

# SUPPORTING MATERIAL: Global hydro-meteorological indicators and changes in the global hydrological cycle and rainfall patterns

Rasmus E. Benestad<sup>1\*</sup>, Cristian Lussana<sup>1</sup>, Julia Lutz<sup>1</sup>,  
Andreas Dobler<sup>1</sup>, Oskar Landgren<sup>1</sup>, Jan Erik Haugen<sup>1</sup>,  
Abdelkader Mezghani<sup>1</sup>, Barbara Casati<sup>2</sup>, Kajsa M. Parding<sup>1</sup>

<sup>1</sup>The Norwegian Meteorological Institute, Oslo, Norway,  
Blindern PO Box 43, 0313 Oslo, Norway

<sup>2</sup>Meteorological Research Division, Environment and Climate Change Canada

\*To whom correspondence should be addressed; E-mail: rasmus.benestad@met.no.

March 30, 2022

**This document provides supporting material for Benestad et al. 'Global hydro-meteorological indicators and changes in the global hydrological cycle and rainfall patterns', and the data and R-code are available from Figshare.com (I) (DOI: 10.6084/m9.figshare.17158133.v7).**

## Introduction

This supporting material provides supporting figures, tables, as well as the output of an R-markdown script used in the analysis presented by Benestad et al.

## Data processing

We downloaded ERA5 netCDF files with hourly time steps from the Copernicus Climate Data Store (CDS) and then used the Climate Data Operators (CDO) software from Max-Planck Institute for Meteorology to extract the global hydrological climate indicators:

- Daily rainfall: `cdo -O -b f32 daysum <input> <output>`
- Global daily precipitation area: `cdo -fldmean -gec,0.001 <input> <output>`
- Global mean daily precipitation: `cdo -fldmean <input> <output>`
- Global wet-region mean precipitation: `cdo fldmean -setrtomiss,-Inf,0.001 <input> <output>`
- 50 °S–50 °N daily precipitation area:  
`cdo -fldmean -gec,0.001 -sellonlatbox,-180,180,-50,50 <input> <output>`

The maps were generated from annually aggregated data:

- `cdo gtc,0.001 <input> mask.nc`
- `cdo yearmean mask.nc fw_annual.nc`
- `cdo ifthen mask.nc <input> wetdayamount.nc`
- `cdo yearmean wetdayamount.nc mu_annual.nc`

The CDO argument 'gec' invokes the greater equal constant, according to the CDO manual (2):

$$o(t, x) = \begin{cases} 1 & \text{if } i(t, x) \geq c \quad \wedge i(t, x), c \neq miss \\ 0 & \text{if } i(t, x) < c \quad \wedge i(t, x), c \neq miss \\ miss & \text{if } i(t, x) = miss \quad \vee c = miss \end{cases} \quad (1)$$

## Potential effects from data assimilation

Precipitation is not assimilated directly in neither ERA5 nor the other reanalyses. This means that observed near-surface precipitation data are not used to constraint the state vector of the numerical model during the data assimilation cycle. However, there is one local exception over North-America where “*ERA5 assimilates the NCEP stage IV quantitative precipitation estimates produced over the USA by combining precipitation estimates from the NEXRAD with gauge measurements. An overview of the method used to assimilate this product is provided by Lopez (2011).*”. Furthermore, the ERA5 paper (3) describes the observations used in great detail (Sec. 5). Hence, the near-surface precipitation is not assimilated but is to a large extent produced by the model which doesn’t change during the simulation period

## Supporting analysis on global hydro-climatological indicators

Figure A shows the 24-hr rainfall area  $A_p$  estimated the various reanalyses: NOAA 20CR, ERA 20C, NCEP1 and ERA5. These estimates show that NOAA 20CR is an outlier with a much larger 24-hr precipitation area than the other reanalyses, but also shows a clear downward sloping trend over the entire period. Neither NOAA 20CR nor ERA 20C involves satellite observations, and all reanalyses indicate a decrease in  $A_p$  to some degree after 1960 between 50 °S to 50 °N, as in (4). However, the global  $A_p$  curve for ERA 20C has remained flat since 1960.

Figure B compares the global and semi-global rainfall area based on data from ERA5 with corresponding data from the Tropical Rain Measurements Missions (TRMM) (5, 6). The comparison suggests similar levels for 3-hr data (roughly the time between each satellite overpass) and more intense events (10 mm/day), but the ERA5 data suggest a greater 24-hr global rainfall area for the threshold of 1 mm/day. This discrepancy may partly be due to the precipitation

being continuously accumulated over 24 hrs in ERA5 whereas the TRMM only gets a few snapshots each day. Also, different spatial resolution, data processing and the fact that one represents remote retrievals whereas the other is a product of model calculations are expected to play a role.

## Supporting wavelet analysis and results

Figure C presents wavelet energies associated with various spatial scales for different zones: the northern temperate, the tropics, and the southern temperate zones. The wavelet analysis indicates that it has primarily been in the tropics where there has been a shift in terms of scale and intensity from 1961–1900 to 1991–2020.

Figure D shows the time series of energies at different spatial scales from 1950 to 2020 when the precipitation fields over the whole Earth’s surface is considered. Analogously, Figures E–G compare the time series of energies at different spatial scales over the tropics and the two temperate zones respectively. For Fig. D and Figs. F– G, we have used the same scales on the y-axis, such that the three figures are easy to compare. However, for Fig. E, we had to change the y-axis scale of the left panel because of the larger amounts of energy in the tropics. The situation over the tropics is shown in Fig. E and it is rather similar to the one presented in Fig. D. There, the energy has been constant or decreased from 1950 until 1985 and differs from the results shown in Fig. D, but after 1985 the energies began to increase over time with a faster growth rate for smaller spatial scales, as can be seen in the right panel of Fig. E. The explanation is that we see an increase of the variability for the precipitation patterns, such that the wavelet coefficients tend to vary more and more over time, and a gradual passage of energy from larger spatial scales to smaller scales. The time series over the temperate zones are shown in Figs. F- G and in both zones there has been a gradual increase in the precipitation energy at all spatial scales, though it was less pronounced than in the tropics. In terms of percentage of the

daily total energy, the situation in the temperate zones does not show the same shift in energies from larger to smaller scales as in the tropics. If such a shift really took place, it was much less pronounced. In fact, the growth rates in the tropics were at least one order of magnitude greater than in the temperate zones and this underlines the importance of the change taking place in that region. The linear trends in the three regions TR, TN and TS are reported in Tab. A. It is worthwhile remarking that the observational system used in the data assimilation cycle of ERA5 is changing significantly on all three regions during the time period considered, especially after 1980 with the introduction of satellites into the data assimilation cycle. It would be reasonable to assume that the impact of variations in the observational system on the representation of precipitation should be the same (i.e. have similar magnitudes) over the three regions. The results show that in the two temperate regions the time series of energies do not show evident discontinuities. In particular, over the south temperate zone the growth of energies continued before and after 1980. However, in the tropics the time series of wavelet energies do not display the same continuity and we assume that the change in behaviour after 1985 can be related to a change in the precipitation regime, although we cannot assess the sensitivity of precipitation to variation in the observational system on the tropics and this should be a topic for further research.

Figure H presents a similar wavelet analysis as in the main paper but for the ERA20C reanalysis and shows a weaker shift between the periods 1961–1990 and 1991–2020. The ERA20C has a horizontal resolution of approximately 125 km (spectral truncation T159) and involved a coupled Atmosphere/Land-surface/Ocean-waves model to reanalyse the weather by assimilating surface observations. Hence, it did not involve satellite observations. The weaker shift in these results may be due to both a coarser spatial resolution as well as fewer observations used for assimilation.

Figures H– I show the results of the wavelet multi-resolution analysis for ERA20C and are

presented in the same layout as for ERA5 results. The comparison between 1961–1990 and 1991–2020 climatologies is shown in Fig. H. The time series of the wavelet energies are shown in Fig. I and this figure can be compared to Fig. D for ERA5. For ERA20C, as for ERA5, the results are also suggesting an increasing predominance of the smaller scales ( $L \sim 4^\circ$ ), but a reduction in the larger scales ( $L \sim 31^\circ$ ). The finer spatial resolution of ERA5 allows for a more detailed investigation of the shift in energies between the smaller spatial scales.

## References and Notes

1. Benestad, Rasmus, Lussana, Cristian, Dobler, Andreas, Hydro-climatological indicators (2022). Type: dataset.
2. U. Schulzweida, CDO User Guide: Climate Data Operator, Version 2.0.0, October 2021, *Tech. rep.*, MPI for Meteorology (2021).
3. H. Hersbach, *et al.*, *Quarterly Journal of the Royal Meteorological Society* **146**, 1999 (2020).
4. R. E. Benestad, *Environmental Research Letters* **13**, 044022 (2018).
5. J. Simpson, R. F. Adler, G. R. North, *Bulletin of the American Meteorological Society* **69**, 278 (1988).
6. C. Kummerow, W. Barnes, T. Kozu, J. Shiue, J. Simpson, *Journal of Atmospheric and Oceanic Technology* **15**, 809 (1998).

## **Figures**

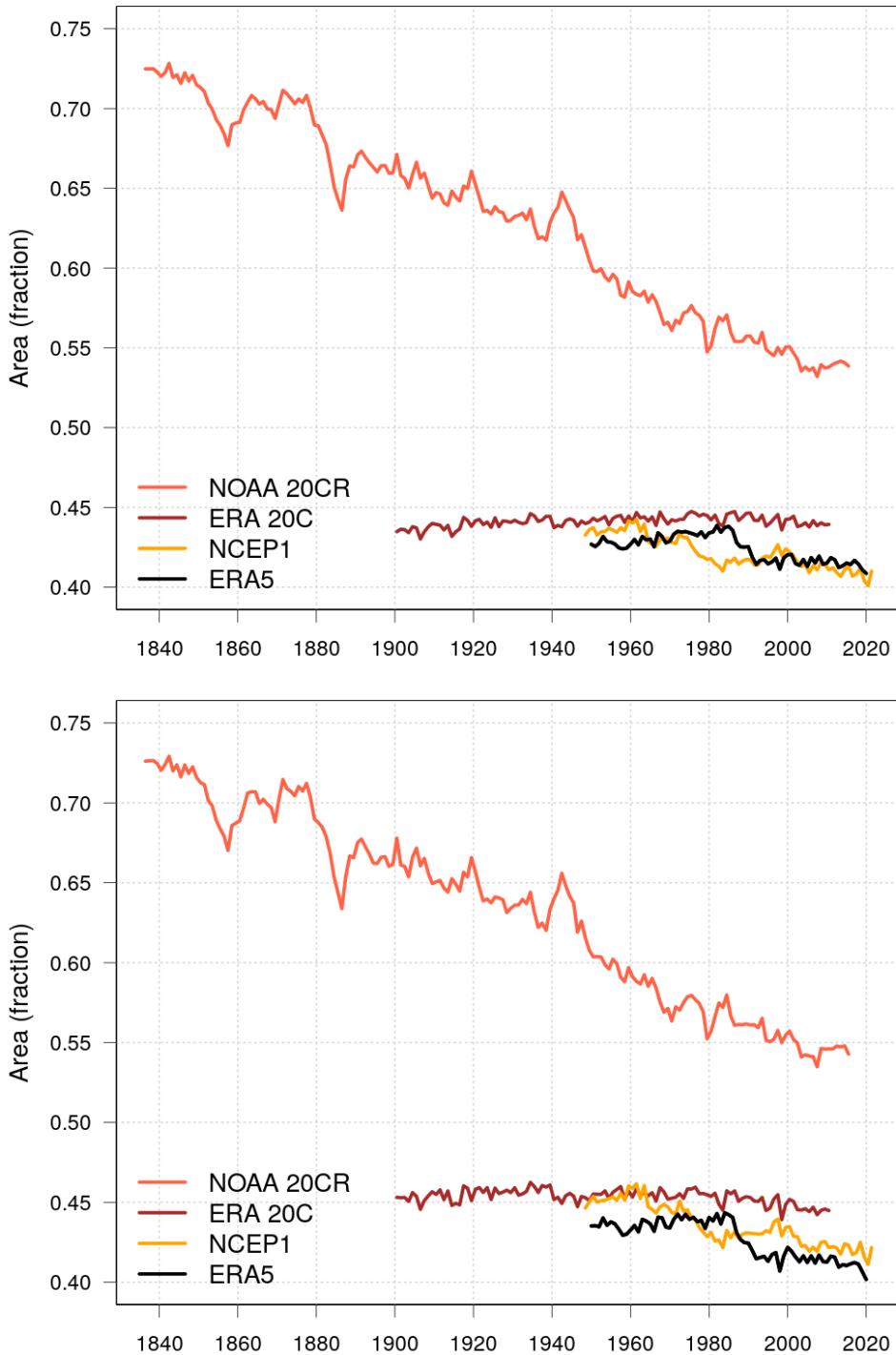


Figure A: Global (top) and 50 °S to 50 °N (bottom) annual mean rainfall surface area fraction from different reanalysis products from 24-hr estimates with a threshold of 1 mm/day. There are substantial variations between reanalyses, and the estimates are expected to be sensitive to both the representation of precipitation mechanisms, the data assimilation method, available data for data assimilation, the time step size and the spatial resolution of the models used in the reanalysis.

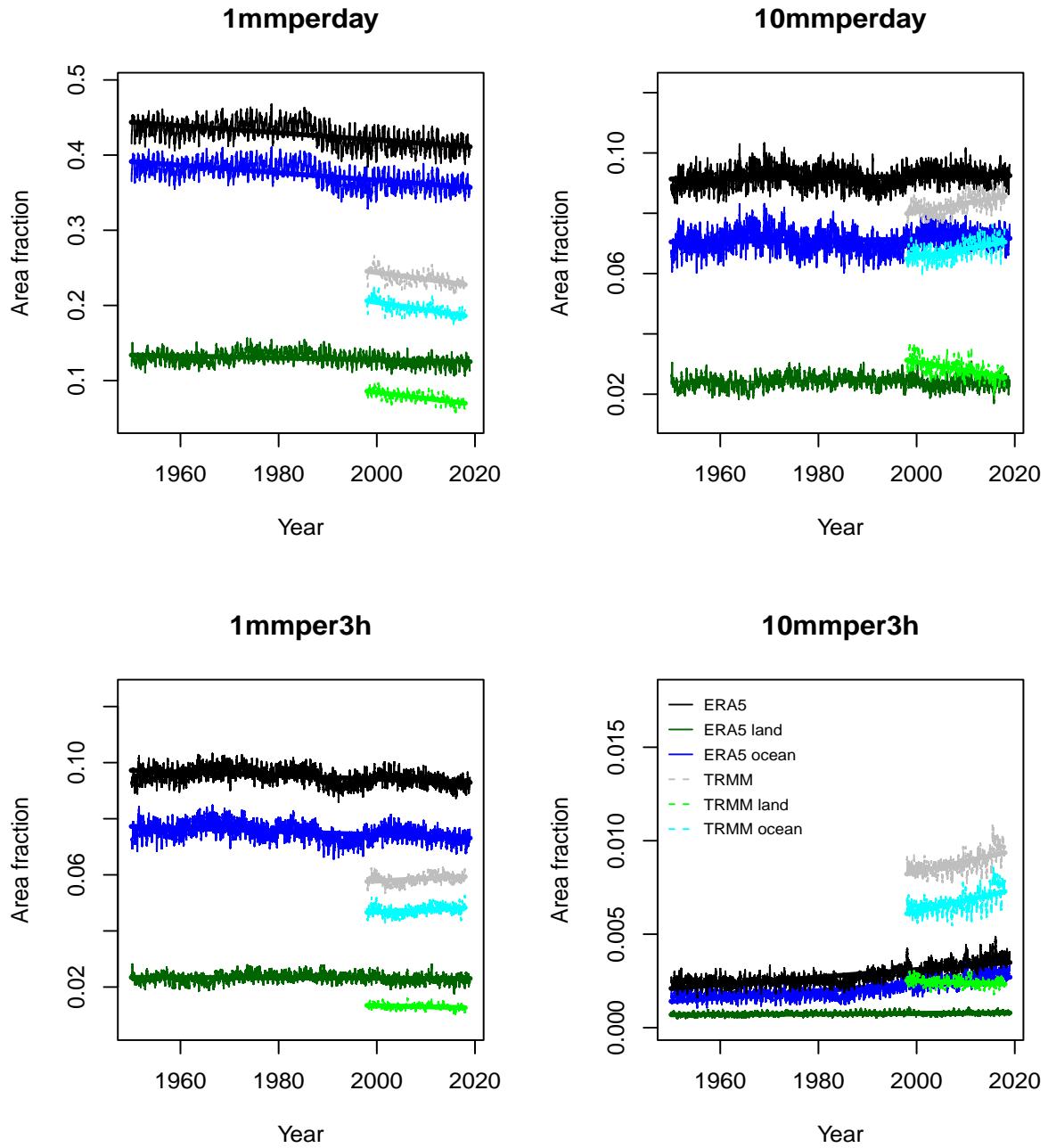


Figure B: Precipitation area from ERA5 compared to the satellite dataset TRMM. Area used is  $\pm 50^\circ$  latitude. The estimates for threshold of 1mm per day and 10mm per 3hr are questionable, but the 1 mm/day estimates were double-checked and verified with previous estimates from (4). It is not expected that the 3hr TRMM data will be accurate because they are snapshot samples for every overpass with a periodicity of approximately 3 hrs.

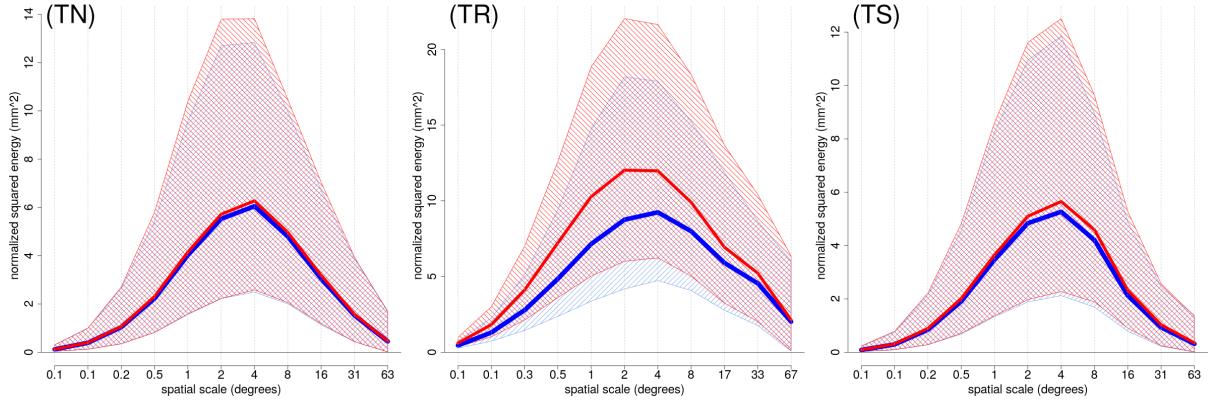


Figure C: Multi-resolution decomposition of daily precipitation fields based on 2-D Haar wavelet transform for the normal periods (blue 1961–1990, pink and red 1991–2020). The solid lines show the wavelet squared energy as a function of the spatial scale, whereas the envelope show the range of the 1st to 99th percentile. The panels show results over: the north temperate zone (TN, latitude in the range 23N – 67N); the tropics (TR, latitude in the range 25S – 25N); the south temperate zone (TS, latitude in the range 67S – 23S). The layout is similar to that of the inset in Fig. 6.

