Network Analysis

Introduction to Network Analysis

DIME Analytics
World Bank

Wednesday, the 12th of June, 2024

Welcome

Session Objectives

- Understand fundamental concepts of network analysis.
- Explore how network analysis is applied in tax regulation and compliance.
- Learn to visualize and analyze network data using R.
- Explore the network analysis tool in R.

Why Study Networks?

- Many economic, political and social interactions are shaped by the structure of their relationships.
  - sharing information.
  - trade of goods and services.
  - political alliances, and so on.
- Networks influence behavior.
  - Crime, voting, smoking.
  - Employment.
  - All networks are different, but also have an underlying structure to model.

Importance of Network Analysis

Network analysis provides valuable insights in many fields by:

Relevance of Network Analysis in Tax Regulation

Applying network analysis in tax contexts can enhance oversight by:

- Detecting clusters of tax evasion and fraud.
- Understanding the flow of transactions between entities.
- Identifying key players and influencers in tax evasion schemes.

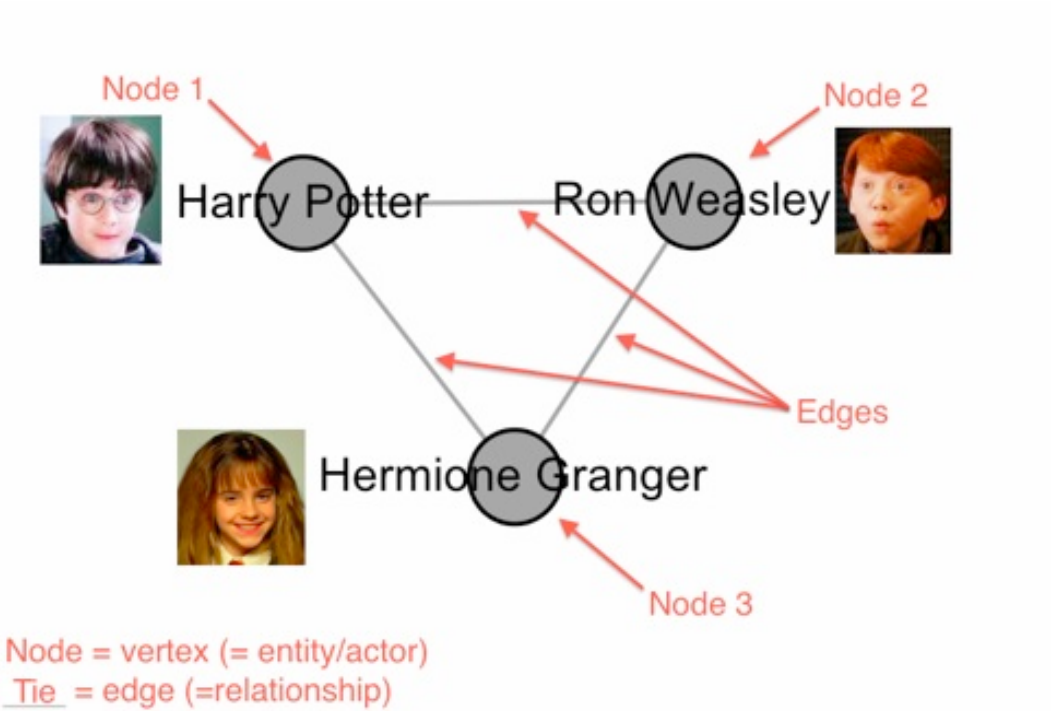
Tax authorities can use this information to refine audit targets and improve compliance strategies.

Network in economics

- Social networks: Connections between individuals
- Trade and Taxes: Flows between economic agents
- Space: Distance between units

First of all … What is a network?

- Structures made up of entities and their relationships/links.
- What happens to an entity is dictated by their position and the structure of the connections.
- This determines what will reach him and what will not.



Basic representation: Matrix

- Matrices and graphs can be used to visualize this structure.
- This is only the arrangement of our data. For instance, for the previous network graph we have the following matrix.

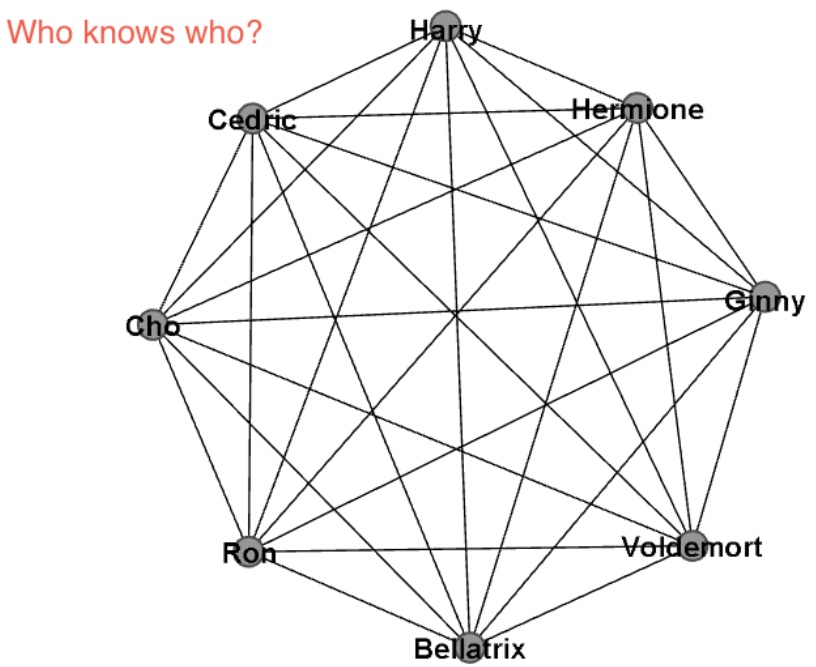
	Harry Potter	Hermione Granger	Ron Weasley
Harry Potter	0	1	1
Hermione Granger	1	0	1
Ron Weasley	1	1	0

This matrix is used to represent the presence (1) or absence (0) of a direct link between characters.

