# Data Analytics Pipeline:

Dataset: Titanic: Machine Learning from Disaster

1. Data Gathering:

Description: The data may have been gathered through the collection of passenger information during the Titanic disaster.

Tools: Online forms, manual data entry, or automated systems for data collection. Tools like web scraping or APIs could be used if the data was obtained from various online sources.

2. Data Ingestion:

Description: Raw data from various sources, including CSV files or databases, is ingested into a central repository for further analysis.

Tools: Python libraries like Pandas for reading CSV files, or SQL for extracting data from databases.

3. Data Processing:

Description: The raw data is cleaned by handling missing values, dealing with outliers, and converting data types for uniformity.

Tools: Pandas for data cleaning and transformation, and tools like OpenRefine or Trifacta for more advanced data wrangling.

4. Data Storage:

Description: Cleaned data is stored in a structured format, such as a relational database, for efficient retrieval.

Tools: MySQL, PostgreSQL, or SQLite for relational databases, or data lakes like AWS S3 for larger datasets.

5. Data Analysis:

Description: Statistical analysis and machine learning techniques are applied to understand patterns, correlations, and predict outcomes related to survival on the Titanic.

Tools: Scikit-Learn, TensorFlow, or PyTorch for machine learning, and statistical tools like R or Python libraries (e.g., NumPy and SciPy) for analysis.

6. Data Exploration:

Description: Exploratory Data Analysis (EDA) is performed to gain insights into the distribution and relationships within the dataset.

Tools: Python libraries such as Matplotlib and Seaborn for visualization, and Jupyter Notebooks for interactive analysis.

7. Data Visualization:

Description: Results of the analysis are presented through visualizations, making it easier for stakeholders to understand the insights.

Tools: Tableau, Matplotlib, Seaborn, or Plotly for creating charts and graphs.

8. Data Communication:

Description: Findings and insights are communicated to stakeholders through reports or presentations.

Tools: Microsoft PowerPoint, Google Slides, or Jupyter Notebooks for creating reports.

9. Data Preservation:

Description: Cleaned and analyzed data is stored for future reference and potential reuse.

Tools: Version control systems like Git for tracking changes, and cloud storage services for preserving datasets.

10. Data Maintenance and Monitoring:

Description: Regular checks are performed to ensure data quality, and the analytics pipeline is monitored for any issues.

Tools: Data monitoring tools, custom scripts, and alerts for anomaly detection.

11. Data Security and Compliance:

Description: Measures are taken to secure sensitive passenger information and ensure compliance with data protection regulations.

Tools: Encryption methods for data security, and compliance tracking tools to adhere to regulations.

12. Optimization and Iteration:

Description: The analytics pipeline is iteratively improved based on feedback and changing requirements.

Tools: Continuous Integration/Continuous Deployment (CI/CD) tools for automated pipeline updates, and regular reviews for optimization.

This example outlines how the Generic Data Analytics Pipeline can be applied to a Kaggle dataset, with hypothetical tools and activities at each stage. Keep in mind that the actual tools and activities may vary based on the specific dataset and analytical goals.

# Data types:

Certainly! Here's a consolidated summary with reasoning:

A. Number of bicycles owned per household:

Quantitative: Represents a numerical quantity (count of bicycles).

Discrete: The count of bicycles is a whole number and cannot be fractional.

Ratio: The variable has a true zero point, as having zero bicycles means the absence of the quantity.

B. Average time taken to commute each day:

Quantitative: Represents a numerical quantity (time in minutes).

Continuous: Time is a continuous variable as it can take any value within a range.

Interval: The variable has equal intervals between values, but there is no true zero point as time cannot be zero.

C. Mode of transport used to commute on Monday:

Qualitative: Represents categories or qualities rather than numerical values.

Nominal: The mode of transport represents distinct categories without any inherent order.

D. Motor vehicle safety rating (Gold, Silver, Bronze):

Qualitative: Represents categories or qualities.

Ordinal: The safety ratings have a meaningful order, but the differences between them may not be uniform.

# Big Data

The classification of "big data" is often based on the three Vs: volume, velocity, and variety. Let's evaluate each situation:

A. **Viewing data for Netflix subscribers and social media sentiment analysis:**

- **Likely Big Data:**

- **Volume:** Netflix has a large subscriber base, and the data includes information about shows and dates watched. Social media sentiment analysis involves processing vast amounts of text data.

- **Variety:** Combines structured data (Netflix viewing history) and unstructured data (social media sentiment).

B. **Sales data from the four DCU campus restaurants and catering facilities in 2020:**

- **Possibly Not Big Data:**

- **Volume:** The volume of sales data from four restaurants might not be massive.

- **Variety:** It's likely to be structured data.

C. **Download of content and metadata from a personal Twitter account:**

- **Not Big Data:**

- **Volume:** Typically, personal Twitter data is not massive.

- **Variety:** May include a mix of media and text data but likely limited in scope.

D. **Player training data from the Irish Rugby Squad:**

- **Possibly Big Data:**

- **Volume:** Depending on the number of players, sensors, and observations, the volume could be substantial.

- **Variety:** Involves different types of data, including sensor data.

**Explanation:**

- Option A is likely to be classified as big data due to the large volume and variety of data involved.

- Option B might not qualify as big data unless the sales data is exceptionally voluminous or diverse.

- Option C is less likely to be big data as personal Twitter data tends to be relatively small in volume.

- Option D has the potential to be big data, especially if it involves a large number of players and extensive sensor data.

In summary, the most likely situation to be classified as big data is option A, primarily due to the significant volume and variety of data involved in Netflix viewing and social media sentiment analysis.

# Data Collection Task

Certainly! Let's provide a more integrated response by incorporating the questions and answers:

### Step 1: Understand the Requirements

1. **Read the Brief Thoroughly:**

   - Understand the context, goals, and objectives outlined in the brief.

2. **Identify Key Stakeholders:**

   - Determine who the key stakeholders are and what their specific needs and concerns might be.

3. **Define Scope and Constraints:**

   - Clarify the scope of the project, including timeframes, geographical areas, and any legal or ethical constraints.

### Step 2: Formulate Questions for the Client

   - Engage with the client to gather more information and clarify uncertainties. Ask questions such as:

     - *What specific insights are they looking for?*

     - *Who are the end-users of the collected data?*

     - *Are there any legal or ethical considerations?*

   **Client's Responses:**

   - *Insights sought include the impact of COVID-19 restrictions on working and commuting behavior in Ireland during 2020.*

   - *End-users include various government departments.*

   - *Legal and ethical considerations include data privacy and compliance with regulations.*

### Step 3: Identify Data Sources and Formats

1. **Examine Available Data:**

   - Explore the types of data available, such as pedestrian footfall, traffic monitoring, surveys, weather, and economic indicators.

   **Data Types Identified:**

   - *Pedestrian footfall data (structured time-series data).*

   - *Traffic monitoring camera records (time-stamped vehicle count data).*

   - *Survey data on working from home (structured survey responses).*

   - *Weather and economic information (time-series data).*

2. **Consider Data Structure:**

   - Understand the structure of each type of data and the relationships between them.

3. **Choose Appropriate File Formats:**

   - Consider the nature of the data when choosing file formats, ensuring compatibility and ease of integration.

### Step 4: Propose a Database Storage Approach

1. **Understand Database Options:**

   - Consider different types of databases, such as relational databases, NoSQL databases, or a combination.

   **Database Storage Approach Proposed:**

   - *Relational Database Management System (RDBMS), specifically PostgreSQL.*

2. **Consider Data Relationships:**

   - Evaluate how different datasets relate to each other and whether a relational database would be beneficial.