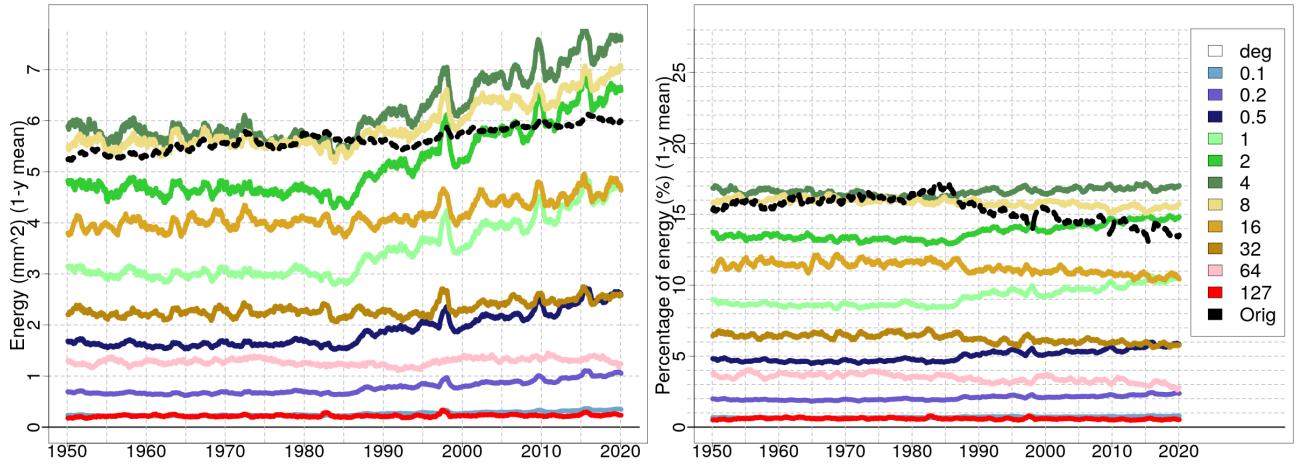


Figure D: Time series of energies of the 2D Haar wavelet coefficients for the multi-resolution analysis of ERA5 daily precipitation over the whole Earth's surface. For each day, the 1-year moving average centered on that day is shown. The left panel shows the energies. The right panel shows the time series as percentages of the total energy. The colors of the lines indicates the spatial scales (in degrees), which are reported on the legend in the right panel. The black line is the squared value of the mean of the precipitation field, that in the wavelet decomposition is the energy of the scaling function, and it can be considered as a reference value for the other energies.

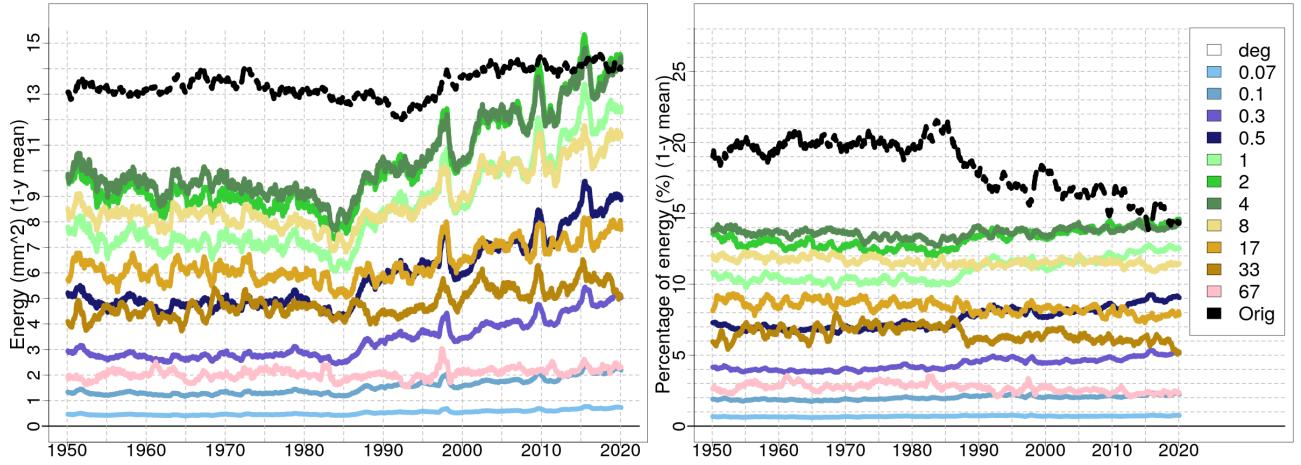


Figure E: Time series of energies of the 2D Haar wavelet coefficients for the multi-resolution analysis of ERA5 daily precipitation over the tropics (latitude 25S – 25N). For each day, the 1-year moving average centered on that day is shown. The left panel shows the energies. The right panel shows the time series as percentages of the total energy. The colors of the lines indicates the spatial scales (in degrees), which are reported on the legend in the right panel. The black line is the squared value of the mean of the precipitation field (i.e. the "Original" fields, without applying the multi-resolution decomposition), that in the wavelet decomposition is the energy of the scaling function, and it can be considered as a reference value for the other energies.

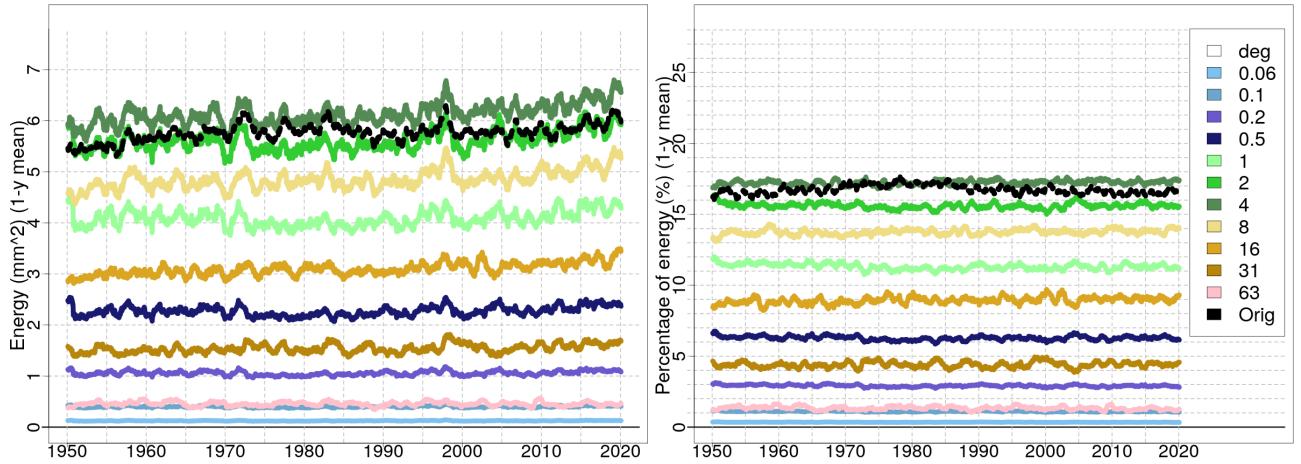


Figure F: Time series of energies of the 2D Haar wavelet coefficients for the multi-resolution analysis of ERA5 daily precipitation over the north temperate zone (latitude in the range 23N – 67N). The layout is the same as for Fig. D.

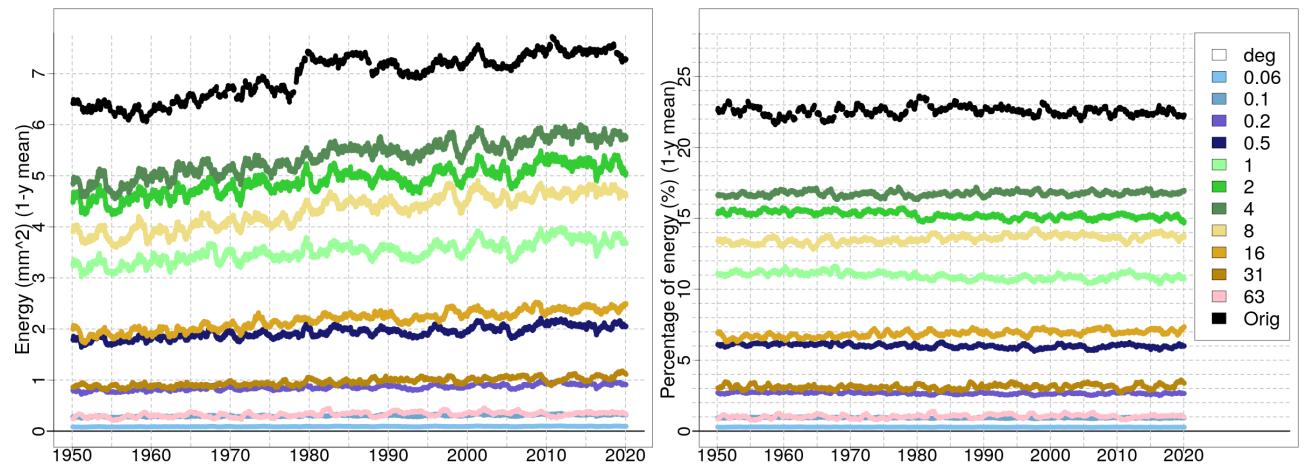


Figure G: Time series of energies of the 2D Haar wavelet coefficients for the multi-resolution analysis of ERA5 daily precipitation over the south temperate zone (latitude in the range 67S – 23S). The layout is the same as for Fig. D.

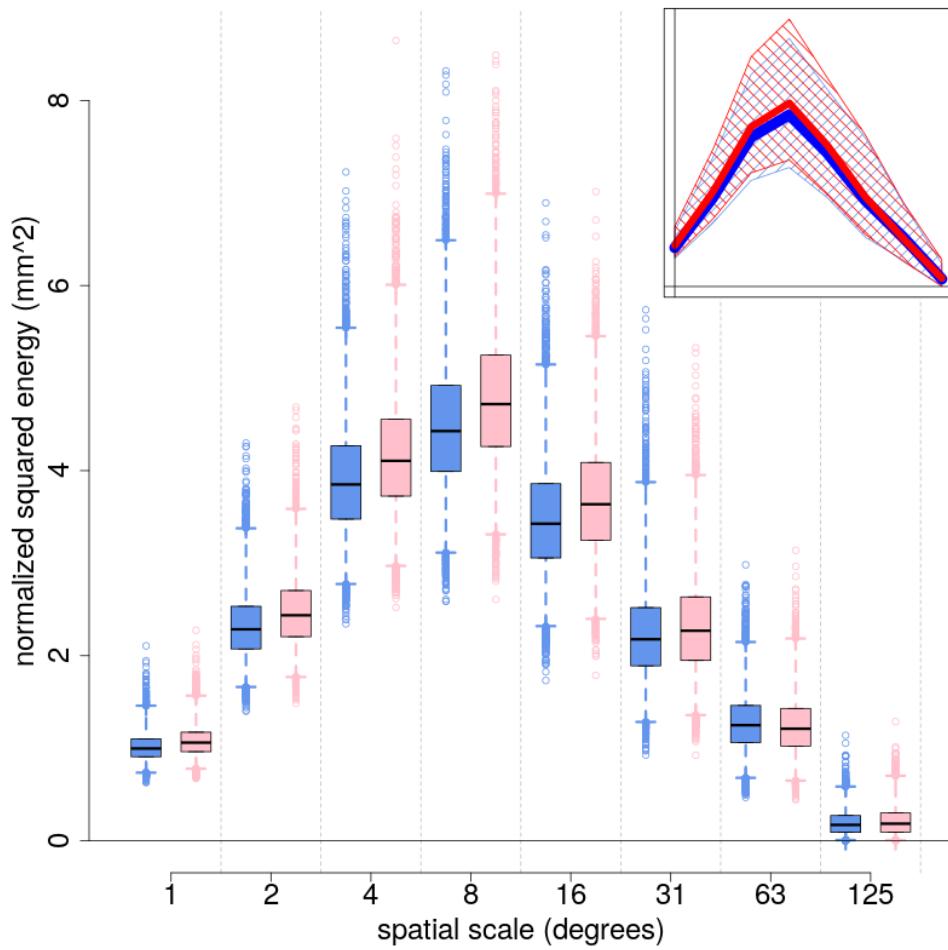


Figure H: ERA20C multi-resolution decomposition of daily precipitation fields based on 2-D Haar wavelet transform (blue 1961–1990, pink and red 1991–2020).

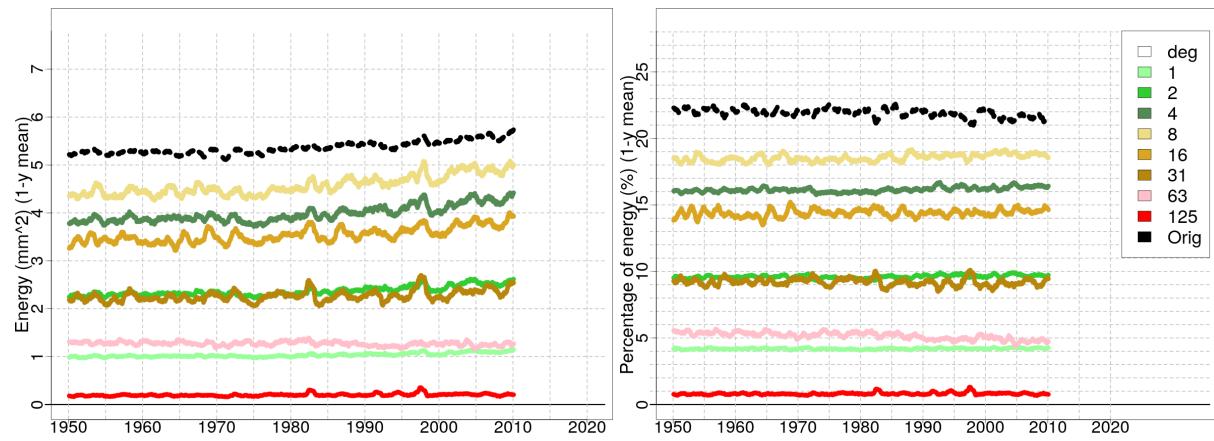


Figure I: Time series of energies of the 2D Haar wavelet coefficients for the multi-resolution analysis of ERA20C daily precipitation over the whole Earth's surface. The layout is the same as for Fig. D.

spatial scale ( $l$ ) (degrees)	$q_{50}(\text{En}^2_l)$ relative change			$q_{50}(\text{En}^2_l/\text{En}^2_{\text{tot}})$ relative change			linear trend 1991–2020 (mm $^2$ /10year)		
	(%)			(%)			TR	TN	TS
	TR	TN	TS	TR	TN	TS			
0.25	+50	+3	+5	+13	0	-2	0.56	0.02	0.02
0.5	+52	+4	+5	+15	0	-2	1.05	0.07	0.05
1	+47	+2	+5	+11	0	-2	1.42	0.12	0.09
2	+40	+3	+5	+6	0	-2	1.55	0.16	0.10
4	+32	+3	+8	-1	0	0	1.41	0.15	0.10
8	+25	+3	+10	-6	0	2	0.99	0.13	0.08
16	+18	+4	+10	-11	0	2	0.57	0.08	0.07
32	+13	+2	+10	-14	-1	4	0.39	0.04	0.02
64	+3	+2	+20	-21	0	10	0.12	-0.01	-0.01

Table A: Summary of the comparison of the multi-resolution decomposition squared energies over three regions on the globe between 1961–1990 and 1991–2020. The three regions are: the north temperate zone (TN, latitude range: 23 °N–67 °N); the tropics (TR, latitude range: 25 °S–25 °N); the south temperate zone (TS, latitude range: 67 °S–23 °S)). The relative change for  $q_{50}(\text{En}^2_l)$  and the linear trend have been defined in Table 1. The relative change for  $q_{50}(\text{En}^2_l/\text{En}^2_{\text{tot}})$  has been defined in Table 2. Note that only spatial scales greater than or equal to 0.25 degrees are shown.

## R-markdown

The following section contains the output of our R-markdown script that contains R-codes of the analysis presented herein, raw output of the execution of the code, and some further explanations. It also provides the numbers reported in the main manuscript. It is included here for the purpose of transparency. The R-markdown scrip itself is available from the FigShare repository together with essential data and allow repeated execution independently of ours.

# ERA5 rain area

REB

4/7/2021

## R Markdown

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

The analysis presented in the paper was carried out through the Climate Data Operators (CDO; <https://code.mpimet.mpg.de/projects/cdo>) and the R-environment <https://cran.r-project.org> (R version 4.1.2 (2021-11-01) – “Bird Hippie” on an Ubuntu platform). The data used here and this very R-markdown script (the source for this PDF file) are available from the FigShare.com repository [https://figshare.com/articles/dataset/Hydro-climatological\\_indicators/17158133](https://figshare.com/articles/dataset/Hydro-climatological_indicators/17158133) (DOI: 10.6084/m9.figshare.17158133.v7). The data on FigShare consist of the following tar-archive:

```
cd ~/Downloads  
tar cvzf hydro-climatological-indicators.tar.gz ERA5_tp_wavelet.power.rda \  
global-precip-area.nc era5_50s50n-precip-area.nc global-precip-mean.nc \  
era5_50s50n-precip-mean.nc global-wet-mean-precip.nc global-precip-area.nc \  
yearmean_fldmean_gec1mm_era20c_tp_daily.nc yearmean_fldmean_gec1mm_NOAA_20CRv3_tp_daily.nc \  
yearmean_fldmean_gec1mm_NCEP_RA1_tp_daily.nc era5_50s50n-precip-area_new.nc \  
fldmean_50S_to_50N/yearmean_fldmean_gec1mm_era20c_tp_daily.nc \  
fldmean_50S_to_50N/yearmean_fldmean_gec1mm_NOAA_20CRv3_tp_daily.nc \  
fldmean_50S_to_50N/yearmean_fldmean_gec1mm_NCEP_RA1_tp_daily.nc \  
global-evap-mean.nc global-olr-mean.nc global-sw+lw-mean.nc global-energy-mean.nc \  
TRMM_Oskar.nc tp.area/* rainarea-era5.Rmd
```

The data can be downloaded onto a local computer with

```
cd Downloads ## Go to the directory where you want the files stored  
## You can use a browser to download the files:  
firefox https://figshare.com/ndownloader/files/31723382  
## Unpack the data  
tar xvf hydro-climatological-indicators.tar.gz
```

Alternatively, you can retrieve to files by pasting <https://figshare.com/ndownloader/files/31723382> into your browser (e.g. on windows platforms) or clicking on this link.

The files `ERA5_mu_year_1950-2020.nc` and `ERA5_fw_year_1950-2020.nc` were not included here due to their large volumes (282Mbytes), but the instructions for generating them from daily ERA5 total precipitation are provided below. The ERA5 data was downloaded from Copernicus C3S: <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview>.

