32.37  & 1.7  \\
## AHMADABAD  & 55  & 35.28  & 1.09  \\
## VERAVAL  & 8  & 31.37  & 0.44  \\
## BHUJ-RUDRAMATA  & 80  & 35.07  & 1.23  \\
## SURAT  & 12  & 35.01  & 3.3  \\
```

```
## HISSAR  & 221  & 31.29  & 0.98  \\
## GADAG   & 650  & 35.28  & 1.65  \\
## KOZHIKODE   & 5  & 33.28  & 0.37  \\
## THIRUVANANTHAPURAM  & 64  & 33.24  & 0.07  \\
## JAGDALPUR   & 553  & 34.81  & 1.42  \\
## PENDRA ROAD   & 625  & 32.27  & 1.23  \\
## GWALIOR   & 207  & 32.56  & 0.88  \\
## INDORE  & 567  & 34.34  & 1.07  \\
## JABALPUR  & 393  & 33.41  & 1.12  \\
## BHOPAL/BAIRAGARH  & 523  & 33.61  & 1  \\
## BOMBAY/SANTACRUZ  & 14  & 32.53  & 0.42  \\
## NAGPUR SONEGA  & 310  & 35.7  & 1.05  \\
## POONA   & 559  & 35.29  & 1.86  \\
## SHOLAPUR  & 479  & 37.52  & 0.81  \\
## BHUBANE  & 46  & 35.08  & 2.58  \\
## BIKANER   & 224  & 32.6  & 1.44  \\
## JAIPUR/SA  & 390  & 31.9  & 1.09  \\
## JODHPUR  & 224  & 33.51  & 1.14  \\
## CUDDALO  & 12  & 32.41  & 0.74  \\
## MADRAS/MINAMBAKKAM  & 16  & 33.71  & 1.81  \\
## TIRUCHCHIRAPALLI  & 88  & 35.57  & 1.6  \\
## AGARTALA  & 16  & 31.56  & 0.91  \\
## NEW DELHI/S  & 216  & 30.27  & 1.14  \\
## LUCKNOW/AMAUSI  & 128  & 32.46  & 1.23  \\
## CALCUTTA/DUM DUM  & 6  & 33.02  & 2.19  \\
```

We see that there is typically one or less five-day heatwave per February-March-April season in most locations. One exception is Surat with an average of 6 events each season. There is also a tendency of higher numbers along the southeastern coast, however, none of these are important places for wheat crops. The mean frequency is about 2 in the interior northern part on our map.

Compare the number of events with the mean temperature, which is expected to approximately follow a Poisson distribution.

```r
mtmax <- colMeans(coredata(tmax.fma),na.rm=TRUE)
srt <- order(mtmax)

## Use GLM and assume a Poisson distribution
cal.nevents <- data.frame(y=c(coredata(nh)),x=c(coredata(tmax.fma)))
fit.nh.glm <- glm(y ~ x, data=cal.nevents,family='poisson')

plot(cal.nevents$x,cal.nevents$y,
     xlab=expression(bar(T[max])),ylab=expression(n[h5]))
#lines(cal.nevents$x,exp(predict(fit.nh)),col='red')
## OLR and not GLM:
srt <- order(cal.nevents$x)
lines(cal.nevents$x[srt],exp(predict(fit.nh.glm))[srt],col='red')
```



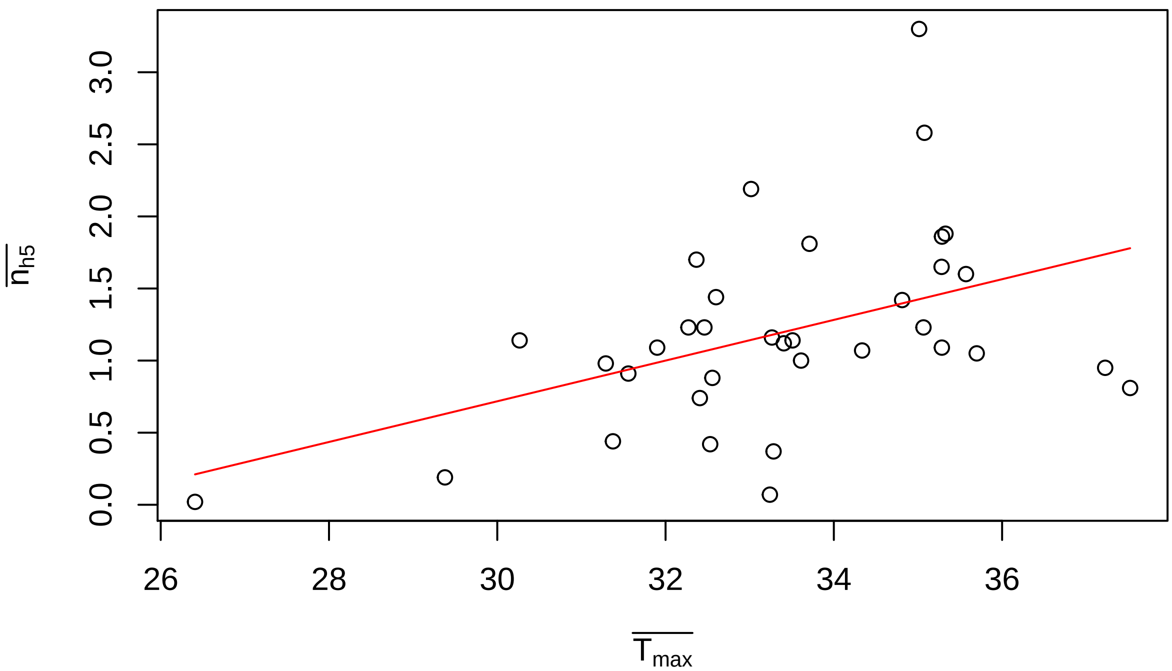

```r
print(summary(fit.nh.glm))
```

```
##
## Call:
## glm(formula = y ~ x, family = "poisson", data = cal.nevents)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0311  -1.3817  -0.1929   0.6382   3.1772
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.66468    0.37212  -9.848   <2e-16 ***
## x            0.11412    0.01094  10.432   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2035.0  on 1504  degrees of freedom
## Residual deviance: 1920.2  on 1503  degrees of freedom
## AIC: 4293.3
##
## Number of Fisher Scoring iterations: 5
```

Compare the {*mean number*} *of events* $\overline{n_{h5}}$ *with the mean temperature, which is expected to converge to a normal distribution with increasing sample size according to the central limit theorem.*

```r
mtmax <- colMeans(coredata(tmax.fma),na.rm=TRUE)
srt <- order(mtmax)

## The scatter seems to work as well with an ordinary linear model (OLR)
## Use an ordinary linear model since it's simpler than the GLM making use of
## the mean number of events rather than an integer number.
```

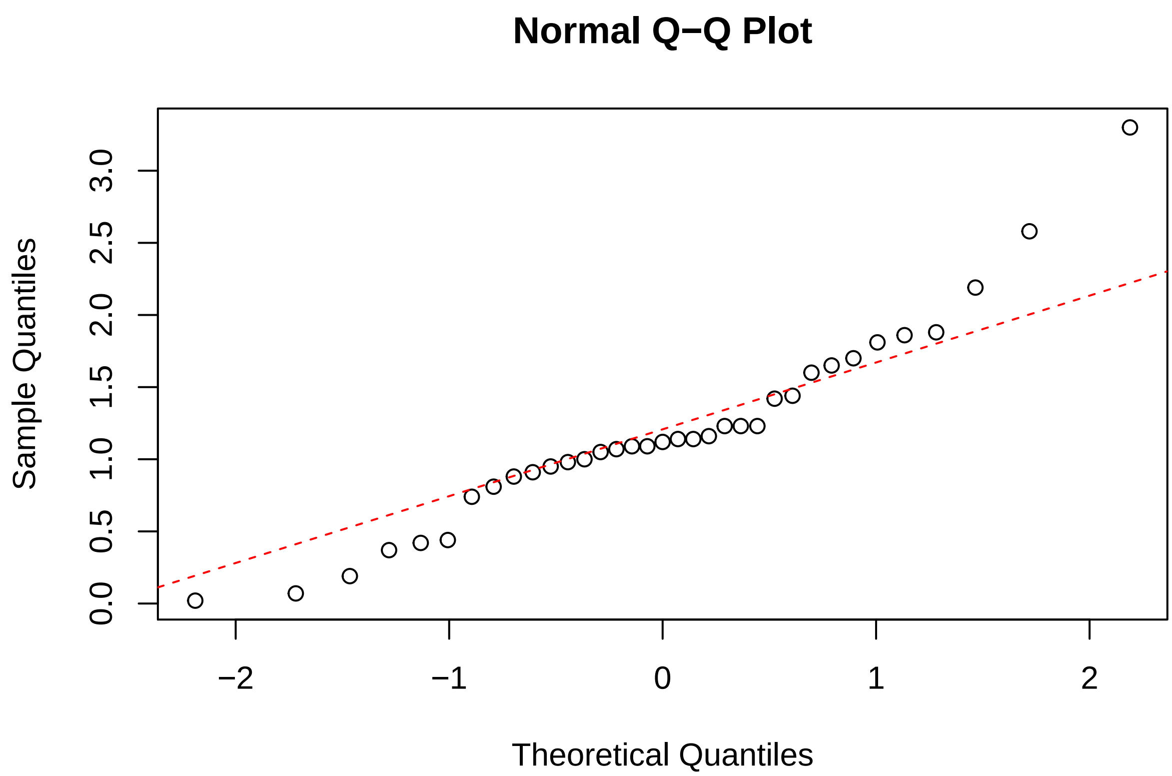
```r
cal.nevents <- data.frame(y=c(coredata(mnh)[srt]),x=c(mtmax[srt]))
fit.nh <- lm(y ~ x, data=cal.nevents)
plot(cal.nevents$x,cal.nevents$y,
     xlab=expression(bar(T[max])),ylab=expression(bar(n[h5])))
#lines(cal.nevents$x,exp(predict(fit.nh)),col='red')
## OLR and not GLM:
lines(cal.nevents$x,predict(fit.nh),col='red')
```




```r
print(summary(fit.nh))
```

```
##
## Call:
## lm(formula = y ~ x, data = cal.nevents)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10498 -0.34546 -0.02757  0.27351  1.87473
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.51602    1.60581  -2.190  0.03573 *
## x            0.14113    0.04802   2.939  0.00597 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6127 on 33 degrees of freedom
## Multiple R-squared:  0.2074, Adjusted R-squared:  0.1834
## F-statistic: 8.636 on 1 and 33 DF,  p-value: 0.005974
```

```r
qqnorm(coredata(mnh))
qqline(coredata(mnh),col='red',lty=2)
```



We see that there is an indication that the mean number of five-day heatwaves $\overline{n_{h5}}$ is influenced by the mean daily maximum temperature for the same season $\overline{T_{[}max]}$.

Estimate the probability (%) of at least one heatwave, assuming it is a stochastic process:

```
pr.heatwave <- 1 - ppois(0,lambda=coredata(mnh))
print(round(100*pr.heatwave,2))
```

```
##      PBO ANANTAPUR      MACHILIPATNAM             NELLORE
##              61.33              68.65               84.74
##            GAUHATI  DIBRUGARH/MOHANBAR              PATNA
##              17.30               1.98               81.73
##          AHMADABAD            VERAVAL       BHUJ-RUDRAMATA
##              66.38              35.60               70.77
##              SURAT             HISSAR               GADAG
##              96.31              62.47               80.80
##          KOZHIKODE  THIRUVANANTHAPURAM           JAGDALPUR
##              30.93               6.76               75.83
##        PENDRA ROAD             GWALIOR              INDORE
##              70.77              58.52               65.70
##           JABALPUR   BHOPAL/BAIRAGARH    BOMBAY/SANTACRUZ
##              67.37              63.21               34.30
##      NAGPUR SONEGA               POONA            SHOLAPUR
##              65.01              84.43               55.51
##            BHUBANE             BIKANER           JAIPUR/SA
##              92.42              76.31               66.38
##            JODHPUR            CUDDALO MADRAS/MINAMBAKKAM
##              68.02              52.29               83.63
##   TIRUCHCHIRAPALLI            AGARTALA          NEW DELHI/S
##              79.81              59.75               68.02
##     LUCKNOW/AMAUSI   CALCUTTA/DUM DUM
```

```
##               70.77              88.81
```

Relationship between the mean temperature and duration of heat waves

The following chunks of R-code show how we calibrated the regression model to quantify the statistical link between the mean temperature and the mean duration.

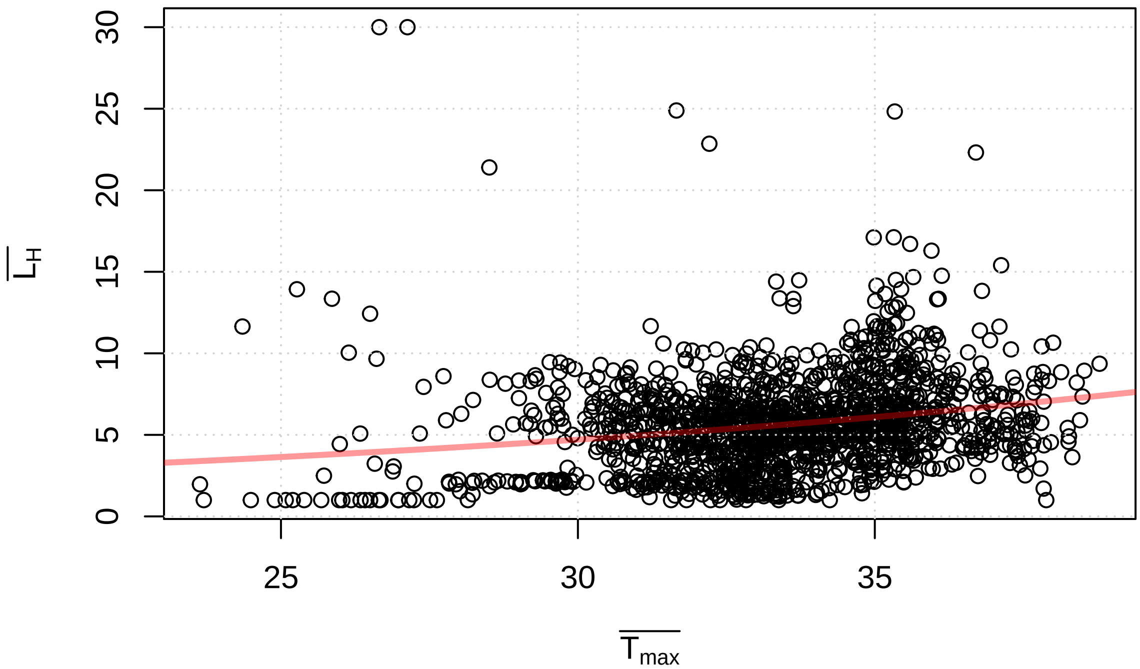
```
## Figure 2.
print('Figure 2')
```

```
## [1] "Figure 2"
```

```
## Synchronise the two records: mean spell length and mean temperature
xy <- merge(round(zoo(lws.fma)),zoo(tmax.fma),all=FALSE)
## In this data.frame, x is the mean heatwave duration and y is the mean temperature
cal.tmax.lws <- data.frame(x=c(coredata(tmax.fma)),y=c(coredata(lws.fma)))
#cal.tmax.lws <- data.frame(y=c(coredata(xy[,1:35])),x=c(coredata(xy)[,36:70]))
ok <- is.finite(cal.tmax.lws$x) & is.finite(cal.tmax.lws$y) & (cal.tmax.lws$y >= 0)
cal.tmax.lws <- cal.tmax.lws[ok,]
cal.tmax.lws.log <- data.frame(y=log(c(coredata(xy[,1:35]))),x=log(c(coredata(xy)[,36:70])))
ok <- is.finite(cal.tmax.lws.log$x) & is.finite(cal.tmax.lws.log$y)
cal.tmax.lws.log <- cal.tmax.lws.log[ok,]
summary(lm(y ~ x, data=cal.tmax.lws.log))
```

```
##
## Call:
## lm(formula = y ~ x, data = cal.tmax.lws.log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.91434 -0.32214  0.05976  0.36594  2.39297
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.4517     0.6808  -10.95   <2e-16 ***
## x             2.5769     0.1942   13.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5485 on 1503 degrees of freedom
## Multiple R-squared:  0.1049, Adjusted R-squared:  0.1043
## F-statistic: 176.1 on 1 and 1503 DF,  p-value: < 2.2e-16
```

```
fit <- glm(y ~ x, data=cal.tmax.lws,family='poisson')
#dev.new()
par(mar=c(5.1,5.1,4.1,2.1))
plot(cal.tmax.lws$x,cal.tmax.lws$y,xlab=expression(bar(T[max])),ylab=expression(bar(L[H])))
pre <- data.frame(x=seq(min(tmax,na.rm=TRUE),max(tmax,na.rm=TRUE),by=0.1))
lines(pre$x,exp(predict(fit,newdata=pre)),col=rgb(1,0,0,0.4),lwd=3)
grid()
```




```
#dev.copy2pdf(file='fig2.pdf')
```

Try the suggestion from reviewer:

```
## Figure 2.
print('Figure 2z')
```

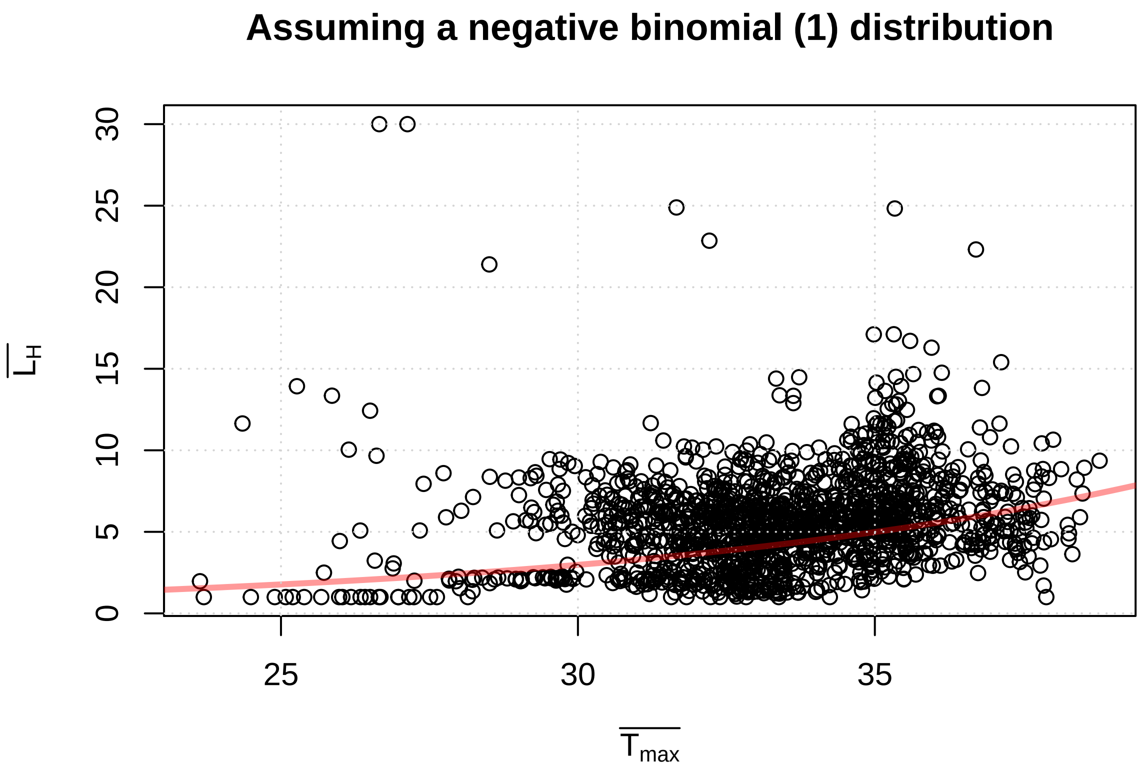
```
## [1] "Figure 2z"
```

```
## Synchronise the two records: mean spell length and mean temperature
cal.tmax.lws <- data.frame(x=c(coredata(tmax.fma)),y=1/c(coredata(lws.fma)))

fit <- glm(y ~ x, data=cal.tmax.lws,family=negative.binomial(1))
print(summary(fit))
```

```
##
## Call:
## glm(formula = y ~ x, family = negative.binomial(1), data = cal.tmax.lws)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9656  -0.7952  -0.7526   0.8326   1.2618
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.009330   0.213234   9.423   <2e-16 ***
## x           -0.103313   0.006464 -15.982   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.0770043)
##
##     Null deviance: 1002.47  on 1504  degrees of freedom
## Residual deviance:  982.73  on 1503  degrees of freedom
```

```
## AIC: 1842.9
##
## Number of Fisher Scoring iterations: 4
```

```
#dev.new()
par(mar=c(5.1,5.1,4.1,2.1))
plot(cal.tmax.lws$x,1/cal.tmax.lws$y,xlab=expression(bar(T[max])),ylab=expression(bar(L[H])),
     main='Assuming a negative binomial (1) distribution')
pre <- data.frame(x=seq(min(tmax,na.rm=TRUE),max(tmax,na.rm=TRUE),by=0.1))
lines(pre$x,1/exp(predict(fit,newdata=pre)),col=rgb(1,0,0,0.4),lwd=3)
grid()
```




```
#dev.copy2pdf(file='fig2z.pdf')
```

Also try the relation using the average numbers for each for each site intead of average for each season (i.e. fewer data points, but more aggregated data):

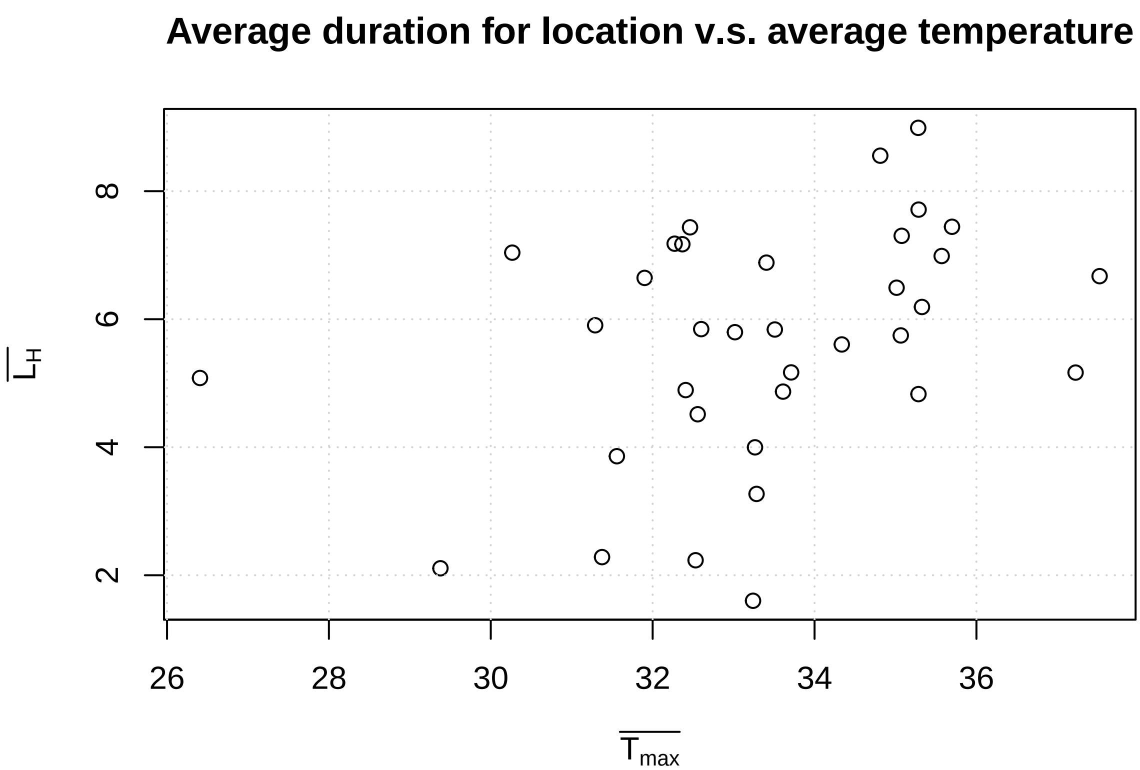
```
## Figure 2.
print('Figure 2x')
```

```
## [1] "Figure 2x"
```

```
## Synchronise the two records: mean spell length and mean temperature
xy <- merge(round(zoo(lws.fma)),zoo(tmax.fma),all=FALSE)
## In this data.frame, x is the mean heatwave duration and y is the mean temperature
cal.tmax.lws1 <- data.frame(x=apply(coredata(tmax.fma),2,'mean',na.rm=TRUE),
                            y=apply(coredata(lws.fma),2,'mean',na.rm=TRUE))
#cal.tmax.lws <- data.frame(y=c(coredata(xy[,1:35])),x=c(coredata(xy)[,36:70]))
ok <- is.finite(cal.tmax.lws1$x) & is.finite(cal.tmax.lws1$y) & (cal.tmax.lws1$y >= 0)
cal.tmax.lws1 <- cal.tmax.lws1[ok,]
fit1 <- lm(y ~ x, data=cal.tmax.lws1)
print(summary(fit1))
```

