Understanding Data

- Data Quality
- Data Transformation
  - Data Normalization

Data Quality

- The quality and quantity of training data is the most important aspect that decides the quality of the ML solution

- The data may be limited by several issues:

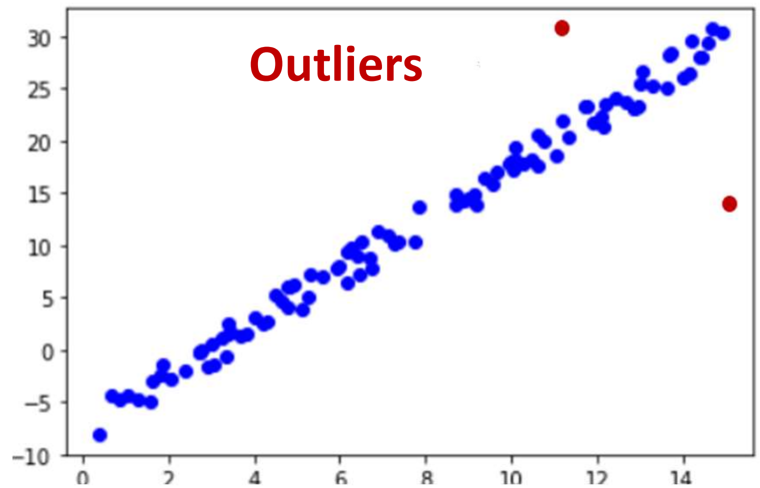
    - Outliers
    - Missing feature values
    - Limited quantity

Outliers in Data

- Outlier is an observation, i.e., unlike the other observation.

- Caused by
  - ✓ Measurement or input error
  - ✓ Data corruption
  - ✓ True outlier observation

- May cause problem during model fitting

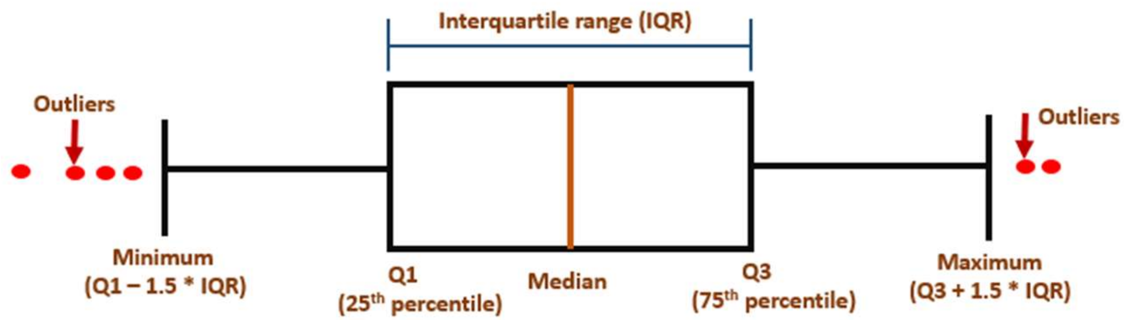


Outliers Detection – Box Plot

Dataset: 172 165 179 80 136 163 835 189 144 182 128

Step 1: 80 128 136 144 163 165 172 179 182 189 835

Step 2:
- ✓ Median: 165
- ✓ Q1: : 80 128 136 144 163 = 136
- ✓ Q3: 172 179 182 189 835 = 182
- ✓ IQR = 182-136 = 46



Step 3:
- ✓ Minimum: 136 – 1.5*46 = 67
- ✓ Maximum: 182 + 1.5*46 = 251

835 is the outlier in the data

Outliers Detection – Z Score

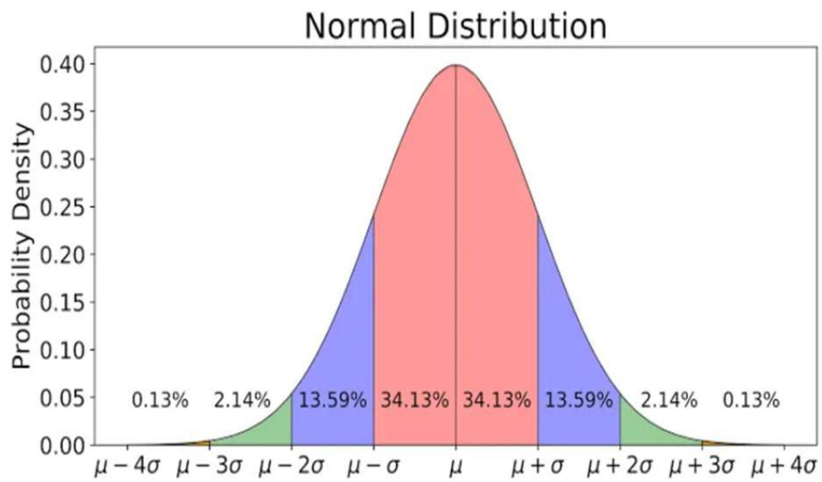
- The Z-score, also known as the standard score, measures how many standard deviations a data point is away from the mean of a dataset

$$Z = \frac{x - \mu}{\sigma}$$

- Approximately 68% of data points fall within 1σ of the mean (Z-scores between -1 and 1)

- About 95% fall within 2σ (Z-scores between -2 and 2)

- Approximately 99.7% fall within 3σ (Z-scores between -3 and 3)



http://www.cs.uni.edu/~campbell/stat/normfact.html

Z-scores assume that the data follows a normal distribution

Possible Solutions for Outlier Detection

- A model/learning algo. that can handle outliers
  - Robust statistics
  - Max-margin classifiers [SVMs]
- Data Cleanup
  - Label correction
  - Outlier detection and removal
    - Use of plots, visualization
    - Statistical measurements (quantiles)

  [“Outlier Analysis”, by Charu C. Aggarwal]

