# Data Cleaning in Pandas - Introduction

[(https://github.com/learn-co-curriculum/dsc-introduction-pandas-etl)](https://github.com/learn-co-curriculum/dsc-introduction-pandas-etl) [(https://github.com/learn-co-curriculum/dsc-introduction-pandas-etl/issues/new/choose)](https://github.com/learn-co-curriculum/dsc-introduction-pandas-etl/issues/new/choose)

## Introduction

In this section, you will learn invaluable skills that will form the foundation of your data processing work. Before you can apply machine learning algorithms or do interesting analyses, you often must clean and transform your data into a suitable format. Such initial data wrangling processes are often referred to as Extract Transform Load (ETL). Our primary tool of choice for performing ETL and basic analyses will be the Pandas package.

## Why ETL?

ETL is an essential first step to data analysis and data science. It also will form the foundation for exploratory data analysis. Often, you will be thrown a dataset that you have little to no information about. In these cases, your first step is to explore the data and get familiar with it. What are the columns? How many observations do you have? Are there missing values? Any outliers? If we have user-level data, how can we explore aggregate trends along features like gender, race, or geography? All of these can be answered by applying ETL to transform raw datasets into alternative useful views.

## Quick ETL Example

While you'll see complete examples and explanations for all of these techniques (and more), here's a quick preview of some ETL techniques covered in this section! For more details, continue on to future lessons!

## Import data

```python
import pandas as pd
df = pd.read_csv('Yelp_Reviews.csv', index_col=0)
df.head()
```

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|

? Help

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|
| **1** | pomGBqfbxcqPv14c3XH-ZQ | 0 | 2012-11-13 | 0 | dDl8zu1vWPdKGihJrwQbpw | 5 |
| **2** | jtQARsP6P-LbkyjbO1qNGg | 1 | 2014-10-23 | 1 | LZp4UX5zK3e-c5ZGSeo3kA | 1 |
| **4** | Ums3gaP2qM3W1XcA5r6SsQ | 0 | 2014-09-05 | 0 | jsDu6QEJHbwP2Blom1PLCA | 5 |
| **5** | vgfcTvK81oD4r50NMjU2Ag | 0 | 2011-02-25 | 0 | pfavA0hr3nyqO61oupj-lA | 1 |
| **10** | yFumR3CWzpfvTH2FCthvVw | 0 | 2016-06-15 | 0 | STiFMww2z31siPY7BWNC2g | 5 |

```
df.shape
```

? Help

```
(2610, 9)
```

# Apply lambda functions

```
df['Review_Word_Length'] = df['text'].map(lambda x: len(x.split()))
df.head()
```

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|
| **1** | pomGBqfbxcqPv14c3XH-ZQ | 0 | 2012-11-13 | 0 | dDl8zu1vWPdKGihJrwQbpw | 5 |
| **2** | jtQARsP6P-LbkyjbO1qNGg | 1 | 2014-10-23 | 1 | LZp4UX5zK3e-c5ZGSeo3kA | 1 |
| **4** | Ums3gaP2qM3W1XcA5r6SsQ | 0 | 2014-09-05 | 0 | jsDu6QEJHbwP2Blom1PLCA | 5 |
| **5** | vgfcTvK81oD4r50NMjU2Ag | 0 | 2011-02-25 | 0 | pfavA0hr3nyqO61oupj-lA | 1 |

Help

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|
| 10 | yFumR3CWzpfvTH2FCthvVw | 0 | 2016-06-15 | 0 | STiFMww2z31siPY7BWNC2g | 5 |

```
df.shape # Previously this was (2610, 9), now we have added a column
```

```
(2610, 10)
```

# Group data

```
df.groupby('business_id')['stars'].mean().head()
```

```
business_id
-050d_XIor1NpCuWkbIVaQ    5.0
-0qht1roIqleKiQkBLDkbw    1.0
-3zffZUHoY8bQjGfPSoBKQ    5.0
-6tvduBzjLI1ISfs3F_qTg    5.0
-9nai28tnoylwViuJVrYEQ    5.0
Name: stars, dtype: float64
```

# Check for duplicates

Check how many we have:

```
df.duplicated().value_counts()
```

```
False    2277
True      333
dtype: int64
```