Types of links

Undirected

- Edges do not have a direction. They represent mutual or bidirectional relationships.
- Common uses include representing family ties, friendships, or connections within drug cartels.

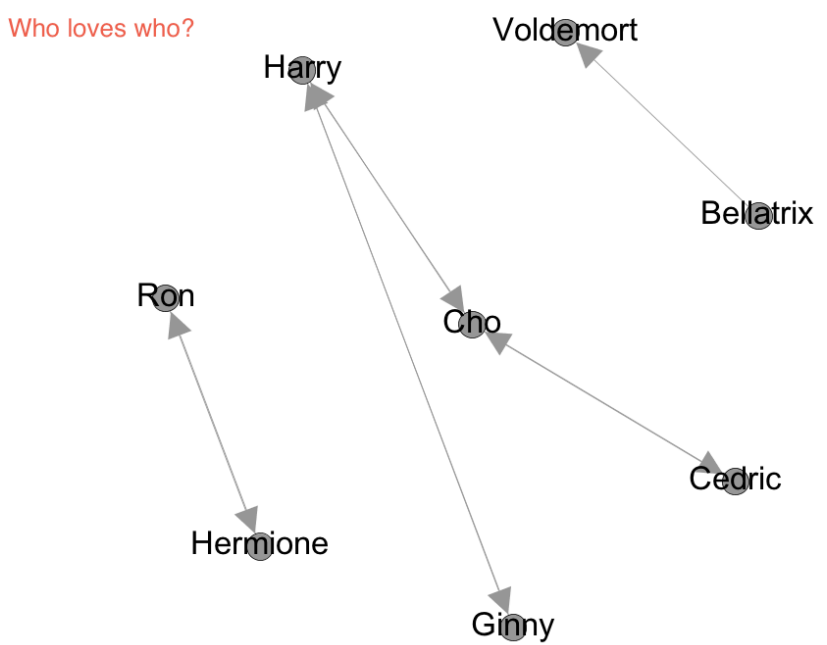


Undirected Network

Types of links

Directed

- Edges have a direction indicating the flow of information or resources.
- Examples include Twitter follower graphs, trade between countries, or financial transactions, shareholders of firms.



Directed Network

Types of links

Weighted and unweighted

- How much time are A and B together weekly? vs. Are A and B friends?

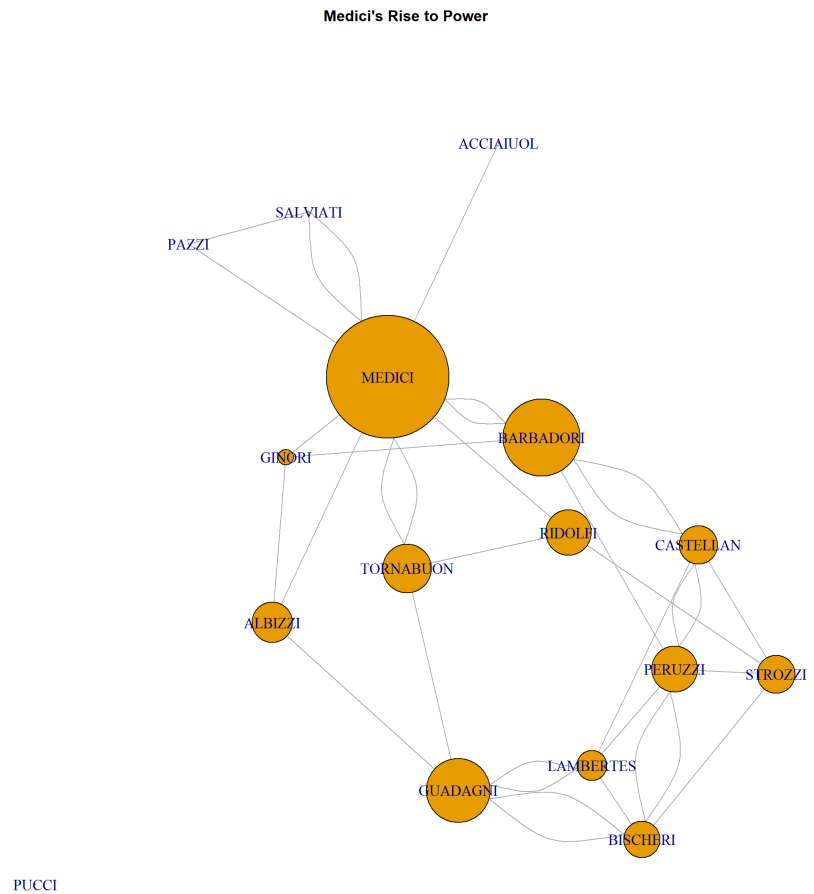
May have multiple (inconsistent) observations of the same link

- A reports whether B is their friend, B reports whether A is their friend
- A reports whether B is their friend at baseline and at endline

Examples

- Medici's rise to power is a very good example of network analysis.
- Padgett and Ansell use network analysis to examine the rise of the Medici family to the height of Florentine politics, where the family came to run the city-state for several generations.
- Scholars argued that it was through tact and skill on the part of the patriarch of the family.
- Through network analysis, Padgett and Ansell show that it was actually due to the Medici family accidentally being in the center of Florentine politics.