Missing Values

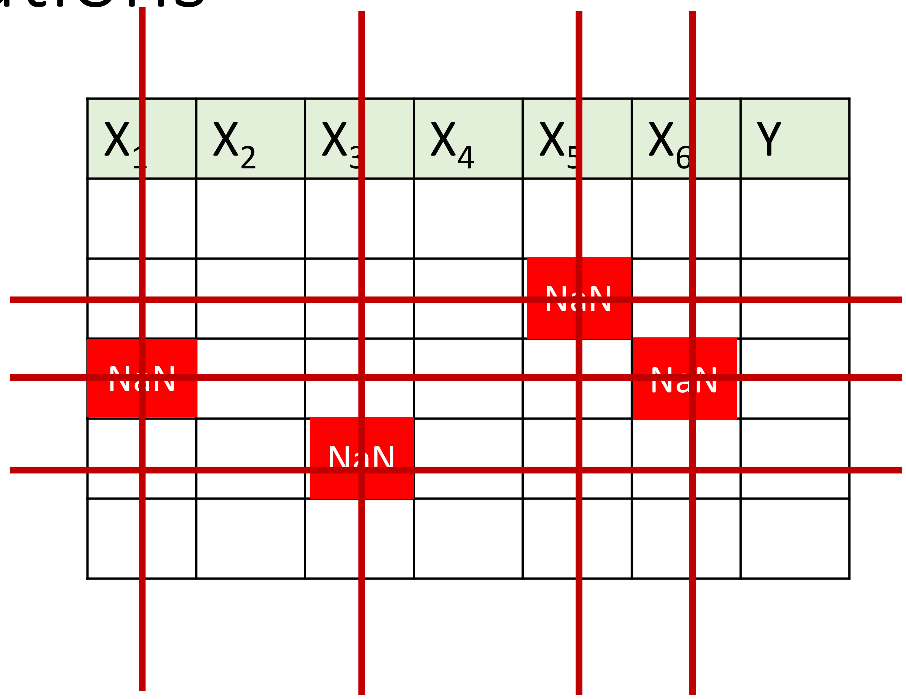
- Missing values in a dataset refer to data points or entries that are not recorded or are incomplete

- Can occur for various reasons:
  - ✓ data entry errors,
  - ✓ incomplete data collection,
  - ✓ sensor failures,
  - ✓ certain information was not available

- Can lead to:
  - ✓ Reduction in accuracy of the model.
  - ✓ A buildup of a biased model, leading to incorrect results

X_1	X_2	X_3	X_4	X_5	X_6	Y
				NaN		
NaN					NaN	
		NaN				

Missing Values – Possible Solutions



❑ ***Dropping Missing Values***

- Remove the feature from all samples
- Remove samples with missing data

❑ ***Imputing Missing Values***

- Mean Value Imputation
- Median Value Imputation
- Mode (Frequent Category) Imputation
- Random Sample Imputation

❑ ***Use models/algorithms that can work with Missing Values***

- Normalize distance in KNN
- Naïve Bayes model

Limited Quantity of Training Data - Problems

- Difficulty in Hyperparameter Tuning

Model Parameters: $\boldsymbol{\theta_i, i = 1..k}$
Training Data: $\boldsymbol{X_i,\ \ i = 1..n}$

Ideally: $\boldsymbol{n \gg k}$

- Overfitting: Happens in case of large $\boldsymbol{k}$ and small $\boldsymbol{n}$.

- Poor Generalization

Limited Quantity of Training Data - Solutions

- **Data Augmentation**: Augment your dataset by creating additional training examples through techniques like data rotation, flipping, cropping, or introducing small perturbations

- **Feature Engineering**: Carefully design and engineer informative features. High-quality features can help your model learn from the available data more effectively

- **Reduction of Parameters**: Consider using simpler models with fewer parameters

- **Alternative Techniques**: Apply regularization techniques or use semi-supervised learning as required

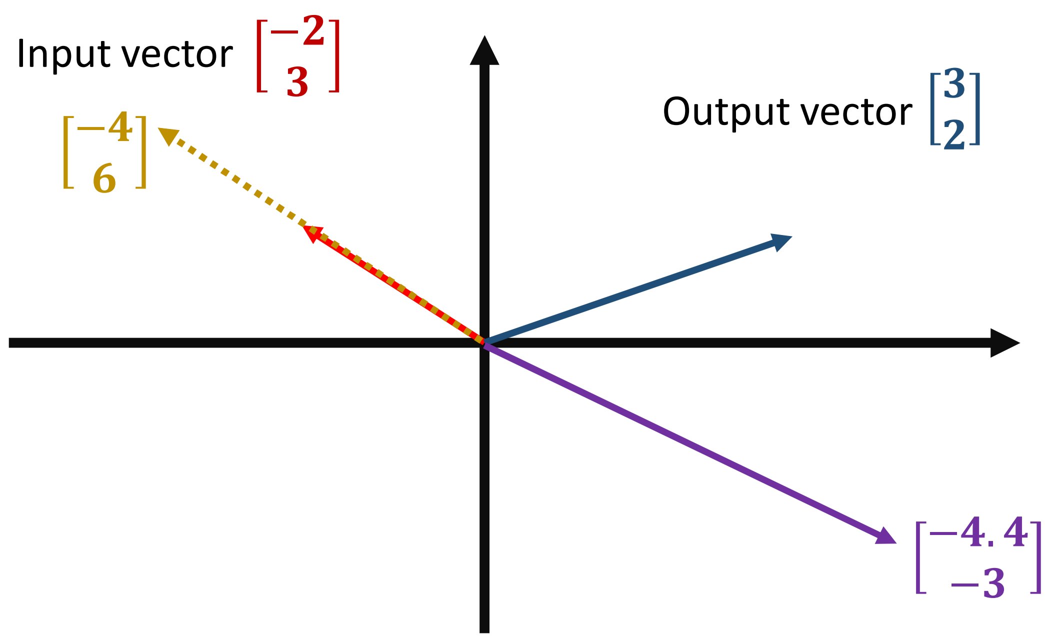
Data Transformation

Linear Transformation

X a = y X : matrix a, y : vector

- Linear Transformation is a function that maps an input vector into an output vector

  - ✓ Rotation $X = \begin{bmatrix} \cos\boldsymbol{\theta} & \sin\boldsymbol{\theta} \\ -\sin\boldsymbol{\theta} & \cos\boldsymbol{\theta} \end{bmatrix}$
    $= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \boldsymbol{\theta} = 90$
  - ✓ Scaling $X = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$
  - ✓ Shear $X = \begin{bmatrix} 1 & -0.8 \\ 3 & 1 \end{bmatrix}$



Linear Transformations – Basis Vectors

$$\hat{\imath} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \hat{\jmath} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