```
##
## Call:
## lm(formula = y ~ x, data = cal.tmax.lws1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9968 -1.0496  0.1583  1.3451  2.7677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.5504     4.5246  -1.006   0.3219
## x             0.3053     0.1353   2.256   0.0308 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.726 on 33 degrees of freedom
## Multiple R-squared:  0.1337, Adjusted R-squared:  0.1074
## F-statistic: 5.091 on 1 and 33 DF,  p-value: 0.0308
```

```
#dev.new()
par(mar=c(5.1,5.1,4.1,2.1))
plot(cal.tmax.lws1$x,cal.tmax.lws1$y,xlab=expression(bar(T[max])),ylab=expression(bar(L[H])),
     main='Average duration for location v.s. average temperature')
pre1 <- data.frame(x=seq(min(tmax,na.rm=TRUE),max(tmax,na.rm=TRUE),by=0.1))
lines(pre$x,exp(predict(fit1,newdata=pre1)),col=rgb(1,0,0,0.4),lwd=3)
grid()
```




```
#dev.copy2pdf(file='fig2x.pdf')
```

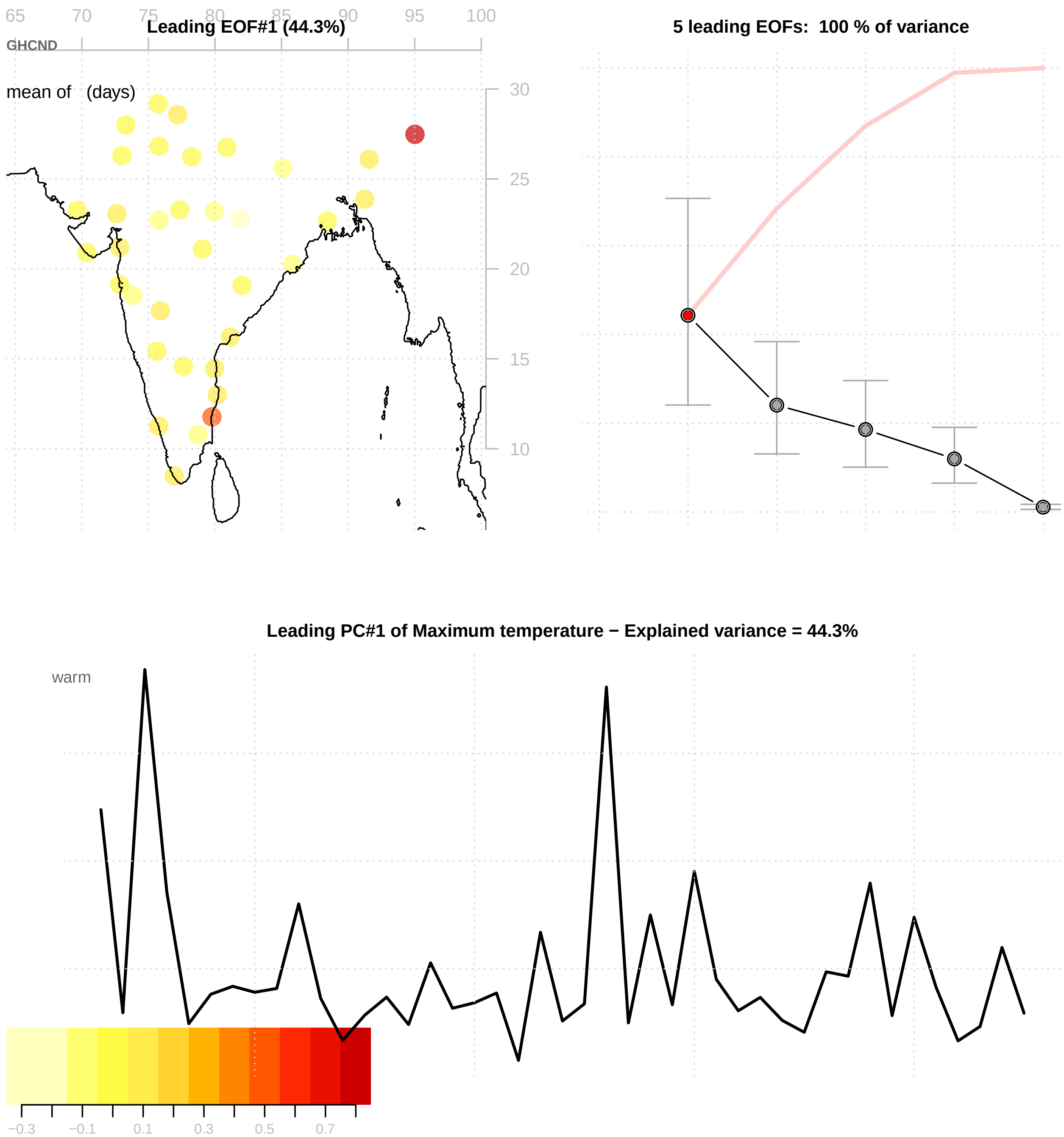
Preparing the predictand

The next part describes how we set up the empirical-statistical downscaling. We used PCA to represent the predictands for several reasons. Because the station records contained redundant information, we could get away with downscaling a few PCs rather than all the stations, and hence save time. It's more efficient. Furthermore, the PCA maintains the observed covariance structure and the correlation between temperatures at different locations. Another reason for using PCA was that it emhpasises the common signal in the data records and supresses noise and errors, which benefits the downscaling that tries to identify the connection to the large-scale conditions. More details about using PCAs to represent the predictands in empirical-statistical downscaling can be found in Benestad et al. (2015) http://dx.doi.org/10.3402/tellusa.v67.28326.

The nature of heatwave duration

The following figure shows what the leading PCA looks like for the mean heatwave duration $\overline{L_H}$*, however, this was not used as predictand in the downscaling - it's shown just to get an idea of where there are large year-to-year variations in the duration of heatwaves with temperatures greater than 35°:*

```
##PCA for the mean length of warm spells:
pca.lws <- PCA(lws.fma,n=5)
plot(pca.lws,new=FALSE)
```



The sites in the interior northern India are associated with greater weights for the leading PCA in connection with interannual variations in the mean spell duration. The five leading modes accounted for 100% of the variance in the seasonal mean heatwave duration.

A sanity check is to look at the maximum and minimum values of the mean lengths of heatwaves:

```
print(summary(coredata(lws)))
```

```
##  PBO ANANTAPUR    MACHILIPATNAM       NELLORE          GAUHATI
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 2.000   Median : 2.000   Median : 3.000   Median : 1.000
##  Mean   : 3.921   Mean   : 4.456   Mean   : 6.082   Mean   : 1.923
```

```
##  3rd Qu.: 5.000   3rd Qu.: 6.000   3rd Qu.: 8.000   3rd Qu.: 2.000
##  Max.   :30.000   Max.   :30.000   Max.   :30.000   Max.   :13.000
##  NA's   :7716     NA's   :7556     NA's   :7533     NA's   :7876
##  DIBRUGARH/MOHANBAR     PATNA          AHMADABAD         VERAVAL
##  Min.   :1.000      Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:1.000      1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median :1.000      Median : 3.000   Median : 3.000   Median : 2.000
##  Mean   :1.962      Mean   : 5.038   Mean   : 5.302   Mean   : 2.598
##  3rd Qu.:2.000      3rd Qu.: 6.000   3rd Qu.: 7.000   3rd Qu.: 3.000
##  Max.   :7.000      Max.   :29.000   Max.   :30.000   Max.   :15.000
##  NA's   :8004       NA's   :7658     NA's   :7804     NA's   :7748
##  BHUJ-RUDRAMATA      SURAT            HISSAR           GADAG
##  Min.   : 1.00   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.00   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 2.000
##  Median : 3.00   Median : 3.000   Median : 4.000   Median : 4.000
##  Mean   : 5.77   Mean   : 4.917   Mean   : 7.024   Mean   : 7.349
##  3rd Qu.: 7.00   3rd Qu.: 6.000   3rd Qu.: 9.250   3rd Qu.:11.000
##  Max.   :30.00   Max.   :30.000   Max.   :30.000   Max.   :30.000
##  NA's   :7711    NA's   :7443     NA's   :7746     NA's   :7997
##    KOZHIKODE      THIRUVANANTHAPURAM   JAGDALPUR       PENDRA ROAD
##  Min.   : 1.000   Min.   :1.000      Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.:1.000      1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 2.000   Median :1.000      Median : 3.000   Median : 3.000
##  Mean   : 3.461   Mean   :1.553      Mean   : 6.758   Mean   : 5.814
##  3rd Qu.: 3.000   3rd Qu.:2.000      3rd Qu.:10.000   3rd Qu.: 8.000
##  Max.   :30.000   Max.   :6.000      Max.   :30.000   Max.   :28.000
##  NA's   :8071     NA's   :8068       NA's   :7973     NA's   :8002
##     GWALIOR          INDORE          JABALPUR     BHOPAL/BAIRAGARH
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.00   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.00   1st Qu.: 1.000
##  Median : 3.000   Median : 3.000   Median : 3.00   Median : 2.000
##  Mean   : 5.026   Mean   : 4.428   Mean   : 4.94   Mean   : 4.695
##  3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 6.00   3rd Qu.: 6.000
##  Max.   :28.000   Max.   :28.000   Max.   :30.00   Max.   :29.000
##  NA's   :7773     NA's   :7953     NA's   :7987    NA's   :7946
##  BOMBAY/SANTACRUZ NAGPUR SONEGA        POONA          SHOLAPUR
##  Min.   : 1.0     Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.0     1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.0     Median : 3.000   Median : 3.000   Median : 2.000
##  Mean   : 2.4     Mean   : 4.608   Mean   : 6.512   Mean   : 3.961
##  3rd Qu.: 3.0     3rd Qu.: 6.000   3rd Qu.: 9.000   3rd Qu.: 4.000
##  Max.   :16.0     Max.   :29.000   Max.   :30.000   Max.   :26.000
##  NA's   :7706     NA's   :7937     NA's   :7945     NA's   :7857
##     BHUBANE          BIKANER         JAIPUR/SA         JODHPUR
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000
##  Median : 3.000   Median : 5.000   Median : 4.000   Median : 4.000
##  Mean   : 5.477   Mean   : 7.166   Mean   : 6.589   Mean   : 6.647
##  3rd Qu.: 7.000   3rd Qu.:10.000   3rd Qu.: 8.000   3rd Qu.: 9.000
##  Max.   :30.000   Max.   :30.000   Max.   :30.000   Max.   :29.000
##  NA's   :7727     NA's   :7846     NA's   :7834     NA's   :7805
##     CUDDALO       MADRAS/MINAMBAKKAM TIRUCHCHIRAPALLI    AGARTALA
##  Min.   : 1.000   Min.   : 1.000     Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 1.000     1st Qu.: 2.000   1st Qu.: 1.000
```

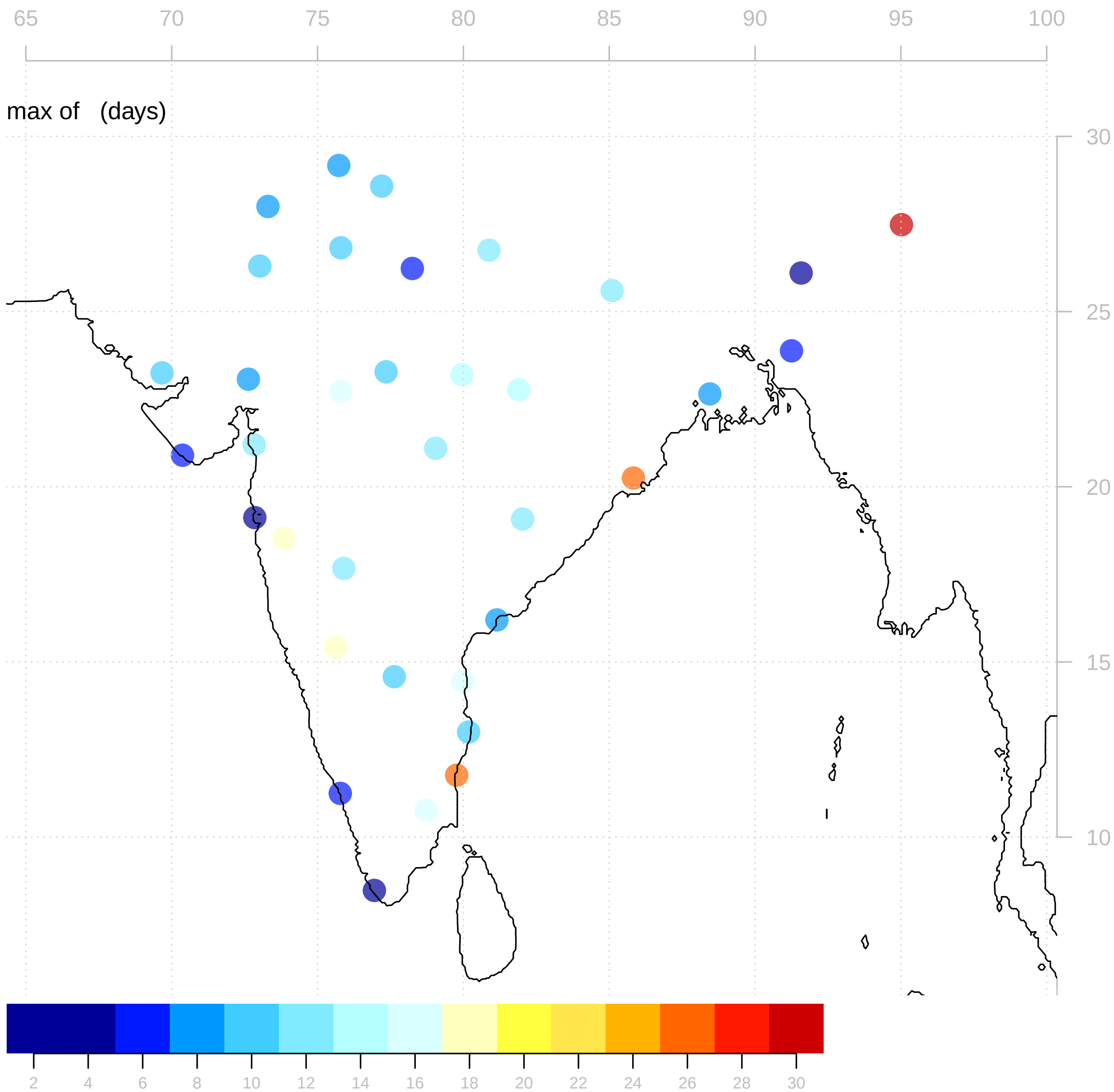
```
## Median : 3.000   Median : 3.000     Median : 4.000   Median : 2.000
## Mean   : 5.107   Mean   : 5.015     Mean   : 6.706   Mean   : 2.844
## 3rd Qu.: 6.000   3rd Qu.: 6.000     3rd Qu.: 9.000   3rd Qu.: 3.000
## Max.   :30.000   Max.   :29.000     Max.   :30.000   Max.   :22.000
## NA's   :7535     NA's   :7429       NA's   :7659     NA's   :7956
##  NEW DELHI/S     LUCKNOW/AMAUSI  CALCUTTA/DUM DUM
## Min.   : 1.000   Min.   : 1.00   Min.   : 1.000
## 1st Qu.: 1.000   1st Qu.: 2.00   1st Qu.: 1.000
## Median : 3.000   Median : 3.00   Median : 2.000
## Mean   : 5.162   Mean   : 5.35   Mean   : 4.256
## 3rd Qu.: 6.000   3rd Qu.: 7.00   3rd Qu.: 5.000
## Max.   :30.000   Max.   :30.00   Max.   :28.000
## NA's   :7577     NA's   :7769    NA's   :7582
```

```
print(summary(coredata(lws.fma)))
```

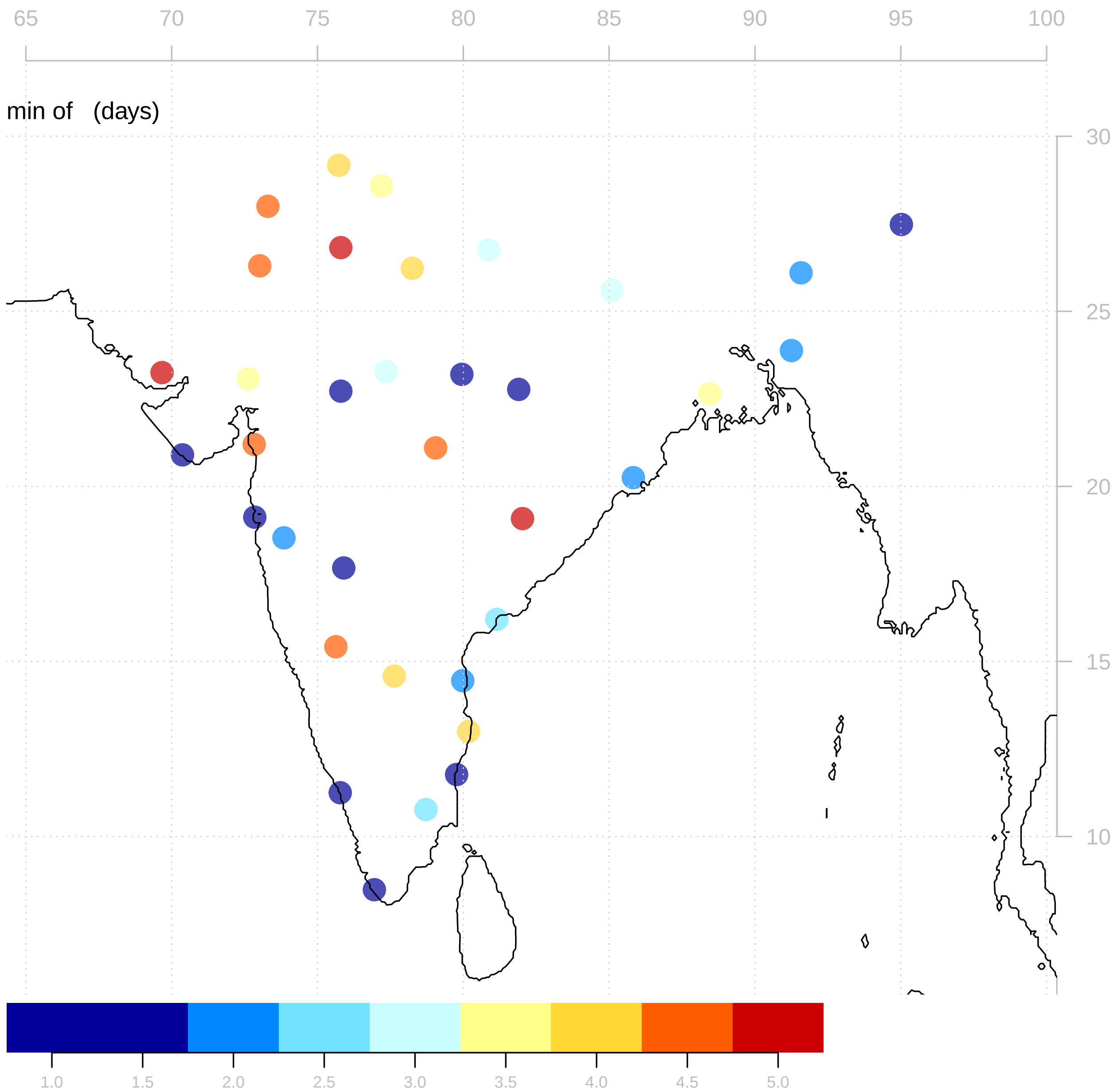
```
## PBO ANANTAPUR   MACHILIPATNAM      NELLORE          GAUHATI
## Min.   :3.631   Min.   :2.021   Min.   : 1.948   Min.   :1.854
## 1st Qu.:4.469   1st Qu.:3.207   1st Qu.: 3.660   1st Qu.:2.072
## Median :5.031   Median :3.908   Median : 6.067   Median :2.121
## Mean   :5.166   Mean   :3.998   Mean   : 6.190   Mean   :2.109
## 3rd Qu.:5.652   3rd Qu.:4.645   3rd Qu.: 7.444   3rd Qu.:2.172
## Max.   :8.209   Max.   :6.542   Max.   :14.148   Max.   :2.261
## DIBRUGARH/MOHANBAR     PATNA          AHMADABAD        VERAVAL
## Min.   : 1.000     Min.   : 2.753   Min.   :3.289   Min.   :1.184
## 1st Qu.: 1.000     1st Qu.: 6.654   1st Qu.:3.720   1st Qu.:1.887
## Median : 1.000     Median : 7.265   Median :4.630   Median :2.145
## Mean   : 5.081     Mean   : 7.171   Mean   :4.830   Mean   :2.284
## 3rd Qu.: 5.083     3rd Qu.: 8.057   3rd Qu.:5.869   3rd Qu.:2.499
## Max.   :30.000     Max.   :10.034   Max.   :7.390   Max.   :4.488
## BHUJ-RUDRAMATA      SURAT            HISSAR           GADAG
## Min.   :4.623   Min.   : 4.447   Min.   :3.508   Min.   : 4.290
## 1st Qu.:5.150   1st Qu.: 5.280   1st Qu.:5.557   1st Qu.: 6.645
## Median :5.591   Median : 6.199   Median :5.979   Median : 8.795
## Mean   :5.746   Mean   : 6.491   Mean   :5.905   Mean   : 8.989
## 3rd Qu.:6.051   3rd Qu.: 7.330   3rd Qu.:6.324   3rd Qu.:11.110
## Max.   :8.181   Max.   :11.372   Max.   :6.834   Max.   :17.115
##   KOZHIKODE     THIRUVANANTHAPURAM   JAGDALPUR       PENDRA ROAD
## Min.   :1.000   Min.   :1.190      Min.   : 4.781   Min.   : 1.000
## 1st Qu.:2.575   1st Qu.:1.362      1st Qu.: 7.917   1st Qu.: 6.052
## Median :3.144   Median :1.512      Median : 8.744   Median : 6.741
## Mean   :3.270   Mean   :1.601      Mean   : 8.554   Mean   : 7.179
## 3rd Qu.:3.936   3rd Qu.:1.746      3rd Qu.: 9.481   3rd Qu.: 8.376
## Max.   :5.777   Max.   :2.443      Max.   :11.061   Max.   :13.340
##    GWALIOR          INDORE          JABALPUR      BHOPAL/BAIRAGARH
## Min.   :3.815   Min.   : 1.000   Min.   : 1.282   Min.   :2.871
## 1st Qu.:4.253   1st Qu.: 3.414   1st Qu.: 5.629   1st Qu.:3.986
## Median :4.493   Median : 4.993   Median : 6.614   Median :4.709
## Mean   :4.515   Mean   : 5.605   Mean   : 6.883   Mean   :4.868
## 3rd Qu.:4.763   3rd Qu.: 6.950   3rd Qu.: 8.776   3rd Qu.:5.592
## Max.   :5.368   Max.   :14.481   Max.   :13.372   Max.   :8.354
## BOMBAY/SANTACRUZ NAGPUR SONEGA        POONA           SHOLAPUR
## Min.   :1.271    Min.   : 4.388   Min.   : 1.918   Min.   : 1.000
## 1st Qu.:1.967    1st Qu.: 6.112   1st Qu.: 4.712   1st Qu.: 4.606
## Median :2.182    Median : 7.484   Median : 6.394   Median : 7.337
```

```
##  Mean   :2.234    Mean   : 7.443   Mean   : 7.711   Mean   : 6.672
##  3rd Qu.:2.493    3rd Qu.: 8.463   3rd Qu.:11.073   3rd Qu.: 8.593
##  Max.   :3.111    Max.   :11.639   Max.   :16.707   Max.   :11.399
##     BHUBANE          BIKANER        JAIPUR/SA         JODHPUR
##  Min.   : 1.802   Min.   :4.357   Min.   :4.591   Min.   :4.471
##  1st Qu.: 3.977   1st Qu.:5.483   1st Qu.:6.202   1st Qu.:4.991
##  Median : 6.003   Median :5.874   Median :6.900   Median :5.802
##  Mean   : 7.302   Mean   :5.845   Mean   :6.645   Mean   :5.838
##  3rd Qu.: 8.576   3rd Qu.:6.264   3rd Qu.:7.223   3rd Qu.:6.255
##  Max.   :24.827   Max.   :7.222   Max.   :8.098   Max.   :8.617
##     CUDDALO       MADRAS/MINAMBAKKAM TIRUCHCHIRAPALLI    AGARTALA
##  Min.   : 1.000   Min.   :3.689      Min.   : 2.087   Min.   :1.787
##  1st Qu.: 1.711   1st Qu.:4.696      1st Qu.: 4.843   1st Qu.:3.526
##  Median : 3.749   Median :4.961      Median : 6.883   Median :3.953
##  Mean   : 4.893   Mean   :5.168      Mean   : 6.987   Mean   :3.860
##  3rd Qu.: 5.956   3rd Qu.:5.426      3rd Qu.: 8.481   3rd Qu.:4.250
##  Max.   :24.892   Max.   :8.331      Max.   :15.402   Max.   :5.948
##   NEW DELHI/S    LUCKNOW/AMAUSI   CALCUTTA/DUM DUM
##  Min.   :3.112   Min.   : 2.939   Min.   :3.006
##  1st Qu.:5.571   1st Qu.: 6.265   1st Qu.:5.268
##  Median :7.510   Median : 7.209   Median :5.912
##  Mean   :7.039   Mean   : 7.435   Mean   :5.797
##  3rd Qu.:8.488   3rd Qu.: 9.247   3rd Qu.:6.541
##  Max.   :9.457   Max.   :11.617   Max.   :7.355
```

