Rhine flood seasonality computation

Berry Boessenkool

August 2017

Data

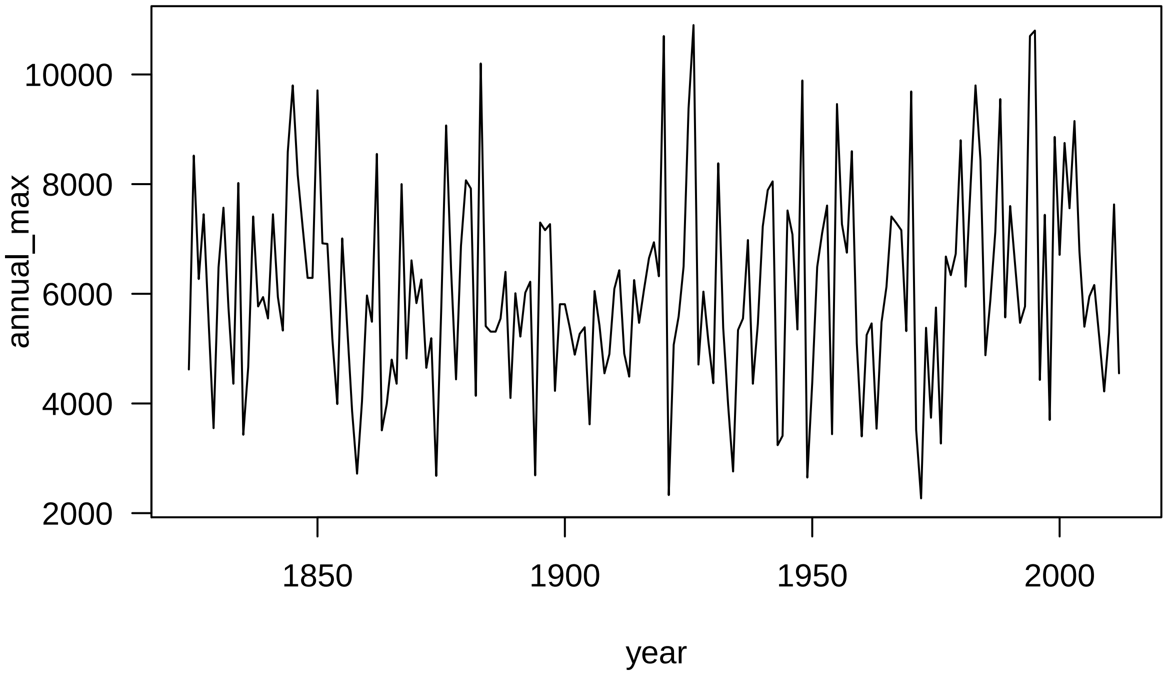
Cologne Rhine gauge daily discharge 1823-10-31 until 2011-12-31.
`rfs`: rhine flood seasonality R package, github.com/brry/rfs

```r
library(rfs)
load(seasFolder("data/dismeta.Rdata"))
```

Annual Maxima

Break between quasi-hydrological years is 117 days before Jan 1. This is the point of minimum discharge amongst all analyzed Rhine gauges.

```r
annual_max <- annualMax("date","Koeln", dis, shift=117)
plot(as.numeric(names(annual_max)), annual_max, type="l", las=1, xlab="year")
```