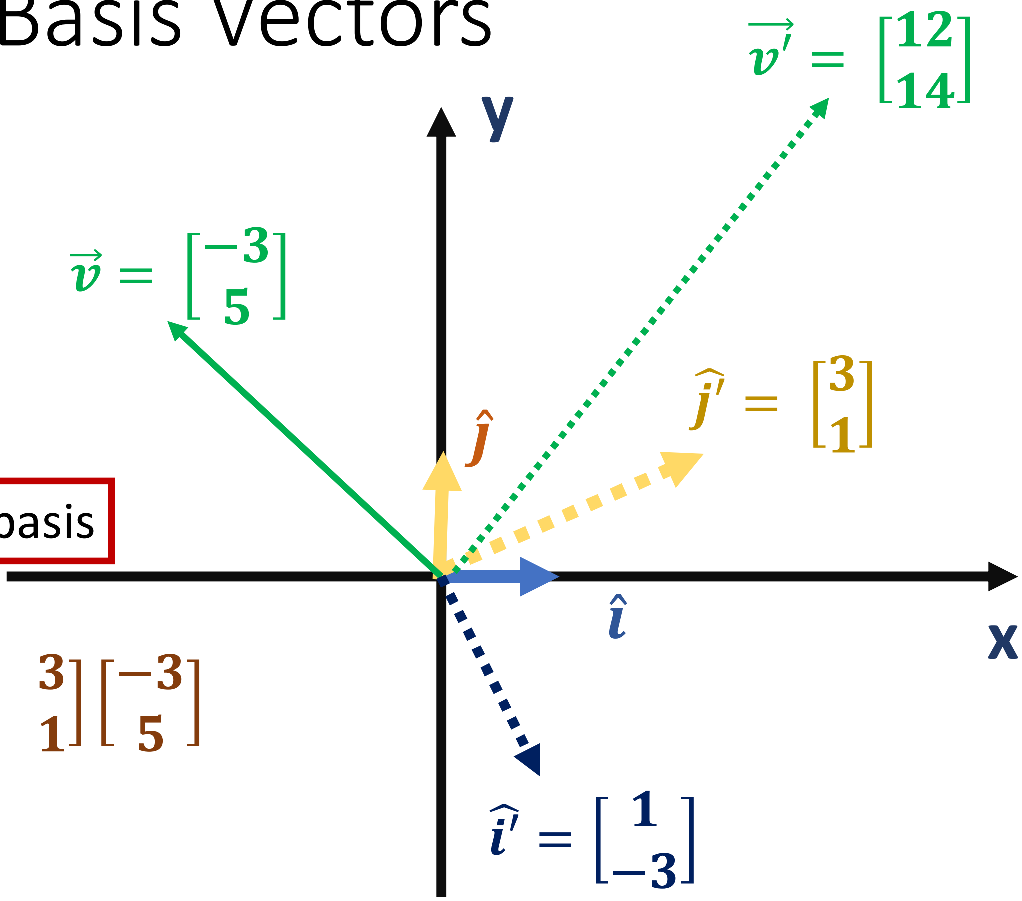
$$\vec{v} = \begin{bmatrix} -3 \\ 5 \end{bmatrix} = -3\hat{\imath} + 5\hat{\jmath}$$

$$\widehat{i'} = \begin{bmatrix} 1 \\ -3 \end{bmatrix}, \quad \widehat{j'} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Transformation of basis

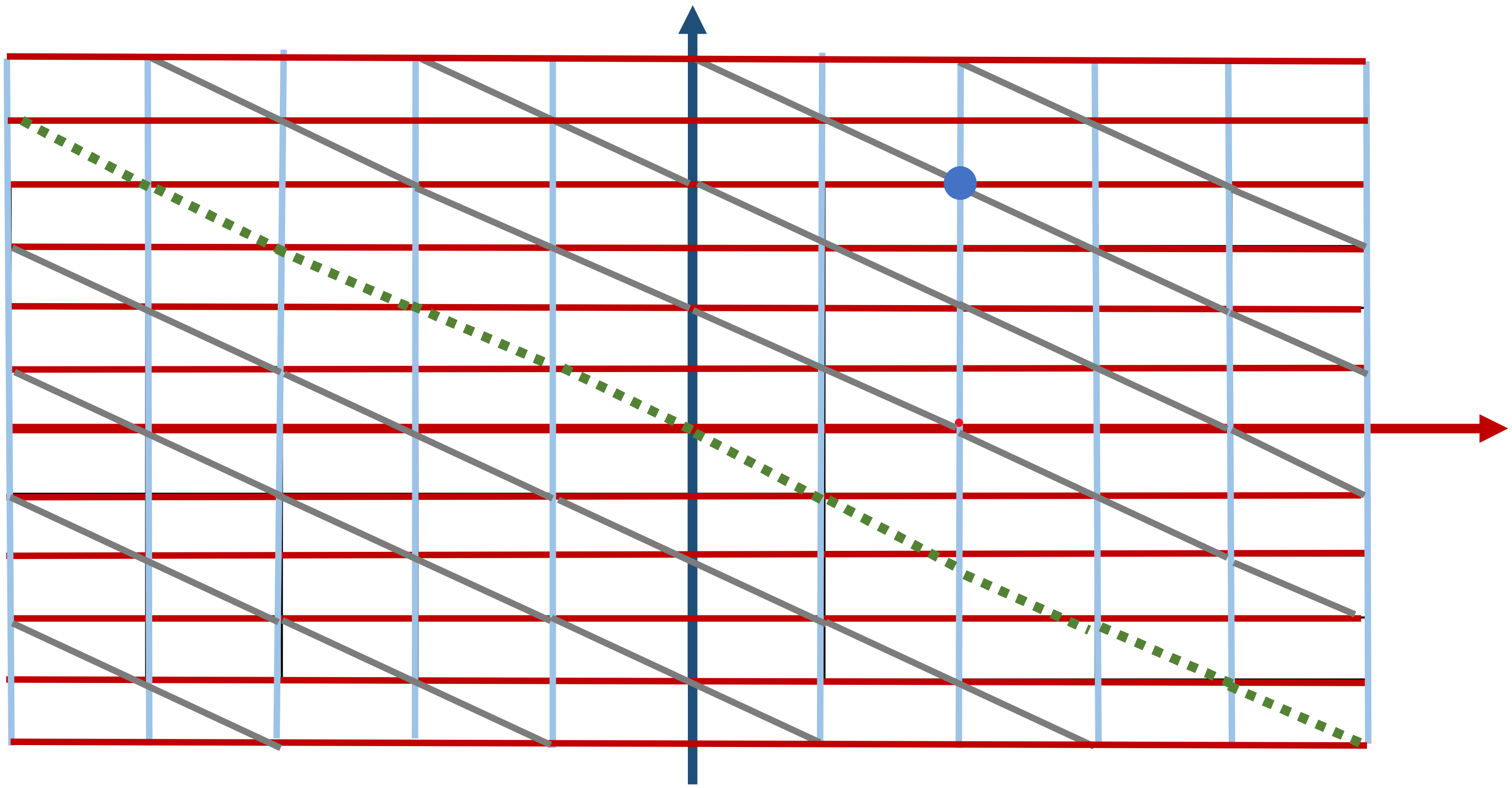
$$\overrightarrow{v'} = -3\widehat{i'} + 5\widehat{j'} = -3\begin{bmatrix} 1 \\ -3 \end{bmatrix} + 5\begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} -3 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$



