Data Collection and Preprocessing

Unit 2

Data Collection

Data collection is a systematic process essential for gathering information that can be analyzed to derive insights and support decision-making across various fields, including business, healthcare, and research.

Data collection involves acquiring observations or measurements systematically to address specific research questions or business needs.

It is the foundational step in data analysis, enabling organizations to make informed decisions based on accurate and relevant information

Types of Data Collection

Primary Data Collection: Involves gathering original data directly from sources.

This method is tailored to specific research objectives and includes techniques such as:

Surveys: Structured questionnaires distributed online, in person, or via phone to gather opinions or characteristics from a group.

Interviews: Direct interactions with respondents, which can be structured, semi-structured, or unstructured.

Experiments: Controlled studies to test hypotheses by manipulating variables and observing outcomes.

Types of Data Collection

Secondary Data Collection:

Utilizes existing data collected by others for different purposes. This method is often quicker and less expensive than primary data collection and includes sources such as:

Published Research: Academic journals, reports, and articles.

Online Databases: Repositories that provide access to statistical information and social surveys.

Government Records: Data maintained by governmental agencies that can be used for analysis

Data Collection Methods

Several methods are employed within these categories:

Surveys: Effective for understanding general characteristics or opinions.

Observations: Gathering data in natural settings without interference.

Web Scraping: Automated extraction of data from websites.

APIs: Interfaces that allow access to structured data from services.

Sensors: Devices that collect real-time data from the environment.

Steps in Data Collection

The process typically involves several key steps:

Define Research Objectives: Clearly articulate what information is needed.

Select Data Sources: Identify where relevant data can be obtained.

Choose Methods and Tools: Decide on the most suitable methods for collection.

Plan Procedures: Develop a detailed plan for how data will be collected and managed.

Collect Data: Implement the chosen methods while ensuring quality control

Data Issues

Data quality issues can significantly impact an organization's ability to make informed decisions and achieve operational efficiency.

Common Data Quality Issues

Inaccurate Data:

Description: Data that contains errors or does not reflect the true values.

Implications: Leads to poor decision-making and unreliable analytics.

Solutions: Implement robust data quality monitoring tools to identify inaccuracies and compare against known accurate datasets for correction

Data Issues

Incomplete Data:

Description: Records missing critical information (e.g., missing ZIP codes or demographic details).

Implications: Results in flawed analyses and operational challenges.

Solutions: Use data validation techniques during data entry and implement processes to fill in missing information

Data Issues

Duplicate Data:

Description: Multiple entries of the same record, which can inflate counts and skew analysis.

Implications: Wastes storage space and leads to confusion in reporting.

Solutions: Employ deduplication technologies to identify and merge or eliminate duplicate records

Data Issues

Inconsistent Formatting:

Description: Variations in how data is formatted (e.g., date formats).

Implications: Causes difficulties in data integration and analysis.

Solutions: Standardize formats across datasets and use automated tools to ensure consistency

Data Issues

Ambiguous Data:

Description: Data that can be interpreted in multiple ways due to unclear definitions or formatting.

Implications: Leads to misinterpretation and inaccurate reporting.

Solutions: Continuously monitor for ambiguity using autogenerated rules to clarify data meanings

Data Issues

Stale Data:

Description: Outdated information that no longer reflects current conditions.

Implications: Can result in missed opportunities or incorrect analyses.

Solutions: Regularly review and update datasets to ensure relevance

Data Issues

Unstructured Data:

Description: Data that lacks a predefined format, making it difficult to analyze (e.g., free-text fields).

Implications: Hinders effective data processing and analysis.

Solutions: Utilize natural language processing (NLP) tools to structure unstructured data for better usability

Data Issues

Hidden or Dark Data:

Description: Data that is collected but not used or analyzed, often overlooked in databases.

Implications: Represents missed insights and opportunities for improvement.

Solutions: Use data cataloging tools to uncover hidden data and assess its potential value

Data Issues

Data Overload:

Description: An overwhelming volume of data that complicates analysis efforts.

Implications: Can lead to analysis paralysis, where important insights are lost in the noise.

Solutions: Implement filtering techniques to prioritize relevant data for analysis

Data Issues

Human Error:

Description: Mistakes made during data entry or management processes.

Implications: Introduces inaccuracies and inconsistencies into datasets.

Solutions: Provide training on data management best practices and automate data capture where possible

Methods to handle incomplete data

1. Deletion Methods

Listwise Deletion: Remove all data for any observation that has one or more missing values. This is straightforward but can lead to a significant loss of data if many observations are incomplete.

Pairwise Deletion: Only exclude missing values when they are needed for a specific analysis, allowing for the use of available data across different variables. This method maximizes data usage but may lead to inconsistencies in results.

Dropping Variables: If a variable has a high proportion of missing data, it may be more effective to remove it entirely from the dataset, especially if it does not contribute significantly to the analysis

Methods to handle incomplete data

2. Imputation Techniques

Mean/Median/Mode Imputation: Replace missing values with the mean, median, or mode of the available data. This is simple but can reduce variability in the dataset.

K-Nearest Neighbors (KNN) Imputation: Estimate missing values based on the values of similar observations. This method considers relationships between variables and can provide more accurate imputations.

Regression Imputation: Use regression models to predict and fill in missing values based on other available data.

Multiple Imputation: Generate multiple datasets with different imputed values for the missing data and analyze them separately before combining results. This method accounts for uncertainty associated with missing values

Noisy Data

Noisy data refers to unwanted variations or distortions in datasets that obscure the underlying patterns or signals.

Noisy data contains irrelevant, erroneous, or random information that can arise from various sources, including measurement errors, data entry mistakes, hardware malfunctions, or environmental factors.

This noise can lead to misleading insights and affect model performance.

Types of Noisy Data

Random Noise: This type of noise is extraneous information that does not correlate with the underlying data. It often occurs due to inaccuracies in measurements or recording methods.

Misclassified Data: Data points that are incorrectly labeled or categorized, often due to human error during data entry or processing.

Outliers: Extreme values that deviate significantly from other observations in the dataset can distort analysis and model training.

Irrelevant Features: Features that do not contribute to the predictive power of the model can introduce noise and complicate analysis.

Measurement Errors: Variations caused by inaccuracies in sensors or instruments used for data collection.

Techniques for handling noisy data

Data Cleaning:

Filtering: Remove unwanted data points or categories from the dataset. This includes filtering out outliers and irrelevant features.

Binning: Grouping data into bins to reduce variance and smooth out noise by replacing values with bin means or medians.

Techniques for handling noisy data

Outlier Detection and Treatment:

Use statistical methods (e.g., Z-scores, IQR) to identify outliers and decide whether to remove them or cap their values.

Techniques for handling noisy data

Feature Selection:

Identify and eliminate irrelevant features using techniques such as:

Filter Methods: Statistical tests to assess feature relevance.

Wrapper Methods: Adding/removing features based on model performance.

Embedded Methods: Regularization techniques that incorporate feature selection within model training.

Techniques for handling noisy data

Regression Techniques:

Use regression analysis to determine relationships between variables and filter out noise by modeling the expected output based on input features.

Machine Learning Algorithms:

Implement robust algorithms that can handle noisy data effectively, such as decision trees or ensemble methods (e.g., random forests) which reduce sensitivity to noise.

