



# Visualization of Incomplete Data Using the MissiG Visualization

Muhammad Imran Bin Abdul Rashid (220349255)

May 2025

BSc. Computer Science

Supervisor - Dr Sara Johansson Fernstad

Word Count: 13517

## **Declaration**

I declare that this dissertation represents my own work except where otherwise stated.”

## **Acknowledgements**

I would first like to thank my project supervisor, Sara Johansson Fernstad for allowing me to participate in this project. She has helped me with the organization of the project and has always given helpful advice. I would like to thank my family and friends for all the support that they have given me throughout this project. To them I dedicate this dissertation.

# Abstract

MissiG is a novel visualisation technique designed to represent key missingness patterns in incomplete data, which are Amount Missing (AM), Joint Missingness (JM), and Conditional Missingness (CM). It was developed in response to the limited availability of visualisation methods capable of effectively communicating these patterns. Prior evaluations have shown that MissiG performs well compared to traditional methods such as Parallel Coordinates and Heatmap. However, like many multivariate visualisation tools, its performance tends to decline as the dimensionality of the dataset increases.

Therefore, this dissertation attempts to investigate the performance of MissiG across datasets with varying numbers of variables, focusing specifically on the linear layout option. The development process is structured into three main stages: designing the layout using Figma, pre-processing datasets to introduce and explore missingness patterns, and implementing the final interactive visualisation using D3.js. The effectiveness of the system is assessed through qualitative user feedback, with an emphasis on the interpretability of missingness patterns and overall usability across datasets of different scales.

Although interactive features such as tooltips, zooming, and scaling were integrated to improve usability, the findings confirm that MissiG's interpretability is still challenged by high-dimensional data. The dissertation concludes by identifying potential areas for improving the MissiG technique and reflecting on the project's limitations that may have influenced the evaluation outcomes.

# Table of Contents

Chapter 1: Introduction .....	1
1.1 Motivation.....	1
1.2 Aim and Objectives.....	2
1.3 Dissertation Outline .....	3
Chapter 2: Background.....	4
2.1 Introduction .....	4
2.2 Missing Data.....	4
2.2.1 Introduction .....	4
2.2.2 Missingness Patterns .....	4
2.2.3 Visualisation Techniques for Missingness .....	5
2.2.4 Challenges in Visualising Missing Data.....	8
2.3 MissiG .....	9
2.3.1 Introduction .....	9
2.3.2 Key Components and Design Choices of MissiG .....	9
2.3.3 Available Layout Options .....	10
2.3.4 Observations and Relevance to Project.....	14
2.4 Visualization Tools.....	15
Chapter 3: Methodology .....	17
3.1 Introduction .....	17
3.2 Layout Design Using Figma .....	17
3.3 Pre-Processing Datasets .....	19
3.4 Visualization of MissiG Using D3.js Library .....	29
3.4.1 Rendering Feature Blocks with SVG Elements.....	29
3.4.2 Interactivity Features.....	31
3.4.3 Zooming, Panning, and Reset Functionality .....	32
Chapter 4: Results and Evaluation.....	33
4.1 Introduction.....	33
4.2 Results and Testing .....	33

4.3 Evaluation & Discussion .....	38
Chapter 5: Conclusion.....	44
5.1 Introduction.....	44
5.2 Fulfilment of Objectives.....	44
5.3 Limitations and Weaknesses.....	45
5.4 Personal Development .....	46
5.5 Future Work .....	46

# Chapter 1: Introduction

## 1.1 Motivation

Data visualisation plays a crucial role in revealing important patterns, distributions, and anomalies, and this applies equally to incomplete datasets. In the context of missing data, an effective visual representation of missingness directly influences the perceived quality of the dataset as poor representation can lead to misinterpretations [1], causing analysts to make inaccurate decisions about which pre-processing methods to apply, potentially leading to biased or misleading results. This highlights the importance of visualising missing data, as it provides a clearer understanding of missingness patterns, thereby supporting informed decision-making during the pre-processing stage. Various methods exist for visualising incomplete data [2], including heatmaps and parallel coordinates, both of which offer valuable insights but come with certain limitations [3]. To improve the effectiveness of missing data visualisation, several novel and modified techniques have been developed in recent years [2], one of which is the Missing Glyph (MissiG) visualisation. As MissiG is a relatively new technique, it is worth further investigation to understand further on its effectiveness, limitations, and potential improvements.

Research on the best visualisation of incomplete data remains limited, despite its crucial role in influencing pre-processing decisions [3]. Addressing this problem requires further exploration into the evaluation, and potential improvements of novel visualisation techniques like MissiG. Additionally, while existing research has demonstrated the high effectiveness of the Missing Glyph (MissiG) visualisation compared to other methods, such as heatmaps and parallel coordinates [3], further investigation is still needed to assess its strengths and limitations across diverse datasets. A broader evaluation, incorporating incomplete datasets with varying sizes, characteristics and domains, can provide deeper insights into the reliability and adaptability of MissiG.

## 1.2 Aim and Objectives

The primary aims of this project are to explore the effectiveness of the linear layout of the MissiG technique in representing missing patterns of datasets with different sizes and missing value distributions. This will be achieved by implementing it on three datasets with different numbers of variables and missing value distributions, evaluating the performance based on the missingness patterns and identifying potential improvements that can be added to the visualisation. The findings of this project might provide useful insights into MissiG visualisation and potentially encourage people to contribute further research in this underexplored field by refining the method or developing better alternatives.

I will achieve the project's aim through the objectives presented below:

- Develop and implement the MissiG glyph using D3.js, a suitable JavaScript library for visualisation.
- Identify or generate three datasets with different sizes (number of variables) and missing value distributions, and implement MissiG visualisation on them.
- Evaluate the effectiveness of the MissiG visualisation on the three datasets through qualitative user feedback.

Furthermore, two additional objectives were originally planned for this project:

- Implementing potential improvements to the MissiG technique.
- Evaluating the effectiveness of those improvements on the three datasets through qualitative user feedback.

These objectives would have supported a comparative analysis between the original and the enhanced versions of the visualisation. However, given the limited time available and the author's lack of prior experience with JavaScript, the approach was adjusted. Instead of running a direct comparison, the project focused on refining the original design first, followed by an evaluation to assess its effectiveness.

## 1.3 Dissertation Outline

### **Introduction**

An introductory section outlining the problem domain, along with a description of the project's overall aim and objectives.

- Motivation
- Aim and Objectives
- Dissertation Outline

### **Research**

Background research related to the project.

### **Methodology**

A section that describes in detail the whole process of the project's development phase including designing, pre-processing and code implementation.

### **Results and Evaluation**

Includes the results of the final products and evaluation made through user feedback.

- Results
- Evaluation and Discussion

### **Conclusion**

- Fulfilment of Objectives
- Limitations and Weaknesses
- Future Work

# Chapter 2: Background

## 2.1 Introduction

For this project, it was important to have the necessary understanding of what missing data is, how it can be handled, and how it can be visualised effectively. Learning about these areas helped guide the development of the visualisation and gave direction to how it would be evaluated later on. This chapter begins by explaining what missing data means, the different types of patterns that exist, the techniques used to represent them and the challenges in developing a visualisation system for them. Then, it introduces the MissiG technique, a novel visualisation designed to highlight missingness patterns in multivariate datasets. The key parts of its design and why it works well are also discussed. Lastly, some of the tools used to build the visualisation, like D3.js and Figma, are explained in specific detail. All of this background helps show why certain choices were made during the project.

## 2.2 Missing Data

### 2.2.1 Introduction

Missing data is a common issue in real-world datasets and refers to the absence of a recorded value for a specific variable in an observation [2]. It is also referred to as incomplete data or missing values and typically arises when no data entry is stored where one is expected [2]. There are several reasons why data may be missing, such as equipment malfunctions, incomplete survey responses, manual entry mistakes, or problems encountered during data integration [2]. Missing values regardless of their cause can compromise the accuracy of analysis, introduce bias into results, and reduce the effectiveness of predictive models [2]. Therefore, understanding the nature and structure of missing data is a crucial step in both data preprocessing and visualisation. This section introduces key missingness patterns and explores visualisation techniques used to represent missingness in data. It also highlights the usual solutions used to handle missing data and the challenges on visualizing missing data.

### 2.2.2 Missingness Patterns

Missing data is usually classified into three main categories, which are Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR) [4]. These categories describe whether the probability of a value being missing is influenced by other information in the dataset. In the case of MCAR, the missingness occurs entirely at random and is not related to any observed or unobserved values. MAR describes situations where missingness is related only to data that is already observed, while MNAR applies when the missingness depends on the value that is itself missing. Although this classification is widely used in statistical contexts, Johansson and Westberg [3] highlight that it can be difficult to apply in early, exploratory phases of analysis, specifically when using visual methods rather than formal statistical models.

As new visualisation techniques have been developed, researchers have also proposed alternative ways to present how missing data is distributed. One such approach, introduced by Wang and Wang (2007) [2], focuses on classifying missingness patterns based on how the missing values are spread across both variables and outcome classes. Their study outlines three key patterns, illustrated in Figure 1, which are Missing At Random (MAR), Uneven Symmetric Missing (USM), and Uneven Asymmetric Missing (UAM). In the MAR pattern, missing values occur randomly throughout the dataset, without focusing on any specific variables or classes. USM describes a situation where some variables contain more missing values than others, and those variables may share a similar missingness structure. UAM refers to cases when missing values appear more often in specific variables and are also biased toward a particular class.

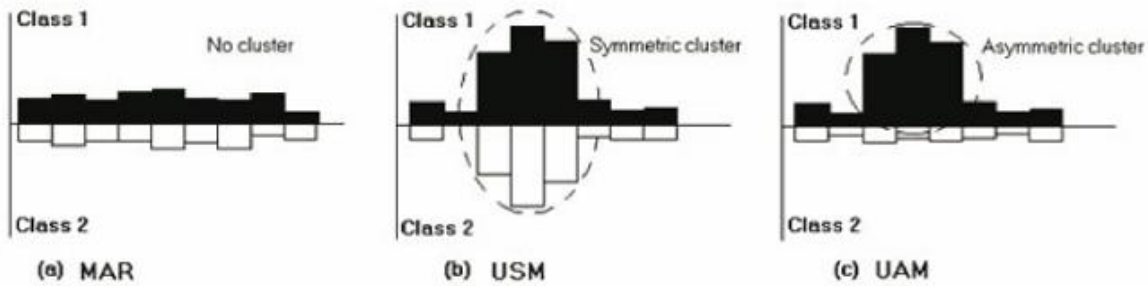


Figure 1: The Critical Patterns of missing values in Classification Data [2]

Furthermore, Johansson Fernstad [4] introduced a practical way to describe how missing data appears across variables by identifying three key patterns, which are Amount Missing (AM), Joint Missingness (JM), and Conditional Missingness (CM). AM refers to how many values are missing in a single variable, making it useful for comparing overall data quality across features. JM describes situations where two or more variables tend to have missing values in the same records, revealing shared gaps that may point to structural issues in the dataset. CM, on the other hand, focuses on the relationship between missing values in one variable and the recorded values in another. This type of pattern helps uncover why certain values might be missing, offering insights that can guide imputation or analysis strategies [3]. The MissiG technique [3], which will be used in this dissertation, was specifically designed to visualise these three patterns.

### 2.2.3 Visualisation Techniques for Missingness

In addition to established techniques such as parallel coordinates and heatmaps, which are commonly used to visualise missingness in incomplete datasets [3], several novel and modified techniques have also been developed. All of the techniques discussed in the following section were discovered and drawn from the review by Sadiq et al. [2], which provides a comprehensive overview of recent advances in missing data visualisation. MissiG, which is the technique used for this dissertation, will not be included in this summary, as it will be discussed in more detail in Section 2.3.

### 2.2.3.1 Novel Techniques

Valero-Mora et al. [5] highlight the importance of assessing data quality and recognising the impact of missing information on analysis outcomes. Noting the lack of dedicated tools for this purpose, they propose a visualisation method that allows users to explore missing data patterns interactively. Figure 2 shows an example of their plot using college admissions data. The plot presents key information such as the distribution of missing and observed values across variables. Each rectangle represents a variable with meaningful patterns. Different colour was used to distinguish between observed data (blue) and missing values (red). Each rectangle is positioned vertically based on the mean of the corresponding variable, while its width reflects the number of cases that share the same missingness pattern. To address limitations related to visual clutter, the authors suggest implementing dynamic features to highlight relevant patterns and improve clarity during exploratory analysis. This recommendation is particularly relevant to this dissertation, as datasets with many variables can also lead to cluttered views when using MissiG, especially in its linear layout [3].

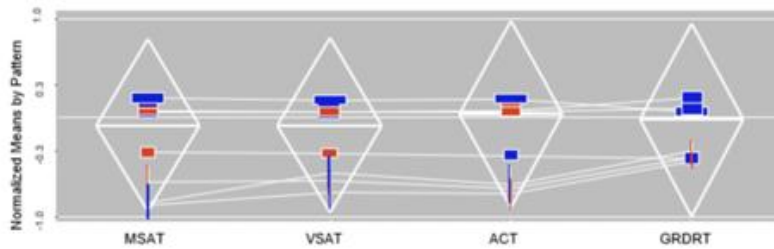


Figure 2: Plot of the missing-value patterns in college data [5]

Furthermore, Jiménez and Macías [6] propose a set of algorithms for visualising missing data in longitudinal studies using lasagna plots, a type of heat map adapted for temporal and categorical data. Their approach integrates ordering, grouping, and sampling techniques to support the identification of missingness patterns in large datasets. The ordering algorithm helps create a visual overview by arranging data based on proportions, making monotone missingness easier to detect. Grouping is based on missing data descriptors and uses clustering methods like k-means to highlight intermittent patterns. Sampling is used to ensure that meaningful proportions are retained when visualising data in matrix form. These methods aim to provide quick, interpretable visual summaries for identifying quality issues and recognising informative structures early in the analysis process. Their effectiveness is demonstrated across four real-world datasets, including the diabetes dataset shown in Figure 3.

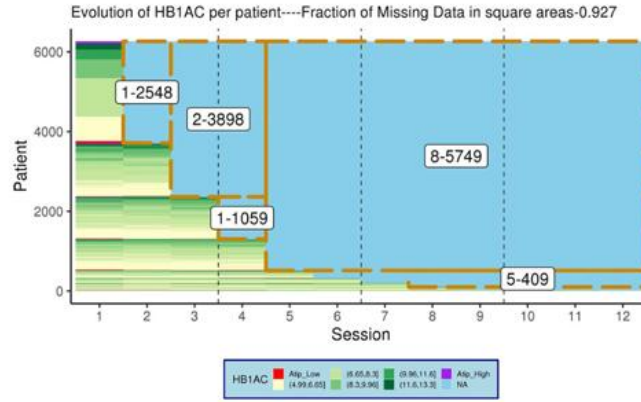
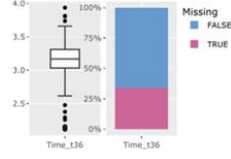


Figure 3: Diabetes data set using a lasagna plot [6]

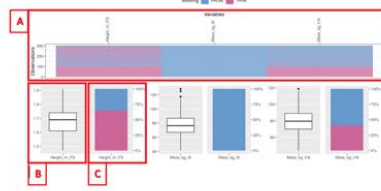
### 2.2.3.2 Modified Techniques

Alsufyani et al. [7] introduced two visualisation methods for exploring missing data patterns, which are the MissVisG glyph and the MissVis plot. As shown in Figure 4a, the MissVisG glyph represents a single variable using a rectangular design that combines a stacked bar chart (red and blue) and a box plot. This allows users to see the proportion of missing (red) and recorded (blue) values, while also identifying the distribution and outliers. It supports exploration of missingness patterns and provides useful context for choosing imputation strategies. The MissVisG glyph can be used on its own or integrated into other visualisation frameworks.

