# STAT0017 Extremes Computer Practical 1

[*This instruction sheet is available on the Topic 1 - practicals section of the STAT0017 Moodle page. I suggest that you download it in order that you can make use of the hyperlinks (in blue).*]

## Preliminaries

The aim is to use some of the R software available to fit extreme value models to data and to make inferences from these models. The code that you will use is a mixture of R functions from contributed R extreme value packages, namely evd, evdbayes, ismev, revdbayes, threshr, and the packages bayesplot and coda (MCMC output analysis). I thank the authors for the time and effort they put into providing these resources. I have also provided a random assortment of extra functions (in the file pjnfunctions.R). In many instances this code is a modified version of code from the packages listed above.

These packages have reference manuals (available on their CRAN pages via the links above) that provide help documentation for individual functions and, in particular, brief example code. Some also have a general package help page (e.g. `?revdbayes`) with a link to an index of functions at the bottom. Of course, you can get help on a particular function: e.g. `?gev.fit`; but you need to know the name of the function first. You can also do a keyword search: e.g. `??gev`.

Some packages also provide user's guides or vignettes, which provide a better overview of the package and explain in more detail what the functions do. These can be hard to find in some cases, so I have provided links to on the practicals section of the Moodle page.

There are several other useful R packages: the CRAN Task View on Extreme Value Analysis provides a summary.

In some cases I have provided complete code to carry out analyses. This shouldn't prevent this from experimenting: the help files for the functions involved will provide information about how you could adapt the code.

## Getting started

Open RStudio install the packages: `evd, evdbayes, ismev, revdbayes, threshr, bayesplot` and `coda`. For example,

```
install.packages("revdbayes")
```

Download the file `pjnfunctions.R` from the Moodle page and save it somewhere on your `N:` drive, e.g. `N:/EVT/pjnfunctions.R`. Read these functions into R using

```
source("N:/EVT/pjnfunctions.R")
```

**Masking of objects**. This occurs when packages contain objects with identical names. For example, if you load `revdbayes`, using `library(revdbayes)`, and then load `evd` then you will get an error message:

The following objects are masked from package:revdbayes:

dgev, pgev, qgev, rgev

Both `revdbayes` and `evd` contain functions `dgev, pgev, qgev, rgev`. This shouldn't matter for the purposes of this practical but it is an issue you should bear in mind more generally. All versions of these functions are available but R will by default use the version in the most recently loaded package. If you specifically want the `revdbayes` version of `dgev`, for example, then you can use `revdbayes::dgev`.

**Bayesian inference**. We haven't covered this properly in lectures (yet). However, it makes sense to introduce the relevant software at the same time as maximum likelihood estimation. The following points provide the bare essentials of Bayesian inference.

- Model parameters are treated as random variables. This is unlike frequentist inference, where they are fixed unknown constants.
- A *prior* distribution is set for the parameters of the model that summaries information (if any) about the parameters that are external to the data.
- Information in the prior is combined with information in the data (contained in the *likelihood*), using Bayes' theorem, to form the *posterior* distribution of the parameters.
- The posterior distribution is typically studied by simulating a very large sample from it, often using Markov Chain Monte Carlo (MCMC).
- MCMC simulates from a Markov chain whose equilibrium distribution equals the posterior distribution. After approximate convergence (following a *burn-in period*) this provides a *dependent* sample approximately from the posterior distribution.
- In a few cases it is possible to simulate a random sample directly from the posterior distribution. This is is what the package revdbayes does, using the *ratio-of-uniforms method* implemented using the rust package.

The evdbayes user's guide provides a good introduction to Bayesian inference it's use in extreme value modelling. It also gives details about Markov Chain Monte Carlo (MCMC). The revdbayes and bayesplot vignettes provide information about posterior predictive inference.

**Example datasets**. The practical is split into sections based on different example datasets and analyses. Each section has a file (available from the Moodle page) containing R code and some brief comments to explain what the code does.

Please do think about what the code is doing and what the output means. Please ask me, or do some reading, if you are unsure.

This practical focuses on

- fitting a Generalised Extreme Value (GEV) distribution to block maxima;
- fitting a Generalised Pareto (GP) distribution to threshold excesses;
- model disgnostics, that is, checking the fit of these models;
- making extreme value inferences.

We use both maximum likelihood estimation and Bayesian inference.

# Portpirie data: GEV analyses of annual maxima

Download the file portpirie.R and open it as a script.

Note: `ismev`, `evd` and `revdbayes` contain these data, but in different formats! `ismev` has a data frame with two columns named `Year` and `SeaLevel`. `evd` and `revdbayes` have a vector of the sea levels. Code is included to convert from `ismev` format if necessary.

# Wave height data: GP analyses of threshold excesses

Download the file waves.R and open it as a script.

In this script far fewer commands are provided than in `portpirie.R`. The general idea is the same: we wish to fit and check an extreme value model and inferences; but now our modelling is based on the GP distribution.

Where general instructions are provided rather than code please make use of the help system to help you to construct the code that you need. In some cases code has been provided but with missing arguments (indicated by a ?).

The data used in this section feature in Northrop et al. (2017).

# R markdown

R markdown merges text, mathematics, R code and output to produce a report. It should come installed with RStudio. Later, you may find it a convenient way to produce at least part of your report for the in-course assessment. See Introduction to R markdown and How R markdown works to find out more.

I have provided an example R markdown file example.Rmd. This is a cutdown and slightly edited version of the main revdbayes vignette. The produce a PDF from using this file use the Knit button in Rstudio (near the top left) and choose the 'Knit to pdf_document' option. This may take a minute or so because quite a bit of simulation is performed. This .Rmd file was originally set up for html formatting: the plots could do with some tweaking for the pdf version. Also provided is the BibTeX file revdbayes.bib used to produce the references.

The vignettes of (other) R packages may provide useful further examples. These are accessible using, for example, `help(package = revdbayes)` and then following the `User guides, package vignettes and other documentation` link and then the `source`.

For a convenient way to see the vignettes of the packages that you have installed, and to access their source code, is

```
browseVignettes()
```

Paul Northrop
24th January 2019