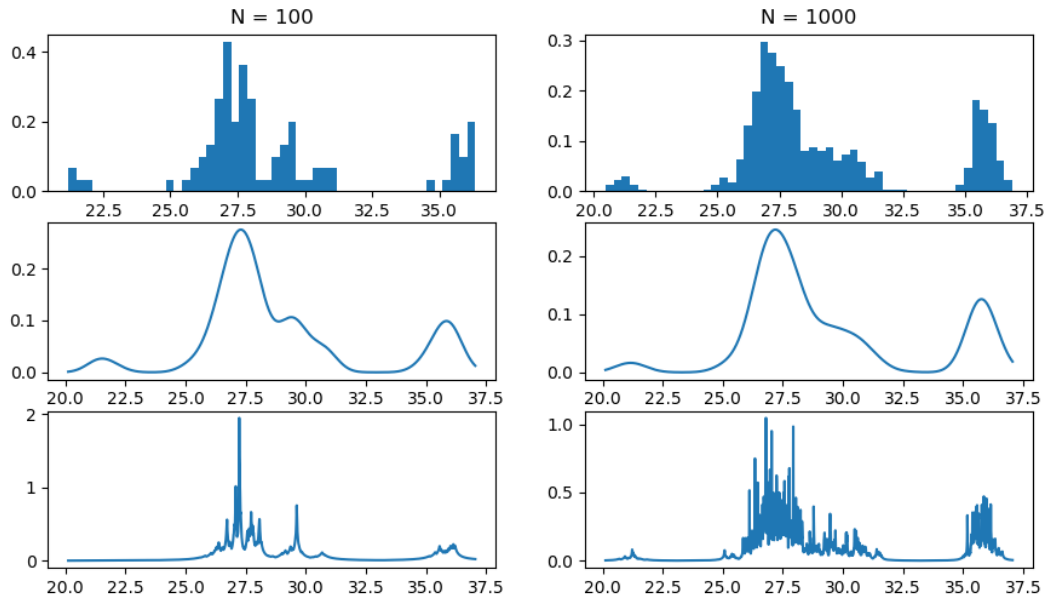
# PRML Assignment1 Report

## Task 1

- The figures of three algorithms at different values of N (num_data) are as follows:

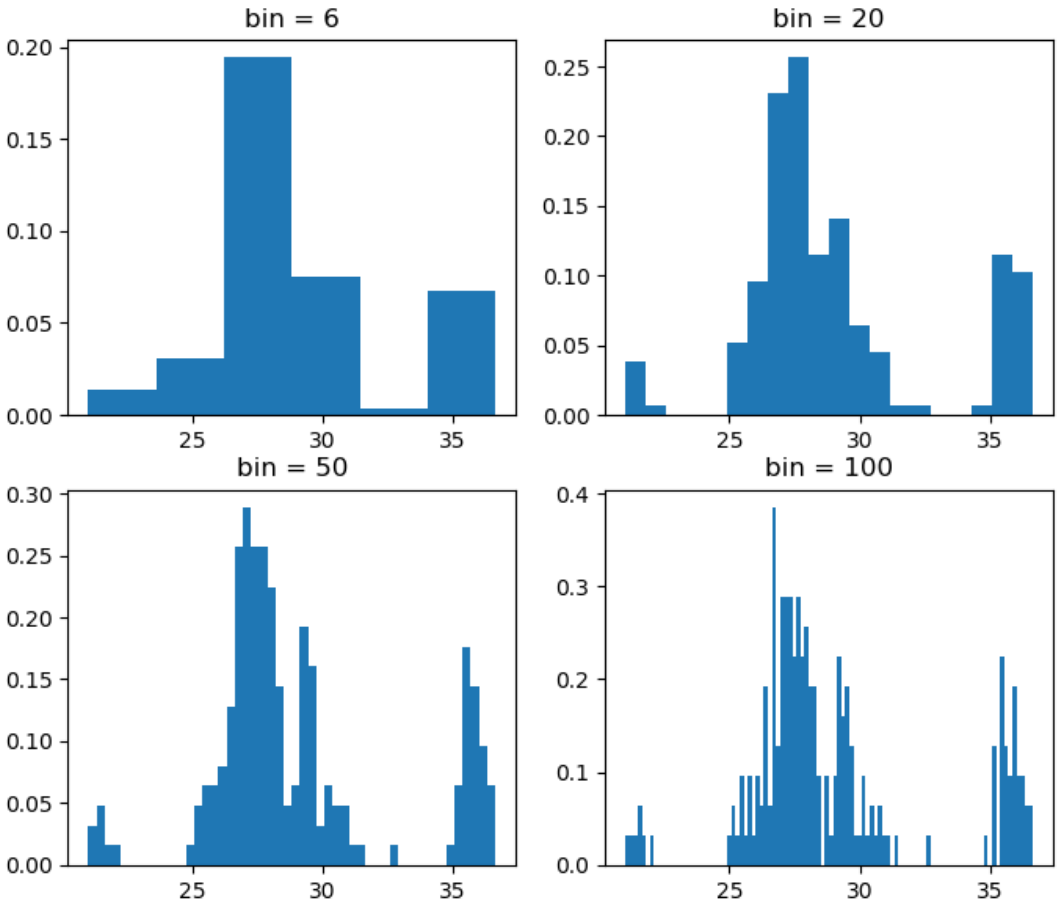



  The picture shows what happen for the estimation with N = 100 and 1000 (here I use bin = 50, h = 0.5 and k = 5).

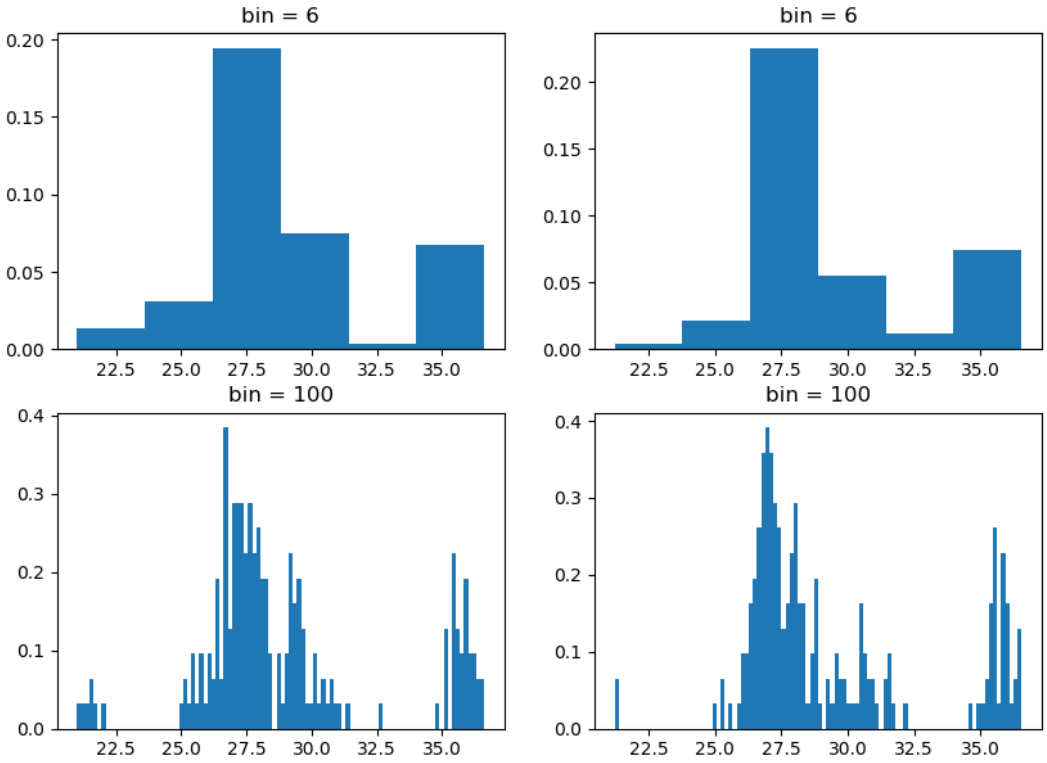
- We can clearly see that for histogram estimation, the figure will be smoother with a bigger N (it's more obvious when N = 10000).
- For kernel density estimation, it seems that there is no obvious difference, maybe the fitting precision will be different?
- For the nearest neighbor methods, with a bigger N, the figure become more 'rough', and there are more points with a much higher value.

## Task 2

- The figures of histogram estimation at different values of bin are as follows:




- It is just like that a higher value of bin give the figure higher resolution. I consider that when using a bigger 'bin', the figure can give more information, like show the distribution more clearly. But if the num_data isn't big enough, when bin's value is a big one, the figure will be more affected by the randomness of the data, namely its robustness is worse. (the following figures are with different 200 sample data, it's obvious that the two figures have more difference with bin = 200)




