

CART

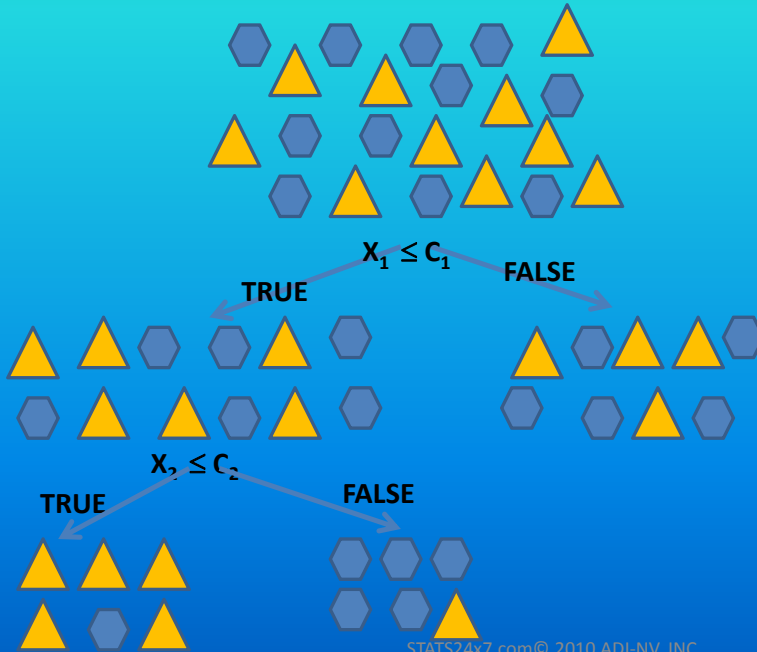
Classification and Regression Trees

Idea behind CART

When data clustered and messy, fitting a smooth function may not be the thing to do.

We can instead split the data into homogeneous clusters, and then just compute the mean of each cluster (i.e., fit a constant).

Recursive Partitioning is used to split data into homogeneous strata. Parent node is split into 2 children nodes – process repeated for each child node. An **impurity measure** is used to select splitting variable at each stage.



Possible Stopping Criteria

- Max #(nodes)
- Min # cases in a node reached

STATS24x7.com© 2010 ADI-NV, INC

3

Impurity Measures

Continuous Variable – sample variance

Qualitative Variable : several approaches

Binary Variable:

$$\text{Variance} = p(1 - p)$$

$$\text{log likelihood} = p \ln(p) + (1 - p) \ln(1 - p)$$

Several categories:

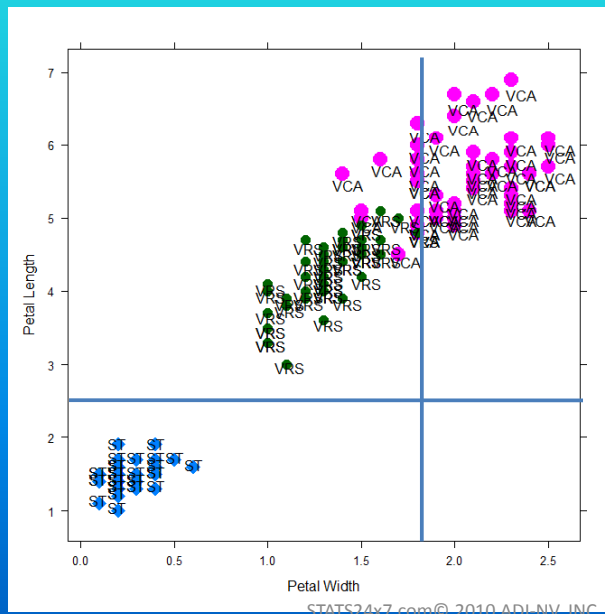
$$\text{max entropy} = - \sum p_k \ln(1 - p_k)$$

$$\text{Gini index} = \sum_{i \neq j} p_i p_j = 1 - \sum_j p_j^2$$

STATS24x7.com© 2010 ADI-NV, INC

4

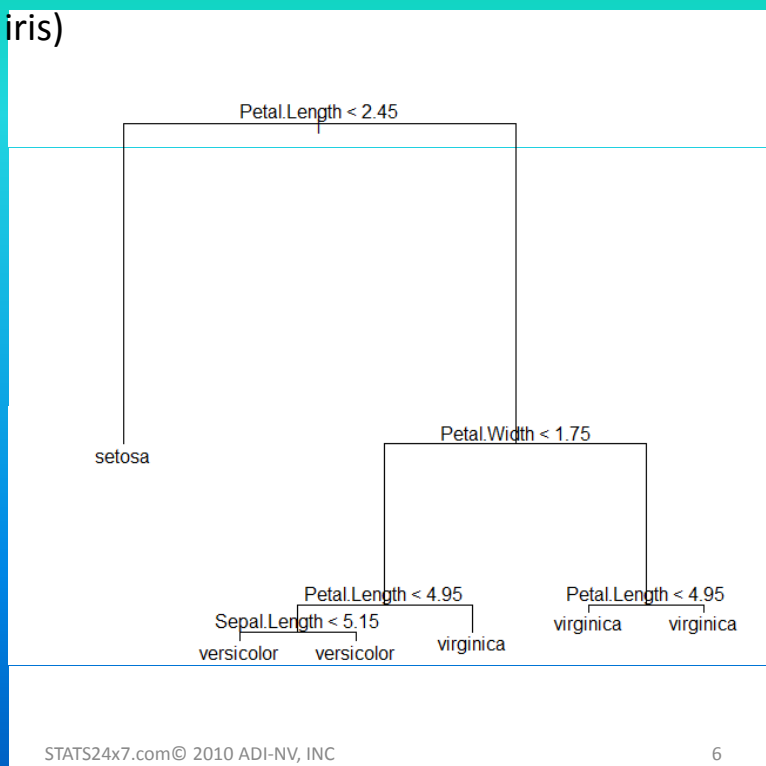
```
iris <- read.csv("M:/DataMining/Data/iris.csv", header=TRUE)
attach(iris)
xyplot(Petal.Length~Petal.Width, data=iris, groups=Species)
```



