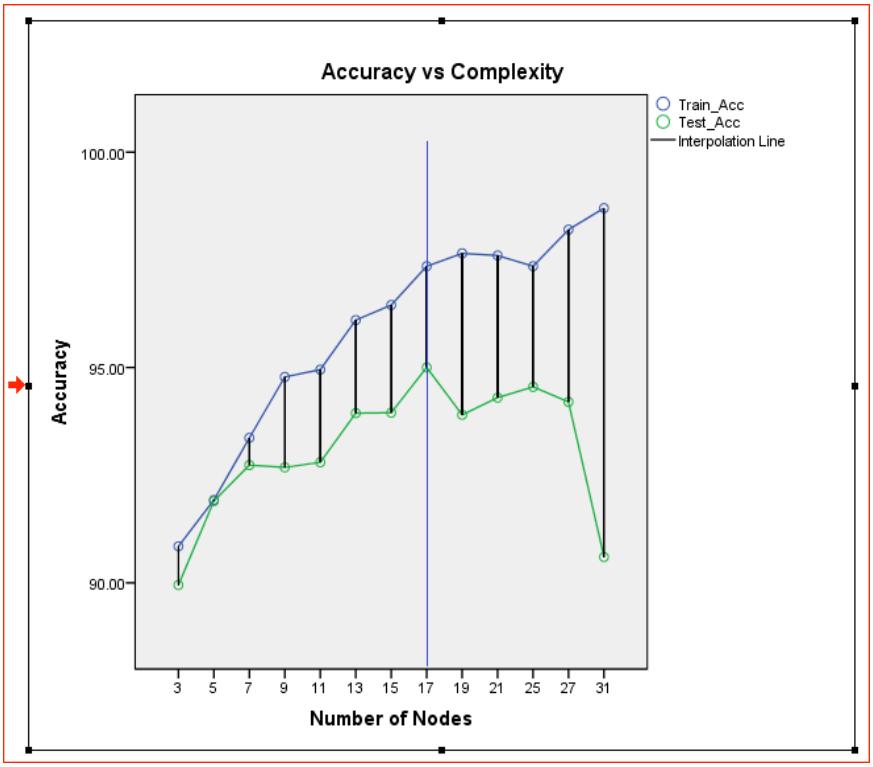
Problem 1
The following is the accuracy vs complexity plot for training and testing data.

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Class
	Independent Variables	V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11
	Validation	Split Sample
	Maximum Tree Depth	10
	Minimum Cases in Parent Node	2
	Minimum Cases in Child Node	2
Results	Independent Variables Included	V10, V1, V7, V6, V11, V9, V3, V2, V4, V8, V5
	Number of Nodes	17
	Number of Terminal Nodes	9
	Depth	6



By observing the above Accuracy plot, the best model I would propose for this particular dataset including the following parameters, minimum of cases for parents is 2 and for child is 2, the stopping condition is with depth 6. In this model, the number of nodes is 17 with 9 terminal nodes. The sample split validation setting is 65% Training and 35% Testing data. The above result implies another feature that SPSS algorithm do not push the tree to the maximum depth as preset. However, the accuracy is desirable in this case.

Classification

Sample	Observed	Predicted 1	Predicted 2	Percent Correct
Training	1	114	0	100.0%
	2	5	103	95.4%
	Overall Percentage	53.6%	46.4%	97.7%
Test	1	35	1	97.2%
	2	3	39	92.9%
	Overall Percentage	48.7%	51.3%	94.9%

Growing Method: CRT
Dependent Variable: Class

Risk

Method	Estimate	Std. Error
Resubstitution	.033	.010
Cross-Validation	.063	.014

Growing Method: CRT
Dependent Variable: Class

The reason I have picked this model because the accuracy of the Testing data (Green Line) is 94.9% reaches the highest point of the curve with the least complexity (=17nodes). Also, the test set accuracy on the misclassication matrix are also acceptable range , the total cases of both predicted values (1 & 2) are very balanced and the max percent accuracy of predicted value 1 reaches 97.2%. In addition, the standard deviation the 10-fold standard deviation is ONLY 0.014 with Estimate value 0.063 which is small enough to conclude that the model will not be overfitting. Also, this could imply that the Training and Testing data is splited into an appropriate portion for the classification model.