Demo

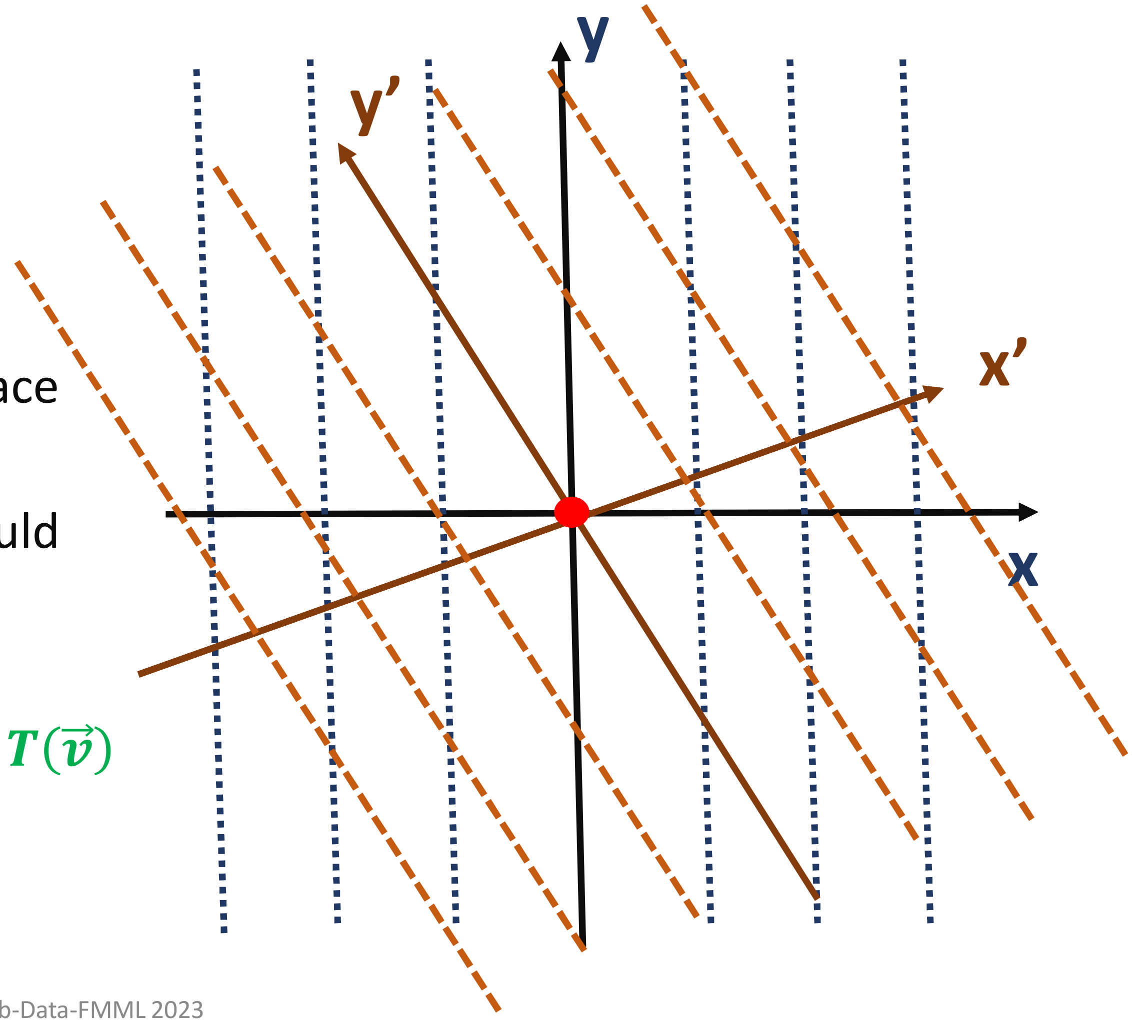
$\hat{\imath} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \hat{\jmath} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $p_1 = (2, 4)$ $\widehat{i'} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \widehat{j'} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$



Linear Transformations

- ✓ A line should remain a line once we transform our coordinate system
- ✓ The origin should remain at the fixed place
- ✓ The distance between the grid lines should remain equidistant

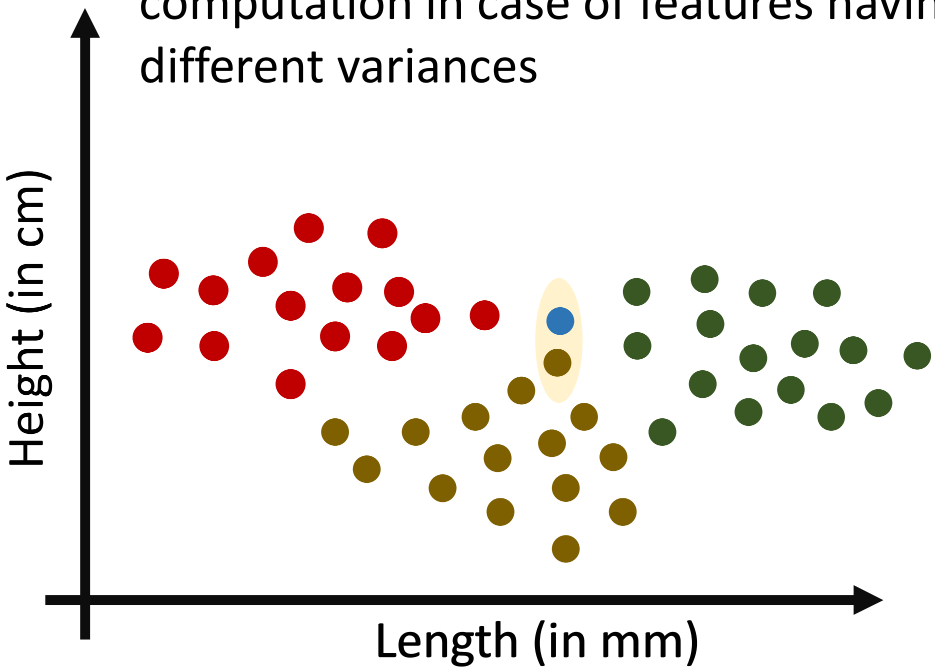
1. Additive Property: $\boldsymbol{T(\vec{u}+\vec{v}) = T(\vec{u}) + T(\vec{v})}$
2. Scalar Multiplication: $\boldsymbol{T(c\,\vec{u}) = c\,T(\vec{u})}$
3. Zero Vector Preservation: $\boldsymbol{T(0) = 0}$



