Close relationships: assortativity & reciprocity

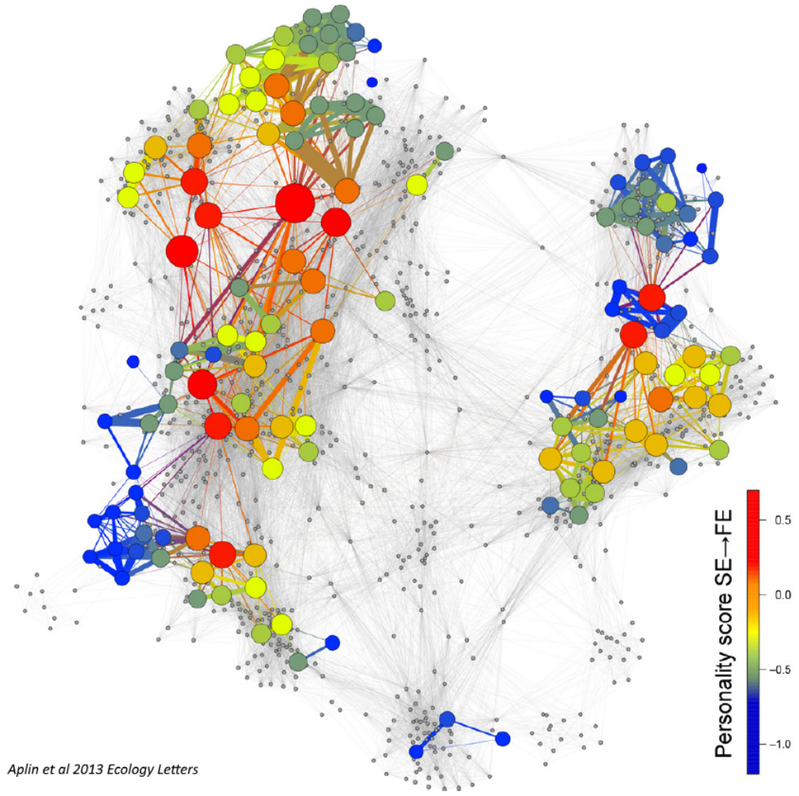
NETWORK ANALYSIS IN R

James Curley
Associate Professor, University of Texas at Austin

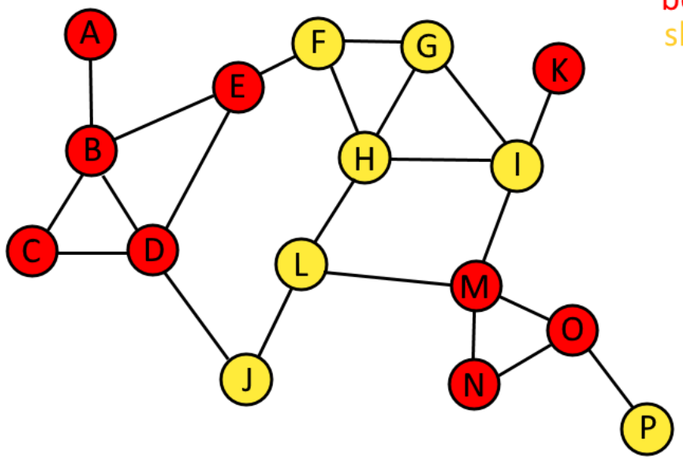
datacamp

Assortativity

The preferential attachment of vertices to other vertices that are similar in numerical or categorical attributes.



