Seismic Response Analysis: Building a Safer Infrastructure

CI7340 – Applied Data Programming

Submitted to:
Faculty of Science, Engineering and Computing
School of Computer Science and Mathematics
Department of Computer Science

Submitted by:

Student Name	KID	Course
Shashwat Bhardwaj	K2149137	MSc Data Science Sep 2021
Vinod Raja	K2102329	MSc Data Science Jan 2021
Padmesh Upadhyay	K2136572	MSc Data Science Sep 2021
Rohit Chamle	K2055291	MSc Data Science Sep 2021

INDEX

List OF Tables

List OF Figures

ABSTRACT

Natural disasters like an earthquake cause an enormous economic loss as well as loss of life. The task in hand is to analyse the given data which describes building infrastructure and damage grade caused by an earthquake in different geographic locations and find pattern or features which would contribute in aggravating infrastructure damage. This research aims to provide local government with crucial insights so that they can devise better earthquake safe infrastructure plans, improving existing infrastructure and possibly predicting damage in the future.

The dataset will be analysed with data science techniques using python and various supporting libraries like NumPy, pandas, matplotlib, seaborn etc. Patterns collected using this analysis will be documented in a report and will be presented in neat graphs and plots.

INTRODUCTION

"Data science means doing analytics work that, for one reason or another, requires a substantial amount of software engineering skills." (Cady Field, 2017, pp. 1)

The main objective of Data Science is to extract valuable information, actionable insights, for strategizing and making well-informed business decisions. Data Science is a field which involves various stages like -

1. Framing the problem,
2. Understanding the data,
3. Extract features,
4. Model and Analyse,
5. Present Results,
6. Deploy code

(Cady Field, 2017, pp. 9)

Data science is widely used across various industries like Banking, E-commerce, Education, OTT platforms like (Netflix, Amazon), etc.

In this project, we are provided with two datasets in csv formats, first contains detailed information on various characteristics of buildings in different geographic locations. This data describes details of their dimensions, foundation, structure as well as their usage. The second one contains damage grade on each the given buildings caused by an earthquake.

Using these datasets, we will try to mine information, analyse it, extract patterns and insights which could potentially provide some valuable inputs to the local governments, enabling them to make well-informed plans for a more secure infrastructure in future, improving existing infrastructure and disaster management.

The initial analysis of dataset has shown correlations between some features of the building with their damage grade. These features may have potentially contributed to a more vulnerable infrastructure and hence aggravated the damage caused to them during the earthquake. This has given rise to various research question which will act as basis of our analysis in this report. Mentioned below are the research questions identified:

Surface and Foundation Analysis:

1) What are the worst impacted regions? What are the more vulnerable land surface conditions in them?
2) In buildings with highest damage, what are the most common foundation types and land surface conditions?

Floor Type and Structure Type Analysis:

3) What is the worst impacted ground floor type and foundation type combination in regions with the highest damage?
4) Which structure type and foundation type combinations are unsafe in regions with most impact?

Reliability feature Analysis:

5) What is the relatively safer floor count per foundation type across different grades of damage?
6) What is safest floor type and roof types combination across different grades of impacts? (Includes ground floor as well as other floor types)

Boundary Condition Analysis:

7) What is the safe floor count for the buildings in regions with highest damage?
8) After what age a building becomes more prone to damage across regions?

Relative property analysis:

9) Does secondary use for a building make it more vulnerable? What type of secondary use of building is most common in regions with highest damage grade?
10) What is most common position of highly damaged buildings?

In search of answers to our research questions we will follow the Data science methodology. We will begin our research with initial data analysis which includes datatype analysis, data cleaning, handling missing values and merging the two datasets. Next, we will perform the Exploratory data analysis where we will deal with outliers or extreme values, calculating descriptive statistics, output of EDA should in insights and patterns which would be presented in the form of graphical visualization like Line chart, Bar plot, Heat-map, Scatter, etc.

