Modelling dependent extremes of high-dimensional heavy-tailed data using extreme value theory

STA4507 Term Paper

Name: Tian Han **Guan**

Student ID: 998978058

Department: Statistics

Program: MSc

Email: tianhan.guan@mail.utoronto.ca

Abstract

In recent years, extreme value theory (EVT) has been a popular topic not only in traditional natural science fields such as hydrology, but also in the finance and insurance industries. In summary, EVT can be considered as a general tool to model extreme or rare events such as a 1 in 100 year flood or a 1 in 1000 year economic stress scenario. Even though EVT is a great theory to model extreme scenarios, apply the EVT can sometimes be difficult especially in the multivariate setting due to the loss of natural ordering in a multidimensional space. This term paper focuses on how to apply the peaks-over-threshold (POT) method of EVT in both the univariate and multivariate settings. The univariate problem focuses more on details of how to apply the POT method, while the multivariate problem focuses on how to capture the positive tail dependence structure using an extreme-value copula. Both problems use the same simulated dataset which is high-dimensional and heavy-tailed.

---

1. Introduction of the extreme value theory

Extreme value theory (EVT) is the study of portion of the data which has extreme deviations from the mean or median of the distribution. Similar to the central limit theorem which plays an important role in modelling the sum of random variables, the EVT gives us a framework of how to model extreme outliers of a random variable. One major use of the EVT is to predict the probability of an extreme event which is much more extreme than any of the observed sample data value. Two popular methods for applying EVT are the block maxima method and the peaks-over-threshold method.

The block maxima method first divides a sequence of data values into successive intervals (called blocks) and focuses on modelling the maximum value of each block. This method provides a very natural and convenient way of defining extreme values, for example, the monthly rainfall amount data over 50 years can be divided into yearly blocks, and the extreme values will be the annual maximum rainfall amount. However, one major disadvantage of the classical block maxima method is that it does not efficiently use all the data as it throws away all the non-maximum value in each block.

On the other hand, the peaks-over-threshold method defines extreme values as the data points which exceed some high-value threshold and then focuses on modelling the amount of exceedance over the threshold. Compared to the block maxima method, the peaks-over-threshold method has a much more efficient use of the sample data, however, defining a proper threshold seems to be less natural in some cases.

2. The peaks-over-threshold method

In this term paper, we will focus on the peaks-over-threshold method which only looks at the data values above some high-value threshold. Note that compared to the block maxima method, the peaks-over-threshold method allows a much more efficient use of data as well as avoids the problem of choosing an appropriate block size for determining the extreme values.

Definition 1 (Conditional excess distribution function)

Let X be a random variable with some distribution function F (which can be unknown). Then the conditional excess distribution function over some high-value threshold u is defined as follows:

$$F_u(y) = P(X - u \leq y | X > u) = \frac{F(u+y) - F(u)}{1 - F(u)}$$

$$where\ 0 \leq y \leq x_F - u, x_F\ is\ the\ right\ endpoint\ of\ F$$

The random variable $X - u|X > u$ is called the conditional excess random variable

The goal of the peaks-over-threshold method is be able to model the conditional excess random variable $X - u|X > u$. The following theorem gives the foundation of how to do this.

Theorem 1 (Pickands-Balkema-de Hann theorem)

Let $X_1, X_2, \ldots X_n$ be a sequence of independent and identically distributioned (IID) random variables with the common conditional excess distribution function F_u. Then for a large class of underlying distribution functions F (see comment 2) and some high-value threshold u,

$$F_u(y) \to G(y) \; as \; u \to \infty$$

where G is a Generalized Pareto distribution (GPD) with some shape parameter $\xi \in R$ and scale parameter $\sigma \geq 0$.

Comments:

1) It should be noted that in order for the GPD to be the limiting distribution, one needs to choose a sufficiently high-value threshold u.

2) Pickands showed that the GPD is the limiting distribution for excess over threshold if and only if the underlying distribution F is in the maximum domain of attraction (MDA) of one of the extreme value distributions.

Important and interesting facts about the Generalized Pareto distribution (GPD):

1) The cumulative distribution function (CDF) of the GPD G(y) is:

$$G(y) = 1 - \left(\mathbf{1} + \frac{\boldsymbol{\xi}\mathbf{y}}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}} \text{ for } \xi \neq 0, \text{where } \left(\mathbf{1} + \frac{\boldsymbol{\xi}\mathbf{y}}{\boldsymbol{\sigma}}\right) = \max\left(0,1 + \frac{\xi y}{\sigma}\right)$$

$$and \; G(y) = 1 - \exp\left(-\frac{y}{\sigma}\right) \text{ for } \xi = 0$$

Note that in case where $\xi = 0$, the CDF of the GPD is just the CDF of the exponential random variable with mean σ.

Also note that in case where $\xi = -1$, the conditional excess random variable follows a Uniform distribution on $[0, \sigma]$.

2) There is also a 3-parameter definition of the GPD which has an additional location parameter u (which can be interpreted as the threshold value). If so, the CDF of the GPD becomes the following:

$$G(x) = 1 - \left(\mathbf{1} + \frac{\boldsymbol{\xi}(\mathbf{x}-\mathbf{u})}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}} \text{ for } \xi \neq 0, \text{where } \left(\mathbf{1} + \frac{\boldsymbol{\xi}(\mathbf{x}-\mathbf{u})}{\boldsymbol{\sigma}}\right)$$

$$= \max\left(0,1 + \frac{\xi(\mathrm{x}-\mathrm{u})}{\sigma}\right), x \geq u$$

$$and\ G(x) = 1 - \exp\left(-\frac{(x-u)}{\sigma}\right) \text{ for } \xi = 0, x \geq u$$

In this term paper, we will use the 2-parameter definition of the GPD by using the threshold value as the location parameter.

3) The shape parameter ξ of the GPD plays an important role in determining the tail behaviour of the distribution:

i) $If\ \xi < 0, \text{then the GPD has a finite upper bound of } -\frac{\sigma}{\xi}$

ii) $If\ \xi \geq 0, \text{then the GPD has an infinite upper bound}$

4) Duality between the generalized extreme value (GEV) distribution and GPD:

Suppose the block maxima of a dataset is being modelling by a generalized extreme value (GEV) distribution H with parameters (ξ, σ, μ) and the excess over threshold is being modelled by a corresponding GPD G with parameters (ξ', σ'), then the following is true:

i) $\xi' = \xi, \text{i. e. the shape parameter of the GPD is the same as the shape parameter of the GEV}$

ii)$\sigma' = \sigma + \xi(u - \mu)$, i.e. the scale parameter of the GPD can be written as a function of the GEV's shape and scale parameter

3. Extreme-value copula

Although recent works have been done on the multivariate peaks-over-threshold method, in this term paper, the multivariate extreme values are being modelled using a copula model. The major advantage of using a copula model is that it allows the modelling of the marginal and the dependence structure to be done separately, which is particularly useful when the marginal come from different distributions (which in this case fitting a multivariate distribution can be very difficult).

Definition 2 (Copula)

Given a random vector $(X_1, X_2, \dots, X_p)$, with marginal distributions

$F_i(x) = P(X_i < x)\ for\ i = 1, .., p$. A copula model C: $[0,1]^p \rightarrow [0,1]$ of $(X_1, X_2, \dots, X_p)$ is defined as follows:

$C(u_1, u_2, \dots, u_p) = P(U_1 < u_1, U_2 < u_2, \dots, U_p < u_p)$, where $U_i \sim Unif(0,1)\ for\ i = 1, .. p$

Theorem 2 (Sklar's theorem)

Every multivariate distribution function F can be expressed in its marginal $F_i(x) = P(X_i < x)$ and a copula model C. Mathematically, this can be expressed as follows:

$$F(x_1, x_2, \dots, x_p) = C(F_1(x_1), F_2(x_2), \dots, F_p(x_p))$$

Note:

1) There exists a unique copula C if all the marginal $F_i(x)$ are continous

2) The converse of Sklar's theorem is also true. Given a copula C and all the marginal $F_i(x)$, one can always define the joint distribution function F.

Tails of copula models

In general, choosing an appropriate copula model to use is based on which part of the multivariate distribution has the strongest dependency. Therefore, one advantage of copula models versus correlation coefficients is that they can focus on particular sections of the dependency (compared to correlation coefficients which only gives a measure for the overall dependence structure).

Since in this term paper we are interested in modelling the dependence structure of the extreme events (right tails) across all dimensions, we will be using a special class of copula models – the extreme-value copulas which are designed to provide better modellings of the dependency on extreme events (tails). However, it should be noted that the extreme-value copulas can also be good models to capture general positive dependence structures (in a non-extreme value context).

Definition 3 (Extreme-value copula)

A copula C is an extreme-value copula if there exists a copula C_F such that

$$C_F\left(u_1^{\frac{1}{n}}, u_2^{\frac{1}{n}}, \dots, u_p^{\frac{1}{n}}\right) \to C(u_1, u_2, \dots, u_p) \; as \; n \to \infty \; for \; all \; (u_1, u_2, \dots, u_p) \in (0,1)^p$$

The copula C_F is said to be in the maximum domain of attraction (MDA) of C.

Definition 4 (Max-stable)

A copula C is max-stable if for all $(u_1, u_2, \dots, u_p) \in (0,1)^p$ and for every positive integer m

$$C(u_1, u_2, \dots, u_p) = C\left(u_1^{\frac{1}{m}}, u_2^{\frac{1}{m}}, \dots, u_p^{\frac{1}{m}}\right)^m$$

It can be shown that a copula C is an extreme-value copula if and only if it is max-stable.

