Risk Quantification for Tech Stocks

Final report for continuous assessment

Iker Caballero

Risk Quantification - Prof. Catalina Bolancé and Prof. Luis Ortiz-Gracia

1 Introduction and Data

The main goal of this project is to use different techniques to quantify the inherent risk in a portfolio composed of three stocks using different methods. The whole analysis will be based on a portfolio composed of three tech stocks (stocks from the technological industry), these are Netflix (NFLX), Alibaba (BABA), and Spotify (SPOT). The technological industry has been growing very fast during the last decade both economically and financially, being a very important portion of many institutions' portfolios. However, its fast industrial development and the quantity invested in this industry carry financial risks that could be quantified.

In order to carry on the quantification of risks we need to work with data on the prices of each stock. We select to work with a window from the 17th of September 2018 to the 17th of September 2020. There is no special reason for the day and the month, but the period from 2018 to 2020 has been selected due to the COVID-19 outbreak and its effects on the financial market. By choosing this window, we are able to obtain prices that reflect the situation before and after the pandemic, which could be interested in our analysis (especially for the extreme value distribution analysis).

This report will start by trying to estimate the VaR and the TVaR using different methods, carry on factor analysis and finish by studying the extreme value distribution. All this has been done through an R code which will be available in the appendix.

2 Estimation of Value-at-Risk and Tail Value-at-Risk

In this section we aim to estimate risk metrics as the Value-at-Risk and the Tail Value-at-Risk through different commonly used methodologies in risk management: the variance-covariance method, the historical simulation method and the Monte Carlo method.

We will present results for each one of the methods an compare them at the end of the section.

2.1 Variance-Covariance Method

The Variance-Covariance method is based on the assumption that the changes in risk factors (in this case, the stock returns) follow a normal or a t-Student distribution, so that linear approximations for the losses will follow the same distribution but with different parameters. This way we can easily estimate the distribution function for the linear approximation losses and obtain risk metrics such as the VaR, the TVaR or the ES. We are specified to assume the the linearized losses follow a normal and a t-Student distribution (using different degrees of freedom), so we will need to compute the expected value and the variance using the adequate formulae for each distribution.

The first step is to estimate the expected value and the variance-covariance matrix for the changes in the risk factors. Because we are dealing with a stocks portfolio, we will use the log-returns as the changes in risk factors. We can take a look at the distributions for the log-returns in Figure 1, showing that they seem to follow a log-normal distribution (a well-known empirical fact).

Hence, we need first to estimate their expected value and their variance-covariance matrix, with the results shown in the next tables:

Stocks:	**NFLX**	**BABA**	**SPOT**
Sample mean:	0.06381%	0.10723%	0.06438%

Table 1: Sample means for the portfolio stocks

Table 1 pictures the different values of the sample mean (the unbiased estimator for the expected value) for the log returns, and allows to see how the returns for AliBaba have been higher in average.

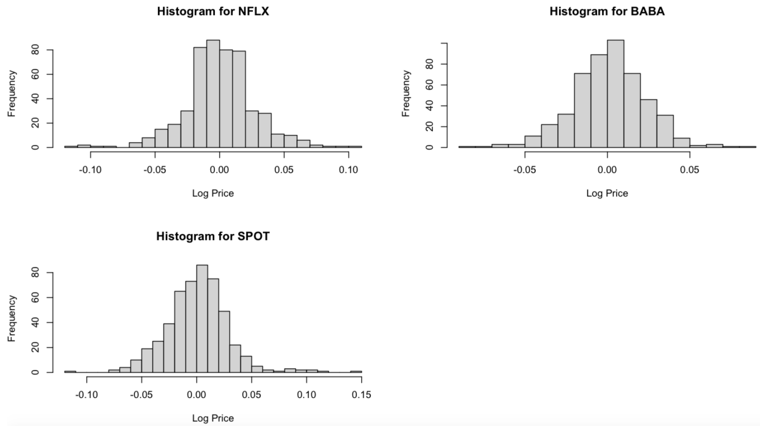

Figure 1: Histograms for the three stocks in our portfolio

Table 2 is the estimated variance-covariance matrix:

Stocks:	**NFLX**	**BABA**	**SPOT**
NFLX	0.0007547	0.0003068	0.0003353
BABA	0.0003068	0.0005008	0.0002801
SPOT	0.0003353	0.0002801	0.0008057

Table 2: Variance-covariance matrix for the three stocks

Looking at the matrix, one can spot that the most volatile stock is Spotify's, even though Netflix is quite near, and that the most correlated stocks are NFLX and SPOT, which is logical as they are both from the US. Finally, we need the weights of each stock at our portfolio at the end of time window considered for the stocks' data. These are obtained from a previous computation, which assumes a functional form for the loss function. These weights are summarized in Table 3, and supposing a portfolio value equal to 1 (in order to simplify computations), we can obtain the expected value and the variance for the linearized losses, shown in Table 4.

Stocks:	**NFLX**	**BABA**	**SPOT**
Weights:	48.26%	26.86%	24.8%

Table 3: Weights of each stock in the portfolio

Statistics:	**Mean**	**Variance**
Values:	-0.07557%	0.0004590469

Table 4: Estimates for the linearized losses

Finally, we can use these parameters to estimate the VaR and the TVaR at a 99% level for both the normal and the t-Student distribution. The values are shown in Table 5:

Stocks:	**Normal**	**t-Student** ($v = 3$)	**t-Student** ($v = 4$)	**t-Student** ($v = 5$)
VaR:	0.04908713	0.05541251	0.05601061	0.05508872
TVaR:	0.05634747	0.08587201	0.07833626	0.07313688

Table 5: VaR and TVaR for different distributions

One can easily observe how the results change depending on the distributions. For the normal distribution case, there is a 1% probability of having losses higher than 0.04908713, while for the t-Student distributions the value is higher. This is due to their tails being thicker than the normal distribution's. Moreover, this pattern can be also seen for the TVaR, being greater whenever the degrees of freedom are less (because the tails are thicker in that case).