3. **Think About Query Requirements:**

   - Consider the types of queries that will be performed, and assess which database system best supports those requirements.

4. **Security and Compliance:**

   - Address data security and compliance requirements to protect sensitive information.

   **Reasoning for Database Storage Approach:**

   - *RDBMS is chosen for its structured organization, support for complex queries, fine-grained access control, and compliance with data security and privacy regulations.*

### Step 5: Provide a Reasoned Recommendation

   - Clearly articulate why you've chosen a particular database storage approach, demonstrating how it aligns with the project requirements, data characteristics, and the ongoing needs of government departments.

### Step 6: Consider Scalability

   - Anticipate future needs and ensure that the proposed system is scalable as the project evolves.

### Step 7: Address Privacy and Ethics

   - Highlight considerations for data privacy and ethical practices, especially when conducting consumer surveys.

### Step 8: Think Long-Term

   - Consider the long-term use of the system, as it needs to support ongoing monitoring by various government departments.

By systematically addressing each aspect of the question and incorporating the answers into the process, you can present a comprehensive and well-reasoned approach to designing the data collection system.

# Metadata

**(i) Simple Example Metadata for a Mobile Phone:**

1. **Model:** iPhone 13 Pro

2. **Operating System:** iOS 15

3. **Storage Capacity:** 256 GB

4. **Color:** Graphite

5. **Processor:** A15 Bionic chip with Neural Engine

**(ii) Metadata Classification:**

1. **Model:** Descriptive metadata - It provides information about the nature, scope, or content of the resource.

2. **Operating System:** Descriptive metadata - It describes the nature of the resource.

3. **Storage Capacity:** Descriptive metadata - It provides quantitative information about the resource.

4. **Color:** Descriptive metadata - It describes the visual characteristics of the resource.

5. **Processor:** Descriptive metadata - It provides information about the resource's characteristics.

6. **Metadata Administrative Elements:**

- **Date of Purchase:** Administrative metadata - It includes information about the creation, maintenance, and use of the resource.

- **Device ID/Serial Number:** Administrative metadata - It provides a unique identifier for the resource.

- **Owner/User Information:** Administrative metadata - It includes details about the owner or primary user of the device.

7. **Metadata Structural Elements:**

- **Folder Structure (if applicable):** Structural metadata - It provides information about the organization and structure of the resource.

- **File Naming Conventions:** Structural metadata - It defines how files are named and organized within the resource.

(iii) Impact of Using a Standard on Metadata Quality with Reference:

Quality Improvement:

Consistency: Using a metadata standard enhances the consistency of metadata across different electronic devices. This consistency ensures that information is described and organized uniformly.

Interoperability: To standards

Searchability: Metadata standards contribute to better searchability. When metadata follows a standardized format, it becomes easier to search and retrieve relevant information about electronic devices.

Challenges:

Achieving Universal Compliance:

Challenge: Ensuring that all electronic devices and data contributors adhere to a specific metadata standard may be challenging.

Data Contributor Compliance:

Challenge: Not all data contributors may be familiar with or willing to adopt the prescribed metadata standard.

Addressing Challenges:

Communication Clear:

To overcome these challenges, clear communication is crucial. Clear communication about the importance of adhering to metadata standards ensures understanding among contributors.

Proper Training:

Providing proper training to data contributors helps in familiarizing them with the standards and ensures that they can effectively implement them.

Ongoing Support:

Ongoing support is necessary to address queries, challenges, or updates related to the metadata standard. This support ensures continued adherence to the standard.

Regular Checks and Validation:

Regular checks and validation mechanisms should be in place to verify whether the metadata provided aligns with the prescribed standard. This helps maintain the quality and consistency of metadata over time.

**Descriptive**

- What the information object is about, intrinsic properties. Helps to understand its nature without interacting with the content directly.
    - Title
    - Author/creater
    - Date of creation
    - Abstract/Summary
    - Genre
    - Spatial coverage ( Areas covered)
    - File format

**Administrative:**

- Who, what, why, where of the object's creation and management. Intrinsic properties
    - Ownership
    - Storage location
    - Similar ex like above

**Structural**

- Information about the structure, format, and composition of the thing described; can be intrinsic and extrensic. Internal structural, format,
    - Heirarchy - chapters, sections or nested components
    - Sequence
    - Logical Relationships
    - Page number
    - Table of contents
    - Navigation info
    - Data model

# Metadata Granuality

It can apply to different dimensions or concepts
**Abstraction -** How close the metadata is to the data or object
**Granuality:** How detailed the metadata is

**Abstraction:**
The concept of abstraction in metadata, as illustrated in the example of a novel, provides a structured hierarchy that helps in understanding and managing information about creative works. Let's delve deeper into the importance of abstraction levels in metadata:

Understanding Metadata Levels:

Abstraction levels help in distinguishing different aspects of metadata, from the conceptual creation (work) to its various expressions, manifestations, and individual items.
This hierarchical structure clarifies the relationship between the abstract intellectual or artistic concept and its tangible representations.
Assessing Value and Usability:

Different abstraction levels cater to distinct needs. For example, when assessing the value and usability of metadata, understanding whether the information pertains to the work as a whole, a specific expression, a manifestation, or an individual item is crucial.
This differentiation becomes particularly important when managing variations, such as different editions or formats of a work.
Complex Objects and Digital Representations:

Abstraction is valuable when dealing with complex objects, especially in the digital realm. For instance, a digital representation of a physical object may involve multiple layers of metadata, from the overall work to specific manifestations or instances.
It accommodates the intricacies of digital objects and their relationships to physical counterparts.
Consistency in Description:

By structuring metadata with varying levels of abstraction, you ensure consistency in describing and organizing information. This consistency aids in effective cataloging, retrieval, and understanding of the data.
Challenges in Interpretation:

Abstraction levels also introduce challenges in interpretation. Users need to be clear about which level of the hierarchy the metadata describes, especially when dealing with manifestations or specific items.
Questions about the "artist" in the context of a photograph of a sculpture highlight the need for precision in describing the level at which metadata applies.
In conclusion, the thoughtful consideration of abstraction levels in metadata is essential for organizing and interpreting information about creative works. It enhances clarity, supports effective data management, and ensures that metadata is applied appropriately to the specific context within the hierarchy.

**Granuality:**

The concept of granularity in metadata descriptions is essential for understanding the level of detail and complexity in the information being captured. The granularity hierarchy you've described, moving from low-level features to high-level semantics and human-recorded information, is a valuable framework. Let's explore each layer of the hierarchy:

Low-Level Features:

Description: Very detailed features derived directly from the digital representation, such as pixel values in a digital image.
Example: Calculated color histograms, pixel intensity values, or spatial frequencies.
Structure Features:

Description: Slightly more complex structural features derived from low-level values, providing information about the arrangement or organization of the data.
Example: Geometric properties, edge detection results, or texture patterns in a digital image.
Semantics:

Description: Represents the meaning or semantics encoded in the digital object, bridging the gap between low-level features and human-understandable information.

Example: Object recognition, scene interpretation, or context-specific interpretations of the digital content.
Semantic Gap:

Description: Refers to the challenge of automatically creating metadata that describes the meaning of a digital object without significant human effort.
Example: While low-level features like pixel values can be automatically extracted, understanding the semantic context or interpreting the content often requires human intervention.
Human-Recorded Information:

Description: Information that generally needs to be recorded by a human, encompassing bibliographic or management data about the creation and instance of the digital object.
Example: Creator details, creation date, copyright information, or any metadata that requires human judgment and context.
Significance of Granularity:

Precision and Detail:

Granularity levels allow for a balance between precision and detail. Low-level features offer fine-grained details, while high-level semantics provide a more abstract understanding.
Automation vs. Human Intervention:

The hierarchy acknowledges the challenge of the semantic gap, emphasizing the need for human intervention in capturing meaning. It recognizes that certain layers of metadata may require automated extraction, while others benefit from human interpretation.
Usability and Context:

Granularity considerations ensure that metadata descriptions align with the usability and context of the digital object. Human-recorded information, for instance, provides context that may be challenging to automate.
Comprehensive Understanding:

By encompassing a range of granularity levels, the hierarchy allows for a more comprehensive understanding of the digital object, catering to both machine-driven analysis and human interpretation.
In conclusion, the granularity hierarchy in metadata descriptions provides a structured approach to capturing and organizing information about digital objects, considering levels of detail, complexity, and the role of human input in meaningful interpretation.