Autoencoders:

Utilize neural networks designed for unsupervised learning to detect anomalies in datasets by reconstructing input data while filtering out noise.

Handling inconsistent data

Inconsistent data can arise from various sources and can lead to inaccurate analyses and unreliable results.

Handling inconsistent data is a crucial aspect of data cleaning and preprocessing in data science.

Syntactic inconsistencies: Variations in data formats (e.g., date formats like DD/MM/YYYY vs. MM/DD/YYYY).

Semantic inconsistencies: Discrepancies in meaning or interpretation of data values (e.g., different spellings for the same category).

Structural inconsistencies: Issues that arise when merging datasets with different schemas or structures.

Step to handle inconsistency

Identify Inconsistencies:

Use data profiling techniques to explore the dataset and identify anomalies, missing values, duplicates, and format discrepancies.

Tools like .info(), .describe(), and .value_counts() in Pandas can help summarize the dataset and reveal inconsistencies 2.

Define Consistency Rules:

Establish clear rules for what constitutes consistent data. This includes specifying formats, naming conventions, and acceptable value ranges 1.

For example, ensure all dates are formatted as "YYYY-MM-DD" or that categorical variables use standardized labels.

Step to handle inconsistency

Data Cleaning Techniques:

Standardization: Normalize data formats across the dataset using functions like .replace() or .map() in Pandas to correct known inconsistencies.

Deduplication: Remove duplicate entries that may inflate sample size or skew analysis results using methods such as drop_duplicates() in Pandas.

Recoding: Use functions to recode categorical variables that may have inconsistent spellings or categories (e.g., converting "old_category" to "corrected_category")

Step to handle inconsistency

Data Validation Checks:

Implement validation checks to ensure that data adheres to defined consistency rules. This includes checking for valid date ranges, numeric limits, and categorical values 13.

Use automated scripts or data quality tools to perform these checks regularly.

Historical Data Analysis:

Analyze historical data to identify patterns of inconsistency. This can help uncover recurring issues and inform preventive measures

Data Pre-processing

It is a data mining technique that involves transforming raw data into a understandable format

Need of Data preprocessing:

- Data in the real world are:
  - Incomplete
  - Noisy
  - Inconsistent
- No Quality data no quality mining results!

Measure of Data Quality

- Accuracy
- Completeness
- Consistent
- Timeliness
- Believability
- Value added
- Interpretability
- Accessibility

Data Preprocessing techniques

- Data Cleaning
- Data integration
- Data transformation
- Data reduction

Data Integration

Data Integration implies combining of data from multiple sources into a data warehouse.

Issues in data integration:

- Entity identification problem
- Redundancy
- Tuple Duplication
- Detecting data value conflicts

Data Transformation

Data transformation is a critical process in data management that involves converting data from one format or structure into another to make it suitable for analysis, storage, and reporting.

Need for Data transformation:

- Enhancing Data Quality
- Facilitating Data Integration
- Optimizing Data Storage
- Improving Accessibility
- Enabling Advanced Analytics
- Supporting Business Intelligence

Data Transformation Techniques

1. Data Cleansing
2. Data Aggregation
3. Data Smoothing
4. Attribute Construction
5. Normalization / Standardization

Example

```
#Import Some libraries

import pandas as pd

import numpy as np
```

Creating data set

```
# Create a sample DataFrame with inconsistent data
data = {
    'Name': ['Ram', 'Hari', 'Suman', 'David', 'Priti'],
    'Age': [25, np.nan, 30, 22, np.nan],  # Some missing values
    'City': ['Kathmandu', 'Pokhara', 'pokhara', 'KATHMANDU', 'BUTWAL'],  # Inconsistent city names
    'Salary': [70000, 80000, np.nan, 60000, 75000]
}
```

Creating data frame

```
df = pd.DataFrame(data)
# Display the original DataFrame
print("Original DataFrame:")
print(df)

#Visualize using head()
df.head()
```

Fill missing values and maintain consistency in the data

```
# Fill missing values in the Age column with the mean age
df['Age'].fillna(df['Age'].mean(), inplace=True)

# Standardize city names to lower case for consistency
df['City'] = df['City'].str.lower()

# Display the cleaned DataFrame
print("\nCleaned DataFrame:")
print(df)
```

Data Transformation

```
# Group by city and calculate average salary
average_salary = df.groupby('City')['Salary'].mean().reset_index()

# Rename columns for clarity
average_salary.rename(columns={'Salary': 'Average_Salary'}, inplace=True)

#Fill missing values of salary column using median
df['Salary'].fillna(df['Salary'].median(),inplace=True)
```

Data Transformation

```
# Display average salary by city
print("\nAverage Salary by City:")
print(average_salary)


# Add a new column for salary after a 10% raise
df['New_Salary'] = df['Salary'] * 1.10


# Display the final transformed DataFrame
print("\nFinal Transformed DataFrame:")
print(df)
```

Outliers

Outliers are data points that deviate significantly from the majority of observations in a dataset.

They can arise from various sources, including measurement errors, data entry mistakes, or genuine variability in the data.

Identifying and handling outliers is crucial in data science as they can skew results, distort statistical analyses, and negatively impact machine learning models.

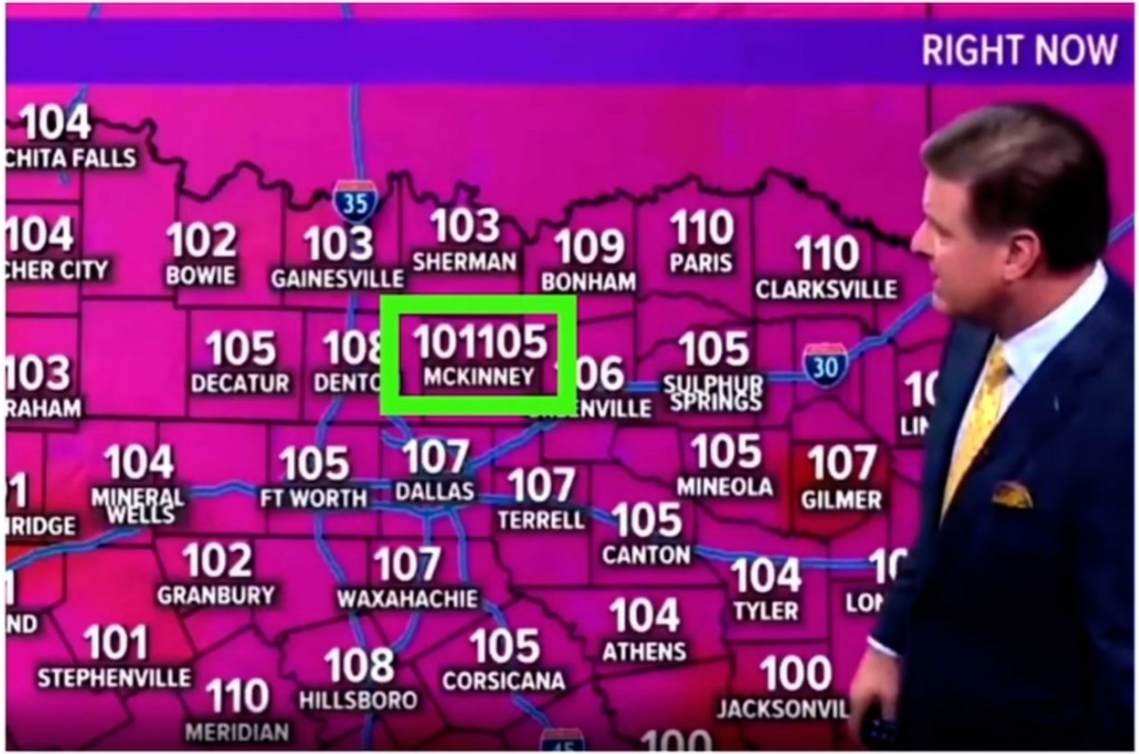
What to do with outliers?