WORKFLOW

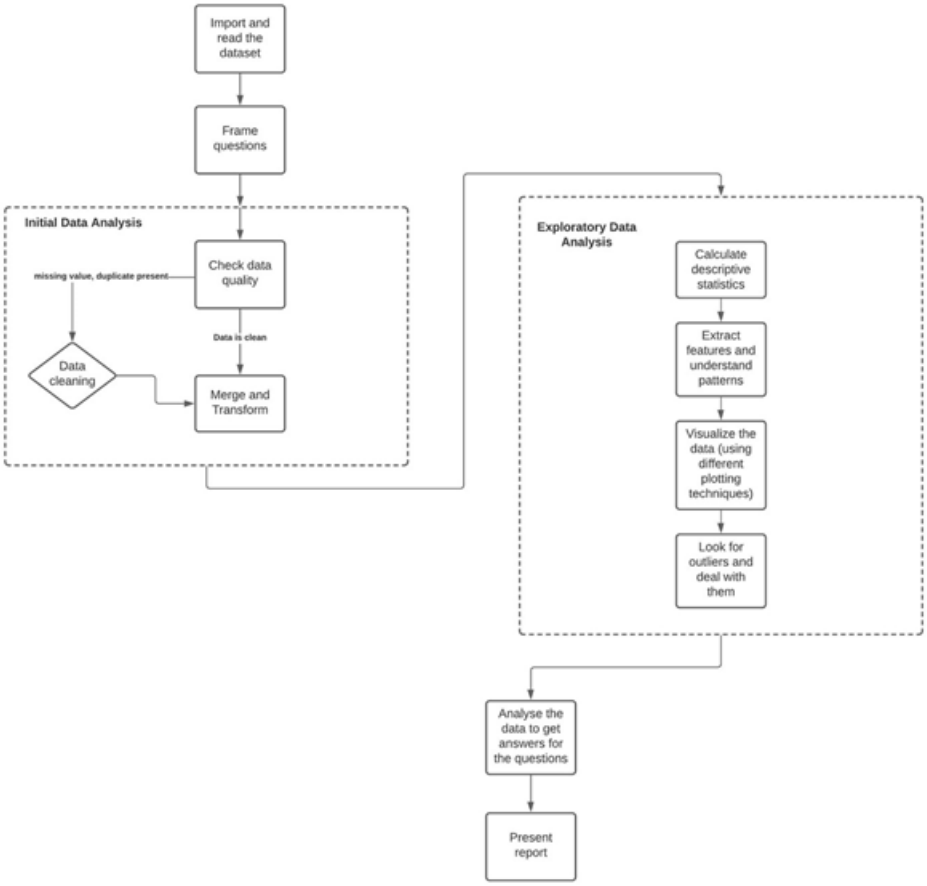


Figure 1: Workflow

TOOLS, DATASET AND PRELIMINARY DATA ANALYSIS

Programming language & Tools:

Python is one of the most popular programming language for Data Science and machine learning projects. It is an Interpreted language and often draws comparison with various other tools/programming languages which are also widely used for data analysis in the industry, namely R, MATLAB, SAS. Mentioned below is a brief comparison of python with the others –

Feature	Python	R	SAS
Cost	Free		Expensive Commercial Software
Learning Curve	Simple to learn	Steep learning curve	Easy to learn, especially for those who have SQL background
	Large community support available		Comprehensive Documentation & dedicated customer service
Data Handling	Good Data Handling capabilities		
	Big data support		
	Parallel Computation		
Data Visualization	Highly advanced visualization		
	Simple customization of plots		Cumbersome customization of plots
Advancement	Quicker Updates		Updates available in new version rollout.
	Chances of errors in latest development		Extensively tested releases, chances of bugs less.
Deep Learning Support	Highly advanced packages like tensor flow and keras	R acts as an interface for python's tensor flow and keras	Work in progress

Figure 2: Comparison for Programming languages (Jain Kunal, 2017)

Python is the second best choice in most situations and is the jack of all trades. There are better options available for Statistics, doing numerical computations or for web parsing but if we want to do all these things in one single project then python becomes the best option. (Cady Field, 2017, pp. 4). For its ease of use, good data handling capabilities, cost effectiveness, we will be using python for analysis in this project.

The given datasets are in csv format and hence Python's pandas, NumPy, Statistics libraries look ideally suited for this use case. For visualization on the other hand, libraries like matplotlib and seaborn seem more fitting due to their ease of use and highly customizable nature. Methods from the above-mentioned libraries will be used to compute correlation between features of buildings, finding/handling null values, plotting visualizations, doing group by and performing other operations.

Amongst various programming tools and Integrated Development environments like Google colab, Jupyter Notebook, PyCharm, and Spyder we would be using Jupyter Notebook for its simplicity and ease of use.

Datasets:

We are given two csv files namely input_features and target_values. Input_features include information on various aspects of building such as its location, age, floor count, foundation type, construction type and even details of its usage and occupancy. Target_values on the other hand contains the grade of damage caused to each building by the earthquake. The grade of damage is categorized from 1 to 3 in ascending order of damage caused. The data in two csv can be joined/merged on Building ID column.

