

**GROUP ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**CT127-3-2-PFDA**

**PROGRAMMING FOR DATA ANALYSIS**

**INTAKE CODE: APD2F2511**

**ASSIGNMENT HAND OUT DATE: 24th November 2025**

**ASSIGNMENT HAND IN DATE: 7th February 2026**

**LECTURER: DR. KULOTHUNKAN PALASUNDRAM**

**Student Details:**

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **TP Number** |
| 1. | Choong Ti Huai | TP078539 |
| 2. | Daniel Chan Zit Fung | TP079018 |
| 3. | Gan Yew Joe | TP077191 |
| 4. | Liu Wei | TP085412 |

Table of Contents

[1 Introduction 3](#_Toc220523433)

[1.1 Project Description 3](#_Toc220523434)

[1.2 Data Description 2](#_Toc220523435)

[1.3 Data Preparation 4](#_Toc220523436)

[1.3.1 Dataset Merging 4](#_Toc220523437)

[1.3.2 Data Format Conversion 9](#_Toc220523438)

[1.3.3 Distribution Check 16](#_Toc220523439)

[1.3.4 Outlier Detection 23](#_Toc220523440)

[1.3.5 Outlier to NA 23](#_Toc220523441)

[1.3.6 Multivariate Outlier Detection 23](#_Toc220523442)

[1.3.7 Multivariate Outlier to NA 23](#_Toc220523443)

[1.3.8 Data Health Check 25](#_Toc220523444)

[1.3.9 Preprocessing for Missing Values Imputation – not yet completed 38](#_Toc220523445)

[1.3.10 Categorical Cleaning 46](#_Toc220523446)

[1.3.11 Final Quality Check 53](#_Toc220523447)

[1.3.12 1.3.x MICE Imputation 62](#_Toc220523448)

[1.3.13 1.3.x Imputation Validation 64](#_Toc220523449)

[1.4 Assumptions 66](#_Toc220523450)

[1.4.1 no remove duplicate 66](#_Toc220523451)

[1.4.2 no pay ransom 66](#_Toc220523452)

[2 2.0 Analysis (Individual) 68](#_Toc220523453)

[2.1 Choong Ti Huai (TP078539) 68](#_Toc220523454)

[2.1.1 2.1.1 Objectives 68](#_Toc220523455)

[2.1.2 2.1.2 Analysis Questions 68](#_Toc220523456)

[2.1.3 2.1.3 Data Preparation 68](#_Toc220523457)

[2.1.4 2.1.4 Data Analysis 69](#_Toc220523458)

[2.1.5 2.1.5 Hypothesis Formulation and Testing 69](#_Toc220523459)

[2.2 Daniel Chan Zit Fung (TP079018) 70](#_Toc220523460)

[2.1.6 2.2.1 Objectives 70](#_Toc220523461)

[2.2.2 Analysis Questions 70](#_Toc220523462)

[2.1.7 2.2.3 Data Preparation 70](#_Toc220523463)

[2.1.8 2.2.4 Data Analysis 70](#_Toc220523464)

[2.1.9 2.2.5 Hypothesis Formulation and Testing 71](#_Toc220523465)

[2.3 Gan Yew Joe (TP077191) 72](#_Toc220523466)

[2.1.10 2.3.1 Objectives 72](#_Toc220523467)

[2.3.2 Analysis Questions 72](#_Toc220523468)

[2.3.3 Data Preparation 72](#_Toc220523469)

[2.3.4 Data Analysis 72](#_Toc220523470)

[2.3.5 Hypothesis Formulation and Testing 73](#_Toc220523471)

[2.4 Liu Wei (TP085412) 74](#_Toc220523472)

[2.1.11 2.4.1 Objectives 74](#_Toc220523473)

[2.1.12 2.4.2 Analysis Questions 74](#_Toc220523474)

[2.1.13 2.4.3 Data Preparation 74](#_Toc220523475)

[2.1.14 2.4.4 Data Analysis 74](#_Toc220523476)

[2.1.15 2.4.5 Hypothesis Formulation and Testing 75](#_Toc220523477)

[3 3.0 Group Hypothesis 76](#_Toc220523478)

[3.1 Group Hypothesis Formulation 76](#_Toc220523479)

[3.1 Group Hypothesis Testing 76](#_Toc220523480)

[3.2 Overall Conclusion 76](#_Toc220523481)

[4 4.0 Summary 77](#_Toc220523482)

[4.1 Limitations and Recommendations 77](#_Toc220523483)

[4.2 Title 77](#_Toc220523484)

[4.3 Title 77](#_Toc220523485)

[5 Title 5 78](#_Toc220523486)

[4.4 Title 78](#_Toc220523487)

[4.5 Title 78](#_Toc220523488)

[4.6 Title 78](#_Toc220523489)

[6 Title 6 79](#_Toc220523490)

[4.7 Title 79](#_Toc220523491)

[4.8 Title 79](#_Toc220523492)

[4.9 Title 79](#_Toc220523493)

[7 Conclusion 80](#_Toc220523494)

[8 Appendix 82](#_Toc220523495)

# Introduction

In a world where digital addiction is on the rise, web infrastructure security has become one of the major concerns of organizations across the globe. Cyber-attacks, or to be more exact, defacements of the websites are not only a technical violation but also a great risk to the reputation and financial standing of an organization. This report captures an analytical teamwork to analyze a massive set of data of cyber-attack cases. With the help of R and RStudio, the project will be able to make raw incident data useful, determining the major trends in the behavior of an attacker and the economic consequences of their activity.

## Project Description

The project is based on the overall analysis of a dataset comprising of 592,765 documented cases of website defacement. Website defacement is a type of cyber-attack in which a third party has modified the visual nature or the contents of a web site, typically to convey a message or embarrass their institution in some way.

The main objective of the analysis is to utilize the sophisticated R programming and data analysis methods to attain the following:

* Data Integration & Quality

Effectively consolidate and harmonize heterogeneous data to form an authoritative single source of truth to analyze.

* Pattern Detection

This is used to discern patterns with respect to the geographical location of attacks as well as the most susceptible web server version to compromise.

* Impact Assessment

Measure the ransom demands, system downtime, and overall financial losses of such attacks to determine the socio-economical impact of such attacks.

* Evidence-Based Insights

Test and formulate hypotheses to bring evidence-based recommendations that may help inform organizational decision-making and cybersecurity policies.

Analytical work is performed by means of a team-based methodology that offers the integration and standardization of raw data of three formats (CSV, Excel, and Text). The situation is based on studying the correlation between such technical server features as IP addresses and host countries and the extent of the damage (financial losses and downtime) and presents a comprehensive picture of the current cyber-threat environment.

## Data Description

The data used in this research is a multidimensional data repository on the case of defacing websites. As a way of giving a concise picture of the data architecture, the variables have been grouped into three areas namely incident metadata, technical infrastructure, and socio-economic impact.

**Tracking and Metadata on Incidents.**

1. Date of Incident (**Date**): This time-based parameter involves the actual calendar date on which the defacement was initially discovered and documented. It is crucial in detecting seasonal pattern or an increase or decrease in hacks over a period of time.
2. Reporting Entity (**Notify**): This variable specifies the individual, organization, or computer system (the so-called notifier) that officially notified the incident of the reporting system. It assists in classifying the presence of discoveries of attacks by independent researchers or publicly facing groups.
3. Targeted Uniform Resource Locator (**URL**): This will give the specific uniform resource locator (web address) of the affected site which will give an opportunity to identify specific industries or fields that are targeted by the hackers.

**Technical Infrastructure Characteristics.**

1. IP Address of Server (**IP**): This is a technical identity which identifies a unique internet protocol (IP) address of the server attacked. It acts as an online signature of the infrastructure that was used in the breach.
2. Hosting Country (**Country**): This refers to the physical or virtual location of the physical or virtual server. The study of this area assists in the realization of at-risk web infrastructure all over the world.
3. Web Server Environment (**WebServer**): This field captures the type of software and version applied to the server (Apache, Nginx, or Microsoft-IIS). It is essential to determine the most commonly exploited technologies of servers.
4. Character Encoding (**Encoding**): This feature determines the character set (e.g., UTF-8, ISO-8859) the defacement message is coded in to give hints of the linguistic or geographical origin of the attackers.

**Impact and Financial Variables.**

1. Ransom Demands (**Ransom**): This numerical variable, measured in thousands of dollars (000), is the amount of monetary extortion request that is linked with the breach. It is a mirror of the direct economic incentive of the cyber-attack.
2. System Downtime (**DownTime**): This measure is used to follow the number of days that the targeted system was not available to authorized users. It is one of the key indicators of the impact of the incident on the functioning.
3. Financial Loss (**Loss**): This feature approximates the total lost revenue by the organization in the attack in addition to the ransom. This is a figure that shows the larger economic loss, lost sales and clean-up costs.

## Data Preparation

The data cleaning step is crucial before doing any analysis of the data. This is said so because it ensures accuracy and reliable insights. Raw datasets often contain duplicates, incorrect entries or outliers that shouldn’t be there in the first place, they bring bad influence on the analysis which causes conclusions to be fundamentally flawed. (Unimrkt, 2023) Hence, this section outlines a 14 steps full data cleaning pipeline from messy data to cleaned, analysis-ready data.

### Dataset Merging

First step into the data cleaning is merging, this ensures that it cleans the data in all three datasets in one go.

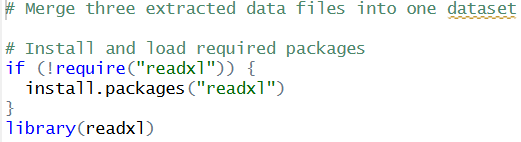


Figure 1 Installation and Loading of Packages

Figure 1 shows the installation and loading of the “readxl” package. The package is used to import data from Microsoft Excel files efficiently.

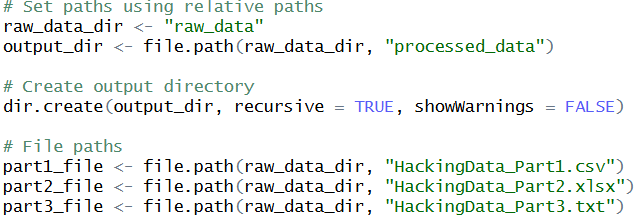


Figure 2 Directory Creation and File Path Defining

In figure 2, it shows the creation of directory and the defining of the file path where the raw datasets are located.

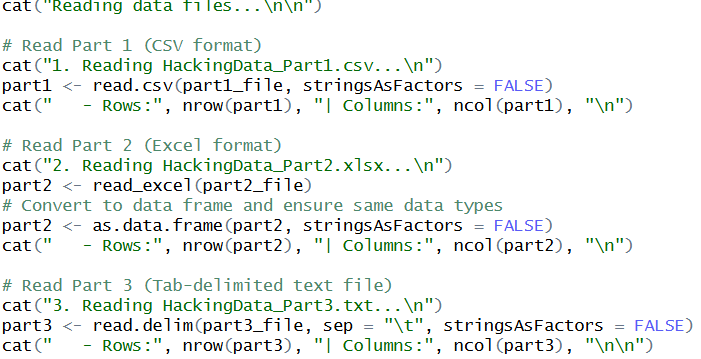


Figure 3 Read and Load Datasets

After the directory is created, the team can then read and load all three raw datasets. Note that different reading functions are used based on the type of files where the dataset is stored. The first part of the dataset is stored within a comma separated value (CSV) format file, where it separates values using commas, hence, the read.csv() function is used to extract the dataset. The second part of the dataset is stored within an excel file, the read\_excel function is used to extract the dataset. Lastly, the third and last part of the dataset is in a tab-delimeted text file also known as the tab separated value (TSV) file, unlike a CSV file, the values in a TSV file is separated with tabs. The read.delim() function with the passed parameter sep = “\t” is used to extract the dataset stored within.

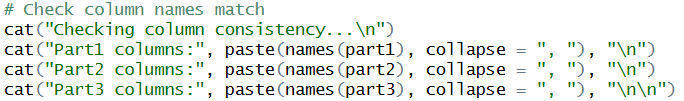


Figure 4 Checking of Column Names

After all the datasets are loaded in, the team then checks the names of the columns. This is done so to ensure the columns stay consistent otherwise the merging process later will ultimately fail.

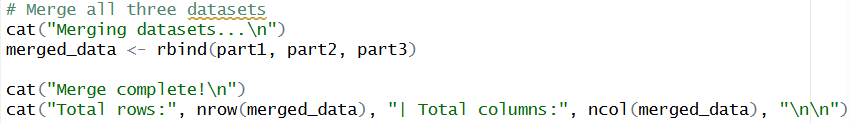


Figure 5 Data Merging

If the column names are consistent, the team can then merge the datasets using the rbind() function, as shown in figure 5.

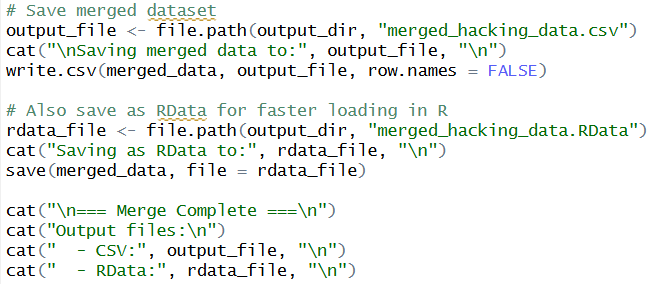


Figure 6 Data Saving

Figure 7 shows the saving of the merged dataset. The merged dataset is saved in two format files, one of which is the common CSV file, and the other is the RData file. The RData file helps provide faster loading in R.

### Data Format Conversion

This step converts the data to appropriate formats for both cleaning and analysis steps. The columns affected include Date, DownTime, Loss and Ransom.

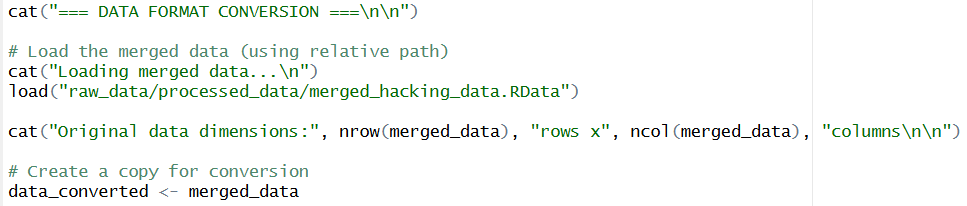


Figure 7 Load Merged Data

The team first load in the merged data and save a copy of it in “data\_converted” as shown in figure 8.

**Date**

The dates within the Date column have multiple formats, this section tries to convert them into the date data type.

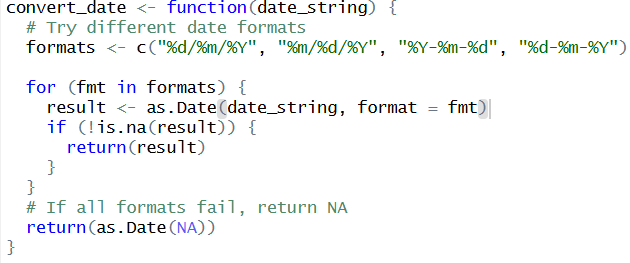


Figure 8 Date Conversion Function

Figure 9 shows the function for the date conversion. Within the function, it loops through various date formats, trying to fit the date. If the function couldn’t find the correct format for the date it returns NA.



Figure 9 Apply Conversion Function

In figure 10, the team converts the dates within the Date column into date data type by applying the date conversion function declared earlier. After the conversion, the dates is then formatted into a standard form, “{year}-{month}-{day}”.

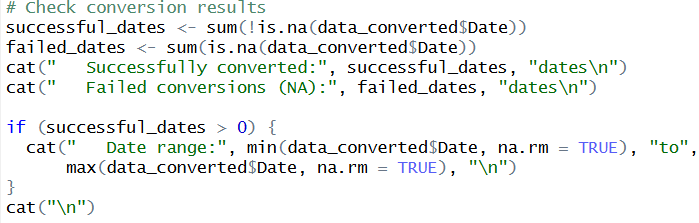


Figure 10 Check Date Conversion Result

As shown in figure 11, a check on the conversion result is carried out. The successful converted dates are stored in the “successful\_dates” whereas the failed ones are stored in “failed\_dates”, which both are then printed out to check the conversion result.

**Ransom**

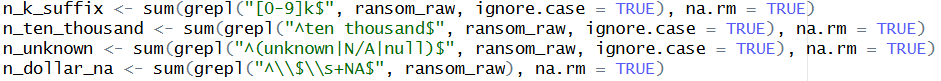


Figure 11 - Check Ransom Conversion Issue

Figure 11 shows the checking of potential failing conversion values within the Ransom variable.

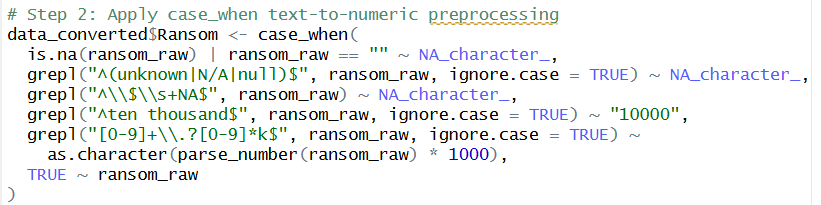


Figure 12 - Ransom Specific Conversion

Figure 12 shows the conversion of the values found earlier.

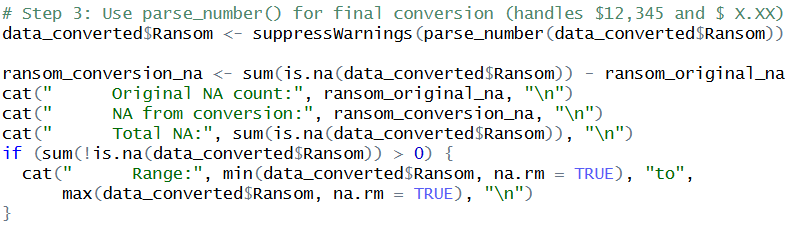


Figure 13 Ransom Conversion

Figure 13 shows the conversion of data in the Ransom column. The data is being converted into numeric data type via the “as.numeric()” and “parse\_number()” function. The number of original NAs in the column and NAs after conversion is stored in “ransom\_original\_na” and “ransom\_conversion\_na”, they are then printed out to see the conversion result.

**DownTime**

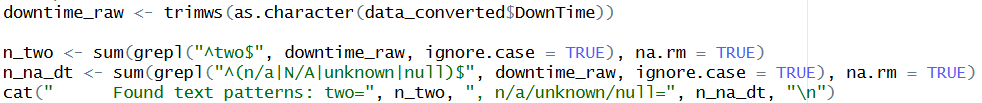


Figure 14 - Check DownTime Conversion Issue

Figure 14 shows the checking process on potential failing conversion.