Building on this, the MissVis plot (Figure 4b) extends the idea to multiple variables. It includes a heat map that highlights missing values in red and recorded values in blue, making it easier to identify patterns within and across variables. Beneath each variable on the heat map, a MissVisG glyph is placed, offering additional detail about distribution and outliers at a glance. This combination of visual elements supports both high-level overviews and more focused comparisons, helping users identify structured missingness and assess data quality more effectively. Additionally, the MissVis plot [7] also shares similarities with the MissiG technique [3] in that it also uses a glyph-based visualisation approach and can be used to enhance other visual methods by adding clarity to the representation of missing data.



(a) The MissVisG



(b) MissVis plot: A) The heat map. B) The box plot. C) The stacked bar chart.

Figure 3: The MissVisG and MissVis plot. [7]

## 2.2.4 Challenges in Visualising Missing Data

According to Alsufyani et al. [2], several key challenges continue to shape the development of visualisation techniques for missing data. One major concern is the limited generalisability of evaluation studies, which are often based on small dataset sizes, controlled settings, or synthetic datasets. As a result, many approaches have yet to be tested in realistic, diverse environments. There is also a narrow focus on specific data types, as most existing techniques are designed for numerical and categorical data, leaving other formats like networks or mixed-type datasets underexplored. The authors also point out that while synthetic data allows for control and precision, it may fail to capture the complexity and variable relationships present in real-world data, reducing the relevance of the findings.

Designing high-quality visualisations that remain clear and uncluttered, especially when working with large datasets, remains a persistent technical challenge. Although some approaches introduce interactivity and focus on smaller subsets to reduce visual noise, achieving a balance between completeness and readability is still difficult. Finally, the survey highlights the need for better integration of visualisation tools with other data workflows, such as imputation software or data profiling platforms. Interactive features can support deeper exploration and improve decision-making, but for missing data visualisation to be widely adopted, it must also align with existing tools used by analysts and data scientists.

Although participant numbers for evaluation may be limited, the datasets used in this dissertation span a range of sizes and dimensionality. One of the core objectives of this project is to apply the MissiG technique to three datasets with varying numbers of variables, allowing its effectiveness to be evaluated across different data scales. This helps explore how well the visualisation adapts to both small and large datasets. In addition, the MissiG design adapted for this work places a strong emphasis on visual clarity. By using simplified, non-overlapping glyphs and avoiding excessive visual density, the visualisation aims to stay interpretable even when dealing with

higher-dimensional data. This directly supports the goal of producing high-quality, low-clutter visualisations of complex missingness patterns. Several of these challenges will be explored further during the development and evaluation phase.

## 2.3 MissiG

### 2.3.1 Introduction

MissiG is a glyph-based visualisation technique introduced by Johansson and Westberg in 2021 [3] to overcome limitations of traditional visualisation methods in representing complex missingness patterns in data. This section provides an overview of the MissiG design, explaining the missingness patterns, the function of each visual component in the MissiG and how different missingness patterns are represented. It also outlines the available layout options, discusses the technique's strengths and highlights its relevance to the goals of this project.

### 2.3.2 Key Components and Design Choices of MissiG

MissiG was specifically designed to represent different types of missingness which are Amount Missing (AM), Joint Missingness (JM), and Conditional Missingness (CM)[3]. The Amount Missing can be defined as the proportion of missing values within a single variable. This pattern can be compared between each variable to help identify where the lack of data may affect the reliability of any analysis [3]. Joint Missingness can be defined as the pattern where two or more variables contain missing values in the same records. Joint Missingness pattern emerges from scenarios where participants decided to skip a few related questions in a survey [3]. Conditional Missingness is a pattern of missingness that influenced by the recorded values in another variable. This pattern can reveal potential reasons behind the missing values and are useful to inform decision on the suitable imputation strategies [3].

The MissiG technique uses a glyph-based approach to visualise patterns of missingness in incomplete data [3]. As shown in Figure 5, each variable is represented by a glyph, a self-contained visual unit that encodes key information about the feature and its missingness patterns. On the left side of each glyph, a grey vertical histogram displays the distribution of recorded values for that feature. Categorical features can be visually distinguished from numerical ones by the presence of gaps between histogram bins. On the right side of the glyph, Amount Missing (AM) is represented by a vertical blue bar, where the height corresponds to the proportion of missing values in that variable.

When a glyph is selected, its border is highlighted in red, and additional visual elements appear to indicate Joint Missingness (JM) and Conditional Missingness (CM). JM is shown using red bar charts in other blocks, indicating shared instances of missingness with the selected variable. CM is visualised through a red mirrored histogram on the right side of unselected variables, showing the distribution of their recorded values in rows where the selected variable is missing. In Figure

5, the CM pattern can be visibly identified as the missing values in Feature C correspond to low recorded values in variable A and high recorded values in variable B [3].

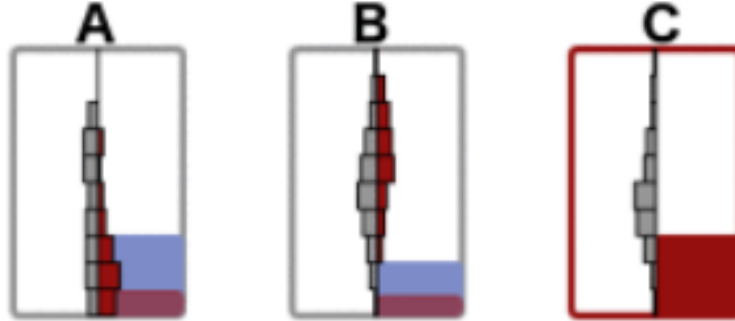


Figure 5: The basic structure of MissiG for three or four variables. Adapted from Johansson and Westberg (2021) [3]

To represent information effectively, MissiG uses three primary visual channels, which are colour, height, and shape. Colour is used to distinguish between missing values, recorded values, and values related to a selected variable. Height represents magnitudes such as the amount missing for each variable or joint missingness between each variable while shape encodes value distributions. These choices align with principles such as Typedness [8], Semantic Relevance [9], and Orderability [8], which emphasise matching visual properties with data semantics and maintaining clarity in visual hierarchies [3].

To ensure consistency and perceptual uniformity, all magnitude-based elements, such as the blue bar (AM) and red bar (JM), are scaled relative to the full height of the glyph. This design choice allows for easier comparison between glyphs. Simplicity and visual separability are also prioritised by avoiding overlapping visual channels and using basic, easily distinguishable components.

MissiG also accounts for saliency and user attention through its use of red to highlight elements associated with a selected feature. This design aligns with principles such as Attention Balance and Focus and Context, ensuring that interactive selections are visually emphasised while non-selected elements remain in the background. The use of strong contrast and intuitive mappings contributes to a visualisation that is both effective and learnable [3].

### 2.3.3 Available Layout Options

MissiG technique offers two options for its layout, which are linear and radial [3]. The linear layout (Figure 6) provides the simplest representation for the glyphs. This layout makes it easier to compare patterns of Amount Missingness (AM) and Joint Missingness (JM) across features as the glyphs are arranged in a single row. By comparing the height of the blue and red bar charts across the glyphs (Figure 6), several information regarding the AM and JM patterns can be visibly identified. For example, variables with the highest number of AM are x3 and x5, while the lowest

are variables  $x_2$  and  $x_6$ . The only variable that has no AM in it and does not share JM with the  $x_5$  is  $x_1$ . The variables with the highest JM with  $x_5$  are  $x_3$  and  $x_4$ , while the lowest are  $x_2$ .

This layout also provides an alternative representation for the JM patterns. The JM patterns are shown by the red arcs connecting the selected variable,  $x_5$  with the other variables that have shared missingness with it. The magnitude of the JM is represented by the thickness of the red arcs (Figure 6), which further clarifies the JM patterns that were shown previously by the red bar charts. CM patterns can also be identified from the figure. Based on the red histograms, this indicates that the high recorded values in variables  $x_1$ ,  $x_2$  and  $x_4$  correspond to the missing values in  $x_5$ . Missing values in  $x_5$  correspond to low recorded values in  $x_3$  as well. These observations show that there exist conditional relationships between missing values in  $x_5$  with the pattern shown from the red histograms in  $x_1$ ,  $x_2$ ,  $x_3$  and  $x_4$ .

Because of the simplicity of the layout, the MissiG glyph can also be used to enhance existing visualisation techniques such as Parallel Coordinates (PC) and Heatmaps [3]. Both of these traditional methods already offer some form of missing value representation. In the PC plot, missing values are displayed below the axis and highlighted in red when a variable is selected. In the Heatmap, missing cells are shown in red, while recorded values are shown in varying shades of grey.

In the enhanced implementation described in the original study as shown in Figure 7 and Figure 8, MissiG is interactively linked with these techniques. When a variable is selected, the corresponding MissiG glyph is updated as usual. In the PC, records with missing values in the selected variable are highlighted in red, making joint missingness patterns more noticeable. For example, higher joint missingness between  $x_5$  and both  $x_3$  and  $x_4$  becomes easier to identify through the MissiG-enhanced PC, compared to using PC alone.

While PC is known to effectively convey conditional missingness, MissiG serves as a valuable addition by visually confirming those patterns, especially in datasets with a large number of missing values. Similarly, when combined with Heatmaps, MissiG supports clearer interpretation of complex missingness structures. Although Heatmaps are generally effective for identifying AM and JM, their performance can diminish when dealing with dense datasets. In such cases, MissiG provides clearer visibility of conditional relationships. For instance, the conditional relationship between missing values in  $x_5$  and high recorded values in  $x_1$  and  $x_4$ . This relationship is harder to detect directly from the Heatmap.

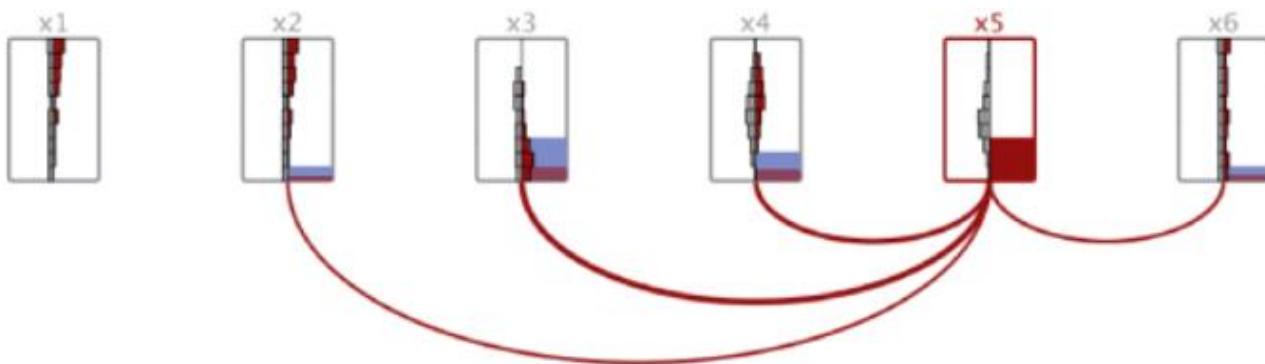


Figure 6: MissiG with linear layout [3]

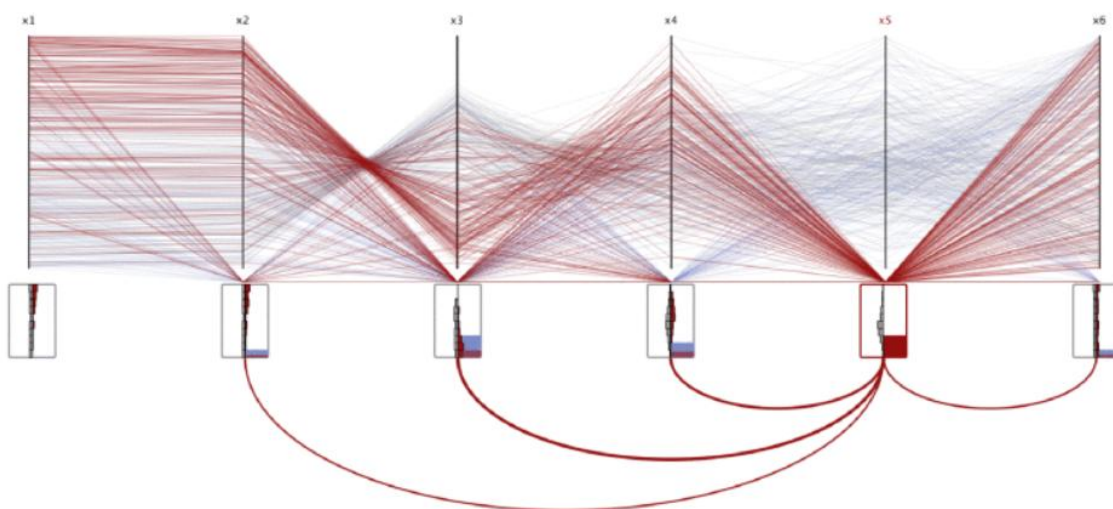


Figure 7: Parallel Coordinates enhanced with MissiG (linear layout) [3]

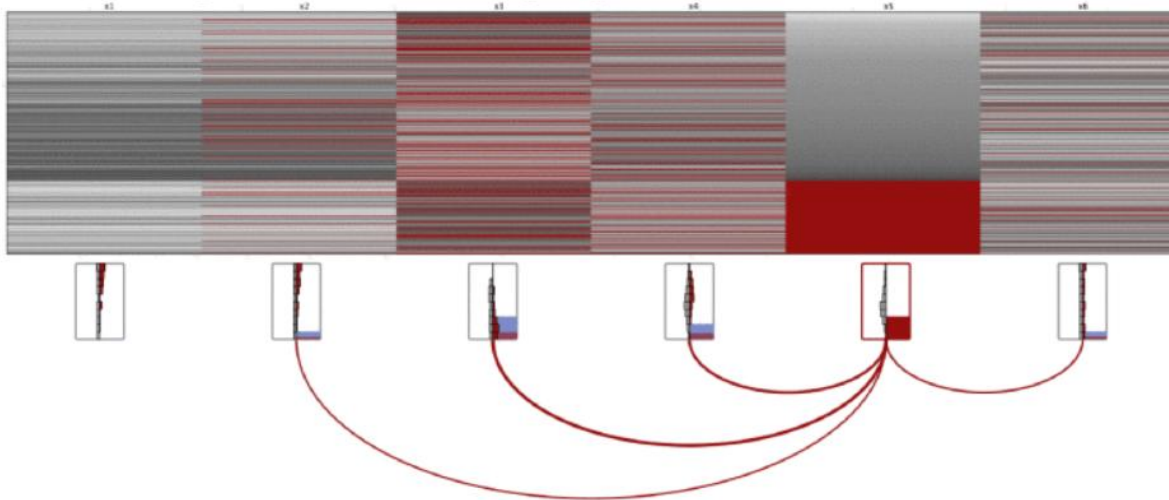


Figure 8: Heatmaps enhanced with MissiG (linear layout) [3]

In contrast to the linear arrangement, the radial layout positions the interactively selected variable at the centre of a circular view, with all other variables arranged around the circumference (Figure 9) [3]. This design supports focused analysis by allowing users to explore the relationship between a single variable of interest and all other variables simultaneously. The Joint Missingness (JM) between the selected variable and surrounding variables is visualised using red bands, where the band width reflects the magnitude of joint missingness, similar to the red arcs used in the linear layout.

One advantage of the radial layout is that it maintains a consistent distance between the central glyph and all others, reducing perceptual distortion and making comparisons between JM magnitudes more intuitive. For example, in Figure 9, x3 and x5 show clearly identifiable high joint missingness, with additional JM observed between x3 and x4. The Conditional Missingness (CM) patterns are less prominent in this layout as the red histograms closely mirror the shapes of the grey histograms, indicating no strong conditional relationships for missing values in x3.

