EXTREME VALUE THEORY

Theoretical and practical application based on CD Project Red stock.

Simply explained with python code.

TOMASZ SIATKOWSKI

TEXT AND PRESENTATION IS ENTIRELY HUMAN WRITTEN

EXTREME VALUE THEORY

Imagine you want to estimate the probability or severity of a rare event that has a significant impact on your life. This could be a flood, an earthquake, or simply your investment portfolio. Traditional statistical approaches often struggle with these issues as they focus on patterns that occur much more frequently. However, understanding these extreme behaviors can be crucial for many, especially for banks where extreme losses or gains define outcomes.

The answer to these problems can be found in extreme value theory, which was developed by the brilliant mathematician Ronald Fisher and statistician Leonard Tippett in the late 1920s.

In this presentation, we will explore how we can better handle extreme events in the financial market compared to standard approaches, which we will briefly discuss. We will demonstrate this process using the example of the Polish company CD Projekt Red, which achieved its success by releasing games from the Witcher universe.

(If you want to recreate this yourself, you can use the code found in the rectangles like the one on the right. This code and the entire library were developed by me. But keep in mind you should follow it from the beginning)

```
pip install gev-analysis
```

VALUE AT RISK

If you know what Value-At-Risk (VaR) is, you can move on to the next slide.

VaR is a concept that emerged as a financial risk measure in the 1980s but was formalized in the 1990s by JP Morgan and subsequently widely adopted in global banking markets. In 1996, it was incorporated into the Basel II standards, which means that most banks involved in market risk still use it today, and this will continue until the planned introduction of Basel III with a new measure known as Expected Shortfall.

VaR indicates the minimum amount you can lose in your portfolio over a specified time period with a given probability. For example, if you calculate that your one-day VaR with a 1% confidence level is -4.5%, it means that the total loss from your portfolio will not exceed -4.5% per day in 99 out of 100 days.

Formally, we write this as:

$$P(x \leq VaR) = \alpha$$

where x is, for example, the daily return, VaR} is our value at risk, and alpha is the confidence level. It is important to remember that VaR does not tell us the average size of this loss, which was the reason for its replacement by Expected Shortfall (ES). However, for this work, it will be sufficient.

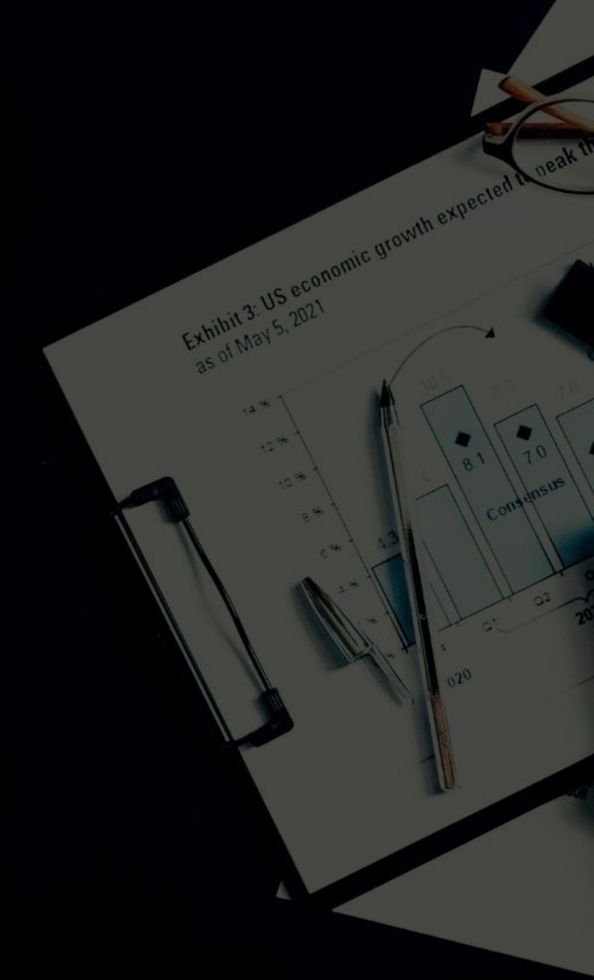
PARAMETRIC VS NONPARAMETRIC APPROACHES

VaR is just a measure and requires us to make several assumptions. First and foremost, we need to decide on the distribution we will work with, as there are two different standard approaches here. We distinguish between:

- Non-parametric methods, which do not require assumptions about the theoretical distribution. Among these, we have empirical distribution functions. This is the simplest way to check how our variable has behaved historically. However, it does not account for data not yet recorded in history.

- Parametric methods, which require assumptions about the theoretical distribution, can reflect such changes but require prior fitting and additional assumptions, e.g., that the distribution is constant over time and does not change (which is rarely true since financial data have periods of higher/lower volatility). In finance, common distributions for returns include the normal distribution, t-Student distribution, Laplace distribution, or skewed distributions.