Visually inspect them:

```
# Use keep=False to keep all duplicates and sort_values to put duplicates next to each otl
df[df.duplicated(keep=False)].sort_values(by='business_id')
```

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|
| **1729** | -GY2fx-8udXPY8qn2HVBCg | 0 | 2016-08-30 | 0 | yQ6P1_CvM94wMLYw1T0UWA | 5 |
| **1729** | -GY2fx-8udXPY8qn2HVBCg | 0 | 2016-08-30 | 0 | yQ6P1_CvM94wMLYw1T0UWA | 5 |
| **754** | -LRlx2j9_LB3evsRRcC9MA | 0 | 2017-10-07 | 0 | kUqPsZmWwLIMSstGHhWssA | 5 |
| **754** | -LRlx2j9_LB3evsRRcC9MA | 0 | 2017-10-07 | 0 | kUqPsZmWwLIMSstGHhWssA | 5 |
| **2767** | -MKWJZnMjSit406AUKf7Pg | 0 | 2015-01-03 | 2 | rJhrQD3-b9GjTso0dxIkwg | 1 |
| **...** | ... | ... | ... | ... | ... | ... |
| **2193** | zKw09ftu1730wEIZBZPoFg | 3 | 2015-01-04 | 0 | JV-yxKxMFp-d0rLDc_2_6w | 5 |
| **496** | zg5rJfgT4jhzg1d6r2twnA | 0 | 2014-06-21 | 0 | Zbj0HgdN3AT4l-mbH-EfjA | 3 |

Help

| | business_id | cool | date | funny | review_id | stars |
|---|---|---|---|---|---|---|
| **496** | zg5rJfgT4jhzg1d6r2twnA | 0 | 2014-06-21 | 0 | Zbj0HgdN3AT4l-mbH-EfjA | 3 |
| **988** | ziv21pDfyrgdhlrlNIgDfg | 0 | 2016-08-11 | 0 | fus9odxu9bjE2lSxfwNfdw | 5 |
| **988** | ziv21pDfyrgdhlrlNIgDfg | 0 | 2016-08-11 | 0 | fus9odxu9bjE2lSxfwNfdw | 5 |

666 rows × 10 columns

# Remove duplicates

```
df = df.drop_duplicates()
df.shape # Previously this was (2610, 10), now we have dropped duplicate rows
```

```
(2277, 10)
```

# Recheck for duplicates

```
df.duplicated().value_counts()
```

```
False    2277
dtype: int64
```

```
# Duplicates should no longer exist
df[df.duplicated(keep=False)].sort_values(by='business_id')
```

| | business_id | cool | date | funny | review_id | stars | text | useful | user_id | Review_Word |
|---|---|---|---|---|---|---|---|---|---|---|

# Create pivot tables

```
# This transforms the data into a person by person spreadsheet and what stars they gave va
       are NaN (null or missing) because people only review a few restaurants of tl
```

Help

```
usr_reviews = df.pivot(index='user_id', columns='business_id', values='stars')
usr_reviews.head()
```

| business_id | -050d_Xlor1NpCuWkbIVaQ | -0qht1roIqleKiQkBLDkbw | -3zffZUI |
|---|---|---|---|
| user_id | | | |
| -0biHfjE0soSptbU5G3nug | NaN | NaN | NaN |
| -2K0yp7IBT_JUOzGkpdJ_g | NaN | NaN | NaN |
| -Opvc9hAWIlZSSPDUsD7NA | NaN | NaN | NaN |
| -Zdxj4wuj4D_899B7tPE3g | NaN | NaN | NaN |
| -_iULENf28RbqL2k0ja5Xw | NaN | NaN | NaN |

5 rows × 2192 columns

# Summary

In this brief introduction, you learned the acronym ETL and got to preview a few examples of ETL processes using pandas. In the upcoming lessons you'll get a much richer understanding of these and other techniques for wrangling your data!

How do you feel about this lesson?

Have specific feedback?

[Tell us here!](https://github.com/learn-co-curriculum/dsc-introduction-pandas-etl/issues/new/choose) (https://github.com/learn-co-curriculum/dsc-introduction-pandas-etl/issues/new/choose)

? Help