Regression Trees

Ex: MLB

Can we predict the `Salary` of a MLB player based on the number of:

- `Years` that he has been in the league
- `Hits` that he made in the previous season

Can we predict player salaries?

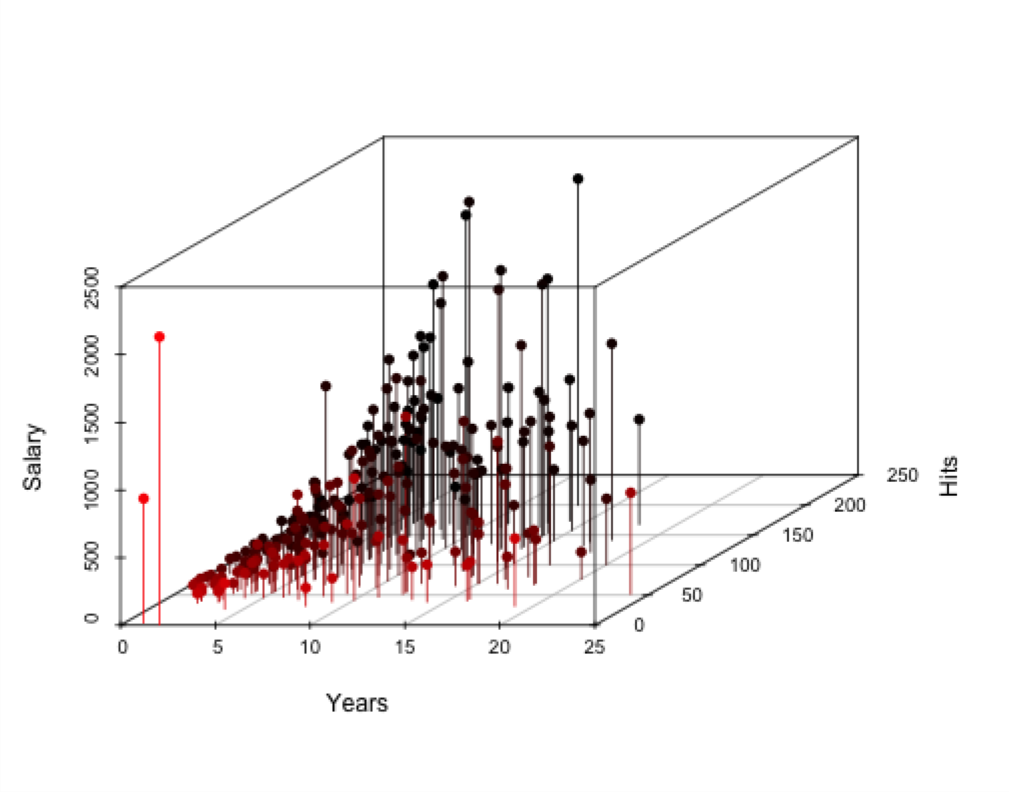
```r
library(ISLR)
dim(Hitters)
```

```
## [1] 322  20
```

```r
names(Hitters)
```

```
##  [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "
##  [6] "Walks"     "Years"     "CAtBat"    "CHits"     "
## [11] "CRuns"     "CRBI"      "CWalks"    "League"    "
## [16] "PutOuts"   "Assists"   "Errors"    "Salary"    "
```

Exploratory Data Analysis



Looks like a good setting for . . . **regression**. Maybe a linear model?