5

install and load R-package tree

```
ir.tr <- tree(Species ~., iris)
plot(ir.tr); text(ir.tr)
```



6

Boston Housing data set is a famous data set in data mining applications.

Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol. 5, 81-102, 1978.

BOSTON HOUSING DATA SET

The data set bostonhousing.csv has data on the following 14 variables:

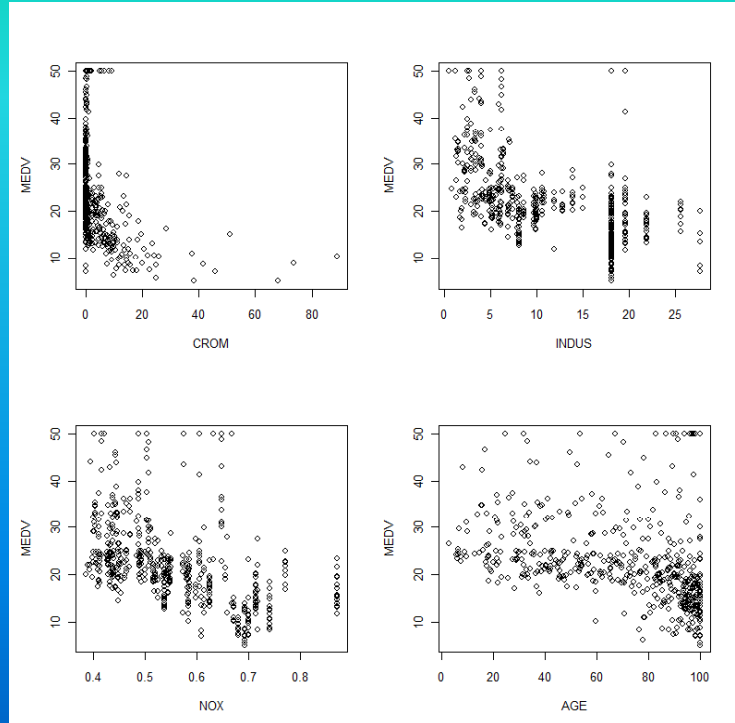
1. CRIM per capita crime rate by town
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS proportion of non-retail business acres per town
4. CHAS Charles River **dummy variable** (= 1 if tract bounds river; 0 otherwise)
5. NOX nitric oxides concentration (parts per 10 million)
6. RM average number of rooms per dwelling
7. AGE proportion of owner-occupied units built prior to 1940
8. DIS weighted distances to five Boston employment centres
9. RAD **index** of accessibility to radial highways
10. TAX full-value property-tax rate per \$10,000
11. PTRATIO pupil-teacher ratio by town
12. B $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
13. LSTAT % lower status of the population
14. MEDV Median value of owner-occupied homes in \$1000's

Develop a model for MEDV as a function of the other variables.

```

nf <- layout(matrix(c(1,2,3,4),2,2,byrow=TRUE), c(2,2), c(2,2), TRUE)
layout.show(nf)
plot(MEDV~CROM)
plot(MEDV~INDUS)
plot(MEDV~NOX)
plot(MEDV~AGE)

