Supporting material for Downscaling the probability of heavy precipitation

Rasmus E. Benestad

Oslo, Norway: 2024-03-05 - 2024-04-26

Contents

1 Introduction

This document can be regarded as a kind of lab notebook for an analysis of precipitation. It shows the merits of two key parameters, *the wet-day frequency* f_w and *wet-day mean precipitation* μ (also referred to as the 'mean precipitation intensity'), and is meant to facilitate an open and transparent process that can be readily replicated by others.

This document is an R-markdown script written for empirical-statistical downscaling of probabilities and statistical properties of heavy 24-hr precipitation in the Nordic countries, based on Benestad et al. (2019). The R-code and results provided herein show how statistical properties of daily precipitation can be estimated based on f_w and μ according to $Pr(X > x) = f_w \exp(-x/\mu)$. This script may be adapted to repeat the analysis for other regions of the world.

The white-paper of Euro-CORDEX (https://cordex.org/wp-content/uploads/2022/08/White-Paper-ESD.pdf) recommends combining empirical-statistical downscaling (ESD) and dynamical downscaling with regional climate models (RCMs), as these two approaches are based on different assumptions and have different strengths and weaknesses. Nevertheless, this study only involves ESD, as its purpose is to demonstrate how ESD can be used to downscale various statistics for 24-hr precipitation and provide information that may be useful for decision-making and climate change adaptation. This is also ment to be a contribution to CORDEX-ESD and Euro-CORDEX.

This work is meant to provide a basis for the report *'Klima i Norge 2100'* as well as preparing the ground for studying the connection between climate change and diarrhoea in the project EU-SPRINGS. It builds on the legacy of the projects KlimaDigital https://www.sintef.no/projectweb/klimadigital/, funded by the Norwegian Research Council (2018-2021), and EU-SPECS (FP7; 2012-2017).

1.1 Set-up: practical and technical considerations

This R-script saves intermediate results in temporary files (saved as R-binary data files `*.rda`) since some of the processing takes long time. It will look for these files and use them for the running of the script, but if they are not found, it will run through the original computations. The temporary files will be saved together with this R-markdown script for the benefit of those who want to replicate this analysis.

```r
## A demonstration of downscaling for seasonal mean temperature
## Rasmus.Benestad-at-met.no 2023-0
## For ICRC CORDEX 2023 Session-D

## The first lines of code prepare the data for empirical-statistical
## downscaling of a CMIP6 ensemble

## Needs the R-package 'esd' available from https://github.com/metno/esd
## Check if you need to get the devtools-package:
install.esd <- ("esd" %in% rownames(installed.packages()) == FALSE)
if (install.esd) {
  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', dependencies = TRUE)
  }

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

```
    install_github('metno/esd')
}

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
```

```
## Registered S3 methods overwritten by 'esd':
##   method        from
##   subset.matrix base
##   subset.zoo    zoo
```

```
## Working directory where the temporary data files are stored
setwd('~/R')
print(sessionInfo()$otherPkgs$esd)
```

```
## Package: esd
## Version: 1.10.79
## Date: 2024-04-29
## Title: Climate analysis and empirical-statistical downscaling (ESD)
##         package for monthly and daily data
## Author: Rasmus E. Benestad, Abdelkader Mezghani, Kajsa M. Parding,
##         Helene B. Erlandsen, Ketil Tunheim, and Cristian Lussana
## Maintainer: Rasmus E. Benestad <rasmus.benestad@met.no>
## Depends: R (>= 4.00.0), ncdf4, zoo
## Imports: graphics, grDevices, stats, utils
## Suggests: LatticeKrig, fields, PCICt, jpeg, jsonlite
## Description: The package contains R functions for retrieving data,
##         making climate analysis and downscaling of monthly mean and
##         daily mean global climate scenarios.
## License: GPL (>= 2)
## URL: http://github.com/metno/esd
## Encoding: UTF-8
## ZipData: no
## RoxygenNote: 7.3.1
## RemoteType: github
## RemoteHost: api.github.com
## RemoteRepo: esd
## RemoteUsername: metno
## RemoteRef: HEAD
## RemoteSha: 5c1b5b4d1ea51e25a9687abef11c05ec8d27449a
## GithubRepo: esd
```

```
## GithubUsername: metno
## GithubRef: HEAD
## GithubSHA1: 5c1b5b4d1ea51e25a9687abef11c05ec8d27449a
## NeedsCompilation: no
## Packaged: 2024-05-02 06:00:45 UTC; rasmusb
## Built: R 4.4.0; ; 2024-05-02 06:01:01 UTC; unix
##
## -- File: /home/rasmusb/R/x86_64-pc-linux-gnu-library/4.4/esd/Meta/package.rds
```

1.2 Settings for downscaling

The choice of predictor domain used for the downscaling was set to match that used for evaluating the model results, shown at the end of this R-markdown script.

The evaluation shown near the end of this R-markdown script reveals an outlier: CESM2-WACCM-FV2. We exclude it from the projections, even if the ensemble itself is getting a bit small.

```
recompute <- FALSE
## The region of predictors from GCMs evaluated at the end of this R-markdows script
regions <- "-5-45E.55-72N"
## Runs which caused the running of the code to crash - to be excluded
bad.runs <- c("EC-E","FGOA","IPSL","KACE","MPI-","UKES")
outlier.runs <- c("CESM")
## Define the spatial domain for the predictor: same as the domain used for evaluation
## of CMIP6 GCMs.
lons <- c(-5,45)
lats <- c(55,72)
```

1.3 Preprocessing ERA5 24-hr precipitation

The ERA5 data (Hersbach and Dee (2016)) was downloaded from the Copernicus Climate Services (C3S) climate data store (CDS) site using `esd::ERA5.CDS()`:

```
ERA5.CDS(param='total_precipitation',varnm='tp',FUN='daysum')
```

The daily global ERA5 data was then aggregated into wet-day frequency f_w and wet-day mean precipitation μ (the threshold for a wet day was 1 mm/day):

```
## R-script to estimate wet-day mean and wet-day frequency from ERA5/EOBS.
## Rasmus Benestad 2023-05-26

clean <- TRUE

## This routine masks the data wet/dry and estimates monthly wet-day frequency
##vand mean precipitation intensity
wetdaymean <- function(ifile,datadir,outdir) {
  ## Make sure that all files have the same longitude/latitude setting as the first file
  print('Ensure correct grid - use the grid from ERA5_tp_1940_daysum.nc')
  system(paste0('cdo -b f32 remapcon,',file.path(datadir,'ERA5_tp_1940_daysum.nc '),
               file.path(datadir,ifile),' out.nc'))
  print('Make mask of precipitation')
  system('cdo gtc,0.001 out.nc mumask.nc')
```

```r
  ## The file mumask.nc is a mask with 0.001 (m) in all time steps with exceeding
  ## data (set to 1) and 0 for all others
  system(paste('cdo ifthen mumask.nc out.nc masked.nc'))
  ## her appliseres masken på ifile og lagres i ofile. (Øvrige punkter får verdien NA.)
  system(paste('cdo monmean masked.nc',file.path(outdir,sub('daysum.nc','mu.nc',ifile))))
  system(paste('cdo monmean mumask.nc',file.path(outdir,sub('daysum.nc','fw.nc',ifile))))
  system(paste('cdo -fldmean masked.nc',file.path(outdir,sub('daysum.nc','TP.wet.nc',ifile))))
  file.remove('mumask.nc')
  file.remove('masked.nc')
  file.remove('out.nc')
}

## Setup
print('Generate the statistics...')
datadir <- '~/data/ERA5/day'
outdir <- '~/Downloads/ERA5-precip-statistics'
old.files <- NULL
if (!exists(outdir)) dir.create(outdir) else {
  old.files <- list.files(path=outdir,full.names = TRUE)
  if (clean) file.remove(old.files)
}

## Save temporary files if the job is not completed...
print('Monthly wet-day-mean and wet-day frequency for each year...')
files <- list.files(path=datadir,pattern='_tp_')
if ((!is.null(old.files)) & (!clean) ) {
  donefiles <- sub('/home/rasmusb/Downloads/ERA5-precip-statistics/ERA5_tp_','',
                    old.files[grep('TP.wet.nc',old.files)])
  donefiles <- sub('_TP.wet.nc','',donefiles)
  files <- files[-match(donefiles,sub('_daysum.nc','',sub('ERA5_tp_','',files)))]
}
print(files)

for (file in files) {
  print(file)
  wetdaymean(file,datadir,outdir)
}
file.remove(file.path(outdir,"first.ERA5grid.nc"))

## Merge the annual files
print('Merge the results')
results <- list.files(path=outdir,pattern='mu.nc',full.names = TRUE)
system(paste('cdo mergetime',paste(results,collapse=' '),
             file.path('~/Downloads/','ERA5_mu_mon.nc')))
file.remove(results)
results <- list.files(path=outdir,pattern='fw.nc',full.names = TRUE)
system(paste('cdo mergetime',paste(results,collapse=' '),
             file.path('~/Downloads/','ERA5_fw_mon.nc')))
file.remove(results)
results <- list.files(path=outdir,pattern='TP.wet.nc',full.names = TRUE)
system(paste('cdo mergetime',paste(results,collapse=' '),
             file.path('~/Downloads/','ERA5_TP.wet.nc')))
file.remove(results)
```

```
## Make annual aggregation
system('cdo yearavg ~/Downloads/ERA5_fw_mon.nc ~/Downloads/ERA5_fw_year.nc')
system('cdo yearavg ~/Downloads/ERA5_mu_mon.nc ~/Downloads/ERA5_mu_year.nc')
```

The subsequent analysis and downscaling relied on aggregated daily ERA5 data processed as described above.

2 Downscaling

Annual local wet-day frequency f_w and wet-day mean precipitation μ were downscaled using f_w and μ respectively to represent the *large-scale predictors*, as shown below. The annual aggregates were used here as a starting point of this kind of analysis, but it may potentially be extended to seasonal statistics to capture aspects that may be stratified in terms of seasons. Extending this analysis to seasonal statistics will increase the volume of results and result in longer scripts and documents.

The predictors were based on ERA5 for calibration and daily precipitation from CMIP6 global climate models (GCMs; Eyring et al. (2015)) for projections. The Downscaling used common EOFs as framework, as explained in the discussion paper Benestad (2021). The common EOF framework provides an additional level of evaluation and quality control in addition a better way to combine GCM output and reanalysis.

```
ssps <- list.files(path='data/CMIP/CMIP6.monthly/wet-day-frequency/')
ssps <- ssps[grep('ssp|hist',ssps)]
print(ssps)
```

```
## [1] "hist"   "ssp126" "ssp245" "ssp370" "ssp585"
```

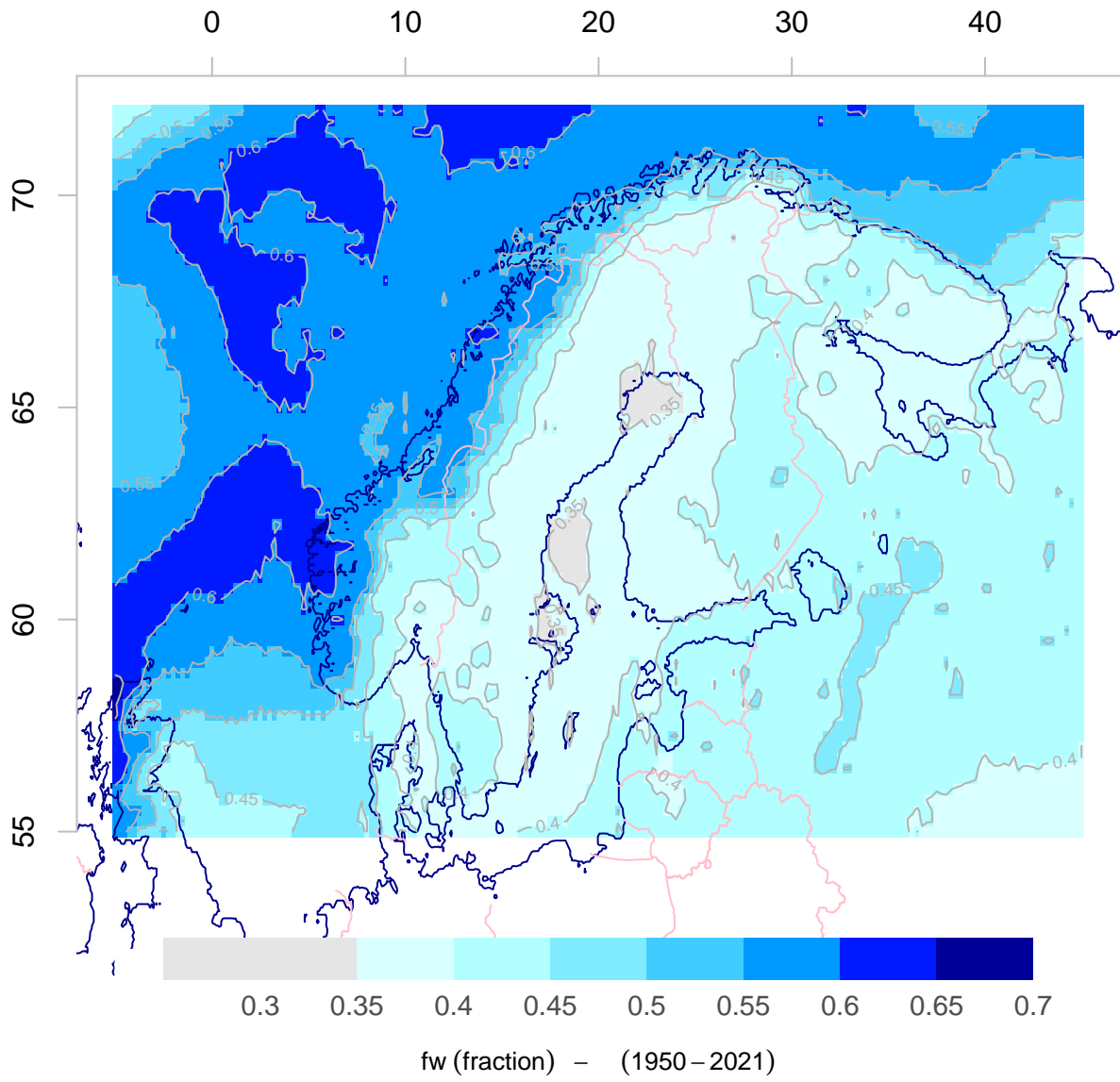
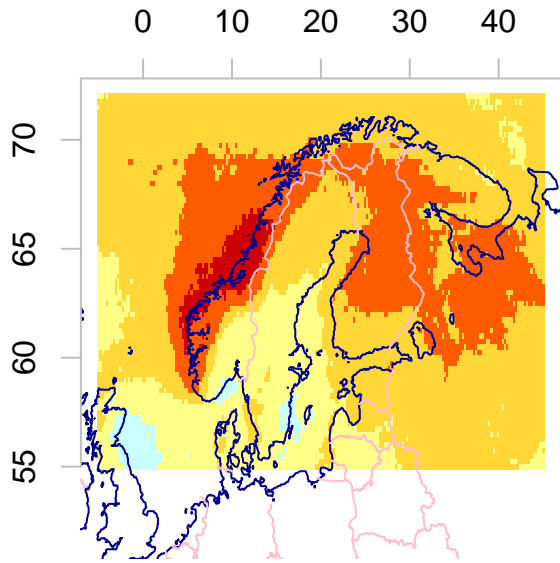
```
tmp.st.fil <- 'Y.dse4fwmu.rda'
it <- 'annual'
ip <- 1:7
esd.results <- list()

#ssp <- 'ssp370'
for (ssp in ssps) {
  for (param in c('fw','mu')) {
    reanalysis <- switch(param,
                         'fw'='~/data/ERA5/ERA5_fw_year.nc',
                         'mu'='~/data/ERA5/ERA5_mu_year.nc')
    ## ERA5 wet-day mean precipitation has units of m/day
    ## CMIP6 wet-day mean precipitation has units of kg/(m² s)
    dse.results <- switch(param,'fw'=paste0('dse4fw.',ssp,'.rda'),
                          'mu'=paste0('dse4mu.',ssp,'.rda'))
    FUN <- switch(param,'fw'='wetfreq', 'mu'='wetmean')
    path <- switch(param,'fw'='~/data/CMIP/CMIP6.monthly/wet-day-frequency/',
                   'mu'='~/data/CMIP/CMIP6.monthly/wet-day-intensity/')
    #file.remove(dse.results) ## uncomment this line if it should be repeated.
    if (!file.exists(dse.results) | recompute) {

      ## Read daily mean temperatures from Italy through the ECA&D database
      #Y <- station(param='precip',src='metnod.thredds')
      file.predictand <- '~/data/ECAD/precip.ecad.nc'
      varid <- 'precip'
```

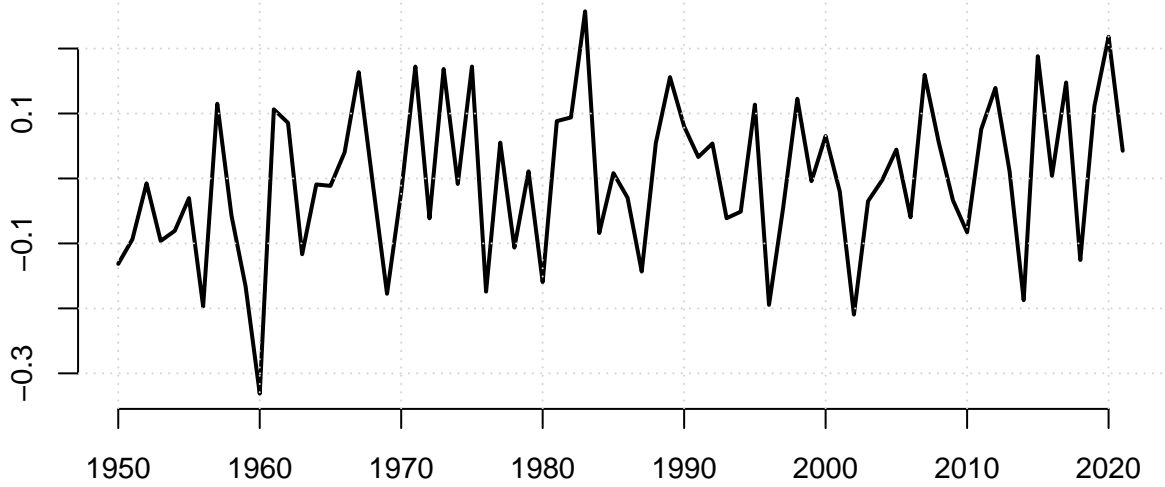
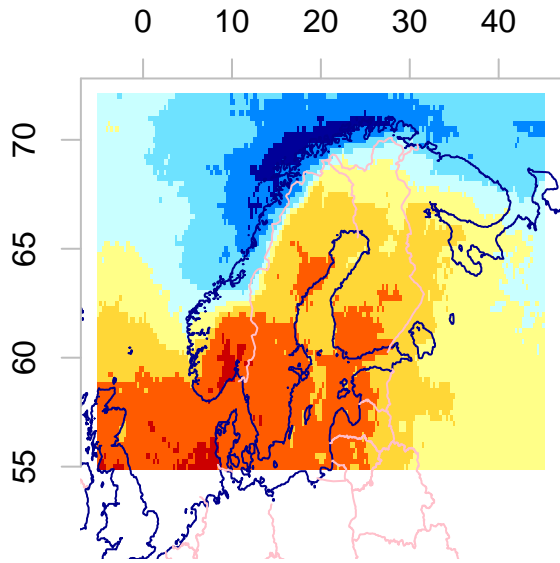
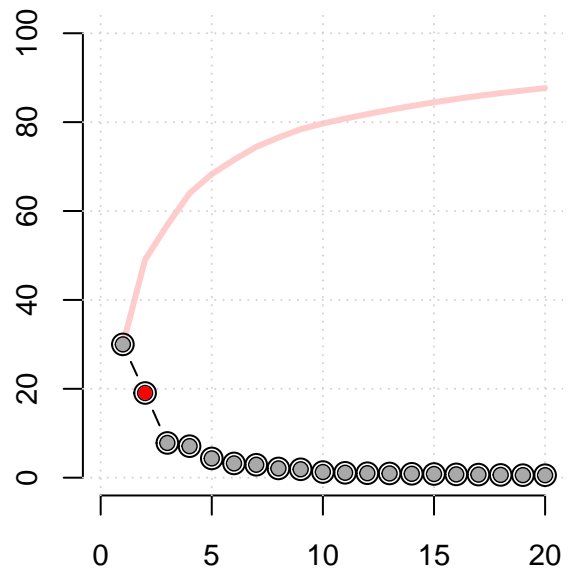
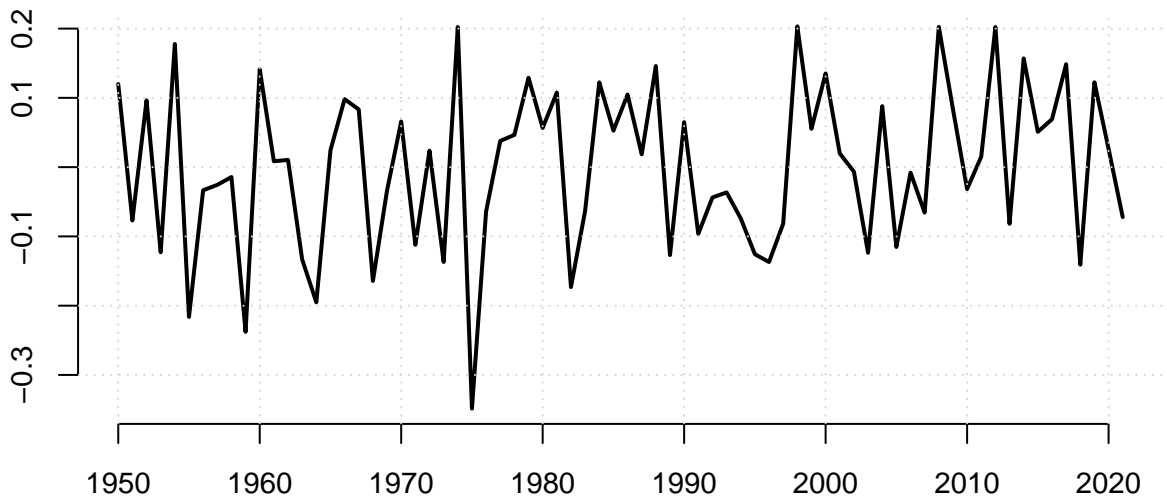
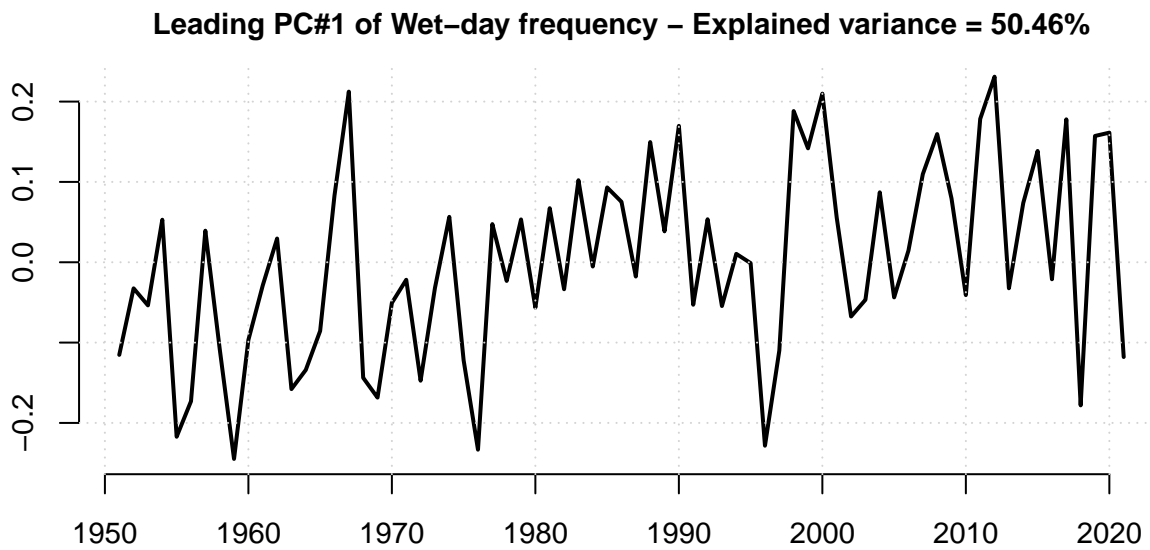
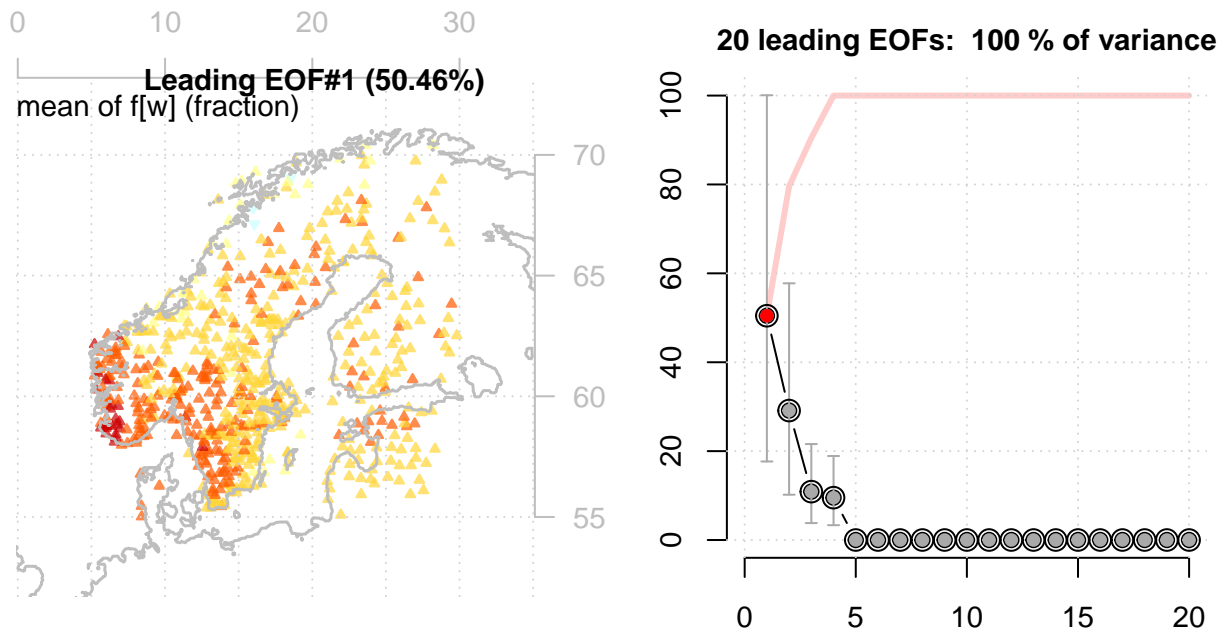
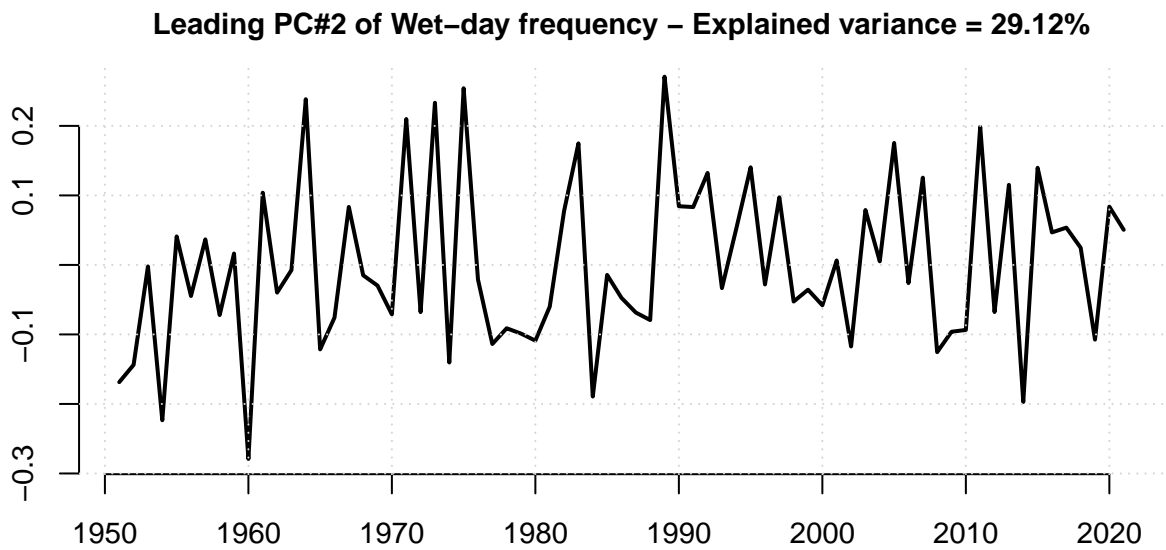
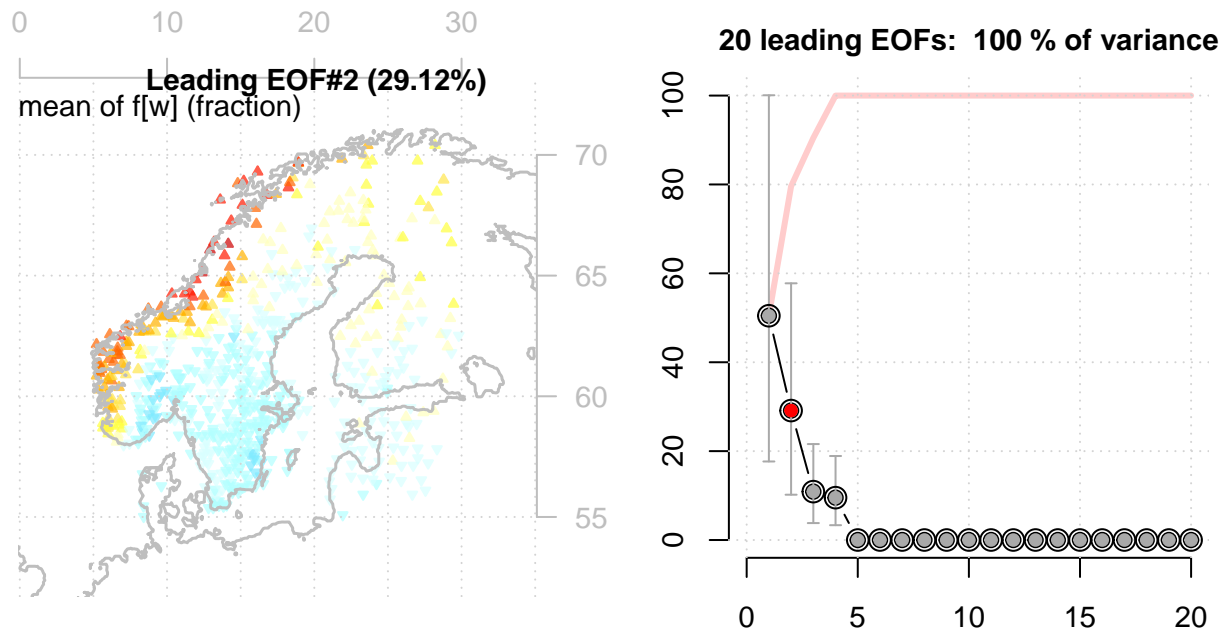
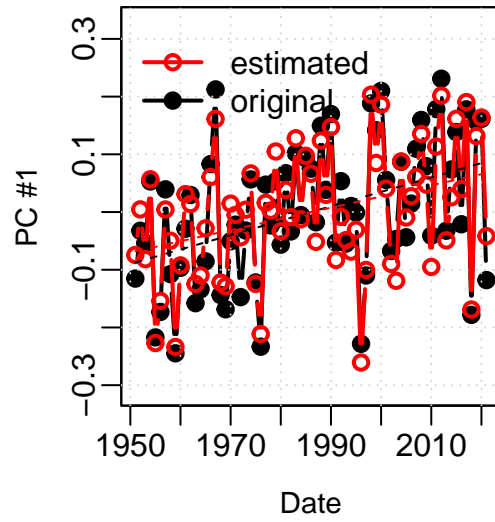
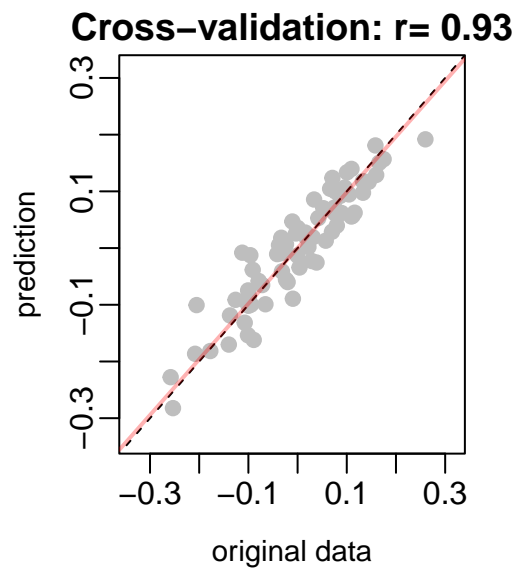
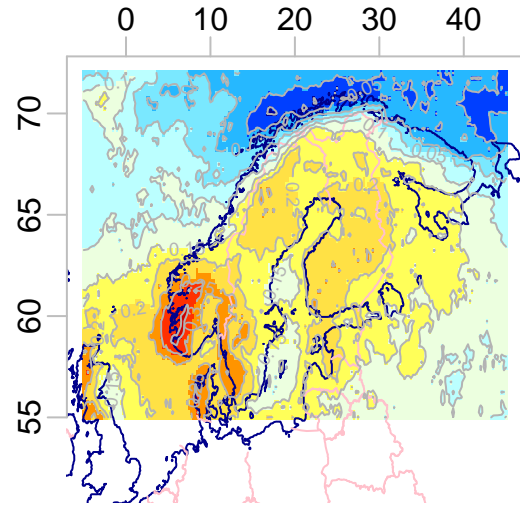
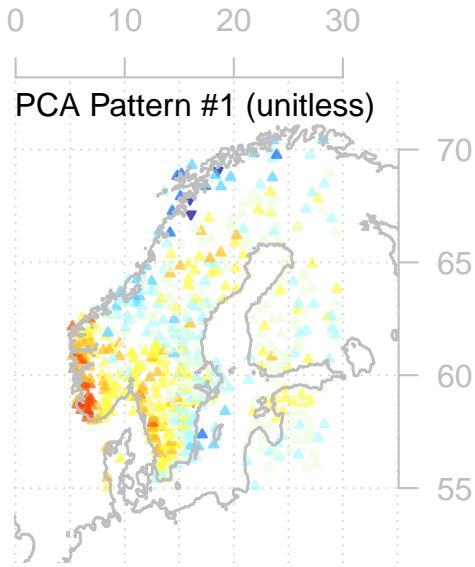
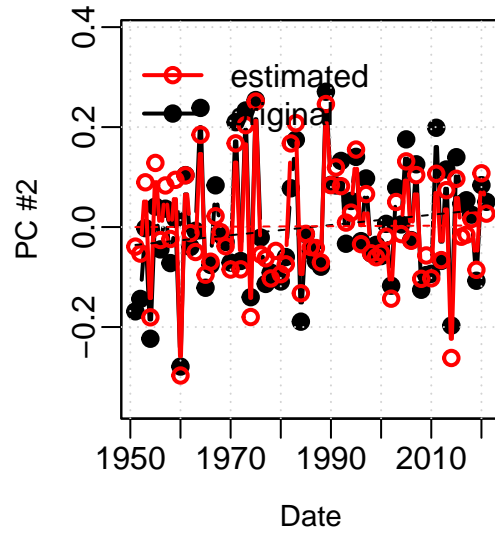
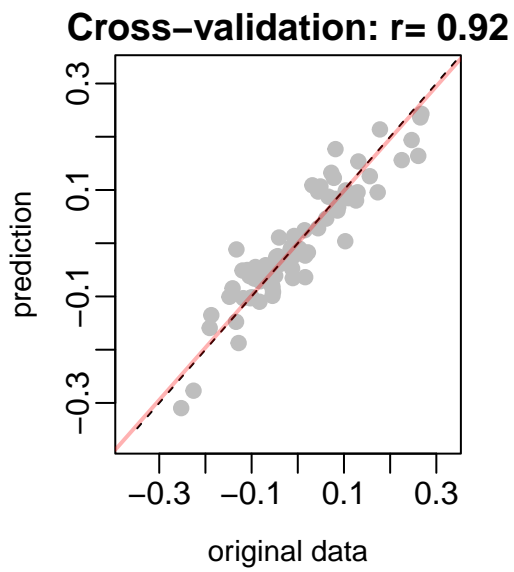
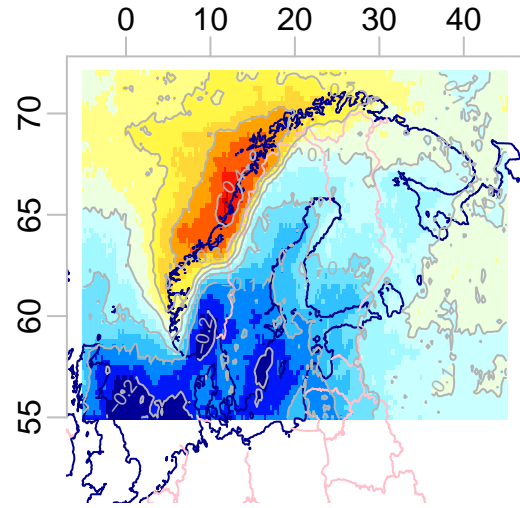
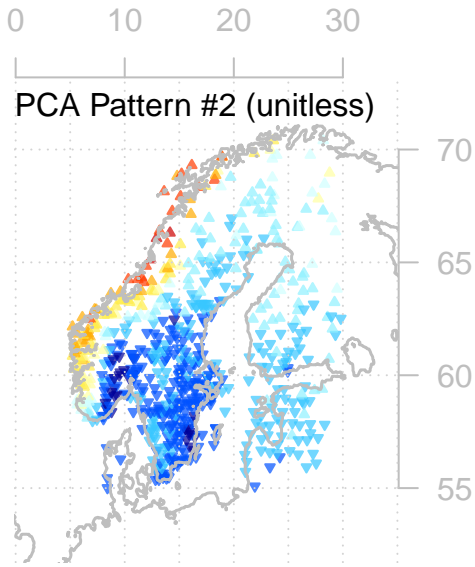
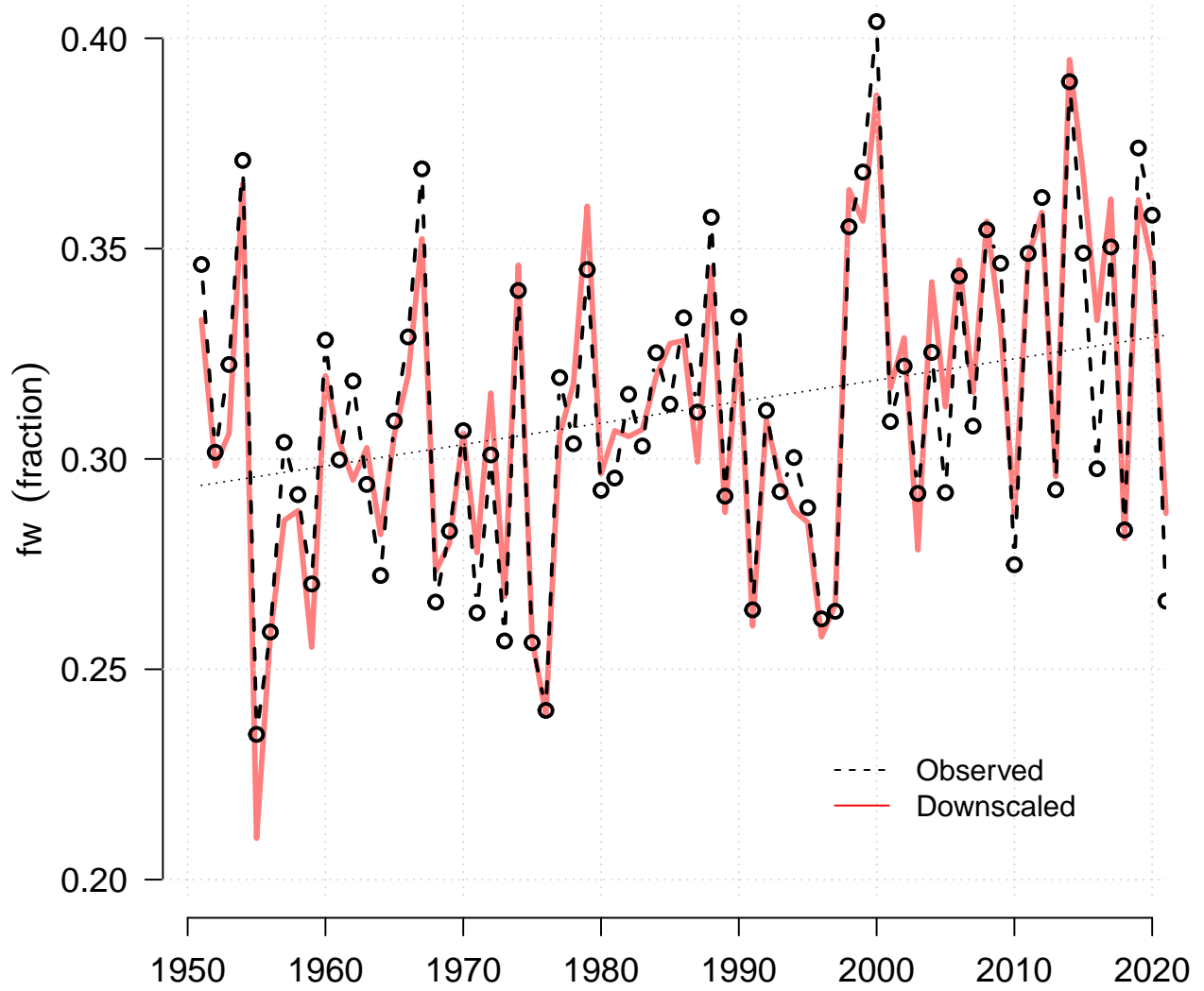
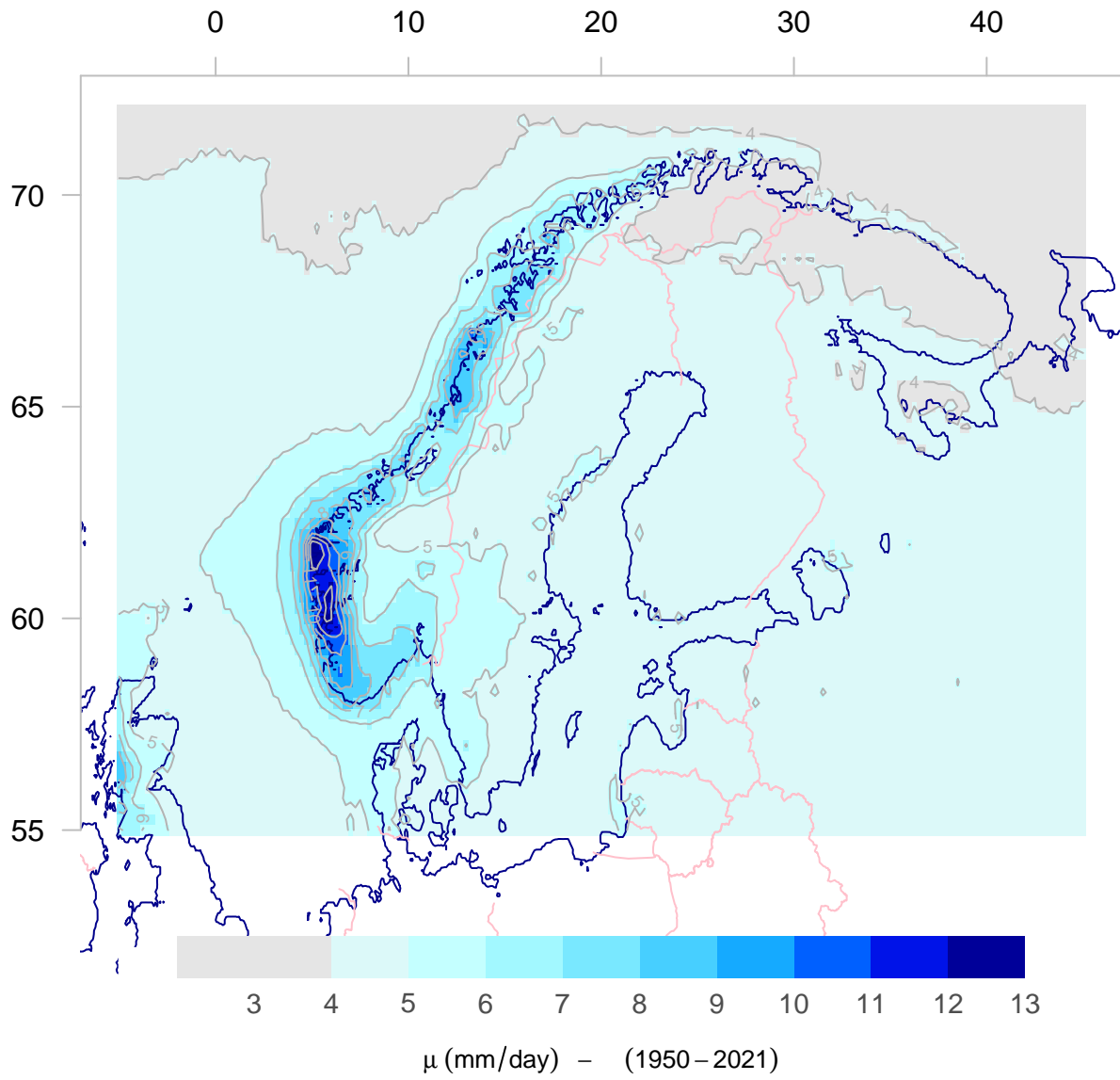
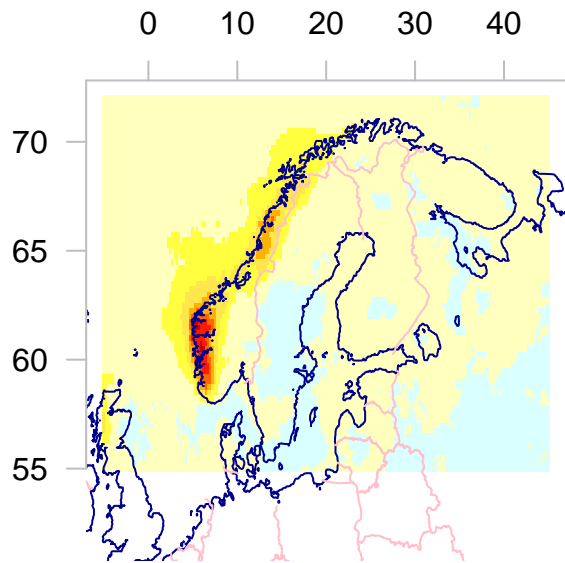
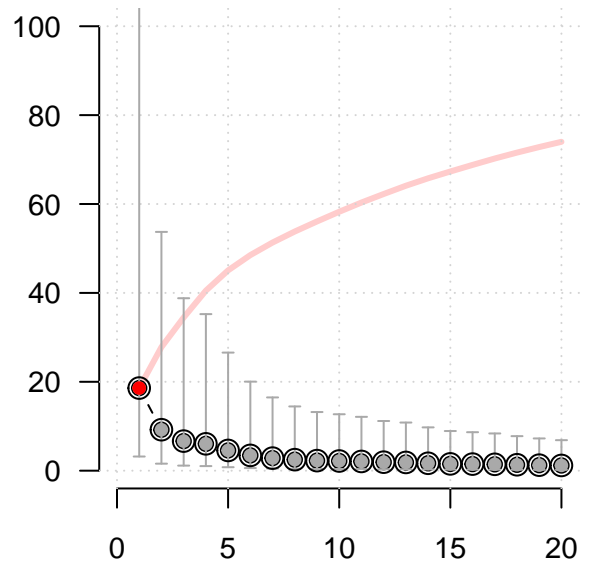
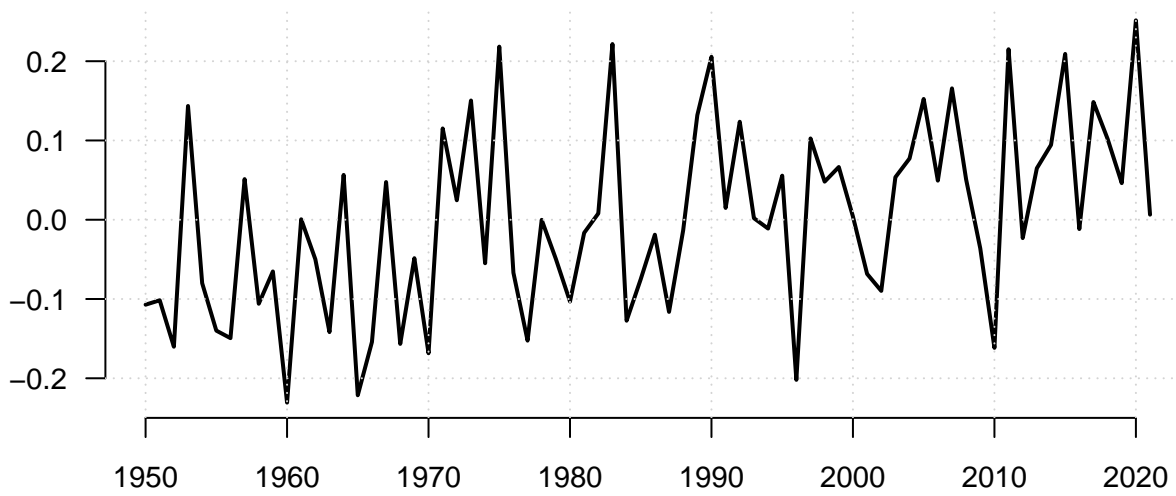
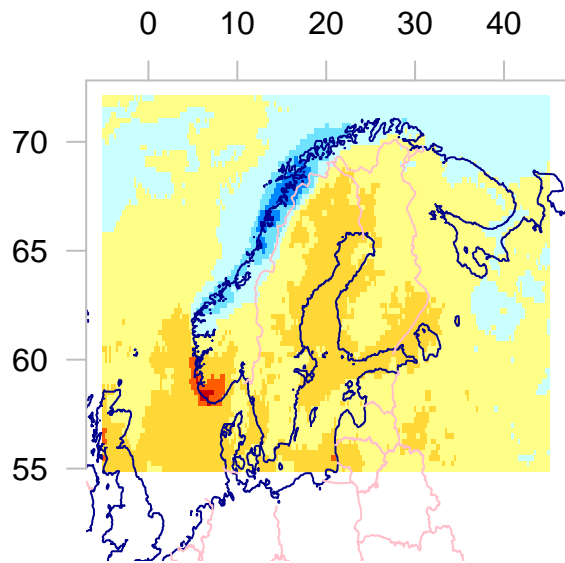
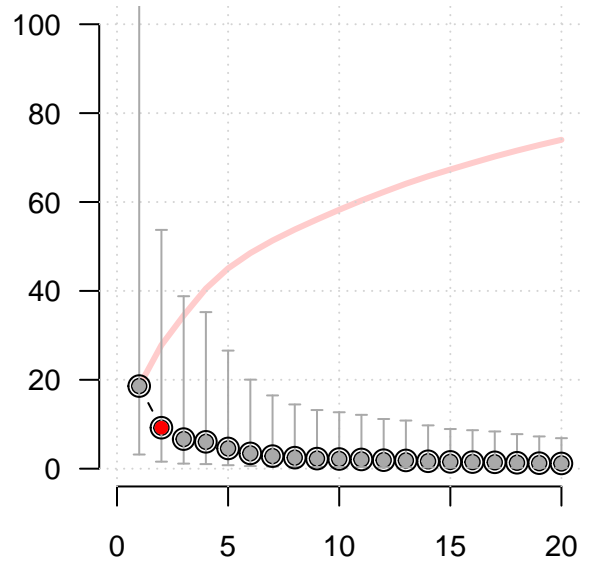
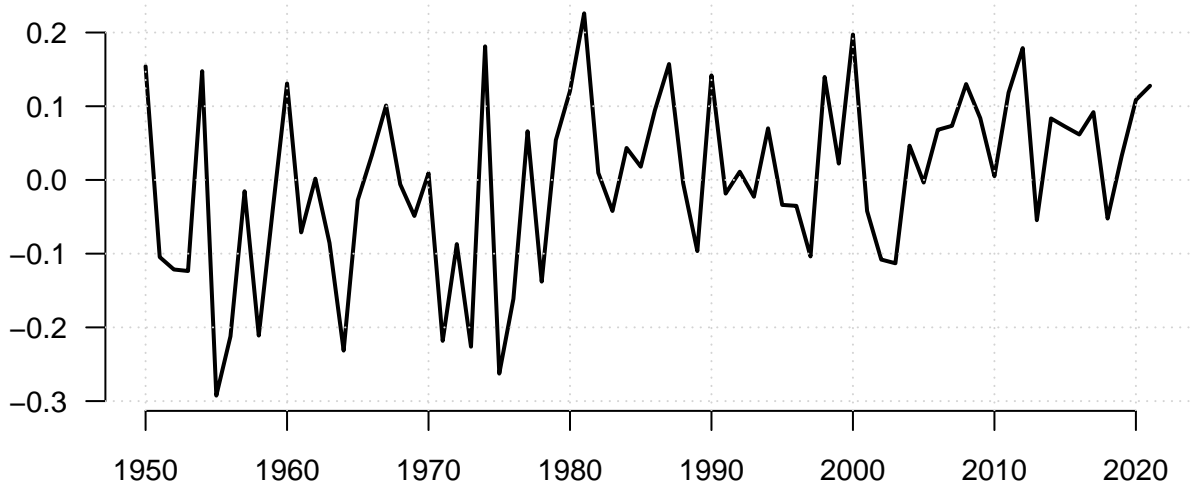
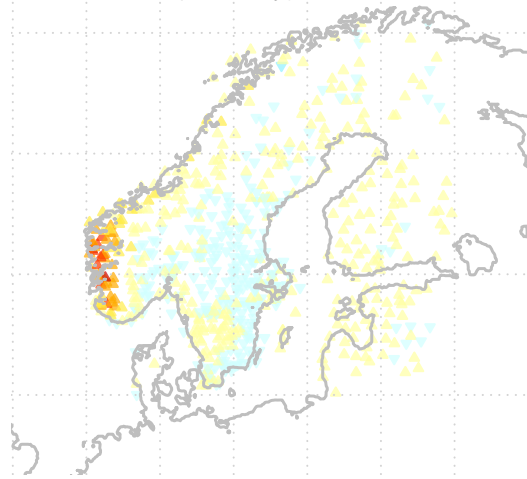
```r
it.predictand <- c(1950,2021)
is.predictand <- list(lon=c(5,30), lat=c(55,72))
if (!file.exists(tmp.st.fil)) {
  Y <- retrieve(file.predictand,lon=is.predictand$lon,
                lat=is.predictand$lat, nmin=50, it=it.predictand)
  save(Y,file=tmp.st.fil)
} else load(tmp.st.fil)
## Find the number of valid (nv) data points in each series
nv <- apply(Y,2,'nv')
## Select only those stations with more than 20000 valid data points
## (is = index space)
y <- subset(Y,is=nv > 20000)

## Estimate the autumn (SON)mean temperature
if (it=='annual') predictand <- annual(y,FUN=FUN) else
  predictand <- subset(as.4seasons(y,FUN=FUN),it=it)
attr(predictand,'variable') <- param
plot(y)
## The names of the stations the qualified
print(loc(y))
## Use the 1950-2022 period
predictand <- subset(predictand,it=c(1951,2022))
## Estimate PCA of the stations
## - using PCA rather than single stations tend to improve
## the signal-to-noise ratio
pca <- PCA(pcafill(predictand),n=5)
plot(pca)

## Get the  ERA5 reanalysis for calibrating the ESD model
if (it=='annual')  {
  era5 <- retrieve(reanalysis,param='tp',lon=lons,lat=lats, it=it.predictand)
  index(era5) <- year(era5)
  predictor <- subset(annual(era5,FUN=FUN))
} else {
  ## REB 2024-03-01: the monthly data looks suspect - poor results.
  era5 <- retrieve(reanalysis,param=param,lon=lons,lat=lats, it=it.predictand)
  predictor <- subset(as.4seasons(era5,FUN='mean'),it=it)
}
attr(predictor,'variable') <- param
## Ensure common unit for ERA5 and CMIP6 runs: kg/(m^2 s)
if (param=='mu') {
  ## Density of water ~ 1000 kg/m^3 so that 1 kg = 1 mm/m^2
  predictor <- predictor*1000  ## m/day -> daysum(kg/(m^2 s))
  attr(predictor,'unit') <- 'mm/day'
}
attr(predictor,'unit') <- switch(param,'fw'='fraction','mu'='mm/day')
map(predictor,new=FALSE)

## Test the link between predictor and predictand
eof <- EOF(predictor)
plot(eof,new=FALSE)
ds <-DS(pca,eof,ip=ip)
plot(ds,new=FALSE)
```

```r
  ## Downscale a CMIP6 ensemble
  ## Path with GCM data
  ## Downscale results
  dse.pca <- DSensemble(pca,predictor=predictor,path=path,
                        FUNX="mean",verbose=FALSE,
                        biascorrect=TRUE,rcp=ssp,ip=ip,
                        nmin=1,pattern='monmean_',
                        lon=lons,lat=lats,rel.cord=FALSE,plot=TRUE)
  ## Only keep the mean map information - smaller data volume
  predictor <- map(predictor)
  ## Grid the covariance  structures in the PCA
  gridmap(dse.pca$pca) -> dse.pca$eof

  nms <- names(dse.pca)
  for (nm in nms[-c(1:2)]) attr(dse.pca[[nm]],'model') <- NULL
  save(dse.pca,ds,predictand,predictor,eof,file=dse.results)
  print('--- Completed downscaling ---')
}
load(dse.results)
#print(paste('Loading downscaled results',dse.results))
load(tmp.st.fil)
#print(paste('Loading original station data',tmp.st.fil))
if (ssp=='ssp370') {
  map(predictor,new=FALSE)
  ## Plot the leading EOF
  plot(eof,type='fill',colbar=list(show=FALSE,pal='t2m'),cex=0.5,new=FALSE)
  par()
  ## Plot the second EOF
  plot(eof,type='fill',ip=2,colbar=list(show=FALSE,pal='t2m'),cex=0.5,new=FALSE)
  ## Plot the PCA
  nv <- apply(Y,2,'nv')
  ## Select only those stations with more than 20000 valid data points
  ## (is = index space). Use the 1950-2022 period
  y <- subset(annual(subset(Y,is=nv > 20000),FUN=FUN),it=c(1951,2022))
  ## Estimate PCA of the stations
  ## - using PCA rather than single stations tend to improve
  ## the signal-to-noise ratio
  pca <- PCA(pcafill(y))
  plot(pca,type='fill',colbar=list(show=FALSE,pal='t2m'),cex=0.5,new=FALSE)
  ## Make sure that the figures come out right in the knitted results
  ## Plot the PCA
  plot(pca,type='fill',ip=2,colbar=list(show=FALSE,pal='t2m'),cex=0.5,new=FALSE)
  ## Make sure that the figures come out right in the knitted results
  ## Plot the results of the downscaling exercise for the leading EOF
  plot(ds,type='fill',colbar1=list(show=FALSE,pal='t2m'),cex=0.5,new=FALSE)
  ## Make sure that the figures come out right in the knitted results
  par()
  ## Plot the results from the downscaling of the second EOF
  plot(ds,type='fill',ip=2,colbar1=list(show=FALSE,pal='t2m'),cex=0.5,new=FALSE)
  ## Extract annual statistics for Oslo
  ds.oslo <- subset(as.station(ds),loc="OSLO BLINDERN")
  plot(ds.oslo,map.show=FALSE,new=FALSE,
       main=paste("Downscaled and observed",param,"for OSLO BLINDERN"))
```

```
      grid()
      ## A hack to grab location names
      obs <- as.station(attr(ds,'original_data'))
      attr(obs,'location') <- loc(ds)
      y.oslo <- subset(obs,is=list(loc="OSLO BLINDERN"))
      lines(y.oslo,type='b',lty=2,lwd=2)
      lines(trend(y.oslo),lty=3)
      legend(1995,min(y.oslo,na.rm=TRUE),c('Observed','Downscaled'),
             col=c('black','red'),lty=c(2,1),bty='n')
      print(c(trend.coef(y.oslo),trend.pval(y.oslo)))
    }

    ## Remove runs which caused the evaluation to crash
    Bad.runs <- c(bad.runs,outlier.runs)
    for (j in length(Bad.runs)) {
      ## Find matching model in the object names
      rmrun <- grep(Bad.runs[j],names(dse.pca))
      #print(rmrun)
      print(paste("Removing",Bad.runs[j],names(dse.pca)[rmrun[1]]))
      model.id <- attr(dse.pca,'model_id')
      set2null <- c()
      for (jj in rmrun) set2null <- c(set2null,jj)

      ## Remove the elements with largest index first to avoid shifting of order.
      set2null <- sort(set2null,decreasing = TRUE)
      for (jj in set2null) dse.pca[[jj]] <- NULL
      ## The names of the object include 'info' and 'pca' in the beginning
      ## and since we weed out elements after order, we need to account for this
      attr(dse.pca,'model_id') <- model.id[-(set2null - 2)]
      #print(names(dse.pca))
      #print(sub('.r1i1p1f1','',attr(dse.pca,'model_id')))
    }

    #print(paste(param,ssp,sep='.'))
    #cat('Memory size: '); print(object.size(dse.pca),unit='Kb')
    nms <- names(dse.pca)
    for (nm in nms[-c(1:2)]) attr(dse.pca[[nm]],'model') <- NULL
    esd.results[[paste(param,ssp,sep='.')]] <- dse.pca
    #cat('Memory size: '); print(object.size(esd.results),unit='Kb')
    rm(list=c('dse.pca','predictor','ds','predictand','eof','Y'))
    gc(reset=TRUE)
  }
}
```

```
## [1] "Removing CESM i7_of.day_CESM"
## [1] "Removing CESM i7_of.day_CESM"
## [1] "Removing CESM i6_of.day_CESM"
## [1] "Removing CESM i6_of.day_CESM"
## [1] "Removing CESM i6_of.day_CESM"
## [1] "Removing CESM i6_of.day_CESM"
```

fw (fraction) — (1950 − 2021)

20 leading EOFs: 87.7 % of variance

Leading PC#1 of Total precipitation – Explained variance = 30%

20 leading EOFs: 87.7 % of variance

Leading PC#2 of Total precipitation – Explained variance = 19.06%

Downscaled and observed fw for OSLO BLINDERN

