**1. First of all I checked if null values is present in dataset and below is result**.

Total Percent

loss\_amt 0 0.0

region 0 0.0

smoker 0 0.0

children 0 0.0

bmi 0 0.0

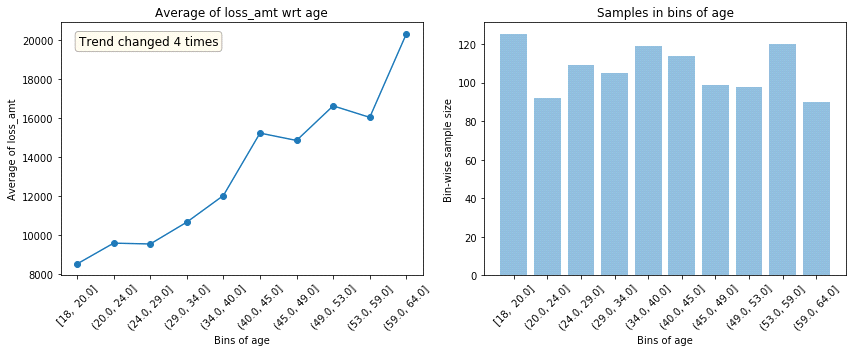
sex 0 0.0

age 0 0.0

No null values found in any features.

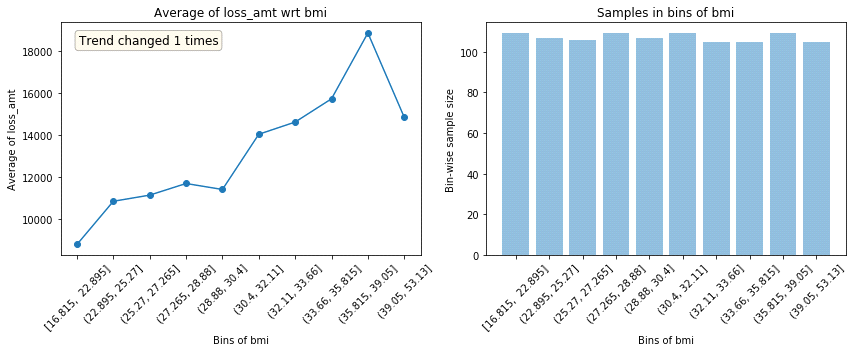
**2. Now divide each numeric feature in 10 bins. Now plot the average loss-amount of each bin against bin range.Below is what i got for each numeric feature.**

**a. Plot for age feature**



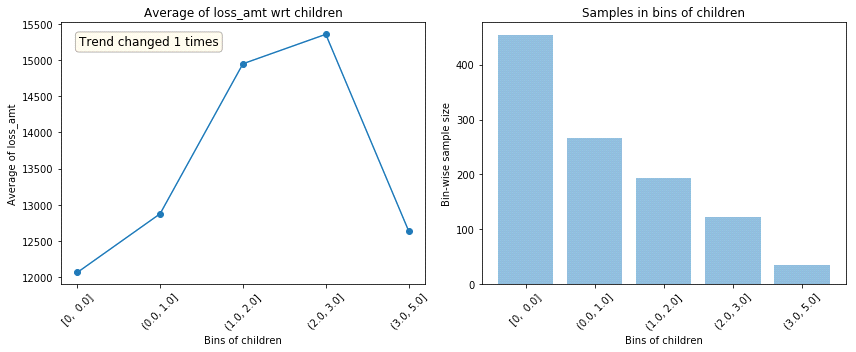
By analyzing the above plot we can say that loss amount increases with increase in age generally but this trend change 4 times in this dataset.Overall we can say that loss amount will be greater for older people.

**b. Plot for body mass index**



Here, loss amount is increasing with greater bmi.But after reaching 39,loss amount decreases with greater bmi.

**c. Plot for number of children**



It is clear from above plot that loss amount increases with number of children.But loss amount deceases when number of children is greater than 3.

Sudden and repeated changes in trend direction could imply noisiness. But, such trend change can also happen because that bin has a very different population in terms of other features and hence, its default rate can’t really be compared with other bins.

**2. Now its time to find corelation of features with local test data.Below is what I got.**

Feature ... Trend\_correlation

0 age ... 0.878175

1 bmi ... 0.666663

2 children ... 0.665936

If a feature doesn’t hold same trend w.r.t. target across train and evaluation sets, it can lead to overfitting. This happens because the model is learning something which is not applicable in test data. Trend correlation helps understand how similar train/test trends are and mean target values for bins in train & test are used to calculate it.Bmi and children feature seems noisy.

**3. There are some statistical insights for each numeric features.**

age bmi children loss\_amt

count 1338.000000 1338.000000 1338.000000 1338.000000

mean 39.207025 30.663397 1.094918 13270.422265

std 14.049960 6.098187 1.205493 12110.011237

min 18.000000 15.960000 0.000000 1121.873900

25% 27.000000 26.296250 0.000000 4740.287150

50% 39.000000 30.400000 1.000000 9382.033000

75% 51.000000 34.693750 2.000000 16639.912515

max 64.000000 53.130000 5.000000 63770.428010

**3. Finally I found learning curve for xgboost model and below is curve.**



Here both training error and cross validation error curve converges means our model is being trained in right direction.