Assortativity



bold
shy

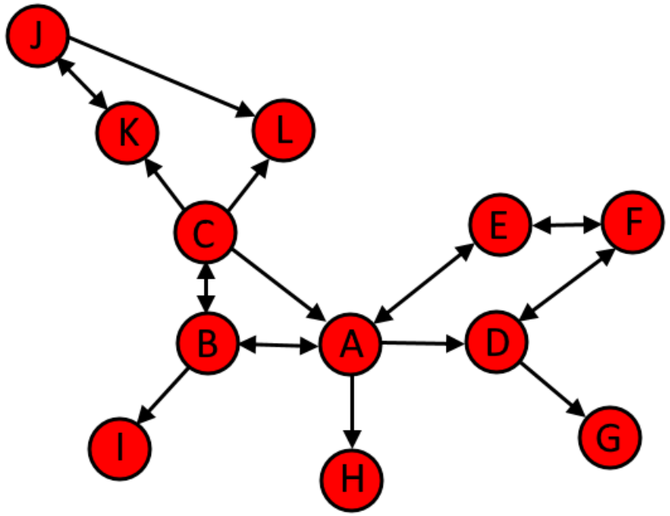
```
assortativity(g, values)
```

```
0.45
```

```
assortativity.degree(
  g,
  directed = FALSE
)
```

```
-0.31
```

Reciprocity




```
reciprocity(g)
```

```
0.6
```

Let's practice!

