

Steps in DS .

① Understanding the problem

② } Exploratory data analysis

③ } Visualization.

— Source of the data?

— Formats

— text

— Binary

— Files

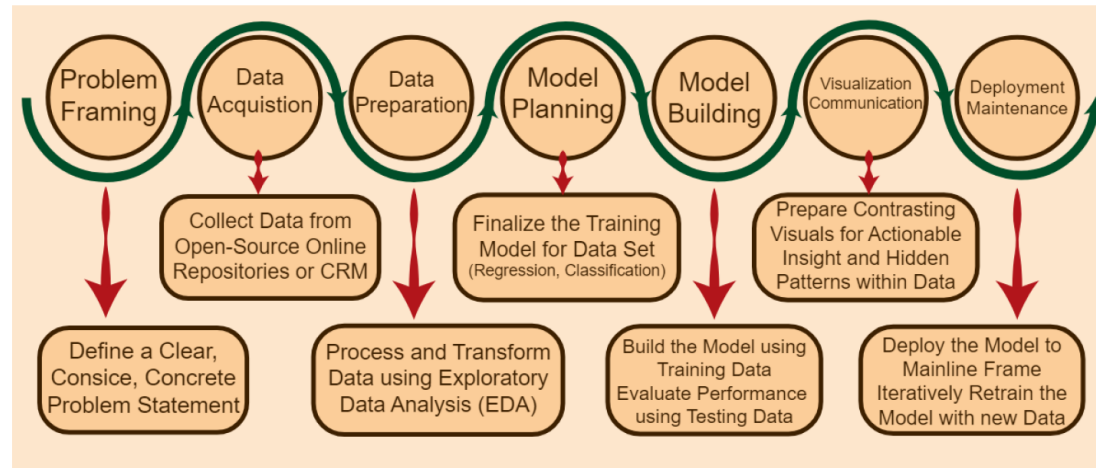
— Database

— from the internet.

— "VISUAL"

— "MATHEMATICAL"

"Problem understanding and clear definition" in data analysis is not necessarily a simple task, and it can be quite challenging. This step involves grasping the core problem that needs to be addressed using data analysis, and **it often requires a combination of skills including domain knowledge, communication skills, and critical thinking.**



Exploratory Data Analysis (EDA) is an approach used in statistics and data science to analyze and investigate data sets, with the goal of summarizing their main characteristics and discovering patterns and trends. It involves performing initial investigations on the data to gain **insights**, spot **anomalies**, test **hypotheses**, and **check assumptions**.

EDA typically employs **statistical graphics** and **data visualization** methods to visually represent the data.

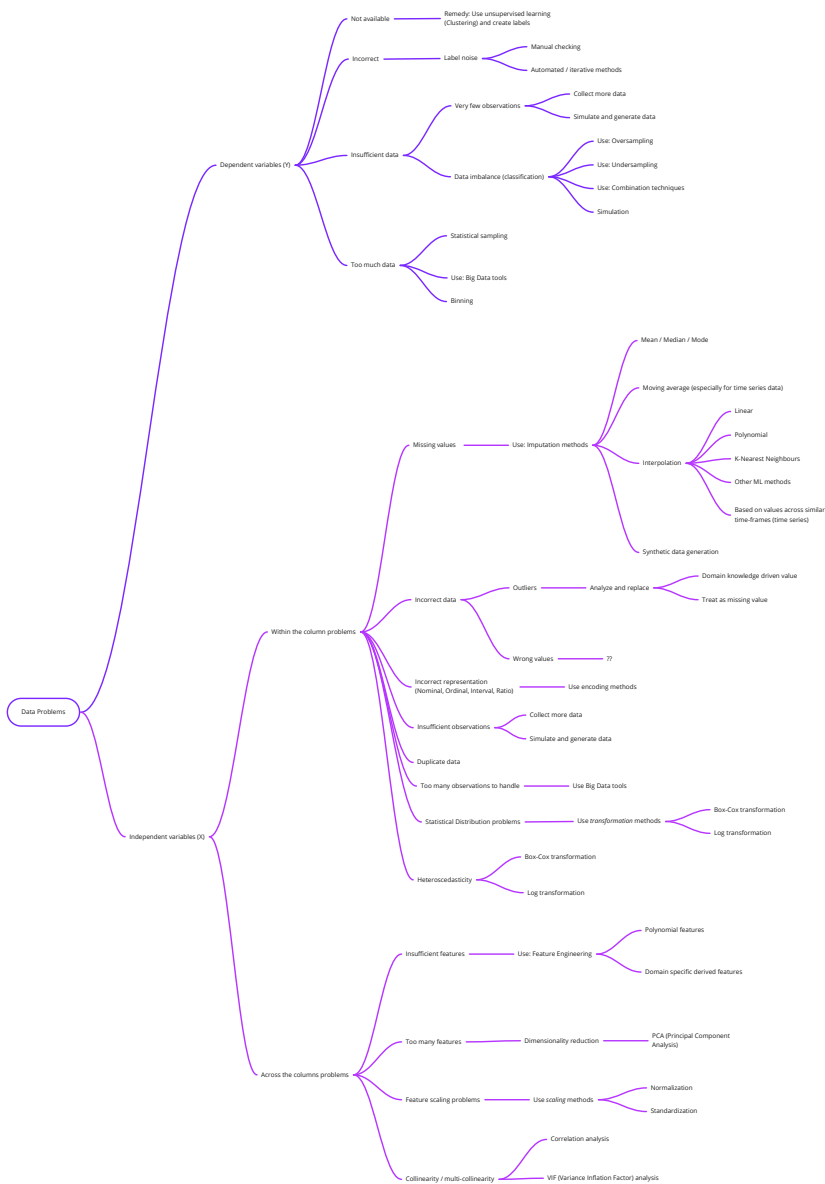
Some key points about exploratory data analysis:

1. **Purpose:** EDA is conducted to understand the data and gather insights before diving into more advanced analysis or modeling techniques.
2. **Data Exploration:** EDA involves exploring the data to identify **patterns, trends, outliers, and other features** that may be unexpected.
3. **Summary Statistics:** EDA often utilizes summary statistics, such as mean, median, mode, and skewness, to describe the central tendency and distribution of the data.
4. **Graphical Analysis:** Graphical representations, such as histograms, box plots, scatter plots, and Q-Q plots, are commonly used in EDA to visualize the data and identify relationships between variables.
5. **Data Cleaning:** EDA may also involve data cleaning and transformation processes to ensure the data is in a suitable format for analysis.

Some common data problems that can be revealed during EDA:

1. **Errors in the data:** EDA can help identify errors in the data, such as incorrect values, missing values, or inconsistencies.
2. **Outliers:** EDA can detect outliers, which are data points that significantly deviate from the rest of the data set.
3. **Unexpected patterns or trends:** EDA can uncover patterns or trends in the data that may be unexpected or contrary to initial assumptions.
4. **Variable relationships:** EDA can reveal relationships between variables, such as correlations or dependencies, which can provide insights into the data.
5. **Data inconsistencies:** EDA can identify inconsistencies in the data, such as duplicate records or conflicting information.

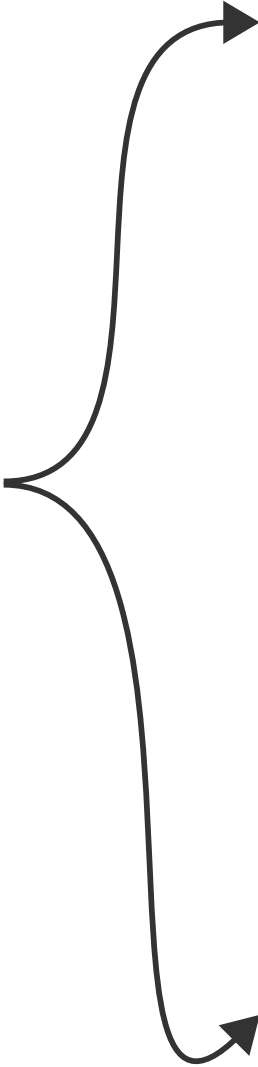
By conducting EDA, data practitioners can gain a better understanding of the data and address these data problems before proceeding with further analysis or modeling. This helps ensure the accuracy and reliability of the results obtained from the data.



Outlier handling is a part of the **Data Cleaning** process. Following is a list of common tasks involved in data cleaning:

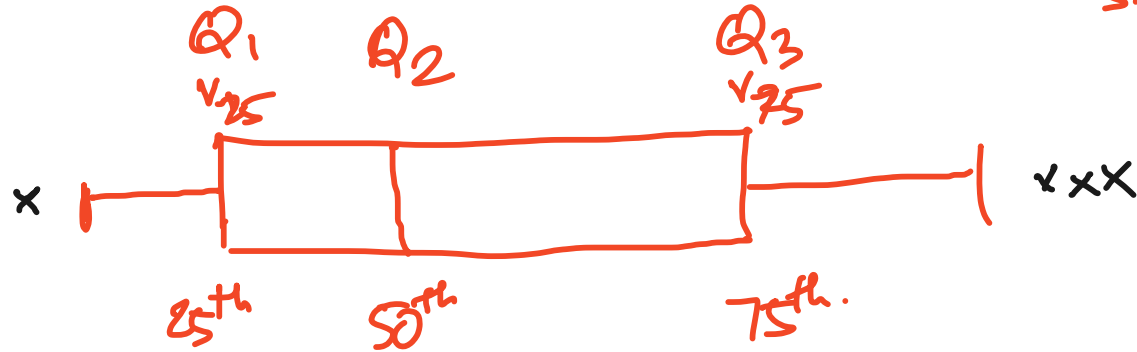
1. **Error correction:** Identifying and correcting errors in the data, such as misspellings, typographical errors, or incorrect values
2. **Duplicate removal:** Identifying and removing duplicate entries in the data set to avoid redundancy and ensure data integrity
3. **Missing data handling:** Dealing with missing data by either **imputing** values or deciding how to handle the missing data in the analysis
4. **Standardization (Data Scaling):** Ensuring consistency in the format, units, and representation of data across the data set
5. **Normalization (Data Scaling):** A data transformation technique that scales values within a range (often 0 to 1) to maintain relative relationships between features.
6. **Outlier detection and treatment:** Identifying outliers, which are extreme values that deviate significantly from the rest of the data, and deciding how to handle them, such as removing them or replacing them with appropriate values

Data cleaning is a crucial step in the data analysis process as it helps to ensure the accuracy and reliability of the results obtained from the data. It requires careful attention to detail and an understanding of the specific data set and its characteristics. By performing data cleaning, data practitioners can improve the quality of the data and make more informed decisions based on reliable and trustworthy information.



Column	Min	Max	Mean
c3	0	187.0776	170.3037
c4	0	174.0258	163.9285
c5	0	0.915783	0.529769
c6	0	3.144694	1.5678
c7	0.447406	2.971087	2.426871
c8	0	22.69527	21.20686
c9	0	10.27313	7.427598
c10	0	0.785255	0.642268
c11	0.002404	59.08272	56.35317
c12	0	13.7972	11.64599
c13	0	36.43647	32.86705
c14	0	5.409382	0.342937
c15	0	5.36033	1.531901
c16	0.019261	21.16151	17.79199
c17	0	50.36344	28.20165
c18	0	41.90749	30.87445
c19	0	17.54499	12.88846
c20	0	11.33466	6.722164
c21	-0.94329	11.36428	7.921272
c22	0	7.757059	4.729896
c23	14.98597	48.65412	42.24162
c24	0	0.915783	0.529769
c25	0	3.144694	1.5678

Outliers.



Create descriptive Statistics.

