CSE 5520 Fall 2021
Homework 6 (Due 11:59 pm, Sunday, October 24, 2021 at HuskyCT)

Visualization of KDE and Monte Carlo method

This homework is to help you practice with some data visualization fundamentals. You are expected to use these visualization techniques and others in your final project. You are required to do this exercise in Python. All plots/graphs must have titles and x-y coordinate tick labels.

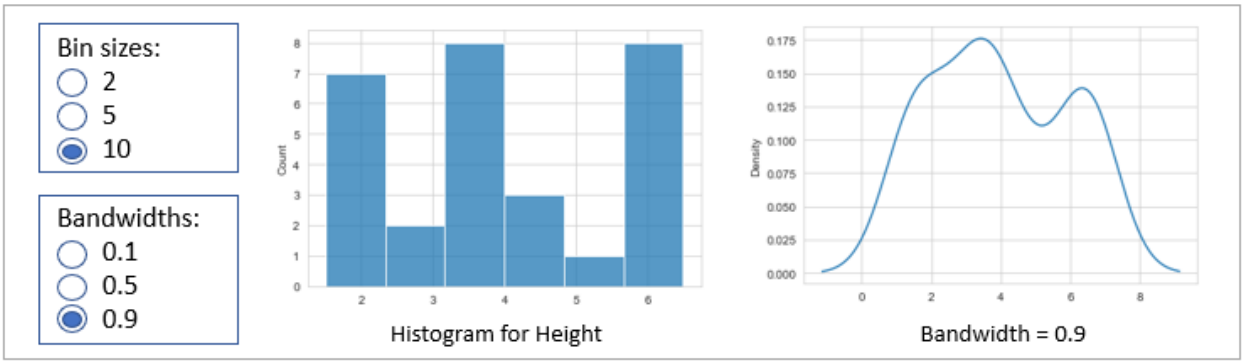
Part 1: Dashboard

This part is to help you get started with your term project by gently introducing Dashboard creation. You were given a template code that creates a Dashboard webpage. Decide one or more data sets of your choice, ideally, related to your term project. If you do not have one yet, you can use the NBA/LOAD data sets used in previous Homework.

Step 1. Using the data set of your choice, you let user create histograms at different bin sizes.

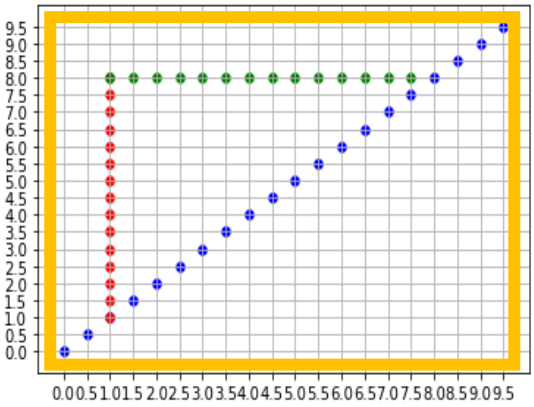
Step 2. Apply Kernel Density Estimation (KDE) to convert your histogram into a KDE plot. Practice with alternative bandwidths that contrast impact of using different bandwidths.

Step 3. Create a simple Dashboard that allows the user to choose multiple (minimum three) different KDE plots using a design similar to the one given below. Note that the panel on the leftmost is to enable user to choose different histogram bin sizes/bandwidths. KDE plot should change based on the choices. The picture given below is only for illustration purpose. Depending on your data set(s) the exact look and feel will vary. You can have your own design. In providing bandwidth options, make it obvious that your system suggests bandwidths calculated based on "rule of thumb" discussed in class. There are multiple ways you can indicate which ones are suggested ones based on "rule of thumb" (e.g., annotate bandwidths with "Recommended 1" and "Recommended 2.).



Step 4. By using a Jupyter markdown cell, provide a brief explanation for how to use your visualization tool. It should be brief. This is just to report your design decisions in creating your visualization framework. Ideally a good visualization web app should be so obvious that no instruction for how to use it other than meaningful titles/labels for the components.

Part 2: Monte Carlo method



Your goal is to find the approximate area ratio (AR) surrounded by blue, red and green dots using Monte Carlo sampling method. Note that here AR = 120/400 = 0.3 where 120 is inclusive of the surrounding-colored dots (except 5 blue dots external to the triangular area).

Step 1. Write a code placing 1000 dots using uniform random distribution for both x and y values within the yellow boxed region that are smaller than the ones appearing on the grid so that you can see where the randomly generated dots land. Dots must appear only on the grid intersection. In this step, you are only showing the placement of dots.

Step 2. Write a code computing AR. One issue you have to think is how to count multiple dots that can potentially land on the same spot. Should each landing be counted or not for the calculation?

Step 3. Repeat Step 2 1000 times to estimate μ and σ for ARs. Since printing 1000 μ's and σ's is cumbersome, create a histogram for the generated values. Perform KDE and show the histogram and its KDE side by side (Do not use the same seed for each run).

Step 4. Repeat Steps 1 – 3 for 500 dots.

Step 5. Discuss your finding when you compare the cases of using 500 dots and 1000 dots using a markdown cell.

You upload your Jupyter notebook in HuskyCT. The file name should be of the following format: HWn_Doe where n is the homework number and Doe denotes the last name.

HWs and Projects, 5% penalty for one day late submission. No acceptance after 5 days late. Extension is allowed only with the supporting medical record.