Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

equal-frequency partitioning

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

equal-frequency partitioning

Bin 1	5, 10, 11, 13
Bin 2	15, 35, 50, 55
Bin 3	72, 92, 204, 215

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

equal-frequency partitioning

Bin 1	5, 10, 11, 13
Bin 2	15, 35, 50, 55
Bin 3	72, 92, 204, 215

What is Smoothing by bin mean/median/boundary?

	B	C	D	E	F	G	H	I	J	K	L
	5	10	11	13	15	35	50	55	72	92	204	215

No of bins:	4
No of elements:	12
No of elements each bin has:	3

How do we define the first bin?
We need a bin that encloses 5, 10 and 11.
(4.5, 11.5]: This is also correct but let's look at Pandas.

What Pandas has created is:
(4.999, 12.5]: Range exlusive of 4.999 and starting from there. Also range inclusive of 12.5 and ending there.

Is it wrong? No.
Next bin:
(12.5, 42.5]: Is it wrapping the elements 13, 15 and 35?

Next bin would start at 42.5. Can we say this?

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

equal-frequency partitioning

Bin 1	5, 10, 11, 13
Bin 2	15, 35, 50, 55
Bin 3	72, 92, 204, 215

What is Smoothing by bin mean/median/boundary?
Replace each bin value is replaced by mean/median/nearest boundary

On smoothing by bin-boundary (bins follow equal-frequency partitioning):

Bin 1: 5, 13, 13, 13
As 5 is closer to boundary value ‘5’. And, 10, 11 are closer to boundary value ‘13’

Bin 2: 15, 15, 55, 55
Bin 3: 72, 72, 215, 215

Original:

Bin 1	5, 10, 11, 13
Bin 2	15, 35, 50, 55
Bin 3	72, 92, 204, 215

Smoothing by equal-frequency binning using the mean of each bin

1. creation of bins
In code: *pd.qcut()*

2. grouping the data according to bins
In code: *df.groupby()*

3. find the mean of each group
In code: *df.groupby().mean()*

4. create a map of bin labels and mean values
In code: it is essentially a dictionary that looks like this:

```
{
  '(4.999, 14.333]': 9.75,
  '(14.333, 60.667]': 38.75,
  '(60.667, 215.0]': 145.75
}
```

A dictionary is simply key-value pairs.

5. Populate a new column containing the mean of each bin for each data point.

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

equal-width partitioning

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

equal-width partitioning

The width of each interval is (215 – 5)/3 = 70.

Bin 1	5, 10, 11, 13, 15, 35, 50, 55, 72
Bin 2	92
Bin 3	204, 205

Perform Smoothing by bin mean/median/boundary.

Bins using equal width partitioning.
Elements: 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

The width of each interval is (215 – 5)/3 = 70.

Domain for bin-1: 5 up to, but not, 75 (= 5 + 70)
Domain for bin-2: 75 to 144
Domain for bin-3: 145 Onwards (inc. 215 from the input data set)

Bin 1	5, 10, 11, 13, 15, 35, 50, 55, 72
Bin 2	92
Bin 3	204, 205