Remarks:

Definition 3 and 4 tell that the class of extreme-value copulas have strong connections with the class of generalized extreme value (GEV) distributions, which makes sense as the extreme-value copulas are the limits of copulas of the marginal maximum in a multidimensional IID sample.

In section 6, we will be looking at how to use some particular extreme-value copula to solve problems in a multivariate extreme value theory context.

4. Problem setup and simulated data

4.1 Problem setup

The theme of this term paper is to use extreme value theory to model some dependent extremes which come from a high-dimensional dataset where the data in each dimension are heavy-tailed. Here, it is also assumed that the data in each dimension comes from a different distribution, which implies the tail behavior of each dimension can be completely different. The goal is to be able to build models that can 1) predict future extreme quantities in each marginal and 2) capture the dependence structure of the extreme quantities among all dimensions.

4.2 Data simulation process

In this term paper, instead of using any real data, a simulated dataset is being generated and used for the problems. The following procedures are used to simulate the high-dimensional heavy-tailed dataset with dependent extremes:

1. Simulate data in each dimension from a heavy-tailed distribution (with appropriate parameters to make sure the distributions have heavy tails)

-Here, a dimension of 3 is chosen to illustrate that the models work in more than 2 dimensions

-The following distributions (with parameters) are chosen for each dimension:

Dimension 1: $X \sim \text{Pareto}(\alpha=2.5, \theta=100)$

Dimension 2: $Y \sim \text{Burr}(\alpha=1, \gamma=3.5, \theta=150)$

Dimension 3: $Z \sim \text{Lognormal}(\mu=2, \sigma=1.5)$

For each marginal dataset, n = 200,000 samples are simulated. The first n = 100,000 samples will be used as the training set for the models and the remaining n = 100,000 samples will be used as the testing set for testing the model performance.

2. To make the 3 marginal datasets to be correlated (also with a strong tail dependence), a t copula (with df=2 and parameter=0.75) is used to simulate the multivariate distribution. The reason to use the t copula is to make sure the right tails of the three marginal distributions possess enough dependency. This gives n = 200,000 vectors of dimension 3 (where the first n = 100,000 vectors will be used as the training dataset and the remaining n = 100,000 vectors will be used as the testing dataset).

3. Note that because the testing dataset is generated using the same underlying algorithm as the training dataset, they should be following the same multivariate distribution. However, because the marginal are heavy-tailed, the extreme values in the 2 datasets may have some discrepancies due to the limited sample size.

4.3 Real-life problem interpretation of the dataset

Although everything up to this point is entirely artificial, a real-life problem can be associated with the above problem setup and simulated dataset as follows:

ABC insurance company has 3 different lines of business: line X, line Y, and line Z. Underwriters of ABC insurance company record reported aggregated loss amount on an hourly basis for all 3 lines of business, for example, in a given hour of a day, the reported aggregated loss amount is 100 for line X, 200 for line Y, and 150 for line Z (and this can be thought as the realized value of a random vector of dimension 3). At this point, ABC insurance company has a

total number of 200,000 hourly records (that is 200,000 data points in approximately 23 years). Assume that the 3 lines of business are known to be correlated but the losses within each line of business have no serial correlation (that is, the IID assumption holds true for the random vectors). Suppose the first 100,000 hourly records will be used for the training set and the rest 100,000 hourly records will be used for the testing set of the models.

Two problems that can be of interest are:

1) Build a model to predict/estimate the return level for a 1 in 1000 and 1 in 5000 loss observation event for each line of business
2) An insurance payment occurs when a loss is above some threshold (called deductible), and the payment amount is the loss amount above the deductible (excess over threshold). A joint payment occurs at a time when there is a payment for all lines of business, and the joint payment amount is the total of the payment amounts for all lines of business. Build a model to predict/estimate the average, median, 75th percentile, and 90th percentile of the joint payment amount of the company

The first question will be addressed in section 5 and the second question will be addressed in section 6.

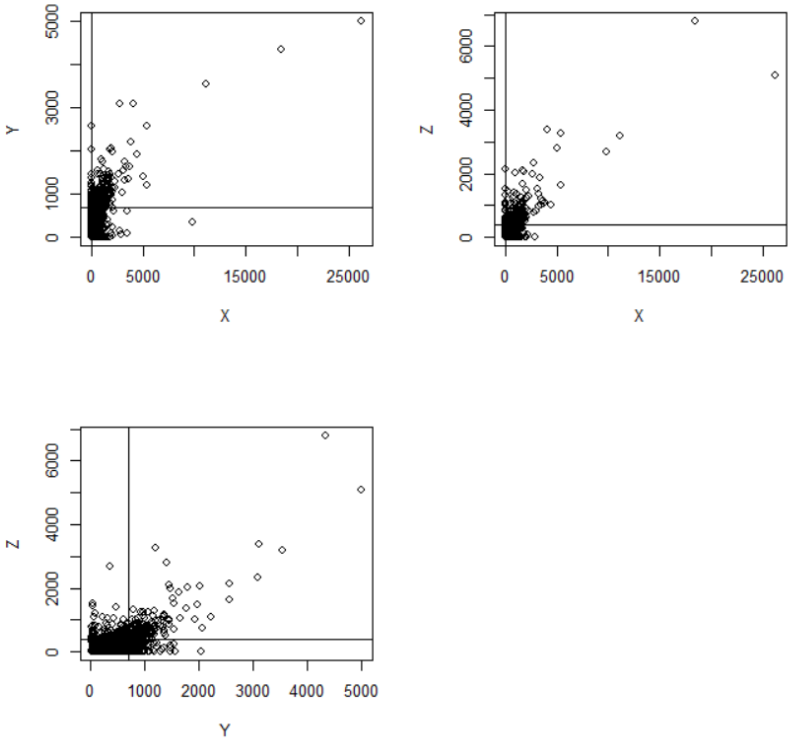
4.4 Data summary

The following table illustrates some important characteristics of the simulated data (the training dataset). From the table it is clear that all three marginal datasets are heavy-tailed (with different tail behaviors).

Table 1: Summary of marginal datasets (training data)

	Skewness	Kurtosis	Count	Max/Mean	Max/99 percentile
X	64.12	8480.35	100,000	394.41	50.02
Y	5.45	117.31	100,000	29.09	9.15
Z	27.60	1831.81	100,000	301.23	28.65

Another key property of the simulated dataset is that the extremes of all marginal datasets are dependent. This can be seen from the pairwise plots below. The upper right box contains all the extremes obtained by the thresholds defined for each marginal (which will be discussed later).



In addition to the pairwise plots above, a correlation matrix using Kendall's tau also shows the dependency across different dimensions.

	X	Y	Z
X	1.0000000	0.7035341	0.7020849
Y	0.7035341	1.0000000	0.7045487
Z	0.7020849	0.7045487	1.0000000

5. Univariate extreme value theory

5.1 Fit a model for each marginal dataset

In this section, we assume for now that the 3 marginal datasets are independent and focusing on building a model on each marginal dataset using the peaks-over-threshold method.

The following steps are used for each of the 3 marginal datasets:

Step 1 Choose a threshold using the mean excess plot. The extreme values will therefore be all data values that are above the chosen threshold.

Step 2 Obtain the conditional threshold exceedance dataset using the threshold in 1)

Step 3 Fit a Generalized Pareto distribution (GPD) using the maximum likelihood estimation (MLE) approach [this includes find the MLE parameters as well as their standard errors]

Step 1 Choose a threshold using the mean excess plot

In order to use the peaks-over-threshold method, the first step is to determine a threshold that can be used to define the extreme values. Picking an appropriate threshold is important as if the threshold is too low then the GPD model may not be a good limiting distribution to use and if the threshold is too high then there will not be a sufficient number of extreme data to fit the GPD.

One popular way of selecting the threshold is by looking at the mean excess plot. Suppose the conditional excess random variable is defined as $X - u|X > u$ (where u is a threshold with $u > 0$), then

$X - u|X > u \sim GPD(\xi, \sigma) \Rightarrow E[X - u|X > u] = \frac{\sigma}{1-\xi} + \left(\frac{\xi}{1-\xi}\right)u$, which is a linear function of u.

