Part II

Scatter plots

Outline

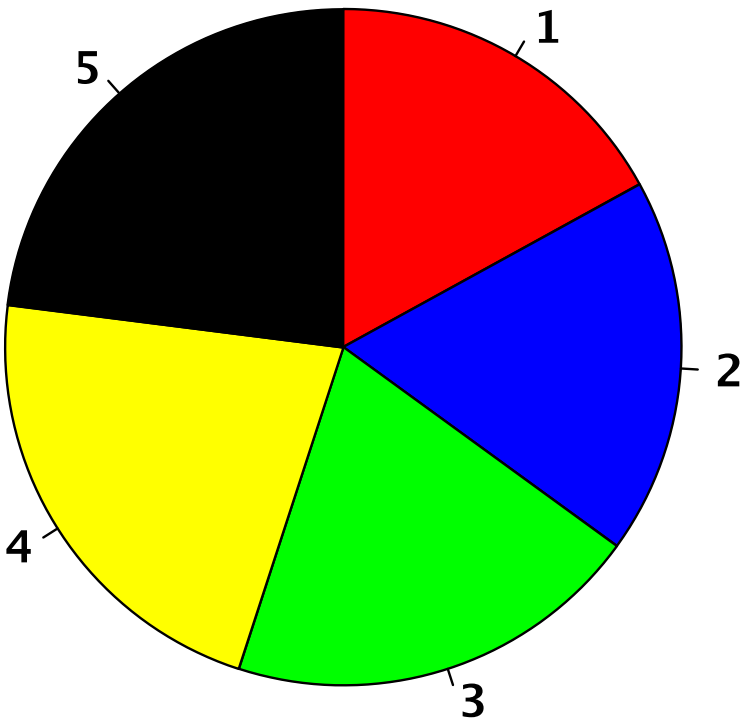
Standard 2D/3D displays

"Old fashioned" methods from exploratory data analysis:

- bar chart and histogram
- boxplot
- scatter plot
- run chart
- etc.

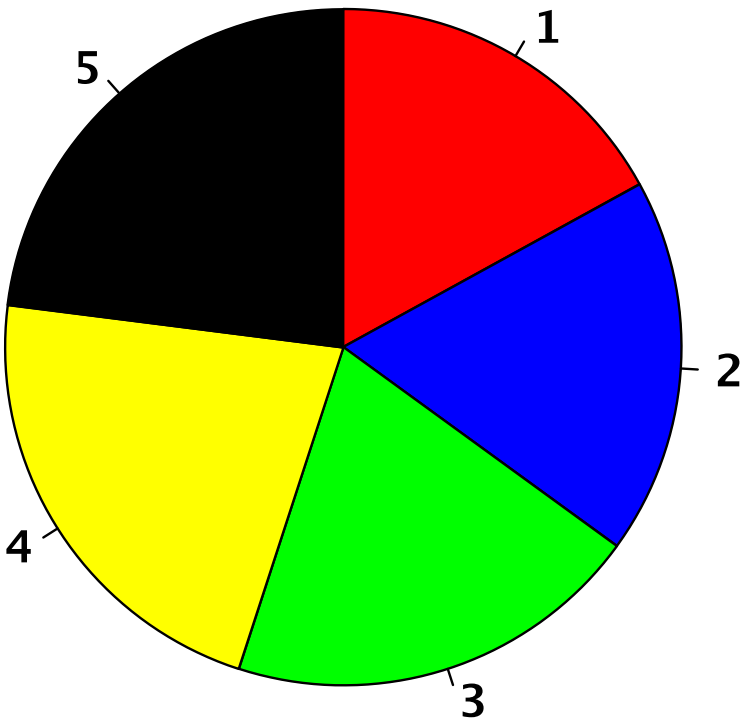
They are still useful!

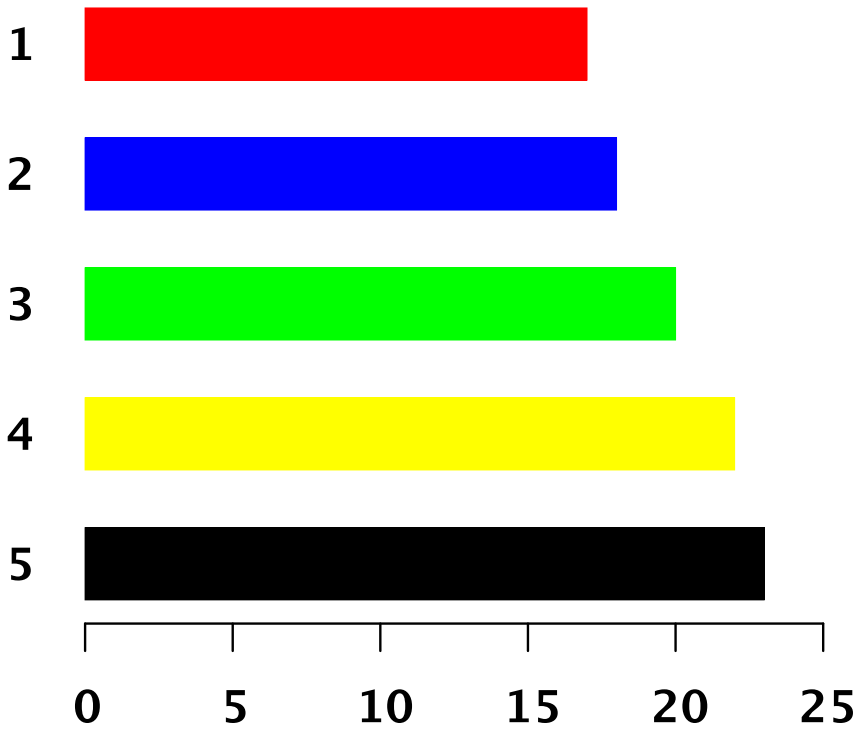
Don't use pie chart (never ever! no kidding!)



http://en.wikipedia.org/wiki/Image:Piechart.svg

Don’t use pie chart (never ever! no kidding!)





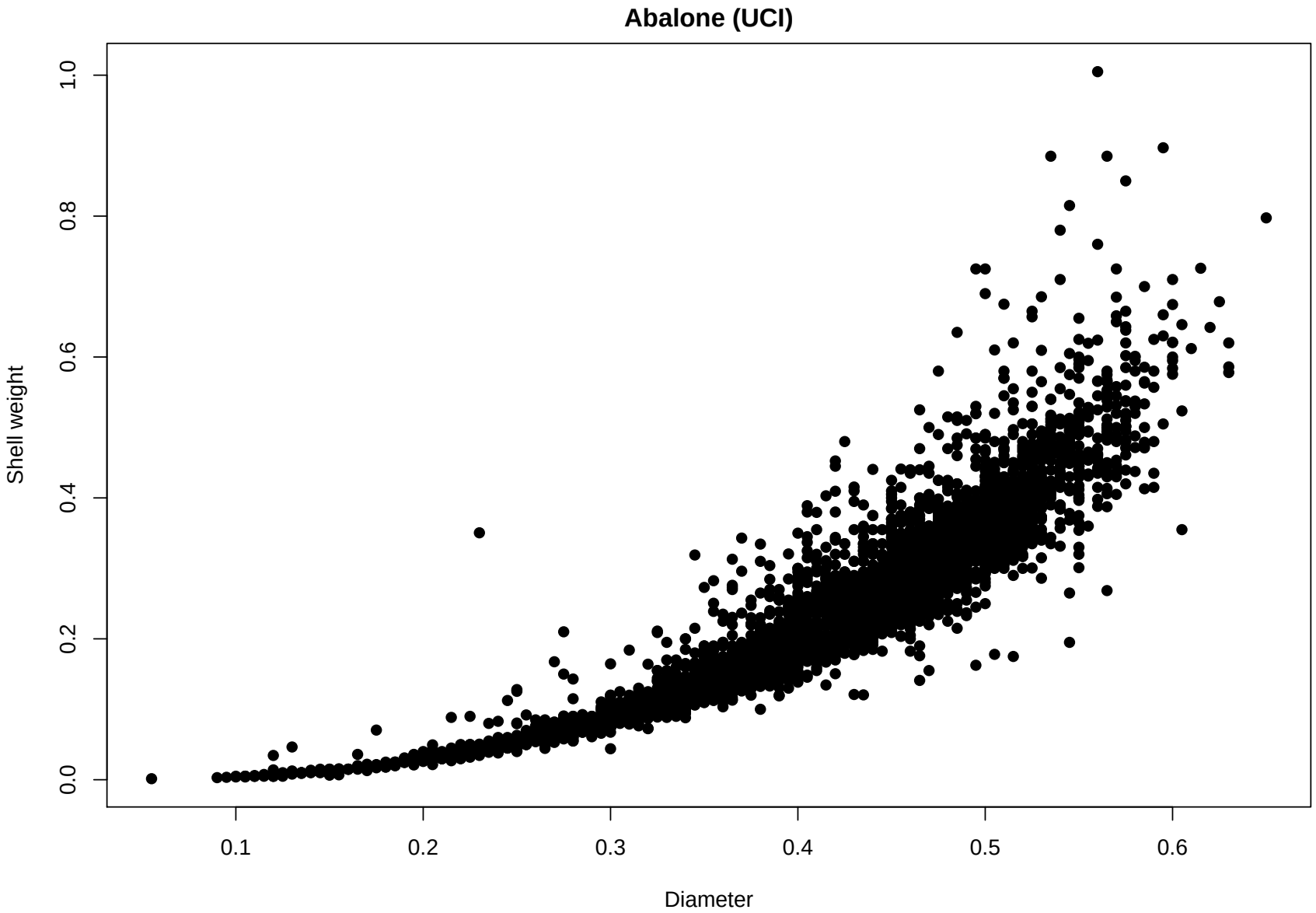
http://en.wikipedia.org/wiki/Image:Piechart.svg

Scatter plot

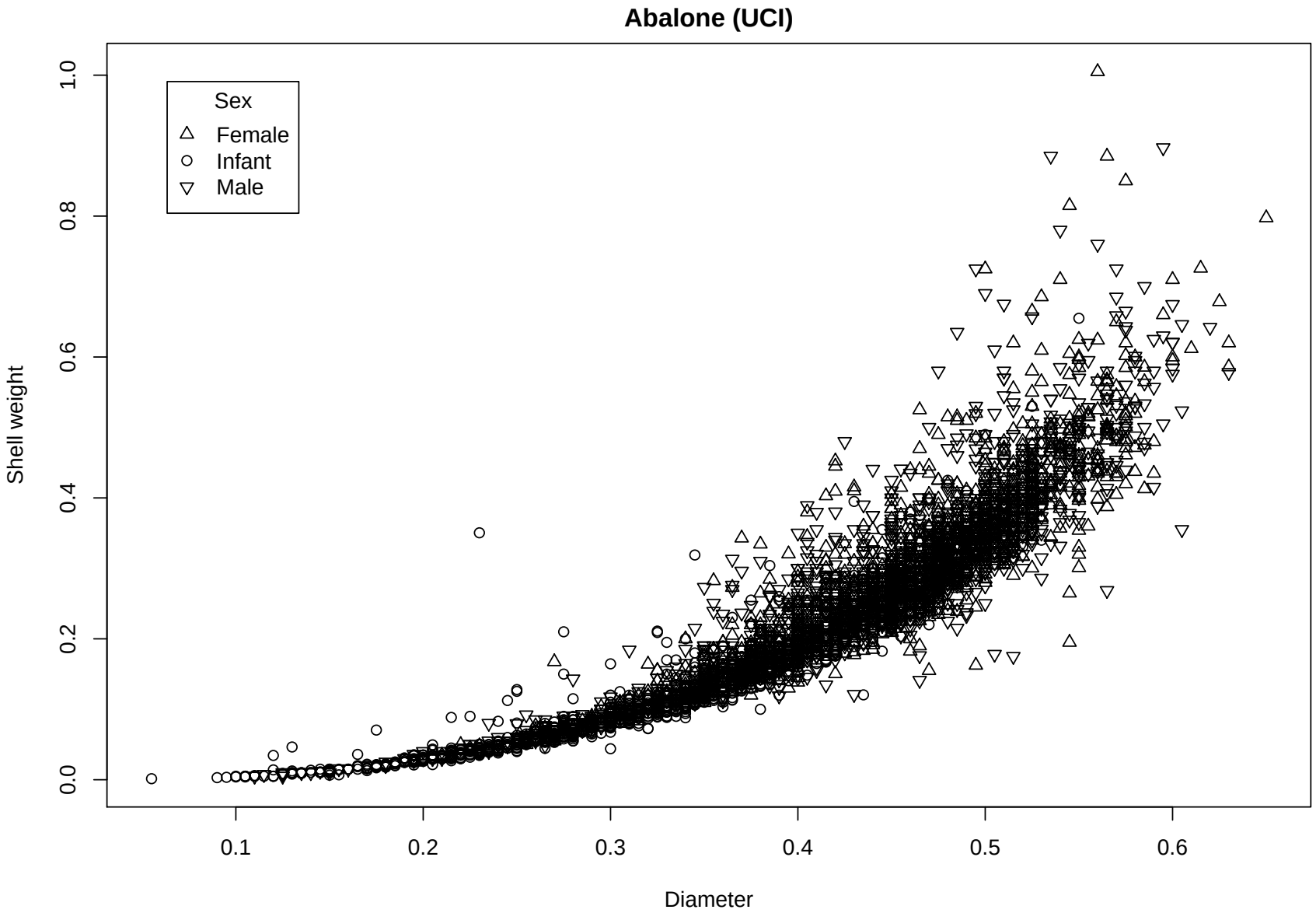
The standard tool

- one point per object
- two dominant numerical characteristics per object
- a few additional characteristics:
  - a nominal variable (hue or shape coded)
  - a numerical variable (lightness or shape coded)
  - a label
- doesn't scale:
  - low dimension only
  - very sensitive to overlapping

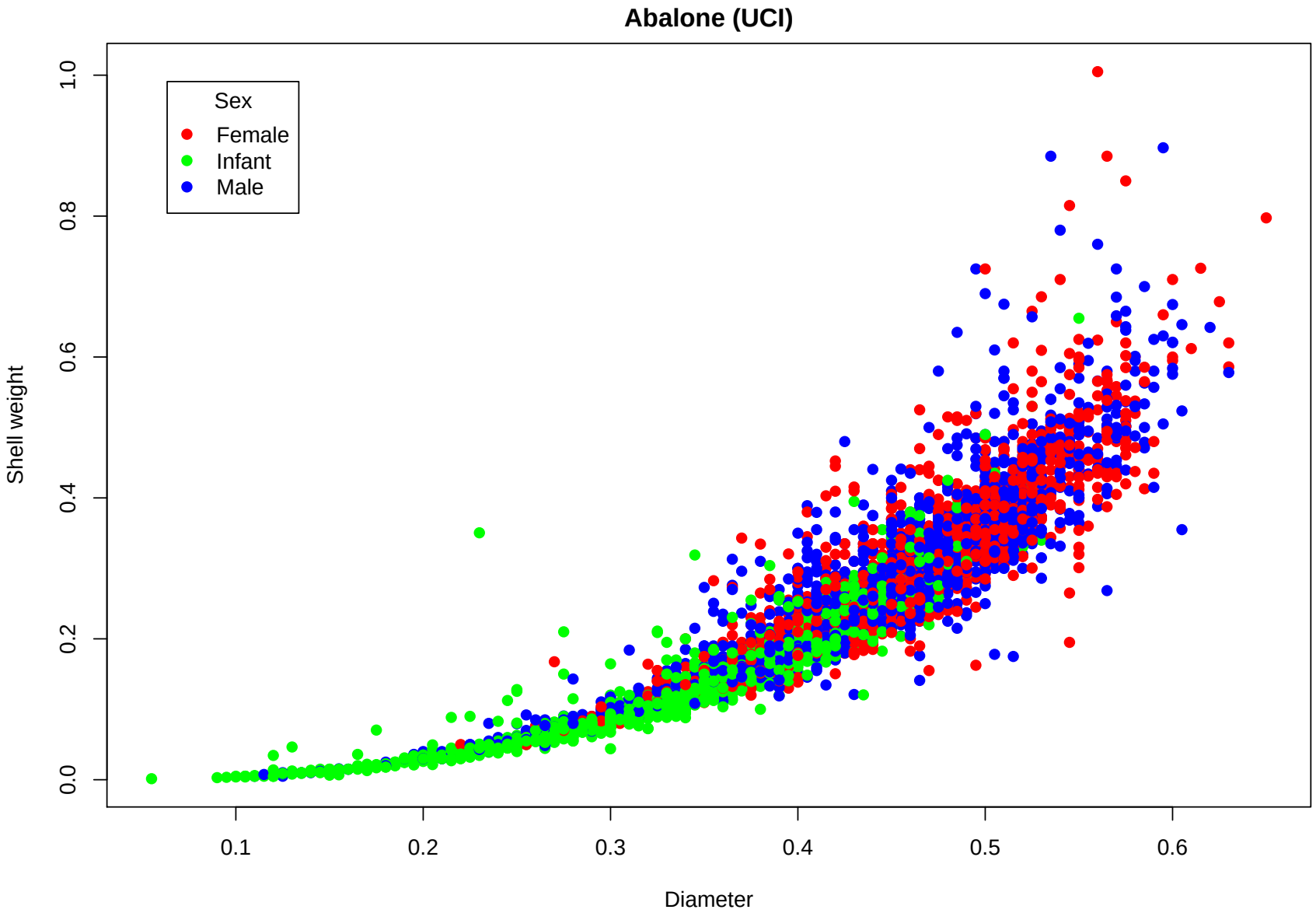
Abalone dataset



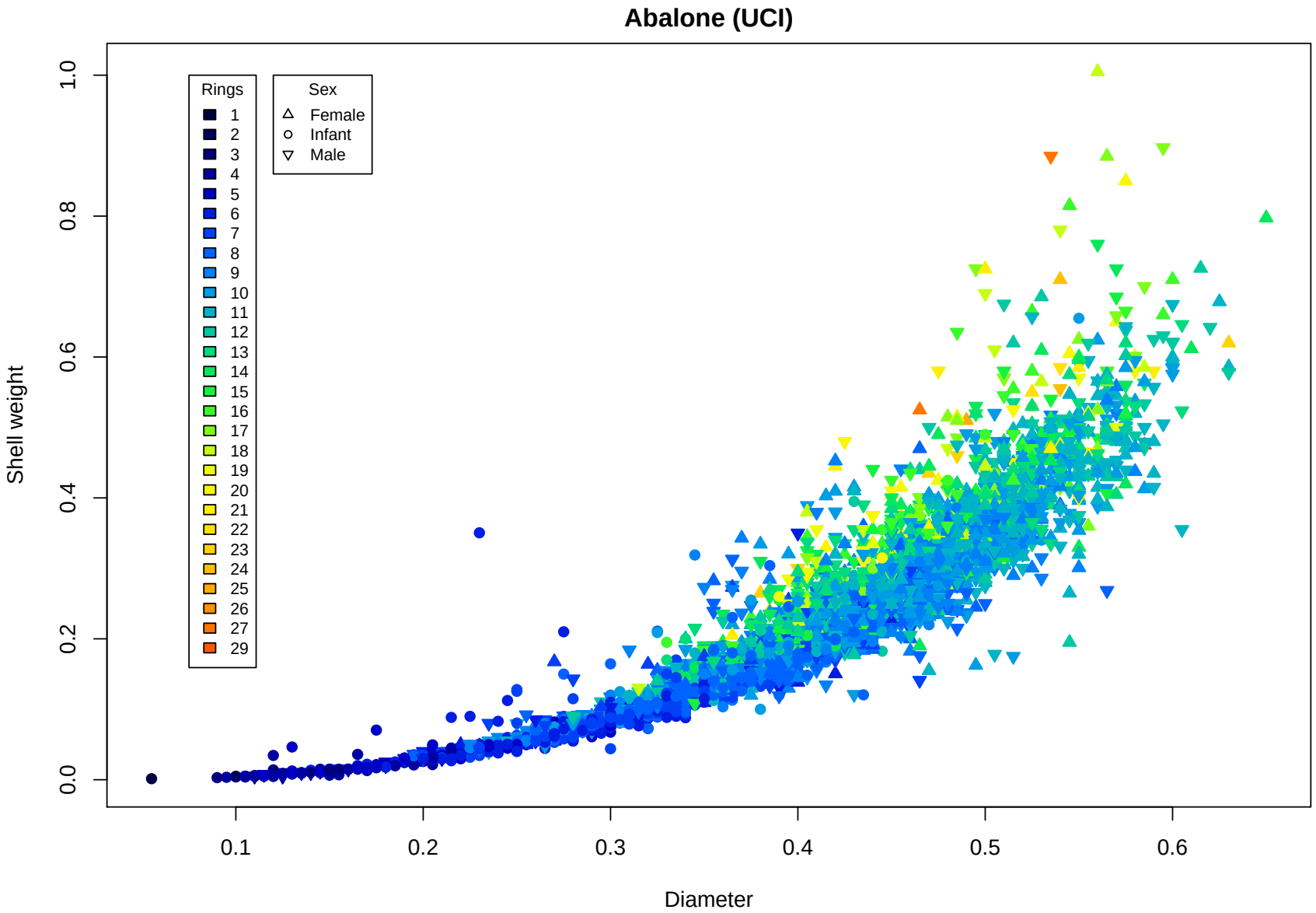
Abalone dataset



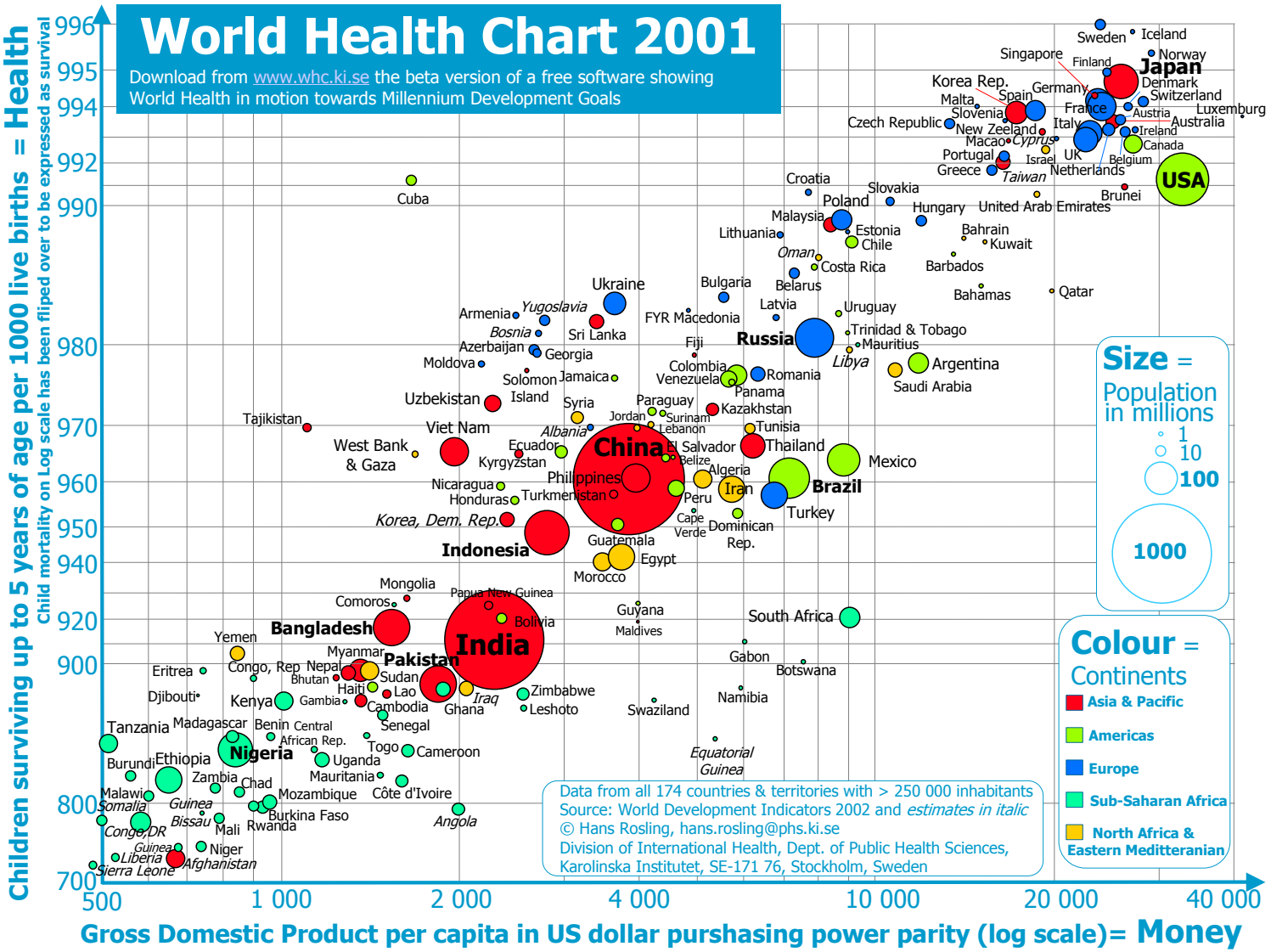
Abalone dataset



Abalone dataset



Word Health Chart



How to improve scalability?

Two scalability problems:

1. Dimension (a.k.a. variable or characteristics or feature):
   - Reduce the number of dimension:
     - user choice
     - automated: selection and extraction
   - Display more than 2/3 dimensions at once: visual layout
2. Object:
   - Reduce the number of objects: clustering and quantization
   - Reduce the size of an object
   - Constrain the display to forbid (or reduce) overlapping

Outline

Automated scalability

Reducing the number of variables (Machine Learning):

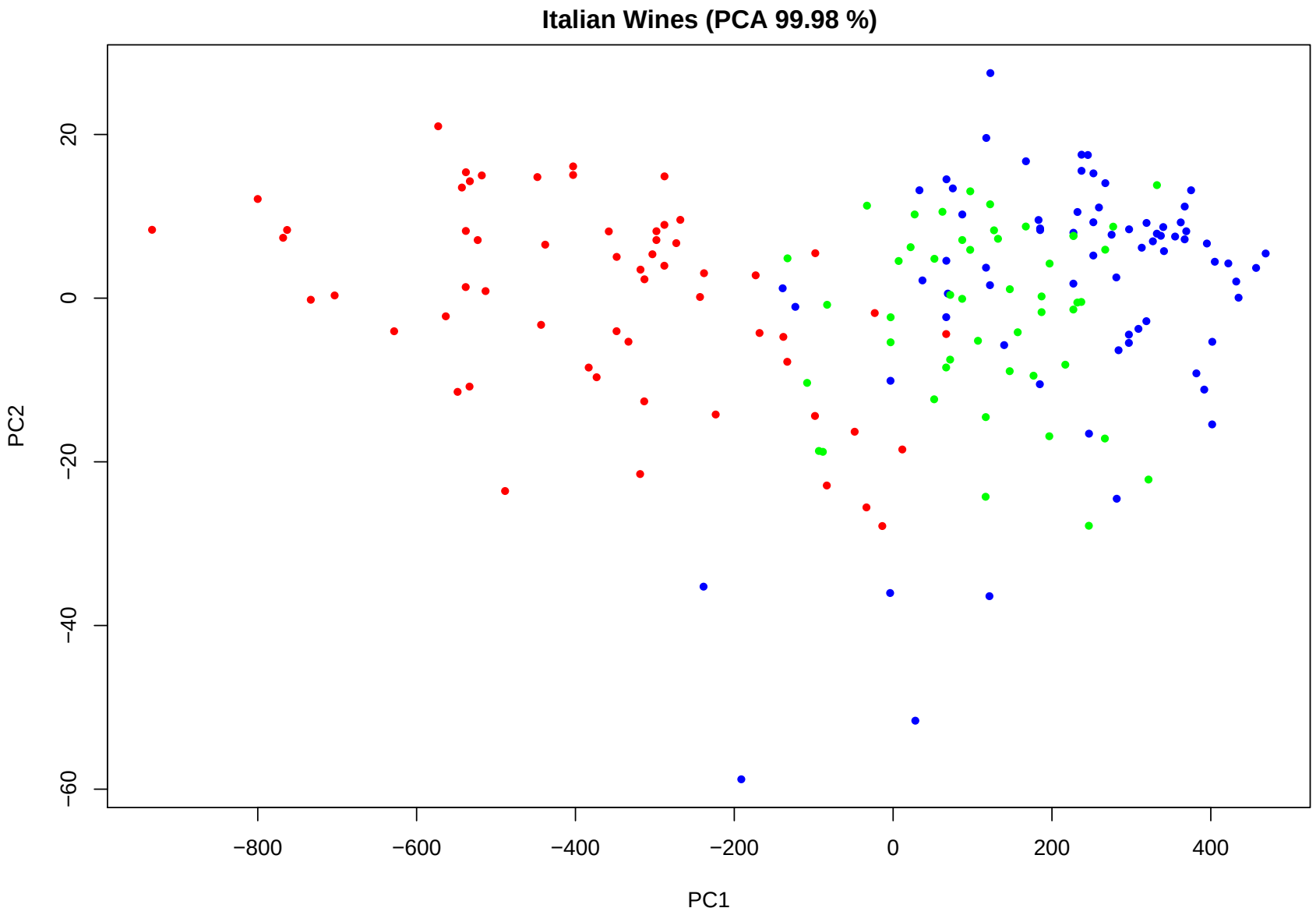
- A major research topic
- Feature selection: keeping a subset of the original variables
- Feature extraction: producing new variables in small quantity
- Supervised: variables should “explain” a target variable
- Or Unsupervised: variables optimize some quality criterion
- Some methods:
  - Principal Component Analysis
  - Linear Discriminant Analysis
  - Sammon’s Mapping
  - Variable Clustering
  - etc.
- see Prof. Lee’s lecture

Difficulties

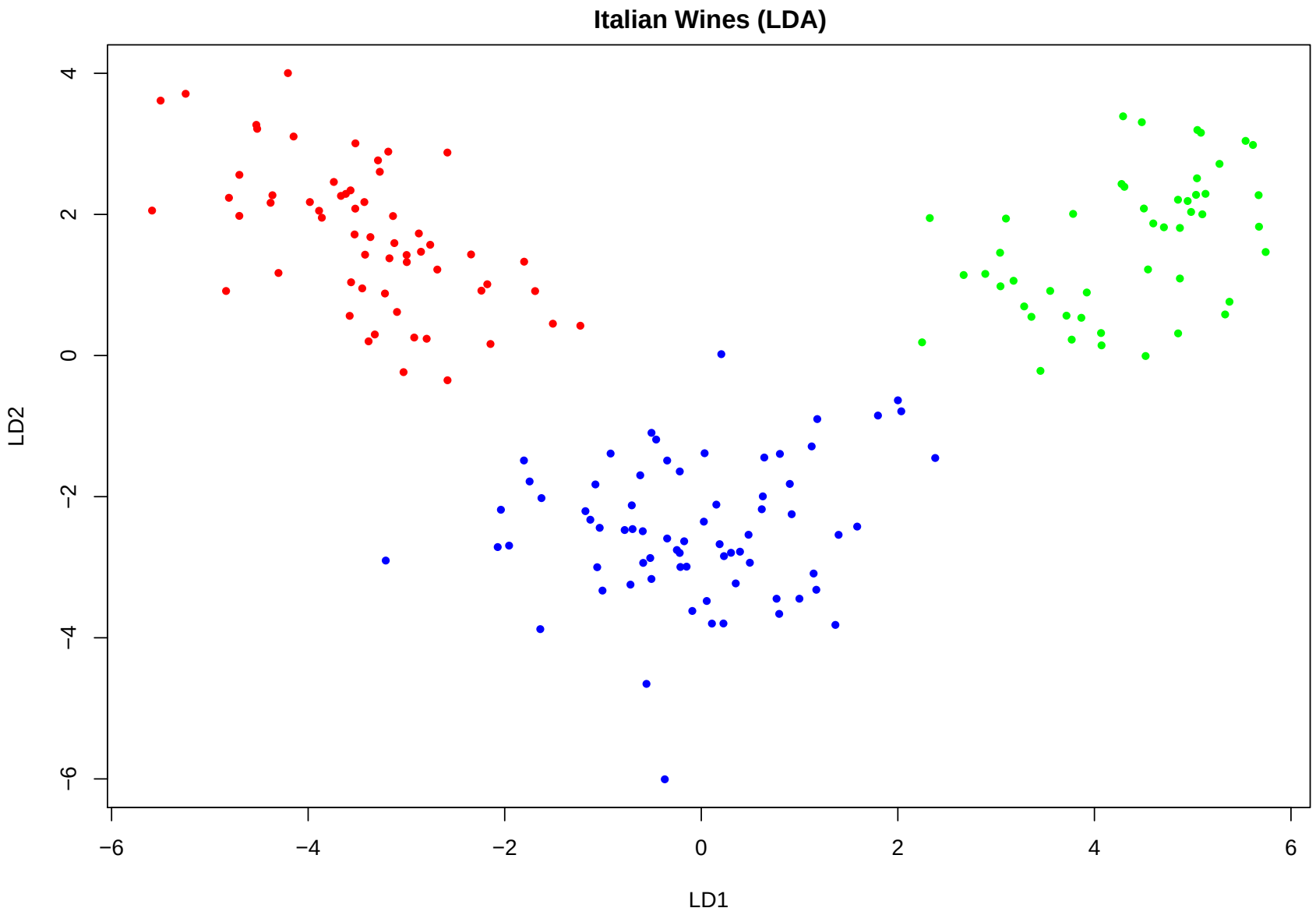
What should be optimized?