The feature of the data includes 11 Nominal values attribute and 1 Class label. The dimension of the entire dataset includes 300 records. The dataset was splited into 75% of training data and 25% Testing data. The names of the variables are not provided.

During the process of building the tree, I could observe that increasing the number of cases allowed in parent and child nodes is decreasing the complexity (number of nodes) of the tree. The mechanism of this effect because increasing the number of cases of nodes would let the node NOT splitting until it reaches the minimum of cases we set. So that the higher values we set on the parent and child nodes, the less complexity of tree we are supposed to get. Also, we could observe the above plot, the least of the complexity we have (e.g. Number of nodes =3), the less accuracy we could get.

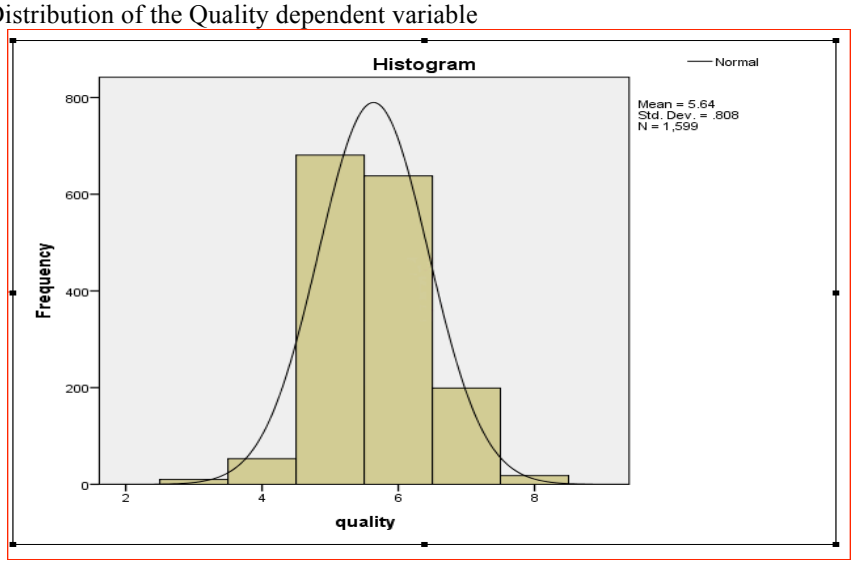
Problem2-1:
1.)Consider each quality level of wine to be a different class. Report how many classes there are and what is the distribution of these classes for the red wine data (how many cases of each class are there).
Ans:
There is 0-10 quality classes per data description. However, we could observe the following distribution and ONLY 3-8 Quality scores are picked by people. The distribution of the scores is pretty much close Normalized. The Frequency of quality rating is shown as below:

- 10 cases for score 3
- 53 cases for score 4
- 681 cases for score 5
- 638 cases for score 6
- 119 cases for score 7
- 18 cases for score 8
- 0 cases for score 0,1,2,9,10
- Total cases : 1599