Extreme value analysis

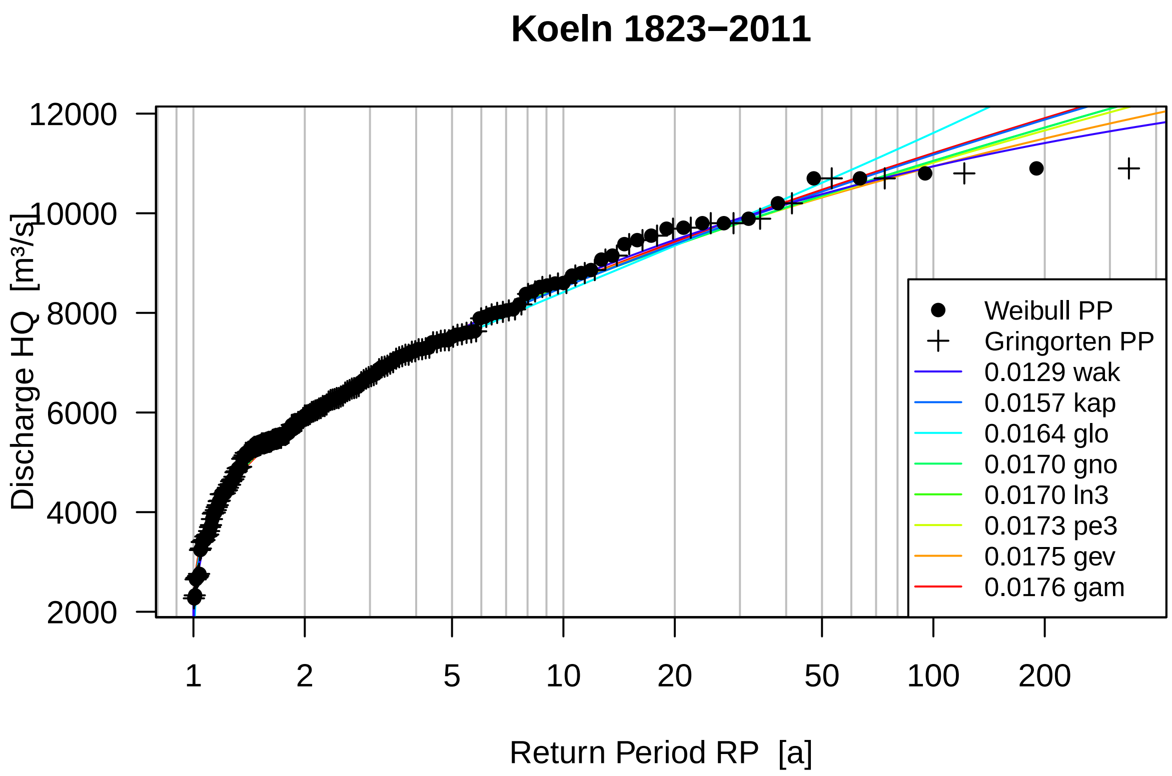
```
library(extremeStat)
dle <- distLextreme(annual_max, gpd=FALSE)
```

```
## Note in distLfit: dat was not a vector.

## Note in distLfit: 1 Inf/NA was omitted from 190 data points (0.5%).

## distLfit execution took 0.21 seconds.
```

```
plotLextreme(dle, log=TRUE, nbest=8, main="Koeln 1823-2011")
```



Computing return levels for given return periods.

A value `x` in a time series has a certain expected frequency to occur or be exceeded: the exceedance probability Pe. The Return Period (RP) of `x` can be computed as follows:

$$RP = \frac{1}{Pe} = \frac{1}{1 - Pne}$$

From that follows for the probability of non-exceedance (Pne):

$$Pne = 1 - \frac{1}{RP}$$

The Return Level (RL) can be computed as

$$RL = quantile(x, \quad prob = Pne)$$

Some common values are presented in the following table:

RP	prob
1.000	0.000
1.112	0.101
2.000	0.500
5.000	0.800
10.000	0.900
20.000	0.950
50.000	0.980
100.000	0.990
200.000	0.995

