Wages by years of education and IQ:
An introduction to Classification and regression trees

Bayesian Learning

Professional Master's in Economics

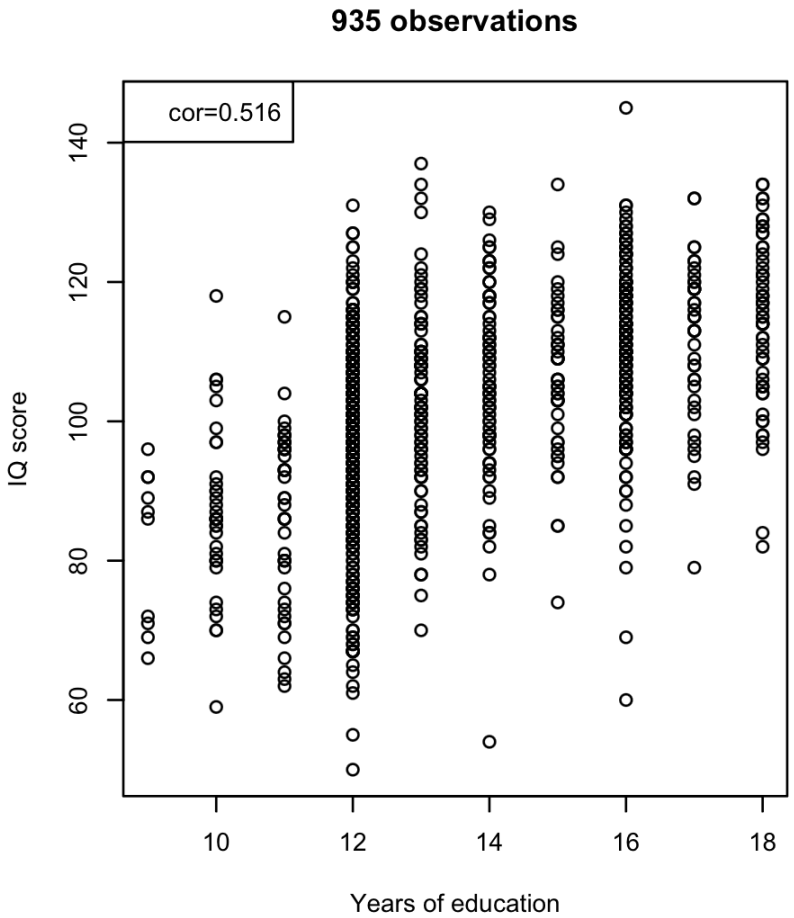
Professor Hedibert F. Lopes

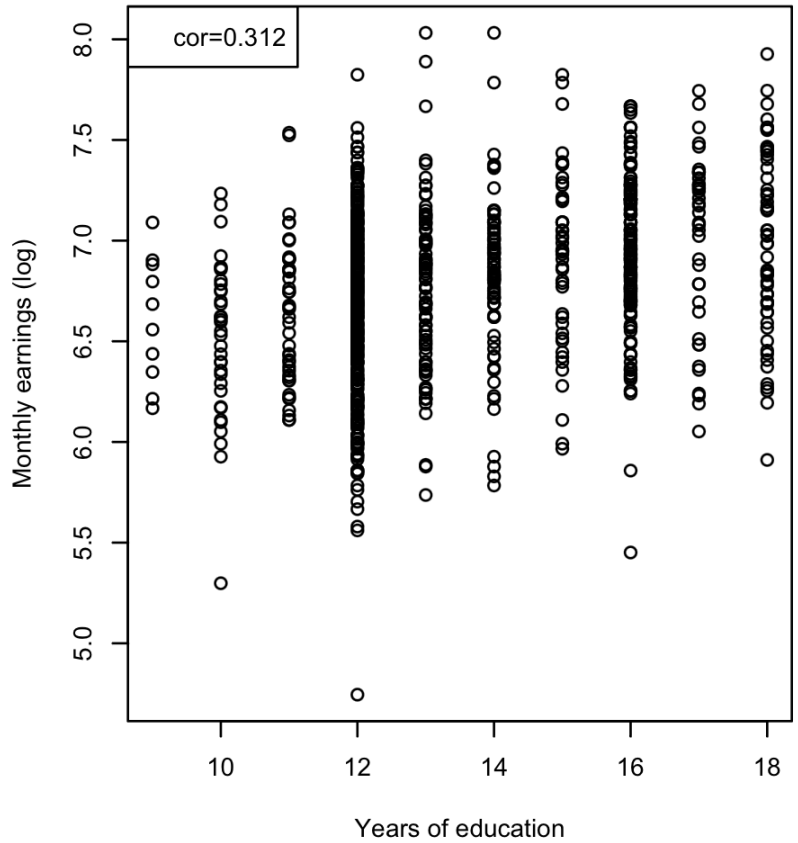
June 2026

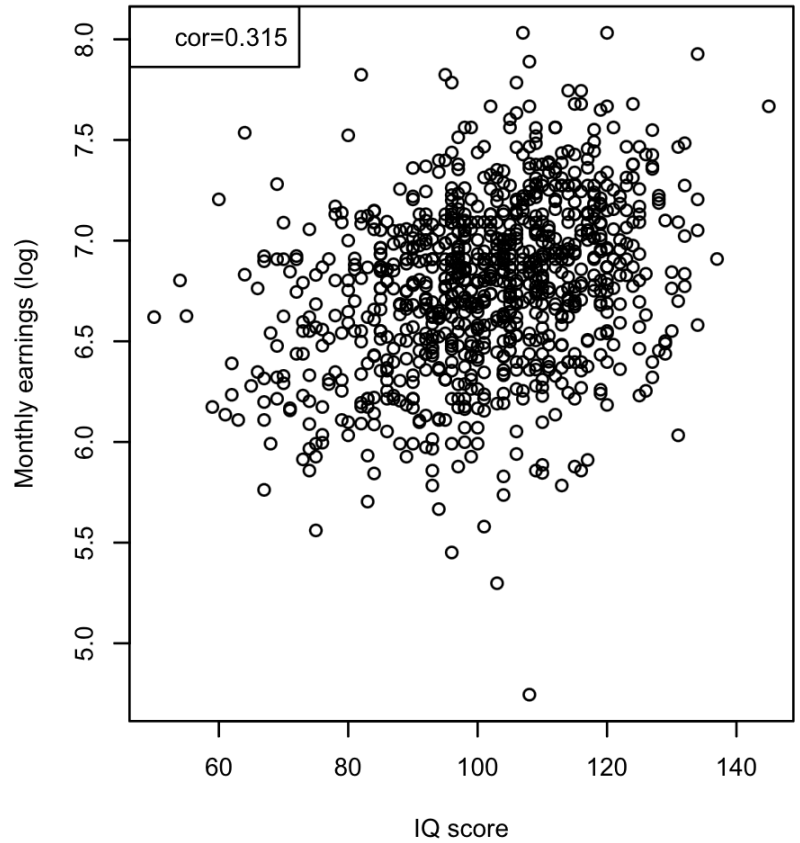
The dataset contains monthly earnings, education, several demographic variables, and IQ scores for 935 men in 1980 and studied in Blackburn and Newmark (1992) Unobserved ability, efficiency wages and interindustry wage, Quarterly Journal of Economics, 107, 1421-36, and replicated pedagogically in the textbook Introductory Econometrics: A Modern Approach (2012 5th edition, South-Western, Cengage Learning) by renowned Professor Jeff Wooldridge. In what follows, we will illustrate how to fit a classification and regression tree (CART) model to learn log wage based on iq (IQ score) and educ (years of education).

```
rm(list=ls())
datafile = "https://hedibert.org/wp-content/uploads/2014/02/wage2-wooldridge.txt"
data     = read.table(datafile)
n        = nrow(data)
```