```



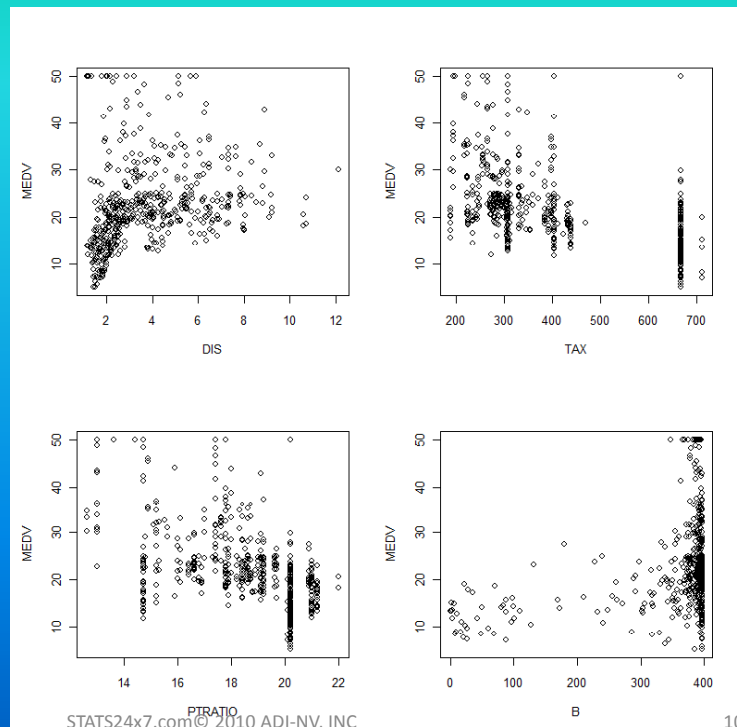
STATS24x7.com © 2010 ADI-NV, INC

9

```

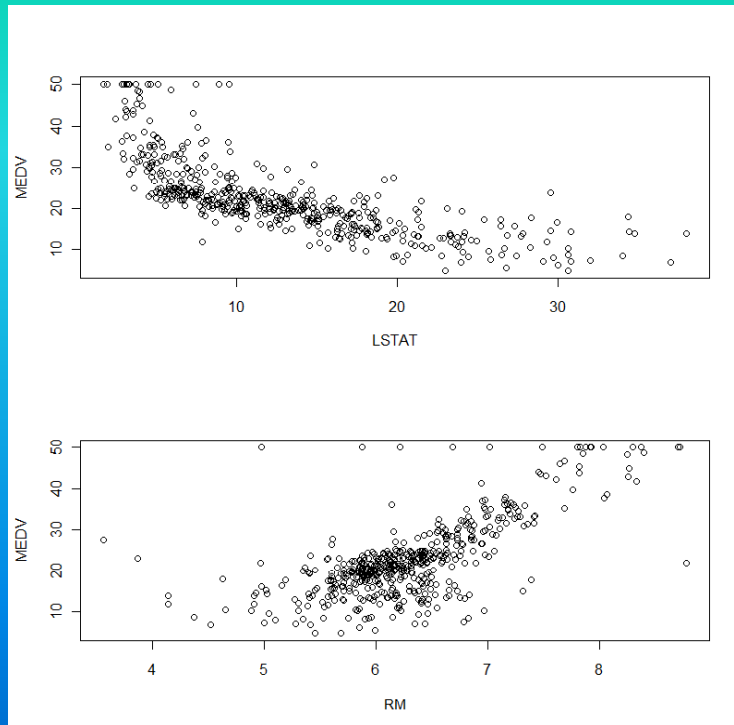
plot(MEDV~DIS)
plot(MEDV~TAX)
plot(MEDV~PTRATIO)
plot(MEDV~B)

```



STATS24x7.com © 2010 ADI-NV, INC

10



```
boston <-  
  read.csv("K:/DataMining/Data/bostonhousing.csv")  
attach(boston)  
  
library(fields)  
stats(boston)  
yields the descriptives of all variables in the dataset.
```

	CROM	ZN	INDUS	CHAS	NOX	RM	AGE
N	506	506	506	506	506	506	506
mean	3.613528	11.364	11.137	0.069	0.555	6.285	68.5749
Std.Dev.	8.601544	23.322	6.86	0.254	0.116	0.703	28.14886
min	0.0063	0	0.46	0	0.385	3.561	2.9
Q1	0.08205	0	5.19	0	0.449	5.886	45.025
median	0.25655	0	9.69	0	0.538	6.209	77.5
Q3	3.677075	12.5	18.1	0	0.624	6.624	94.075
max	88.9762	100	27.74	1	0.871	8.78	100

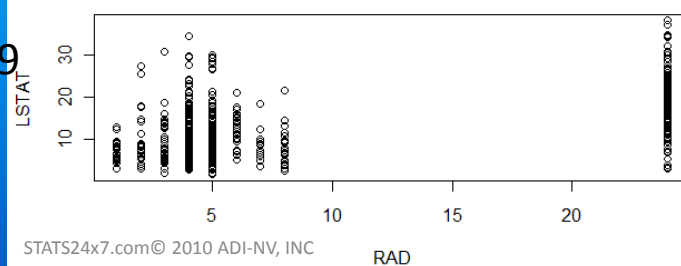
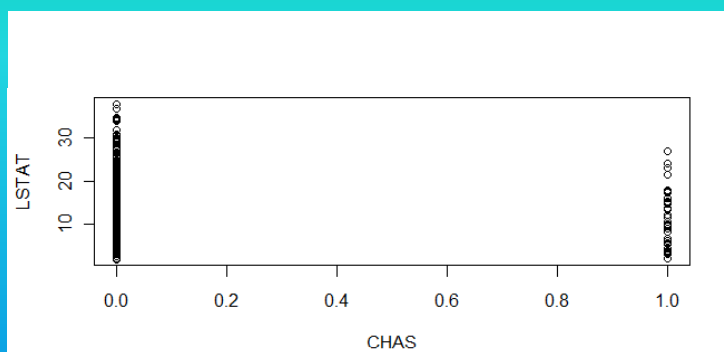
	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
N	506	506	506	506	506	506	506
mean	3.795	9.549	408.237	18.456	356.674	12.653	22.533
Std.Dev.	2.106	8.707	168.537	2.165	91.295	7.141	9.197
min	1.13	1	187	12.6	0.32	1.73	5
Q1	2.1	4	279	17.4	375.378	6.95	17.025
median	3.207	5	330	19.05	391.44	11.36	21.2
Q3	5.188	24	666	20.2	396.225	16.955	25
max	12.127	24	711	22	396.9	37.97	50

```
layout(1:2)
plot(LSTAT~CHAS)
plot(LSTAT~RAD)
```

