DATA ENGINEERING MODEL IMPROVEMENT DATA CLEANING

- Data Quality (validity, accuracy, completeness, consistency, uniformity)
- The workflow (inspection, cleaning, verifying, reporting)
- Inspection (data profiling, visualizations, software packages)
- Cleaning ()

DATA QUALITY

DATA QUALITY

VALIDITY

- The degree to which the data conform to defined business rules or constraints.

Data-Type Constraints: values in a particular column must be of a particular datatype, e.g., boolean, numeric, date, etc.

Range Constraints: typically, numbers or dates should fall within a certain range.

Mandatory Constraints*:* certain columns cannot be empty.

Unique Constraints: a field, or a combination of fields, must be unique across a dataset.

Set-Membership constraints**:** values of a column come from a set of discrete values, e.g. enum values. For example, a person's gender may be male or female.

Foreign-key constraints**:** as in relational databases, a foreign key column can't have a value that does not exist in the referenced primary key.

Regular expression patterns: text fields that have to be in a certain pattern. For example, phone numbers may be required to have the pattern (999) 999–9999.

ACCURACY

- The degree to which the data is close to the true values.
- While defining all possible valid values allows invalid values to be easily spotted, it does not mean that they are accurate.

COMPLETENESS

- The degree to which all required data is known.
- Missing data is going to happen for various reasons. One can mitigate this problem by questioning the original source if possible, say re-interviewing the subject.
- Chances are, the subject is either going to give a different answer or will be hard to reach again.

CONSISTENCY AND UNIFORMITY

- **Consistency**
- The degree to which the data is consistent, within the same data set or across multiple data sets.
- Inconsistency occurs when two values in the data set contradict each other.

- **Uniformity**
- The degree to which the data is specified using the same unit of measure.

THE WORK FLOW

THE WORK FLOW

- The workflow is a sequence of three steps aiming at producing high-quality data and taking into account all the criteria we've talked about.

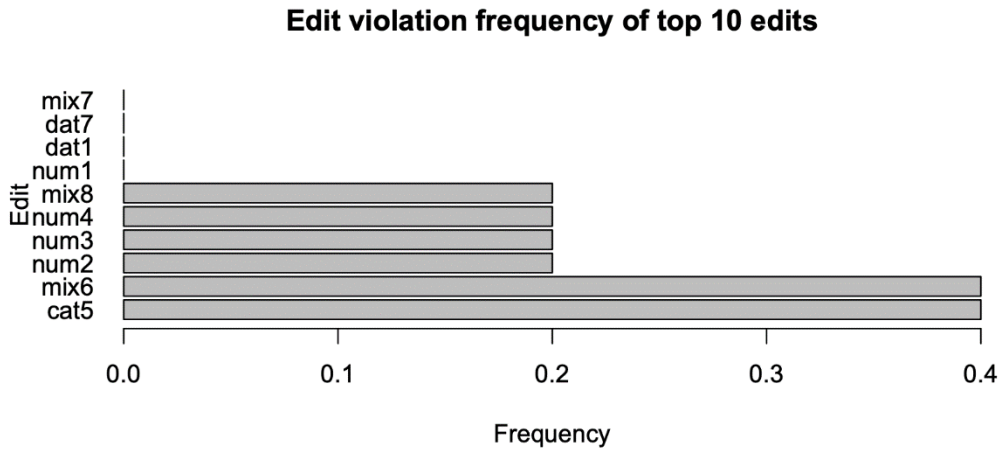
1. **Inspection:** Detect unexpected, incorrect, and inconsistent data.
2. **Cleaning:** Fix or remove the anomalies discovered.
3. **Verifying:** After cleaning, the results are inspected to verify correctness.
4. **Reporting:** A report about the changes made and the quality of the currently stored data is recorded.

INSPECTION

- Inspecting the data is time-consuming and requires using many methods for exploring the underlying data for error detection.
- **Data profiling**

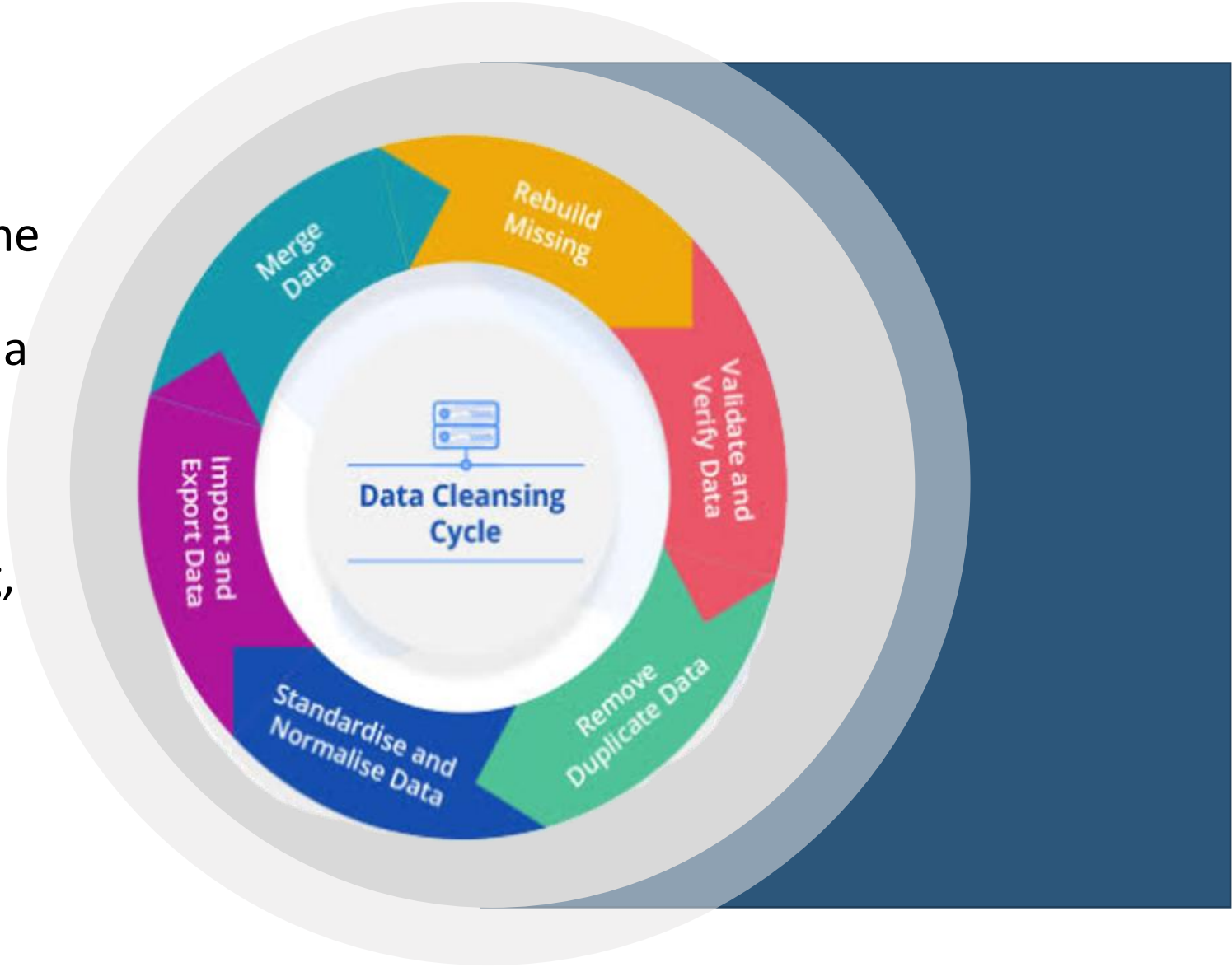
  A **summary statistics** about the data, called data profiling, is really helpful to give a general idea about the quality of the data.

