Risk and Environmental Sustainability: R Tutorial

Linda Mhalla (adapted from Hugo Winter)

Introduction

The aim of these exercises is to aide understanding of the theory introduced in the lectures by giving the students a chance for some hands-on experience with a statistical computing program. The most common statistical programming language is `R`. For more information see https://www.r-project.org/ and this tutorial to `R`.

Annual maxima

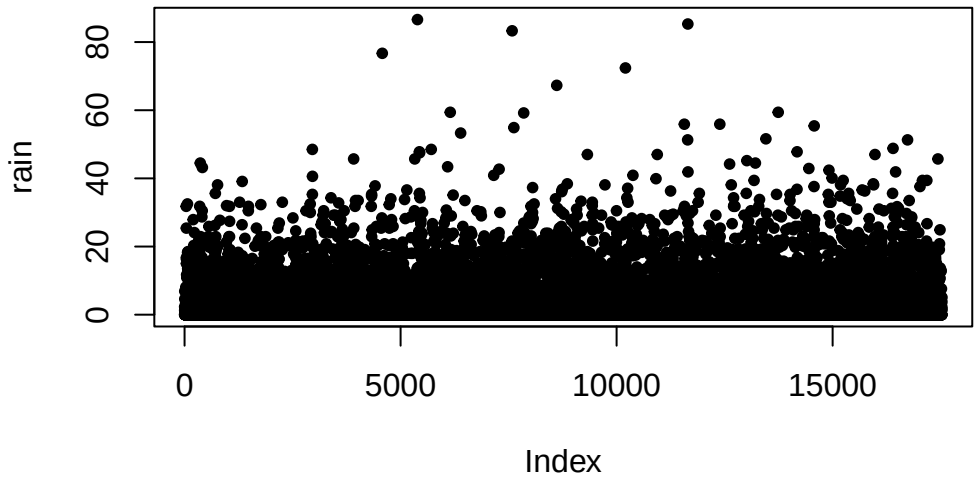
Walkthrough

The first set of exercises will concern the concept of modelling maxima. We are going to analyse daily rainfall data (in mm) from south-west England. The data set comes from the `ismev` package and can be loaded in with the following commands:

```
# install.packages("ismev") # uncomment and run. Only needs to be run once
library(ismev)
data(rain)
```

A variable called `rain` should now have been loaded into the workspace (type `ls()` into the terminal to check if this is the case). We shall plot the data to get a feel of what the data look like.

```
plot(rain, pch=20)
```



Firstly we are going to model the annual maxima of the rainfall data. Before we can fit a distribution, the annual maxima need to be extracted:

```
years <- rep(1:48, rep(c(365,365,366,365), times = 12))[-17532]
#this creates a day-wise year indicator.
#if you want to check the syntax of e.g. the rep function
#you can run the help command
# ?rep
rain.ann.max <- unlist(lapply(X = split(rain,years), FUN = max))
```

Now we have a variable `rain.ann.max` that contains annual maxima. Maxima can be modelled using the Generalised Extreme Value (GEV) distribution with parameters η, τ and ξ which refer to the location, scale, and shape parameters respectively. It is possible to fit a GEV distribution with likelihood using the `fevd` function from the `extRemes` package. To load in the package:

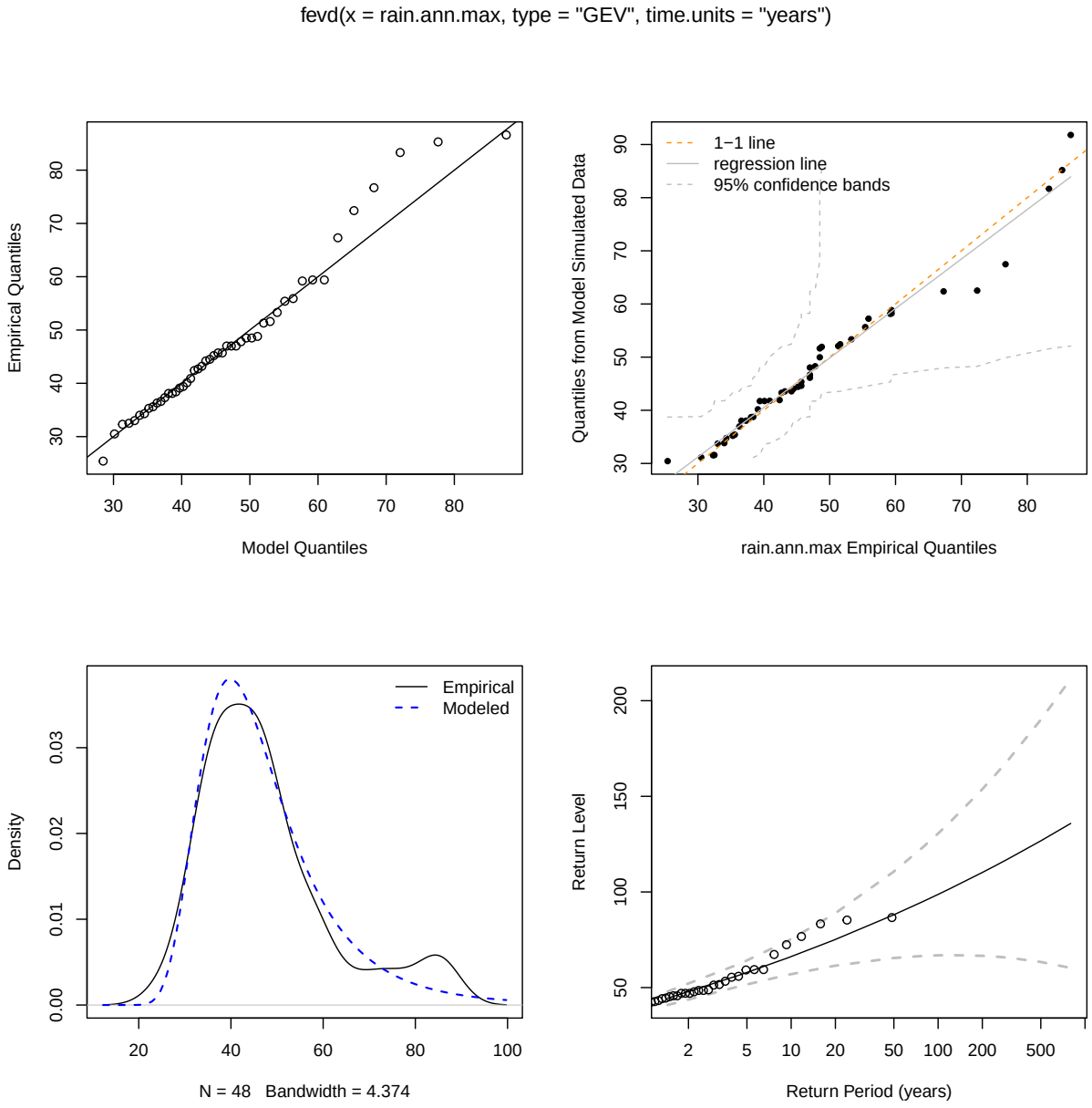
```
# install.packages("extRemes") # uncomment and run. Only needs to be run once
library(extRemes)
```

Then to fit the GEV you need to type:

```
gev.fit <- fevd(x = rain.ann.max, type = "GEV", time.units = "years")
```

The output from the fit is contained within the variable `gev.fit`. To obtain diagnostic plots to assess the fit then type:

```
plot(gev.fit)
```



From the output we can see that the QQ-plot and return level plot suggest that the distribution provides a good fit to the data. We shall now look directly at the estimates of the location, scale and shape parameters (η, τ, ξ) by typing

```
gev.fit$results$par
```

```
  location      scale      shape
40.7830335  9.7284060  0.1072355
```