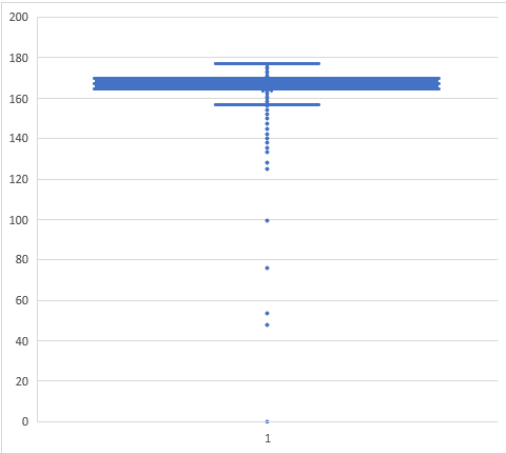
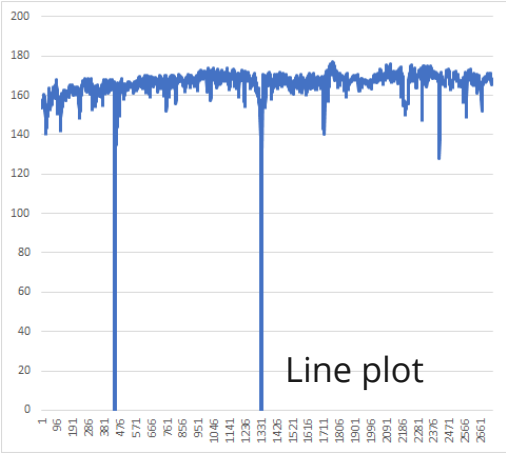
- Mean
- Median
- mode
- Std. dev / var.
- Skewness
- Kurtosis -
- 25^{th} percentile
- 50^{th} " (Median)
- 75^{th} "

$$\begin{aligned} I.Q.R &= Q_3 - Q_1 \\ &= (V_{75}) - (V_{25}) \end{aligned}$$

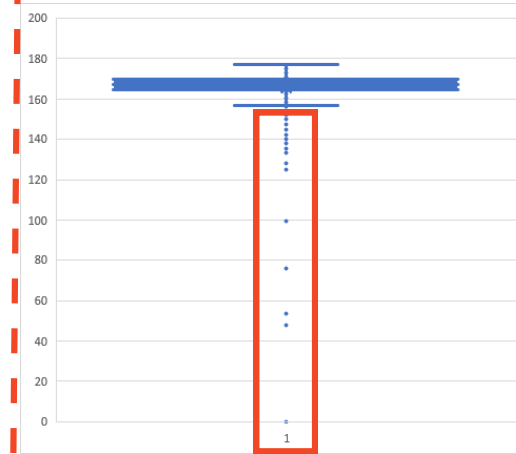
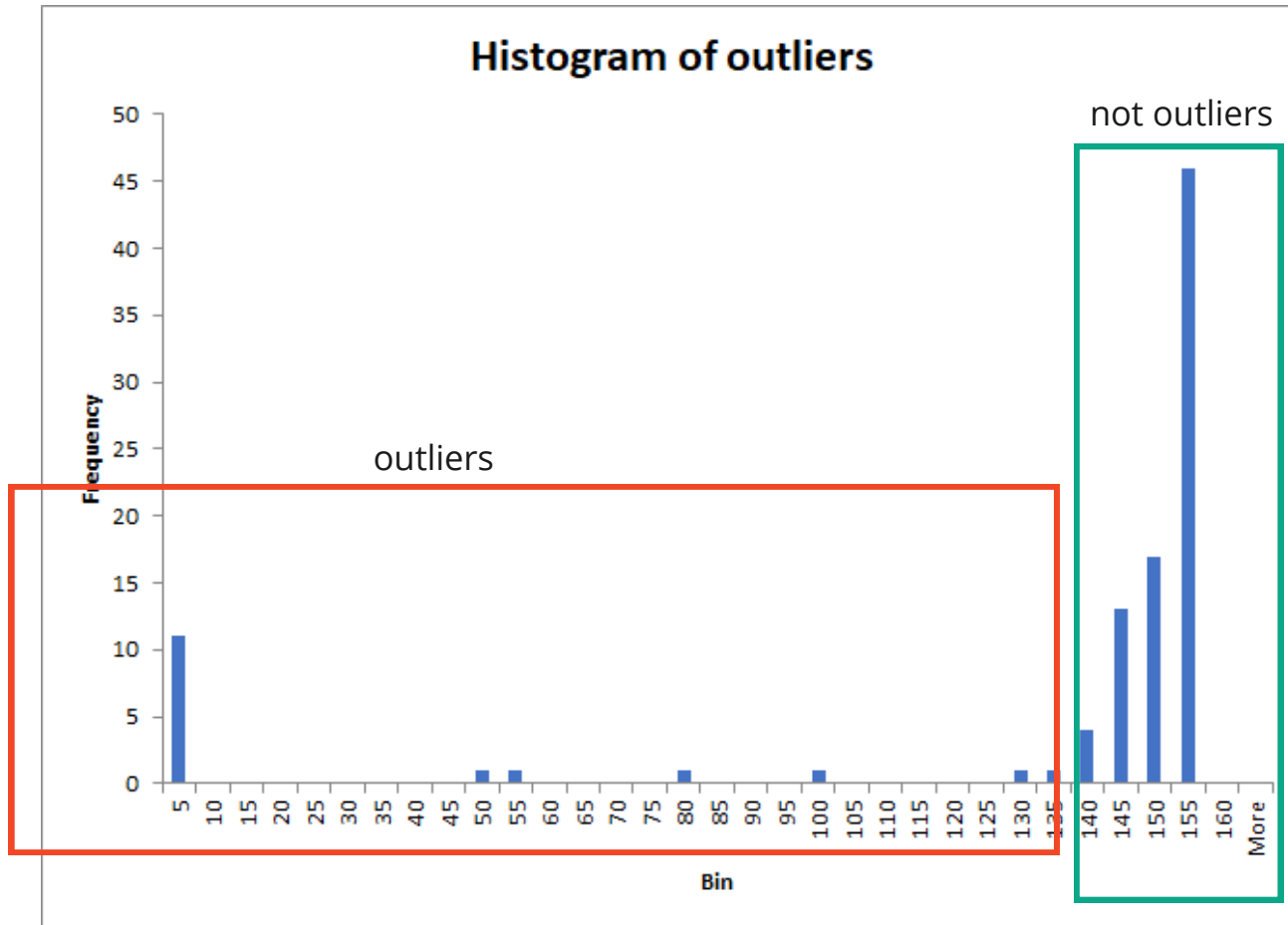
$$LB = Q_1 - 1.5 \times IQR$$

$$HB = Q_3 + 1.5 \times IQR$$

	A	B	C	D	E	F
6	c1	c2	c3	c4	c5	c6
7	01-Sep-13 00:00:00	1	161.1432	153.8311	0.63172	2.414032
8	02-Sep-13 00:00:00	1	164.1032	156.9562	0.656179	2.355487
9	03-Sep-13 00:00:00	1	165.3228	157.8616	0.639334	2.701887
10	04-Sep-13 00:00:00	1	165.7317	158.027	0.652345	2.459543
11	05-Sep-13 00:00:00	1	162.6875	155.1138	0.657465	1.909607
12	06-Sep-13 00:00:00	1	162.6016	154.7801	0.668482	1.52945
13	07-Sep-13 00:00:00	1	162.4659	154.8351	0.740854	1.409602
14	08-Sep-13 00:00:00	1	166.0759	158.1119	0.782362	1.449534
15	09-Sep-13 00:00:00	1	165.2525	157.0692	0.746412	1.515374
16	10-Sep-13 00:00:00	1	165.241	157.0487	0.745851	1.655467
17	11-Sep-13 00:00:00	1	165.4747	157.125	0.792748	1.573377
18	12-Sep-13 00:00:00	1	165.2264	157.442	0.788428	1.736178
19	13-Sep-13 00:00:00	1	165.9279	158.3007	0.730994	1.798794
20	14-Sep-13 00:00:00	1	168.2197	160.6276	0.618005	2.271551
21	15-Sep-13 00:00:00	1	166.0266	158.6136	0.574357	2.471939
22	16-Sep-13 00:00:00	1	166.8304	159.5535	0.603332	2.374928
23	17-Sep-13 00:00:00	1	164.1426	156.4439	0.592301	2.123248
24	18-Sep-13 00:00:00	1	166.4962	158.1226	0.612283	2.423496
25	19-Sep-13 00:00:00	1	162.8119	154.3308	0.574474	2.519138
26	20-Sep-13 00:00:00	1	160.8477	152.1846	0.386043	2.757094
27	21-Sep-13 00:00:00	1	155.8484	148.1267	0.39832	2.789573
28	22-Sep-13 00:00:00	1	148.2437	141.4792	0.415291	2.766708
29	23-Sep-13 00:00:00	1	147.8628	140.7158	0.377464	2.763958
30	24-Sep-13 00:00:00	1	151.7811	143.3661	0.473188	2.586392
31	25-Sep-13 00:00:00	1	154.0191	145.4945	0.507609	2.561311
32	26-Sep-13 00:00:00	1	153.9183	145.2496	0.508395	2.085402



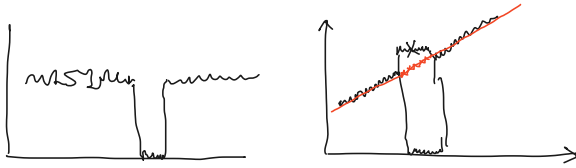
c4	
Mean	163.9284816
Standard Error	0.363181665
Median	166.1539939
Mode	0
Standard Deviation	14.9875705
Sample Variance	224.6272695
Kurtosis	94.71693965
Skewness	-9.197660397
Range	174.0257593
Minimum	0
Maximum	174.0257593
Sum	279170.2041
Count	1703
Quartiles	
Q1 (25th percentile)	163.2876459
Q2 (50th percentile)	166.1539939
Q3 (75th percentile)	168.4890902
IQR	5.201444298
LB	155.4854794
UB	176.2912566



Outlier analysis

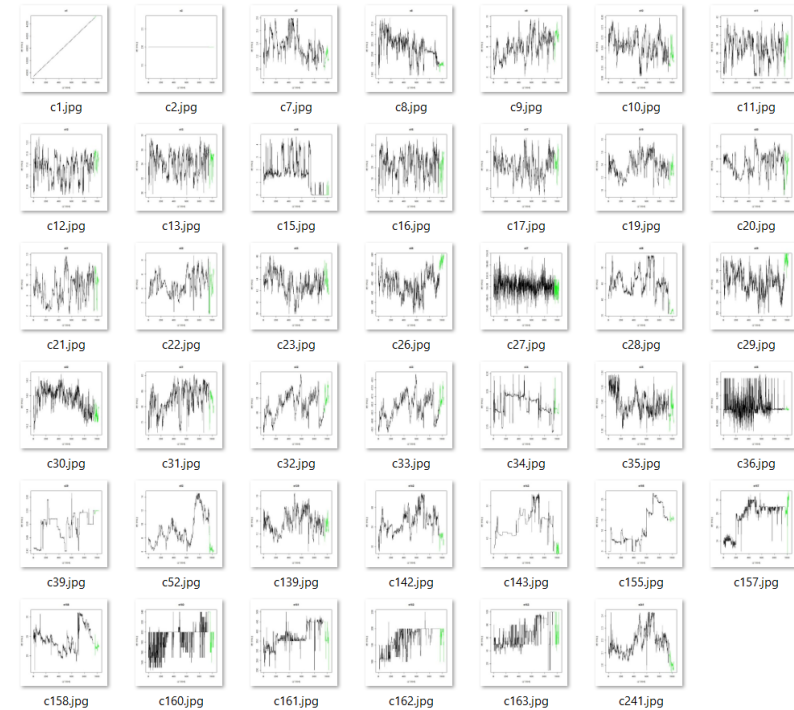
One way to further analyze the outliers and decide the cut-off bounds
(after applying the standard outlying detecting rule)

Strategies for replacing missing values / replacing outliers

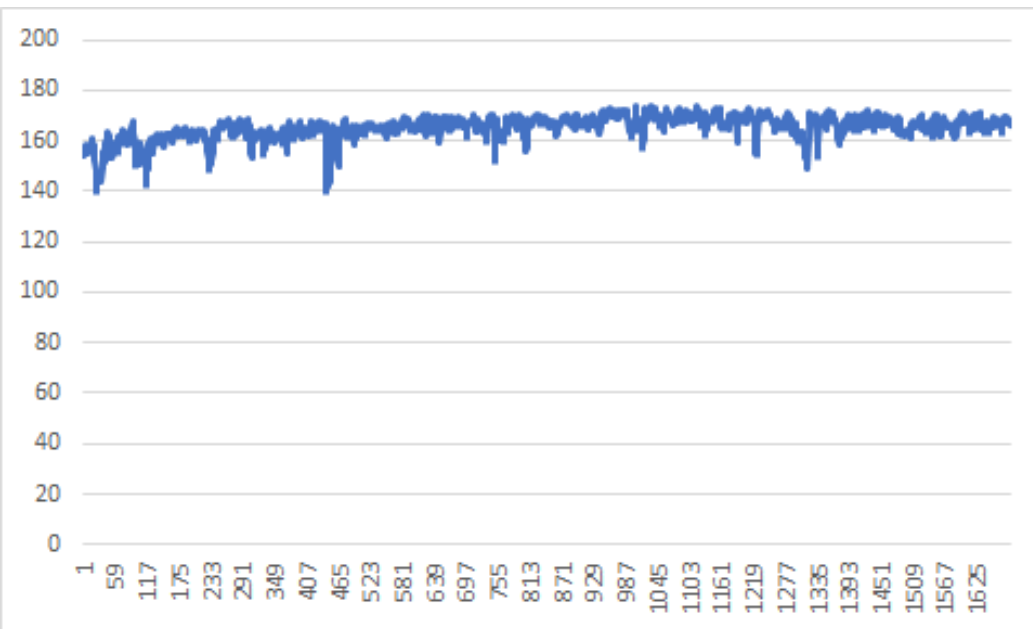
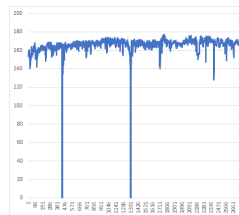


There are several strategies for replacing outliers during Exploratory Data Analysis (EDA). Here are some of the most accepted strategies:

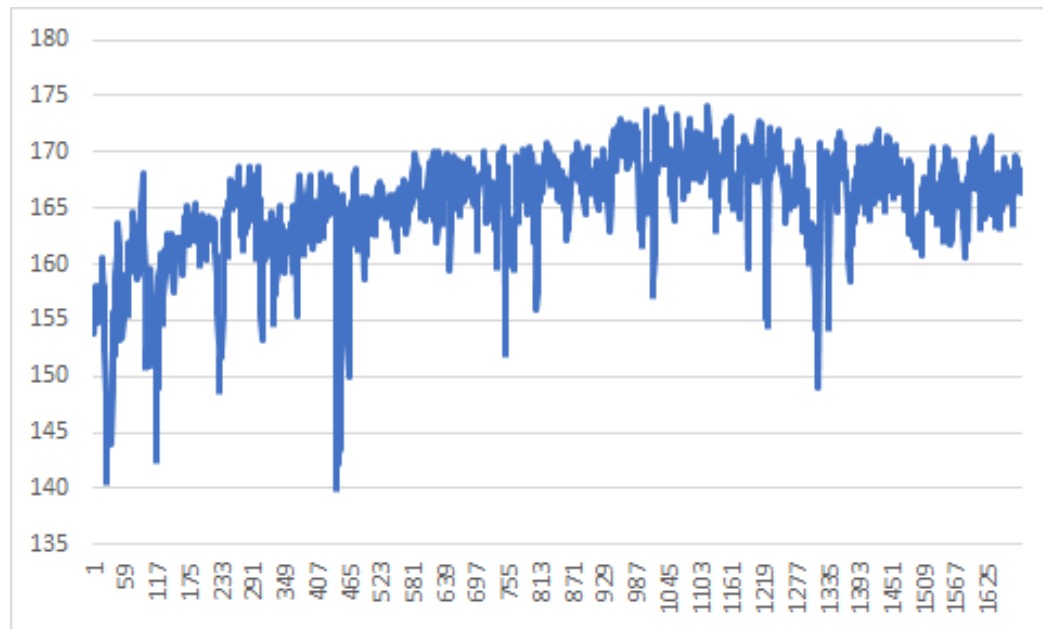
1. **Winsorization:** This method replaces the extreme values with the nearest “good” data point, rather than truncating them completely
2. **Imputation:** Imputation involves replacing the outlier values with a reasonable estimate, such as the mean or median of the data set
3. **Trimming:** Trimming involves removing the outliers from the data set altogether
4. **Cap the data:** This method involves setting a cap on the maximum and minimum values of the data, so that extreme values are replaced with the maximum or minimum value
5. **Use robust estimation techniques:** Robust estimation techniques, such as certain kinds of regression that are less sensitive to outliers



Data trends are important while deciding outlier / missing data handling and data imputation methods to be used

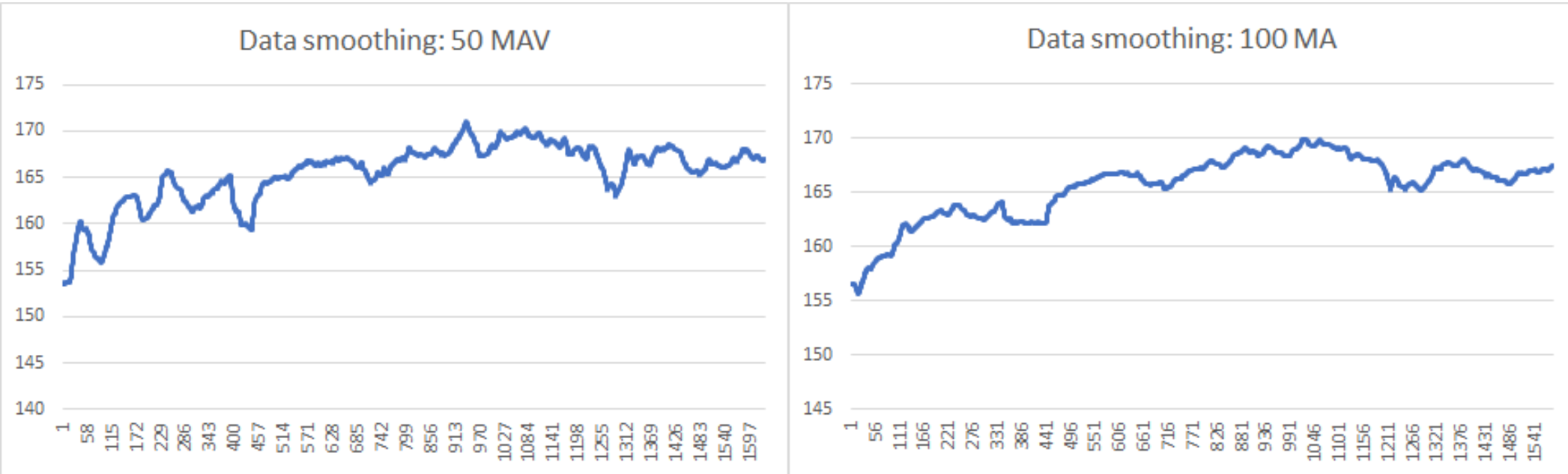


After dropping the outliers



After re-scaling the display

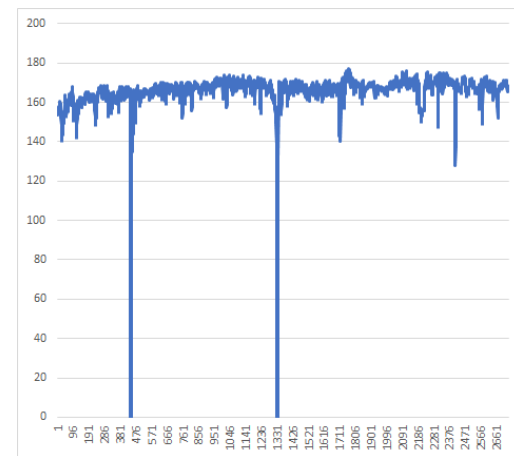
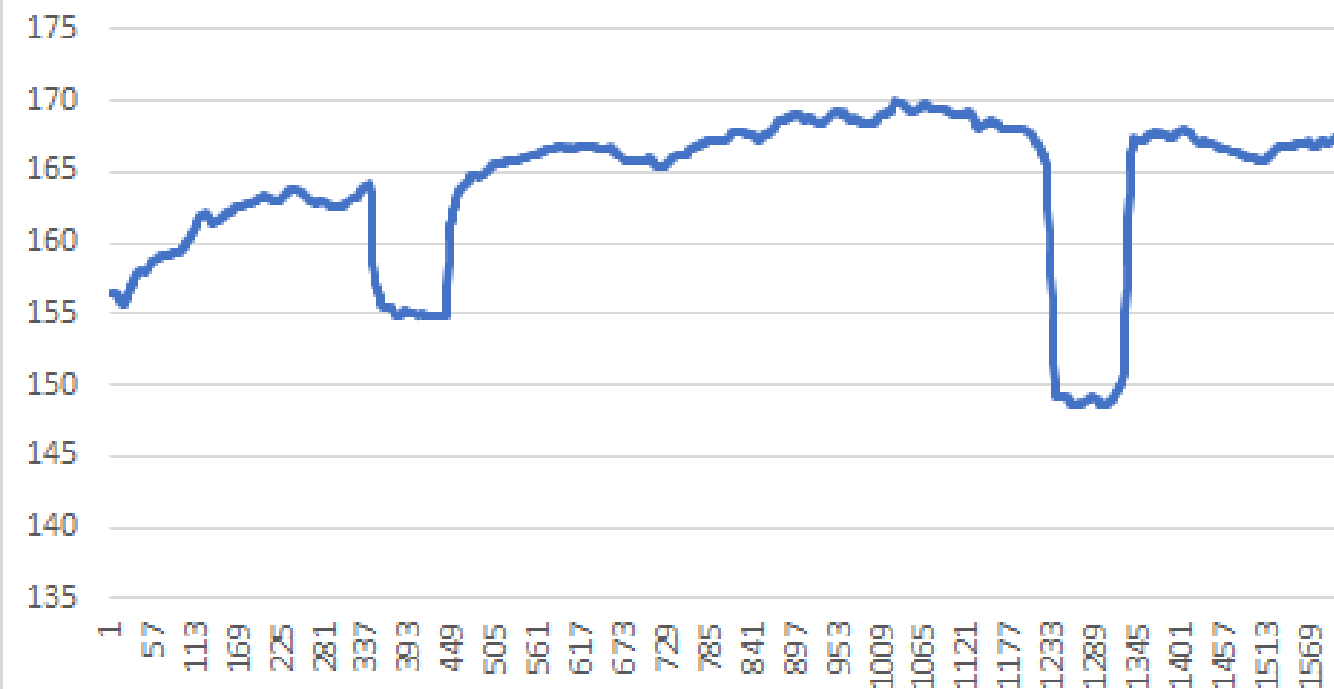
Moving Averages for Data Smoothing



Moving averages can be used for:

- Filling up missing values
- Replacing outliers
- Eliminating noise
- Understanding data trends

Effect of outliers on MA of raw data



Data Scaling: Impact of scale differences

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	y	x1	x2	x3	x4	Random		SUMMARY OUTPUT						
2	0.038117	0	26	0	0	26								
3	0.896468	0.005556	24.00003	1.71E-07	9.53E-10	24		Regression Statistics						
4	0.159546	0.011111	22.00012	1.37E-06	1.52E-08	22		Multiple R	0.894828832					
5	0.863764	0.016667	28.00028	4.63E-06	7.72E-08	28		R Square	0.800718639					
6	1.106349	0.022222	24.00049	1.1E-05	2.44E-07	24		Adjusted R Square	0.796189517					
7	1.010169	0.027778	25.00077	2.14E-05	5.95E-07	25		Standard Error	0.341093776					
8	0.278498	0.033333	23.00111	3.7E-05	1.23E-06	23		Observations	181					
9	1.114231	0.038889	23.00151	5.88E-05	2.29E-06	23								
10	1.029804	0.044444	28.00198	8.78E-05	3.9E-06	28		ANOVA						
11	0.37387	0.05	28.0025	0.000125	6.25E-06	28			df	SS	MS	F	Significance F	
12	0.971634	0.055556	28.00309	0.000171	9.53E-06	28		Regression	4	82.27607	20.56902	176.7934	1.61E-60	
13	0.975377	0.061111	24.00373	0.000228	1.39E-05	24		Residual	176	20.47671	0.116345			
14	1.079774	0.066667	22.00444	0.000296	1.98E-05	22		Total	180	102.7528				
15	1.24279	0.072222	23.00522	0.000377	2.72E-05	23								
16	0.644699	0.077778	25.00605	0.000471	3.66E-05	25			Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
17	0.656177	0.083333	24.00694	0.000579	4.82E-05	24		Intercept	0.675715271	0.333001	2.029167	0.043948	0.018526	1.332905
18	1.095492	0.088889	27.0079	0.000702	6.24E-05	27		x1	3.263090159	0.43373	7.523326	2.64E-12	2.40711	4.119071
19	1.115275	0.094444	28.00892	0.000842	7.96E-05	28		x2	0.005231504	0.012756	0.410128	0.682211	-0.01994	0.030405
20	1.512548	0.1	26.01	0.001	0.0001	26		x3	-26.52365687	1.766408	-15.0156	2.43E-33	-30.0097	-23.0376
21	0.639396	0.105556	28.01114	0.001176	0.000124	28		x4	23.39934766	1.499779	15.60186	5.09E-35	20.43948	26.35921
22	1.406627	0.111111	24.01235	0.001372	0.000152	24								
23	1.172479	0.116667	23.01361	0.001588	0.000185	23								
24	0.909356	0.122222	24.01494	0.001826	0.000223	24								

In general, any machine learning algorithm that relies on distance measures or similarity measures between data points is sensitive to the scale of the features. Therefore, it is important to scale the data appropriately before applying these algorithms.

1. **k-Nearest Neighbors (k-NN):** k-NN is sensitive to the scale of the features, as it relies on distance measures between data points.
2. **Support Vector Machines (SVM):** SVM is sensitive to the scale of the features, as it tries to maximize the margin between the decision boundary and the support vectors.
3. **Linear Regression:** Linear regression is sensitive to the scale of the features, as it tries to minimize the sum of squared errors between the predicted and actual values.
4. **Neural Networks:** Neural networks can be sensitive to the scale of the features, as large differences in the scale of the input features can lead to slow convergence or poor performance.
5. **Principal Component Analysis (PCA):** PCA is sensitive to the scale of the features, as it tries to find the directions of maximum variance in the data.

ML algorithms NOT susceptible to data scaling issues

1. **Tree-based algorithms:** Tree-based algorithms, such as decision trees and random forests, are fairly insensitive to the scale of the features.
2. **Naive Bayes:** Naive Bayes is less sensitive to the scale of the features, as it calculates probabilities based on the frequency of each feature.
3. **Ensemble methods:** Ensemble methods, such as bagging and boosting, can be less sensitive to the scale of the features due to their ability to combine multiple models.
4. **Deep Learning:** Deep learning algorithms can be less sensitive to the scale of the features due to their ability to learn hierarchical representations of the data.

