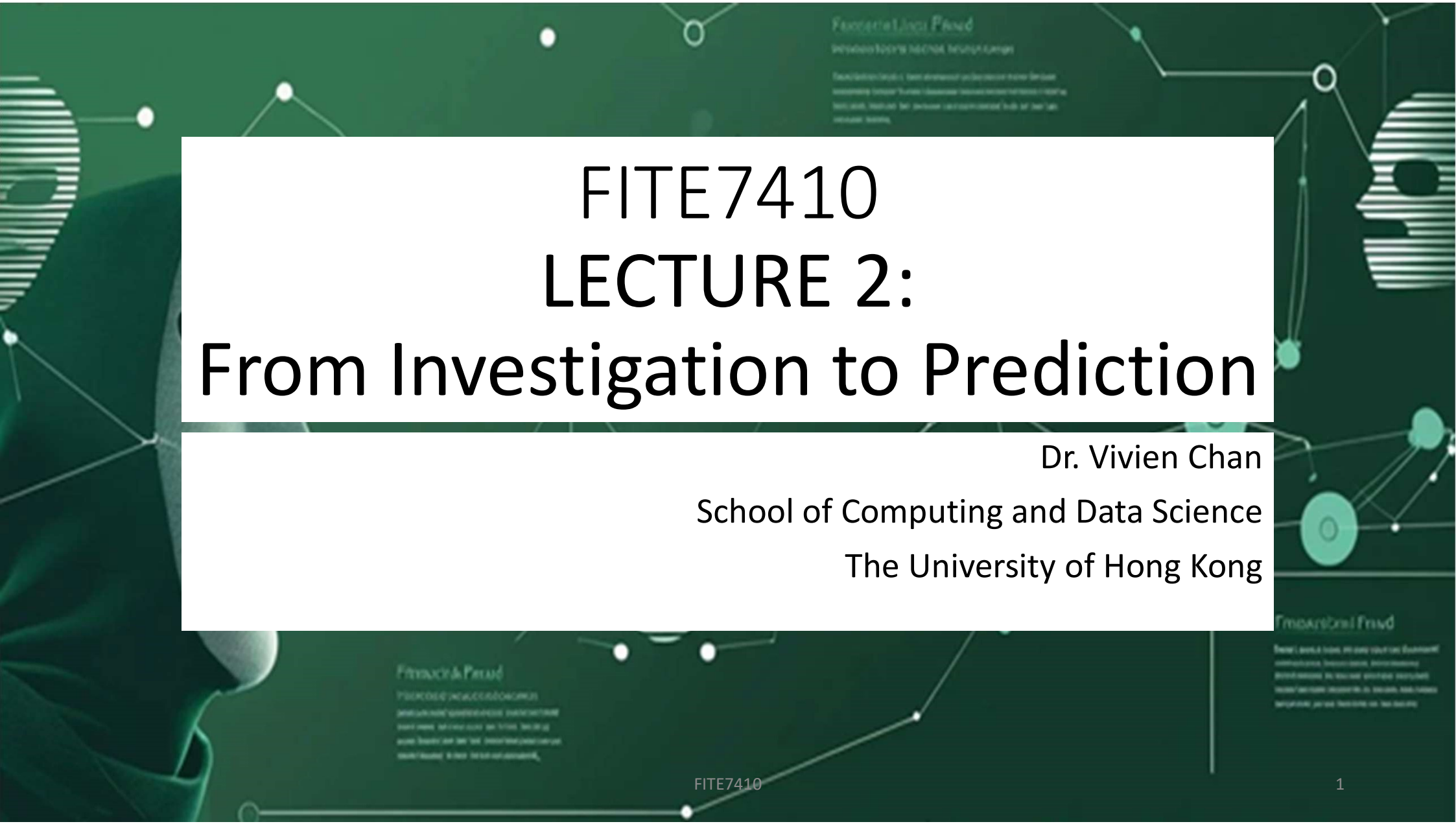


FITE7410
LECTURE 2:
From Investigation to Prediction

Dr. Vivien Chan

School of Computing and Data Science

The University of Hong Kong

Agenda

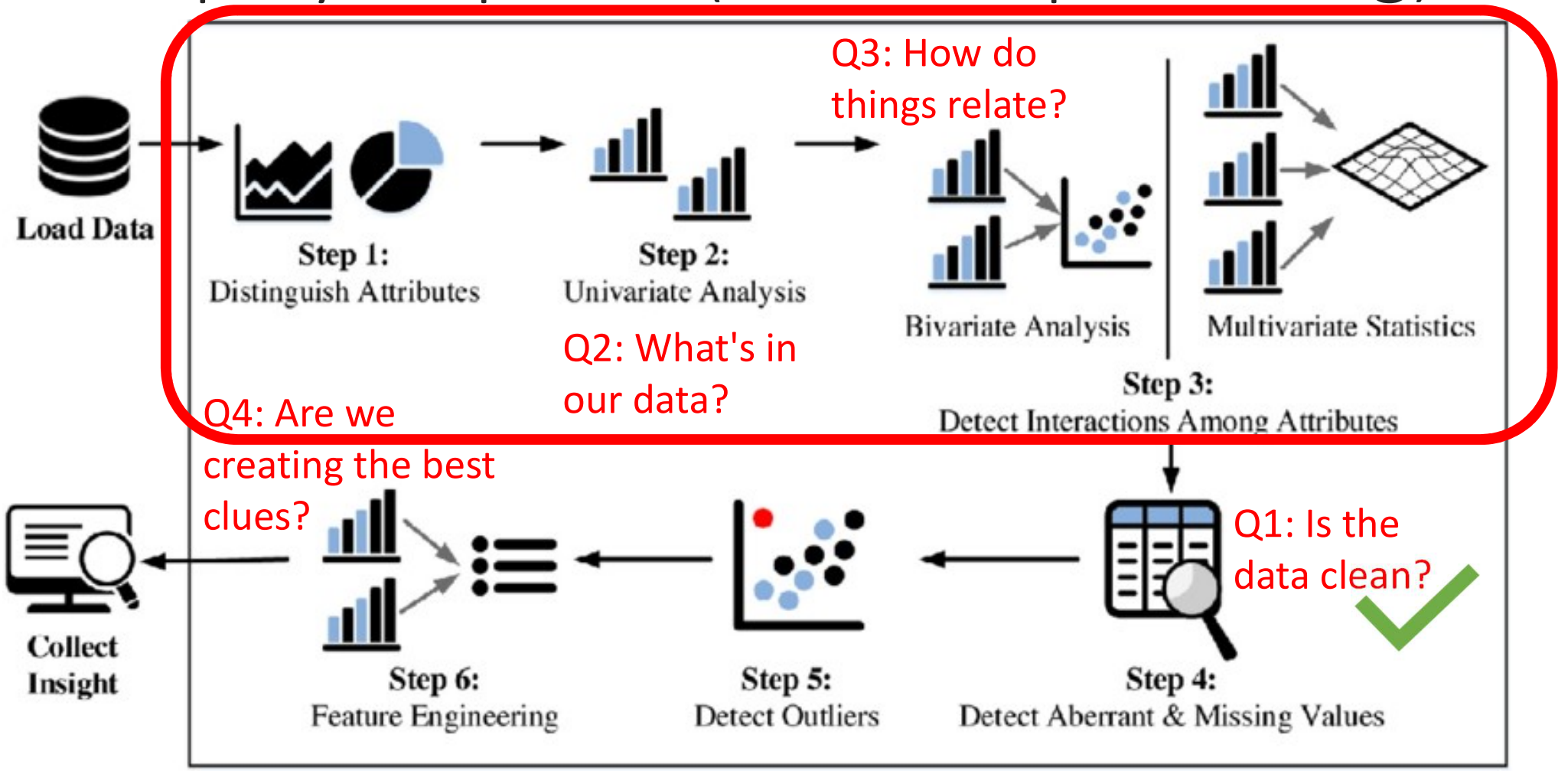
Our goal is to move from suspicion to evidence. When a company's financials seem too good to be true, we need a systematic way to find potential manipulation.

Lecture goals:

- 1/ Complete the Investigation: Move from univariate clues to uncovering hidden relationships in the data.
- 2/ Prepare the Data for the Model: Address the critical issues (skew, imbalance) we discover to build a reliable fraud detector.

Using EDA: The Initial Investigation

Step-by-step EDA (Data Pre-processing)



Ref: Ghosh et al. (2018). A comprehensive review of tools for exploratory analysis of tabular industrial datasets

EDA – Step 1: Distinguish Attributes

Objective

- To identify the attributes in a dataset in order to formulate a clearer goal of the data analytics process
- To understanding the meaning of each attribute before analyzing the data

Explore what?

- Attribute names, datatypes, number of attributes, etc.
- Continuous vs Categorical data types

Techniques

- Descriptive summary of the dataset

EDA – Statistics Examples

- Make use of descriptive statistics, including, mean, median, standard deviations, percentile distribution, etc.
  - Mean value and median value are basic descriptive statistics for continuous variables; while mode (i.e. most frequently occurring value) is used for categorical variables
  - Standard deviation can provide insight with respect to how much data is spread around the mean.
  - Percentile value like 10, 25, 75, 90 percentile provide information about the distribution of the data

- Descriptive statistics can be used as a preview to check the symmetry or asymmetry of the distribution, i.e. skewness of the data.

EDA – statistics examples

- Basic statistics measurement

	Time	V1	V2	V3	V4
count	1282.000000	1282.000000	1282.000000	1282.000000	1282.000000
mean	95362.820593	-0.005273	-0.031155	-0.040710	0.049703
std	47314.564520	2.121696	1.935197	1.517273	1.490202
min	172.000000	-40.470142	-37.520432	-17.474421	-3.559353
25%	55779.250000	-0.921363	-0.622638	-0.886693	-0.831241
50%	87777.500000	0.015252	0.076608	0.171572	0.033485
75%	138647.000000	1.296679	0.843327	0.969347	0.793492
max	172704.000000	2.312894	6.762856	3.428548	11.427809

- In additional to calculating the above statistics for the whole sample set, can also calculating for each target set (i.e. fraud and non-fraud cases) to observe the differences between the target sets (e.g. different average age)

EDA – Step 2: Univariate Analysis

Objective

- Gain a better or deeper understanding of each attribute
- Identify combinations of attributes for subsequent analysis
- Identify “noise” in the dataset, e.g. missing values, outliers
- Discretize continuous variables, i.e. convert continuous variables into categorical variables

Explore what?

- Centrality: i.e. mean, median, mode
- Dispersion: i.e. range, variance, standard deviation, skewness and kurtosis

Techniques

- Visualization tools, e.g. histogram, boxplot, scatter-plot

Visualization

- Descriptive statistics sometimes may be difficult to interpret, visual plots of the distributions of the involved variables will give insights about the data and problem
- Missing data, outliers, and distribution can be more easily identified using visualization of the data
- Visualization might include plotting of histograms, box plots, etc.

EDA – visualization example

- Pie Charts – represents a variable's distribution
- Histogram – visualize the central tendency and variability of the data
- Scatter plot – visualize the correlation patterns in data
- …

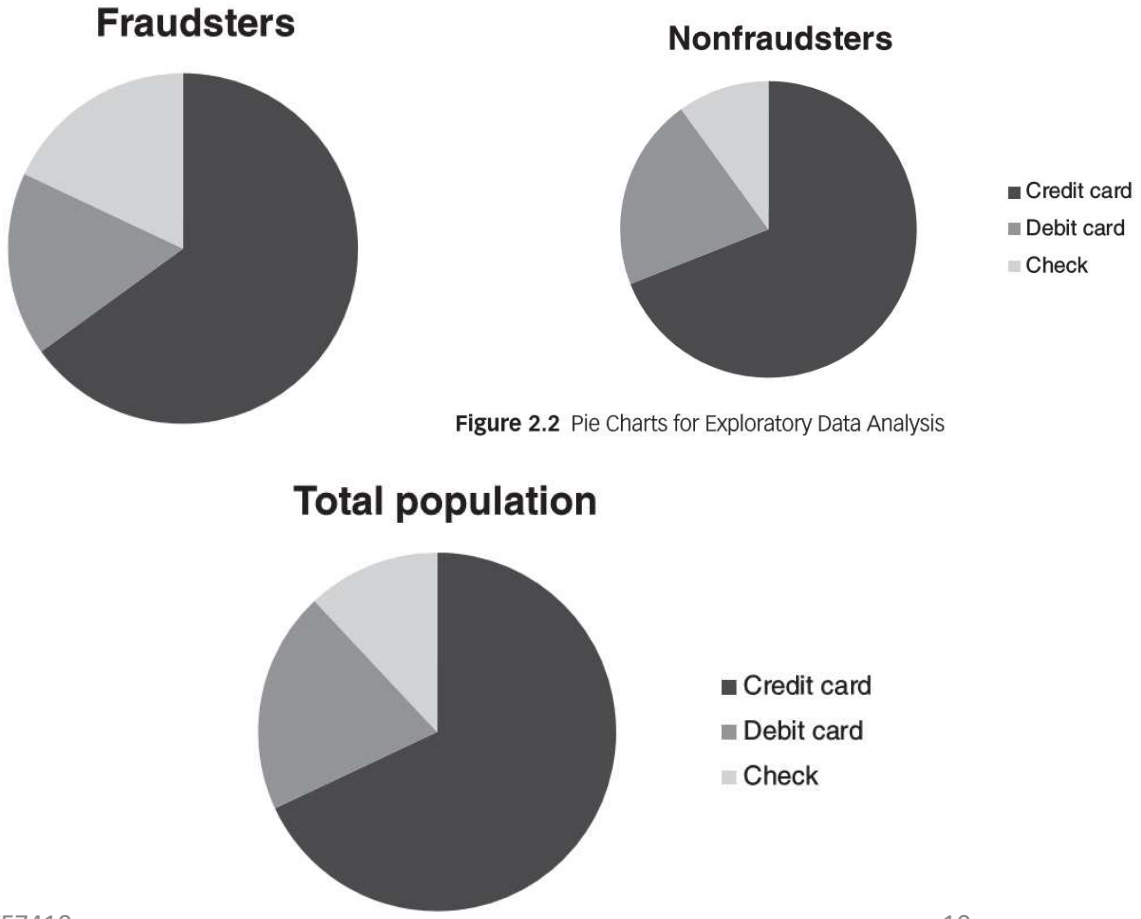


Figure 2.2 Pie Charts for Exploratory Data Analysis

EDA – visualization example

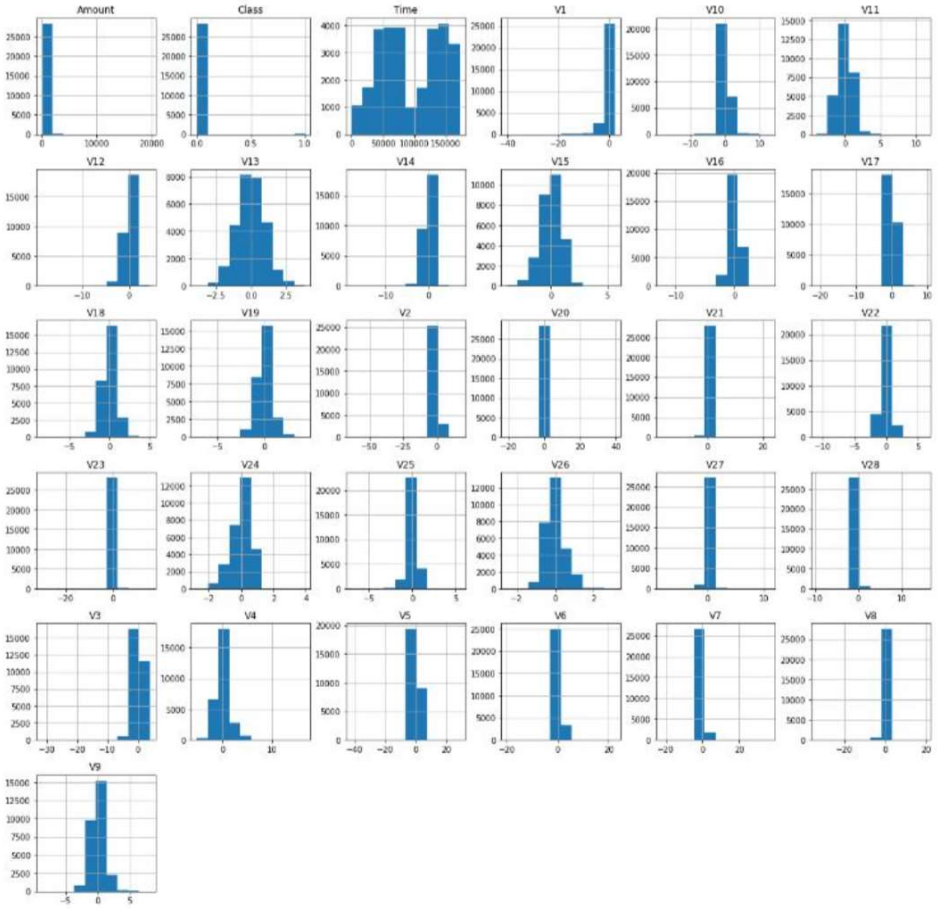


Figure 1: Histogram Showing Dataset Features

EDA – Step 3: Bi-/Multi-variate Analysis

Objective

- To detect interactions among attributes
- Understand relations among the attributes
- Identify incompatibilities among attribute values
- Generate optimal feature combinations

Explore what?