NETWORK ANALYSIS IN R

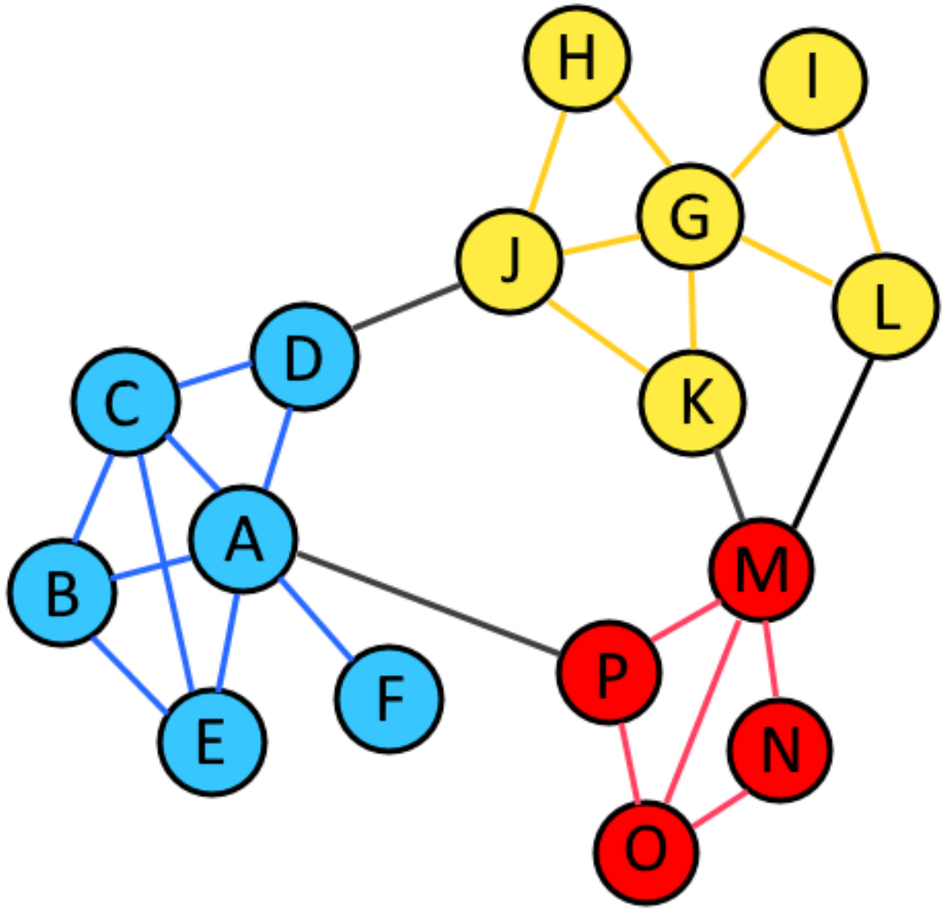
Community detection

NETWORK ANALYSIS IN R

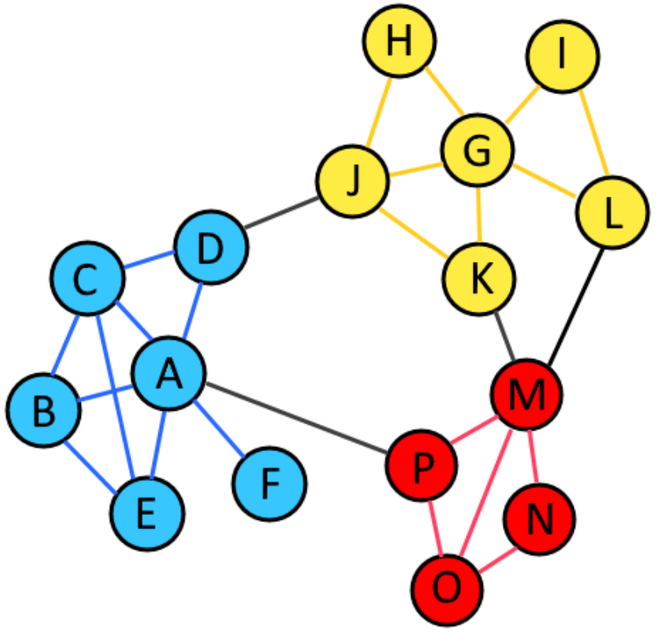
James Curley

Associate Professor, University of Texas at Austin

Community detection in networks



Fast-greedy detection



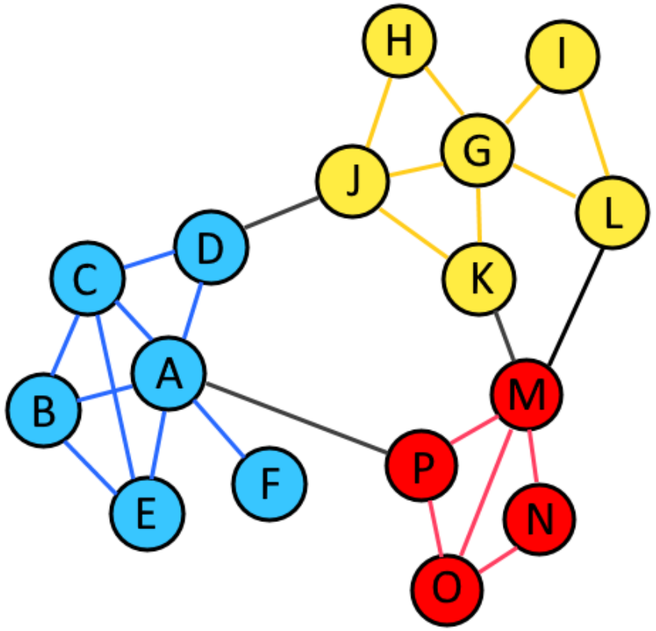

```
fastgreedy.community(g)
```

```
IGRAPH clustering fast greedy,
groups: 3, mod: 0.5
+ groups:
  $`1`
  [1] "A" "B" "C" "D" "E" "F"

  $`2`
  [1] "J" "G" "H" "I" "K" "L"

  $`3`
  [1] "M" "N" "O" "P"
```

Edge-betweenness detection




```
edge.betweenness.community(g)
```

```
IGRAPH clustering edge betweenness,
groups: 3, mod: 0.5
+ groups:
  $`1`
  [1] "A" "B" "C" "D" "E" "F"

  $`2`
  [1] "J" "G" "H" "I" "K" "L"

  $`3`
  [1] "M" "N" "O" "P"
```

```
x <- fastgreedy.community(g)
length(x)
```

```
[1] 3
```

