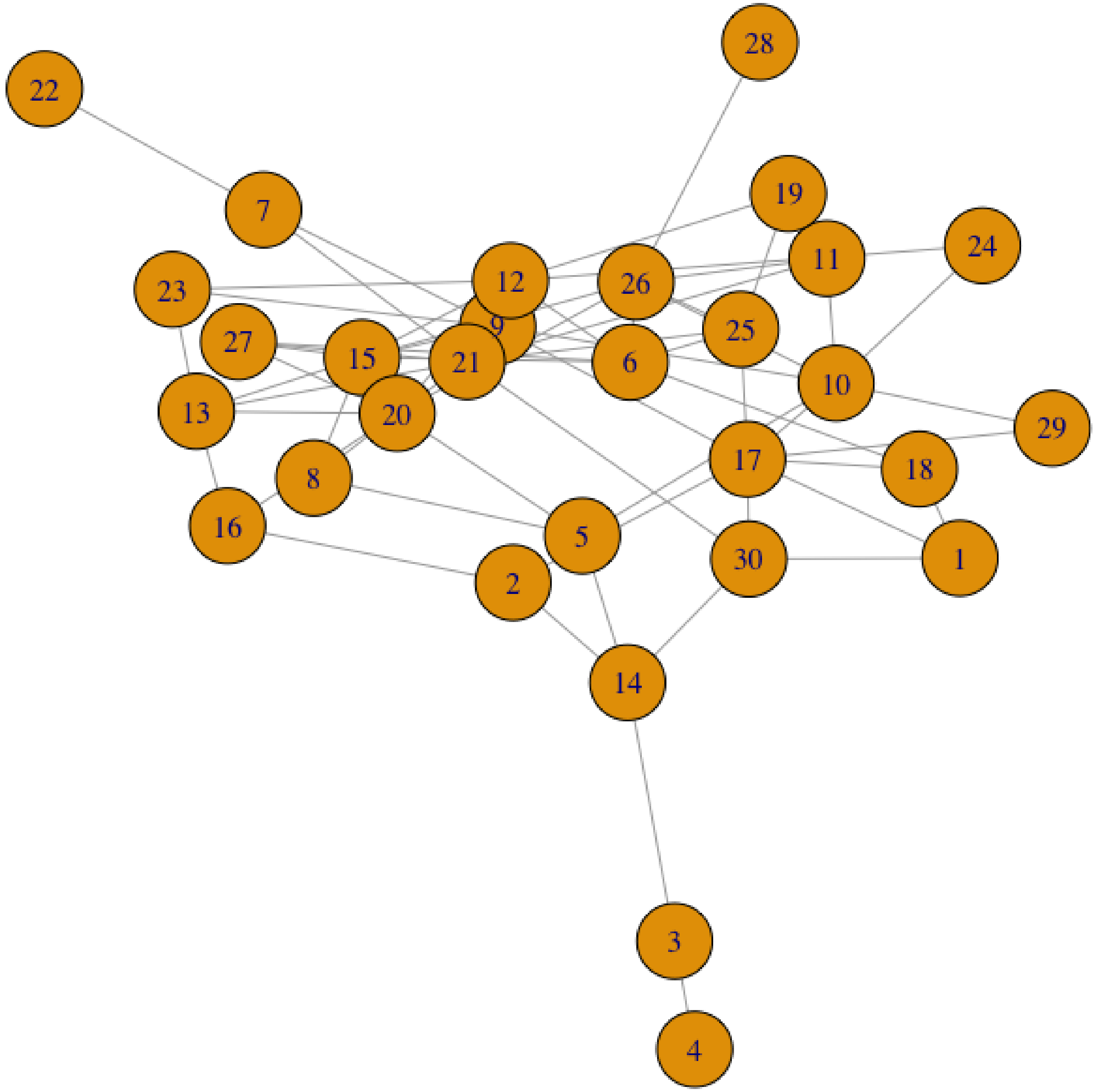
Other packages for plotting graphs

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor

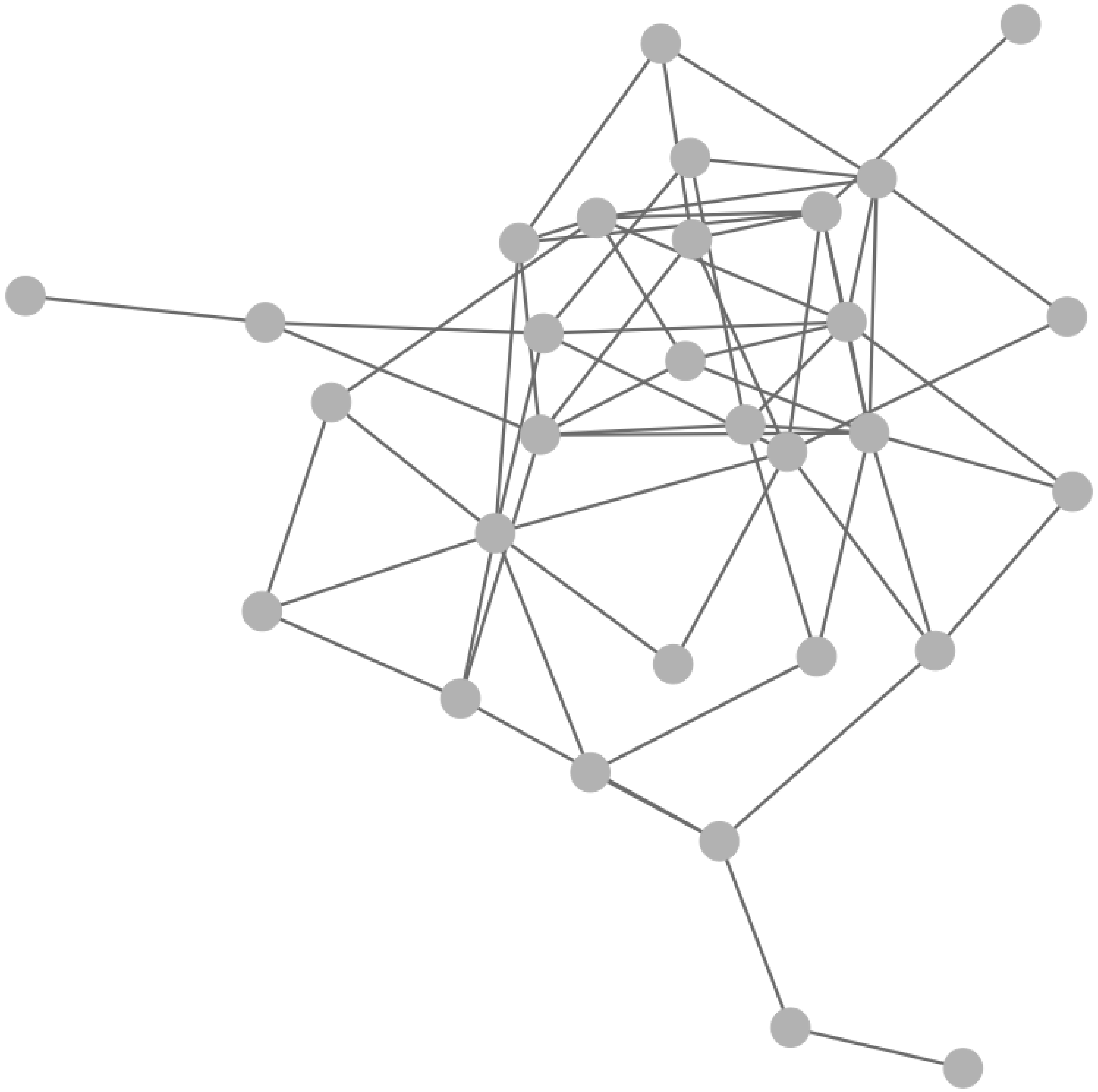
Generating data to plot

```r
library(ggnetwork)
library(igraph)
library(GGally)
library(intergraph)

rand_g <- erdos.renyi.game(30, .15, "gnp", directed = F)
rand_g <- simplify(rand_g)

plot(rand_g)
```