```
##            X trend.pvalue
##  0.005098029  0.015933708
## [1] "Removing CESM i6_of.day_CESM"
```

20 leading EOFs: 74 % of variance

Leading PC#1 of Total precipitation – Explained variance = 18.59%

20 leading EOFs: 74 % of variance

Leading PC#2 of Total precipitation – Explained variance = 9.22%

PCA Pattern #1 (unitless)

Cross−validation: r= 0.96

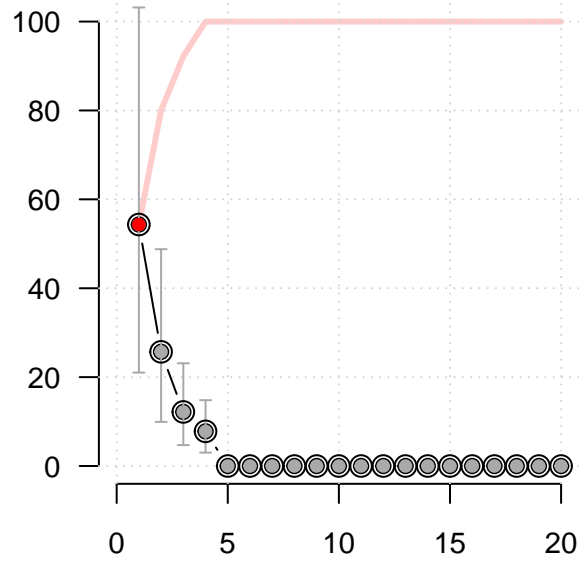
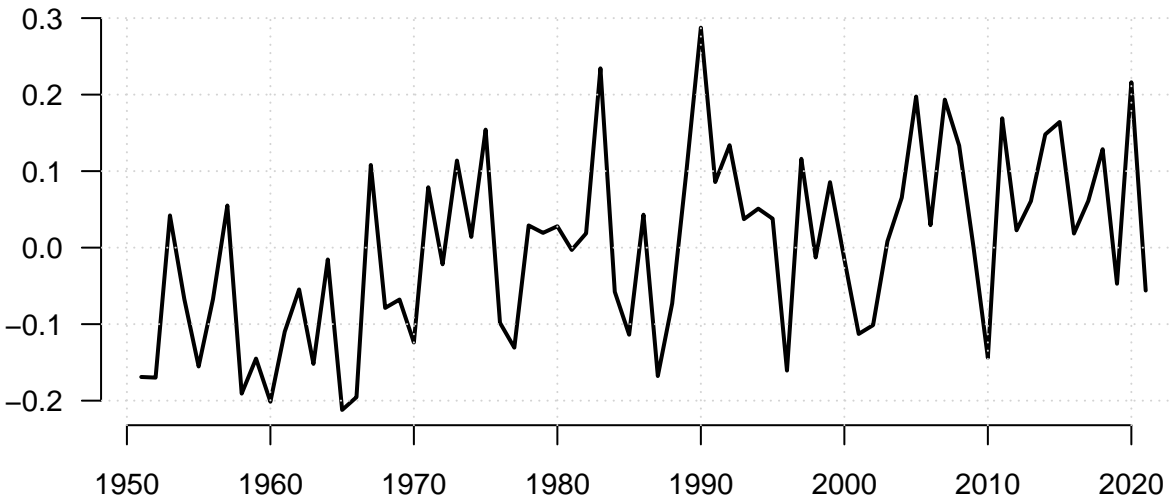
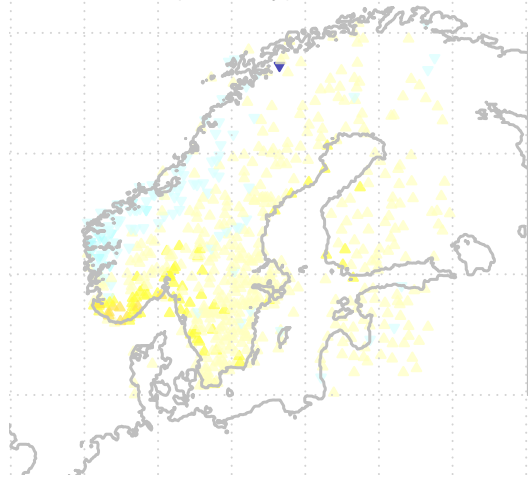
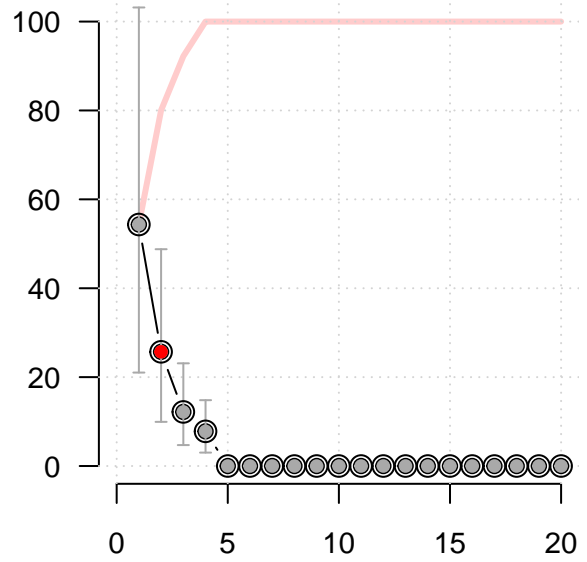
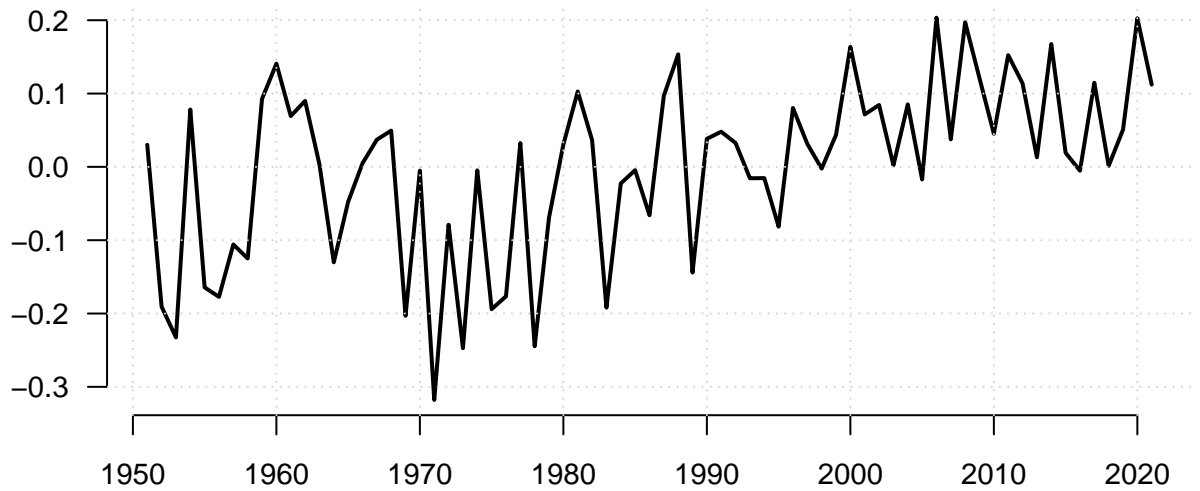
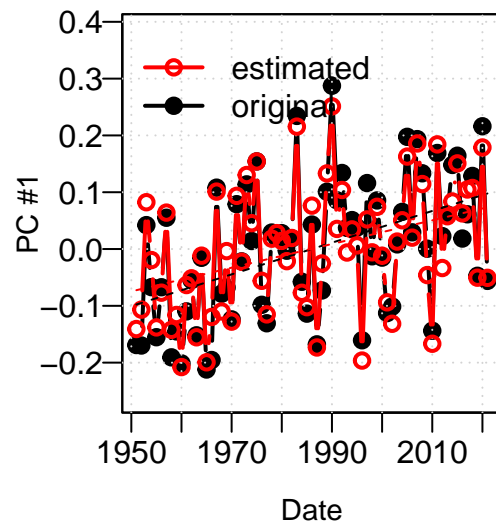
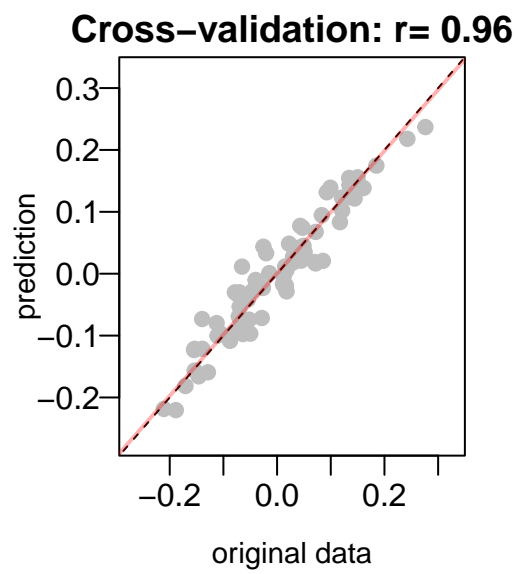
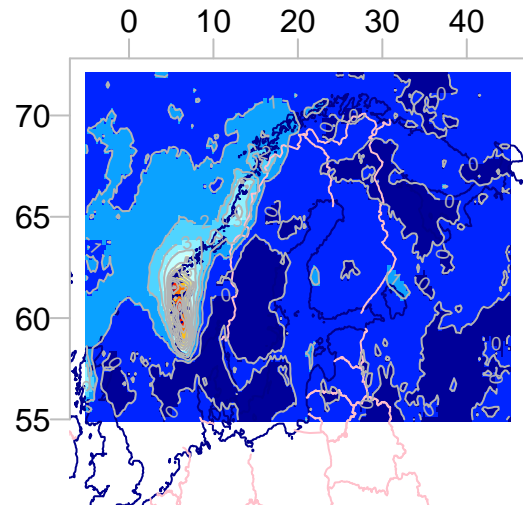
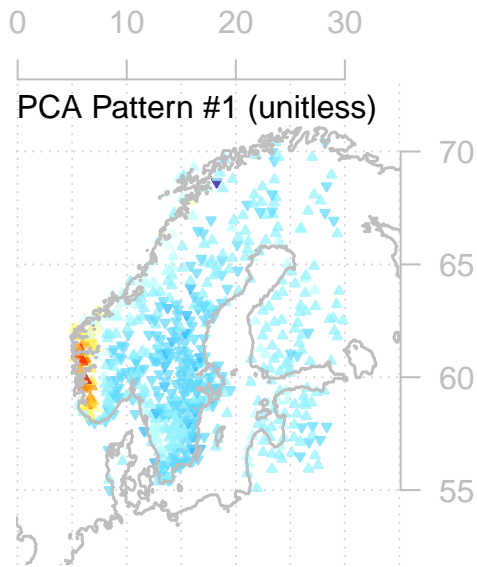
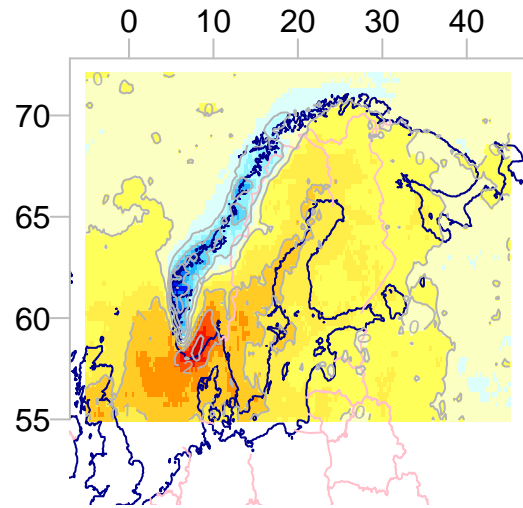
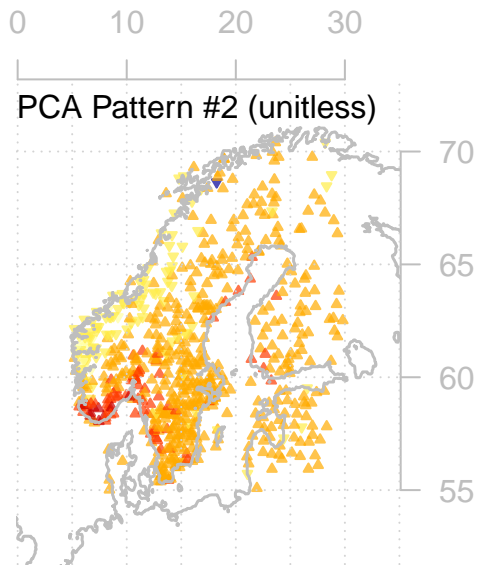
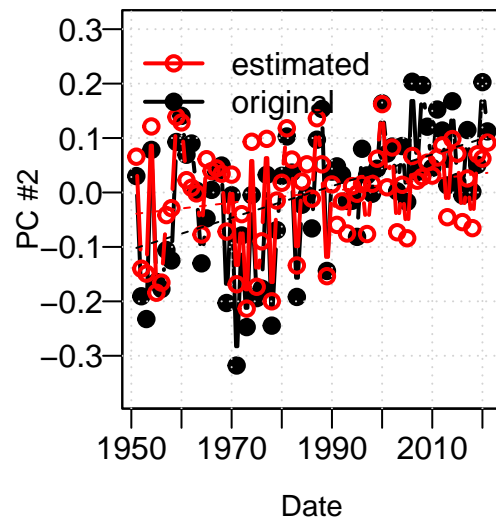
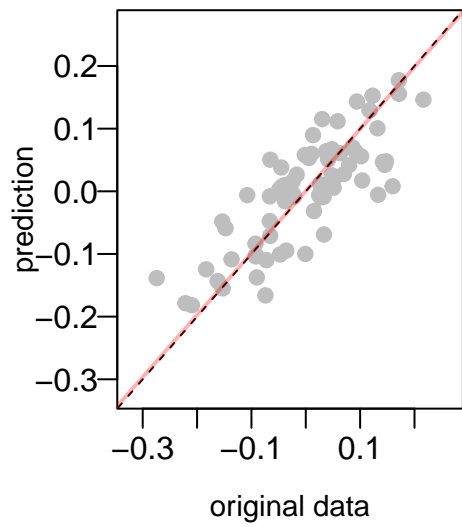
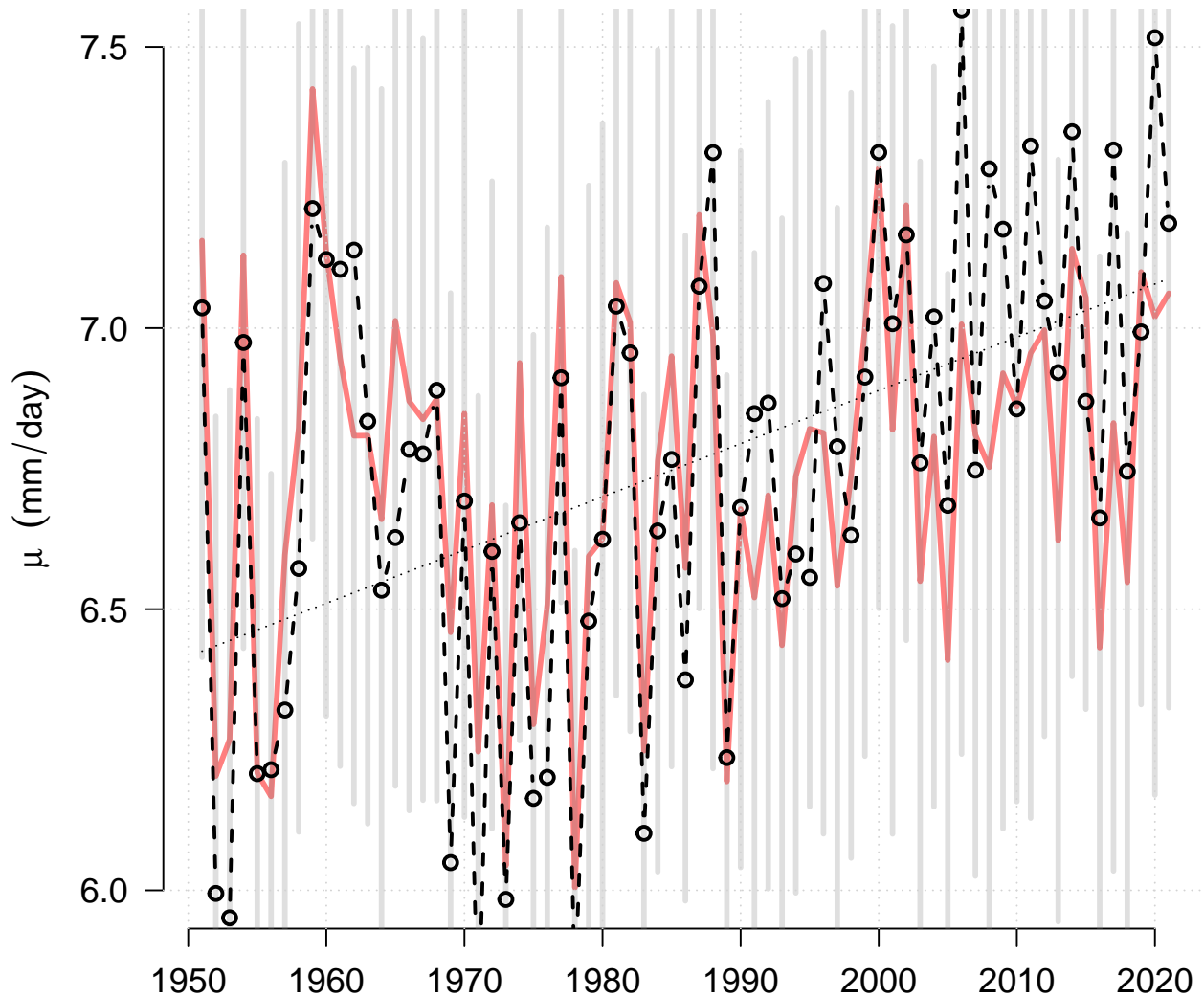
PCA Pattern #2 (unitless)

Cross−validation: r= 0.81

prediction

original data

PC #2

Date

Downscaled and observed mu for OSLO BLINDERN

```
##           X trend.pvalue
## 9.464053e-02 3.354855e-05
## [1] "Removing CESM i6_of.day_CESM"
## [1] "Removing CESM i6_of.day_CESM"
## [1] "Removing CESM i6_of.day_CESM"
```

To check the results, we plotted maps of the predictors from ERA5. The mean f_w indicated higher values near the western coasts of Norway, as expected. The results of an empirical orthogonal function (EOF) analysis of f_w from the ERA5 reanalysis also indicated pronounced variability over the ocean and in the north. The leading EOF accounted for 30% of the variance and the leading 20 EOFs represented 88% of the total variance. The second EOF accounted for 19% of the variance and represents an important part of the variability.

For the downscaling exercise based on ERA5 only, we took a random sample - in this case Oslo-Blindern. The downscaled results for f_w gave a good and close match between historical observations and downscaled f_w using f_w from ERA5 as predictor. The same exercise and evaluation for μ also indicated a match, but not as good as for f_w. Moreover, downscaled μ based on ERA5 μ underestimated the observed trend since 2000. In both cases, there was an upward trend in annual f_w and μ, both being significant at the 5%-level.

2.0.1 The predictand - aggregated rain gauge measurements

The chunk above shows a map with all the station coordinates with colour coded elevation. The predictand spanned the time interval 1950-2023, ad covered longitudes between 5-30E, latitudes 55-71N and elevations 0m - 2062m.

The predictand was represented in terms of principal component analysis (PCA) results of annual f_w estimated from the rain gauge data from ECA&D (Klein Tank et al. (2002)), as grouping the station data through PCA seems to increase the signal-to-noise ratio (Benestad and Mezghani (2015)). The spatial pattern of the matches between the predictand and predictor for PC 1 indicated similar geographical weights and the cross-validation gave a strong match with correlation of 0.93. The calibrated model managed to reproduce most of the observed interannual variability in PC#1 for local f_w in the Nordic countries. Corresponding results for the second EOF gave a cross-validation correlation of 0.92.

The map of mean μ also indicate higher values near the coast and the leading EOF accounts for 19% of the variance. The 20 leading EOFs account for 74% of the total variance, suggesting that the wet-day mean precipitation has stronger presence of small-scale variability. The second EOF explained 9%, and the higher order modes (not shown) also contributed to the variability.

A downscaling exercise based only on annual μ from ERA5 and rain gauges also gave a good match between the predictor and predictand: similar geographical weights and a cross-validation correlation of 0.96. The second EOF for μ explained 8% of the variance. For the second mode, the cross-validation correlation was 0.81, connected to geographical weights along the coast of Norway.

There was a good match between precipitation statistics aggregated from rain gauge measurements and the ERA5 reanalysis.

In other words, the results of the downscaling indicate that both f_w and μ to a large extent can be estimated based on their respective large-scale equivalents from the ERA5 reanalysis. There is, however, a question whether the GCMs are able to skillfully reproduce the same large-scale features in f_w and μ, and an evaluation of their skill is provided at the end of this R-markdown script.

2.1 Diagnostics of the results

The following chunk provides information of 652 locations with rain gauge data downscaled and a model ensemble originally with 30 members used here to estimate future local climate projections. Some of the members were removed due to data issues resulting in technical crash when running the code. Also, one suspect member identified from the GCM evaluation was removed from the projection of future rainfall statistics.

```r
load(tmp.st.fil)
## A sanity check - how much memory is taken by the results.
print(object.size(esd.results))
```

```
## 241623840 bytes
```

```r
## Show the scenarios/projections.
print(names(esd.results))
```

```
##  [1] "fw.hist"   "mu.hist"   "fw.ssp126" "mu.ssp126" "fw.ssp245" "mu.ssp245"
##  [7] "fw.ssp370" "mu.ssp370" "fw.ssp585" "mu.ssp585"
```

```
## Summary of countries:
print(table(cntr(Y)))
```

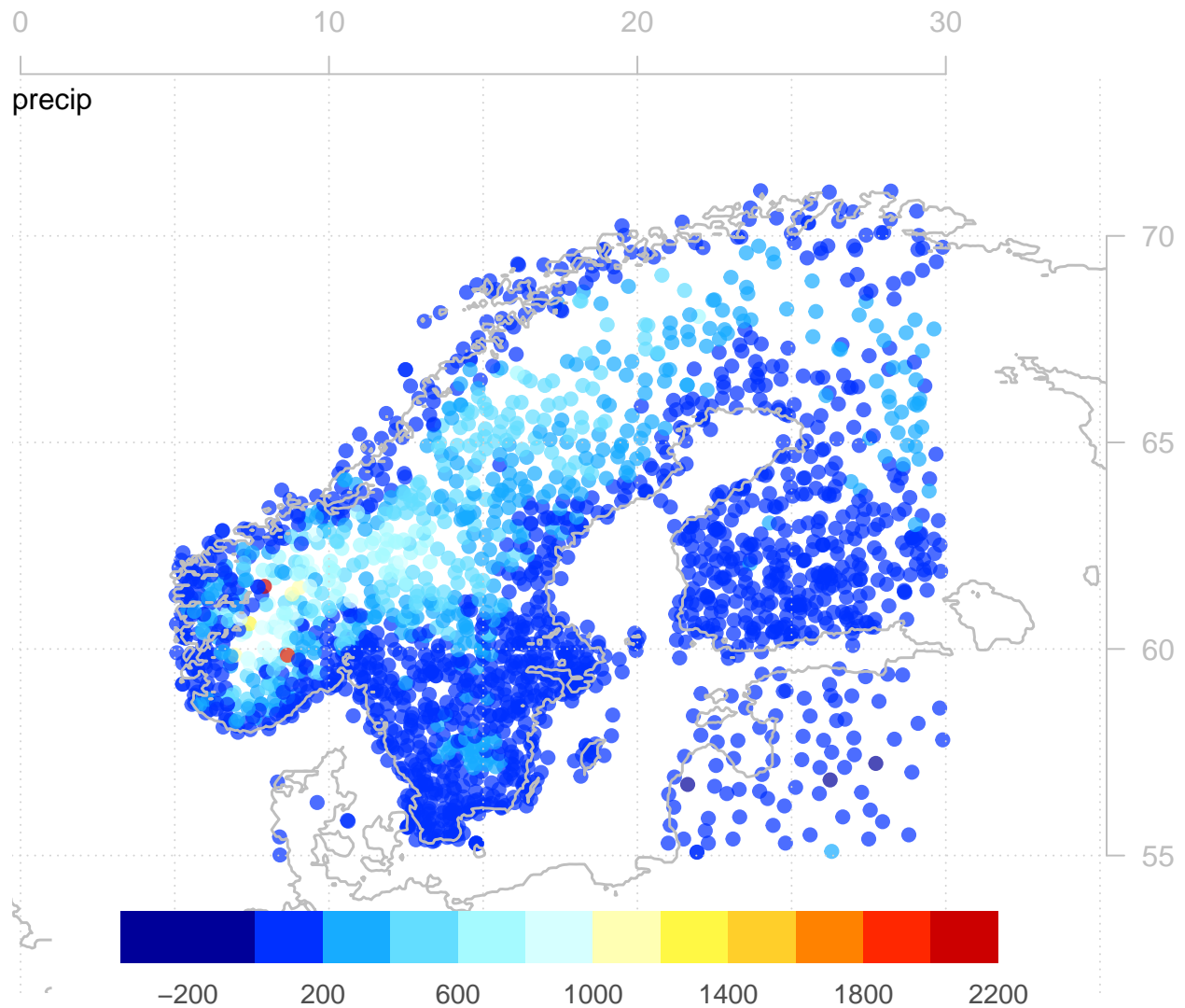
```
##
##              Belarus                 Denmark                        Estonia
##                    4                      14                             27
##              Finland                 Germany                         Latvia
##                  443                       1                             29
##            Lithuania                  Norway Russian Federation (the)
##                   13                     669                             11
##               Sweden
##                  920
```

```
## Time interval for the predictand
print(range(index(Y)))
```

```
## [1] "1950-01-01" "2021-12-31"
```

```
## Spatial distribution of the predictand: lon lat alt
print(c('Lons:',round(range(lon(Y))),'Lats:',round(range(lat(Y))),'Alts:',range(alt(Y))))
```

```
## [1] "Lons:" "5"      "30"      "Lats:" "55"      "71"      "Alts:" "-100"   "2062"
```

```
altitude <- function(x,na.rm=TRUE) alt(x)
## Show the locations of all stations considered - also short series
map(Y,FUN='altitude',cex=1,colbar=list(pal='terrain',show=TRUE),new=FALSE)
```

```
## Location names:
print(paste(length(loc(esd.results$fw.ssp370)),'locations and ',
            length(names(esd.results$fw.ssp370))-3,'model runs following SSP370'))
```

```
## [1] "652 locations and  29 model runs following SSP370"
```

```
print(sort(loc(esd.results$fw.ssp370)))
```

```
##    [1] "AANEKOSKI KALANIEMI"      "ABISKO"
##    [3] "ABJORSBRATEN"             "ABY"
##    [5] "ADA"                      "AHTARI MYLLYMAKI"
##    [7] "AINAZI"                   "AKROKEN"
##    [9] "AL III"                   "ALAVUS SULKAVANKYLA"
##   [11] "ALBERGA"                  "ALFOTEN II"
##   [13] "ALGARAS"                  "ALINGSAS_D"
##   [15] "ALSVAG I VERSTERALEN II"  "ALTA LUFTHAVN"
##   [17] "ALUKSNE"                  "ALVSBYN_A"
##   [19] "ANDOYA"                   "ANGARDSVATNET"
```

```
##  [21] "ANGE-PARTEBODA D"          "ANKENES"
##  [23] "ARJEPLOG A"                "ASELE"
##  [25] "ASERAL"                    "ASKER"
##  [27] "ATNSJOEN"                  "ATORP"
##  [29] "AUNET"                     "AURLAND"
##  [31] "AURSUND"                   "AVASJO-BORGAFJALL_D"
##  [33] "BACKEFORS"                 "BAKKE"
##  [35] "BAKSJONASET"               "BALLASTVIKEN"
##  [37] "BANGDALEN"                 "BARDUFOSS"
##  [39] "BARKESTAD"                 "BAUSKA"
##  [41] "BAVERTRASK"                "BEITO"
##  [43] "BERGELIGREND"              "BERGVIK"
##  [45] "BESSAKER"                  "BIRI"
##  [47] "BISPGARDEN"                "BISPGARDEN 1"
##  [49] "BJORHEIM I RYFYLKE"        "BJORKASEN"
##  [51] "BJORNHOLT"                 "BJORNSTORP"
##  [53] "BJURHOLM"                  "BLAIKEN"
##  [55] "BLANKTJERNMOEN I KVIKNE"   "BODOE VI"
##  [57] "BOLLEBYGD"                 "BOLLERUP"
##  [59] "BOLMEN"                    "BONES I BARDU"
##  [61] "BORAS"                     "BOTNEN I FORDE"
##  [63] "BOVERDAL"                  "BREKKE I SOGN"
##  [65] "BRIKSDAL"                  "BROMOLLA"
##  [67] "BULKEN"                    "BURAN"
##  [69] "CHARLOTTENBERG"            "DAGDA"
##  [71] "DALE I SUNNFJORD II"       "DAUGAVPILS"
##  [73] "DEJEFORS"                  "DIVIDALEN"
##  [75] "DJURSATRA"                 "DOBELE"
##  [77] "DODRE"                     "DRALINGE"
##  [79] "DRANGEDAL"                 "DREVSJO"
##  [81] "EDSELE"                    "EGERSUND"
##  [83] "EIDE PA NORDMORE"          "EIDSVOLL VERK"
##  [85] "EIKEMO"                    "EKSINGEDAL"
##  [87] "ENKOPING"                  "ENONKOSKI SIMANALA"
##  [89] "ERESFJORD"                 "ESKILSTUNA"
##  [91] "ESPEDALEN"                 "EVERTSBERG"
##  [93] "FAERDER FYR"               "FAGELSJO_D"
##  [95] "FAGERED"                   "FAGERHEDEN"
##  [97] "FAGERHULT_D"               "FAGERSTA"
##  [99] "FALKOPING"                 "FALSTERBO"
## [101] "FALUN"                     "FANA - STEND"
## [103] "FINNBACKA_D"               "FINSPANG"
## [105] "FLAHULT"                   "FLEKKEFJORD"
## [107] "FLEN"                      "FOLDSAE"
## [109] "FOLLDAL"                   "FONE_D"
## [111] "FORSE"                     "FRANDESTA"
## [113] "FROYSET"                   "GADDEDE"
## [115] "GARDERMOEN"                "GEILO"
## [117] "GEIRANGER"                 "GENDALEN"
## [119] "GILLASTIG"                 "GLOMMERSTRASK"
## [121] "GODEGARD"                  "GOL - STAKE"
## [123] "GOTSKA SANDON"             "GRANGESBERG"
## [125] "GREBBESHULT"               "GRIMELI I KRODSHERAD"
## [127] "GRINDSJON"                 "GRONBAEK-ALLINGSKOVGARD"
```

```
## [129] "GRONLIDEN"                   "GRUNDFORSEN"
## [131] "GRYCKSBO_D"                   "GRYTHYTTAN"
## [133] "GULBENE"                      "GULLBRA"
## [135] "GULLSPANG"                    "GUNNARN"
## [137] "GUSTAVSFORS"                  "HAAPAVESI MUSTIKKAMAKI"
## [139] "HAFSLO"                       "HAGGARDA"
## [141] "HAILUOTO OJAKYLA"             "HALDEN"
## [143] "HALLSTABERG"                  "HALMSTAD"
## [145] "HALSAFJORD II"                "HAMEENLINNA LAMMI PAPPIL"
## [147] "HAMEENLINNA PIRTTIKOSKI"      "HANKO TVARMINNE"
## [149] "HANO_A"                       "HAPARANDA"
## [151] "HARNOSAND"                    "HARSFJARDEN"
## [153] "HASSLEBY"                     "HASSLEHOLM"
## [155] "HATLESTRAND"                  "HAUKEDAL"
## [157] "HAUSJARVI LAVINTO"            "HAVELUND"
## [159] "HAVRARYD_D"                   "HEDBERG"
## [161] "HEDE"                         "HEDEMORA"
## [163] "HEDEN"                        "HEDRUM"
## [165] "HEINAVESI PALOKKI"            "HEINOLA ASEMANTAUS"
## [167] "HELLAND I GJESDAL"            "HELSINKI KAISANIEMI"
## [169] "HELTERMAA"                    "HEMAVAN"
## [171] "HEMSE"                        "HEREFOSS"
## [173] "HIASEN I SIGDAL"              "HITRA"
## [175] "HJALTA_KRAFTVERK"             "HJO"
## [177] "HOBURG"                       "HOBY"
## [179] "HOEYDALSMO"                   "HOFORS"
## [181] "HOGBO"                        "HOGEMALEN"
## [183] "HOGLEKARDALEN"                "HOGNESTAD"
## [185] "HOIDALEN I SOLUM"             "HOLE"
## [187] "HOLJES"                       "HOLMBO_D"
## [189] "HORDA"                        "HORKEN"
## [191] "HORNINDAL"                    "HOVLANDSDAL"
## [193] "HUDIKSVALL"                   "HUITTINEN SALLILA"
## [195] "HULT"                         "HUNDSEID I VIKEDAL"
## [197] "HUSKVARNA"                    "HUSNES"
## [199] "HUSTADVATN"                   "HVALER"
## [201] "HYLTAN"                       "HYNDEVAD"
## [203] "HYVINKAA HYVINKAANKYLA"       "IDKERBERGET"
## [205] "IDRE_D"                       "IGSI I HOBOL"
## [207] "INARI RAJA-JOOSEPPI AKUJ"     "INDAL_D"
## [209] "INNERDALEN"                   "INNVIK"
## [211] "JACKVIK"                      "JARPLIDEN"
## [213] "JARPSTROMMEN"                 "JARVSO"
## [215] "JELGAVA"                      "JOENSUU PYHASELKA"
## [217] "JOGEVA"                       "JOHVI"
## [219] "JOKIOINEN JOKIOISTEN OBS"     "JOKKMOKK"
## [221] "JORMLIEN"                     "JOTKAJAVRE"
## [223] "JOUTSA LEIVONMAKI SAVENA"     "JUNSELE"
## [225] "JUUKA NIEMELA"                "JUUPAJOKI HYYTIALA"
## [227] "JYVASKYLA LENTOASEMA"         "JYVASKYLA MUURATJARVI"
## [229] "KAARINA YLTOINEN"             "KALAJOKI METSAKYLA"
## [231] "KALMAR_D"                     "KANGASNIEMI KIRKONKYLA"
## [233] "KANKAANPAA NIINISALO PUO"     "KARBOLE"
## [235] "KARESUANDO"                   "KARLSBORG_MO"
```

```
## [237] "KARLSHAMN"                  "KARUNGI"
## [239] "KARVIA ALKKIA"               "KASA"
## [241] "KATKESUANDO"                 "KATRINEHOLM_D"
## [243] "KAUHAVA LENTOKENTTA"         "KAUNISVAARA"
## [245] "KAUSTINEN TASTULA"           "KAVSJO_D"
## [247] "KAXAS-AFLO_D"                "KIHNU"
## [249] "KILAGARDEN"                  "KILEGREND"
## [251] "KINDSJON_D"                  "KINSARVIK"
## [253] "KISA_D"                      "KISE PA HEDMARK"
## [255] "KIURUVESI KORPIJOKI"         "KJELSAS I SORKEDALEN"
## [257] "KJEVIK"                      "KJOBLI I SNAASA"
## [259] "KLAIPEDA"                    "KLIPPAN"
## [261] "KLIPPEN"                     "KNARED"
## [263] "KOKEMAKI PEIPOHJA HYRKOL"    "KOLKA"
## [265] "KOLSVA"                      "KOMSTORP"
## [267] "KORSA"                       "KORSBERGA"
## [269] "KOUVOLA ANJALA"              "KRAKMO"
## [271] "KRAMFORS"                    "KRANGEDE_A"
## [273] "KRISTIANSTAD"                "KRISTINEHAMN"
## [275] "KROKSHULT_D"                 "KROKSJO"
## [277] "KUNDA"                       "KURESSAARE"
## [279] "KUUSAMO KIUTAKONGAS"         "KUUSIKU"
## [281] "KVIKKJOKK-ARRENJARKA"        "KVIKNE I OSTERDAL"
## [283] "KYRKTASJO_D"                 "LAANE-NIGULA"
## [285] "LAFORSEN"                    "LAHTI LAUNE"
## [287] "LAINIO"                      "LANDSKRONA_D"
## [289] "LANDSORT"                    "LANDVIK"
## [291] "LANNA"                       "LANNAS"
## [293] "LAVIK"                       "LAXBACKEN"
## [295] "LEINSTRAND"                  "LEIPIKVATTNET"
## [297] "LEIRFJORD"                   "LEKNES I LOFOTEN"
## [299] "LEKSAND"                     "LESSEBO"
## [301] "LETAFORS"                    "LIAFOSS"
## [303] "LIDKOPING"                   "LIEN I SELBU"
## [305] "LIEPAJA"                     "LIFJELL"
## [307] "LILLHAMRA"                   "LINDESBERG"
## [309] "LINDESNES FYR"               "LINHULT_D"
## [311] "LINKOEPING-MALMSLAETT"       "LISJO"
## [313] "LISSKOGSASEN"                "LIST/SYLT"
## [315] "LISTA FYR"                   "LJUNGBY"
## [317] "LJUNGDALEN"                  "LJUSNEDAL"
## [319] "LOBONAS_D"                   "LOHJA PORLA"
## [321] "LOKSMYR"                     "LOTEN_D"
## [323] "LOUISEFRED"                  "LOVANGER"
## [325] "LOVBERGA_D"                  "LOVSTA"
## [327] "LUDVIKA"                     "LULEA FLYGPLATS"
## [329] "LULEA-BERGNASET"             "LUND"
## [331] "LUNNER"                      "LUROY"
## [333] "LYCKSELE_A"                  "LYSEBOTN"
## [335] "MAANINKA HALOLA"             "MALA"
## [337] "MALILLA"                     "MALMAGEN"
## [339] "MALMBERGET"                  "MALMO"
## [341] "MALUNG"                      "MANDAL II"
## [343] "MANGSBODARNA"                "MARIDALSOSET"
```

```
## [345] "MARIESTAD"               "MARISTOVA"
## [347] "MAUDAL"                  "MERSRAGS"
## [349] "MESTAD I ODDERNES"       "MO I RANA III"
## [351] "MOGEN"                   "MORA_A"
## [353] "MORJARV"                 "MORKO"
## [355] "MOTALA"                  "MULTRA"
## [357] "MUNKFORS"                "MUONIO ALAMUONIO"
## [359] "MURJEK"                  "MYKLAND"
## [361] "MYKLEBUST I BREIM"       "MYSKELASEN_D"
## [363] "NAGGEN"                  "NAMDALSEID"
## [365] "NARVA"                   "NAVELSJO"
## [367] "NELAUG"                  "NES PA HEDMARK"
## [369] "NOKIA TOTTIJARVI"        "NORD ODAL"
## [371] "NORDBY (FANO)"           "NORDDAL"
## [373] "NORDSTRAND"              "NORDSTRAUM I KVAENANGEN"
## [375] "NORRBY"                  "NORRBY_S"
## [377] "NORRSUNDET"              "NORRVEDA"
## [379] "NORSBORG_11"             "NOTODDEN"
## [381] "NURMES MUJEJARVI"        "NYBERGET"
## [383] "NYBRO-SLATTINGEBYGD D"   "NYHAMMAR"
## [385] "NYNASHAMN"               "OCKELBO"
## [387] "ODESHOG"                 "OERBEKKEDALEN"
## [389] "OGESTAD"                 "OJE_D"
## [391] "OKSENDAL"                "OKSNINGOY"
## [393] "OKSOEY FYR"              "OLANDS NORRA UDDE"
## [395] "OLASTORP"                "OLOFSTROM"
## [397] "ON"                      "OREBRO"
## [399] "ORJE"                    "ORLAND"
## [401] "ORNSKOLDSVIK"            "ORSKAR_A"
## [403] "ORSKOG"                  "OS I OESTERDAL"
## [405] "OSBY"                    "OSKARSHAMN"
## [407] "OSLO BLINDERN"           "OSTANA"
## [409] "OSTANVIK"                "OSTAS I HEGRA"
## [411] "OSTRE TOTEN - APELSVOLL" "OUTOKUMPU VIURUNIEMI"
## [413] "OVERANG"                 "OVERTORNEA"
## [415] "OVERUM"                  "OVRE SIRDAL"
## [417] "OVRE_SOPPERO"            "OVRE_SVARTLA"
## [419] "OVSTEDAL"                "OXELOSUND"
## [421] "PAJALA"                  "PAKRI"
## [423] "PARAINEN UTO"            "PARKAJOKI"
## [425] "PARNU"                   "PAVILOSTA"
## [427] "PITEA"                   "PORJUS"
## [429] "PORSA II"                "POSTMYR I DRANGEDAL"
## [431] "PRASTKULLA"              "PRESTSTULEN"
## [433] "PRIEKULI"                "PSKOV"
## [435] "PUNKAHARJU LAUKANSAARI"  "PUOTTAURE"
## [437] "RAMSJOHOLM"              "RATAN"
## [439] "RAULAND"                 "REGNA_D"
## [441] "REINLI"                  "REMNINGSTORP"
## [443] "RENA - HAUGEDALEN"       "REZEKNE"
## [445] "RIDDARHYTTAN"            "RIGA"
## [447] "RINDAL"                  "RISINGE"
## [449] "RISNES I FJOTLAND"       "RISTNA"
## [451] "RJUKAN"                  "ROBACKSDALEN"
```

```
## [453] "ROLDAL"                      "RONNEBY-BREDAKRA"
## [455] "RORSBO"                      "RORVIK_D"
## [457] "RORVIKVATN VED VADHEIM"      "ROSENDAL"
## [459] "ROSSARED"                    "ROSTA"
## [461] "ROVANIEMI APUKKA"            "ROVANIEMI LENTOASEMA"
## [463] "RUCAVA"                      "RUHNU"
## [465] "RUJIENA"                     "RUSTEFJELBMA"
## [467] "RYGGE"                       "SADILIDEN"
## [469] "SAETERMOEN II"               "SAFFLE"
## [471] "SAITTAROVA_D"                "SALA"
## [473] "SALDUS"                      "SALLA KELLOSELKA"
## [475] "SALO KARKKA"                 "SAND I RYFYLKE II"
## [477] "SANDANE"                     "SANDEFJORD"
## [479] "SANDHAMN"                    "SANDNAS"
## [481] "SANDVIKEN"                   "SARNA"
## [483] "SATENAS"                     "SATRA_GARD"
## [485] "SATTRA"                      "SAUDA"
## [487] "SAVSJO"                      "SAVUKOSKI KK"
## [489] "SEGERSTAD"                   "SEINAJOKI PELMAA"
## [491] "SIAULIAI"                    "SIDSJO"
## [493] "SIHCCAJAVRI"                 "SIIKAJOKI REVONLAHTI"
## [495] "SIIKALATVA KESTILA"          "SIMONSTORP_O"
## [497] "SJOGARDE"                    "SKARA"
## [499] "SKARA 1"                     "SKATTUNGBYN"
## [501] "SKEDEMOSSE"                  "SKILLINGARYD"
## [503] "SKJAK"                       "SKJAK II"
## [505] "SKJENALDFOSSEN I ORKDAL"     "SKJORBY"
## [507] "SKOVDE"                      "SKREADALEN II"
## [509] "SKRIVERI"                    "SKULTE"
## [511] "SKULTUNA"                    "SKURUP_D"
## [513] "SODANKYLA LOKKA"             "SODANKYLA VUOTSO"
## [515] "SODANKYLA. LAPIN ILMATIE"    "SODERKOPING"
## [517] "SODERTALJE"                  "SOGNDAL - SELSENG"
## [519] "SOKNA II"                    "SOLA"
## [521] "SOLLEFTEA"                   "SONGLI"
## [523] "SORABY"                      "SORBYTORP"
## [525] "SORLI"                       "SORVE"
## [527] "SOSJO"                       "SOVETSK"
## [529] "SOYLAND I GJESDAL"           "STADLANDET"
## [531] "STALON_11"                   "STEHAG"
## [533] "STEIGEN"                     "STENDE"
## [535] "STENSELE"                    "STOCKHOLM"
## [537] "STOMNE"                      "STORBRON"
## [539] "STORFINNFORSEN"              "STROMSFOSS SLUSE"
## [541] "SULDALSVATN"                 "SULITJELMA"
## [543] "SUNNDALSORA III"             "SUOLOVUOPMI"
## [545] "SUSENDAL"                    "SVANSTEIN"
## [547] "SVAPPAVAARA"                 "SVEG"
## [549] "SVENSKA HOGARNA"             "SVILAND"
## [551] "SVINHULT_D"                  "SYSMA JOUTSJARVI"
## [553] "TAFJORD"                     "TAKLE"
## [555] "TALLIDEN"                    "TALLINN"
## [557] "TANDSBYN"                    "TANGER"
## [559] "TARENDO"                     "TARTU-TORAVERE"
```

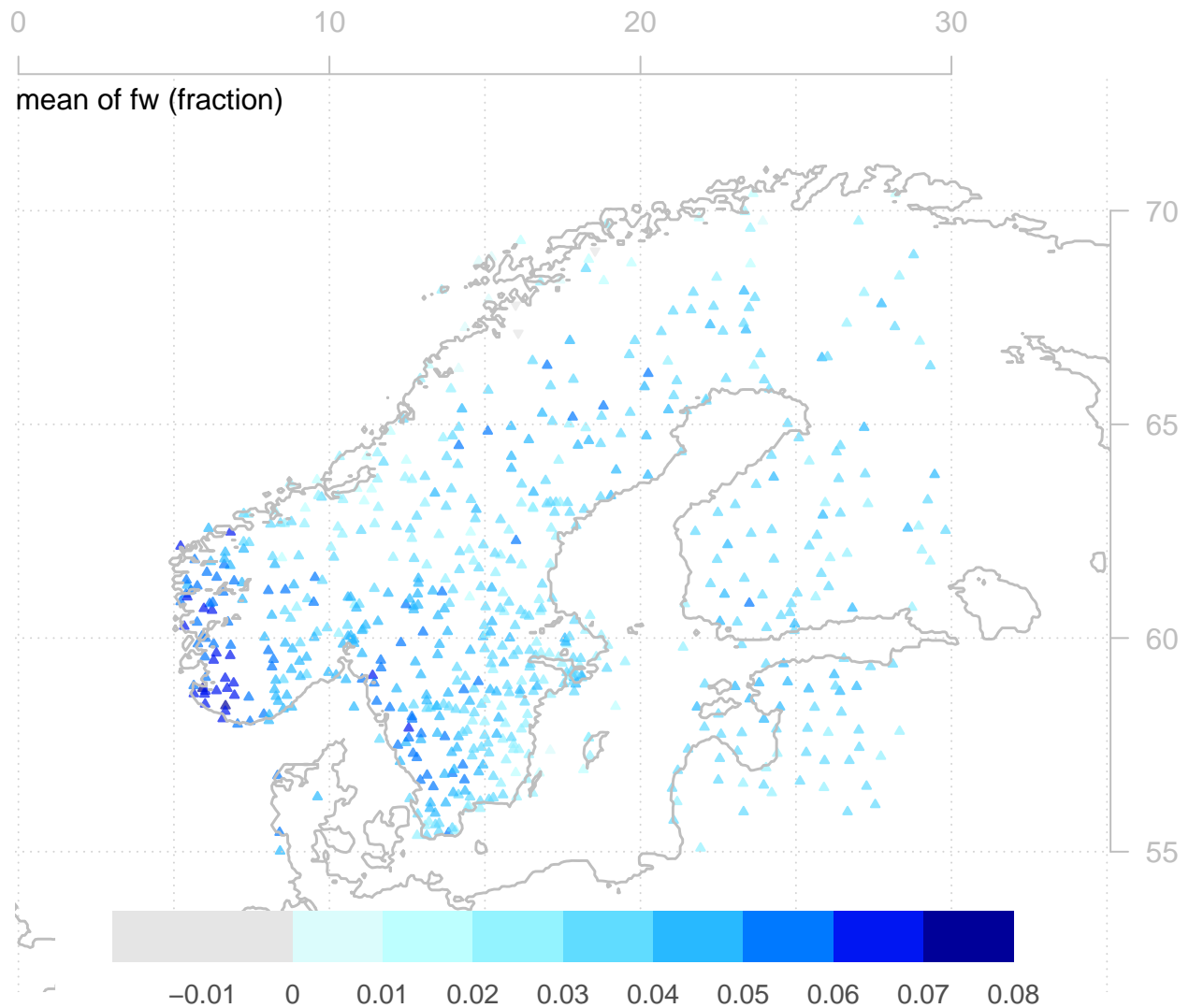
```
## [561] "TASAN"                     "TEUVA KAUPPILANKYLA"
## [563] "TIIRIKOJA"                 "TJALLMO"
## [565] "TJARNAS"                   "TOHOLAMPI ORAVALA"
## [567] "TOMELILLA"                 "TOOMA"
## [569] "TORNTORP"                  "TORSBY"
## [571] "TORUNGEN FYR"              "TOSTEBERGA"
## [573] "TOVDAL"                    "TRANAS"
## [575] "TRANEBERG"                 "TRANGSLET"
## [577] "TRANSTRAND_D"              "TRELLEBORG"
## [579] "TROMSO"                    "TROMSO - LANGNES"
## [581] "TUDDAL"                    "TULLINGE A"
## [583] "TUNHOVD"                   "TUNNSJO"
## [585] "TURI"                      "TUSTERVATNET II"
## [587] "TVEITSUND"                 "TVINGELSHED"
## [589] "TYSSE"                     "UDDEVALLA"
## [591] "UKKESTAD"                  "ULLANGER"
## [593] "ULRICEHAMN"                "ULRIKA"
## [595] "ULVSJO"                    "UNGSBERG"
## [597] "UPLO"                      "UPPSALA"
## [599] "UPPSALA_FLYGPLATS"         "UTAJARVI SARKIJARVI"
## [601] "UTGARD"                    "UTSJOKI KEVO-1"
## [603] "UUSIKAUPUNKI NERVANDERIN"  "VAALA PELSO"
## [605] "VADSTENA"                  "VAERNES"
## [607] "VAEXJOE"                   "VAIKE-MAARJA"
## [609] "VALGA"                     "VALLBO"
## [611] "VALLENTUNA"                "VALSJON"
## [613] "VANERSBORG"                "VANG I VALDRES"
## [615] "VANTAA HELSINKI-VANTAAN "  "VARBERG"
## [617] "VARNE"                     "VASTERAS-HASSLO"
## [619] "VASTERHANINGE"             "VASTVALLA"
## [621] "VATTHOLMA"                 "VEA"
## [623] "VENTSPILS"                 "VESANTO SONKARI"
## [625] "VESTERVIG"                 "VESTRE GAUSDAL"
## [627] "VIDSEL"                    "VIEREMA KAARAKKALA"
## [629] "VIGRA"                     "VIHTI HIISKULA"
## [631] "VIITASAARI KOLIMA KARNA"   "VIK I SOGN III"
## [633] "VILAKA"                    "VILJANDI"
## [635] "VILSANDI"                  "VIMMERBY"
## [637] "VINGA"                     "VIRTSU"
## [639] "VISBY"                     "VISBY_FLYGPLATS"
## [641] "VITTANGI"                  "VOMB"
## [643] "VORU"                      "VRENA"
## [645] "VYBORG"                    "WIAD"
## [647] "YANISKOSKI"                "YSTAD"
## [649] "YTTERBERG_D"               "YTTERHOGDAL"
## [651] "ZILANI"                    "ZOSENI"
```

```r
## Ensemble members from the elements of the data object:
#print(names(esd.results$fw.ssp370))
print(attr(esd.results$fw.hist,'model_id'))
```

```
##  [1] "ACCESS.CM2.r1i1p1f1"       "ACCESS.ESM1.5.r1i1p1f1"
##  [3] "AWI.ESM.1.1.LR.r1i1p1f1"   "BCC.CSM2.MR.r1i1p1f1"
##  [5] "BCC.ESM1.r1i1p1f1"         "CanESM5.r1i1p1f1"
```

```
##  [7] "CMCC.CM2.HR4.r1i1p1f1"        "CMCC.CM2.SR5.r1i1p1f1"
##  [9] "CMCC.ESM2.r1i1p1f1"           "CNRM.CM6.1.r1i1p1"
## [11] "CNRM.CM6.1.HR.r1i1p1"         "CNRM.ESM2.1.r1i1p1"
## [13] "EC.Earth3.r1i1p1f1"           "EC.Earth3.AerChem.r1i1p1f1"
## [15] "EC.Earth3.CC.r1i1p1f1"        "EC.Earth3.Veg.r1i1p1f1"
## [17] "EC.Earth3.Veg.LR.r1i1p1f1"    "FGOALS.f3.L.r1i1p1f1"
## [19] "FGOALS.g3.r1i1p1f1"           "GFDL.CM4.r1i1p1f1"
## [21] "GFDL.CM4.r1i1p1f1"            "GFDL.ESM4.r1i1p1f1"
## [23] "HadGEM3.GC31.LL.r1i1p1f3"     "HadGEM3.GC31.MM.r1i1p1f3"
## [25] "IITM.ESM.r1i1p1f1"            "INM.CM4.8.r1i1p1f1"
## [27] "INM.CM5.0.r1i1p1f1"           "IPSL.CM5A2.INCA.r1i1p1"
## [29] "IPSL.CM6A.LR.r1i1p1"          "IPSL.CM6A.LR.INCA.r1i1p1"
## [31] "KACE.1.0.G.r1i1p1f1"          "KIOST.ESM.r1i1p1f1"
## [33] "MIROC.ES2L.r1i1p1f2"          "MIROC6.r1i1p1f1"
## [35] "MPI.ESM.1.2.HAM.r1i1p1f1"     "MPI.ESM1.2.HR.r1i1p1f1"
## [37] "MPI.ESM1.2.LR.r1i1p1f1"       "MRI.ESM2.0.r1i1p1f1"
## [39] "NESM3.r1i1p1f1"               "NorCPM1.r1i1p1"
## [41] "NorESM2.LM.r1i1p1f1"          "NorESM2.MM.r1i1p1f1"
## [43] "SAM0.UNICON.r1i1p1f1"         "TaiESM1.r1i1p1f1"
## [45] "UKESM1.0.LL.r1i1p1f2"
```

```r
## Show locations of the sites that were included in the downscaling graphically
map(esd.results$fw.ssp370$pca,cex=0.5,new=FALSE)
```

mean of fw (fraction)

```
## Show the available emission scenarios
print(names(esd.results))
```

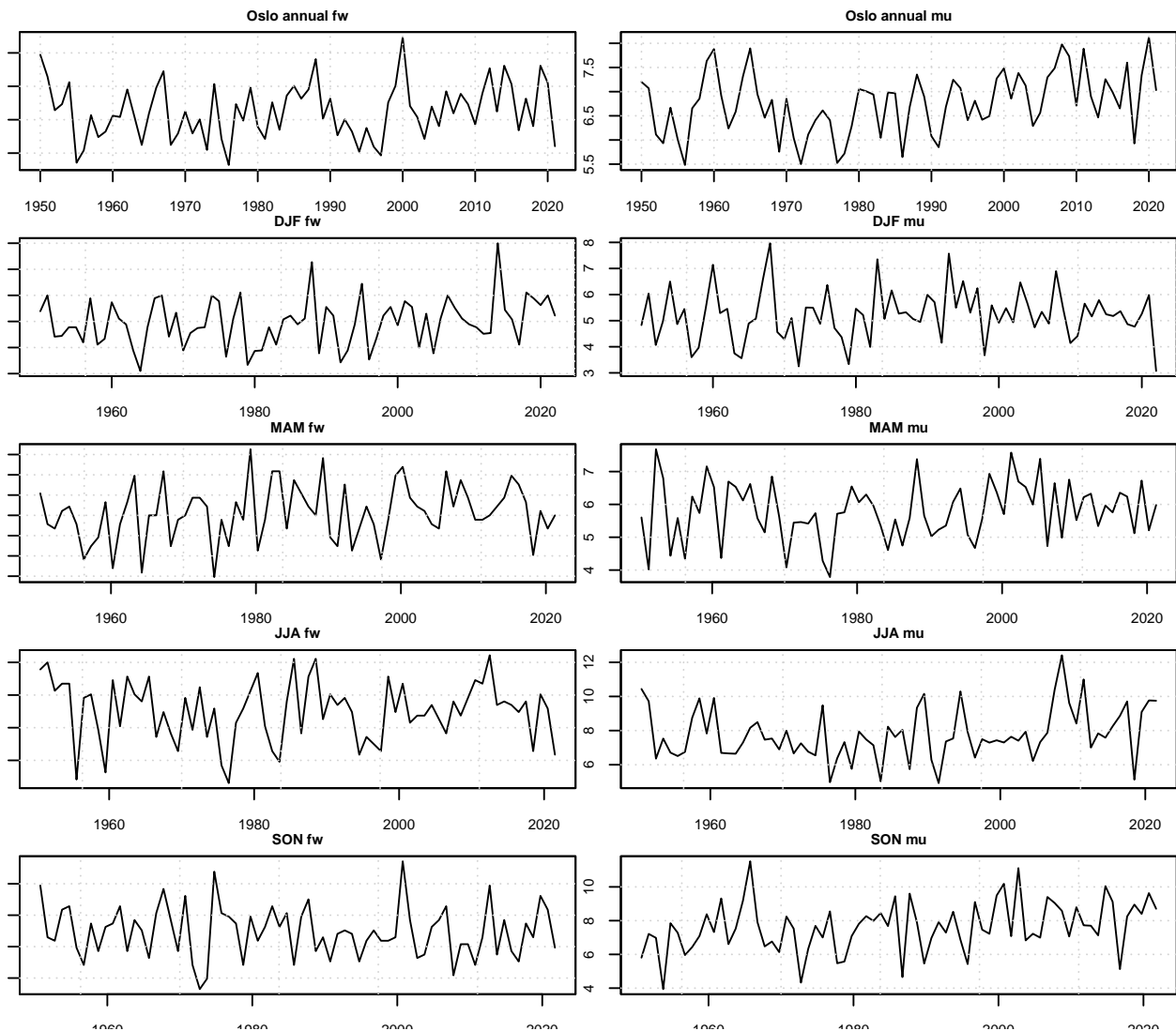
```
##  [1] "fw.hist"   "mu.hist"   "fw.ssp126" "mu.ssp126" "fw.ssp245" "mu.ssp245"
##  [7] "fw.ssp370" "mu.ssp370" "fw.ssp585" "mu.ssp585"
```

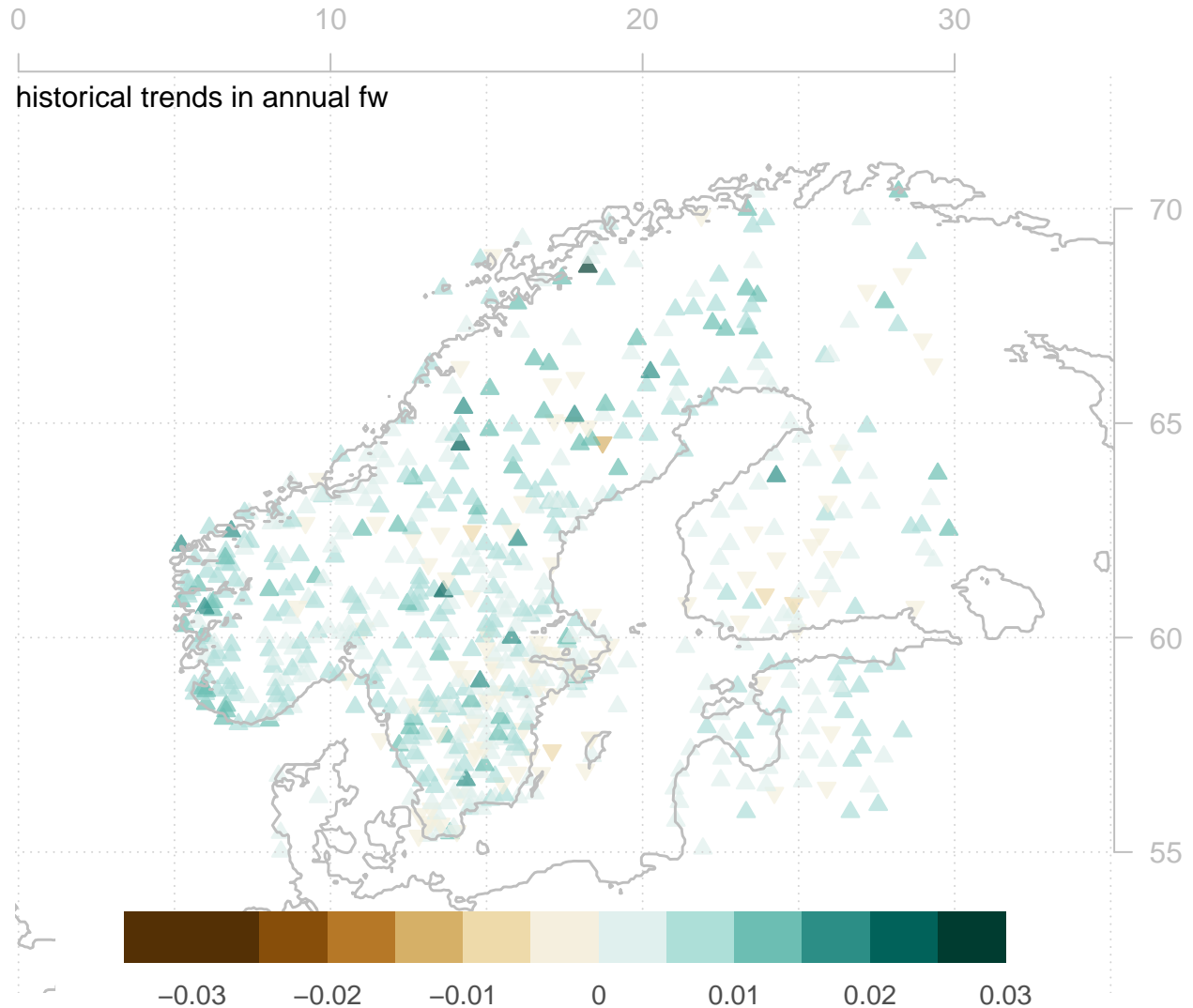
The results can also be inspected by showing the time series for one location, in this case Oslo and the SSP370 emission scenario.

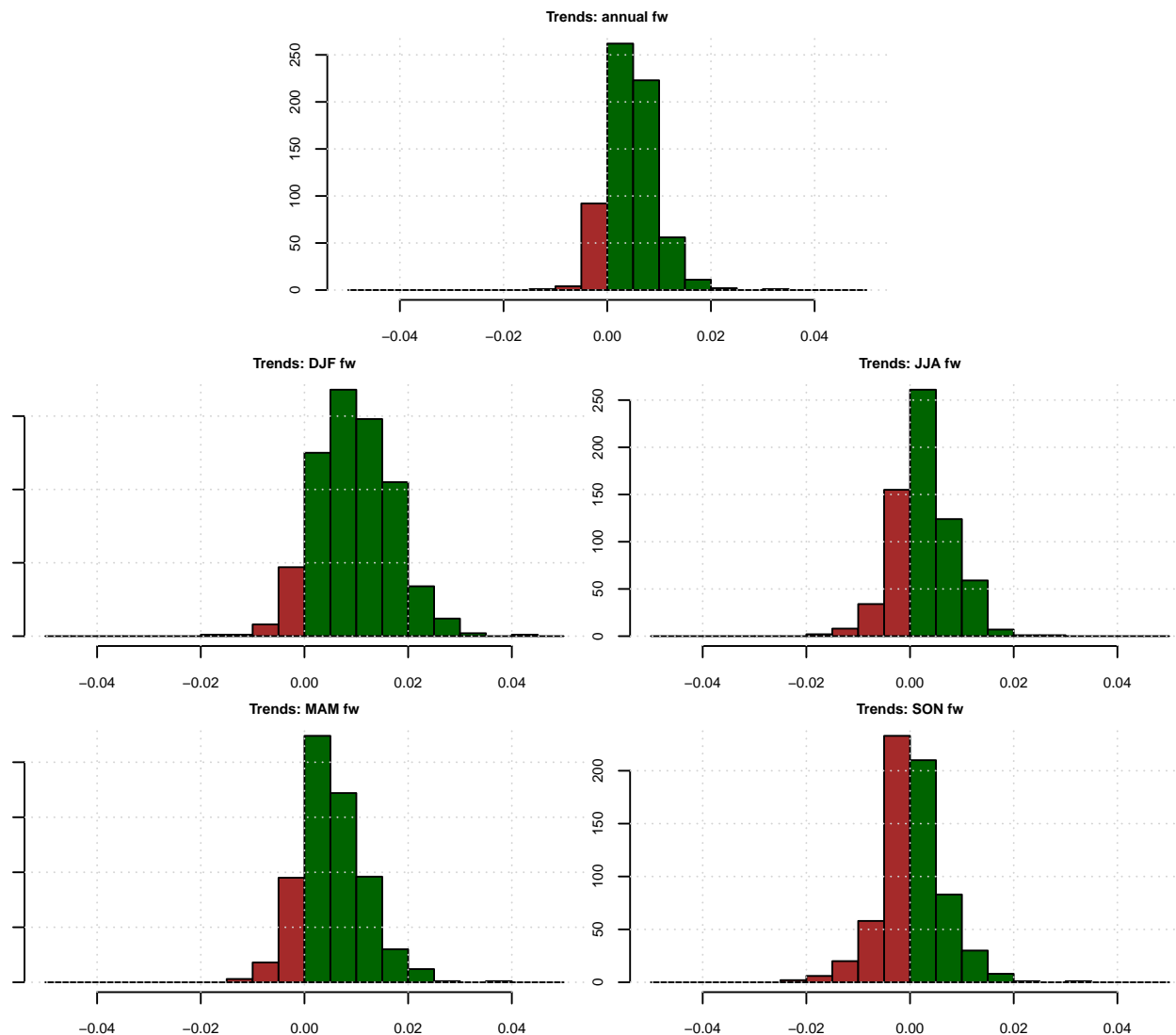
2.1.1 Historical statistics f_w and μ

Since the tend in the annual mean may mask seasonally differing trends, it's instructive to examine trend estimates for the historical data. Below, this is illustrated for Oslo but also statistics are presented for all sites with long time series (more than 20.000 valid data points).

