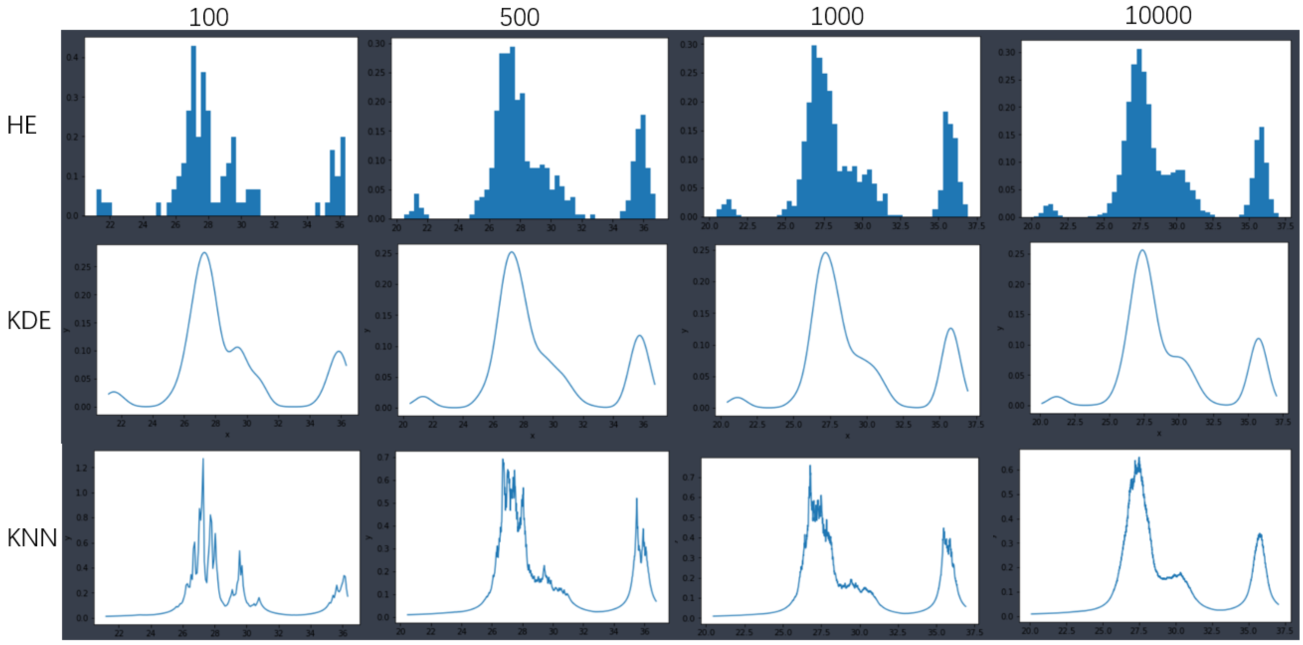
report - assignment 1 : non-parametric density estimation

Introduction

In this assignment, three non-parametric density estimation algorithms(histogram estimation, kernel density estimate and the nearest neighbor method) are used to estimate the distribution of a given data set. To choose the best parameters for each algorithm, several evaluation methods are also put into use. The sequence of my report would follow the sequence of the tasks.

Task 1: variation of the number of data used

I put 12 plots(4 number of data * 3 algorithms) in a table for comparison. Here I set bins = 50 for histogram method, h = 0.5 for KDE, K = number of data * 0.05 for KNN.