The R-package `esd` is open-source and freely available from <https://github.com/metno/esd> and the wiki-page on this site provides instructions about installation as well as links to documentation. The `RColorBrewer` and `Kendall` are available from <https://cran.r-project.org> and can be installed directly through R-studio.

```

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.default base
##   subset.matrix  base
##   subset.zoo     zoo

library(RColorBrewer)
library(Kendall)

```

## Data processing

We used CDO to process the data and calculate the annual wet-day mean precipitation and the wet-day mean frequency from daily ERA5 total precipitation:

```

cdo -O -b f32 daysum ERA5_tp_${YEAR}.nc ERA5_tp_${YEAR}_daysum.nc
cdo mergetime ERA5_tp_${YEAR}_daysum.nc ERA5_tp_day_1950-2020.nc
cdo gtc,0.001 ERA5_tp_day_1950-2020.nc mumask.nc
cdo ifthen mumask.nc ERA5_tp_day_1950-2020.nc wetdaymean.nc
cdo yearmean wetdaymean.nc mu_year.nc
cdo yearmean mumask.nc fw_year.nc
rm mumask.nc

```

CDO was also used to calculate the hydro-climatological indicators from the reanalyses:

```

cdo -fldmean -gec,0.001 ERA5_tp_day_1950-2020.nc global-precip-area.nc
cdo -fldmean ERA5_tp_day_1950-2020.nc global-precip-mean.nc
cdo fldmean -setrmiss,-Inf,0.001 ERA5_tp_day_1950-2020.nc global-wet-mean-precip.nc
## Semi-global
cdo -fldmean -gec,0.001 -sellonlatbox,-180,180,-50,50 ERA5_tp_day_1950-2020.nc
                           era5_50s50n-precip-area.nc

```

## Functions

The function `areatest` was used in the calculations below and to provide summary statistics. The function `walkertest` was used to test for ‘field significance,’ i.e. when there are multiple hypothesis testings for statistical significance involving the p-value, e.g. for many grid boxes.

```

## This function estimates the proportion of area with increasing trends
areatest <- function(x) {
  d <- dim(x); y <- x
  for (i in 1:d[1]) {
    x[i,] <- (x[i,>0])*cos(pi*lat(x)/180)
    y[i,] <- cos(pi*lat(x)/180)
  }
}
```

```

    }
    return(round(100*sum(x,na.rm=TRUE)/sum(y,na.rm=TRUE)))
}

## Test for field significance
walkertest <- function(x,alpha.global=0.05) {
  ## Based on Wilks, 2006; DOI: 10.1175/JAM2404.1
  nr <- length(x)
  p.Walker <- 1 - exp( log(1 - alpha.global)/nr )
  print(paste('Passes Walker test: ',min(x,na.rm=TRUE) < p.Walker))
}

## Plot trend
plottrend <- function(x,y,col="red") {
  xyear = seq(1,by=by,length.out=length(x))
  trendlm = lm(y ~ x)
  trendpval = summary(trendlm)$coefficients[2,4] # Pr(>/t|)
  print(paste("Pr(>|t|)      :",trendpval))
  print(paste("Slope [/yr] :",signif(trendlm$coefficients[2],3)))
  if (trendpval < 0.01) { lwd = 3 } else { lwd = 1 }
  matlines(as.Date(paste0(x[c(1,length(x))],"-01-01")),trendlm$coefficients[1] + trendlm$coefficients[2],
  ,col=col, lwd = lwd)
}

## Moving average
ma <- function(x,n=5,sides=2){
  filter(x,rep(1/n,n), sides=sides)
}

```

## Maps of mean conditions and trends

### Wet-day mean precipitation

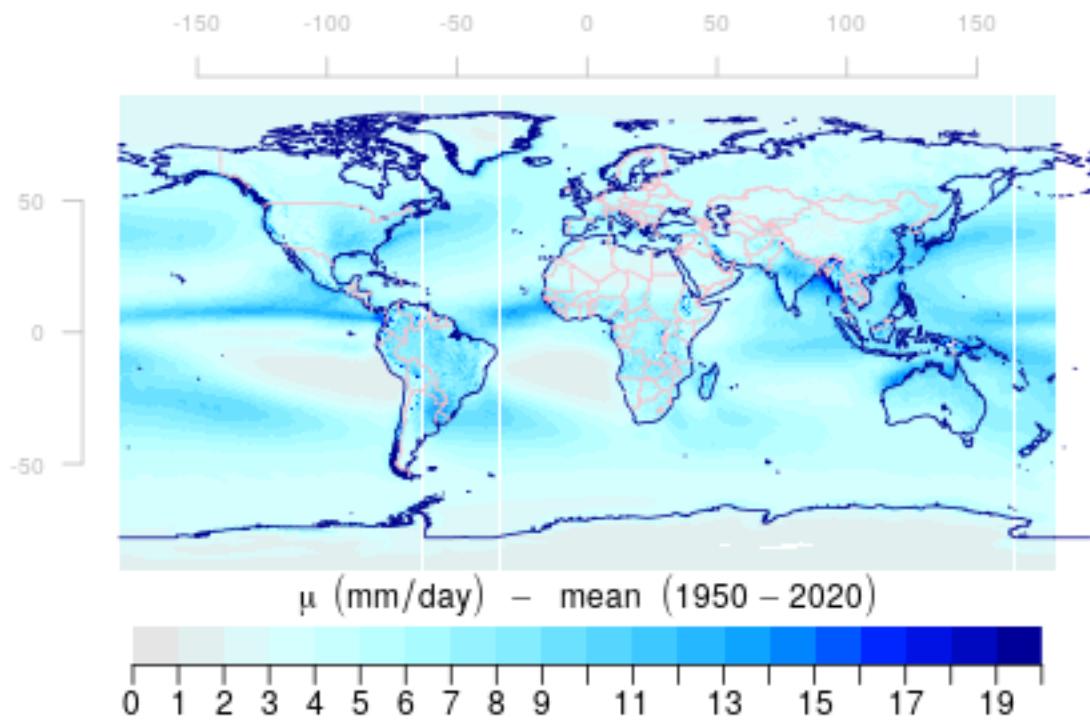
The mean

```

mu <- retrieve '~/Downloads/ERA5_mu_year_1950-2020.nc'*1000 # units in mm

## [1] "Warning : Frequency found in the attribute does not match the frequency detected in data"
attr(mu,'variable') <- 'mu'
attr(mu,'units') <- 'mm/day'
class(mu) <- c('field','annual','zoo')
map(mu,type='fill',new=FALSE,colbar=list(breaks=seq(0,20,by=1))) -> mmu

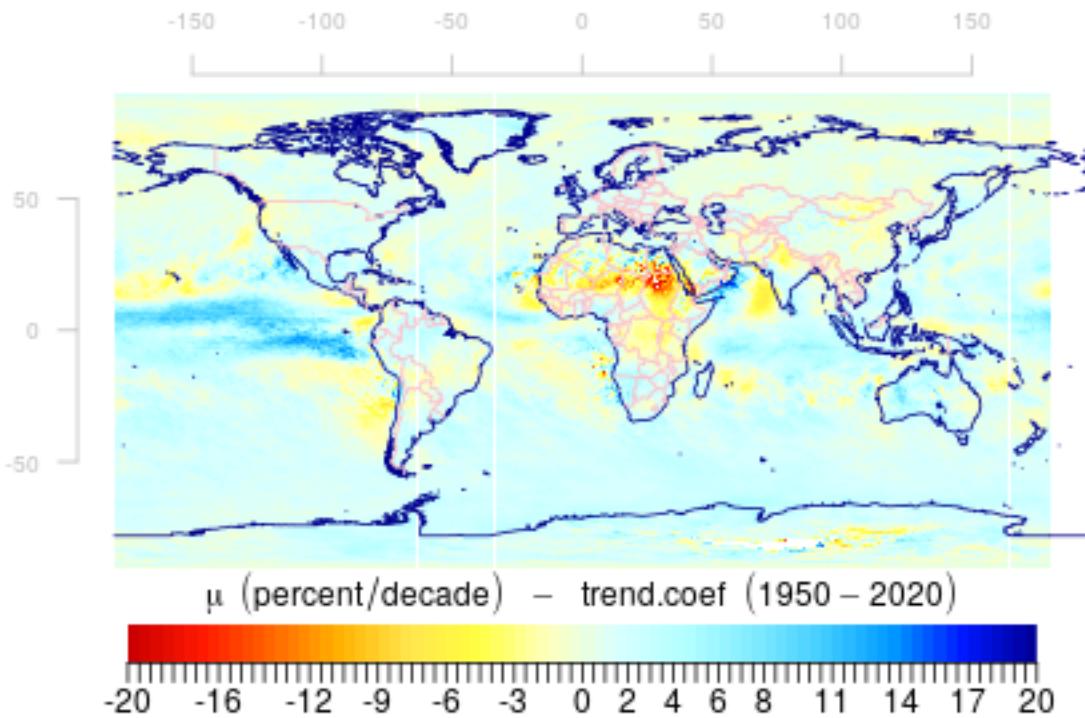
```



```

map(mu,FUN='trend',plot=FALSE) -> tmu
ptmu <- 100*tmu/mmu
attr(ptmu,'unit') <- 'percent/decade'
class(ptmu) <- class(mmu)
map(ptmu,colbar=list(pal='t2m',breaks=seq(-20,20,by=0.5),rev=TRUE),type='fill',new=FALSE)

```



```

print('Proportion of global area with increase in wet-day mean precipitation according to ERA5:')

## [1] "Proportion of global area with increase in wet-day mean precipitation according to ERA5:"

print('period: 1950-2020')

## [1] "(period: 1950-2020)"
print(areatest(tmu))

## [1] 78

pmu <- map(mu,FUN='trend.pval',plot=FALSE)
walkertest(pmu)

## [1] "Passes Walker test: TRUE"
## The same story for 50S-50N
print('50S-50N:')

## [1] "50S-50N:"
print(areatest(subset(tmu,is=list(lon=c(-180,360),lat=c(-50,50)))))

## [1] 76
walkertest(subset(pmu,is=list(lon=c(-180,360),lat=c(-50,50))))

## [1] "Passes Walker test: TRUE"

```

Fig. 1 - raw version also showing all trend estimates - also those that are not statistically significant. Fig.1 in

the main paper is based on these results.

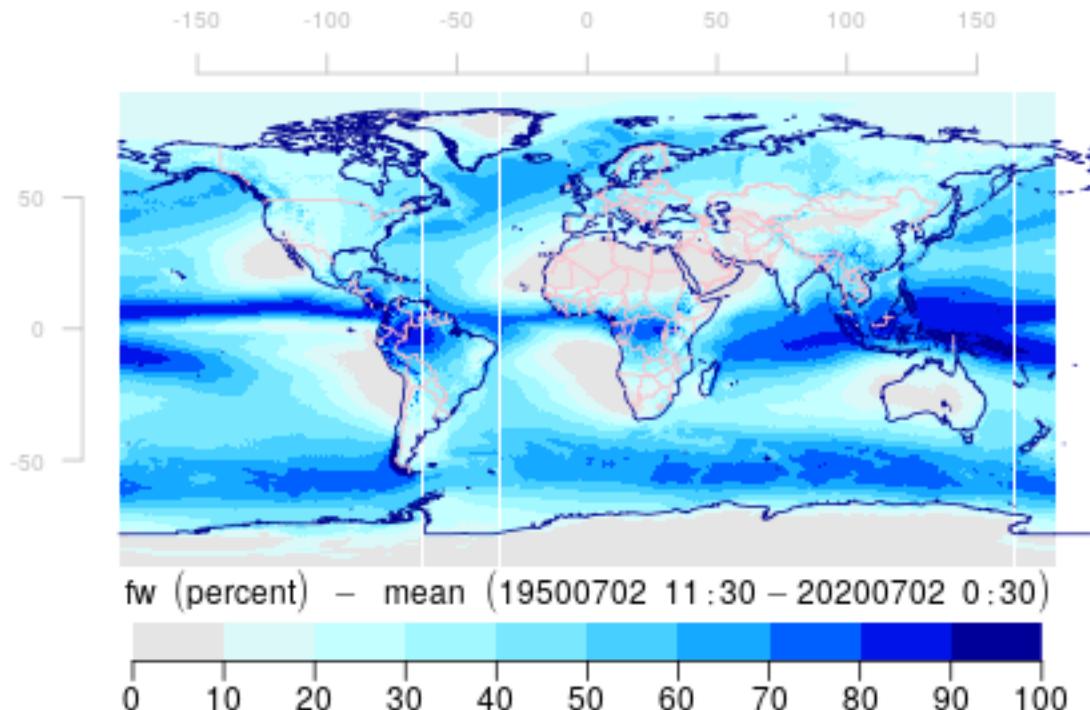
```
gmmu <- aggregate.area(mu,FUN='mean')
attr(gmmu,'unit') <- 'mm/day'
plot(gmmu, map.show=FALSE)
```

The global mean  $\langle \mu \rangle$  estimated from annual mean  $\mu$  for the respective ERA5 grid points indicate a shift in the trend character after 1985. .

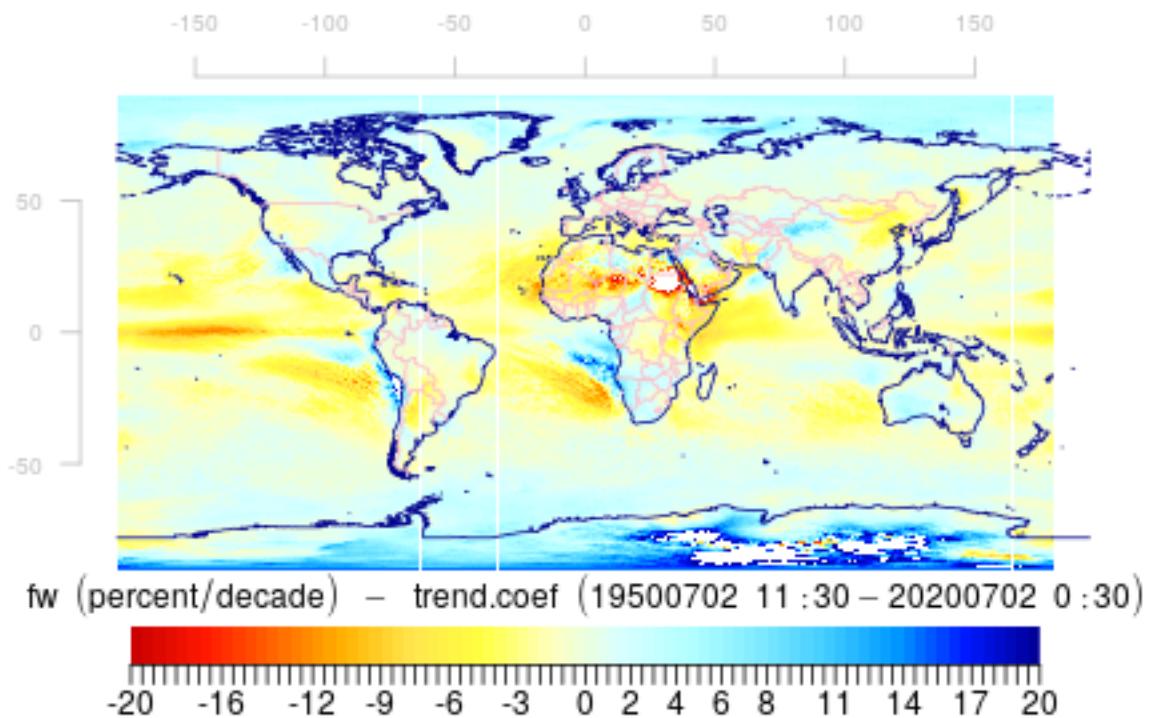
### Wet-day frequency

```
fw <- 100*retrieve('~/Downloads/ERA5_fw_year_1950-2020.nc')

## [1] "Warning : Frequency found in the attribute does not match the frequency detected in data"
attr(fw,'variable') <- 'fw'
attr(fw,'units') <- 'percent'
map(fw,type='fill',new=FALSE) -> mfw
```



```
map(fw,FUN='trend',colbar=list(pal='t2m',breaks=seq(-0.06,0.06,by=0.01),rev=TRUE),type='fill',plot=FALSE)
ptfw <- 100*tfw/mfw
attr(ptfw,'unit') <- 'percent/decade'
map(ptfw,colbar=list(pal='t2m',breaks=seq(-20,20,by=0.5),rev=TRUE),type='fill',new=FALSE)
```



```

print('Proportion of global area with increase in wet-day frequency:')

## [1] "Proportion of global area with increase in wet-day frequency:"
print('period: 1950-2020')

## [1] "(period: 1950-2020)"
print(areatest(tfw))

## [1] 39

pfw <- map(fw,FUN='trend.pval',plot=FALSE)
walkertest(pfw)

## [1] "Passes Walker test: TRUE"
## The same story for 50S-50N
print('50S-50N:')

## [1] "50S-50N:"
print(areatest(subset(tfw,is=list(lon=c(-180,360),lat=c(-50,50)))))

## [1] 26
walkertest(subset(tfw,is=list(lon=c(-180,360),lat=c(-50,50)))))

## [1] "Passes Walker test: TRUE"

```

Fig. 2 - raw version also showing all trend estimates - also those that are not statistically significant. Fig.1 in

the main paper is based on these results.

The global mean of the annual wet-day frequency

```
gmfw <- aggregate.area(fw,FUN='mean')
attr(gmfw,'unit') <- '%'
plot(gmfw)
```

There has been a slight but marked shift in global mean  $\langle f_w \rangle$  from 43% before 1985 to about 41.5 after 1990 and the period when new satellite retrievals were included in the data assimilation.

