STAT0017 ICA 1 2018-19

Student number: 15024705

2019-03-21

Contents

1 Extremal Types Example

a)

$$F(x) = 1 - exp(\frac{1}{x})$$

$$f(x) = \frac{d}{dx}(1 - exp(\frac{1}{x}))$$

$$f(x) = \frac{exp(\frac{1}{x})}{x^2}$$

$$h(x) = \frac{1 - F(x)}{f(x)}$$

$$h(x) = \frac{1 - (1 - exp(\frac{1}{x}))}{\frac{exp(\frac{1}{x})}{x^2}}$$

$$h(x) = x^2$$

$$h'(x) = 2x$$

$$\lim_{x \to 0} h'(x) = 0$$

0 is the shape parameter for a Gumbel distribution.

To find a_n and b_n, we need to solve the following equations:

1. $a_n = h(b_n)$
2. $1 - F(b_n) = 1/n$

$$1 - (1 - exp(\frac{1}{b_n})) = \frac{1}{n}$$

$$exp(\frac{1}{b_n}) = \frac{1}{n}$$

$$\frac{1}{b_n} = log(\frac{1}{n}) = -log(n)$$

$$b_n = \frac{-1}{log(n)}$$

$$h(x) = x^2$$

$$a_n = \frac{1}{(log(n))^2}$$

b)

The Weibull distribution has a support for $x \geq 0$

Let X ~Weibull(α, β)

We can transform the Weibull distribution to a Gumbel distribution by taking, $log(X) = Y$

$$P(Y \leq y) = P(log(X) \leq y) = P(X \leq exp(y))$$
$$= 1 - exp[-(\frac{exp(y)}{\alpha})^{\beta}]$$
$$= 1 - exp[-exp(y - log(a)) \times \beta)]$$
$$= 1 - exp[-exp(\frac{y - log(\alpha)}{1/\beta})]$$
$$= 1 - exp[-exp(\frac{y - u}{\sigma})]$$

which is the Gumbel distribution with support $y \geq 0$,
where $u = log(\alpha)$ and $\sigma = 1/\beta$

The negative Weibull has support for $x < 0$ which is similar to the distribution function given in the question. By taking the transformation $log(-X) = Y$ instead, we will get the Gumbel distribution with support for $y < 0$.

c)

The theory tells us that if the random sample $m_1, \ldots, m_N$ comes from the distribution function (1) as stated in the question, it will tend to the distribution function of the Gumbel distribution as n tends to infinity. In practice, n needs to be a large number, however, usually we do not know how large n needs to be. As such, it is hard to know when the distribution (1) will tend to a Gumbel distribution and we are rarely in the limiting case where n is infinity. Before n is large enough, the shape parameter will not be 0, as such a GEV distribution will give a better approximation as it is flexible in allowing the shape parameter to vary.

d)

We will use penultimate approximation which is given by:

$$\xi_n = h'(b_n)$$

where:

$$h'(x) = \frac{d}{dx}(\frac{1 - F(x)}{f(x)})$$
$$1 - F(b_n) = 1/n$$

From part a), we have:

$$h'(x) = 2x$$
$$b_n = \frac{-1}{log(n)}$$

$$h'(b_n) = \frac{-2}{log(n)}$$

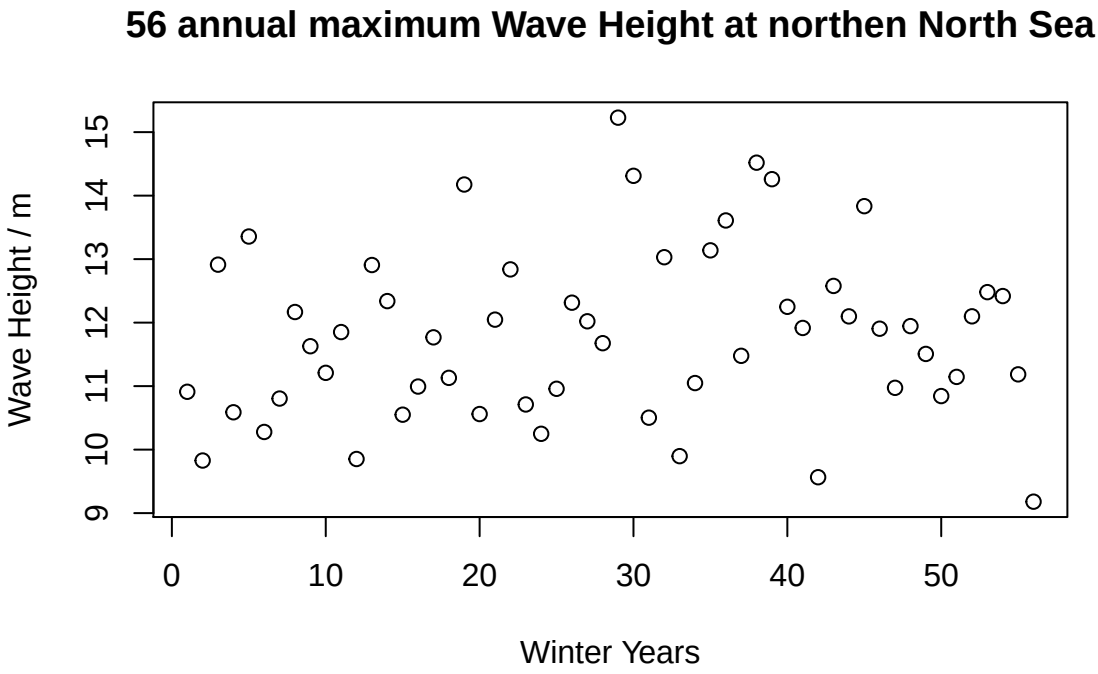
We have n = 365:

$$h'(b_n) = \frac{-2}{log(365)} = -0.338988$$

[25]

2 Exploratory analysis

2.1 Winter maxima (wm)



Sample Statistics:

```
#Mean
mean(wm$Hs)
```

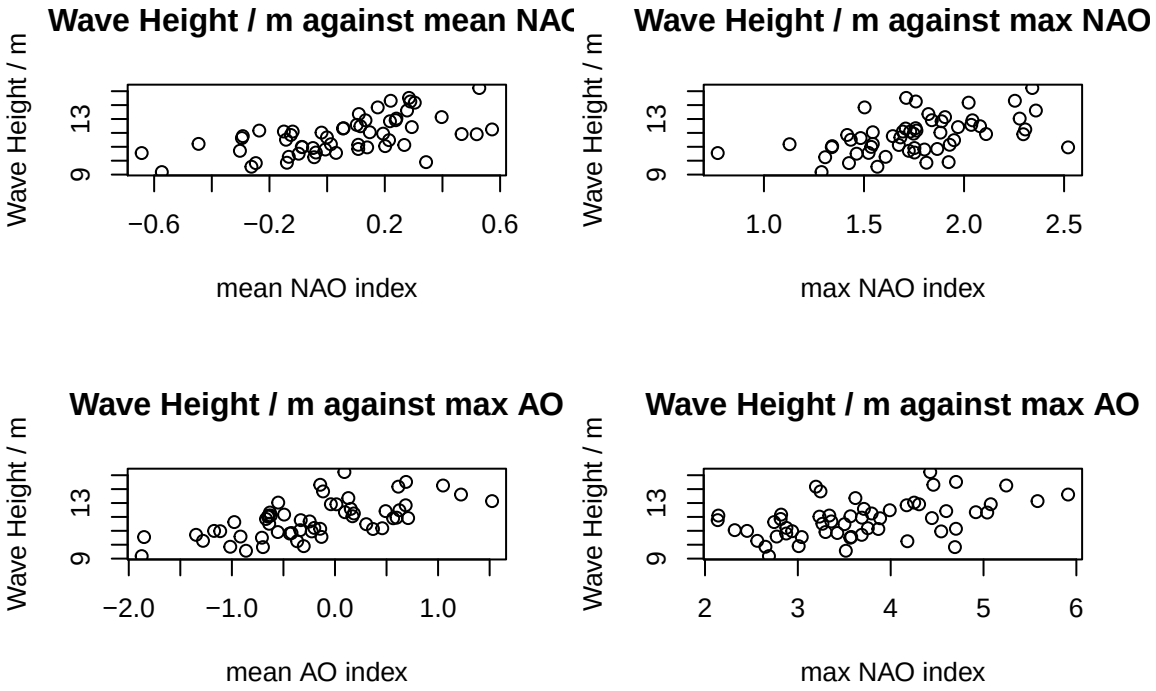
```
## [1] 11.81357
```

```
#Variance
var(wm$Hs)
```

