data %>%   
 head()

## # A tibble: 6 x 30  
## roe net\_profit\_ratio gross\_profit\_rate net\_profits eps  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 18.53 8.74 29.4144 21022.6063 1.9043  
## 2 32.52 13.66 34.6221 39.2993 0.4679  
## 3 15.55 23.18 29.1261 778.5809 0.5767  
## 4 21.05 200.76 66.1915 78.0190 0.3377  
## 5 8.53 27.81 52.7716 521.8662 0.1857  
## 6 5.21 3.63 30.6634 233.3839 0.1085  
## # ... with 25 more variables: business\_income <dbl>, bips <dbl>,  
## # arturnover <dbl>, arturndays <dbl>, inventory\_turnover <dbl>,  
## # inventory\_days <dbl>, currentasset\_turnover <dbl>,  
## # currentasset\_days <dbl>, mbrg <dbl>, nprg <dbl>, nav <dbl>,  
## # targ <dbl>, epsg <dbl>, seg <dbl>, currentratio <dbl>,  
## # quickratio <dbl>, cashratio <dbl>, icratio <chr>, sheqratio <dbl>,  
## # adratio <dbl>, cf\_sales <dbl>, rateofreturn <dbl>, cf\_nm <dbl>,  
## # cf\_liabilities <dbl>, cashflowratio <dbl>

看到下面的数据结构，模型的异常值非常多，很多分布是不均匀的。

skim(data)

## Skim summary statistics  
## n obs: 2842   
## n variables: 30   
##   
## Variable type: character   
## variable missing complete n min max empty n\_unique  
## icratio 0 2842 2842 2 12 0 2824  
##   
## Variable type: numeric   
## variable missing complete n mean sd p0  
## adratio 0 2842 2842 39.37 19.35 1.74   
## arturndays 0 2842 2842 90.1 81.32 1e-04   
## arturnover 0 2842 2842 1979.4 1e+05 0.5   
## bips 0 2842 2842 6.23 9.69 0.056   
## business\_income 0 2842 2842 8487.83 58471.96 13.18   
## cashflowratio 0 2842 2842 28.89 63.68 -2128.18   
## cashratio 0 2842 2842 96.6 152.67 0.7   
## cf\_liabilities 0 2842 2842 0.23 0.56 -21.28   
## cf\_nm 0 2842 2842 1.82 16.27 -102.88   
## cf\_sales 0 2842 2842 0.099 0.38 -8.25   
## currentasset\_days 0 2842 2842 487.46 1635.06 17.01   
## currentasset\_turnover 0 2842 2842 1.28 1.03 0.0044  
## currentratio 0 2842 2842 2.67 2.99 0.06   
## eps 0 2842 2842 0.51 0.6 -0.2   
## epsg 0 2842 2842 53.1 573.28 -1100   
## gross\_profit\_rate 0 2842 2842 32.41 16.94 -29.51   
## inventory\_days 0 2842 2842 185.15 370.75 0.0058  
## inventory\_turnover 0 2842 2842 63.93 1277.78 0.042   
## mbrg 0 2842 2842 34.24 281.29 -95.59   
## nav 0 2842 2842 67.32 1198.37 -36.55   
## net\_profit\_ratio 0 2842 2842 12.72 25.32 -4.43   
## net\_profits 0 2842 2842 464.78 1757.66 -2623.33   
## nprg 0 2842 2842 1530.02 76660.74 -99.64   
## quickratio 0 2842 2842 2.16 2.78 0.059   
## rateofreturn 0 2842 2842 0.06 0.08 -0.46   
## roe 0 2842 2842 10.13 9.53 -288.34   
## seg 0 2842 2842 67.03 1196.74 -72.51   
## sheqratio 0 2842 2842 60.63 19.35 1.51   
## targ 0 2842 2842 52.65 728.71 -47.79   
## p25 median p75 p100 hist  
## 23.78 38.08 53.4 98.49 ▃▇▇▇▆▃▁▁  
## 31.74 71.65 122.16 718.42 ▇▃▁▁▁▁▁▁  
## 2.95 5.02 11.34 5501149.24 ▇▁▁▁▁▁▁▁  
## 2.1 3.93 7.02 246.19 ▇▁▁▁▁▁▁▁  
## 651.07 1534.69 4095.42 1930911 ▇▁▁▁▁▁▁▁  
## 5.47 19.49 42.41 453.66 ▁▁▁▁▁▁▇▁  
## 28.7 52.62 104.7 2899.54 ▇▁▁▁▁▁▁▁  
## 0.044 0.15 0.34 3.57 ▁▁▁▁▁▁▇▂  
## 0.42 1.05 1.84 677.47 ▁▇▁▁▁▁▁▁  
## 0.033 0.11 0.2 3.25 ▁▁▁▁▁▇▁▁  
## 229.54 348.96 519.57 81818.18 ▇▁▁▁▁▁▁▁  
## 0.69 1.03 1.57 21.16 ▇▁▁▁▁▁▁▁  
## 1.28 1.85 3.02 50.14 ▇▁▁▁▁▁▁▁  
## 0.17 0.36 0.66 17.9 ▇▁▁▁▁▁▁▁  
## -26.9 4.74 32.24 19900 ▇▁▁▁▁▁▁▁  
## 20.6 29.82 41.01 99.94 ▁▁▃▇▅▂▁▁  
## 51.39 94.84 174.33 8633.09 ▇▁▁▁▁▁▁▁  
## 2.07 3.8 7.01 61592.09 ▇▁▁▁▁▁▁▁  
## 1.47 12.96 29.26 8748.37 ▇▁▁▁▁▁▁▁  
## 5.07 13.04 39.33 63009.32 ▇▁▁▁▁▁▁▁  
## 4.77 9.47 15.71 972.76 ▇▁▁▁▁▁▁▁  
## 60.13 128.67 328.39 46416 ▇▁▁▁▁▁▁▁  
## -3.97 17.66 50.96 4086723.17 ▇▁▁▁▁▁▁▁  
## 0.88 1.39 2.46 45.59 ▇▁▁▁▁▁▁▁  
## 0.018 0.057 0.1 0.66 ▁▁▁▇▃▁▁▁  
## 4.84 8.51 13.57 66.43 ▁▁▁▁▁▁▇▁  
## 4.76 11.93 39.27 63009.32 ▇▁▁▁▁▁▁▁  
## 46.6 61.92 76.22 98.26 ▁▁▃▆▇▇▇▃  
## 5.08 15.1 36.24 37643.75 ▇▁▁▁▁▁▁▁