The dataset features are of integer, binary and object datatypes. Data in columns like foundation type, land surface condition, roof type, floor type and plan configuration seem suitable for categorization. This would make these columns co-creatable with other columns, especially the damage grade.

Initial Data Analysis:

Initial Data analysis(IDA) is the process of data inspection which is carried out after data collection and before formal statistical analysis. IDA involves loading, transforming, and wrangling the dataset to make it suitable for statistical analysis. Data Wrangling is a process of fetching raw data and extracting something more appropriate for further analysis. One may create a software utility to extract data from whatever the source may be, cleaning it and converting it into a more usable format.

The initial data pre-processing involves following steps

- **Framing the Problem:** Asking the right question is the most important step in data science and no amount of technical expertise can make up for the time lost in trying to solve the wrong problem.
- **Removing unnecessary data**: Unnecessary data involves artificial/dummy values in a column or data that contains incorrect features.
- **Checking for outliers**: Outliers can skew the data set which will have impact on analysis and provide wrong results.
- **Checking for missing data**: Missing value or null value can hinder the analysis. Missing data can be handled by either dropping the row, filling with 0 or using mean/median.
- **Data visualization**: Data visualization technique is used to observe the data distribution, check for any outliers/extreme values and to find correlation between columns.
- **Possible data reconstruction**: This step involves merging and transforming the dataset by categorizing applicable columns, to make them suitable for visualization and correlating with other columns. (Dr. Barman Nabajeet, 2021)

Framing the research questions has already been done and document in the report. Next step would be to check for unnecessary data. For this we will have to import the given data sets using pandas' read csv method into a DataFrame.

```
In [35]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 260601 entries, 0 to 260600
Data columns (total 40 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   building_id                             260601 non-null  int64
 1   geo_level_1_id                          260601 non-null  int64
 2   geo_level_2_id                          260601 non-null  int64
 3   geo_level_3_id                          260601 non-null  int64
 4   count_floors_pre_eq                     260601 non-null  int64
 5   age                                     260601 non-null  int64
 6   area_percentage                         260601 non-null  int64
 7   height_percentage                       260601 non-null  int64
 8   land_surface_condition                  260601 non-null  object
 9   foundation_type                         260601 non-null  object
 10  roof_type                               260601 non-null  object
 11  ground_floor_type                       260601 non-null  object
 12  other_floor_type                        260601 non-null  object
 13  position                                260601 non-null  object
 14  plan_configuration                      260601 non-null  object
 15  has_superstructure_adobe_mud            260601 non-null  int64
 16  has_superstructure_mud_mortar_stone     260601 non-null  int64
 17  has_superstructure_stone_flag           260601 non-null  int64
 18  has_superstructure_cement_mortar_stone  260601 non-null  int64
 19  has_superstructure_mud_mortar_brick     260601 non-null  int64
 20  has_superstructure_cement_mortar_brick  260601 non-null  int64
 21  has_superstructure_timber               260601 non-null  int64
 22  has_superstructure_bamboo               260601 non-null  int64
 23  has_superstructure_rc_non_engineered    260601 non-null  int64
 24  has_superstructure_rc_engineered        260601 non-null  int64
 25  has_superstructure_other                260601 non-null  int64
 26  legal_ownership_status                  260601 non-null  object
 27  count_families                          260601 non-null  int64
 28  has_secondary_use                       260601 non-null  int64
 29  has_secondary_use_agriculture           260601 non-null  int64
 30  has_secondary_use_hotel                 260601 non-null  int64
 31  has_secondary_use_rental                260601 non-null  int64
```