**Movie**:
**Abstraction**:

Work:
Description: The abstract entity, the distinct intellectual or artistic creation with no single material manifestation.
Example: "Inception" by Christopher Nolan

Expression:
Description: The multiple realizations of the work in some particular medium or notation, where it can actually be perceived.

Example: Movie version, DVD release, Streaming service version

Manifestation:
Description: Each of the formats of an expression that has the same appearance, but not necessarily the same implementation.
Example: Theatrical release, Blu-ray edition, Digital download

Item:
Description: A single exemplar of a manifestation, representing a unique copy or instance.
Example: Your personal Blu-ray copy, a specific digital download

**Granularity**:
Low-Level Features:
Example: Pixel values, color histograms, audio waveforms from individual frames and scenes.

Structure Features:
Example: Geometric properties of objects within frames, scene transitions, audio track structure.

Semantics:
Example: Object recognition (characters, locations), genre identification, sentiment analysis of dialogue.

Semantic Gap:
Example: While automated tools can identify scenes, recognizing the emotional tone or the underlying cultural references may require human interpretation.

Human-Recorded Information:
Example: Director information, cast details, genre categorization, plot summaries, and audience ratings. These details provide context and interpretation.

Justification:
Precision and Detail: Low-level features offer technical precision, capturing detailed information about the visual and audio components. Human-recorded information adds detail and context, providing a comprehensive understanding.

Automation vs. Human Intervention: While low-level features can be automatically extracted, semantic understanding and human-recorded information often involve human interpretation, especially in recognizing the artistic and emotional aspects of the movie.

Usability and Context: Human-recorded information enhances usability by providing context that is crucial for audiences seeking to understand the movie on both technical and interpretive levels.

Comprehensive Understanding: The combination of abstraction and granularity provides a comprehensive understanding of the movie, catering to both technical analysis and broader interpretation for diverse audiences.

# Web Scraping

**Scraping Data from a Website:**

Scraping data from a website involves a systematic process to extract information. Here's a concise description of the steps involved:

1. **Have a Plan:**

   - Before starting, plan how to identify the specific data items on the web page you want to scrape. This includes understanding the HTML structure and the location of the desired information.

2. **Request Webpage:**

   - Use methods like `urlopen` or `requests` to send an HTTP request to the website and retrieve the HTML content of the page.

3. **Parse HTML:**

   - Utilize HTML parsing libraries such as `lxml` or `beautifulsoup` to navigate through the HTML structure and extract the desired data elements. This involves locating the HTML tags, classes, or identifiers associated with the data.

4. **Store Data:**

   - Store the extracted data in a structured format, such as a list or dictionary, for further processing and analysis.

5. **Format as Required:**

   - Depending on your needs, format the collected data into the desired output format, such as CSV, JSON, a DataFrame, or SQL, making it ready for analysis or storage.

**Rules for Web Scraping:**

1. **Check Terms and Conditions:**

   - Before initiating any scraping activity, carefully review the terms and conditions of the website. It's essential to respect the website owner's rules and policies regarding data usage. If the terms explicitly prohibit scraping, it's crucial to seek permission or find alternative data sources.

2. **Be Considerate and Cautious:**

- Exercise caution and be considerate of the website's server capacity. A computer can send web requests much faster than a human user can. To avoid overloading the server and potentially causing a denial of service attack, space out your requests. Respect the website's bandwidth and processing capabilities to maintain a positive relationship with the server.

**Additional Considerations:**

- **Adapt to Changes:**

- Web scraping is susceptible to changes in website layouts. Websites may update their design or structure, leading to the need for code adjustments. Be prepared to adapt your scraping code if the website undergoes changes.

- **Handle Inconsistencies:**

- Web pages may have inconsistencies, requiring manual clean-up even after data extraction. Account for variations or unexpected elements in the HTML structure and be ready to implement additional cleaning steps in your code.

In summary, while web scraping is a powerful tool for data extraction, it should be conducted responsibly by adhering to ethical guidelines, respecting website terms, and being adaptable to changes in website structure.

# Errors or artefacts

Approach for Identifying Errors or Artifacts:

Understand the Dataset:

Carefully read the dataset description and understand the variables, their meanings, and the overall structure of the data.

Exploratory Data Analysis (EDA):

Use statistical and visual methods to explore the dataset. This can include summary statistics, histograms, scatter plots, and other visualization techniques to identify patterns and potential anomalies.

Data Profiling Tools:

**Formatting Issues:**

- Formatting Is Irregular between Different Tables/Columns
- Extra Whitespace (select for “ABC” or “ABC ”)

- Irregular Capitalization
- Inconsistent Delimiters (commas or tabs)
- Irregular NULL Format (“” or NULL or NA)
- Invalid Characters (ascii or unicode)
- Weird or Incompatible Datetimes (Day-Month or Month-Day, timezones)
- Operating System Incompatibilities (eg, ‘\n’ or ‘\r\n’)
- Wrong Software Versions

**Content Issues:**

- Duplicate Entries
- Multiple Entries for a Single Entity
- Missing Entries
- NULLs  (what do they mean?)
- Huge Outliers
- Out-of-Date Data
- Artificial Entries (eg, €999,999 salary)
- Irregular Spacings (eg, gaps in time series)
- Incorrect or inconsistent units (metres or inches, dollars or euros))

### Formatting Issues:

1. **Irregular Formatting between Different Tables/Columns:**

   - **Likely Introduction Phase:** Gathering - Import

   - **Possible Introduction Cause:** Different tables or columns may have been imported from diverse sources with varying formatting standards.

2. **Extra Whitespace (e.g., "ABC" or "ABC "):**

   - **Likely Introduction Phase:** Processing - Cleaning

   - **Possible Introduction Cause:** Whitespace may be introduced during data entry or import, and cleaning is required to remove extra spaces.

3. **Irregular Capitalization:**

   - **Likely Introduction Phase:** Processing - Cleaning

   - **Possible Introduction Cause:** Capitalization inconsistencies may arise during data entry or import, necessitating cleaning efforts.

4. **Inconsistent Delimiters (commas or tabs):**

   - **Likely Introduction Phase:** Gathering - Import

   - **Possible Introduction Cause:** Importing data with different delimiters can result in inconsistency during the initial data import.

5. **Irregular NULL Format (“” or NULL or NA):**

   - **Likely Introduction Phase:** Processing - Cleaning

   - **Possible Introduction Cause:** Null values may be represented in various formats during data entry or import, requiring standardization.

6. **Invalid Characters (ascii or unicode):**

   - **Likely Introduction Phase:** Gathering - Import

   - **Possible Introduction Cause:** Invalid characters may be introduced during data extraction or import from external sources.

7. **Weird or Incompatible Datetimes (Day-Month or Month-Day, timezones):**

   - **Likely Introduction Phase:** Processing - Cleaning

   - **Possible Introduction Cause:** DateTime inconsistencies may arise due to varied date formats during data entry or import.

