Lab 3: CART Analysis

Trang Nguyen, Dennis Liu

3/9/2017

## Objective

Today's lab is the last one in the Lab Series of **Bad Practices of Experimental Designs and Statistical Analysis**. This lab introduces *CART Analysis*, also known as *Classification and Regression Tree*, as an alternative to overcome some limitations of *T-Test* and *ANOVA*.

## About CART Analysis

CART refers to Classification adn Regression Tree, which are the two techniques that use decision tree to predict the response variables by ranking the importance of each given explanatory variable and recursively splitting levels of each variable. **Classification Tree** is used when the response variable is categorical while **Regression Tree** is used when the response variable is continuous.  
This method is commonly used in the industry because it is simple to understand and interpret. Most importantly, it does not require any assumptions of the data.

## Lab Activity

### 1. Getting Data

This lab has the same data setup as the previous lab, *Lab 02: ANOVA*.

library(readr)  
gen\_tangram <- read\_csv("gentangram.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## X1 = col\_integer(),  
## `Player Alias` = col\_integer(),  
## NumShapes = col\_integer(),  
## RequestedTime = col\_integer(),  
## TimeUsed = col\_double(),  
## TimerDisplay = col\_integer(),  
## TimerHint = col\_double(),  
## NumClicks = col\_integer(),  
## Won = col\_integer(),  
## HintsEnabled = col\_integer(),  
## HintsUsed = col\_integer(),  
## Timestamp = col\_datetime(format = ""),  
## TimeUsed\_log = col\_double(),  
## factor\_gender = col\_integer()  
## )

## See spec(...) for full column specifications.

puzzlename <- c("A Nice Lighthouse", "Diamond", "House of Tangrams", "The Hook")  
gen\_tangram$PuzzleName2 <- ifelse(gen\_tangram$PuzzleName %in% puzzlename, gen\_tangram$PuzzleName, "Others")  
gen\_tangram$PuzzleName2 <- as.factor(gen\_tangram$PuzzleName2)  
gen\_tangram$TimeUsed\_log <- log(gen\_tangram$TimeUsed)  
gen\_tangram$level\_gender <- as.factor(gen\_tangram$level\_gender)

### 2. Conducting CART analysis

This lab uses **party**, which is the most updated R package for decision trees. The advantage of this package compared to previous ones is that it implements conditional inference trees rather the traditional decision tree algorithms. The traditional approach tends to select variables that have many possible splits or missing values so it will bias towards categorical variables with more levels. Conditional inference tree can avoid this bias by using the signicance test procedure. You can refer to the paper [*Unbiased Recursive Paritioning: A Conditional Inference Framework*] (<http://statmath.wu-wien.ac.at/~zeileis/papers/Hothorn+Hornik+Zeileis-2006.pdf>) (Hothon, Hornik, & Zeileis) for more details.

The follow code conducts the decision tree based on 4 variables: *Puzzle Name*, *Gender*, *Number of Hints Used*, *Whether Hints are Enabled*.

library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

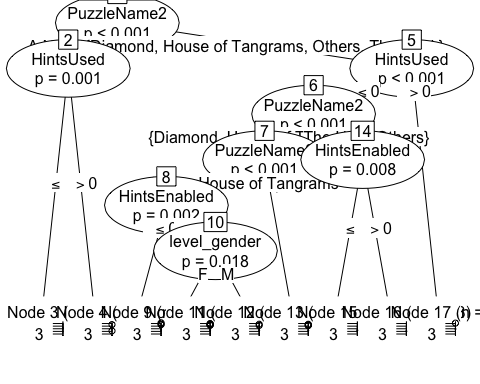
## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

fit <- ctree(TimeUsed\_log ~ level\_gender + PuzzleName2 + HintsUsed + HintsEnabled,   
 data=na.omit(gen\_tangram))  
plot(fit)



Question 1: According to the plot of the decision tree, what can you conclude about the influence of Gender in the Game Completion Time?

Question 2: Try adding or removing one variable from the model and compare the result with the previous model.

## Best Practice for Using CART analysis

Like other machine learning algorithms, CART analysis can face the problem of overfitting the data, which means that the built decision tree can only be applied to this specific data set. To minimize this problem, people often split the data set into *train* and *test* data set so that the model can be trained by the *train* data set and then be tested on the *test* data set.