28
22
19
7
24
11
12
26
23
9
25
27
15
21
6
10
29
13
20
17
18
8
16
5
30
1
2
14
3
4

Basic ggnet2

```
net <- asNetwork(rand_g)
ggnet2(net)
```

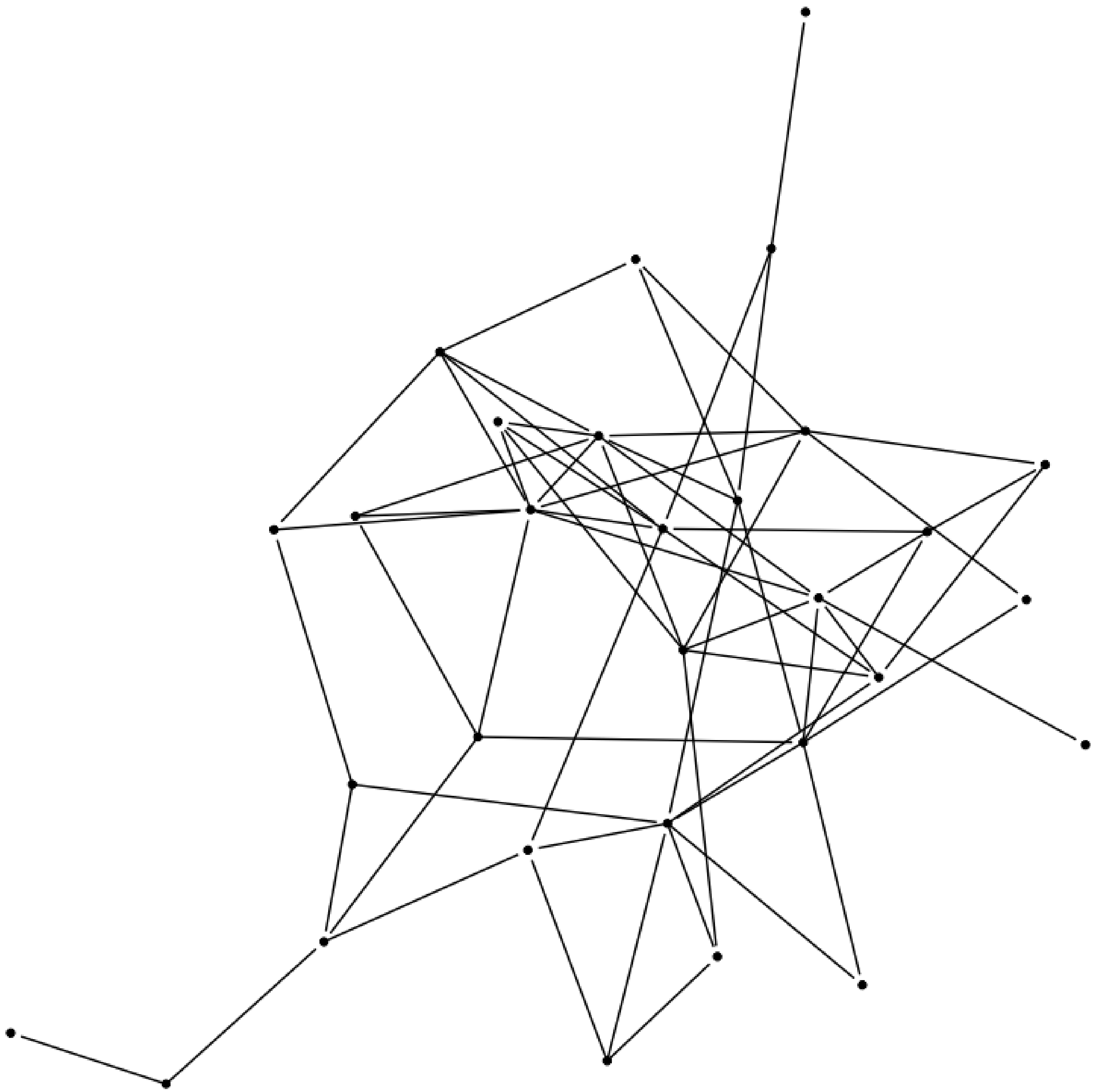


Basic ggnetwork

```
gn <- ggnetwork(rand_g)
g <- ggplot(gn, aes(x = x, y = y, xend = xend, yend = yend)) +
        geom_edges() +
        geom_nodes() +
        theme_blank()
head(gn)
```

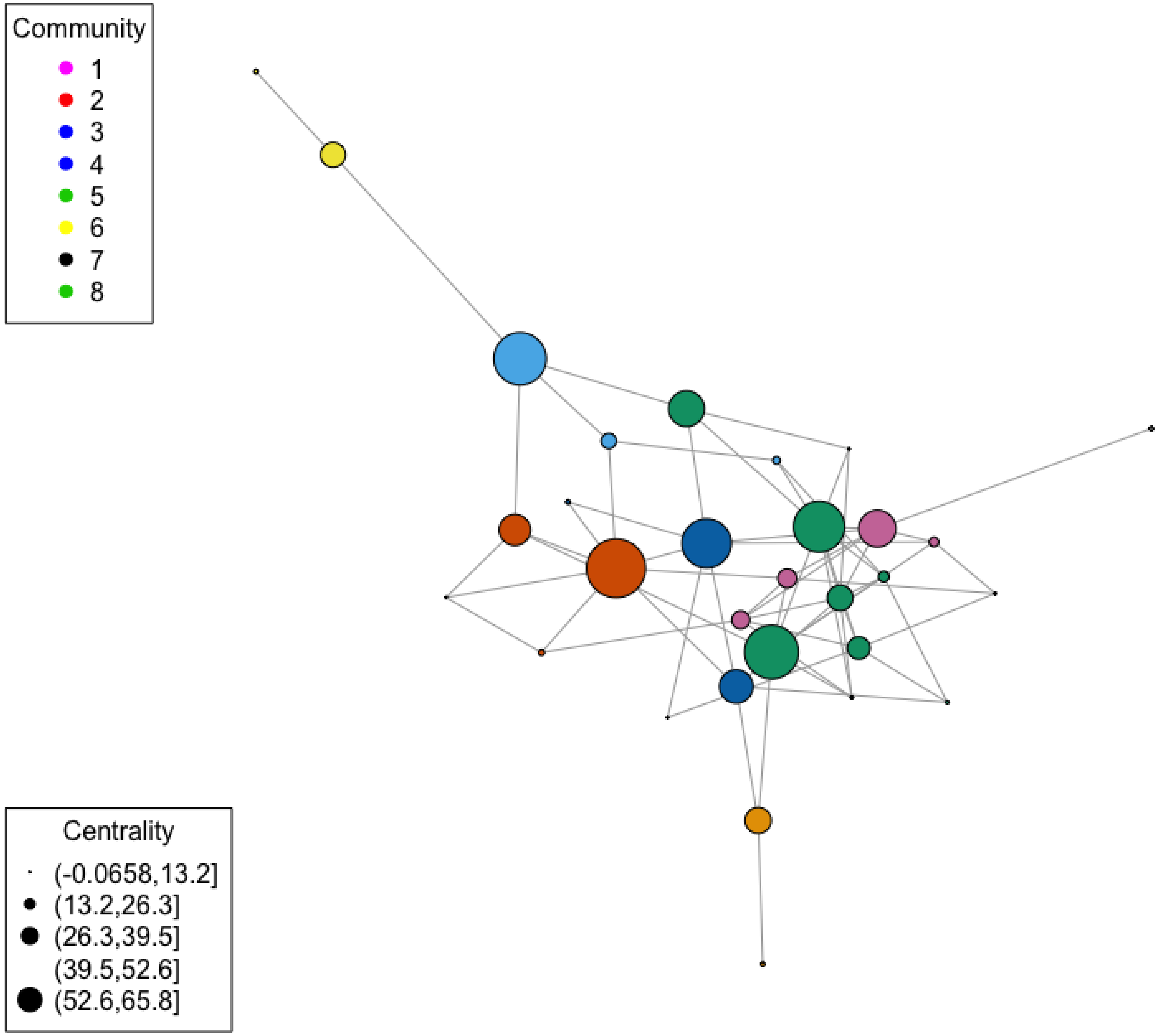
```
          x          y  na.x vertex.names      xend       yend na.y
1 0.4729841 0.01697675 FALSE            1 0.4729841 0.01697675   NA
2 0.1883442 0.42284666 FALSE            2 0.1883442 0.42284666   NA
3 0.3485247 0.82865654 FALSE            3 0.3485247 0.82865654   NA
4 0.3905894 1.00000000 FALSE            4 0.3905894 1.00000000   NA
```

```
plot(g)
```