## Global hydro-climate indicators

```
print('*** Area: ***')

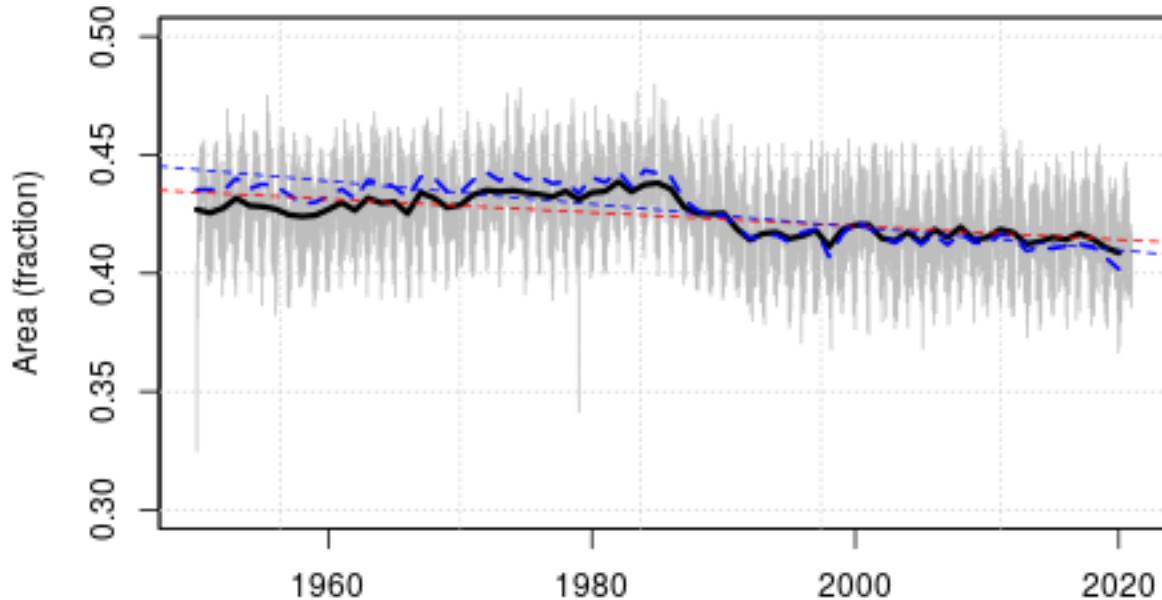
## [1] "*** Area: ***"

ncid <- nc_open '~/Downloads/global-precip-area.nc')
A <- ncvar_get(ncid, 'tp')
t <- as.Date(ncvar_get(ncid, 'time')/24, origin='1900-01-01')
nc_close(ncid)
A <- zoo(x=A,order.by=t)

## 50S-50N:
ncid <- nc_open '~/Downloads/era5_50s50n-precip-area.nc')
a <- ncvar_get(ncid, 'tp')
t <- as.Date(ncvar_get(ncid, 'time')/24, origin='1900-01-01')
nc_close(ncid)
a <- zoo(x=a,order.by=t)

plot(A, main='Fractional global rainfall area per day', ylab='Area (fraction)', xlab='',
      sub='source: ERA5', ylim=c(0.3,0.5), col='grey')
grid()
A <- year2date(annual(A))
a <- year2date(annual(a))
it <- (year(a) >= 1998) & (year(a) <= 2016)
lines(A, lwd=3)
lines(a, col='blue', lty=2, lwd=2)
call1 <- data.frame(A=coredata(A), t=index(A), a=coredata(a))
trendfit1 <- lm(A ~ t, data=call1)
trndfit1 <- lm(a ~ t, data=call1)
abline(trendfit1, col='red', lty=2)
abline(trndfit1, col='blue', lty=2)
```

## Fractional global rainfall area per day



source: ERA5

```
print(summary(trendfit1))

##
## Call:
## lm(formula = A ~ t, data = cal1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.0092422 -0.0044785 -0.0009774  0.0035265  0.0138773 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.285e-01 8.558e-04 500.708 < 2e-16 ***
## t          -7.827e-07 9.226e-08 -8.484 2.61e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005819 on 69 degrees of freedom
## Multiple R-squared:  0.5105, Adjusted R-squared:  0.5034 
## F-statistic: 71.97 on 1 and 69 DF,  p-value: 2.61e-12
print(predict(trendfit1,newdata=data.frame(t=as.Date(c('1950-01-01','2020-12-31')))))

##
##           1         2
## 0.4342296 0.4139329

## Change as in the TRMM
print(predict(trndfit1,newdata=data.frame(t=as.Date(c('1998-01-01','2016-12-31')))))
```

```

##           1           2
## 0.4204710 0.4112344

Fig. 3.

print('*** Total rainfall: ***')

## [1] "*** Total rainfall: ***"

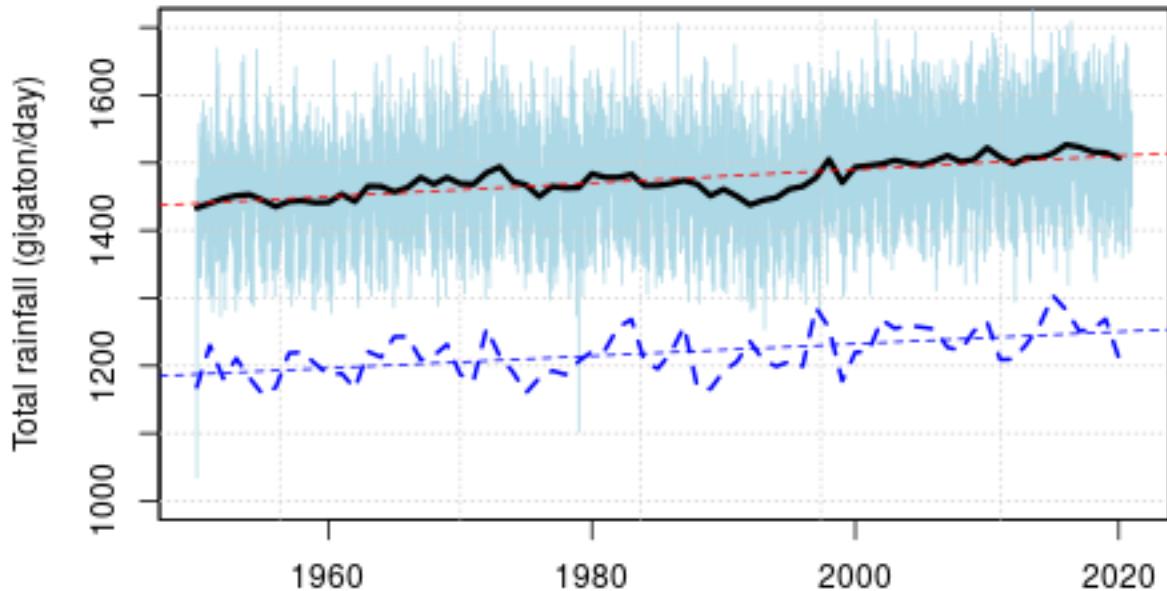
ncid <- nc_open '~/Downloads/global-precip-mean.nc'
tp <- ncvar_get(ncid,'tp') * 4*pi*6371000^2 # Mean precipitation per m^2 times total area
t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
tp <- zoo(x=tp/1.0e9,order.by=t)

ncid <- nc_open '~/Downloads/era5_50s50n-precip-mean.nc'
xtp <- ncvar_get(ncid,'tp') * 4*pi*6371000^2 * 0.77 # Mean precipitation per m^2 times total area
t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
xtp <- zoo(x=xtp/1.0e9,order.by=t)

plot(tp, main='Global rainfall amount per day',ylab='Total rainfall (gigaton/day)',xlab='',
      sub='source: ERA5',ylim=c(1000,1700),col='lightblue')
grid()
tp <- year2date(annual(tp))
xtp <- year2date(annual(xtp))
lines(tp,lwd=3)
lines(xtp,lwd=2,lty=2,col='blue')
cal2 <- data.frame(tp=coredata(tp),t=index(tp),xtp = xtp)
trendfit2 <- lm(tp ~ t, data=cal2)
trndfit2 <- lm(xtp ~ t, data=cal2)
abline(trendfit2,col='red',lty=2)
abline(trndfit2,col='blue',lty=2)

```

## Global rainfall amount per day



source: ERA5

```
print(summary(trendfit2))

##
## Call:
## lm(formula = tp ~ t, data = cal2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -44.534  -7.406   3.315   8.463  30.714 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.460e+03  2.167e+00  673.90 <2e-16 ***
## t            2.748e-03  2.336e-04   11.76 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.73 on 69 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6625 
## F-statistic: 138.4 on 1 and 69 DF,  p-value: < 2.2e-16

print(predict(trendfit2,newdata=data.frame(t=as.Date(c('1950-01-01','2020-12-31')))))

##
##      1      2
## 1440.059 1511.316

print(predict(trndfit2,newdata=data.frame(t=as.Date(c('1998-01-01','2016-12-31')))))
```

```

##          1          2
## 1230.517 1247.577

Fig. 4.

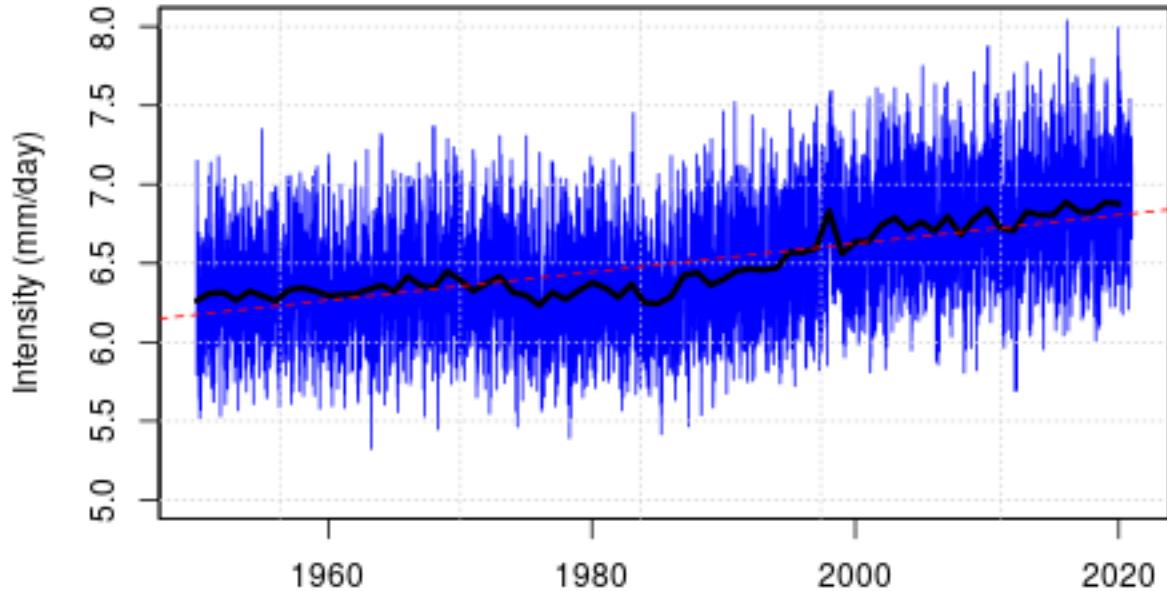
print('*** Intensity: ***')

## [1] "*** Intensity: ***"

ncid <- nc_open '~/Downloads/global-wet-mean-precip.nc'
mu <- ncvar_get(ncid,'tp')
t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
mu <- zoo(x=mu,order.by=t)
plot(mu*1000, main='Global wet-mean precipitation',ylab='Intensity (mm/day)',xlab='',
     sub='source: ERA5 (threshold=1 mm)',ylim=c(5,8),col='blue')
grid()
mu <- year2date(annual(mu))
lines(mu*1000,lwd=3)
cal3 <- data.frame(mu=coredata(mu)*1000, t=index(mu))
trendfit3 <- lm(mu ~ t, data=cal3)
abline(trendfit3,col='red',lty=2)

```

**Global wet-mean precipitation**



source: ERA5 (threshold=1 mm)

```

print(summary(trendfit3))

##
## Call:
## lm(formula = mu ~ t, data = cal3)

```

```

## 
## Residuals:
##      Min       1Q   Median      3Q     Max
## -0.24683 -0.08024  0.02512  0.07354  0.22485
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.355e+00 1.478e-02 429.92 <2e-16 ***
## t          2.466e-05 1.594e-06 15.48 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1005 on 69 degrees of freedom
## Multiple R-squared:  0.7763, Adjusted R-squared:  0.7731
## F-statistic: 239.5 on 1 and 69 DF,  p-value: < 2.2e-16
print(predict(trendfit3,newdata=data.frame(t=as.Date(c('1950-01-01','2020-12-31')))))

##           1           2
## 6.174801 6.814310

```

Fig. 5.

## Other reanalyses

**TRMM and ERAINT** We compared  $A_p$  with similar estimates for both TRMM and ERAINT from Benestad (2018) as the published results were available from the `preciparea` package available from <https://github.com/brasmus/preciparea>

```

## Install preciparea from GitHub:
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)
}
library(devtools)

## Loading required package: usethis
## Install the package if it's not already installed:
install.preciparea <- ("preciparea" %in% rownames(installed.packages()) == FALSE)

if (install.preciparea) {
  print('Need to install the devtools package')
  ## You need online access.
  install_github('brasmus/preciparea')
}

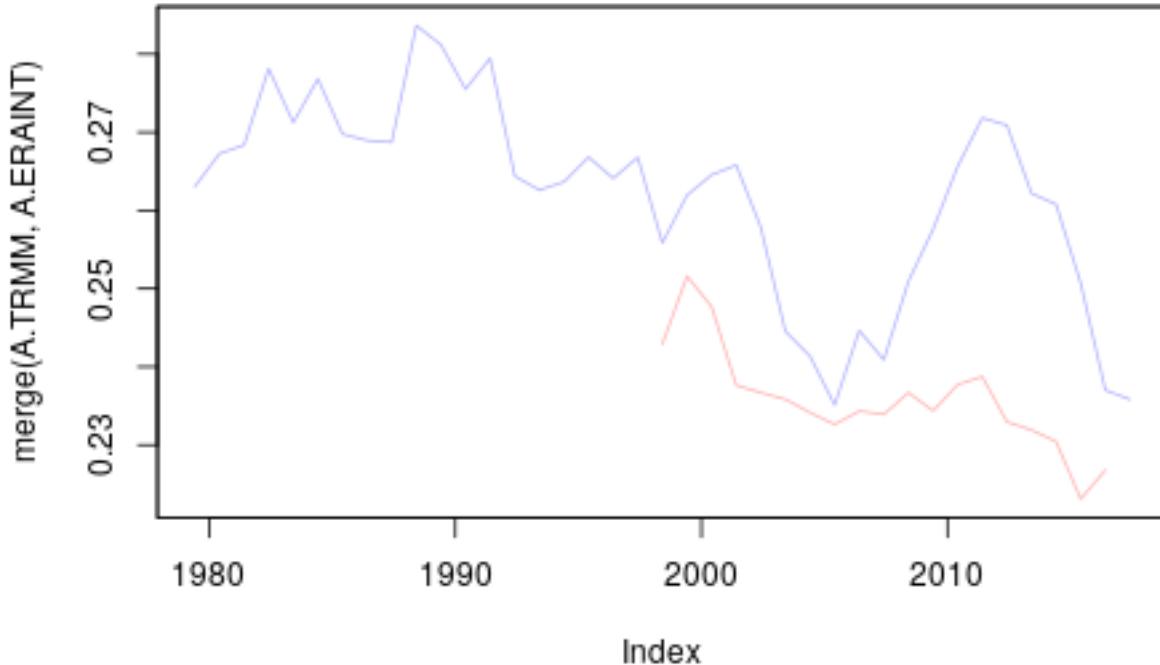
library(preciparea)
data("Parea") ## Area in square kms
data("tpa.eraint")
## Earth's radius <https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html>
re <- 6.371e3 ## Area in square kms
Ae <- 0.77* 4*pi*re^2 ## 50S-50N represents 77% of Earth surface area

```

```

A.TRMM <- annual(Parea[,1])/Ae
A.ERAINT <- annual(tpa.eraint[,3])/(Ae*1e6) ## m^2 -> fractional area
index(A.TRMM) <- as.Date(paste0(year(A.TRMM), '-06-01'))
index(A.ERAINT) <- as.Date(paste0(year(A.ERAINT), '-06-01'))
plot(merge(A.TRMM,A.ERAINT),plot.type='single',col=c(rgb(1,0,0,0.3),rgb(0,0,1,0.3)))

```



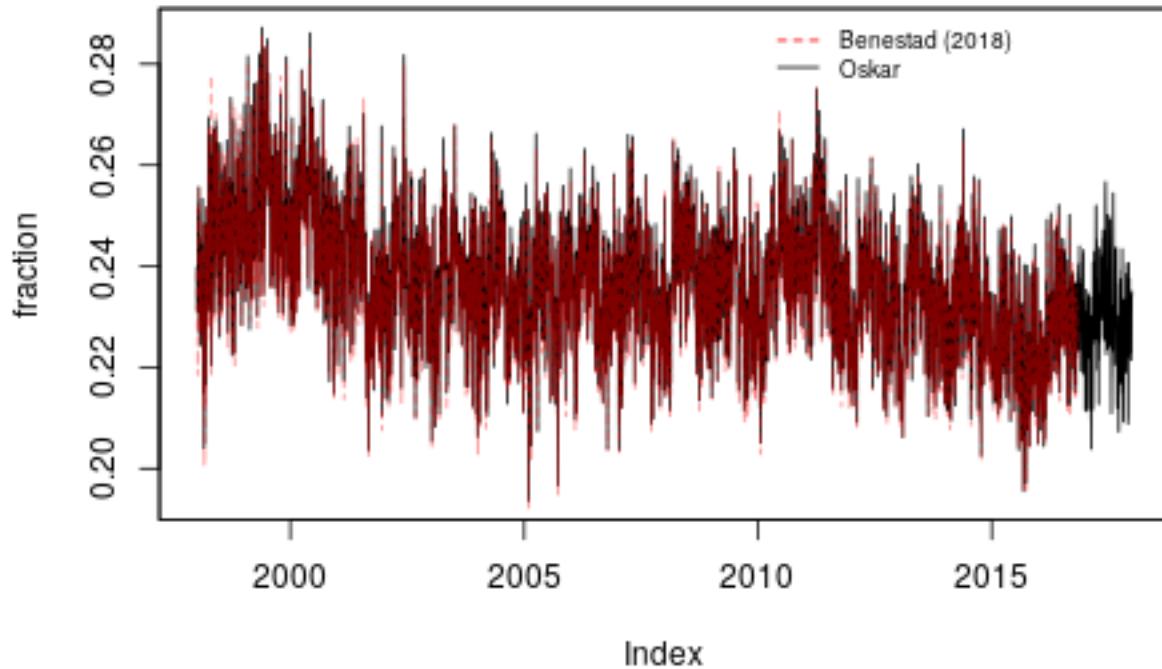
Do the comparison:

```

ncid <- nc_open '~/Downloads/TRMM_Oskar.nc')
trmm <- ncvar_get(ncid, 'precipitation')
t <- round(ncvar_get(ncid, 'time'))
t <- as.Date(paste(substr(as.character(t), 1, 4), substr(as.character(t), 5, 6), substr(as.character(t), 7, 8), substr(as.character(t), 9, 10), sep = '-'))
nc_close(ncid)
trmm.oskar <- zoo(x=trmm,order.by=t)
plot(trmm.oskar,main='Check TRMM: Precipitation (> 1mm/day) area',ylab='fraction')
lines(Parea[,1]/Ae,lty=2,col=rgb(1,0,0,0.5))
legend(as.Date('2010-01-01'),0.29,c('Benestad (2018)', 'Oskar'),col=c('red','black'),lty=c(2,1),bty='n',cex=0.8)

```

## Check TRMM: Precipitation (> 1mm/day) area



The fraction on  $A_p$  presented in Benestad (2018) using PMEL Ferret are identical to those estimated by Oskar based on CDO.

