GG501

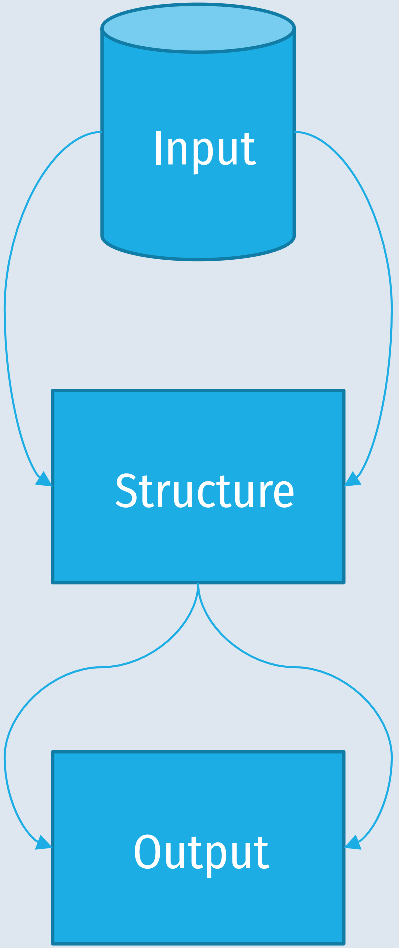
7. Parameterization and validation I

Models – parameterization & validation

- Any time we fit a model, we have to make choices
  - What data goes in
  - What settings or configutations need to be set to run the model
  - this is parameterization
  - these are model-dependent

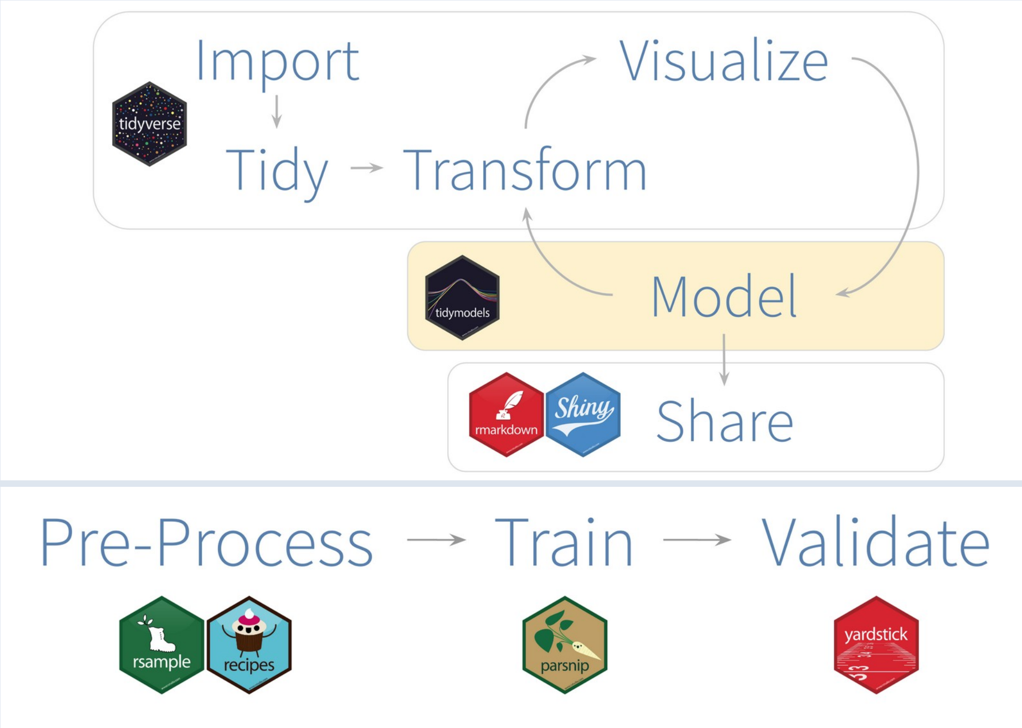
Many variety of models

- Each take some input data
- Attempt to generalize about the underling data-generating-process
- Can be used for a variety of purposes
  - description
  - explanation
  - prediction



Tidymodels

- provide a clean and unified interface for modelling in data as part of an overall tidy workflow
- a collection of packages that focus on common aspects of statistical modelling and support many different versions of models implemented in different packages

