ST429 – Statistical Methods for Risk Management

Green finance and COVID-19

Group 1

Candidate numbers: 10279, 12878, 20411 & 15200

1 Introduction

In this rapidly evolving world, sustainable development is a necessity. Pollution levels have increased significantly and greenhouse gas emissions are depleting the ozone layer, causing climate change. Many measures are being taken on an international scale to minimise the effect of GHGs and climate change, for example, the Paris agreement of 2015. This treaty laid out medium to long term goals to reduce the global temperature level back to pre-industrial levels. The signing of this agreement caused many companies to change the way they were doing business, to adapt to the climate needs.

A few years latter, there have been no clear signs of improval. Furthermore, the world has been struck by unprecedented events which have strongly impacted the economy: The covid outbreak, the Brexit trade agreement, the U.S. ellection, etc. which have been of utmost relevance when studying market risk.

The following project will analyse market data from the past 5 years for 10 different "green" companies in the US market. The selected businesses are sumarised in table 1. These companies have been playing a major role in reducing the risk of climate change by promoting sustainable development and reducing greenhouse gases.

Company Name	**Ticker**	**Industry**
General Electric Company NextEra Energy Partners First Trust Global Wind Energy	GE NEP FAN	Wind Power
Enphase Energy, Inc. First Solar, Inc. Sunrun, Inc.	ENPH FSLR RUN	Solar
Fuel Tech, Inc. Invesco Cleantech ETF	FTEK PZD	Pollution Controls
FuelCell Energy, Inc. Tesla, Inc.	FCEL TSLA	Green transport

Table 1: Chosen companies

Data from 17/12/2015 to 17/12/2020 was downloaded from Yahoo finance and imported in the statistical environment R for posterior analysis. Different elements of the stock data were evaluated to construct various results presented in section 2. The stock price and log return time series were analysed in sections 2.1. Section 2.2. considered the loss random variable for the portfolio and was fitted to a normal and student-t distributions in 2.3. Value at risk and expected shortfall, two risk measures based on the loss distribution, were estimated and analysed in 2.4.

Bivariate copulas were fitted to pairs of stocks through maximum likelihood fits in 2.5, and the aggregated value at risk for those pairs was estimated through Monte Carlo simulations in 2.6. Finally, a principal component analysis was carried out in 2.7 to study which risk factor changes in the portfolio carry the most weight. The first loading vector was used to construct an index, whose dependence to the constituent stocks was studied through copula fits.

2 Results

2.1 Stock analysis

2.1.1 Stock prices

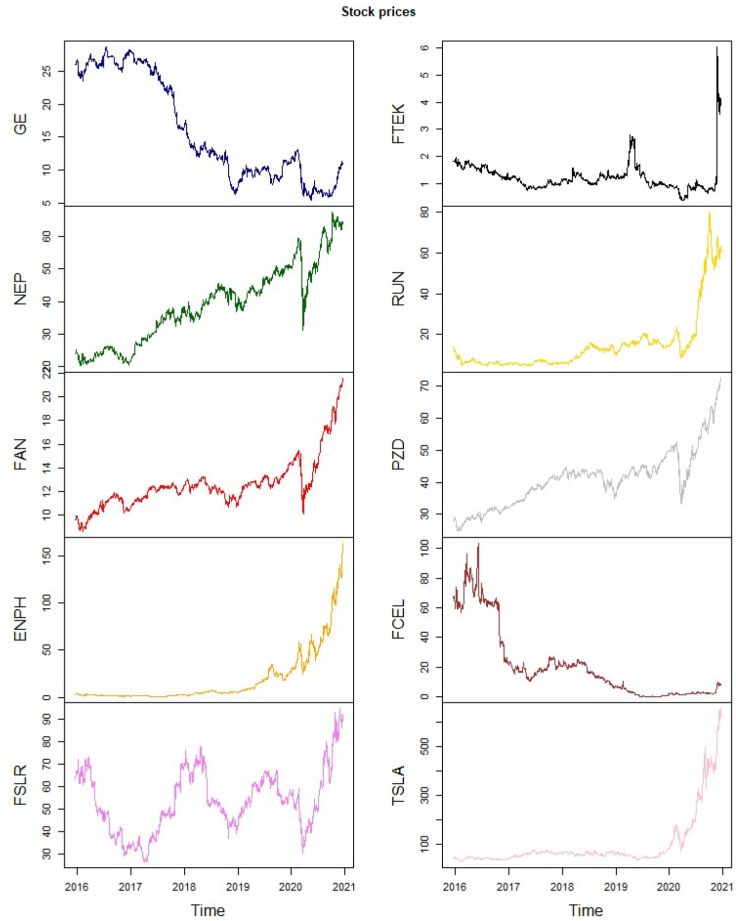


Figure 1: Stock price time series for the 10 companies for the 5 year period from 17/12/2015 to 17/12/2020.

The stock prices have been plotted as a time series and the following is a summary of the trends of individual stock prices:

- The stock price of GE gradually decreases from 2016, reaching a minimum on 14/05/2020 at \$5.47.
- The stock price of *NEP, FAN, PZD* increased from 2015 to 2020 but during march of 2020 the stock price fell rapidly.
- The stock price of *ENPH* was nearly constant until may 2019, but after that the stock price increased rapidly until 17/12/2020 for a high of \$163.52.
- The stock price of *FSLR* is very volatile and has been constantly changing from 2015 to 2020.
- The stock price of *FTEK* was rather constant until February 2019 and then, it increased until May 2019 and returned to its previous value shortly after. The stock price rose very rapidly in October 2020 and is currently at all time high of \$ 6.04.
- The stock price of *RUN* has shown not much variation until March 2020 where it sharply fell down, but after that has seen a steep growth and reached all time high on 01/10/2020 at \$79.97.
- The stock price of *FCEL* was gradually changing until November 2015, after which it fell rapidly and then it started to gradually decrease again.
- *TSLA* grew dramatically from \$ 46.09 on 18/12/2015 to a maximum of 655.90 on 17/12/2020, with most of the growth occuring in 2020. During March of that year, the stock price decreased before recovering quickly afterwards.

Many of rapid variations in the stock price have occurred in 2016 and 2020, which suggests that there might have be occurrence of special events during this time. Based on online research, the following events could be a possibility for these variations:

- **Coronavirus outbreak** in march of 2020 that lead to falling of share prices for many of the stocks.
- **2016 U.S. election**: Donald Trump was elected in 2016 and many of the companies in our portfolio stocks rose during this year.
- **2020 U.S. election**: Joe Biden was elected in November 2020, causing large absolute returns on the days preceding the election.
- **Inclusion of TSLA to the S&P 500 index:** Causing the stock prices of the car manufacturer to rise even further towards the end of 5 year period.

2.1.2 Log returns

Log returns is one of the methods used to calculate returns, which assumes continuous compounding. If we denote the stock price at time t by S_t, and we let the stock price on the previous day bt S_{t-1}, the daily log returns are given by the difference:

$$\text{Daily log returns} = \log(S_t) - \log(S_{t-1}) = \log\left(\frac{S_t}{S_{t-1}}\right) = \log(1 + R_t)$$

where R_t are the simple returns $R_t = \frac{S_t - S_{t-1}}{S_{t-1}}$.

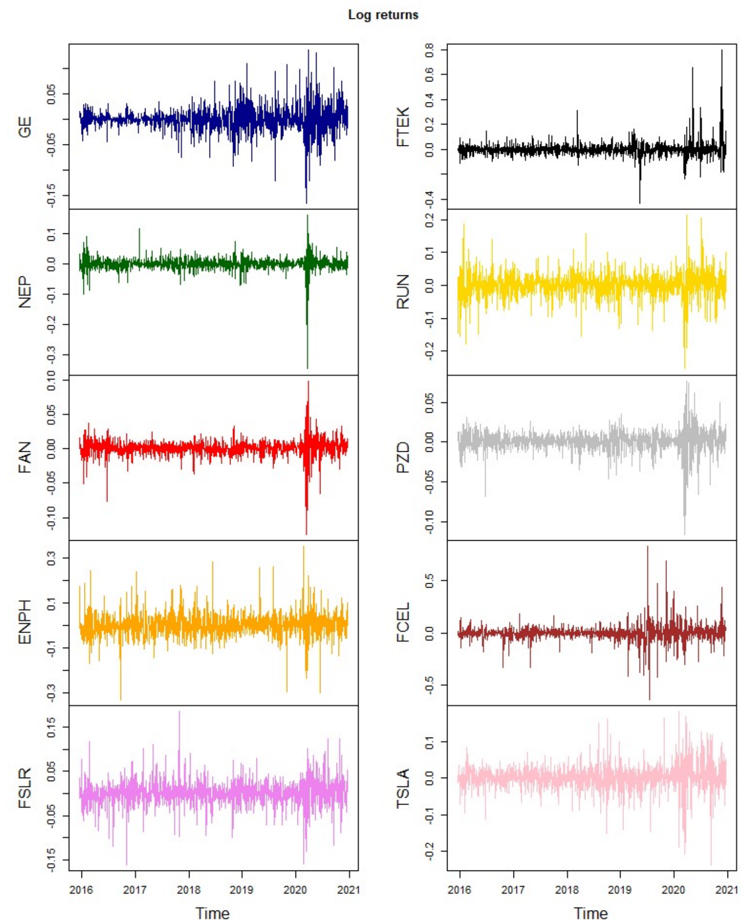


Figure 2: Log return series for the 10 companies for the 5 year period from 17/12/2015 to 17/12/2020.

As mentioned previously there has been a lot of variation in the year 2020 which is also evident when we look at the log returns. The following is a brief summary of trends of daily log returns:

- *GE, NEP, FAN, RUN, PZD* and *TSLA* have seen their biggest changes in daily log returns in 2020. Their stocks fell down in March and rose latter on within the same year. This is why they have their biggest absolute changes in the same year.
- *ENPH* and *FSLR* have seen their biggest negative log return in 2016.
- *FTEK* and *FCEL* was their biggest negative log returns in 2019.
- The biggest positive log returns happened in 2020 for *ENPH* and *FTEK*, 2019 for *FCEL* and 2017 for *FSLR*.