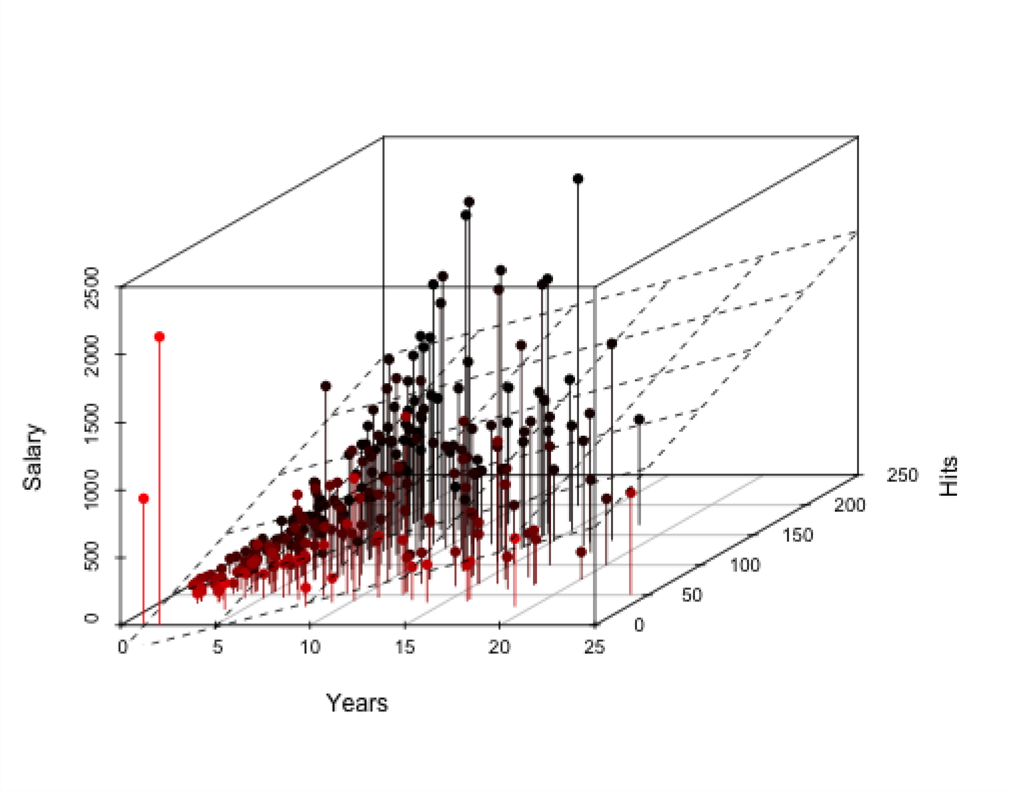
Our old friend

```
m1 <- lm(Salary ~ Years + Hits, data = Hitters)
```

```
coef(summary(m1))
```

```
##                Estimate Std. Error   t value     Pr(>|
## (Intercept) -199.250976 67.4689750 -2.953224 3.432509e
## Years         36.950116  4.7187203  7.830537 1.236069e
## Hits           4.312438  0.5012647  8.603116 7.461315e
```

We get: predictions



We get: predictions, cont.