The chunk of R-code below presents an analysis with different threshold values and for both semi-global, ocean and land areas for ERA5 and TRMM.

```
#navg = 90 # 90-day moving average. A function defined above.
par(mfrow=c(2,2))
for (threshold in c("1mmperday", "10mmperday", "1mmper3h", "10mmper3h")) {
  print(threshold)

  ncfile = paste0("~/Downloads/tp.area/", threshold, ".lat50.land/cat.1950-2018.nc")
  nc = nc_open(ncfile)
  ERA5.land = ncvar_get(nc, "tp")
  nc_close(nc)

  ncfile = paste0("~/Downloads/tp.area/", threshold, ".lat50.ocean/cat.1950-2018.nc")
  nc = nc_open(ncfile)
  ERA5.ocean = ncvar_get(nc, "tp")
  nc_close(nc)

  ncfile = paste0("~/Downloads/tp.area/", threshold, ".lat50.global/cat.1950-2018.nc")
  nc = nc_open(ncfile)
  ERA5.global = ncvar_get(nc, "tp")
  nc_close(nc)
```

```

ncfile = paste0("~/Downloads/tp.area/TRMM/",threshold,".lat50.global/cat.1998-2017.nc")
nc = nc_open(ncfile)
TRMM.global = ncvar_get(nc,"precipitation")
nc_close(nc)

ncfile = paste0("~/Downloads/tp.area/TRMM/",threshold,".lat50.ocean/cat.1998-2017.nc")
nc = nc_open(ncfile)
TRMM.ocean = ncvar_get(nc,"precipitation")
nc_close(nc)

ncfile = paste0("~/Downloads/tp.area/TRMM/",threshold,".lat50.land/cat.1998-2017.nc")
nc = nc_open(ncfile)
TRMM.land = ncvar_get(nc,"precipitation")
nc_close(nc)

if (length(grep("per3h",threshold)) > 0) { by = 1/8; navg = 30*8 }
if (length(grep("perday",threshold)) > 0) { by = 1; navg = 30 }

datesERA5 = seq.Date(as.Date("1950-01-01"),as.Date("2018-12-31"),length.out=length(ERA5.global)) # by=by
datesTRMM = seq.Date(as.Date("1998-01-01"),as.Date("2017-12-31"),length.out=length(TRMM.global)) # by=by

ylim = range(c(ERA5.land,ERA5.ocean,ERA5.global, TRMM.global, TRMM.land, TRMM.ocean))

print("ERA5: land, ocean, global")
x = datesERA5
xtrend = seq(1950,2019,length.out=length(x))
y = ERA5.land
plot(x,ma(y,navg),col="darkgreen",type="l",ylim=ylim,xlab="Year",ylab="Area fraction", main=threshold)
plottrend(xtrend,y,col="darkgreen")
y = ERA5.ocean
matlines(x,ma(y,navg),col="blue")
plottrend(xtrend,y,col="blue")
y = ERA5.global
matlines(x,ma(y,navg))
plottrend(xtrend,y,col="black")

print("TRMM: land, ocean, global")
x = datesTRMM
xtrend = seq(1998,2018,length.out=length(x))
y = TRMM.land
matlines(x,ma(y,navg),col="green", lty=2)
plottrend(xtrend,y,col="green")

y = TRMM.ocean
matlines(x,ma(y,navg),col="cyan", lty=2)
plottrend(xtrend,y,col="cyan")

y = TRMM.global
matlines(x,ma(y,navg),col="gray", lty=2)
plottrend(xtrend,y,col="gray")

if (threshold=="10mmper3h") legend(x="topleft",legend=c("ERA5","ERA5 land","ERA5 ocean","TRMM","TRMM land"))

```

```

,"gray","green","cyan"),lty=c(1,1,1,2,2,2),cex=0.7,bty='n')

} # end of threshold loop

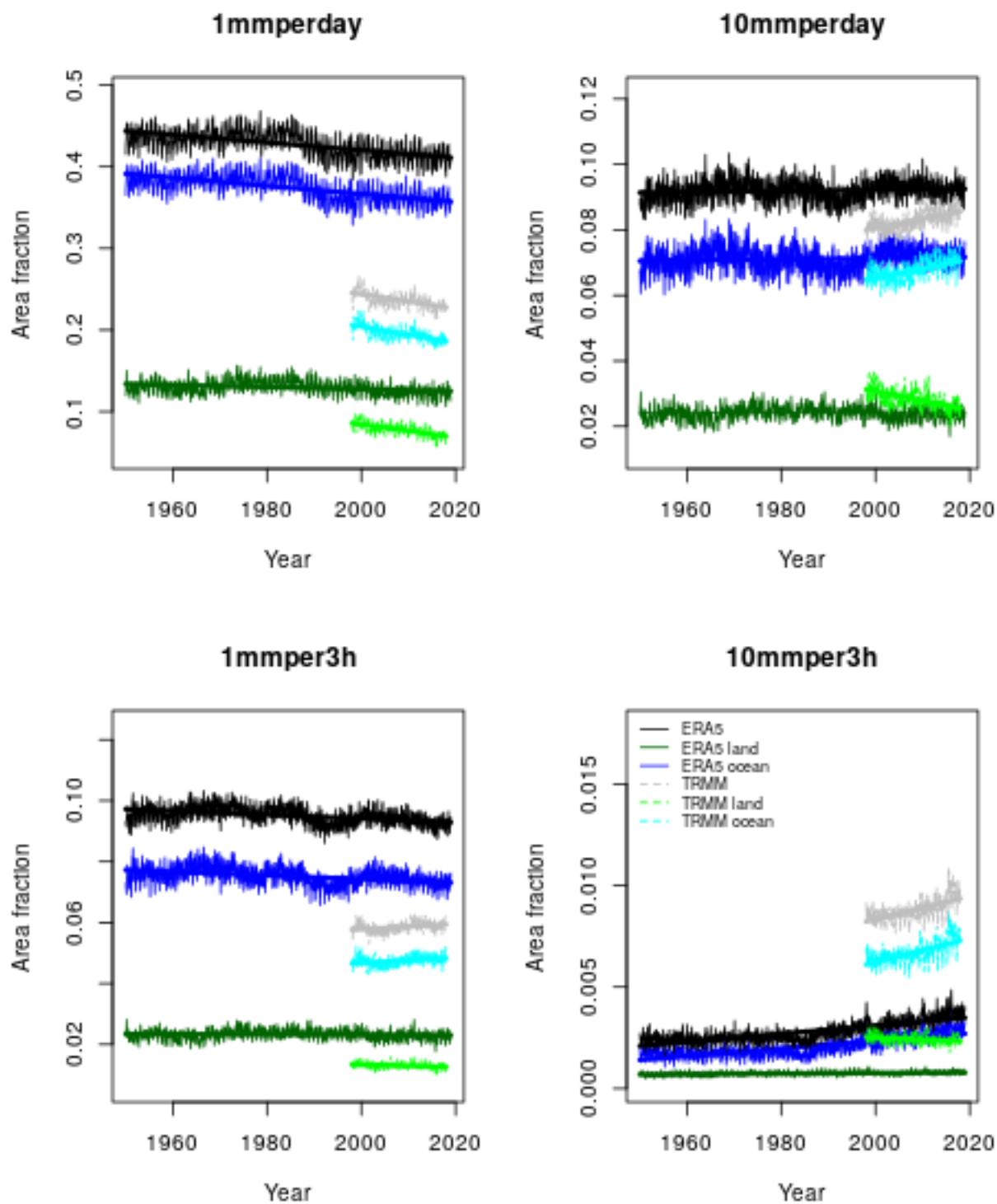
## [1] "1mmpiday"
## [1] "ERA5: land, ocean, global"

## [1] "Pr(>|t|)      : 1.11090039271121e-258"
## [1] "Slope [/yr]   : -0.000128"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : -0.000497"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : -0.000473"
## [1] "TRMM: land, ocean, global"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : -0.000802"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : -0.001"
## [1] "Pr(>|t|)      : 1.3043098342704e-293"
## [1] "Slope [/yr]   : -0.000886"
## [1] "10mmpiday"
## [1] "ERA5: land, ocean, global"

## [1] "Pr(>|t|)      : 0.751940441976242"
## [1] "Slope [/yr]   : -3.56e-07"
## [1] "Pr(>|t|)      : 5.94286409447086e-17"
## [1] "Slope [/yr]   : 1.73e-05"
## [1] "Pr(>|t|)      : 2.59217003118436e-15"
## [1] "Slope [/yr]   : 1.61e-05"
## [1] "TRMM: land, ocean, global"
## [1] "Pr(>|t|)      : 1.99413973495487e-230"
## [1] "Slope [/yr]   : -0.000284"
## [1] "Pr(>|t|)      : 9.51567970254732e-133"
## [1] "Slope [/yr]   : 0.000282"
## [1] "Pr(>|t|)      : 8.64110269285187e-141"
## [1] "Slope [/yr]   : 0.000285"
## [1] "1mmp3h"
## [1] "ERA5: land, ocean, global"

## [1] "Pr(>|t|)      : 3.65257344094251e-28"
## [1] "Slope [/yr]   : -5.64e-06"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : -6.02e-05"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : -6.26e-05"
## [1] "TRMM: land, ocean, global"
## [1] "Pr(>|t|)      : 7.5328183484575e-129"
## [1] "Slope [/yr]   : -4e-05"
## [1] "Pr(>|t|)      : 6.1700484164789e-206"
## [1] "Slope [/yr]   : 8.32e-05"
## [1] "Pr(>|t|)      : 9.0191410283371e-188"
## [1] "Slope [/yr]   : 8.71e-05"
## [1] "10mmp3h"
## [1] "ERA5: land, ocean, global"

```



```

## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : 1.42e-06"
## [1] "Pr(>|t|)      : 0"
## [1] "Slope [/yr]   : 1.87e-05"
## [1] "Pr(>|t|)      : 0"

```

```

## [1] "Slope [ /yr] : 2.01e-05"
## [1] "TRMM: land, ocean, global"
## [1] "Pr(>|t|) : 2.52039522911311e-117"
## [1] "Slope [ /yr] : -9.92e-06"
## [1] "Pr(>|t|) : 0"
## [1] "Slope [ /yr] : 5.93e-05"
## [1] "Pr(>|t|) : 0"
## [1] "Slope [ /yr] : 5.68e-05"

The mismatch in the levels between the TRMM and ERA5 may perhaps be due to the fact that the TRMM measures instants of events whereas the ERA5 reanalysis calculates the accumulated rainfall over 3hrs and 24 hrs.

years <- seq(1800,2020,20)
plot_years <- seq(as.Date("1800-01-01"),as.Date("2020-01-01"),by="20 years")

ncid <- nc_open '~/Downloads/global-precip-area.nc')
A <- ncvar_get(ncid,'tp')
t <- as.Date(ncvar_get(ncid,'time'))/24,origin='1900-01-01'
nc_close(ncid)
A <- zoo(x=A,order.by=t)
A <- year2date(annual(A))

ncid <- nc_open '~/Downloads/yearmean_fldmean_gec1mm_era20c_tp_daily.nc')
A20C <- ncvar_get(ncid,'tp')
t20C <- as.Date(ncvar_get(ncid,'time'),origin='1900-01-01')
nc_close(ncid)
A20C <- zoo(x=A20C,order.by=t20C)

ncid <- nc_open '~/Downloads/yearmean_fldmean_gec1mm_NOAA_20CRv3_tp_daily.nc')
A20CR <- ncvar_get(ncid,'prate')
t20CR <- as.Date(ncvar_get(ncid,'time'))/24,origin='1800-01-01'
nc_close(ncid)
A20CR <- zoo(x=A20CR,order.by=t20CR)

ncid <- nc_open '~/Downloads/yearmean_fldmean_gec1mm_NCEP_RA1_tp_daily.nc')
ANCEP <- ncvar_get(ncid,'prate')
tNCEP<- as.Date(ncvar_get(ncid,'time'))/24,origin='1800-01-01'
nc_close(ncid)
ANCEP <- zoo(x=ANCEP,order.by=tNCEP)

plot(A, ylab='Area (fraction)',xlab='',ylim=c(0.4,0.75),xlim=range(t,t20C,t20CR),col="black",xaxt="n",l
axis(1,at=plot_years,label=years)

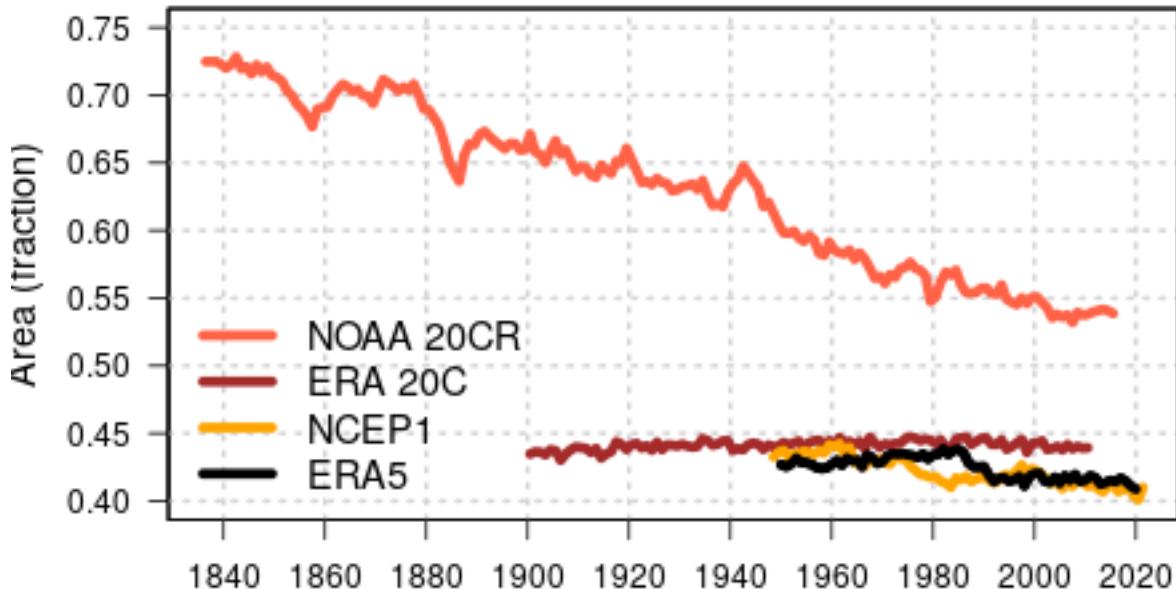
abline(h=seq(0.25,0.8,0.05),col="lightgray",lty="dotted",lwd=2)
abline(v=plot_years,col="lightgrey",lty="dotted",lwd=2)

lines(A20C,col='brown',lty=1,lwd=5)
lines(A20CR,col='tomato',lty=1,lwd=5)
lines(ANCEP,col='orange',lty=1,lwd=5)
lines(A,col='black',lty=1,lwd=5)

legend("bottomleft",col=c("tomato","brown","orange","black"),lwd=5,legend=c("NOAA 20CR","ERA 20C","NCEP"))

box()

```



Print out some trend estimates:

```
print("ERA5: 1961-2020")

## [1] "ERA5: 1961-2020"
calX <- data.frame(A = coredata(A), t=year(A))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))

##
## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##      Min       1Q       Median       3Q      Max
## -0.0092423 -0.0044784 -0.0009777  0.0035267  0.0138770
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.9916888  0.0668938 14.825 < 2e-16 ***
## t          -0.0002859  0.0000337 -8.484 2.61e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005819 on 69 degrees of freedom
## Multiple R-squared:  0.5105, Adjusted R-squared:  0.5034
## F-statistic: 71.97 on 1 and 69 DF,  p-value: 2.61e-12
```

```

print(predict(trendfitX,newdata=data.frame(t=c(1961,2020))))
```

```

##           1          2
## 0.4310851 0.4142183
print("NCEP1: 1961-2020")
```

```

## [1] "NCEP1: 1961-2020"
calX <- data.frame(A = coredata(ANCEP), t=year(ANCEP))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))
```

```

##
## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##       Min     1Q   Median     3Q    Max
## -0.0123409 -0.0031870  0.0000616  0.0032295  0.0104331
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.275e+00 5.125e-02 24.88 <2e-16 ***
## t          -4.300e-04 2.582e-05 -16.65 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004745 on 72 degrees of freedom
## Multiple R-squared:  0.7939, Adjusted R-squared:  0.791
## F-statistic: 277.3 on 1 and 72 DF,  p-value: < 2.2e-16
print(predict(trendfitX,newdata=data.frame(t=c(1961,2020))))
```

```

##           1          2
## 0.4319379 0.4065662
print("NOAA 20CR: 1961-2015")
```

```

## [1] "NOAA 20CR: 1961-2015"
calX <- data.frame(A = coredata(A20CR), t=year(A20CR))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))
```

```

##
## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##       Min     1Q   Median     3Q    Max
## -0.038337 -0.005258 -0.000419  0.005578  0.035541
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.787e+00 3.066e-02 90.89 <2e-16 ***
## t          -1.120e-03 1.592e-05 -70.35 <2e-16 ***
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0111 on 178 degrees of freedom
## Multiple R-squared:  0.9653, Adjusted R-squared:  0.9651
## F-statistic:  4949 on 1 and 178 DF,  p-value: < 2.2e-16
print(predict(trendfitX,newdata=data.frame(t=c(1961,2015))))
```

```

##           1          2
## 0.5907030 0.5302358
```

```

print("ERA 20C: 1961-2015")
```

```

## [1] "ERA 20C: 1961-2015"
```

```

calX <- data.frame(A = coredata(A20C), t=year(A20C))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))
```

```

##
```

```

## Call:
## lm(formula = A ~ t, data = calX)
##
```

```

## Residuals:
##       Min     1Q   Median     3Q    Max
## -0.0087288 -0.0022577  0.0002984  0.0023511  0.0058726
##
```

```

##
```

```

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.500e-01 1.810e-02 19.342 < 2e-16 ***
## t           4.662e-05 9.256e-06  5.037 1.88e-06 ***
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Residual standard error: 0.003124 on 109 degrees of freedom
## Multiple R-squared:  0.1888, Adjusted R-squared:  0.1814
## F-statistic: 25.37 on 1 and 109 DF,  p-value: 1.884e-06
print(predict(trendfitX,newdata=data.frame(t=c(1961,2015))))
```

```

##           1          2
## 0.4414629 0.4439806
```

```

#####
##50S to 50N
#####
ncid <- nc_open('~/Downloads/era5_50s50n-precip-area_new.nc')
A <- ncvar_get(ncid,'tp')
t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
A <- zoo(x=A,order.by=t)
A <- year2date(annual(A))

ncid <- nc_open('~/Downloads/fldmean_50S_to_50N/yearmean_fldmean_gec1mm_era20c_tp_daily.nc')
```

```

A20C <- ncvar_get(ncid,'tp')
t20C <- as.Date(ncvar_get(ncid,'time'),origin='1900-01-01')
nc_close(ncid)
A20C <- zoo(x=A20C,order.by=t20C)

ncid <- nc_open('~/Downloads/fldmean_50S_to_50N/yearmean_fldmean_gcm1mm_NOAA_20CRv3_tp_daily.nc')
A20CR <- ncvar_get(ncid,'prate')
t20CR <- as.Date(ncvar_get(ncid,'time')/24,origin='1800-01-01')
nc_close(ncid)
A20CR <- zoo(x=A20CR,order.by=t20CR)

ncid <- nc_open('~/Downloads/fldmean_50S_to_50N/yearmean_fldmean_gcm1mm_NCEP_RA1_tp_daily.nc')
ANCEP <- ncvar_get(ncid,'prate')
tNCEP<- as.Date(ncvar_get(ncid,'time')/24,origin='1800-01-01')
nc_close(ncid)
ANCEP <- zoo(x=ANCEP,order.by=tNCEP)

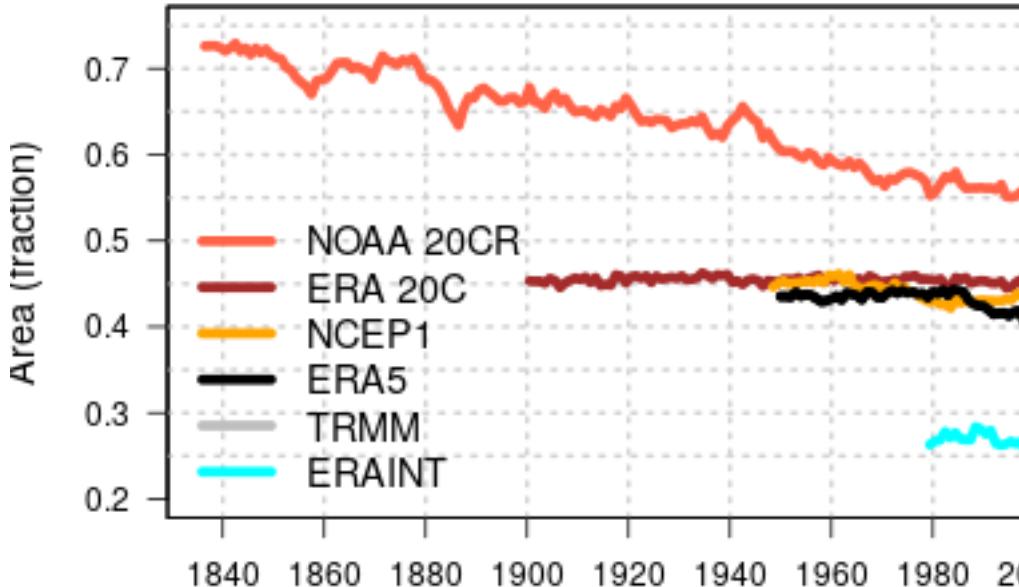
plot(A, ylab='Area (fraction)',xlab='',ylim=c(0.2,0.75),xlim=range(t,t20C,t20CR),col="black",xaxt="n",lty=1,at=plot_years,label=years)

abline(h=seq(0.25,0.8,0.05),col="lightgray",lty="dotted",lwd=2)
abline(v=plot_years,col="lightgrey",lty="dotted",lwd=2)

lines(A20C,col='brown',lty=1,lwd=5)
lines(A20CR,col='tomato',lty=1,lwd=5)
lines(ANCEP,col='orange',lty=1,lwd=5)
lines(A,col='black',lty=1,lwd=5)
lines(A.TRMM,col='grey',lty=1,lwd=5)
lines(A.ERAIANT,col='cyan',lty=1,lwd=5)

legend("bottomleft",col=c("tomato","brown","orange","black","grey","cyan"),lwd=5,legend=c("NOAA 20CR","
box()

```



#### 50S-50N - semi-global analysis.

Fig. S9. A comparison between the precipitation area estimated from different reanalyses. The results for TRMM and ERAINT shown here were taken from Benestad (2018) and were estimated through PMEL Ferret rather than CDO.