- Goal dependent:
  - outlier detection $\Rightarrow$ maximize distances between outliers and central objects
  - visual clustering $\Rightarrow$ respect the neighborhood relationships
  - rule finding $\Rightarrow$ keep original variables
  - etc.
- Machine learning algorithms optimize abstract quantities (e.g., reconstruction error)
- Links between ML optimality and visual usefulness is unclear (at best!)

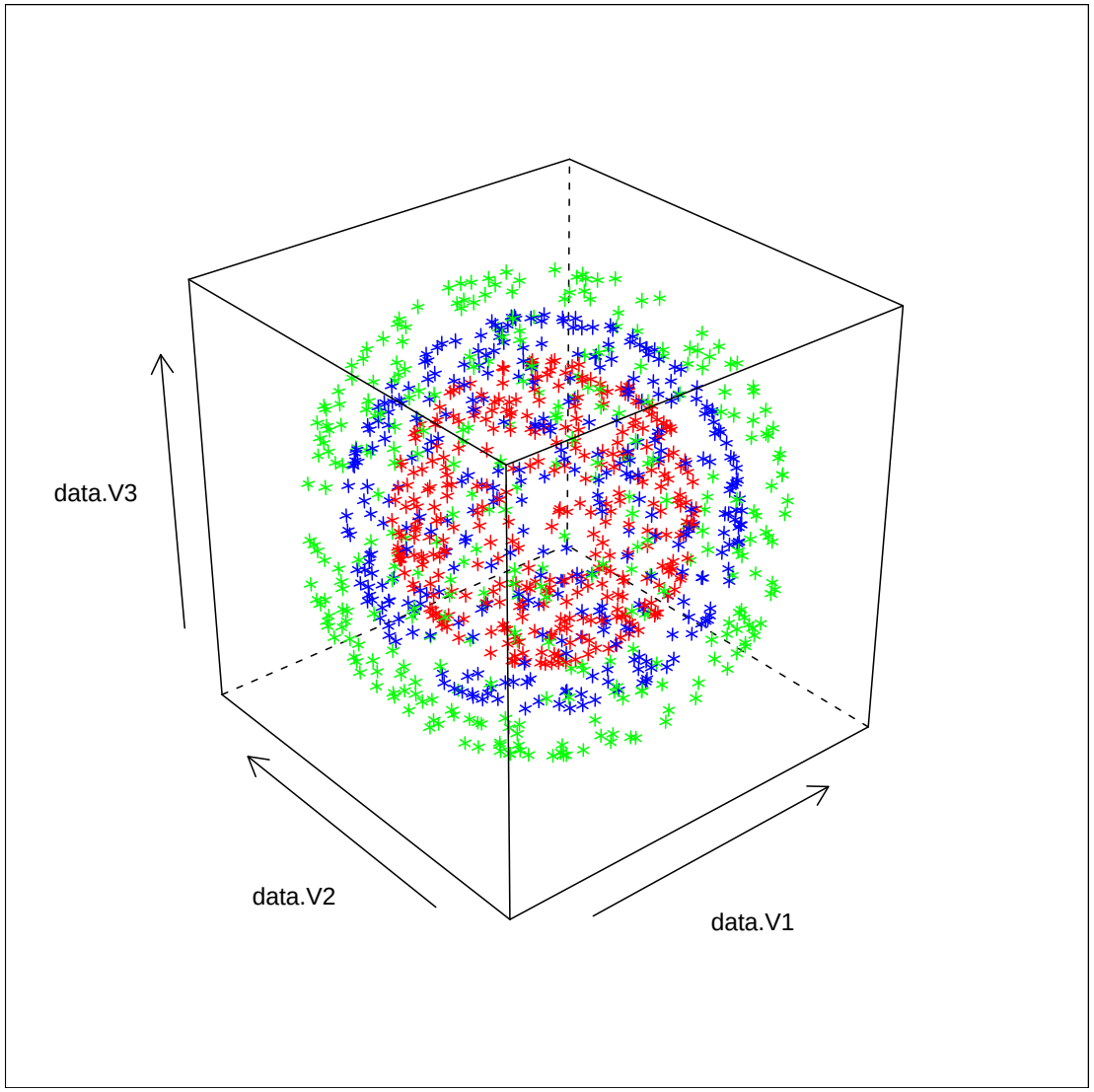
Linear projections



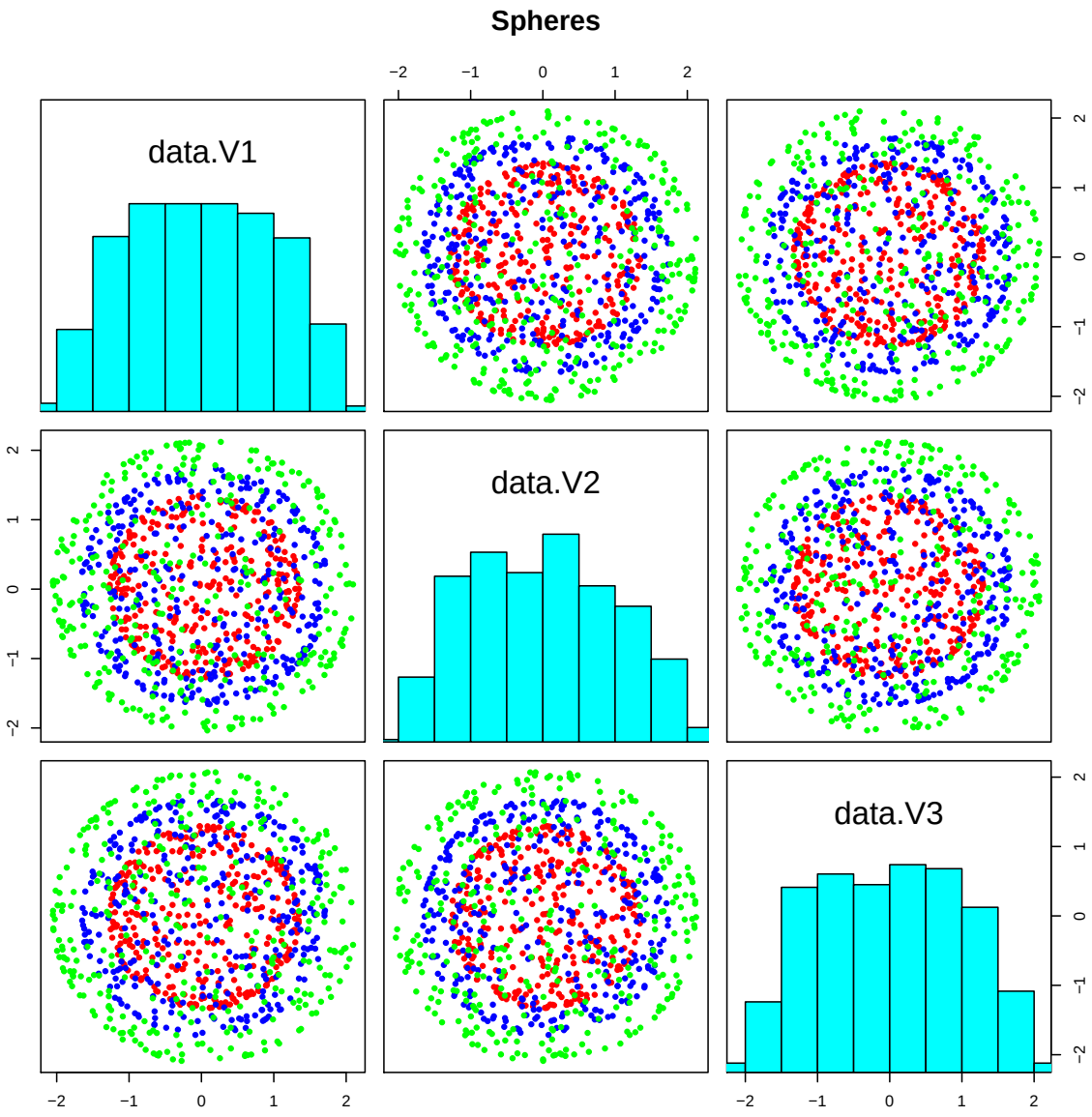
Linear projections



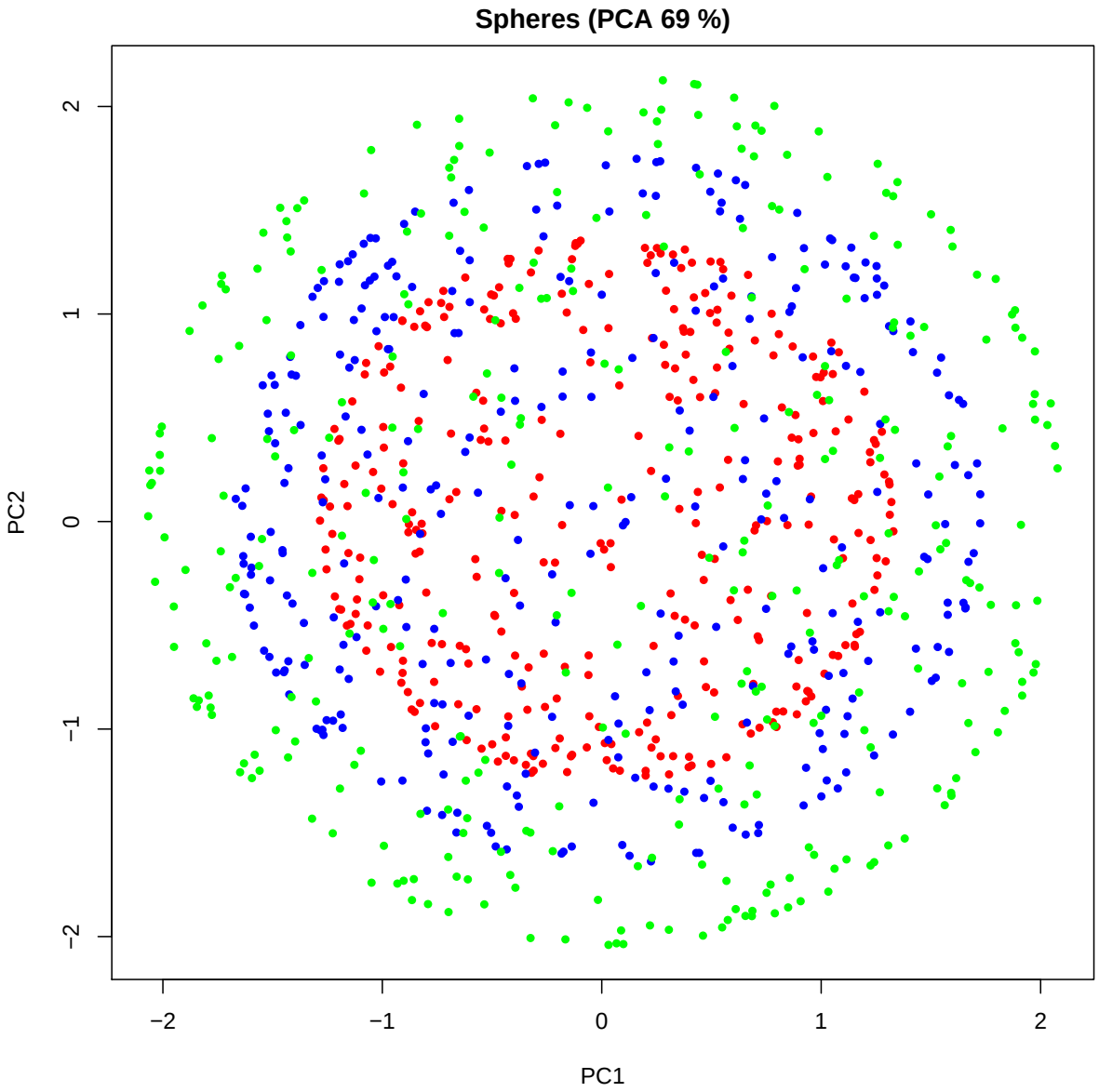
Linear projections



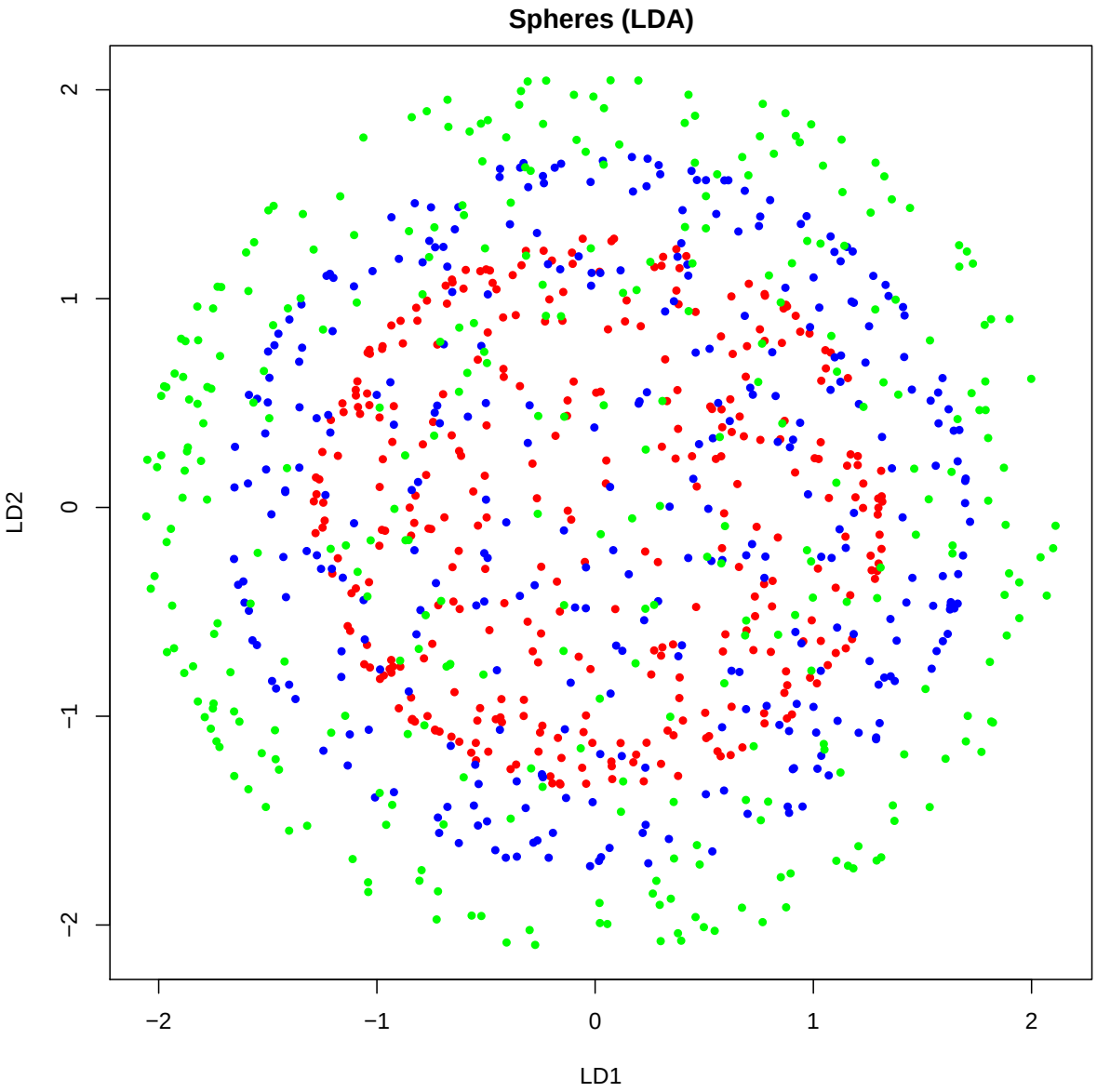
Linear projections



Linear projections



Linear projections



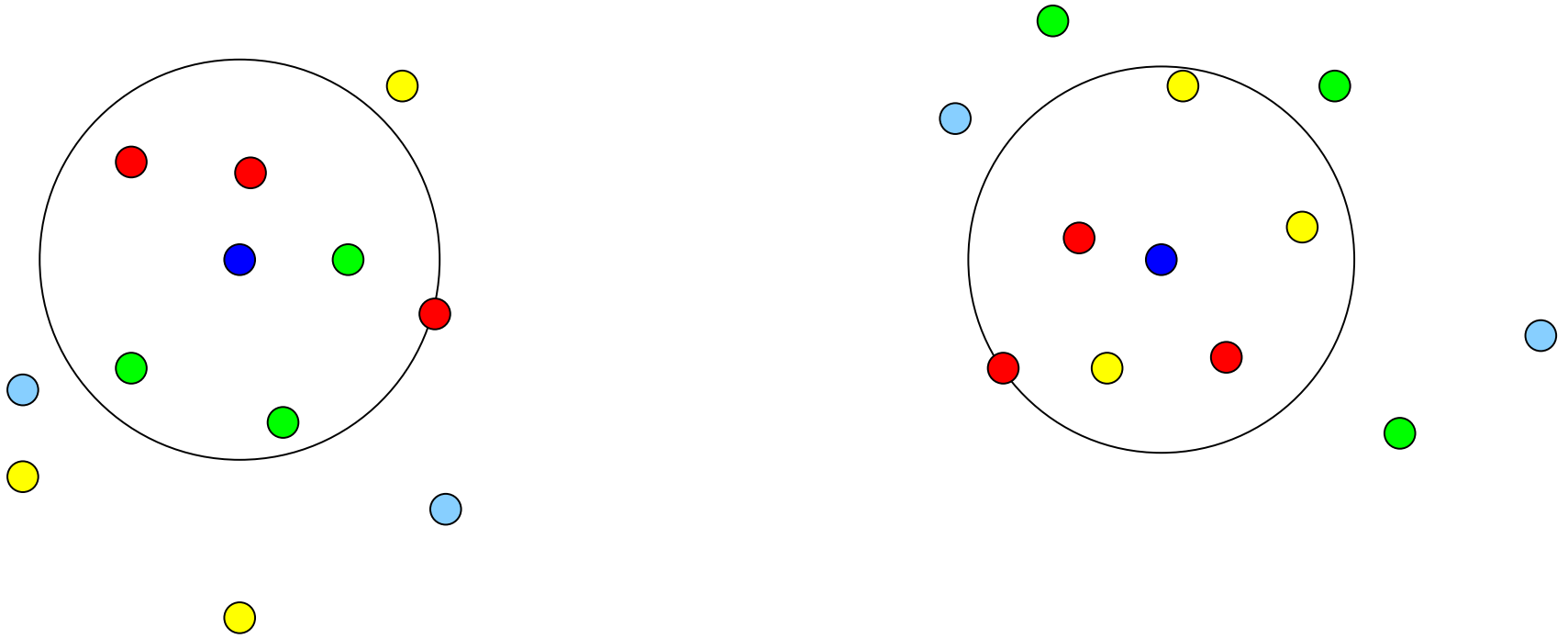
Neighborhood structure preservation

To enable visual cluster analysis on projected dataset, the neighborhood structure of the data must be preserved by the projection.

- Quality measures (Venna & Kaski, 2001 $\rightarrow$ 2007):
  - Trustworthiness: neighbors on the screen are real neighbors
  - Continuity: real neighbors are neighbors on the screen
  - Precision $\simeq$ Trustworthiness
  - Recall $\simeq$ Continuity
- Optimization methods:
  - Stochastic Neighbor Embedding (Hinton & Roweis, 2002)
  - Neighbor Retrieval Visualizer (Venna & Kaski, 2006)

