Fancy graphics #1

Force-directed diagrams

Philippe Van Kerm
University of Luxembourg and LISER

2020 e-London Stata user group meeting
September 11, 2020

[Outline]

Examples

Principles and mechanics

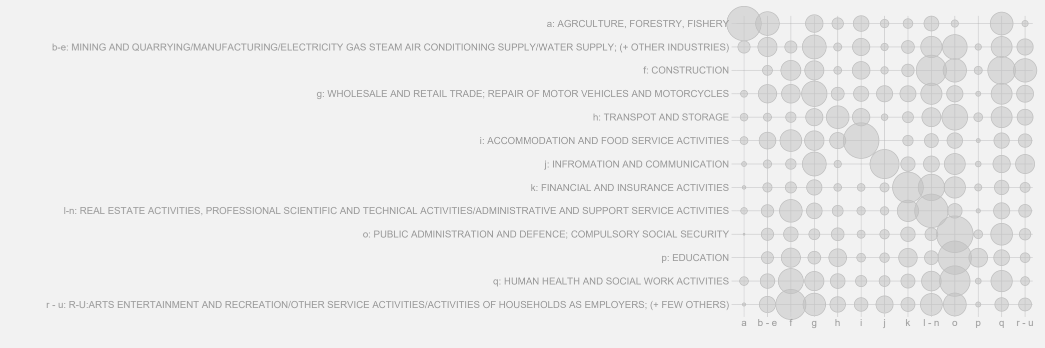
Implementation

Examples

Principles and mechanics

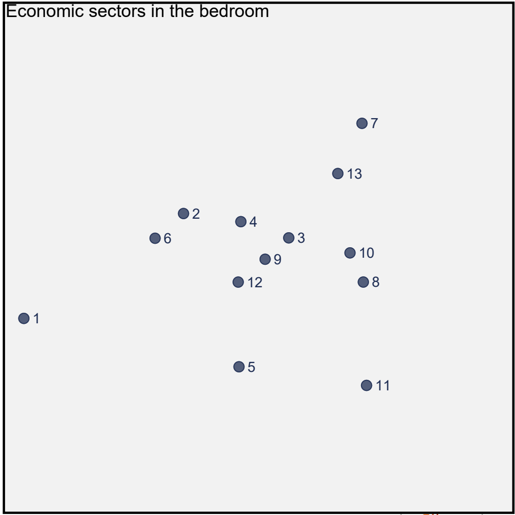
Implementation

My starting point...



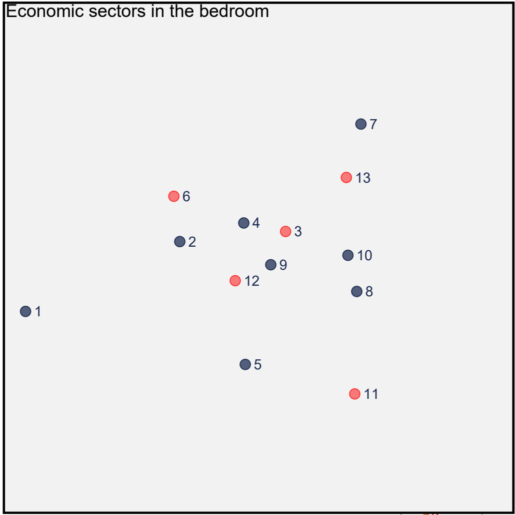
Network visualisation (as a force-directed diagram)

- The matrix is a network
- Each sector is a node
- Sectors with strong ‘bedroom association’ located near each other
- (See Thomas Grund’s `nwcommands` for serious network analysis with Stata; Corten (SJ, 2011), Miura (SJ, 2012).)

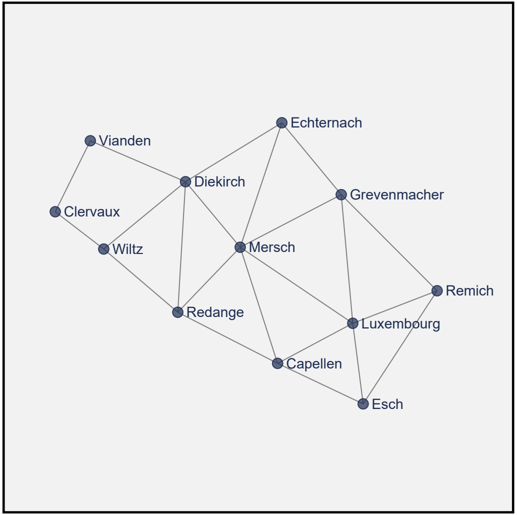


Network visualisation (as a force-directed diagram)

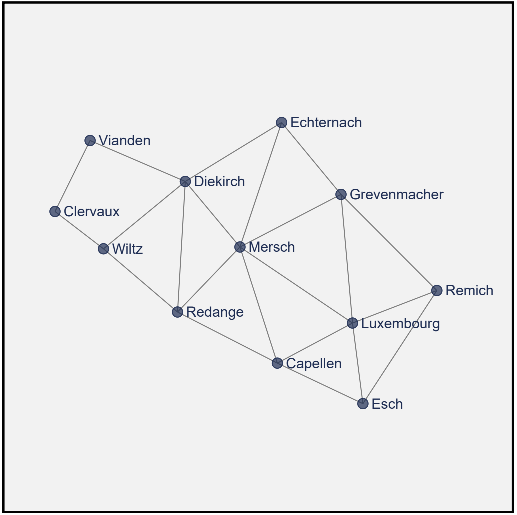
- The matrix is a network
- Each sector is a node
- Sectors with strong 'bedroom association' located near each other
- (See Thomas Grund's `nwcommands` for serious network analysis with Stata; Corten (SJ, 2011), Miura (SJ, 2012).)

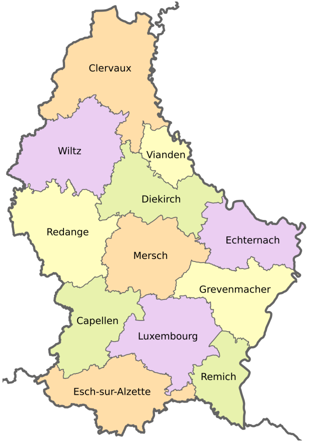


Another network visualisation (as a force-directed diagram)



Another network visualisation (as a force-directed diagram)

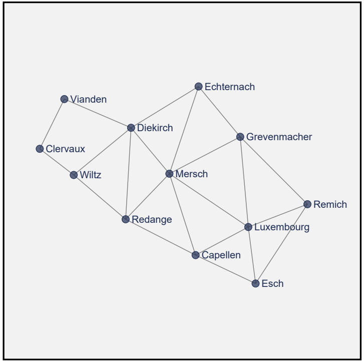




Another network visualisation package in the making

fdnetgraph *varname* [*if*] [*in*] , links(*fromvar tovar [strengthvar]*) [showlinks(...) varcolor(varname) ...]

