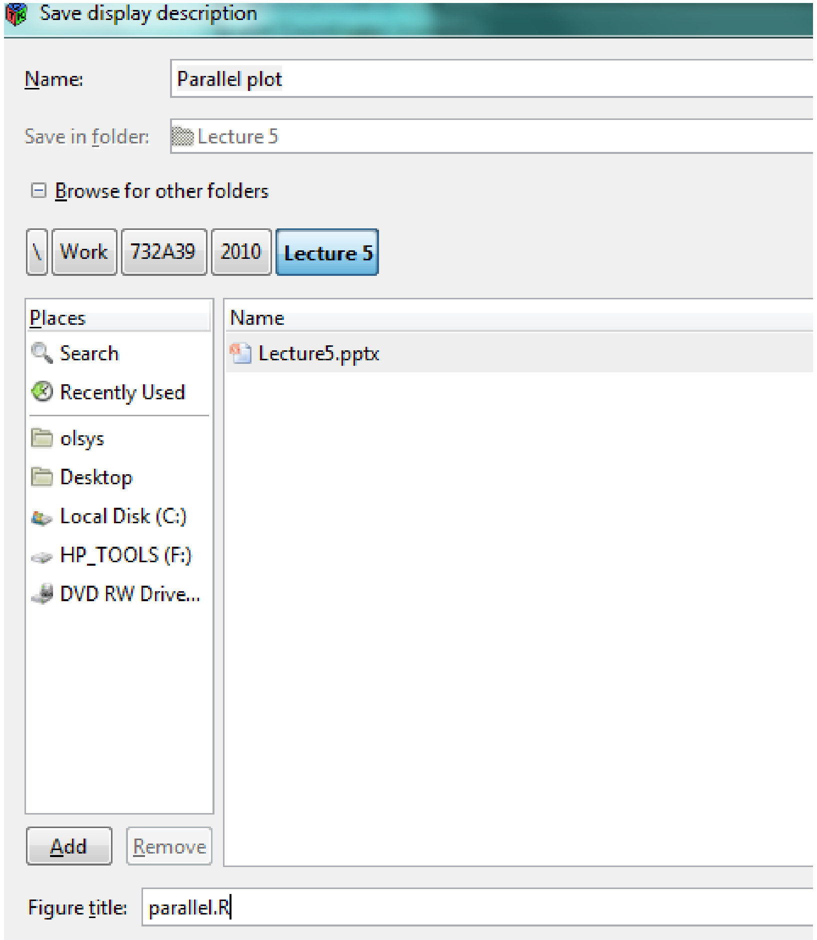
Lecture 5:

Ggobi and R
Numerical methods and visualization
Animation

Publication quality graphics

- Use Tools-> Save Display description

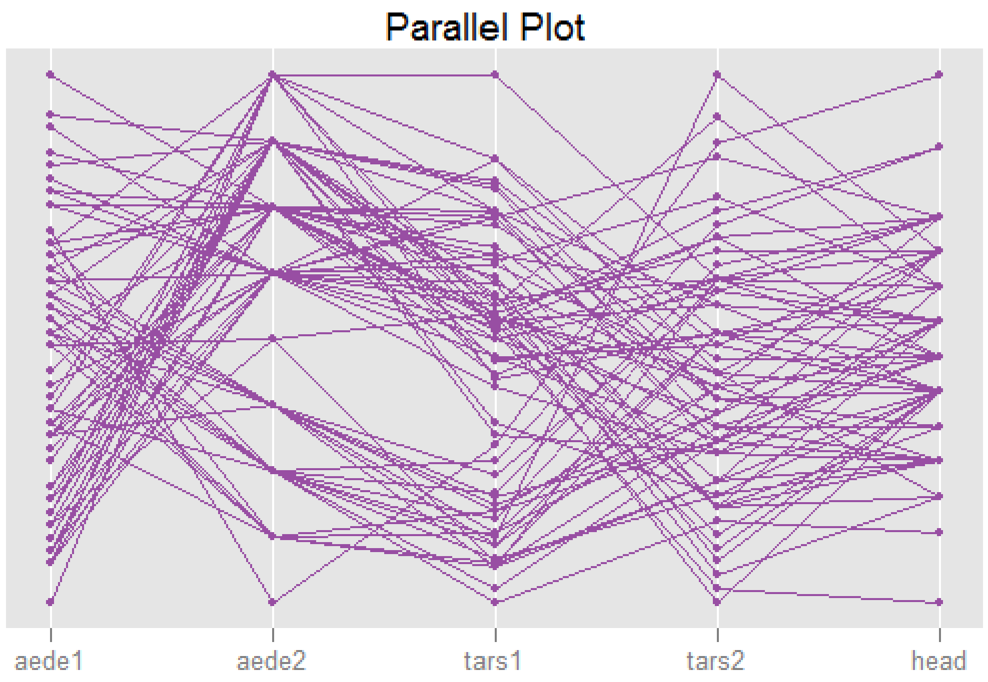


Publication quality graphics

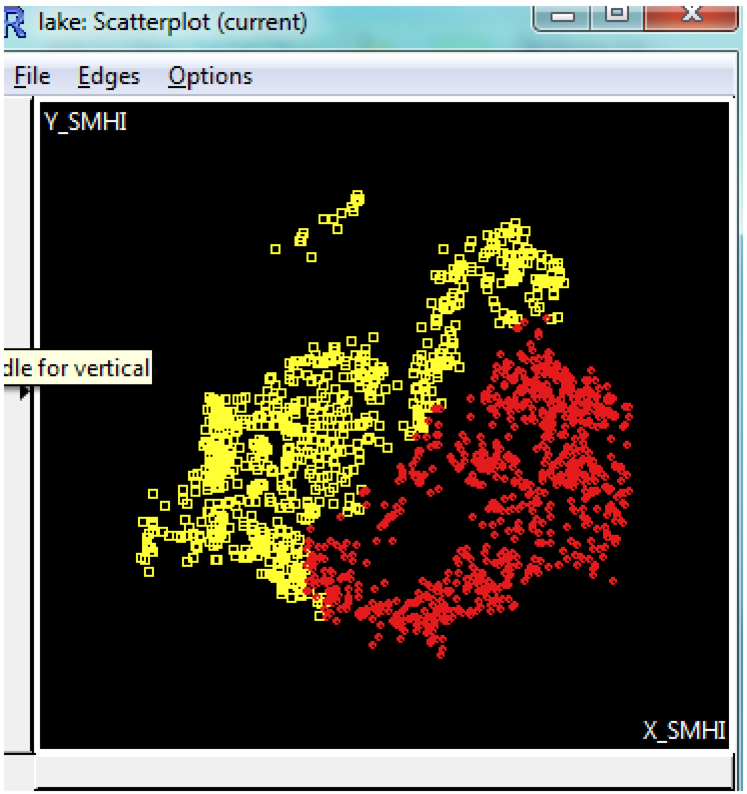
- In R:

```
library(DescribeDisplay);
d<-dd_load("parallel.R");
 ggplot(d);
```

