10 Data Cleaning Techniques Every Analyst Should Master

1. Handling Missing Data

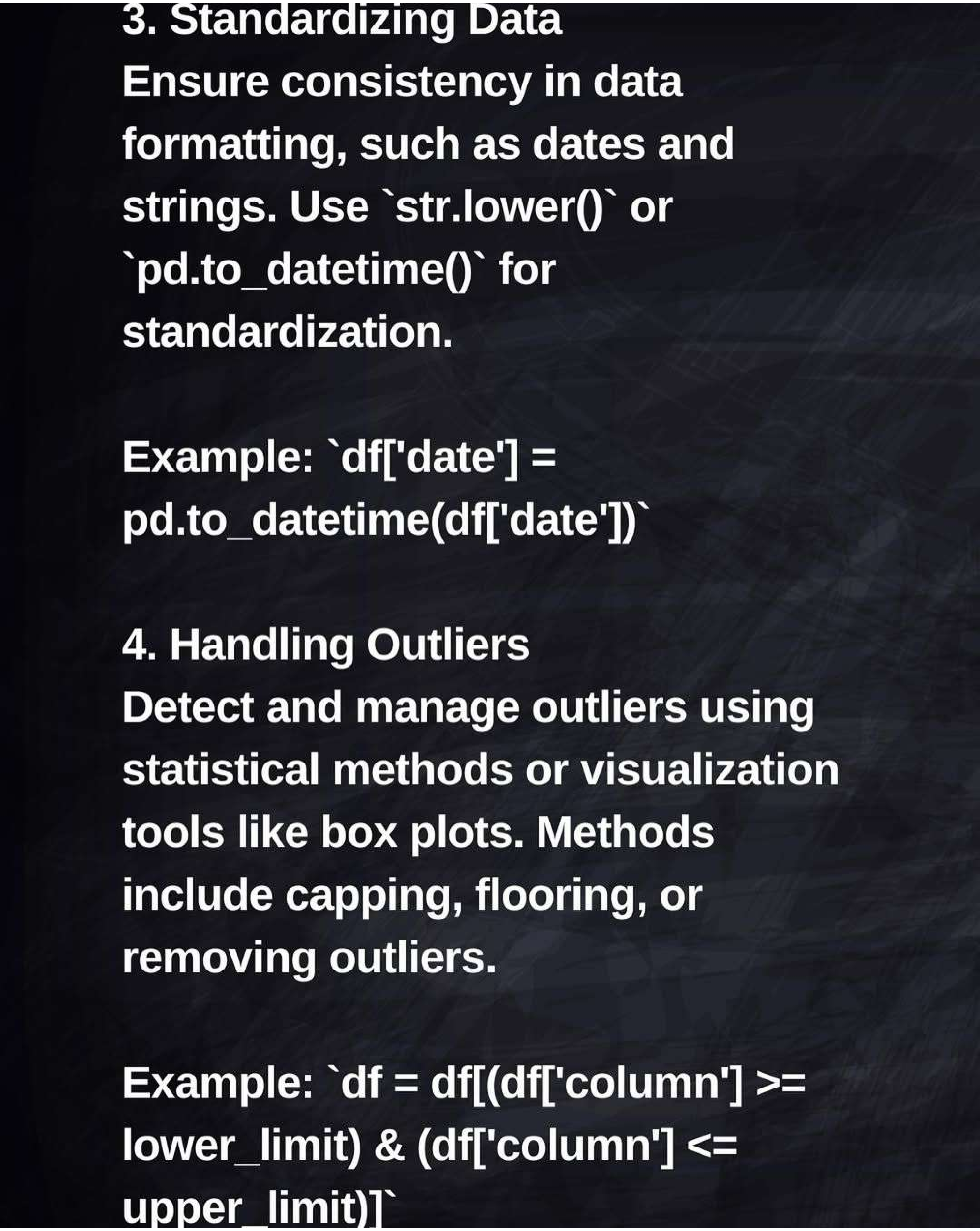
Use methods like imputation (mean, median, mode) or deletion to handle missing values. In Python, `pandas` functions such as `fillna()` or `dropna()` are useful.

Example: `df.fillna(df.mean())` replaces missing values with the column mean.

2. Removing Duplicates

Identify and remove duplicate records to ensure the dataset is accurate and reliable. Use `drop_duplicates()` in pandas.

Example:
`df.drop_duplicates(inplace=True)`



3. Standardizing Data

Ensure consistency in data formatting, such as dates and strings. Use `str.lower()` or `pd.to_datetime()` for standardization.

Example: `df['date'] = pd.to_datetime(df['date'])`

4. Handling Outliers

Detect and manage outliers using statistical methods or visualization tools like box plots. Methods include capping, flooring, or removing outliers.

Example: `df = df[(df['column'] >= lower_limit) & (df['column'] <= upper_limit)]`

5. Correcting Data Types

Ensure all columns have the correct data types for analysis. Use `astype()` in pandas to convert data types.

Example: `df['column'] = df['column'].astype('int')`

6. Normalizing and Scaling Data

Normalize or scale data to bring all values into a similar range, which is essential for algorithms like K-Means clustering. Use `StandardScaler` or `MinMaxScaler` from `scikit-learn`.

Example: `from sklearn.preprocessing import StandardScaler; df_scaled =

7. Encoding Categorical Variables
Convert categorical data into numerical format using techniques like one-hot encoding or label encoding. Use `pd.get_dummies()` or `LabelEncoder`.

Example: `df_encoded = pd.get_dummies(df, columns=['category'])`

8. Dealing with Inconsistent Data
Identify and correct inconsistencies in data entries, such as typos or inconsistent naming conventions.

Example: `df['column'] = df['column'].replace({'val1':'value1', 'val2':'value2'})`

9. Parsing and Extracting Data

Extract relevant information from complex data types such as strings or dates. Use string methods or regular expressions.

Example: `df['year'] = df['date'].dt.year`

10. Combining Multiple Data Sources

Merge or concatenate multiple datasets to create a comprehensive dataset. Use `merge()` or `concat()` in pandas.

Example: `df_combined = pd.merge(df1, df2, on='key_column')`