Estimates of  $A_p$  greater than 0.5 are unrealistic as such high numbers imply precipitation over more than 50% of the global surface area every day. These results suggest that the NOAA 20CR reanalysis does not provide a reliable account of the global precipitation statistics. These results may be sensitive to subjective the spatial resolution of the models and the threshold set to distinguish wet and dry days.

The estimated global 24-hr precipitation surface area in the different reanalyses also is expected to depend on the typical spatial extent of the rainfall systems. Those with very limited spatial extent are likely to be misrepresented in reanalyses with coarser spatial resolution than those that cover a wider area. We can try to shed some more light on typical spatial scales connected to rainfall patters with a 2D wavelet analysis applied to high-resolution data - an analysis presented later on .

```

print("ERA5: 1961-2020")

## [1] "ERA5: 1961-2020"

calX <- data.frame(A = coredata(A), t=year(A))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))

##
## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.00000 -0.00000 -0.00000 -0.00000  0.00000
## 
```

```

## -0.0134383 -0.0042792 -0.0002548  0.0045149  0.0161670
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.3918760  0.0780224   17.84   <2e-16 ***
## t          -0.0004862  0.0000393  -12.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006787 on 69 degrees of freedom
## Multiple R-squared:  0.6892, Adjusted R-squared:  0.6847
## F-statistic: 153 on 1 and 69 DF, p-value: < 2.2e-16
print(predict(trendfitX,newdata=data.frame(t=c(1961,2020))))
```

```

##           1          2
## 0.4384602 0.4097751
print("NCEP1: 1961-2020")
```

```

## [1] "NCEP1: 1961-2020"
```

```

calX <- data.frame(A = coredata(ANCEP), t=year(ANCEP))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))
```

```

##
## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##       Min     1Q     Median      3Q     Max
## -0.014752 -0.003010 -0.000248  0.002927  0.013213
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.511e+00  6.055e-02   24.95   <2e-16 ***
## t          -5.418e-04  3.051e-05  -17.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005606 on 72 degrees of freedom
## Multiple R-squared:  0.8141, Adjusted R-squared:  0.8116
## F-statistic: 315.4 on 1 and 72 DF, p-value: < 2.2e-16
print(predict(trendfitX,newdata=data.frame(t=c(1961,2020))))
```

```

##           1          2
## 0.4483717 0.4164058
print("NOAA 20CR: 1961-2015")
```

```

## [1] "NOAA 20CR: 1961-2015"
```

```

calX <- data.frame(A = coredata(A20CR), t=year(A20CR))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))
```

```

## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -0.042577 -0.005715 -0.000363  0.005809  0.039899
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.711e+00 3.361e-02   80.65 <2e-16 ***
## t          -1.079e-03 1.745e-05  -61.81 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01216 on 178 degrees of freedom
## Multiple R-squared:  0.9555, Adjusted R-squared:  0.9552
## F-statistic: 3821 on 1 and 178 DF, p-value: < 2.2e-16
print(predict(trendfitX,newdata=data.frame(t=c(1961,2015))))
```

```

##           1         2
## 0.5955332 0.5372930
print("ERA 20C: 1961-2015")
```

```

## [1] "ERA 20C: 1961-2015"
calX <- data.frame(A = coredata(A20C), t=year(A20C))
trendfitX <- lm(A ~ t, data=calX)
print(summary(trendfitX))
```

```

## Call:
## lm(formula = A ~ t, data = calX)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -0.0116583 -0.0029218  0.0006505  0.0031483  0.0077312
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.730e-01 2.406e-02  23.817 < 2e-16 ***
## t          -6.113e-05 1.230e-05  -4.968 2.52e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004154 on 109 degrees of freedom
## Multiple R-squared:  0.1846, Adjusted R-squared:  0.1772
## F-statistic: 24.68 on 1 and 109 DF, p-value: 2.522e-06
print(predict(trendfitX,newdata=data.frame(t=c(1961,2015))))
```

```

##           1         2
## 0.4531022 0.4498012
```

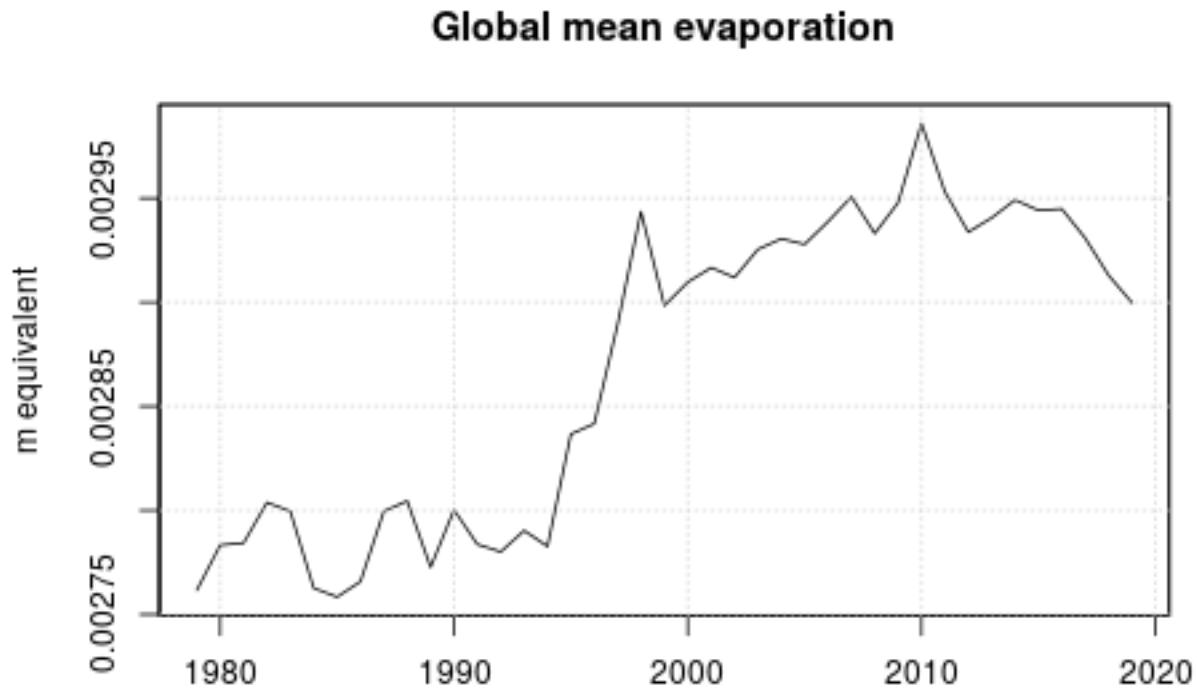
## Other indicators

**Global mean evaporation** Evaporation is an interesting aspect of the global hydrological cycle which represent a closed look within the planetary system. It represent an exchange of  $H_2O$  between the surface (oceans, land, cryosphere and vegetation) and the atmosphere and can be related to the global precipitation amount  $E = -P$  when the atmosphere and the surface are in equilibrium over time.

```
print('*** Evaporation: ***')

## [1] "*** Evaporation: ***"

ncid <- nc_open('~Downloads/global-evap-mean.nc')
E <- ncvar_get(ncid, 'e_0001')
t <- as.Date(ncvar_get(ncid, 'time')/24, origin='1900-01-01')
nc_close(ncid)
evap <- zoo(abs(E), order.by=t)
plot(annual(evap), main='Global mean evaporation', ylab='m equivalent', xlab='')
grid()
```



According to ERA5, the global evaporation shifted in the 1990s. It's not clear whether this shift was a real change or if it was a consequence of technical issues (e.g. introduction of new observing instruments). This shift also is likely to affect the precipitation and the rainfall patterns.

**Global mean outgoing longwave radiation - planetary heat loss** The global outgoing radiation (OLR) provides a measure of the planetary heat loss to space and is connected to the planetary energy balance. It may also be influenced by a changing greenhouse effect or shifts in the global hydrological cycle.

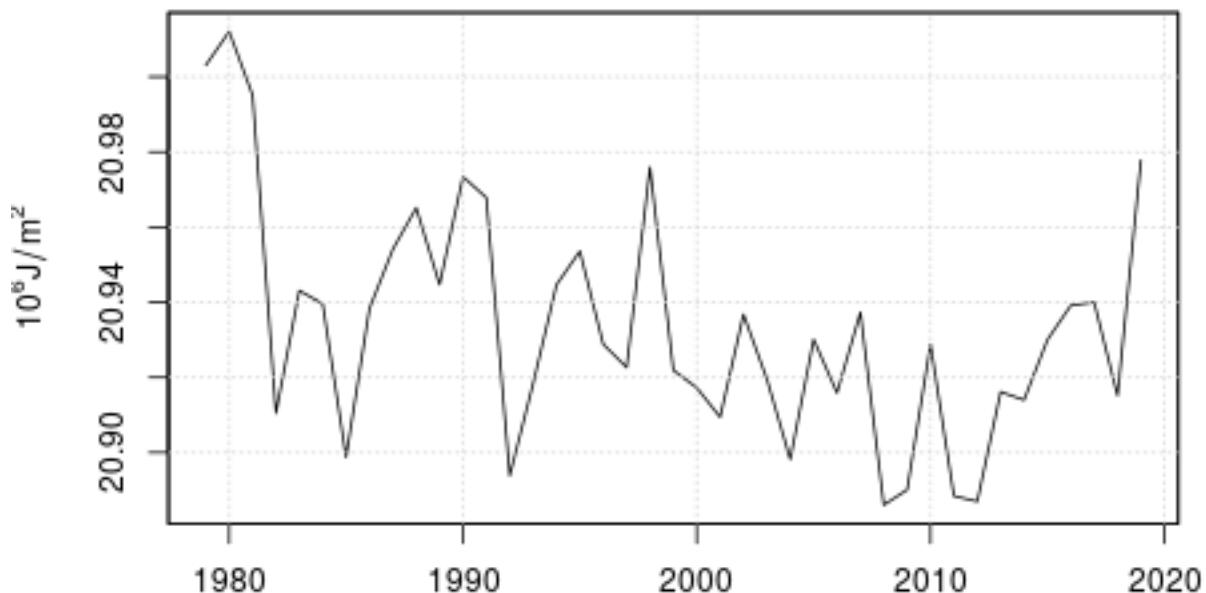
```

print('*** toa_outgoing_longwave_flux: ***')

## [1] "*** toa_outgoing_longwave_flux: ***"
ncid <- nc_open '~/Downloads/global-olr-mean.nc')
ttr <- ncvar_get(ncid,'ttr_0001') # J m**-2
t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
olr <- zoo(x=abs(ttr)*1.0e-6,order.by=t)
plot(annual(olr),main='Top of the Atmosphere outgoing longwave flux',ylab=expression(10^6*J/m^2),xlab=
grid()

```

## Top of the Atmosphere outgoing longwave flux



Apart from minor year-to-year variations (about 0.5% compared to its mean level), there was no clear trend in the OLR.

```

print('*** surface fluxes: ***')

Surface fluxes

## [1] "*** surface fluxes: ***"
ncid <- nc_open '~/Downloads/global-sw+lw-mean.nc')
slhf <- ncvar_get(ncid,'slhf') # surface_upward_latent_heat_flux
ssr <- ncvar_get(ncid,'ssr') # surface_net_downward_shortwave_flux
str <- ncvar_get(ncid,'str') # surface_net_upward_longwave_flux
strd <- ncvar_get(ncid,'strd') # Surface thermal radiation downwards

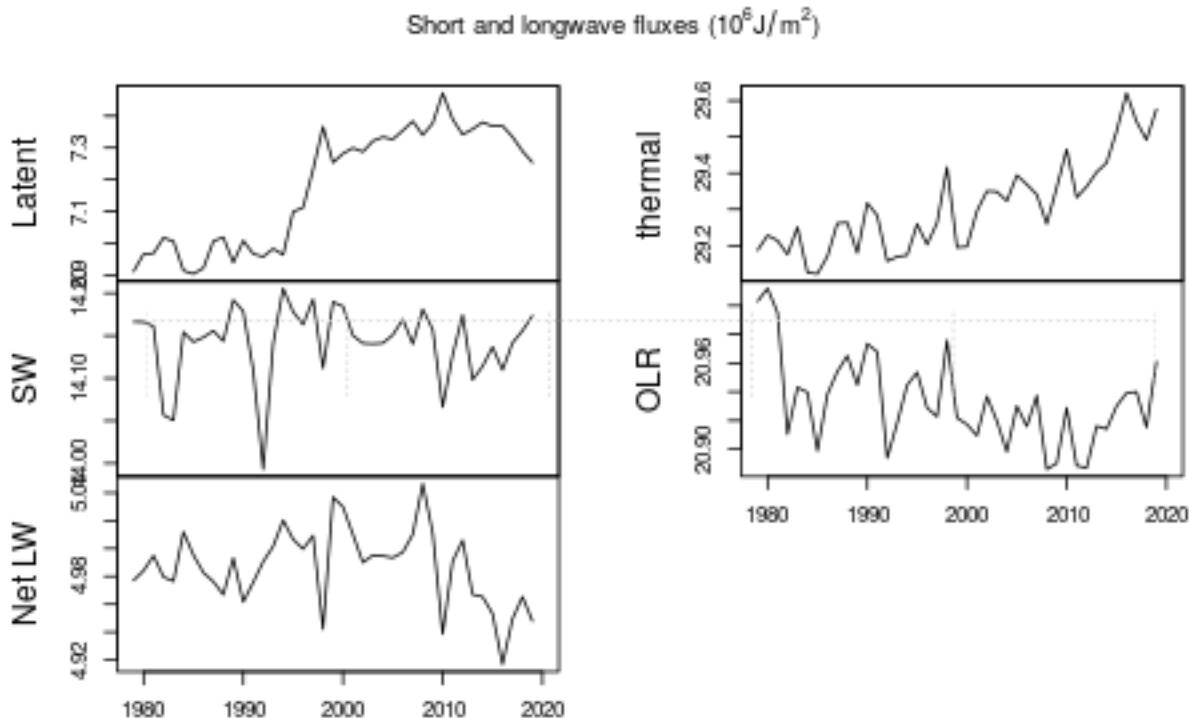
```

```

ttr <- ncvar_get(ncid,'ttr')    # toa_outgoing_longwave_flux
t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
X <- cbind(slhf[1],ssr[1],str[1],strd[1],ttr[1])
sw.lw <- zoo(x=abs(X)*1.0e-6,order.by=t)
colnames(sw.lw) <- c('Latent','SW','Net LW','thermal','OLR')
plot(annual(sw.lw),main=expression(paste('Short and longwave fluxes (' ,10^6*J/m^2, ')')),  

     xlab='',ylab=colnames(sw.lw))
grid()

```



The most pronounced traits in the surface fluxes of the ERA5 are a shift in the latent flux (matching the evaporation) and an upward trend in the thermal flux. There were no clear trend in the short wave (SW) nor the net longwave (LW) fluxes, both which exhibited stronger seemingly random year-to-year variations.

