Statistical Modelling of Environmental Extremes
Lab sheet Practical Session 3

Daniela Castro-Camilo · Institute Henri Poincaré, Paris – March 2022

To get started, open R/RStudio and install and load the necessary libraries:

Non-stationary threshold exceedances – Precipitation data

For the following tasks you need to load the `prcpFC.txt` dataset.

Task 1 - Naive approach

1. Use the `ismev::gpd.fit` function to fit a stationary GPD model to exceedances to excesses over 1in. Use the mean residual life plot (MRLP) to check if 1in is a suitable threshold.

MRLPs can sometimes be hard to interpret as a method for threshold selection. A complementary technique is to fit the GPD at a range of thresholds, and to look for stability of parameter estimates. The idea is that if a GPD(σ_{u_0}, ξ) is a reasonable model for excesses of a threshold u_0, then excesses of a **higher** threshold $u > u_0$ should follow a GPD(σ_u, ξ). The shape parameter for both models is the same, but the scale changes (remember that the GPD scale depends on the threshold). The relationship between both scales is

$$\sigma_u = \sigma_{u_0} + \xi(u - u_0) \Rightarrow \sigma_u^\star = \sigma_u - \sigma_{u_0} = \xi(u - u_0).$$

Note that $\sigma_u^\star$ is constant w.r.t. u. Consequently, **if u_0 is a valid threshold for excesses, then estimates of both $\sigma_u^\star$ and ξ should be constant above u_0.**

The argument above suggests that we plot $\sigma_u^\star$ and ξ against u and select u_0 as the lowest value of u for which the estimates remain near-constant (why?). Naturally, CIs should be included. Confidence intervals for ξ can be obtained from the variance-covariace matrix V (e.g., the `cov` object in the `ismev::gpd.fit` output) but for $\sigma_u^\star$ we need the delta method (bootstrap could also be used). Doing the math, we get

$$\text{Var}(\sigma_u^\star) \approx \nabla\sigma_u^{\star\top} V \nabla\sigma_u^\star, \quad \text{where} \quad \nabla\sigma_u^\star = \left[\frac{\partial\sigma_u^\star}{\partial\sigma_u}, \frac{\partial\sigma_u^\star}{\partial\xi}\right] = [1, -u].$$

2. Create a function called `tsplot` that computes the above. Use it to check that the selected threshold is a *good* choice.

Task 2 - Slightly less naive approach

Using the threshold selected in Task 1, fit a GPD model with a linear trend in the log-scale, i.e.,

$$\begin{aligned} Y_t &\sim \text{GPD}(\sigma(t), \xi) \\ \log(\sigma(t)) &= \gamma_0 + \gamma_1 t \end{aligned}$$

Task 3 - Single season approach

The focus here is to fit a GPD to data from a season that gives rise to the "most extreme". This approach assumes that data are stationary within the season. Using the data from July-Aug-Sept, fit a stationary GPD model to excesses over 1in.

Task 4 - Seasonal piecewise approach

The exploratory plots showed a seasonal variation of daily precipitation with months. To account for this monthly effect, we can fit 12 stationary GP models, one for each month, where the threshold is month-specific. This approach assumes stationarity within each month and could be a good way to circumvent modelling the seasonality directly.

1. Fit a stationary GP model for each month. For simplicity, you can choose the thresholds as the 99% quantile in each month. You can report results in a table with columns, `month`, `u` (threshold), `nexc` (number of exceedances), `pexc` (proportion of exceedances), `sc.mle` (scale MLE), `sc.se` (scale std error), `sh.mle` (shape MLE), `sh.se` (shape std error).

This approach avoids the problems of non–stationarity as a result of seasonal variability, provided we can safely assume stationarity within each seasonal unit (in this case, each month). However, in terms of return level inference, it would not make practical sense to have monthly varying estimates of the r–year return level z_r. To include information from all months in our return level estimation procedure, we need to solve for $\hat{z}_r$ the equation

$$\prod_{m=1}^{12} \left\{ 1 - \hat{\zeta}_u \left[1 + \hat{\xi}_m \left(\frac{\hat{z}_r - u_m}{\hat{\sigma}_m} \right) \right]_+^{-1/\hat{\xi}_m} \right\} - \left(1 - \frac{1}{r n_y} \right) = 0,$$

where n_y is the (average) number of observations per year and $\hat{\zeta}_u$ is the estimated proportion of exceedances.