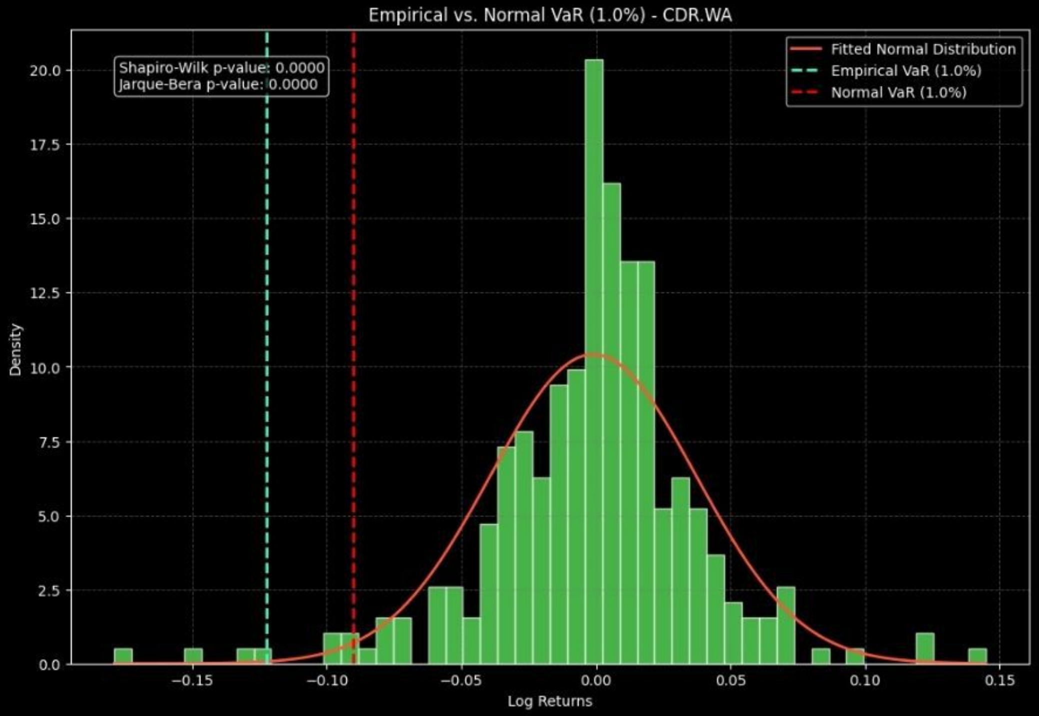
Moreover, in standard parametric approaches, there is often another problem. Fittings performed, for example, using Maximum Likelihood Estimation, focus on minimizing errors for the entire sample, which often leads to misfits in the tails, i.e., situations potentially most important for risk managers. Similarly, the method of moments focuses only on central moments ommiting concept of extremes.



REAL LIFE PROBLEM

PRACTICAL ISSUE

As promised, let's take CD Projekt Red as an example. Below, we see the distribution of logarithmic returns for the period from January 1, 2020, to March 31, 2021. This was a stressful period due to COVID-19, making market risk management particularly important



The chart shows the empirical distribution and the fitted normal distribution using the MLE method. At the same time, their VaRs are marked. From the shape alone, it can be seen that our empirical distribution does not belong to the normal distribution. It exhibits fat tails and a peaked top. This is also confirmed by the Shapiro-Wilk and Jarque-Bera tests, which reject our null hypothesis of normality.

This highlights a problem that affects practically the entire equity market, where our empirical VaR is significantly further from the center of the distribution than would be expected from our statistical models, in this case, the normal distribution. This is very dangerous and threatens improper risk estimation.

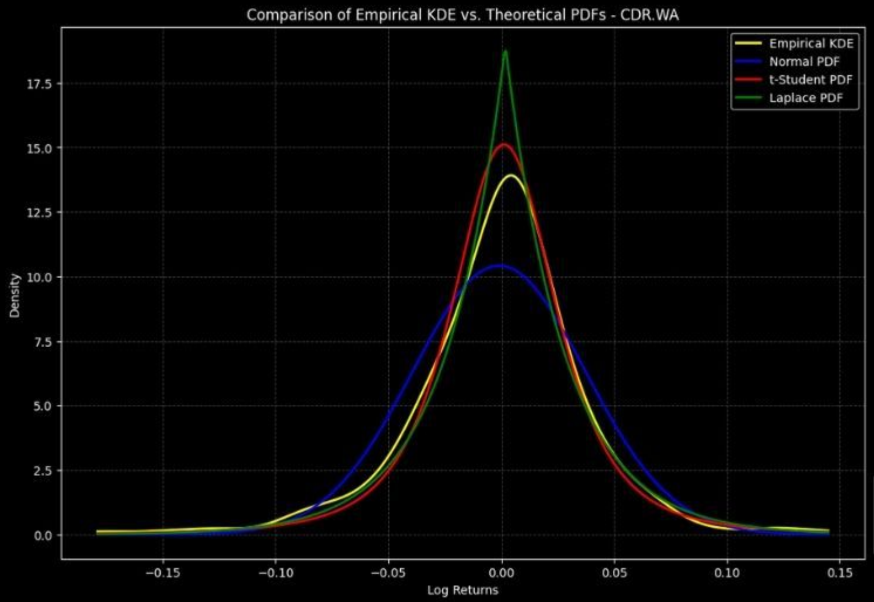
```
from gev_analysis import NormalityVaR, GPDExtremeValueAnalysis, GEVExtremeValueAnalysis
normality = NormalityVaR("CDR.WA", "2020-02-01", "2021-03-31", 0.01) #CDR.WA is ticker, 0.01 is alfa
normality.plot_results()
```

PRACTICAL ISSUE

Let's take it a step further and see how it looks for other distributions. In the table on the right, we have added distributions with slightly fatter tails - the t-Student and Laplace distributions. All of these are compared with the empirical distribution in the form of a kernel density function. It is shown that none of them can accurately measure the risk that occurred in reality, as we can see by comparing the VaR.