2.2 Historical Simulation Method

The second method we use is the historical simulation, which basically consists in working directly with the linear loss function and the data by fitting a given distribution and obtaining the desired risk measures through this fitted distribution. In our case, we are asked to assume both a normal and a t-Student distribution for the linearized losses, so we fit both distributions to the data.

To do so, we first need to compute what would be the losses of the portfolio (the historical simulation comes from this), which is obtained by multiplying the value of the portfolio by the product between the log-returns and the weights of each stock. Some relevant statistics for this data is shown in Table 6, and the distribution (the histogram) is plotted in Figure 2.

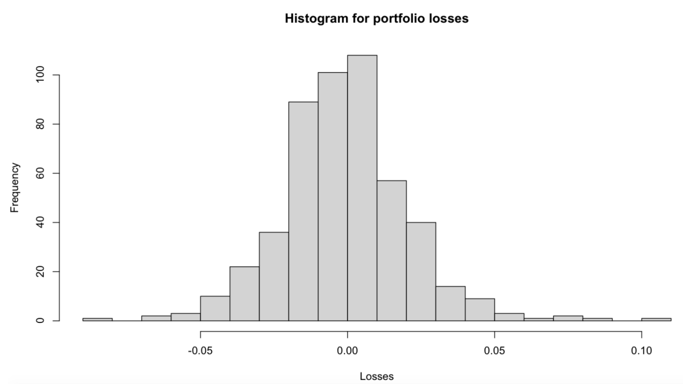

Figure 2: Histogram for the portfolio losses in the historical simulation

Statistics:	**Min**	**Mean**	**Median**	**Max**
Values:	-0.0837096	-0.0007558	-0.0013334	0.1066246

Table 6: Statistics for the portfolio linearized losses

These losses seem to follow a normal or a t-Student distribution, and we can highlight the fact that, in average, we would make a little profit in the ssimulation (as the loss is negative, which is translated as a profit). Then, we fit the data to the distributions and now we can obtain both the VaR and the TVaR for each distribution, which appear in Table 7 with the AIC values for each model.

Stocks:	**Normal**	**t-Student** ($v = 4$)	**t-Student** ($v = 4.95$)	**t-Student** ($v = 4$)
AIC:	-2420.243	-2445.764	-	-
VaR:	0.04908703	0.06401426	0.05782425	0.05762217
TVaR:	0.05634736	0.08867184	0.07621006	0.07581871

Table 7: VaR and TVaR for different distributions

As we can see from the AIC values, the fit is better for the t-Student distribution with 4.95 degrees of freedom than for the normal distribution. Moreover, we can see how, for the fitted values, the VaR do not differs, while it differs relatively more when we use 4 degrees of freedom instead of the 4.95

obtained. When it comes to the TVaR, we see the same pattern as before: it is bigger for t-Student distributions because of their tails' nature and it is reduced when augmenting the degrees of freedom. All in all, the results are similar as with the previous method, even thogh there are differences, so the interpretation of the results is merely the same.

2.3 Monte Carlo Method

The last method is the Monte Carlo method, and it is one of the most used methods in financial practice. In this method, one assumes some distribution for the historical performance of the risk factors data (the log-returns) and generates random sequences for the evolution of this factors in order to compute the losses at each one of these random sequences and finally obtain a distribution of portfolio losses that allows to obtain the desired risk metrics.

For this part we must use different models for multivariate behaviour for the changes in the log-returns, so we divide the analysis depending on the model that we use.

2.3.1 Multivariate Normal and t-Student distributions

The first type of models we deal with are the once seen in the previous sections: the normal and the t-Student distributions. In order to carry on the simulation, we generate one million observations of a multivariate normal and a for the multivariate t-Student distribution with parameters equal to the estimated vector of means and variance-covariance matrix from previous computations.

Then, we compute the linearized losses of the portfolio in the usual way, so with all the data for the simulated linearized losses we can obtain the distribution of linear losses and compute both the VaR and the TVaR, represented in Table 8. We remark that, even if known that the correlation between the assets is high, we also show the risk measures of the same simulations but assuming independence, so we can use these values as benchmark.

Stocks:	**Normal**	**Normal (indep)**	**t-Student**	**t-Student (indep)**
VaR:	0.0490139	0.03836253	0.05332425	0.03832312
TVaR:	0.05625097	0.04387404	0.07424206	0.04369525

Table 8: VaR and TVaR for normal and t-Student distributions

For our simulation, we can see that the values are pretty similar to all the other methods for the non-independent distributions. One can compare these value to the ones obtained assuming independence and see how not taking into account dependence makes the VaR and the TVaR considerably lower.

2.3.2 Generalized hyperbolic distributions

Considering a more general kind of distributions could allow the risk measures to be more precise if they do model the data better. Hence, in this section we consider generalized hyperbolic distributions for our data. Because we are asked to just use the best possible model, we will first need to fit different distributions to the data and compare them using goodness-of-fit measures such as the AIC, one of the most used measures.