```
oslo <- subset(Y,loc='OSLO BLINDERN')
layout(matrix(1:10,5,2))
par(mar=c(1.5,1,1.5,1),cex.lab=0.7,cex.axis=0.7,cex.main=0.7)
```

```
## Wet-day frequency
plot(zoo(annual(oslo,FUN='wetfreq')),main='Oslo annual fw'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetfreq'),it='djf')),main='DJF fw'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetfreq'),it='mam')),main='MAM fw'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetfreq'),it='jja')),main='JJA fw'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetfreq'),it='son')),main='SON fw'); grid()
## Wet-day mean precipitation
plot(zoo(annual(oslo,FUN='wetmean')),main='Oslo annual mu'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetmean'),it='djf')),main='DJF mu'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetmean'),it='mam')),main='MAM mu'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetmean'),it='jja')),main='JJA mu'); grid()
plot(zoo(subset(as.4seasons(oslo,FUN='wetmean'),it='son')),main='SON mu'); grid()
```

```
## Trend statistics for wetday frequency fw
nv <- apply(Y,2,'nv')
is <- nv >= 20000 ## only apply trend statistics on long time series
fw.ann <- trend.coef(subset(annual(Y,FUN='wetfreq'),is=is))
fw.djf <- trend.coef(subset(as.4seasons(Y,FUN='wetfreq'),it='djf',is=is))
```

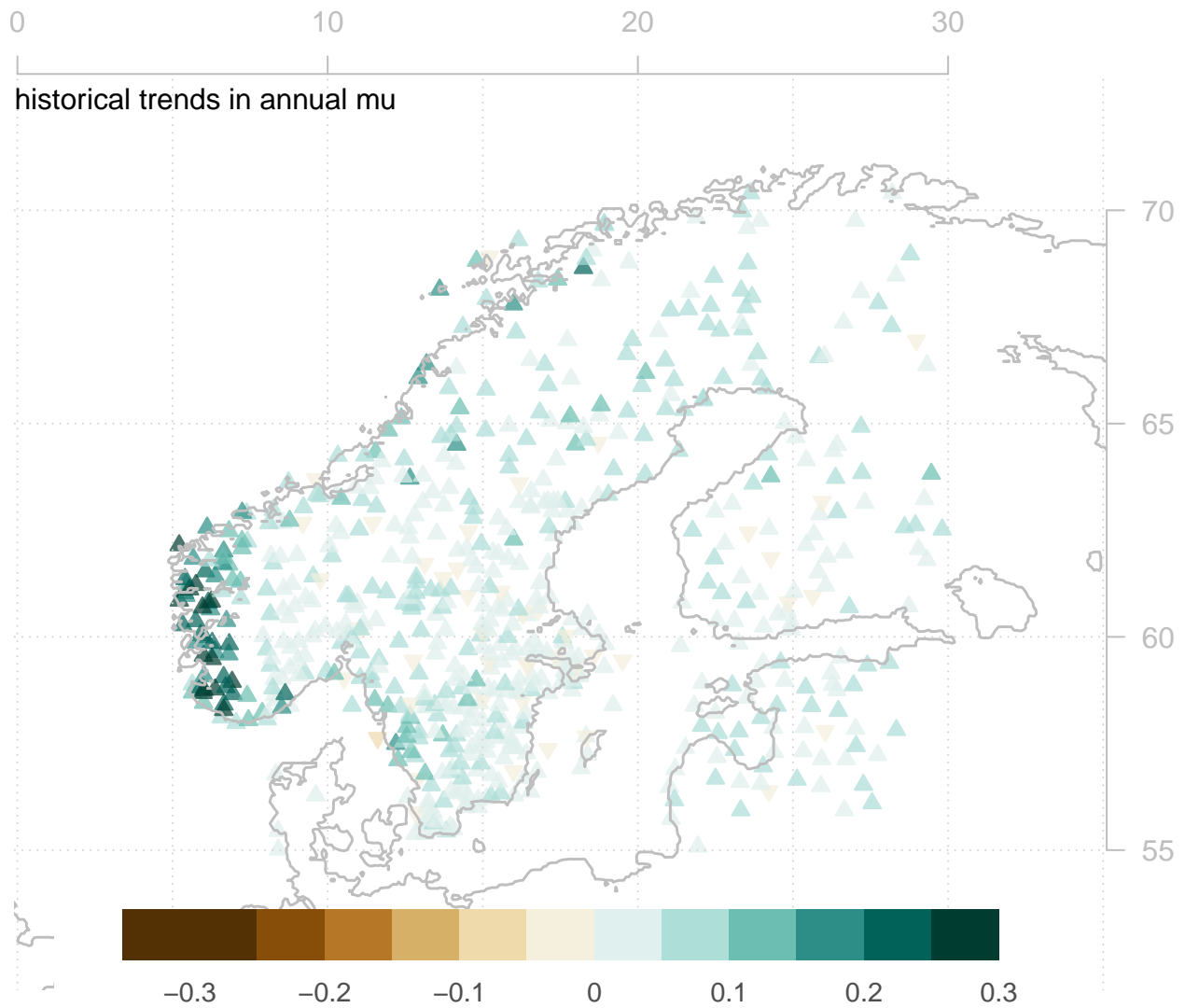
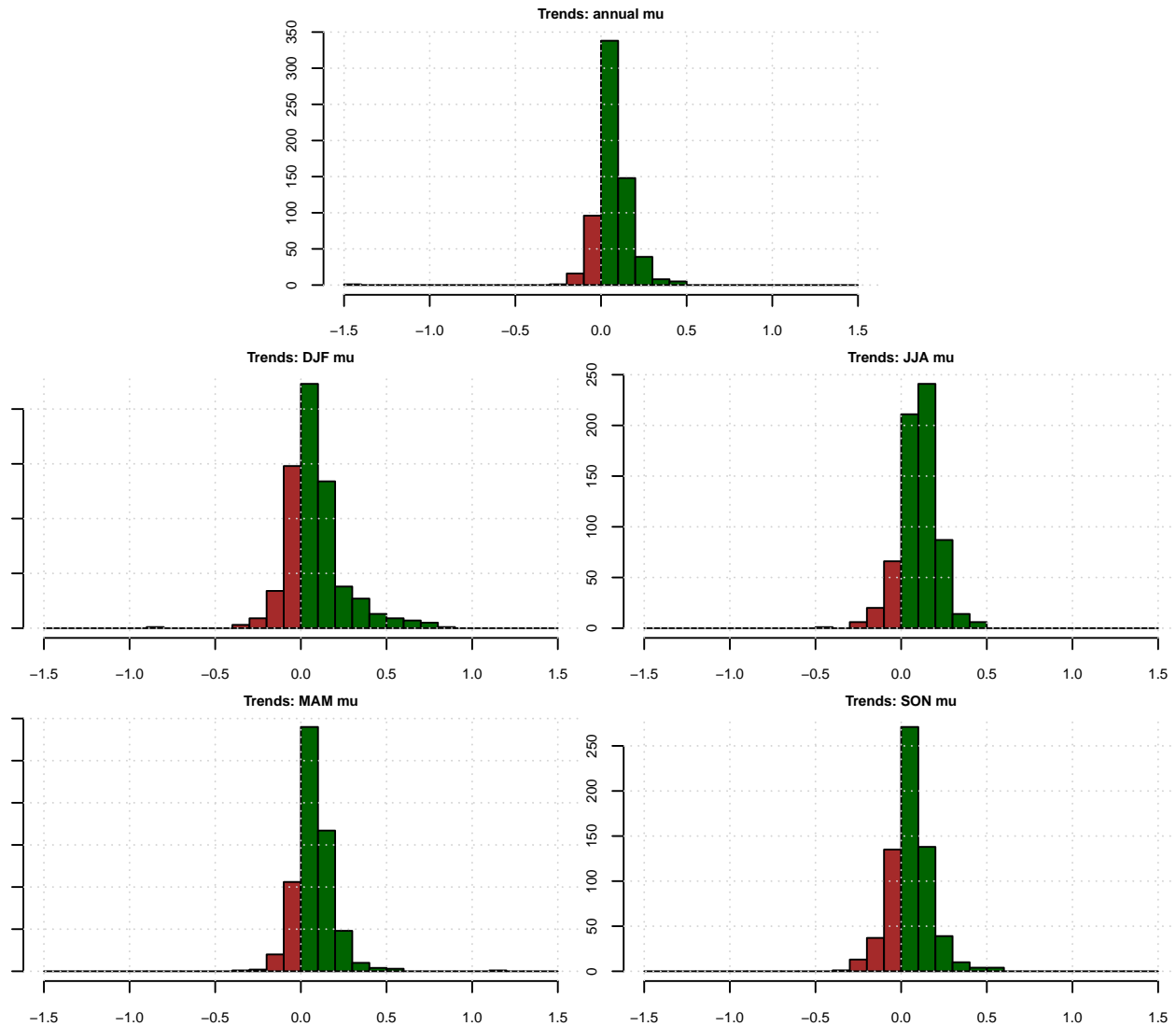
```
fw.mam <- trend.coef(subset(as.4seasons(Y,FUN='wetfreq'),it='mam',is=is))
fw.jja <- trend.coef(subset(as.4seasons(Y,FUN='wetfreq'),it='jja',is=is))
fw.son <- trend.coef(subset(as.4seasons(Y,FUN='wetfreq'),it='son',is=is))

## Visualise the statistics
breaks <- seq(-0.03,0.03,by=0.005)
par(mfcol=c(1,1))
map(subset(annual(Y,FUN='wetfreq'),is=is),FUN='trend',cex=1,
    colbar=list(pal='precip.ipcc',breaks=breaks,rev=TRUE),showaxis=FALSE,
    main='historical trends in annual fw',new=FALSE)
```

```
breaks <- seq(-0.05,0.05,by=0.005)
col <- rep('brown',length(breaks)); col[breaks >= 0] <-'darkgreen'
layout(matrix(c(0,2,3,1,2,3,1,4,5,0,4,5),3,4))
par(mar=c(2,1,1.5,1))
hist(fw.ann,main='Trends: annual fw',breaks=breaks,col=col); grid()
hist(fw.djf,main='Trends: DJF fw',breaks=breaks,col=col); grid()
hist(fw.mam,main='Trends: MAM fw',breaks=breaks,col=col); grid()
hist(fw.jja,main='Trends: JJA fw',breaks=breaks,col=col); grid()
```

```
hist(fw.son,main='Trends: SON fw',breaks=breaks,col=col); grid()
```

```
## Trend statistics for wet-day frequency mu
mu.ann <- trend.coef(subset(annual(Y,FUN='wetmean'),is=is))
```

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

```
mu.djf <- trend.coef(subset(as.4seasons(Y,FUN='wetmean'),it='djf',is=is))
mu.mam <- trend.coef(subset(as.4seasons(Y,FUN='wetmean'),it='mam',is=is))
mu.jja <- trend.coef(subset(as.4seasons(Y,FUN='wetmean'),it='jja',is=is))
mu.son <- trend.coef(subset(as.4seasons(Y,FUN='wetmean'),it='son',is=is))
par(mfcol=c(1,1))
breaks <- seq(-0.3,0.3,by=0.05)
map(subset(annual(Y,FUN='mean'),is=is),FUN='trend',cex=1,
    colbar=list(pal='precip.ipcc',breaks=breaks),showaxis=FALSE,
    main='historical trends in annual mu',new=FALSE)
```

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

historical trends in annual mu

```
breaks <- seq(-1.5,1.5,by=0.1)
col <- rep('brown',length(breaks)); col[breaks >= 0] <-'darkgreen'
layout(matrix(c(0,2,3,1,2,3,1,4,5,0,4,5),3,4))
par(mar=c(2,1,1.5,1))
hist(mu.ann,main='Trends: annual mu',breaks=breaks,col=col); grid()
hist(mu.djf,main='Trends: DJF mu',breaks=breaks,col=col); grid()
hist(mu.mam,main='Trends: MAM mu',breaks=breaks,col=col); grid()
hist(mu.jja,main='Trends: JJA mu',breaks=breaks,col=col); grid()
hist(mu.son,main='Trends: SON mu',breaks=breaks,col=col); grid()
```

The analysis of historical data suggests that there has been mainly increases in the wet-day frequency f_w, albeit weak. Most pronounced increases are seen in winter (DJF).The map suggests that there is a weak increase in annual f_w in general over the Nordic region.

The analysis of historical observations also suggest a majority of locations with increasing wet-day mean precipitation μ for the annual mean as well as most seasons. The map showing trends in annual μ suggest a general historical increase in intensity over the Nordic region.

2.1.2 Simulatons: f_w and μ

Note, the different parameters f_w and μ and emission scenarios (HIST/SSPs) involve somewhat varying sets of GCMs.

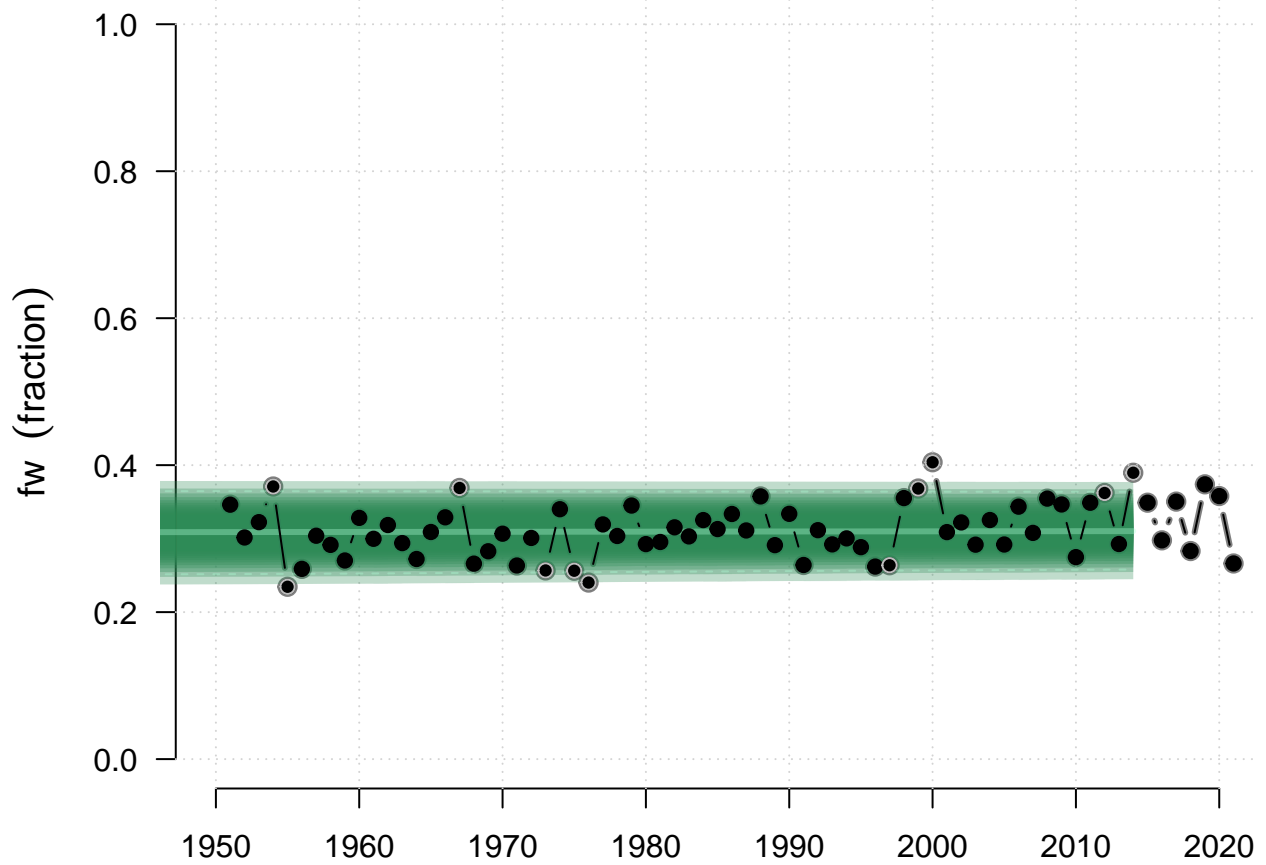
The following R-code chunk presents multi-model ensembles downscaled annual f_w and μ for Oslo for the historical run ("HIST") and future projections ("SSPs").

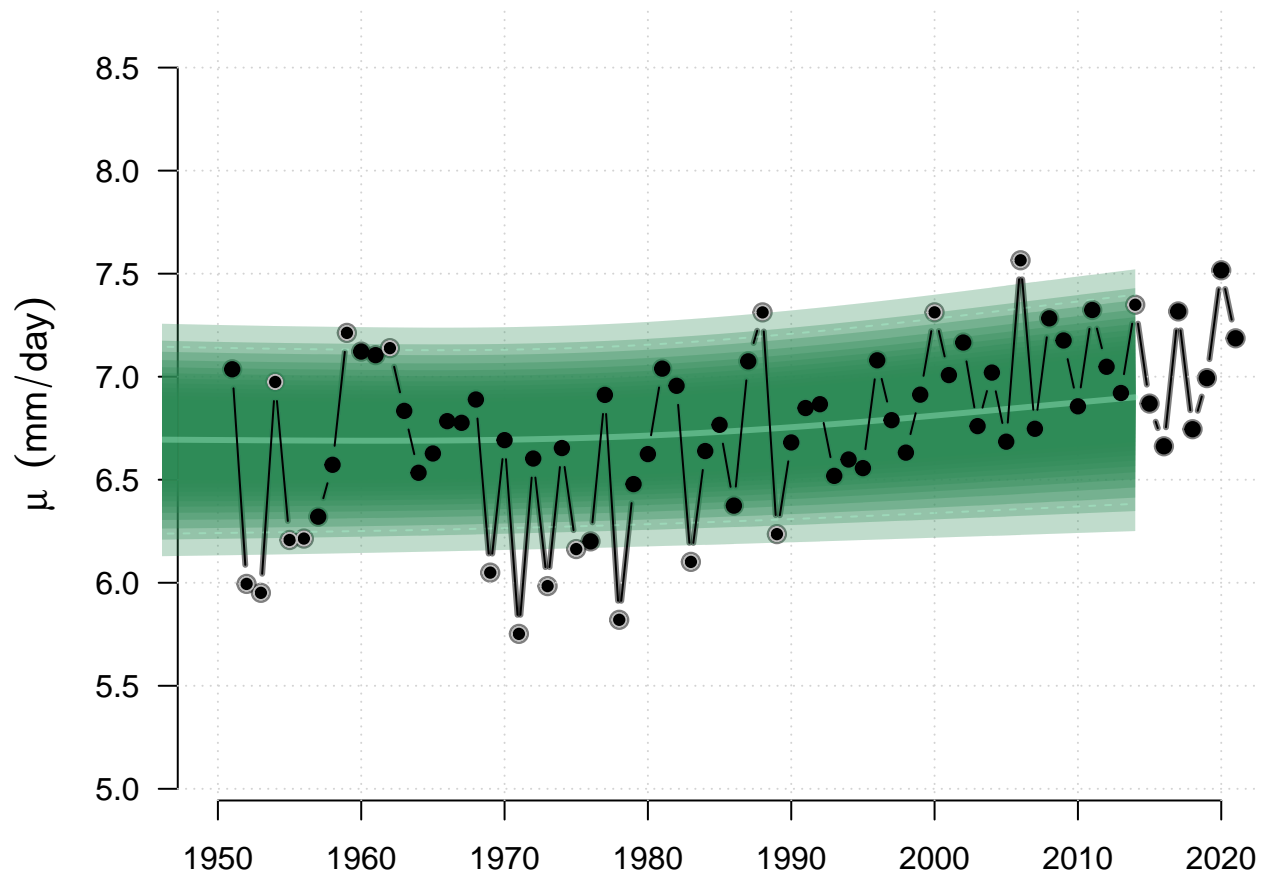
```r
## The allN function extracts the time interval for which the ensemble has valid
## data for all its members
allN <- function(x,verbose=FALSE) {
```

```r
  ## Remove members with short time span
  nt <- length(index(x))
  cls <- class(x)
  if (verbose) {print('Remove members with short time span:'); print(dim(x))}
  nv <- apply(coredata(x),2,FUN='nv')
  if (verbose) print(nv)
  im <- nv >= quantile(nv,probs = 0.75)
  x <- subset(x,im=im,verbose=verbose)
  ## Remove times with incomplete ensemble
  if (verbose) {print('Select interval with complet ensemble:'); print(dim(x))}
  nv <- apply(coredata(x),1,FUN='nv')
  if (verbose) print(nv)
  it <- nv==max(nv,na.rm=TRUE)
  y <- subset(x,it=it,verbose=verbose)
  y <- attrcp(x,y)
  class(y) <- cls
  if (sum(it) != dim(y)[1]) browser()
  ## Adjust the offset of the ensemble to match recent years of the observations
  if (verbose) print('Adjust offset')
  obs <- attr(x,'station')
  yrs <- sort(year(obs),decreasing = TRUE)
  i1 <- is.element(year(obs),year(x))
  i2 <- is.element(year(x),year(obs))
  mobs <- round(mean(coredata(obs[i2])),2)
  if (verbose) print(mobs)
  for (i in 1:dim(x)[2]) x[,i] <- x[,i] - mean(coredata(x[i1,i])) + mobs
  return(y)
}
## HIST
## The wet-day frequency
print("HIST")
```

```
## [1] "HIST"
```

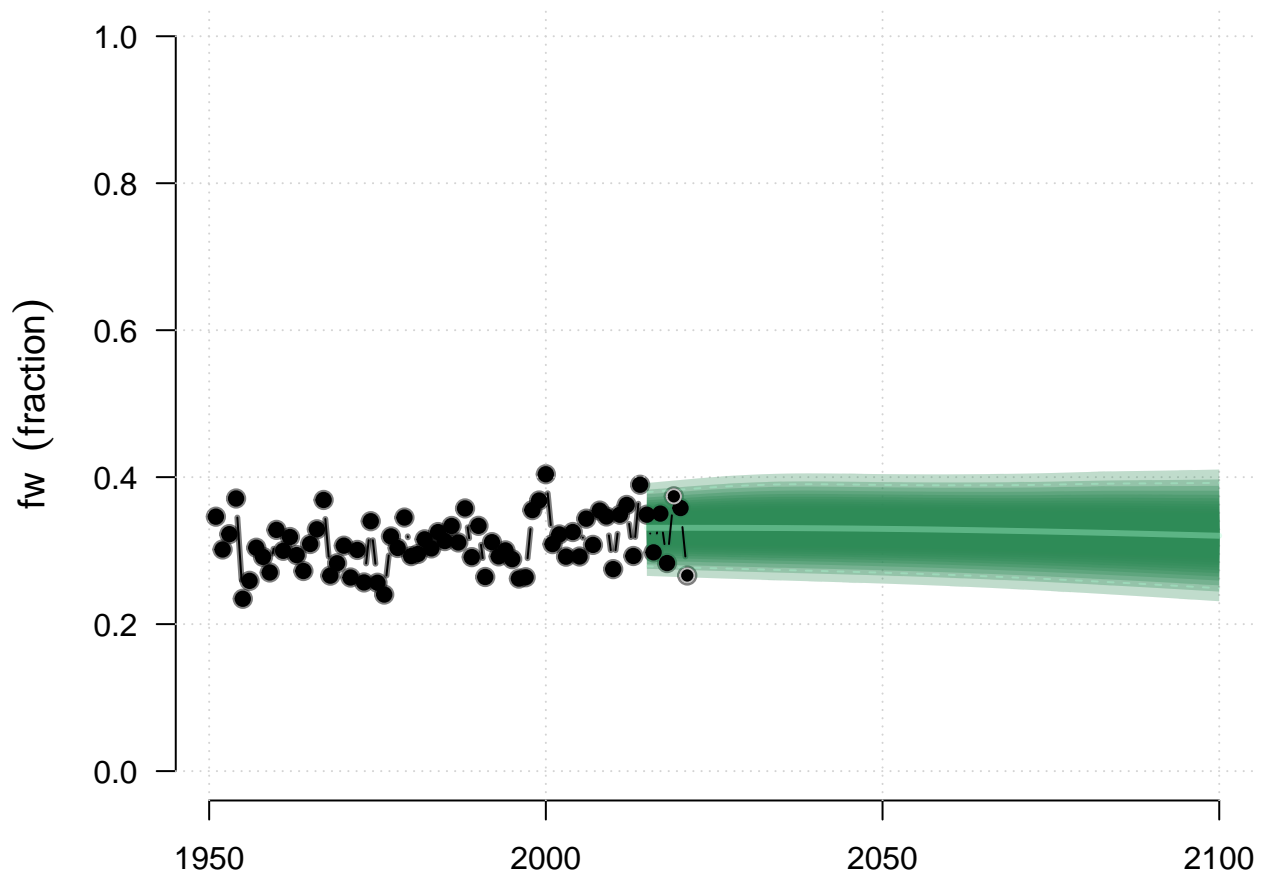
```r
fw.oslo.hist <- as.station(esd.results$fw.hist)[["OSLO BLINDERN"]]
## Remove members with incomplete data
plot(allN(fw.oslo.hist),ylim=c(0,1),xlim=c(1950,2020),new=FALSE,
     main='Oslo: wet-day frequency: HIST')
```

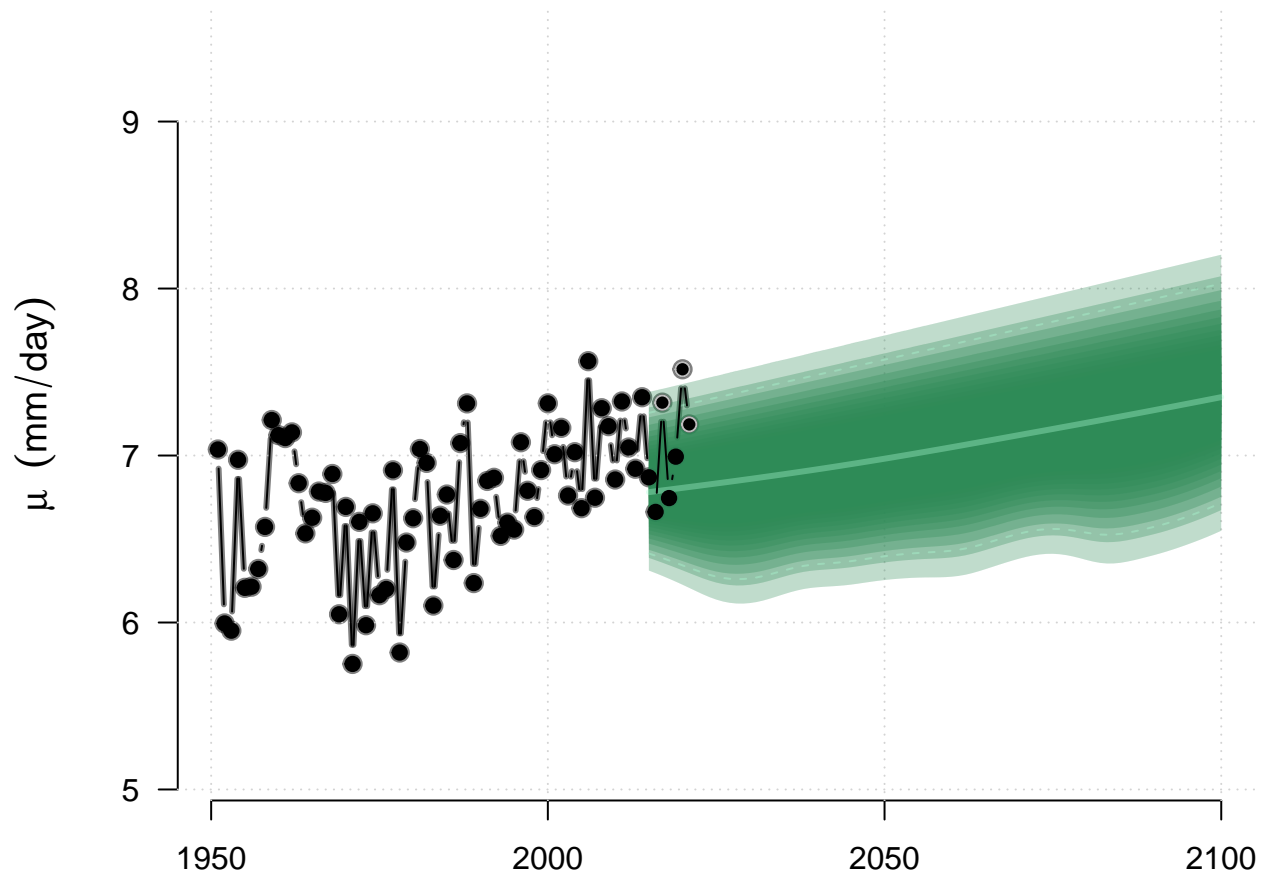
Oslo: wet–day frequency: HIST

```
mu.oslo.hist <- as.station(esd.results$mu.hist)[["OSLO BLINDERN"]]
plot(allN(mu.oslo.hist),xlim=c(1950,2020),new=FALSE,
     main='Oslo: mean precipitation intensity: HIST')
```

Oslo: mean precipitation intensity: HIST

```
## SSP370
print("SSP370")
```

```
## [1] "SSP370"
```

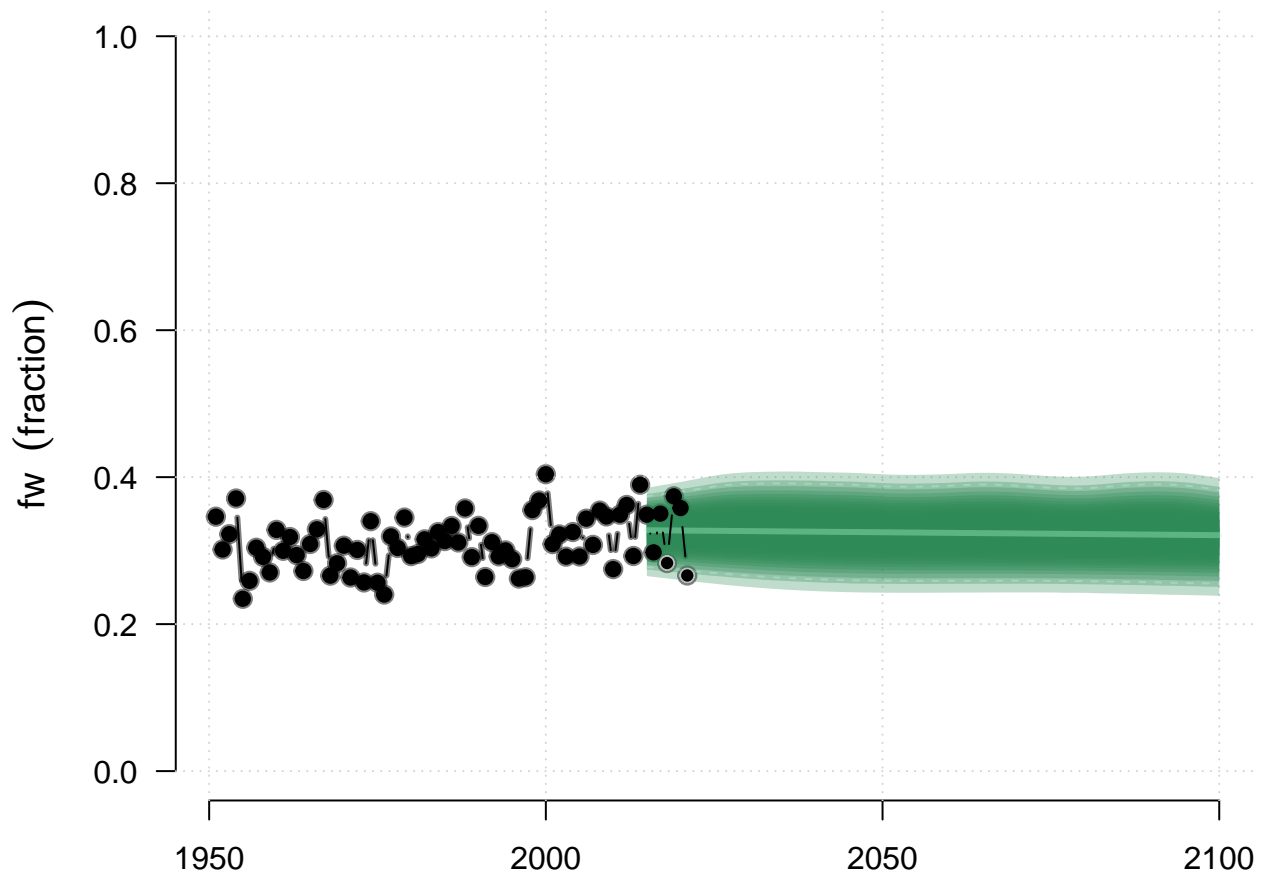
```
## The wet-day frequency
fw.oslo.ssp370 <- as.station(esd.results$fw.ssp370)[["OSLO BLINDERN"]]
## Remove members with incomplete data
plot(allN(fw.oslo.ssp370),ylim=c(0,1),new=FALSE,
     main='Oslo: wet-day frequency: SSP370')
```

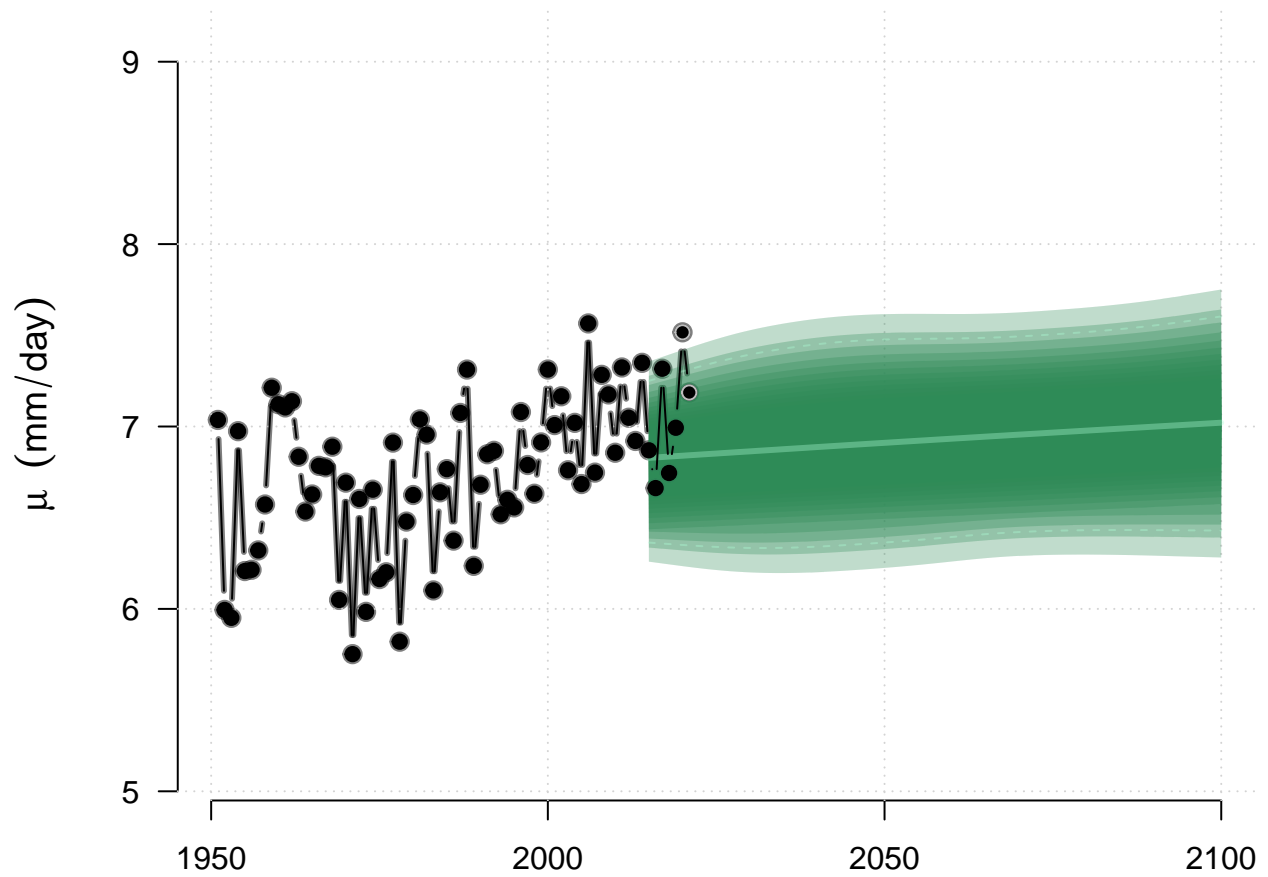
Oslo: wet-day frequency: SSP370

```
mu.oslo.ssp370 <- as.station(esd.results$mu.ssp370)[["OSLO BLINDERN"]]
plot(allN(mu.oslo.ssp370),new=FALSE,
     main='Oslo: mean precipitation intensity: SSP370')
```

Oslo: mean precipitation intensity: SSP370

```
## SSP126
print("SSP126")
```

```
## [1] "SSP126"
```

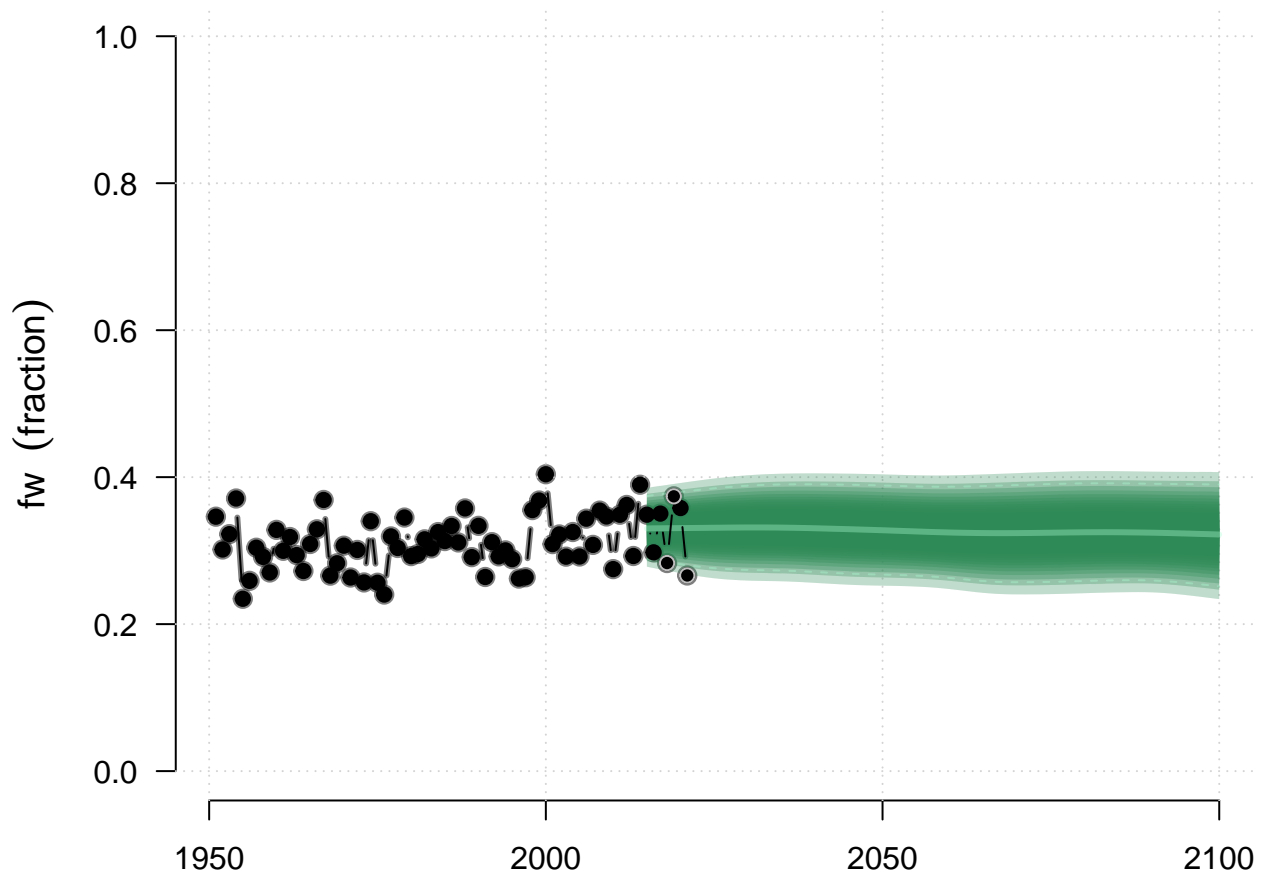
```
fw.oslo.ssp126 <- as.station(esd.results$fw.ssp126)[["OSLO BLINDERN"]]
## Remove members with incomplete data
plot(allN(fw.oslo.ssp126),ylim=c(0,1),new=FALSE,
     main='Oslo: wet-day frequency: SSP126')
```

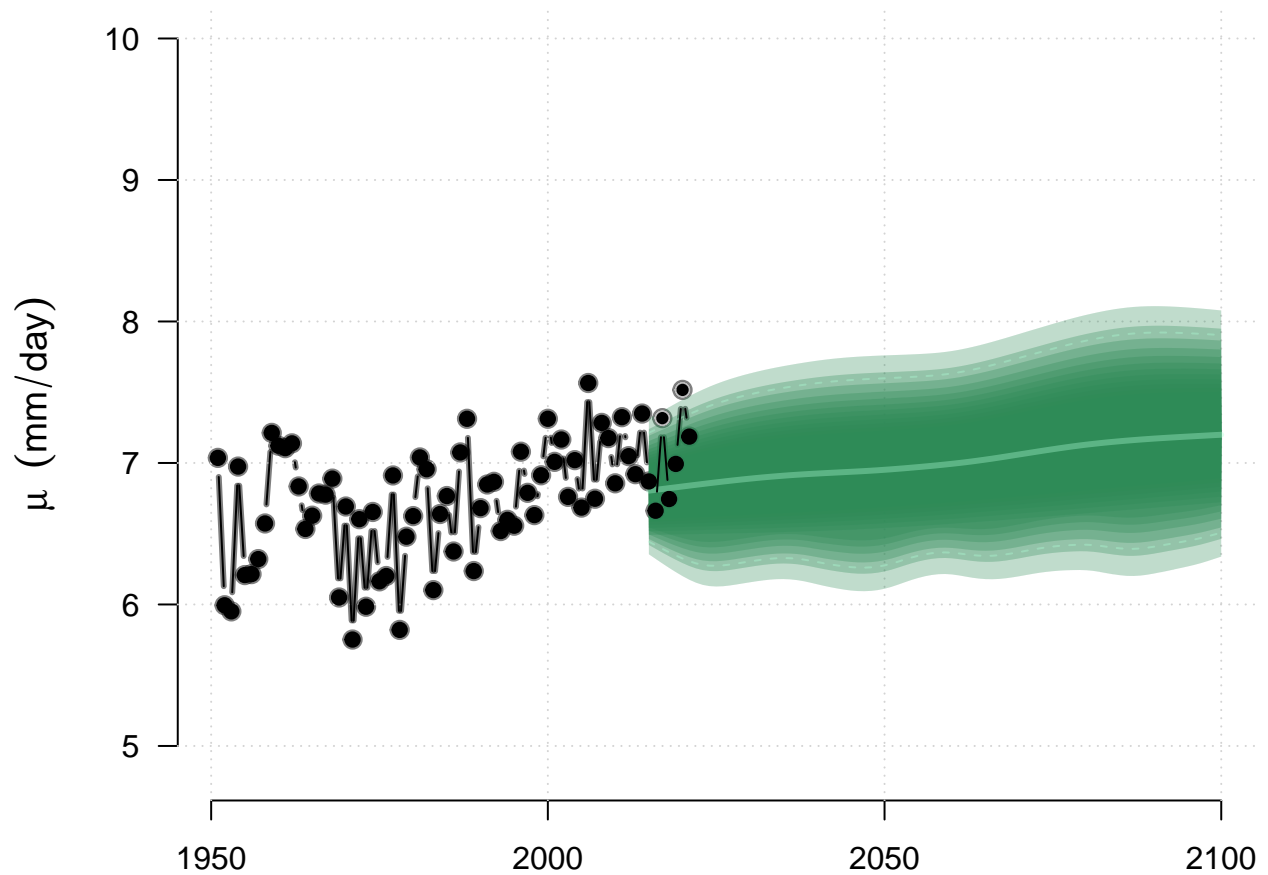
Oslo: wet−day frequency: SSP126

```
mu.oslo.ssp126 <- as.station(esd.results$mu.ssp126)[["OSLO BLINDERN"]]
plot(allN(mu.oslo.ssp126),new=FALSE,
     main='Oslo: mean precipitation intensity: SSP126')
```

Oslo: mean precipitation intensity: SSP126

```
## SSP245
print("SSP245")
```

```
## [1] "SSP245"
```

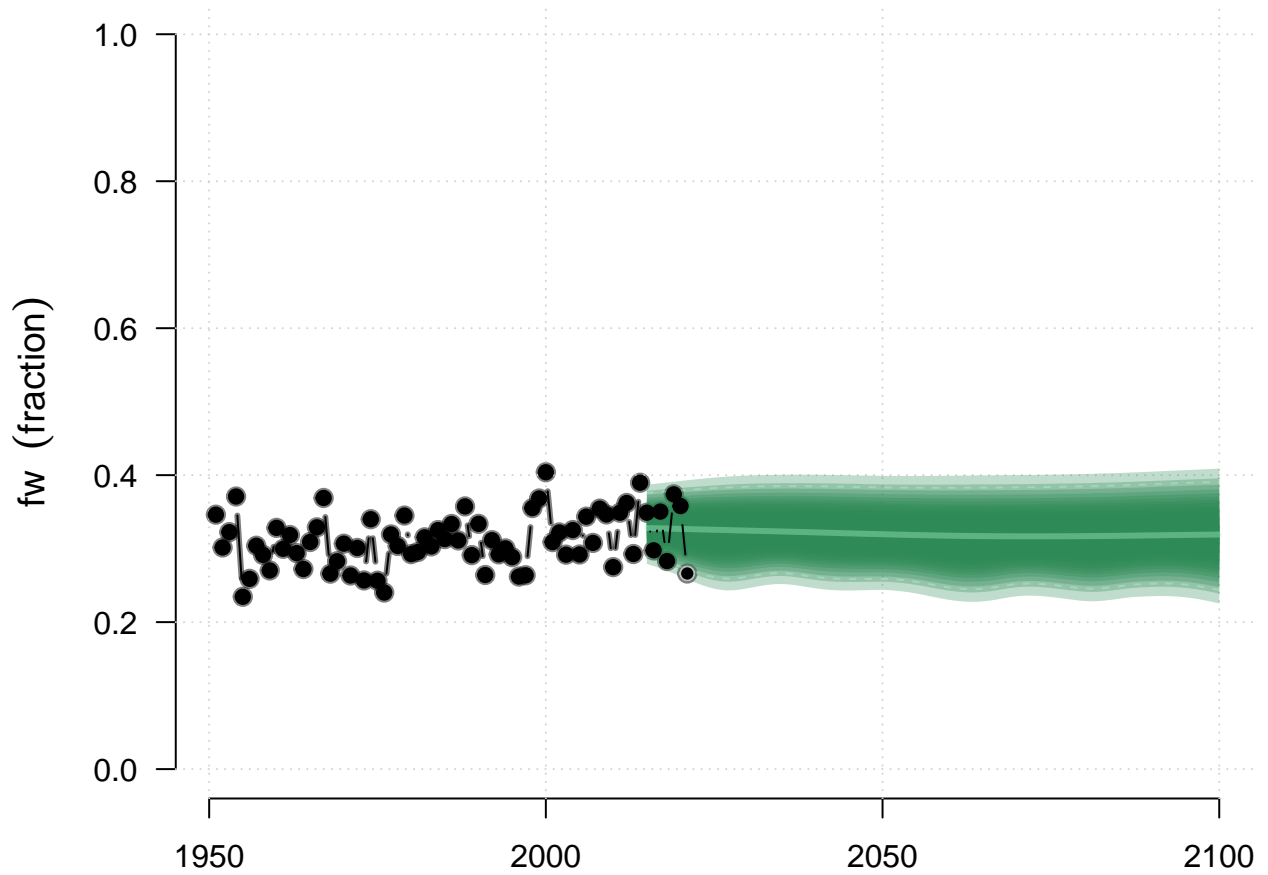
```
fw.oslo.ssp245 <- as.station(esd.results$fw.ssp245)[["OSLO BLINDERN"]]
## Remove members with incomplete data
plot(allN(fw.oslo.ssp245),ylim=c(0,1),new=FALSE,
     main='Oslo: wet-day frequency: SSP245')
```

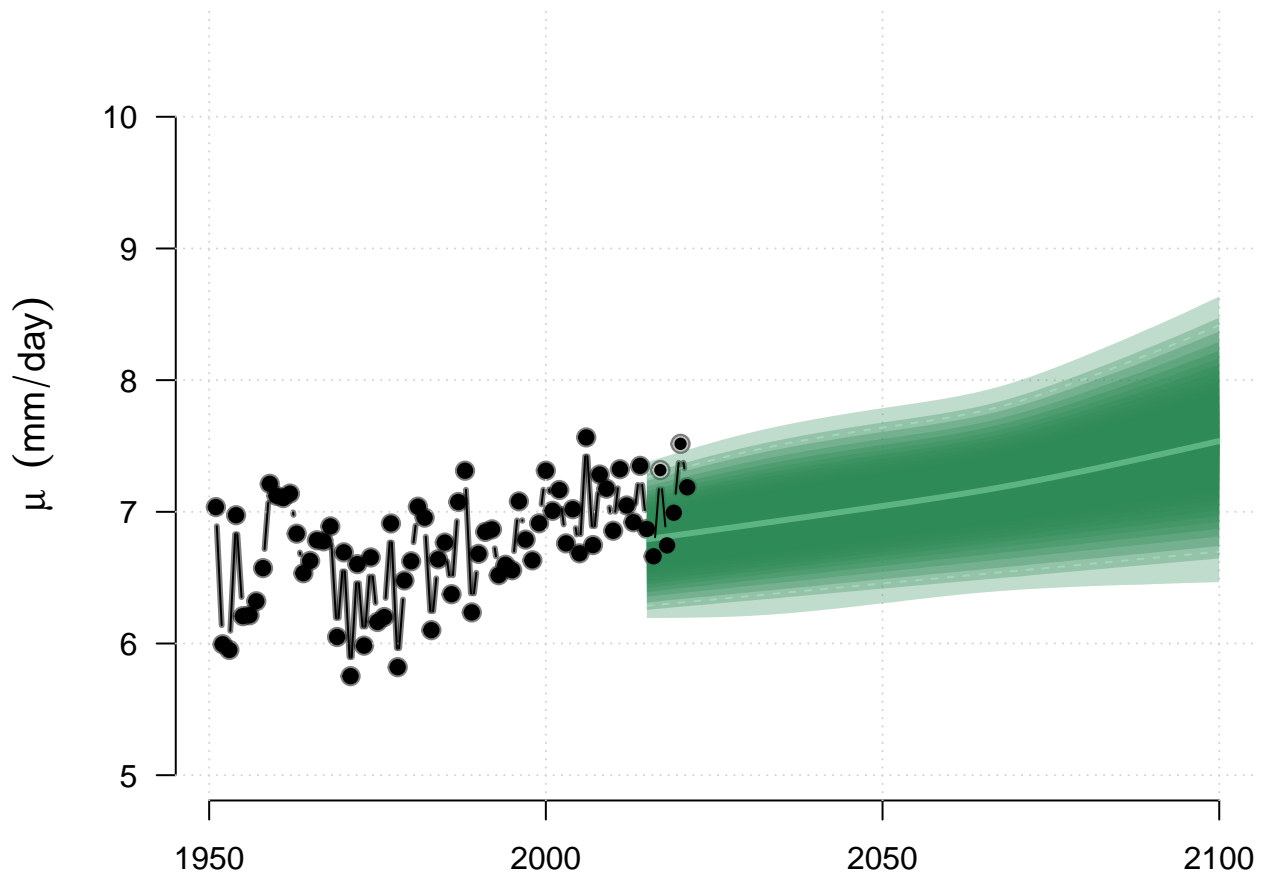
Oslo: wet−day frequency: SSP245

```
mu.oslo.ssp245 <- as.station(esd.results$mu.ssp245)[["OSLO BLINDERN"]]
plot(allN(mu.oslo.ssp245),new=FALSE,
     main='Oslo: mean precipitation intensity: SSP245')
```

Oslo: mean precipitation intensity: SSP245

```
## SSP585
print("SSP585")
```

```
## [1] "SSP585"
```

```
fw.oslo.ssp585 <- as.station(esd.results$fw.ssp585)[["OSLO BLINDERN"]]
## Remove members with incomplete data
plot(allN(fw.oslo.ssp585),ylim=c(0,1),new=FALSE,
     main='Oslo: wet-day frequency: SSP585')
```

Oslo: wet−day frequency: SSP585

```
mu.oslo.ssp585 <- as.station(esd.results$mu.ssp585)[["OSLO BLINDERN"]]
plot(allN(mu.oslo.ssp585),new=FALSE,
     main='Oslo: mean precipitation intensity: SSP585')
```

Oslo: mean precipitation intensity: SSP585

The results for the historical runs indicated that the downscaled ensemble generally span the observed annual f_w (black symbols) tightly, but μ is low biased because it uses a climatology defined for the whole period of observations and there is a historical upward trend in μ.

The local projection based on SSP370 indicated that no clear future trend in f_w, which is consistent with the past. For μ, there is a slight increasing tendency and there has been an long-term increase in the observations (black symbols). The downscaled results seem to underestimate the magnitude of the long-term trend, judging from the historical trend (HIST and SSP370), but the projections for μ connected to SSP585 indicated more a pronounced long-term trend.

The length of the scenario projections was determined by the availability of all members within the ensemble. Some models dropped out after 2050, and changes in the ensemble sample gave misleading impressions. The results for SSP245 for μ suffered from small ensemble size.

The downscaled μ for SSP126 were spurious, indicating that there may have been a data issue, most likely with the extracted information from the CMIP6 GCMs.

2.1.3 Table of selected station projections

```r
cities <- sort(c("STOCKHOLM","ABISKO","FALUN","GEIRANGER","GEILO","HALDEN",
            "KRISTINEHAMN","MALMO","TALLINN","HELSINKI KAISANIEMI",
            "TRANEBERG","GRONBAEK-ALLINGSKOVGARD","VESTERVIG",
            "NORDBY (FANO)","GOTSKA SANDON","OSLO BLINDERN"))
```

```
ssps <- c('ssp370','ssp126','ssp245','ssp585')
sites.ssps <- list()
for (city in cities) {
  for (ssp in ssps) {
  fw.ssp370 <- as.station(esd.results[[paste0('fw.',ssp)]])[[city]]
  mu.ssp370 <- as.station(esd.results[[paste0('mu.',ssp)]])[[city]]
  sites.ssps[[paste0('fw.',ssp,'.',tolower(city))]] <- fw.ssp370
  sites.ssps[[paste0('mu.',ssp,'.',tolower(city))]] <- mu.ssp370
  }
}

## Table on wet-day frequency
fw.sites <- names(sites.ssps)[grep('fw.',names(sites.ssps))]
print('Location/SSP 2071-2100 & mean fw & std fw & mean mu & std mu \\n ')
```

```
## [1] "Location/SSP 2071-2100 & mean fw & std fw & mean mu & std mu \\n "
```

```
for (fw.site in fw.sites) {
  z.fw <- allN(sites.ssps[[fw.site]])
  z.mu <- allN(sites.ssps[[sub('fw.','mu.',fw.site)]])
  print(paste(sub('fw.','',fw.site),round(mean(window(z.fw,start=2071,end=2100)),2),
              round(sd(window(z.fw,start=2071,end=2100)),2),
              round(mean(window(z.mu,start=2071,end=2100)),2),
              round(sd(window(z.mu,start=2071,end=2100)),2),'\\n',sep=' & '))
}
```

```
## [1] "ssp370.abisko & 0.21 & 0.02 & 4.14 & 0.11 & \\n"
## [1] "ssp126.abisko & 0.21 & 0.02 & 4.1 & 0.11 & \\n"
## [1] "ssp245.abisko & 0.21 & 0.02 & 4.13 & 0.12 & \\n"
## [1] "ssp585.abisko & 0.21 & 0.02 & 4.16 & 0.13 & \\n"
## [1] "ssp370.falun & 0.29 & 0.02 & 5.52 & 0.12 & \\n"
## [1] "ssp126.falun & 0.29 & 0.02 & 5.44 & 0.11 & \\n"
## [1] "ssp245.falun & 0.29 & 0.02 & 5.49 & 0.13 & \\n"
## [1] "ssp585.falun & 0.29 & 0.03 & 5.57 & 0.15 & \\n"
## [1] "ssp370.geilo & 0.36 & 0.04 & 6.23 & 0.2 & \\n"
## [1] "ssp126.geilo & 0.36 & 0.04 & 6.07 & 0.2 & \\n"
## [1] "ssp245.geilo & 0.36 & 0.04 & 6.16 & 0.24 & \\n"
## [1] "ssp585.geilo & 0.35 & 0.04 & 6.31 & 0.24 & \\n"
## [1] "ssp370.geiranger & 0.44 & 0.06 & 9.24 & 0.78 & \\n"
## [1] "ssp126.geiranger & 0.44 & 0.06 & 9.1 & 0.78 & \\n"
## [1] "ssp245.geiranger & 0.44 & 0.06 & 9.12 & 0.8 & \\n"
## [1] "ssp585.geiranger & 0.44 & 0.05 & 9.28 & 0.9 & \\n"
## [1] "ssp370.gotska sandon & 0.27 & 0.03 & 5.27 & 0.21 & \\n"
## [1] "ssp126.gotska sandon & 0.28 & 0.03 & 5.08 & 0.19 & \\n"
## [1] "ssp245.gotska sandon & 0.27 & 0.03 & 5.17 & 0.24 & \\n"
## [1] "ssp585.gotska sandon & 0.27 & 0.03 & 5.37 & 0.26 & \\n"
## [1] "ssp370.gronbaek-allingskovgard & 0.36 & 0.04 & 5.76 & 0.16 & \\n"
## [1] "ssp126.gronbaek-allingskovgard & 0.36 & 0.04 & 5.59 & 0.15 & \\n"
## [1] "ssp245.gronbaek-allingskovgard & 0.36 & 0.04 & 5.68 & 0.2 & \\n"
## [1] "ssp585.gronbaek-allingskovgard & 0.36 & 0.04 & 5.85 & 0.22 & \\n"
## [1] "ssp370.halden & 0.34 & 0.05 & 7.18 & 0.43 & \\n"
## [1] "ssp126.halden & 0.34 & 0.05 & 7.06 & 0.35 & \\n"
## [1] "ssp245.halden & 0.34 & 0.05 & 7.16 & 0.46 & \\n"
```

```
## [1] "ssp585.halden & 0.33 & 0.05 & 7.24 & 0.44 & \\n"
## [1] "ssp370.helsinki kaisaniemi & 0.32 & 0.04 & 5.89 & 0.34 & \\n"
## [1] "ssp126.helsinki kaisaniemi & 0.32 & 0.04 & 5.62 & 0.28 & \\n"
## [1] "ssp245.helsinki kaisaniemi & 0.32 & 0.04 & 5.75 & 0.36 & \\n"
## [1] "ssp585.helsinki kaisaniemi & 0.31 & 0.04 & 6.05 & 0.43 & \\n"
## [1] "ssp370.kristinehamn & 0.32 & 0.03 & 6.16 & 0.24 & \\n"
## [1] "ssp126.kristinehamn & 0.32 & 0.03 & 6.01 & 0.22 & \\n"
## [1] "ssp245.kristinehamn & 0.32 & 0.03 & 6.11 & 0.26 & \\n"
## [1] "ssp585.kristinehamn & 0.31 & 0.03 & 6.25 & 0.3 & \\n"
## [1] "ssp370.malmo & 0.3 & 0.02 & 5.72 & 0.25 & \\n"
## [1] "ssp126.malmo & 0.31 & 0.02 & 5.47 & 0.22 & \\n"
## [1] "ssp245.malmo & 0.3 & 0.03 & 5.6 & 0.29 & \\n"
## [1] "ssp585.malmo & 0.3 & 0.03 & 5.86 & 0.34 & \\n"
## [1] "ssp370.nordby (fano) & 0.38 & 0.04 & 6.16 & 0.24 & \\n"
## [1] "ssp126.nordby (fano) & 0.38 & 0.04 & 5.96 & 0.2 & \\n"
## [1] "ssp245.nordby (fano) & 0.38 & 0.04 & 6.08 & 0.27 & \\n"
## [1] "ssp585.nordby (fano) & 0.37 & 0.04 & 6.27 & 0.3 & \\n"
## [1] "ssp370.oslo blindern & 0.32 & 0.04 & 7.24 & 0.41 & \\n"
## [1] "ssp126.oslo blindern & 0.32 & 0.04 & 6.99 & 0.35 & \\n"
## [1] "ssp245.oslo blindern & 0.32 & 0.04 & 7.15 & 0.45 & \\n"
## [1] "ssp585.oslo blindern & 0.32 & 0.04 & 7.38 & 0.47 & \\n"
## [1] "ssp370.stockholm & 0.28 & 0.03 & 5.31 & 0.21 & \\n"
## [1] "ssp126.stockholm & 0.29 & 0.03 & 5.15 & 0.17 & \\n"
## [1] "ssp245.stockholm & 0.29 & 0.03 & 5.24 & 0.22 & \\n"
## [1] "ssp585.stockholm & 0.28 & 0.03 & 5.39 & 0.27 & \\n"
## [1] "ssp370.tallinn & 0.34 & 0.04 & 5.67 & 0.31 & \\n"
## [1] "ssp126.tallinn & 0.35 & 0.04 & 5.39 & 0.28 & \\n"
## [1] "ssp245.tallinn & 0.34 & 0.04 & 5.52 & 0.36 & \\n"
## [1] "ssp585.tallinn & 0.33 & 0.04 & 5.82 & 0.4 & \\n"
## [1] "ssp370.traneberg & 0.3 & 0.04 & 5.78 & 0.23 & \\n"
## [1] "ssp126.traneberg & 0.31 & 0.04 & 5.7 & 0.21 & \\n"
## [1] "ssp245.traneberg & 0.3 & 0.04 & 5.76 & 0.23 & \\n"
## [1] "ssp585.traneberg & 0.3 & 0.04 & 5.83 & 0.27 & \\n"
## [1] "ssp370.vestervig & 0.37 & 0.04 & 6 & 0.18 & \\n"
## [1] "ssp126.vestervig & 0.37 & 0.04 & 6.01 & 0.16 & \\n"
## [1] "ssp245.vestervig & 0.37 & 0.04 & 6.01 & 0.18 & \\n"
## [1] "ssp585.vestervig & 0.36 & 0.05 & 5.99 & 0.2 & \\n"
```

2.2 Is the ensemble normally distributed?

It's useful to check if the Oslo results $x \sim \mathcal{N}(\mu, \sigma^2)$, because the information embedded in an ensemble with normally distributed members can be quantified in terms of the ensemble mean and standard deviation. The test for normal distribution was only applied to the interval when all ensemble members had valid data.

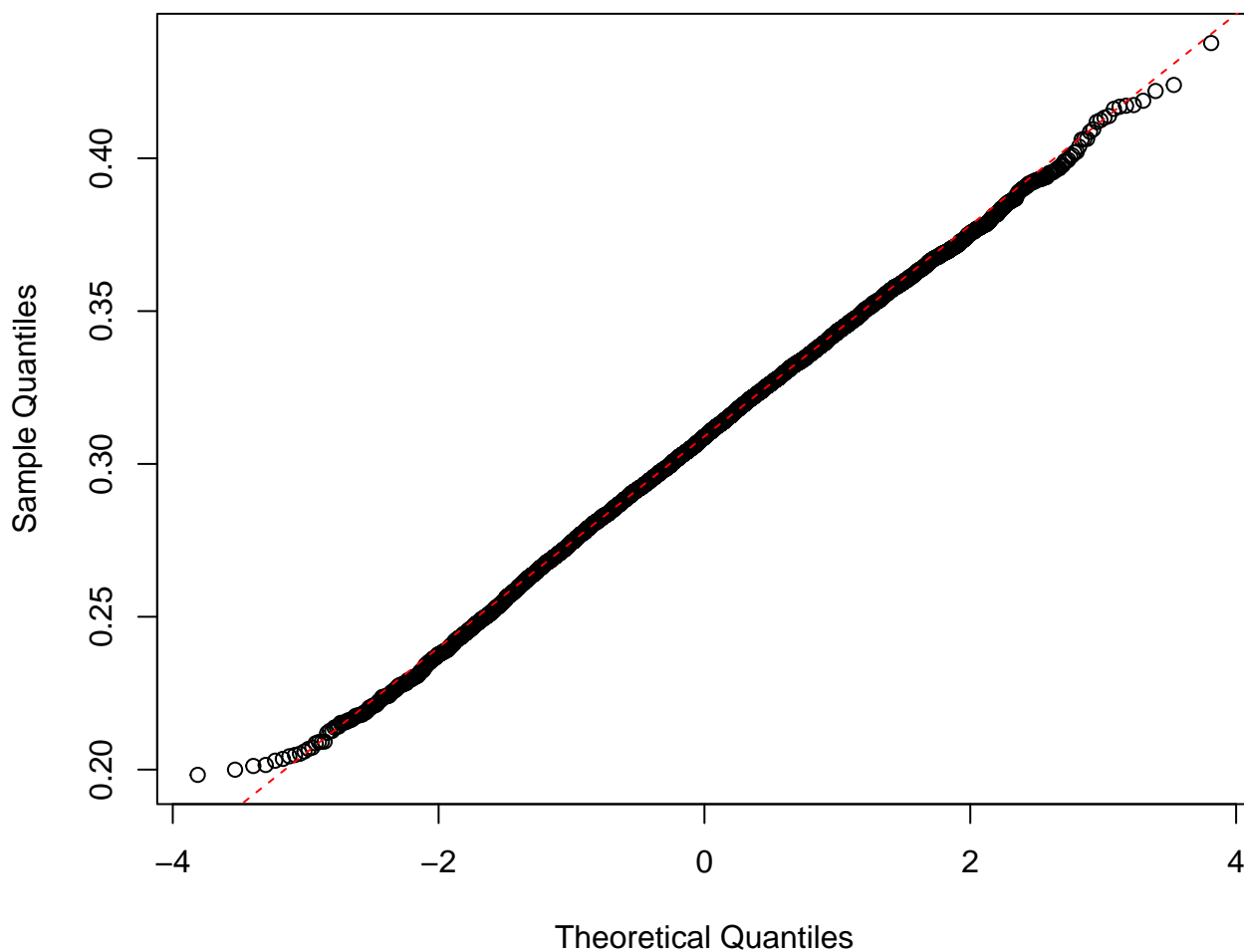
```
print("HIST")
```

```
## [1] "HIST"
```

```
## HIST
qqnorm(coredata(allN(fw.oslo.hist)),main='Oslo: Wet-day frequency HIST')
qqline(coredata(allN(fw.oslo.hist)),lty=2,col='red')
```

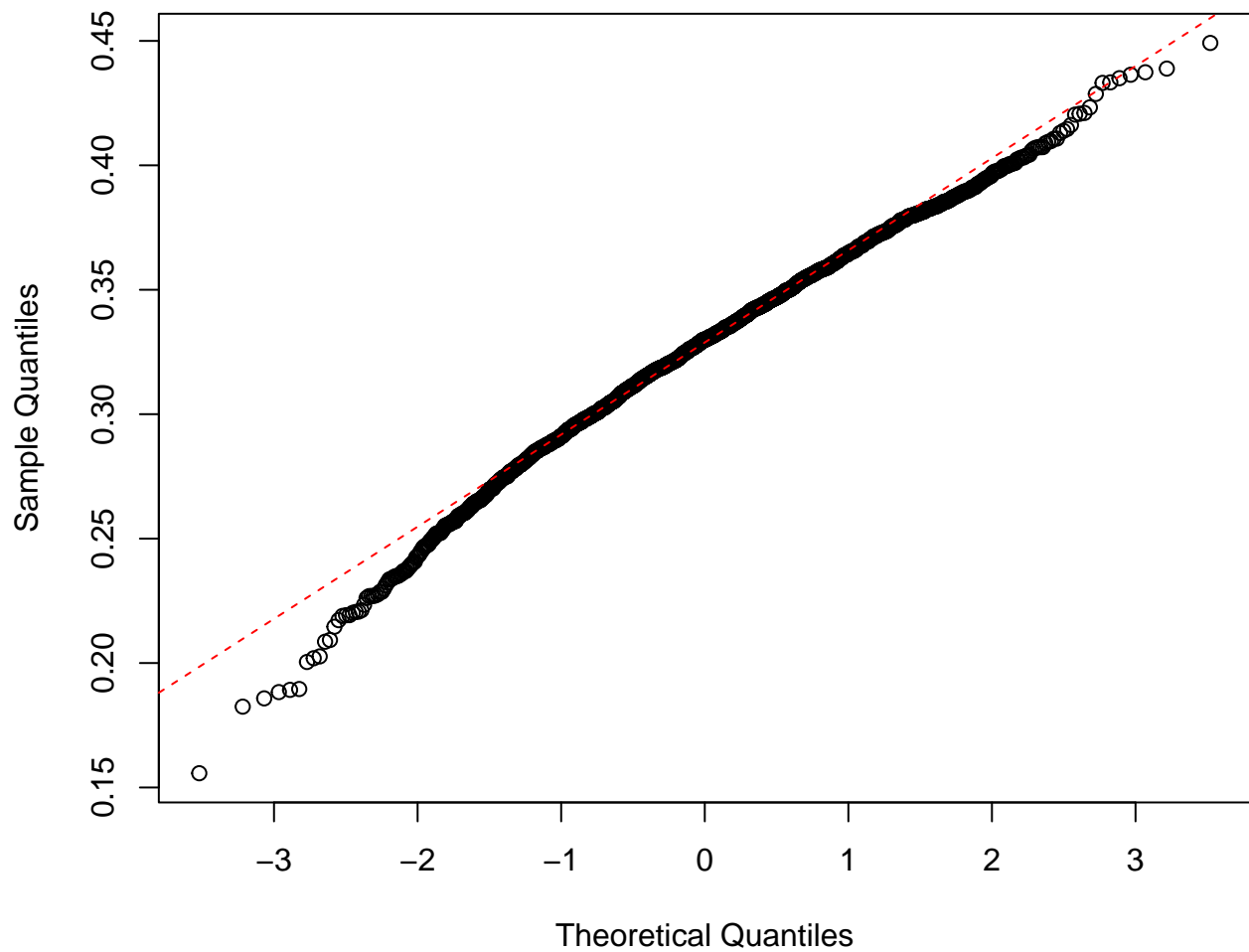
Oslo: Wet–day frequency HIST

```r
print("SSP370")
```

```
## [1] "SSP370"
```

```r
## SSP370
print(apply(coredata(allN(fw.oslo.ssp370)),2,sd,na.rm=TRUE))
```

```
##  [1] 0.03469396 0.02387360 0.02627759 0.03146228 0.03192348 0.04274472
##  [7] 0.03118964 0.03174998 0.03519799 0.03646780 0.03023387 0.03532079
## [13] 0.02781565 0.03237995 0.02916607 0.03269944 0.02768156 0.03161069
## [19] 0.04391644 0.03864741 0.04346389 0.02963541 0.03030145 0.02474196
## [25] 0.04558090 0.02900068 0.03414044
```

```r
## One outlier
coredata(fw.oslo.ssp370)[coredata(fw.oslo.ssp370) > 1] <- NA
qqnorm(coredata(allN(fw.oslo.ssp370)),main='Oslo: Wet-day frequency SSP370')
qqline(coredata(allN(fw.oslo.ssp370)),lty=2,col='red')
```

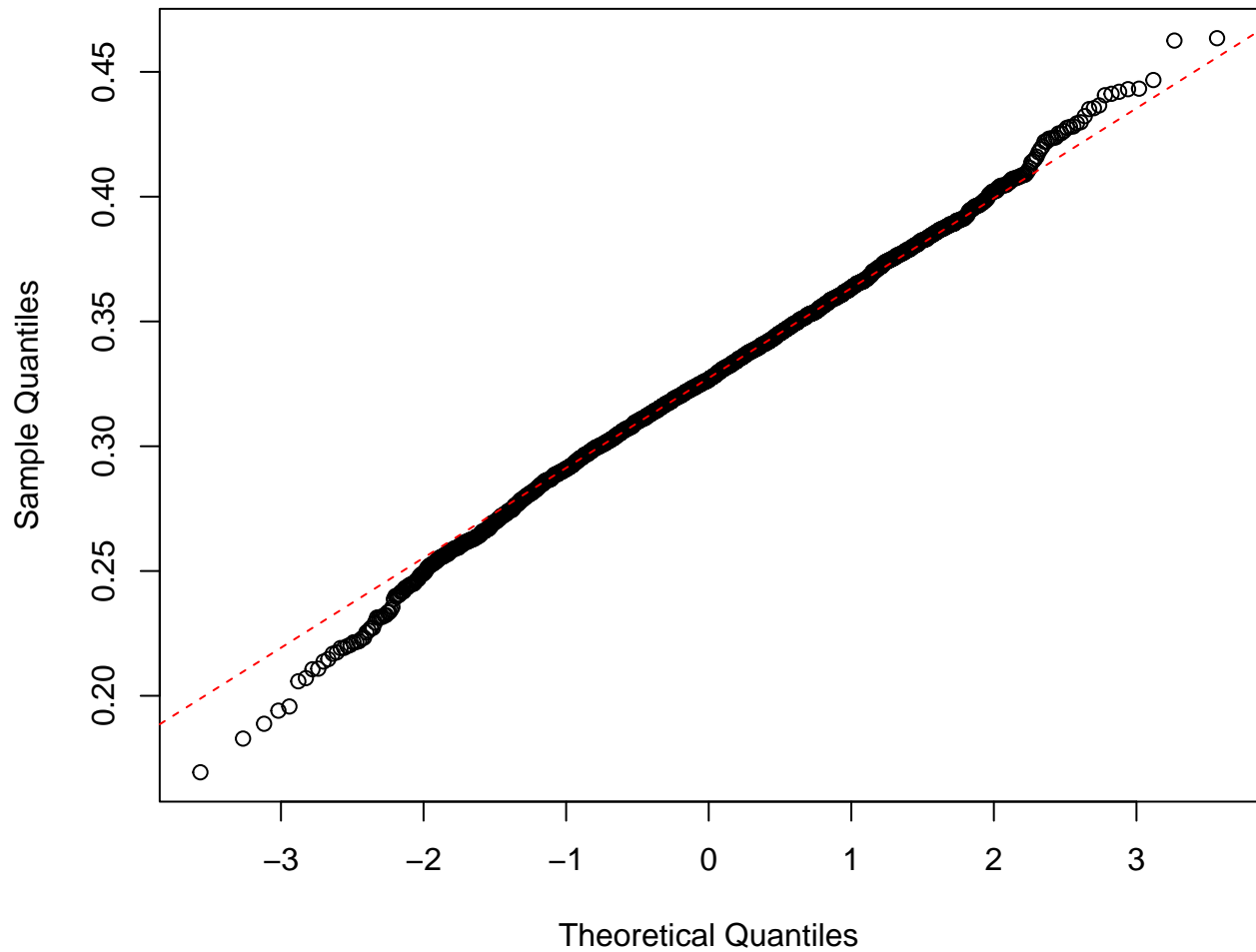
Oslo: Wet−day frequency SSP370

```
print("SSP245")
```

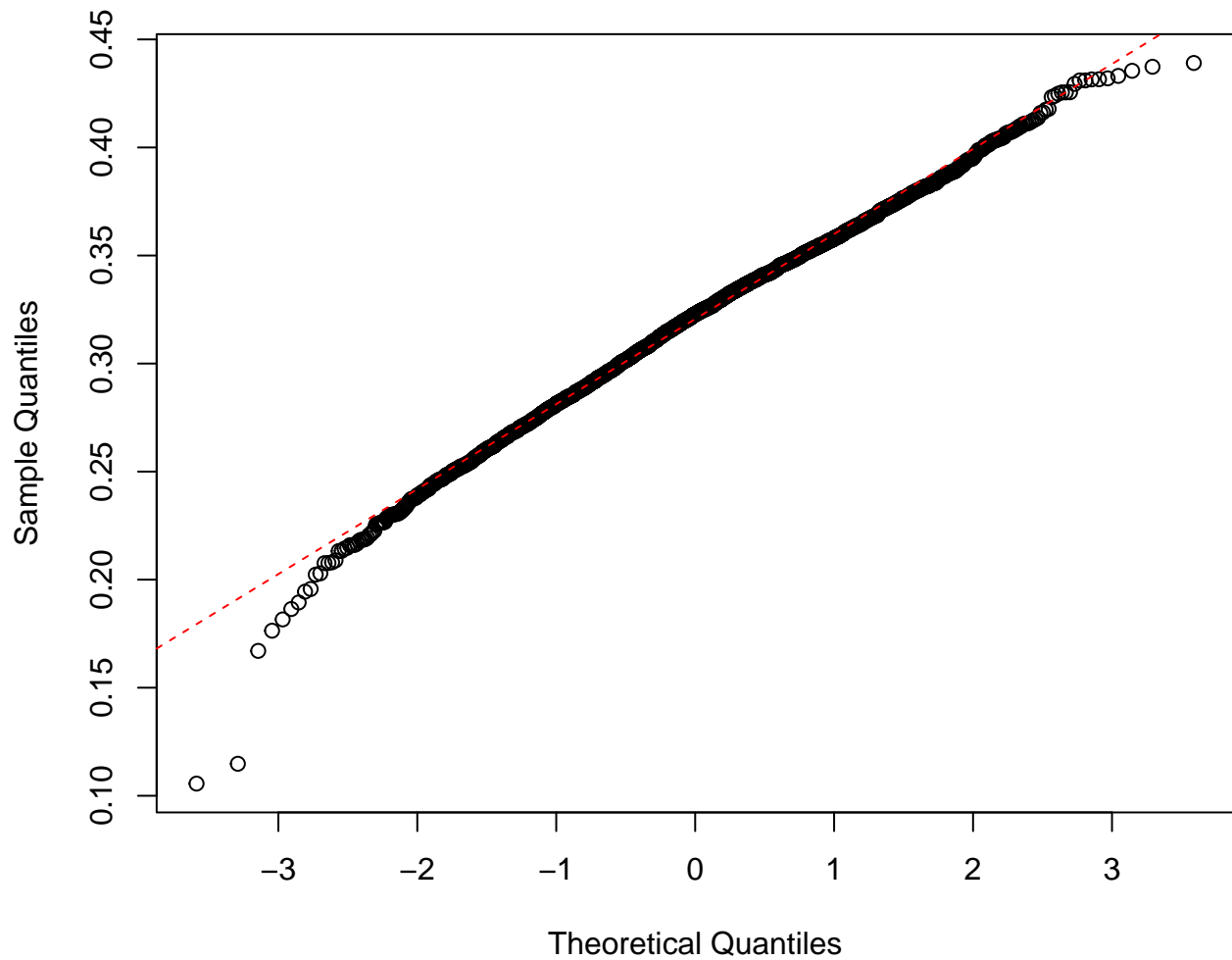
```
## [1] "SSP245"
```

```
## The SSP245
qqnorm(coredata(allN(fw.oslo.ssp245)),main='Oslo: Wet-day frequency SSP245')
qqline(coredata(allN(fw.oslo.ssp245)),lty=2,col='red')
```

Oslo: Wet−day frequency SSP245

```r
print("SSP585")
```

```
## [1] "SSP585"
```

```r
## The SSP585
qqnorm(coredata(allN(fw.oslo.ssp585)),main='Oslo: Wet-day frequency SSP585')
qqline(coredata(allN(fw.oslo.ssp585)),lty=2,col='red')
```

Oslo: Wet-day frequency SSP585

The ensembles for both f_w were close to being normally distributed for the bulk of the members.