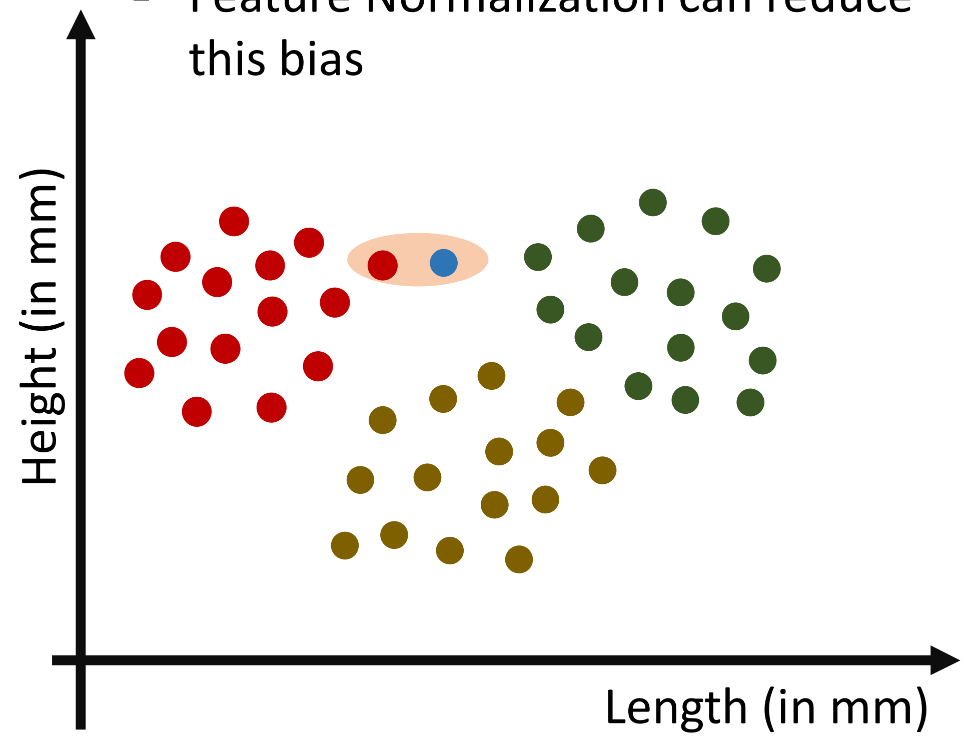
Data Normalization

Data Normalization

- Some features dominating distance computation in case of features having different variances



- Feature Normalization can reduce this bias



Feature Normalization

Price (y)	Lotsize(X_1)	Bedroom(X_2)	BR (X_3)	Age(X_4)
42000	5850	3	1	2
38500	4000	2	1	5
49500	3060	3	1	10
60500	6650	3	1	7
61000	6360	2	1	3
66000	4160	3	1	8

$$y = a_1X_1 + a_2X_2 + a_3X_3 + a_4X_4$$

Sample 1: $42000 = a_1 \times 5850 + a_2 \times 3 + a_3 \times 1 + a_4 \times 2$

$X_2, X_3, X_4 \ll X_1$

Should we neglect X_2, X_3 and X_4?

Why Normalization?

- A data set having numeric features covering distinctly different ranges (for example, weight and height, meters, miles, etc)
- A single numeric feature covering a wide range, such as "city population.“
- The above conditions with vastly different values will lead to a change in the weight of the variables, leading to low model accuracy.
- Normalization is used to transform data in a way that they are either dimensionless and/or have similar distributions, in turn improving accuracy by giving equal importance to features.

Standardization, Z-Score Normalization

- Involves transforming features, so that they have a mean (μ) of 0 and a standard deviation (σ) of 1