两个.R文档编码不一样，所以我合并了。

# 决策树

library(rpart)  
library(rpart.plot)  
library(lattice)  
library(caret)

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

set.seed(1000)  
train\_index <- sample(nrow(data),0.7\*nrow(data),replace = F)  
train <- data[train\_index,]  
test <- data[-train\_index,]  
nrow(train)

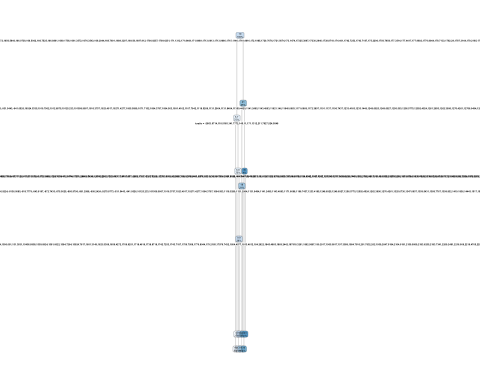
## [1] 1989

nrow(test)

## [1] 853

bfit <- rpart(roe ~ ., train,method = "anova")  
rpart.plot(x = bfit)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



roe是变量

labs do not fit even at cex 0.15, there may be some overplotting 这是警告信息，说明模型需要剪枝，过拟合了。

这里使用到变量少的原因是

rpart.control(minsplit = 20, minbucket = round(minsplit/3), cp = 0.01,   
 maxcompete = 4, maxsurrogate = 5, usesurrogate = 2, xval = 10,  
 surrogatestyle = 0, maxdepth = 30, ...)

minsplit在分之前，每个node上必须拥有的样本量最小值。 因此这些防止过拟合的方式，使得树不能太大，因此有些变量就没有用到。 但是你看看通过bagging的随机森林就不会，它使用过很多次，因此理论上，每个变量都可能会用到，因此Importances表上基本上反馈的变量就多了。

# 随机森林

library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

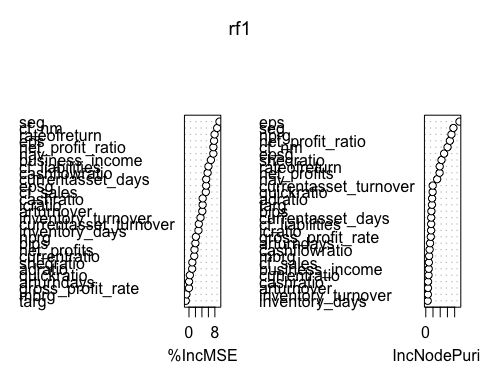
rf1 <- randomForest(  
 roe ~ .,   
 data=train,   
 importance=TRUE,   
 mtry=2,   
 ntree=100,  
 proximity=TRUE)  
print(rf1)

##   
## Call:  
## randomForest(formula = roe ~ ., data = train, importance = TRUE, mtry = 2, ntree = 100, proximity = TRUE)   
## Type of random forest: regression  
## Number of trees: 100  
## No. of variables tried at each split: 2  
##   
## Mean of squared residuals: 61.59078  
## % Var explained: 39.74

print(importance(rf1,type = 2))

## IncNodePurity  
## net\_profit\_ratio 16690.869  
## gross\_profit\_rate 3465.857  
## net\_profits 9562.283  
## eps 23326.161  
## business\_income 2026.526  
## bips 3869.956  
## arturnover 1537.291  
## arturndays 2961.129  
## inventory\_turnover 1445.427  
## inventory\_days 1340.517  
## currentasset\_turnover 4788.844  
## currentasset\_days 3739.589  
## mbrg 2313.286  
## nprg 17774.272  
## nav 7869.884  
## targ 4275.371  
## epsg 12694.804  
## seg 19121.034  
## currentratio 1908.771  
## quickratio 4733.161  
## cashratio 1576.204  
## icratio 3691.276  
## sheqratio 10901.320  
## adratio 4357.624  
## cf\_sales 2137.444  
## rateofreturn 9651.142  
## cf\_nm 15083.234  
## cf\_liabilities 3695.545  
## cashflowratio 2657.098

varImpPlot(rf1)



模型结果出来了，但是我觉得效果不好。 看参数结果。 Type of random forest: regression，但是只有39.43。 因此如果你是完全做预测的话，需要还要调整参数。

你看参考这个[日志](https://jiaxiangli.netlify.com/2018/01/machine-learning-with-tree-based-models-in-r/)调整参数。

# 导出.R文档