# Extreme Value Analysis Workshop I: Maxima and Minima

## Tuesday 5th September, 2017

## 1.1 Getting Started

In this session we will use the R package `extRemes`. To install the package

```
> install.packages("extRemes")
```

Each time you want to use this package you will need to load it into your current R workspace using

```
> library(extRemes)
```

Other packages that do roughly similar things to the `extRemes` package include `texmex`, `ismev` and `evd`.

You should also install and load the `lubridate` package and, if you like fancy graphics, the `ggplot2` package.

## 1.2 Modelling Sea Ice Minima

Daily sea ice extents in the Arctic and Antarctic have been downloaded from

`https://nsidc.org/data/seaice_index/archives.html`

You can find the spreadsheets at **XYZ**.

### 1.2.1 Exploratory Data Analysis

First download the annual minimum sea ice extents for the Arctic and Antarctic. Load these into R, e.g.

```
> arctic <- read.csv("N_seaice_extent_daily_v2.1.csv")
```

We will start with some exploratory analysis of the data.

1. To see what variables the `arctic` data set contains use

   ```
   > names(arctic)
   ```

2. How many observations have you got? Try

   ```
   > nrow(arctic)
   ```

3. Now plot the data in time. First we should add a dates column to our data frame; this is easiest using the `dmy` function in the `lubridate` package:

```
> library(lubridate)
> arctic$Date <- dmy(paste(arctic$Day,arctic$Month,arctic$Year,sep="-"))
```

Plotting can either be done using the base `plot` function or the plotting functions available in `ggplot2`. Using the base function:

```
> plot(arctic$Date,arctic$Extent,xlab="Date",ylab="Extent")
```

Using `ggplot2`:

```
> ggplot(data=arctic)+geom_point(aes(x=Date,y=Extent))
```

Comment on any trends or patterns you see in the data.

4. Finally we will check the dependence structure of the data using an autocorrelation plot (ACF). In base graphics this is straightforward:

```
> acf(arctic$Extent,lag.max=365)
```

Comment on what you have seen. Would you expect this?

### 1.2.2 Extracting maxima/minima

To extract the annual minima first set up an empty vector in which to collect the minima, then use a `for` loop to do the actual extraction:

```
#Create a matrix with two columns: years and minima

> arcticMin <- matrix(0,nrow=2017-1977,ncol=2)

#Set up the years column

> arcticMin[,1] <- c(1978:2017)

#Extract the minima

> for(i in 1978:2017){
    arcticMin[i-1977,2] <- min(arctic$Extent[arctic$Year==i],na.rm=T) #extent
}

#Convert to data frame structure

> arcticMin <- as.data.frame(arcticMin)
> names(arcticMin) <- c("Year","Extent")
```

Again, we should plot the data to check that they look sensible:

```
> ggplot(data=arcticMin)+geom_point(aes(x=Year,y=Extent))
```

What do you notice about the first and last of the minima? Why is this? What would you do about it?

### 1.2.3 Fitting a GEV model

Following the previous findings, first remove the first and last annual minima:

```
> d <- nrows(arcticMin)
> arcticMin_arcticMin[-c(1,d),]
```

We also need to negate the minima as all the R functions that we will use are set up to work with maxima. Since

$$\min\{x_1, \ldots, x_n\} = -\max\{-x_1, \ldots, -x_n\}$$

we can model the negated minima as if they were maxima. The only thing that requires care is that return levels produced from the model will need to be negated to get them onto the 'original' measurement scale.

The `extRemes` package had a single function named `fevd` for fitting all types of univariate extreme value models, this includes the GEV for block maxima/minima and the GP for peaks over threshold data.

1. Since data are minima we need the GEV model. Using the `fevd` function, specifying the model type to be "GEV":

   ```
   > max.fit.1 <- fevd(Extent,data=arcticMin,type="GEV")
   > summary(max.fit.1)
   ```

   This provides maximum likelihood estimates of the parameters; what are they? What would 95% confidence intervals, based on the estimated standard errors, be?

2. What are your confidence intervals for the parameters?

3. Now we look at model fit using some visual diagnostics:

   ```
   plot(max.fit.1)
   ```

What plots are given automatically? How would you interpret each one?

4. Finally we can estimate any desired return levels using the `return.level` function, making sure to negate the results since we modelled the negative annual minima:

```
> -return.level(max.fit.1,c(5,50,100))
```

What are the estimated return levels?