- **Visualizations**
  - By analysing and visualizing the data using statistical methods such as mean, standard deviation, range, or quantiles, one can find values that are unexpected and thus erroneous.



CLEANING

- Data cleansing or data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data



IRRELEVANT DATA

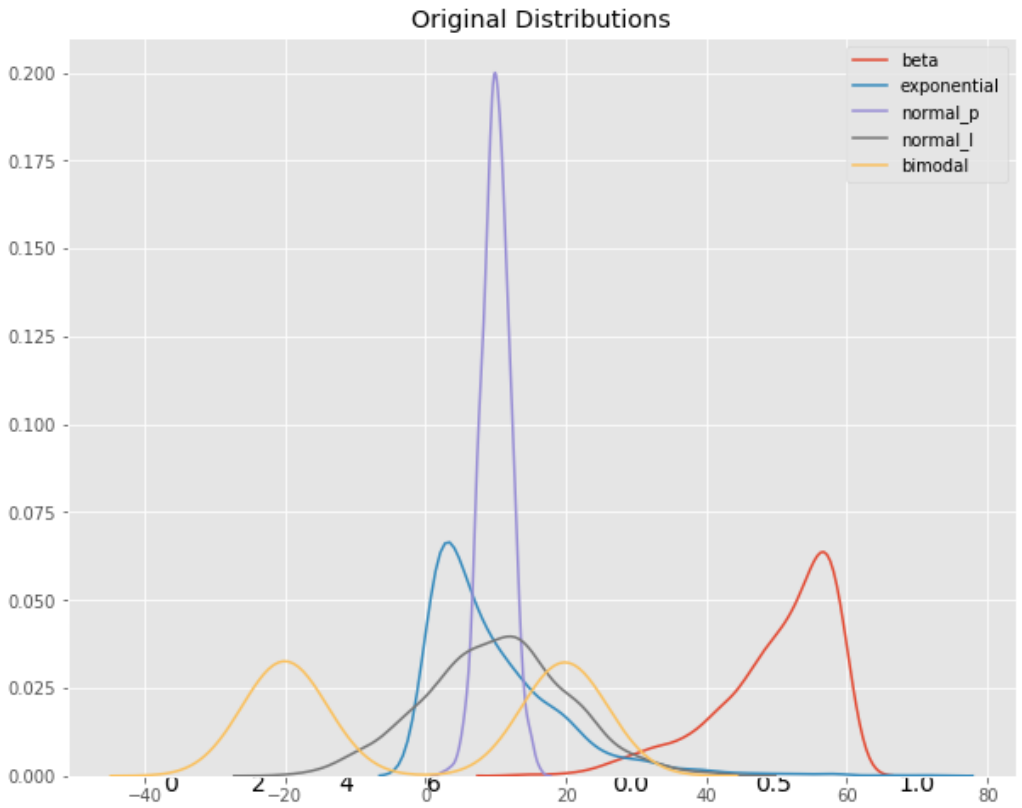
- Irrelevant data are those that are not actually needed, and don't fit under the context of the problem we're trying to solve.
- For example, if we were analyzing data about the general health of the population, the phone number wouldn't be necessary — *column-wise*.

DUPLICATES

- Duplicates are data points that are repeated in your dataset.
- It often happens when for example
- Data are combined from different sources
- The user may hit submit button twice thinking the form wasn’t actually submitted.
- A request to online booking was submitted twice correcting wrong information that was entered accidentally in the first time.