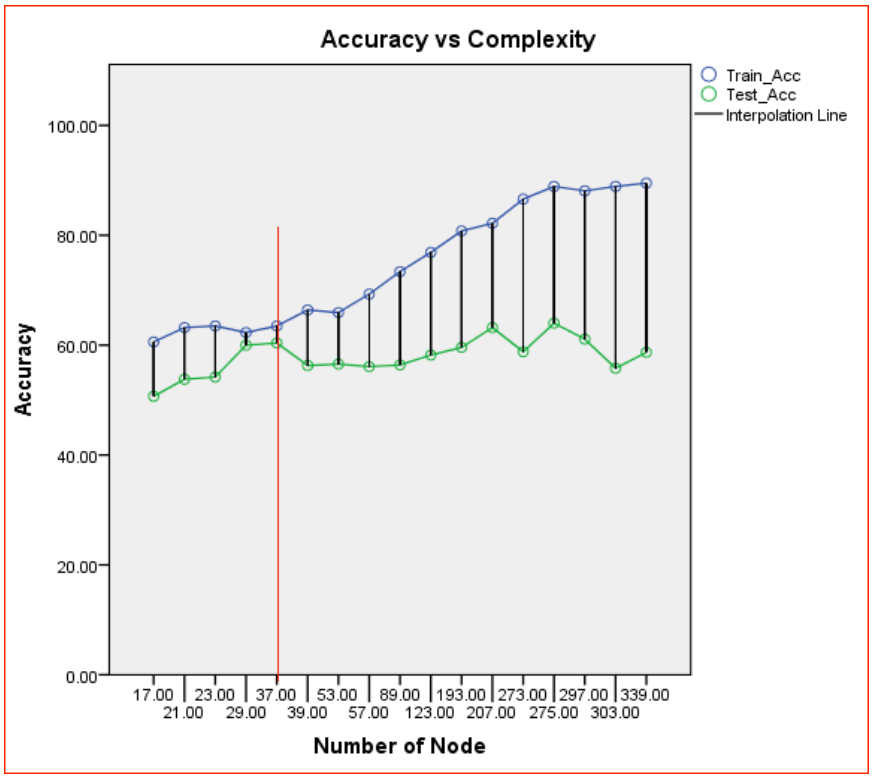
quality

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	10	.6	.6	.6
	4	53	3.3	3.3	3.9
	5	681	42.6	42.6	46.5
	6	638	39.9	39.9	86.4
	7	199	12.4	12.4	98.9
	8	18	1.1	1.1	100.0
	Total	1599	100.0	100.0	

The following is the Distribution of the Quality dependent variable



Problem 2-2:



By observing the above Accuracy plot, the best model I would propose for this particular dataset including the following parameters, minimum of cases for parents is 50 and for child is 10, the stopping condition is with depth 5. In this model, the number of nodes is 37 with 19 terminal nodes. The Sample split parameters was set 70% of Training and 30% of Testing Data

Classification

Sample	Observed	Predicted 3	4	5	6	7	8	Percent Correct
Training	3	0	0	5	2	0	0	0.0%
	4	0	0	27	10	0	0	0.0%
	5	0	0	381	95	1	0	79.9%
	6	0	0	156	292	13	0	63.3%
	7	0	0	16	81	49	0	33.6%
	8	0	0	0	5	4	0	0.0%
	Overall Percentage	0.0%	0.0%	51.5%	42.7%	5.9%	0.0%	63.5%
Test	3	0	0	2	1	0	0	0.0%
	4	0	0	11	5	0	0	0.0%
	5	0	0	168	36	0	0	82.4%
	6	0	0	66	102	9	0	57.6%
	7	0	0	4	40	9	0	17.0%
	8	0	0	1	5	3	0	0.0%
	Overall Percentage	0.0%	0.0%	54.5%	40.9%	4.5%	0.0%	60.4%

Growing Method: CRT
Dependent Variable: quality

The reason I have picked this model because the accuracy of the Training data (Blue Line) is 63.5% Testing data (Green Line) is 60.4% reaches the highest point of the curve with the least complexity. Also, the test set accuracy on the misclassication matrix are also acceptable range , the total cases of both predicted values (5, 6,&7) in Training Set and Testing set are very similar which aligns with the Accuracy vs complexity plot.

Risk

Method	Estimate	Std. Error
Resubstitution	.350	.012
Cross-Validation	.432	.012

Growing Method: CRT
Dependent Variable: quality

Classification

Observed	Predicted 3	4	5	6	7	8	Percent Correct
3	0	0	8	2	0	0	0.0%
4	0	0	31	21	1	0	0.0%
5	0	0	494	177	10	0	72.5%
6	0	0	150	440	48	0	69.0%
7	0	0	22	71	106	0	53.3%
8	0	0	0	6	12	0	0.0%
Overall Percentage	0.0%	0.0%	44.1%	44.8%	11.1%	0.0%	65.0%