Scaling Method	Description	Use Cases / Scenarios
Standardization	Scales data to have mean 0 and standard deviation 1.	Suitable for algorithms assuming normal distribution, SVMs.
Normalization Min-Max Scaler	Scales data to the range [0, 1] while preserving relative relationships.	Useful when features have varying ranges, distance-based algorithms.
Max Abs Scaler	Scales data by dividing by the maximum absolute value, preserves sparsity.	For sparse data, centered at zero, not sensitive to outliers.
Robust Scaler	Scales data using median and interquartile range to handle outliers.	When data contains outliers, preserving feature distributions.
Quantile Transformer	Transforms data to follow a uniform or normal distribution.	Mitigating impact of outliers, making data distribution more Gaussian-like.
Power Transformation Box-Cox Transformation	Applies power transformations to make data distributions more Gaussian-like.	For skewed data, making it more suitable for Gaussian-based models.
Unit Vector Scaling	Scales each feature by dividing by its magnitude, ensuring vectors have unit norm.	Useful for algorithms that rely on vector distances.
Log Transformation	Applies logarithmic transformation to data to compress large ranges.	Reducing impact of extremely large values, skewed data.
Mean Centering	Subtracts the mean of each feature from its values, resulting in mean-centered data.	When you want data centered around zero.

Box-Cox Transform

Original 'x' → transformed 'x'

Box-Cox Transform

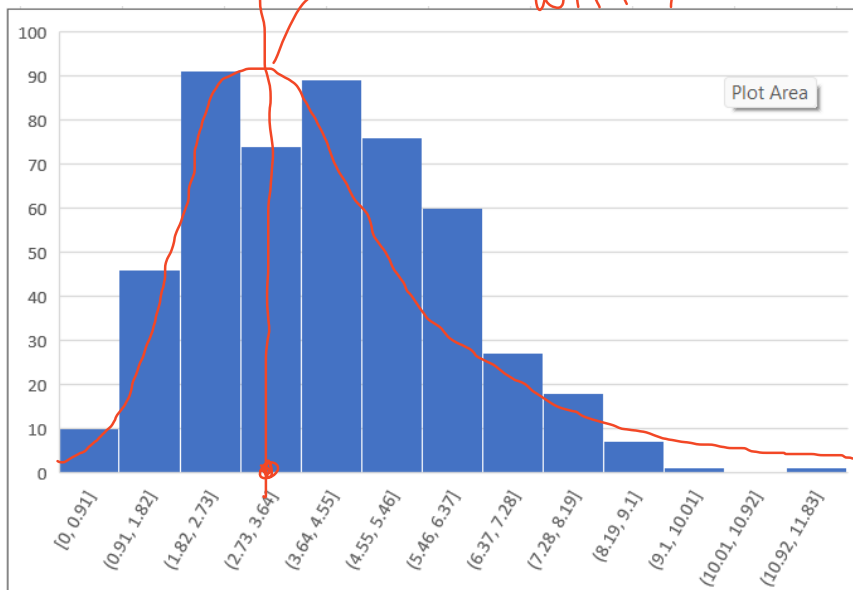
$$X(\lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$

① Removes 'Skew' from the data.
② Makes the Variance 'uniform'.

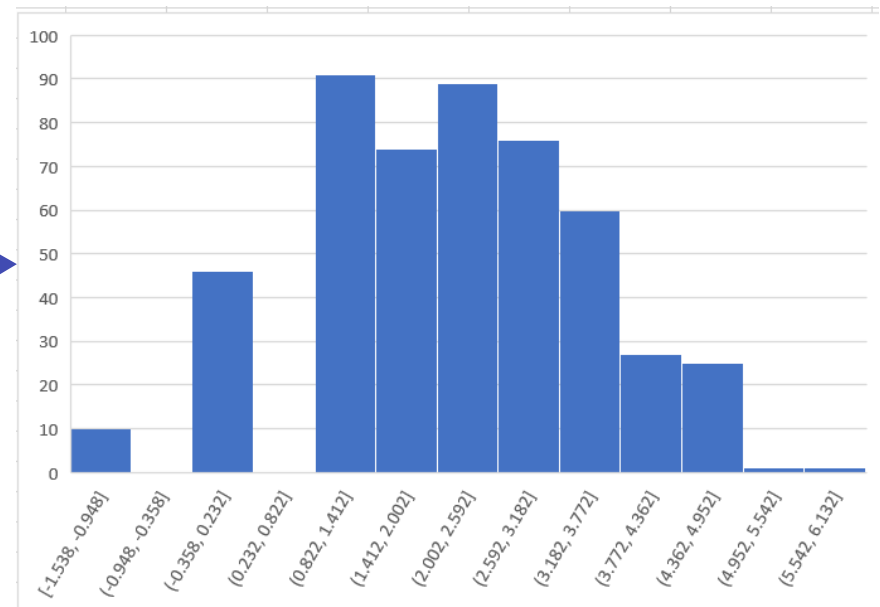
$\lambda \in \{-5, \dots, +5\}$ } iterative process.

" MAXIMUM LIKELIHOOD ESTIMATION "

POISSON DISTRIBUTION
with $\lambda =$



Normal Distribution

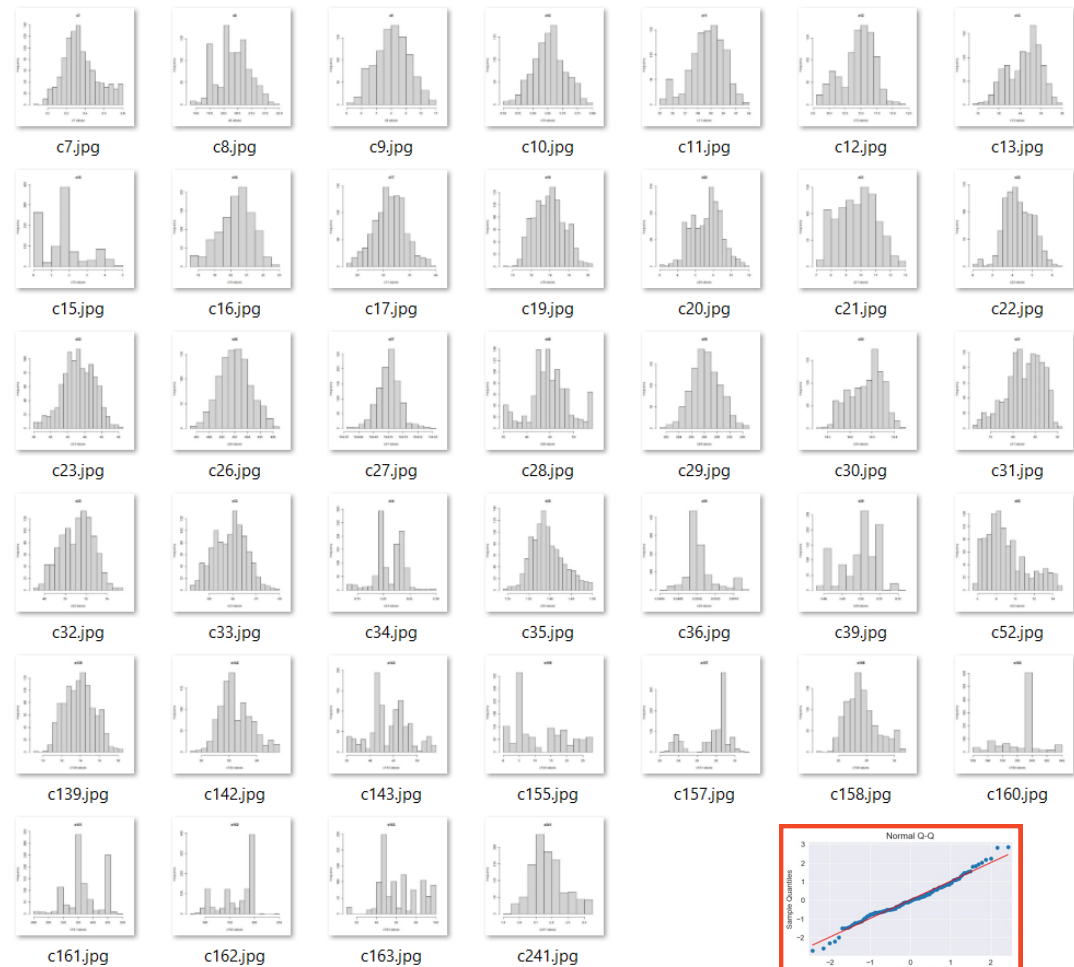


Normalization:

- Use normalization when the distribution of the data is not Gaussian or when you have varying scales in your data.
- Normalize the data when the algorithm you are using does not make assumptions about the data distribution, such as k-nearest neighbors or artificial neural networks.
- Normalize the data when you want to bring all variables to the same range and preserve the shape and distribution of the data.

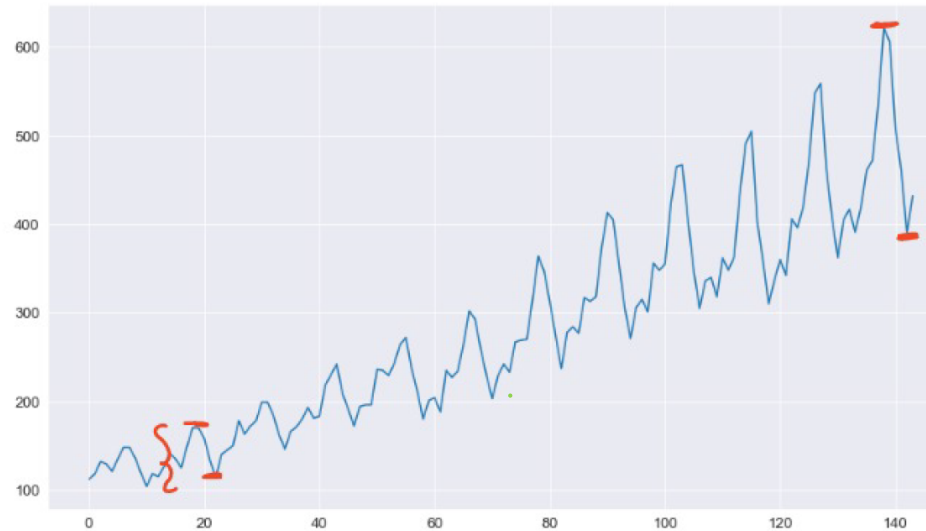
Standardization:

- Use standardization when the data follows a Gaussian distribution or when you see a bell-curve in your data.
- Standardize the data when you want to transform it to have a mean of 0 and a standard deviation of 1.
- Standardization is beneficial when dealing with unsupervised learning algorithms or when your dataset has extreme high or low values (outliers).



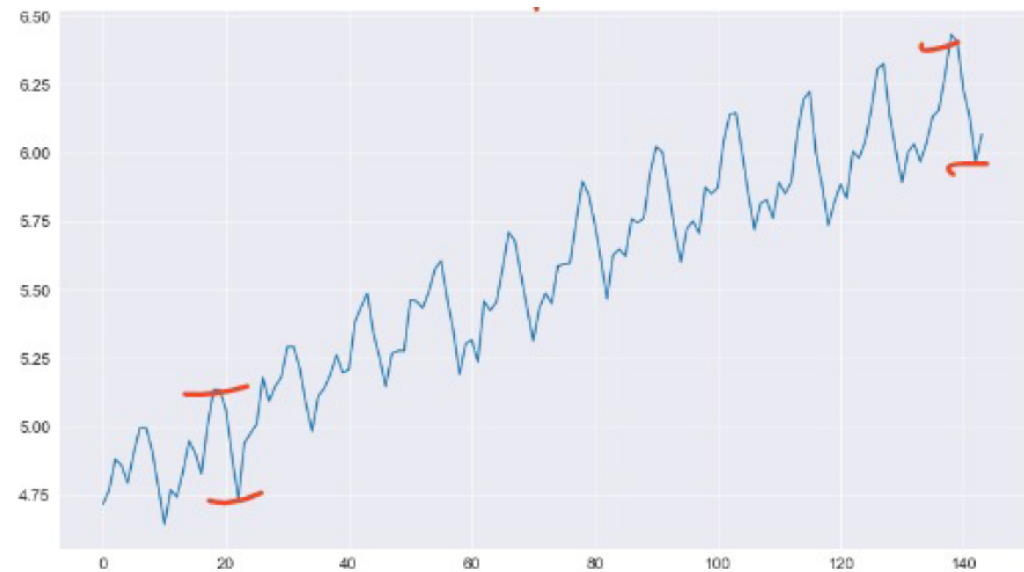
The distribution of data assumes significance in the context of **data scaling**