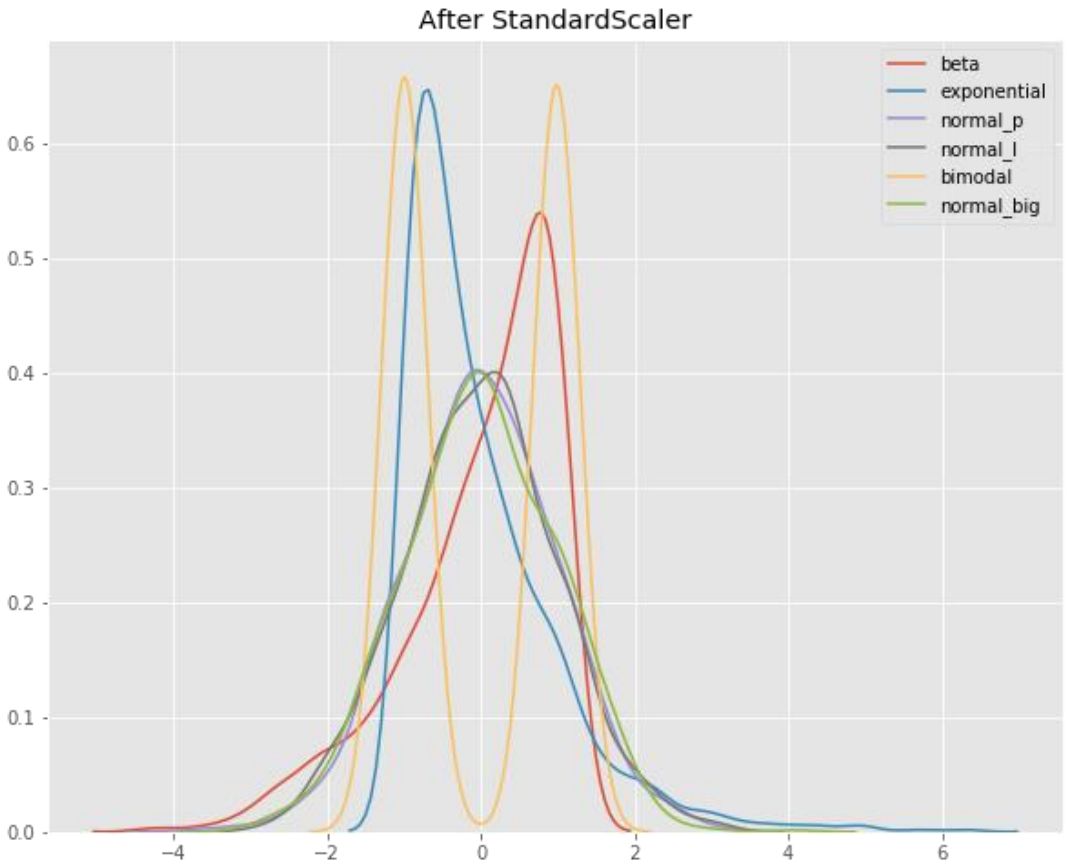
SCALING / TRANSFORMATION

- Scaling means to transform your data so that it fits within a specific scale, such as 0–100 or 0–1.
- It can also help in making certain types of data easier to plot. For example, we might want to reduce skewness to assist in plotting (when having such many outliers). The most commonly used functions are log, square root, and inverse.
- It can also help in comparing the different data like ‘age’ and ‘salary’ in our data set.



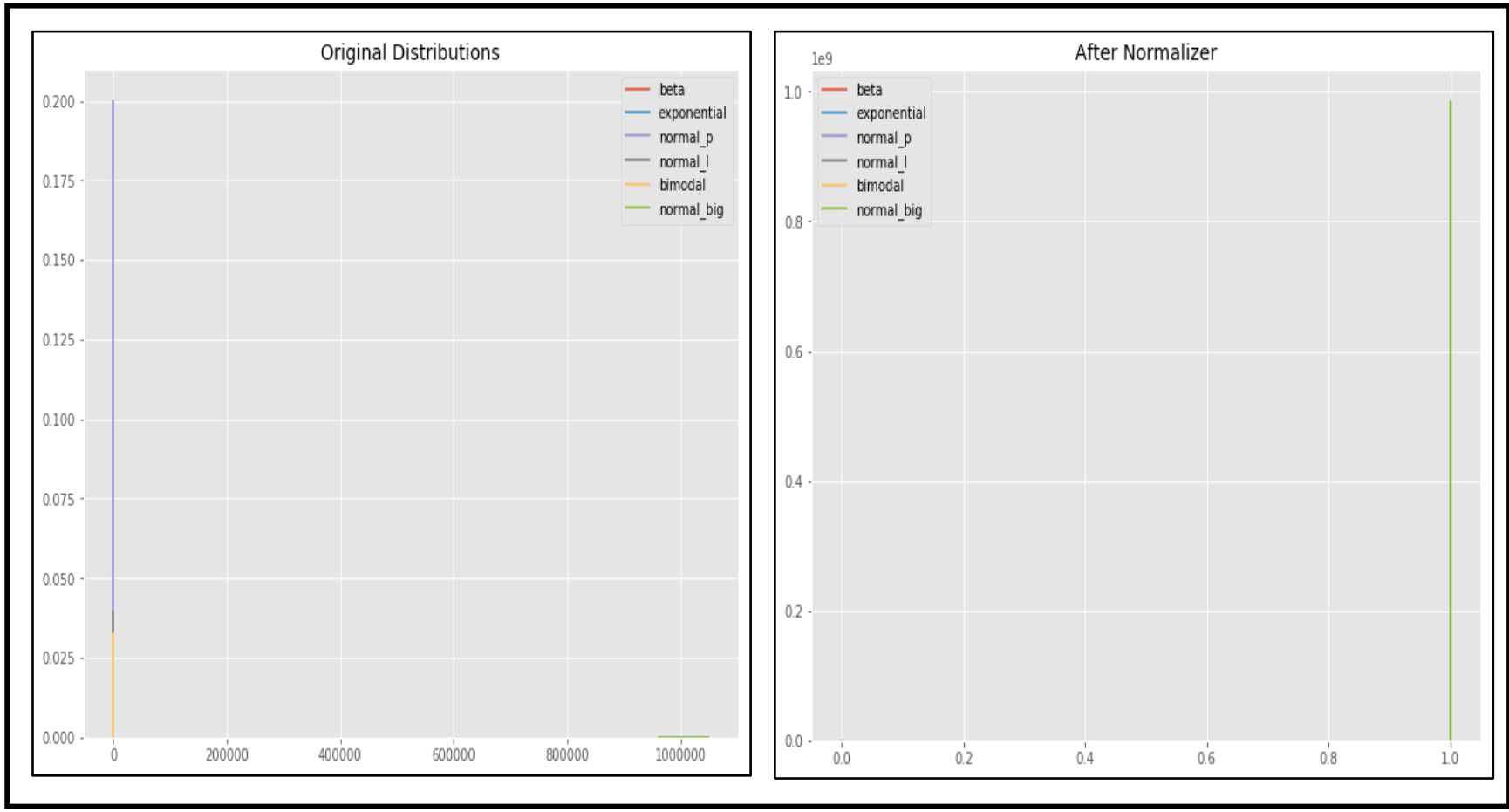
THE STANDARD SCALER

- The Standard Scaler is one of the most widely used scaling algorithms out there. It assumes that your data follows a Gaussian distribution
- The mean and the standard deviation are calculated for the feature and then the feature is scaled based on:
- ***(xi–mean(x))/stdev(x)***
- The idea behind **Standard Scaler** is that it will transform your data, such that the distribution will have a mean value of 0 and a standard deviation of 1.
- If the data is not normally distributed, it's not recommended to use the **Standard Scaler**.



