IMPERIAL

Imperial College London
Department of Mathematics

Extreme Value modelling with Missing Data

IOANNIS SPANOS

CID: 02483672

Supervised by Zak Varty and Euan McGonigle

August 30, 2024

Submitted in partial fulfilment of the requirements for the MSc in Statistics at Imperial College London

The work contained in this thesis is my own work unless otherwise stated.

Signed: IOANNIS SPANOS Date: August 30, 2024

Acknowledgements

I would like to express my gratitude to my supervisors Zak Varty and Euan McGonigle for their helpful guidance during the writing of this thesis, and also my fellow students in our research group because the conversations we had helped understand Extreme Value Theory in more depth.

I would also like to thank my family, without the support of whom I would not have been able to complete this piece of work.

Abstract

The General Pareto Distribution (GPD) is the standard way of analysing extreme events in many areas such as finance, insurance, ecology etc. What happens when large amounts of data are missing? Common multiple imputation algorithms focus on the means of the data, and thus do not work well for extreme data that follow the GPD. In this paper we will introduce an algorithm to impute data that follow the GPD. We will then assess its performance using simulations.

1 Introduction

In most cases, the data sets statisticians and applied scientists work with are incomplete. This can happen for many reasons; lack of adequate measuring equipment, loss of parts of the data set through the years or just human errors. Ignoring the missing data (complete case analysis or CCA) is a convenient approach in this case, but it can lead to biased analyses. Another, still naive, approach is to impute each missing datum with a value, e.g. the mean value of the columns it belongs. However, in this case our analysis cannot be accurate, since we will not include the uncertainty of guessing the value properly.

There have been many efforts to overcome the above issues in the past years. Among the data imputation methods proposed, multiple imputation (MI) (Rubin, 2004) is the most popular. It is based on the idea that imputing different values for each missing datum will improve the performance of our estimation. In this way, MI creates several data sets in each of which we conduct our analysis. We then combine our results to get our final estimate. Van Buuren (2018) gives a thorough explanation of why multiple imputation is preferred over other methods.

There are many algorithms that are used to impute data under multiple imputation. The most common ones are based on linear regression with a number of covariates. However, linear regression assumes that data are conditionally normally distributed, which is not true when we work with extreme events.

Extreme values are present in almost every real-life application. They refer to the data that are too large or too small and their behaviour differs from the rest of the data set, i.e. outliers. However, in many applications it is the extreme values that we try to model. This is because we often want to quantify the probability of observing data more extreme than already observed, and of course estimate the extent the data can reach. This has applications in many areas such as finance (Fernandez, 2003), insurance (Adesina et al., 2016), healthcare (Thomas et al., 2016), ecology (Coles and Casson, 1998) and engineering (Holmes and Moriarty, 1999).

Extreme Value Theory (EVT) is a useful tool in this direction. As Coles (2001) points out, it is a unique statistical theory because it models the unusual rather than the common. Indeed, EVT uses asymptotic arguments to fit a known distribution to extreme data. But how do we define which value is extreme and which one is not? The most common way is through the Generalised Pareto Distribution (GPD).

One of the main results of EVT (Theorem 2.1.1) states that, under some regularity conditions, data over a threshold have exceedances that folow the GPD. As a result, this gives us a definition for extreme data (data over a threshold) and their distribution (the GPD). We will see in Section 2 how to determine the threshold and what the closed form of the GPD is.

So extreme value analysis has two steps: determining the threshold over which data are considered extreme and then fitting the GPD to this data. In the case we have missing values, assuming a normal relationship between the data and the covariates will not work for our extreme data, since they follow the GPD. As a result, we need to impute the missing values using the GPD.

The concept of using different distributions than the normal to perform MI is not new. Van Buuren (2018) highlights that, using the GAMLSS package (Rigby and Stasinopoulos, 2005) for fitting a variety of statistical models, we can impute data using any distribution we believe is the most appropriate. Jong and Spiess (2014) and Jong et al. (2016) implement this idea for some distributions. However, both papers exclude the GPD.

The main problem we face when we try to perform MI with the GPD is that we do not know which data are extreme because we do not know their values (if we did they would not be missing). Only the data over a threshold follow the GPD and as a result fitting the GPD to all data will lead in inaccurate estimates.

In this paper we will introduce a way around this. Specifically, we will firstly use logistic regression to determine which data are above the given threshold and then we will perform MI with the GPD. We will use simulated data to assess the performance of this algorithm and compare it to the CCA.

In Section 2 we mention the basic results from EVT and multiple imputation. Even though there is a number of useful books on both topics, we will follow the notation from Coles (2001) for EVT and Van Buuren (2018) for multiple imputation. Section 3 gives an outline of the algorithm we propose and Section 4 tests our method and compares it to the CCA for simulated data. In Section 5 we conclude the results of our analysis, highlight its advantages and drawbacks, and propose further research on the topic.

2 Background Knowledge

Before we introduce our algorithm in Section 3, we will provide an overview of analysing extremes with the GPD and the theory of MI.

2.1 The General Pareto Distribution

Let Y_i, $i = 1, \dots, n$ be a time series of independent and identically distributed (i.i.d.) random variables with marginal distribution function F. The GPD is defined in the following theorem.

Theorem 2.1.1. (Leadbetter, 1983) Let $M_n = \max(Y_1, \ldots, Y_n)$ and suppose that there exist a_n, b_n such that $P(\frac{M_n - b_n}{a_n} \leq z) \to G(z)$, where G is non-degenerate. Then, for u large, we have approximately

$$P(Y - u \leq y | Y > u) = 1 - \left(1 + \frac{\xi y}{\sigma}\right)^{-\frac{1}{\xi}} := H(y) \tag{1}$$

defined on the set $D = \{y : y > 0 \text{ and } 1 + \frac{\xi y}{\sigma} > 0\}$. If $\xi = 0$ then H becomes

$$H(y) = 1 - \exp\left(-\frac{y}{\sigma}\right). \tag{2}$$

We call the distribution with distribution function H the General Pareto Distribution. What Theorem 2.1.1 essentially tells us is that above a specific threshold u, we can approximate the distribution of the exceedances $Y - u$ with the GPD. The shape parameter ξ is key in understanding the qualitative behaviour of the GPD. For $\xi < 0$ the distribution has upper bound $u - \frac{\sigma}{\xi}$, while for $\xi > 0$ and $\xi = 0$ we have no upper bounds. We can see the above more clearly in Figure 1.