One interesting property of the GPD is the **threshold stability property** which states:

Suppose $X - v|X > v \rightarrow GPD(\xi, \sigma)$ *for some threshold* v,

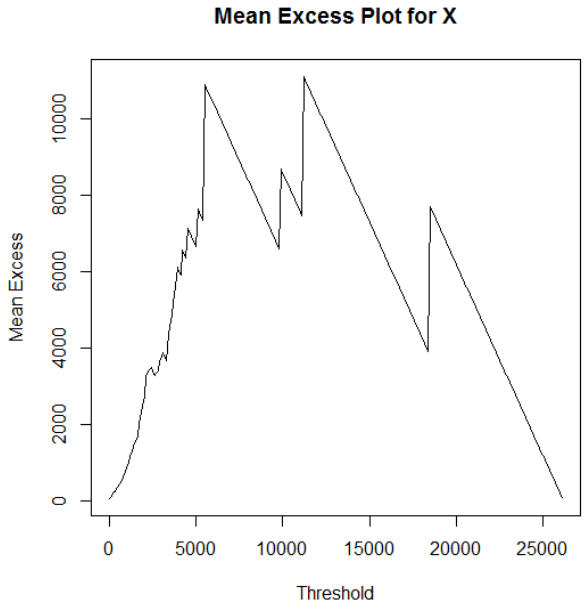
then for any threshold $u > v, X - u|X > u \rightarrow GPD(\xi, \sigma)$

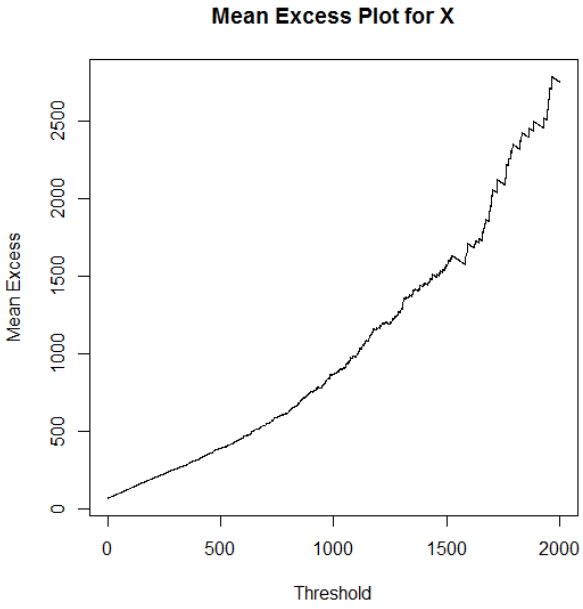
In other words, the same GPD model can be used for any threshold higher than the lowest threshold in which the the conditional excess random variable converges to the GPD. This property is important because all we need to do is to find the lowest threshold u which the mean excess of X over u possesses enough linearity features. This leads to the mean excess plot which provides a graphical method of finding a proper threshold for defining the extremes.

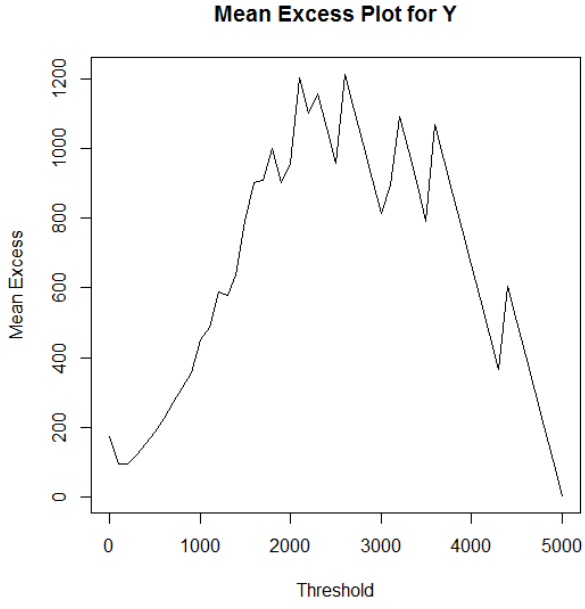
The mean excess plot (also called the mean residual life plot) plots a list of threshold values against their mean excesses, and from the plot, we will look for the lowest threshold u which linearity starts to appear in the plot.

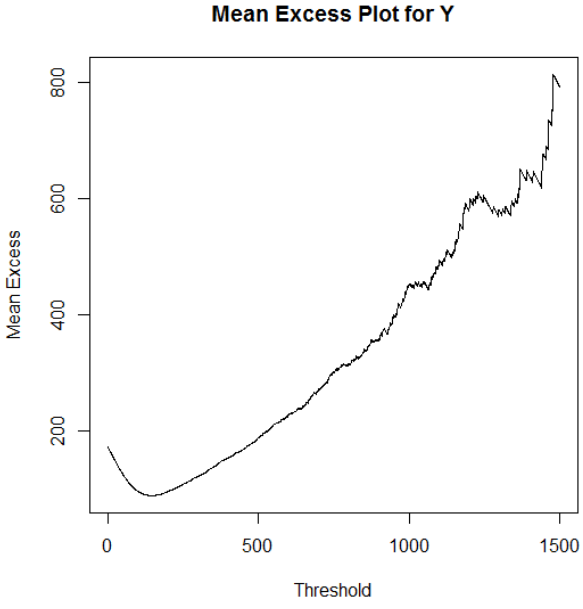
The left-side graphs below plot all possible thresholds against the mean excesses on a rougher scale (every 100 thresholds), while the right-side graphs throw away the large-variance thresholds and only looks at the portions of thresholds which are more linear with a finer scale (every 1 threshold). The right-side mean excess plots can be thought as a close-look version of the mean excess plots on the left.

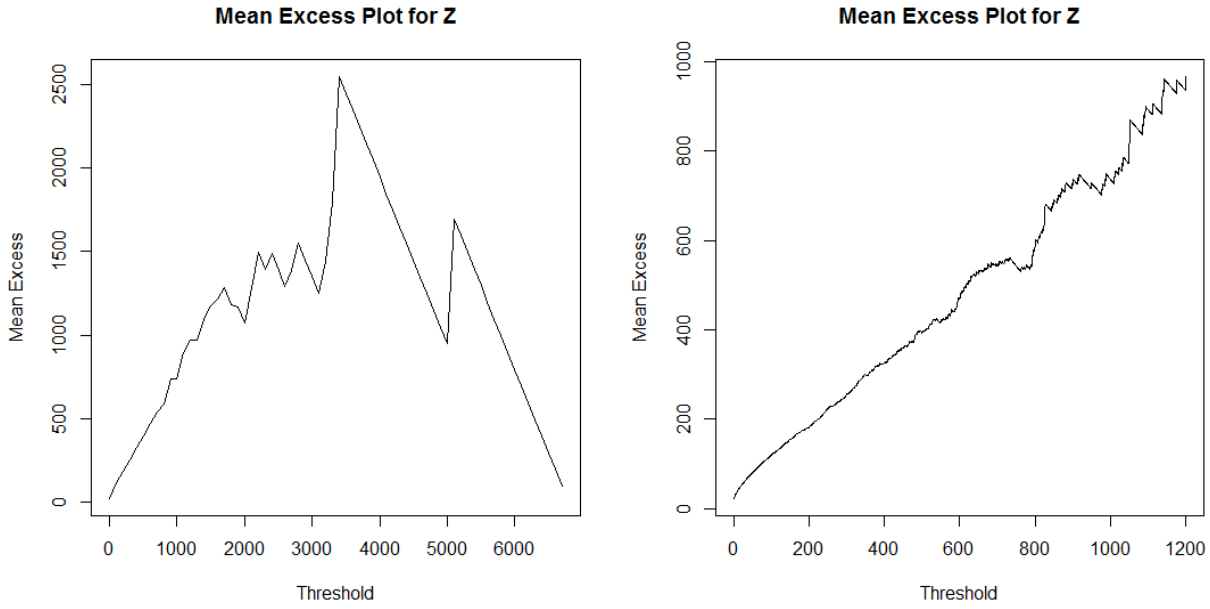
From the mean excess plots, the thresholds for X, Y, and Z are chosen to be as $u_X = 800, u_Y = 700, and\ u_Z = 400$ respectively.


Mean Excess Plot for X
Mean Excess
0
2000
4000
6000
8000
10000
0
5000
10000
15000
20000
25000
Threshold


Mean Excess Plot for X
Mean Excess
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
Threshold


Mean Excess Plot for Y
Mean Excess
0
200
400
600
800
1000
1200
0
1000
2000
3000
4000
5000
Threshold


Mean Excess Plot for Y
Mean Excess
200
400
600
800
0
500
1000
1500
Threshold



Step 2 Obtain the conditional threshold exceedance datasets using the thresholds in 1)

Using the thresholds obtained in step 1 for each of the 3 datasets, the conditional threshold exceedance datasets are obtained and summarized as below. Note that the thresholds chosen give enough number of extreme values to be used in the fitting of the GPDs. Also note that the extreme data in each dimension is also heavy-tailed.

Table 2: Summary of the conditional threshold exceedance datasets

	Count	**Skewness**	**Kurtosis**	**Max/Mean**	**Max/99 percentile**
$X-u_x|X>u_x$	**373**	9.62	113.98	40.94	4.31
$Y-u_y|Y>u_y$	**397**	4.91	35.89	15.87	1.81
$Z-u_z|Z>u_z$	**350**	5.60	46.35	19.58	2.26

Step 3 Fit a Generalized Pareto distribution (GPD) using the maximum likelihood estimation (MLE) approach [this includes find the MLE parameters as well as their standard errors]

The following procedures are used to obtain the maximum likelihood estimates (MLE) (as well as the corresponding standard errors) of the parameters for the GPD:

1) First derive the negative log-likelihood function of the GPD based on a sample of n data points (here assume the shape parameter $\xi \neq 0$, if $\xi = 0$, then we know the GPD is just an exponential distribution with mean σ, and the MLE of σ can be easily obtained by the sample mean of the data).