Using R plot, you can further enhance it in Inkscape



Seeing R data using GGobi



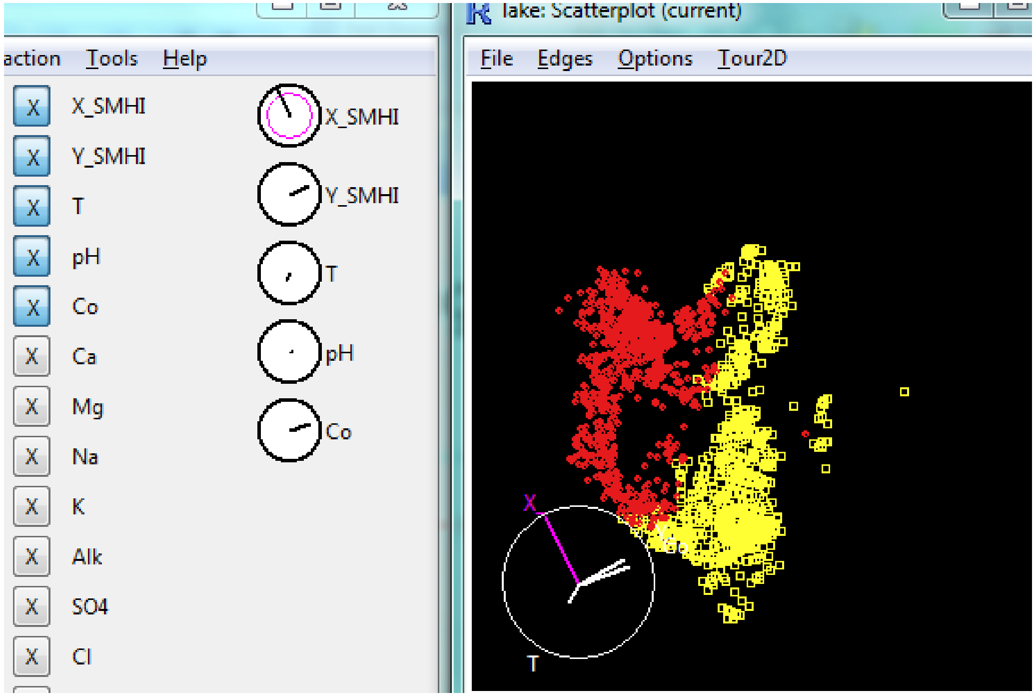

```
> lake <- read.csv("lakesurvey.csv");
> lake
  X_SMHI Y_SMHI   T   pH    Co    Ca    Mg    N
1 653600 162119 4.4 7.15  7.49 0.262 0.201 0.23
2 654145 161816 3.8 7.49 10.60 0.477 0.275 0.24
3 654353 162104 5.4 7.46 13.50 0.742 0.278 0.32
4 654646 164746 5.5 7.52 15.40 0.872 0.240 0.44
5 654804 159298 3.8 7.60 13.70 0.720 0.410 0.25
6 654847 158874 4.0 7.67 16.20 0.733 0.512 0.39
7 654893 159467 2.7 6.77 16.70 0.379 0.224 0.92
8 655284 161919 5.8 7.32  9.18 0.471 0.173 0.21
9 655587 158869 2.9 6.59  3.64 0.161 0.066 0.09
```

```
lake<-read.csv("lakesurvey.csv")
library(rggobi);
gg<-ggobi(lake);
 glyph_color(gg[1])<-c(rep(1,800), rep(2,869))
glyph_type(gg[1])<-c(rep(5,800), rep(6,869))
```

Seeing R using Ggobi

- Tours

```
> d <- displays(gg)[[1]]
> pmode(d) <- "2D Tour"
> ggobi_display_get_tour_projection(d)
               [,1]          [,2]
 [1,] -6.884923e-06  1.365615e-05
 [2,]  1.247204e-05  7.038482e-06
 [3,] -1.091227e-02 -1.702719e-02
 [4,]  2.080103e-02 -3.868532e-03
 [5,]  2.632767e-04  1.008248e-04
 [6,]  0.000000e+00  0.000000e+00
 [7,]  0.000000e+00  0.000000e+00
 [8,]  0.000000e+00  0.000000e+00
 [9,]  0.000000e+00  0.000000e+00
[10,]  0.000000e+00  0.000000e+00
[11,]  0.000000e+00  0.000000e+00
[12,]  0.000000e+00  0.000000e+00
[13,]  0.000000e+00  0.000000e+00
[14,]  0.000000e+00  0.000000e+00
```



