Current focus

- Huge amount of functionality in GGobi, and eventually want to be able to control it all from R (or any scripting language)
- Currently focused on getting data (and "meta" data) into and out of GGobi
- Simple, but surprisingly powerful

Getting data into rggobi

- Really easy!
  - library(rggobi)
  - ggobi(tips)

Modifying data

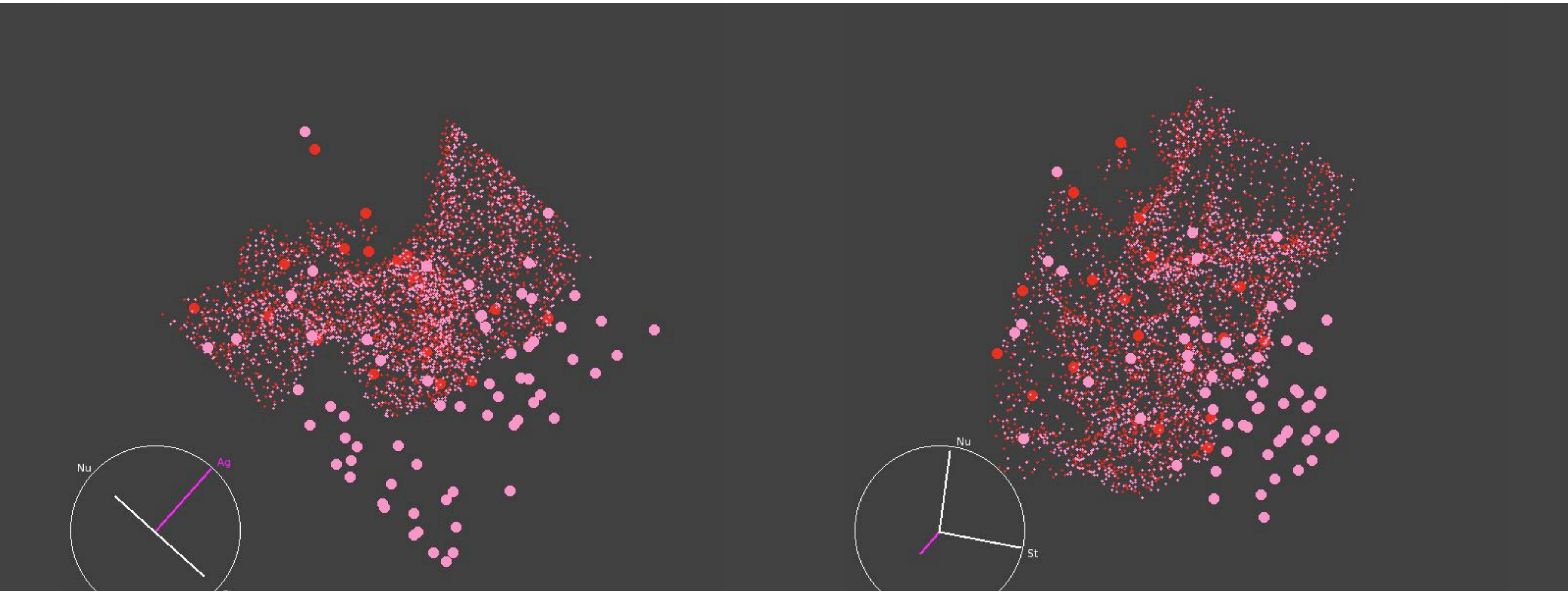
- A little bit trickier, because you need to keep a reference around
  - g <- ggobi(mtcars)
  - x <- g["mtcars"] or x<-g[1]
- x acts (almost) like a regular data.frame but changes are synced with GGobi

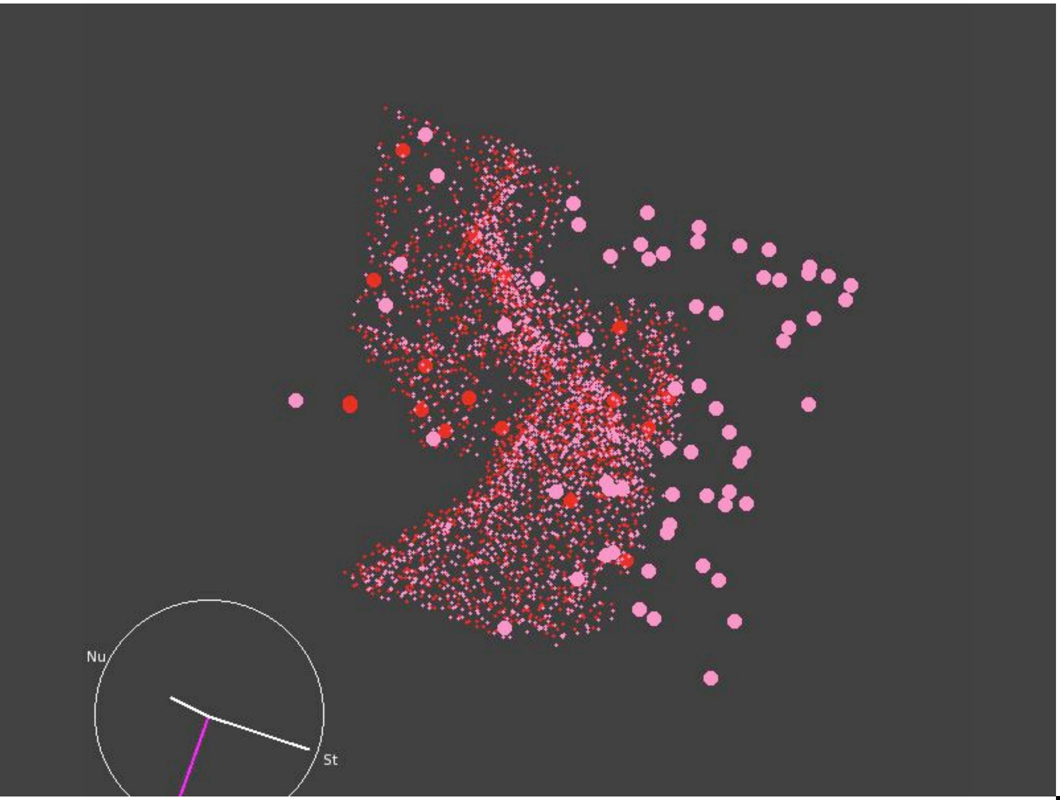
“Brushing”

- glyph_size(x)
- glyph_colour(x)
- glyph_type(x)
- shadowed(x)
- excluded(x)
- (and selected(x))

Some examples

- Animation (Pong)
- Exploring high-dimensional classification surfaces
- Functional data analysis
- Exploratory model analysis
- Microarray checking and analysis (Michael’s explorase)
- SpikeSorting (up next)





The future

- Construction of custom visualisations from R
- (Distant future): reproducible graphical analysis by storing GGobi sessions as R code

Installation

- Windows: very easy
- Linux: relatively easy (and very easy on Debian)
- OS X: requires developer skills
  (as "easy" as building R packages on Windows)
- GGobi 2.0 is **much** less buggy than previous versions

http://ggobi.org/rggobi

(Or google for rggobi)