Figure 3: Data Info IDA

We then observe the datatypes and null counts for each columns in respective DataFrames. This will give us a target(s) for any null handling if required.

```
In [37]: df.isna().sum()

Out[37]: building_id                             0
         geo_level_1_id                          0
         geo_level_2_id                          0
         geo_level_3_id                          0
         count_floors_pre_eq                     0
         age                                     0
         area_percentage                         0
         height_percentage                       0
         land_surface_condition                  0
         foundation_type                         0
         roof_type                               0
         ground_floor_type                       0
         other_floor_type                        0
         position                                0
         plan_configuration                      0
         has_superstructure_adobe_mud            0
         has_superstructure_mud_mortar_stone     0
         has_superstructure_stone_flag           0
         has_superstructure_cement_mortar_stone  0
         has_superstructure_mud_mortar_brick     0
         has_superstructure_cement_mortar_brick  0
         has_superstructure_timber               0
         has_superstructure_bamboo               0
         has_superstructure_rc_non_engineered    0
         has_superstructure_rc_engineered        0
         has_superstructure_other                0
         legal_ownership_status                  0
         count_families                          0
         has_secondary_use                       0
         has_secondary_use_agriculture           0
         has_secondary_use_hotel                 0
         has_secondary_use_rental                0
         has_secondary_use_institution           0
         has_secondary_use_school                0
         has_secondary_use_industry              0
         has_secondary_use_health_post           0
         has_secondary_use_gov_office            0
         has_secondary_use_use_police            0
         has_secondary_use_other                 0
         damage_grade                            0
         dtype: int64

In [38]: df.isna().any().sum()

Out[38]: 0

In [39]: df.shape

Out[39]: (260601, 40)
```

Figure 4: Null checks in dataset

Next we check the dataset for outliers in relevant numeric columns using box-whisker plot or scatter plots. Plotting a histogram can be a good option to check distribution of values in

various columns. Once the outliers and/or extreme values are identified, null or missing values will be handled. Missing values can be handled by replacing them with one of the Central Tendency or instead the column could be discarded if it is not relevant for our research.

```
In [31]: df = pd.DataFrame()

In [32]: df = pd.merge(df2, df3, on='building_id')

In [33]: df.head()
```

Out[33]:

	building_id	geo_level_1_id	geo_level_2_id	geo_level_3_id	count_floors_pre_eq	age	area_percentage	height_percentage	land_surface_condition	foundation
0	802906	6	487	12198	2	30	6	5	t	
1	28830	8	900	2812	2	10	8	7	o	
2	94947	21	363	8973	2	10	5	5	t	
3	590882	22	418	10694	2	10	6	5	t	
4	201944	11	131	1488	3	30	8	9	t	

5 rows × 40 columns

```
In [34]: df.tail()
```

Out[34]:

	building_id	geo_level_1_id	geo_level_2_id	geo_level_3_id	count_floors_pre_eq	age	area_percentage	height_percentage	land_surface_condition	foun
260596	688636	25	1335	1621	1	55	6	3	n	
260597	669485	17	715	2060	2	0	6	5	t	
260598	602512	17	51	8163	3	55	6	7	t	
260599	151409	26	39	1851	2	10	14	6	t	
260600	747594	21	9	9101	3	10	7	6	n	

5 rows × 40 columns

Figure 5: Merging two datasets

The two datasets will be merged into a single DataFrame using Building Id column. Correlation of Damage Grade will be checked with other features to asses key features for Exploratory Data Analysis.

EXPLORATORY DATA ANALYSIS

Introduction to EDA

Exploratory data analysis (EDA) is a comparatively more creative stage in the Data Science Roadmap. It involves playing with data, correlating features to give you deeper insight on what data represents, by using basic statistics and data visualization techniques like scatter plot, Bar plot etc.

EDA is very important stage and gives below expected outputs –

- **Understanding data:** To get a more intuitive sense of the data, identify important features and possibly extract patterns in them.
- **Hypothesis:** Checking correlation coefficient between different features which could possibly lead to creation of set of hypotheses.
- **Detect Anomalies**: outliers and extreme values in data