Variable	Mean	StDev	Min	Q1	Median	Q3	Max
wage	958	404	115	669	905	1160	3078
Log wage	6.78	0.42	4.75	6.51	6.81	7.06	8.03
educ	13.5	2.2	9	12	12	16	18
iq	101.3	15.1	50	92	102	112	145


935 observations
cor=0.516
IQ score
Years of education
60
80
100
120
140
10
12
14
16
18


cor=0.312
Monthly earnings (log)
Years of education
5.0
5.5
6.0
6.5
7.0
7.5
8.0
10
12
14
16
18


cor=0.315
Monthly earnings (log)
IQ score
5.0
5.5
6.0
6.5
7.0
7.5
8.0
60
80
100
120
140

Linear regression

```
> summary(lm(salary~educ+iq))

Call:
lm(formula = salary ~ educ + iq)

Residuals:
     Min       1Q   Median       3Q      Max
-2.01601 -0.24367  0.03359  0.27960  1.23783

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.6582876  0.0962408  58.793  < 2e-16 ***
educ        0.0391199  0.0068382   5.721 1.43e-08 ***
iq          0.0058631  0.0009979   5.875 5.87e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3933 on 932 degrees of freedom
Multiple R-squared:  0.1297,	Adjusted R-squared:  0.1278
F-statistic: 69.42 on 2 and 932 DF,  p-value: < 2.2e-16
```

Classification and regression tree (CART)

Let us start with a much smaller dataset (20 randomly selected observations) so we can visualize the steps of fitting a CART model.

```
set.seed(2718282)
n1   = 20
ind = sort(sample(1:n,size=n1,replace=FALSE))
salary1 = log(data[ind,1])
educ1    = data[ind,5]
iq1      = data[ind,3]
```

Note: Individuals 31 and 715 have 12 years of education and IQ equal to 103. Their salaries are 1000 and 1559, respectively.

ind	salary1	educ1	iq1
4	6.476972	12	96
17	7.215240	15	109
20	6.154858	12	101
31	6.907755	12	103
45	7.466228	18	125
151	7.309881	17	113
162	6.932448	15	85
244	7.016610	16	118
281	6.762730	16	111
305	7.562162	16	99
366	6.396930	11	96
392	6.934397	12	115
530	6.845880	18	98
620	5.966147	15	74
646	7.050989	16	117
663	6.908755	12	77
715	7.351800	12	103
778	6.842683	16	113
798	6.955593	16	122
870	6.331502	12	104