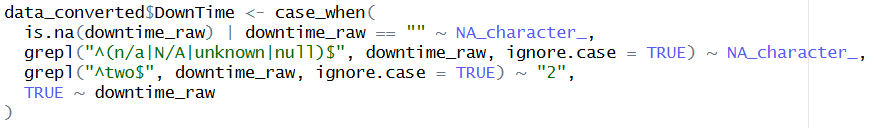
****

Figure 15 - DownTime Specific Conversion

Afterwards, a specific conversion is done on the variable.

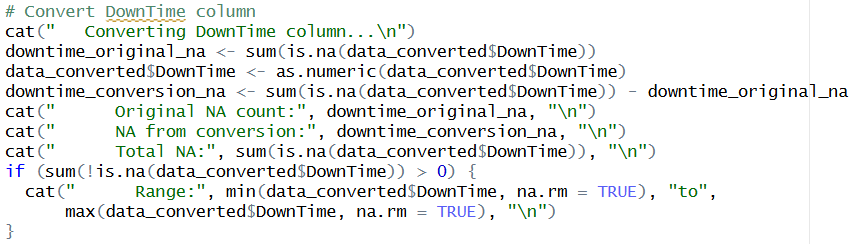


Figure 16 DownTime Conversion

Same goes to the DownTime column, the numeric conversion in Ransom column is applied here as well, with “downtime\_original\_na” storing the number of the original NAs within the column before the conversion and “downtime\_conversion\_na” storing the number of NAs after the conversion.

**Loss**

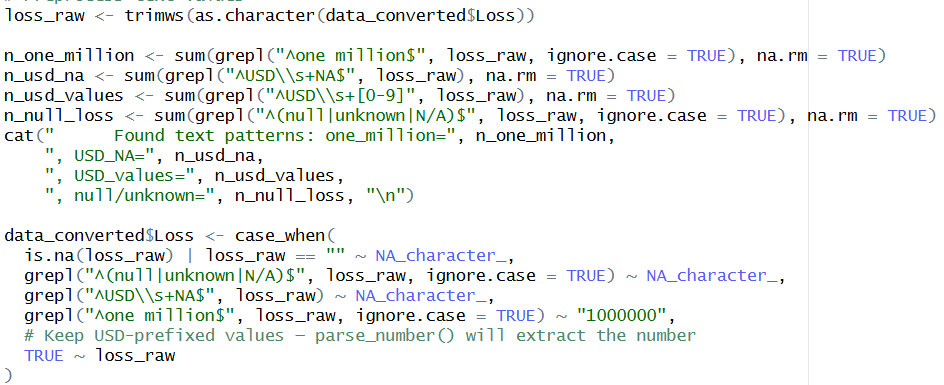
****

Figure 17 - Loss Specific Conversion

Figure 17 shows the specific conversion process on the Loss variable.

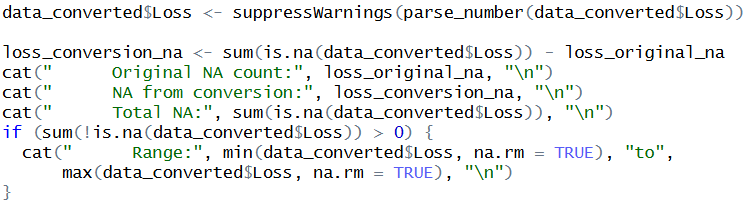


Figure 18 Loss Conversion

The Loss column goes through the same process as Ransom and DownTime column did, converting losses with the column into numeric data types and showing the conversion result.

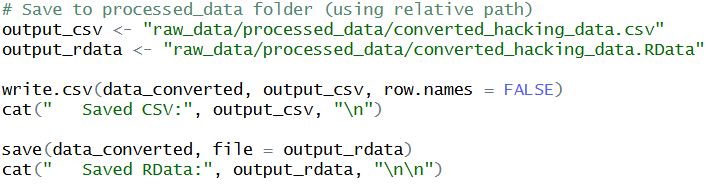


Figure 18 Save Converted Data

Figure 18 shows the saving of the converted data. The converted data is saved through two different file formats, CSV and RData.

### Distribution Check

A distribution check is done after the merging and conversion of the dataset to determine if the data follows normal or skewed distribution, it involves various steps including descriptive statistics, visualization check, normality diagnostic tests and skewness analysis.

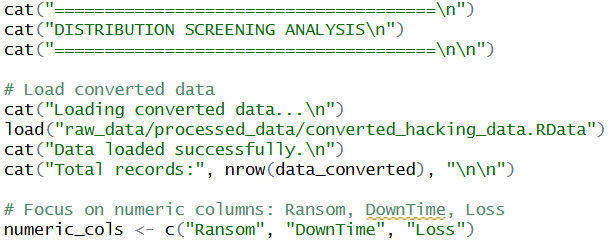


Figure 16 Data Loading

Figure 16 shows the loading of the merged data and declaring the columns to focus this distribution check, which are the Ransom, DownTime and Loss columns.

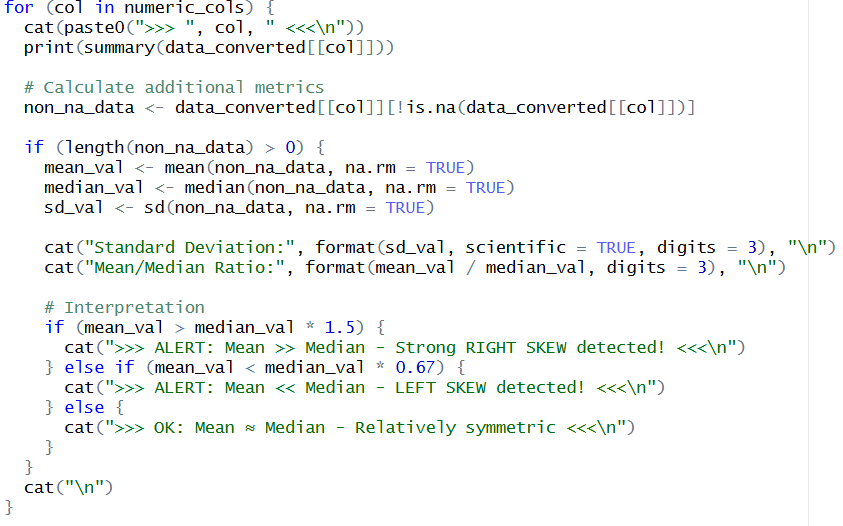


Figure 17 Statistics Summary for Numerical Variables

The process begins with grabbing a list of column names and going through them one by one using a for loop. It then runs the “summary()” function to show the minimum, 1st quartile, median, mean 3rd quartile and maximum. Any missing values are removed before the skewness detection to avoid calculation error. Afterwards, the calculation of standard deviation and mean median ratio is carried out. If the mean value is greater than 1.5 times the median value, it indicates the presence of right skew of that column in the dataset whereas if the mean value is less than 0.67 times the median value indicates left skew.

#### Visualization Check

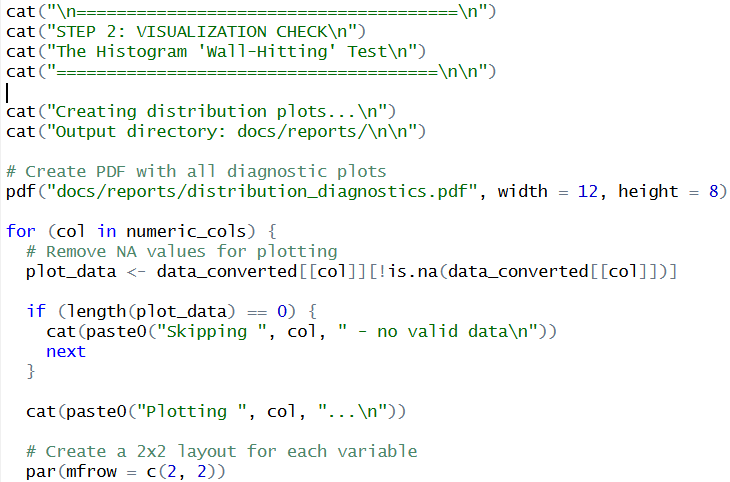


Figure 18 – Preparation for Visualization

Figure 18 shows the preparation done before the visualization of the data. It loops through each focused numerical column and filters out the missing values.

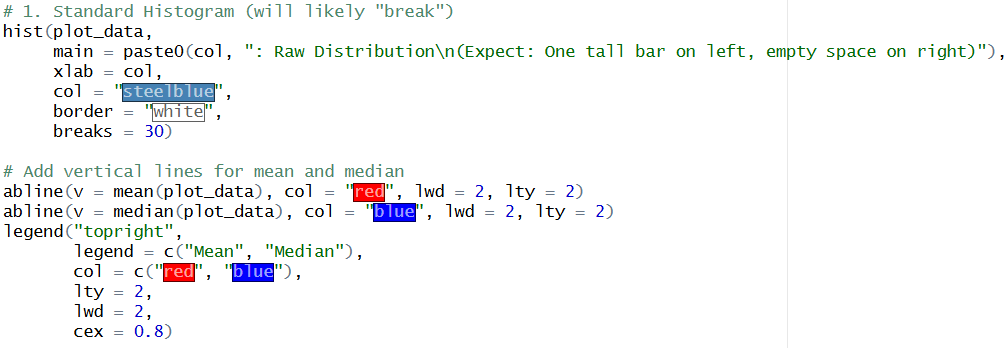


Figure 19 - Plot Raw Histogram

Figure 19 shows the plotting of a raw histogram. It visualizes the “tail” direction which helps identify whether the data is skewed, and if it is, in which direction is it skewing to.

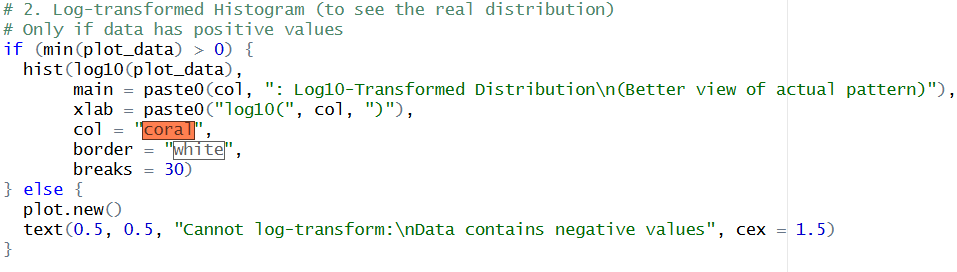


Figure 20 - Plot Log-transformed Histogram

To get a better look at the shape of the data, the team uses log-transformed histogram which utilizes the log10 transformation to squash the massive numbers, as shown in figure 20.

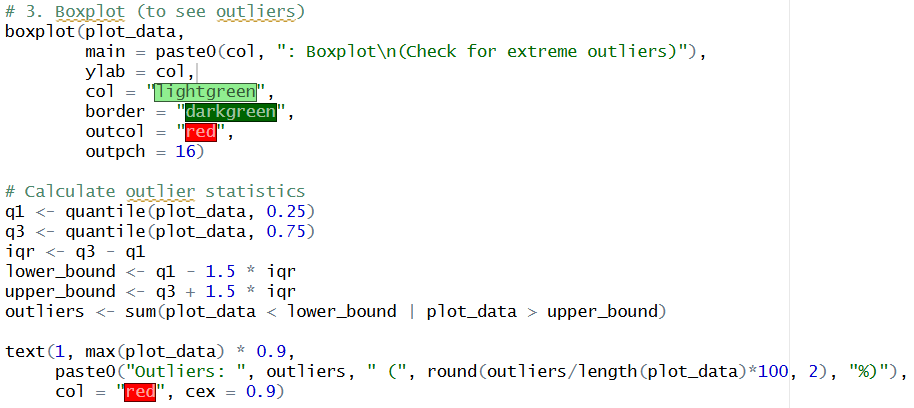


Figure 21 - Plot Boxplot

Figure 21 shows the plotting of a boxplot on the data. Boxplot visualizes the range of the data where the box holds the middle 50% of the data and any individual dots located outside the whiskers are the outliers.

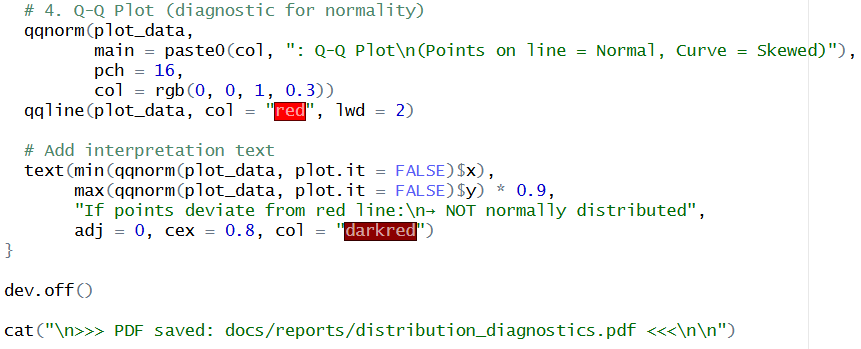


Figure 22 - Plot Q-Q plot

Figure shows the plotting of a Q-Q plot of the data. The Q-Q plot compares the data against a theoretical perfect normal distribution. If the dots are on the red line indicates the data is normal, whereas if the dots are curving away from the line indicates the data is skewed.

##### Ransom

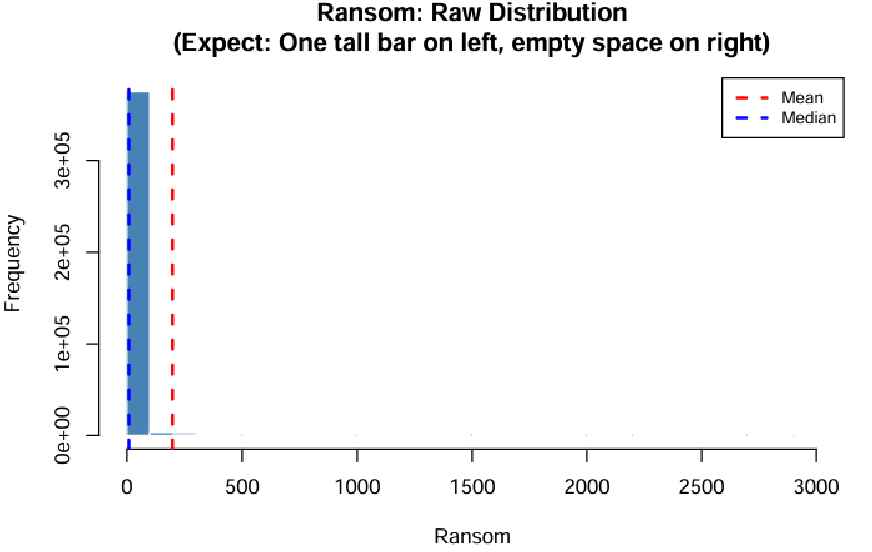


Figure 23 - Ransom Histogram

Figure 23 shows the raw histogram plot of the Ransom variable. It is extreme right skewed, with a tail going to the right which indicates the potential existence of outliers.

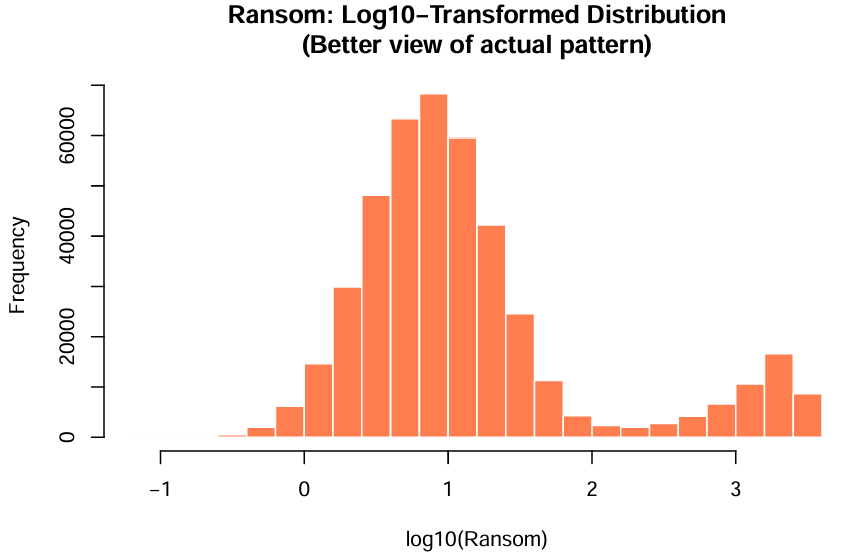


Figure 24 - Ransom Log-Transformed Histogram

Figure 24 shows the output of the Ransom’s Log-Transformed histogram plotting. The plot shows a bimodal distribution pattern where there are two maxima or peaks seen in the histogram. (Bimodal Distribution, 2025)

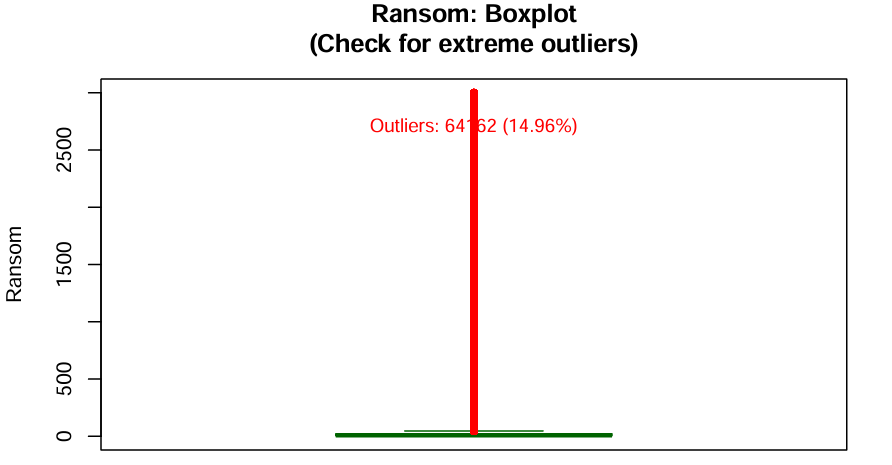


Figure 25 - Ransom Boxplot

Figure 25 shows the boxplot plotting of the Ransom variable. There are a lot of dots lies outside the “whiskers”, they represent the outliers. The outliers takes up 14.96% of the entire data and requires cleaning.