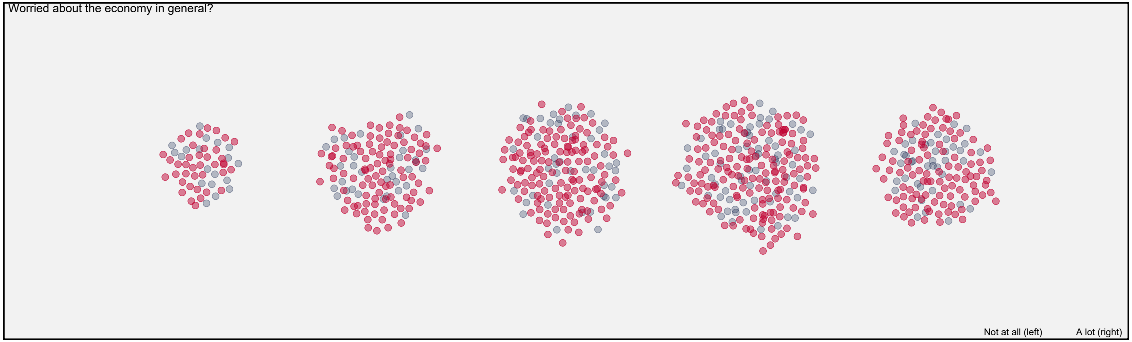
(too many options to discuss here (fiddling with the construction and display options; see below))



Other force-directed diagrams: ‘beeswarm’ plots

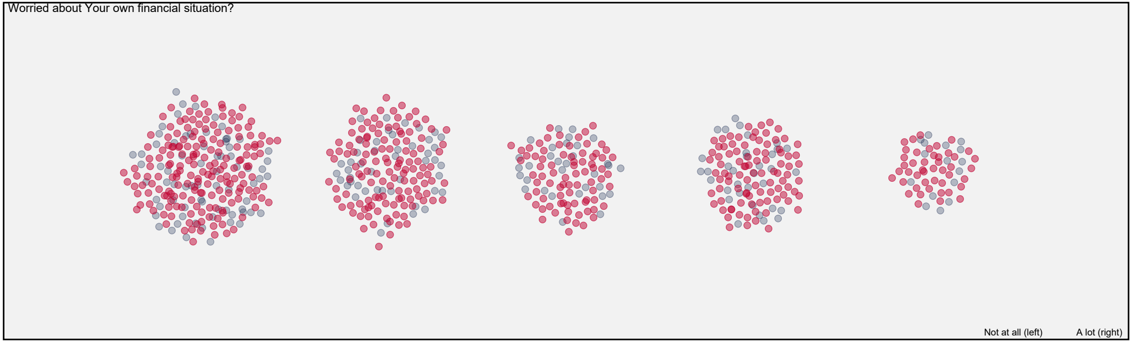
Other force-directed diagrams: ‘beeswarm’ plots

Worried about the economy?



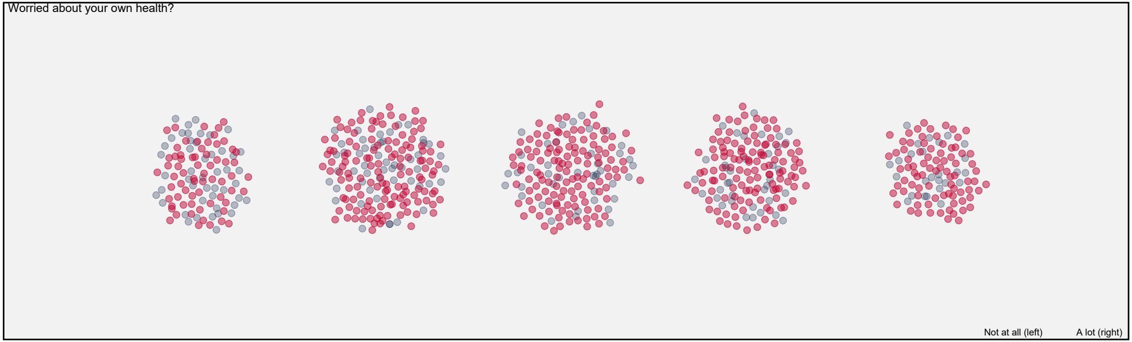
Other force-directed diagrams: ‘beeswarm’ plots

Worried your own finances?



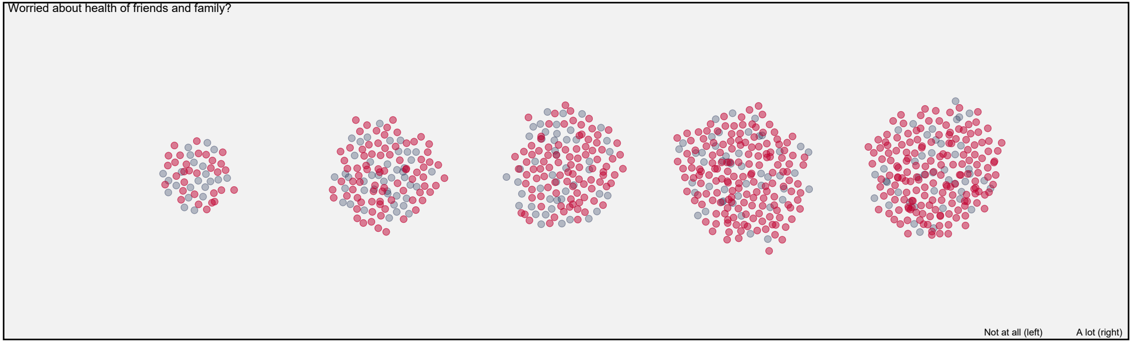
Other force-directed diagrams: ‘beeswarm’ plots

Worried your health?