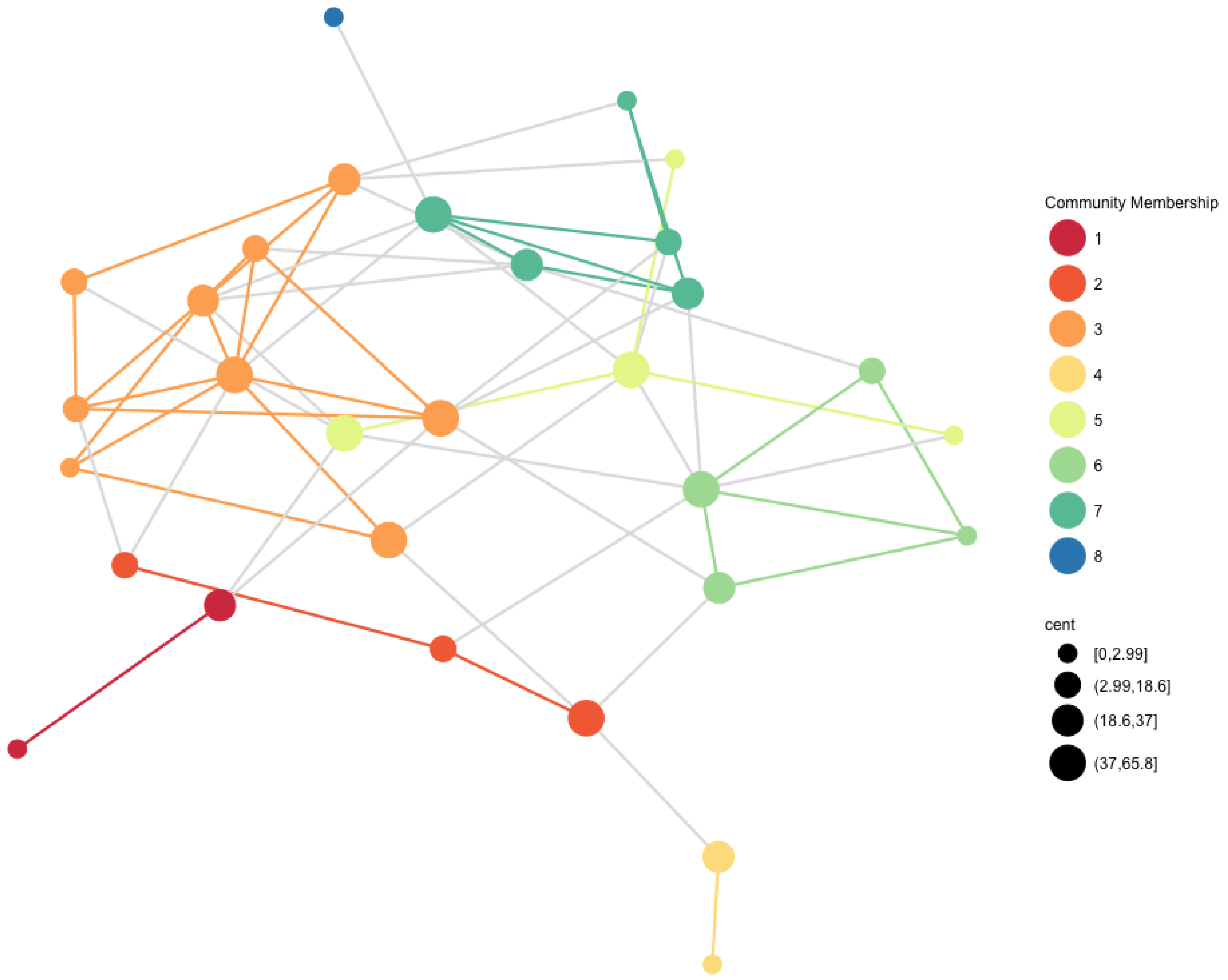
Plotting graphs with attributes

```r
# Add attributes
V(rand_g)$cent <- betweenness(rand_g)
V(rand_g)$comm <- membership(cluster_walktrap(rand_g))
# Make plot
plot(rand_g, vertex.label = NA, margin = 0,
     vertex.color = V(rand_g)$comm,
     vertex.size = V(rand_g)$cent / 6)
# Add legend for community membership
legend('topleft', legend= sort(unique( V(rand_g)$comm)),
col=  sort(unique(V(rand_g)$comm)), pch = 19, title = "Community")
# Add cuts and then get quantiles for size legend
cc <- cut(V(rand_g)$cent, 5)
scaled <- quantile(V(rand_g)$cent, seq(0.3, 0.9, length = 5)) / 25
# Add size legend for centrality
legend('bottomleft', legend= levels(cc),
  pt.cex = scaled, pch = 19, title = "Centrality")
```


Community
1
2
3
4
5
6
7
8
Centrality
(-0.0658,13.2]
(13.2,26.3]
(26.3,39.5]
(39.5,52.6]
(52.6,65.8]

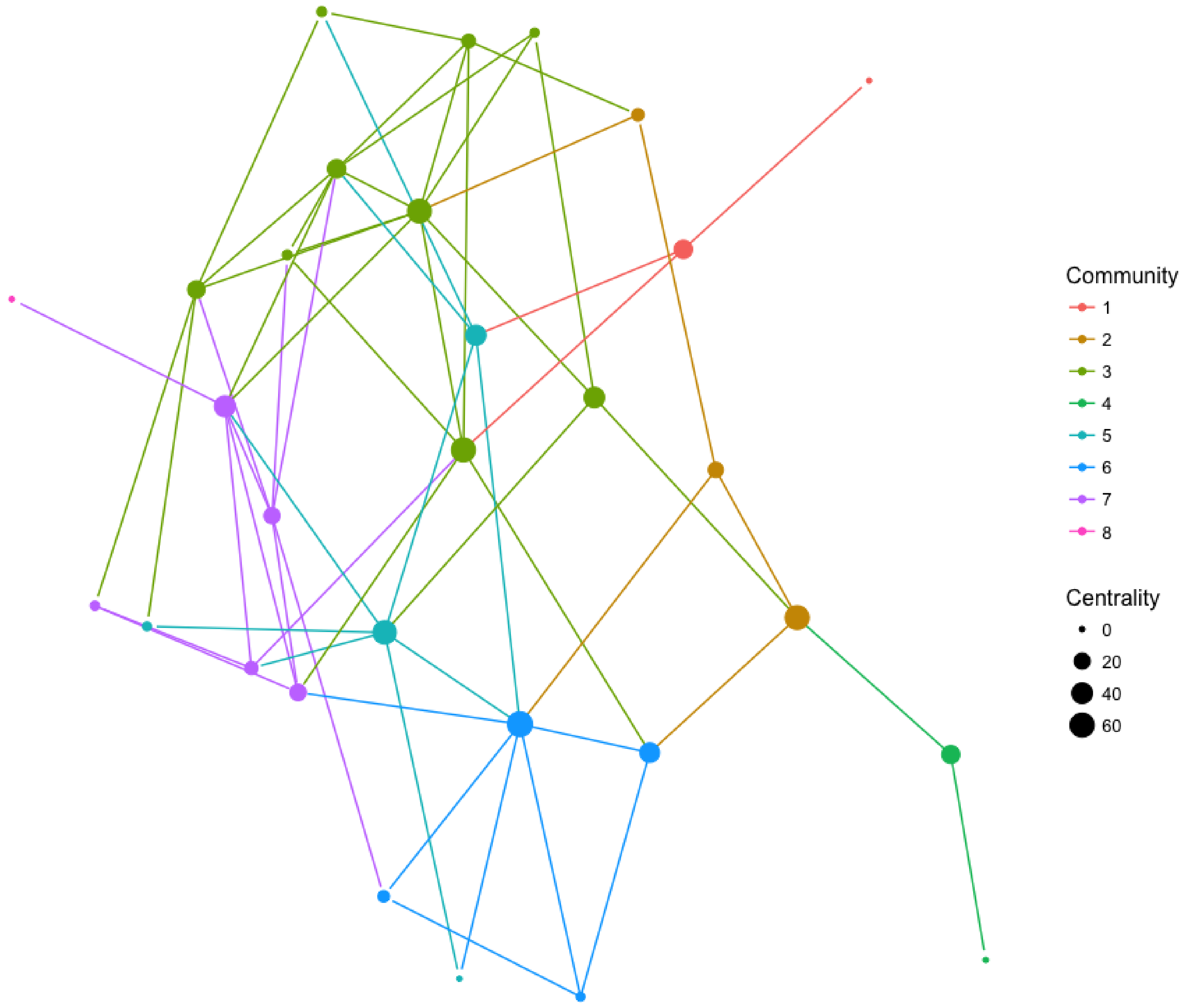
ggnet2 plot with attributes

```
net <- asNetwork(rand_g)

ggnet2(net,
       node.size = "cent",
       node.color = "comm",
       edge.size = 0.8,
       color.legend = "Community Membership",
       color.palette = "Spectral",
       edge.color = c("color", "gray88"),
       size.cut = TRUE,
       size.legend = "Centrality")
```


