Session 6: Social Networks

Welcome!

TA: Jennifer Lin

MMSS 211: Institutions, Rules, & Models in Social Science

2022-05-03

Networks

You've likely experienced times in your life where you meet someone who is ostensibly a stranger but after a brief conversation, you realize you have a mutual friend. You might have remarked "What a small world!"

Our daily interactions is full of things that can be represented by networks, which can help us better understand how the world works.

Humans are social creatures. We make connections and these links dictate many of the decisions we make in our daily lives.

We also connect things in our world through semantics and shared meanings.

Goals

1. Understand basic network structure
2. Learn how to generate network data
3. Become familiar with graphing networks in R using base R and ggplot2

The Data -- The Office

Network Structures

Nodes

- **Clustering**: The probability that two associates of a node are linked.
- **Centrality**: How concentrated the links are around a small number of nodes.

Network Structures

Edges

- **Path Length**: Number of steps it takes to connect a pair of nodes. Directly connected nodes have a path length of 1.
- **Degree**: Number of connections. In-Degree is the number of connections sent to a node and Out-Degree is the number of connections sent from a node.
- **Betweeness**: The extent to which a node lies between other nodes. Higher betweeness signals that nodes are more dependent on others for access to information or valued goods.
- **Directed** and **Undirected**: Directed networks occur when out-degrees are not reciprocated while undirected networks are reciprocated.
- **Density**: Ratio of ties in a network to the total number of possible ties

Adjacency Matrix

An **adjacency matrix** is a square matrix with the names of the nodes as row names and as column names.

The content in the body shows whether the two notes are connected (coded as 1) or not (coded as 0).

```
## 5 x 5 sparse Matrix of class "dgCMatrix"
##         Ann Bob Charlie David Emilia
## Ann       .   1       .     1      1
## Bob       1   .       1     .      .
## Charlie   .   1       .     .      1
## David     1   .       .     .      .
## Emilia    1   .       1     .      .
```

Data Structures for Networks

When plotting networks in R, we can use adjacency matrices (See previous slide) or Node and Edge lists.

Node and Edge lists are dataframes that show data for each node (node list) and how they are connected (edge list)

Node List

```
person <- c("Ann", "Bob", "Charlie", "David", "Emilia", "Fiona", '
major <- c("Econ", "Politial Science", "Econ", "Psychology", "Pol
gender <- c("Female", "Male", "Male", "Male", "Female", "Female",
nodelist <- data.frame(person, major, gender)
nodelist
```

```
##     person                              major gender
## 1      Ann                               Econ Female
## 2      Bob                    Politial Science   Male
## 3  Charlie                               Econ   Male
## 4    David                         Psychology   Male
## 5   Emilia                   Political Science Female
## 6    Fiona Psychology + Political Science Female
## 7    Gabby                   Political Science Female
## 8    Henry                      History + Econ   Male
## 9      Ian                             History   Male
## 10    Jake                             History   Male
## 11    Kyle               Psychology + History   Male
```

Edge List

```
person <- c("Ann", "Bob", "Charlie", "David", "Emilia", "Fiona", '
target <- c("Charlie", "Emilia", "Henry", "Fiona", "Fiona", "Gabby
edgelist <- data.frame(person, target)
edgelist
```

```
##     person  target
## 1      Ann Charlie
## 2      Bob  Emilia
## 3  Charlie   Henry
## 4    David   Fiona
## 5   Emilia   Fiona
## 6    Fiona   Gabby
## 7    Gabby     Bob
## 8    Henry     Ian
## 9      Ian    Jake
## 10    Jake   Henry
## 11    Kyle   David
## 12    Kyle    Jake
```

Reading in the Data

I prepared some data that converts the episode script to a network of co-appearances by scene. We can read it in as follows:

```
Season_edge_Weight <- read.csv("Edges/Halloween.csv") %>%
  select(-X)
```

from	to	weight
Angela	Devon	1
Angela	Kevin	2
Angela	Michael	2
Angela	Oscar	2
Angela	Pam	1
Children	Kid	1

Here is a Node List

```
Season_nodes <- Season_edge_Weight %>%
  gather(position, person, from:to) %>%
  distinct(person, .keep_all = FALSE) %>%
  mutate(
    department = case_when(
      person %in% c("Angela", "Oscar", "Kevin") ~ "Accounting",
      person %in% c("Jim", "Dwight", "Stanley", "Phyllis") ~ "Sale
      person == "Pam" ~ "Reception",
      person %in% c("Michael", "Jan", "Craig", "Dan",
                    "DavidWallace") ~ "Management",
      TRUE ~ "Other Department"
    ),
    color = case_when(
      department == "Accounting" ~ "green4",
      department == "Sales" ~ "blue",
      department == "Reception" ~ "yellow",
      department == "Management" ~ "orange",
      department == "Other Department" ~ "purple"
    )
  )
```

person	department	color
Angela	Accounting	green4
Children	Other Department	purple
Creed	Other Department	purple
Devon	Other Department	purple
Dwight	Sales	blue
Hank	Other Department	purple

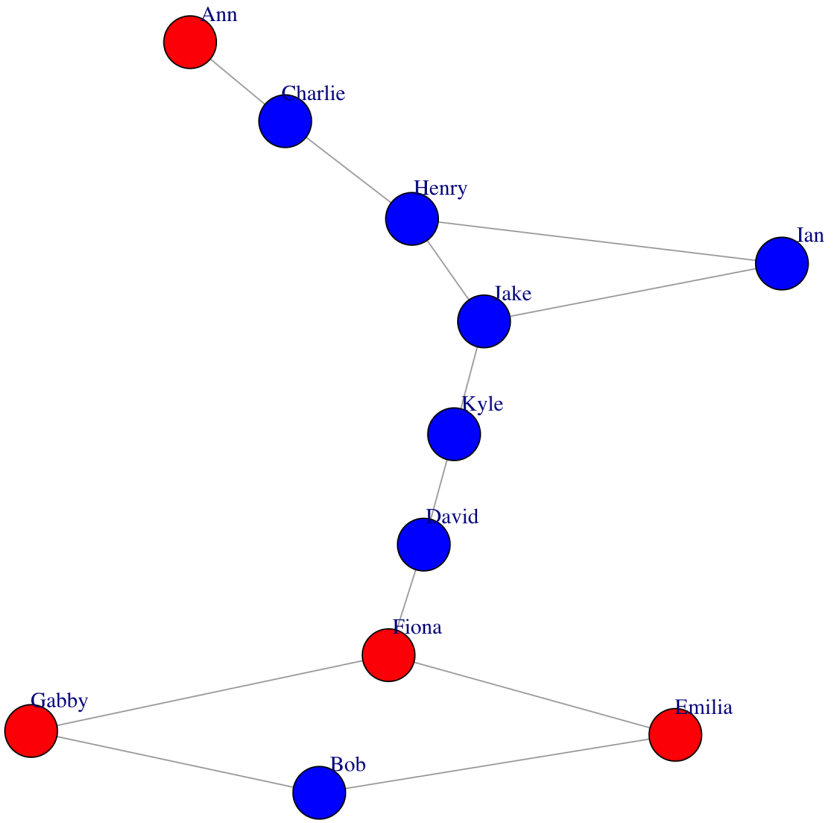
What Network?