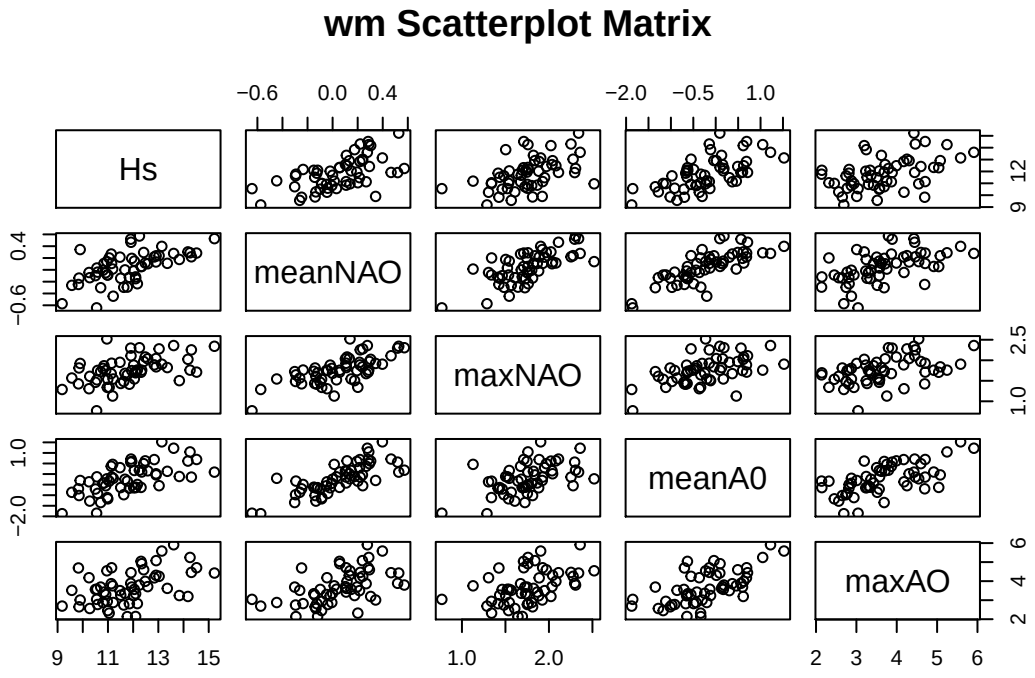
```
## [1] 1.799212
```

```
#Median
median(wm$Hs)
```

```
## [1] 11.8095
```



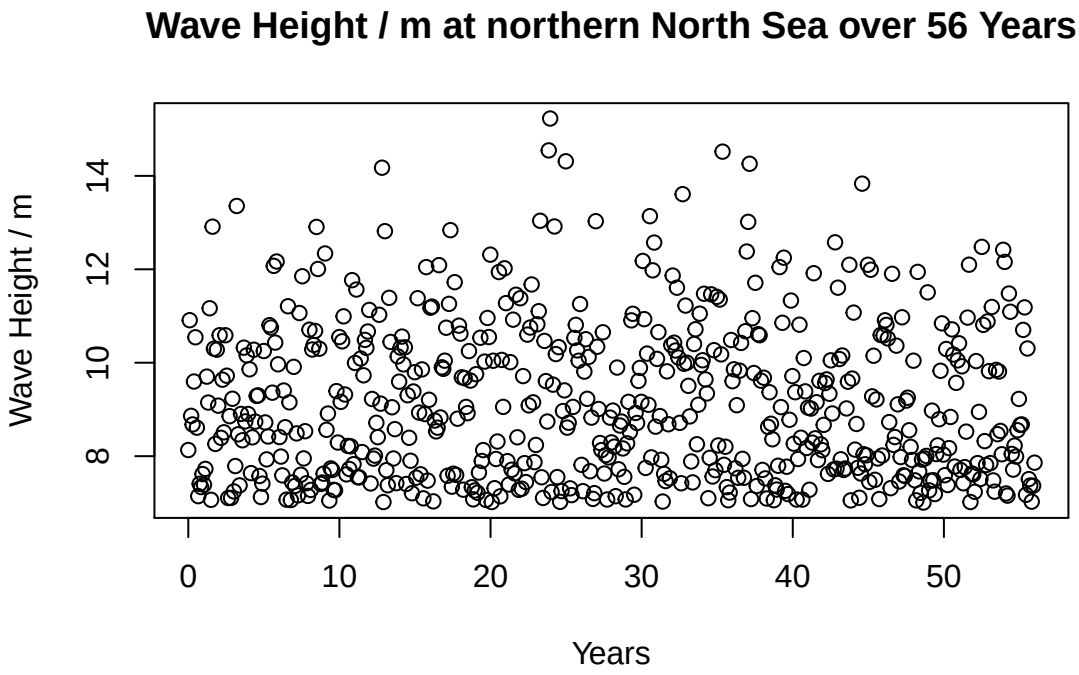
All of the covariates seem to have a linear relationship with the wave height annual maxima. This suggests that they could be useful in predicting wave height.



In addition, all the various covariates (meanNAO, maxNAO, meanAO and maxAO) seem to have linear relationships with each other. This colinearity in the covariates may cause some problems when using them

to predict Hs, and must be treated with care.

2.2 Storm peaks (pot)



Sample Statistics:

```
#Mean
mean(pot$Hs)
```

```
## [1] 9.153897
```

```
#Variance
var(pot$Hs)
```

```
## [1] 2.745975
```

```
#Median
median(pot$Hs)
```

```
## [1] 8.851
```

2.3 Comments

When working with the maxima method, we have a lot less data points to work with compared to the points over threshold method.

Plotting annual maximum wave height against years, there are few values that are low and few values that are high, most of the data seems to be centered in the middle, this could suggest that a GEV distribution would be suitable.

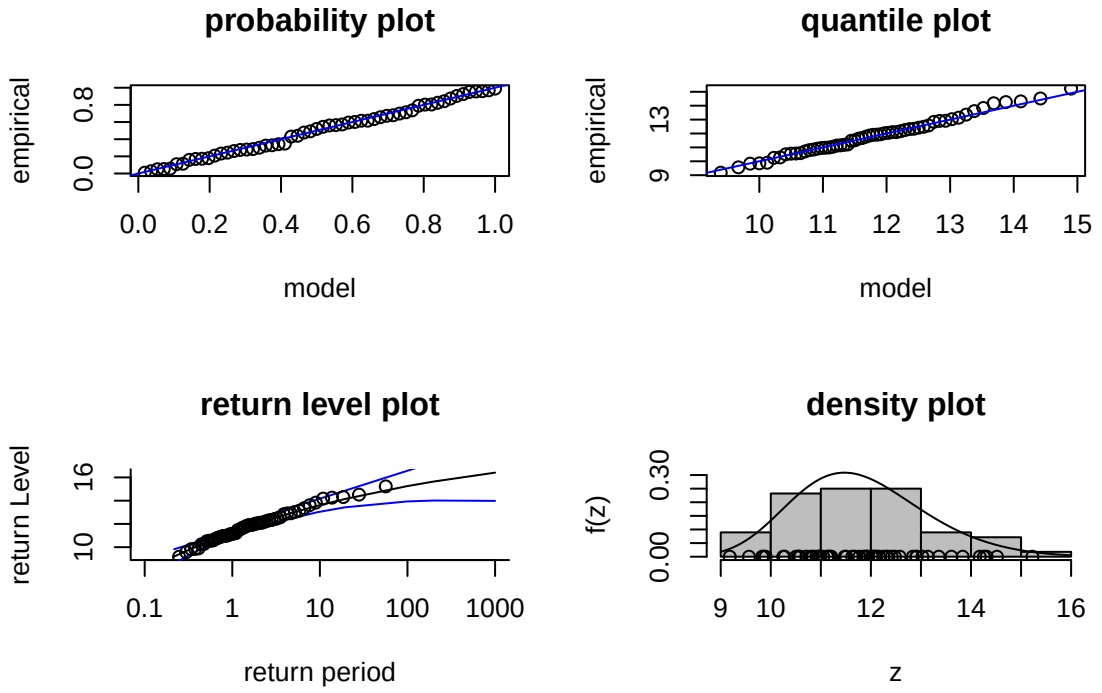
On the other hand, when using the points of threshold data, there are a lot of data points at lower wave heights, and decreasingly few as the wave height increases, which corresponds to a GP distribution.