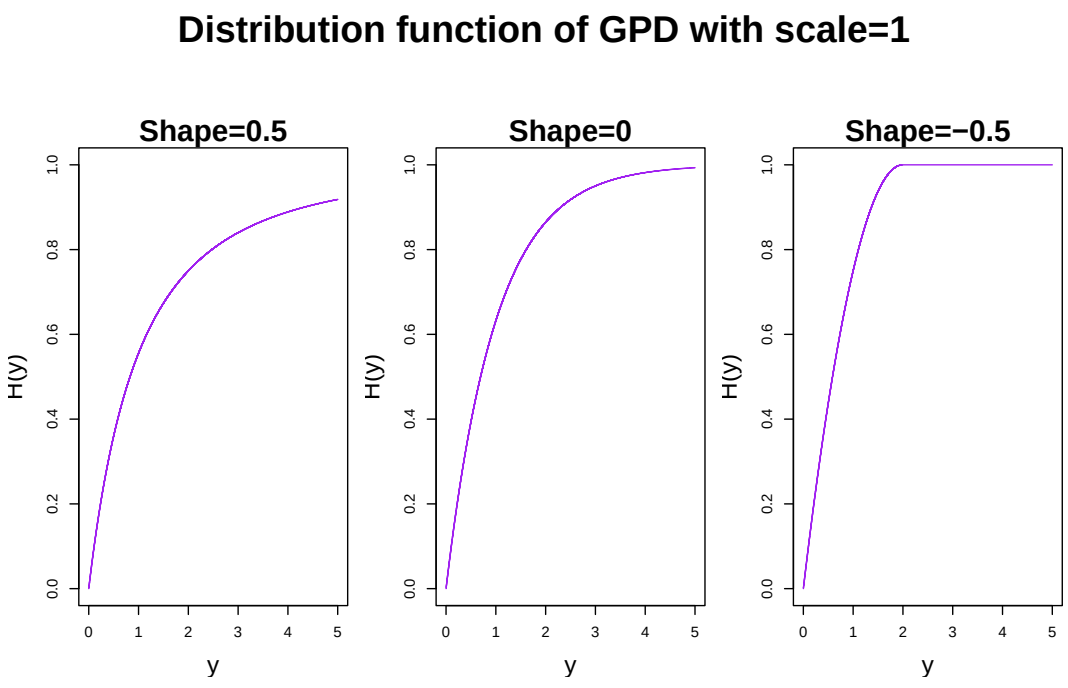

Figure 1: Plots of the distribution function of the GPD with $u = 0$, $\sigma = 1$ and $\xi = 0.5$ (left), $\xi = 0$ (middle) and $\xi = -0.5$ (right).

In Figure 1 we can clearly see that the distributions with $\xi \geq 0$ do not have a maximum (i.e. obtain the value 1), while for $\xi < 0$ we have a maximum in $y = 2$.

The question now is, when we have data that follow the GPD for threshold u how can we estimate the parameters σ and ξ? Even though there are various ways to do this, the robustness and the asymptotic properties of the maximum likelihood estimation (MLE) make it the most common option (Coles, 2001). For $\xi \neq 0$ the likelihood function of the GPD for k excesses y_i of a threshold u is

$$l(\sigma, \xi) = -k \log \sigma - (1 + \frac{1}{\xi}) \sum_{i=1}^{k} \log(1 + \frac{\xi y_i}{\sigma}), \tag{3}$$

while for $\xi = 0$ it is

$$l(\sigma) = -k \log \sigma - \frac{1}{\sigma} \sum_{i=1}^{k} y_i. \tag{4}$$

Functions (3) and (4) cannot be maximised directly and thus we usually need approximating methods.

The main advantage MLE has over other methods of estimation the asymptotic normality of the estimators. This property holds for the parameters of the GPD, apart from the case where $\xi \leq -0.5$ (Smith, 1985). However, this only happens in limited applications. As a result, for $\xi < -0.5$ confidence intervals for the parameters of the GPD can be constructed as usual.

In all the above we have assumed we know the threshold u in Theorem 2.1.1. However, how do we choose u properly? Since, Theorem 2.1.1 is a limit theorem, we want u to be sufficiently high so the data over it can be safely assumed to follow the GPD. However, the higher the threshold the less data we model. This will result in too wide confidence intervals of the estimates, and thus the estimation will not be precise enough. The existence of this trade-off for the values of u require us to be careful when choosing threshold. There are two main heuristic methods used for this reason.

Mean Residual Life (MRL) Plot The MRL plot is based on the fact that the mean of the GPD should change linearly with the threshold after an adequate threshold u_0. The mean of the GPD for $\xi < 1$ is

$$\mathbb{E}(Y) = \frac{\sigma}{1-\xi},$$

while for $\xi \geq 1$ it is undefined. Suppose that $Y_1, \ldots, Y_n$ is a general i.i.d. time series and $u = u_0$ is a threshold that satisfies Theorem (2.1.1). Then every threshold $u > u_0$ should satisfy Theorem (2.1.1). As Davison and Smith (1990) shows, if σ_{u_0} and σ_u are the scales of the GPD with thresholds u_0 and u respectively, then $\sigma_u = \sigma_{u_0} + \xi(u - u_0)$. Using this, for $\xi < 1$ we have

$$\mathbb{E}(Y - u|Y > u) = \frac{\sigma_u}{1-\xi} = \frac{\sigma_{u_0} + \xi(u - u_0)}{1-\xi} = \mathbb{E}(Y - u_0|Y > u_0) + u - u_0. \quad (5)$$

From (5) is shown that after a suitable threshold u the mean of the GPD should change linearly with the threshold. We want to choose the minimum threshold u_0 after which there is linearity in the mean. For this reason the plot of the mean of the GPD for different values of the threshold is useful.

Fitting the GPD for a range of thresholds Like the MRL plot, the threshold range (TR) plot is a useful heuristic tool to choose an appropriate value for the threshold u when modelling the extremes in a time series. It is based on the following two observations.

- The change in threshold u does not change the parameter ξ.
- The fact that for two thresholds u and u_0 with $u > u_0$, we have $\sigma_u = \sigma_{u_0} + \xi(u - u_0)$ (Davison and Smith, 1990). If u_0 is a suitable threshold, then the ξ's for the models with u and u_0 should be the same. Thus, if we define $\sigma^* = \sigma_u - \xi u$, for $u > u_0$ then σ^* should remain constant.

The above provide us with a useful way to choose a threshold: we fit the GPD for a

range of thresholds and we plot the estimated ξ and σ^* over the thresholds. The smaller threshold u_0 after which ξ and σ^* are approximately constant is the one we use for our modelling.

The methods mentioned before are the ones most commonly used for the choice of threshold when modelling extremes with the GPD. Our approach is to produce both the MRL and TR plots and choose the optimal value of threshold that agrees with both. We will see how this is done in Section 4.

2.1.1 Stationary series

All the results mentioned so far are based on the assumption that Y_i are i.i.d. Theorem 2.1.2 ensures that the same same ideas can be applied when the time series is stationary. Firstly, we give this useful definition.

Definition 2.1.1. A stationary series $Y_1, \dots, Y_n$ follows the $D(u_n)$ condition if for all $i_1 < \cdots < i_p < j_1 < \cdots < j_q$ with $i_p - j_1 > l$ we have

$$|P(Y_1 \leq u_n, \dots Y_{i_p} \leq u_n, Y_{j_1} \leq u_n, \dots, Y_{j_q} \leq u_n) -$$

$$-P(Y_1 \leq u_n, \dots Y_{i_p} \leq u_n) P(Y_{j_1} \leq u_n, \dots, Y_{j_q} \leq u_n)| \leq a(n, l),$$

where $a(n, l_n) \to 0$ for l_n such that $\frac{l_n}{n} \to 0$.

Definition 2.1.1 tells us that points that are sufficiently far away in a time series can be treated as independent. This is useful for the following.

Theorem 2.1.2. (Leadbetter, 1983) Let $Y_1, \dots, Y_n$ be a stationary series and $Y_1^*, \dots, Y_n^*$ be i.i.d. with the same marginal likelihood. Let also, $M_n = \max(Y_1, \dots, Y_n)$ and $M_n^* = \max(Y_1^*, \dots, Y_n^*)$. Then, under the condition in Definition 2.1.1, there exist $a_n > 0$ and b_n such that

$$P(\frac{M_n - b_n}{a_n} \leq z) \to G_1(z), \tag{6}$$

where G_1 is non-degenerate, if and only if,

$$P(\frac{M_n^* - b_n}{a_n} \leq z) \to G_2(z), \tag{7}$$

where G_2 is non-degenerate and $G_1 = G_2^\theta$, $0 < \theta \leq 1$.

Conditions (6) and (7) are of the form of Theorem 2.1.1, which are required for the approximation of the distribution of extremes from the GPD. Theorem 2.1.2 essentially says that if the extremes in our stationary series, which are the points over threshold u, are far away in time from each other, then we can assume the GPD.