Before generating networks, write down

1. Nodes
2. Edges
3. Weighted or unweighted?
4. Directed or undirected?

Generating Network Statistics with `igraph`

```
g <- graph_from_data_frame(edgelist, directed=FALSE, vertices=node
V(g)$color <- ifelse(nodelist$gender == "Female", "red", "blue")
plot(
  g, vertex.color=V(g)$color,
  vertex.size = 14, vertex.label.dist=1.5,
  vertex.label.cex = 1)
```

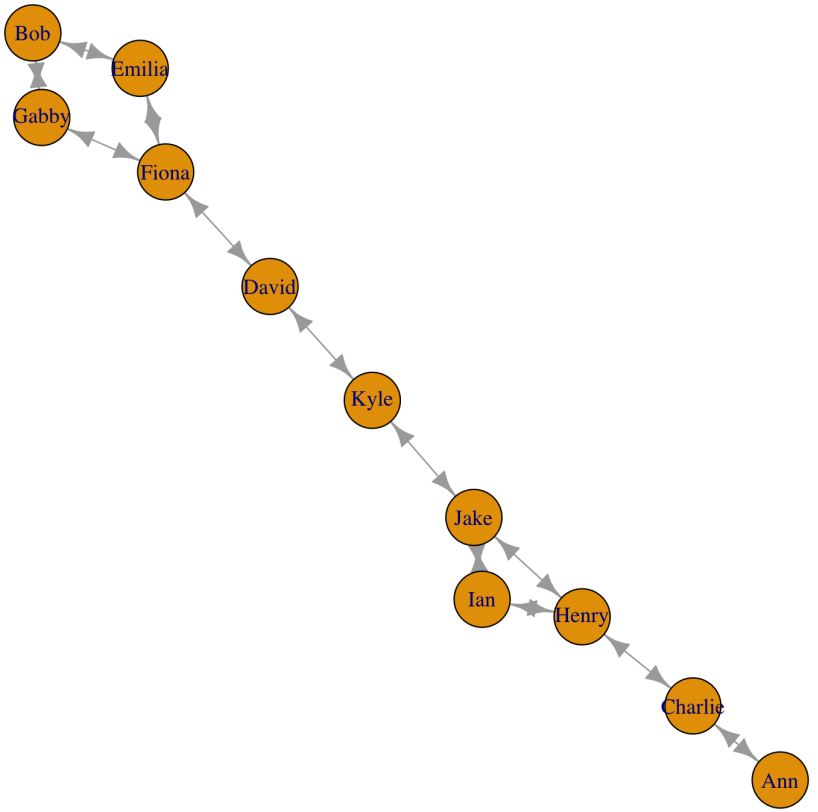

Ann
Charlie
Henry
Ian
Jake
Kyle
David
Fiona
Gabby
Emilia
Bob

Graph from Adjacency Matrix

```
sample_edges <- get.adjacency(g) %>%
  graph_from_adjacency_matrix()
sample_edges
```

```
## IGRAPH 7c97887 DN-- 11 24 --
## + attr: name (v/c)
## + edges from 7c97887 (vertex names):
##  [1] Charlie->Ann     Emilia ->Bob     Gabby  ->Bob
##  [4] Ann    ->Charlie Henry  ->Charlie Fiona  ->David
##  [7] Kyle   ->David   Bob    ->Emilia  Fiona  ->Emilia
## [10] David  ->Fiona   Emilia ->Fiona   Gabby  ->Fiona
## [13] Bob    ->Gabby   Fiona  ->Gabby   Charlie->Henry
## [16] Ian    ->Henry   Jake   ->Henry   Henry  ->Ian
## [19] Jake   ->Ian     Henry  ->Jake    Ian    ->Jake
## [22] Kyle   ->Jake    David  ->Kyle    Jake   ->Kyle
```

```
plot(sample_edges)
```


Bob
Emilia
Gabby
Fiona
David
Kyle
Jake
Ian
Henry
Charlie
Ann

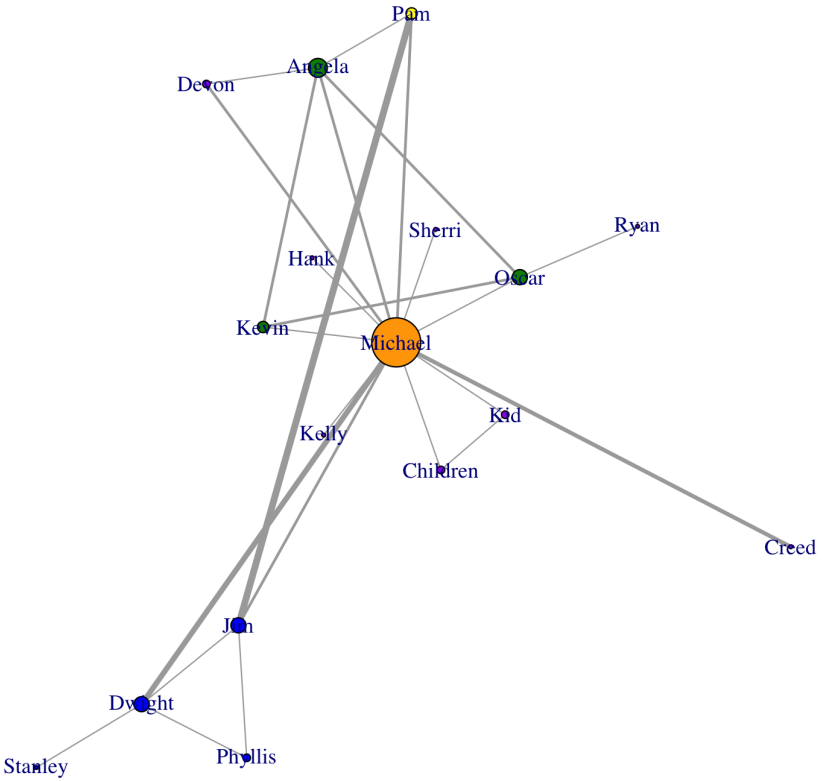
With The Office

```
# Get data from node and edge list
Halloween <- graph_from_data_frame(
  Season_edge_Weight, directed = FALSE, vertices=Season_nodes)
# Get weights
E(Halloween)$weight <- Season_edge_Weight$weight
# Get Colors
V(Halloween)$color <- Season_nodes$color
# Get network statistics
# Degrees
degree <- degree(Halloween, mode="all")
# Clustering coefficient
cluster <- transitivity(Halloween)
# Path Length
path_length <- distances(Halloween)
# Average Path Length
mean_path = mean_distance(Halloween)
# Betweeness
betweeness <- betweenness(Halloween)
# Density
density <- edge_density(Halloween, loops=TRUE)
```

Plot the Network!