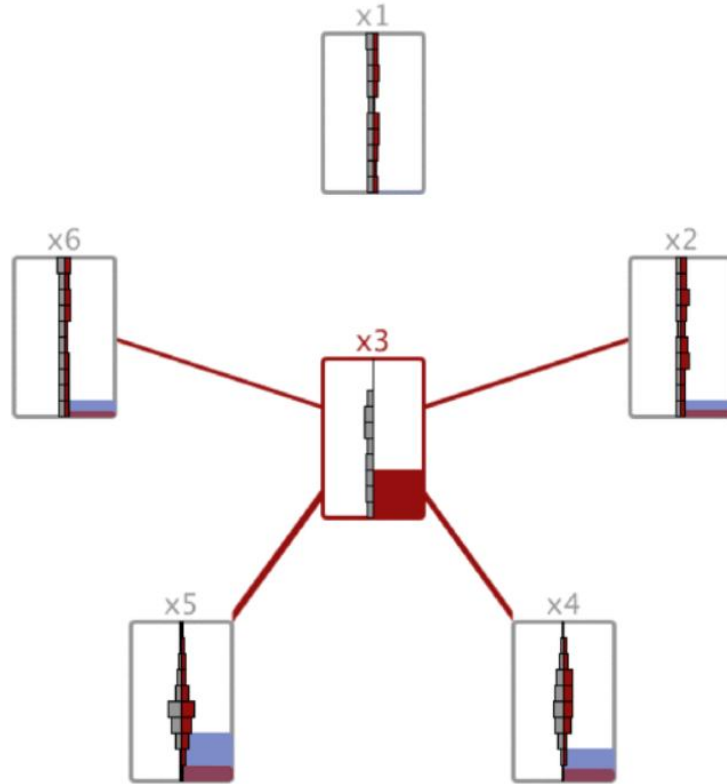


Figure 9: MissiG with radial layout [3]

### 2.3.4 Observations and Relevance to Project

MissiG has been shown to outperform other common visualisation techniques, such as parallel coordinates and heatmaps, particularly in representing complex missing data patterns [3]. In their evaluation, Johansson and Westberg (2021) [3] conducted a user study that compared MissiG to these techniques in terms of clarity, speed, and accuracy when identifying different types of missingness. The results indicated that MissiG enabled users to identify Amount Missing (AM) and Joint Missingness (JM) more effectively, especially in scenarios involving multiple variables. Although Parallel Coordinates were found to be more suitable for identifying Conditional Missingness (CM), the addition of MissiG helped confirm and clarify these patterns, particularly when data density increased [3]. However, as with other multivariate visualisation methods, MissiG's effectiveness can decline when the number of variables increases significantly. A higher number of variables leads to more glyphs and visual elements on screen, which may reduce readability and interpretation. The number of variables that can be displayed effectively depends on several factors, including screen size, the distribution of missing values, and the chosen layout. For example, datasets in which only a few variables contain missing values tend to produce much cleaner, more interpretable visualisations than datasets where missingness is widespread. To

manage this challenge, Johansson and Westberg (2021) [3] suggested using a simplified glyph design that incorporates details on demand, allowing users to focus on relevant elements while reducing visual clutter.

For this project, the linear layout was selected due to its simplicity and ease of interpretation. This decision also reflects the author’s limited experience with JavaScript and the practical advantages of working with a more straightforward layout. The linear layout allows glyphs to be arranged in a single row, making it easier to compare AM and JM patterns across features. However, it is worth noting that the radial layout is the better option in representing the JM pattern through the red arcs. This is due to the consistent distance from the central variable, which prevents from the perceptual bias that can be caused by the arc length in linear layout. Furthermore, visual clutter is expected to be a key challenge in this project, particularly given the use of multivariate datasets with missing values for the visualisation. The possible solution for the issue might be, as mentioned previously, which is the addition of details on demand [3]. This key challenge influences the addition of tooltips during the code implementation process described in section 3.4.2. Nonetheless, the linear layout will be the layout option when implementing the MissiG technique.

## 2.4 Visualization Tools

This project makes use of two visualisation tools, Figma and D3.js, with each supporting a different stage of the development process. Figma, a web-based design platform widely recognised for its collaborative features and comprehensive design capabilities [10], offers tools such as vector editing, flexible grid systems, and access to a vast library of design components [11]. By leveraging these features, developers can more effectively explore layout options, address potential design constraints early in the process, and ensure that the final implementation is both functional and user-friendly [10]. This is relevant to this project, as the usability and interpretability of the MissiG visualisation, especially across datasets with varying dimensionality, are closely tied to how effectively missingness patterns can be identified and understood [3]. Figure 10 shows that Figma offers basic vector shapes like rectangle and line, which can be added, modified and arranged as needed to create the mock-up version of MissiG with linear layout [3] during the designing stage of the project’s development.

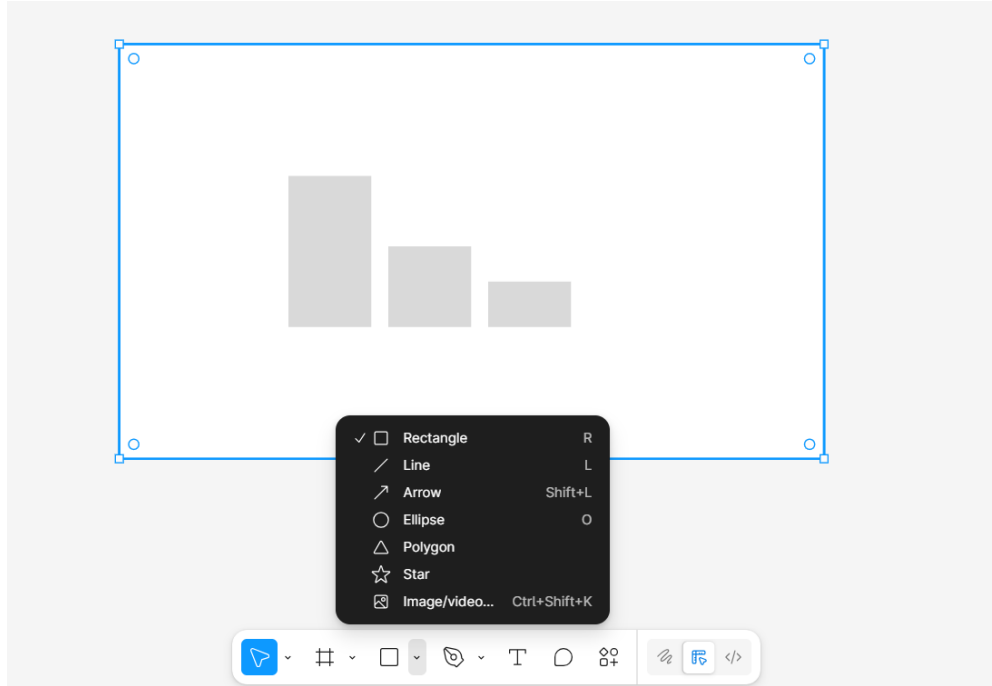


Figure 10: Shapes Drawn using Figma [10]

To implement the MissiG visualisation, this project uses D3.js (Data-Driven Documents), a JavaScript library designed for creating highly interactive, web-based visualisations. Unlike many traditional visualisation frameworks, D3 adopts a representation-transparent approach that works directly with the browser’s native Document Object Model (DOM), allowing data to be bound directly to elements on the page [12]. D3.js offers control over layout, styling, and interactivity, which are essential for encoding missingness patterns such as Amount Missing (AM), Joint Missingness (JM), and Conditional Missingness (CM). Since the MissiG design incorporates interactive components like red arcs (for JM) and red histograms (for CM) that appear dynamically when a variable is selected [3], D3’s transformation model allows these elements to be rendered efficiently without redrawing the entire visualisation. Additionally, built-in modules such as `d3.select()` and `d3.transition()` support smooth interactivity and dynamic updates, while enabling features like zooming and panning [12]. These capabilities make D3.js particularly well-suited for a glyph-based visualisation like MissiG, which demands both low-level control over SVG elements and responsive behaviour based on user input. While other visualisation libraries such as Plotly and Chart.js offer simpler interfaces for generating standard chart types, they were not selected for this project due to their limited flexibility in supporting customised layouts and advanced interactive features. These libraries are well-suited for visualising data through predefined formats like bar charts, line graphs, or scatter plots, but they lack the low-level control needed to build glyph-based visualisations that rely on precise manipulation of SVG elements [13][14].

# Chapter 3: Methodology

## 3.1 Introduction

The methodology for this project is structured around three main stages, which are designing, pre-processing, and visualising. These stages reflect the practical steps involved in developing and deploying the MissiG visualisation system. The sections that follow outline the process of designing the layout using Figma, preparing and modifying the datasets for use, and implementing the final interactive visualisation using D3.js library. Each stage includes detailed explanations, relevant code references, and supporting figures to illustrate how missingness patterns were visualised across three datasets with different levels of dimensionality.

## 3.2 Layout Design Using Figma

As the design phase did not involve any coding, Figma was used to explore and determine a suitable layout for the MissiG visualisation, as illustrated in Figure 11. The design is based on the original design for the linear layout for MissiG [3]. A linear layout was selected due to its simplicity, which aligned well with the author's limited experience with JavaScript, the primary language that will be mostly used during the implementation.

The summary of the design includes:

- Title of the dataset.
- A rectangular block (glyph) for each feature.
- A label for each feature.
- Depending on whether the feature block is selected or not, each feature block will contain a grey histogram (recorded values), a red histogram (to show CM), a blue bar chart (AM) and a red bar chart (JM).
- The thickness of the border and colour of the rectangular block are used to differentiate the selected feature (thick and red) from the unselected features (thin and black)
- Linked red arcs between the selected feature with the unselected features that have joint missingness with it.
- Zooming and Panning
- Descriptive tooltips for key elements like red arcs and blue chart (when hovered over).
- The scaling reduction for all elements when the number of features or variables increases

A modification was made to the original MissiG design [3], specifically concerning the positioning of the red histogram. In the original implementation, selecting a feature (variable) block would trigger a red histogram to appear on the right side of the layout, mirroring the existing grey histogram. The grey histogram represents the distribution of recorded values for the selected feature, while the red histogram shows the distribution of the subsets of recorded values that contain missing values of the selected feature. This mirroring helped distinguish between two types of recorded values, which are general recorded values (grey), and recorded values conditioned on missingness in another feature (red).

Although this design proved effective in prior evaluations [3], the red histogram was repositioned in this project to avoid overlapping with the bar charts located on the right side of the visual layout. During early design considerations, it was observed that when missingness and joint missingness values were very low, the corresponding red bars would appear very small and visually subtle. If the red histogram were positioned in front of these bars, especially when the red histogram contained high values for its lower distribution, it would obscure important visual cues. Additionally, since both the red histogram and the joint missingness bars use the same colour, this overlap could lead to the users' confusion when attempting to distinguish between the elements.

During this phase of development, it became clear that the available screen space for visualisation would be a significant constraint. With the linear layout, and assuming the element sizes shown in Figure 11, the visualisation cannot accommodate more than a limited number of feature blocks. This poses a challenge, as the project involves datasets with varying dimensionality. For instance, when visualising datasets with more than seven variables, some blocks may not be fully visible on screen, limiting usability and clarity.

Several solutions were considered to address this issue. One option was to explore an alternative layout in which glyphs are arranged to fully utilise available space. As shown in Figure 12, this design can support up to 21 feature blocks by adjusting the scale of each element. While effective for fitting high-dimensional datasets on a single screen, this layout comes with trade-offs. Specifically, it removes one of the key features of the linear layout, the red arcs that represent Joint Missingness (JM). Without this element, users lose an alternative visual cue for interpreting joint missingness patterns. Additionally, the stacked row arrangement limits the ability to compare bar heights across features, and the overall density of visual elements may overwhelm users when datasets contain many missingness patterns. An alternative solution involved adding a horizontal scroll mechanism. While technically simple, this approach raises usability concerns. Excessive scrolling disrupts the flow of interpretation and may hinder users from identifying patterns quickly, ultimately reducing the effectiveness of the MissiG technique in exploratory analysis.

The best and most practical solution was to apply scaling reduction. By proportionally reducing the size of all visual elements (except for the dataset's title), the visualisation can adapt to different dataset sizes while maintaining a consistent structure. To support usability at smaller scales, zooming and panning functionality was also added. This allows users to navigate large visualisations without losing interpretability. Additionally, the visual clutter problem is also a major concern for the MissiG technique, especially in high-dimensional datasets with many missingness patterns. Therefore, tooltip functionality was incorporated (see Figure 11), offering an alternative method for accessing detailed information when visual clutter increases.

Nevertheless, the layout was developed with a strong focus on usability and interpretability, particularly in representing AM, JM, and CM patterns. With the design of the MissiG linear layout finalised, the next section outlines the pre-processing steps applied to the datasets used in this project.