The Likelihood Function $L(\xi, \sigma|\mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\sigma}\left(\mathbf{1} + \frac{\boldsymbol{\xi} y_i}{\boldsymbol{\sigma}}\right)^{-\left(1+\frac{1}{\xi}\right)} = \frac{1}{\sigma^n}\prod_{i=1}^{n}\left(\mathbf{1} + \frac{\boldsymbol{\xi}\mathbf{y_i}}{\boldsymbol{\sigma}}\right)^{-\left(1+\frac{1}{\xi}\right)}$

$\Rightarrow$ The negative loglikelihood function is $-l = -\log\left(L(\xi, \sigma|\mathbf{y})\right)$

$$= n\log(\sigma) + \left(1 + \frac{1}{\xi}\right)\sum_{i=1}^{n} log\left(\mathbf{1} + \frac{\boldsymbol{\xi}\mathbf{y_i}}{\boldsymbol{\sigma}}\right), \text{where } \mathbf{1} + \frac{\boldsymbol{\xi}\mathbf{y_i}}{\boldsymbol{\sigma}}$$

$$= max(0, 1 + \frac{\xi y_i}{\sigma})$$

2) Due to the form of the negative log-likelihood function, there is no analytic solution for the maximum likelihood estimates. Therefore, the Newton-Raphson method is being used to solve the problem numerically by finding the parameter values which minimize the negative log-likelihood function (which is the same as maximize the log-likelihood function).

Here is how the algorithm looks like:

The problem here is to solve the equation $f(\boldsymbol{\theta}) = \nabla[-l(\boldsymbol{\theta})] = \mathbf{0}, where\ \boldsymbol{\theta} = (\xi, \sigma)$

Step 1: Begin with some initial parameter value $\boldsymbol{\theta}$

Step 2: While $\left|\left|\nabla f(\boldsymbol{\theta})\right|\right| > 0$,

$$Update\ \boldsymbol{\theta}\ with\ \ \boldsymbol{\theta} - H(\boldsymbol{\theta})^{-1}\nabla f(\boldsymbol{\theta})$$

where H is the Hessian matrix and ∇ is the gradient function

The Newton-Raphson method is performed using R's nlm function which gives the solution that minimizes a user-defined function.

Remarks about the Newton-Raphson method:

Although in general Newton's method converges very fast, it is very sensitive to the initial starting values, therefore, a number of initial starting values are used to run Newton's method multiples times in order to make sure it actually converges to the correct solution. Here, the mean excess plots obtained in Step 1 are useful in choosing some "good seeds" as follows:

-First estimate the slope (m) and intercept (b) of the chosen linear section of the mean excess plot (visually)

-Using the fact that $E[X - u|X > u] = \frac{\sigma}{1-\xi} + \left(\frac{\xi}{1-\xi}\right)u$, *solve the following 2 equations*:

$$1)\ m = \frac{\xi}{1-\xi} \Rightarrow \text{gives the starting value for } \xi$$

$$2)\ b = \frac{\sigma}{1-\xi} \Rightarrow \textit{gives the starting value for } \sigma$$

3) Finally, find the standard errors of the MLEs obtained in 2):

i)The Hessian matrix $H(\boldsymbol{\theta}) = \frac{d}{d\boldsymbol{\theta}} l(\boldsymbol{\theta})$, and *the negative Hessian matrix* $(-H)(\boldsymbol{\theta}) = \frac{d}{d\boldsymbol{\theta}}[-l(\boldsymbol{\theta})]$

The Fisher Information Matrix $I(\boldsymbol{\theta}) =$

$-\frac{d}{d\boldsymbol{\theta}} l(\boldsymbol{\theta}), and\ the\ observed\ Fisher\ Information\ Matrix\ I(\boldsymbol{\theta}') =$

$-\frac{d}{d\boldsymbol{\theta}} l(\boldsymbol{\theta}'), [here\ it\ means\ evaluated\ at\ \ \boldsymbol{\theta}'\ where\ \boldsymbol{\theta}'\ is\ the\ MLE\ of\ \boldsymbol{\theta}]$

Therefore, the negative Hessian matrix (second partial derivative of the negative log-likelihood) evaluated at the MLE is the same as the observed Fisher information matrix evaluated at the MLE.

ii) Since the inverse of the Fisher Information Matrix is an asymptotic estimator of the variance-covariance matrix,

$Var(\boldsymbol{\theta}') = I^{-1}(\boldsymbol{\theta}') =$

$(-H)^{-1}(\boldsymbol{\theta}')\ since\ \boldsymbol{\theta}' \sim \boldsymbol{MVN}\left(\boldsymbol{\theta};\ \boldsymbol{I^{-1}}(\boldsymbol{\theta}')\right), where\ \boldsymbol{\theta}'\ is\ the\ MLE\ of\ the\ true\ parameter\ \boldsymbol{\theta}$

iii) Finally, the estimated standard errors of the MLE can be obtained by taking the square root of the diagonal entries. In other words,

$$\text{se}(\boldsymbol{\theta}') = \sqrt{(-H)^{-1}(\boldsymbol{\theta}')}$$

The following table summarizes the fitting results for each of the marginal dataset. The threshold exceedance rate $\boldsymbol{\lambda}$ is defined as $\boldsymbol{\lambda} = \frac{\textbf{\# of values>threshold}}{\boldsymbol{data\ size}}$

Table 3: Summary of estimated GPD parameters

	Shape parameter (ξ)	**Scale parameter (σ)**	**Threshold (u)**	**Threshold Exceedance rate (λ)**
X	**0.55** (se=0.07864)	**267.60** (se=23.99940)	800	0.00373
Y	**0.33** (se=0.06537)	**180.36** (se=14.55839)	700	0.00397
Z	**0.42** (se=0.07478)	**190.74** (se=16.99947)	400	0.0035

5.2 Validate the models

It is always important to validate fitted models in order to make sure the models are accurate with respect to the data. In this section, the accuracy of the 3 fitted GPD models are tested using the PP plot and QQ plot (which are common ways to see the goodness-of-fit of the fitted models).

Definition 5 (PP plot)

A PP (Probability-Probability) plot plots the empirical cumulative distribution function of the ordered sample data $\{x_{(i)}\}$ against the fitted cumulative distribution function F of the ordered sample data $\{x_{(i)}\}$. If the model is a good representation of the true underlying data distribution, then most points in the plot will be close to the straight line y=x.

Mathematically, the plot is

$\left(F(x(i)), \frac{i}{n+1}\right)$, where $\{x(i)\}'s$ are the ordered sample data and $\frac{i}{n+1}$ is the empirical cdf

Definition 6 (QQ plot)

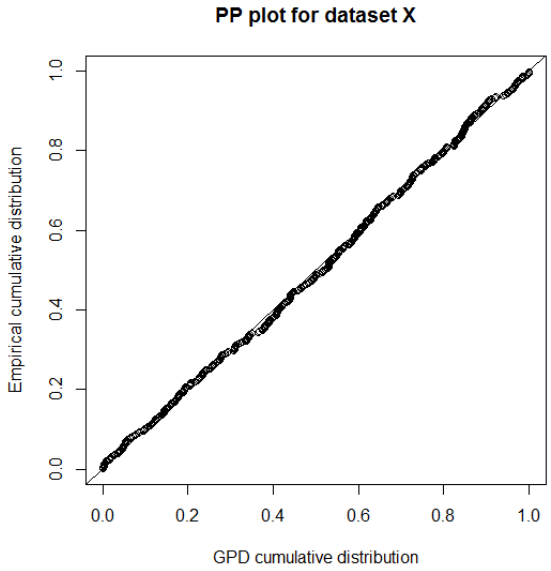
A QQ (Quantile-Quantile) plot plots the fitted distribution's quantiles against the empirical quantiles. If the model is a good representation of the true underlying data distribution, then most points in the plot will be close to the straight line y=x.

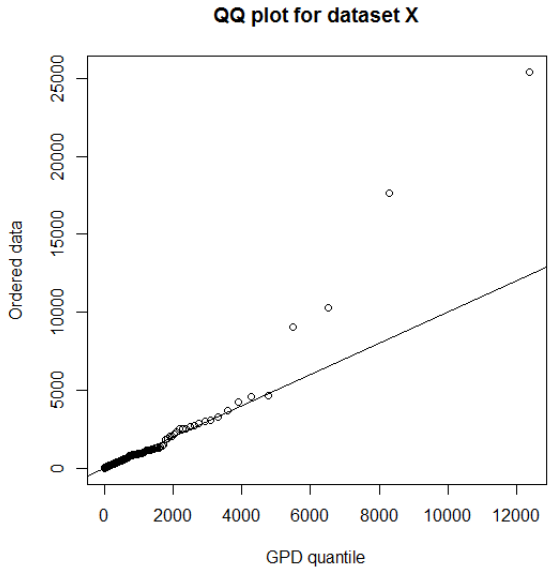
Mathematically, the plot is

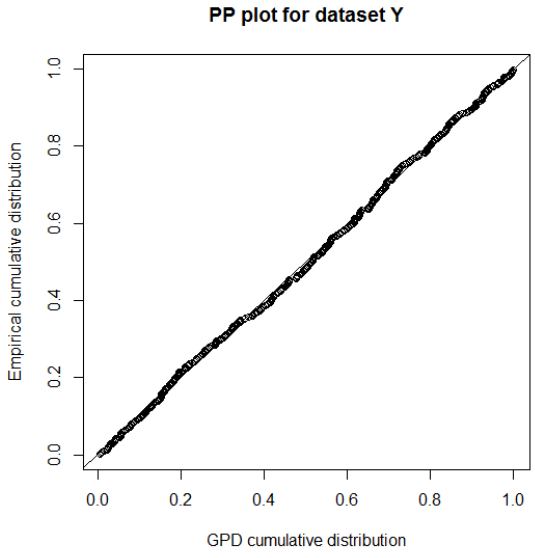
$\left(F^{-1}\left(\frac{i}{n+1}\right), x(i)\right)$, where $\{x(i)\}'s$ are the ordered sample data. Here,