- People have lots of advice on what to do with “outliers”…
  - **Remove** data point
  - **Impute** new value
  - **Winsorization**
  - Use **robust** analysis methods

“Everyone in McKinney is dead.”

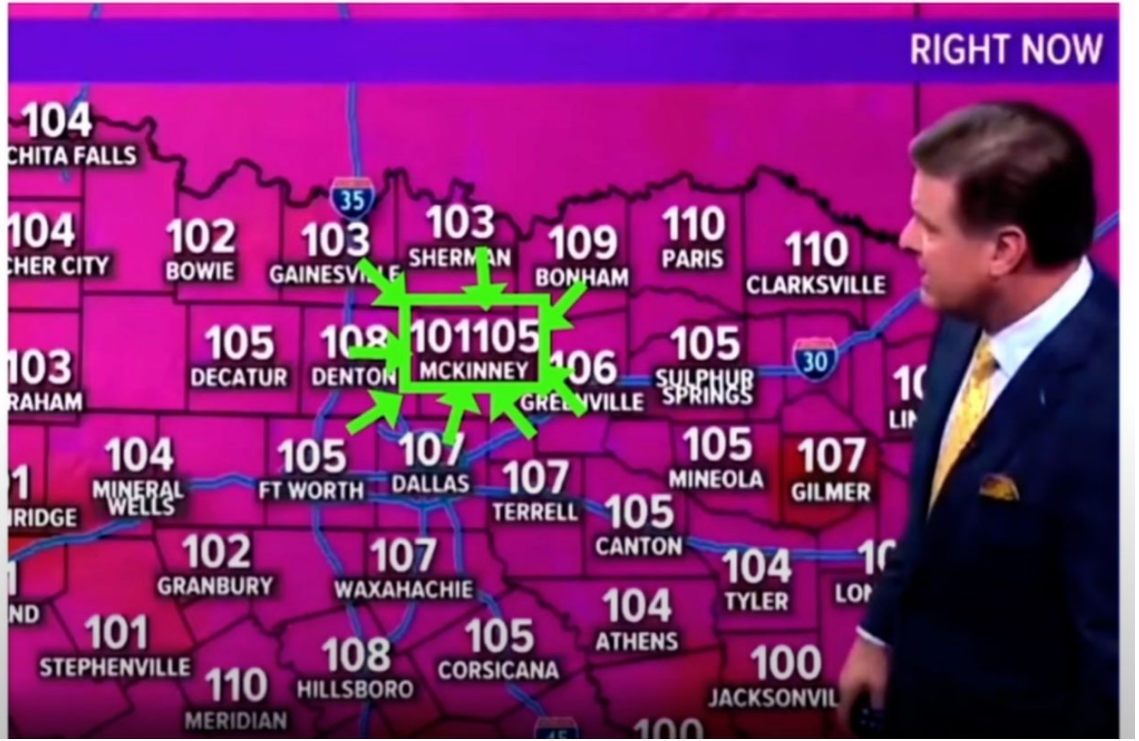
In a famous news segment, the weatherman laments the death of all residents of McKinney, Texas.

101105 degrees is very hot (even in Fahrenheit).



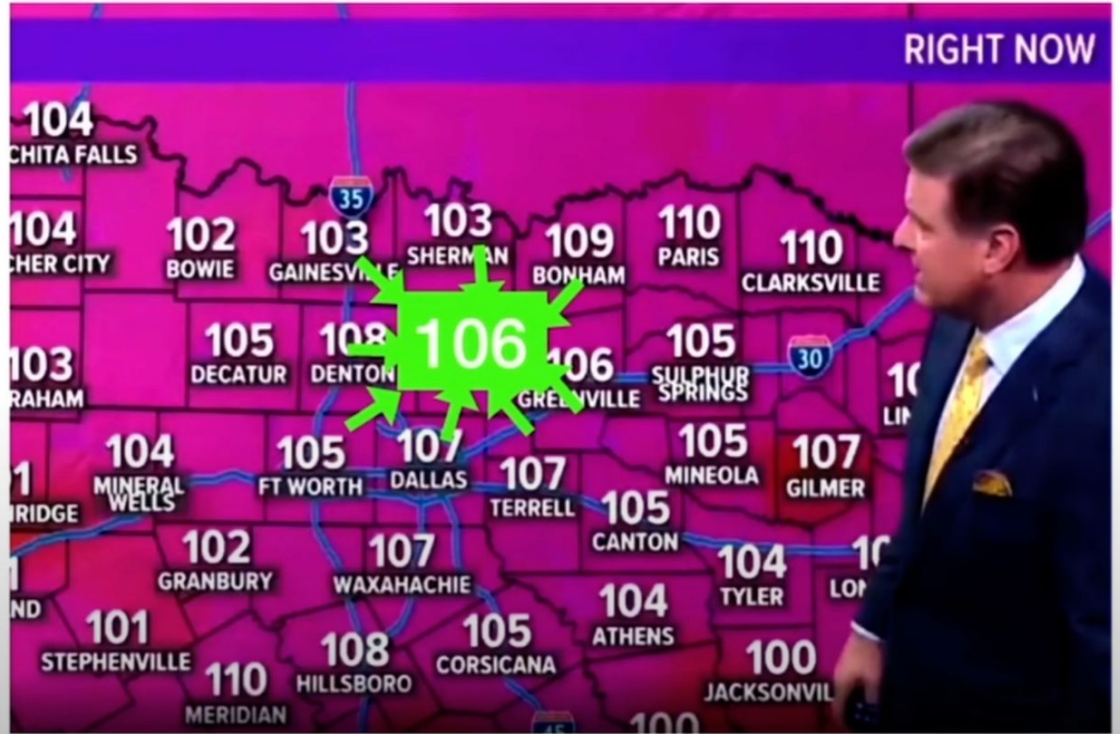
“Everyone in McKinney is dead.”

- Sometimes outliers are clearly just mistakes.
- We could just remove McKinney from the data.
- We could use contextual clues to guess whether they meant to type 101 or 105 or 110.
- We could impute a new value by using nearby values.