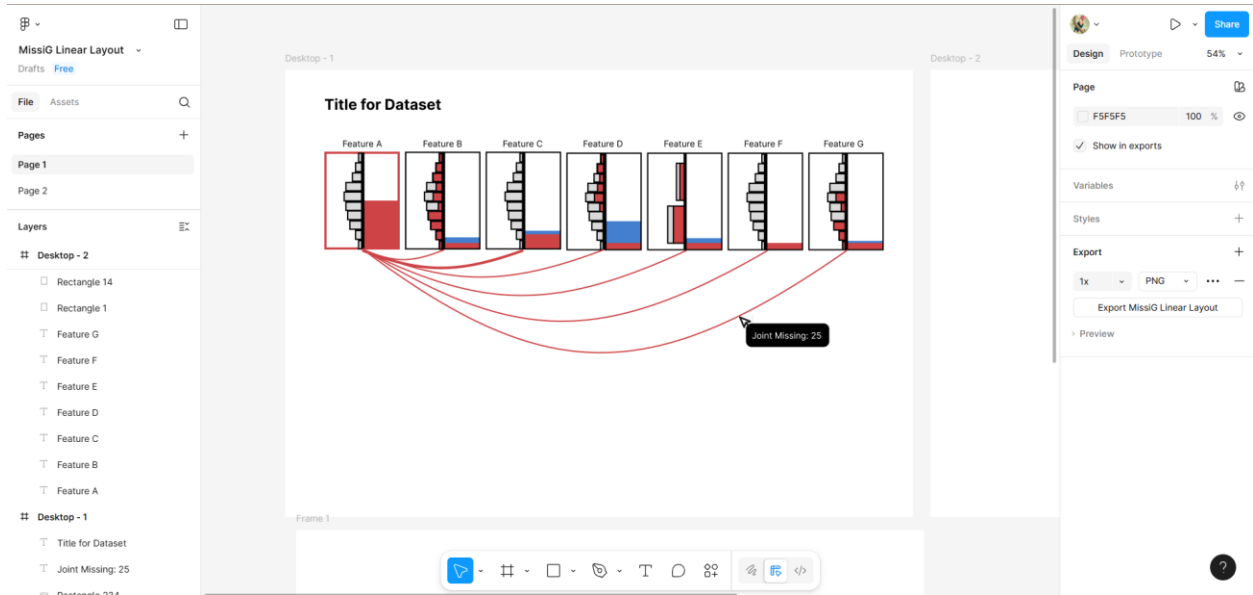


Figure 11: MissiG Linear Layout designed using Figma

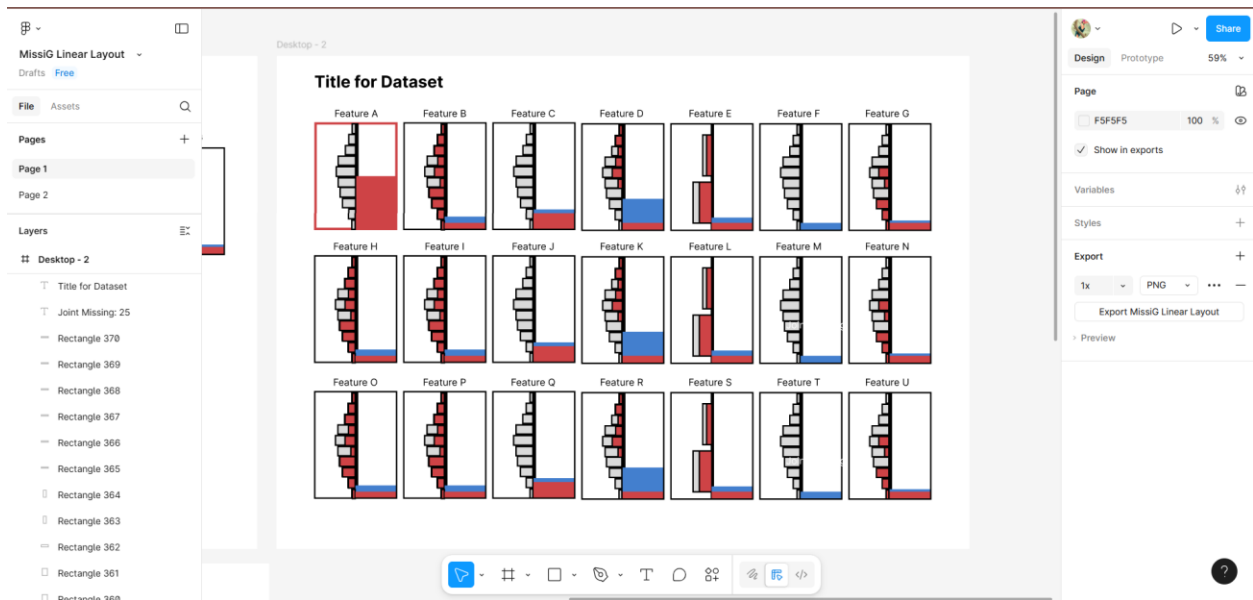


Figure 12: Layout that utilise all available space

### 3.3 Pre-Processing Datasets

This project uses three datasets with varying numbers of variables to assess the performance of the MissiG visualisation across different levels of dimensionality. The total number of records (rows) in each dataset is not a limiting factor, as the MissiG design has been shown to maintain its effectiveness on datasets with a high number of records [3]. Instead, the primary focus of this project is on the number of variables (columns or dimensions), which directly impacts the complexity and visual readability of the MissiG output. Therefore, each dataset is defined as follows:

- The small dataset contains a few variables in the range of 5 to 7.
- The medium dataset contains a number of variables in the range of 11 to 14.
- The large dataset contains many variables in the range of 18 to 24.

As publicly available datasets with meaningful missing values, particularly those that reveal Conditional Missingness (CM), are limited, the search for them was conducted through Kaggle [15], a widely used platform offering diverse datasets across multiple domains. From this platform, two datasets were selected based on specific criteria to ensure suitability for the project's visualisation. First, each dataset needed to have a usability rating of at least 8.00, indicating a sufficient level of structure and completeness for analysis. Second, the dataset had to contain at least one medium or strong correlation between two variables, which is essential for simulating CM patterns. These criteria helped ensure that the selected datasets would be both reliable and analytically meaningful for exploring missingness patterns using MissiG.

To support the dataset selection process, Google Colab [16] was used as the primary platform for exploratory analysis. It provided a convenient environment for both coding and documentation, enabling the author to perform initial data checks and compute correlation values using Python libraries such as Pandas. For example, the `corr()` method was used to identify relationships between variables, while `pd.get_dummies()` helped convert categorical data where necessary. These tools made it possible to assess whether each dataset met the project's selection criteria, particularly in identifying variable pairs with moderate to strong correlations. The findings from this stage were documented directly within the Google Colab notebook and played a key role in selecting datasets best suited to support the visualisation of Conditional Missingness (CM) in the MissiG system. As shown in Figure 13, a summary of the correlation values for the Student Performance dataset was generated within Google Colab [16]. Additional information, such as the number of missing values, data types, and unique values, was also collected through the same platform.

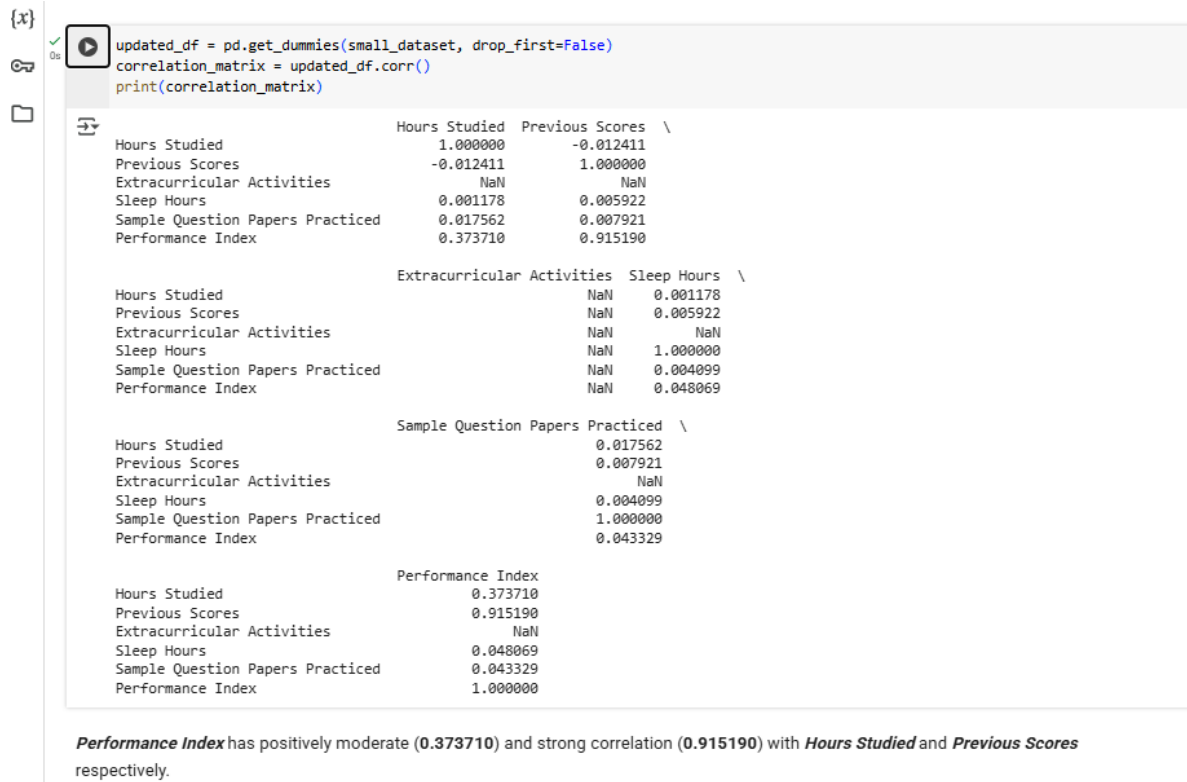


Figure 13: Dataset information obtained using Figma

The first dataset, Student Performance [17], is a complete synthetic dataset that provides insight into factors influencing students' academic outcomes, such as hours studied and previous scores. There is a medium correlation ( $\sim 0.37$ ) between Performance Index and Hours Studied, and a strong correlation ( $\sim 0.91$ ) between Performance Index and Previous Scores, as shown in Figure 13. This dataset was selected to represent the small-scale dataset in the visualisation, as it includes only six variables, which are Hours Studied, Previous Scores, Extracurricular Activities, Sleep Hours, Sample Question Papers Practiced, and Performance Index. In addition to its manageable size, the dataset was chosen for its clear and relatable context, which is expected to support user interpretation, particularly in identifying Conditional Missingness (CM) patterns during evaluation. All six variables in the dataset will be used for the visualisation.

For the medium-sized dataset, the Sleep Health and Lifestyle Dataset [18] will be used. This dataset covers 13 variables related to sleep, health and lifestyle of the respondents. These include variables such as Occupation, Quality of Sleep and Physical Activity Level. The dataset has missing values only for the variable Sleep Disorder. It was chosen as there are many medium and strong correlations among the variables, specifically between Sleep Duration, Quality of Sleep and Stress Level. For example, there is a strong correlation between Sleep Duration and variables such as Quality of Sleep ( $\sim 0.88$ ) and Stress Level ( $\sim -0.81$ ). These will help in introducing the CM pattern in the dataset, which can be visibly identified through the MissiG technique. Both the small and medium datasets have a usability rating of 10.00. This is reflected by the detailed information given about the dataset, such as an explanation of what each variable represents and clear variable

names. For this dataset, as the Person ID will be excluded, only 12 variables will be used for the visualisation.

The Sleep Health and Lifestyle Dataset [18] was selected as the medium-sized dataset for this study. It comprises 13 variables that capture a range of sleep, health, and lifestyle factors among respondents. These include indicators such as Occupation, Quality of Sleep, and Physical Activity Level. Notably, missing values occur only within the Sleep Disorder variable. Despite that, this dataset was chosen due to the presence of several medium to strong correlations among key variables, particularly between Sleep Duration, Quality of Sleep, and Stress Level. For instance, Sleep Duration exhibits a strong positive correlation with Quality of Sleep (0.88) and a strong negative correlation with Stress Level (−0.81). These relationships are well-suited to demonstrating the MissiG technique, as the induced CM pattern can be visually identified through the resulting visualisation. For visualisation purposes, the Person ID field will be excluded, leaving 12 variables to be used in the analysis. In terms of usability, both the small and medium datasets received a rating of 10.00. This reflects the datasets’ comprehensive documentation, including clearly named variables and explanatory descriptions for each feature.

The Kamyr Digester [19] dataset was the only dataset not sourced from Kaggle and was the only one found to be incomplete, with missingness patterns AM, JM and CM. It has 23 variables and serves as the large-scale dataset used in this project. The dataset consists of variables associated with the pulp quality metric [19]. This dataset was selected because it has previously been used with the MissiG design [3], where it demonstrated better performance compared to traditional techniques such as PC and heatmaps in visualising missingness patterns, especially those involving the CM pattern [3]. Although many high-dimensional datasets are available on Kaggle, only a limited number are suitable for visualisation using MissiG. This is primarily due to factors such as a usability rating below 8.00, an excessive number of categories in categorical variables, weak or non-existent correlations among key variables, or the need for complex preprocessing steps that fall beyond the scope of this project.

For the Kamyr Digester dataset, one variable will be excluded (Observation), leaving 22 variables for visualisation purposes. A summary of the characteristics of all three datasets used in this project is provided in Table 1.

*Table 1: Summary of datasets used in this project*

Label	Datasets	Descriptions
<b>Small</b>	Student Performance (Multiple Linear Regression) [17]	Number of columns (dimensions/ features/variables): 6 Number of rows (records/ entries): 9999 Variables (with data type) included: - <ul style="list-style-type: none"> <li>• Hours Studied (int64)</li> <li>• Previous Scores (int64)</li> <li>• Extracurricular Activities (object)</li> <li>• Sleep Hours (int64)</li> <li>• Sample Question Papers Practiced (int64)</li> <li>• Performance Index (float64)</li> </ul> Initial presence of missingness (NULL): None

<b>Medium</b>	Sleep Health and Lifestyle Dataset [18]	<p>Number of columns (dimensions/ features): 12  Number of rows (records/ entries): 374  Variables (with data type) included: -</p> <ul style="list-style-type: none"> <li>• Gender (object)</li> <li>• Age (int64)</li> <li>• Occupation (object)</li> <li>• Sleep Duration (float64)</li> <li>• Quality of Sleep (int64)</li> <li>• Physical Activity Level (int64)</li> <li>• Stress Level (int64)</li> <li>• BMI Category (object)</li> <li>• Blood Pressure (object)</li> <li>• Heart Rate (int64)</li> <li>• Daily Steps (int64)</li> <li>• Sleep Disorder (object)</li> </ul> <p>Initial presence of missingness (NULL): Sleep Disorder (219)</p>
<b>Large</b>	Kamyr Digester [19]	<p>Number of columns (dimensions/ features): 22  Number of rows (records/ entries): 301  Variables (with data type) included: -</p> <ul style="list-style-type: none"> <li>• Y-Kappa (float64)</li> <li>• ChipRate (float64)</li> <li>• BF-CMratio (float64)</li> <li>• BlowFlow (float64)</li> <li>• ChipLevel4 (float64)</li> <li>• T-upperExt-2 (float64)</li> <li>• T-lowerExt-2 (float64)</li> <li>• UCZAA (float64)</li> <li>• WhiteFlow-4 (float64)</li> <li>• AAWhiteSt-4 (float64)</li> <li>• AA-Wood-4 (float64)</li> <li>• ChipMoisture-4 (float64)</li> <li>• SteamFlow-4 (float64)</li> <li>• Lower-HeatT-3 (float64)</li> <li>• Upper-HeatT-3 (float64)</li> <li>• ChipMass-4 (float64)</li> <li>• WeakLiquorF (float64)</li> <li>• BlackFlow-2 (float64)</li> <li>• WeakWashF (float64)</li> <li>• SteamHeatF-3 (float64)</li> <li>• T-Top-Chips-4 (float64)</li> <li>• SulphidityL-4 (float64)</li> </ul>

		<p>Initial presence of missingness (NULL): -</p> <ul style="list-style-type: none"> <li>• ChipRate (4)</li> <li>• BF-CMratio (14)</li> <li>• BlowFlow (13)</li> <li>• ChipLevel4 (1)</li> <li>• T-upperExt-2 (1)</li> <li>• T-lowerExt-2 (1)</li> <li>• UCZAA (24)</li> <li>• WhiteFlow-4 (1)</li> <li>• AAWhiteSt-4 (141)</li> <li>• AA-Wood-4 (1)</li> <li>• ChipMoisture-4 (1)</li> <li>• SteamFlow-4 (1)</li> <li>• Lower-HeatT-3 (1)</li> <li>• Upper-HeatT-3 (1)</li> <li>• ChipMass-4 (1)</li> <li>• WeakLiquorF (1)</li> <li>• BlackFlow-2 (1)</li> <li>• WeakWashF (1)</li> <li>• SteamHeatF-3 (1)</li> <li>• T-Top-Chips-4 (1)</li> <li>• SulphidityL-4 (141)</li> </ul>
--	--	--

In cases where datasets did not originally contain missing values [17] or have not enough missing values [18], missingness was introduced selectively based on the correlation between variables. As there are enough correlations between variable Performance Index with variables Hours Studied and Previous Scores, missing values will be injected in the small dataset. As shown in the Figure 14, the dataset was first sorted in ascending order by Performance Index, and a subset of records representing the lowest-performing students was selected. Within this group, missing values were then injected into Hours Studied and Previous Scores, with the proportion of missingness for each variable randomly chosen between 25% and 35%. This approach will display the conditional missingness pattern clearly in the visualisation, where missing values in variables Hours Studied and Previous Scores will correspond to low recorded values of the variable Performance Index. In addition to this, random missingness was applied to several other variables as shown in Figure 15, including Sleep Hours, Sample Question Papers Practiced, Extracurricular Activities, and a small portion of the Performance Index variable itself, to simulate a missing values scenario commonly observed in real-world datasets.

```

9
10 # Sorting by variable 'Performance Index' (lowest performers first)
11 df_sorted = df.sort_values(by="Performance Index")
12
13 # Range for missing fraction (between 25% and 35%)
14 min_frac = 0.25
15 max_frac = 0.35
16
17 # Generate different random fractions for column 'Hours Studied' & 'Previous Scores'
18 missing_frac_hours = np.random.uniform(min_frac, max_frac)
19 missing_frac_prev_scores = np.random.uniform(min_frac, max_frac)
20
21 # Determine how many rows to mask for each column (lowest performers only)
22 n_missing_hours = int(missing_frac_hours * len(df_sorted))
23 n_missing_prev_scores = int(missing_frac_prev_scores * len(df_sorted))
24
25 # Select indices from low performers
26 low_perf_indices_hours = df_sorted.head(n_missing_hours).index
27 low_perf_indices_prev_scores = df_sorted.head(n_missing_prev_scores).index
28
29 # Introduce missingness to selected indices
30 df.loc[low_perf_indices_hours, 'Hours Studied'] = np.nan
31 df.loc[low_perf_indices_prev_scores, 'Previous Scores'] = np.nan
32

