Classification and Regression Trees (CART)

Regression Trees: Ozone data

This example is adapted from the book "Extending the Linear Model with R", by Julian J. Faraway. Here is a quote from that book describing the data:

> We apply the regression tree methodology to study the relationship between atmospheric ozone concentration and meteorology in the Los Angeles Basin in 1976. A number of cases with missing variables hve been removed for simplicity [but Evan notes that trees are among the regression and classification methods that are best able to handle missing data]. We wish to predict the ozone level from the other predictors.

The variables in the data set are as follows:

- `O3`: Ozone concentration (ppm) at Sandbug Air Force Base
- `vh`: Vandenburg 500 millibar height (inches)
- `wind`: wind speed (miles per hour)
- `humidity`: humidity (percent)
- `temp`: temperature (degrees C)
- `ibh`: inversion base height (feet)
- `dpg`: Daggett pressure gradient (mmhg)
- `ibt`: inversion base temperature (degrees F)
- `vis`: visibility (miles)
- `doy`: day of the year

```
library(faraway)
```

```
##
## Attaching package: 'faraway'
```

```
## The following object is masked from 'package:GGally':
##
##     happy
```

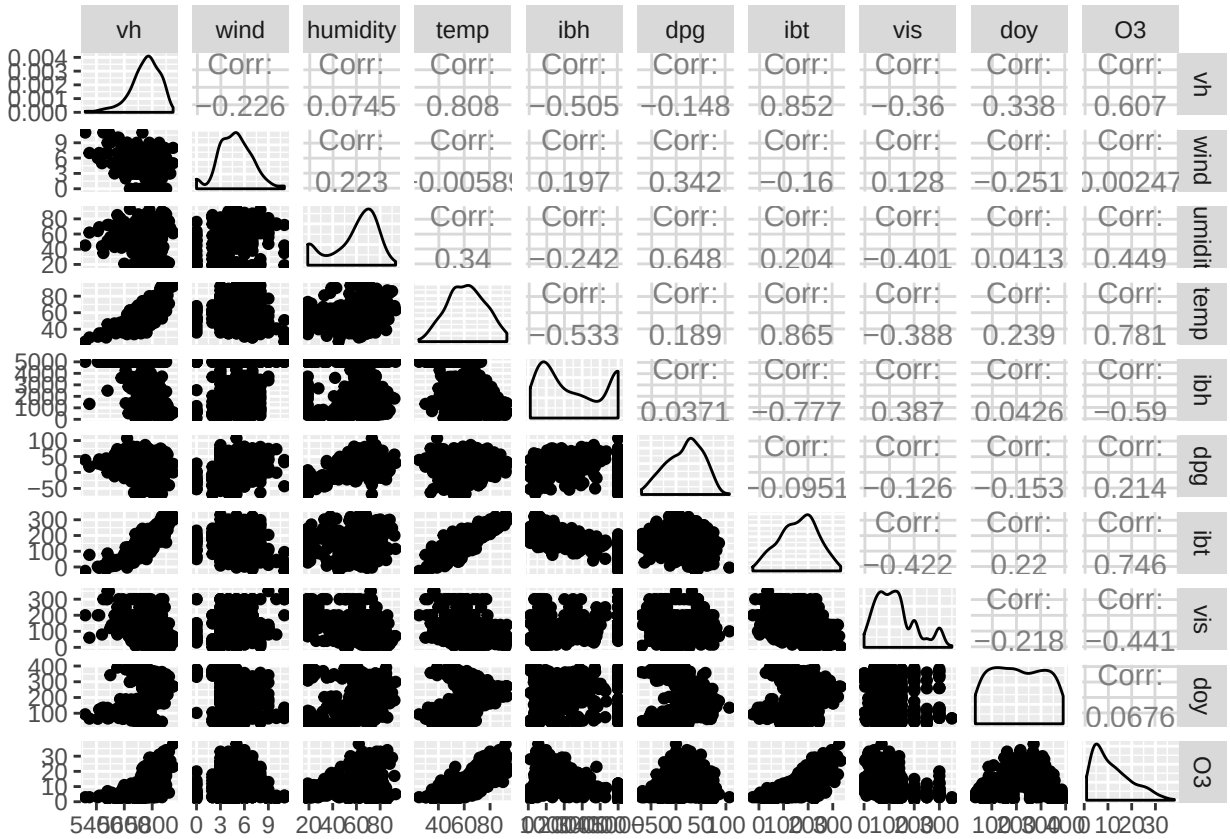
```
head(ozone)
```

```
##   O3   vh wind humidity temp  ibh dpg ibt vis doy
## 1  3 5710    4       28   40 2693 -25  87 250  33
## 2  5 5700    3       37   45  590 -24 128 100  34
## 3  5 5760    3       51   54 1450  25 139  60  35
## 4  6 5720    4       69   35 1568  15 121  60  36
## 5  4 5790    6       19   45 2631 -33 123 100  37
## 6  4 5790    3       25   55  554 -28 182 250  38
```