The distributions that we consider are the multivariate normal, both the symmetric and asymmetric multivariate t-Student, both the symmetric and asymmetric multivariate hyperbolic, both the symmetric and asymmetric multivariate normal inverse Gaussian, both the symmetric and asymmetric multivariate variance-gamma and both the symmetric and asymmetric multivariate generalized hyperbolic.

Using the log-returns from the data, we fit all of these distributions and obtain the following values for the AIC in Table 9a and 9b:

Distrib.:	**Normal**	**t-Stud. (Sym)**	**t-Stud. (Asym)**	**Hyperb. (Sym)**	**Hyperb. (Asym)**
AIC:	-6970.42	-7079.428	-7074.357	-7075.179	-7070.005

Table 9a: AIC values for generalized hyperbolic distributions

Dist.:	**NIG (Sym)**	**NIG (Asym)**	**VG (Sym)**	**VG (Asym)**	**GHyp. (Sym)**	**GHyp. (Asym)**
AIC:	-7078.613	-7073.558	-7074.671	-7069.361	-7077.683	-7072.627

Table 9b: AIC values for generalized hyperbolic distributions

As we can see, the symmetric t-Student seems to be the best fit for our data, so we use this distribution in order to simulate the sequences of observations, compute the linear losses and obtain the portfolio losses distribution from which one can obtain the VaR and the TVaR. As usual, the results from the simulation of the symmetric multivariate t-Student are reported in the following table:

Levels:	$\alpha = 0.99$	$\alpha = 0.995$	$\alpha = 0.999$
VaR:	0.05420065	0.06413667	0.09088315
TVaR:	0.06981834	0.08114787	0.1115041

Table 10: VaR and TVaR for the symmetric multivariate t-Student

In contrast to the other cases, we consider different levels for the VaR and the TVaR, and we can see the effect of the level increase in both risk measures, making them bigger for each level.

2.3.3 Gumbel and Clayton copulas

The multivariate models considered before expand the possibilities for obtaining good estimates for the VaR and the TVaR, but if one wants to expand it even more, trying different dependence structures through the usage of copulas and marginal distributions may be a great option, as concrete structures of dependence with different marginals can yield various multivariate distributions that might be even more adequate than the ones previously considered.

In this case, we are asked to use Gumbel and Clayton's copula considering normal and t-Student marginal distributions for the stocks. The first thing we need to do is to transform the data to pseudo-observations, so that we transform the original variables to uniformly distributed variables. We show different bivariate plots for the pseudo-observations in Figure 3.

Through these plots, we could detect that there might be some tail dependency on the lower tail, but it is a little bit difficult to spot upper tail dependency on some of the plots. Moreover, the plot seems like a Clayton copula plot, so we might expect a better fit for it.

The next thing would be computing the copula parameter for both copulas. This is done through rank correlation measures such as Kendall's tau ρ_τ, as it can be proved there is a relation between the copula parameter θ and ρ_τ that leads to a method of moments estimator for θ using the estimated ρ_τ. By obtaining the rank correlations matrix for the portfolio one can obtain different parameters (for the different pairwise correlation) and, hence, obtain different estimations for θ. From all of this possible parameters, we choose the parameter with the maximum value and use it to fit the pseudo-observations from the data to the different copulas.

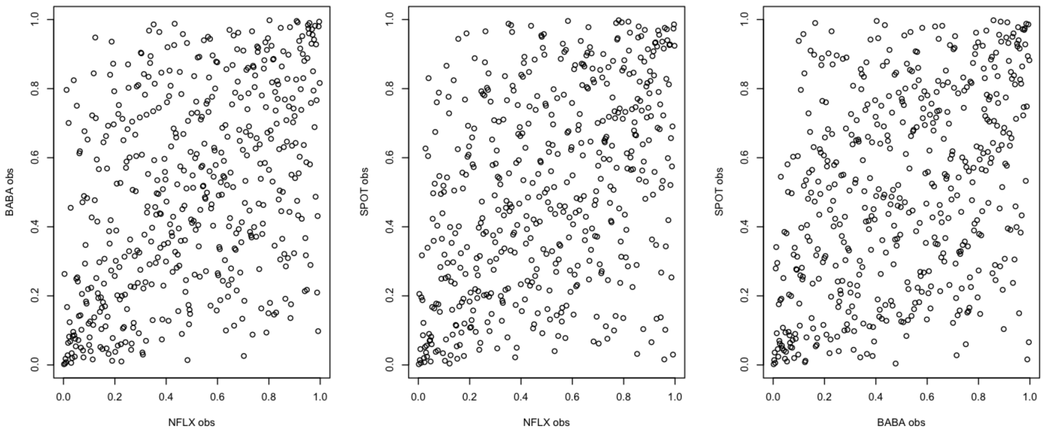

Figure 3: Bivariate plots for the different stocks

Table 11 shows the matrix composed of Kendall's tau (rank correlations) and Table 12 shows the AIC values and the estimated parameters for the fit of the pseudo-observations to both copulas.