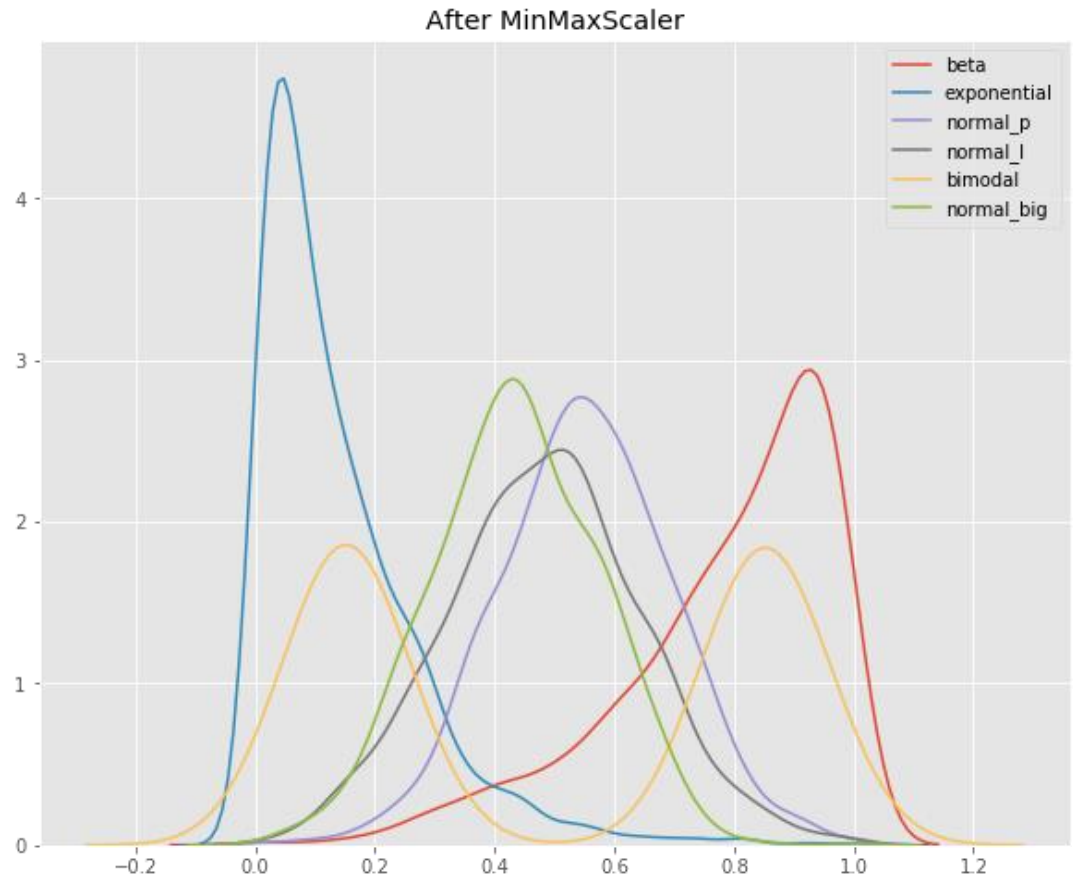
THE NORMALIZER

- The Normalizer normalises samples individually to unit norm.
- ***xi/ sqrt(x ^{2}i+y ^{2}i+z ^{2}i)***
- Each sample (i.e. each row of the data matrix) with at least one non zero component is rescaled independently of other samples so that its norm (*l1* or *l2*) equals one.
- The **normalizer** scales each value by dividing each value by its magnitude in *n*-dimensional space for *n* number of features.



THE MIN-MAX SCALER

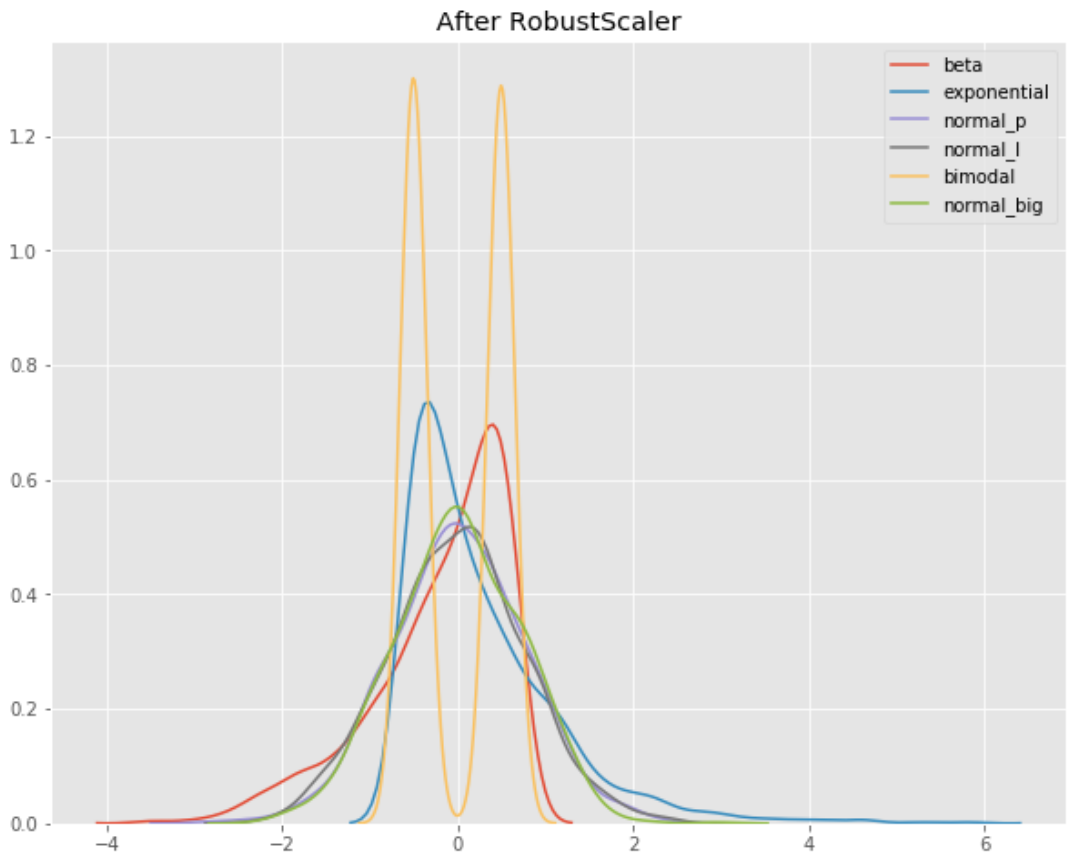
- The Min-Max Scaler uses the following formula for calculating each feature:
- ***(xi–min(x))/ (max(x)–min(x))***
- It transforms the data so that it's now in the range you specified. We specifiy the range by passing in a tuple to the **feature_range** parameter.
- By default, it transforms the data into a range between 0 and 1 (-1 and 1, if there are negative values).
- It can be used as an alternative to The Standard Scaler or when the data is not normally distributed.



