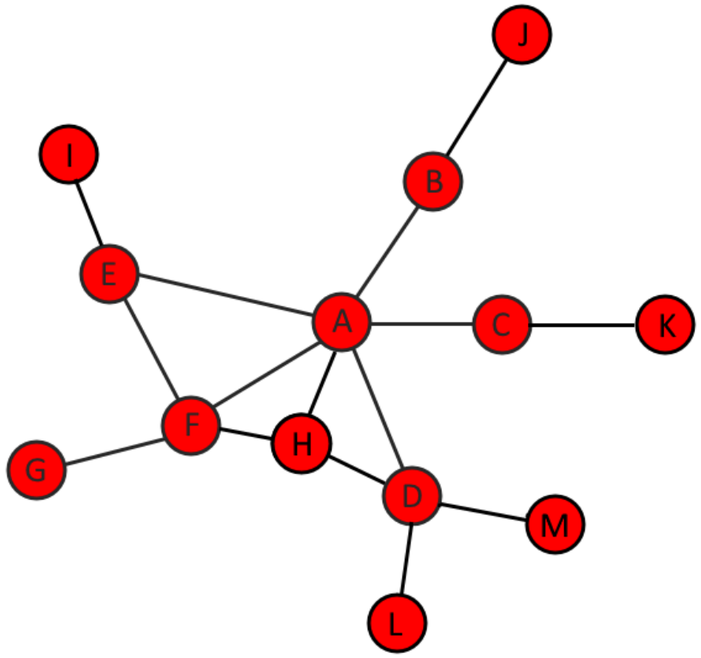
Network Structure

NETWORK ANALYSIS IN R

James Curley

Associate Professor, University of Texas at Austin

Eigenvector centrality

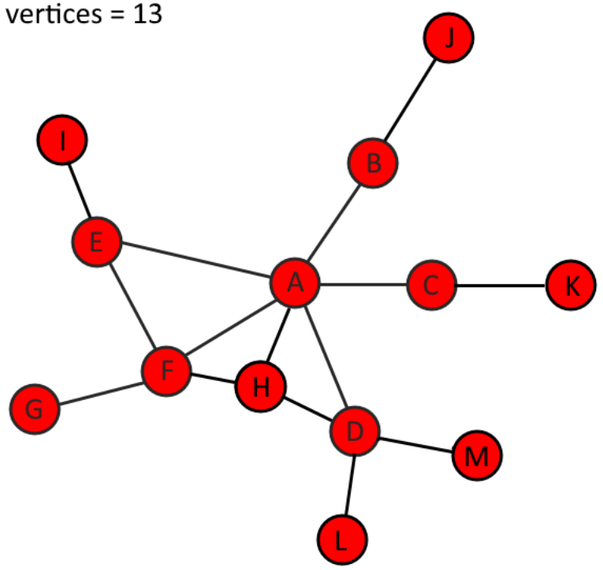



```
eigen_centrality(g)$vector
```

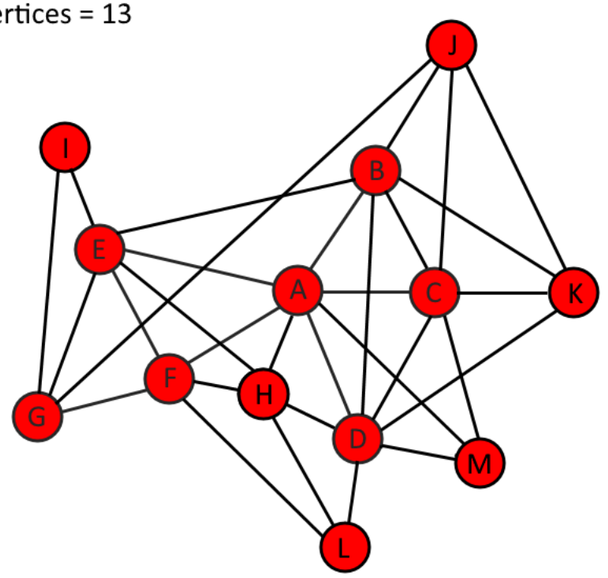
```
   A    B    C    D    E    F    G
1.00 0.33 0.33 0.63 0.58 0.76 0.23

   H    I    J    K    L    M
0.71 0.17 0.10 0.10 0.19 0.19
```