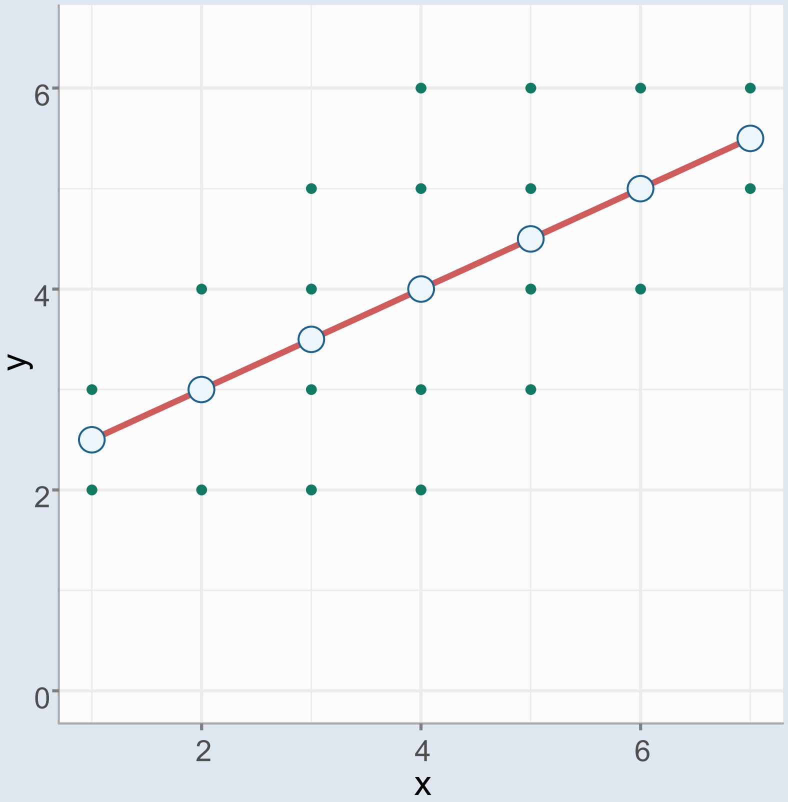


Linear model in R `lm`

- simple linear regression model available in R function `lm`
- linear model fits a relationship between covariates and the conditional mean of the response or dependent variable
- has strict assumptions regarding independence of error terms which have implicaitons for using with spatial / environmental data (and temporally correlated data)

Linear regression in R

- Recalling lm in r
- Each point represents a single observation
- The red line is the line of *best fit*
  - all predicted values from this model will fall on the line of best fit
- The line goes through each *conditional mean*
  - It goes through the mean at each value of x
    - e.g. When x = 1, mean of y = 2.5 (the conditional mean of y at x = 1 is 2.5)



Generalized linear models

- Flexible generalization of ordinary linear regression.
- Allows for outcomes that have other than a normal distribution
- R implementation considers all models and link functions implemented in the R function `glm`

Linear	Normally distributed outcome
Logistic	Binary outcome
Multi-nomial	Multi-class outcome
Poisson	Count outcome

glm in R

```r
## an example with offsets from Venables & Ripley (2002, p.189)
utils::data(anorexia, package = "MASS")

anorex.1 <- glm(Postwt ~ Prewt + Treat + offset(Prewt),
                family = gaussian, data = anorexia)
summary(anorex.1)
```

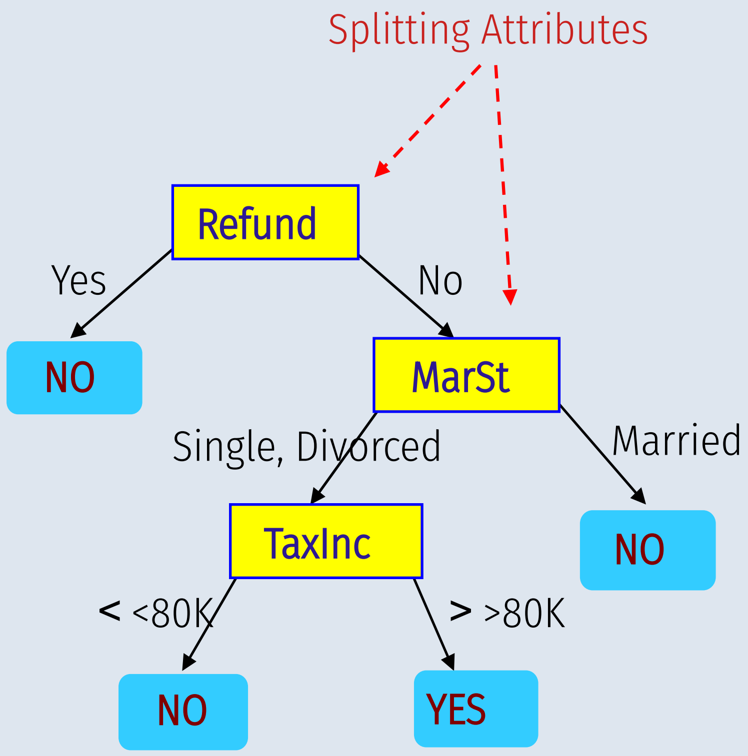
```r
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
data.frame(treatment, outcome, counts) # showing data
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson()
anova(glm.D93)
summary(glm.D93)
```

```
> summary(anorex.1)

Call:
glm(formula = Postwt ~ Prewt + Treat + offset(Prewt), family = gaussian,
    data = anorexia)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-14.1083   -4.2773   -0.5484    5.4838   15.2922

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.7711    13.3910   3.717 0.000410 ***
Prewt        -0.5655     0.1612  -3.509 0.000803 ***
TreatCont    -4.0971     1.8935  -2.164 0.033999 *
TreatFT       4.5631     2.1333   2.139 0.036035 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 48.69504)

    Null deviance: 4525.4  on 71  degrees of freedom
Residual deviance: 3311.3  on 68  degrees of freedom
AIC: 489.97

Number of Fisher Scoring iterations: 2
```