Log transformation applied to data prior to Time-series analysis

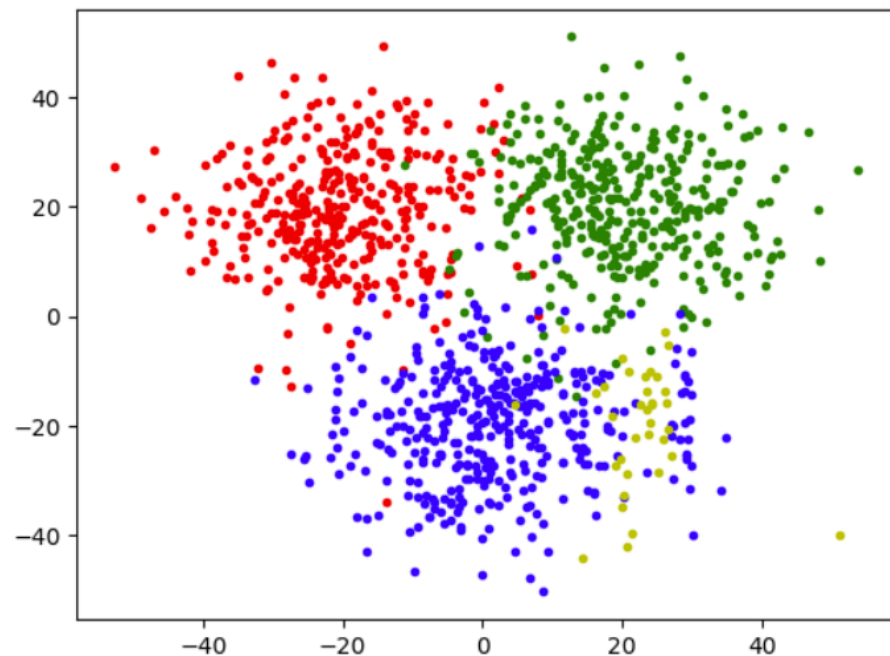


Original data

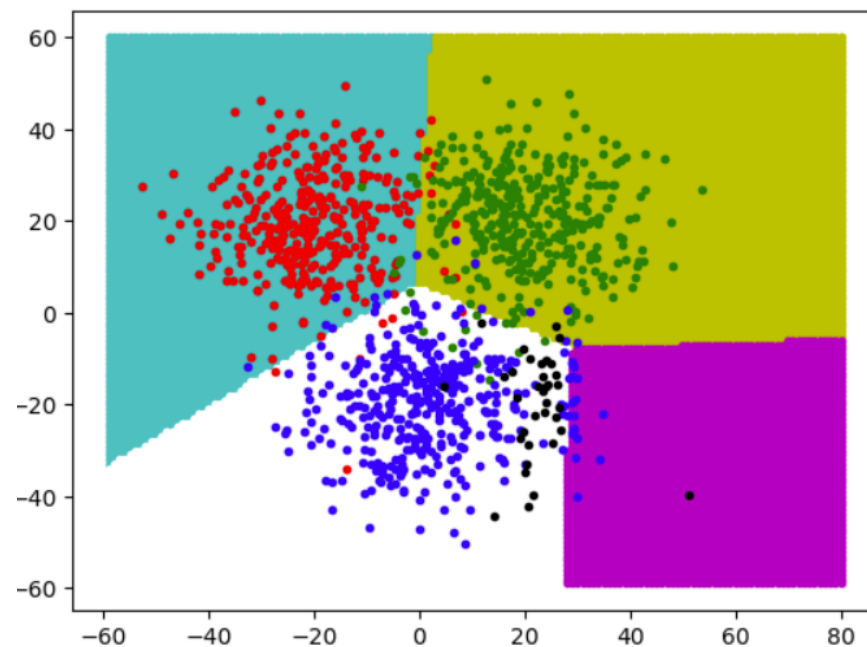
After 'log transformation'



Data Imbalance



One of the classes (class 3) has inadequate representation



Most instances are wrongly classified

Classification Report:

	precision	recall	f1-score	support
1	0.96	0.96	0.96	360
2	0.95	0.95	0.95	360
3	0.88	0.92	0.90	383
4	0.06	0.03	0.04	35

Confusion Matrix:

```
[[345  9  6  0]
 [ 7 343 10  0]
 [ 8  7 351 17]
 [ 0  2  32  1]]
```

Dealing with Data Imbalances

Data imbalance is a common problem in machine learning, where one class has a significantly higher number of observations than the other. This can lead to biased models and poor performance on the minority class. There are several techniques for dealing with data imbalance, including:

1. Use the right evaluation metrics: Accuracy is not always the best metric to evaluate model performance on imbalanced data. Instead, metrics such as precision, recall, F1-score, and AUC-ROC can provide a better understanding of model performance.
2. Resample the training set: Resampling techniques involve modifying the training set to balance the class distribution. This can be done by either **undersampling** the majority class or **oversampling** the minority class, **SMOTE** (Synthetic Minority Oversampling Technique), or a mix of these strategies.
3. Change the algorithm: Some algorithms are better suited for imbalanced data than others. For example, decision trees and random forests can handle imbalanced data well.
4. Use ensemble methods: Ensemble methods such as bagging, boosting, and stacking can be used to combine multiple models and improve the performance on imbalanced data.
5. Collect more data: Collecting more data can help to balance the class distribution and improve model performance.

Imbalanced Data -

- Oversampling of the lesser sample
- Undersampling of the dominant obs.
- Combination of the two -
- Synthetic (simulated) data.
- Data augmentation (images).

The Python package **imblearn (imbalanced-learn)** implements these strategies

<https://imbalanced-learn.org/stable/>

Data Imbalance

Review the following confusion matrix:

253	5
5	12

In this case, clearly there is data imbalance since 258 observations belong to Class-0, while only 17 observations belong to Class-1

Observe that even though only 12 out of 17 Class-1 observations were correctly classified (corresponding to = 70.58% accuracy figure for the class, the **overall 'accuracy'** of the classifier is:

$$(12 + 253) / (12 + 253 + 5 + 5) = 265 / 275 = \mathbf{96.36\%}$$

The metrics of the classifier are as follows:

- **Accuracy: 96.36%**
- Sensitivity (Recall): 70.59%
- Specificity: 98.06%
- Precision: 70.59%
- **F1-Score: 70.59%**

IMBALANCED DATA :

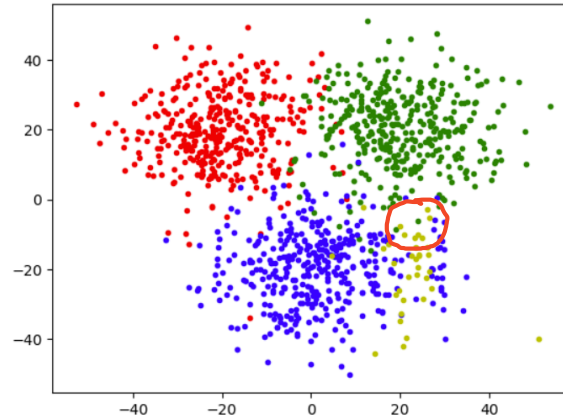
"The dominant class will hijack the model"

→ Accuracy values will be (falsely) high.

Handling Class Imbalances

- Over-sampling of minority class
- Under-sampling of majority class
- Combination

The scikit-learn module **imblearn** implements functions to tackle data-imbalance problems.



ENN

ENN is often used as a preprocessing step in combination with other techniques such as over-sampling (e.g., SMOTE) or under-sampling to address the imbalance issue while simultaneously improving the quality of the data.

Cleaning using ENN refers to the use of Edited Nearest Neighbors (ENN) as a method for cleaning or refining imbalanced datasets. ENN is a technique commonly employed in combination with over-sampling or under-sampling methods to improve the quality of the dataset, particularly in the context of classification tasks. Here's how cleaning using ENN typically works:

1. **Selecting a data point:** Iterate through each instance in the dataset.
2. **Identifying nearest neighbors:** Determine the k-nearest neighbors of the selected instance using a distance metric (e.g., Euclidean distance).
3. **Majority voting:** Check if the majority class label among the nearest neighbors differs from the label of the selected instance. If it does, consider the instance for removal.
4. **Removing inconsistent instances:** Remove the instances for which the majority class label among its nearest neighbors is different from its own label.

By removing instances that are misclassified by their nearest neighbors, ENN aims to improve the quality of the dataset by eliminating noisy or borderline instances. This can lead to a more robust and reliable training dataset, which in turn can improve the performance of classifiers, especially in scenarios where class imbalance is a concern.

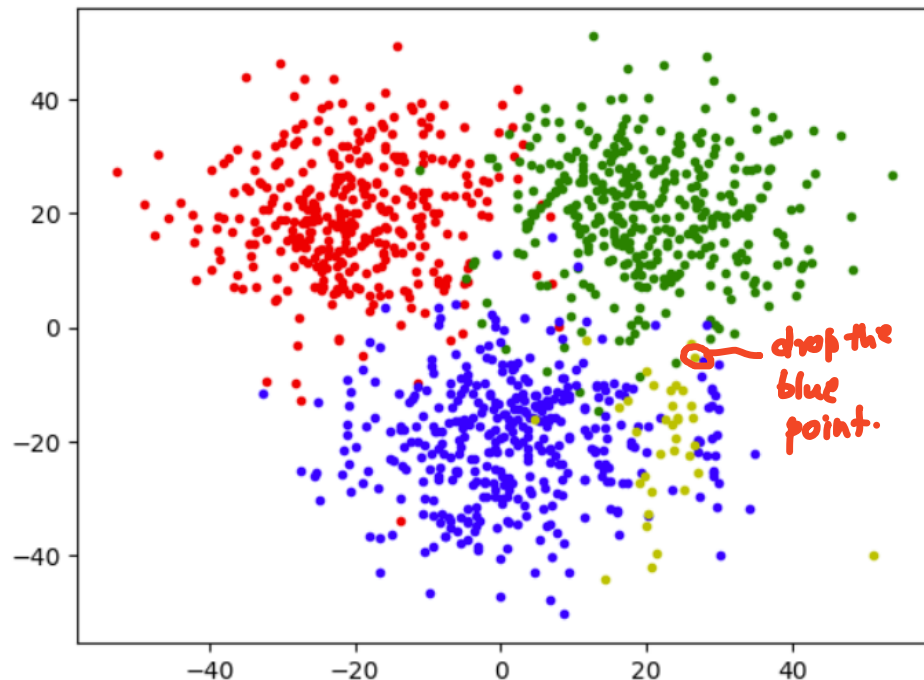
Tomek Links

Tomek links are pairs of instances in a dataset that belong to different classes and are close to each other, meaning they are nearest neighbors.

Specifically, a Tomek link exists between two instances if there are no other instances closer to each other than these two instances are to each other.

Tomek links are often used as a technique for cleaning or refining the dataset. The idea is to remove the majority class instance of each Tomek link, which can help improve the decision boundary of a classifier, especially in scenarios where there is significant overlap between classes or where the majority class instances dominate the decision-making process.

Removing the majority class instances that form Tomek links can help in addressing the issue of imbalanced classes by potentially reducing the dominance of the majority class and improving the overall performance of the classifier, particularly in terms of metrics like precision, recall, and F1-score.



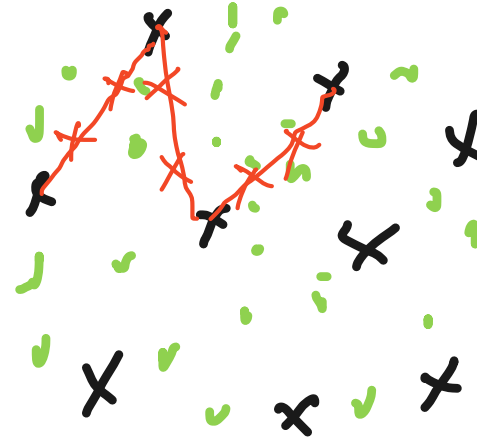
SMOTE

SMOTE stands for Synthetic Minority Over-sampling Technique. It's a popular method used in the context of handling imbalanced datasets, particularly in machine learning classification tasks.

The SMOTE method works by generating synthetic samples from the minority class to alleviate the class imbalance problem. It does this by interpolating between existing minority class instances. Here's how it typically works:

1. **Selecting a minority class instance:** Randomly choose an instance from the minority class.
2. **Finding its nearest neighbors:** Find the k-nearest neighbors for this instance (typically using Euclidean distance or other distance metrics).
3. **Creating synthetic samples:** Randomly select one of the nearest neighbors and use it to create a new synthetic instance. This synthetic instance is created by choosing a point along the line segment joining the selected minority class instance and its nearest neighbor.
4. **Repeat:** Repeat steps 1-3 until the desired balance between classes is achieved.

By generating synthetic samples, SMOTE effectively increases the representation of the minority class in the dataset without simply duplicating existing instances, thus helping to address the class imbalance problem. This can lead to better generalization and performance of classifiers trained on imbalanced datasets, especially when combined with other techniques like Tomek links or under-sampling of the majority class.