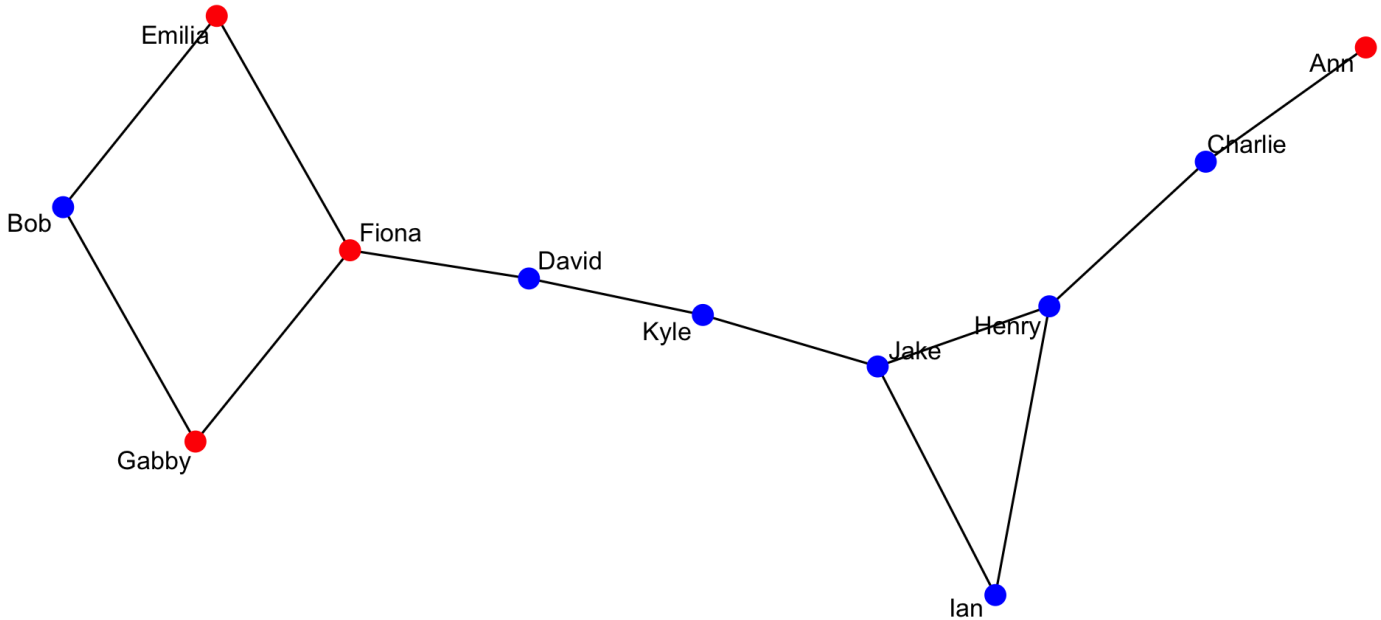
```
plot(
  Halloween,
  edge.width = E(Halloween)$weight,
  layout=layout_with_kk,
  vertex.size=degree,
  vertex.color=V(Halloween)$color)
title("Co-Occurances: Halloween")
```

Co-Occurances: Halloween



Networks with ggraph

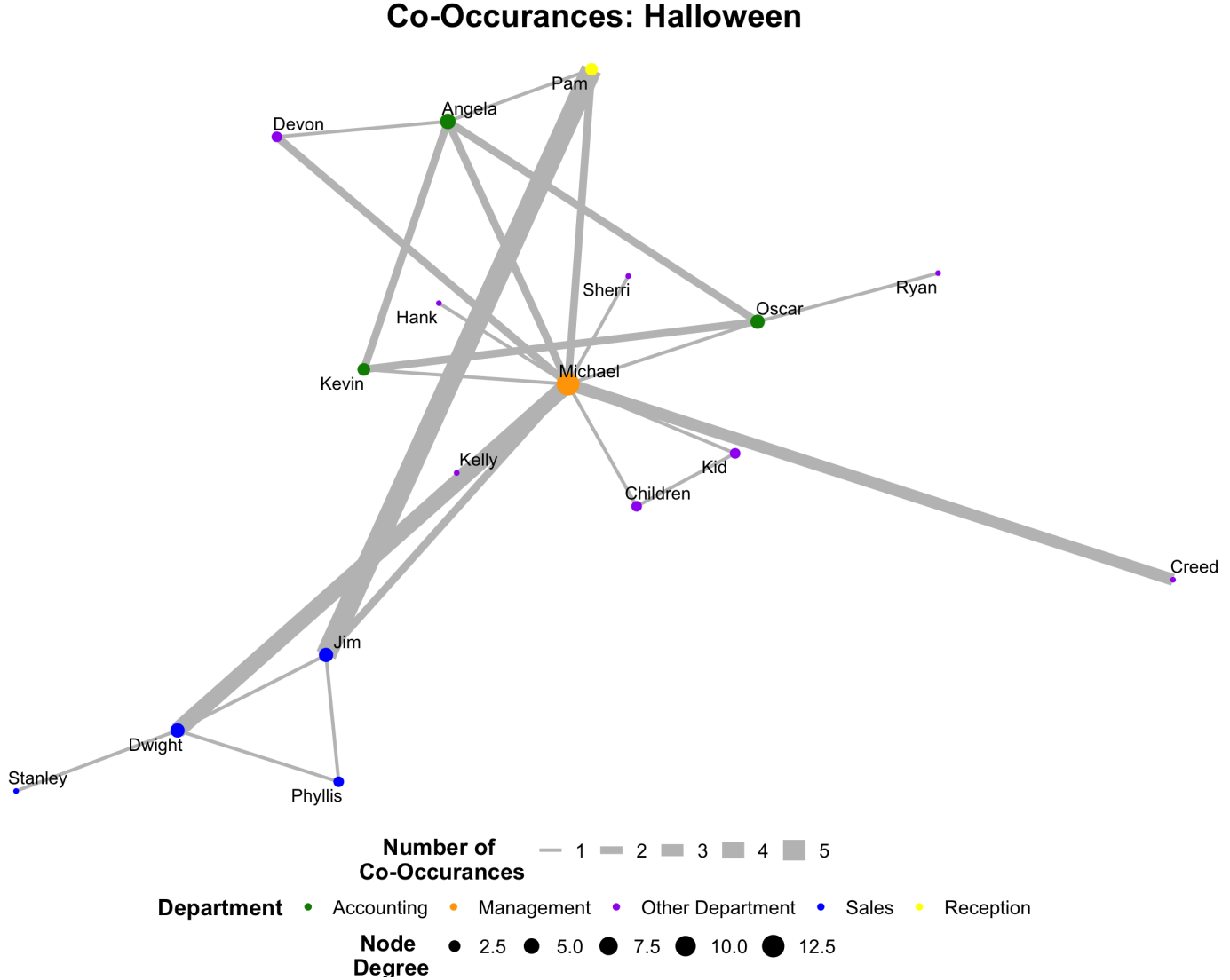
```
g <- tbl_graph(edges = edgelist, directed=FALSE, nodes=nodelist)
ggraph(g) +
  geom_edge_link()+
  geom_node_point(aes(color = gender), size = 4)+
  geom_node_text(aes(label = person), size = 4, repel = TRUE) +
  scale_color_manual(
    breaks = c("Male", "Female"),
    values = c(
      "Male" = "blue",
      "Female" = "red"
    )
  )+
  theme_void()+
  theme(legend.position = 'none')
```