Notice that

CHAS is an indicator (dummy) variable, and

RAD is a categorical (index) variable with 9 distinct values – so need 8 dummy variables.



```
dim(boston)
# [1] 506 14
```

```
bos <- matrix(NA, nrow=506, ncol=14)
bos <- as.matrix(boston)
```

```
# RAD has 9 levels, so need 8 dummy variables
DRAD1 <- matrix(0, nrow=506, ncol=1)
DRAD2 <- matrix(0, nrow=506, ncol=1)
DRAD3 <- matrix(0, nrow=506, ncol=1)
DRAD4 <- matrix(0, nrow=506, ncol=1)
DRAD5 <- matrix(0, nrow=506, ncol=1)
DRAD6 <- matrix(0, nrow=506, ncol=1)
DRAD7 <- matrix(0, nrow=506, ncol=1)
DRAD8 <- matrix(0, nrow=506, ncol=1)
```

```
for (i in 1:506)
{
  if (RAD[i]==1) DRAD1[i] <- 1
  if (RAD[i]==2) DRAD2[i] <- 1
  if (RAD[i]==3) DRAD3[i] <- 1
  if (RAD[i]==4) DRAD4[i] <- 1
  if (RAD[i]==5) DRAD5[i] <- 1
  if (RAD[i]==6) DRAD6[i] <- 1
  if (RAD[i]==7) DRAD7[i] <- 1
  if (RAD[i]==8) DRAD8[i] <- 1
}
```

```
lm1 <-
lm(MEDV~CROM+INDUS+CHAS+NOX+RM+AGE+DIS+TAX+PTRATIO+B
+LSTAT+DRAD1+DRAD2+DRAD3+DRAD4+DRAD5+DRAD6+DRAD7+DR
AD8)
summary(lm1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(t)
(Intercept)	43.748990	5.972154	7.325	9.97e-13 ***
CROM	-0.100591	0.033085	-3.040	0.002490 **
INDUS	-0.012843	0.063848	-0.201	0.840663
CHAS	2.598506	0.875356	2.969	0.003140 **
NOX	-17.819389	3.951770	-4.509	8.17e-06 ***
RM	3.914301	0.422262	9.270	< 2e-16 ***
AGE	-0.004622	0.013351	-0.346	0.729325
DIS	-1.244827	0.188120	-6.617	9.69e-11 ***
TAX	-0.006146	0.003892	-1.579	0.114921
PTRATIO	-1.165676	0.137567	-8.473	2.86e-16 ***
B	0.009524	0.002699	3.529	0.000457 ***
LSTAT	-0.521490	0.051327	-10.160	< 2e-16 ***
DRAD1	-6.744298	1.804636	-3.737	0.000208 ***
DRAD2	-5.467031	1.890413	-2.892	0.004000 **
DRAD3	-2.844654	1.749869	-1.626	0.104675
DRAD4	-4.505391	1.405118	-3.206	0.001433 **
DRAD5	-4.656084	1.437148	-3.240	0.001278 **
DRAD6	-6.298036	1.538710	-4.093	4.98e-05 ***
DRAD7	-2.661927	1.826728	-1.457	0.145704
DRAD8	-3.161463	1.701878	-1.858	0.063826 .

Multiple R-squared: 0.7421, Adjusted R-squared: 0.732

F-statistic: 73.6 on 19 and 486 DF, p-value: < 2e-16

STATS24x7.com © 2010 ADI-NV, INC

17

Coefficients:

	Estimate	Std. Error	t value	Pr(t)
(Intercept)	39.335339	5.234947	7.514	2.73e-13 ***
CROM	-0.088983	0.030914	-2.878	0.004171 **
CHAS	2.629026	0.861666	3.051	0.002403 **
NOX	-18.727568	3.307407	-5.662	2.54e-08 ***
RM	3.916544	0.408371	9.591	< 2e-16 ***
DIS	-1.204978	0.165035	-7.301	1.15e-12 ***
PTRATIO	-1.159077	0.129557	-8.946	< 2e-16 ***
B	0.009336	0.002656	3.515	0.000480 ***
LSTAT	-0.531909	0.048022	-11.076	< 2e-16 ***
DRAD1	-4.197494	1.170845	-3.585	0.000371 ***
DRAD2	-2.754293	1.094040	-2.518	0.012134 *
DRAD4	-2.242361	0.623589	-3.596	0.000356 ***
DRAD5	-2.310129	0.683590	-3.379	0.000784 ***
DRAD6	-4.184201	1.052826	-3.974	8.12e-05 ***

Residual standard error: 4.754 on 492 degrees of freedom

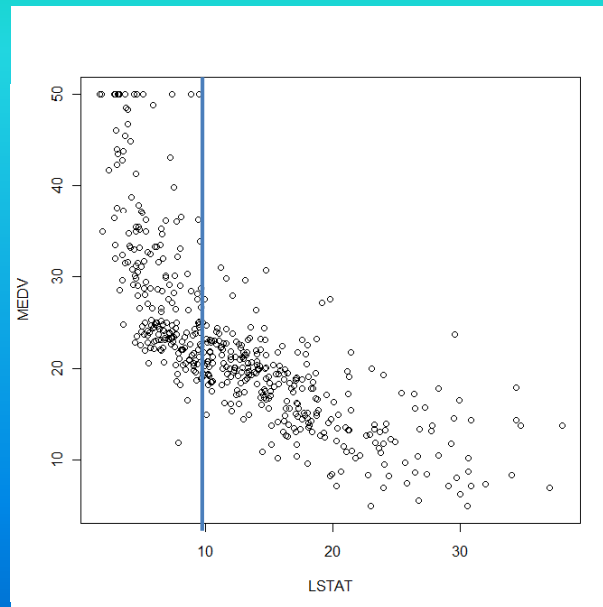
Multiple R-squared: 0.7397, Adjusted R-squared: 0.7328

F-statistic: 107.5 on 13 and 492 DF, p-value: < 2.2e-16

STATS24x7.com © 2010 ADI-NV, INC

18

