Social Networks: A Fast Tour From People to Groups!

Matthew A. Hoover

02 May 2017

Welcome!

- Data scientist at Gallup
- Ph.D. in public policy
  - Dissertation on the effects of children's social networks on education in rural Afghanistan
  - Research on how social networks affected individual decision-making
- Previous life, 15+ years in international development along with additional work for a healthcare startup

What is social network analysis?

- Understanding the structure, composition, and purpose of people's social networks, whether in-person or online
- It helps answers questions from "how do my friends and acquaintances affect my behaviors" to "from whom can I seek support in a given situation"
- **Structure** identifies how ties connect people in certain ways – are there mutual ties, triangles, cycles?
- **Composition** describes the characteristics of people that are connected – are they the same gender, about the same age?
- **Purpose** of a particular network varies – is it a support network, drug seeking/using network, professional network?

Let's start small: The individual

- The individual plays the key role in most econometric analyses

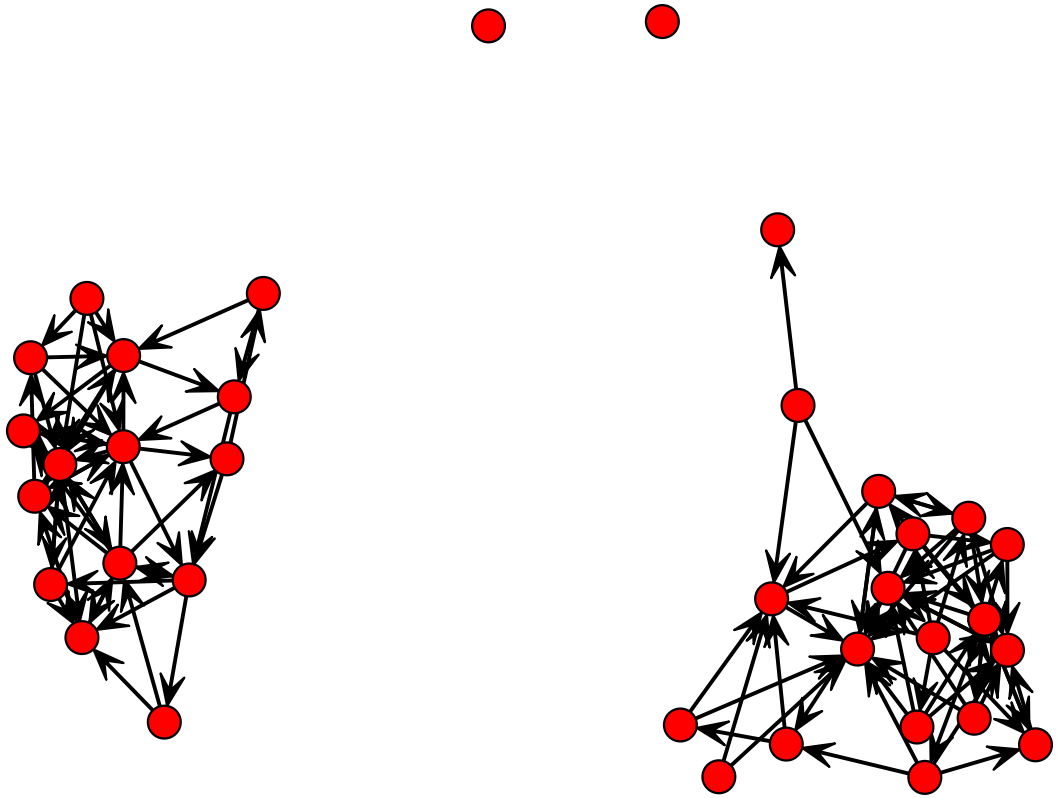
Types of analysis (thank goodness for computers!)

- Early network analysis (circa 1920s) was done by hand on small networks (10-12 people)
- With computers, our reach is greatly expanded:
  - Descriptive statistics
  - Dyadic models
  - Exponential random graph models
  - Stochastic actor oriented models
- Underneath it all though, visualization of networks has been an important analysis tool since the beginning

Visualization – the problem

- The problem with network visualization is quite simple: there are programs that do good visualization (`Gephi`, `Visone`) and there are programs that do good statistical work (`R`), but there were really none that did both (`Python` is starting to change this)
- This posed a problem to me, as I had 62 networks I needed to visualize for my dissertation and I didn't want to use a lot of different programs to accomplish my tasks (data management, analysis, visualization)
- Since each network can be thought of as a square matrix of 1's and 0's, I opted to figure out a solution inside `R` or with a program that `R` could plug into
- `Visone` had an `R` plug-in, but due to changes in `R` with release 3.0, that no longer worked
- I was stuck with visualizations in `R`, which left me with this:

Visualization – the problem



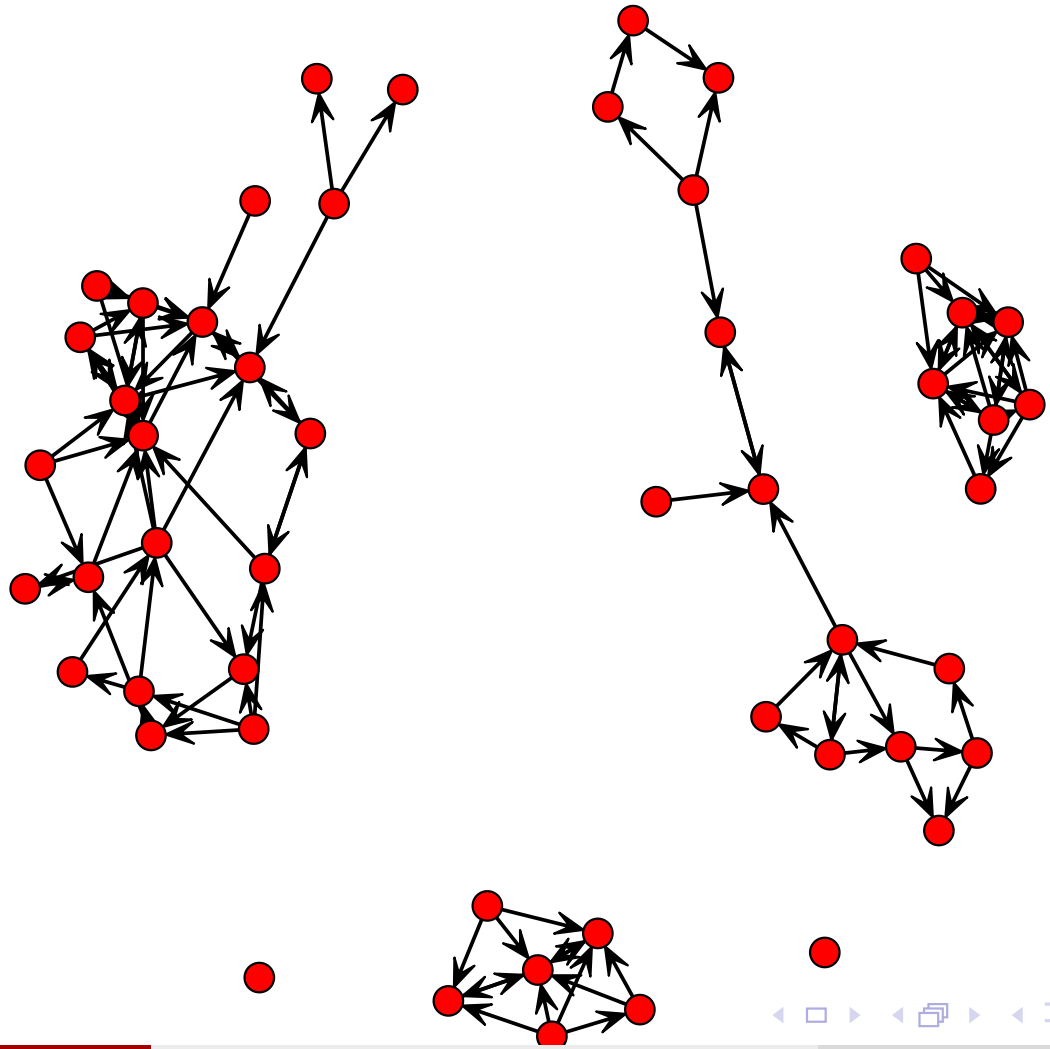
(The start of) a solution

- I was unhappy with a number of things:
  - The color palette was jarring
  - The resolution and plotting area was quite limited
  - Controlling attributes (size/color of nodes) was difficult
  - Most importantly, I could not "set" node coordinates to replicate a different network with the same actors
- As I thought about solutions, I was using the `ggplot2` package in other work, so I thought I would try and create a `ggplot2`-like visualization for networks
- Let's look a bit more at the native and my `ggplot2`-inspired visualization

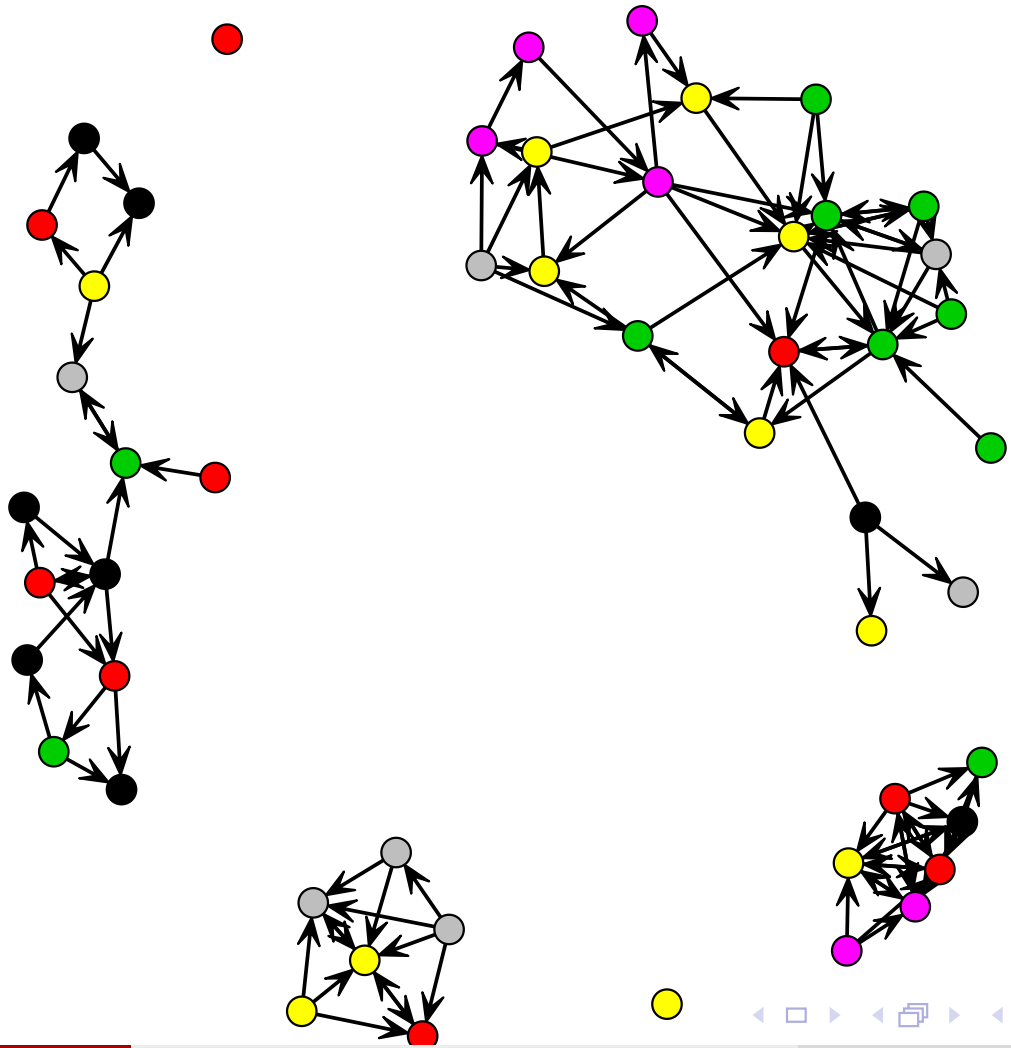
Native R plotting: gplot

- Uses R's base graphing features, layout, and language
- Limited color palette (8 colors), then it will recycle through colors
- Adding legends is difficult
- No way to capture/fix node coordinates
- However, very flexible in the format data can be fed to the function
- Let's look at it within R