Neighbor Preservation

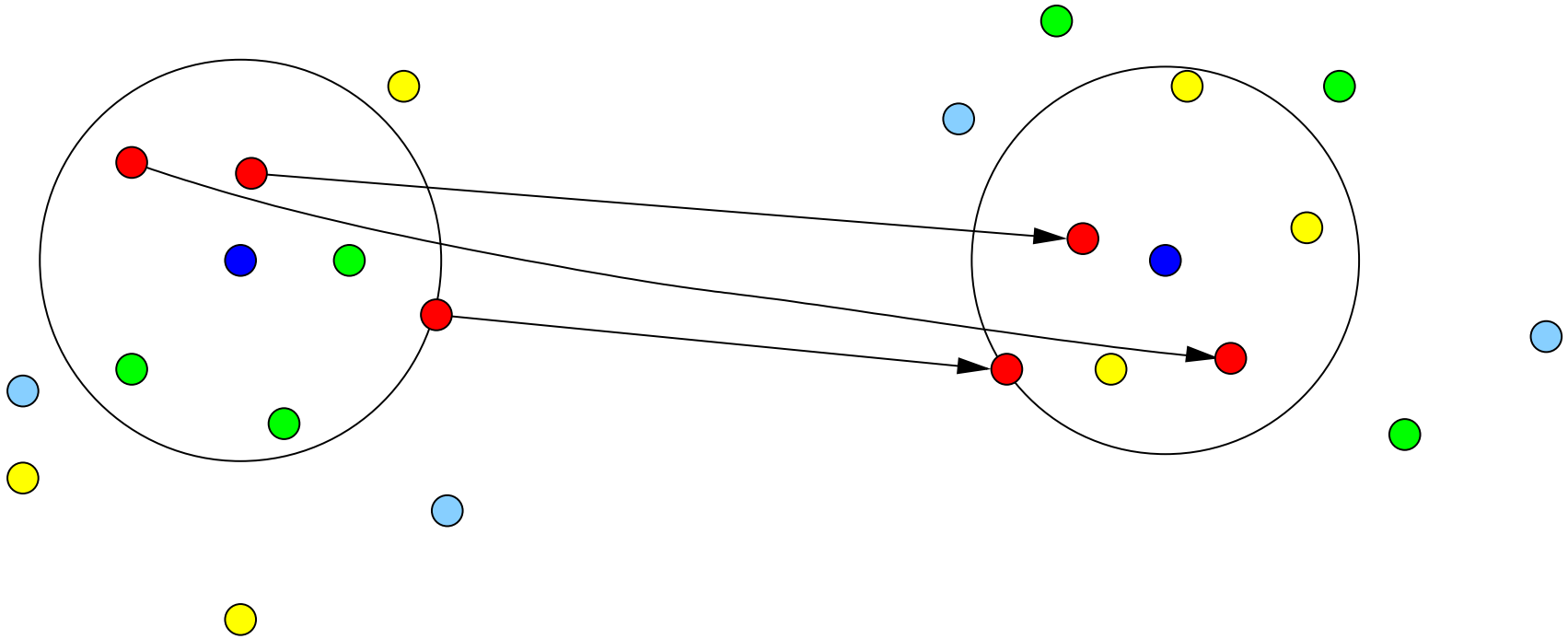
Original space to projection space



6 neighbors

Neighbor Preservation

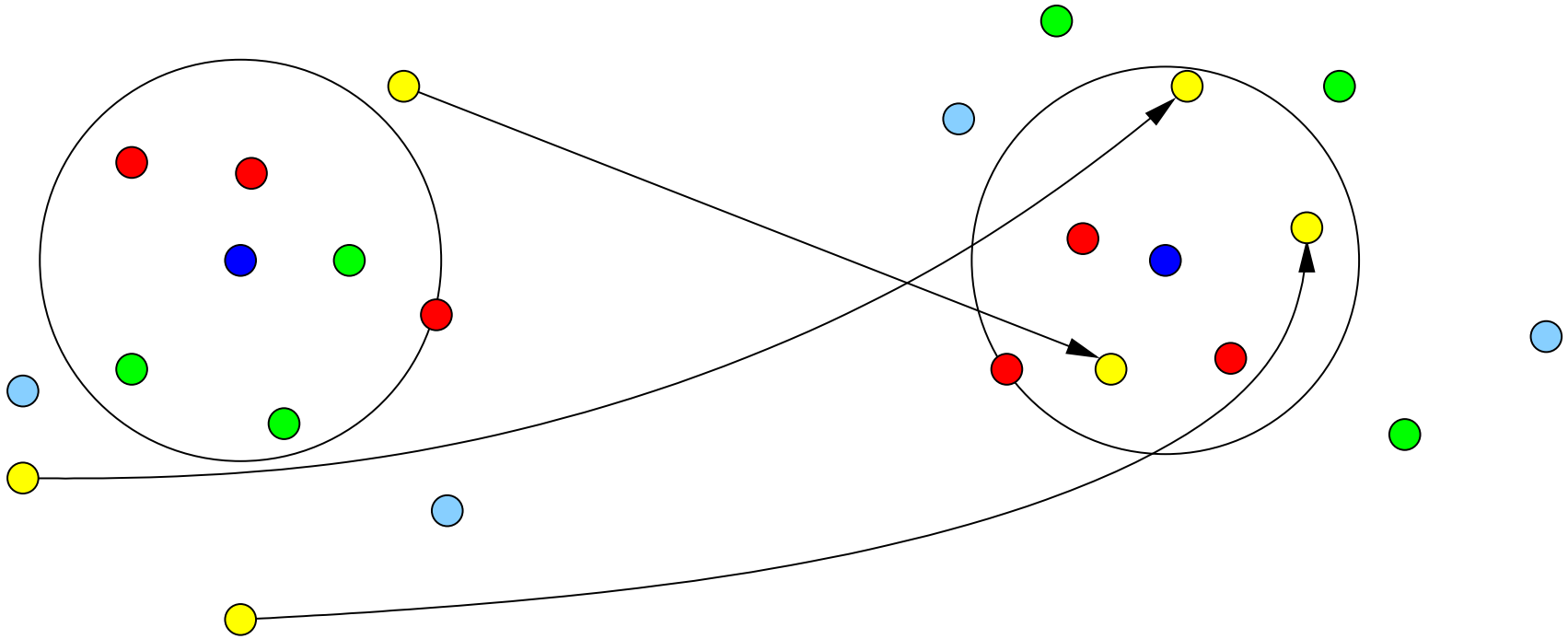
Original space to projection space



Correct projection

Neighbor Preservation

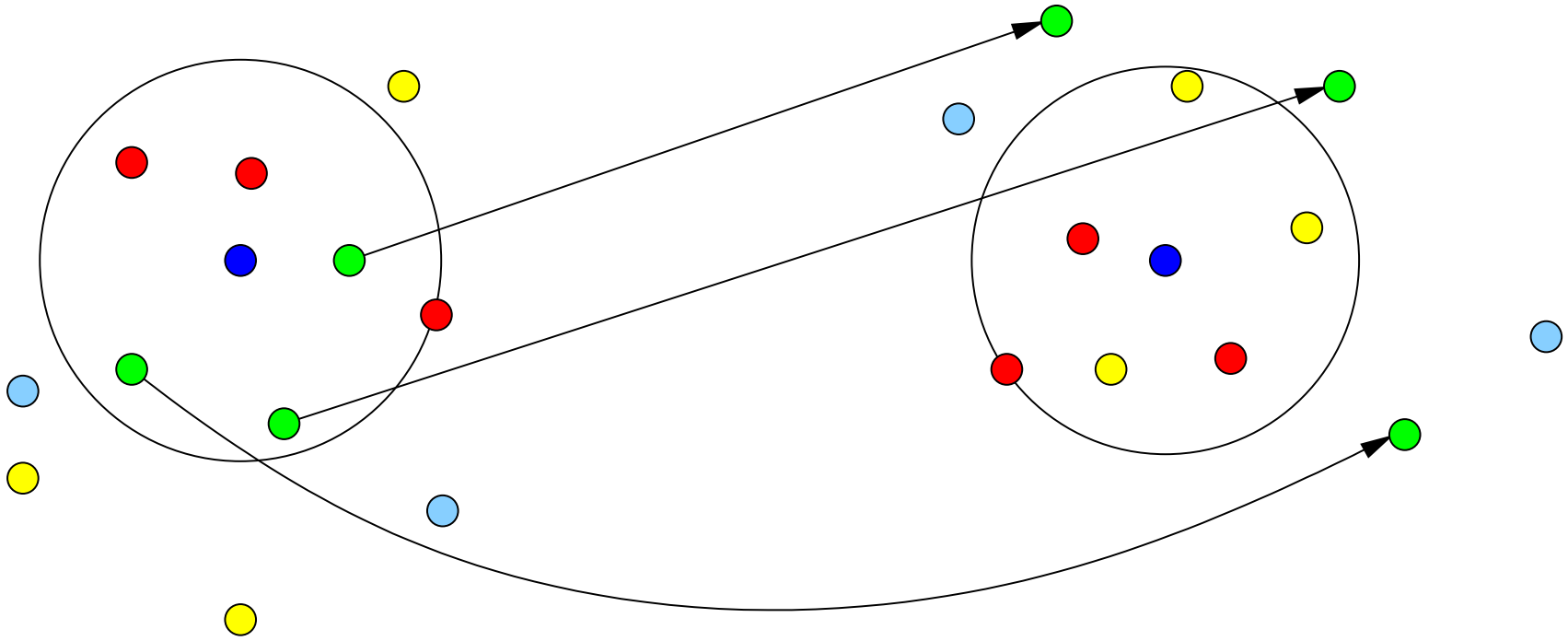
Original space to projection space



Trustworthiness violation

Neighbor Preservation

Original space to projection space



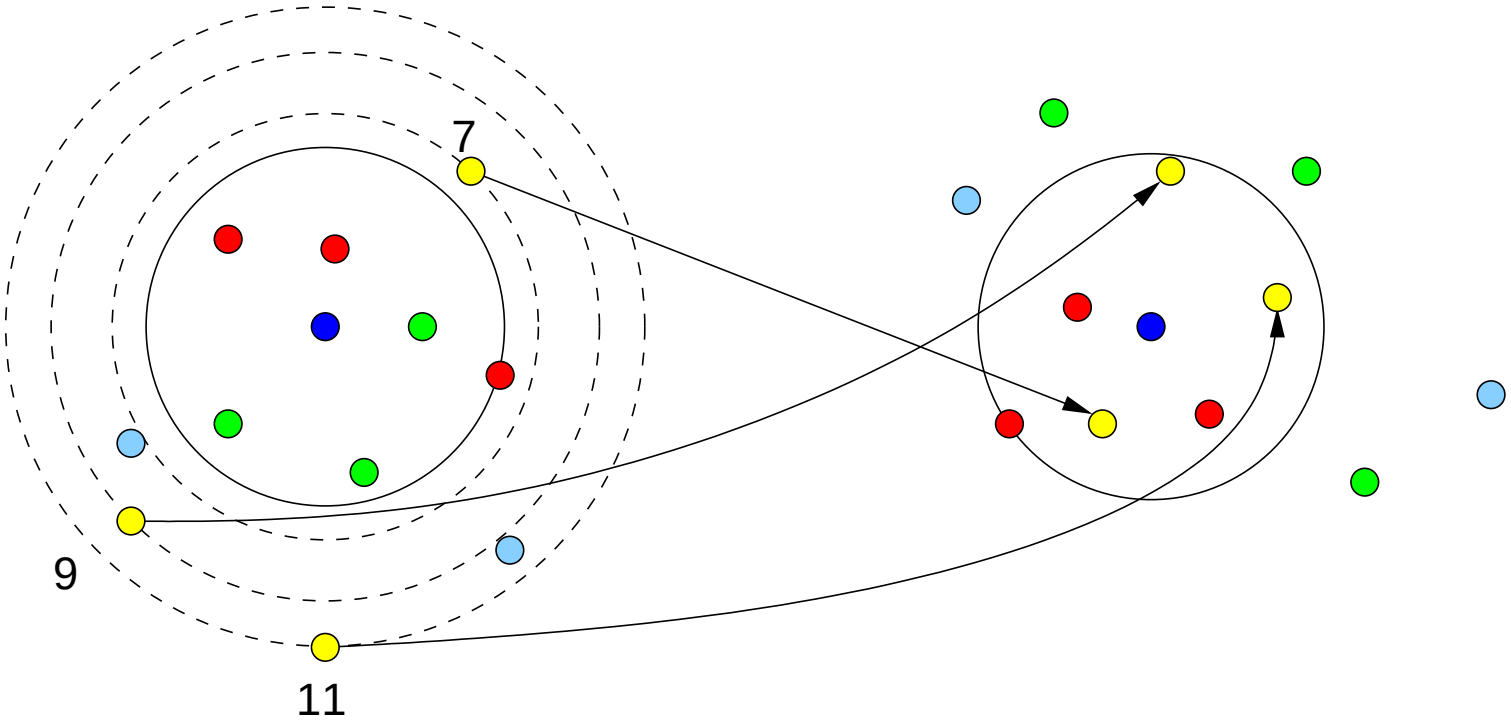
Continuity violation

Trustworthiness and Precision

Can you trust neighbors in the projection space?

- $O_k(x_i)$: k-nn of x_i in the original space
- $P_k(x_i)$: k-nn of x_i in the projection space
- $U_k(x_i) = P_k(x_i) \setminus O_k(x_i)$
- Precision
  - maximal precision: $P_k(x_i) \subset O_k(x_i)$
  - mean on i of $1 - \dfrac{\#U_k(x_i)}{\#P_k(x_i)}$
- Trustworthiness
  - rank preservation: $r^O(x_j, x_i)$ rank of x_j as a neighbor of x_i in the original space
  - $M_1(k) = 1 - \dfrac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{x_j \in U_k(x_i)} \left(r^O(x_j, x_i) - k \right)$

Trustworthiness and Precision



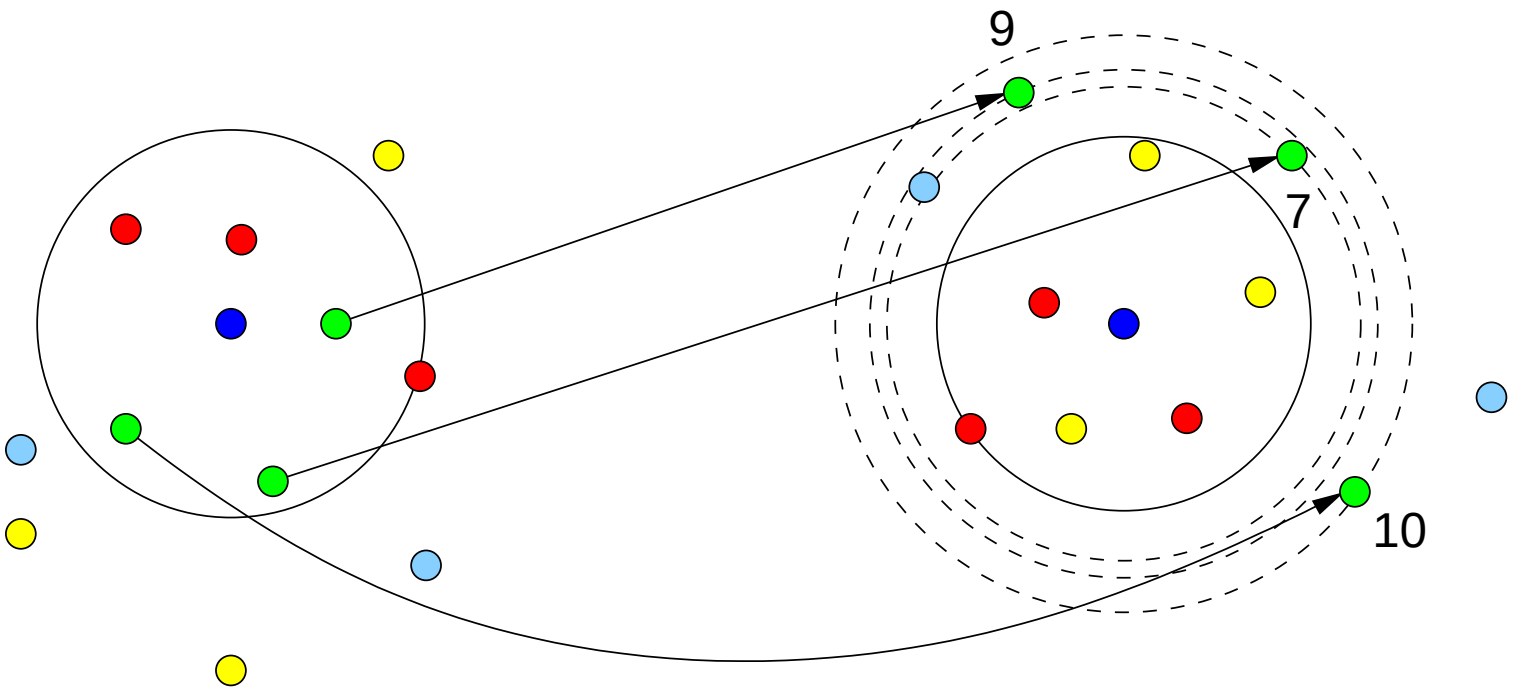
- Contribution to precision: 0.5
- Contribution to trustworthiness: 9

Continuity and Recall

Do you miss neighbors in the projection space?

- $O_k(x_i)$: k-nn of x_i in the original space
- $P_k(x_i)$: k-nn of x_i in the projection space
- $V_k(x_i) = O_k(x_i) \setminus P_k(x_i)$
- Recall
  - maximal recall: $O_k(x_i) \subset P_k(x_i)$
  - mean on i of $1 - \dfrac{\# V_k(x_i)}{\# O_k(x_i)}$
- Continuity
  - rank preservation: $r^P(x_j, x_i)$ rank of x_j as a neighbor of x_i in the projection space
  - $M_2(k) = 1 - \dfrac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{x_j \in V_k(x_i)} \left(r^P(x_j, x_i) - k\right)$

Continuity and Recall



- Contribution to recall: 0.5
- Contribution to continuity: 8

Stochastic Neighbor Embedding (SNE)

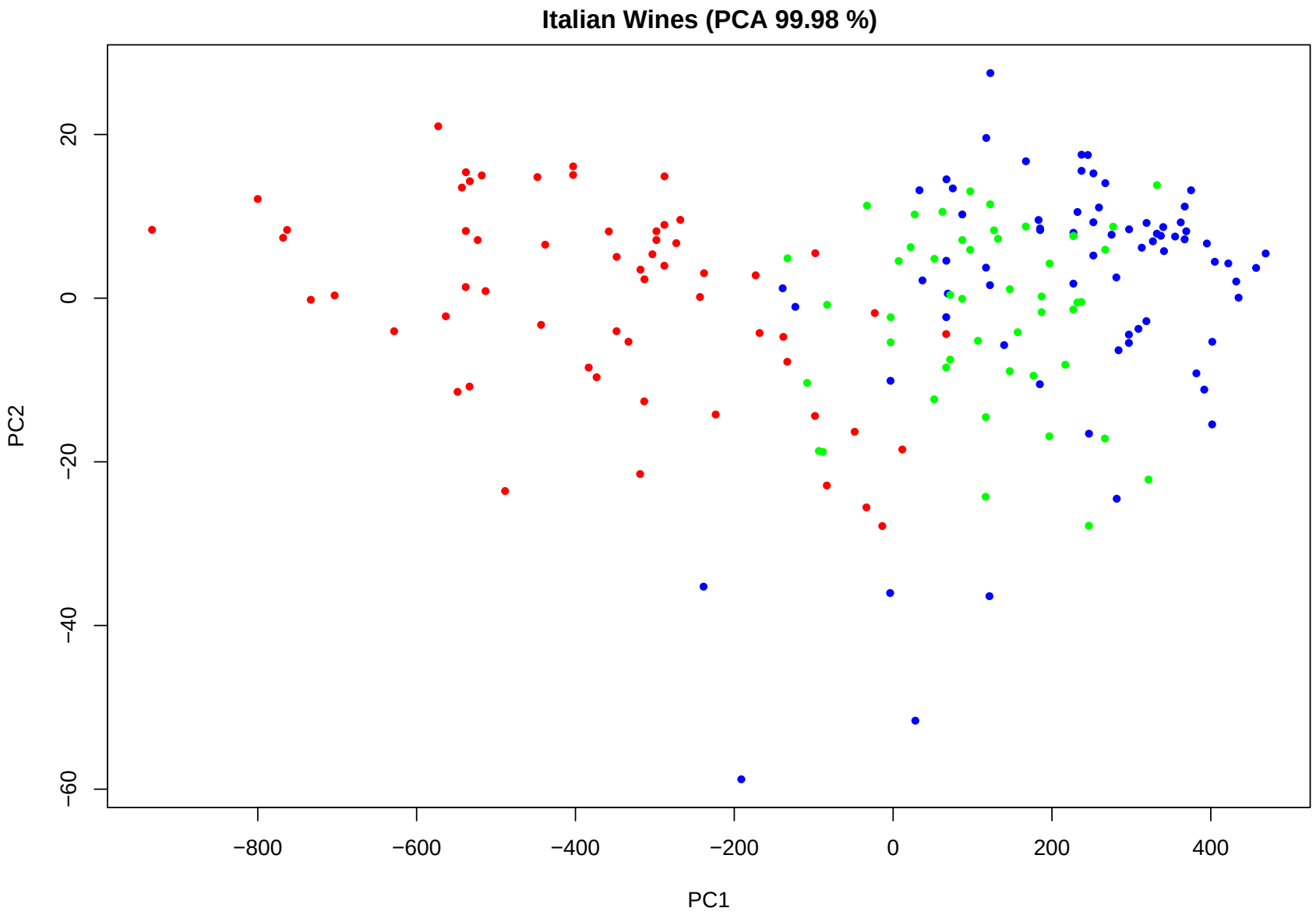
Approximating the "neighborhood distribution" (Hinton & Roweis, 2002)

- Original space: $p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq j} \exp(-d_{ik}^2)}$ (d_{ij} dissimilarity between x_i and x_j)
- x_i projected to y_i
- Projection space: $q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq j} \exp(-\|y_i - y_k\|^2)}$
- Minimize the Kullback-Leibler divergence between $p_{i.}$ and $q_{i.}$, i.e.,

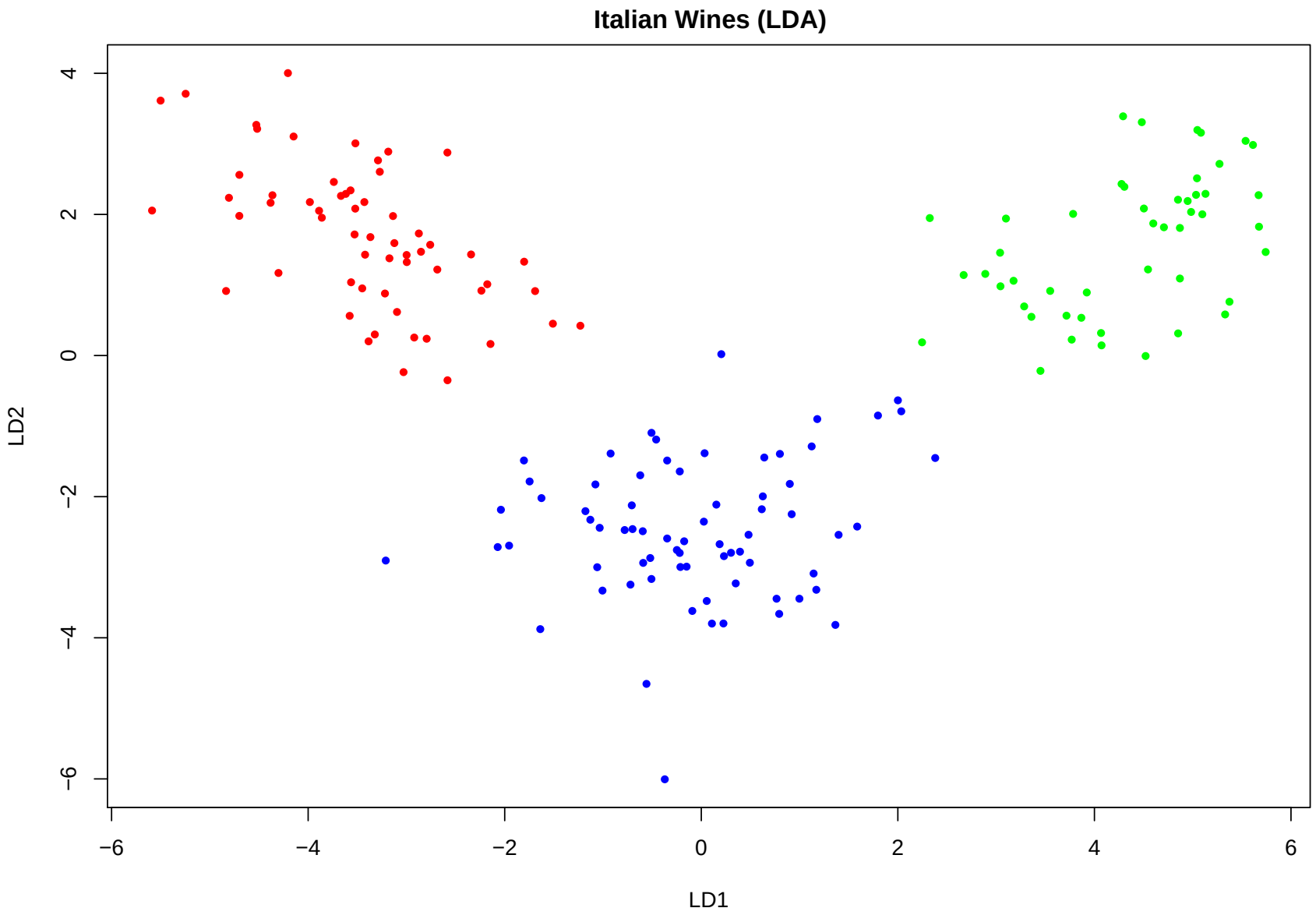
$$C = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- Corresponds to a smoothed version of the recall (Venna & Kaski, 2006)

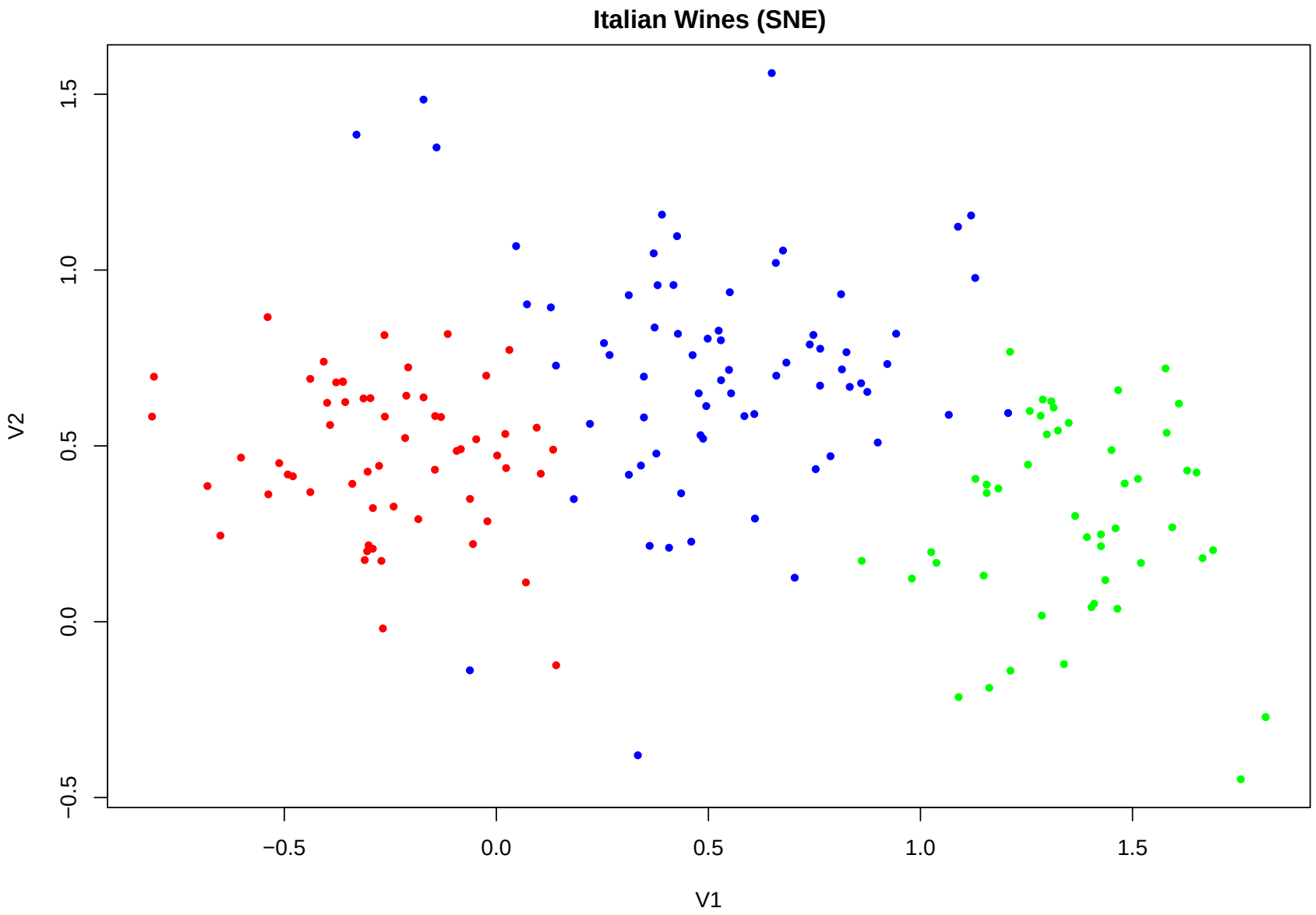
An example


Italian Wines (PCA 99.98 %)
PC2
20
0
−20
−40
−60
−800
−600
−400
−200
0
200
400
PC1

