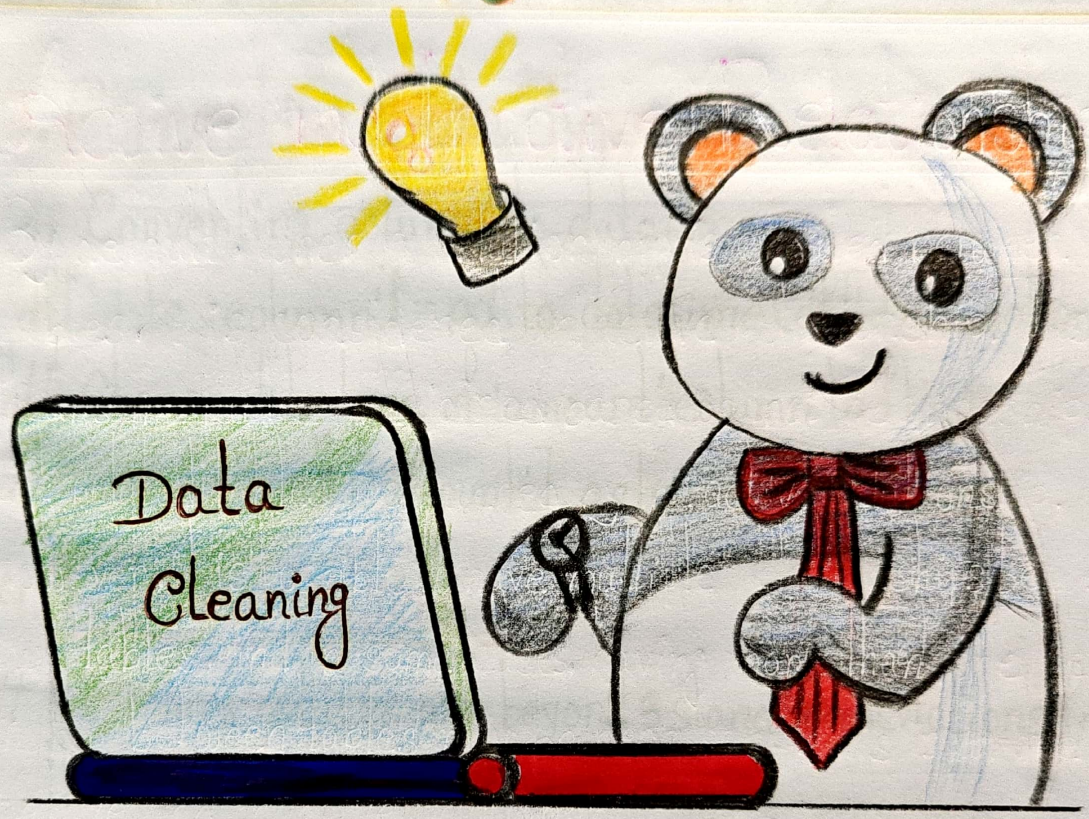
Top 20 Data Cleaning Methods Along with Python Coding Examples



1. Removing Duplicates :

```
Import pandas as pd
# Assuming df is your DataFrame
df.drop_duplicates(inplace = True)
```

2. Handling Missing Values :

```
# Removing rows with missing values
df.dropna(inplace = True)
# Filling missing values with mean
df.fillna(df.mean(), inplace = True)
```

3. Correcting Data Types :

```
# Converting column to datetime
df['date_column'] = pd.to_datetime(df['date_column'])
# Change data type of 'amount' column to float
df['amount'] = df['amount'].astype(float)lower
```

4. Filtering Outliers :

```
# Assuming outliers are beyond 3 standard deviations from
  the mean
df = df[(df['column'] < df['column'].mean()+3*df['column'].
std()) & (df['column'] > df['column'].mean()-3*df['column'].std())]
```

5. Fixing Typos and Inconsistencies :

```
#Correcting categorical values
df['category_column'].replace({'incorrect_value':'correct_value'
}, inplace = True)
```

6. Parsing Dates and Time:

```
# Assuming date column is in string format
df['date_column'] = pd.to_datetime(df['date_column'])
```

7. Removing Irrelevant Variables:

```
# Dropping columns
df.drop(['irrelevant_column1', 'irrelevant_column2'],
axis=1, inplace=True)
```

8. Handling Special Characters:

```
# Assuming special characters are in column 'text column'
df['text_column'] = df['text_column'].str.replace
(r'[^\x00-\x7F]+', '')
```

9. Dealing with White Spaces:

```
# Removing leading and trailing white spaces
df['text_column'] = df['text_column'].str.strip()
```

10. Converting Text to Lower/Upper Case:

```
# Converting text to lower case
df['text_column'] = df['text_column'].str.lower()
```

11. Handling Inconsistent Formatting:

```
# Assuming column 'text_column' needs consistent formatting
df['text_column'] = df['text_column'].apply(lambda x: x.capitalize())
```

12. Binning or Discretization:

```
# Assuming binning based on quantiles
df['binned_column'] = pd.qcut(df['numeric_column'], q=4, labels=False)
```

13. Removing Incomplete Records:

```
# Removing rows with missing values
df.dropna(inplace=True)
```

14. Encoding Categorical Variables:

```
# Using one-hot encoding
df_encoded = pd.get_dummies(df, columns=['categorical_column'])
```

15. Filling Missing Values:

```
# Filling missing values with mean
df.fillna(df.mean(), inplace=True)
```

16. Handling Skewed Data:

```
import numpy as np
# Applying log transformation
df['skewed_column'] = np.log(df['skewed_column'])
```

17. Feature Engineering:

```
# Creating new feature based on existing ones
df['new_feature'] = df['feature1'] * df['feature2']
```

18. Removing outliers:

```
# Remove outliers using Tukey's method
Q1 = df['column'].quantile(0.25)
Q3 = df['column'].quantile(0.75)
IQR = Q3 - Q1
df = df[~((df['column'] < (Q1 - 1.5 * IQR)) | (df['column'] > (Q3 + 1.5 * IQR)))
```

19. Encoding:

```
# One-hot encoding
df = pd.get_dummies(df, columns=['categorical_column'])
```

20. Renaming Columns:

```
>>> df.rename(columns={'Sales':'Revenue'}, inplace=True)
```