Stocks:	**NFLX**	**BABA**	**SPOT**
NFLX	1.0000000	0.3078317	0.2958717
BABA	0.3078317	1.0000000	0.2819880
SPOT	0.2958717	0.2819880	1.0000000

Table 11: Rank correlations matrix using Kendall's tau

Dist.:	**Gumbel**	**Clayton**
θ:	1.444735	0.8894705
AIC:	-235.2584	-282.4499

Table 12: Copula parameters and AIC values

The results show that the Clayton copula seems to be a better for the pseudo-observations extracted from the data than the Clayton copula, supporting our first guess of lower tail dependency. Now that we have fitted the two copulas, we need to fit the log-returns of each to a univariate normal and t-Student distribution, so that we can simulate log-returns for a distribution with the specified copula and normal or t-Student marginal distributions. The AIC Values for the normal and the t-Student fitted marginal distributions are shown on Table 13:

Stocks:	**NFLX**	**BABA**	**SPOT**
Normal:	-2171.632	-2376.67	-2138.905
t-Student:	-3324.047	-3581.488	-3281.145

Table 13: AIC values for the marginal distribution fittings

Because of the AIC values obtained, one can see that the AIC is lower for all the individual stocks in our portfolio when adjusting the data to a t-Student margin rather than a normal margin. Hence, the t-Student margins might be a more realistic assumption relative to the normal margins assumption, meaning that the tails for the univariate distributions are thicker (extreme values are more likely).

Now that copulas and margins have been fitted, we can begin with the simulation of observations in order to estimate the VaR and the TVaR. The mechanism is the common one: we simulate observations from a distribution and then we compute the linearized losses for each simulation, so that we can form a linearized losses distribution and obtain risk measures from it. The VaR and TVaR estimates for the simulations, the different combinations of copula and marginal distributions are represented in tables 14 to 16:

Stocks:	**Normal-Gumbel**	**Normal-Clayton**	**t-Gumbel**	**t-Clayton**
VaR:	0.05113590	0.05635560	0.05732144	0.06333923
TVaR:	0.05828915	0.06567268	0.07396246	0.08560246

Table 14: VaR and TVaR estimates for different combinations at a 99% level

Stocks:	**Normal-Gumbel**	**Normal-Clayton**	**t-Gumbel**	**t-Clayton**
VaR:	0.05647572	0.06326783	0.06776360	0.07702692
TVaR:	0.06305436	0.07188173	0.08610572	0.1019366

Table 15: VaR and TVaR estimates for different combinations at a 99.5% level

Stocks:	**Normal-Gumbel**	**Normal-Clayton**	**t-Gumbel**	**t-Clayton**
VaR:	0.06722870	0.07733864	0.09572551	0.11525031
TVaR:	0.07297759	0.08486184	0.1202468	0.1487479

Table 16: VaR and TVaR estimates for different combinations at a 99.9% level

From all these simulations and estimates, we can see how the VaR and TVaR for the copula and the margins that best fit our data (the Clayton copula with t-Student margins) yields values a little bit further from the values we have obtained with the previous models for the same 99% level. Of course, whenever the level increases, all the estimates increase as well. Additionally, we spot that the Clayton copula estimates are larger than the ones using the Gumbel copula, and that the t-Student margins produce higher results as well for the same copula. This might be the effect of the lower tail dependency and the thickness of the tails of the margins.

3 Factorial Analysis

In the previous sections we estimated common risk measures such as the VaR and the TVaR using different methods of estimation and fitting different models and dependency structures for the data in order to increase the goodness of fit of our estimates to the real distribution for portfolio losses. Nonetheless, our focus was the estimation of the risk measures, but not the analysis of the data itself to infer from it. We just took the data is given and obtained various statistics used for the estimation procedures, but we have not tried to explain the data itself.

One useful analysis normally applied in financial practice is factorial analysis, which focus on explaining the randomness of the risk factor changes (the log-returns, in our case) through a set of common factors. The main advantage of using this method is to reduce the dimensions of the problem, so one can infer easier from the data by looking at less results.

The theoretical modeling for a random vector $\mathbf{X}$ uses a random vector of common factors $\mathbf{F}$ and a vector of random errors ϵ, where $\mathbf{a}$ and $\mathbf{B}$ represent a vector and a matrix of constant parameters, respectively.

$$\mathbf{X} = \mathbf{a} + \mathbf{B}\mathbf{F} + \epsilon \tag{1}$$

The model assumes that the mean of the errors is null and that the different errors are not correlated between each other. Moreover, we assume that the correlation between the factors and the errors is null. However, is important to remark that independence between factors and errors is not assumed, and it just implies independence when linear correlation fully determines the distribution, as in the case of a normal distribution.

Using this model, we can analyze the impact of the common factors through **B** and the variance-covariance matrix of the model, obtained from the various regressions that will be estimated. Hence, we will do the analysis using just one common factor for the three companies: the S&P500 stock index. This is the most important index, even though we could have used other indexes such as the NASDAQ100 or use more than one factor or index, and we have selected it to represent a macroeconomic source of risk.

Therefore, we first start by analyzing normality of the risk factors **X** and the common factor **F** and we then analyze the different regressions and the estimated parameters for observed factors.

3.1 Normality Tests

As we mentioned previously, the theoretical model does not assume independence for the errors and the common factors, but it is possible if the variables in our model are normally distributed, as the linear correlations determine the distribution. Hence, it is logical to begin by testing normality. This is done through common tests such as the Mardia Test and the joint normal test.

The Mardia test is a test designed to check whether the skewness and the kurtosis of the distribution of the variables are similar to a multivariate Gaussian distribution, where the null hypothesis is that the variables are distributed like this. The other joint normality test we will use is the Kolmogorov-Smirnov test for multivariate normality, whose null hypothesis is that the variables follow a multivariate normal distribution. The results of both tests are represented in Table 17 and Figure 4.