```
## HIST
print("HIST")
```

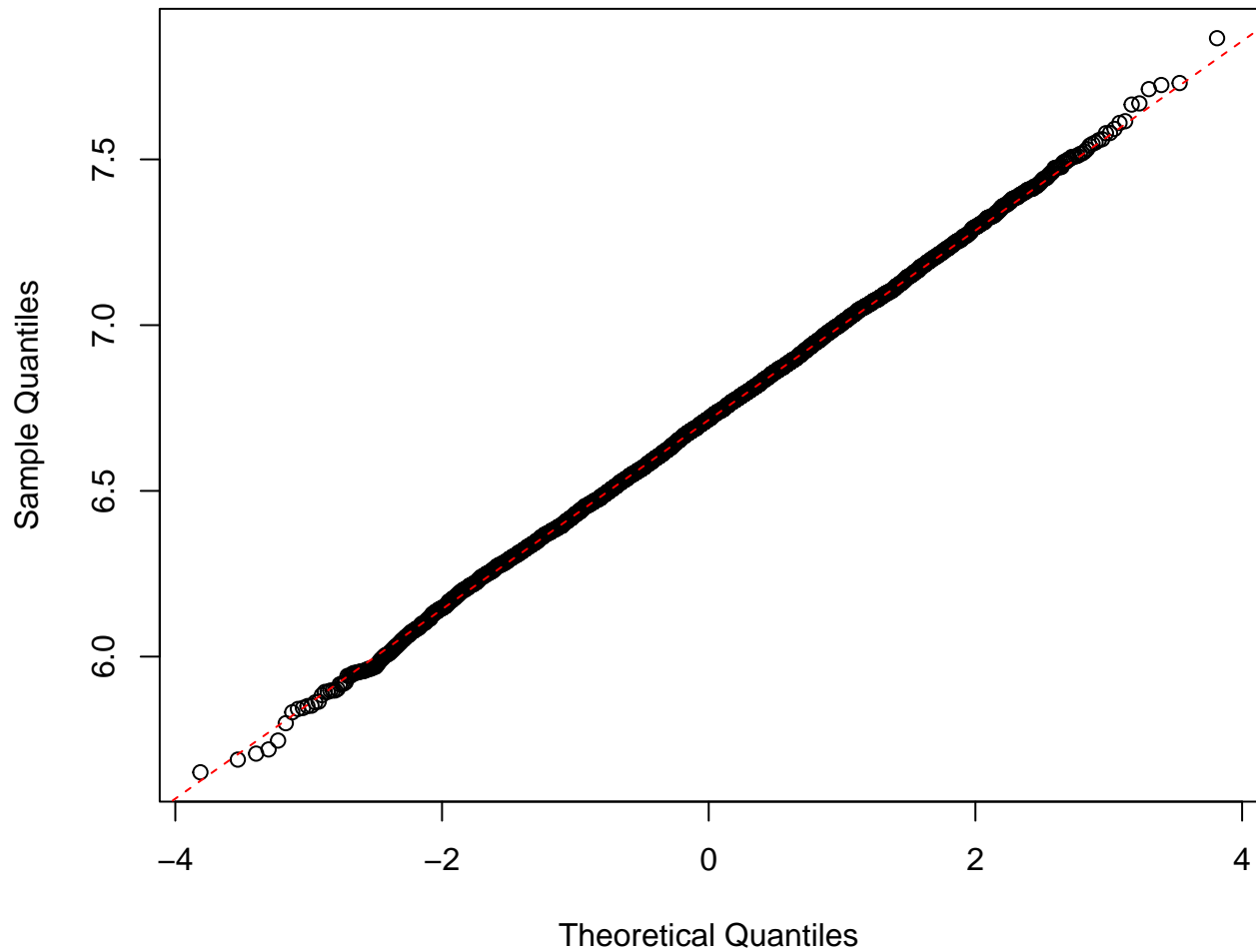
```
## [1] "HIST"
```

```
qqnorm(coredata(allN(mu.oslo.hist)),main='Oslo: Wet-day mean precipitation HIST')
qqline(coredata(allN(mu.oslo.hist)),lty=2,col='red')
```

Oslo: Wet−day mean precipitation HIST

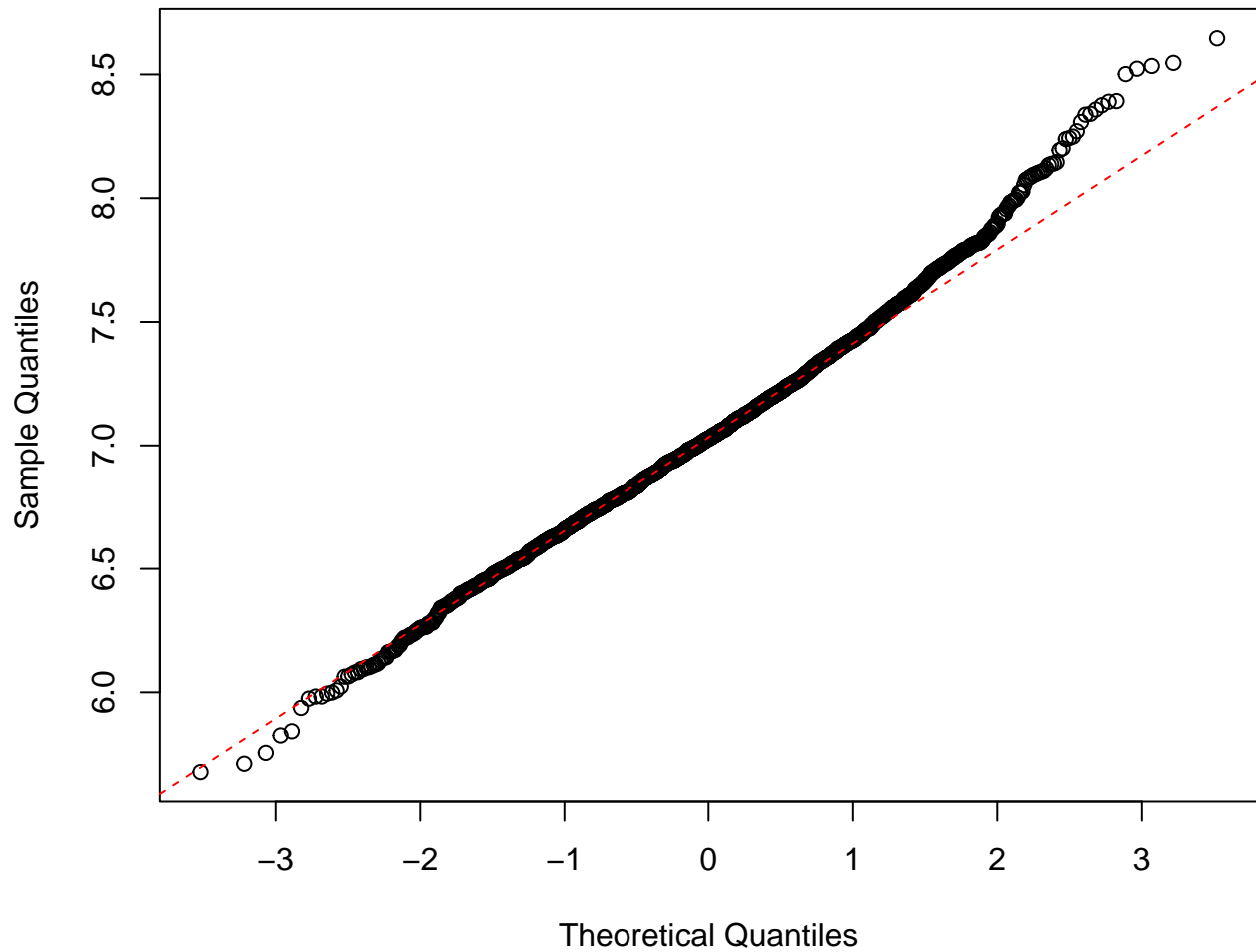
Theoretical Quantiles

```r
print("SSP370")
```

```
## [1] "SSP370"
```

```r
## SSP370
print(apply(coredata(allN(mu.oslo.ssp370)),2,sd,na.rm=TRUE))
```

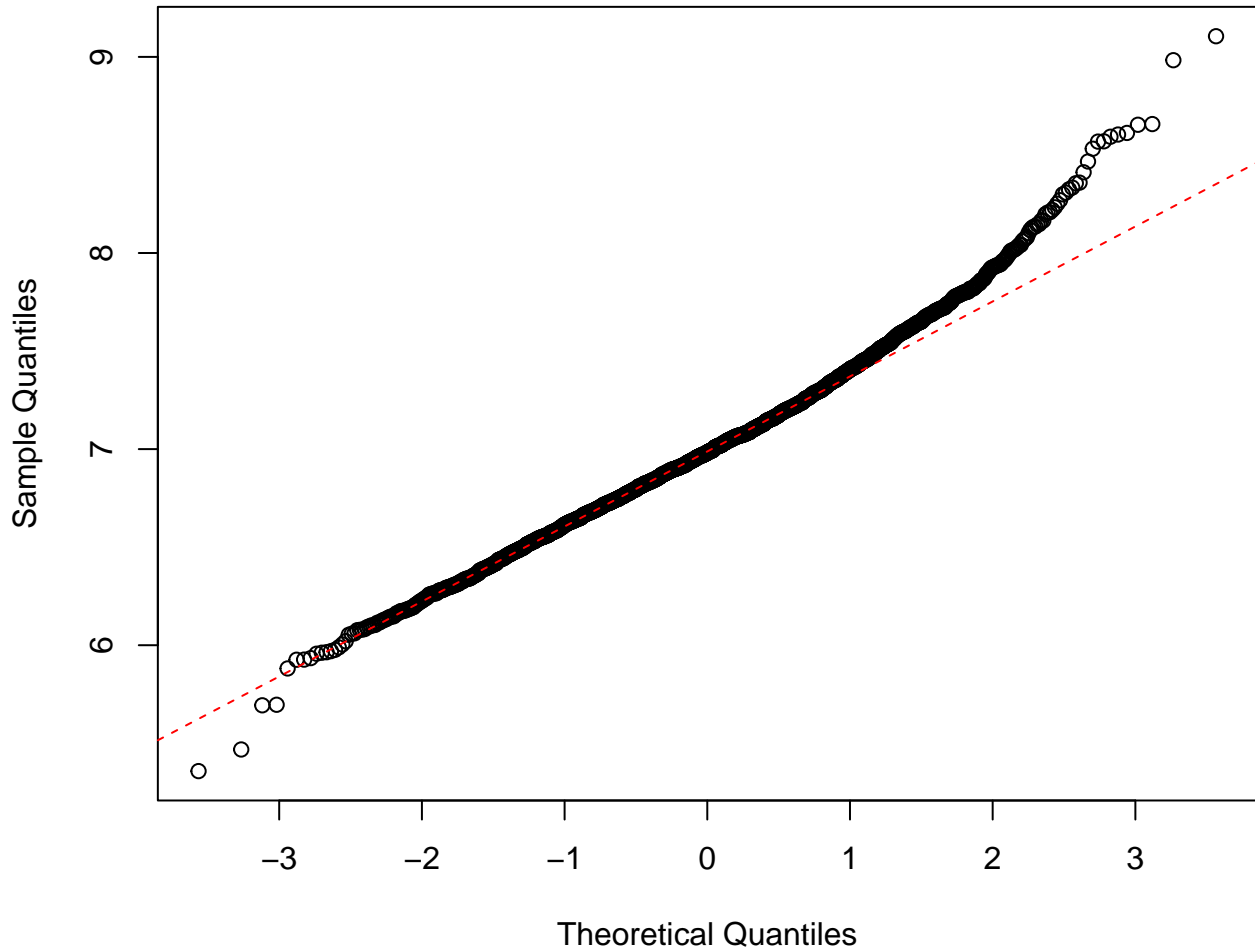
```
##  [1] 0.3996081 0.3643308 0.3768630 0.3343901 0.4029313 0.3483477 0.4361300
##  [8] 0.3142895 0.5224117 0.4602720 0.2854689 0.3727871 0.4846129 0.4368130
## [15] 0.2293145 0.3063472 0.3646238 0.3751429 0.3699007 0.3579772 0.5888074
## [22] 0.4273460 0.3234919 0.2972347 0.1923826 0.4152745 0.5493704
```

```r
qqnorm(coredata(allN(mu.oslo.ssp370)),main='Oslo: Wet-day mean precipitation SSP370')
qqline(coredata(allN(mu.oslo.ssp370)),lty=2,col='red')
```

Oslo: Wet−day mean precipitation SSP370

```
print("SSP245")
```

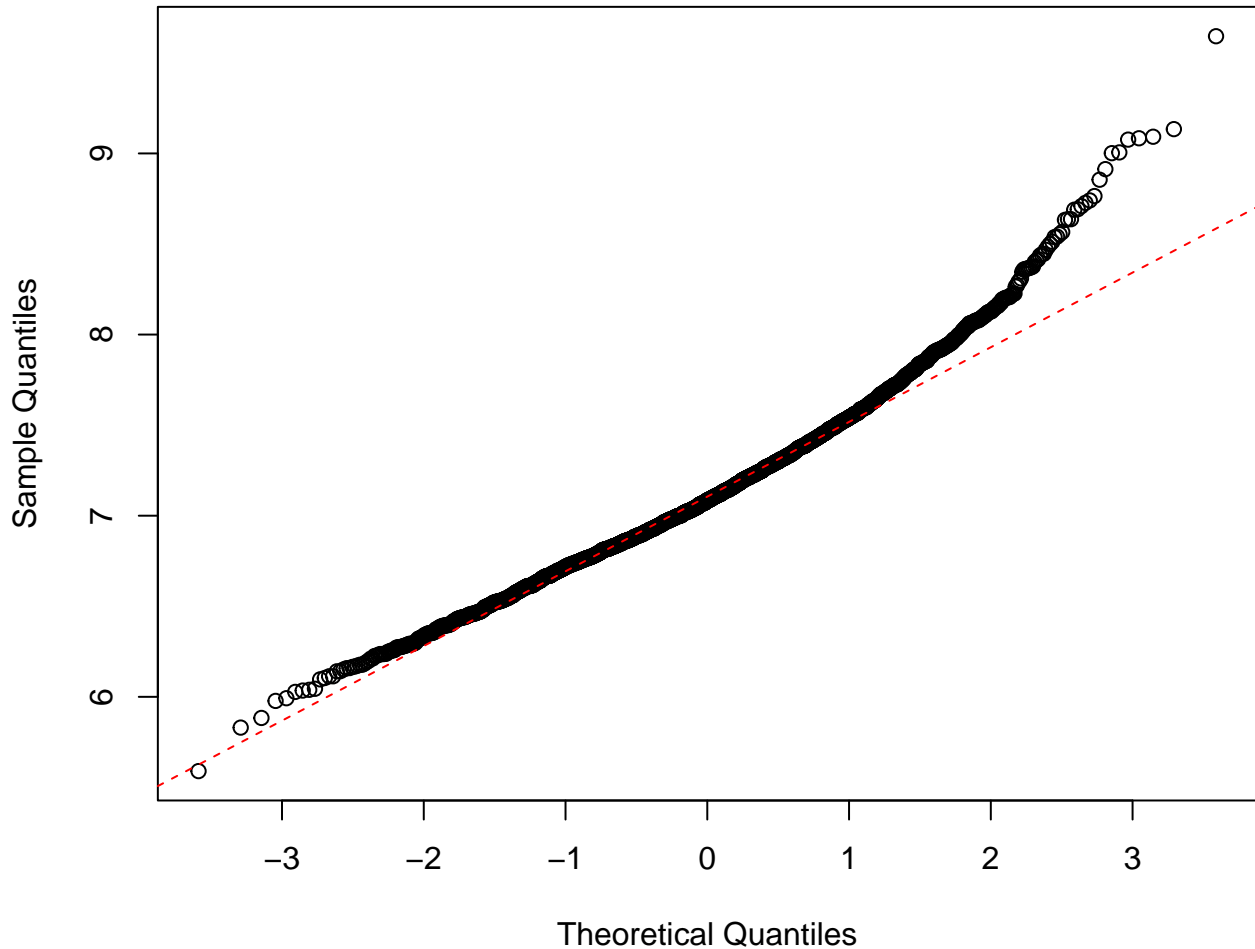
```
## [1] "SSP245"
```

```
## SSP245
qqnorm(coredata(allN(mu.oslo.ssp245)),main='Oslo: Wet-day mean precipitation SSP245')
qqline(coredata(allN(mu.oslo.ssp245)),lty=2,col='red')
```

Oslo: Wet−day mean precipitation SSP245

```
print("SSP585")
```

```
## [1] "SSP585"
```

```
## SSP585
qqnorm(coredata(allN(mu.oslo.ssp585)),main='Oslo: Wet-day mean precipitation SSP585')
qqline(coredata(allN(mu.oslo.ssp585)),lty=2,col='red')
```

Oslo: Wet−day mean precipitation SSP585

The results for μ also indicated that the distribution of ensemble members was close to a normal distribution, but there was one exception for the data points at the high end in SSP585. Hence, the ensemble results for μ were reasonably close to being normally distributed.

This means that the range of downscaled results can be approximated according to $x \sim \mathcal{N}(\mu, \sigma^2)$.

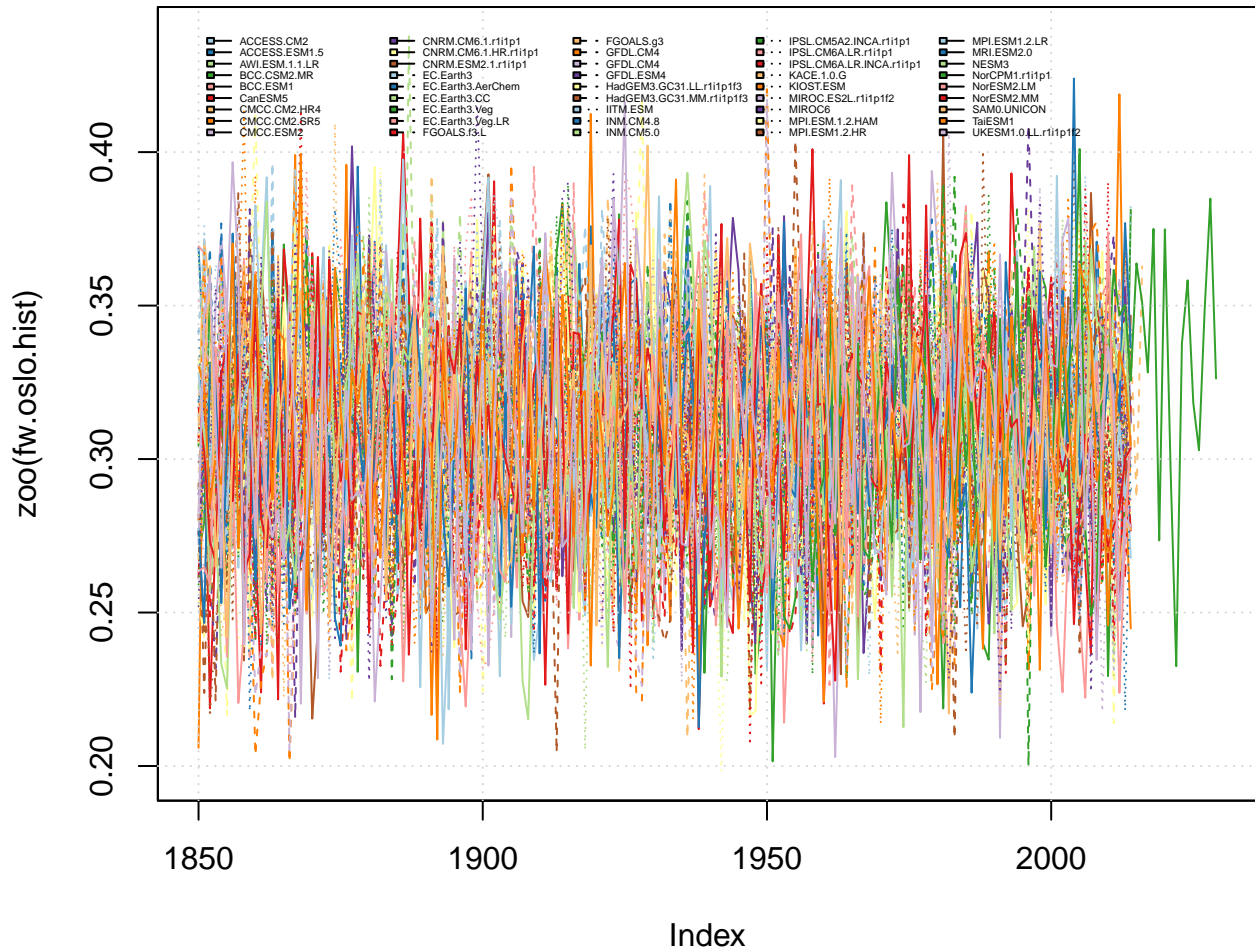
2.2.1 Individual ensemble members

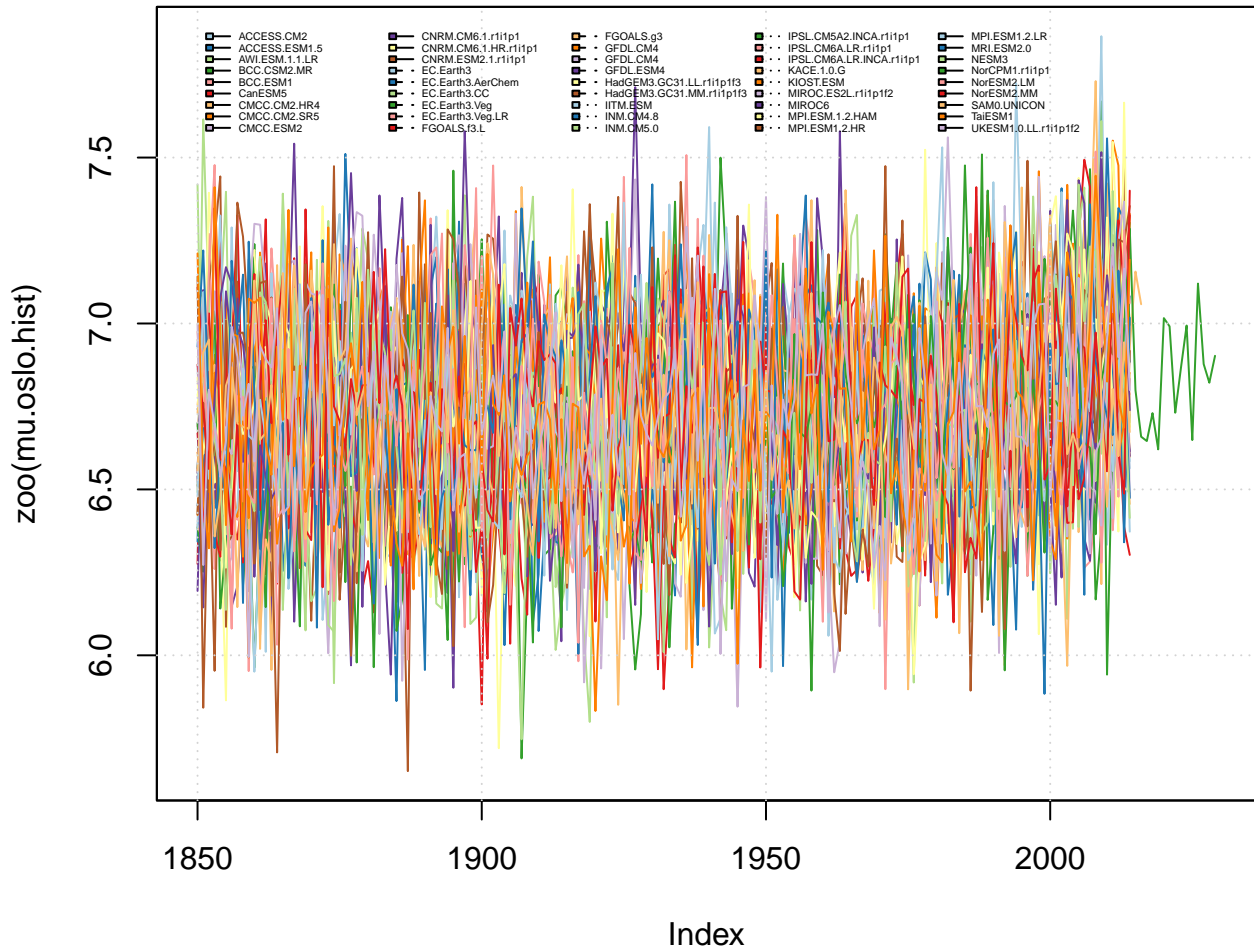
The chunk below shows the individual members of the ensemble, revealing that two members of SSP126 and the SSP585 members go all the way to year 2300 whereas most stop in 2100. There is little tendency of clustering and the results suggest that the ensemble members behave more or less in a similar way (after bad runs were precluded).

```
library(RColorBrewer)
print("HIST")
```

```
## [1] "HIST"
```

```
models <- sub('.r1i1p1f1','',attr(esd.results$fw.hist,'model_id'))
col <- brewer.pal(12, 'Paired')
lty=c(rep(1,12),rep(2,12),rep(3,12))
plot(zoo(fw.oslo.hist),col=col,plot.type='single',main='Oslo: HIST FW',lty=lty)
grid()
legend(1850,0.44,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: HIST FW

```
plot(zoo(mu.oslo.hist),col=col,plot.type='single',main='Oslo: HIST MU')
grid()
legend(1850,7.9,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: HIST MU

```
s <- round(apply(coredata(fw.oslo.hist),2,min,na.rm=TRUE),2); names(s) <- models
print(sort(s))
```

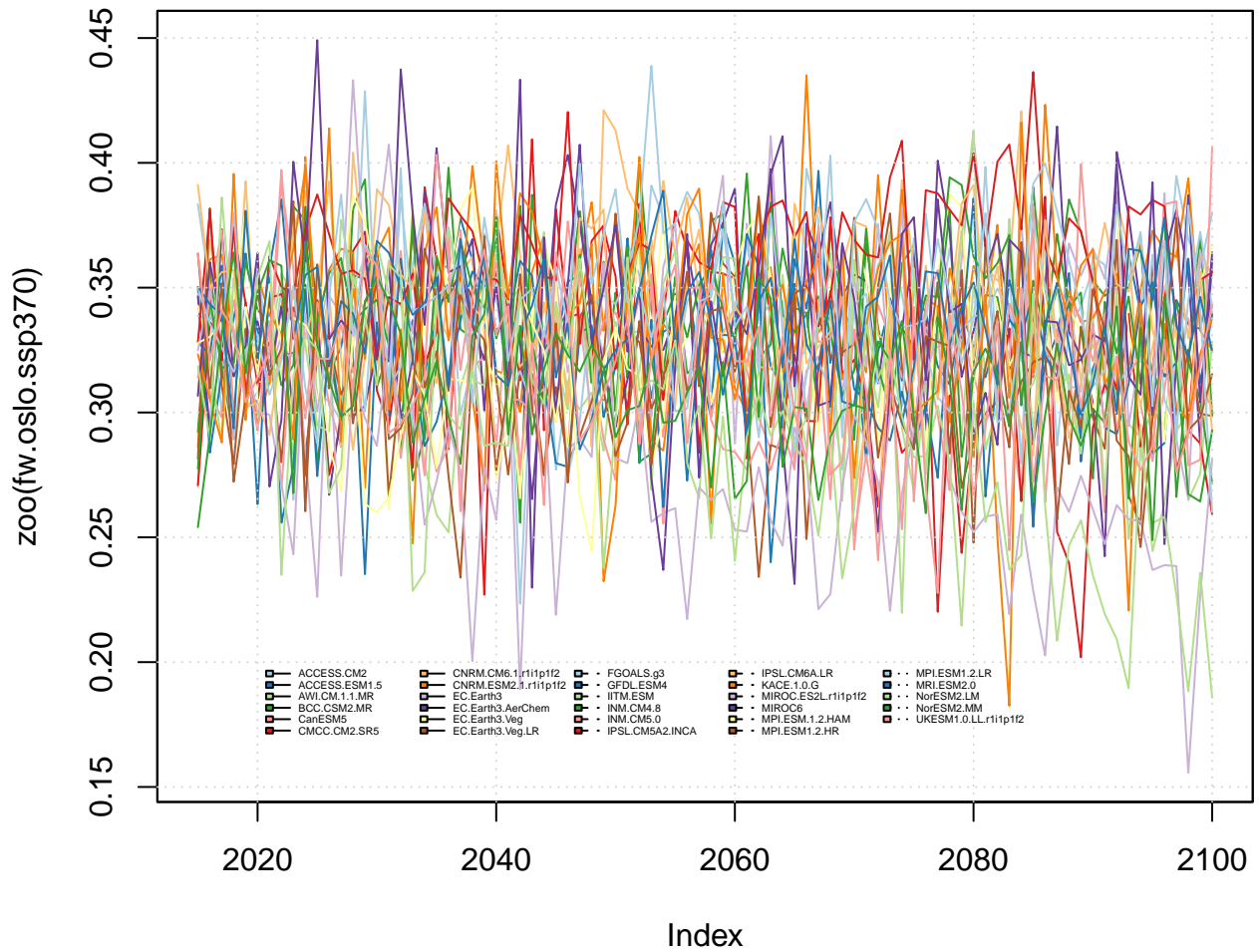
```
##              BCC.CSM2.MR                   CMCC.ESM2             EC.Earth3.Veg
##                     0.20                       0.20                      0.20
##                  GFDL.CM4                   GFDL.CM4 HadGEM3.GC31.MM.r1i1p1f3
##                     0.20                       0.20                      0.20
##           MPI.ESM.1.2.HAM                   BCC.ESM1                   CanESM5
##                     0.20                       0.21                      0.21
##             CMCC.CM2.SR5                  FGOALS.g3 HadGEM3.GC31.LL.r1i1p1f3
##                     0.21                       0.21                      0.21
##                 INM.CM5.0 IPSL.CM6A.LR.INCA.r1i1p1                 KIOST.ESM
##                     0.21                       0.21                      0.21
##              MPI.ESM1.2.LR                   MRI.ESM2.0                     NESM3
##                     0.21                       0.21                      0.21
##                   TaiESM1                 ACCESS.CM2               CMCC.CM2.HR4
##                     0.21                       0.22                      0.22
##        CNRM.CM6.1.HR.r1i1p1       CNRM.ESM2.1.r1i1p1                 GFDL.ESM4
##                     0.22                       0.22                      0.22
##                 INM.CM4.8                  KACE.1.0.G       MIROC.ES2L.r1i1p1f2
```

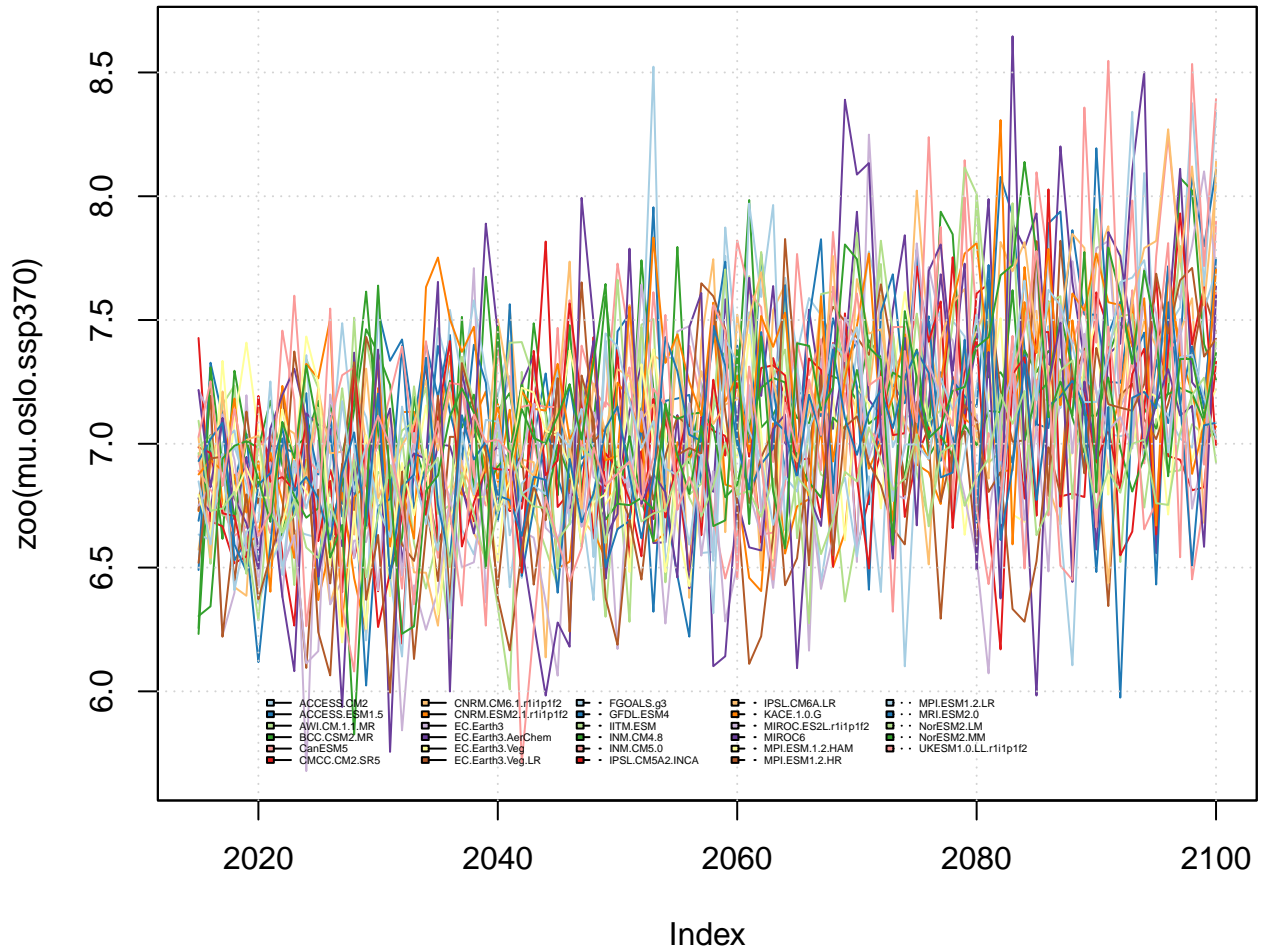
```
##                 0.22                    0.22                    0.22
##               MIROC6             MPI.ESM1.2.HR              NorESM2.LM
##                 0.22                    0.22                    0.22
##    UKESM1.0.LL.r1i1p1f2         EC.Earth3.AerChem         EC.Earth3.Veg.LR
##                 0.22                    0.23                    0.23
##           FGOALS.f3.L                 IITM.ESM     IPSL.CM5A2.INCA.r1i1p1
##                 0.23                    0.23                    0.23
##      IPSL.CM6A.LR.r1i1p1            NorCPM1.r1i1p1              NorESM2.MM
##                 0.23                    0.23                    0.23
##          ACCESS.ESM1.5            AWI.ESM.1.1.LR         CNRM.CM6.1.r1i1p1
##                 0.24                    0.24                    0.24
##              EC.Earth3             EC.Earth3.CC              SAMO.UNICON
##                 0.24                    0.24                    0.24
```

```r
print("SSP370")
```

```
## [1] "SSP370"
```

```r
models <- sub('.r1i1p1f1','',attr(esd.results$fw.ssp370,'model_id'))
plot(zoo(fw.oslo.ssp370),col=col,plot.type='single',main='Oslo: ssp370 FW')
grid()
legend(2020,0.2,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: ssp370 FW

```
plot(zoo(mu.oslo.ssp370),col=col,plot.type='single',main='Oslo: ssp370 MU')
grid()
models <- sub('.r1i1p1f1','',attr(esd.results$mu.ssp370,'model_id'))
legend(2020,6,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: ssp370 MU

```
s <- round(apply(coredata(mu.oslo.ssp370),2,max,na.rm=TRUE),2); names(s) <- models
print(sort(s))
```

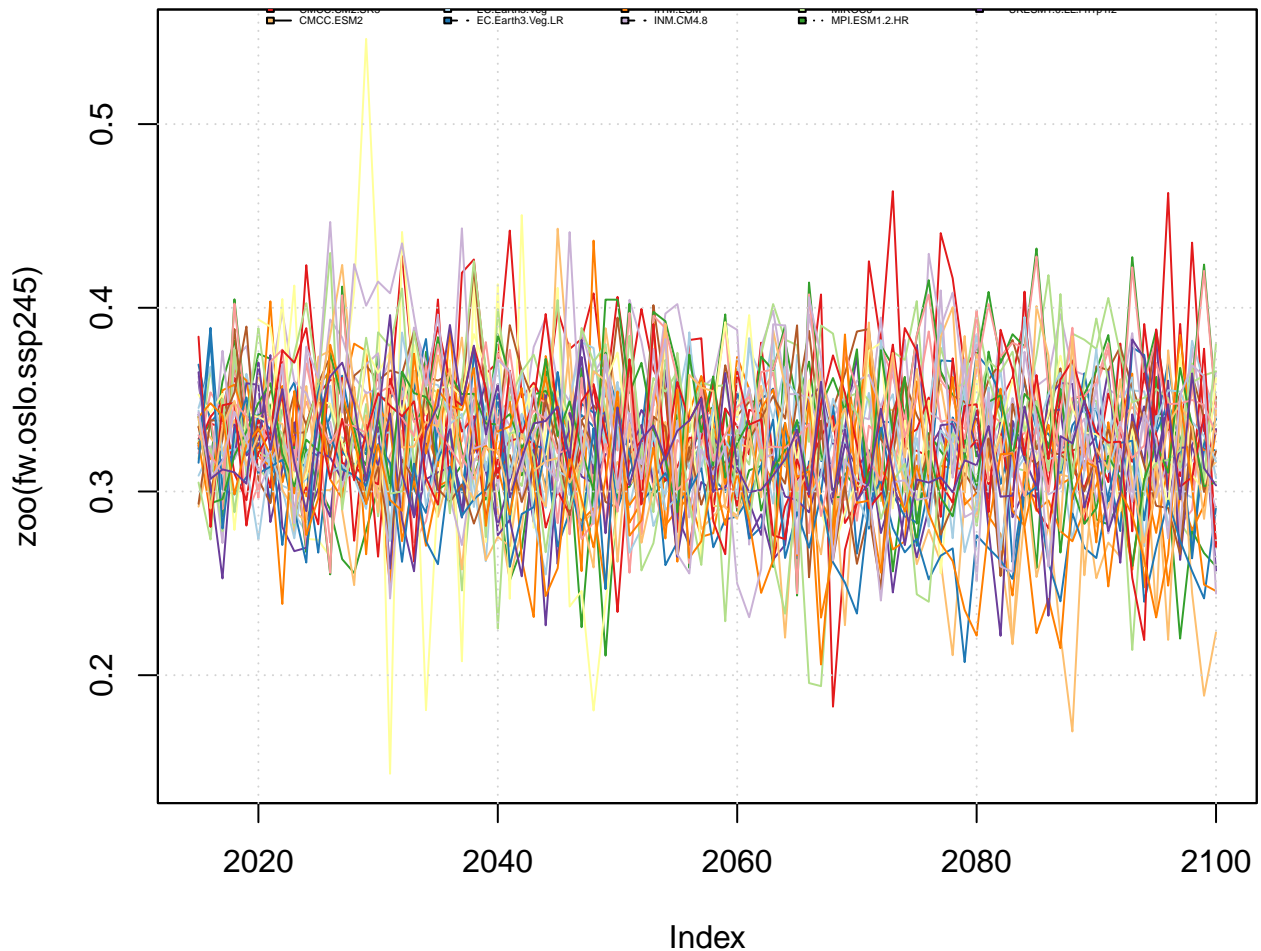
```
##          NorESM2.LM    MPI.ESM.1.2.HAM             INM.CM4.8
##                7.34              7.43                  7.62
## CNRM.ESM2.1.r1i1p1f2       MRI.ESM2.0          MPI.ESM1.2.LR
##                7.71              7.72                  7.74
##           INM.CM5.0     EC.Earth3.Veg  MIROC.ES2L.r1i1p1f2
##                7.77              7.78                  7.81
##         CMCC.CM2.SR5  EC.Earth3.Veg.LR         MPI.ESM1.2.HR
##                7.82              7.82                  7.83
##        AWI.CM.1.1.MR     IPSL.CM5A2.INCA           BCC.CSM2.MR
##                8.00              8.03                  8.07
##        ACCESS.ESM1.5           IITM.ESM  CNRM.CM6.1.r1i1p1f2
##                8.11              8.12                  8.14
##          NorESM2.MM          GFDL.ESM4               CanESM5
##                8.14              8.19                  8.24
##           EC.Earth3        IPSL.CM6A.LR             KACE.1.0.G
##                8.25              8.27                  8.31
##          ACCESS.CM2  EC.Earth3.AerChem             FGOALS.g3
```

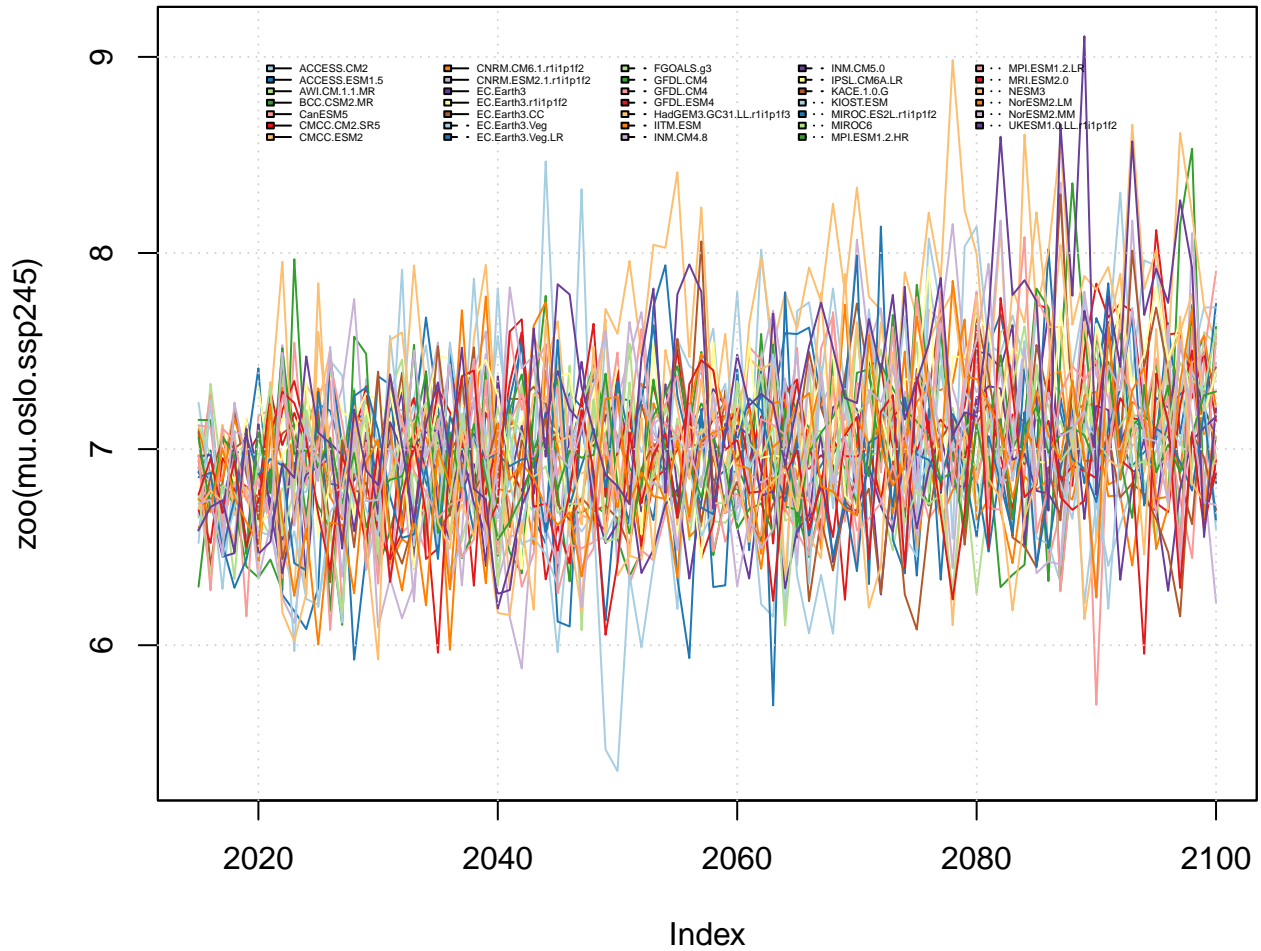
```
##                      8.38                   8.50                      8.52
## UKESM1.0.LL.r1i1p1f2                  MIROC6
##                      8.55                   8.65
```

```
print("SSP245")
```

```
## [1] "SSP245"
```

```
models <- sub('.r1i1p1f1','',attr(esd.results$fw.ssp245,'model_id'))
plot(zoo(fw.oslo.ssp245),col=col,plot.type='single',main='Oslo: ssp245 FW')
grid()
legend(2020,0.6,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: ssp245 FW

```
plot(zoo(mu.oslo.ssp245),col=col,plot.type='single',main='Oslo: ssp245 MU')
grid()
models <- sub('.r1i1p1f1','',attr(esd.results$mu.ssp245,'model_id'))
legend(2020,9,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

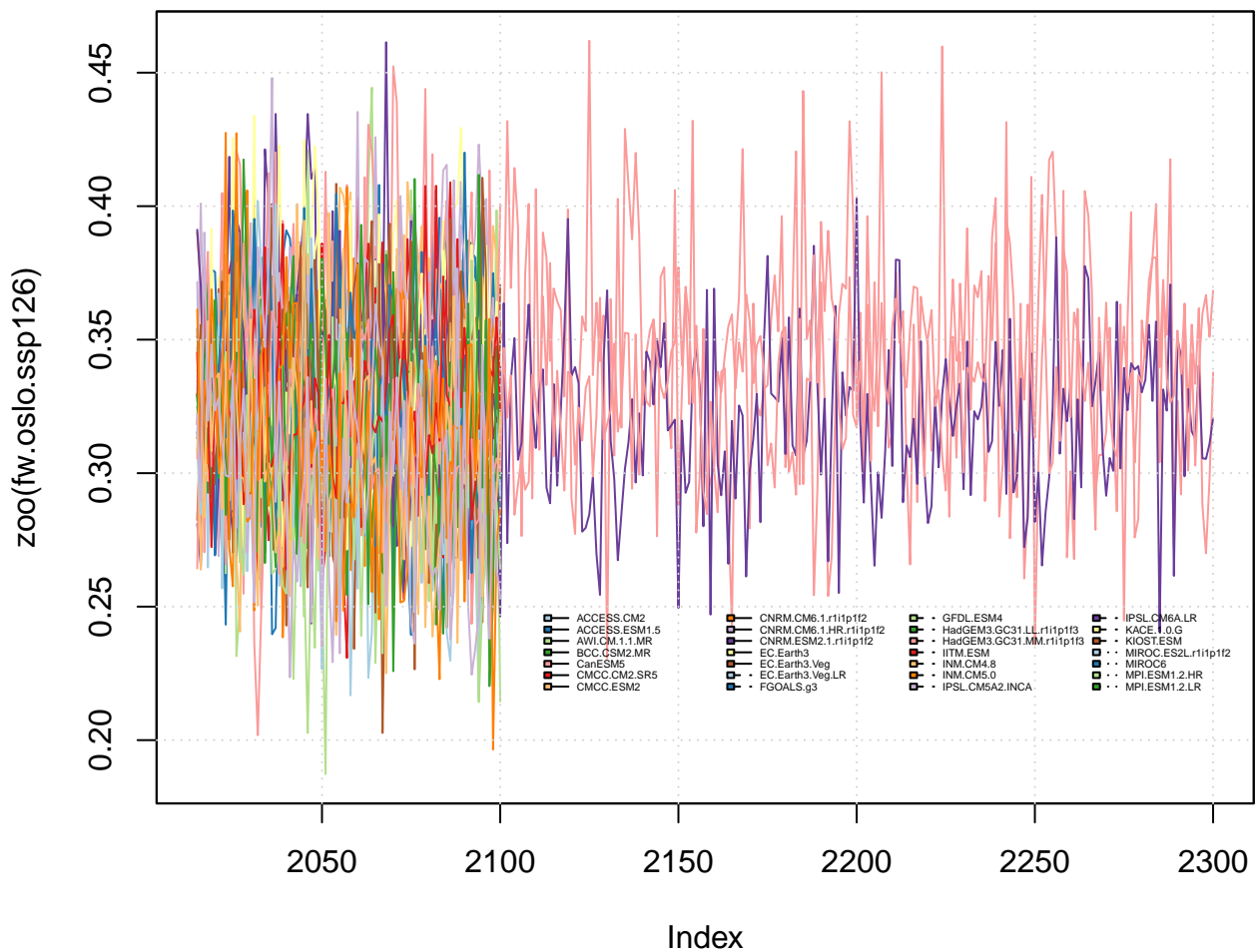
Oslo: ssp245 MU

```
s <- round(apply(coredata(mu.oslo.ssp245),2,min,na.rm=TRUE),2); names(s) <- models
print(sort(s))
```

```
##            EC.Earth3.Veg         EC.Earth3.Veg.LR                  CanESM5
##                     5.36                     5.69                     5.70
##                INM.CM4.8                    NESM3                MRI.ESM2.0
##                     5.88                     5.93                     5.96
##               KIOST.ESM       CNRM.CM6.1.r1i1p1f2               NorESM2.LM
##                     5.97                     5.98                     6.00
##            AWI.CM.1.1.MR                 GFDL.CM4                KACE.1.0.G
##                     6.08                     6.08                     6.08
##             BCC.CSM2.MR HadGEM3.GC31.LL.r1i1p1f3                   MIROC6
##                     6.10                     6.10                     6.10
##              ACCESS.CM2                 GFDL.CM4     UKESM1.0.LL.r1i1p1f2
##                     6.13                     6.17                     6.19
##       CNRM.ESM2.1.r1i1p1f2              CMCC.CM2.SR5                EC.Earth3
##                     6.22                     6.23                     6.26
##                INM.CM5.0             MPI.ESM1.2.LR                CMCC.ESM2
##                     6.26                     6.26                     6.27
##                 IITM.ESM               NorESM2.MM                GFDL.ESM4
```

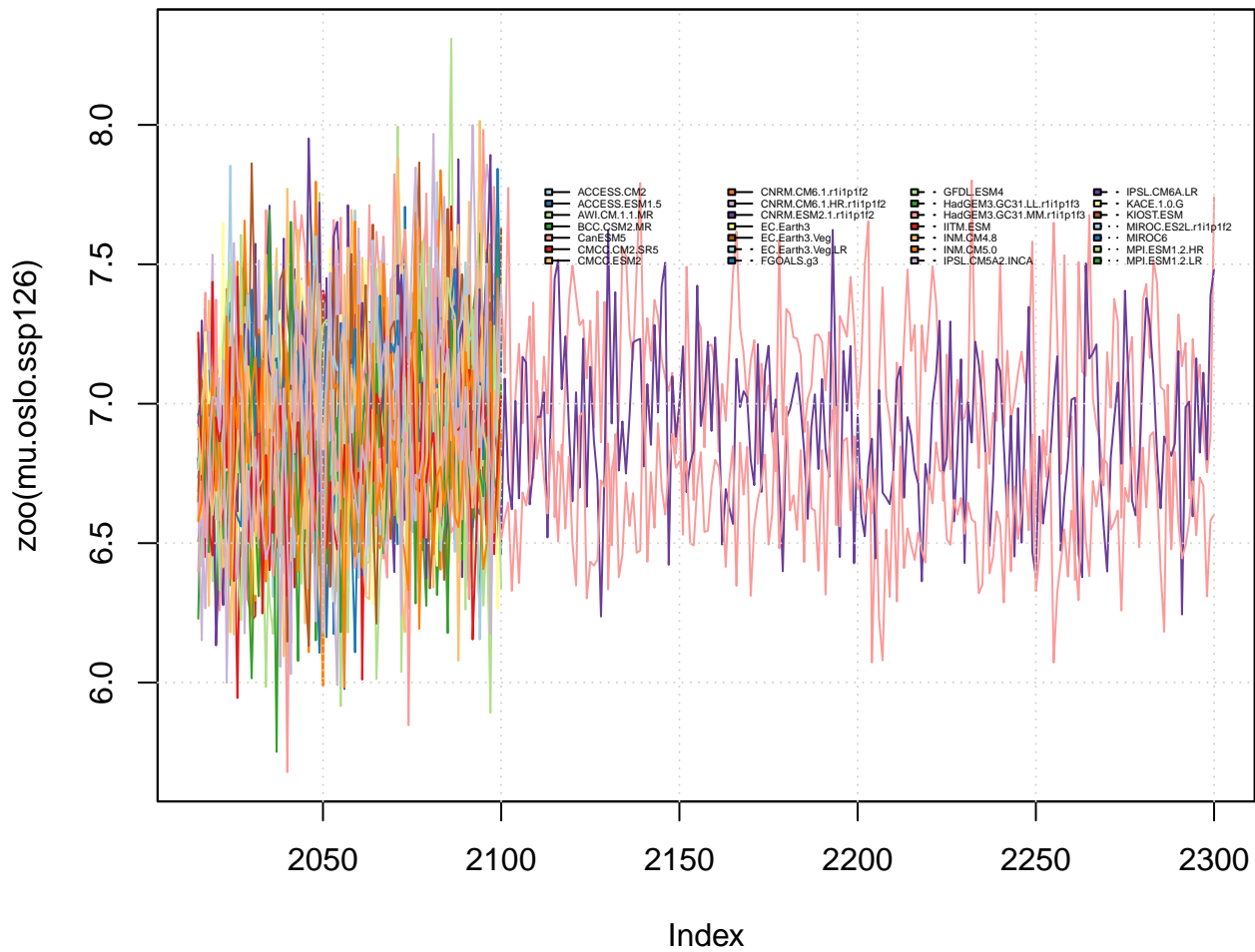
```
##                6.28                  6.30                  6.34
##       ACCESS.ESM1.5   MIROC.ES2L.r1i1p1f2         IPSL.CM6A.LR
##                6.35                  6.38                  6.39
##        EC.Earth3.CC            FGOALS.g3          MPI.ESM1.2.HR
##                6.40                  6.46                  6.46
##  EC.Earth3.r1i1p1f2
##                6.88
```

```r
print("SSP126")
```

```
## [1] "SSP126"
```

```r
models <- sub('.r1i1p1f1','',attr(esd.results$fw.ssp126,'model_id'))
plot(zoo(fw.oslo.ssp126),col=col,plot.type='single',main='Oslo: SSP126 FW')
grid()
legend(2110,0.25,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

```r
plot(zoo(mu.oslo.ssp126),col=col,plot.type='single',main='Oslo: SSP126 MU')
grid()
```

```
models <- sub('.r1i1p1f1','',attr(esd.results$mu.ssp126,'model_id'))
legend(2110,7.8,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: SSP126 MU

```
s <- round(apply(coredata(fw.oslo.ssp245),2,max,na.rm=TRUE),2); names(s) <- models
print(sort(s))
```

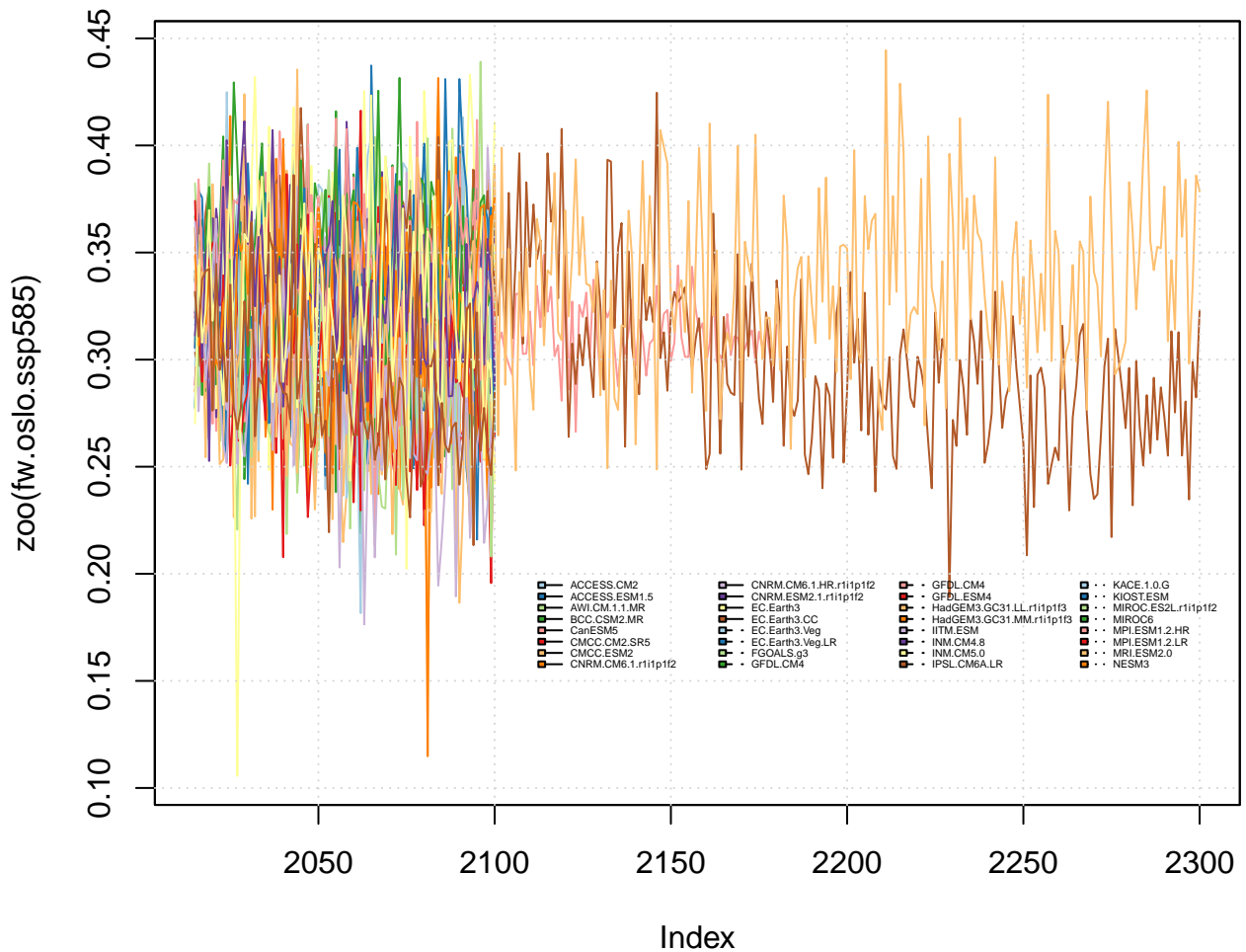
##	AWI.CM.1.1.MR	BCC.CSM2.MR	CanESM5
##	0.38	0.38	0.38
##	CNRM.ESM2.1.r1i1p1f2	IPSL.CM5A2.INCA	IPSL.CM6A.LR
##	0.38	0.38	0.38
##	MIROC6	ACCESS.CM2	ACCESS.ESM1.5
##	0.38	0.39	0.39
##	CMCC.ESM2	EC.Earth3.Veg.LR	FGOALS.g3
##	0.39	0.39	0.39
##	INM.CM5.0	KIOST.ESM	NorESM2.MM
##	0.39	0.39	0.39
##	KACE.1.0.G	MIROC.ES2L.r1i1p1f2	MPI.ESM1.2.LR
##	0.40	0.40	0.40
##	MRI.ESM2.0	NESM3	<NA>

```
##                 0.40                        0.40                        0.40
##              IITM.ESM                EC.Earth3.Veg                   GFDL.ESM4
##                 0.41                        0.42                        0.42
##            NorESM2.LM    CNRM.CM6.1.HR.r1i1p1f2    HadGEM3.GC31.LL.r1i1p1f3
##                 0.42                        0.43                        0.43
## HadGEM3.GC31.MM.r1i1p1f3                MPI.ESM1.2.HR       CNRM.CM6.1.r1i1p1f2
##                 0.43                        0.43                        0.44
##              INM.CM4.8       UKESM1.0.LL.r1i1p1f2                CMCC.CM2.SR5
##                 0.44                        0.45                        0.46
##             EC.Earth3
##                 0.55
```

```r
print("SSP585")
```

```
## [1] "SSP585"
```

```r
models <- sub('.r1i1p1f1','',attr(esd.results$fw.ssp585,'model_id'))
plot(zoo(fw.oslo.ssp585),col=col,plot.type='single',main='Oslo: SSP585 FW')
grid()
legend(2110,0.2,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: SSP585 FW

```
plot(zoo(mu.oslo.ssp585),col=col,plot.type='single',main='Oslo: SSP585 MU')
grid()
models <- sub('.r1i1p1f1','',attr(esd.results$mu.ssp585,'model_id'))
legend(2110,6.5,models,col,cex=0.3,bty='n',lty=lty,ncol=5)
```

Oslo: SSP585 MU

```
s <- round(apply(coredata(mu.oslo.ssp585),2,max,na.rm=TRUE),2); names(s) <- models
print(sort(s))
```

```
##          MPI.ESM1.2.LR    CNRM.ESM2.1.r1i1p1f2              INM.CM4.8
##                   7.52                   7.67                   7.70
##            BCC.CSM2.MR              KIOST.ESM    MIROC.ES2L.r1i1p1f2
##                   7.71                   7.73                   7.81
##               IITM.ESM              INM.CM5.0             ACCESS.CM2
##                   7.82                   7.84                   7.93
##              FGOALS.g3           ACCESS.ESM1.5                TaiESM1
##                   7.93                   7.97                   8.07
##               GFDL.CM4             NorESM2.MM               GFDL.CM4
##                   8.08                   8.08                   8.09
##        EC.Earth3.Veg.LR             NorESM2.LM    CNRM.CM6.1.r1i1p1f2
```

```
##                       8.10                             8.13                             8.18
##                 CMCC.CM2.SR5                        KACE.1.0.G                            NESM3
##                       8.19                             8.21                             8.21
##                MPI.ESM1.2.HR                     EC.Earth3.CC                           MIROC6
##                       8.35                             8.37                             8.37
##                   CMCC.ESM2                         EC.Earth3                        MRI.ESM2.0
##                       8.44                             8.45                             8.62
## HadGEM3.GC31.MM.r1i1p1f3         UKESM1.0.LL.r1i1p1f2                     EC.Earth3.Veg
##                       8.69                             8.69                             8.85
## HadGEM3.GC31.LL.r1i1p1f3                       GFDL.ESM4                      AWI.CM.1.1.MR
##                       8.91                             9.08                             9.13
##                     CanESM5     CNRM.CM6.1.HR.r1i1p1f2                      IPSL.CM6A.LR
##                       9.20                             9.65                             9.82
```

```r
s <- round(apply(coredata(mu.oslo.ssp585),2,max,na.rm=TRUE),2); names(s) <- models
print(sort(s))
```

```
##                MPI.ESM1.2.LR     CNRM.ESM2.1.r1i1p1f2                        INM.CM4.8
##                       7.52                             7.67                             7.70
##                  BCC.CSM2.MR                       KIOST.ESM         MIROC.ES2L.r1i1p1f2
##                       7.71                             7.73                             7.81
##                    IITM.ESM                         INM.CM5.0                      ACCESS.CM2
##                       7.82                             7.84                             7.93
##                   FGOALS.g3                     ACCESS.ESM1.5                          TaiESM1
##                       7.93                             7.97                             8.07
##                    GFDL.CM4                       NorESM2.MM                         GFDL.CM4
##                       8.08                             8.08                             8.09
##              EC.Earth3.Veg.LR                      NorESM2.LM       CNRM.CM6.1.r1i1p1f2
##                       8.10                             8.13                             8.18
##                 CMCC.CM2.SR5                        KACE.1.0.G                            NESM3
##                       8.19                             8.21                             8.21
##                MPI.ESM1.2.HR                     EC.Earth3.CC                           MIROC6
##                       8.35                             8.37                             8.37
##                   CMCC.ESM2                         EC.Earth3                        MRI.ESM2.0
##                       8.44                             8.45                             8.62
## HadGEM3.GC31.MM.r1i1p1f3         UKESM1.0.LL.r1i1p1f2                     EC.Earth3.Veg
##                       8.69                             8.69                             8.85
## HadGEM3.GC31.LL.r1i1p1f3                       GFDL.ESM4                      AWI.CM.1.1.MR
##                       8.91                             9.08                             9.13
##                     CanESM5     CNRM.CM6.1.HR.r1i1p1f2                      IPSL.CM6A.LR
##                       9.20                             9.65                             9.82
```

The ensemble members tend to follow similar behaviour and there is no obvious clustering or outlier.