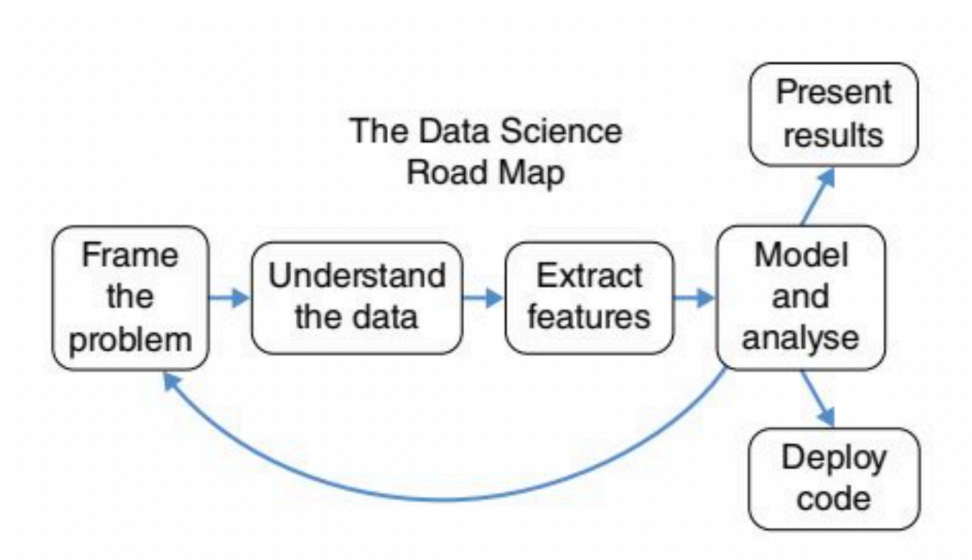


Figure 6: Data Science Roadmap

(Cady Field, 2017, pp. 9)

Descriptive Statistics:

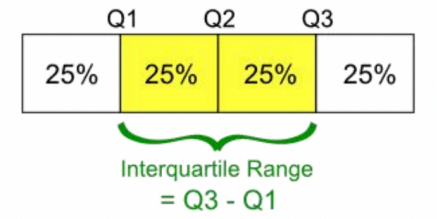
Descriptive Statistics involves describing various features of dataset and their relationships with other features using numeric calculations, graphs or tables. Descriptive Statistics can be measured by –

1. **Central Tendency:** 'Central Number' which can be best used to summarise the entire set of features recorded.
    - **Mean**: A mean or 'Average' is the central number around which the whole data is spread out. It is best for Numeric data without outliers.

$$M = \bar{x} = \frac{1}{N} \cdot \sum_{i=1}^{N} value_i = \frac{sum(values)}{count(values)}$$

- **Median**: A median is the central value of a logically sorted dataset.
- **Mode**: A mode is the most common value in your data. Unlike median and mean, mode can be used to describe central tendency of both Numeric and Non-Numeric data.

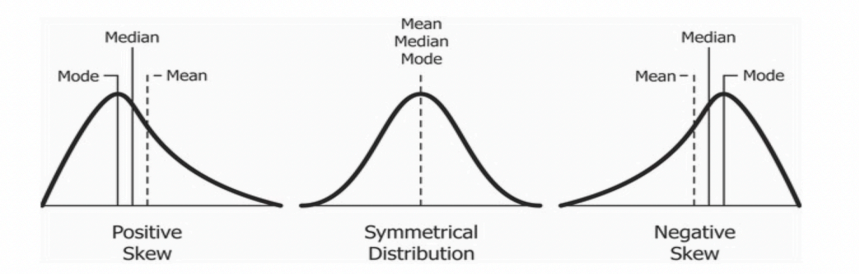
2. **Spread**: Spread of your data is aimed to quantify the variability in your data.
   - **Range**: Range is the difference between the highest and the lowest value in a dataset Measuring Range of set may not be useful if data contains outlier(s).
   - **Percentiles**: It is a way to represent relative position of the value in a set. Example, if a value is 90th percentile then 90% values in the set and smaller that it.
   - **Inter Quartile Range**: IQR is calculated by measuring the spread by dividing data into smaller quartiles. IQR is difference between 75th and 25th percentile of the sorted data.



   - **Standard Deviation**: Standard deviation is the average difference between the mean and each value in the data. Lower value means data is less spread out while a higher value means data points are spread out to a higher range values.

$$\text{S.D.} = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n}(x-\bar{x})^2}$$

   - **Variance**: Variance is the square of the standard deviation.
   - **Z score**: A Z-score is the number of standard medians; a particular number is away from mean.



(Narkhede Sarang, 2018)

Choice of methodology for calculating central tendency will depend on the distribution of data, if data contains outliers or extreme values then calculating the mean may not provide any intuitive interpretation. However it may not be a huge problem for Median. Hence it would make sense to check for data spread beforehand using a Box and Whisker Plot.

Central Tendency may be calculated for numerical features like family count, building age, floor count as well as area and height percentage of the building. For features with Categorical data like foundation type, floor types, position and plan configuration, mode value may give similar insights. For features containing binary data like secondary use of building and superstructure type, occurrence frequency calculation per variation in various damage grades may be useful in revealing crucial patterns.