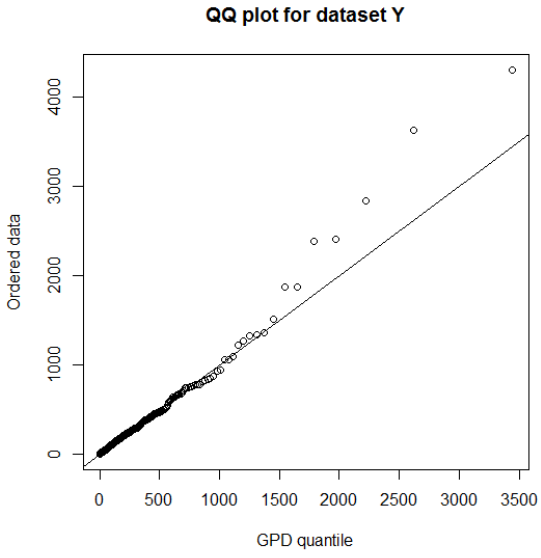
$x(i)$ *represents the empirical estimate of the* $\left(\frac{i}{n+1}\right)$ *quantile*

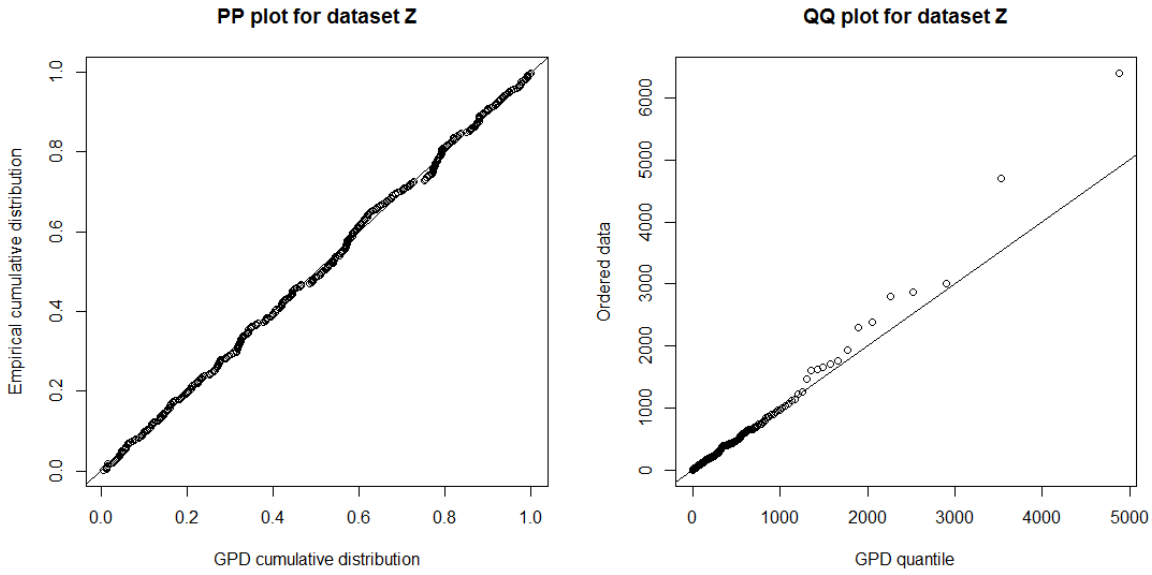
From all 3 pairs of PP plot and QQ plot below, we can conclude that the 3 fitted GPD models provide good fit of the training datasets. Note that the existence of outliers at the far right in the QQ plots tell us that all 3 marginal datasets are heavy-tailed.











5.3 Predict the return levels using the fitted models

Since we conclude all 3 fitted GPD models are good fits for the extremes in each dataset, it is now time to answer the first question proposed in section 4.2, that is:

1) What is the return level for a 1 in 1000 and 5000 loss observation event for each line of business?

To answer this question, first need to derive a formula of calculating the return level estimates using the GPD model.

Definition 7 (Return level)

In extreme value theory, the **return level z_t** represents a 1 in t observation event, that is:

$$\text{P(an observation} > z_t) = \frac{1}{t}$$

In other words, z_t can be understood as the magnitude of the observation that we expect to see once every t observations (or the (1-1/t)th quantile of the data).

How to estimate the return level z_t

Case 1: Assume $X - u|X > u \sim GPD(\xi, \sigma)$ *and* $\xi \neq 0$, using the cumulative distribution function, we know that:

$$P(X - u < y|X > u) = 1 - \left(\mathbf{1} + \frac{\boldsymbol{\xi y}}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}}, \qquad \text{where } \mathbf{1} + \frac{\boldsymbol{\xi y}}{\boldsymbol{\sigma}} = max\left(0, 1 + \frac{\xi y}{\sigma}\right)$$

$$\Rightarrow P(X - u > y|X > u) = \left(\mathbf{1} + \frac{\boldsymbol{\xi y}}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}}$$

$$\Rightarrow P(X > u + y|X > u) = \left(\mathbf{1} + \frac{\boldsymbol{\xi y}}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}}$$

$$\Rightarrow \frac{P(X > u + y)}{P(X > u)} = \left(\mathbf{1} + \frac{\boldsymbol{\xi y}}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}}$$

$$\Rightarrow P(X > u + y) = P(X > u)\left(\mathbf{1} + \frac{\boldsymbol{\xi y}}{\boldsymbol{\sigma}}\right)^{-\frac{1}{\xi}}$$

Substitute the true parameters (ξ, σ) with the estimated paramters (ξ', σ'), let $x = u + y$, and estimate $P(X > u)$ with the threshold exceedance rate $\boldsymbol{\lambda}$ defined before, we can solve the following:

$$P(X > z_t) \approx \lambda\left(1 + \frac{\xi'(z_t - u)}{\sigma'}\right)^{-\frac{1}{\xi'}} = \frac{1}{t}$$

$$\Rightarrow \lambda\left(1+\frac{\xi'(z_t-u)}{\sigma'}\right)^{-\frac{1}{\xi'}}=\frac{1}{t}$$

$$\Rightarrow \text{the return level } z_t = u+\frac{\xi'}{\sigma'}((t\lambda)^{\xi'}-1)$$

Case 2: Assume $X-u|X>u \sim GPD(\xi,\sigma)$ and $\xi=0$, using the cumulative distribution function, we know that:

$$P(X-u<y|X>u) = 1-\exp(-\frac{y}{\sigma})$$

$$\Rightarrow P(X-u>y|X>u) = \exp(-\frac{y}{\sigma})$$

$$\Rightarrow P(X>u+y|X>u) = \exp(-\frac{y}{\sigma})$$

$$\Rightarrow \frac{P(X>u+y)}{P(X>u)} = \exp(-\frac{y}{\sigma})$$

$$\Rightarrow P(X>u+y) = P(X>u)\exp(-\frac{y}{\sigma})$$

Substitute the true parameters σ with the estimated parameter σ', let $x=u+y$, and estimate $P(X>u)$ with the threshold exceedance rate λ defined before, we can solve the following:

$$P(X>z_t) \approx \lambda\exp(-\frac{z_t-u}{\sigma'}) = \frac{1}{t}$$

$$\Rightarrow \lambda\exp(-\frac{z_t-u}{\sigma'}) = \frac{1}{t}$$

$$\Rightarrow \text{the return level } z_t = u+\sigma' log(\lambda t)$$

Using the formula derived above, we can now estimate the 1 in 1000 and 1 in 5000 observation event for each line of business. The following table summarizes the results to the question:

Table 4: Estimated return levels

	1 in 1000 observation event (z_{1000})	1 in 5000 observation event (z_{5000})
Line X	1318.17	2759.30
Line Y	1115.89	1724.91
Line Z	1114.74	1858.29

Up to this point, all we have used is just the 100,000 samples in the training dataset (to fit the model and estimate return levels). Now we can use the 100,000 samples in the testing dataset to see how accurate the predicted return levels are compared to the realized/actual data. The realized/actual return levels are just the empirical quantiles of the testing data. For example, the 1000-observation return level is the same as the 99.9% quantile, and the 5000-observation return level is the same as the 99.98% quantile of the testing data.

The following table compares the estimated return levels vs the realized/actual return levels:

Table 5: Estimated return levels vs actual return levels

	Estimated z_{1000}	Realized z_{1000}	% Diff	Estimated z_{5000}	Realized z_{5000}	% Diff
Line X	1318.17	1305.98	**0.93%**	2759.30	2815.90	**-2.01%**
Line Y	1115.89	1194.64	**-6.59%**	1724.91	1642.10	**5.04%**
Line Z	1114.74	1031.84	**8.03%**	1858.29	1702.82	**9.13%**

Conclusion:

From the table above, we can see that all estimates are within 10% of the realized values. Given the heavy-tailedness of the datasets as well as the remote return levels, the estimates can be considered as reasonably good predictions for the extreme events.