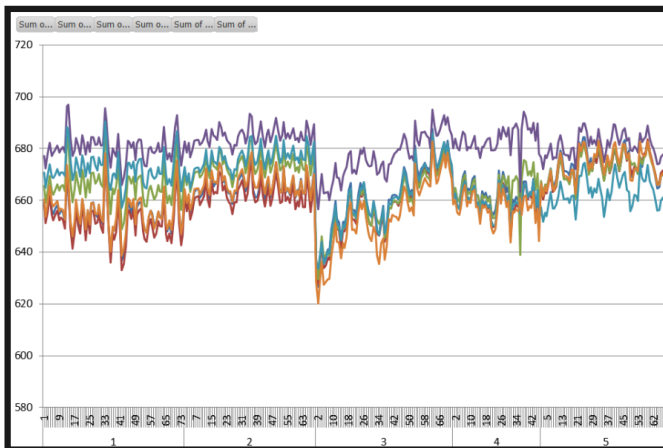
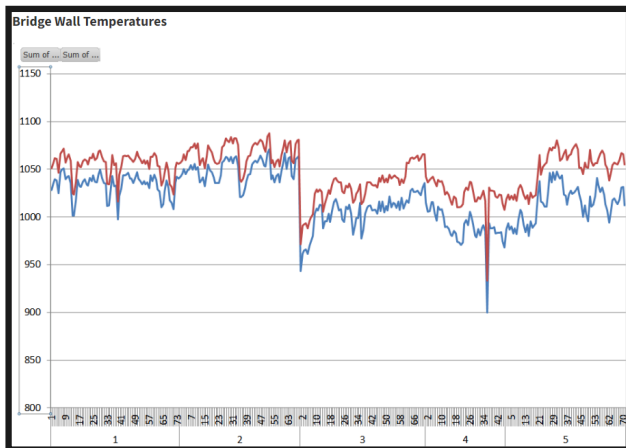
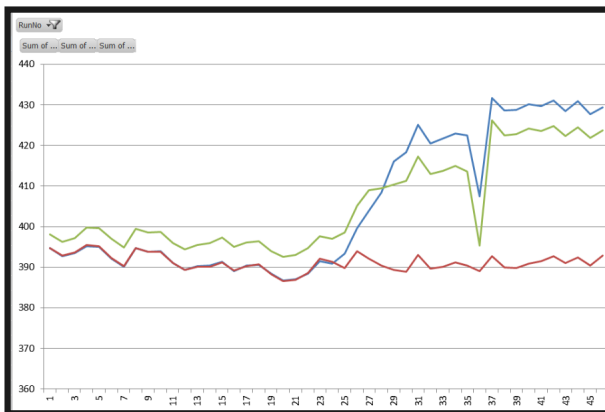
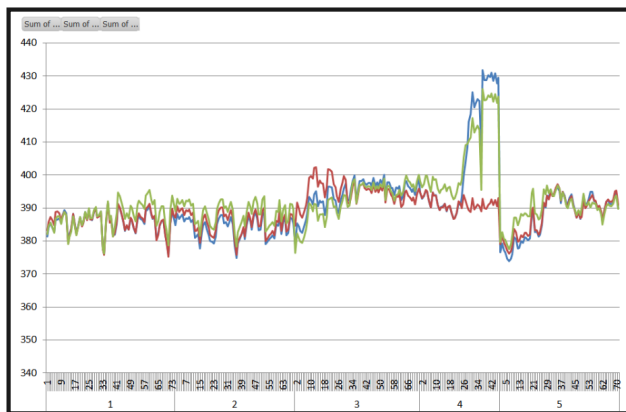


SMOTE & other methods
in 'imblearn'

EDA / Data Preparation : Features (ie. Columns)

- Number of features and their impact on models: Reduce the number of features to improve model quality
- Identification of interdependent features
 - Feature Correlation : Heat Maps
 - Multicollinearity detection : VIF - Variance Inflation
 - Feature encoding : Categorical Features
 - Feature Engineering
 - Feature Reduction
 - PCA : Principal Component Analysis
 - t-SNE : t-distributed Stochastic Neighbour Embedding

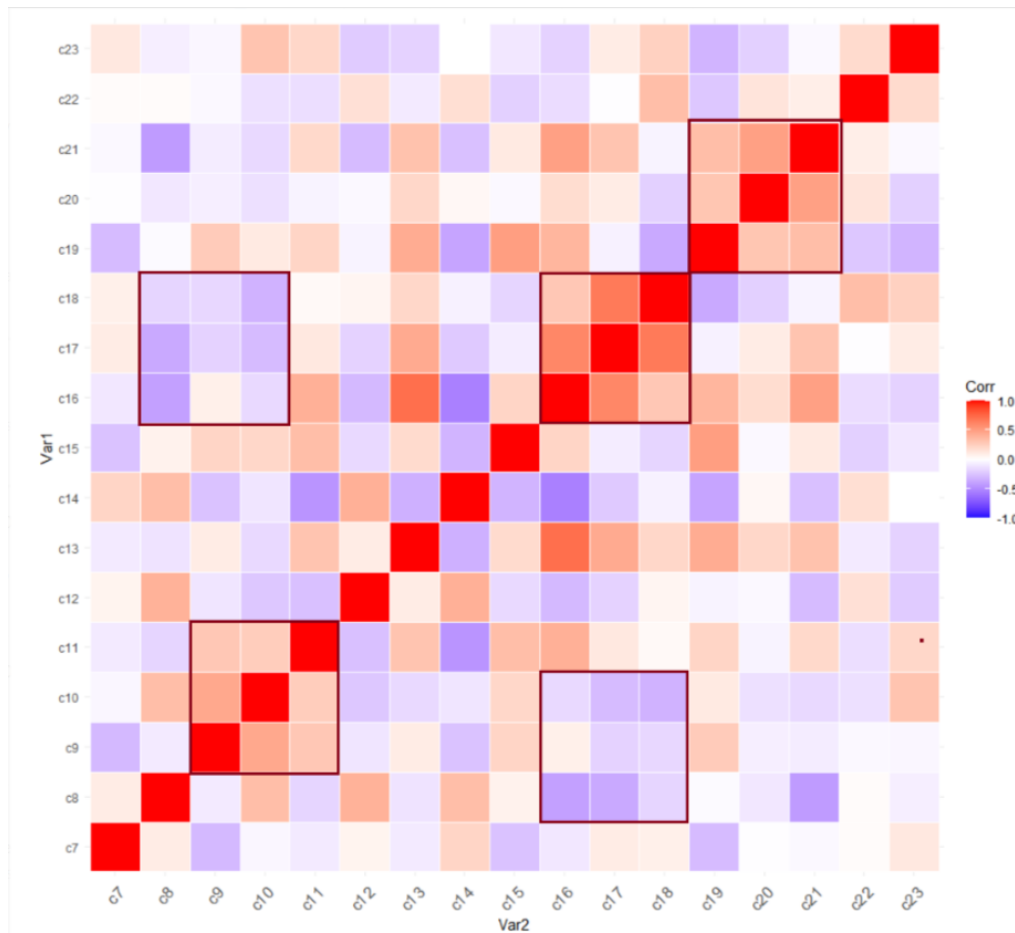
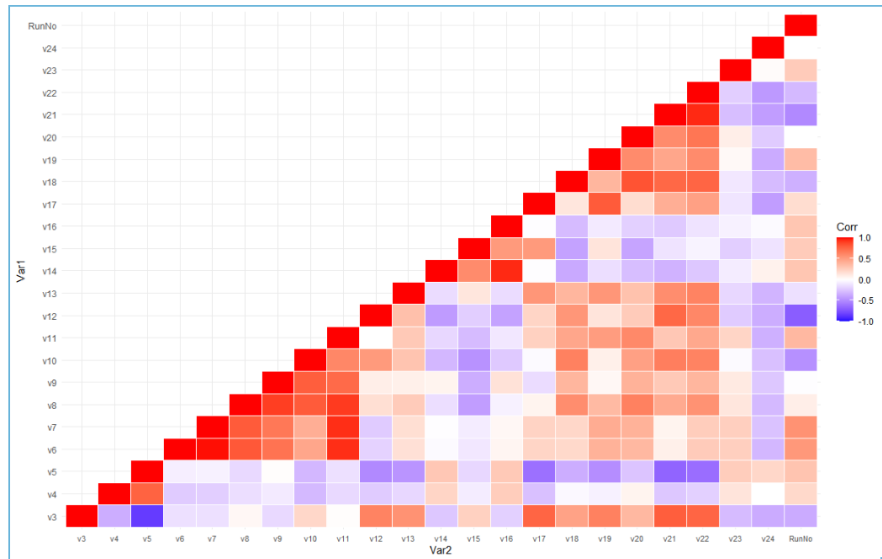
Pair-wise Correlation of Features



This method of pair-wise or group-wise plotting of features are convenient to detect feature dependencies if there are only a handful of features.

For a very large feature set, correlation **heat maps** are used. They can be automatically, and exhaustively created, and they provide a consolidated view of **feature correlations**.

Correlation Heat Maps



Multicollinearity Detection

Variance Inflation Factor (**VIF**)

The Variance Inflation Factor (VIF) quantifies how much the variance of a regression coefficient is inflated due to multicollinearity in the model. Mathematically, the VIF for a particular feature is calculated by performing a linear regression of that feature on all the other features.

For a given feature X_i , the VIF is defined as:

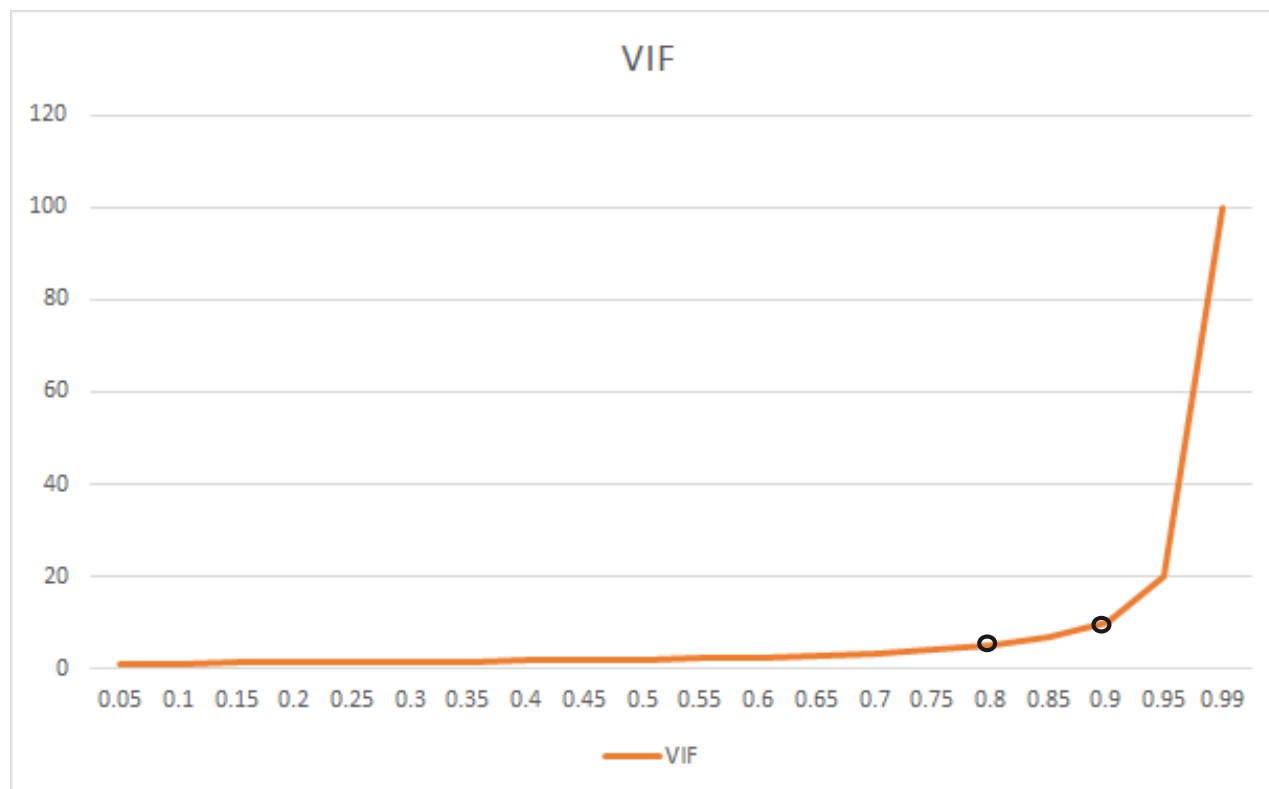
$$\text{VIF}(X_i) = \frac{1}{1-R_i^2}$$

Where:

- R_i^2 is the R^2 value obtained by regressing the feature X_i against all other features.

1. For each feature X_i in your dataset, run a linear regression using X_i as the dependent variable and all other features as independent variables.
2. Obtain the R^2 value from this regression, denoted as R_i^2 .
3. Calculate the VIF for X_i using the formula above.

A VIF of 1 indicates that there's no multicollinearity, whereas a VIF above 1 suggests that the feature is correlated with other features. A common rule of thumb is that a VIF above 10 indicates high multicollinearity, but this threshold can vary depending on the context.