Emilia
Ann
Charlie
Bob
Fiona
David
Kyle
Henry
Jake
Gabby
Ian

With The Office

```
# Generate network object from prepped data
Halloween_network <- tbl_graph(edges = Season_edge_Weight, nodes =
  # Calculate network statistics
  mutate(
    degree = centrality_degree(),
    betweeness = centrality_betweenness(),
    path = node_distance_from(),
    mean_path = graph_mean_dist(),
    triangles = local_triangles(),
    cluster = 2*triangles/(degree*(degree-1))
  )
```

```
ggraph(Halloween_network, layout = 'kk') +
  geom_edge_link(aes(width = weight), color = "gray75")+
  geom_node_point(aes(size = degree, color = department))+
  geom_node_text(aes(label = person), repel = TRUE) +
  scale_edge_width(name = "Number of \nCo-Occurances")+
  scale_size_continuous(name = "Node \nDegree")+
    scale_color_manual(
    name = "Department",
    breaks = c("Accounting", "Management", "Other Department", "Sa
    values = c(
      "Accounting" = "green4",
      "Management" = "orange",
      "Other Department" = "purple",
      "Sales" = "blue",
      "Reception" = "yellow" )
  )+
  labs(
    title = "Co-Occurances: Halloween"
  )+
  theme_void()+
  theme(
    plot.title          = element_text(
      hjust = 0.5, size = 20, colour="black", face = "bold"),
    plot.subtitle       = element_text(
      hjust = 0.5, size = 16, colour="black", face = "bold"),
    legend.title        = element_text(
      hjust = 0.5, size = 14, colour="black", face = "bold"),
```


Co-Occurances: Halloween
Pam
Angela
Devon
Sherri
Ryan
Oscar
Hank
Kevin
Michael
Kelly
Kid
Children
Creed
Jim
Dwight
Stanley
Phyllis
Number of Co-Occurances
1
2
3
4
5
Department
Accounting
Management
Other Department
Sales
Reception
Node Degree
2.5
5.0
7.5
10.0
12.5

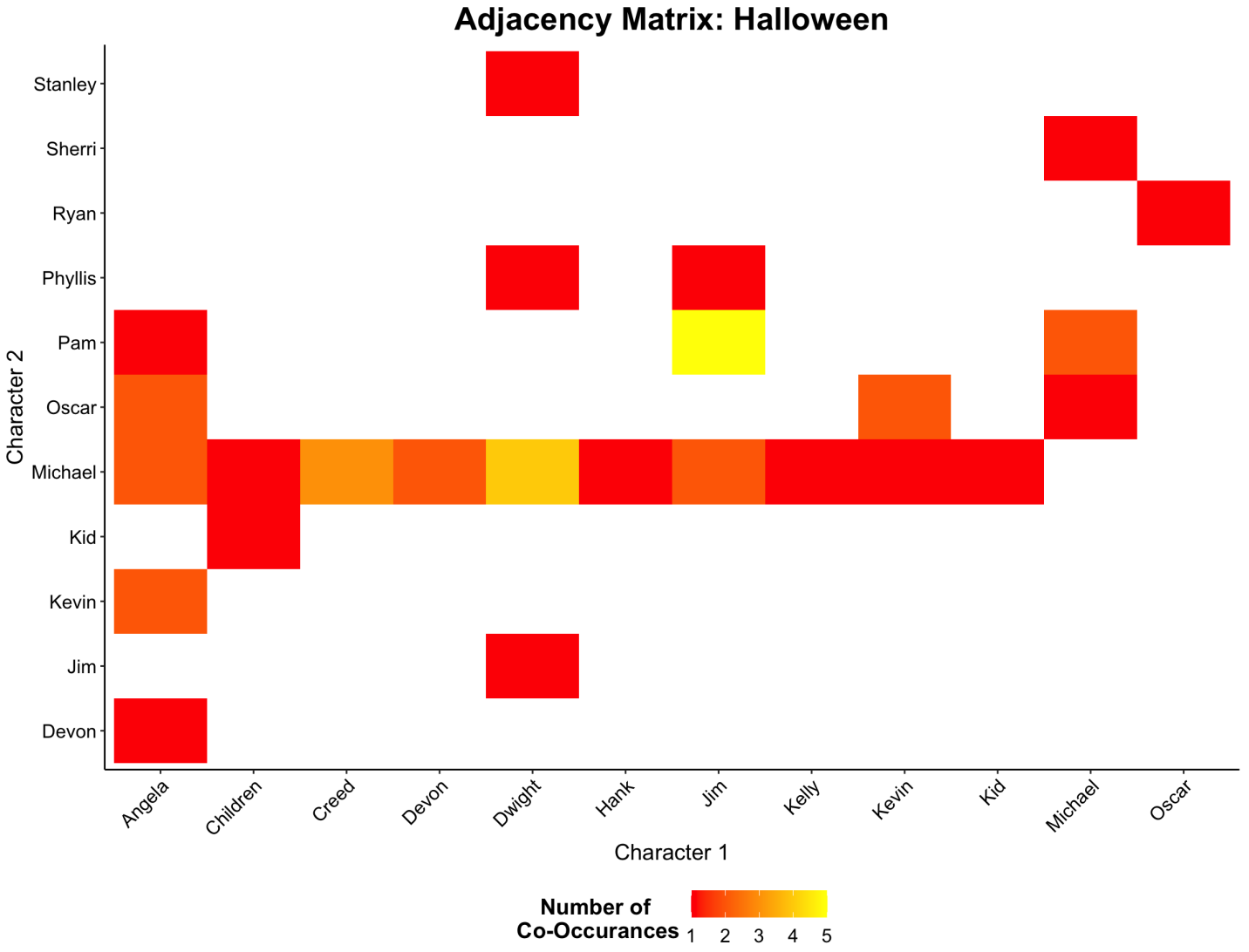
Adjacency Matrices

Demonstrate co-occurrences like a network but plots like a correlation matrix.

X and Y are both the nodes, the place where the lines cross is shaded in if there is a co-appearance

We can draw these matrices using a simple `ggplot`

```
ggplot(Season_edge_Weight, aes(x = from, y = to, fill = weight)) +
  geom_raster()+
  labs(
    title = "Adjacency Matrix: Halloween",
    fill = "Number of \nCo-Occurances"
  )+
  xlab("Character 1")+
  ylab("Character 2")+
  scale_fill_gradient(low = "red", high = "yellow", na.value = NA)
  theme_classic()+
  theme(
    plot.title          = element_text(
      hjust = 0.5, size = 20, colour="black", face = "bold"),
    plot.subtitle       = element_text(
      hjust = 0.5, size = 16, colour="black", face = "bold"),
    legend.title        = element_text(
      hjust = 0.5, size = 14, colour="black", face = "bold"),
    plot.caption        = element_text(size = 10, colour="black"),
    axis.title          = element_text(size = 14, colour="black"),
    axis.text.x         = element_text(
      size = 12, colour="black", angle = 45, hjust = 1),
    axis.text.y         = element_text(size = 12, colour="black"),
    legend.position     = 'bottom',
    legend.direction    = "horizontal",
    legend.text         = element_text(size = 12, colour="black")
  )
```


Adjacency Matrix: Halloween
Stanley
Sherri
Ryan
Phyllis
Pam
Oscar
Michael
Kid
Kevin
Jim
Devon
Character 2
Angela
Children
Creed
Devon
Dwight
Hank
Jim
Kelly
Kevin
Kid
Michael
Oscar
Character 1
Number of Co-Occurances
1
2
3
4
5

A Note on Aesthetics

CHAPTER 11

VISUALIZATION OF POLITICAL NETWORKS

JÜRGEN PFEFFER

Data -- Flights Around the World

The data we will use for this demonstration is on airlines and their flight routes.