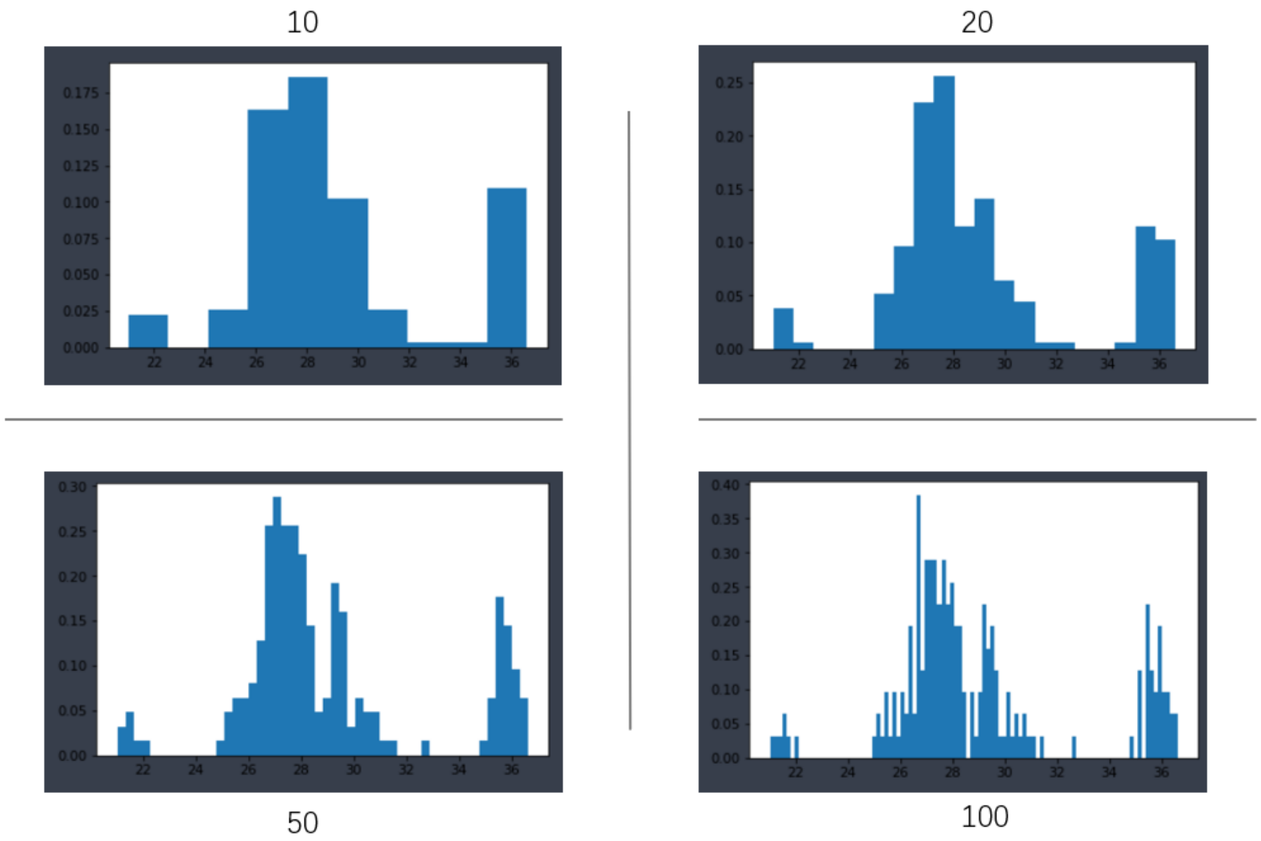


It is obvious that the more data we use, the better quality our estimation is(especially for histogram estimation and the k nearest neighbor method). We can attribute it to lower contingency with more data. Less data means not only rougher curve, but also a possibility of loss of the peak.

In the following part, I use **num = 200** for exploration. Too big number would affect both efficiency and simplicity of parameter tune, since it works well when number is big and we can not easily judge which parameter choice is better.

Task 2: variation of the number of bins for histogram method

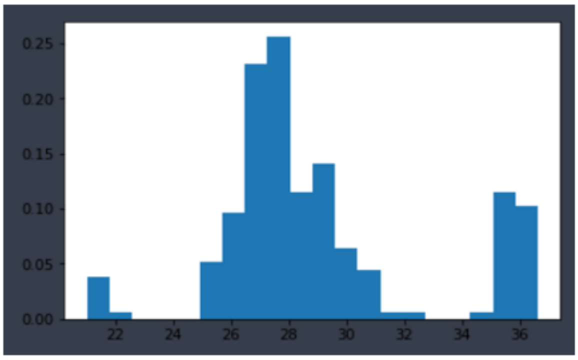
Below shows the influence of estimation quality while changing the number of bins for histogram method.