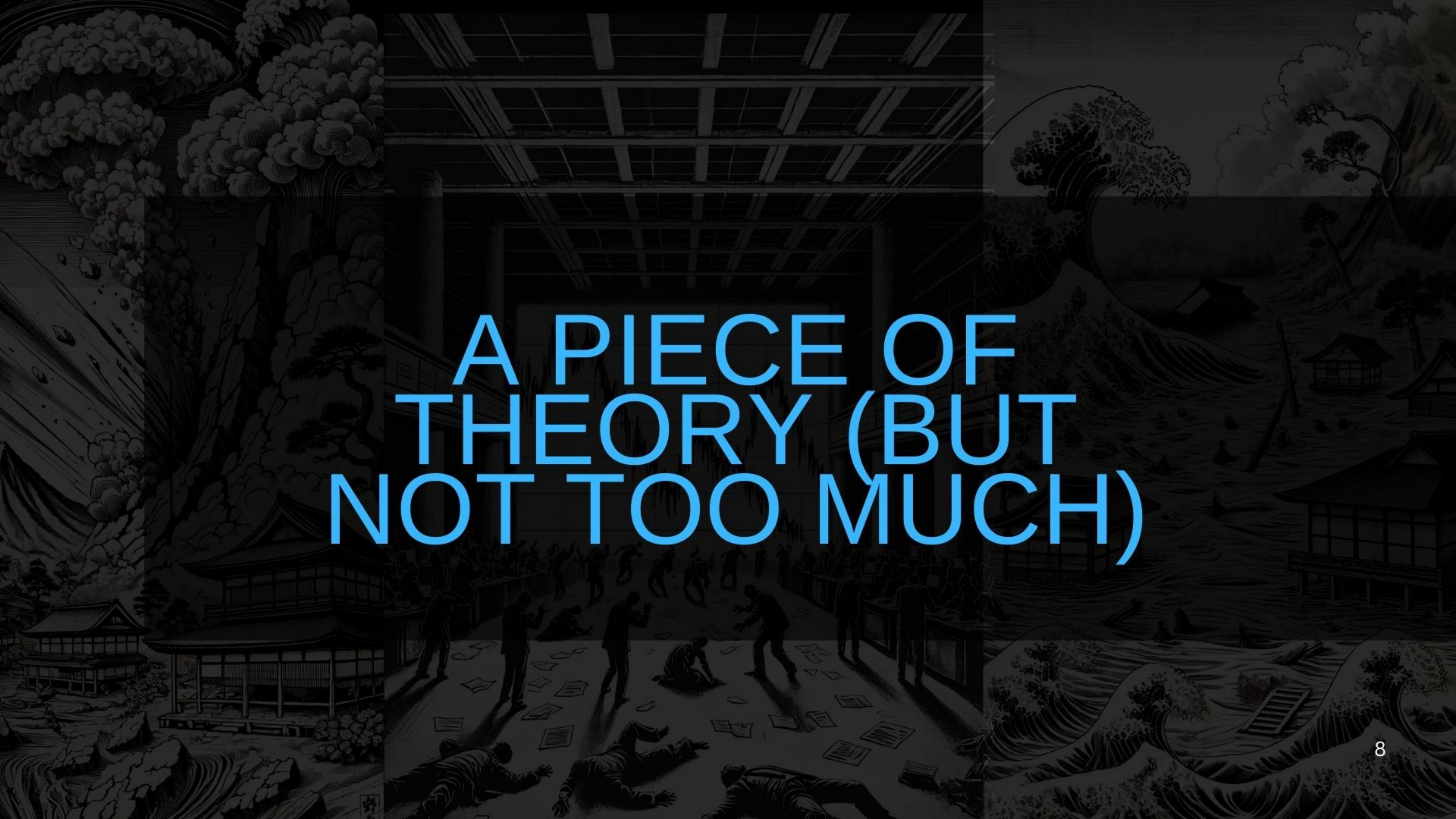
Distribution	Param1	Param2	Kolm.-Smirn. p-value	VaR 1%
Normal	-0.0010	0.0383	0.0155	-0.0901
T-Student	2.8791 (ν)	0.0242 (σ), 0.0012 (μ)	0.5121	-0.1127
Laplace	0.0018	0.0264	0.7808	-0.1015
Empirical				-0.1222

normality.distribution_summary()



On the left side, we compare our distributions in the form of a Probability Density Function (PDF).

normality.plot_distributions()



A PIECE OF THEORY (BUT NOT TOO MUCH)

This is indeed an ideal example to use for Extreme Value Theory (EVT). Let's delve into the theory a bit. EVT is to extremes what the Central Limit Theorem (CLT) is to means or sums. According to EVT, the extremes of a function can converge to two distributions: the Generalized Extreme Value (GEV) distribution or the Generalized Pareto Distribution (GPD), regardless of the original data distribution. The difference between these distributions comes from the methods of determining extremes.

EVT

GEV

For GEV distributions, extremes are determined using blocks. This means that we divide our time-dependent variable into equally defined blocks, from which the largest (or smallest) value is considered an extreme - maximum (or minimum).

In this way, the identified extremes (preferably standardized beforehand) will converge to one of the three possible distributions that together form the GEV distribution, which we will discuss in more detail on the next slide.

GPD

For GPD distributions, extremes are determined using a Peak-Over-Threshold method. Threshold is predetermined to identify values that are greater (or smaller) and considered as extremes.

Here, it is also necessary to standardize these extremes beforehand, but instead of subtracting the means, we subtract the threshold value. This creates an excess distribution, which will also converge to one of the three possible distributions from the GPD distribution.

GEV

The GEV distributions are:

- Gumbel (Type 1)
- Fréchet (Type 2)
- Weibull (Type 3)

What differentiates them is the behavior of the tail of our distribution. This is best illustrated using the formulas for the Cumulative Distribution Function (CDF):

$$F(x,\xi,\mu,\sigma) = \begin{cases} \exp\left(-\exp\left(-\frac{x-\mu}{\sigma}\right)\right), & \text{if } \xi = 0, \\ \exp\left(-\left[1+\xi\frac{x-\mu}{\sigma}\right]^{-\frac{1}{\xi}}\right), & \text{if } \xi \neq 0. \end{cases}$$

Where ξ is the tail parameter, μ is the location parameter, and σ is the scale parameter. Depending on the parameter, we will be talking about a different distribution, which we can see in the table on the right.

Knowing the parameters, we can calculate VaR using simplified formulas provided on the right side below the table for the significance α level.

