### Exercise 2.7: Data Analysis and Visualization in Django

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 the data can help clarifying user’s needs leading to improved application/website’s performance and user experience.

1. Read the [Django official documentation on QuerySet API](https://docs.djangoproject.com/en/3.2/ref/models/querysets/). Note down the different ways in which you can evaluate a QuerySet.

* QuerySet can be interated also asynchronously. (async for in)
* Slicing a QuerySet that has been evaluated also returns list.
* len() returns the length of the result list.
* list() ex. List(Entry.objects.all()) would force to return list instead of tuple.
* bool() returns True if at least one result is met the condition, otherwise false.

1. 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:

QuerySet is a buit-in data structure in Django so it can be used easily in Django applications. QuerySet is designed specifically for working with data stored in relational database and facilitates to write complex queries.

Disadvantages of QuerySet:

QuerySet can only be used with relational database and it is not suitable for working with databases with large amount entries as well as use in non-Django applications.

Advantages of DataFrame:

DataFrame is a versatile structure that can be used to work with a variety of data sources, including CSV files, Excel spreadsheets and databases. DataFrame is optimized for working with in-memory data, so it can be very fast and efficient for analyzing small to medium datasets.

Disadvantages of DataFrame:

DataFrame can be memory-intensive and slow, especially working with large datasets which can lead to performance issues. Also DataFrame can be more difficult to use than QuerySet for making queries involving multiple tables and joins.

Conclusion:

While QuerySet is powerful when working with relational database in Django applications, DataFrame offers more flexibility.