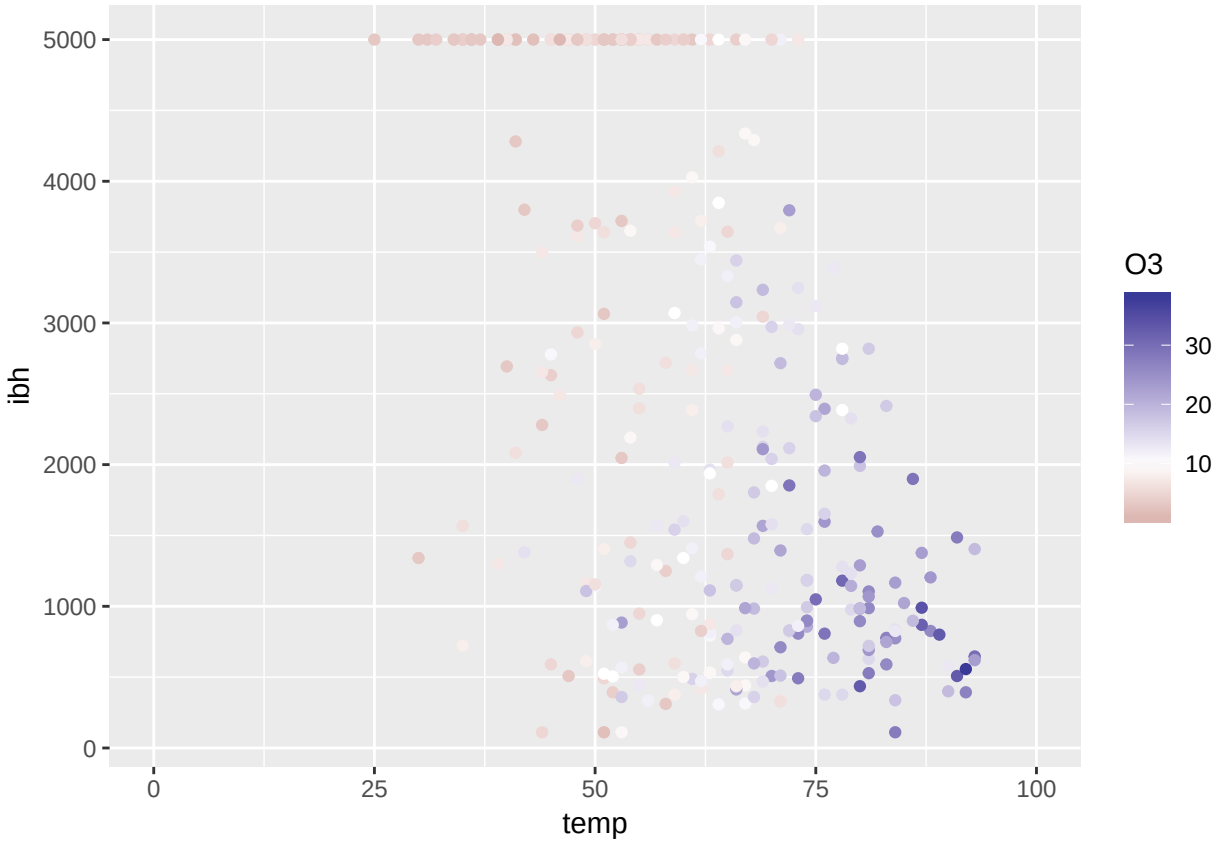
```
dim(ozone)
```

```
## [1] 330  10
```

```
ggpairs(ozone %>% select(vh:doy, O3))
```




```r
ggplot(data = ozone, mapping = aes(x = temp, y = ibh, color = O3)) +
  geom_point() +
  scale_color_gradient2(midpoint = median(ozone$O3)) +
  xlim(c(0, 100))
```