8. **Operating System Incompatibilities (e.g., ‘\n’ or ‘\r\n’):**

   - **Likely Introduction Phase:** Gathering - Import

   - **Possible Introduction Cause:** Differences in operating systems during data import can result in line-ending inconsistencies.

9. **Wrong Software Versions:**

   - **Likely Introduction Phase:** Gathering - Import

   - **Possible Introduction Cause:** Data collected or imported using different software versions may lead to compatibility issues.

### Content Issues:

10. **Duplicate Entries:**

    - **Likely Introduction Phase:** Gathering - Capture

    - **Possible Introduction Cause:** Duplicates can occur during the initial data collection phase.

11. **Multiple Entries for a Single Entity:**

- **Likely Introduction Phase:** Gathering - Import

- **Possible Introduction Cause:** Importing data from different sources may lead to multiple entries for the same entity.

12. **Missing Entries:**

- **Likely Introduction Phase:** Gathering - Capture

- **Possible Introduction Cause:** Missing entries can occur during the initial data collection phase.

13. **NULLs (what do they mean?):**

- **Likely Introduction Phase:** Processing - Cleaning

- **Possible Introduction Cause:** Null values may be introduced during data entry or import, and their meaning needs clarification.

14. **Huge Outliers:**

- **Likely Introduction Phase:** Analysing - Statistics

- **Possible Introduction Cause:** Outliers may be identified during statistical analysis.

15. **Out-of-Date Data:**

- **Likely Introduction Phase:** Gathering - Capture

- **Possible Introduction Cause:** Data may become outdated during the initial collection phase.

16. **Artificial Entries (e.g., €999,999 salary):**

- **Likely Introduction Phase:** Gathering - Capture

- **Possible Introduction Cause:** Artificial entries may be introduced during data collection.

17. **Irregular Spacings (e.g., gaps in time series):**

- **Likely Introduction Phase:** Processing - Cleaning

- **Possible Introduction Cause:** Spacing irregularities may arise during data cleaning or aligning.

18. **Incorrect or Inconsistent Units (metres or inches, dollars or euros):**

- **Likely Introduction Phase:** Gathering - Import

- **Possible Introduction Cause:** Unit inconsistencies may be introduced during the initial data import.

### Assumptions:

- The assumption is that data cleaning and processing efforts are part of the generic data analytics pipeline and that inconsistencies may be identified and addressed during these phases.

- The identification of issues assumes that adequate data profiling and exploratory data analysis are conducted during the data analytics pipeline phases.

- The specific causes mentioned are illustrative and can vary based on the context of the dataset and the nature of the data sources.

# Avoid or mitigate above error

### Formatting Issues:

1. **Irregular Formatting between Different Tables/Columns:**

   - **Data Quality Method:** Standardize formatting during the data import process.

   - **Reasoning:** Applying consistent formatting standards during the import phase ensures uniformity across tables and columns, reducing the risk of irregularities.

2. **Extra Whitespace (e.g., "ABC" or "ABC "):**

   - **Data Quality Method:** Implement whitespace removal during data cleaning.

   - **Reasoning:** Removing extra whitespace ensures uniformity in data and prevents issues arising from variations in spacing.

3. **Irregular Capitalization:**

   - **Data Quality Method:** Standardize capitalization during data cleaning.

   - **Reasoning:** Consistent capitalization reduces ambiguity and enhances data consistency, making it easier to analyze and interpret.

4. **Inconsistent Delimiters (commas or tabs):**

- **Data Quality Method:** Validate and standardize delimiters during the data import process.

- **Reasoning:** Ensuring consistent delimiters improves data structure and prevents errors during subsequent processing.

5. **Irregular NULL Format (“” or NULL or NA):**

- **Data Quality Method:** Standardize NULL representations during data cleaning.

- **Reasoning:** Standardizing NULL formats ensures clarity and consistency, reducing confusion about missing or undefined values.

6. **Invalid Characters (ascii or unicode):**

- **Data Quality Method:** Validate and clean invalid characters during data import.

- **Reasoning:** Removing invalid characters ensures data integrity and prevents issues related to encoding discrepancies.

7. **Weird or Incompatible Datetimes (Day-Month or Month-Day, timezones):**

- **Data Quality Method:** Standardize datetime formats during data cleaning.

- **Reasoning:** Consistent datetime formats facilitate accurate temporal analysis and prevent interpretation errors.

8. **Operating System Incompatibilities (e.g., ‘\n’ or ‘\r\n’):**

- **Data Quality Method:** Normalize line endings during data import.

- **Reasoning:** Ensuring uniform line endings prevents compatibility issues across different operating systems.

9. **Wrong Software Versions:**

- **Data Quality Method:** Document and verify software versions during data gathering.

- **Reasoning:** Keeping track of software versions helps ensure compatibility and provides context for potential issues.

### Content Issues:

10. **Duplicate Entries:**

- **Data Quality Method:** Implement deduplication methods during data processing.

- **Reasoning:** Removing duplicates ensures data accuracy and prevents distortions in analysis results.

11. **Multiple Entries for a Single Entity:**

- **Data Quality Method:** Validate and merge entries during data cleaning.

- **Reasoning:** Consolidating multiple entries ensures a single representation for each entity, improving data consistency.

12. **Missing Entries:**

- **Data Quality Method:** Implement imputation techniques or investigate missing data during data cleaning.

- **Reasoning:** Addressing missing entries enhances dataset completeness and reduces potential biases in analysis.

13. **NULLs (what do they mean?):**

- **Data Quality Method:** Clearly define and document NULL representations during data gathering.

- **Reasoning:** Providing clear definitions avoids ambiguity and ensures a standardized understanding of NULL values.

14. **Huge Outliers:**

- **Data Quality Method:** Investigate and validate outliers during statistical analysis.

- **Reasoning:** Identifying and validating outliers helps distinguish genuine data points from errors, enhancing the accuracy of statistical analysis.

15. **Out-of-Date Data:**

- **Data Quality Method:** Regularly update data during the gathering phase.

- **Reasoning:** Keeping data up-to-date ensures its relevance and reliability for analysis purposes.

16. **Artificial Entries (e.g., €999,999 salary):**

- **Data Quality Method:** Investigate and filter out artificial entries during data cleaning.

- **Reasoning:** Detecting and removing artificial entries improves the accuracy of analysis results.

17. **Irregular Spacings (e.g., gaps in time series):**

   - **Data Quality Method:** Validate and interpolate missing values during data cleaning.

   - **Reasoning:** Filling gaps in time series ensures a continuous and consistent dataset for analysis.

18. **Incorrect or Inconsistent Units (metres or inches, dollars or euros):**

   - **Data Quality Method:** Standardize units during data cleaning.

   - **Reasoning:** Ensuring uniform units prevents errors in quantitative analysis and facilitates meaningful comparisons.

### Assumptions:

- These data quality methods assume that data cleaning and preprocessing are integral parts of the data analytics pipeline.

- Documentation and communication are essential to ensure that standardized practices are followed across the data analytics workflow.

- Periodic reviews and updates to data quality processes help adapt to changing requirements and maintain a high standard of data integrity.

# potential personal or Sensitive Data

### Identifying Potential Personal or Sensitive Data:

**(i) Can you identify any potential personal or sensitive data in the provided sample dataset? Why or why not?**

**Example Dataset: "Customer_Sales_Data.csv"**

1. **Review Dataset Columns:**

   - Check columns like "Customer Name," "Address," and "Email" for personally identifiable information.

2. **Look for Identifiers:**

   - Identify columns containing unique identifiers such as "Customer ID" or "Social Security Number."

3. **Sensitive Attributes:**

   - Examine columns like "Health Condition," "Annual Income," or "Credit Card Number" for sensitive information.