```

Figure 14: Code snippet to introduce CM in the small dataset

```

33 # Function to introduce random missingness in other columns
34 def random_missingness(df, column, min_frac, max_frac, seed): 4 usages
35     np.random.seed(seed)
36     frac = np.random.uniform(min_frac, max_frac)
37     missing_indices = df.sample(frac=frac, random_state=seed).index
38     df.loc[missing_indices, column] = np.nan
39     return df
40
41 # Introduce random missingness to other columns
42 df = random_missingness(df, column='Sleep Hours', min_frac: 0.10, max_frac: 0.15, seed=3)
43 df = random_missingness(df, column='Sample Question Papers Practiced', min_frac: 0.07, max_frac: 0.13, seed=4)
44 df = random_missingness(df, column='Extracurricular Activities', min_frac: 0.10, max_frac: 0.15, seed=3)
45 df = random_missingness(df, column='Performance Index', min_frac: 0.03, max_frac: 0.05, seed=4)
46
47 # Save the updated dataset
48 output_path = "data/student_performance_with_mnan.csv"
49 df.to_csv(output_path, index=False)
50 print(f"Conditional Missingness(CM) introduced successfully and saved into: {output_path}")
51

```

Figure 15: Code snippet to introduce random missingness into other variables in small dataset

Furthermore, a custom script was also developed to introduce both conditional and random missingness into the medium dataset as shown in the Figure 16 and Figure 17. Conditional missingness was applied in cases where the likelihood of missing values depended on observed values in other variables (strong correlations). For example, records with high stress levels (Stress Level  $\geq 7.5$ ) were assigned missing values in Sleep Duration, Quality of Sleep, or both, reflecting the impact of psychological stress on sleep-related data availability. Similarly, low physical activity levels (Physical Activity Level  $< 50$ ), triggered missingness in Daily Steps, and specific

categories of Sleep Disorder (Insomnia and Sleep Apnea) correspond to targeted missingness in Sleep Duration and Quality of Sleep, respectively. Additionally, the CM pattern was also introduced for healthcare professionals, where nurses and doctors had missing values in sleep-related variables. This design simulates lifestyle-related data gaps that may arise due to the demanding nature of their profession. In real-world settings, individuals in these roles often struggle to consistently monitor or record their sleep activity, resulting in incomplete or unavailable data in this domain. Finally, random missingness was added to selected columns such as Age, Heart Rate, and BMI Category, ensuring the dataset contained a mixture of structured and unstructured missingness patterns suitable for visual analysis.

As the visualisation required rendering histograms to represent recorded values from other unselected variables, all datasets originally stored in CSV format were converted into JSON (JavaScript Object Notation) format, as shown in Figure 18 and Figure 19. JSON was chosen due to its structured format and the flexibility it offers when working with hierarchical data in JavaScript-based environments. Given the linear layout established in Figure 11, each feature in the dataset needed to support the rendering of multiple visual elements, including bar charts, histograms, and arcs. As the JSON format allowed each feature to be represented as an object containing all relevant information, this will significantly simplify the process of looping through features and accessing their properties during visualisation. Each feature object included the following attributes: the name of the feature, its type (categorical or numerical), the number of missing values (Amount Missing), the number of joint missingness instances with other features, and a summary of recorded values for other features in rows where the current feature was missing. Depending on the feature type, information such as the actual recorded values or their frequencies was also stored to support the visualisation of datasets.

```

... 8  ### Introduce Conditional Missingness for visualization ###
9  # if 'Stress Level' >= 7.5, introduce missing values for variable 'Sleep Duration' or 'Quality of Sleep' or both
10 high_stress = df[df['Stress Level'] >= 7.5]
11 n_stress_missing = int(np.random.uniform( low= 0.15, high= 0.2) * len(high_stress))
12 stress_indices = high_stress.sample(n=n_stress_missing, random_state=0).index
13 choice = np.random.choice(['Sleep Duration', 'Quality of Sleep', 'both'])
14 if choice == 'Sleep Duration':
15     df.loc[stress_indices, 'Sleep Duration'] = np.nan
16 elif choice == 'Quality of Sleep':
17     df.loc[stress_indices, 'Quality of Sleep'] = np.nan
18 else: # 'both'
19     df.loc[stress_indices, ['Sleep Duration', 'Quality of Sleep']] = np.nan
20
21 # if 'Physical Activity Level' < 50, introduce missing values for variable 'Daily Steps'
22 low_activity = df[df['Physical Activity Level'] < 50]
23 n_activity_missing = int(np.random.uniform( low= 0.05, high= 0.15) * len(low_activity))
24 activity_indices = low_activity.sample(n=n_activity_missing, random_state=1).index
25 df.loc[activity_indices, 'Daily Steps'] = np.nan
26
27 # if 'Sleep Disorder' == 'Insomnia', introduce missing values for variable 'Sleep Duration'
28 insomnia = df[df['Sleep Disorder'] == 'Insomnia']
29 n_insomnia_missing = int(np.random.uniform( low= 0.15, high= 0.2) * len(insomnia))
30 insomnia_indices = insomnia.sample(n=n_insomnia_missing, random_state=2).index
31 df.loc[insomnia_indices, 'Sleep Duration'] = np.nan
32
33 # if 'Sleep Disorder' == 'Sleep Apnea', introduce missing values for variable 'Quality of Sleep'
34 apnea = df[df['Sleep Disorder'] == 'Sleep Apnea']
35 n_apnea_missing = int(np.random.uniform( low= 0.15, high= 0.2) * len(apnea))
36 apnea_indices = apnea.sample(n=n_apnea_missing, random_state=3).index
37 df.loc[apnea_indices, 'Quality of Sleep'] = np.nan
38
39 # if 'Occupation' == Nurse or Doctor, introduce missing values for variable 'Sleep Duration' or 'Quality of Sleep' or both
40 nurse_doctor = df[df['Occupation'].isin(['Nurse', 'Doctor'])]
41 n_nd_missing = int(np.random.uniform( low= 0.0, high= 0.12) * len(nurse_doctor))
42 nd_indices = nurse_doctor.sample(n=n_nd_missing, random_state=4).index
43 choice = np.random.choice(['sleep_duration', 'quality_sleep', 'both'])
44 if choice == 'sleep_duration':
45     df.loc[nd_indices, 'Sleep Duration'] = np.nan
46 elif choice == 'quality_sleep':
47     df.loc[nd_indices, 'Quality of Sleep'] = np.nan
48 else: # 'both'
49     df.loc[nd_indices, ['Sleep Duration', 'Quality of Sleep']] = np.nan
50

```

Figure 16: Code Snippet to Introduce CM in the Medium Dataset

```

50
51 # Function to introduce random missingness in other columns
52 def random_missingness(df, column, min_frac, max_frac, seed):
53     np.random.seed(seed)
54     frac = np.random.uniform(min_frac, max_frac)
55     missing_indices = df.sample(frac=frac, random_state=seed).index
56     df.loc[missing_indices, column] = np.nan
57
58 # Introduce random missingness to other columns
59 random_missingness(df, column='Age', min_frac= 0.05, max_frac= 0.1, seed=10)
60 random_missingness(df, column='Physical Activity Level', min_frac= 0.0, max_frac= 0.15, seed=11)
61 random_missingness(df, column='Stress Level', min_frac= 0.0, max_frac= 0.15, seed=12)
62 random_missingness(df, column='BMI Category', min_frac= 0.0, max_frac= 0.15, seed=13)
63 random_missingness(df, column='Blood Pressure', min_frac= 0.0, max_frac= 0.15, seed=14)
64 random_missingness(df, column='Heart Rate', min_frac= 0.0, max_frac= 0.15, seed=15)
65 random_missingness(df, column='Occupation', min_frac= 0.0, max_frac= 0.05, seed=16)
66
67 # Save the updated dataset
68 output_path = "data/Sleep_health_and_lifestyle_dataset_with_mnar.csv"
69 df.to_csv(output_path, index=False)
70 print(f"Conditional Missingness(CM) introduced successfully and saved into: {output_path}")
71

```

Figure 17: Code Snippet to Introduce Random Missingness into Other Variables (Medium Dataset)

```

20
21 features_data = [] # A list to include information for each feature
22 features = df.columns.tolist() # A list that contain columns names in the dataset
23
24 ## Loop through each feature
25 for feature in features:
26     column = df[feature] # extract column data
27     missing_mask = column.isna() # set up boolean mask for missing values
28     missing_count = missing_mask.sum() # calculate total number of missing values in the current feature
29
30 # Labelling type for each feature
31 if column.dtype == 'object':
32     feature_type = "categorical"
33 elif pd.api.types.is_numeric_dtype(column):
34     # removing any missing values, ensuring only non-missing data are analyzed and calculate the number of unique values
35     unique_vals = column.dropna().unique()
36     # if few unique integer-like values, it considers as categorical type
37     if len(unique_vals) <= 5 and all(float(x).is_integer() for x in unique_vals):
38         feature_type = "categorical"
39     else:
40         feature_type = "numerical"
41 else:
42     feature_type = "categorical"
43
44 # Store recorded values for the numerical features or category counts for the categorical features
45 if feature_type == "numerical":
46     recorded_values = column.dropna().tolist()
47     categories = None
48 else:
49     recorded_values = None
50     categories = column.dropna().value_counts().to_dict()
51
52
53 # Calculate JM with every other feature
54 joint_missing = []
55 for other in features:
56     if other == feature:
57         joint_missing.append(0) # prevent JM with itself
58     else:
59         both_missing = df[feature].isna() & df[other].isna()#
60         joint_missing.append(int(both_missing.sum()))
61

```

Figure 18: Code Snippet to Convert CSV to JSON

```

61
62 # Store recorded values for other features conditioned on values for this feature are missing
63 conditioned_on_missing = {}
64 for other in features:
65     if other != feature:
66         conditioned_vals = df.loc[missing_mask, other].dropna().tolist()
67         conditioned_on_missing[other] = conditioned_vals
68
69 # A dictionary to store information for each feature
70 feature_info = {
71     "name": feature,
72     "type": feature_type,
73     "missing": int(missing_count),
74     "jointMissing": joint_missing,
75     "conditionedOnMissing": conditioned_on_missing
76 }
77
78 # Add recorded values or categories to the JSON file
79 if feature_type == "numerical":
80     feature_info["recorded"] = recorded_values
81 else:
82     feature_info["categories"] = categories
83
84 # Append feature info to the dataset list
85 features_data.append(feature_info)
86
87 # Save the JSON file to output path
88 with open(output_path, "w") as f:
89     json.dump(features_data, f, indent=2)
90
91 print(f"JSON file saved into: {output_path}")
92

```

Figure 19: Code Snippet to Convert CSV to JSON

## 3.4 Visualization of MissiG Using D3.js Library

### 3.4.1 Rendering Feature Blocks with SVG Elements

For this project, the HTML, CSS, and JavaScript code for the MissiG visualisation were combined into a single .html file. This approach simplified development and testing by allowing all components of the visualisation to be managed within a unified environment. The decision was also influenced by the author's limited experience with JavaScript-based visualisation frameworks and the primary focus of the project, which was to produce visual output rather than building a modular web application.

The layout for the MissiG visualisation was implemented using Scalable Vector Graphics (SVG), which allows control over the positioning and styling of visual elements. Each feature in the dataset is represented as a feature block, a self-contained group of SVG elements that includes labels, bar charts, and histograms. These feature blocks were arranged in a linear layout, using fixed values for width (rectWidth) and height (rectHeight) to maintain visual consistency across all features.

D3.js methods such as `d3.select()` and `d3.append()`, as shown in Figure 20, were used extensively to create and manipulate these feature blocks. For each feature, its corresponding block was generated within a parent SVG `<g>` group element, positioned based on its index and the total number of features. Additionally, the coordinates (x, y) of each block were calculated dynamically and stored in an attribute to support further interactions, such as drawing red arcs between blocks and applying scaling reduction for high-dimensional datasets.

```

missiG-small.html x
1 <html lang="en">
2 <body>
3 <script>
4 d3.json("data/student_performance_with_amr.json").then(data => {
5
6 });
7
8 svg.call(zoom);
9
10 // 3. Layout Configuration
11 const rectWidth = 200, rectHeight = 300, padding = 15;
12 const maxBlockPerRow = 22;
13
14 // Total Number of Records(rows)
15 let totalRows;
16 if (data[0].recorded) {
17   totalRows = data[0].recorded.length + data[0].missing;
18 } else if (data[0].categories) {
19   const recordedCount = Object.values(data[0].categories).reduce((a, b) => a + b, 0);
20   totalRows = recordedCount + data[0].missing;
21 } else {
22   totalRows = data[0].missing;
23 }
24
25 const n = data.length;
26 const screenWidth = 1920;
27 const blocksPerRow = Math.min(n, maxBlockPerRow);
28 const requiredWidth = blocksPerRow * (rectWidth + padding);
29 const scaleFactor = Math.min(1, screenWidth / requiredWidth);
30 container.attr("transform", `scale(${scaleFactor})`);
31
32 // 4. Creating Each Feature Block
33 data.forEach((feature, i) => {
34   const col = i % maxBlockPerRow;
35   const row = Math.floor(i / maxBlockPerRow);
36   const x = padding + col * (rectWidth + padding);
37   const y = 80 + row * (rectHeight + 200);
38   feature.layout = { x, y };
39
40   // Feature Label
41   container.append("text")
42     .attr("x", x + rectWidth / 2)
43     .attr("y", y - 10)
44     .attr("text-anchor", "middle")
45     .attr("font-size", "14px")
46     .attr("font-weight", "bold")
47     .text(feature.name || "Name of Feature");
48
49   // Feature Box Group
50   const g = container.append("g")
51     .attr("transform", `translate(${x}, ${y})`)
52     .attr("id", `feature-${i}`)
53     .attr("class", "feature-block")
54     .on("click", () => handleSelectFeatureJM(i, feature));
55
56   // Feature Outline and Divider Line
57   g.append("rect")
58     .attr("width", rectWidth)
59     .attr("height", rectHeight)
60     .attr("fill", "white")
61     .attr("stroke", "black")
62     .attr("stroke-width", 1);
63   g.append("line")
64     .attr("x1", rectWidth / 2)

```

Figure 20: Snippet of code related to the creation of feature blocks

Based on the specifications outlined in Section 3.2, each feature block contains:

- A feature label that was added using the <text> element.
- A divider line which splits the block into left and right halves.
- A blue-coloured bar chart located on the right side of the feature block, which represents the amount of missingness (AM).
- A grey histogram on the left, which represents the distribution of recorded values for the feature.

The histograms were generated differently depending on the type of feature being represented, as shown in Figure 21. For numerical features, the d3.bin() function was used to generate histogram bins from the raw recorded values. These bins were then visualised as rectangular bars using d3.rect(), with bar widths scaled proportionally to the bin counts through d3.scaleLinear().