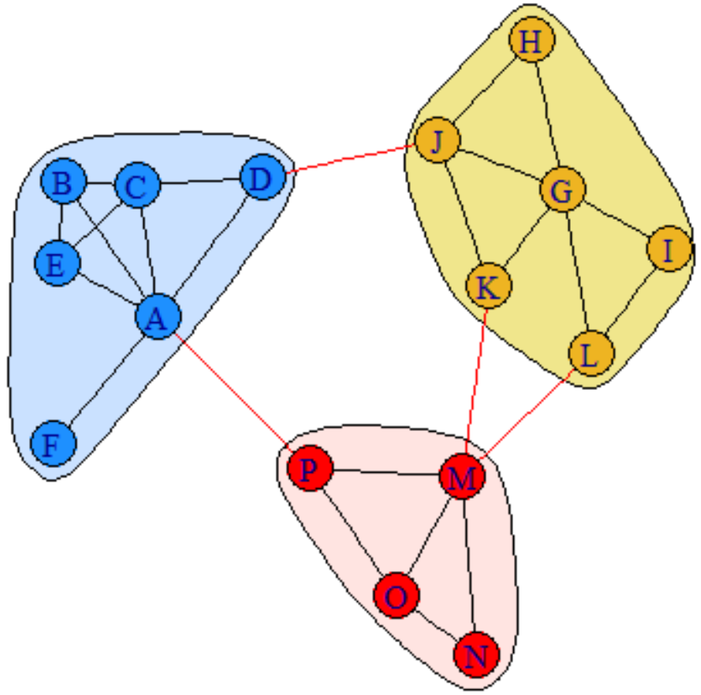
```
sizes(x)
```

```
Community sizes
1 2 3
6 6 4
```

```
membership(x)
```

```
A B C D E F J G H I K L M N O P
1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3
```

```
plot(x, g)
```



Let's practice!

