STAT0017 ICA 1 2018-19

Student number: 17052480

2019-03-22

Extremal Types Example

[25]

Question 1 (a)

From the question we can see:

$$F(x) = P(X \leq x) = \begin{cases} 1 - e^{1/x}, & \text{for } x < 0 \ . \\ 1, & \text{for } x \geq 0 \ . \end{cases} \tag{*}$$

$$M_n = max\{x_1, ..., x_n\}$$

Hazard Function:

$$h(x) = \frac{1 - F(x)}{f(x)} \tag{1}$$
$$= \frac{e^{1/x}}{e^{1/x} \times (-x^{-2})} \tag{2}$$
$$= -x^2 \tag{3}$$

Derivative of h(x): $h^{'} = -2x$

$$\lim_{x \to x^F} h^{'}(x) \to \xi \tag{4}$$
$$Let \ \xi = -2M_n \tag{5}$$
$$We \ can \ see: \ x \sim GEV(0, 1, -2M_n) \tag{6}$$

From the slides we have:

$$\begin{cases} 1 - F(b_n) = 1/n \\ a_n = h(b_n) \end{cases}$$

solve it we have:

$$\begin{cases} b_n = -\frac{1}{log(n)} \\ a_n = -\frac{1}{log^2(n)} \end{cases}$$

Question 1 (b)

When $\xi < 0$ the GEV distribution has light upper tail with the finite upper limit which is $\mu - \sigma/\xi$.

Let $\mu - \sigma/\xi = 0$ then this distribution with a finite upper end point of 0 has an upper end point of infinity.

Question 1 (c)

We assume $A = \{x : 0 < F(x) < 1\}$, and $x^* = sup_{x \in A} A$.

Here F(x) can be any function including the function ($*$).

For $\forall x, x < x^*$, we have $Pr(M_n \leq x) = F^n(x) \to 0, as\ n \to \infty$.

For $\forall x, x \geq x^*$, we have $Pr(M_n \leq x) = F^n(x) \to 1, as\ n \to \infty$.

This is mean whatever x or F(x), $M_n = 0 or 1$ when $n \to \infty$.

The M_n is Degenerate distribution, which is useless.

That is why we need fit a GEV model.

Question 1 (d)

The log-likelihood:

$$\xi_n \approx h^{'}(x)|_x = u(n)\ where\ u(n) = F^{-1}(1 - 1/n)$$

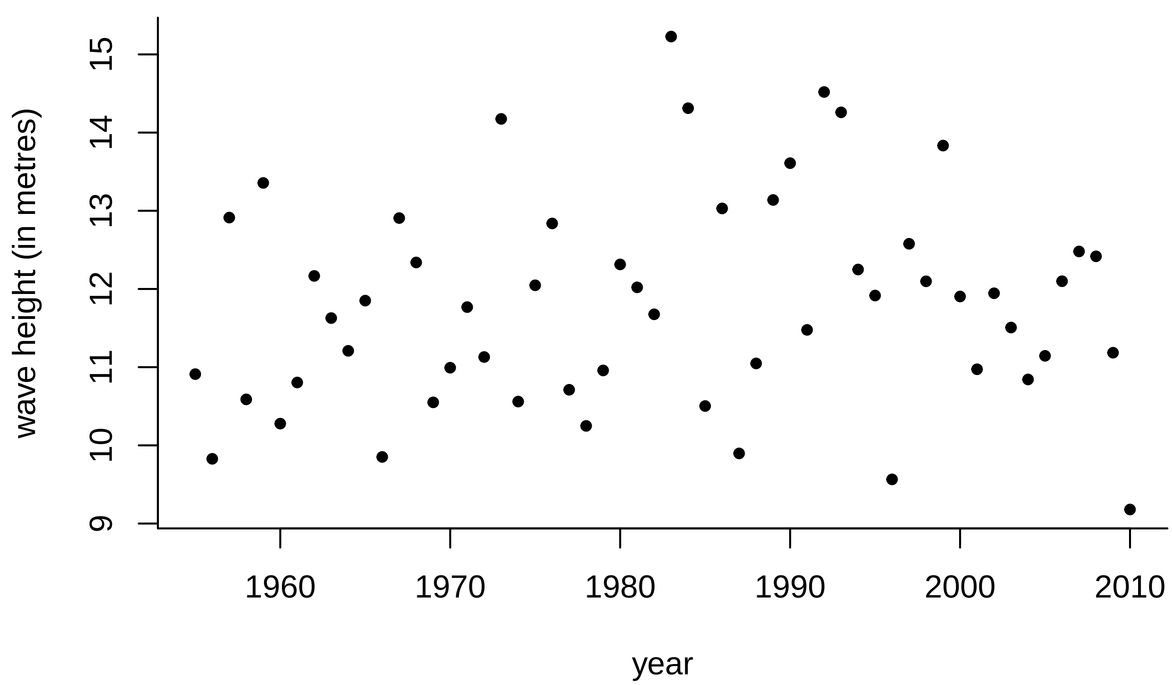
Thus

$$\xi = h^{'}(x) = -2/ln(365) = -0.3400$$

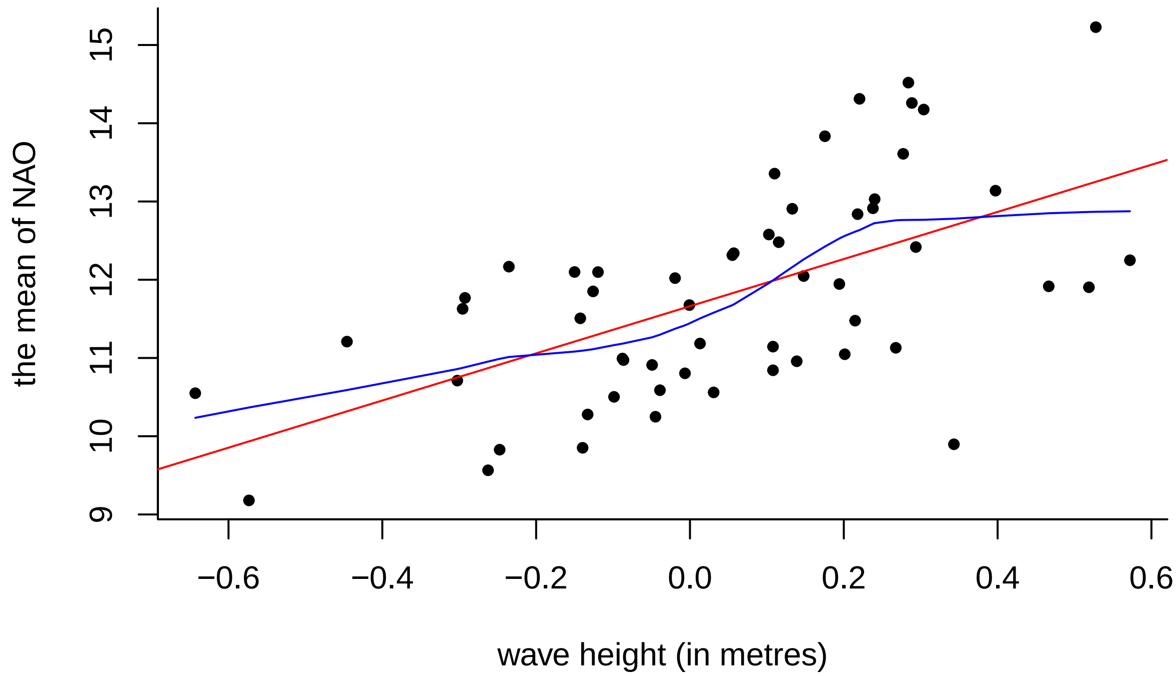
Exploratory analysis

Winter maxima (wm)

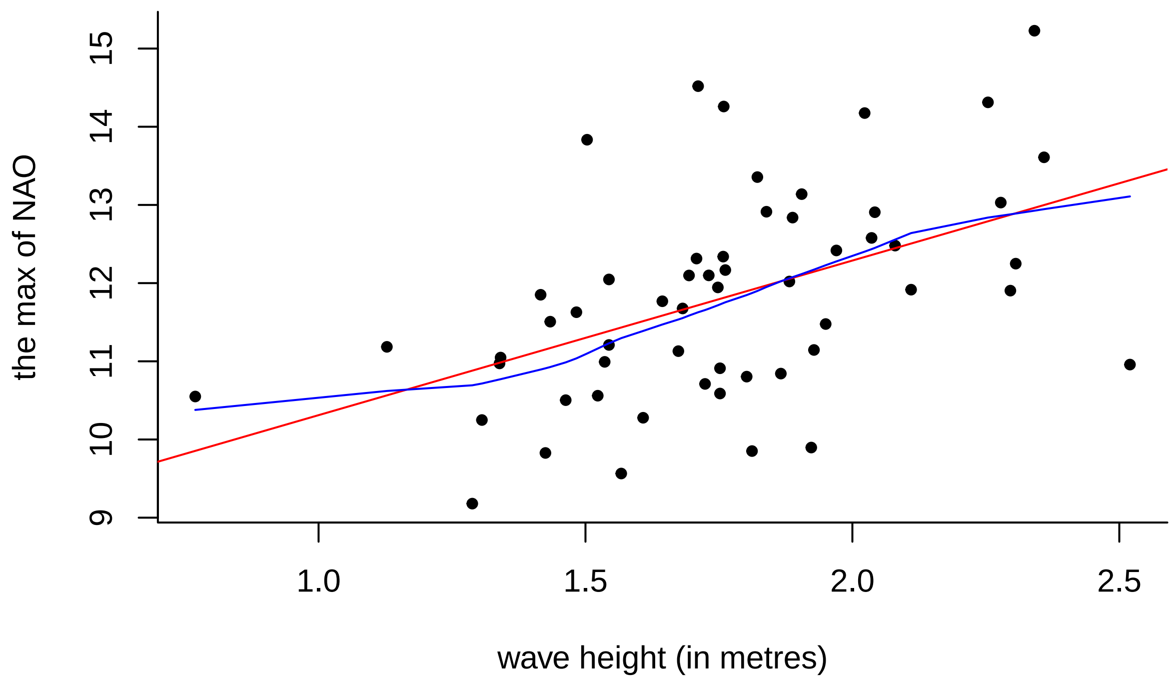
```
# Add R code here (and similarly elsewhere)
plot(wm$waterYear,wm$Hs,bty="l",pch=20,xlab="year",ylab="wave height (in metres)")
```



