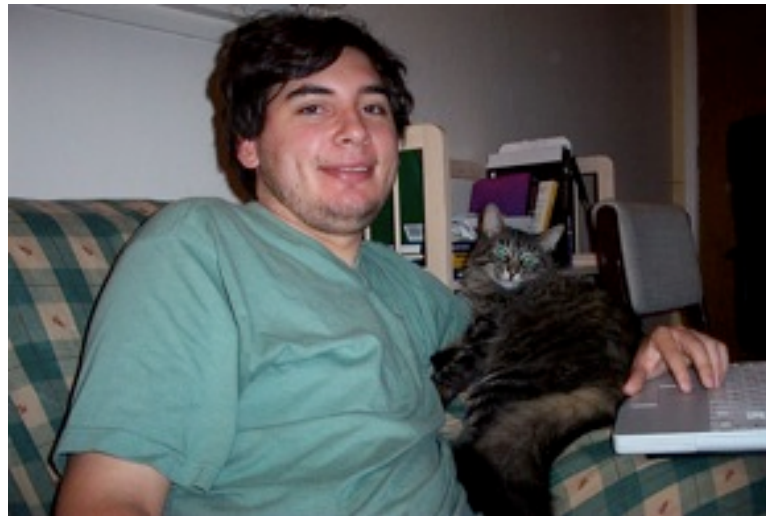


Network Visualization Using Processing: lucid sketches of elusive mechanisms, fast!

Sergiy Nesterko
Statistics Department
Harvard University

Connecting The Dots, Oct. 22, 2010

Based on joint work with Joe Blitzstein



Outline

- What is respondent-driven sampling (RDS)?
- Networks with homophily
- Visualization of RDS on different networks
- Processing remarks and some code snippets
- Conclusion

Respondent-driven sampling (RDS)

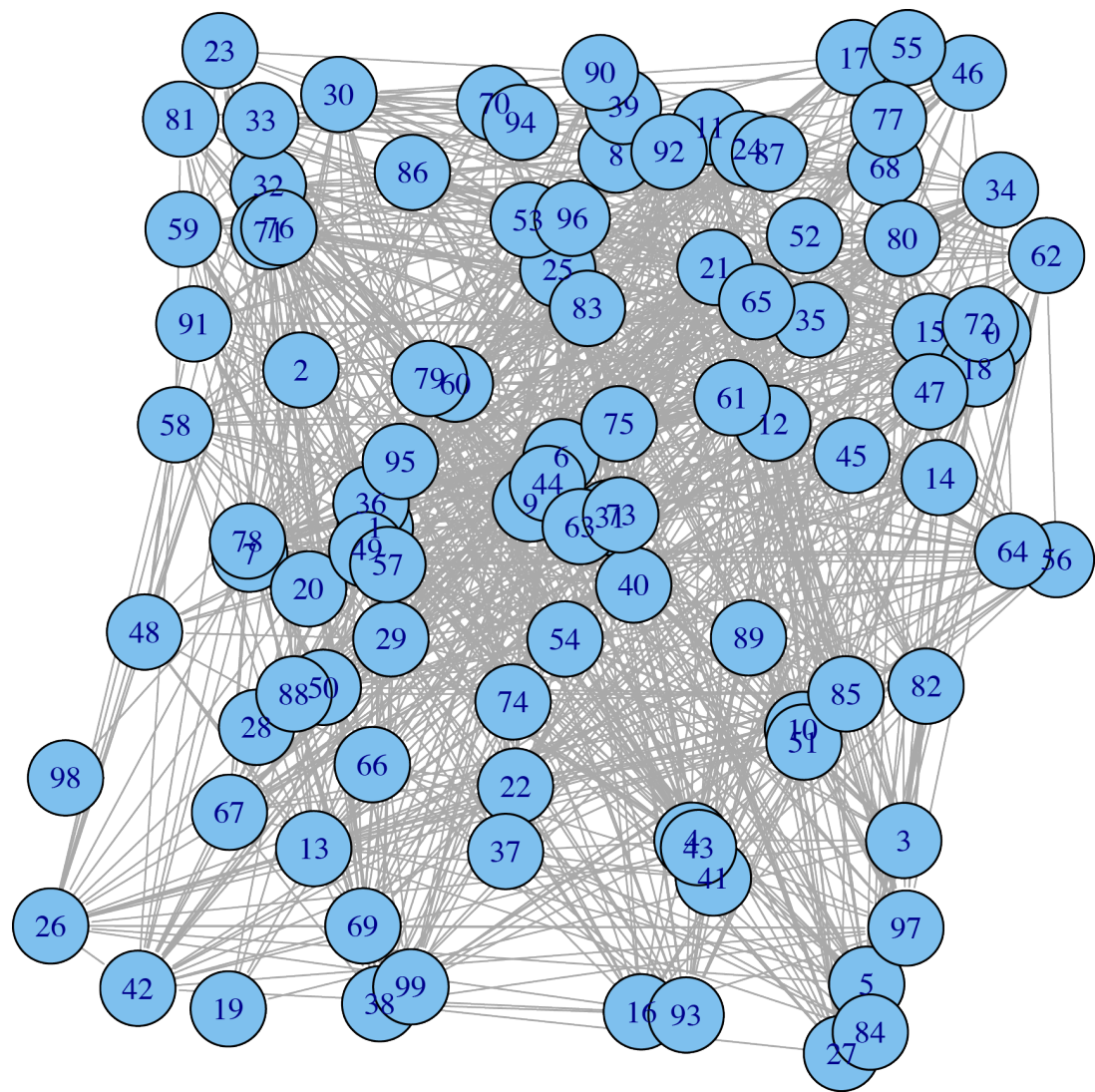
- Access hard to reach populations (homeless people, injection drug users, HIV infected individuals ...)
- Find a convenience sample, give them coupons and via financial incentive encourage to recruit friends to participate
- Goal: estimate population mean using collected data (e.g., mean income, proportion of homeless within injection drug users ...)