For measuring spread for the given dataset, the most fruitful measurements are expected to be the correlation and IQR. Calculating correlation of Damage grade feature with other features like foundation type, land surface type, various super structure type etc will prove crucial in giving a direction to the analysis. It would make sense to investigate and look for such correlations. Calculating the Range, IQR calculation and Standard Deviation for features like age, floor count and family count living in the building should highlight the spread of data and will reveal the outliers and/or extreme values (if any). A Z-score may also give us a good insight when computed for properties like floor count and family count living in the building since data spread in these is expected to be lesser than other features. Calculating the percentile may not give an output since the data set is huge in our case.

Data visualization:

Data visualization is a branch of data science that deals with graphical representation of data using effective data visualization methods like statistical graphics, plots, information graphics and other tools to make the insights/pattern more comprehendible for non-technical and business users. (Tableau, 2021)

Following are the key points to remember while developing visuals:

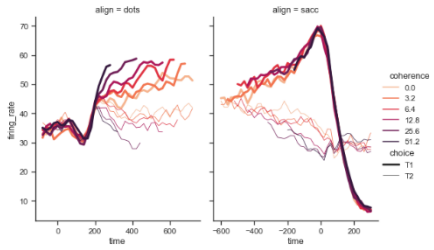
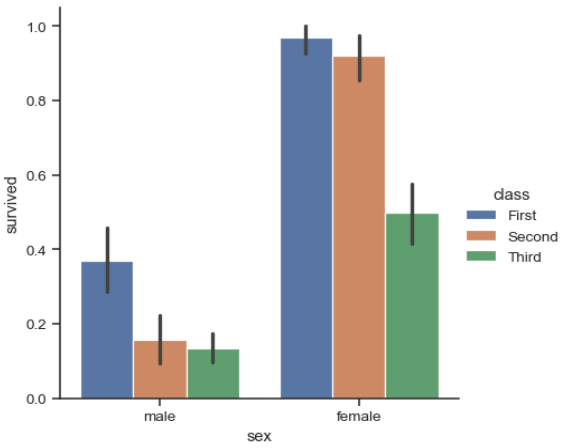
- **Best visual for data**: it is important to understand the volume and type of data in hand to select the best suited charts, plots, or graph for target audience.
- **Lie Factor:** It is a value to describe the relation between size of effect shown in visualization and the size of effect in dataset.

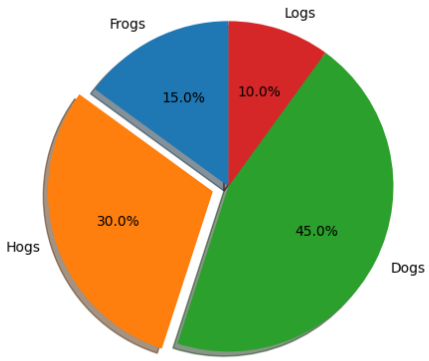
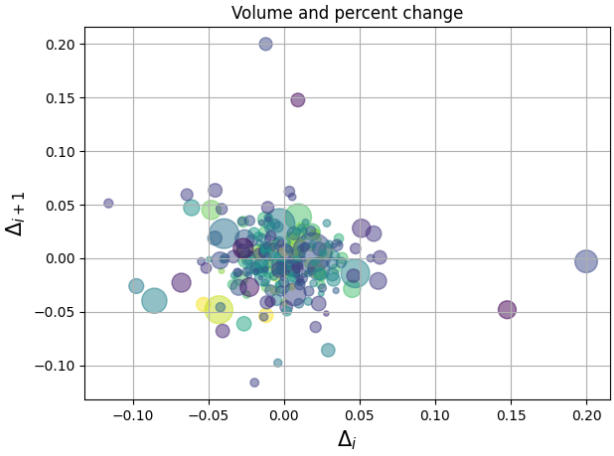
$$\text{Lie Factor} = \frac{\text{Size of graphic}}{\text{Size of Data}}$$

- **Focus on key areas:** ensure that the key insights or areas of analysis are well highlighted.
- **Keep it simple:** ensure that your design and visuals are simple, readable, and easy to understand.