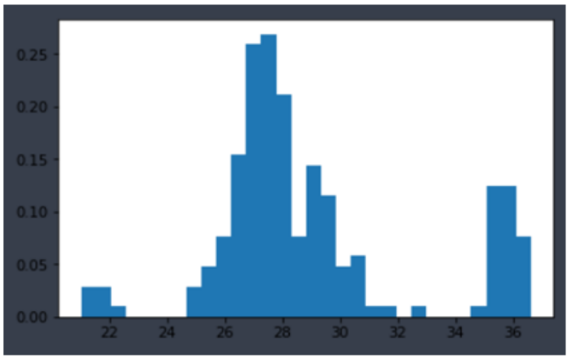
When the number of bins is too small(such as 10), we can clearly see several blocks rather than a approximate curve. So the information it provides is limited and the quality of estimation is not good. However, when the number of bins is too large(such as 100), we find that result is severely affected by randomness, and the function becomes more discontinuous. In other word, we are under the impact of contingency. So it is not a good choice either. What we should do is finding a moderate number to improve the quality of estimation.

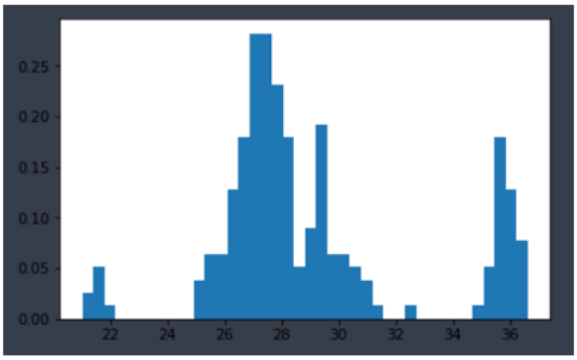
Here comes the question that how to pick the best (or good) choice for the number of bins. The first method is narrowing the scope of number gradually. For example, both 20 and 50 perform not bad in the picture above, and we can guess the best choice is between them. Therefore, we can try [20, 30, 40, 50] for comparison.