**how could you pick the best (or good) choice for this number of bins?**

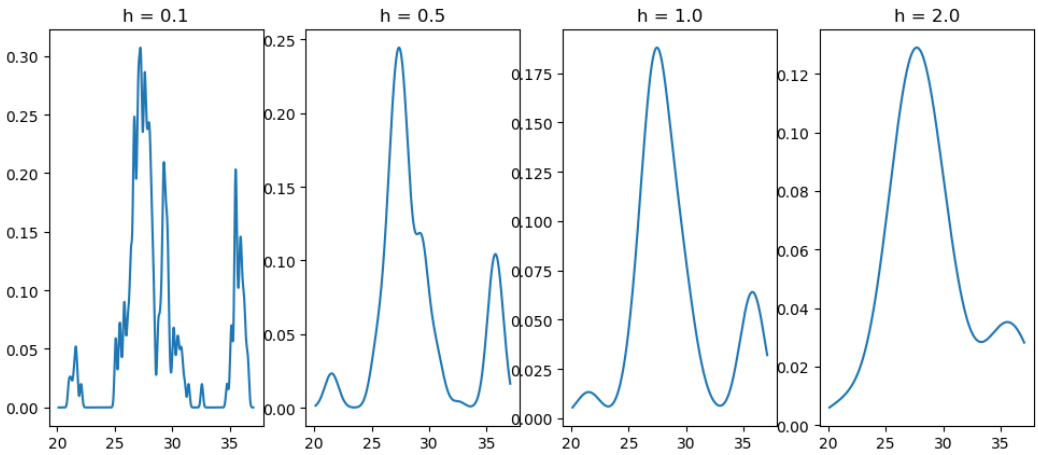
- For a certain num_data, a good estimation need to be robust and won't lose to much information(if bin is too small the data in biggish range will be the same, so some information of them will be lost). In my opinion, we can design an function to assess each estimation with different bin. For example, for each bin, we can try 5~10 sets of data, calculate variance of every bin between these sets, and sum them up, get the value S. Now we calculate S/bin, and the result can reflect the robustness of this estimation to a certain extent. Maybe then calculate result*f(bin)? (f(bin) is positively correlated with bin)

  At last we compare those results of the function with different bin, and we can get a best choice.

  *According to the related references, there have been some good methods to estimate bin, like **Sturges' formula**, **Rice Rule** and so on. None of these method can be called the best, so we can only get an approximate value of h by these method, which is about 10(I am confused with this answer) with the num_data = 200

# Task3

- The figures of kernel density estimation at different values of h are as follows:



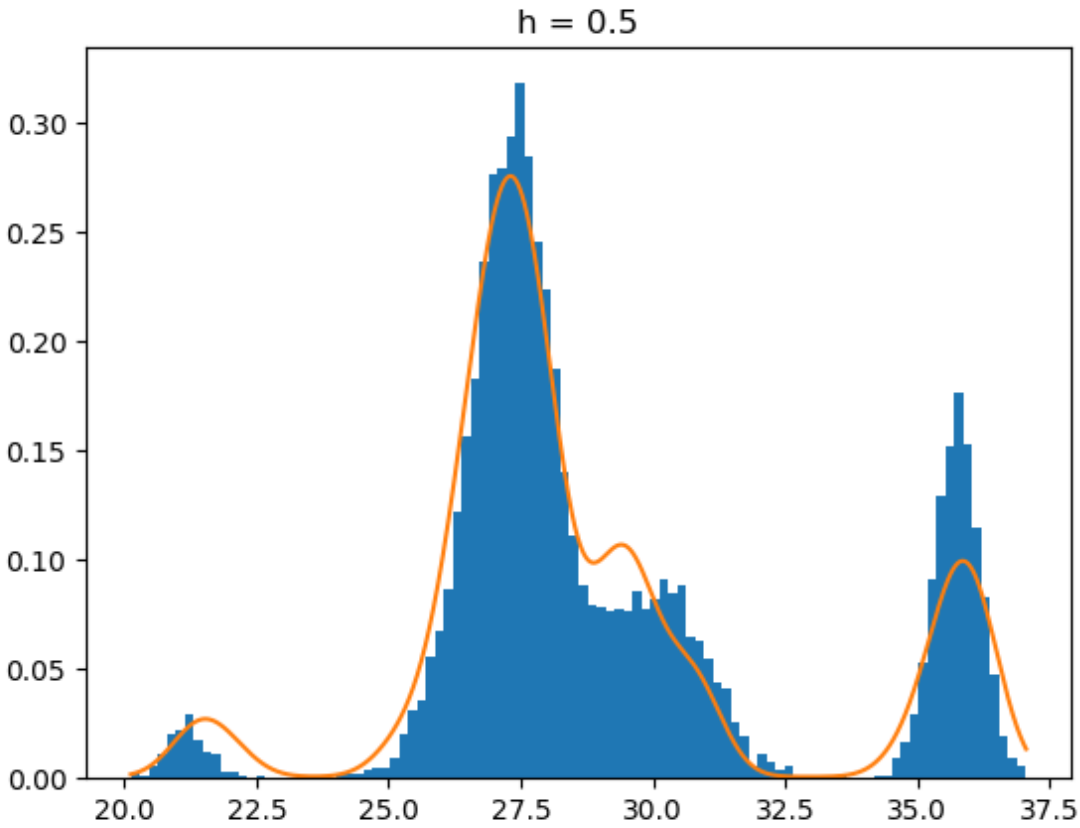

  we can see that the h is bigger, the curve is smoother. If the h is big enough, the curve will change to a smooth parabola.