```r
map(lws.fma,FUN='max',new=FALSE)
```

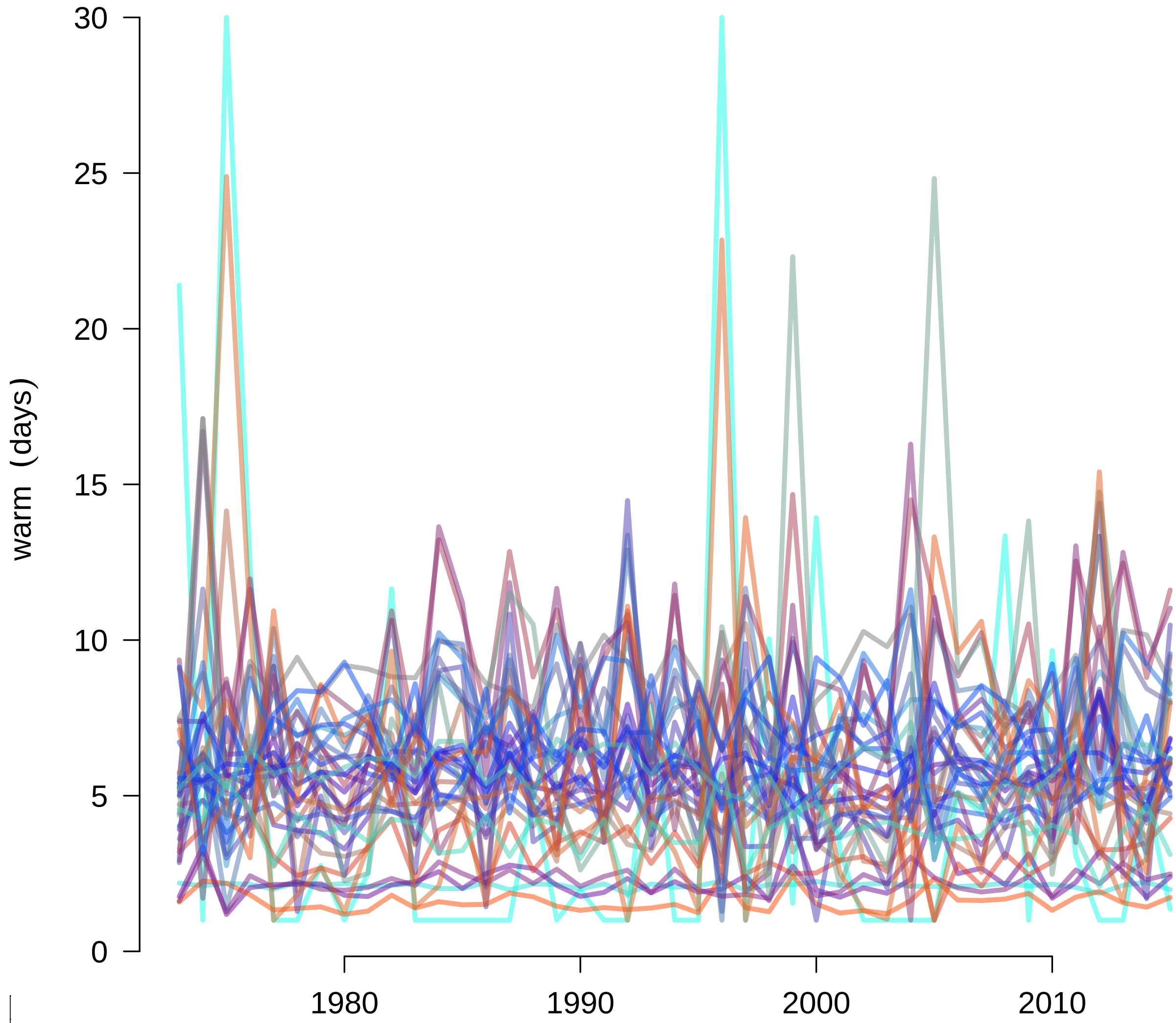



```
map(lws.fma,FUN='min',new=FALSE)
```



One site has negative minimum values for the mean length, but all spell lengths are positive. This is what the time series looks like:

```
plot(lws.fma,errorbar=FALSE,new=FALSE,map.show=FALSE)
```

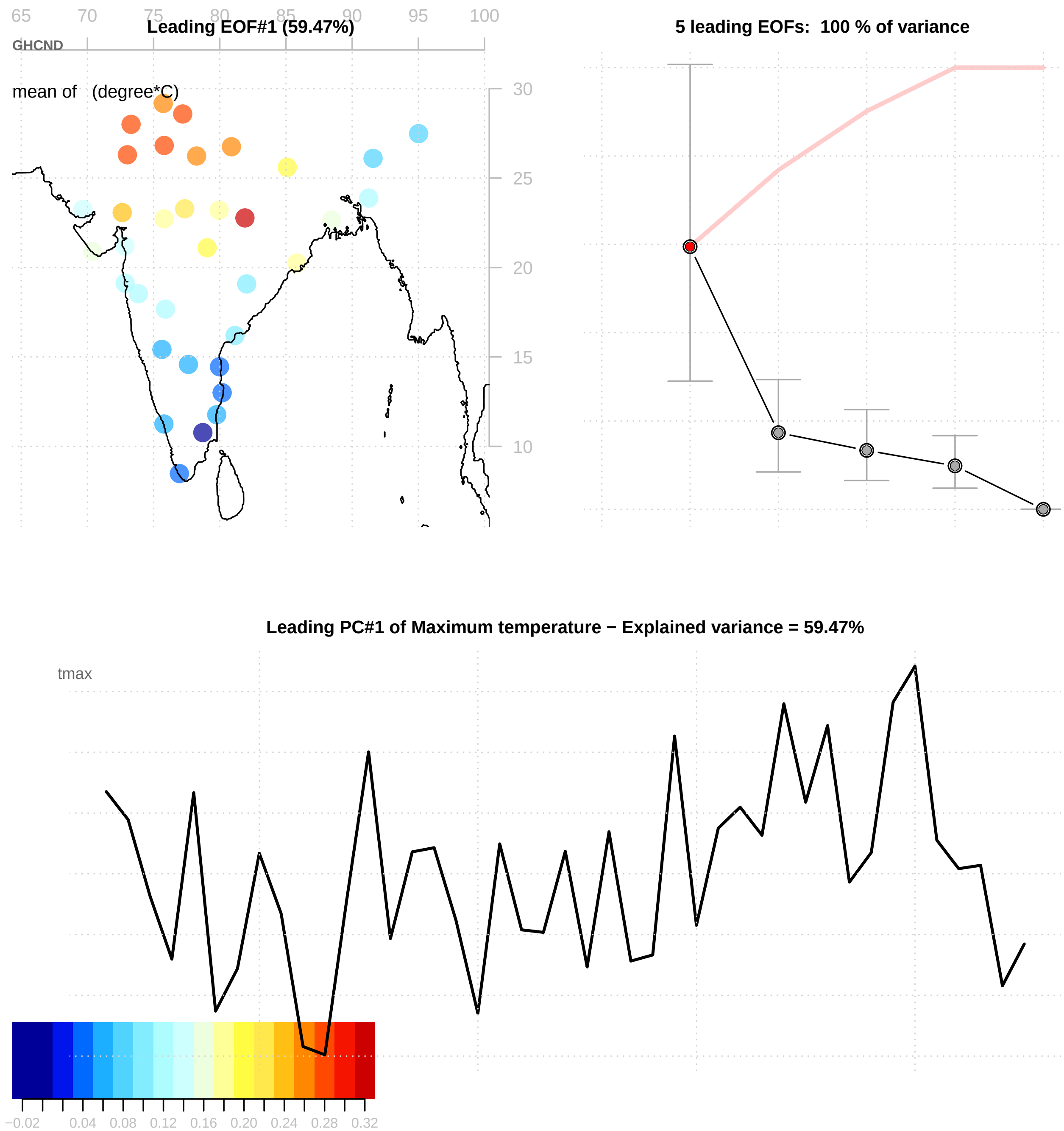


The reason for negative values in $\overline{L_H}$ is gaps of missing data and the use of `pcafill` *to fill in the gaps. Similar inaccuracies can be expected for the mean temperature $\overline{T[max]}$. This interpolation, however, does not add any new information and does not affect the PCAs used for the downscaling much other than weighting the internal information slightly differently - it merely makes the PCA possible by removing the stumbling blocks of missing data.*

Evaluation of the predictand data:

The next figure shows the leading PCA for the February-March-April mean daily maximum temperature $\overline{T[max]}$ which was used as predictand in the downscaling.

```
## PCA for the mean daily maximum temperature:
pca.tmax <- PCA(tmax.fma,n=5)
plot(pca.tmax,new=FALSE)
```



The five leading PCAs account for 100% of the variance and the leading PCA reveals a pattern with strongest weights in the interior northern India.

Downscaling seasonal mean of the daily maximum temperature

The following chuncks of R-code apply empirical-statistical downscaling to large multi-model ensembles. The first chunk downscales the simulations for the intermediate emission scenario RCP4.5 where total radiative forcing is stabilized before 2100:

```
print('Downscaling')
```

```
## [1] "Downscaling"
```

```
if (!file.exists("dse.tmax.india.rcp45.rda")) {
  T2m <- retrieve("~/data/air.mon.mean.nc",lon=c(65,95),lat=c(5,27))
  T2m <- aggregate(subset(T2m,it=month.abb[2:4]),year)
  dse.tmax.india.rcp45 <- DSensemble.pca(pca.tmax,predictor=T2m,biascorrect = TRUE,ip=1:4,it=month.abb[
  save(dse.tmax.india.rcp45,file="dse.tmax.india.rcp45.rda")
} else load("dse.tmax.india.rcp45.rda")
```

The results are saved locally, so the downscaling (which takes some time) needs to be done only once. Repeated runs with this script will be faster.

Downscale the simulations for the high emission scenario RCP8.5:

```
## ----dsensemblehigh---------------------------------------------------------
if (!file.exists("dse.tmax.india.rcp85.rda")) {
  T2m <- retrieve("~/data/air.mon.mean.nc",lon=c(65,95),lat=c(5,27))
  T2m <- aggregate(subset(T2m,it=month.abb[2:4]),year)
  dse.tmax.india.rcp85 <- DSensemble.pca(pca.tmax,predictor=T2m,rcp='rcp85',biascorrect = TRUE,ip=1:4,i
  save(dse.tmax.india.rcp85,file="dse.tmax.india.rcp85.rda")
} else load("dse.tmax.india.rcp85.rda")
```

Downscale the simulations for the low emission scenario RCP2.6:

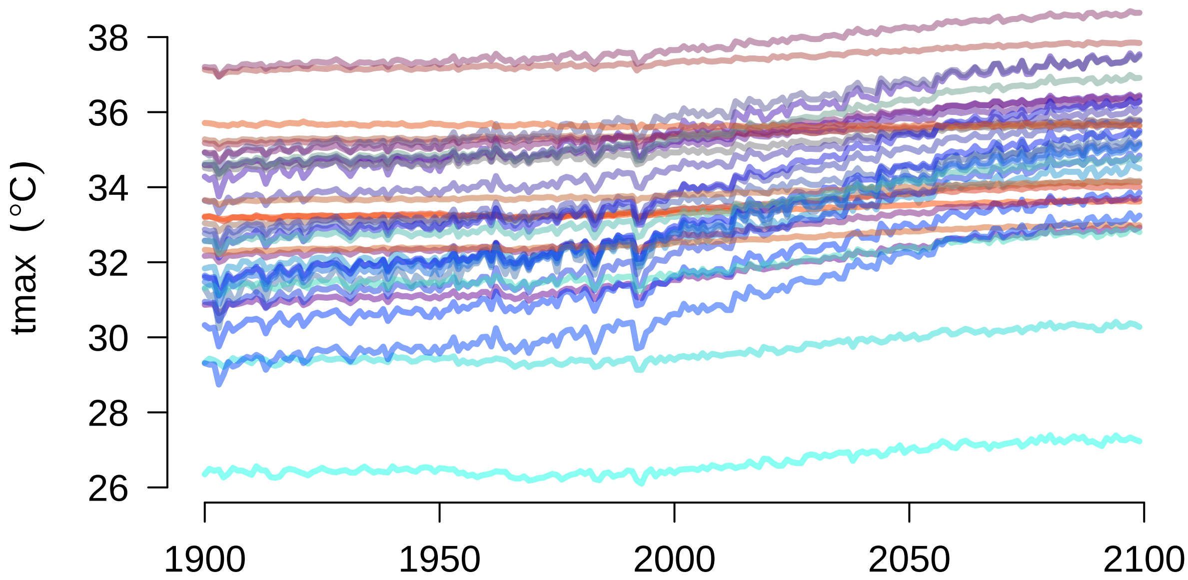
```
## ----dsensemblelow----------------------------------------------------------
if (!file.exists("dse.tmax.india.rcp26.rda")) {
  T2m <- retrieve("~/data/air.mon.mean.nc",lon=c(65,95),lat=c(5,27))
  T2m <- aggregate(subset(T2m,it=month.abb[2:4]),year)
  dse.tmax.india.rcp26 <- DSensemble.pca(pca.tmax,predictor=T2m,rcp='rcp26',biascorrect = TRUE,ip=1:4,i
  save(dse.tmax.india.rcp26,file="dse.tmax.india.rcp26.rda")
} else load("dse.tmax.india.rcp26.rda")
```

The downscaled results are stored in a compact and efficient way, making use of redundancy to save space. To make sense of these results, some postprocessing is needed to 'recover' information that resemble the original station records. For more information about data strategies, see Benestad et al (2017) 'A strategy to effectively make use of large volumes of climate data for climate change adaptation' https://www.sciencedirect.com/science/article/pii/S2405880717300043.

Evaluation of the downscaled results

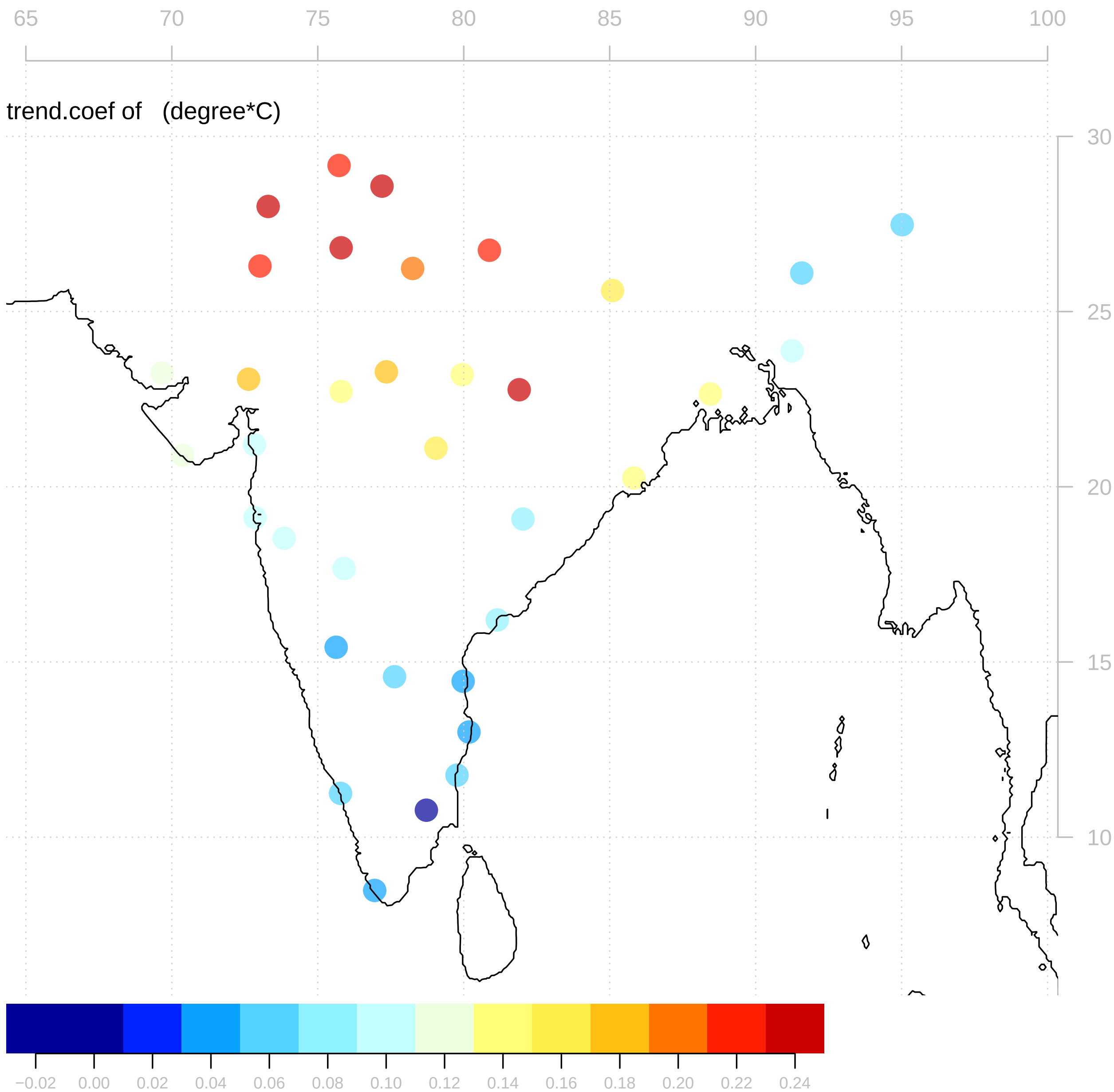
The downscaled results are for multi-model simulations with global climate models that have been subject to empirical-statistical downscaling. Since the downscaling was based on PCAs, it is necessary to convert the data back to the station format using `as.station`.

```
## Plot the ensemble mean for all stations
plot(as.station(as.station(dse.tmax.india.rcp45)),new=FALSE,map.show=FALSE)
```

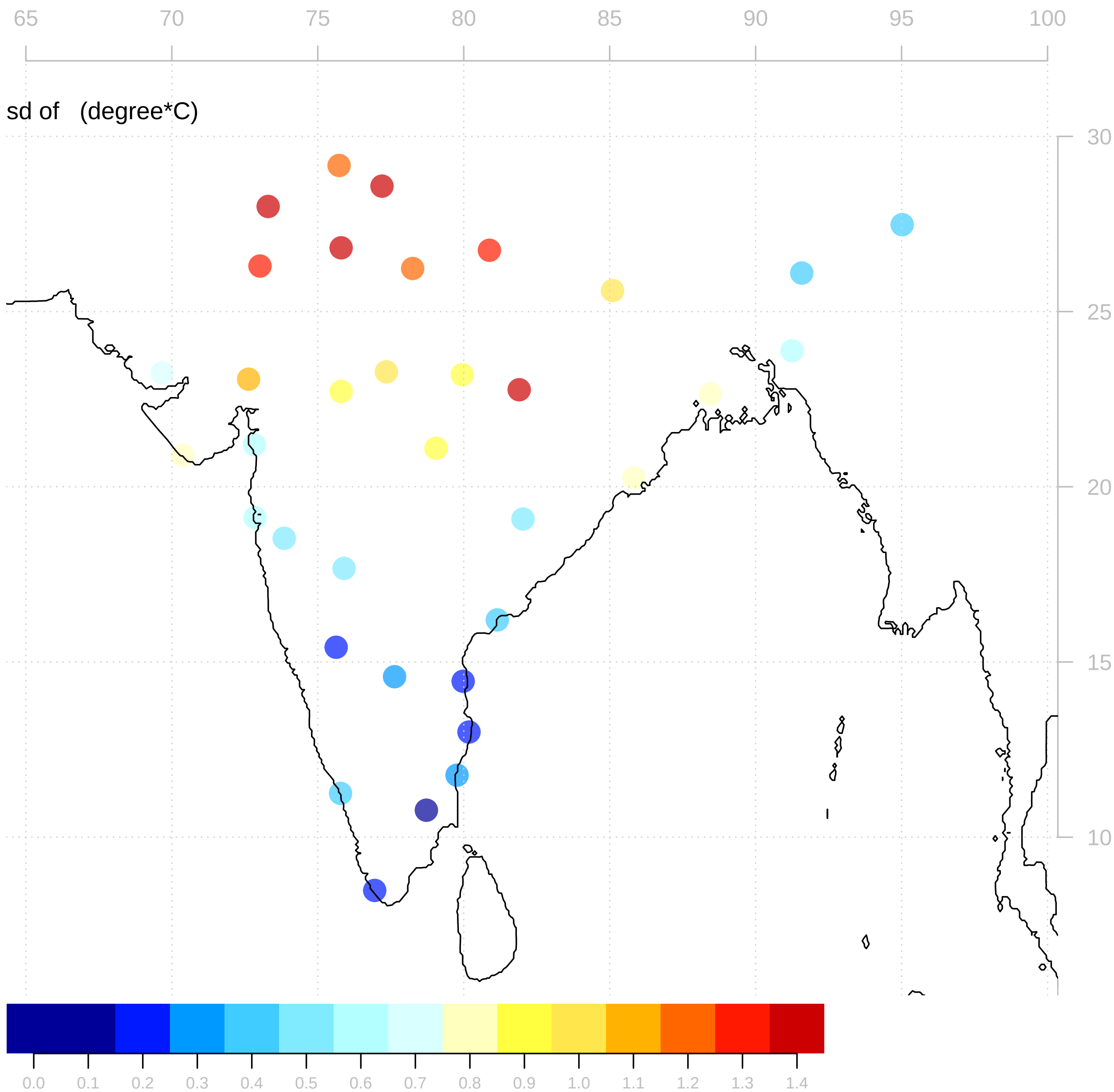


The picture of the ensemble mean for all sites in this case suggests a small number of sites with unrealistic outcomes: too weak variance and unrealistic trends.