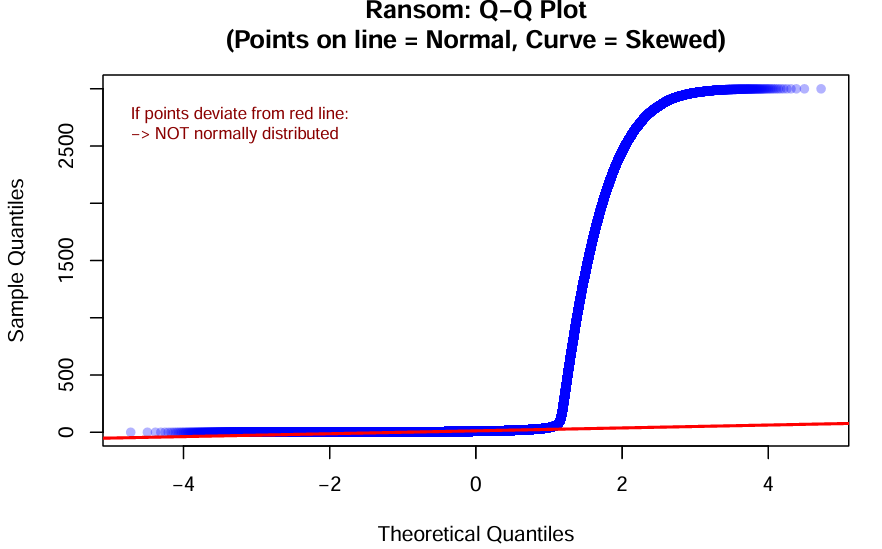


Figure 26 - Ransom Q-Q Plot

Figure 26 shows the plotting of the Q-Q plot on the Ransom variable. From the plot, the team sees the blue line drifts away from the red line midway which indicates there is skewness to the data and is not normally distributed.

##### DownTime

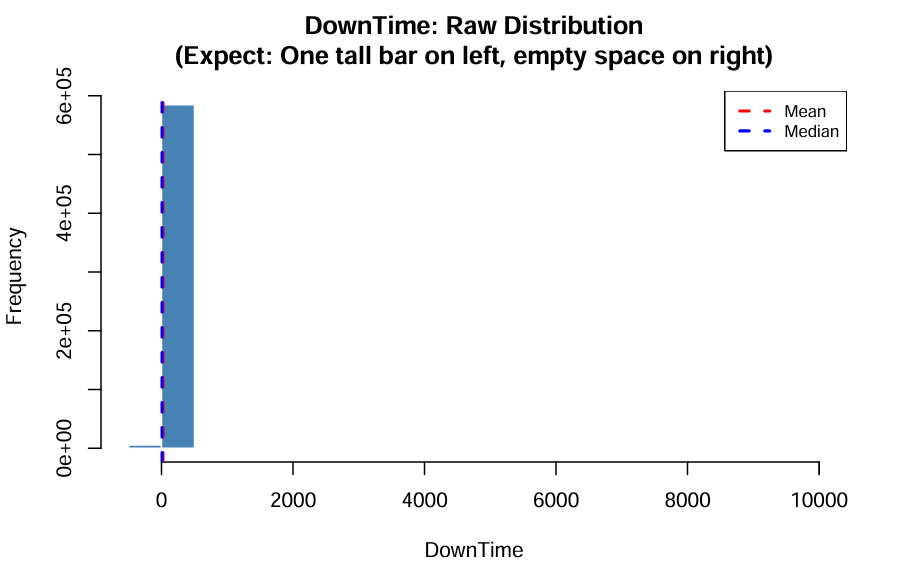


Figure 27 - DownTime Histogram

The plot shows that the DownTime variable is also extremely right skewed.

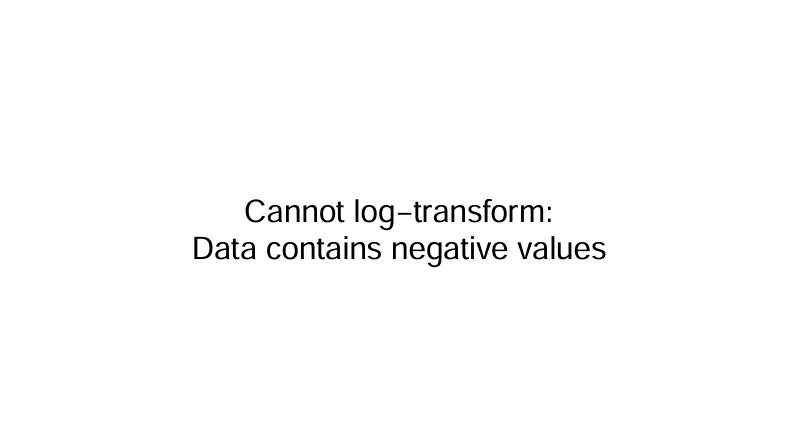


Figure 28 - DownTime Log-Transformed Histogram

The team couldn’t plot the Log-Transformed histogram for the DownTime variable as there are negative values within the data. This raises a critical warning to deal with the negative values in the cleaning steps ahead.

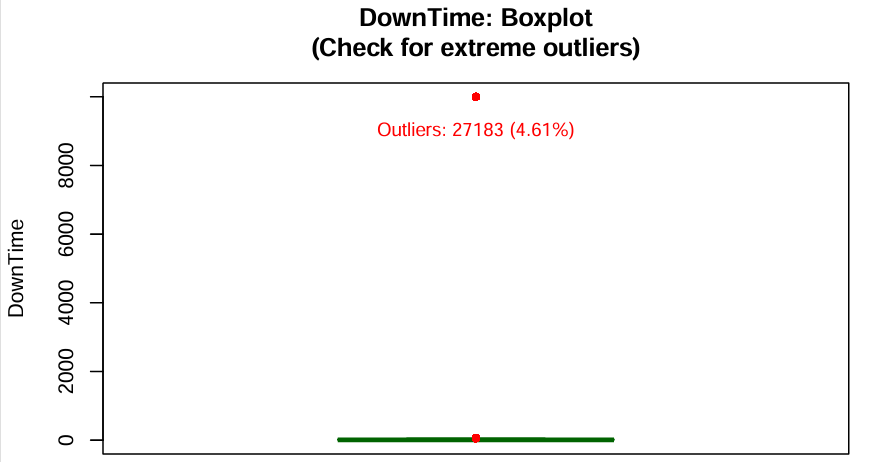


Figure 29 - DownTime Boxplot

Even though the plot shows there are far fewer outliers compared to the Ransom variable, but the outliers spread significantly away from the box which is a concern and requires cleaning.

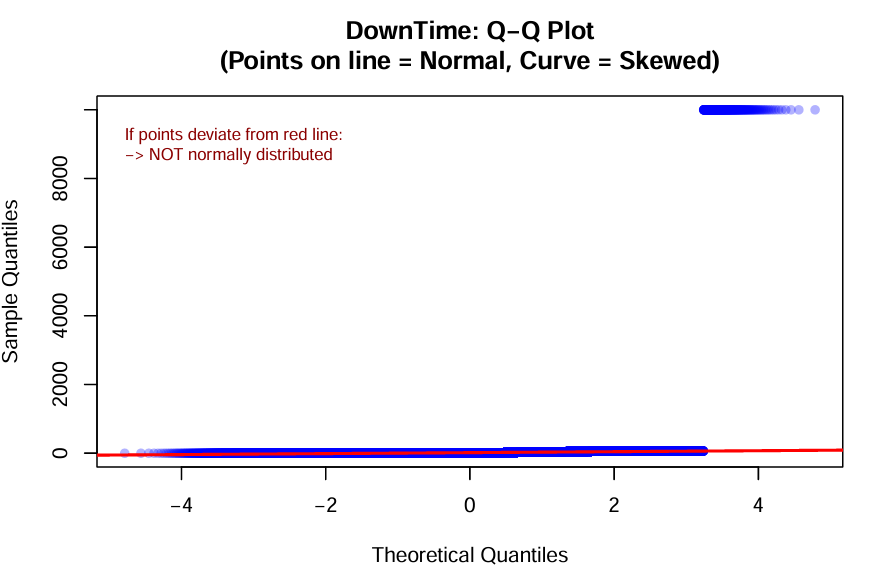


Figure 30 - DownTime Q-Q Plot

The blue line stays on the red line for most of the part but shows a split right at the end. This indicates that the data is not normally distributed.

##### Loss

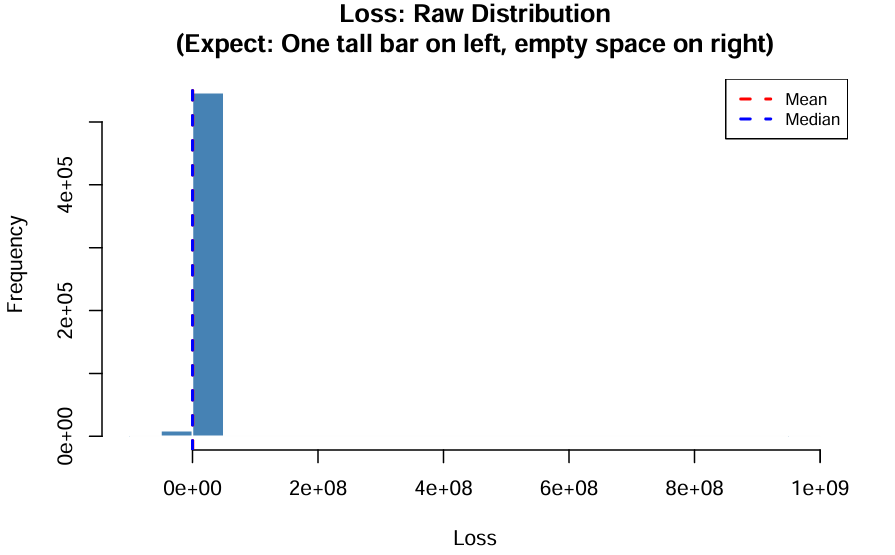


Figure 31 - Loss Histogram

Similar to the past two variables, the Loss variable shows an extreme skewness to the left.

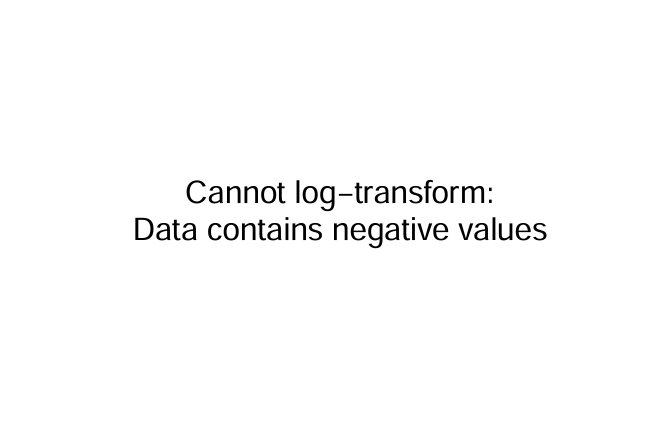


Figure 32 - Loss Log-Transformed Histogram

The Loss variable has the same issue as the DownTime variable, showing the existence of negative values within their data.

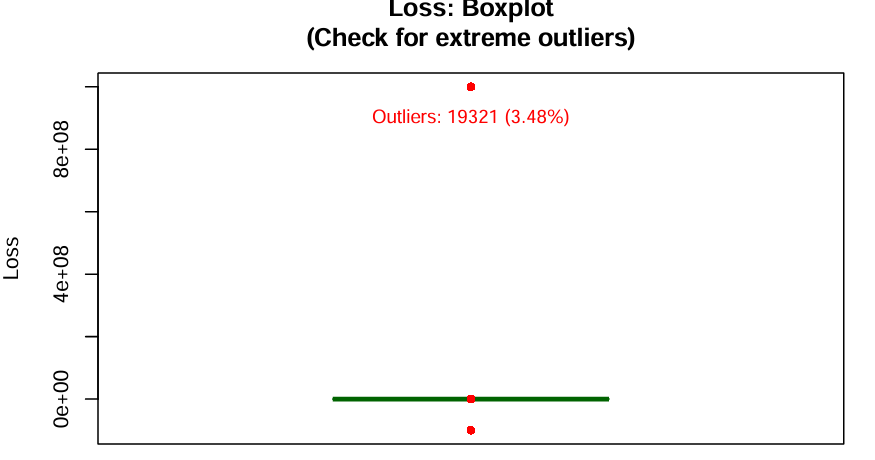


Figure 33 - Loss Boxplot

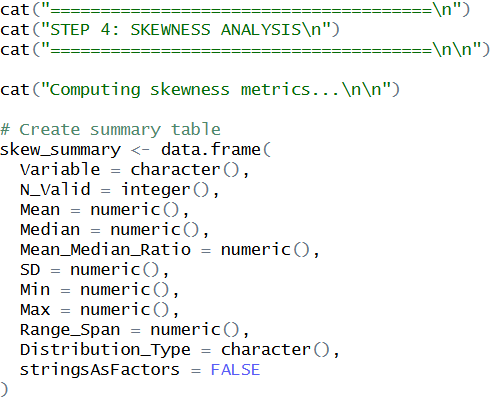
The Loss variable represents the least number of outliers occupying 3.48% of the entire data. But the magnitude of these outliers is massive which requires cleaning too.

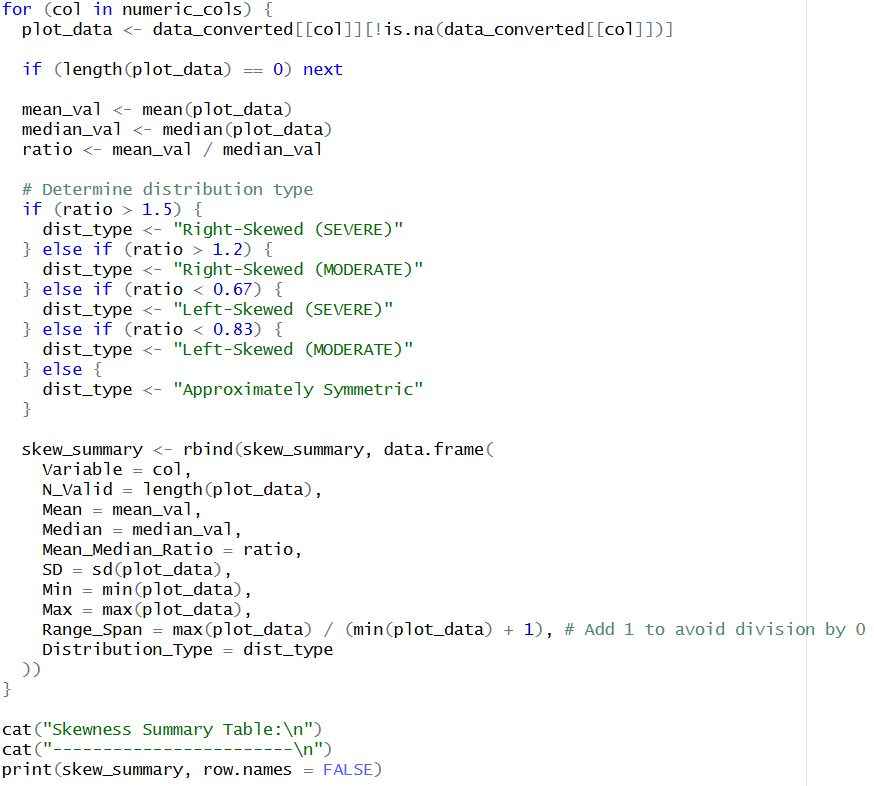


Figure 34 - Loss Q-Q Plot

From the plot, the team discovers the variable is not normally distributed either, as there are two splits of the blue line, one at the beginning and the other at the end.

#### Skewness Analysis



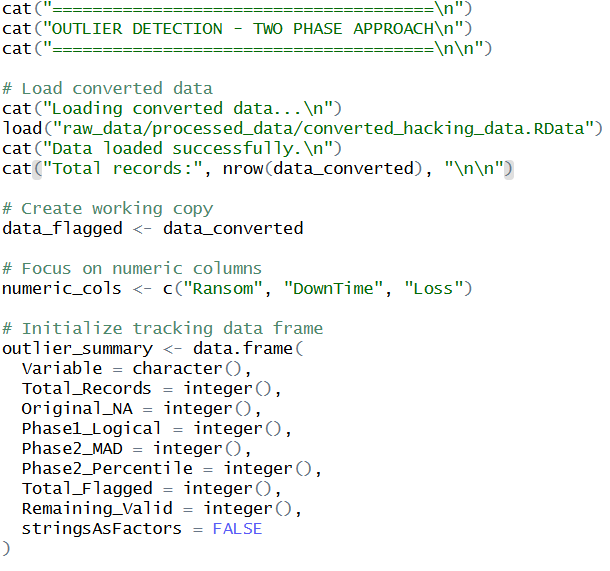


Afterwards, the team calculates and summarizes the skewness of each focused variable which later be used to analyze the shape of the data.