Stock	GE	NEP	FAN	ENPH	FSLR	FTEK	RUN	PZD	FCEL	TSLA
Average daily log returns ($\times 10^{-5}$)	-6.9	78.4	64.1	317	29.8	62.3	120	76.0	-162	211
Annualised volatility	1.23	1.03	0.57	2.43	1.30	2.69	1.79	0.62	3.63	1.62
Skewness (5y period)	-0.09	-3.05	-1.42	0.25	-0.08	4.89	-0.47	-1.38	0.95	-0.22
Kurtoris (5y period)	6.23	51.8	19.5	6.36	4.73	62.7	4.82	12.77	20.7	6.18

Table 2: Summary statistics for stocks in portfolio.

As expected from the stylised facts, daily mean returns are close to zero and annualised volatilities are close to 1%. Most stocks are negatively skewed, with very high kurtosis, which is an indication of fat tails.

2.2 Portfolio loss analysis

2.2.1 Constructing the loss random variable

Denote the value of the portfolio at time t by V_t. We define a random loss as the negative change in value:

$$L_{t+1} = -(V_{t+1} - V_t) = -\Delta V_{t+1}.$$

For our 10-stock portfolio, denote by $S_{t,i}$ the value of the i^{th} stock at time t and λ_i the number of shares in the i^{th} stock. The total stock value is then: $\sum_{i=1}^{10} \lambda_i S_{t,i}$. If we choose the risk factor changes to be log returns:

$$X_{t+1} = \log(S_{t+1,i}) - \log(S_{t,i}) = \log\left(\frac{S_{t+1,i}}{S_{t,i}}\right),$$

we can write the losses as:

$$L_{t+1} = -V_t \sum_{i=1}^{d} \omega_{t,i}(e^{X_{t+1,i}} - 1) \tag{1}$$

where V_t is the total investment at time t and $\omega_{t,i} = \frac{\lambda_i S_{t,i}}{V_t}$ are the portfolio weights for the i^{th} stock at time t.

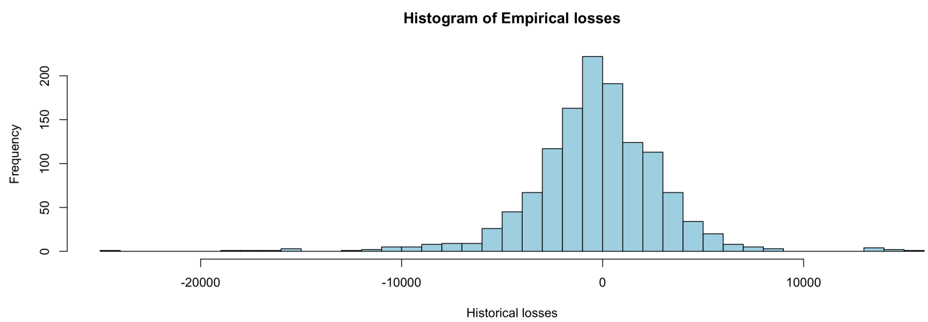


Figure 3: Histogram of the empirical losses for the porfolio, calculated following equation 1. The majority of losses occur close to zero, and a few extremes are seen around 15000 and below -10000.

2.2.2 Loss analysis

The 10 stock portfolio loss distribution was implemented in R using a user defined function, assuming an initial investment of $1000 in each stock.

Historical losses/ Empirical Losses					
Minimum	**Maximum**	**Expected Losses**	**Standard Deviation**	**Skewness**	**Kurtosis**
-24246	15541	-316	3335	-0.644	6.252

Table 3: Summary statistics for the portfolio loss random variable

As seen in table 1, the maximum loss is 15541, which occurred on 12/03/2020, while the minimum (maximum profit) is -24241 which happened on 19/02/2020. Again, we can see the distribution is negatively skewed, implying that the left tail is longer than the right tail. The loss distribution has a kurtosis of 6.25 which is a sign of leptokurticity: when compared to a normal distribution, its tails are longer and fatter, and its central peak is higher and sharper.

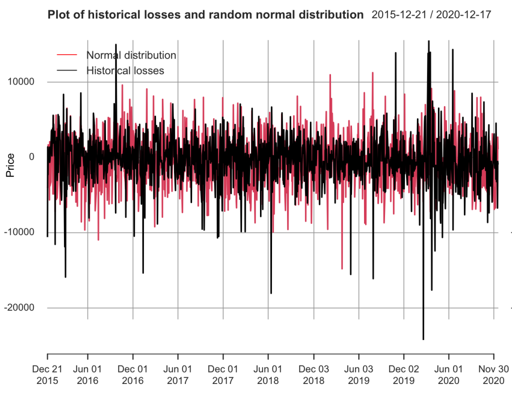
2.3 Student-t and normality study

The loss random variable was fitted to a normal and student-t distribution for further study. The fit parameters for both distributional assumptions are given in table 4.

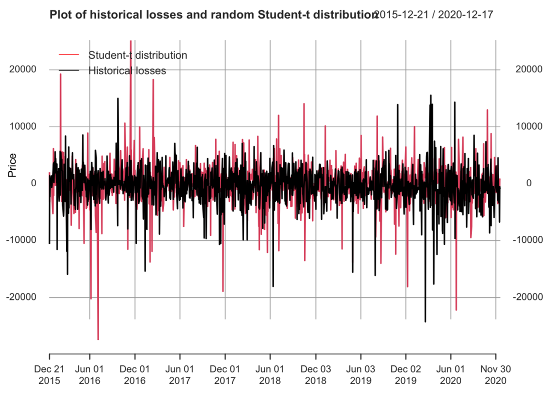
Normal		Student t		
μ	σ	**Location Param.**	**Scale Param.**	**Degrees of freedom**
-316	3334	-267	1675	2.15

Table 4: Fit parameters for normal and student-t studies

Furthermore 4a and 4b show the historical losses, which a simulation of normal a student-t distributions with the fitted parameters from table 1 respectively. As expected, the normal distribution fails to capture the extreme values present in the empirical data (e.g. peak in June 2020). While the simulated t-student does succeed in this, it fails to capture the volatility clustering which is so characteristic of returns time series.



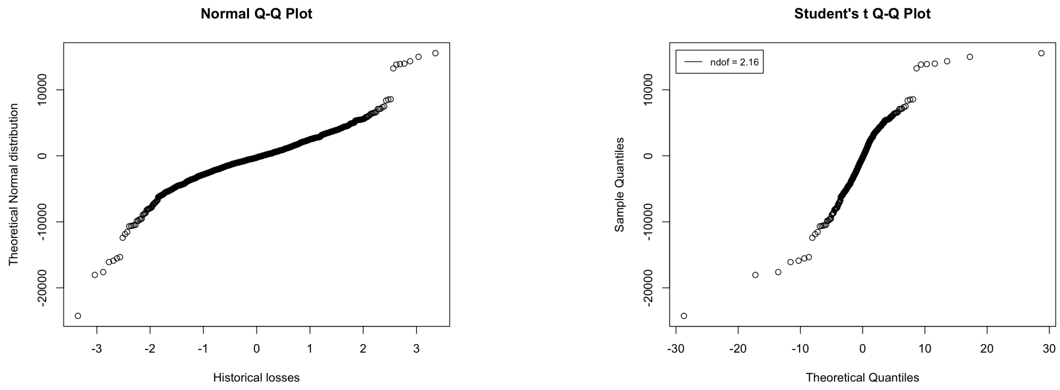
(a) Historical losses (black) and a random normal distribution with the fitted parameters (red).



(b) Historical losses (black) and a random student-t distribution with the fitted parameters (red).

Figure 4: Historical losses and simulated fitted distributions

Quantile-quantile plots for normal and student-t fits are provided in 5a and 5b respectively. The normal QQ-plots is S-shaped, which once again provides evidence for heavy tails. This was also inferred by looking at the skewness and kurtosis values of the historical losses. The student-t QQ plot with $\nu \approx 2$ is relatively more linear for smaller quantiles, with some major deviations after the 10^{th} quantile.



(a) QQ plot for the quantiles of the empirical distribution against the quantiles of the fitted normal.

(b) QQ plot for the quantiles of the empirical distribution against the quantiles of the fitted student-t.

Figure 5: Quantile-Quantile plots

To fully discard the normality hypothesis, the Jarque-Bera test statistic was computed for the loss random variable to be 1.67×10^6. This comes as no surprise since the observed skewness and kurtosis are significantly different from that of normal distribution's.

2.4 Value at risk and Expected shortfall capital requirements

Value-at-risk is a statistical measure of the riskiness of financial entities or portfolios of assets. It is a risk measure based on loss distribution, and is defined as the maximum amount expected to be lost over a given time horizon, at a pre-defined confidence level. Formally, for a loss $L \sim F_L$, the value at risk at confidence level $\alpha \in (0,1)$ is:

$$\mathrm{VaR}_\alpha(L) = \inf\{x \in \mathbb{R} \mid F_L(x) \geq \alpha\}$$

The expected shortfall is a risk measure sensitive to the shape of the tail of the distribution of returns on a portfolio. It is calculated by averaging all of the returns in the distribution that are worse than the VaR of the portfolio at a given level of confidence. Formally, for a loss $L \sim F_L$ with $\mathbb{E}[|L|) < \infty$, the expected shortfall at confidence level $\alpha \in (0,1)$ is:

$$\mathrm{ES}_\alpha = \frac{1}{1-\alpha}\int_\alpha^1 q_u(F_L)du$$

where $q_u(F_L) = \mathrm{VaR}_u(L)$.

The VaR and ES were calculated for the portfolio losses at various confidence levels α.

α	0.5	0.6	0.7	0.8	0.9	0.93	0.95	0.97	0.99
VaR	-267	332	1035	2045	3317	3849	4390	5376	7111
ES	1991	2480	3082	3889	5189	5885	6598	7792	11321

Table 5: Value at risk and Expected shortfall for loss random variable at different confidence levels α

As expected, the capital requirements for the expected shortfall is higher than the value at risk at the same confidence level α. Both risk measures are rapidly increasing when alpha gets close to 1, however, VaR attains a positive value at $\alpha = 0.6$, while the expected shortfall remains positive from $\alpha = 0.2$. For a fixed α of e.g. $\alpha = 0.9$, the minimum capital required to satisfy VaR is 3117, while it is 5189 for ES, a 66% higher.