Regression tree with 2 variables:

```
# Load rpart package (stands for "Recursive Partitioning")
library(rpart)
```

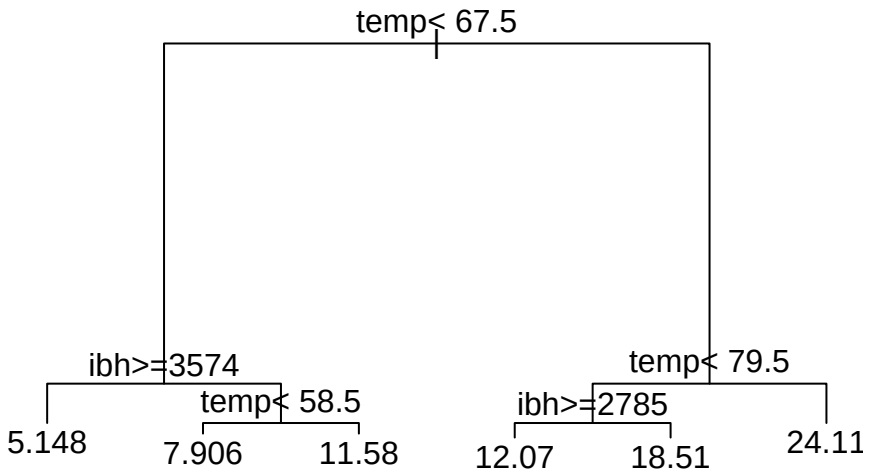
```
##
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:faraway':
##
##     solder
```

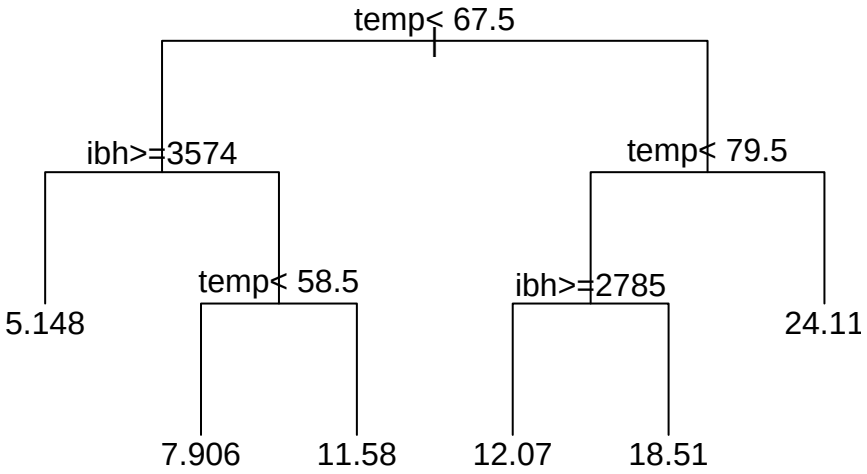
```
# fit regression tree model
roz <- rpart(O3 ~ temp + ibh, data = ozone)
print(roz)
```

```
## n= 330
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 330 21115.4100 11.775760
##    2) temp< 67.5 214  4114.3040  7.425234
##      4) ibh>=3573.5 108   689.6296  5.148148 *
##      5) ibh< 3573.5 106  2294.1230  9.745283
##       10) temp< 58.5 53  1008.5280  7.905660 *
##       11) temp>=58.5 53   926.8679 11.584910 *
##    3) temp>=67.5 116  5478.4400 19.801720
##      6) temp< 79.5 72  2626.0000 17.166670
##       12) ibh>=2785 15   348.9333 12.066670 *
##       13) ibh< 2785 57  1784.2460 18.508770 *
##      7) temp>=79.5 44  1534.4320 24.113640 *
```

```
# print first picture of resulting tree
plot(roz, margin = 0.1)
text(roz)
```




```
# print second picture of resulting tree
plot(roz, margin = 0.1, uniform = TRUE)
text(roz)
```