Example of a decision tree

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training data



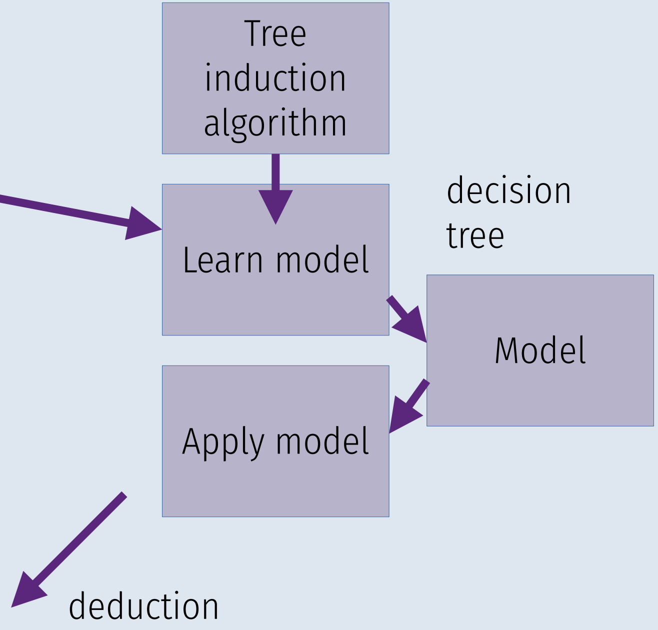
Decision tree

Decision tree classification task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	95K	?