3 Extreme value (EV) modelling of H_s

3.1 GEV modelling of winter maxima

3.1.1 Maximum Likelihood-Based Inference

```
wm.gev <- gev.fit(wm$Hs, show = FALSE) #Fit gev model to data using ismev package
wmX.gev <- fevd(wm$Hs,type ="GEV") #Fit gev model to data using extRemes package

pjn.gev.diag(wm.gev)
```



The probability plot shows that the data fits the distribution well, however this could be misleading as it hides descrepancies in the tails. This is seen in the quantile plot.

In the return level plot, the confidence bands grow wider as the return period increases which is expected.

The density plot looks like a slightly right skewed bell curve which corresponds to a possible domain of attraction in the GEV distribution.

```
### Symmetric 100*conf% confidence intervals for mu, sigma and xi ...
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
```

100 year return level 95% confidence intervals using extRemes package

```
##Normal method
ci(wmX.gev, method="normal", xrange=c(14,19), return.period = 100, alpha = 0.05)
```

[1] "100-year return level: 15.253"

[1] "95% Confidence Interval: (13.9345, 16.5718)

```
##Profile likelihood
ci(wmX.gev, method="proflik", xrange=c(14,19), return.period = 100, alpha = 0.05)
```

[1] "100-year return level: 15.253"

[1] "95% Confidence Interval: (14.5263, 17.6961)"

Predictive Inference:

Interval estimate of H_s^{100} with coverage probability of 95%

```
wmlowlim95 <- 0.025^(1/100)
wmuplim95 <- 0.975^(1/100)

qgev(c(wmlowlim95,wmuplim95), loc = wm.gev$mle[1],
     scale = wm.gev$mle[2], shape = wm.gev$mle[3])
```

```
## [1] 14.39648 16.92847
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%:

$$P(H_s^{100} \leq z|data) = G_{GEV}(z; \bar{\mu}, \bar{\sigma}, \bar{\xi})^{100}$$

We need to find: $P(H_s^{100} > z) = 0.01$

$$P(H_s^{100} \leq z) = 0.99$$
$$G_{GEV}(z; \bar{\mu}, \bar{\sigma}, \bar{\xi})^{100} = 0.99$$

where G is the cumulative density function of the GEV distribution.

$$G_{GEV}(z; \bar{\mu}, \bar{\sigma}, \bar{\xi}) = 0.99^{1/100}$$

```
root100 <- 0.99^(1/100)

#Extremes Package
qevd(root100, loc = wm.gev$mle[1], scale = wm.gev$mle[2],
     shape = wm.gev$mle[3], type = c("GEV"))
```

```
## [1] 17.22004
```

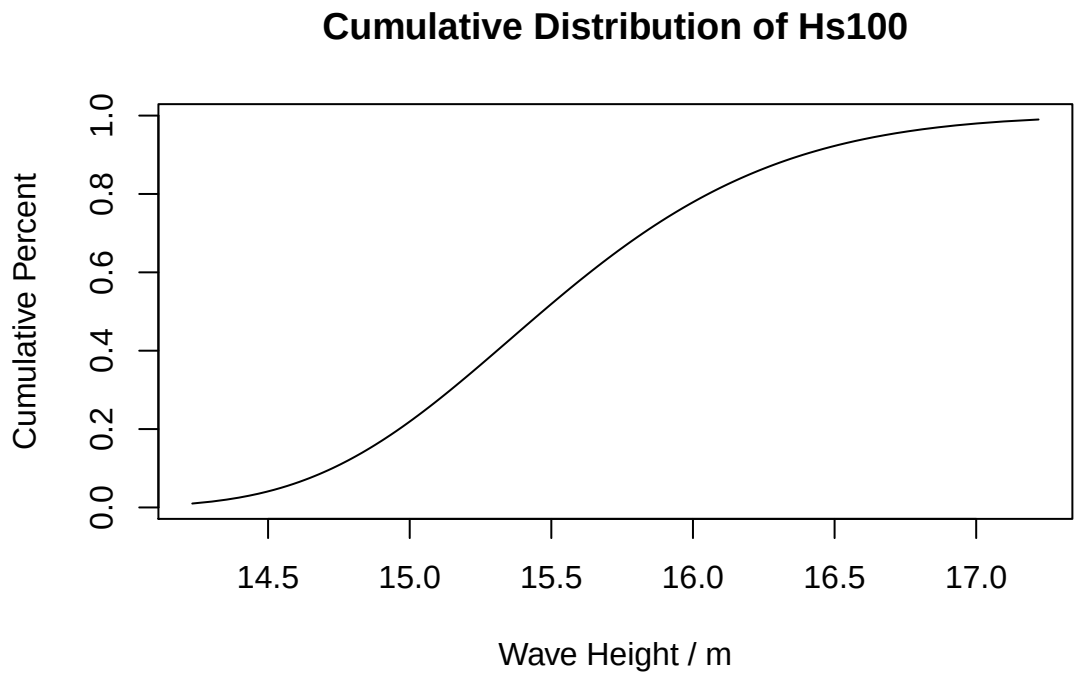
```
#revdpackage
qgev(root100, loc = wm.gev$mle[1], scale = wm.gev$mle[2], shape = wm.gev$mle[3])
```

```
## [1] 17.22004
```

Cumulative Distribution Plot (CDF) of H_s^{100}

```
wm_mle_cdf_y <- seq(0.01,0.99,0.005)
wm_mle_cdfroot <- seq(0.01,0.99,0.005)^(1/100)
wm_mle_cdf_x <- qevd(wm_mle_cdfroot, loc = wm.gev$mle[1],
                     scale = wm.gev$mle[2], shape = wm.gev$mle[3], type = c("GEV"))
```

```
plot(wm_mle_cdf_x,wm_mle_cdf_y, main = "Cumulative Distribution of Hs100"
     ,xlab = "Wave Height / m", ylab = "Cumulative Percent", type = 'l' )
```



3.1.2 Comments

The shape parameter using a GEV fit has a maximum likelihood estimate of -0.1534719, this suggests that there is a light lower tail. However, the 95% confidence interval is (-0.3295374,-0.06511831). This suggests there is quite a lot of uncertainity in the shape parameter which can be in the domain of attraction of the negative Weibull, Frechet or Gumbel distribution.

The equi-tailed confidence intervals for H_s^{100} has a higher upper annd lower limit than that of the 100-year return level which is to be expected since H_s^{100} refers to the highest wave height in a 100 year period, on the other hand, a 100-year return level refers to the wave height that is expected to be exceeded once in a 100 years. However, the profile-likelihood confidence intervals of the 100-year return level has a higher upper limit than that of the equi-tailed interval estimates of H_s^{100}. The profile likelihood interval is shorter on the lower limit as there is more data thus less uncertainity when the values are lower and conversely longer on the upper limit as there is less data and more uncertainty when the values are higher. By using an equi-tailed interval estimate, we may be underestimating the uncertainty of extreme events.

Maximum likelihood method uses the parameter estimates without taking into account the uncertainty in these estimates which could result in underestimating the predictions.

The extRemes and revdbayes package produced almost identitcal results.

3.1.3 Bayesian Inference

We choose a flat prior and use the data to find a posterior distribution since we do not have strong prior beliefs.

```r
flat_gev_prior <- set_prior(prior = "flatflat", model = "gev")

wm.gevp_bayes  <- rpost(n = 3000, model = "gev", prior = flat_gev_prior,
                        data = wm$Hs, nrep = 50)
```

Posterior 95% interval estimates for H_s^{100}:

```r
#Highest Predictive 95% Interval
predict(wm.gevp_bayes, n_years = 100, level = c(95), hpd = TRUE, type = "i")$short
```

```
##        lower    upper n_years level
## [1,] 13.9689 18.91238     100    95
```

```r
#Normal 95% Interval
predict(wm.gevp_bayes, n_years = 100, level = c(95), hpd = TRUE, type = "i")$long
```