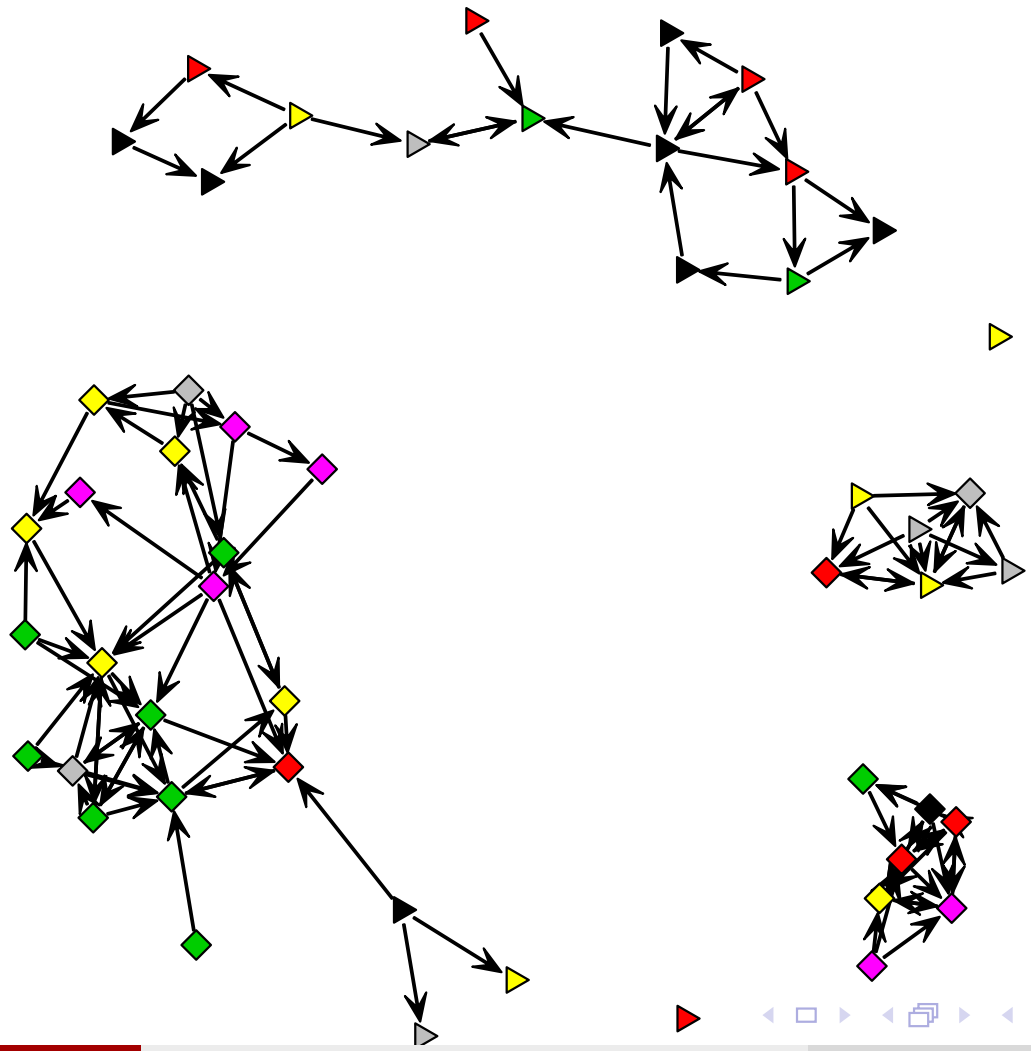
gplot: Basic plot



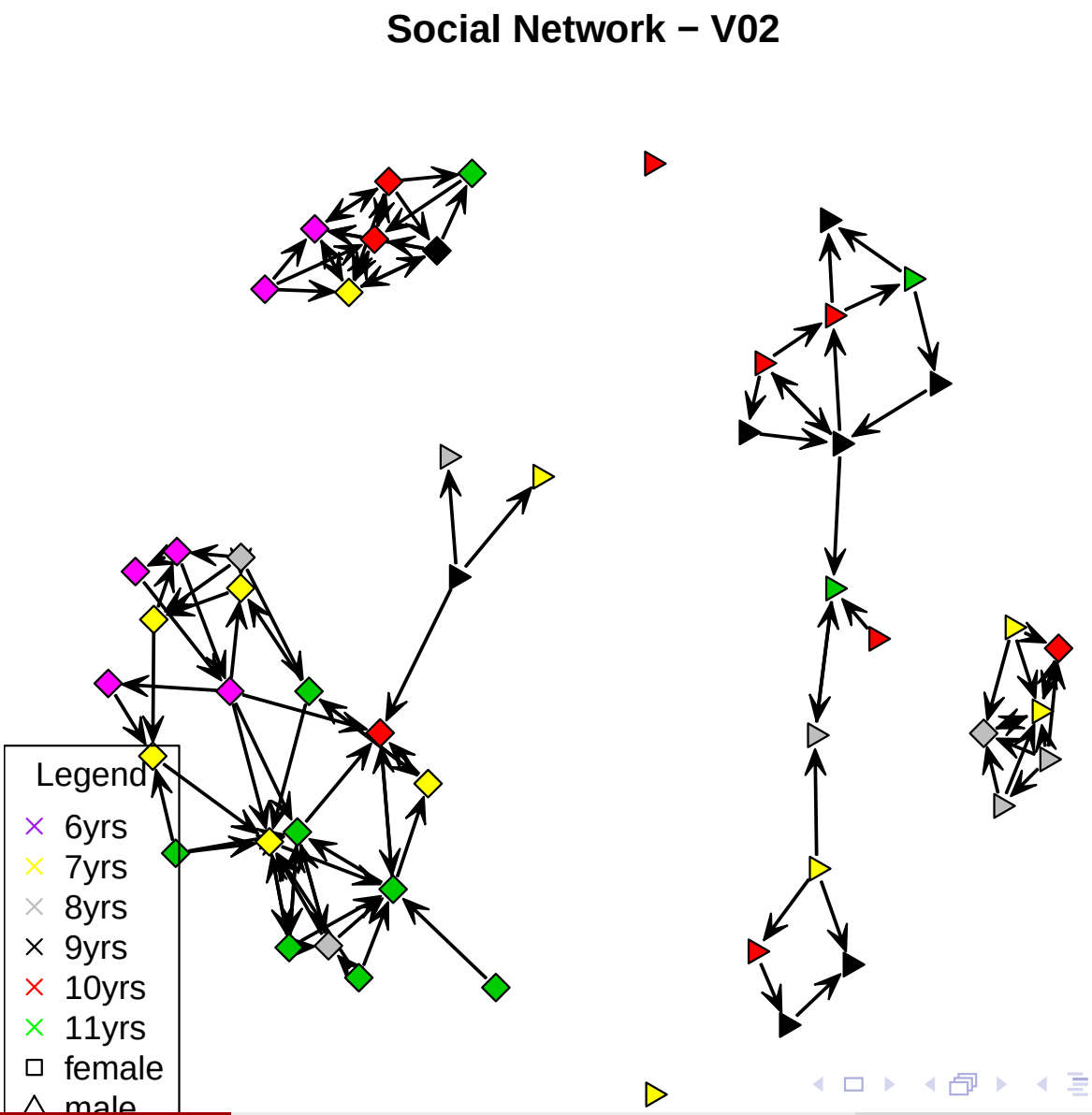
gplot: Colored nodes



gplot: Colored and shaped nodes



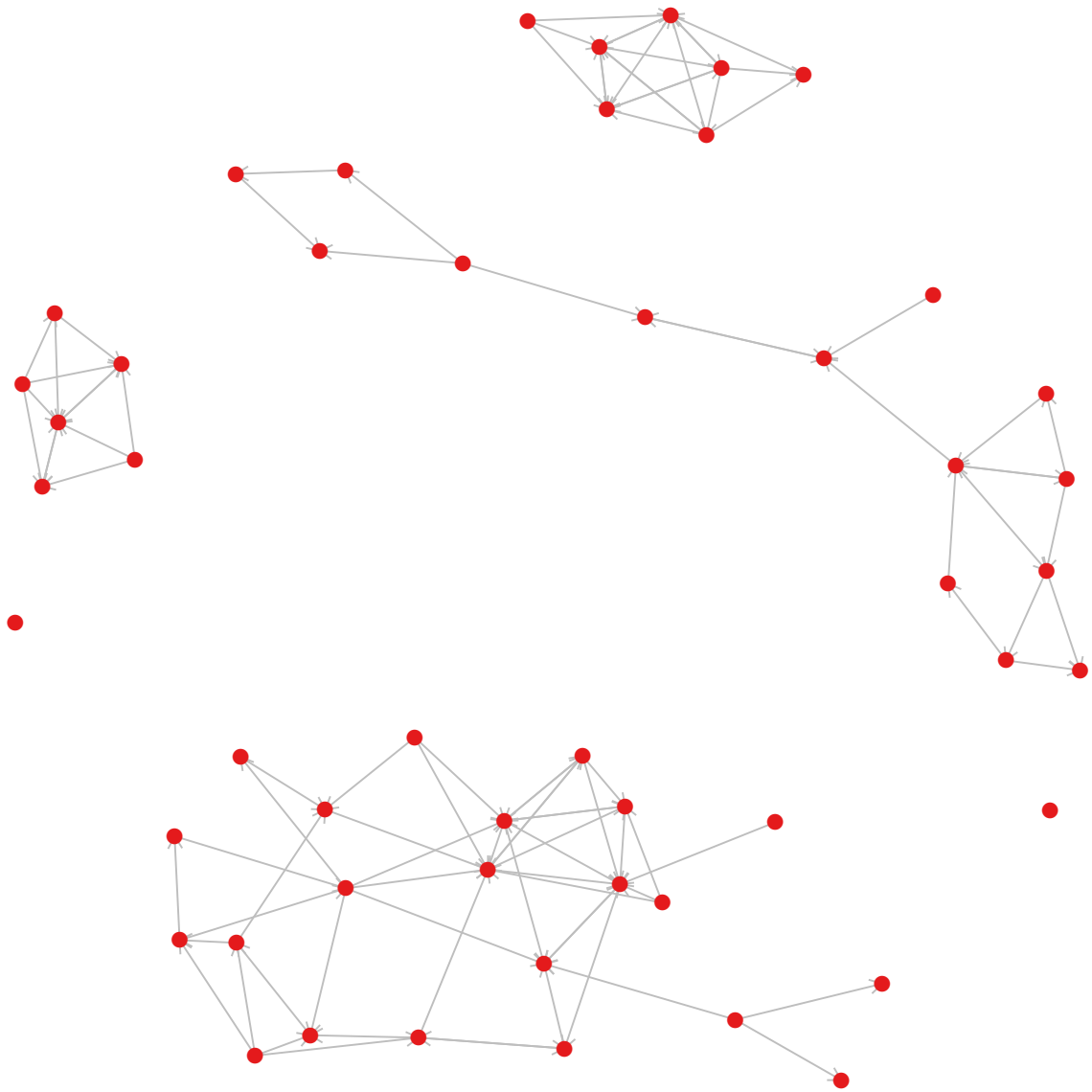
gplot: With legend



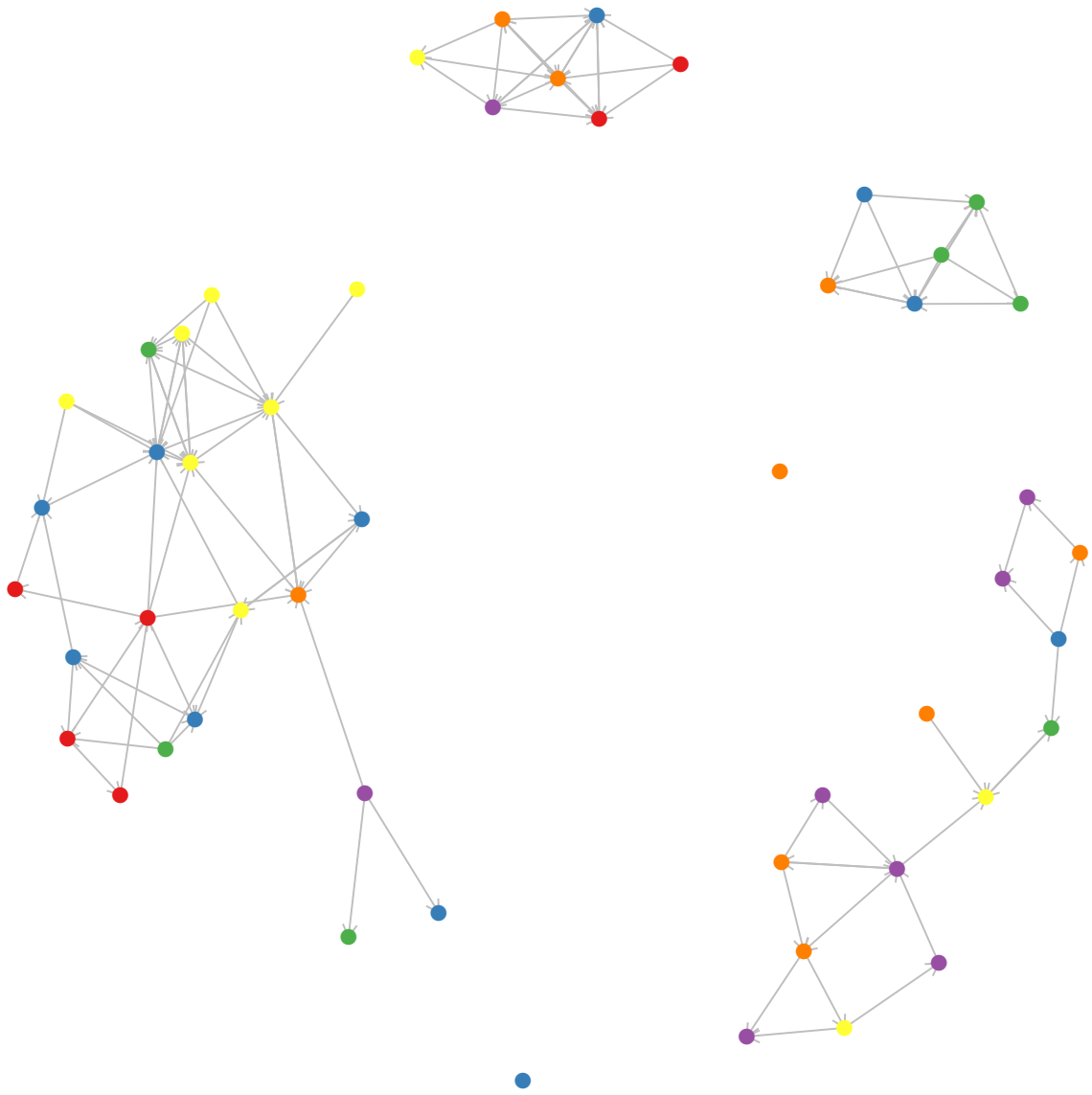
Plotting with ggplot2: ggnet

- Utilizes `ggplot2` for setup and graphing
- Code builds on component pieces of `ggplot2`, such as lines, points, axes, and legends
- Can specify coordinates for nodes or use random layout
- Legends auto-generated
- Variety of color palettes and objects types can be used for attributes
- Requires a specific data object class (`network` class)
- Let's have a look within `R`