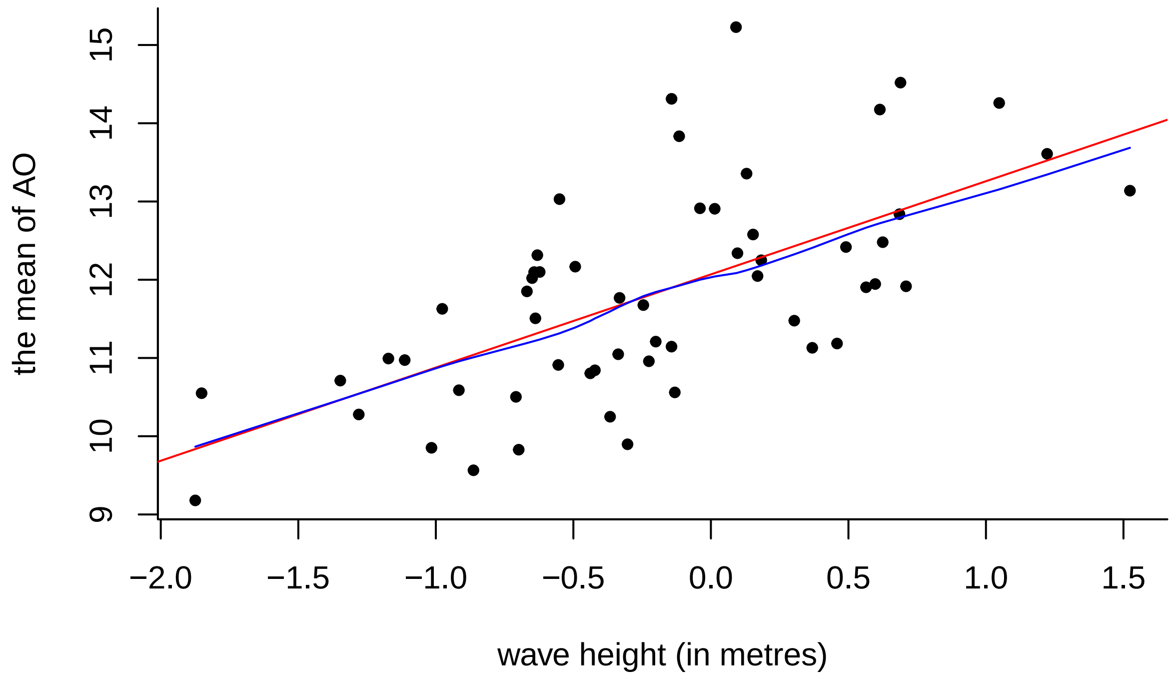
```
#(wm$Hs~wm$meanNAO)
plot(wm$meanNAO,wm$Hs,bty="l",pch=20,xlab="wave height (in metres)",ylab="the mean of NAO")
abline(lm(wm$Hs~wm$meanNAO), col="red") # regression line (wm$Hs~wm$meanNAO)
lines(lowess(wm$Hs~wm$meanNAO), col="blue") # lowess line (wm$Hs~wm$meanNAO)
```



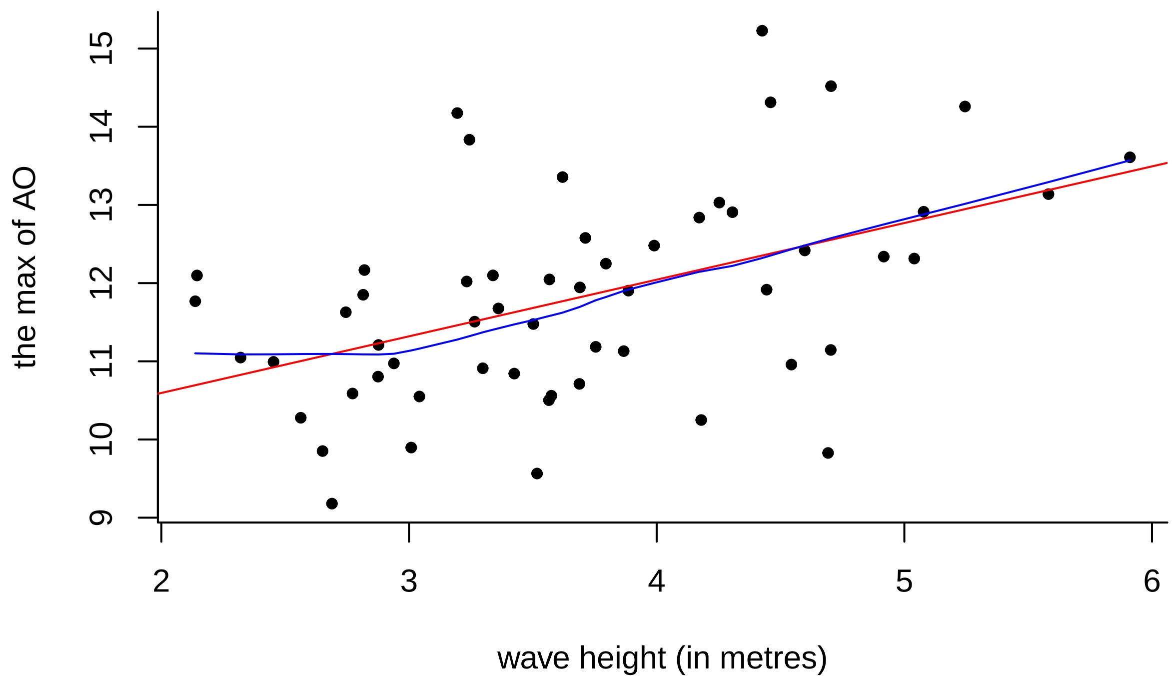

```
#(wm$Hs~wm$maxNAO)
plot(wm$maxNAO,wm$Hs,bty="l",pch=20,xlab="wave height (in metres)",ylab="the max of NAO")
abline(lm(wm$Hs~wm$maxNAO), col="red") # regression line (wm$Hs~wm$maxNAO)
lines(lowess(wm$Hs~wm$maxNAO), col="blue") # lowess line (wm$Hs~wm$maxNAO)
```



```
#(wm$Hs~wm$meanAO)
plot(wm$meanAO,wm$Hs,bty="l",pch=20,xlab="wave height (in metres)",ylab="the mean of AO")
abline(lm(wm$Hs~wm$meanAO), col="red") # regression line (wm$Hs~wm$meanAO)
lines(lowess(wm$Hs~wm$meanAO), col="blue") # lowess line (wm$Hs~wm$meanAO)
```

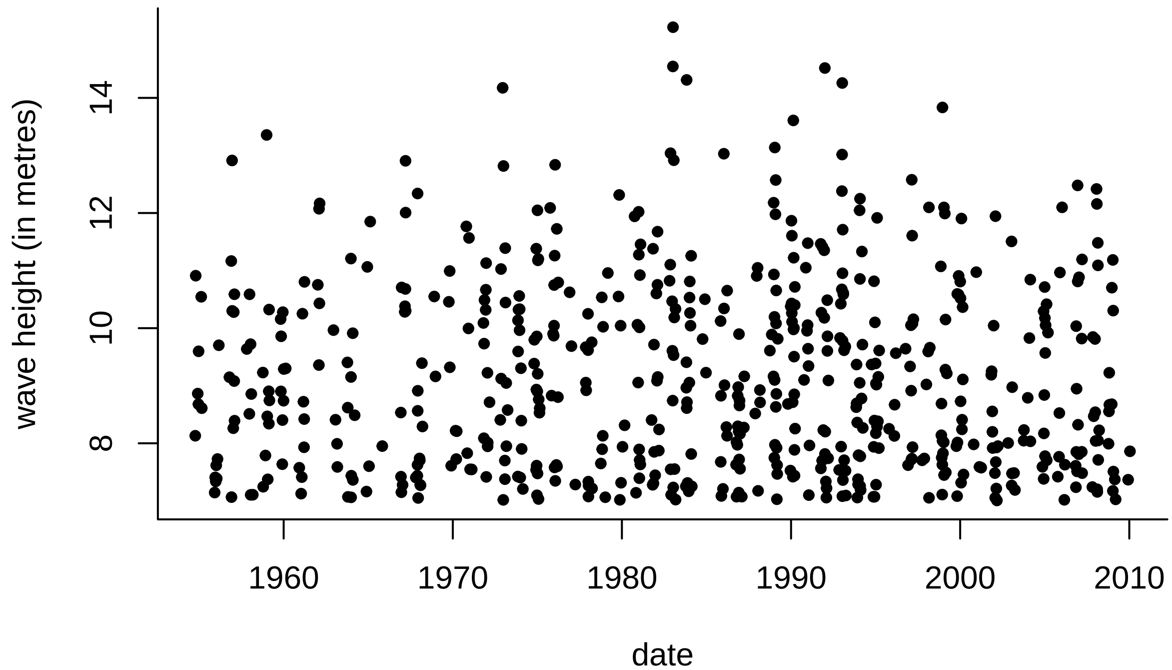



```
#(wm$Hs~wm$maxAO)
plot(wm$maxAO,wm$Hs,bty="l",pch=20,xlab="wave height (in metres)",ylab="the max of AO")
abline(lm(wm$Hs~wm$maxAO), col="red") # regression line (wm$Hs~wm$maxAO)
lines(lowess(wm$Hs~wm$maxAO), col="blue") # lowess line (wm$Hs~wm$maxAO)
```