For categorical features, frequency counts were obtained from predefined category mappings. These were rendered as histogram bins with gaps between them, using d3.scaleBand() to map

categories to vertical positions, with widths again scaled linearly to represent counts for each category.

```

110 // 3. Functions for Drawing Histogram for Numerical/Categorical Features
111 function renderNumericalHistogram(g, feature) {
112     const values = feature.recorded;
113     const [minValue, maxValue] = d3.extent(values);
114     const binCount = Math.min(10, new Set(values).size);
115     const thresholds = d3.range(minValue, maxValue, (maxValue - minValue) / binCount);
116     feature.thresholds = thresholds;
117     feature.domain = [minValue, maxValue];
118
119     const bins = d3.bin().domain([minValue, maxValue]).thresholds(thresholds)(values);
120     const barHeight = rectHeight / bins.length;
121     const maxCount = d3.max(bins, d => d.length);
122     feature.grayMaxCount = maxCount;
123
124     const scaleX = d3.scaleLinear().domain([0, maxCount]).range([0, rectWidth / 2 - 1]);
125
126     g.selectAll(".hist-bar")
127       .data(bins)
128       .enter()
129       .append("rect")
130       .attr("x", d => rectWidth / 2 - scaleX(d.length))
131       .attr("y", (d, i) => rectHeight - (i + 1) * barHeight)
132       .attr("width", d => scaleX(d.length))
133       .attr("height", barHeight)
134       .attr("fill", "#ccc")
135       .attr("stroke", "#000")
136       .attr("stroke-width", 0.5)
137       .on("mouseover", (event, d) => {
138         tooltip.style("visibility", "visible").html(
139           `Range: ${d.x0.toFixed(2)}-${d.x1.toFixed(2)} | Count: ${d.length}`);
140       })
141       .on("mouseout", (event) => tooltip.style("top", (event.clientY + 10) + "px").style("left", (event.clientX) + "px"))
142       .on("mouseover", () => tooltip.style("visibility", "hidden"));
143 }
144
145 function renderCategoricalHistogram(g, feature) {
146     const categories = Object.keys(feature.categories) || [];
147     const counts = Object.values(feature.categories) || [];
148     const yScale = d3.scaleBand().domain(categories).range([0, rectHeight]).padding(0.10);
149     const barHeight = yScale.bandwidth();
150     const maxCount = Math.max(...counts);
151     feature.grayMaxCount = maxCount;
152     const scaleX = d3.scaleLinear().domain([0, maxCount]).range([0, rectWidth / 2 - 1]);
153
154     g.selectAll(".hist-bar")
155       .data(categories.map((cat, idx) => ({ category: cat, count: counts[idx] })))
156       .enter()
157       .append("rect")
158       .attr("x", d => rectWidth / 2 - scaleX(d.count))
159       .attr("y", d => yScale(d.category))
160       .attr("width", d => scaleX(d.count))
161       .attr("height", barHeight)
162       .attr("fill", "#ccc")
163       .attr("stroke", "#000")
164       .attr("stroke-width", 0.5)
165       .on("mouseover", (event, d) => {
166         tooltip.style("visibility", "visible").html(
167           `Category: ${d.category} | Count: ${d.count}`);
168       })
169       .on("mouseout", (event) => tooltip.style("top", (event.clientY + 10) + "px").style("left", (event.clientX) + "px"))
170       .on("mouseover", () => tooltip.style("visibility", "hidden"));
171 }

```

Figure 21: Functions to render numerical and categorical histogram (grey)

### 3.4.2 Interactivity Features

Based on the original design for MissiG [3] and the design specified in Section 3.2, several interactive features were implemented using D3.js event handling functions. These interactions were designed to reveal deeper insights into missing data patterns, specifically Joint Missingness (JM) and Conditional Missingness (CM). The interaction is initiated when a user clicks on a feature block. This action triggers the `handleSelectFeatureJM()` function, which will perform several visual updates. First, the selected feature block is highlighted by increasing the thickness of the feature's border and changing both the colour of the feature's border and the amount of missingness (AM) bar to red. Then, three additional visual elements are rendered:

- Red bar charts are displayed within all other unselected feature blocks to indicate the degree of joint missingness (JM) shared with the selected feature. The height of these red bars is scaled according to the JM count.

- Red arcs are drawn between the selected feature block and each feature block with which it shares joint missingness. These arcs are curved for visual clarity, and their thickness encodes the magnitude of the joint missingness (JM).
- Red histograms are rendered in the left panel of other unselected feature blocks to represent conditional missingness (CM). These histograms show the distribution of recorded values for other unselected features that are subsets with the missing values in the selected feature.

Interactivity was further extended through the implementation of mouseover, mousemove, and mouseout events to display descriptive tooltips. When users hover over red bars, red arcs, or histogram bins, a tooltip appears near the cursor, providing contextual information, such as missing counts, value ranges, or category frequencies. This multi-layered interactivity enables users to explore complex missingness patterns that would be difficult to observe in high-dimensional datasets. It also supports visual discovery without overwhelming the user, as the information box only appears when a mouse cursor hovers over key elements. The key elements include all the visual elements that help in displaying missingness patterns (AM, JM and CM), such as red arcs and red histogram bins. When users hover over red arcs, the tooltip reveals the corresponding JM count, indicating the number of joint missing values shared between the selected and connected features. For red histogram bins, the tooltip displays the value range covered by the bin and the number of observed values within it. This targeted interactivity provides users with detail insights into both the structure and extent of missingness across different variables or features.

### 3.4.3 Zooming, Panning, and Reset Functionality

To support better navigation, zooming and panning functionality was implemented using the `d3.zoom()` method. This allowed users to freely scale, move around the visualisation space and making it easier to explore patterns without being constrained by the initial layout. The zoom behaviour was applied to a `<g>` layer (zoom-layer) inside the main SVG container. This layer wraps all visual elements and responds to user interactions such as mouse wheel movements and drag gestures. Zooming is limited to a predefined scale extent to prevent over-zooming or distortion. The transformation applied by `d3.zoom()` updates the `transform` attribute of the layer, enabling smooth scaling and repositioning without affecting the internal structure of the glyphs.

A reset mechanism was also added to improve usability. Pressing the “R” key on the user’s keyboard returns the visualisation to its original state by clearing any glyph highlights (red arcs, bars, and histograms) and resetting the zoom level to the default. This is particularly useful after a complex exploration of joint and conditional missingness patterns, allowing users to quickly return to a clean overview. All these functionalities will make the visualisation more accessible and scalable, especially when working with high-dimensional datasets.

# Chapter 4: Results and Evaluation

## 4.1 Introduction

This chapter presents the results of the visualisation, beginning with a series of tests conducted to verify that the implementation meets the specifications outlined in Section 3.2. It then provides a detailed account of the evaluation process, which was based on qualitative user feedback done on 3 participants, followed by a discussion of the key insights and findings that emerged from the evaluation.

## 4.2 Results and Testing

The MissiG visualisation using a linear layout was successfully rendered for all three datasets, as illustrated in Figures 22, 23, and 24. Each dataset was displayed with its respective title and a set of glyphs corresponding to the number of variables present in the dataset. The scale reduction mechanism implemented during development also functioned as intended. This is evidenced by the proportional reduction in the size of glyphs as the number of variables increased, ensuring that the visualisation remained legible across datasets of varying dimensionality.

### Student Performance

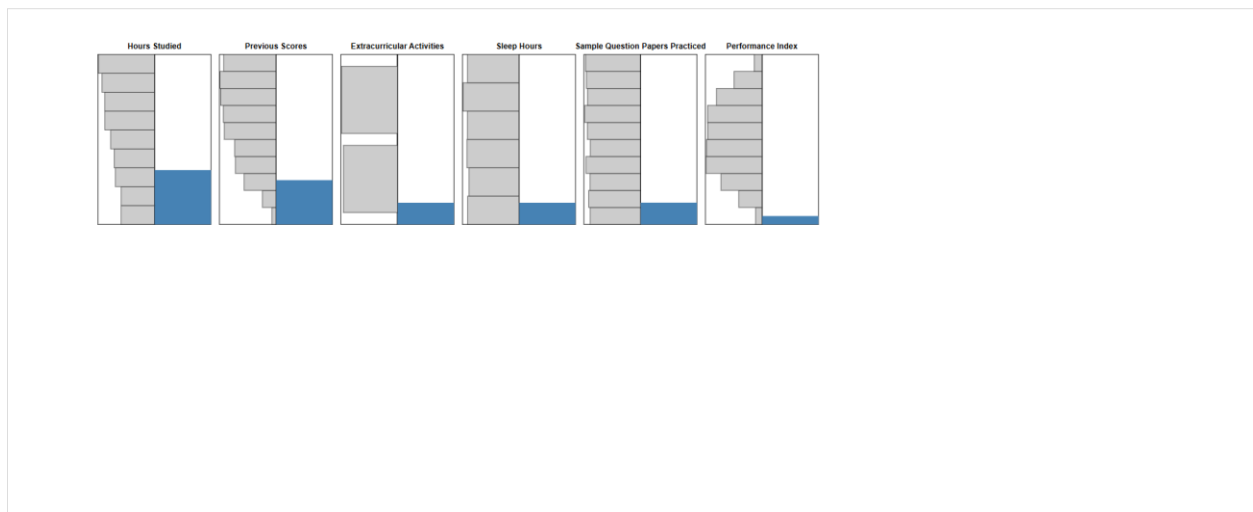


Figure 22: MissiG visualization of small dataset using linear layout

## Sleep Health and Lifestyle

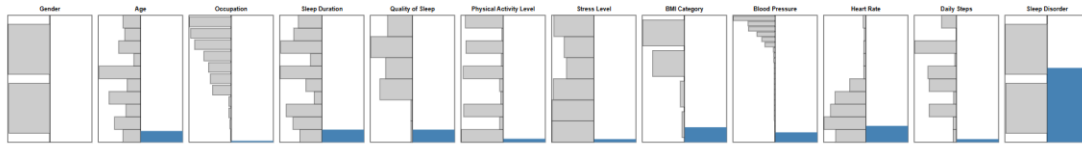


Figure 23: MissiG visualization of medium dataset using linear layout

## Kamyr Digester

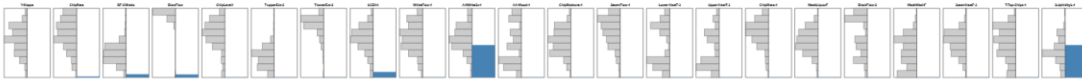


Figure 24: MissiG visualisation of large dataset using linear layout

When a specific feature block is clicked, several elements in the visualisation are updated or added dynamically (Figure 25). These include: -

- A colour change in the missingness bar of the selected feature from blue to red, indicating Amount Missing (AM).
- The addition of red bar charts in other feature blocks to represent Joint Missingness (JM).
- The addition of red arcs connecting the selected feature block to others with shared missingness (JM).
- The addition of red histograms in unselected features to visualise Conditional Missingness (CM).

Based on Figure 22, 23 and 24, the scale reduction was also applied as the number of variables visualised in datasets increases.

An issue of visual clutter was identified in the initial implementation of red arcs representing Joint Missingness (JM) between feature blocks, as shown in Figure 25. The problem arose from the overlapping of red arcs, which may hinder users' readability of JM patterns. This was caused by the use of a fixed arc height, specifically a static value of 60 pixels, which did not account for the varying horizontal distances between feature blocks in the linear layout (see Figure 26). To address this limitation, the arc height calculation was revised to dynamically scale based on the horizontal distance between the selected feature and each unselected feature. The updated code, shown in Figure 27, replaces the fixed height with a proportional value using the formula:

$$\text{Math.abs}(x2 - x1) / 3.$$

This approach will adjust the arc curvature relative to the distance between the source ( $x1$ ) and target ( $x2$ ) positions, resulting in arcs that are more evenly spaced and less prone to visual overlap. As demonstrated in Figure 28, this adjustment significantly improves the readability of the visualisation. The arcs are now distributed more clearly across the canvas, allowing users to better interpret joint missingness patterns without distraction from overlapping elements.

#### Student Performance

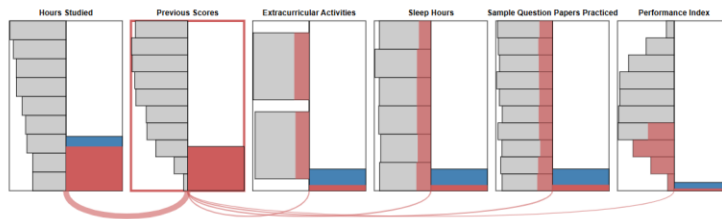


Figure 25: MissiG visualisation with updated interactivity elements

```

// Drawing curved red arcs (JM) between selected feature and unselected features
const source = selectedFeature._layout;
const target = feature._layout;

const x1 = source.x + rectWidth / 2;
const y1 = source.y + rectHeight;
const x2 = target.x + rectWidth / 2;
const y2 = target.y + rectHeight;

const path = d3.path();
path.moveTo(x1, y1);
path.bezierCurveTo(x1, y1 + 60, x2, y2 + 60, x2, y2);

```

Figure 26: Snippet of red arcs codes

```

269 // Drawing curved red arcs (JM) between selected feature and unselected features
270 const source = selectedFeature._layout;
271 const target = feature._layout;
272
273
274 const x1 = source.x + rectWidth / 2;
275 const y1 = source.y + rectHeight;
276 const x2 = target.x + rectWidth / 2;
277 const y2 = target.y + rectHeight;
278
279 const path = d3.path();
280 path.moveTo(x1, y1);
281 const arcHeight = Math.abs(x2 - x1) / 3;
282 path.bezierCurveTo(x1, y1 + arcHeight, x2, y2 + arcHeight, x2, y2);
283

```

Figure 27: Snippet of updated red arcs codes

## Student Performance

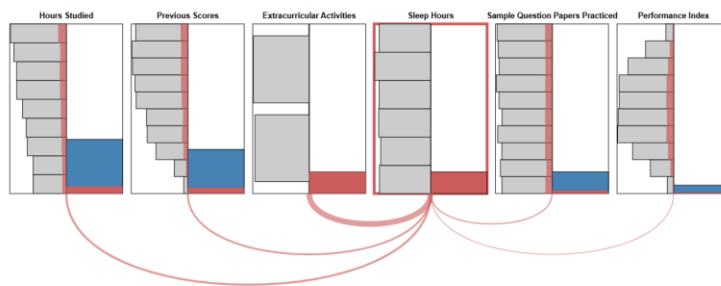
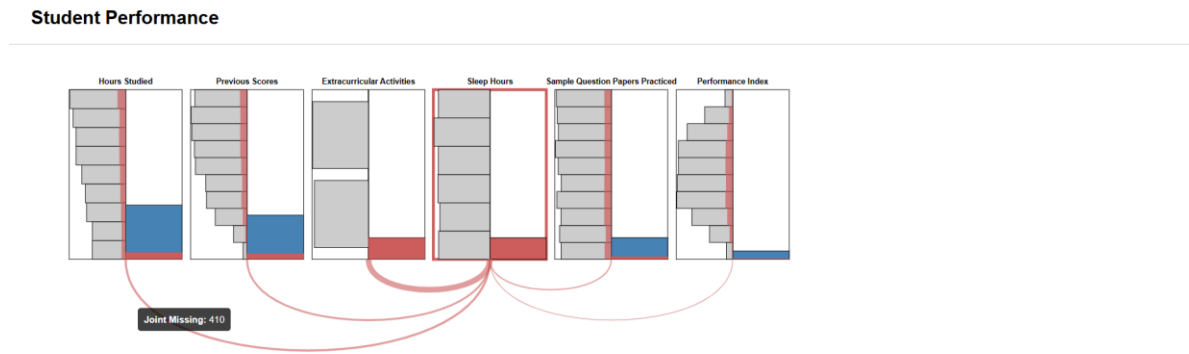


Figure 28: MissiG with updated red arcs measurement (small dataset)

Descriptive tooltips are also displayed when the cursor hovers over key visual elements, providing contextual information to support interpretation. This functionality is illustrated in Figure 29. The elements that were applied with this functionality include all the elements that support the discovery of missingness patterns.