### Outlier Detection

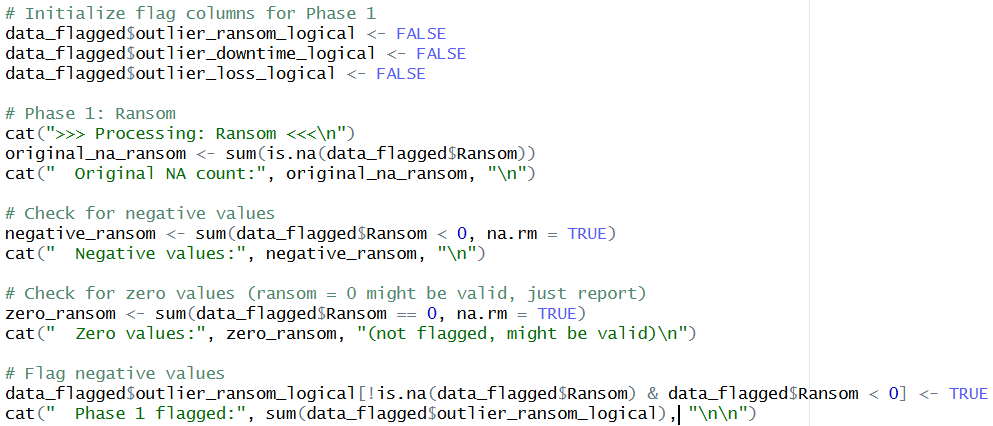
This step essentially detects and flags the outliers in two phases. The first focuses on physically impossible or logically invalid values like negatives and zeros data. Whereas the second phase uses statistical methods to detect extreme outliers, methods include modified Z-Score (MAD-based) and percentile method which flags values beyond 99.9th percentile. The modified Z-Score works by calculating the distance of observations from the median. The formal for the modified Z-Score:*.* (fawwazmts, 2024)

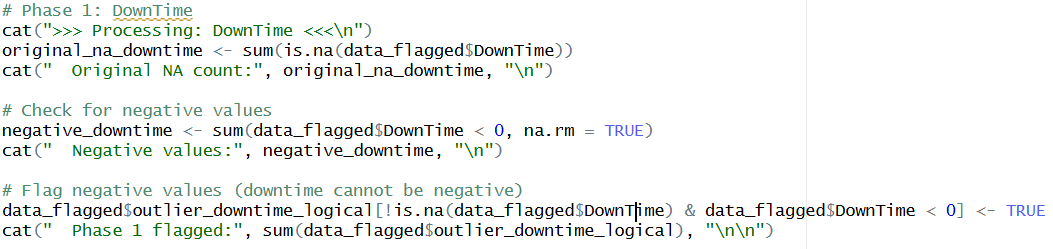
#### Load Dataset

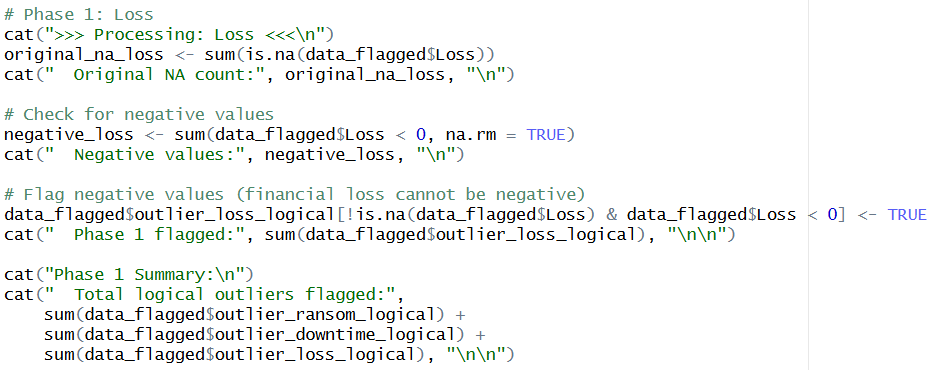


Start by loading in the dataset as usual.

#### Phase 1

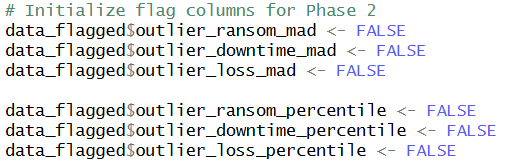




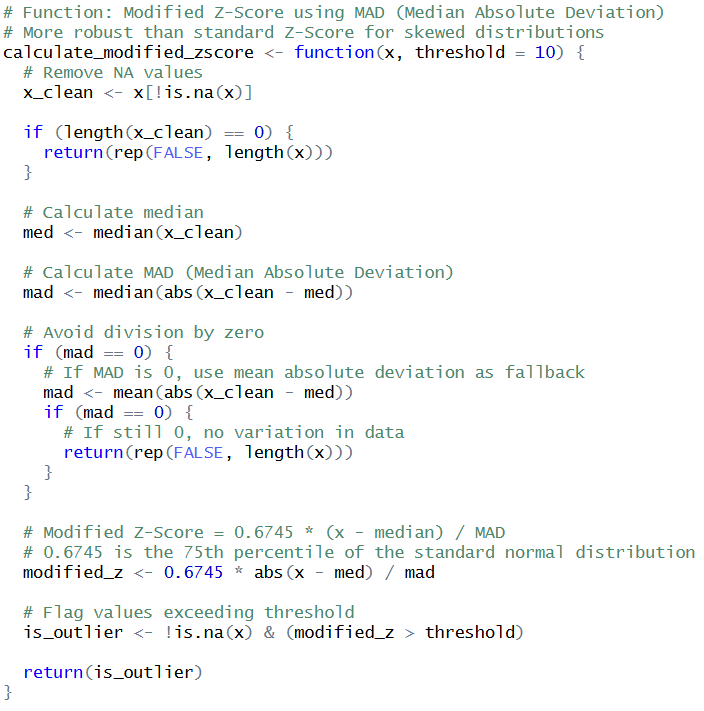


Phase 1 deals with the negative values within the Ransom, DownTime and Loss variables by detecting the negative numbers.

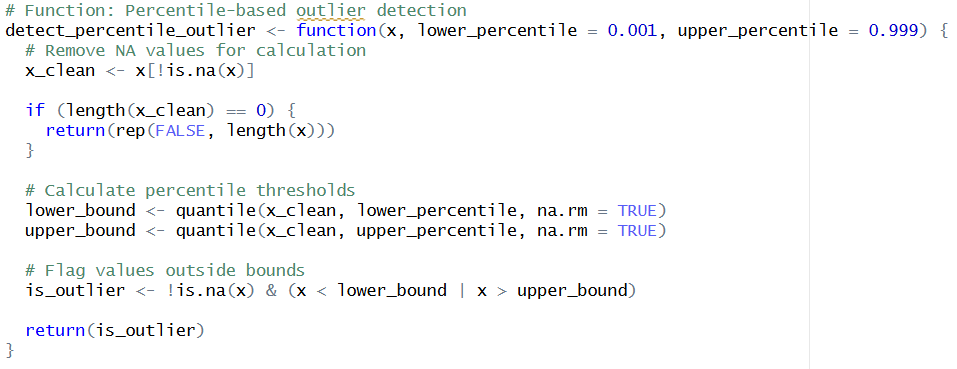
#### Phase 2



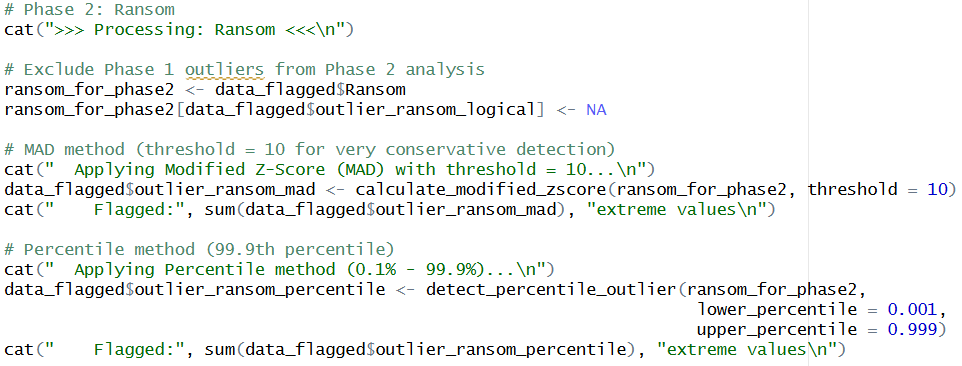
Initialization for the process later.

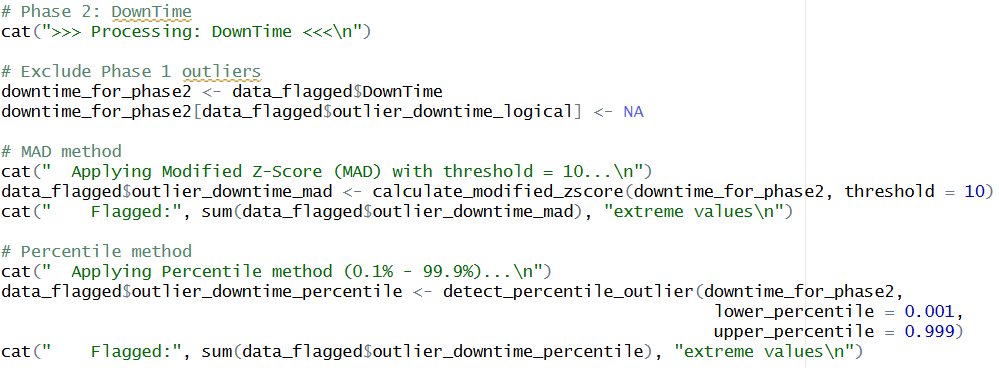


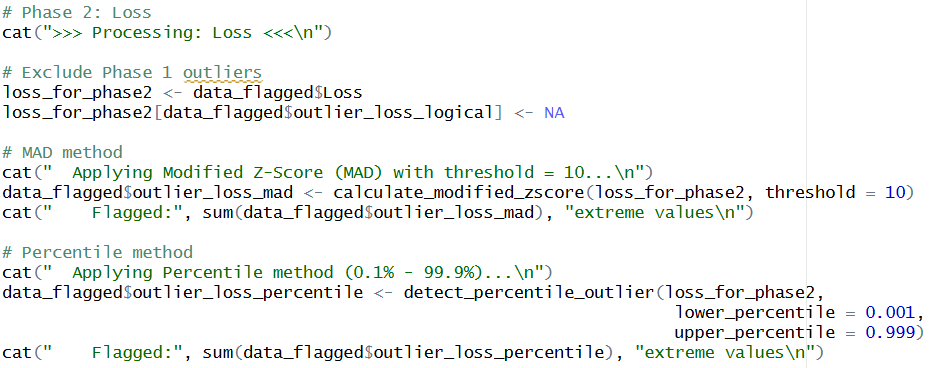
Declare a function for the modified Z-Score.



Declare a function for the percentile-based outlier detection.







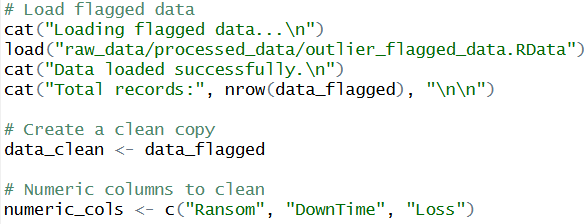
Apply both declared functions on the Ransom, DownTime and Loss variables.



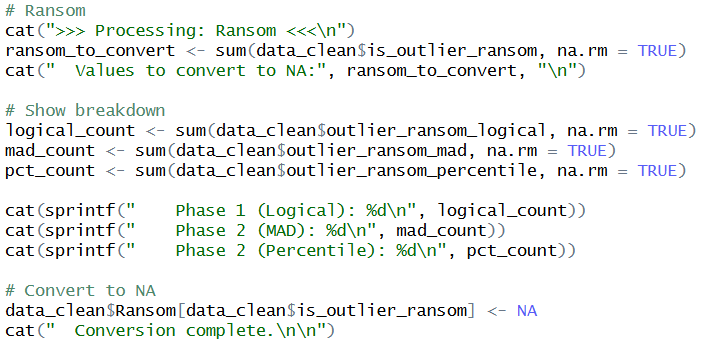
Create a new column that shows whether the data is an outlier.

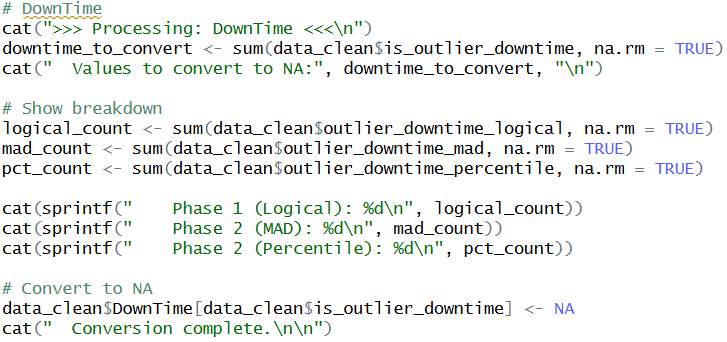
### Outlier to NA

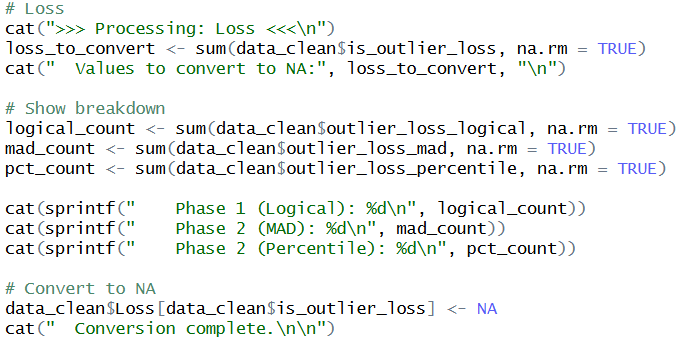
This step involves converting the flagged outliers earlier to NAs.



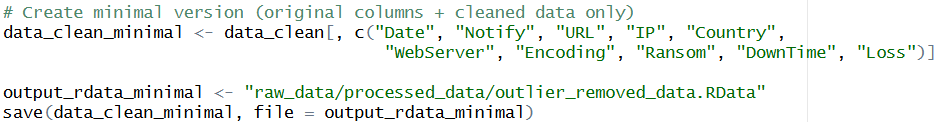
Load the data as always.







Convert the flagged outliers from all three variables to NAs.



Save the converted file in RData for better efficiency.

### Multivariate Outlier Detection

Purpose: Detect “weird combinations” using Isolation Forest

Method: Isolation Forest - detects structural anomalies in multivariate data

Sources to attached: what is Isolation Forest?

### Multivariate Outlier to NA

Purpose: Convert flagged multivariate outliers to NA, preserve the flags for later inspection.

Columns affected:

### Data Health Check

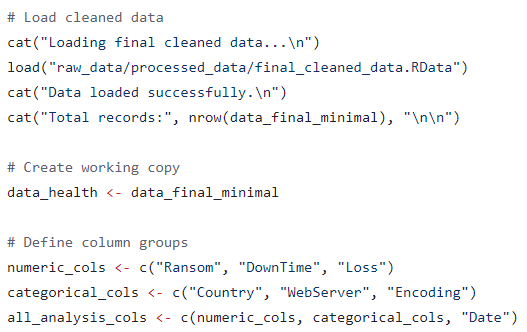
**Part 1: Setup and Preparation**

After converting all the flagged multivariate outliers into NA, the team performed a data health check on the whole dataset to visualize the missing data patterns and post-cleaning distribution quality.



The data health check process started with installing and loading the required packages. Specifically, the 5 packages installed in this step are: “**naniar**”, “**VIM**”, “**moments**”, “**ggplot2**”, and “**gridExtra**”.

1. **naniar:** This package makes the summarizing and missing values handling processes easier. According to (Tierney, 2024), this package provides principled, tidy ways to summarize, visualize, and manipulate the missing data. **naniar** has minimal deviations compared to the workflow in **ggplot2** and **tidy data**.
2. **VIM:** This package provides tools for visualization, imputation, and exploration of missing and multivariate data (Templ, n.d.).
3. **moments:** This package is used to calculates statistical shape like skewness and kurtosis (Komsta, 2022).
4. **ggplot2:** This package allows one to visualize the data using graph plotting (ggplot2, n.d.).
5. **gridExtra:** This package provides a number of user-level functions to work with “grid” graphics, especially in drawing tables and arranging multiple grid-based plots on a page (Auguie, 2017).

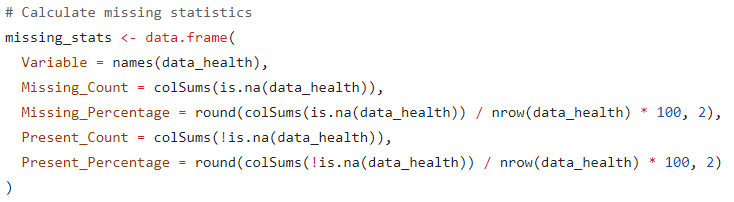


After loading the required packages, the process continued with loading the specific .RData file the group saved previously. Here, a copied version of the cleaned dataset named “**data\_health**” was created to prevent messing up with the original version. The whole data health check process will be working on the “**data\_health**” dataset. After that, the column names were being divided into numerical columns and categorical columns so that they can be easily referred to in the future.

**Part 2: Missing Data Diagnosis**

The first step to perform data health check is to create a statistics table. The following R code generated a new data frame called “**missing\_stats**” that calculates four metrics for every column:

1. **Missing\_Count:** Count of missing values (NA).
2. **Missing\_Percentage:** Percentage of missing values.
3. **Present\_Count:** Count of present values.
4. **Present\_Percentage:** Percentage of present values.



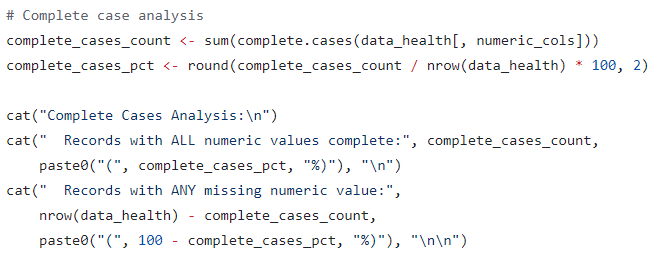
Then, the group sorted the table so that the columns with the most missing data will appear at the top. The purpose of this step is to simplify the data inspection.



Then, the sorted table is being displayed in the console without row numbers.



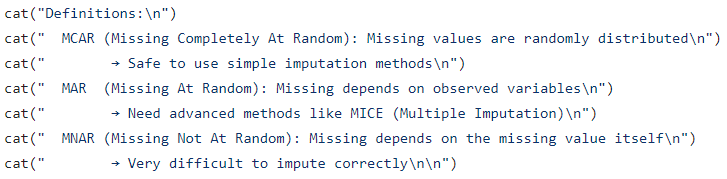
After that, the group counted how many rows have zero missing data and how many rows have at least one missing value. Following that, the group displayed the exact count and percentage of those perfect rows and incomplete rows. This approach gave the group a clear and deep understanding of the NA patterns in the dataset.



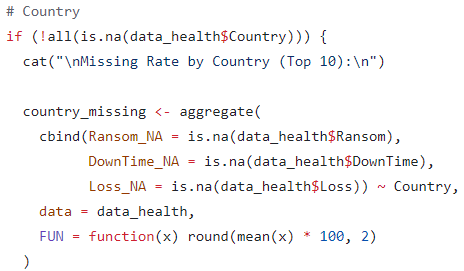
**Part 3: Missingness Mechanism Test**

This is the most complex statistical part in the data health check. Here, the group ran statistical tests to answer why the data was missing. The missing values were divided into three categories:

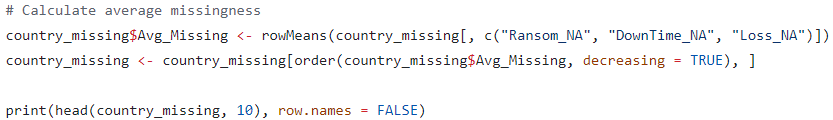
1. **Missing Completely at Random (MCAR):** The missing values are randomly distributed. The group will perform NA imputation using simple imputation methods for this category.
2. **Missing at Random (MAR):** The missing values all depends on the observed variables. The group will perform NA imputation using advanced methods like Multivariate Imputation by Chained Equations (MICE).
3. **Missing Not at Random (MAR):** The missing values all depends on the missing value itself.