Density

density = 0.19
edges = 15
vertices = 13



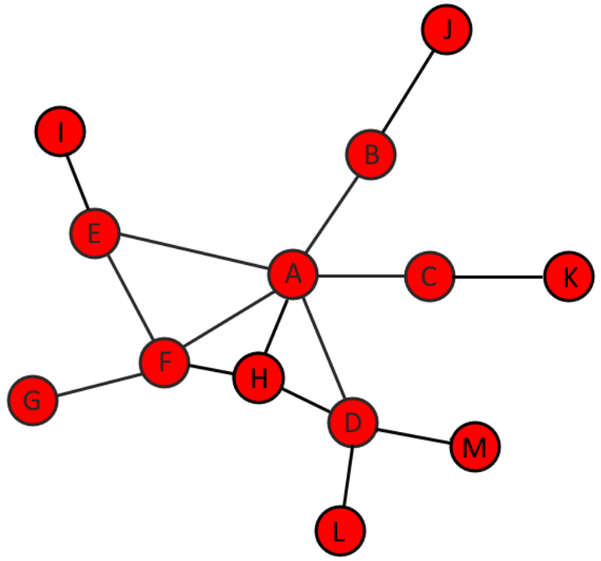
density = 0.38
edges = 30
vertices = 13



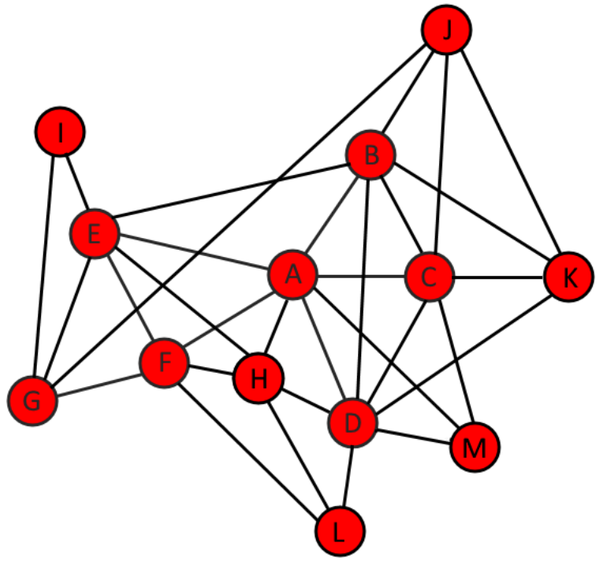

```
edge_density(g)
```

Average path length

average path length = 2.47



average path length = 1.81




```
mean_distance(g, directed = FALSE)
```

Let's practice!

NETWORK ANALYSIS IN R

Network Randomizations

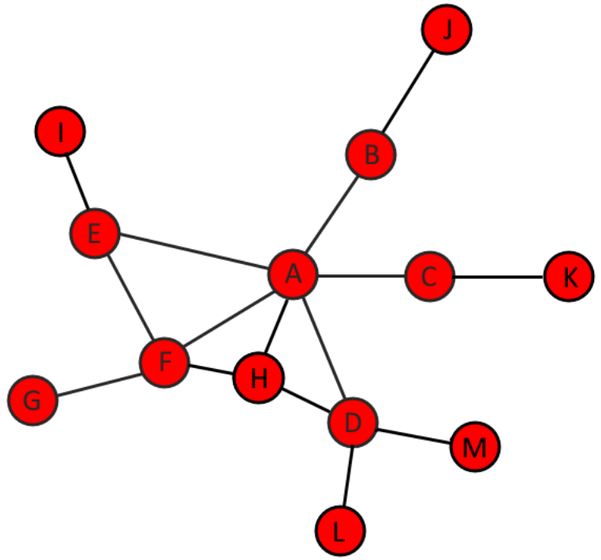
NETWORK ANALYSIS IN R

James Curley

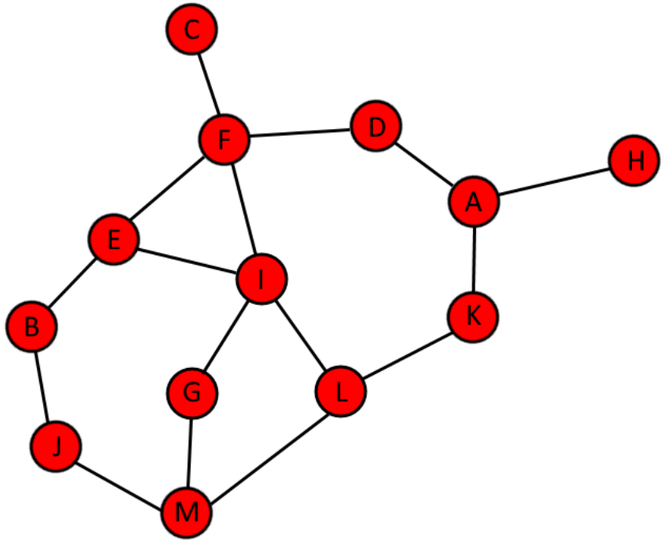
Associate Professor, University of Texas at Austin