```
plot(LSTAT, MEDV)
```



STATS24x7.com © 2010 ADI-NV, INC

19

Split data at LSTAT = 10.

```
a <- 10
```

```
b.left <- boston[,"LSTAT"] < a
```

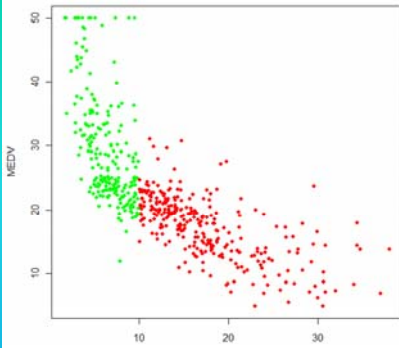
```
b.right <- boston[,"LSTAT"] >= a
```

Plot the two split on the same graph:

```
plot(boston[,"LSTAT"], boston[,"MEDV"], xlab="LSTAT",  
ylab="MEDV", col=c("red", "green")[b.left+1], pch=16, cex=.8)
```

STATS24x7.com © 2010 ADI-NV, INC

20

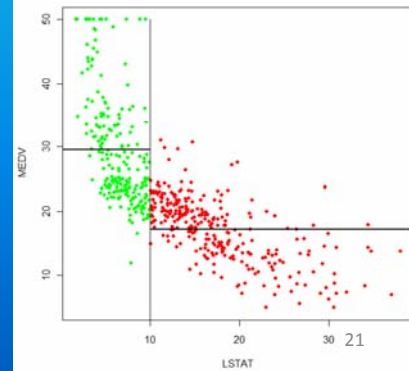


add lines at mean response values for the two halves

```
lines(c(a,a), c(0,50))
```

```
lines(c(0,a), rep(mean(boston[b.left,"MEDV"]), 2), lwd=2)
```

```
lines(c(a,40), rep(mean(boston[b.right,"MEDV"]), 2), lwd=2)
```



STATS24x7.com© 2010 ADI-NV, INC

How do we know LSTAT = 10 gives the best 1st split?

```
# sort LSTAT values for plotting
```

```
v.lstat <- sort(unique(boston[,"LSTAT"]))
```

```
ss <- rep(NA, length(v.lstat))
```

```
for(i in 1:(length(v.lstat)))
```

```
{
```

```
  spl <- v.lstat[i]
```

```
  b.left <- boston[,"LSTAT"] < spl;  b.right <- !b.left
```

```
  ss[i] <-
```

```
    sum((boston[b.left,"MEDV"] - mean(boston[b.left,"MEDV"]))^2) +
```

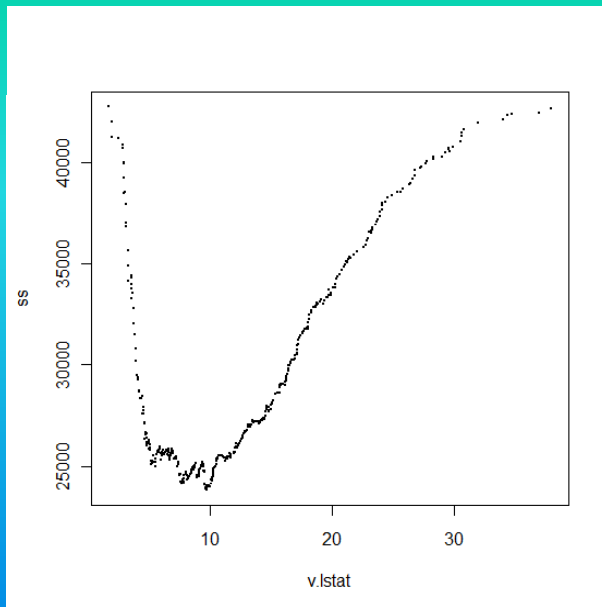
```
    sum((boston[b.right,"MEDV"] - mean(boston[b.right,"MEDV"]))^2)
```

```
}
```

```
plot(v.lstat, ss, pch=16, cex=.2)
```

STATS24x7.com© 2010 ADI-NV, INC

22



It is easy to see that $LSTAT = 10$ gives the minimum SUM OF SQUARES (SS) over *ALL POSSIBLE SPLITS*.