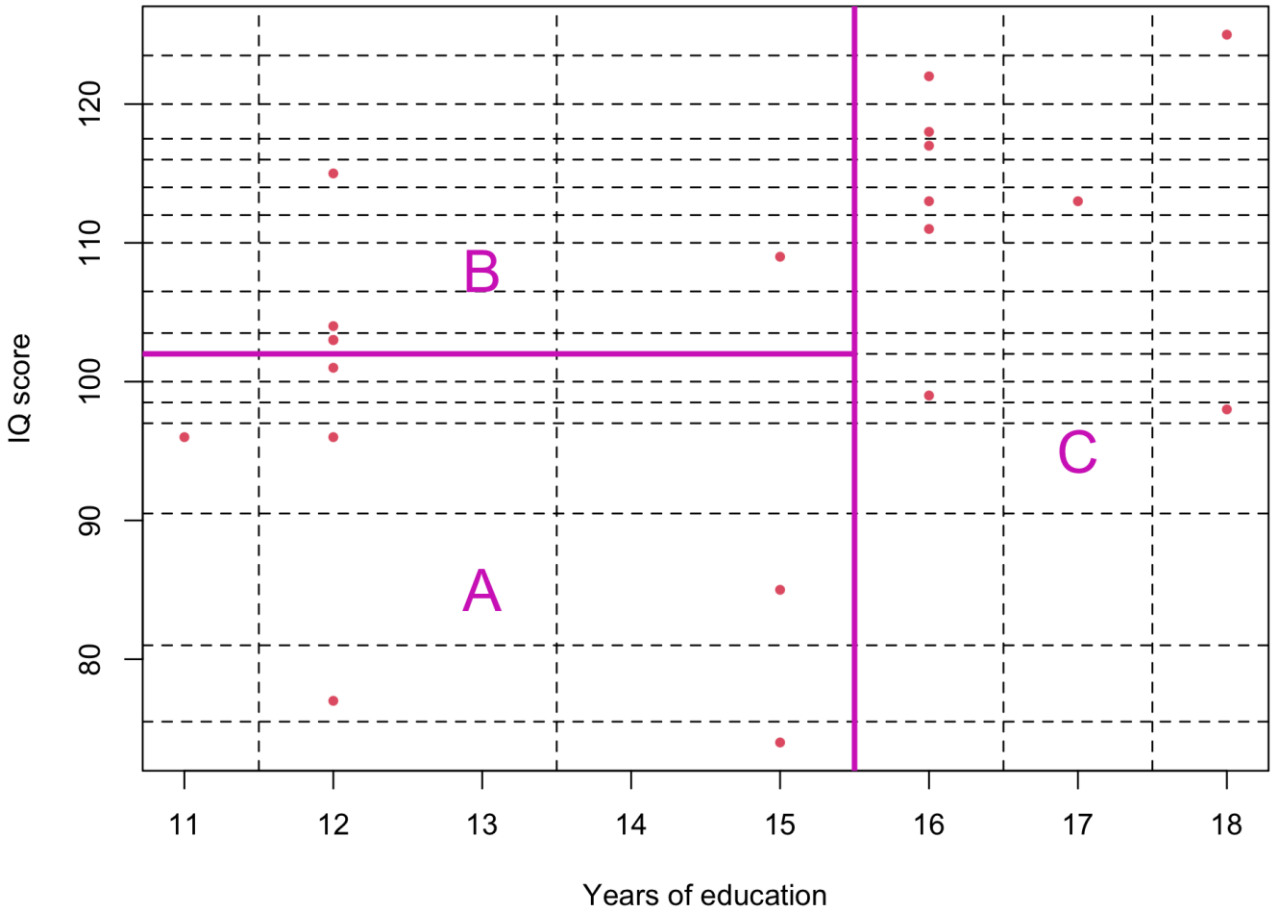
Tree structure and griddy search

The first decision is selecting the variable to start the tree (x1 or x2) and which value to pick as cut-off (splitting value).

The pair **(variable,cutoff)** that minimizes the deviance is (educ,15.5).

More precisely, we split R2 in two mutually exclusive regions: educ <=15.5 and educ > 15.5.

Then, for educ <= 15.5, the choice is **(IQ,102)**



Regressão linear é mais parcimoniosa.
Árvore é mais interpretável.

R package tree

tree {tree} R Documentation

Fit a Classification or Regression Tree

Description

A tree is grown by binary recursive partitioning using the response in the specified formula and choosing splits from the terms of the right-hand-side.

Usage

```
tree(formula, data, weights, subset,
     na.action = na.pass, control = tree.control(nobs, ...),
     method = "recursive.partition",
     split = c("deviance", "gini"),
     model = FALSE, x = FALSE, y = TRUE, wts = TRUE, ...)
```

Arguments

`formula`
: A formula expression. The left-hand-side (response) should be either a numerical vector when a regression tree will be fitted or a factor, when a classification tree is produced. The right-hand-side should be a series of numeric or factor variables separated by +; there should be no interaction terms. Both `.` and `-` are allowed: regression trees can have `offset` terms.

`data`
: A data frame in which to preferentially interpret `formula`, `weights` and `subset`.

`weights`
: Vector of non-negative observational weights; fractional weights are allowed.