NETWORK ANALYSIS IN R

Interactive network visualizations

NETWORK ANALYSIS IN R

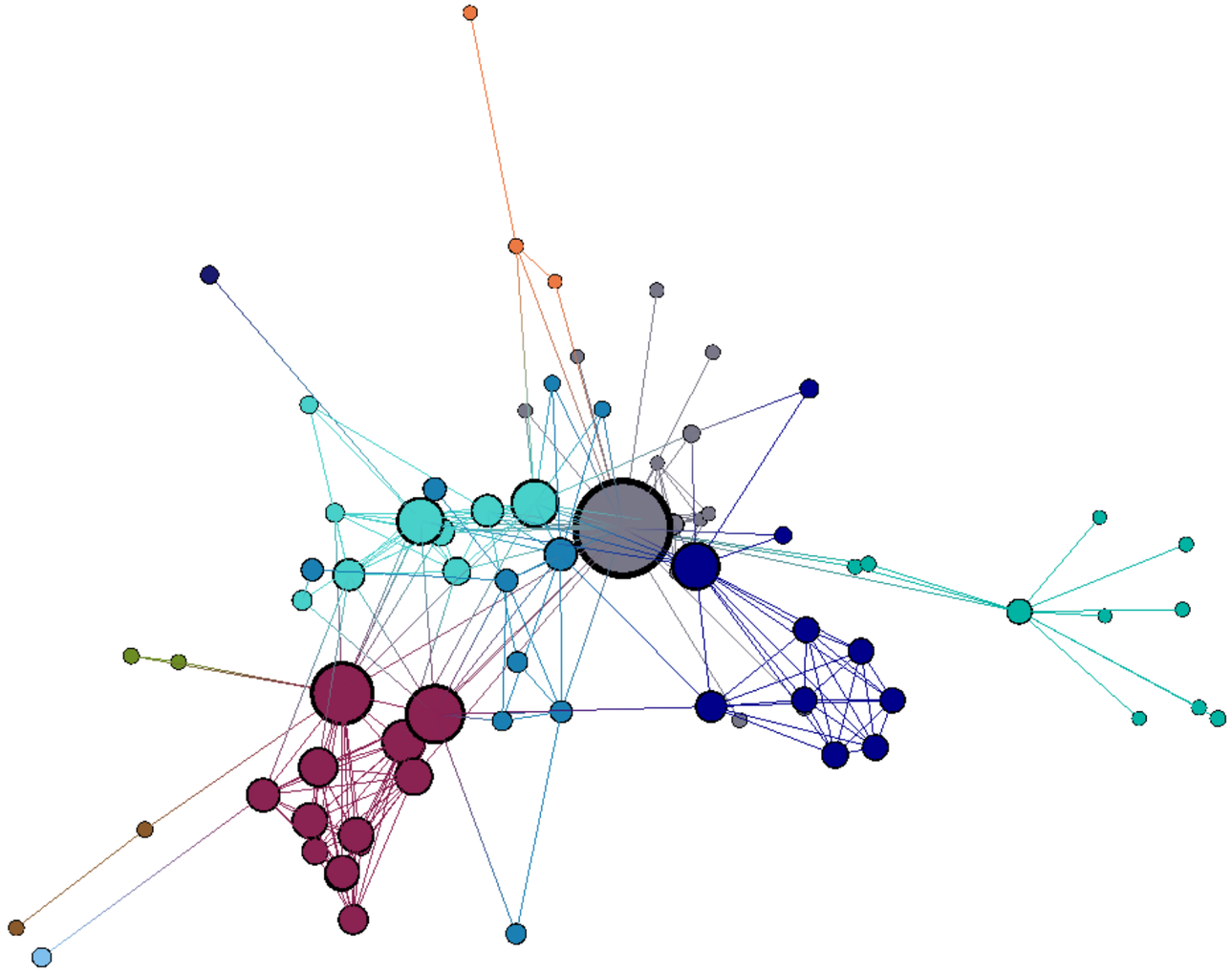
James Curley

Associate Professor, University of Texas at Austin

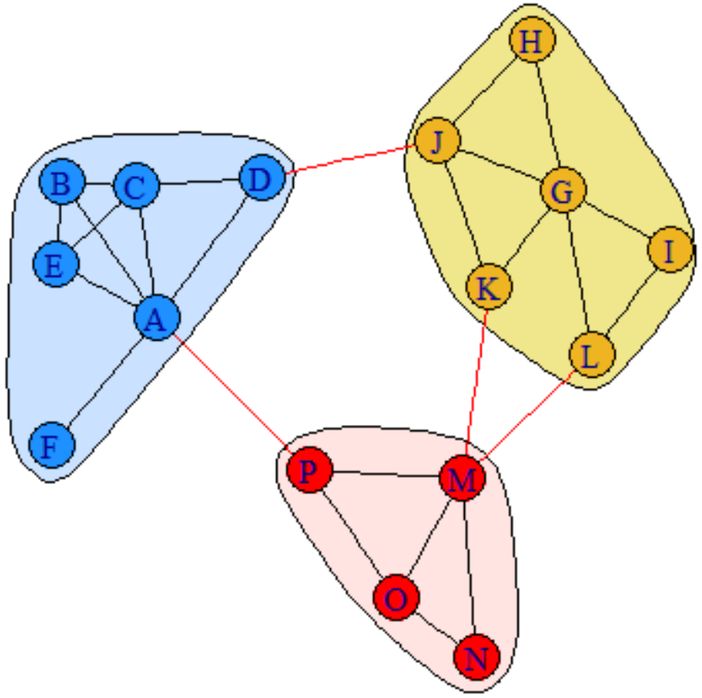
R network visualization packages

- igraph
- statnet
- ggnet
- ggnetwork
- ggraph
- visNetwork
- networkD3
- sigma
- rgexf (igraph to Gephi)
- **threejs**