A plot for the value at risk and Expected shortfall at different confidence levels is provided in figure 6. It was also studied how the empirical VaR and ES compare to that of a normal distribution. Plots are provided for VaR and ES in figures 7a and 7b respectively. The VaR for empirical losses is higher than the normal losses up until $\alpha = 0.5$ and after that, the normal losses have shown a marginally higher capital requirement. On the other hand, the expected shortfall for normal losses is on the higher side when compared to empirical losses, until their intersection at $\alpha = 0.94$ (where ES = 6379).

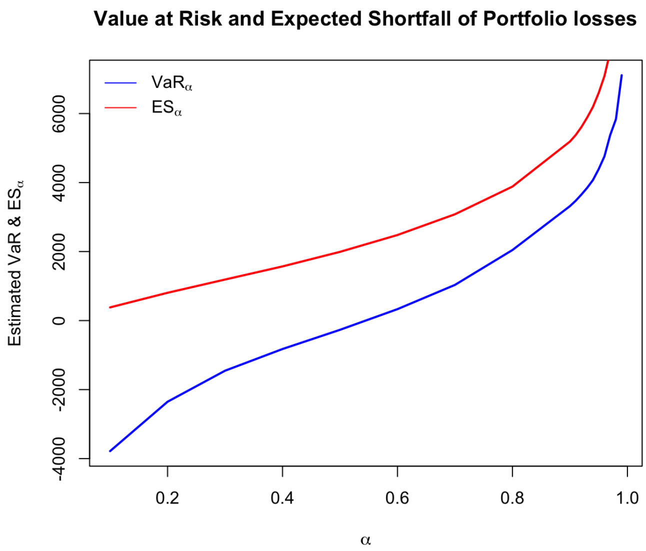
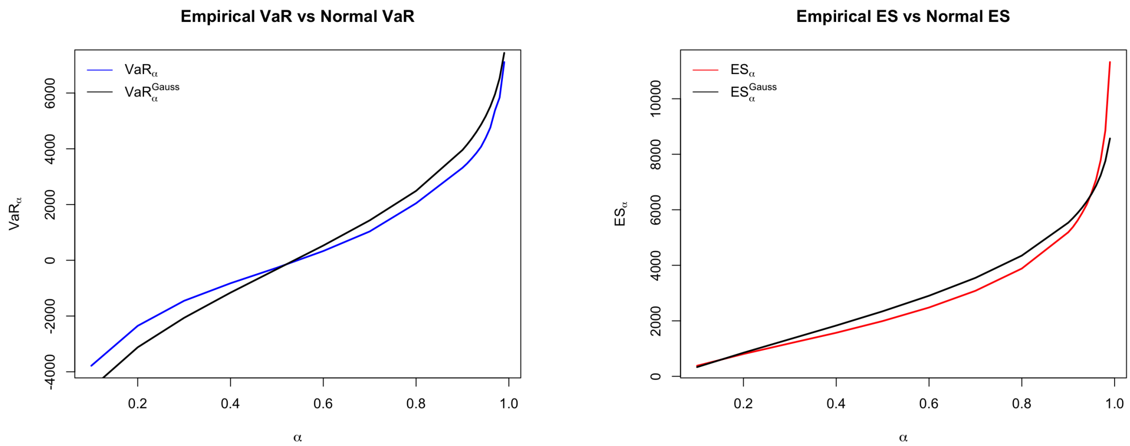


Figure 6: Plot of VaR and ES for the portfolio losses at different α.



(a) Value at Risk

(b) Expected Shortfall

Figure 7: Comparing empirical VaR and ES against normal distribution.

Finally, let us consider $\alpha = 0.8$. The ES at this confidence level is 3884, while the VaR is 2045. By drawing a straight line parallel to the x-axis, we can see the $ES_{0.8}$ curve intersects the VaR curve at $\alpha = 0.93$. This means that if we consider ES at confidence level $\alpha = 0.8$, then the same amount of capital would be required as considering VaR at $\alpha = 0.93$. This is seen in figure 8.

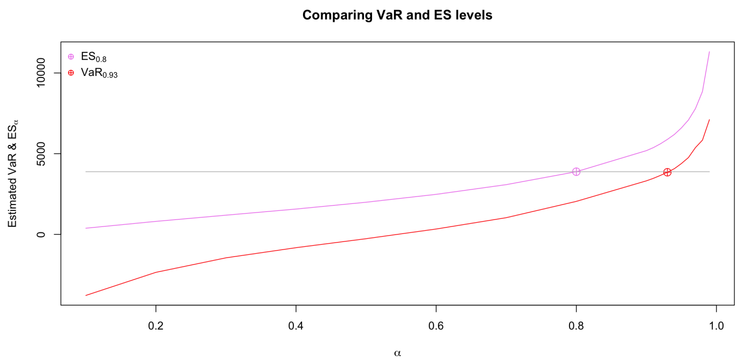


Figure 8: Comparing the $ES_{0.8}$ level with the correspoding VaR level.

2.5 Rank correlations and copula fits

The Pearson correlation provides the simplest approach to describe dependence between random vectors traditionally. It is defined as:

$$\rho_{XY} = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$$

However, the coefficient has its shortcomings such as the fact that it is a measure just for linear dependence or not being invariant nonlinear strictly increasing transformations. Furthermore, it does not describe the structure of the dependence but only its strength. Using instead, the Copula-based approach, gives us a more complete view of the dependence structure. Pairs of stocks from the portfolio will be fitted using different copulas: the Gaussian Copula, the t-Student Copula, the Gumbel Copula and the Clayton Copula. They are fitted to the data using the maximum likelihood method, through the `loglikCopula` function from the copula package, and then optimised to get the ML estimate. The ML estimator is defined as:

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_{\theta\in\Omega} \sum_{i=1}^{n} \ln(c_\theta(u_i, v_i))$$

where Ω is the paramter space, and $c_\theta(u_i, v_i)$ is the underlying copula. Furthermore, rank correlations measures, Kendall's Tau and Spearman's Rho, are implemented to examine the degree concordance of each pair.

Kendall's Tau for random variables X_1 and X_2 is defined as:

$$\rho_\tau(X_1, X_2) = \mathbb{E}[\mathrm{sign}((X_1 - \tilde{X}_1)(X_2 - \tilde{X}_2))]$$

where $(\tilde{X}_1, \tilde{X}_2)$ is independent of (X_1, X_2) but has the same joint cdf. Spearman's rho is defined as the Pearson correlation of the margins:

$$\rho_S(X_1, X_2) = \rho(F_1(X_1), F_2(X_2))$$

One can estimate Kendall's τ by the sample version of Kendall's tau, r_n^τ:

$$r_n^\tau = \frac{1}{\binom{n}{2}} \sum_{1<i_1<i_2<n} \mathrm{sign}\{(Y_{i_1,1} - Y_{i_2,1})(Y_{i_1,2} - Y_{i_2,2})\}$$

An estimator for Spearman's rho is given by the sample correlation computed from componentwise scaled ranks (called pseudo-observations) of the data.

$$r_n^S(X_i, Y_i) = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X}\sigma_{\mathrm{rg}_Y}}$$

where rg_X and rg_Y are the ranks of X_i and Y_i.

Furthermore, given Kendall's τ, one can use the following relationships to find a calibrated parameter fit (estimator) for the 4 tested copulas:

$$\hat{\rho}^{\mathrm{Gauss}} = \hat{\rho}^{\mathrm{t}} = \sin(\frac{\pi}{2} r^\tau) \qquad \hat{\theta}^{\mathrm{Gu}} = \frac{1}{1 - r^\tau} \qquad \hat{\theta}^{\mathrm{Cl}} = \frac{2r^\tau}{1 - r^\tau} \tag{2}$$

this estimator can then be compared to the fitted coefficients from the MLE.

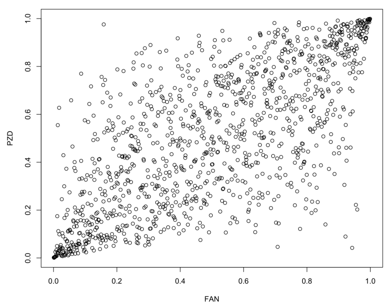
Out of the ${}_{10}C_2 = 45$ possible stock return pairings we decided to focus only on 4 specific ones, namely: ("GE", "PZD"), ("FAN", "PZD"), ("FSLR", "PZD") and ("RUN", "PZD"). The decision was made based on the fact that they posses the highest log-likelihood (> 100) across all fits. Table 6 below presents the estimated parameters for each model.

Looking at the results from the table, we can see that Student-t is the overall best fit for all four pairs, since it has the highest loglikelihoods and the fitted coefficients are the closest to the estimators. It is easily noticeable that all pairs show positive dependence, although there is much higher co-movement in the 'FAN-PZD' pair, which shows stronger tail dependences than the other pairs as well, as shown in the scatterplot in figure 9a.

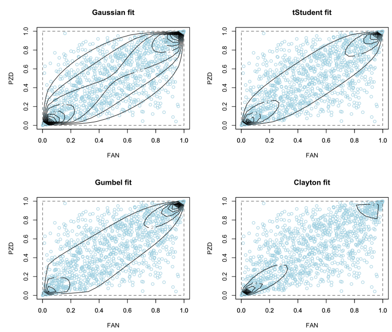
Pair	Param	Gaussian	tStudent	Gumbel	Clayton	Rank Correl.
GE-PZD	Fit coefficient	$\rho = 0.463$	$\rho = 0.438$ $\nu = 2.89$	$\theta = 1.419$	$\theta = 0.817$	$r^\tau = 0.290$ $r^S = 0.410$
	Max LogLikelihood	145	190	149	148	
	Estimator	$\hat{\rho} = 0.440$	$\hat{\rho} = 0.440$	$\hat{\theta} = 1.408$	$\hat{\theta} = 0.817$	
FAN-PZD	Fit coefficient	$\rho = 0.750$	$\rho = 0.734$ $\nu = 4.22$	$\theta = 2.060$	$\theta = 2.154$	$r^\tau = 0.518$ $r^S = 0.697$
	Max LogLikelihood	486	512	467	405	
	Estimator	$\hat{\rho} = 0.728$	$\hat{\rho} = 0.440$	$\theta = 2.077$	$\hat{\theta} = 2.154$	
FSLR-PZD	Fit coefficient	$\rho = 0.532$	$\rho = 0.547$ $\nu = 5.05$	$\theta = 1.531$	$\theta = 1.178$	$r^\tau = 0.370$ $r^S = 0.519$
	Max LogLikelihood	204	228	202	155	
	Estimator	$\hat{\rho} = 0.550$	$\hat{\rho} = 0.550$	$\hat{\theta} = 1.589$	$\hat{\theta} = 1.178$	
RUN-PZD	Fit coefficient	$\rho = 0.469$	$\rho = 0.477$ $\nu = 3.99$	$\theta = 1.427$	$\theta = 0.917$	$r^\tau = 0.314$ $r^S = 0.443$
	Max LogLikelihood	151	184	148	142	
	Estimator	$\hat{\rho} = 0.474$	$\hat{\rho} = 0.474$	$\hat{\theta} = 1.459$	$\hat{\theta} = 0.917$	

Table 6: Table of results after fitting the different pairs to the 4 copulas. The estimators were calculated from equation 2, after computing the (sample version) Kendall tau.