Examples: Medici



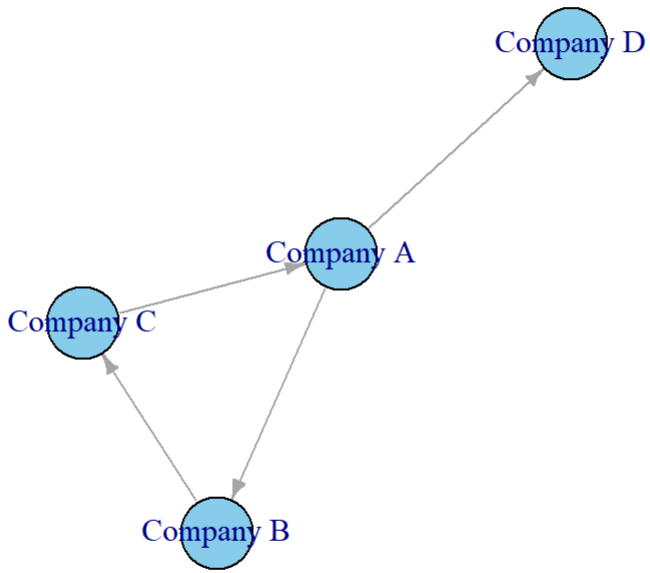
We will come back to this example later on the presentation.

Examples: What does it have to taxes?

- Application in Tax Regulation: Identifying relationships between taxpayers, companies, and financial transactions.

- We will start introducing a simple company example to show this (we will use this imaginary example thorught the presentation), and then some real-life examples.

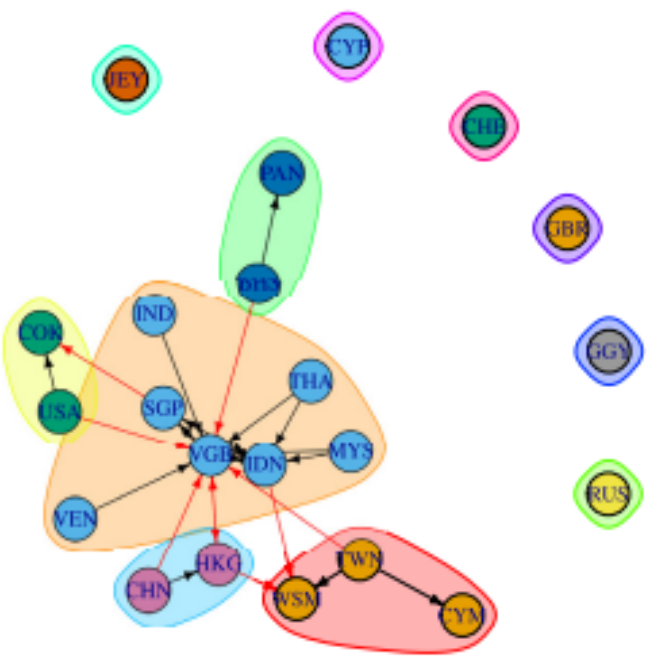
Basic Network Diagram



Real Examples: Panama Papers' Offshoring Network Behavior

Study Overview

- **Focus**: Investigation of the offshoring network exposed by the Panama Papers.
- **Objective**: To uncover hidden structures and connections in global offshoring activities.



Real Examples: Panama Papers' Offshoring Network Behavior

Key Findings

- **Critical Nodes**: Identification of significant countries and their roles within the intricate global network.
- **Network Dynamics**: In-depth characterization of connectivity, showing how entities influence and interact within the financial secrecy framework.
- **Implications for Regulation**: Understanding these connections helps regulators and policymakers tailor interventions and policies to combat financial secrecy effectively.

Example: Heard it Through the Grapevine

Study Overview

- **Focus**: Examination of the direct and network effects of a tax enforcement field experiment on firms.
- **Objective**: To assess how interventions targeted at specific nodes affect the broader network.

Key Findings

- **Network Dynamics**: Enforcement actions against specific firms result in widespread changes in compliance behaviors throughout their networks.
- **Implications for Regulation**: Insights from this study guide tax authorities in crafting more focused and effective tax enforcement strategies, potentially increasing compliance and revenue.

Network Analysis: Understanding the Challenge

Now that we know the basics, let's understand the challengues. One of the main challenges in network analysis is the complexity.

- Imagine a 30 nodes-network. The potential network configurations—ranging from empty to fully connected—are enormous.
  - Calculating possible friendships leads to 435 potential links ($\binom{30}{2}$).
  - Each link can either exist or not, leading to 2^{435} possible networks > atoms in the universe

Network Analysis: Understanding the Challenge

Managing the complexity

1. **Graphical Models & Matrices**:
    - Simplify visualization and enable mathematical analysis, making it easier to represent and study network structures.
2. **Focus on Key Metrics**:
    - Emphasize crucial network metrics such as degree, centrality, connectivity, and clustering. These metrics highlight important network features without needing to consider every possible configuration.
3. **Visualization Tools**:
    - Use tools that allow for intuitive understanding of complex network dynamics through graphical representations.

We will focus on 2 and 3 during this presentation.

Simplifying the Complexity: Network Metrics

Degree

- The **degree** of a node in a network represents the number of connections it has to other nodes. This can be further categorized into:

1. Indegree: The number of incoming edges to a node.
2. Outdegree: The number of outgoing edges from a node.

Average degree

The average degree of a network is calculated as the total number of edges divided by the total number of nodes. It's a simple yet powerful measure to understand the connectivity of the network.

Total Edges/Total Nodes=Average Degree.

Average degree: use

Let's consider two hypothetical tax groups within the same industry:

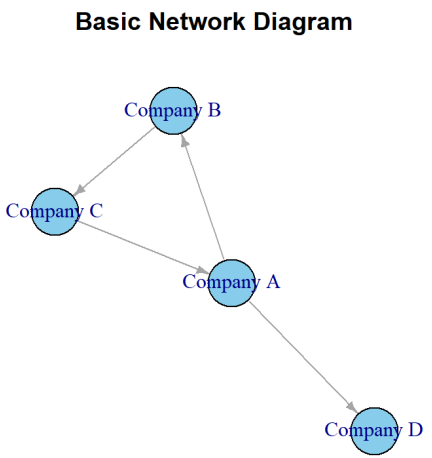
- Group A: 10 businesses with 50 transactions. 50/10 = 5
- Group B: 10 businesses with 20 transactions. 20/10 = 2

Interpretation:

- Group A has a higher average degree -> more interconnected network. Vibrant business ecosystem but sophisticated financial structures potentially obscuring true economic activities.
- Group B has a lower average degree -> indicating less interconnectivity. Simpler financial structure, potentially making it easier for tax authorities to audit.