Random graphs

density = 0.19
vertices = 13



density = 0.21
vertices = 13




```
erdos.renyi.game(n = gorder(g), p.or.m = edge_density(g), type = "gnp")
```

Random graphs & randomization tests

1. Generate 1000 random graphs based on the original network - e.g. with the same number of vertices and approximate density.
2. Calculate the average path length of the original network.
3. Calculate the average path length of the 1000 random networks.
4. Determine how many random networks have an average path length greater or less than the original network's average path length.

Generate 1000 random graphs:

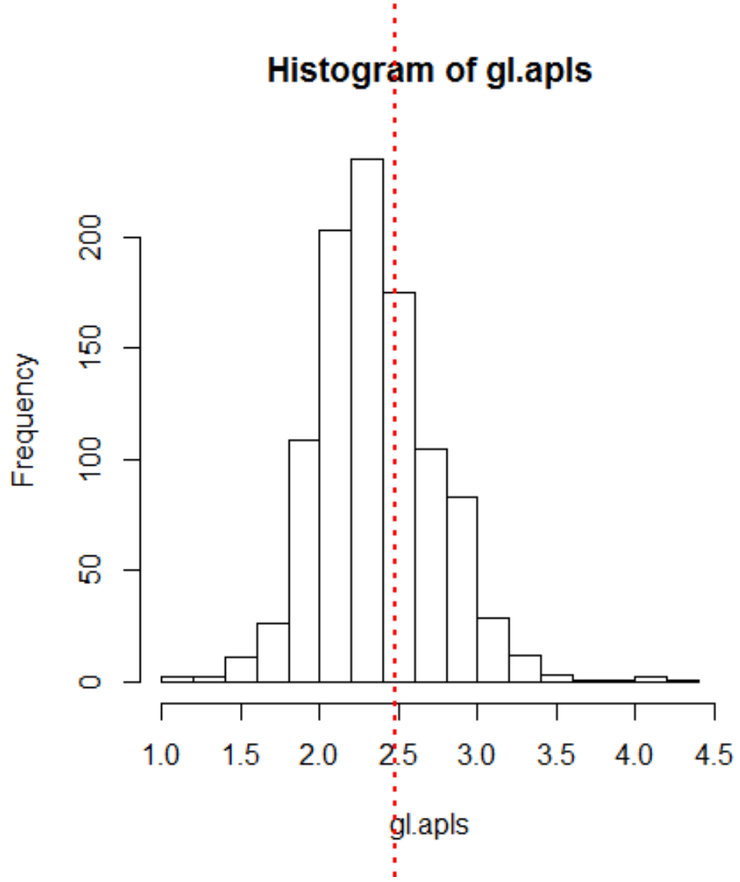
```
gl <- vector('list',1000)

for(i in 1:1000){
    gl[[i]] <- erdos.renyi.game(
                    n = gorder(g),
                    p.or.m = edge_density(g),
                    type = "gnp"
                     )
}
```

Calculate average path length of 1000 random graphs:

```
gl.apls <- unlist(
  lapply(gl, mean_distance, directed = FALSE)
)
```

Comparing to the original network

```
hist(gl.apls, breaks = 20)

abline(
  v = mean_distance(
    g, directed=FALSE
    ),
      col = "red",
      lty = 3,
      lwd = 2
)
```



Let's practice!