Crude MLE was sufficient for fitting Gauss, tStudent and Gumbel copulas, however, inversion of the Kendall tau estimator was used for Clayton copula fitting. For this reason, the calibrated parameter is exactly equal to the fitted coefficient. This was done by setting the `method` variable to `"itau"` in the `fitCopula` function from the copula package.

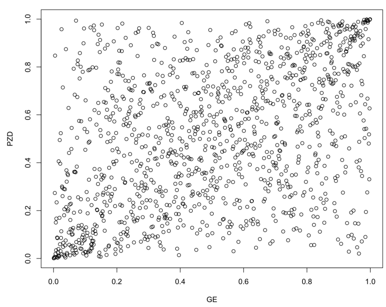


(a) Scatter plot

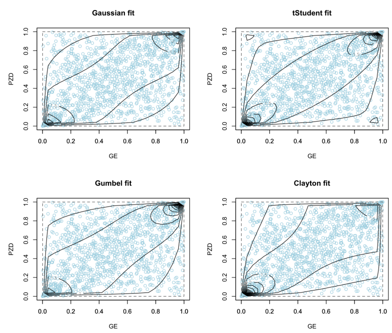


(b) Fitted copula contours

Figure 9: First trust global wind energy (FAN) and Invesco Cleantech ETF (PZD) joint dependence.



(a) Scatter plot



(b) Fitted copula contours

Figure 10: First trust global wind energy (FAN) and General Electric (GE) joint dependence.

2.6 Aggregated Value at Risk

2.6.1 VaR estimates by Monte Carlo simulation

In the previous section, we obtained 4 pairs of stock losses with relatively large dependence, which are:

- Pair 1: ("FAN", "PZD")
- Pair 2: ("GE", "PZD")
- Pair 3: ("FSLR", "PZD")
- Pair 4: ("RUN", "PZD")

We also saw how their dependence structure could be represented most accurately by four t-student copulas respectively. These fitted copulas will be used to explore the aggregated value at risk of each pair. E.g. if we denote the losses of one stock by L_1 and the losses of the other by L_2 such that $(L_1, L_2) \sim C$ for some copula C, we would like to know VaR$(L_1 + L_2)$.

This was done using a Monte Carlo simulation approach (MCS) as follows:

1. Divide all the data into a *test group* (last 99 data points) and a *training group* (the rest of data). Get the best fitting t copula on the training group and simulate 2 arrays of 1500 random numbers with this copula relationship (should be enough to make results stable). These arrays will range from 0 to 1, so they can be seen as values of a cumulative distribution function.
2. Separately simulate each loss via the empirical distribution's quantiles then add them up to get the aggregated loss ($L = L_1 + L_2$).
3. Calculate VaRs under different thresholds α for the simulated aggregated loss.
4. Repeat steps (1-3) above 5000 times and average out the VaR at each threshold over all such iterations. This will be the final aggregated VaR.

The aggregated VaR for the 4 pairs at different confidence levels α is given in Table 7:

(α)	Pair 1	Pair 2	Pair 3	Pair 4
0.80	0.0127	0.0184	0.0226	0.0281
0.82	0.0142	0.0205	0.0253	0.0313
0.84	0.0159	0.0230	0.0283	0.0350
0.86	0.0178	0.0259	0.0317	0.0392
0.88	0.0201	0.0294	0.0356	0.0441
0.90	0.0228	0.0337	0.0401	0.0499
0.92	0.0263	0.0389	0.0455	0.0568
0.94	0.0312	0.0458	0.0527	0.0658
0.95	0.0344	0.0504	0.0574	0.0721
0.96	0.0387	0.0562	0.0636	0.0802
0.97	0.0444	0.0643	0.0721	0.0916
0.98	0.0538	0.0766	0.0852	0.1104
0.99	0.0770	0.1001	0.1101	0.1488

Table 7: Agreggated VaR results for the 4 pairs, calculated by Monte Carlo simulation.

2.6.2 Backtesting by failure ratio method

According to concept of failure ratio created by Kupiec (1995), backtesting for the quoted VaR values can be carried out to test whether we chose an accurate copula model to fit the dependence of losses.

Here we use the *unconditional coverage test likelihood ratio* (Christoffersen, 1998), which has is based on the same theory as the Kupiec Failure Ratio. The objective is to record the number of minus log returns in the test sample that exceed VaR from the MC simulation. This is known as the failure time. Then, one can test if we should accept specific covering ratio (α) of the VaR.

According to the likelihood ration equation:

$$LR_{uc} = -2\log\{(1-p)^{n-x}p^x\} + 2\log\left\{\left(1-\frac{x}{n}\right)^{n-x}\left(\frac{x}{n}\right)^x\right\} \tag{3}$$

where x is the failure time, n is the size of the test sample and $p = 1 - \alpha$ is the null hypothesis. Then LR_{uc} obeys a $\chi^2_{\nu=1}$ distribution.

Confidence level (α)	Pair 1		Pair 2		Pair 3		Pair 4	
	Failure time	**p-value**	**Failure time**	**p-value**	**Failure time**	**p-value**	**Failure time**	**p-value**
0.80	19	0.8399	20	0.9600	21	0.7647	23	0.4301
0.82	17	0.8291	19	0.7595	18	0.9625	21	0.4156
0.84	16	0.9651	16	0.9651	14	0.6081	20	0.2695
0.86	15	0.7440	15	0.7440	14	0.9677	16	0.5436
0.88	11	0.7832	12	0.9704	13	0.7324	14	0.5222
0.90	9	0.7598	11	0.7168	11	0.7168	11	0.7168
0.92	7	0.7285	10	0.4576	10	0.4576	7	0.7285
0.94	5	0.6830	7	0.6621	8	0.4063	7	0.6621
0.95	4	0.6509	5	0.9816	4	0.6509	7	0.3722
0.96	4	0.9837	4	0.9837	4	0.9837	7	0.1580
0.97	3	0.9859	2	0.5440	3	0.9859	5	0.2749
0.98	2	0.9886	1	0.4372	3	0.4958	2	0.9886
0.99	0	0.1583	1	0.9920	2	0.3701	1	0.9920

Table 8: Backtesting by failure ratio method.

Obviously, p-values are all very high and far away from rejecting the null hypothesis. This means the model we fit previously is acceptable for research.

2.6.3 Comparison with the comonotone copula (M)

For comonotone copula (M), namely $(L_1, L_2) \sim M$, the same method was used for calculating VaRs under different thresholds. Similarly, Expected Shortfall (ES) is easy to compute via the empirical loss distribution.

Comparison with the t-copula $(L_1, L_2) \sim C^t$ is performed in figure 11:

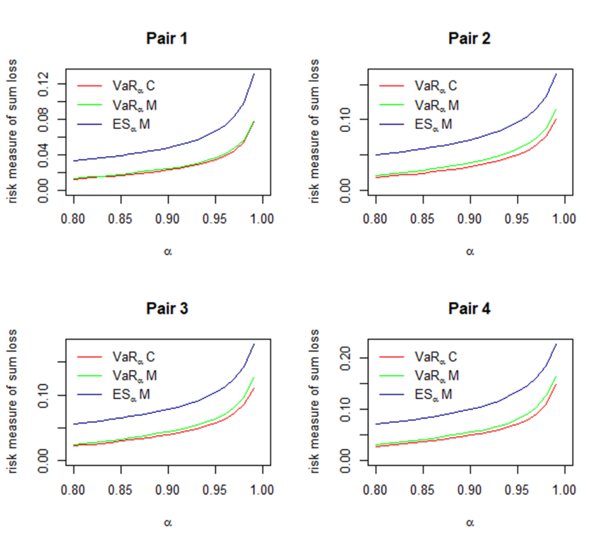
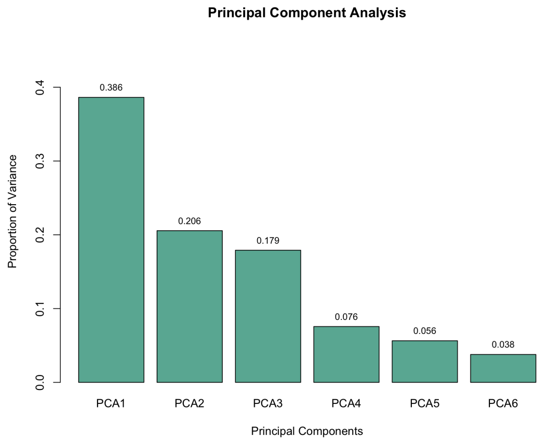


Figure 11: Plots comparing the aggregated VaR assuming a t-copula and a comonotone copula.

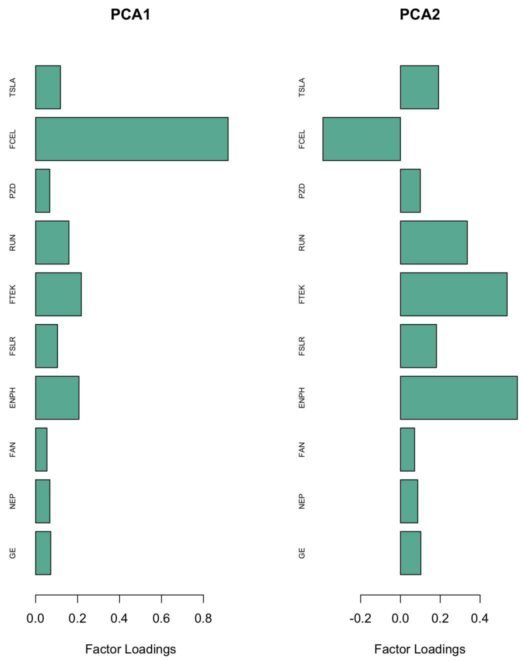
The above plots indicate that $(L_1, L_2) \sim M$ provide an upper bound of all copulas such that VaR with M is larger than any other VaR under the same threshold. Moreover, ES with M is always greater than VaR as per usual.