Other force-directed diagrams: ‘beeswarm’ plots

Worried about your friend and family’s health?



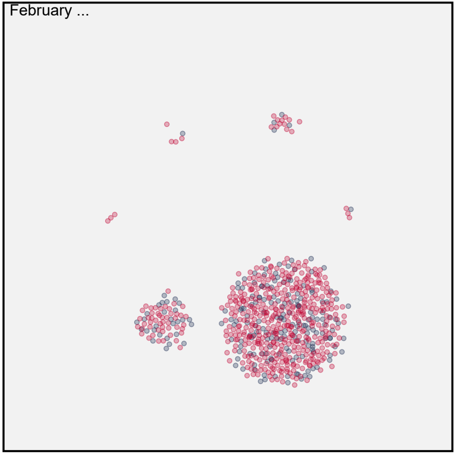
Variations on the same theme: other ‘beeswarm’ plots

The evolution of employment: telework, short-time employment, ‘parental’ leave, sick leave, job loss

February

early April

June



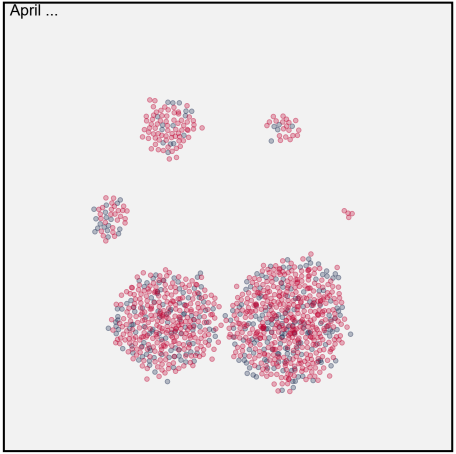
Variations on the same theme: other ‘beeswarm’ plots

The evolution of employment:
telework, short-time employment,
‘parental’ leave, sick leave, job loss

February

early April

June



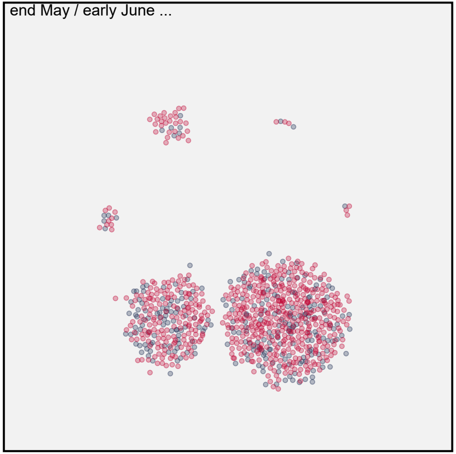
Variations on the same theme: other ‘beeswarm’ plots

The evolution of employment: telework, short-time employment, ‘parental’ leave, sick leave, job loss

February

early April

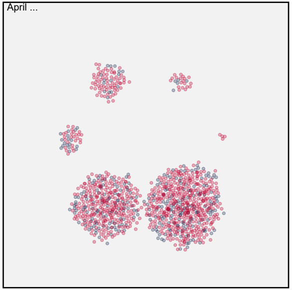
June



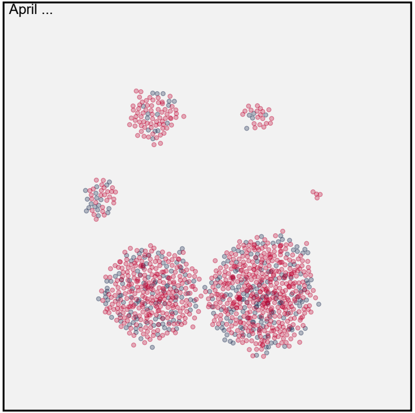
A beeswarm plot package in the making

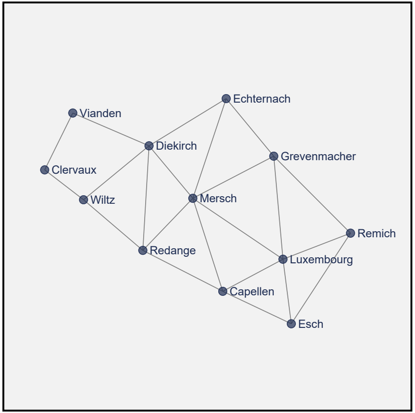
beeswarm [*varname*] [*if*] [*in*] ,
[...varcolor(varname) varsymbol(varname)...]

(too many options to discuss here (fiddling with the construction, the display, the choice of locations etc.; see below))



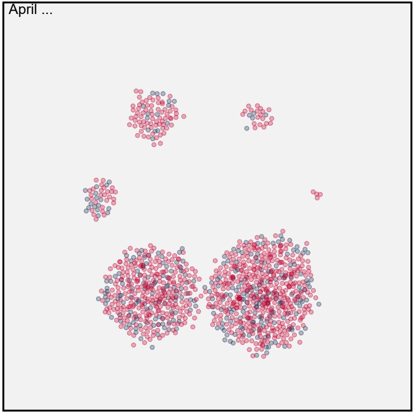
What is the commonality?





What is the commonality?

- Simple scatterplots...
- ... but elements have **no pre-defined location** on the canvas
- Key to the drawing is calculating the plotting positions!



Examples

Principles and mechanics

Implementation

Force-directed layouts

- Element positions determined by a stochastic simulation algorithm
- Elements “interact” with each other in order to find their position on the canvas

Force-directed layouts

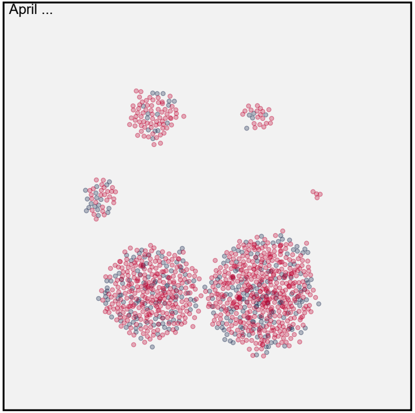
- Element positions determined by a stochastic simulation algorithm
- Elements “interact” with each other in order to find their position on the canvas
- Interactions through combinations of few simple forces:
  - Gravitational forces (positive or negative; attractions or repulsion)
  - Spring forces (towards target distances)
  - Collisions

Force-directed layouts

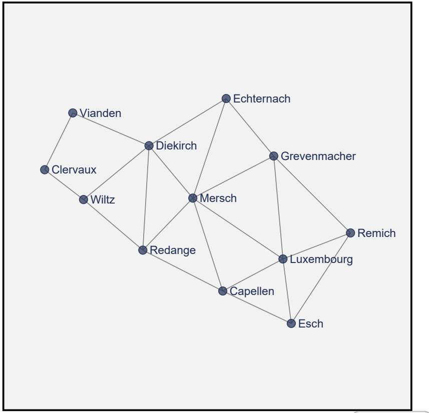
- Element positions determined by a stochastic simulation algorithm
- Elements “interact” with each other in order to find their position on the canvas
- Interactions through combinations of few simple forces:
    - Gravitational forces (positive or negative; attractions or repulsion)
    - Spring forces (towards target distances)
    - Collisions
- Start from random positions and iterate until convergence to a stable plot
- Stochastic: randomness in the resulting plot (set your `seed`)!