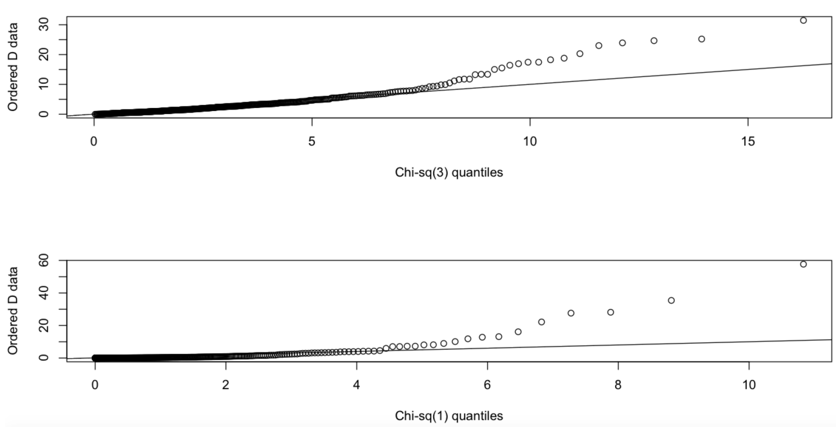

Figure 4: Q-Q plot for **X** (above) and **F** (below)

Tests:	**Mardia - X**	**Mardia - F**	**KS - X**	**KS - F**
Skewness:	3.663736e-15	0.0000000	-	-
Kurtosis:	0.000000e+00	0.0000000	-	-
KS:	-	-	2.462758e-07	0.0000000

Table 17: P-values for the normality tests

The p-values obtained for tests are clear and show that we can reject the null hypothesis of multivariate normality for both the **X** and the **F**. This also means that there is no independence between the errors and between the errors and the factors, so we might take into account this fact using more sophisticated methods. Anyway, we will adhere to our proposed analysis by now.

3.2 Analysis with the S&P500

Now we can estimate the regressions for the model, obtaining then an estimation for **a** and **B**. These parameters are represented in the following table with their respective p-values for an univariate test for a null hypothesis of a zero value (the estimated value is not significantly different from zero):

X:	**NFLX**	**BABA**	**SPOT**
a:	0.0001267996	0.0003264373	0.0001508966
P-values:	0.779	0.345	0.759
B:	0.8764558229	0.8130960567	0.7692003435
P-values:	2e-16	2e-16	2e-16

Table 18: Results from the regressions

The results show that the S&P500 log-returns is a significant factor to take into account when explaining the log-returns of the stocks in our portfolio, as all the p-values reject the null hypothesis of a null effect. When it comes to the intercepts, it seems as they are not significant in any of the regressions. However, we are not very interest in **a** but in **B**, so it is a minor problem for our analysis.

Finally, we are able to analyze the variability of the log-returns of our assets through this results. Because the variance-covariance matrix of our model will be the sum between the variance that comes from the common risk factor (the S&P500) and the error terms, we can decompose it into these parts and see the effect of the S&P500 and obtain the matrix of correlations.

Table 21 represents the correlation matrix for the errors. The variance-covariance matrices are not very interesting per se in our analysis, so we do not include them in the report. However, other methods could be applied to further study variability in our model.

Errors:	**NFLX**	**BABA**	**SPOT**
NFLX	1.0000000	0.2606512	0.2493779
BABA	0.2606512	1.0000000	0.2336037
SPOT	0.2493779	0.2336037	1.0000000

Table 19: Correlation matrix for the variability explained by the error terms

By looking at the correlation matrix for the error terms of the different regressions, we can see that the values are greater than 0.2, which is a common used cut-off value for correlations to determine whether the common factors represent linear dependence between the changes in risk factors, so the S&P500 does not represent it.

4 Extreme Value Distribution Analysis

In the last part of our analysis, we will focus on the modeling and analysis of the extremes or the tails of the loss distribution for our portfolio. Even though the other sections work with the whole distribution of losses, we did not work explicitly with the tails of the distribution. The extreme value theory allows us to model and analyze the distribution of maximum values, and hence, to work with the right tail of the portfolio losses distribution.

In this project we must analyze the extremes by finding which distribution fits best the behaviour of the right tail and adjust the distribution to the data in order to estimate the VaR at the 99% level. Hence, we divide the analysis in these two parts.

4.1 Right tail Extreme Value distribution

To find which is the right extreme value distribution for our data, we can use the Hill estimator, which estimates the reciprocal of the scale parameter ξ and allows to obtain it. The result for the Hill estimator with the data for the positive losses in our portfolio, represented also as a graph in Figure 5 (see the text in the upper part), shows that $\xi > 0$, which means that the positive losses behave as a Pareto distribution. This also means that the tail of the extreme value distribution is heavy or thick, and that the distribution has first and second moments.

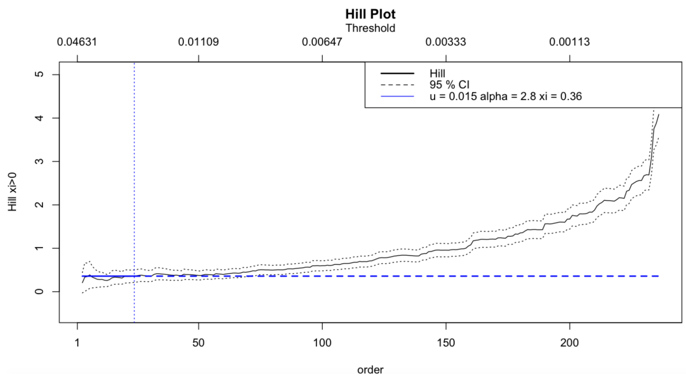

Figure 5: Hill plot and ME

4.2 VaR estimation

Now that we have chosen the Pareto distribution as the best model, we just need to fit the data to the distribution and estimate the VaR value for a 99% level. In order to do so, we first have to obtain a threshold u to select the tail we will be working with and then obtain the percentile that corresponds to the VaR at 99% level if we took into account the whole range of positive losses (not just the ones that surpass our threshold).