Growing Method: CRT
Dependent Variable: quality

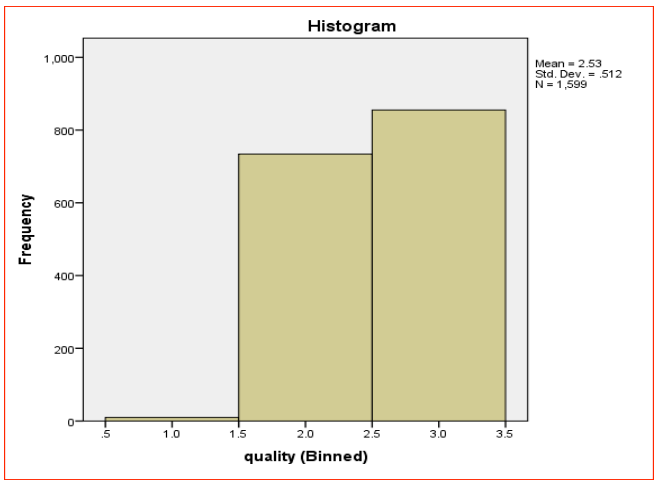
In addition, the 10-fold classification is processed with the parameters as above including Depth, parent nodes and child nodes setting. The standard deviation the 10-fold standard deviation is ONLY 0.012 with Estimate value 0.432 which is small enough to conclude that the model will not be overfitting.

The feature of the data includes 11 Nominal values attribute and 1 Class label (Quality=1-10). The dimension of the entire dataset includes 1599 records. The dataset was splited into 70% of training data and 30% Testing data. The names of the independent variables are provided as the following:

1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol
12- Quality (1-10 levels)

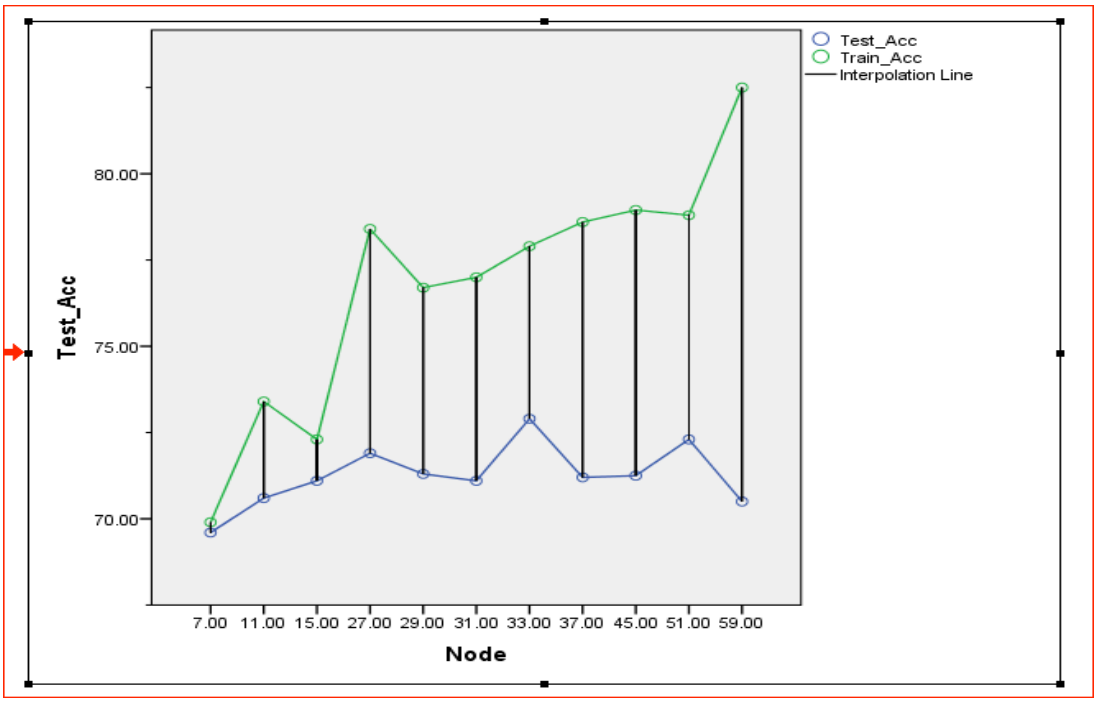
During the process of building the tree, I could observe that increasing the number of cases allowed in parent and child nodes is decreasing the complexity (number of nodes) of the tree. The mechanism of this effect because increasing the number of cases of nodes would let the node NOT splitting until it reaches the minimum of cases we set. So that the higher values we set on the parent and child nodes, the less complexity of tree we are supposed to get. Also, we could observe the above plot, the least of the complexity we have (e.g. Number of nodes =17), the less accuracy we could get.