ξ (Shape Parameter)	Tail Behavior
ξ>0	Heavy tails (Fréchet): The distribution has heavy tails. Extreme events are more probable (e.g., power-law behavior).
ξ=0	Exponential tails (Gumbel): The distribution has medium tails. Extreme events decay exponentially (e.g., normal-like).
ξ<0	Bounded tails (Weibull): The distribution is bounded.

$$\mathrm{VaR}(\xi,\mu,\sigma,\alpha) = \begin{cases} \mu + \frac{\sigma}{\xi}\left[(-\log(\alpha))^{-\xi} - 1\right], & \text{if } \xi \neq 0, \\ \mu - \sigma\log(-\log(\alpha)), & \text{if } \xi = 0. \end{cases}$$

GPD

The GPD distributions are:

- Pareto
- Exponential

The situation is similar here in terms of tails. Below is the CDF:

$$G_{\xi,\beta}(y) = \begin{cases} 1 - \left(1 + \frac{\xi}{\beta_\mu} y\right)^{-\frac{1}{\xi}}, & \text{if } \xi \neq 0, \\ 1 - \exp\left(-\frac{y}{\beta_\mu}\right), & \text{if } \xi = 0. \end{cases}$$

Where ξ is the tail parameter, μ is the threshold value, β is the scale parameter, and y is the standardized (by the threshold value) extreme. The issue of the tail parameter is identical to that in GEV.

Knowing the parameters, we can again calculate VaR using the inverse CDF functions derived from the formula on the right below the table.

ξ (Shape Parameter)	Tail Behavior
ξ>0	Heavy tails (Pareto)
ξ=0	Exponential tails (Gumbel): The distribution has medium tails. Extreme events decay exponentially (e.g., normal-like).
ξ<0	Bounded tails

$$\mathrm{VaR}_{\xi,\beta} = u + \frac{\beta}{\xi}\left[(\alpha\frac{N}{N_u})^{-\xi} - 1\right]$$

Here, N is the total number of returns, while N_u is the number of returns above the threshold.

PARAMETER SELCTION PROCESS

PRACTICAL APPLICATION

As we can see, both methods behave in a very similar way, and therefore the code will also be analogous. The biggest problem we will face is the correct selection of parameters: block size (for GEV) or the appropriate threshold (for GPD). This issue is usually highly subjective and can significantly affect our VaR results, which we will demonstrate shortly.

Let’s start with the code:

```
#Here, We first define our analysis objects for both distributions. At this point we don’t need provide significant treshold level or block size parameter - it should be random value (in my case is -0.04, and 15) , because in next step we will be selecting those parameters, so now they are only temporary)
gpd_analysis = GPDExtremeValueAnalysis(symbol=”CDR.WA”, start="2020-01-01", end="2021-03-31", alpha_input=0.01, threshold=-0.04)
extreme_analysis = GEVExtremeValueAnalysis(symbol = “CDR.WA”, start="2020-01-01", end="2021-03-31", alpha=0.01, block_size=15)
```

GEV

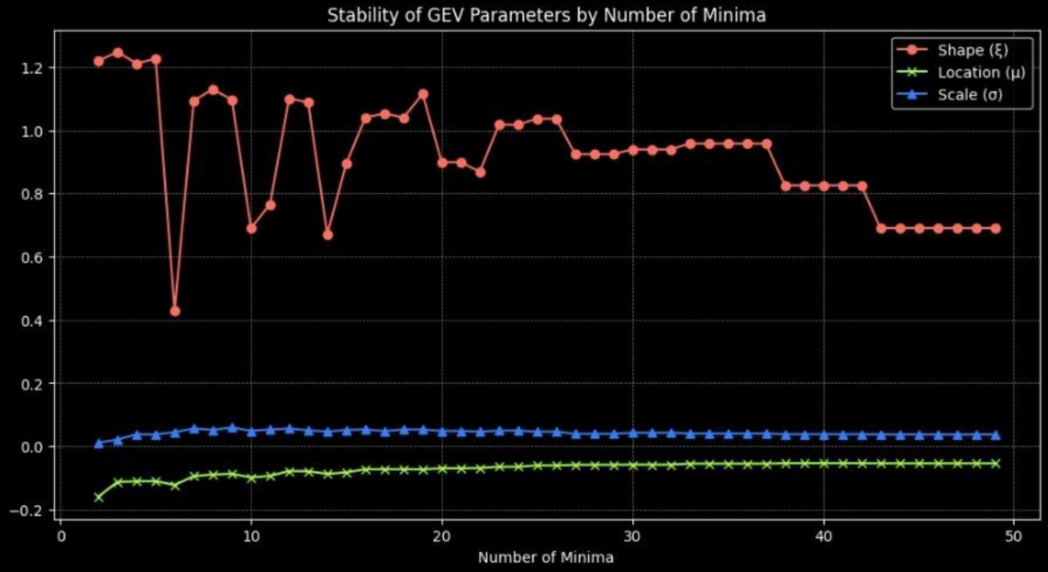
For GEV distributions, we need to correctly identify how many days fit into one block, i.e., the block size. The number of such days automatically affects the number of minima. For example, if we have 300 days and want to have 30 minima, one block will contain 10 days (300:30). Unfortunately, in this method, we do not have full control over the minima, as we automatically select the smallest (or largest) values for a given block. Since markets have periods of rises and falls that often cluster, there may be situations where the minima (or maxima) are positive (or negative) values, which means we should remain extra vigilant. From experience, I can say that the best point is where our parameters, including VaR, stabilize, which will be shown in the following slides.

