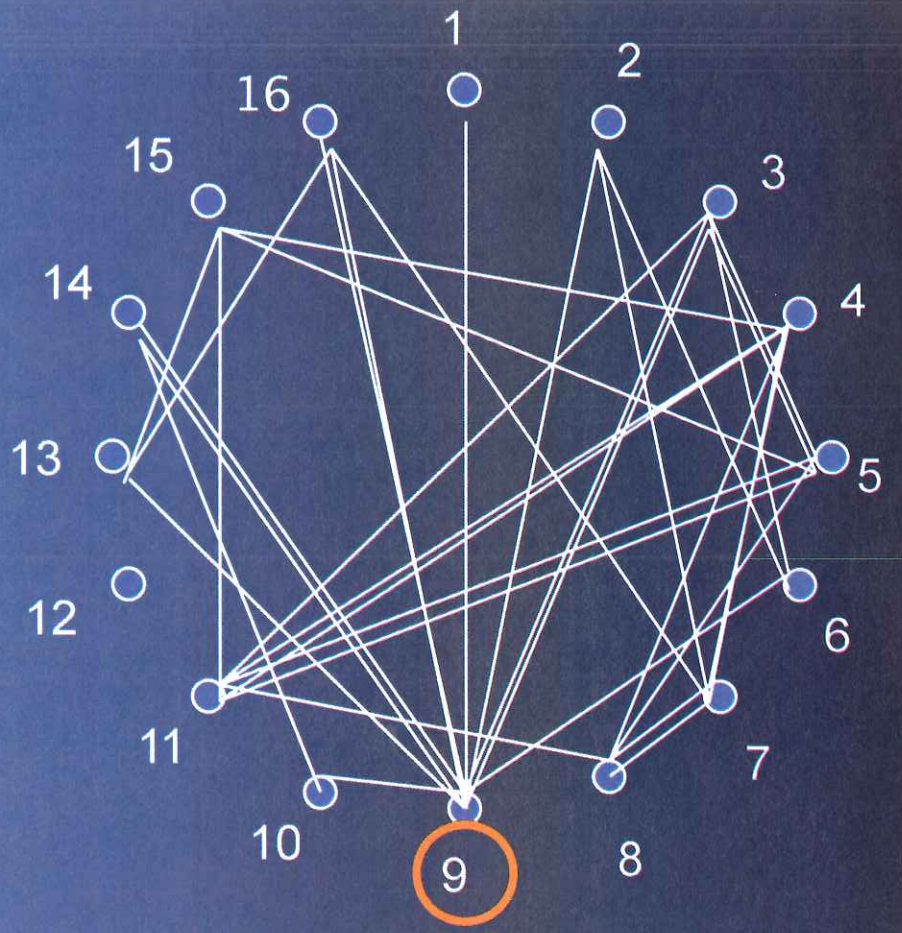
Investigating Network Statistics

Exploration project by Callie McRee and Xinyi Chen

Medici: Marriage and Loans

For our project, we were given data from Florentine families during the Renaissance. The data was organized into a single matrix, a number 2 representing both a marriage and a loan connection and a number 1 representing a single connection. We showed these by either a single or a double edge connecting nodes on the representation to the right.



Degree Distribution

# of links (x axis)	# of nodes with k links (y
1	1
2	1
3	4
4	2
5	2
6	3
7	1
8	0
9	0
10	0
11	1

Degree Distribution

- Our graph resembles a bell curve more than a free-scale network. This means there are less clusters, but still some.
- The majority of families have either three or six connections (four families having three and three families having six).

Diameter

- Def: Diameter is the least amount of steps to get everyone in the network connected.
- To find: begin by making the matrix a boolean matrix. Then, raise it to the next power adding in the previous matrix, keeping it in boolean form. When the entire matrix is filled by 1s, all people in the network are connected.
- The power we raised to (in this case, 4) is the least number of steps to achieve this.
- We had to remove family #12 because they were connected to no other families in the network.

- Final result:
  Boolean(B^4+B^3+B^2+B)
  =[1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1

Average path (geodesic)

- Def: The average amount of steps connecting all nodes.
- Geodesic: the shortest distance between two nodes.
- To find average path length: average the number of new connections in each power. Our answer was 2.08, meaning that on average, nodes could be connected by about 2 steps.