Problem 2 -3): Binning into 3 categories (<=3, 4-6, >7+)



Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	quality (Binned)
	Independent Variables	fixedacidity, volatileacidity, citricacid, residualsugar, chlorides, freesulfurdioxide, totalsulfurdioxide, density, pH, sulphates, alcohol
	Validation	Split Sample
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	10
	Minimum Cases in Child Node	2
Results	Independent Variables Included	alcohol, density, chlorides, sulphates, volatileacidity, citricacid, totalsulfurdioxide, fixedacidity, pH, residualsugar, freesulfurdioxide
	Number of Nodes	15
	Number of Terminal Nodes	8
	Depth	3



Based on the above, the Quality dependent variable was binned to 3 Categories and then performing the Tree Classification based on various parameters entries. The final model I picked is shown on the above with the following parameters: Parents case = 10, Child case=2, Terminal Node=8, Node =15, Depth=3. The reason for picking this model based the Training (72.3%) and Testing (71.1%) Accuracy Result which shows the accuracy of the testing set data has the best Accuracy with the least distance from the training set of data. So that, this probably would avoid over-fitting situation. Also, we could see that the Accuracy of the testing result doesn't improve a lot even the complexity (Number of nodes) increases.

Risk

Method	Estimate	Std. Error
Resubstitution	.266	.011
Cross-Validation	.316	.012

Growing Method: CRT
Dependent Variable: quality (Binned)

Classification

Observed	Predicted <= 3	Predicted 4 - 6	Predicted 7+	Percent Correct
<= 3	0	7	3	0.0%
4 - 6	0	501	233	68.3%
7+	0	183	672	78.6%
Overall Percentage	0.0%	43.2%	56.8%	73.4%

Growing Method: CRT
Dependent Variable: quality (Binned)

In addition to the above, 10-fold classification with same parameters are also carried out. This result actually quite aligned with the previous result. The standard deviation of cross-validation is 0.012 with Estimate 0.316 which means the small enough to conclude that this model would not be over-fitting.

Classification

Sample	Observed	Predicted <= 3	Predicted 4 - 6	Predicted 7+	Percent Correct
Training	<= 3	0	2	3	0.0%
	4 - 6	0	389	130	75.0%
	7+	0	176	421	70.5%
	Overall Percentage	0.0%	50.6%	49.4%	72.3%
Test	<= 3	0	5	0	0.0%
	4 - 6	0	154	61	71.6%
	7+	0	72	186	72.1%
	Overall Percentage	0.0%	48.3%	51.7%	71.1%

Growing Method: CRT
Dependent Variable: quality (Binned)

By observing the above, the <=3 case are taking very small portion on the classification which is true because the <=3 quality bin takes up very small portion of entire dataset. The overall percentage of the 4-6 and 7+ categories between the Training and Testing set are very similar although their Correct percent is not very high, but this model is the best among all other models because the training and testing dataset result does not differ too much which avoids the over-fitting problem. Also this actually could conclude that Training and Testing samples are split in appropriate portion.

Problem 2- 4: Comparing performance between model with original class data and binned class data.
Ans:
The Binned class model apparently has a higher Accuracy result than the Original Class data Model. This result might due to less class target after smoothing for the classification process. And the binning class method is actually make the result is more presentable for the reader or business user. With the higher accuracy model, the user could plug in a unknown data to the model in the future to get a dependable result which is the reason that we run this classification process.

Problem 2- 5: Other improvement on the above model:
suggestions:

- Bin the Class labels into 2 Categories instead of 3 shown above
- Adjust the the Training and Testing dataset portion, 75% of training and 25% of testing in this case