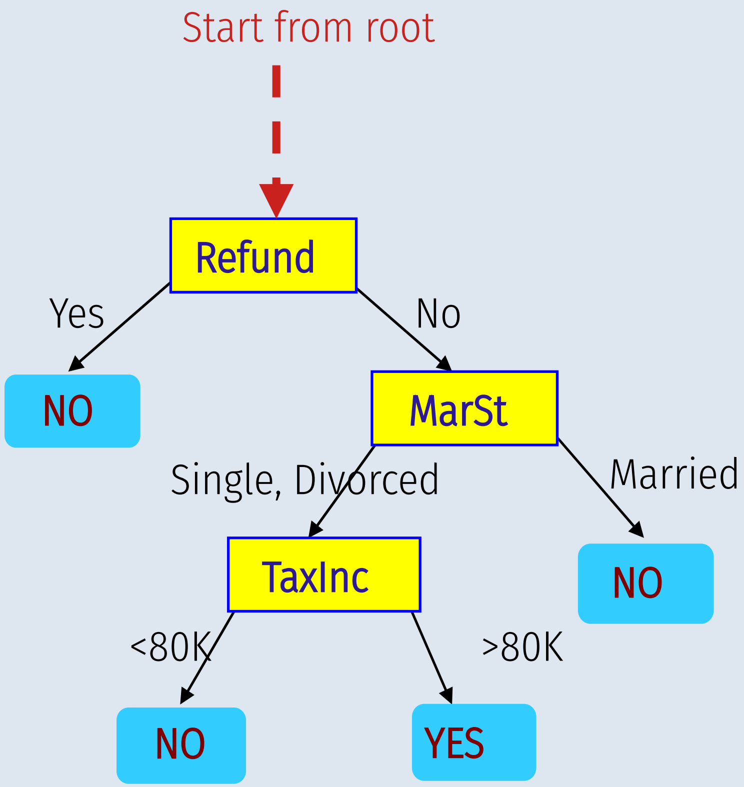
induction



Training set

Test set

Apply model to test data



Test data

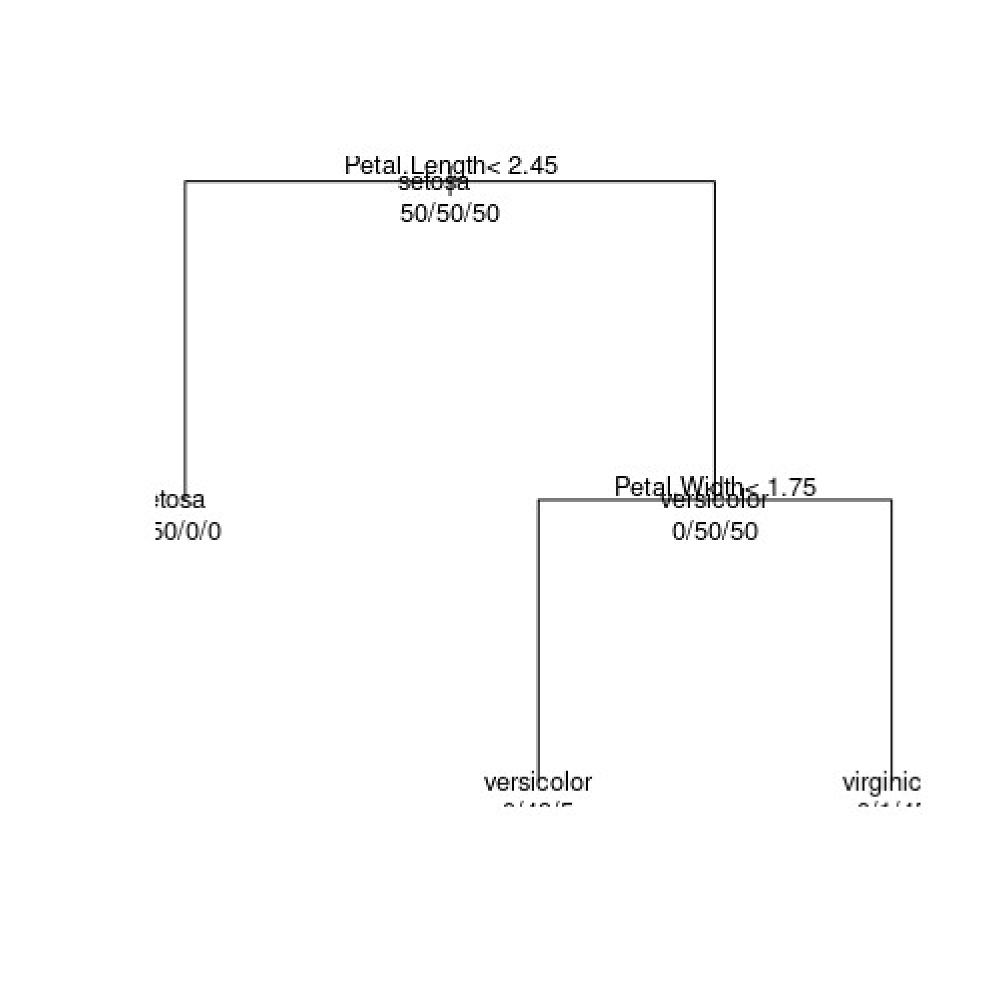
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Decision trees

- Used for classifying data by partitioning attribute space
- Tries to find decision boundaries for specified optimality criteria
- Leaf nodes contain class labels, representing classification decisions
- Keeps splitting nodes based on split criterion, such as
  - GINI index, information gain or entropy
- Pruning necessary to avoid overfitting