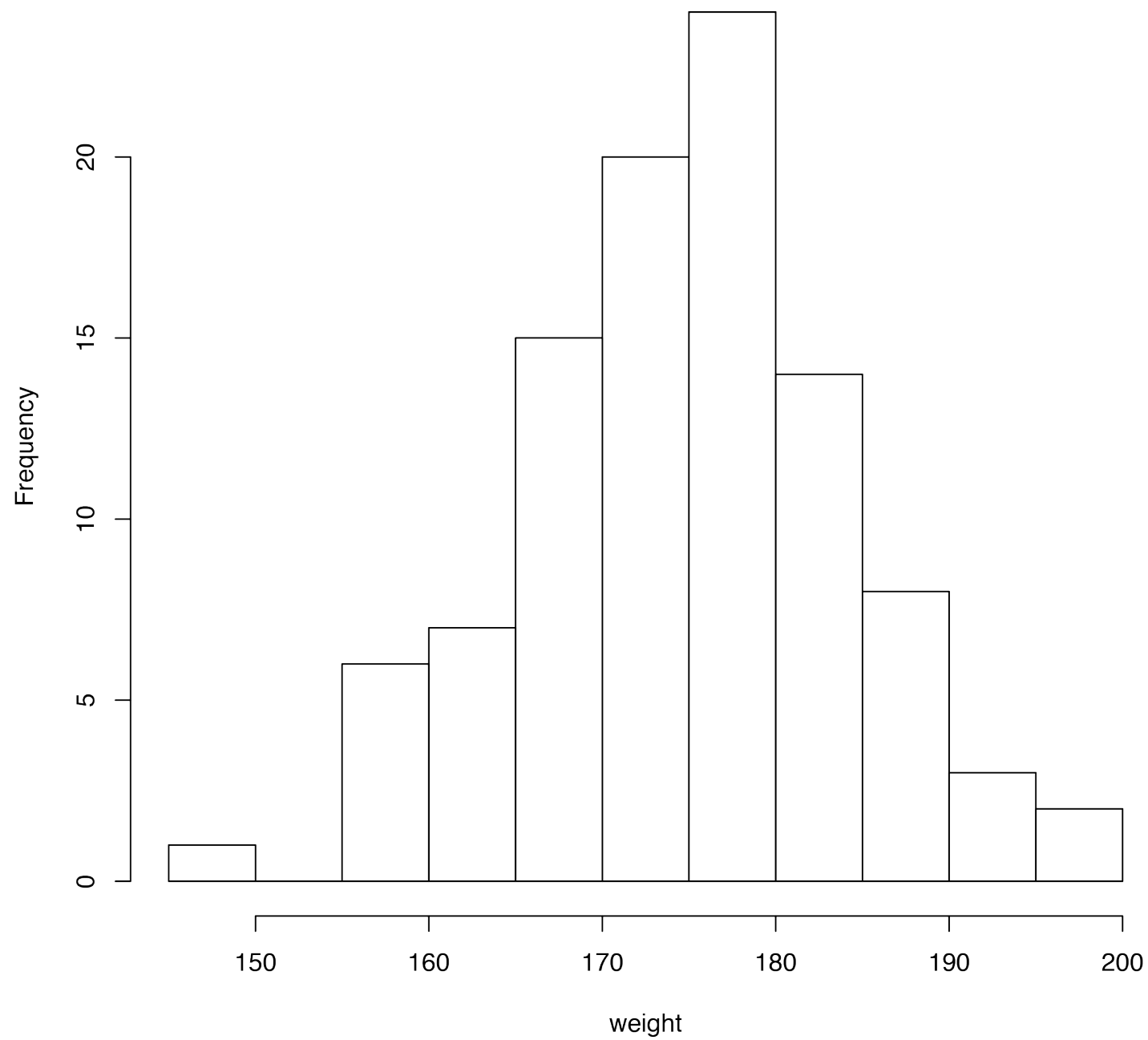
RDS features

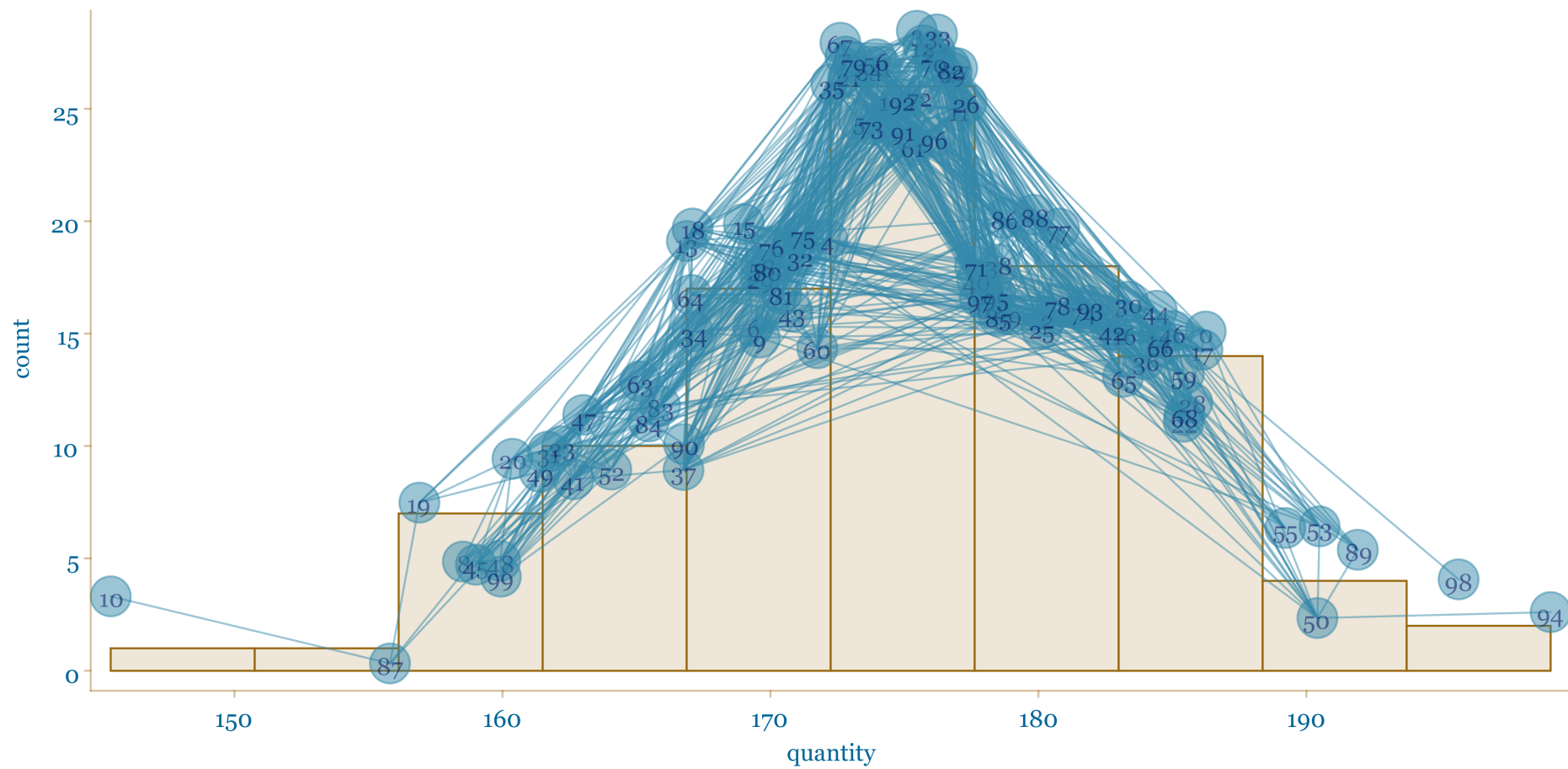
- Number of coupons
- Probability that a respondent won't use all coupons
- Different network topologies
- Different preferences for recruitment
- *How to make sense of all of them?*