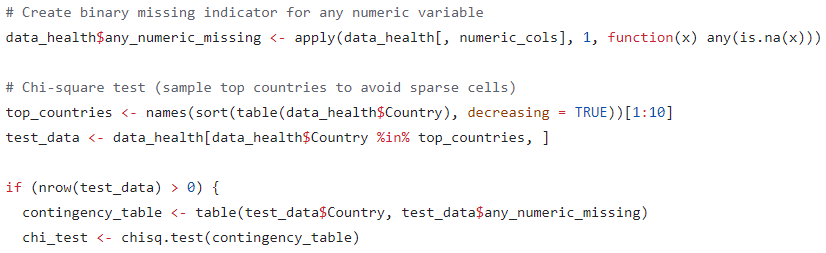
Starting with the categorical variables first, the grouped chose “**Country”** column as the starting point. The group created a summary table named “**country\_missing**” that calculated the average missing percentage for “**Ransom”**, “**DownTime**”, and “**Loss**” specifically for each country to examine whether specific countries have worse data quality than others.



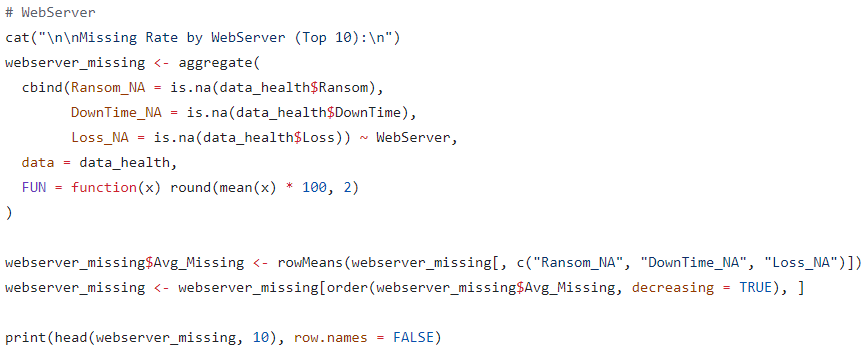
Cleaned up the table to make it readable by creating a single score for each country and sorting the countries by how bad their data quality was then displayed only the top 10 rows of it.



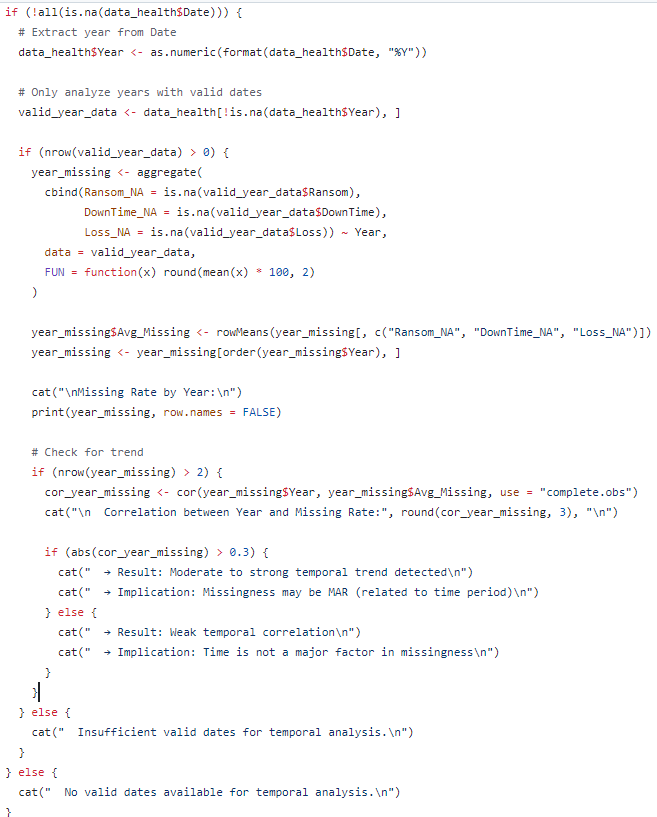
After that, the group use Mathematics to prove if the relationships was real or just coincidence. The **apply( )** function created a binary missing flag column for every row to convert complex missing data into a label. The, the “**top\_countries**” filter isolated only the 10 most frequent-exists countries to ensure the dataset is large enough for the statistical math to be reliable. Finally, **chisq.test( )** function calculated a p-value to scientifically prove whether the missing data was random (MCAR) or systematic (MAR) as mentioned above. This informed the group about which imputation method to be used.



Besides performing the missingness mechanism test on the “**Country**” column, the group also did the same approach to the “**WebServer**” column.



Not only that, but the group also tried to examine whether the data missingness was related to time.

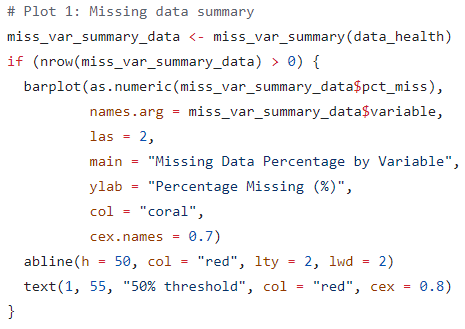


**Part 4: Missing Pattern Visualization**

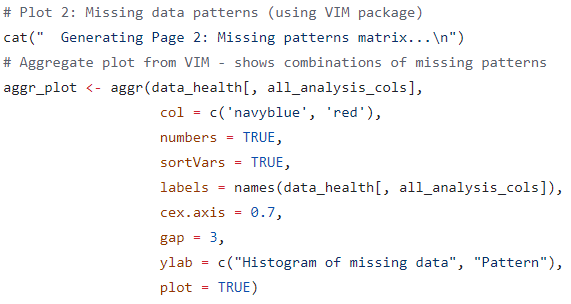
While looking at raw numbers is extremely challenging at times as it required highly-focus and is time-consuming, the team generated a PDF report named “**missing\_data\_analysis.pdf**”.



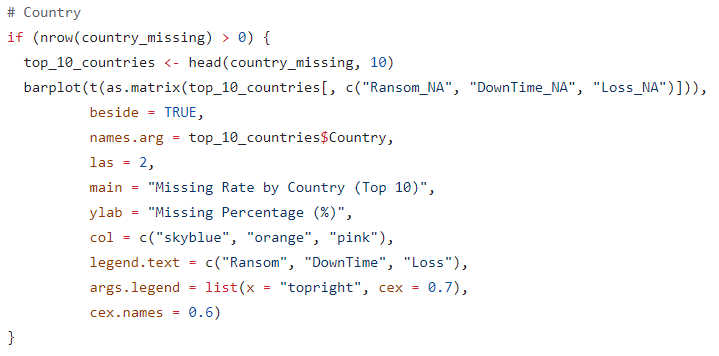
There were several graphs being inserted into the PDF. The first graph is the summary plot in a bar chart drawn using the **naniar** package.



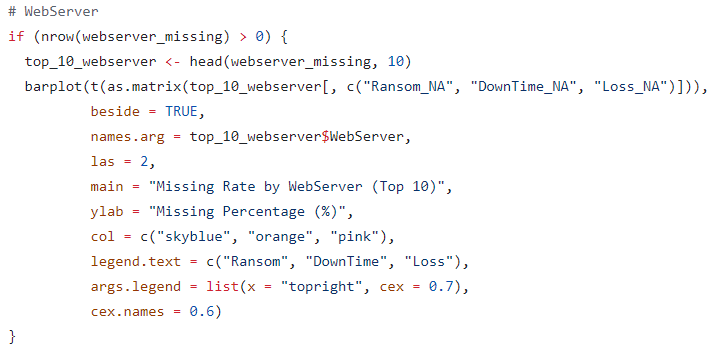
The second graph is the matrix plot that visualizes the combinations of missingness drawn using the **VIM** package.



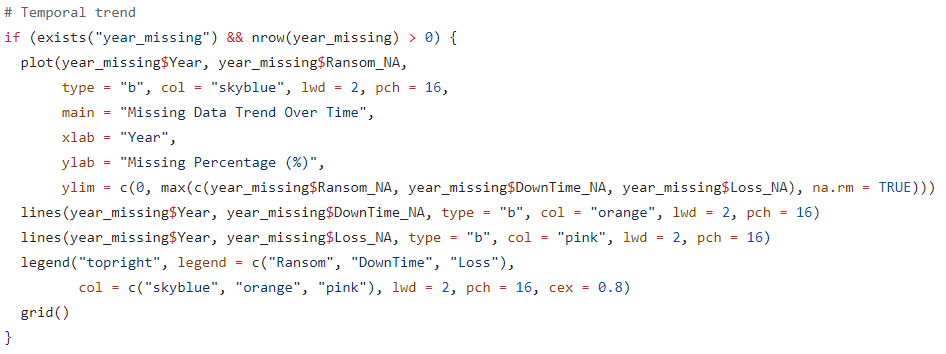
The third graph is the category plot that showed the missing rates for the top 10 countries visualized using a bar chart.



The fourth graph is the category plot that showed the missing rates for the top 10 webservers visualized using a bar chart.



The fifth graph is a line chart that showed the changing of missing data percentages from year to year.

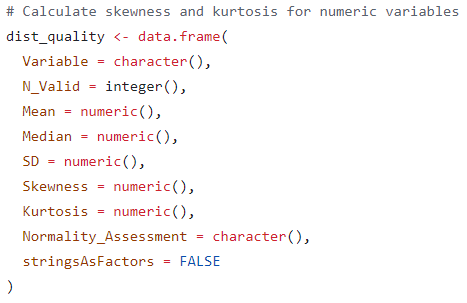


Finally, saved and closed the PDF file.

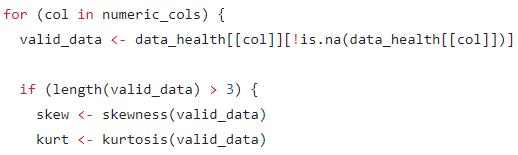


**Part 5: Post Cleaning Distribution**

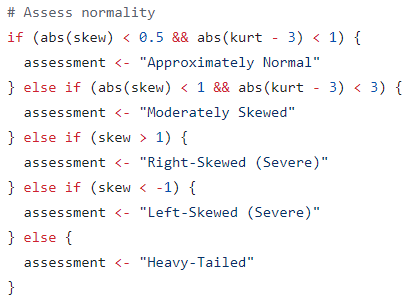
The purpose of this process is to assess the overall data quality after the outlier removal. Before filling in the missing values, it is compulsory for one to understand the shape of the existing data. This process started with creating an empty data frame named “**dist\_quality**” to store the results.



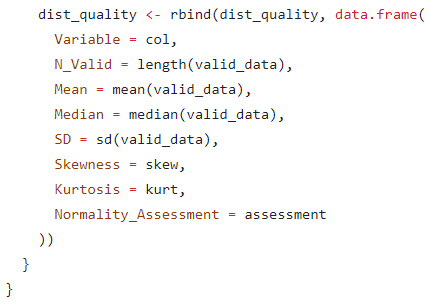
Processed the columns one by one. The flow of the columns to be processed were “**Ransom**” 🡪 “**DownTime**” 🡪 “**Loss**”. First, the group filtered out the empty rows to look only at valid data. Then, **skewness( )** calculates whether the data is leaning towards left or right, and **kurtosis( )** calculates whether the peak of the distribution is “sharp like a needle” or “flat like a plateau”.



Assigned a human-readable label to all the numbers based on the threshold values.



Wrote the math and the diagnosis into the table “**dist\_quality**” created.

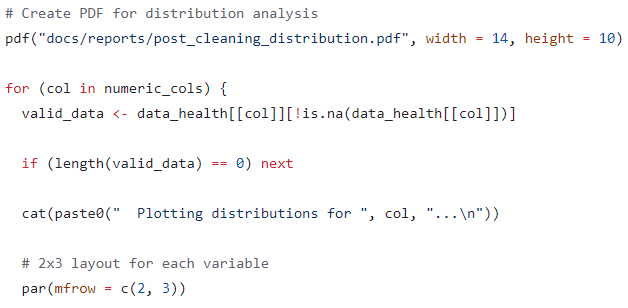


After that, displayed the full “**dist\_quality**” table in the console without the row numbers.



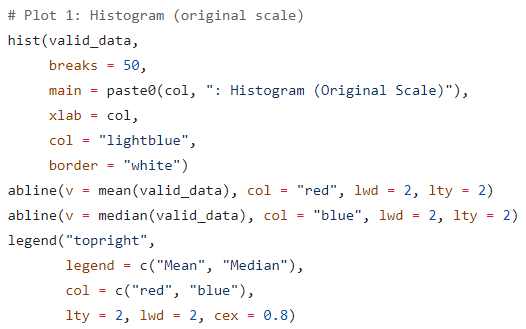
**Part 6: Distribution Visualization**

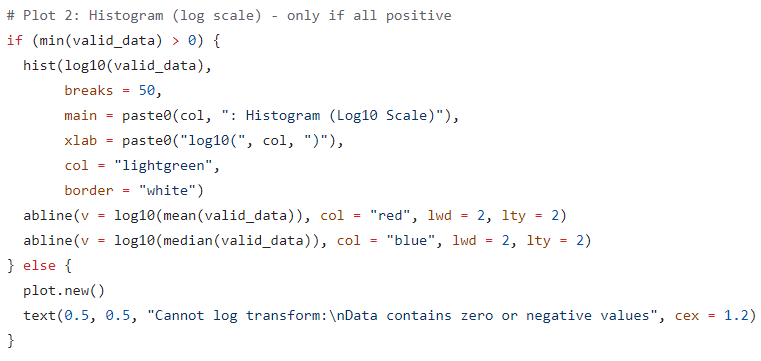
For a better distribution visualization experience, the group createda PDF file named “**post\_cleaning\_distribution.pdf**”. Inside the PDF file, every single page will store 6 distinct generated charts for each numeric column (Ransom, DownTime, and Loss).

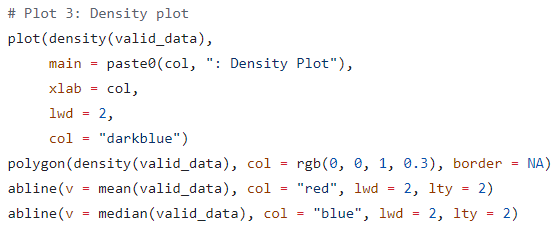


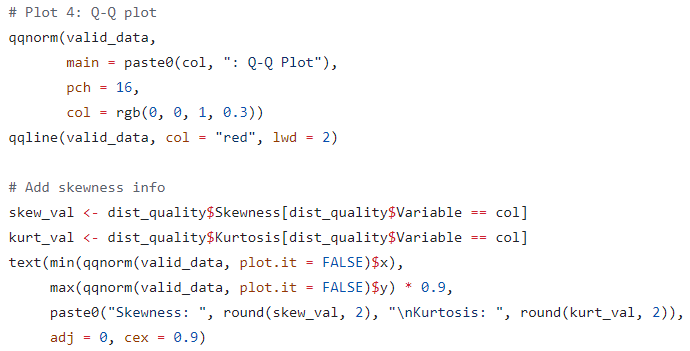
Specifically, the 6 types of charts are:

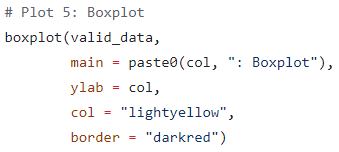
* 1. **Histogram (Original):** Shows the raw data shape.
  2. **Histogram (Log Scale):** Shows what the data looks like if we compress it (log10).
  3. **Density Plot:** A smooth curve of the distribution.
  4. **Q-Q Plot:** A technical chart to check if data fits a "Normal" distribution.
  5. **Boxplot:** Shows the median and range.
  6. **ECDF:** Shows the cumulative probability.

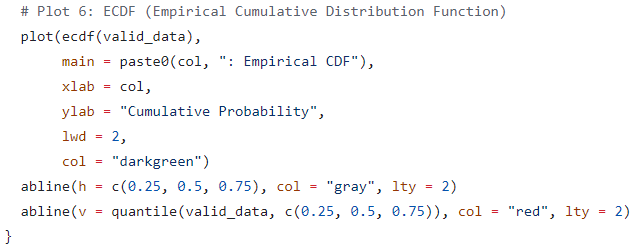












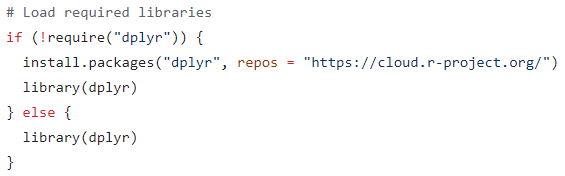
Finally, closed the PDF file that contained a total of 18 charts to save it using the command **dev.off( )**. Then, the data health check step can be considered complete, and it represents that the group is ready to continue with preprocessing for missing values imputation.



### Preprocessing for Missing Values Imputation – not yet completed

**Part 1: Setup and Data Loading**

This step prepares the data structure before filling the missing values. First, load the necessary libraries.



Then, load the specific .RData file.

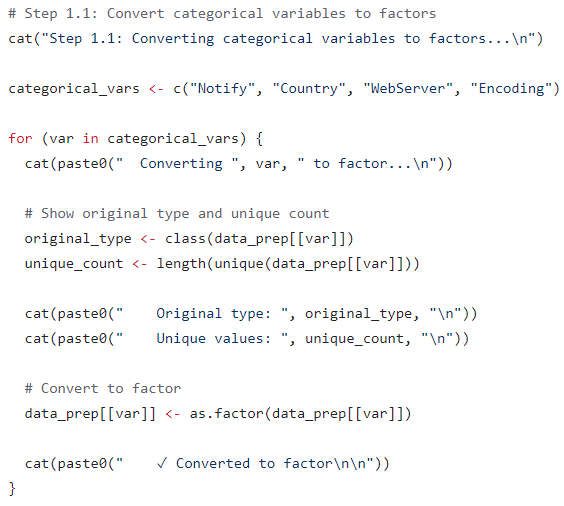


The group created a copied version of the dataset named “**data\_prep**” to prevent messing up with the original version. The whole missing values imputation preprocessing process will be working on the “**data\_prep**” dataset.