Degree and Average Degree

The degree of a node indicates the number of direct connections. Average degree is a measure of the overall connectivity of the network.




```r
# Calculate degrees
node_degrees <- degree(graph)

# Calculate average degree
average_degree <- mean(degree(graph))
```

```
## Degrees of each node:
## Company A Company B Company C Company D
##         3         2         2         1
## Average degree: 2
```

Network Metrics: Path Length

- **Path Length** refers to the number of edges in the shortest path between two nodes in a network. Steps required to connect two points.

Importance of Path Length:

- **Efficiency**: Shorter path lengths generally indicate more efficient networks where information or resources can travel more quickly between nodes.

Example:

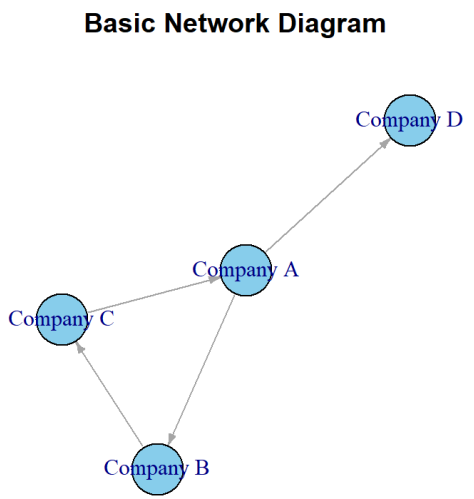
In a school communication network, if the path length between the principal and any teacher averages to 2, it suggests that information can quickly and efficiently reach any teacher either directly or through one intermediary.

Network Metrics: Geodesic Distance

- The length of the shortest path between two nodes.
- Going back to our company example…

Geodesic distance: example

1. What's the geodesic distance between A and the rest of the firms?
2. What's the geodesic distance between B and D?




```
1 distances <- distances(graph)
```

```
##           Company A Company B Company C Company D
## Company A         0         1         1         1
## Company B         1         0         1         2
## Company C         1         1         0         2
## Company D         1         2         2         0
```

Network Metrics: centrality

- Centrality is one of the most relevant indicators to understand a network.
- How should we define the idea of centrality? We might imagine that someone "central" to the network is someone who holds some sort of important position -> Many ways we can think of centrality.

Degree centrality

- The degree of a node is the number of edges that start from or point to a node.
- Again, with the previous example A has degree 3.
- Now, degree centrality shows the number of nodes another node could possibliy connect with in the network.
- It is the most commonly used centrality measure.

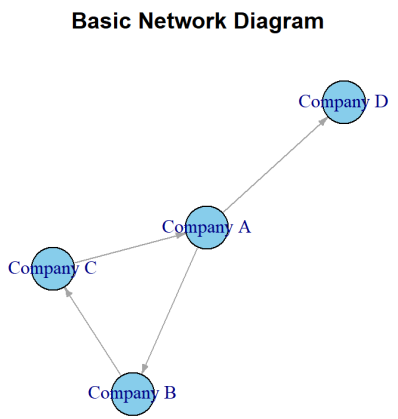
Degree & degree centrality: code

```r
# Calculate degrees (in-degree + out-degree in a directed graph)
node_degrees <- degree(graph)

# Calculate degree centrality
# For directed graphs, it's often useful to consider in-degree and out-degree separately
in_degree_centrality <- degree(graph, mode = "in") / (vcount(graph) - 1)
out_degree_centrality <- degree(graph, mode = "out") / (vcount(graph) - 1)

# Company A is connected with every node in the graph -> 1 (100%)

total_degree_centrality <- node_degrees / (vcount(graph) - 1)
```

```
## Degrees of each node:
## Company A Company B Company C Company D
##         3         2         2         1
##
## In-Degree Centrality:
## Company A Company B Company C Company D
## 0.3333333 0.3333333 0.3333333 0.3333333
##
## Out-Degree Centrality:
## Company A Company B Company C Company D
## 0.6666667 0.3333333 0.3333333 0.0000000
##
## Total Degree Centrality:
## Company A Company B Company C Company D
## 1.0000000 0.6666667 0.6666667 0.3333333
```

Network Metrics: Betweeness centrality

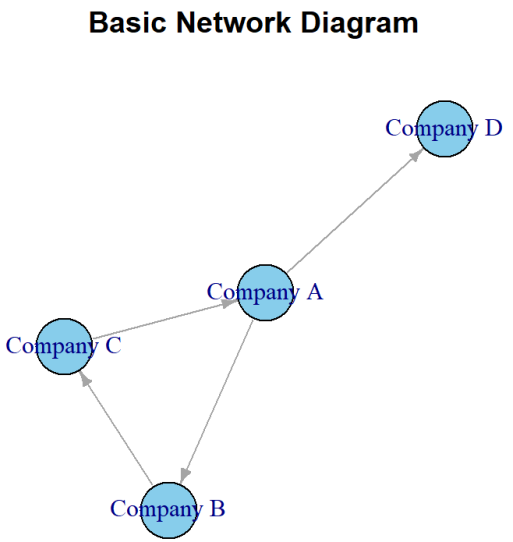
- Here we consider how often a node lies on the shortest path between two other nodes.




```
# Calculate betweenness centrality for all nodes
betweenness_centrality <- betweenness(graph, v=V(graph), directed=TRUE)

# Extract betweenness centrality for Company A and Company B
betweenness_A <- betweenness_centrality["Company A"]
betweenness_B <- betweenness_centrality["Company B"]
```

- Company A: 3
- Company B: 1

Betweeness centrality: example



- This indicates that “Company A” plays a more significant role in connecting different parts of the network. This could mean that “Company A” is crucial for the flow of information or resources and could be a strategic target for further analysis or monitoring.
- Companies with high betweenness centrality can be seen as having strategic importance within the network.