2. Implement a solution to the function above (using, e.g., `uniroot`) to estimate the 100-year return level precipitation.

Note that the solution to the above equation only allows us to provide point estimates for the return levels. A complete analysis would also report the uncertainty of these estimates. Standard deviations of the return level estimates can be obtained by the delta method or profile likelihood (not covered).

Task 5 - Smoothly varying seasonal parameters using GAMs

Exploratory plots showed that there might be non-linear monthly and yearly effects. One way to capture these effects is using generalised additive (extreme-value) models. In a nutshell, this framework allows the GPD/GEV parameters to flexibly vary as a function of covariates, in our case, months and years. Recalling our notation from Session 1, a GPD-GAM for our data can be specified as follows

$$\begin{aligned} Y(t) &\sim \text{GPD}(\sigma(t), \xi) \\ \log(\sigma(t)) &= \gamma_0 + f_1(\texttt{month}(t)) + f_2(\texttt{year}(t)), \quad t = 1, 2, \ldots, \end{aligned} \tag{1}$$

where $\texttt{month}(t)$ and $\texttt{year}(t)$ are the month and year of observation t, respectively. As mentioned in Session 1, the smooth terms f_1 and f_2 are represented using known basis functions. In `evgam`, smooth terms are represented using penalised regression **splines** (as in the `mgcv` package. In fact, `evgam` is a bridge between the different smooths available in `mgcv` and extreme-value distributions with parameters of GAM form). Examples are

- Linear splines: $f(t) = \beta_0 + \beta_1 t + \sum_l \beta_{1+l}(t - k_l)_+$
- Quadratic splines: $f(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \sum_l \beta_{2+l}(t - k_l)_+^2$
- Cubic splines: $f(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \sum_l \beta_{3+l}(t - k_l)_+^3$
- Polynomial splines: $f(t) = \beta_0 + \beta_1 t + \ldots + \beta_Q t^Q + \sum_l \beta_{Q+l}(t - k_l)_+^Q$

where $\{k_l\}_{l=1}^L$ are a pre-specified set of knots in an interval $[a, b]$ with $k_1 = a$ and $k_L = b$ and β_l denote coefficients. Every spline function $f(t)$ of a given degree can be uniquely represented as a linear combination of, e.g., B-splines $b_l(t)$ of that same degree, i.e.,

$$f(t) = \sum_{l=1}^{L} \beta_l b_l(t) \tag{2}$$

Figure 1 shows different types of B-splines. Note that the are other types of splines; see, e.g., the `mgcv` documentation and Wood (2006).

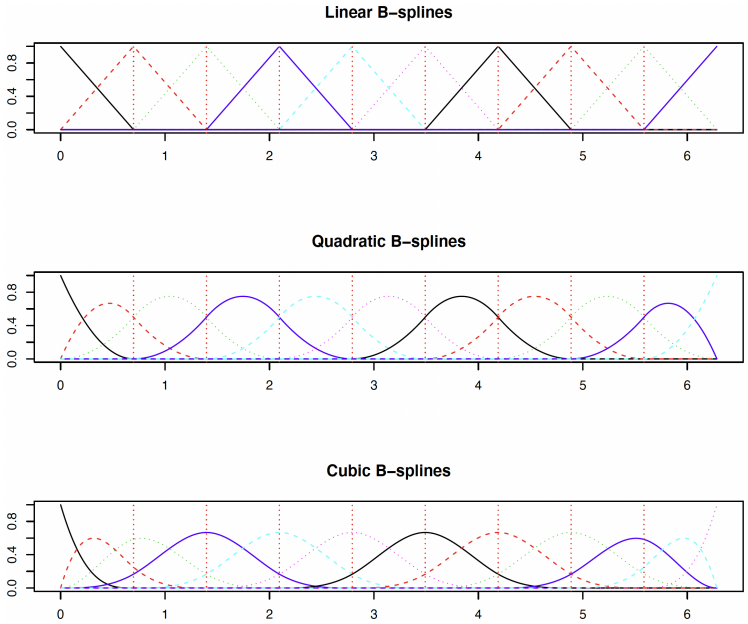

Figure 1: Different types of B-splines. Taken from KAUST STAT380 class notes by R. Huser.

