Decision Trees

Dean D'souza

2016-08-09

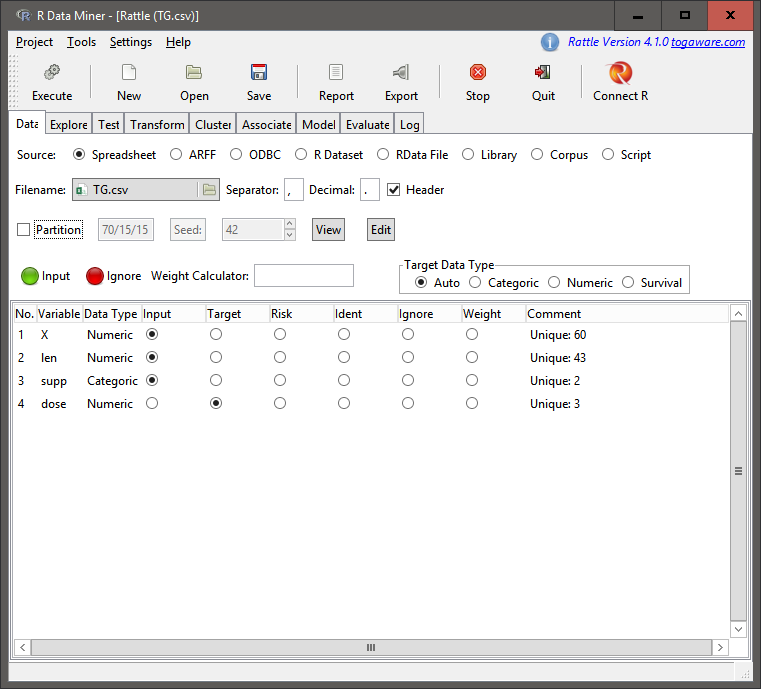
## Soltuions:

For the purpose of this assignment I chose the ToothGrowth dataset from base R. The description of the dataset is as follows:

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Description |
| len | numeric | Tooth Length |
| supp | factor | Supplement type (VC or OJ) |
| dose | numeric | Dose in milligrams per day |

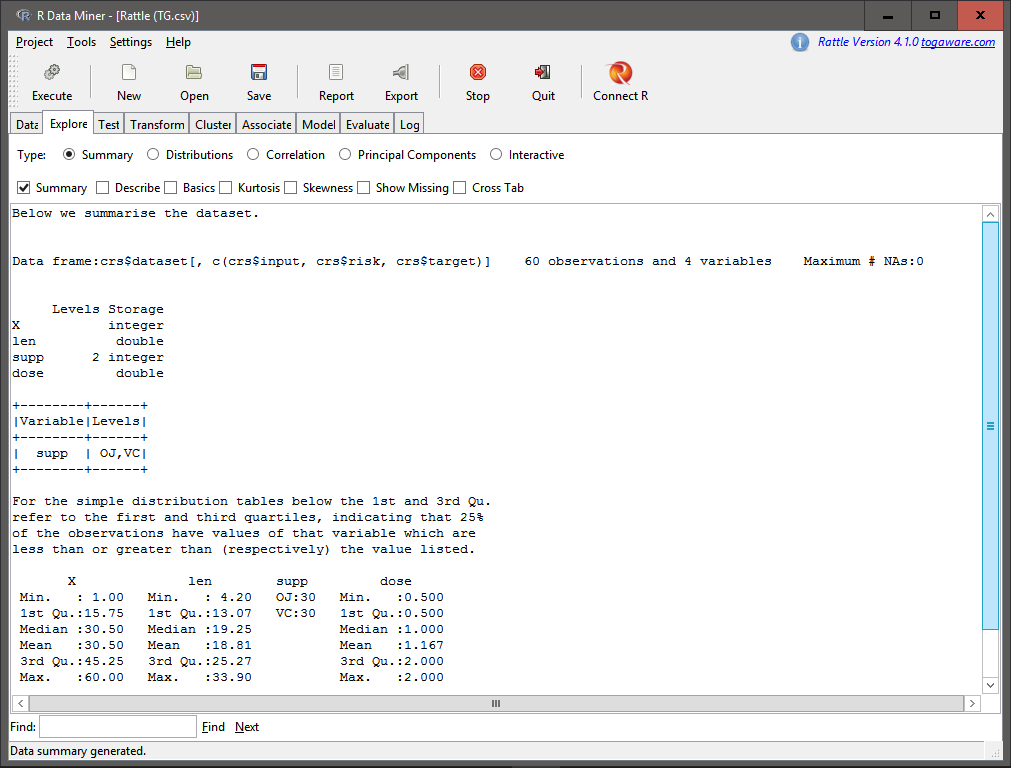
* **First, build a decision tree using Rattle just as we did here on a data set of your choosing.**
* Take several screen shots of your work and paste into a MS Word, LibreOffice, of pdf file, explain each step
* Tell me what you learned from the decision tree