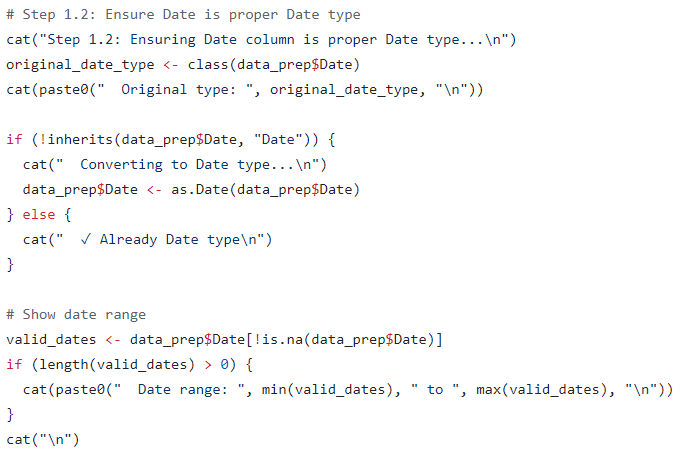


**Part 2: Variable Categorization**

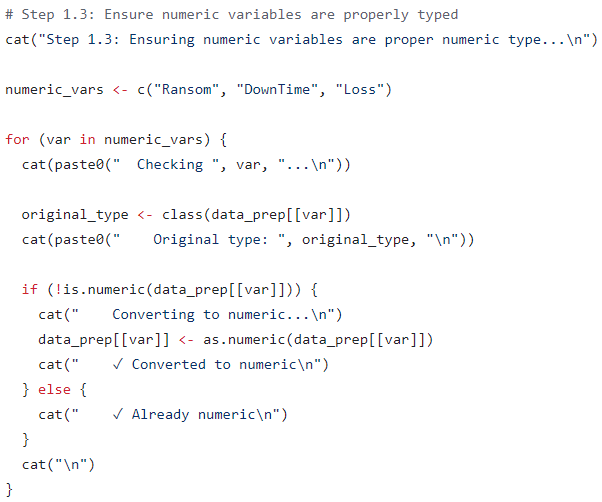
This process converted the categorical columns into Factors. A Factor in R is a special way of storing categorical data. In R, Factors are stored as integers with corresponding labels (GeeksforGeeks, n.d.). This is essentially significant for statistical models to understand groups. The chosen categorical columns to be converted to Factors are “**Notify**”, “**Country**”, “**WebServer**”, and “**Encoding**”



After Factors conversion, the group proceeded to ensure that the datatype of “**Date**” column is Date instead of any other data type.



While “**Ransom**”, “**DownTime**”, and “**Loss**” columns are supposed to be numerical data type, the group converted them into numbers.

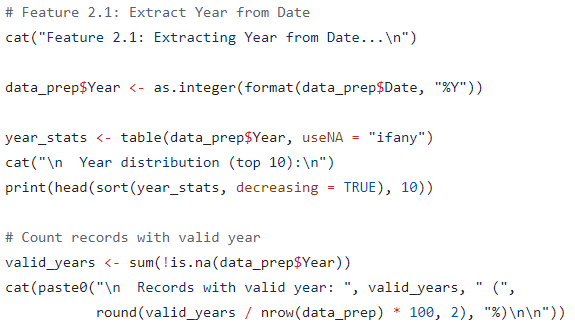


Finally, displayed the summary after the categorization for better understanding of the current dataset.

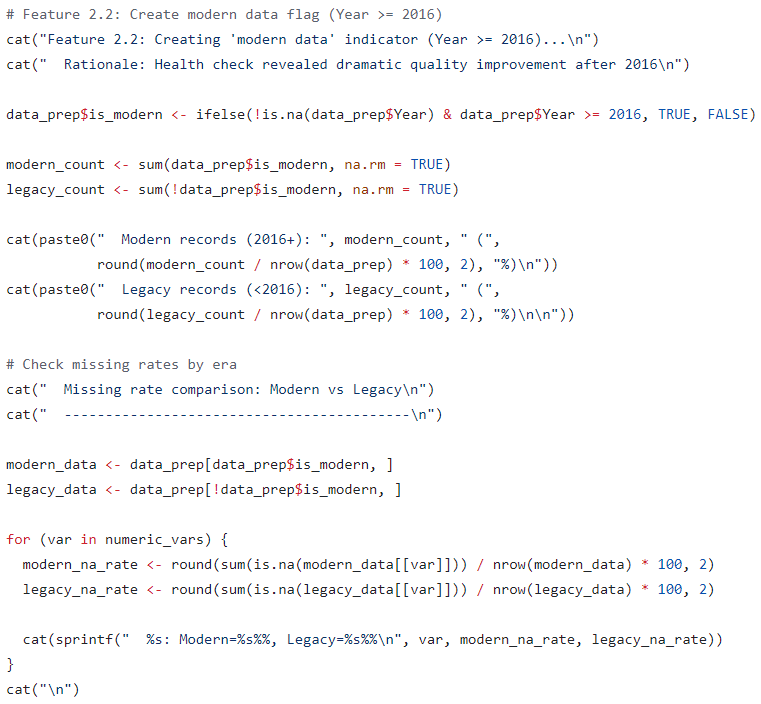


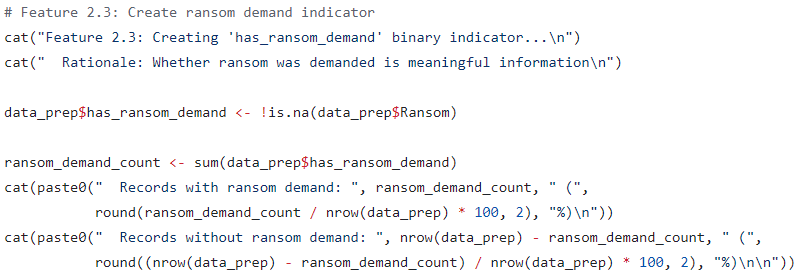
**Part 3: Feature Engineering**

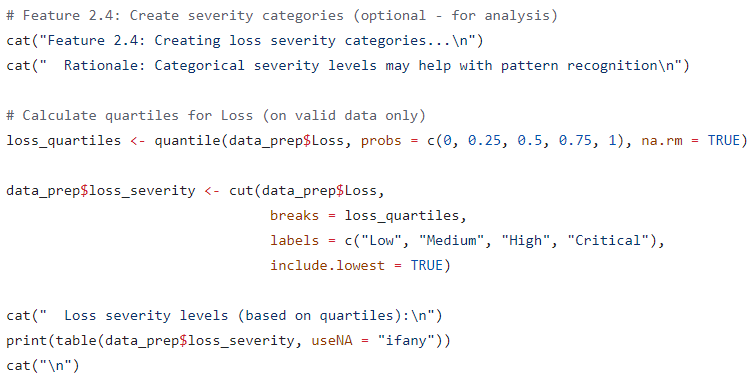
After converting all the columns into appropriate datatype, the group performed feature engineering on the dataset to create useful derived variables. Starting with the time feature, the group extracted the year out from the date.

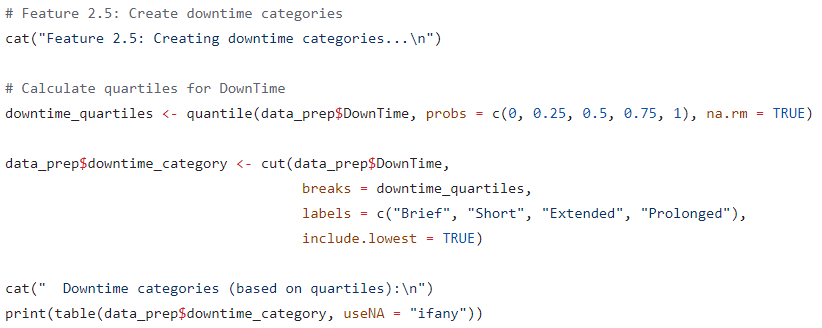


After that, the group created a flag called “**is\_modern**”. Here, all rows that have year earlier than 2016 will be labelled as “**TRUE**”, and vice versa. This is because the grouped assumed the data after 2016 has a higher quality. Refer to section 1.4.x for more details.



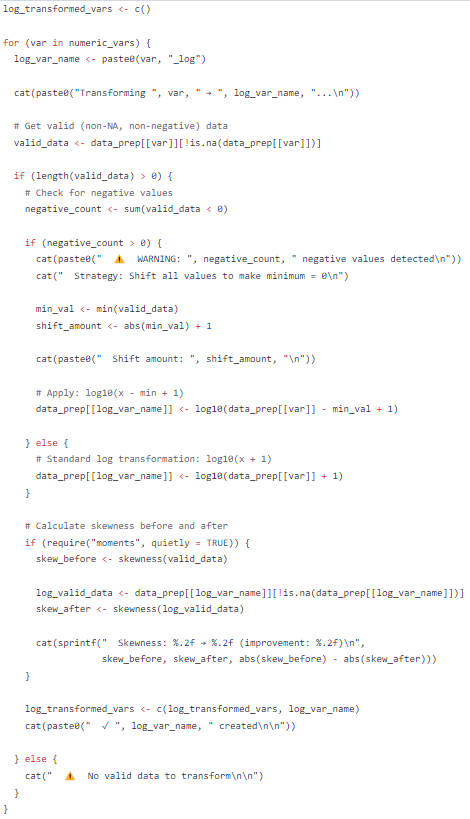






**Part 4: Log Transformation**

Following that, the group performed log transformation to normalize the skewed distributions. This was due to the fact that most ransoms were small, but a few were massive, which confused the imputation algorithms. In order to make the data looked like a normal Bell curve and the final filling of missing values much more accurate, the group applied a log transformation to squash the massive numbers down.



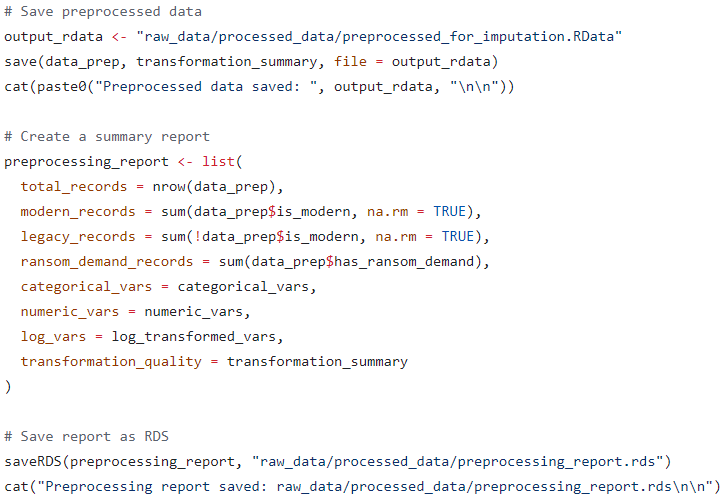
**Part 5: Transformation Quality Check**

After transforming the data, the group performed transformation quality check to compare the distributions before and after the log transformation. In this step, the group mainly observed whether the skewness drops from 2.0 (bad) to 0.2 (good). If it drops, the transformation was then labelled as “Excellent”.



**Part 6: Data Saving**

Finally, the group saved the cleaned, categorized, and transformed dataset into a new .RData file named “**preprocessed\_for\_imputation.RData**” and all the statistics that were useful for generating reports in the future without needing to load the huge dataset into a small summary file named “**preprocessing\_report.rds**”.



### Categorical Cleaning

This step focuses on cleaning and standardizing high-cardinality categorical variables. The columns affected in this step will be “**WebServer**”, “**Country”**, “**Notify**”, “**URL**”, “**IP**”, and “**Encoding**”.

**Part 1: Setup and Data Loading**

The first step in categorical cleaning required importing the necessary libraries.



Then, load the specific .RData file.

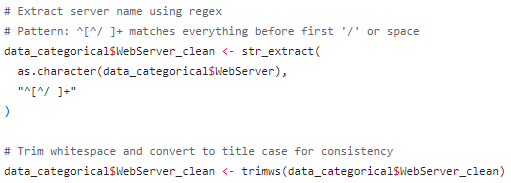


The group created a copied version of the cleaned dataset named “**data\_categorical**” to prevent messing up with the original version. The whole categorical cleaning process will be working on the “**data\_categorical**” dataset.

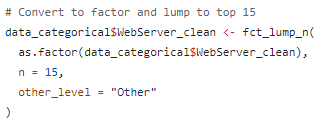


**Part 2: “WebServer” Column Cleaning**

The group have chosen the “**WebServer**” column as the starting point. The first step into cleaning the “**WebServer**” column was extracting the server’s name using regular expressions (regex) and trimming the whitespace.



The “**WebServer**” column was continued to be converted into Factors. The group only kept the top 15 most common server types.

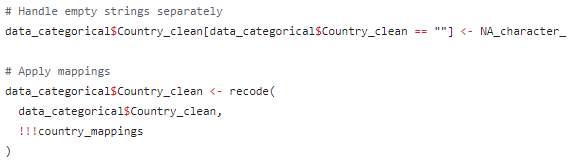


**Part 3: “Country” Column Cleaning**

The second chosen column was “**Counry**”. The first step into cleaning the “**Country**” column was converting them into uppercase.

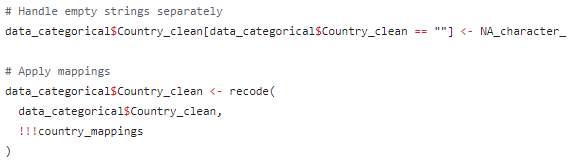


The second step was to change all the empty string into NA value.

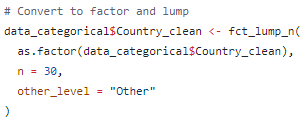


The following step was to standardize the “**Country**” column using a mapping list



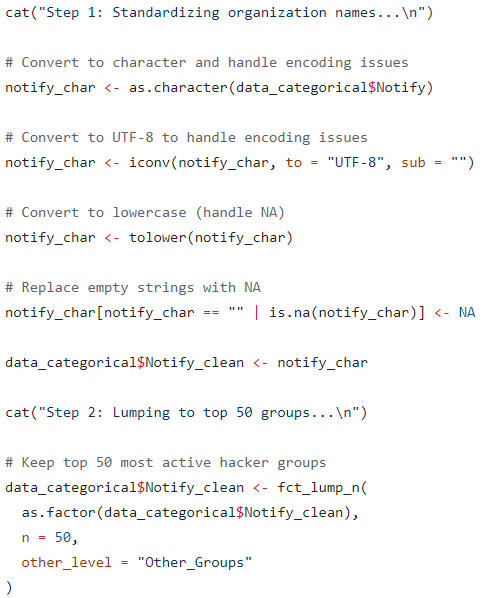


Finally, similar to the “**WebServer**” column, but the group only kept the top 30 most common countries appeared in the dataset.



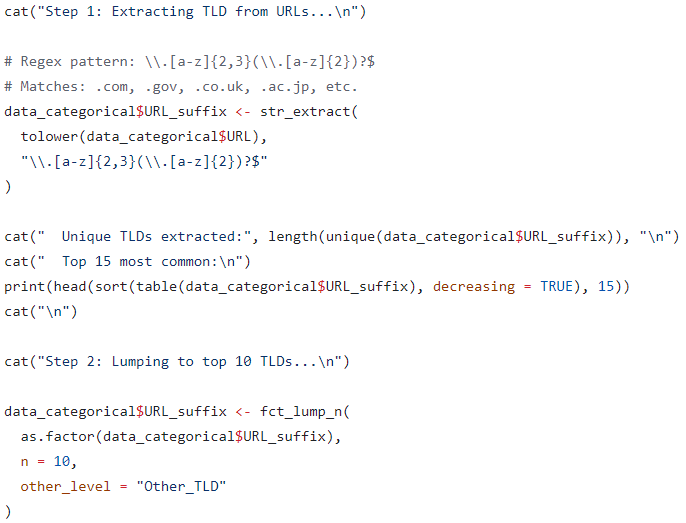
**Part 4: “Notify” Column Cleaning**

The third chosen column was the “**Notify**” column. The group converted the “**Notify**”column into character string, then converted them into UTF-8 encoding format. Following that, the group continued to convert the column into lowercase and replaced all the empty string with NA value. The group kept the top 50 most common active hacker groups only.



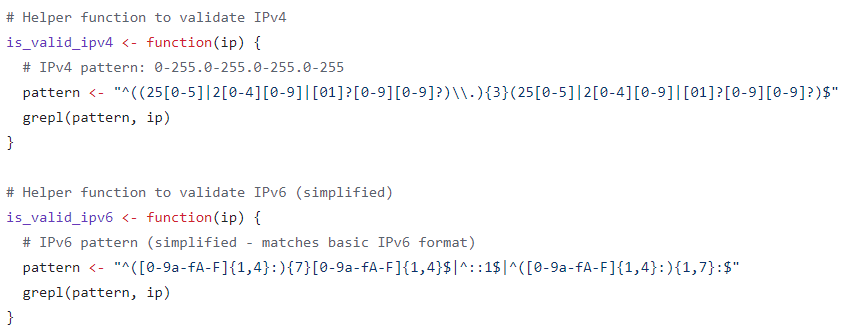
**Part 5: “URL” Column Cleaning**

The fourth chosen column was the “**URL**” column. The group extracted the Top-Level Domain (TLD) from the URL and only kept the top 10 most common being attacked websites.



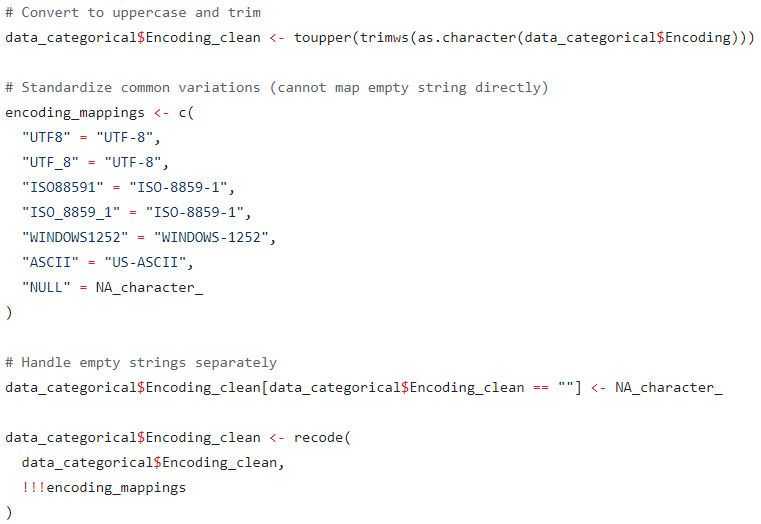
**Part 6: “IP” Column Cleaning**

For the “**IP**” address column, the group created two functions two filter out those data that did not looks like a valid IPv4 or IPv6 address.