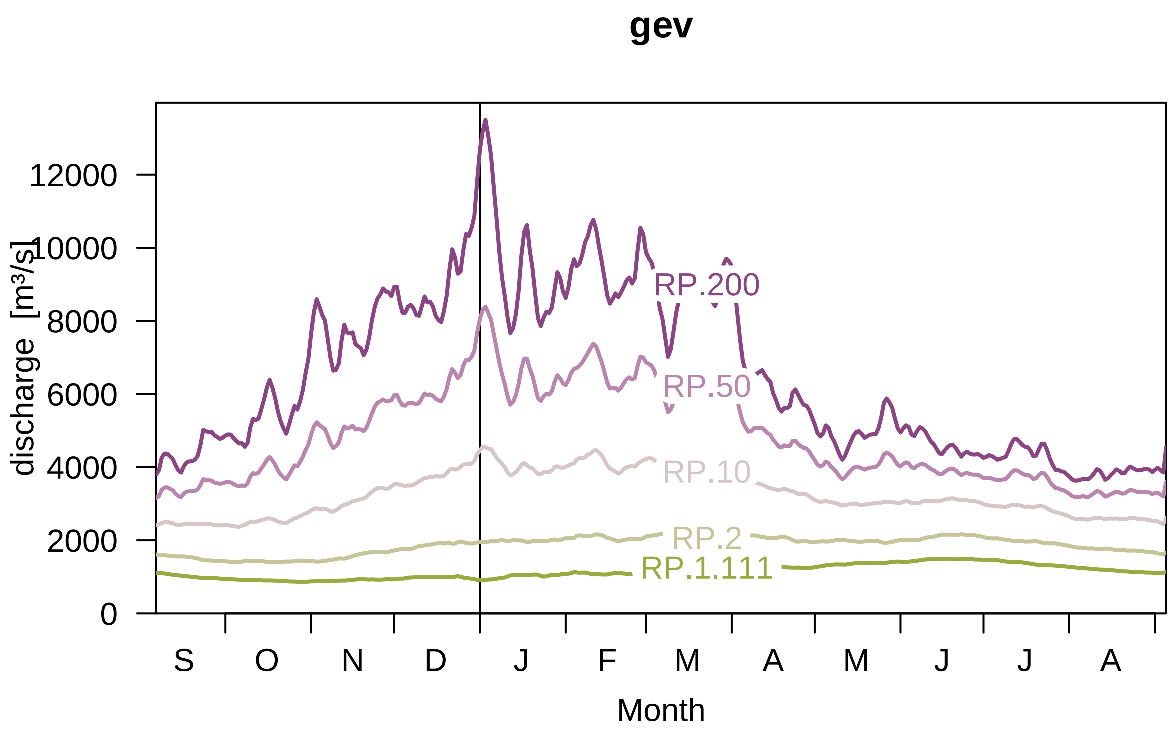
Flood seasonality

In classical extreme value analysis, the Block Maxima approach is often used. The extremes (maxima in blocks of e.g. years) are supposed to converge to the General Extreme Value distribution (GEV). Hence the GEV is fitted to the observations and the quantile function of the distribution used to get return levels. Often, hydrological years are used to ensure independence of the maxima. They split the year at a time when streamflow values tend to be small, e.g. in Germany at the end of October.

To look at the seasonality of floods, we have a similar approach, albeit not with actual block extrema. Return levels are computed via the GEV for each day of the year (DOY) separately. As a reference, they are also computed empirically with base R's `quantile` function with `type=8`.

The computation and visualization is implemented in the `rfs` package:

```r
load(seasFolder("data/dismeta.Rdata"))
qdoy <- qdoyCompute("date", "Koeln", data=dis, shift=117) # 10 secs for full time series
qdoyVis(qdoy, shift=117)
```