2.1.2 Non-stationary series

After stationary data we now consider non-stationarity. The distribution of elements in non-stationary time series changes over time or other covariates. Let X be a $n \times p$ matrix of covariates for time series $Y_1, \ldots, Y_n$. Then, in order to include non-stationarity in our analysis we model the parameters of the GPD in the following general form

$$\theta(X) = f(X\beta),$$

where θ can be either σ or ξ, and β is a parameter vector. An example for the scale parameter is $\sigma(t) = \exp(\beta_0 + \beta_1 t)$, where β_0 and β_1 are to be estimated.

It is usually suggested to consider the shape parameter ξ constant over time in order to ensure stability in the behaviour of the GPD (Coles, 2001).

2.2 Multiple Imputation

Multiple Imputation (MI) is the most popular method for imputing data that are missing. We will now introduce the main ideas behind MI that we will use in the following sections. Let us start with some definitions.

Let y be the response data and X the p covariates for the model of interest. We denote the full data with $D = (y, X) = (d_{i,j})_{1 \leq i \leq n, 1 \leq j \leq p+1}$. Some of y or X might be missing so we have that $y = (y_{obs}, y_{mis})$, $X = (X_{obs}, X_{mis})$ and $D = (D_{obs}, D_{mis})$, where *obs* is for observed data and *mis* for missing. We define the matrix $R = (r_{i,j})_{1 \leq i \leq n, 1 \leq j \leq p+1}$ as the response indicator for D. So we have that $r_{i,j} = 0$ if $d_{i,j}$ is missing and $r_{i,j} = 1$ else. The distribution of R with its parameter ψ is called the *missing data model*. Generally, the missing data model may depend on D_{obs} and D_{mis}. The following definition is important for what follows.

Definition 2.2.1. With the above notation we say that:

- The data are missing completely at random (MCAR) if the distribution of R does not depend on D, i.e.

$$P(R = 0|D, \psi) = P(R = 0|\psi).$$

- The data are missing missing at random (MAR) if the distribution of R does not depend on the missing data or

$$P(R = 0|D, \psi) = P(R = 0|D_{obs}, \psi).$$

- The data are missing not at random (MNAR) if $P(R = 0|D, \psi)$ cannot be simplified.

Of course the above also hold for $P(R = 1|D, \psi)$ in each case.

In Definition 2.2.1 it is evident that MCAR is the simplest case and thus most simple imputation methods work for MCAR. As Van Buuren (2018) points out, when we have MCAR or MAR data MI is a useful tool. The case of MNAR is a bit more complex and requires further knowledge about the data. For our purpose we will assume that the data we work with are MAR at worst.

2.2.1 Rubin's rules

Suppose we want to estimate a parameter vector Q using the data D. Let $\hat{Q}$ be an estimate of Q if we observed all values. Also, let U be the covariance matrix of $\hat{Q}$. Then, MI aims that $\hat{Q}$ is unbiased and confidence valid (Rubin, 1996). This means that

$$\mathbb{E}(\hat{Q}|D) = Q \text{ (unbiased)}$$

and

$$\mathbb{E}(U|D) \geq V(\hat{Q}|D) \text{ (confidence valid).}$$

However, we cannot compute $\hat{Q}$, because of the missing data. This is why we need an estimate $\bar{Q}$ of $\hat{Q}$. MI works iteratively by imputing the missing data according to the observed values.

The main results of this section are motivated by the Bayesian approach. The purpose of our analysis is to know the posterior distribution $Q|D_{obs}$ of Q. It can be shown that the mean of $Q|D_{obs}$ is

$$\mathbb{E}(Q|D_{obs}) = \mathbb{E}(\mathbb{E}[Q|D]|D_{obs}). \tag{8}$$

Equation (8) suggests the following idea for estimating Q. Assume we have m iterations. From each of these we get an estimate $\hat{Q}_l$ of Q. So an estimate $\bar{Q}$ of $\hat{Q}$ is

$$\bar{Q} = \frac{\sum_{l=1}^{m} \hat{Q}_l}{m} \tag{9}$$

Note that $\bar{Q}$ as defined in (9) is an unbiased estimate of $\hat{Q}$.

For a statistical analysis to be complete we need to provide an estimate of the variance of Q. Using the Law of Total Variance, the posterior variance of $Q|D_{obs}$ is

$$V(Q|D_{obs}) = \mathbb{E}[V(Q|D)|D_{obs}] + V[\mathbb{E}(Q|D)|D_{obs}]. \tag{10}$$

In (10) the first component $\mathbb{E}[V(Q|D)|D_{obs}]$ refers to the variance within each imputation, while the second component $V[\mathbb{E}(Q|D)|D_{obs}]$ is the variance between the different

imputations. For m imputations we have that

$$\bar{U} = \frac{\sum_{l=1}^{m} U_l}{m},$$

is an unbiased estimate of the within-variance, while

$$B = \frac{\sum_{l=1}^{m} (\bar{Q} - Q_l)'(\bar{Q} - Q_l)}{m - 1}$$

is an unbiased estimate of the between-variance.

Using the above Rubin (1987) proves that the total variance of $\bar{Q}$ is

$$T = \bar{U} + \left(1 + \frac{1}{m}\right) B. \tag{11}$$

Equation (11) indicates that variance in $\bar{Q}$ comes from three sources:

1. The within-variance $\bar{U}$
2. The between-variance B
3. The simulation variance $\frac{1}{m}B$, because we are estimating Q using m imputations.

The steps described in this section are called Rubin's Rules (Rubin, 1987) and ensure that the results of the MI can be considered valid.

2.2.2 Assessing MI algorithms

So far we have seen how to construct an MI algorithm in order to yield valid results (in the sense of subsection 2.2.1). But when we have an MI algorithm, how will we assess its performance? Our main tool for evaluating MI procedures is simulation study (Van Buuren, 2018).

Our approach is as follows: we simulate data that follow a model with parameters Q. We then treat some of our data as missing (either MCAR, MAR or MNAR) and we

perform MI to gain an estimate $\bar{Q}$ of Q. The above steps have to be repeated in order to ensure the robustness of the algorithm. Suppose we repeat our simulations k times. In order to assess our algorithm we use the following performance measures.

- **Percentage of Bias (PB)**: The percentage of bias is used to estimate how important the difference between our estimate and the true value is. It is defined as

$$PB = 100 \times |\frac{\mathbb{E}\bar{Q} - Q}{Q}|,$$

or for k simulations

$$PB = 100 \times |\frac{\frac{\sum_{i=1}^{k} \bar{Q}_i}{k} - Q}{Q}|.$$

The performance of an algorithm is considered sufficient if the PB is less than 5% (Demirtas et al., 2008).

- **Coverage Rate (CR)**: The coverage rate is the percentage of confidence intervals of $\bar{Q}$ that contain the true value Q. An algorithm is considered acceptable if the CR is around the same as the nominal rate of the confidence intervals. For example, for 95% confidence intervals a CR of less than 90% is concerning.

- **Average Width (AW) of confidence intervals:** The average width of the confidence intervals over the number of simulations k is important to ensure that our algorithm can provide adequate precision when estimating Q. For example, too wide AW will result in high CR but it is of no practical use for estimating Q, as it contains too many values.

The above measures of efficiency will be useful in Section 4 where we will assess the performance of the algorithm we propose.

2.2.3 Normal MI algorithm

In the previous section we provided the main theory of MI. It is time to see how MI works in practice. We will give the example of the linear regression which is commonly

used. Then, in Section 3 we will introduce the proposed algorithm for extreme data.

Suppose we want to fit a linear regression with response the univariate variable Y and covariate the univariate variable X using sample (y, x):

$$y_i = \beta x_i + \epsilon_i, \epsilon_i \sim N(0, \sigma^2),$$

For this simple example we will assume that only y has missing values. MI will be based on linear regression for the observed data, so suppose we have

$$y_{obs,i} = \beta_{obs} x_{obs,i} + \epsilon_i, \epsilon \sim N(0, \sigma^2). \tag{12}$$

In (15), x_{obs} refers to the covariate of y_{obs}. If x_{mis} are the (observed) covariate of y_{mis}, then in each iteration l, $l = 1, \ldots, m$ of MI we sample $y_{mis,i}$ from $N(\beta_{obs} x_{mis,i}, \sigma^2)$ and we get the imputed data set D_{imp}.

