

Steps in DS.

① Understanding the problem

② } Exploratory data analysis

③ } Visualization.

- Source of the data?

- Formats

- text

- Binary

- Files

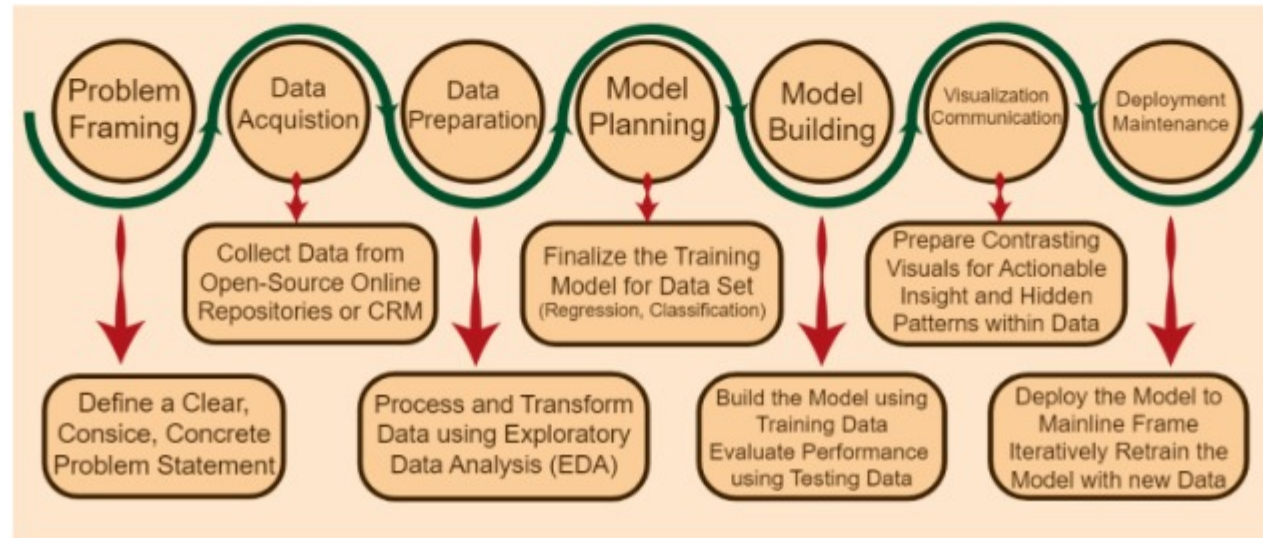
- Database

- 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.

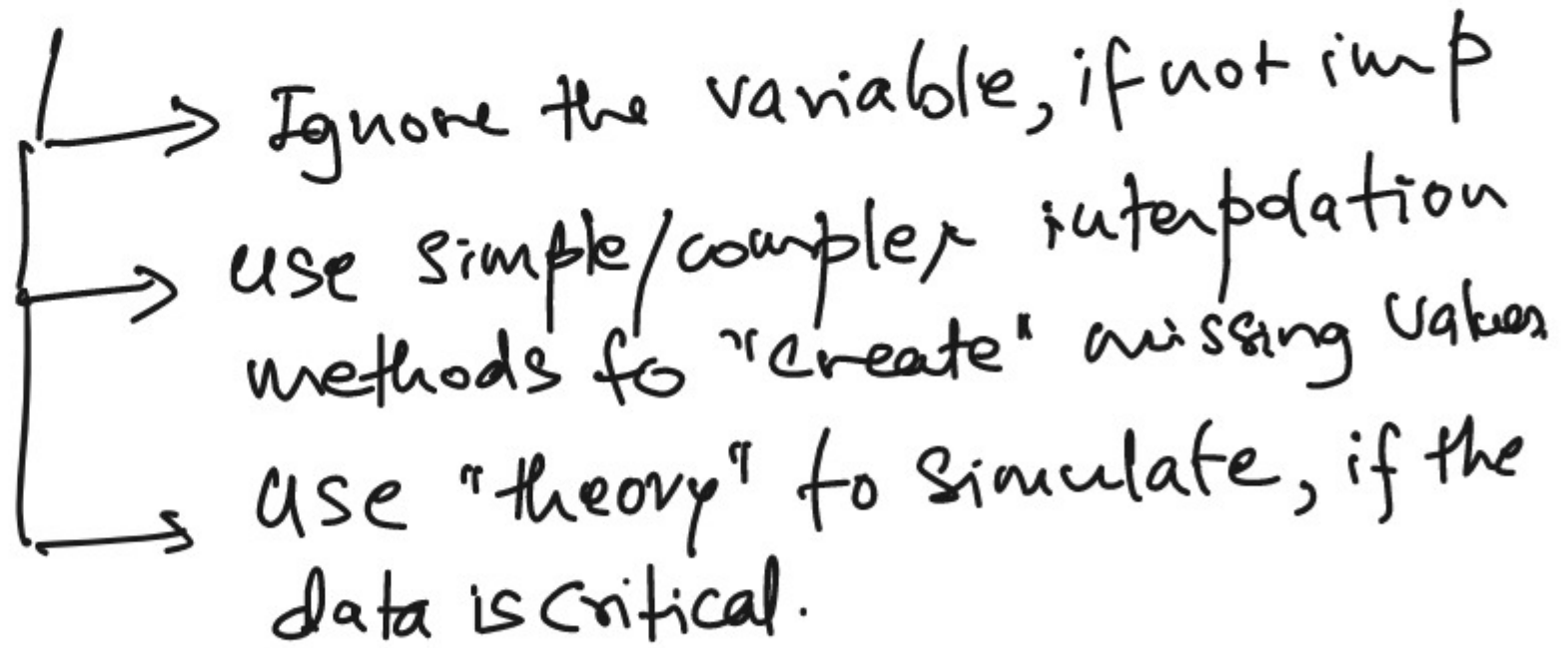
Some common data problems that can be revealed during EDA:

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.