6. Multivariate extreme value theory

6.1 Define the problem

In this section, we will answer the second question proposed in section 4, that is:

> An insurance payment occurs when a loss is above some threshold (called deductible), and the payment amount is the loss amount above the deductible (excess over threshold). A joint payment occurs at a time when there is a payment for all lines of business, and the joint payment amount is the total of the payment amounts for all lines of business. Build a model to predict/estimate the average, median, 75th percentile, and 90th percentile of the joint payment amount of the company

For this question, the following things should be noted:

1) An insurance payment is modelled by the conditional excess random variables $X - u_X | X > u_X, Y - u_Y | Y > u_Y, and\ Z - u_Z | Z > u_Z$, where u_X, u_Y, u_Z are called the deductibles. That is, from an insurer's perspective, given a loss is above the deductible, the insurer will need to pay the insured with the loss amount above the deductible. In this question, we will let the

deductibles to be $u_X = 800, u_Y = 700, u_Z = 400$ which are the same thresholds used in section 5.

2) Here, a joint payment amount is defined as the sum of the payment amounts given all 3 lines of business need to make a payment at the same time, that is, $(x - u_X) + (y - u_Y) + (z - u_Z)$ for some loss event *(x,y,z)* which satisfies $x > u_X, y > u_Y, z > u_Z$. For example, suppose a loss of *(1000,800,650)* has occurred, then the joint payment amount will be *(1000-800)+(800-700)+(650-400)=550*. A loss of *(1000, 800,300)* incurs no joint payment amount as for line Z no payment is going to be paid as it is below the deductible.

3) In this context, a ***multivariate extreme*** can therefore be defined as the event

$$(x > u_X, y > u_Y, z > u_Z)\ where\ (x, y, z) \in (\mathrm{X}, \mathrm{Y}, \mathrm{Z})$$

Using the ***multivariate threshold*** (u_X, u_Y, u_Z), the ***multivariate conditional excess random variable*** can be defined as $(X - u_X, Y - u_Y, Z - u_Z)|(X > u_X, Y > u_Y, Z > u_Z)$

The key difficulty here is how to model the dependence structure of the multivariate extremes. As mentioned before in section 3, in this term paper, we will use the an extreme-value copula - the Gumbel-Hougaard copula to model the dependence structure, where the marginal will still be modelled using GPD models (using GPDs here will be a "loose assumption" as the thresholds are now pre-determined but we will see the fitted GPD models are still good models for each of the marginal).

6.2 The Gumbel-Hougaard (Logistic) Copula

Definition 8 (Archimedean copula)

A copula C is an Archimedean copula if it can be represented by the following form:

$$C(u_1, u_2, \dots, u_p; \theta) = \varphi^{[-1]}(\varphi(u_1; \theta) + \varphi(u_2; \theta) + \cdots + \varphi(u_p; \theta); \theta)$$

where φ is called the generator function.

One reason why Archimedean copulas are popular is because these copulas can be used to model high-dimensional dependence structure with a single parameter θ which can tell the strength of dependence easily.

Definition 9 (Gumbel-Hougaard copula)

The Gumbel-Hougaard (also called Logistic) copula defined by a single parameter θ has the following form:

$$C(u_1, u_2, \dots, u_p) = \exp\left\{-\left[(-\log u_1)^\theta + (-\log u_2)^\theta + \cdots + (-\log u_p)^\theta\right]^{\frac{1}{\theta}}\right\} where\ \theta \geq 1$$

Comments about the Gumbel-Hougaard copula:

1) The Gumbel copula is both extreme-value and Archimedean

2) Since the Gumbel copula is Archimedean, its single parameter θ measures the strength of dependence, where $\theta = 1$ indicates complete independence and $\theta = \infty$ indicates complete dependence

3) In practice, the Gumbel copula is useful in capturing positive tail dependence structures (i.e. strong dependence in the right tail)

6.3 Build a model for the multivariate excess over threshold (joint payment) dataset

Step 1: Obtain the multivariate extreme dataset and the joint payment dataset

Using the definition of a multivariate extreme, the multivariate extreme dataset is obtained (in total out of the 100,000 training samples, 152 multivariate extremes are obtained). Then, by subtracting the multivariate threshold (deductible) $(800, 700, 400)$, the multivariate excess over threshold (or joint payment) dataset is obtained and summarized as below.

Note that the marginal data of the joint payment dataset is again heavy-tailed and there exists strong positive dependence as shown in the rank correlation matrix below (which is a sign to use the Gumbel copula).

Table 6: Summary of the joint payment dataset

*Number of multivariate extremes/joint payments = 152 (out of 100,000 training samples)

**Call the dimensions of the joint payment dataset X*, Y*, and Z* where

$X^* = X - u_X | (X > u_X, Y > u_Y, Z > u_Z)$

$Y^* = Y - u_Y | (X > u_X, Y > u_Y, Z > u_Z)$

$Z^* = Z - u_Z | (X > u_X, Y > u_Y, Z > u_Z)$

	Skewness	Kurtosis	Max/Mean	Mean	Median	3rd quantile
X*	6.84	55.92	23.63	1074.67	489.25	904.31
Y*	3.64	19.29	9.53	274.56	451.41	486.08
Z*	4.31	26.65	12.56	509.16	269.93	576.82

Table 7: Rank correlation matrix of the joint payment data (using Spearman's rho)

```
          payments1 payments2 payments3
payments1 1.0000000 0.5729604 0.5721198
payments2 0.5729604 1.0000000 0.6563331
payments3 0.5721198 0.6563331 1.0000000
```

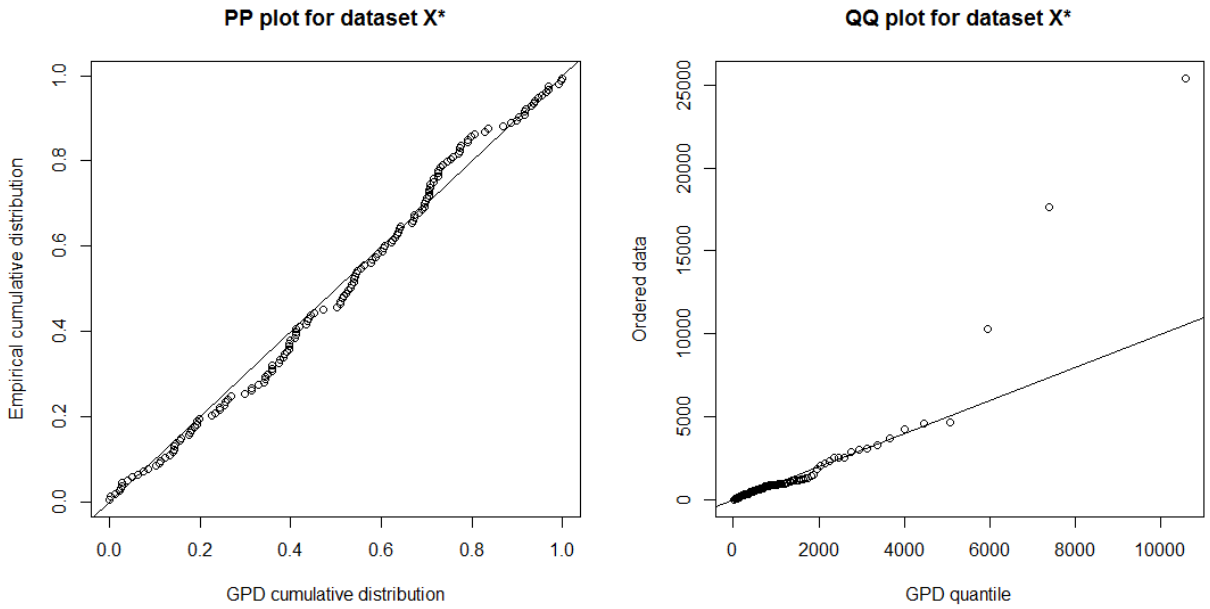
Step 2: Fit a GPD model for each marginal

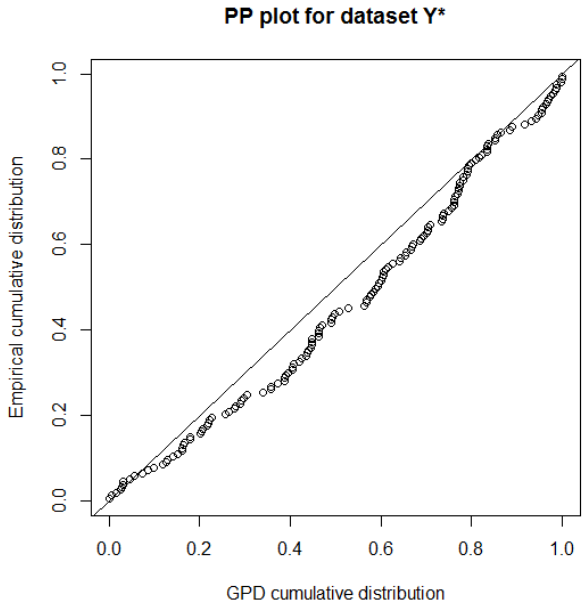
This step is the same as all the steps done in section 5 using the peaks-over-threshold method.