```
map(as.station(as.station(dse.tmax.india.rcp45)),FUN='trend',new=FALSE)
```




```
map(as.station(as.station(dse.tmax.india.rcp45)),FUN='sd',new=FALSE)
```




```r
trends <- apply(as.station(as.station(dse.tmax.india.rcp45)),2,FUN='trend.coef')
sds <- apply(as.station(as.station(dse.tmax.india.rcp45)),2,FUN='sd')
names(trends) <- loc(dse.tmax.india.rcp45$pca)
names(sds) <- loc(dse.tmax.india.rcp45$pca)
print('Ensemble mean trends')
```

```
## [1] "Ensemble mean trends"
```

```r
print(trends)
```

```
##       PBO ANANTAPUR      MACHILIPATNAM             NELLORE
##         0.042396096        0.065087516         0.023157649
##             GAUHATI  DIBRUGARH/MOHANBAR               PATNA
##         0.055960474        0.052446884         0.149482933
##           AHMADABAD            VERAVAL      BHUJ-RUDRAMATA
##         0.179966763        0.117512745         0.105432506
```

```
##             SURAT              HISSAR              GADAG
##       0.093922582         0.200470389         0.033192114
##         KOZHIKODE  THIRUVANANTHAPURAM           JAGDALPUR
##       0.050201601         0.025479122         0.071568851
##       PENDRA ROAD             GWALIOR              INDORE
##       0.235390290         0.198933352         0.136133493
##          JABALPUR   BHOPAL/BAIRAGARH    BOMBAY/SANTACRUZ
##       0.135751268         0.163707792         0.087729474
##      NAGPUR SONEGA              POONA            SHOLAPUR
##       0.141952614         0.083776931         0.081666506
##           BHUBANE             BIKANER           JAIPUR/SA
##       0.131472349         0.224371531         0.223340019
##           JODHPUR             CUDDALO MADRAS/MINAMBAKKAM
##       0.213358575         0.040532488         0.029019236
##   TIRUCHCHIRAPALLI           AGARTALA          NEW DELHI/S
##      -0.001881341         0.084184857         0.224452292
##     LUCKNOW/AMAUSI    CALCUTTA/DUM DUM
##       0.205166999         0.125502850
```

```r
print('Ensemble mean sd')
```

```
## [1] "Ensemble mean sd"
```

```r
print(sds)
```

```
##      PBO ANANTAPUR      MACHILIPATNAM             NELLORE
##          0.2550684          0.3893135           0.1386033
##            GAUHATI  DIBRUGARH/MOHANBAR              PATNA
##          0.3653257          0.3534464           0.9080629
##          AHMADABAD            VERAVAL       BHUJ-RUDRAMATA
##          1.0663086          0.7036899           0.6285310
##              SURAT             HISSAR               GADAG
##          0.5629462          1.1879303           0.1996272
##          KOZHIKODE  THIRUVANANTHAPURAM          JAGDALPUR
##          0.3087965          0.1563670           0.4285867
##        PENDRA ROAD            GWALIOR              INDORE
##          1.3993506          1.1827468           0.8085784
##           JABALPUR   BHOPAL/BAIRAGARH    BOMBAY/SANTACRUZ
##          0.8086450          0.9744948           0.5263355
##      NAGPUR SONEGA              POONA            SHOLAPUR
##          0.8409180          0.4998924           0.4897819
##            BHUBANE            BIKANER           JAIPUR/SA
##          0.7949753          1.3361993           1.3325964
##            JODHPUR             CUDDALO MADRAS/MINAMBAKKAM
##          1.2726125          0.2444346           0.1744724
##   TIRUCHCHIRAPALLI           AGARTALA          NEW DELHI/S
##          0.0344218          0.5198012           1.3390391
##     LUCKNOW/AMAUSI    CALCUTTA/DUM DUM
##          1.2182644          0.7549569
```

The projected trends for RCP4.5 were in the range 0.20-0.25° C/decade in the interior and northern parts of India, and weaker in the south and east. The locations with suspicious trend estimates (less than 0.05° C/decade) were: PBO ANANTAPUR, NELLORE, GADAG, THIRUVANANTHAPURAM, CUDDALO, MADRAS, and TIRUCHCHIRAPALLI. The same sites also showed low variability (<0.3° C). These were also sites with low weights in the leading PCA shown above. # KPA 2018-05-31: The PCA didn't reflect much variance for the southern sites because the underlying observations had low variance in these stations.

Compare map(restation,FUN="sd") and map(tmax.fma) and you will see that it is exactly the same. Shouldn't we then expect low variance in the downscaled results for these stations too?

We can also check the station data individually and see that the PCA didn't reflect much variance for the southern sites:

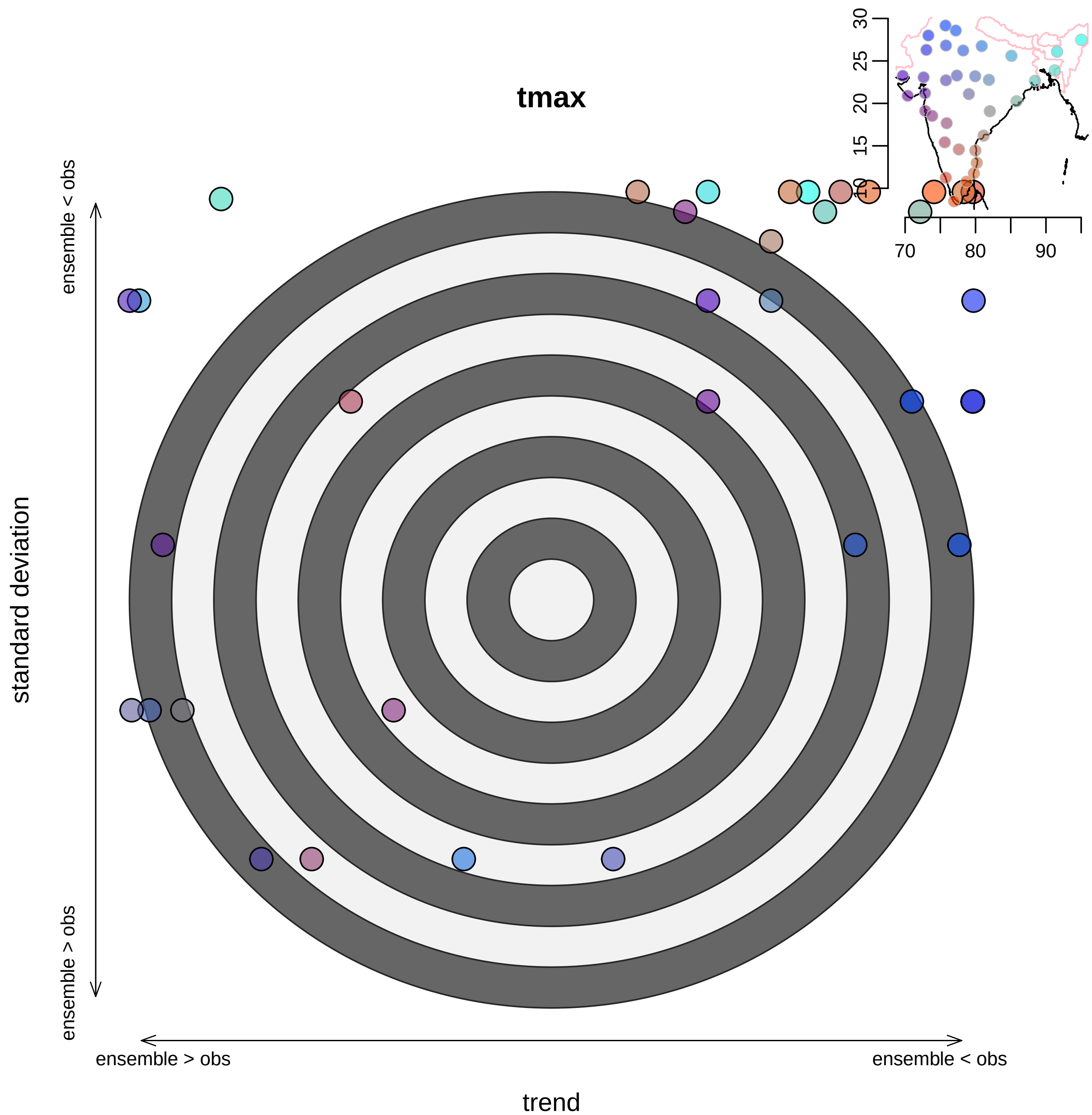
`{r, PCA-capture, fig.width=8, fig.height=8} #restation <- as.station(dse #attr(restation,'variable') <- 'tmax' #attr(restation,'unit') <- 'degC' #plot(restation,new=FALSE,map.show=FALSE) #map(restation,FUN=' #sds <- apply(restation,2,FUN='sd') #print('Std of tmax represented by the PCA') #print(sds) #`

Some of the same stations do have slightly less variance than the others, but one explanation for the different performance in the empirical-statistical downscaling is that the variability of the cited stations is mostly represented by different PCs than for the sites which were well reproduced.

Diagnosis and evaluation of the downscaled ensembles

The skill of simulating the trend and inter-annual variability for the different sites can be summarised in the following figure:

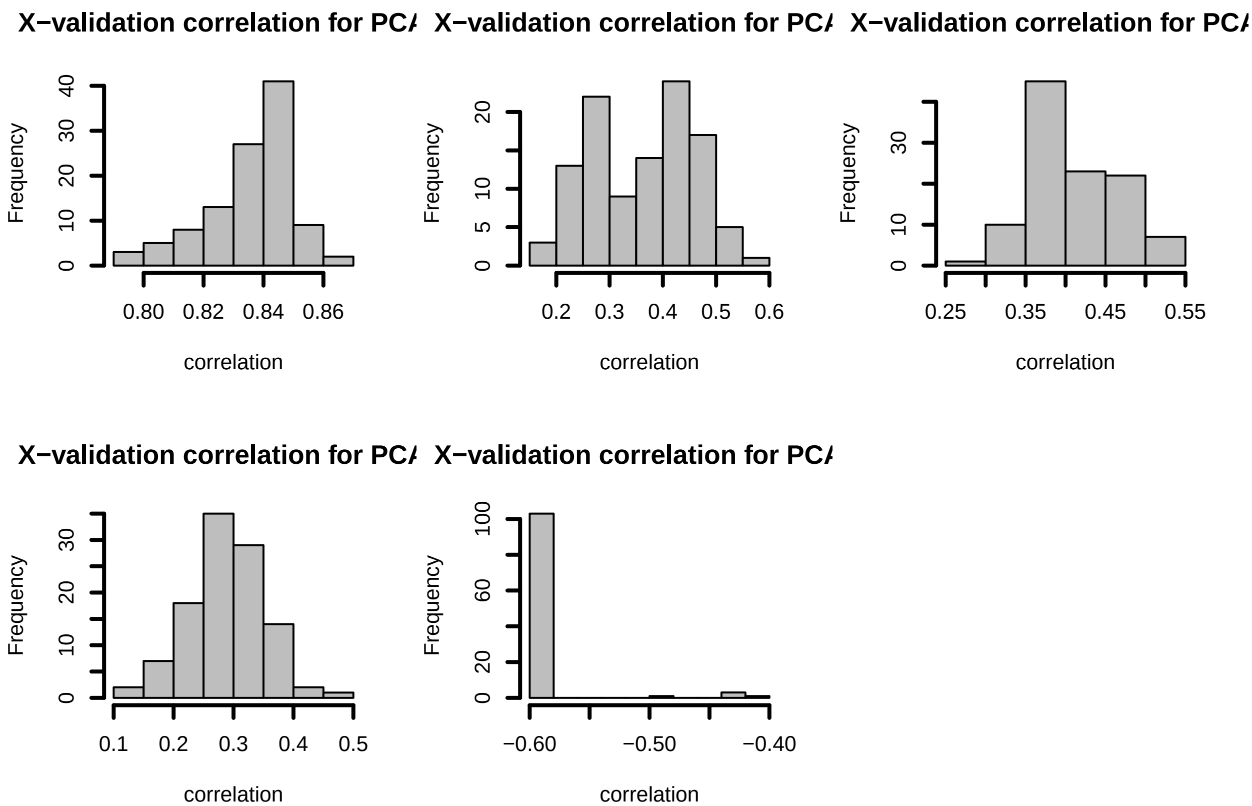
```r
diagnose(dse.tmax.india.rcp45,new=FALSE)
```



The plot shows how well the trend for the common interval (1970-2015) corresponded between the downscaled projections and the actual observations (x-axis) and how well the magnitude of the interannual variability was reproduced (y-axis). The colours of the symbols correspond with those in the map, and the locations performing less well are those in the southern part of India which had lower weights in the leading PCA. The points in upper right corner represent the locations where the downscaled results were associated with weaker trends and weaker interannual variations than seen in the observations. The points within the 'target' represent sites with skillful downscaling for the entire ensemble, and include sites more relevant for sites with wheat-crops.

We can also evaluate the downscaling of the different GCMs by examining the cross-validation correlations for each PC. In this case, it involved a five-fold cross-validation, meaning that the records were divided into five segments and four were used to calibrate the models whereas the last one was used for independent evaluation.

```
crossval(dse.tmax.india.rcp45,plot=TRUE)
par(mfcol=c(1,1))
```

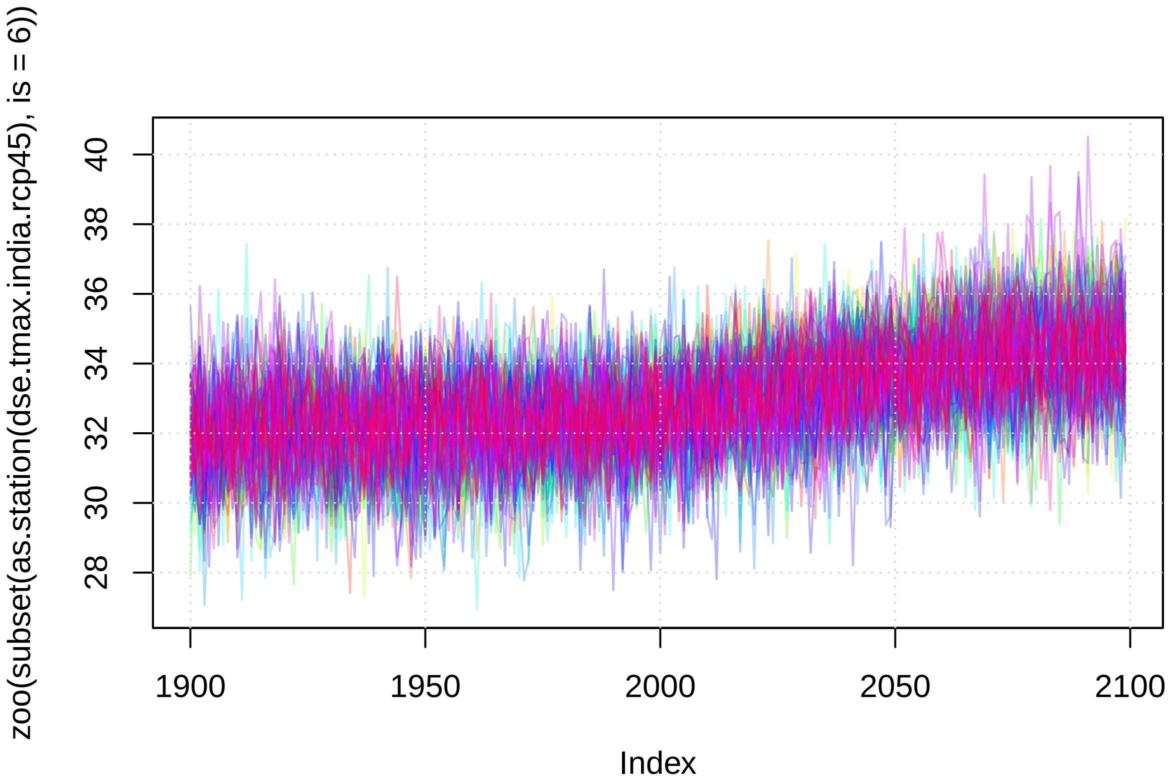


The results from the cross-validation indicate that the leading PCA for the stations was skillfully simulated, and that modes 2-4 could be described as moderately skillful. PCA 5, which carries lowest weights, was associated with negative skill.

Example of downscaled results

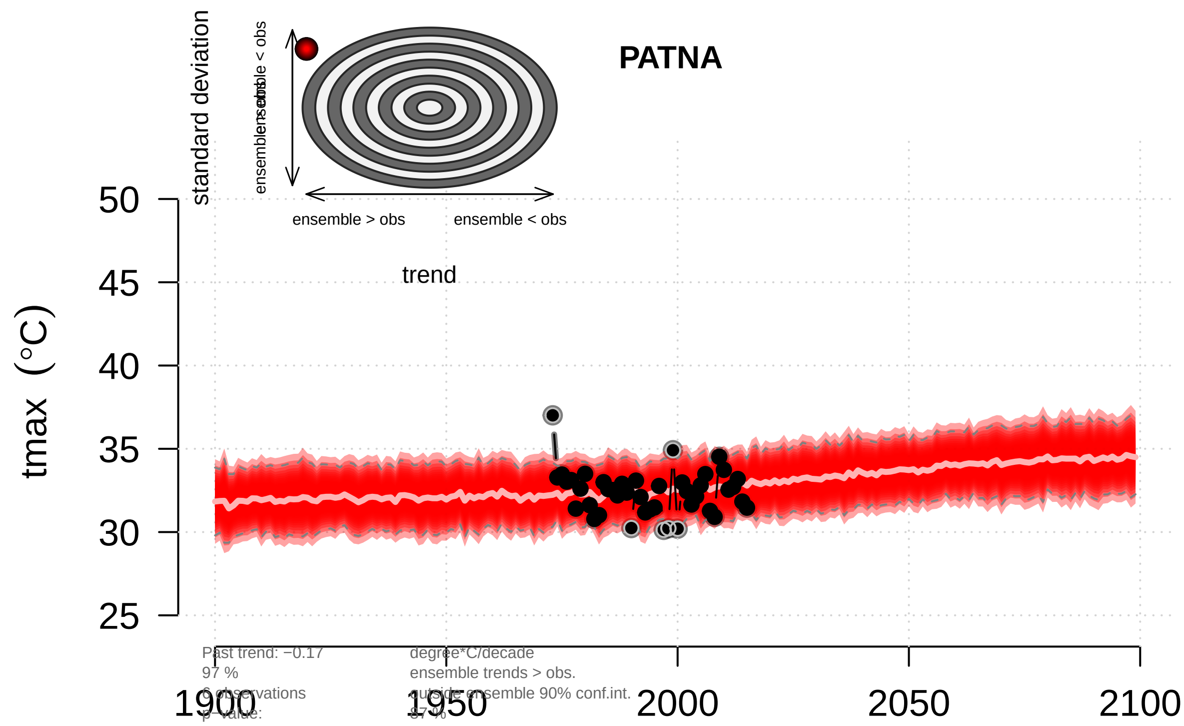
There is an abundance of information hidden in the compact downscaled results, and we provide some examples of what it contains in the following chuncks of R-code. For instance we can plot the downscaled results for anyone randomly selected site. Here, the different downscaled simulations (different models or runs) are shown in different colours.

```
plot(zoo(subset(as.station(dse.tmax.india.rcp45),is=6)),plot.type='single',
     col=rainbow(108,alpha=0.3))
grid()
```



The same results can be potted in terms as ensemble statistics such as the 90% confidence interval and compared with the observations:

```
plot(subset(as.station(dse.tmax.india.rcp45),is=6),new=FALSE,map.show=FALSE)
```



The figure shows the observed Feb-Apr mean daily maximum temperature $\overline{T_{[}max]}$ as a black time series and the ensemble statistics in red. The light central line is the ensemble mean and the gray-dashed lines mark the 90% confidence region. The map insert shows the site of the observations. This plot also presents diagnostics for this particular site, and the 'target diagram' indicates that the downscaled simulations were associated with stronger trends than observations (which probably is errorious due to the suspect data point in the start of the observations) but reasonable range of interannual variability.

This evaluation suggests that the downscaled results are reasonably well estimated.

The connection between the mean temperature and the mean duration of heat waves.

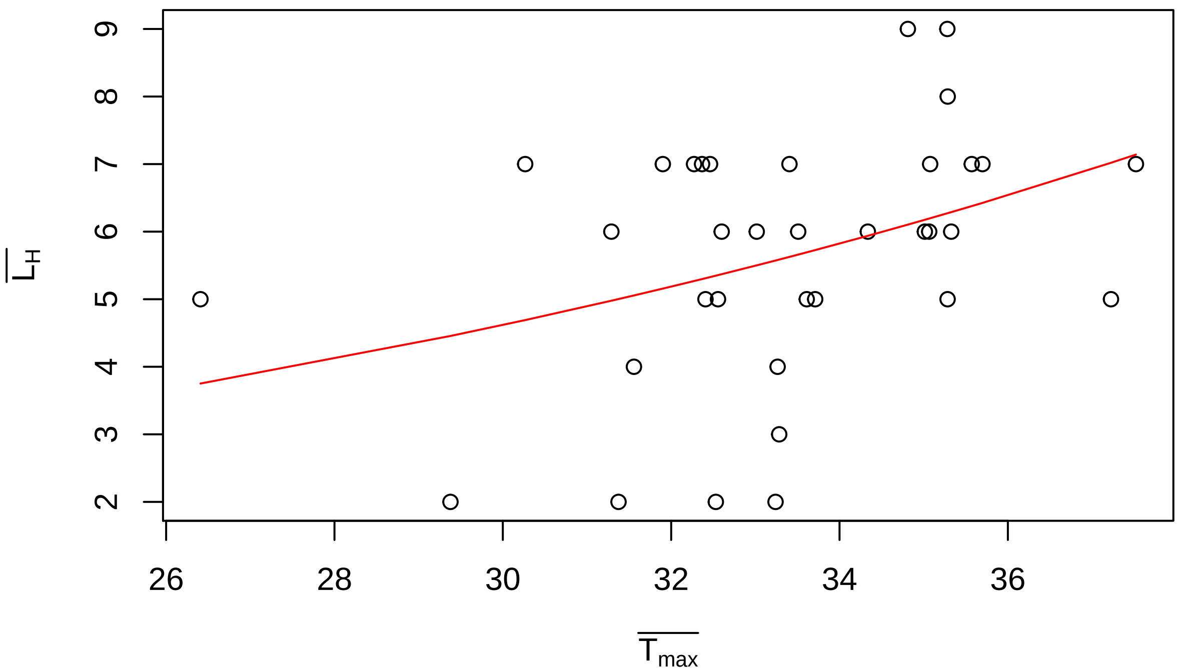
We have now some usefuldl projections for the Feb-Apr mean daily maximum temperature $\overline{T_{[}max]}$ for locations in India relevant for wheat crops, and need to make use of these projections to infer the consequence for heatwaves with temperatures exceeding 35° C. We can build this analysis on empirical information as presented below:

```
## Re-calibrate the model for Indian data
print('Re-calibrate')
```

```
## [1] "Re-calibrate"
```

```
i1 <- is.element(loc(tmax.fma),loc(lws.fma))
i2 <- is.element(loc(lws.fma),loc(tmax.fma))
calfit <- data.frame(x=colMeans(tmax.fma,na.rm=TRUE)[i1],y=round(colMeans(lws.fma,na.rm=TRUE)[i2]))
checkfit <- data.frame(x=colMeans(tmax.fma,na.rm=TRUE)[i1],y=apply(tmax.fma,2,FUN='sd',na.rm=TRUE)[i2])
ok <- is.finite(calfit$x) & is.finite(calfit$y) & (calfit$y >= 0)
calfit <- calfit[ok,]
```

```
attr(calfit,'max(x)') <- colMeans(tmax.fma,na.rm=TRUE)[i1]
#fit <- lm(y ~ I(x) + I(x^2),data=calfit)
fit <- glm(y ~ x,data=calfit,family='poisson')
#dev.new()
plot(calfit,xlab=expression(bar(T[max])),ylab=expression(bar(L[H])))
srt <- order(calfit$x)
lines(calfit$x[srt],exp(predict(fit))[srt],col='red')
```

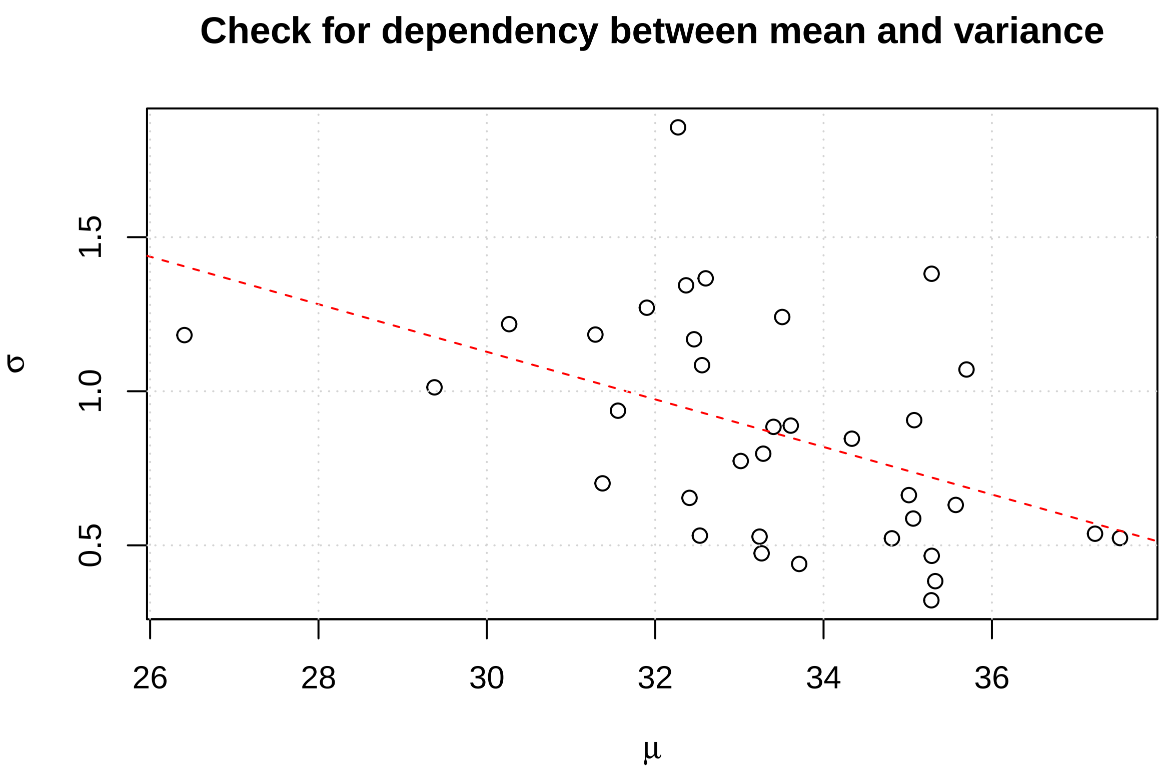


We used a generalised linear model (GLM) to quantify the relation between $\overline{T[max]}$ and $\overline{L_H}$. There is not a tight and strong fit between these two parameters in the Indian data, but similar calibration for a larger sample of European data that we consider to have higher quality suggest a stronger relationship (below). This quantified relation can nevertheless provide a crude and realistic scaling of the effect on the heatwave duration.

Test assumption that variance does not vary with the mean

If the pdf is merely shifted with the chnging means, i.e. that the variance is constant, then this provides an explanation for why the

```
## Test assumption: the variance is not systematically affeced bythe mean:
plot(checkfit$x,checkfit$y,xlab=expression(mu),ylab=expression(sigma),
     main='Check for dependency between mean and variance')
abline(lm(y ~x,data=checkfit),col='red',lty=2)
grid()
```



```r
## Summary of regression analysis between the seasonal mean temperature and seasonal standard deviaiton
print(summary(lm(y ~x, data=checkfit)))
```