4. **Check Data Types:**

   - Look for columns with data types suggesting personal data, such as strings for names or addresses.

5. **Contextual Understanding:**

   - Consider the purpose of the dataset; if it's related to customer transactions, personal or sensitive data may be present.

**Conclusion:**

   - The dataset may contain personal or sensitive data if it includes columns with names, addresses, unique identifiers, or sensitive attributes like health or financial information.

---

### Legally Working with Personal or Sensitive Data:

**(ii) What process should you follow if you want to legally work with personal or sensitive data?**

**Example Process:**

1. **Understand Data Protection Laws:**

   - Research and comply with relevant data protection laws, such as GDPR or local privacy regulations.

2. **Data Classification:**

- Classify data into categories (e.g., personal, sensitive) based on legal definitions and requirements.

3. **Obtain Consent:**

- If required, obtain explicit consent from customers before collecting or processing their personal data.

4. **Implement Security Measures:**

- Employ encryption, access controls, and secure storage to protect personal and sensitive data.

5. **Anonymization and Pseudonymization:**

- Anonymize or pseudonymize data to minimize the risk of identification while preserving analytical utility.

6. **Data Minimization:**

- Only collect and process the data necessary for sales analysis, avoiding unnecessary personal details.

7. **Document Processing Activities:**

- Maintain a data processing record documenting the purpose, legal basis, and procedures for handling personal or sensitive data.

8. **Data Subject Rights:**

- Establish processes to respect and respond to data subject rights, allowing individuals to access or correct their data.

9. **Data Breach Response:**

- Develop and implement a response plan to address any potential data breaches, following legal reporting requirements.

Pseudo code encryption

anonymization

**Conclusion:**

- Legal and ethical handling of personal or sensitive data involves compliance with data protection laws, obtaining consent, implementing security measures, and respecting individuals' rights over their data. Adhering to these practices ensures responsible and lawful data management.

# All graphs and its description and usage

1. Line Chart:

Description: A line chart displays data points connected by straight line segments. It's used to show trends over time or continuous data.

Use Case: Displaying stock prices over a period, tracking temperature changes throughout the year.

2. Bar Chart:

Description: A bar chart represents data using rectangular bars. It's suitable for comparing categories or groups.

Use Case: Comparing sales performance across different regions.

3. Histogram:

Description: A histogram displays the distribution of a single numerical variable. It shows the frequency of data within defined intervals.

Use Case: Analyzing the distribution of exam scores in a class.

4. Pie Chart:

Description: A pie chart shows parts of a whole, with each sector representing a proportion of the total.

Use Case: Illustrating the percentage distribution of expenses in a budget.

5. Scatter Plot:

Description: A scatter plot displays individual data points on a two-dimensional graph, illustrating the relationship between two numerical variables.

Use Case: Examining the correlation between study hours and exam scores.

6. Box Plot (Box-and-Whisker Plot):

Description: A box plot displays the distribution of a numerical variable and identifies outliers.

Use Case: Analyzing the distribution of salaries in different departments.

7. Violin Plot:

Description: A violin plot combines aspects of a box plot and a kernel density plot to show the distribution of a numerical variable.

Use Case: Comparing the distribution of test scores between two groups.

8. Heatmap:

Description: A heatmap displays the intensity of values in a matrix using colors. It's often used to show correlations.

Use Case: Visualizing the correlation matrix of various financial indicators.

9. Area Chart:

Description: An area chart is similar to a line chart but fills the area beneath the line, emphasizing the magnitude of change.

Use Case: Showing the cumulative contribution of different revenue streams over time.

10. Radar Chart:

markdown

Copy code

- **Description:** A radar chart displays multivariate data in the form of a two-dimensional chart with three or more quantitative variables.

- **Use Case:** Comparing the performance of athletes in different sports based on multiple criteria.

11. Bubble Chart:

markdown

Copy code

- **Description:** A bubble chart extends the concept of a scatter plot by introducing a third variable, represented by the size of the bubbles.

- **Use Case:** Visualizing the relationship between GDP, population, and healthcare spending for different countries.

12. Polar Plot:

markdown

Copy code

- **Description:** A polar plot is a circular graph with radial axes, useful for displaying cyclical or periodic data.

- **Use Case:** Showing wind speed and direction over different days.

13. Tree Map:

vbnet

Copy code

- **Description:** A tree map represents hierarchical data using nested rectangles, with each level represented by a different color or shading.

- **Use Case:** Displaying the market share of different product categories and subcategories.

14. Waterfall Chart:

markdown

Copy code

- **Description:** A waterfall chart shows the cumulative effect of sequentially introduced positive or negative values.

- **Use Case:** Illustrating changes in net income over consecutive financial quarters.

15. Network Graph:

markdown

Copy code

- **Description:** A network graph displays interconnected nodes and edges, representing relationships between entities.

- **Use Case:** Visualizing social networks or relationships between various components in a system.

16. Parallel Coordinates Plot:

vbnet

Copy code

- **Description:** A parallel coordinates plot visualizes multivariate data by representing each observation as a line connecting points on parallel axes.

- **Use Case:** Analyzing and comparing the performance of products based on multiple attributes.

17. Word Cloud:

markdown

Copy code

- **Description:** A word cloud visually represents the frequency of words in a given text, with more frequently used words appearing larger.

- **Use Case:** Summarizing the most common words in customer reviews.

18. Choropleth Map:

markdown

Copy code

- **Description:** A choropleth map uses color variations to represent statistical data across geographical regions.

- **Use Case:** Displaying population density or unemployment rates by country.

19. Gantt Chart:

markdown

Copy code

- **Description:** A Gantt chart illustrates a project schedule by representing tasks or activities along a timeline.

- **Use Case:** Planning and tracking progress in project management.

20. Sunburst Chart:

markdown

Copy code

- **Description:** A sunburst chart is a radial diagram used to represent hierarchical data with nested rings.

- **Use Case:** Visualizing the breakdown of expenses in a budget across mu

# Python Code for above

Python Code Examples:

Here are examples of creating some of these charts using Python with Pandas, Matplotlib, and Seaborn:

pip install matplotlib seaborn plotly pandas).

1. Line Chart:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Example data
data = {'Year': [2010, 2011, 2012, 2013, 2014],
        'Sales': [50, 55, 70, 65, 80]}

df = pd.DataFrame(data)

# Plotting
plt.plot(df['Year'], df['Sales'])
plt.xlabel('Year')
plt.ylabel('Sales')
plt.title('Sales Over Time')
plt.show()
```

2. Bar Chart:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Example data
data = {'Category': ['A', 'B', 'C', 'D'],
        'Values': [30, 45, 55, 25]}
```

```
df = pd.DataFrame(data)

# Plotting
plt.bar(df['Category'], df['Values'])
plt.xlabel('Category')
plt.ylabel('Values')
plt.title('Comparison of Categories')
plt.show()
```

3. Histogram:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Example data
data = {'Values': [15, 25, 35, 45, 55, 65, 75, 85, 95]}

df = pd.DataFrame(data)

# Plotting
plt.hist(df['Values'], bins=5)
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Distribution of Values')
plt.show()
```

4. Pie Chart:

python

Copy code

```
import pandas as pd
```

```
import matplotlib.pyplot as plt

# Example data
data = {'Category': ['A', 'B', 'C', 'D'],
        'Values': [30, 45, 55, 25]}

df = pd.DataFrame(data)

# Plotting
plt.pie(df['Values'], labels=df['Category'], autopct='%1.1f%%')
plt.title('Distribution of Categories')
plt.show()
```

5. Scatter Plot:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Example data
data = {'X': [1, 2, 3, 4, 5],
        'Y': [10, 15, 25, 30, 35]}

df = pd.DataFrame(data)

# Plotting
plt.scatter(df['X'], df['Y'])
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatter Plot')
plt.show()
```