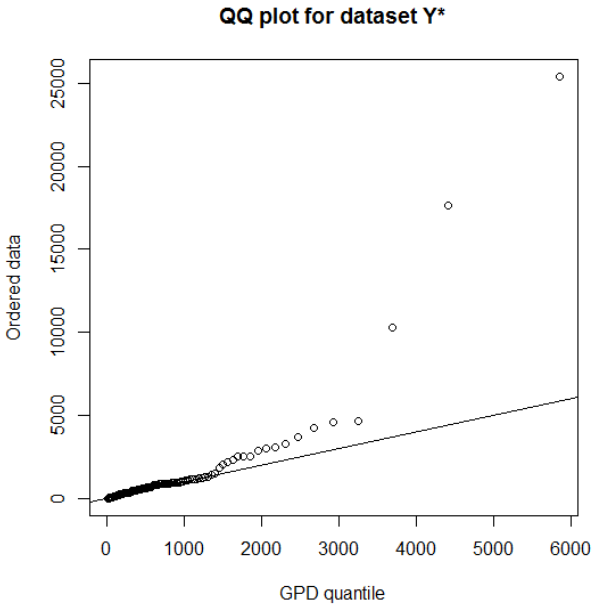
Table 8: Summary of estimated GPD parameters for each dimension of the joint payment dataset

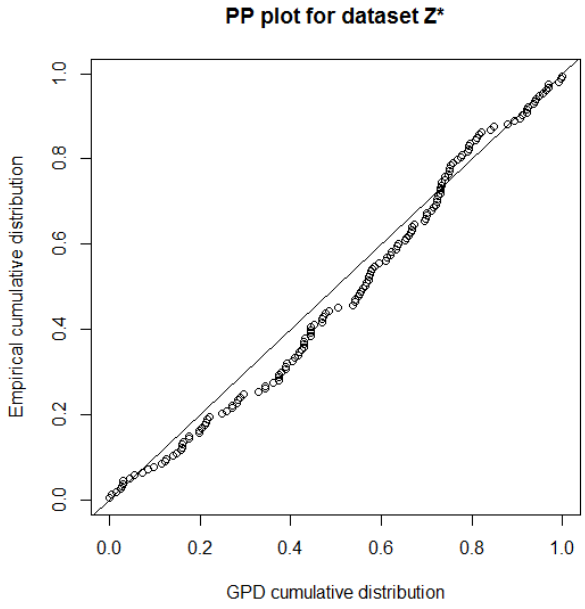
	Shape parameter (ξ)	Scale parameter (σ)
X*	**0.46** (se=0.10556)	**542.20** (se=69.38323)
Y*	**0.20** (se=0.08563)	**358.07** (se=41.80776)
Z*	**0.29** (se=0.09710)	**344.98** (se=42.98866)

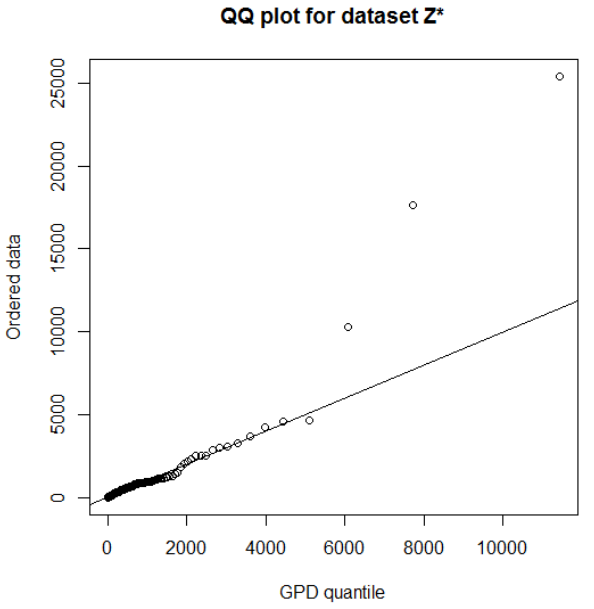
Again PP plots and QQ plots are used to validate the model performance for each dimension's fitting. From all 3 pairs of PP plot and QQ plot below, we can conclude that the 3 fitted GPD models provide good fit of the marginal joint payment datasets. Note that the existence of outliers at the far right in the QQ plots tell us that all 3 marginal are heavy-tailed.




PP plot for dataset Y*
Empirical cumulative distribution
GPD cumulative distribution
0.0
0.2
0.4
0.6
0.8
1.0


QQ plot for dataset Y*
Ordered data
GPD quantile
0
5000
10000
15000
20000
25000
0
1000
2000
3000
4000
5000
6000


PP plot for dataset Z*
Empirical cumulative distribution
GPD cumulative distribution
0.0
0.2
0.4
0.6
0.8
1.0


QQ plot for dataset Z*
Ordered data
GPD quantile
0
5000
10000
15000
20000
25000
0
2000
4000
6000
8000
10000

Step 3: Fit a 3-dimensional Gumbel copula

Estimation of the 3-dimensional Gumbel copula

The most common way of fitting a copula model is by using the maximum likelihood estimation approach. Using the MLE approach, the marginal distributions can be either known or unknown. Here are some very brief summaries on how this approach works:

1) Let a 3-dimensional Gumbel-Hougaard copula be $C(u,v,w), where\ (u,v,w) \in (0,1)^3$

First find the copula density $c(u,v,w) = \frac{\partial^3}{\partial u\, \partial v\, \partial w} C(u,v,w)$

2) Using the multivariate excesses over threshold (joint payment) dataset and the 3 fitted GPD models, obtain the n*3 matrix (where n = # of data points) of pseudo-observations in $(0,1)^3$.

For example, here let the 3-dimensional excesses over threshold dataset be $D = [D_1\ D_2\ D_3]\ where\ D_1, D_2, D_3\ are\ the\ excess\ over\ threshold\ in\ each\ dimension\ X^*, Y^*, and\ Z^*$.

Then, the pseudo-observation matrix $D' = [H_1(D_1)\ \ H_2(D_2)\ H_3(D_3)]$, where H_1, H_2, H_3 are the 3 fitted GPD models in dimension X*, Y*, and Z*.

3) Finally the single parameter θ is being estimated by maximizing the pseudo-likelihood

$$\sum_{i=1}^{n} \log c\big(H_1(D_{1i}),\ H_2(D_{2i})\ H_3(D_{3i})\big), \qquad where\ D_{ji} = i_{th}\ data\ value\ in\ dimension\ j$$

The fitting of the copula is done using R's function fitCopula using the copula package. The following results are obtained:

```
Fit based on "maximum likelihood" and 152 3-dimensional observations.
Copula: gumbelCopula
alpha
1.727
The maximized loglikelihood is 77.75 Optimization converged
```

This tells us that the fitted Gumbel copula has a parameter value (R uses alpha instead of theta) is $\boldsymbol{\theta = 1.727}$ (which indicates the multivariate excesses over threshold dataset has some positive dependence structure as $\theta > 1$).

6.4 Estimate statistical measures using the fitted model

Simulate the multivariate distribution using the copula model and the marginal distributions

Now, we have modelled 1) the marginal of the multivariate excess random variables X*, Y*, and Z* using 3 GPD models as well as 2) the dependence structure across X*, Y*, and Z* using the Gumbel copula, and we are ready to simulate the multivariate distribution of the joint payment. Here, this is done by using R's function Mvdc in the copula package. The Mvdc function uses the following user-defined functions to simulate the multivariate samples:

1) dGPD – the density function of a GPD
2) pGPD – the distribution function of a GPD
3) qGPD – the quantile function of a GPD
4) rGPD – a function to simulate GPD samples by transforming Unif(0,1) samples

500,000 samples of the multivariate distribution are simulated in this case. The sum of each sample (i.e. x*+y*+z*) gives 500,000 simulated joint payment amounts and we can now answer the question:

> What are the average, median, 75th percentile, and 90th percentile of the joint payment amount of the company?

The following table summarizes the results to the question:

Average of joint payment amount	2,566
Median of joint payment amount	1,267
75th percentile of joint payment amount	2,643
90th percentile of joint payment amount	5,306

As in the univariate case, the final step is to compare the estimated results with the realized/actual results obtained using the 100,000 samples in the testing dataset. The realized/actual results are obtained using the following steps:

1) Get the multivariate extreme dataset using the same multivariate threshold

2) Subtract the multivariate threshold from the multivariate extreme dataset to get the multivariate excess over threshold (joint payment) dataset

3) Sum each row of the joint payment dataset to get a list of joint payment amount dataset

4) Obtain the statistical measures

The following table compares the estimated results vs the realized/actual results:

Joint Payment Amount	**Estimated**	**Realized/Actual**
Average	2,566	2,077
Median	1,267	1,114

75th percentile	2,643	2,725
90th percentile	5,306	4,512

Conclusion:

From the table above, we can see that overall the estimates are reasonably close to the realized/actual measures. Given the heavy-tailedness of the multivariate extremes as well as the very limited data size, the estimates can be considered as reasonably good predictions.

Final remarks about using the peaks-over-threshold method in the multivariate setting

As seen in this section, solving problems using the peaks-over-threshold method in multivariate settings is more challenging than in the univariate case. First, compared to define an extreme in the univariate case, define a multivariate extreme can be difficult as we lose the natural ordering of the real line in more than one dimension. It should be noted that in this context the definition of a multivariate extreme is completely arbitrary, that means one can also define a multivariate extreme in another way which can make the modelling of multivariate extremes more difficult. In addition, definition of extreme correlation can be difficult in high dimension case, here, the positive tail dependence structure is captured using a Gumbel copula but in other real applications, choosing an appropriate copula can be also challenging. For instance, the Gumbel copula used in this problem only considers symmetric tail dependence structure, and in case when there is doubt about the symmetry of the dependence structure, other extreme-value copulas may be considered (such as the asymmetric version of the Gumbel copula or Galambos copula).