```
##         lower    upper n_years level
## [1,] 14.29133 19.92367     100    95
```

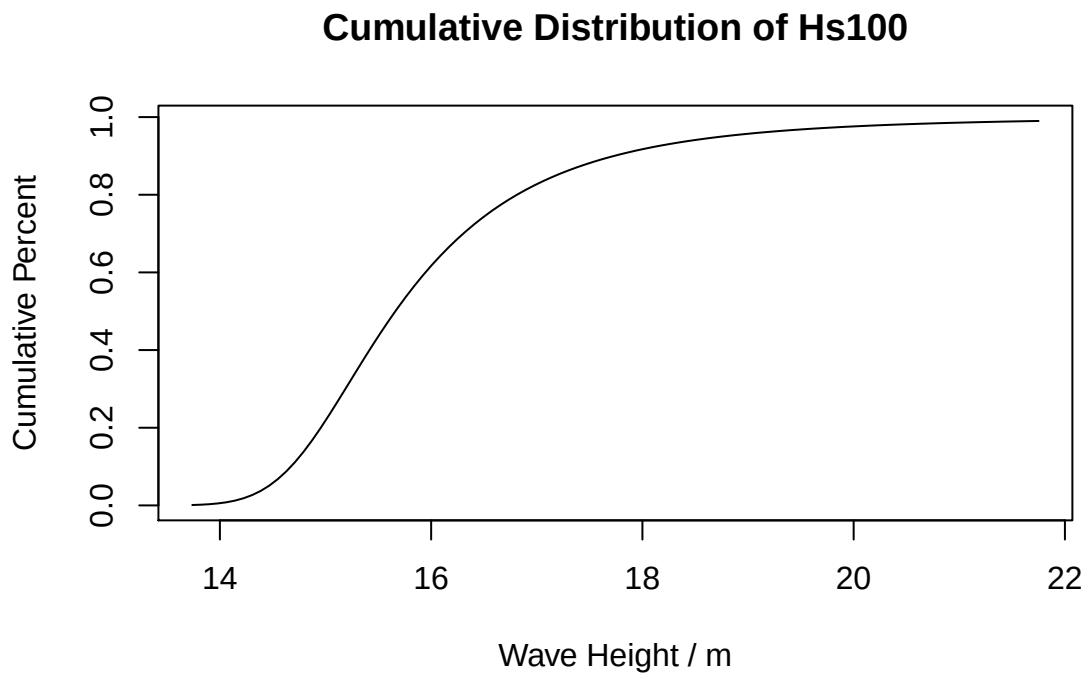
An estimate of the value that is exceeded by H_s^{100} with a probability 1%:

```r
predict(wm.gevp_bayes, n_years = 100, type = 'p')$x[100]
```

```
## [1] 21.75024
```

Cumulative Distribution Plot (CDF) of H_s^{100}:

```r
plot(predict(wm.gevp_bayes, n_years = 100, type = 'p')$x,
     predict(wm.gevp_bayes, n_years = 100, type = 'p')$y,
     main = "Cumulative Distribution of Hs100",
     xlab = "Wave Height / m", ylab = "Cumulative Percent", type = 'l' )
```



3.1.4 Comments

We choose a gev flat prior as we have no strong prior views on the behaviour of H_s^{100}.

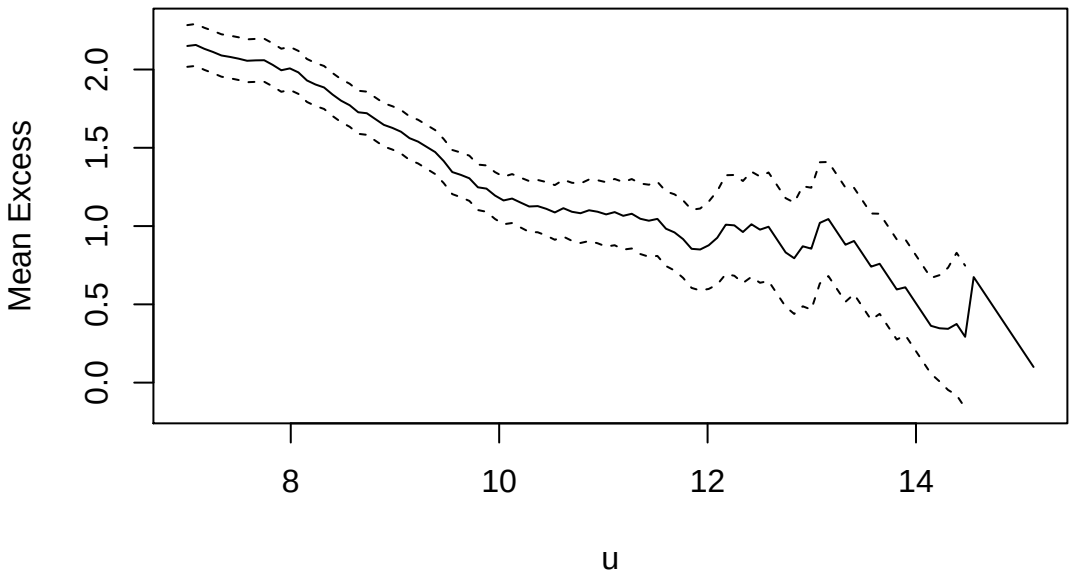
The 95% interval estimates of H_s^{100} are much wider for the Bayesian method than for the maximum likelihood method. An estimate of the value that is exceeded by H_s^{100} with a probability 1% is also significantly higher for the Bayesian method than for the maximum likelihood method.

The Bayesian method treats the parameters as random variables, thus it takes into account the variability in the parameter estimates, which may explain the increased uncertainty in its confidence interval and the larger value that is exceeded by H_s^{100} with a probability 1%.

3.2 Binomial-GP modelling of storm peaks

3.2.1 Mean residual life plot

```
mrl.plot(pot$Hs)
```



For the Mean Residuals Plot, we need to choose lowest threshold in which the plot above looks approximtely linear taking into account sample variability.

It looks as though when the threshold is around 10, the plot starts becoming linear. However, when the threshold goes past 11, the plot jumps around a lot. This is probably due to the limited data available at the tail end, so we can disregard this.

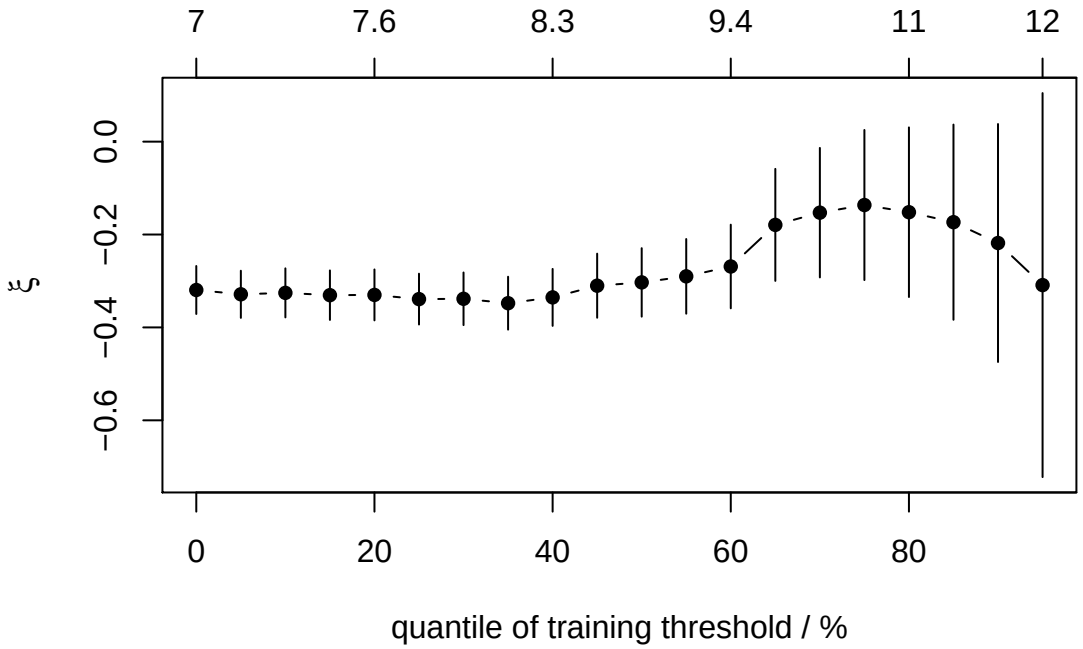
3.2.2 Stability plots

Vector of thresholds at various quantiles