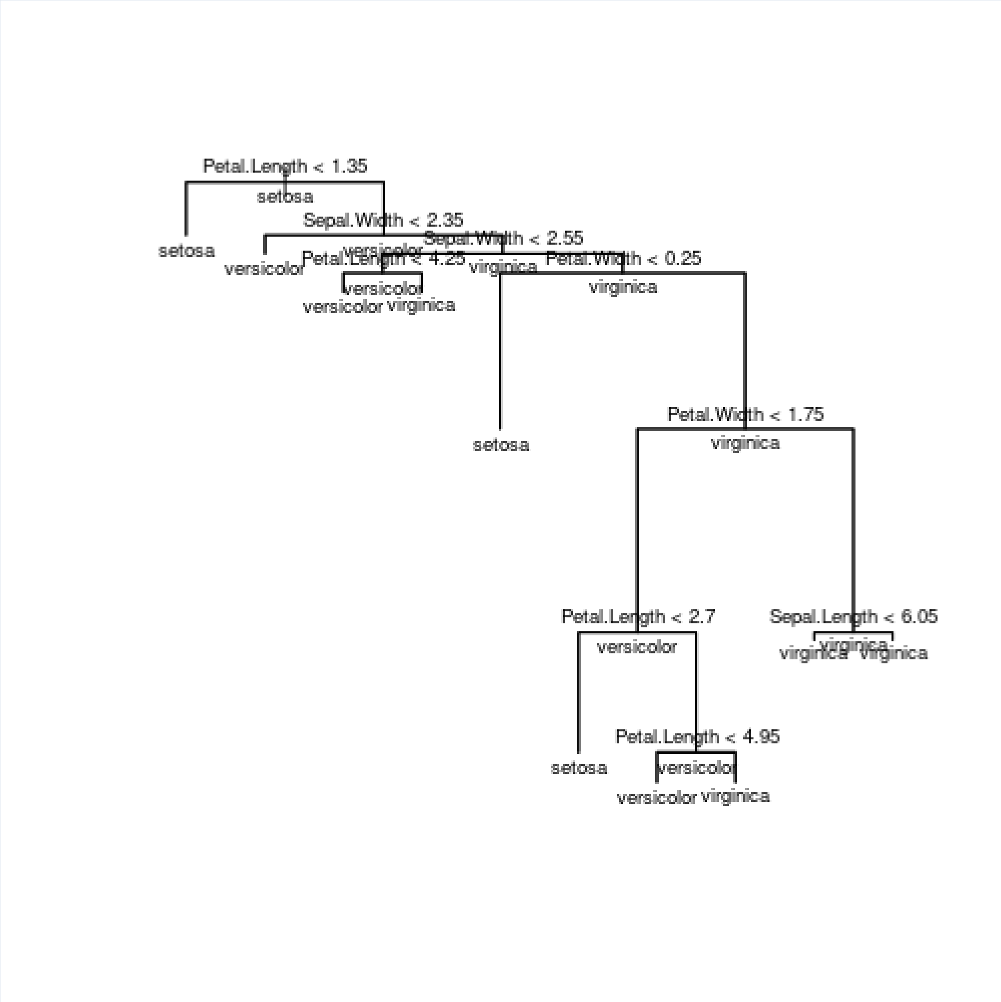
Decision trees in R

```
mydata<-data.frame(iris)
library(rpart)
model<-rpart(Species ~ Sepal.Length +
               Sepal.Width + Petal.Length +
               Petal.Width,
             data=mydata,
             method="class")
plot(model)
text(model,use.n=TRUE,all=TRUE,cex=0.8)
```