- Bivariate analysis : analyses the associations between 2 attributes only
- Multivariate analysis : analyses the associations among more than 2 attributes

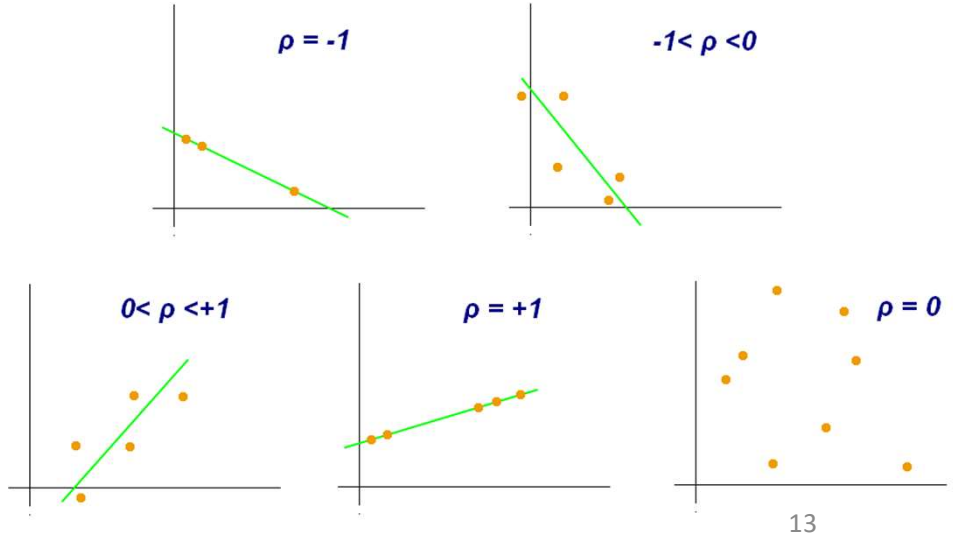
Techniques

- Visualization tools, e.g. histogram, boxplot, scatter-plot, heatmap
- Correlation analysis, e.g. Pearson correlation
- Performs bivariate analysis BEFORE multivariate analysis. With bivariate analysis, will have a clear idea of the compatibility of an attribute pair. Then, can combine more attributes with pairs for further analysis

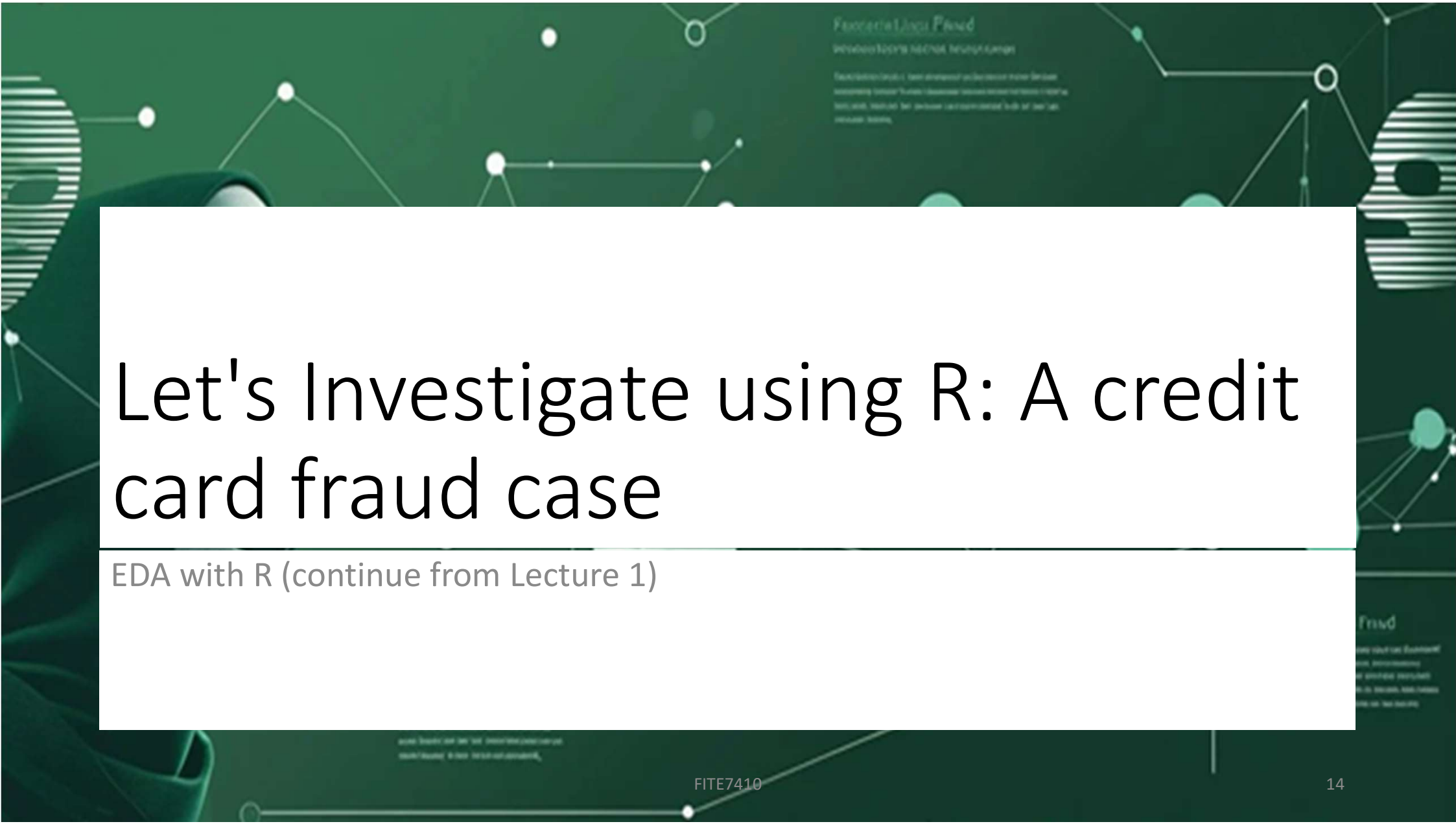
Correlation Analysis

- It is a powerful technique that helps you understand the relations between different variables, i.e. whether the variables are positively or negatively correlated
- Pearson correlation is most commonly used
  - the covariance of the two variables divided by the product of their standard deviations
  - If the coefficient lies between ±0.5 to ±1.0, it is highly correlated
  - If the coefficient is below ±0.29, the correlation is low

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$



Source: Wikipedia



Let's Investigate using R: A credit card fraud case

EDA with R (continue from Lecture 1)

EDA with R

- The Scenario: "We've just been handed a dataset of credit card transactions from a European bank. We know there's fraud in it, but we don't know where or what it looks like. Our job is to find the first clues."
- Our Guiding Questions:
  - What does a "normal" transaction look like?
  - Are there any obvious anomalies or outliers?
  - Do fraudulent transactions behave differently from normal ones?
- The Tool: We will use R to ask these questions
  - Data visualization with R - *Rob Kabacoff* (2020)
  - https://rkabacoff.github.io/datavis/index.html

Example of EDA using R

- Dataset : https://www.kaggle.com/mlg-ulb/creditcardfraud
- The Credit Card Fraud Detection Dataset comprises transactions that European credit card holders made in September 2013. The dataset shows transactions that occurred in two days.
- The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

What are the goals of EDA?

How to achieve the goals?

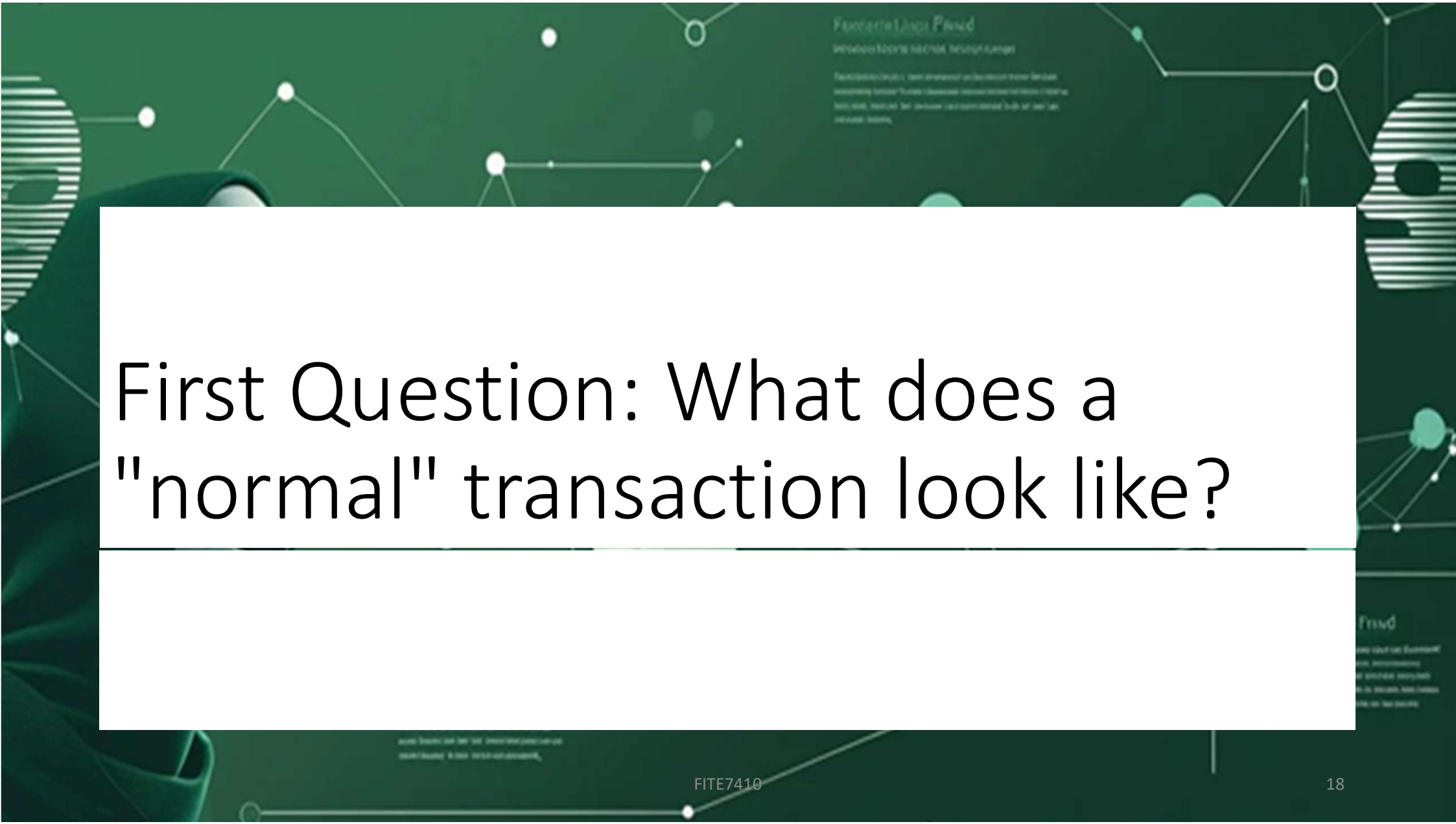
Example – Loading packages and libraries

```
#Loading packages and libraries

library(ggplot2) #visualization library
library(ggcorrplot) #correlation graph library

library(tidyverse) # metapackage of all tidyverse packages
```

#library for plotting the samples

```
#load csv dataset
data <- read.csv('../input/creditcardfraud/creditcard.csv')
```

Path of the source file



First Question: What does a "normal" transaction look like?

Example - Distinguish Attributes

```
#show structure of the dataset
print("Structure of dataset")
str(data)
```

```
[1] "Structure of dataset"
'data.frame':   284807 obs. of  31 variables:
 $ Time  : num  0 0 1 1 2 2 4 7 7 9 ...
 $ V1    : num  -1.36 1.192 -1.358 -0.966 -1.158 ...
 $ V2    : num  -0.0728 0.2662 -1.3402 -0.1852 0.8777 ...
 $ V3    : num  2.536 0.166 1.773 1.793 1.549 ...
 $ V4    : num  1.378 0.448 0.38 -0.863 0.403 ...
 $ V5    : num  -0.3383 0.06 -0.5032 -0.0103 -0.4072 ...
 $ V6    : num  0.4624 -0.0824 1.8005 1.2472 0.0959 ...
 $ V7    : num  0.2396 -0.0788 0.7915 0.2376 0.5929 ...
 $ V8    : num  0.0987 0.0851 0.2477 0.3774 -0.2705 ...
 $ V9    : num  0.364 -0.255 -1.515 -1.387 0.818 ...
 $ V10   : num  0.0908 -0.167 0.2076 -0.055 0.7531 ...
 $ V11   : num  -0.552 1.613 0.625 -0.226 -0.823 ...
 $ V12   : num  -0.6178 1.0652 0.0661 0.1782 0.5382 ...
 $ V13   : num  -0.991 0.489 0.717 0.508 1.346 ...
 $ V14   : num  -0.311 -0.144 -0.166 -0.288 -1.12 ...
 $ V15   : num  1.468 0.636 2.346 -0.631 0.175 ...
 $ V16   : num  -0.47 0.464 -2.89 -1.06 -0.451 ...
 $ V17   : num  0.208 -0.115 1.11 -0.684 -0.237 ...
 $ V18   : num  0.0258 -0.1834 -0.1214 1.9658 -0.0382 ...
 $ V19   : num  0.404 -0.146 -2.262 -1.233 0.803 ...
 $ V20   : num  0.2514 -0.0691 0.525 -0.208 0.4085 ...
 $ V21   : num  -0.01831 -0.22578 0.248 -0.1083 -0.00943 ...
 $ V22   : num  0.27784 -0.63867 0.77168 0.00527 0.79828 ...
 $ V23   : num  -0.11 0.101 0.909 -0.19 -0.137 ...
 $ V24   : num  0.0669 -0.3398 -0.6893 -1.1756 0.1413 ...
 $ V25   : num  0.129 0.167 -0.328 0.647 -0.206 ...
 $ V26   : num  -0.189 0.126 -0.139 -0.222 0.502 ...
 $ V27   : num  0.13356 -0.00898 -0.05535 0.06272 0.21942 ...
 $ V28   : num  -0.0211 0.0147 -0.0598 0.0615 0.2152 ...
 $ Amount: num  149.62 2.69 378.66 123.5 69.99 ...
 $ Class : int  0 0 0 0 0 0 0 0 0 0 ...
```

Example - Distinguish Attributes

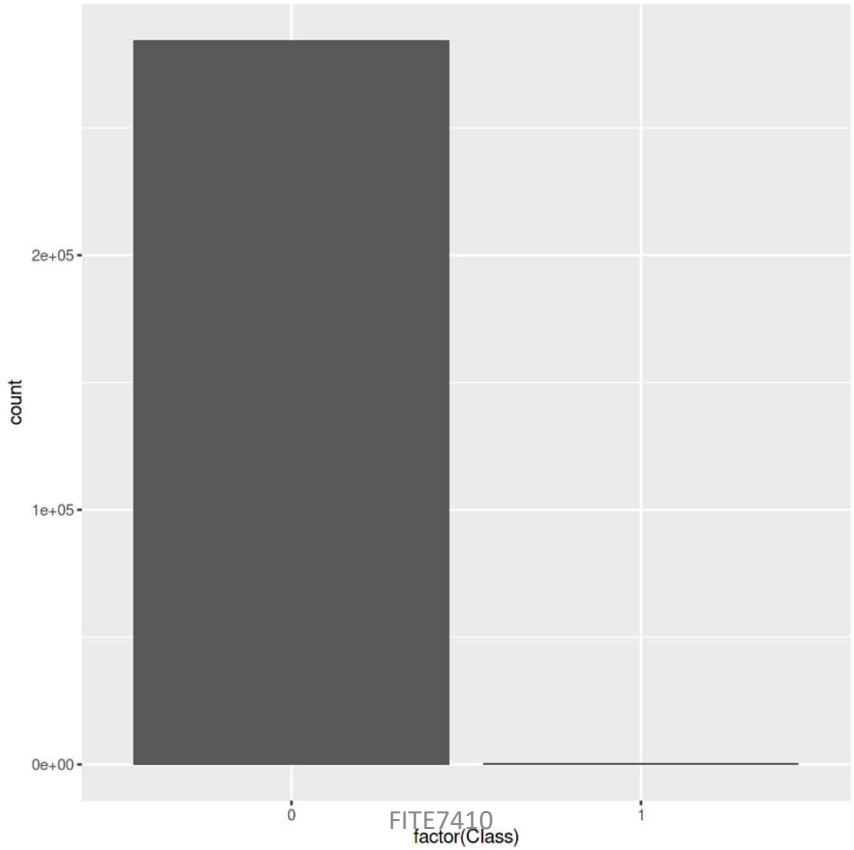
```
#show summary statistics of the dataset
print("Summary statistics")
summary(data)
```

```
[1] "Summary statistics"
      Time              V1                   V2                   V3
 Min.   :     0   Min.   :-56.40751   Min.   :-72.71573   Min.   :-48.3256
 1st Qu.: 54202   1st Qu.: -0.92037   1st Qu.: -0.59855   1st Qu.: -0.8904
 Median : 84692   Median :  0.01811   Median :  0.06549   Median :  0.1799
 Mean   : 94814   Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.0000
 3rd Qu.:139320   3rd Qu.:  1.31564   3rd Qu.:  0.80372   3rd Qu.:  1.0272
 Max.   :172792   Max.   :  2.45493   Max.   : 22.05773   Max.   :  9.3826
       V4                   V5                   V6                   V7
 Min.   :-5.68317   Min.   :-113.74331   Min.   :-26.1605   Min.   :-43.5572
 1st Qu.:-0.84864   1st Qu.:  -0.69160   1st Qu.: -0.7683   1st Qu.: -0.5541
 Median :-0.01985   Median :  -0.05434   Median : -0.2742   Median :  0.0401
 Mean   : 0.00000   Mean   :   0.00000   Mean   :  0.0000   Mean   :  0.0000
 3rd Qu.: 0.74334   3rd Qu.:   0.61193   3rd Qu.:  0.3986   3rd Qu.:  0.5704
 Max.   :16.87534   Max.   :  34.80167   Max.   : 73.3016   Max.   :120.5895
       V8                   V9                  V10                  V11
 Min.   :-73.21672   Min.   :-13.43407   Min.   :-24.58826   Min.   :-4.79747
 1st Qu.: -0.20863   1st Qu.: -0.64310   1st Qu.: -0.53543   1st Qu.:-0.76249
 Median :  0.02236   Median : -0.05143   Median : -0.09292   Median :-0.03276
 Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.00000
 3rd Qu.:  0.32735   3rd Qu.:  0.59714   3rd Qu.:  0.45392   3rd Qu.: 0.73959
 Max.   : 20.00721   Max.   : 15.59500   Max.   : 23.74514   Max.   :12.01891
      V12                  V13                  V14                  V15
 Min.   :-18.6837   Min.   :-5.79188   Min.   :-19.2143   Min.   :-4.49894
 1st Qu.: -0.4056   1st Qu.:-0.64854   1st Qu.: -0.4256   1st Qu.:-0.58288
 Median :  0.1400   Median :-0.01357   Median :  0.0506   Median : 0.04807
 Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.0000   Mean   : 0.00000
 3rd Qu.:  0.6182   3rd Qu.: 0.66251   3rd Qu.:  0.4931   3rd Qu.: 0.64882
 Max.   :  7.8484   Max.   : 7.12688   Max.   : 10.5268   Max.   : 8.87774
      V16                  V17                  V18
 Min.   :-14.12985   Min.   :-25.16280   Min.   :-9.498746
 1st Qu.: -0.46804   1st Qu.: -0.48375   1st Qu.:-0.498850
 Median :  0.06641   Median : -0.06568   Median :-0.003636
 Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.000000
 3rd Qu.:  0.52330   3rd Qu.:  0.39968   3rd Qu.: 0.500807
 Max.   : 17.31511   Max.   :  9.25353   Max.   : 5.041069
      V19                  V20                  V21
 Min.   :-7.213527   Min.   :-54.49772   Min.   :-34.83038
 1st Qu.:-0.456299   1st Qu.: -0.21172   1st Qu.: -0.22839
 Median : 0.003735   Median : -0.06248   Median : -0.02945
 Mean   : 0.000000   Mean   :  0.00000   Mean   :  0.00000
 3rd Qu.: 0.458949   3rd Qu.:  0.13304   3rd Qu.:  0.18638
 Max.   : 5.591971   Max.   : 39.42090   Max.   : 27.20284
      V22                  V23                  V24
 Min.   :-10.933144   Min.   :-44.80774   Min.   :-2.83663
 1st Qu.: -0.542350   1st Qu.: -0.16185   1st Qu.:-0.35459
 Median :  0.006782   Median : -0.01119   Median : 0.04098
 Mean   :  0.000000   Mean   :  0.00000   Mean   : 0.00000
 3rd Qu.:  0.528554   3rd Qu.:  0.14764   3rd Qu.: 0.43953
 Max.   : 10.503090   Max.   : 22.52841   Max.   : 4.58455
      V25                  V26                  V27
 Min.   :-10.29540   Min.   :-2.60455   Min.   :-22.565679
 1st Qu.: -0.31715   1st Qu.:-0.32698   1st Qu.: -0.070840
 Median :  0.01659   Median :-0.05214   Median :  0.001342
 Mean   :  0.00000   Mean   : 0.00000   Mean   :  0.000000
 3rd Qu.:  0.35072   3rd Qu.: 0.24095   3rd Qu.:  0.091045
 Max.   :  7.51959   Max.   : 3.51735   Max.   : 31.612198
      V28                 Amount              Class
 Min.   :-15.43008   Min.   :    0.00   Min.   :0.000000
 1st Qu.: -0.05296   1st Qu.:    5.60   1st Qu.:0.000000
 Median :  0.01124   Median :   22.00   Median :0.000000
 Mean   :  0.00000   Mean   :   88.35   Mean   :0.001728
 3rd Qu.:  0.07828   3rd Qu.:   77.17   3rd Qu.:0.000000
 Max.   : 33.84781   Max.   :25691.16   Max.   :1.000000
```

Example –Univariate Analysis-”Class”

** Examples of bar charts

```
#Bar chars plotting using ggplot and geom_bar()
ggplot(data, aes(x = factor(Class))) +
       geom_bar()
```



ggplot() is used to construct the initial plot object, and is almost always followed by + to add component to the plot.