Average path (geodesic)

```
B=[0 0 0 0 0 0 0 0 1 0 0 0
   0 0 0;
   0 0 0 0 0 1 1 0 1 0 0 0
   0 0 0;
   0 0 0 0 1 1 0 0 1 0 1 0
   0 0 0;
   0 0 0 0 0 0 1 1 0 0 1 0
   0 1 0;
   0 0 1 0 0 0 0 1 0 0 1 0
   0 1 0;
   0 1 1 0 0 0 0 0 1 0 0 0
   0 0 0;
   0 1 0 1 0 0 0 1 0 0 0 0
   0 0 1;
   0 0 0 1 1 0 1 0 0 0 1 0
   0 0 0;
   1 1 1 0 0 1 0 0 0 1 0 1
   1 0 1;
   0 0 0 0 0 0 0 0 1 0 0 0
   1 0 0;
   0 0 1 1 1 0 0 1 0 0 0 0
   0 1 0;
   0 0 0 0 0 0 0 0 1 0 0 0
   0 1 1;
   0 0 0 0 0 0 0 0 1 1 0 0
   0 0 0;
   0 0 0 1 1 0 0 0 0 0 1 1
   0 0 0;
   0 0 0 0 0 0 1 0 1 0 0 1
   0 0 0]
```

```
C=B*B+B;
for x=1:15;
  for  y=1:15;
   if (C(x,y)>=1)
      C(x,y)=1;
   end
  end
end
```

```
C=B*B*B+B*B+B;
for x=1:15;
  for  y=1:15;
   if (C(x,y)>=1)
      C(x,y)=1;
   end
  end
end
n=0;
for a=1:15;
for s=1:15;
n=n+C(a,s);
end
end
n
n=213
```

```
C=B*B*B*B+B*B*B+B*B
   +B;
for x=1:15;
  for  y=1:15;
   if (C(x,y)>=1)
      C(x,y)=1;
   end
  end
end
n=0;
for a=1:15;
for s=1:15;
n=n+C(a,s);
end
```

Clustering Coefficient

- Def:

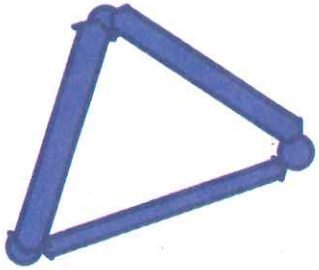
  Triples: any three nodes that are connected

  Open Triples: triples connected by two edges

  Closed triples: triples connected by three edges

  Global Clustering Coefficient: Closed triples divided by all triples.

Open triple



Closed triple

```
B=[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1;
   0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0;
   0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0;
   0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0;
   0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0;
   0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
   0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1;
   0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   1 1 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1;
   0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 1 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]

m=0;
for x=1:15
for y=1:15
 for z=1:15
if (B(x,y)+B(x,z)+B(y,z)
    ==3)
m=m+1;
end
    end
    end
    end
```

Degree Centrality

- Def: the number of edges coming out of each node.
- To find: sum each row
- In our case, the Medici (#9) had the highest number of connections, showing its high centrality in this network.

- Person 1: 1
- Person 2: 3
- Person 3: 6
- Person 4: 6
- Person 5: 6
- Person 6: 3
- Person 7: 6
- Person 8: 5
- Person 9: 11
- Person 10: 2
- Person 11: 7
- Person 12: 0
- Person 13: 3
- Person 14: 3

Closeness Centrality

- Def: the average of the inverse of the shortest path length for each node.
- To find: for a certain node, compute all the distances from this node to all other nodes, take each entry's inverse, and take the average of each row.

Closeness centrality:
Person 1: 0.456
Person 2: 0.567
Person 3: 0.622
Person 4: 0.561
Person 5: : 0.578
Person 6: 0.567
Person 7: 0.589
Person 8: 0.550
Person 9:0 .744
Person 10: 0.489
Person 11: 0.611
Person 12: 0.589
Person 13: 0.489