2.7 Principal component analysis

Principal component analysis is a technique employed to find a series of orthogonal (i.e. uncorrelated) linear combinations that account for most of the variability of the original data. This method was carried out to study which risk factor changes in the portfolio carry the most weight.



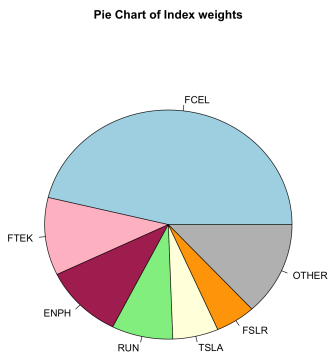
(a) Bar plot of the sample variances l_j of the first six principal components. Above each bar, the proportion of total variance explained by each of the components is given $\frac{l_j}{\sum_{j=1}^{10} l_j}$



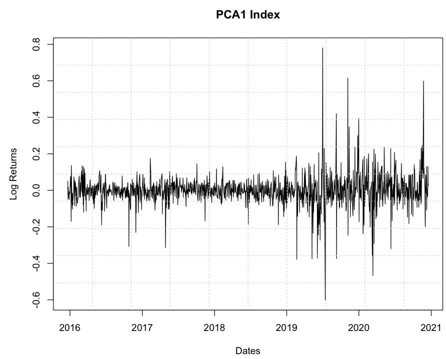
(b) Barplot summarising the loading vectors g_1 (left) and g_2 (right) for the first two principal components.

Figure 12: Principal component analysis for the portfolio of stocks.

Figure 12a shows how almost 60% of the total variance is captured by the first two principal components. Furthermore, from 12b we can note how the first vector of loadings is largely dominated by *FuelCell Energy*. The three most volatile stocks are: *FuelCell, Fuel Tech* and *Enphase Energy*, with annualised volatilities of approximately 3.63, 2.69 and 2.43 respectively. These are precisely the stocks with the largest loadings in PCA1 at 0.92, 0.22 and 0.21 respectively. By examining the second vector of loadings, one can note that *FuelCell* is negatively weighted this time and interestingly, the solar companies as well as *Fuel Tech* have the highest weights.



(a) Pie chart of the first component loadings, which are used as weights in computing the index.



(b) Log returns time series for the index generated by the first principal component.

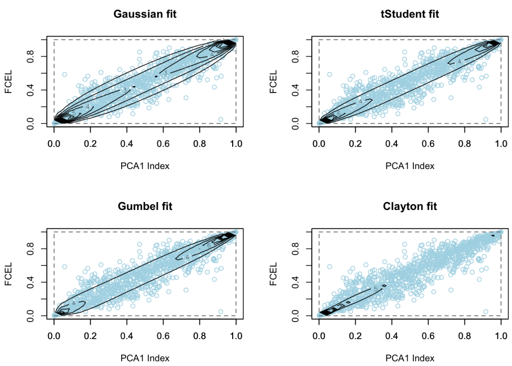
Furthermore, we can note that principal component loadings are all positive, which allows us to construct an index by multiplying our portfolio by the PCA1 eigenvector, and summing over all stocks. The log returns for this index are shown in figure 14b. 14a gives a graphical illustration of index weights by means of a pie chart.

The dependence structure between the index and the individual constituents can be studied by fitting copulas. To this end, three stocks were chosen: the most highly weighted *FuelCell* (FCEL), one with medium weight *Enphase Energy* (ENPH) and the least weighted *First Trust Global Wind Energy* (FAN), with relative weights of 46%, 11% and 3% respectively.

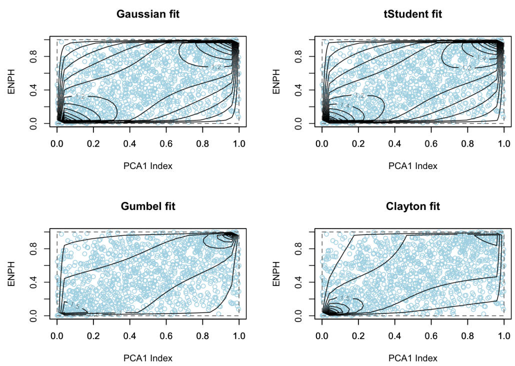
Stock	Param.	Gaussian	tStudent	Gumbel	Clayton	Rank correl.
FCEL	**Fit coefficients**	$\rho = 0.942$	$\rho = 0.94$ $\nu = 3.58$	$\theta = 4.56$	$\theta = 7.25$	$r^\tau = 0.931$ $r^S = 0.784$
	Max LogLikelihood	1261	1325	1280	927	
	Estimator	$\hat{\rho} = 0.943$	$\hat{\rho} = 0.943$	$\hat{\theta} = 4.62$	$\hat{\theta} = 7.25$	
ENPH	**Fit coefficients**	$\rho = 0.394$	$\rho = 0.402$ $\nu = 18.2$	$\theta = 1.31$	$\theta = 0.74$	$r^\tau = 0.394$ $r^S = 0.271$
	Max LogLikelihood	102	104	84	82	
	Estimator	$\hat{\rho} = 0.413$	$\hat{\rho} = 0.413$	$\hat{\theta} = 1.37$	$\hat{\theta} = 0.74$	
FAN	**Fit coefficients**	$\rho = 0.36$	$\rho = 0.35$ $\nu = 14.3$	$\theta = 1.25$	$\theta = 0.55$	$r^\tau = 0.315$ $r^S = 0.216$
	Max LogLikelihood	81	83	69	68	
	Estimator	$\hat{\rho} = 0.332$	$\hat{\rho} = 0.332$	$\hat{\theta} = 1.27$	$\hat{\theta} = 0.55$	

Table 9: The dependence structure between the PCA1 index and each of the stocks was modelled by means of bivariate normal, t-student, Gumbel and Clayton copulas. For each stock, the (sample version) Spearman rho and Kendall tau were calculated. Again, crude MLE was sufficient for fitting Guass, tStudent and Gumbel copulas, but Kendall tau inversion was used for Clayton copula fitting.

As expected, FCEL has very high rank correlations since it's the most dominant stock in the index. A scatter plot of the pseudo-observations for the stock and Index, along with the fitted copula contours is provided in figure 14a. The highest log likelihood maximisation occurs for a t-student copula fit with $\rho = 0.94$ and $\nu = 3.58$, followed by a Gumbel fit with $\theta = 4.56$, which suggests a slight assymetry towards the upper tail. Furthermore, we can see how the t-student fit best recreates the parameter estimator, which confirms this is the best model for the joint dependence.



(a) *FuelCell Energy, Inc.*



(b) Enphase Energy, Inc.

Figure 14: Scatter plots for the pseudo-observations of the PCA1 index and the energy stocks. Contours for the fitted Gaussian, tstudent, Gumbel and Clayton copulas have been overlaid.

A similar plot is provided for *Enphase Energy* in 14b. This stocks sees a significantly smaller dependence with the index ($r^S < 0.3$). Nevertheless, the highest log-likelihood from the studied copulas is once again for the t-student fit, closely followed by the Gaussian model, which is strong evidence for radial symmetry.

3 Conclusion

Climate risk and sustainability play a very important role not only in today's ever-changing world, but also in shaping its future. Even though there have been progress in reducing the risks of climate change and switching to more sustainable policies, like the case of the Paris agreement, the apparent lack of commitment from several countries in respecting the agreement, together with major events such as COVID-19, US elections and Brexit have more or less left the world in the same place, or worse. In this project, we studied how 10 different companies, which focus on sustainability and reducing climate risk, have been affected by all these events.

Our results, show that the companies' stock prices have seen more fluctuations in 2016 and 2020, which can possibly be linked to events such as U.S elections in both years, or the COVID-19 outbreak. Log-returns similarly follow the same trend. Furthermore results from the loss analysis show that maximum losses occurred on 12/03/2020, in line with the first wave of the virus. When we look at the VaR and ES capital requirements as expected ES has higher capital requirements for the same value of α. In order to measure dependence between different pairs of stocks we fitted several copula models to our stock and estimated rank correlations (Kendall's Tau & Spearman's Rho). From the four pairs chosen to describe, results show that the Student t copula is the overall best fit for all pairs. When comparing the copula of aggregated losses for different pairs to the comonotone copula it is worth noting that the comonotone copula provides an upper bound of all copulas, and obviously ES with M is always higher than VaR. Lastly, results from the PCA almost 60% of the total variance is captured by the first two principal components and also that the in the first vector, the most volatile stock FuelCell Energy carries significantly more weight than the other stocks. FCEL also, as expected being the being the most highly weighted component, has very high rank correlations with the PCA1 index.

In general this project is useful since it helps our understanding as to how major events, affect a specific sector as a whole and its component individually, in terms of their correlation and concordance in face of such events.