1 SEP 2020

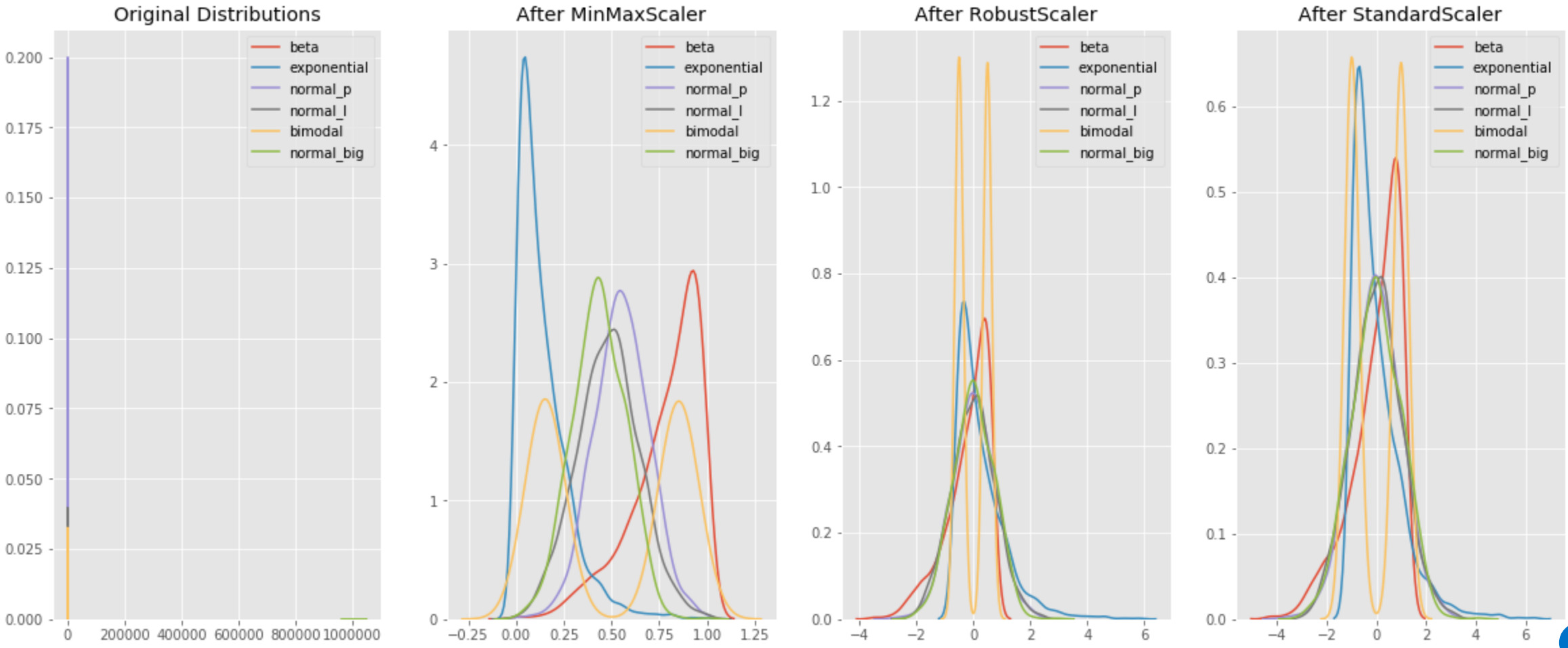
ROBUST SCALER

- Robust Scaler transforms the feature vector by subtracting the median and then dividing by the interquartile range (75% value — 25% value).
- Note that the range for each feature after Robust Scaler is applied is larger than it was for MinMax Scaler.
- Use Robust Scaler if you want to reduce the effects of outliers, relative to MinMax Scaler.



COMPARISON

Here are plots of the original distributions before and after MinMax Scaler, Robust Scaler, and Standard Scaler have been applied.



WRAP

- Use MinMax Scaler as the default if you are transforming a feature. It’s non-distorting.
- Use Robust Scaler if you have outliers and want to reduce their influence. However, you might be better off removing the outliers, instead.
- Use Standard Scaler if you need a relatively normal distribution.
- Use Normalizer sparingly — it normalizes sample rows, not feature columns. It can use l2 or l1 normalization.

HANDLE MISSING DATA

- Given the fact the missing values are unavoidable leaves us with the question of what to do when we encounter them. Ignoring the missing data is the same as digging holes in a boat; It will sink.
- The 2 most commonly recommended ways of dealing with missing data actually suck.
- They are:
- **Dropping** observations that have missing values
- **Imputing** the missing values based on other observations

Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40	nan	Yes
France	35	58000	Yes
Spain	nan	52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

TYPES OF IMPUTERS

- There are two types of imputers:
- Univariate: Uni implies one. Hence, this imputer fills the values in a feature by using the non-missing values from that feature.
- Multivariate: This uses the entire feature set to estimate the missing values.

TYPES OF IMPUTER

Univariate Imputation

- The **Simple Imputer** class provides basic strategies for imputing missing values.
- Missing values can be imputed with a provided constant value, or using the statistics(mean, median or most frequent) of each column in which the missing values are located.
- This class also allows for different missing values encodings.

Multivariate Imputation

- A more sophisticated approach is to use the **Iterative Imputer** class, which models each feature with missing values as a function of other features, and uses that estimate for imputation.
- It does so in an iterated round-robin fashion: at each step, a feature column is designated as output y and the other feature columns are treated as inputs X.
- A regressor is fit on (X, y) for known y. Then, the regressor is used to predict the missing values of y. This is done for each feature in an iterative fashion , and then is repeated for max-iter imputation rounds.
- The results of the final imputation round are returned.

HANDLING CATEGORICAL VARIABLES

- Handling categorical variables is another integral aspect of Machine Learning. Categorical variables are basically the variables that are discrete and not continuous. Ex — color of an item is a discrete variables where as its price is a continuous variable.
- Categorical variables are further divided into 2 types —
- **Ordinal categorical variables** — These variables can be ordered. Ex — Size of a T-shirt. We can say that M<L<XL.
- **Nominal categorical variables** — These variables can't be ordered. Ex — Color of a T-shirt. We can't say that Blue<Green as it doesn't make any sense to compare the colors as they don't have any relationship.

ONE HOT ENCODING

- One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

CompanyName	Categoricalvalue	Price
VW	1	20
Acura	2	10011
Honda	3	50000
Honda	3	10000