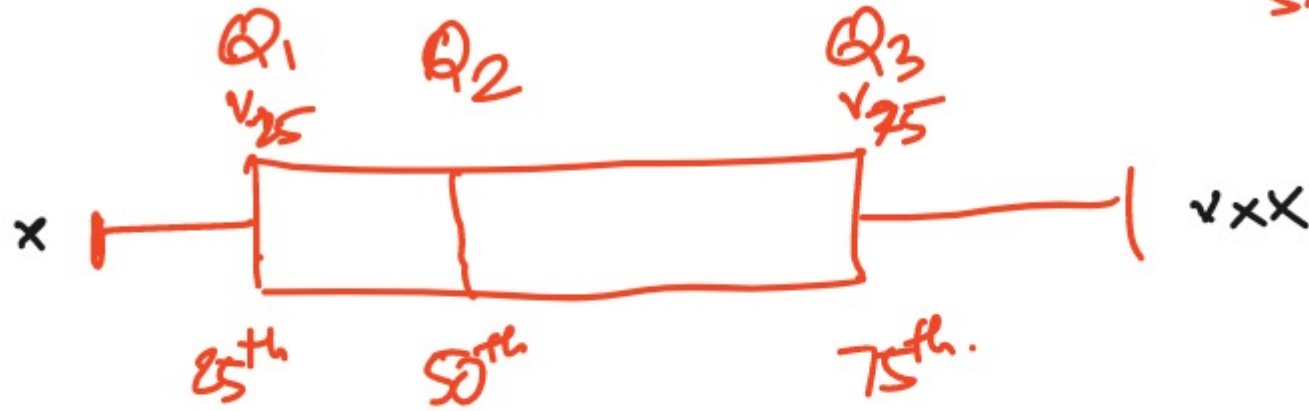
Problems in a data set.

- Entire missing observations... → Can I aggregate & create good data aggregation?
- Most variables have value in an observation but some don't



- Outliers: → detect & fix.

Outliers.



Create descriptive statistics.

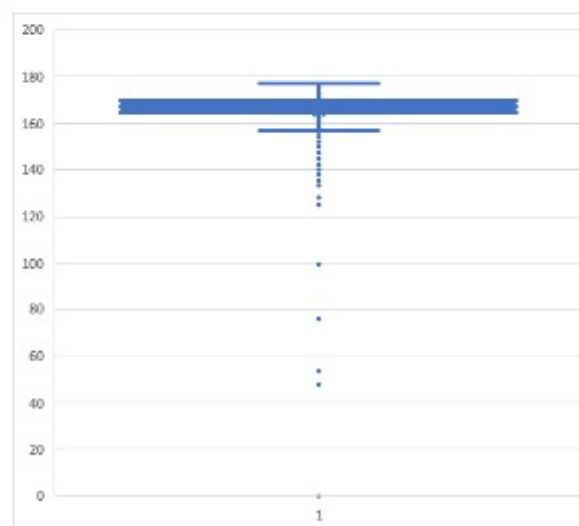
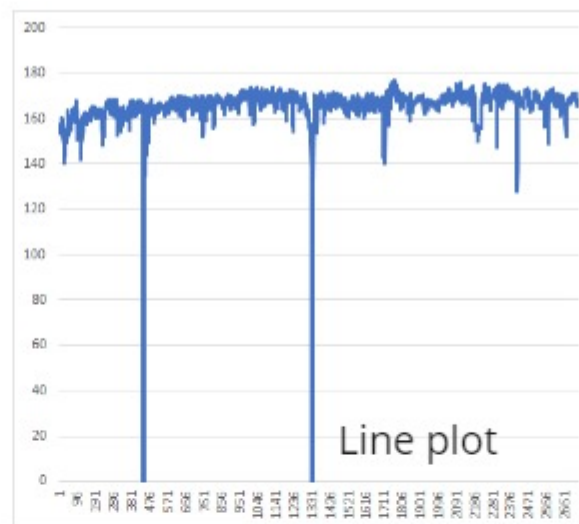
- Mean
- Median
- Mode
- Std. dev / var.
- Skewness
- Kurtosis -
- 25^{th} percentile
- 50^{th} " (Median)
- 75^{th} "