Different combination of forces lead to different types of plots

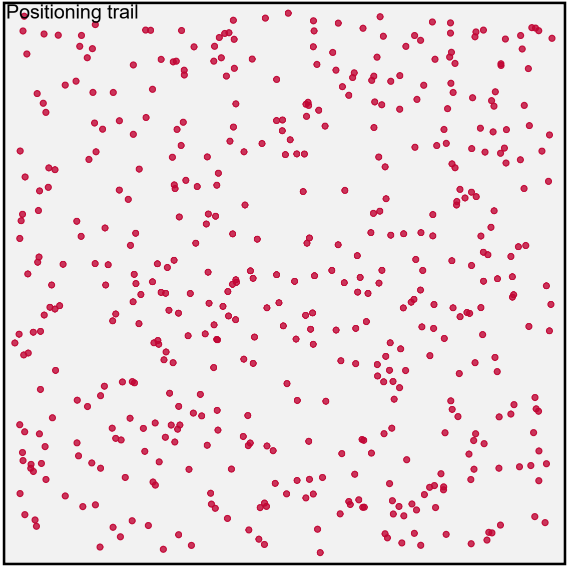
Attraction forces between points and towards ‘anchors’ (hives) + collisions



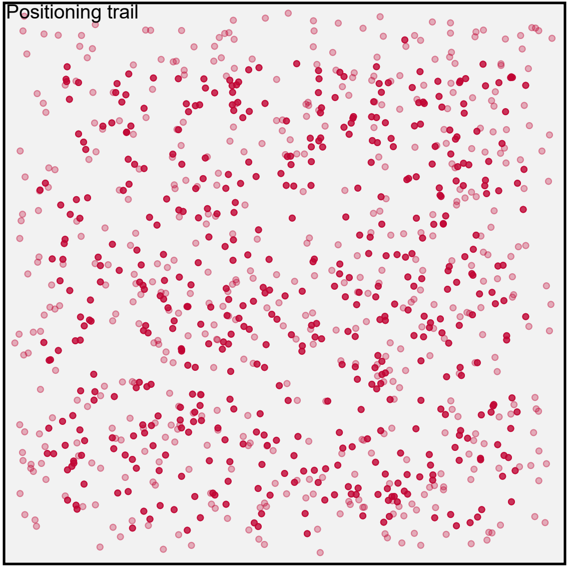
Spring forces to target connected node distances and repulsion