Homophily

- Wiki: **“Homophily (i.e., love of the same) is the tendency of individuals to associate and bond with similar others.”**
- i.e., “Birds of a feather flock together”
- We study Respondent-Driven Sampling (RDS) on networks with varying degrees of homophily







Switch to dynamic visualization!
(not in the slides, will put it online at
incontemplation.com soon)

Development features

- About 2.5K lines of code, roughly 2 weeks part-time to develop without prior Processing knowledge
- Use classes, power of inheritance
- More verbose than R or Python
- Recyclable code/classes
- Ability to export

Some code

```
public class NetVertex extends PlottingObject{
    public float ypos, xpos;
    public float yposPX = 0;
    public float xposPX = 0;
    public float wdh, hght;
    // color settings in superclass!
    // This is to keet track of all the other vertices on the network
    public boolean needVertexInfoUpdate;
    public NetVertex[] allVertecis;
    // record how far from other vertices this one wants to be
    public float mindistToOthersPX;
    // a flag to know that this one is currently moving
    public boolean movingNow, stopped;
    public NetVertex (float x, float y, float wide, float high, float mindistPX,
        float opacity, float speedOfMovement, int col, int textcol,
        int strokeWidth, int theID, PlottingArea pa)
    {
        // update pixel coordinates from PA coordinates
    }
    public void updateposPX () {
        // update PA coordinates from pixel (PX) coordinates
    }
    public void updateposFromPX () {
    }
    public void plot () {
        // update the PA coordinates from pixel coordinates, very important!
        // This is because when we move, we change the pixel coordinates,
        // not the PA coordinates
        updateposFromPX();
        // text settings and text
        ellipseMode(CENTER);
        fill(colour, al);
    }
}
```

Conclusion

- Visualization allows us to make research happen faster
- For example, we see that under inverse homophily the process jumps from tail to tail of the distribution
- Provides additional motivation
- Creating compelling and helpful visualizations is not that hard (just a little!)
- Network visualization is still a vastly unexplored field, and needs *our* input

Thank you

- My advisor Joe Blitzstein
- Audience
- Ben Fry/creators of Processing

stat.harvard.edu; incontemplation.com