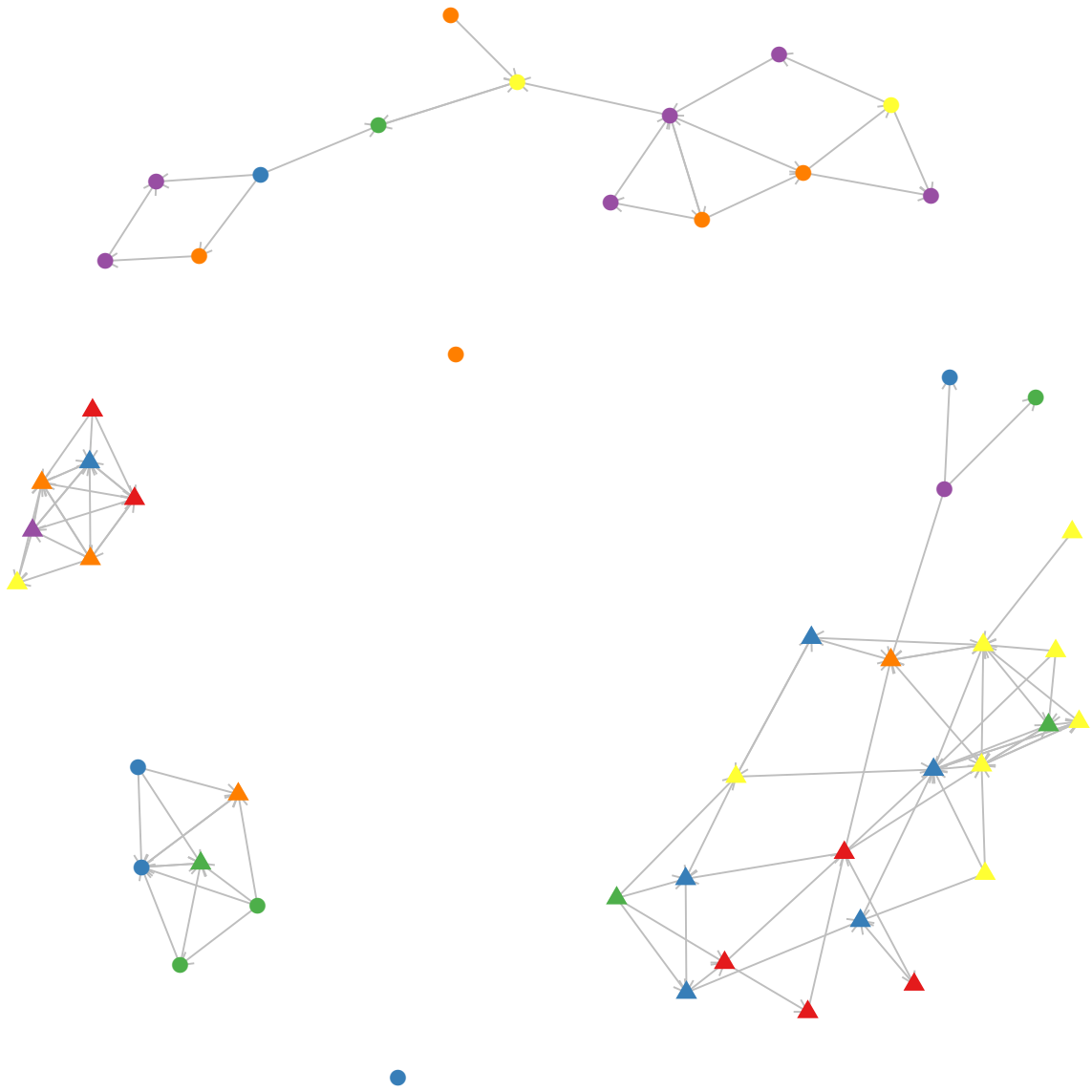
ggnet: Basic plot



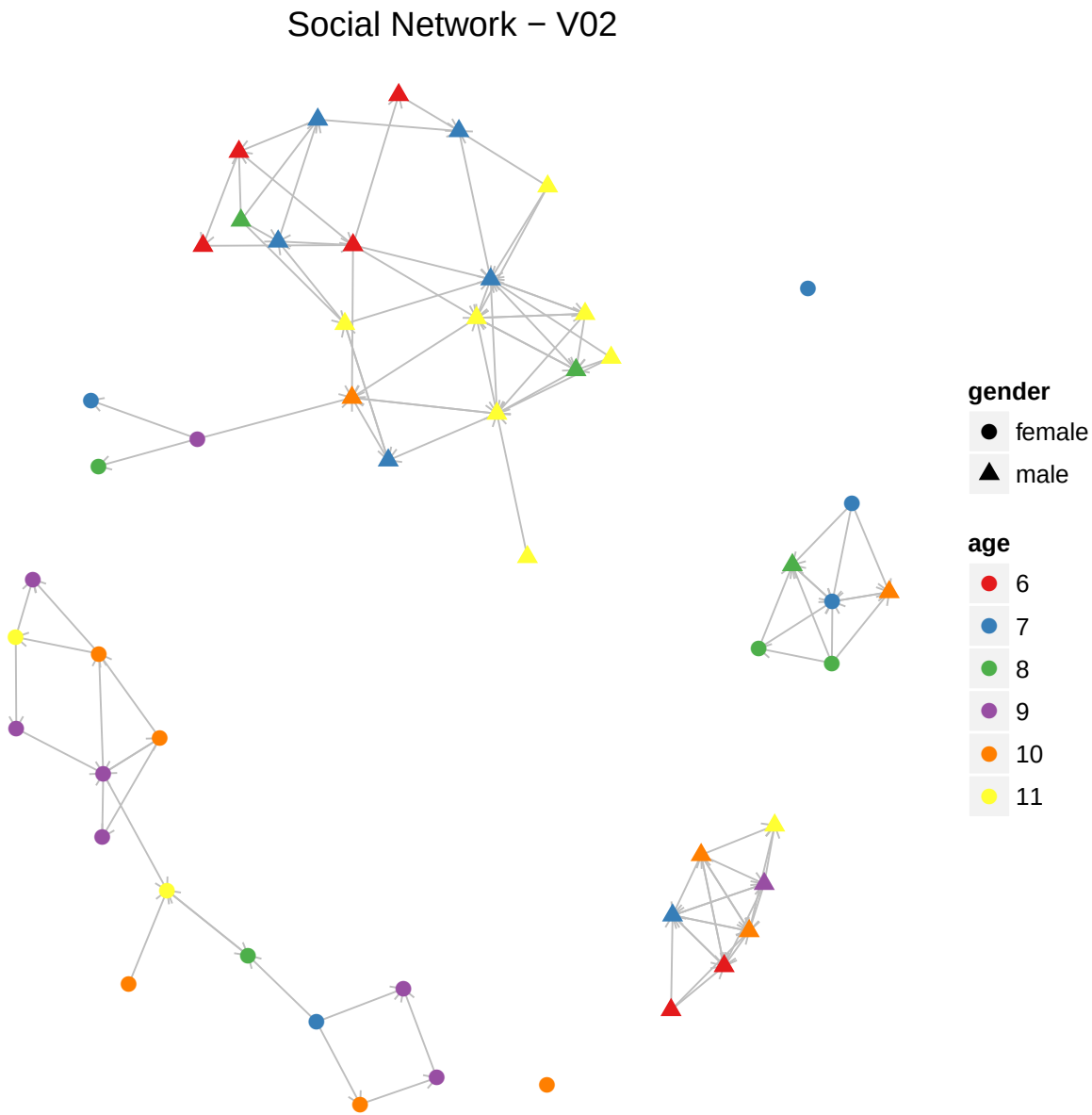
ggnet: Colored nodes



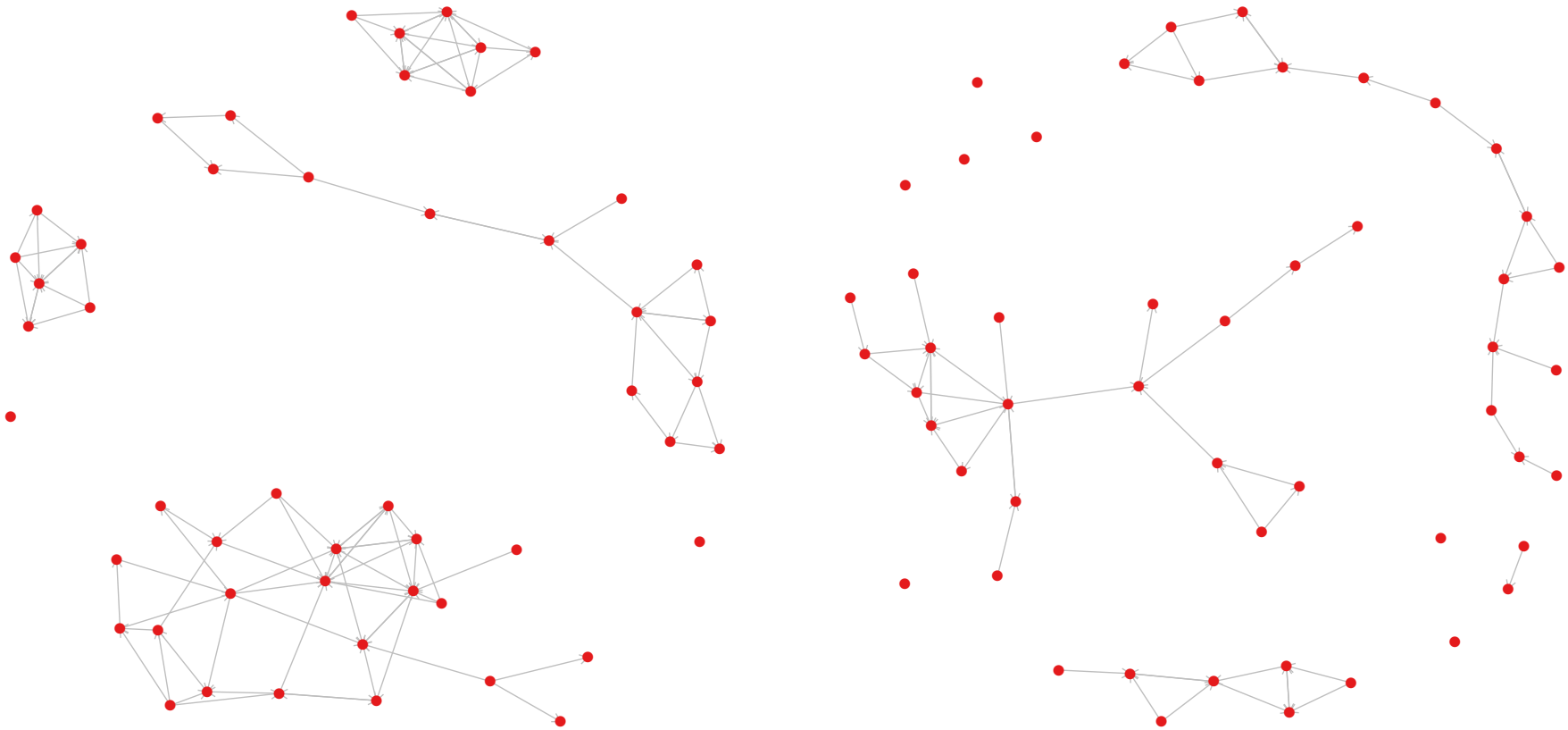
ggnet: Colored and shaped nodes



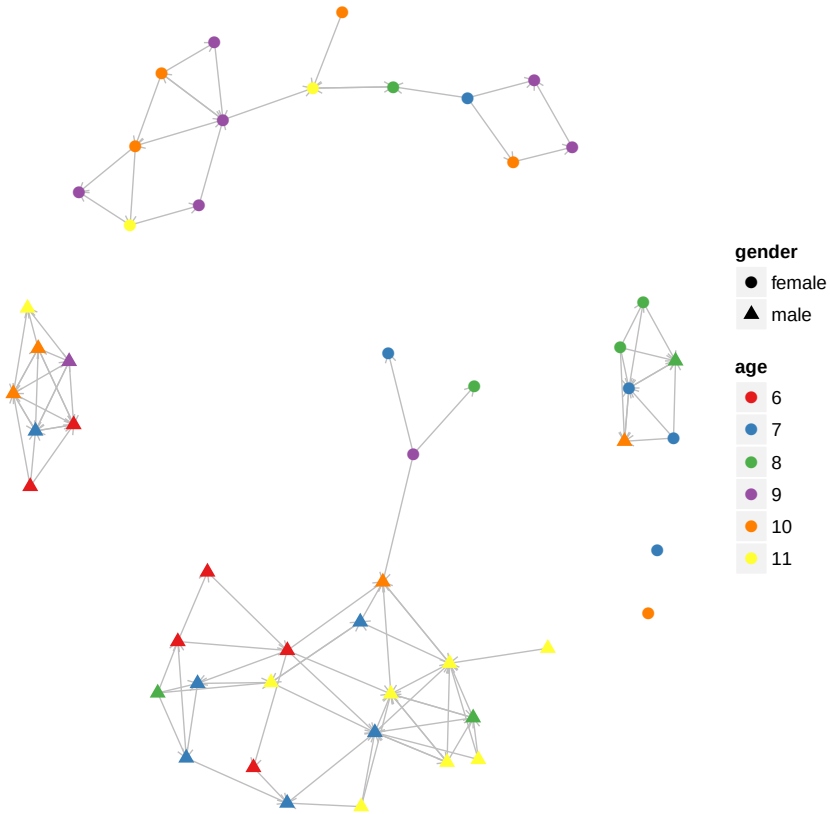
ggnet: With legend

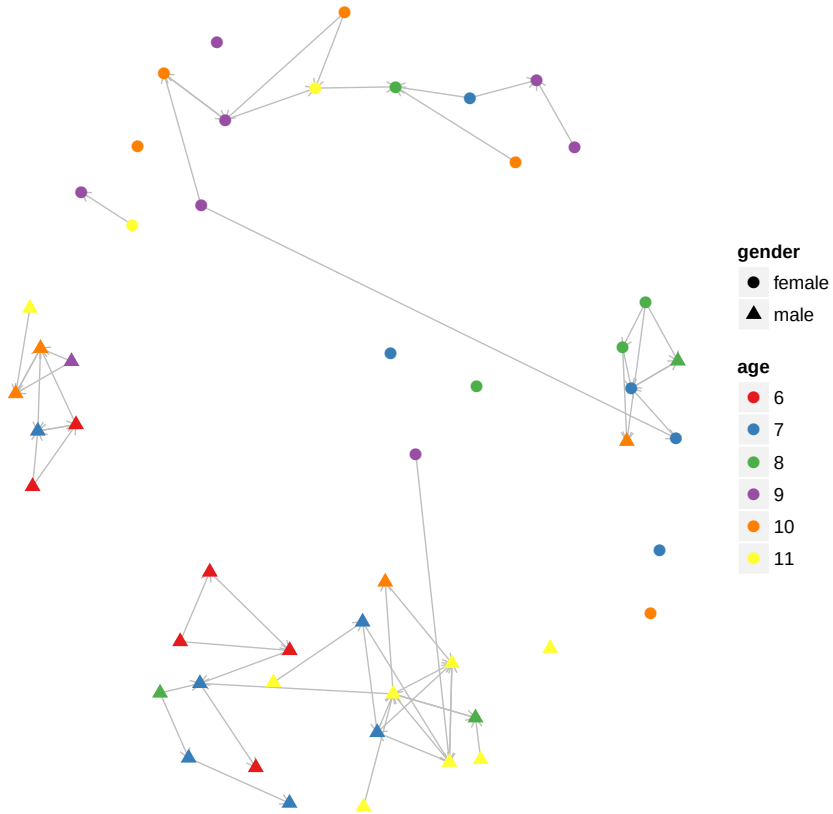


ggnet: Same network, different ties

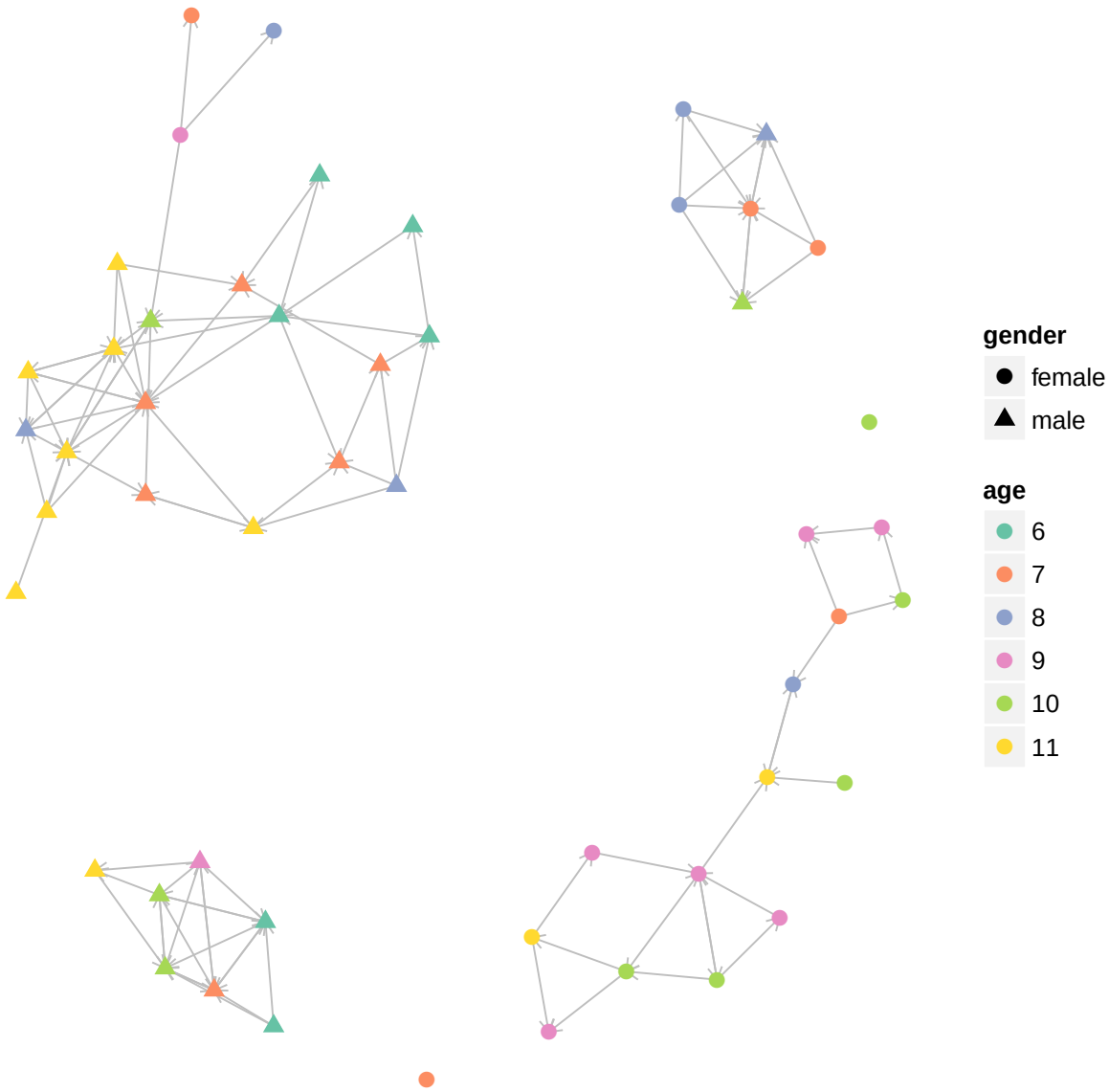


