**Final Project - ST635 Fall 2017**  Qian Yu

**Executive Summary:**

|  |  |
| --- | --- |
| The final model I chosen is logistic regression with glm() function on the right side screenshot.  The reason I chosen it are two:  Firstly, this model includes all the variables that is significant to factor “chd”( coronary heart disease) :   * Variables “famhist” and “age” are significant in 0.001 level, and variables “Tobacco”, “ldl” , and “typea” are significant in 0.01 level. * It has the best performance (error rate for both training data and testing data) than the individual variable model I tried, and the original model with all the variables that this dataset has. * Secondly, this model is better than the one other method – decision tree   Variable “famhist” is the most important in determining the prediction. By 1 unit “famhist” increase, the chance to get “chd” raise by 0.91. (I changed the data type from factor to numeric. So 1 is absent and 2 is present, the different is 1.)  The second important variable is “ldl” By 1 unit “famhist” increase, the chance to get “chd” raise by 0.16.  The percentage of correct predictions on the training data is (256+87)/462 which is equal to 74.2424%. In other words 25.7576% is the training error rate.  To test the logistic regression model, I used a training period of current alcohol consumption greater than 20 g per day. 1  The error rate of this testing period is only 0.245%  It could also say that when the status of “chd” is present, the model is right only 43.40% of the time.  (46/ (60+46))  But for weeks when the status of “chd” is absent, the model is right as high as 90.9502% of the time.  (201/ (201+20)) |  |

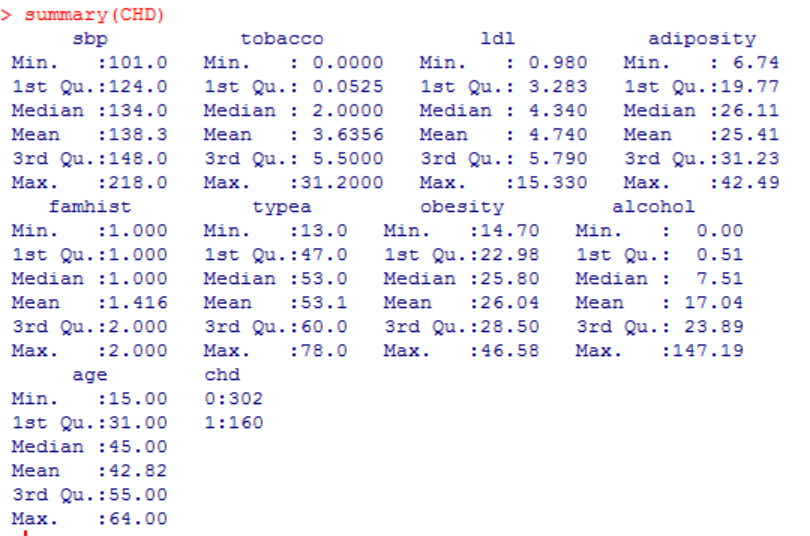
1. There is no unit in data set, I checked the average of current alcohol consumption of this data, and it is around 17. Also, most of the recommendations for alcohol consumption issued by government bodies in various countries are 20. <http://www.iard.org/policy-tables/drinking-guidelines-general-population/>

**Outline the analysis:**

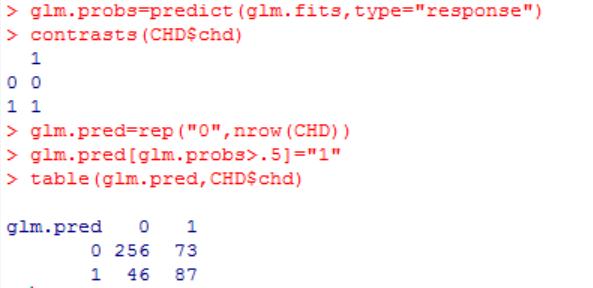
**Logistic Regression:**

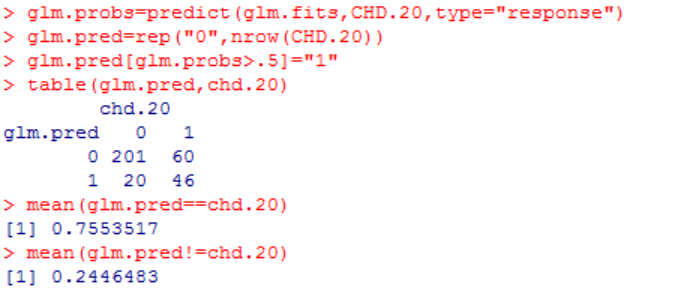
1. When I tried to run the logistic regression, I realize variable “famhist” is factor; “chd” is numeric. So I use function to switch their data type to make the analysis more accurate by CHD$famhist=as.numeric(CHD$famhist) ; CHD$chd=as.factor(CHD$chd). Then I run a summary to check if any data set up correctly by Summary(CHD).