```
u_vec_ns <- quantile(pot$Hs, probs = seq(0, 0.95, by = 0.05))
```

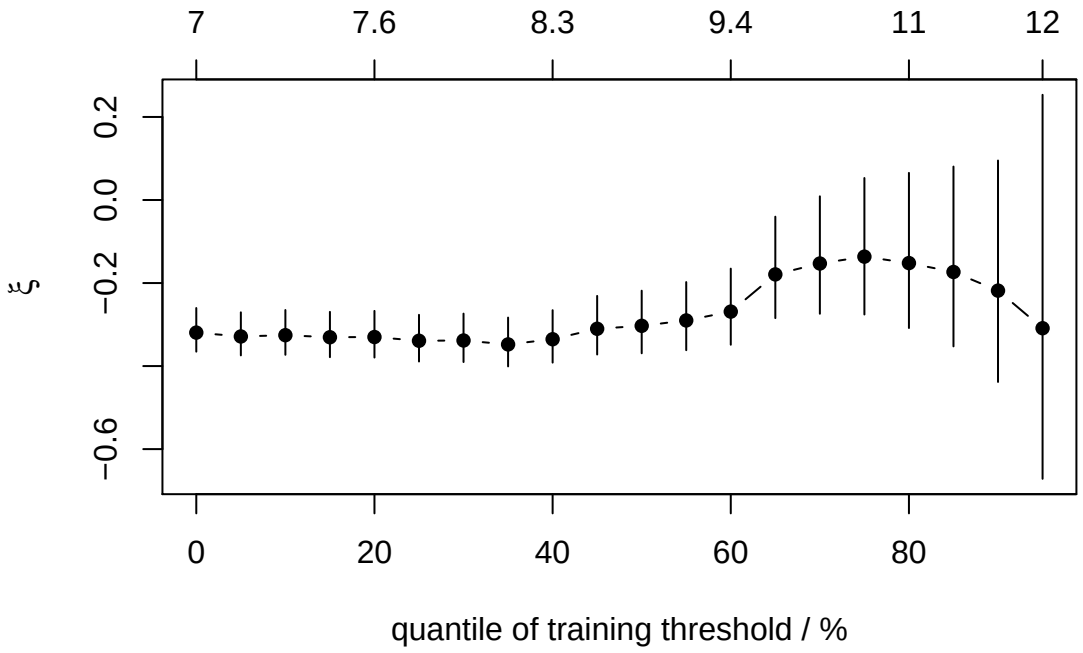
Parameter Estimate Stability

```
ns_stab <- stability(pot$Hs, u_vec = u_vec_ns)

plot(ns_stab, top_scale = "opposite")
```



Profile likelihood

```
ns_stabprof <- stability(pot$Hs, u_vec = u_vec_ns, prof = TRUE)

## Fitting at threshold number ...
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
plot(ns_stabprof, top_scale = "opposite")
```



For the Stability plots, we need to choose the lowest threshold in which the estimates above it seem constant, taking into account sample variability.

Looking at the any of the 2 above plots will give around the same information in deciding the threshold.

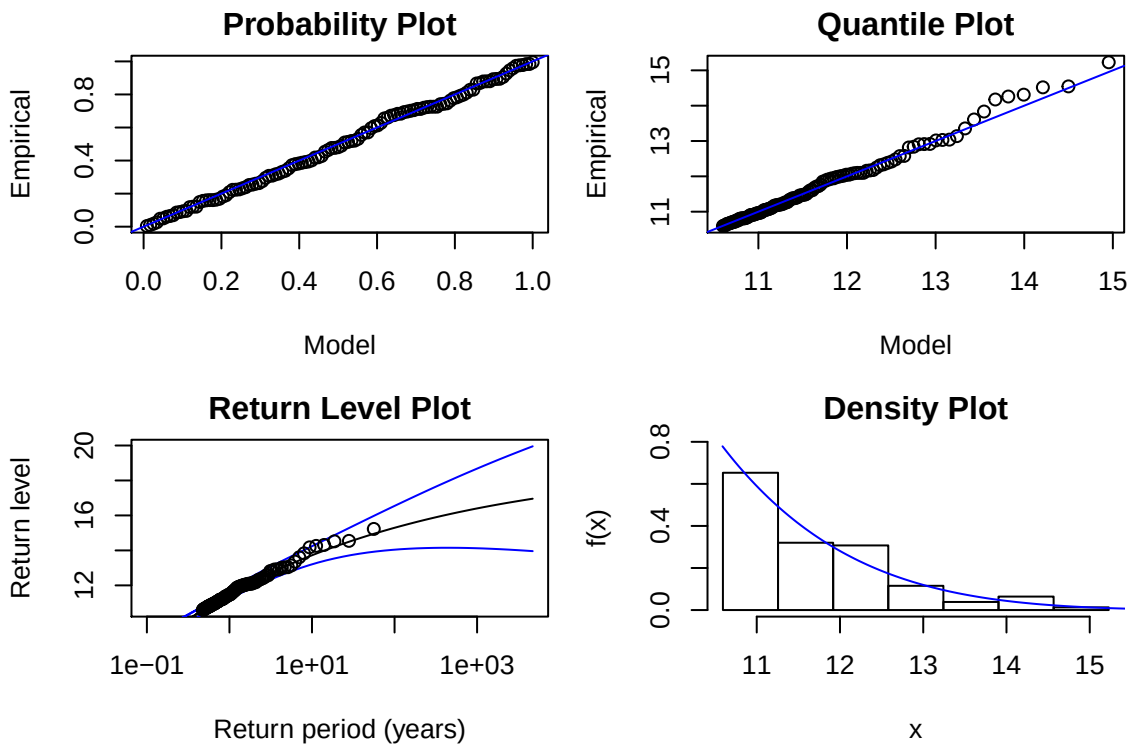
The plots look to stabilize around the the 80% threshold which corresponds to around 10 in actual value. The variance of the points are lower when the threshold is lower as there are more data points. Conversely, the higher when the threshold is higher as there are less data points. An argument can be made for other points, as it is hard to account for sample variability objectively, and it can make a huge difference.

From the Stability and Mean Residual life plot, we have choosen the 80% quantile which corresponds to around 10 as the threshold.

3.2.3 Maximum Likelihood-Based Inference

```
pot.npy <- length(pot$Hs)/56 #10.625

#Fit gp model to data using ismev package
pot.gpd <- gpd.fit(pot$Hs,threshold = u_vec_ns[["80%"]] ,npy = pot.npy, show = FALSE)

#Fit gp model to data using extRemes package
potX.gpd <- fevd(pot$Hs,threshold = u_vec_ns[["80%"]],type ="GP"
                 , span = 56, time.units = "10.625/year")
```

Diagnostic plots



Similar to the GEV diagnostics plot, the probability plot of the GPD shows that the data fits the distribution well, however this could be misleading as it hides descrepancies in the tails. This is seen in the quantile plot.

In the return level plot, the confidence bands grow wider as the return period increases which is expected.

The density plot looks like a negative exponential plot which corresponds to that of a GP distribution.

Symmetric 100*conf% confidence intervals for mu, sigma and xi:

```
pjn.gpd.conf(pot.gpd, conf = 0.95)
```

```
## $low.lim
##         pu     sigmau         xi
##  0.1263760  0.9563546 -0.3346192
##
## $up.lim
##         pu     sigmau         xi
## 0.27026262 1.61221855 0.03095151
```

95% Confidence intervals for the 100-year return level:

```
#Normal
ci(potX.gpd, method="normal", xrange=c(14,19), return.period = 100, alpha = 0.05)
```

[1] "100-year return level: 15.292"

[1] "95% Confidence Interval: (13.9612, 16.6231)

```
#Profile Likelihood
ci(potX.gpd, method="proflik", xrange=c(14,18), return.period = 100, alpha = 0.05)
```

[1] "100-year return level: 15.292"

[1] "95% Confidence Interval: (14.511, 17.5864)"

Predictive Inference:

Interval estimate of H_s^{100} with coverage probability of 95%:

Threshold at 80% quantile

```
potuplim95 <- 0.975^(1/(pot.npy*100))
potlowlim95 <- 0.025^(1/(pot.npy*100))
pu <- sum(pot$Hs > u_vec_ns[["80%"]])/length(pot$Hs)

qgp(c((1-potlowlim95)/pu,(1-potuplim95)/pu), loc = u_vec_ns[["80%"]],
    scale = pot.gpd$mle[1], shape = pot.gpd$mle[2], lower.tail = FALSE)
```

```
## [1] 14.47201 16.89936
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%

$$P(H_s^{100} \leq z) = P(H_s \leq z)^{100n_y}$$

where n_y is the mean number of observations per year.