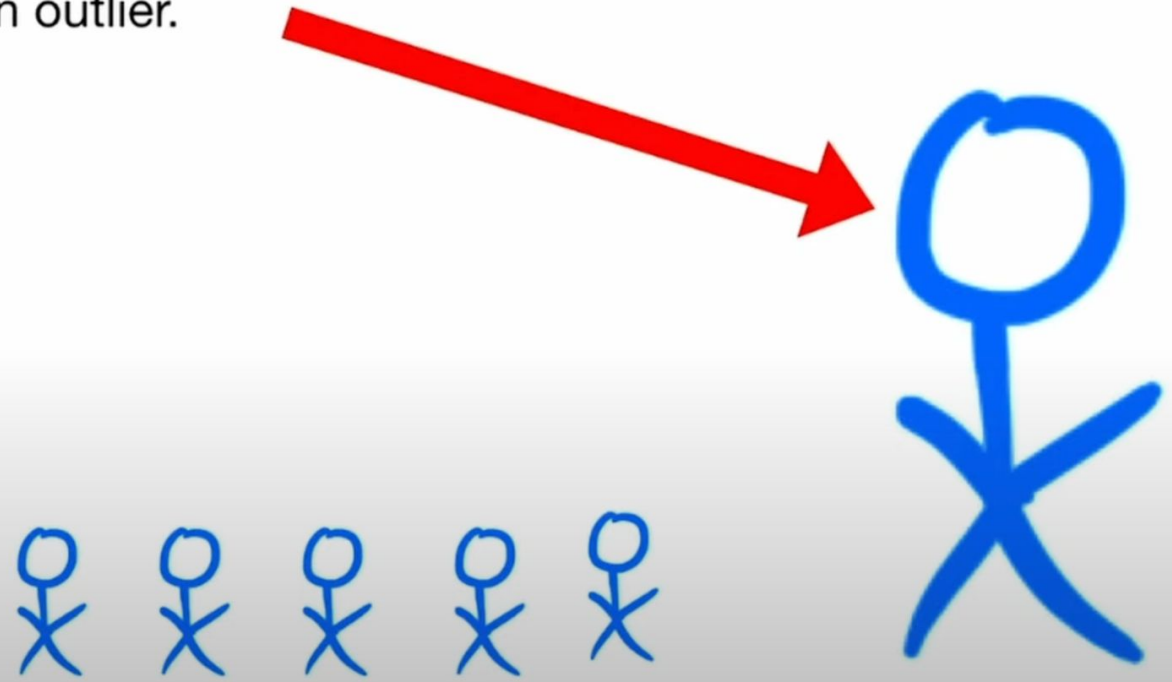
“Everyone in McKinney is dead.”

- Sometimes outliers are clearly just mistakes.
- We could just remove McKinney from the data.
- We could use contextual clues to guess whether they meant to type 101 or 105 or 110.
- We could impute a new value by using nearby values.



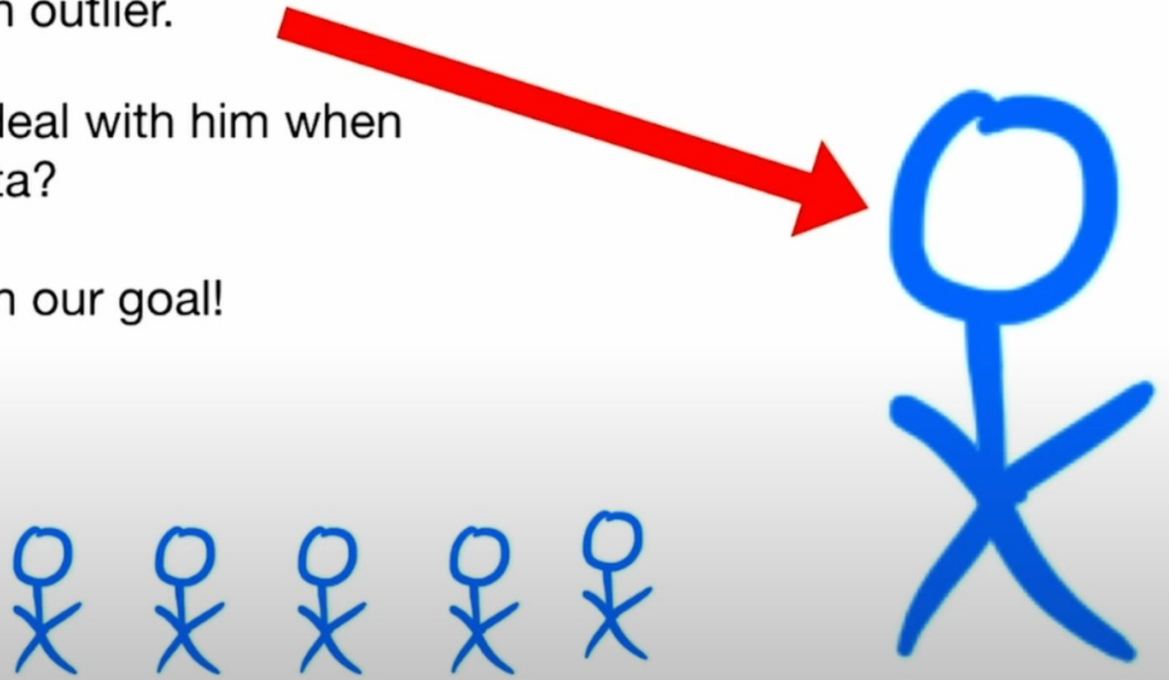
Sometimes outliers are real.

- This guy is an outlier.



If we want to understand the experience of a typical person when they play basketball, we can probably ignore him.

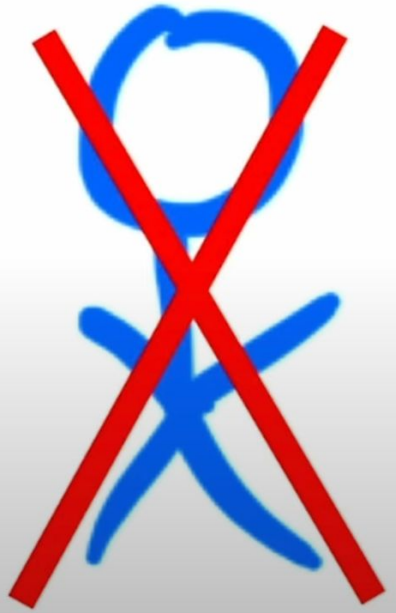
- This guy is an outlier.
- How do we deal with him when analyzing data?
- It depends on our goal!



If we want to understand the experience of a typical person when they play basketball, we can probably ignore him.

(Remove the outlier.)

- This guy is an outlier.
- How do we deal with him when analyzing data?
- It depends on our goal!

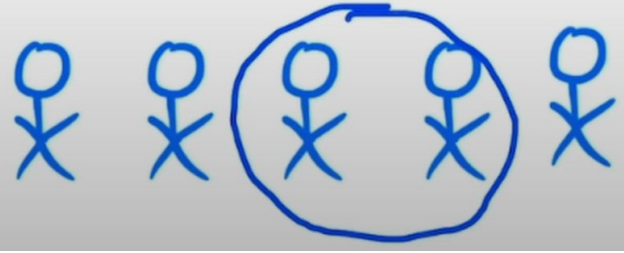


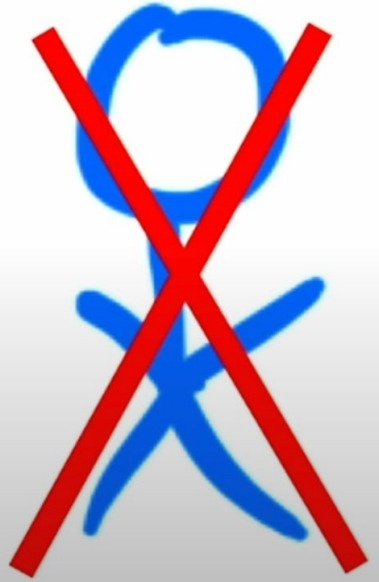
Almost equivalent to removing him would be just using a method that is ***robust*** (resistant) to outliers, like the median.