When fitting the data to a Pareto distribution, we can obtain a VaR at 99% level for the observations that surpass our threshold, call it $VaR^u_{99\%}$, and, hence, the VaR at 99% for the whole distribution just by adding the threshold to $VaR^u_{99\%}$. In Table 20 we show not just the VaR but also other relevant quantities, such as the percentile for the VaR in the distribution for the value above the threshold if we took into account all the distribution, called $conf$.

Parameter:	u	$conf$	$VaR^u_{99\%}$	$VaR^u_{99\%}$
Value:	0.015	0.09126582	0.05636392	0.07136392

Table 20: Values computed for the Pareto fit

As we can see, the VaR obtained is higher than the other VaR values obtained with the previous methodologies (at the same level). However, it is expected as we are just considering the extreme values distribution, which might be a richer analysis to compute extreme losses in the context of our portfolio.

Appendix

```
##################################################################################
#                            R code for the Project                              #
##################################################################################

# Set working directory

setwd("/Users/ikercaballerobragagnini/Desktop/UPC/Risk Quantification")

# Install libraries

library(QRM)
library(copula)
library(readr)
library(evmix)
library(evir)
library(ercv)
library(condmixt)

#--------------------------------------------------------------------------------#
#                                  Exercise 1                                    #
#--------------------------------------------------------------------------------#

# a)

eq_stocks <- read.table("prices.csv",sep=",",header=TRUE)
View(eq_stocks)
summary(eq_stocks)
eq_prices <- cbind(eq_stocks[2],eq_stocks[3],eq_stocks[4])
summary(eq_prices)
attach(eq_prices)

mat_prices <- as.matrix(eq_prices)
n <- nrow(mat_prices)
colnames(mat_prices)<-c("NFLX","BABA","SPOT")
colnames(mat_prices)

mat_prices.ln <- diff(log(mat_prices))
mean.ln <- colMeans(mat_prices.ln)
mean.ln <-as.matrix(mean.ln)
Covar.ln <- var(mat_prices.ln)
Covarn.ln <- (n-1)*Covar.ln/n
Covarn.ln <- as.matrix(Covarn.ln)

weights <- c(0.4826,0.2682,0.2488)
weights <- as.matrix(weights)
sum(weights)
V <- 1

EL <-- V*t(weights)%*%mean.ln
VL <- V^2*(t(weights)%*%Covarn.ln%*%weights)

alpha <- 0.99

VaR.Norm <- qnorm(alpha,EL,sqrt(VL))
VaR.Norm

ES.Norm <- EL+sqrt(VL)*dnorm(qnorm(alpha))/(1-alpha)
ES.Norm

mu <- EL
```

```
nu <- 4
sigma_t <- VL * (nu-2)/nu
qt(alpha, nu)
VaR.tStudent <- mu +sqrt(sigma_t)*qt(alpha,nu)
VaR.tStudent
dt(qt(alpha, nu), nu)
ES.tStudent<-EL+sqrt(sigma_t)*( dt(qt(alpha, nu), nu)/(1-alpha) )*(nu+qt(alpha, nu)*qt(
ES.tStudent

# b)

SH.L<-(-V)*(mat_prices.ln%*%weights)
SH.L
summary(SH.L)

mod.norm <- fit.norm(SH.L)
attributes(mod.norm)

mod.t <- fit.st(SH.L)
attributes(mod.t)

AIC_n<-2*2-2*mod.norm$ll.max
AIC_t<-2*3-2*mod.t$ll.max

qnorm(alpha,mean=mod.norm$mu,sd=sqrt(mod.norm$Sigma))
ESnorm(alpha,mu=mod.norm$mu,sd=sqrt(mod.norm$Sigma))

qst(alpha,4,mu=mod.t$par.ests[2], sd=mod.t$par.ests[3])
ESst(alpha,4,mu=mod.t$par.ests[2], sd=mod.t$par.ests[3])

qst(alpha,5,mu=mod.t$par.ests[2], sd=mod.t$par.ests[3])
ESst(alpha,5,mu=mod.t$par.ests[2], sd=mod.t$par.ests[3])

qst(alpha,mod.t$par.ests[1],mu=mod.t$par.ests[2], sd=mod.t$par.ests[3])
ESst(alpha,mod.t$par.ests[1],mu=mod.t$par.ests[2], sd=mod.t$par.ests[3])

# c)

# Multivariate Normal and t-Student

rep <- 1000000
mc_obs <- rmvnorm(rep, mean.ln, Covarn.ln)

mc_obs <-as.matrix(mc_obs)
mc_Lsim <- (mc_obs%*%weights)
mc_Lsim <- -V*mc_Lsim
quantile(mc_Lsim, alpha)
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha)])

mc_obs <- rmvt(rep, mean.ln, Covarn.ln)

mc_obs <-as.matrix(mc_obs)
mc_Lsim <- (mc_obs%*%weights)
mc_Lsim <- -V*mc_Lsim
quantile(mc_Lsim, alpha)