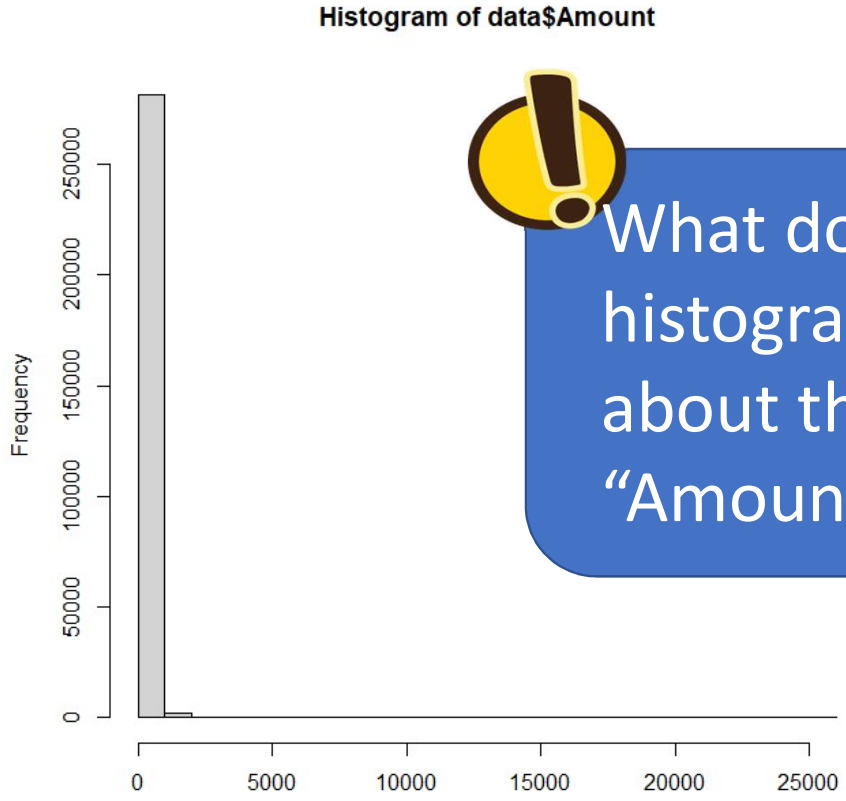
What does this histogram tell you about the variable “Class”?

Example –Univariate Analysis “Amount”

```
help(hist)
```

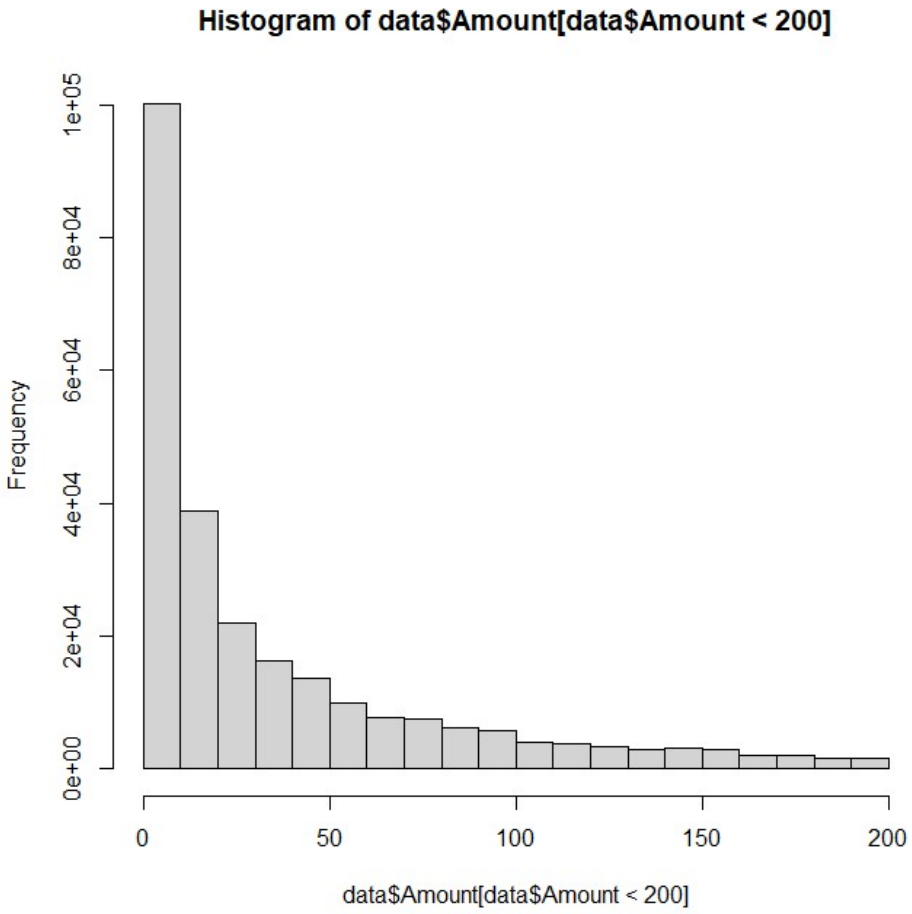
Use “help” to check the R documentation

```
#Histogram
hist(data$Amount)
```



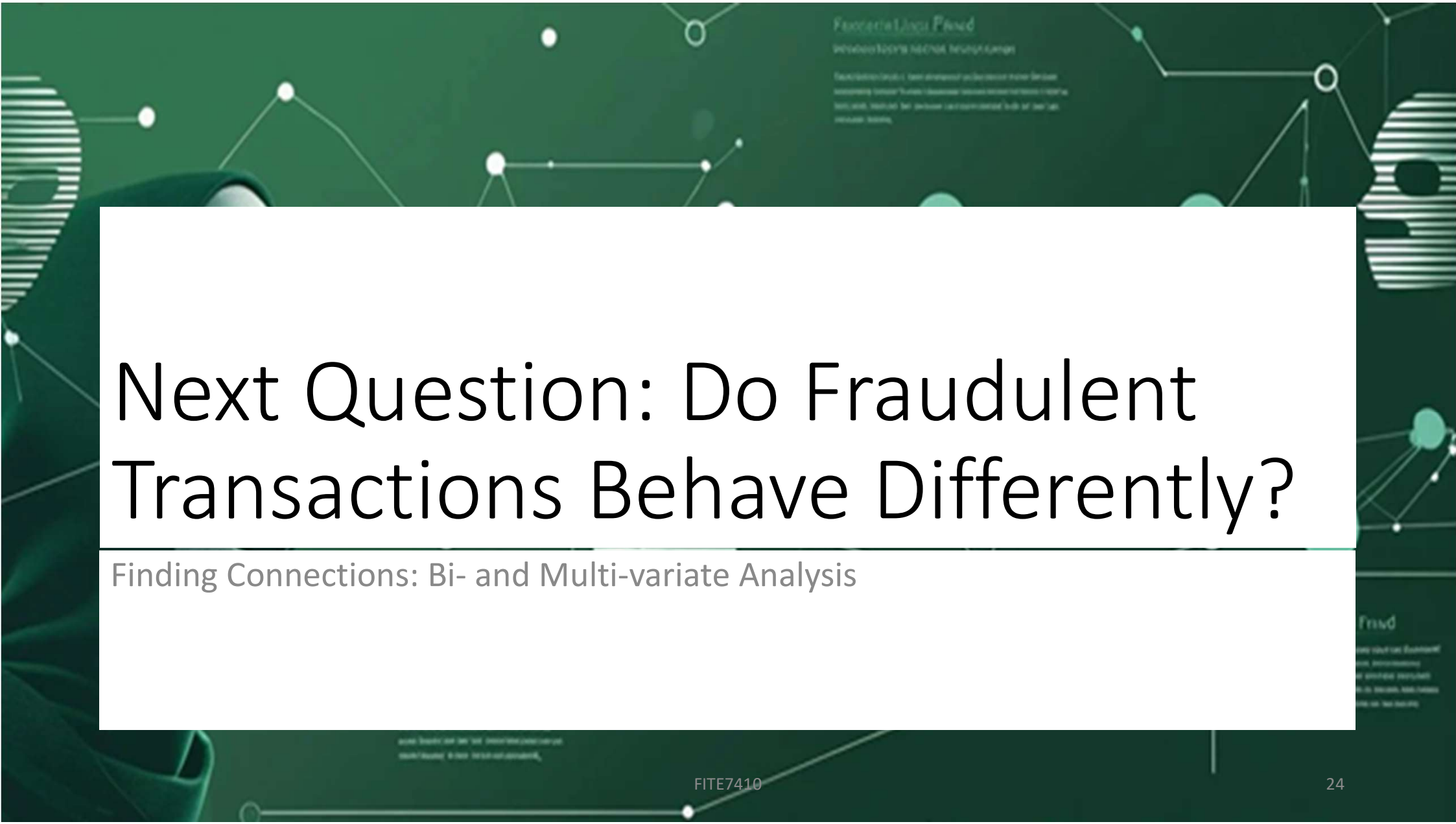
What does this histogram tell you about the variable “Amount”?

Example –Univariate Analysis-”Amount”




```r
#Plot Histogram for Amount smaller than $200
hist(data$Amount[data$Amount < 200])
```

Now, with the “Amount” limited to under $200, what does this histogram tell you about the variable “Amount”?

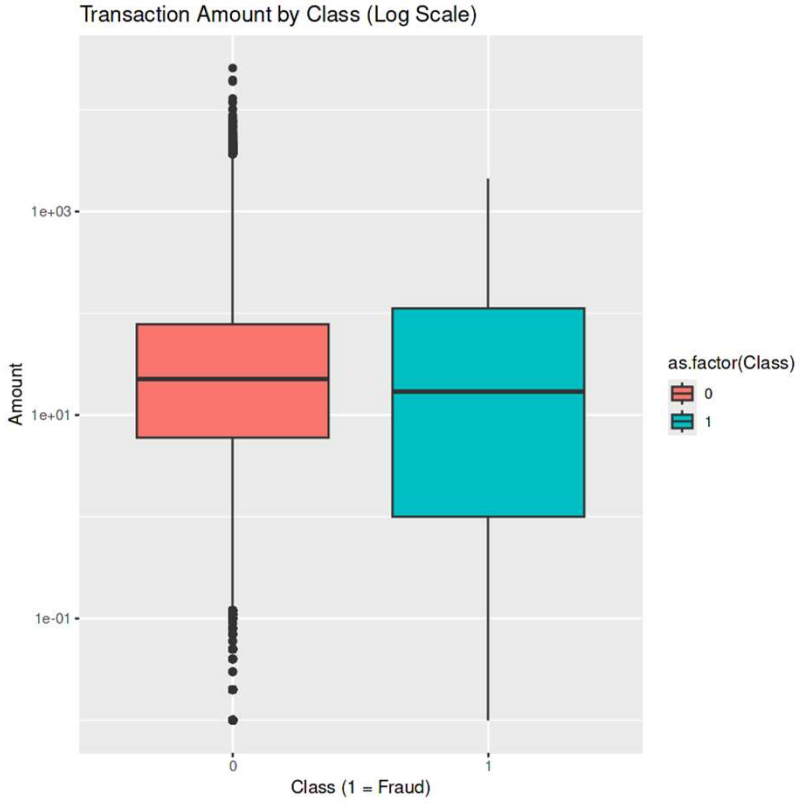


Next Question: Do Fraudulent Transactions Behave Differently?

Finding Connections: Bi- and Multi-variate Analysis

Example –Bi-/Multi-variate Analysis

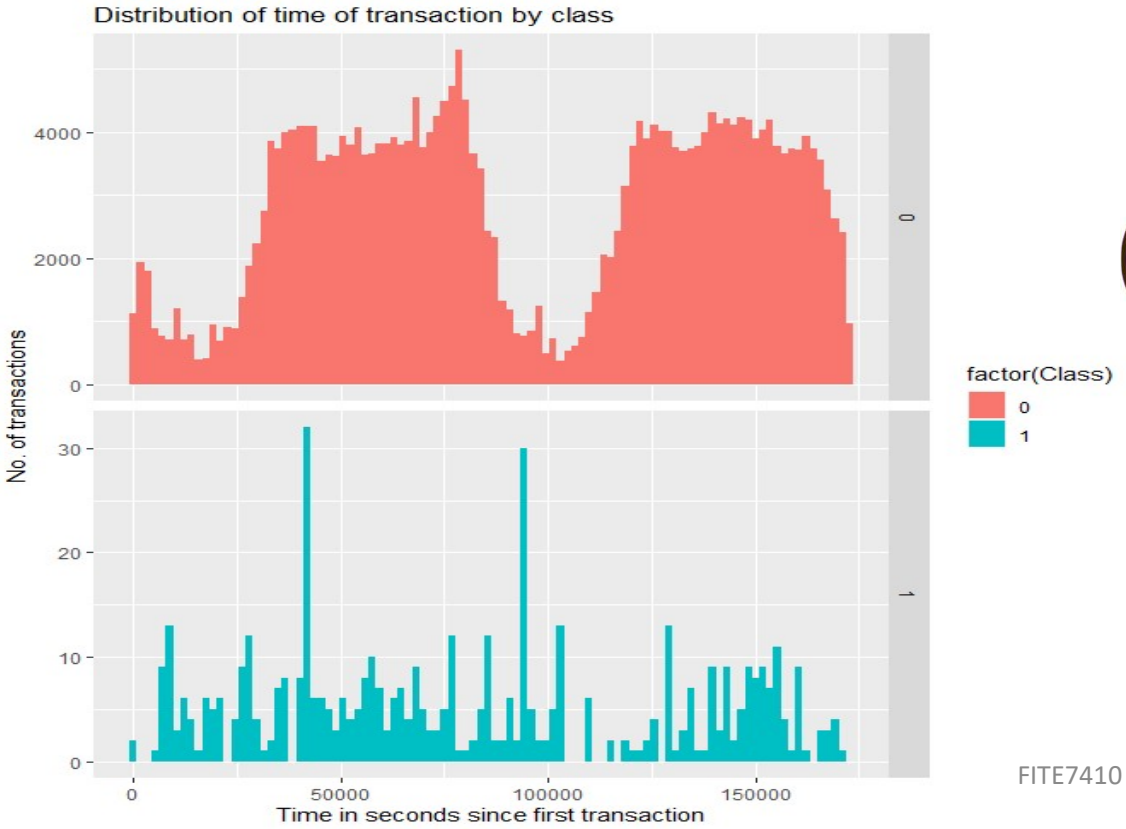
```
# Boxplot comparing amounts by class
ggplot(data, aes(x = as.factor(Class), y = Amount, fill = as.factor(Class))) +
  geom_boxplot() +
  scale_y_log10() + # Use a log scale to see distributions clearly
  labs(title = "Transaction Amount by Class (Log Scale)", x = "Class (1 = Fraud)", y = "Amount")
```



What insights can you get from this figure?

Example –Bi-/Multi-variate Analysis

```
#Histogram plot of variable Time and Class label
ggplot(data, aes(x = Time, fill = factor(Class))) +
    geom_histogram(bins = 100) +
    labs(x = 'Time in seconds since first transaction', y = 'No. of transactions') +
    ggtitle('Distribution of time of transaction by class') +
    facet_grid(Class ~ ., scales = 'free_y')
```



What insights can you get from this figure?

Example –Bi-/Multi-variate Analysis

** Examples of plotting correlations

Step 1

```
#Compute the correlations among the variables

# select numeric variables
df <- dplyr::select_if(data, is.numeric)

# calulate the correlations
corr_mat <- cor(df, use="complete.obs")
round(corr_mat,2)
```

package::functionname
i.e. use the “dplyr” package and function call is “select_if()”

Can be rewritten as follows:

```
library(dplyr)
df <- select_if(data, is.numeric)
```

Example –Bi-/Multi-variate Analysis

** Examples of plotting correlations

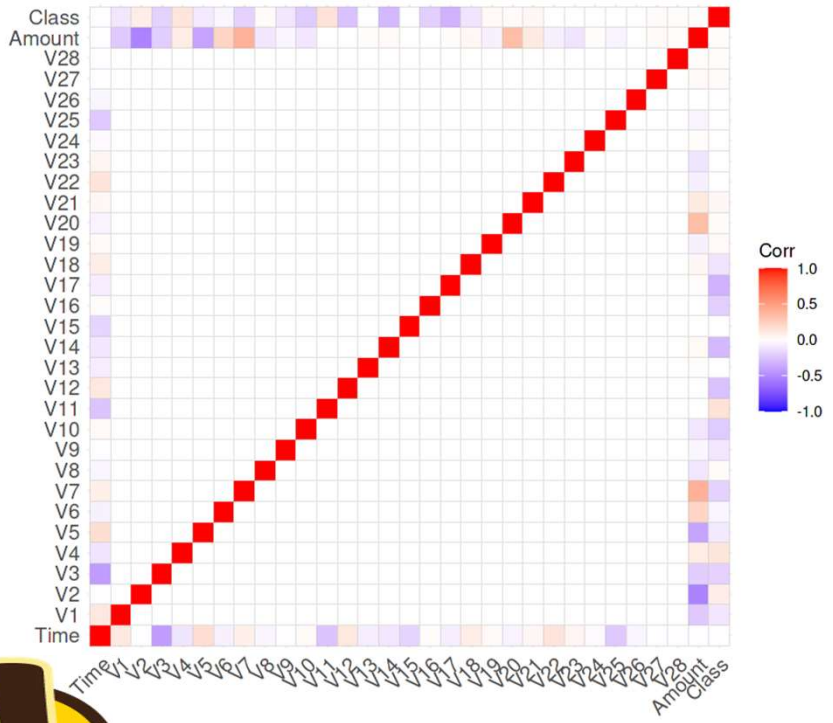
Step 2

```
#Plot correlation heat map

ggcorrplot(corr_mat)
ggcorrplot(corr_mat, lab = TRUE)
```

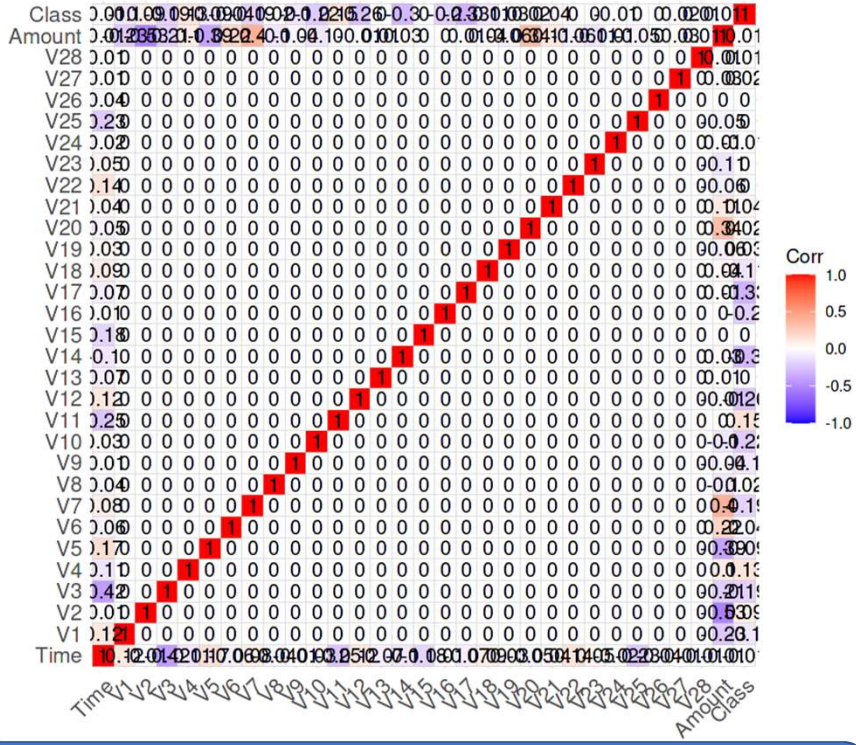
lab = TRUE overlays the correlation coefficients (as text) on the plot

Example –Bi-/Multi-variate Analysis

ggcorrplot(corr_mat)



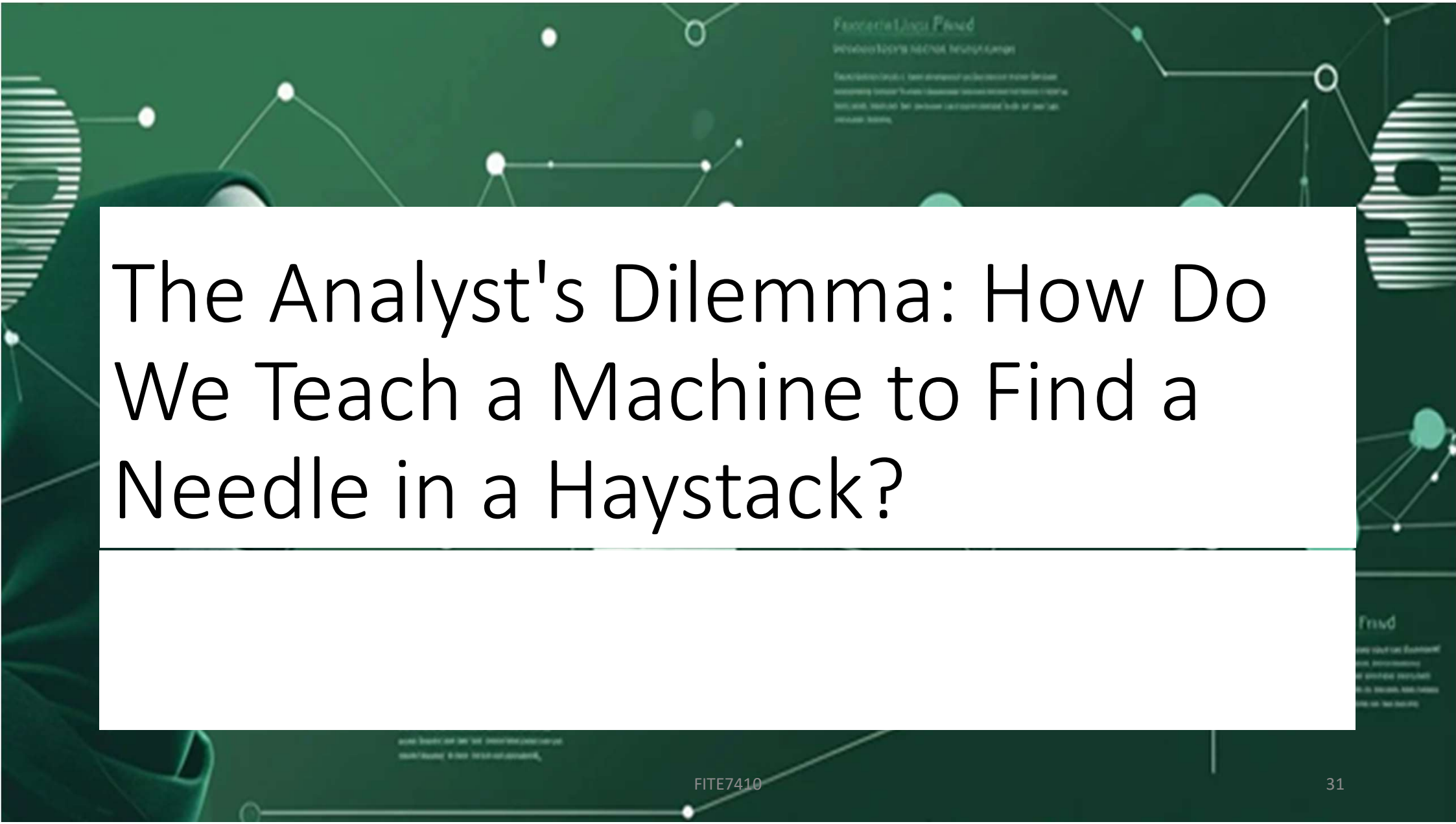
ggcorrplot(corr_mat, lab = TRUE)



What insights can you get from the correlations among the attributes?