Then, we perform linear regression

$$Y_{imp,i} = \beta_l X_{imp,i} + \epsilon_i$$

to estimate β_l. Repeating the above procedure for $l = 1, \ldots, m$ we get the estimate $\bar{\beta}$ of β using 2.2.1.

The above framework accounts for the variability in the imputed values, however, it does not include the uncertainty of estimating β in each iteration. In order to resolve that we use bootstrap (Efron, 1979).

Let n_1 be the size of the observed response data y_{obs} and n_0 the size of the missing data vector y_{mis}. Then, the bootstrap normal MI works as shown below.

Algorithm 2.2.1. (Heitjan and Little, 1991)

1. Generate a bootstrap sample (y'_{obs}, x'_{obs}) of size n_1 of (y_{obs}, x_{obs}).

2. Estimate β' from $y'_{obs} = \beta' x'_{obs} + \epsilon'$.

3. Generate vector z of size n_0 so that $z \sim N(0,1)$.

4. Sample n_0 elements from $y_{imp} = \beta' x_{mis} + z\sigma'$ and create the data set D_{imp} which contains (y_{obs}, x_{obs}) and (y_{imp}, x_{mis})

5. Estimate β_l from the data set D_{imp}.

6. Repeat for imputation $l = 1, \ldots, m$.

Algorithm 2.2.1 accounts both for the variability in the missing values but also for the uncertainty of estimating β_l, $l = 1, \ldots, m$.

3 A method for Multiple Imputation of extremes

We saw in Section 2 that with algorithm 2.2.1 and Rubin's Rules we can perform multiple imputation when we have that Y follows the normal distribution conditionally on X.

However, this is not true for data that follow the GPD. Let's assume that $\theta(X) = (\sigma(X), \xi(X))$ and that $Y \sim GPD(\theta(X))$. Then we can perform MI in a similar way as in Algorithm 2.2.1 but now instead of the normal distribution we will use the GPD.

Following the notation of Section 2 let $D = (y, X)$ be the entire data set, $D_{obs} = (y_{obs}, X_{obs})$ be the observed data and $D_{mis} = (y_{mis}, X_{mis})$ be the missing data. As before we will assume that we have missing values only in the response. Let n_1 be the length of y_{obs} and n_0 the length of y_{mis}. We can do MI on the GPD data y_{mis} with Algorithm 3.0.1.

Algorithm 3.0.1. 1. Generate a bootstrap sample (y'_{obs}, X'_{obs}) of size n_1 of (y_{obs}, X_{obs}).

2. Estimate θ' from $y'_{obs} \sim GPD(\theta'(X'_{obs}))$.

3. Sample n_0 elements from $y_{imp} \sim GPD(\hat{\theta}'(X_{mis}))$ and create the data set D_{imp} which contains (y_{obs}, X_{obs}) and (y_{imp}, X_{mis}).

4. Estimate θ_l from the data set D_{imp}.

5. Repeat for imputation $l = 1, \ldots, m$.

Algorithm 3.0.1 works when we know that all of y follow the GPD. However, in most cases we have data y where only the ones over a threshold u follow the GPD. Those are the data we call extreme. The rest of the data will have a different distribution. So in step 2. of Algorithm 3.0.1 we will get an inaccurate estimate $\hat{\theta}'$ and the data we will impute in step 3. will not be of the same distribution as the ones missing.

In order to avoid this and ensure we only impute the data over the threshold using the GPD we have to construct a model that classifies the data depending on covariate X. Logistic regression can be helpful in this direction.

We divide the observed data in two classes. Class 1 will be the data that are over threshold u and class 0 the data below (or equal to) this threshold. To determine u the MRL and TR plots on the observed data D_{obs} can be used, as described in Section 2.

We chose logistic regression because it incorporates the uncertainty in classifying the data. Logistic regression does not just classify the data, but it estimates the probability that each datum belongs in each class. This is useful for our algorithm, as we want it to take into account the variability in estimating the probabilities. We will show how we do this later.

Let Z_{obs} be the categorical variable of the classes in the observed data. Before performing logistic regression with X_{obs} as covariate we have to be careful. Usually, the data that follow the GPD represent only a small percentage of the data set. So, we expect Z_{obs} to be an unbalanced categorical variable. However, logistic regression is sensitive in unbalanced data. For this reason we will use random undersampling (Chawla et al., 2002).

Let k_1 be the number of observed data that are in class 1 and k_0 in class 0. Then we have that $k_0 >> k_1$. For this reason we randomly select k_1 elements from class 0 and we construct the data set $D_{under} = (Z, X)$, which has $2k_1$ rows and $p + 1$ columns, where

p is the number of covariates in X. Now our data set is perfectly balanced with 50% of the data belonging in class 1 and 50% in class 0.

We are now ready to perform logistic regression on D_{under}. Let us assume, for simplicity, that we have linear logistic regression. Let $p_i = P(z_i = 1|X_i) = P(y_i > u|X_i)$, where X_i is the vector of covariates $X_i = (X_{i,1}, \ldots, X_{i,p})$ of y_i. Then we have the model

$$\text{logit}(p_{under,i}) = \beta X^T_{under,i}, \tag{13}$$

where $\beta = (\beta_1, \ldots, \beta_p)$. Parameter β can be estimated as usual.

The next step is to use the model in (13) to predict the probability that $y_{mis,i} > u$. For this we use the estimate $\hat{\beta}$ of β and the model

$$\text{logit}(p_{mis,i}) = \hat{\beta} X_{mis,i}. \tag{14}$$

Now we sample from the Bernoulli distribution $z_{mis,i} \sim Bernoulli(p_{mis,i})$ for the missing data. The data that are such that $z_{mis,i} = 1$ are the ones on which we will impute values using the GPD as in Algorithm 3.0.1.

The above algorithm can be summarised below.

Algorithm 3.0.2.

1. Use MRL and TR plots on the observed data to find threshold u over which data follow the GPD.
2. Define variable Z such that $z_i = 1$ for $y_i > u$ and z_i for $y_i \leq u$.
3. Randomly undersample D_{obs} creating $D_{under} = (Y_{under}, Z_{under}, X_{under})$, where 50% of data have $z_{obs,i} = 1$ and 50% $z_{obs,i} = 0$.
4. Perform logistic regression to estimate β where $\text{logit}(p_{under,i}) = \beta X^T_{under,i}$.
5. Estimate the probabilities $p_{mis,i}$ for the missing data from $\text{logit}(p_{mis,i}) = \hat{\beta} X^T_{mis,i}$.

6. For the missing data sample $z_{mis,i} \sim Bernoulli(p_{mis,i})$. Define $D'_{mis} = (y_{mis}, X_{mis})$, where $z_{mis,i} = 1$ and $D'_{obs} = (y_{obs}, X_{obs})$ such that $y_{obs} > u$.

7. Apply Algorithm 3.0.1 on D'_{mis} and D'_{obs}.

The reason why we sample from the Bernoulli distribution in step 6 of Algorithm 3.0.2 instead of just assigning the data to class 1 if $p_{mis,i} > 0.5$ is because we want our algorithm to incorporate the uncertainty in predicting the probabilities $p_{mis,i}$. This makes Algorithm 3.0.2 more robust.

4 Simulation

We will now begin testing the algorithm we presented in Section 3. Specifically, we want to see how Algorithm 3.0.2 works compared to the complete case analysis (CCA) when the scale of the GPD depends on a covariate. The reason we chose to keep the shape parameter constant is that, as mentioned in Section 2, it is uncommon to consider the shape parameter changing. We will simulate a data set of $n = 10,000$ rows and two columns where we will only have missingness in the response.