Each row is an airline and their flight route from one airport to another in the world.

```
# Flights Data
url <- "https://raw.githubusercontent.com/jpatokal/openflights/mas
data_file <- "data/routes.dat"
flights <- read.csv(paste0(url, data_file), header = FALSE)

names <- c("Airline", "AirlineID", "From", "FromID",
           "To", "ToID", "Codeshare", "stops", "Equipment")
colnames(flights) <- names
```

Looking at Major US Airports

There are many airports represented in this dataset. I am just going to zoom in on a handful of major US airports.

Many of these are major cities or reflect hubs for the major US carriers of American Airlines, United Airlines and Delta Airlines.

```
Airports <- c(
  "EWR", "IAD", "ORD", "IAH", "DEN", "LAX", "SFO",
  "FLL", "MIA", "CLT", "MSP", "CMH", "ATL", "DFW",
  "JFK", "LGA", "BOS", "DCA", "IAD", "DTW", "SLC",
  "MCO", "LAS", "SEA", "PHL", "BWI", "TPA", "BNA"
  )
```

Getting a Network

```
short_list <- flights %>%
  filter(From %in% Airports) %>%
  filter(To %in% Airports) %>%
  mutate(
    pair_ID = paste0(From, " " , To)
  ) %>%
  group_by(pair_ID) %>%
  summarise(
    n = n(),
    .groups = 'keep'
    ) %>%
  tidyr::separate(
    pair_ID,
    into = c("from", "to"),
    sep = "\\s",
    remove = TRUE
  )
```

Getting a Network

```
network <- as_tbl_graph(short_list) %>%
  mutate(
    degree = centrality_degree(),
    betweeness = centrality_betweenness(),
    path = node_distance_from(),
    mean_path = graph_mean_dist(),
    triangles = local_triangles(),
    cluster = 2*triangles/(degree*(degree-1))
  )
```

Plotting a Network

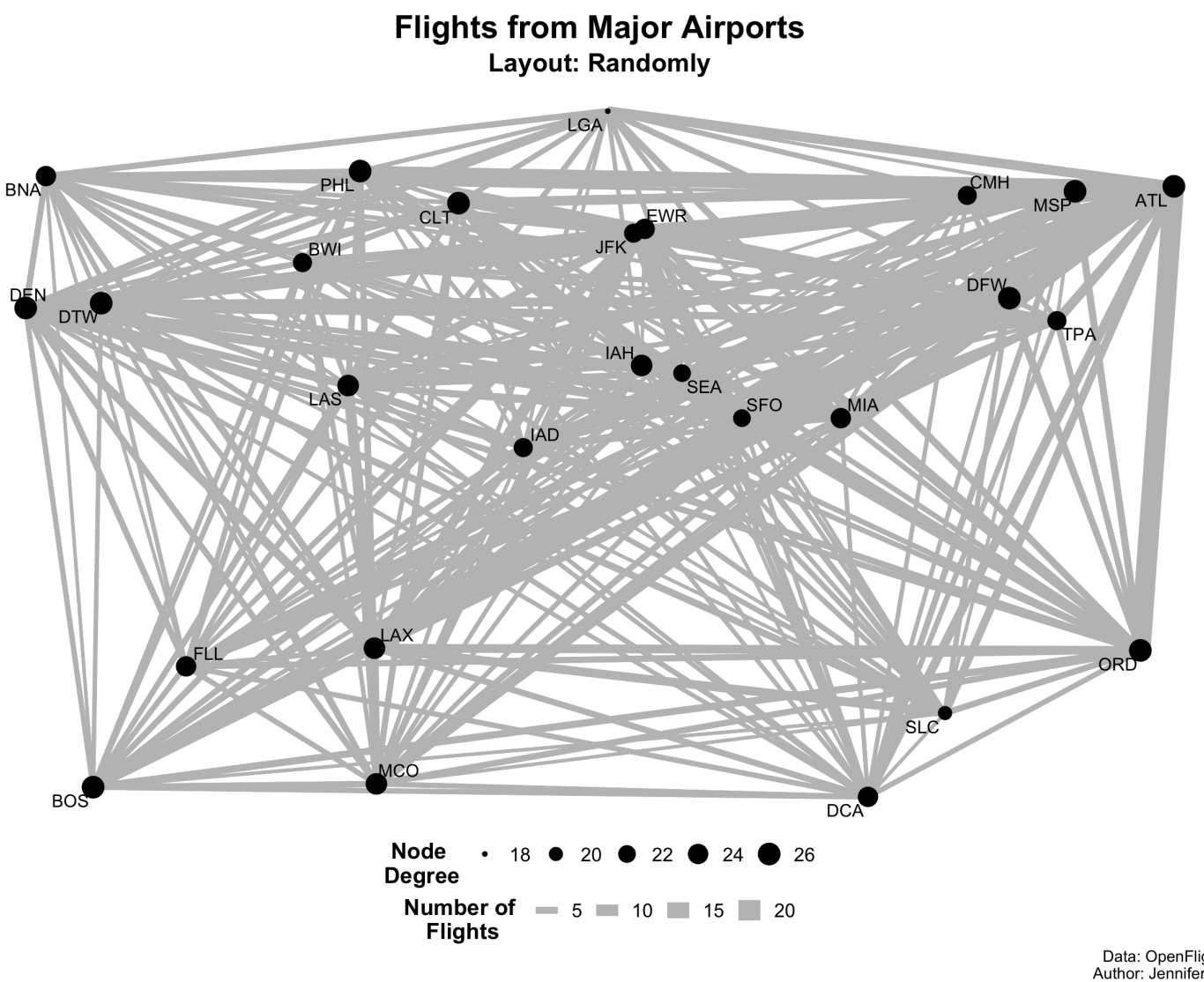
The Theme Function

```
theme_gnetwork <- function(){
  theme_void()+
    theme(
      plot.title          = element_text(
        hjust = 0.5, size = 20, colour="black", face = "bold"),
      plot.subtitle       = element_text(
        hjust = 0.5, size = 16, colour="black", face = "bold"),
      legend.title        = element_text(
        hjust = 0.5, size = 14, colour="black", face = "bold"),
      plot.caption        = element_text(size = 10, colour="black"),
      legend.position     = 'bottom',
      legend.direction    = "horizontal",
      legend.box          = "vertical",
      legend.text         = element_text(size = 12, colour="black")
    )
}
```