Appendix (R code)

1) Simulate the high-dimensional heavy-tailed dataset in section 4

```
#Use a t-copula to simulate a correlated high-dimensional heavy-tailed dataset[with significant positive
#tail dependence]
set.seed(12345) #for testing dataset, use set.seed(54321)
mycop <- tCopula(param=0.75, dim=3, dispstr = "ex", df = 2) #define the t-copula
myMvd <- mvdc(copula=mycop, margins=c("pareto", "burr", "lnorm"),   #define the marginals
        paramMargins=list(list(shape=2.5, scale=100),
                list(shape1=1, shape2=3.5, scale = 150),
                list(meanlog = 2, sdlog = 1.5)) )

data <- rMvdc(n=100000,mvdc=myMvd) #simulate the multivariate distribution
colnames(data) <- c("X", "Y", "Z")

#extract each dimension's data
X = data[,1]
Y = data[,2]
Z = data[,3]

#visualize extremal dependence in the right tails
par(mfrow=c(1,3))
plot(X,Y)
plot(X,Z)
plot(Y,Z)
```

2) Define and extract the multivariate extreme values using the multivariate threshold in section 6

```
selection <- function(dat,u1,u2,u3) {
        dataa = NULL
        for (i in 1:dim(dat)[1]) {
```

```
                if ((dat[i,][1]>u1) && (dat[i,][2]>u2) && (dat[i,][3]>u3)) {
                        dataa = rbind(dataa,dat[i,])
                }
        }
        dataa
}
##Obtain the joint payment (multivariate conditional excess over threshold) dataset
#Multivariate extremes
extremes = selection(data,800,700,400) #multivariate extremes

#Multivariate conditional excess over threshold dataset
u1 = 800
u2 = 700
u3 = 400
payments1 = extremes[,1]-u1
payments2 = extremes[,2]-u2
payments3 = extremes[,3]-u3
payments = cbind(payments1,payments2,payments3) #joint payments
```

3) Generate a mean excess plot

```
#Construct a user-defined Mean Excess Plot (also called the Mean Residual Life Plot)
MEplot = function(dat,upper,step) {
        x = NULL
        u = seq(0,upper,step)
        for (i in 1:length(u)) {
                exceedance = dat[dat>u[i]]
                x[i] = mean(exceedance-u[i])
        }
        plot(x~u,type='l',main='Mean Excess Plot for Z',xlab='Threshold',ylab='Mean Excess')
}
```

4) Find the MLE parameters (and their standard errors) of a GPD using the MLE approach

```
#Step 1: Write a function to calculate the negative log-likelihood (NLogL)
gpd.nll = function(theta) {
	shape = theta[1]
	scale = theta[2]
	k = min(1+shape*y/scale)
	n = length(y)
	nll = NULL
	if ((k < 10e-6) || (scale < 0)) {
		nll = -1e10
	}
	else if (scale == 0) {
		nll = n*log(scale)+sum(y)/scale
	}
	else {
		nll = n*log(scale)+(1/shape+1)*sum(log(1+shape*y/scale))
	}
	return(nll)
}

#Step 2: Use the Newton's method to find the parameters that minimize the NLogL
model = nlm(gpd.nll,theta,hessian=TRUE) #for some initial starting parameter value theta

#Step 3: Obtain the standard errors for the parameter estimates
varcov = solve(model$hessian)
se = sqrt(diag(varcov))
```

5) Generate a PP plot for a GPD

```
#Distribution function a GPD
gpd_cdf = function(y,theta) {
	shape = theta[1]
```

```
	scale = theta[2]
	k = max(0,1+shape*y/scale)
	if (shape == 0) {
		gpd = 1-exp(-y/scale)
	}
	else {
		gpd = 1-k^(-1/shape)
	}
	return(gpd)
}

#Plot the PP plot
pp_plot = function(y,theta) {
	ordery = sort(y)
	empcdf = NULL
	gpdcdf = NULL
	for (i in 1:length(y)) {
		empcdf[i] = i/(length(y)+1)
		gpdcdf[i] = gpd_cdf(ordery [i],theta)
	}
	plot(empcdf~gpdcdf,main="PP plot",xlab="GPD cumulative distribution",ylab="Empirical cumulative distribution")
	abline(0,1)
}
```

6) Generate a QQ plot for a GPD

```
#Inverse of the distribution function of a GPD
gpd_inv = function(q,theta) {
	shape = theta[1]
	scale = theta[2]
	if (shape == 0) {
```

```
                gpdinv = -scale*(log(1-q))
        }
        else {
                gpdinv = scale*((1-q)^(-shape)-1)/shape
        }
        return(gpdinv)
}
#Plot a QQ plot
qq_plot = function(y,theta) {
        ordery = sort(y)
        empcdf = NULL
        gpdinv = NULL
        for (i in 1:length(y)) {
                empcdf[i] = i/(length(y)+1)
                gpdinv[i] = gpd_inv(empcdf[i],theta)
        }
        plot(ordery~gpdinv,main="QQ plot",xlab="GPD quantile",ylab="Ordered data")
        abline(0,1)
}
```

7) Use a fitted GPD model to estimate return levels

```
zt = function(u,theta,rate,t) {
        shape = theta[1]
        scale = theta[2]
        if (shape == 0) {
                return(u+scale*log(t*rate))
        }
        else {
                return(u+scale/shape*((t*rate)^shape-1))
        }
}
```

8) Density function, distribution function, quantile function, and random generation function for the GPD

```
#Density function for GPD
dGPD = function(x,shape,scale) {
	den = NULL
	for (i in 1:length(x)) {
		k = max(0,1+shape*x[i]/scale)
		if (shape == 0) {
			gpdden = 1/scale*exp(-y[i]/scale)
		}
		else {
			gpdden = 1/scale*(k^(-(1+1/shape)))
		}
		den = c(den,gpdden)
	}
	return(den)
}
#Distribution function for GPD
pGPD = function(y,shape,scale) {
	cdf = NULL
	for (i in 1:length(y)) {
		k = max(0,1+shape*y[i]/scale)
		if (shape == 0) {
			gpdcdf = 1-exp(-y[i]/scale)
		}
		else {
			gpdcdf = 1-k^(-1/shape)
		}
		cdf = c(cdf,gpdcdf)
	}
	return(cdf)
}
```

```
#Quantile function for GPD
qGPD = function(q,shape,scale) {
	qt = NULL
	for (i in 1:length(q)) {
		if (shape == 0) {
			gpdinv = -scale*(log(1-q[i]))
		}
		else {
			gpdinv = scale*((1-q[i])^(-shape)-1)/shape
		}
		qt = c(qt,gpdinv)
	}
	return(qt)
}
#Random generation function for GPD
rGPD = function(n,shape,scale) {
	sim = NULL
	for (i in 1:n) {
		u = runif(1)
		if (shape == 0) {
			gpdinv = -scale*(log(1-u))
		}
		else {
			gpdinv = scale*((1-u)^(-shape)-1)/shape
		}
		sim = c(sim,gpdinv)
	}
	return(sim)
}
```

9) Fit a Gumbel copula to the joint payment dataset and simulate a joint payment dataset using

the copula and marginal GPDs

```
#First fit the Gumbel copula
#Step 1
m1 = pGPD(payments[,1],theta1[1],theta1[2])
m2 = pGPD(payments[,2],theta2[1],theta2[2])
m3 = pGPD(payments[,3],theta3[1],theta3[2])
m = cbind(m1,m2,m3)
#Step 2
cop_model = gumbelCopula(dim=3)
#Step 3
fit = fitCopula(cop_model,m,method='ml')
theta = coef(fit)

#Simulate the multivariate distribution
myCop = gumbelCopula(dim=3, param=theta)
myMvd <- mvdc(copula=myCop, margins=c("GPD", "GPD", "GPD"),
        paramMargins=list(list(shape=theta1[1], scale=theta1[2]),
                list(shape=theta2[1], scale=theta2[2]),
                list(shape=theta3[1], scale=theta3[2])) )
datasim = rMvdc(50000,myMvd)
payamt = rowSums(datasim)
```

Reference

DAVISON, A. C., & Smith, R. L. (1990). Models for Exceedances over High Thresholds. J. R. Statist. Soc. Retrieved March 3, 2018.

COLES, S., HEFFERNAN, J., & TAWN, J. (1999). Dependence Measures for Extreme Value Analyses. Extremes 2:4. Retrieved March 05, 2018.

Gudendorf, G., & Segers, J. (2009). Extreme-Value Copulas. Retrieved March 10, 2018, from https://arxiv.org/pdf/0911.1015.pdf.

Embrechts, P., Lindskog, F., & McNeil, A. (2001). Modelling Dependence with Copulas and Applications to Risk Management. Retrieved March 15, 2018, from https://people.math.ethz.ch/~embrecht/ftp/copchapter.pdf.

Alice, M. (2015, October/November). Modelling Dependence with Copulas in R. Retrieved March 06, 2018, from https://www.r-bloggers.com/modelling-dependence-with-copulas-in-r/

Extreme Value Theory. (n.d.). Retrieved March 18, 2018, from https://en.wikipedia.org/wiki/Extreme_value_theory

Fawcett, L. (n.d.). Modelling Environmental Extremes. Retrieved March 18, 2018, from http://www.mas.ncl.ac.uk/~nlf8/teaching/mas8391/background/

Bensalah, Y. (n.d.). Steps in Applying Extreme Value Theory to Finance: A Review. Retrieved March 18, 2018, from https://www.banqueducanada.ca/wp-content/uploads/2010/01/wp00-20.pdf