ggnet: Same network, different ties, node placement set

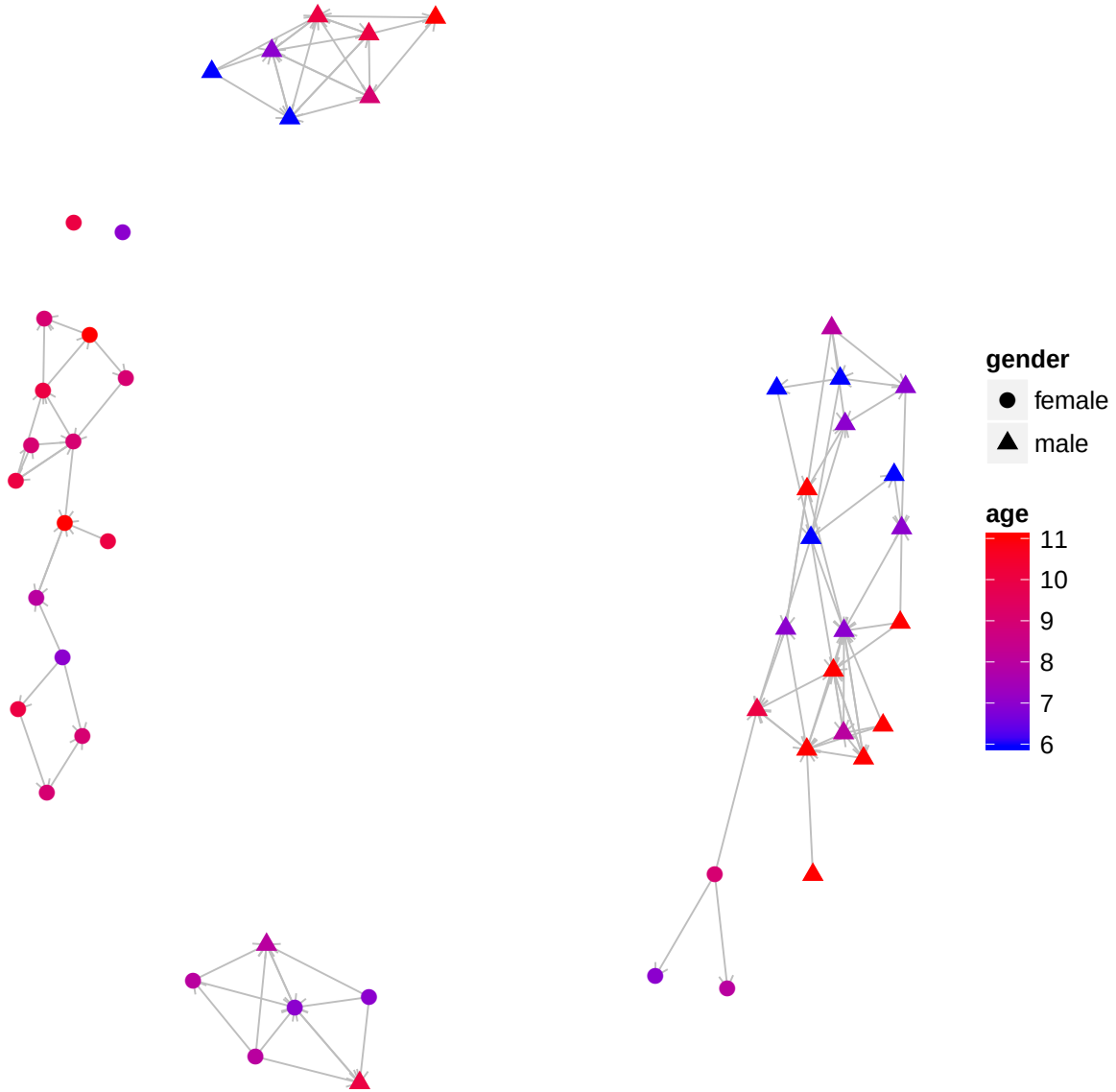




ggnet: Different color palette



ggnet: Color gradient



Comparing and contrasting gplot and ggnet

- gplot has better flexibility for input; ggnet requires a specific data source
- ggnet has streamlined parameter entry
- gplot has a greater variety of attributes for nodes and edges
- ggnet has easier legend capabilities
- gplot can handle one- and two-mode networks; ggnet does not have the capability, currently
- ggnet allows for color gradient on continuous variables
- Both have extensive outputting capabilities for use outside of R

Summary

- Do it yourself! I had a need and while the solution isn't perfect, it was fun to put something together and keep working on it
- Next steps:
  - Build out the two-mode features of `ggnet`
  - Consolidate and create more efficient code
  - Create an `R` package for submission to CRAN
- Which function to use depends on need – `gplot` for initial visualizations, `ggnet` for final visualizations

Thank you!

- Thank you for your time!
- Email: matthew.a.hoover@gmail.com
- Questions?