Network Metrics: Exploring Advanced Types of Centrality

K-Path Centrality

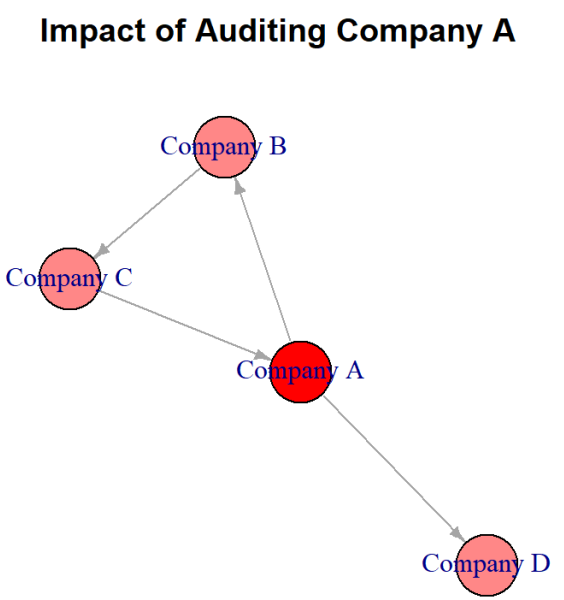
- **Definition**: Measures the closeness of a node to all other nodes within ‘k’ steps, beyond direct connections.
- **Insight**: Evaluates how effectively a node can influence or reach others within a few intermediaries. Good for large nertworks.

Eigenvector Centrality

- **Definition**: A measure where a node’s importance is enhanced by being connected to other highly important nodes.
- **Insight**: Considers the quality of connections.

How Can a Tax Authority Use This for Audits?

- Consider a scenario where a tax authority decides to audit ‘Company A’ based on suspicious transaction patterns.
- This audit impacts not only Company A but also its direct connections within the network, such as Companies B,C,D, due to their transactional relationships.



Network Metrics: Clustering

Clustering Coefficient is a measure of the degree to which nodes in a network tend to cluster together. It reflects the likelihood that two neighbors of a node are also neighbors of each other.

Importance of Clustering:

- **Community Structure**: High clustering coefficients often indicate a strong presence of community-like structures within the network.
- **Resilience**: Networks with high clustering may be more resilient to disruptions, as there are multiple pathways for reconnecting nodes.

Clustering: Example

In social networks, a high clustering coefficient might be observed in a group of friends who all know each other. This can imply a tightly-knit community where information or influences spread rapidly among members.

Using our company's network.

Clustering: code

```
1  clust_coeff <- transitivity(graph, type = "localaverage")
2  cat("Clustering Coefficient:", clust_coeff)
```

Clustering Coefficient: 0.7777778

- The neighborhoods in this network are quite densely connected.
- This means that if we pick a node randomly, there is a 77.78% chance that two of its neighbors are also connected to each other.

Network metrics: Modularity

- **Modularity** is a measure used to assess the strength of division of a network into modules or communities.
- High modularity indicates that there are dense connections within modules but sparse connections between them, which can be particularly insightful for tax authorities.

Importance of Modularity:

- **Fraud Detection**: Helps identify clusters of taxpayers or businesses that may be engaging in similar types or patterns of tax evasion or avoidance.
- **Audit Efficiency**: Facilitates targeted audits by recognizing distinct groups that may require different handling or investigation strategies.

Calculating Modularity with the Louvain Method

The Louvain method is a popular algorithm used for detecting communities in large networks because it optimizes the modularity score during community detection.

Example Code:

Here is how you can calculate modularity using the Louvain method in R:

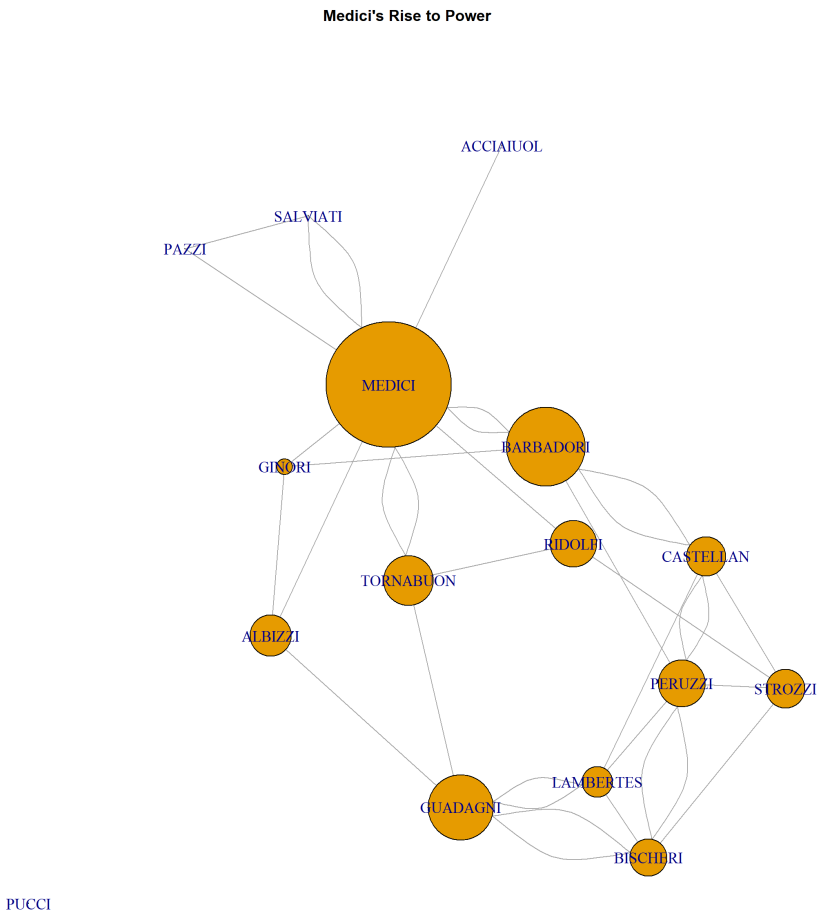
```
cluster_louvain()
```

Real-Life Implications Consider a scenario where a network of businesses shows high modularity:

- Sector Specific Clusters: Businesses in similar sectors or engaging in similar strategies may form distinct clusters.
- Tailored Audits: Understanding these clusters allows tax authorities to tailor their audit strategies more effectively, potentially focusing on sectors prone to specific compliance issues or tax avoidance strategies.

Practice: Hands-on

Let’s test these concepts using our Medici example.



Practice: Introduction

Objective: We will use our Medici data to learn to apply basic network analysis techniques using real-world datasets.

Tools Needed: RStudio, `igraph` package.

Duration: 25 minutes

Data source: I will use our Medici example, download it from here.

⏰ For later, you can also try the code using other data from here.

1. Setup Your Environment

Install R Packages

Ensure `igraph` and `scales` is installed and load it:

Download data

- Visit Historical Networks - SKuDe and download Medici dataset here.
- The .zip will contain different files, we will need edges and nodes.

Load data in R

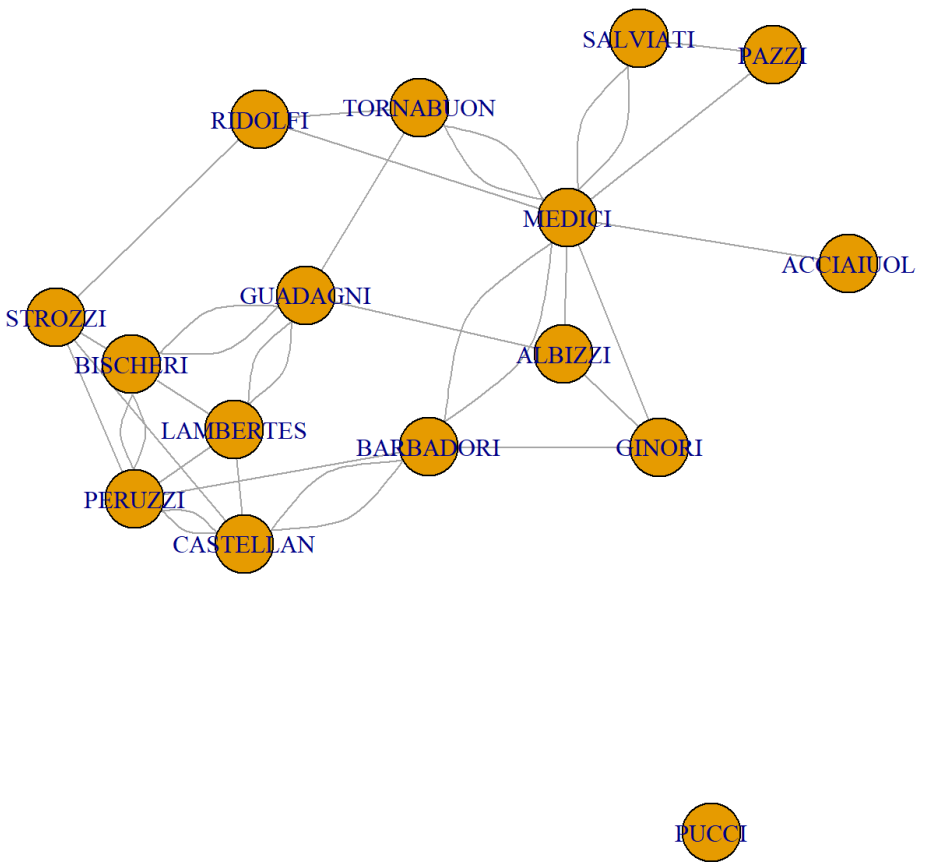
```
1  edges <- read.csv("path_to_your_edges_file.csv")
2  nodes <- read.csv("path_to_your_nodes_file.csv")
```

2. Create Network Graph

Let's start with the simplest version of a network graph.

```r
g <- graph_from_data_frame(# set the edges
                           d=edges,
                           # set the nodes
                           vertices=nodes,
                           # directed or undirected
                           directed=FALSE)

plot(g)
```

2. Create Network Graph



3. Compute Network Metrics

- In this section, we will compute the different network metrics using the provided R code snippets.
- Reflect on how each metric might indicate the Medici family's strategic positioning and influence within the network.

1. Degree Centrality

This metric will help us understand which families had the **most connections** within the network.

```r
# Calculate degree centrality
V(g)$degree <- degree(g)
# Display degree centrality for all nodes
degree(g)
```

```
## ACCIAIUOL   ALBIZZI BARBADORI  BISCHERI CASTELLAN    GINORI  GUADAGNI LAMBERTES
##         1         3         6         6         6         3         6         5
##    MEDICI     PAZZI   PERUZZI     PUCCI   RIDOLFI  SALVIATI   STROZZI TORNABUON
##        11         2         7         0         3         3         4         4
```

3. Compute Network Metrics

2. Eigenvector Centrality

Nodes with high eigenvector centrality are those that are connected to many nodes who themselves have high scores.

```
# Calculate eigenvector centrality
V(g)$eigenvector <- eigen_centrality(g)$vector
# Display eigenvector centrality for all nodes
round(V(g)$eigenvector, 3)
```

```
## ACCIAIUOL   ALBIZZI BARBADORI  BISCHERI CASTELLAN    GINORI  GUADAGNI LAMBERTES
##     0.153     0.345     0.868     0.838     0.894     0.369     0.711     0.739
##    MEDICI     PAZZI   PERUZZI     PUCCI   RIDOLFI  SALVIATI   STROZZI TORNABUON
##     0.860     0.214     1.000     0.000     0.338     0.344     0.546     0.493
```

- **Quality** of the connection. Nodes with high eigenvector centrality are those that are connected to many nodes who themselves have high scores.
- For example, Peruzzi has the highest score of 1.0, indicating it is extremely well-connected within the network to other highly connected nodes.