```
##
## Call:
## lm(formula = y ~ x, data = checkfit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40200 -0.24397 -0.06696  0.16280  0.90377
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.44374    0.84725   4.065  0.00028 ***
## x           -0.07719    0.02534  -3.047  0.00453 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3233 on 33 degrees of freedom
## Multiple R-squared:  0.2195, Adjusted R-squared:  0.1959
## F-statistic: 9.281 on 1 and 33 DF,  p-value: 0.004529
```

There is a scatter in the mean-stdv points, and even a systematic dependency that can be considered statistical significant at the 5%-level. However the slope $|m|$ is small compared to the intercept $|c|$: $|b| \ll |c|$ and the best fit is strongly influenced by outliers.

Probabilities associated with 5-day heatwaves

We also need to account for how the probability for at least one heatwave occuring in a season may change with changing mean February–April mean temperature. Here we estimate the probabilities for one or more

5-day heatwaves for a randomly selected site (the sixth on the list):

```
## Use the ensemble mean for each station (hence repeat 'as.station' twice) for best estimate of the fu
dse.tmax.is6.45 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp45)),is
mnh.is6.45 <- predict(fit.nh,newdata=dse.tmax.is6.45)
pr.heatwave.is6.45 <- 1 - ppois(0,lambda=coredata(mnh.is6.45))
print(pr.heatwave.is6.45)
```

```
##         1         2         3         4         5         6         7
## 0.6234068 0.6251134 0.6257333 0.6006614 0.6107867 0.6260682 0.6254169
##         8         9        10        11        12        13        14
## 0.6226346 0.6335872 0.6331013 0.6274405 0.6308304 0.6348719 0.6219434
##        15        16        17        18        19        20        21
## 0.6257774 0.6233334 0.6315101 0.6302644 0.6309274 0.6374432 0.6354325
##        22        23        24        25        26        27        28
## 0.6268215 0.6239633 0.6367154 0.6382667 0.6384281 0.6351669 0.6374715
##        29        30        31        32        33        34        35
## 0.6454569 0.6378791 0.6300425 0.6209429 0.6274202 0.6362641 0.6384506
##        36        37        38        39        40        41        42
## 0.6369605 0.6301399 0.6350230 0.6363456 0.6218618 0.6425819 0.6433488
##        43        44        45        46        47        48        49
## 0.6407249 0.6363091 0.6261179 0.6337774 0.6360657 0.6337014 0.6335636
##        50        51        52        53        54        55        56
## 0.6383620 0.6294013 0.6385463 0.6411785 0.6529256 0.6260037 0.6416698
##        57        58        59        60        61        62        63
## 0.6333755 0.6400477 0.6369922 0.6462475 0.6496055 0.6386953 0.6563226
##        64        65        66        67        68        69        70
## 0.6461227 0.6399176 0.6420595 0.6257406 0.6365635 0.6437844 0.6290744
##        71        72        73        74        75        76        77
## 0.6420682 0.6406132 0.6425091 0.6441549 0.6502342 0.6337837 0.6479079
##        78        79        80        81        82        83        84
## 0.6502489 0.6575539 0.6449803 0.6452382 0.6527899 0.6386263 0.6271688
##        85        86        87        88        89        90        91
## 0.6394807 0.6542745 0.6552929 0.6468565 0.6573893 0.6547380 0.6551142
##        92        93        94        95        96        97        98
## 0.6551553 0.6251645 0.6265965 0.6465276 0.6505366 0.6558211 0.6620315
##        99       100       101       102       103       104       105
## 0.6513388 0.6624384 0.6586589 0.6615242 0.6701644 0.6615091 0.6684911
##       106       107       108       109       110       111       112
## 0.6591368 0.6721843 0.6587937 0.6659380 0.6686472 0.6669849 0.6621712
##       113       114       115       116       117       118       119
## 0.6642499 0.6811109 0.6783728 0.6707454 0.6843415 0.6785089 0.6748899
##       120       121       122       123       124       125       126
## 0.6808552 0.6765464 0.6853259 0.6779022 0.6849486 0.6804841 0.6896603
##       127       128       129       130       131       132       133
## 0.6862720 0.6911290 0.6930872 0.6893940 0.6885517 0.6873173 0.6956685
##       134       135       136       137       138       139       140
## 0.6938482 0.6979950 0.6958097 0.6908494 0.7056244 0.6968070 0.7109012
##       141       142       143       144       145       146       147
## 0.7048700 0.7002290 0.7046211 0.6982428 0.7085360 0.7068360 0.7076944
##       148       149       150       151       152       153       154
## 0.7100973 0.7134338 0.7128531 0.7113338 0.7136800 0.7055864 0.7125981
##       155       156       157       158       159       160       161
## 0.7098014 0.7125207 0.7222028 0.7212119 0.7274693 0.7229741 0.7251897
##       162       163       164       165       166       167       168
```

```
## 0.7219779 0.7255859 0.7275138 0.7269526 0.7256416 0.7276885 0.7225436
##       169       170       171       172       173       174       175
## 0.7307507 0.7348242 0.7235471 0.7270954 0.7294934 0.7314013 0.7324393
##       176       177       178       179       180       181       182
## 0.7303491 0.7290678 0.7315623 0.7368742 0.7364742 0.7431009 0.7344222
##       183       184       185       186       187       188       189
## 0.7354063 0.7388330 0.7380945 0.7381678 0.7383425 0.7318891 0.7404761
##       190       191       192       193       194       195       196
## 0.7423497 0.7327551 0.7366916 0.7409276 0.7371648 0.7428062 0.7355909
##       197       198       199       200
## 0.7380014 0.7463115 0.7433093 0.7414168
```

```
dse.tmax.is6.26 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp26)),is
mnh.is6.26 <- predict(fit.nh,newdata=dse.tmax.is6.26)
pr.heatwave.is6.26 <- 1 - ppois(0,lambda=coredata(mnh.is6.26))
dse.tmax.is6.85 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp85)),is
mnh.is6.85 <- predict(fit.nh,newdata=dse.tmax.is6.85)
pr.heatwave.is6.85 <- 1 - ppois(0,lambda=coredata(mnh.is6.85))
```

These calculations are used below in Figure 4.

Figure 4

Make use of the geometric distribution

Here we estimate the probabilites for 5-day heatwaves (T>35° C) based on the downscaled ensembles. For the geometric distribution of number of failures until first success, we use the formula $Pr(Y = k) = (1 - p)^{(}k - 1)p$ *for which the mean is* $\mu = (1 - p)/p$ *and* $p = 1/\mu$ *and* $k = 1, 2, \ldots$*. This framework is used to estimate the probability that a hot day (T>35° C) turns into a heat wave longer than five days.*

```
print('example station')
```

```
## [1] "example station"
```

```
dse.is6 <- subset(as.station(dse.tmax.india.rcp45),is=6)
mwl.is6 <- dse.is6; pdf.is6 <- dse.is6;
for (i in 1:dim(pdf.is6)[2]) {
  mwl.is6[,i] <- exp(predict(fit,newdata=data.frame(x=dse.is6[,i])))
  pdf.is6[,i] <- pgeom(q=5,prob=1/(mwl.is6[,i]),lower.tail=FALSE)
}
attr(mwl.is6,'variable') <- 'heatwave-duration'
attr(mwl.is6,'unit') <- 'days'

## The probability of a 5-day long hot episode given the mean temperature (divide by 2 since q50 has be
pbad <- zoo(100*apply(coredata(pdf.is6),1,FUN=quantile,probs=0.5),order.by=index(dse.is6)) * 0.5
class(pbad) <- class(Y)
pbad <- attrcp(subset(Y,is=6),pbad)
attr(pbad,'variable') <- 'Pr(L > 5 days)'
attr(pbad,'unit') <- '%'
```

The calculations are repeated for the high emission scenario:

```
## ------------------------------------------------------------------------
## High scenarion RCP8.5:
dse.is6h <- subset(as.station(dse.tmax.india.rcp85),is=6)
mwl.is6h <- dse.is6h; pdf.is6h <- dse.is6h
```

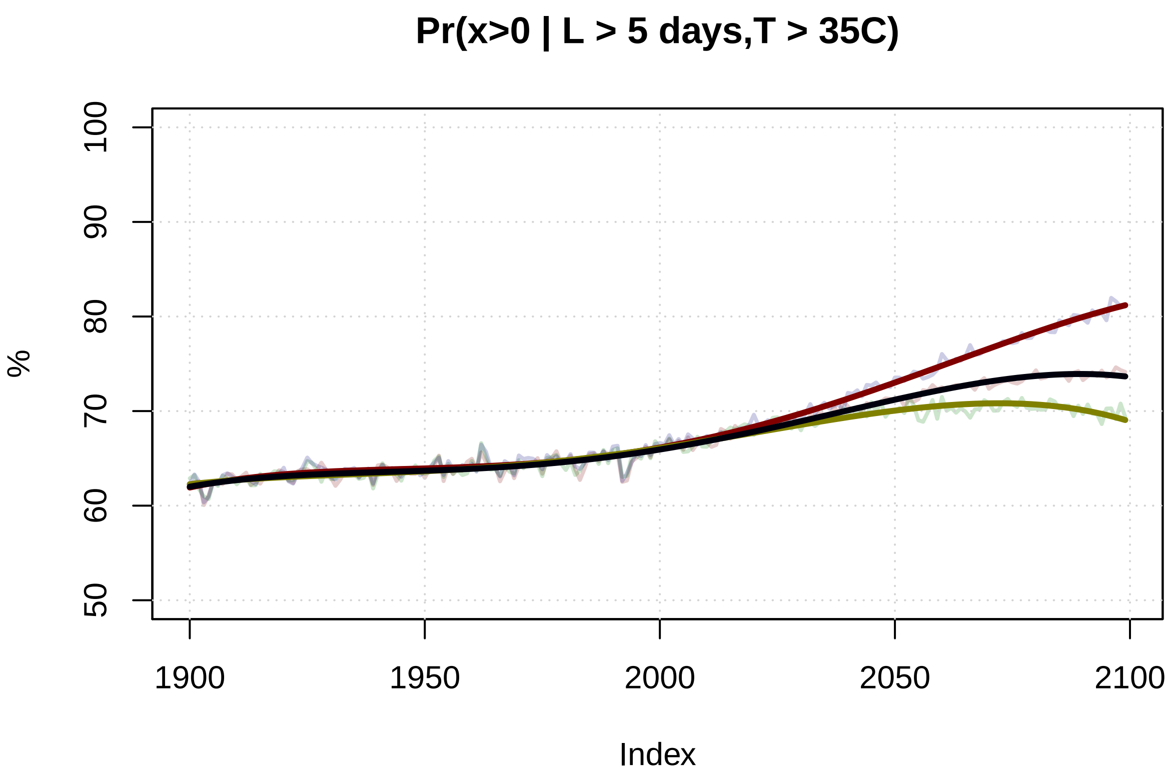
```
for (i in 1:dim(pdf.is6h)[2]) {
  mwl.is6h[,i] <- exp(predict(fit,newdata=data.frame(x=dse.is6h[,i])))
  pdf.is6h[,i] <- pgeom(q=5,prob=1/(mwl.is6h[,i]),lower.tail=FALSE)
}

## Plot the probability of a 5-day long hot episode (divide by 2 since q50 has been used)
pbadh <- zoo(100*apply(coredata(pdf.is6h),1,FUN=quantile,probs=0.5),order.by=index(dse.is6h)) * 0.5
class(pbadh) <- class(Y)
pbadh <- attrcp(subset(Y,is=6),pbadh)
attr(pbadh,'variable') <- 'Pr(L > 5 days)'
attr(pbadh,'unit') <- '%'
```

The calculations are repeated for the low emission scenario:

```
## -----------------------------------------------------------------------
## Low scenario RCPO 2.6:
dse.is6l <- subset(as.station(dse.tmax.india.rcp26),is=6)
mwl.is6l <- dse.is6l; pdf.is6l <- dse.is6l
for (i in 1:dim(pdf.is6l)[2]) {
  mwl.is6l[,i] <- exp(predict(fit,newdata=data.frame(x=dse.is6l[,i])))
  pdf.is6l[,i] <- pgeom(q=5,prob=1/(mwl.is6l[,i]),lower.tail=FALSE)
}

## Plot the probability of a 5-day long hot episode (divide by 2 since q50 has been used)
pbadl <- zoo(100*apply(coredata(pdf.is6l),1,FUN=quantile,probs=0.5),order.by=index(dse.is6l)) * 0.5
class(pbadl) <- class(Y)
pbadl <- attrcp(subset(Y,is=6),pbadl)
attr(pbadl,'variable') <- 'Pr(L > 5 days)'
attr(pbadl,'unit') <- '%'
```

The results are plotted in Figure 4 for a random site. The first plot (4a) shows the probability of one or more heatwaves in a season:

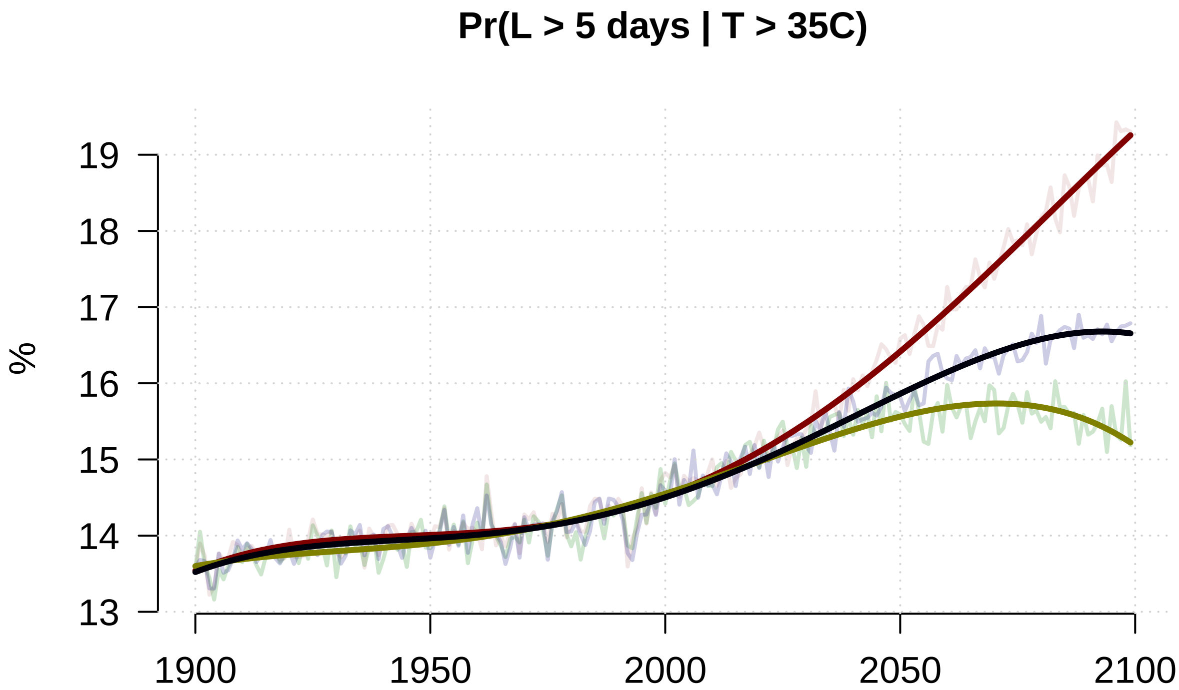
```
## Figure 4a
## Plot the probability of one or more events with daily maximum temperature
## above 35 degrees C lasting more than five days in the February-April season
pr.heatwave.is6.45 <- zoo(pr.heatwave.is6.45,order.by=index(pbadh))
pr.heatwave.is6.26 <- zoo(pr.heatwave.is6.26,order.by=index(pbadh))
pr.heatwave.is6.85 <- zoo(pr.heatwave.is6.85,order.by=index(pbadh))

plot(100*pr.heatwave.is6.45,main='Pr(x>0 | L > 5 days,T > 35C)',
     col=rgb(0.5,0,0,0.2),ylim=c(50,100),ylab='%',lwd=2,new=FALSE)
text(index(pbadh)[10],23.5,loc(pbadh))
lines(100*pr.heatwave.is6.26,lwd=2,col=rgb(0,0.5,0,0.2))
lines(100*pr.heatwave.is6.85,lwd=2,col=rgb(0,0,0.5,0.2))
grid()
## Add trend models
lines(trend(100*pr.heatwave.is6.85,model="y ~ I(t) + I(t^2) + I(t^3) + I(t^4)"),col=rgb(0.5,0,0),lwd=3)
lines(trend(100*pr.heatwave.is6.26,model="y ~ I(t) + I(t^2) + I(t^3) + I(t^4)"),col=rgb(0.5,0.5,0),lwd=
lines(trend(100*pr.heatwave.is6.45,model="y ~ I(t) + I(t^2) + I(t^3) + I(t^4)"),col=rgb(0,0,0.05),lwd=3
```




```
#dev.copy2pdf(file='fig4a.pdf')
```

The second panel (4b) shows the probability that a hot day turns into a heatwave.

```
## Figure 4b
## Plot the probability that a hot day (>35 degrees C) turns into a heatwave lasting more
## more than five days in the February-April season.
plot(pbadh,main='Pr(L > 5 days | T > 35C)',map.show=FALSE,
     col=rgb(0.5,0,0,0.2),lwd=2,new=FALSE)
text(index(pbadh)[10],23.5,loc(pbadh))
lines(pbadl,lwd=2,col=rgb(0,0.5,0,0.2))
lines(pbad,lwd=2,col=rgb(0,0,0.5,0.2))
grid()
## Add trend models
lines(trend(pbadh,model="y ~ I(t) + I(t^2) + I(t^3) + I(t^4)"),col=rgb(0.5,0,0),lwd=3)
lines(trend(pbadl,model="y ~ I(t) + I(t^2) + I(t^3) + I(t^4)"),col=rgb(0.5,0.5,0),lwd=3)
lines(trend(pbad,model="y ~ I(t) + I(t^2) + I(t^3) + I(t^4)"),col=rgb(0,0,0.05),lwd=3)
```




```
#dev.copy2pdf(file='fig4b.pdf')
```

Table 1 and 2 - Predicted and observed frequency of heatwaves

In Table 1 we quantify and summarise the probabilities for at least one heatwave (T>35° C over more than 5 consecutive days) during the February-April season for all sites, emission scenarios and for a selection of time slices. We also compare the projected probabilities with the observed frequency of hot events.

Table 2 shows the probabilities that a warm day (T>° C35) in February-April turns into a heatwave (i.e., that it lasts more than 5 consecutive days) for all sites, emission scenarios and for a selection of time slices. Here we include for comparison the fraction of hot days that last more than five days.

The following chunks of code were used to generate the contents of the tables, first for the present:

```
# Calculations for Table 1 - probability of at least one heatwave in a season
## Use the ensemble mean for each station (hence repeat 'as.station' twice) for best estimate of the fu
dse.tmax.2010.45 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp45)),i
mnh.2010.45 <- predict(fit.nh,newdata=dse.tmax.2010.45)
pr.heatwave.2010.45 <- 1 - ppois(0,lambda=coredata(mnh.2010.45))
dse.tmax.2010.26 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp26)),i
mnh.2010.26 <- predict(fit.nh,newdata=dse.tmax.2010.26)
pr.heatwave.2010.26 <- 1 - ppois(0,lambda=coredata(mnh.2010.26))
dse.tmax.2010.85 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp85)),i
mnh.2010.85 <- predict(fit.nh,newdata=dse.tmax.2010.85)
pr.heatwave.2010.85 <- 1 - ppois(0,lambda=coredata(mnh.2010.85))

# Calculations for Table 2 - probability of a hot day turning into a heatwave
dse.2010.rcp45 <-
  subset(as.station(as.station(dse.tmax.india.rcp45,FUN='quantile',probs=0.5)),it=c(2008,2012))
```

```
newdata <- data.frame(x=c(coredata(dse.2010.rcp45)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) * 0.5
coredata(dse.2010.rcp45) <- zzz

dse.2010.rcp26 <-
  subset(as.station(as.station(dse.tmax.india.rcp26,FUN='quantile',probs=0.5)),it=c(2008,2012))
newdata <- data.frame(x=c(coredata(dse.2010.rcp26)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) * 0.5
coredata(dse.2010.rcp26) <- zzz

dse.2010.rcp85 <-
  subset(as.station(as.station(dse.tmax.india.rcp85,FUN='quantile',probs=0.5)),it=c(2008,2012))
newdata <- data.frame(x=c(coredata(dse.2010.rcp85)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) *  0.5
coredata(dse.2010.rcp85) <- zzz
```

Then for the near future 2050:

```
# Table 1
dse.tmax.2050.45 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp45)),i
mnh.2050.45 <- predict(fit.nh,newdata=dse.tmax.2050.45)
pr.heatwave.2050.45 <- 1 - ppois(0,lambda=coredata(mnh.2050.45))
dse.tmax.2050.26 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp26)),i
mnh.2050.26 <- predict(fit.nh,newdata=dse.tmax.2050.26)
pr.heatwave.2050.26 <- 1 - ppois(0,lambda=coredata(mnh.2050.26))
dse.tmax.2050.85 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp85)),i
mnh.2050.85 <- predict(fit.nh,newdata=dse.tmax.2050.85)
pr.heatwave.2050.85 <- 1 - ppois(0,lambda=coredata(mnh.2050.85))

# Table 2
dse.2050.rcp45 <-
  subset(as.station(as.station(dse.tmax.india.rcp45,FUN='quantile',probs=0.5)),it=c(2058,2052))
newdata <- data.frame(x=c(coredata(dse.2050.rcp45)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) *  0.5
coredata(dse.2050.rcp45) <- zzz

dse.2050.rcp26 <-
  subset(as.station(as.station(dse.tmax.india.rcp26,FUN='quantile',probs=0.5)),it=c(2058,2052))
newdata <- data.frame(x=c(coredata(dse.2050.rcp26)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) *  0.5
coredata(dse.2050.rcp26) <- zzz

dse.2050.rcp85 <-
  subset(as.station(as.station(dse.tmax.india.rcp85,FUN='quantile',probs=0.5)),it=c(2058,2052))
newdata <- data.frame(x=c(coredata(dse.2050.rcp85)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) * 0.5
coredata(dse.2050.rcp85) <- zzz
```

Then for the far future 2100:

```
# Table 1
dse.tmax.2100.45 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp45)),i
mnh.2100.45 <- predict(fit.nh,newdata=dse.tmax.2100.45)
pr.heatwave.2100.45 <- 1 - ppois(0,lambda=coredata(mnh.2100.45))
dse.tmax.2100.26 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp26)),i
mnh.2100.26 <- predict(fit.nh,newdata=dse.tmax.2100.26)
pr.heatwave.2100.26 <- 1 - ppois(0,lambda=coredata(mnh.2100.26))
dse.tmax.2100.85 <- data.frame( x = c(t(coredata( subset(as.station(as.station(dse.tmax.india.rcp85)),i
mnh.2100.85 <- predict(fit.nh,newdata=dse.tmax.2100.85)
pr.heatwave.2100.85 <- 1 - ppois(0,lambda=coredata(mnh.2100.85))

# Table 2
dse.2100.rcp45 <-
  subset(as.station(as.station(dse.tmax.india.rcp45,FUN='quantile',probs=0.5)),it=c(2095,2100))
newdata <- data.frame(x=c(coredata(dse.2100.rcp45)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) * 0.5
coredata(dse.2100.rcp45) <- zzz

dse.2100.rcp26 <-
  subset(as.station(as.station(dse.tmax.india.rcp26,FUN='quantile',probs=0.5)),it=c(2095,2100))
newdata <- data.frame(x=c(coredata(dse.2100.rcp26)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) * 0.5
coredata(dse.2100.rcp26) <- zzz

dse.2100.rcp85 <-
  subset(as.station(as.station(dse.tmax.india.rcp85,FUN='quantile',probs=0.5)),it=c(2095,2100))
newdata <- data.frame(x=c(coredata(dse.2100.rcp85)))
mlength <- exp(predict(fit,newdata=newdata))
zzz <- 100*pgeom(q=5,prob=1/(mlength),lower.tail=FALSE) * 0.5
coredata(dse.2100.rcp85) <- zzz
```

Now we set up the contents of Table 1 and print the table so that it can be copied straight into the LaTeX manuscript:

```
## Table 1
r1 <- round(cbind(100*pr.heatwave.2010.45,100*pr.heatwave.2050.45,100*pr.heatwave.2100.45))
colnames(r1) <- paste(rep('RCP 4.5',3),c('2010','2050','2100'))

r2 <- round(cbind(100*pr.heatwave.2050.26,100*pr.heatwave.2100.26))
colnames(r2) <- paste(rep('RCP 2.6',2),c('2050','2100'))

r3 <- round(cbind(100*pr.heatwave.2050.85,100*pr.heatwave.2100.85))
colnames(r3) <- paste(rep('RCP 8.5',2),c('2050','2100'))

tab1 <- cbind(round(100*(nf.gt.5)),r1,r2,r3)
colnames(tab1)[1] <- 'obs.freq'
rownames(tab1) <- substr(loc(dse.2010.rcp45),1,9)
write.table(tab1,sep='  &  ',eol='  \\\\  \n',quote=FALSE)
```

```
## obs.freq  & RCP 4.5 2010  & RCP 4.5 2050  & RCP 4.5 2100  & RCP 2.6 2050  & RCP 2.6 2100  & RCP 8.5 2
## PBO ANANT  & 60  & 83  & 83  & 84  & 83  & 83  & 84  & 85  \\
## MACHILIPA  & 63  & 70  & 72  & 73  & 71  & 71  & 73  & 76  \\
```