Community Membership
1
2
3
4
5
6
7
8
cent
[0,2.99]
(2.99,18.6]
(18.6,37]
(37,65.8]

ggnetwork plot with attributes

```r
gn <- ggnetwork(rand_g)

g <- ggplot(gn, aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(aes(color = as.factor(comm))) +
  geom_nodes(aes(color = as.factor(comm), size = cent)) +
  theme_blank()  +
  guides(
    color = guide_legend(title = "Community"),
    size = guide_legend(title = "Centrality"))

plot(g)
```


Community
1
2
3
4
5
6
7
8
Centrality
0
20
40
60

Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R

Interactive visualizations

CASE STUDIES: NETWORK ANALYSIS IN R

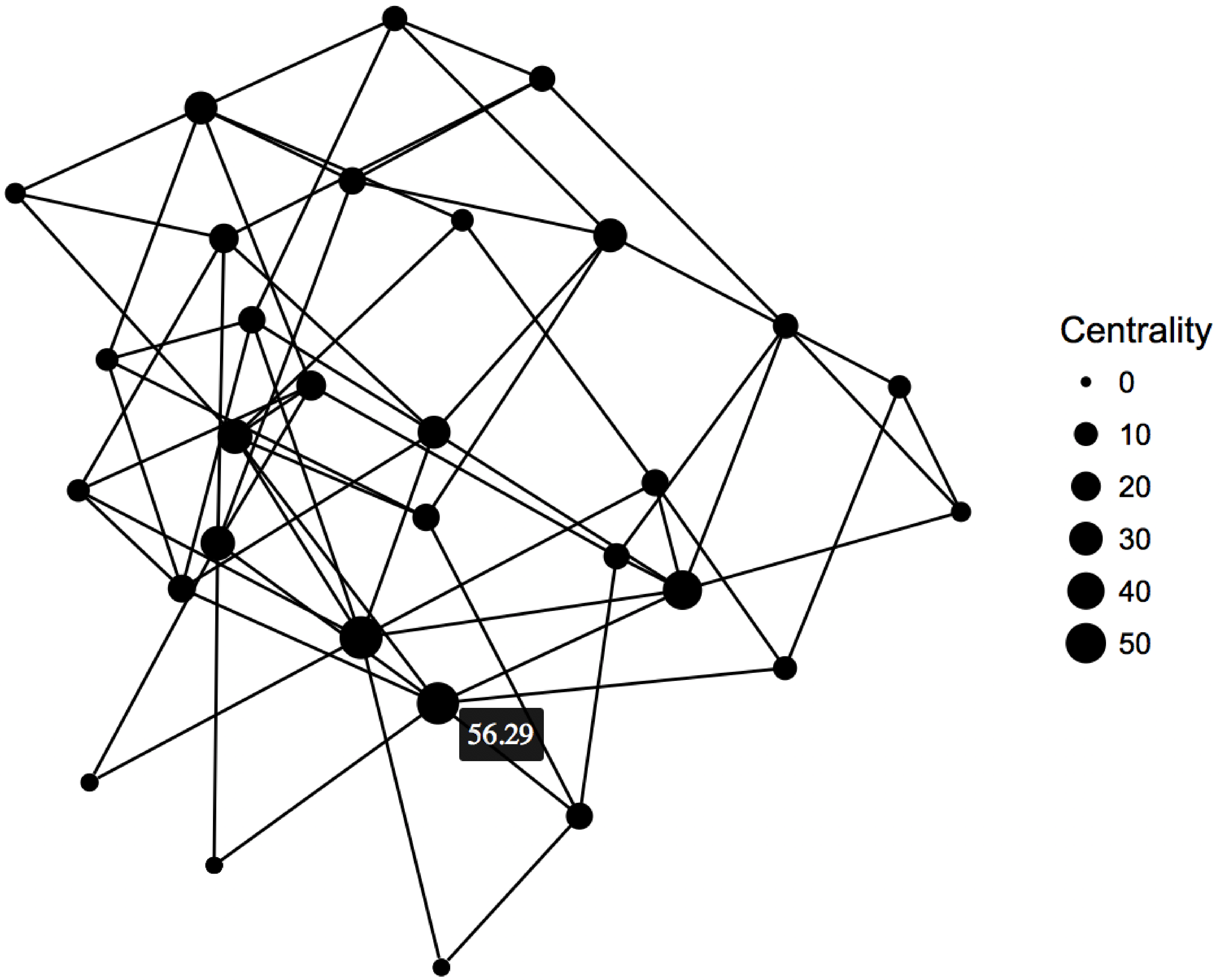
Edmund Hart
Instructor

Generating some data

```r
library(igraph)
library(ggnetwork)
library(ggiraph)
library(htmlwidgets)
library(networkD3)

# Create random graph
rand_g <- erdos.renyi.game(30, 0.12, "gnp", directed = FALSE)
rand_g <- simplify(rand_g)

V(rand_g)$cent <- betweenness(rand_g)
```