Space for two pictures:

What's the predicted ozone level for a day with a temperature of 75 degrees and an inversion base height of 2000 feet?

```
test_data <- data.frame(
  temp = 75, ibh = 2000
)

predict(roz, newdata = test_data)
```

```
##        1
## 18.50877
```

Notation/Mathematical Description:

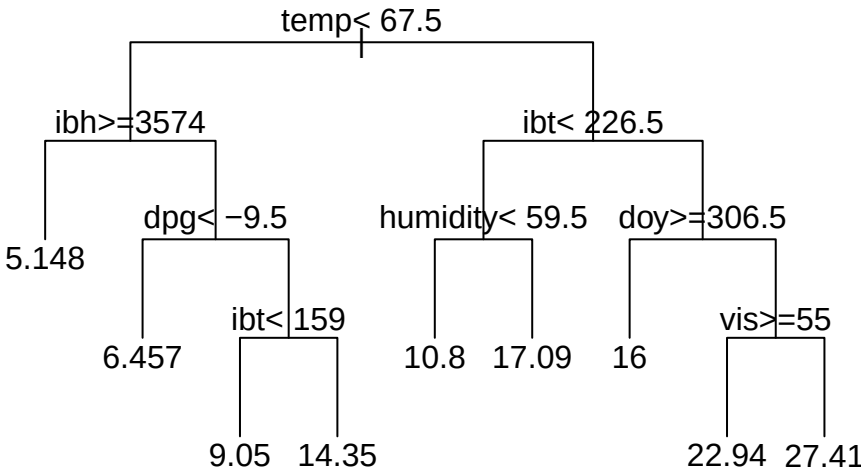
$\widehat{f}(x_i) = \sum_{m=1}^{|T|} I_{R_m}(x_i)\hat{y}_m$

- $|T|$ is the number of *terminal nodes* in the tree.
- R_m is the set of values for explanatory variables that fall into the mth terminal node. (R for region or rectangle)
- $I_{R_m}(x_i) = \begin{cases} 1 \text{ if } x_i \text{ is in region } R_m \\ 0 \text{ otherwise} \end{cases}$
- $\hat{y}_m$ is the predicted value in terminal node number m.

More covariates:

```
roz_2 <- rpart(O3 ~ ., data = ozone)

# print picture of resulting tree
plot(roz_2, margin = 0.1, uniform = TRUE)
text(roz_2)
```



Classification Trees: Heart Disease data

We have data on 303 patients who presented with chest pain. The response variable is `AHD`, which is is "Yes" if an angiogrphic test indicates presence of heart disease, and "No" otherwise. There are 13 predictor variables which are a mix of quantitative and categorical variables.

```
library(readr)
heart <- read_csv("http://www.evanlray.com/data/islr/Heart.csv") %>%
  select(-X1) %>% # drop leading column of row numbers
  mutate_at(c("Sex", "ChestPain", "Fbs", "RestECG", "ExAng", "Slope", "Thal", "AHD"), factor)
```

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

```
head(heart)
```

```
## # A tibble: 6 x 14
##     Age Sex   ChestPain RestBP  Chol Fbs   RestECG MaxHR ExAng Oldpeak
##   <int> <fct> <fct>      <int> <int> <fct> <fct>   <int> <fct>   <dbl>
## 1    63 1     typical      145   233 1     2         150 0         2.3
## 2    67 1     asymptom~    160   286 0     2         108 1         1.5
## 3    67 1     asymptom~    120   229 0     2         129 1         2.6
## 4    37 1     nonangin~    130   250 0     0         187 0         3.5
## 5    41 0     nontypic~    130   204 0     2         172 0         1.4
## 6    56 1     nontypic~    120   236 0     0         178 0         0.8
## # ... with 4 more variables: Slope <fct>, Ca <int>, Thal <fct>, AHD <fct>
```

```
# too many variables for a pairs plot to be effective
# instead, plot each explanatory variable by response
make_plot <- function(explanatory_var) {
  if(is.numeric(heart[[explanatory_var]])) {
    p <- ggplot(data = heart, mapping = aes_string(x = explanatory_var, color = "AHD")) + geom_density()
  } else {
    p <- ggplot(data = heart, mapping = aes_string(x = explanatory_var, fill = "AHD")) + geom_bar()
  }

  return(p)
}

ncol(heart)
```