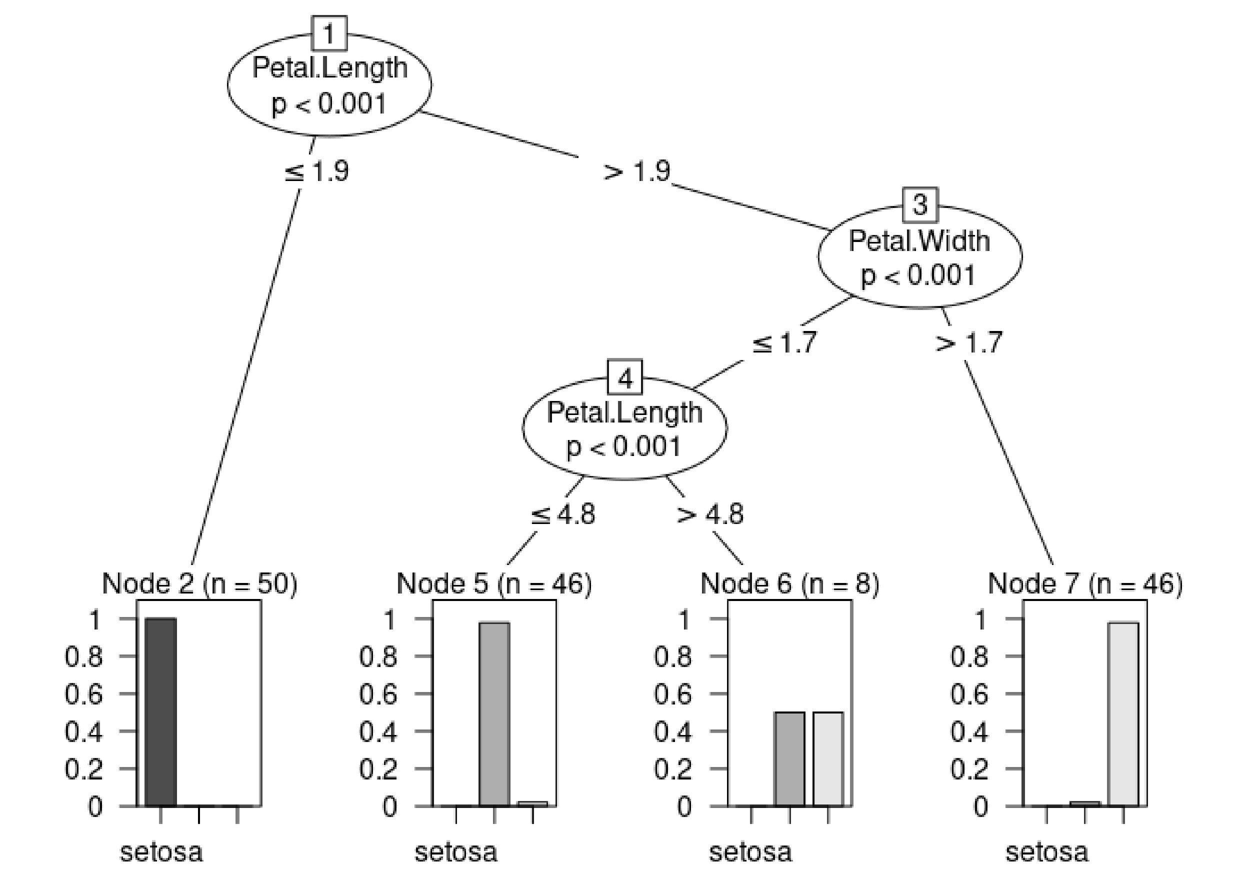
Decision trees in R

```
library(tree)
model1<-tree(Species ~ Sepal.Length +
               Sepal.Width + Petal.Length +
               Petal.Width,
             data=mydata,
             method="class",
             split="gini")
plot(model1)
text(model1,all=TRUE,cex=0.6)
```



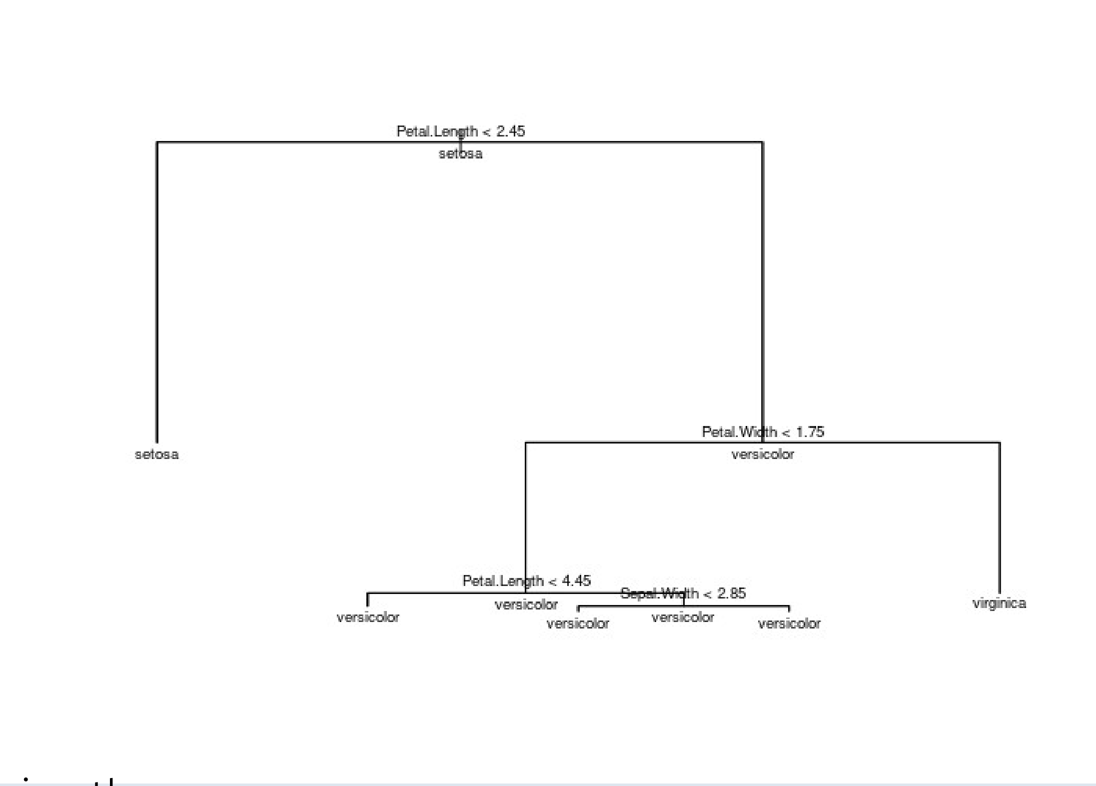
Decision trees in R

```
library(party)
model2←ctree(Species ~
                Sepal.Length +
                Sepal.Width +
                Petal.Length +
                Petal.Width,
              data=mydata)
plot(model2)
```