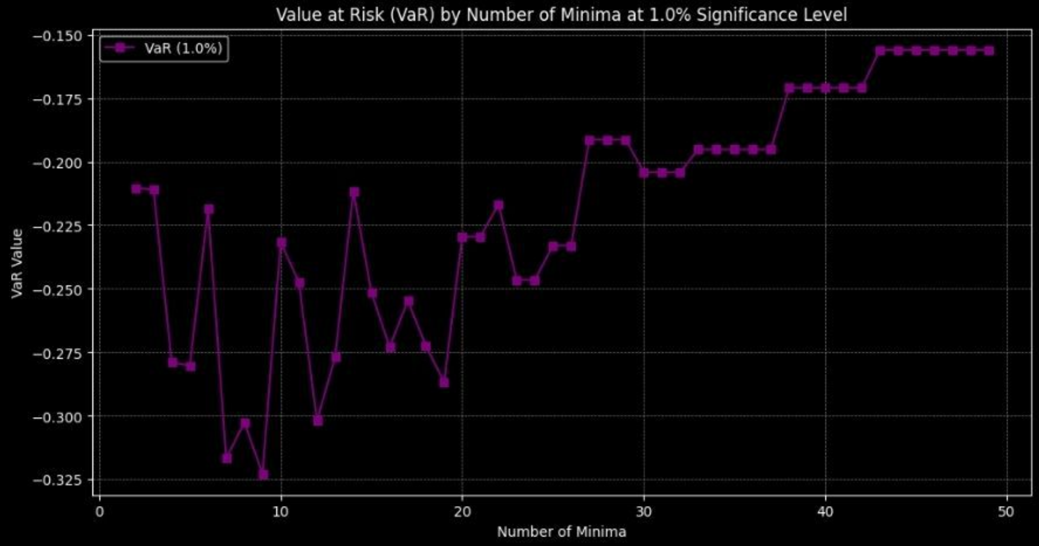
GPD

For GPD distributions, the threshold also presents a significant challenge. The threshold we choose greatly affects the appropriate VaR value, much more so than block sizes. There is no single best method for selecting it. However, there are several guidelines that are good to follow. The main and most common one is to choose a threshold where the mean of all exceedances (i.e., our returns exceeding the threshold) significantly stabilizes. It is also good to choose thresholds where the tail parameter is stable and, to be consistent with the theory, less than 1. This applies to both distributions (if it is greater, then theoretically such a distribution has infinite variance, and we cannot calculate and interpret VaR).

GEV

```
#Parameter stability
extreme_analysis.stability(range(2, 50))
```

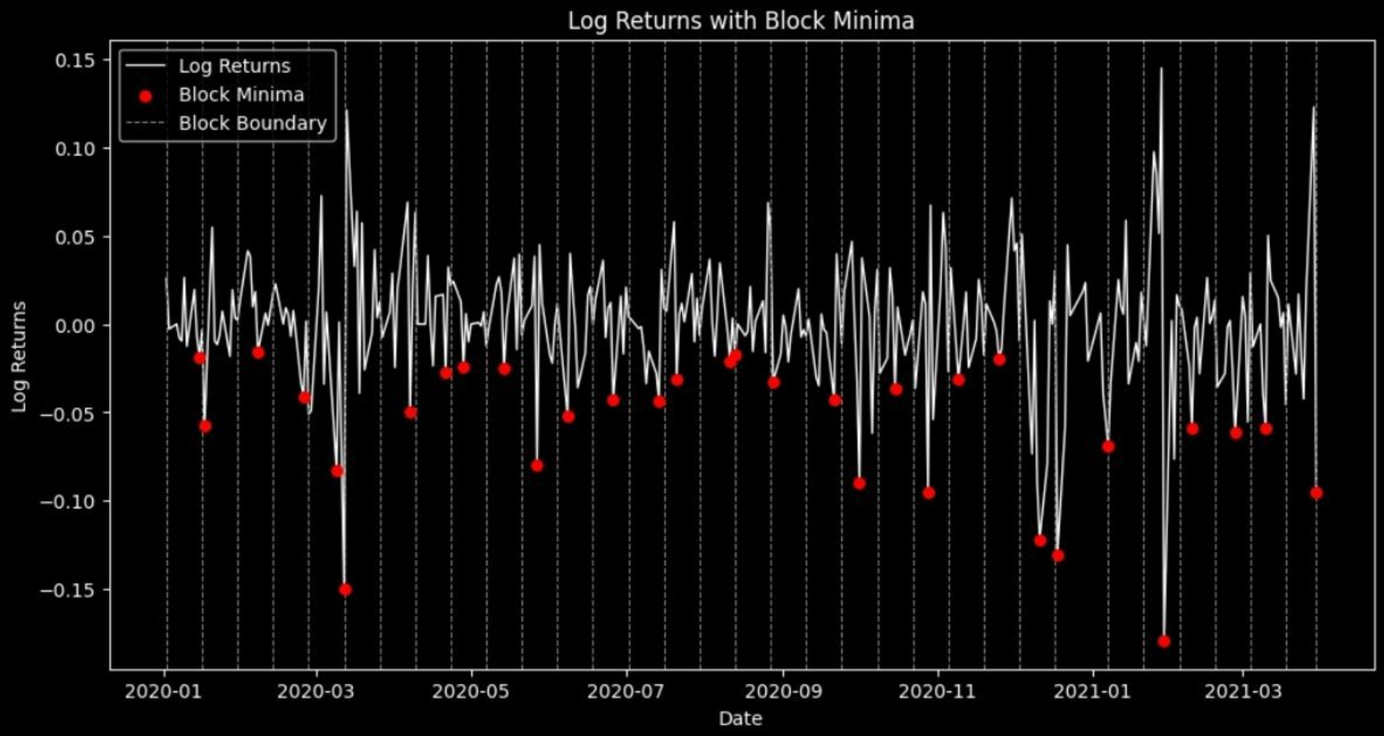




The charts show the change in parameters based on the number of minima. We see that in the case of GEV, the tail parameter is the most variable, which is a natural phenomenon. The first stabilization period occurs when the number of minima exceeds 20. However, as we mentioned, we aim to find stable periods for tail parameters below 1. In this case, it is around 30 minima, which, considering the total number of observations at 296, gives us approximately 10 days in each block (296:30).