NETWORK ANALYSIS IN R

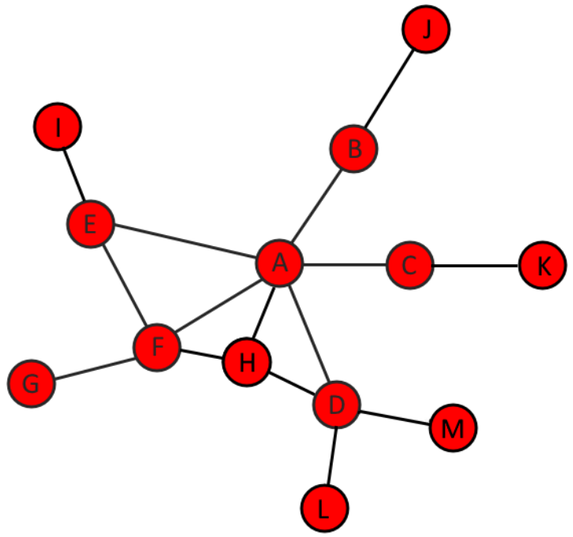
Network substructures

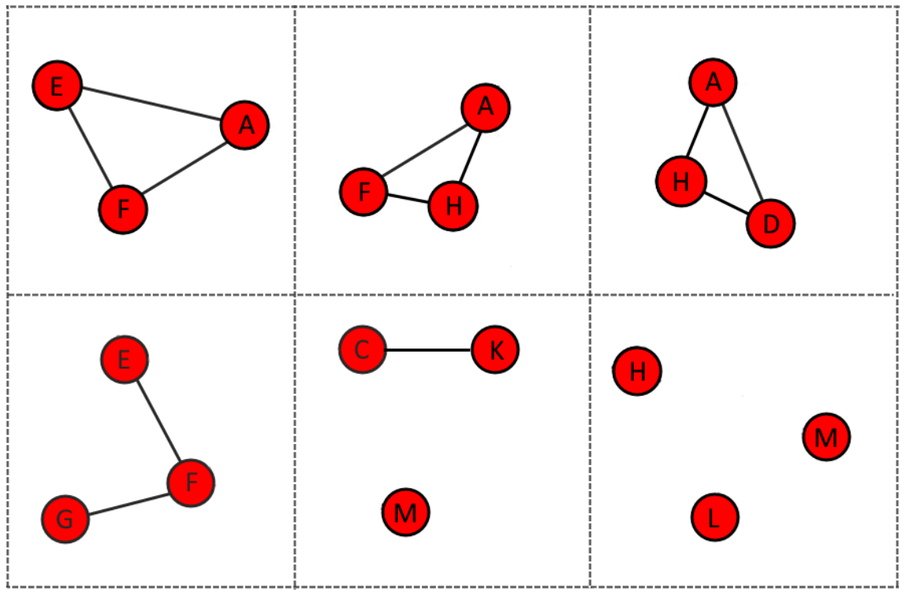
NETWORK ANALYSIS IN R

James Curley

Associate Professor, University of Texas at Austin

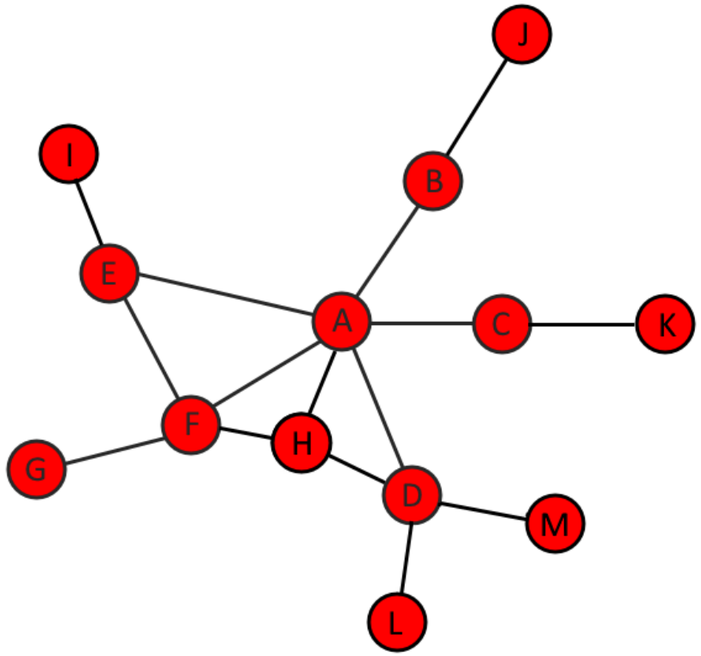
Transitivity






```
triangles(g)
```

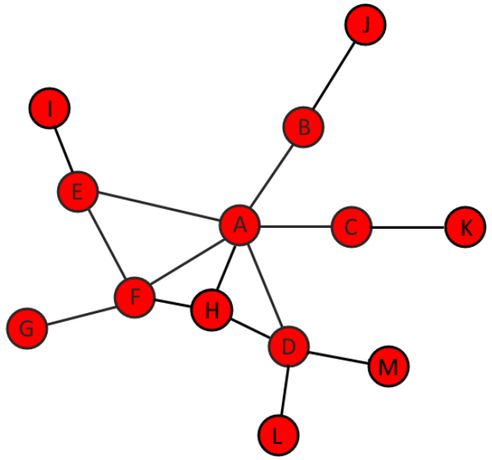
Global transitivity




```
transitivity(g)
```

```
[1] 0.26
```