One hot encoding

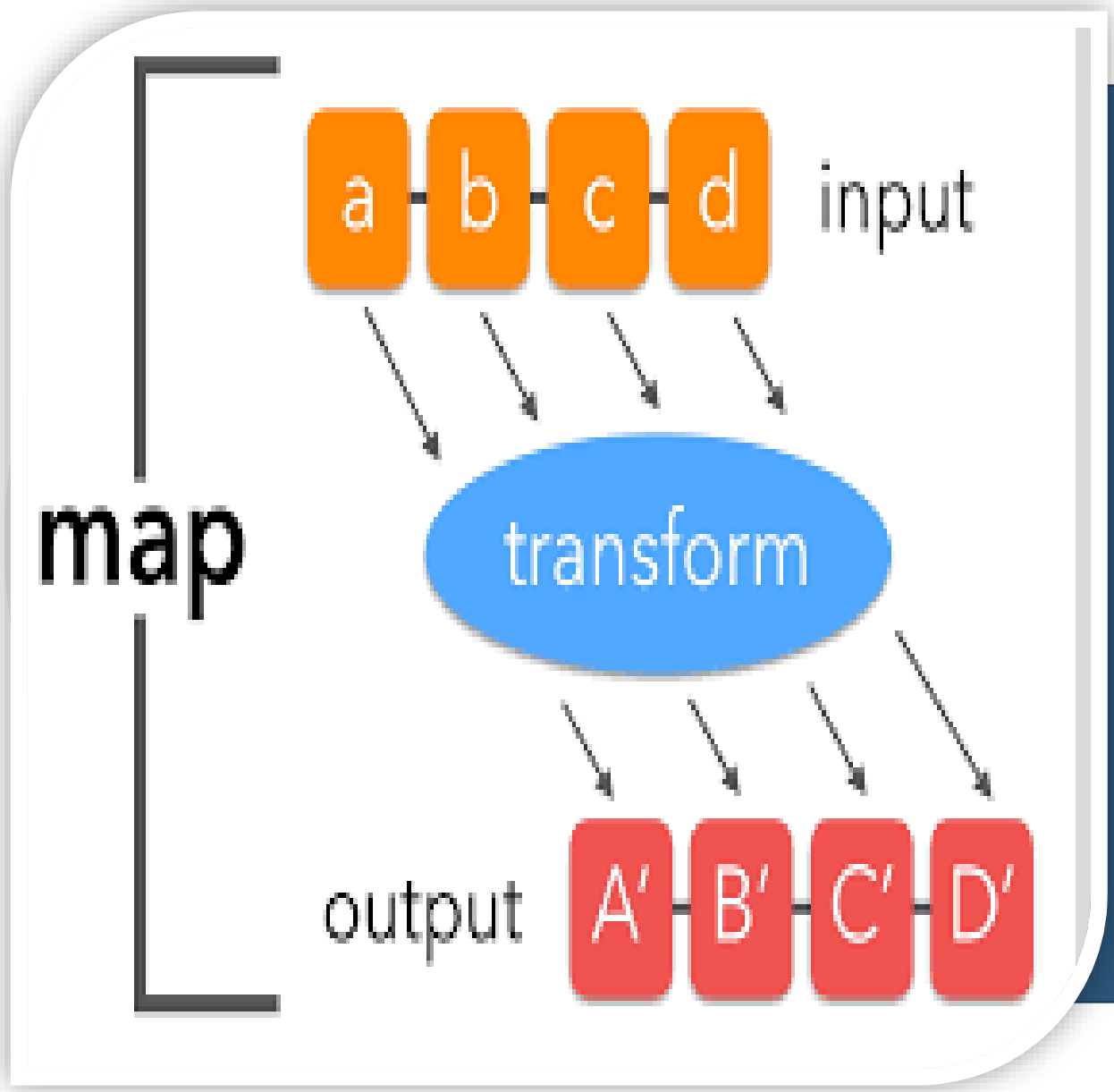
VW	Acura	Honda	Price
1	0	0	20000
0	1	0	10011
0	0	1	50000
0	0	1	10000

GET DUMMIES

The get_dummies() function is used to convert categorical variable into dummy/indicator variables.

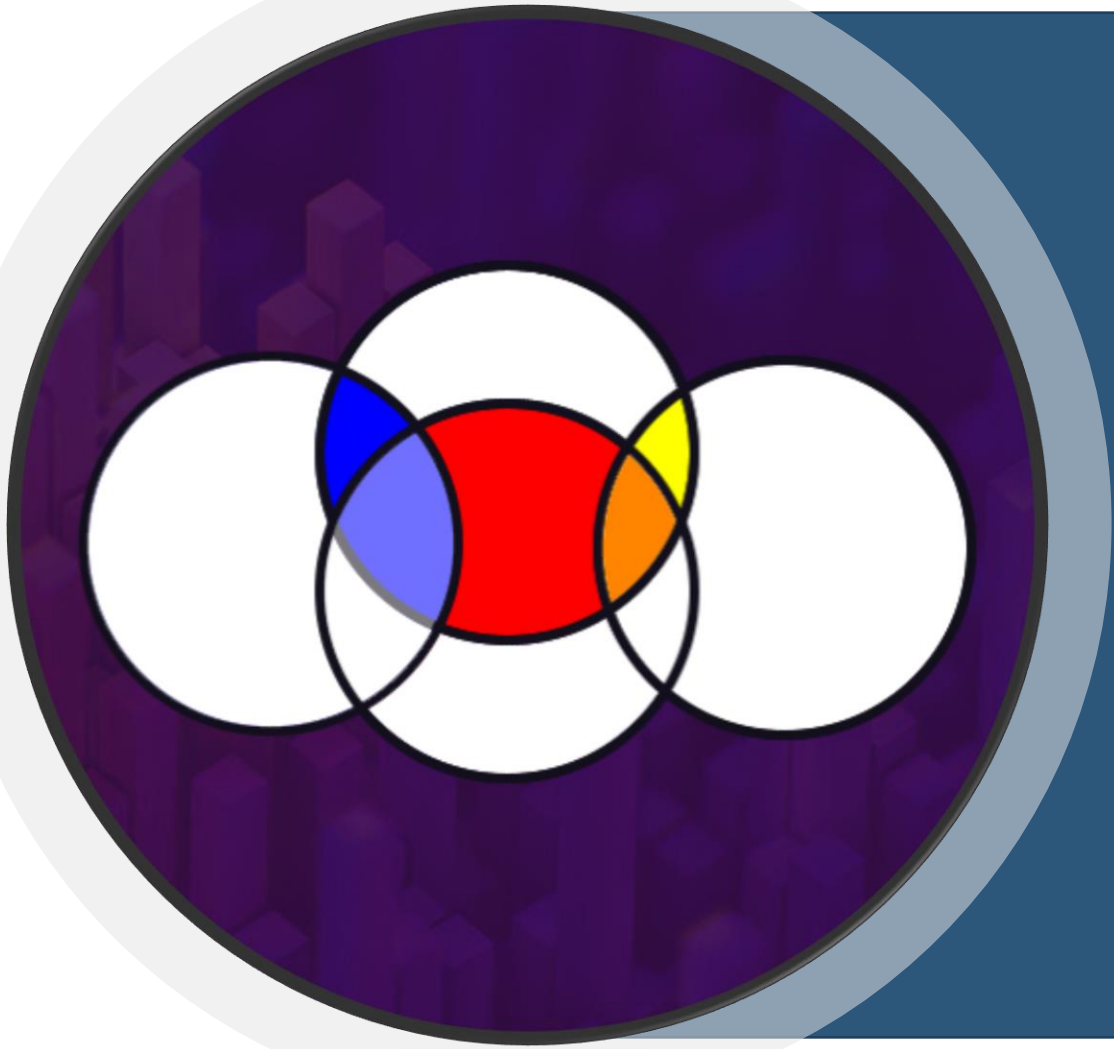
MAP FUNCTION

- The map() function applies a given function to each item of an iterable (list, tuple etc.) and returns a list of the results.