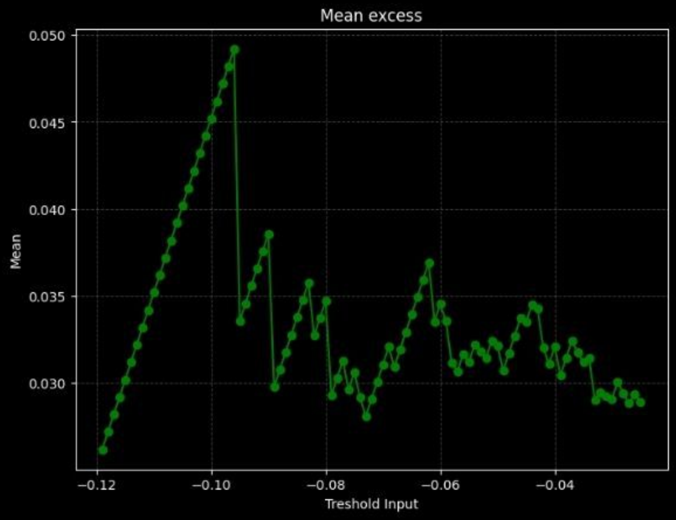
Having the parameters, we can also observe the VaR values and see that it stabilizes for the same number of minima. For our chosen number of 30 minima, it will be approximately -0.19 (-19%). Some might say that this is very high and not possible and probably is right, but the next slide will show that these values are not necessarily unattainable.

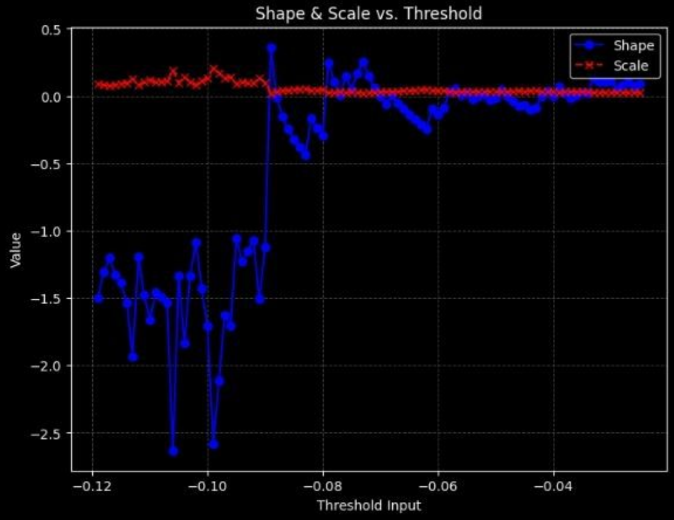
```python
#We define our analysis once again and changing block size to chosen “10".
extreme_analysis = GEVExtremeValueAnalysis(“CDR.WA”, "2020-01-01", "2021-03-31", 0.01, 10)
extreme_analysis.plot_minima()
```

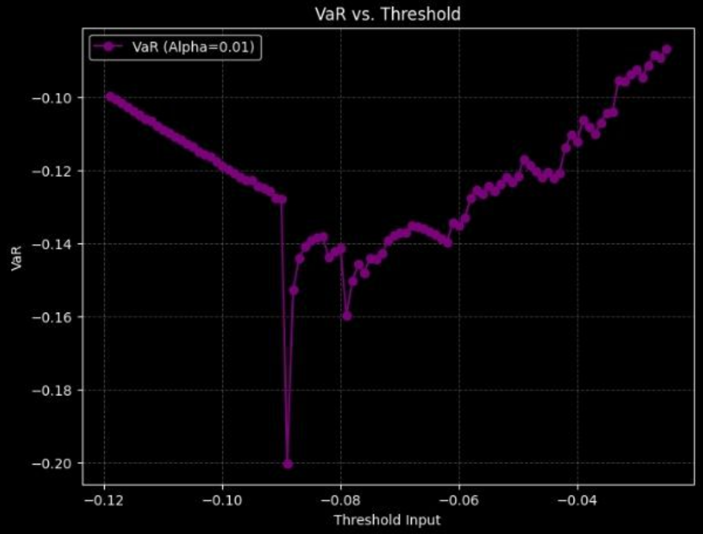


This chart beautifully illustrates our minima (in red) and blocks on the return chart (dashed line). It is very important for our analysis, as we must strive to ensure that the red dots are as low as possible. Remember, sometimes it isn’t possible to exclude all such extremes, for example, for the period between July and October 2020. In this case, it is sufficiently good since most of the minima actually reflect returns at a level of at least -5%. It is also worth noting that returns at around -15% occurred at least twice in our period (one is even -17.5%), which does not significantly deviate from our -19% for 296 observations and a 1% significance level.