The first column will correspond to observations from random variable X, which follows the normal distribution with $\mu = 0$ and $\sigma^2 = 1$, i.e. $X \sim N(0, 1)$.

The second column of our data set will correspond to our response variable Y. The first 90% ($n_1 = 9,000$) of Y will follow the normal distribution with mean $1 + X$ and variance 1, i.e. $Y_i \sim N(1 + X_i, 1)$ for $i = 1, \ldots, n_1$. We will then consider the threshold $u = \max_{i=1,\ldots,n_1}(y_i)$. The rest 10% of Y will be over u and we will have that $Y - u$ will follow the GPD with scale $\log(\sigma(X)) = 1 + X$ and shape $\xi = 0.5$, i.e. $Y_i - u \sim GPD(\exp(1 + x_i), 0.5)$ for $i = n_1 + 1, \ldots, n$.

As we described in Section 2 we want to perform logistic regression to find the probability that a data point y_i is over the threshold u. For simplicity in our logistic regression model we will rearrange observations X in ascending order so that the larger the x_i the

bigger the probability that $y_i > u$ and we have linear logistic regression.

We will consider two cases. The first is when the data are MCAR. This means that, using the notation of Section 2, $P(r_i = 0) = 0.5$. The second case is when the data are MAR. Then, $P(r_i = 0) = P(r_i = 0|x_i)$, i.e. the probability that y_i is missing only depends on x_i. Specifically, we are interested to see what happens in the case where we have more missing data in the extremes.

For this reason, we define the following missing data mechanism: $\text{logit}(P(r_i = 0)) = 0.5 + x_i$. Then we have that the higher the x_i the bigger the probability that y_i is missing. Since we have assumed that the y_i that follow the GPD correspond to higher values of x_i the above mechanism ensures we have more missing values in the extremes. We can also see that on average we have

$$\text{logit}(P(r_i = 0)) = 0.5 \Rightarrow P(r_i = 0) = \frac{1}{1 + \sqrt{e}} \approx 0.6,$$

so on average 60% of data are missing.

We can see how the probability of y_i missing changes with x_i in Figure 2 which is generated from an instance of the data.

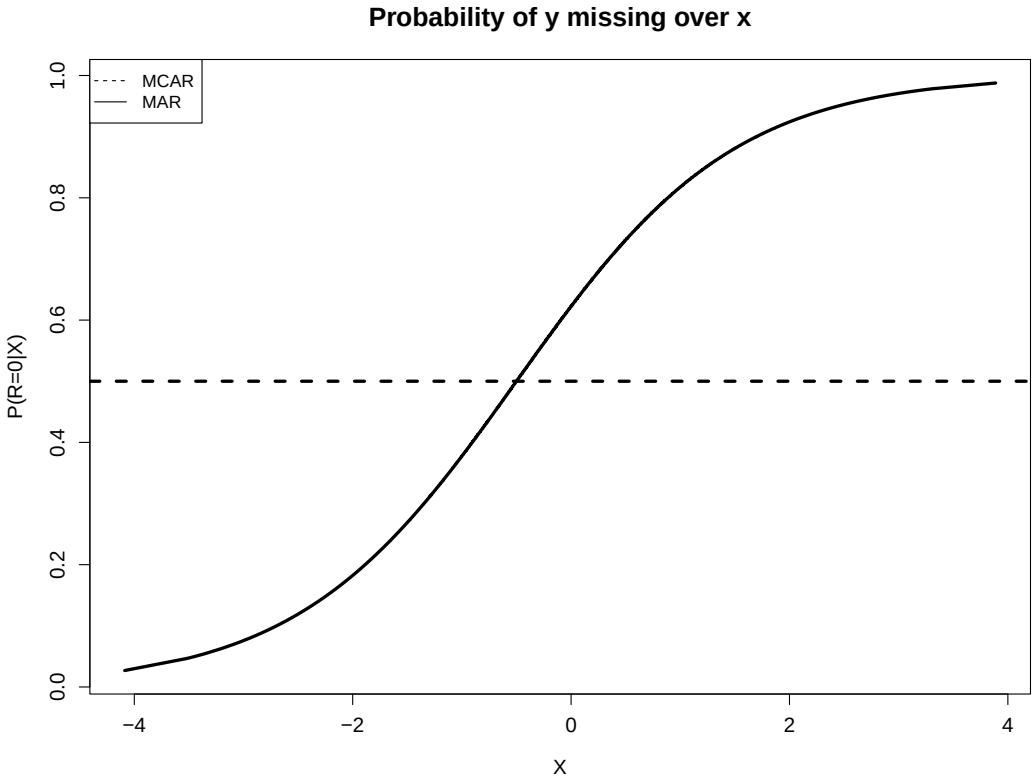

Figure 2: Plot of $P(R = 0|X)$ over X for the MCAR and the MAR case for an instance of the data.

We notice in Figure 2 that, indeed, we have more probability that data are missing in the extremes. Let us test how Algorithm 3.0.2 works compared to the CCA for our simulated data. An instance of the data can be seen in Figure 3.

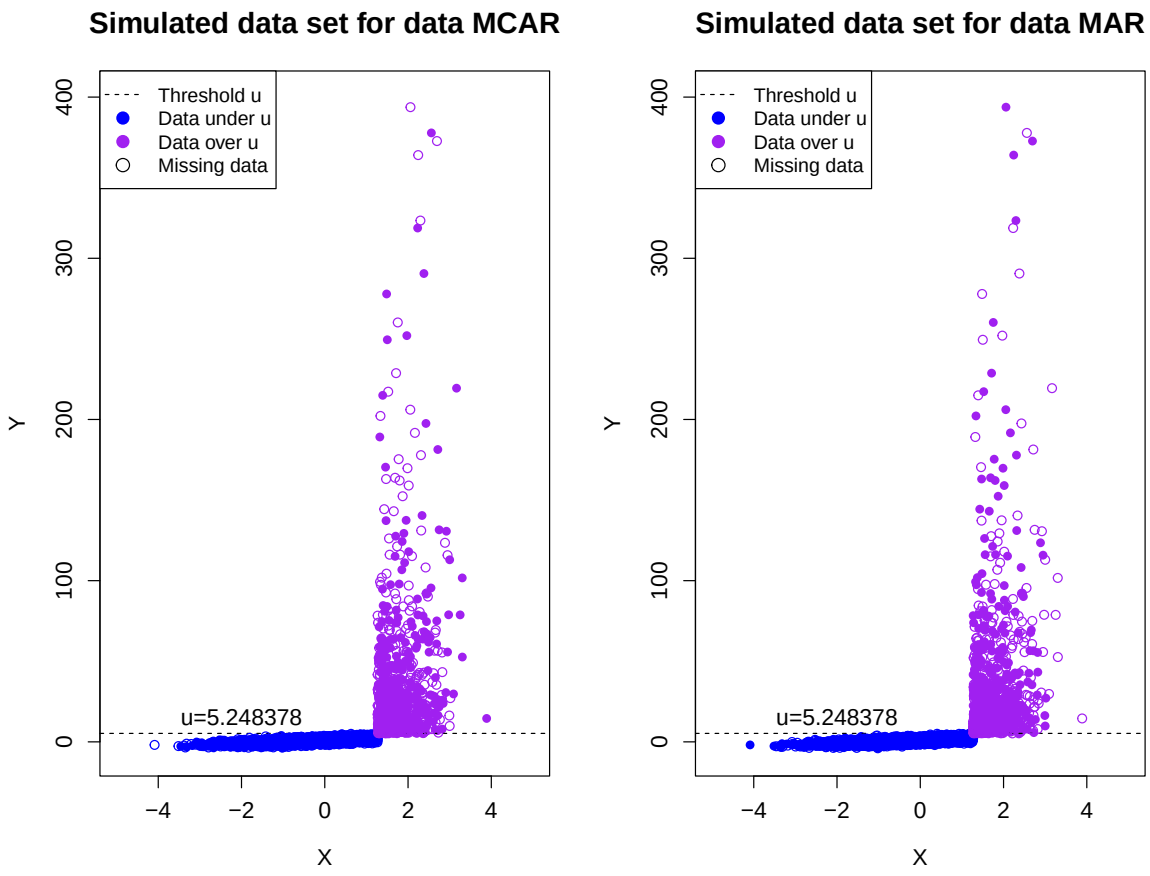

Figure 3: Generating an instance of the data for the MCAR (left) and MAR (right) case.