```
a <- 10
b.left <- boston["LSTAT"] < a
b.right <- boston["LSTAT"] = a
```

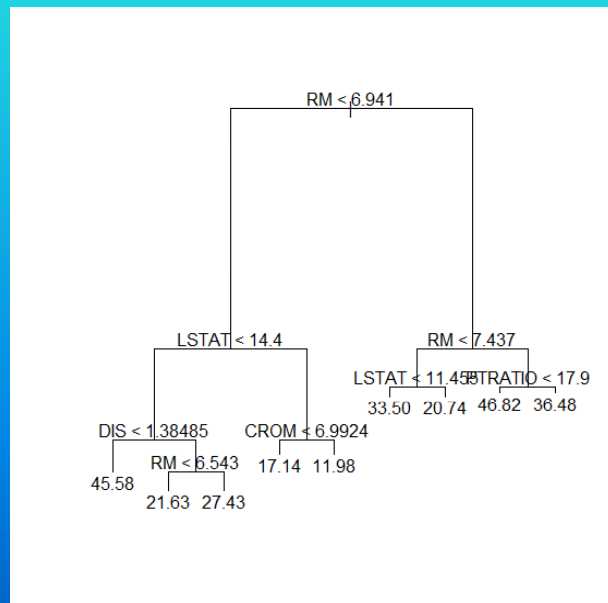
```
ss.10 <- sum((boston[b.left,"MEDV"] -
mean(boston[b.left,"MEDV"]))^2) +
+ sum((boston[b.right,"MEDV"] -
mean(boston[b.right,"MEDV"]))^2)
```

```
ss.10
[1] 24111.08
```

```
min(ss)
[1] 23820.1
```

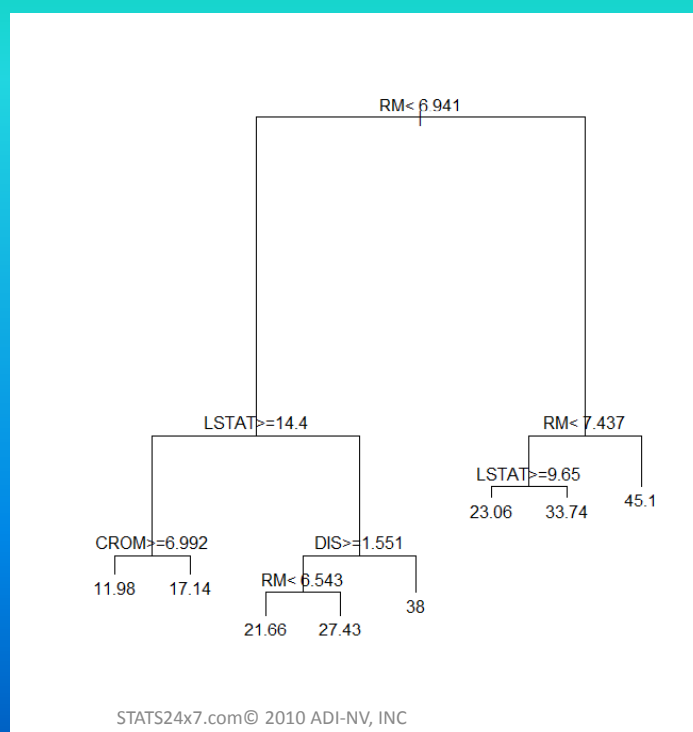
install and load R-package tree

```
boston.tr <- tree(MEDV ~., boston)  
plot(boston.tr); text(boston.tr)
```



#Rpart is another R-package for CART, install and load rpart

```
fit <- rpart(MEDV~CROM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+TAX+PTRATIO+B+LSTAT,  
data=boston)  
plot(fit)  
text(fit)
```



The party package of R uses a parametric model in recursive partitioning:

- 1) A parametric model is fitted to the full data.
- 2) Parameter instability is tested over a set of partitioning variables.
- 3) The variable with highest instability is used to split data.
- 4) Procedure is repeated for each child node.

Boston housing data will be used to illustrate the procedure.

```
# Convert CHAS and RAD to factors
boston$CHAS <- factor(boston$CHAS,
levels=0:1, labels = c("no", "yes"))
boston$RAD <- factor( boston$RAD, ordered =
TRUE)
```

```

# OLS is used for the model fitting, and
# instability of parameter is assessed using Bonferroni
# adjustment
control.par <- mob_control(alpha=.05, bonferroni=TRUE,
minspl=40, objfun=deviance, verbose=TRUE)