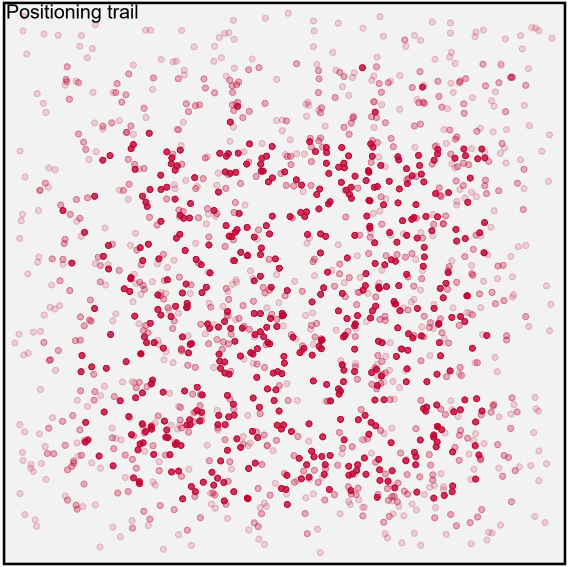
Example for a beeswarm plot



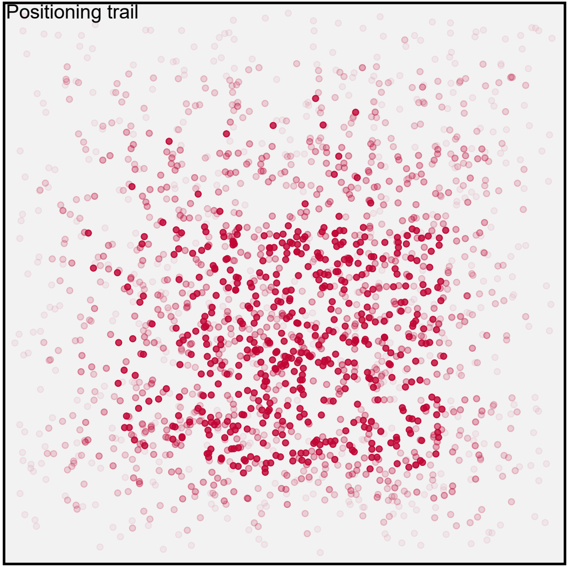
Example for a beeswarm plot



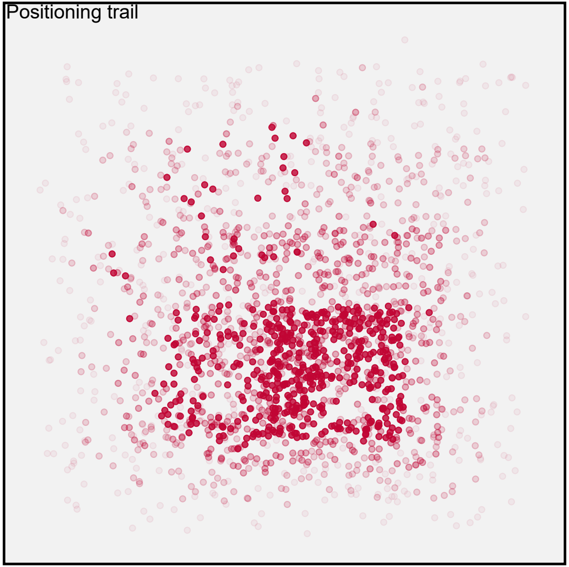
Example for a beeswarm plot



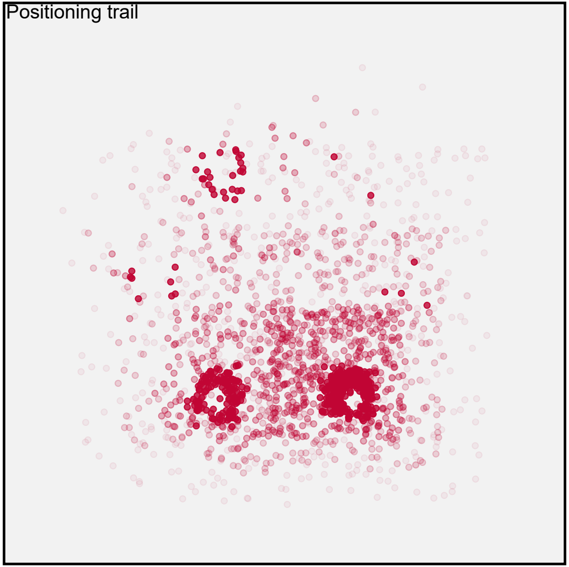
Example for a beeswarm plot



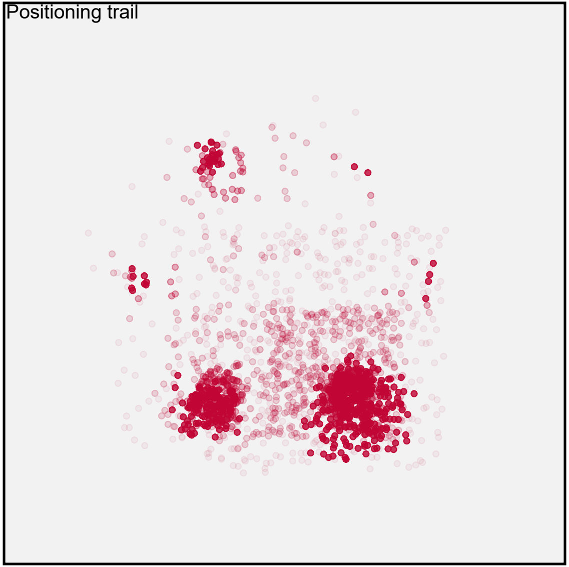
Example for a beeswarm plot



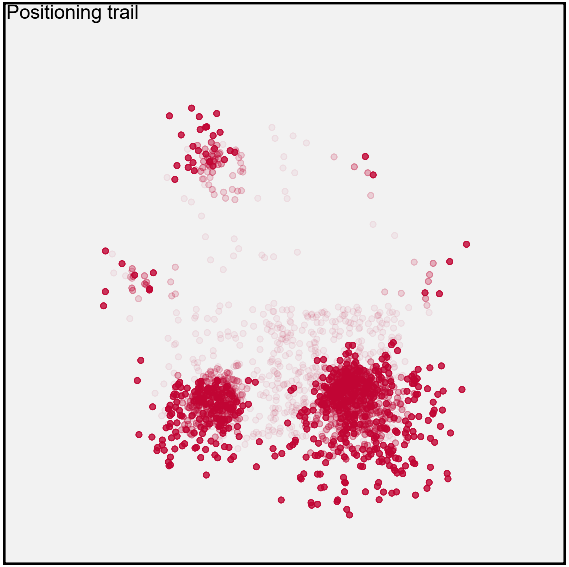
Example for a beeswarm plot



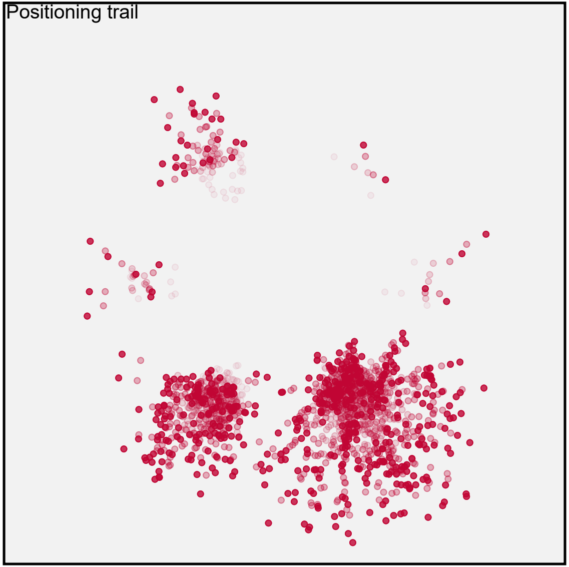
Example for a beeswarm plot



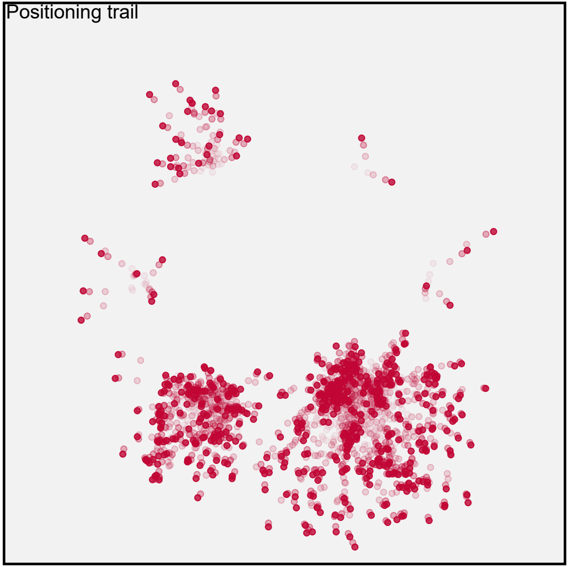
Example for a beeswarm plot



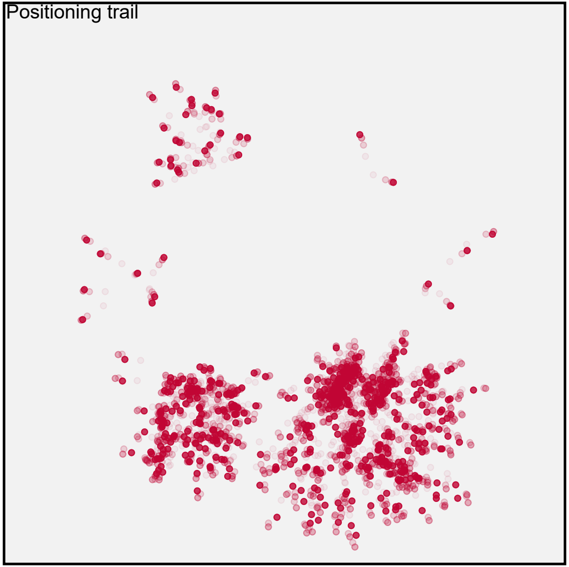
Example for a beeswarm plot



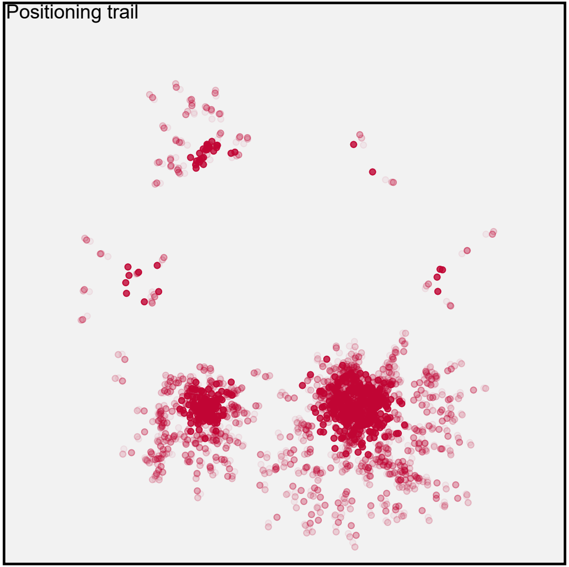
Example for a beeswarm plot



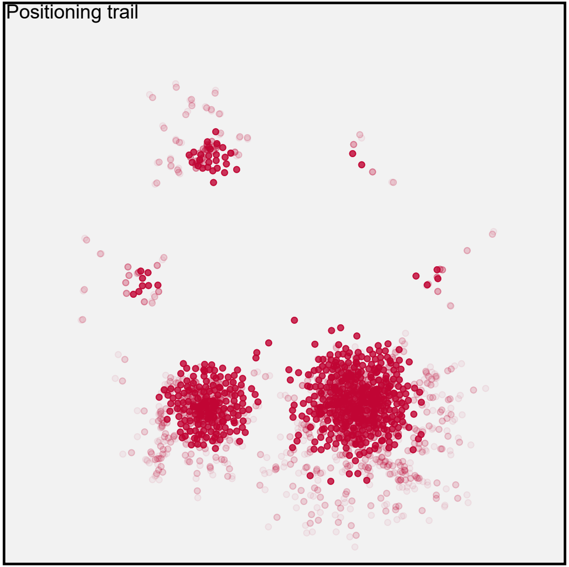
Example for a beeswarm plot



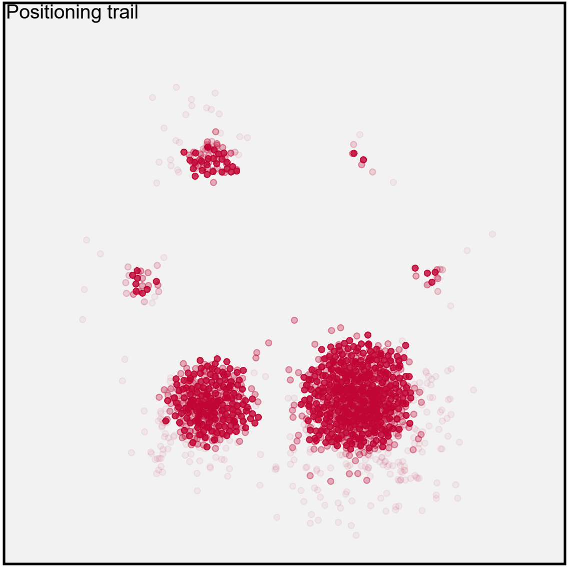
Example for a beeswarm plot



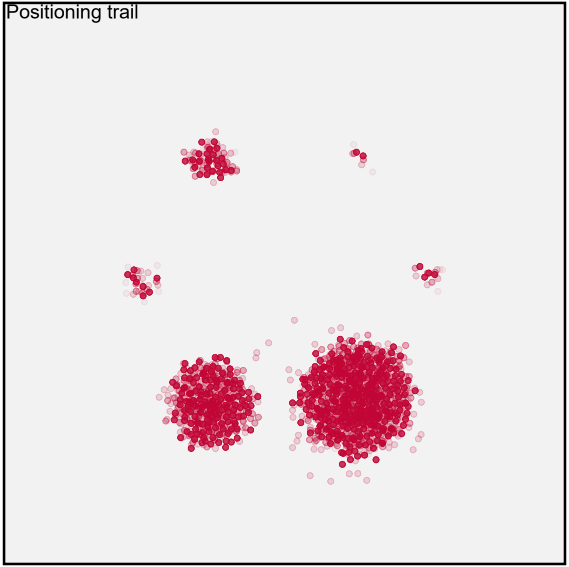
Example for a beeswarm plot



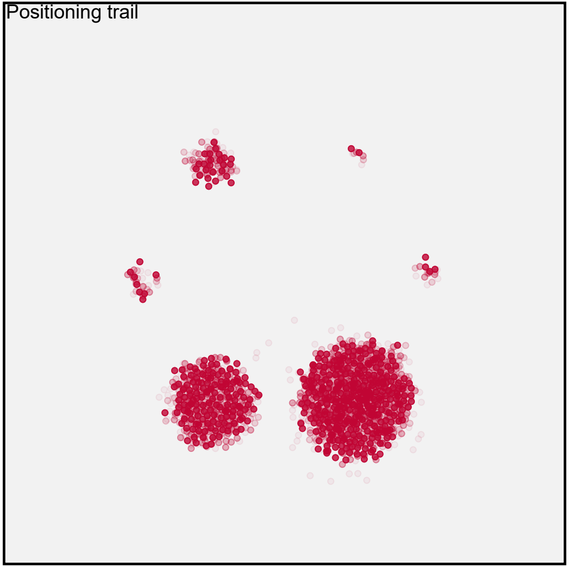