3. Compute Network Metrics

3. Betweeness centrality

Shortest path -> A node plays a **crucial role in facilitating communication** across the network.

```
1  V(g)$betweenness <- betweenness(g)
```

```
## ACCIAIUOL   ALBIZZI BARBADORI  BISCHERI CASTELLAN    GINORI  GUADAGNI LAMBERTES
##  0.000000  4.803743 17.478788  3.910042  4.335294  0.753268 12.125758  2.688800
##    MEDICI     PAZZI   PERUZZI     PUCCI   RIDOLFI   SALVIATI   STROZZI TORNABUON
## 44.082739  0.000000  6.290820  0.000000  6.128788  0.000000  4.312121  7.089840
```

- MEDICI (44.08): This node has the highest betweenness centrality, suggesting it frequently lies on the shortest paths between other nodes.
- Families with high betweenness centrality could control or influence the spread of information or resources across the network.

3. Network Metrics

4. Clustering coefficient

The clustering coefficient measures the likelihood that a node's neighbors are also connected to each other.

```
library(igraph)

# Calculate local clustering coefficients
V(g)$clustering <- transitivity(g, type = "local")

# Create a dataframe to display node names alongside their clustering coefficients
clustering_df <- data.frame(
  Node = V(g)$name,
  ClusteringCoefficient = round(V(g)$clustering, 3)
)
```

3. Network Metrics

4. Clustering coefficient

Node	ClusteringCoefficient
ACCIAIUOL	NaN
ALBIZZI	0.333
BARBADORI	0.333
BISCHERI	0.500
CASTELLAN	0.500
GINORI	0.667
GUADAGNI	0.167
LAMBERTES	0.500
MEDICI	0.143
PAZZI	1.000
PERUZZI	0.500

Node	ClusteringCoefficient
PUCCI	NaN
RIDOLFI	0.333
SALVIATI	1.000
STROZZI	0.333
TORNABUON	0.333

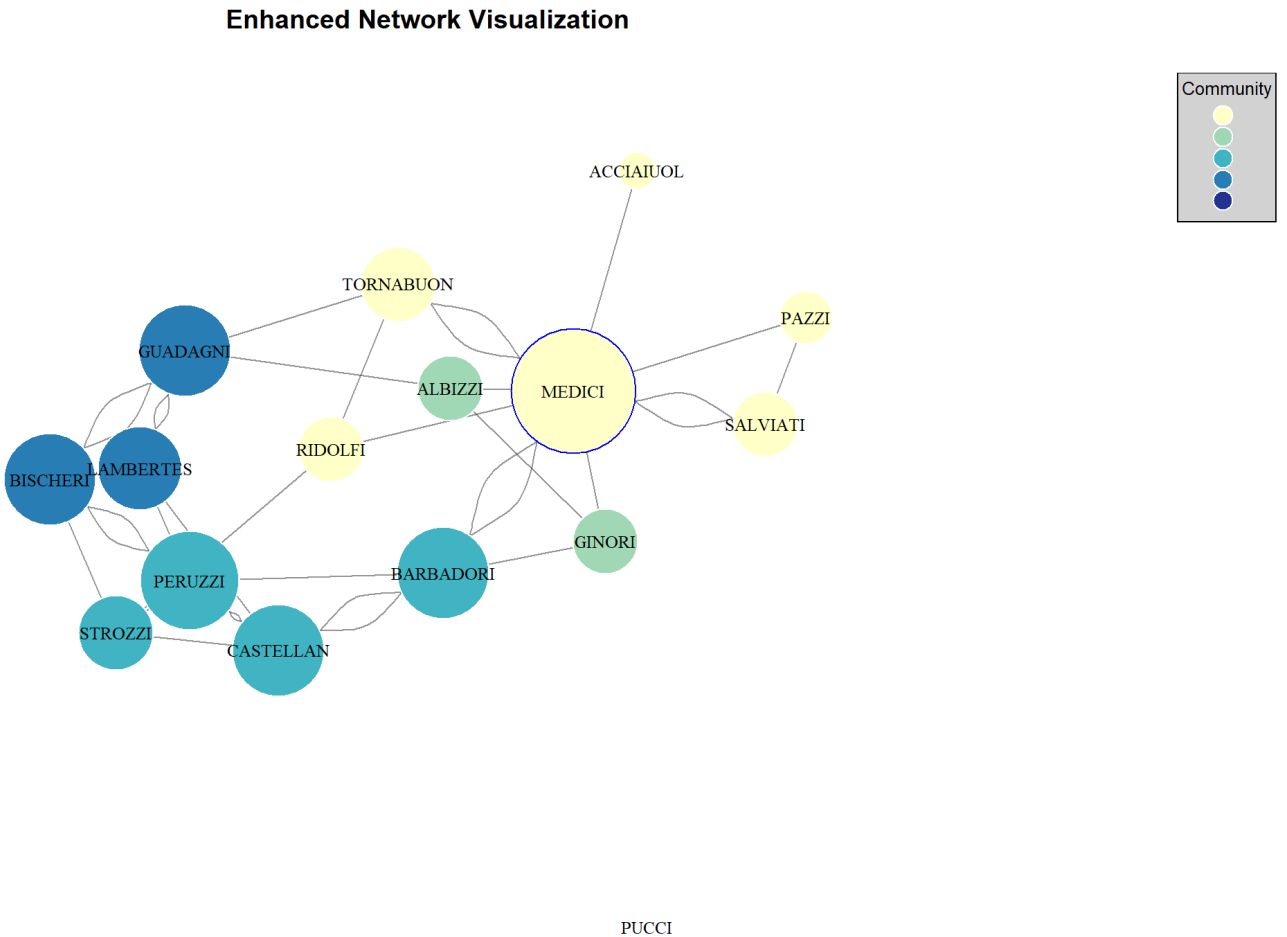
Understanding which families formed close-knit groups can reveal sub-communities within the network, showing how family alliances or rivalries might have shaped social dynamics.