4 R code

```
# Packages for the whole project
library(normtest)
library(qrmdata)
library(qrmtools)
library(QRM)
library(MASS)
library(xts)
library(copula)

### Question 1 (written by: 12878)

# Importing data and converting into xts format
data5Y   <- read.delim("E:/LSE/St␣429/data5Y.txt")
dat      <- data.frame(data5Y)
rawdata <- dat[dat[,2]!=0&dat[,3]!=0&dat[,4]!=0&dat[,5]!=0&dat[,6]!=0&
                 dat[,7]!=0&dat[,8]!=0&dat[,9]!=0&dat[,10]!=0&dat[,11]!=0,]
date     <- as.Date(as.character(rawdata[, 1]),"%d/%m/%Y")
stocks   <- xts(rawdata[, -1], date)

### Question 2 (written by: 12878)

# Individual stocks
GE    <- stocks[, 1]
NEP   <- stocks[, 2]
FAN   <- stocks[, 3]
ENPH  <- stocks[, 4]
FSLR  <- stocks[, 5]
FTEK  <- stocks[, 6]
RUN   <- stocks[, 7]
PZD   <- stocks[, 8]
FCEL  <- stocks[, 9]
TSLA  <- stocks[, 10]

# Plotting a stock return
plot(zoo(stocks), xlab = "Time", main = "Stock␣prices", col = c("darkblue",
```

```
  "darkgreen", "red", "orange", "violet", "black", "gold", "grey", "brown", "pink"))

# Creating the portfolio and log-returns
portfolio     <- stocks
log_s         <- log(portfolio)
tt            <- nrow(stocks)
lr_portfolio <- diff(log_s)[2:tt, ] #Log returns on portfolio
lr_GE         <- lr_portfolio[, 1]
lr_NEP        <- lr_portfolio[, 2]
lr_FAN        <- lr_portfolio[, 3]
lr_ENPH       <- lr_portfolio[, 4]
lr_FSLR       <- lr_portfolio[, 5]
lr_FTEK       <- lr_portfolio[, 6]
lr_RUN        <- lr_portfolio[, 7]
lr_PZD        <- lr_portfolio[, 8]
lr_FCEL       <- lr_portfolio[, 9]
lr_TSLA       <- lr_portfolio[, 10]

Avg_lr             <- cbind(mean(lr_GE), mean(lr_NEP), mean(lr_FAN), mean(lr_ENPH), mean(lr_FSLR
                       mean(lr_FTEK), mean(lr_RUN), mean(lr_PZD), mean(lr_FCEL), mean(lr_TSLA))
colnames(Avg_lr) <- c("GE", "NEP", "FAN", "ENPH", "FSLR", "FTEK", "RUN", "PZD", "FCEL", "TSLA")

plot(zoo(lr_portfolio), xlab = "Time", main = "Log␣returns", col = c("darkblue",
  "darkgreen", "red", "orange", "violet", "black", "gold", "grey", "brown", "pink"))

### Question 3 (written by: 12878)

init_investment <- rep(1000, 10) # initial investment
price_portfolio <- stocks[1, ] # initial price of stock
No_shares <- int_investment/price_portfolio # number of shares = Lambda
X <- diff(log_s)[2:tt, ] # risk factor changes
CV_portfolio <- sum(as.matrix(stocks[tt, ])*No_shares) # current value of portfolio {current ti
weights_portfolio <- as.matrix(No_shares*as.matrix(stocks[tt, ])/CV_portfolio) # weights of each


loss.sim <- function(Xval, proportion, value){ # Function to create the losses for the portfoli
  # arguments:
  # Xval ... matrix or vector of d risk-factor changes
  # proportion ... row vector of d weights representing the proportion invested in each stock
  # value ... scalar representing the value of the portfolio

  if (is.matrix(Xval)){
    prod <- (exp(Xval)-1) %*% t(proportion)
  } else {
    n <- length(Xval)
    prod <- proportion * (exp(Xval)-1)
  }
  loss <- -value * prod
  return(loss)
}

# Historical losses for the portfolio:
Loss_sim            <- loss.sim(as.matrix(X), weights_portfolio, CV_portfolio)
colnames(Loss_sim) <- c("Historical␣losses")
summary(Loss_sim)
Exp_loss            <- mean(Loss_sim,na.rm=TRUE) # Expected losses
hist(Loss_sim, breaks = 50, xlab = "Historical␣losses",
     main = "Histogram␣of␣Empirical␣losses",col = "light␣blue")

# Normalizing loss variable: Function to normalize the data
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
Loss_sim_standard <- normalize(Loss_sim)

# Lineralizing losses
```

```
losslin.sim <- function(Xval, proportion, value){
  if (is.matrix(Xval)){
    n <- dim(Xval)[1]
    prod <- (Xval) %*% t(proportion)
  } else {
    n <- length(Xval)
    prod <- proportion * Xval
  }
  loss <- -value * prod
  return(loss)
}

# Historical linearized losses for the portfolio
losslin_sim           <- losslin.sim(X, weights_portfolio, CV_portfolio)
colnames(losslin_sim) <- c("Linearized␣␣Losses")
summary(losslin_sim)

hist(losslin_sim, breaks = 50,xlab = "Linearlized␣losses",
     main = "Histogram␣of␣linearlized␣losses", col = "grey")

combinelin_loss <- xts(cbind(losslin_sim,Loss_sim), order.by = date[2:tt])
plot.xts(combinelin_loss, xlab = "Time", ylab = "Price", screens = factor(1, 1),
         auto.legend = TRUE, col = c("darkblue","gold"),
         main = "Plot␣of␣linearlized␣losses␣and␣Historical␣losses")
addLegend(legend.loc = "bottomleft", bty = "n", y.intersp = 1.2,lty = rep(1,2),
          col = c("darkblue","gold"),
          c("Linearized␣losses", "Historical␣losses"))


# Mean and variance of linearnized loss (Theoretical calculation)
muX.hat      <- colMeans(X,na.rm = TRUE)
sigmaX.hat   <- var(X,na.rm = TRUE)
meanLosslin  <- -value_portfolio*sum(weights_portfolio*muX.hat)
varlosslin   <- value_portfolio^2*(weights_portfolio %*% sigmaX.hat %*% t(weights_portfolio))

# Fitting a distribution- normal distribution
set.seed(16)
#Finding the mean and variance of the fitted normal distribution:
fit_normal <- fitdistr(Loss_sim, densfun = "normal")
# Plot histogram for the loss function to check where most values are present:
hist(Loss_sim, breaks = 100, main = "Histogram␣of␣historical␣losses",
     xlab = "Historical␣losses", col = "green")
# Create a random normal distribution with similar mean and variance
fit_normal_data          <- rnorm(nrow(Loss_sim), fit_normal$estimate[1], fit_normal$estimate[2]
fit_normal_combineddata <- xts(cbind(Loss_sim, rnorm(nrow(Loss_sim), fit_normal$estimate[1],
                           fit_normal$estimate[2])), order.by = date[2:tt])
colnames(fit_normal_combineddata) <- c("Historical␣losses", "Normal␣distribution")

# Plotting historical losses and random normal distribution
plot.xts(fit_normal_combineddata, xlab = "Time", ylab = "Price",
         screens = factor(1, 1), auto.legend = TRUE,
         main = "Plot␣of␣historical␣losses␣and␣random␣normal␣distribution" )
addLegend(legend.loc = "bottomleft", bty = "n", y.intersp = 1.2,
          lty = rep(1, 2), col = c("red","black"),
          c("Normal␣distribution","Historical␣losses"))
hist(fit_normal_combineddata[, 2], breaks = 20,
     xlab = "Random␣normal␣distribution",
     main = "Histogram␣of␣Random␣normal␣distribution",
     col = "blue") # Plotting histogram for the random normal distribution

# Q-Q plot & Jarque-Bera test (To check the normality of the log returns(stocks),loss values, p
qqplot(Loss_sim,fit_normal_data,xlab = "Historical␣losses",ylab = "Normal␣distribution",col="Da
title("QQ␣plot") #QQ plot with the random generated normal distribution with same mean and SD
abline(0, 1, col = "grey",lwd = 1)
norm_test_loss <- jb.norm.test(Loss_sim) # Jarque beta test for historical losses
norm_test_lrportfolio <- jb.norm.test(lr_portfolio) # Jarque-Bera test for portfolio
```

```r
# Normal test for log returns of all stocks
norm_test_lrGE   <- jb.norm.test(lr_GE)
norm_test_lrNEP  <- jb.norm.test(lr_NEP)
norm_test_lrFAN  <- jb.norm.test(lr_FAN)
norm_test_lrENPH <- jb.norm.test(lr_ENPH)
norm_test_lrFSLR <- jb.norm.test(lr_FSLR)
norm_test_lrFTEK <- jb.norm.test(lr_FTEK)
norm_test_lrRUN  <- jb.norm.test(lr_RUN)
norm_test_lrPZD  <- jb.norm.test(lr_PZD)
norm_test_lrFCEL <- jb.norm.test(lr_FCEL)
norm_test_lrTSLA <- jb.norm.test(lr_TSLA)


# VaR and ES for Losses -- different levels of alpha -- empirical distribution
alpha <- c(seq(0.1,0.8,0.1), seq(0.9,1,0.01))
VaR.hs <- quantile(Loss_sim, alpha, na.rm = TRUE) # Calculating VaR for empirical distribution
ES.hs <- rep(0, length(alpha)) # To calculate true ES for any distribution
for(i in 1:length(alpha)) {
  values <- Loss_sim[Loss_sim > VaR.hs[i]]
  ES.hs[i] <- mean(values, na.rm = TRUE)
}
ran <- range(VaR.hs, ES.hs, na.rm = TRUE) # creating range for y-axis
plot(alpha, VaR.hs, type = "l", ylim = ran, xlab = expression(alpha), lwd = 2,
     ylab = expression("Estimated VaR & ES"[alpha])) # true ES_alpha
lines(alpha, ES.hs , type = "l", col = "maroon3", lwd = 2) # ES_alpha estimate
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 2),
       col = c("black", "maroon3"), legend = c(expression(VaR[alpha]),
       expression(ES[alpha])))

## Compare them with empirical distribution & normal distribution

# Var and ES for normal distribution

# Calculating VaR for normal distribution:
var_n <- qnorm(alpha,mean = fit_normal$estimate[1], sd = fit_normal$estimate[2])
# Calculating ES for normal distribution:
ES_n <- ESnorm(alpha,mu = fit_normal$estimate[1], sd = fit_normal$estimate[2])

