MAS8304: Problems sheet 1

Estimating Environmental Extremes

Answers to all questions (except question 4) should be placed in the homework submission box in the foyer of the Maths & Stats General Office (Herschel Building 3rd floor) by no later than **4pm, Monday 25th February** (please note that the new hand–in arrangements for coursework affect this module!).

1. Given the cumulative distribution function for the generalised extreme value (GEV) distribution, as provided by Equations (2.6) and (2.7) in the lecture notes (for $\xi \neq 0$ and $\xi = 0$, respectively), obtain the corresponding probability density function for the GEV. For the case $\xi = 0$, show that the log–likelihood is given by

$$\ell(\mu, \sigma; \boldsymbol{x}) = -m\log\sigma - \sum_{i=1}^{m} \left\{ \frac{x_i - \mu}{\sigma} \right\} - \sum_{i=1}^{m} \exp\left\{ -\left(\frac{x_i - \mu}{\sigma} \right) \right\},$$

where m is the number of block maxima.

2. A type I extreme value distribution, with distribution function given by

$$G(x; \mu, \sigma) = \exp\left\{ -\exp\left[-\left(\frac{x - \mu}{\sigma} \right) \right] \right\}$$

for $-\infty < \mu < \infty$ and $\sigma > 0$, is used to model annual maximum wind speeds (mph) at Corpus Christi, Texas, for the years 1943 and 1976 (inclusive). The maximum likelihood estimates for μ and σ are found to be $\hat{\mu} = 45.86$ and $\hat{\sigma} = 3.56$; the hessian matrix for the log–likelihood is also found as

$$H = \begin{pmatrix} -2.542674 & 1.591807 \\ 1.591807 & -4.486644 \end{pmatrix}.$$

(a) Estimate the standard errors for $\hat{\mu}$ and $\hat{\sigma}$, and use these to construct 95% confidence intervals in the usual way.

(b) Find $\mathsf{cor}(\hat{\mu}, \hat{\sigma})$.

(c) Obtain an expression for the r–year return level using this type I extreme value distribution. Thus, estimate the 50–year and 500–year return level wind speeds at Corpus Christi, and obtain standard errors for these estimates via the delta method. Why is it not advisable to construct confidence intervals for return level estimates in the usual way?

3. Suppose $X_1, X_2 \ldots, X_n$ is a sequence of independent Gumbel random variables. By letting $a_n = 1$ and $b_n = \log n$, show that the distribution of $(M_n - b_n)/a_n$ is of extreme value type, where

$$M_n = \max\{X_1, X_2, \ldots, X_n\}.$$

Practical work

Open R by clicking on `Start` → `All Programs` → `Statistical Software` → `R` → `R`. You will also need to access data for some of these questions from the course website, so have this open:

`www.mas.ncl.ac.uk/~nlf8/Teaching/MAS8304/home.html`

No solution to question 4 is required in your submission.

4. Recall the first step in a typical analysis of extremes: your data may need to pre–processed to extract the block maxima to which the generalised extreme value distribution can be fitted. In the sea–surge example used throughout Chapter 2 of the lecture notes, no data pre–processing was necessary, as we already had the set of annual maximum sea surges for the years 1955–2004 (inclusive).

(a) Suppose you have daily data collected over a period of seven years – from 2003 to 2009 (inclusive) – and you want to extract the seven annual maxima from this series. We can do this in R. First of all, set up a vector of years using the code:

```
> year=seq(2003,2009,1)
```

The R function `seq(`a`,`b`,`c`)` creates a sequence of values from a up to b in steps of c. Take a look at your vector `year` to make sure you get the desired result, by typing:

```
> year
```

(b) If we use calendar years as our blocks, we should ensure that we take into account leap years. The following code creates a vector the same length as `year` called `ind`; this will indicate whether or not the corresponding year is a leap year by taking the value 366 for leap years, and 365 elsewhere (a leap year has 366 days, a non–leap year 365). Look through the code, and make sure you fully understand each line, before applying this in R.

```
> year.over.four=year/4
> rounded=round(year.over.four)
> ind=vector("numeric",length(year))
> for(i in 1:length(ind))
  {
    if(rounded[i]==year.over.four[i])
      {
        ind[i]=366
      }
    else{
      ind[i]=365
        }
   }
```

(c) When obtaining the set of annual maxima, we want to find the maximum value between `lower` and `upper` in each block: for example, for our first year of data, `lower` would be 1 and `upper` 365 (since 2003 is not a leap year), for the second year of data, `lower` would be 366 and `upper` 731 (since 2004 *is* a leap year).

We can compute `lower` and `upper` for each year using the following code – read through this carefully, and then type it into R.

```
> upper=vector("numeric",length(ind))
> for(i in 1:length(upper))
  {
     upper[i]=sum(ind[1:i])
   }
> lower=vector("numeric",length(ind))
> lower[1]=1
> for(i in 2:length(lower))
  {
    lower[i]=upper[i-1]+1
  }
```

(d) Now typing

```
> cbind(year,ind,lower,upper)
```

gives

```
     year ind lower upper
[1,] 2003 365     1   365
[2,] 2004 366   366   731
[3,] 2005 365   732  1096
[4,] 2006 365  1097  1461
[5,] 2007 365  1462  1826
[6,] 2008 366  1827  2192
[7,] 2009 365  2193  2557
```

(e) Suppose our daily measurements observed over the seven years are stored in the vector `mydata`; then the following code would extract the set of annual maxima:

```
> anmax=vector("numeric",length(year))
> for(i in 1:length(anmax))
   {
    anmax[i]=max(mydata[lower[i]:upper[i]])
   }
```

Again, read through this code carefully and make sure you fully understand what will happen in the `for` loop. *Do not attempt to execute this part of the code, as you won't have anything stored in* `mydata`*!*

(f) How would you amend the above procedure, as outlined in parts (a)–(e), if you had *hourly* observations?