We build the decision tree using rattle as follows:



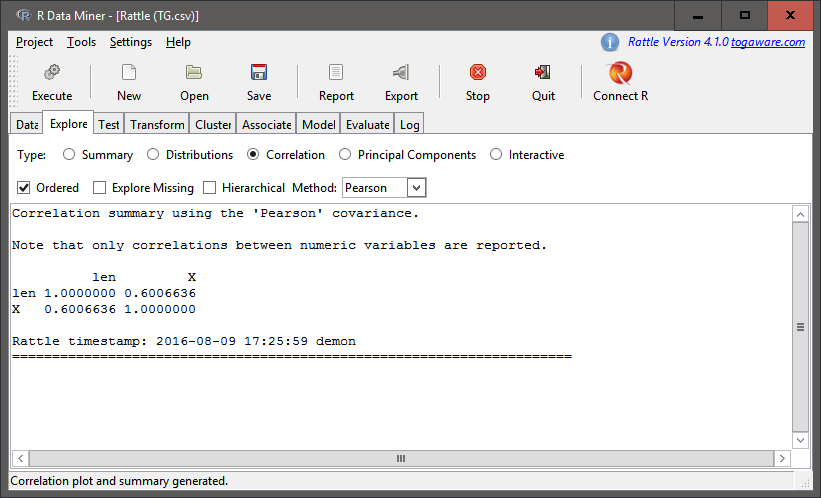
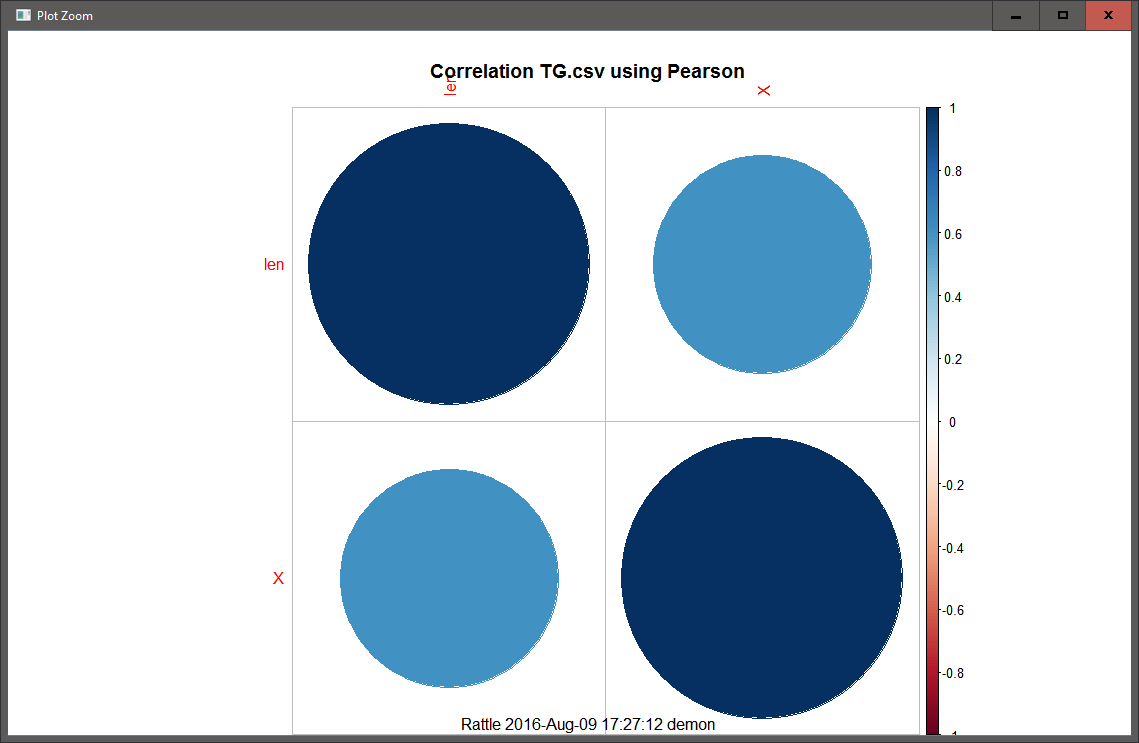
Setup