2.3 Ensemble mean and and standard deviation

The analysis below provides consise information about the ensemble in terms of the ensemble mean and standard deviation.

```r
## Limit the ensemble mean and standard deviation to when the ensemble has all
## members represented
fw.oslo.hist <- allN(fw.oslo.hist)
```

```r
FW.mean.hist <- zoo(apply(as.data.frame(fw.oslo.hist),1,FUN='mean',na.rm=TRUE),
                    order.by=year(fw.oslo.hist))
FW.std.hist <- zoo(apply(as.data.frame(fw.oslo.hist),1,FUN='sd',na.rm=TRUE),
                    order.by=year(fw.oslo.hist))
mu.oslo.hist <- allN(mu.oslo.hist)
MU.mean.hist <- zoo(apply(as.data.frame(mu.oslo.hist),1,FUN='mean',na.rm=TRUE),
                    order.by=year(mu.oslo.hist))
MU.std.hist <- zoo(apply(as.data.frame(mu.oslo.hist),1,FUN='sd',na.rm=TRUE),
                    order.by=year(mu.oslo.hist))
fw.oslo.ssp370 <- allN(fw.oslo.ssp370)
FW.mean.ssp370 <- zoo(apply(as.data.frame(fw.oslo.ssp370),1,FUN='mean',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp370))
FW.std.ssp370 <- zoo(apply(as.data.frame(fw.oslo.ssp370),1,FUN='sd',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp370))
mu.oslo.ssp370 <- allN(mu.oslo.ssp370)
MU.mean.ssp370 <- zoo(apply(as.data.frame(mu.oslo.ssp370),1,FUN='mean',na.rm=TRUE),
                      order.by=year(mu.oslo.ssp370))
MU.std.ssp370 <- zoo(apply(as.data.frame(mu.oslo.ssp370),1,FUN='sd',na.rm=TRUE),
                      order.by=year(mu.oslo.ssp370))
fw.oslo.ssp126 <- allN(fw.oslo.ssp126)
FW.mean.ssp126 <- zoo(apply(as.data.frame(fw.oslo.ssp126),1,FUN='mean',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp126))
FW.std.ssp126 <- zoo(apply(as.data.frame(fw.oslo.ssp126),1,FUN='sd',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp126))
mu.oslo.ssp126 <- allN(mu.oslo.ssp126)
MU.mean.ssp126 <- zoo(apply(as.data.frame(mu.oslo.ssp126),1,FUN='mean',na.rm=TRUE),
                      order.by=year(mu.oslo.ssp126))
MU.std.ssp126 <- zoo(apply(as.data.frame(mu.oslo.ssp126),1,FUN='sd',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp126))
fw.oslo.ssp245 <- allN(fw.oslo.ssp245)
FW.mean.ssp245 <- zoo(apply(as.data.frame(fw.oslo.ssp245),1,FUN='mean',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp245))
FW.std.ssp245 <- zoo(apply(as.data.frame(fw.oslo.ssp245),1,FUN='sd',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp245))
mu.oslo.ssp245 <- allN(mu.oslo.ssp245)
MU.mean.ssp245 <- zoo(apply(as.data.frame(mu.oslo.ssp245),1,FUN='mean',na.rm=TRUE),
                      order.by=year(mu.oslo.ssp245))
MU.std.ssp245 <- zoo(apply(as.data.frame(mu.oslo.ssp245),1,FUN='sd',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp245))
fw.oslo.ssp585 <- allN(fw.oslo.ssp585)
FW.mean.ssp585 <- zoo(apply(as.data.frame(fw.oslo.ssp585),1,FUN='mean',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp585))
FW.std.ssp585 <- zoo(apply(as.data.frame(fw.oslo.ssp585),1,FUN='sd',na.rm=TRUE),
                      order.by=year(fw.oslo.ssp585))
mu.oslo.ssp585 <- allN(mu.oslo.ssp585)
MU.mean.ssp585 <- zoo(apply(as.data.frame(mu.oslo.ssp585),1,FUN='mean',na.rm=TRUE),
                      order.by=year(mu.oslo.ssp585))
MU.std.ssp585 <- zoo(apply(as.data.frame(mu.oslo.ssp585),1,FUN='sd',na.rm=TRUE),
                      order.by=year(mu.oslo.ssp585))
```

Plot time series of the ensemble mean and standard deviation to get a visual quality check:

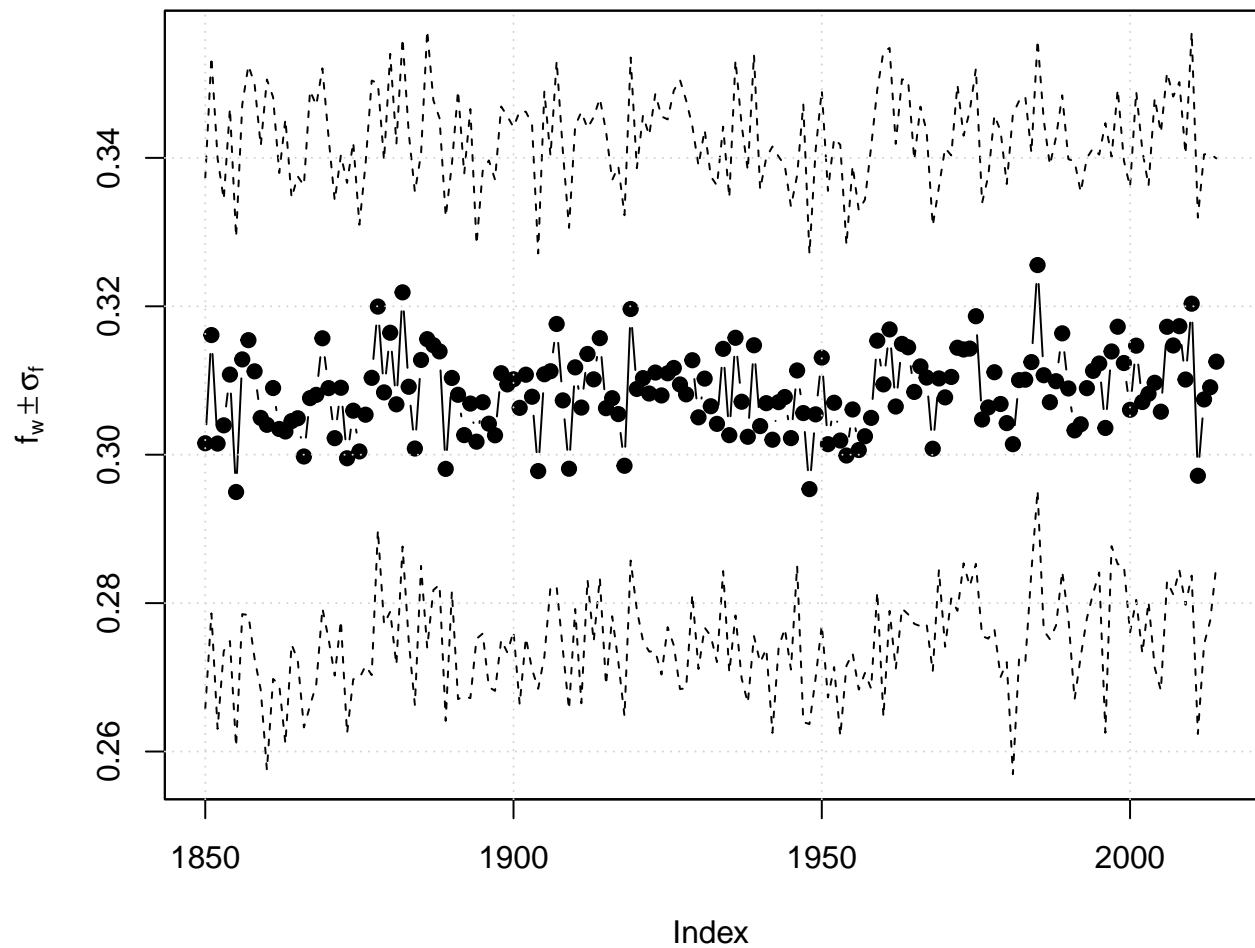
```
print("HIST FW")
```

[1] "HIST FW"

```
plot(FW.mean.hist,type='b',pch=19,main='Oslo: Wet-day frequency - HIST',
     ylab=expression(f[w]%+-%sigma[f]),
     ylim=range(FW.mean.hist + c(-1,1)*FW.std.hist,na.rm=TRUE))
```

Warning in '*.default'(c(-1, 1), FW.std.hist): longer object length is not a
multiple of shorter object length

```
lines(FW.mean.hist - FW.std.hist,lty=2)
lines(FW.mean.hist + FW.std.hist,lty=2)
grid()
```

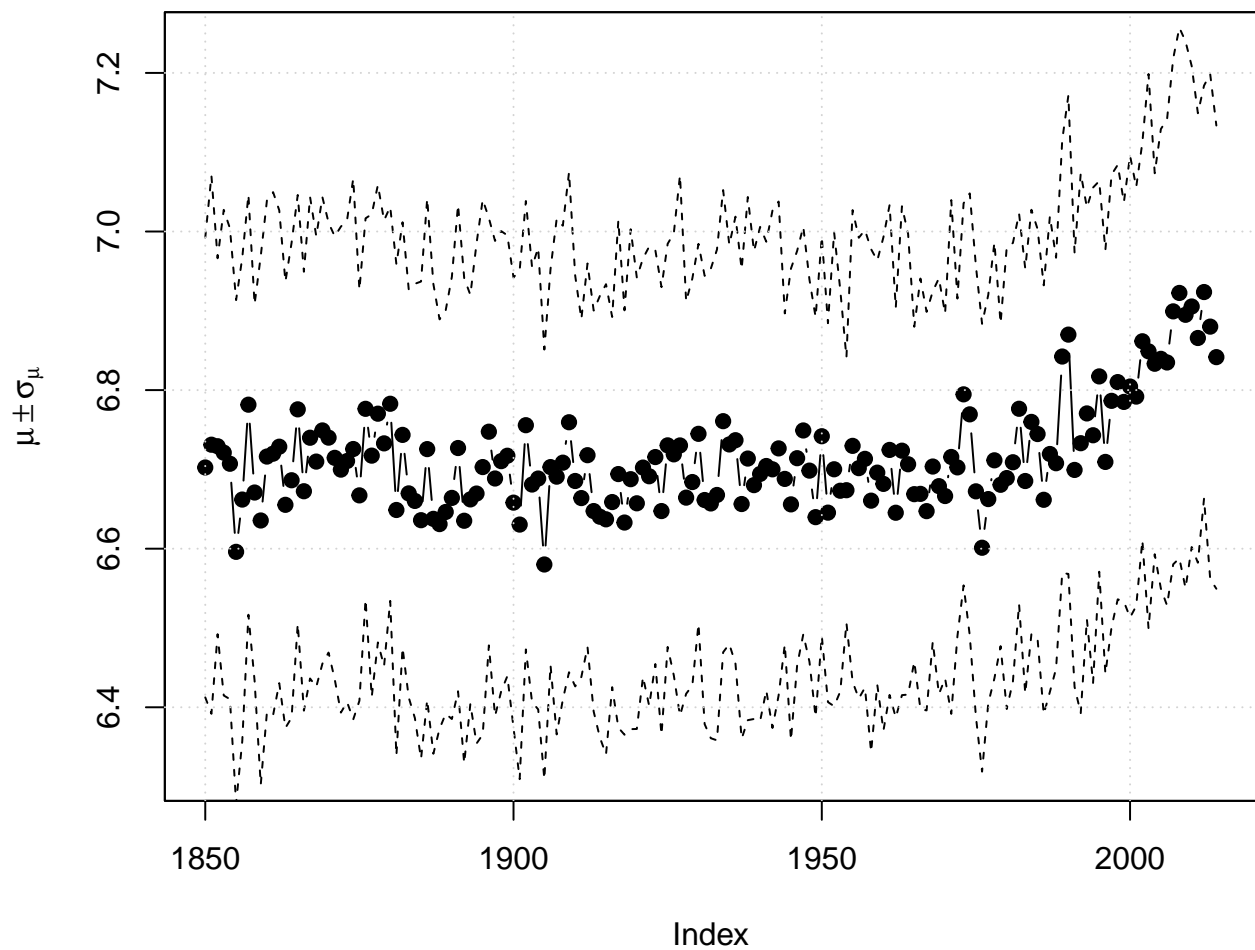
Oslo: Wet–day frequency – HIST

```
print("HIST MU")
```

[1] "HIST MU"

```
plot(MU.mean.hist,type='b',pch=19,main='Oslo: Wet-day intensity - HIST',
     ylab=expression(mu%+-%sigma[mu]),
     ylim=range(MU.mean.hist + c(-1,1)*MU.std.hist,na.rm=TRUE))
```

```
## Warning in '*.default'(c(-1, 1), MU.std.hist): longer object length is not a
## multiple of shorter object length
```

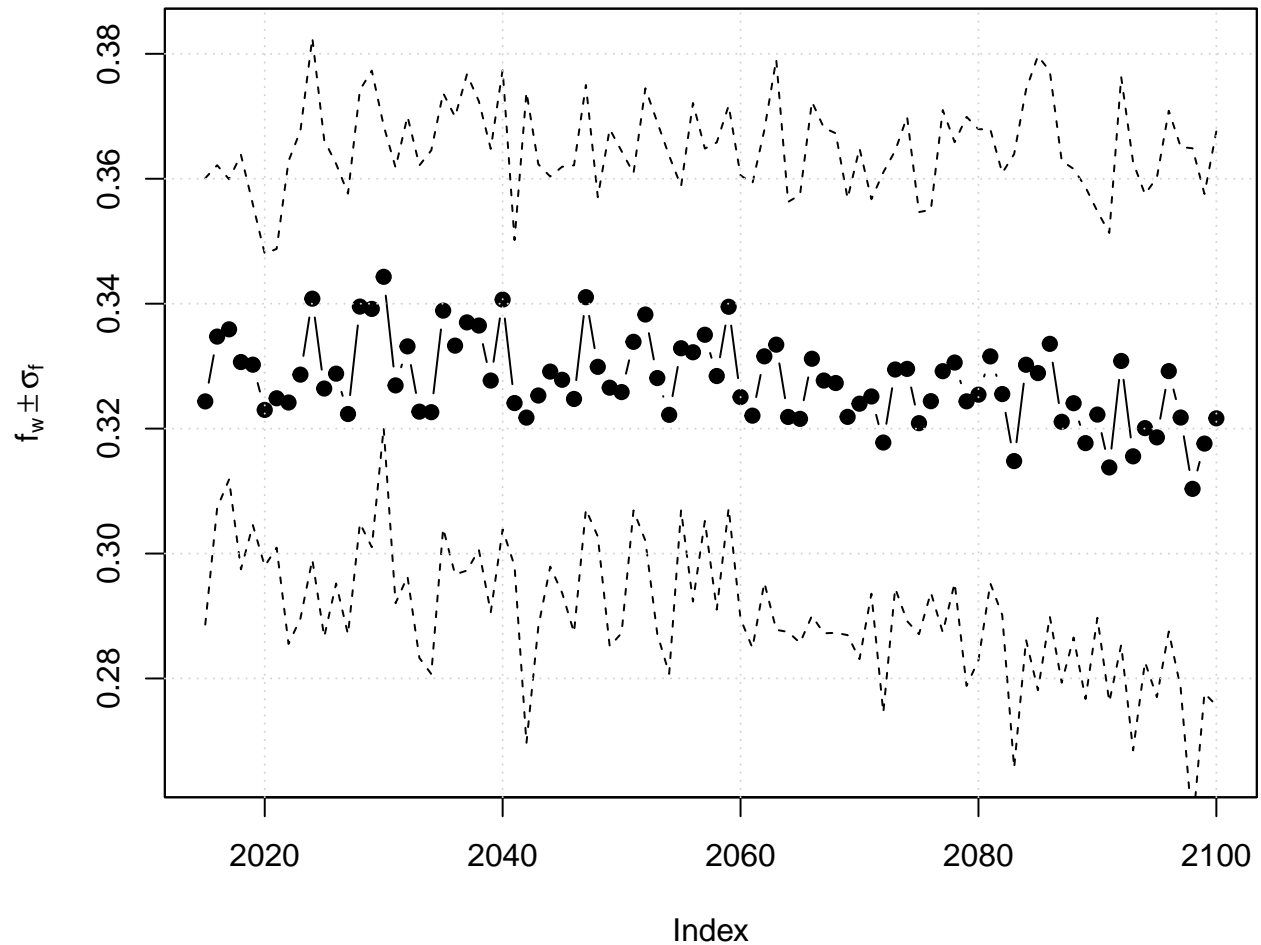
```
lines(MU.mean.hist - MU.std.hist,lty=2)
lines(MU.mean.hist + MU.std.hist,lty=2)
grid()
```

Oslo: Wet–day intensity – HIST

```
print("SSP370 FW")
```

```
## [1] "SSP370 FW"
```

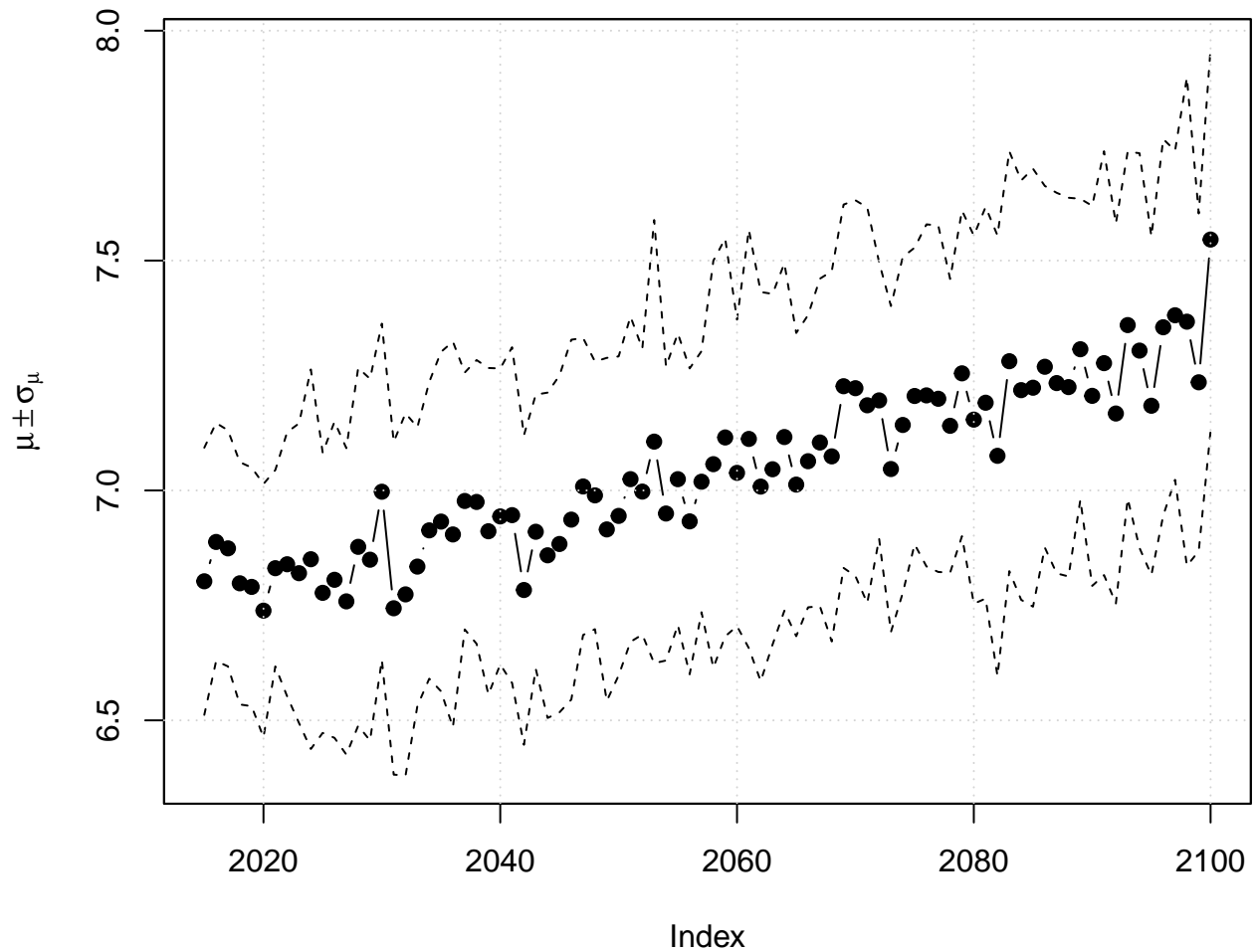
```
plot(FW.mean.ssp370,type='b',pch=19,main='Oslo: Wet-day frequency - SSP370',
     ylab=expression(f[w]%+-%sigma[f]),
     ylim=range(FW.mean.ssp370 + c(-1,1)*FW.std.ssp370,na.rm=TRUE))
```

```
lines(FW.mean.ssp370 - FW.std.ssp370,lty=2)
lines(FW.mean.ssp370 + FW.std.ssp370,lty=2)
grid()
```

Oslo: Wet–day frequency – SSP370

```
print("SSP370 MU")
```

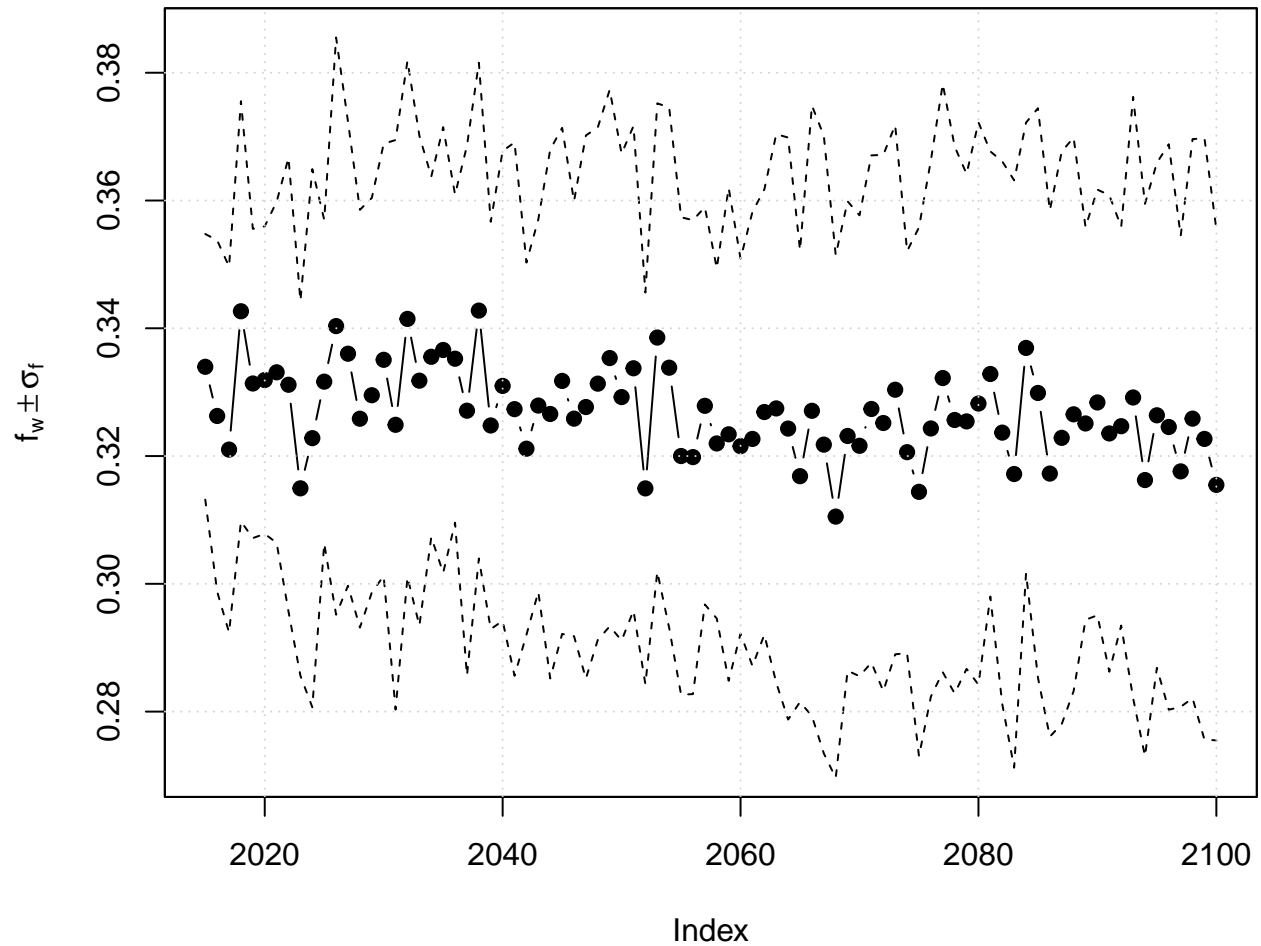
```
## [1] "SSP370 MU"
```

```
plot(MU.mean.ssp370,type='b',pch=19,main='Oslo: Wet-day intensity - SSP370',
     ylab=expression(mu%+-%sigma[mu]),
     ylim=range(MU.mean.ssp370 + c(-1,1)*MU.std.ssp370,na.rm=TRUE))
lines(MU.mean.ssp370 - MU.std.ssp370,lty=2)
lines(MU.mean.ssp370 + MU.std.ssp370,lty=2)
grid()
```

Oslo: Wet–day intensity – SSP370

```
print("SSP245 FW")
```

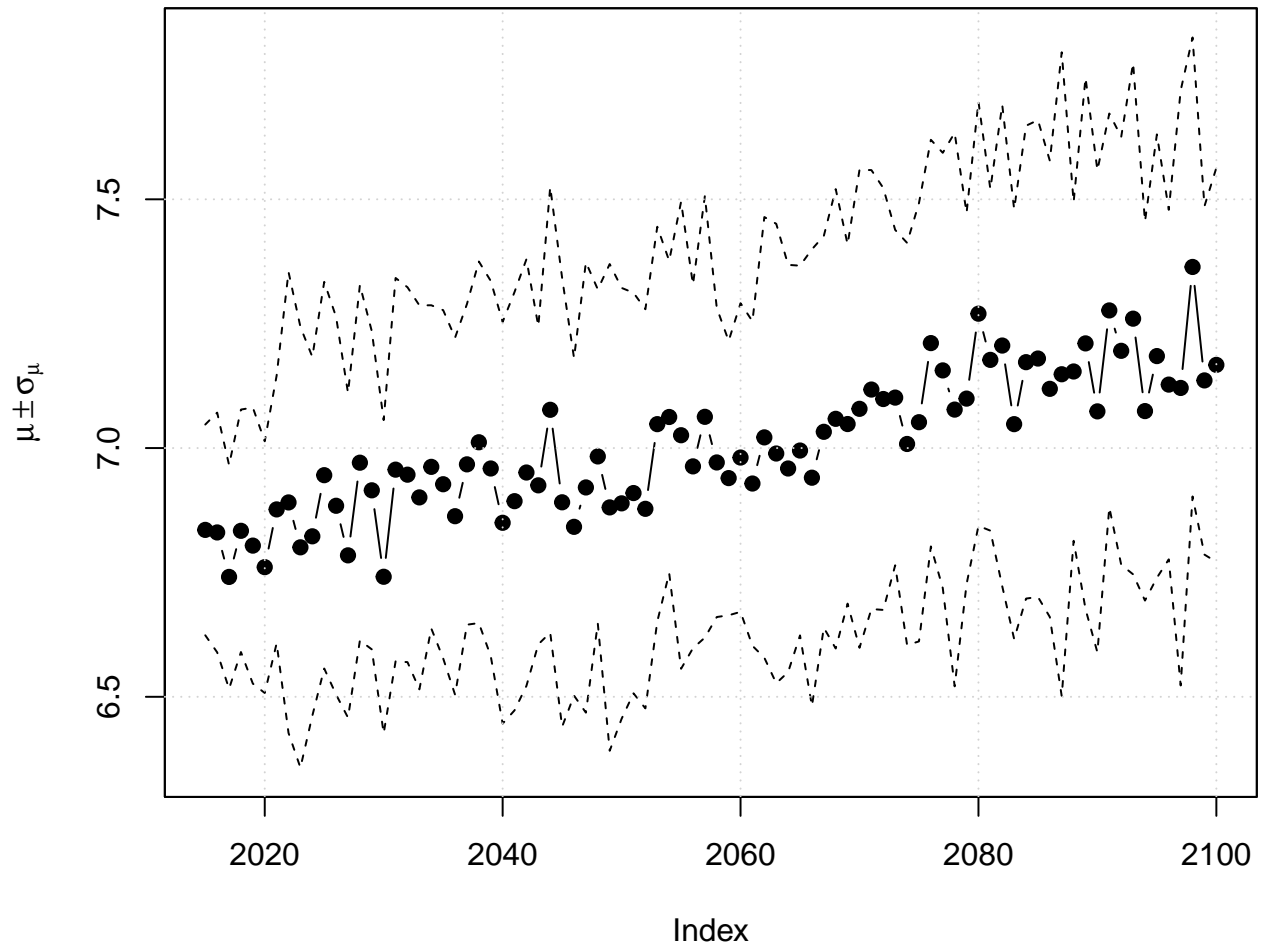
```
## [1] "SSP245 FW"
```

```
plot(FW.mean.ssp245,type='b',pch=19,main='Oslo: Wet-day frequency - SSP245',
     ylab=expression(f[w]%+-%sigma[f]),
     ylim=range(FW.mean.ssp245 + c(-1,1)*FW.std.ssp245,na.rm=TRUE))
lines(FW.mean.ssp245 - FW.std.ssp245,lty=2)
lines(FW.mean.ssp245 + FW.std.ssp245,lty=2)
grid()
```

Oslo: Wet–day frequency – SSP245

```
print("HSSP245 MU")
```

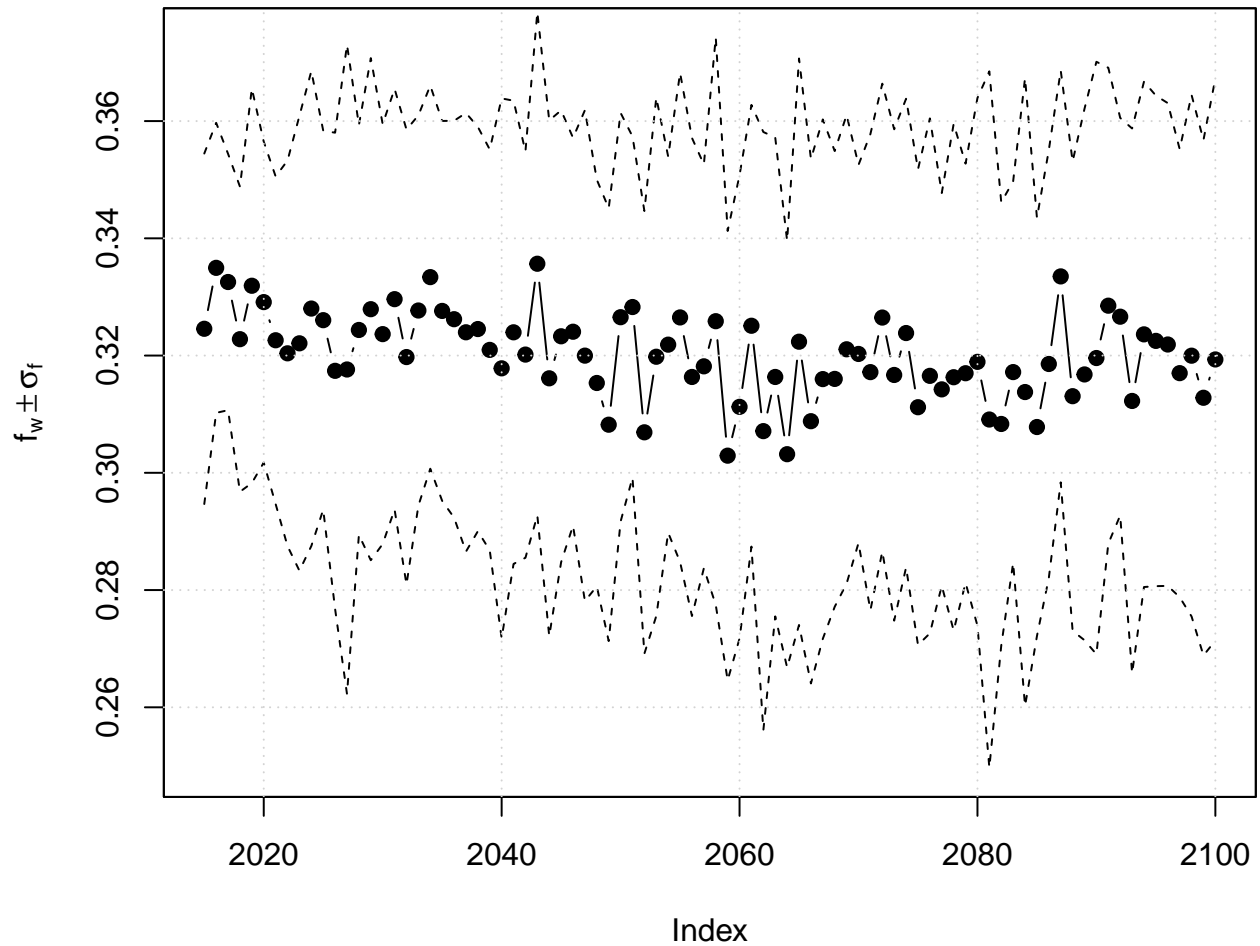
```
## [1] "HSSP245 MU"
```

```
plot(MU.mean.ssp245,type='b',pch=19,main='Oslo: Wet-day intensity - SSP245',
     ylab=expression(mu%+-%sigma[mu]),
     ylim=range(MU.mean.ssp245 + c(-1,1)*MU.std.ssp245,na.rm=TRUE))
lines(MU.mean.ssp245 - MU.std.ssp245,lty=2)
lines(MU.mean.ssp245 + MU.std.ssp245,lty=2)
grid()
```

Oslo: Wet–day intensity – SSP245

```
print("SSP585 FW")
```

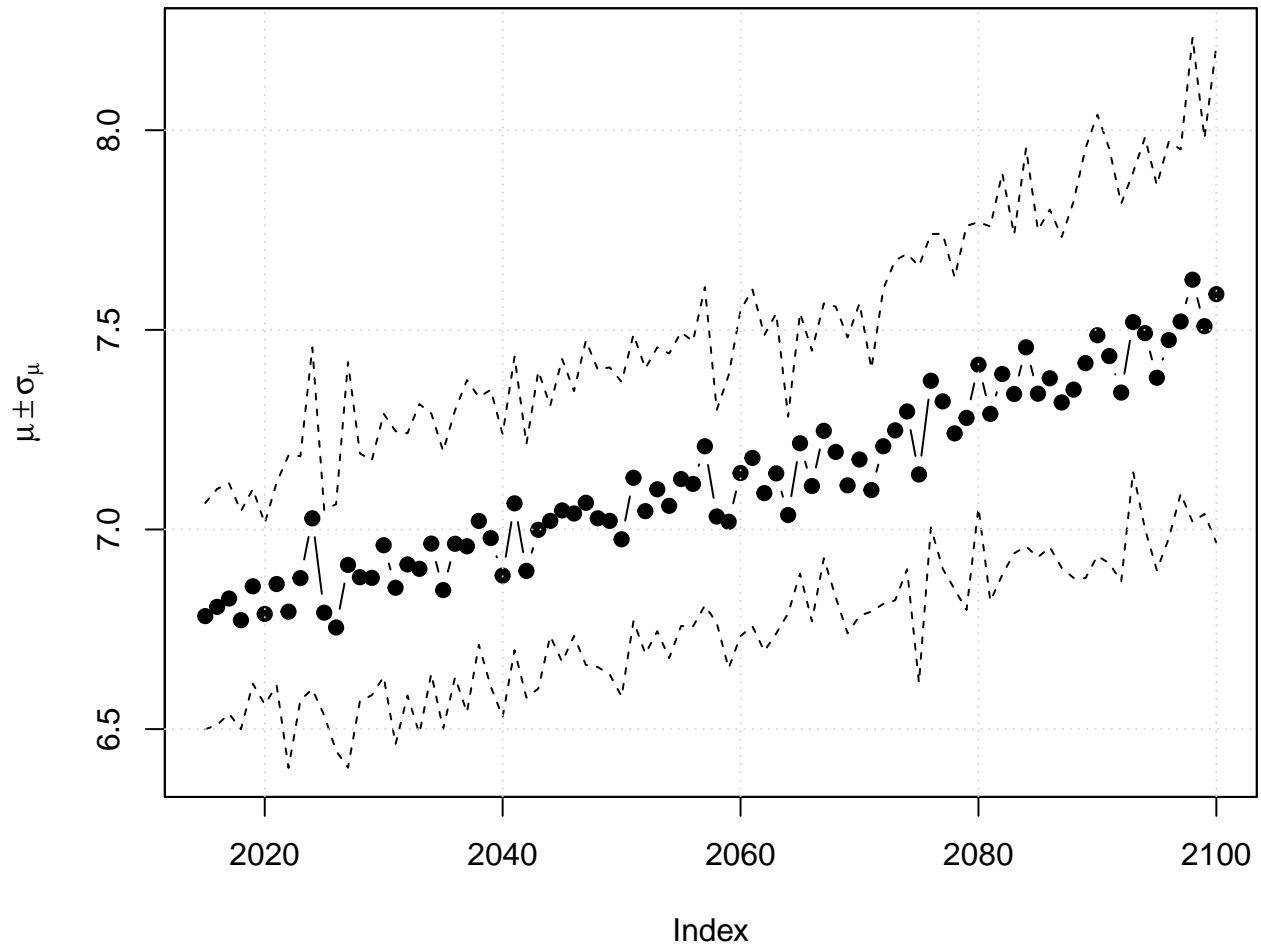
```
## [1] "SSP585 FW"
```

```
plot(FW.mean.ssp585,type='b',pch=19,main='Oslo: Wet-day frequency - SSP585',
     ylab=expression(f[w]%+-%sigma[f]),
     ylim=range(FW.mean.ssp585 + c(-1,1)*FW.std.ssp585,na.rm=TRUE))
lines(FW.mean.ssp585 - FW.std.ssp585,lty=2)
lines(FW.mean.ssp585 + FW.std.ssp585,lty=2)
grid()
```

Oslo: Wet–day frequency – SSP585

```
print("SSP585 MU")
```

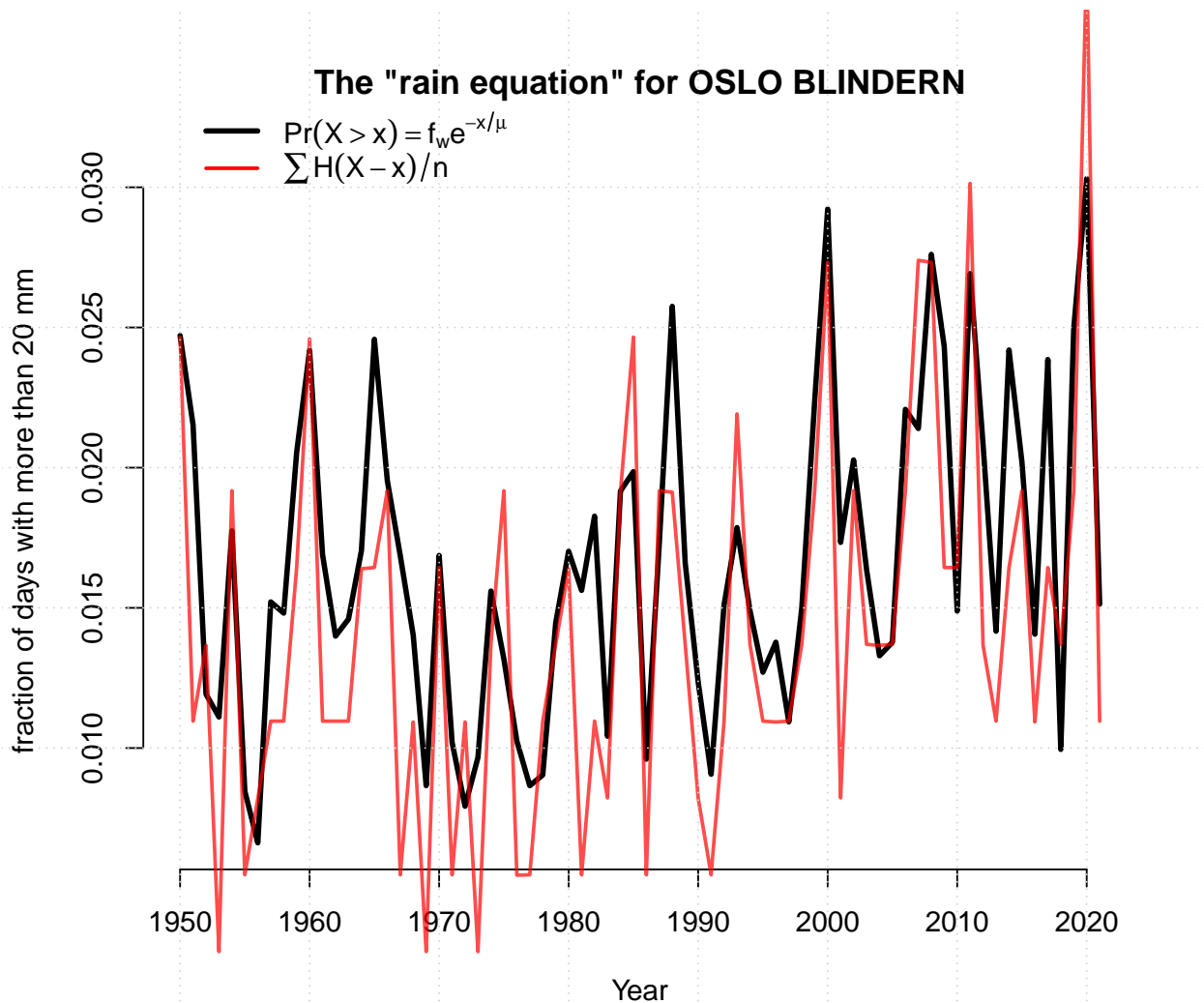
```
## [1] "SSP585 MU"
```

```
plot(MU.mean.ssp585,type='b',pch=19,main='Oslo: Wet-day intensity - SSP585',
     ylab=expression(mu%+-%sigma[mu]),
     ylim=range(MU.mean.ssp585 + c(-1,1)*MU.std.ssp585,na.rm=TRUE))
lines(MU.mean.ssp585 - MU.std.ssp585,lty=2)
lines(MU.mean.ssp585 + MU.std.ssp585,lty=2)
grid()
```

Oslo: Wet–day intensity – SSP585

2.4 Local 24-hr precipitation statistics and heavy precipitation

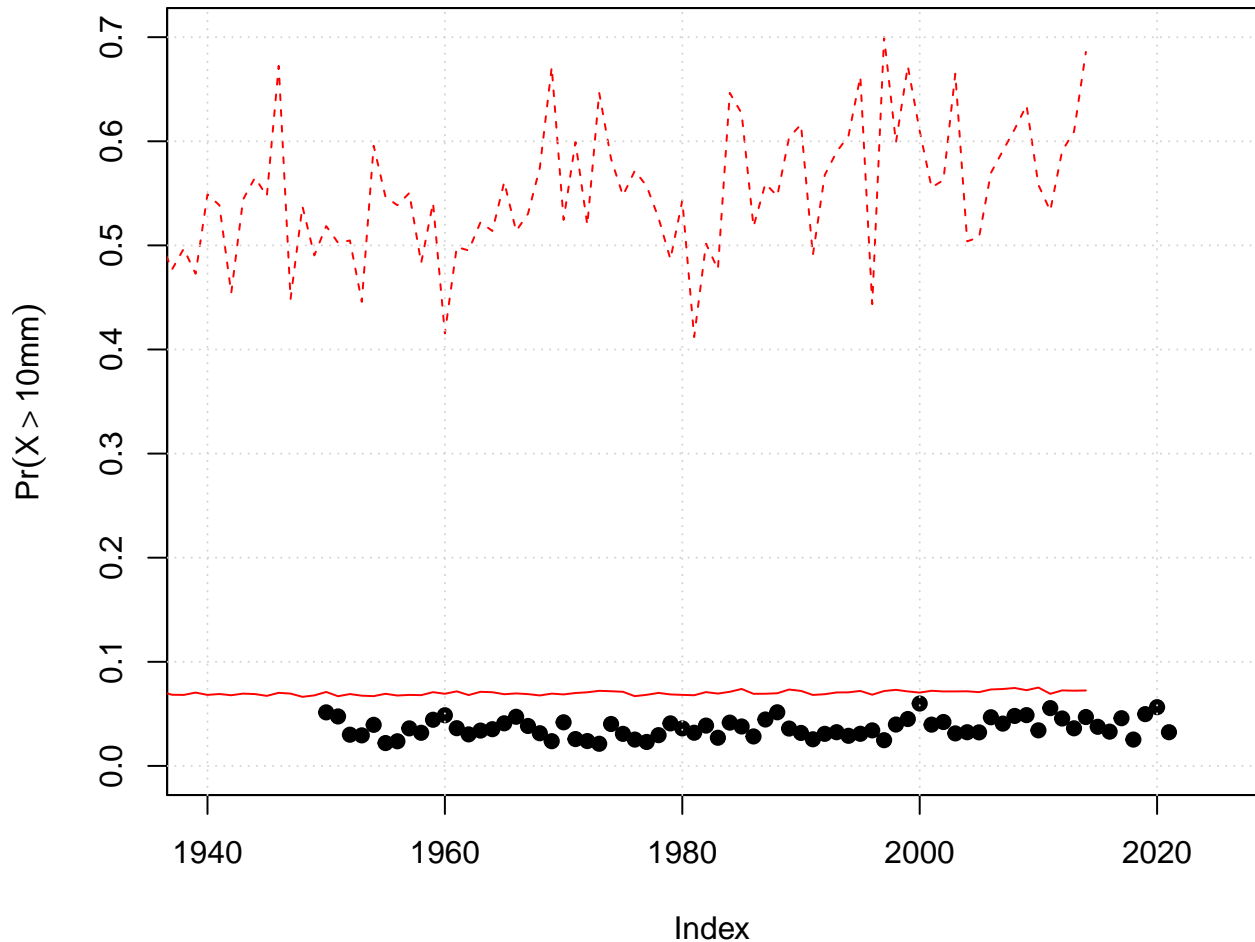
The following chunk reads the original daily precipitation saved as a temporary file and extracts data for Oslo which will be used in the following exercises.

The R-chunk below tests the rain equation against observations for Oslo daily rainfall:

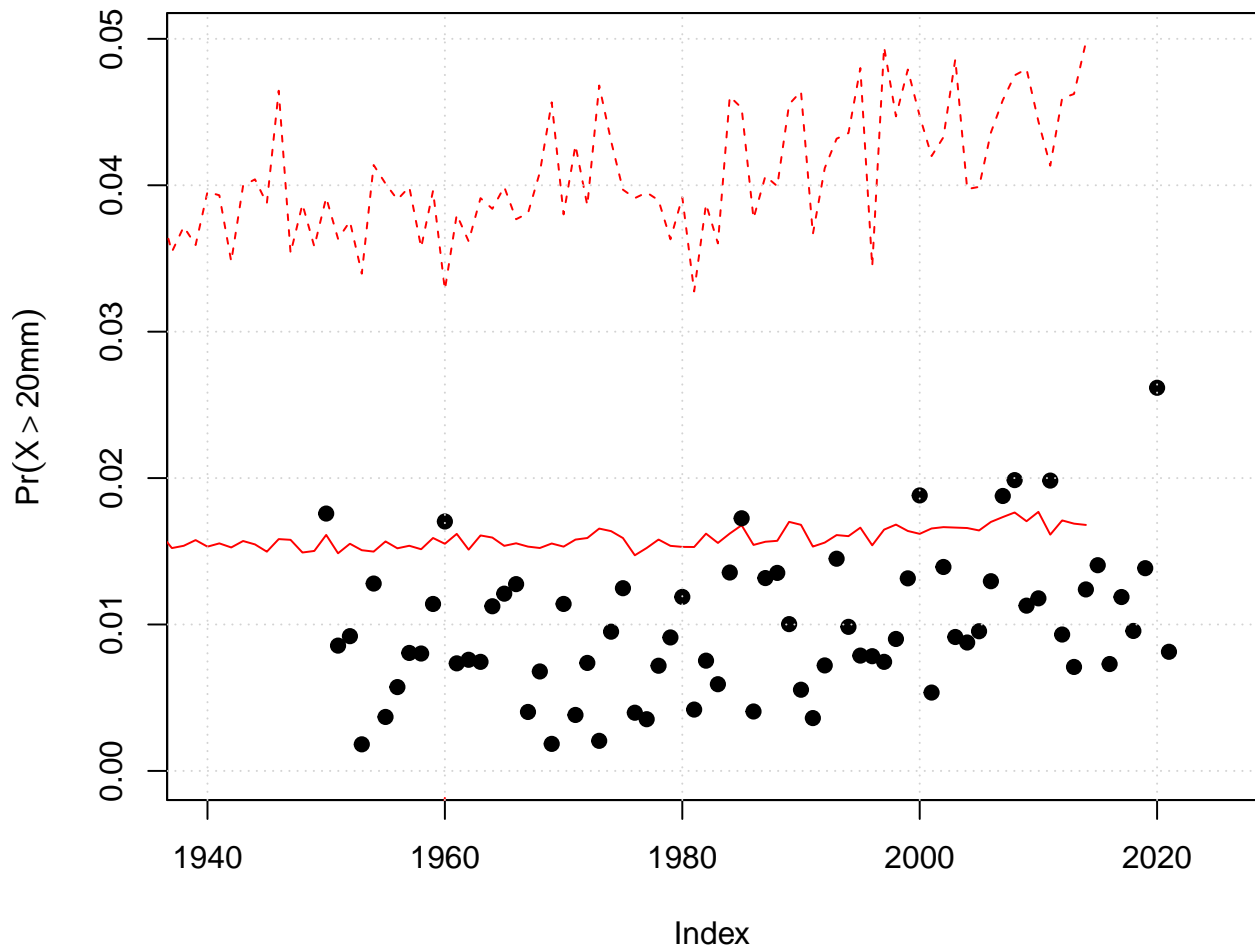
```
y <- subset(Y,is=loc(fw.oslo.ssp370))
z <- test.rainequation(y)
```

Although the match is not perfect, there is a clear correlation between the estimated probability (black curve) and the fraction of days with precipitation exceeding 20 mm/day (red curve).

We can compare the rain equation from observations with the downscaled results from HIST:

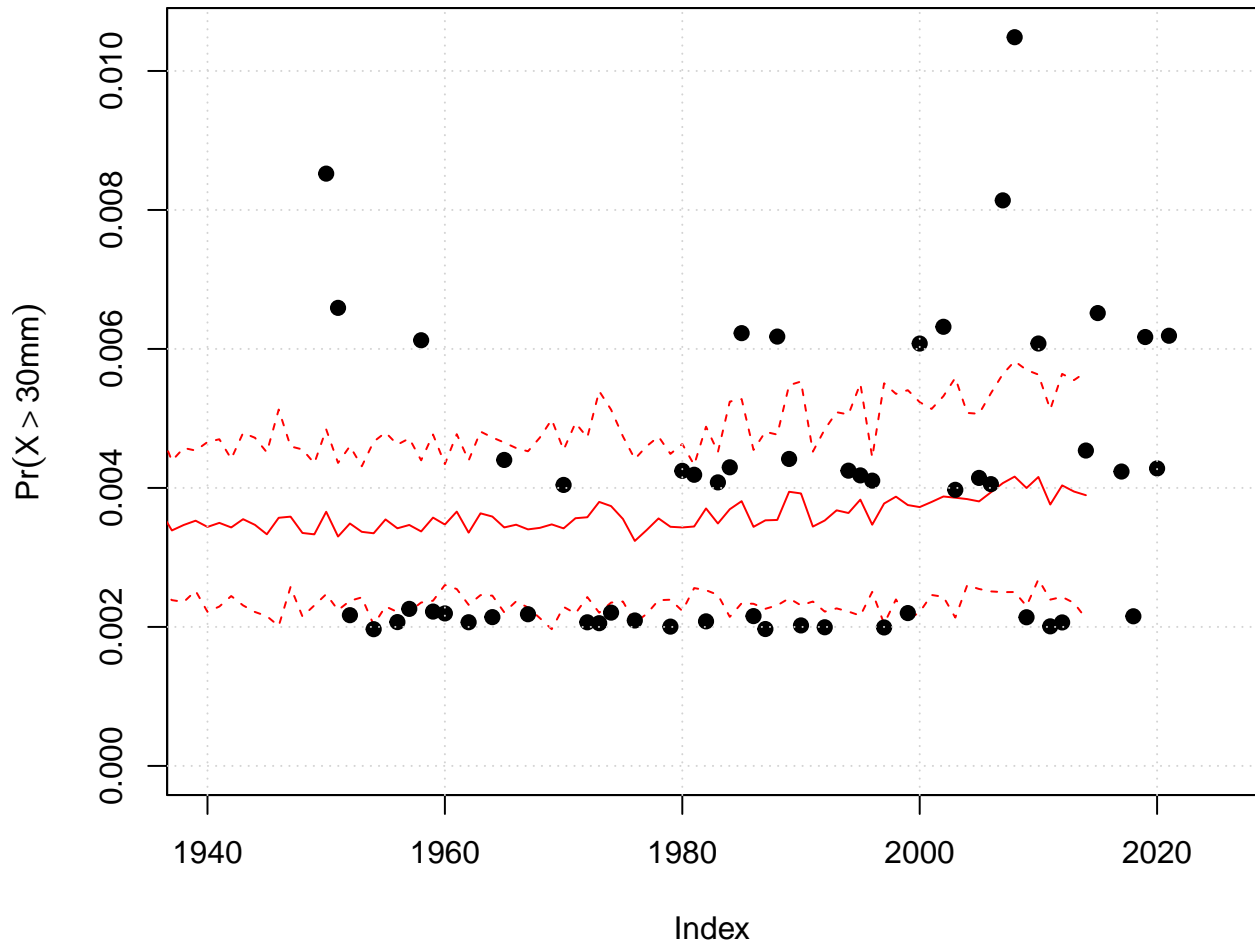
```
m.x10 <- zoo(FW.mean.hist*exp(-10/MU.mean.hist),order.by=index(fw.oslo.hist))
s.x10 <- zoo(
  sqrt( (FW.std.hist*exp(-10/MU.mean.hist)/FW.mean.hist*exp(-10/MU.mean.hist))^2 +
        (FW.mean.hist*exp(-10/MU.mean.hist)/FW.std.hist*exp(-10/MU.mean.hist))^2 ),
  order.by=index(fw.oslo.hist))
plot(m.x10,main='Oslo: Fraction of days with more than 10 mm precipitation',
     ylab=expression(Pr(X > 10*mm)),col='red',xlim=c(1940,2025),
     ylim=c(0,max(m.x10+s.x10,na.rm=TRUE)))
lines(m.x10+s.x10,lty=2,col='red')
lines(m.x10-s.x10,lty=2,col='red')
points(rainequation(y,threshold=10),pch=19)
grid()
```

Oslo: Fraction of days with more than 10 mm precipitation

A comparison between the probability of days with more than 21 mm precipitation from the historical ensemble of downscaled f_w and μ (solid curve shows ensemble mean and dashed curves indicate the ensemble mean plus/minus one standard deviation) gave a good match with the general level observed numbers (symbols), although the predicted ensemble was systematically highe then the observations.

```r
m.x20 <- zoo(FW.mean.hist*exp(-20/MU.mean.hist),order.by=index(fw.oslo.hist))
s.x20 <- zoo(
  sqrt( (FW.std.hist*exp(-20/MU.mean.hist)/FW.mean.hist*exp(-20/MU.mean.hist))^2 +
        (FW.mean.hist*exp(-20/MU.mean.hist)/FW.std.hist*exp(-20/MU.mean.hist))^2 ),
  order.by=index(fw.oslo.hist))
plot(m.x20,main='Oslo: Fraction of days with more than 20 mm precipitation',
     ylab=expression(Pr(X > 20*mm)),col='red',xlim=c(1940,2025),
     ylim=c(0,max(m.x20+s.x20,na.rm=TRUE)))
lines(m.x20+s.x20,lty=2,col='red')
lines(m.x20-s.x20,lty=2,col='red')
points(rainequation(y,threshold=20),pch=19)
grid()
```

Oslo: Fraction of days with more than 20 mm precipitation

A comparison between the probability of days with more than 20 mm precipitation from the historical ensemble of downscaled f_w and μ (solid curve shows ensemble mean and dashed curves indicate the ensemble mean plus/minus one standard deviation) gave a good match with the general level observed numbers (symbols). Nevertheless, the downscaled ensemble spans a wider range of fractions than the observed interannual variations in the fractions.

```
m.x30 <- zoo(FW.mean.hist*exp(-30/MU.mean.hist),order.by=index(fw.oslo.hist))
s.x30 <- zoo(
  sqrt( (FW.std.hist*exp(-30/MU.mean.hist)/FW.mean.hist*exp(-30/MU.mean.hist))^2 +
        (FW.mean.hist*exp(-30/MU.mean.hist)/FW.std.hist*exp(-30/MU.mean.hist))^2 ),
  order.by=index(fw.oslo.hist))
plot(m.x30,main='Oslo: Fraction of days with more than 30 mm precipitation',
     ylab=expression(Pr(X > 30*mm)),col='red',xlim=c(1940,2025),
     ylim=c(0,max(rainequation(y,threshold=30),m.x30+s.x30,na.rm=TRUE)))
lines(m.x30+s.x30,lty=2,col='red')
lines(m.x30-s.x30,lty=2,col='red')
points(rainequation(y,threshold=30),pch=19)
grid()
```

Oslo: Fraction of days with more than 30 mm precipitation

The results for days with more than 30 mm precipitation also agreed well with the observations.

2.5 Intensity-Duration-Frequency (IDF) analysis

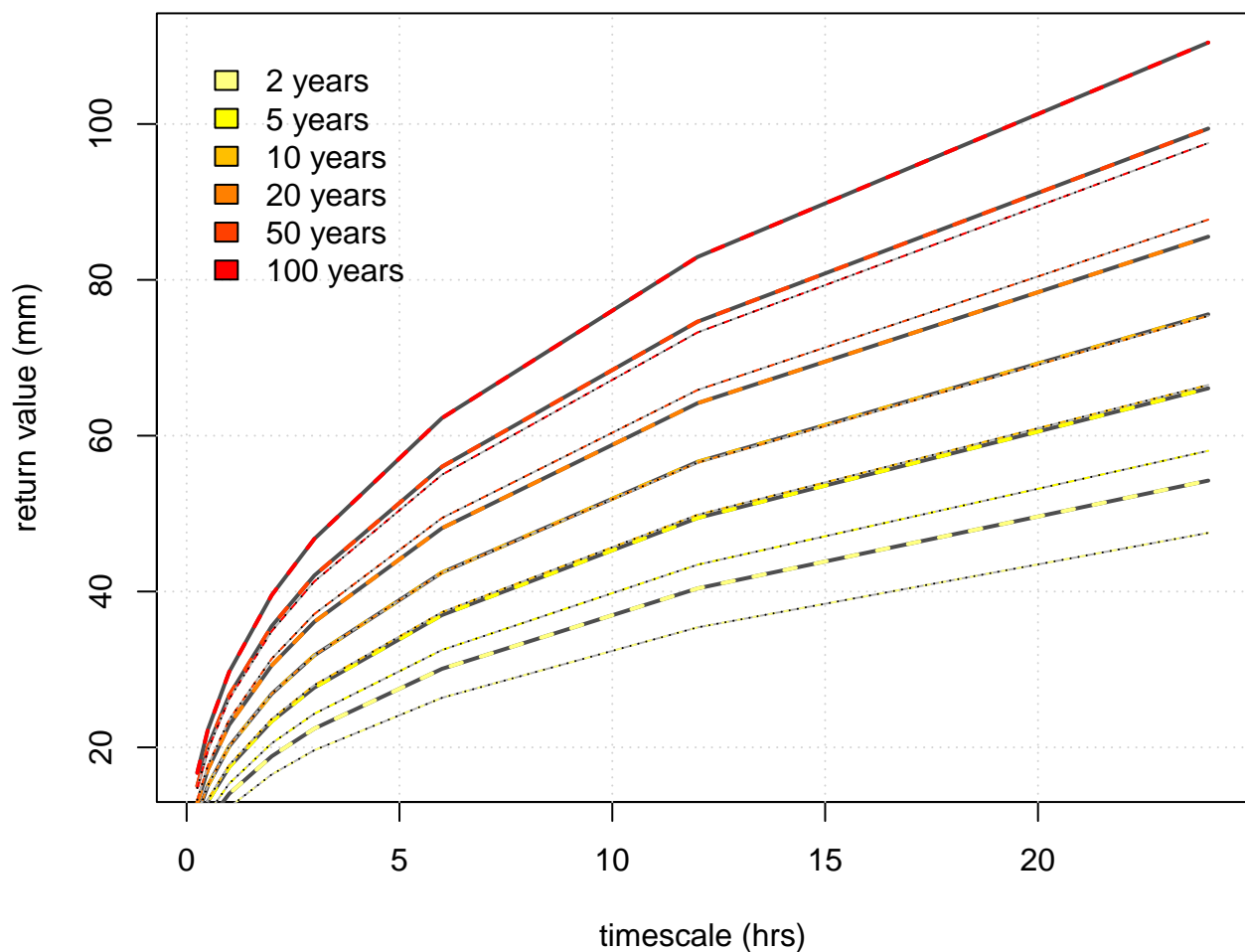
We can aggregate the ensemble and use its statistics to estimate intensity-duration-frequency (IDF) curves for selected locations according to Benestad et al. (2020): $x_\tau = \alpha\mu(L/24)^\zeta \ln(f_w\tau)$. We used data-frames and aggregation to estimate intensity-duration-frequency (IDF) curves for the ensemble mean as well as selected quantiles of the ensemble distribution.

The shape of the IDF curves are expected to reflect the presence and severity of various meteorological phenomena that generate precipitation: e.g. convection, cyclones, frontal systems, atmospheric rivers. It is not certain that all of them are skilfully reproduced by GCMs, but the information about their relative contribution to the total rainfall is embedded in the shape of these curves and derived semi-empirically.

```
## SSP370 assume the minima are now
print(range(index(MU.mean.ssp370))); print(range(index(FW.mean.ssp370)))
```

```
## [1] 2015 2100
```

```
## [1] 2015 2100
```

```
x1 <- list(mu=min(MU.mean.ssp370,na.rm=TRUE),fw=min(FW.mean.ssp370,na.rm=TRUE))
idf1 <- IDF(x1,plot=FALSE)
## SSP370 assmus the maxima are towards theend of the series
x2 <- list(mu=max(MU.mean.ssp370,na.rm=TRUE),fw=max(FW.mean.ssp370,na.rm=TRUE))
idf2 <- IDF(x2,plot=FALSE)
## Plot the IDF for the future: thick curves
plot.IDF(idf2)
## Compare with IDF representing the present: thin curves
lines.IDF(idf1,col='grey70')
lines.IDF(idf1,lty=2)
lines.IDF(idf1,lty=3,col='black')
```

```
## Scaling fractions assuming SS370
idf.scaling21 <- idf2/idf1
print("SSP370 IDF scaling ratio - just ensemble mean")
```

```
## [1] "SSP370 IDF scaling ratio - just ensemble mean"
```

```r
print(idf.scaling21)
```

```
##               2 years  5 years 10 years 20 years 50 years 100 years
## 0.25 hours 1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 0.5 hours  1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 1 hours    1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 2 hours    1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 3 hours    1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 6 hours    1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 12 hours   1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## 24 hours   1.141219 1.138122 1.136315 1.134832 1.133237  1.132238
## attr(,"L")
## [1]  0.25  0.50  1.00  2.00  3.00  6.00 12.00 24.00
## attr(,"tau")
## [1]    2    5   10   20   50  100
## attr(,"class")
## [1] "IDF"    "matrix"
```

```r
## SSP585 Beginning of the interval
x3 <- list(mu=min(MU.mean.ssp585,na.rm=TRUE),fw=min(FW.mean.ssp585,na.rm=TRUE))
idf3 <- IDF(x3,plot=FALSE)
## SSP585 End of the interval
x4 <- list(mu=max(MU.mean.ssp585,na.rm=TRUE),fw=max(FW.mean.ssp585na.rm=TRUE))
idf4 <- IDF(x4,plot=FALSE)
## SSP585
idf.scaling43 <- idf4/idf3
print("SSP585 IDF scaling ratio - just ensemble mean")
```

```
## [1] "SSP585 IDF scaling ratio - just ensemble mean"
```

```r
print(idf.scaling43)
```

```
##               2 years 5 years 10 years 20 years 50 years 100 years
## 0.25 hours 1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 0.5 hours  1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 1 hours    1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 2 hours    1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 3 hours    1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 6 hours    1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 12 hours   1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## 24 hours   1.378678 1.34245 1.321338 1.304025 1.285414  1.273768
## attr(,"L")
## [1]  0.25  0.50  1.00  2.00  3.00  6.00 12.00 24.00
## attr(,"tau")
## [1]    2    5   10   20   50  100
## attr(,"class")
## [1] "IDF"    "matrix"
```

```r
## SSP370 with variability
x6 <- list(mu=min(MU.mean.ssp370,na.rm=TRUE) - mean(MU.std.ssp370,na.rm=TRUE),
           fw=min(FW.mean.ssp370,na.rm=TRUE) - mean(FW.std.ssp370,na.rm=TRUE))
```

```r
idf6 <- IDF(x6,plot=FALSE)
## SSP370 Beginning of the interval
x7 <- list(mu=max(MU.mean.ssp370,na.rm=TRUE) + mean(MU.std.ssp370[1:5],na.rm=TRUE),
           fw=max(FW.mean.ssp370,na.rm=TRUE) + mean(FW.std.ssp370[1:5],na.rm=TRUE))
idf4 <- IDF(x7,plot=FALSE)
## SSP370 End of the interval
idf.scaling43 <- idf4/idf3
print("SSP370  IDF scaling ratio - ensemble mean and -/+ standard deviation")
```

```
## [1] "SSP370  IDF scaling ratio - ensemble mean and -/+ standard deviation"
```

```r
print(idf.scaling43)
```

```
##             2 years  5 years 10 years 20 years 50 years 100 years
## 0.25 hours 1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 0.5 hours  1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 1 hours    1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 2 hours    1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 3 hours    1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 6 hours    1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 12 hours   1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## 24 hours   1.201361 1.194851 1.191058 1.187948 1.184604  1.182511
## attr(,"L")
## [1]  0.25  0.50  1.00  2.00  3.00  6.00 12.00 24.00
## attr(,"tau")
## [1]    2    5   10   20   50 100
## attr(,"class")
## [1] "IDF"    "matrix"
```

```r
## SSP585 with variability
x6 <- list(mu=min(MU.mean.ssp585,na.rm=TRUE) - mean(MU.std.ssp585,na.rm=TRUE),
           fw=min(FW.mean.ssp585,na.rm=TRUE) - mean(FW.std.ssp585,na.rm=TRUE))
idf6 <- IDF(x6,plot=FALSE)
## SSP585 Beginning of the interval
x7 <- list(mu=max(MU.mean.ssp585,na.rm=TRUE) + mean(MU.std.ssp585,na.rm=TRUE),
           fw=max(FW.mean.ssp585,na.rm=TRUE) + mean(FW.std.ssp585,na.rm=TRUE))
idf4 <- IDF(x7,plot=FALSE)
## SSP585 End of the interval
idf.scaling43 <- idf4/idf3
print("ssp585  IDF scaling ratio - ensemble mean and -/+ standard deviation")
```

```
## [1] "ssp585  IDF scaling ratio - ensemble mean and -/+ standard deviation"
```

```r
print(idf.scaling43)
```

```
##             2 years  5 years 10 years 20 years 50 years 100 years
## 0.25 hours 1.232541 1.225783 1.221844 1.218615 1.215143   1.21297
## 0.5 hours  1.232541 1.225783 1.221844 1.218615 1.215143   1.21297
## 1 hours    1.232541 1.225783 1.221844 1.218615 1.215143   1.21297
## 2 hours    1.232541 1.225783 1.221844 1.218615 1.215143   1.21297
## 3 hours    1.232541 1.225783 1.221844 1.218615 1.215143   1.21297
```

```
## 6 hours     1.232541 1.225783 1.221844 1.218615 1.215143    1.21297
## 12 hours    1.232541 1.225783 1.221844 1.218615 1.215143    1.21297
## 24 hours    1.232541 1.225783 1.221844 1.218615 1.215143    1.21297
## attr(,"L")
## [1]  0.25  0.50  1.00  2.00  3.00  6.00 12.00 24.00
## attr(,"tau")
## [1]   2   5  10  20  50 100
## attr(,"class")
## [1] "IDF"    "matrix"
```

Our comparison between IDF curves estimated from the SSP370 ensemble mean for the start and the end of the runs suggests similar scaling factors for different duration, as expected from the formulae used here $(\alpha\mu(L/24)^\zeta \ln(f_w\tau))$. The scaling factors vary with the return interval, 1.13–1.14 for downscaled results for the SSP370 ensemble mean, but these figures may be somewhat conservative since f_w and μ are not entirely specified by the large-scale conditions and it doesn't include decadal variability. For SSP585, the scaling ratios are 1.27–1.38.

If we account for decadal variability, subtracting one ensemble spread standard deviation from the start and adding one standard deviation to the end, the scaling obtained for SSP370 is 1.18–1.20.

However, if the GCMs do not skillfully reproduce phenomena such as convection and frontal systems with sufficient realism, we can expect these estimates to be somewhat conservative.

2.5.1 PCA-based analysis?

An alternative approach is based on Parding et al. (2023) using PCA to represent the curves of the IDF curves and a regression analysis to link them to climatic parameters.

2.6 Downscaling number of wet-days directly

In addition to downscaling statistical parameters such as f_w and μ, it may be possible to estimate other types of statistical information such as probabilities or number of events directly.

The number of days is a discrete number and doesn't follow a normal distribution. For this reason, the analysis is focused on a single station rather than a group of stations, because PCA is designed to operate on the covariance matrix of continuous and normally distributed data. Single station series have lower signal-to-noise ratio than station groups represented in terms of PCA.

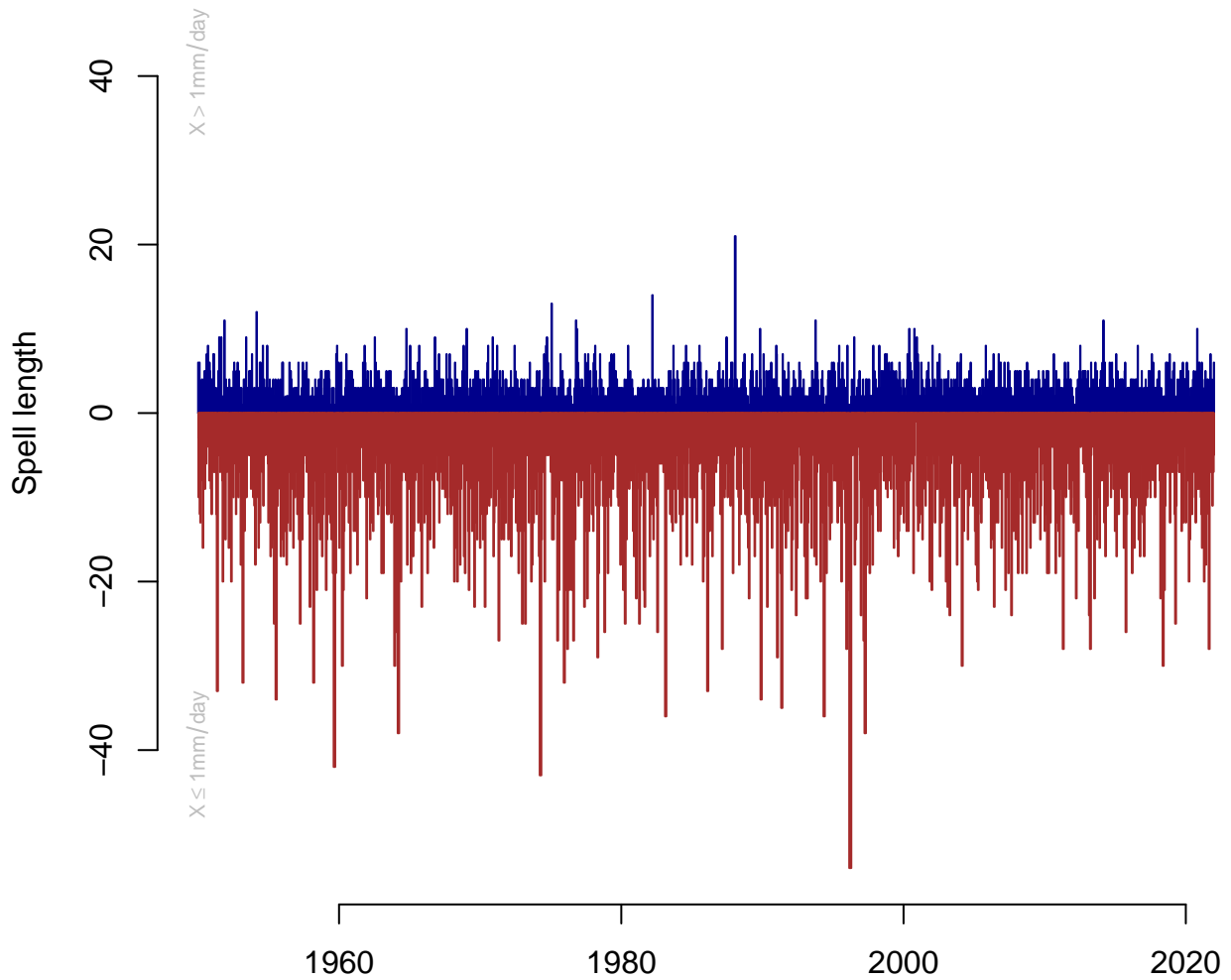
2.7 Examine droughts

Information about extremes such as drought can be quantified in terms of the probability of the duration of dry spells exceeding a given threshold. If the duration statistics approximately follows a geometric distribution, then a key parameter is the mean duration. Here is an example how the annual mean dry-spell duration is utilised in connection with an empirical-statistical downscaling exercise for Norway.

