# Handling Bad, Missing, and Duplicate Data

Chris Achard

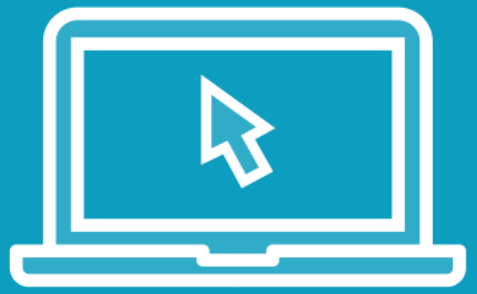
@nanohopdev www.nanohop.com

# Overview

## Cleaning Bad Data

- What is ”bad” data?
- Define your goal
- Drop, fill, or replace

# Demo



- Strip white space
- Replace bad data
- Fill missing data
- Drop bad data
- Drop duplicate data

# Stripping White Space

# Replacing Bad Data with NaN

# Filling Missing Data with a Value

# Dropping Rows of Data

# Identifying and Dropping Duplicate Data

# Review

```
df.title.str.strip()
```

- Strip whitespace from entire column

```
df.title.transform(lambda x: x.strip())
```

- Strip with a lambda function for greater flexiblity

```
from numpy import nan

df.replace({ 'colName': { 'value': nan } })
```

```
inplace=True
```

```
df.loc[df.col == 'value', ['col']] = nan
```

- Import nan from numpy and replace all occurences of a specific value in a column
- Remember to add inplace=True to change original data
- Filter with loc and fill with NaN

```
df.fillna(-1)
```

- Fill all NaN values in entire dataset

```
df.fillna(value={‘col': 0})
```

- Fill NaN values in a specific column

```
Inplace=True
```

- Use inplace=True to change original data

```
df.dropna()
```

◀ Drop rows with ANY NaN values

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

◀ Drop rows with ALL NaN values

```
df.dropna(thresh=15)
```

◀ Drop rows with AT LEAST a certain number of NaN values

```
df.dropna(subset=[’col1', ’col2'],
inplace=True)
```

◀ Only look at certain columns

```
df.drop_duplicates()
```

- Drop all duplicates

```
df.drop_duplicates(subset=['col1'])
```

- Drop duplicates if they match across certain columns

```
data.drop_duplicates(keep=False)
```

- Keep ‘first’, ‘last’ or False

```
data.loc[data.duplicated(subset=['col1', 'col2'], keep=False)]
```

- Find and see duplicates using .loc across specific columns