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

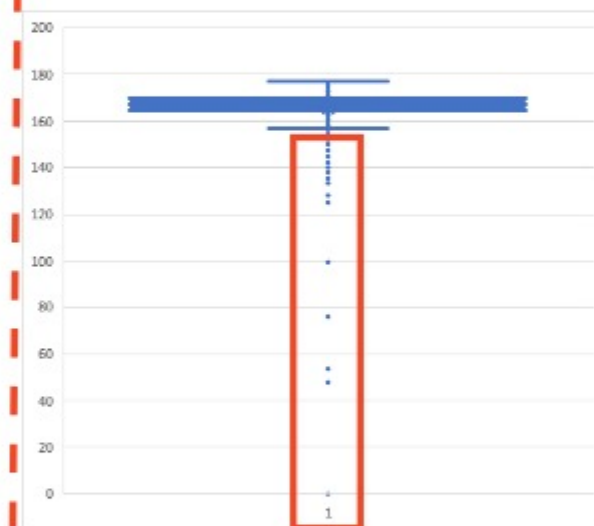
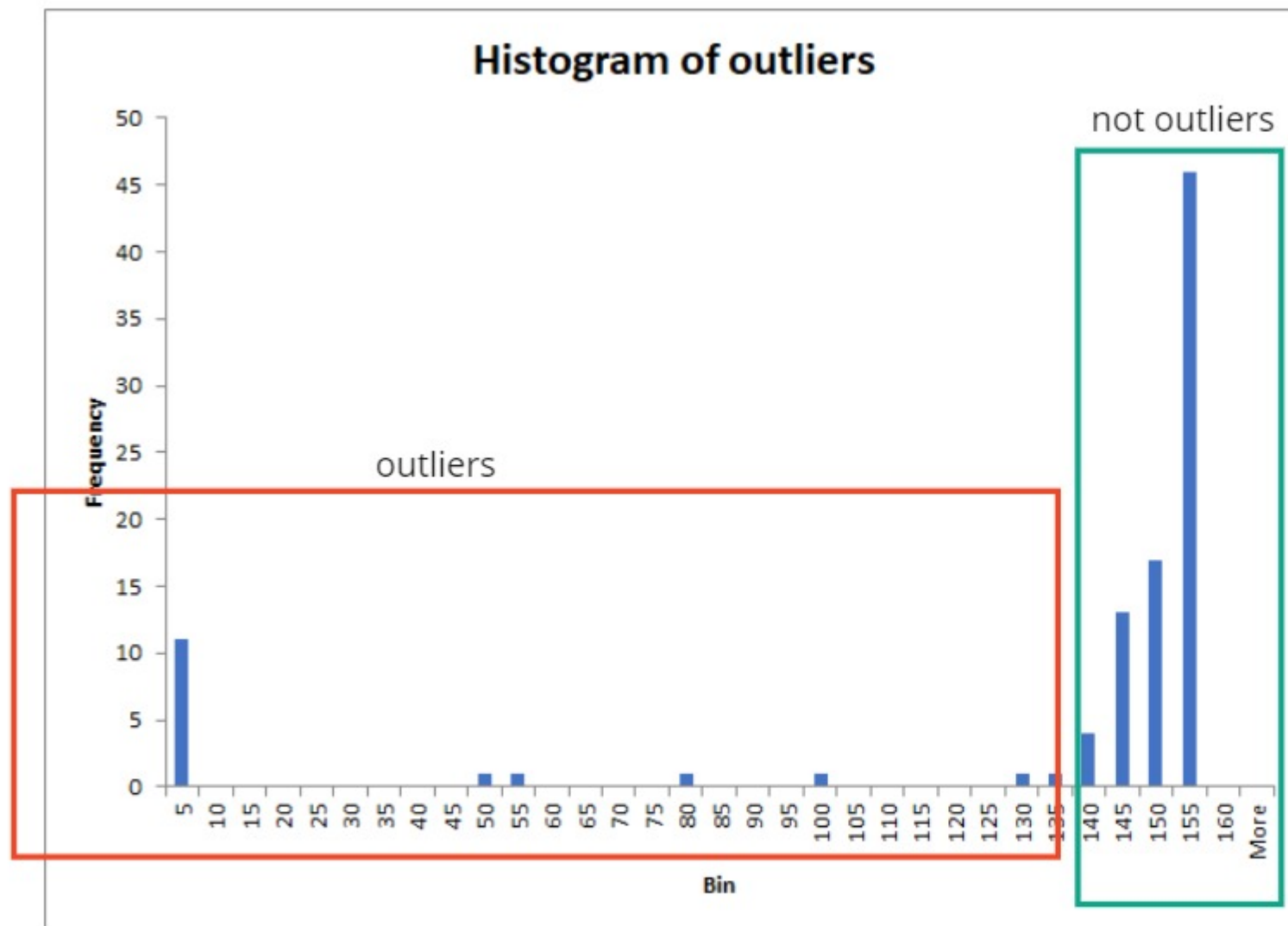
$$LB = Q_1 - 1.5 \times IQR$$