4. Network Visualization

Now, we will use everything we have computed to enhance our network visualization.

```r
library(scales)
library(igraph)
# Create the graph from the edge list
g <- graph_from_data_frame(edges, vertices=nodes, directed=FALSE)

# Calculate metrics
V(g)$degree <- degree(g)
V(g)$betweenness <- betweenness(g)
V(g)$eigenvector <- eigen_centrality(g)$vector
V(g)$clustering <- transitivity(g, type = "local")

# Normalize metrics for visualization purposes
V(g)$degree_size <- sqrt(V(g)$degree) * 10  # Adjust size factor for better visibility
V(g)$betweenness_color <- rescale(V(g)$betweenness, c(0, 1))
V(g)$eigenvector_size <- rescale(V(g)$eigenvector, c(1, 10))

# Define colors for communities using a sophisticated color palette
community_colors <- RColorBrewer::brewer.pal(max(membership(cluster_louvain(g))), "YlGnBu")

# Set seed for reproducible layout
set.seed(123)

# Plot the network
plot(g, layout=layout_with_fr,  # Fruchterman-Reingold layout
     vertex.size = V(g)$degree_size,  # Size nodes by normalized degree
     vertex.color = community_colors[membership(cluster_louvain(g))],  # Color nodes by community
```

4. Network Visualization



5. Why Medici Rose to Power?

After analyzing various network metrics, we can draw several conclusions about the strategic positioning and influence of the Medici family within the Florentine social network.

1. **Control and Influence (Betweenness Centrality)**
    - The Medici's high betweenness centrality indicates their crucial role as intermediaries in the network. They often acted as the primary pathway through which information and resources flowed.
2. **Gatekeeping (Degree Centrality)**
    - With a high degree centrality, the Medici had numerous direct connections, positioning them as gatekeepers within the network.
3. **Strategic Partnerships (Eigenvector Centrality)**
    - The Medici's high eigenvector centrality underscores their connections to other influential nodes (families) within the network.

Using Netowrk Visualization Tool

Objectives

In this final part of our presentation, we aim to:

- **Effectively navigate** the application and utilize its features to analyze network data.
- **Understand** the dynamics within corporate networks through interactive visualizations.

Live Demonstration: Exploring the Network Graph

Demonstration:

- First, I will show you how the app is structured, specifically where the network graph functionality is implemented. This will give you insight into how the data flows and is visualized.

Exercise: Applying Network Analysis

Now that you are familiar with how the tool works, it's your turn to apply the concepts of network analysis using our app.

Exercise Steps

1. **Input ID Search**:
    - **Action**: Enter the ID of a company you suspect might be central in the network.
    - **Tip**: Opt for companies that are large, have widespread operations, or are at the core of industry sectors.
    - **Note**: Avoid companies with excessively large connections as the app may take longer to load.
2. **Analyze the Network Graph**:
    - **Observation**: Use the interactive graph to examine the connections and the positioning of the selected company.
    - **Features**: Hover over nodes to see direct connections and click on nodes to get detailed information.

R Code for Identifying Central Nodes

To assist you in identifying central nodes within the network, use the following R code snippet in your local R environment or a setup that supports R:

```r
library(igraph)
library(tidyverse)

# Assuming 'graph' is your network graph object
# Calculate degree centrality
degree_centrality <- degree(graph, mode = "all")

# Sort nodes by centrality to find the most central nodes
central_nodes <- sort(degree_centrality, decreasing = TRUE)

# Display the top central nodes
print(names(central_nodes)[1:5])  # Adjust number as needed
```

Thank You!

Thank you for your attention and participation.

Any Questions?

Please feel free to ask any questions!

Acknowlegements

- This presentation was created using the DIME template DIME Template
- Also, we thank John Loeser for his help in the materials.
- Yanne Broux and Silke Vanbeselaere for the Harry Potter examples
- Social and Economic Networks: Online. Stanford University. Professor Matthew O. Jackson. Coursera
- Data for the Medici example from Network Catalog

Useful Resources

Want to go further? Here are some great resources to get you started:

Books and Online Materials

- **"Networks, Crowds, and Markets: Reasoning About a Highly Connected World"** by David Easley and Jon Kleinberg
  - A foundational text offering insights into the intersection of networks, economics, and social sciences. Consult here
- **"Social Network Analysis for Startups"** by Maksim Tsvetovat and Alexander Kouznetsov
  - An introduction to social network analysis with practical applications using Python and R.

Useful Resources: Online Courses

- **Social Network Analysis (SNA) Course** on Coursera
  - Offered by the University of Michigan, this course covers techniques and R programming for analyzing social networks.
  - Social Network Analysis Course on Coursera
- **Network Analysis in R** on DataCamp
  - Focuses on using R for network analysis, covering everything from basic network concepts to advanced network analysis techniques.
  - Network Analysis in R on DataCamp

Useful Resources: Websites and Online Tools

- **igraph Documentation**
  - Comprehensive guide and reference for using the `igraph` R package, one of the most popular tools for network analysis in R.
  - igraph Package Documentation
- **Statnet**
  - `Statnet` is a suite of R packages for network analysis that includes tools for network creation, visualization, and analysis.
  - Statnet Project Website
- Introduction to R for Data Science: A LISE 2020 Guidebook. Chapter 7