- According to the common sense, we know that usually a smooth curve is more robust than a rough one. But through observation, we can find that when h is too big, the curve will become too much different. So we still need to find a tradeoff between big and too big. Here is my solution:

  **how to choose h?**

- By observed, I consider that if the curve of KDE is smooth enough, its robustness will also be good enough. Then, the key to evaluate the estimation should be its predictive accuracy. Here I use histogram estimation with bin = 100 and num_data = 10000 to help observe, Because

  such a big num_data guarantees its robustness and a relatively big bin make its details are also great. In this case, we can think of it as a standard curve and we can compare it with the KDE, like this:




The estimation shown in the figure above is a good one, In fact when we choose h at the range of (0.4,0.5), it all work out pretty well. (It's inevitably that there is some deviation for the num_data is just 100)

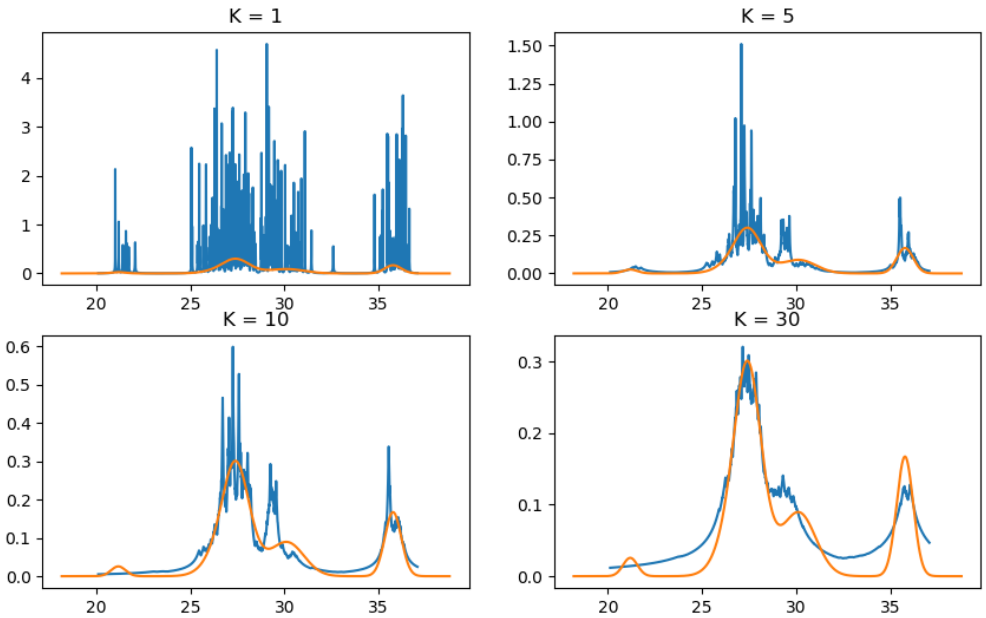
* I find a method of bandwidth estimation in Wikipedia called **rule-of-thumb**. The result is estimated by the following formula:

$$h = (\frac{4\sigma^5}{3n})^{\frac{1}{5}}$$

where σ is the standard deviation of the samples，and we can get a h that approximately has value of 1 (but I don't think this is better)

## Task4

- The figures of nearest neighbor methods at different values of K are as follows:




when we have a small k, the figure will be very uneven. Like the Figure with K = 1, in which there are lots of points have a extremely high value. With the increase of K, the number of "high points" gradually reduces and be more smooth. After K is greater than 10 (but not too big), the Figure look more and more like the true figures.

**please show that the nearest neighbor method does not always yield a valid distribution**

- For each of the x coordinates(x∈(-∞,+∞)), the y will be (K/N) * 1/V. The K/N will be a constant c, and V is linearly dependent on distance, so that we can assume that it equals to kx+b. So the functional integration will be :

$$\int_{-\infty}^{\infty} \frac{c}{kx+b}$$

and that won't converge to 1. Also I do a sum with the program. and when K = 1, the functional integration at x∈(min_range, max_range) is above 3, so the result is clear.