It is important to assess the uncertainty associated with these parameter estimates. To obtain estimates for 95% confidence intervals we run:

```
ci.fevd(x = gev.fit, alpha = 0.05, type = "parameter")
```

```
fevd(x = rain.ann.max, type = "GEV", time.units = "years")

[1] "Normal Approx."

         95% lower CI   Estimate 95% upper CI
location   37.6941916 40.7830335   43.8718754
scale       7.3991505  9.7284060   12.0576614
shape      -0.1055497  0.1072355    0.3200207
```

Finally, we need to estimate return levels outside the scope of our data using the fitted GEV model:

```
gev.rl <- return.level(x = gev.fit,
                       return.period = c(10,100,1000,10000),
                       do.ci = TRUE, alpha = 0.05)
```

By typing `gev.rl` into the terminal we can see the 10-, 100-, 1000- and 10000-year return levels with 95% confidence intervals.

```
gev.rl
```

```
fevd(x = rain.ann.max, type = "GEV", time.units = "years")

[1] "Normal Approx."

                       95% lower CI  Estimate 95% upper CI
10-year return level        56.67333  65.54301     74.41268
100-year return level       66.85346  98.63615    130.41884
1000-year return level      58.37286 140.34002    222.30718
10000-year return level     21.17530 193.64365    366.11200
```

Exercises

The `ismev` package also contains the dataset `portpirie` which contains annual maximum sea levels (in m) at Port Pirie in South Australia over the period 1923–1987. These data are already annual maxima so you don't need to extract them. After running `data(portpirie)` the data can be accessed by typing `portpirie$SeaLevel` into the terminal.

1. Plot the sea level data.
2. Fit the GEV distribution to the data. What are the parameter estimates telling you about the shape of the distribution?
3. Does the GEV provide a good fit to the observed data?
4. Can you obtain 95% confidence intervals for the parameters?
5. What are the 95% confidence intervals for the 50-year and 10,000-year return levels? Set the argument `alpha = 0.3` in the function `return.level`. What has this done to the intervals? Why?