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

	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

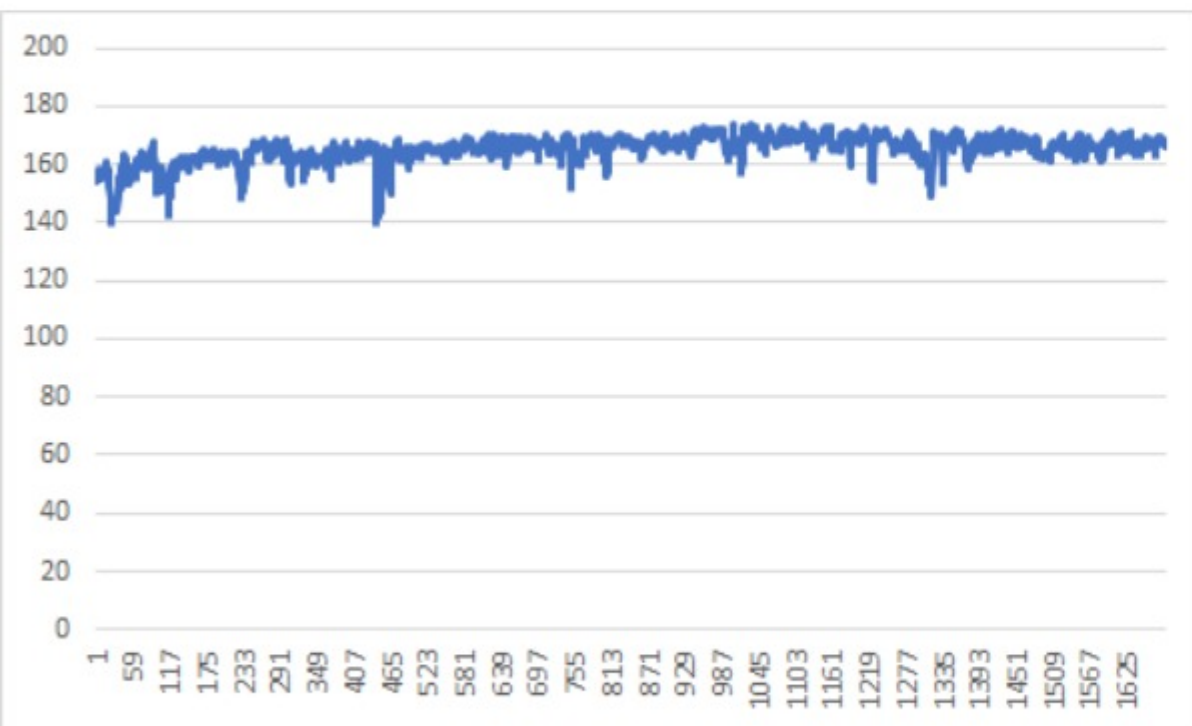


c4	
Mean	163.9284816
Standard Error	0.363181665
Median	166.1539939
Mode	0
Standard Deviation	14.9875705
Sample Variance	224.6272695
Kurtosis	94.71693965
Skewness	-9.197660397
Range	174.0257593
Minimum	0
Maximum	174.0257593
Sum	279170.2041
Count	1703
Quartiles	
Q1 (25th percentile)	163.2876459
Q2 (50th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