$$MSE_{train} = \frac{1}{n} RSS$$

```r
mean(m1$res^2)
```

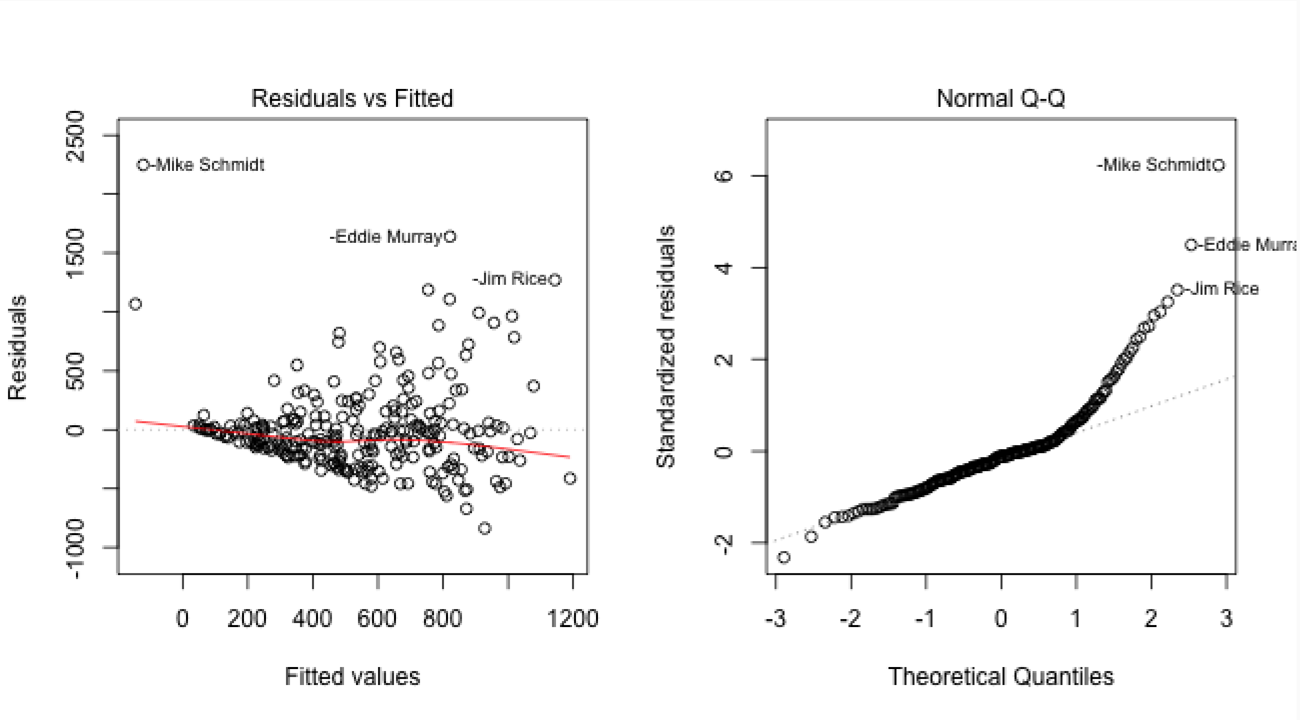
```
## [1] 132477.9
```

(Or better, use $CV_{(k)}$)

We get: a generative/probability model

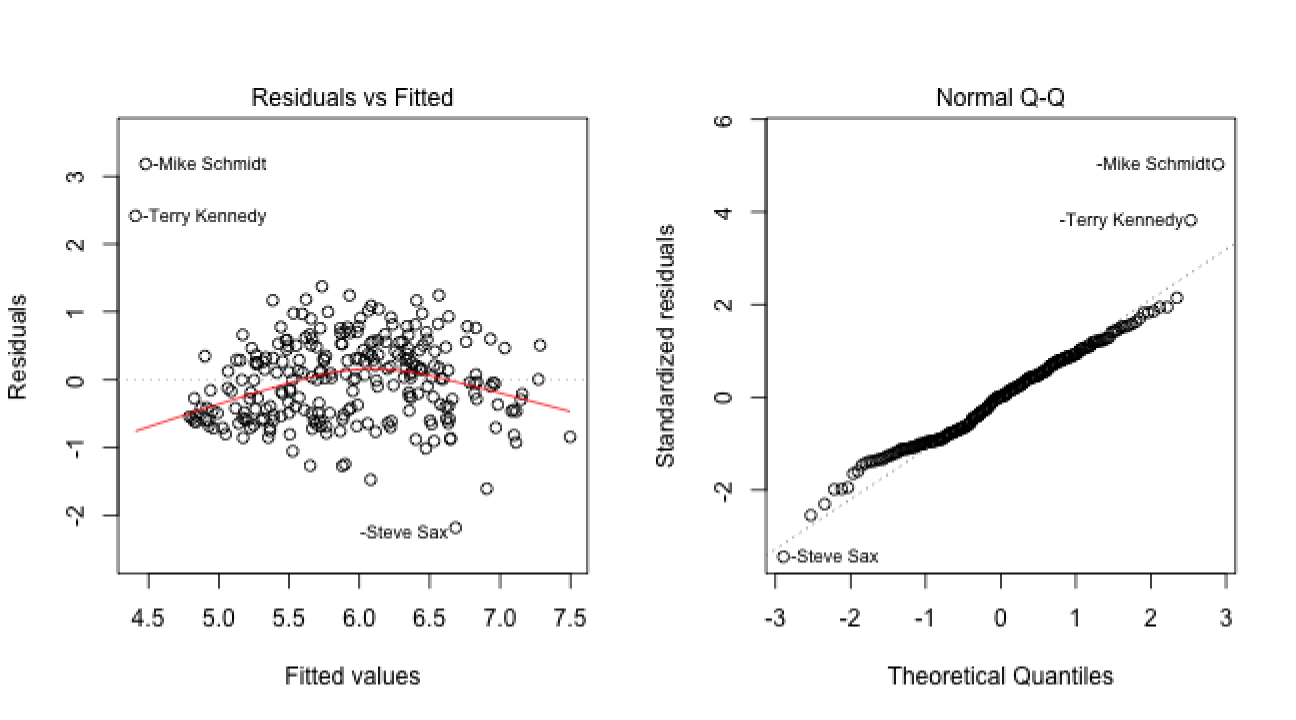
OLS regression:

$$P(Y = y \,|\, X = x) = N(\beta_0 + \beta_1 x, \sigma^2)$$



Quick fix?

```
Hitters$logSalary <- log(Hitters$Salary)
m2 <- lm(logSalary ~ Years + Hits, data= Hitters)
```



We get: description, kinda

```
summary(m2)$coef
```

```
##                Estimate  Std. Error   t value       Pr(
## (Intercept) 4.275128697 0.118395330 36.108930 2.717617
## Years       0.098162730 0.008280464 11.854737  3.32469
## Hits        0.008665097 0.000879625  9.850899  1.16384
```

- An *increase* in `Years` is associated with an *increase* in Salary, on average.
- An *increase* in `Hits` is associated with an *increase* in Salary, on average.

As the model becomes more complex, description becomes more difficult. Let's try something completely different.

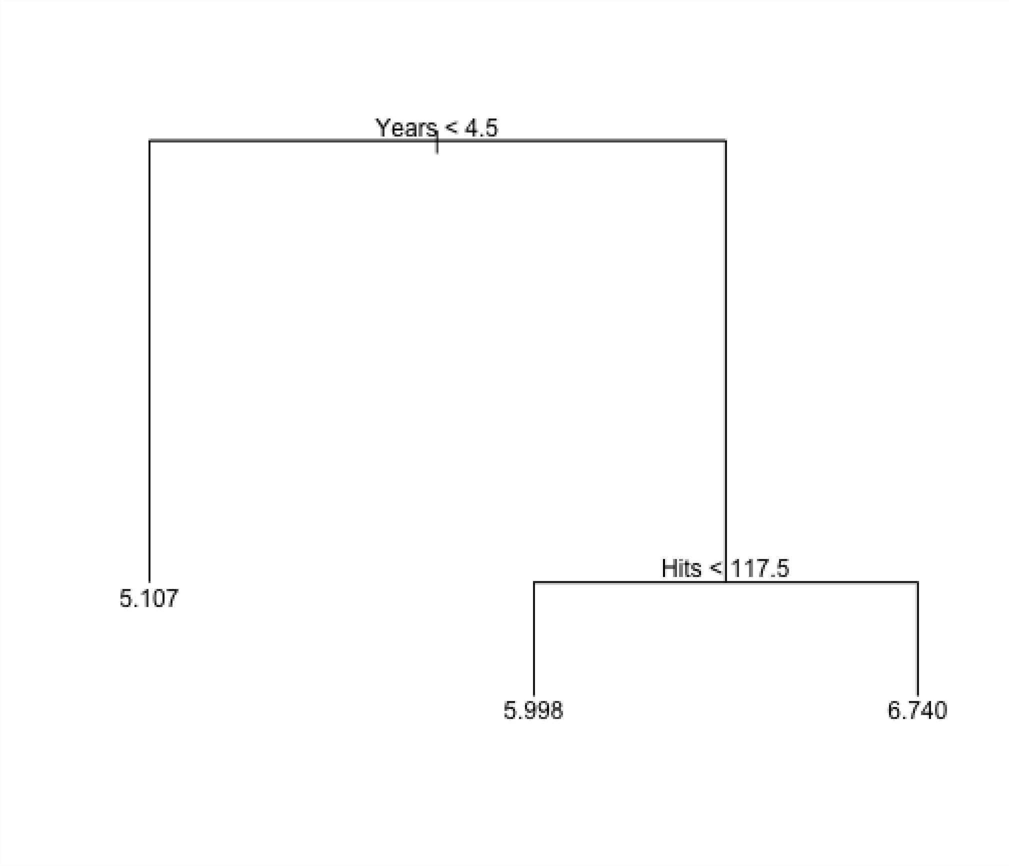
Regression Tree

A method to predict a continuous response, Y, using a series of p predictors, X, by recursive binary splitting to minimize RSS.