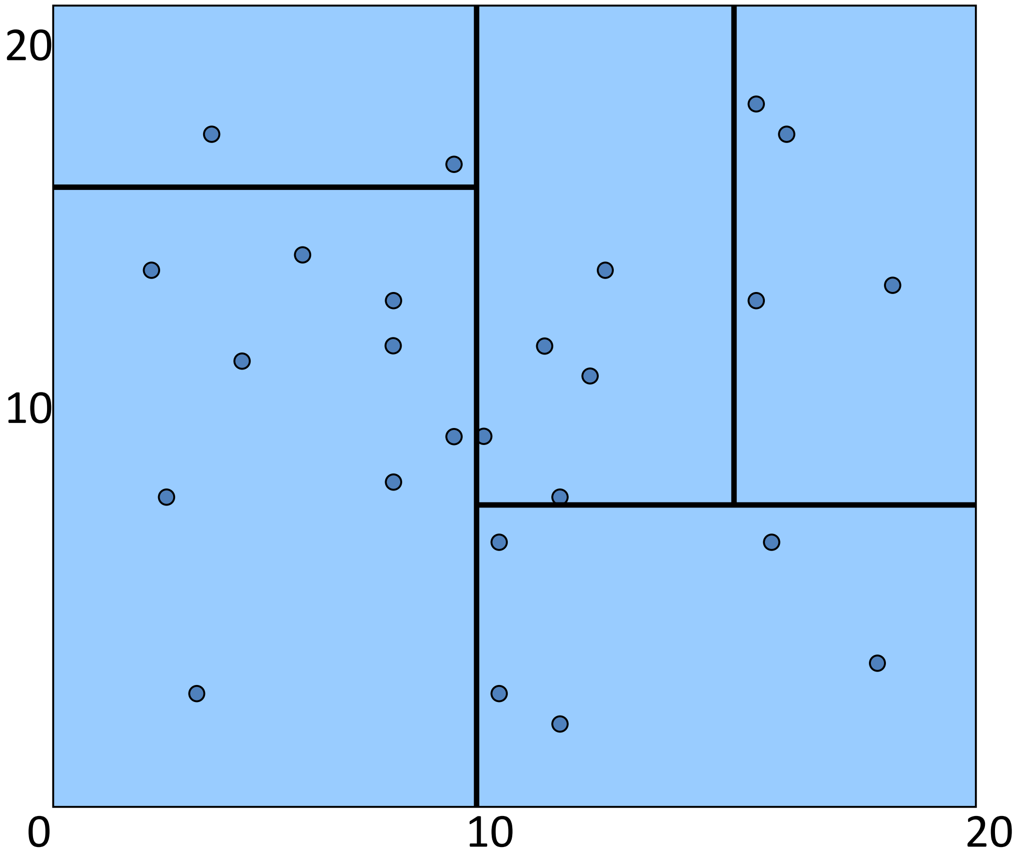
Visualizing and numerical methods

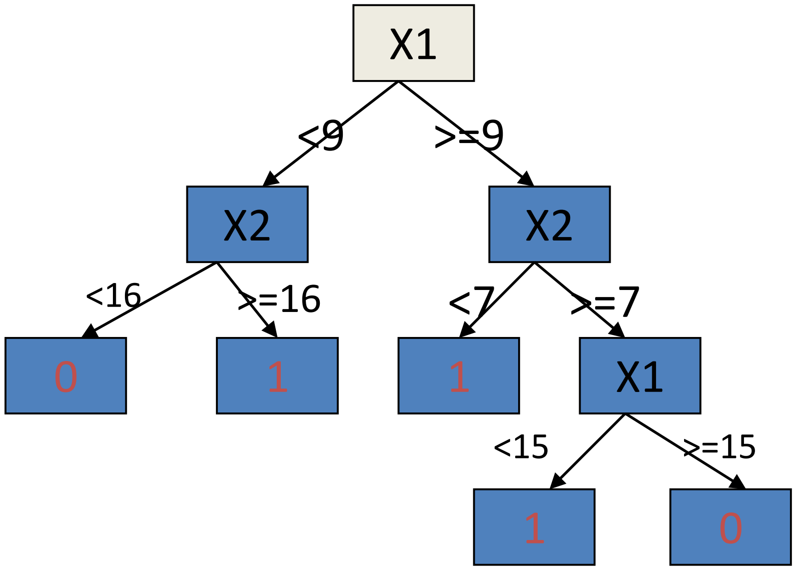
Data mining methods, aims:

- Classification (X,Y given)
  - Linear Discriminant analysis
  - Decision trees
  - Neural networks
  - …

- Clustering (X given)
  - Hierarchical clustering
  - Kohonen maps
  - …

Decision trees





Decision trees

```
+ > crabs<-read.csv("australian-crabs.csv")
> crabs
  species    sex index   FL   RW   CL   CW   BD
1    Blue   Male     1  8.1  6.7 16.1 19.0  7.0
2    Blue   Male     2  8.8  7.7 18.1 20.8  7.4
3    Blue   Male     3  9.2  7.8 19.0 22.4  7.7
4    Blue   Male     4  9.6  7.9 20.1 23.1  8.2
5    Blue   Male     5  9.8  8.0 20.3 23.0  8.2
6    Blue   Male     6 10.8  9.0 23.0 26.5  9.8
7    Blue   Male     7 11.1  9.9 23.8 27.1  9.8
```

```
crabs<-read.csv("australian-crabs.csv");
library(rpart);
tree<-rpart(species~FL+RW+CL+CW+BD,data=crabs);
```