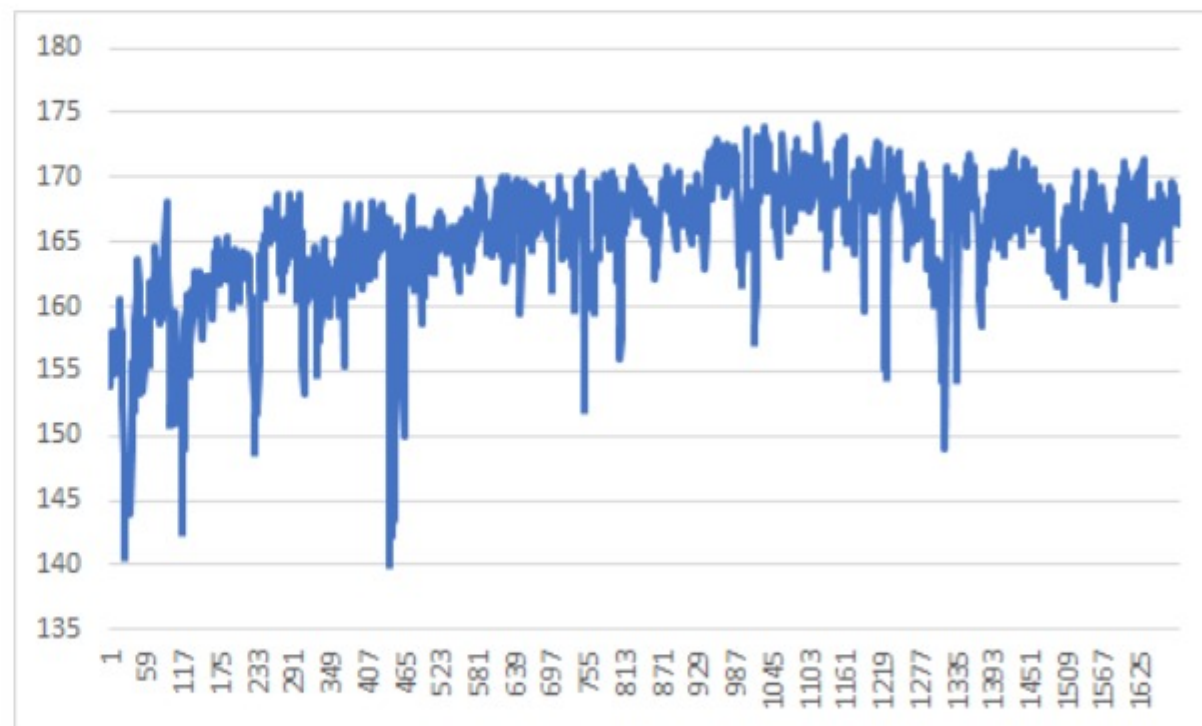


Outlier analysis

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

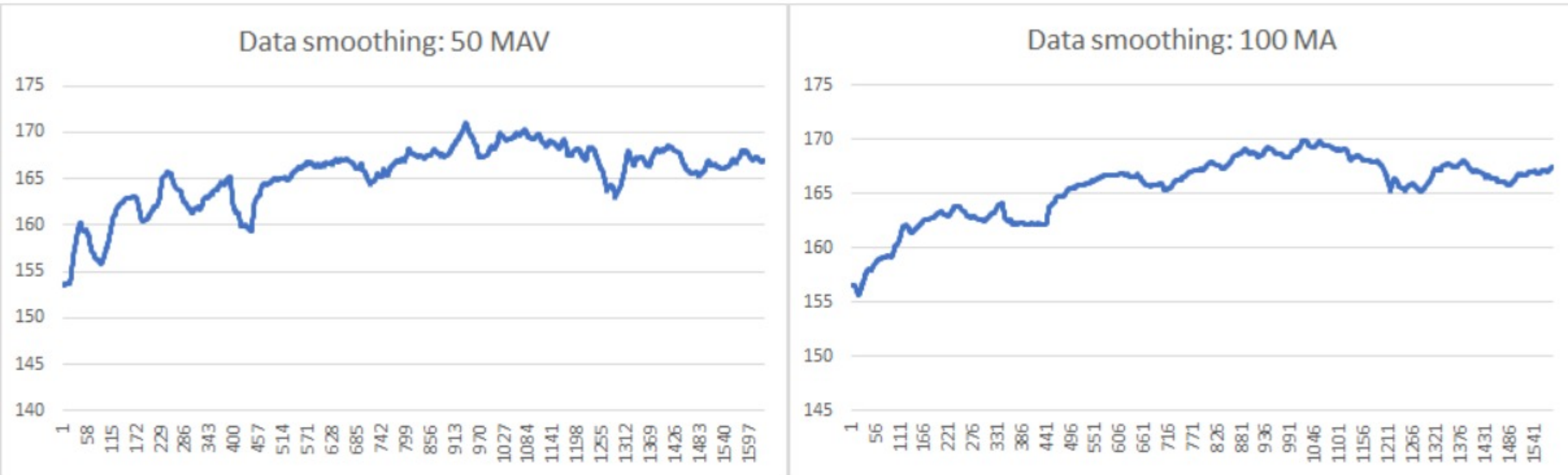


After dropping the outliers



After re-scaling the display

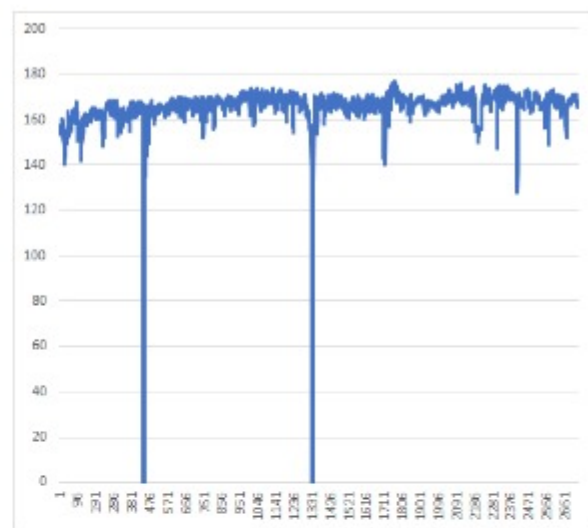
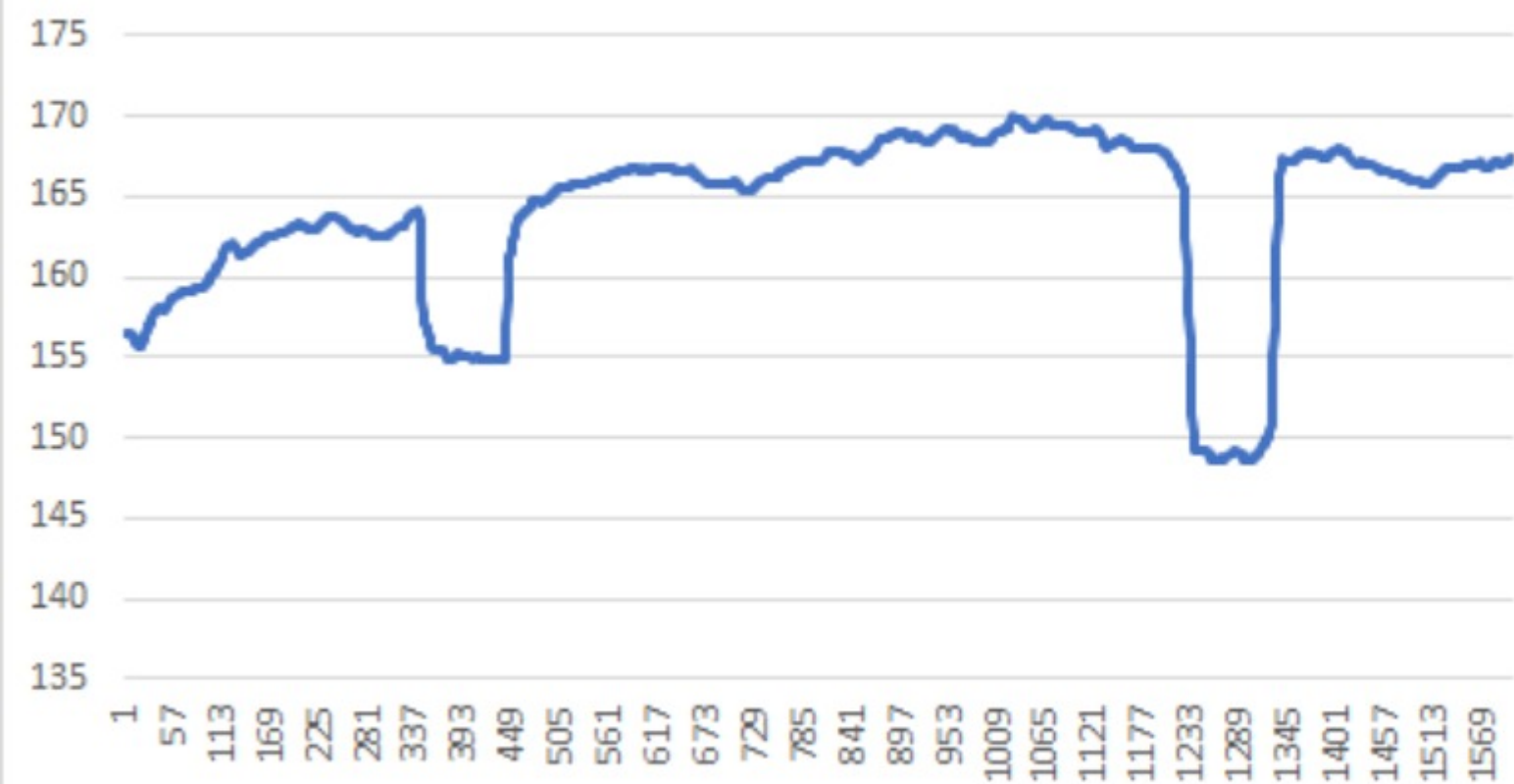
Moving Averages for Data Smoothing



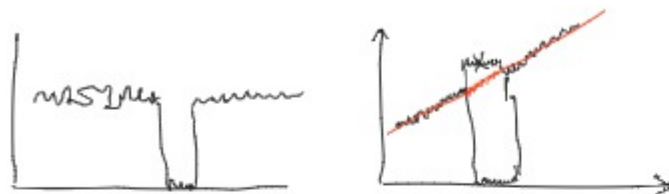
Moving averages can be used for:

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

Effect of outliers on MA of raw data



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

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.