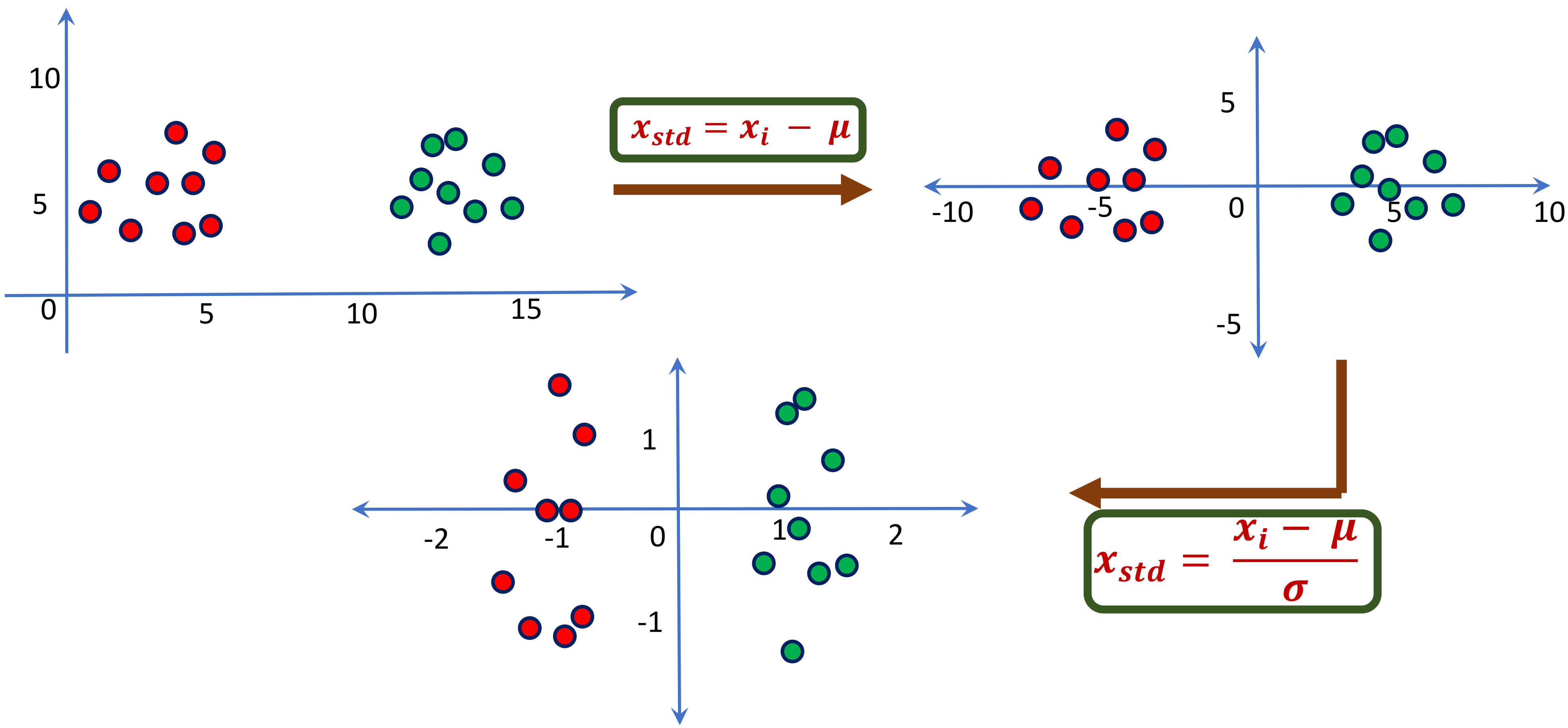
$$x_{std} = \frac{x_i - \mu}{\sigma}$$

- Necessary for algorithms sensitive to feature scales, like support-vector machines, k-means clustering etc.

- Ensures equal influence of each feature during model training, avoiding biases due to different feature scales

```
from sklearn.preprocessing import StandardScaler

# Assuming X is your feature data
# Initialize the StandardScaler
scaler = StandardScaler()

# Fit the scaler on the data and transform the data
X_scaled = scaler.fit_transform(X)

# X_scaled now contains the standardized features
```

Standardization, Z-Score Normalization



Min Max Normalization/Rescaling/Feature Scaling

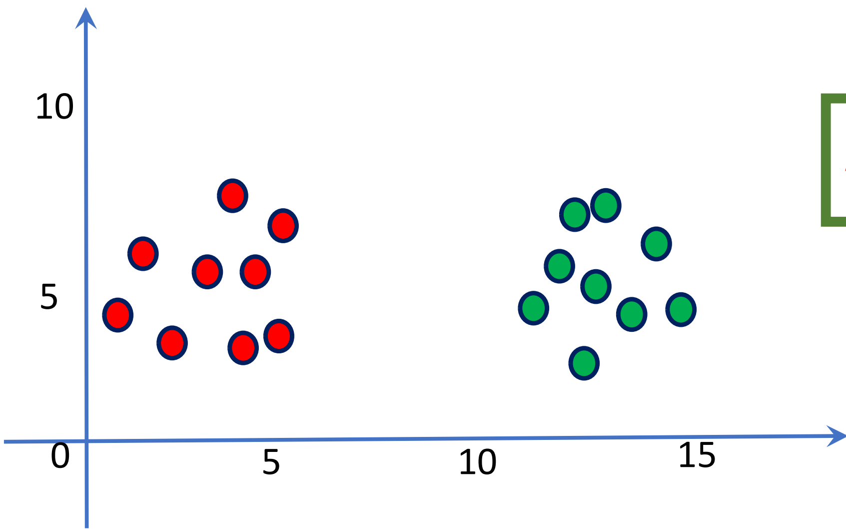
- Converts feature values from their natural range within a specific range, typically [0,1]

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

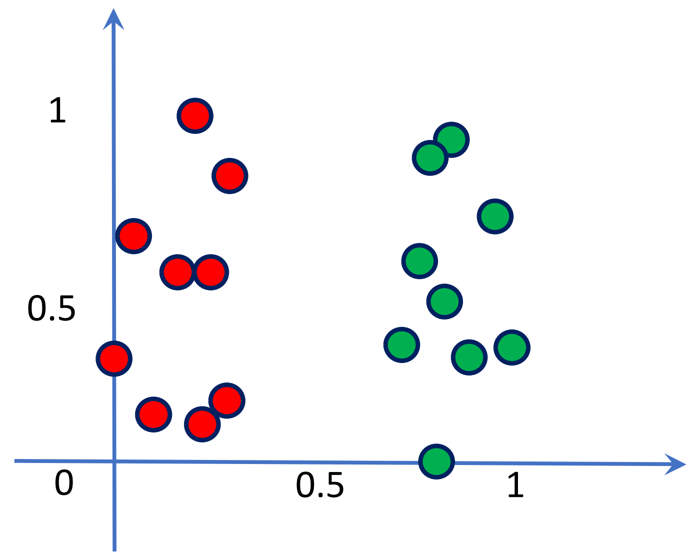
- ✓ Data to be uniformly distributed across the range, e.g.: age. Income may not be a good choice
- ✓ The upper and lower bounds on the data should be known with few or no outliers
- ✓ Influenced by outliers

```
from sklearn.preprocessing import MinMaxScaler

# Assuming X is your feature data
# Initialize the MinMaxScaler
scaler = MinMaxScaler()

# Fit the scaler on the data and transform the data
X_normalized = scaler.fit_transform(X)

# X_normalized now contains the min-max normalized features
```