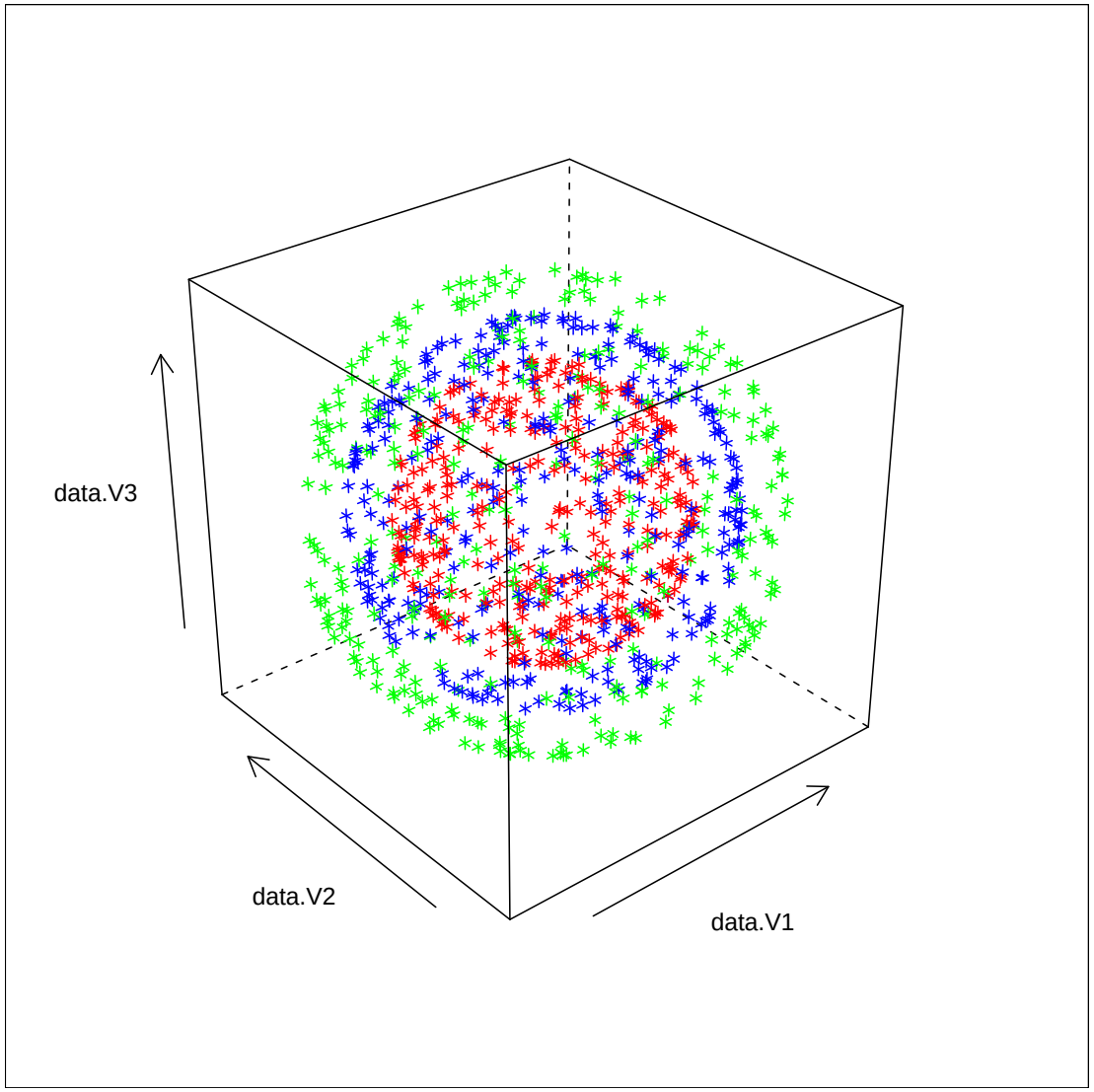
An example



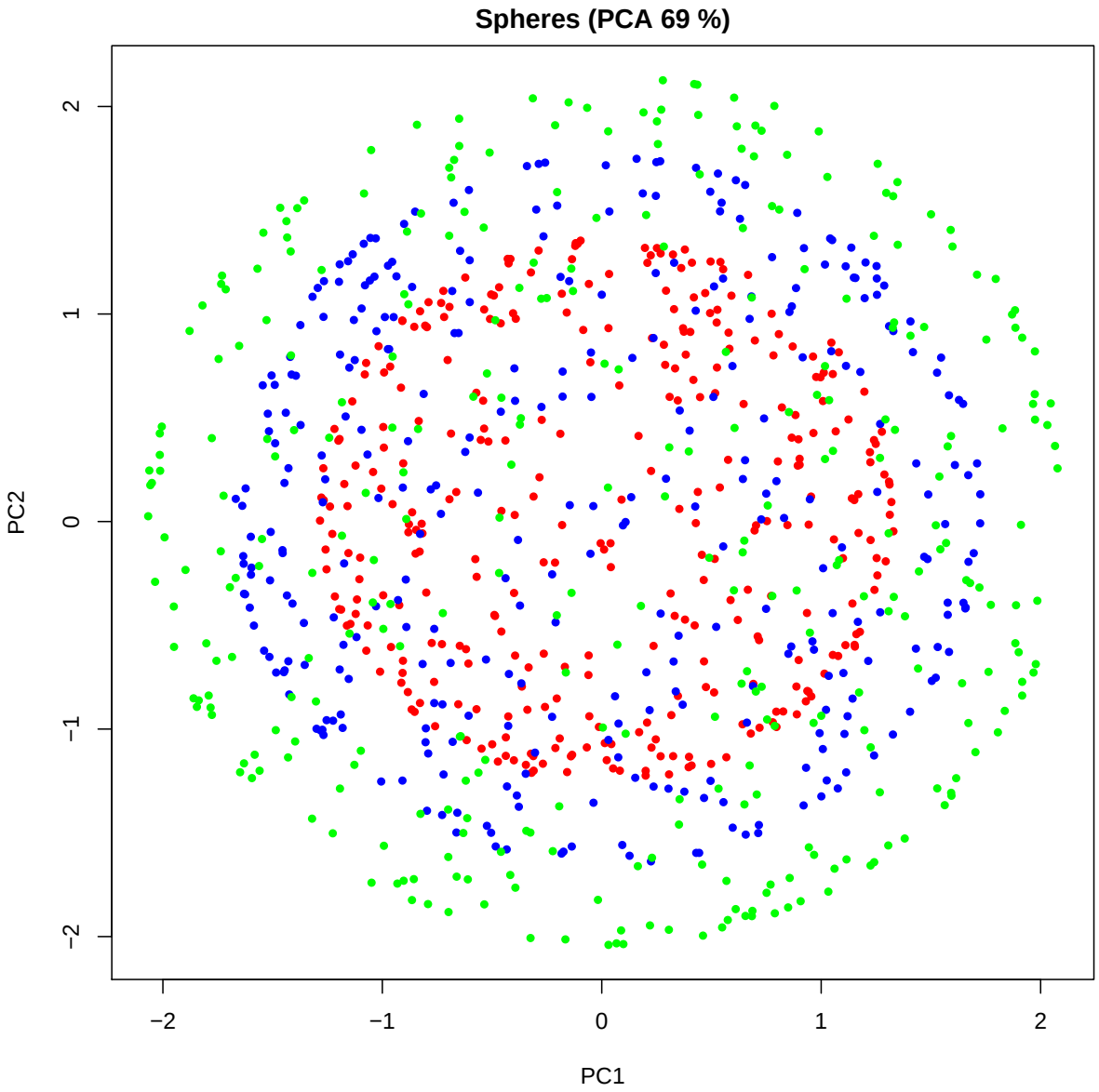
An example



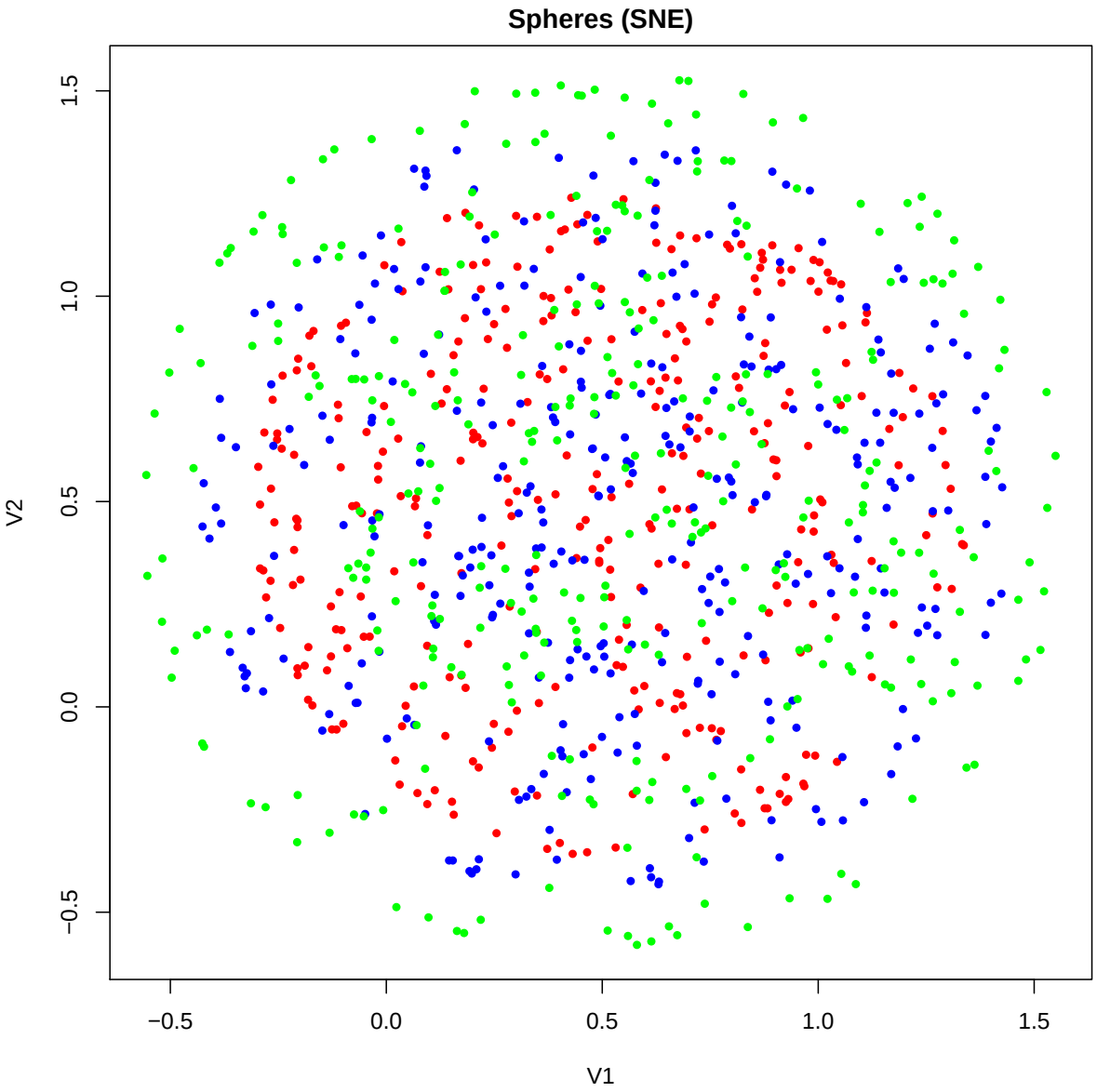
No miracle



No miracle



No miracle



Neighbor Retrieval Visualizer

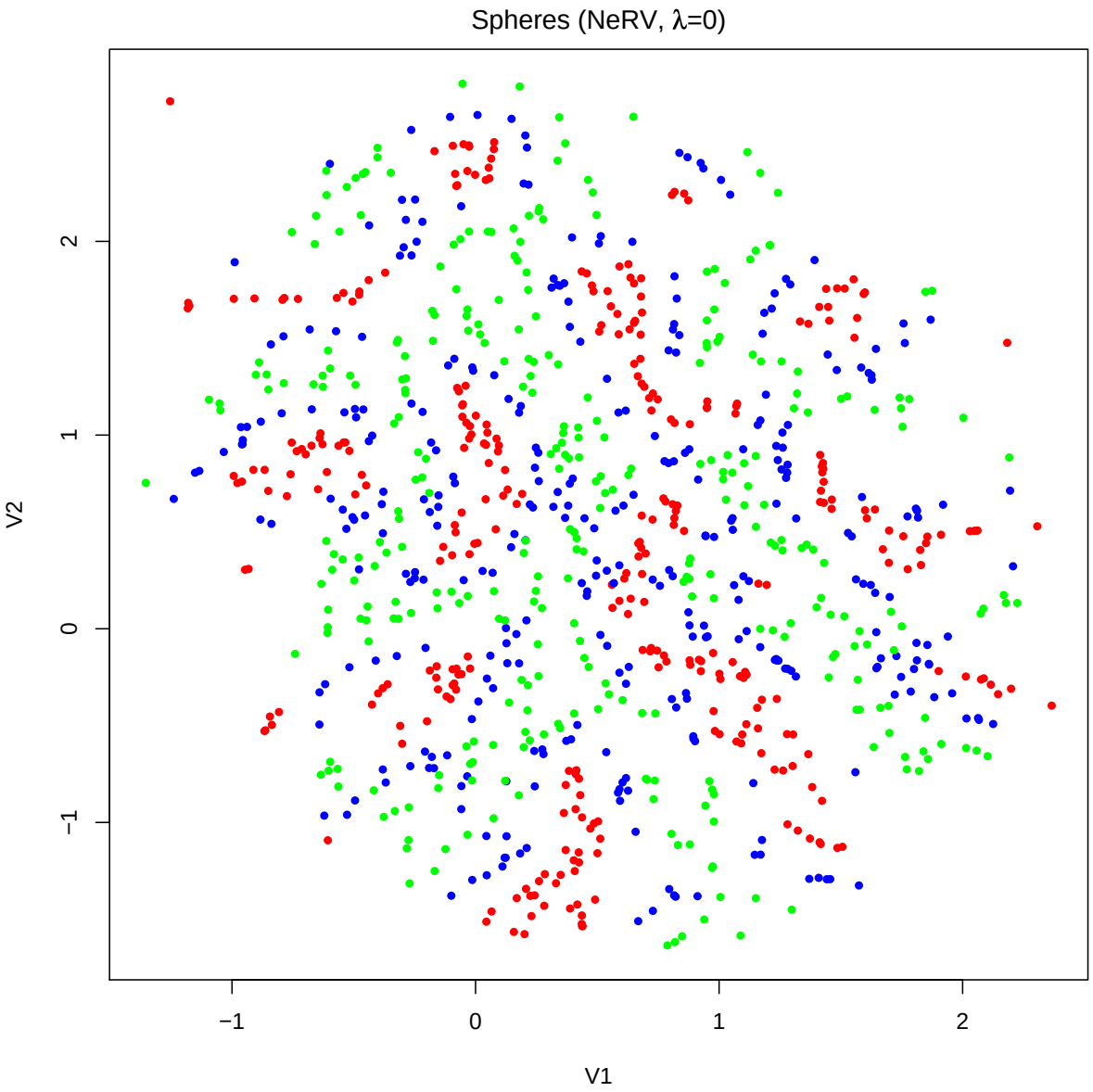
Optimizing smoothed versions of precision and recall (Venna & Kaski, 2006)

- inspired by stochastic neighbor embedding (SNE)
- minimization of

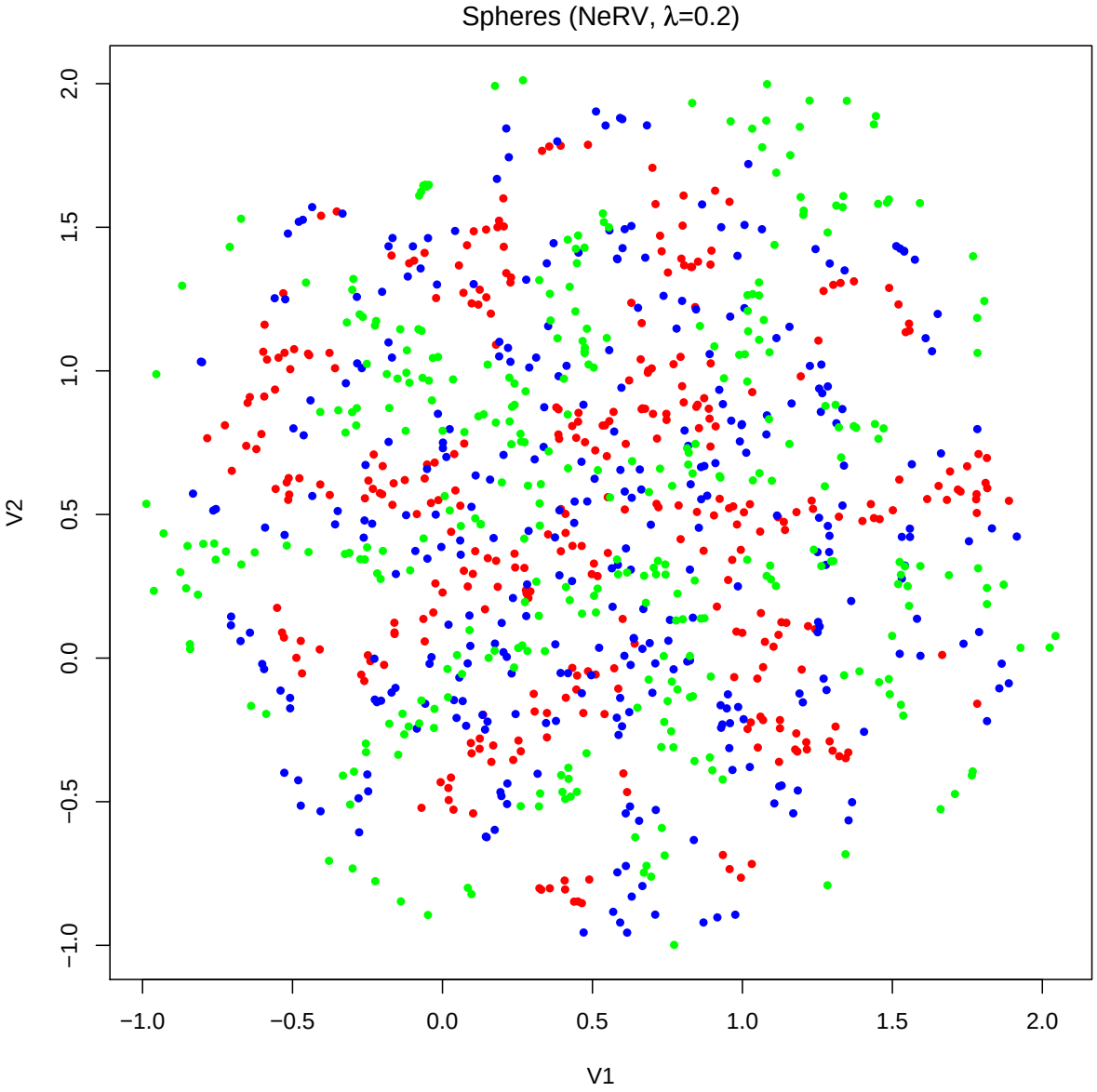
$$\lambda \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} + (1-\lambda) \sum_i \sum_j q_{ij} \log \frac{q_{ij}}{p_{ij}}$$

- $\lambda = 1$ (SNE) $\Rightarrow$ recall
- $\lambda = 0 \Rightarrow$ precision
- λ is user chosen (can be chosen to optimize a quality measure)
- slow ($O(N^3)$ per iteration for N objects, as SNE)

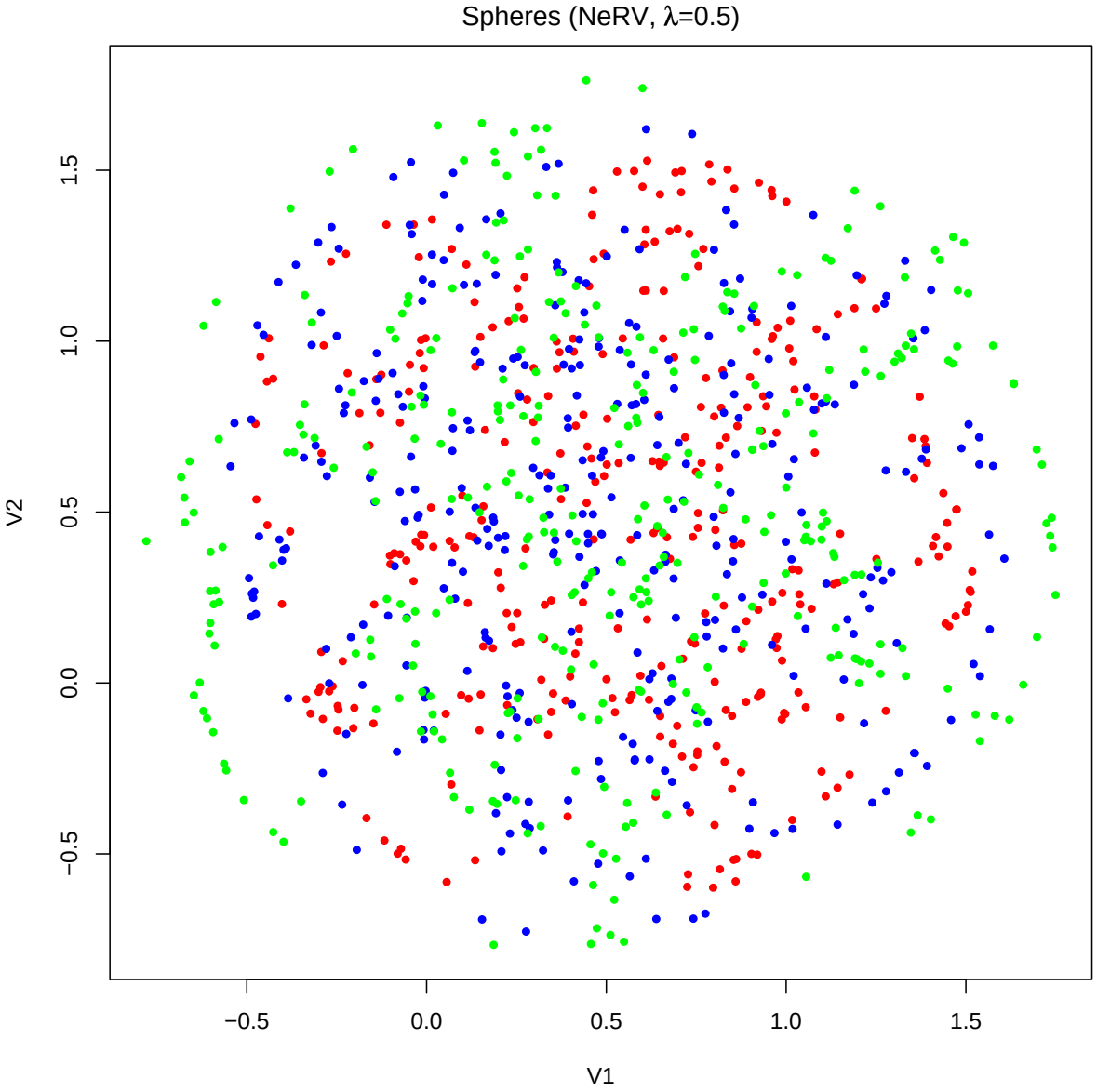
Neighbor Retrieval Visualizer (No class info!)



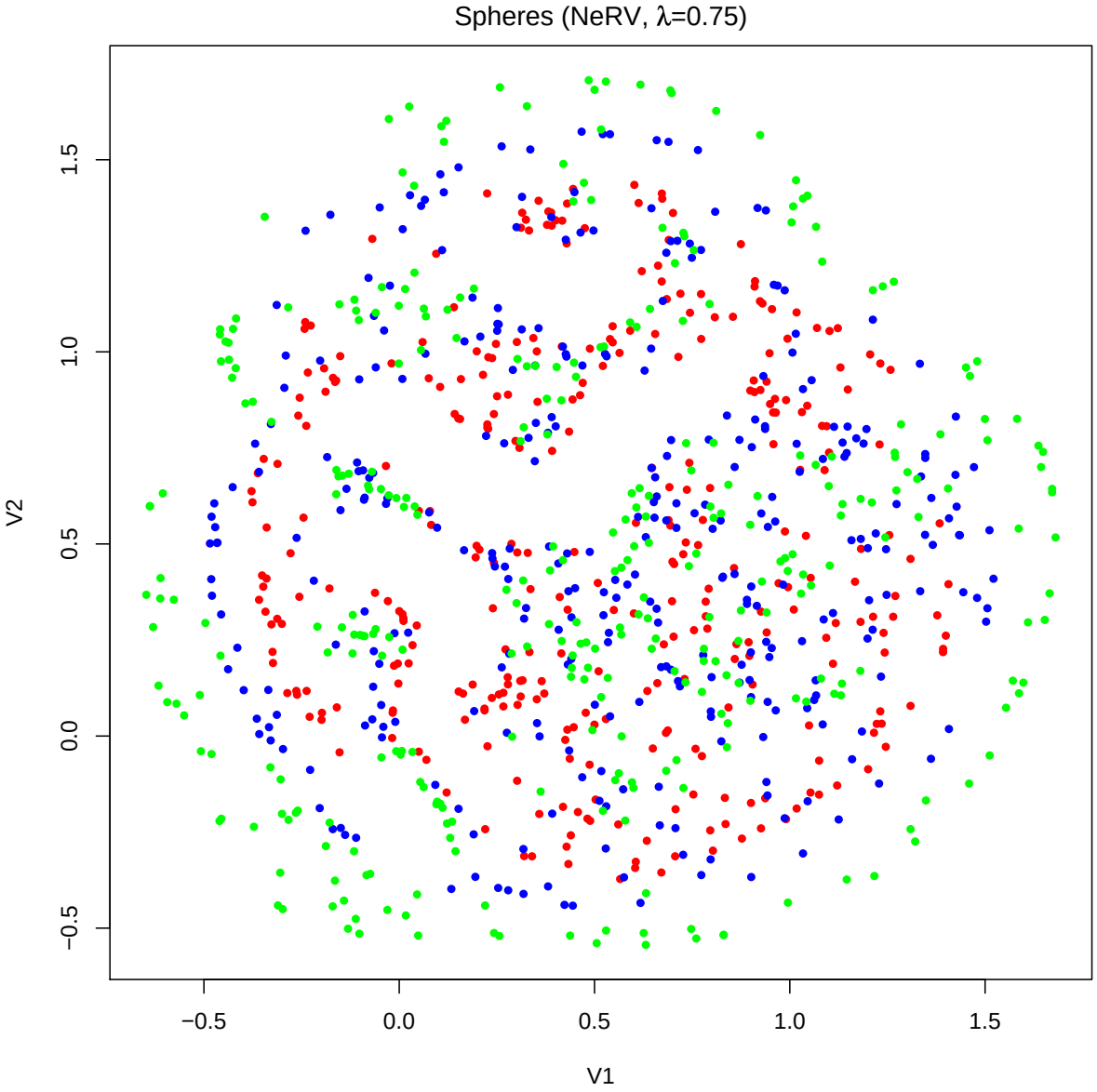
Neighbor Retrieval Visualizer (No class info!)



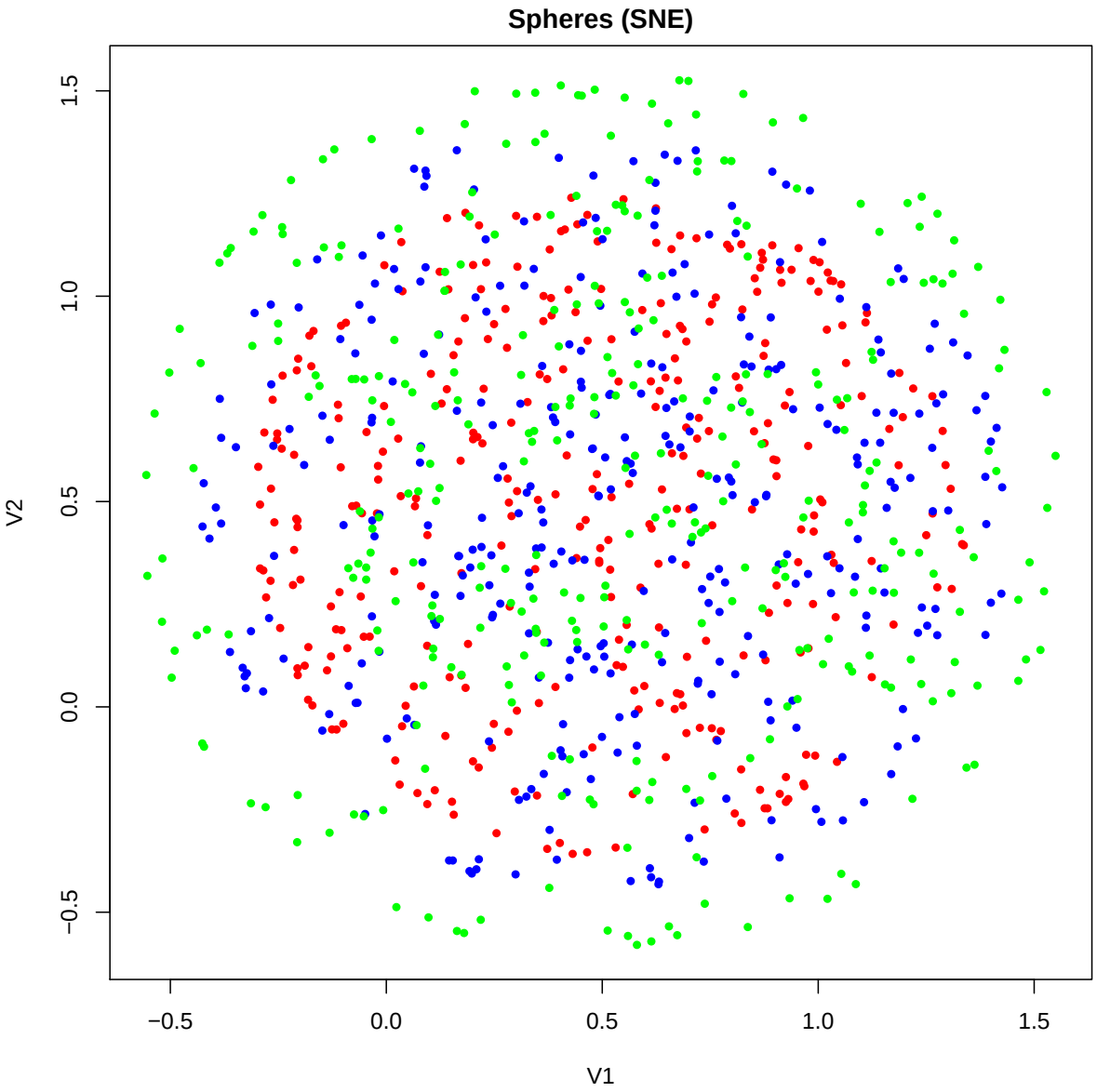
Neighbor Retrieval Visualizer (No class info!)



Neighbor Retrieval Visualizer (No class info!)



Neighbor Retrieval Visualizer (No class info!)



Outline

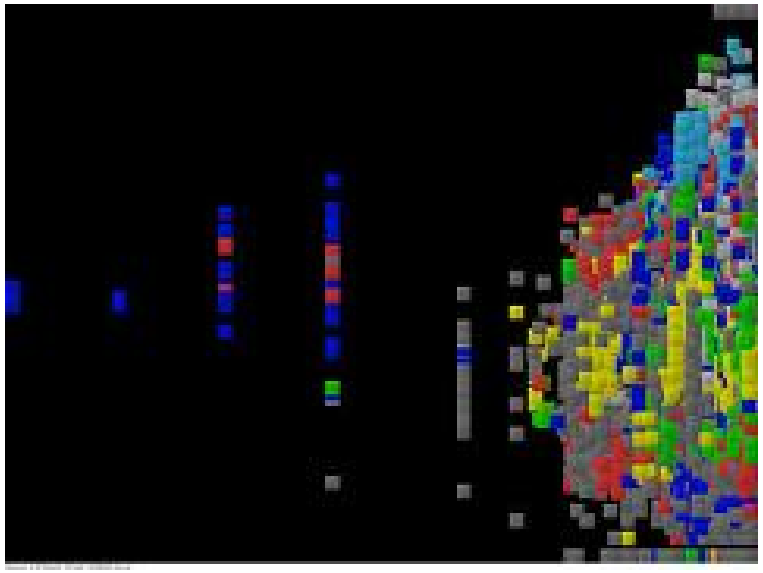
Reducing the overlapping problem

An open research topic.

- Rendering:
  - Transparency
  - stereo vision
  - overlap counting (via openGL!)
- Interactivity:
  - sub-sampling
  - zooming and panning
  - excentric labeling
  - magic lenses
  - distortion
- AI:
  - clustering
  - force field approach

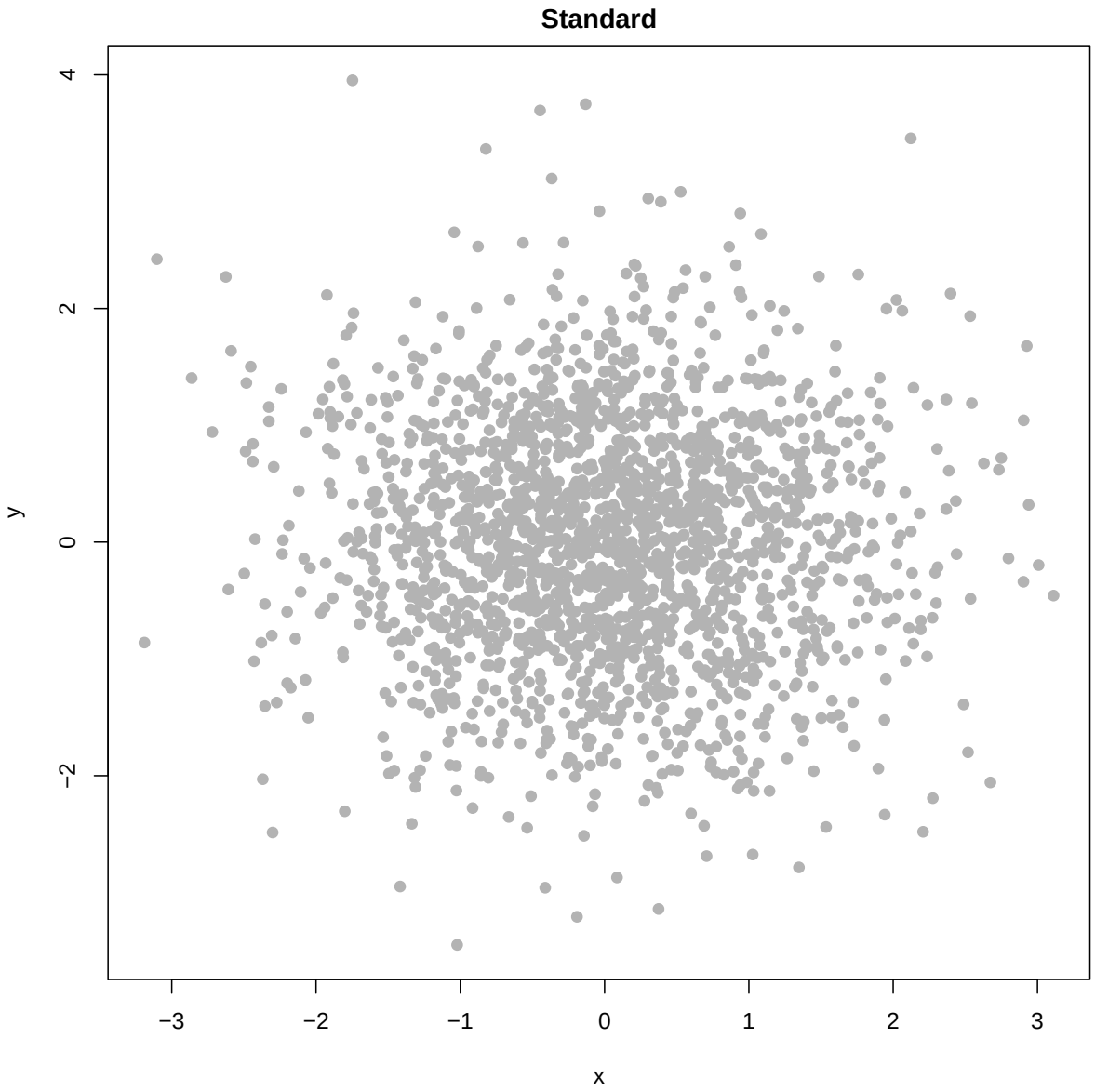
Transparency

- Dots are not fully opaque
- Overlapping produces:
  - Enhanced saturation for identical original colors
  - Blended colors for distinct original colors
- Similar idea: overlap counting (left) of the scatter plot (right)

