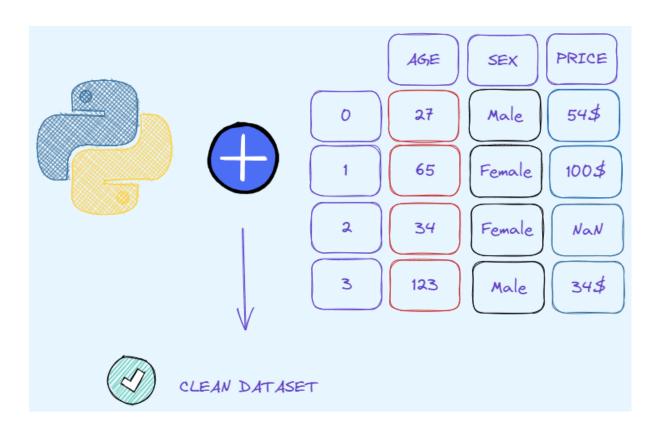# Data Cleaning Cheat Sheet in Python - By Eugenia Anello

Table of Contents:

1. Dealing with Missing data

Check missing data in each column of the dataset

```
df.isnull().sum()
```

Delete missing data

```
df.dropna(how='all')
```

Drop columns that have missing values

```
df.dropna(how='columns')
```

Drop specific columns that have missing values

```
df.dropna(subset=['municipal','city'])
```

Replace missing values with Mean/Median/Mode

```
df['price'].fillna(df['price'].mean())
df['age'].fillna(df['age'].median())
df['type_building'].fillna(df['type_building'].mode())
```

Forward Fill - Fill missing values with values before them

```
df.fillna(method='ffill')
```

Backward Fill - FIll missing values with values after them

```
df.fillna(method='bfill')
```

Fill missing values using the interpolation method

```
df['stock_price'] =
df['stock_price'].interpolate(method='polynomial',order=2)
```

2. Dealing with Duplicates

Check if there are duplicates

```
df.duplicated().sum()
```

Extract duplicate rows from the dataframe

```python
df[df.duplicated()]
```

Drop duplicates

```python
df.drop_duplicates()
```

Aggregate data

```python
df.groupby('id').agg({'price':'mean'}).reset_index()
```

3. Outlier detection

Detect range of values for each column of the dataset

```python
df.describe([x*0.1 for x in range(10)])
```

Display boxplot to display the distribution of a column

```python
import seaborn as sns
sns.boxplot(x=df['age'])
```

Display histogram to display the distribution of a column

```python
sns.displot(data=df['column1'])
```

Remove outliers

```python
df = df[df['age']<df['age'].quantile(0.9)]
```

Outlier detection with machine learning models, like Isolation Forest

```python
if = IsolationForest(random_state=42)
if.fit(X)
y_pred = if.predict(X)
```

4. Encode categorical features

Apply one-hot-encoding to a categorical column

```python
from sklearn.prepreprocessing import OneHotEncoder
ohe = OneHotEncoder()
encoded_data =
pd.DataFrame(ohe.fit_transform(df[['type_building','color']]).toarray())
new_df = df.join(encoded_data)
```

Apple label-encoding to a categorical column

```python
from sklearn.prepreprocessing import LabelEncoder
le = LabelEncoder()
df['column1'] = le.fit_transform(df['price_levels'])
```

5. Transformation

Standardize features by removing the mean and scaling to unit variance

```python
from sklearn.processing import StandardScaler
X_std = StandardScaler().transform(X)
```

Rescale features into the range [0,1]

```python
from sklearn.processing import MinMaxScaler
X_mms = MinMaxScaler().transform(X)
```

Scale features exploiting statistics that are robust to outliers

```python
from sklearn.processing import RobustScaler
X_rs = RobustScaler().transform(X)
```