We need to find: $P(H_s^{100} > z) = 0.01$

$$P(H_s^{100} \leq z) = P(H_s \leq z)^{100n_y} = 0.99$$

$$P(H_s \leq z)^{100n_y} = (1 - P(H_s > z))^{100n_y}$$
$$(1 - P(H_s > z))^{100n_y} = (1 - P(H_s > z|H_s > u)P(H_s > u))^{100n_y} = 0.99$$

where u is the threshold.

$$(1 - P(H_s > z|H_s > u)P(H_s > u)) = 0.99^{\frac{1}{100n_y}}$$

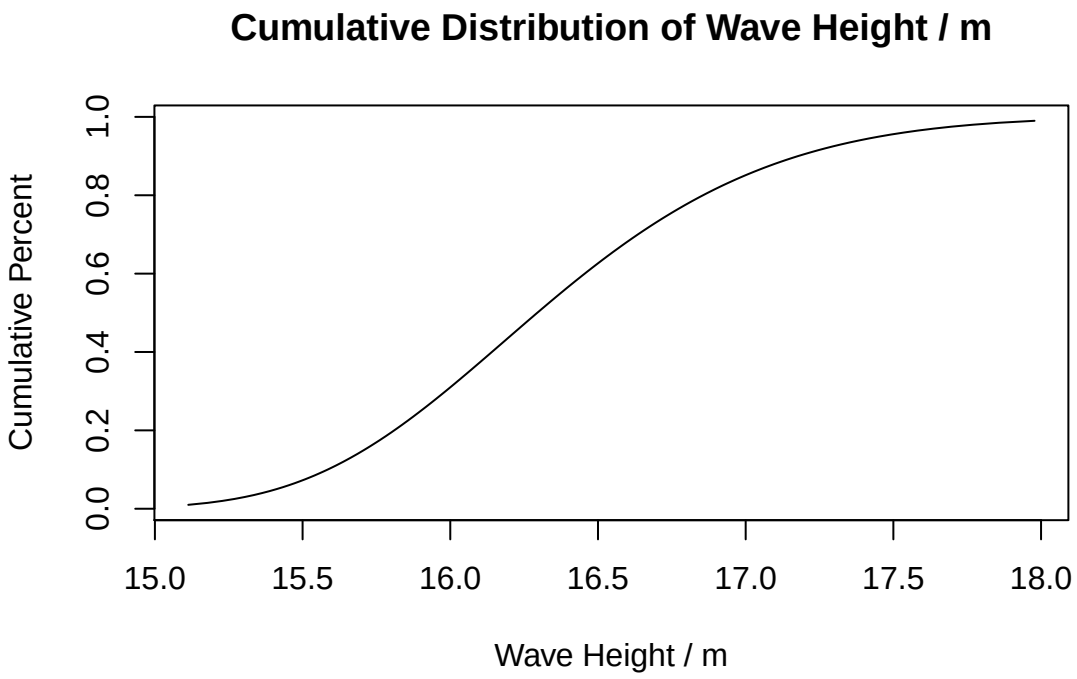
$$P(H_s > z|H_s > u) = \frac{1 - 0.99^{\frac{1}{100n_y}}}{P(H_s > u)}$$

where $P(X > z|X > u)$ can be found from the Generalized Pareto Distribution and P(X>u) is the probability of the observations above the threshold.

```
pot1percent <- 0.99^(1/(pot.npy*100))

qgp(((1-pot1percent)/pu), loc = u_vec_ns[["80%"]],  scale = pot.gpd$mle[1],
    shape = pot.gpd$mle[2], lower.tail = FALSE)
```

```
## [1] 17.18051
```

Cumulative Distribution Plot (CDF) of H_s^{100}:



We repeat a portion of the process for thresholds at the 70% quantile and 90% quantile for comparison.

Threshold at 70% quantile:

```
#70%
pot.gpd70 <- gpd.fit(pot$Hs,threshold = u_vec_ns[["70%"]] ,npy = pot.npy, show = FALSE)

pu70 <- sum(pot$Hs > u_vec_ns[["70%"]])/length(pot$Hs)
```

Interval estimate of H_s^{100} with coverage probability of 95%:

```
qgp(c((1-potlowlim95)/pu70,(1-potuplim95)/pu70), loc = u_vec_ns[["70%"]],
    scale = pot.gpd$mle[1], shape = pot.gpd$mle[2], lower.tail = FALSE)
```

```
## [1] 14.21313 16.49166
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%:

```
qgp((1-pot1percent)/pu70, loc = u_vec_ns[["70%"]], scale = pot.gpd$mle[1],
    shape = pot.gpd$mle[2], lower.tail = FALSE)
```

```
## [1] 16.75558
```

Threshold at 90% quantile:

```
#90%
pot.gpd90 <- gpd.fit(pot$Hs,threshold = u_vec_ns[["90%"]] ,npy = pot.npy, show = FALSE)

pu90 <- sum(pot$Hs > u_vec_ns[["90%"]])/length(pot$Hs)
```

Interval estimate of H_s^{100} with coverage probability of 95%:

```
qgp(c((1-potlowlim95)/pu90,(1-potuplim95)/pu90), loc = u_vec_ns[["90%"]],
    scale = pot.gpd$mle[1], shape = pot.gpd$mle[2], lower.tail = FALSE)
```

```
## [1] 14.77458 17.46444
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%:

```
qgp((1-pot1percent)/pu90, loc = u_vec_ns[["90%"]], scale = pot.gpd$mle[1],
    shape = pot.gpd$mle[2], lower.tail = FALSE)
```

```
## [1] 17.776
```

3.2.4 Comments

The shape parameter of the GEV distribution and GPD distribution are very similar which corresponds to the theory that the shape parameter of the GPD is the same as that of the corresponding GEV limit.

In general, choosing a threshold is not an objective method. For that reason, we have done analysis for thresholds at 70% and 90% for comparison. The 95% interval estimates for the 70%, 80% and 90% thresholds are (14.21313,16.49166), (14.47201,16.89936) and (14.77458,17.46444) respectively. It seems that when the threshold is increased, the upper and lower limits of the interval estimates increase. The differences in these estimates are quite significant, up to approximately 5% of difference. The choice of threshold plays an important role in the predictions.

According to the theory, the choice of the threshold should be high as possible, but in practice, this is not necessarily the most ideal as there is less data as the threshold increases. It is basically a bias-variance trade off. A low threshold introduces biases in the model, whereas a high threshold increases the variance of the estimates.

3.2.5 Bayesian Inference

Choose uninformative priors:

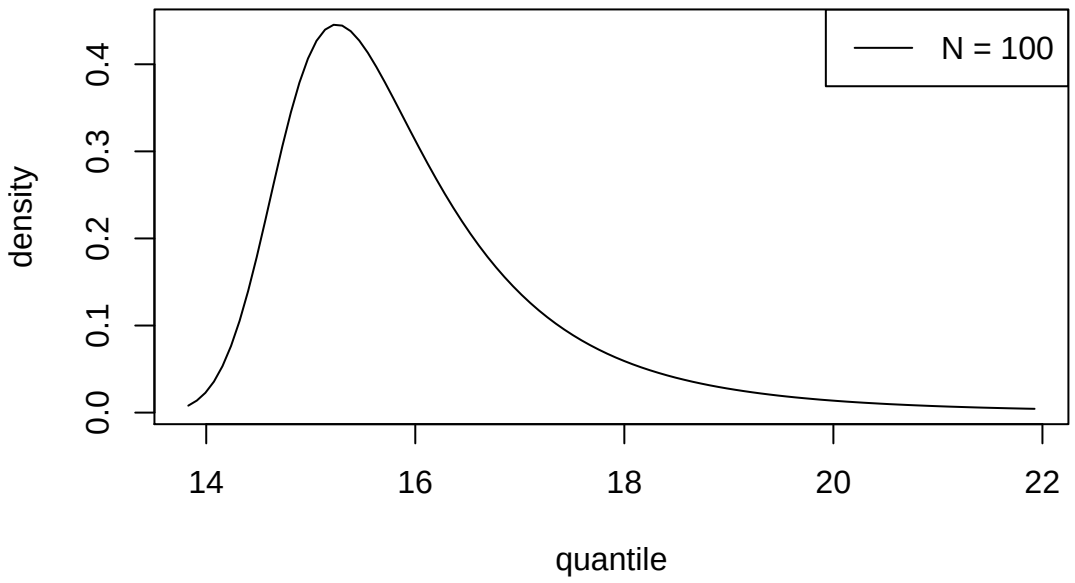
```
pot_prior<- set_prior(prior = "flatflat", model = "gp", min_xi = -1)
pot_bpprior<- set_bin_prior(prior = "jeffreys")
```

Compute posterior distribution:

```
pot.bayes.gpd  <- rpost(n = 3000, model = "bingp", prior = pot_prior, data = pot$Hs,
                  npy = pot.npy, bin_prior = pot_bpprior,
                  thresh = u_vec_ns[["80%"]])
```

Probability distribution plot:

```
plot(predict(pot.bayes.gpd, type = "d", n_years = 100))
```



Predictive Inference:

Posterior 95% interval estimates for H_s^{100}

```
#Highest Predictive 95% Interval
predict(pot.bayes.gpd, n_years = 100, level = c(95), hpd = TRUE, type = "i")$short
```

```
##         lower    upper n_years level
## [1,] 14.02822 18.97495     100    95
```

```
#Normal 95% Interval
predict(pot.bayes.gpd, n_years = 100, level = c(95), hpd = TRUE, type = "i")$long
```

```
##         lower    upper n_years level
## [1,] 14.34649 20.03012     100    95
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%:

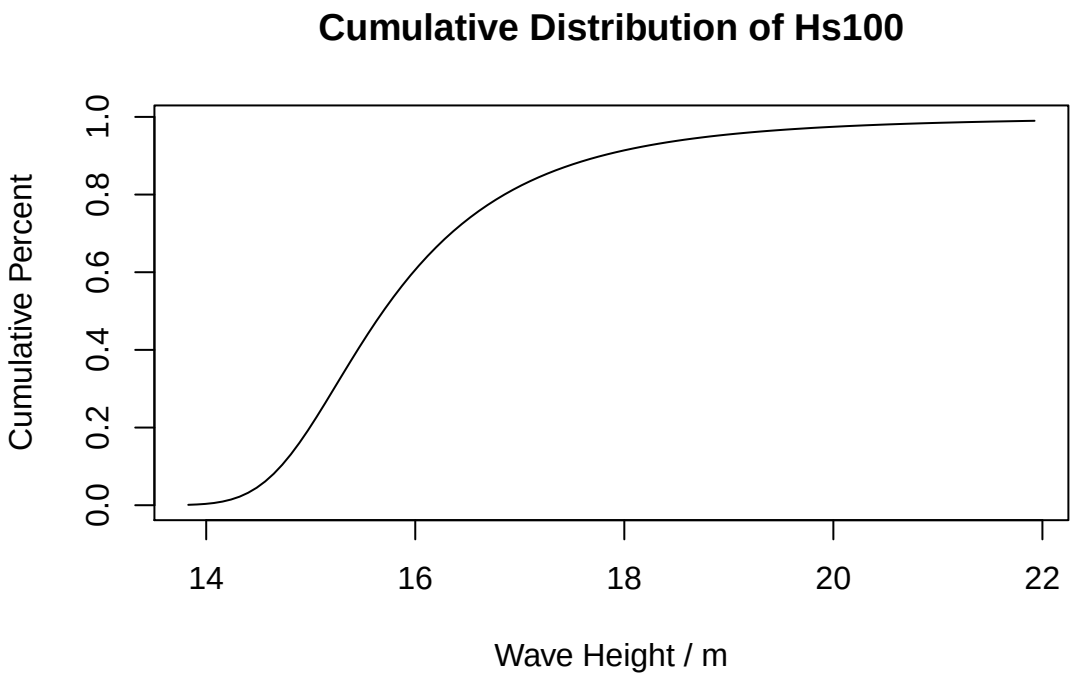
Find z for $P(H_s^{100} > z) = 0.01$

```
predict(pot.bayes.gpd, n_years = 100, type = 'p')$x[100]
```

```
## [1] 21.92453
```

Cumulative Distribution Plot (CDF) of H_s^{100}:

```
plot(predict(pot.bayes.gpd, n_years = 100, type = 'p')$x,
     predict(pot.bayes.gpd, n_years = 100, type = 'p')$y,
     main = "Cumulative Distribution of Hs100",
     xlab = "Wave Height / m", ylab = "Cumulative Percent", type = 'l' )
```



Threshold at 70% Quantile:

```
pot.bayes.gpd70  <- rpost(n = 3000, model = "bingp", prior = pot_prior, data = pot$Hs,
                 npy = pot.npy, bin_prior = pot_bpprior, thresh = u_vec_ns[["70%"]])
```

Interval estimate of H_s^{100} with coverage probability of 95%:

```
predict(pot.bayes.gpd70, n_years = 100, level = c(95), hpd = TRUE, type = "i")$short
```

```
##         lower    upper n_years level
## [1,] 14.08994 18.31413     100    95
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%

```
predict(pot.bayes.gpd70, n_years = 100, type = 'p')$x[100]
```

```
## [1] 20.33107
```

Threshold at 90% Quantile

```
pot.bayes.gpd90  <- rpost(n = 3000, model = "bingp", prior = pot_prior, data = pot$Hs,
                 npy = pot.npy, bin_prior = pot_bpprior, thresh = u_vec_ns[["90%"]])
```

Interval estimate of H_s^{100} with coverage probability of 95%:

```
predict(pot.bayes.gpd90, n_years = 100, level = c(95), hpd = TRUE, type = "i")$short
```

```
##         lower    upper n_years level
## [1,] 14.05358 18.95918     100    95
```

An estimate of the value that is exceeded by H_s^{100} with a probability of 1%

```
predict(pot.bayes.gpd90, n_years = 100, type = 'p')$x[100]
```

```
## [1] 23.42622
```

3.2.6 Comments

We choose a flat gp prior and jefferys prior for the binomial portion of the model, both of which are uninformative priors as we do not have strong prior beliefs.

The predict function provides the option to calculate the shortest predictive interval of a given confidence level by setting the parameter 'hpd' to be true. This is desirable as a narrower interval represents less uncertainty. We will therefore use the shortest interval for the interval estimate. The 95% interval estimates are (14.06853,18.34827), (14.03596,18.9851) and (14.0566,18.96718) for threshold quantiles at 70,80,90 respectively. Unlike the MLE method, the lower and upper limits of the interval estimates vary by increasing and decreasing values as the threshold increases. The difference in the interval estimates are also not as significant.

However, when comparing the interval estimates found using MLE and the Bayesian method, we find that there there is quite a sigifnicant difference, which perhaps can be explained by the Bayesian framework taking into account the uncertainity of parameter estimates.

[25]

3.3 Reporting to your client

The four types of analyses done are as below:

1. Generalized Extreme Value Distribution with Maximum Liklihood estimation (GEV_MLE)
2. Generalized Extreme Value Distribution with Bayesian framework (GEV_Bayes)
3. Generalized Pareto Distribution with Maximum Liklihood estimation(GPD_MLE)
4. Generalized Pareto Distribution with Bayesian framework (GPD_MLE)

First, we compare the interval estimate of H_s^{100} with coverage probability of 95%.

GEV_MLE: (14.39648,16.92847)

GEV_Bayes: (13.95543,19.11209)

GPD_MLE: (14.47201,16.89936)

GPD_Bayes: (14.03596,18.9851)

The confidence intervals when using MLE estimates tends to be narrower. This is to be expected as the MLE estimates use point estimates and do not take into account the uncertainity of these point estimates. However, this uncertainty is taken into account when using the Bayesian method, therefore the intervals using the Bayesian framework have a wider confidence interval. It is important that we take into account the uncertainity in the point estimations as the amount of data available is quite limited. Therefore, it is recommended to use one of the Bayesian methods.

Next, we compare the value z for which $P(H_s^{100} > z) = 0.01$.

GEV_MLE: 17.22004

GEV_Bayes: 21.75024

GPD_MLE: 17.18051

GPD_Bayes: 22.3633

From the results, it shows that for predictive inferences the MLE methods tend to give significantly smaller values than the Bayesian methods. The MLE methods could be underestimating the extreme values as they do not take into account the uncertainity of the point estimates that are used as the parameters of the prediction.

The GEV_MLE and GPD_MLE methods have similar values, likewise GEV_Bayes and GPD_Bayes methods have similar values. This could perhaps suggest the differences between GPD and GEV methods do not have significantly differing results.

To choose between the GEV and GPD methods, we explore some published literature. The paper by Bucher and Zhou/Block Maxima vs. Peak-over-Threshold (2018), suggests that there is a general consensus among extreme value statistics researchers that GPD model produces more efficient estimators as more observations are used for the calculation. On the otherhand, the GEV model seems to be more consistent in return periods etimation. However, ultimately between the two models, neither method was shown to be objectively better.

Since the two models seem to perform similarly, we recommend using the GEV model as it easier to perform since determining a threshold for the GPD model is difficult and may be inconsistent as the choice of the threshold is largely subjective and may differ greatly from person to person. Differing choices of the threshold may significantly affect the model although possibly to a lower degree when using the Bayesian method.

The analysis we have chosen to present is the GEV_Bayes one.