- This guy is an outlier.
- How do we deal with him when analyzing data?
- It depends on our goal!

Almost equivalent to removing him would be just using a method that is ***robust*** (resistant) to outliers, like the median.

- This guy is an outlier.
- How do we deal with him when analyzing data?
- It depends on our goal!

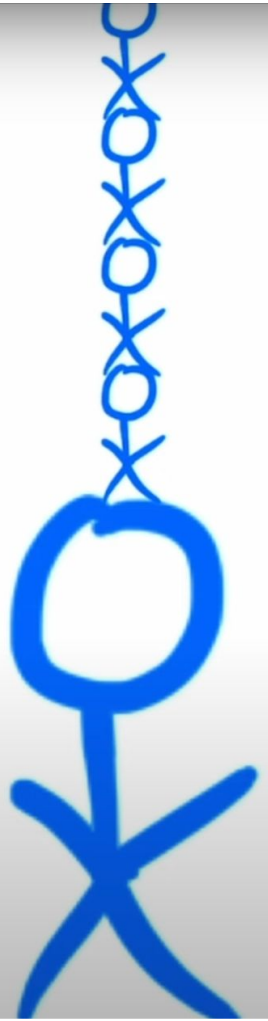




If we want to understand how many people we need to build a tall tower, then we definitely should not ignore the "outlier."

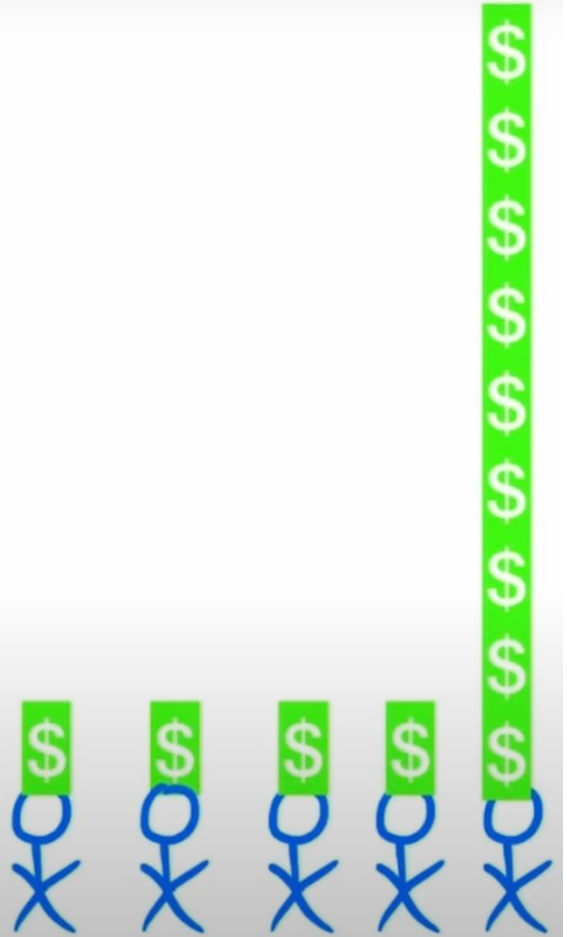
- This guy is an outlier.
- How do we deal with him when analyzing data?
- It depends on our goal!

The mean is a good measure, ***even with outliers***, when what we really care about is the ***total***.



Should we remove Jeff Bezos when considering the average income?

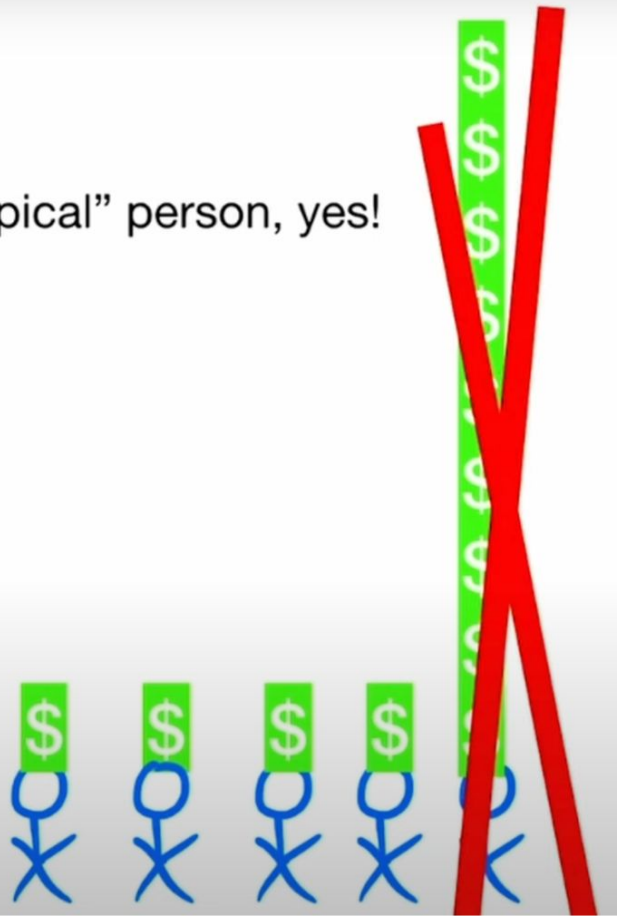
The mean is a good measure, ***even with outliers***, when what we really care about is the ***total***.



Should we remove Jeff Bezos when considering the average income?

- If we want to consider the financial experience of a “typical” person, yes!
- Removing the outlier would solve this.

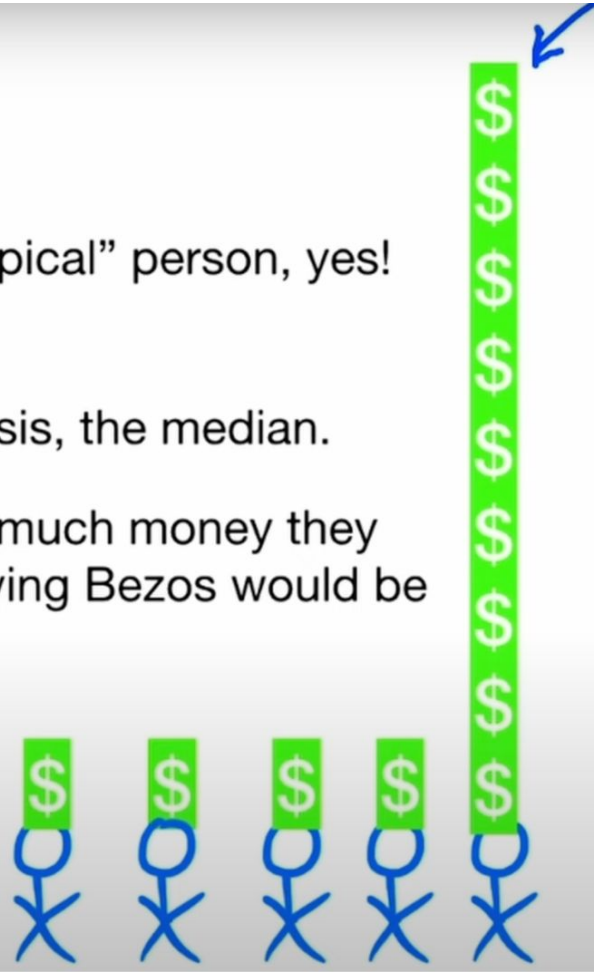
The mean is a good measure, *even with outliers*, when what we really care about is the *total*.