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha)])

# (Independence)

mc_obs <- rmvnorm(rep, mean.ln, diag(diag(Covarn.ln)))
```

```
mc_obs <- as.matrix(mc_obs)
mc_Lsim <-- -V*(mc_obs%*%weights)

quantile(mc_Lsim, alpha)
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha)])

mc_obs <- rmvnorm(rep, mean.ln, diag(diag(Covarn.ln)))
mc_obs <- as.matrix(mc_obs)
mc_Lsim <-- -V*(mc_obs%*%weights)

quantile(mc_Lsim, alpha)
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha)])

# Generalized Hyperbolic Distributions

mod.gauss <- fit.gaussmv(mat_prices.ln)

mod.t1 <- fit.tmv(mat_prices.ln,symmetric = TRUE)

mod.t2 <- fit.tmv(mat_prices.ln,symmetric = FALSE)

mod.hyp1<- fit.hypmv(mat_prices.ln,symmetric = TRUE)

mod.hyp2<- fit.hypmv(mat_prices.ln,symmetric = FALSE)

mod.nig1<- fit.NIGmv(mat_prices.ln,symmetric = TRUE)

mod.nig2<- fit.NIGmv(mat_prices.ln,symmetric = FALSE)

mod.vg1<- fit.VGmv(mat_prices.ln,symmetric = TRUE)

mod.vg2<- fit.VGmv(mat_prices.ln,symmetric = FALSE)

mod.ghyp1<- fit.ghypmv(mat_prices.ln,symmetric = TRUE)

mod.ghyp2<- fit.ghypmv(mat_prices.ln,symmetric = FALSE)

AIC(mod.gauss)
AIC(mod.t1)
AIC(mod.t2)
AIC(mod.hyp1)
AIC(mod.hyp2)
AIC(mod.nig1)
AIC(mod.nig2)
AIC(mod.vg1)
AIC(mod.vg2)
AIC(mod.ghyp1)
AIC(mod.ghyp2)

alpha<-c(0.99, 0.995,0.999)
weights <- c(0.4826,0.2682,0.2488)
sum(weights)
V<-1

r<-1000000

Sim_mat_prices.ln_t1<-rghyp(r, object = mod.t1)
mc_Lsim <- -V*(Sim_mat_prices.ln_t1%*%weights)

quantile(mc_Lsim, alpha)
```

```
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[1])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[2])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[3])])

# Copulas

unif_data <- pobs(mat_prices.ln)

par(mfrow=c(1,3))
plot(unif_data [,1],unif_data [,2],xlab = "NFLX obs",ylab = "BABA obs")
plot(unif_data [,1],unif_data [,3],xlab = "NFLX obs",ylab = "SPOT obs")
plot(unif_data [,2],unif_data [,3],xlab = "BABA obs",ylab = "SPOT obs")

MatrixCorrlin<-cor(mat_prices.ln)
MatrixCorrlin

p1<-MatrixCorrlin[1,2]
p2<-MatrixCorrlin[1,3]
p3<-MatrixCorrlin[2,3]

rotau<-Kendall(mat_prices.ln)
rotau

ParGum1<-1/(1-rotau[1,2])
ParGum1

ParGum2<-1/(1-rotau[1,3])
ParGum2

ParGum3<-1/(1-rotau[2,3])
ParGum3

ParGum<-max(ParGum1,ParGum2,ParGum3)

ParClay1<-(2*rotau[1,2])/(1-rotau[1,2])
ParClay1

ParClay2<-(2*rotau[1,3])/(1-rotau[1,3])
ParClay2

ParClay3<-(2*rotau[2,3])/(1-rotau[2,3])
ParClay3

ParClay<-max(ParClay1,ParClay2,ParClay3)


mod.GAUSS1 <- fit.norm(mat_prices.ln[,1])
aic<-4-2*mod.GAUSS1$ll.max
aic

mod.GAUSS2 <- fit.norm(mat_prices.ln[,2])
aic<-4-2*mod.GAUSS2$ll.max
aic

mod.GAUSS3 <- fit.norm(mat_prices.ln[,3])
aic<-4-2*mod.GAUSS3$ll.max
aic

mod.t1 <- fit.st(mat_prices.ln[,1])
```

```
aic<-6-3*mod.t1$ll.max
aic

mod.t2 <- fit.st(mat_prices.ln[,2])
aic<-6-3*mod.t2$ll.max
aic

mod.t3 <- fit.st(mat_prices.ln[,3])
aic<-6-3*mod.t3$ll.max
aic

gumb.cop0 <- gumbelCopula(ParGum, dim =3)
gumbCopEst<- fitCopula(gumb.cop0,unif_data , method="mpl")
AIC(gumbCopEst)
Sim_gumbelcopula <- rCopula(r, gumb.cop0)

clay.cop<- claytonCopula(param =ParClay, dim = 3)
clay.CopEst<- fitCopula(clay.cop,unif_data , method="mpl")
AIC(clay.CopEst)
Sim_clayC<-rCopula(r,clay.cop)

alpha<-c(0.99, 0.995,0.999)
weights <- c(0.4826,0.2682,0.2488)
V<-1


sim.mat_prices.ln1 <- qnorm(Sim_gumbelcopula[,1], mean=mod.GAUSS1$mu, sd=sqrt(mod.GAUS
sim.mat_prices.ln2 <- qnorm(Sim_gumbelcopula[,2], mean=mod.GAUSS2$mu, sd=sqrt(mod.GAUS
sim.mat_prices.ln3 <- qnorm(Sim_gumbelcopula[,3], mean=mod.GAUSS3$mu, sd=sqrt(mod.GAUS
Sim_mat_prices.ln_norm <- cbind(sim.mat_prices.ln1,sim.mat_prices.ln2,sim.mat_prices.ln
mc_Lsim <- -V*(Sim_mat_prices.ln_norm%*%weights)

quantile(mc_Lsim, alpha)

mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[1])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[2])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[3])])