`subset`
: An expression specifying the subset of cases to be used.

R package tree

```
library(tree)
fitted = tree(salary1 ~ educ1 + iq1)
summary(fitted)
```

```
Regression tree:
tree(formula = salary1 ~ educ1 + iq1)
Number of terminal nodes:  3
Residual mean deviance:  0.1204 = 2.047 / 17
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.61660 -0.24520 -0.03985  0.00000  0.29430  0.47190
```

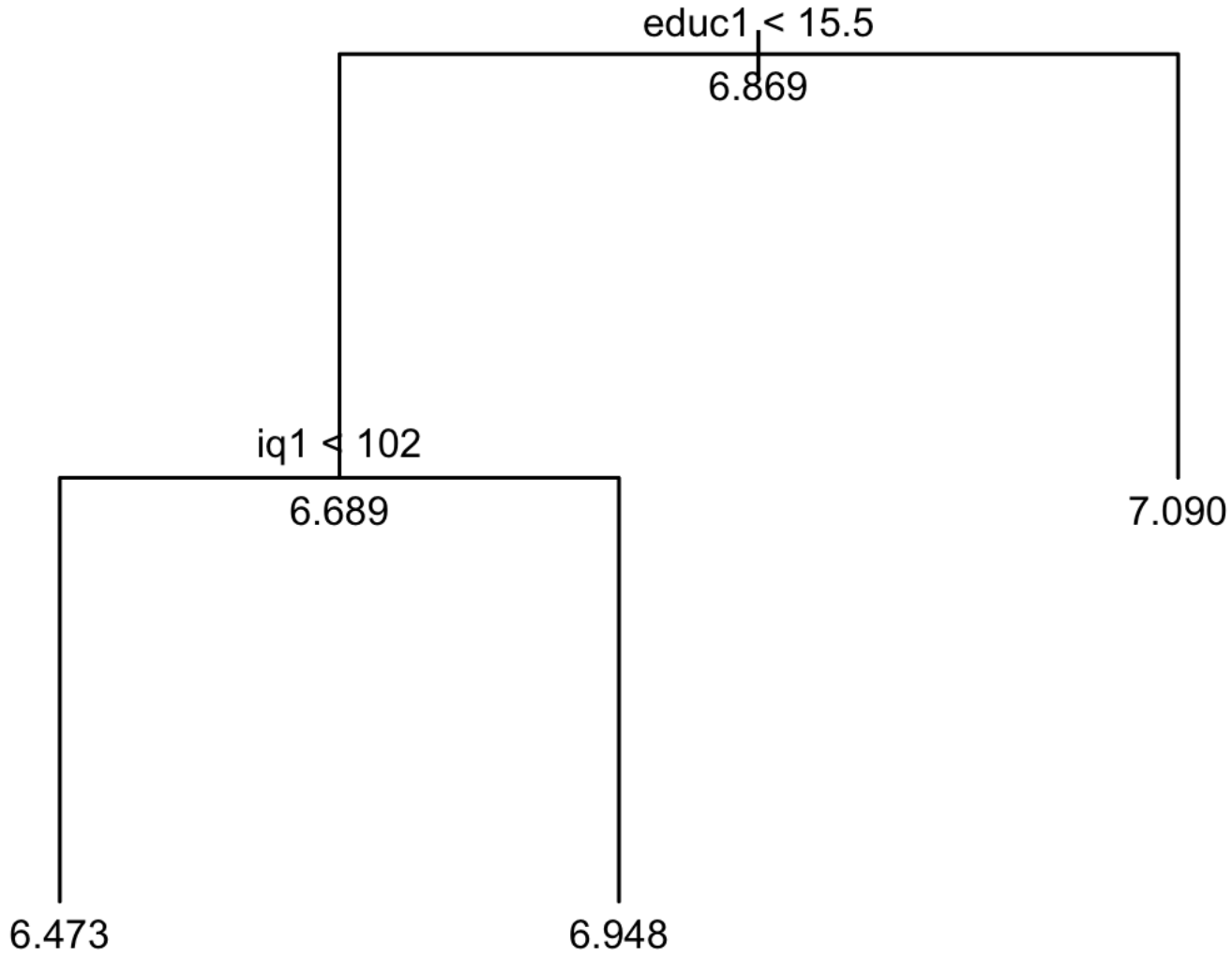
Fitted tree

```
fitted

plot(fitted,type="uniform")
text(fitted,pretty=0,all=TRUE,cex=0
```

```
node), split, n, deviance, yval
      * denotes terminal node

1) root 20 3.4610 6.869
  2) educ1 < 15.5 11 1.9980 6.689
    4) iq1 < 102 6 0.7649 6.473 *
    5) iq1 > 102 5 0.6163 6.948 *
  3) educ1 > 15.5 9 0.6657 7.090 *
```