Example for a beeswarm plot



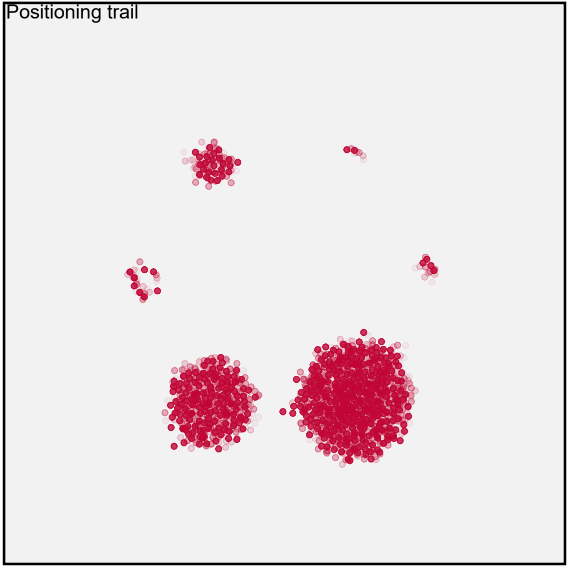
Example for a beeswarm plot



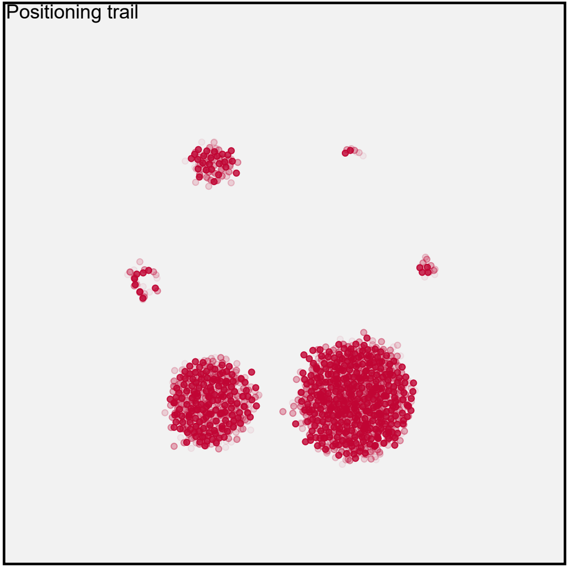
Example for a beeswarm plot



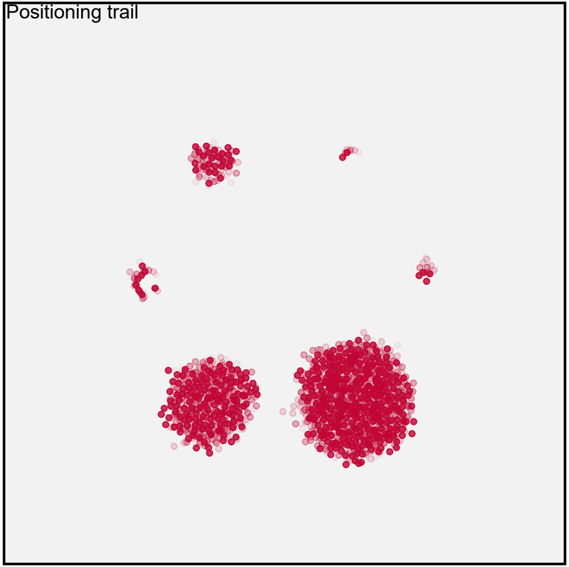
Example for a beeswarm plot



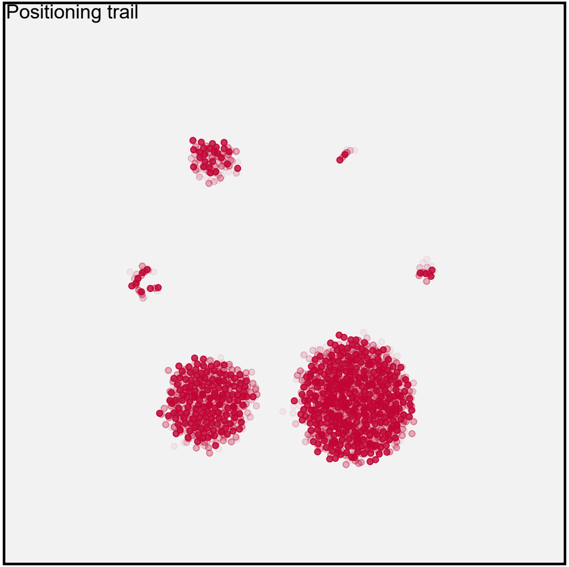
Example for a beeswarm plot



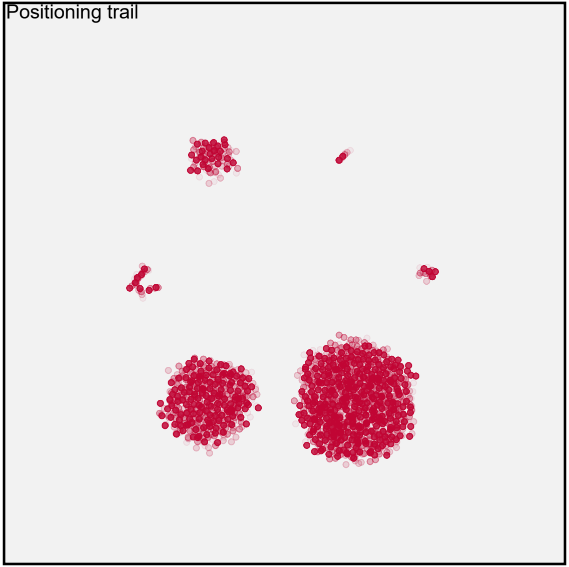
Example for a beeswarm plot



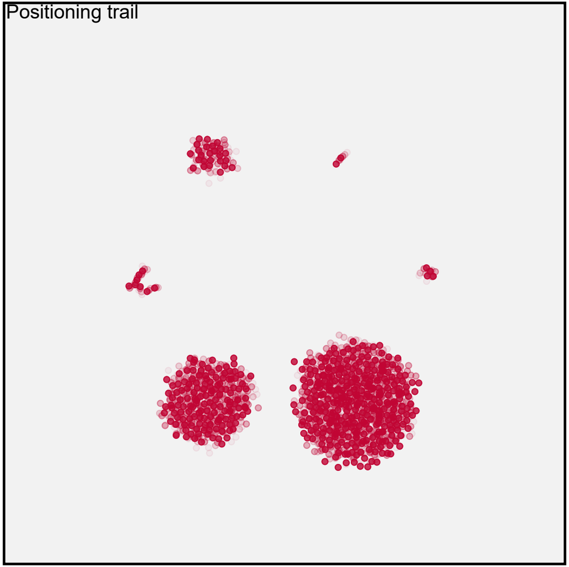
Example for a beeswarm plot



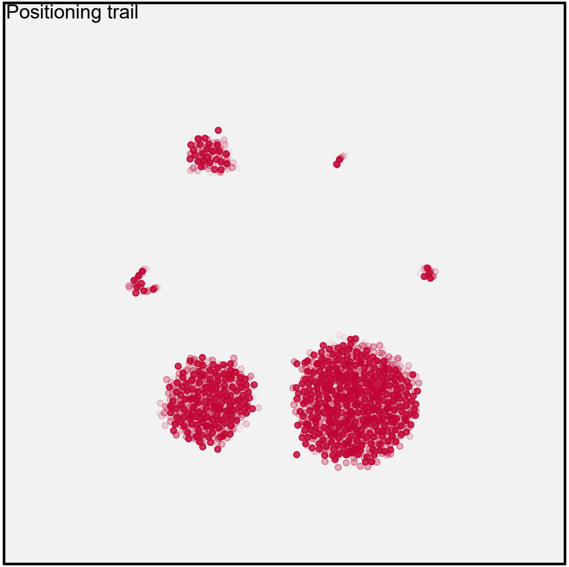
Example for a beeswarm plot



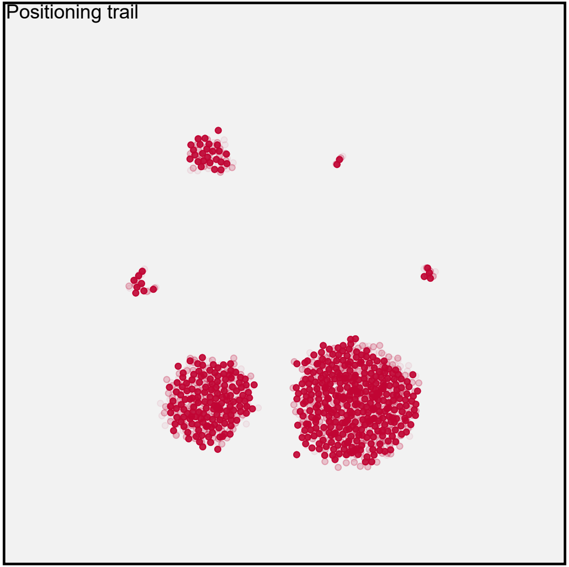
Example for a beeswarm plot



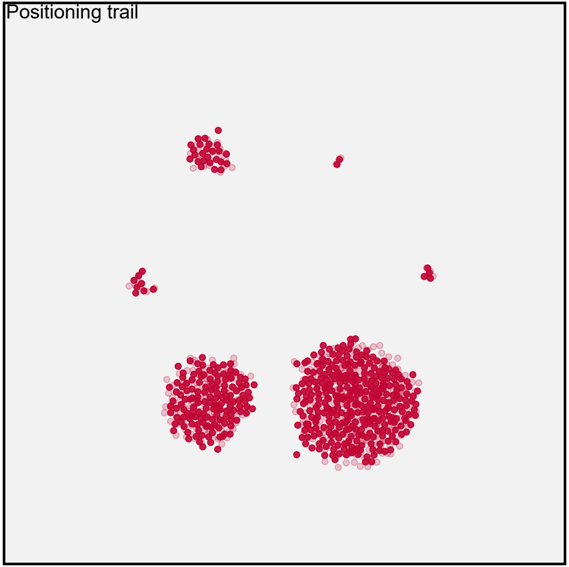
Example for a beeswarm plot



Example for a beeswarm plot



Example for a beeswarm plot



Examples

Principles and mechanics

Implementation

Implementation

beeswarm

```
... Stata code ...
      parsing data ...

... Mata call ...
...   pass views  ...
...   create an instance of a
     Swarm class ...
...   run simulation (Swarm.fly