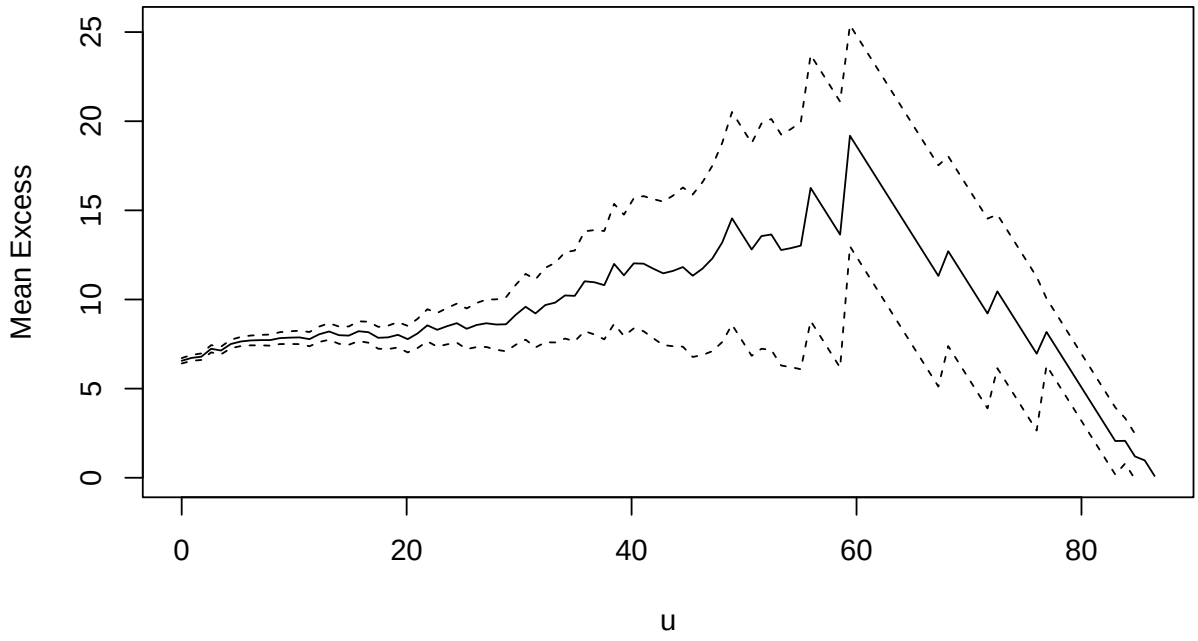
Threshold exceedances

Walkthrough

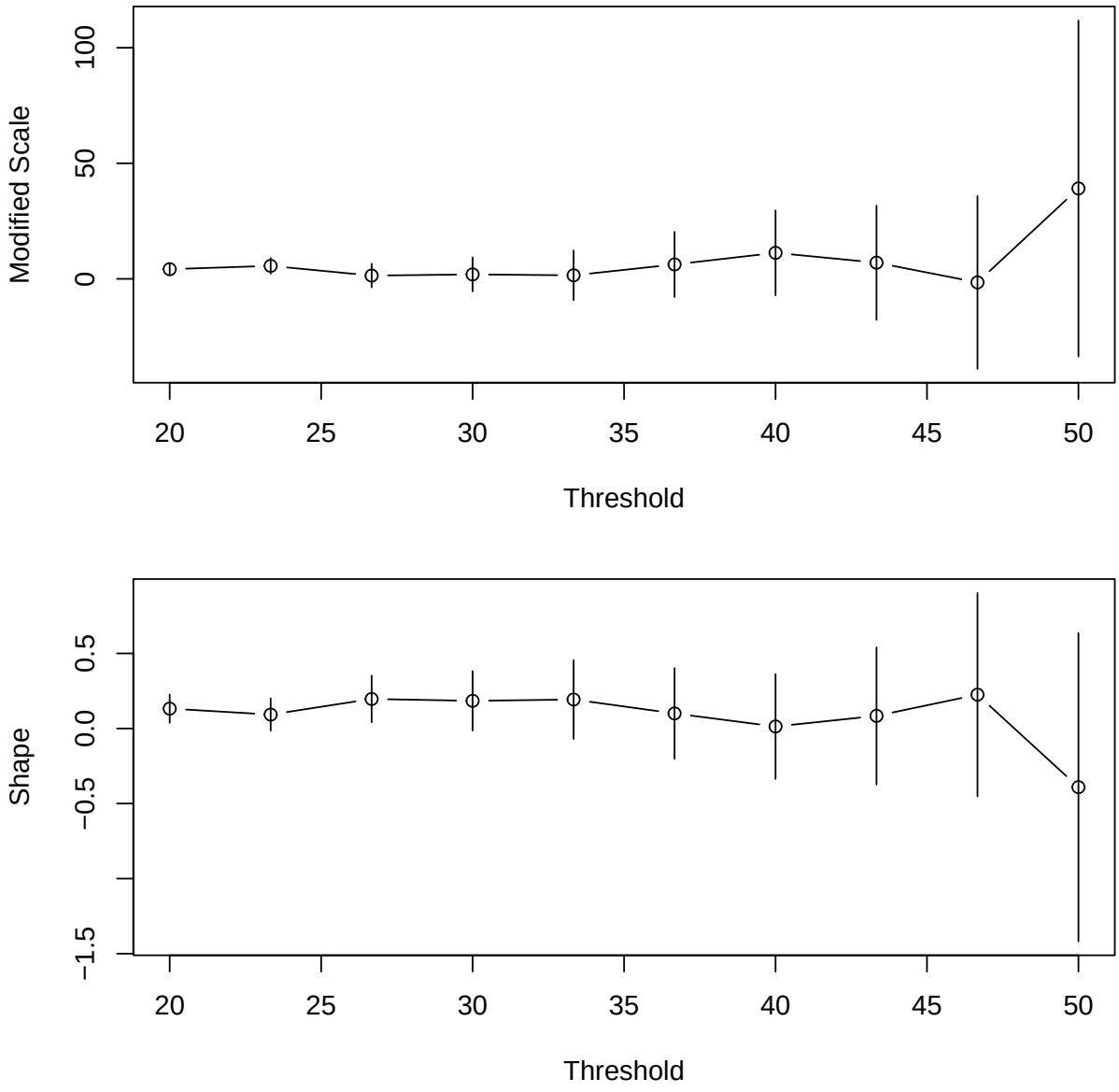
When modelling maxima, much data is discarded by just using the annual maxima; this is an inefficient way of estimating parameters. There may be other large values that are not the maximum but could still provide important information about the distribution of large values. To overcome this problem we now turn to the problem of modelling extremes using threshold exceedance methods. The Generalised Pareto Distribution (GPD) is the main distribution for modelling exceedances above a high threshold (Davison and Smith, 1990).