lm.fit = lm(wage ~ iq + educ + exper + tenure + age + married + black + meduc)

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.882540   0.176769  27.621  < 2e-16 ***
iq           0.004294   0.001061   4.046 5.69e-05 ***
educ         0.050235   0.007899   6.359 3.31e-10 ***
exper        0.013166   0.004085   3.223  0.00132 **
tenure       0.008272   0.002676   3.091  0.00206 **
age          0.009615   0.005014   1.918  0.05546 .
married      0.175870   0.041228   4.266 2.22e-05 ***
black       -0.120267   0.043257  -2.780  0.00555 **
meduc        0.011328   0.004882   2.320  0.02057 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3698 on 848 degrees of freedom
  (78 observations deleted due to missingness)
Multiple R-squared:  0.2236,	Adjusted R-squared:  0.2162
F-statistic: 30.52 on 8 and 848 DF,  p-value: < 2.2e-16
```

Regressão é mais parcimoniosa

cart.fit = tree(wage ~ iq + educ + exper + tenure + age + married + black + meduc)

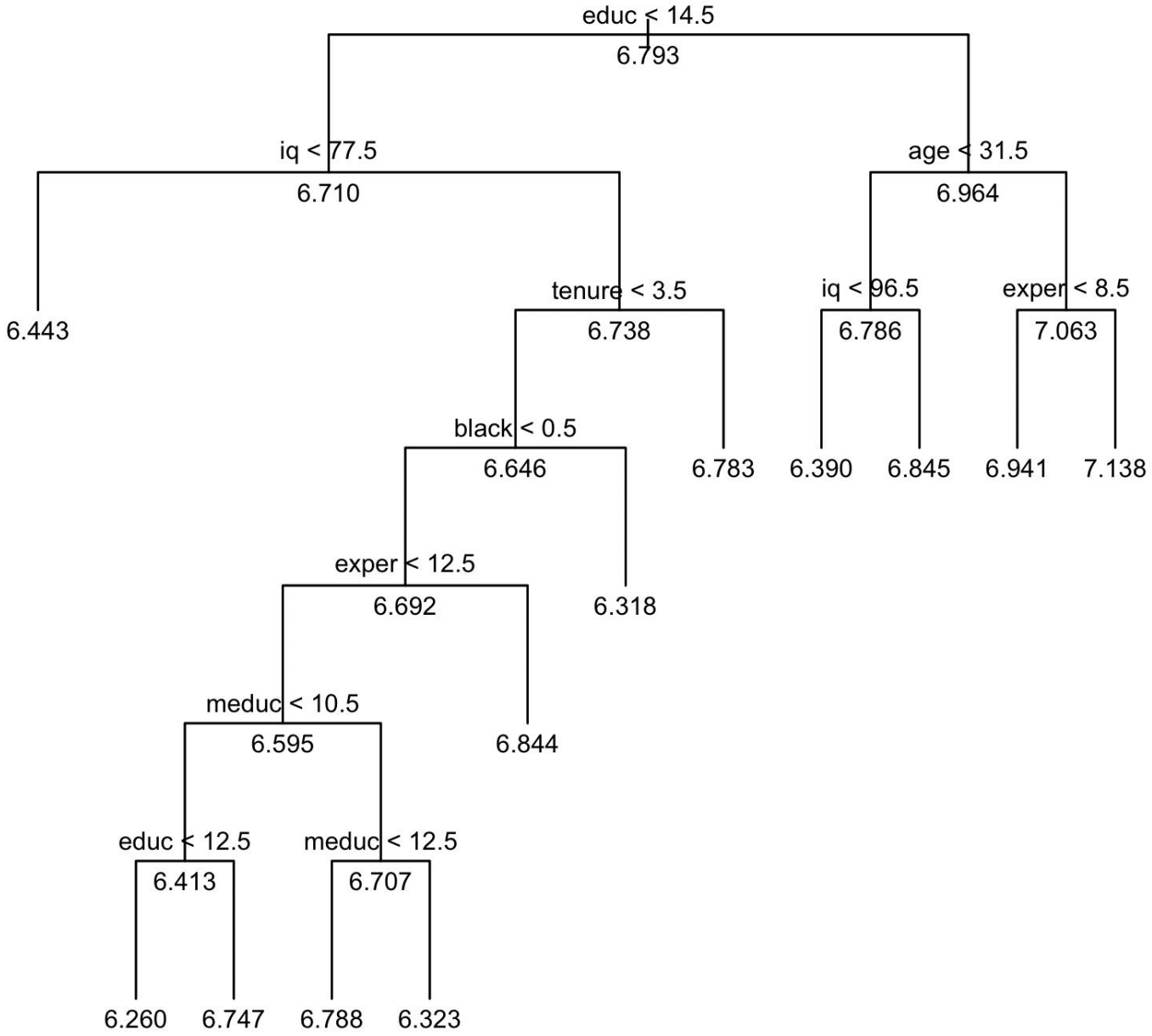
```
Regression tree:
tree(formula = wage ~ iq + educ + exper + tenure + age + married +
    black + meduc)
Variables actually used in tree construction:
[1] "educ"   "iq"     "tenure" "black"  "exper"  "meduc"  "age"
Number of terminal nodes:  12
Residual mean deviance:  0.1319 = 111.4 / 845
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.51600 -0.23200  0.02064  0.00000  0.24430  1.24900
```

CART é mais interpretável