Use patterns and Compare Aspects: you can display similar type of information with the help of patterns, and you can establish a pattern by using similar chart type, colours, or other elements. (GoBeyond.AI, 2019)

Type of Charts and graphs commonly used to present data are:

Type	Example	What?	Use
Line Chart	 *Figure 7: Line Plot*	A line chart represent data that changes data over continuously over time	-When comparing two or more variables or information over a given time-period. (Waskom Michael, 2021)
Bar Chart	 *Figure 8: Bar Plot*	Bar chart represents categorical data in form of rectangular bars.	-When you represent data that are grouped into nominal or ordinal categories. -To show the comparison between/among the data. -Bar graphs are ideal to show the distribution when data have more categories

			(more than three) (Waskom Michael, 2021)
Pie Chart	 *Figure 9: Pie Chart*	Pie charts represent data and statistics in easy and understandable "pie-slice" format and illustrate numerical proportion.	-When you want to show the composition of category in the given data set. -It's very useful and easy to display nominal or ordinal categories. -To show the proportional data. (Hunter, Dale, Firing, Droettboom and Matplotlib development team, 2021)
Scatter Plot	 *Figure 10: Scatter Plot*	Scatter plots visualize the relationship between two variables.	-Help in determining the relationship between two variables. -In predicting behaviour of dependent variable based on the measure of independent variable. (Hunter, Dale, Firing, Droettboom and Matplotlib development team, 2021)

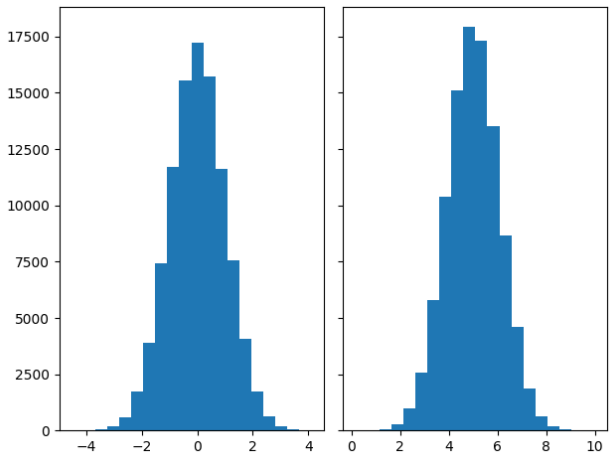
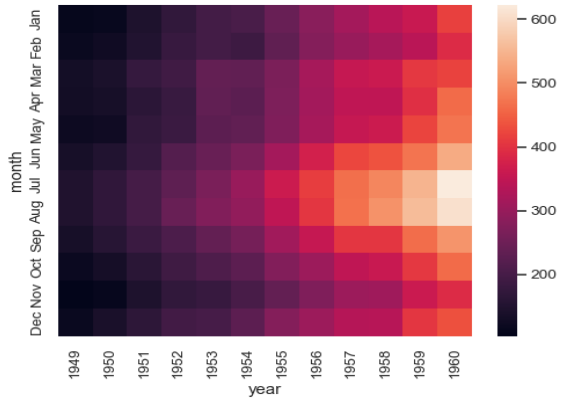
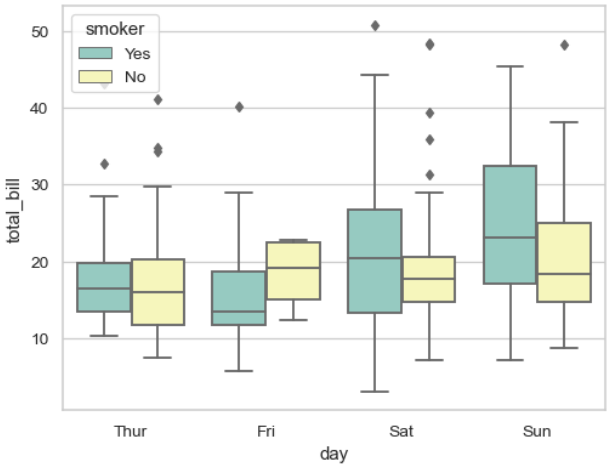
Histogram	 *Figure 11: Histogram*	Histogram shows frequency distribution of dataset in ordered rectangular columns.	-To communicate data distribution. -It is used when data is continuous. -Help in summarizing the large dataset. (Hunter, Dale, Firing, Droettboom and Matplotlib development team, 2021)
Heat Map	 *Figure 12: Heat Map*	it is graphical representation of numeric data. Individual data points present in the data set are represented by different colour.	-Simplifies numeric data and depicts it using colour. -Easily shows the most and least or high and low density points of a feature. (Waskom Michael, 2021)
Box & Whisker Plot	 *Figure 13: Box & Whisker Plot*	It is a way to graphically represent the data distribution through their quartiles and to find extreme/outliers in dataset.	Useful in displaying when distribution is skewed. -Helpful in detection of outliers and extreme values. -Useful when comparison between two or more datasets is required. (Waskom Michael, 2021)

Table 1: Types of Visualizations

In our analysis and visualization purpose we will be using following visualizations:

- **Histogram:** To show the data distribution of geo_level_1_id over the damage_grade.
- **Bar Plot:** To show impact of earthquake on different superstructure by comparing with damage grade.
- **Line chart:** To show the spread of ground floor type impact of earthquake by comparing with damage grade.
- **Box and Whisker Plot:** to visualize the mean and median of building age and impact of earthquake. This will also help in finding the outlier or extreme value.
- **Heat Map:** This is used to visualize the age data points over foundation type in geo_level_1_id.