What We Found in the initial EDA findings?

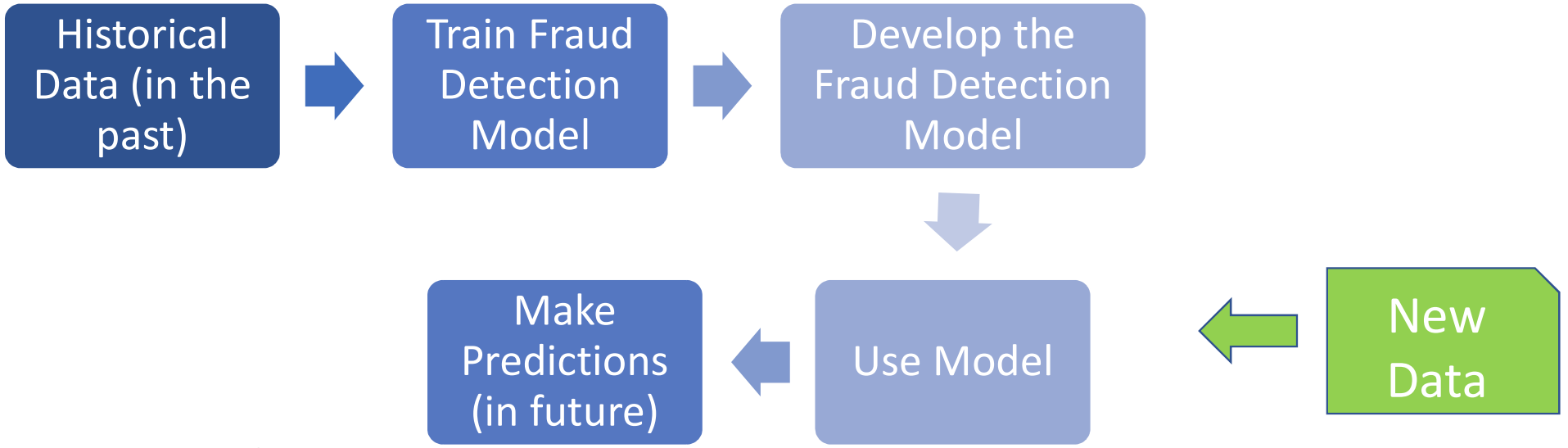
- Finding 1 (Univariate):
  - Transaction amounts are highly skewed, with many small transactions and a few very large outliers. The outliers are our first area of interest.
- Finding 2 (Bivariate):
  - Contrary to intuition, fraudulent transactions tend to be smaller on average than legitimate ones, suggesting fraudsters are trying to avoid detection.
- Finding 3
  - Most importantly, the dataset is severely imbalanced. Legitimate transactions vastly outnumber fraudulent ones (as we will see), making it nearly impossible for a model to learn without help.



The Analyst's Dilemma: How Do We Teach a Machine to Find a Needle in a Haystack?

Financial Fraud Analytics Model

Fraud Detection Model

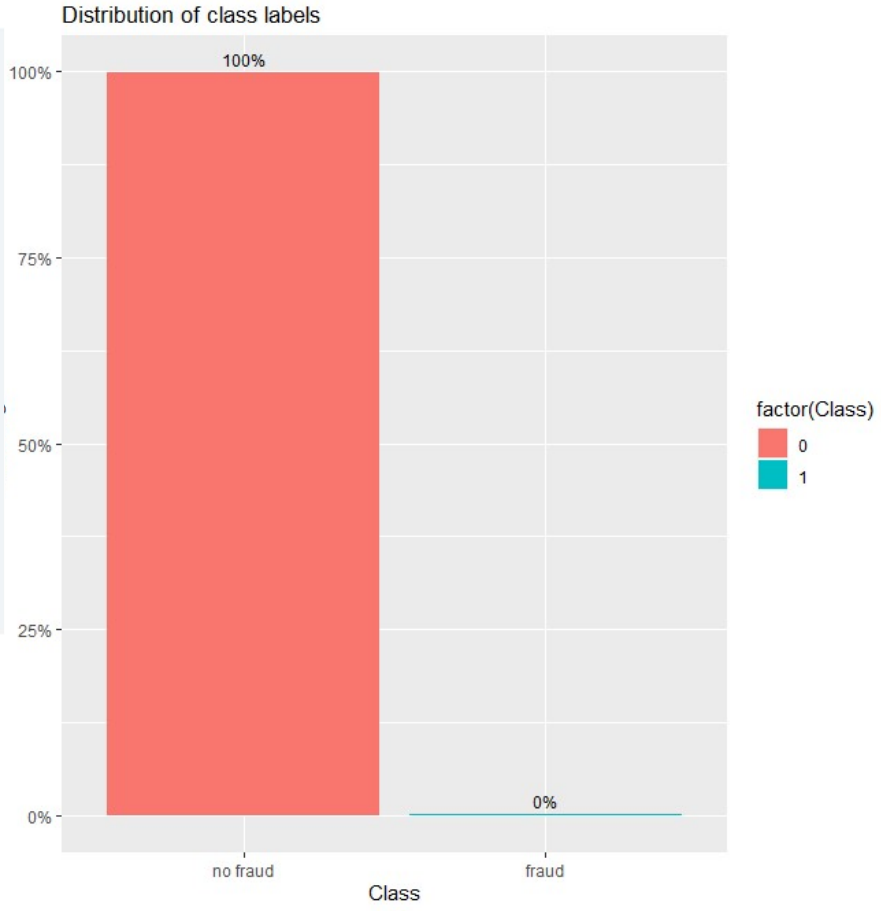


Frequency of re-training the model depends on:

- Volatility of the fraud behaviour
- Detection power of the current model
- Amount of (similar) confirmed cases already available in the database
- Rate at which new cases are being confirmed
- Required effort to retrain the model

Example – Univariate Analysis-"Class"

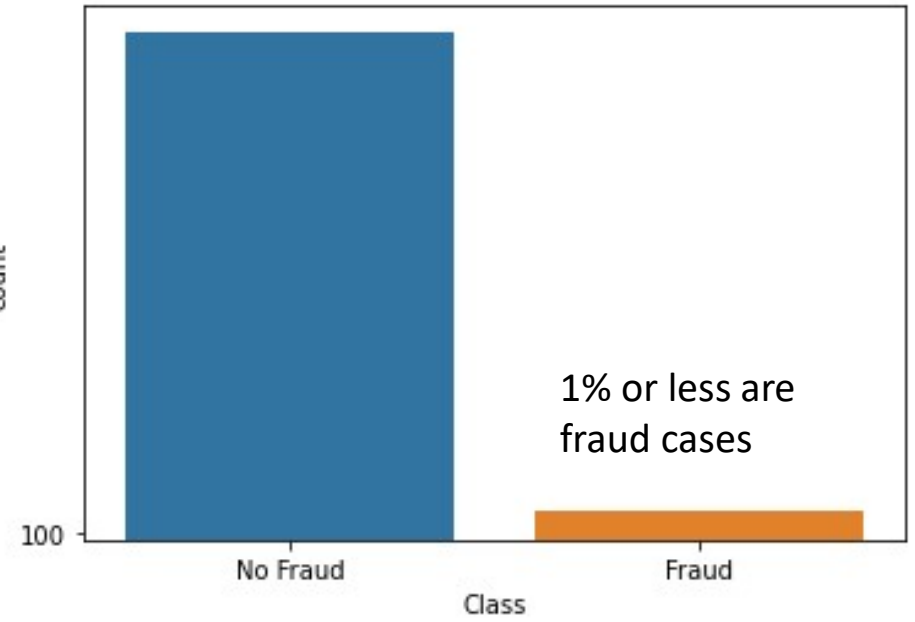
```r
#Bar charts

ggplot(data, aes(x=factor(Class),
                 y = prop.table(after_stat(count)), fill=factor(Class),
                 label = scales::percent(prop.table(after_stat(count))))) +
       geom_bar(position = "dodge") +
       geom_text(stat = 'count',
                 position = position_dodge(.9),
                 vjust = -0.5,
                 size = 3) +
       scale_x_discrete(labels = c("no fraud","fraud")) +
       scale_y_continuous(labels = scales::percent) +
       labs(x = 'Class', y = 'Percentage') +
       ggtitle("Distribution of class labels")
```



What is the problem with this dataset, if we wish to train a model on this?

What is imbalance dataset?

- Imbalance dataset also known as skewed dataset
- Imbalanced datasets are a special case for classification problem where the class distribution is not uniform among the classes.
- Typically, they are composed by two classes: The majority class and the minority class.



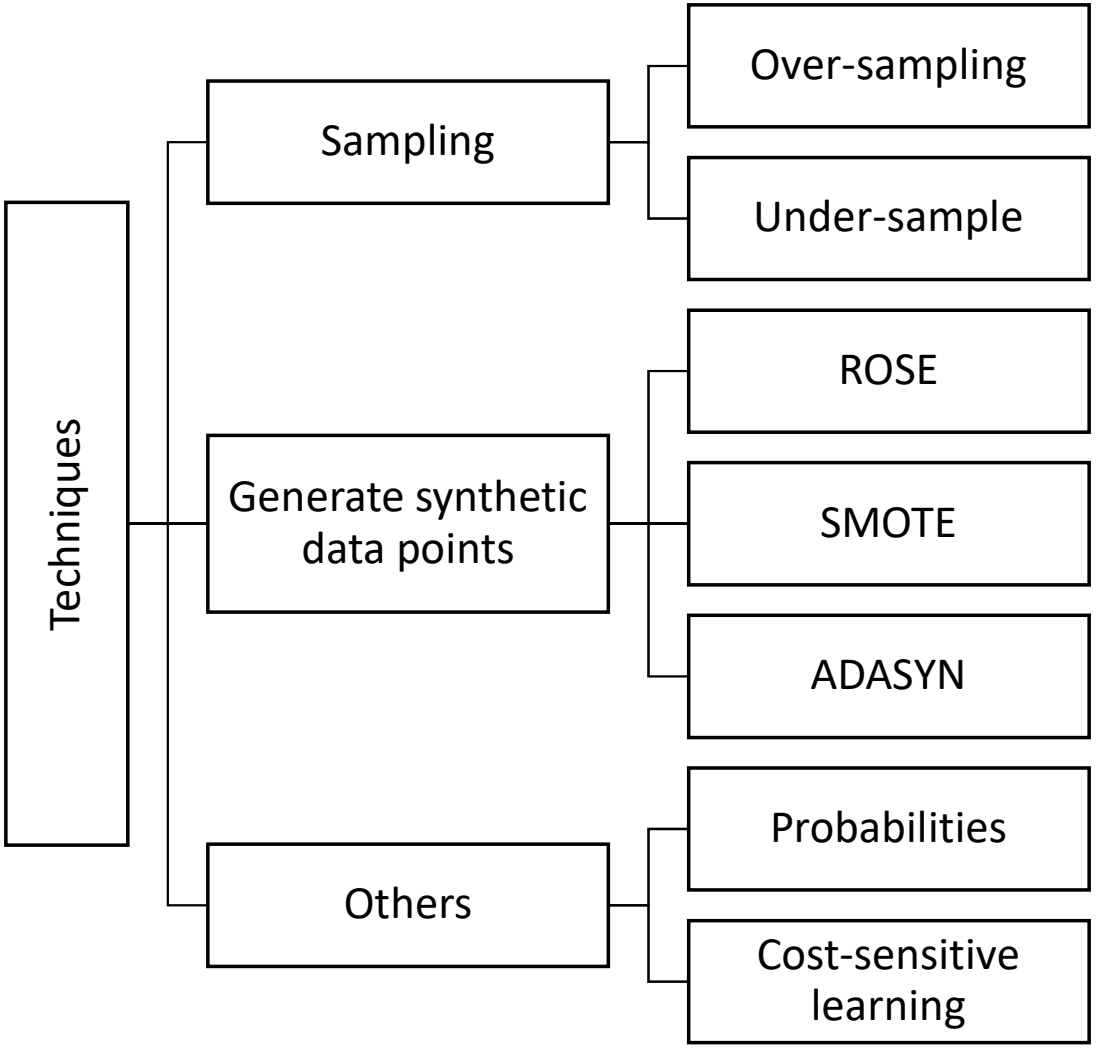
What are the problems of imbalance dataset?

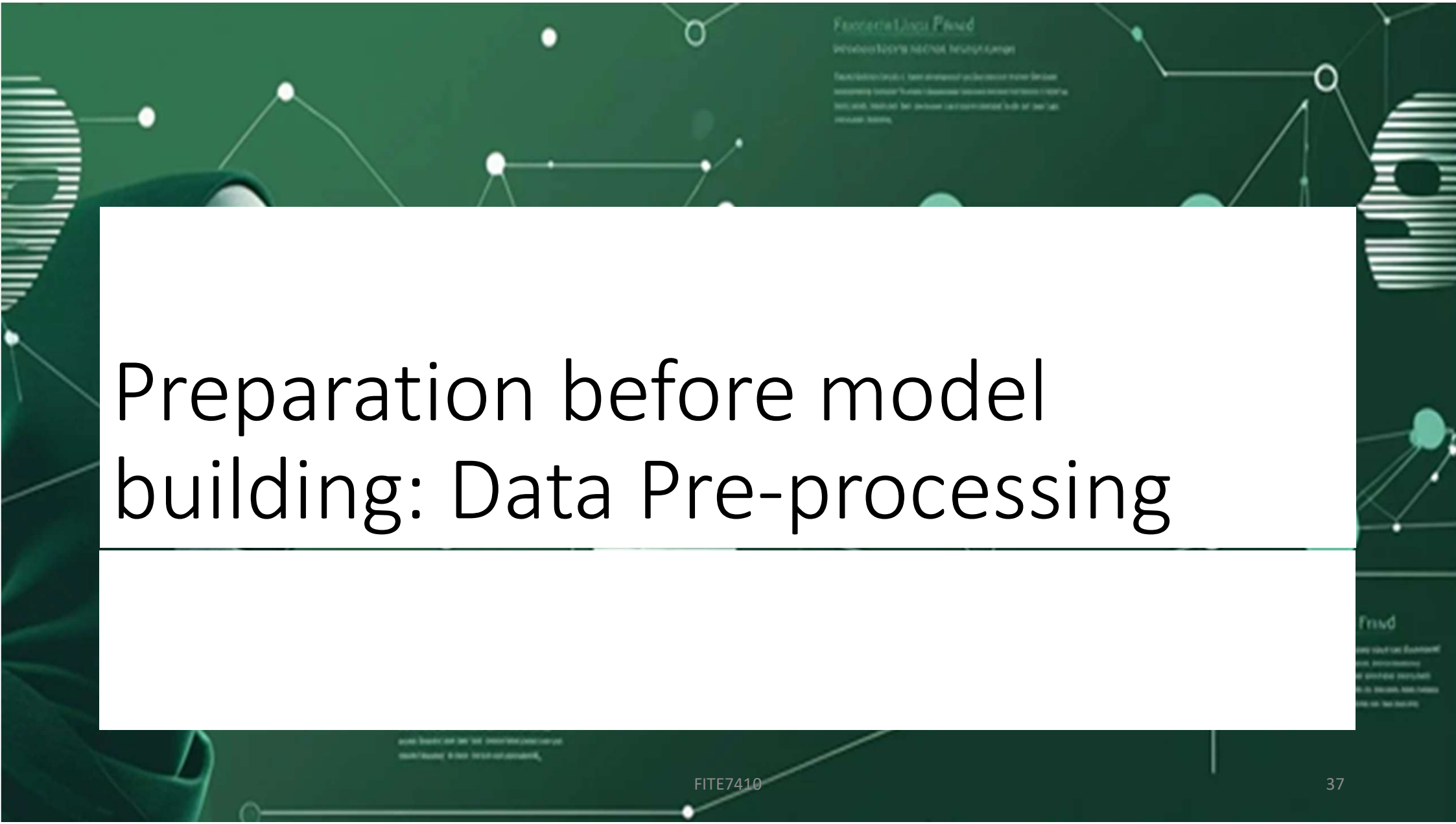
What are the problems?

- Problems of imbalance dataset in machine learning:
  - Most machine learning models assume an equal distribution of classes
  - A model may focus on learning the characteristics of majority class due to the abundance of samples available for learning
  - Many machine learning models will show bias towards majority class, leading to incorrect conclusions

- Slight imbalance vs Severe imbalance
  - If the data set is only slightly imbalance (e.g. ratio of 4:6), can still be used for training

How do we force the model to pay attention to the rare fraud cases?