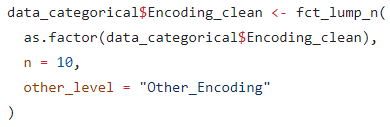


**Part 7: “Encoding” Column Cleaning**

For the “**Encoding**” column, the group converted them all into uppercase and timed the whitespaces, standardized them into formal encoding format, and changed the empty strings into NA values.

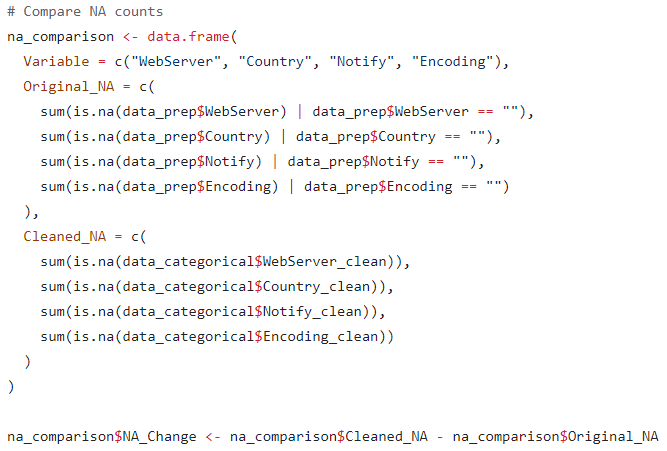


Similar to the “**URL**” column, the group only kept the top 10 most encoding patterns for the “**Encoding**” column.



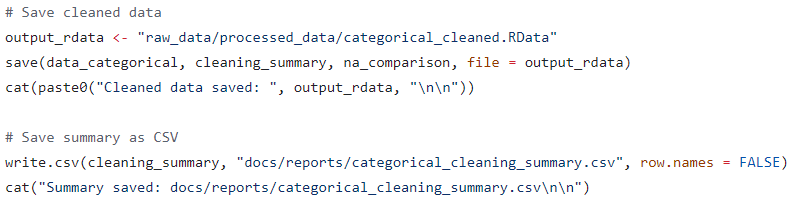
**Part 8: Post-Cleaning Comparison**

After cleaning all the categorical columns, the group made a comparison between the total amount of NA values before and after the categorical cleaning.



**Part 9: Data Saving**

Finally, the group saved the cleaned dataset into a new .RData file named “**categorical\_cleaned.RData**” and all the summary of categorical columns cleaning into a small summary file named “**categorical\_cleaning\_summary.csv**”.



### Final Quality Check

This step aimed to perform a final check across the whole dataset before it moved to the NA imputation. In this step, the group wanted to ensure that the data cleaning in all the previous steps did not destroy the information and quality of the dataset by accident to maintain the variables useful for more accurate predictions. Here, three specific checks were ran, including the information content analysis, multivariate associations analysis, and final missing patterns analysis.

**Part 1: Setup and Data Loading**

The first step in final quality checking required importing the necessary libraries.



Then, load the specific .RData file.



The group created a copied version of the cleaned dataset named “**data\_final\_check**” to prevent messing up with the original version. The whole data quality check process will be working on the “**data\_final\_check**” dataset.



**Part 2: Information Content Analysis**

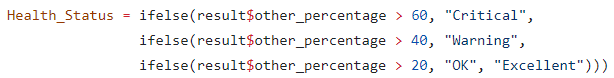
In this part, the group mainly checked the percentage of the value “**Other**” in the dataset. This is because the variable will become useless to predict anything if most of the data were labelled as “**Other**”. The group created a custom function named “**calculate\_other\_percentage**”to automate the mathematics and reduce redundant code.



The categorical columns to be analyzed here were “**WebServer**”, “**Country**”, “**Notify**”, “**URL**”, and “**Encoding**”.



The thresholds being set for the health status were 20, 40, and 60.

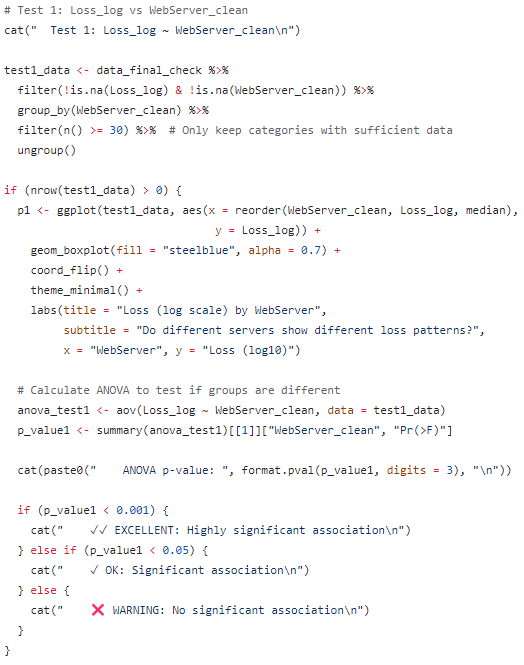


**Part 3: Multivariate Association Analysis**

This part of step examined whether the predictors (“**WebServer**”, “**Country**”, “**Notify**”, and “**URL**”) were correlated to the missing values in the columns the group wanted to fill (“**Loss**”, “**Ransom**”, and “**DownTime**”). The methods being used in this multivariate association analysis were Analysis of Variance (ANOVA) and Boxplots. The group divided the columns into four pairs of associations. For that, the group created a PDF file to store all of the association analysis graphs.



The first pair of association were the “**Loss**” and “**WebServer**” columns.



The second pair of association were the “**Loss**” and “**Country**” columns.



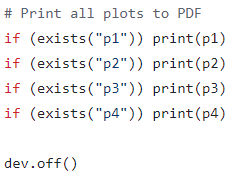
The third pair of association were the “**Ransom**” and “**Notify**” columns.



The fourth pair of association were the “**DownTime**” and “**URL**” columns.

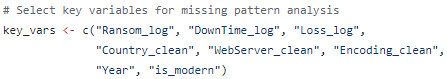


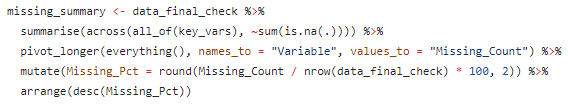
Stored all the graphs into the PDF file and saved the file.



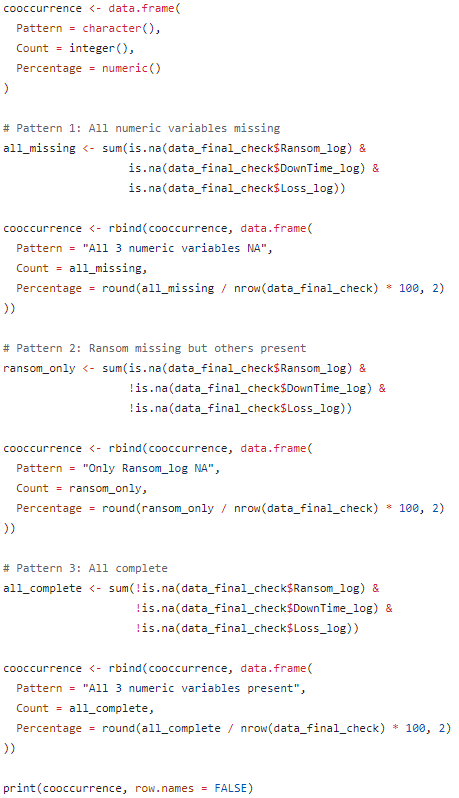
**Part 4: Final Missing Pattern Analysis**

While all the variables are now cleaned and transformed into log scale, this part of step aimed to re-evaluate how much data were missing and where they were. The group calculated the exact percentage of the missing values for every key variable, which included both the log-transformed data and cleaned categories.

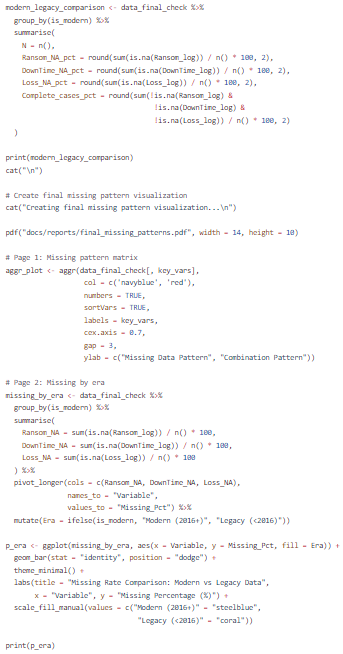




The group also performed a multivariate co-occurrence analysis to investigate the structural dependencies between the missing values. The group aimed to quantify the frequency of the three distinct data states, which were the complete cases, partial cases, and total loss cases, in order to validate the feasibility of multivariate imputation.

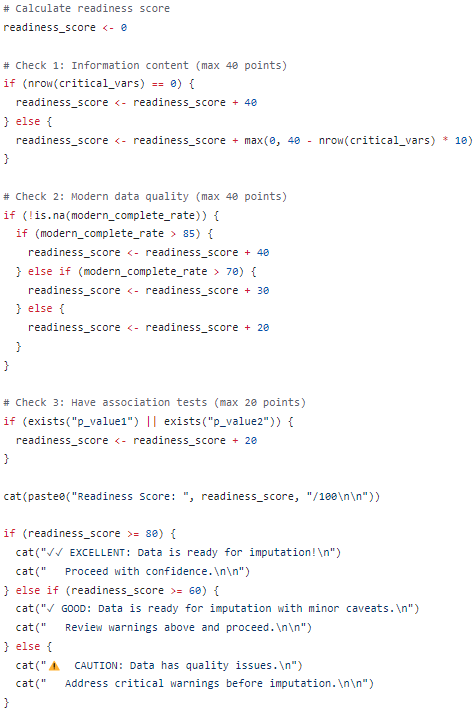


Backed by the assumptions made in section 1.4.x, the group separated the data into two categories: “modern” and “legacy”. The group conducted a stratified assessment to compare the NA in the “modern” records (post – 2016) and “legacy” records (pre – 2016).



**Part 5: Readiness Score**

Finally, the group calculated whether the dataset was ready for NA imputation based on the readiness score assigned to the categorical variables, the modern data, and the predictors.



### 1.3.x MICE Imputation

Purpose: Impute missing values using Multivariate Imputation by Chained Equations

Strategy: Separate imputation for Modern (2016+) and Legacy (<2016) data

Method: PMM (Predictive Mean Matching) - robust to skewness

**Part 1: Setup and Data Loading**

The first step in final quality checking required importing the necessary libraries.



Then, load the specific .RData file.



The group created a copied version of the cleaned dataset named “**data\_imputation**” to prevent messing up with the original version. The whole MICE imputation process will be working on the “**data\_imputation**” dataset.



**Part 2: Modern Data Imputation**

**Part 3: Legacy Data Imputation**

**Part 4: Combine Result**

**Part 5: Save Results**

### 1.3.x Imputation Validation

Purpose: Visual validation and spot checks of MICE imputation quality

**Part 1: Setup and Data Loading**

The first step in final quality checking required importing the necessary libraries.



Then, load the specific .RData file.



The group created a copied version of the cleaned dataset named “**data\_imputation**” to prevent messing up with the original version. The whole MICE imputation process will be working on the “**data\_imputation**” dataset.



Ransom distribution comparison

Downtime distribution comparison

Loss distribution comparison

Combined comparison

**Part x: Extreme Value Check**

**Part x: Save File**

**14. Visual Comparison – 2b confirmed**

## Assumptions

Short opening: …Several assumptions were made…

### no remove duplicate

While the dataset contains duplicate records, we have opted to retain them rather than performing de-duplication. These entries likely represent distinct security incidents occurring in close succession. Given that the 'time' field lacks the precision (e.g., milliseconds) to differentiate between them, removing these rows would lead to an underestimation of the actual attack frequency and total downtime.

**Note**: “Date” section only mention the date, it didn’t mention the time (hours, minutes, seconds, milliseconds), it is possible that the attackers held the attack multiple times in a day. The attackers held the attacks at the same location, using the same servers, and that is possible to cause the datasets have duplicate values in Downtime and Loss.

### no pay ransom

NA in Ransom is set to 0, because victims decide not to pay ransom

To attach source: why should not pay for ransom.

### 1.4.3 Assumption 3

assumptions --> encoding/webserver leave as it is. Too much weird values. Clean only whenever use this.

Assumption 4

Assumption 5

Assumptions 6

No negative value, all numerical values are “absoluted” to positive values.

# 2.0 Analysis (Individual)

Short introduction here

Each member: research objectives, analysis questions, data preparation, data analysis, and hypothesis formulation and testing.

## 2.1 Choong Ti Huai (TP078539)

text

### 2.1.1 Objectives

The objectives for this project are:

1. To analyze the relationship between the **Country** of the hosting server and the resulting **Ransom** amount paid following a cyber-attack to identify geographical regions that yield the highest financial demands.
2. To investigate the impact of the **Date** (temporal trends) on the resulting financial **Loss** caused by the attack to determine if specific periods or seasons are associated with higher financial damages.

### 2.1.2 Analysis Questions

RO1:

1. Which top 10 countries are responsible for the highest total Ransom payments?

2. Is there an explicit difference in the average Ransom amount paid across different Geographical Countries?

RO2:

1. How does the total Loss change from month to month or year to year?

2. Which specific months or years show the highest financial Loss?

### 2.1.3 Data Preparation

Describe any additional preparation work you did to perform or enhance your individual analysis.

Include explanations for each step and justify your approach.

Provide R code snippets with explanations and outputs for each step taken.

### 2.1.4 Data Analysis

Conduct exploratory and inferential analyses relevant to your objectives.

Include summary statistics and visualizations with interpretations.

Clearly indicate which objective each analysis supports.

Include explanations for each step and justify your approach.

Provide R code snippets with explanations and outputs for each step taken.

### 2.1.5 Hypothesis Formulation and Testing

Formulate hypotheses based on your findings and perform appropriate statistical tests in R.

Interpret and explain your test results comprehensively.

Include explanations for each step and justify your approach.

Provide R code snippets with explanations and outputs for each step taken.

## 2.2 Daniel Chan Zit Fung (TP079018)

### Objectives

The objectives of this project were:

1. To investigate which **WebServer** and specific **Encoding** method suffer the most **Loss** during cyber-attacks.
2. To analyze the relationship between the **Country** of the server and the resulting **DownTime** following a cyber-attack.

### Analysis Questions

To investigate which **WebServer** and specific **Encoding** method suffer the most **Loss** during cyber-attacks.

* What are the top 10 WebServer and Encoding methods suffer the most Loss during cyber-attack?
* Within each WebServer category, which specific Encoding method is associated with the highest mean Loss?

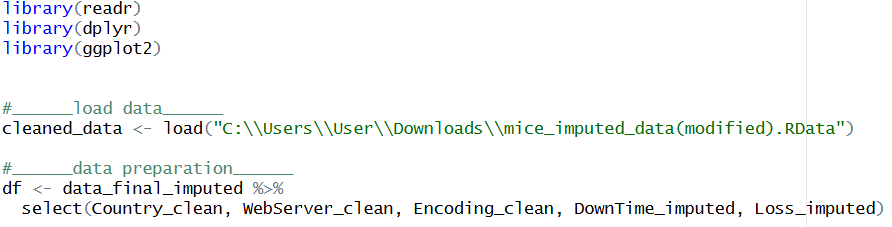
To analyze the relationship between the **Country** of the server and the resulting **DownTime** following a cyber-attack.

* Is there a significant difference in median DownTime between servers hosted in 'developed' Country versus 'developing' Country?

Note that the countries will be categorized into two groups, Developed and Developing, according to the United Nation published material. (Country classification)

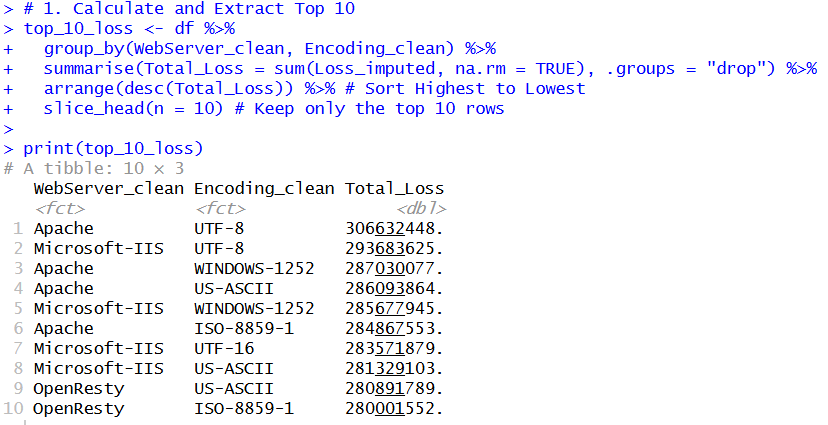
* What are the top countries experiencing the longest mean DownTime following a cyber-attack?

### Data Preparation

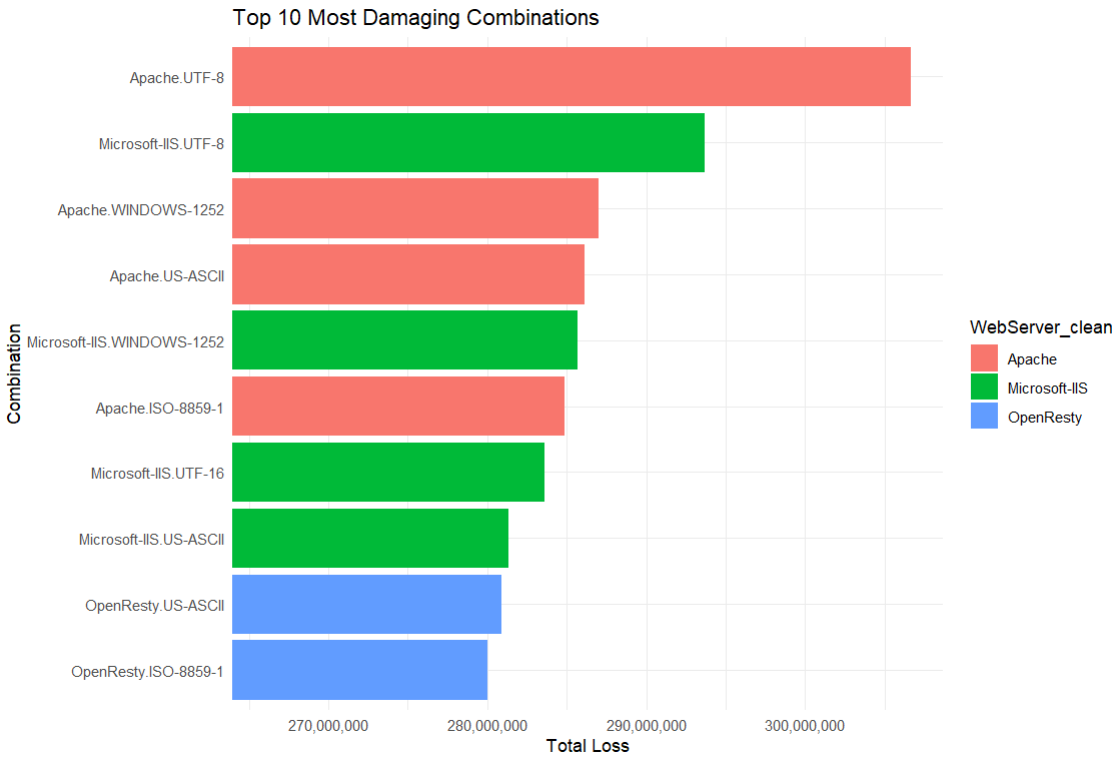


### Data Analysis

What are the top 10 WebServer and Encoding methods suffer the most Loss during cyber-attack?