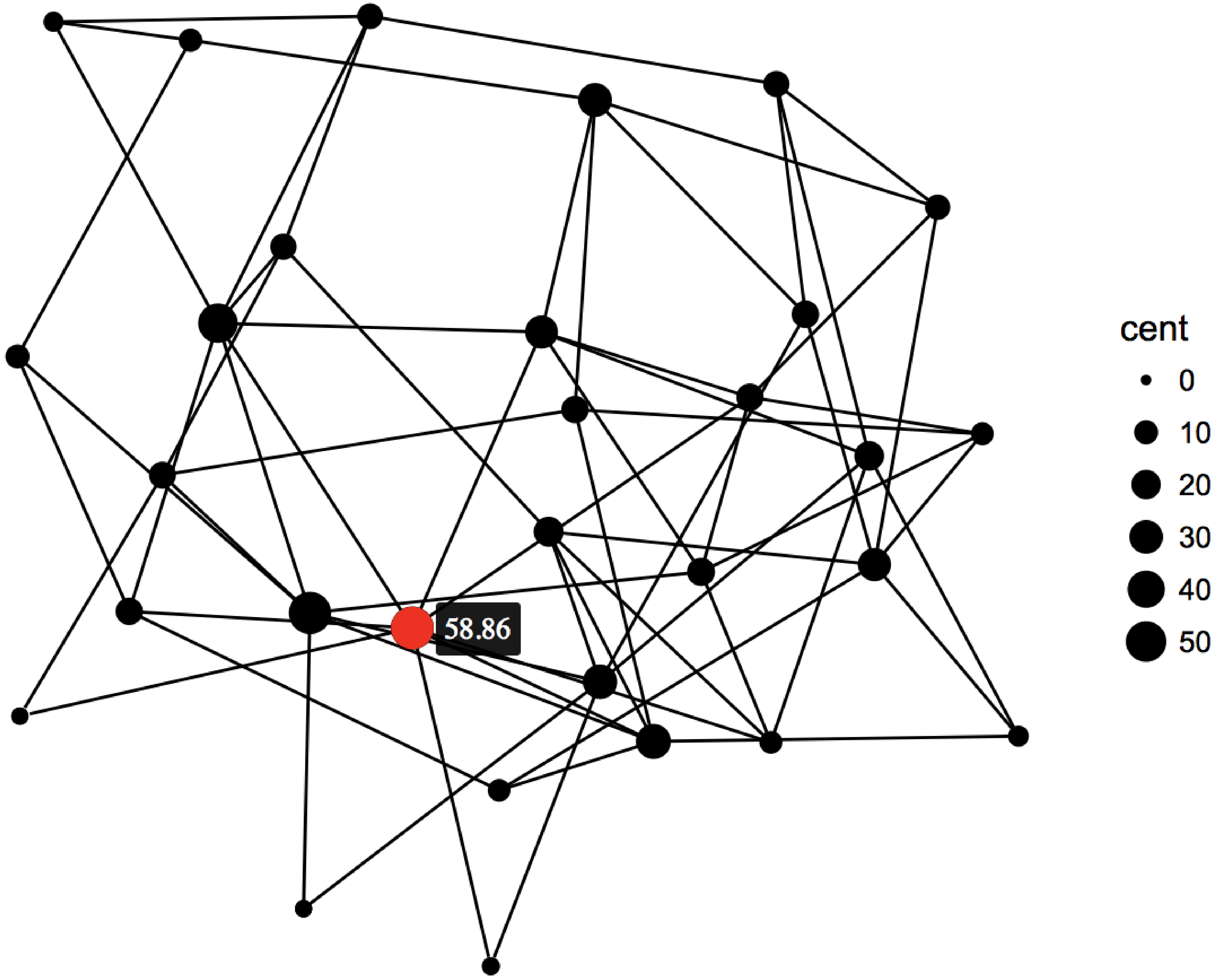
Interactive plots with ggiraph

```
# Plot graph with ggplot2 and ggnetwork
g <- ggplot(ggnetwork(rand_g),
            aes(x = x, y = y, xend = xend, yend = yend)) +
     geom_edges(color = "black") +
     geom_nodes(aes(size = cent))+ theme_blank() +
     guides(size = guide_legend(title = "Centrality"))
# Create ggiraph object
my_gg <- g + geom_point_interactive(aes(tooltip = round(cent, 2)),
        size = 2)
# Display ggiraph object
ggiraph(code = print(my_gg))
```


Centrality
0
10
20
30
40
50
56.29

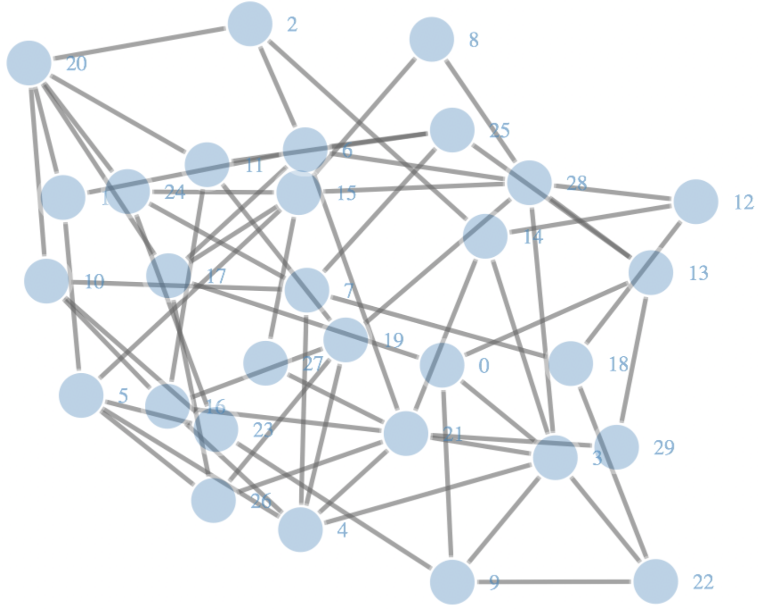
ggiraph customization

```
my_gg <- g + geom_point_interactive(aes(tooltip = round(cent, 2),
                                        data_id = round(cent, 2)),
                                    size = 2)

hover_css = "cursor:pointer;fill:red;stroke:red;r:5pt"

ggiraph(code = print(my_gg),
        hover_css = hover_css,
        tooltip_offx = 10,
        tooltip_offy = -10)
```


58.86
cent
0
10
20
30
40
50

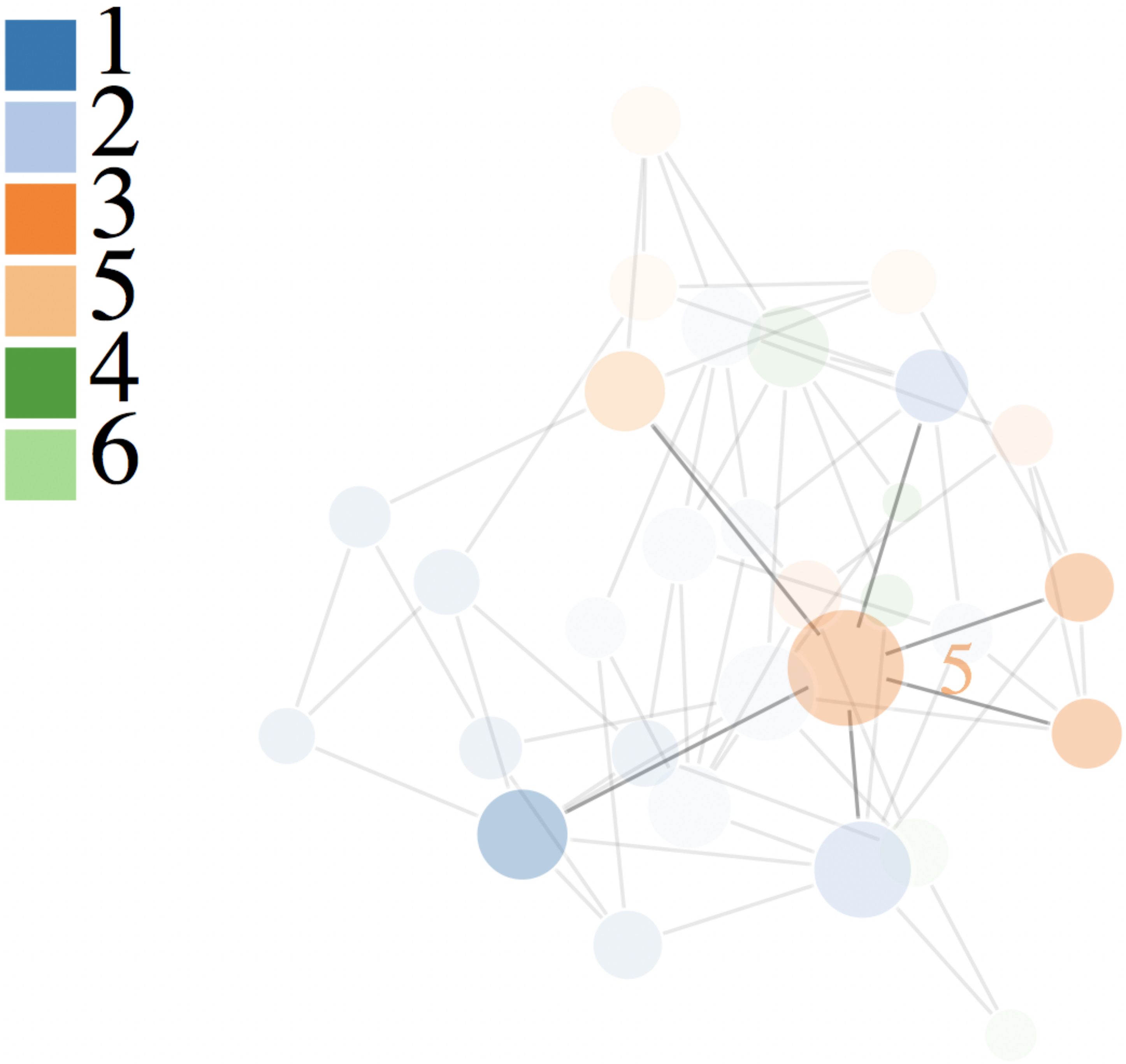
Plotting with networkD3

```
# Convert the igraph object
nd3 <- igraph_to_networkD3(rand_g)

# Create a simple network
simpleNetwork(nd3$links)
```