20

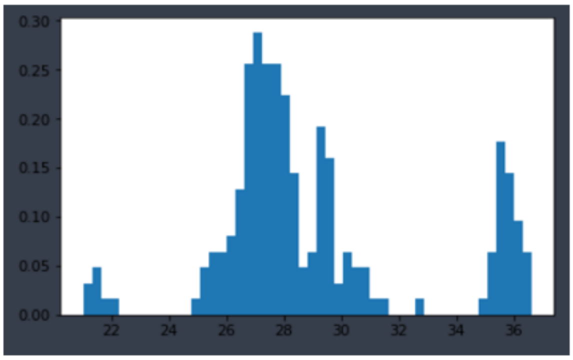


30



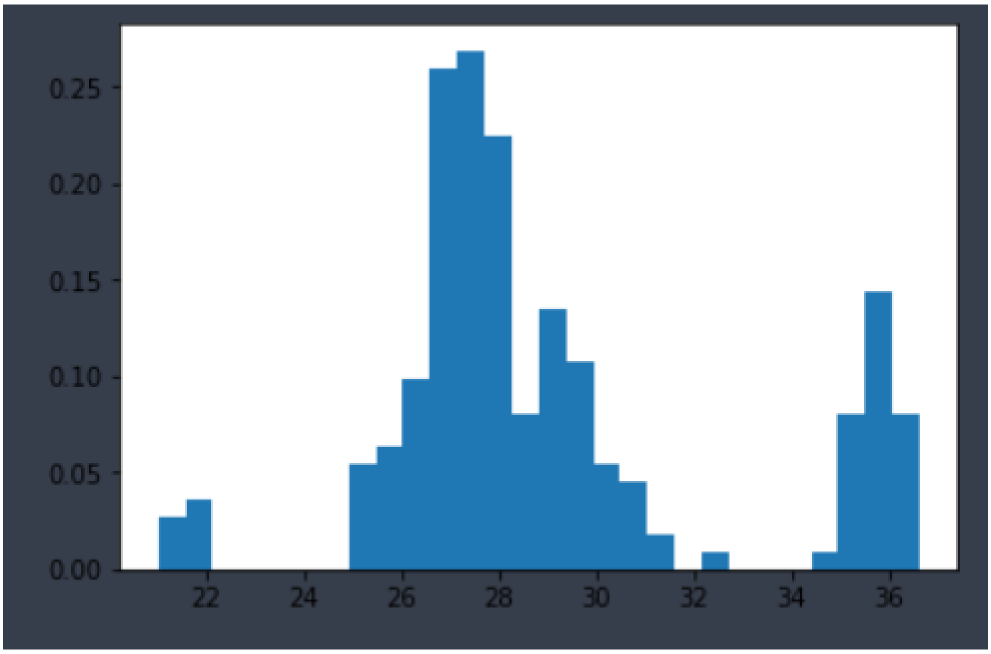


40



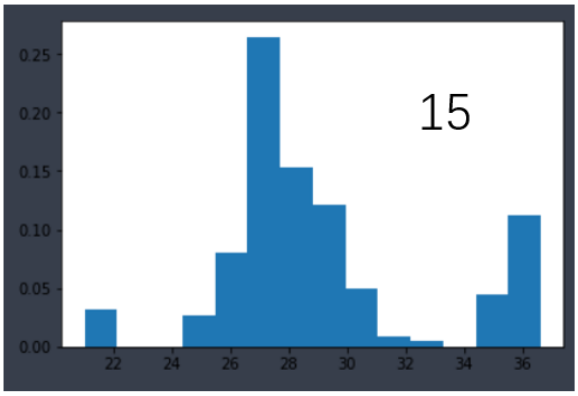
50

By contrast, 20 and 30 perform better and we can narrow the scope to [20,30]. Repeating this method, we can find the best answer. In my experiment, 28 is a relatively reasonable choice as shown in figure below.



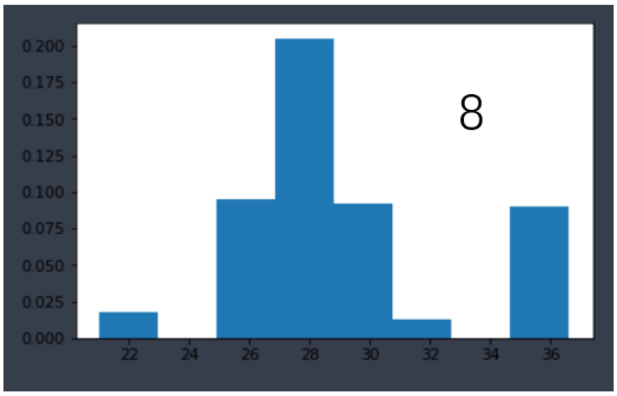
One of the drawback of this method is that we judge the performance subjectively, which may lead to some mistakes especially when performance is close. But it can help us find a good but probably not the best parameter quickly.

Of course, we can search for some reference to help us make decision. But different methods perform differently when condition changes, so it is better to use them to narrow the scope instead of making decision. For example, here display two methods performing not well in this problem.