     ()) ...

... Stata graphics code ...
...   clear canvas ...
...   scatter and pcspike ...
```

fdnetgraph

```
... Stata code ...
      parsing data ...

... Mata call ...
...   pass views  ...
...   create an instance of a
     Swarm class ...
...   run simulation (Swarm.fly
     ()) ...

... Stata graphics code ...
...   clear canvas ...
...   scatter and pcspike ...
```

Generic component

```
  Mata library defining a Swarm class
```

Main variables and functions of the Swarm class

The Swarm class is the engine which calculates the element positions and pass it back to calling ado file.

```
class swarm {
  // Variables:
  real matrix  BeePos    //
      positions
  real matrix  BeeVel    //
      velocities (movement)
  real matrix  BeeAcc    //
      accelerations
  real matrix  BeeLinks  //
      connections
  class params scalar Params //
       long list of force
      parameters
  ...
  // Functions:
  ...
  }
```

```
class swarm {
  // Variables:
  ...
  // Functions:
  void initialize()
  void fly()
  real matrix centeringForce()
  real matrix flyinghomeForce()
  real matrix
      peerbeeteractionForce()
  ...
  }
```

Pros and cons of Stata here

👍

- Great combo:
  - Stata for handling and parsing source data
  - Mata for handling calculations/simulations
- Mata (class) programming is neat
- `twoway graph` commands flexible (more than they may seem)

👎

- `twoway graph` can be impractical ...
  - Controlling graph element dimensions (Aaargh!)
  - Plotregion dimension as residual (Ouch!)
  - Marker dimensions with weights??
  - (NB: no graph class digging—higher-level `twoway graph` commands only)
- Animation (and interaction) gives force-directed graphs another dimension —`https://flowingdata.com/2019/03/06/women-men-timeuse/` (see p5 or d3)

Thanks!

Comments and suggestions welcome.

`beeswarm` and `fdnetgraph` will be ‘released’ in the coming weeks/months (it needs a bit of fine-tuning and documentation!)