Controlling the number of nodes

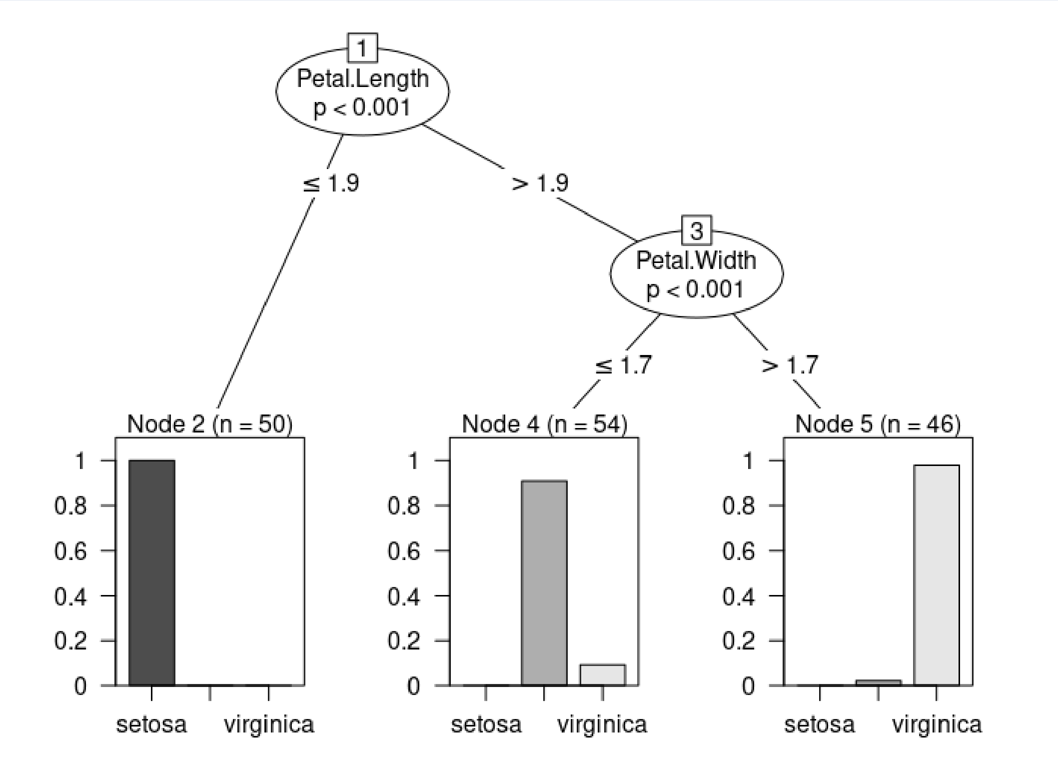
```
library(tree)
mydata←data.frame(iris)
model1←tree(Species ~ Sepal.Length +
               Sepal.Width + Petal.Length +
               Petal.Width,
             data=mydata,
             method="class",
             control = tree.control(nobs =
150, mincut = 10))
plot(model1)
text(model1,all=TRUE,cex=0.6)
predict(model1,iris)
```



Note how the number of nodes is reduced by increasing the minimum number of observations in a child node!