Storm peaks (pot)

```
plot(pot$date,pot$Hs,bty="l",pch=20,xlab="date",ylab="wave height (in metres)")
```



Comments

The relationship between wave height and all of variables can be roughly seen through the scatter plot. As can be seen from the line graph, the more the red line coincides with the blue line, the better the linear regression model fits. As can be seen from the plot, the average of AO has a better fitting relationship with the height of the waves. Further analysis needs to be carried out, followed by analysis. [10]

Extreme value (EV) modelling of H_s

GEV modelling of winter maxima

```
wm.gev <- gev.fit(wm[,1])#fit GEV model
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 94.17556
##
## $mle
## [1] 11.2726903  1.2064985 -0.1534719
##
## $se
## [1] 0.18140907 0.12880447 0.09956538
```

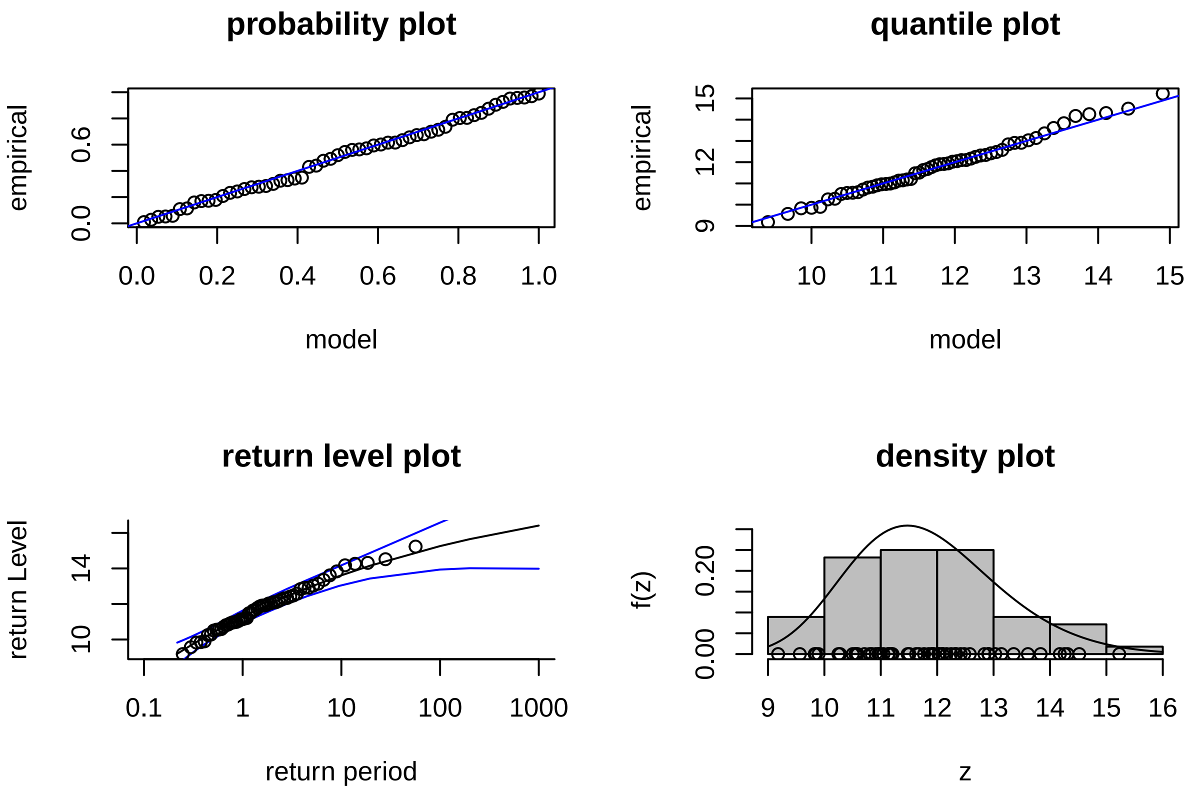
```r
wm.gev$mle#GEV MLEs
```

```
## [1] 11.2726903  1.2064985 -0.1534719
```

```r
wm.gev$se#standard errors of MLEs
```

```
## [1] 0.18140907 0.12880447 0.09956538
```

```r
pjn.gev.diag(wm.gev) #PJN's GEV diagnostics
```



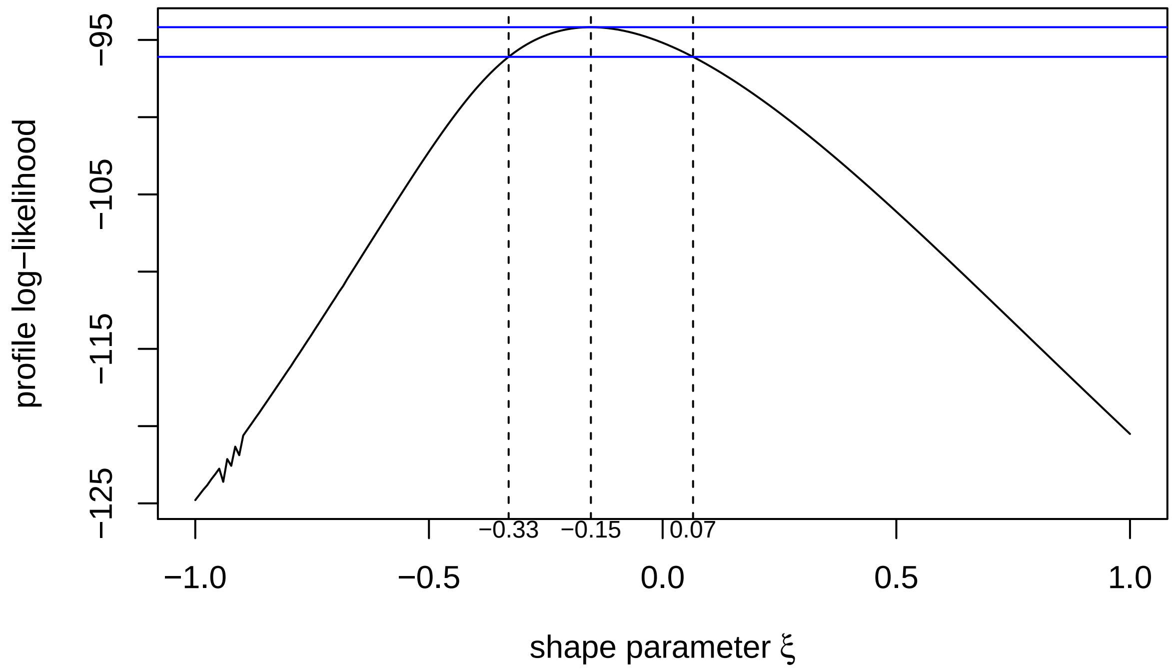
Maximum Likelihood-Based Inference

```r
pjn.gev.conf(wm.gev, conf = 0.95)
```

```
## $low.lim
## [1] 10.9171351  0.9540464 -0.3486164
##
## $up.lim
## [1] 11.62824559  1.45895067  0.04167269
```

```r
pjn.gev.profxi(wm.gev, -1, 1, conf = 0.95)
```

```
## If routine fails, try changing plotting interval
```



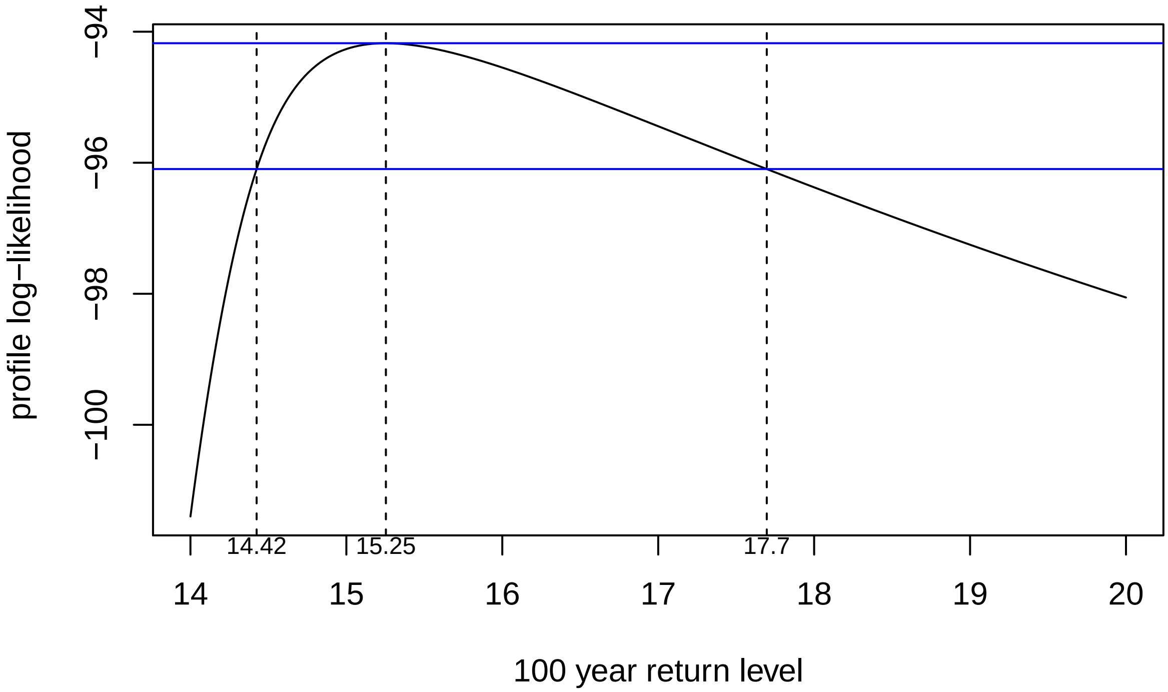

```
## $low.lim
## [1] -0.3294846
##
## $xi.mle
## [1] -0.1534719
##
## $up.lim
## [1] 0.06508564
```

```
pjn.gev.conf.ret.levels(wm.gev, m = 100, conf = 0.95)
```

```
## $low.lim
##          [,1]
## [1,] 13.93437
##
## $zp.mle
## [1] 15.25355
##
## $up.lim
##          [,1]
## [1,] 16.57272
```

```
# Profile log-likelihood for the 100 year return level
pjn.gev.prof(wm.gev,m=100,14,20)# PJN's version
```

```
## If routine fails, try changing plotting interval
```




```
## $low.lim
## [1] 14.42462
##
## $xp.mle
## [1] 15.25355
##
## $up.lim
## [1] 17.69618
```

Comments

First, the model is fitted, and the three parameters of the model are 11.2726903, 1.2064985, -0.1534719 respectively. Their variances are 0.18140907, 0.12880447, 0.09956538 respectively. Through the diagnosis, we can see that the tail of data is basically appropriate, and it can be seen the model's trend.