Plotting a Network

The Code

```
ggraph(network, layout = 'igraph', algorithm = "randomly")+
  geom_edge_link(aes(width = n), color = "gray75")+
  geom_node_point(aes(size = degree), colour = "black")+
  geom_node_text(aes(label = name), repel = TRUE) +
  scale_edge_width(name = "Number of \nFlights")+
  scale_size_continuous(name = "Node \nDegree")+
  labs(
    title = "Flights from Major Airports",
    subtitle = "Layout: Randomly",
    caption = "Data: OpenFlights
    Author: Jennifer Lin"
  )+
  theme_gnetwork()
```

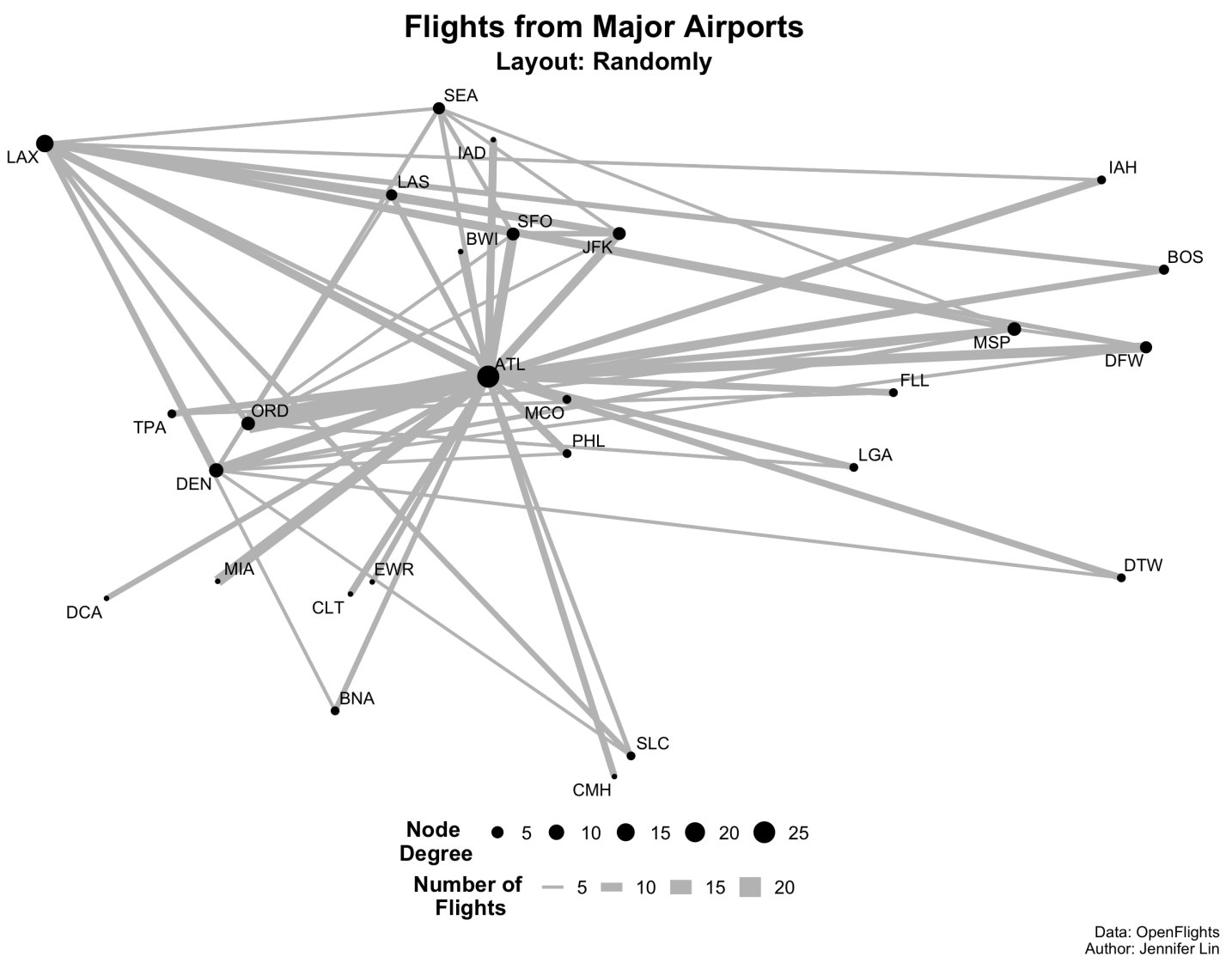

Flights from Major Airports
Layout: Randomly
LGA
BNA
PHL
CLT
EWR
JFK
CMH
MSP
ATL
BWI
DEN
DTW
DFW
TPA
IAH
SEA
LAS
SFO
MIA
IAD
LAX
FLL
ORD
SLC
MCO
BOS
DCA
Node Degree
18
20
22
24
26
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin

Global Threshold

That was a lot.

Pfeffer (2017) recommends looking at the Global Thresholds -- which takes the top number of connections to account in plotting the networks, which can effectively get rid of a lot of this noise.

```
network <- short_list %>%
  filter(n >= 5) %>%
  as_tbl_graph() %>%
  mutate(
    degree = centrality_degree(),
    betweeness = centrality_betweenness(),
    path = node_distance_from(),
    mean_path = graph_mean_dist(),
    triangles = local_triangles(),
    cluster = 2*triangles/(degree*(degree-1))
  )
```