5. To plot the return levels directly, you can specify `type=''rl"` in the `plot` function:

```
> plot(max.fit.1,type="rl",rperiods=seq(2,1000,by=1))
```

Note that this does not account for the need to negate return levels, and this must be done by hand using the `return.level` function.

```
R <- return.level(max.fit.1,seq(2,1000,by=1),do.ci=T)

max.rl <- matrix(0,nrow=999,ncol=4)
max.rl[,1] <- seq(2,1000,by=1)
max.rl[,2] <- -R[,1] #lower confidence interval
max.rl[,3] <- -R[,2] #point estimates
max.rl[,4] <- -R[,3] #upper confidence interval

max.rl[,1] <- -log(-log(1-1/max.rl[,1]))
max.rl <- as.data.frame(max.rl)
names(max.rl) <- c("ReturnPeriod","ReturnLevel","LowerCI","UpperCI")

ggplot(data=max.rl)+geom_line(aes(x=ReturnPeriod,y=ReturnLevel))
  +geom_line(aes(x=ReturnPeriod,y=LowerCI))
  +geom_line(aes(x=ReturnPeriod,y=UpperCI))
  +labs(x="Return Period",y="Return level (degrees C)")
  +scale_x_continuous(breaks=log(c(5,10,50,100,500,1000)),
        labels=c("5","10","50","100","500","1000"))
```

Comment on the return levels and their confidence intervals, especially for long return periods.

### 1.3 Fitting a GEV model with time as a covariate

Our exploratory analysis suggested that the minimum sea ice extent is declining over time. We can account for this in our GEV model. Ways to do this by putting a yearly trend in the location parameter of the GEV:

$$\mu(t) = \mu_0 + \mu_1 t, \quad t = 1, \ldots, 38 \tag{1}$$

and/or by doing the same to the scale parameter:

$$\log \sigma(t) = \sigma_0 + \sigma_1(t) \tag{2}$$

Remember that the $\log$ link-function is used to ensure that $\sigma(t)$ is always positive. Let's look at how to do this in R.

1. First scale the 'Year' covariate to be on the range $[1, 38]$ (this is jut to help the numerical optimiser used in the model fitting):

   ```
   > arcticMin$ScaledYear <- arcticMin$Year-1978
   ```

   Now fit model (1),

   ```
   > max.fit.2 <- fevd(Extent,data=arcticMin,type="GEV",location.fun=~1+ScaledYear)
   ```

   Use the `summary` function to obtain the parameter estimates.

2. The next step is to test whether or not the annual trend in the location parameter is significant using the likelihood ratio test,

   ```
   > lr.test(max.fit.1,max.fit.2)
   ```

   What do you conclude from this test?

3. Since an annual trend in the location parameter seems to be important, we keep this in whilst we test for an annual trend in the scale parameter as in model (2):

   ```
   > max.fit.3 <- fevd(Extent,data=arcticMin,type="GEV",location.fun=~1+ScaledYear,
         scale.fun=~1+ScaledYear)
   > lr.test(max.fit.2,max.fit.3)
   ```

   Is this term also significant? What would be your preferred model?

4. Do a model fit check on the best fitting model `max.fit.2` using the `plot` function,

   ```
   > plot(max.fit.2)
   ```

5. Using the best fitting model we estimate the ‘effective return levels’. Because we are modelling negated minima, a little bit of code is required to put the return levels back on the original scale.

   ```
   #First estimate the return levels
   #Gives 3 by 38 matrix (3 return periods, 38 data points)

   > rl.2 <- erlevd(max.fit.2,period=c(2,20,100))
   > dim(rl.2)

   #Negate return levels

   > rl.2.neg <- -rl.2


   #Create data frame for ggplot2 of return level against year

   > rl.2.neg <- -rl.2
   > period <- c(rep(2,38),rep(20,38),rep(100,38))
   > rl.final <- cbind(rep(arcticMin$Year,3),as.vector(t(rl.2.neg)),period)
   > rl.final <- as.data.frame(rl.final)
   > rl.final[,3] <- as.factor(rl.final[,3])
   > levels(rl.final[,3]) <- c("2yr","20yr","100yr")
   > names(rl.final) <- c("Year","RL","RP")

   #Plot
   > ggplot()+geom_line(data=rl.final,aes(x=Year,y=RL,color=RP))
        +geom_line(aes(x=Year,y=-Extent),data=arcticMin)
   ```

   Comment on your return levels. Do they seem sensible?

## 1.4 Challenge

Can you fit a similar model to the annual maximum sea ice extent for the Arctic data? What about the annual minima and maxima for the Antarctic. Remember, when modelling the maxima you do not negate the data before modelling. This means that you do not need to worry about negating the return levels produced by the model either.