More complex networkD3

```
# Add attributes, group is community, and cent is centrality.
nd3$nodes$group = V(rand_g)$comm
nd3$nodes$cent = V(rand_g)$cent
# Plot the graph
forceNetwork(Links = nd3$links,
             Nodes = nd3$nodes,
             Source = 'source',
             Target = 'target',
             NodeID = 'name',
             Group = 'group',
             Nodesize = 'cent',
             legend = T,
             fontSize = 20)
```


1
2
3
5
4
6
5

Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R

Alternative visualizations

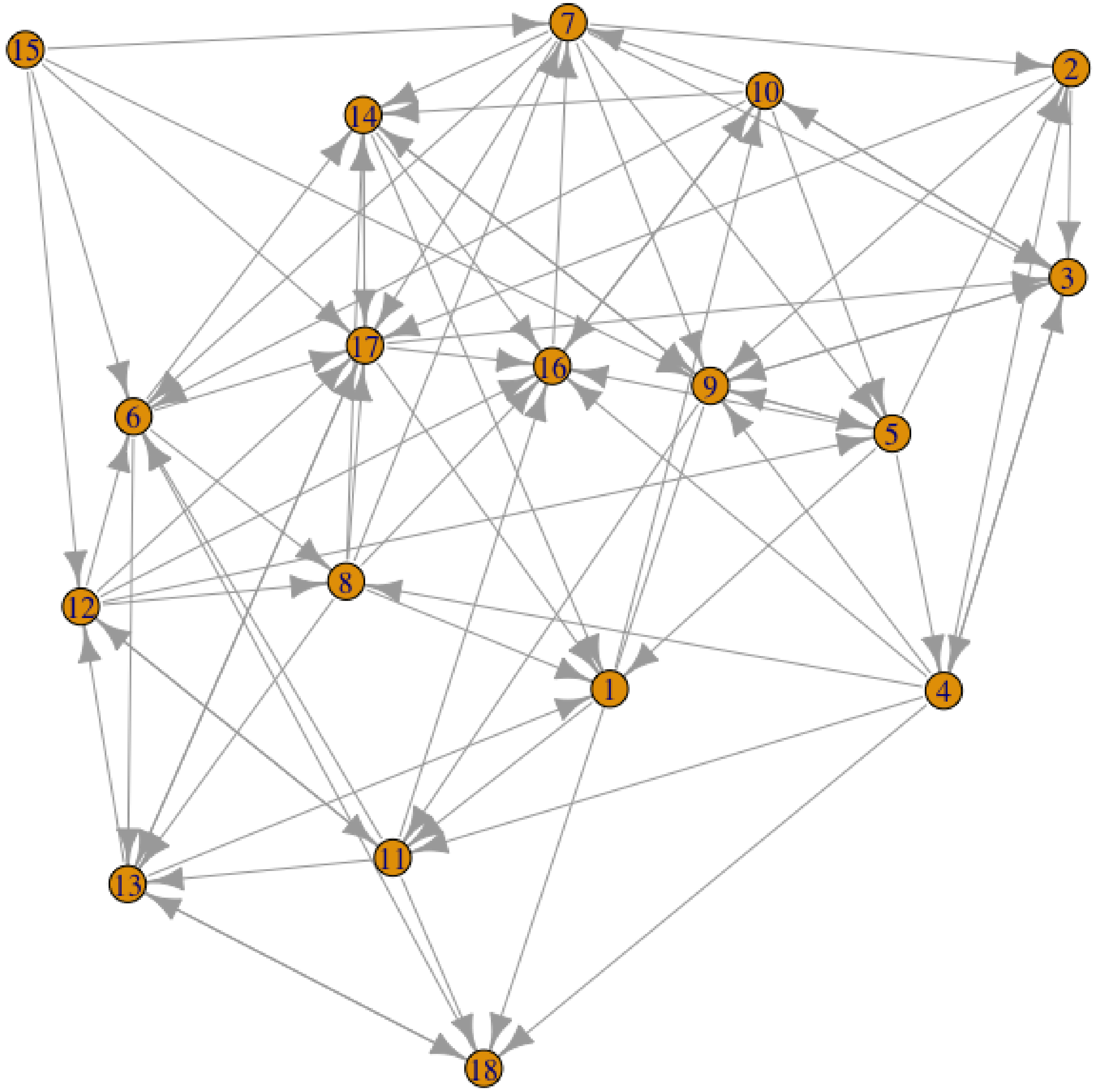
CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart

Instructor

Introduction to hive plots

```r
library(HiveR)
library(igraph)

# Create random graph
rand_g <- erdos.renyi.game(18, 0.3, "gnp", directed = TRUE)

# Plot random graph
plot(rand_g, vertex.size = 7)
```