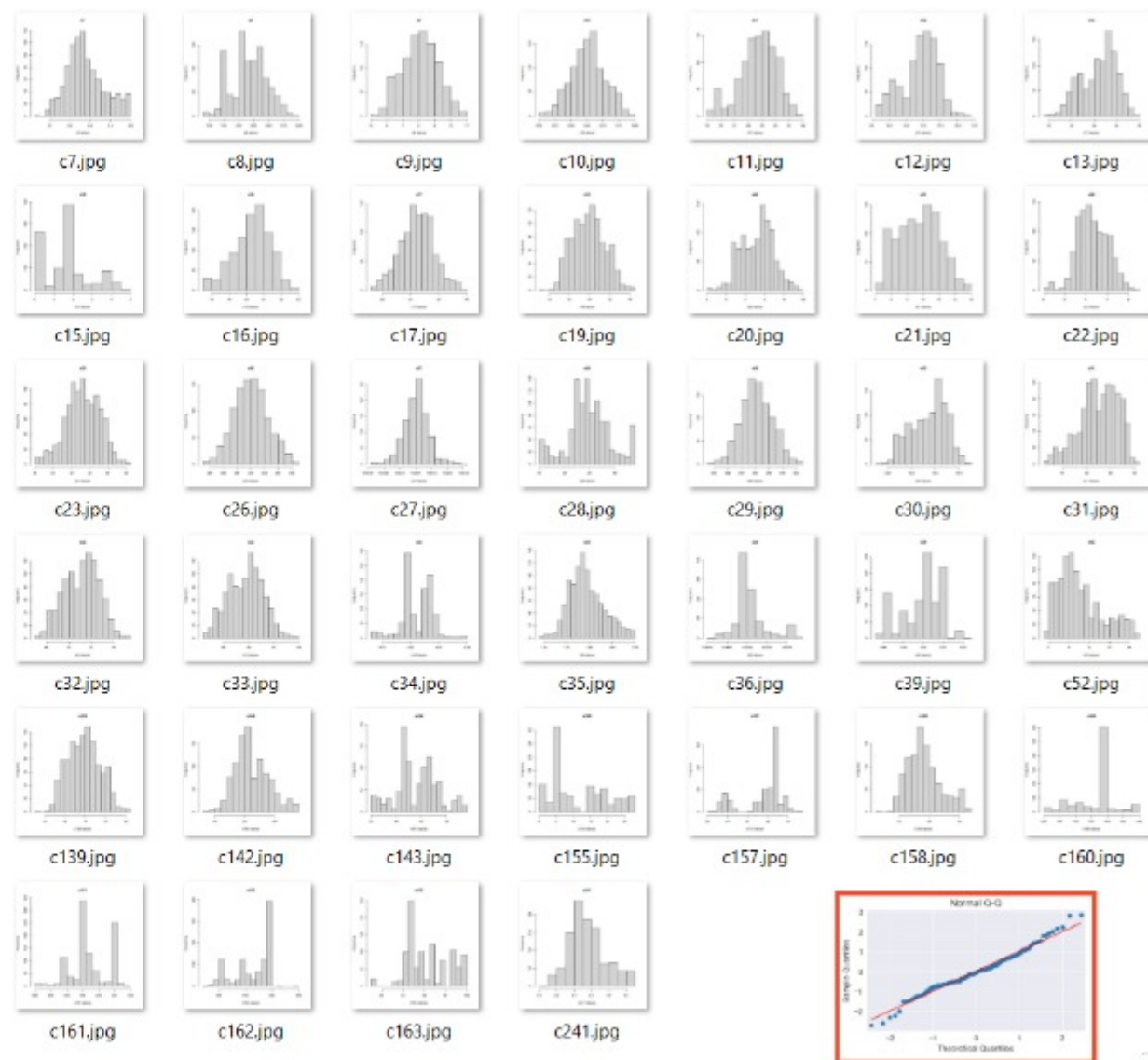
Column	Min	Max	Mean
c3	0	187.0776	170.3037
c4	0	174.0258	163.9285
c5	0	0.915783	0.529769
c6	0	3.144694	1.5678
c7	0.447406	2.971087	2.426871
c8	0	22.69527	21.20686
c9	0	10.27313	7.427598
c10	0	0.785255	0.642268
c11	0.002404	59.08272	56.35317
c12	0	13.7972	11.64599
c13	0	36.43647	32.86705
c14	0	5.409382	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

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**

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.