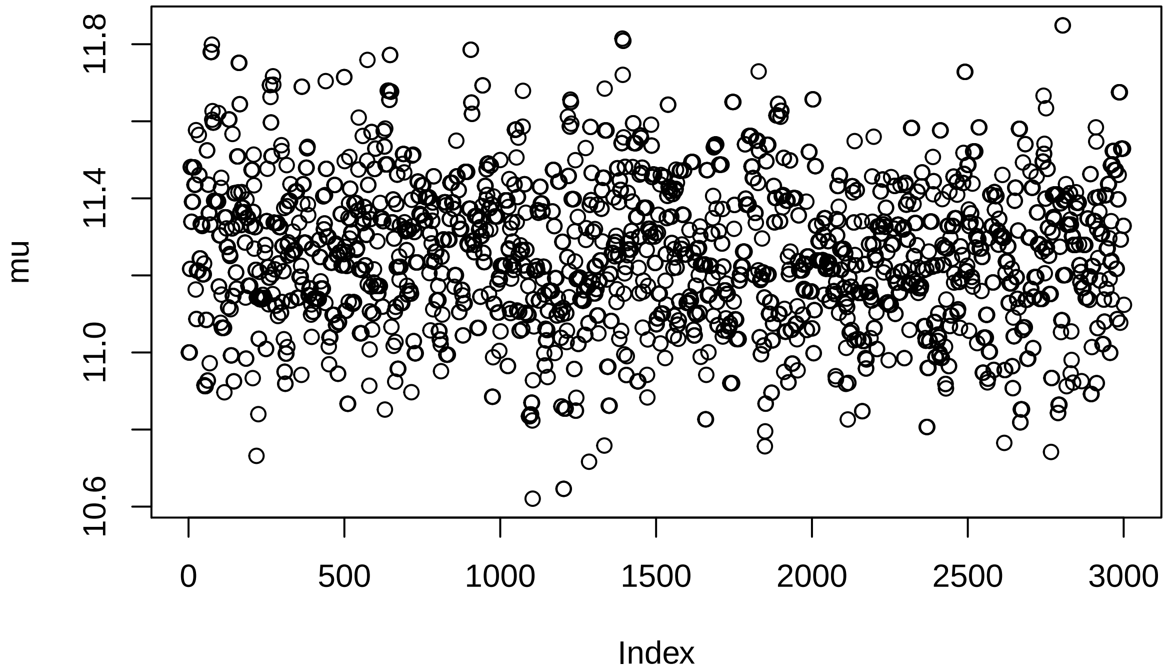
Bayesian Inference

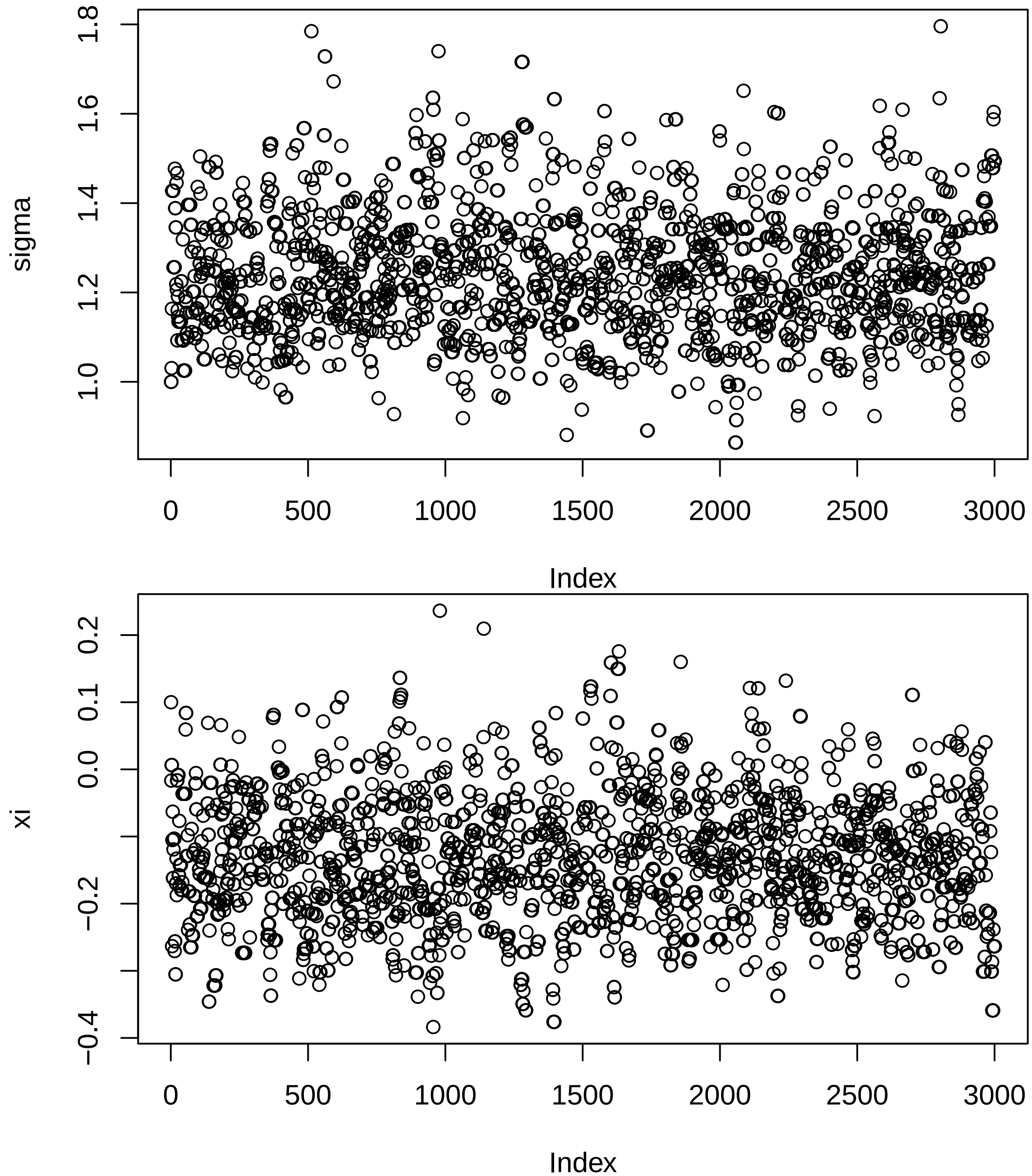
```
my.ylab <- c("mu","sigma","xi")
# Set independent normal priors with large variances
prior.cov.mat <- diag(c(10000, 10000, 100))
pn <- prior.norm(mean = c(0, 0, 0), cov = prior.cov.mat)

# Starting well away from the MLEs
init <- c(11,1,0.1)
# SDs of the proposal distributions for (mu, ln(sigma), xi)
prop.sd <- c(0.18,0.12,0.1)
prop.sd.auto <- ar.choice(init = init, prior = pn, lh = "gev",
                          data = wm$Hs, psd = prop.sd,
                          tol = rep(0.02, 3))$psd
```

```
## Accept Rate values and proposal standard deviations at each iterations...
## Accept Rate   Prop. Std
## 0.68 0.63 0.65    0.18 0.12 0.1
## 0.34 0.31 0.3     0.54 0.36 0.3
## 0.57 0.5 0.56     0.27 0.18 0.15
## 0.56 0.54 0.54    0.263 0.165 0.145
## 0.51 0.5 0.47     0.311 0.184 0.167
## 0.46 0.44 0.42    0.385 0.218 0.195
## 0.41 0.41 0.4     0.456 0.248 0.195
```

```
post <- posterior(3000, init = init, prior = pn, lh = "gev",
                  data = wm$Hs, psd = prop.sd.auto)
for (i in 1:3) plot(post[,i],ylab=my.ylab[i])
```