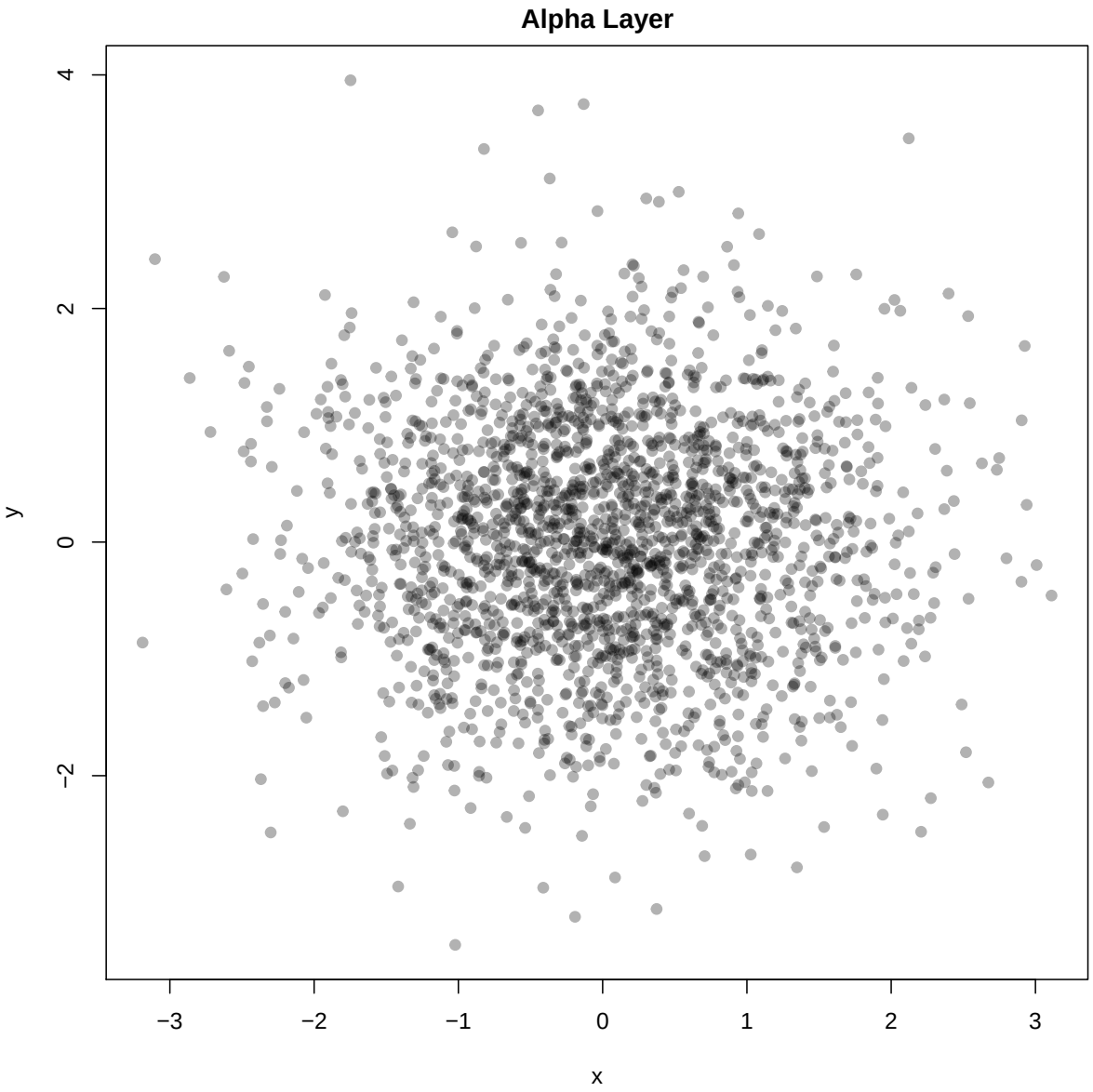


http://www.cs.umd.edu/hcil/millionvis/

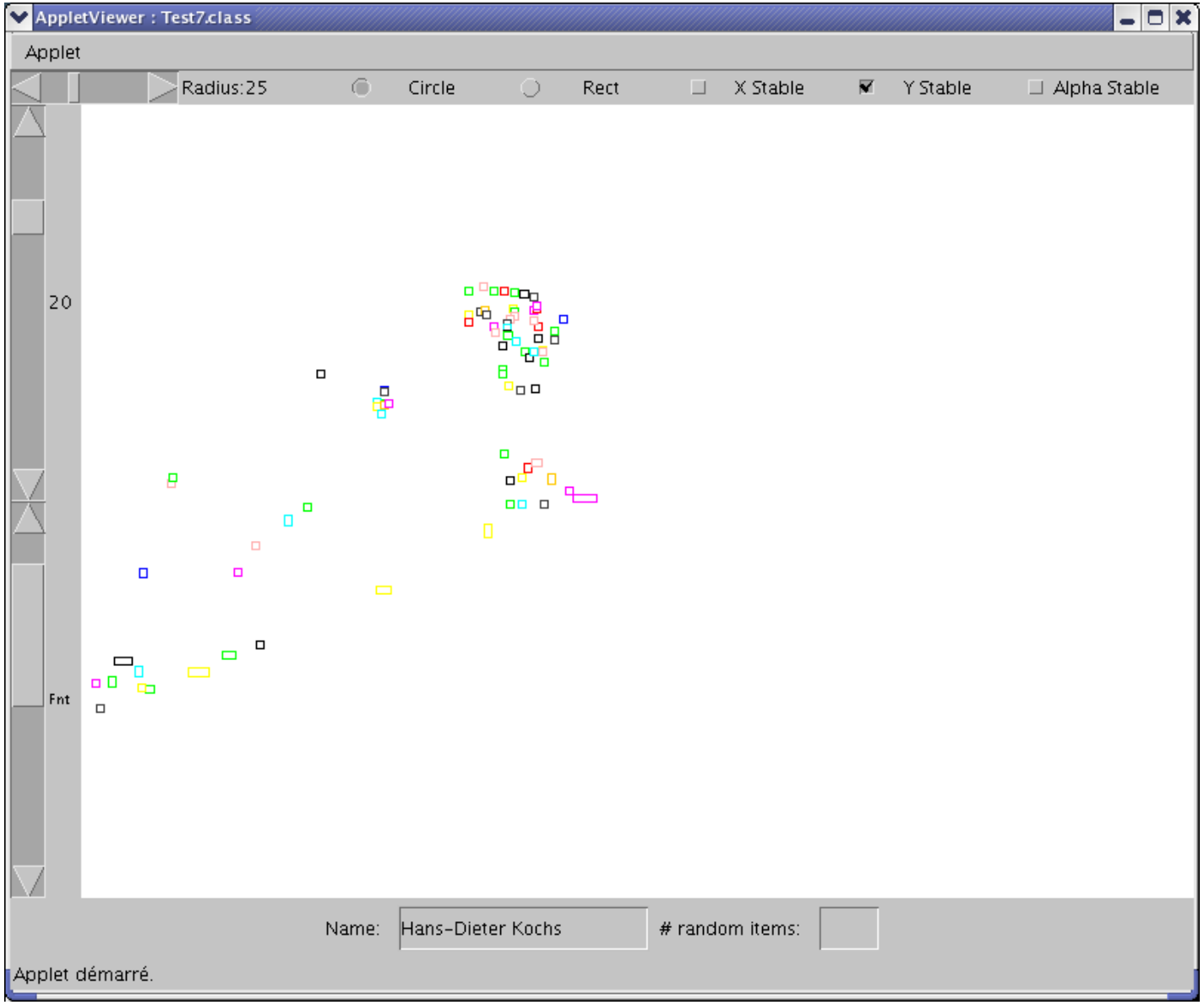
Transparency



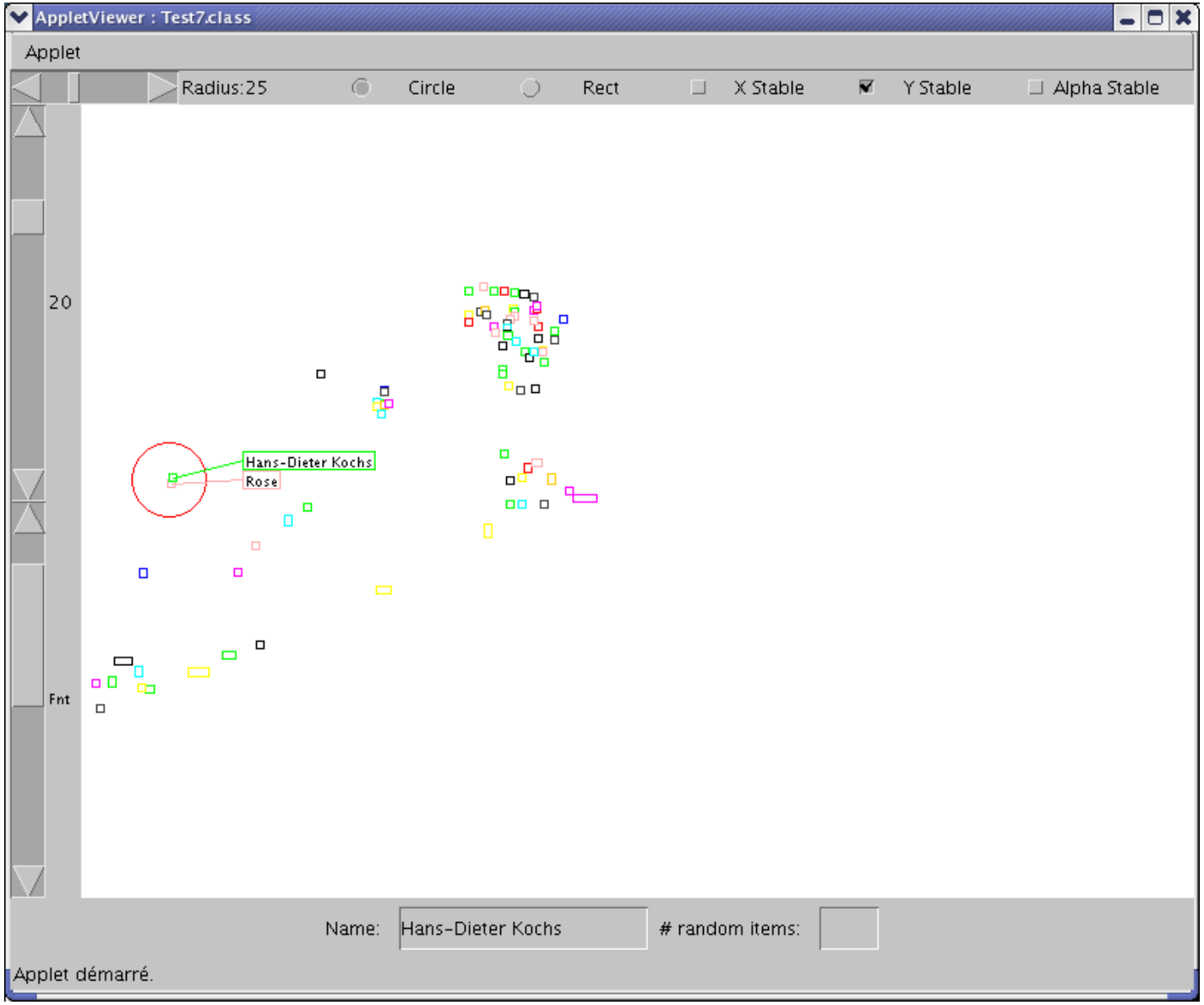
Transparency



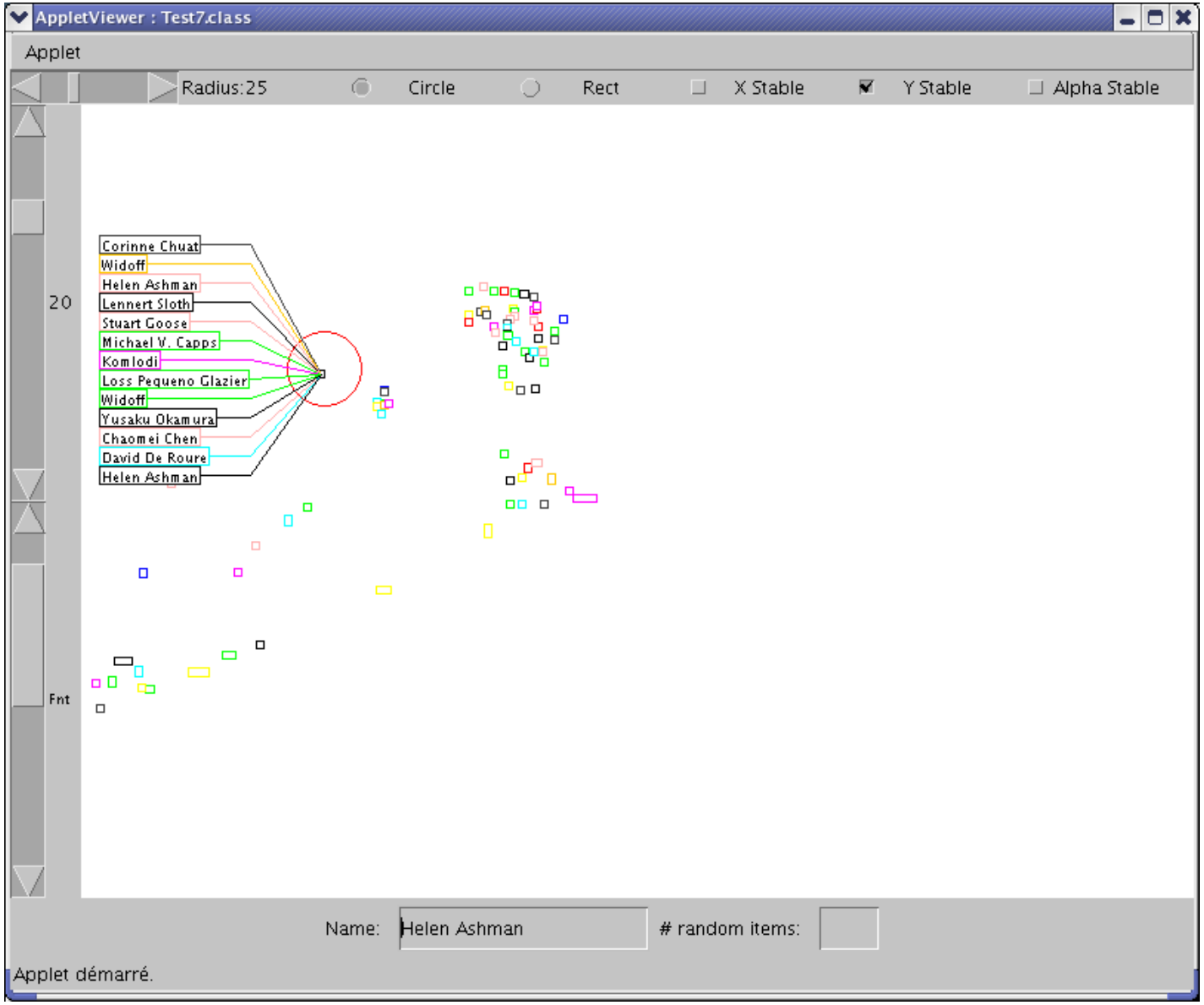
Excentric labeling



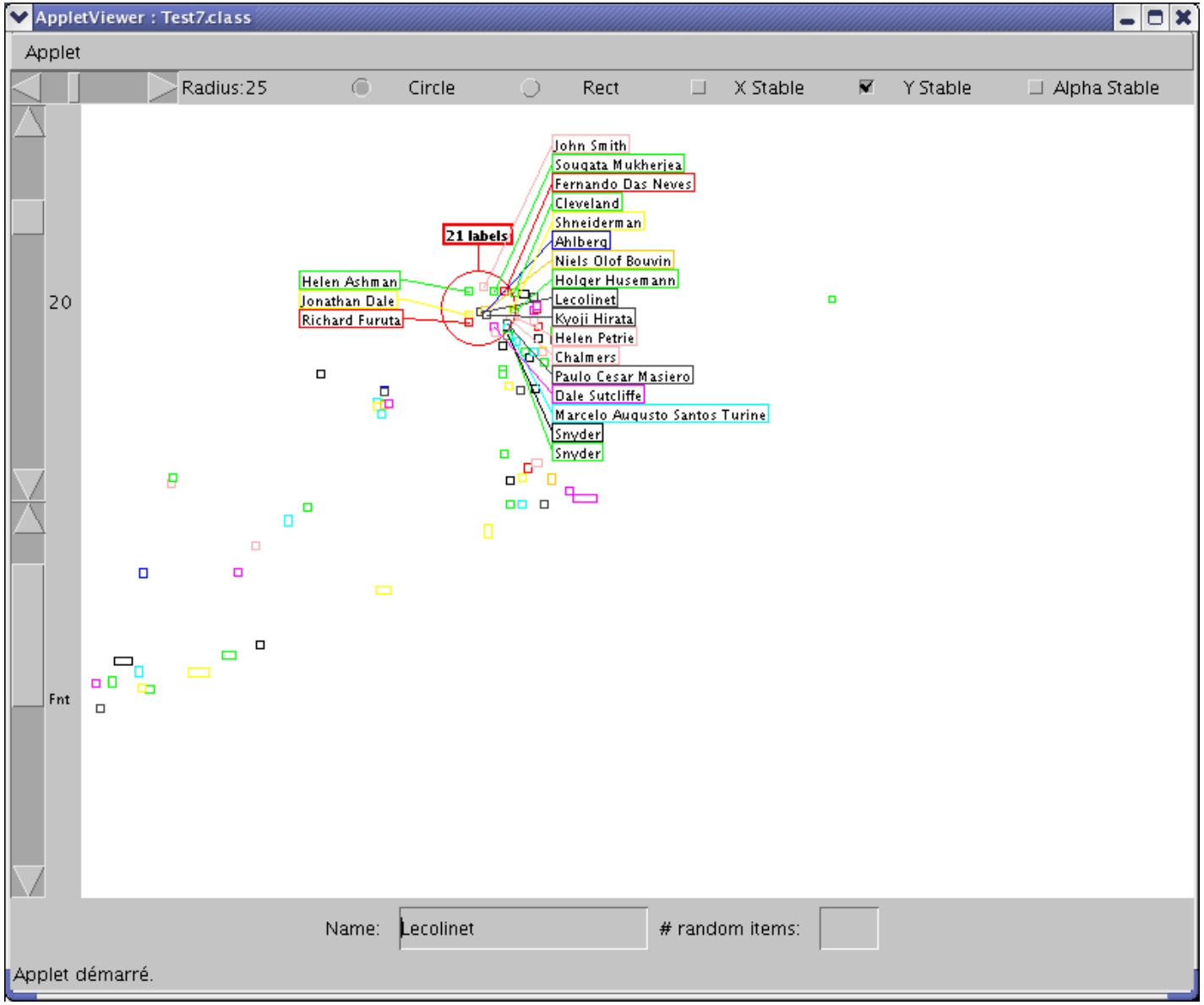
Excentric labeling



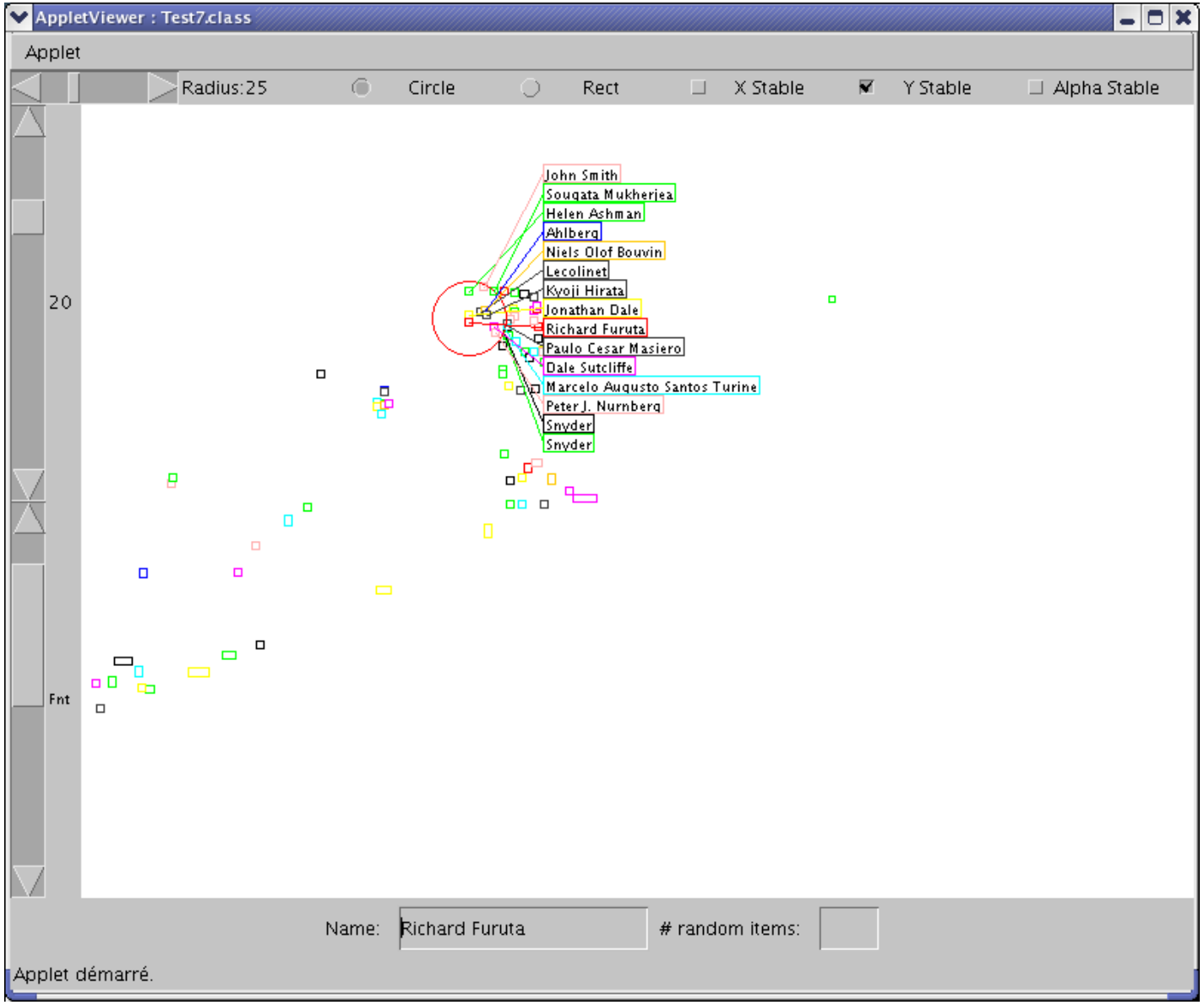
Excentric labeling



Excentric labeling



Excentric labeling

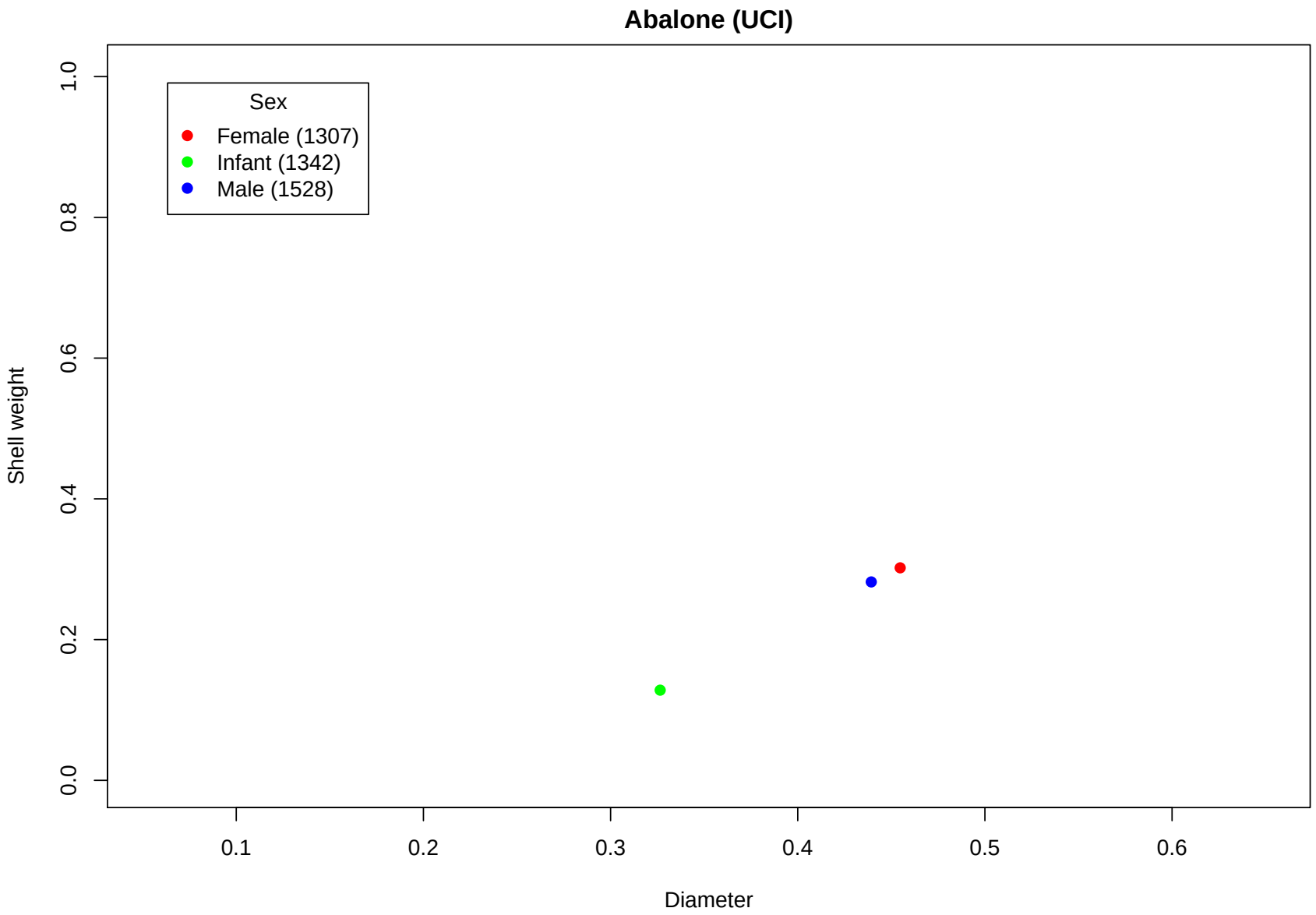


Clustering

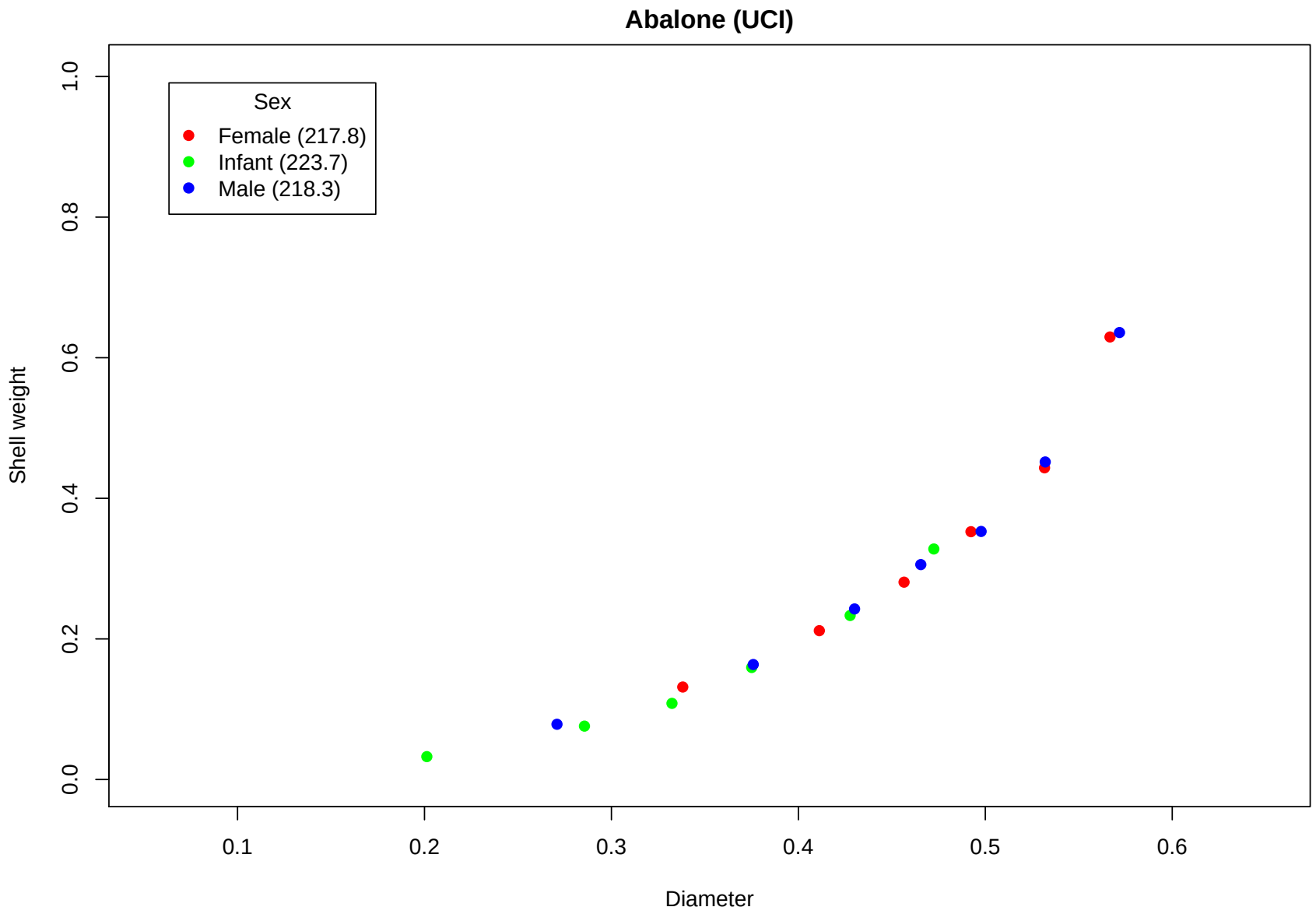
Reduction of the number of objects:

- Very general solution (not limited to scatter plot)
- Clustering $\Rightarrow$ Cluster prototype $\Rightarrow$ Prototype visualization
- Interactivity:
  - Number of clusters:
    - deterministic clustering
    - least disturbance (e.g., hierarchical)
  - Cluster extension (objects or summary)