6. Box Plot (Box-and-Whisker Plot):

python

Copy code

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Example data
data = {'Category': ['A', 'B', 'A', 'B', 'A', 'B'],
        'Values': [30, 45, 55, 25, 65, 75]}

df = pd.DataFrame(data)

# Plotting
sns.boxplot(x='Category', y='Values', data=df)
plt.xlabel('Category')
plt.ylabel('Values')
plt.title('Box Plot')
plt.show()
```

7. Violin Plot:

python

Copy code

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Example data
data = {'Category': ['A', 'B', 'A', 'B', 'A', 'B'],
        'Values': [30, 45, 55, 25, 65, 75]}
```

```
df = pd.DataFrame(data)

# Plotting
sns.violinplot(x='Category', y='Values', data=df)
plt.xlabel('Category')
plt.ylabel('Values')
plt.title('Violin Plot')
plt.show()
```

8. Heatmap:

python

Copy code

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Example data
data = {'A': [1, 2, 3, 4],
        'B': [4, 3, 2, 1],
        'C': [2, 4, 1, 3]}

df = pd.DataFrame(data)

# Plotting
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

9. Area Chart:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Example data
data = {'Year': [2010, 2011, 2012, 2013, 2014],
        'Sales': [50, 55, 70, 65, 80]}

df = pd.DataFrame(data)

# Plotting
plt.fill_between(df['Year'], df['Sales'], color='skyblue', alpha=0.4)
plt.xlabel('Year')
plt.ylabel('Sales')
plt.title('Sales Over Time (Area Chart)')
plt.show()
```

10. Radar Chart:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt
from math import pi

# Example data
data = {'Category': ['A', 'B', 'C', 'D'],
        'Values': [30, 45, 55, 25]}

df = pd.DataFrame(data)

# Plotting
```

```
categories = list(df['Category'])
values = list(df['Values'])

values += values[:1]
categories += categories[:1]

angles = [n / float(len(categories)) * 2 * pi for n in range(len(categories))]
angles += angles[:1]

plt.polar(angles, values, marker='o')
plt.fill(angles, values, alpha=0.25)
plt.xticks(angles[:-1], categories)
plt.title('Radar Chart')
plt.show()
```

11. Bubble Chart:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Example data
data = {'X': [1, 2, 3, 4, 5],
        'Y': [10, 15, 25, 30, 35],
        'Size': [20, 40, 60, 80, 100]}

df = pd.DataFrame(data)

# Plotting
plt.scatter(df['X'], df['Y'], s=df['Size'])
plt.xlabel('X')
```

```
plt.ylabel('Y')
plt.title('Bubble Chart')
plt.show()
```

12. Polar Plot:

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Example data
data = {'Category': ['A', 'B', 'C', 'D'],
        'Values': [30, 45, 55, 25]}

df = pd.DataFrame(data)

# Plotting
ax = plt.subplot(111, polar=True)
theta = np.linspace(0, 2*np.pi, len(df['Category']), endpoint=False)
values = df['Values'].tolist()
values += values[:1]
theta += theta[:1]
ax.plot(theta, values, marker=
ax.plot(theta
'o')
ax.fill(theta, values, alpha=
ax.fill(theta, values,

ax.fill
0.25)
```

ax.set_xticks(theta[:-

ax.set_xticks(theta

ax.set_xt

1])

ax.set_xticklabels(df[

ax.set_xticklabels

ax.'Category'])

plt.title('Polar Plot')

plt.show()

13. Tree Map:

python

Copy code

import squarify

import matplotlib.pyplot as plt

# Example data

data = {

data =

'Category': ['A', 'B', 'C', 'D'],

'Values': [30, 45, 55, 25]}

df = pd.DataFrame(data)

df =

# Plotting

squarify.plot(sizes=df[

squarify

'Values'], label=df['Category'], alpha=0.7)

plt.axis(

plt

'off')

plt.title('Tree Map')

plt.show()

plt.show

14. Waterfall Chart:

python

Copy code

import matplotlib.pyplot as plt

# Example data

data = {

data

'Category': ['Start', 'Phase 1', 'Phase 2', 'End'],

'Values': [20, 10, -5, 25]}

df = pd.DataFrame(data)

# Plotting

plt.bar(df[

plt.bar(df

'Category'], df['Values'], color=['blue', 'green', 'red', 'blue'])

plt.xlabel('Category')

plt.ylabel(

plt.ylabel

'Values')

```
plt.title(
plt
'Waterfall Chart')
plt.show()
```

15. Network Graph:

python

```
Copy code
import networkx as nx
import matplotlib.pyplot as plt

# Example data
data = {
data =
'Node1': ['A', 'B', 'C', 'D'],
      'Node2': ['B', 'C', 'D', 'A']}

df = pd.DataFrame(data)

# Creating a graph
G = nx.from_pandas_edgelist(df,
G = nx.from_pandas_edg

G = nx.from_pandas_ed
'Node1', 'Node2')

# Plotting
nx.draw(G, with_labels=
nx.draw
True, font_weight='bold', node_color='skyblue', edge_color='gray', node_size=1000)
plt.title('Network Graph')
```

plt.show()

plt.show()
``

plt.show()
16. Parallel Coordinates Plot:
python

Copy code
from pandas.plotting import parallel_coordinates
import matplotlib.pyplot as plt

# Example data
data = {
data =
'Category': ['A', 'B', 'C', 'D'],
    'Value1': [30, 45, 55, 25],
    'Value2': [20, 40, 60, 80]}

df = pd.DataFrame(data)

df = pd.DataFrame

df
# Plotting
parallel_coordinates(df,
parallel_coordinates

'Category', colormap='viridis')

plt.title('Parallel Coordinates Plot')

plt.show()

plt.show

plt

17. Word Cloud:

python

Copy code

from wordcloud import WordCloud

import matplotlib.pyplot as plt

# Example data

text =

text

"data analysis is fun and interesting"

# Generating word cloud

wordcloud = WordCloud(width=

wordcloud = WordCloud

800, height=400, background_color='white').generate(text)

# Plotting

plt.figure(figsize=(10, 5))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title(

plt.title

'Word Cloud')

```
plt.show()


plt.show()
```
18. Choropleth Map:

python

```
Copy code
import plotly.express as px

# Example data
data = {'Country': ['USA', 'Canada', 'Mexico'],
        'Population': [327, 38, 126]}

df = pd.DataFrame(data)

# Plotting
fig = px.choropleth(df, locations=
fig = px
'Country', locationmode='country names', color='Population',
                    color_continuous_scale=
                    color
'Viridis', title='Choropleth Map')
fig.show()
```
19. Gantt Chart:

python

```
Copy code
import plotly.express as px

# Example data
```

```
data = {
data =

data
'Task': ['Task A', 'Task B', 'Task C'],
    'Start': ['2023-01-01', '2023-02-01', '2023-03-01'],
    'Finish': ['2023-02-15', '2023-03-15', '2023-04-01']}

df = pd.DataFrame(data)
df[

df = pd.DataFrame(data)
df



df = pd.DataFrame(data)



df
'Start'] = pd.to_datetime(df['Start'])
df['Finish'] = pd.to_datetime(df['Finish'])

# Plotting
fig = px.timeline(df, x_start=
fig = px.timeline
'Start', x_end='Finish', y='Task', title='Gantt Chart')
fig.update_yaxes(categoryorder=
fig.update_yaxes(category

fig
'total ascending')
```

```
fig.show()
```

20. Sunburst Chart:

python

Copy code

```
import plotly.express as px

# Example data
data = {'Category': ['A', 'B', 'C', 'D'],
        'Parent': ['', 'A', 'B', 'C'],
        'Values': [30, 45, 55, 25]}

df = pd.DataFrame(data)

# Plotting
fig = px.sunburst(df, path=['Parent', 'Category'], values='Values', title='Sunburst Chart')
fig.show()
```

# Question from previous paper

**You are tasked with creating a visualisation showing "the distribution of expenditure by staff level across 4 quarters in 2019". Use the sample data to create a graph to achieve this goal. You can use any tool to create the graph or you can sketch (pencil/paper) or mock up (e.g., powerpoint). Annotate or describe any specific design ideas. Paste your graph, sketch or mock-up diagram in the answer section below. Marks will be awarded for appropriate choice of graph, effective visualisation design and clarity of the message.**

Bar Chart:

Use a bar chart with the x-axis representing the staff levels and the y-axis representing the average expense claim value (€).