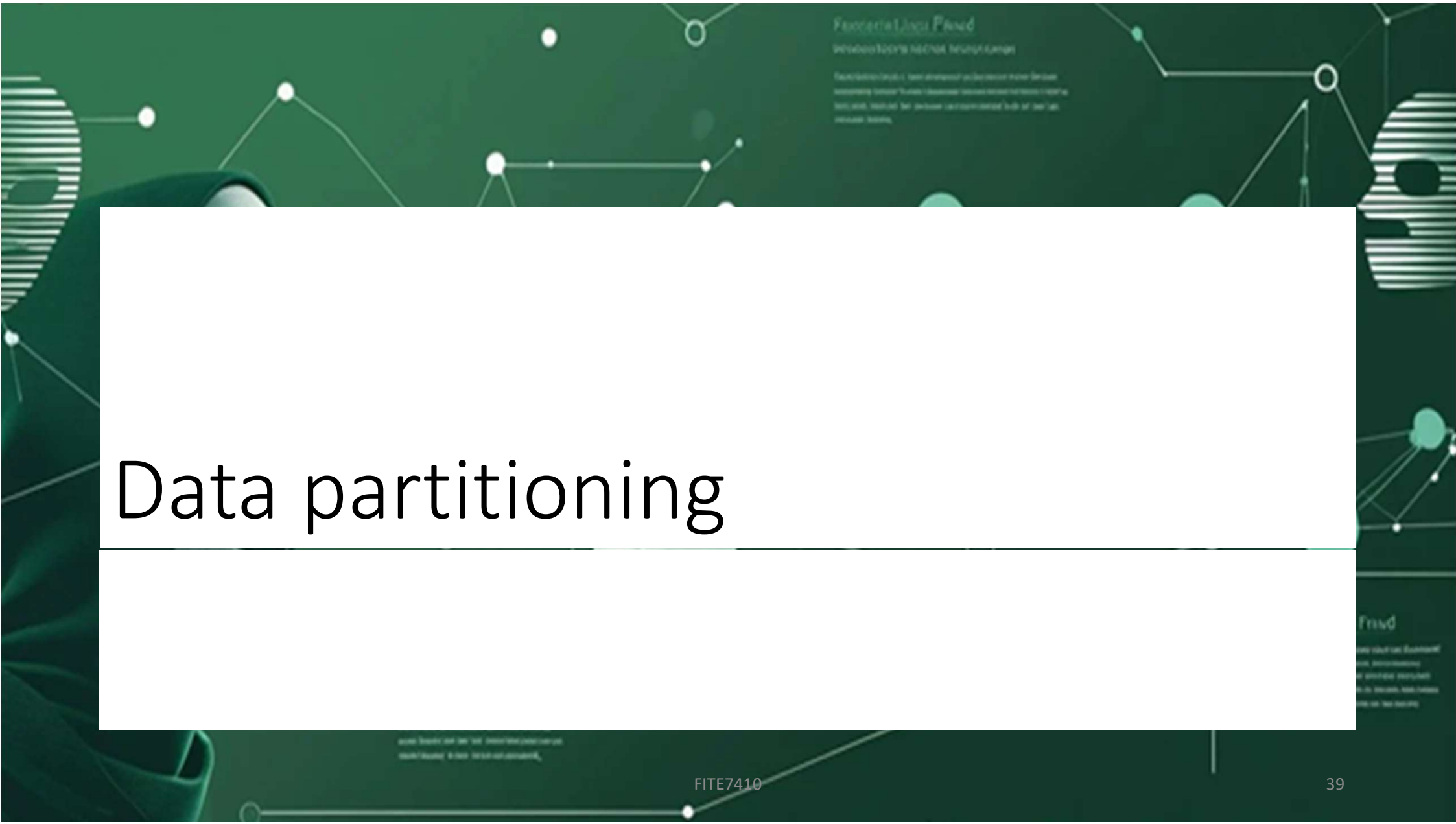




Preparation before model building: Data Pre-processing

Data pre-processing

1. **Data Cleaning**: Handling missing values and addressing outliers.
2. **Data Partitioning**: Splitting the data into training, validation, and test sets.
3. **Handling Imbalance**: Applying techniques like SMOTE or undersampling.
4. **Feature Engineering**: Creating new, more informative variables.
5. **Feature Scaling**: Standardizing or normalizing features to a common scale.



Data partitioning

Sample Data Set

- Split the sample data set into 2-3 datasets

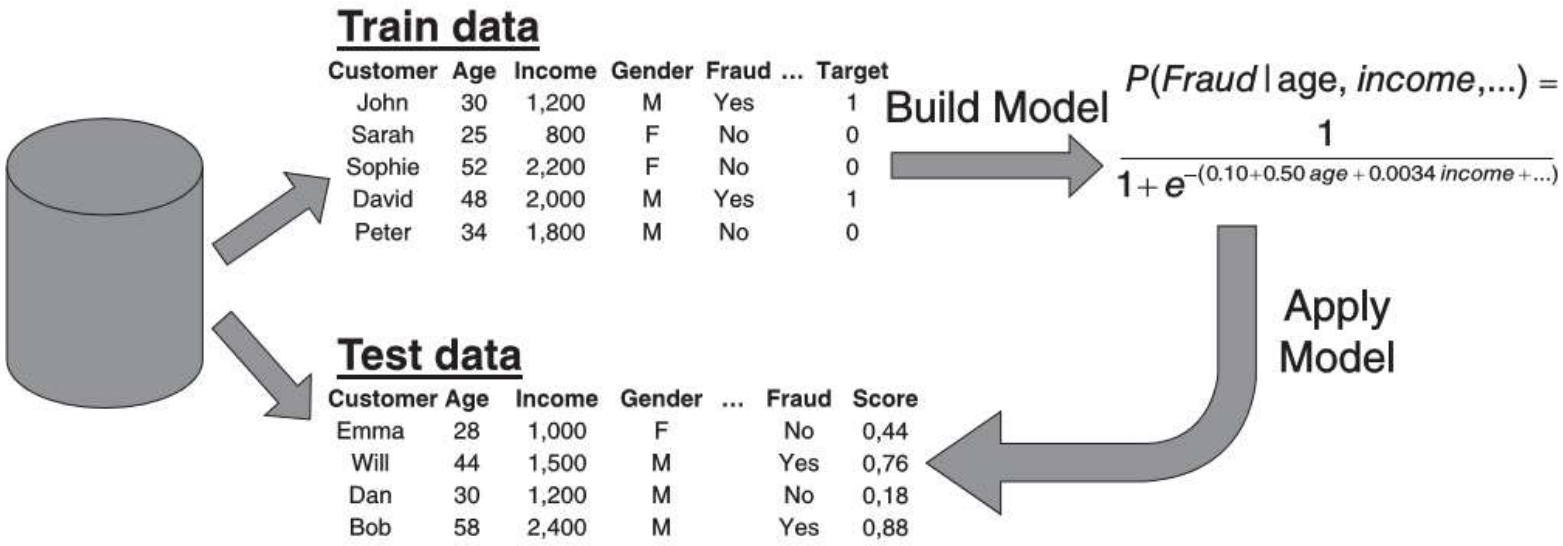


Figure 4.34 Training Versus Test Sample Set Up for Performance Estimation

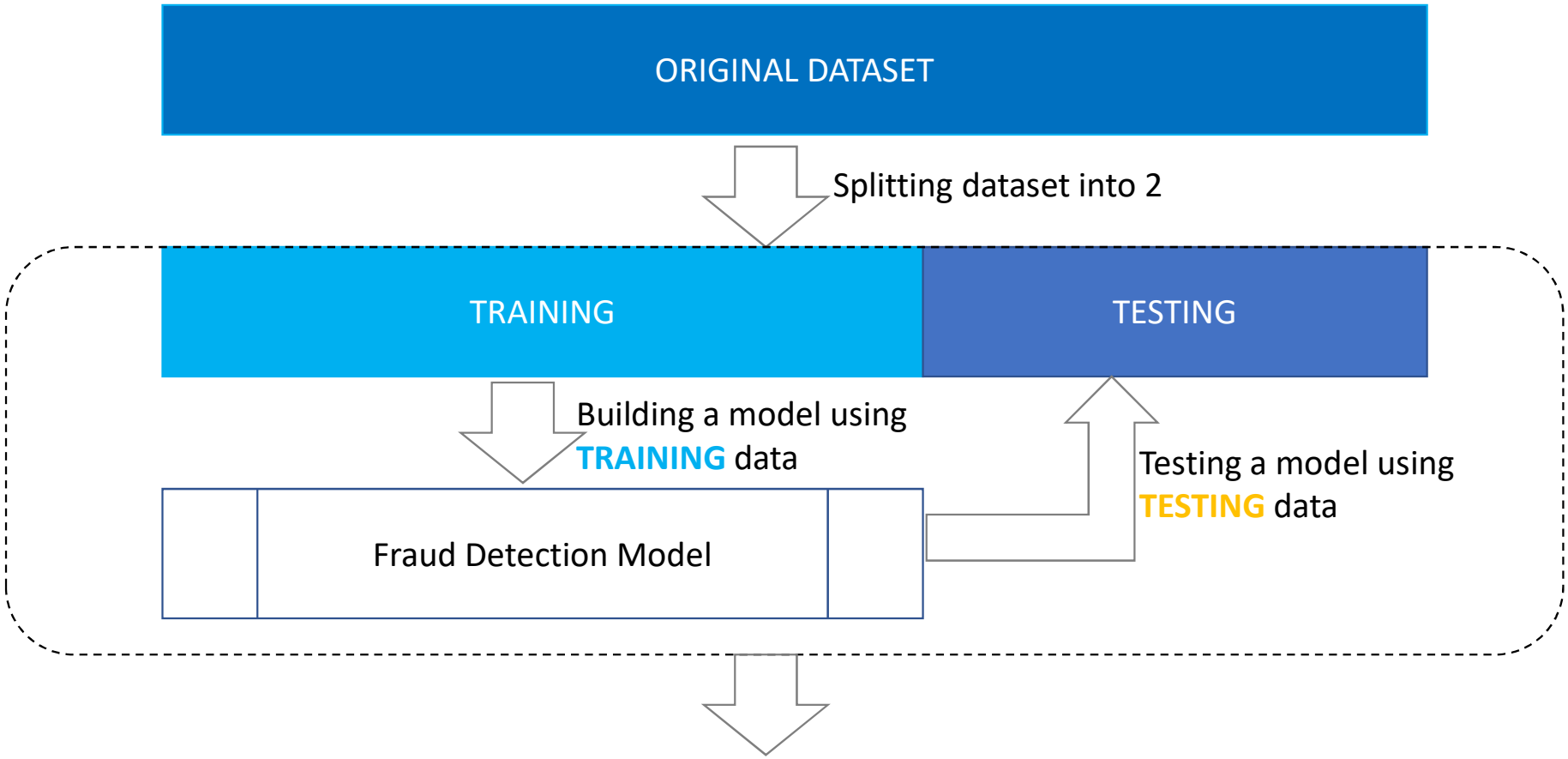
Source: Fraud Analytics using Descriptive, Predictive, and Social Network Techniques, 1st ed, John Wiley & Sons Inc.

Splitting the data set

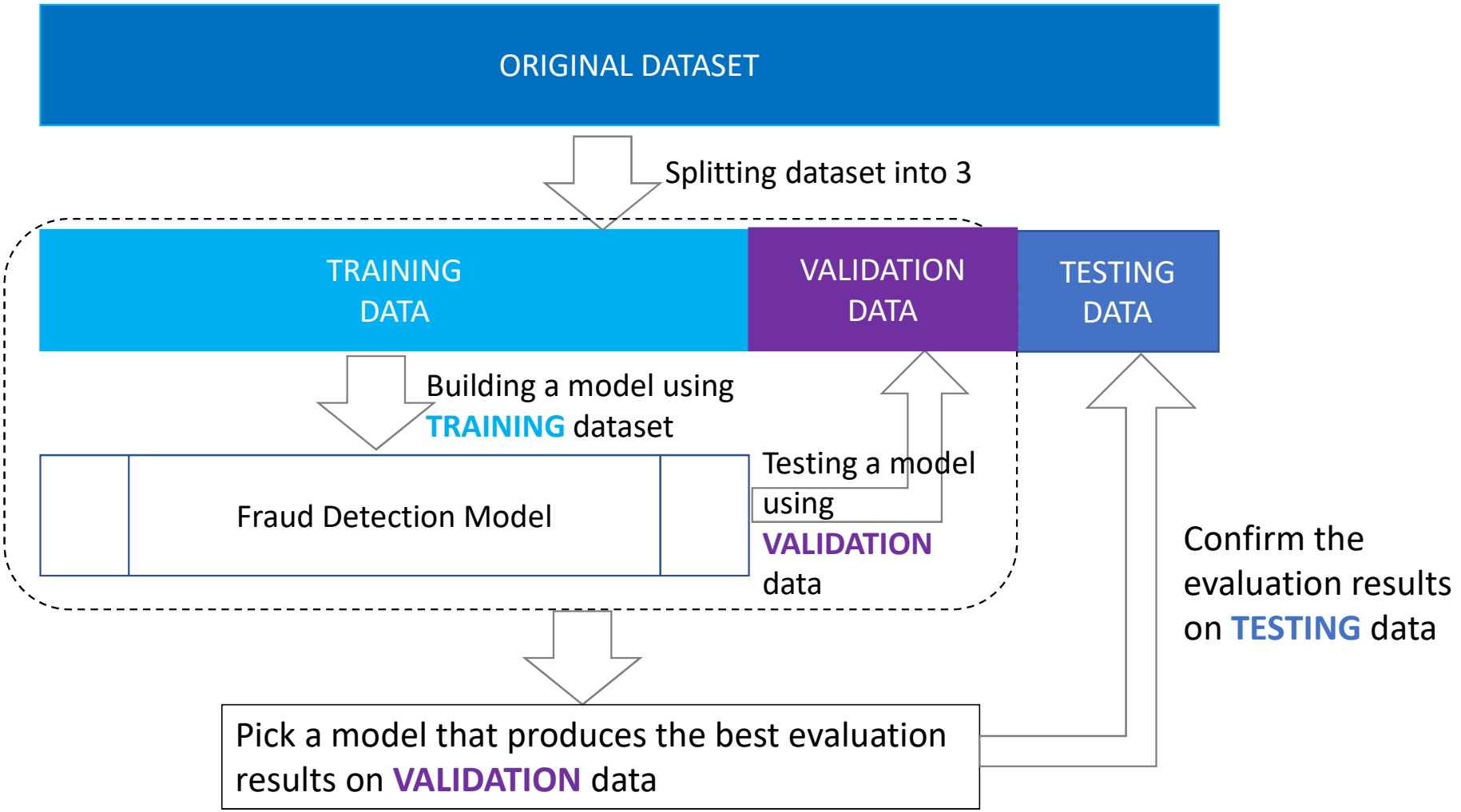
- Observations used for training should not be used for testing or validation
- If validation data set is not required,
  - 70% for training
  - 30% for testing
- If validation data set is required,
  - 40% for training data
  - 30% for validation data
  - 30% for testing data

Source: Fraud Analytics using Descriptive, Predictive, and Social Network Techniques, 1st ed, John Wiley & Sons Inc.

Scenario 1


ORIGINAL DATASET
Splitting dataset into 2
TRAINING
TESTING
Building a model using
TRAINING data
Testing a model using
TESTING data
Fraud Detection Model
Pick a model that produces the best
evaluation results on TESTING data

Scenario 2


ORIGINAL DATASET
Splitting dataset into 3
TRAINING DATA
VALIDATION DATA
TESTING DATA
Building a model using TRAINING dataset
Fraud Detection Model
Testing a model using VALIDATION data
Confirm the evaluation results on TESTING data
Pick a model that produces the best evaluation results on VALIDATION data

If we can train a model using the training data and evaluate it using the testing data. So, Why do we need a validation data set?

Training vs Validation vs Testing

Training data	**Purpose**: to build the model
	A dataset of observations used during the learning process
	The goal is to produce a trained (fitted) model that generalizes well to new, unknown data
	Training data should not be used for validation or testing
Validation data	**Purpose**: to be used during model development (e.g. making stopping decision in decision tree)
	A dataset of observations used to tune the hyperparameters of a prediction model (e.g. decision of when to stop growing a decision tree)
	Training stopped with the minimum error on the validation set
Testing data	**Purpose** : to test the performance of the model
	A dataset of observations independent of the training and validation datasets
	Used only to assess the performance of a trained prediction model
	Overfitting – a better fitting of the training dataset than the testing dataset

Packages in R
Partition the data

- The **caret** package (short for Classification And REgression Training) contains functions to streamline the model training process for complex regression and classification problems.

- Reference: https://topepo.github.io/caret/index.html

```
# Partition the data
library(caret) # for models
```

Partition the data

```r
# Convert class to factor for data modeling
data$Class <- as.factor(data$Class)
```

```r
# Set the seed for reproducibility
set.seed(123)

# Split the data into a TRAINING set and TESTING set
train_index <- createDataPartition(data$Class, times = 1, p = 0.8, list =F)
train <- data[train_index,]
test <- data[-train_index,]
```

Target variable

the number of partitions to create

the percentage of data that goes to training

"F" means the result is a matrix with rows and columns

Result of the data partition

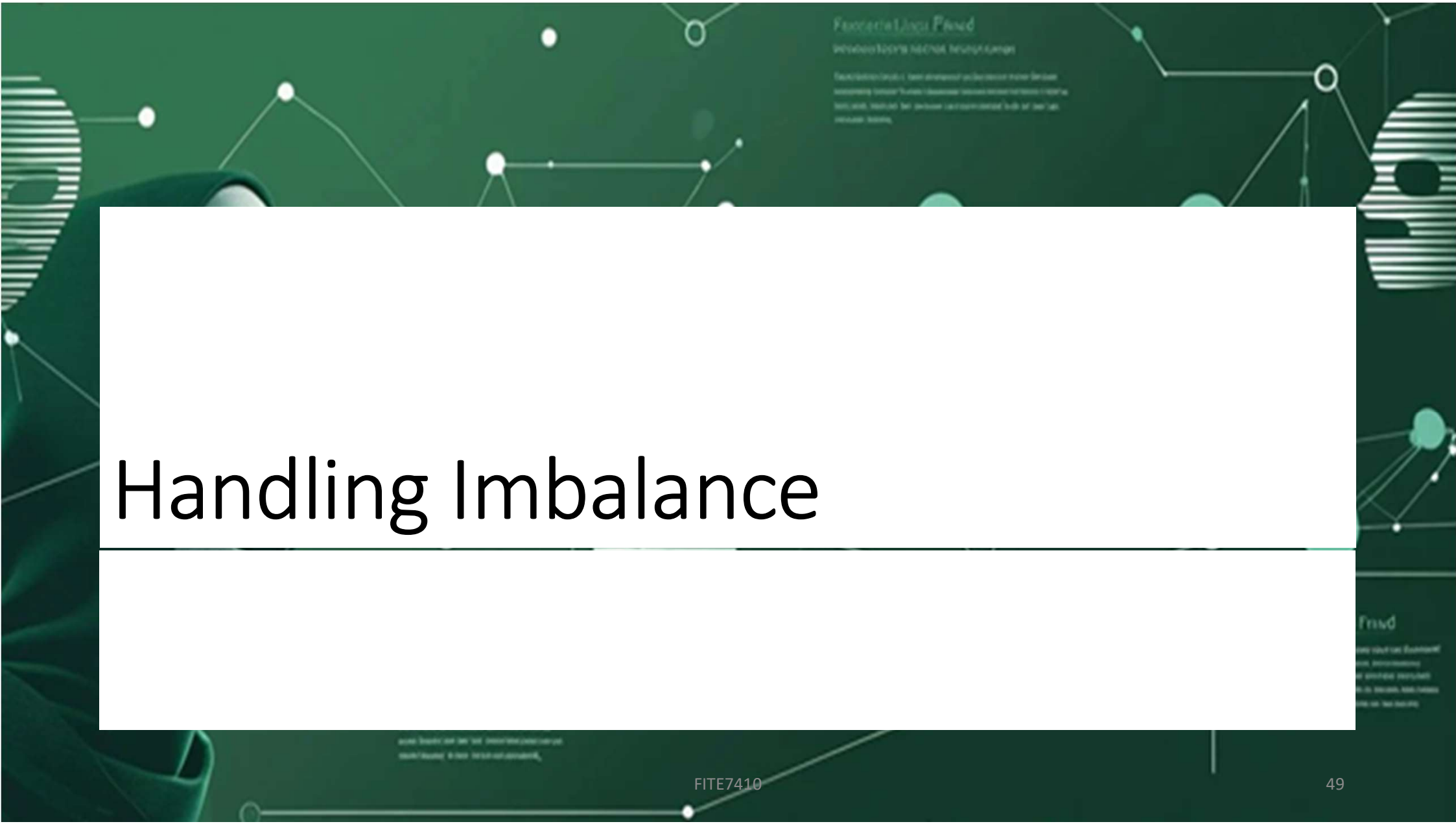
```r
#Show the partition results
print("Original dataset")
table(data$Class)

print("Training dataset")
table(train$Class)

print("Testing dataset")
table(test$Class)
```

```
[1] "Original dataset"
     0      1 
284315    492 
[1] "Training dataset"
     0      1 
227461    385 
[1] "Testing dataset"
    0     1 
56854   107 
```