# VaR plot for empirical and normal distribution
plot(alpha, VaR.hs, type = "l", col ="green", xlab = expression(alpha),
     ylab = expression(Var[alpha]), lwd = 2)
lines(alpha, var_n, type = "l", col = "red", lwd = 2)
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 2),
       col= c("green", "red"), legend = c(expression(VaR[alpha]),
       expression(VaR_normal[alpha])))

# Expected shortfall(ES) plot for empirical and normal distribution
plot(alpha, ES.hs, type = "l", col = "blue", ylim = ran,
     xlab = expression(alpha), ylab = expression(ES[alpha]), lwd = 2)
lines(alpha, ES_n, type = "l",col = "gold2",lwd = 2)
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 2),
       col = c("blue", "gold2"), legend = c(expression(ES[alpha]),
       expression(ES_normal[alpha])))

# Fixing alpha = 0.8 and then calculating ES for the empirical distribution
alpha2        <- 0.8
VaR.hs_fixed <- quantile(Loss_sim, alpha2, na.rm = TRUE)
ES.hs_fixed  <- rep(0, length(alpha2))

for(i in 1:length(alpha2)) {
  values          <- Loss_sim[Loss_sim > VaR.hs_fixed[i]]
  ES.hs_fixed[i] <- mean(values, na.rm = TRUE)
}

ran <- range(VaR.hs, ES.hs, na.rm = TRUE)
```

```
plot(alpha, VaR.hs, type = "l", ylim = ran, xlab = expression(alpha),
     col = "violet",lwd = 2, main = expression("VaR at ES"[alpha=0.8]),
     ylab = expression("VaR "[alpha])) # VaR_alpha
points(alpha2, ES.hs_fixed, pch = 17, col = "green", cex = 2)
abline(ES.hs_fixed, 0, col = "grey", lwd = 0.5)
points(0.93, ES.hs_fixed, pch = 18,col = "blue",cex = 2)
abline(v = 0.93, col = "darkgrey", lwd = 0.5)
legend("topleft", bty = "n", y.intersp = 1.2, pch = c(17, 18),
       col = c("green", "blue"),cex = 2, legend = c(expression(ES[alpha=0.8]),
       expression(VaR[alpha=0.93])))

### Question 4 (written by: 15200 (up to line 366) and 10279 (from line 367))

# Extract dates and remove from data frame.
Dates     <- strptime(dat$DATE, "%d/%m/%Y")
dat$DATE  <- NULL # Remove dates from dataframe.

# Convert dataframe to matrix
Prices     <- data.matrix(dat)
LogPrices  <- log(Prices)
LogReturns <- diff(LogPrices)

# Remove First Entry in dates vector:
#   This is done because when we calculate the log returns as the difference
#   in log prices, we do not produce a value at the first data point.
Dates    <- Dates[2:length(Dates)]

# Copula fitting:
FitCopulas <- function(stock1, stock2) {
  Dat <- cbind(LogReturns[, stock1], LogReturns[, stock2])
  colnames(Dat) <- c(stock1, stock2)

  # Delete rows where both stocks have zero return:
  #     These correspond to holidays  (e.g. 25 December)
  Dat <- Dat[Dat[, 1]!=0 & Dat[,2] !=0, ]
  CopulaX <- pobs(as.matrix(Dat))

  par(mfrow = c(1,1))
  plot(CopulaX, xlab = stock1, ylab = stock2)
  par(mfrow = c(2,2))

  print("Sample version of rank correlations:")
  cat("Spearman rho: ", Spearman(CopulaX)[1,2], '\n')
  cat("Kendal tau: ", Kendall(CopulaX)[1,2], '\n')
  cat('\n')

  # -------------------------
  #       Gauss Copula:
  # -------------------------
  cop_model <- normalCopula(dim = 2)
  normFit <- fitCopula(cop_model, CopulaX, method = 'ml')

  cat("Gaussian Fit: ", '\n')
  cat("Rho: ", coef(normFit), '\n')

  rho    <- coef(normFit)
  cat("Calibrated paramter (Spearman): ", 2* sin(pi/6 * Spearman(CopulaX))[1,2], '\n')
  cat("Calibrated paramter (Kendall): ",  sin(pi/2 * Kendall(CopulaX))[1,2], '\n')

  # Overlay fitted copula contours:
  normal <- normalCopula(param = rho, dim = 2)
  plot(CopulaX, xlab="", ylab="", col = "light blue", yaxt="n", main = "Gaussian fit")
  par(new=TRUE)
  contour(normal, dCopula, xlab = stock1, ylab = stock2)

  # Calculate corresponding loglikelihood
```

```
normLL <- loglikCopula(param = coef(normFit), u=CopulaX, copula = normal)
cat("LogLikelihood␣max:␣", normLL, '\n', '\n')

# -------------------------
#       t Copula:
# -------------------------
cop_model <- tCopula(dim = 2)
tFit <- fitCopula(cop_model, CopulaX, method = 'ml')
cat("tStudent␣Fit:␣", '\n')
cat("(Rho,␣ndof):␣", coef(tFit), '\n')

rho <- coef(tFit)[1]
ndof <- coef(tFit)[2]
cat("Calibrated␣paramter␣(Kendall):␣",  sin(pi/2 * Kendall(CopulaX))[1,2], '\n')

# Overlay fitted copula contours:
tstu <- tCopula(param = rho,  df = ndof, dim = 2)
plot(CopulaX, xlab="", ylab="", col = "light␣blue", yaxt="n", main = "tStudent␣fit")
par(new=TRUE)
contour(tstu, dCopula, xlab = stock1, ylab = stock2)

# Calculate corresponding loglikelihood
tstuLL <- loglikCopula(param = coef(tFit), u=CopulaX, copula = tstu)
cat("LogLikelihood␣max:␣", tstuLL, '\n', '\n')

# -------------------------
#    Archimidean Copulas:
# -------------------------
# Gumbel:
cop_model <- gumbelCopula(dim = 2)
gumbFit <- fitCopula(cop_model, CopulaX, method = 'ml')
cat("Gumbel␣Fit:␣",  '\n')
cat("Theta:␣", coef(gumbFit), '\n')

theta <- coef(gumbFit)
cat("Calibrated␣paramter␣(Kendall):␣",  1/(1-Kendall(CopulaX))[1,2], '\n')

# Overlay fitted copula contours:
gumb <- gumbelCopula(param = theta, dim = 2)
plot(CopulaX, xlab="", ylab="", col = "light␣blue", yaxt="n", main = "Gumbel␣fit")
par(new=TRUE)
contour(gumb, dCopula, xlab = stock1, ylab = stock2)

# Calculate corresponding loglikelihood
gumbLL <- loglikCopula(param = coef(gumbFit), u=CopulaX, copula = gumb)
cat("LogLikelihood␣max:␣", gumbLL, '\n', '\n')

# Clayton:
cop_model <- claytonCopula(dim = 2)
clayFit <- fitCopula(cop_model, CopulaX, method = 'itau')

cat("Clayton␣Fit:␣", '\n')
cat("Theta:␣", coef(clayFit), '\n')

theta <- coef(clayFit)
cat("Calibrated␣paramter␣(Kendall):␣", (2*Kendall(CopulaX)/(1-Kendall(CopulaX)))[1,2] , '\n')

# Overlay fitted copula contours:
clay <- claytonCopula(param = theta, dim = 2)
plot(CopulaX, xlab="", ylab="", col = "light␣blue", yaxt="n", main = "Clayton␣fit")
par(new=TRUE)
contour(clay, dCopula, xlab = stock1, ylab = stock2)

# Calculate corresponding loglikelihood
clayLL <- loglikCopula(param = coef(clayFit), u=CopulaX, copula = clay)
cat("LogLikelihood␣max:␣", clayLL, '\n', '\n')
```

```
}

FitCopulas("GE", "TSLA")
FitCopulas("GE", "RUN")
FitCopulas("GE", "NEP")
FitCopulas("GE", "PZD")
FitCopulas("GE", "FCEL")
FitCopulas("GE", "FTEK")
FitCopulas("GE", "PZD")
FitCopulas("GE", "FAN")
FitCopulas("GE", "FSLR")
FitCopulas("NEP", "FAN")
FitCopulas("NEP", "TSLA")
FitCopulas("NEP", "ENPH")
FitCopulas("NEP", "FSLR")
FitCopulas("NEP", "FTEK")
FitCopulas("NEP", "RUN")
FitCopulas("NEP", "PZD")
FitCopulas("NEP", "FCEL")
FitCopulas("FAN", "PZD")
FitCopulas("FAN", "FSLR")
FitCopulas("FAN", "ENPH")
FitCopulas("FAN", "TSLA")
FitCopulas("FAN", "FTEK")
FitCopulas("FAN", "FCEL")
FitCopulas("FAN", "RUN")
FitCopulas("ENPH", "FSLR")
FitCopulas("ENPH", "PZD")
FitCopulas("ENPH", "TSLA")
FitCopulas("ENPH", "RUN")
FitCopulas("ENPH", "FTEK")
FitCopulas("ENPH", "FCEL")
FitCopulas("FSLR", "FTEK")
FitCopulas("FSLR", "RUN")
FitCopulas("FSLR", "PZD")
FitCopulas("FSLR", "FCEL")
FitCopulas("FSLR", "TSLA")
FitCopulas("FTEK", "FCEL")
FitCopulas("FTEK", "RUN")
FitCopulas("FTEK", "PZD")
FitCopulas("FTEK", "TSLA")
FitCopulas("RUN", "TSLA")
FitCopulas("RUN", "FCEL")
FitCopulas("RUN", "PZD")
FitCopulas("PZD", "FCEL")
FitCopulas("PZD", "TSLA")
FitCopulas("FCEL", "TSLA")

### Question 5 (written by: 20411)

## Calculate VaR (& ES) by Monte-Carlo Simulation Method

Loss <- as.matrix(-X)
alpha3 <- c(seq(0.8, 0.99, 0.01))

# Make loss(log return) arrays
gridm <- function(stock){

  grid <- cbind(Loss[, stock[1]], Loss[, stock[2]])
  grid <- grid[grid[, 1] != 0 & grid[, 2] != 0, ] # get rid of non-trading days
  return(grid)

}

# (L1, L2) ~ C
VaRcalC <- function(train, alpha) {
```

```
  # function for calculating VaR when using t-student copula(C)

  CopulaX <- apply(train, 2, edf, adjust = 1)
  tFit <- fitCopula(tCopula(dim = 2), CopulaX, method = 'ml') # best copula models chosen from c

  sumVaRC_sim <- matrix(0, nrow = 5000, ncol = length(alpha))
  for (i in 1:5000) { # repeat 5000 times
    U <- rCopula(1500, tCopula(param = coef(tFit)[1],
                              df = coef(tFit)[2])) # generate two random arrays compelling copu
    L1 <- quantile(train[, 1], U[, 1])
    L2 <- quantile(train[, 2], U[, 2]) # get two loss samples with the copula relation above
    L <- L1 + L2 # get sum of losses sample
    sumVaRC_sim[i, ] <- quantile(L, alpha) # calculate VaR from sum of losses sample
  }

  sumVaRC <- apply(sumVaRC_sim, 2, mean) # use mean of simulated VaRs to approach the real VaR
  table <- rbind(alpha, sumVaRC) # record result
  return(table)

}

# Backtesting
Backtest <- function(test, alpha, sumVaRC) {