sim.mat_prices.ln1t <- qst(Sim_gumbelcopula[,1], mu=mod.t1$par.ests[2],
sd=mod.t1$par.ests[3],mod.t1$par.ests[1])
sim.mat_prices.ln2t <- qst(Sim_gumbelcopula[,2], mu=mod.t2$par.ests[2],
sd=mod.t2$par.ests[3],mod.t2$par.ests[1])
sim.mat_prices.ln3t <- qst(Sim_gumbelcopula[,3], mu=mod.t3$par.ests[2],
sd=mod.t3$par.ests[3],mod.t3$par.ests[1])
Sim_mat_prices.ln_t <- cbind(sim.mat_prices.ln1t,sim.mat_prices.ln2t,sim.mat_prices.ln3
mc_Lsim <- -V*(Sim_mat_prices.ln_t%*%weights)

quantile(mc_Lsim, alpha)

mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[1])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[2])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[3])])


sim.mat_prices.ln1 <- qnorm(Sim_clayC[,1], mean=mod.GAUSS1$mu, sd=sqrt(mod.GAUSS1$Sigma
sim.mat_prices.ln2 <- qnorm(Sim_clayC[,2], mean=mod.GAUSS2$mu, sd=sqrt(mod.GAUSS2$Sigma
sim.mat_prices.ln3 <- qnorm(Sim_clayC[,3], mean=mod.GAUSS3$mu, sd=sqrt(mod.GAUSS3$Sigma
Sim_mat_prices.ln_norm <- cbind(sim.mat_prices.ln1,sim.mat_prices.ln2,sim.mat_prices.ln
mc_Lsim <- -V*(Sim_mat_prices.ln_norm%*%weights)

quantile(mc_Lsim, alpha)
```

```
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[1])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[2])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[3])])


sim.mat_prices.ln1t <- qst(Sim_clayC[,1], mu=mod.t1$par.ests[2],
sd=mod.t1$par.ests[3],mod.t1$par.ests[1])
sim.mat_prices.ln2t <- qst(Sim_clayC[,2], mu=mod.t2$par.ests[2],
sd=mod.t2$par.ests[3],mod.t2$par.ests[1])
sim.mat_prices.ln3t <- qst(Sim_clayC[,3], mu=mod.t3$par.ests[2],
sd=mod.t3$par.ests[3],mod.t3$par.ests[1])
Sim_mat_prices.ln_t <- cbind(sim.mat_prices.ln1t,sim.mat_prices.ln2t,sim.mat_prices.ln3
mc_Lsim <- -V*(Sim_mat_prices.ln_t%*%weights)

quantile(mc_Lsim, alpha)

mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[1])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[2])])
mean(mc_Lsim[mc_Lsim > quantile(mc_Lsim, alpha[3])])

#----------------------------------------------------------------------------------#
#                                    Exercise 2                                    #
#----------------------------------------------------------------------------------#

data<-read.table("sp_prices.csv", header=TRUE, sep=";")
summary(data)
time<-nrow(data)

X<-cbind(data$NFLX,data$BABA,data$SPOT)
summary(X)
factor<-cbind(data$SP500)

# Normally Tests

MardiaTest(X)
MardiaTest(factor)

par(mfrow=c(2,1))
jointnormalTest(X)
jointnormalTest(factor)

# Analysis with the S&P500

mreg<-lm(X~factor)
summary(mreg)

b<-coef(mreg)
b
b<-b[2:2,]
b<-t(t(b))
b

var_x<-var(X)
var_x

var_f<-var(factor)
var_f

bb<-b%*%var_f%*%t(b)
bb

res_var<-var_x-bb
diag_res_var<-sqrt(diag(diag(res_var)))
```

```
inv_diag_res_var<-solve(diag_res_var)
res_corr<-inv_diag_res_var%*%res_var%*%inv_diag_res_var
res_corr

#------------------------------------------------------------------------------------#
#                                  Exercise 3                                       #
#------------------------------------------------------------------------------------#

stocks <- read.table("sp_prices.csv", header=TRUE, sep=";")

head(stocks)
mat_prices.ln <- cbind(stocks[2],stocks[3],stocks[4])
mat_prices.ln <- as.matrix(mat_prices.ln)

n<-nrow(mat_prices.ln)
n

colnames(mat_prices.ln)<-c("NFLX","BABA","SPOT")
colnames(mat_prices.ln)


weights <- c(0.4826,0.2682,0.2488)
SH.L<--(mat_prices.ln%*%weights)
summary(SH.L)

L_pos<-SH.L[(SH.L>0)]

np<-length(L_pos)
L_pos<-L_pos[order(L_pos)]
q<-0.99

hillplot(L_pos)

L<-sort(L_pos, decreasing = FALSE)
L
ne<-sum(L>0.015)
Hpar<-hillest(L, ne)
Hpar


u<-0.015
par.GPD<-gpd(L_pos,threshold = u)
par.GPD

Lu<-L[(L>=u)]
nu<-length(Lu)
fr<-(np-nu)/np
fr
conf<-q-fr
conf
VaR_GPD=qgpd((conf/(1-fr)),par.GPD$par.ests[1],0, par.GPD$par.ests[2])
VaR_GPD+u
```