100%

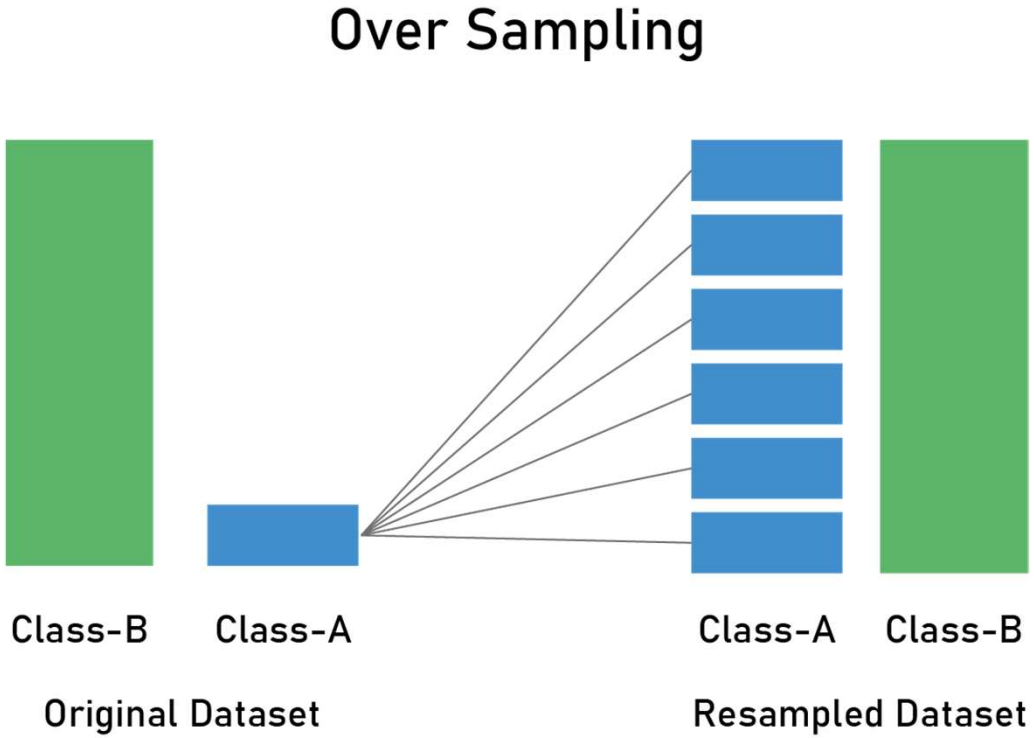
80%

20%



Handling Imbalance

How to handle imbalanced dataset



Over sampling

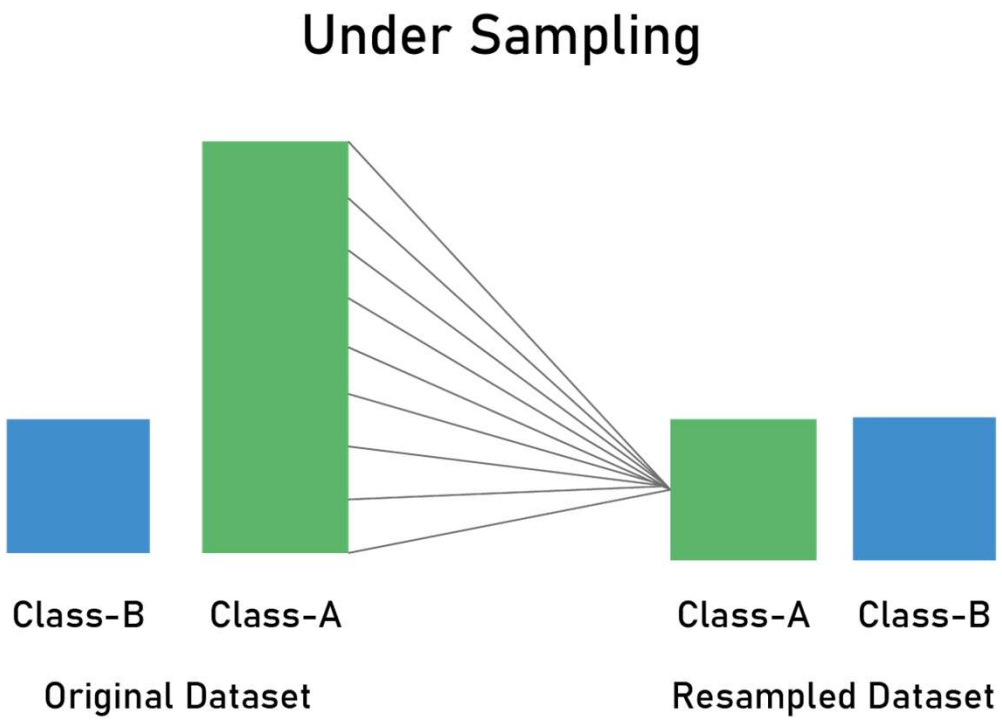
- It helps to increase the number of minority class examples in the dataset.
- One of the main advantages of oversampling is no information is lost from both the majority and minority classes during the process.

How to handle imbalanced dataset

Under sampling

- it helps to reduce the number of majority class examples in the dataset.

What are the problems with over- or under-sampling?



How to handle imbalanced dataset

ROSE (Random Over Sampling Example) (Menardi and Torelli, 2014))

- combines techniques of oversampling and undersampling by generating an augmented sample of data (especially belonging to the rare class)
- thus helping the classifier in estimating a more accurate classification rule, because the same attention will be addressed to both the classes
- the synthetic generation of new examples allows for strengthening the process of learning as well as estimating the distribution of the chosen measure of accuracy

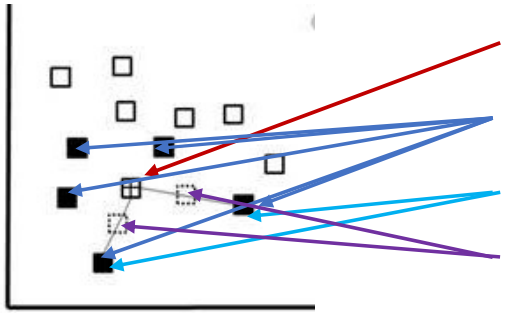
How to handle imbalanced dataset

SMOTE (Synthetic Minority Over-Sampling Technique) (Chawla et al., 2001)

- Creates synthetic observations based upon the existing minority observations
- Combines the synthetic oversampling of the minority class with undersampling the majority class
- SMOTE proven to be better than either under-/over-sampling. It is also proven to be valuable for fraud detection

How to handle imbalanced dataset

SMOTE (Synthetic Minority Over-Sampling Technique) (Chawla et al., 2001)



Steps

1/ Select one minority class observation

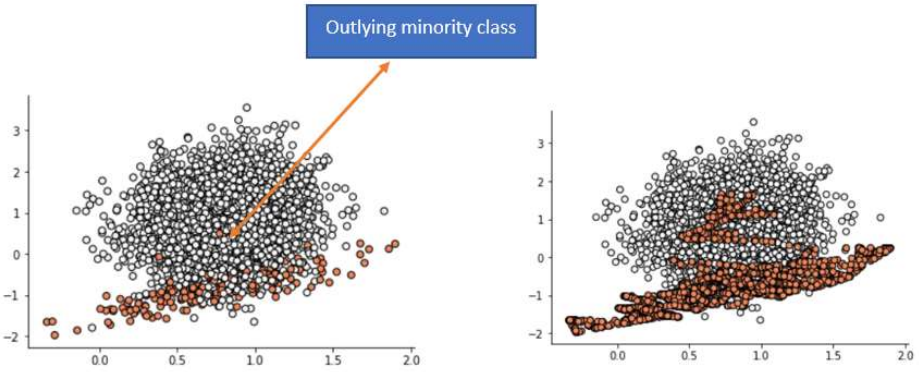
2/ Calculate 5 nearest neighbour

3/ Randomly select 2 nearest neighbour

4/ Randomly create 2 new observations as samples

PROBLEM with SMOTE

If there are observations in the minority class which are outlying and appears in the majority class, it causes a problem for SMOTE, by creating a line bridge with the majority class.



How to handle imbalanced dataset

ADASYN (Adaptive Synthetic sampling)

- a generalization of the SMOTE algorithm
- it takes into account the distribution of density
- it measures the K-nearest neighbors for all minority instances, then calculates the class ratio of the minority and majority instances to create new samples
- For example,
  - Impurity ratio is calculated for all minority data points
  - Higher the ratio, more synthetic data points are created. E.g. the synthetic data points of Obs3 will be 4 times that of Obs2

Example: k=5, i.e. only look at 5 neighbours

Fraud class data points	Fraud class Neighbours	Non-fraud class Neighbours	Impurity Ratio
Observation 1	3	2	0.4
Observation 2	4	1	0.2
Observation 3	1	4	0.8
Observation 4	5	0	0

Packages in R
Imbalance Data Handling

1- ROSE: *The package only implements the algorithm Random Over Sampling*
Link: ROSE
https://cran.r-project.org/web/packages/ROSE/index.html

2- DMwR: *The package reads as "Data Mining with R" and comes with implementation of SMOTE algorithm. SMOTE algorithm uses nearest neighbor concept to oversample the minority class.*
Link: DMwR
https://cran.r-project.org/src/contrib/Archive/DMwR/

NOTE: DMwR removed from CRAN

NOTE: Refer to R Tutorial in Lecture 1

3-Smotefamily*: A Collection of Oversampling Techniques for Class Imbalance Problem*
Based on SMOTE
Link: smotefamily
https://cran.r-project.org/web/packages/smotefamily/index.html

Example - Oversampling

```
library(tidyverse) # for data manipulation
library(caret) # for models
library(ROSE) #for over-/under-sampling and ROSE function
```

```
# oversampling
set.seed(9560)
train_ov <- ovun.sample(Class ~., data=cc_train, method="over")$data
print("OVERSAMPLING result")
table(train_ov$Class)
```

```
[1] "OVERSAMPLING result"

     0      1
227452 226940
```

Formula, meaning is "Class" is predicted by ("~") ALL other variables (".")

Specify the method, i.e. oversampling

Example – Under sampling

```
library(tidyverse) # for data manipulation
library(caret) # for models
library(ROSE) #for over-/under-sampling and ROSE function
```

```
# undersampling with seed parameter

train_un <- ovun.sample(Class ~., data=cc_train, seed=3, method="under")$data
print("UNDERSAMPLING result")
table(train_un$Class)
```

```
[1] "UNDERSAMPLING result"

  0   1
407 394
```

Set the seed

Specify the method, i.e. under sampling

Example - SMOTE

```r
# SMOTE
# Install the smotefamily package
install.packages("smotefamily")

# Load the smotefamily library
library(smotefamily)

# SMOTE sampling
set.seed(9560)
train_smote <- SMOTE(X = cc_train[,-31], target = cc_train$Class, K=5)

print("SMOTE sampling result")
table(train_smote$data$class)
```

ALL independent variables, except the target variable, which is at column 31

K = number of nearest neighbour

```
[1] "SMOTE sampling result"

     0      1
227452 227338
```

Example - ROSE

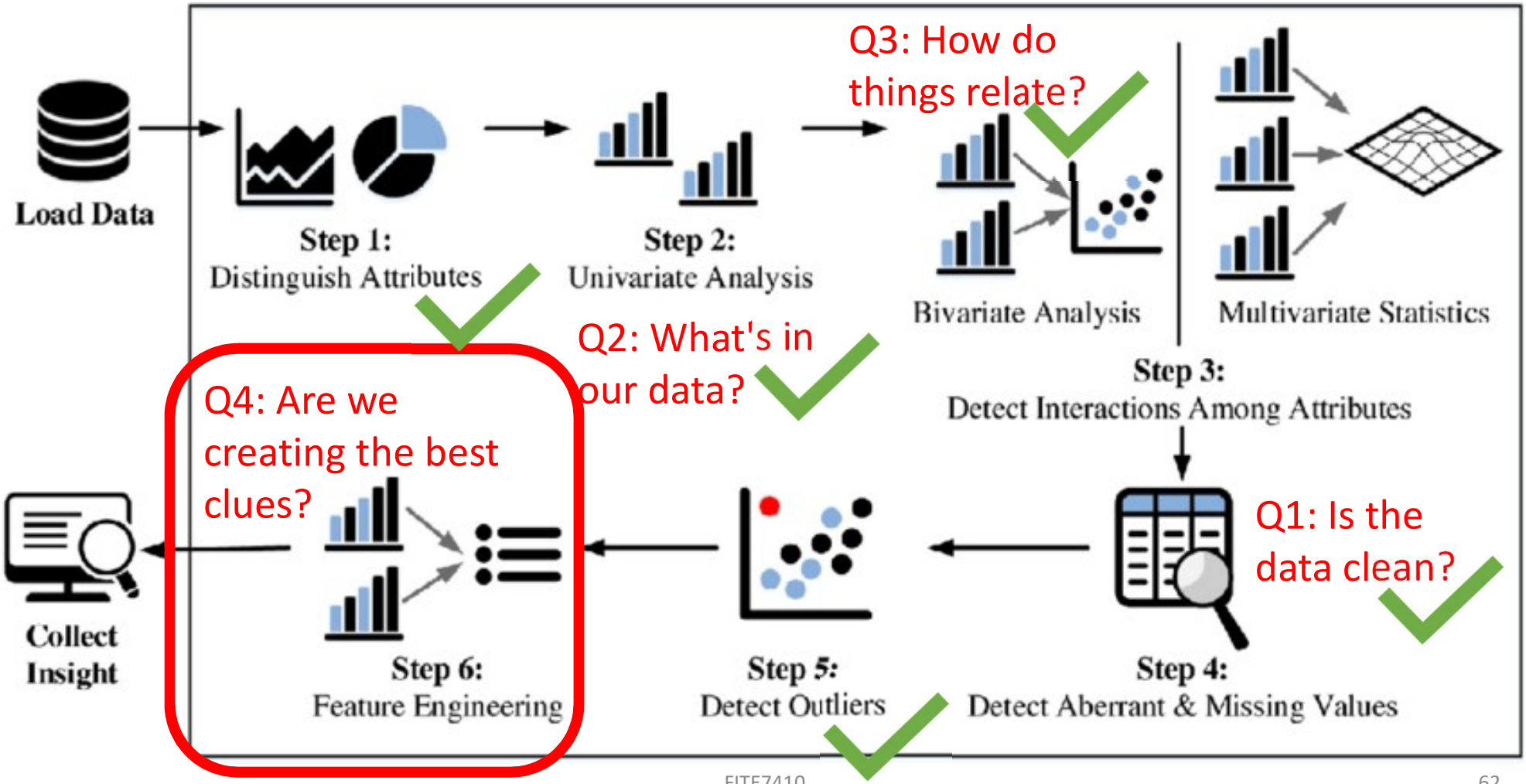
```
library(tidyverse) # for data manipulation
library(caret) # for models
library(ROSE) #for over-/under-sampling and ROSE function
```

```
# ROSE
set.seed(9560)
train_rose <- ROSE(Class ~., data=cc_train)$data
print("ROSE sampling result")
table(train_rose$Class)
```

```
[1] "ROSE sampling result"

     0      1
114081 113765
```

Feature Scaling and Engineering

Step-by-step EDA (Data Pre-processing)



Ref: Ghosh et al. (2018). A comprehensive review of tools for exploratory analysis of tabular industrial datasets

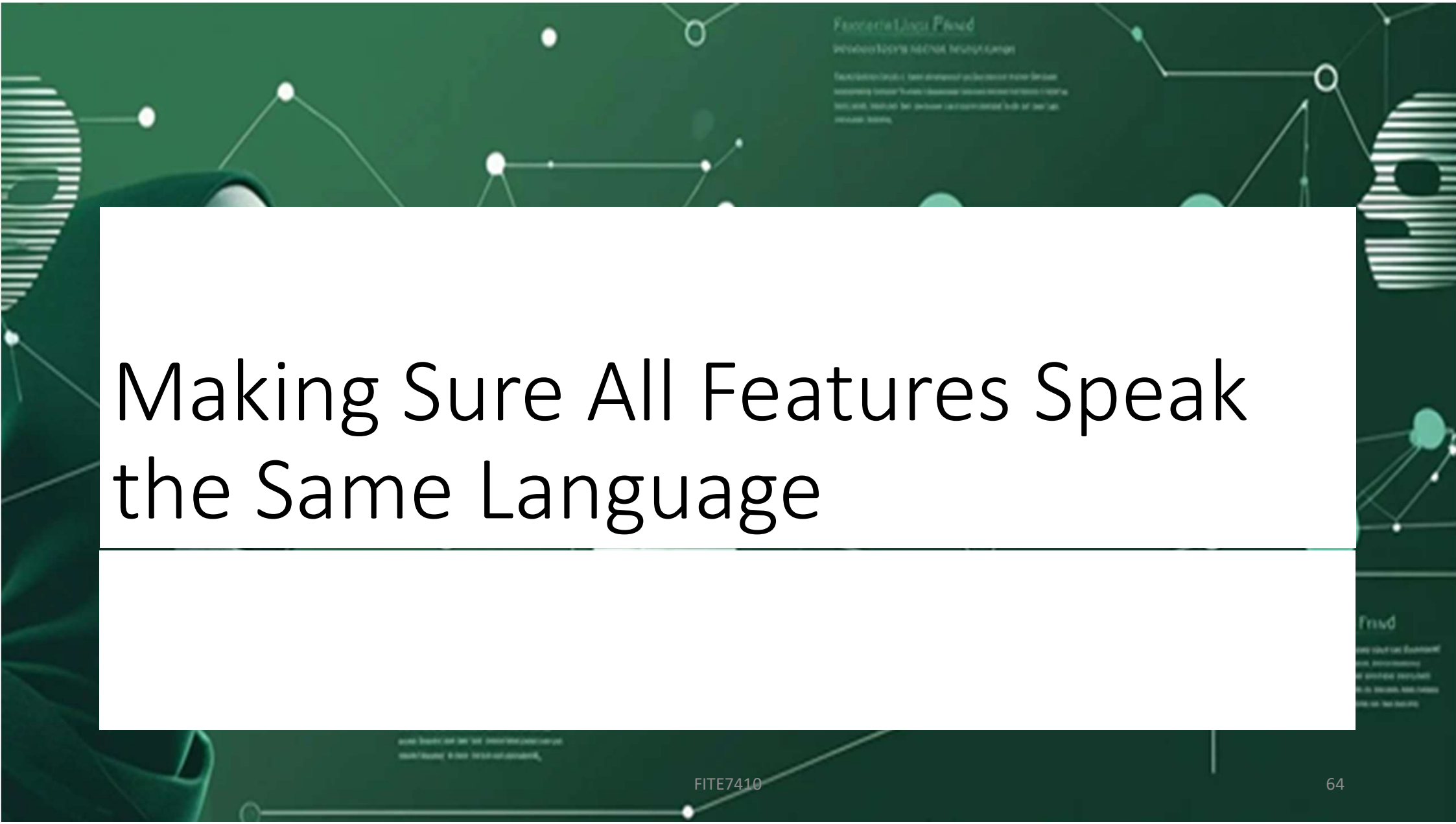
EDA – Step 6: Feature Engineering

Objective

- Last step of EDA after obtaining detailed insights of the dataset
- Creation and transformation of variables
- Dimensionality reduction

Techniques

- **Variable creation** : To ease the data analysis process through a/ turning non-linear relation into linear relation; b/ help to simplify the understanding of complex attributes in the dataset
- **Variable transformation** : a/ binning or categorization strategies on split up continuous variables into categories to gather more insight from them; b/ normalization - a type of variable transformation that helps to convert skewed distributions into more symmetric distributions.