Regression Tree

A method to predict a continuous response, Y, using a series of p predictors, X, by recursive binary splitting to minimize RSS.

MLB Tree

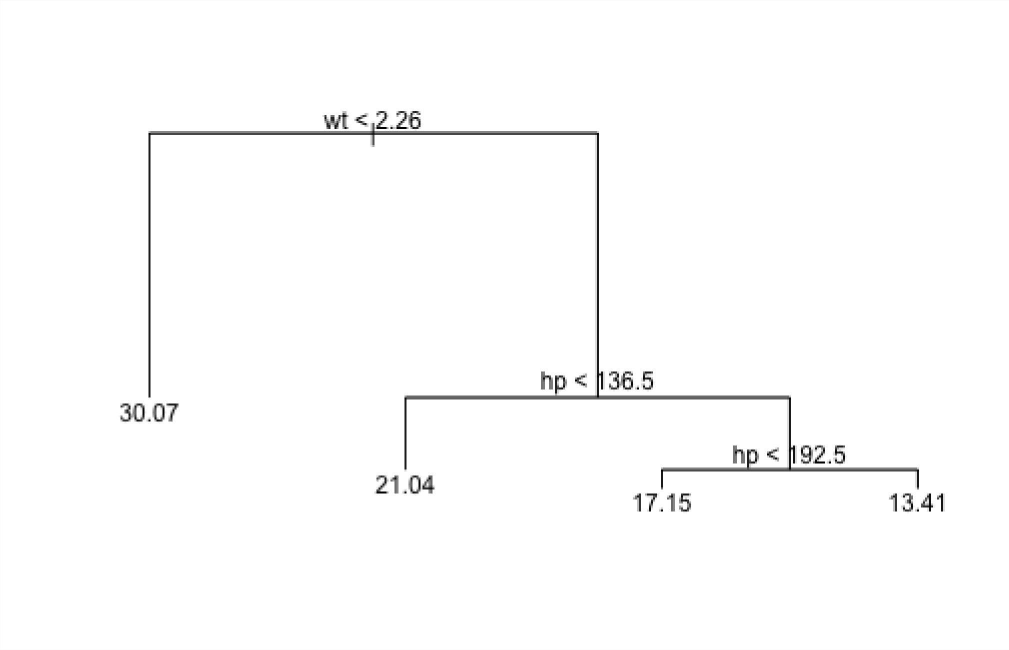


Boardwork

Interpretation

- `Years` is the most important factor in contributing to salary, with less-experienced players earning less.
- Given a player is less-experienced, `Hits` has little impact on `Salary`.
- Given a player is more experienced, those with more `Hits` have a higher `Salary`.

Practice #1: Draw the predictor space corresponding to the following tree (it's `mtcars`...sorry).



What would you expect the signs of the corresponding regression slopes to be?

```
m2 <- lm(mpg ~ hp + wt, data = mtcars)
summary(m2)$coef
```

```
##                Estimate Std. Error   t value     Pr(>|
## (Intercept) 37.22727012 1.59878754 23.284689 2.565459e
## hp          -0.03177295 0.00902971 -3.518712 1.451229e
## wt          -3.87783074 0.63273349 -6.128695 1.119647e
```

Practice #2 + boardwork

The Algorithm

1. Use RBS to grow a large tree on full data, stopping when every leaf has a small number of obs.
2. Apply cost-complexity pruning to obtain a best subtree for many values of α.

3. Use k-fold CV to choose α. For each fold:

    - Repeat (1) and (2) on training data.
    - Compute the test MSE on all subtrees (one test MSE per α). Average the test MSEs for each α and choose α that minimizes.

4. Use that α to select your best subtree in (2)