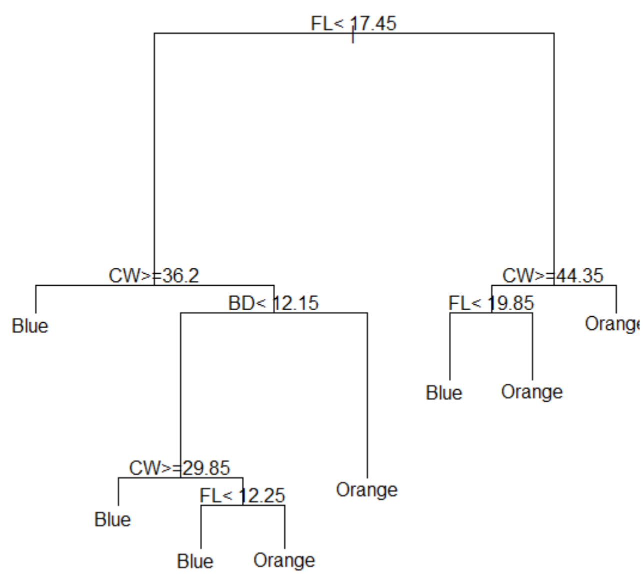
Decision trees

```
> library(rpart)
> tree<-rpart(species~FL+RW+CL+CW+BD, data=crabs)
> tree
n= 200

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 200 100 Blue (0.50000000 0.50000000)
   2) FL< 17.45 135  47 Blue (0.65185185 0.34814815)
     4) CW>=36.2 40   4 Blue (0.90000000 0.10000000) *
     5) CW< 36.2 95  43 Blue (0.54736842 0.45263158)
      10) BD< 12.15 62  13 Blue (0.79032258 0.20967742)
        20) CW>=29.85 21   0 Blue (1.00000000 0.00000000) *
        21) CW< 29.85 41  13 Blue (0.68292683 0.31707317)
          42) FL< 12.25 34   6 Blue (0.82352941 0.17647059) *
          43) FL>=12.25 7   0 Orange (0.00000000 1.00000000) *
      11) BD>=12.15 33   3 Orange (0.09090909 0.90909091) *
   3) FL>=17.45 65  12 Orange (0.18461538 0.81538462)
     6) CW>=44.35 33  12 Orange (0.36363636 0.63636364)
      12) FL< 19.85 11   0 Blue (1.00000000 0.00000000) *
      13) FL>=19.85 22   1 Orange (0.04545455 0.95454545) *
     7) CW< 44.35 32   0 Orange (0.00000000 1.00000000) *
```

Decision trees

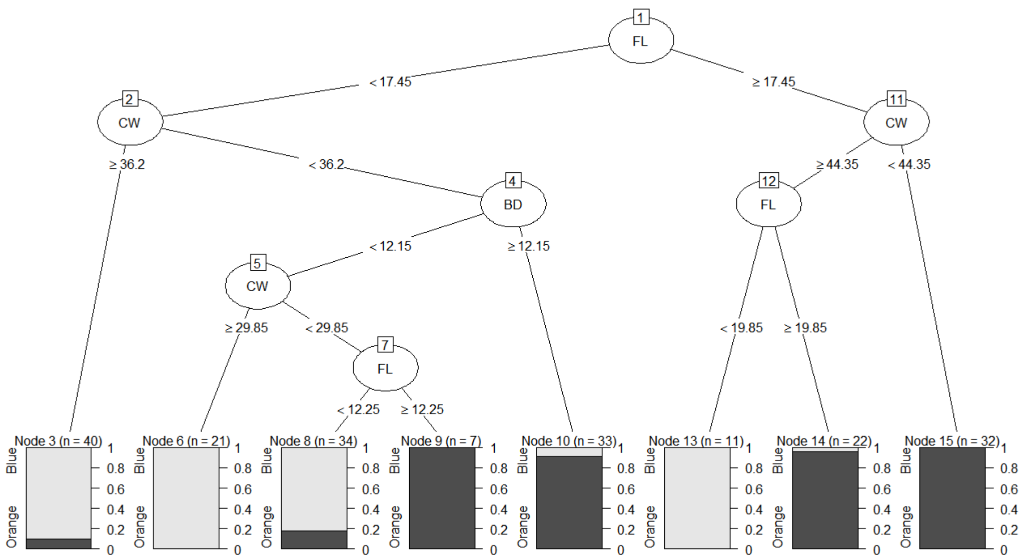
- Plotting the tree:

```
plot(tree);
text(tree);
```

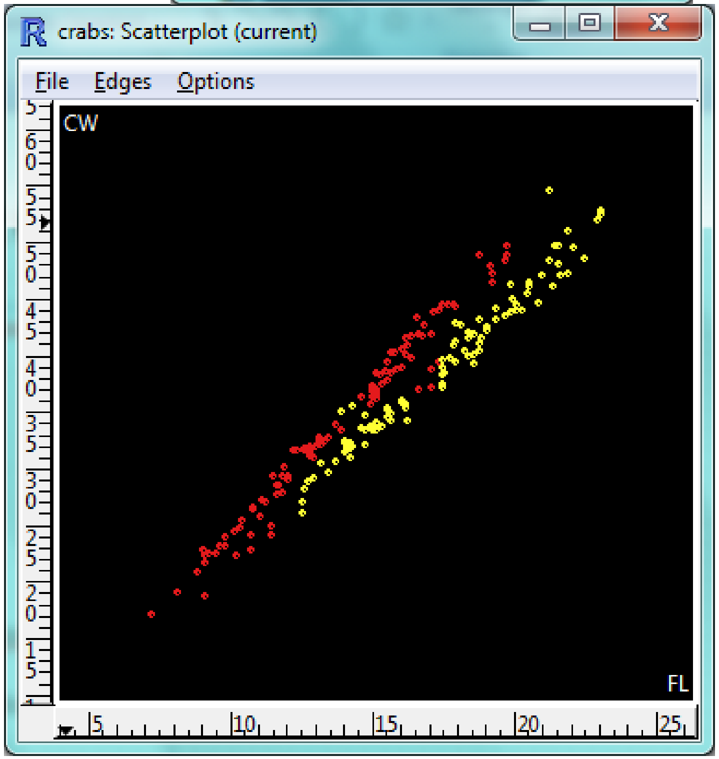


- A better visualization:
  - Use library **partykit**

```
plot(as.party(tree))
```