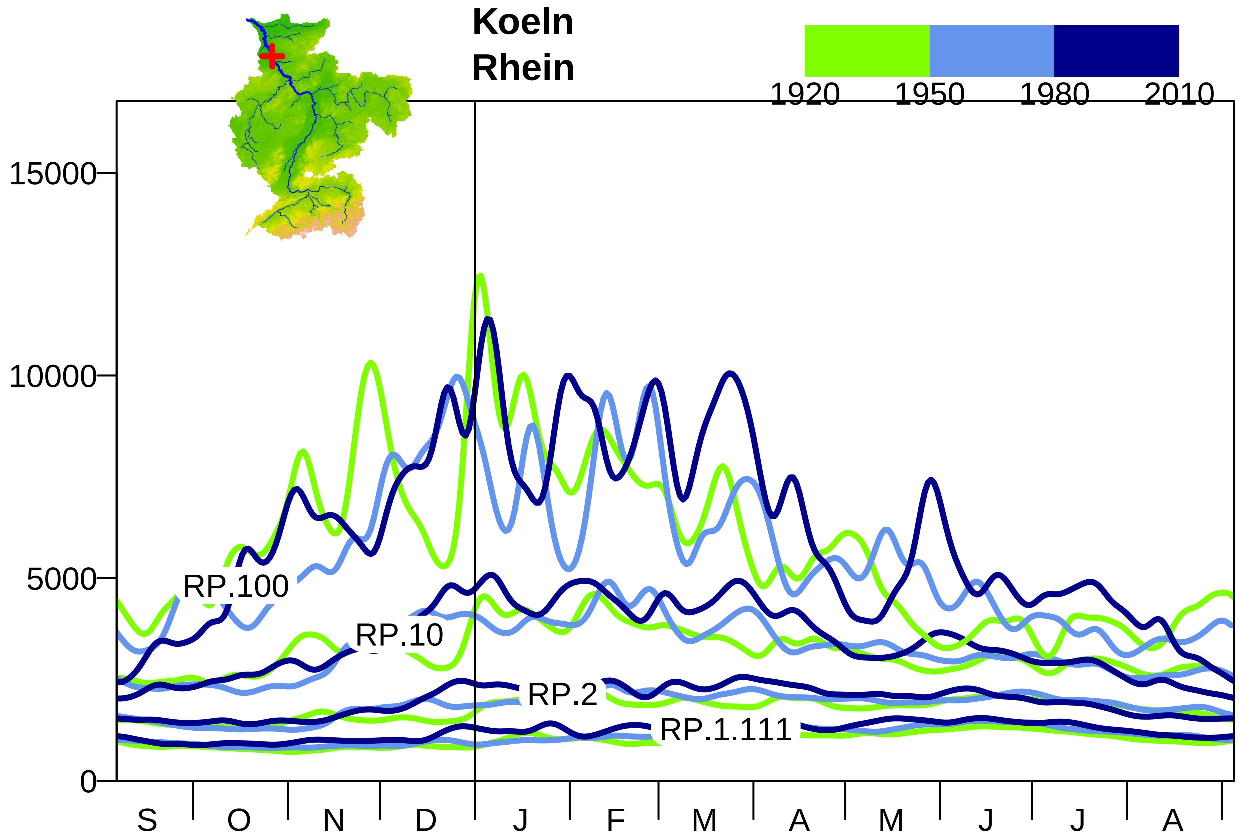


To compare the empirical and the GEV approach, an external pdf (>click to open<) is generated with the following code.

```
# Function to compare distributions:
distcomp <- function(dist, qd=qdoy, RPs=c(20,50,100), ...)
  {
  RPs <- paste0("RP.", sort(RPs))
  ciBand(qd[dist,RPs[3],],qd[dist,RPs[1],],qd[dist,RPs[2],], xaxs="i", ...)
  }
library(berryFunctions) # for lim0
# actual plot generation:
pdf("qdoy.pdf", height=5)
  par(mar=c(3,5,2,0.5), mgp=c(3.5,0.7,0))
  qdoyVis(qdoy, ylim=lim0(10e3), shift=117, dist="empirical")
  qdoyVis(qdoy, ylim=lim0(10e3), shift=117, dist="gev")
  #
  distcomp("empirical", xaxt="n", xlab="", main="Q20, Q50 and Q100 in Cologne, 1823-2011",
         ylab="discharge  [m\U{00B3}/s]")
  seasAxis(shift=117)
  distcomp("gev", add=TRUE, colm=4)
  legend("topright", c("empirical", "gev"), fill=c(3,4))
dev.off()
```

To examine changes of seasonalty, the analysis is performed separately for the observations in consecutive windows of 30 years length. For Cologne, to stick with the example, this looks like the following figure.

```
load(seasFolder("data/seas.Rdata"))
qdoyVisPeriods("Koeln", seas, sd=3)
```



Some smoothing of the line (via FFT) is already applied here to bring out the signal in the noise. The degree of smoothing can be specified with the sd parameter.

To interactively examine seasonality changes, the following app was created.

```
shiny::runApp(".")
```