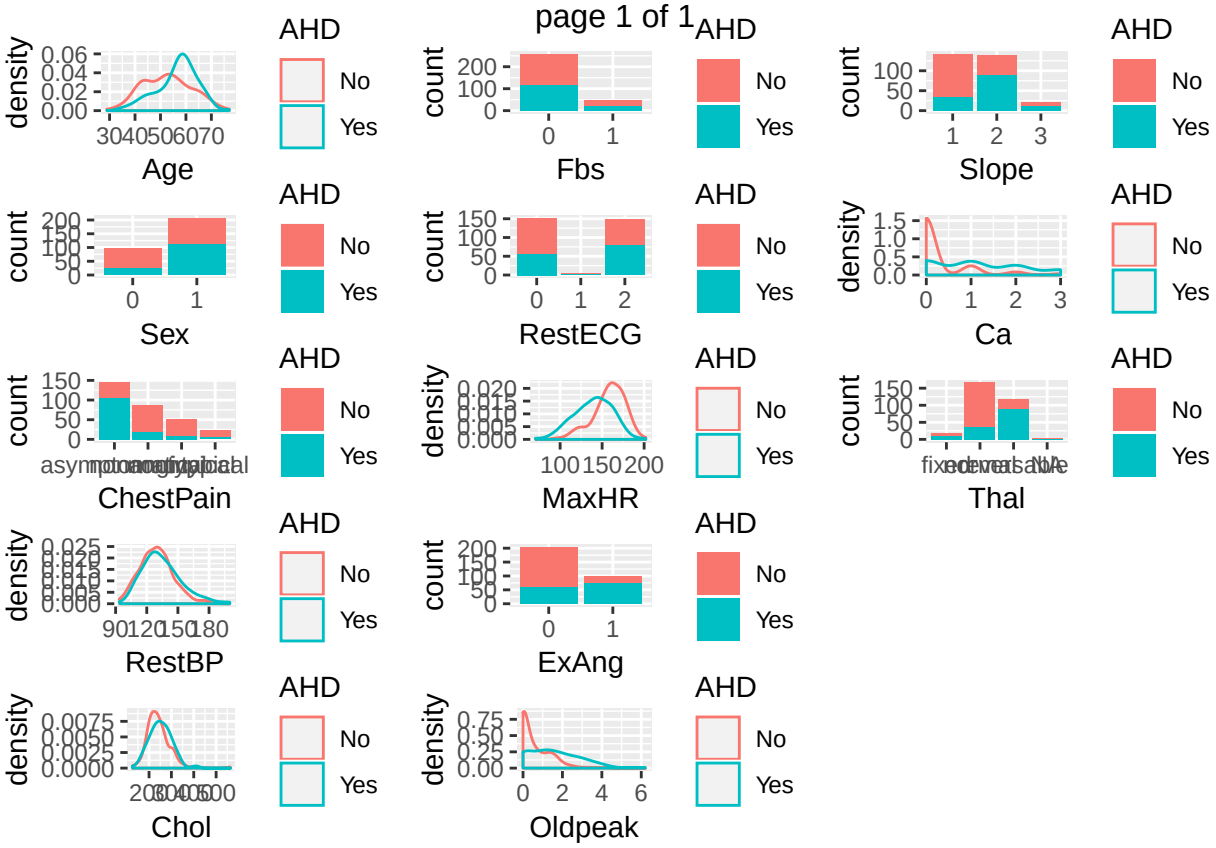
```
## [1] 14
```

```
plots <- map(colnames(heart)[1:13], make_plot)

p_combined <- marrangeGrob(grobs = plots, ncol = 3, nrow = 5)
```

```
## Warning: Removed 4 rows containing non-finite values (stat_density).
```

