Extreme Value Analysis Workshop II: Peaks over Thresholds

Tuesday 5th September, 2017

1.1 Modelling PoT of Greenland temperatures

Temperature observations from the Summit station on Greenland have been obtained from the Greenland Climate Network:

`http://cires1.colorado.edu/steffen/gcnet/`

Daily maxima have been extracted and the spreadsheet can be found in the same github repository as before.

1.1.1 Exploratory Data Analysis

First download the maximum daily temperature data. Load these into R, e.g.

```
> summit <- read.csv("summit_daily_max.csv")
```

As before it is useful to start with some exploratory analysis of the data.

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

   ```
   > names(summit)
   ```

2. How many observations have you got? Try

   ```
   > nrow(summit)
   ```

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

   ```
   > plot(summit$Date,summit$Temperature,xlab="Date",ylab="Temperature")
   ```

   Using `ggplot2`:

   ```
   > ggplot(data=summit)+geom_point(aes(x=Date,y=Temperature))
   ```

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

4. Finally we will check the dependence structure of the data using the autocorrelation plot (ACF) and the partial autocorrelation plot (PACF).

   (a) In base graphics the ACF is straightforward:

   ```
   > acf(summit$Temperature,lag.max=365,na.action=na.pass)
   ```

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

   (b) Try looking at the partial autocorrelation function instead:

   ```
   pacf(summit$Temperature,lag.max=365,na.action=na.pass)
   ```

   You can zoom in on the interesting part by changing the value of `lag.max`; try `lag.max=31`. What do you see?

1.1.2 Extracting PoT events

We start by assuming that the extreme events, defined as threshold exceedances, are independent. Later on we will look at extracting clusters of extreme events. You can manually extract threshold exceedances as follows:

```
#First remove all missing observations

> summitNoNa <- na.omit(summit)

#Next define a threshold e.g., a high quantile of the data

> u <- quantile(summitNoNa$Temperature,0.9)

#Finally extract exceedances and associated dates

> summitEx <- summitNoNa[summitNoNa$Temperature>u,]
```

Again, we should plot the data to check that they look sensible. In `base` graphics:

```
> plot(summitEx$Date,summitEx$Temperature)
```

or in `ggplot2`,

```
> ggplot(data=summitEx)+geom_point(aes(x=Date,y=Temperature))
```

1.1.3 Fitting a GP model

Recall that the `extRemes` package had a single function named `fevd` for fitting all types of univariate extreme value models, including the GP for peaks over threshold data.

1. Using the `fevd` function, specifying the model type to be `"GP"`:

```
> pot.fit.1 <- fevd(Temperature,data=summitNoNa,threshold=u,type="GP")
> summary(pot.fit.1)
```

This provides maximum likelihood estimates of the parameters; what are they?

2. What are the 95% confidence intervals for the parameters?

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

```
plot(pot.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:

```
> return.level(pot.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(pot.fit.1,type="rl",rperiods=seq(2,1000,by=1))
```

Note that the points are the empirical estimates of the return levels. Comment on the return levels and their confidence intervals, especially for long return periods.

1.2 Clusters of extreme events

Our preliminary GP model fit did not show a good fit to the highest temperatures. This may be because of (i) dependence in the extremes and/or (ii) the effects of un-identified covariates. We will investigate (i) by declustering the extreme events using the runs method. Recall that this needs us to specify the threshold (we can use the same one as used in the previous part of the analysis) and the run length. In this example we will use a run length of 2 days:

```
> summitDecl <-
    decluster(summitNoNa$Temperature,threshold=u,data=summitNoNa,method="runs",r=3)
```

1. Look at a summary of the cluster identification algorithm using the `print` function,

   ```
   > print(summitDecl)
   ```

   How many clusters were there?

2. To print the clusters try

   ```
   plot(summitDecl,col="orange")
   ```

3. Finally for and estimate of the extremal index:

   ```
   > extremalindex(summitNoNa$Temperature,threshold=u,method="runs",run.length=3)
   ```

   What is the estimated extremal index? Use this value to give the average cluster size. Is there strong extremal dependence amongst the temperatures?

4. By changing the (i) threshold and (ii) run length, assess the sensitivity to the choice of both of these variables. What do you notice? Is this what you would expect?

Next we fit a GP to the *cluster maxima*.

1. Use the `fevd` function, but this time apply it to the output from the `decluster` function:

   ```
   > pot.fit.2 <- fevd(x=summitDecl,threshold=u,type="GP")
   ```

2. To look at the results of the model fit

   ```
   > print(pot.fit.2)
   ```

What are the parameter estimates, their confidence intervals and the log-likelihood for the model?

3. Use the same graphical diagnostics as before to assess the fit of the model,

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

   Comment on what you see.

4. For this model fit, obtain estimates of 5, 50 and 100 year return levels using the `return.level` function. Are these much different to the return levels obtained when assuming independent extremes?

1.3 Challenges

1. How would you go about investigating whether or not there was a time trend in the peaks over threshold data that we have just looked at?
2. Can you carry out a similar analysis to the above for the *lowest* temperatures at the Summit site?