Min Max Normalization/Rescaling/Feature Scaling



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



Feature Clipping/Capping

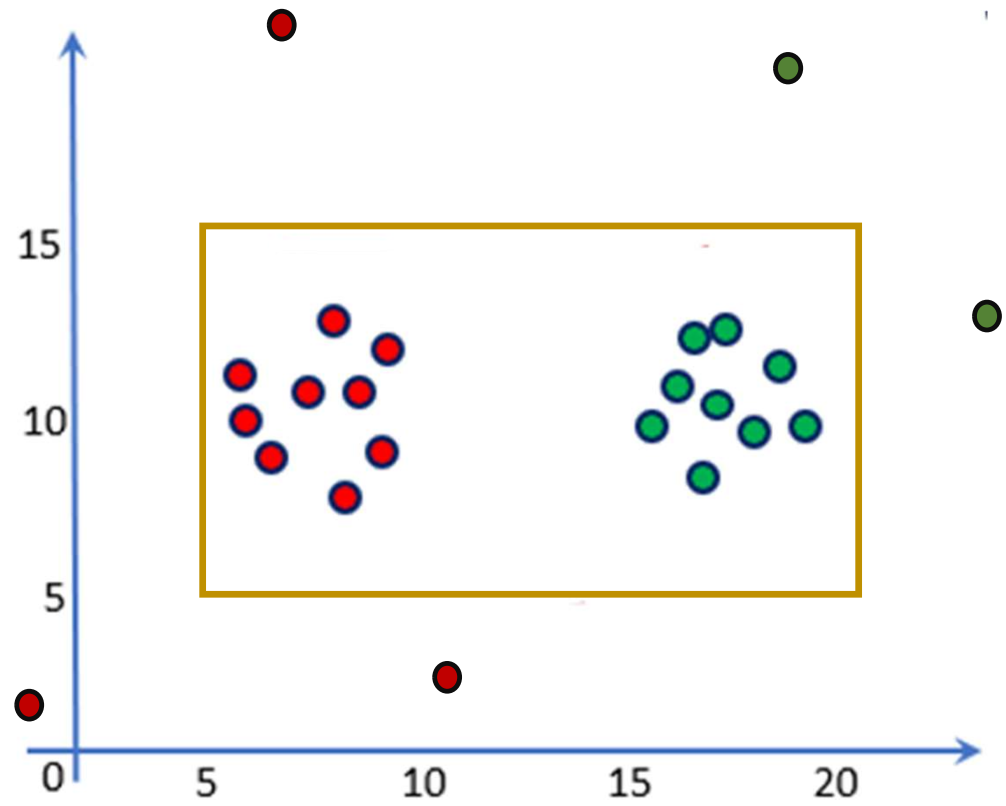
- Takes care of outliers, by limiting or bounding the extreme values of a feature within a specified range
- **Determine clipping thresholds**: The upper and lower thresholds are typically determined based on a specific percentile or a fixed value
- **Cap extreme values**: The feature is constrained within a specific range

```python
import numpy as np

def clip_feature(feature, lower_threshold, upper_threshold):
    """
    Clip the feature values between lower_threshold and upper_threshold.
    """
    clipped_feature = np.clip(feature, lower_threshold, upper_threshold)
    return clipped_feature

# Example usage
feature = np.array([10, 15, 200, 5, 25, 180])
lower_threshold = 0
upper_threshold = 100

clipped_feature = clip_feature(feature, lower_threshold, upper_threshold)
print("Original feature:", feature)
print("Clipped feature:", clipped_feature)
```

Feature Clipping/Capping

- Assume that in the figure:

$x \, \epsilon \, [5, 20]$
$y \, \epsilon \, [5, 15]$

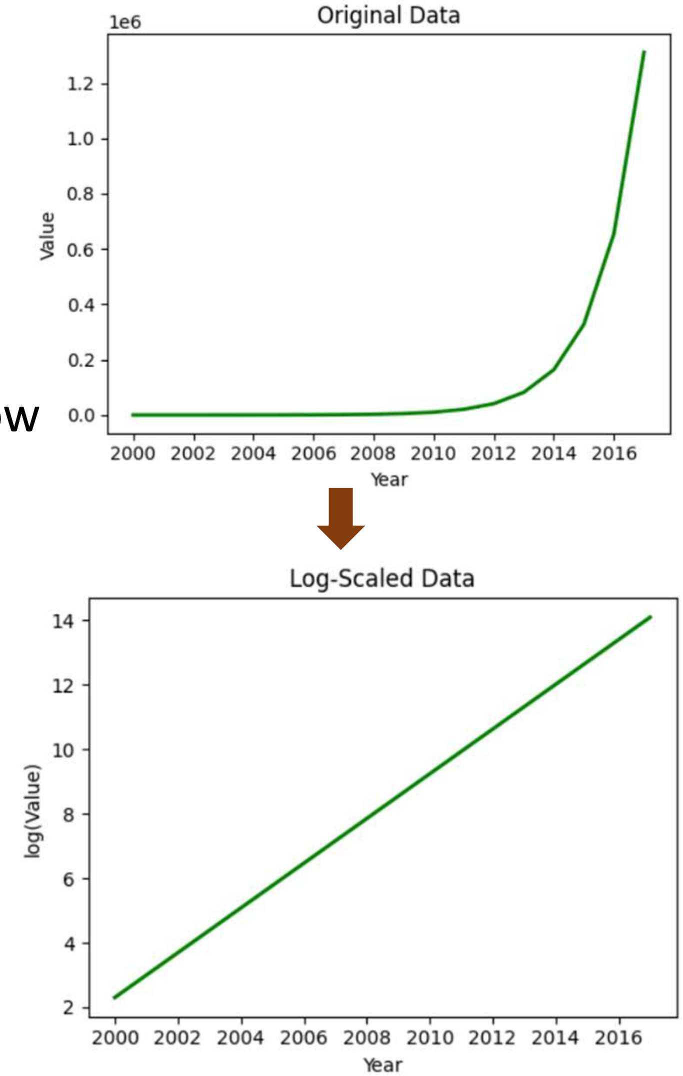


Log Scaling

- Log Scaling is used when dealing with data that spans a wide range of values

- It computes the log to compress a wide range to a narrow range, for a more balanced visualization

- It helps in revealing patterns and trends in the data, especially in cases where there is a significant difference in magnitude between the smallest and largest values