15
7
2
14
10
3
17
16
9
5
6
8
12
1
4
11
13
18

Introduction to hive plots

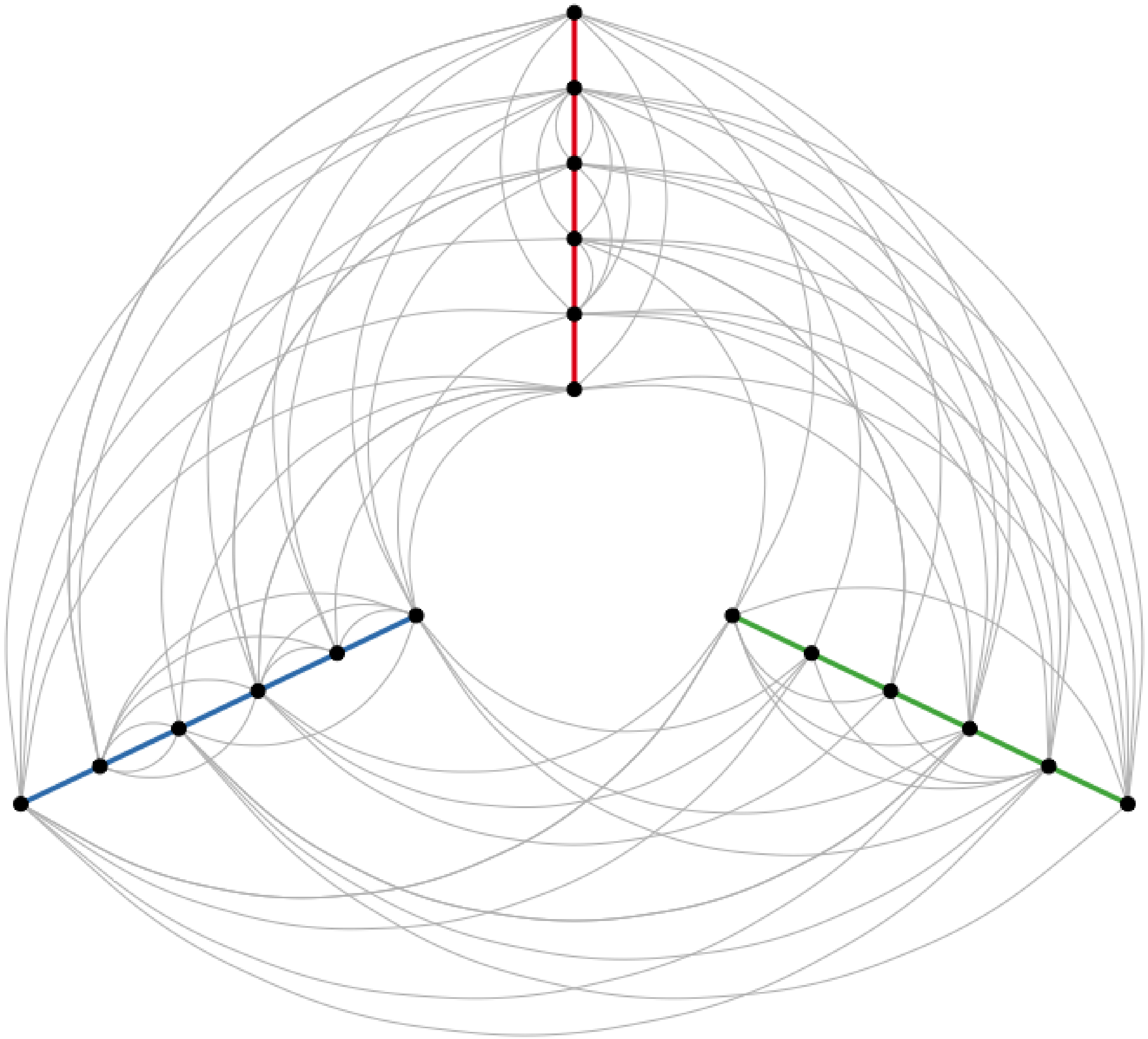
```
# Convert to dataframe for hive plots and add weights
rand_g_df <- as.data.frame(get.edgelist(rand_g))
rand_g_df$weight <- 1
# Convert to hive object
rand_hive <- edge2HPD(edge_df = rand_g_df)
# Set the axis and the radius of each node
rand_hive$nodes$axis <- sort(rep(1:3, 6))
rand_hive$nodes$radius <- as.double(rep(1:6, 3))
```

Introduction to hive plots

```
# See how nodes are modified
rand_hive$nodes
```

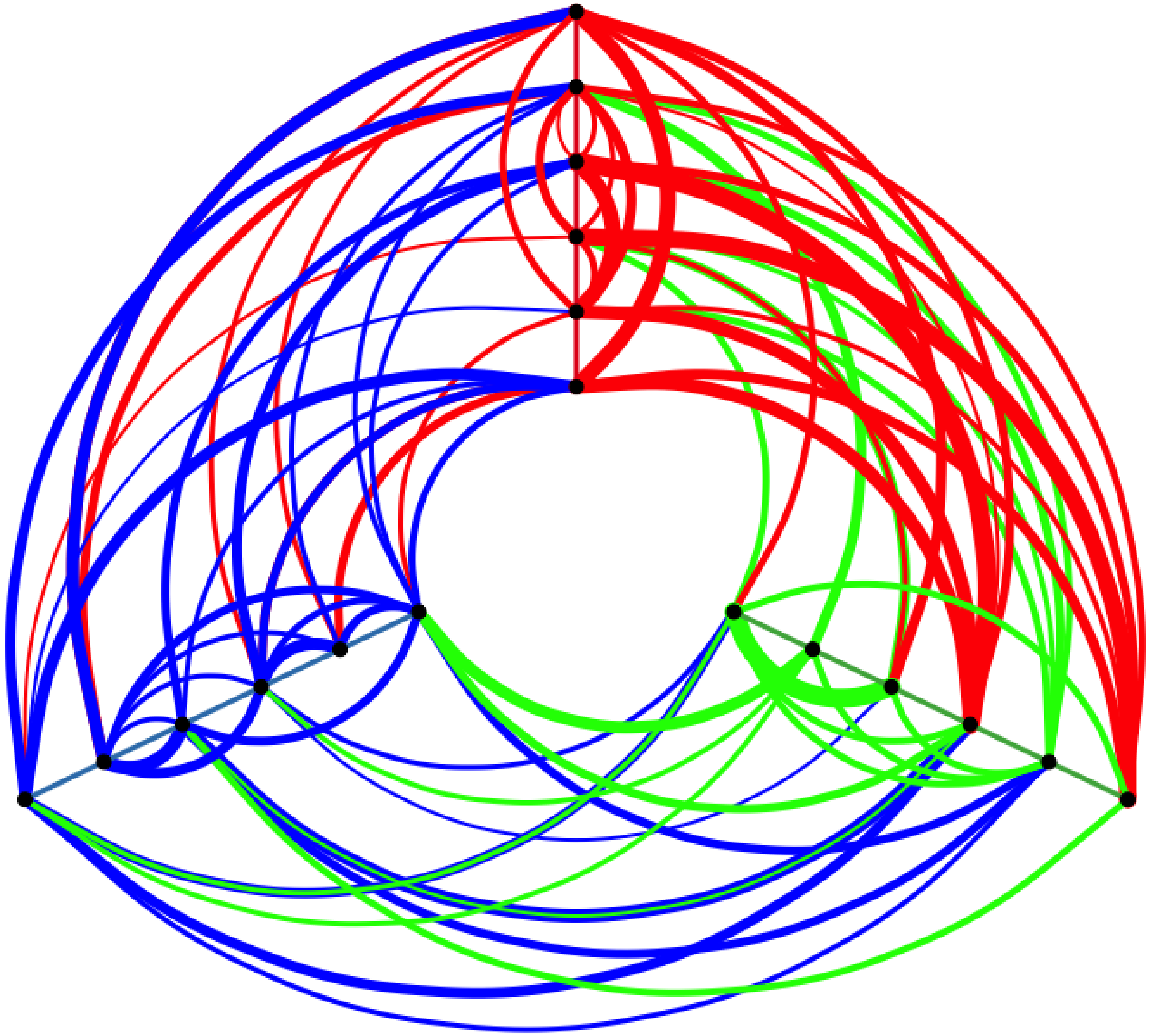
```
id lab axis radius size color
1   2    1      1    1 black
2   8    1      2    1 black
3   9    1      3    1 black
4   3    1      4    1 black
5   4    1      5    1 black
6   7    1      6    1 black
7  11    2      1    1 black
8  14    2      2    1 black
9  18    2      3    1 black
```

```
# See hive plot
plotHive(rand_hive, method = "abs", bkgnd = "white")
```