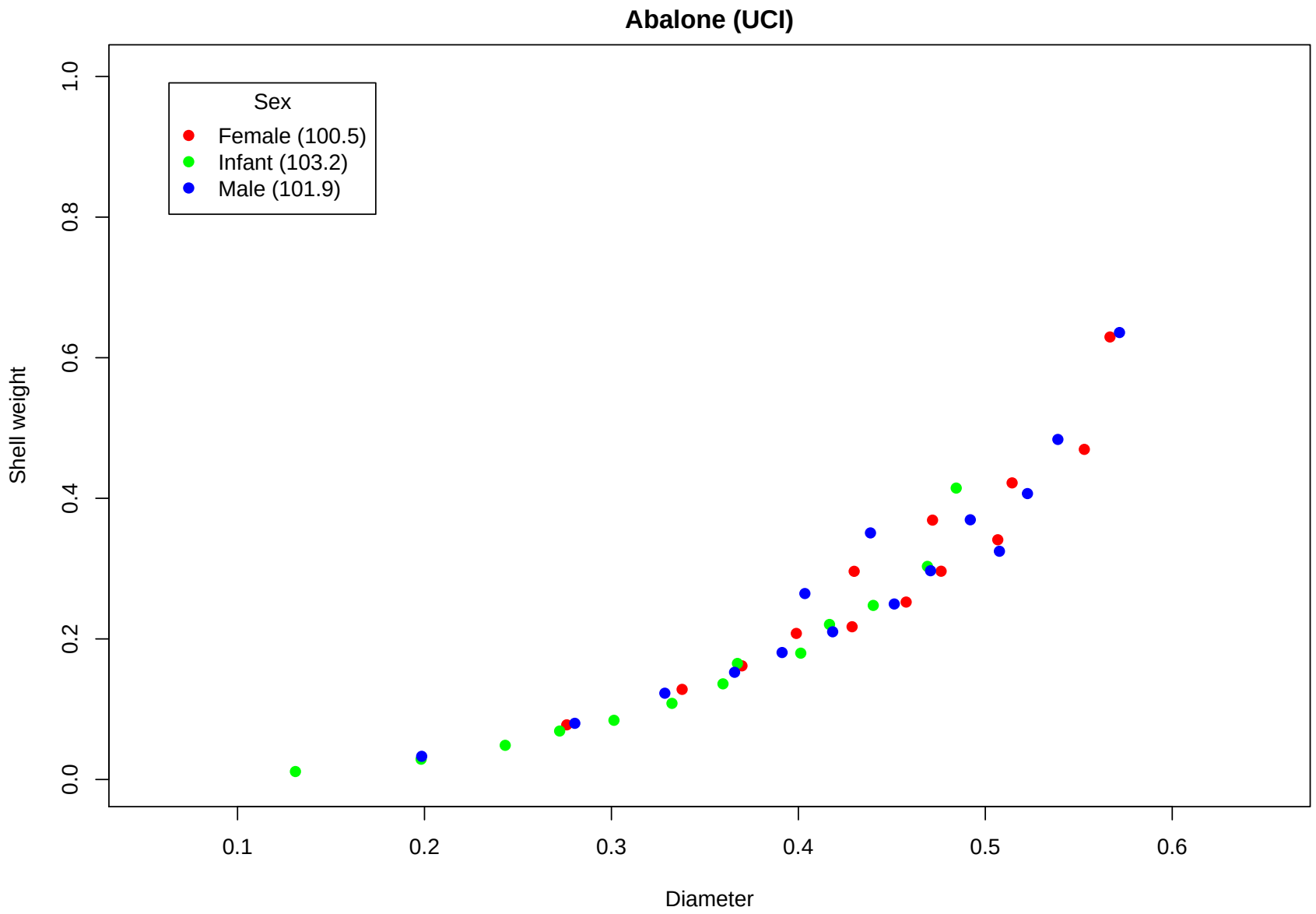
Hierarchical clustering



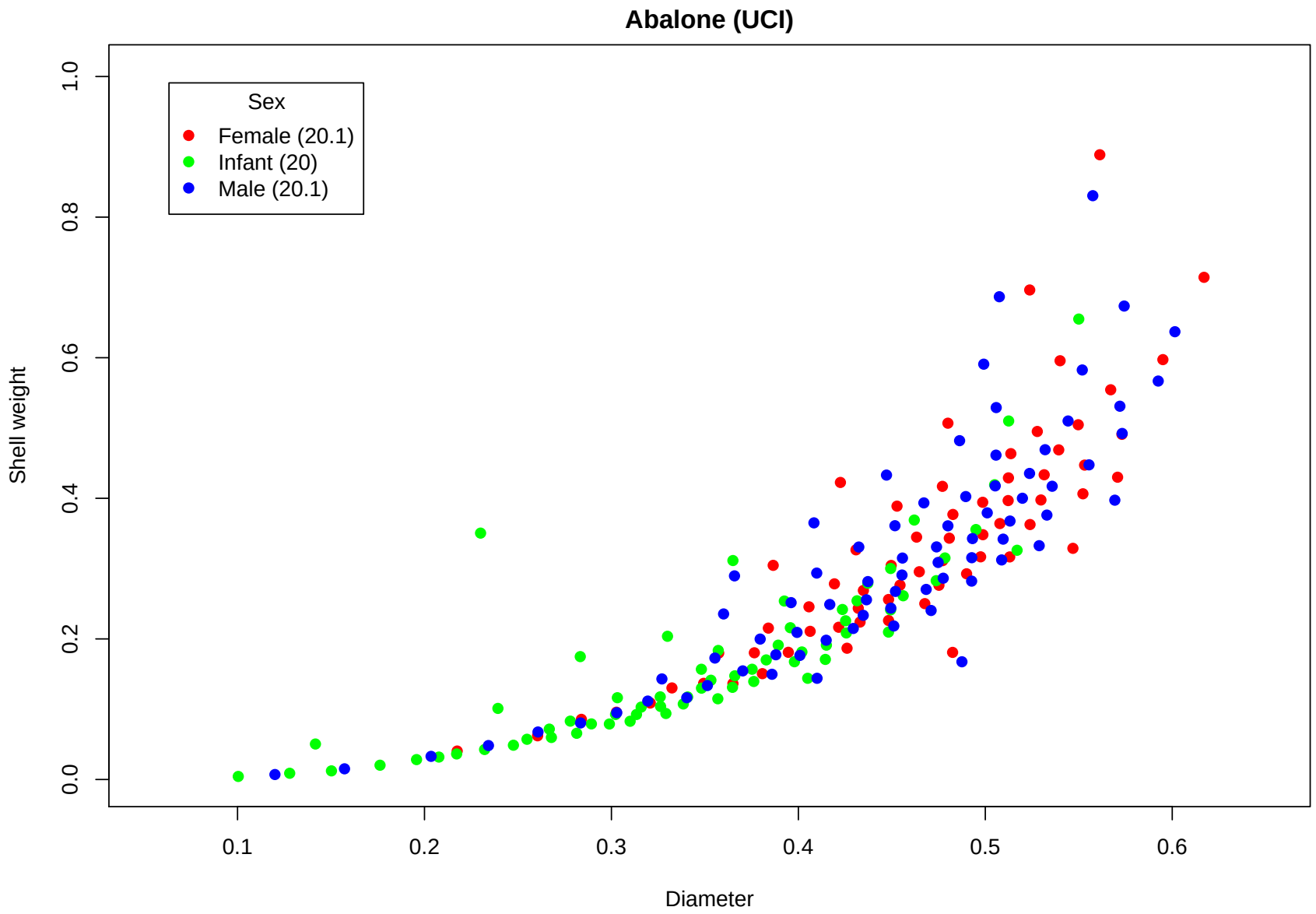
Hierarchical clustering



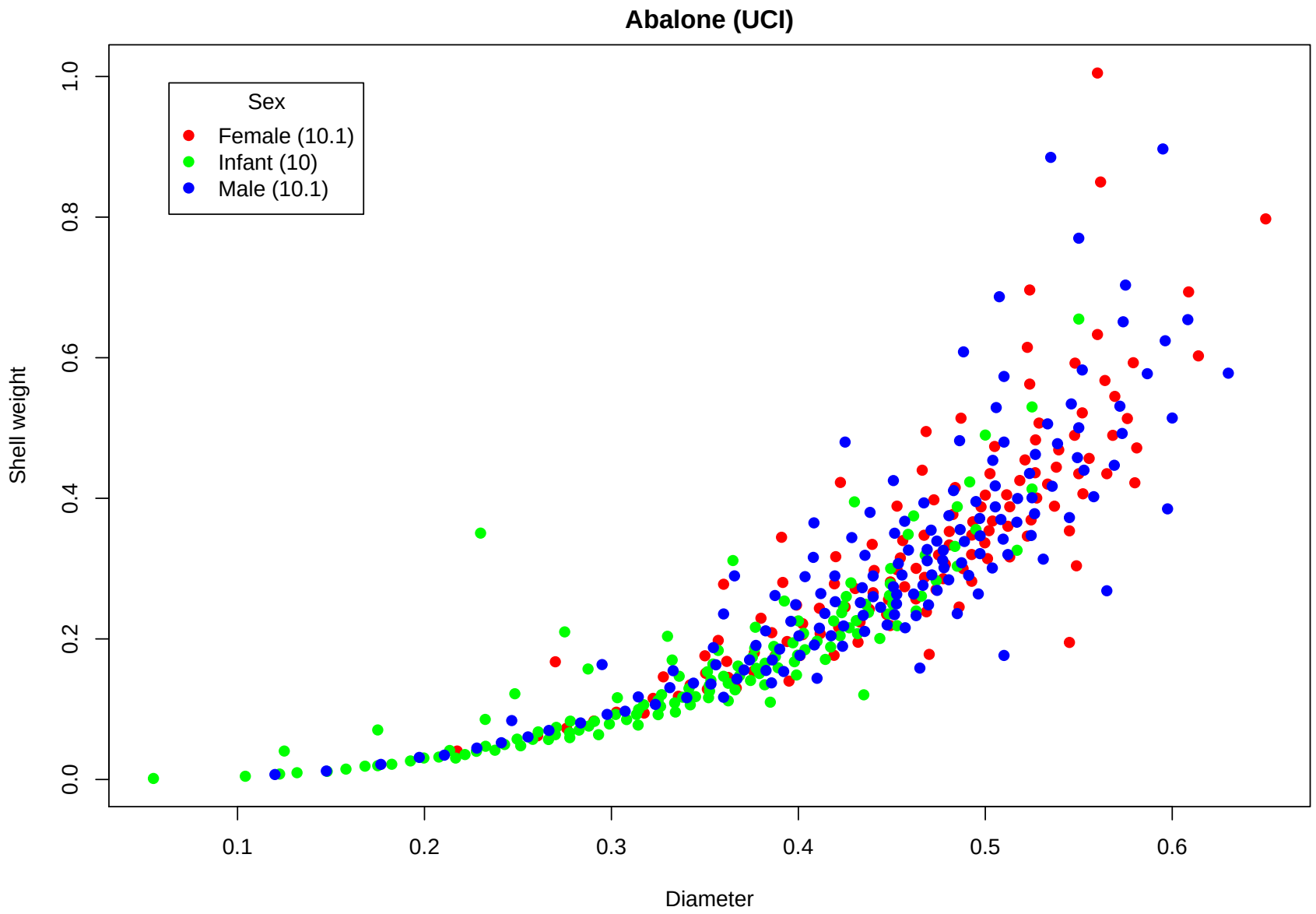
Hierarchical clustering



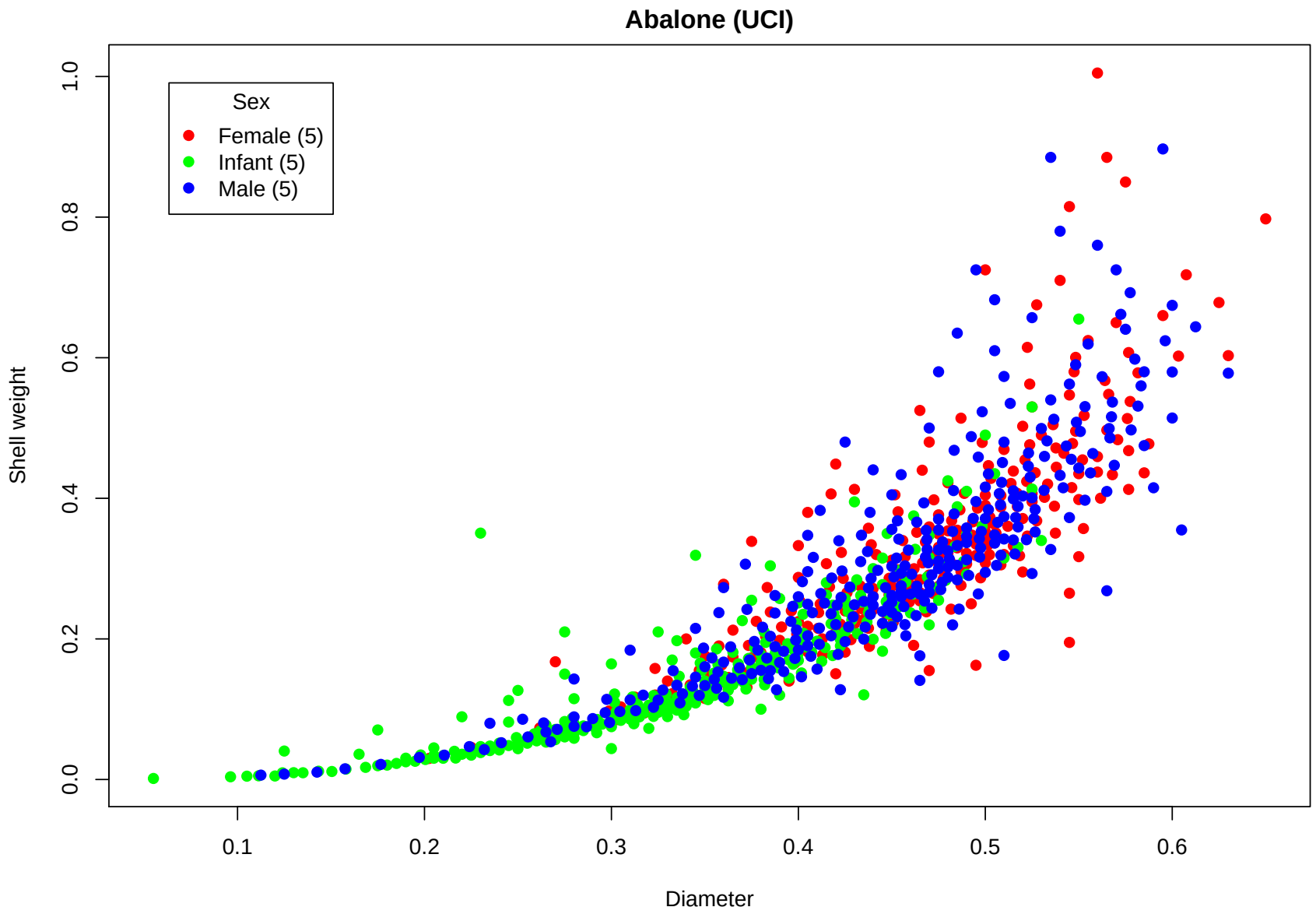
Hierarchical clustering



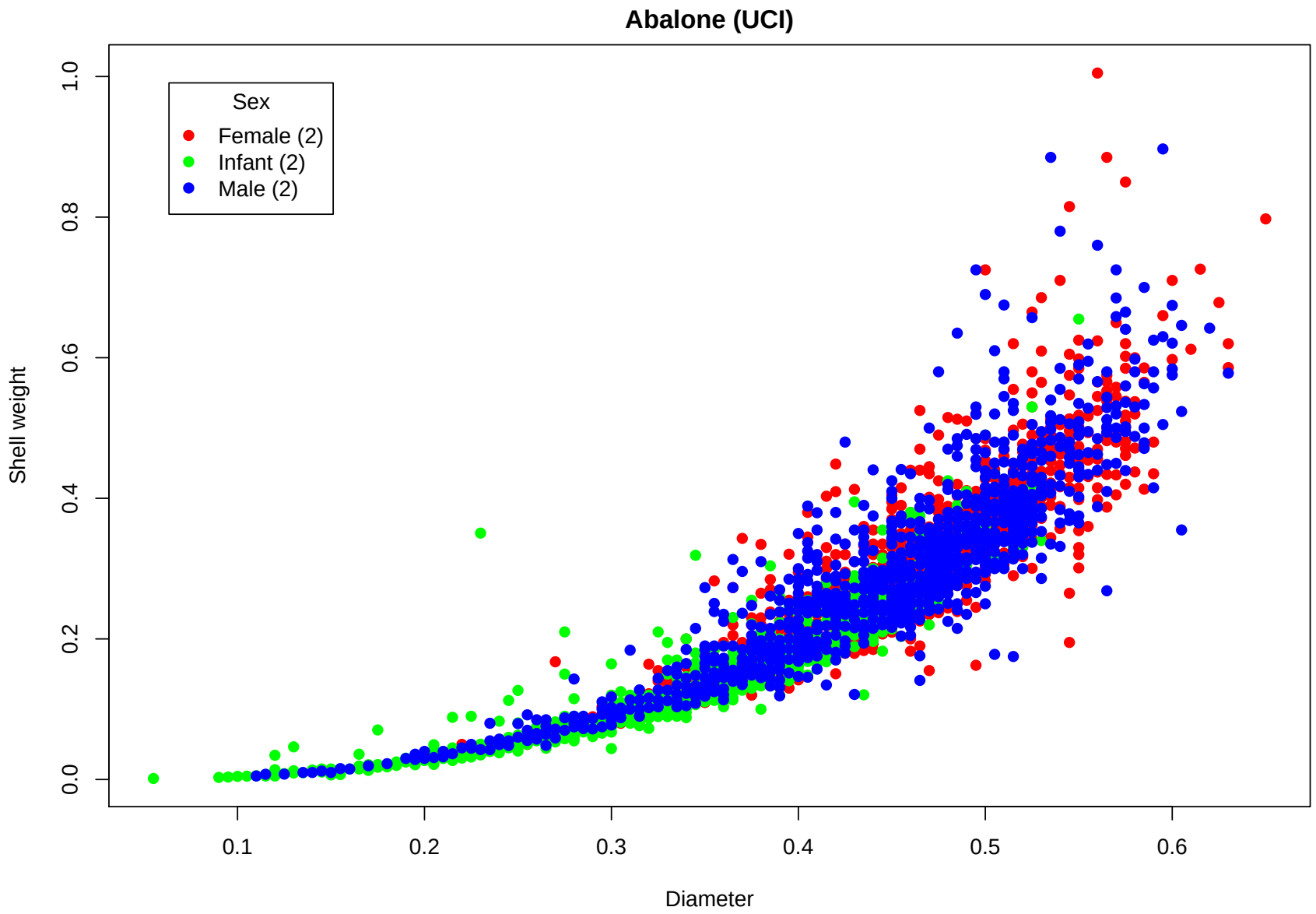
Hierarchical clustering



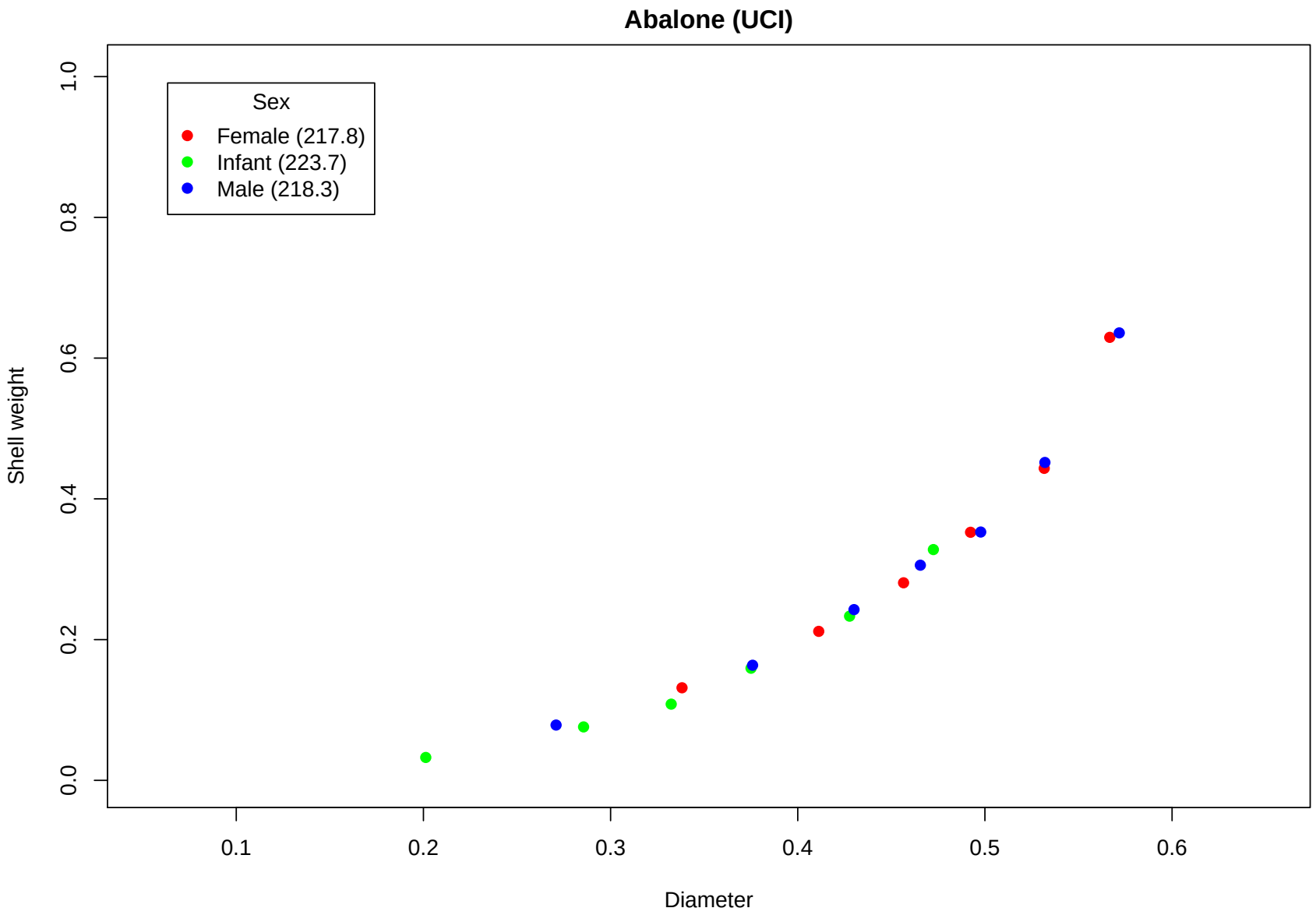
Hierarchical clustering



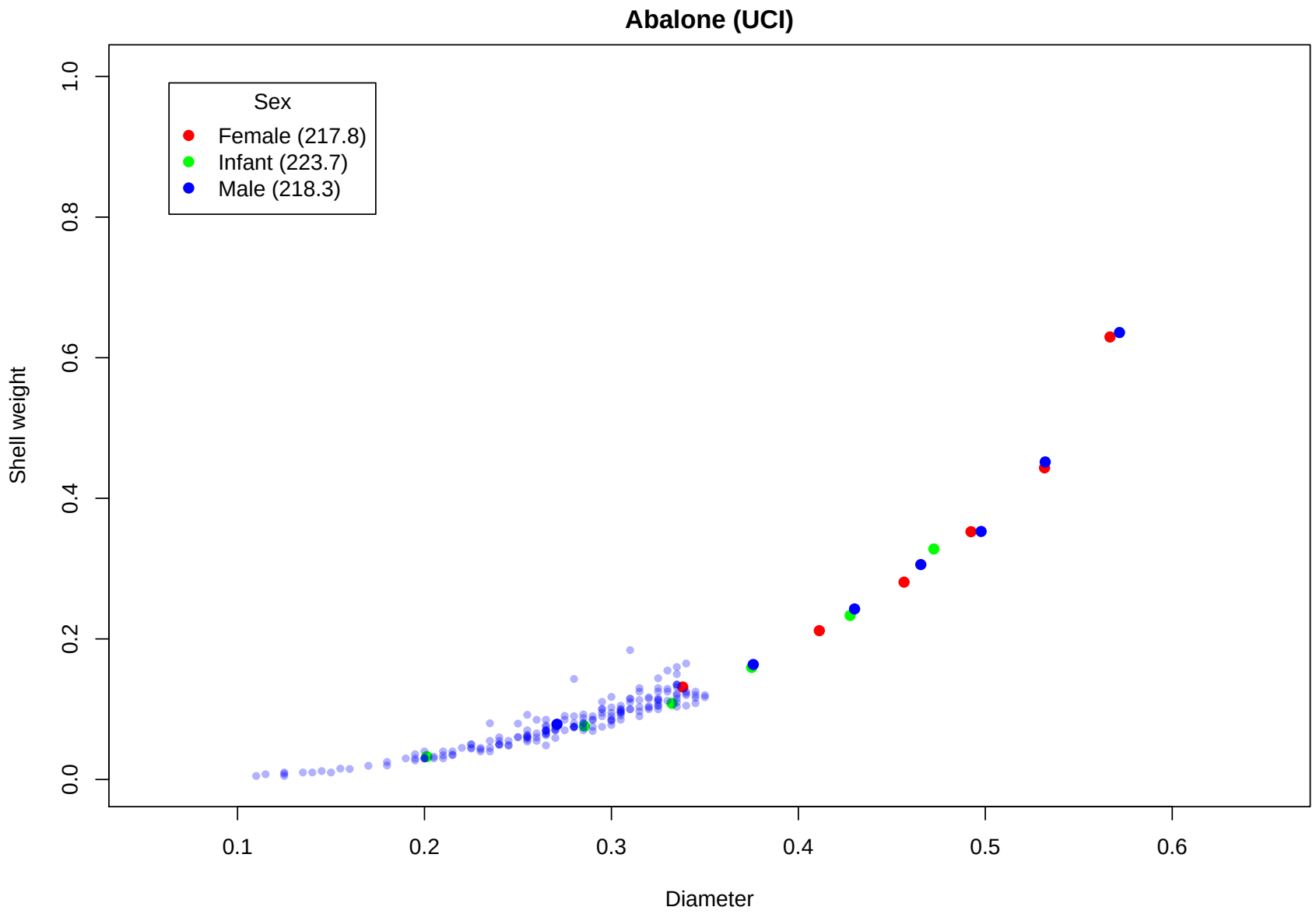
Hierarchical clustering



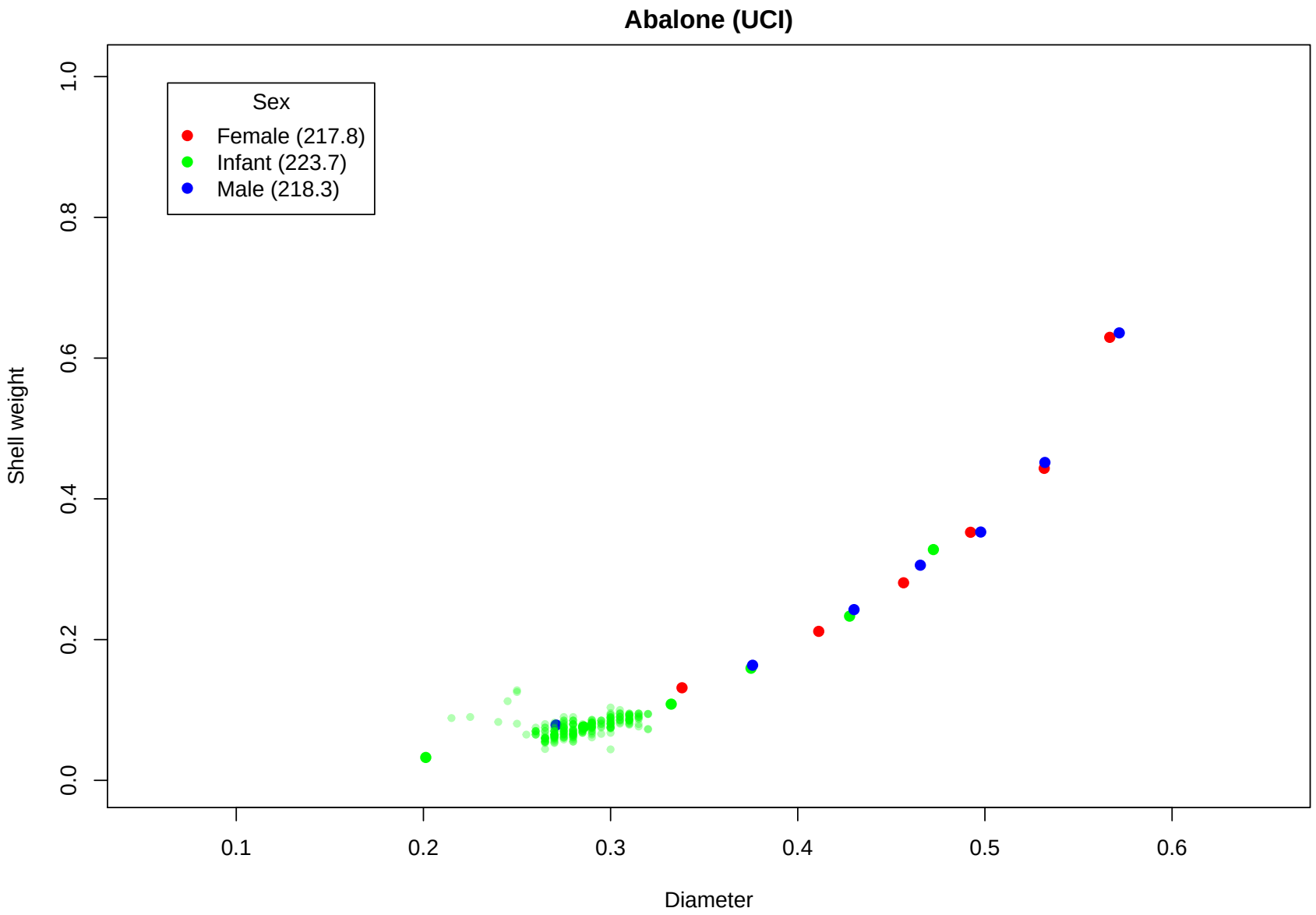
Cluster content



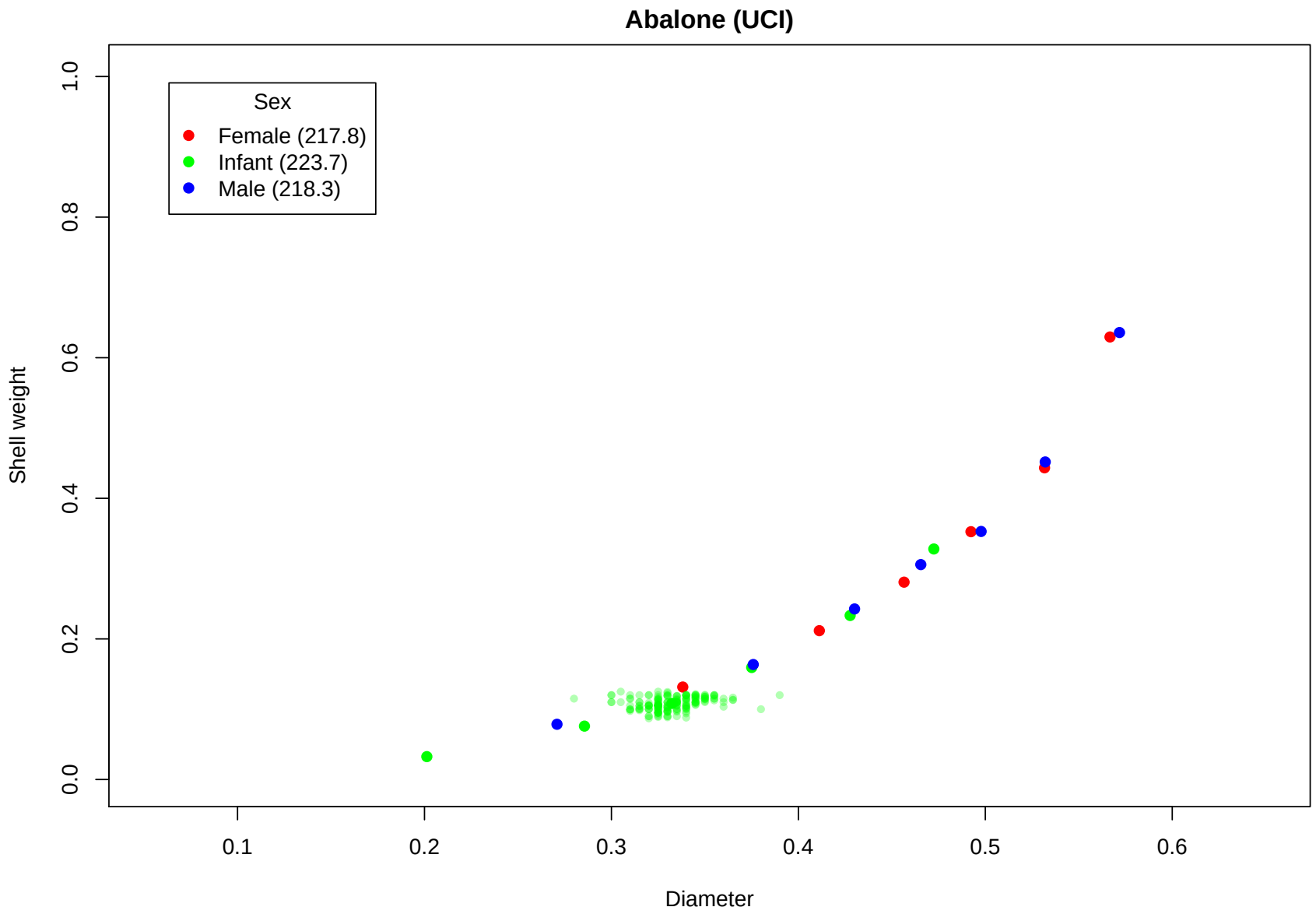
Cluster content



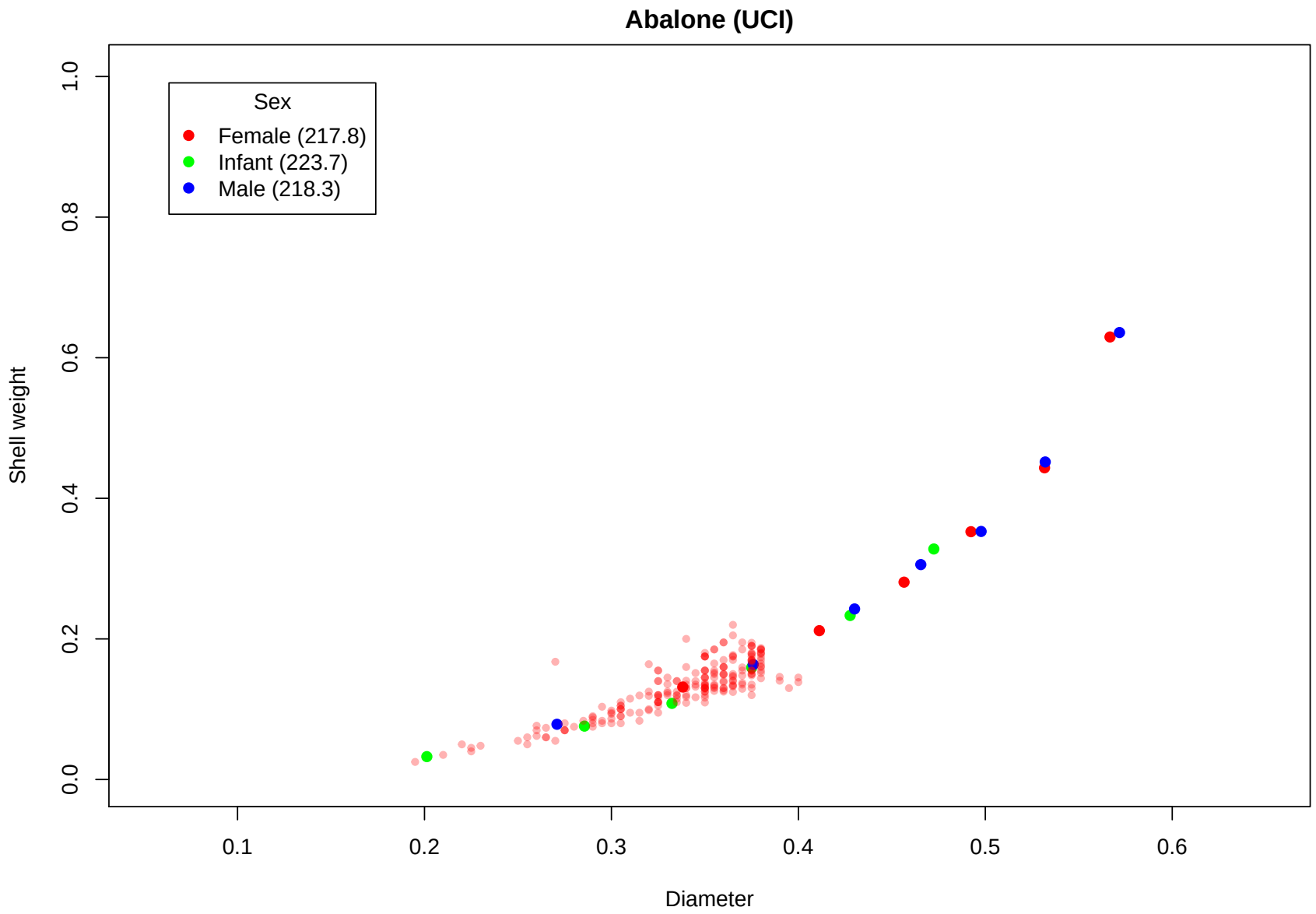
