1. **How do you treat duplicate records?**

   We can use df.drop_duplicates for removing Duplicates record

   We can also detect them by using df.duplicated().sum()

2. **Difference between dropna() and fillna() in Pandas?**

Method	Description	Use When
dropna()	Removes rows/columns with missing values	You want to ignore null rows
fillna()	Replaces missing values with a value	You want to retain all rows

3. **What is outlier treatment and why is it important?**

Ans: - Outlier treatment refers to the process of identifying and handling data points that deviate significantly from other observations in a dataset. These unusual values are known as **outliers** and can arise due to errors, variability in data, or rare events.

(a) **Skew Results**
Outliers can distort statistical metrics like the **mean, standard deviation, and correlation**, leading to misleading interpretations.

(b) **Affect Model Accuracy**
Machine learning models (especially linear models) can be highly sensitive to outliers, resulting in **poor predictions or overfitting**.

(c) **Impact Visualizations**
Plots like histograms, box plots, or scatter plots may become **hard to interpret** due to extreme values.

(d) **Influence Business Decisions**
In domains like finance or healthcare, **outliers can mask real trends or falsely signal risks**.

Treatment Techniques:

- **Capping**: Use IQR method to limit extreme values
- **Removal**: Drop rows outside expected range

- **Transformation**: Apply log or sqrt

4. **Explain the process of standardizing data.**

   And : - Standardizing data (also called **Z-score normalization**) is the process of transforming your data so that it has:

   - **Mean = 0**
   - **Standard Deviation = 1**

   This is especially important for machine learning models that are sensitive to the **scale** of the data (e.g., linear regression, k-means, SVM).

   Why Standardize?

   Different features may have different units (e.g., age in years vs. income in dollars), which can:

   - **Bias models** toward features with larger ranges.
   - **Slow down training** for gradient-based algorithms.

   Standardization puts all features on the same scale, ensuring **fair contribution** from each feature.

   Formula for Standardization (Z-score):

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

   Where:

   - X = Original value
   - μ = Mean of the feature
   - σ = Standard deviation of the feature

5. **How do you handle inconsistent data formats (e.g., date/time)?**

   We can use pd.to_datetime() to standardize dates:

```
df['date'] = pd.to_datetime(df['date'], dayfirst=True)
```

6. **What are common data cleaning challenges?**

   (a) Missing values
   (b) Duplicates
   (c) Inconsistent column names
   (d) Incorrect data types
   (e) Outliers
   (f) Categorical inconsistencies
   (g) Mismatched date/time formats

7. **How can you check data quality?**

(a) isnull().sum() → Missing values
(b) .duplicated().sum() → Duplicates
(c) .describe() → Summary stats for outliers
(d) .info() → Data types
(e) Unique

8. What are missing values and how do you handle them?

Missing values (NaN) occur when data is not recorded or corrupted.

Handling techniques:

(a) **Remove rows**: df.dropna()

(b) **Impute with value**: df.fillna(0) or df.fillna(df['Income'].mean())

(c) **Forward/backward fill**: df.fillna(method='ffill')