```
## NELLORE  & 79  & 77  & 78  & 78  & 78  & 78  & 78  & 79  \\
## GAUHATI   & 9  & 48  & 51  & 53  & 50  & 49  & 53  & 59  \\
## DIBRUGARH  & 2  & 20  & 26  & 28  & 24  & 22  & 27  & 36  \\
## PATNA   & 84  & 67  & 71  & 74  & 70  & 69  & 74  & 81  \\
## AHMADABAD  & 63  & 78  & 81  & 83  & 80  & 80  & 83  & 88  \\
## VERAVAL   & 12  & 61  & 65  & 68  & 64  & 64  & 67  & 74  \\
## BHUJ-RUDR  & 67  & 77  & 79  & 80  & 78  & 78  & 80  & 84  \\
## SURAT   & 100  & 77  & 79  & 80  & 78  & 78  & 80  & 83  \\
## HISSAR  & 60  & 62  & 68  & 72  & 66  & 66  & 71  & 81  \\
## GADAG   & 74  & 77  & 78  & 78  & 78  & 78  & 78  & 80  \\
## KOZHIKODE  & 16  & 70  & 72  & 72  & 71  & 71  & 72  & 75  \\
## THIRUVANA  & 2  & 70  & 70  & 71  & 70  & 70  & 71  & 72  \\
## JAGDALPUR  & 77  & 76  & 77  & 78  & 77  & 77  & 78  & 81  \\
## PENDRA RO  & 70  & 68  & 73  & 77  & 72  & 72  & 77  & 86  \\
## GWALIOR   & 49  & 68  & 73  & 76  & 72  & 72  & 75  & 84  \\
## INDORE  & 65  & 75  & 78  & 79  & 77  & 76  & 79  & 84  \\
## JABALPUR  & 53  & 71  & 74  & 77  & 73  & 73  & 76  & 82  \\
## BHOPAL/BA  & 53  & 72  & 76  & 78  & 75  & 75  & 78  & 84  \\
## BOMBAY/SA  & 21  & 67  & 70  & 71  & 69  & 68  & 71  & 76  \\
## NAGPUR SO  & 65  & 79  & 81  & 83  & 81  & 81  & 83  & 87  \\
## POONA   & 88  & 78  & 79  & 80  & 79  & 79  & 80  & 83  \\
## SHOLAPUR  & 60  & 84  & 85  & 86  & 84  & 84  & 85  & 88  \\
## BHUBANE  & 95  & 77  & 80  & 82  & 79  & 79  & 81  & 86  \\
## BIKANER   & 74  & 69  & 75  & 78  & 73  & 72  & 76  & 85  \\
## JAIPUR/SA  & 58  & 66  & 72  & 75  & 70  & 70  & 74  & 84  \\
## JODHPUR  & 58  & 73  & 77  & 80  & 76  & 76  & 79  & 87  \\
## CUDDALO  & 51  & 66  & 67  & 68  & 67  & 67  & 68  & 70  \\
## MADRAS/MI  & 91  & 72  & 72  & 73  & 72  & 72  & 73  & 74  \\
## TIRUCHCHI  & 79  & 78  & 78  & 78  & 78  & 78  & 78  & 78  \\
## AGARTALA  & 42  & 62  & 65  & 67  & 64  & 64  & 67  & 73  \\
## NEW DELHI  & 67  & 57  & 65  & 69  & 62  & 62  & 68  & 80  \\
## LUCKNOW/A  & 65  & 68  & 73  & 76  & 72  & 71  & 75  & 84  \\
## CALCUTTA/  & 88  & 70  & 73  & 75  & 72  & 72  & 74  & 81  \\
```

Then we set up the contents of Table 2 and print the table so that it can be copied straight into the LaTeX manuscript:

```
## Table 2
colnames(dse.2010.rcp45) <- loc(dse.2010.rcp45)
pr.45.2010 <- apply(dse.2010.rcp45,2,'mean')
colnames(dse.2050.rcp45) <- loc(dse.2050.rcp45)
pr.45.2050 <- apply(dse.2050.rcp45,2,'mean')
colnames(dse.2100.rcp45) <- loc(dse.2100.rcp45)
pr.45.2100 <- apply(dse.2100.rcp45,2,'mean')
r1 <- round(cbind(pr.45.2010,pr.45.2050,pr.45.2100))
colnames(r1) <- paste(rep('RCP 4.5',3),c('2010','2050','2100'))

colnames(dse.2010.rcp26) <- loc(dse.2010.rcp26)
pr.26.2010 <- apply(dse.2010.rcp26,2,'mean')
colnames(dse.2050.rcp26) <- loc(dse.2050.rcp26)
pr.26.2050 <- apply(dse.2050.rcp26,2,'mean')
colnames(dse.2100.rcp26) <- loc(dse.2100.rcp26)
pr.26.2100 <- apply(dse.2100.rcp26,2,'mean')
r2 <- round(cbind(pr.26.2050,pr.26.2100))
colnames(r2) <- paste(rep('RCP 2.6',2),c('2050','2100'))
```

```r
colnames(dse.2010.rcp85) <- loc(dse.2010.rcp85)
pr.85.2010 <- apply(dse.2010.rcp85,2,'mean')
colnames(dse.2050.rcp85) <- loc(dse.2050.rcp85)
pr.85.2050 <- apply(dse.2050.rcp85,2,'mean')
colnames(dse.2100.rcp85) <- loc(dse.2100.rcp85)
pr.85.2100 <- apply(dse.2100.rcp85,2,'mean')
r3 <- round(cbind(pr.85.2050,pr.85.2100))
colnames(r3) <- paste(rep('RCP 8.5',2),c('2050','2100'))

tab2 <- cbind(round(100*(f.gt.5)),r1,r2,r3)
colnames(tab2)[1] <- 'obs.freq'
rownames(tab2) <- substr(rownames(tab1),1,9)
write.table(tab2,sep=' & ',eol=' \\\\ \n',quote=FALSE)
```

```
## obs.freq  & RCP 4.5 2010  & RCP 4.5 2050  & RCP 4.5 2100  & RCP 2.6 2050  & RCP 2.6 2100  & RCP 8.5 2
## PBO ANANT  & 32  & 20  & 20  & 21  & 20  & 20  & 21  & 21  \\
## MACHILIPA  & 21  & 15  & 16  & 16  & 16  & 16  & 16  & 17  \\
## NELLORE  & 34  & 18  & 18  & 18  & 18  & 18  & 18  & 18  \\
## GAUHATI   & 5  & 11  & 12  & 12  & 11  & 11  & 12  & 13  \\
## DIBRUGARH  & 14  & 8  & 8  & 9  & 8  & 8  & 9  & 9  \\
## PATNA   & 42  & 15  & 16  & 17  & 16  & 15  & 17  & 19  \\
## AHMADABAD  & 30  & 18  & 19  & 20  & 19  & 19  & 20  & 23  \\
## VERAVAL   & 3  & 13  & 14  & 15  & 14  & 14  & 15  & 17  \\
## BHUJ-RUDR  & 30  & 18  & 19  & 19  & 18  & 18  & 19  & 21  \\
## SURAT   & 36  & 18  & 18  & 19  & 18  & 18  & 19  & 20  \\
## HISSAR  & 32  & 14  & 15  & 16  & 15  & 14  & 16  & 19  \\
## GADAG   & 45  & 18  & 18  & 18  & 18  & 18  & 18  & 19  \\
## KOZHIKODE  & 13  & 16  & 16  & 16  & 16  & 16  & 16  & 17  \\
## THIRUVANA  & 1  & 15  & 16  & 16  & 16  & 16  & 16  & 16  \\
## JAGDALPUR  & 45  & 17  & 18  & 18  & 18  & 18  & 18  & 19  \\
## PENDRA RO  & 32  & 15  & 17  & 18  & 16  & 16  & 18  & 21  \\
## GWALIOR   & 23  & 15  & 17  & 17  & 16  & 16  & 17  & 21  \\
## INDORE  & 27  & 17  & 18  & 18  & 18  & 17  & 19  & 21  \\
## JABALPUR  & 27  & 16  & 17  & 17  & 16  & 16  & 17  & 20  \\
## BHOPAL/BA  & 29  & 16  & 17  & 18  & 17  & 17  & 18  & 21  \\
## BOMBAY/SA  & 4  & 15  & 15  & 16  & 15  & 15  & 16  & 17  \\
## NAGPUR SO  & 37  & 18  & 19  & 20  & 19  & 19  & 20  & 22  \\
## POONA   & 40  & 18  & 19  & 19  & 18  & 18  & 19  & 20  \\
## SHOLAPUR  & 35  & 20  & 21  & 21  & 21  & 21  & 21  & 23  \\
## BHUBANE  & 38  & 18  & 19  & 20  & 19  & 19  & 19  & 22  \\
## BIKANER   & 38  & 15  & 17  & 18  & 16  & 16  & 18  & 21  \\
## JAIPUR/SA  & 38  & 14  & 16  & 17  & 16  & 15  & 17  & 21  \\
## JODHPUR  & 38  & 16  & 18  & 19  & 17  & 17  & 19  & 22  \\
## CUDDALO  & 22  & 14  & 15  & 15  & 15  & 15  & 15  & 16  \\
## MADRAS/MI  & 30  & 16  & 16  & 16  & 16  & 16  & 16  & 17  \\
## TIRUCHCHI  & 40  & 18  & 18  & 18  & 18  & 18  & 18  & 18  \\
## AGARTALA  & 22  & 14  & 14  & 15  & 14  & 14  & 15  & 16  \\
## NEW DELHI  & 33  & 12  & 14  & 15  & 14  & 14  & 15  & 19  \\
## LUCKNOW/A  & 41  & 15  & 16  & 17  & 16  & 16  & 17  & 21  \\
## CALCUTTA/  & 34  & 15  & 16  & 17  & 16  & 16  & 17  & 19  \\
```

Region for wheat crops

Define regions for wheat crops that are used in maps.

```
west_lat <- c(28.50, 31.82, 33.19, 34.05, 29.80, 25.81, 28.50)
west_lon <- c(71.86, 74.08, 72.68, 73.72, 78.45, 74.82, 71.86)
cent_lat <- c(25.81, 29.80, 27.56, 22.05, 25.81)
cent_lon <- c(74.82, 78.45, 83.24, 80.89, 74.82)
east_lat <- c(22.05, 27.56, 26.95, 24.10, 22.13, 22.05)
east_lon <- c(80.89, 83.24, 88.00, 87.94, 84.83, 80.89)
```

Identify locations with questionable results

Identify locations with large differences between observed and estimated frequency of 5-day heatwaves.

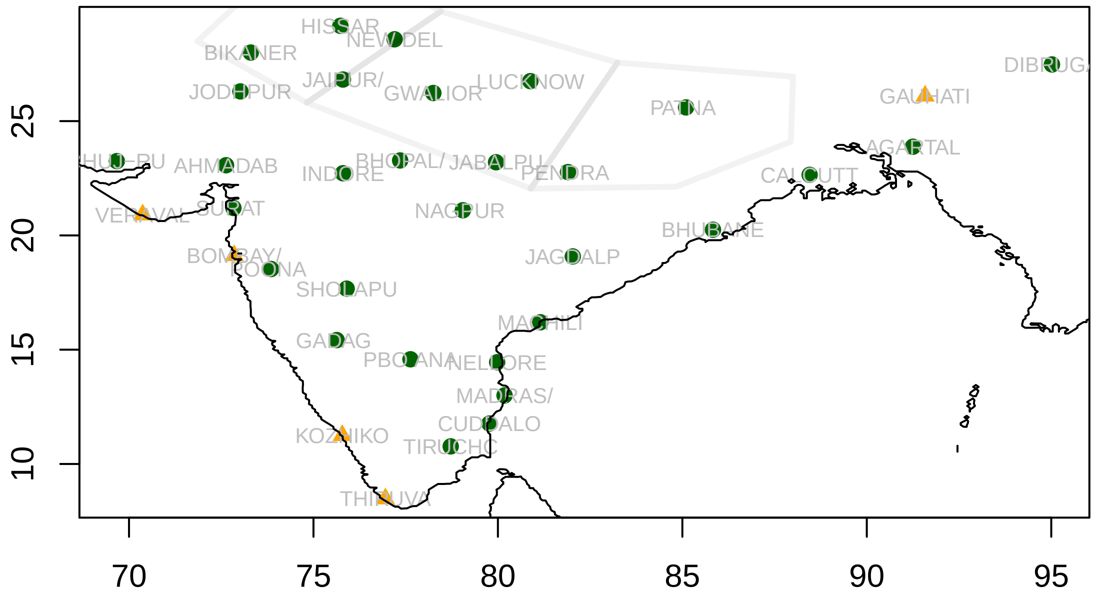
The first map shows a comparison between the observed seasonal frequency of heatwaves and the projected probability of at least one heatwave (T>35° C more than 5 days). The locations with an orange triangle are those with a poor match in Table 1.

```
err <- abs(100*(nf.gt.5) - 100*(pr.heatwave))
print(err)
```

```
##      PBO ANANTAPUR     MACHILIPATNAM           NELLORE
##          0.8607814         5.8606842         5.6712220
##            GAUHATI  DIBRUGARH/MOHANBAR             PATNA
##          8.0017610         0.3454487         1.9892826
##          AHMADABAD           VERAVAL      BHUJ-RUDRAMATA
##          3.5876530        23.9684509         3.3288818
##              SURAT            HISSAR             GADAG
##          3.6883167         2.0037738         6.3764045
##          KOZHIKODE  THIRUVANANTHAPURAM         JAGDALPUR
##         14.6474972         4.4350366         0.9155877
##        PENDRA ROAD            GWALIOR            INDORE
##          1.0033004         9.6844995         0.5828692
##           JABALPUR    BHOPAL/BAIRAGARH   BOMBAY/SANTACRUZ
##         13.8836484         9.7236838        13.3650855
##      NAGPUR SONEGA             POONA          SHOLAPUR
##          0.1100540         3.9393561         4.9509229
##            BHUBANE           BIKANER          JAIPUR/SA
##          2.9262376         1.8886195         8.2388157
##            JODHPUR           CUDDALO MADRAS/MINAMBAKKAM
##          9.8785629         1.1258178         7.0630881
##    TIRUCHCHIRAPALLI          AGARTALA        NEW DELHI/S
##          0.7405808        17.8871125         0.5762374
##     LUCKNOW/AMAUSI   CALCUTTA/DUM DUM
##          5.6544632         0.4362321
```

```
ile <- err/(100*(nf.gt.5)) > 0.5  ## identify the locations with large error
pch <- rep(19,length(err)); col <- rep('darkgreen',length(err))
pch[ile] <- 17; col[ile] <- 'orange'
plot(lon(lws),lat(lws),pch=pch,col=col,xlab='',ylab='')
text(lon(lws),lat(lws),substr(loc(lws),1,7),cex=0.7,col='grey')
data(geoborders)
lines(geoborders)
```

```r
lines(west_lon,west_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(cent_lon,cent_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(east_lon,east_lat,lwd=3,col=rgb(0,0,0,0.05))
```

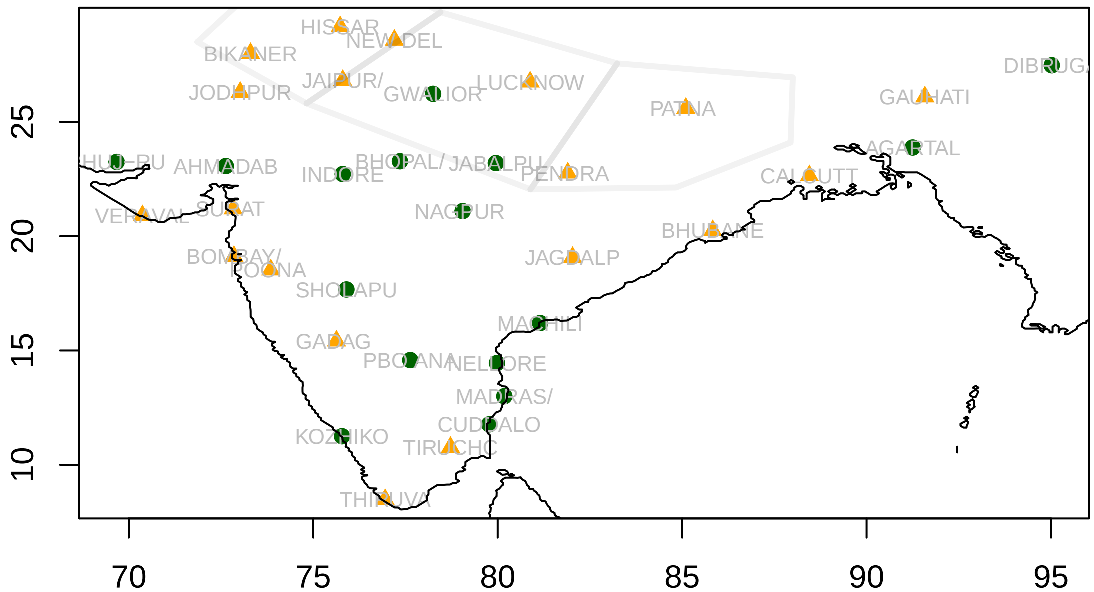


The second map shows the differences between the observed portion of hot days with a duration longer than five days and the projected probability that a hot day turns into a more than five day long heatwave. The locations with an orange triangle are those with a poor match in Table 2.

```r
err <- abs(100*(f.gt.5) - pr.45.2010)
print(err)
```

```
##      PBO ANANTAPUR      MACHILIPATNAM            NELLORE
##          11.582458           5.935178          15.898645
##            GAUHATI  DIBRUGARH/MOHANBAR             PATNA
##           6.462610           6.350828          27.463362
##          AHMADABAD            VERAVAL     BHUJ-RUDRAMATA
##          11.840918          10.163485          12.769283
##              SURAT             HISSAR              GADAG
##          18.772630          18.287543          27.267425
##          KOZHIKODE THIRUVANANTHAPURAM          JAGDALPUR
##           2.764765          14.582536          27.271363
##        PENDRA ROAD            GWALIOR             INDORE
##          17.079353           8.030975          10.614898
##           JABALPUR   BHOPAL/BAIRAGARH  BOMBAY/SANTACRUZ
##          11.510658          12.690190          10.234796
##       NAGPUR SONEGA              POONA           SHOLAPUR
##          18.383399          22.494378          14.733388
##            BHUBANE            BIKANER          JAIPUR/SA
##          20.522896          23.315233          23.405940
##            JODHPUR            CUDDALO MADRAS/MINAMBAKKAM
##          21.451554           7.075977          14.388534
##   TIRUCHCHIRAPALLI           AGARTALA         NEW DELHI/S
##          21.692976           8.774780          20.070394
##      LUCKNOW/AMAUSI  CALCUTTA/DUM DUM
##          25.680906          19.096666
```

```r
ile <- err/(100*(f.gt.5)) > 0.5  ## identify the locations with large error
pch <- rep(19,length(err)); col <- rep('darkgreen',length(err))
pch[ile] <- 17; col[ile] <- 'orange'
plot(lon(lws),lat(lws),pch=pch,col=col,xlab='',ylab='')
text(lon(lws),lat(lws),substr(loc(lws),1,7),cex=0.7,col='grey')
data(geoborders)
lines(geoborders)

lines(west_lon,west_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(cent_lon,cent_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(east_lon,east_lat,lwd=3,col=rgb(0,0,0,0.05))
```



The maps above highlight the sites with a difference greater than 50% between observed and projected probabilities for the present. Based on these results, it looks like the approach to estimate the probability of one or more heatwaves in a season (Table 1, top map) is somewhat more successful than calculating the probability that a hot day will turn into a heatwave (Table 2, lower map), although it may be related to the short data series rather than the underlying statistical assumptions. It is important to keep in mind that the frequency of observed heatwaves are uncertain due to problems with data availability and quality. We nevertheless get some crude results for probabilities despite sparse data of questionable quality for some sites. These results may be some of the best information there is for the probability of future heatwaves in India.

Figure 5

Based on the estimated probabilities for 5-day heatwaves at station levels, maps of probabilities for 5-day heatwaves in the far future 2100 were generated though gridding. Here a kriging method was used (based on work done at iMAGE/NCAR) that used elevation as a covariate.

```r
## Figure 5a Map of the probability of at least one event per season at 2100 RCP4.5
demo(gridmap,ask=FALSE)
```

```
##
##
##  demo(gridmap)
##  ---- ~~~~~~~
##
```

```
## > gridmap <- function(Y,FUN='mean',colbar=NULL,project='lonlat',xlim=NULL,ylim=NULL,zlim=NULL,verbos
## +
## +   if (verbose) print(paste('gridmap',FUN))
## +   if (is.null(xlim)) xlim <- range(lon(Y))
## +   if (is.null(ylim)) ylim <- range(lat(Y))
## +   if (!is.null(dim(Y))) {
## +       y <- apply(Y,2,FUN,na.rm=TRUE)
## +   } else {
## +      y <- Y  ## single specific date
## +   }
## +
## +   ## Get data on the topography on the 5-minute resolution
## +   if (verbose) print('Use etopo5 elevation data')
## +   data(etopo5)
## +   etopo5 <- subset(etopo5,
## +                    is=list(lon=range(lon(Y))+c(-1,1),
## +                            lat=range(lat(Y))+c(-1,1)))
## +   ## Mask the sea: elevations below 1m below sea level is masked.
## +   etopo5[etopo5<=-1] <- NA
## +   if (!is.null(zlim)) {etopo5[(etopo5<min(zlim)) | ((etopo5>max(zlim)))] <- NA}
## +
## +   ## Set the grid to be the same as that of etopo5:
## +   if (verbose) print('Use same structure as etopo5')
## +   grid <- structure(list(x=lon(etopo5),y=lat(etopo5)),class='gridList')
## +
## +   ## Flag dubplicated stations:
## +   if (verbose) print('Check for duplicates')
## +   ok <- !(duplicated(lon(Y)) & duplicated(lat(Y)))
## +
## +   ## Kriging
## +   if (verbose) print(paste('Apply kriging to',sum(ok),'locations'))
## +
## +   ## KMP 2017-08-07: moved require(LatticeKrig) down here because
## +   ## it interfered with function unit which is used in subset.pattern
## +   require(LatticeKrig)
## +   obj <- LatticeKrig( x=cbind(lon(Y)[ok],lat(Y)[ok]),
## +                       y=y[ok],Z=alt(Y)[ok])
## +
## +   ##  obj <- LatticeKrig( x=cbind(lon[ok],lat[ok]), y=z[2,ok],Z=alt[ok])
## +   if (verbose) print('Predict surface')
## +   w <- predictSurface(obj, grid.list = grid,Z=etopo5)
## +   w$z[is.na(etopo5)] <- NA
## +
## +   ## Get rid of packages that have functions of same name:
## +   detach("package:LatticeKrig")
## +   detach("package:fields")
## +   detach("package:spam")
## +   detach("package:grid")
## +   detach("package:maps")
## +
## +   ## Convert the results from LatticeKrig to esd:
## +   W <- w$z
## +   attr(W,'variable') <- varid(Y)[1]
## +   attr(W,'unit') <- esd::unit(Y)[1]
```

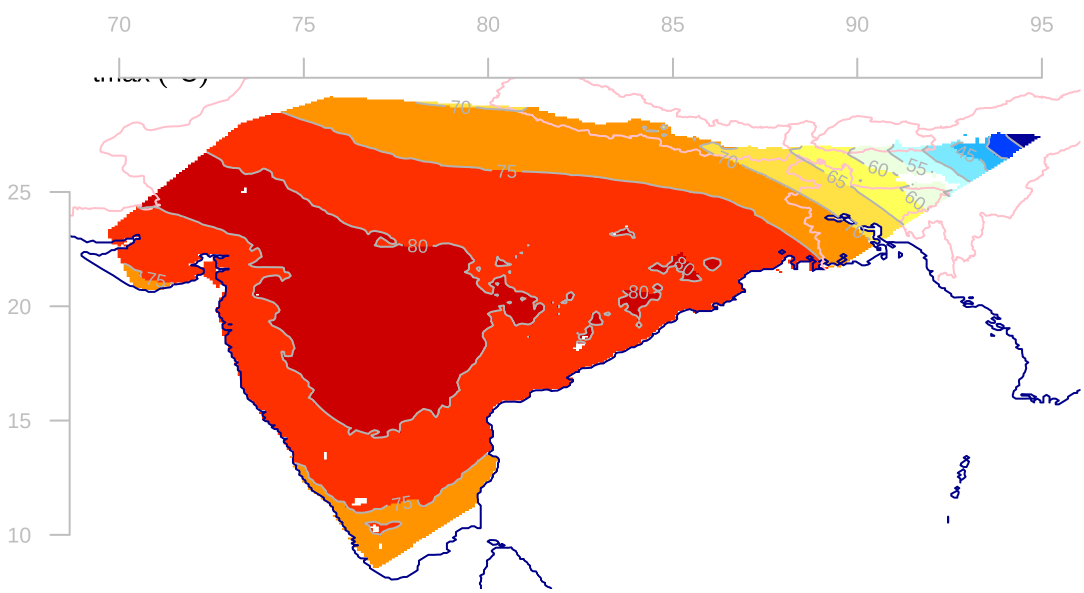
```
## +   attr(W,'longitude') <- w$x
## +   attr(W,'latitude') <- w$y
## +   class(W) <- class(etopo5)
## +
## +   ## Make the graphics
## +   if (verbose | plot) print("make the map")
## +   map(W,xlim=xlim,ylim=ylim,zlim=zlim,colbar=colbar,project=project)
## +
## +   invisible(W)
## + }
```