Each staff level (Lab worker, Administrator, Management, Senior Management, CEO) will have a corresponding bar.

Different colors can be used to distinguish between the quarters.

Quarterly Distribution:

To represent the distribution across quarters, you can either use grouped bars for each staff level, where each bar is divided into four segments (one for each quarter), or use side-by-side bars for each quarter.

Annotations:

Includes notes

Clearly label the axes with "Staff Levels" on the x-axis and "Average Expense Claim Value (€)" on the y-axis.

In

Colors:

Use a color scheme that is easily distinguishable and makes sense for the purpose of your visualization. For example, you could use different shades for each quarter.

Title:

Add a clear and concise title, such as "Distribution of Expenditure by Staff Level Across Quarters in 2019".

**Given the scenario below choose the most appropriate graph type to visualise the message and justify your choice referencing the message and the probable data types and indicating the marks and visual attributes that will be used to encode the data.**

**“Compare the energy efficiency ratings (0-5 star) vs price (in Euro) for consumer electronics such as fridges, ovens, heaters, etc.”**

Scatter Plot:

Message:

A scatter plot is effective when you want to observe the relationship between two continuous variables, in this case, energy efficiency ratings (numeric) and price (numeric).

It allows you to see if there is any pattern, trend, or correlation between energy efficiency and price.

Data Types:

Energy Efficiency Rating: Numeric (0-5 stars) - discrete or ordinal data.

Price: Numeric (in Euro) - continuous data.

Marks and Visual Attributes:

Encoding Energy Efficiency Ratings (0-5 Stars):

Use of

Encoding Price (in Euro):

Represent the price on the y-axis using the position of the points. Higher on the y-axis corresponds to a higher price.

Use a color gradient or size variation to encode price. For instance, lighter colors or larger markers for higher prices.

Interactivity (Optional):

Consider adding interactivity, like hover-over tooltips, to provide detailed information about each point, such as the exact energy efficiency rating and price.

Title and Axes Labels:

Clearly label the x-axis as "Energy Efficiency Rating (0-5 Stars)" and the y-axis as "Price (Euro)".

Include a title that succinctly conveys the purpose of the visualization, such as "Comparison of Energy Efficiency Ratings vs. Price for Consumer Electronics."

Example Code (using Python with Matplotlib and Seaborn):

pytho

Copy code

```
import matplotlib.pyplot as plt

import seaborn as sns


# Example data (replace this with your actual dataset)

data = {

data = {

'Energy Efficiency Rating': [3, 4, 2, 5, 1],
```

```
'Price (Euro)': [500, 800, 300, 1200, 200]}


df = pd.DataFrame(data)




df = pd.DataFram



df =

# Plotting

sns.scatterplot(x='Energy Efficiency Rating', y='Price (Euro)', data=df, hue='Energy Efficiency Rating', palette='viridis', s=100)

plt.title(

plt.t

'Comparison of Energy Efficiency Ratings vs. Price for Consumer Electronics')

plt.xlabel(

plt.xlabel

'Energy Efficiency Rating (0-5 Stars)')

plt.ylabel('Price (Euro)')

plt.show()
```

# PREATTENTIVE FEATURES

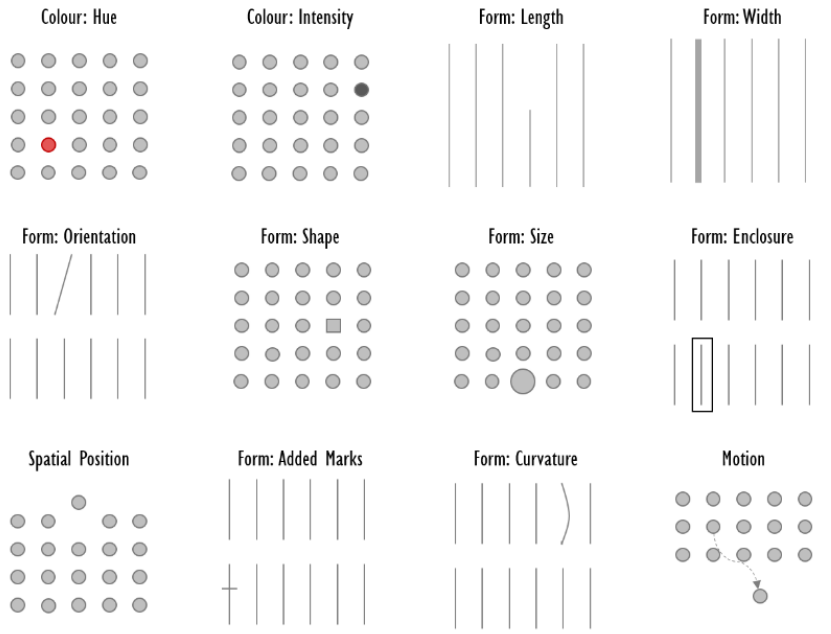



*Fig: Common preattentive attributes (from Few (2004), Healy and Enns (2012) and Knaflic (2016)).*

# COMMUNICATION

When approaching a question that asks you to identify the main communication purpose of an image and explain your reasoning, consider the following steps:

Observe the Image:

Take a moment to carefully observe the image. Look for key elements, such as text, visuals, colors, and overall layout.

Identify Key Components:

Identify the key components in the image, including any text, images, charts, or graphs.

Consider Design Elements:

Evaluate the use of design elements such as color, font, size, and layout. Consider how these elements contribute to the overall message.

Think About Audience:

Consider the likely audience for the image. Who is the intended viewer, and what information might be most relevant to them?

Communication Purpose:

Formulate an understanding of the main communication purpose. What message is the image trying to convey, and what action or understanding is it aiming to achieve?

Refer to Design Guidelines:

If the question mentions specific design guidelines, refer to them. Common design principles include clarity, simplicity, hierarchy, contrast, and alignment. Explain how these principles are applied in the image.

Provide Specific Examples:

Support your identification of the main communication purpose by pointing out specific elements in the image. For instance, mention if there is a call-to-action, a central message, or a focus on specific data points.

Consider Context:

Think about the context in which the image might be used. Is it part of a presentation, an advertisement, an infographic, or a report? The context can provide clues about the primary communication goal.

