

Exercise 2.7: Data Analysis and Visualization in Django

Learning Goals

- Work on elements of two-way communication like creating forms and buttons
- Implement search and visualization (reports/charts) features
- Use QuerySet API, DataFrames (with pandas), and plotting libraries (with matplotlib)

Reflection Questions

1. Consider your favorite website/application (you can also take CareerFoundry). Think about the various data that your favorite website/application collects. Write down how analyzing the collected data could help the website/application.

Analyzing data is key to any website, it is perhaps the best way to differentiate your product from other by providing a different analysis and subsequently a different view and user experience when using your application. Not only can it be a creative outlet, but if data is used to drive design choices it can help to achieve the desired user experience and optimize website usage.

2. Read the Django [official documentation on QuerySet API](#). Note down the different ways in which you can evaluate a QuerySet.
 - Iteration
 - Indexing
 - Slicing
 - Counting
 - Existence checks
 - Aggregation
3. In the Exercise, you converted your QuerySet to DataFrame. Now do some research on the advantages and disadvantages of QuerySet and DataFrame, and explain the ways in which DataFrame is better for data processing.

Advantages of QuerySet:

- Built-in to Django making it easy to use for Django Applications
- API makes it easier to write complex queries
- Optimized to handle large datasets
- Specifically designed for relational databases

Disadvantages of QuerySet:

- Only works with relational databases
- Can have speed issues with large datasets
- Best suited to work with Django Applications only

Advantages of DataFrame:

- Can work with a variety of data sources

- Optimized to work with in-memory data, making it fast for small to medium sized datasets
- Built to easily manipulate and analyze tabular data

Disadvantages of DataFrame:

- Memory intensive, potentially causing problems for large datasets
- Compared to QuerySet:
 - Can be more difficult when using complex queries
 - Can be slower with relational datasets (needs to load data into memory before processing)

QuerySet seems to be the superior choice when specifically working with relational databases in a Django application. However, DataFrame ultimately can be justified in many other use cases due to its flexibility, power, support for a variety of data sources, and performance when it comes to processing data.