# model MEDV as a function of LSTAT and RM plus all
partitioning variables
mob1 <-
mob(MEDV~LSTAT+RM | CROM+CHAS+NOX+DIS+PTRATIO
+B+RAD, control = control.par, model = linearModel)

```

Fluctuation tests of splitting variables:

	CROM	CHAS	NOX	DIS	PTRATIO	B
statistic	7.399234e+01	4.908879e+01	6.321138e+01	1.013319e+02	7.60051e+01	3.339619e+01
p.value	1.006287e-13	3.258859e-08	2.555821e-11	6.655466e-20	3.55881e-14	7.302362e-05

RAD

statistic	7.685939e+01
p.value	2.288228e-14

Best splitting variable: DIS

Perform split? yes

Node properties:

DIS <= 1.6582; criterion = 1, statistic = 101.332

Fluctuation tests of splitting variables:

	CROM	CHAS	NOX	DIS	PTRATIO	B
statistic	8.187771e+01	4.110237e+01	9.560163e+01	3.497670e+01	7.997064e+01	7.067411e+01
p.value	1.685016e-15	1.708528e-06	1.323506e-18	3.388114e-05	4.530194e-15	5.515470e-13

RAD

statistic	8.343392e+01
p.value	7.511532e-16

Best splitting variable: NOX

Perform split? yes

Node properties:
NOX <= 0.655; criterion = 1, statistic = 95.602

Fluctuation tests of splitting variables:

	CROM	CHAS	NOX	DIS	PTRATIO	B
statistic	3.457337e+01	12.8541418	22.21193255	19.97970227	8.014125e+01	12.5715672
p.value	4.080716e-05	0.4898428	0.01306554	0.03494108	4.104767e-15	0.5283952

RAD
statistic 21.61332177
p.value 0.01705065

Best splitting variable: PTRATIO
Perform split? yes

Node properties:
PTRATIO <= 19.2; criterion = 1, statistic = 80.141

Fluctuation tests of splitting variables:

	CROM	CHAS	NOX	DIS	PTRATIO	B
statistic	19.9675392	10.8026857	16.1420955	22.06771089	4.134665e+01	11.8472459
p.value	0.0265156	0.7113743	0.1323357	0.01021371	8.179616e-07	0.5622984

RAD
17.23960984
p.value 0.08478723

Best splitting variable: PTRATIO
Perform split? yes

Node properties:

PTRATIO \leq 14.9; criterion = 1, statistic = 41.347

Fluctuation tests of splitting variables:

	CROM	CHAS	NOX	DIS	PTRATIO B	RAD	
statistic	9.9762064	12.4129558	8.3077281	11.2807251	8.980504	10.1135121	6.6781977
p.value	0.7880782	0.4519443	0.9460855	0.6086947	0.896195	0.7705713	0.9955196

Best splitting variable: CHAS

Perform split? no

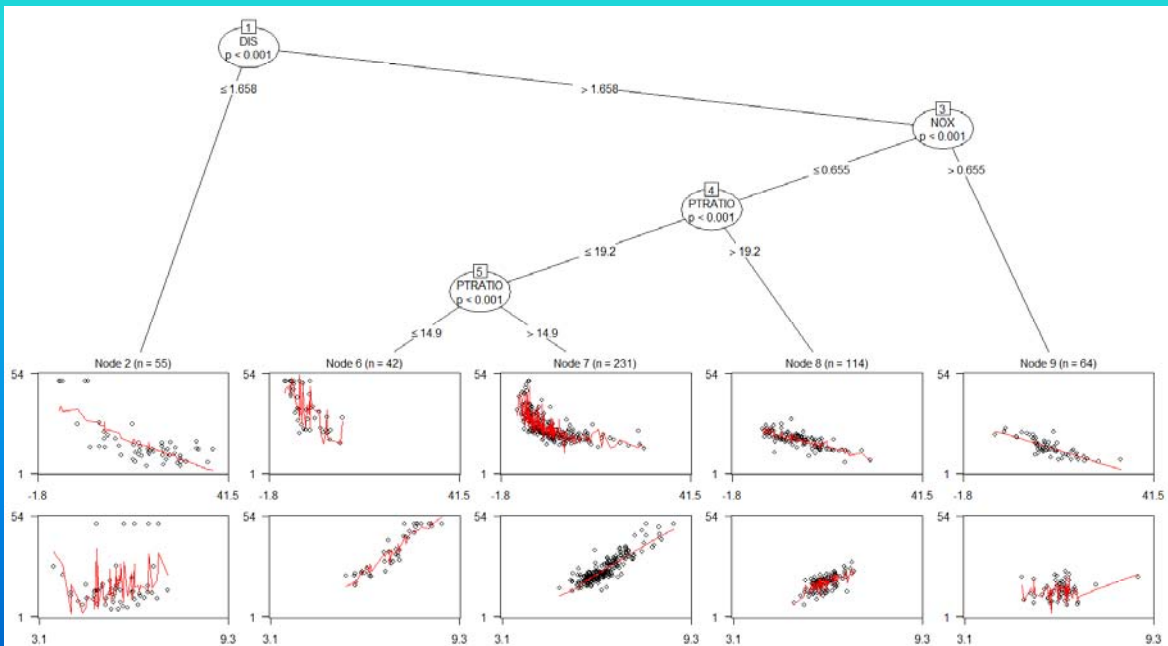
Fluctuation tests of splitting variables:

	CROM	CHAS	NOX	DIS	PTRATIO B	RAD	
statistic	5.0424258	1.575127	4.2758081	10.1220794	8.649540	3.922884	9.342154
p.value	0.9951466	1.000000	0.9993386	0.5162606	0.725515	0.999801	0.627116

Best splitting variable: DIS

Perform split? no

get visual output
plot(mob1)



```
coef(mob1)
  (Intercept)  LSTAT  RM
2  47.805558 -1.02831030 -1.4659481
6 -32.765950 -0.76153809 10.3638032
7 -33.907963 -0.07031379  9.3655556
8  3.372720 -0.46940157  3.5698254
9  28.379282 -0.72025904 -0.1238229
```

CART can be used for logistic regression as well.

The R-package mlbench has a benchmark data set – Pima Indians Diabetes data.