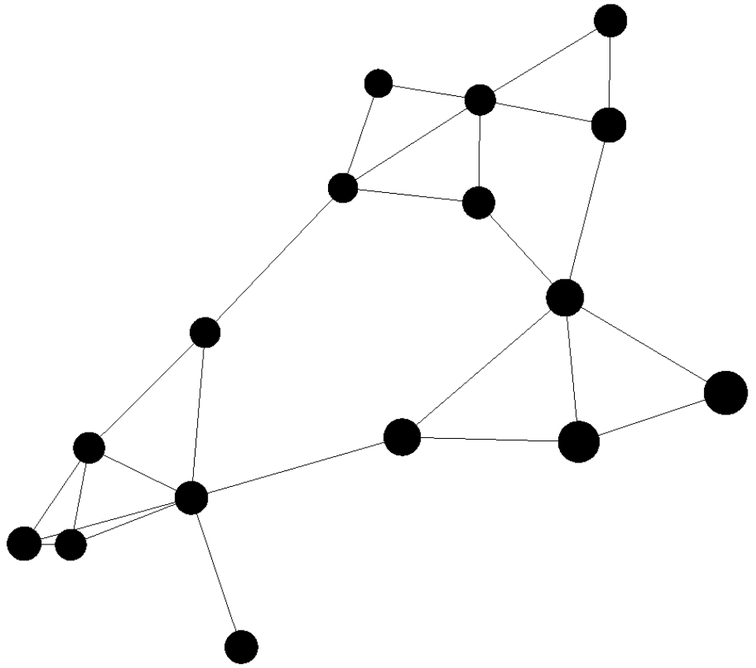
threejs



Creating a threejs visualization




```
library(threejs)
graphjs(g)
```

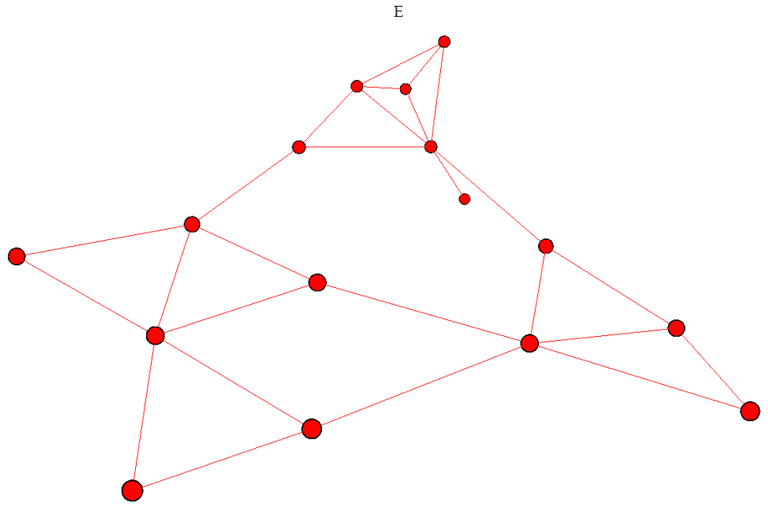


Adding attributes

```
g <- set_vertex_attr(
  g,
  "label",
  value = V(g)$name
  )

g <- set_vertex_attr(
  g,
  "color",
  value = "mistyrose"
  )

graphjs(g, vertex.size = 1)
```

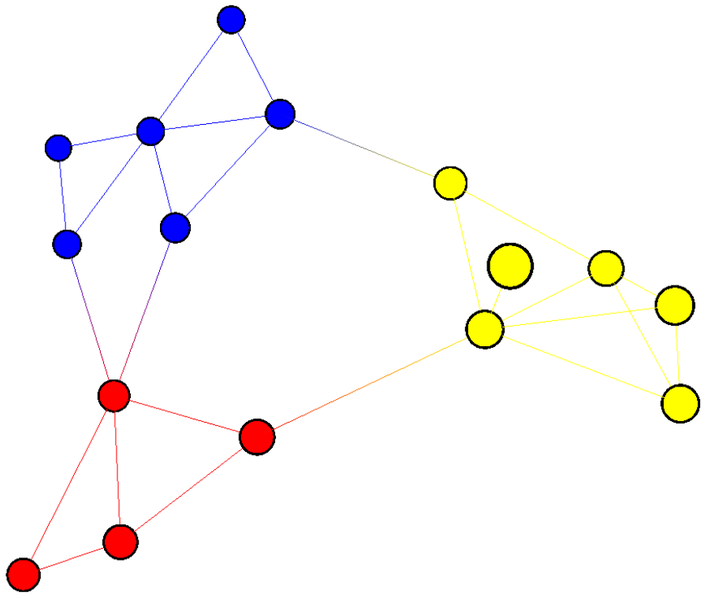


Coloring communities

```
x = edge.betweenness.community(g)
i <-  membership(x)

g <- set_vertex_attr(
  g,
  "color",
  value = c(
    "yellow", "blue", "red"
    )[i]
  )

graphjs(g)
```



Let's practice!

NETWORK ANALYSIS IN R