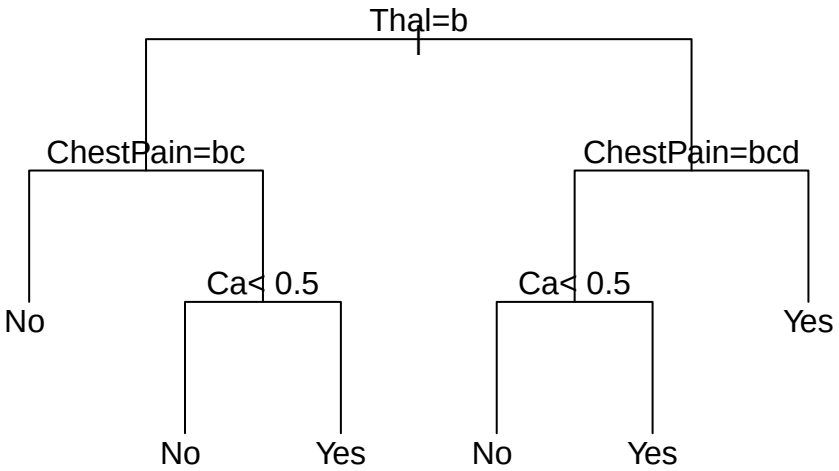
```
print(p_combined)
```




```
## [[1]]
## NULL
```

```
rhd <- rpart(AHD ~ ., data = heart)

# print second picture of resulting tree
plot(rhd, margin = 0.1, uniform = TRUE)
text(rhd)
```




```
levels(heart$Thal)
```

```
## [1] "fixed"      "normal"     "reversable"
```

```
levels(heart$ChestPain)
```

```
## [1] "asymptomatic" "nonanginal"   "nontypical"   "typical"
```

What's the predicted class for someone whose Thallium stress test results are normal, whose chest pain symptoms are typical, and has a "Ca" (not sure what that stands for) of 2?

Overview of Estimation

Reminder of Notation

Recall our mathematical formulation:

$\widehat{f}(x_i) = \sum_{m=1}^{|T|} I_{R_m}(x_i)\hat{y}_m$

- $|T|$ is the number of *terminal nodes* in the tree.
- R_m is the set of values for explanatory variables that fall into the mth terminal node. (R for region or rectangle)
- $I_{R_m}(x_i) = \begin{cases} 1 \text{ if } x_i \text{ is in region } R_m \\ 0 \text{ otherwise} \end{cases}$
- $\hat{y}_m$ is the predicted value in terminal node number m.

Parameters to Estimate

- **Split Points**: for each branch of the tree, what covariate is used and at which value does the split occur? This determines the R_m.
- **Regression Constants**: In each terminal node, what is the predicted value $\hat{y}_m$?

Optimization target for regression

$$RSS = \sum_{i=1}^{n}(\widehat{y}_i - y_i)^2 = \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_m)^2$$

Optimization target for classification

We could use classification error rate, but in practice other options are used more frequently.

For node m, let p_{jm} be the proportion of the observations in node m that have class j.

One common choice is the **Gini index**: $1 - \sum_J p_{jm}^2$

Top-down Estimation Algorithm (discussed for regression; classification similar)

1. Start with a tree with only one region/terminal node (the same prediction is made for all observations).
   - Predicted value is mean of all observations
   - Calculate the RSS based on that prediction
2. Repeat the following until a stopping criterion is met:
   a. For every terminal node in the current tree, consider all possible split points for each covariate $X_1, \ldots, X_p$.
      - For each possible split, predicted values will be the mean of all observations in the newly created leaves
      - Calculate RSS based on that split
   b. Select the split that achieved the lowest RSS

Commonly used stopping criteria:

- All regions contain 5 or fewer observations (for example)
- A maximum number of terminal nodes has been reached
- No improvement in RSS larger than ε can be realized

Regularizing number of terminal nodes

A smaller number of terminal nodes is less likely to overfit. Choose tree to minimize:

$RSS + \lambda|T|$

Use cross-validation to select λ.