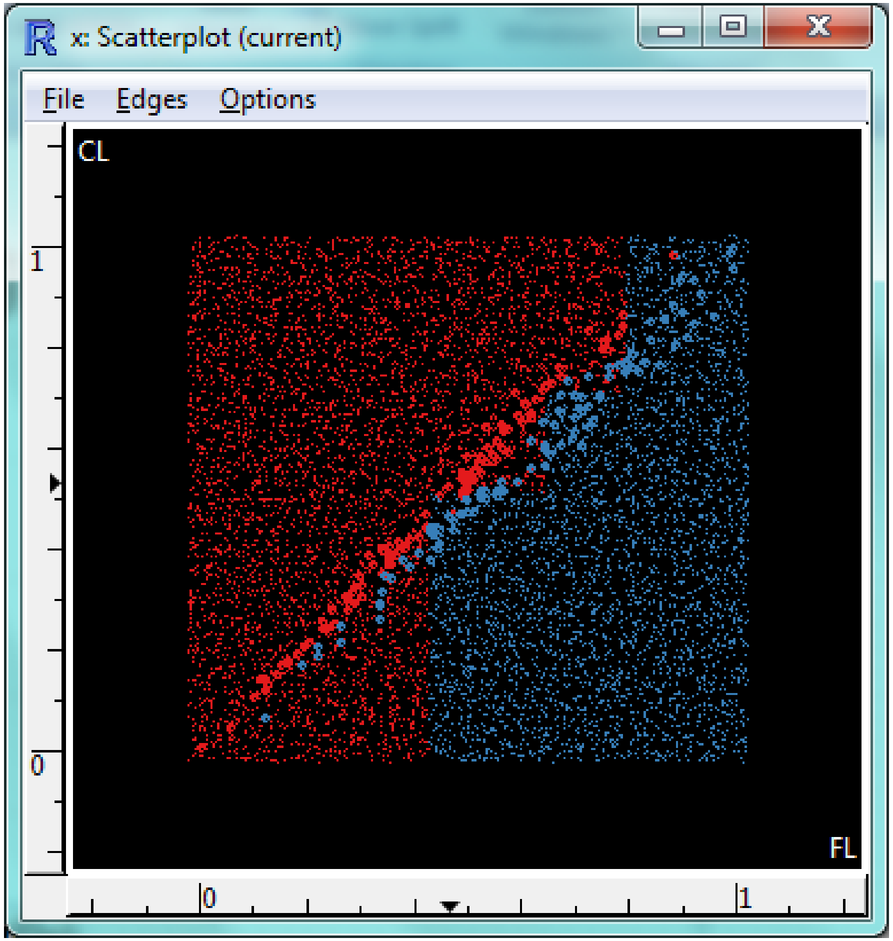
```
> probabilities<-predict(tree, newdata=crabs)
> probabilities
       Blue     Orange
1   0.82352941 0.1764706
2   0.82352941 0.1764706
3   0.82352941 0.1764706
4   0.82352941 0.1764706
5   0.82352941 0.1764706
6   0.82352941 0.1764706
7   0.82352941 0.1764706
```




```
crabspredict<-
    probabilities[1,]>probabilities[2,];
gg<-ggobi(crabs);
glyph_color(gg[1])<-crabspredict+1;
```

- SEE 2D-tour also

Seeing boundaries



Use "classifly" for different methods (decision trees, neural networks, LDA)

1. Close all Ggobi windows
2. Run:

```
library(classifly);
classifly(crabs, species~FL+CL+RW, rpart);
```

Clustering

- To find groups of object that are close to each other and separated from other groups

Hierarchical clustering:

1. Put each observation in a cluster
2. Merge 2 nearest clusters into 1 cluster
3. Repeat 2 until you get 1 cluster

Concepts:

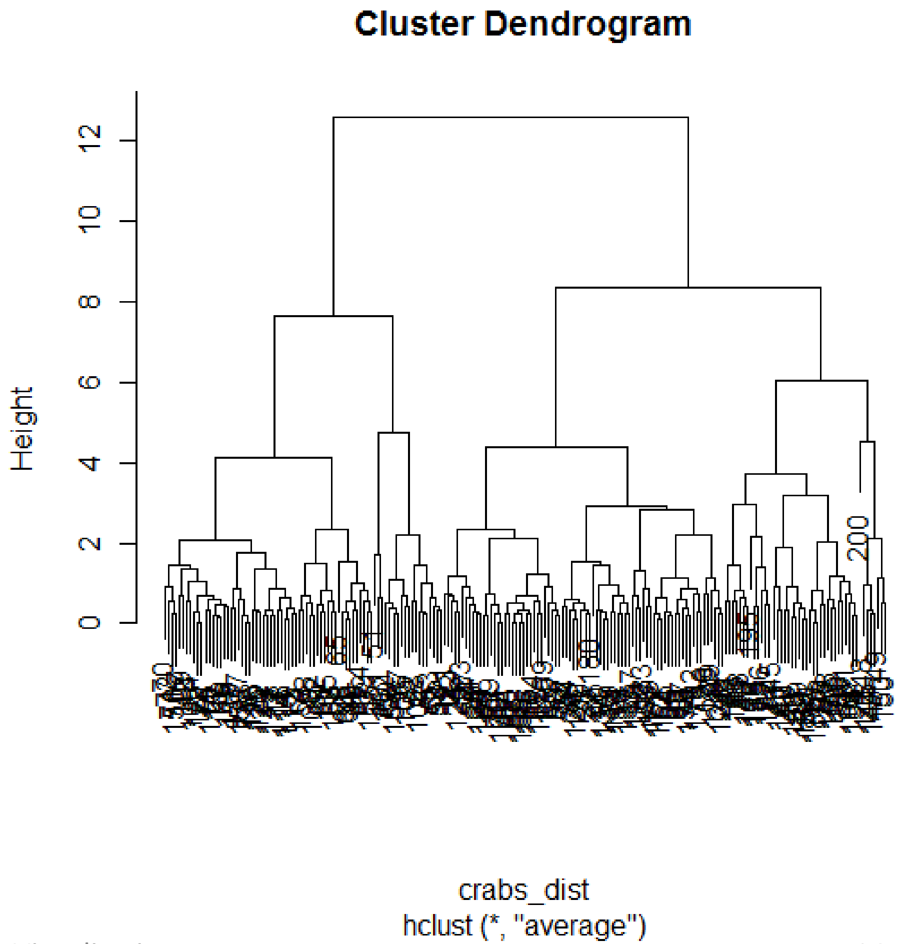
- Dendrogram
- Threshold

→ Use the given shreshold to define the amount of clusters.