**Global energy partition** The global hydrological cycle is connected to various forms of energy, and in particular latent heat fluxes. Since the ERA5 reanalysis also provides ready estimates of various energy, we were curious to inspect the energy partition between various forms:

```

print('*** energy partitions: ***')

## [1] "*** energy partitions: ***"

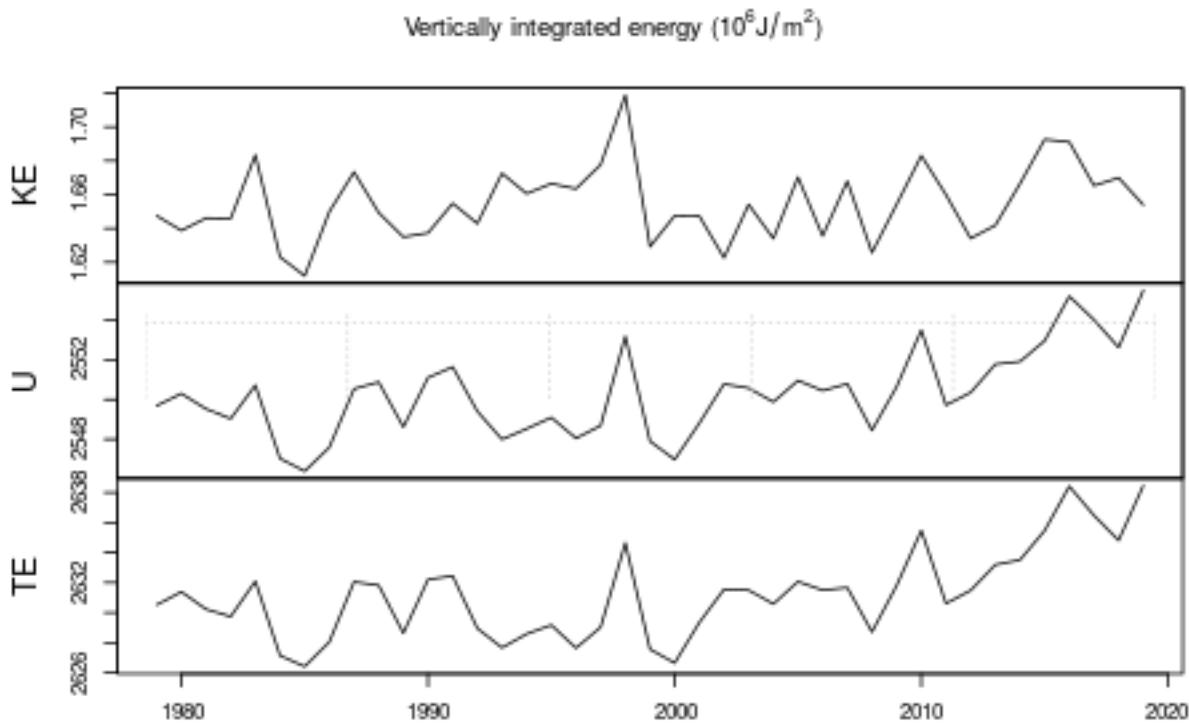
ncid <- nc_open '~/Downloads/global-energy-mean.nc'
ke <- ncvar_get(ncid,'p59.162_0001') # Vertical integral of kinetic energy J m**-2
U <- ncvar_get(ncid,'p60.162_0001') # Vertical integral of thermal energy
TE <- ncvar_get(ncid,'p63.162_0001') # Vertical integral of total energy

```

```

t <- as.Date(ncvar_get(ncid,'time')/24,origin='1900-01-01')
nc_close(ncid)
X <- cbind(ke,U,TE)
energy <- zoo(x=X*1.0e-6,order.by=t)
colnames(energy) <- c('KE','U','TE')
plot(annual(energy),main=expression(paste('Vertically integrated energy (',10^6*J/m^2,')')),
      xlab='',ylab=colnames(energy))
grid()

```



```

print(round(colMeans(energy,na.rm=TRUE),2))

##      KE        U        TE
##    1.65 2550.28 2631.19

print(trend.coef(energy))

##      KE          U          TE
## 0.0003724985 0.0896964936 0.1248318231

print(round(100*c(trend.pval(energy[,1]),trend.pval(energy[,2]),trend.pval(energy[,3])),2))

## trend.pvalue trend.pvalue trend.pvalue
##       15.38         0.01         0.01

```

Most of the energy in the atmosphere is from internal thermal energy ( $2550 \times 10^6 * J/m^2$ ) and only a small portion is from kinetic energy ( $1.7 \times 10^6 * J/m^2$ ). There has been a statistically significant increase in both Earth's total thermal and total energy.

## Wavelets and the global mean temperature

The R-code provided in the two three chunks were used in the multi-resolution wavelet analysis performed herein by Cristian Lussana and Barbara Casati. The code is also maintained at <https://github.com/metno/twave>.

```
#-----
# Author: Barbara Casati
# Description: R function which performs
# a 2D Haar discrete wavelet transform
# of a dyadic field (obs) and evaluates
# squared energy for each scale component,
# squared energy total and percentage for each scale.
#-----

wavEn2 <- function (obs)
{
  #
  # check input field has dyadic square domain
  #
  if (dim(obs)[1] != dim(obs)[2]) {
    stop("Input matrices must be squared")
  }
  if (log2(dim(obs)[1]) - floor(log2(dim(obs)[1])) != 0) {
    stop("Input matrices must have dimensions equal to a power of 2")
  }
  if ((dim(obs)[1] == dim(obs)[2]) & (log2(dim(obs)[1]) - floor(log2(dim(obs)[1])) == 0)) {
    #
    # begin computation
    #
    N <- log2(dim(obs)[1])
    library(waveslim)
    Eo.dwt <- dwt.2d(obs, wf = "haar", J = log2(dim(obs)[1]))
    En2o <- numeric()
    for (i in 1:N) {
      En2o[i] <- mean((Eo.dwt[[1 + 3 * (i - 1)]]/2^i)^2) +
        mean((Eo.dwt[[2 + 3 * (i - 1)]]/2^i)^2) +
        mean((Eo.dwt[[3 + 3 * (i - 1)]]/2^i)^2)
    }
    En2o <- c(En2o, mean(obs)^2)
    En2o.tot <- sum(En2o)
    En2o.perc <- En2o/En2o.tot
    #
    # check: sum(En2o) == mean(obs^2)
    #
    z <- list(En2o = En2o, En2o.tot = En2o.tot, En2o.perc = En2o.perc)
    return(z)
  }
}

#!/usr/bin/env Rscript
## From Cristian Lussana 2021-12-10
#
#-----
# ~~~ tpwave.r ~~~
```

```

# Multiresolution analysis of prec based on discrete wavelet transformation
#
# Author: CL (cristianl@met.no)
#-----

library(ncdf4)
library(waveslim)
library(raster)
library(argparser)

#
#-----
# Read command line arguments

p <- arg_parser("wave_it")

p <- add_argument(p, "--date1", help="begin (%Y-%m-%d)", type="character", default="1950-01-01")

p <- add_argument(p, "--date2", help="end (%Y-%m-%d)", type="character", default="1950-01-02")

p <- add_argument(p, "--ffin", help="input file", type="character", default=".//data/ERA5_tp_day_1950-01-01.nc")

p <- add_argument(p, "--ffout", help="output file", type="character", default="ERA5_tp_day_example_wav.nc")

p <- add_argument(p, "--extent", help="extent", type="numeric", nargs=Inf, default=NA)

p <- add_argument(p, "--extent_lab", help="label", type="character", default=NA)

argv <- parse_args(p)

#
#-----
# Define extent based on the command line argument
if ( !is.na( argv$extent_lab) ) {
  # Europe
  if ( argv$extent_lab == "EU" ) argv$extent <- c( -40, +75, 10, 85)
  # Africa
  if ( argv$extent_lab == "AF" ) argv$extent <- c( -40, +75, -55, 45)
  # Asia
  if ( argv$extent_lab == "AS" ) argv$extent <- c( +25, +179, -5, 85)
  # Oceania
  if ( argv$extent_lab == "OC" ) argv$extent <- c( +40, +179, -70, 10)
  # North America
  if ( argv$extent_lab == "NA" ) argv$extent <- c(-179, -10, 10, 85)
  # South America
  if ( argv$extent_lab == "SA" ) argv$extent <- c(-179, -10, -70, 20)
  # nothern hemisphere
  if ( argv$extent_lab == "NH" ) argv$extent <- c(-179, +179, 0, 85)
  # southern hemisphere
  if ( argv$extent_lab == "SH" ) argv$extent <- c(-179, +179, -85, 0)
  # tropics
  if ( argv$extent_lab == "TR" ) argv$extent <- c(-179, +179, -25, 25)
  # temperate north
}

```

```

if ( argv$extent_lab == "TN" ) argv$extent <- c(-179, +179, 23, 67)
# temperate south
if ( argv$extent_lab == "TS" ) argv$extent <- c(-179, +179, -67, -23)
}

#
#-----
# Constants

date_format <- "%Y-%m-%d"
proj4.wgs84 <- "+proj=latlong +datum=WGS84"

#
#-----
# time

if ( !is.na(argv$date1) & !is.na(argv$date2)) {
  Rdate1 <- strptime( paste0(argv$date1, "T00"), format=paste0( date_format, "T%H"), tz="UTC")
  Rdate2 <- strptime( paste0(argv$date2, "T23"), format=paste0( date_format, "T%H"), tz="UTC")
}

#
#-----
# open input file

if ( !file.exists( argv$ffin)) q()
nc <- nc_open( argv$ffin)
tp <- nc$var[[2]]
ndims <- tp$ndims
start <- rep(1,ndims)
varsizes <- tp$varsizes

#
#-----
# define timesteps to read from file

tseq <- as.POSIXct( (tp$dim[[3]]$vals * 60 * 60), origin="1900-01-01", tz="UTC")

if ( !is.na(argv$date1) & !is.na(argv$date2)) {
  ixt <- which( as.numeric(tseq) >= as.numeric(Rdate1) &
                as.numeric(tseq) <= as.numeric(Rdate2))
} else {
  ixt <- 1:length(tseq)
}

#
#-----
# define spatial grids

lons_tot <- tp$dim[[1]]$vals
lats_tot <- tp$dim[[2]]$vals
w <- raster( nrows=length(lats_tot), ncols=2*length(lons_tot),
             xmjn=min(lons_tot-360)-0.125, xmjn=max(lons_tot)+0.125,

```

```

        ymn=min(lats_tot)-0.125, ymx=max(lats_tot)+0.125,
        crs=proj4.wgs84)
r <- crop(w, extent(c(-180.1, 179.8, -90.125, 90.125)))

if ( !any( is.na( argv$extent))) {
  q <- crop( r, argv$extent)
} else {
  q <- r
}

xy<-xyFromCell(q,1:nrow(q))
lons<-sort(unique(xy[,1]))
lats<-sort(unique(xy[,2]),decreasing=T)
rm(xy)

# number of rows (cols) of the dyadic grid
dimdy <- 2**ceiling( log2( max( dim(q)[1:2])))

# dyadic grid
s <- raster( nrows=dimdy, ncols=dimdy,
             xmn=min(lons)-0.125, xmx=max(lons)+0.125,
             ymn=min(lats)-0.125, ymx=max(lats)+0.125,
             crs=proj4.wgs84)

nnscales <- log2(dimdy)+1
N <- nnscales-1
listscales <- 2**seq(1,nnscales)-1)*1

if ( length(ixt) == 0) q()

#
#-----
# elaboration

nt      <- length(ixt)
En2o_t<-array(data=NA,dim=c(nnscales,nt))

for( i in 1:nt ) {
  if ( ( i %% 10) == 0) cat( paste(i,"/",nt,"\n"))
  start <- rep(1,ndims)
  start[ndims] <- ixt[i]
  count <- varsize
  count[ndims] <- 1
  data <- ncvar_get( nc, tp, start=start, count=count )
  w[] <- cbind( t(data), t(data)) * 1000
  r <- crop(w, extent(c(-180.1, 179.8, -90.125, 90.125)))
  if ( !any( is.na( argv$extent))) {
    q <- crop( r, argv$extent)
  } else {
    q <- r
  }
  u   <- resample(q,s,method="bilinear")
  obs <- as.matrix(u)
}

```

```

Eo.dwt<-dwt.2d(obs, wf = "haar", J = N)
En2o<-vector(mode="numeric",length=N)
for (j in 1:N) {
  En2o[j] <- mean((Eo.dwt[[1 + 3 * (j - 1)]]/2^j)^2) +
    mean((Eo.dwt[[2 + 3 * (j - 1)]]/2^j)^2) +
    mean((Eo.dwt[[3 + 3 * (j - 1)]]/2^j)^2)
}
En2o_t[1:N,i] <- En2o[1:N]
En2o_t[(N+1),i]<-cellStats(u,stat="mean",na.rm=T)**2
}
nc_close(nc)

#
#-----
# write output file (Rdata-format)
# r= raster with the specification of the original ERA5 grid
# q= raster with the specification of the original ERA5 grid, but cropped
# s= raster with the specification of the high-res dyadic domain
# lons/lats = longitudes and latitudes of the "q" grid
# dimdy = number of rows/columns of the dyadic domain "s"
# N = number of spatial scales used in the decompositions (mother wavelets)
# listscales = spatial scale size (in number of gridpoints)
# nnscales = N+1, total number of spatial scales (mother + father)
# En2o_t = matrix with the results of the decomposition. The dimensions are: nnscales x number_of_time_
# tseq = sequence of date/time steps
# ixt = indexes of the time steps actually used
#
save( file=argv$ffout, r, q, s, lons, lats,
      dimdy, N, listscales, nnscales,
      En2o_t, tseq, ixt)

#
#-----

```

## Wavelets and the global mean temperature

The following chunk processes the wavelet power and compares them with the global mean surface temperature (gst) and stores in a temporary R binary data file ERA5\_tp.wavelet.power.rda. More R-code and data are also available from <https://github.com/metno/tpwave>.

```

if (!file.exists('ERA5_tp.wavelet.power.rda')) {
  path <- '~/.data/tpwave'
  wlfiles <- list.files(path=path,pattern='ERA5_tp_day_waveEn_ALL_',full.names = TRUE)
  ## Get the annual mean power of the various wavelet components
  for (i in 1:length(wlfiles)) {
    print(wlfiles[i])
    load(wlfiles[i])
    ## Take the annual mean power
    if (i==1) power <- rowMeans(En2o_t) else
      power <- cbind(power,rowMeans(En2o_t))
  }
  power <- zoo(t(power),order.by=1950:2020)
}
```

```

## Get the global mean temperature from ERA5:
gst <- aggregate.area(retrieve('~/Downloads/ERA5_t2m_year.nc'),FUN='mean')
index(gst) <- year(gst)
gst <- subset(gst,year=c(1950,2020))

## Synchronise the two series
power <- matchdate(power,gst)
gst <- matchdate(gst,power)
save(power,gst,file='ERA5.tp.wavelet.power.rda')
} else load('ERA5.tp.wavelet.power.rda')

for (i in 1:(dim(power)[2])) {
  cal <- data.frame(y=coredata(power[,i]),x=coredata(gst))
  fit <- lm(y ~ x, data = cal)
  print(summary(fit))
  if (i==1) power.fit <- c(predict(fit)) else
    power.fit <- cbind(power.fit,c(predict(fit)))
}

##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##       Min        1Q     Median        3Q        Max
## -0.033039 -0.008223 -0.000136  0.012183  0.028602
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.316385   0.068215 -19.30   <2e-16 ***
## x            0.110971   0.004803  23.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01348 on 69 degrees of freedom
## Multiple R-squared:  0.8856, Adjusted R-squared:  0.8839 
## F-statistic: 533.9 on 1 and 69 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##       Min        1Q     Median        3Q        Max
## -0.104979 -0.026043  0.003263  0.038258  0.075358
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -4.28875   0.22711 -18.88   <2e-16 ***
## x            0.35612   0.01599  22.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## Residual standard error: 0.04489 on 69 degrees of freedom
## Multiple R-squared:  0.8779, Adjusted R-squared:  0.8761
## F-statistic:   496 on 1 and 69 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -0.281152 -0.066146  0.006351  0.102363  0.186240
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.03102   0.57985 -19.02   <2e-16 ***
## x            0.90921   0.04082  22.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1146 on 69 degrees of freedom
## Multiple R-squared:  0.8779, Adjusted R-squared:  0.8761
## F-statistic:   496 on 1 and 69 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -0.52438 -0.11257  0.01127  0.15897  0.36891
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.74198   1.05506 -17.76   <2e-16 ***
## x            1.56196   0.07428  21.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2086 on 69 degrees of freedom
## Multiple R-squared:  0.865, Adjusted R-squared:  0.8631
## F-statistic: 442.2 on 1 and 69 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##       Min     1Q Median     3Q    Max
## -0.62157 -0.14572  0.02697  0.18541  0.50472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.25364   1.37271 -15.48   <2e-16 ***
## x            1.85957   0.09665  19.24   <2e-16 ***

```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2713 on 69 degrees of freedom
## Multiple R-squared: 0.8429, Adjusted R-squared: 0.8406
## F-statistic: 370.2 on 1 and 69 DF, p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -0.6282 -0.1825  0.0721  0.1862  0.4762
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.66455   1.34629 -13.86  <2e-16 ***
## x            1.75144   0.09479  18.48  <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2661 on 69 degrees of freedom
## Multiple R-squared: 0.8319, Adjusted R-squared: 0.8294
## F-statistic: 341.4 on 1 and 69 DF, p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -0.42801 -0.11023  0.01989  0.12700  0.42790
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.13625   0.97570 -12.44  <2e-16 ***
## x            1.26973   0.06869  18.48  <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1929 on 69 degrees of freedom
## Multiple R-squared: 0.832, Adjusted R-squared: 0.8295
## F-statistic: 341.6 on 1 and 69 DF, p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -0.288796 -0.080895  0.000714  0.094118  0.198796
##
## Coefficients:

```

```

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.83856    0.60308 -9.681 1.74e-14 ***
## x           0.70404    0.04246 16.581 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1192 on 69 degrees of freedom
## Multiple R-squared:  0.7994, Adjusted R-squared:  0.7965
## F-statistic: 274.9 on 1 and 69 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19666 -0.05456  0.01175  0.06537  0.21821
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.39060    0.43615 -5.481 6.47e-07 ***
## x           0.33146    0.03071 10.794 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08621 on 69 degrees of freedom
## Multiple R-squared:  0.6281, Adjusted R-squared:  0.6227
## F-statistic: 116.5 on 1 and 69 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.146904 -0.045620  0.006367  0.044045  0.110142
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.56708    0.31341   1.809   0.0748 .
## x           0.05027    0.02207   2.278   0.0258 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06195 on 69 degrees of freedom
## Multiple R-squared:  0.06995, Adjusted R-squared:  0.05648
## F-statistic:  5.19 on 1 and 69 DF,  p-value: 0.02582
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max

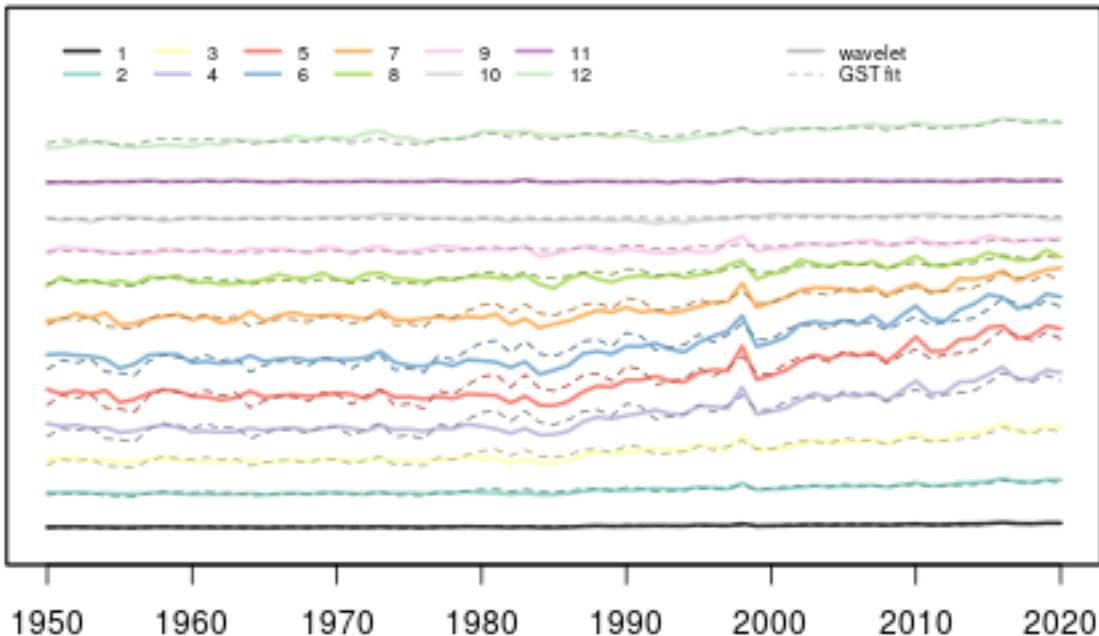
```

```

## -0.033218 -0.012691  0.000260  0.009349  0.057412
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.215500   0.092634  -2.326   0.0229 *
## x           0.030714   0.006522   4.709 1.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01831 on 69 degrees of freedom
## Multiple R-squared:  0.2432, Adjusted R-squared:  0.2323
## F-statistic: 22.18 on 1 and 69 DF,  p-value: 1.244e-05
##
##
## Call:
## lm(formula = y ~ x, data = cal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.200464 -0.060633  0.009122  0.072794  0.212906
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.77227    0.51873  -5.344 1.11e-06 ***
## x           0.59220    0.03652   16.215 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1025 on 69 degrees of freedom
## Multiple R-squared:  0.7921, Adjusted R-squared:  0.7891
## F-statistic: 262.9 on 1 and 69 DF,  p-value: < 2.2e-16
##
## Visualise the results:
power.fit <- zoo(power.fit,order.by=1950:2020)
np <- dim(power)[2]
cols <- c("#333333", brewer.pal(12, 'Set3'))
par(yaxt='n',mar=c(5,2,3,2))
for (i in 1:np) {
  coredata(power.fit)[,i] <- coredata(power.fit)[,i] - coredata(power[1,i]) + i - 0.5
  coredata(power)[,i] <- coredata(power)[,i] - coredata(power[1,i]) + i - 0.5
}
plot(power,plot.type='single',lwd=2,col=cols, main='Wavelet components',
      ylim=c(0,15),xlab='',ylab='')
legend(1950,15,1:np,col=cols,bty='n',ncol = 6,lty=1,lwd=2,cex=0.6)
legend(2000,15,c('wavelet','GST fit'),col='grey',lty=c(1,2),lwd=c(2,1),cex=0.6,bty='n')
#for (i in 1:np) text(1950,power[1,i],i,cex=0.5)
for (i in 1:np) lines(power.fit[,i],col=cols[i],lty=2)
for (i in 1:np) lines(power.fit[,i],col=rgb(0,0,0,0.3),lty=2)

```

## Wavelet components



**Fig. 8**

The following chunk provides the code used for generating Fig. 8. This figure provides a demonstration of how we may downscale rain patterns based on the results from Fig. 7, but in this case, the global mean temperature was actually not used.