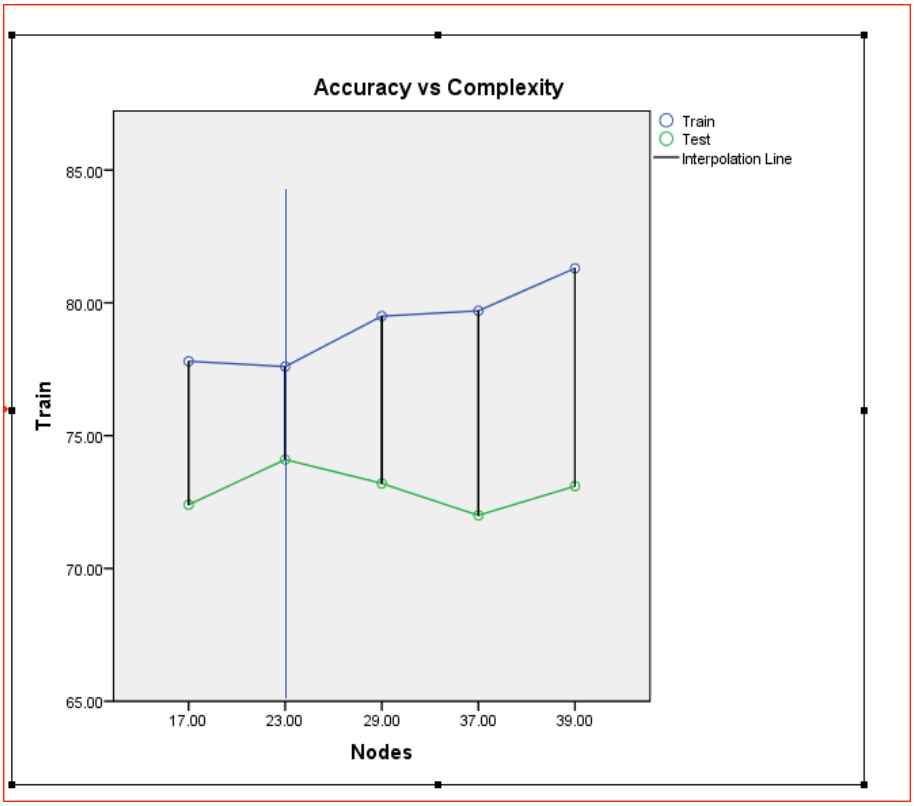
Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	quality (Binned)
	Independent Variables	volatileacidity, density, sulphates, alcohol, fixedacidity, citricacid, residualsugar, chlorides, freesulfurdioxide, totalsulfurdioxide, pH
	Validation	Split Sample
	Maximum Tree Depth	5
	Minimum Cases in Parent Node	2
	Minimum Cases in Child Node	2
Results	Independent Variables Included	alcohol, density, chlorides, sulphates, volatileacidity, citricacid, totalsulfurdioxide, fixedacidity, residualsugar, pH, freesulfurdioxide
	Number of Nodes	23
	Number of Terminal Nodes	12
	Depth	5

Classification

Sample	Observed	Predicted <= 5	Predicted 6+	Percent Correct
Training	<= 5	445	130	77.4%
	6+	150	526	77.8%
	Overall Percentage	47.6%	52.4%	77.6%
Test	<= 5	125	44	74.0%
	6+	46	133	74.3%
	Overall Percentage	49.1%	50.9%	74.1%

Growing Method: CRT
Dependent Variable: quality (Binned)



Problem 3:

a.) Feature Selection and Feature extraction

- Feature Selection - Remove the irrelevant or redundant attributes (Selecting the most relevant attributes)
- Feature Extraction – Transform/Combine the original data feature to a new set of data feature in order to reduce the dimension of the original dataset

b.) Training and Testing data

- Training data – is used as reference to build the algorithm model
- Testing data - is used to validate the model and check how good the model is built to predict the dependent variables (Class label)

c.) Parametric reduction techniques and non-parametric reduction techniques

- Parametric reduction techniques is used a model to perform data reduction. This method depends on the model and the set parameters to fit data in the model. So that this method (e.g. linear regression) heavily depends on the model and parameters , so that users need to make sure that the model is chosen correctly for the dataset being analyzed because the result might reduce much more data than it is actually necessary to.
- Non-parametric reduction techniques is the opposite to the Parametric reduction techniques. This method directly apply the reduction techniques to the data without using models.

d.) Uniform binning and non-uniform binning

- Uniform binning - which is named as Equal-width partitioning. This method is basically divide the range into N intervals of equal size. The outliers might dominate the presentation and also this method does not handle the skewed data well enough.
- Non-uniform binning – which is named as Equal-depth partitioning. This method divides the range into N intervals, each containing approximately same number of samples.

e.) convariance matrix and correlation matrix.

- Convariance matrix and correlation matrix are actually very similar, which is positively correlated when the result turns out as positive value, and negatively correlated with negative value. Zero means they data are not correlated at all.
- Correlation matrix is a scaled version of Covariance Matrix. The covariance matrix is used when the data are in the same or similar scale or units. If not, then use correlation matrix instead.