Clustering

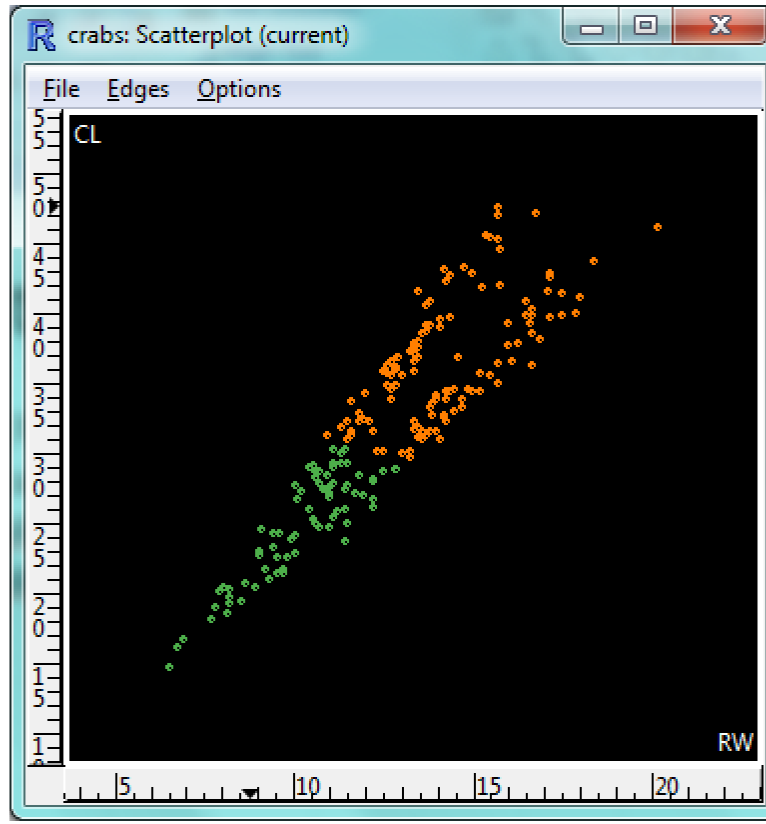
- See dendrogram

```
x<-data.frame(crabs$CL, crabs$RW);
crabs_dist<-dist(x);
crabs_dend<-hclust(crabs_dist,
   method="average")
plot(crabs_dend);
```



Clustering

```
gd<-gg[1];
clust2=cutree(crabs_dend, k=2);
glyph_color(gd)[clust2==1]<-4;
glyph_color(gd)[clust2==2]<-5;
```



Dynamic plots

Interactive and dynamic plots

- Interactive= user can manipulate the plot to change it
- Dynamic= user is observing a moving picture

- Interactive:
  - All GGobi plots

- Dynamic
  - Tours
  - Motion charts

Animation: good or bad?

- Advantages
  - Effective at attracting attention
  - Same object can be used at another time point → visualizing one more dimension with same graphs (time)
  - Easily perceived in peripheral vision →many features can be captured at one time point
- Disadvantages
  - Can be powerful force for destraction
  - Unappropriate transformation (transition)→ false conclusions
  - Speed of the graphics may hide important details /make it too boring (unefficient)

Animation: recommendations

- Maintain valid data during transitions
  - Example: using splines
- Careful when using interpolations: use appropriate models
- Group similar transitions
- Minimize occlusion
- Use simple transistions
- If trajectory is stable, use slow-in, slow-down
- Make transition as long as needed but no longer.

Gapminder

- See http://www.gapminder.org

- Gapminder is
  – A database that stores important world statistics (by country)
  – Each dataset can be analysed online with **Motion Chart**
  – Offline version can be downloaded

1. Go to Gapminder
2. Click Data and look through available datasets
3. Choose set ”Life expectancy (years)”
4. Run and analyse Motion Chart

