Reference guide: Data cleaning in Python

This reference guide contains common functions and methods that data professionals use to clean data. The reference guide contains three different tables of useful tools, each grouped by cleaning category: missing data, outliers, and label encoding.

Save this course item

You may want to save a copy of this guide for future reference. You can use it as a resource for additional practice or in your future professional projects. To access a downloadable version of this course item, click the link below and select "Use Template."

Reference guide: Data cleaning in Python

OR

If you don't have a Google account, you can download the item directly from the attachment below.

Reference guide_ Data cleaning in Python
DOCX File

Missing data

The following pandas functions and methods are helpful when dealing with missing data.

df.info()

- **Description:** A `DataFrame` method that returns a concise summary of the dataframe, including a 'non-null count,' which helps you know the number of missing values

Example:

```
print(df)
print()
df.info()
```

Run
Reset

pd.isna() / pd.isnull()

- **Description:** `pd.isna()` is a pandas function that returns a same-sized Boolean array indicating whether each value is null (you can also use `pd.isnull()` as an alias). Note that this function also exists as a `DataFrame` method.

Example:

```
print(df)
print('\n After pd.isnull(): \n')

pd.isnull(df)
```

Run
Reset

pd.notna() / pd.notnull()

- **Description:** A pandas function that returns a same-sized Boolean array indicating whether each value is NOT null (you can also use `pd.notnull()` as an alias). Note that this function also exists as a `DataFrame` method.

Example:

```
print(df)
print('\n After notnull(): \n')
pd.notnull(df)
```

Run
Reset

df.fillna()

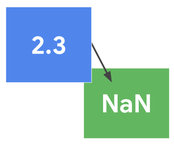
- **Description:** A `DataFrame` method that fills in missing values using specified method

Example:

```
print(df)
print('\n After fillna(): \n')

df.fillna(2)
```

Run
Reset



df.replace()

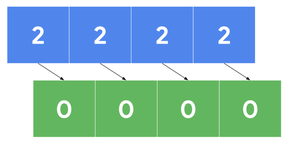
- **Description:** A `DataFrame` method that replaces specified values with other specified values. Can also be applied to pandas `Series`.

Example:

```
print(df)
print('\n After replace(): \n')

df.replace('Aves', 'bird')
```

Run
Reset



df.dropna()

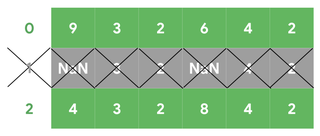
- **Description:** A `DataFrame` method that removes rows or columns that contain missing values, depending on the axis you specify.

Example:

```
print('Original df: \n \n', df)
print('\n After dropna(axis=0): \n')
print(df.dropna(axis=0))

print('\n After dropna(axis=1): \n')
print(df.dropna(axis=1))
```

Run
Reset



Outliers

The following tools are helpful when dealing with outliers in a dataset.

df.describe()

- **Description:** A `DataFrame` method that returns general statistics about the dataframe which can help determine outliers

Example:

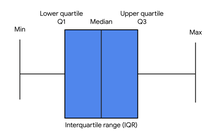
```
print(df)
print()
df.describe()
```

Run
Reset

sns.boxplot()

- **Description:** A seaborn function that generates a box plot. Data points beyond 1.5x the interquartile range are considered outliers.

Example:



Label encoding

The following tools are helpful when performing label encoding.

df.astype()

- **Description:** A `DataFrame` method that allows you to encode its data as a specified dtype. Note that this method can also be used on `Series` objects.

Example:

```
print(df)
print('\n Original dtypes of df: \n')

print(df.dtypes)

print('\n dtypes after casting \'class\' column as categorical: \n')

df['class'] = df['class'].astype('category')

print(df.dtypes)
```

Run
Reset

Series.cat.codes

- **Description:** A `Series` attribute that returns the numeric category codes of the series.

Example:

```
# Cast 'class' column as categorical
df['class'] = df['class'].astype('category')

print('\n \'class\' column: \n')
print(df['class'])

print('\n Category codes of \'class\' column: \n')

df['class'].cat.codes
```

Run
Reset

pd.get_dummies()

- **Description:** A function that converts categorical values into new binary columns—one for each different category

Example:

index	rain
0	mild
1	mild
2	heavy
3	scattered
4	heavy
5	severe
6	severe
7	mild
8	heavy
9	scattered
10	scattered

index	rain_mild	rain_scattered	rain_heavy	rain_severe
0	1	0	0	0
1	1	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	0	1	1
5	0	0	0	1
6	0	0	0	1
7	1	0	0	0
8	0	0	1	0
9	0	1	0	0
10	0	1	0	0

LabelEncoder()

- **Description:** A transformer from `scikit-learn.preprocessing` that encodes specified categories or labels with numeric codes. Note that when building predictive models it should only be used on target variables (i.e., *y* data).

Example:

It can be used to normalize labels:

```
from sklearn.preprocessing import LabelEncoder

# Instantiate LabelEncoder()
encoder = LabelEncoder()

data = [1, 2, 2, 6]

# Fit to the data
encoder.fit(data)

# Transform the data
transformed = encoder.transform(data)

# Reverse the transformation
inverse = encoder.inverse_transform(transformed)

print('Data =', data)
print('\n Classes: \n', encoder.classes_)
print('\n Encoded (normalized) classes: \n', transformed)
print('\n Reverse from encoded classes to original: \n', inverse)

```

Run
Reset

It can be used to convert categorical labels into numeric:

```
from sklearn.preprocessing import LabelEncoder

# Instantiate LabelEncoder()
encoder = LabelEncoder()

data = ['paris', 'paris', 'tokyo', 'amsterdam']

# Fit to the data
encoder.fit(data)

# Transform the data
transformed = encoder.transform(data)

# New data
new_data = [0, 2, 1, 1, 2]

# Get classes of new data
inverse = encoder.inverse_transform(new_data)

print('Data =', data)
print('\n Classes: \n', list(encoder.classes_))
print('\n Encoded classes: \n', transformed)
print('\n New data =', new_data)
print('\n Convert new_data to original classes: \n', list(inverse))
```

Run
Reset

Key takeaways

There are many tools that data professionals can use to perform data cleaning on a wide range of data. The information you learn from missing data, outliers, and transforming categorical to numeric data will help you prepare datasets for further analysis throughout your career.

Mark as completed

Like **Dislike** **Report an issue**