	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								

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	y	x1	x2	x3	x4		x1	x2	x3	x4		SUMMARY OUTPUT						
2	0.038117	0	26	0	0		0	0.577715	0	0								
3	0.896468	0.005556	24.00003	1.71E-07	9.53E-10		0.005587	0.288853	1.74E-07	9.74E-10		Regression Statistics						
4	0.159546	0.011111	22.00012	1.37E-06	1.52E-08		0.011173	0	1.39E-06	1.56E-08		Multiple R	0.894829					
5	0.863764	0.016667	28.00028	4.63E-06	7.72E-08		0.01676	0.866622	4.71E-06	7.89E-08		R Square	0.800719					
6	1.106349	0.022222	24.00049	1.1E-05	2.44E-07		0.022346	0.28892	1.12E-05	2.49E-07		Adjusted R Square	0.79619					
7	1.010169	0.027778	25.00077	2.14E-05	5.95E-07		0.027933	0.433394	2.18E-05	6.09E-07		Standard Error	0.341094					
8	0.278498	0.033333	23.00111	3.7E-05	1.23E-06		0.03352	0.144576	3.77E-05	1.26E-06		Observations	181					
9	1.114231	0.038889	23.00151	5.88E-05	2.29E-06		0.039106	0.144634	5.98E-05	2.34E-06								
10	1.029804	0.044444	28.00198	8.78E-05	3.9E-06		0.044693	0.866867	8.93E-05	3.99E-06		ANOVA						
11	0.37387	0.05	28.0025	0.000125	6.25E-06		0.050279	0.866943	0.000127	6.39E-06			df	SS	MS	F	Significance F	
12	0.971634	0.055556	28.00309	0.000171	9.53E-06		0.055866	0.867028	0.000174	9.74E-06		Regression	4	82.27607	20.56902	176.7934	1.61E-60	
13	0.975377	0.061111	24.00373	0.000228	1.39E-05		0.061453	0.289388	0.000232	1.43E-05		Residual	176	20.47671	0.116345			
14	1.079774	0.066667	22.00444	0.000296	1.98E-05		0.067039	0.000624	0.000301	2.02E-05		Total	180	102.7528				
15	1.24279	0.072222	23.00522	0.000377	2.72E-05		0.072626	0.145169	0.000383	2.78E-05								
16	0.644699	0.077778	25.00605	0.000471	3.66E-05		0.078212	0.434156	0.000478	3.74E-05			Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
17	0.656177	0.083333	24.00694	0.000579	4.82E-05		0.083799	0.289852	0.000588	4.93E-05		Intercept	0.790809	0.093229	8.482418	8.76E-15	0.606818	0.9748
18	1.095492	0.088889	27.0079	0.000702	6.24E-05		0.089385	0.72329	0.000714	6.38E-05		x1	3.244962	0.43132	7.523326	2.64E-12	2.393737	4.096187
19	1.115275	0.094444	28.00892	0.000842	7.96E-05		0.094972	0.86787	0.000857	8.14E-05		x2	0.036221	0.088316	0.410128	0.682211	-0.13807	0.210516
20	1.512548	0.1	26.01	0.001	0.0001		0.100559	0.57916	0.001017	0.000102		x3	-26.084	1.737131	-15.0156	2.43E-33	-29.5123	-22.6558
21	0.639396	0.105556	28.01114	0.001176	0.000124		0.106145	0.868191	0.001196	0.000127		x4	22.88368	1.466728	15.60186	5.09E-35	19.98904	25.77832
22	1.406627	0.111111	24.01235	0.001372	0.000152		0.111732	0.290632	0.001395	0.000156								
23	1.172479	0.116667	23.01361	0.001588	0.000185		0.117318	0.146381	0.001615	0.000189								

Gradient Descent-Based Algorithms: Algorithms like linear regression, logistic regression, and neural networks that use gradient descent optimization can be sensitive to feature scales. Different scales can lead to slow convergence or require careful tuning of learning rates.

There are several different data scaling methods used in data preprocessing to transform features into a common scale. Each method has its own advantages and is suited for different situations. Here are some common data scaling methods:

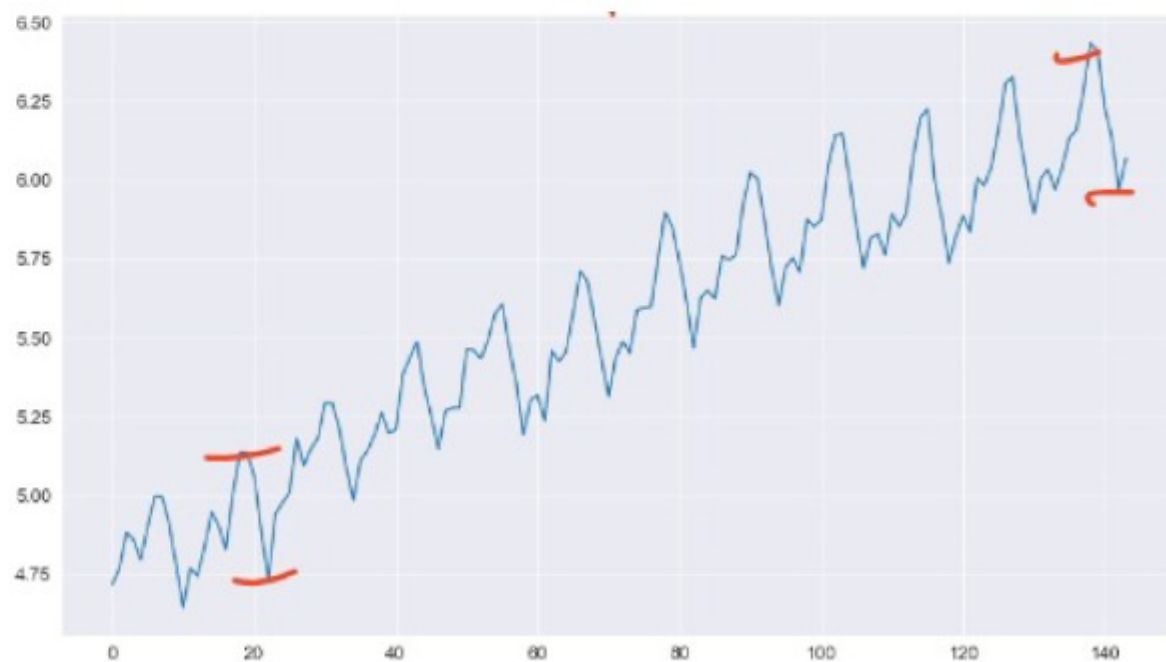
- **Standardization (Z-score normalization):** Scales data to have a mean of 0 and a standard deviation of 1. This method assumes a Gaussian distribution and is less affected by outliers.
- **Normalization (Min-Max scaling):** Transforms data to a range between 0 and 1, preserving the relative relationships between values. It's suitable when you want to maintain the distribution shape.
- **Max Abs Scaler:** Scales data to the range [-1, 1] by dividing through the maximum absolute value in the dataset. Useful for data that's centered at zero and sparse.
- **Robust Scaler:** Uses median and interquartile range (IQR) to scale data. It's less sensitive to outliers than standardization and normalization.
- **Quantile Transformer:** Maps the data to a uniform or normal distribution. Useful for mitigating the impact of outliers and for creating a more Gaussian-like distribution.
- **Power Transformation (Box-Cox or Yeo-Johnson):** Applies power transformations to make the data distribution more Gaussian-like, useful when dealing with skewed data.
- **Unit Vector Scaling:** Scales each feature by dividing by its magnitude, ensuring that each feature lies within the range of -1 to 1.
- **Log Transformation:** Taking the logarithm of the values can help with compressing large ranges and dealing with skewed data distributions.
- **Mean Centering:** Subtracts the mean of each feature from its values, making the mean of the centered data 0.
- **PCA (Principal Component Analysis):** Can be used for dimensionality reduction, which indirectly scales the data by transforming it into a new coordinate system.

