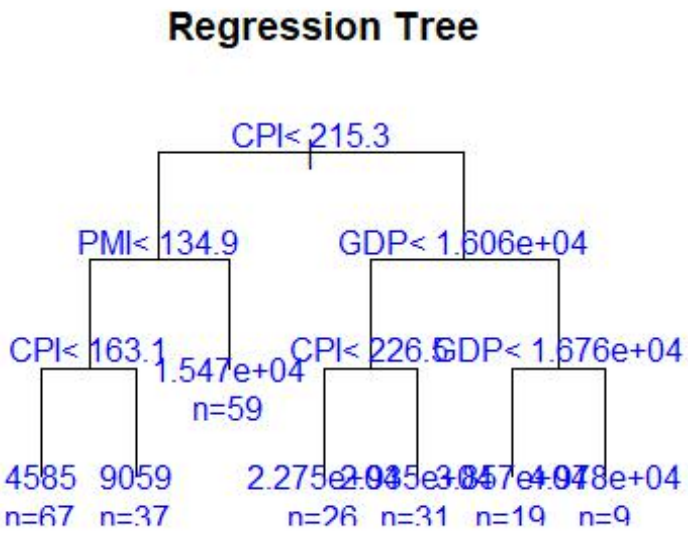
Regression Tree

```
clipboard <-read.table("clipboard",header=T)
set.seed(1)
samp_20_proj<-sample(nrow(clipboard),.2*nrow(clipboard))
data_train<-clipboard[-samp_20_proj,]
data_test_x<-clipboard[,-12]
data_test_y<-clipboard[,12]

library(rpart)

## Warning: package 'rpart' was built under R version 3.4.3

tree_project<-rpart(Sales~.,data=data_train,method='anova',control=rpar
t.control(maxdepth=8,minbucket=5,minsplit=5,cp=0.01))
printcp(tree_project)

##
## Regression tree:
## rpart(formula = Sales ~ ., data = data_train, method = "anova",
##     control = rpart.control(maxdepth = 8, minbucket = 5, minsplit = 5,

##         cp = 0.01))
##
## Variables actually used in tree construction:
## [1] CPI GDP PMI
##
## Root node error: 3.9894e+10/248 = 160862844
##
## n= 248
##
##         CP nsplit rel error  xerror     xstd
## 1 0.678671      0  1.000000 1.00840 0.110100
## 2 0.117994      1  0.321329 0.34158 0.046807
## 3 0.081430      2  0.203336 0.26693 0.036158
## 4 0.019241      3  0.121906 0.17903 0.033377
## 5 0.015438      4  0.102665 0.17000 0.029029
## 6 0.011962      5  0.087227 0.15179 0.024955
## 7 0.010000      6  0.075265 0.13332 0.023685
```

```
treefit<-prune(tree_project, cp=tree_project$cptable[which.min(tree_pro
ject$cptable[,'xerror'])])
printcp(treefit)

##
## Regression tree:
## rpart(formula = Sales ~ ., data = data_train, method = "anova",
##     control = rpart.control(maxdepth = 8, minbucket = 5, minsplit = 5,

##         cp = 0.01))
##
## Variables actually used in tree construction:
## [1] CPI GDP PMI
##
## Root node error: 3.9894e+10/248 = 160862844
##
## n= 248
##
##         CP nsplit rel error  xerror     xstd
## 1 0.678671      0  1.000000 1.00840 0.110100
## 2 0.117994      1  0.321329 0.34158 0.046807
## 3 0.081430      2  0.203336 0.26693 0.036158
## 4 0.019241      3  0.121906 0.17903 0.033377
## 5 0.015438      4  0.102665 0.17000 0.029029
## 6 0.011962      5  0.087227 0.15179 0.024955
## 7 0.010000      6  0.075265 0.13332 0.023685

plot(treefit,uniform=T, branch=1, margin=0.1, main="Regression Tree")
text(treefit,use.n=T, col="blue")

#cross-validation
library(caret)

## Warning: package 'caret' was built under R version 3.4.3

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.4.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.2
```




```
split=0.80
train_control <- trainControl(method="cv", number=5)
treefit2<- train(Sales~PMI+GDP+CPI, data=data_train, trControl=train_co
ntrol, method="rpart",parms=list(method='anova'))

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measu
res.

print(treefit2)

## CART
##
## 248 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 199, 198, 198, 199, 198
## Resampling results across tuning parameters:
##
##   cp          RMSE      Rsquared   MAE
##   0.08142956  5316.545  0.8345271  3891.193
```

```
##   0.11799360  7087.825  0.6952698  5620.969
##   0.67867064  9532.967  0.6554518  7679.411
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.08142956.
```

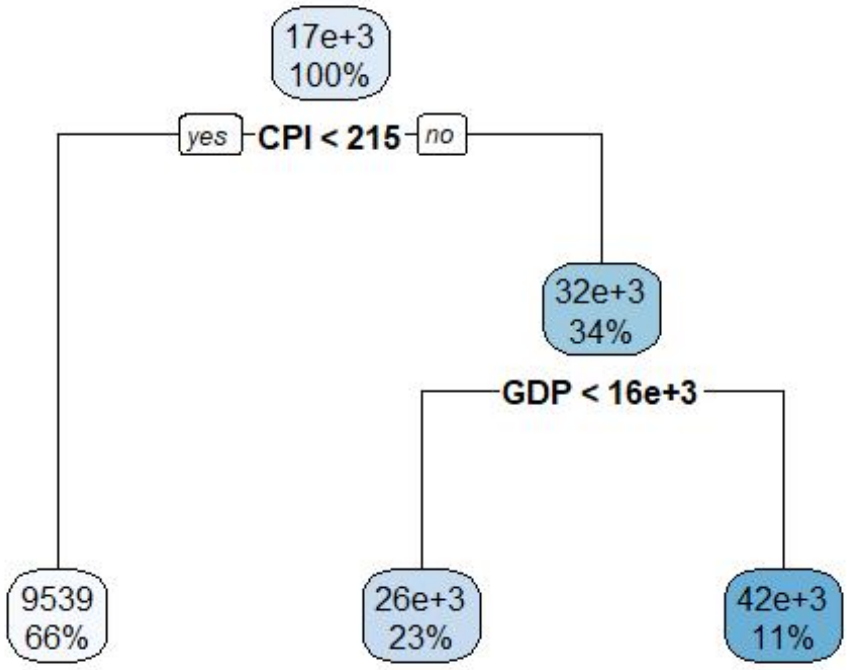
```
treefit2$finalModel
```

```
## n= 248
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 248 39893990000 17084.660
##   2) CPI< 215.325 163  4573205000  9539.417 *
##   3) CPI>=215.325 85  8245904000 31553.760
##     6) GDP< 16061.24 57  1474141000 26338.040 *
##     7) GDP>=16061.24 28  2064528000 42171.500 *
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.4.3
```

```
rpart.plot(treefit2$finalModel)
```




```
##test
proj_pred<-predict(treefit2,data_test_x)
(rss_pred<- mean((data_test_y-proj_pred)^2))
```

```
## [1] 32598807
```

```
rsquare<-function(true,predicted){
  sse<-sum((predicted-true)^2)
  sst<-sum((true-mean(true))^2)
  rsq<-1-sse/sst
}
Rsquare<-rsquare(data_test_y,proj_pred)
```