Local transitivity




```
transitivity(g,
             vids = 'A',
             type = "local")
```

```
0.2
```

```
count_triangles(g, vids = 'A')
```

```
3
```

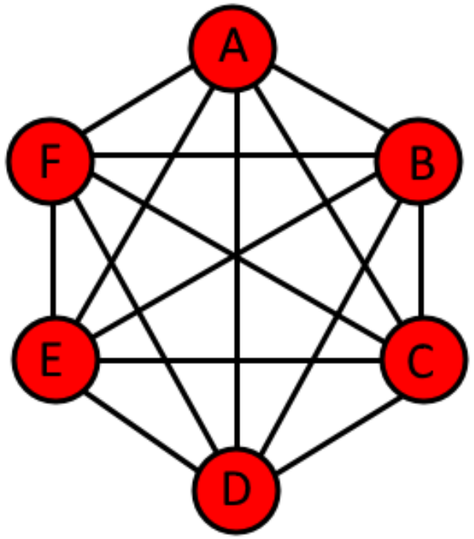
```
count_triangles(g, vids = 'F')
```

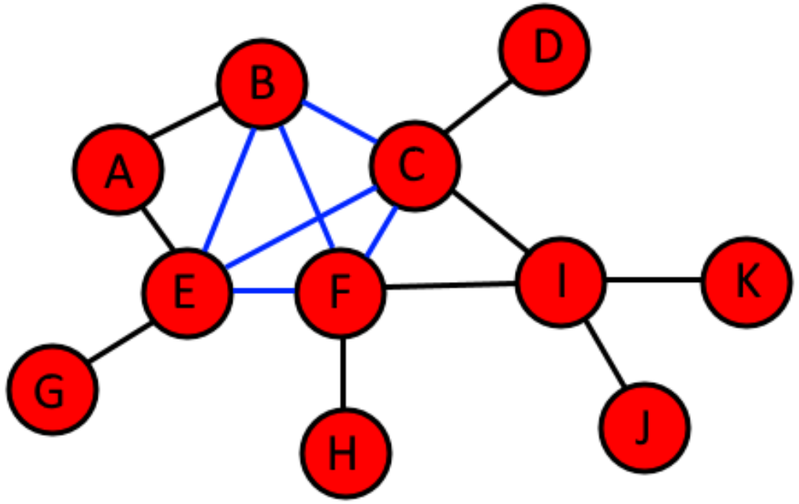
```
2
```

```
transitivity(g,
             vids = 'F',
             type = "local")
```

```
0.33
```

Cliques

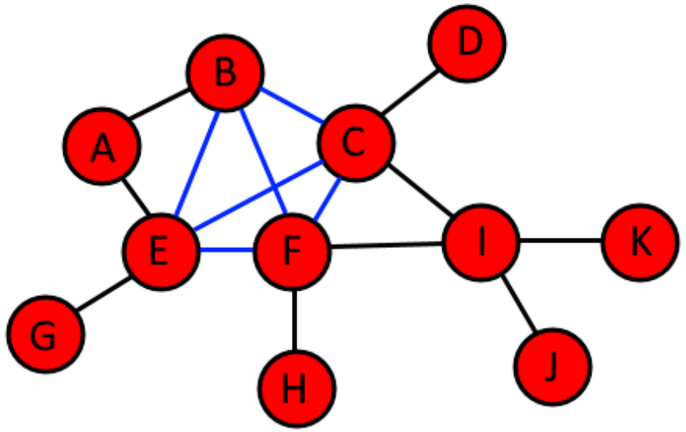




Identifying cliques

```
largest_cliques(g)
```

```
+ 4/11 vertices, named:
[1] C F B E
```




```
max_cliques(g)
```

```
...
[[6]]
+ 3/11 vertices, named:
[1] A B E

[[7]]
+ 3/11 vertices, named:
[1] I C F

[[8]]
+ 4/11 vertices, named:
[1] E B F C
```

Let's practice!

NETWORK ANALYSIS IN R