The above screen shot shows the initial screen after laoding the dataset and then selecting the response variable as dose.

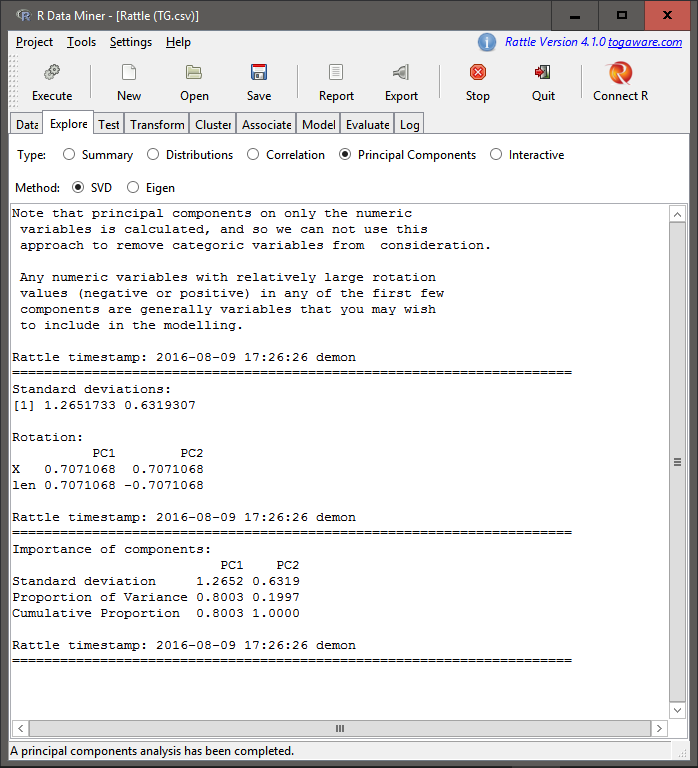
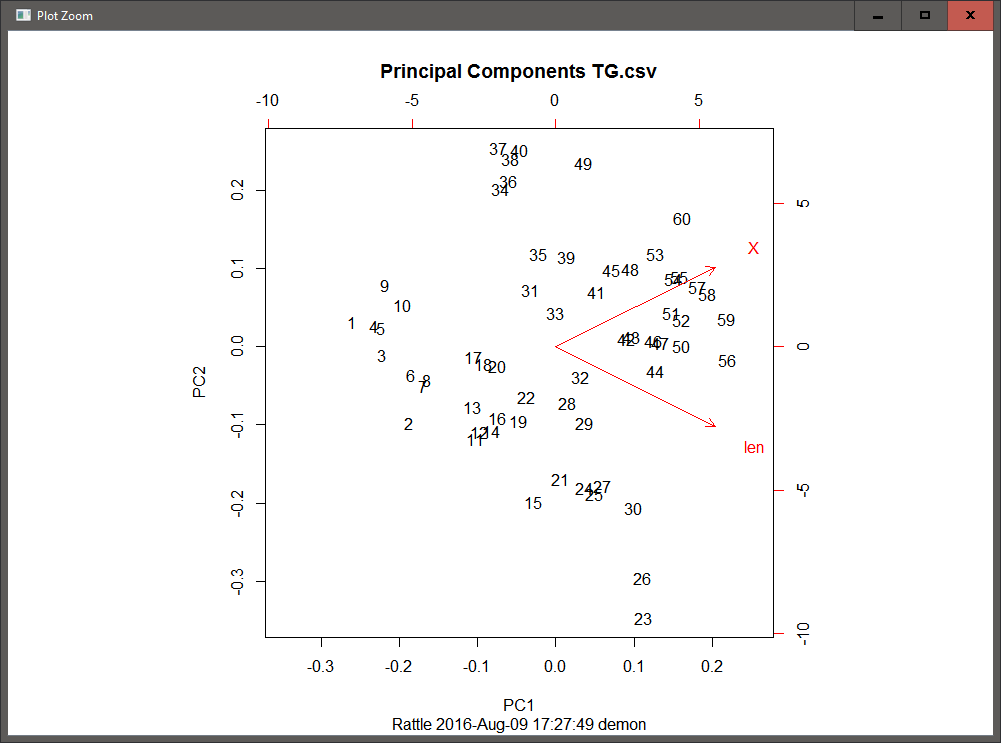