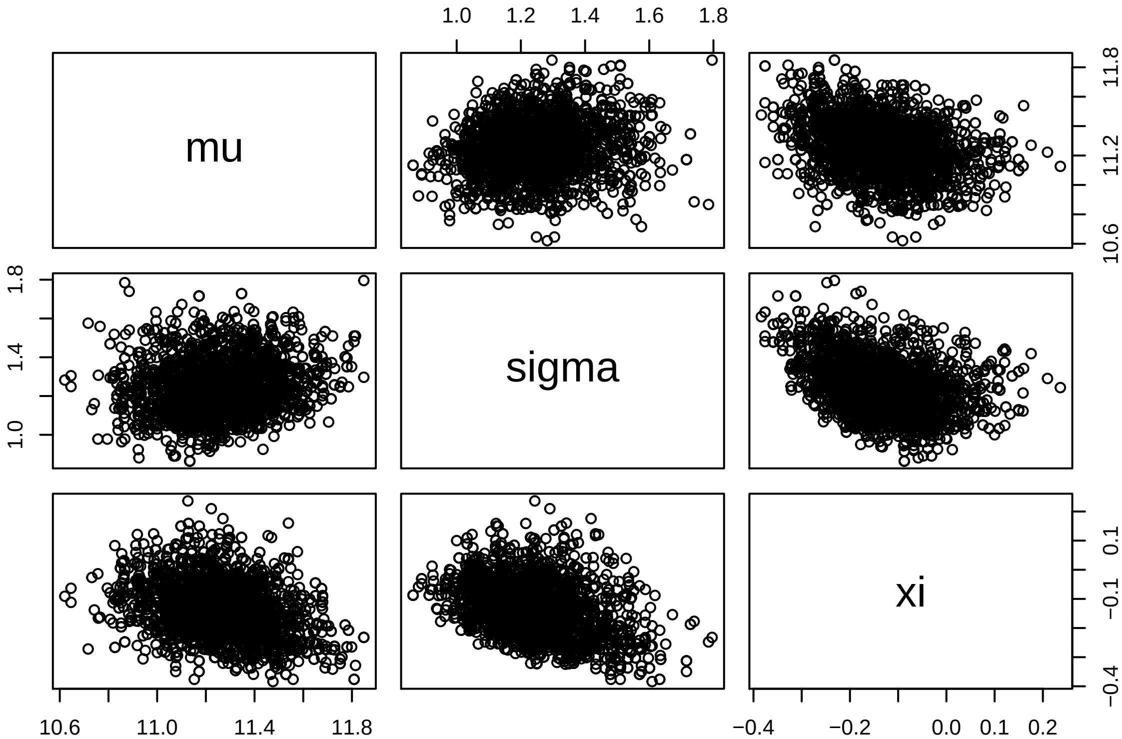





```r
attr(post,"ar")
```

```
##             mu sigma   xi total
## acc.rates 0.39  0.42 0.43  0.41
## ext.rates 0.01  0.03 0.20  0.08
```

```r
# Posterior dependence among parameters
pairs(post)
```



Comments

Bayesian Inference

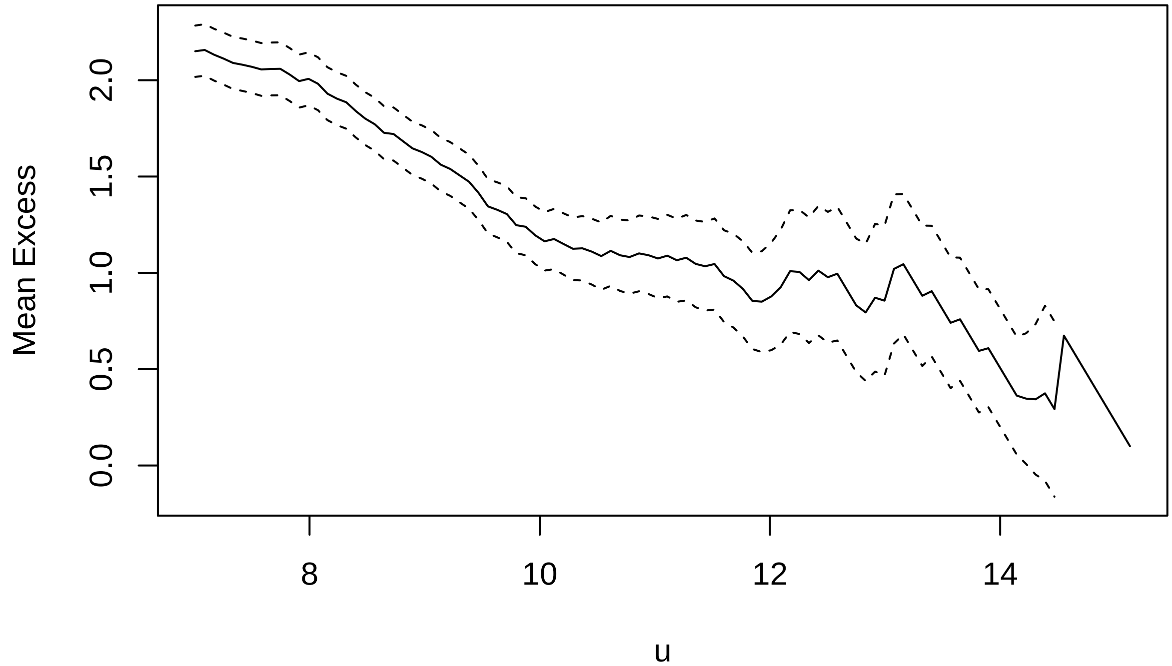
Binomial-GP modelling of storm peaks

```
pot.gp <- gpd.fit(pot[,1], 10)
```

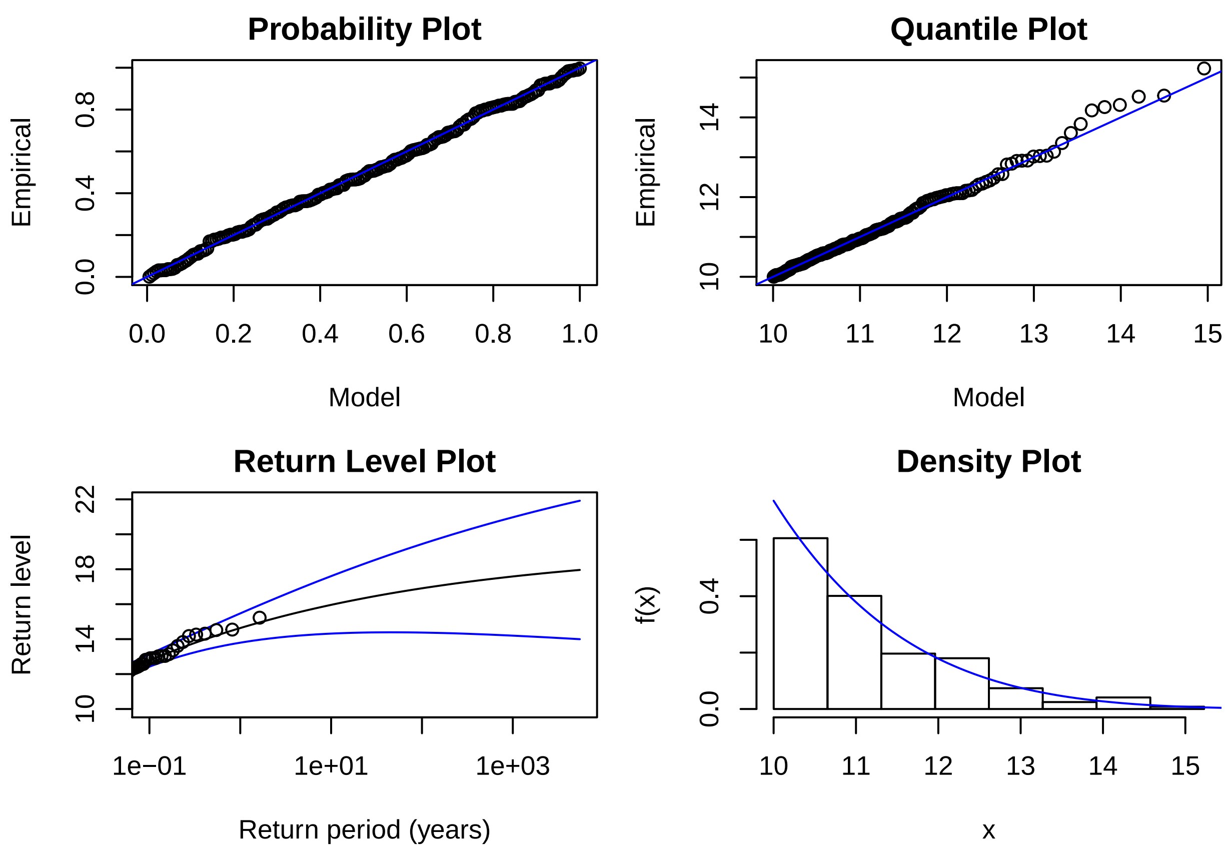
```
## $threshold
## [1] 10
##
## $nexc
## [1] 187
##
## $conv
## [1] 0
##
## $nllh
## [1] 216.3687
##
## $mle
## [1]  1.3539303 -0.1458176
##
## $rate
## [1] 0.3142857
##
## $se
## [1] 0.1366724 0.0703894
```

Threshold selection

```r
mrl.plot(pot[,1])
```




```r
# PJN's GP diagnostics
pjn.gpd.diag(pot.gp)
```