Should we remove Jeff Bezos when considering the average income?

- If we want to consider the financial experience of a “typical” person, yes!
- Removing the outlier would solve this.
- Equivalently, we could use a resistant method of analysis, the median.
- However, if the government wants to understand how much money they could possibly get from a proposed income tax, removing Bezos would be counterproductive!

The mean is a good measure, *even with outliers*, when what we really care about is the *total*.



Should we remove Jeff Bezos when considering the average income?

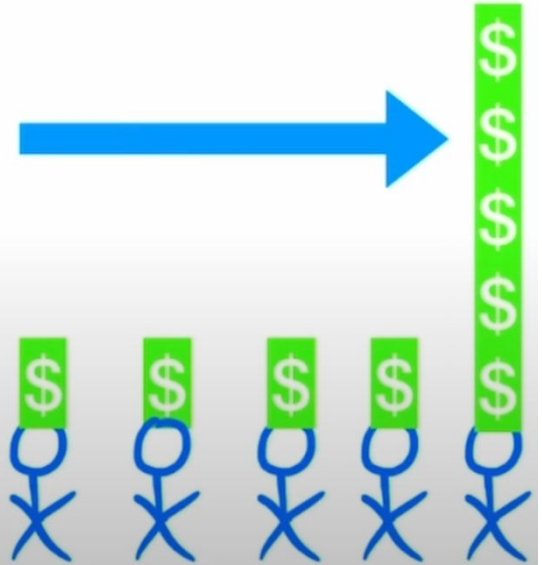
- Removing an outlier has other counter-productive effects.
- If we remove Jeff, we lost all his other data too!

If we want to learn about the relationship between personal characteristics (personality, education, height, etc) and income, then removing Jeff Bezos just because he's an outlier in one respect might make us lose valuable information on the relationship between variables!

Should we remove Jeff Bezos when considering the average income?

- Removing an outlier has other counter-productive effects.
- If we remove Jeff, we lost all his other data too!

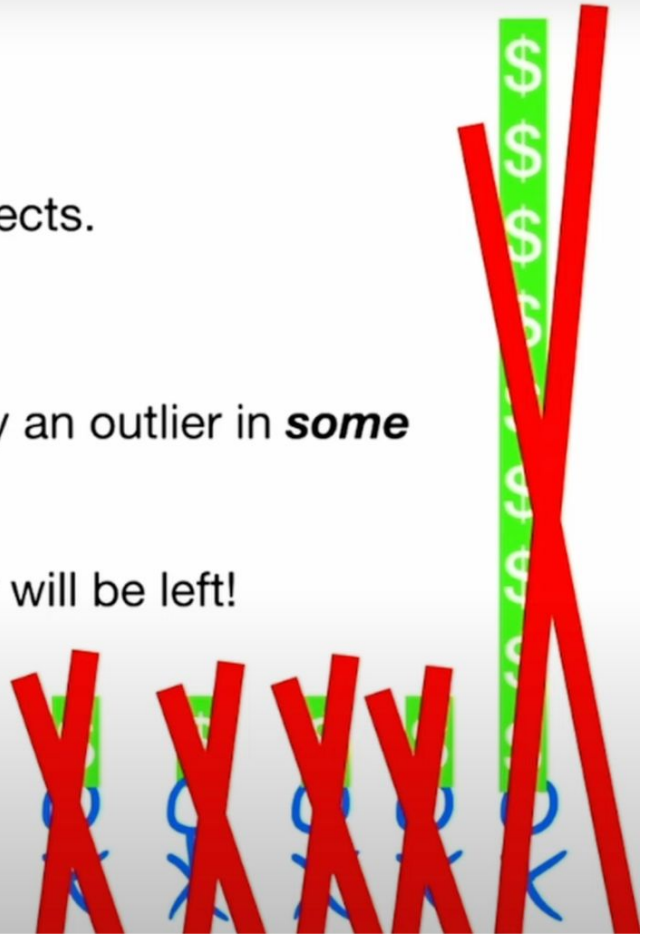
The implication of Winsorization is that, after a certain point, the extra income doesn't really matter. Is that true?



One alternative to removing outliers is called *Winsorization* which is a compromise where we keep outliers but make them a little less outlier-y by capping their value.

Should we remove Jeff Bezos when considering the average income?

- Removing an outlier has other counter-productive effects.
- If we remove Jeff, we lost all his other data too!
- If we have enough variables, each subject is probably an outlier in ***some*** way.
- If we remove outliers for every variable, soon nobody will be left!
- You really should have a good ***reason*** for removing outliers.



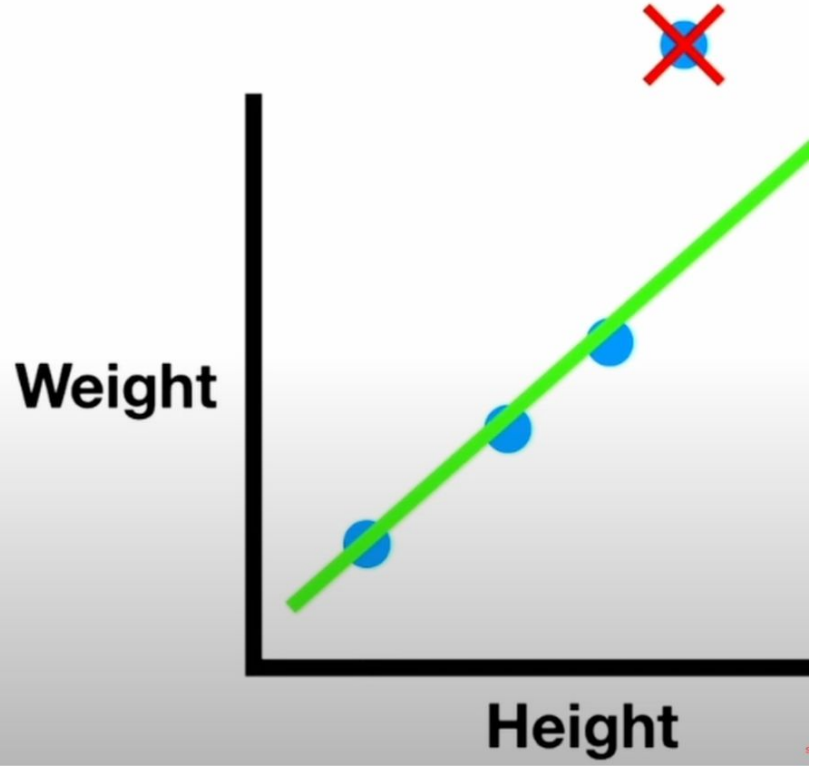
Should we remove Jeff Bezos when considering the average income?

- Removing outliers arbitrarily results in ***overconfident and misleading analysis.***

"Everybody makes the same amount of money."

- Removing outliers arbitrarily results in ***overconfident and misleading analysis.***

"I can predict weight perfectly from height."