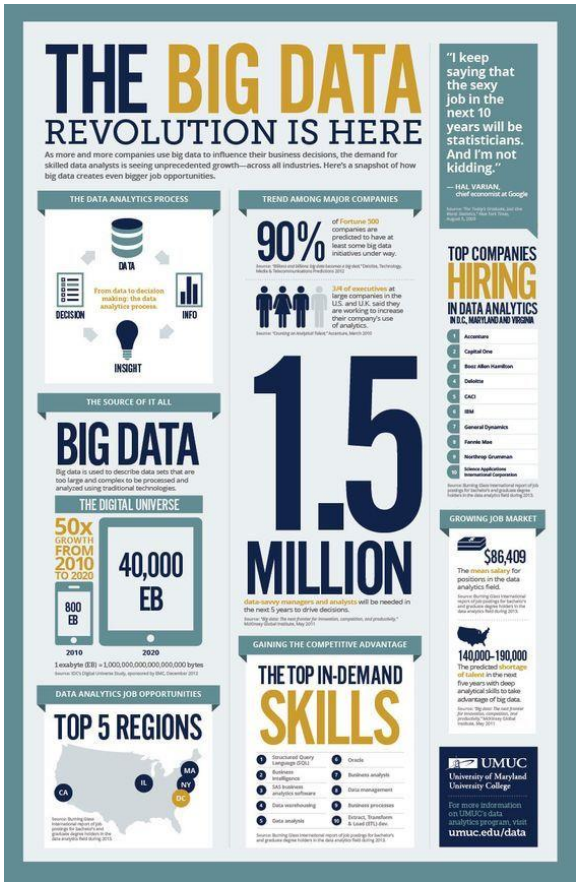
Articulate Your Reasoning:

Clearly articulate your reasoning. Explain how the design choices contribute to the overall communication goal and how they adhere to design principles.

Provide a Concise Answer:

Summarize your observations and reasoning in a concise answer that directly addresses the main communication purpose of the image.

**Previous Question**




The main communication purpose of the image you sent is to educate people about the importance of big data and the growing demand for data analysts.

The poster is designed to be informative and to make people want to learn more about big data. It uses a variety of design guidelines to achieve this goal:

Clear and concise headline: The headline, "THE BIG DATA REVOLUTION IS HERE," is clear and concise. It immediately tells the viewer what the poster is about.

Use of data and statistics: The poster uses a variety of data and statistics to support its claims about the importance of big data and the growing demand for data analysts. For example, the poster states that "90% of Fortune 300 companies are predicted to have big data teams under development" and that "the median salary for positions in the data analytics field is $86,409."

Use of visuals: The poster uses a variety of visuals to make the information more engaging and easier to understand. For example, the poster uses a chart to show the growth of the big data job market and a graphic to illustrate the data analytics process.

Use of white space: The poster uses white space effectively to make the information more readable and less cluttered.

Overall, the poster is well-designed and effective in communicating its message. It is clear, concise, informative, and engaging.

Here are some additional specific design guidelines that support the main communication purpose of the image:

The use of a bright blue background and white text creates a high-contrast design that is easy to read and visually appealing.

The use of bold text and large fonts for the headline and key information makes the poster stand out and easy to scan.

The use of a grid layout helps to organize the information in a logical and easy-to-understand way.

The use of icons and graphics helps to illustrate complex concepts and make the information more visually appealing.

Overall, the design of the image is effective in communicating the main message of the importance of big data and the growing demand for data analysts.

| |
|---|
| Sender<br><br>Receiver<br><br>Message<br><br>Medium<br><br><br>Justification |

Sender:

The sender of the image is the University of Maryland Global Campus (UMUC). This is evident from the text on the image that says "UMUC is preparing you for the in-demand field of big data analytics." The phrase "UMUC is preparing you" suggests that the image is intended to promote UMUC's data analytics programs.

Receiver:

The receivers of the image are individuals who are interested in learning more about big data analytics and pursuing a career in this field. This includes aspiring data analysts, students who are curious about data science, and professionals who want to enhance their skillset with data analytics knowledge.

Message:

The main message of the image is to emphasize the importance and growing demand of big data analytics. The image highlights the high salary potential for data analysts and the abundance of job opportunities available in this field. It also stresses the fact that UMUC is a leading provider of data analytics education and training.

Medium:

The most probable medium for the image is digital advertising, such as social media ads or banner ads on websites. This is because digital advertising allows UMUC to reach a wide audience of potential students who are interested in data analytics. The image's visually appealing design and clear messaging make it effective for use in digital advertising campaigns.

Assumptions:

The analysis of the sender, receiver, message, and medium is based on the following assumptions:

The image is intended to promote UMUC's data analytics programs.

The image is targeted towards individuals who are interested in learning more about big data analytics and pursuing a career in this field.

The image is being used in a digital advertising campaign.

# Problems with the graph

Certainly! Below are various possible problems with a graph and potential improvements, along with justifications for each. Note that these examples are general and may not apply specifically to any given graph:

Unclear Axis Labels:

Issue: The graph has unclear or missing axis labels.

Improvement: Clearly label both axes, including units of measurement if applicable.

Justification: Clarity in axis labeling is fundamental for understanding the context of the data.

Misleading Scale:

Issue: The y-axis scale is not consistent or is truncated, leading to potential misinterpretation of the data.

Improvement: Adjust the y-axis scale to ensure it accurately represents the data without truncation.

Justification: Consistent and accurate scales are crucial for preventing misinterpretation.

Inappropriate Chart Type:

Issue: The chosen chart type may not be the most suitable for the data being presented.

Improvement: Consider alternative chart types that better represent the relationships within the data.

Justification: Choosing the right chart type enhances the effectiveness of data communication.

Color Issues:

Issue: Poor color choices or excessive use of color can lead to confusion.

Improvement: Choose a color palette that is accessible and ensures clear differentiation between data elements.

Justification: Color should enhance, not hinder, the interpretation of the graph.

Overcrowded Labels:

Issue: Labels on the graph are overcrowded, making it difficult to read.

Improvement: Consider rotating labels, using abbreviations, or providing a legend to declutter the graph.

Justification: Legible labels are essential for effective communication.

Missing Title:

Issue: The graph lacks a clear and informative title.

Improvement: Add a title that succinctly describes the main message or purpose of the graph.

Justification: A title provides context and guides viewers in understanding the graph's content.

Inconsistent Data Representation:

Issue: Data elements are inconsistently represented, making comparisons challenging.

Improvement: Ensure uniformity in how data points are represented (e.g., consistent use of symbols or colors).

Justification: Consistency aids in straightforward interpretation and comparison.

Data Overload:

Issue: Too much data or information on the graph can overwhelm the viewer.

Improvement: Simplify the graph by focusing on key data points or providing interactive features for exploration.

Justification: Clarity is compromised when there is information overload.

Lack of Context or Annotations:

Issue: The graph lacks context or explanatory annotations.

Improvement: Add contextual information or annotations to guide viewers in interpreting the data.

Justification: Contextual information enhances the viewer's understanding of the significance of the data.

Non-Zero Baseline for Bar Charts:

Issue: Bar charts with a baseline starting at a value other than zero can be misleading.

Improvement: Ensure that bar charts start from zero to avoid exaggeration of differences.

Justification: A zero baseline provides an accurate representation of the magnitude of differences.

# ENCODING MARKERS

ENCODING CHANNELS

CHARACTERISTICS

| | POSITION | SHAPE | SIZE | BRIGHTNESS | HUE | ANGLE | TEXTURE | MOTION |
|---|---|---|---|---|---|---|---|---|
| SELECTIVE | ✓ | ✓ | ✓ | ✓ | ✓ | ≈ (somewhat) | ✓ | ✓ |
| ASSOCIATIVE | ✓ | ≈ (somewhat) | ✓ | ✓ | ✓ | ≈ (somewhat) | ✓ | ✓ |
| ORDINAL | ✓ | ✗ | ✓ | ✓ | ✗ | ≈ (somewhat) | ✗ | ✓ |
| QUANTITATIVE | ✓ | ✗ | ✓ | ≈ (somewhat) | ✗ | ≈ (somewhat) | ✗ | ✓ |
| RANGE | High | High | High | Limited | Limited | Limited | High | Limited |