*Figure 29: MissiG visualisation with a descriptive tooltip*

The zooming and panning functionality operates as intended (see Figure 30), which will help users to explore the datasets with ease, specifically to accommodate the scaling reduction of elements that was done on the medium and large datasets. Lastly, the reset key (r) successfully restores the visualisation to its original state, clearing all the added and updated elements.

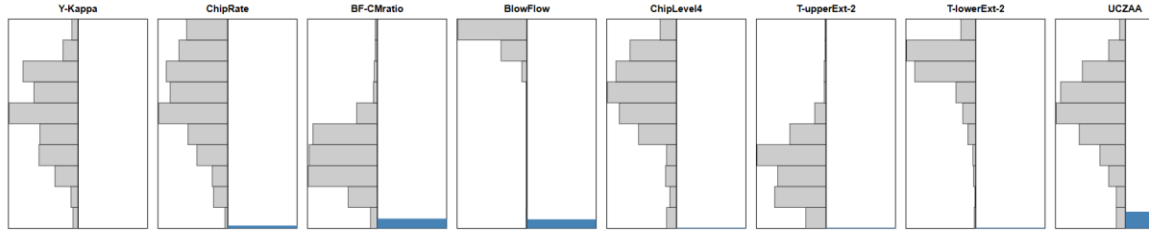


Figure 30: Zooming and panning functionalities on MissiG (large dataset)

## 4.3 Evaluation & Discussion

### 4.3.1 Potential Areas of Misinterpretation and Weaknesses

One potential limitation of the current visualisation lies in the use of colour encoding, specifically the use of red colour to represent recorded values in unselected variables. Since the same colour is also used to indicate joint missingness between the selected variable and other unselected variables, there is a risk of misinterpretation. Users that are not used to the MissiG design may incorrectly assume that the red histograms in the unselected variables represent the distribution of missing values within those variables, rather than conveying the pattern of conditional missingness relative to the selected variable. Without a proper explanation, this overlap in colour usage may lead to confusion and misreading of the visualised patterns.

Furthermore, without hovering over the element, which will reveal exactly what it represents (Joint Missingness), some users might also interpret the red arcs differently than what they should be. In addition, users may also assume that the red arcs indicate statistical correlation between variables, rather than simply the co-occurrence of missing values within the same records. The arcs may also be interpreted as visualising a directional flow of missingness from one variable to another, suggesting causal relationships that do not actually exist.

### 4.3.2 Evaluation

To assess the interpretability and effectiveness of the MissiG visualisation across three different datasets with varying dimensionality, a semi-structured interview protocol was conducted using three participants. All participants were university students, with varying levels of experience in data visualisation. Specifically, only one participant had prior hands-on experience with visualisation tools, while the other two possessed only a general understanding of basic data visualisation concepts and tools. For confidentiality purposes of this summary, the participants are

referred to as User A, User B, and User C. Additionally, the terms features and variables were also used interchangeably throughout this summary of evaluation.

The visualisation of the three datasets was evaluated on a 15.6-inch laptop screen with a resolution of 1920×1080 pixels. The interview session began with a brief introduction to the key features of the visualisation interface, including the reset key ‘r’ or ‘R’ on keyboard, zooming, and panning functionalities. Participants were also informed that the purpose of the visualisation was to represent patterns of missing data within the dataset. Users were also asked whether they had prior experience or knowledge of data visualisation. Only User B reported having prior experience, while Users A and C had limited exposure to only general tools such as histogram and line charts.

To validate the concerns outlined in the previous section, an initial quick exploration phase was conducted using the small dataset. During this phase, participants were asked to interpret the visualisation without interacting with any elements, specifically, without hovering over any visual components that would trigger tooltip descriptions. Instead, users were prompted to observe and interpret the visualisation based solely on its static presentation. All the users were guided through the interface by pointing at each key visual element and asked to explain what they believed it represented. This method aimed to assess the intuitive clarity of the visualisation design in the absence of interactive guidance. The responses revealed several misinterpretations:

- All users initially assumed that the red histograms represented the distribution of missing values within each variable.
- User A interpreted the red arcs as indicators of the magnitude of statistical correlation between variables.
- User C suggested that red arcs represent a directional flow of missing values from one variable to another.

The above confirmed the expected misinterpretations outlined in Section 4.3.1. Despite that, User B was the only participant who correctly identified most of the key visual elements. User B accurately hypothesised the meaning of the red arcs, correctly associating them with the Joint Missingness (JM) pattern. This interpretation was based on a visual comparison of the arc thickness between two features and their corresponding red bar charts, indicating a deeper level of pattern recognition. In contrast, the remaining users were largely unable to provide answers to this question. It is also important to note that all participants were explicitly informed in advance that it was acceptable if they could not interpret certain elements, as the purpose of the exercise was to observe natural understanding rather than to test prior knowledge. The responses indicated suggested that, while the MissiG visualisation is effective in representing missingness patterns, its design can be easily misunderstood without adequate explanation, particularly by users with limited experience in data visualisation.

Following the exploratory phase, participants were given a structured briefing that introduced the MissiG design and its representation of three missingness patterns, which are Amount Missing (AM), Joint Missingness (JM), and Conditional Missingness (CM). The datasets used in the evaluation were also presented, and users were informed of the availability of interactive tooltips that appear when hovering over key elements. The briefing session lasted approximately 15 to 25 minutes to ensure all the participants have full understanding of the MissiG design. All users

demonstrated a clear understanding of AM and JM following the explanation. However, it took them a longer time to understand the CM pattern, which is represented by the relationship of the missing values in a selected variable with the bias distribution shown by its subsets of recorded values in other variables. Then, the next exploration phase was initiated. The participants were given a task to explore all the datasets while being given a consistent set of questions for each of them:

1. Which feature appears to have the highest/ lowest/no missing values (AM)?
2. Do you observe any joint missingness (JM) between variables? How did you identify it (e.g., red bars or red arcs)?
3. Is there any evidence of conditional missingness (CM) in the dataset?
4. What additional information did you gain from the visualisation?
5. Overall, are the missingness patterns identifiable in this scale?

All participants were able to successfully identify Amount of Missingness (AM) patterns across all three datasets. However, as dataset size increased, this task became more challenging due to the reduction in the size of visual elements required to accommodate higher dimensionality. Users, specifically noted that in the large dataset (Kamyr Digester), features with only a single missing value, such as T-Tops-Chips-4 were difficult to detect without utilising the zoom or hover functions. In contrast, users found it easier to interpret Joint Missingness (JM) patterns across all dataset sizes, primarily due to the visibility of red arcs, which clearly highlighted missingness relationships between features. In instances where only one joint missing value existed, again observed in the Kamyr Digester dataset, participants reported that the red arc was crucial for detecting the pattern. The corresponding red bar was too thin to be noticeable without interaction, such as zooming or hovering. User responses indicate that as dataset dimensionality increases, red arcs become the most relied-upon visual cue for identifying JM patterns. It is also worth noting that participants frequently depended on descriptive tooltips to assist in interpretation, particularly when distinguishing between features with low or no missing values. This was especially relevant in the Kamyr Digester dataset, which contained numerous features with only a single missing value as shown in Figure 31.

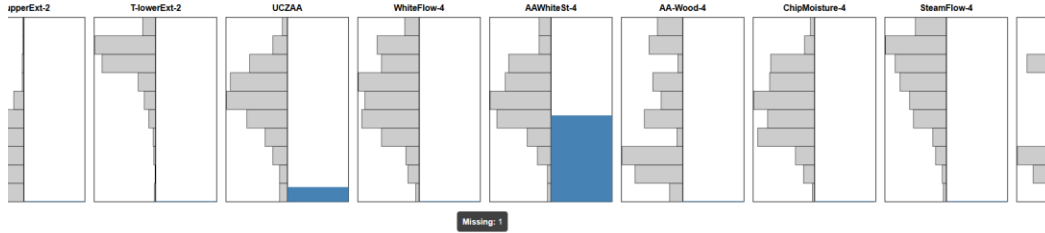


Figure 31: MissiG with linear layout implemented on Kamyr Digester dataset. AM patterns for certain variables are too small to be identified.

Concerning Conditional Missingness (CM) patterns, user responses gathered were mixed. All participants required more time to identify CM patterns as the dataset size increased, highlighting a correlation between the dimensionality of data and the difficulty in interpreting missingness patterns. The dataset with the most frequently unidentified CM patterns was the medium-sized dataset (Sleep Health and Lifestyle). This difficulty was caused by the introduction of multiple CM conditions within a dataset that contained only 374 records. As a result, several individual records exhibited multiple missing values across various variables, preventing a clear distribution of recorded values, which can help in identifying the CM pattern. For instance, red histograms for Stress Level, which were intended to reflect CM relationships with missing values in Quality of Sleep or Sleep Duration, appeared to follow random distributions, making it difficult for users to detect meaningful CM patterns between those variables.

The Kamyr Digester dataset also posed challenges in identifying CM patterns due to its high dimensionality and reduced visual scale. Users needed to inspect individual red histograms closely before recognising CM patterns. An example involved identifying a CM relationship between missing values in BF-CMratio and low recorded values of SulphidityL-4, which took users several minutes of detailed inspection before identifying the CM pattern. On the other hand, the small dataset (Student Performance) was the easiest for users to interpret in terms of CM patterns. This was primarily because no scale reduction was applied, and a lower number of variables prevented visual clutter. Consequently, all the MissiG elements and missingness patterns were presented clearly, allowing participants to derive their observations more efficiently.

Participants also emphasised the importance of understanding the logical context of the data to accurately interpret missingness patterns, particularly CM. This was especially evident in the small and medium datasets, where users could relate variable relationships to real-world scenarios. In contrast, participants reported difficulty in interpreting CM patterns within the Kamyr Digester dataset due to limited familiarity with its domain-specific variables. Without a clear conceptual understanding of these variables, users found it difficult to apply logical assumptions and instead had to rely on extended exploration and inspection.

When asked about the additional insights gained from using the MissiG visualisation, most users reported that they were able to identify potential correlations between variables that could be inferred by comparing their distribution pattern of recorded values. However, User B and C pointed out that certain features, such as Blood Pressure in the Sleep Health and Lifestyle dataset, appeared as numerical despite being categorical. This misinterpretation was attributed to the sequential arrangement of category bins from lowest to highest, which visually resembled a continuous numerical scale. Additionally, the large number of categories compressed the bin widths, making it difficult to distinguish whether the variable was categorical or numerical based only on its visual presentation.

Participants also pointed out another weakness in the visualisation, particularly in the Kamyr Digester dataset. Features with only a single missing value appeared visually similar to features with no missing values. This was primarily due to the thickness of the blue bar chart outline, which reduced the visibility of the missingness indicator and hindered users' ability to distinguish subtle differences in missingness patterns. These usability issues highlight areas for future improvement, particularly in enhancing the clarity of categorical bin presentation and refining the visual treatment of features with minimal missingness.

Finally, users were asked to suggest potential improvements to enhance the effectiveness of the MissiG visualisation. User A recommended using a more diverse colour scheme to improve the distinction between visual elements. In particular, the user suggested against using red for histograms, as this colour was strongly associated with missingness indicators and led to confusion when interpreting recorded value distributions. User A also suggested incorporating a legend to serve as a quick reference, which will be very helpful, especially for users unfamiliar with the visual encodings of MissiG technique. This feedback was aligned with the several instances where the user expressed uncertainty and required clarification regarding the meaning of specific visual components.

User C proposed a more substantial change by recommending a complete redesign of the layout, particularly to accommodate datasets with a large number of variables. The user observed that the current linear layout, while effective for smaller datasets, may become less efficient or readable as dimensionality increases. User B echoed User A's concerns about the use of red, stating that the colour causes confusion during the evaluation process. However, the user also reported becoming more familiar with the visual encodings over time and adapting to the interface through continued interactions. Additionally, User B acknowledged that challenges in interpreting large datasets are to be expected, given the inherent visual and cognitive complexity of high-dimensional data.

Suggestions from users emphasised that, while interactive visual refinements can improve usability, the underlying difficulty of exploring such data should be addressed through supportive interface features, such as clearer legends or improved layout strategies. The response from User B also implied that the familiarity of using the technique will scale with the rate of identifying the missingness patterns.

In summary, all participants were able to identify AM and JM patterns with relative ease, especially when guided by red arcs and interactive tooltips. However, interpreting Conditional Missingness (CM) proved more challenging, especially in the medium and large datasets, due to overlapping of conditional missingness conditions, reduced visual scale, and unfamiliar domain-

specific variables. Participants emphasised the importance of both visual clarity and contextual understanding to accurately interpret CM patterns. Users also point out weaknesses in the visualisation and suggest some ideas that can be adapted to improve the visualisation.

The effectiveness of the MissiG visualisation is assessed primarily in terms of usability and interpretability. Based on participant responses, the system was generally found to be usable across datasets with varying dimensionality. Users were able to navigate the visualisation comfortably using the zooming, panning, and reset key functionalities, which supported interaction and exploration. However, the interpretability of the visualisation, particularly the identification of missingness patterns, declined as the number of variables increased.

Several factors contributed to this drop in performance. As dataset dimensionality increased, the system applied scale reduction to ensure that all variables could be displayed on a single screen. While necessary, this reduction in visual element size made it more difficult for users to quickly recognise missingness patterns, requiring more detailed inspection and prolonged interaction.

Another challenge emerged from the structure of the missingness itself. In the medium-sized dataset (Sleep Health and Lifestyle), some Conditional Missingness (CM) patterns were not successfully identified due to overlapping missingness introduced during the pre-processing phase. In the large dataset (Kamyr Digester), many variables contained only a single missing value, which, when combined with the bold outline of the bar chart borders, made it difficult for users to distinguish Amount Missing (AM) patterns without zooming in. Despite these issues, the presence of red arcs offered a helpful alternative by allowing users to detect joint or related missingness patterns more easily.

Overall, the evaluation suggests that while MissiG with a linear layout is effective at lower dimensions, its interpretability decreases as dataset complexity increases. The visual scaling required to support high-dimensional data, combined with overlapping missingness patterns and subtle visual distinctions, impacts the clarity of the output and the speed at which users can extract insights. Nonetheless, the feedback also highlights areas where the system can be improved, such as enhancing visual clarity and providing more guided interaction to support interpretation at scale.

# Chapter 5: Conclusion

## 5.1 Introduction

This chapter concludes the dissertation by reflecting on how the objectives were met, identifying the project's limitations, and outlining potential improvements to the current implementation. It also offers suggestions for future work that may benefit researchers and developers working on the same or similar visualisation techniques.

## 5.2 Fulfilment of Objectives

The objectives of the project are: -

- Develop and implement the MissiG glyph using D3.js, a suitable JavaScript library for visualisation.
- Identify or generate three datasets with different sizes (number of variables) and missing value distributions, and implement MissiG visualisation on them.
- Evaluate the effectiveness of the MissiG visualisation on the three datasets through qualitative user feedback.

The first objective of the project, implementing the MissiG technique using the D3.js library, was successfully achieved. The final visualisation includes interactive features such as zooming, panning, a reset key, and informative tooltips to improve usability. A scale reduction mechanism was also introduced to maintain clarity when rendering a large number of variables. The implementation followed the layout design created during the initial development stage in Figma. Although a minor issue related to the measurement of red arcs was encountered, it was promptly resolved during the testing phase to remove visual clutter.

The second objective was also fulfilled. The project incorporated three datasets representing small, medium, and large scales, each with a different number of variables. For datasets that were originally complete or lacked sufficient missingness, such as Student Performance and Sleep Health and Lifestyle, custom Python scripts were developed to introduce both conditional and random missingness. The Kamyr Digester dataset, which already contained real missing values, was used as the large-scale dataset for development and evaluation.