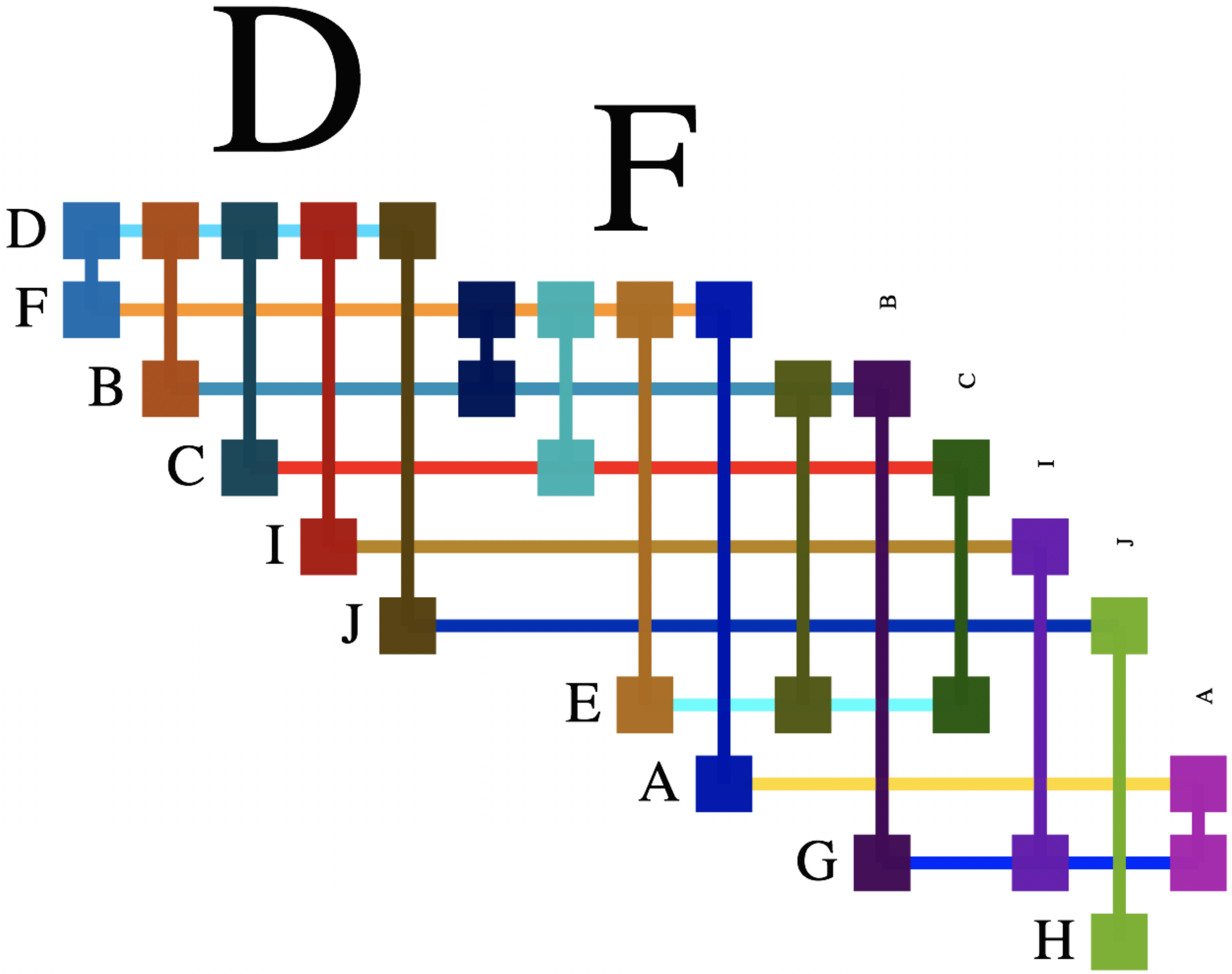
Modifying hive plots

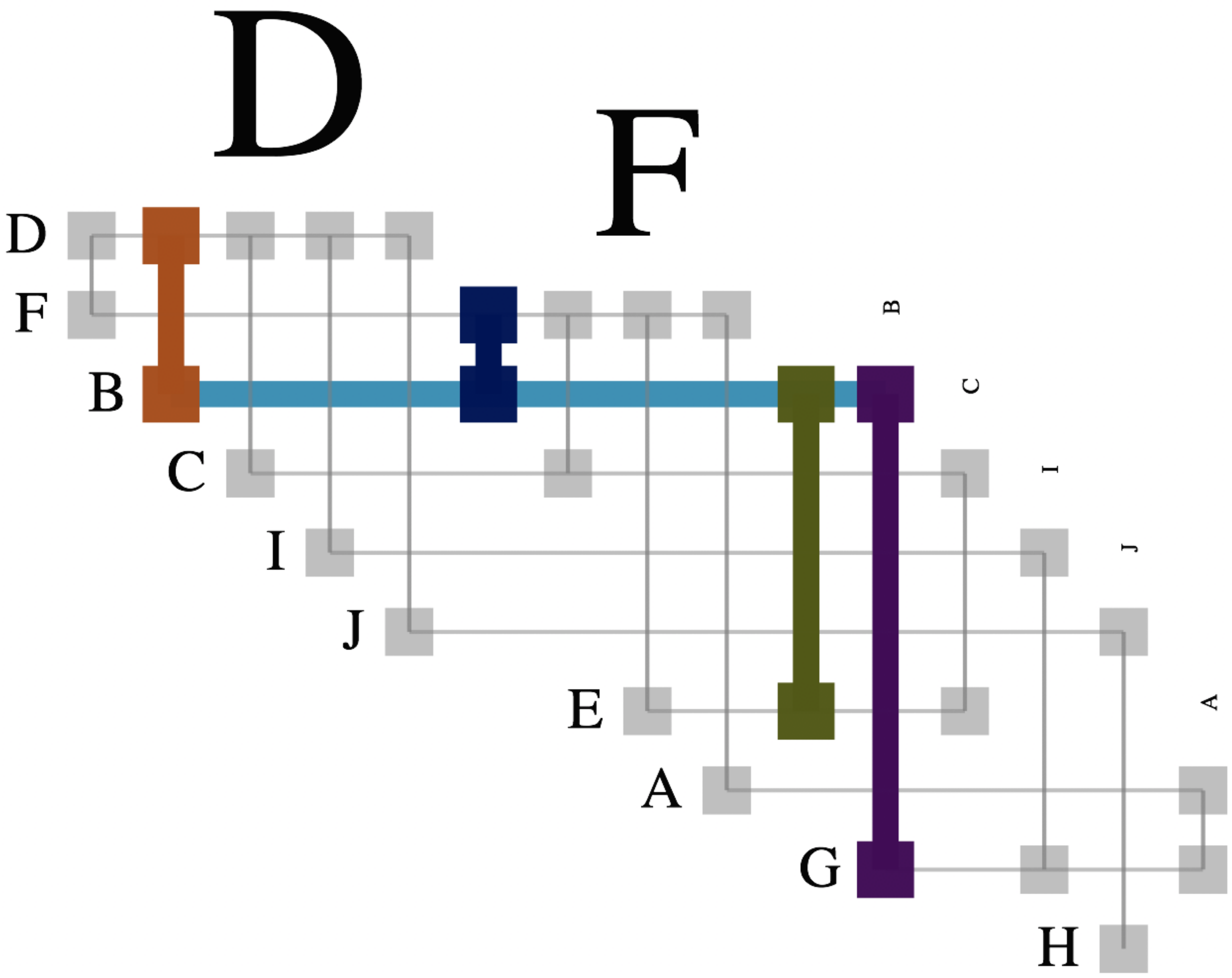
```
# Setting location of each node
rand_hive$nodes$axis <- sort(rep(1:3, 6))
rand_hive$nodes$radius <- as.double(rep(1:6, 3))
# Add weights to each edge
rand_hive$edges$weight <- as.double(
                      rpois(length(rand_hive$edges$weight), 5)
                          )

# Add color based on edge origination
rand_hive$edges$color[rand_hive$edges$id1 %in% 1:6] <- 'red'
rand_hive$edges$color[rand_hive$edges$id1 %in% 7:12] <- 'blue'
rand_hive$edges$color[rand_hive$edges$id1 %in% 13:18] <- 'green'
# Plot
plotHive(rand_hive, method = "abs", bkgnd = "white")
```



Biofabric plots

```r
# Create random graph
rand_g <- erdos.renyi.game(10, 0.3, "gnp", directed = FALSE)
rand_g <- simplify(rand_g)
# Add names to vertices
V(rand_g)$name <- LETTERS[1:length(V(rand_g))]
# Create biofabric plot
biofbc <- bioFabric(rand_g)
bioFabric_htmlwidget(biofbc)
```


D
F
D
F
B
C
I
J
E
A
G
H
B
C
I
J
A


D
F
D
F
B
C
I
J
E
A
G
H
B
C
I
J
A

Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R