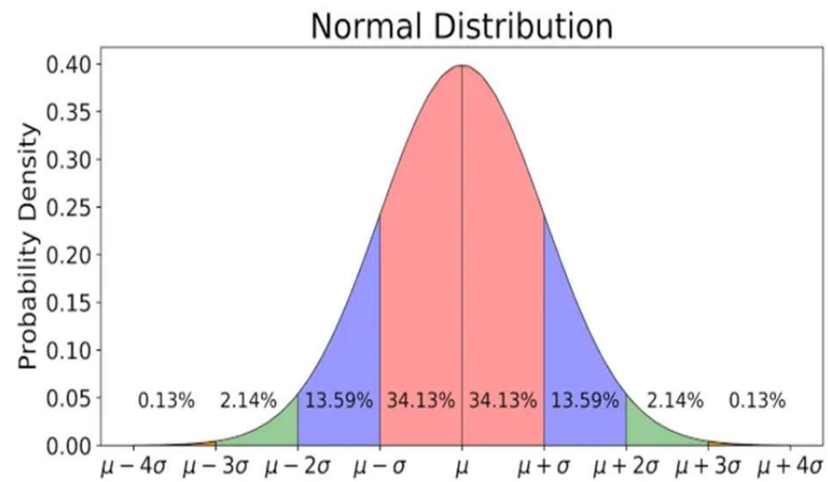
Summary

Normalization Technique	Formula	When to Use
Standardization	$x_{std} = \frac{x_i - \mu}{\sigma}$	When the feature distribution does not contain extreme outliers
Min-Max Scaling	$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$	When the feature is ore-or-less uniformly distributed across a fixed range
Feature Clipping	If x > max, then x’ = max If x < min, then x’ = min	When the feature contains some extreme outliers
Log Scaling	x’ = log(x)	When the feature conforms to the power law

Back-Up Slides

Box-Plot

- About 68.26% of the whole data lies within one standard deviation (<σ) of the mean (μ), taking both sides into account, the pink region in the figure.
- About 95.44% of the whole data lies within two standard deviations (**2σ**) of the mean (μ), taking both sides into account, the pink+blue region in the figure.
- About 99.72% of the whole data lies within three standard deviations (**<3σ**) of the mean (μ), taking both sides into account, the pink+blue+green region in the figure.
- And the rest 0.28% of the whole data lies outside three standard deviations (**>3σ**) of the mean (μ), taking both sides into account, the little red region in the figure. **And this part of the data is considered as outliers.**
- The first and the third quartiles, *Q1* and *Q3*, lies at **-0.675σ** and **+0.675σ** from the mean, respectively.



http://www.cs.uni.edu/~campbell/stat/normfact.html

Scale 1

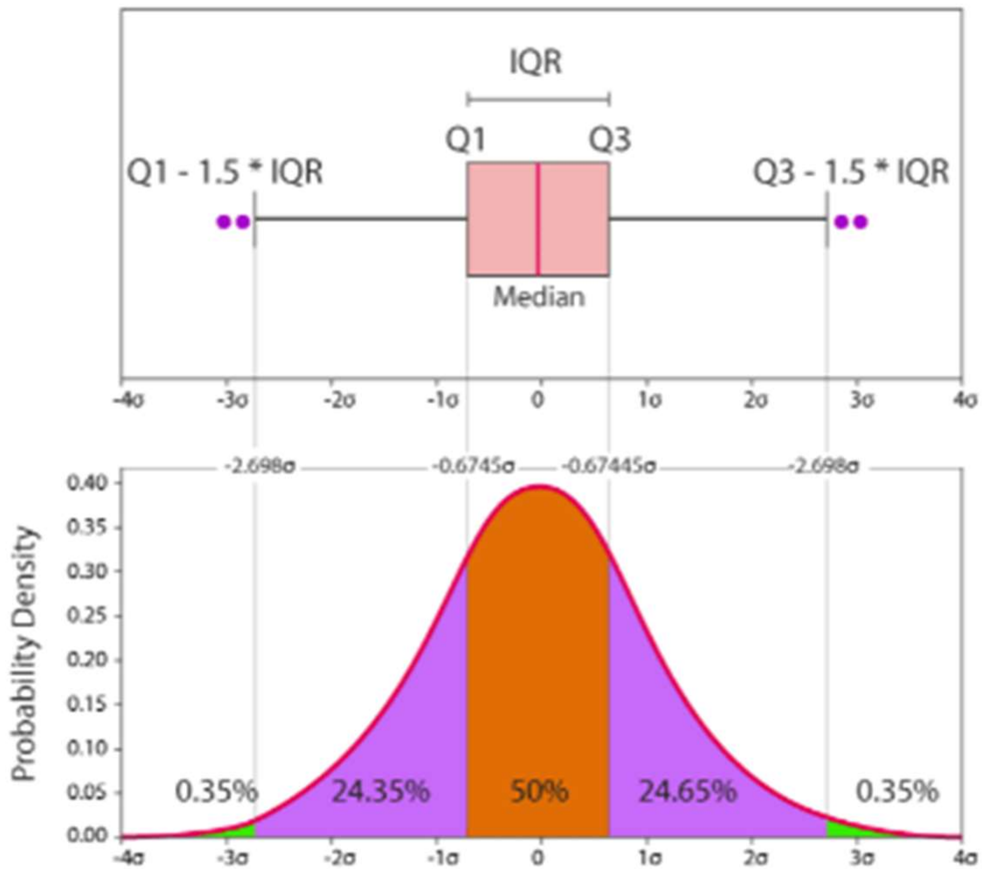
```
Lower Bound:

= Q1 - 1 * IQR
= Q1 - 1 * (Q3 - Q1)
= -0.675σ - 1 * (0.675 - [-0.675])σ
= -0.675σ - 1 * 1.35σ
= -2.025σ

Upper Bound:

= Q3 + 1 * IQR
= Q3 + 1 * (Q3 - Q1)
= 0.675σ + 1 * (0.675 - [-0.675])σ
= 0.675σ + 1 * 1.35σ
= 2.025σ
```

Scale 2

```
Lower Bound:

= Q1 - 2 * IQR
= Q1 - 2 * (Q3 - Q1)
= -0.675σ - 2 * (0.675 - [-0.675])σ
= -0.675σ - 2 * 1.35σ
= -3.375σ

Upper Bound:

= Q3 + 2 * IQR
= Q3 + 2 * (Q3 - Q1)
= 0.675σ + 2 * (0.675 - [-0.675])σ
= 0.675σ + 2 * 1.35σ
= 3.375σ
```

Scale 1.5

```
Lower Bound:

= Q1 - 1.5 * IQR
= Q1 - 1.5 * (Q3 - Q1)
= -0.675σ - 1.5 * (0.675 - [-0.675])σ
= -0.675σ - 1.5 * 1.35σ
= -2.7σ

Upper Bound:

= Q3 + 1.5 * IQR
= Q3 + 1.5 * (Q3 - Q1)
= 0.675σ + 1.5 * (0.675 - [-0.675])σ
= 0.675σ + 1.5 * 1.35σ
= 2.7σ
```



Boxplot on a normal distribution