```
attr(pr.heatwave.2100.45,'longitude') <- lon(dse.2100.rcp45)
attr(pr.heatwave.2100.45,'latitude') <- lat(dse.2100.rcp45)
attr(pr.heatwave.2100.45,'altitude') <- alt(dse.2100.rcp45)
attr(pr.heatwave.2100.45,'unit') <- esd::unit(dse.2100.rcp45)
attr(pr.heatwave.2100.45,'variable') <- varid(dse.2100.rcp45)
print(summary(pr.heatwave.2100.45))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2789  0.7199  0.7658  0.7433  0.7958  0.8560
```

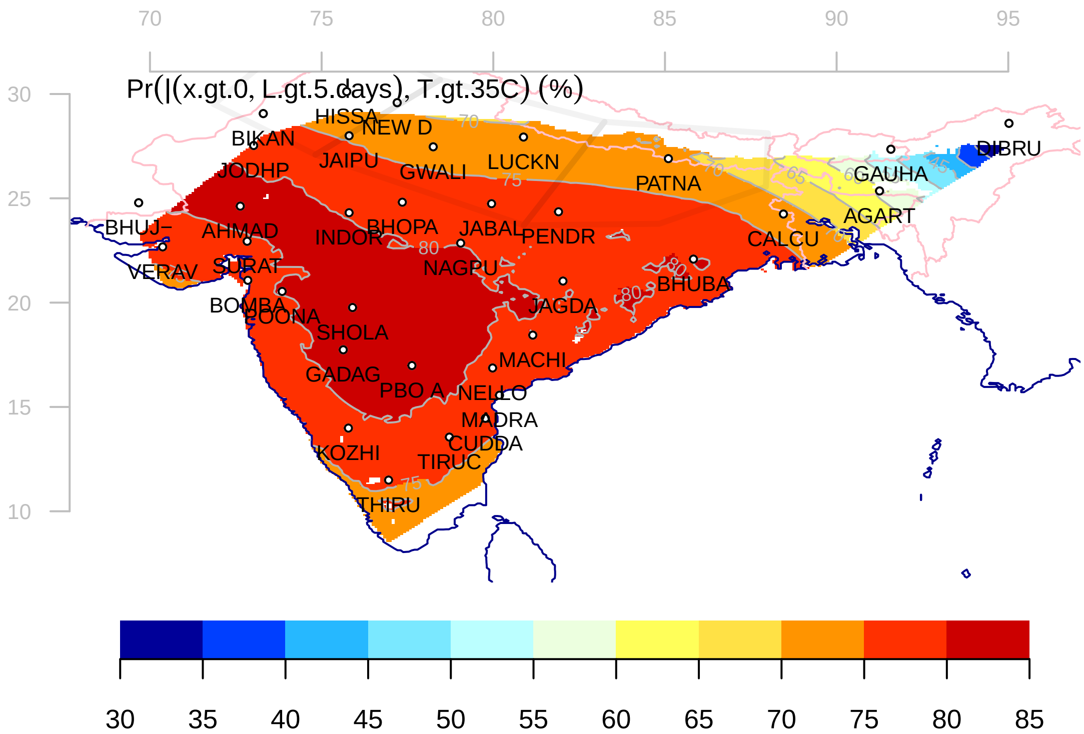
```
prng0 <- gridmap(Y=100*pr.heatwave.2100.45,zlim=c(0,max(alt(tmax))+100),verbose=TRUE)
```

```
## [1] "gridmap mean"
## [1] "Use etopo5 elevation data"
## [1] "Use same structure as etopo5"
## [1] "Check for duplicates"
## [1] "Apply kriging to 35 locations"
## [1] "Predict surface"
## [1] "make the map"
```



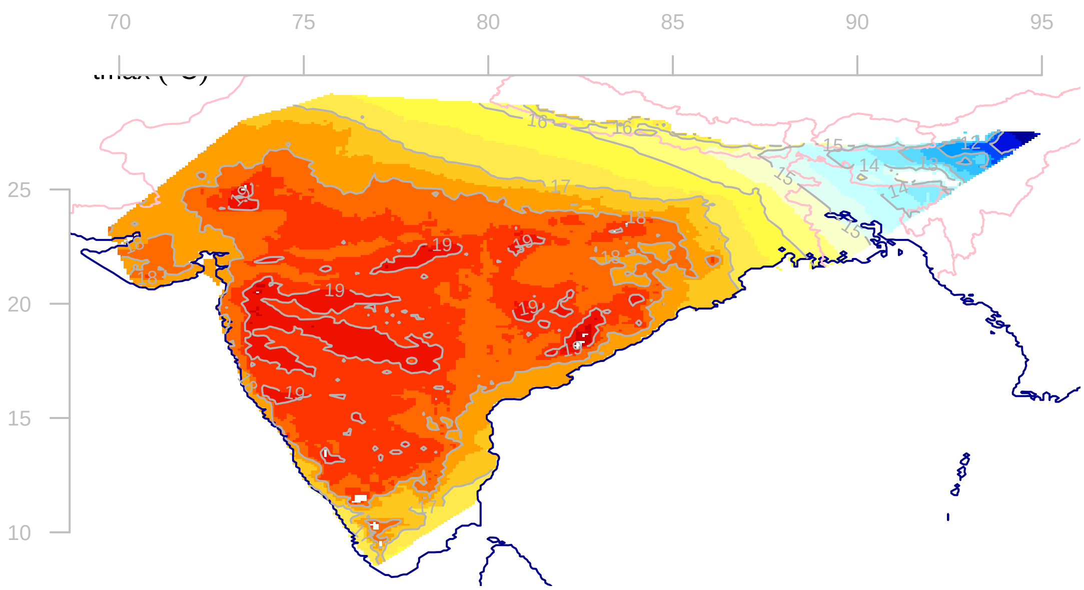

```
attr(prng0,'variable') <- 'Pr(x.gt.0|L.gt.5.days,T.gt.35C)'
attr(prng0,'unit') <- '"%"'
map(prng0,new=FALSE)
par(fig=c(0,1,0.1,1))
points(lon(dse.2100.rcp45),lat(dse.2100.rcp45),pch=19,col='white',cex=0.5)
points(lon(dse.2100.rcp45),lat(dse.2100.rcp45),cex=0.5)
```

```
text(lon(dse.2100.rcp45),lat(dse.2100.rcp45),substr(loc(dse.2100.rcp45),1,5),cex=0.7,pos=1)
lines(west_lon,west_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(cent_lon,cent_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(east_lon,east_lat,lwd=3,col=rgb(0,0,0,0.05))
```



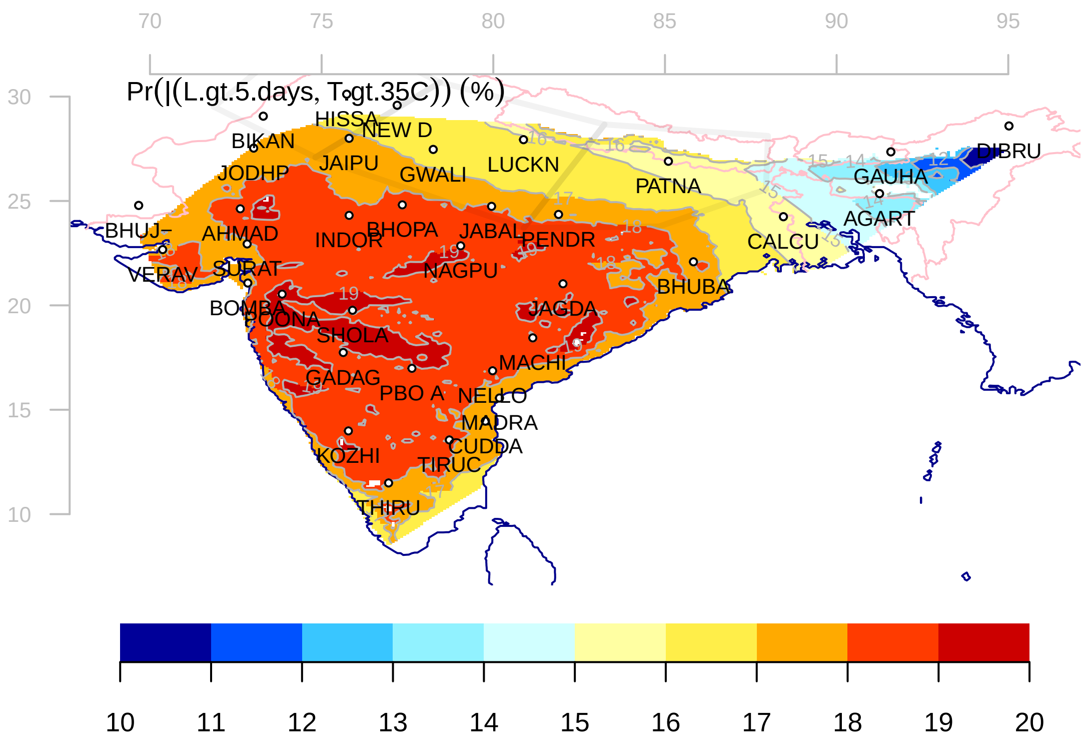

```
#dev.copy2pdf(file='fig5a.pdf')
```

```
## Figure 5b Map of the probability of a hot day (Tmax>35C) turning into a heat wave lasting more than
prgt5d <- gridmap(dse.2100.rcp45,zlim=c(0,max(alt(tmax))+100))
```




```
attr(prgt5d,'variable') <- 'Pr(L.gt.5.days|T.gt.35C)'
attr(prgt5d,'unit') <- '"%"'
```

```r
map(prgt5d,new=FALSE)
par(fig=c(0,1,0.1,1))
points(lon(dse.2100.rcp45),lat(dse.2100.rcp45),pch=19,col='white',cex=0.5)
points(lon(dse.2100.rcp45),lat(dse.2100.rcp45),cex=0.5)
text(lon(dse.2100.rcp45),lat(dse.2100.rcp45),substr(loc(dse.2100.rcp45),1,5),cex=0.7,pos=1)
lines(west_lon,west_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(cent_lon,cent_lat,lwd=3,col=rgb(0,0,0,0.05))
lines(east_lon,east_lat,lwd=3,col=rgb(0,0,0,0.05))
```




```r
#dev.copy2pdf(file='fig5b.pdf')
```

More supporting material

For completeness, we present maps of gridded daily Indian maximum/minimum temperatures for the present.

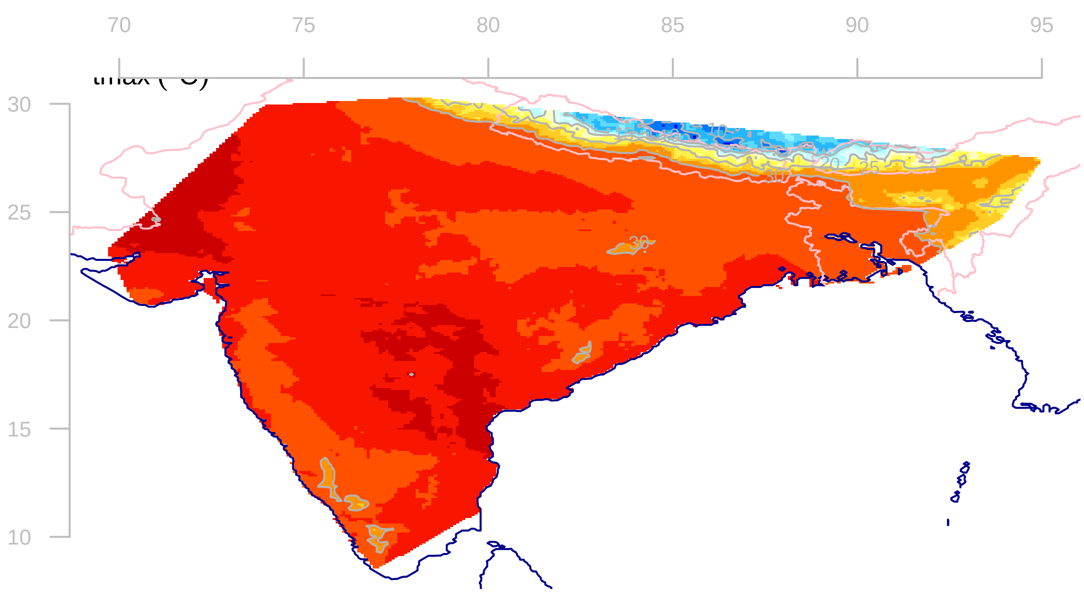
Temperature maps for the present.

Maps of the annual mean observed daily maximum and minimum temperatures from GHCN for the period 1960-2015.

```r
##
ztmax <- gridmap(tmax)
```

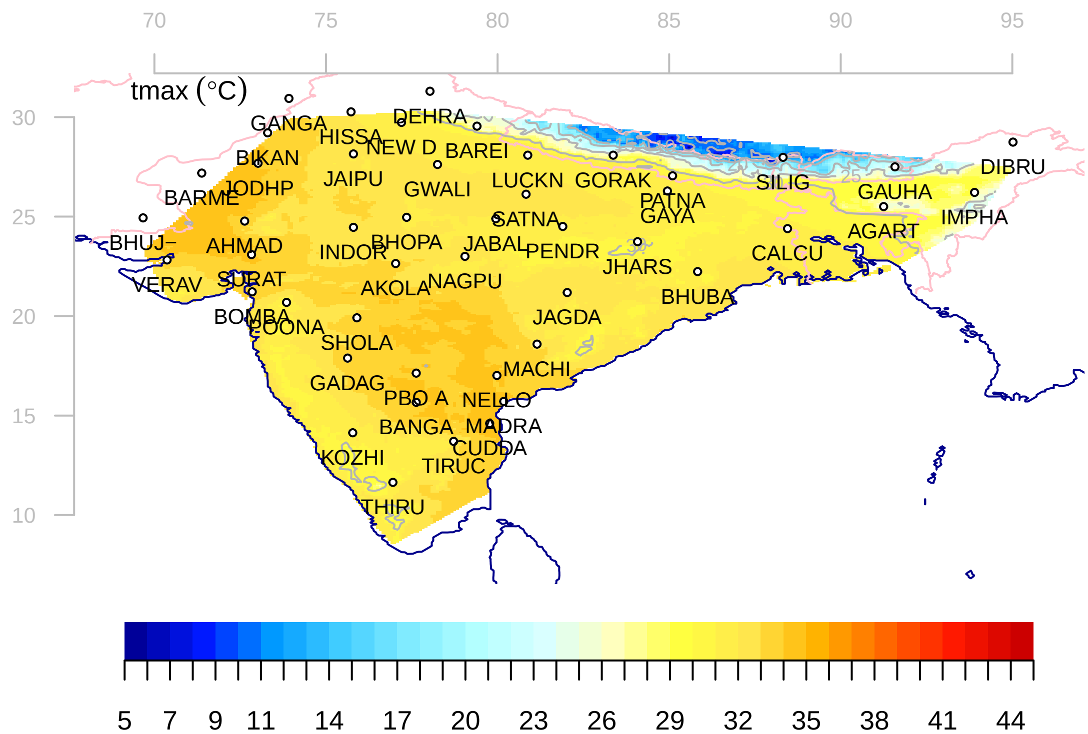
```
## Loading required package: LatticeKrig

## Loading required package: spam

## Loading required package: grid

##
## Attaching package: 'grid'
```

```
## The following object is masked from 'package:esd':
##
##     unit

## Spam version 2.1-4 (2018-04-12) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: fields

## Loading required package: maps

##
## Attaching package: 'maps'

## The following object is masked from 'package:esd':
##
##     map

## See www.image.ucar.edu/~nychka/Fields for
##  a vignette and other supplements.

##
## Attaching package: 'fields'

## The following objects are masked from 'package:esd':
##
##     image.plot, imageplot.info, imageplot.setup, poly.image
```




```r
map(ztmax,colbar=list(breaks=seq(5,45,by=1)),new=FALSE)
par(fig=c(0,1,0.1,1))
```

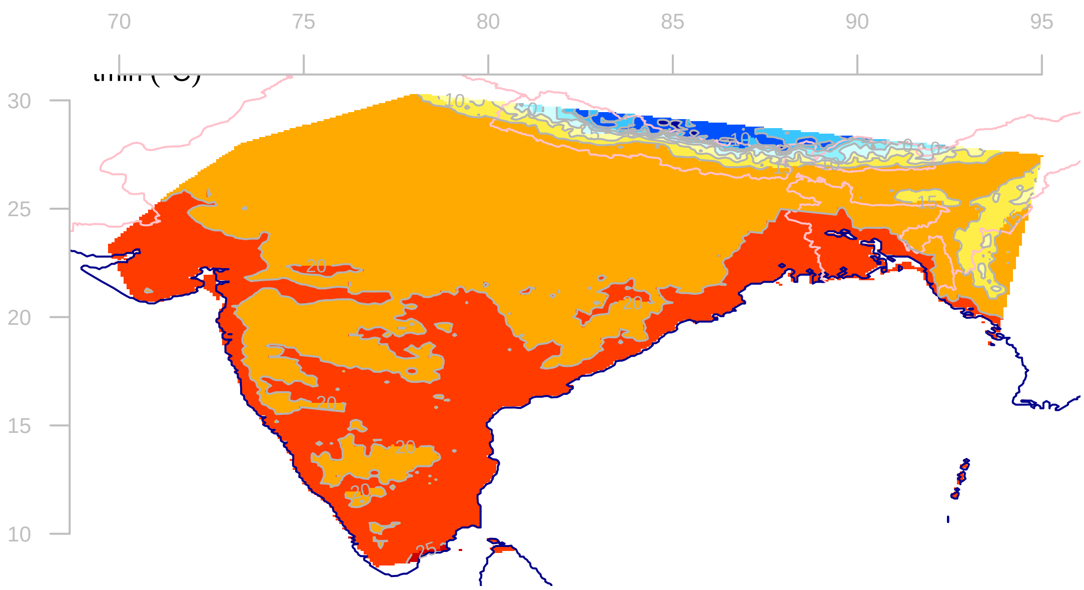
```
points(lon(tmax),lat(tmax),pch=19,col='white',cex=0.5)
points(lon(tmax),lat(tmax),cex=0.5)
text(lon(tmax),lat(tmax),substr(loc(tmax),1,5),cex=0.7,pos=1)
```



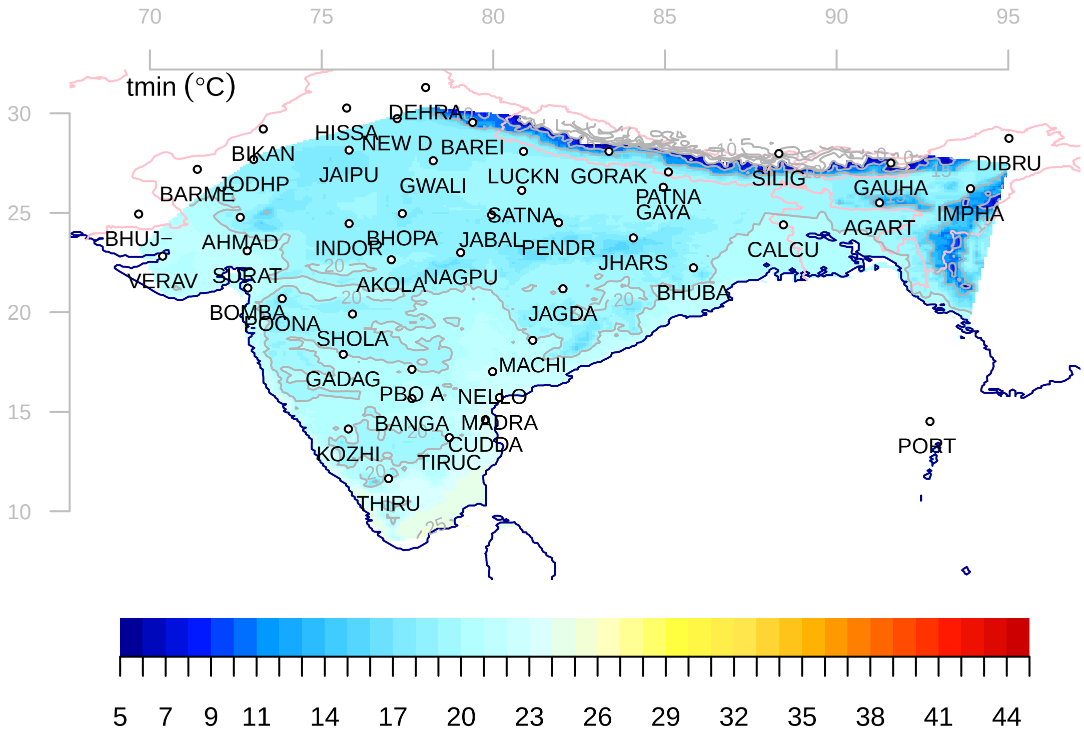

```
##
ztmin <- gridmap(tmin)
```

```
## Loading required package: LatticeKrig

## Loading required package: spam

## Loading required package: grid

##
## Attaching package: 'grid'

## The following object is masked from 'package:esd':
##
##     unit

## Spam version 2.1-4 (2018-04-12) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: fields
```

```
## Loading required package: maps
##
## Attaching package: 'maps'
## The following object is masked from 'package:esd':
##
##     map
## See www.image.ucar.edu/~nychka/Fields for
##  a vignette and other supplements.
##
## Attaching package: 'fields'
## The following objects are masked from 'package:esd':
##
##     image.plot, imageplot.info, imageplot.setup, poly.image
```




```
map(ztmin,colbar=list(breaks=seq(5,45,by=1)),new=FALSE)
par(fig=c(0,1,0.1,1))
points(lon(tmin),lat(tmin),pch=19,col='white',cex=0.5)
points(lon(tmin),lat(tmin),cex=0.5)
text(lon(tmin),lat(tmin),substr(loc(tmin),1,5),cex=0.7,pos=1)
```

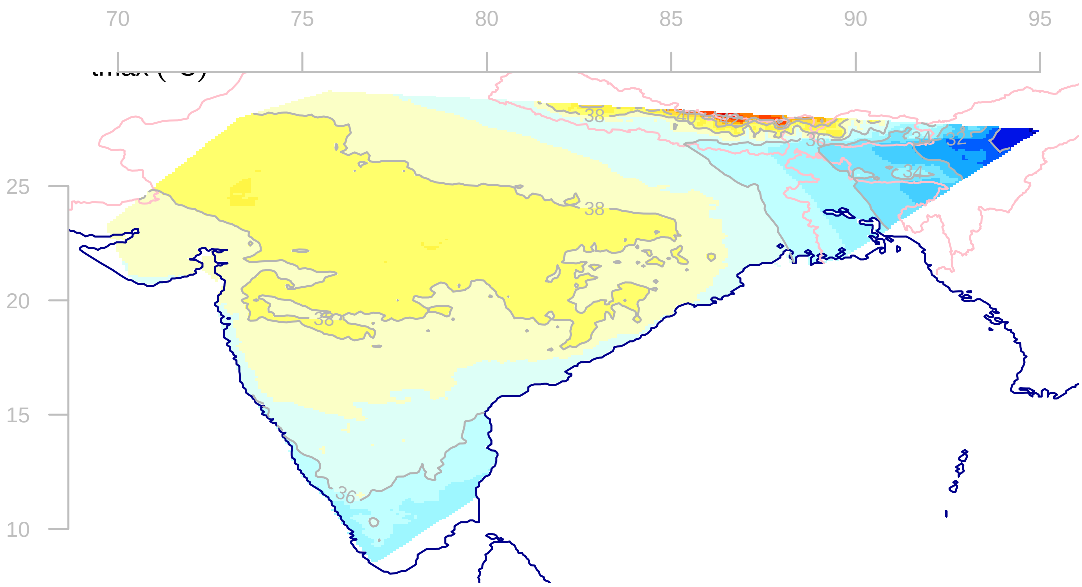


We also produced maps for projected Feb-April mean $\overline{T[max]}$ for the sake of completeness. They are based on the downscaled results for the PCA. The station structure of results was recovered before the grididng, and below is a map of downscaled daily maximum temperature for 2099 assuming the RCP8.5 emission scenario.

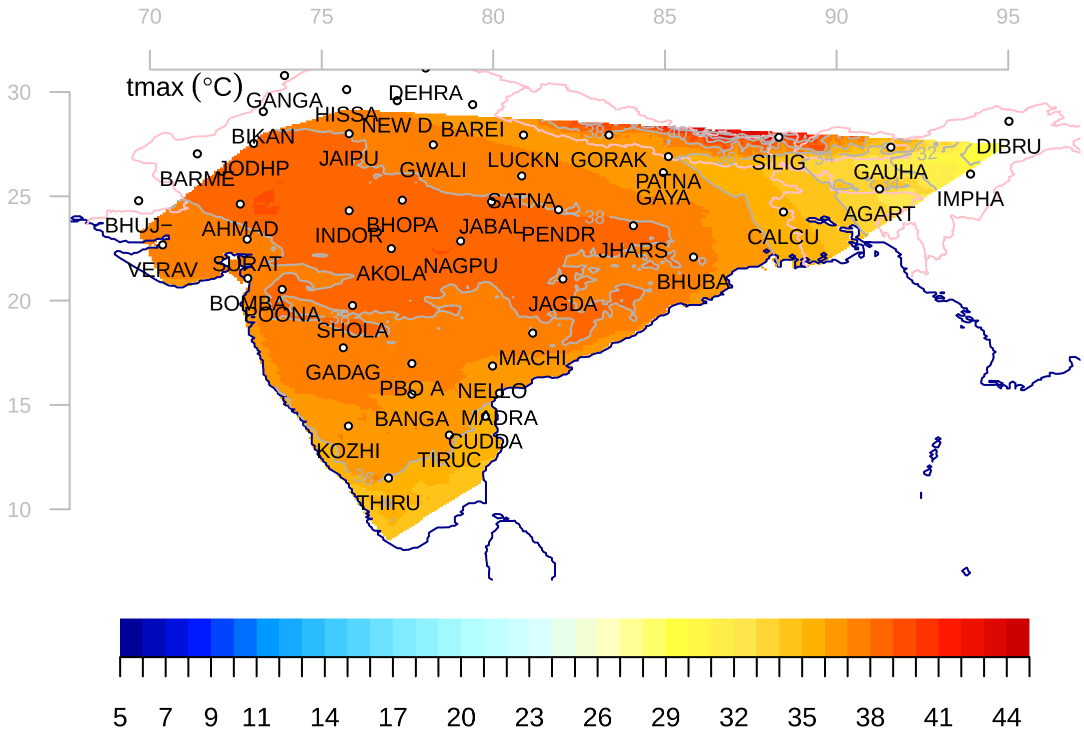
```
dsetmax.2099 <- gridmap(subset(as.station(as.station(dse.tmax.india.rcp85)),it=2099))
```