VIF v/s R-square

R-square	VIF
0.05	1.052631579
0.1	1.111111111
0.15	1.176470588
0.2	1.25
0.25	1.333333333
0.3	1.428571429
0.35	1.538461538
0.4	1.666666667
0.45	1.818181818
0.5	2
0.55	2.222222222
0.6	2.5
0.65	2.857142857
0.7	3.333333333
0.75	4
0.8	5
0.85	6.666666667
0.9	10
0.95	20
0.99	100

	A	B	C	D	E	F	G	H	I
1	c1	c2	c26	c27	c28	c29	c30	c31	c32
2	43344		2 493.7968	104.5539	41.1876	290.9653	14.37955	71.73199	48.67901
3	43345		2 493.6619	104.5132	41.58075	290.6212	14.31532	78.59982	48.05742
4	43346		2 495.6449	104.5025	40.74457	292.1524	14.56618	78.83246	47.32059
5	43347		2 494.354	104.4529	40.28818	292.6762	14.60518	72.73663	47.98046
6	43348		2 492.0514	104.4886	41.26669	289.0175	14.54893	76.62107	48.2173
7	43349		2 491.2304	104.5097	42.63592	286.8439	14.31609	76.83934	48.43282
8	43350		2 493.0255	104.4522	41.96914	289.3589	14.47512	77.22903	48.33145
9	43351		2 492.9835	104.479	42.39755	290.8424	14.18896	73.14102	47.37012
10	43352		2 492.5621	104.531	43.21701	290.1102	14.11881	72.42022	47.36809
11	43353		2 493.8725	104.5807	43.00113	291.9806	14.21165	73.24146	47.38106
12	43354		2 490.2188	104.604	42.60375	287.1388	14.25636	73.24146	48.72817
13	43355		2 489.2321	104.472	42.89505	284.6288	14.26377	74.96662	49.60527
14	43356		2 491.5731	104.4412	44.05164	286.618	14.26902	76.23442	50.06172
15	43357		2 492.4099	104.4934	44.50718	287.3763	14.27284	77.44611	50.30747
16	43358		2 490.4405	104.4737	43.78121	285.1875	14.26373	80.78736	50.59053
17	43359		2 490.9521	104.5132	43.82019	285.7735	14.26143	79.81733	50.24661
18	43360		2 491.9521	104.5001	43.59271	286.4335	14.27795	81.64099	50.50682
19	43361		2 492.0228	104.5208	43.05299	287.2706	14.36223	81.6224	50.40502
20	43362		2 491.425	104.5444	43.69969	285.4068	14.16046	81.08435	50.34545
21	43363		2 491.4424	104.4848	43.57138	285.7367	14.1837	80.95044	50.32148
22	43364		2 492.6671	104.4642	42.89322	287.0949	14.25549	82.38933	50.27493

**How to 'drop' features
based on VIF analysis?**

Features	VIF	Features	VIF
const	1.031338e+07	c27	1.060532
c26	1.328880e+01	c30	4.330190
c28	3.225659e+01	c39	4.803262
c29	1.157359e+01	c157	6.593328
c31	1.785831e+01	c158	4.129364
c32	8.938850e+01	c160	1.843026
c33	9.569349e+01	c161	3.467590
c139	1.186148e+02	c162	3.160080
c142	3.273434e+01	c163	1.878506
c143	2.579679e+01	c7	2.774363
c155	1.036910e+01	c8	5.818705
c16	1.055594e+01	c9	6.480519
c19	1.122551e+02	c10	6.367653
		c11	2.923712
		c12	3.019417
		c13	6.077563
		c15	4.695586
		c17	3.781340
		c20	4.543217
		c21	3.737963
		c22	2.498634
		c23	7.632798
		c34	2.127750
		c35	3.572582
		c36	1.075612

The following strategy can be used to remove features:

1. Identify the features with high VIF values: The first step is to identify the features with high VIF values, typically above 5 or 10, which indicate that the feature is highly correlated with other features in the model.
2. Evaluate the importance of the features: The next step is to evaluate the importance of the features based on their relevance to the research question, their interpretability, and their contribution to the model's performance.
3. Remove the least important features: Based on the evaluation, the least important features can be removed from the model to reduce multicollinearity and improve the model's performance.
4. Re-evaluate the model: After removing the features, it is important to re-evaluate the model's performance using appropriate metrics and cross-validation techniques to ensure that the model is still accurate and reliable.

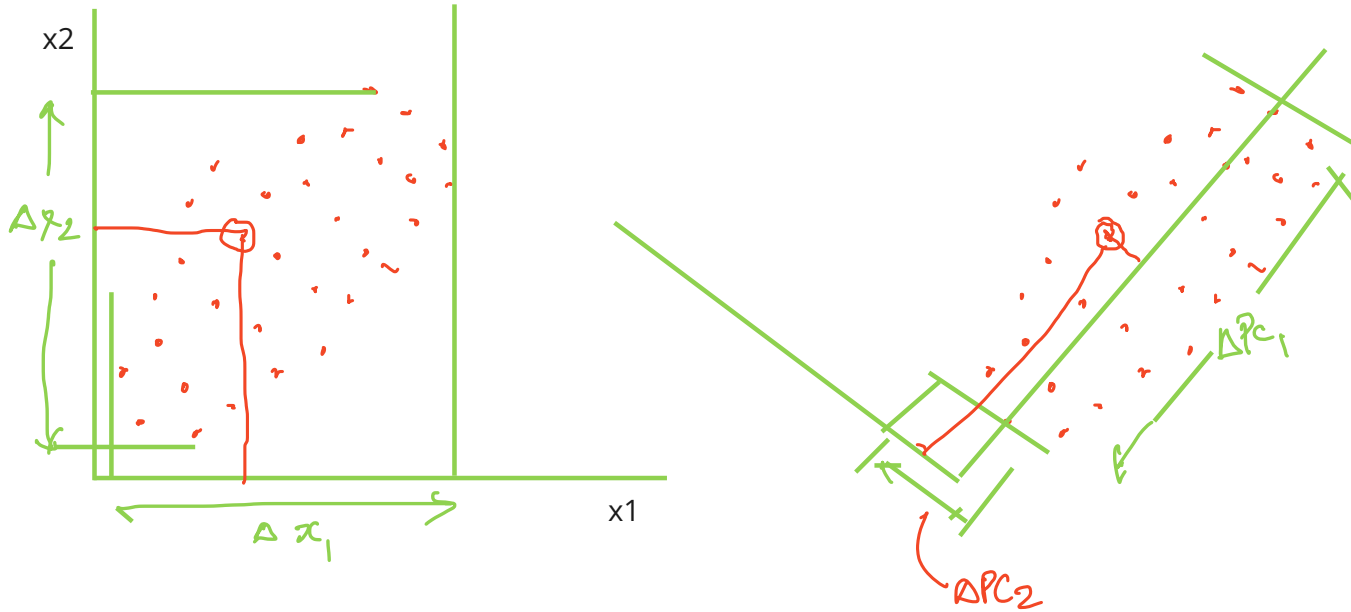
How to 'drop' features based on VIF analysis?

CAVEAT

It is important to note that **removing features can result in loss of information and may affect the interpretability of the model.**

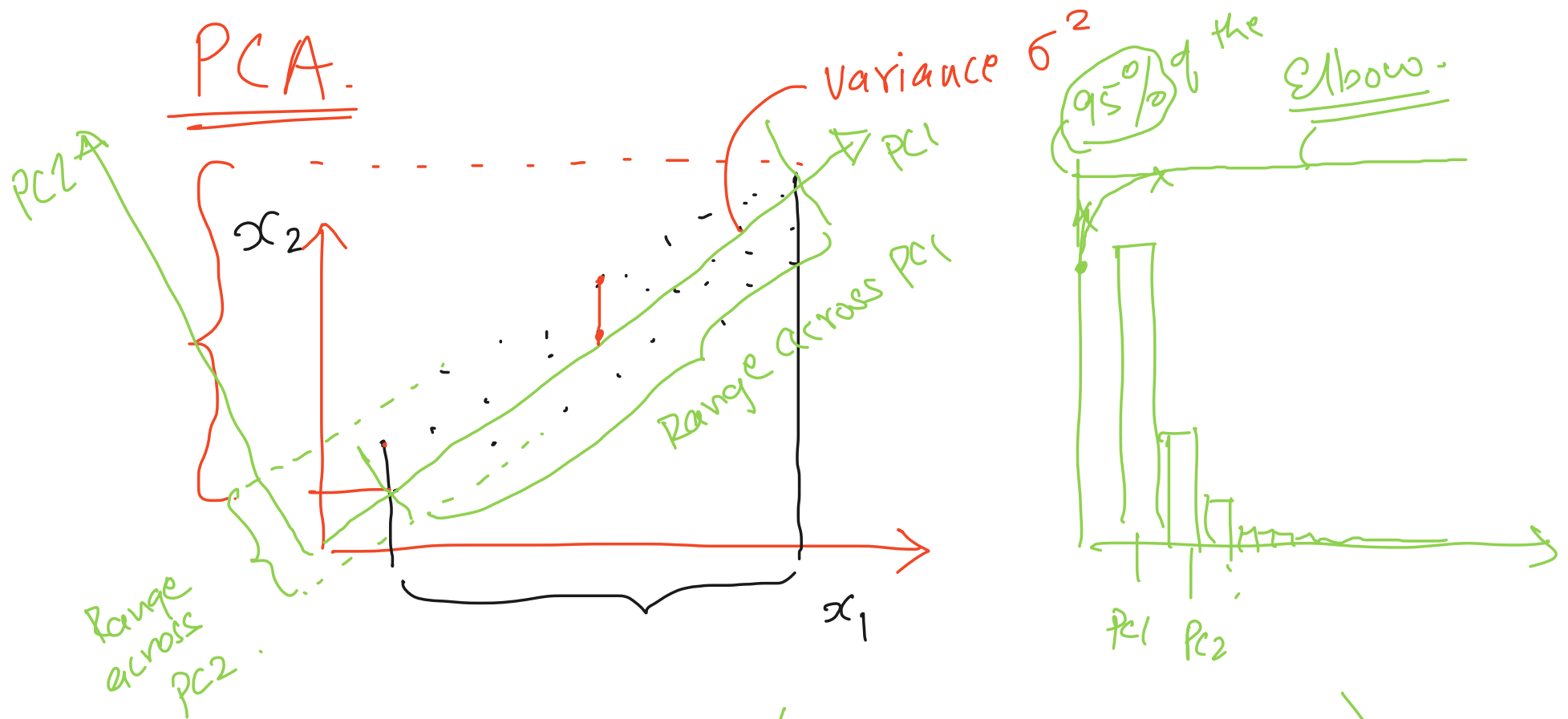
Therefore, it is important to carefully consider the trade-offs between multicollinearity and model performance and to choose the appropriate strategy based on the research question, the data, and the assumptions of the model

PCA - Principal Component Analysis



Principal Components are components along re-aligned axes such that PC1 'captures' or 'explains' or 'accounts for' the maximum variance, followed by PC2, PC3, and so on.

If the first few PCs cumulatively explain more than, say, 90% of the data variance, they may be sufficient for creating effective ML models. PCA thus serves the purpose of 'feature reduction'.



$$Y = f'(PC_1, PC_2, PC_3, \dots, PC_k)$$

Downside \rightarrow "NO EXPLAINABILITY"

PC = linear combination of X_i

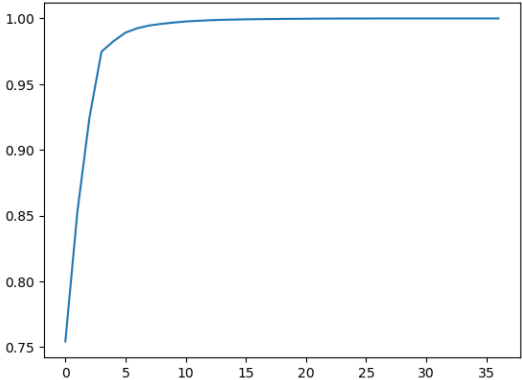
$$\underline{PC_1} = a_1 x_1 + b_1 x_2 + c_1 x_3 + d_1 x_4 + \dots$$

This data has more than 40 columns (features)

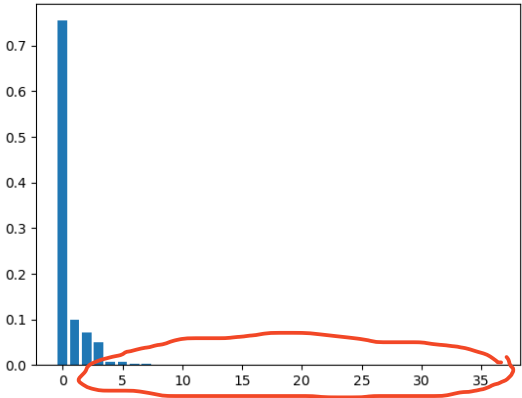
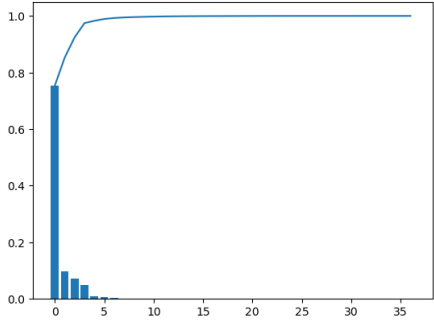
	A	B	C	D	E	F	G	H	I
1	c1	c2	c26	c27	c28	c29	c30	c31	c32
2	43344	2	493.7968	104.5539	41.1876	290.9653	14.37955	71.73199	48.67901
3	43345	2	493.6619	104.5132	41.58075	290.6212	14.31532	78.59982	48.05742
4	43346	2	495.6449	104.5025	40.74457	292.1524	14.56618	78.83246	47.32059
5	43347	2	494.354	104.4529	40.28818	292.6762	14.60518	72.73663	47.98046
6	43348	2	492.0514	104.4886	41.26669	289.0175	14.54893	76.62107	48.2173
7	43349	2	491.2304	104.5097	42.63592	286.8439	14.31609	76.83934	48.43282
8	43350	2	493.0255	104.4522	41.96914	289.3589	14.47512	77.22903	48.33145
9	43351	2	492.9835	104.479	42.39755	290.8424	14.18896	73.14102	47.37012
10	43352	2	492.5621	104.531	43.21701	290.1102	14.11881	72.42022	47.36809
11	43353	2	493.8725	104.5807	43.00113	291.9806	14.21165	73.24146	47.38106
12	43354	2	490.2188	104.604	42.60375	287.1388	14.25636	73.24146	48.72817
13	43355	2	489.2321	104.472	42.89505	284.6288	14.26377	74.96662	49.60527
14	43356	2	491.5731	104.4412	44.05164	286.618	14.26902	76.23442	50.06172
15	43357	2	492.4099	104.4934	44.50718	287.3763	14.27284	77.44611	50.30747
16	43358	2	490.4405	104.4737	43.78121	285.1875	14.26373	80.78736	50.59053
17	43359	2	490.9521	104.5132	43.82019	285.7735	14.26143	79.81733	50.24661
18	43360	2	491.9521	104.5001	43.59271	286.4335	14.27795	81.64099	50.50682
19	43361	2	492.0228	104.5208	43.05299	287.2706	14.36223	81.6224	50.40502
20	43362	2	491.425	104.5444	43.69969	285.4068	14.16046	81.08435	50.34545
21	43363	2	491.4424	104.4848	43.57138	285.7367	14.1837	80.95044	50.32148
22	43364	2	492.6671	104.4642	42.89322	287.0949	14.25549	82.38933	50.27493