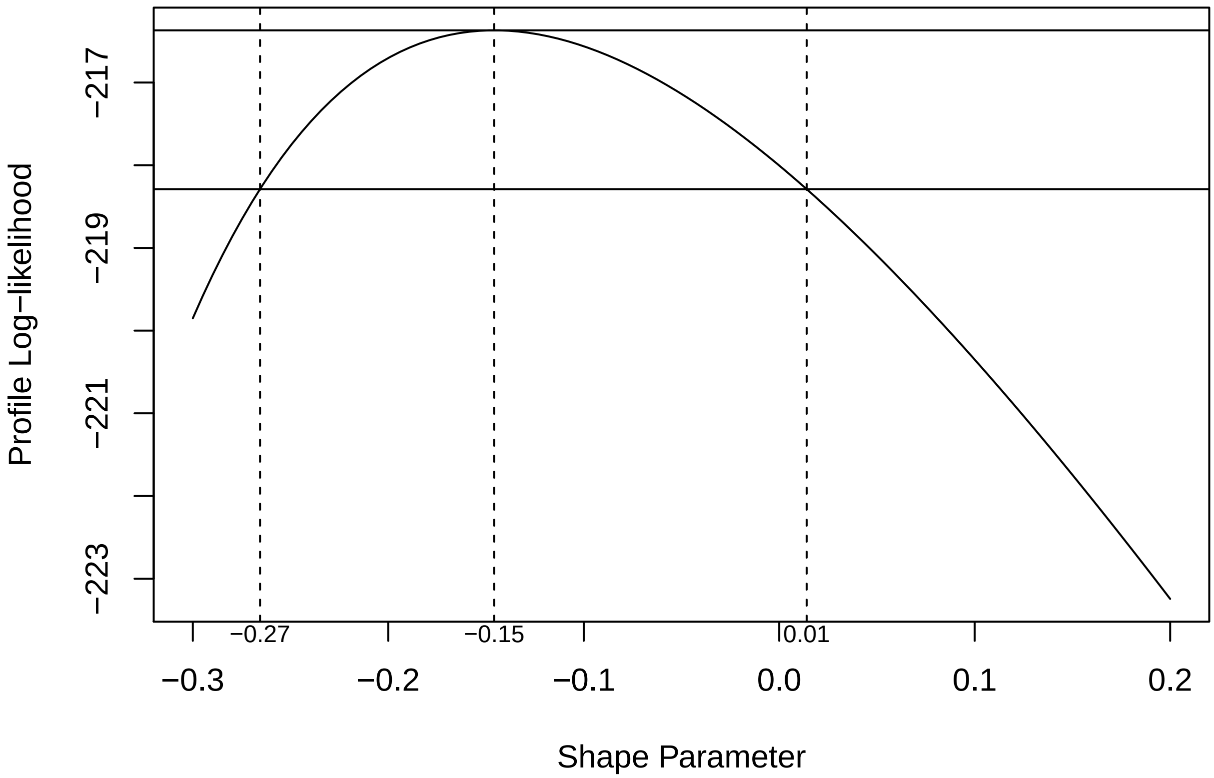
Maximum Likelihood-Based Inference

```
#Symmetric 100*conf% confidence intervals for p_u, sigma_u and xi
pjn.gpd.conf(pot.gp, conf = 0.95)
```

```
## $low.lim
##         pu     sigmau         xi
##  0.2477490  1.0860573 -0.2837783
##
## $up.lim
##           pu       sigmau           xi
##  0.380822407  1.621803250 -0.007856923
```

```
#PJN's version of ismev::gpd.profxi
pjn.gpd.profxi(pot.gp, xlow = -0.3, xup = 0.2, conf = 0.95)
```

```
## If routine fails, try changing plotting interval
```



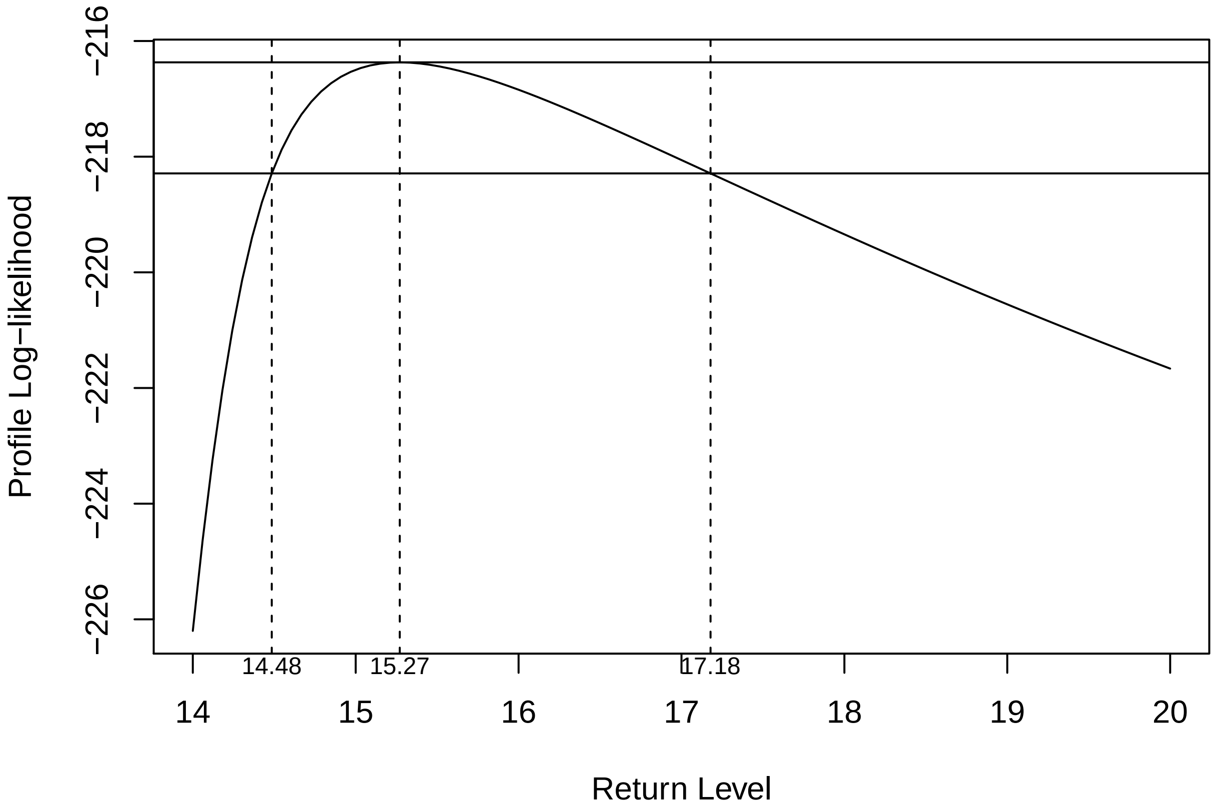

```
## $low.lim
## [1] -0.2656067
##
## $xi
## [1] -0.1458176
##
## $up.lim
## [1] 0.01404194
```

```
#Symmetric 100*conf% confidence intervals for the 100-year return level
pjn.gpd.conf.ret.levels(pot.gp, m = 100, npy = 10, conf = 0.95)
```

```
## $low.lim
##          [,1]
## [1,] 14.10175
##
## $xm.mle
## [1] 15.27059
##
## $up.lim
##          [,1]
## [1,] 16.43943
```

```
#Profile log-likelihood for the 100-year return level
pjn.gpd.prof(pot.gp, m = 100, xlow = 14, xup = 20, npy = 10, conf = 0.95)
```

```
## If routine fails, try changing plotting interval
```



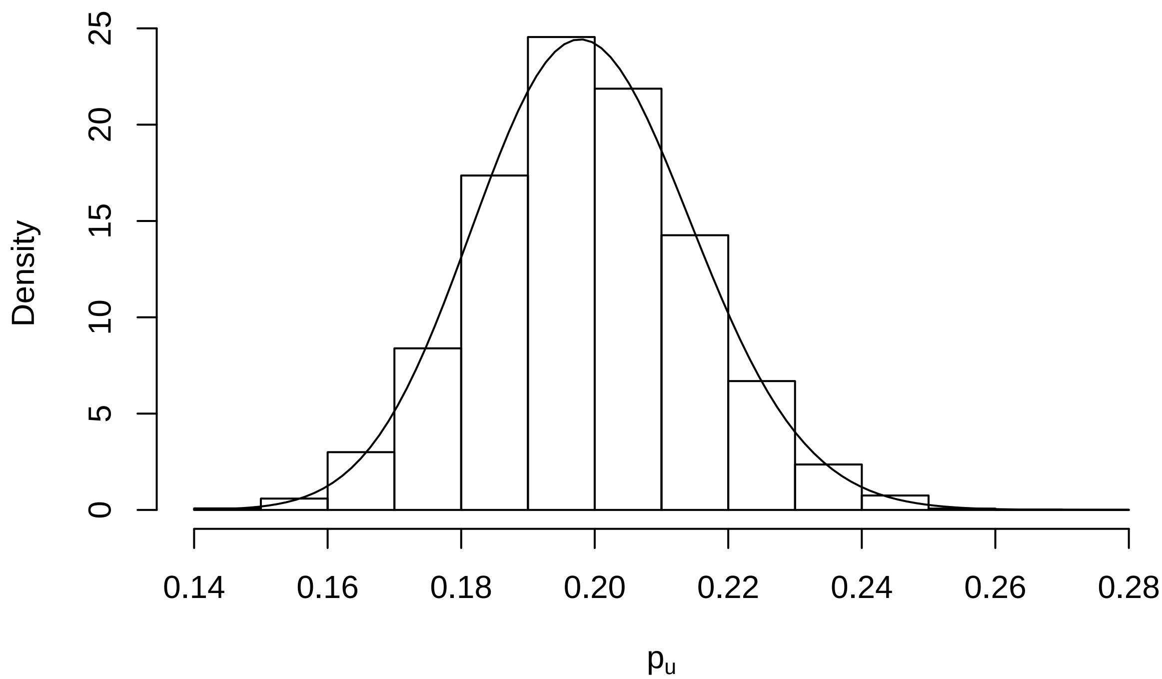

```
## $low.lim
## [1] 14.48478
##
## $xm.mle
## [1] 15.27059
##
## $up.lim
## [1] 17.17829
```