Summary

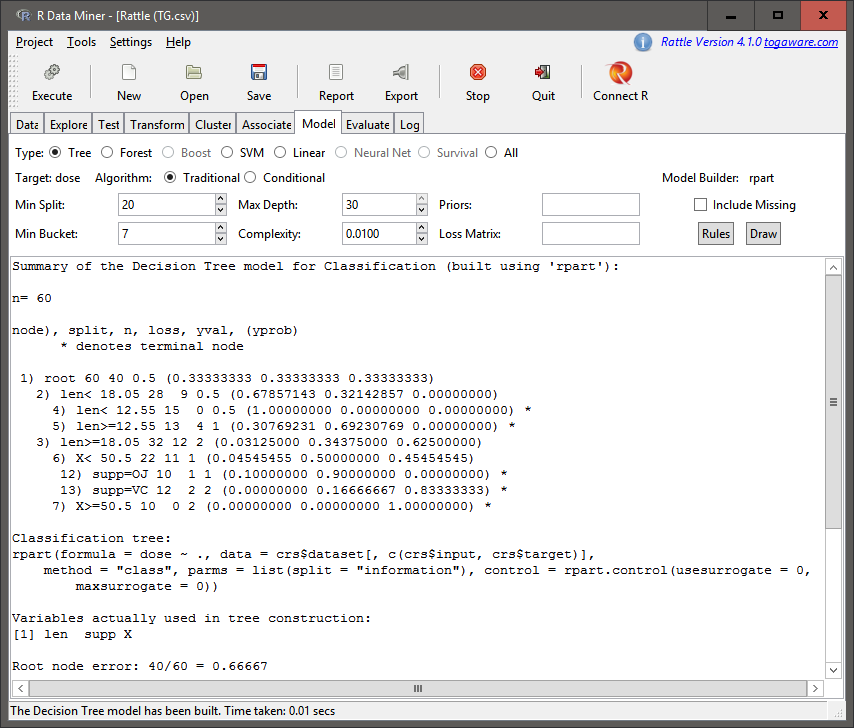
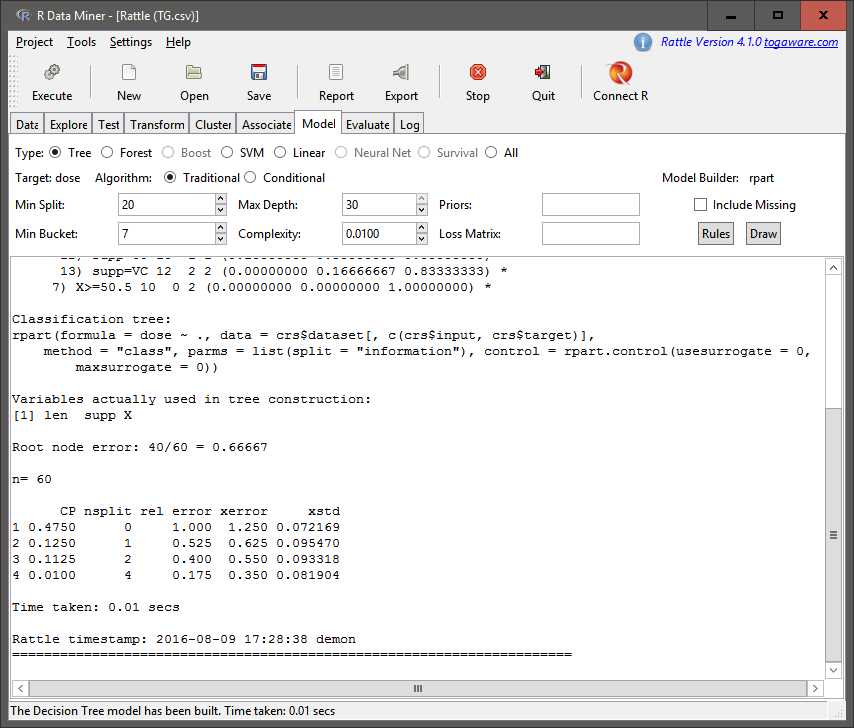
In order to get a better idea about the data we perform a summary on the data as shown above.  
 We then observe the correlation between varaibles as follows:

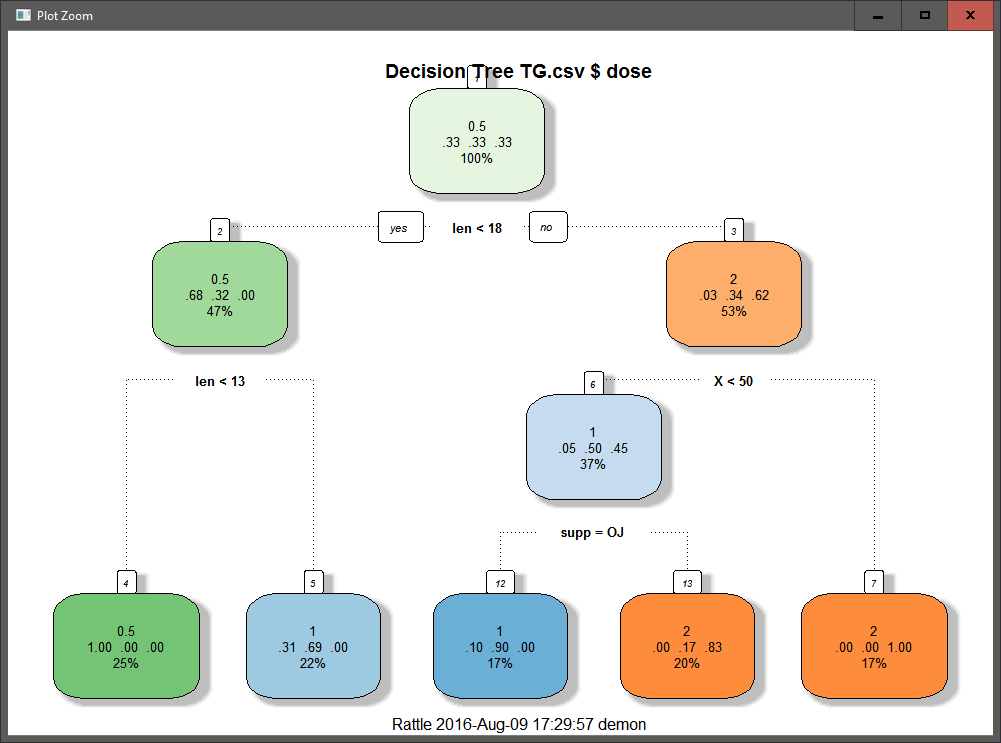
The above correlation matrix helps us get a better idea about how the variables are correlated. The correlation plot which comes out in the main RStudio plots tab gives us a visual perspective.  
 While not necessary to be done in rattle, an additional step which I took was to perform principal component analysis on the data (while exploring the functions of rattle) as follows:

The analysis and the plot helps us see that the len covariate can explain around 70% of the data.  
 We now begin to construct the decision trees through the model tab of rattle as follows(without setting the Loss Matrix):