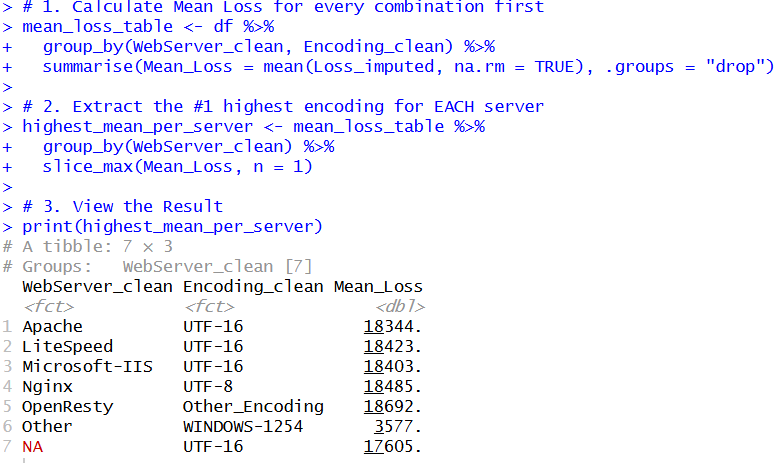


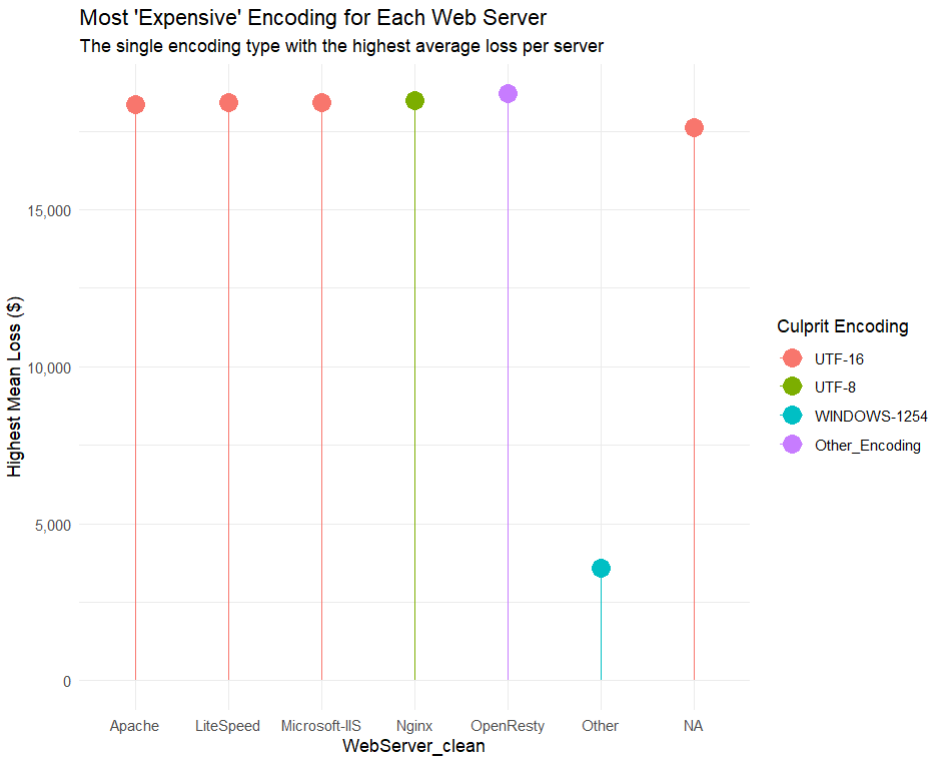
The above table shows the top 10 **WebServer** and **Encoding** method that suffers the most during cyber-attacks. From the table, the team discovers there is difference in total **Loss** between different **WebServer** and **Encoding** combinations.



The bar chart above shows the plotting of the top 10 **WeServer** and **Encoding** combinations. The plot shows a significant difference between the first combination, Apache as **WebServer** and UTF-8 as **Encoding** method, and the second combination, Microsoft-IIS as **WebServer** and UTF-8 as **Encoding** method.

Within each WebServer category, which specific Encoding method is associated with the highest mean Loss?



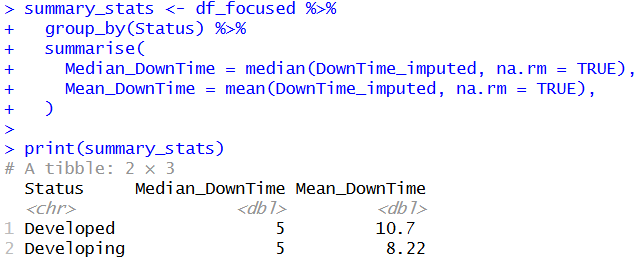


The table and chart above shows the **Encoding** method that results in the highest mean loss across each **WebServer**. The team sees that type of Encoding method within each **WebServer** that results in the most mean **Loss** is different.

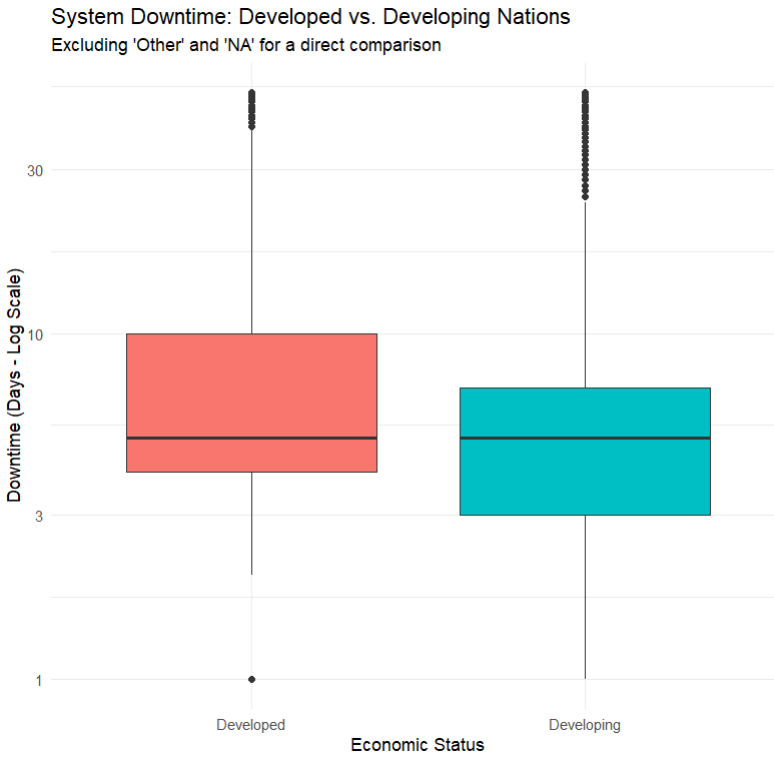
Is there a significant difference in median DownTime between servers hosted in 'developed' Country versus 'developing' Country?



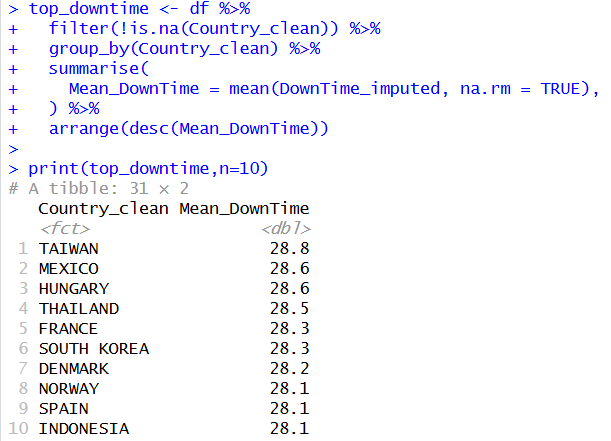
First categorized the Country into two groups, Developed and Developing country. Remove the “Other” category as there might be a mix of developed and developing country lies within, which might be unfair to the other category if it is labeled as one of those two categories and also brings bias to the data.

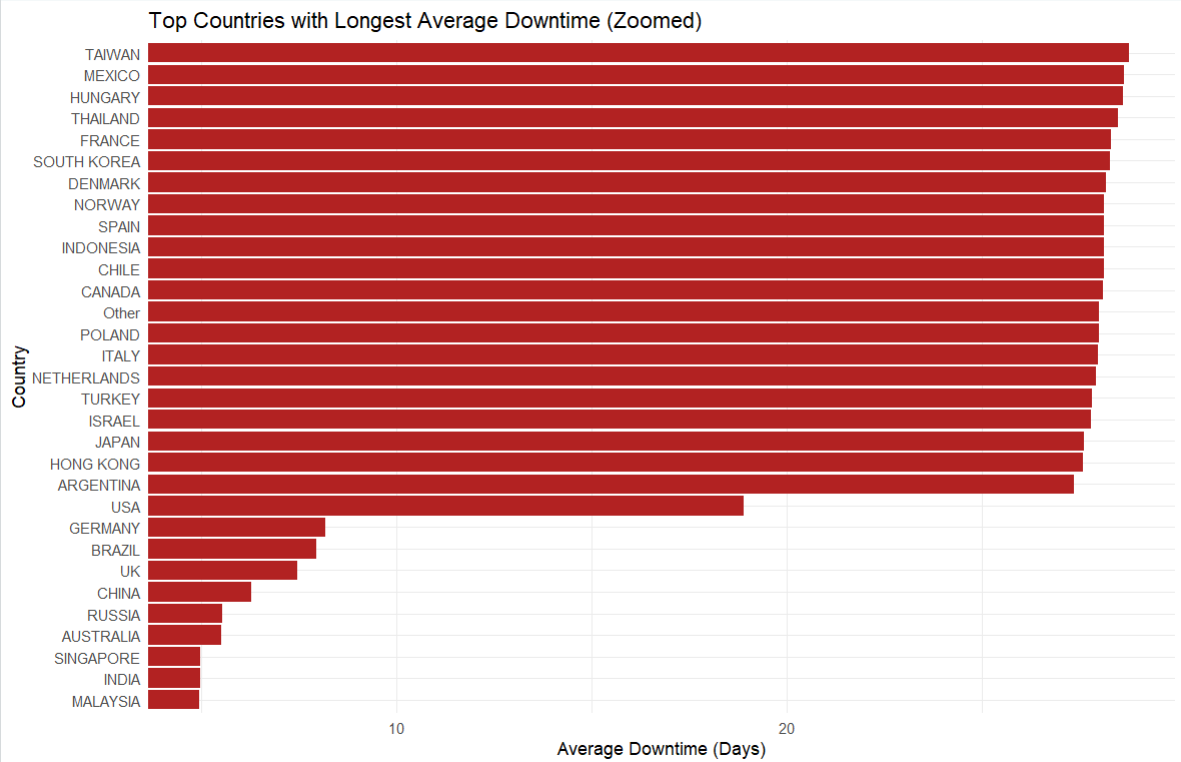


The table shows the median and mean DownTime according to developed and developing Country followed a cyber-attack. It shows no difference in the median, but some minor difference in the mean DownTime.



The box plot shows the distribution of log-scaled DownTime for Developed and Devoloping country. The reason for a log-scaled DownTime is because the data is not normal. The plot indicates that the black lines for those two categories stays at the same level which means they have no difference in the median value. Whereas the box for Developed country is leveled slightly higher than that of Developing country, indicating 50% of its data lean towards a greater DownTime.





The summary table and the bar chart shows the top Country that has the most mean DownTime. From both materials, the team discovers there is a deference in mean DownTime depending on the Country.

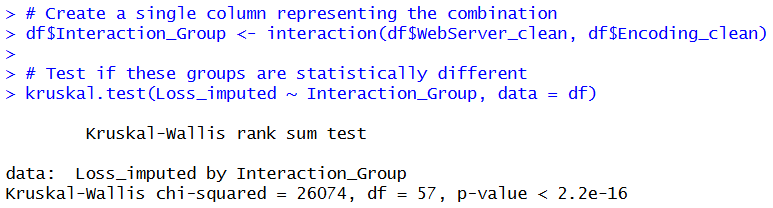
### Hypothesis Formulation and Testing

What are the top 10 WebServer and Encoding methods suffer the most Loss during cyber-attack?

H0 : WebServer and Encoding jointly have no significant influence on the total Loss during cyber-attack.

H1 : WebServer and Encoding jointly have significant influence on the total Loss during cyber-attack.

Since the data within the Loss variable is not normally distributed and the team is comparing categorical variables with numerical variable which involves three or more groups, the Kruskal-Wallis test is proposed.



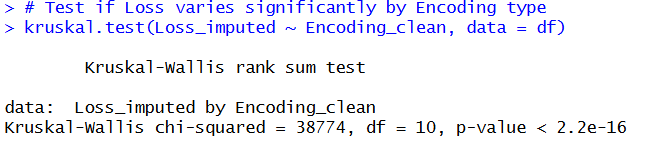
Based on the result, p-value shows 2.2e-16 which is sufficient to reject the null hypothesis.

Within each WebServer category, which specific Encoding method is associated with the highest mean Loss?

H0 : There is no significant difference between Encoding type and mean Loss within a specific WebServer during a cyber-attack.

H1 : There is a significant difference between Encoding type and mean Loss within a specific WebServer during a cyber-attack.

Since the data within the Loss variable is not normally distributed and the team is comparing categorical variables with numerical variable which involves more than two groups, the Kruskal-Wallis test is proposed.



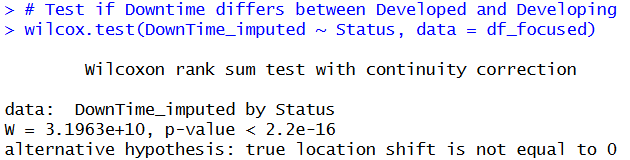
Based on the result, p-value shows 2.2e-16 which is sufficient to reject the null hypothesis.

Is there a significant difference in median DownTime between servers hosted in 'developed' Country versus 'developing' Country?

H0 : There is no significant difference in median DownTime between Developed and Developing country followed by a cyber-attack.

H1 : There is a significant difference in median DownTime between Developed and Developing country followed by a cyber-attack.

Since the data within the DownTime variable is not normally distributed and the team is comparing categorical variables with numerical variable which involves exactly two groups, the Wilcoxon Rank test is proposed.



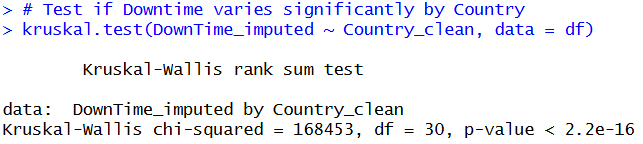
Based on the result, p-value shows 2.2e-16 which is sufficient to reject the null hypothesis.

What are the top countries experiencing the longest mean DownTime following a cyber-attack?

H0 : There is no significant difference between Country and mean DownTime followed by a cyber-attack.

H1 : There is a significant difference between Country and mean DownTime followed by a cyber-attack.

Since the data within the DownTime variable is not normally distributed and the team is comparing categorical variables with numerical variable which involve more than two groups, the Kruskal-Wallis test is proposed.



Based on the result, p-value shows 2.2e-16 which is sufficient to reject the null hypothesis.

## Gan Yew Joe (TP077191)

text

### 2.3.1 Objectives

The objectives of this project were to:

1. To examine whether specific forms of reporting entities are linked to the system recovery period duration using the influence of the reporting source (**Notify**) on the **DownTime** recorded.
2. To examine the relationship existing between system **DownTime** and financial **Loss** with the aim of establishing whether the factors are correlated in a linear manner with each other.

### Analysis Questions

To examine whether specific forms of reporting entities are linked to the system recovery period duration using the influence of the reporting source (**Notify**) on the **DownTime** recorded.

* What is the reporting source (Notify) that has the largest average system DownTime?
* Does recovery time (DownTime) differ significantly, by whether the attack was reported by professional security organization or independent organization?

To examine the relationship existing between system **DownTime** and financial **Loss** with the aim of establishing whether the factors are correlated in a linear manner with each other.

* How much money does the company lose in an average extra system DownTime?
* Are there any special threshold days of DownTime that the financial Loss increases exponentially and not linearly?

### Data Preparation

text

### Data Analysis

text

### Hypothesis Formulation and Testing

Text

## 2.4 Liu Wei (TP085412)

Text

### 2.4.1 Objectives

The objectives of this project are:

1. To examine if the website extension (Top Level Domain) in the **URL** affects the amount of financial **Loss** from an attack.
2. To analyze the impact of the **WebServer** version on the server **DownTime** to determine which types of server remain unavailable for the longest duration.

### 2.4.2 Analysis Questions

Text

RO1:

Do certain website extensions (like .com or .gov) have higher financial **Losses** than others?

Which specific website extension is linked to the most money lost in total?

RO2:

Which **WebServer** version results in the highest number of **DownTime** days?

Is there a big difference in **DownTime** between the different types of **WebServers**?

### 2.4.3 Data Preparation

text

### 2.4.4 Data Analysis

text

### 2.4.5 Hypothesis Formulation and Testing

Text

# Group Hypothesis

In this section, the team forms a composite group hypothesis that will integrate all one hypothesis proposed by each member previously.

## Group Hypothesis Formulation

text

## Group Hypothesis Testing

Text

## Overall Conclusion

text

# 4.0 Summary

Short introduction here

## Limitations and Recommendations

text

## Title

Text

## Title

text

# Title 5

Short introduction here

## Title

text

## Title

Text

## Title

text

# Title 6

Short introduction here

## Title

text

## Title

Text

## Title

text

# Conclusion

References

Don’t do anything here, leave it as the last step.

# Appendix