In Figure 3 we notice that indeed in the MAR case we have more data missing in the extremes than in the MCAR. We want to see if Algorithm 3.0.2 can perform well in both cases. For this reason we run 1000 simulations of $m = 5$ imputations with Algorithm 3.0.2 where we use the GPD distribution

$$GPD(\sigma(x) = \sigma_0 + \sigma_1 x, \xi),$$

with threshold the original threshold $u = \max_{i=1,\ldots,n_1}(y_i)$ and the regression model

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_i. \tag{15}$$

We choose (15) because we designed the data so that linear logistic regression would be ideal. In a real application we would have to study the data to find the best possible logistic regression model.

We run the simulations as described above and we get that the CCA and Algorithm 3.0.2 have a few differences. The purpose of our model is to estimate the parameters σ_0, σ_1 and xi. We will firstly analyse the situation for data MCAR.
In the CCA we have quite high PB: 8.73%, 2.9% and 10.7% for σ_0, σ_1 and ξ respectively. This is a sign of not adequate performance. However, Algorithm 3.0.2 performs worse with PB 11.3%, 8.4% and 22.1%. Of course the performance of Algorithm 3.0.2 is bad compared to the CCA according to the PB measure.
When comparing the CR (for 95% confidence intervals) for the parameters we see that the CCA has 94.1%, 95.2% and 93.3% for σ_0, σ_1 and ξ, while Algorithm 3.0.2 has 94.2%, 95% and 93.4%. Now it seems that Algorithm 3.0.2 manages to be close to the CCA in the CR and even slightly outperform it.
However, we should take into account the AW of the two methods. CCA has 1.11, 0.61 and 0.26 average widths for the confidence intervals of σ_0, σ_1 and ξ. Algorithm 3.0.2 has a lot larger widths for σ_0 and σ_1, around 3 and 2 each, but it has quite small AW for ξ, which is 0.29.

Now we consider the case where data are MAR. CCA now has even smaller PB than before in all parameters, while Algorithm 3.0.2 has PB over 15%. This is of course worrying.
The CR for both methods follow the same trend as in the MCAR case: CCA has better CR of 94%, 95% and 93% for σ_0, σ_1 and ξ, while Algorithm 3.0.2 has 93%, 94% and 93.3%.
In this case the AW of CCA are also much smaller than Algorithm 3.0.2.

From the above we can conclude that the CCA manages to estimate the parameters better than Algorithm 3.0.2. On the other hand, Algorithm 3.0.2 may not be as precise, however, because its confidence intervals are wider, it sometimes manages to slightly outperform CCA when it comes to CR. More detailed results are given in 7.

4.1 Reducing the sample size

It is of interest to see how each method generalises in different conditions. Let us firstly consider what happens when we reduce the size of the extremes. We will now consider the extremes to be 5% of the whole data set. An instance of our data is shown in Figure 4.

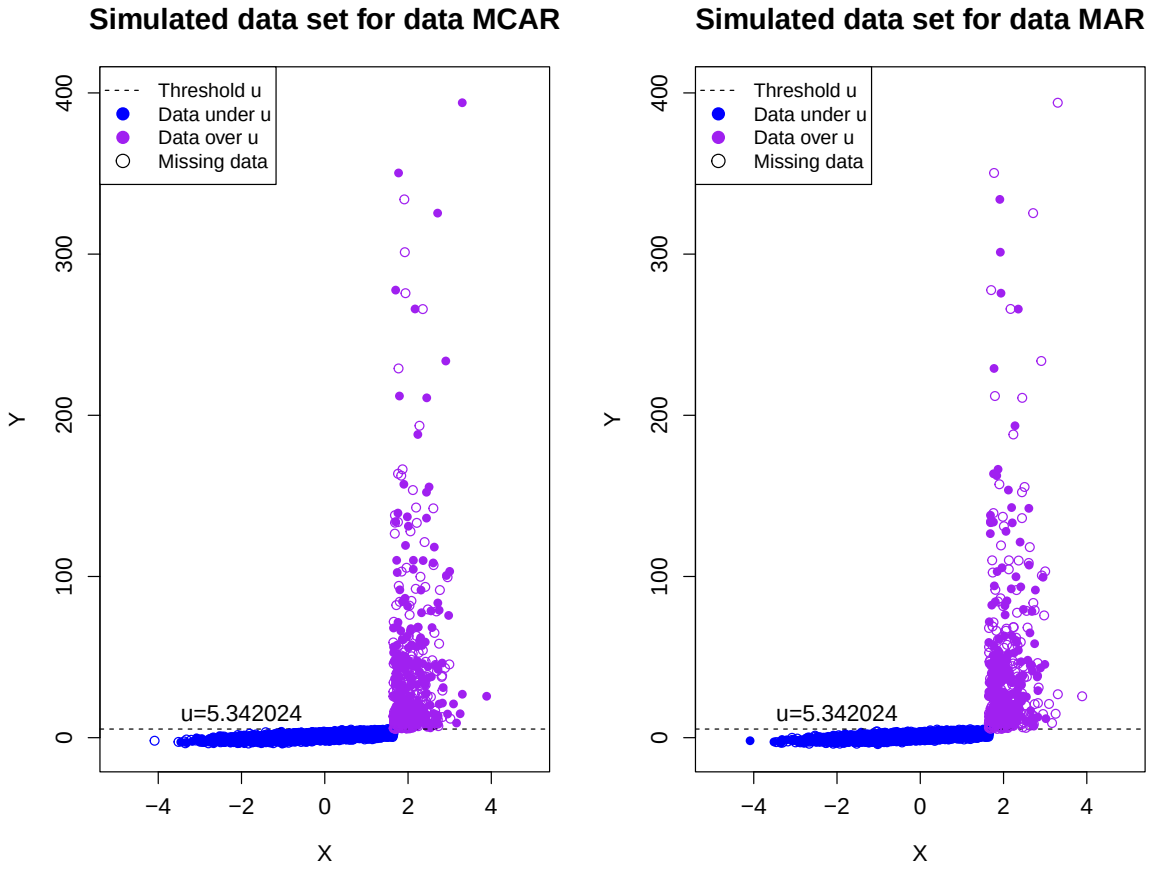

Figure 4: Generating an instance of the data for the MCAR (left) and MAR (right) case when the extreme are 5% of the data.

In Figure 4 we can clearly see that the extreme values are much less than in Figure 3. We will observer how this will impact the performance of CCA and Algorithm 3.0.2.

We start with the MCAR case. Now we see that CCA has very small PB, less that 1% for σ_0 and σ_1 and less than 4% for ξ. On the other hand Algorithm 3.0.2 has PB over

5% for all parameters, which is a concerning performance.
However, when it comes to CR Algorithm 3.0.2 manages perform almost as good as CCA. Indeed, CCA has CR 95% for σ_0 and σ_1 and 94% for ξ, while Algorithm 3.0.2 has 94% for the parameters of the scale and 92% for ξ.
The AW for both methods is interesting. CCA has a lot larger AW in σ_0 and σ_1, 2 and 0.95 respectively, after reducing the size of extremes while in the shape parameter the AW is 0.37, not much higher than before. This is the case for Algorithm 3.0.2 too. The AW for σ_0 is 4.44, for σ_1 is 2.37, while for ξ is 0.41.