MULTICOLLINEARITY

- Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model.
- This means that an independent variable can be predicted from another independent variable in a regression model.
- This means that an independent variable can be predicted from another independent variable in a regression model.
- Multicollinearity may not affect the accuracy of the model as much. But we might lose reliability in determining the effects of individual features in your model – and that can be a problem when it comes to interpretability.



DETECTING MULTICOLLINEARITY USING VIF

- VIF determines the strength of the correlation between the independent variables. It is predicted by taking a variable and regressing it against every other variable.

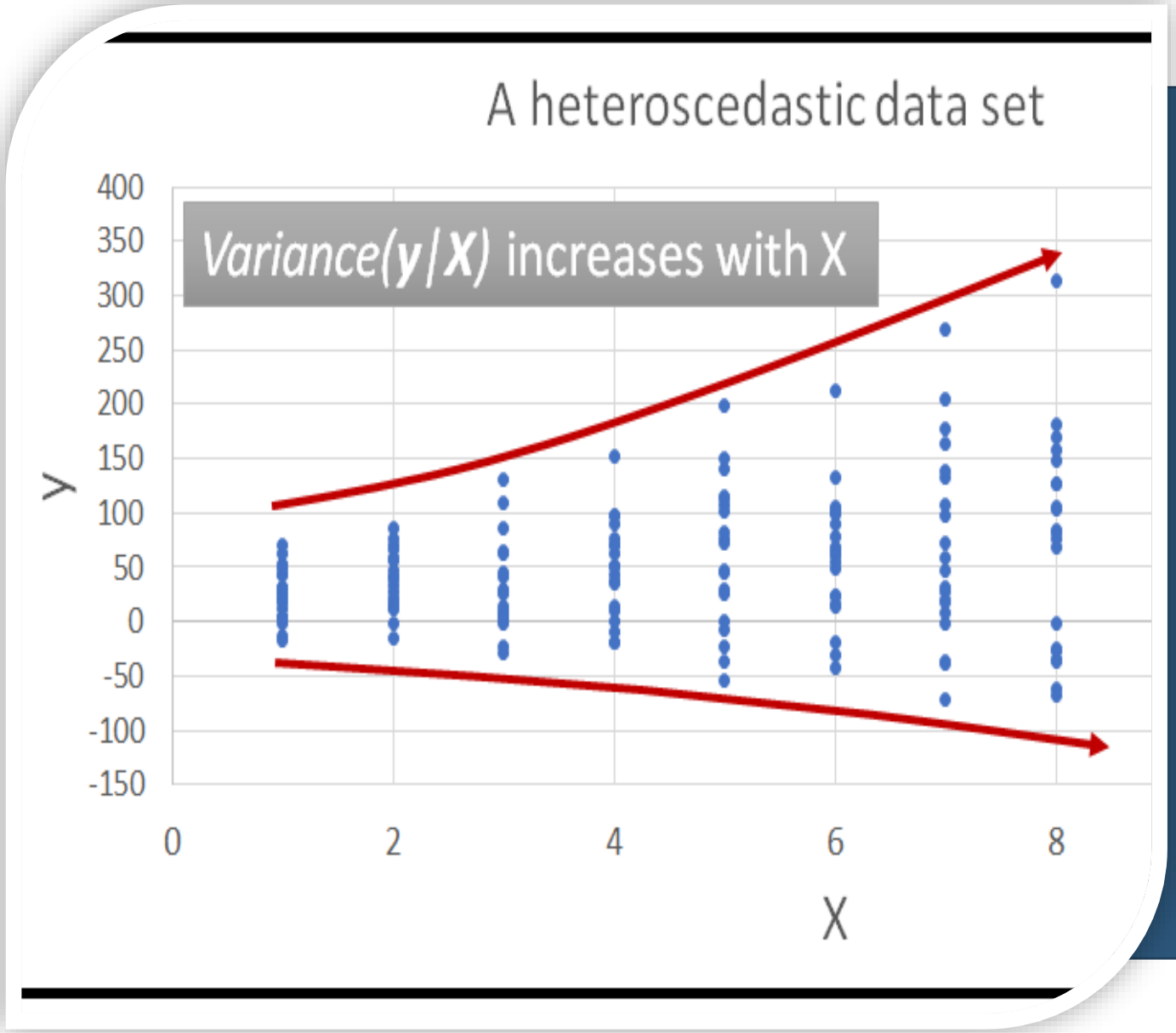
or

- VIF score of an independent variable represents how well the variable is explained by other independent variables.