Gapminder

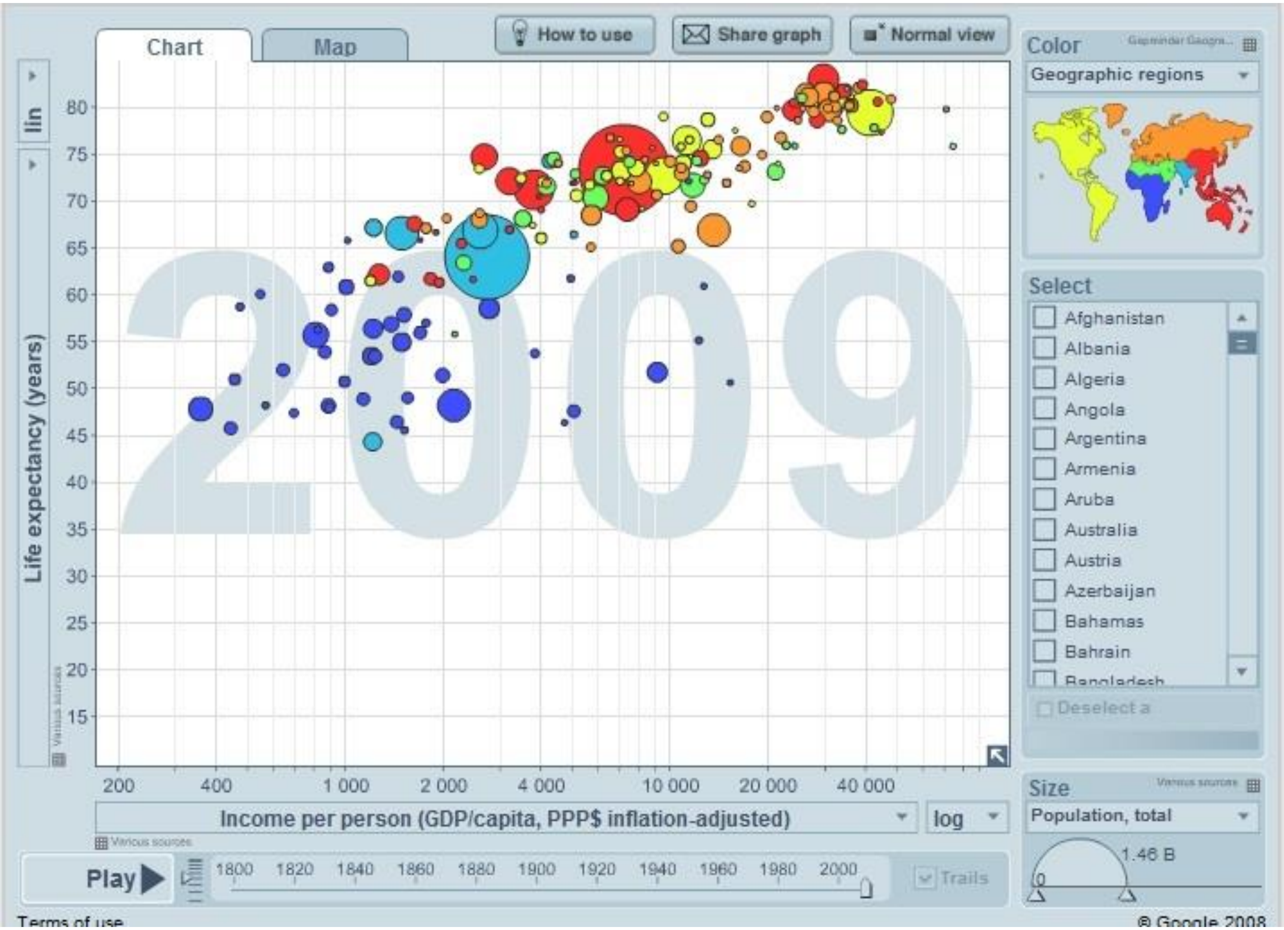


Chart
Map
How to use
Share graph
Normal view
lin
Life expectancy (years)
80
75
70
65
60
55
50
45
40
35
30
25
20
15
2009
200
400
1 000
2 000
4 000
10 000
20 000
40 000
Income per person (GDP/capita, PPP$ inflation-adjusted)
log
Play
1800
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
Trails
Color
Geographic regions
Select
Afghanistan
Albania
Algeria
Angola
Argentina
Armenia
Aruba
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Size
Population, total
1.46 B
Terms of use
© Google 2008

Gapminder

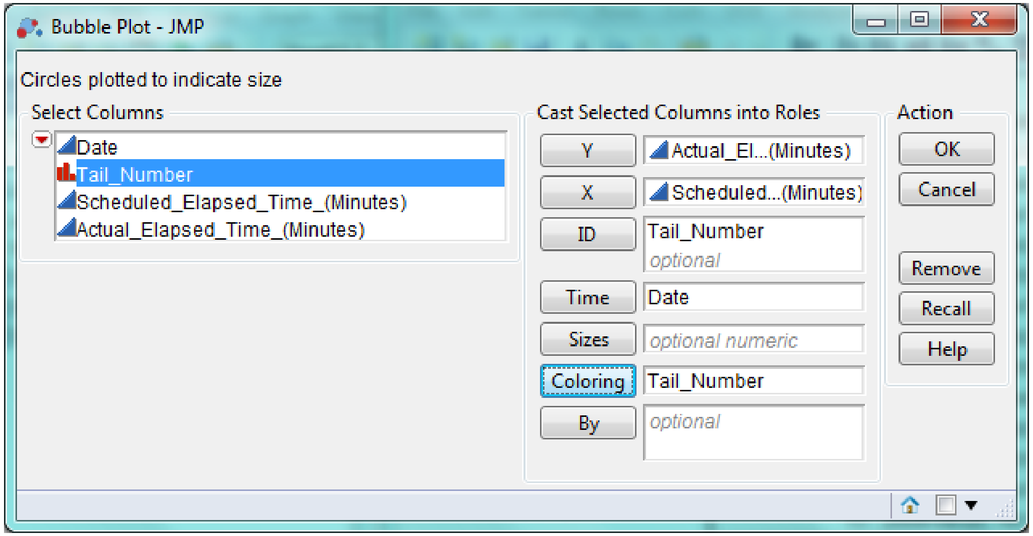
- Comment available tools:
  - Axes
  - Lin/log
  - Time scale
  - Size indicator and bubble size
  - Color (how countries are marked)
  - Select countries
  - Identify (move mouse over a bubble)
  - Play
  - Speed of video
  - Trails
  - Chart/Map
  - Full screen
  - Opacity (visibility of the non-selected countries)
  - Zoom
  - How to use

Gapminder

- Life expectancy set
  - How the minimum and maximum life expectancies changed after 200 years
  - Which countries had in general the highest life expectancy? Lowest lifexpectancy? Highest income level? Lowest income level?
  - What happened after 1946 and why
  - Compare development of China and USA (mark this countries and use ”trace”)

Motion Chart - JMP

- Excel file should contain
  - 1 column= observation name (country)
  - 1 column= time
  - 2 column= X och Y variables
  - More columns= Region, Bubble size