Let us consider the MAR case again. We see again that CCA has very small PB (less than 2%) for σ_0 and σ_1, while for ξ the PB is 6%. This is not the case with Algorithm 3.0.2 which has PB over 30%. This is a big difference not only with CCA but also with itself in the MCAR case. This shows that Algorithm 3.0.2 can be inaccurate for small sample sizes.
The CCA and Algorithm 3.0.2 have high CR for σ_0 and σ_1 when data are MAR as well. The CR are over 94% in both, with CCA being slightly higher. However, both methods perform badly in the CR for the shape parameter. The CR of CCA for ξ is 92%, while for Algorithm 3.0.2 only 90%. These rates are bad for 95% confidence intervals.
The AW for σ_0 and σ_1 in CCA are 3.24 and 1.61 respectively, while in Algorithm 3.0.2 5.51 and 3.62. For ξ the AW are less than 0.65 in both cases.

In any case we notice that CCA performs a lot better in terms of PB and AW, while when it comes to the CR the two methods are close. It seems, though, that reducing the size of the extremes had a bigger impact on Algorithm 3.0.2. A reason for this is that Algorithm 3.0.2 fits many models (a logistic regression and $m = 5$ imputations of the GPD) and as a result it needs more data.

4.2 Changing the threshold u

So far we have seen how Algorithm 3.0.2 performs compared to the CCA when data are MCAR and MAR, and when we reduce the number of extremes in the data set. We saw

that the CCA generally performs better. However, in all the above we assumed that we knew the exact threshold u above which the data follow the GPD.

In Section 2 we mentioned that in real applications the threshold is chosen using the MRL and TR plots. As those methods are heuristic, it is almost impossible to be able to determine the exact threshold. So we would like Algorithm 3.0.2 to be able to generalise well when the threshold we use for the GPD is not the actual one.

Let u_1 be the actual threshold and u_2 the threshold we use to fit the GPD. We are interested to see what happens in the case where $u_2 < u_1$, because then we will have that the GPD data will be contaminated from some normally distributed data.

We have chosen $u_1 = \max_{i=1,\dots,n_1}(y_i)$. Let u_2 be the top 99.5 quantile of y_i, $i = 1, \dots, n_1$. This means that during our analysis there will be 45 normally distributed elements to which we will fit the GPD. For the instance of our data we presented above this can be seen in Figure 5.

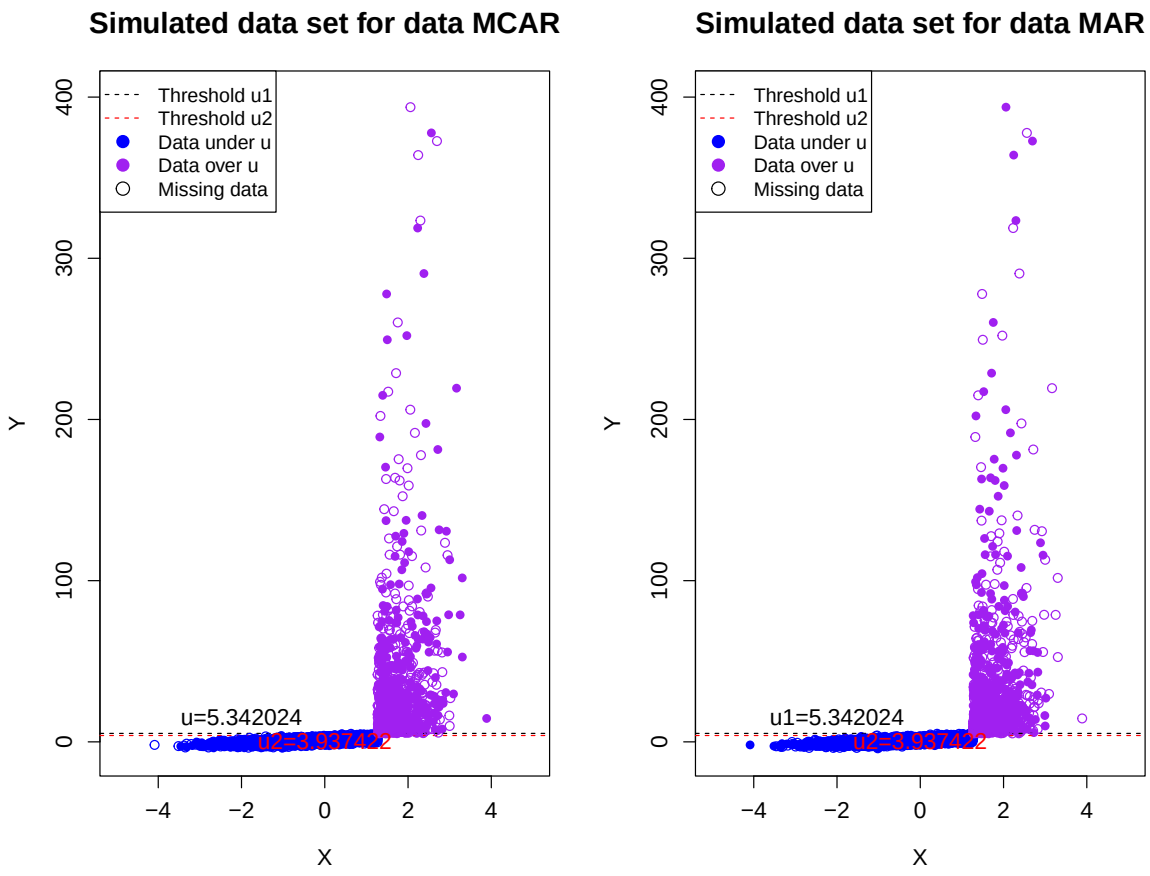

Figure 5: Generating an instance of the data for the MCAR (left) and MAR (right) with two thresholds u_1 and u_2.

In Figure 5 we observe that u_1 and u_2 are very close to each other. We now want to see how this difference will impact the performance of both the CCA and Algorithm 3.0.2.

We simulate the data, firstly, for the MCAR case. We now get that the PB in the CCA are 45.5%, 31.57% and 73.72% for σ_0, σ_1 and ξ. Algorithm 3.0.2 gives PB 43%, 27.2% and 99%. This means that even though both methods work badly, Algorithm 3.0.2 is more precise when estimating σ_0 and σ_1 than CCA.
When we check the CR we see that Algorithm 3.0.2 performs even better than the CCA. Indeed the CR are 77%, 65% and 77.2% for σ_0, σ_1 and ξ respectively, while the CR in the CCA are 60%, 46.4% and 75%. This is again bad performance from both algorithms

but we see that Algorithm 3.0.2 works a lot better.
The AW in both cases are small (less than 1.5), with Algorithm 3.0.2 having wider confidence intervals.

In the MAR case the results are similar. CCA has higher PB than Algorithm 3.0.2 in all parameters apart from ξ where Algorithm 3.0.2 has very high PB. However, the CR for σ_0 and σ_1 are very small, less than 10%, in the CCA case while in Algorithm 3.0.2 they are around 45%. In parameter ξ both methods perform similarly at 86%. In the end, Algorithm 3.0.2 has wider confidence intervals (bigger AW).

The above results are interesting. Even though when we fitted the GPD with the actual threshold u_1 CCA was working better than Algorithm 3.0.2, when we chose a slightly lower threshold Algorithm 3.0.2 managed to generalise better.

This is important and shows that Algorithm 3.0.2 is robust. The reason for this is that in practical applications where we will almost definitely choose a wrong threshold, we need a way to deal with missing data that is robust. This is the biggest advantage of Algorithm 3.0.2.