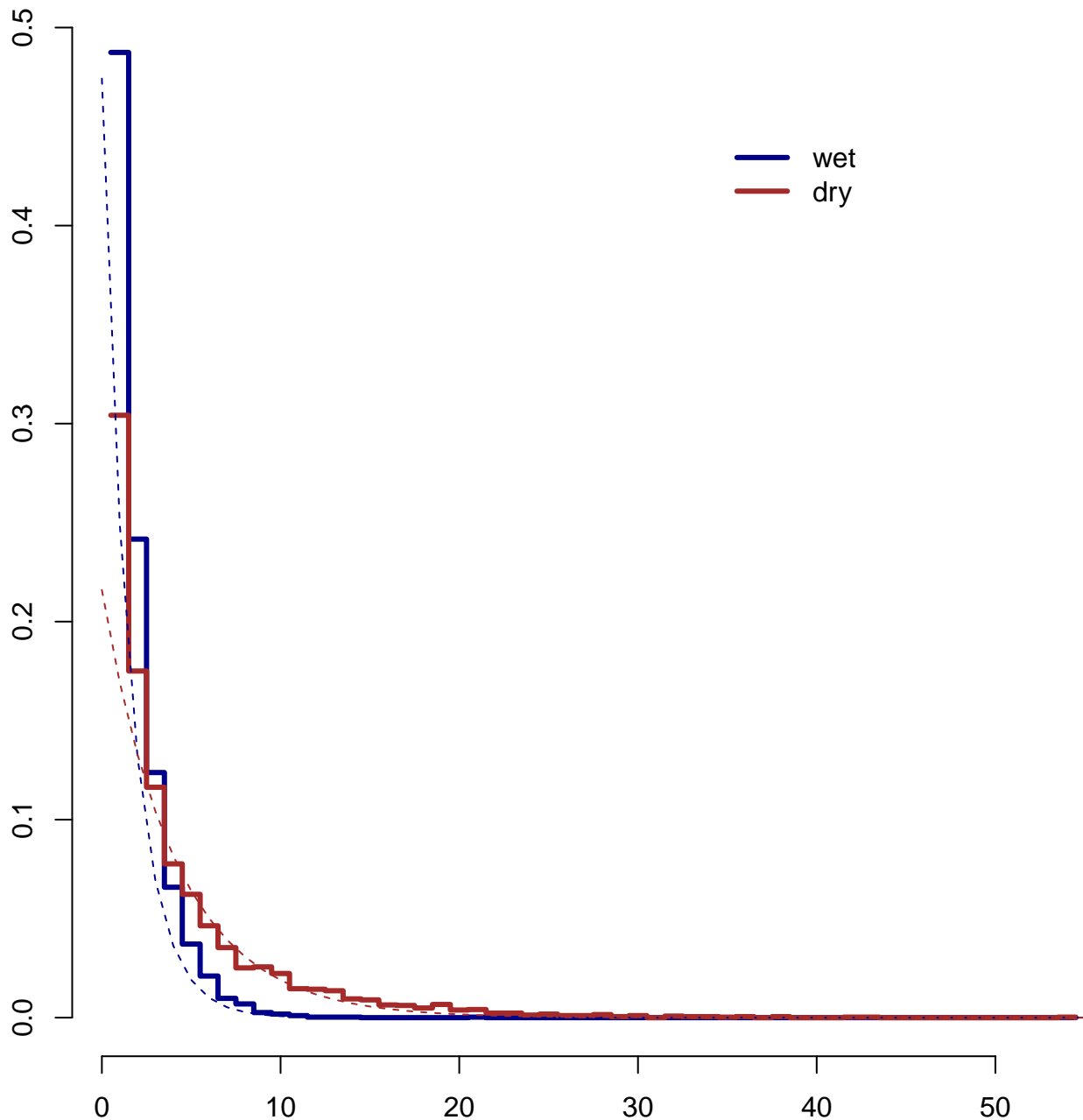
```
## Extract the spell length statistics of daily precipitation data from Oslo
## with a wet-day threshold of 1 mm/day.
s <- spell(y,threshold=1)
```

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

```
plot(s,new=FALSE)
```

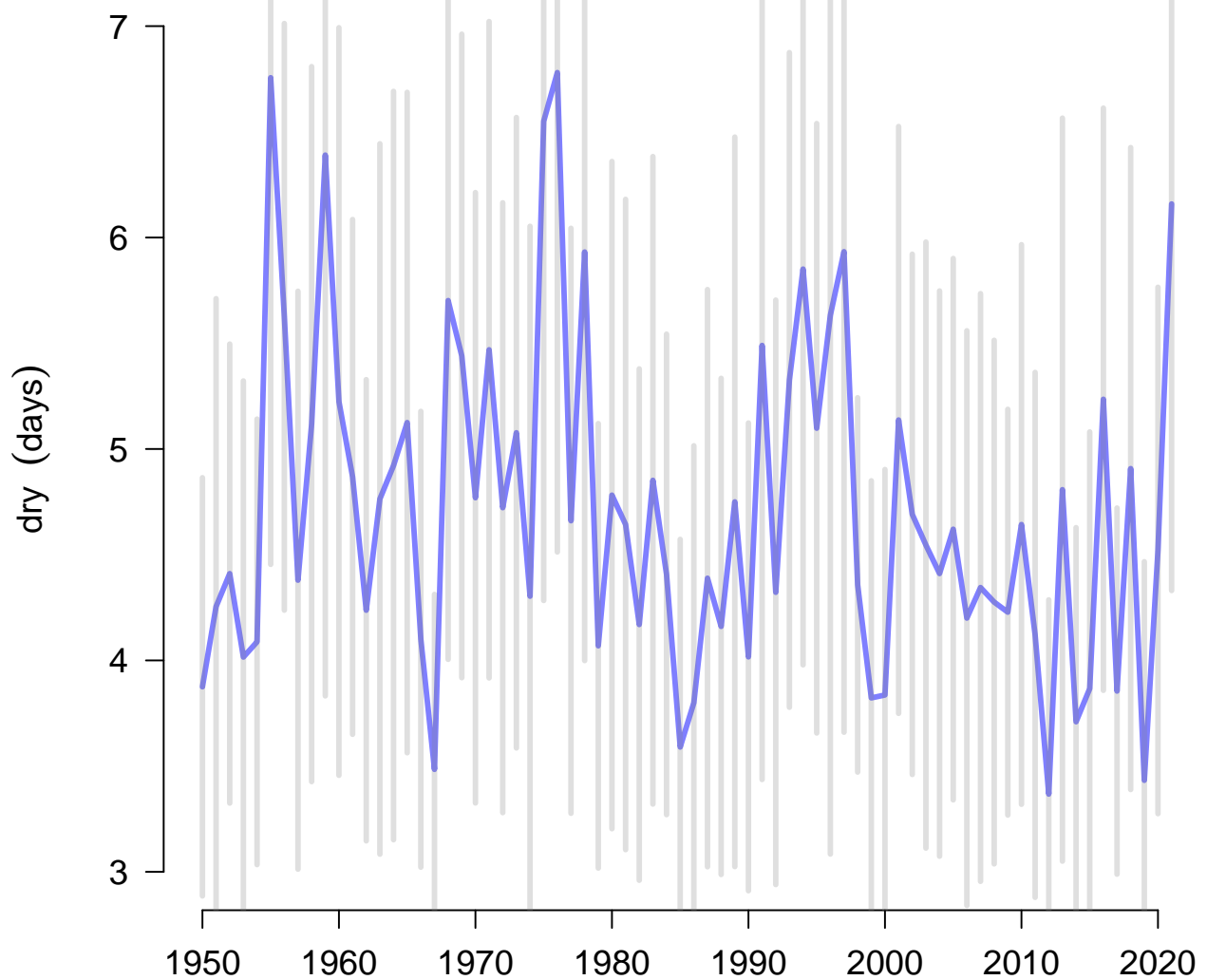
OSLO BLINDERN: wet and dry

```
## NULL
```

```
## Compare the statistics to a geometric distribution
par(mar=c(2,2,2,2))
hist(s,new=FALSE)
```

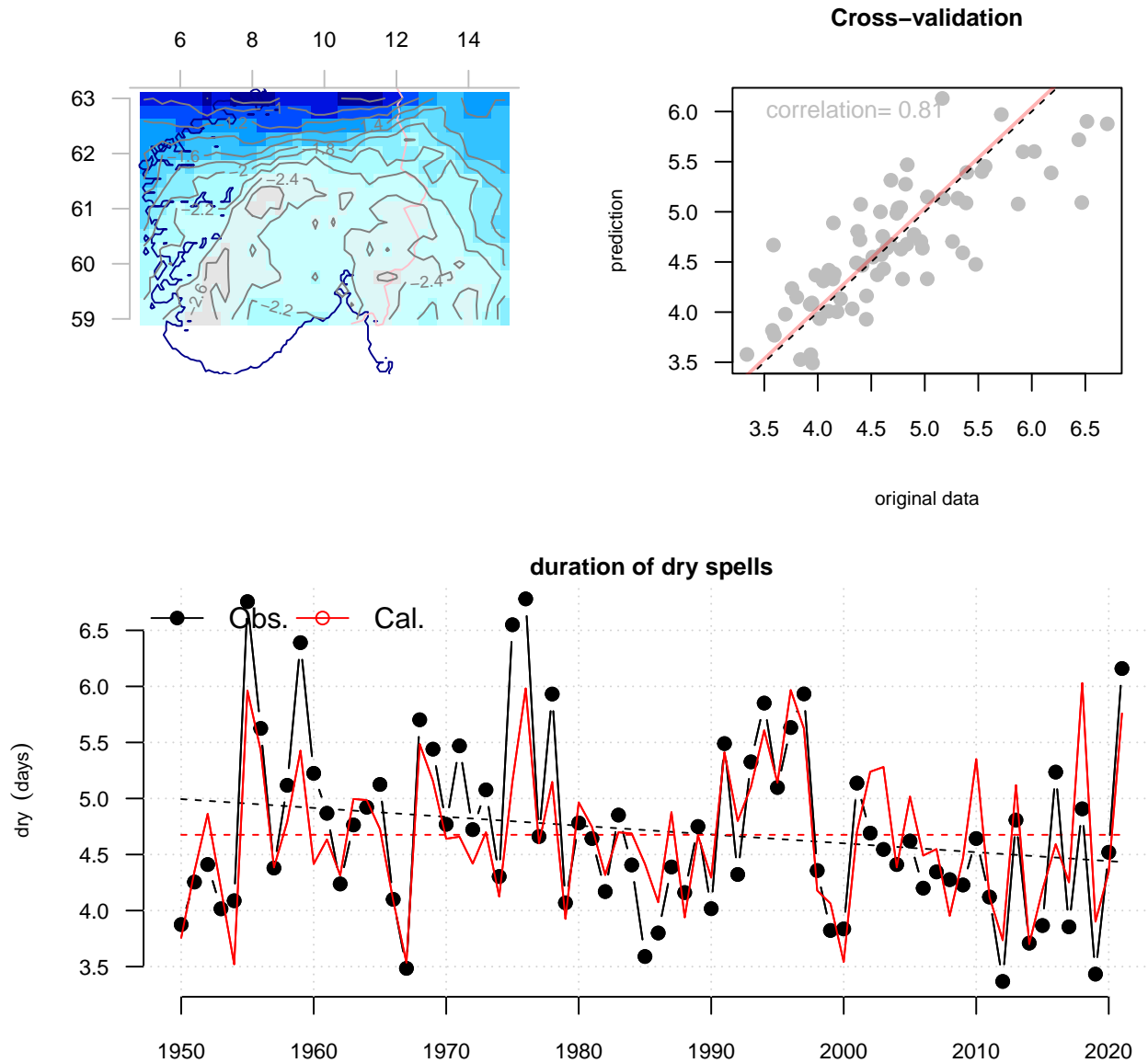
OSLO BLINDERN wet and dry spell duration

The duration statistics of wet and dry are shown above and the histogram compares their empirical distributions (solid) with the geometric distribution (dashed lines). The geometric distribution gives an approximate representation of the statistics, but there are some deviations for the shorter spells. for the geometric distribution $Pr(x = k) = (1 - p)^{k-1}p$, the success probability p is a key parameter and $k = [1, 2, 3, \ldots]$ is the duration in days. The mean duration is $1/p$, and hence is the inverse of the parameter defining the shape of the distribution. Because of this, we seek the mean duration of dry spells to estimate the probability of droughts with a given number of successive number of dry days.

We can aggregate the dry spell duration statistics into the annual mean dry spell length and test whether it is related to the wet-day frequency on larger spatial scales. Below is an attempt for Oslo precipitation:

```
mean.dry <- annual(subset(s,is=2),FUN='mean',nmin=1)
plot(mean.dry,main='Oslo: Annual mean dry-spell duration',new=FALSE)
```

Oslo: Annual mean dry–spell duration

```
FW <- retrieve('~/data/ERA5/ERA5_fw_year.nc',lon=c(5,15),lat=c(59,63))
```

```
## [1] "Warning : Frequency found in the attribute does not match the frequency detected in data"
```

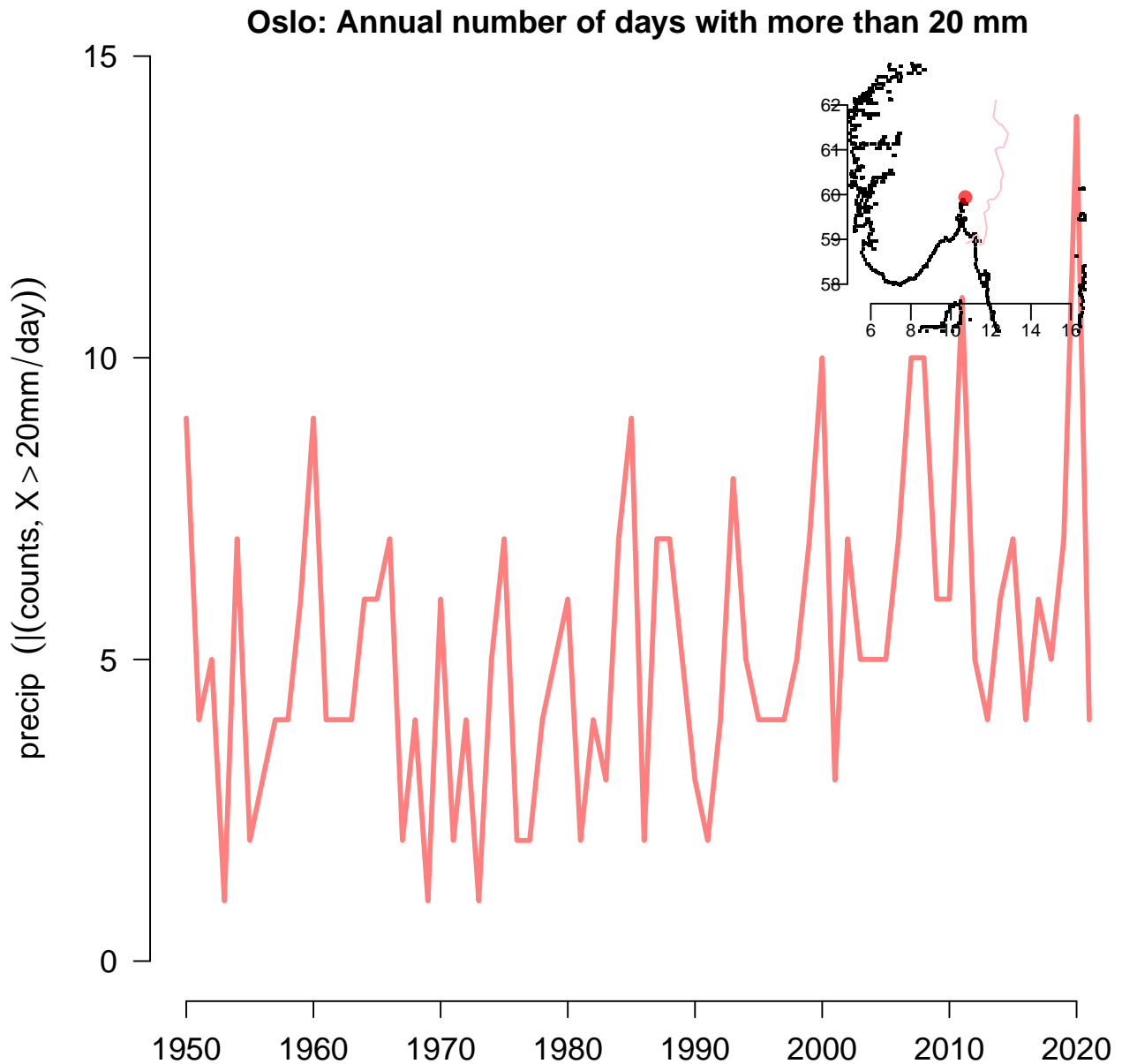
```
index(FW) <- year(FW)
eof.fw <- EOF(FW)
attr(eof.fw,'time') <- range(index(FW))
ds.dry <- DS(mean.dry,eof.fw,method = 'glm')
plot(ds.dry,new=FALSE)
```

The results from this exercise suggests that the mean duration approximately follows the large-scale wet day frequency f_w from the ERA5 reanalysis, which is not unexpected. The spatial pattern with low f_w for ERA5 in the vicinity of Oslo also supports this connection, although there are two local minima and none is exactly on top of Oslo.

2.8 Examine the number of very wet-days

For the number of events, we can use a framework based on a stochastic process with Poisson distribution and a rate λ so that $Pr(X = k) = \frac{\lambda^2 e^{-\lambda}}{k!}$. We can explore the predictor $n \propto f_w e^{-x/\mu}$ for different thresholds $x = [20, 21, ...]$ to explore pattern and possibility for extrapolation. Test for Oslo. GLM.

One way to downscale information about extremes is to estimate the number of events over a period and assume that the number would follow a Poisson distribution if it were random. Here is an example with number of 'very wet days' (more than 20 mm/day) per year.

Here, we expect that the number of wet days follows a similar pattern as shown above with `test.rainequation` and a predictor derived through a transformation of both f_w and μ according to $Pr(X > x) = f_w \exp(-x/\mu)$:

```
nvwet <- annual(y,FUN='count',threshold=20,nmin=1)
plot(nvwet,main='Oslo: Annual number of days with more than 20 mm',new=FALSE)
```

```
MU <- 1000*retrieve('~/data/ERA5/ERA5_mu_year.nc',lon=c(5,15),lat=c(59,63))
```

```
## [1] "Warning : Frequency found in the attribute does not match the frequency detected in data"
```

```
index(MU) <- year(MU)
attr(MU,'unit') <- 'mm/day'
## Transform the fields into
fFWMU <- FW*exp(-20/MU)
class(fFWMU) <- class(MU)
## Copy the attributes
fFWMU <- attrcp(MU,fFWMU)
```

```
attr(fFWMU,'variable') <- 'probability'
attr(fFWMU,'unit') <- 'fraction'
print("FW*exp(-20/MU)")
```

```
## [1] "FW*exp(-20/MU)"
```

```
eof.ffwmu <- EOF(fFWMU)
attr(eof.ffwmu,'time') <- range(index(FW))
ds.nvwet1 <- DS(nvwet,eof.ffwmu,method = 'glm')
plot(ds.nvwet1,new=FALSE)
```

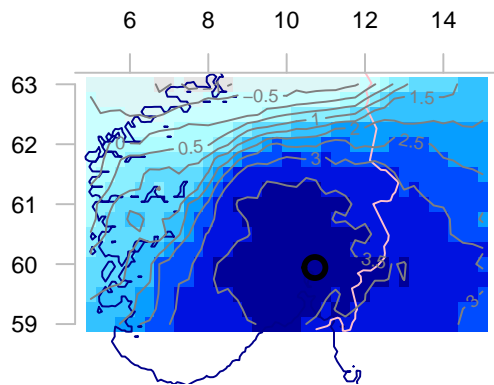
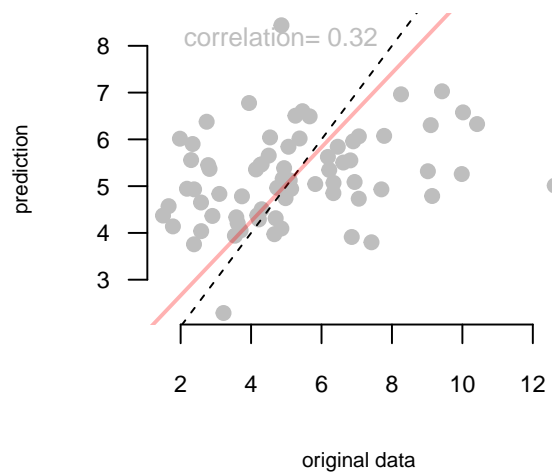
```
print("FW")
```

```
## [1] "FW"
```

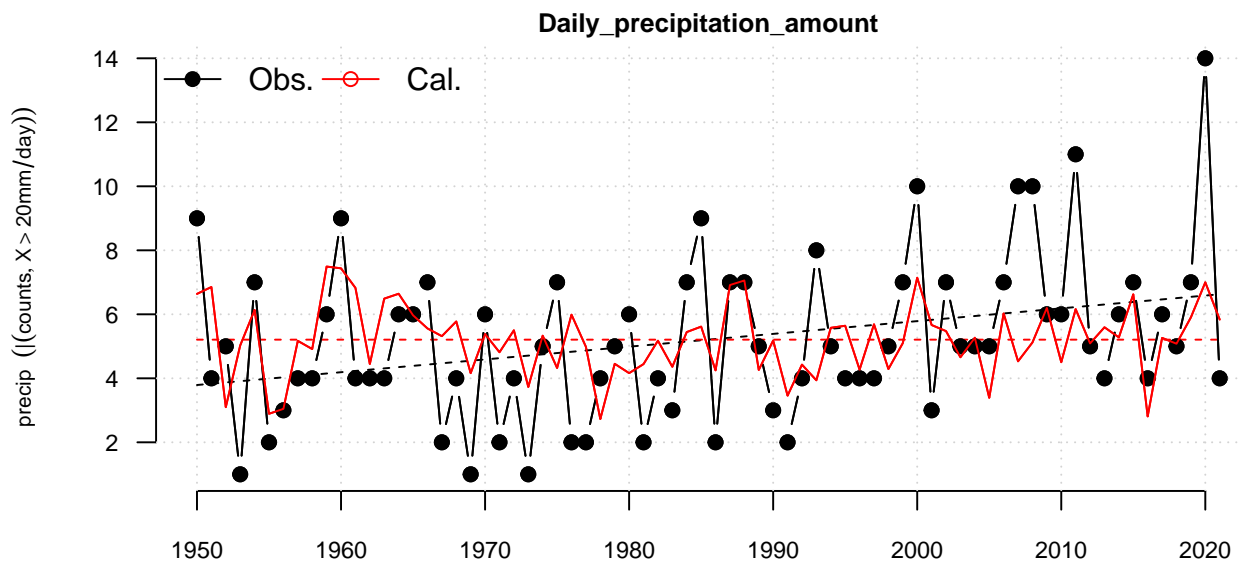
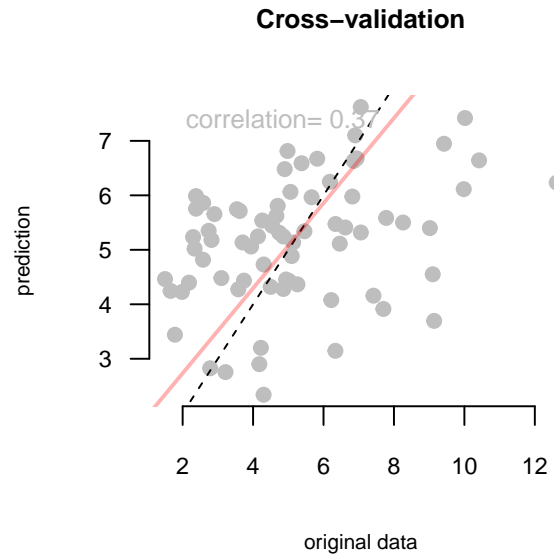
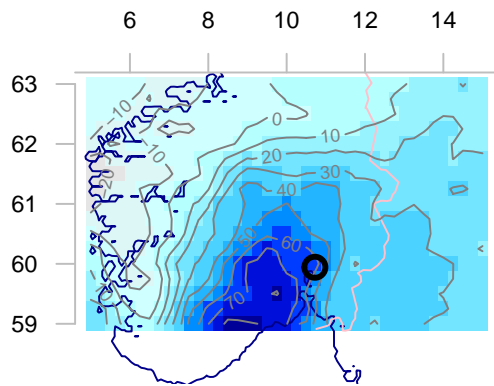
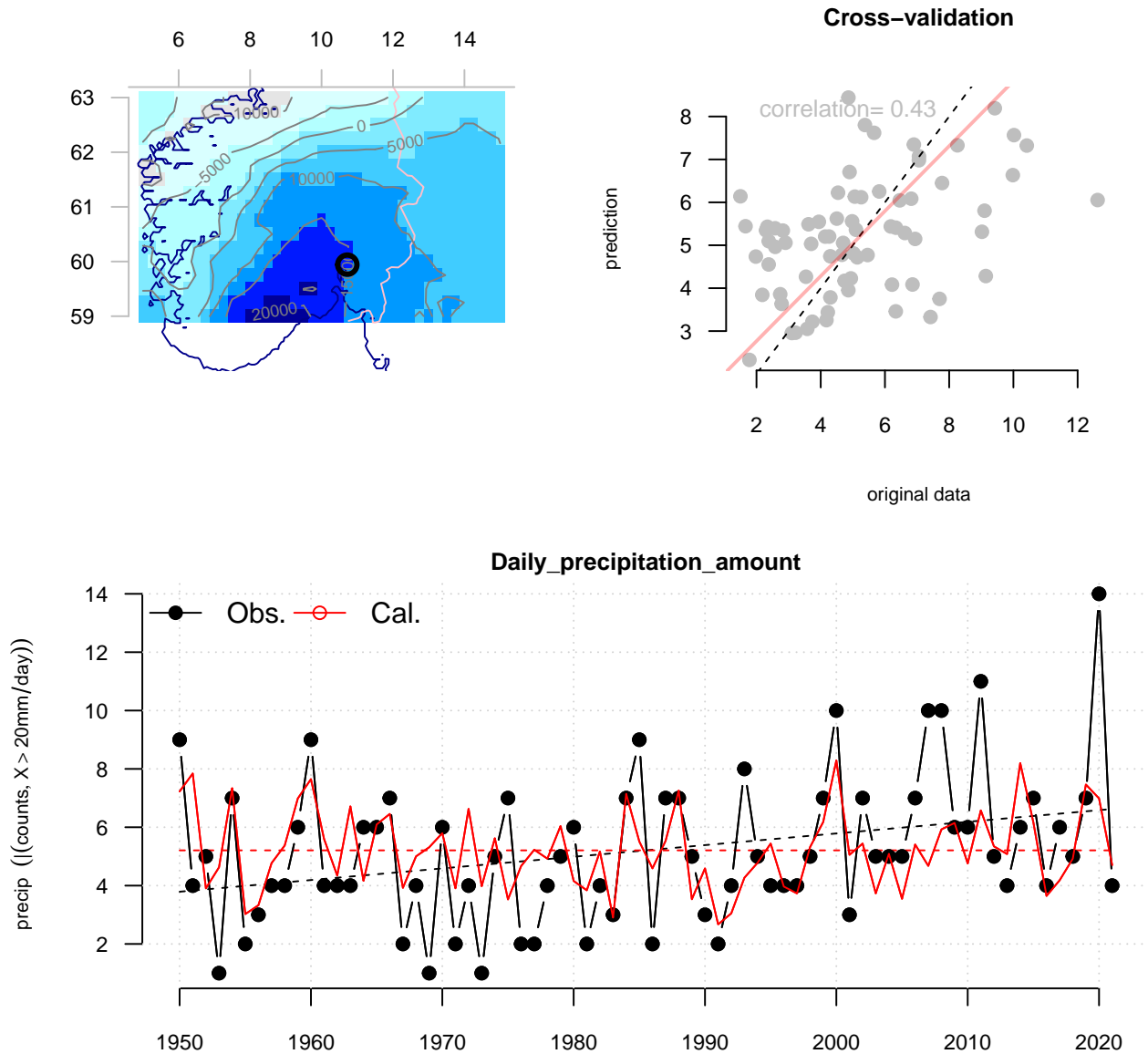
```
eof.fw <- EOF(FW)
attr(eof.fw,'time') <- range(index(FW))
ds.nvwet2 <- DS(nvwet,eof.fw,method = 'glm',)
plot(ds.nvwet2,new=FALSE)
```

```
print("MU")
```

```
## [1] "MU"
```

```
eof.mu <- EOF(MU)
attr(eof.mu,'time') <- range(index(FW))
ds.nvwet3 <- DS(nvwet,eof.mu,method = 'glm')
plot(ds.nvwet3,new=FALSE)
```

Cross–validation

Daily_precipitation_amount

```
print("FW*MU")
```

```
## [1] "FW*MU"
```

```
PT <- 365.25*FW*MU
class(PT) <- class(FW)
attr(PT,'variable') <- 'precip'
attr(PT,'unit') <- 'mm'
PT <- attrcp(FW,PT)
eof.fwmu <- EOF(PT)
attr(eof.fwmu,'time') <- range(index(FW))
ds.nvwet4 <- DS(nvwet,eof.fwmu,method = 'glm')
plot(ds.nvwet4,new=FALSE)
```

The results from downscaling the number of wet days directly did not capture the observations well in this case. Hence, it seems better to estimate the number/probability of events through downscaling of f_w and μ from their large-scale fields. However, the spatial weights in ERA5 are in the vicinity of Oslo, supporting a connection, albeit imperfect in terms of the magnitude of the number of events.

2.9 Maps and gridded results

The gridded data makes it possible to show maps for the Nordic region. Here `gridmap`involved kriging with altitude as a covariable, based on the R-package `LatticeKrig` https://cran.r-project.org/web/packages/LatticeKrig/index.html from UCAR. The default for `dsensemble`-objects in the `esd` package is to show the ensemble mean:

We want to produce maps of change in $Pr(X > x) = f_w \exp{(-x/\mu)}$ based on the formulation suggested by Benestad et al. (2019): $dPr/dt = \exp{(-x/\mu)}df_w/dt + f_w x/\mu^2 \exp{(-x/\mu)}d\mu/dt$. Hence, we require maps for f_w, df_w/dt, μ, and $d\mu/dt$ for each emission scenario.

2.9.1 Wet-day frequency

The following chunk provides maps of mean downscaled f_w and its trend based on the SSP370 emission scenario.

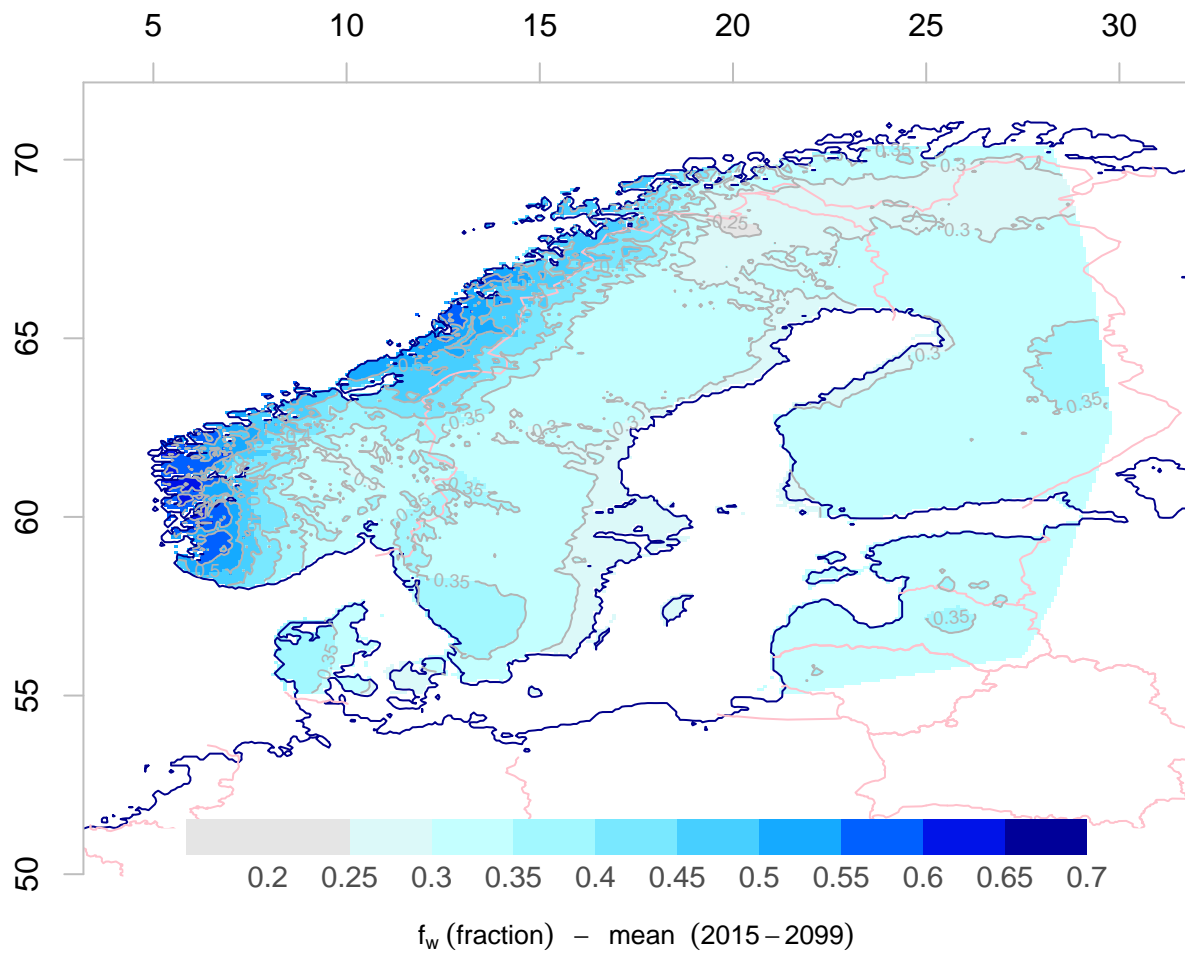
```r
## function estimating proportional trend in terms of %
ptrend <- function(x,na.rm=TRUE) {
  mx <- map(x,FUN='mean',plot=FALSE)
  tx <- map(x,FUN='trend',plot=FALSE)
  y <- 100*tx/mx
  class(y) <- class(tx)
  y <- attrcp(tx,y)
  attr(y,'unit') <- '"%"'
  return(y)
}

## This function extracts the interval where all members of the ensemble are present. It differs from '
## as it works on the PCA-based ensemble results as opposed to single stations.
allMs <- function(x) {
  nt <- length(index(x))
  cls <- class(x)
  ## Remove members with short time span
  nv <- unlist(lapply(x,function(x) if (length(dim(x))==2) y <-nv(x[,1]) else
    y <- NA))[-c(1,2,length(x))]
  im <- nv >= quantile(nv,probs = 0.75)
  x <- subset(x,im=im)
  ## Select the time interval where the ensemble is complete.
  it1 <- unlist(lapply(x,function(x) min(index(x[is.finite(x)]))))[-c(1,2,length(x))]
  it2 <- unlist(lapply(x,function(x) max(index(x[is.finite(x)]))))[-c(1,2,length(x))]
  it <- c(max(it1,na.rm=TRUE),min(it2,na.rm=TRUE))
  print(it)
  x <- subset(x,it=it)
  class(x) <- cls
  return(x)
}

esd.results$fw.ssp370 <- allMs(esd.results$fw.ssp370)
```

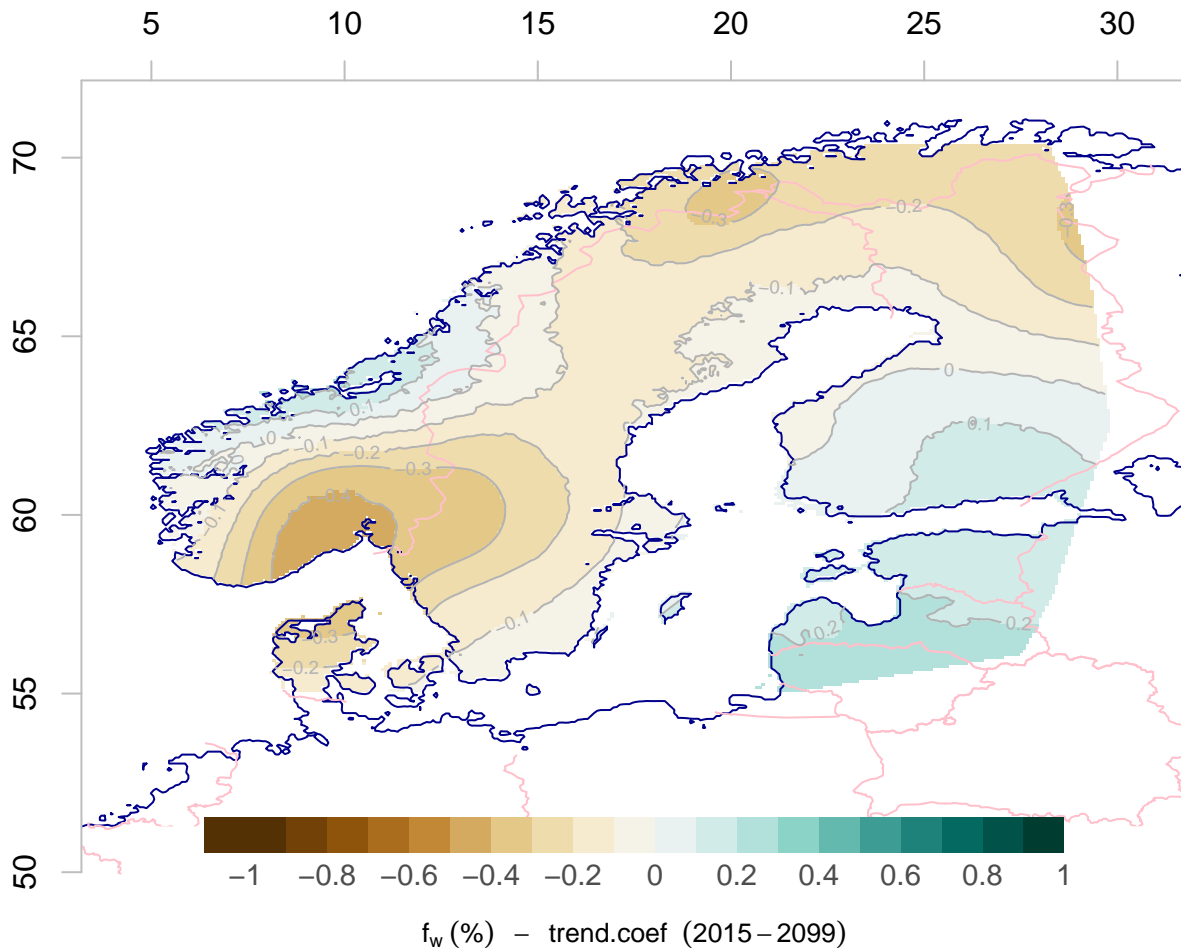
```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
```

```
## Warning in max(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

```r
attr(esd.results$fw.ssp370$eof,'variable') <- 'f[w]'
m.fw.ssp370 <- map(esd.results$fw.ssp370,FUN='mean',new=FALSE)
```

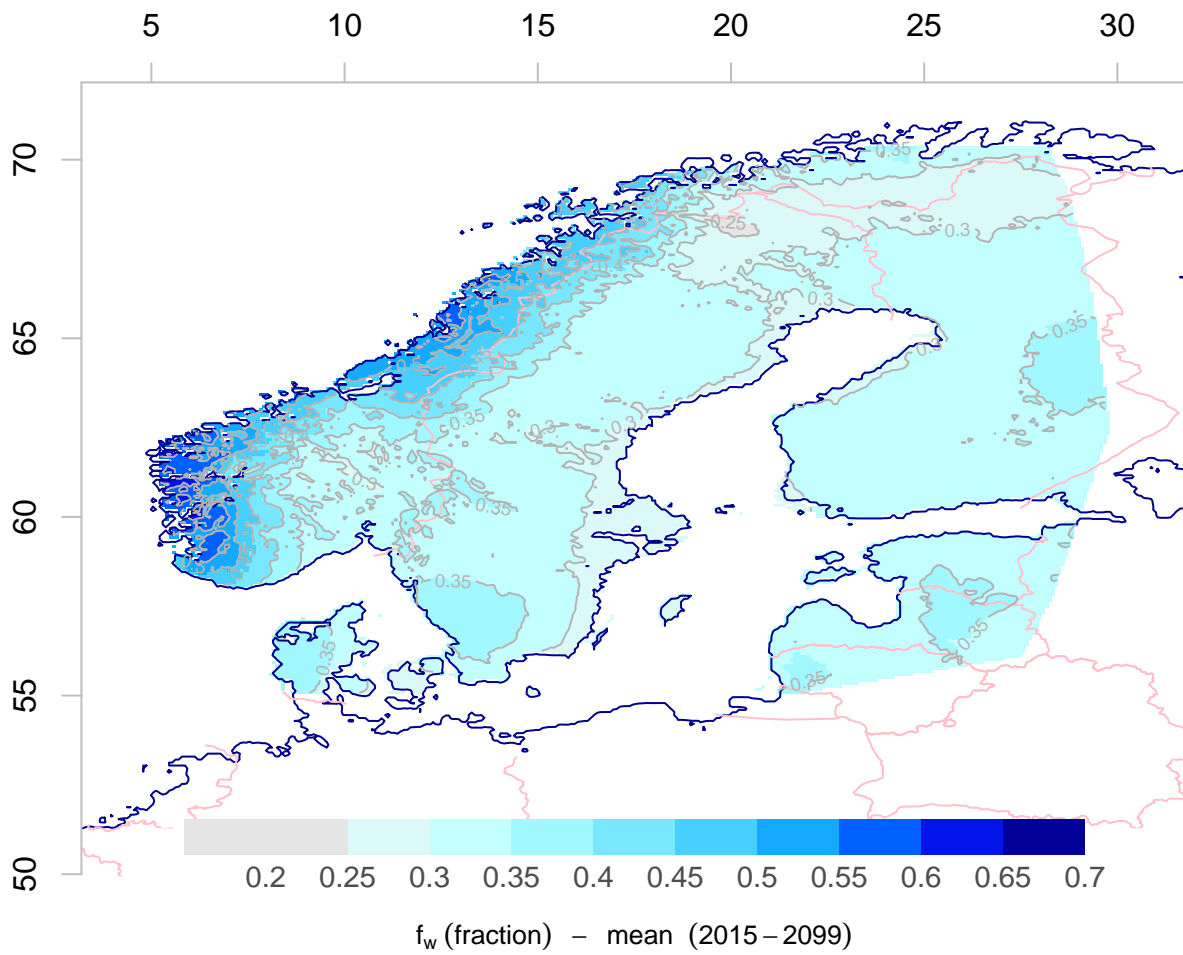
f$_w$ (fraction) – mean (2015 – 2099)

```
d.fw.ssp370 <- ptrend(esd.results$fw.ssp370)
map(d.fw.ssp370,colbar=list(breaks=seq(-1,1,by=0.1),pal='precip.ipcc'),new=FALSE)
```

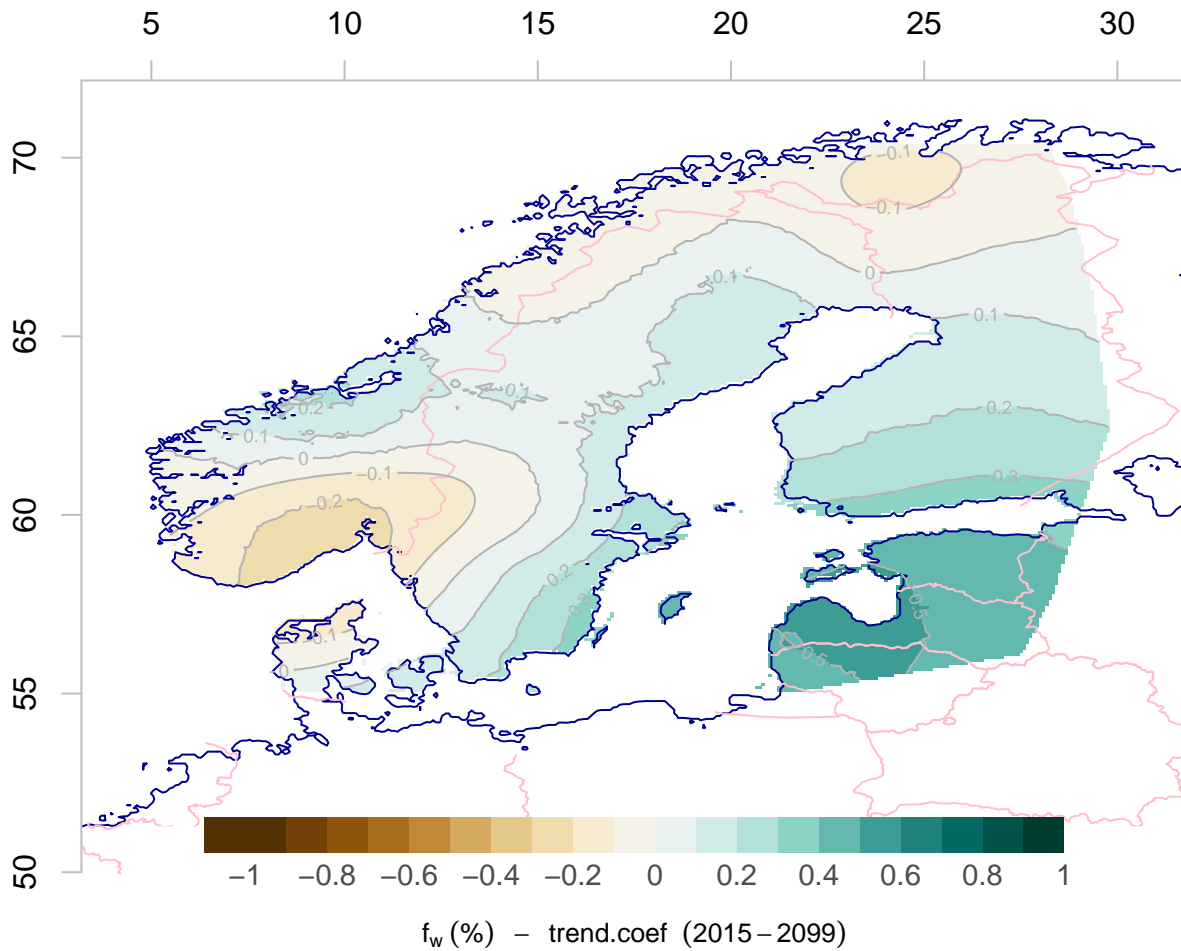
$f_w (\%)$ − trend.coef (2015 − 2099)

```
esd.results$fw.ssp126 <- allMs(esd.results$fw.ssp126)
```

```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

```
attr(esd.results$fw.ssp126$eof,'variable') <- 'f[w]'
m.fw.ssp126 <- map(esd.results$fw.ssp126,FUN='mean',new=FALSE)
```

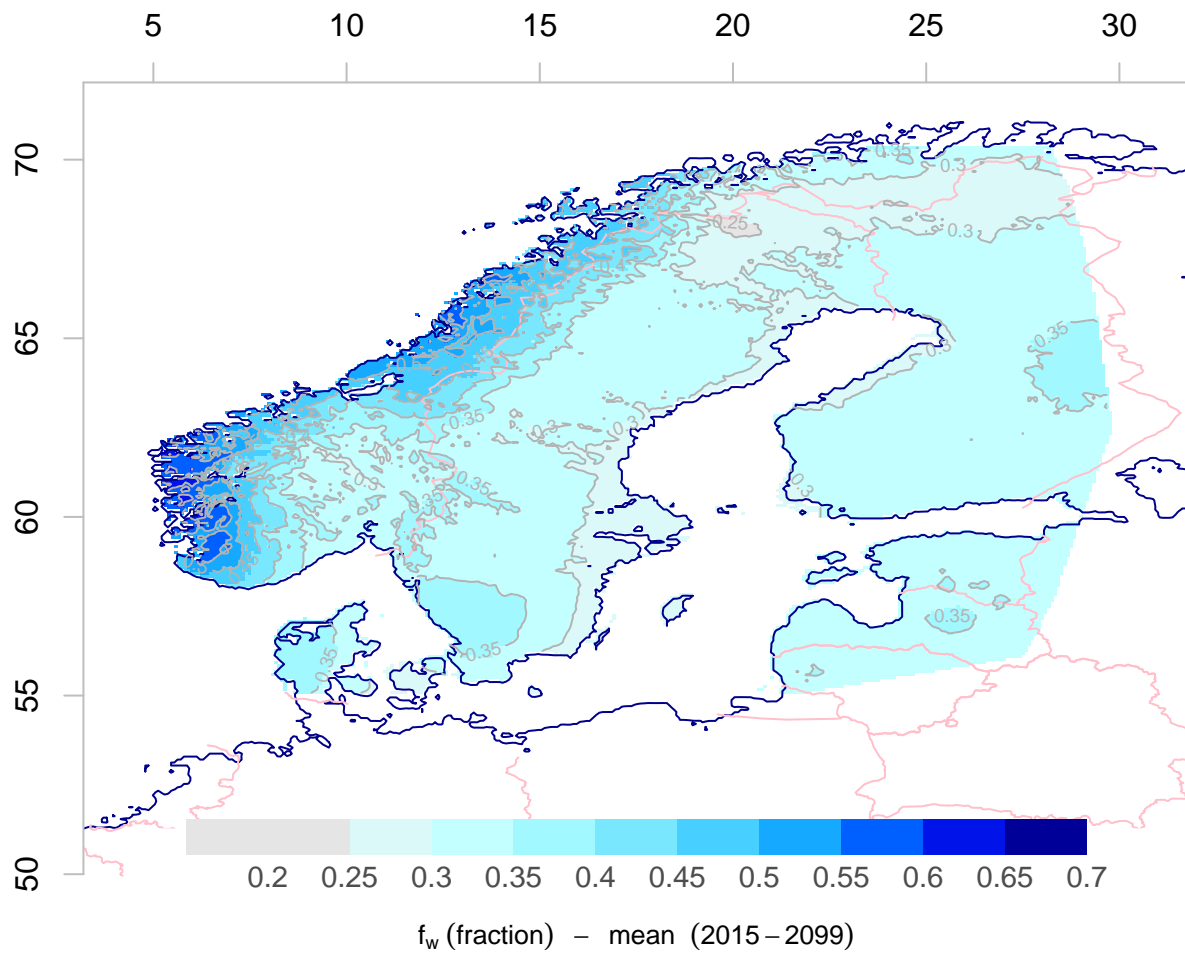
f$_w$ (fraction) − mean (2015 − 2099)

```
d.fw.ssp126 <- ptrend(esd.results$fw.ssp126)
map(d.fw.ssp126,colbar=list(breaks=seq(-1,1,by=0.1),pal='precip.ipcc'),new=FALSE)
```

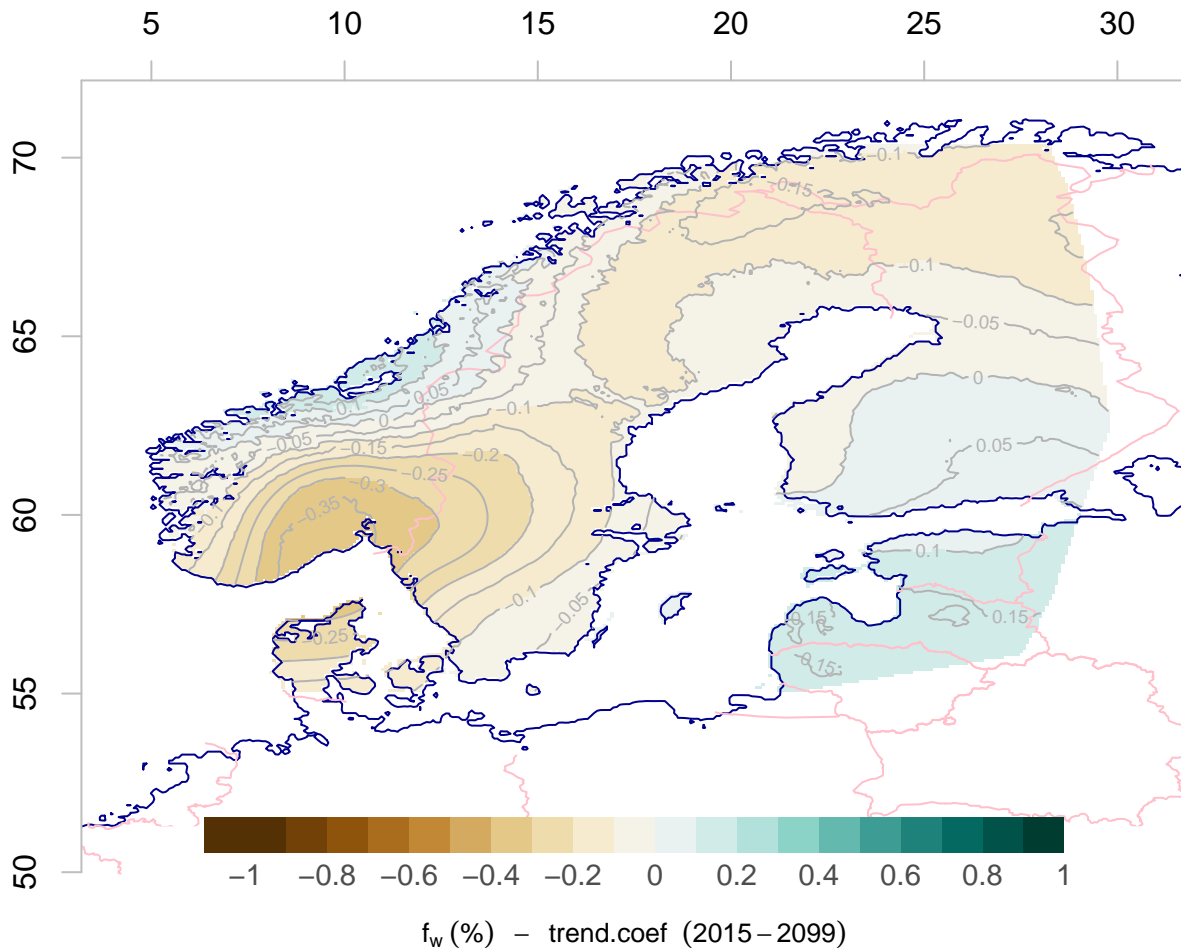
f$_w$ (%) $-$ trend.coef (2015 $-$ 2099)

```
esd.results$fw.ssp245 <- allMs(esd.results$fw.ssp245)
```

```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

```
attr(esd.results$fw.ssp245$eof,'variable') <- 'f[w]'
m.fw.ssp245 <- map(esd.results$fw.ssp245,FUN='mean',new=FALSE)
```

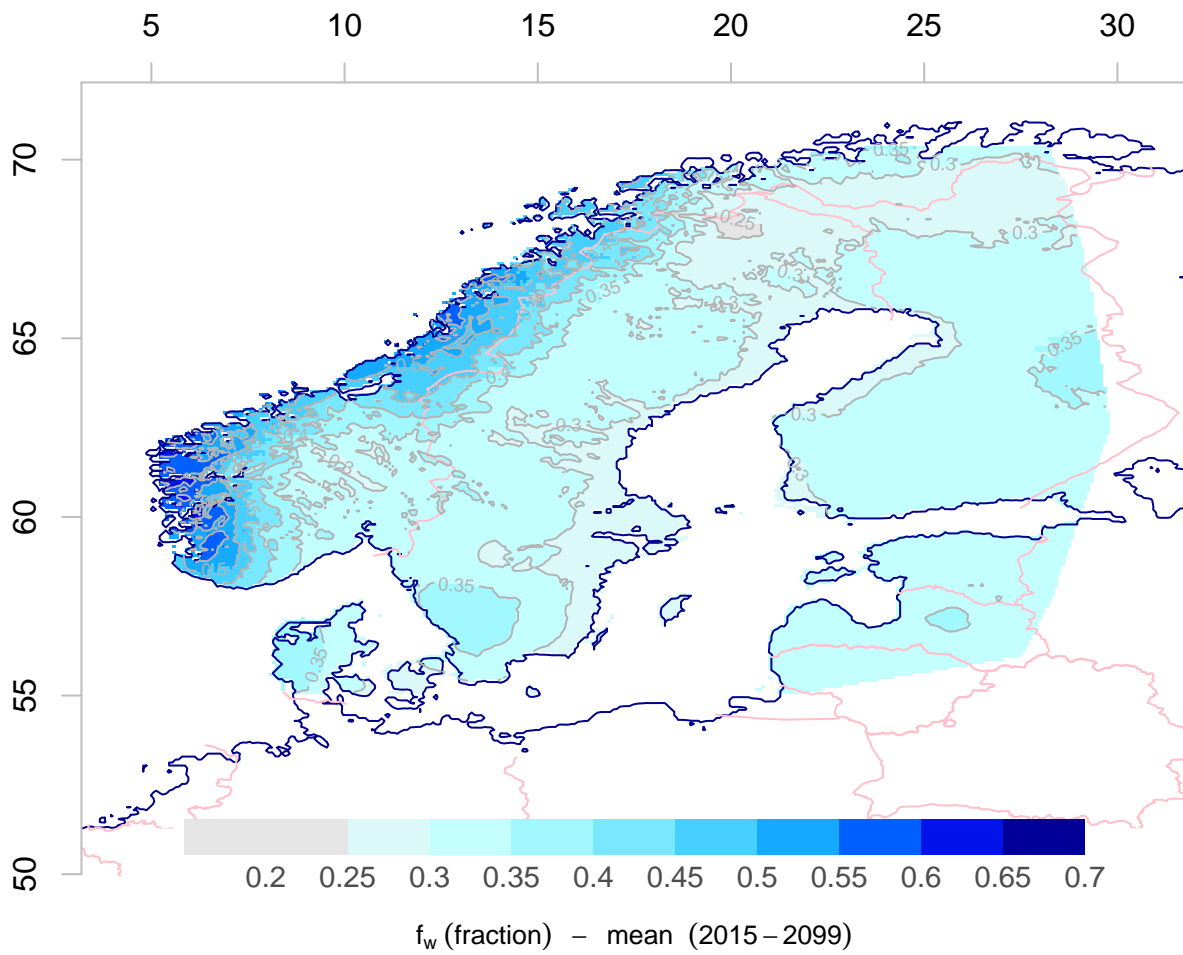
f$_w$ (fraction) – mean (2015 – 2099)

```
d.fw.ssp245 <- ptrend(esd.results$fw.ssp245)
map(d.fw.ssp245,colbar=list(breaks=seq(-1,1,by=0.1),pal='precip.ipcc'),new=FALSE)
```

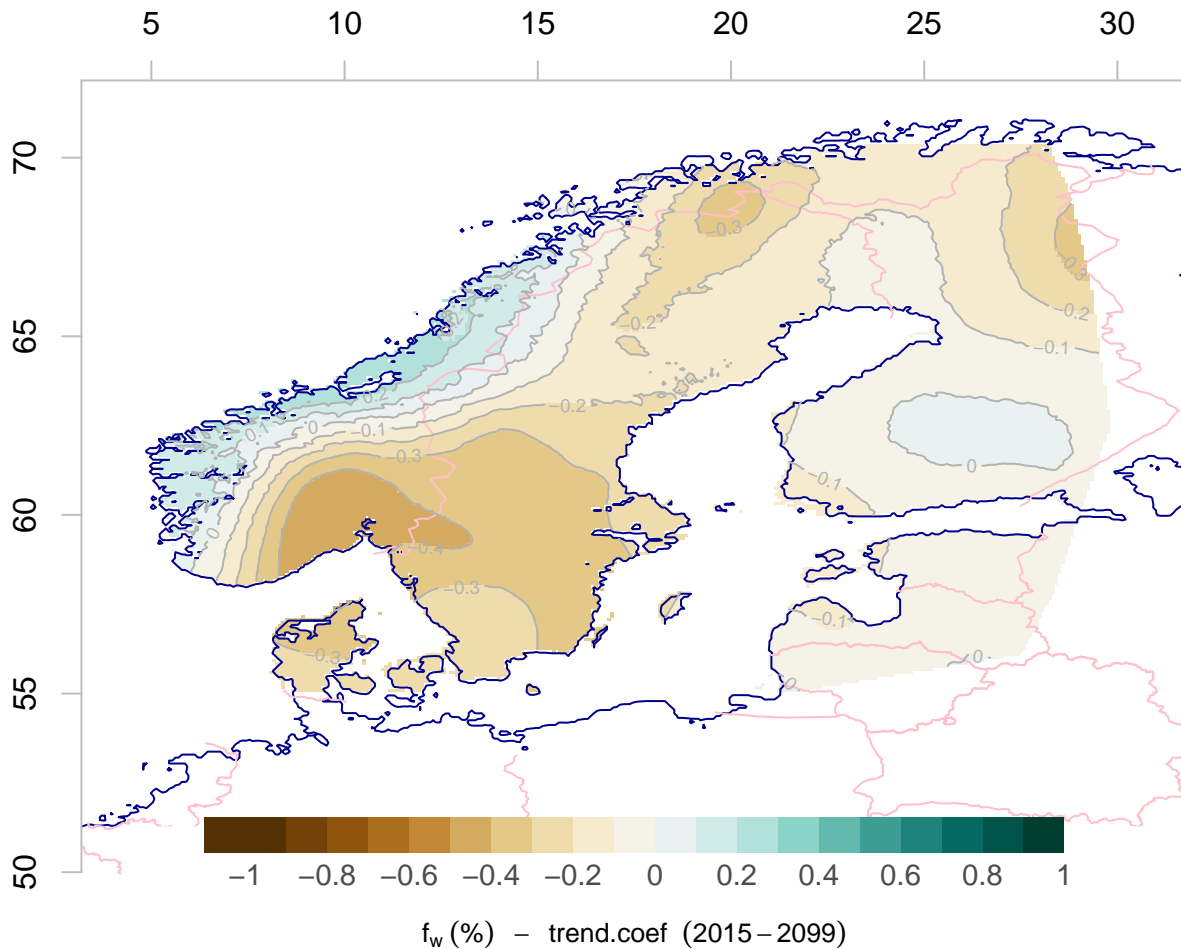
f$_w$ (%) − trend.coef (2015−2099)

```
esd.results$fw.ssp585 <- allMs(esd.results$fw.ssp585)
```

```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

```
attr(esd.results$fw.ssp585$eof,'variable') <- 'f[w]'
m.fw.ssp585 <- map(esd.results$fw.ssp585,FUN='mean',new=FALSE)
```

f_w (fraction) – mean (2015 – 2099)

```
d.fw.ssp585 <- ptrend(esd.results$fw.ssp585)
map(d.fw.ssp585,colbar=list(breaks=seq(-1,1,by=0.1),pal='precip.ipcc'),new=FALSE)
```

f_w (%) – trend.coef (2015 – 2099)

The highest annual frequency of rainy days is near the west coast but the trends are close to zero, the proportional changes are typically less than 1% per decade. The projected proportional trends (% with respect to mean f_w) vary between increasing and decreasing f_w over different regions, however, for SSP370 the projections indicate more reductions rather than increases in f_w, especially over southern Scandinavia.

The following chunk provides corresponding maps for μ, also based on downscaled results for the SSP370 emission scenario.