5. Daily rainfall accumulations have been recorded at 25 sites in England over a period of 74 years, from 1938 to 2011 inclusive. You have been randomly allocated to one of these sites. Go to the assignments section of the course website and click on the "rainfall data" link next to the material for Problems sheet 1; then right–click on your name and `Save Target As` "rainfall.txt", making sure you save your data file in a suitable folder on your `H:` drive.

Now you can load the rainfall data into R by typing

```
> rain=scan('rainfall.txt')
```

This should read the rainfall data into R and store them in the vector `rain`. If you get the following message:

```
Cannot open file 'rainfall.txt': No such file or directory
```

then you do not have R open in the same directory as you saved your data file. To change to the correct directory, click `File`→`Change dir...` and then browse through your folders until you find the folder in which you saved your data file – select this folder, and then click `OK`. Repeating the `scan` code above should now successfully load your rainfall data into R.

(a) By hand/using a calculator, work out how many observations there *should* be in `rain`. Now use the R function `length`, or otherwise, to find out how many observations you *actually* have. What do you notice?

(b) Actually, data is missing for the year 1944 during the second world war. How does this explain your answer to part (a)?

(c) Following the example code given in question 4(a), set up a `year` vector in R for your rainfall data – being careful to exclude the year 1944. Now follow the example R code shown in the rest of question 4 to extract the set of annual rainfall maxima for your site, storing your set of maxima in the vector `anmax`.

Use the R command `summary` to obtain the five number summary for your annual maximum rainfall accumulations (min, lower quartile, median, upper quartile, max), and include these in your solutions.

(d) Produce a time series plot by plotting `anmax` against `year`, and include this plot in your solutions (using appropriate labels for the title/axes). Also produce a histogram of your annual maxima. Comment.

6. This question will make use of the annual maximum rainfall accumulations you extracted from your rainfall dataset in question 5.

(a) Following the material in Section 2.2.2 of the lecture notes, use the `nlm` function in R to fit the GEV to the set of annual maximum rainfall accumulations at your site. In your solution, include the output R produces when you apply the `nlm` routine, including the `minimum` of the function, the `estimates`, the convergence `code` and the number of `iterations` (see page 23 of the lecture notes). What is the value of your maximised GEV log–likelihood?

(b) Again, following the example in Section 2.2.2 of the lecture notes, use R to invert the hessian matrix and thus obtain the variance–covariance matrix V. In your solutions, write down V, and also write down your corresponding estimated standard errors for $\hat{\mu}$, $\hat{\sigma}$ and $\hat{\xi}$. Use these standard errors to construct 95% confidence intervals for the GEV parameters in the usual way, and comment.

(c) Estimate the 10–year, 100–year and 1000–year return levels for annual maximum rainfall accumulations at your site. Also, use the delta method to estimate the standard errors for your return level estimates, following the example in Section 2.2.4 of the lecture notes. It is not necessary to include any R code here; just produce a table summarising your results similar to Table 2.2 in the lecture notes.

(d) Use the function `gev.fit` within the ismev package in R to confirm your answers to parts (a) and (b) – include in your solutions the R output given on execution of `gev.fit`. What are the differences between the `gev.fit` fitting procedure and fitting the GEV "from first principles" using `nlm`?

7. The R package evd is another add–on package for extreme value analyses. Load the package into R by typing

```
> library(evd)
```

We will use many of the functions provided by `evd` in the next problems sheet. For now, one interesting suite of functions is that which calculates the GEV probability density (`dgev`), cumulative distribution function (`pgev`), quantile function (`qgev`) and allows random generation of extremes from a GEV (`rgev`). Explore this suite of functions by typing

```
> ?pgev
```

These functions work in exactly the same way as th analogous functions for the Normal distribution or the Poisson distribution – for example, `dnorm`, `rnorm` or `ppois`.

(a) Generate a sequence of values between 0 and 50, in steps of 0.1, using the command `seq` (see question 4(a) for a reminder). Store this sequence in `x`.

(b) Now use the command `dgev` to find the GEV probability density for each value in `x`, using $\mu = 20$, $\sigma = 3.5$ and $\xi = 0.3$ by typing

```
> y1=dgev(x,loc=20,scale=3.5,shape=0.3)
```

(c) Produce a density plot for the GEV $\mathcal{G}(\mu = 20, \sigma = 3.5, \xi = 0.3)$ distribution, by plotting `y1` against `x`. Use `plot`, and join the points by using `type=l`.

(d) Following parts (b) and (c), also produce a plot of the GEV densities when $\xi = 0$ and $\xi < 0$. Superimpose these densities on the plot you produced in part (c) by using the `lines` command (instead of `plot`), and use different colours to distinguish between the three plots by using the argument `color='blue'`, for example.

(e) What do you notice when you compare your three density plots?

(f) Now experiment with three different values of μ, keeping both σ and ξ constant.

(g) Now experiment with three different values of σ, keeping both μ and ξ constant.