[7.54261365e-01 9.87869904e-02 7.12964455e-02 5.03510215e-02
7.98749378e-03 6.55441064e-03 3.36504308e-03 2.09332441e-03
1.18752903e-03 1.01662107e-03 7.74969179e-04 5.43124838e-04
4.13982528e-04 3.22268906e-04 1.99904846e-04 1.82358764e-04
1.22357384e-04 9.84701179e-05 8.87915581e-05 8.02456161e-05
7.55522105e-05 6.24074825e-05 5.04485632e-05 3.08481492e-05
1.86827712e-05 1.49416419e-05 1.33783774e-05 2.66135642e-06
2.02403308e-06 9.51393151e-07 7.49623526e-07 2.39654410e-07
1.99355029e-07 8.26718788e-08 7.06173620e-08 4.41103905e-08
2.33592435e-11]

PCA reveals that only about 6 PCs explain more than 90% variance in the data, and only the first 6 PCs need be used to create ML models



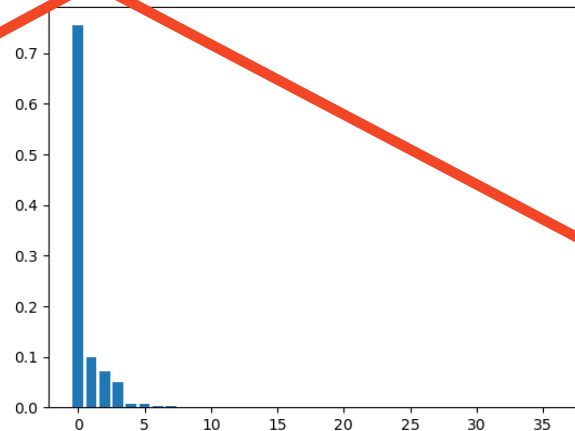
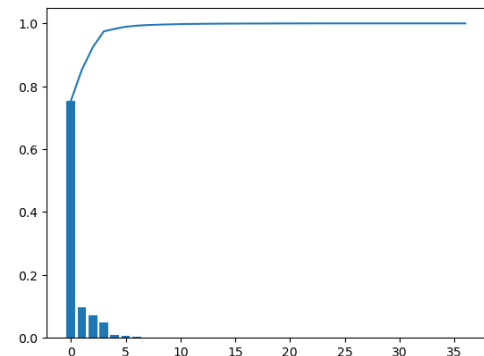
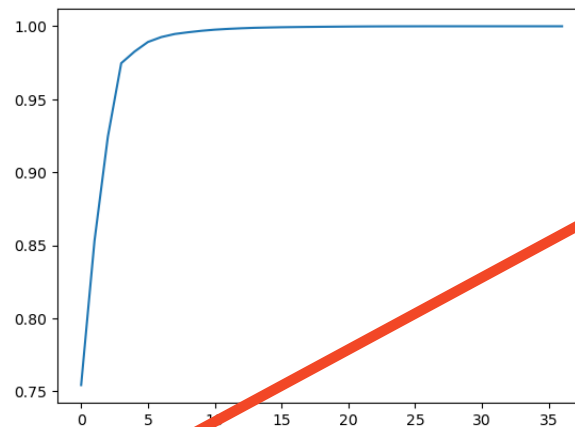
However, there is a problem with these results ... see next slide.



	A	B	C	D	E	F	G	H	I
1	c1	c2	c26	c27	c28	c29	c30	c31	c32
2	43344	2	493.7968	104.5539	41.1876	290.9653	14.37955	71.73199	48.67901
3	43345	2	493.6619	104.5132	41.58075	290.6212	14.31532	78.59982	48.05742
4	43346	2	495.6449	104.5025	40.74457	292.1524	14.56618	78.83246	47.32059
5	43347	2	494.5111	104.4529	40.28818	292.6762	14.60518	72.73663	47.98046
6	43348	2	492.0514	104.4886	41.26669	289.0175	14.54893	76.62107	48.2173
7	43349	2	491.2304	104.5091	42.63592	286.8439	14.31609	76.83934	48.43282
8	43350	2	493.0255	104.4522	41.55914	289.3589	14.47512	77.22903	48.33145
9	43351	2	492.9835	104.479	42.39755	290.8424	14.18896	73.14102	47.37012
10	43352	2	492.5621	104.531	43.21701	290.1102	14.11881	72.42022	47.36809
11	43353	2	493.8725	104.5807	43.00113	291.9806	14.21165	73.24146	47.38106
12	43354	2	490.2188	104.604	42.60375	287.1388	14.25806	73.24146	48.72817
13	43355	2	489.2321	104.472	42.89505	284.6288	14.26377	74.06662	49.60527
14	43356	2	491.5731	104.4412	44.05164	286.618	14.26902	76.23442	50.06172
15	43357	2	492.4099	104.4934	44.50718	287.3763	14.27284	77.44611	50.50747
16	43358	2	490.4405	104.4737	43.78121	285.1875	14.26373	80.78736	50.59053
17	43359	2	490.9521	104.5132	43.82019	285.7735	14.26143	79.81733	50.24661
18	43360	2	491.9521	104.5001	43.59271	286.4335	14.27795	81.64099	50.50682
19	43361	2	492.0228	104.5208	43.05299	287.2706	14.36223	81.6224	50.40502
20	43362	2	491.425	104.5444	43.69969	285.4068	14.16046	81.08435	50.34545
21	43363	2	491.4424	104.4848	43.57138	285.7367	14.1837	80.95044	50.32148
22	43364	2	492.6671	104.4642	42.89322	287.0949	14.25549	82.38933	50.27033

[7.54261365e-01 9.87869904e-02 7.12964455e-02 5.03510215e-02
7.98749378e-03 6.55441064e-03 3.36504308e-03 2.09332441e-03
1.18752903e-03 1.01662107e-03 7.74569179e-04 5.43124838e-04
4.13982528e-04 3.22268906e-04 1.99904846e-04 1.82358764e-04
1.22357384e-04 9.84701170e-05 8.87915581e-05 8.02456161e-05
7.55522105e-05 6.24074825e-05 5.04485632e-05 3.08481492e-05
1.86827712e-05 1.49416419e-05 1.33783774e-05 2.66135642e-06
2.02403308e-06 9.51393151e-07 7.49623526e-07 2.39654410e-07
1.99355029e-07 8.26718788e-08 7.06173620e-08 4.41103905e-08
2.53592435e-11]

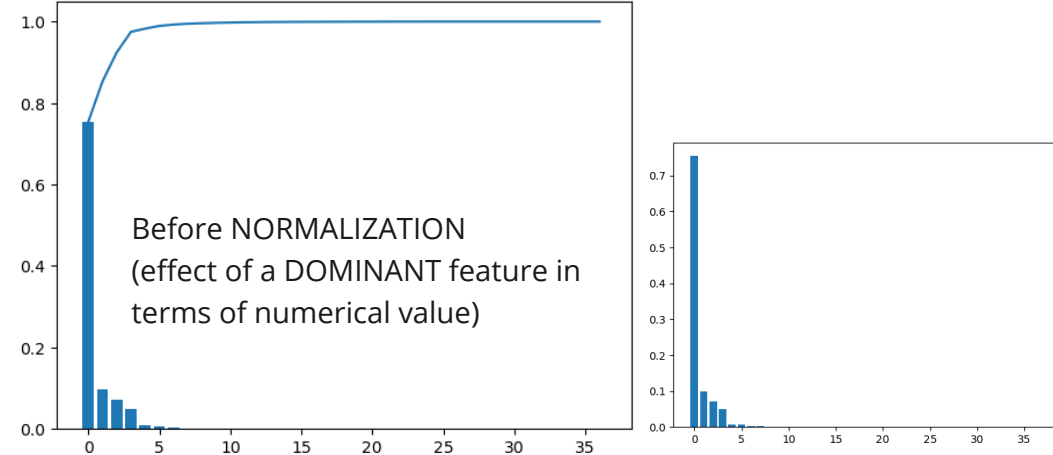
Problem: One feature seems to be very 'dominant'. This is happening because of incompatible scales of the features



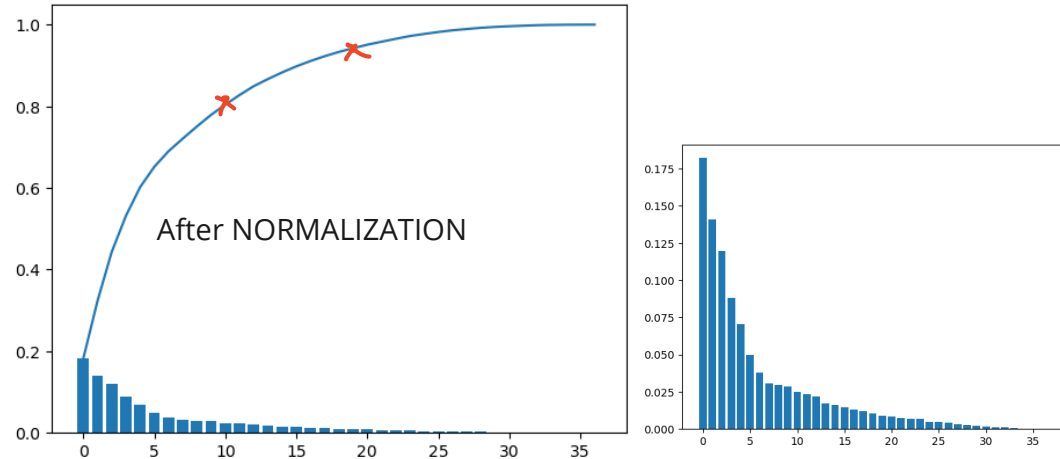
Importance of NORMALIZATION

Note : Data needs to be normalized before performing PCA

```
[7.54261365e-01 9.87869904e-02 7.12964455e-02 5.03510215e-02  
7.98749378e-03 6.55441064e-03 3.36504308e-03 2.09332441e-03  
1.18752903e-03 1.01662107e-03 7.74969179e-04 5.43124838e-04  
4.13982528e-04 3.22268906e-04 1.99904846e-04 1.82358764e-04  
1.22357384e-04 9.84701179e-05 8.87915581e-05 8.02456161e-05  
7.55522105e-05 6.24074825e-05 5.04485632e-05 3.08481492e-05  
1.86827712e-05 1.49416419e-05 1.33783774e-05 2.66135642e-06  
2.02403308e-06 9.51393151e-07 7.49623526e-07 2.39654410e-07  
1.99355029e-07 8.26718788e-08 7.06173620e-08 4.41103905e-08  
2.33592435e-11]
```



```
[1.82187813e-01 1.41144034e-01 1.19882002e-01 8.83869333e-02  
7.02808938e-02 4.99639610e-02 3.80999668e-02 3.08267651e-02  
2.98342213e-02 2.84815612e-02 2.49268628e-02 2.35483667e-02  
2.17418607e-02 1.73831972e-02 1.63501067e-02 1.47978112e-02  
1.30720181e-02 1.18307523e-02 1.05092250e-02 9.01490576e-03  
8.63737539e-03 7.21624942e-03 7.11038058e-03 6.89243449e-03  
5.03197986e-03 4.82978737e-03 4.14962998e-03 3.18816726e-03  
2.93711660e-03 2.14386597e-03 1.64215733e-03 1.35758835e-03  
1.28773381e-03 6.51744979e-04 3.97614261e-04 1.45522755e-04  
1.17392623e-04]
```



Uses of PCA

Data understanding:

- PCA can be used to identify the most important variables in a dataset and to understand the relationships between them.
- PCA can be used to detect outliers and anomalies in the data by identifying data points that are far from the rest of the data.

Data visualization:

- PCA can be used to visualize high-dimensional data in a low-dimensional space, typically 2D or 3D, by reducing the dimensionality of the data while preserving the most important information.
- PCA can be used to identify clusters and patterns in the data that are not visible in the high-dimensional space.

Data simplification:

- PCA can be used to simplify the data by reducing the number of variables or features in the dataset while preserving the most important information.
- PCA can be used to remove noise and redundancy from the data by identifying the most important components and ignoring the rest.

Dimensionality reduction:

- PCA can be used to reduce the dimensionality of the data by finding a new set of variables that are smaller than the original set but still contain most of the information in the original set.
- PCA can be used to transform a large set of variables into a smaller one that still contains most of the information in the large set.

Machine learning:

- PCA can be used as a preprocessing step in machine learning to reduce the dimensionality of the data and to improve the performance of the model.
- PCA can be used to remove multicollinearity and to improve the interpretability of the model by identifying the most important variables.