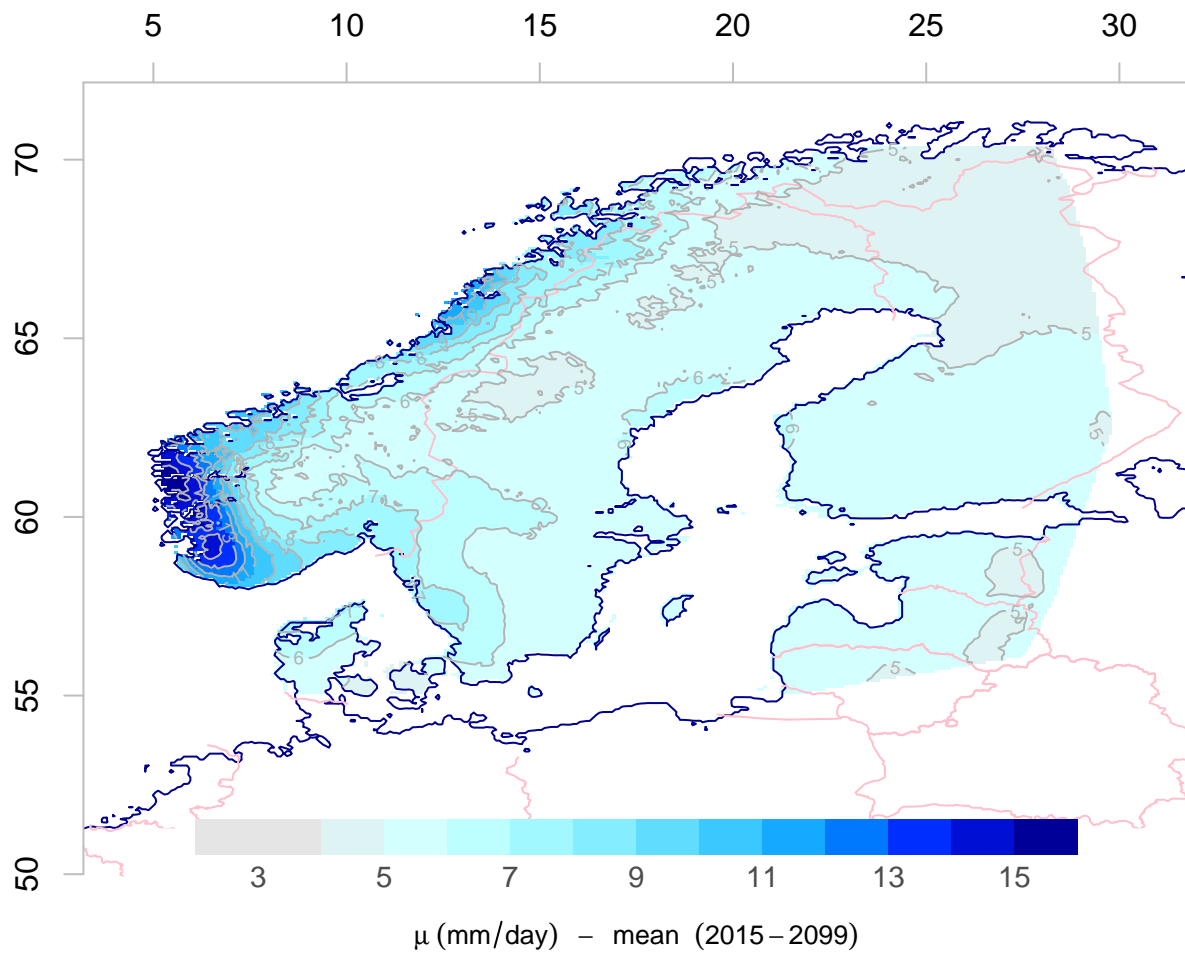
```
esd.results$mu.ssp370 <- allMs(esd.results$mu.ssp370)
```

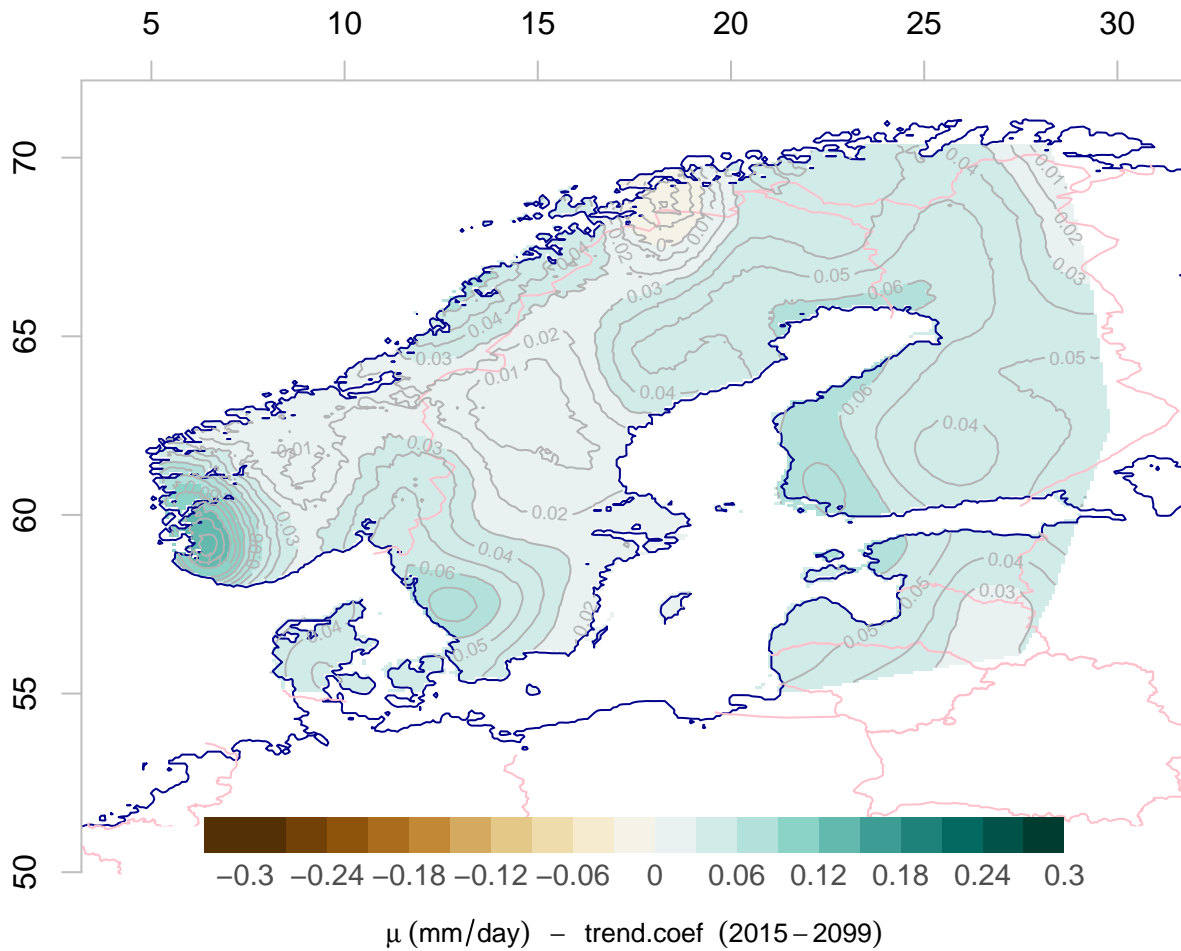
```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
```

```
## Warning in max(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

```
m.mu.ssp370 <- map(esd.results$mu.ssp370,FUN='mean',new=FALSE)
```

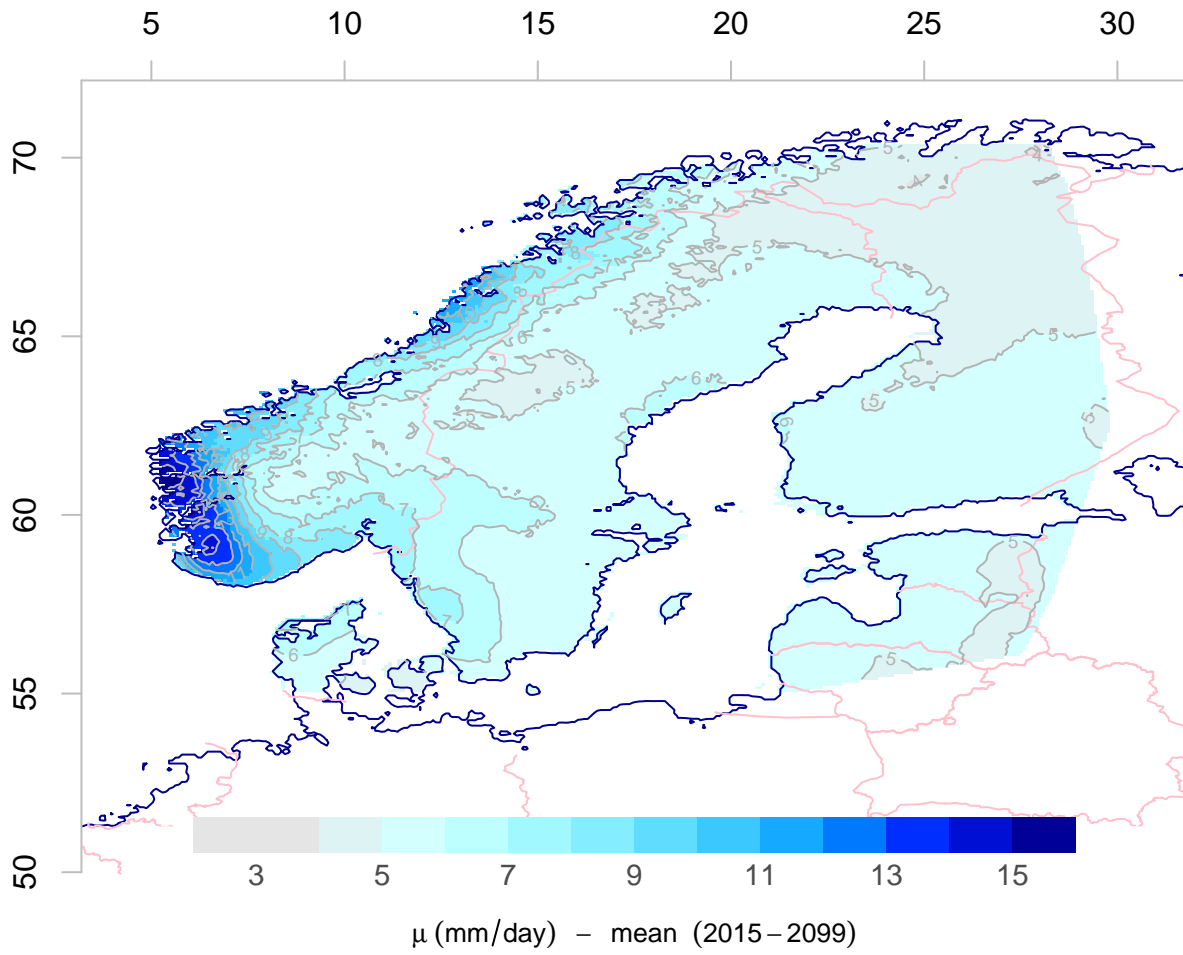
μ (mm/day) − mean (2015−2099)

```
d.mu.ssp370 <- map(esd.results$mu.ssp370,FUN='trend',
                colbar=list(breaks=seq(-0.3,0.3,by=0.03),pal='precip.ipcc'),new=FALSE)
```

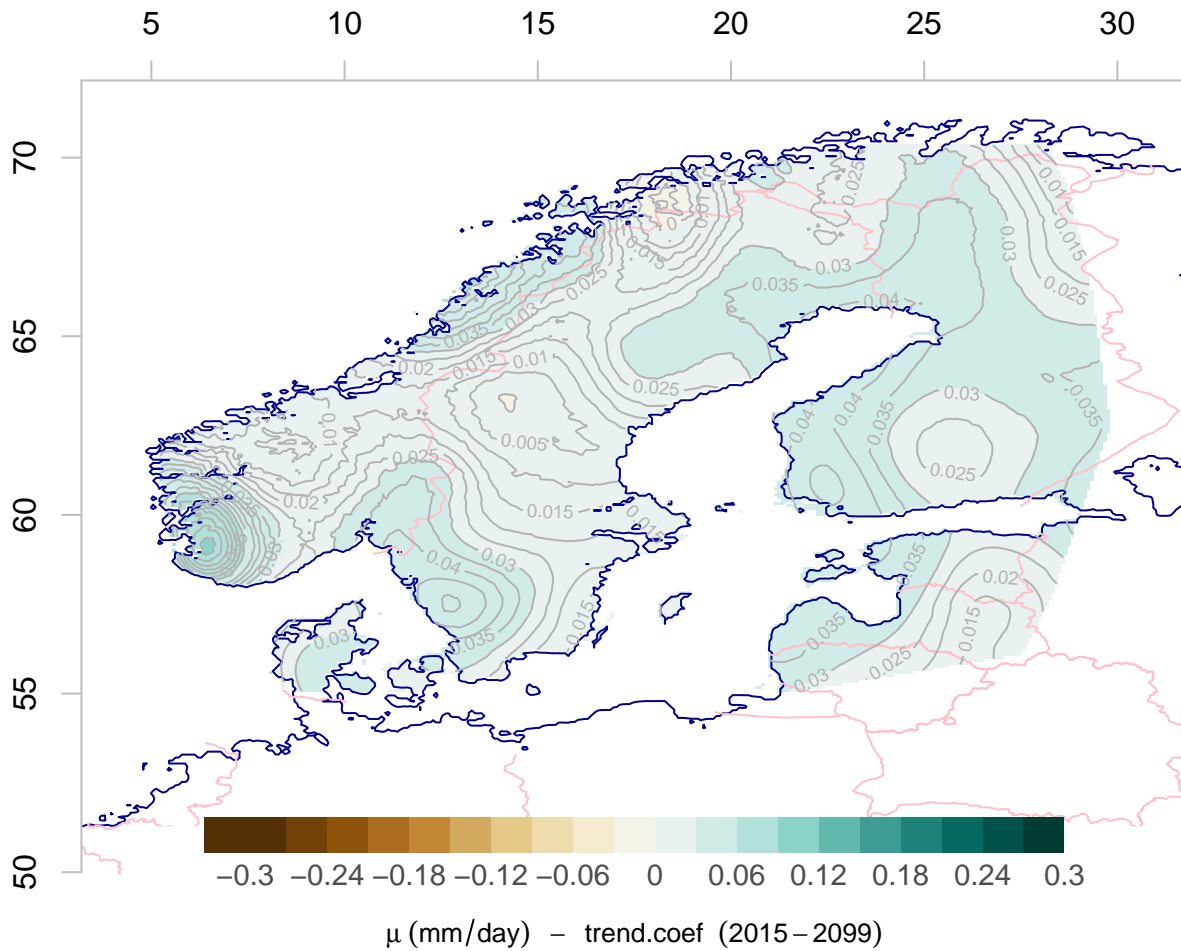
μ (mm/day) − trend.coef (2015−2099)

```
## For some reason, the SSP126 caused errors
#m.mu.ssp126 <- map(esd.results$mu.ssp126,FUN='mean',new=FALSE)
#d.mu.ssp126 <- map(esd.results$mu.ssp126,FUN='trend',new=FALSE)
esd.results$mu.ssp245 <- allMs(esd.results$mu.ssp245)
```

```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

```
m.mu.ssp245 <- map(esd.results$mu.ssp245,FUN='mean',new=FALSE)
```

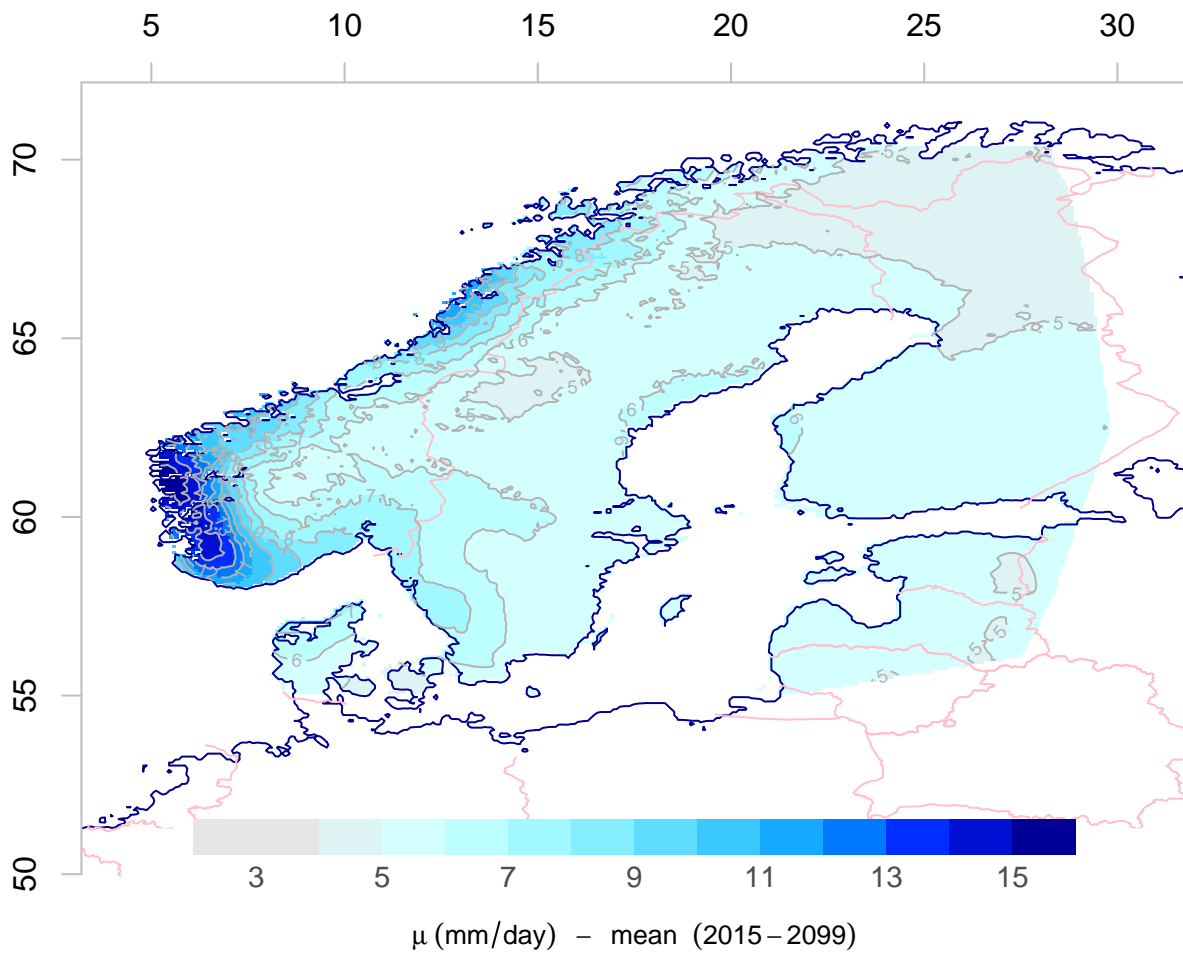
$\mu\,(\text{mm/day})\,-\,\text{mean}\,(2015-2099)$

```
d.mu.ssp245 <- map(esd.results$mu.ssp245,FUN='trend',
                colbar=list(breaks=seq(-0.3,0.3,by=0.03),pal='precip.ipcc'),new=FALSE)
```

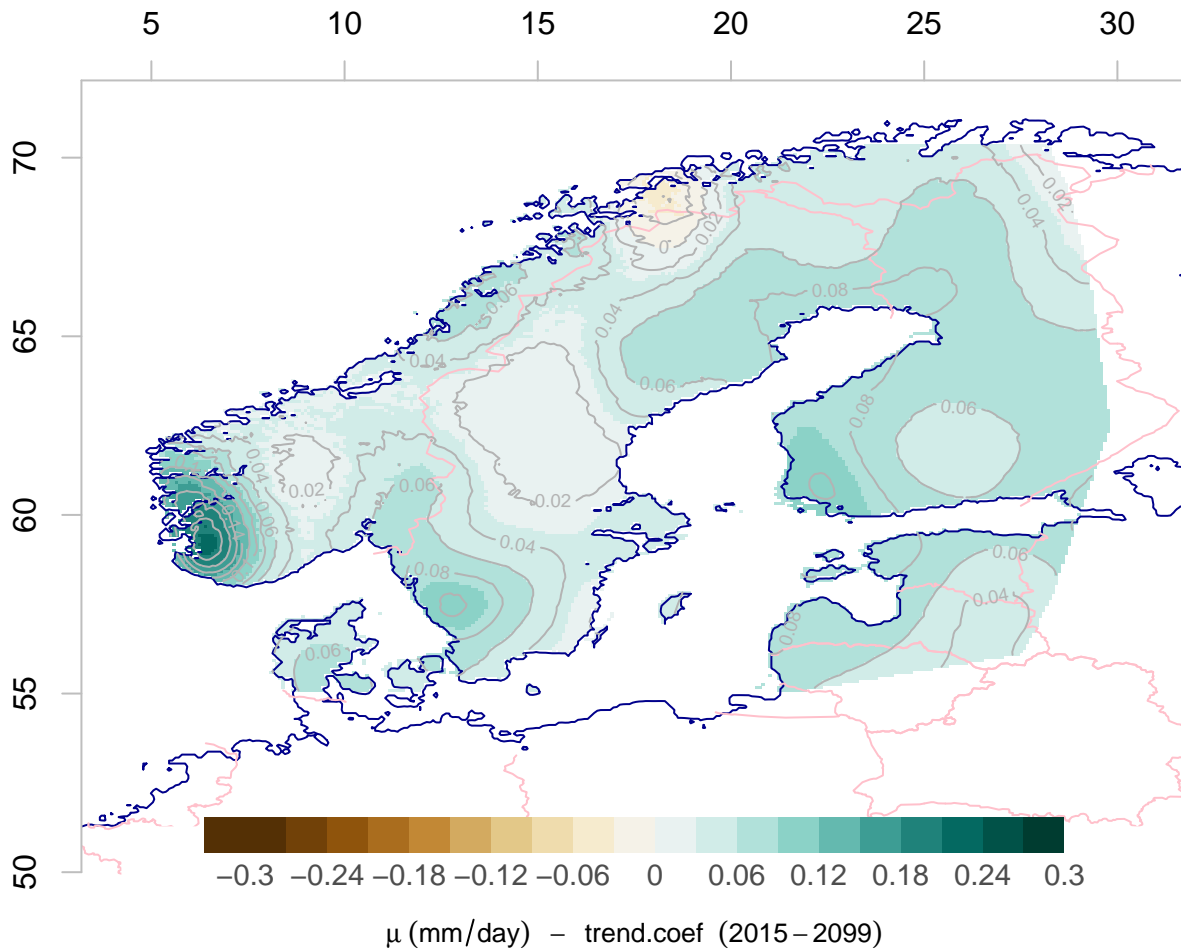
μ (mm/day) − trend.coef (2015 − 2099)

```
esd.results$mu.ssp585 <- allMs(esd.results$mu.ssp585)
```

```
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to min;
## returning Inf
## Warning in min(index(x[is.finite(x)])): no non-missing arguments to max;
## returning -Inf
```

```
## [1] 2015 2100
```

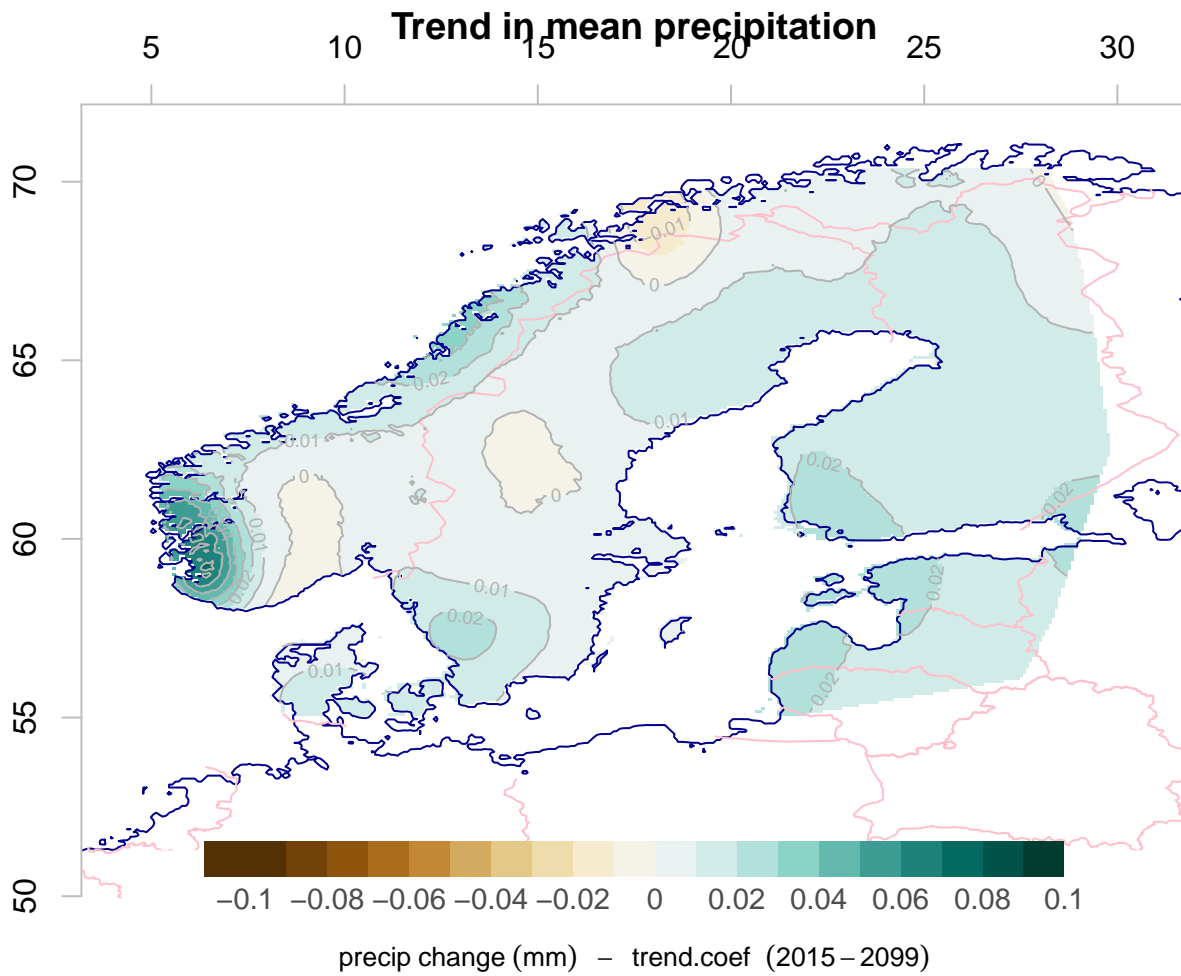
```
m.mu.ssp585 <- map(esd.results$mu.ssp585,FUN='mean',new=FALSE)
```

μ (mm/day) − mean (2015−2099)

```r
d.mu.ssp585 <- map(esd.results$mu.ssp585,FUN='trend',
                   colbar=list(breaks=seq(-0.3,0.3,by=0.03),pal='precip.ipcc'),new=FALSE)
```

μ (mm/day) $-$ trend.coef (2015 $-$ 2099)

The highest intensity is seen near the west coast of Norway and most of the region is projected to see long-term increases in μ. The projected trends for the future suggest increasing μ for most regions, but there are also some exceptions.

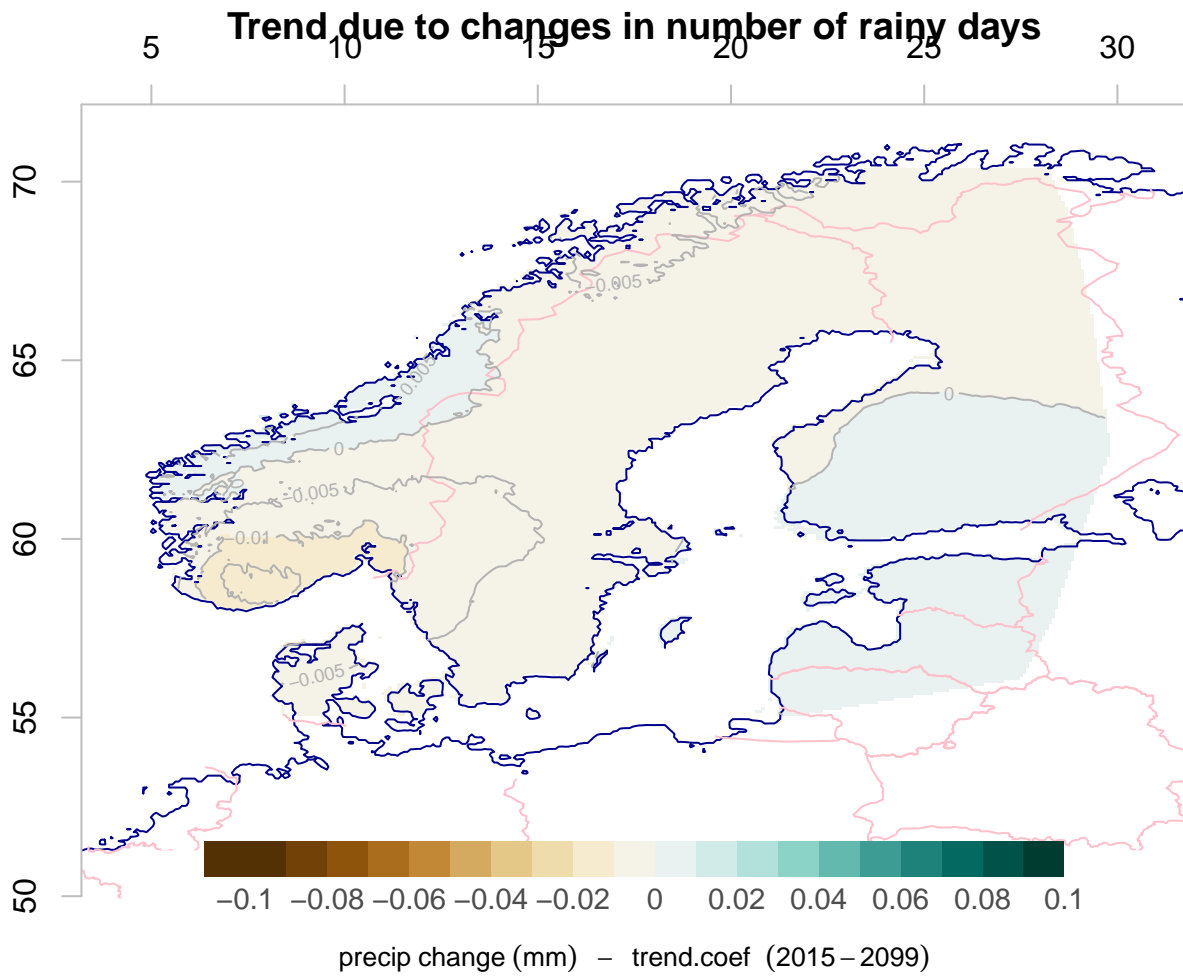
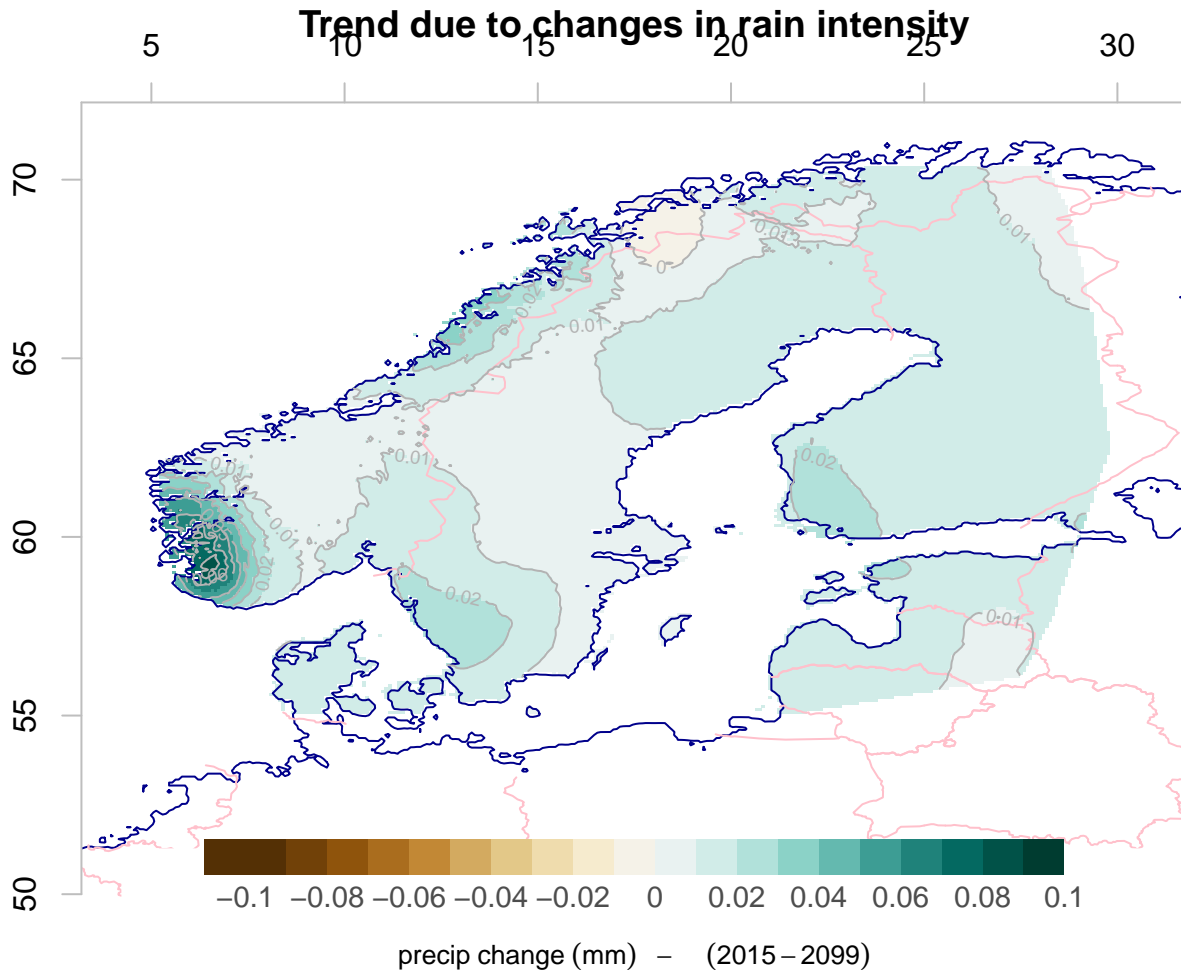
2.9.2 Changes in total precipitation

It is useful to analyse trends in the mean precipitation $\overline{x} = f_w \mu$, as $\delta \overline{X} = \mu \delta f_w + f_w \delta \mu$.

```
## Use the trend in fraction as opposed to the proportional trend in the following
## calculations
d.fw.ssp370 <- map(esd.results$fw.ssp370,FUN='trend',plot=FALSE)
dp1 <- d.fw.ssp370*m.mu.ssp370
attr(dp1,'variable') <- 'precip change'
attr(dp1,'unit') <- 'mm'
dp2 <- m.fw.ssp370*d.mu.ssp370
## Trend in rainfall:
map(dp1+dp2,colbar=list(breaks=seq(-0.1,0.1,by=0.01),pal='precip.ipcc'),
    main='Trend in mean precipitation')
```

```
attr(dp2,'variable') <- 'precip change'
attr(dp2,'unit') <- 'mm'
map(dp1,colbar=list(breaks=seq(-0.1,0.1,by=0.01),pal='precip.ipcc'),
    main='Trend due to changes in number of rainy days')
```

```
map(dp2,colbar=list(breaks=seq(-0.1,0.1,by=0.01),pal='precip.ipcc'),
    main='Trend due to changes in rain intensity')
```

Trend due to changes in rain intensity

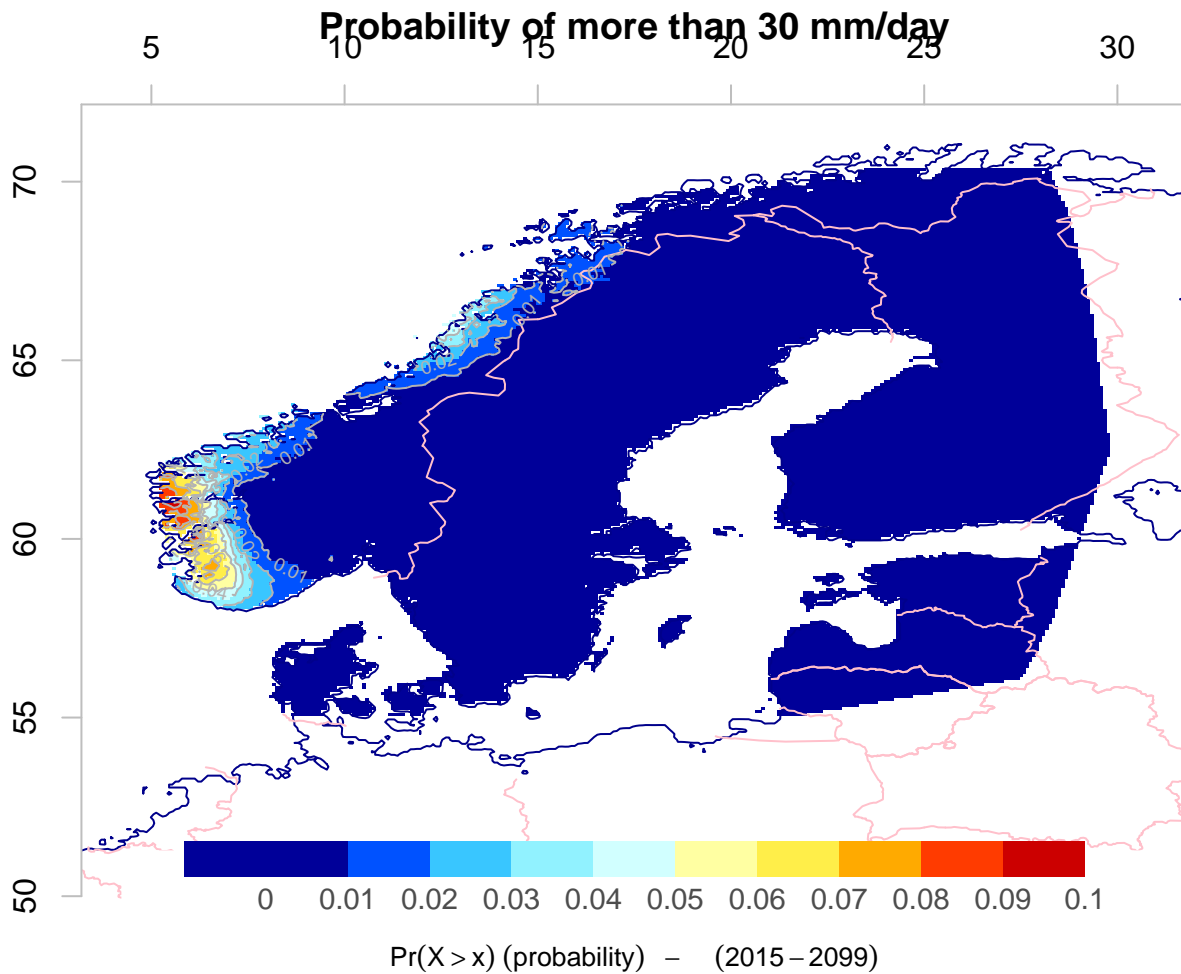
precip change (mm) –   (2015 – 2099)

In some regions, the contributions to the trend in the mean (total) precipitation amounts are highest from changing number of rainy days (dark blue shades), but there are also regions where increased intensity play a bigger role (red regions). This may be a result of a general lack of trend in f_w and where random fluctuations connected to internal variability swings around zero.

2.9.3   Changes in $Pr(X > x)$

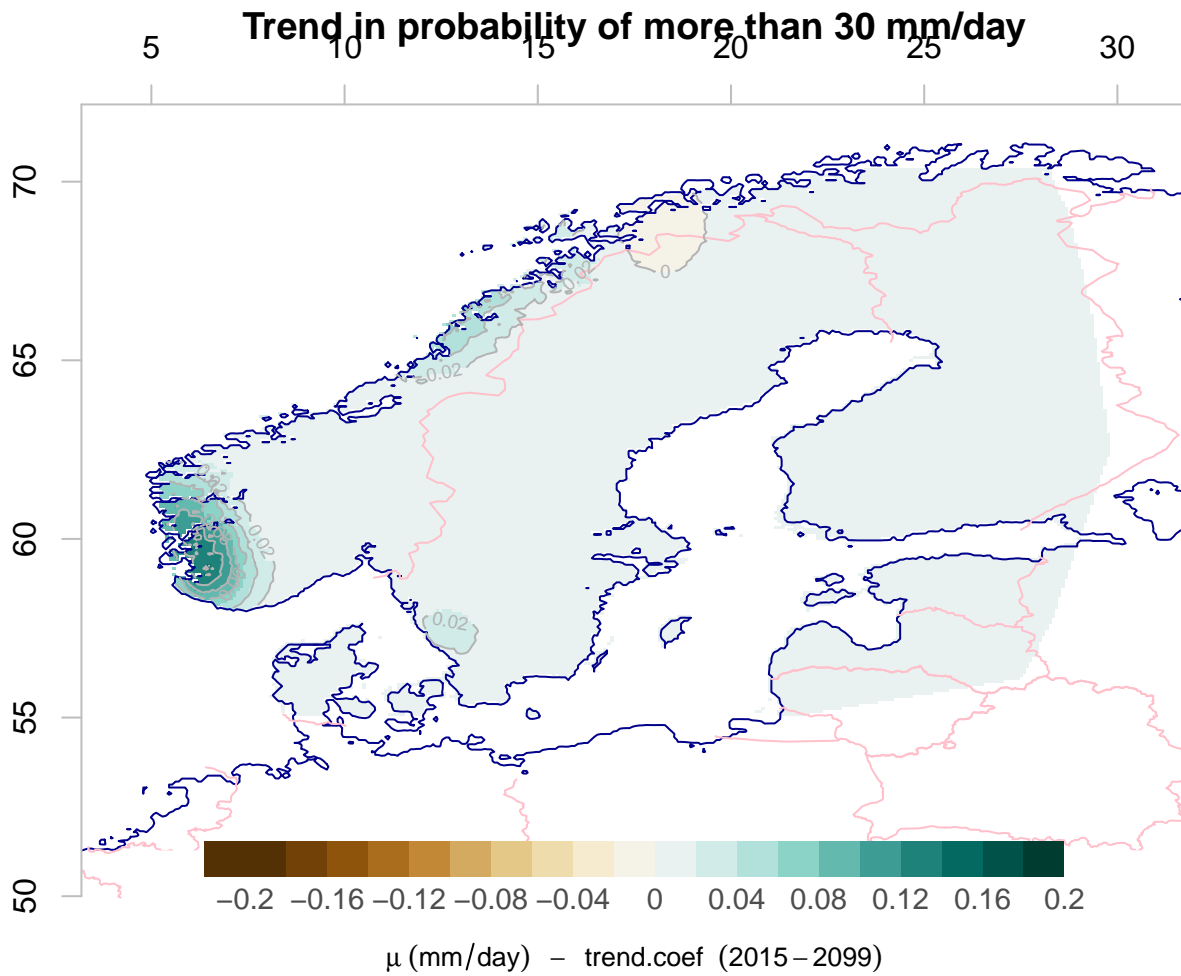
The maps of f_w and μ can also be used to calculate changes in the probability of heavy daily rainfall. We can estimate the contributions to changing probabilities for heavy rainfall from our parameterised equation $Pr(X > x) = f_w exp(-x/\mu)$ through the product rule for differentiation: $dPr(X > x)/dt = df_w/dt exp(-x/\mu) + f_w x/\mu^2 exp(-x/\mu)d\mu/dt$:

```
x <- 30
exp.x <- exp(-x/m.mu.ssp370)
Pr.gt.x <- m.fw.ssp370*exp.x
attr(Pr.gt.x,'variable') <- 'Pr(X>x)'
attr(Pr.gt.x,'unit') <- 'probability'
map(Pr.gt.x,main='Probability of more than 30 mm/day')
```

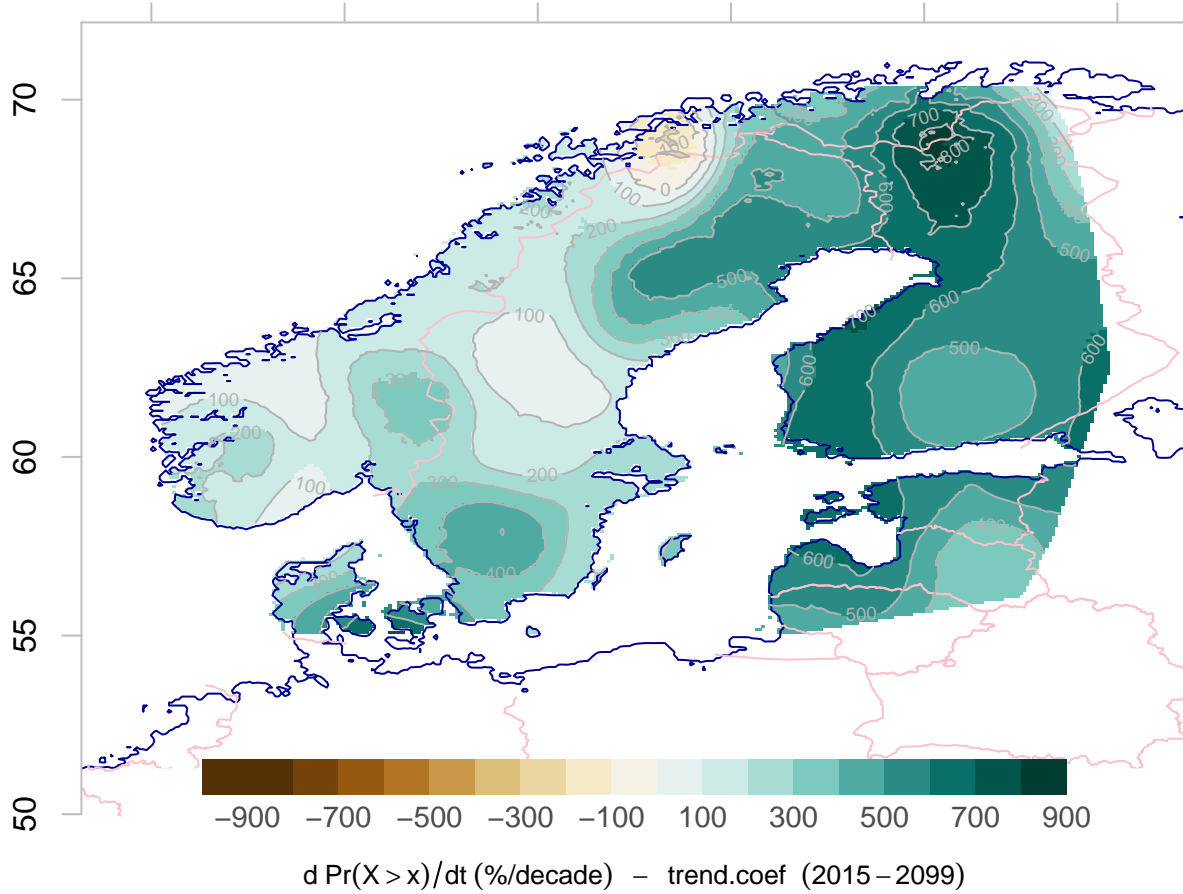
Probability of more than 30 mm/day

Pr(X > x) (probability)   −   (2015 − 2099)

```
dPr.dfw <- exp.x*d.fw.ssp370
dPr.dmu <- x*m.fw.ssp370*exp.x/m.mu.ssp370^2 * d.mu.ssp370
## Trend in probability of daily precipitation exceeding 30 mm/day
dPr.gt.30.ssp370 <- 100*(dPr.dfw + dPr.dmu)
map(dPr.gt.30.ssp370,colbar=list(breaks=seq(-0.2,0.2,by=0.02),pal='precip.ipcc'),
    main='Trend in probability of more than 30 mm/day')
```

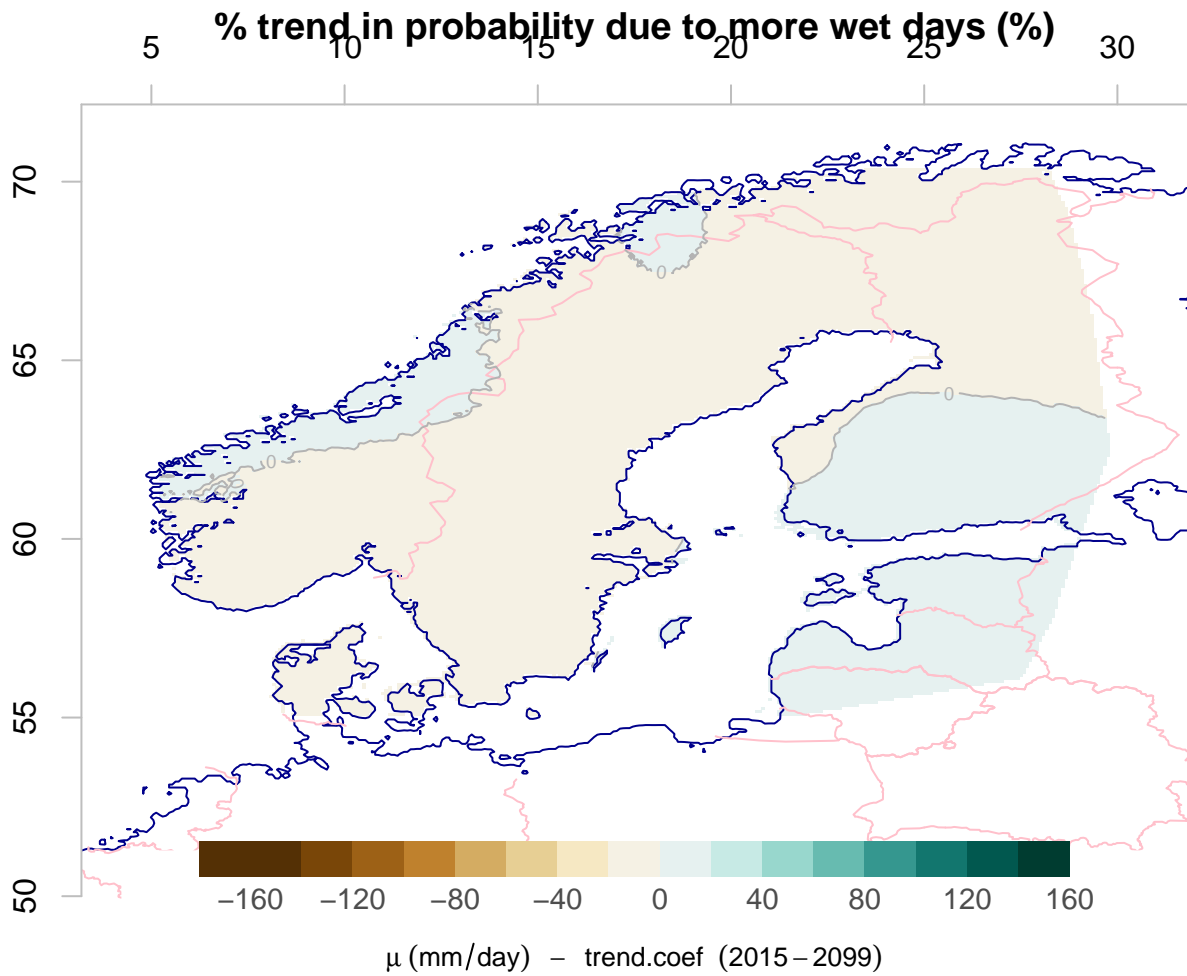
Trend in probability of more than 30 mm/day

μ (mm/day) – trend.coef (2015 – 2099)

```
## Relative change:
ptPr <- 100*dPr.gt.30.ssp370/Pr.gt.x
attr(ptPr,'variable') <- 'd Pr(X>x)/dt'
attr(ptPr,'unit') <- '"%/decade"'
map(ptPr,colbar=list(breaks=seq(-900,900,by=100),pal='precip.ipcc'),
    main='Relative trend in probability of more than 30 mm/day')
```

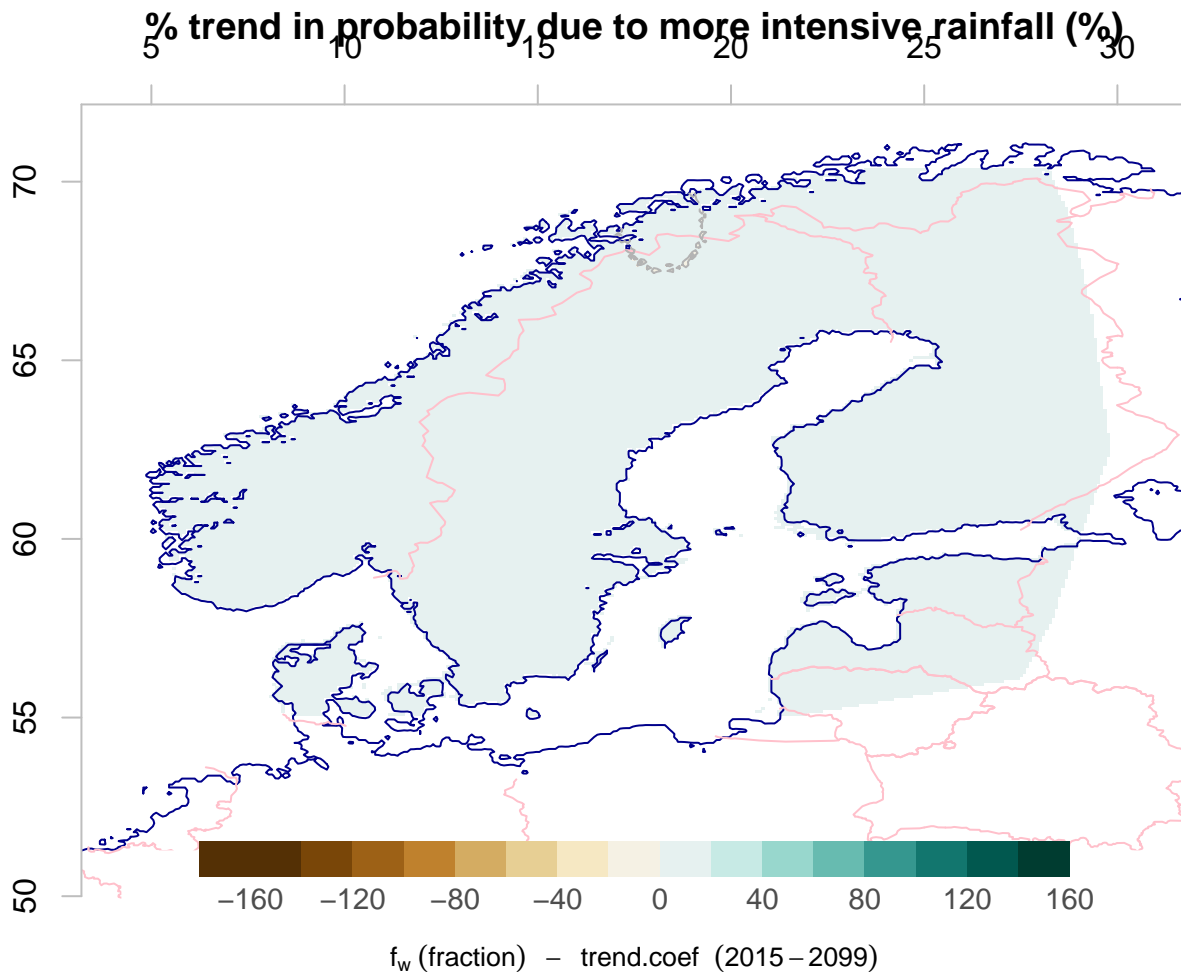
Relative trend in probability of more than 30 mm/day

$$d\,\mathrm{Pr}(X > x)/dt\,(\%/\mathrm{decade}) \quad - \quad \mathrm{trend.coef}\ (2015-2099)$$

```
## Attribution of change to changing number of days or changing
map(100*dPr.dfw/dPr.gt.30.ssp370,colbar=list(breaks=seq(-160,160,by=20),pal='precip.ipcc'),
    main='% trend in probability due to more wet days (%)')
```

% trend in probability due to more wet days (%)

μ (mm/day) − trend.coef (2015 − 2099)

```
map(100*dPr.dmu/dPr.gt.30.ssp370,colbar=list(breaks=seq(-160,160,by=20),pal='precip.ipcc'),
    main='% trend in probability due to more intensive rainfall (%)')
```

% trend in probability due to more intensive rainfall (%)

f_w (fraction) – trend.coef (2015 – 2099)

The increase in μ implies an increase in the probability of days with more than 30 mm/day by 0-20% relative to the original probability. The strongest trend was found in northern Finland.

2.10   Return values

Return values refer to the typical maximum amount of precipitation that can be expected over a time interval of τ years.

Here, 1-year, 10-year and 25-year return values were calculated using the approximate expression $x_\tau = \alpha\mu\ln(f_w\tau)$, based on derivations in Benestad et al. (2019). The corresponding rate of change (trend) was calculated based on $x_\tau = \alpha\ln(f_w\tau)\delta\mu + \alpha\mu\ln(f_w\tau)$, taking $d/dt\ln(f_w\tau) = d\ln(u)/du * du/dt = 1/u\,du/dt = 1/f_w * df_w/dt$.

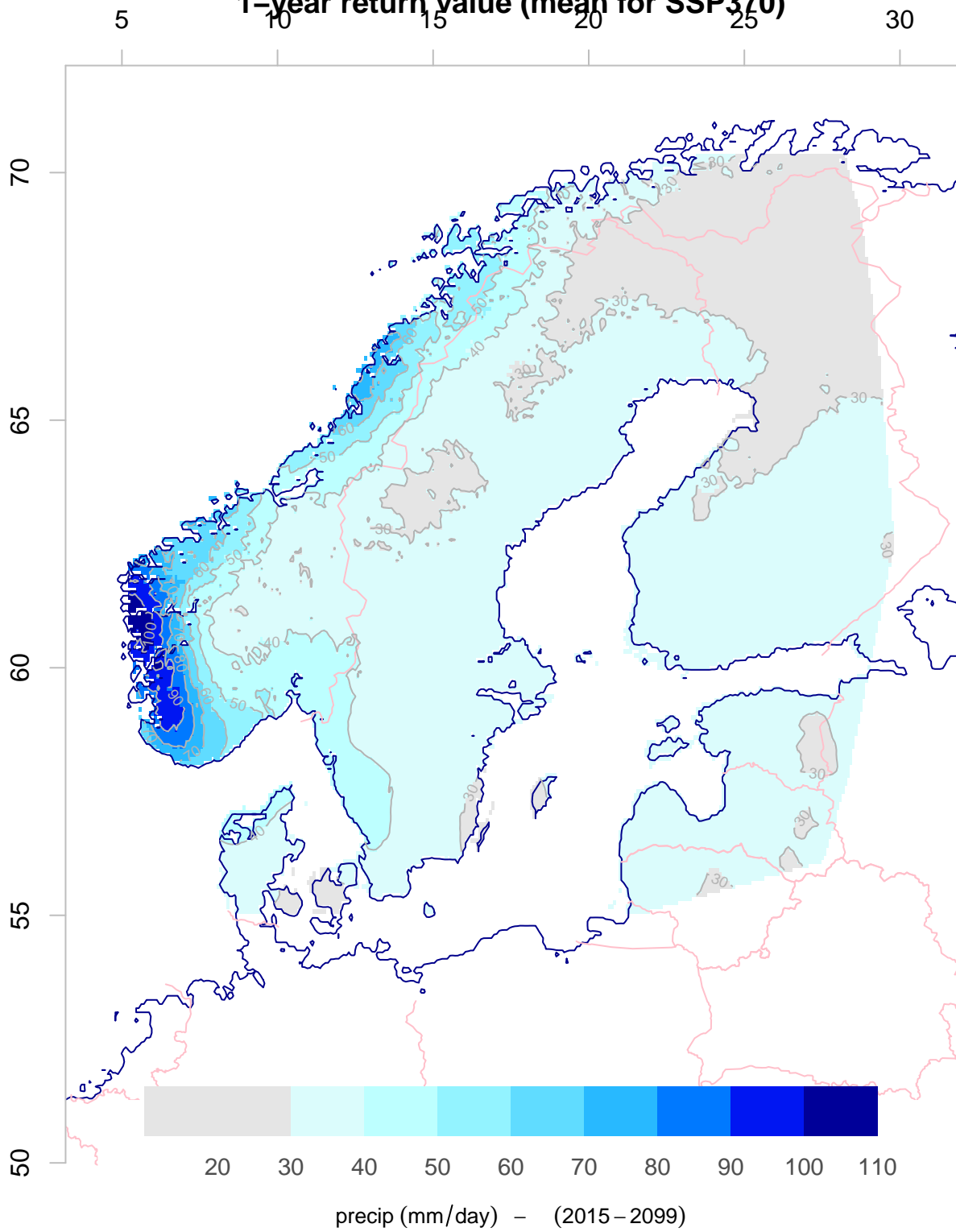
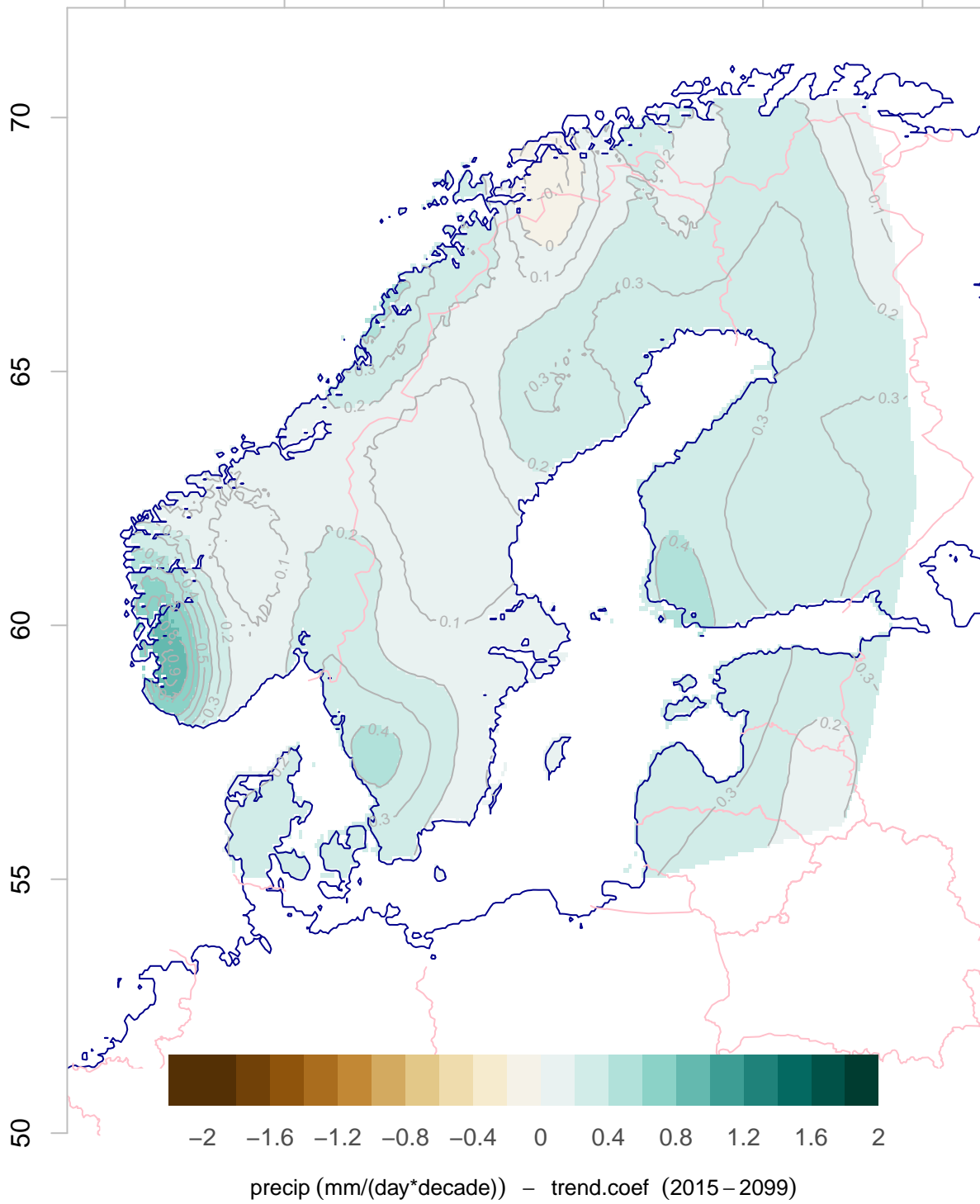
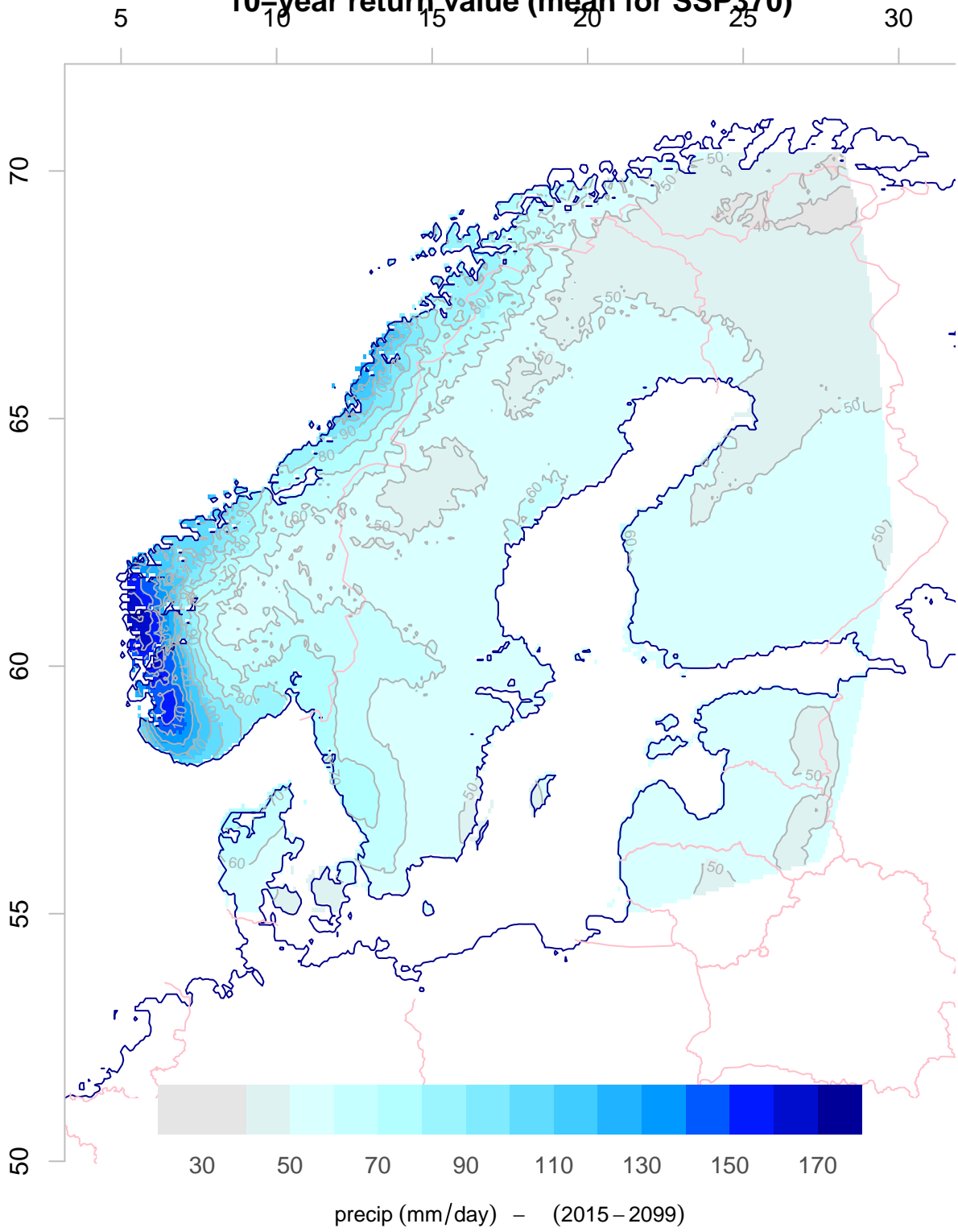
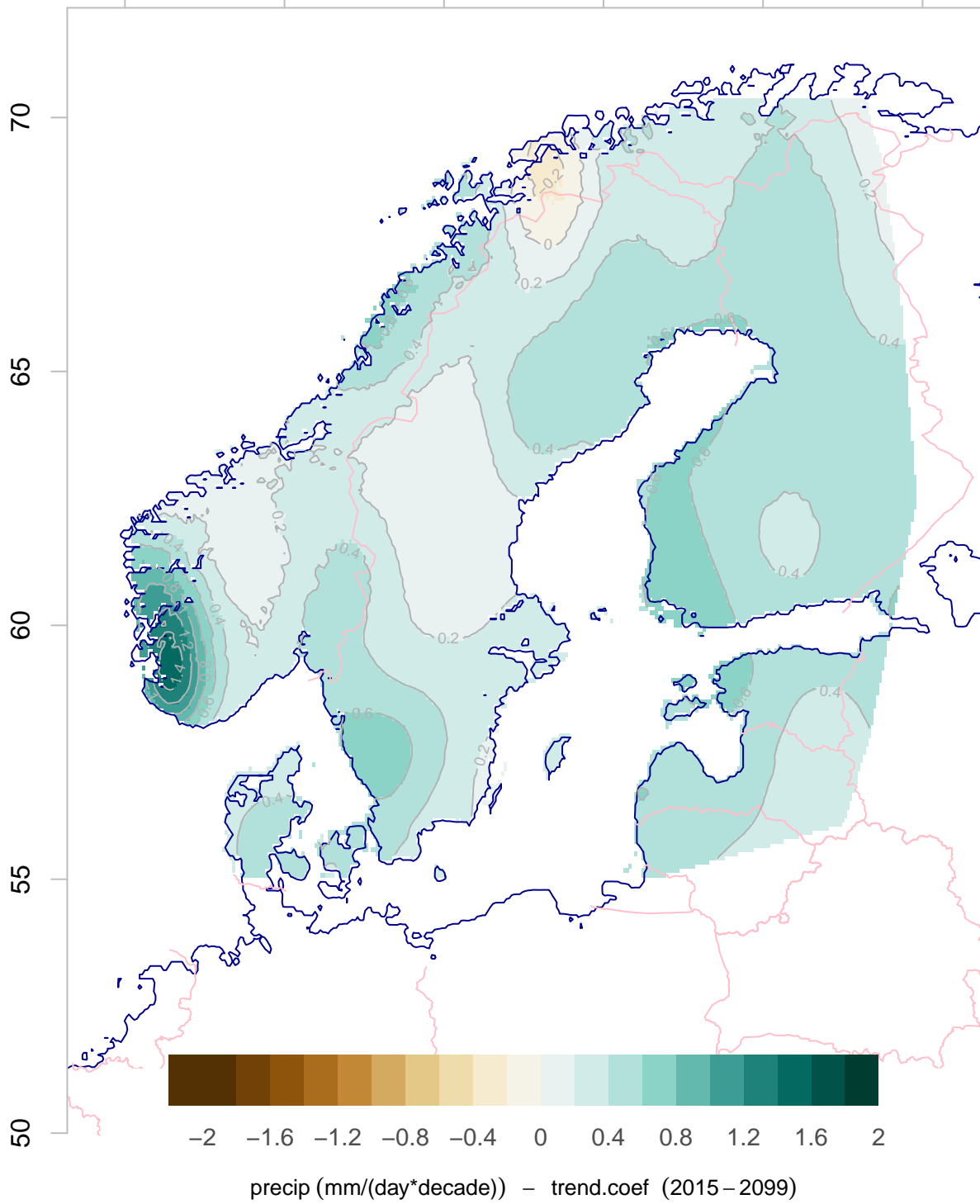
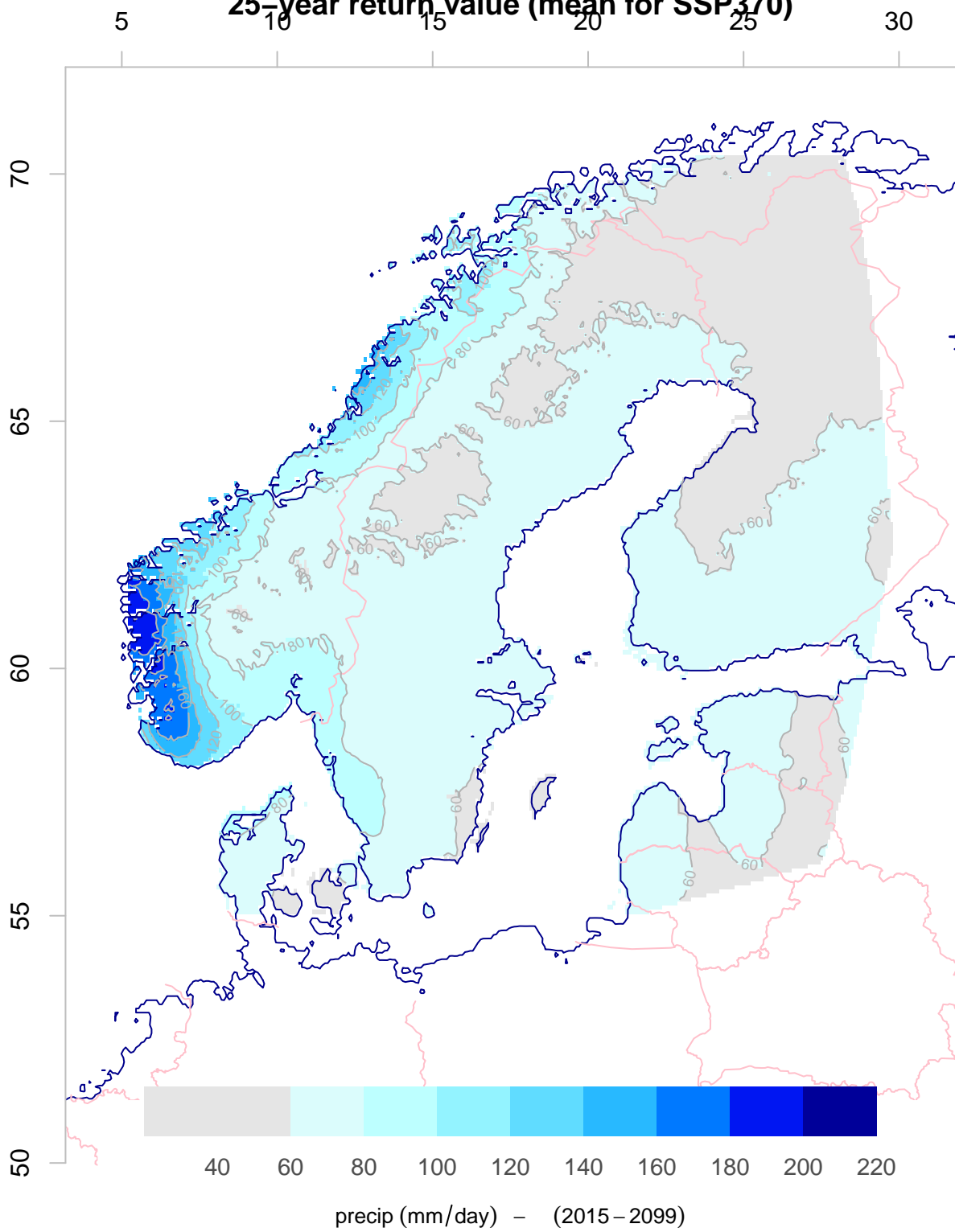
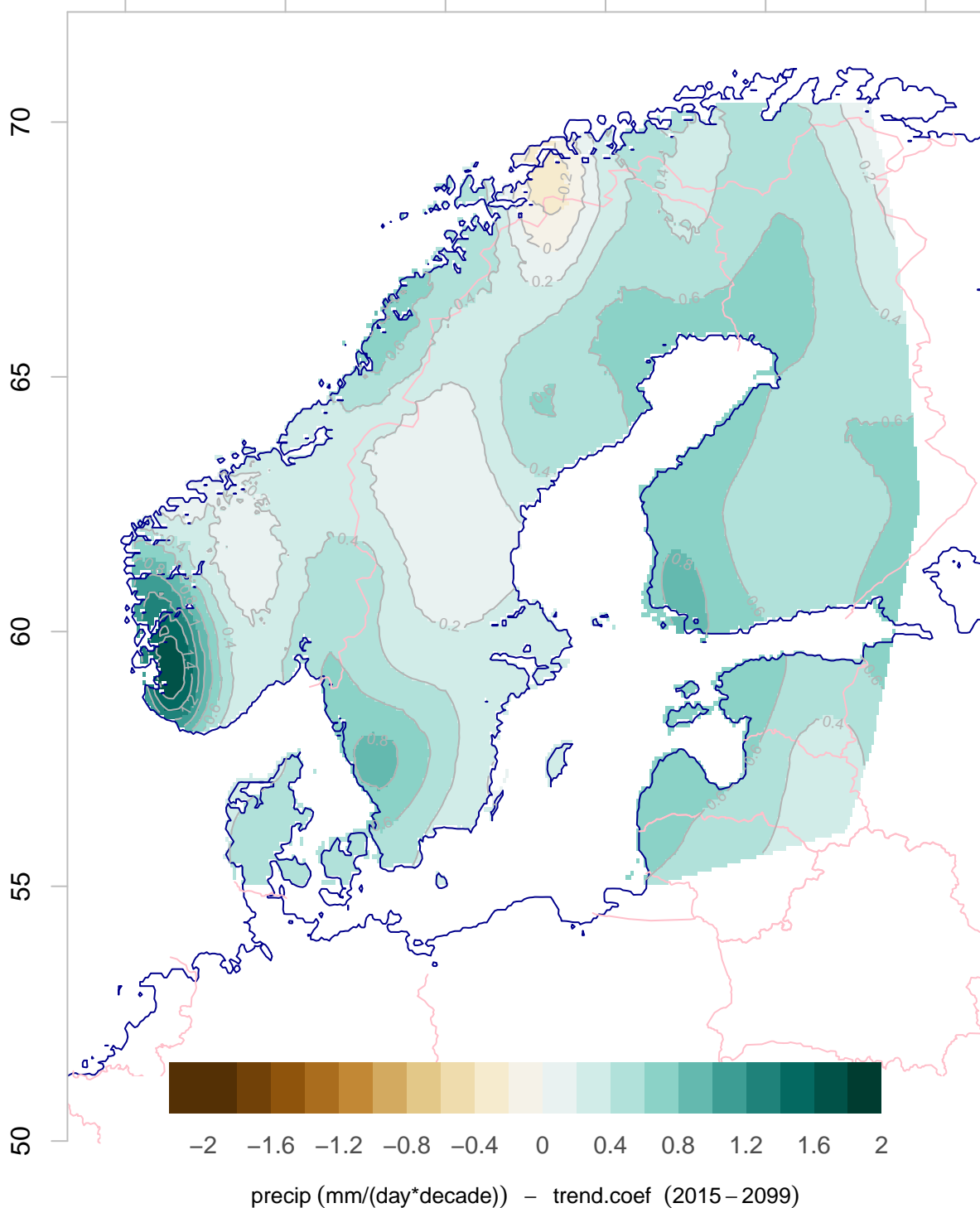
```r
## The coefficient alpha is log-linear
c.alpha = c(1.256, 0.064)
layout(matrix(1,1,1))
## tau is the return-interval in number of years
for (tau  in c(1, 10, 25)) {
  ## Estimate the value for alpha for given return-interval
  alpha <- c.alpha[1] + c.alpha[2] * log(tau)
  ## Estimate the mean return-value based on the above expression:
  x.L <- alpha * m.mu.ssp370 * log(m.fw.ssp370 * tau * 365.25)
  attr(x.L,'variable') <- 'precip'
```

```r
  attr(x.L,'unit') <- 'mm/day'
  map(x.L,main=paste0(tau,'-year return value (mean for SSP370)'),new=FALSE)
  ## Estimate change
  d.x.L <- alpha * d.mu.ssp370 * log(m.fw.ssp370 * tau * 365.25) +
    alpha * m.mu.ssp370 * d.fw.ssp370/m.fw.ssp370
  attr(d.x.L,'variable') <- 'precip'
  attr(d.x.L,'unit') <- '"mm/(day*decade)"'
  ## The Knitting doesn't get these figures quite right
  par(new=FALSE,mfcol=c(1,1))
  map(d.x.L,colbar=list(breaks=seq(-2,2,by=0.2),pal='precip.ipcc'),
      main=paste0('Trend in the ',tau,'-year return value (mean for SSP370)'),new=FALSE)
}
```

precip (mm/(day*decade)) – trend.coef (2015 – 2099)

25–year return value (mean for SSP370)

precip (mm/day) – (2015 – 2099)

precip (mm/(day*decade)) – trend.coef (2015 – 2099)

The spatial patterns of the mean 1-year, 10-year, and 25-year return values simulated for the 2015-2099 period and estimated on the basis of f_w and μ (SSP370) are very similar, but their magnitudes vary between ~100 mm, ~150 mm, and ~200 mm. The greatest magnitudes are seen along the west coast of Norway and the smallest magnitudes in over Finland.