Closeness Centrality

```
B=[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
   0 0 0 0 0 1 1 0 1 0 0 0 0 0 0;
   0 0 0 0 1 1 0 0 1 0 1 0 0 0 0;
   0 0 0 0 0 0 1 1 0 0 1 0 0 1 0;
   0 0 1 0 0 0 0 1 0 0 1 0 0 1 0;
   0 1 1 0 0 0 0 0 1 0 0 0 0 0 0;
   0 1 0 1 0 0 0 1 0 0 0 0 0 0 1;
   0 0 0 1 1 0 1 0 0 0 1 0 0 0 0;
   1 1 1 0 0 1 0 0 0 1 0 1 1 0 1;
   0 0 0 0 0 0 0 0 1 0 0 0 1 0 0;
   0 0 1 1 1 0 0 1 0 0 0 0 0 1 0;
   0 0 0 0 0 0 0 0 1 0 0 0 0 1 1;
   0 0 0 0 0 0 0 0 1 1 0 0 0 0 0;
   0 0 0 1 1 0 0 0 0 0 1 1 0 0 0;
   0 0 0 0 0 0 1 0 1 0 0 1 0 0 0]

C=B*B+B
for x=1:15
for y=1:15
if (B(x,y)==0)
if (C(x,y)>0)
B(x,y)=2;
 end
end
end
end

D=B
B=[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
   0 0 0 0 0 1 1 0 1 0 0 0 0 0 0;
   0 0 0 0 1 1 0 0 1 0 1 0 0 0 0;
   0 0 0 0 0 0 1 1 0 0 1 0 0 1 0;
   0 0 1 0 0 0 0 1 0 0 1 0 0 1 0;
   0 1 1 0 0 0 0 0 1 0 0 0 0 0 0;
   0 1 0 1 0 0 0 1 0 0 0 0 0 0 1;
   0 0 0 1 1 0 1 0 0 0 1 0 0 0 0;
   1 1 1 0 0 1 0 0 0 1 0 1 1 0 1;
   0 0 0 0 0 0 0 0 1 0 0 0 1 0 0;
   0 0 1 1 1 0 0 1 0 0 0 0 0 1 0;
   0 0 0 0 0 0 0 0 1 0 0 0 0 1 1;
   0 0 0 0 0 0 0 0 1 1 0 0 0 0 0;
   0 0 0 1 1 0 0 0 0 0 1 1 0 0 0;
   0 0 0 0 0 0 1 0 1 0 0 1 0 0 0]

E=B*B*B+B*B+B
for a=1:15
for b=1:15
if (D(a,b)==0)
if (E(a,b)>0)
 D(a,b)=3;
 end
 end
end
end
G=B*B*B*B+B*B*B+B*B+B
for c=1:15
for d=1:15
if (D(c,d)==0)
if (G(c,d)>0)
D(c,d)=4;
end
end
end
end
1./D
```

Closeness Centrality

0.5000 0.5000 0.5000 0.2500 0.3333 0.5000 0.3333 0.2500 1.0000 0.5000 0.3333 0.5000
0.5000 0.3333 0.5000

0.5000 0.5000 0.5000 0.5000 0.3333 1.0000 1.0000 0.5000 1.0000 0.5000 0.3333 0.5000
0.5000 0.3333 0.5000

0.5000 0.5000 0.5000 0.5000 1.0000 1.0000 0.3333 0.5000 1.0000 0.5000 1.0000 0.5000
0.5000 0.5000 0.5000

0.2500 0.5000 0.5000 0.5000 0.5000 0.3333 1.0000 1.0000 0.3333 0.2500 1.0000 0.5000
0.2500 1.0000 0.5000

0.3333 0.3333 1.0000 0.5000 0.5000 0.5000 0.5000 1.0000 0.5000 0.3333 1.0000 0.5000
0.3333 1.0000 0.3333

0.5000 1.0000 1.0000 0.3333 0.5000 0.5000 0.5000 0.3333 1.0000 0.5000 0.5000 0.5000
0.5000 0.3333 0.5000

0.3333 1.0000 0.3333 1.0000 0.5000 0.5000 0.5000 1.0000 0.5000 0.3333 0.5000 0.5000
0.3333 0.5000 1.0000

0.2500 0.5000 0.5000 1.0000 1.0000 0.3333 1.0000 0.5000 0.3333 0.2500 1.0000 0.3333
0.2500 0.5000 0.5000

1.0000 1.0000 1.0000 0.3333 0.5000 1.0000 0.5000 0.3333 0.5000 1.0000 0.5000 1.0000
1.0000 0.5000 1.0000

0.5000 0.5000 0.5000 0.2500 0.3333 0.5000 0.3333 0.2500 1.0000 0.5000 0.3333 0.5000
1.0000 0.3333 0.5000

0.3333 0.3333 1.0000 1.0000 1.0000 0.5000 0.5000 1.0000 0.5000 0.3333 0.5000 0.5000
0.3333 1.0000 0.3333

Without the Medici

# of links	# of nodes with
0	1
1	2
2	4
3	0
4	2
5	1
6	3

Without the Medici

- The graph still doesn't resemble a bell curve exactly. This is mainly because the data set is so small. However, it doesn't resemble a free-scale graph either, showing that there is not very much clust

Without the Medici

- The visual representation of connections in the network without the Medici family is much more disjointed.
- Many families have less connections, and family 1 became an isolated node (12 is as well, but it was even with the Medici family in the