Log transformation applied to data prior to Time-series analysis



Original data

After 'log transformation'



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.

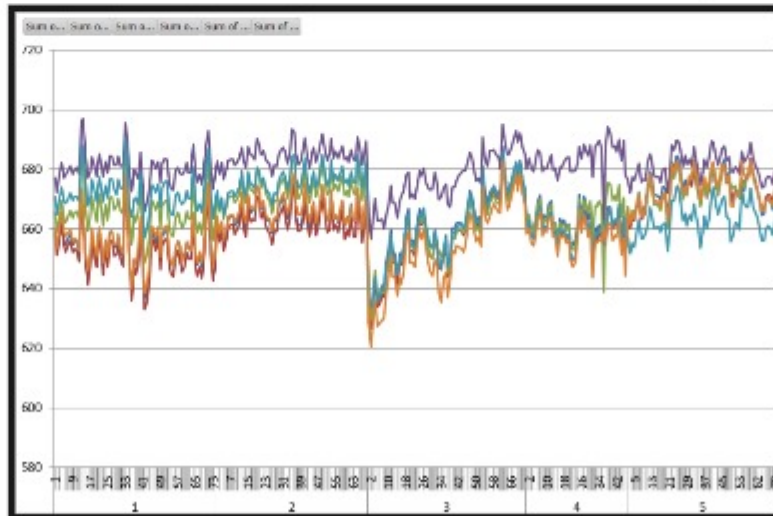
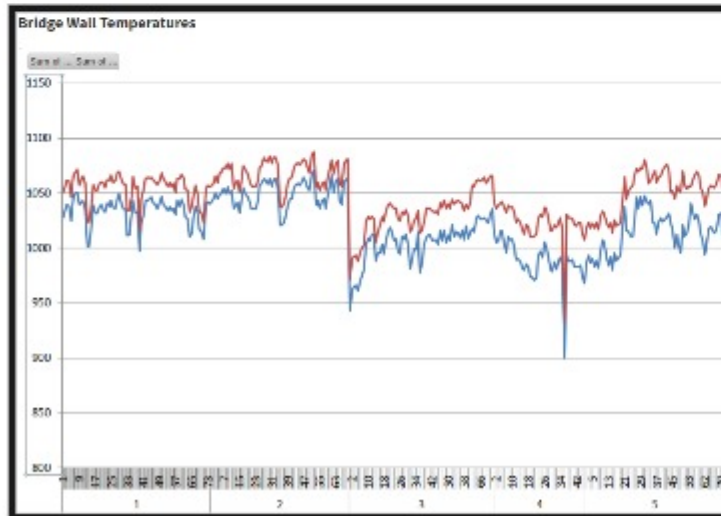
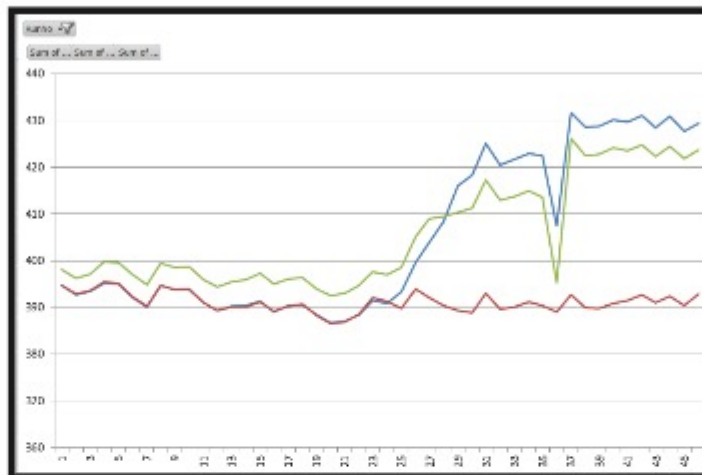
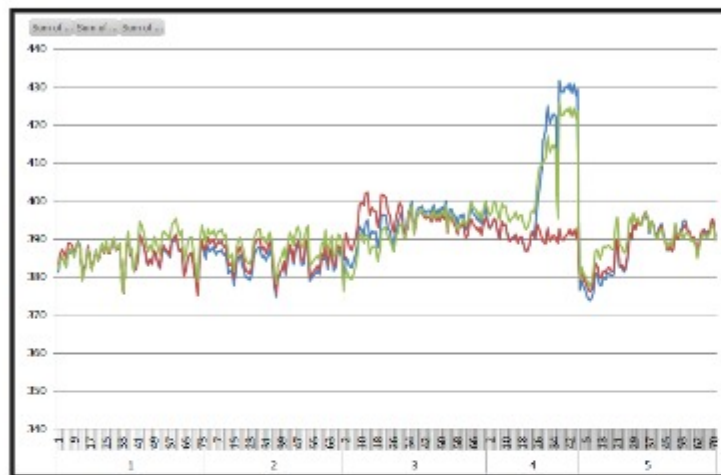
ML algorithms NOT susceptible to data scaling issues

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.

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