Two diagnostics for choosing the threshold were given in the lecture, which we apply here for the daily rainfall data contained in `rain`:

```
par(mar = c(4,4,2,1))
mrl.plot(rain)
```




```
par(mar = c(4,4,2,1))
gpd.fitrange(rain, umin = 20, umax = 50)
```



As in Coles (2001) we could choose the threshold $u = 30$ mm:

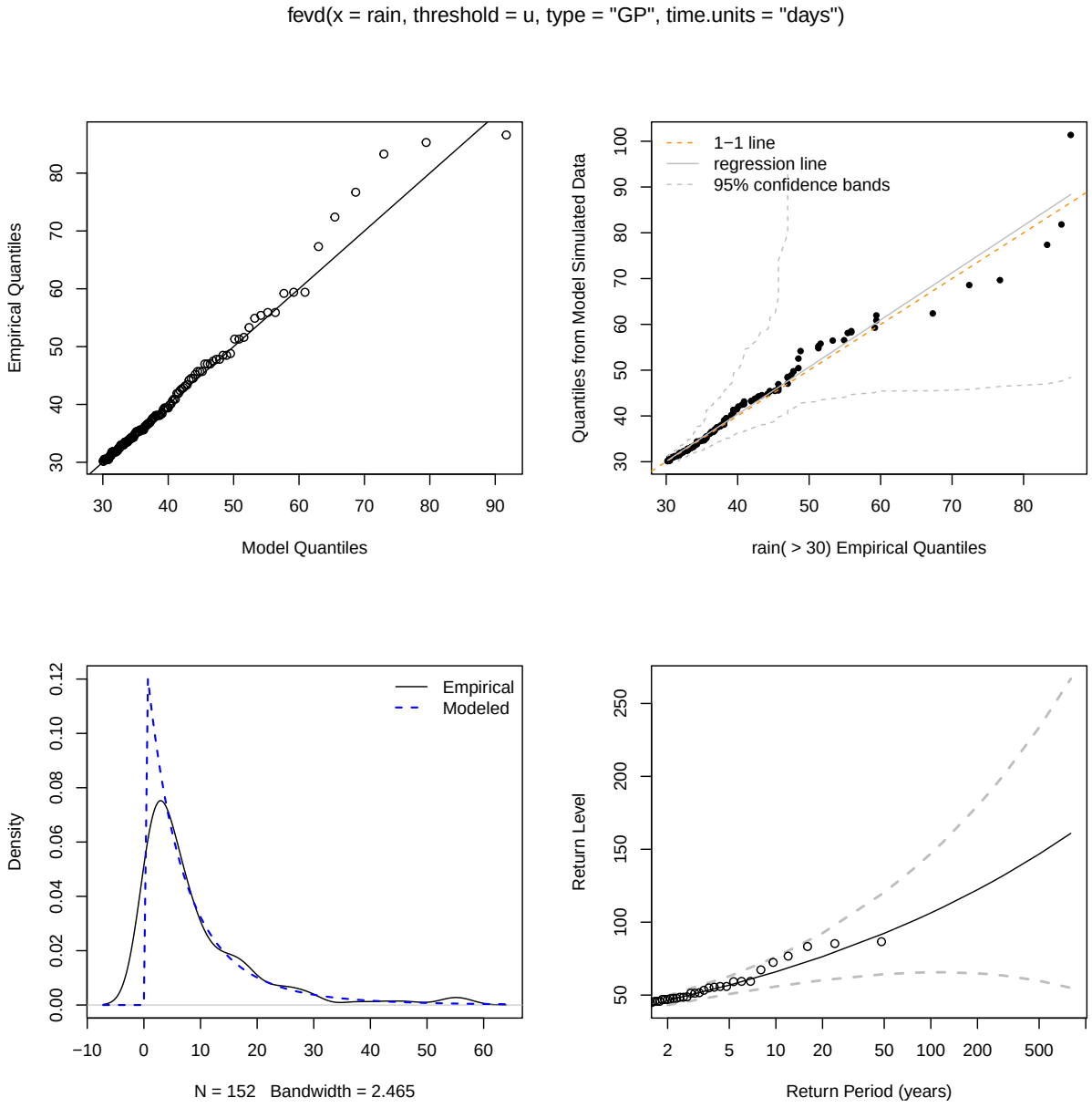
```
u <- 30
```

Now we can fit the GPD using the command:

```
gpd.fit <- fevd(x = rain, threshold = u, type = "GP", time.units = "days")
```

As with the GEV fit, diagnostic plots to assess the fit can be obtained using `plot(gpd.fit)`.

```
plot(gpd.fit)
```



The diagnostics suggest that the GPD is providing an adequate fit to the data. Estimates and confidence intervals for each of the parameters can be obtained in a similar way as above:

```
gpd.fit$results$par
```

```
   scale    shape
7.440252 0.184498
```

```
ci.fevd(x = gpd.fit, alpha = 0.05, type = "parameter")
```

```
fevd(x = rain, threshold = u, type = "GP", time.units = "days")

[1] "Normal Approx."

      95% lower CI Estimate 95% upper CI
scale   5.56158167 7.440252    9.3189227
shape  -0.01385378 0.184498    0.3828497
```

```
gpd.rl <- return.level(x = gpd.fit,
                       return.period = c(10,100,1000,10000),
                       do.ci = TRUE, alpha = 0.05)
gpd.rl
```

```
fevd(x = rain, threshold = u, type = "GP", time.units = "days")

[1] "Normal Approx."

                        95% lower CI  Estimate 95% upper CI
10-year return level        55.89296  65.96142     76.02989
100-year return level       65.62238 106.34231    147.06225
1000-year return level      51.96049 168.09755    284.23461
10000-year return level    -12.13137 262.54099    537.21335
```

Exercises

The choice of threshold is observed to be a very uncertain exercise and subjective. Let's see what happens when the threshold is set at the wrong level.

1. Set the threshold at `u <- 50`. To see how many data points are used in the analysis enter `sum(rain > u)`. What happens to the estimates for the parameters and return levels?
2. Now set the threshold at `u <- 3`. Plot the model fit diagnostics, are there any issues? What proportion of the data are being used in the model fit?

3. You may have noticed when obtaining the confidence intervals above that the lower bound is often negative and thus we are effectively saying that we can have negative rainfall! This occurs as the standard approach to defining confidence intervals is to use the delta method.
4. One approach to remedy this is to use bootstrapping. The parameters and return levels can then be estimated for each resampled dataset.
5. What return levels do you get from typing:

```
gpd.rl.bs <- return.level(x = gpd.fit, return.period = 10000,
                          do.ci = TRUE, alpha = 0.05, method = "boot")
```

6. How does `gpd.rl.bs` differ from `gpd.rl`?