```
# read data
data("PimaIndiansDiabetes2", package = "mlbench")
pid <- PimaIndiansDiabetes2
library(fields)
stats(PimaIndiansDiabetes2)
```

```
# prints the 1-st 6 lines of a dataframe
head(pid)
```

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
1	6	148	72	35	NA	33.6	0.627	50	pos
2	1	85	66	29	NA	26.6	0.351	31	neg
3	8	183	64	NA	NA	23.3	0.672	32	pos
4	1	89	66	23	94	28.1	0.167	21	neg
5	0	137	40	35	168	43.1	2.288	33	pos
6	5	116	74	NA	NA	25.6	0.201	30	neg

```
library(fields)
stats(pid)
```

	pregnant	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
N	768	763	733	541	394	757	768	768	NA
mean	3.845052	121.687	72.405	29.153	155.548	32.457	0.472	33.241	NA
Std.Dev.	3.369578	30.536	12.382	10.477	118.776	6.925	0.331	11.76	NA
min	0	44	24	7	14	18.2	0.078	21	NA
Q1	1	99	64	22	76.25	27.5	0.244	24	NA
median	3	117	72	29	125	32.3	0.373	29	NA
Q3	6	141	80	36	190	36.6	0.626	41	NA
max	17	199	122	99	846	67.1	2.42	81	NA
missing	0	5	35	227	374	11	0	0	NA

Variables triceps and insulin have a large number of missing (NA) values, and hence should be dropped before predicting $P(\text{diabetes} = \text{POS})$.

If the NA values are removed from the data first, and then the variables triceps and insulin removed, we lose too many observations:

```
pid1 <- na.omit(pid) # remove NA
pid2 <- pid1[,-c(4,5)] # then remove triceps and insulin
```

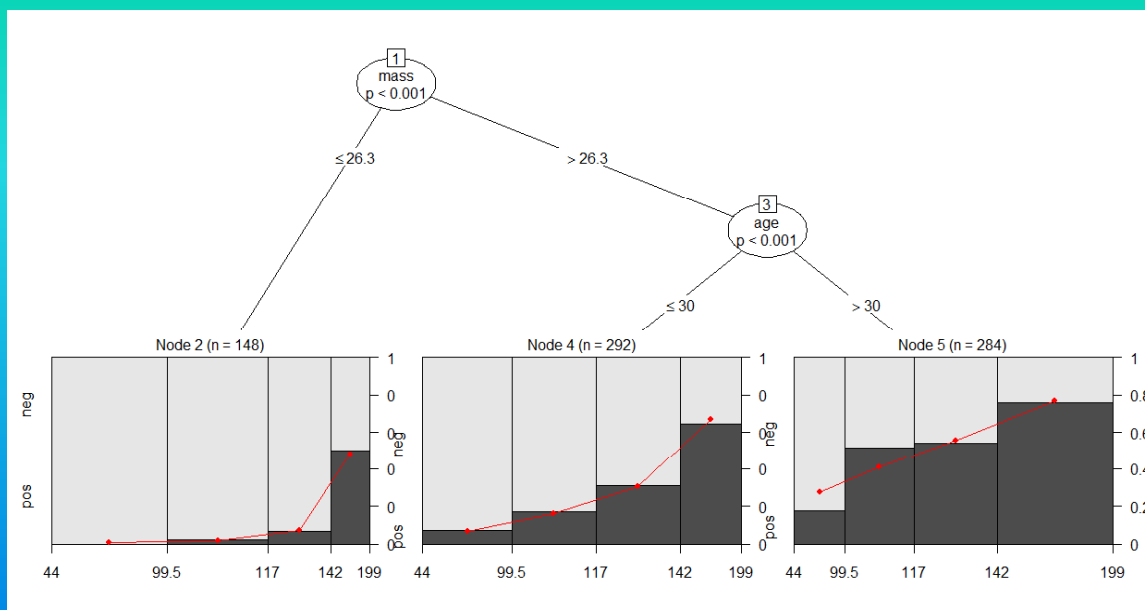
```
dim(pid2)
[1] 392 7
```

```
pid3 <- pid[,-c(4,5)] # first remove triceps and insulin
pid4 <- na.omit(pid3) # then remove NA
```

```
dim(pid3)
[1] 768 7
```

```
log1 <-  
mob(diabetes~glucose | pregnant+glucose+pressure+mass+pedigree  
+age, data = pid3, model=glinearModel,family=binomial())
```

```
plot(log1) # plots the regression tree (see next slide)
```



Node 2 shows: women with low BMI have lower risk of diabetes, risk goes up with glucose.

Node 4 shows: women with average and high bmi , <= 30 years have higher average risk which increases with glucose.

Node 5 shows: women with average and high bmi , > 30 years have higher average risk which increases slowly with glucose.

```
> coef(log1)
      (Intercept)  glucose
2      -10.999447  0.06456780
4       -6.573067  0.04504490
5       -3.318569  0.02748038
```

```
> exp(coef(log1)[,2])
      2      4      5
1.066698 1.046075 1.027861
```

Odds go up by **6.7%**, **4.6%**, and **2.78%** with glucose in the three partitions.