

Graphing Dino Fun World Assignment

Purpose

In this assignment, you will be creating visualizations based on data from the Dino Fun World amusement park dataset (same as the previous assignment). You will be extracting relevant information from the database and interpreting it by generating pie, bar, line, and boxplot charts.

Technical Requirements

If you choose to work on your assignment locally, you can use the following versions:

- Python 3.12
- Sqlite3
- Pandas == 1.5.3
- Matplotlib == 3.8.3
- Numpy == 1.26.4

Assignment Description

Impressed by your previous work, the administrators of Dino Fun World have asked you to create some charts that they can use in their next presentation to upper management. The data used for this assignment will be the same as the data used for the previous assignment.

The administrators would like you to create four graphs:

Graph 1: A pie chart depicting visits to thrill ride attractions.

- For this question, display the pie chart in the notebook and print the data used to create the pie chart as a list of lists (ex: [['Ride 1', 10], ['Ride 2', 100], ...])

Graph 2: A bar chart depicting total visits to food stalls.

- Please query attractions with category equal to "Food"
- For this question, display the bar chart in the notebook and print the data used to create the bar chart as a list of lists (ex: [['Stall 1', 10], ['Stall 2', 50], ...])

Graph 3: A line chart depicting attendance at the newest ride, Atmosfear over the course of the day.

- For this question, use data from the table "sequences". You can assume that all activity sequences are aligned by time (i.e., the first node of all sequences occurred at the same time) and are of the same length.
- You only need to consider the first 16 hours of records in the sequences, which is considered the open hours of the park in one day. Specifically, because an activity sequence from the "sequences" table was updated/recorded every 5 minutes, you need to extract the first 192 items ($16h * 60m / 5m = 192$).
- Display the line chart in the notebook and print the data used to create the chart as a list of lists or tuple lists
 - The first item in the list is irrelevant, but you can put in some meaningful information. The second item is the count of visits at that moment. For example, your output should look like this (in Python syntax; not relevant to the correct answer): `[(0, 0), (1, 7), (2, 3), ..., (190, 4), (191, 5)]`.

Graph 4: A box-and-whisker plot depicting total visits to the park's Kiddie Rides.

- For this question, display the box plot in the notebook and print the number of visits to each ride as a list (ex: `[3, 4, 5, 6, ...]`). The data list must be ordered by AttractionID (a field in the table "attraction").

Directions

Accessing Ed Lessons

You will complete and submit your work through Ed Lessons. Follow the directions to correctly access the provided workspace:

1. Go to the Canvas Assignment, "**Submission: Graphing Dino Fun World Assignment**".
2. Click the "**Load Submission...in new window**" button.
3. Once in Ed Lesson, select the assignment titled "**Graphing Dino Fun World Assignment**".
4. Review the resources provided in the demonstration.
5. When ready, click on the code challenge and start working in the notebook titled "**Assignment2.ipynb**".

Assignment Directions

The database provided by the park administration is formatted to be readable by any SQL database library. The course staff recommends the `sqlite3` library. The database contains three tables, named 'checkin', 'attractions', and 'sequences'. The database file is named 'dinofunworld.db' and is available in the '/course/data/CSE-578/dinofunworld.db' path.

Note: Please note that the database file is accessible through the learner submission workspace, which requires establishing a connection with the database. For downloading the dataset and potentially working locally, refer to the overview document page in your course.

The information contained in each of these tables is listed below:

checkin:

- The check-in data for all visitors for the day in the park. The data includes two types of check-ins: inferred and actual checkins.
- Fields: visitorID, timestamp, attraction, duration, type

attraction:

- The attractions in the park by their corresponding AttractionID, Name, Region, Category, and type. Regions are from the VAST Challenge map such as Coaster Alley, Tundra Land, etc. Categories include Thrill rides, Kiddie Rides, etc. Type is broken into Outdoor Coaster, Other Ride, Carousel, etc.
- Fields: AttractionID, Name, Region, Category, type

sequences:

- The check-in sequences of visitors. These sequences list the position of each visitor to the park every five minutes. If the visitor has not entered the park yet, the sequence has a value of 0 for that time interval. If the visitor is in the park, the sequence lists the attraction they have most recently checked in to until they check in to a new one or leave the park.
- Fields: visitorID, sequence

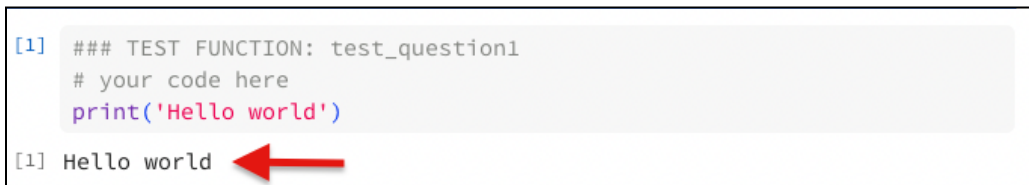
Using the provided data, create the four visualizations that the administration requested: the pie chart, bar chart, line chart, and box-and-whisker plot.

Submission Directions for Assignment Deliverables

This assignment will be auto-graded. You must complete and submit your work through Ed Lesson's code challenge to receive credit for the course:

1. In order for your answers to be correctly registered in the system, you must place the code for your answers in the cell indicated for each question.

- a. You should submit the assignment with the output of the code in the cell's display area. The display area should contain only your answer to the question with no extraneous information, or else the answer may not be picked up correctly.
 - b. Each cell that is going to be graded has a set of comment lines (ex: `### TEST FUNCTION: test_question1`) at the beginning of the cell. **This line is extremely important and must not be modified or removed.**
2. After completing the notebook, run each code cell individually or click "**Run All**" at the top to print the outputs.



The screenshot shows a Jupyter Notebook cell. The code area contains three lines: `### TEST FUNCTION: test_question1`, `# your code here`, and `print('Hello world')`. The output area below shows `[1] Hello world`. A red arrow points to the output text.

3. When you are ready to submit your completed work, click on "**Mark**" at the bottom right of the screen.
4. You will know you have successfully completed the assignment when feedback appears for each test case with a score.
5. If needed: to resubmit the assignment in Ed Lesson
 - a. Edit your work in the notebook
 - b. Run the code cells again
 - c. Click "**Mark**" at the bottom of the screen

Your submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

Evaluation

There are four parts in the grading, and each part has one test case where the total number of points for all parts is 50. If the submission is correct, you will see "The data used for the chart is correct. The plot is a valid chart." with scores for each part. If your output data is correct but the graph is not, you will receive a partial score. **The auto-grader first validates your output data, and if it is correct, it proceeds to evaluate the correctness of the graph.** If the submission fails, the grader will return the corresponding error messages.