Square-root choice

$k = \lceil \sqrt{n} \rceil$



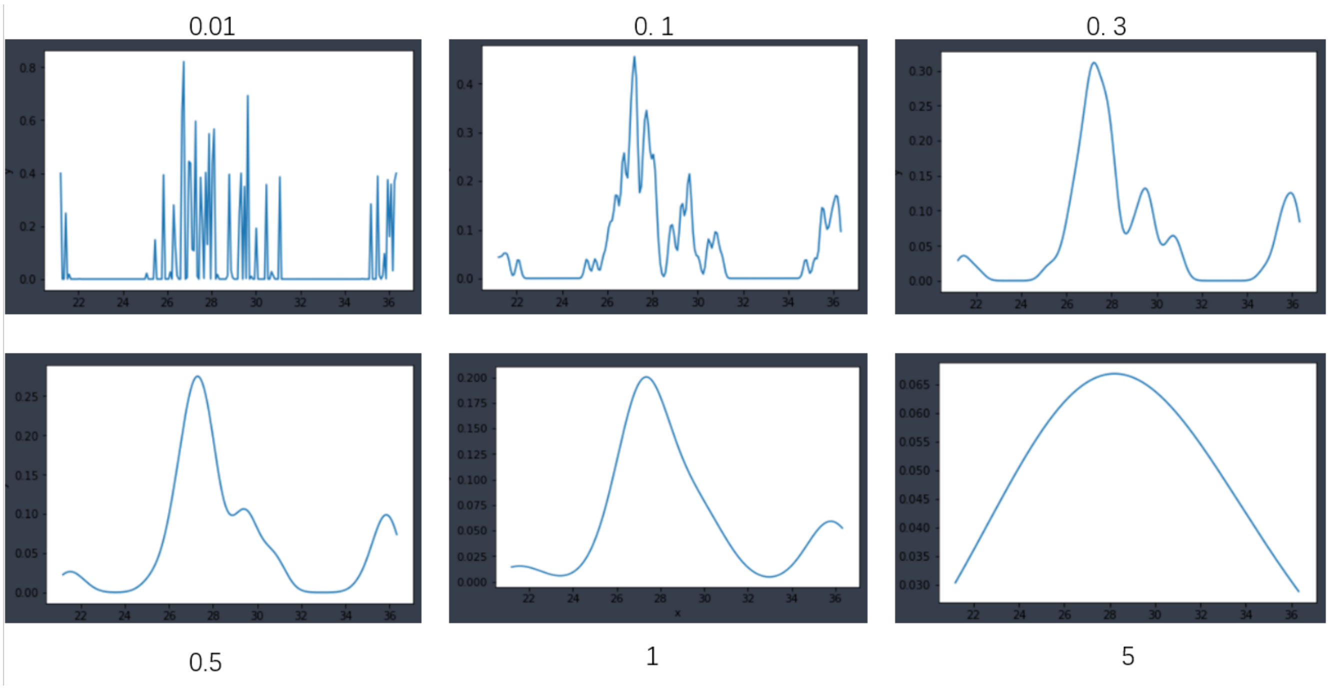
Sturges' formula

$k = \lceil \log_2 n \rceil + 1$

In addition, some methods like cross validation may work to choose the best bins number. They will be introduced in Task 3.

Task 3: variation of the *h* for Gaussian kernel method

First, we explore the influence of h for Gaussian kernel method in a large scale. I set **num = 100** and vary the h, and the result shows below:

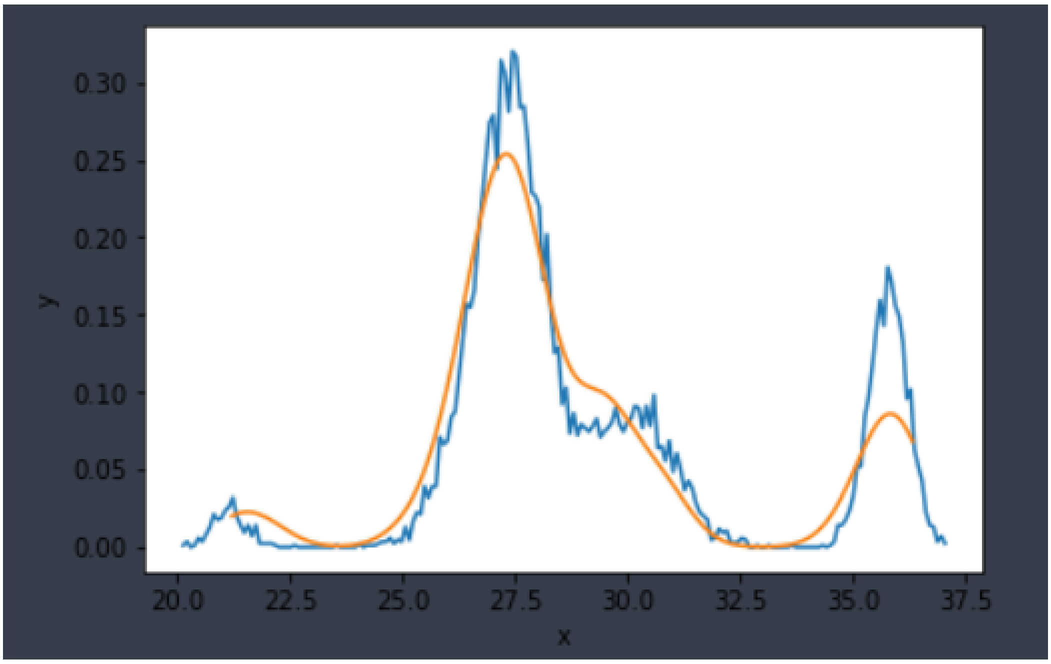


We can find that setting the h too small doesn't provide much more information than the raw data, while setting it too large oversmooths the data, which makes it mostly look like the kernel function instead of the origin function. Therefore,we need to choose a moderate number like what we do in TASK2.

a tricky method

Obviously it is not reasonable to use the true distribution to evaluate the quality of estimation. However, we can simulate the true distribution by histogram method with a large data number. That is to say, we take the estimation of the histogram method with a large data number as the true distribution, and thus we can use loss function such as MSE, MAE or Log-cosh to evaluate the quality of our estimation. The reason why we can do this is that frequency is approaching probability with a large data number.

I try MSE as the evaluation method and iterate by adding 0.01 to h each time.In this method, **h = 0.62** is the best solution as the figure shows below. It is higher than expected but acceptable.