GPD







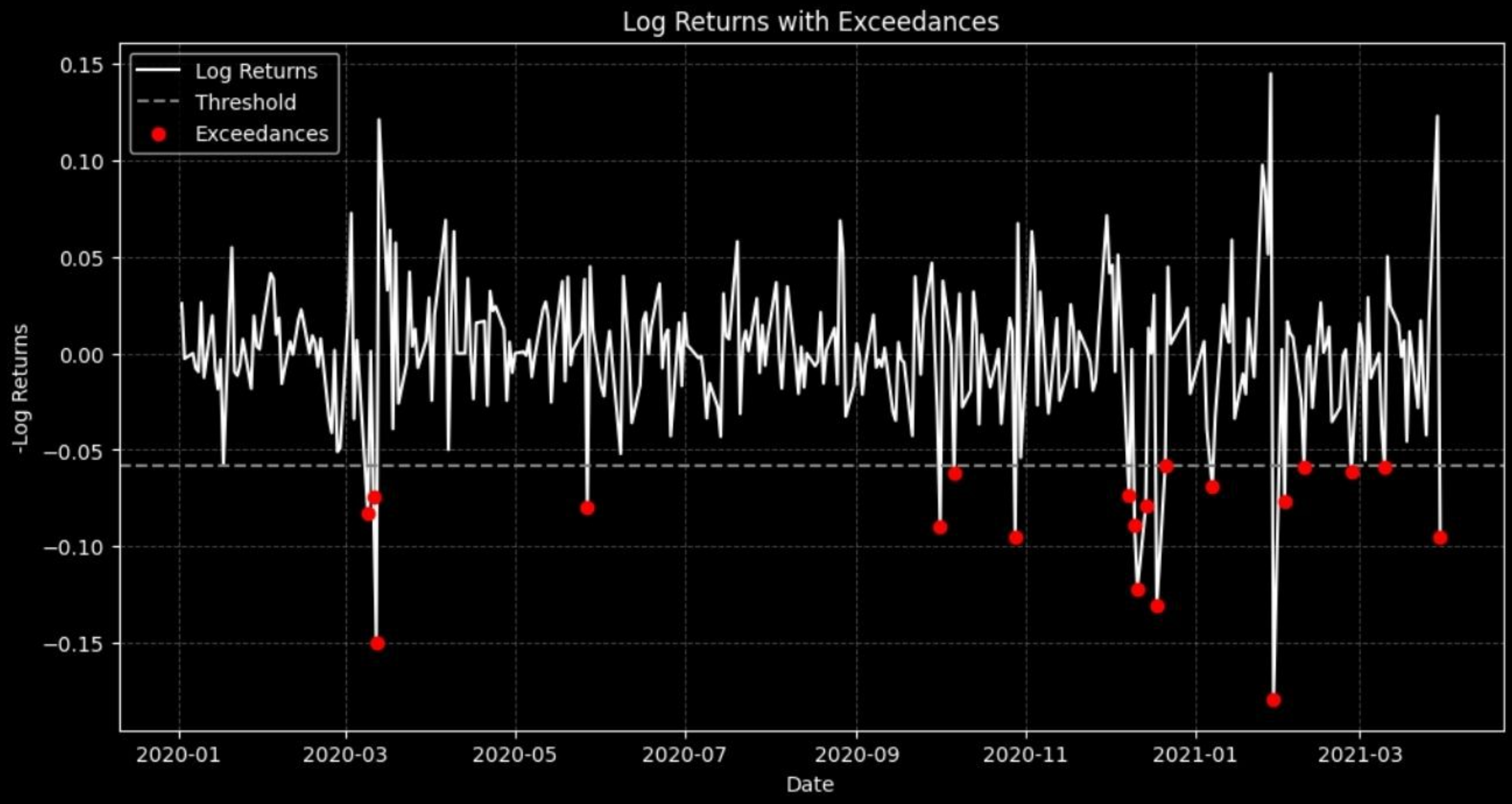
Here, we also have charts showing the behavior of our parameters depending on the threshold level for Peak Over Treshold method (GPD). The first threshold at which the mean excess stabilizes (subjectively) is at approximately -0.058 (-5.8%).

A similar trend is observed for the scale and shape parameters.

For our threshold the VaR value is approximately -0.13 (-13%). It's important to note that there is a significant range between other possible scenarios. If someone chooses a threshold level of -0.08, they might get a VaR that is 3 percentage points lower, around -16%. Conversely, if the threshold is -0.03, the VaR could be around -10%. This is the biggest drawback of EVT.

```
#Parameter stability for GDP.
import numpy as np
thr_list = np.arange(-0.025, -0.12, -0.001)
gpd_analysis.stability(thr_list)
```

```
#We define our analysis once again and changing threshold for -0.058
gpd_analysis = GPDExtremeValueAnalysis(symbol="CDR.WA", start="2020-01-01", end="2021-03-31", alpha_input=0.01, threshold=-0.058)
gpd_analysis.plot_minima()
```



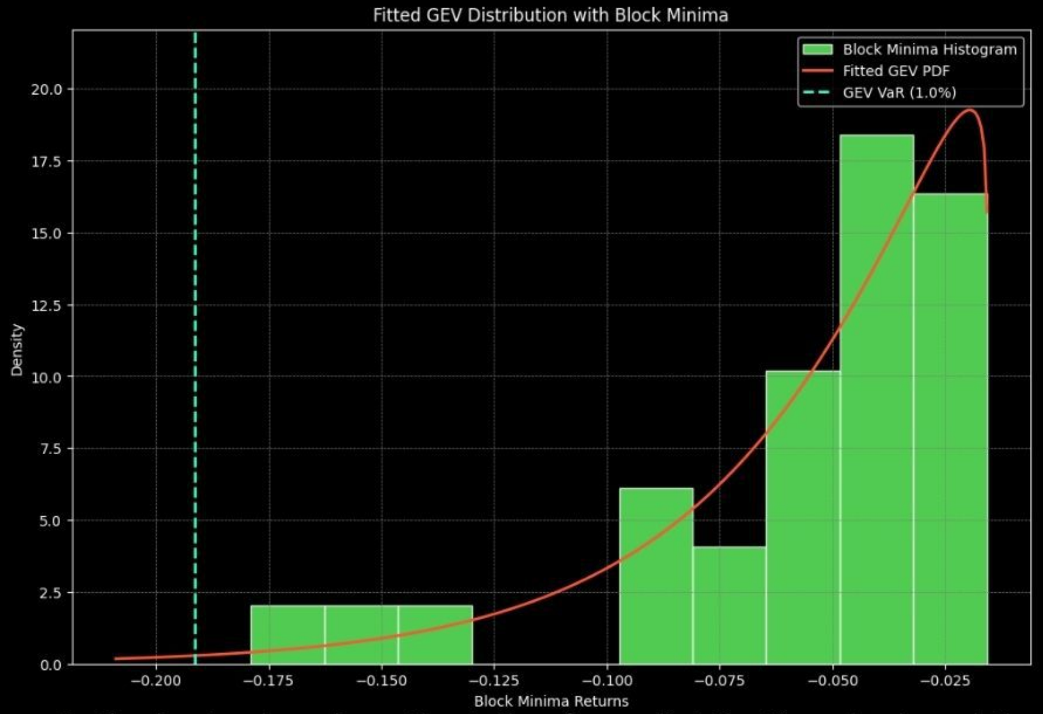
With a threshold set at -0.058, we identify approximately 20 instances where this level has been exceeded. This threshold appears to achieve an optimal balance between the total number of observations (296) and the number of extreme events. Such a selection ensures sufficient data to analyze extremes while minimizing biases that could result from either an insufficient or excessive number of extreme cases.

