Putting it all together

CLEANING DATA IN PYTHON

Daniel Chen
Instructor

Putting it all together

- Use the techniques you've learned on Gapminder data
- Clean and tidy data saved to a file
  - Ready to be loaded for analysis!
- Dataset consists of life expectancy by country and year
- Data will come in multiple parts
  - Load
  - Preliminary quality diagnosis
  - Combine into single dataset

Useful methods

- `import pandas as pd`
- `df = pd.read_csv('my_data.csv')`
- `df.head()`
- `df.info()`
- `df.columns`
- `df.describe()`
- `df.column.value_counts()`
- `df.column.plot('hist')`

Data quality

```
def cleaning_function(row_data):
        # data cleaning steps
        return ...

df.apply(cleaning_function, axis=1)

assert (df.column_data > 0).all()
```

Combining data

- `pd.merge(df1, df2, …)`
- `pd.concat([df1, df2, df3, …])`

Let's practice!

CLEANING DATA IN PYTHON

Initial impressions of the data

CLEANING DATA IN PYTHON

Daniel Chen
Instructor

Principles of tidy data

- Rows form observations
- Columns form variables
- Tidying data will make data cleaning easier
- `melt()` turns columns into rows
- `pivot()` will take unique values from a column and create new columns

Checking data types

```
df.dtypes
```

```
df['column'] = pd.to_numeric(df['column'])
```

```
df['column'] = df['column'].astype(str)
```

Additional calculations and saving your data

```
df['new_column'] = df['column_1'] + df['column_2']
```

```
df['new_column'] = df.apply(my_function, axis=1)
```

```
df.to_csv('my_data.csv')
```

Let's practice!

CLEANING DATA IN PYTHON

Final thoughts

CLEANING DATA IN PYTHON

Daniel Chen
Instructor

You've learned how to...

- Load and view data in `pandas`
- Visually inspect data for errors and potential problems
- Tidy data for analysis and reshape it
- Combine datasets
- Clean data by using regular expressions and functions
- Test your data and be proactive in finding potential errors

Let's practice!

CLEANING DATA IN PYTHON