SUMMARY AND CONCLUSION

A DataFrame will be constructed to inspect data types and null values for two datasets provided to us. Our next step will be to merge the two datasets using the "building_id" column. Following this, we will categorize a few non-numeric columns. We will then be able to check the correlation between various columns with "damage_grade" and use this information to prepare a few research questions aimed at identifying the factors leading to structural aggravation caused by an earthquake in particular buildings. We will analyse the dataset and present the results using Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn. As part of our next practical report, we will present the actual investigations and results from the research questions posed in this report.

REFERENCES

Cady Field (2017) *The Data Science Handbook Ebook central* [Online] Available at: https://ebookcentral.proquest.com/lib/kingston/reader.action?docID=4790656 (Accessed: 22/10/2021) pp. 1

Cady Field (2017) *The Data Science Handbook Ebook central* [Online] Available at: https://ebookcentral.proquest.com/lib/kingston/reader.action?docID=4790656 (Accessed: 22/10/2021) pp. 4

Cady Field (2017) *The Data Science Handbook Ebook central* [Online] Available at: https://ebookcentral.proquest.com/lib/kingston/reader.action?docID=4790656 (Accessed: 24/10/2021) pp. 9

Cady Field (2017) *The Data Science Handbook Ebook central* [Online] Available at: https://ebookcentral.proquest.com/lib/kingston/reader.action?docID=4790656 (Accessed: 24/10/2021) pp. 9

Dr. Barman Nabajeet (2021) 'Exploratory Data Analysis' [PowerPoint presentation] *CI7340: Applied Data.* Available at: https://kingston.app.box.com/s/rbgvcwhwts9sr5qb6h2st1702eyyyk6h (Accessed: 28/10/2021)

GoBeyond.AI (2019) *7 Key Principles of Effective Data Visualization* Available at: https://medium.com/gobeyond-ai/7-key-principles-of-effective-data-visualization-b854b0b81946 (Accessed: 30/10/2021)

Hunter John, Dale Darren, Firing Eric, Droettboom Michael and MatplotLib development team (2021) *Basic pie chart* Available at: Basic pie chart — Matplotlib 3.4.3 documentation (Accessed: 4/11/2021)

Hunter John, Dale Darren, Firing Eric, Droettboom Michael and MatplotLib development team (2021) *Histogram* Available at: Histograms — Matplotlib 3.4.3 documentation (Accessed: 4/11/2021)

Hunter John, Dale Darren, Firing Eric, Droettboom Michael and MatplotLib development team (2021) *Scatter Demo2* Available at: Scatter Demo2 — Matplotlib 3.4.3 documentation (Accessed: 4/11/2021)

Jain Kunal (2017) *Python vs. R vs. SAS – which tool should I learn for Data Science?* Available at: https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/ (Accessed: 24/10/2021)

Narkhede Sarang (2018) *Understanding Descriptive Statistics* Available at: https://towardsdatascience.com/understanding-descriptive-statistics-c9c2b0641291 (Accessed: 7/11/2021)

Tableau (2021) *Data Visualization* Available at: https://www.tableau.com/en-gb/learn/articles/data-visualization (Accessed: 4/11/2021)

Waskom Michael (2021) *Line plots on multiple facets* Available at: Line plots on multiple facets — seaborn 0.11.2 documentation (pydata.org) (Accessed: 6/11/2021)

Waskom Michael (2021) *Plotting with categorical data* Available at: Plotting with categorical data — seaborn 0.11.2 documentation (pydata.org) (Accessed: 6/11/2021)

Waskom Michael (2021) *seaborn.heatmap* Available at: seaborn.heatmap — seaborn 0.11.2 documentation (pydata.org) (Accessed: 7/11/2021)

Waskom Michael (2021) *seaborn.boxplot* Available at: seaborn.boxplot — seaborn 0.11.2 documentation (pydata.org) (Accessed: 7/11/2021)