5 Discussion

Section 4 gave us interesting results. The threshold u that we choose to fit the GPD model plays an important role in the performance of our analysis. If we take as u the actual threshold above which the data follow the GPD then ignoring the missingness in the response can be the optimal choice. We saw that Algorithm 3.0.2 had trouble working with smaller samples while the CCA did not.
However, in a practical setting we cannot be sure that the threshold we choose is the real one. In this case, we saw in Section 4 that Algorithm 3.0.2 managed to generalise better and gave us better results.
This highlights the usefulness of Algorithm 3.0.2, because if we use it in a real-life

application we can expect to gain more accurate results, because even a small difference in the threshold will not impact its performance too much.

Note that the simulations were based in the simple case where we only have missingness n the response. However, most of the times we have missing data in both the response and the covariates. The reason why we chose to concentrate on this simple case is that it is the building block for the more complex case of missing data in the covariates (Van Buuren, 2018).

A version of Algorithm 3.0.2 that works with missing data in the covariates could be part of further research in the topic in order to establish whether the idea in Algorithm 3.0.2 can be used in more complex missing data situations.

6 Endmatter

For the results in Section 4 of this paper we used programming language R (R Core Team, 2022) and the packages "ismev" (functions written by Janet E. Heffernan with R port and documentation provided by Alec G. Stephenson., 2018) and "evd" (Stephenson, 2002).
The code is available here under the CC-BY-4.0 license. Please feel free to access the code and try it with your own data.

7 Supplementary Materials

7.1 Simulation Results

Here we provide the results of our simulations in the form of tables for a more detailed presentation. We will not provide any analysis here, since we did this in Section 4.

Simulation with 1000 extreme data and exact threshold u The tables from our first simulation can be found below.

- For MCAR data and CCA

Parameter	RB	PB	CR	AW
β_0	-0.0873	8.7285	0.941	1.1051
β_1	0.0283	2.8276	0.952	0.6058
ξ	0.0534	10.6758	0.933	0.2635

- For MCAR and Algorithm 3.0.2

Parameter	RB	PB	CR	AW
β_0	0.1127	11.2718	0.942	2.975
β_1	-0.0843	8.4303	0.95	2.0241
ξ	0.2032	40.6466	0.9343	0.2974

- For MAR and CCA

Parameter	RB	PB	CR	AW
β_0	0.0001	0.01	0.9429	1.5675
β_1	0.0008	0.08	0.95	0.9188
ξ	-0.0079	1.5789	0.9329	0.3253

- For MAR and Algorithm 3.0.2

Parameter	RB	PB	CR	AW
β_0	0.2272	22.7229	0.9325	3.4338
β_1	-0.1702	17.0173	0.9409	2.3104
ξ	0.2169	43.3716	0.93	0.3849

Simulation with 500 extreme data and exact threshold u The tables from our second simulation can be found below.

- For MCAR and CCA

Parameter	RB	PB	CR	AW
β_0	-0.0002	0.0186	0.9538	2.0017
β_1	0.0052	0.5243	0.9468	0.9481
ξ	-0.0177	3.5483	0.9357	0.3701

- For MCAR and Algorithm 3.0.2

Parameter	RB	PB	CR	AW
β_0	0.1319	13.194	0.9391	4.4371
β_1	-0.0719	7.187	0.9371	2.3701
ξ	0.0409	8.1819	0.9215	0.4079

- For MAR and CCA

Parameter	RB	PB	CR	AW
β_0	-0.0129	1.2928	0.9496	3.2398
β_1	0.0161	1.6137	0.947	1.6109
ξ	-0.0308	6.1662	0.921	0.5104

- For MAR and Algorithm 3.0.2

Parameter	RB	PB	CR	AW
β_0	0.5573	55.7272	0.9433	5.507
β_1	-0.3165	31.6502	0.9378	3.6164
ξ	0.1877	37.5455	0.8976	0.624

Simulation with 1000 extreme data and threshold $u_2 < u_1$ The tables from our third simulation can be found below.

- For MCAR and CCA

Parameter	RB	PB	CR	AW
β_0	-0.4551	45.5062	0.6036	1.0562
β_1	0.3157	31.569	0.4635	0.5849
ξ	0.3686	73.7209	0.7472	0.2353

- For MCAR and Algorithm 3.0.2

Parameter	RB	PB	CR	AW
β_0	-0.44	42.99	0.7689	1.706
β_1	0.2772	27.7199	0.6455	0.8268
ξ	0.4956	99.1099	0.7719	0.2767

- For MAR and CCA

Parameter	RB	PB	CR	AW
β_0	-1.2201	122.0138	0.1041	1.4217
β_1	0.7722	77.2237	0.0641	0.8496
ξ	0.536	107.2038	0.8637	0.2976

- For MAR and Algorithm 3.0.2

Parameter	RB	PB	CR	AW
β_0	-1.1634	116.3396	0.4561	2.6404
β_1	0.6719	67.1942	0.4242	2.1166
ξ	5.0001	1000.0265	0.8683	0.3735

References

Adesina, O. S., Adeleke, I., and Oladeji, T. F. (2016). Using extreme value theory to model insurance risk of nigeria's motor industrial class of business. *The Journal of Risk Management and Insurance*, 20(1):40–51.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag London Limited.

Coles, S. and Casson, E. (1998). Extreme value modelling of hurricane wind speeds. *Structural Safety*, 20(3):283–296. ID: 271417.

Davison, A. C. and Smith, R. L. (1990). Models for exceedances over high thresholds. *Journal of the Royal Statistical Society.Series B (Methodological)*, 52(3):393–442. 28.

Demirtas, H., Freels, S. A., and Yucel, R. M. (2008). Plausibility of multivariate normality assumption when multiply imputing non-gaussian continuous outcomes: a simulation assessment. *Journal of Statistical Computation and Simulation*, 78(1):69–84.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26.

Fernandez, V. (2003). Extreme value theory and value at risk. *Revista de Análisis Económico*, 18(1).

functions written by Janet E. Heffernan with R port, O. S. and documentation provided by Alec G. Stephenson., R. (2018). *ismev: An Introduction to Statistical Modeling of Extreme Values*. R package version 1.42.

Heitjan, D. F. and Little, R. J. (1991). Multiple imputation for the fatal accident reporting system. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 40(1):13–29.

Holmes, J. D. and Moriarty, W. W. (1999). Application of the generalized pareto distribution to extreme value analysis in wind engineering. *Journal of Wind Engineering and Industrial Aerodynamics*, 83(1):1–10. ID: 271422.

Jong, R. D., Buuren, S. V., and Spiess, M. (2016). Multiple imputation of predictor

variables using generalized additive models. *Communications in Statistics-Simulation and Computation*, 45(3):968–985.

Jong, R. D. and Spiess, M. (2014). *Robust multiple imputation*, pages 397–411. Improving Survey Methods. Routledge.

Leadbetter, M. R. (1983). Extremes and local dependence in stationary sequences. *Zeit. Wahrscheinl.-theorie*, 65:291–306.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rigby, R. and Stasinopoulos, D. (2005). Generalized additive models for location, scale and shape.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley Sons Inc, New York, NY.

Rubin, D. B. (1996). Multiple imputation after 18 years. *Journal of the American statistical Association*, 91(434):473–489.

Rubin, D. B. (2004). The design of a general and flexible system for handling nonresponse in sample surveys. *The American Statistician*, 58(4):298–302.

Smith, R. L. (1985). Maximum likelihood estimation in a class of nonregular cases. *Biometrika*, 72(1):67–90.

Stephenson, A. G. (2002). evd: Extreme value distributions. *R News*, 2(2):31–32.

Thomas, M., Lemaitre, M., Wilson, M. L., Viboud, C., Yordanov, Y., Wackernagel, H., and Carrat, F. (2016). Applications of extreme value theory in public health. *PloS one*, 11(7):e0159312.

Van Buuren, S. (2018). *Flexible Imputation of Missing Data*. CRC press.