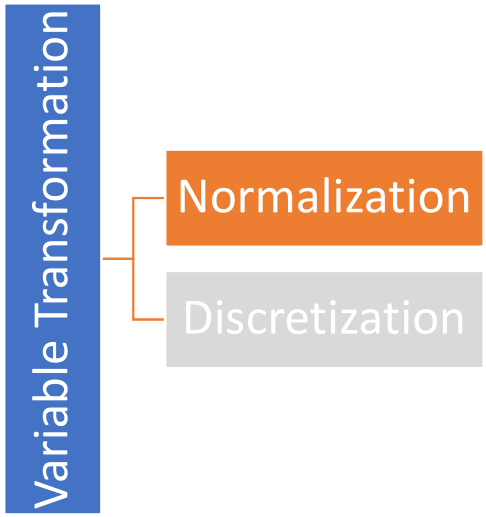


Making Sure All Features Speak the Same Language

Data Transformation

- Data Transformation is the process of converting the data from one format to another easy-to-read format so as to use for data analysis.
- Also known as ETL (Extract, Transform, Load) – data is extracted from multiple sources, transformed to a single format, and loaded into a data warehouse for data analysis process.
- It serves several purposes:
  - For easy comparison among different data sets with diverse format
  - For easy combination with other data sets to provide insights
  - To perform aggregation of data

Techniques of Data Transformation



1/ Normalization

- It is a technique to change the values of attributes to a common scale, without distorting differences in the ranges of attribute.
- It is required only when attributes have different ranges. For example,
  - AGE range from 0 – 100; INCOME range from 10,000 – 100,000
  - Problem : INCOME might have larger affect on the predictive power of the model due to its larger value (100 times larger than AGE)

Normalization - Simple transformation

- For continuous variables
  - Min-Max Normalization (Range Normalization)

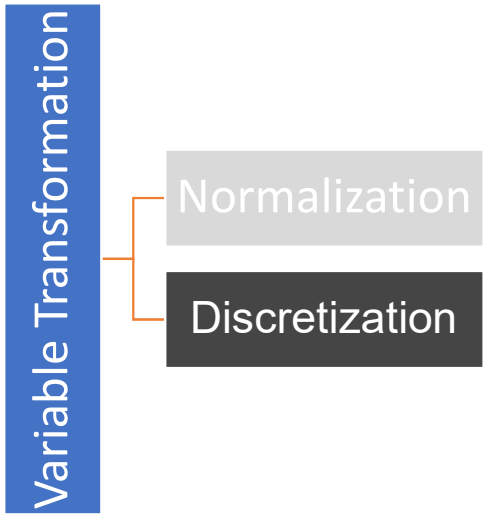
$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

  - Z-score Standardization

$$x_{scaled} = \frac{x - mean}{sd}$$

  - (natural) log or base-10 log
  - Square root
  - Inverse
  - Square
  - Exponential
  - Centering (subtract mean)

Techniques of Data Transformation



2/ Discretization

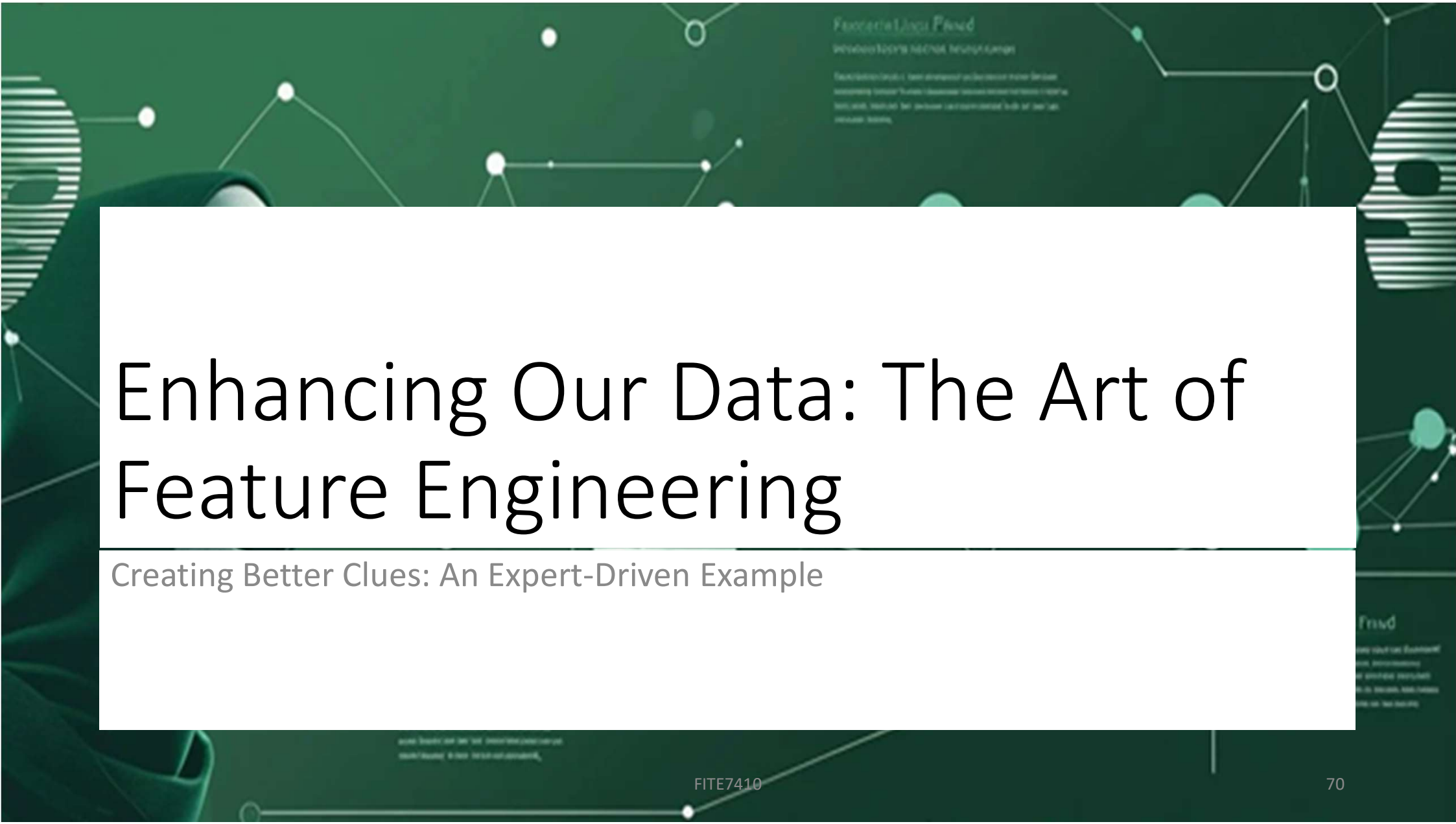
- It is a technique to replace the values of numeric attribute (continuous variable) by conceptual values (discrete/categorical variable).
- Method : Binning transformation (or categorization)
- Examples
  - replace AGE (numeric value) with AGEGROUP (children, youth, adult, elderly)
  - group rare levels into one discrete group “OTHER”, e.g. use “OTHER” to represent those with values that occur less than a specified cutoff value (e.g. less than 5%)

Binning transformation

- Methods
  - Equal interval binning
  - Equal frequency binning
- For example,
  - Create 2 bins of equal range:
    - BIN 1 (range 1,000 – 1,500) : A, B, C, F
    - BIN 2 (range 1,500 – 2,000) : D, E
  - Create 2 bins of equal frequency:
    - BIN 1 : A, B, C
    - BIN 2 : D, E, F

CUSTOMER	INCOME
A	1,000
B	1,200
C	1,300
D	2,000
E	1,800
F	1,400

- The above methods do not take into consideration the target variable (e.g. FRAUD)

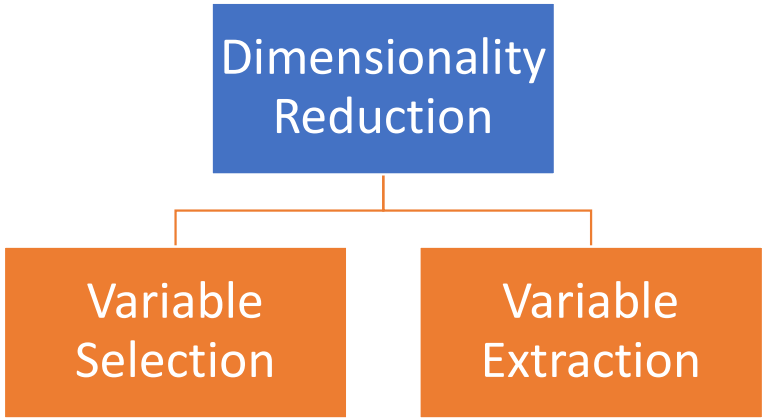


Enhancing Our Data: The Art of Feature Engineering

Creating Better Clues: An Expert-Driven Example

Dimensionality Reduction

- Purposes
  - Operational efficiency, with reduced amount of time and memory required
  - Better interpretability with easier visualization using fewer variables
  - Eliminate irrelevant features. For example, on average only 10-15 variables are useful in fraud detection models.

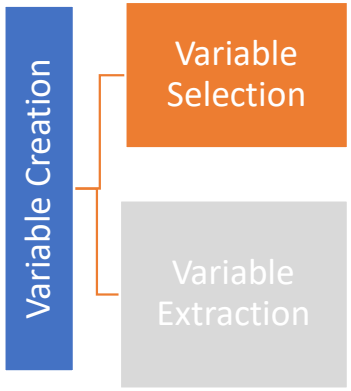


Techniques:

- Feature Elimination:
  - Drop some features that may be unimportant.
  - While the approach is simple, may lose useful information present in those dropped features.
- Feature Extraction:
  - Transform the original set of features into another set of features.
  - To pack the most important information into as few derived features as possible
  - Reduce the number of dimensions by dropping some of the derived features. But don't lose complete information from the original features: derived features are a linear combination of the original features.

Variable selection

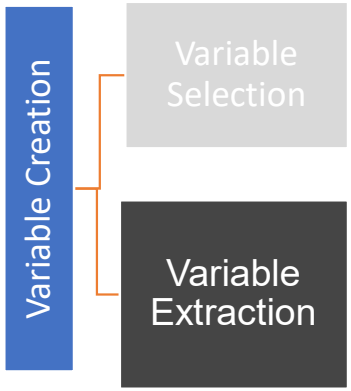
- Input variables are selected (or filtered) based on the usefulness or relation with Target variables (e.g. fraud vs non-fraud, risk index, etc)
- Methods include:
  - Correlation with target variable
  - Information criteria
  - Clustering of variables



TARGET VARIABLE

INPUT		Continuous Target (e.g., CLV, LGD)	Categorical Target (e.g., churn, fraud, credit risk)
	Continuous variable	Pearson correlation	Fisher score
	Categorical variable	Fisher score/ANOVA	Information value Cramer's V Gain/entropy

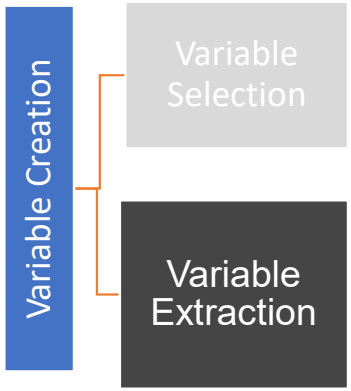
Principle Component Analysis (PCA)



- PCA is a technique :
  - to reduce the dimensionality of data by forming new variables that are linear composites of the original variables
  - useful for exploratory data analysis, allowing you to better visualize the variation present in a dataset with many variables

- Basics of PCA are as follows:
  - A dataset with many variables
  - Simplify that dataset by turning the original variables into a smaller number of "Principal Components"

Principle Component Analysis (PCA)



- Theoretically, the number of Principle Components is the same as the number of variables or number of samples, whichever is smaller. BUT, we usually only keep those principle components that have high correlations

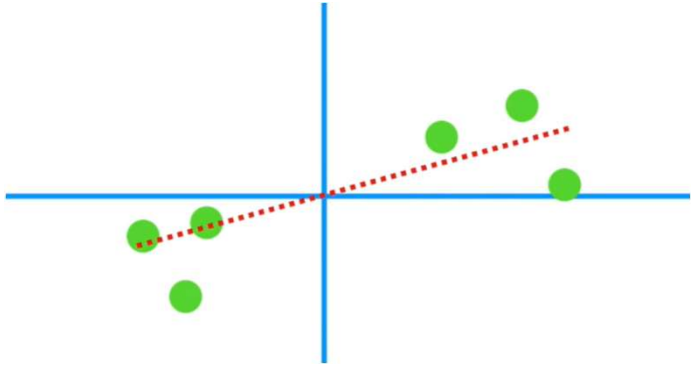
- Formula : Z = YV ,where

Z is matrix of principal components scores (n x m)
Y is standardized data matrix (n x p)
V is matrix of eigenvectors (p x m)
p is the number of variables
n is the number of observations
m is the first m principle components

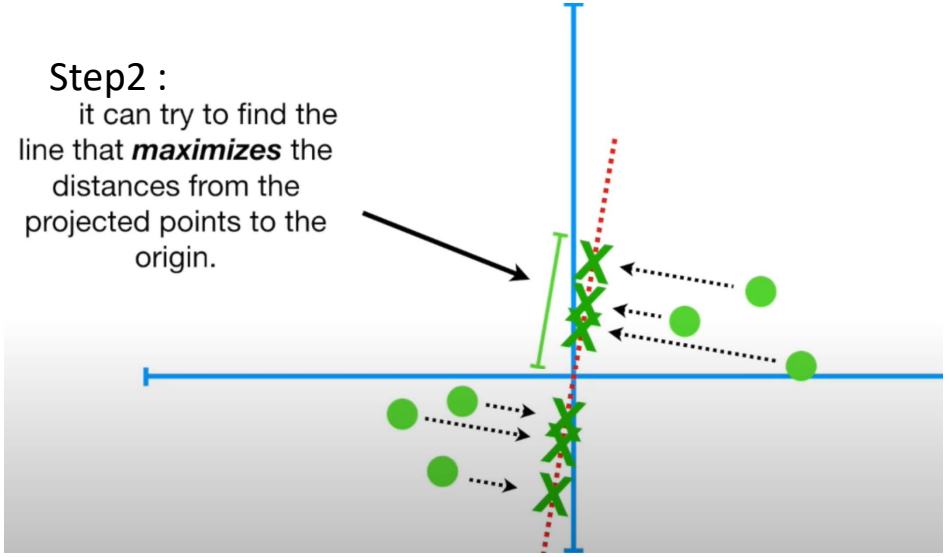
The mathematics underlying PCA are somewhat complex, so I won't go into too much detail

PCA – explained #1

Step1 : Draw a random line that goes through the origin



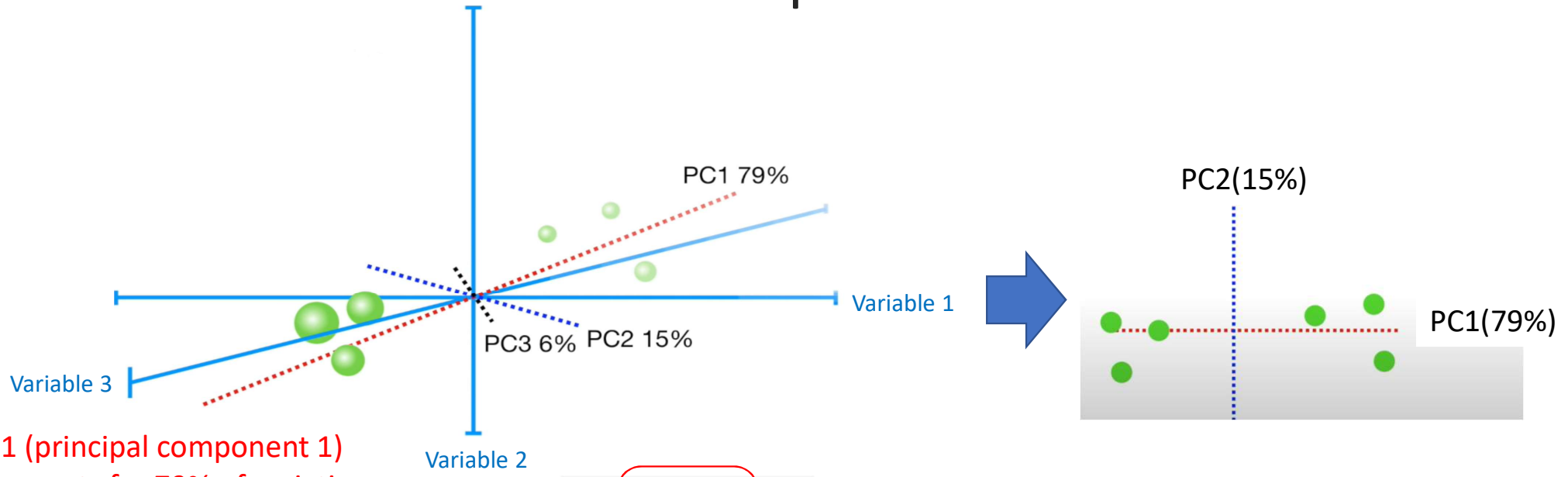
Step2 :
it can try to find the line that ***maximizes*** the distances from the projected points to the origin.



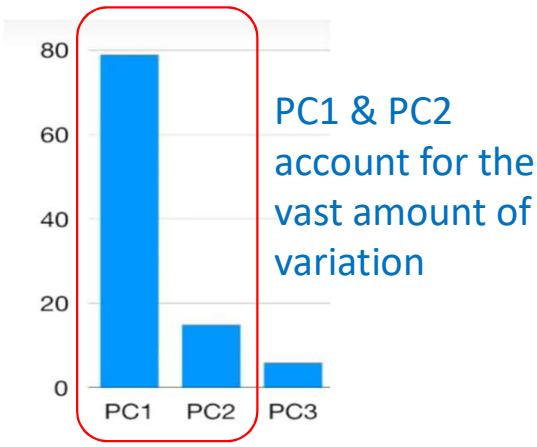
- **PC1**, the first principal component, is the straight line that shows the most substantial variance in the data.
- This **linear transformation** fits this dataset to a new coordinate system in such a way that the most significant **variance** is found on the first coordinate, and each subsequent coordinate is **orthogonal** to the last and has a lesser variance.

Source: https://www.youtube.com/watch?app=desktop&v=FgakZw6K1QQ

PCA – explained #2



PC1 (principal component 1)
- accounts for 79% of variation
- data points spread more along variable 1
- variable 1 is more important than variable 2 and 3
- linear combination of variable 1,2 and 3



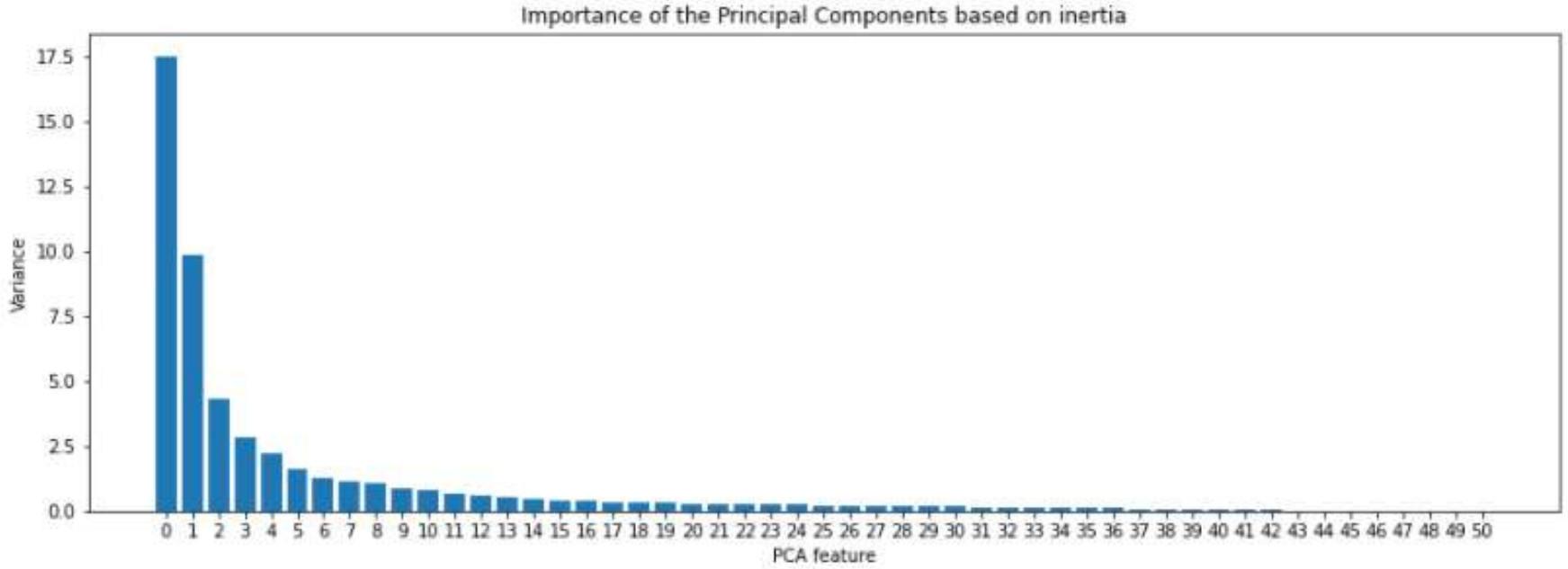
PC1 & PC2 account for the vast amount of variation

2-D graph (right-side of the figure) using just PC1 and PC2:

- would be a good approximation of 3-D graph (left-side of the figure)
- since it would account for 94% (PC1+PC2) of the variation in the data

Source: https://www.youtube.com/watch?app=desktop&v=FgakZw6K1QQ

PCA - example



- From the above graph, the first few principal components are most important
- Thus, we can use the first few principal components for building the model instead of using 50+ variables

Case study-Understanding the Landscape with Univariate Analysis

Case Background

- Background:
  - Credit card division of a large international bank in Brazil
  - Credit card holders who don't want to pay the annual fees may call the bank asking for cancelation or a fee reduction
  - Bank representatives negotiate with the clients about the fees. During the discount negotiation process, bank representatives should follow the bank policy; they cannot offer discounts higher than their authority. And within their jurisdiction, they should also give top priority to the benefit of the bank. In other words, they should offer the lowest discounts acceptable to the clients.