```
## Loading required package: LatticeKrig

## Loading required package: spam

## Loading required package: grid

##
## Attaching package: 'grid'

## The following object is masked from 'package:esd':
##
##     unit

## Spam version 2.1-4 (2018-04-12) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: fields

## Loading required package: maps
```

```
##
## Attaching package: 'maps'
## The following object is masked from 'package:esd':
##
##     map
## See www.image.ucar.edu/~nychka/Fields for
##  a vignette and other supplements.
##
## Attaching package: 'fields'
## The following objects are masked from 'package:esd':
##
##     image.plot, imageplot.info, imageplot.setup, poly.image
```




```
map(dsetmax.2099,colbar=list(breaks=seq(5,45,by=1)),new=FALSE)
par(fig=c(0,1,0.1,1))
points(lon(tmax),lat(tmax),pch=19,col='white',cex=0.5)
points(lon(tmax),lat(tmax),cex=0.5)
text(lon(tmax),lat(tmax),substr(loc(tmax),1,5),cex=0.7,pos=1)
```



Map of temperature change between 2010 and 2099:

```
dsetmax.2010 <- gridmap(subset(as.station(as.station(dse.tmax.india.rcp85)),it=2010))
```

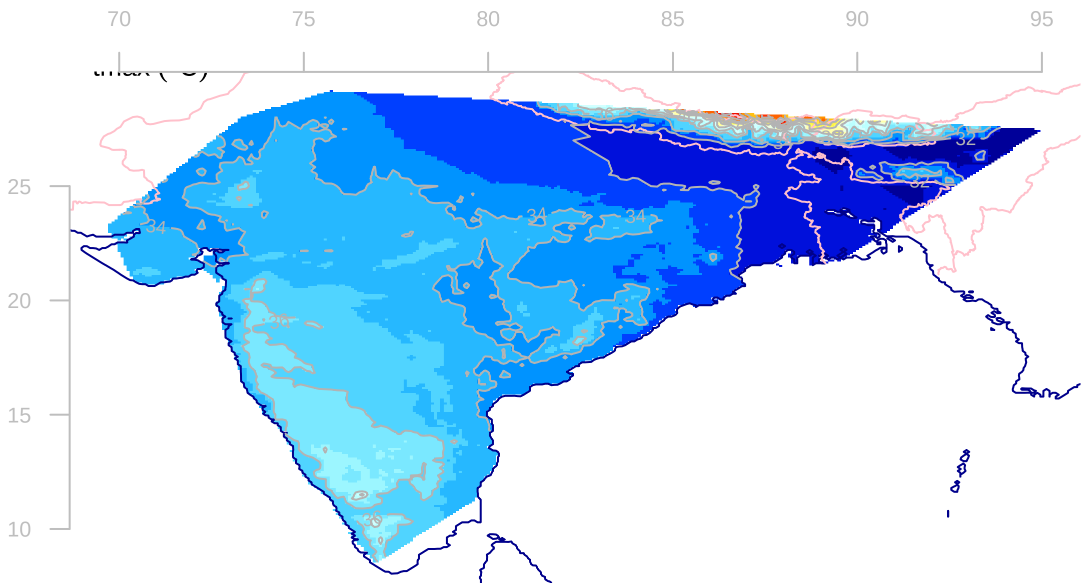
```
## Loading required package: LatticeKrig

## Loading required package: spam

## Loading required package: grid

##
## Attaching package: 'grid'

## The following object is masked from 'package:esd':
##
##     unit

## Spam version 2.1-4 (2018-04-12) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: fields

## Loading required package: maps

##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:esd':
##
##     map
```

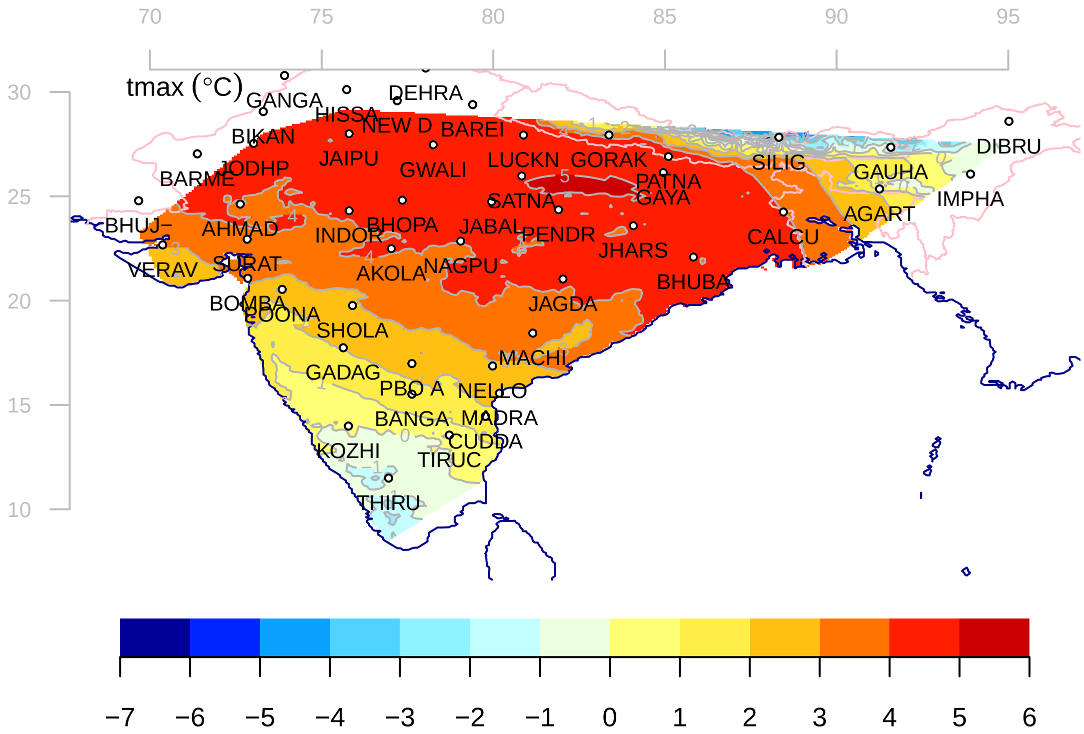
```
## See www.image.ucar.edu/~nychka/Fields for
##  a vignette and other supplements.
```

```
##
## Attaching package: 'fields'
```

```
## The following objects are masked from 'package:esd':
##
##     image.plot, imageplot.info, imageplot.setup, poly.image
```




```r
dsetmax.2010 <- dsetmax.2099 - dsetmax.2010
map(dsetmax.2010,new=FALSE)
par(fig=c(0,1,0.1,1))
points(lon(tmax),lat(tmax),pch=19,col='white',cex=0.5)
points(lon(tmax),lat(tmax),cex=0.5)
text(lon(tmax),lat(tmax),substr(loc(tmax),1,5),cex=0.7,pos=1)
```



Supporting analysis

Some supporting material is presented below, based on similar calculations applied to European data (ECA&D) that provide larger samples and better control of the data quality, i.e. less missing data and less need to fill in data voids.

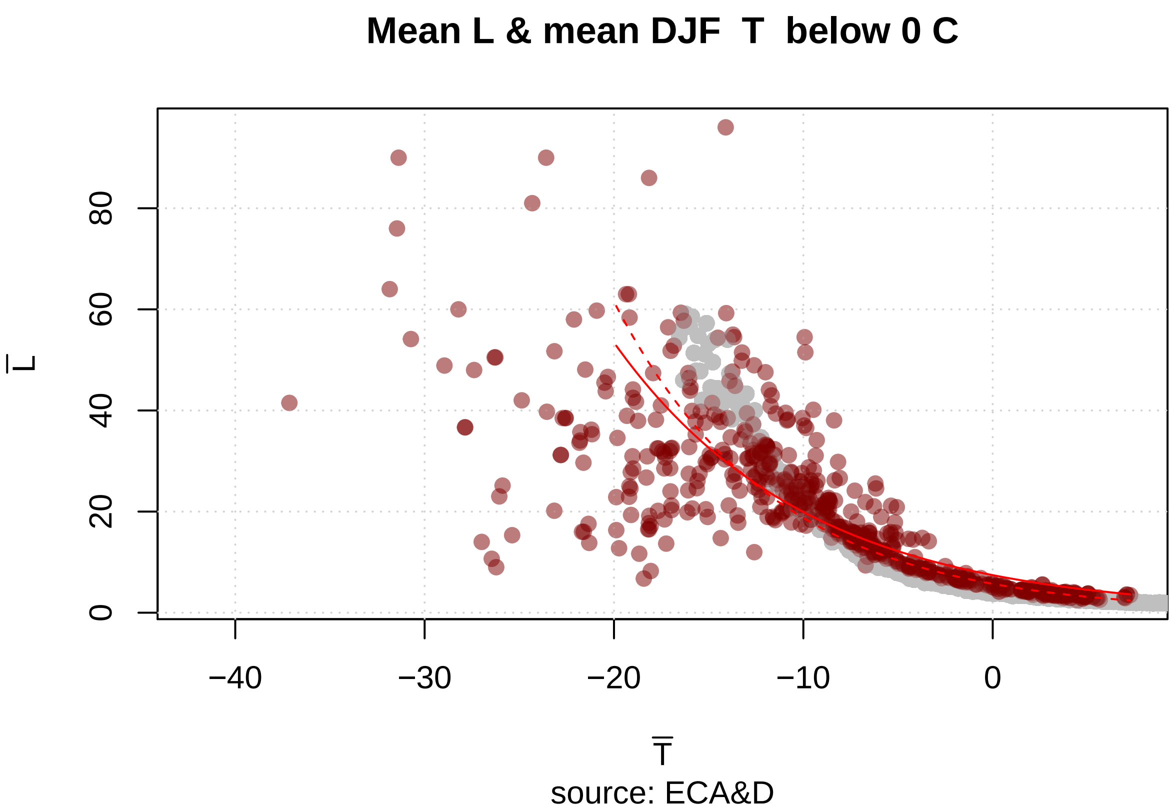
```
## Function: autocorrelation

AR <- function(n,mean=1,sd=1,a1=0.8) {
  rn <- rnorm(n,mean=mean,sd=sd)
  for (i in 2:n) rn[i] <- (a1*rn[i-1] + (1-a1)*rn[i])
  invisible(rn)
}
```

Compare the mean temperature and spell length statistics for data in Europe to see if similar dependencies exist outside India. This is a supporting analysis which can lend some confidence to the results for India. The quality of the Indian data is unknown, whereas the European observational time series have gone through some quality control and homogeneity checks and have less missing data points. We use different temperature thresholds for the European data and look at both cold and warm spells.

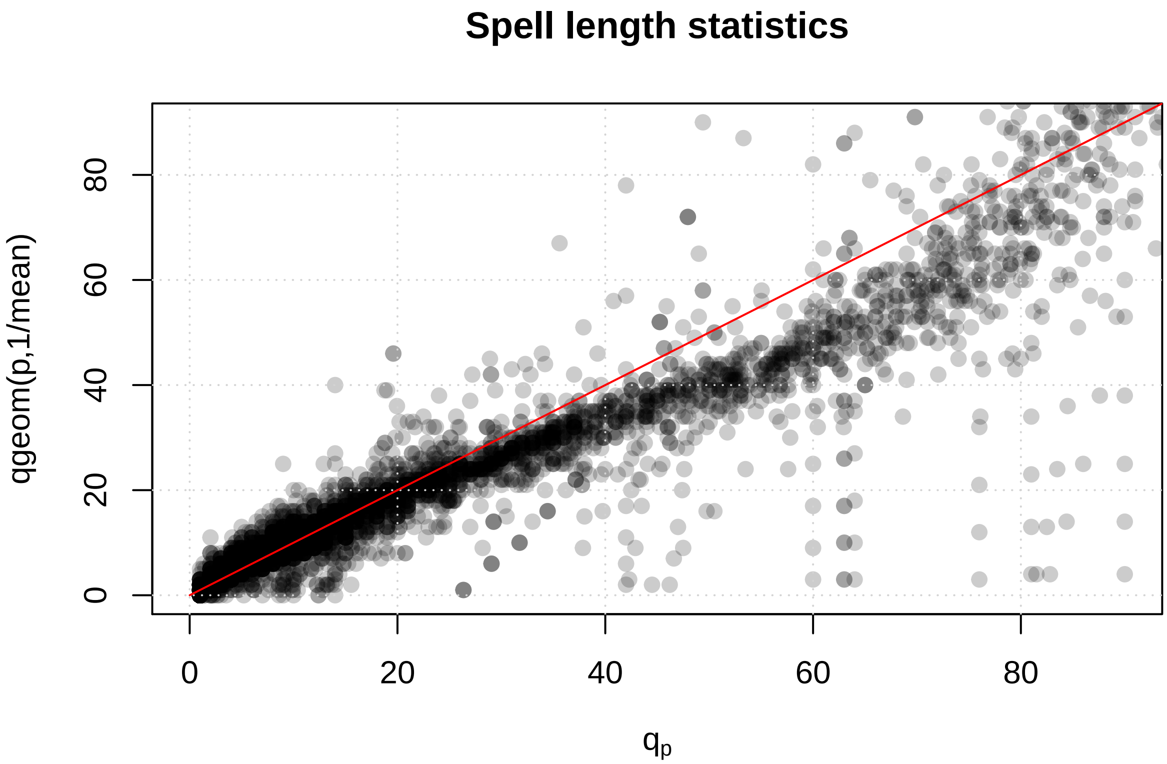
```
## Script that reads European temperature data and explores the connection between
## the seasonal mean temperature and the mean length of the warm/cold spells

for (it in c('djf','jja')) {

  if (it =='djf') {
    cold <- TRUE
    threshold <- 0
    is <- 2
  } else {
    cold <- FALSE
```

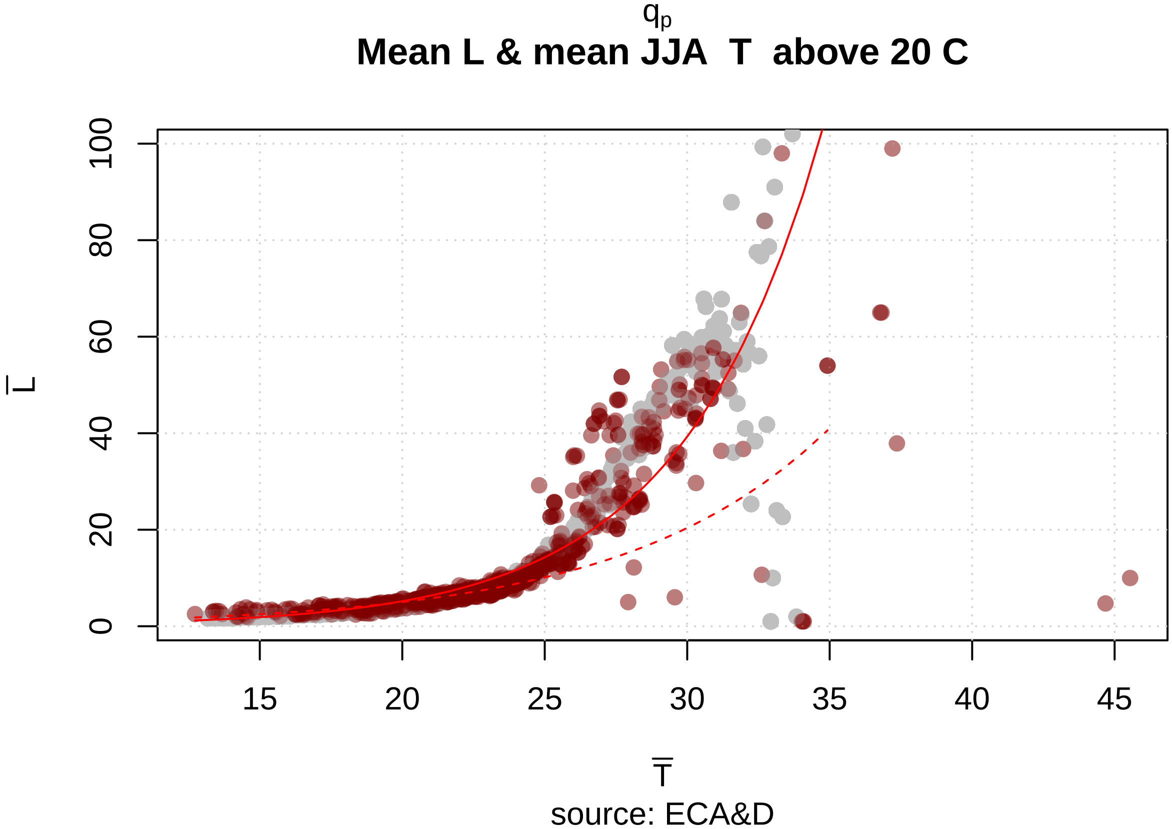
```
  threshold <- 20
  is <- 1
}

ss <- select.station(src='ecad',param='tx',nmin=75)
d <- dim(ss)
x <- rep(NA,d[1]); y <- x; std <- y
q.spell <- rep(NA,d[1]*10); dim(q.spell) <- c(d[1],10); q.geom <- q.spell

if (!file.exists(paste('ecad.tg.',it,'.rda',sep=''))) {
  for (i in seq(d[1])) {
    z <- station(ss[i,])
    print(loc(z))
    ## Make sure that there are values above and below the given threshold - otherwise
    ## spell will not work.
    if ( (sum(z > threshold,na.rm=TRUE)>1000) & (sum(z < threshold,na.rm=TRUE)> 1000) ) {
      s <- spell(z,threshold=threshold)
      ## Quality check: durations longer than a season (100 days) are not credible
      sc <- coredata(s); sc[sc > 100] <- NA; sc -> coredata(s)
      y[i] <- mean(subset(subset(s,is=is),it=it),na.rm=TRUE)

      ## Compare the spell-distribution with a geometric distribution
      q.spell[i,] <- quantile(subset(subset(s,is=is),it=it),probs=seq(0.05,0.95,by=0.1),na.rm=TRUE)
      q.geom[i,] <- qgeom(p=seq(0.05,0.95,by=0.1),prob=1/(y[i]))
      std[i] <- sd(subset(z,it=it),na.rm=TRUE)
      x[i] <- mean(subset(z,it=it),na.rm=TRUE)
    }
  }
  save(x,y,s,std,q.spell,q.geom,file=paste('ecad.tg.',it,'.rda',sep=''))
} else load(paste('ecad.tg.',it,'.rda',sep=''))
## Plot results

x[x > 50] <- NA              ## Removestations with crazy values
if (it=='djf') x[x>10] <- NA ## Remove stations with warm climate for the freezing analysis
plot(x,y,main=paste('Mean L & mean',toupper(it),' T ',
                 c('below','above')[c(cold,!cold)],threshold,'C'),
   sub='source: ECA&D',pch=19,col=rgb(0.5,0,0,0.3),
   xlab=expression(bar(T)),ylab=expression(bar(L)))
grid()

## Monte-Carlo simulations to compare spell length with
z <- station(ss[1,])
mstd <- 1.5*quantile(std,0.99,na.rm=TRUE)
if (!cold) mx <- mean(x,na.rm=TRUE) else mx <- 0
nmc <- 300
ymc <- rep(NA,nmc); xmc <- ymc
for (i in 1:nmc) {
  m <- seq(mx-mstd,mx+mstd,length=nmc)[i]
  coredata(z) <- AR(length(z),mean=m,sd=mstd,a1=0.7)
  s <- spell(z,threshold=threshold)
  if (length(s) > 0) {
    ymc[i] <- mean(subset(subset(s,is=is),it=it),na.rm=TRUE)
    xmc[i] <- m
```

```
      points(xmc,ymc,pch=19,col='grey75')
    }
  }

  points(x,y,pch=19,col=rgb(0.5,0,0,0.3))

  ix <- order(x); x <- x[ix]; y <- y[ix]
  ok <- is.finite(x) & is.finite(y)
  x <- x[ok]; y <- y[ok]; std <- std[ok]
  calfit <- data.frame(x=x[(x > -20) & (x < 35)],y=y[(x > -20) & (x < 35)]) ## Exclude the noisy tails
  calfit2 <- data.frame(x=x[(x > -20) & (x < 35)],y=1/y[(x > -20) & (x < 35)]) ## Exclude the noisy tai
  attr(calfit,'max(x)') <- max(x,na.rm=TRUE)
  #fit <- lm(y ~ I(x) + I(x^2),data=calfit)
  fit <- glm(y ~ x,data=calfit,family='poisson')
  #fit <- glm(y ~ x,data=calfit,family='binomial')
  fit2 <- glm(y ~ x, data=calfit2,family=negative.binomial(1))
  print(summary(fit))
  lines(calfit$x,exp(predict(fit)),col='red')
  lines(calfit$x,1/exp(predict(fit2)),col='red',lty=2)
  #lines(1/calfit$x,exp(predict(fit)),col='red')
  #dev.copy2pdf(file=paste('fig1',c('a','b')[c(cold,!cold)],'.pdf',sep=''))

  attr(x,'description') <- paste(it,'mean temperature (degC)')
  if (cold) attr(y,'description') <- 'mean cold spell length (days)' else
            attr(y,'description') <- 'mean warm spell length (days)'
  attr(y,'threshold') <- threshold
  attr(x,'label') <- expression(bar(T))
  attr(x,'Monte-Carlo') <- xmc
  attr(y,'label') <- expression(bar(tau[T < T0]))
  attr(y,'Monte-Carlo)') <- ymc
  meanspell <- data.frame(meanT=x,meanL=y,std=std)
  attr(meanspell,'fit') <- fit
  attr(meanspell,'geometric.fit') <- data.frame(q.spell=q.spell,q.geom = q.geom)
  save(meanspell,file=paste('meanspell',it,c('below','above')[c(cold,!cold)],threshold,'.rda',sep=''))

  ## Test if the spell length statistics is close to geometric
  plot(c(q.spell),c(q.geom),main='Spell length statistics',
      xlim=range(0,90),ylim=range(0,90),pch=19,col=rgb(0,0,0,0.2),
      xlab=expression(q[p]),ylab='qgeom(p,1/mean)')
  grid()
  lines(range(q.spell,q.geom,na.rm=TRUE),range(q.spell,q.geom,na.rm=TRUE),col='red')
  #dev.copy2pdf(file=paste('fig3',c('a','b')[c(cold,!cold)],'.pdf',sep=''))
}
```




```
## [1] "Retrieving data from 1 records ..."
## [1] "1 TMAX 100010 STENSELE SWEDEN ECAD"
##
## Call:
## glm(formula = y ~ x, family = "poisson", data = calfit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.2225  -0.8126  -0.4765   0.5969   9.5870
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.006530   0.019382  103.53   <2e-16 ***
## x           -0.098594   0.001551  -63.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6158.8  on 608  degrees of freedom
## Residual deviance: 1602.5  on 607  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 4
```






```
## [1] "Retrieving data from 1 records ..."
## [1] "1 TMAX 100010 STENSELE SWEDEN ECAD"
##
## Call:
```

```
## glm(formula = y ~ x, family = "poisson", data = calfit)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -13.0036   -0.6427   -0.3200    0.1469    4.7300
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.402910   0.067149  -35.78   <2e-16 ***
## x            0.202483   0.002515   80.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7919.7  on 706  degrees of freedom
## Residual deviance: 1488.6  on 705  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 5
```