The package `evgam` mainly relies on its eponymous function `evgam()`. Its main arguments are `evgam(formula, data, family)`. `formula` is a list comprising formulae: one formula compatible with `mgcv::s()` for each GPD/GEV parameter. `mgcv::s()` is a generic function to define splines. It should be specified in terms of a covariate (`month` and `year` in our case) and can take many additional arguments. The most important ones are

- `k`: basis dimension.
- `bs`: penalised smoothing basis to use, e.g., `cr` for cubic regression, `cc` for cyclic cubic regression or `tp` for thin plate regression splines (default). Thin plate splines are very flexible and parsimonious and do not need the `k` specification.

For example, `s(month, bs="cc", k=12)` defines a cyclic cubic spline over the 12 months, while `s(year)` defines a thin plate spline over the years. Using their respective fixed basis expansions, estimation of the unknown smooth functions boils down to inference over the coefficients; see (2). In `evgam` (and `mgcv`), inference is performed using penalised likelihood methods (Wood, 2011, Wood et al., 2016). The following code fits the model in (2):

```
fmla_gpd = list(excess ~ s(month, bs="cc", k=11) + s(year), ~1)
fit3 = evgam(fmla_gpd,
             data = fort_gpd,
             family = "gpd")
```

Note that `formula` is a list of length 2 specified one formula for each parameter (scale and shape). The expression `~1` means that the shape parameter is assumed to be constant (it can be changed). Note that it is not necessary to specify a log-link for the scale since the log-link is used through `evgam` for any parameters with support in $\mathbb{R}^+$. The other arguments, `data` and `family` specified the exceedance data (not the original data) and the GP family of distributions. Run the code above and use `summary` and `plot` to investigate the fit.

The following lines show how to plot the annual effect by month. This plot is useful to see differences between the effects of different months throughout the years. Explore and run the code.

```
## Cyclic annual effect by month
year2pred = seq(min(fort_gpd$year),max(fort_gpd$year), length.out = 100)
```

```r
yeffect = NULL
months = sort(unique(fort_gpd$month))
for(mm in months){
  newdata = data.frame("month" = mm, "year" = year2pred)
  pred = predict(fit3, type = "response", newdata = newdata)
  yeffect = cbind(yeffect, as.numeric(pred$scale))
}
colnames(yeffect) = month.abb[months]
head(yeffect)

## Plot by month
par(mar = c(3,3.2,1.5,0.5), mgp = c(1.6,0.5,0),font.main = 1.3,
    cex = 1.3, cex.main = 1)
col2use = viridis::plasma(length(months))
plot(year2pred, yeffect[,1], type = "l", ylab = "Fitted scale", xlab = "Time",
     col = col2use[1], ylim = range(yeffect), lty = 2)
text(x = year2pred[1], y = yeffect[1,1], labels = month.abb[months[1]], cex=0.6,
     col = col2use[1])
for(mm in 2:length(months)){
  lines(year2pred, yeffect[ ,mm], col = col2use[mm])
  text(x = year2pred[6*mm], y = yeffect[6*mm,mm],
       labels = month.abb[months[mm]], cex = 0.6, col = col2use[mm])
}
```

Task 6 - Threshold exceedances using quantile regression

Non-parametric quantile regression can be use to estimate a time-varying threshold $u(t)$. Exceedances over that threshold can then be fitted using stationary or non-stationary GP models. Using the naive model, fit a GPD to excesses over the 99.5% time-varying quantile.

Caution must be taken with using this procedure. This is a two-steps approach, which means that uncertainty in choosing $u(t)$ should be propagated (although in most applications it is usually neglected). Considering that we are using non-parametric techniques to estimate high quantiles, uncertainty can be highly non-negligible!

Question: Can you think of another way to estimate a time-varying threshold?

References

- Wood, S. N. (2006). *Generalized additive models: an introduction with R*. Chapman and hall/CRC.
- Wood, S. N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 73(1), 3–36.
- Wood, S. N., Pya, N. and Säfken, B. (2016) Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association* 111(516), 1548– 1563.