QUESTION: A bank finds that it's losing revenue on credit card renewals because representatives are giving out discounts. Is this just good customer service, or could it be fraud? Let's use our framework to break it down.

Source: Liu, Qi. (2019). An Application of Exploratory Data Analysis in Auditing – Credit Card Retention Case. 10.1108/978-1-78743-085-320191001.

Define scope of Fraud Data Analytics

- What is the type of credit card fraud scenario?
  - Example: Credit card renewal discount offer

- What are the objectives of the fraud data analytics?
  - To identify any non-compliance behaviours of credit card representatives that would case revenue loss of the credit card company

Source: Liu, Qi. (2019). An Application of Exploratory Data Analysis in Auditing – Credit Card Retention Case. 10.1108/978-1-78743-085-320191001.

The dataset

- The account master data is a large dataset with 60,309,524 records and 504 fields.
- Description of 8 selected attributes in this credit card case

Attribute Name (Source Database)	Description
Call Length (Retention)	The duration of each call in seconds
Call Location (Retention)	The location of the customer service center
Agent Number (Retention)	ID of the bank representative answering the call
Supervisor Number (Retention)	ID of the representative's supervisor
Sequential Number (Retention & Account Master)	Sequence Number of an account
Annual Fee (Retention)	Original annual fees of a credit card
Output Annual Fee (Retention)	Actual annual fees paid by clients
Number of Cards (Account Master)	Number of cards associated with each account

Source: Liu, Qi. (2019). An Application of Exploratory Data Analysis in Auditing – Credit Card Retention Case.

Data pre-processing

- Data pre-processing
  - E.g. Data transformation
    - achieved by the logarithm function.
  - E.g. Feature re-engineering: Creation of new attributes – 'Discount'
  - 2 attributes related to 'Discount'
    - Original fee = original annual fee
    - Actual fee = actual annual fee paid

$$Discount = \frac{(Original\ fee - Actual\ fee)}{Original\ fee} \times 100\%$$

Q: What does the following values mean?

Case 1: Discount = 0%

Case 2: Discount = 100%

Case 3: Discount = -ve value

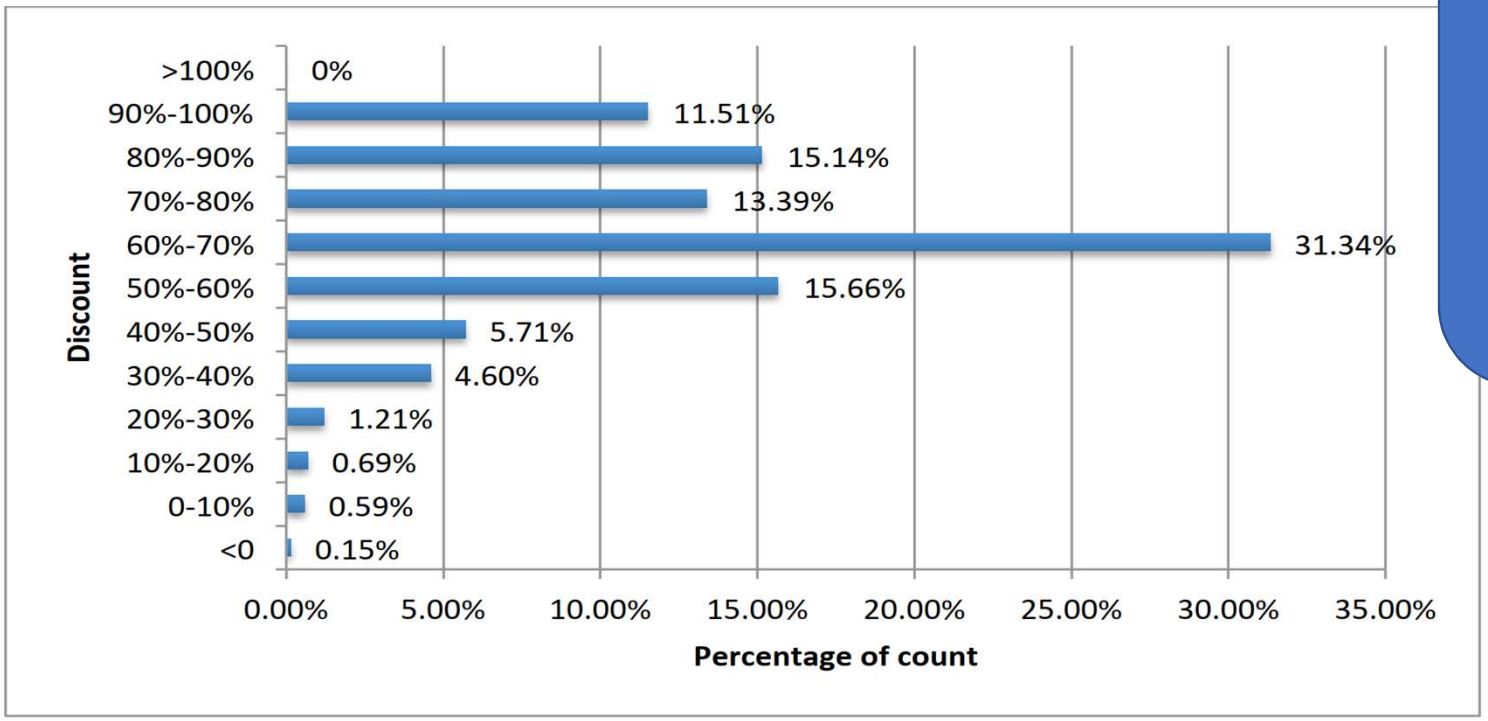
Source: Liu, Qi. (2019). An Application of Exploratory Data Analysis in Auditing – Credit Card Retention Case. 10.1108/978-1-78743-085-320191001.

EDA - Univariant Analysis

- Descriptive statistics used in this study include frequency distribution, summary statistics (mean and standard deviation), and categorical summarization.
- Looking at the following one by one:
  - ‘Discount’ offer patterns and frequencies
  - ‘Discount’ by agent number (i.e. credit card representatives)
  - ‘Discount’ by call lengths
  - ‘Discount’ by call locations
  - ….

EDA - Univariate Analysis : ‘Discount’

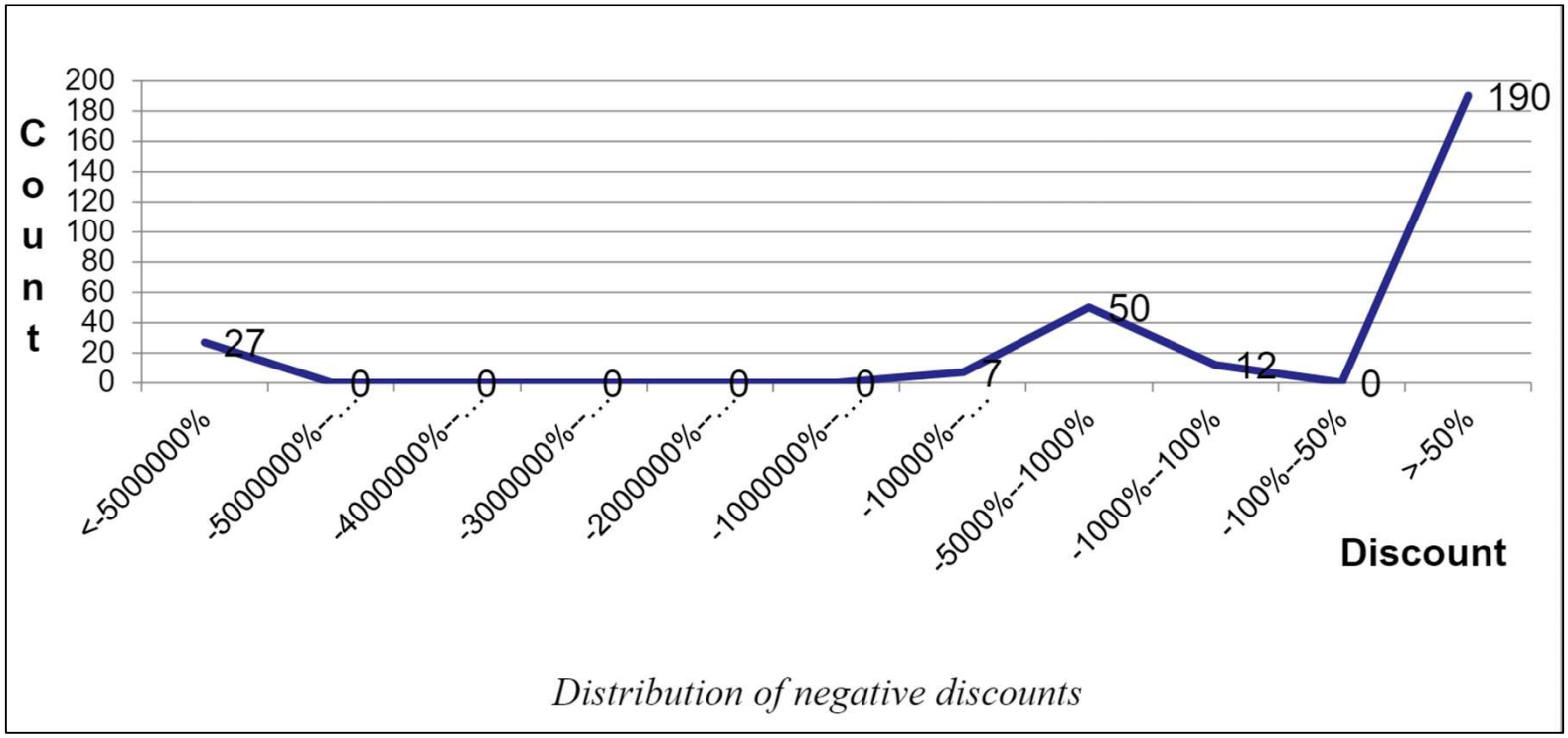
Field Name	Mean	Median	Minimum	Maximum	Standard deviation
Discount	-2.326.04%	60%	-27,944,522.22%	100.00%	219933.88%



Any insights you get from this initial univariate analysis?

EDA - Univariate Analysis : ‘Discount’

Field Name	Mean	Median	Minimum	Maximum	Standard deviation
Discount	-2.326.04%	60%	-27,944,522.22%	100.00%	219933.88%

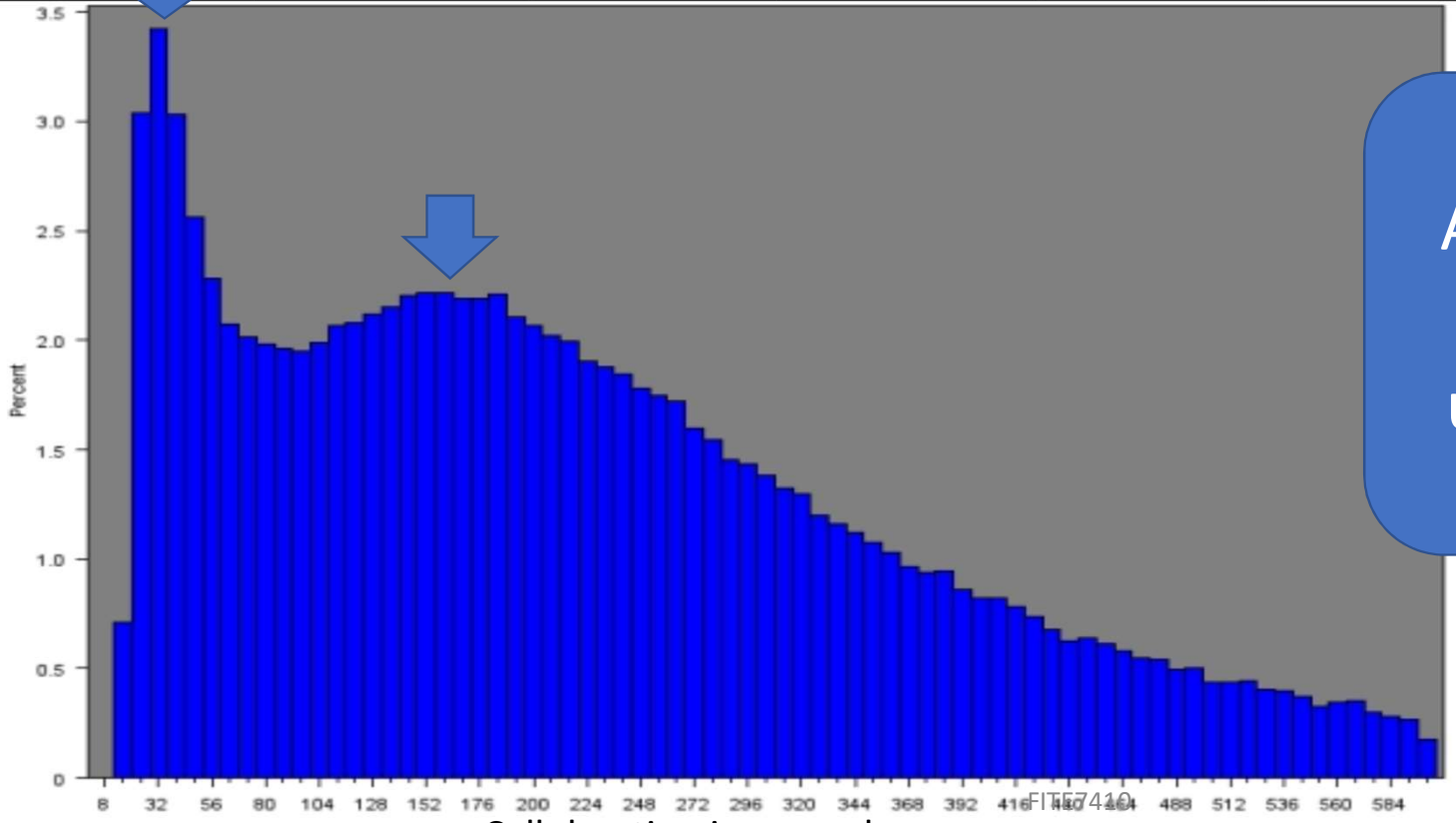


Distribution of negative discounts

Why is –ve discount recorded?

EDA - Univariate Analysis : 'Call duration'

Minimum	Maximum	Mean	Median	90th Percent	Count
10	6561	255	206	514	195694



Call duration in seconds

Any insights you get from this initial univariate analysis?

Key Takeaways- From Raw Data to Predictive Asset

EDA is your first investigative step—ask questions of the data.

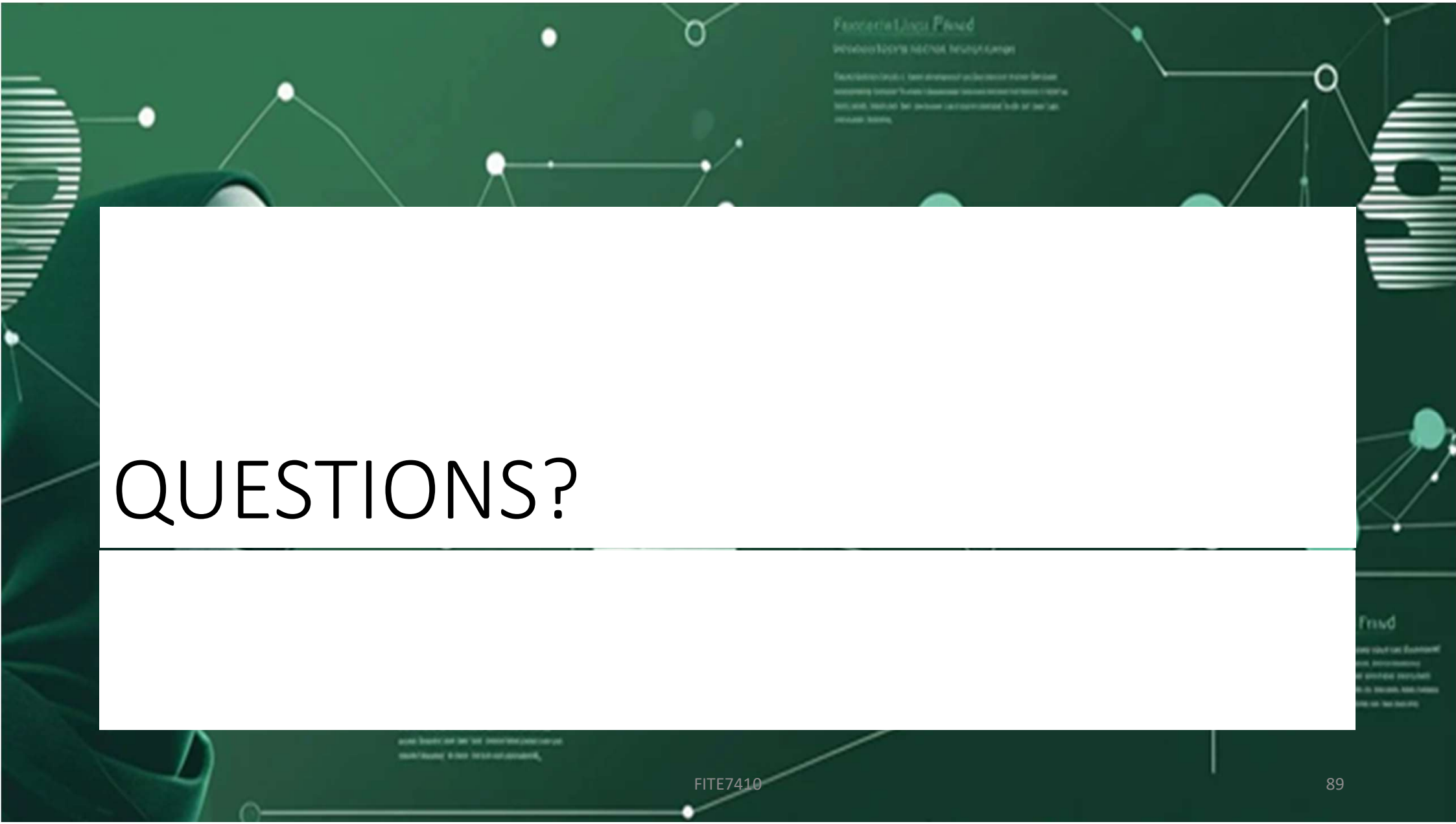
EDA is not just for finding red flags; it's for diagnosing the data's fitness for modeling.

Fraud detection model starts with data pre-processing.

- Class imbalance is the primary enemy of a fraud detection model. Identifying and solving it with techniques like SMOTE is not optional—it is essential.
- A structured pre-processing workflow (Partition -> Balance -> Scale) is what transforms messy, real-world data into a reliable source of truth for your model.

References

- Bart Baesens, Veronique Van Vlasselaer, Wouter Verbeke (2015). Fraud Analytics using Descriptive, Predictive, and Social Network Techniques, 1st ed, John Wiley & Sons Inc.
- Leonard W. Vona (2017). Fraud Data Analytics Methodology: The Fraud Scenario Approach to Uncovering Fraud in Core Business Systems, John Wiley & Sons, Inc.
- Sabău, A.-I., Gherai, R. S., & Todea, A. (2021). A statistical model of fraud risk in financial statements. Case for companies listed on the Bucharest Stock Exchange. *Risks*, *9*(12), 222. https://doi.org/10.3390/risks9120222



QUESTIONS?