  # function for Unconditional Coverage Test Likelihood Ratio

  n <- nrow(test) # total number of test sample losses
  failuretimes <- rep(0, length(alpha))
  LR_uc <- rep(0, length(alpha))
  Pvalue <- rep(0, length(alpha))
  for(k in 1:length(alpha)) {
    x <- sum(apply(test, 1, sum) >= sumVaRC[2, k]) # the number of losses exceeding VaR under sp
    p <- 1-alpha[k] # significance level
    failuretimes[k] <- x # the number of failure times
    LR_uc[k] <- -2*log((1-p)^(n-x)*p^x)+2*log((1-x/n)^(n-x)*(x/n)^x) # Likelihood Ratio of Uncon
    Pvalue[k] <- 1-pchisq(LR_uc[k], df = 1) # LR_uc obeys chi-square(1) distribution
  }

  table <- rbind(failuretimes, LR_uc, Pvalue) # record result
  return(table)
}


# (L1, L2) ~ M
VaREScalM <- function(grid, alpha) {
  # function for calculating VaR and ES when using comonotone copula(M)

  sumVaRM_sim <- matrix(0, nrow = 5000, ncol = length(alpha))
  sumESM_sim <- matrix(0, nrow = 5000, ncol = length(alpha))
  for(i in 1:5000) { # repeat 5000 times
    U <- runif(1500) # generate random uniform distribution numbers
    L1 <- quantile(grid[, 1], U) # use all samples to fit because there is no need for backtesti
    L2 <- quantile(grid[, 2], U) # get two loss samples with the comonotone copula(M) relation
    L <- L1 + L2 # get sum of losses sample
    sumVaRM_sim[i, ] <- quantile(L, alpha) # calculate VaR from sum of losses sample

    for(j in 1:length(alpha)) {
      sumESM_sim[i, j] <- mean(L[L >= sumVaRM_sim[i, j]]) # ES equals to mean of loss values whi
    }
  }

  sumVaRM <- apply(sumVaRM_sim, 2, mean) # use mean of simulated VaRs to approach the real VaR
  sumESM <- apply(sumESM_sim, 2, mean) # use mean of simulated ESs to approach the real ES
  table <- rbind(sumVaRM, sumESM) # record result
  return(table)

}
```

```r
result <- function(stock, alpha) {
  # get well-prepared data and combine all results from functions above

  grid <- gridm(stock)
  train <- seq(1, nrow(grid)-99)
  losstest <- grid[-train, ] # use last 99 samples for backtesting copula model
  losstrain <- grid[train, ] # use the rest of samples for training to get copula model

  C <- VaRcalC(losstrain, alpha)
  bt <- Backtest(losstest, alpha, C)
  M <- VaREScalM(grid, alpha)
  table <- rbind(C, bt, M)
  return(table)

}


## Chosen pairs for research from all combinations in the portfolio

chosen <- rbind(c("FAN", "PZD"), c("GE", "PZD"), c("FSLR", "PZD"), c("RUN", "PZD"))
list1 <- result(chosen[1, ], alpha3)
list2 <- result(chosen[2, ], alpha3)
list3 <- result(chosen[3, ], alpha3)
list4 <- result(chosen[4, ], alpha3)


## Plot for VaR and ES results of all chosen pairs

par(mfrow = c(2, 2))
# Pair 1
plot(list1[1, ], list1[2, ], type = "l", xlab = expression(alpha),
     xlim = c(0.8, 1), ylab = expression("risk␣measure␣of␣sum␣loss"),
     ylim = c(0, max(list1[7, ])), col = "red", main = "Pair␣1")
lines(list1[1, ], list1[6, ], type = "l", col = "green")
lines(list1[1, ], list1[7, ], type = "l", col = "darkblue")
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 3),
       col = c("red", "green", "darkblue"),
       legend = c(expression(VaR[alpha]~C),
                  expression(VaR[alpha]~M),
                  expression(ES[alpha]~M)))

# Pair 2
plot(list2[1, ], list2[2, ], type = "l", xlab = expression(alpha),
     xlim = c(0.8, 1), ylab = expression("risk␣measure␣of␣sum␣loss"),
     ylim = c(0, max(list2[7, ])), col = "red", main = "Pair␣2")
lines(list2[1, ], list2[6, ], type = "l", col = "green")
lines(list2[1, ], list2[7, ], type = "l", col = "darkblue")
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 3),
       col = c("red", "green", "darkblue"),
       legend = c(expression(VaR[alpha]~C),
                  expression(VaR[alpha]~M),
                  expression(ES[alpha]~M)))

# Pair 3
plot(list3[1, ], list3[2, ], type = "l", xlab = expression(alpha),
     xlim = c(0.8, 1), ylab = expression("risk␣measure␣of␣sum␣loss"),
     ylim = c(0, max(list3[7, ])), col = "red", main = "Pair␣3") #
lines(list3[1, ], list3[6, ], type = "l", col = "green")
lines(list3[1, ], list3[7, ], type = "l", col = "darkblue")
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 3),
       col = c("red", "green", "darkblue"),
       legend = c(expression(VaR[alpha]~C),
                  expression(VaR[alpha]~M),
                  expression(ES[alpha]~M)))
```

```
# Pair 4
plot(list4[1, ], list4[2, ], type = "l", xlab = expression(alpha),
     xlim = c(0.8, 1),ylab = expression("risk␣measure␣of␣sum␣loss"),
     ylim = c(0, max(list4[7, ])), col = "red", main = "Pair␣4")
lines(list4[1, ], list4[6, ], type = "l", col = "green")
lines(list4[1, ], list4[7, ], type = "l", col = "darkblue")
legend("topleft", bty = "n", y.intersp = 1.2, lty = rep(1, 3),
       col = c("red", "green", "darkblue"),
       legend = c(expression(VaR[alpha]~C),
                  expression(VaR[alpha]~M),
                  expression(ES[alpha]~M)))


### PCA (written by: 15200)

# PCA analysis:
PCA <- prcomp(LogReturns)
summary(PCA)

# prcomp returns a list with class "prcomp" containing the following components:
# stdev, rotation: matrix of whose cols are eigenvectors, x: centered data, center: mean.
G   <- PCA$rotation # each column is the eigenvector g_i
mu  <- PCA$center
Y   <- PCA$x
var <- (PCA$sdev)^2

# PCA rotation changes the sign of loadings, but the interpretation is the same:
# (see e.g. https://stats.stackexchange.com/questions/88880/does-the-sign-of-scores-or-of-loadi
# revert signs to have positive loadings on PCA1 (to create index later on).
FirstComp  <- -G[, 1]
SecondComp <- -G[, 2]

# Check that sum of variances of PCs equals sum of variances of original data
S <- var(LogReturns)
totalVar <- sum(diag(S))
c(totalVar, sum(var))

# Plot first 6 PCAs relative to total variance:
FirstSix <- head(var/totalVar, n=-4) # pick first 6.

# Choose ylim that'll leave sufficient space above the tallest bar
ylim     <- c(0, 1.2*max(FirstSix))

# Produce bar plot:
plt <-  barplot(FirstSix, xlab = "Principal␣Components", ylim = ylim,
                names=c("PCA1", "PCA2", "PCA3", "PCA4", "PCA5", "PCA6"),
                ylab = "Proportion␣of␣Variance", col ="#69b3a2",
                main = "Principal␣Component␣Analysis")

# Add value on top of bar:
text(x = plt, y = FirstSix, label = round(FirstSix, 3), pos = 3, cex = 0.8, col = "black")

par(mfrow = c(1, 2))
StockNames <- colnames(LogReturns)
barplot(FirstComp, names = StockNames, horiz = TRUE, cex.names = 0.6,
        col = "#69b3a2", main = "PCA1", xlab = "Factor␣Loadings")
barplot(SecondComp, names = StockNames, horiz = TRUE, cex.names = 0.6,
        col = "#69b3a2", main = "PCA2", xlab = "Factor␣Loadings")

# Calculate stock volatilities in 5-year period:
Volatilities <- 100*apply(LogReturns, 2, sd)

sort(Volatilities, decreasing = TRUE)              # 5yr period volatilities
sort(Volatilities, decreasing = TRUE)/sqrt(5)      # Annual volatilites
sort(Volatilities, decreasing = TRUE)/sqrt(5*252) # Daily volatilities
sort(FirstComp, decreasing = TRUE)                 # PCA1 Loadings
```

```r
# Study of dependence structure:
# ------------------------------
# 1. Contruct index:
# Simple Pie Chart of index weights
par(mfrow = c(1, 1))
RelWeights      <- FirstComp/sum(FirstComp)
OrderedWeights <- sort(RelWeights, decreasing=TRUE)
lbls            <- c(names(OrderedWeights)[1:6], "OTHER")
slices          <- c(OrderedWeights[1:6], 1-sum(OrderedWeights[1:6]))
colors          <- c("light␣blue", "pink", "maroon", "light␣green",
                    "light␣yellow", "orange", "gray")
pie(slices, labels = lbls, main = "Pie␣Chart␣of␣Index␣weights", col = colors)

# Construct index by matmult LogRet with FirstComp
Index <- LogReturns %*% FirstComp # matrix multiplication
plot(Dates, Index, type = "l", ylab = "Log␣Returns", main = "PCA1␣Index")
grid(10, 10, col = "lightgray", lty = "dotted", equilogs = FALSE)

par(mfrow = c(1, 1))
plot(Dates, LogReturns[, "FCEL"], type = "s", col = "dark␣blue",
     ylab = "Log␣Returns", main = "FCEL")
plot(Dates, Index, type="s", col = "red",
     ylab = "Log␣Returns", main = "PCA1␣Index")

# Copula fitting:
# Chose 3 stocks with high (FCEL), medium (ENPH) and low (FAN) weights.
# Copula fitting: (use same function in question 4)

# Append PCA1 Index to LogReturns matrix:
LogReturns            <- cbind(LogReturns, Index)
colnames(LogReturns) <- c(StockNames, "PCA1␣Index")

FitCopulas("PCA1␣Index", "FCEL")
FitCopulas("PCA1␣Index", "ENPH")
FitCopulas("PCA1␣Index", "FAN")
```