Comments

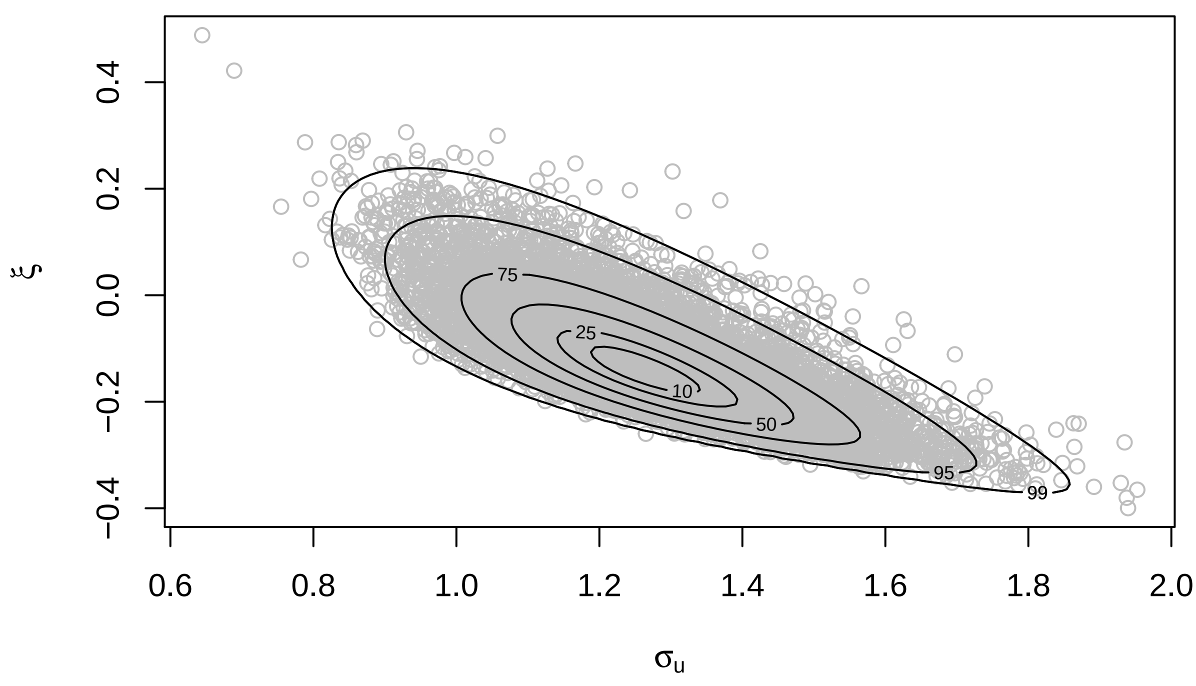
The estimated of ξ is -0.15, and the corresponding 95% confidence interval is [-0.33,0.07].

Bayesian Inference

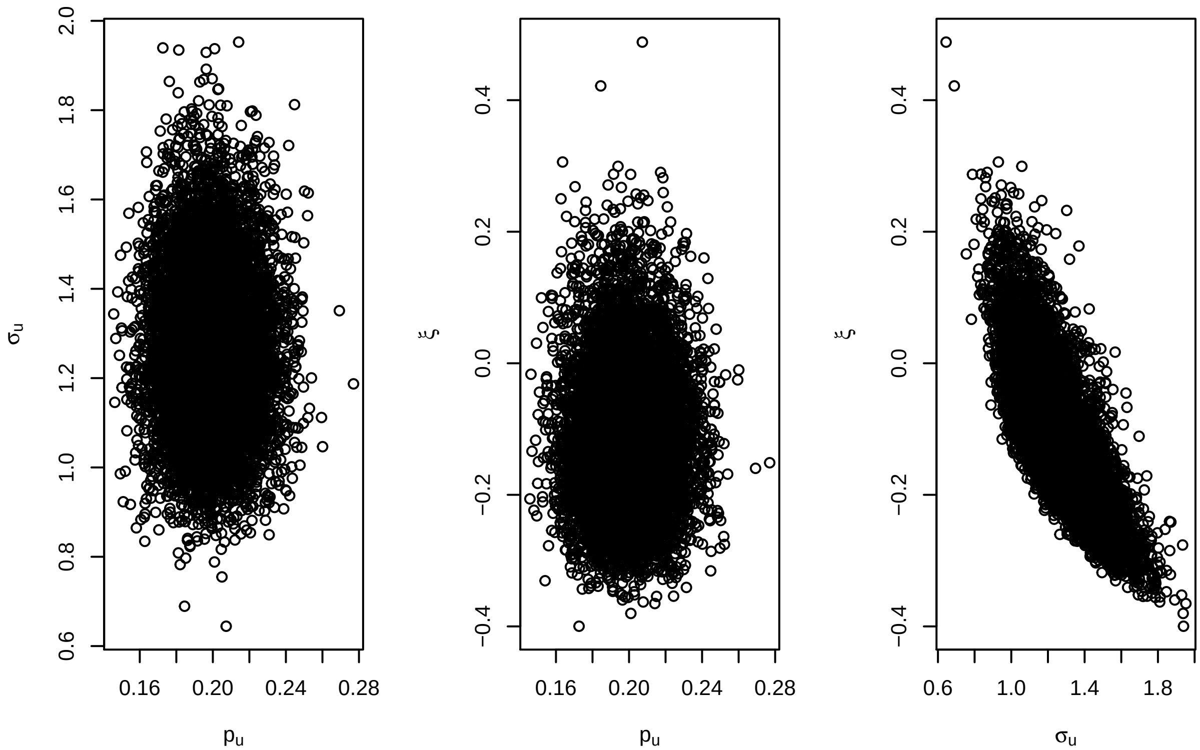
```
fp <- set_prior(prior = "flat", model = "gp", min_xi = -1)
bp <- set_bin_prior(prior = "jeffreys")
u <- quantile(pot[,1], probs = 0.8)
bgpg <- rpost(n = 10000, model = "bingp", prior = fp, thresh = u, data = pot[,1],
              bin_prior = bp, npy = 10)
plot(bgpg, pu_only = TRUE)
```




```r
plot(bgpg, add_pu = FALSE)
```




```r
plot(bgpg, add_pu = TRUE)
```



Comments

It can also be seen that the likelihood function of the 100 year return level. It takes a constant attempt to return the plot to find th likelihood function is asymmetrical, the confidence interval is also asymmetrical with respect to the maximum likelihood estimate. [25]

Reporting to your client

From the plot we can see that the return period is 100. The estimated level of reconstruction for the year is 15.25, and the corresponding 95% confidence interval is [14.42,17.7].

[15]

EV regression modelling of winter maximum H_s on NAO

Build a GEV regression model

[15]

```
evd_fit0 <- fgev(wm[,1])
covaryear <- data.frame(scaled_year = (wm[,2]-1955)/(2010-1955))
covar2 <- data.frame(scaled_year = (wm[,2]-1955)/(2010-1955), NOA = wm[, 3])
evd_fit1 <- fgev(wm[,1], nsloc = covaryear)
evd_fit2 <- fgev(wm[,1], nsloc = covar2)

anova(evd_fit2, evd_fit1, evd_fit0)
```

```
## Analysis of Deviance Table
##
##          M.Df Deviance Df   Chisq Pr(>chisq)
## evd_fit2    5   165.01
## evd_fit1    4   188.00  1 22.9924  1.626e-06 ***
## evd_fit0    3   188.35  1  0.3516     0.5532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
fitted(evd_fit2) # strictly speaking this should be coef()
```

```
##            loc locscaled_year         locNOA          scale          shape
##     11.4528656     -0.2301308      3.6792878      1.1570164     -0.4557251
```

```r
std.errors(evd_fit2)
```

```
##            loc locscaled_year         locNOA          scale          shape
##      0.3028233      0.5141350      0.7878129      0.1657038      0.1852270
```

