

Programming Assignment 4 - Dino Fun World Time Series Analysis

Assignment Description

The administrators of Dino Fun World, a local amusement park, have asked you, one of their data analysts, to perform three data analysis tasks for the park. These tasks will involve understanding, analyzing, and graphing attendance data that the park has provided for you to use in the form of a database.

Question 1: The park's administrators are worried about the attendance at the ride 'Atmosfear' in the data window. To assuage their fears, they have asked you to create a control chart of the total attendance at this ride. Using the provided data, create a control chart displaying the attendance, mean, and standard deviation bands at one and two standard deviations.

Question 2: Some of the park's administrators are having trouble interpreting the control chart graph of 'Atmosfear' attendance, so they ask you to also provide a moving average chart of the attendance in addition to the control chart created in Part 1. In this case, they request that you use 50 samples for the size of the moving average window.

Question 3: In order to have options concerning the graphs presented, the park's administrators also ask you to provide a 50-sample moving average window with the average computed with exponential weighting (i.e., an exponentially weighted moving average) over the same 'Atmosfear' attendance data.

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 read only directory of the Jupyter Notebook environment

(i.e., readonly/dinofunworld.db). It can also be accessed by selecting File > Open > readonly/dinofunworld.db.

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 data provided, perform the required analyses and create the requested charts.

Technical Requirements

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

- Python 3.6
- Sqlite3
- Pandas == 0.23.3
- Matplotlib == 2.2.2
- Numpy == 1.13.3


Submission Directions for Assignment Deliverables

This assignment will be auto-graded. We recommend that you use Jupyter Notebook in your browser to complete and submit this assignment. 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. In addition, 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.

Each cell that is going to be graded has a set of comment lines at the beginning of the cell. These lines are extremely important and must not be modified or removed. (Graded Cell and PartID comments must be in the same line for proper execution of code.)

Please execute each cell in Jupyter Notebook before submitting.

```
► In [1]: # Graded Cell, PartID: NDnou
          # Question 1: What is the most popular attraction to visit in the park?
          # Notes: Your output should be the name of the attraction.
          print('Hello World')
```

Hello World 

Evaluation

There are three parts in the grading, and each part has one test case where the total number of points for all parts is 30. If some part of your data is incorrect, you will get a score of 0.0. If the submission fails, we will return the corresponding error messages. If the submission is correct, you will see "Correct" with 10 points for each part.