(1 is absent and 2 is present, the different is 1.)



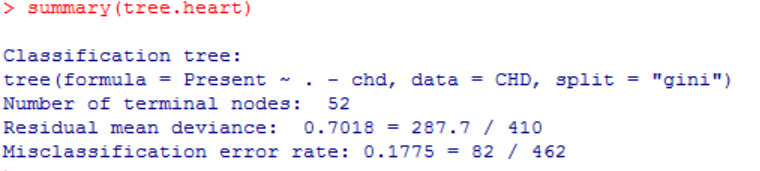
1. I use the full data set to perform a logistic regression with “chd” as the response and other nine variables as predictors by fit.glm <- glm(chd~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity + alcohol + age, data = CHD, family = binomial).
2. I used summary function again to find if any predictors appear to be statistically significant by summary(fit.glm). I found out that variables “famhist” and “age” are significant as their p-value is less than 0.001 ; and variables “Tobacco”, “ldl”, and “typea” are significant , as well, as their p-value is less than 0.01. Other four variables are not significant.
3. The predict () function can be used to predict the probability that coronary heart disease (chd) is present or absent by given values of all the predictors glm.probs=predict(glm.fits,type="response"). And the values correspond to the probability of the chd will be present, since it is dummy variable with a 1 for present contrasts(CHD$chd).
4. For creating a vector, I used glm.pred=rep("0",nrow(CHD)) and glm.pred[glm.probs>.5]="1".
5. I calculated the error rate on the training data by table(glm.pred,CHD$chd) and mean(glm.pred== CHD$chd)
6. Then I choose to used only the significant variables as a new model , glm.fits=glm(chd~ tobacco + ldl + famhist + typea + age, data = CHD, family=binomial) to run the step 4, 5, and 6 again, and looking for a better performance.



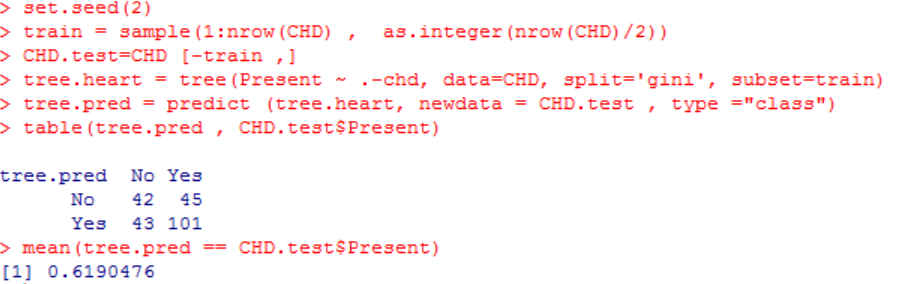
1. To test the logistic regression model, I used a training period of current alcohol consumption greater than 20 g per day, I used train=(CHD$alcohol>20), and CHD.20=CHD[!train,]
2. Then compute the error rate of the new model for the particular data period. 

**Classification Trees:**

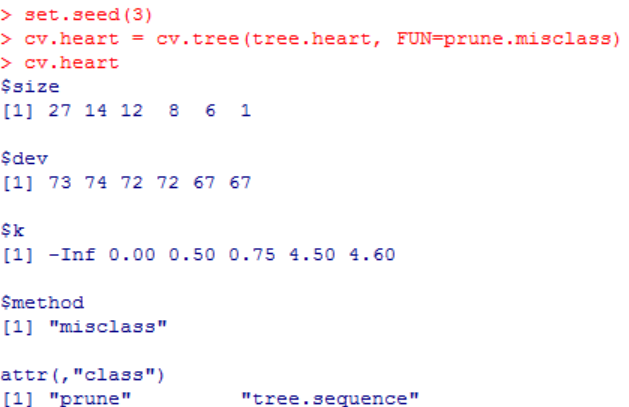
1. I first used classification trees to analyze this data set. CHD = read.table("CHD\_data.csv", sep=',' , header=TRUE) and library (tree)
2. I created a binary variable, called Present, which takes on a value of 1 as Present and 0 as absent. CHD$Present=as.factor(ifelse(CHD$ chd >=1, "No", "Yes")) and tree.heart = tree(Present∼.-chd, split='gini', data=CHD), then use summary() function to check the result.



1. Next step is estimating its test error, and create the training and testing by divided the data set as two parts. Then check the result.



1. To consider whether pruning the tree might lead to improved results, I used function cv.tree() performs cross-validation in order to determine the optimal level of tree complexity. And I found out that the tree with 1 or 6 terminal nodes results in the lowest cross-validation error rate, with 67 cross-validation errors.



1. Then I applied the predict() function with the test data, and the result improve from 0.619 to 0.641.