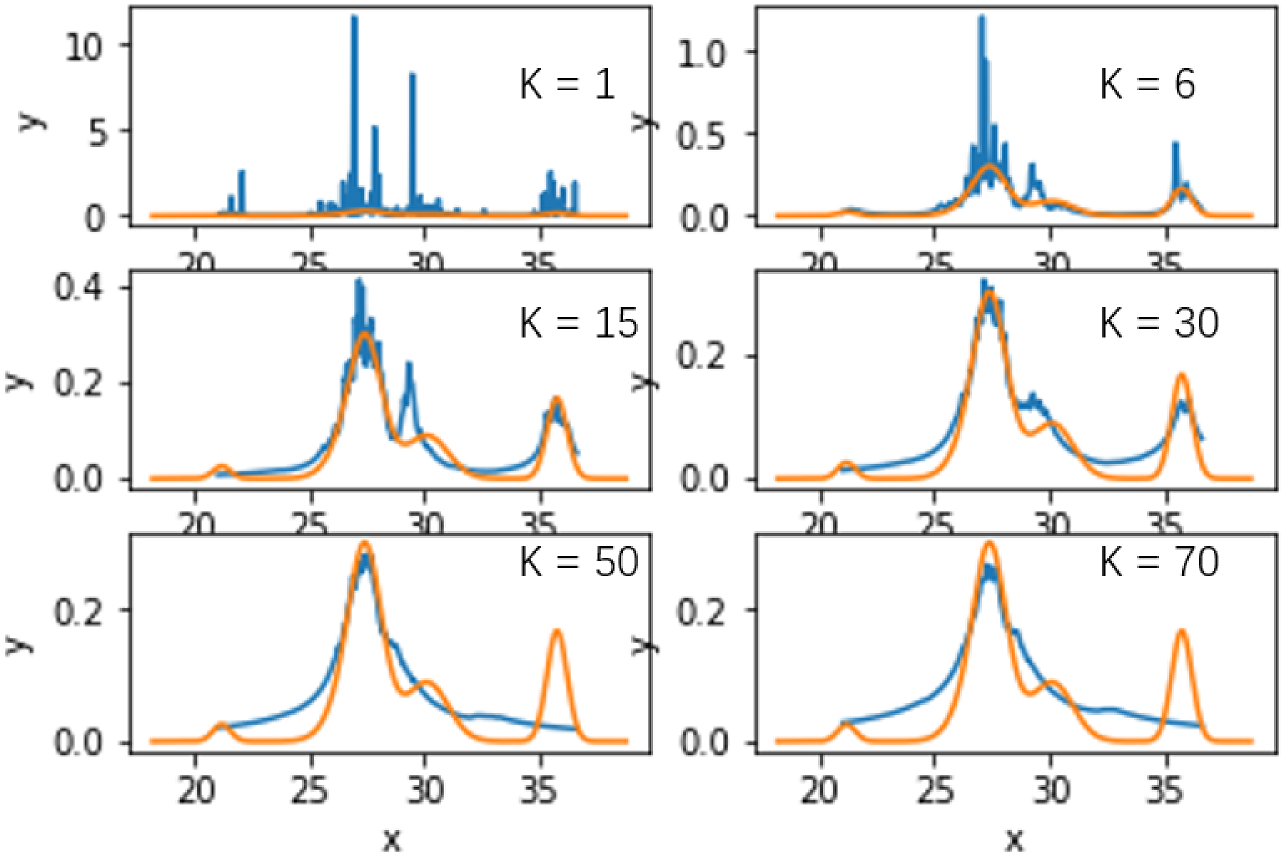


KFold cross-validation

We can split our data into train set and test set, and use the test set to evaluate the quality of estimation. Since number of sample is small, we can divide our data into 4 parts, taking three of them as train set and one of them as test set and repeating four times(KFold with 4 n_splits). However, this method does not work well in my algorithm, which shows that h = 1 is the best solution. I guess I have chosen a wrong evaluation standard, and it is needed to explore more.

Task 4: variation of the *K* for KNN method

Pictures of KNN method with a variation of K show below, in which case I set data number = 100:



When K is very small(such as 1), we can find that the curve is rather sharp, which can barely demonstrate the true distribution. When K grows larger(such as 15), the figure matches the true distribution better and better. However, when K becomes a very large number(such as 50 and 70), the second peak(for x between 34 and 36) can not be seen anymore, which is not what we want either. Again, we should try methods introduced in Task2 and Task3 to help us decide the moderate number.

Prove: the nearest neighbor method does not always yield a valid distribution

- A theoretical method

  We have known that (K/N) is a constant in the formula **Y = K/(N*V)**, so the only part that matters is **V.**

  For X bigger than biggest number of set, the K nearest neighbor for X is the same, and the distance between X and its K nearest neighbor is approximately in positive correlation with X itself. That means we would face a problem like that:

  $$\int \frac{m}{kx+b}dx$$

  It turns out to be a logarithm function, and the integral diverges, which can never be 1.

- An empirical method

  Compare the KNN function to the true distribution and it is easy to find that the KNN function has a bigger area than the true distribution(it is especially obvious for K = 6). If we do not trust our eyes, I calculate the integration for K = 1 and K = 6 between (min(sample_data), max(sample_data)).

  When K = 1, the answer is 4.458 and when K = 6, the answer is 1.279, both of which are larger than 1. When K grows larger, the answer becomes closer to 1. Of course, if we calculation the integration between (-inf, inf), we will get no answer as proved in the theoretical method.