$$VIF_i = \frac{1}{1 - R_i^2}$$

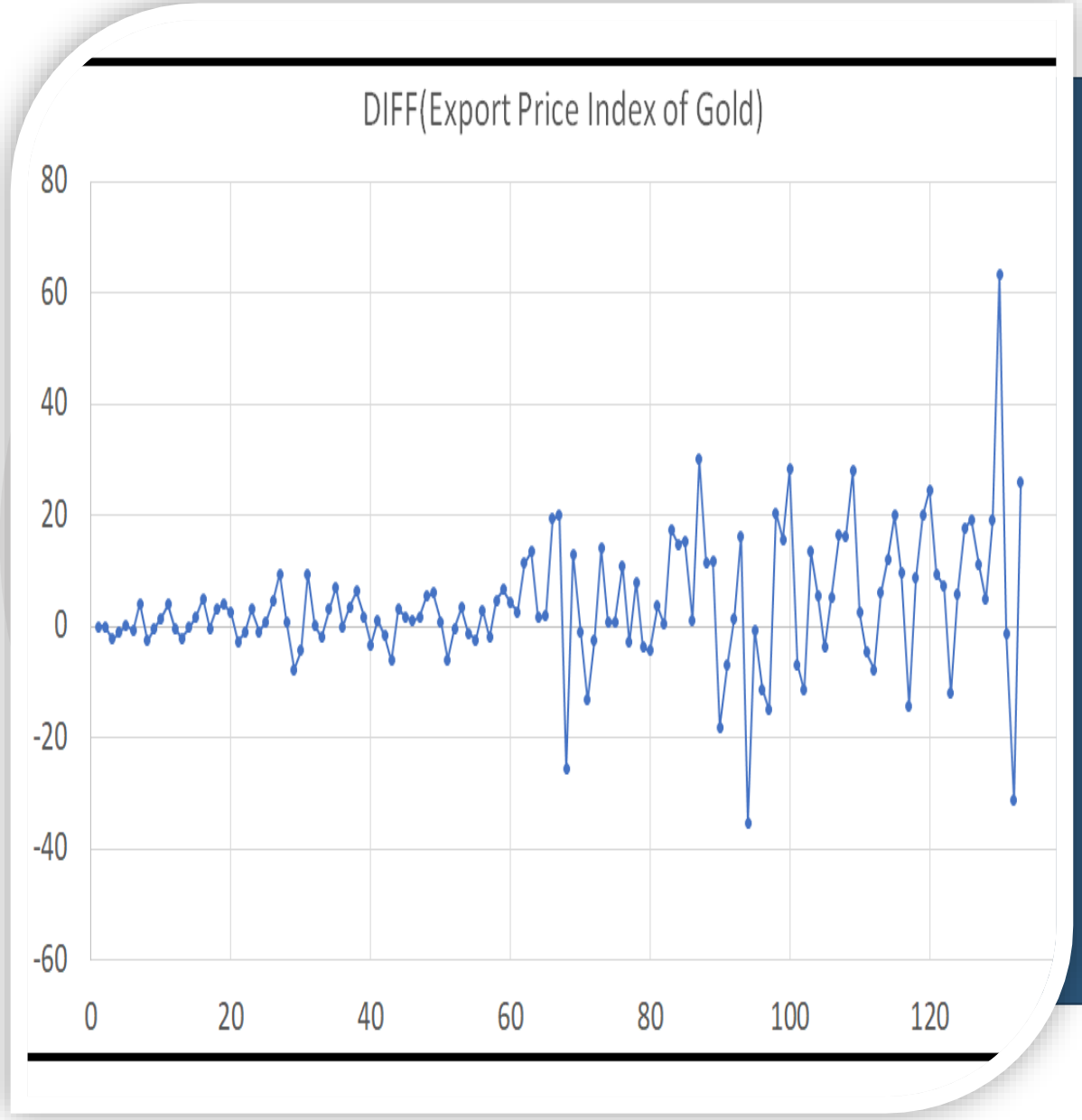
HETEROSCEDASTICITY

- Heteroscedasticity in the context of regression modeling, is what you have in your data when the *conditional variance* in your data is not constant.
- Conditional variance is the variability that you see in the dependent variable ***y*** for each value of the explanatory variables ***X***, or each value of time period ***t*** (in case of time series data).



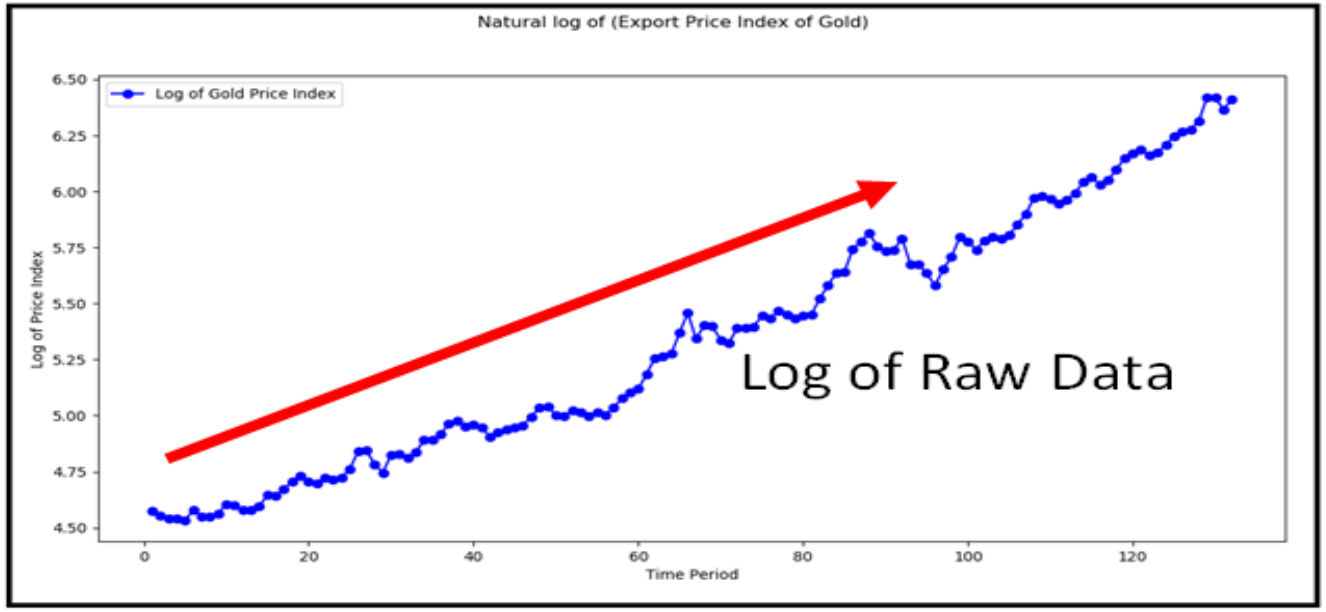
CAUSES AND FORMS OF HETEROSCEDASTICITY

- A common form of heteroscedasticity is when the amount of fluctuation is a fraction of the value. This happens commonly with monetary data such as:
- Prices (stock prices, prices of goods, medical procedure costs),
- Expenditures (household expenditure, employee wages, rents),
- Price indexes (the Gold Price Index example illustrated above).



HOW TO FIX THE PROBLEM:

- Log-transform the ***y*** variable to 'dampen down' some of the heteroscedasticity, then build an OLSR model for *log(**y**)*.
- Use a **G**eneralized **L**inear **M**odel (**GLM**) such as the Negative Binomial regression model which does not assume that the data set is homoscedastic. An NB regression model can work especially well if your data is discrete and non-negative.
- Use a **W**eighted **L**east **S**quares (**WLS**) or a **G**eneralized **L**east **S**quares (**GLS**) model two models that do not assume a homoscedastic variance. The *Python stats models* package supports both models in the statsmodels.api package.



BENEFITS OF DATA CLEANING

- **Higher Productivity :** Data is the most critical asset of any business. When the data is accurate, it increases the overall productivity of the company. It gets more comfortable for the sales team to target the relevant leads and convert them. Relevant data increases productivity and leads to higher ROI.
- ***Decision Making :*** It also helps the data scientists to analyse the data with more accuracy and help the companies in taking the crucial decisions

THANK YOU