Cluster content



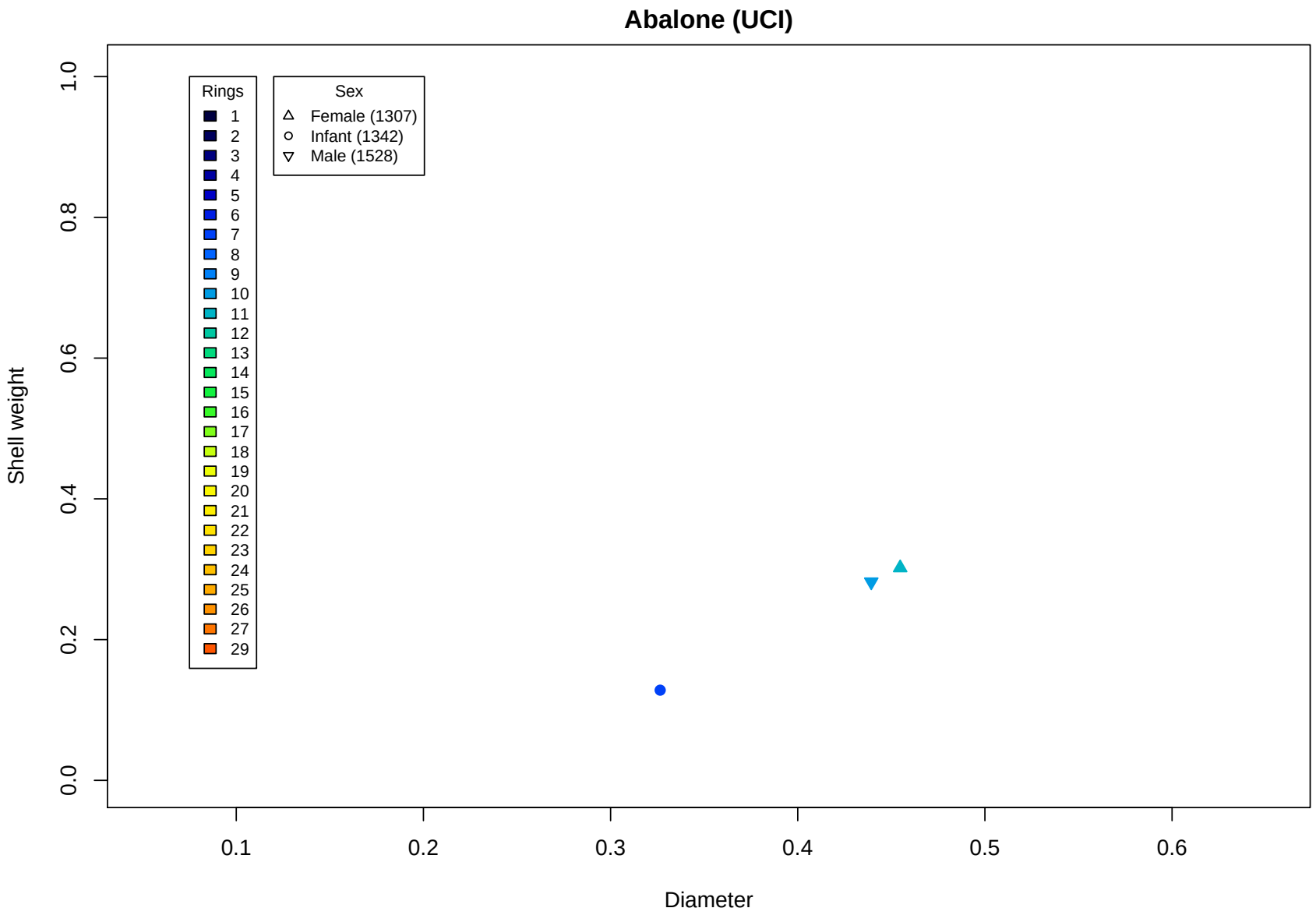
Cluster content



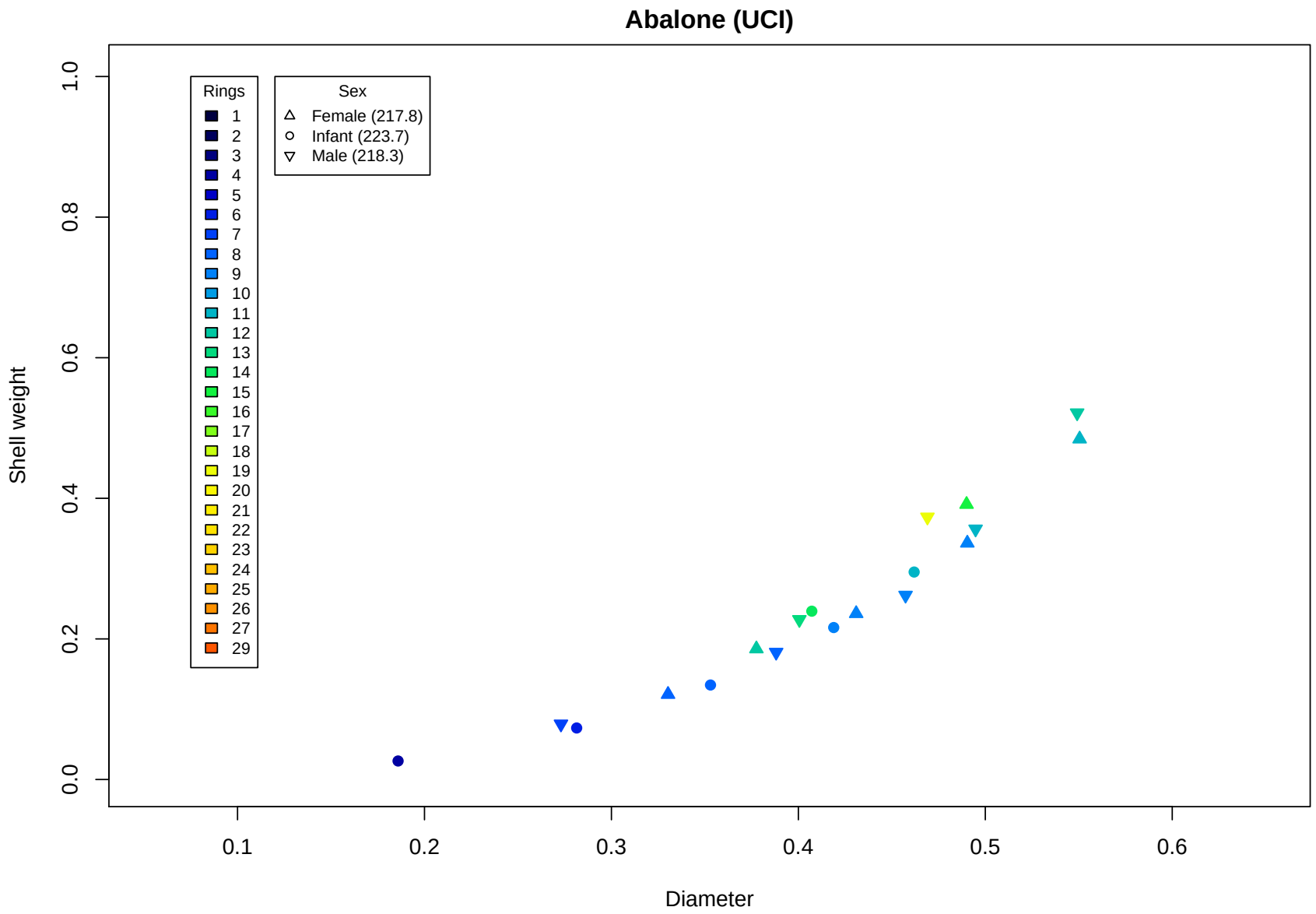
Cluster content



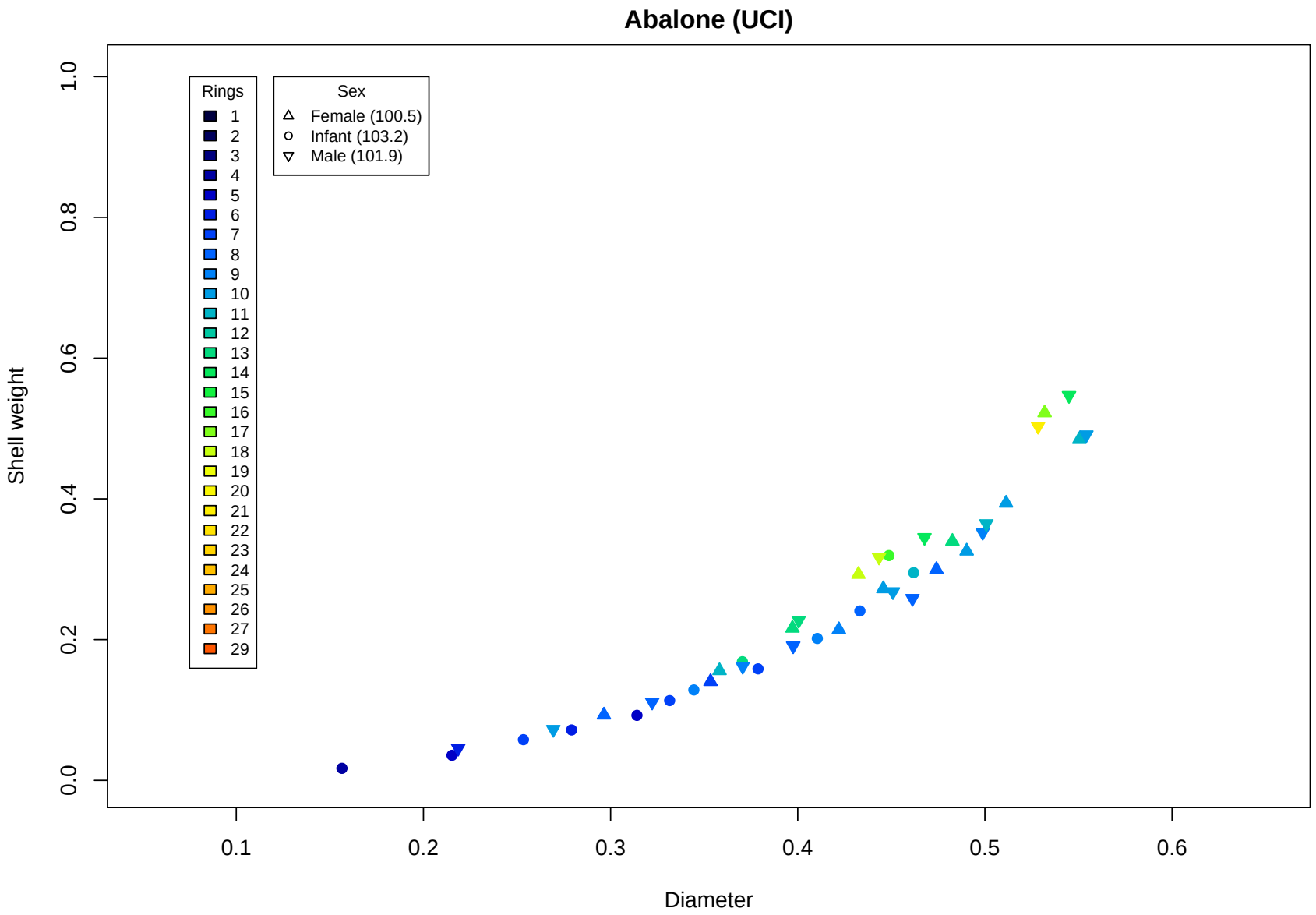
Hierarchical clustering



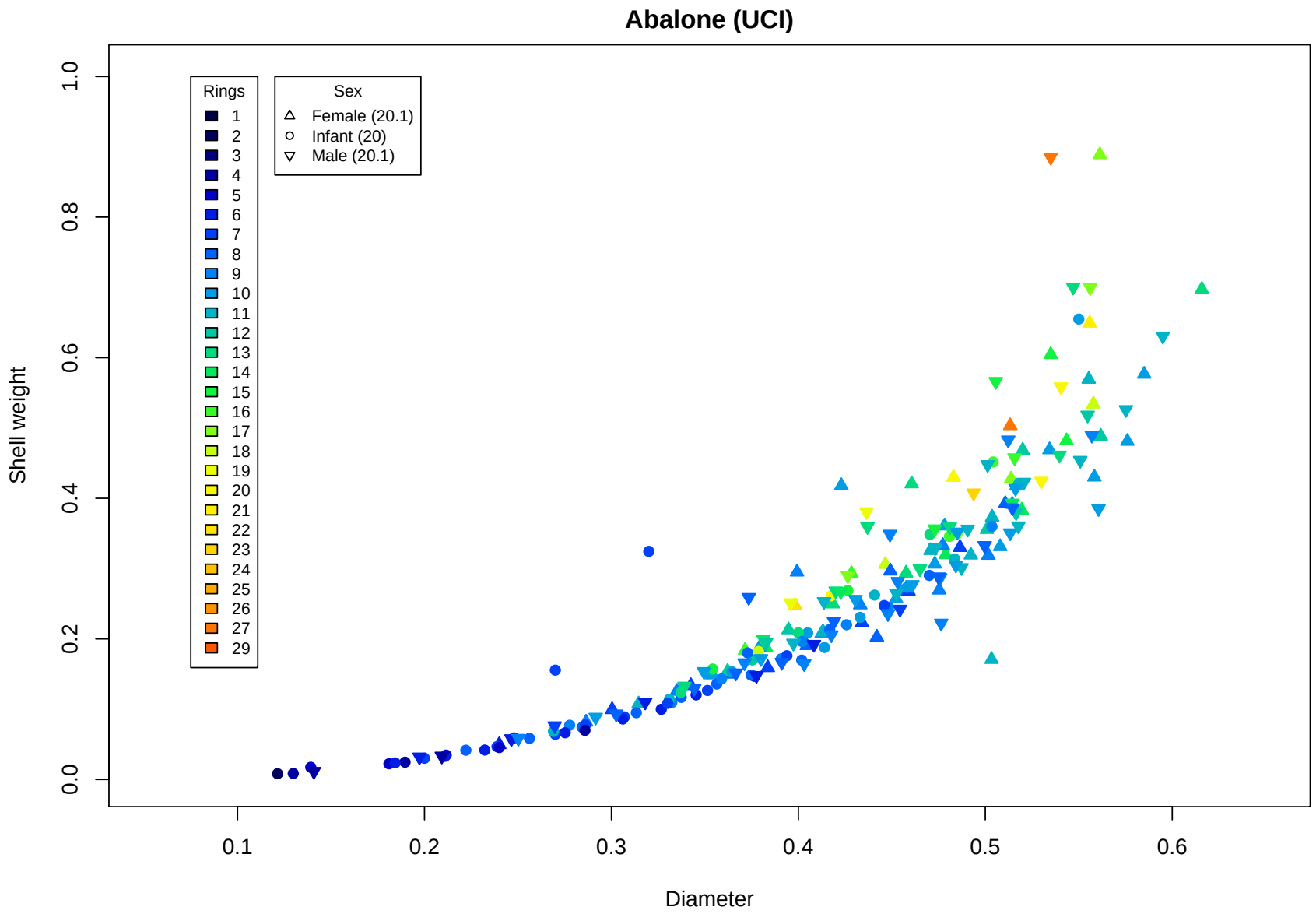
Hierarchical clustering



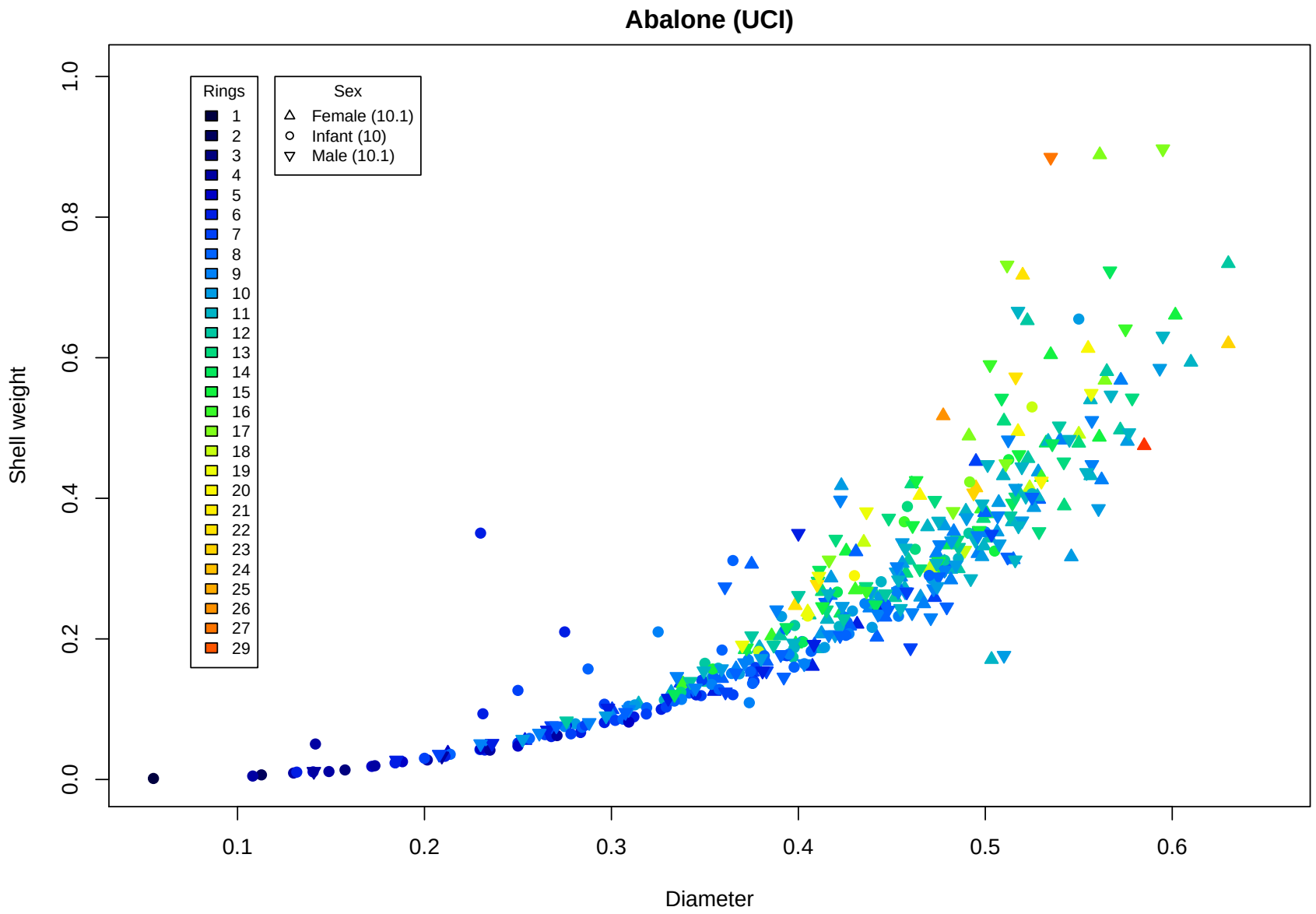
Hierarchical clustering



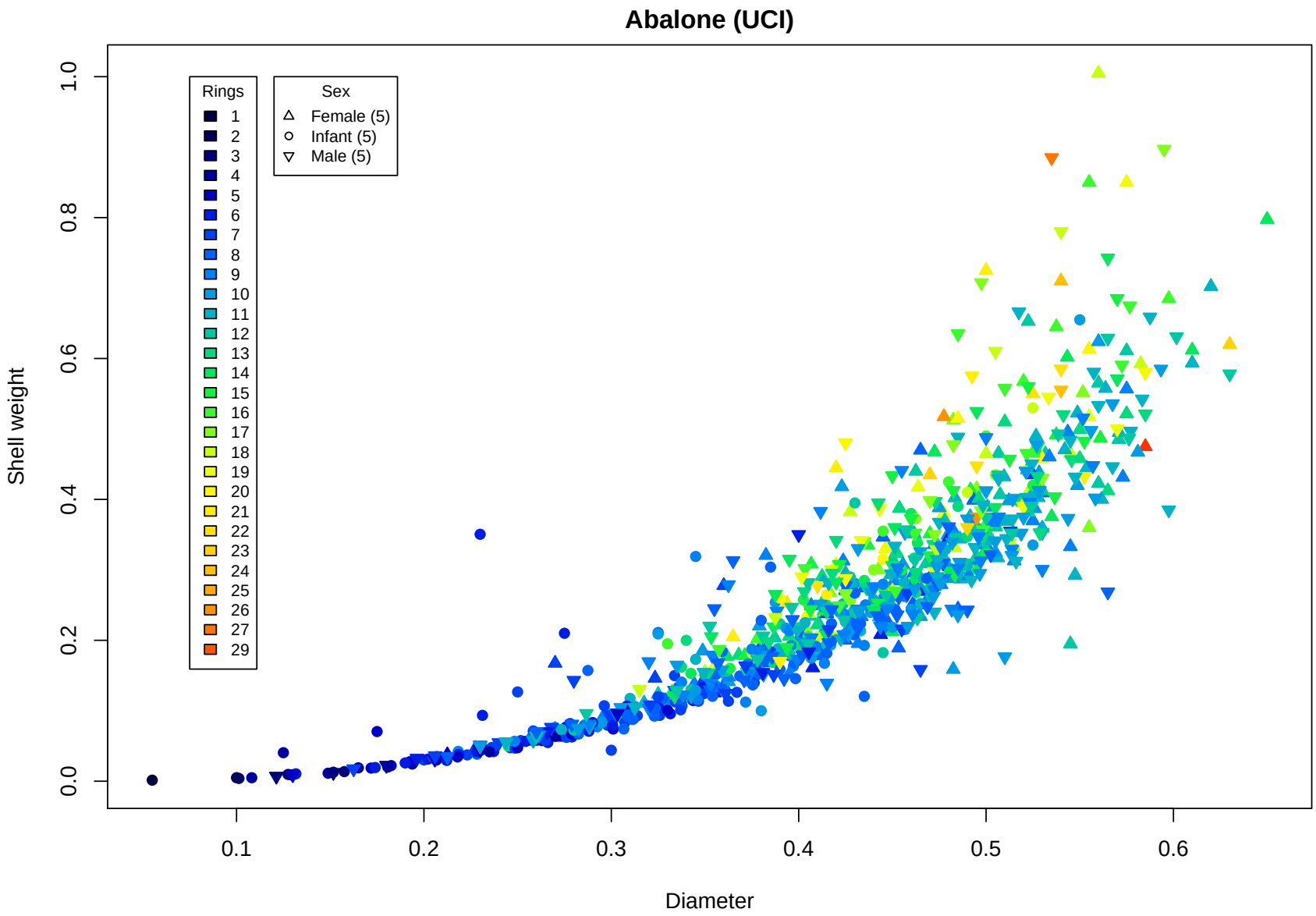
Hierarchical clustering



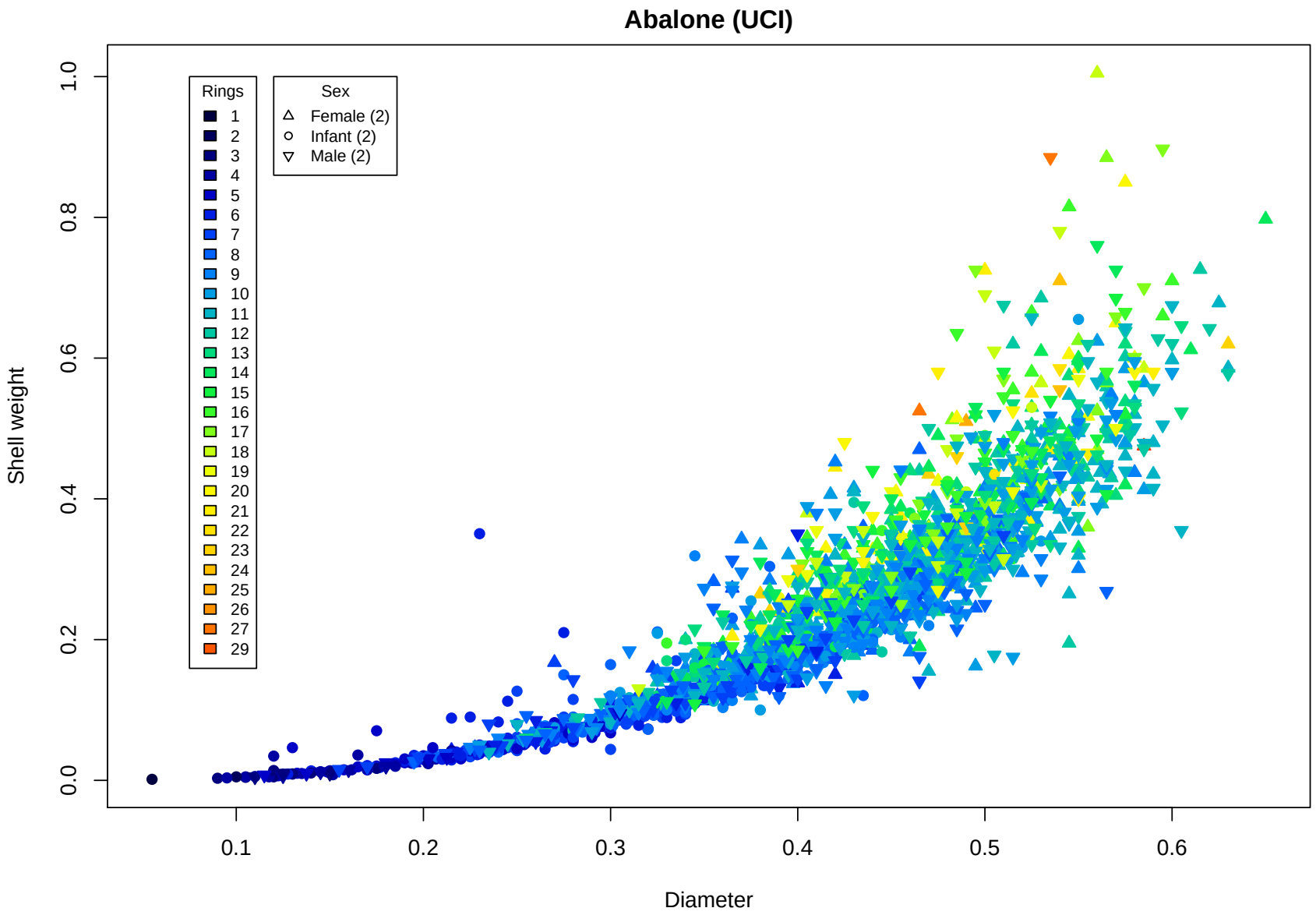
Hierarchical clustering



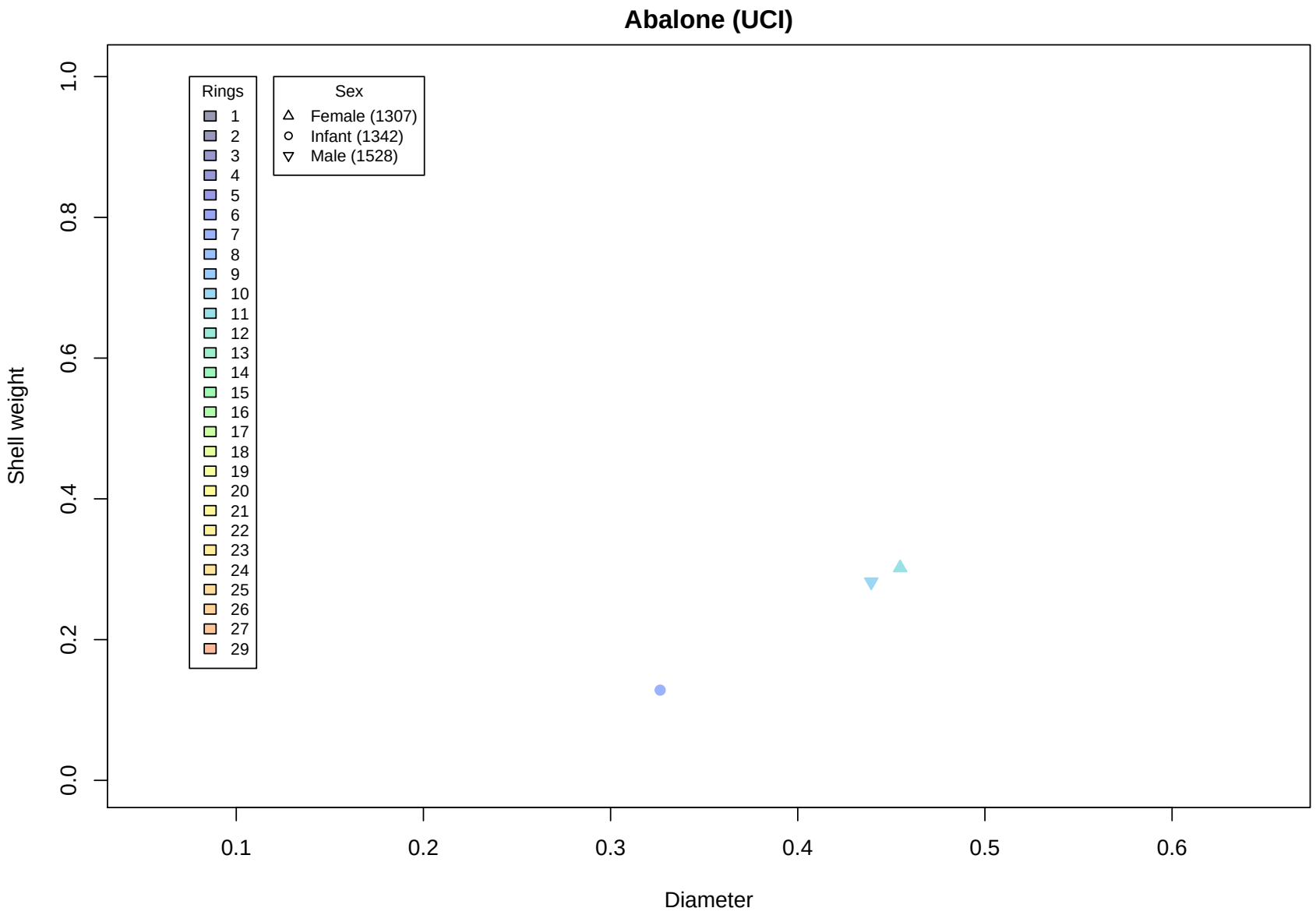
Hierarchical clustering



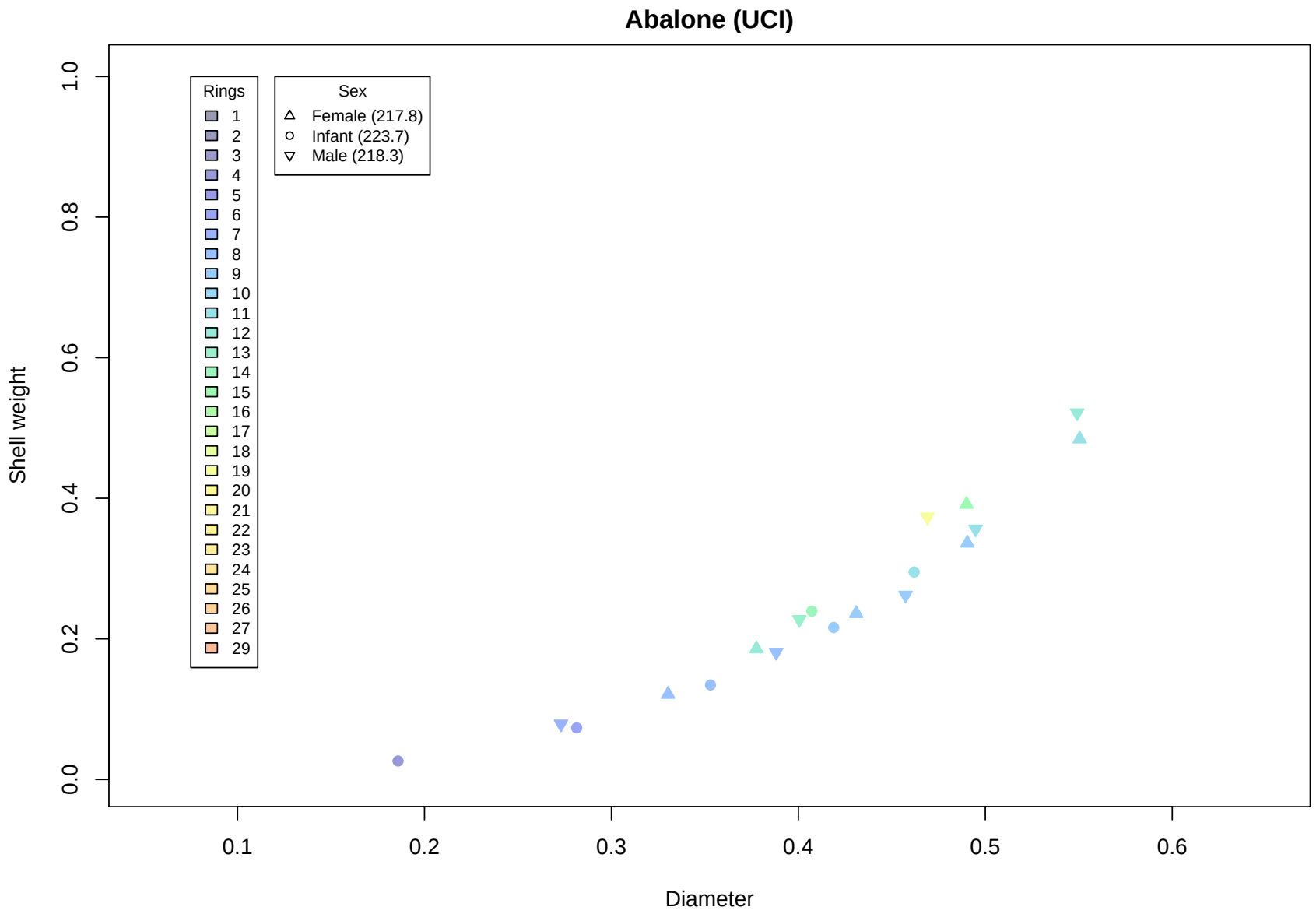
Hierarchical clustering