```
#!/usr/bin/env Rscript

library(ncdf4)
library(waveslim)
library(raster)
library(argparser)
library(rgdal)

#=====
# FUNCTIONS - FUNCTIONS - FUNCTIONS - FUNCTIONS - FUNCTIONS - FUNCTIONS -
#=====

#+
plot_multires <- function( env, prefix="fig", shp=NA, par=NA) {
#-
  options( warn = 2)

  if ( !any(is.na(shp))) {
    b_or <- readOGR( shp[1], shp[2], verbose=F)
```

```

    b      <- spTransform( b_or, crs(env$r))
}

# set dyadic domain
xmn <- extent( env$r)[1]
xmx <- extent( env$r)[2]
ymn <- extent( env$r)[3]
ymx <- extent( env$r)[4]

nx <- ncol( env$r)
ny <- nrow( env$r)

n <- ceiling( log( max(nx,ny), 2))

rdyad <- raster( extent( xmn, xmx, ymn, ymx),
                  ncol=2**n, nrow=2**n, crs=crs( env))
rdyad[] <- 0

cat( paste( "dyadic domain, nx ny dx dy >", ncol(rdyad), nrow(rdyad), round( res(rdyad)[1]), round( r

dw<-list()
dw$wf <- "haar"
dw$n_levs_mx <- 11
dw$boundary <- "periodic"

# Initialization of the wavelet structures
# dwt_out. dwt.2d class.
dwt_out <- dwt.2d( as.matrix(rdyad), wf=dw$wf, J=dw$n_levs_mx, boundary=dw$boundary)
for (l in 1:dw$n_levs_mx) {
  dwt_out[[3*(l-1)+1]][] <- l*10 + 1 # LHi - wavelet coefficients
  dwt_out[[3*(l-1)+2]][] <- l*10 + 2 # HLi - wavelet coefficients
  dwt_out[[3*(l-1)+3]][] <- l*10 + 3 # HHHi - wavelet coefficients
}

# scaling function coefficients (father wavelet)
dwt_out[[3*(dw$n_levs_mx-1)+4]][] <- dw$n_levs_mx*10 + 4 # LL dw$n_levs_mx
for (i in 1:length(dwt_out)) dwt_out[[i]][] <- 0

# -- wave --
cat( paste( "wavelet decomposition","\n"))

# interpolate onto the dyadic grid
cat( paste( "resample...","\r"))
s <- env$r
cat( paste( "resample... ok","\n"))
s[s<0] <- 0

# define parameters used for plotting the figure
if ( any( is.na(par))) {
  rng_or <- range( c(getValues(env$r), getValues(s)))
  br_or <- c(0,0.1,1,2,5,10,100,280)
  col_or <- c("beige", colorRampPalette(c("azure", "cyan", "cornflowerblue", "blue", "purple"))(length(
} else {

```

```

rng_or <- par$rng_or
br_or <- par$br_or
col_or <- par$col_or
}
dx <- res(rdyad)[1]*100
dy <- res(rdyad)[2]*100

a <- vector()
a_l <- vector()
# transformation operator
cat( paste( "dwt...","\r"))
dwt <- dwt.2d( as.matrix(s), wf=dw$wf, J=dw$n_levs_mx, boundary=dw$boundary)
cat( paste( "dwt... ok","\n"))

# what happens when we enflate the energy?
# coefficients are those used for plotting Fig.8 of Benestad et al. (2022) referring to Hurricane Kat
# they have been obtained using the code "tpwave_multires_ampfact.r"
# coeff1 is the ratio between column3 and column2 of Tab.1
# coeff2 is the square root of "mres_reldev_future_q50_a" in "tpwave_multires_ampfact.r"
# coeff1, used to rescale Katrina's wavelet coefficients over the climate 1961-1991
coeff1 <- sqrt( c( 0.7700456, 0.7512890, 0.7406982, 0.7515059, 0.7958232, 0.8367114, 0.8771996, 0.910
# coeff2, used to rescale Katrina's wavelet coefficients over the climate 2021-2050
coeff2 <- sqrt( c( 1.290512, 1.329671, 1.343901, 1.346885, 1.306342, 1.253900, 1.209975, 1.165373, 1.

dwt_out1 <- dwt_out
dwt_out2 <- dwt_out

for (i in 1:length(dwt_out1)) dwt_out1[[i]][] <- 0
for (i in 1:length(dwt_out2)) dwt_out2[[i]][] <- 0

# rescale wavelet coefficients
for (l in 1:dw$n_levs_mx) {
  dwt_out1[[1+3*(l-1)]] <- coeff1[1] * dwt[[1+3*(l-1)]]
  dwt_out1[[2+3*(l-1)]] <- coeff1[1] * dwt[[2+3*(l-1)]]
  dwt_out1[[3+3*(l-1)]] <- coeff1[1] * dwt[[3+3*(l-1)]]
  dwt_out2[[1+3*(l-1)]] <- coeff2[1] * dwt[[1+3*(l-1)]]
  dwt_out2[[2+3*(l-1)]] <- coeff2[1] * dwt[[2+3*(l-1)]]
  dwt_out2[[3+3*(l-1)]] <- coeff2[1] * dwt[[3+3*(l-1)]]
}

dwt_out1[[4+3*(dw$n_levs_mx-1)]] <- dwt[[4+3*(dw$n_levs_mx-1)]]
dwt_out2[[4+3*(dw$n_levs_mx-1)]] <- dwt[[4+3*(dw$n_levs_mx-1)]]

qqor <- rdyad
qqin1 <- rdyad
qqin2 <- rdyad

# dwt[[4+3*(dw$n_levs_mx-1)]][] <- 0

qqor[] <- idwt.2d( dwt)
qqin1[] <- idwt.2d( dwt_out1)
qqin2[] <- idwt.2d( dwt_out2)

```

```

# x1 <- -107
# x2 <- -70
# y1 <- 10
# y2 <- 34
x1 <- argv$extent[1]
x2 <- argv$extent[2]
y1 <- argv$extent[3]
y2 <- argv$extent[4]

qqor<-crop(qqor,c(x1,x2,y1,y2))
qqin1<-crop(qqin1,c(x1,x2,y1,y2))
qqin2<-crop(qqin2,c(x1,x2,y1,y2))
qqor_a <- qqor
qqin1_a <- qqin1
qqin2_a <- qqin2
minn <- min( getValues(qqor))
maxx <- max( getValues(qqor))
maxx <- as.numeric( quantile( getValues(qqor), probs=0.999))
maxx <- 350
qqor[qqor>maxx] <- maxx
qqor[qqor<1] <- 0
qqin1[qqin1<1] <- 0
qqin2[qqin2<1] <- 0
qqin1[qqin1>maxx] <- maxx
qqin2[qqin2>maxx] <- maxx

# call the library used to load nice color tables
# library(fool)
# col_or<-load_color_table(path="/home/cristianl/projects/rpackages/fool/color_tables", abbrv="precip_1")
# can be replaced with the following
col_or <- c( "beige",rev( rainbow(11)))
br_or <- seq( minn, maxx, length=(length(col_or)+1))
br_or <- c(0, 1, 5, 10, 20, 30, 40, 50, 70, 120, 180, 240, 300)

# plot the panel (b) of Fig. 8 in Benestad et al. (2022)
# original precipitation field
png( file=paste0( "katrina_orig_2005.png"), height=800, width=1200)
par(mar=c(3,3,1,1))
image( qqor, main="", xlab="", ylab="", col=col_or, breaks=br_or,xlim=c(x1,x2),ylim=c(y1,y2),axes=F)
if ( !any(is.na(shp))) plot( b, add=T, lwd=3)
rect(x1,x2,y1,(y1+(y2-y1)/3),col="white")
legend(x=x1,y=(y1+(y2-y1)),fill=rev(col_or),legend=rev(c(br_or[2:length(br_or)])),cex=2.65)
axis(1,cex.axis=2)
axis(2,cex.axis=2)
box()
text(x=x1+13,y=y2-0.8,labels="(b) 2005-08-28",cex=4)
par(new=T)
par(mar=c(3,10,36,3))
plot(xyFromCell(qqor_a,1:ncell(qqor_a))[,1],getValues(qqor_a),axes=F,pch=21,bg="cornflowerblue",col="darkgray"
           ,xlim=c(x1+4.5,x2-2.1),ylim=c(0,300))
abline(h=seq(0,1000,by=50),col="darkgray",lty=2)
abline(h=seq(0),col="cornflowerblue",lty=1,lwd=5)
axis(2,cex.axis=1.5,las=1)

```

```

box()
dev.off()

# plot the panel (a) of Fig. 8 in Benestad et al. (2022)
# precipitation field rescaled according to 1961-1990 climate
png( file= paste0( "katrina_deflated_1961_1990.png"), height=800, width=1200)
par(mar=c(3,3,1,1))
image( qqin1, main="", xlab="", ylab="", col=col_or, breaks=br_or,xlim=c(x1,x2),ylim=c(y1,y2),axes=F)
if ( !any(is.na(shp))) plot( b, add=T, lwd=3)
rect(x1,x2,y1,(y1+(y2-y1)/3),col="white")
box()
text(x=x1+13,y=y2-0.8,labels="(a) 1961-1990",cex=4)
par(new=T)
par(mar=c(3,10,36,3))
plot(xyFromCell(qqin1_a,1:ncell(qqin1_a))[,1],getValues(qqin1_a),axes=F,pch=21,bg="cornflowerblue",co
    xlim=c(x1+4.5,x2-2.1),ylim=c(0,300))
abline(h=seq(0,1000,by=50),col="darkgray",lty=2)
abline(h=seq(0),col="cornflowerblue",lty=1,lwd=5)
axis(2,cex.axis=1.5,las=1)
box()
dev.off()

# plot the panel (c) of Fig. 8 in Benestad et al. (2022)
# precipitation field rescaled to match an hypothetical 2021-2050 climate
png( file= paste0( "katrina_eninflated_2021_2050.png"), height=800, width=1200)
par(mar=c(3,3,1,1))
image( qqin2, main="", xlab="", ylab="", col=col_or, breaks=br_or,xlim=c(x1,x2),ylim=c(y1,y2),axes=F)
if ( !any(is.na(shp))) plot( b, add=T, lwd=3)
rect(x1,x2,y1,(y1+(y2-y1)/3),col="white")
box()
text(x=x1+13,y=y2-0.8,labels="(c) 2021-2050",cex=4)
par(new=T)
par(mar=c(3,10,36,3))
plot(xyFromCell(qqin2_a,1:ncell(qqin2_a))[,1],getValues(qqin2_a),axes=F,pch=21,bg="cornflowerblue",co
    xlim=c(x1+4.5,x2-2.1),ylim=c(0,300))
abline(h=seq(0,1000,by=50),col="gray",lty=2)
axis(2,cex.axis=1.5,las=1)
box()
dev.off()
}

# Function to plot color bar
`color.bar` <- function(col, breaks, nticks=11, title='', cutTails=T,
    legtxt="",legdig=0,
    x1=1000000,
    y1=6450000,
    x2=1050000,
    y2=7530000,
    dx=50000,
    cex=2.5) {
#   scale = (length(lut)-1)/(max-min)
nbr<-length(breaks)
if (cutTails) {

```

```

min<-min(breaks[2:(nbr-1)])
max<-max(breaks[2:(nbr-1)])
ticks<-round(seq(2, (nbr-1), len=nticks),0)
} else {
  min<-min(breaks)
  max<-max(breaks)
  ticks<-round(seq(1, nbr, len=nticks),0)
}
dy<-(y2-y1)/length(col)
rect(x1, y1-dx,
      x2+1.5*dx, y2+dx,
      col="beige", border=NA)
text((x2+1.5*dx/2),
      (y1+dy/2+(ticks-1)*dy),
      round(breaks[ticks],legdig),
      cex=cex)
text((x1+x2)/2,
      y2+dx/2,
      legtxt,
      cex=cex)
for (i in 1:(length(col))) {
  y = (i-1)*dy + y1
  rect(x1,y,x2,y+dy, col=col[i], border=NA)
}
# rect(x1, y1,
#       x2+1.5*dx, y2+dx, border="black")
}

#=====
# MAIN - MAIN -
#=====

#
#-----
# Read command line arguments

p <- arg_parser("wave_it")

p <- add_argument(p, "--date1",
                  help="begin (%Y-%m-%d)",
                  type="character",
                  default="2005-08-28")

p <- add_argument(p, "--date2",
                  help="end (%Y-%m-%d)",
                  type="character",
                  default="2005-08-28")

p <- add_argument(p, "--ffin",
                  help="input file",
                  type="character",
                  default=".//data/ERA5_tp_day_20050828.nc")

```

```

p <- add_argument(p, "--extent",
                  help="extent",
                  type="numeric",
                  nargs=4,
                  default=c(-107,-70,10,34))

argv <- parse_args(p)

#
#-----


# Constants

date_format <- "%Y-%m-%d"
proj4.wgs84 <- "+proj=longlat +datum=WGS84"

#
# time

if ( !is.na(argv$date1) & !is.na(argv$date2)) {
  Rdate1 <- strptime( paste0(argv$date1, "T00"), format=paste0( date_format, "T%H"), tz="UTC")
  Rdate2 <- strptime( paste0(argv$date2, "T23"), format=paste0( date_format, "T%H"), tz="UTC")
}

#
# open input file

if ( !file.exists( argv$ffin)) q()
nc <- nc_open( argv$ffin)
tp <- nc$var[[2]]
ndims <- tp$ndims
start <- rep(1,ndims)
varsizes <- tp$varsizes

#
# define timesteps to read from file

tseq <- as.POSIXct( (tp$dim[[3]]$vals * 60 * 60), origin="1900-01-01", tz="UTC")

if ( !is.na(argv$date1) & !is.na(argv$date2)) {
  ixt <- which( as.numeric(tseq) >= as.numeric(Rdate1) &
                as.numeric(tseq) <= as.numeric(Rdate2))
} else {
  ixt <- 1:length(tseq)
}

#
# define spatial grids

lons_tot <- tp$dim[[1]]$vals
lats_tot <- tp$dim[[2]]$vals
w <- raster( nrows=length(lats_tot), ncols=2*length(lons_tot),
              xmjn=min(lons_tot-360)-0.125, xmjn=max(lons_tot)+0.125,

```

```

    ymn=min(lats_tot)-0.125, ymx=max(lats_tot)+0.125,
    crs=proj4.wgs84)
r <- crop(w, extent(c(-180.1, 179.8, -90.125, 90.125)))
xy<-xyFromCell(r,1:ncell(r))
lons<-sort(unique(xy[,1]))
lats<-sort(unique(xy[,2]),decreasing=T)
rm(xy)

# number of rows (cols) of the dyadic grid
dimdy <- 2**ceiling( log2( max( dim(r)[1:2])))

# dyadic grid
s <- raster( nrows=dimdy, ncols=dimdy,
             xmn=min(lons)-0.125, xmx=max(lons)+0.125,
             ymn=min(lats)-0.125, ymx=max(lats)+0.125,
             crs=proj4.wgs84)

nnscales <- log2(dimdy)+1
N <- nnscales-1
listscales <- 2**(seq(1,nnscales)-1)*1

if ( length(ixt) == 0) q()

#
# elaboration

nt      <- length(ixt)
En2o_t<-array(data=NA,dim=c(nnscales,nt))

# loop over the time steps
for( i in 1:nt ) {
  if ( ( i %% 10) == 0) cat( paste(i,"/",nt,"\n"))
  # read gridded data from nc-file
  start <- rep(1,ndims)
  start[ndims] <- ixt[i]
  count <- varsize
  count[ndims] <- 1
  data <- ncvar_get( nc, tp, start=start, count=count )
  # trick to get the precipitation field onto the expected grid
  w[] <- cbind( t(data), t(data)) * 1000
  r <- crop(w, extent(c(-180.1, 179.8, -90.125, 90.125)))
  # resample the precipitation field onto the dyadic domain
  u <- resample( r, s, method="bilinear")
  # store the raster on a dedicated environment
  env <- new.env( parent = emptyenv())
  env$r <- u
  # call plotting function
  # plot_multires(env,shp=c("/home/cristianl/data/geoinfo/TM_WORLD_BORDERS_LATLON/TM_WORLD_BORDERS-0.2.s
  plot_multires(env)
} # end loop over the time steps

#
# close the input file

```

```
nc_close(nc)

#
# Normal exit

q( status=0)
```

Benestad, R.E. 2018. "Implications of a Decrease in the Precipitation Area for the Past and the Future." *ERL*. IoP. <https://doi.org/10.1088/1748-9326/aab375>.