Methods for handling outliers

1. **Statistical Methods**
   - Z-Score Method: Calculates how many standard deviations a data point is from the mean. Typically, a Z-score greater than 3 or less than -3 indicates an outlier.
   - Interquartile Range (IQR): Identifies outliers by calculating the range between the first (Q1) and third quartiles (Q3).

     Any value below $Q1 - 1.5 \times IQR$ or

     above $Q3 + 1.5 \times IQR$ is considered an outlier.

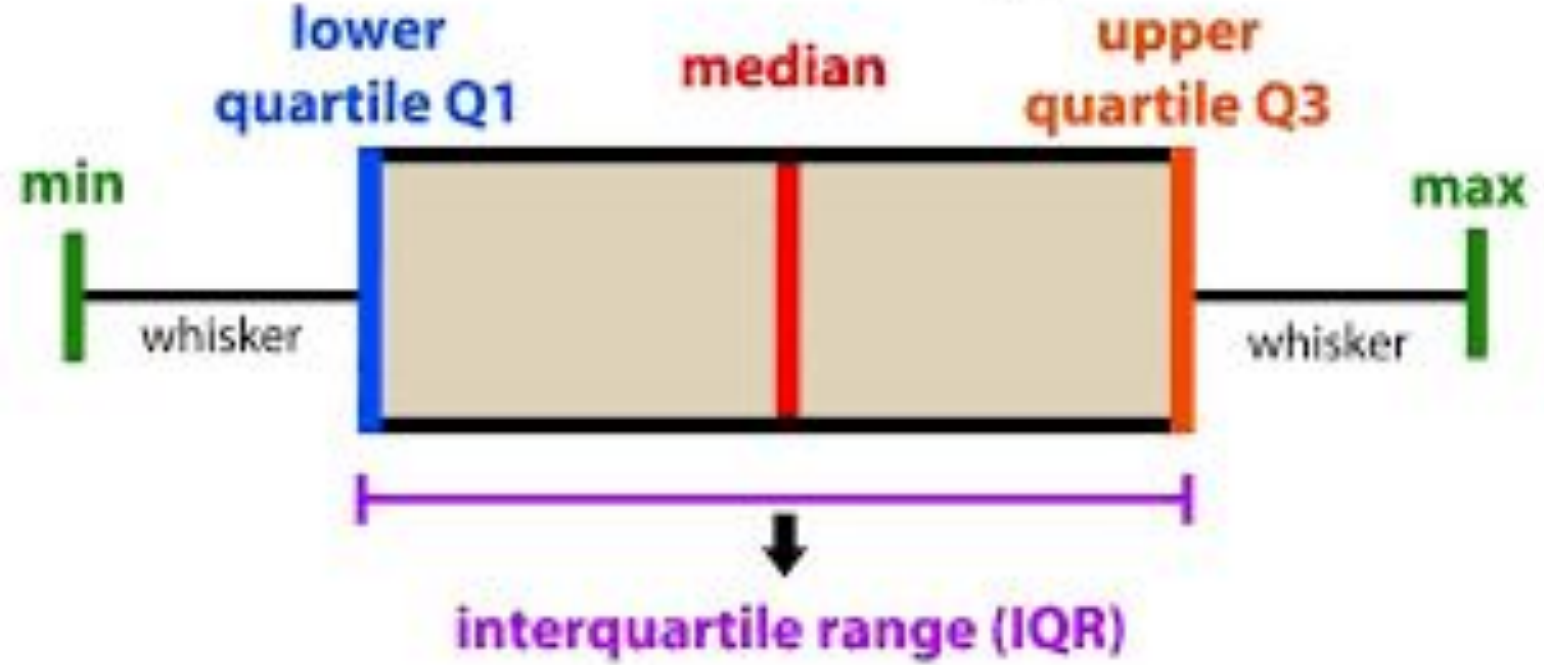
   - Percentile Method: Sets bounds using percentiles (e.g., 5th and 95th percentiles) to identify outliers outside these limits.

Methods for handling outliers

2. **Visualization Technique**

   - Box Plots: Graphically display the distribution of data and highlight potential outliers as points outside the "whiskers."
   - Scatter Plots: Help visualize relationships between variables and identify points that deviate significantly from others.

introduction to data analysis: Box Plot



Methods for handling outliers

3. **Machine Learning Technique**

- Isolation Forests: An ensemble method specifically designed for anomaly detection that isolates observations by randomly partitioning the data.

- Local Outlier Factor (LOF): Measures the local density deviation of a given data point compared to its neighbors.

Provide Necessary Packages and importing the data

```
import numpy as np
import pandas as pd
data = {
    'ID':[1,2,3,4,5],
    'Name':['John','Jane','Mike','Sara','Tom'],
    'Age':[25,30,35,40,45],
    'Gender': ['Male','Male','Female','Female','Male'],
    'Location':['New York','Los Angeles','Chicago','Houston','Miami']
}
df = pd.DataFrame(data)
print(df)
```

Using Label Encoder

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['Gender'] = le.fit_transform(df['Gender'])

print(df)
```

Example: Importing Necessary Libraries

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

Create dataset with outlier and dataframe

```
# Create a sample dataset with outliers

data = np.array([10, 12, 12, 13, 12, 14, 15, 16, 100, 200])  # Notice the outliers at 100 and 200


# Convert to DataFrame for easier manipulation

df = pd.DataFrame(data, columns=['Value'])


# Display original data

print("Original Data:")

print(df)
```

Create a function to detect outliers

```
# Function to detect outliers using IQR
def detect_outliers_iqr(data):
    Q1 = data['Value'].quantile(0.25)
    Q3 = data['Value'].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return (data['Value'] < lower_bound) | (data['Value'] > upper_bound)
```

Detect and print outliers

```
# Detect outliers
outliers = detect_outliers_iqr(df)


# Print detected outliers
print("\nDetected Outliers:")
print(df[outliers])
```

Visualize the outlier using scatter plot

```
plt.figure(figsize=(15, 6))

plt.scatter(range(len(df)), df['Value'], c=['blue' if not x else 'red' for x in outliers], label='Data Points')

plt.axhline(y=df['Value'].mean(), color='green', linestyle='--', label='Mean')

plt.title('Dataset with Outliers Highlighted (Scatter Plot)')

plt.xlabel('Index')

plt.ylabel('Value')

plt.legend()

plt.show()
```

Visualize using box plot

```
sns.boxplot(x=df['Value'])

plt.title('Dataset with Outliers (Box Plot)')