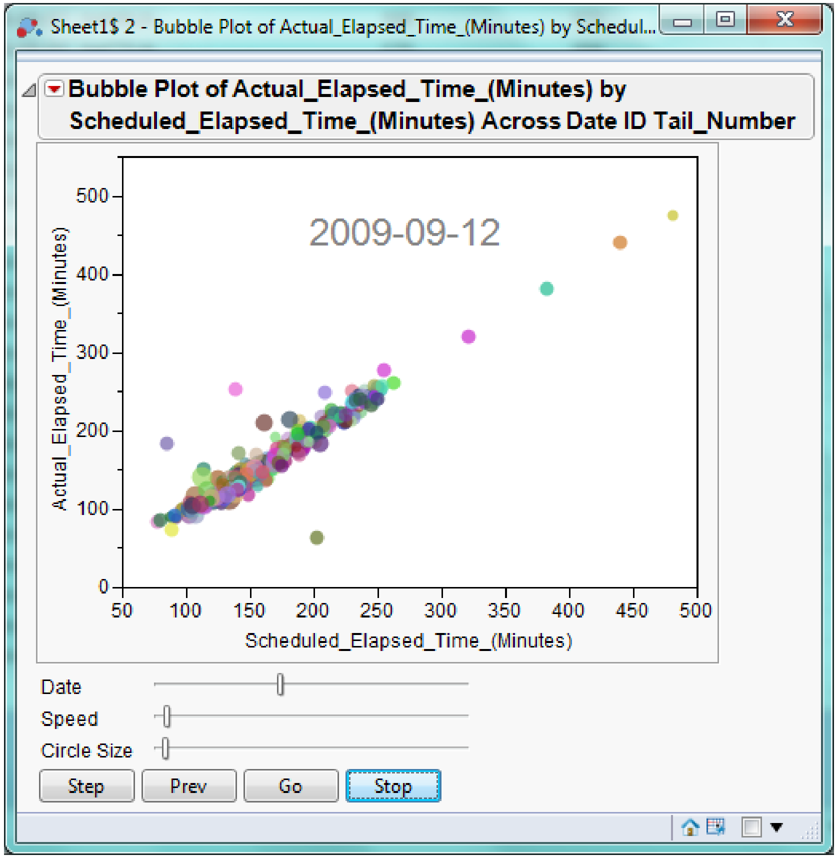
- SAS JMP:
  - Graph→ Bubble plot

Example

Delays of the airlines

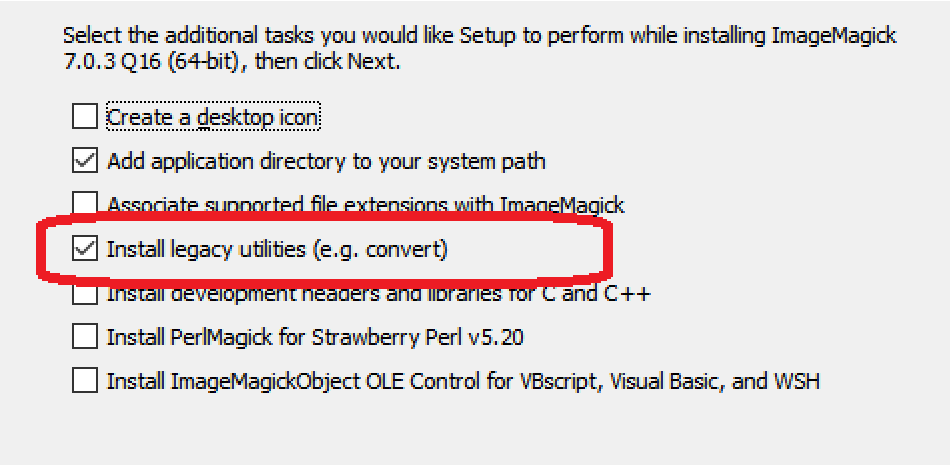
Motion Chart - JMP

- See which airlines delay or come earlier (deviations Y=X)
- Mark one or more bubbles and then use “Trail bubbles” or “Trail lines”
- Show legend
- Choose speed, pause



Dynamic from static

- Dynamic graphs can be created by combining several static graphs
- Install ImageMagick to make GIF animation
  - Mark option ”Install legacy utilities”
- Download FFMPEG for video files
- Package **animation**



Animation

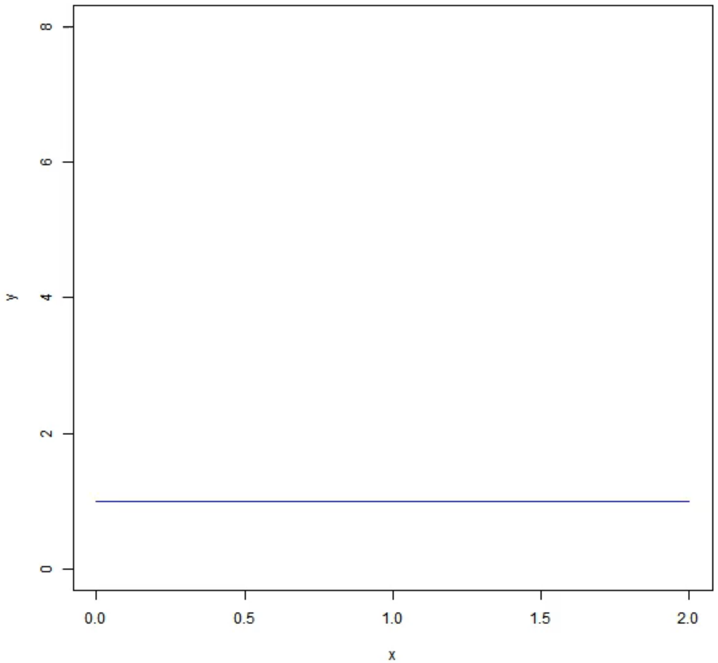
- saveGIF(code, movie.name, interval, ani.width, ani.height,…)
  - Code: the loop producing animation
  - Interval: time interval between frames
  - Ani.width, ani.height – animation size

- saveVideo(code, video.name,…)
- saveLatex(…)
  - PDF animation

Dynamic from static

Example: Visualizing $y=x^a$, a=0..3

```
#specifying where ffmpeg is located
ani.options(ffmpeg="D:\\ffmpeg\\bin\\ffmpeg
.exe")


saveVideo({
 x<-seq(0,2,0.01)
   for (a in seq(0,3,by=0.01)) {
     y<-x^a
     plot(x,y, type="l", col="blue", ylim=c(0,8))
   }
},video.name="Z:\\732A39_732A98\\2016\\Le
cture 5\\Powers.mp4", interval=0.05,
ani.width=600,ani.height=600
)
```



Reading home

- http://www.gapminder.org
- Chapter 8, p.302-326
- Heer, Jeffrey, and George Robertson. "Animated transitions in statistical data graphics." *IEEE transactions on visualization and computer graphics* 13.6 (2007): 1240-1247.