Finally, the third objective, evaluating the effectiveness of the MissiG visualisation, was addressed through a semi-structured interview involving three participants. A structured set of questions focused on the interpretability of missingness patterns and the overall usability of the system was used for the interviews with the participants. Feedback was collected and analysed in detail. Based on the results, the project successfully met its primary aim, which is to investigate the effectiveness of the MissiG technique across datasets of varying dimensionality.

### 5.3 Limitations and Weaknesses

Several limitations and weaknesses were identified during the development of this project. One of the primary limitations was the narrow project scope. The evaluation focused solely on the linear layout of the MissiG technique, which means that insights into its performance are restricted to a single layout configuration. As a result, the broader capability of the MissiG system, including comparisons between the linear and radial layouts, could not be fully assessed.

Another notable limitation lies in the datasets used for visualisation. Among the three datasets, only the Kamyr Digester dataset is a real-world dataset, with naturally occurring missingness patterns. The other two datasets (Student Performance and Sleep Health and Lifestyle) are synthetic or originally complete and were modified using customised Python scripts to introduce different type of missingness patterns. This approach was necessary due to the limited availability of real-world datasets that contain rich missingness structures suitable for visualisation with MissiG. Additionally, these datasets were selected based on strong correlations between variables to allow for the introduction of Conditional Missingness (CM), while random missingness was also added to simulate real-world conditions. However, these controlled environments may have introduced bias into the evaluation, as users might have found it easier to detect patterns that were artificially created rather than organically formed.

A further challenge was observed in the visualisation of complex CM patterns, particularly in the Sleep Health and Lifestyle dataset. The combination of several overlapping conditional relationships led to random-looking red histograms, making it difficult for participants to interpret the CM patterns accurately. In these cases, the issue stemmed not from the MissiG technique or the visualisation's interactive features, but from the visual ambiguity introduced by excessive overlap in missingness distributions.

Several limitations were identified in the evaluation process itself. Only three participants were involved in the semi-structured interviews, which is a small sample size for drawing general conclusions. Furthermore, majority of participants that involved in the evaluation, had limited experience in data visualisation, which may have influenced their ability to interpret complex patterns and added variability to the feedback. This restricts the generalisability of the findings and highlights the need for broader user testing in future studies.

Finally, several weaknesses of the project lie in the quality of visualisation, which were identified by participants during the evaluation. For instance, in cases where a variable contained only a single missing value, the red bar indicating AM or JM became very thin and overlapped with the glyph's border, making it difficult to detect. Some participants also expressed confusion regarding the rendering of categorical histograms, especially when a variable had many categories. The bins appeared visually continuous, giving the impression of a numerical distribution. As the number of categories increased, the individual bin height was reduced to fit the available space, making the gaps between bins less visible unless zooming or tooltips were used. These issues may affect the perceived quality and clarity of the visualisation.

## 5.4 Personal Development

Working on this project allowed me to grow significantly in both technical and personal capacities. Prior to the dissertation, my understanding of data visualisation was limited to basic methods like scatter plots and bar charts. A lot of new and modified methods were discovered during the research phase of the dissertation, providing a clearer view of how visualisation techniques can be applied to complex challenges, specifically in dealing with missing data.

One of the most valuable parts of this journey was learning how to use D3.js. With little experience in JavaScript, the author gradually became more confident by actively coding and consulting documentation. This experience led to a stronger understanding of interactive visualisation, especially with SVG manipulation and transitions, both of which were crucial for building the MissiG interface.

The project also deepened my understanding of the role of missing data visualisation. I have learned how to visually express patterns such as AM, JM, and CM, and began to recognise how these representations can support better decision-making in data preprocessing. By conducting semi-structured interviews, I have gained practical experience in usability evaluation and saw the importance of building visual tools that are informative and user centred.

Overall, the project helped me to develop confidence in working with unfamiliar technologies, tackle complex problems independently, and understand how thoughtful design like MissiG, contributes to a meaningful data analysis.

## 5.5 Future Work

Future developments of the MissiG visualisation system could focus on enhancing both the clarity and usability of the visual elements. During the evaluation phase, users suggested improvements such as introducing distinct colour encodings and incorporating an informative legend to better explain the meaning of visual components like red arcs, histograms, and bar charts. Additionally, each missingness patterns can be assigned to different colour encodings. These changes may increase interpretability of missingness patterns, specifically for users with limited experience in data visualisation.

Furthermore, evaluation process also presents several improvement opportunities. In this project, qualitative feedback was gathered from a small number of participants through semi-structured interviews. Future evaluations could benefit from involving more people from diverse backgrounds, which would improve the reliability and generalisability of the results. Employing quantitative evaluation methods, such as usability testing and task performance measurement, would also allow for more objective assessments. For instance, measuring task completion time or the number of interactions required to identify specific missingness patterns might provide concrete metrics for comparing usability across users. These methods would offer more precise results for evaluating the effectiveness of the visualisation and identifying usability issues.

Another important direction for future work is the implementation of the radial layout, an alternative to the linear layout used in this project. The radial design is particularly effective at representing Joint Missingness (JM) through evenly distributed arcs around a central variable, minimising perceptual bias introduced by the arcs' length. A comparative implementation of both layouts within the same system would allow users to switch between views and evaluate which layout is more suitable based on dataset characteristics. Such a feature would also enable a comparative analysis of layout performances, providing more insights into the strengths and weaknesses of each layout.

The domain scope of this project was intentionally broadened, covering datasets from various sources to evaluate MissiG's adaptability. However, future work could explore the visualisation's application in domain-specific contexts, such as medical, scientific, or social science datasets, where the implications of missing data are more diverse. Specifying the visualisation to domain-specific needs, such as variable-specific expectations in experimental science, could increase its practical needs.

Finally, future projects should aim to rely more heavily on real-world datasets that contain naturally occurring missingness. While synthetic datasets offer control for development and testing, they may not fully capture the complexity of real data. Using real-world data would not only enhance the validity and reliability of the findings but also provide stronger evidence of the system's effectiveness in practical settings.

# Bibliography

- [1] Eaton, C., Plaisant, C. and Drizd, T. (2005). Visualizing Missing Data: Graph Interpretation User Study. *Lecture Notes in Computer Science*, pp.861–872. doi: [https://doi.org/10.1007/11555261\\_68](https://doi.org/10.1007/11555261_68).
- [2] Alsufyani, S., Forshaw, M. and Fernstad, S.J. (2024). Visualization of missing data: a state-of-the-art survey. *arXiv (Cornell University)*. doi: <https://doi.org/10.48550/arxiv.2410.03712>.
- [3] Johanssonfernstad, S. and Johansson, J. (2021). To Explore What Isnt There Glyph-based Visualization for Analysis of Missing Values. *IEEE Transactions on Visualization and Computer Graphics*, pp.1–1. doi: <https://doi.org/10.1109/tvcg.2021.3065124>.
- [4] Fernstad, S.J. (2018). To identify what is not there: A definition of missingness patterns and evaluation of missing value visualization. *Information Visualization*, 18(2), pp.230–250. doi: <https://doi.org/10.1177/1473871618785387>.
- [5] Valero-Mora, P., Rodrigo, M.F., Sanchez, M. and SanMartin, J. (2019). A Plot for the Visualization of Missing Value Patterns in Multivariate Data. *Practical assessment, research & evaluation*, 24(1), p.9. doi: <https://doi.org/10.7275/94ra-1y55>.
- [6] Jiménez, E. and Macías, R. (2022). Graphical Tools for Visualization of Missing Data in Large Longitudinal Phenomena. *Computer Graphics Forum*, 41(1), pp.438–452. doi: <https://doi.org/10.1111/cgf.14445>.
- [7] Alsufyani, S., Forshaw, D.M., Del Din, D.S., Yarnall, P.A., Rochester, P.L. and Fernstad, S.J. (2024). Multi-level visualization for exploration of structures in missing data. *EG UK Computer Graphics & Visual Computing*, pp.1–9. doi: <https://doi.org/10.2312/cgvc.20241212>.
- [8] Chung, D.H., Legg, P.A., Parry, M.L., Bown, R., Griffiths, I.W., Laramee, R.S. and Chen, M. (2013). Glyph sorting: Interactive visualization for multi-dimensional data. *Information Visualization*, 14(1), pp.76–90. doi: <https://doi.org/10.1177/1473871613511959>.
- [9] Maguire, E., Rocca-Serra, P., Sansone, S.-A., Davies, J. and Chen, M. (2012). Taxonomy-Based Glyph Design—with a Case Study on Visualizing Workflows of Biological Experiments. *IEEE Transactions on Visualization and Computer Graphics*, 18(12), pp.2603–2612. doi: <https://doi.org/10.1109/tvcg.2012.271>.
- [10] Huang, T. (2024). *FEAD: Figma-Enhanced App Design Framework for Improving UI/UX in Educational App Development*. [online] doi: <https://doi.org/10.48550/arXiv.2412.06793>.
- [11] Figma (2024). *Figma: the Collaborative Interface Design tool*. [online] Figma. Available at: <https://www.figma.com/>.
- [12] Bostock, M., Ogievetsky, V. and Heer, J. (2011). D<sup>3</sup> Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), pp.2301–2309. doi: <https://doi.org/10.1109/tvcg.2011.185>.

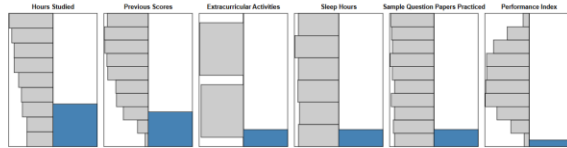
- [13] plotly.com. (n.d.). *Plotly JavaScript Graphing Library*. [online] Available at: <https://plotly.com/javascript/>.
- [14] www.chartjs.org. (n.d.). *Chart.js documentation*. [online] Available at: <https://www.chartjs.org/docs/latest/>.
- [15] Kaggle (2024). *Kaggle: Your home for data science*. [online] Kaggle.com. Available at: <https://www.kaggle.com/>.
- [16] Google (2019). *Google Colaboratory*. [online] Google.com. Available at: <https://colab.research.google.com/>.
- [17] www.kaggle.com. (n.d.). *Student Performance (Multiple Linear Regression)*. [online] Available at: <https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression?resource=download>.
- [18] THARMALINGAM, L. (2023). *Sleep Health and Lifestyle Dataset*. [online] www.kaggle.com. Available at: <https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset>.
- [19] Dayal, B.S., Macgregor, J.F., Taylor, P.A., R. Kildaw and S. Marcikic (1994). Application of feedforward: neural networks and partial least squares regression for modelling kappa number in a continuous Kamyr digester: how multivariate data analysis might help pulping. 95(1), pp.26–32.

# Appendix

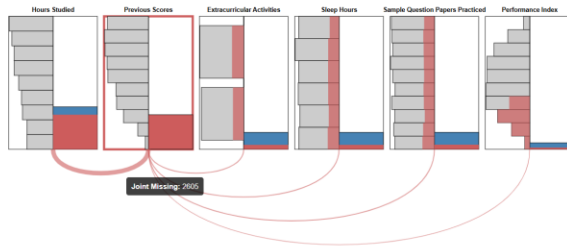
## Final Results - MissiG

### 1. Student Performance (Small Dataset)

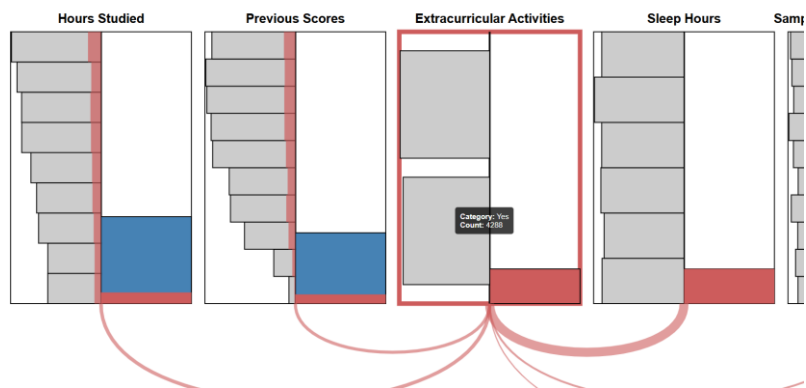
Student Performance



Student Performance

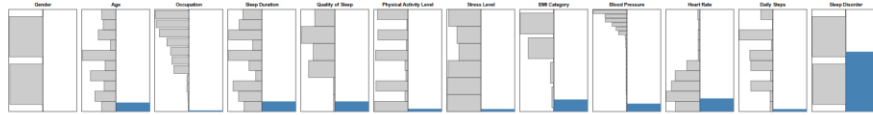


Student Performance

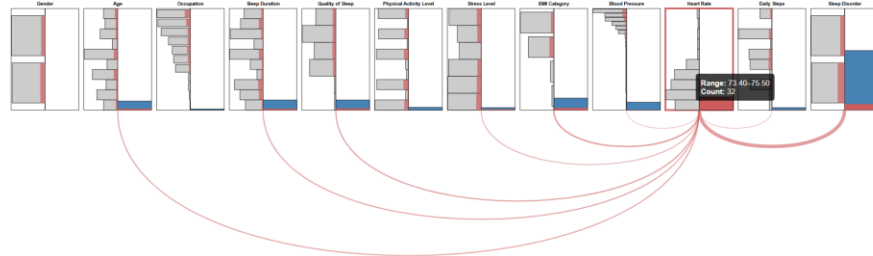


## 2. Sleep Health and Lifestyle (Medium Dataset)

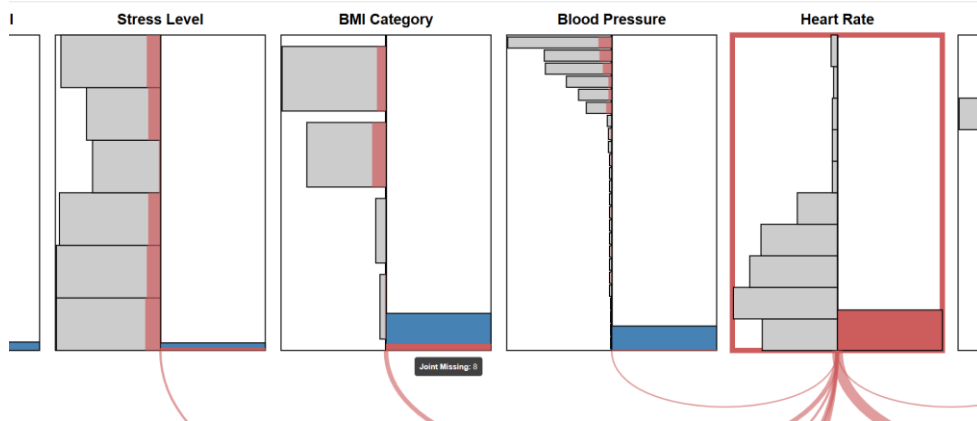
Sleep Health and Lifestyle



Sleep Health and Lifestyle

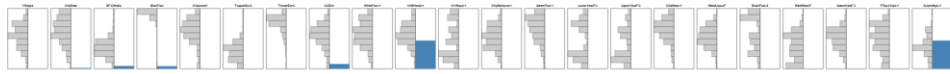


Sleep Health and Lifestyle

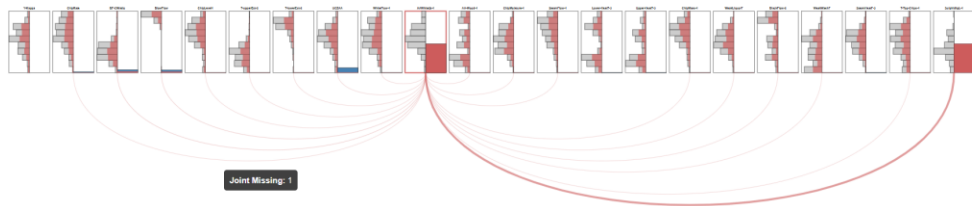


### 3. Kamyr Digester (Large Dataset)

Kamyr Digester



Kamyr Digester



Kamyr Digester