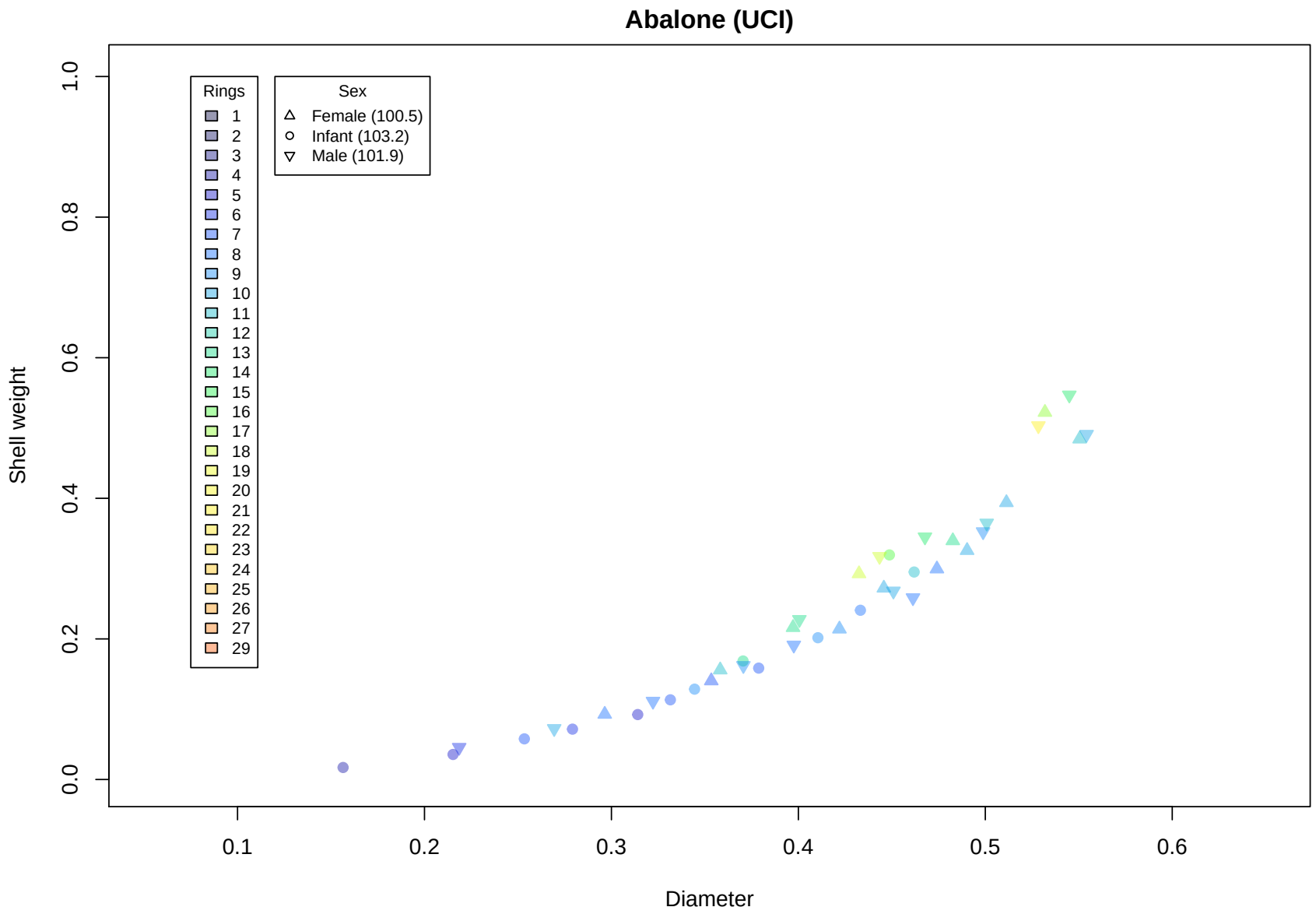
Hierarchical clustering and Transparency



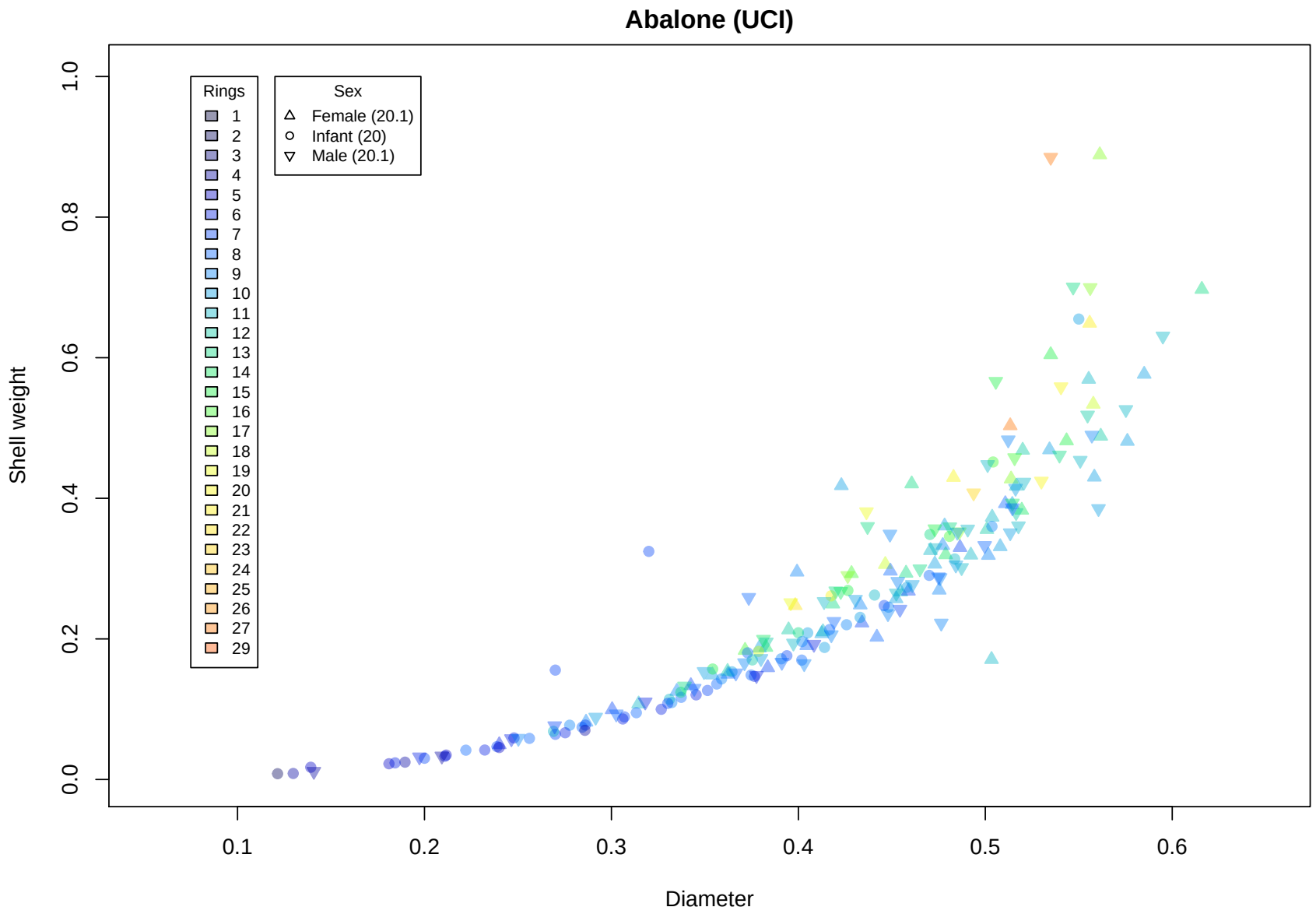
Hierarchical clustering and Transparency



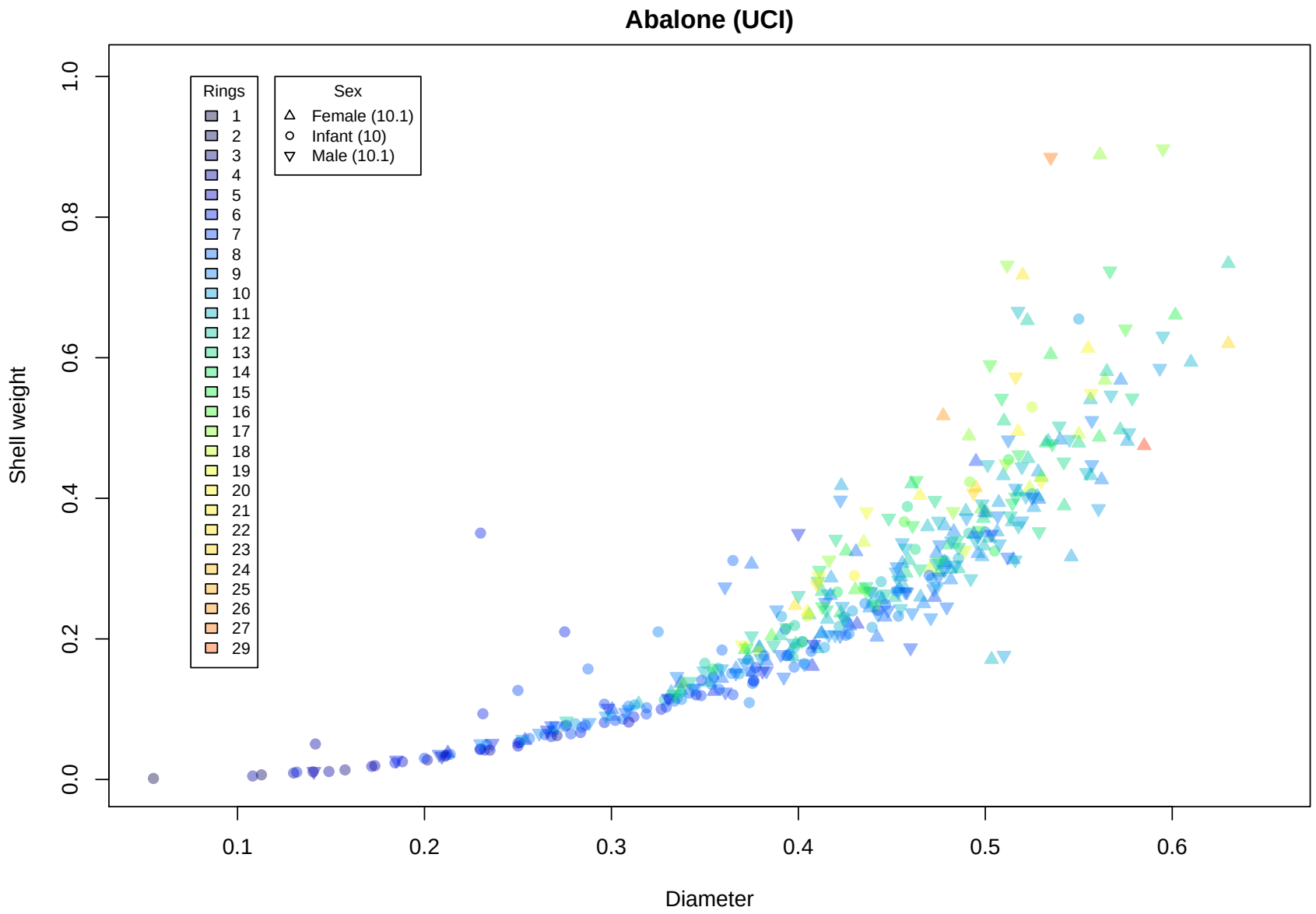
Hierarchical clustering and Transparency



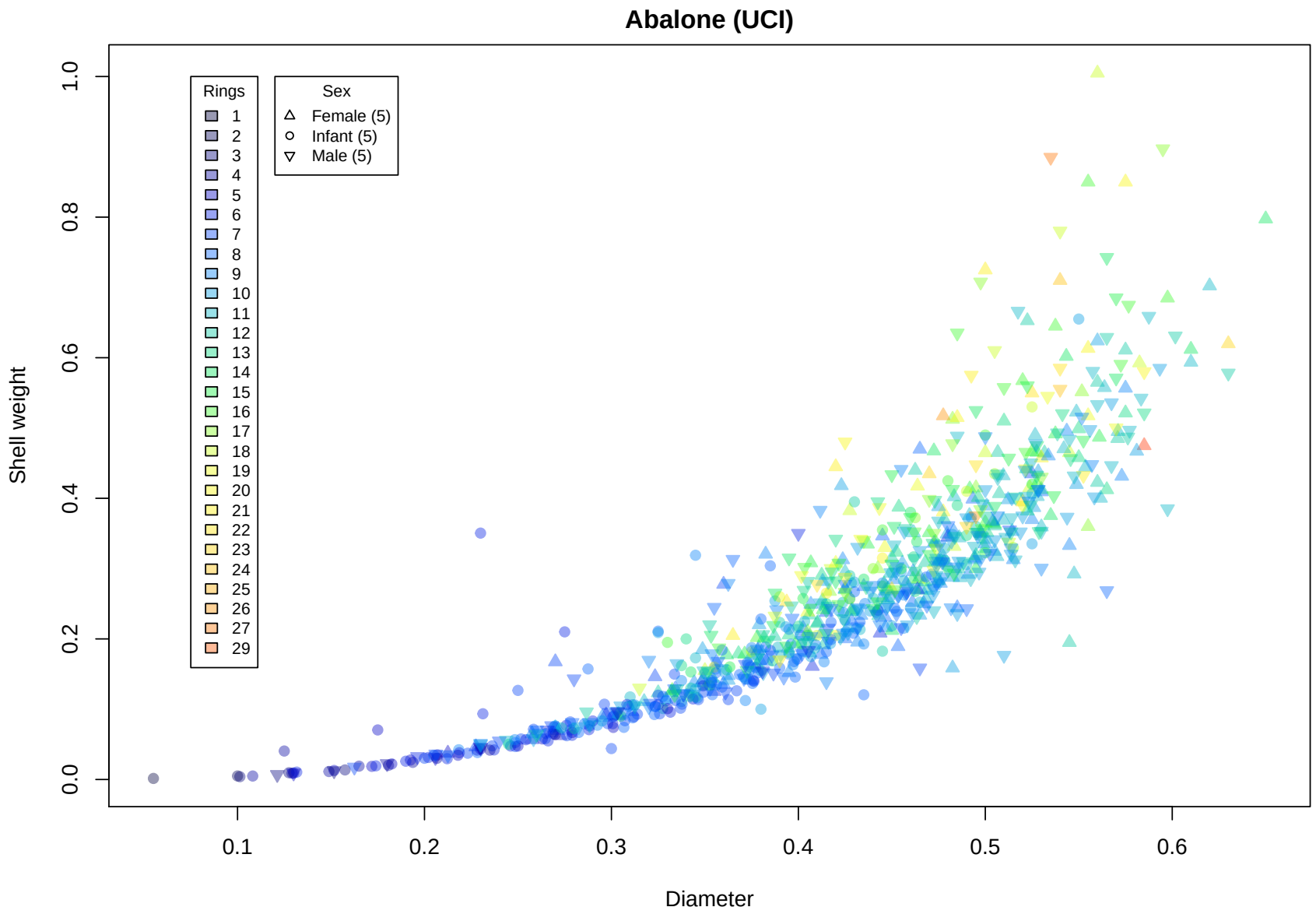
Hierarchical clustering and Transparency



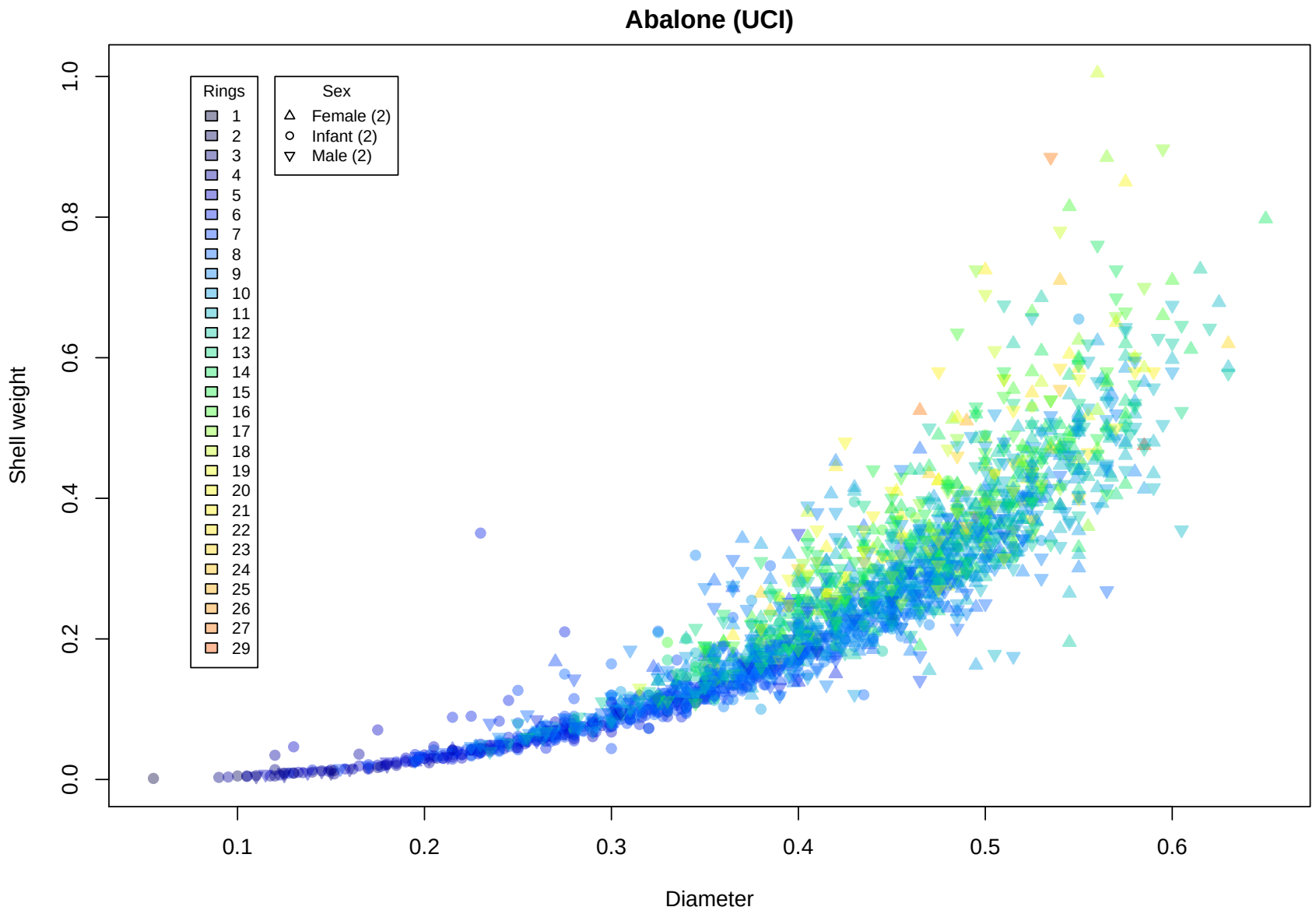
Hierarchical clustering and Transparency



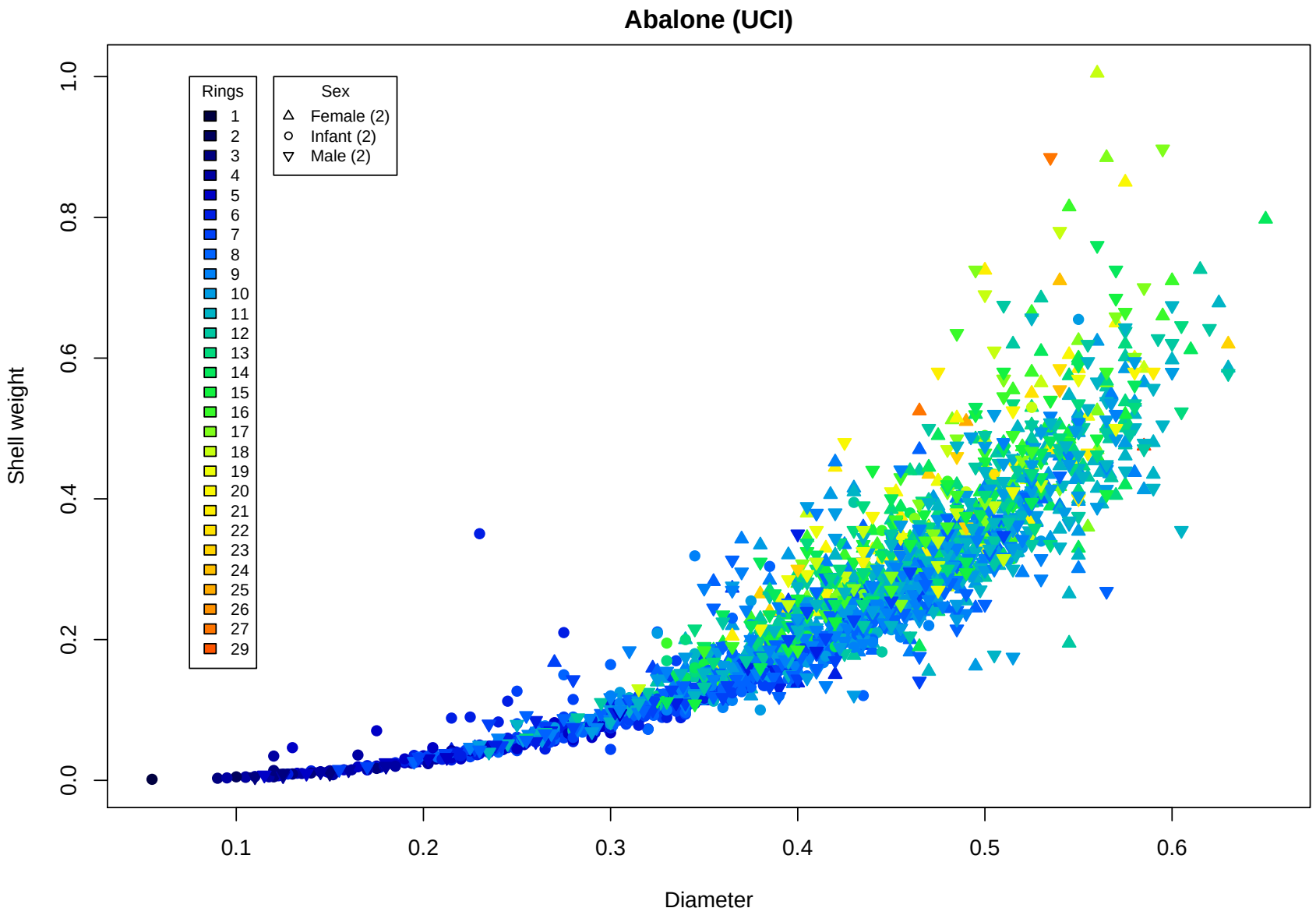
Hierarchical clustering and Transparency



Hierarchical clustering and Transparency



Hierarchical clustering and Transparency



Hierarchical clustering and Transparency