```
node), split, n, deviance, yval
      * denotes terminal node

  1) root 857 149.400 6.793
    2) educ < 14.5 575  90.800 6.710
      4) iq < 77.5 55   6.274 6.443 *
      5) iq > 77.5 520  80.180 6.738
       10) tenure < 3.5 171  40.400 6.646
         20) black < 0.5 150  34.520 6.692
           40) exper < 12.5 92  24.720 6.595
             80) meduc < 10.5 35   9.213 6.413
              160) educ < 12.5 24   4.454 6.260 *
              161) educ > 12.5 11   2.973 6.747 *
             81) meduc > 10.5 57  13.640 6.707
              162) meduc < 12.5 47  10.570 6.788 *
              163) meduc > 12.5 10   1.288 6.323 *
           41) exper > 12.5 58   7.587 6.844 *
         21) black > 0.5 21   3.316 6.318 *
       11) tenure > 3.5 349  37.620 6.783 *
    3) educ > 14.5 282  46.310 6.964
      6) age < 31.5 101  15.470 6.786
       12) iq < 96.5 13   2.445 6.390 *
       13) iq > 96.5 88  10.680 6.845 *
      7) age > 31.5 181  25.870 7.063
       14) exper < 8.5 69  11.100 6.941 *
       15) exper > 8.5 112  13.110 7.138 *
```

Variables in the tree



- Educ
- IQ
- Age
- Tenure
- Exper
- Black
- Meduc