The 3 requirements are as below using the GEV_Bayes method:

A: An interval estimate of H_s^{100} with a coverage probability 95%

```
predict(wm.gevp_bayes, n_years = 100, level = c(95), hpd = TRUE, type = "i")$short
```

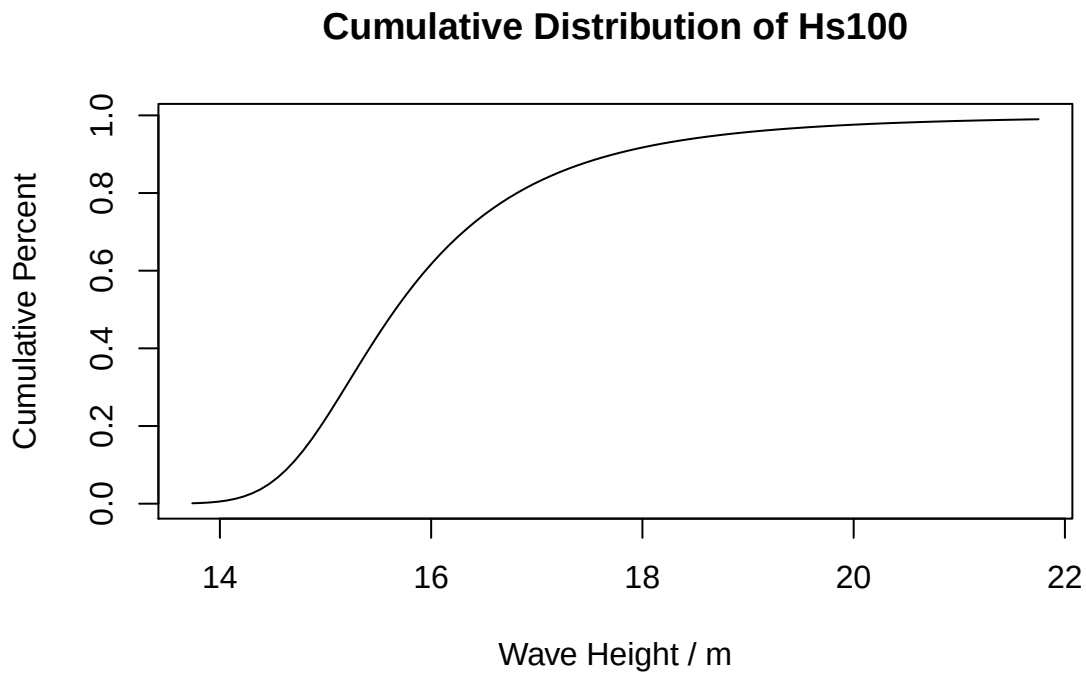
```
##        lower    upper n_years level
## [1,] 13.9689 18.91238     100    95
```

B: An estimate of the value that is exceeded by H_s^{100} with a probability 1%

```
predict(wm.gevp_bayes, n_years = 100, type = 'p')$x[100]
```

```
## [1] 21.75024
```

Cumulative Distribution Plot (CDF) of H_s^{100}



4 EV regression modelling of winter maximum H_s on NAO

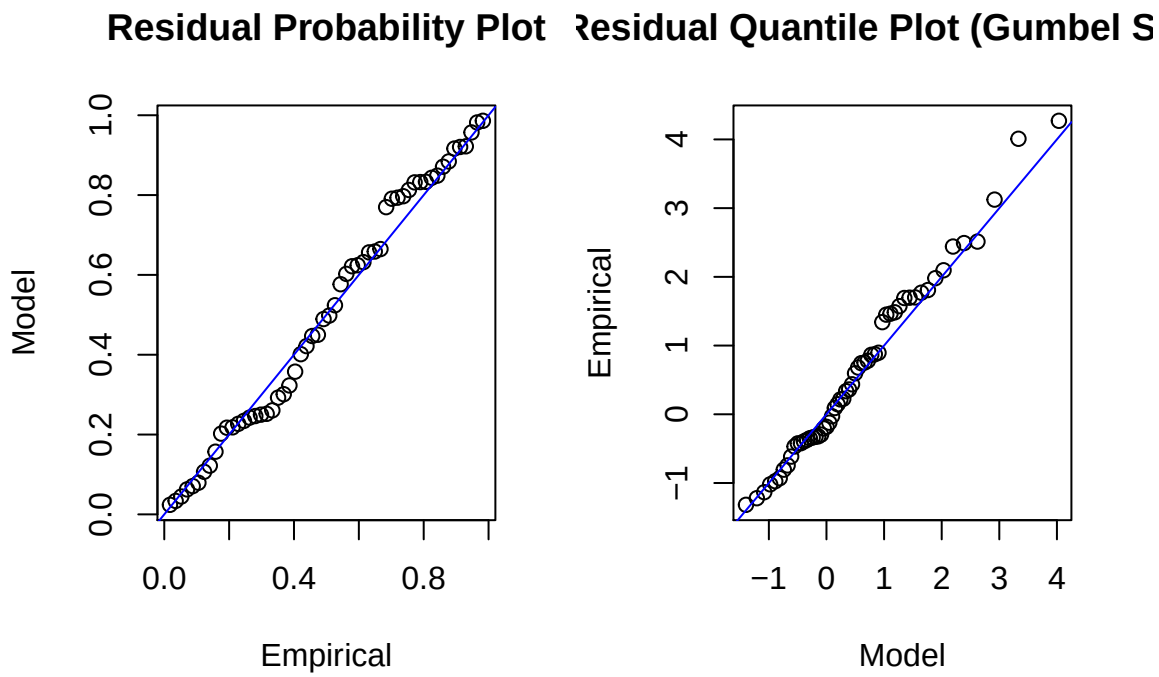
4.1 Build a GEV regression model

```
scaled.year <- (wm[,2]-1955)/(2010-1955)
ydat <- cbind(scaled.year,wm[,c(3:6)])
ymat <-matrix(as.numeric(unlist(ydat)),nrow=nrow(ydat))
colnames(ymat) <- c("scaled year", "meanNAO", "maxNAO", "meanAO","maxAO")


GEV_Model <- gev.fit(wm$Hs,ydat=ymat, mul = c(4), sigl =  c(2))
```

```
## $model
## $model[[1]]
## [1] 4
##
## $model[[2]]
## [1] 2
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
##
## $conv
## [1] 0
##
## $nllh
## [1] 77.07671
##
## $mle
## [1] 11.6402808  1.1216425  0.8937640  0.6076349 -0.2003818
##
## $se
## [1] 0.1494989 0.1680809 0.1061476 0.2925930 0.1297908
```

The year covariate is scaled from the interval (1955,2010) to (0,1) to prevent convergence issues. The final model was chosen using an interative process. We started with the most basic model (with no covariates) and started by adding single and multiple covariates in the location, scale and shape parameters. Models with low negative log likelihood were chosen. These models were then compared using the likelihood ratio test. From there we chose the model that is parsimonious and has a low negative log likelihood.

```
gev.diag(GEV_Model)
```



The diagnostics plots show the residual probability plots. From the residual plots, most of the observations lie on the x = y line which is indicator that the model is reasonable.

4.2 Inference for H_s^{100}

Aim: To estimate the the probability, $P(H_s^{100} \leq x)$ for some value x of interest.

Let X_i be the annual storm maxima.

Since the annual storm maxima can be considered as independent, the largest value in 100 years can be expressed as:

$$P(H_s^{100} \leq x) = \prod_{i=1}^{100} P(X_i \leq x)$$

$$= \prod_{i=1}^{100} G_{GEV}(x, \mu(y_i), \sigma(z_i), \xi)$$

where
$G_{GEV}()$ is the GEV CDF
$\mu(y_i) = \mu_0 + \mu_1 y_i$
$\mu(y_i) = 11.6402808 + 1.1216425 y_i$ (coefficients given by model)
$\sigma(z_i) = \sigma_0 + \sigma_1 z_i$
$\sigma(z_i) = 0.8937640 + 0.6076349 z_i$ (coefficients given by model)
$\xi = -0.2003818$ (coefficients given by model)
y_i is the value of the meanAO of i^{th} year
z_i is the value of the meanNAO of i^{th} year

Each Z_i has a different GEV distrubution conditional on the covariates meanNAO and meanAO. For each year use the values of meanNAO and meanAO to get the value of the parameters for the GEV distribution

using the above formula. With the parameters of the GEV and some value x of interest, we can find the probability $P(X_i \leq x)$. Repeat this for the 100 years and multiply all these probabilities together.

The following code will be useful in finding an estimate for $P(H_s^{100} \leq x)$ for a supplied value of x.

```
#The function takes in 3 arguments, x (number) is the value of interest,
#y (numeric vector) is the vector of meanAO values,
#z (numeric vector) is the vector of meanNAO values.
#The shape paramter is fixed as per the model.
#Need revdbayes package to be loaded

Estimate <- function(x=0,y=c(0),z=c(0)){

  mu_y <- 11.6402808 + 1.1216425*y

  sig_z <- 0.8937640 +  0.6076349*z

  shape <- rep(-0.2003818,length(y))

  probs <- pgev(x, loc =  mu_y, scale =  sig_z, shape = shape) #revdbayes library

  return(prod(probs))
}
```

[10]