3 Evalutation of GCMs ability to reproduce large-scale f_w and μ

3.1 Evaluation of global climate models

One assumption behind downscaling is to utilise large-scale features that the global climate models (GCMs) can skilfully reproduce and the dependency of local scale climate upon these. Hence it's crucial to evaluate the GCM output to establish whether they skillfully reproduce real features, and in particular, it is important to evaluate how well the GCMs reproduce large-scale climate change and variability before using them as a basis for climate change adaptation. ESD can include some degree of evaluation, e.g. through the use of common empirical orthogonal functions, henceforth referred to as "common EOFs" (Benestad (2001)). This concept is explained more recently in Benestad (2021). Here, the acronym 'EOF' refers to 'empirical orthogonal functions' which are similar to principal component analysis and closely connected to the mathematical concepts known as eigenfunctions and eigenvalues.

Below is an analysis that provides an objective and deep evaluation of the predictors used to represent the large-scale conditions in downscaling as they are simulated by the GCMs, focusing on the most salient information embedded in the data. It goes beyond the framework suggested in Benestad (2001) by comparing large multi-model ensembles with a focus on the the ensemble's ability to reproduce physical conditions such as the mean annual cycle, interannual variability and historical long-term trends, rather than emphasising individual members.

Another objective is to compare the evaluation of two generations of GCM simulations: CMIP5 and CMIP6. Here is a demonstration of how such a comparison can make use of the models' ability to reproduce the mean annual cycle, the interannual variability and historical trends over 1959-2021 in this case. The 63-year long period 1959-2021 is chosen because monthly ERA5 data are available in a single data set from 1950 until present time and because 2021 was the last year with complete data for all months at the time of the analysis.

More information about EOFs and common EOFs are available from https://doi.org/10.1142/6908 and a short hands-on demonstration is available ads a YouTube https://youtu.be/32mtHHAoq6k.

The evaluation of GCM results based on common EOFs is implemented with the help of the R-package `esd` and similar to the analysis presented in Benestad et al. (2023). The following chunk activates it and sets the working environment

3.1.1 Special functions

The following code chunk defines a number of functions used in this analysis:

```r
## Function to split the CMIP file names into model, ssp and RIPF
decipher <- function(x) {
  i <- gregexpr('_',x)[[1]]
  model <- substr(x,i[2]+1,i[3]-1)
  ssp <- substr(x,i[3]+1,i[4]-1)
  ripf <- substr(x,i[4]+1,i[4]+8)
  interval <- substr(x,i[5]+1,regexpr('.nc',x)[[1]]-1)
  return(c(model,ssp,ripf,interval))
}

## Method for calculating common EOFs of large multi-model ensembles with GCMs'
## mean annual cycle.  Most of its operations involve preparation of the data and
## collecting the right data from multiple data files.
EOF.GCMsAC <- function(x,masked=FALSE,lon=NULL,lat=NULL,verbose=FALSE,...) {
  ## This function combines the annual cycles from different GCMs and performs a
  ## common EOF on the joint data
```

```r
  print('EOF.GCMsAC')
  runs <- names(x)
  runs <- runs[-grep('era5',runs)]
  #print(runs)
  if (length(grep('era5',names(x)))>0) X <- subset(x[['era5']],is=list(lon=lon,lat=lat))
  if (masked) X <- mask(X,land=FALSE)
  X <- regrid(X,is=list(lon=seq(min(lon(X)),max(lon(X)),by=1),
                        lat=seq(min(lat(X)),max(lat(X)),by=1)))
  n <- length(runs)
  for (i in 1:n) {
    if (verbose) print(runs[i])
    z <- subset(x[[runs[i]]],is=list(lon=lon,lat=lat))
    if (verbose) map(z)
    X <- combine(X,z)
  }
  ## < Debugging >
  #if (verbose) {print(str(X,max.lev=1)); print(names(attributes(X)))}
  #for (j in 2:attr(X,'n.apps')) attr(X,paste0('appendix.',j)) <- NULL
  #attr(X,'n.apps') <- 1
  ## > Debugging <
  eof <- EOF(X,anomaly=FALSE,verbose=verbose,...)
  invisible(eof)
}

## Method for calculating common EOFs of the GCMs' annually aggregated data
## Most of its operations involve preparation of the data and collecting the
## right data from multiple data files.
EOF.GCMsAM <- function(x,masked=FALSE,lon=NULL,lat=NULL,...) {
  ## This function combines the annual means from different GCMs and performs a
  ## common EOF on the joint data
  #print('EOF.GCMsAM')
  runs <- names(x)
  era5 <- runs[grep('era5',runs)][1]
  runs <- runs[-grep('era5',runs)]
  #print(runs)
  if (length(grep(era5,names(x)))>0) X <- subset(x[[era5]],is=list(lon=lon,lat=lat))
  if (masked) X <- mask(X,land=FALSE)
  X <- regrid(X,is=list(lon=seq(min(lon(X)),max(lon(X)),by=1),
                        lat=seq(min(lat(X)),max(lat(X)),by=1)))
  n <- length(runs); print(n)
  for (i in 1:n) {
    #print(runs[i])
    z <- subset(x[[runs[i]]],is=list(lon=lon,lat=lat))
    attr(z,'source') <- runs[i]
    X <- combine(X,z)
  }
  #print('Calculate EOF for joint dataset')
  eof <- EOF(X,anomaly=FALSE,...)
  invisible(eof)
}

## Function for estimating root-mean-square-error (RMSE) of the common EOFs for
```

```r
## the mean annual cycle
RMSE.AC.CEOF <- function(x,srcid="source") {
  n <- attr(x,'n.apps')
  d <- dim(x)
  rmse <- rep(0,n); modelid <- rep('NA',n)
  for (i in 1:n) {
    y <- attr(x,paste0('appendix.',i))
    for (j in (1:d[2])) rmse[i] <- rmse[i] +
        (RMSE(coredata(x[,j]),coredata(y[,j]))*attr(x,'eigenvalue'))^2
    rmse[i] <- round(sqrt(rmse[i])/sqrt(sum(attr(x,'eigenvalue')^2)),3)
    modelid[i] <- attr(y,srcid)
  }
  names(rmse) <- modelid
  rmse <- rmse[order(rmse,decreasing=FALSE)]
  attr(rmse,'modelid') <- modelid
  return(rmse)
}

## Convert the list with trend maps into a field object to speed up the analysis and to
## enable EOFs for reducing the data volume.
list2field <- function(x,it='annual',lon=NULL,lat=NULL,plot=FALSE,...) {
  #print('list2field')
  cn <- names(x)
  if (!is.null(it)) if (is.character(it)) cn <- names(x)[grep(it,names(x))]
  n <- length(cn)
  #print(cn)
  Y <- subset(x[[cn[1]]],is=list(lon=lon,lat=lat)) # GCMs
  nxy <- length(Y)
  X <- matrix(rep(NA,n*nxy),n,nxy)
  for (i in 1:n) {
    z <- regrid(subset(x[[cn[i]]],is=list(lon=lon,lat=lat)),is=Y)
    X[i,] <- c(z)
    #print(c(i,cn[i],round(mean(X[i,],na.rm=TRUE),2)))
  }

  X <- as.field(zoo(X,order.by=1:n),param=param,unit=esd::unit(Y),lon=lon(Y),lat=lat(Y),
                info='trend maps: list2field (GCMsAC.R)',longname='trend coefficients')

  if (plot) map(X,FUN='sd',
                main='Multi-modal ensemble trend spread (CMIP6)',sub='1959-2021')
  par(new=FALSE)
  #par(fig=c(0,0.3,0,0.3),new=TRUE)
  #hist(c(r),lwd=3,col=rgb(0,0,0,0.5))
  attr(X,'ID') <- cn
  invisible(X)
}

## The following function evaluates given PC of the ensemble by comparing the
## rank of the observations. If the observations and model results belong to the
## same statistical population, then the rank should be random between 0 and 100%.
## We can use a Monte-Carlo approach to estimate confidence interval.
## x is the common EOF-object and ip is the PC number.
evalGCMsAM <- function(x,cmip,ip=1,N.test=300,breaks=seq(0,1,by=0.05),plot=TRUE,
```

```
                         verbose=FALSE) {
  d <- dim(x)
  n <- attr(x,'n.apps')
  cols <- rownames(table(cmip))
  if (length(cols)==1) cols <- rep(cols,2)
  #rank <- apply(coredata(x),2,order)
  X <- matrix(rep(NA,d[1]*(n+1)),n+1,d[1])
  X[1,] <- coredata(x)[,ip]
  for (i in 1:n) {
    if (length(X[i+1,])==length(coredata(attr(x,paste0('appendix.',i)))[,ip]))
      X[i+1,] <- coredata(attr(x,paste0('appendix.',i)))[,ip]
  }
  cmip[!is.finite(apply(X[-1,],1,mean))] <- NA
  X6 <- X[c(T,is.element(cmip,cols[1])),]; d6 <- dim(X6)
  X5 <- X[c(T,is.element(cmip,cols[2])),]; d5 <- dim(X5)
  eval5 <- apply(X5,2,function(x) order(x)[1]/length(x))
  eval6 <- apply(X6,2,function(x) order(x)[1]/length(x))
  ## Monte-Carlo null-distribution:
  null5 <- matrix(rep(NA,d5[1]*N.test),N.test,d5[1])
  null6 <- matrix(rep(NA,d6[1]*N.test),N.test,d6[1])
  for (i in 1:N.test) {
    X5[] <- rnorm(length(X5)); dim(X5) <- d5
    X6[] <- rnorm(length(X6)); dim(X6) <- d6
    null5[i,] <- apply(X5,1,function(x) order(x)[1]/length(x))
    null6[i,] <- apply(X6,1,function(x) order(x)[1]/length(x))
  }
  z5 <- apply(null5,1,function(x) hist(x,breaks=breaks,plot=FALSE)$density)
  z6 <- apply(null6,1,function(x) hist(x,breaks=breaks,plot=FALSE)$density)
  h5 <- hist(eval5,breaks=breaks,plot=FALSE)
  h6 <- hist(eval6,breaks=breaks,plot=FALSE)
  attr(h5,'Monte-Carlo null.distribution') <- z5
  attr(h6,'Monte-Carlo null.distribution') <- z6

  ## Plot the results
  if (plot) {
    plot(eval5,type='b',col='brown',lty=2,pch=19,cex=1.5,ylim=c(0,1),
         main='Rank evaluation statistics',xlab='year',ylab='rank in ensemble')
    lines(eval6,type='b',col='darkgreen',lty=3,pch=19)
    grid()
  }
  return(list(eval.CMIP5=h5,eval.CMIP6=h6,rank.CMIP5=eval5,rank.CMIP6=eval6,
              CMIP5=X5,CMIP6=X6))
}

# Not used...
# hatched <- function(m,s,col='grey50') {
#   polygon(c(1:12,rev(1:12)),c(m+s,m-1),col=col,border=col)
# }

## Replace missing data with mean values for estimating EOFs of wet-day mean
## precipitation (NA when there are no wet days)
na2mean <- function(x) {
  d <- dim(x)
```

```r
  z <- apply(x,2,'mean',na.rm=TRUE)
  for (it in 1:d[1]) {
    ina <- !is.finite(x[it,])
    x[it,ina] <- z[ina]
  }
  return(x)
}
```

3.2 Calculations

The following chunk of code is used for initialising the calculations. It contains some options (parameters) used for carrying out this analysis.

```r
print('---<GCMSACAM.R>---')
```

```
## [1] "---<GCMSACAM.R>---"
```

```r
par0 <- par()
plot <- TRUE
reprocess <- FALSE
outpath <- '~/R/'

paths <- c('~/data/CMIP5.monthly/','~/data/CMIP6.monthly/')
run <- 'r1i1p1f1|r1i1p1'
bad.data <- NULL
```

3.2.1 Post- and pre-processing

The post-processing can be a bit time-consuming and the code therefore searches for results saved previously in the specified path. It will only carry out all the pre-processing if there is no data files with the expected file name.

```r
print('Post-processing:')
```

```
## [1] "Post-processing:"
```

```r
print(regions); print(paths)
```

```
## [1] "-5-45E.55-72N"
```

```
## [1] "~/data/CMIP5.monthly/" "~/data/CMIP6.monthly/"
```

```r
for (ir in 1:length(regions)) {
  for (path in paths) {
    ssps <- list.files(path=path)
    ssps <- ssps[grep("RCP45|ssp245|hist",ssps)]
    for (ssp in ssps) {
      print(ssp)
      for (param in c('tas', 'pr', 'psl', 'fw', 'mu')) {
```

```r
  if (param=='pr') {
    gcmfiles <- list.files(path=paste0(path,ssp),pattern='pr_Amon',full.names = TRUE)
    gcmfiles <- gcmfiles[grep(run,gcmfiles)]
    gcmfiles <- gcmfiles[grep('_1850-|_1860-|_1861-',gcmfiles)]
    reanalysis <- '~/data/ERA5/ERA5_tp_mon.nc'
  } else if (param=='tas') {
    gcmfiles <- list.files(path=paste0(path,ssp),pattern='tas_Amon',full.names = TRUE)
    gcmfiles <- gcmfiles[grep(run,gcmfiles)]
    gcmfiles <- gcmfiles[grep('_1850-|_1860-|_1861-',gcmfiles)]
    reanalysis <- '~/data/ERA5/ERA5_t2m_mon.nc'
  } else if (param=='psl') {
    gcmfiles <- list.files(path=paste0(path,ssp),pattern='psl_Amon',full.names = TRUE)
    gcmfiles <- gcmfiles[grep(run,gcmfiles)]
    gcmfiles <- gcmfiles[grep('_1850-|_1860-|_1861-',gcmfiles)]
    reanalysis <- '~/data/ERA5/ERA5_slp_mon.nc'
  } else if (param=='mu') {
    ssp <- 'hist'
    gcmfiles <- list.files(path=paste0(path,'wet-day-intensity/',ssp),
                           pattern='monmean_int_of_pr_day',full.names = TRUE)
    gcmfiles <- gcmfiles[grep(run,gcmfiles)]
    gcmfiles <- gcmfiles[grep('_1850|_1860|_1861',gcmfiles)]
    reanalysis <- '~/data/ERA5/monthly_wet-day-intensity/monmean_int_of_ERA5_tp_1940-2022.nc'
    it <- c(1959,2014)
  } else if (param=='fw') {
    ssp <- 'hist'
    gcmfiles <- list.files(path=paste0(path,'wet-day-frequency/',ssp),
                           pattern='monmean_freq_of_pr_day',full.names = TRUE)
    gcmfiles <- gcmfiles[grep(run,gcmfiles)]
    gcmfiles <- gcmfiles[grep('_1850|_1860|_1861',gcmfiles)]
    reanalysis <- '~/data/ERA5/monthly_wet-day-frequency/monmean_freq_of_ERA5_tp_1940-2022.nc'
    it <- c(1959,2014)
  }

  output <- paste0('GCMsACAM.',param,'.',tolower(ssp),'.',regions[ir],'.rda')
  print(output)

  if (!file.exists(output) | (reprocess)) {
    print(gcmfiles); i <- 0
    Y <- list(); Z <- list(); W <- list(); AM <- list()

    print(paste('Number of CMIP6 files to read=',length(gcmfiles)))
    for (gcmfile in gcmfiles) {
      ## Estimate the mean annual cycle
      print(gcmfile)
      X <- retrieve(gcmfile,it=c(1959,2021),lon=lon[,ir],lat=lat[,ir])
      if (param=='pr') {
        X <- X * 30 ## mm/month
        attr(X,'unit') <- 'mm/month'
        FUN='sum'
      } else FUN='mean'
      print(c(varid(X),esd::unit(X),round(range(X,na.rm=TRUE))))
      #cim <- paste(attr(X,'model_id'),attr(X,'realization'),i,sep='.')
      cim <- paste(decipher(gcmfile)[1:3],collapse='.')
```

```r
## Mean annual cycle
acm <- aggregate(X,month,na.rm=TRUE)
## Mean annual variability
acs <- aggregate(X,month,FUN='sd',na.rm=TRUE)
## Annual total
act <- annual(X,FUN=FUN)
djf <- annual(subset(X,it='djf'),FUN=FUN,nmin=3)
mam <- annual(subset(X,it='mam'),FUN=FUN,nmin=3)
jja <- annual(subset(X,it='jja'),FUN=FUN,nmin=3)
son <- annual(subset(X,it='son'),FUN=FUN,nmin=3)
AM[[cim]] <- act
AM[[paste0('djf.',cim)]] <- djf
AM[[paste0('mam.',cim)]] <- mam
AM[[paste0('jja.',cim)]] <- jja
AM[[paste0('son.',cim)]] <- son
## Past trends
if (plot) par(par0)
act <- map(annual(X),FUN='trend',plot=plot)
print('Trend statistics'); print(summary(c(act)))
actDJF <- map(annual(subset(X,it='djf'),nmin=3),FUN='trend',plot=plot)
actMAM <- map(annual(subset(X,it='mam'),nmin=3),FUN='trend',plot=plot)
actJJA <- map(annual(subset(X,it='jja'),nmin=3),FUN='trend',plot=plot)
actSON <- map(annual(subset(X,it='son'),nmin=3),FUN='trend',plot=plot)
Y[[cim]] <- acm
Z[[cim]] <- acs
W[[paste(cim,param,'trend.annual',sep='.')]] <- act
W[[paste(cim,param,'trend.DJF',sep='.')]] <- actDJF
W[[paste(cim,param,'trend.MAM',sep='.')]] <- actMAM
W[[paste(cim,param,'trend.JJA',sep='.')]] <- actJJA
W[[paste(cim,param,'trend.SON',sep='.')]] <- actSON
#map(Y[[cim]])
i <- i + 1
}
## Also show the era5 reanalysis
print('Retrieve ERA5 reanalysis also')
X <- retrieve(reanalysis,it=c(1959,2021),lon=lon[,ir],lat=lat[,ir])
if (param=='pr') {
  X <- X * 30 * 1000 ## m/day -> mm/month
  attr(X,'unit') <- 'mm/month'
  FUN='sum'
} else FUN='mean'
acm <- aggregate(X,month,na.rm=TRUE)
acs <- aggregate(X,month,FUN='sd',na.rm=TRUE)
## Annual total
act <- annual(X,FUN=FUN)
djf <- annual(subset(X,it='djf'),FUN=FUN,nmin=3)
mam <- annual(subset(X,it='mam'),FUN=FUN,nmin=3)
jja <- annual(subset(X,it='jja'),FUN=FUN,nmin=3)
son <- annual(subset(X,it='son'),FUN=FUN,nmin=3)
Y[['era5']] <- acm
Z[['era5']] <- acs
AM[['era5']] <- act
AM[['djf.era5']] <- djf
```

```
          AM[['mam.era5']] <- mam
          AM[['jja.era5']] <- jja
          AM[['son.era5']] <- son
          act <- map(annual(X),FUN='trend',plot=plot)
          actDJF <- map(annual(subset(X,it='djf'),nmin=3),FUN='trend',plot=plot)
          actMAM <- map(annual(subset(X,it='mam'),nmin=3),FUN='trend',plot=plot)
          actJJA <- map(annual(subset(X,it='jja'),nmin=3),FUN='trend',plot=plot)
          actSON <- map(annual(subset(X,it='son'),nmin=3),FUN='trend',plot=plot)
          W[['era5.trend.annual']] <- act
          W[['era5.trend.DJF']] <- actDJF
          W[['era5.trend.MAM']] <- actMAM
          W[['era5.trend.JJA']] <- actJJA
          W[['era5.trend.SON']] <- actSON
          GCMs.AC <- Y; GCMs.std <- Z; GCMs.trend <- W; GCMs.AM <- AM
          save(GCMs.AC,GCMs.std,GCMs.trend,GCMs.AM,file=paste0(outpath,output))
        } else load(output)
      }
    }
  }
}
```

3.2.2   Dignostics

The diagnostics involve calculating and plotting the common EOFs of the post-processed data. The plots of the common EOFs show the spatial pattern (upper left), the variance (based on the eigenvalues) associated with each pattern and the principal components (PCs) representing each GCM (red and darkgreen curves) and the ERA5 reanalysis (black curve). The two first figures show the common EOFs for the mean annual cycle whereas the next two show common EOFs of annual mean or seasonal maps in addition to one EOF analysis applied to trend maps for the historical period 1959-2021.

3.2.3   Wet-day frequency

```
##-----------------------------------------------------------
## Diagnostics

seasons <- c('annual','djf','mam','jja','son')
ssp <- 'hist'

for (masked in c(FALSE)) {

  for (region in regions) {
    for (param in c('tas', 'pr', 'psl','fw','mu')[4]) {

      output <- file.path('~/R',paste0('GCMsACAM.',param,'.',ssp,'.',region,'.rda'))
      #output <- paste0('GCMsACAM.',param,'.',ssp,'.',region,'.rda')
      load(output)
      GCMs.AM.all <- GCMs.AM
      names(GCMs.AM) <- sub('of.pr.day_','',names(GCMs.AM))
      names(GCMs.AC) <- sub('of.pr.day_','',names(GCMs.AC))
      names(GCMs.trend) <- sub('of.pr.day_','',names(GCMs.trend))
```

```
## Exclude known bad data
if (length(bad.data) > 0) {
  for (bad in bad.data) {GCMs.AC[[bad]] <- NULL;
  GCMs.AM[[bad]] <- NULL; GCMs.trend[[bad]] <- NULL}
}

if (region==regions[1]) print(names(GCMs.AC))

## Fix problems
n <- length(GCMs.AC)
m <- length(GCMs.AM)
for (i in 1:n)  attr(GCMs.AC[[i]],'source') <- names(GCMs.AC)[i]
for (i in 1:m)  attr(GCMs.AM[[i]],'source') <- names(GCMs.AM)[i]

## Show the CMIP5 and CMIP56 with different colours
col <- names(GCMs.AC)
col[!is.element(col,'era5')] <- 'darkgreen'
#print(table(col))
ngcms <- length(col)

## Use common EOFs of mean annual cycle to compare the GCMs' ability to reproduce the
## mean annual cycle with that seen in the ERA5 reanalysis
GCMs.AC[['era5']] <- aggregate.grid(GCMs.AC[['era5']],
                                    is=list(lon=lon(GCMs.AC[[1]]),lat=lat(GCMs.AC[[1]])))
ceof.ac <- EOF.GCMsAC(GCMs.AC,masked)

## Scalar metric: Use variance-weighted (eigenvalues) sum of RMSE of the PCs wrt ERA5
w.rmse <- RMSE.AC.CEOF(ceof.ac,srcid="source_id")
outlier <- 5
print(paste('Outlier:',attr(w.rmse,'modelid')[outlier]))
col[outlier] <- 'darkblue'

if (plot) {
  par(par0)
  plot(ceof.ac,col=col,alpha=0.2,new=FALSE)
  par(par0)
  plot(ceof.ac,col=col,alpha=0.5,ip=2,new=FALSE)
}

## Table of ranking
## Generate table for LaTeX document:
print(paste('RMSE of PCs of the annual mean cycle for',region,param))
write.table(w.rmse,sep='   &   ', eol = ' \\\\ \n', quote = FALSE)

for (is in seasons[1]) {
  print(paste('Season',is))
  am.names <- names(GCMs.AM.all)
  if (is!='annual') keep <- am.names[grep(is,am.names)] else
    keep <- am.names[-grep('djf|mam|jja|son',am.names)]
  GCMs.AM <- GCMs.AM.all[keep]
  names(GCMs.AM) <- sub('of.pr.day_','',names(GCMs.AM))
  names(GCMs.trend) <- sub('of.pr.day_','',names(GCMs.trend))
```

```
      GCMs.AM[['era5']] <- aggregate.grid(GCMs.AM[['era5']],
                                          is=list(lon=lon(GCMs.AM[[1]]),lat=lat(GCMs.AM[[1]])))
    ceof.am <- EOF.GCMsAM(GCMs.AM,masked)
    if (plot) {
      par(par0)
      plot(ceof.am,col=col,alpha=0.3,new=FALSE)
      par(par0)
      plot(ceof.am,col=col,alpha=0.3,ip=2,new=FALSE)
    }
}

print('Trend analysis')
## Use EOFs of trend maps to assess the trends in the GCMs:
Z <- list2field(GCMs.trend,plot=FALSE)
if (masked) Z <- mask(Z,land=FALSE)
n <- length(attr(Z,'ID'))
col <- attr(Z,'ID')
print(col)
col[] <- 'brown'
col[grep('era5',attr(Z,'ID'))] <- 'black'
teof <- EOF(Z,anomaly=FALSE)
n <- dim(teof)[1]
if (plot) {
  par(par0)
  plot(teof,new=FALSE)
  ## Add a point indicating the ERA5 trends
  par(fig=c(0.05,1,0.025,0.475),mar=c(3,3,2,2),new=TRUE)
  lines((1:n)[is.element(col,'brown')],teof[is.element(col,'brown'),1],
        col='brown',lty=2,lwd=1)
  lines((1:n)[is.element(col,'darkgreen')],teof[is.element(col,'darkgreen'),1],
        col='darkgreen',lwd=1)
  points((1:n)[is.element(col,'black')],teof[(1:n)[is.element(col,'black')],1],
          col='black',cex=1.5,pch=19)
  par(par0)
  plot(teof,ip=2,new=FALSE)
  ## Add a point indicating the ERA5 trends
  par(fig=c(0.05,1,0.025,0.475),mar=c(3,3,2,2),new=TRUE)
  lines((1:n)[is.element(col,'brown')],teof[is.element(col,'brown'),2],
        col='brown',lty=2,lwd=1)
  lines((1:n)[is.element(col,'darkgreen')],teof[is.element(col,'darkgreen'),2],
        col='darkgreen',lwd=1)
  points((1:n)[is.element(col,'black')],teof[(1:n)[is.element(col,'black')],2],
          col='black',cex=1.5,pch=19)
}

## Another way to evaluate is with a scatterplot of the PCs of the two leading modes:
tnms <- names(GCMs.trend); tnms <- tnms[grep('annual',tnms)]
print(tnms)
icmip5 <- grep('rcp',tnms)
icmip6 <- grep('ssp',tnms)
if (plot) {
  par(par0)
  plot(teof[,1],teof[,2],xlab='Mode 1',ylab='Mode2');
```

```
        points(teof[n,1],teof[n,2],pch=19,col='black',cex=1.5);
        points(teof[icmip5,1],teof[icmip5,2],col='red',lwd=2)
        points(teof[icmip6,1],teof[icmip6,2],col='blue',lwd=2)
        text(teof[-n,1],teof[-n,2],1:(n-1),cex=0.6,col='grey40',pos=1)
      }
    }
  }
}
```

```
##  [1] "ACCE" "AWI-" "BCC-" "CanE" "CESM" "CMCC" "CNRM" "EC-E" "FGOA" "GFDL"
## [11] "HadG" "IITM" "INM-" "IPSL" "KACE" "KIOS" "MIRO" "MPI-" "MRI-" "NESM"
## [21] "NorC" "NorE" "SAMO" "TaiE" "UKES" "era5"
## [1] "EOF.GCMsAC"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Outlier: CESM2-WACCM-FV2"
```

```
## [1] "RMSE of PCs of the annual mean cycle for -5-45E.55-72N fw"
## x \\
## CNRM-ESM2-1   &  0.186 \\
## CMCC-ESM2   &  0.197 \\
## GFDL-ESM4   &  0.2 \\
## MIROC6   &  0.218 \\
## CESM2-WACCM-FV2   &  0.219 \\
## TaiESM1   &  0.222 \\
## HadGEM3-GC31-MM   &  0.224 \\
## MRI-ESM2-0   &  0.232 \\
## IPSL-CM6A-LR-INCA   &  0.238 \\
## NorESM2-MM   &  0.238 \\
## UKESM1-0-LL   &  0.252 \\
## SAM0-UNICON   &  0.255 \\
```

```
## CanESM5   &  0.258 \\
## INM-CM5-0   &  0.268 \\
## NorCPM1   &  0.269 \\
## KACE-1-0-G   &  0.276 \\
## AWI-ESM-1-1-LR   &  0.277 \\
## KIOST-ESM   &  0.281 \\
## ACCESS-ESM1-5   &  0.285 \\
## EC-Earth3-Veg-LR   &  0.288 \\
## FGOALS-g3   &  0.292 \\
## MPI-ESM1-2-LR   &  0.292 \\
## BCC-ESM1   &  0.295 \\
## NESM3   &  0.309 \\
## IITM-ESM   &  0.32 \\
## [1] "Season annual"
## [1] 25
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
## [1] "Warning - different units: m 1"
```

```
## [1] "Trend analysis"
##  [1] "ACCE.fw.trend.annual" "AWI-.fw.trend.annual" "BCC-.fw.trend.annual"
##  [4] "CanE.fw.trend.annual" "CESM.fw.trend.annual" "CMCC.fw.trend.annual"
##  [7] "CNRM.fw.trend.annual" "EC-E.fw.trend.annual" "FGOA.fw.trend.annual"
## [10] "GFDL.fw.trend.annual" "HadG.fw.trend.annual" "IITM.fw.trend.annual"
## [13] "INM-.fw.trend.annual" "IPSL.fw.trend.annual" "KACE.fw.trend.annual"
## [16] "KIOS.fw.trend.annual" "MIRO.fw.trend.annual" "MPI-.fw.trend.annual"
## [19] "MRI-.fw.trend.annual" "NESM.fw.trend.annual" "NorC.fw.trend.annual"
## [22] "NorE.fw.trend.annual" "SAMO.fw.trend.annual" "TaiE.fw.trend.annual"
## [25] "UKES.fw.trend.annual" "era5.trend.annual"
```

fw (1) — (1 − 26)

20 leading EOFs: 98.6 % of variance

Leading PC#1 of trend coefficients - Explained variance = 45.05%

```
##  [1] "ACCE.fw.trend.annual" "AWI-.fw.trend.annual" "BCC-.fw.trend.annual"
##  [4] "CanE.fw.trend.annual" "CESM.fw.trend.annual" "CMCC.fw.trend.annual"
##  [7] "CNRM.fw.trend.annual" "EC-E.fw.trend.annual" "FGOA.fw.trend.annual"
## [10] "GFDL.fw.trend.annual" "HadG.fw.trend.annual" "IITM.fw.trend.annual"
## [13] "INM-.fw.trend.annual" "IPSL.fw.trend.annual" "KACE.fw.trend.annual"
## [16] "KIOS.fw.trend.annual" "MIRO.fw.trend.annual" "MPI-.fw.trend.annual"
## [19] "MRI-.fw.trend.annual" "NESM.fw.trend.annual" "NorC.fw.trend.annual"
## [22] "NorE.fw.trend.annual" "SAMO.fw.trend.annual" "TaiE.fw.trend.annual"
## [25] "UKES.fw.trend.annual" "era5.trend.annual"
```

When it comes to the wet-day frequency f_w the evaluations gave different results depending on the exact region. PC1 of the common EOFs for the mean annual cycle was surprisingly sensitive to modest variations in the selected geographical domain, especially for regions that included a large fraction of ocean it seems. A domain covering most of Europa gave a similar mean annual structure in PC1 for most CMIP6 models and ERA5, except for one GCM. The analysis indicates that one GCM is an outlier (CESM2-WACCM-FV2) and produced a mean annual shape distinct from the rest of the CMIP6 ensemble.

A smaller regional domain seemed to give a poorer match in the mean annual cycle covariance structure of f_w in the CMIP6 ensemble and ERA5. A mismatch in the mean annual cycle in f_w may be interpreted as a limitation of the models' ability to simulate changes. But the mean annual cycle was not resolved in the downscaling of annual f_w in the section above, although a limited representation of the mean annual cycle should underscore that the downscaled results have some caveats.

The common EOF evaluation suggested that the CMIP6 representation of interannual variations in annual f_w in general matched that of ERA5.

The common EOF of maps with trend coefficients for f_w (one map for each model run) suggested that the CMIP6 matched the long-term trends in f_w found in ERA5, albeit with a substantial scatter within the model ensemble.

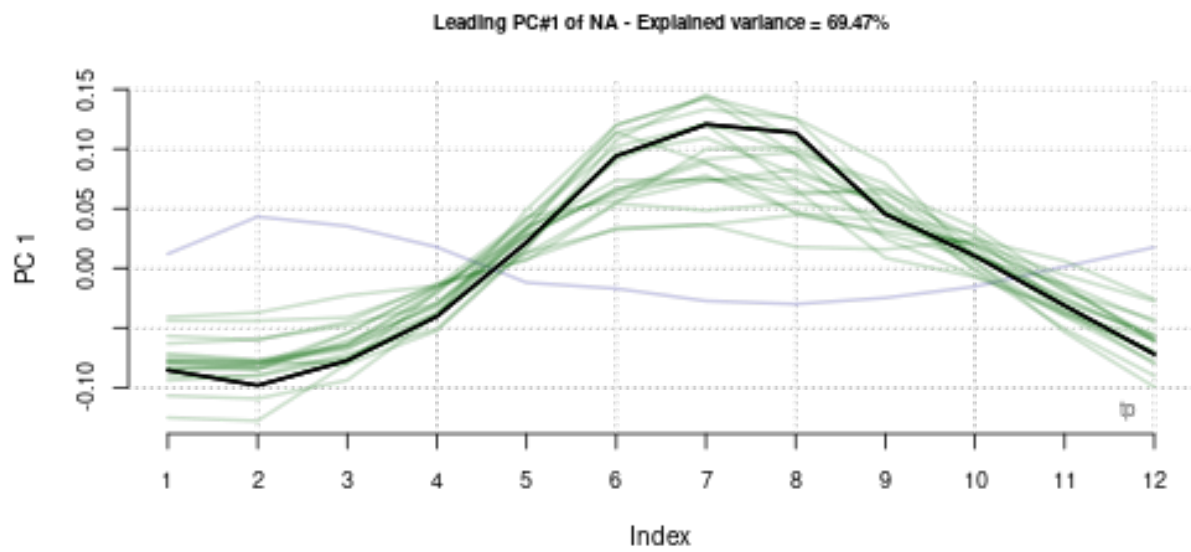
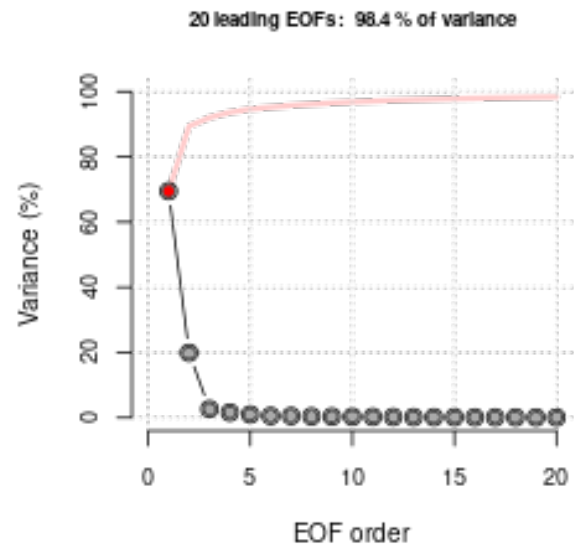
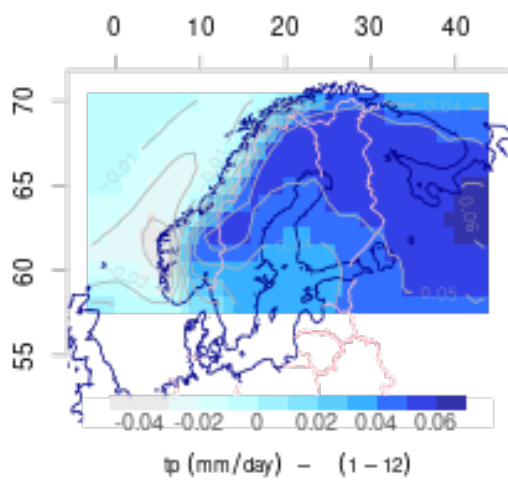
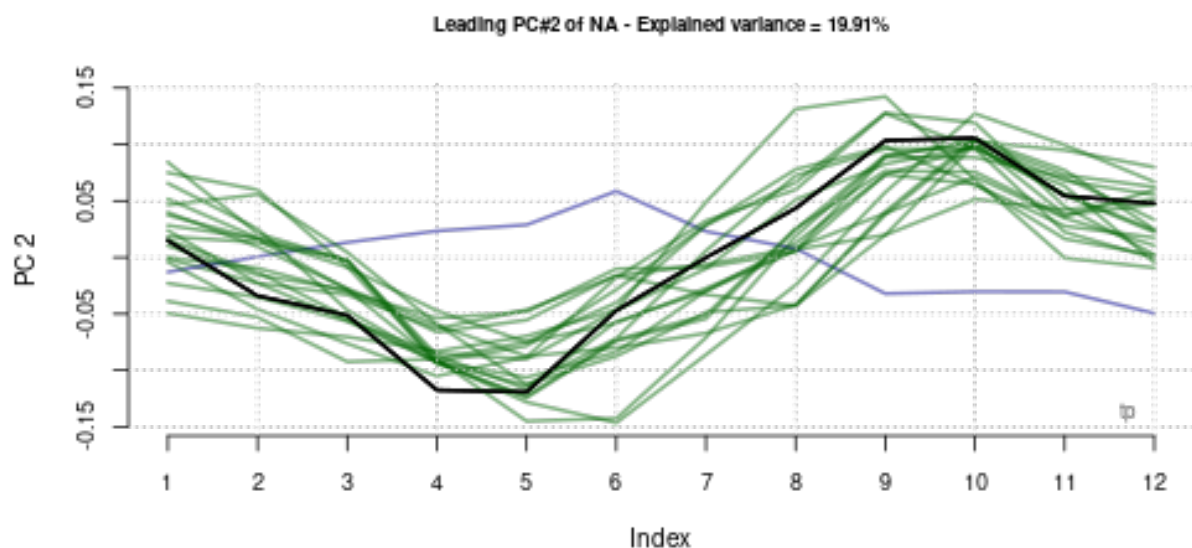
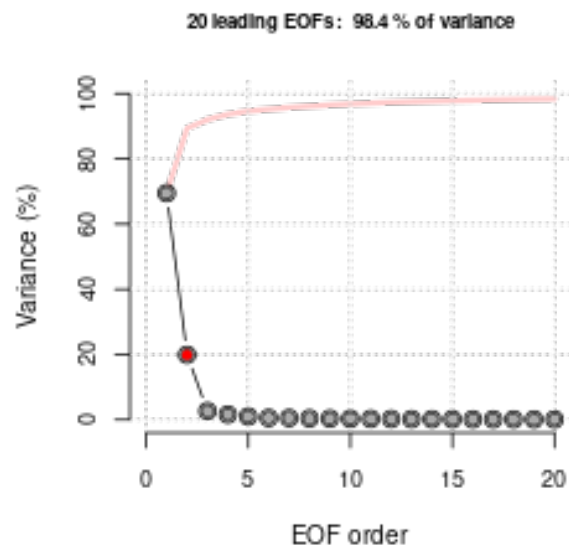
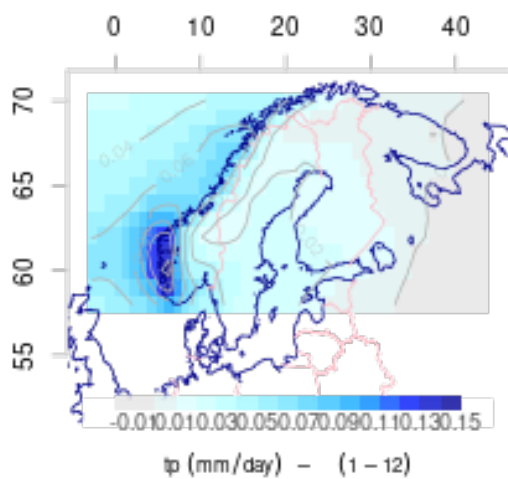
3.3 Wet-day mean μ

```
##---------------------------------------------------------------
## Diagnostics

for (masked in c(FALSE)) {

  for (region in regions) {
    for (param in c('tas', 'pr', 'psl','fw','mu')[5]) {
      output <- file.path('~/R',paste0('GCMsACAM.',param,'.',ssp,'.',region,'.rda'))
      #output <- paste0('GCMsACAM.',param,'.',ssp,'.',region,'.rda')
      load(output)
      GCMs.AM.all <- GCMs.AM
      names(GCMs.AM) <- sub('of.pr.day_','',names(GCMs.AM))
      names(GCMs.AC) <- sub('of.pr.day_','',names(GCMs.AC))

      ## Exclude known bad data
      if (length(bad.data) > 0) {
        for (bad in bad.data) {GCMs.AC[[bad]] <- NULL; GCMs.AM[[bad]] <- NULL;
        GCMs.trend[[bad]] <- NULL}
      }

      if (region==regions[1]) print(names(GCMs.AC))

      ## Fix problems
      n <- length(GCMs.AC)
      m <- length(GCMs.AM)
      for (i in 1:n)  attr(GCMs.AC[[i]],'source') <- names(GCMs.AC)[i]
      for (i in 1:m)  attr(GCMs.AM[[i]],'source') <- names(GCMs.AM)[i]

      ## CIESM_ssp245_r1i1p1f1 seems to have units of m rather than mm
      if (param=='mu') {
        for (i in 1:n) {
          if (max(GCMs.AC[[i]],na.rm=TRUE) < 1) GCMs.AC[[i]] <- 1000*GCMs.AC[[i]]
          attr(GCMs.AC[[i]],'units') <- 'mm/day'
        }
      }

      ## Show the CMIP5 and CMIP56 with different colours
      col <- names(GCMs.AC)
      col[!is.element(col,'era5')] <- 'darkgreen'
      print(table(col))
      ngcms <- length(col) -1

      ## Use common EOFs of mean annual cycle to compare the GCMs' ability to reproduce the
      ## mean annual cycle with that seen in the ERA5 reanalysis
      GCMs.AC[['era5']] <- aggregate.grid(GCMs.AC[['era5']],
                                          is=list(lon=lon(GCMs.AC[[1]]),lat=lat(GCMs.AC[[1]])))
```

```r
## Some model results cause the code to crash for some reason
## - remove them from the analysis

for (bad in bad.runs) GCMs.AC[[bad]] <- NULL
print(names(GCMs.AC)); ngcms <- length(GCMs.AC)-1

ceof.ac <- EOF.GCMsAC(GCMs.AC,masked)

## Estimate ensemble mean and standard deviation for plotting
pc1.ac <- matrix(rep(NA,12*ngcms),12,ngcms); pc2.ac <- pc1.ac
for (i in 1:ngcms) {
  pc1.ac[,i] <- attr(ceof.ac,paste0('appendix.',i))[,1]
  pc2.ac[,i] <- attr(ceof.ac,paste0('appendix.',i))[,2]
}

## Scalar metric: Use variance-weighted (eigenvalues) sum of RMSE of the PCs wrt ERA5
w.rmse <- RMSE.AC.CEOF(ceof.ac,srcid="source_id")
outlier <- 5
print(paste('Outlier:',attr(w.rmse,'modelid')[outlier]))
col[outlier] <- 'darkblue'

if (plot) {
  par(par0)
  plot(ceof.ac,col=col,alpha=0.2,new=FALSE)
  par(par0)
  plot(ceof.ac,col=col,alpha=0.5,ip=2,new=FALSE)
}

## Table of ranking
## Generate table for LaTeX document:
write.table(w.rmse,sep='   &  ', eol = ' \\\\ \n', quote = FALSE)

## Use common EOFs of annually aggregated data to compare the interannual
## variability simulated by GCMs with the ERA5 reanalysis

for (is in seasons[1]) {
  print(paste('Season',is))
  am.names <- names(GCMs.AM.all)
  if (is!='annual') keep <- am.names[grep(is,am.names)] else
    keep <- am.names[-grep('djf|mam|jja|son',am.names)]
  GCMs.AM <- GCMs.AM.all[keep]
  names(GCMs.AM) <- sub('of.pr.day_','',names(GCMs.AM))

  if (param=='mu') {
    for (i in 1:n) {
      if (max(GCMs.AM[[i]],na.rm=TRUE) < 1) GCMs.AM[[i]] <- 1000*GCMs.AM[[i]]
      attr(GCMs.AM[[i]],'units') <- 'mm/day'
    }
  }

  GCMs.AM[['era5']] <- aggregate.grid(GCMs.AM[['era5']],
                                      is=list(lon=lon(GCMs.AM[[1]]),lat=lat(GCMs.AM[[1]])))
  GCMsAM <- lapply(GCMs.AM,'na2mean')
```

```
    ceof.am <- EOF.GCMsAM(GCMsAM,masked)
    if (plot) {
      par(par0)
      plot(ceof.am,col=col,alpha=0.3,new=FALSE)
      par(par0)
      plot(ceof.am,col=col,alpha=0.3,ip=2,new=FALSE)
    }
}

## Fix a bug: ERA5 units was in m rather than mm
GCMs.trend$era5.trend.annual <- 1000*GCMs.trend$era5.trend.annual
## Use EOFs of trend maps to assess the trends in the GCMs:
Z <- list2field(GCMs.trend,plot=FALSE)
if (masked) Z <- mask(Z,land=FALSE)
n <- length(attr(Z,'ID'))
col <- attr(Z,'ID')
print(col)
col[] <- 'brown'
col[grep('era5',attr(Z,'ID'))] <- 'black'
teof <- EOF(Z,anomaly=FALSE)
n <- dim(teof)[1]
if (plot) {
  par(par0)
  plot(teof,type='fill',colbar=list(show=FALSE,pal='t2m'),new=FALSE)
  ## Add a point indicating the ERA5 trends
  par(fig=c(0.05,1,0.025,0.475),mar=c(3,3,2,2),new=TRUE)
  lines((1:n)[is.element(col,'brown')],teof[is.element(col,'brown'),1],
        col='brown',lty=2,lwd=1)
  lines((1:n)[is.element(col,'darkgreen')],teof[is.element(col,'darkgreen'),1],
        col='darkgreen',lwd=1)
  points((1:n)[is.element(col,'black')],teof[(1:n)[is.element(col,'black')],1],
         col='black',cex=1.5,pch=19)
  par(par0)
  plot(teof,ip=2,type='fill',colbar=list(show=FALSE,pal='t2m'),new=FALSE)
  ## Add a point indicating the ERA5 trends
  par(fig=c(0.05,1,0.025,0.475),mar=c(3,3,2,2),new=TRUE)
  lines((1:n)[is.element(col,'brown')],teof[is.element(col,'brown'),2],
        col='brown',lty=2,lwd=1)
  lines((1:n)[is.element(col,'darkgreen')],teof[is.element(col,'darkgreen'),2],
        col='darkgreen',lwd=1)
  points((1:n)[is.element(col,'black')],teof[(1:n)[is.element(col,'black')],2],
         col='black',cex=1.5,pch=19)
}

## Another way to evaluate is with a scatterplot of the PCs of the two leading modes:
tnms <- names(GCMs.trend); tnms <- tnms[grep('annual',tnms)]
print(tnms)
icmip5 <- grep('rcp',tnms)
icmip6 <- grep('ssp',tnms)
if (plot) {
  par(par0)
  plot(teof[,1],teof[,2],xlab='Mode 1',ylab='Mode2');
  points(teof[n,1],teof[n,2],pch=19,col='black',cex=1.5);
```

```
      points(teof[icmip5,1],teof[icmip5,2],col='red',lwd=2)
      points(teof[icmip6,1],teof[icmip6,2],col='blue',lwd=2)
      text(teof[-n,1],teof[-n,2],1:(n-1),cex=0.6,col='grey40',pos=1)
    }
  }
 }
}
```

```
##  [1] "ACCE" "AWI-" "BCC-" "CanE" "CESM" "CMCC" "CNRM" "EC-E" "FGOA" "GFDL"
## [11] "HadG" "IITM" "INM-" "IPSL" "KACE" "KIOS" "MIRO" "MPI-" "MRI-" "NESM"
## [21] "NorC" "NorE" "SAMO" "TaiE" "UKES" "era5"
## col
## darkgreen      era5
##        25         1
##  [1] "ACCE" "AWI-" "BCC-" "CanE" "CESM" "CMCC" "CNRM" "GFDL" "HadG" "IITM"
## [11] "INM-" "KIOS" "MIRO" "MRI-" "NESM" "NorC" "NorE" "SAMO" "TaiE" "era5"
## [1] "EOF.GCMsAC"
## [1] "Outlier: CESM2-WACCM-FV2"
```

```
## x \\
## MRI-ESM2-0   &  0.227 \\
## KIOST-ESM   &  0.237 \\
## NorESM2-MM   &  0.239 \\
## CESM2-WACCM-FV2   &  0.246 \\
## BCC-ESM1   &  0.267 \\
## NorCPM1   &  0.267 \\
## GFDL-ESM4   &  0.27 \\
## CMCC-ESM2   &  0.284 \\
## CanESM5   &  0.285 \\
## TaiESM1   &  0.287 \\
## NESM3   &  0.295 \\
## CNRM-ESM2-1   &  0.296 \\
## MIROC6   &  0.3 \\
```

```
## SAMO-UNICON    &  0.305 \\
## ACCESS-ESM1-5    &  0.313 \\
## AWI-ESM-1-1-LR    &  0.313 \\
## IITM-ESM    &  0.342 \\
## HadGEM3-GC31-MM    &  0.345 \\
## INM-CM5-0    &  0.354 \\
## [1] "Season annual"
## [1] 25
```

```
##  [1] "of.pr.day_ACCE.mu.trend.annual" "of.pr.day_AWI-.mu.trend.annual"
##  [3] "of.pr.day_BCC-.mu.trend.annual" "of.pr.day_CanE.mu.trend.annual"
##  [5] "of.pr.day_CESM.mu.trend.annual" "of.pr.day_CMCC.mu.trend.annual"
##  [7] "of.pr.day_CNRM.mu.trend.annual" "of.pr.day_EC-E.mu.trend.annual"
##  [9] "of.pr.day_FGOA.mu.trend.annual" "of.pr.day_GFDL.mu.trend.annual"
## [11] "of.pr.day_HadG.mu.trend.annual" "of.pr.day_IITM.mu.trend.annual"
## [13] "of.pr.day_INM-.mu.trend.annual" "of.pr.day_IPSL.mu.trend.annual"
## [15] "of.pr.day_KACE.mu.trend.annual" "of.pr.day_KIOS.mu.trend.annual"
## [17] "of.pr.day_MIRO.mu.trend.annual" "of.pr.day_MPI-.mu.trend.annual"
## [19] "of.pr.day_MRI-.mu.trend.annual" "of.pr.day_NESM.mu.trend.annual"
## [21] "of.pr.day_NorC.mu.trend.annual" "of.pr.day_NorE.mu.trend.annual"
## [23] "of.pr.day_SAMO.mu.trend.annual" "of.pr.day_TaiE.mu.trend.annual"
## [25] "of.pr.day_UKES.mu.trend.annual" "era5.trend.annual"
```

20 leading EOFs: 98.7 % of variance

Leading PC#1 of trend coefficients - Explained variance = 69.75%

```
##  [1] "of.pr.day_ACCE.mu.trend.annual" "of.pr.day_AWI-.mu.trend.annual"
##  [3] "of.pr.day_BCC-.mu.trend.annual" "of.pr.day_CanE.mu.trend.annual"
##  [5] "of.pr.day_CESM.mu.trend.annual" "of.pr.day_CMCC.mu.trend.annual"
##  [7] "of.pr.day_CNRM.mu.trend.annual" "of.pr.day_EC-E.mu.trend.annual"
##  [9] "of.pr.day_FGOA.mu.trend.annual" "of.pr.day_GFDL.mu.trend.annual"
## [11] "of.pr.day_HadG.mu.trend.annual" "of.pr.day_IITM.mu.trend.annual"
## [13] "of.pr.day_INM-.mu.trend.annual" "of.pr.day_IPSL.mu.trend.annual"
## [15] "of.pr.day_KACE.mu.trend.annual" "of.pr.day_KIOS.mu.trend.annual"
## [17] "of.pr.day_MIRO.mu.trend.annual" "of.pr.day_MPI-.mu.trend.annual"
## [19] "of.pr.day_MRI-.mu.trend.annual" "of.pr.day_NESM.mu.trend.annual"
## [21] "of.pr.day_NorC.mu.trend.annual" "of.pr.day_NorE.mu.trend.annual"
## [23] "of.pr.day_SAMO.mu.trend.annual" "of.pr.day_TaiE.mu.trend.annual"
## [25] "of.pr.day_UKES.mu.trend.annual" "era5.trend.annual"
```

Some of the grids in ERA5 and GCMs did not produce a single rainy day (threshold 1 mm/day) in some months and caused missing values (NA). This was a problem for dry regions, but when it happened for a rare occasion, the NA could be replaced with the mean value (`na2mean()`) to allow the EOF calculation to proceed.

The PCs for the seasonal cycle in μ had one outlier (CESM2-WACCM-FV2) compared to the rest of the ensemble for the chosen domain with large fraction of maritime environment. The analysis of the mean seasonal cycle and match between ERA5 and CMIP6 were sensitive to the geographical region selected for the common EOFs, but the interannual variability represented by the CMIP6 ensemble was generally consistent with that of ERA5.

A smaller geographical region suggested more similar mean annual cycle than a larger spatial domain, as well as a better match of the trend coefficients for μ. The analysis also indicated quite good match in the mean annual cycle over the Barents Sea and Arctic Norwegian Sea.

When the mean annual cycle in μ from the CMIP6 ensemble doesn't correspond to that in ERA5 it may

suggest that the models have a limited ability to simulate changes. The mean annual cycle was not resolved in the downscaling of annual μ in the section above, but a limited representation of the mean annual cycle should underscore that the downscaled results probably have some caveats.

The evaluation of trend coefficient maps suggested that ERA5 was mostly within the CMIP6 ensemble spread, and sometimes near the limits of the CMIP6 ensemble range.

These results suggest that it's important to use a suitable domain choice when using μ as predictor for empirical-statistical downscaling (ESD).

3.3.1   Figures for the manuscript

The following chunk shows how the figures in the manuscript were generated: Fig. 1. Station map Fig. 2. f_w and μ for Oslo. (1x2) Fig. 3. Maps of mean and tends in f_w and μ (2x2) Fig. 4. Precipitation trends and contribution from wet-days and intensity (3x) Fig. 5. Number of days with 30 mm (1). Fig. 6. Maps of Pr(X>30mm): mean, trend and relative trend (3x1). Fig. 7. Maps of mean and trend in 10-year and 25-year return values (2x2). Fig. 8. IDF curves.

```r
## Fig.1
map(annual(Y,FUN='sum'),FUN='mean',cex=1,colbar=list(breaks=seq(0,4000,by=250),
                                                pal='precip.ipcc',show=TRUE))
## Fig.2
layout(matrix(c(1,2),1,2))
plot(allN(fw.oslo.ssp370),ylim=c(0,1),
     main='Oslo: wet-day frequency: SSP370',set2par0=FALSE)
plot(allN(mu.oslo.ssp370),
     main='Oslo: mean precipitation intensity: SSP370',set2par0=FALSE)

## Fig.3
layout(matrix(c(1,2,3,4),2,2))
par(mar=c(2,0.5,1.5,0.5))
map(m.fw.ssp370,FUN='mean',type='fill',showaxis=FALSE,main=expression(paste('Mean ',f[w],' (SSP370)')))
map(d.fw.ssp370,colbar=list(breaks=seq(-0.3,0.3,by=0.05),pal='precip.ipcc'),type='fill',
    ,main=expression(paste('Trend ',f[w],' (SSP370)')),showaxis=FALSE)
map(m.mu.ssp370,FUN='mean',type='fill',showaxis=FALSE,main=expression(paste('Mean ',mu,' (SSP370)')))
map(d.mu.ssp370,colbar=list(breaks=seq(-0.3,0.3,by=0.05),pal='precip.ipcc'),type='fill',
    main=expression(paste('Trend ',mu,' (mm/decade; SSP370)')),showaxis=FALSE)

## Fig.4
layout(matrix(c(0,2,1,2,1,3,0,3),2,4))
par(mar=c(2,0.5,1.5,0.5))
map(dp1+dp2,colbar=list(breaks=seq(-0.1,0.1,by=0.01),pal='precip.ipcc'),
    main='Trend in mean precipitation',showaxis=FALSE)
map(dp1,colbar=list(breaks=seq(-0.1,0.1,by=0.01),pal='precip.ipcc'),
    main='Trend due to changes in number of rainy days',showaxis=FALSE)
map(dp2,colbar=list(breaks=seq(-0.1,0.1,by=0.01),pal='precip.ipcc'),
    main='Trend due to changes in rain intensity',showaxis=FALSE)

# Fig.5
plot(m.x30,main='Oslo: Fraction of days with more than 30 mm precipitation',
     ylab=expression(Pr(X > 30*mm)),col='red',xlim=c(1940,2025),
     ylim=c(0,max(rainequation(y,threshold=30),m.x30+s.x30,na.rm=TRUE)))
lines(m.x30+s.x30,lty=2,col='red')
lines(m.x30-s.x30,lty=2,col='red')
```

```r
points(rainequation(y,threshold=30),pch=19)
grid()

# Fig.6
layout(matrix(c(1,2),1,2))
par(mar=c(2,0.5,1.5,0.5))
map(Pr.gt.x,main='Probability of more than 30 mm/day',type='fill',showaxis=FALSE)
map(ptPr,colbar=list(breaks=seq(-900,900,by=100),pal='precip.ipcc'),
    main='Relative trend in Pr(X>30mm)',type='fill',showaxis=FALSE)

## Fig.7
layout(matrix(c(1,2,3,4),2,2))
par(mar=c(2,0.5,1.5,0.5))
for (tau  in c(10, 25)) {
  ## Estimate the value for alpha for given return-interval
  alpha <- c.alpha[1] + c.alpha[2] * log(tau)
  ## Estimate the mean return-value based on the above expression:
  x.L <- alpha * m.mu.ssp370 * log(m.fw.ssp370 * tau * 365.25)
  attr(x.L,'variable') <- 'precip'
  attr(x.L,'unit') <- 'mm/day'
  map(x.L,main=paste0(tau,'-year return value (mean for SSP370)'),
      type='fill',showaxis=FALSE)
  ## Estimate change
  d.x.L <- alpha * d.mu.ssp370 * log(m.fw.ssp370 * tau * 365.25) +
    alpha * m.mu.ssp370 * d.fw.ssp370/m.fw.ssp370
  attr(d.x.L,'variable') <- 'precip'
  attr(d.x.L,'unit') <- '"%/decade"'
  ## The Knitting doesn't get these figures quite right
  map(d.x.L,colbar=list(breaks=seq(-2,2,by=0.2),pal='precip.ipcc'),
      main=paste0('Trend in the ',tau,'-year return value'),
      type='fill',showaxis=FALSE)
}

## Fig.8
## Plot the IDF for the future: thick curves
plot.IDF(idf2)
## Compare with IDF representing the present: thin curves
lines.IDF(idf1,col='grey70')
lines.IDF(idf1,lty=2)
lines.IDF(idf1,lty=3,col='black')
```

References:

Benestad, R.E., 2021. A Norwegian Approach to Downscaling (preprint). Climate; Earth system modeling. https://doi.org/10.5194/gmd-2021-176

Benestad, R.E., 2001. A comparison between two empirical downscaling strategies. Int. J. Climatology 21, 1645–1668.

Benestad, R.E., Lutz, J., Dyrrdal, A.V., Haugen, J.E., Parding, K.M., Dobler, A., 2020. Testing a simple formula for calculating approximate intensity-duration-frequency curves. Environmental Research Letters. https://doi.org/10.1088/1748-9326/abd4ab

Benestad, R.E., Mezghani, A., 2015. On downscaling probabilities for heavy 24-hour precipitation events at seasonal-to-decadal scales. Tellus A 67. https://doi.org/10.3402/tellusa.v67.25954

Benestad, R.E., Mezghani, A., Lutz, J., Dobler, A., Parding, K.M., Landgren, O.A., 2023. Various ways of

using empirical orthogonal functions for climate model evaluation. Geoscientific Model Development 16, 2899–2913. https://doi.org/10.5194/gmd-16-2899-2023

Benestad, R.E., Parding, K.M., Erlandsen, H.B., Mezghani, A., 2019. A simple equation to study changes in rainfall statistics. Environmental Research Letters 14, 084017. https://doi.org/10.1088/1748-9326/ab2bb2

Eyring, V., Bony, S., Meehl, G.A., Senior, C., Stevens, B., Stouffer, R.J., Taylor, K.E., 2015. Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organisation. Geoscientific Model Development Discussions 8, 10539–10583. https://doi.org/10.5194/gmdd-8-10539-2015

Hersbach, H., Dee, D., 2016. ERA5 reanalysis is in production. ({ECMWF} {Newsletter} No. 147). ECMWF, Reading, United Kingdom.

Klein Tank, A.M.G. and J.B.W., Konnen, G.P., Böhm, R., Demarée, G., Gocheva, A., Mileta, M., Pashiardis, S., Hejkrlik, L., Kern-Hansen, C., Heino, R., Bessemoulin, P., Müller-Westermeier, G., Tzanakou, M., Szalai, S., Pálsdóttir, T., Fitzgerald, D., Rubin, S., Capaldo, M., Maugeri, M., Leitass, A., Bukantis, A., Aberfeld, R., Engelen, A.F.V. van, Førland, E., Mietus, M., Coelho, F., Mares, C., Razuvaev, V., Nieplova, E., Cegnar, T., López, J.A., Dahlström, B., Moberg, A., Kirchhofer, W., Ceylan, A., Pachaliuk, O., Alexander, L.V., Petrovic, P., 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. International Journal of Climatology 22, 1441–1453.

Parding, K.M., Benestad, R.E., Dyrrdal, A.V., Lutz, J., 2023. A principal-component-based strategy for regionalisation of precipitation intensity–duration–frequency (IDF) statistics. Hydrology and Earth System Sciences 27, 3719–3732. https://doi.org/10.5194/hess-27-3719-2023