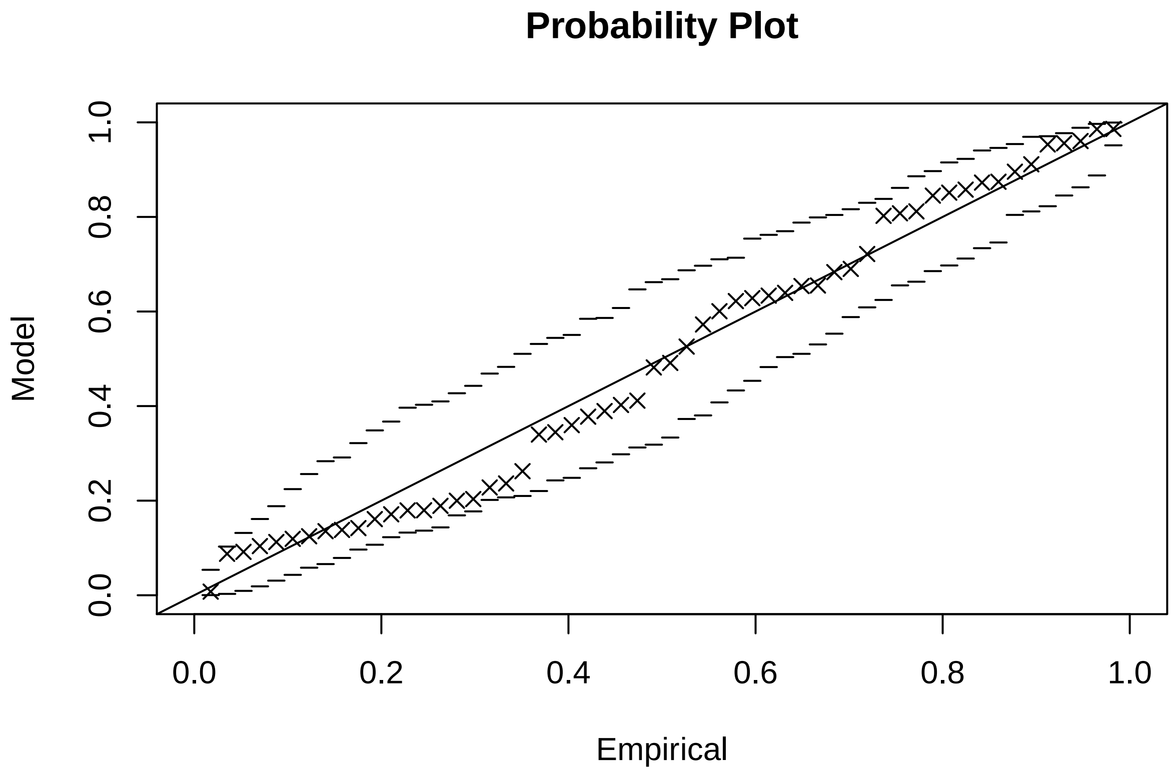
```r
vcov(evd_fit2)
```

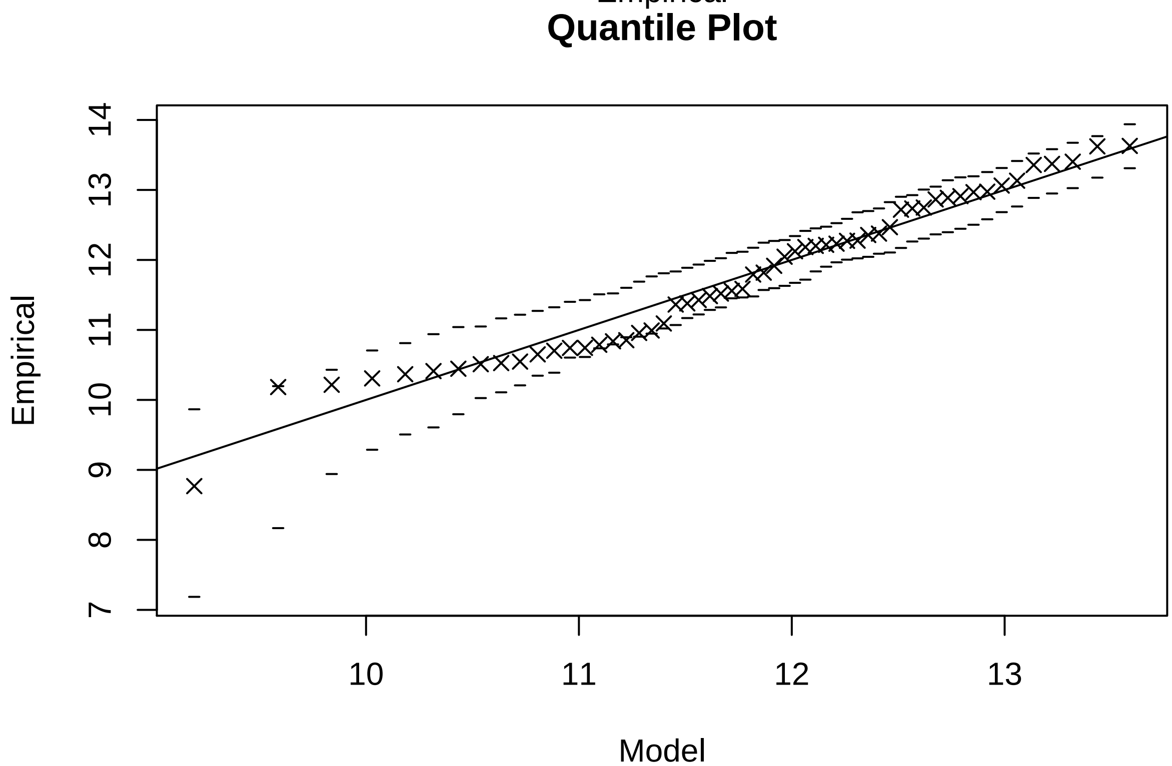
```
##                         [,1]         [,2]        [,3]          [,4]
## loc             0.0917019264 -0.126467342  0.06213421  0.0001108082
## locscaled_year -0.1264673420  0.264334832 -0.07455634  0.0013291718
## locNOA          0.0621342061 -0.074556339  0.62064916  0.0731847564
## scale           0.0001108082  0.001329172  0.07318476  0.0274577420
## shape          -0.0119302511 -0.003519922 -0.11231086 -0.0245374017
##                        [,5]
## loc            -0.011930251
## locscaled_year -0.003519922
## locNOA         -0.112310864
## scale          -0.024537402
## shape           0.034309052
```

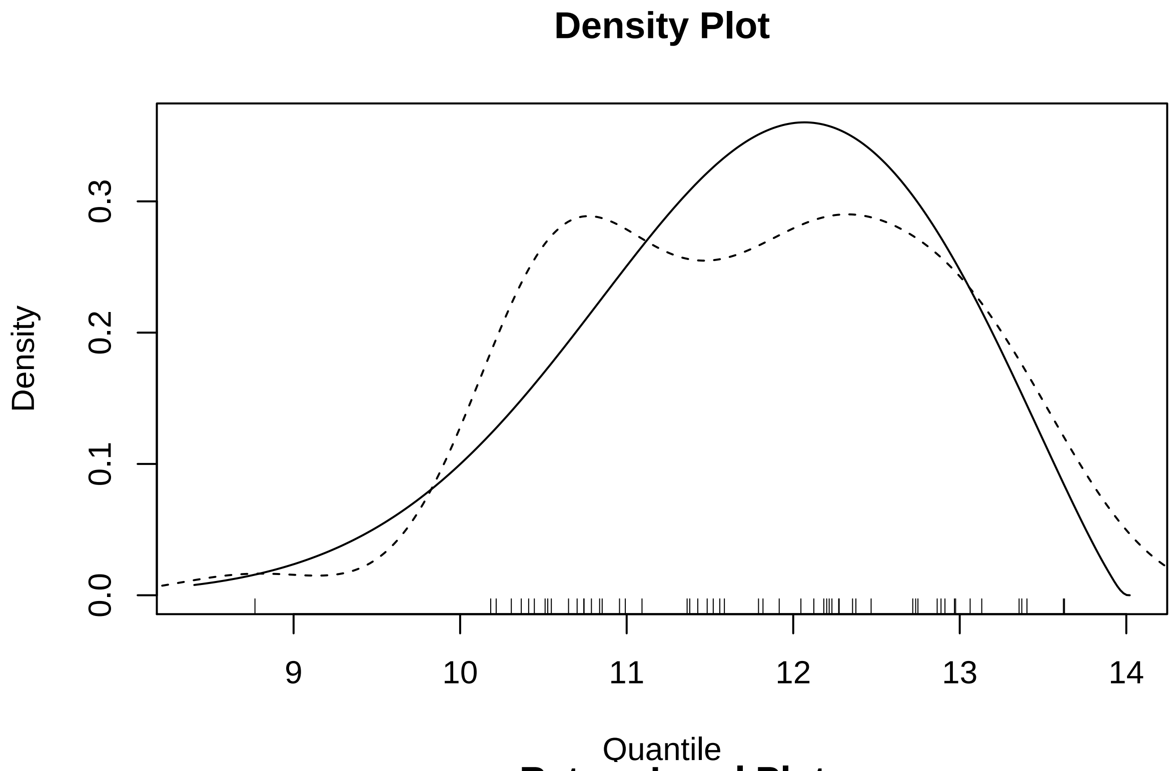
```r
confint(evd_fit2)
```

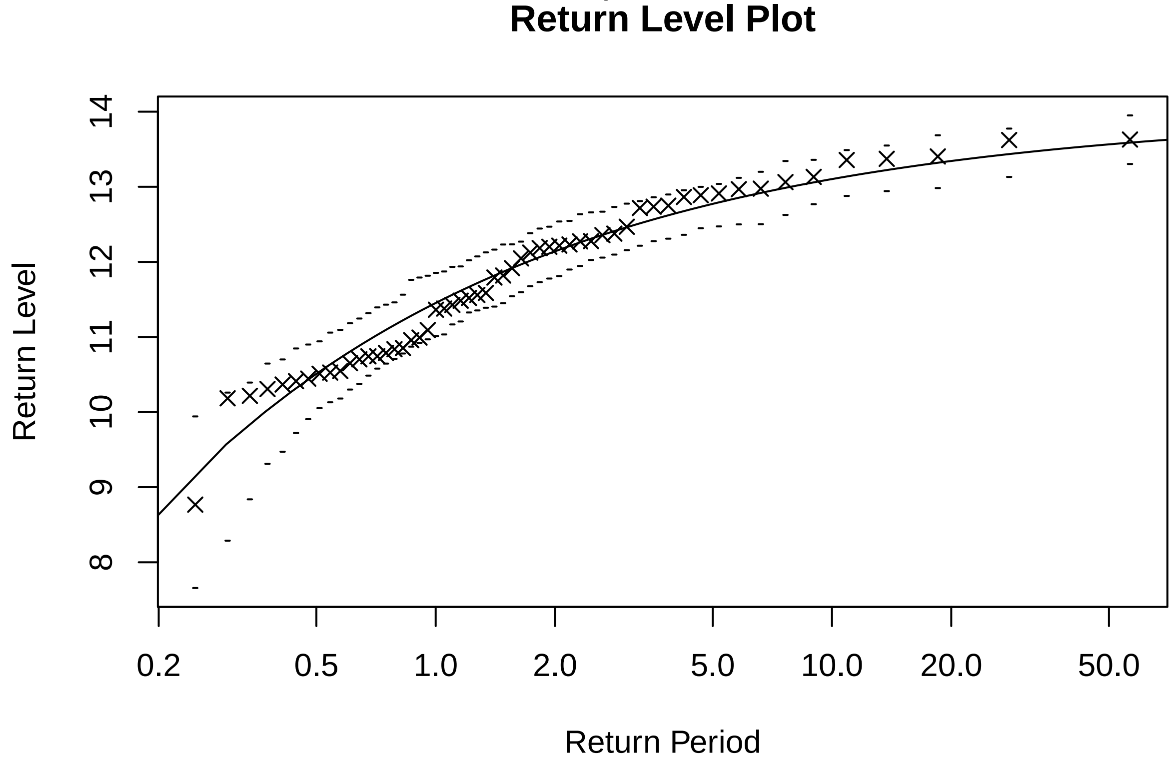
```
##                      2.5 %      97.5 %
## loc            10.8593430 12.04638832
## locscaled_year -1.2378170  0.77755532
## locNOA          2.1352029  5.22337267
## scale           0.8322430  1.48178987
## shape          -0.8187634 -0.09268677
```

```r
plot(evd_fit2)
```


Probability Plot
Model
Empirical
0.0
0.2
0.4
0.6
0.8
1.0


Quantile Plot
Empirical
Model
7
8
9
10
11
12
13
14





Inference for H_s^{100}

[10]