plt.show()
```

Normalization

Normalization refers to the process of scaling individual samples to have unit norm.

The most common method is Min-Max Scaling, which transforms features to a specified range, typically [0, 1].

$X_{norm} = (X - X_{min}) / (X_{max} - X_{min})$ are the minimum and maximum values in the feature, respectively. This method is best used when the distribution is not Gaussian or when the standard deviation is very small.

Standardization

Standardization transforms data to have a mean of 0 and a standard deviation of 1.

This technique is also known as Z-score normalization.

Formula:

$z = (x - \mu) / \sigma$.

Dependent and Independent Variables

An independent variable stands alone. The value does not change due to the impact of any other variable. The researcher manipulates or changes the independent variable to measure its impact on other variables.

Independent variables, in some cases, can already exist, like age, but it is not dependent on any other variable. The variable plays a significant role in research by assisting you in examining causal relationships and testing hypotheses.

Dependent and Independent Variables

A dependent variable (DV) as the name suggests, depends on other variables. It is the variable that is being tested in the experiment. A researcher measures the outcome of the experiment to see how other variables cause changes in the value of a dependent variable.

The causal relationship between dependent & independent variables in research studies helps you assess the effects, associations, and correlations.

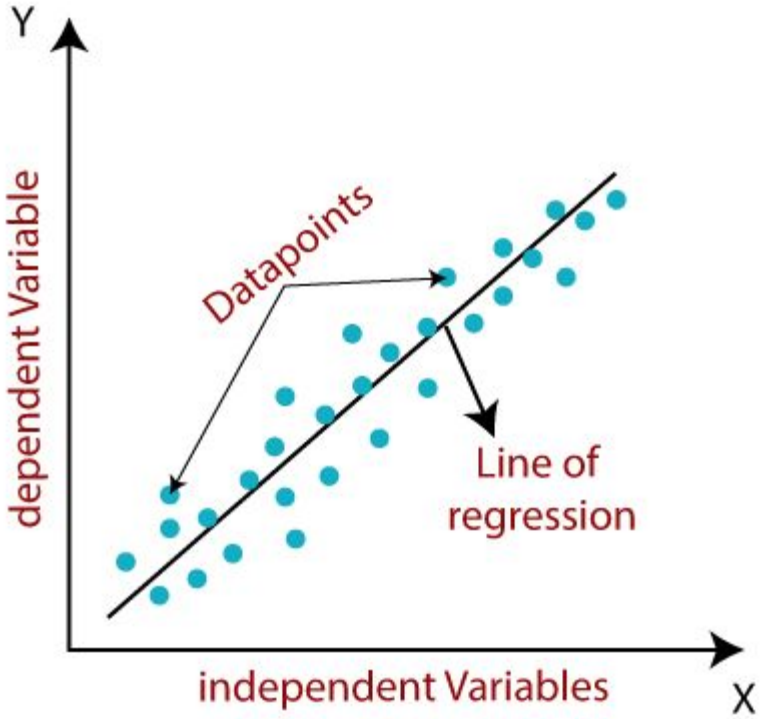
Linear Regression in Machine Learning

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Linear Regression (cont.)

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:



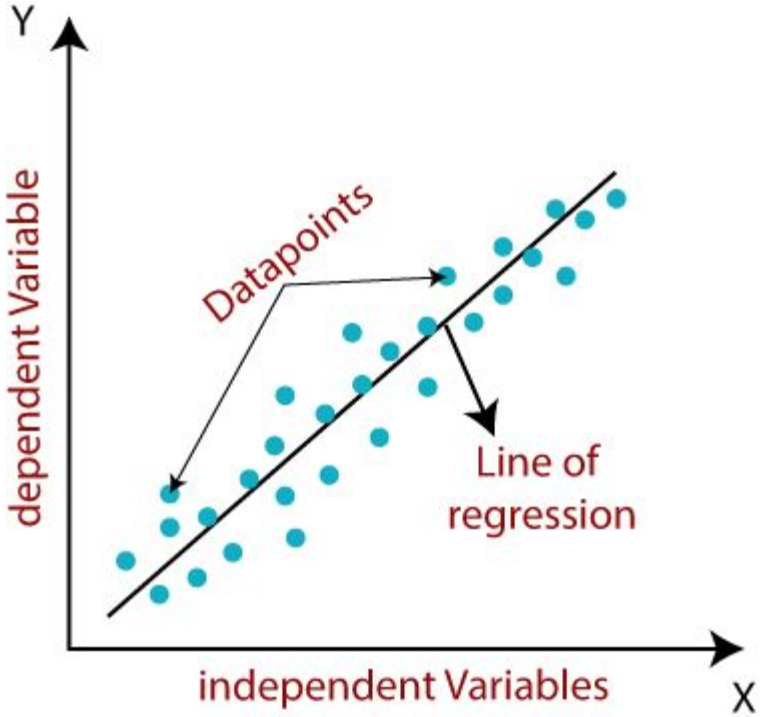
Linear Regression (cont.)

Mathematically, we can represent linear regression as:

$$y= a_0+a_1x+ \varepsilon$$

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.



Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- **Simple Linear Regression:**
  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- **Multiple Linear regression:**
  If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Finding the best fit-line

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

Cost function-

- The different values for weights or coefficient of lines (a_0, a_1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
- Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
- We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**.

For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

$$MSE = 1\frac{1}{N}\sum_{i=1}^{n}(y_i - (a_1x_i + a_0))^2$$

Where,

N=Total number of observation
Yi = Actual value
$(a1x_i+a_0)$= Predicted value.

Problem Deninition:

Find a quadratic regression model for the following data:

X	Y
1	1
2	2
3	1.3
4	3.75
5	2.25

Solution:

Let the simple linear regression model be

$$y = a + bx$$

Steps to find **a** and **b,**

First, find the mean and covariance.

Means of x and y are given by,

$$\bar{x} = \frac{1}{n}\sum x_i$$

$$\bar{y} = \frac{1}{n}\sum y_i$$

The variance of x is given by,

$$\mathrm{Var}\,(x) = \frac{1}{n-1}\sum(x_i - \bar{x}_i)^2$$

The covariance of x and y, denoted by Cov(x, y)is defined as,

$$\mathrm{Cov}(x, y) = \frac{1}{n-1}\sum(x_i - \bar{x})(y_i - \bar{y})$$

Now the values of a and b can be computed using the following formulas:

$$b = \frac{\mathrm{Cov}\,(x, y)}{\mathrm{Var}\,(x)}$$

$$a = \bar{y} - b\bar{x}$$

First, find the mean of x and y,

$$
\begin{aligned}
n &= 5 \\
\bar{x} &= \frac{1}{5}(1.0 + 2.0 + 3.0 + 4.0 + 5.0) \\
&= 3.0 \\
\bar{y} &= \frac{1}{5}(1.00 + 2.00 + 1.30 + 3.75 + 2.25) \\
&= 2.06
\end{aligned}
$$

Next, find the Covariance between x and y,

$$
\operatorname{Cov}(x, y) = \frac{1}{n-1} \sum (x_i - \bar{x})(y_i - \bar{y})
$$

$$
\begin{aligned}
\operatorname{Cov}(x, y) &= \frac{1}{4}[(1.0 - 3.0)(1.00 - 2.06) + \cdots + (5.0 - 3.0)(2.25 - 2.06)] \\
&= 1.0625
\end{aligned}
$$

Now find the variance of x,

$$\text{Var}(x) = \frac{1}{n-1}\sum(x_i - \bar{x}_i)^2$$

$$\text{Var}(x) = \frac{1}{4}[(1.0-3.0)^2 + \cdots + (5.0-3.0)^2]$$
$$= 2.5$$

Now, find the intercept and coefficients,

$$b = \frac{\text{Cov}(x,y)}{\text{Var}(x)}$$

$$a = \bar{y} - b\bar{x}$$

$$b = \frac{1.0625}{2.5}$$
$$= 0.425$$

$$a = 2.06 - 0.425 \times 3.0$$
$$= 0.785$$

Therefore, the linear regression model for the data is,

$$y = 0.785 + 0.425\,x$$