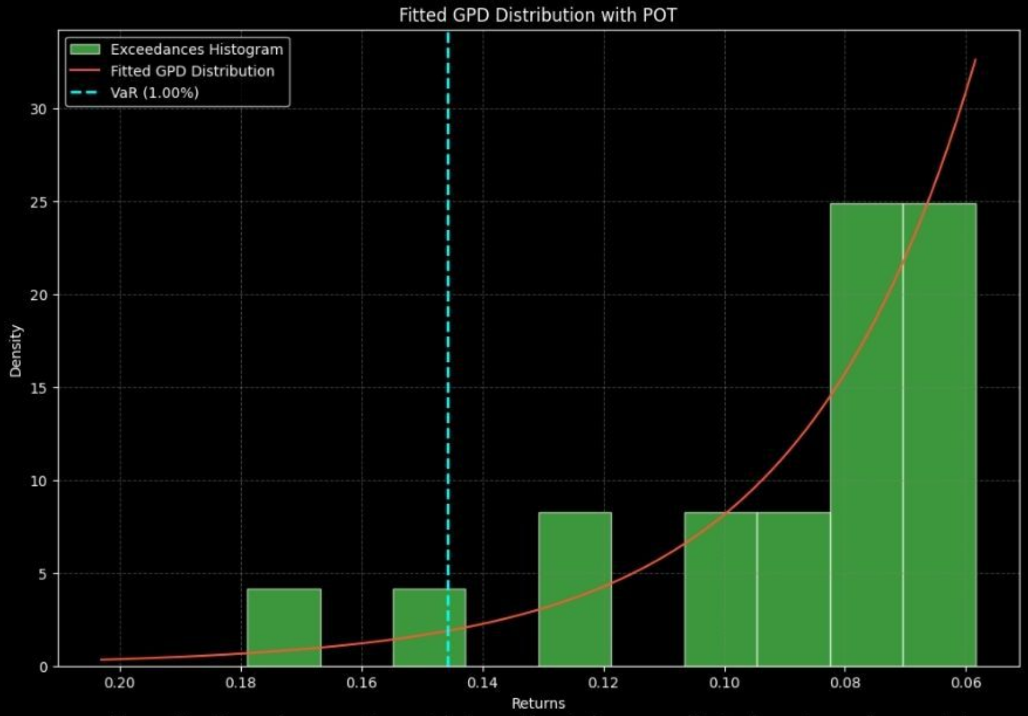
RESULTS AND COMPARISON

#Fitted distribution
extreme_analysis.plot_distribution()

COMPARISON GEV AND GPD

#Fitted distribution
gpd_analysis.plot_distribution()





Both charts show how the tails of our distributions look and the corresponding VaR values. One thing that immediately stands out is how the VaR from the block method is much higher than all the others, including those from classic distributions. This is a common issue and, in my opinion, happens because the block method struggles to properly capture true extremes. It often lumps together small extremes with the really big ones, which leads to inflated results. From my experience, when there aren't big gaps between extremes, the GEV distribution tends to give more realistic VaR values. But in this case, the block method ends up with VaR values that go beyond anything we've actually seen. On the other hand, the POT method seems to handle this much better, giving VaR values that make a lot more sense.

#Summary GEV (Block Min)
extreme_analysis.summary()

COMPARISON GEV AND GPD

#Summary GPD (POT)
gpd_analysis.summary()

Parameter	Value
Shape (ξ)	0.924295
Location (μ)	-0.058437
Scale (σ)	0.039584
Number of Minima	30
Total Observations	296
VaR (1.0%)	-19.12%
Distribution Type	GEV (Frechet -Heavy Tails)

Parameter	Value
Shape (ξ)	0.030953
Threshold	-0.058
Scale (σ)	0.030214
Number of Exceedances	20
Total Observations	296
VaR (1.0%)	-14.56%
Distribution Type	GPD (Pareto-type - Heavy Tails)

If we had to choose between the two methods, the POT method would likely be considered more reliable in terms of VaR values, despite the relatively greater stability of parameters on the block minima side. However, it's important to remember that no model is perfect, and each will always carry the risk of errors. Therefore, it's good practice to look at several models and draw conclusions based on their weaknesses and shortcomings, while also appreciating their strengths. In this case, the POT method provides a the best estimation of extremes in comparison to standard parametric approaches and even Block Minima method which we can nicely illustrate on the QQ chart on the next slide.

^for POT method charts are reversed as there was some special trick that had to be applied to correctly caputure minima and caluclate VaR, but results are correct., so only thing that changes is perppective we look at tails

COMPARISON GEV AND GPD

#QQ plot for block min GEV

```
extreme_analysis.qq_plot()
```

#QQ plot for exceedences POT

```
gpd_analysis.qq_plot()
```