Controlling the number of nodes

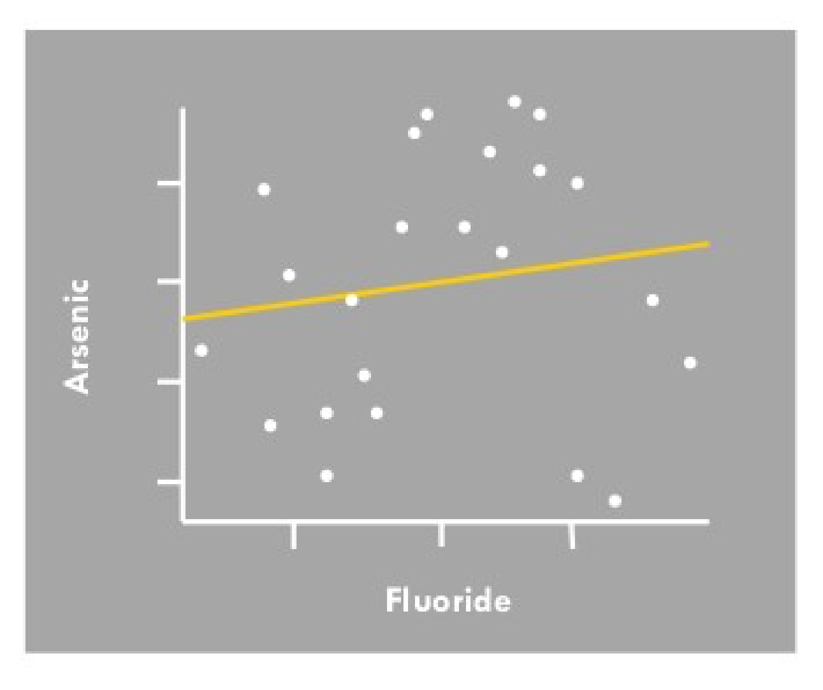
```
library(party)
model2<-ctree(Species ~
                Sepal.Length + Sepal.Width +
                Petal.Length + Petal.Width,
              data = mydata,
              controls =
ctree_control(maxdepth=2))
plot(model2)
```



Note that setting the maximum depth to 2
has reduced the number of nodes

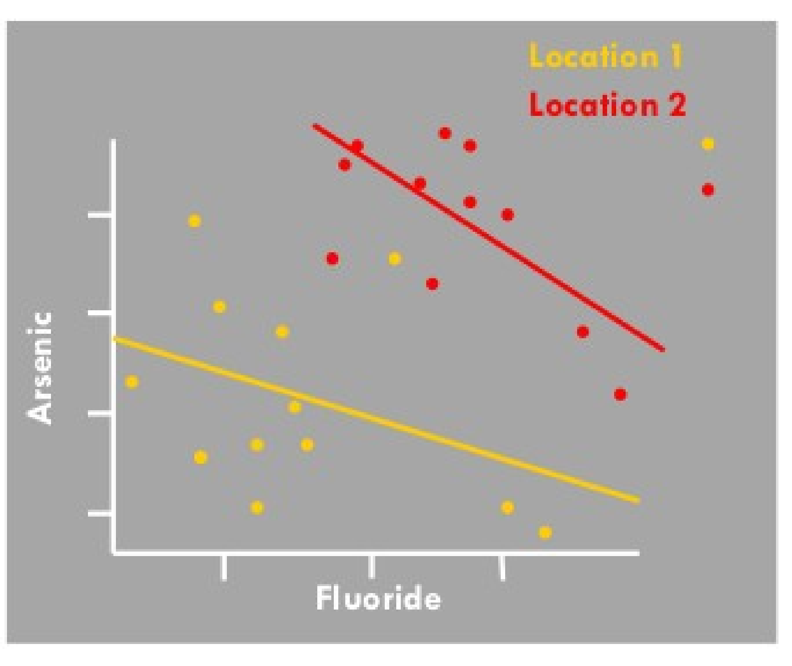
Spatial variation in model fit

- When fitting a ‘global’ model (i.e., a single model) for a process observed over different spatial locations, we must assume that the relationship(s) are constant over space
  - this is often an incorrect assumption
- Here we have a line of best fit through two variables of water quality parameters, fluoride and arsenic



Spatial variation in model fit

- Split up global data into regions and fit separate models for each region
- The challenge is how to define homogeneous regions
  - neighbourhoods
  - ecological zones
  - spatially-constrained cluster analysis
- The extreme is to estimate a new model at each location with a subset of neighbouring observations as the dataset
  - ‘geographically-weighted regression’



Key concepts to consider when working with environmental models

- *Data input:* quality, sources of bias, errors, consistency, training vs. testing
- *Model characteristics:* number of parameters, complexity, assumptions, representations
- *Model evaluation:* model fit, overfitting / generalizability
- *Model use:* how will models results be used, how can they be misused, etc