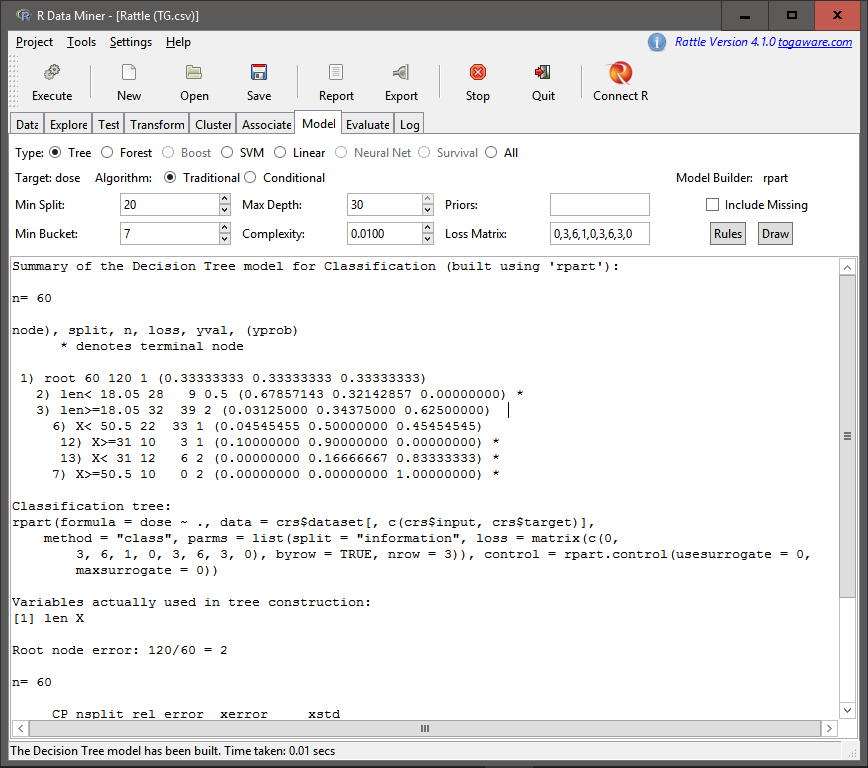
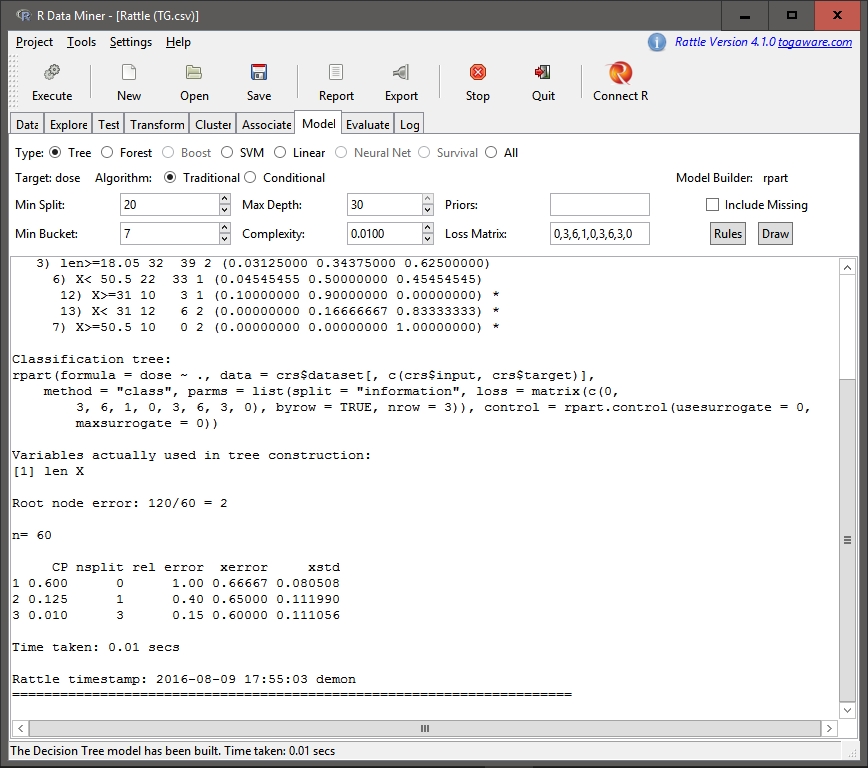
 

As we can see from the above output, at node 1, there are 60 rows of data with 40 classified wrongly and the decision at this point is 0.5. At node 2 we further split the data based on len being less than 18.05, such that this node has 28 rows with 9 classified wrongly. Further nodes of 4 and 5 split based on the condition of len being less than or greater than(or equal to) 12.55 such that in node 4 none are classified wrongly while at node 5 4 are classified wrongly with a dose of 1.  
 At node 3 (when len is greater than or equal to 18.05) we see that there are 32 rows such that 12 are wrongly classified as 2. the next split takes place over the intercept (represented by X, which is the intercept on the x axis) being less than or greater than (or equal to) 50.5 such that at node 6 there are 22 rows with 11 wrongly classified as 1 and at node 7 there are 10 rows out which all are classfied as 2. Further splitting takes place after node 6 to give node 12 with supp=OJ with 10 rows such that 1 is wrongly classified as 1 and node 13 with supp=VC with 12 rows such that 2 are wrongly classified as 2.  
 We can further draw the tree diagram using the 'draw' button to obtain it under the plot tab in RStudio as follows:

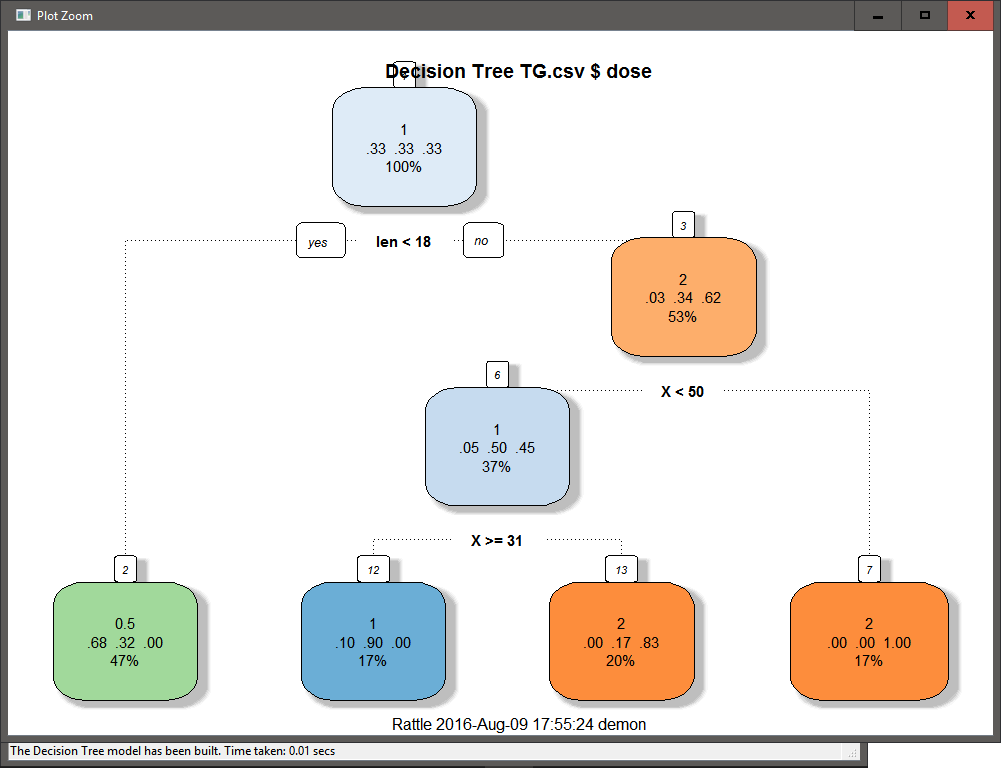


Decision Tree 1

We now rebuild the decision tree with the Loss Matrix as follows:

Here we see that the only difference with the previous decision tree is that the nodes 12 and 13 are now split on the X intercept of 31, as well the distribution of the rows and the observations wrongly classified. We draw the decision tree as follows:



Decision Tree 1

However, the above decision tree (in my opinion) is not as ggood as the previous one due to the amount of observations wrongly classified, especially in the case of the nodes 12 and 13. However, it does try to take into account the effects fo wrongly classifying higher doses of medicine.

* **Build another model of your choice using Rattle on the same data set**
* You can run Association Analysis, PCA, Regression, .whatever
* Tell me what you learned from this analysis and from the decision tree combined

For the purpose of the second part of this assignment I chose to perform a regression analysis on the dataset I chose. Following are the steps taken for the purpose of the analysis:

# Setting Seed for reproducible research  
set.seed(695949)  
  
# Creating correlation matrix  
TG\_cor<-cor(data.matrix(TG))  
TG\_cor

## len supp dose  
## len 1.0000000 -0.2438927 0.8026913  
## supp -0.2438927 1.0000000 0.0000000  
## dose 0.8026913 0.0000000 1.0000000

The above correlation matrix shows us that there is a strong positive correlation between len and dose variables and also a lesser negative correlation between supp and len variables. We hence try to find the linear combos to see if we should possibly remove supp from the linear regression model.

#Checking which variables can be removed  
findLinearCombos(data.matrix(TG))

## $linearCombos  
## list()  
##   
## $remove  
## NULL

The findLinearCombos() function did not throw out supp variable from the model (though this might be because we converted the entire data frame into a numeric matrix). We now begin to partition the data as follows:

# Creating Training and Test Data partitions  
TG\_sv<-createDataPartition(TG$dose, p=.75,list=FALSE)  
TG\_train<-TG[TG\_sv,]  
TG\_test<-TG[-TG\_sv,]

We build the model as follows(we consider the interaction between len and supp as the correlation was significant enough to be considered initially and we can also verify the need to include the interaction in the following part):

# Creating the linear model  
TG\_lm1<-lm(dose~len+supp+len\*supp, data = TG)  
  
# Details of the linear model  
summary(TG\_lm1)

##   
## Call:  
## lm(formula = dose ~ len + supp + len \* supp, data = TG)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.72692 -0.26911 0.00158 0.21805 0.72733   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.321530 0.220467 -1.458 0.150   
## len 0.072021 0.010179 7.076 2.59e-09 \*\*\*  
## suppVC 0.318064 0.268353 1.185 0.241   
## len:suppVC -0.003041 0.013029 -0.233 0.816   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3621 on 56 degrees of freedom  
## Multiple R-squared: 0.6854, Adjusted R-squared: 0.6685   
## F-statistic: 40.66 on 3 and 56 DF, p-value: 4.381e-14

The summary statistics tells us that we do not need to take into account the interaction between supp and len and so we rebuild the model as follows:

# Rebuilding the model  
TG\_lm2<-lm(dose~len+supp, data = TG)  
  
# Details of the new model  
summary(TG\_lm2)

##   
## Call:  
## lm(formula = dose ~ len + supp, data = TG)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.72537 -0.26228 0.00877 0.22246 0.72551   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.283181 0.145777 -1.943 0.05702 .   
## len 0.070165 0.006301 11.135 6.31e-16 \*\*\*  
## suppVC 0.259611 0.095595 2.716 0.00874 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3591 on 57 degrees of freedom  
## Multiple R-squared: 0.6851, Adjusted R-squared: 0.674   
## F-statistic: 61.99 on 2 and 57 DF, p-value: 5.005e-15

As seen from the summary statistics, we do not need to remove any varaibales and should confirm usign the anova() function as follows:

# Performing anova  
anova(TG\_lm2,TG\_lm1)

## Analysis of Variance Table  
##   
## Model 1: dose ~ len + supp  
## Model 2: dose ~ len + supp + len \* supp  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 57 7.3485   
## 2 56 7.3414 1 0.0071414 0.0545 0.8163

As seen from the anova table, by removing the interaction between supp and len, we have created a model with fewer degress of freedom that can be considered as a better fit. We now move on to predicting the values and comparing with the data as follows:

# Predicting the with the test dataset  
TG\_p<-predict(TG\_lm2, TG\_test)  
  
# Observing the results  
summary(TG$dose)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.500 0.500 1.000 1.167 2.000 2.000

summary(TG\_train$dose)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.500 0.500 1.000 1.167 2.000 2.000

summary(TG\_test$dose)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.500 0.500 1.000 1.167 2.000 2.000

summary(TG\_p)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3413 0.7728 1.1340 1.1750 1.5620 2.0460

The above summaries tells us a few things about the model. While the model is not as accurate, the difference between actual and predicted values seems to be small enough to be considered as a good model. Additionally, further training of the model through a larger dataset may be required for the residuals to be minimized.

* **Tell me why you prefer RStudio over Rattle or the opposite or some other tool of your choice over Rstudio. Just looking for opinions here, but I am curious HOW you evaluate the different tools.**

I prefer RStudio over rattle mostly because of the flexibility of the commands and the understanding I can obtain through the documentation. This may be due to my background in computer science, but I like to have the ability to know about the finer details of the functions being executed and having the ability to pass parameters to modify the output as I need. Again, this may be because I am just used to looking through code. That being said, I do find Rattle to be a useful tool to obtain quick results. However, for me, it would be something I would use to get get brief overviews of the data and perform quick analysis before trying to get down to the actual construction of models or graphics.