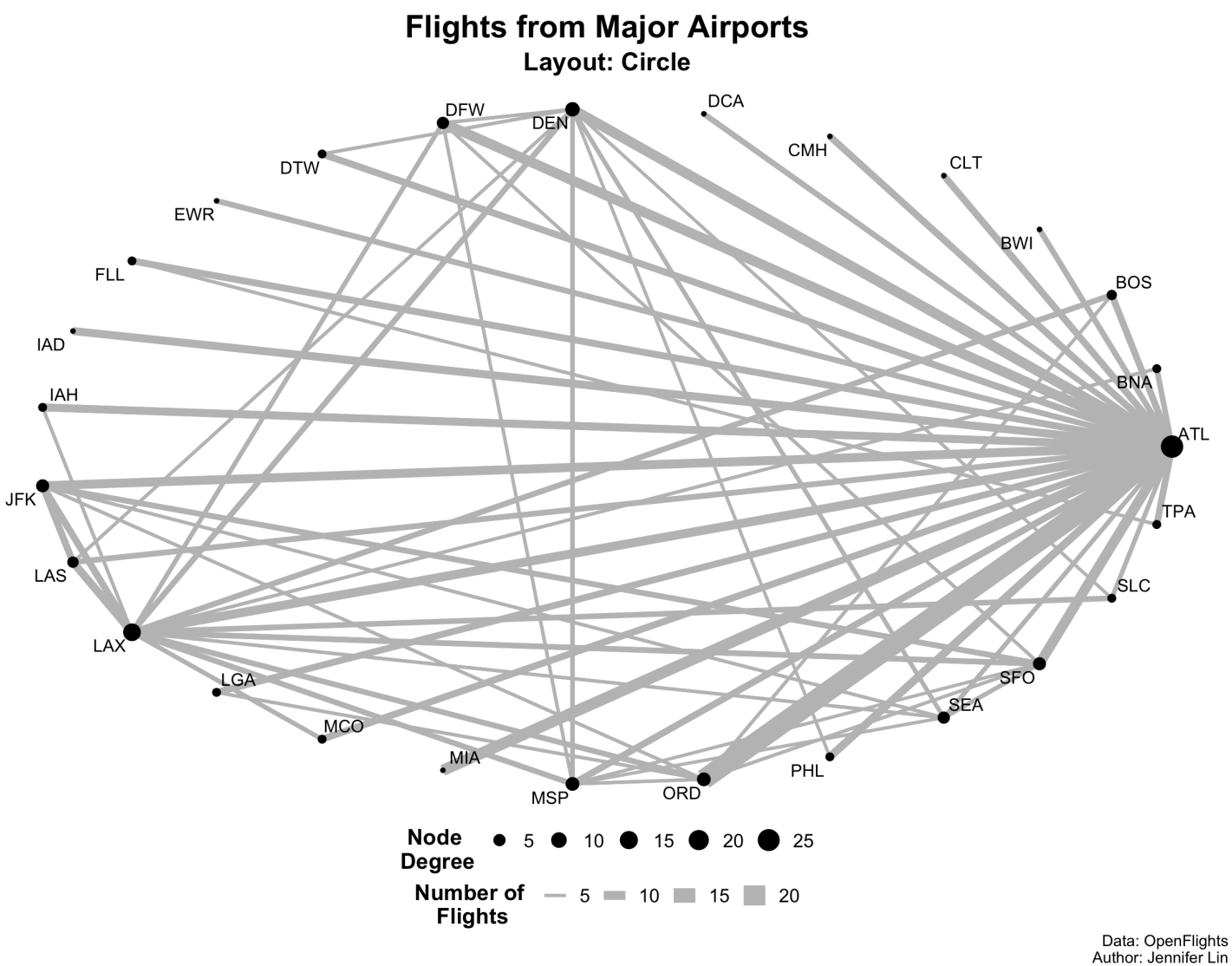

Flights from Major Airports
Layout: Randomly
SEA
LAX
IAD
IAH
LAS
SFO
BWI
JFK
BOS
MSP
DFW
ATL
FLL
ORD
TPA
MCO
PHL
LGA
DEN
DTW
MIA
EWR
DCA
CLT
BNA
SLC
CMH
Node Degree
5
10
15
20
25
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin

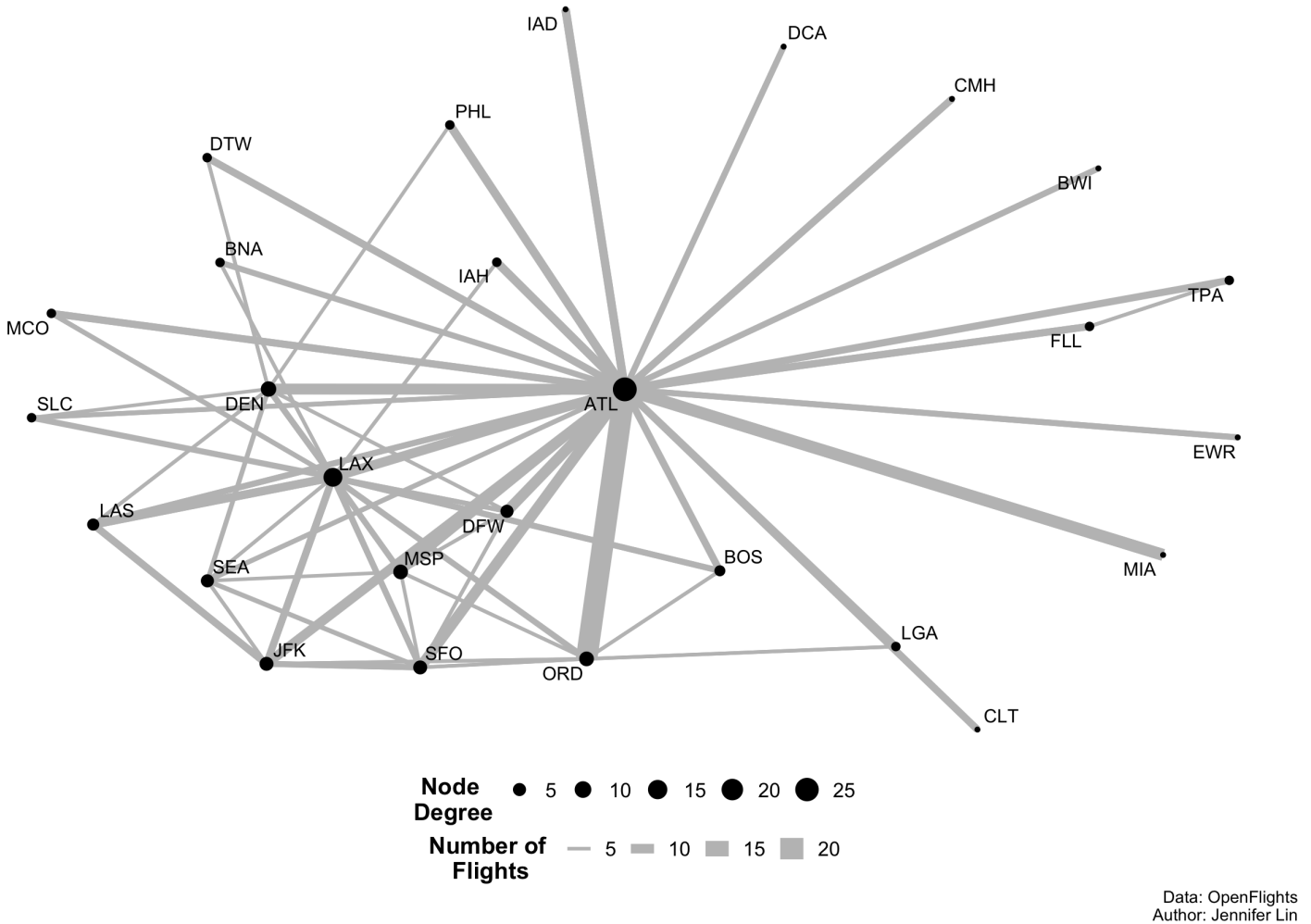
Try Different Layouts

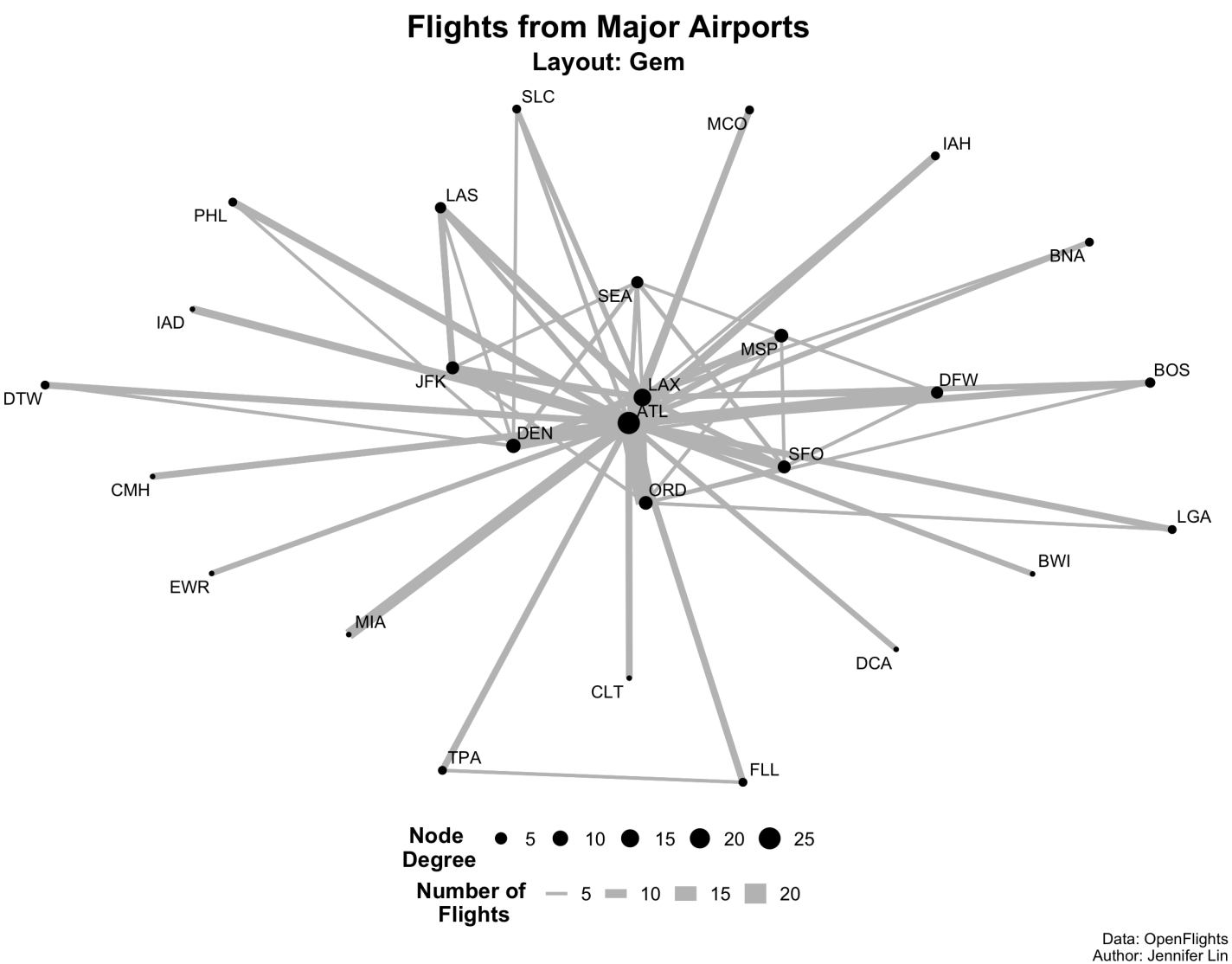
In the past, people have come up with many different ways of rearranging networks on a diagram to avoid things looking like a mess.

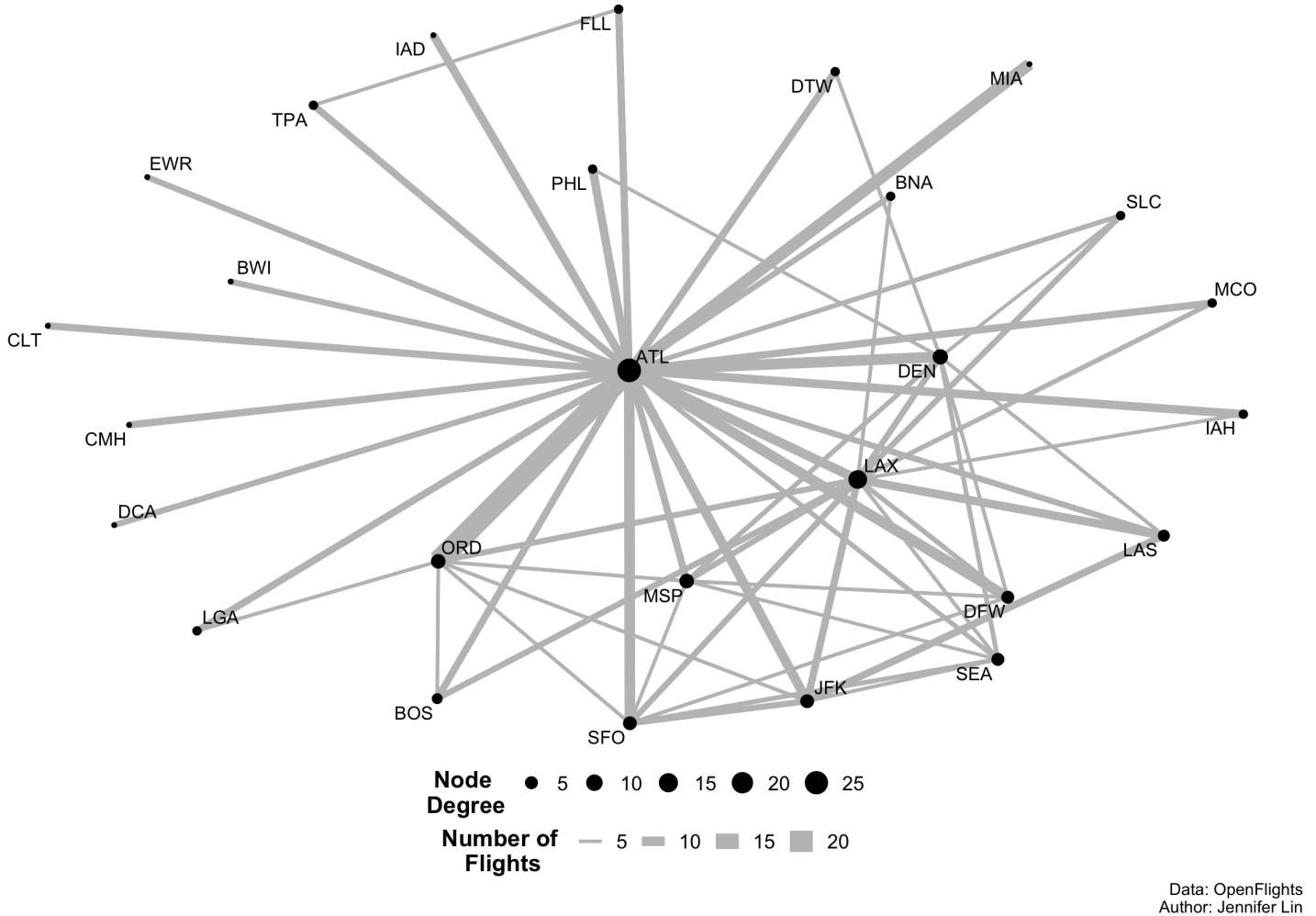
In the following slides, we will use the same Global Threshold configuration to show the data under different layouts.

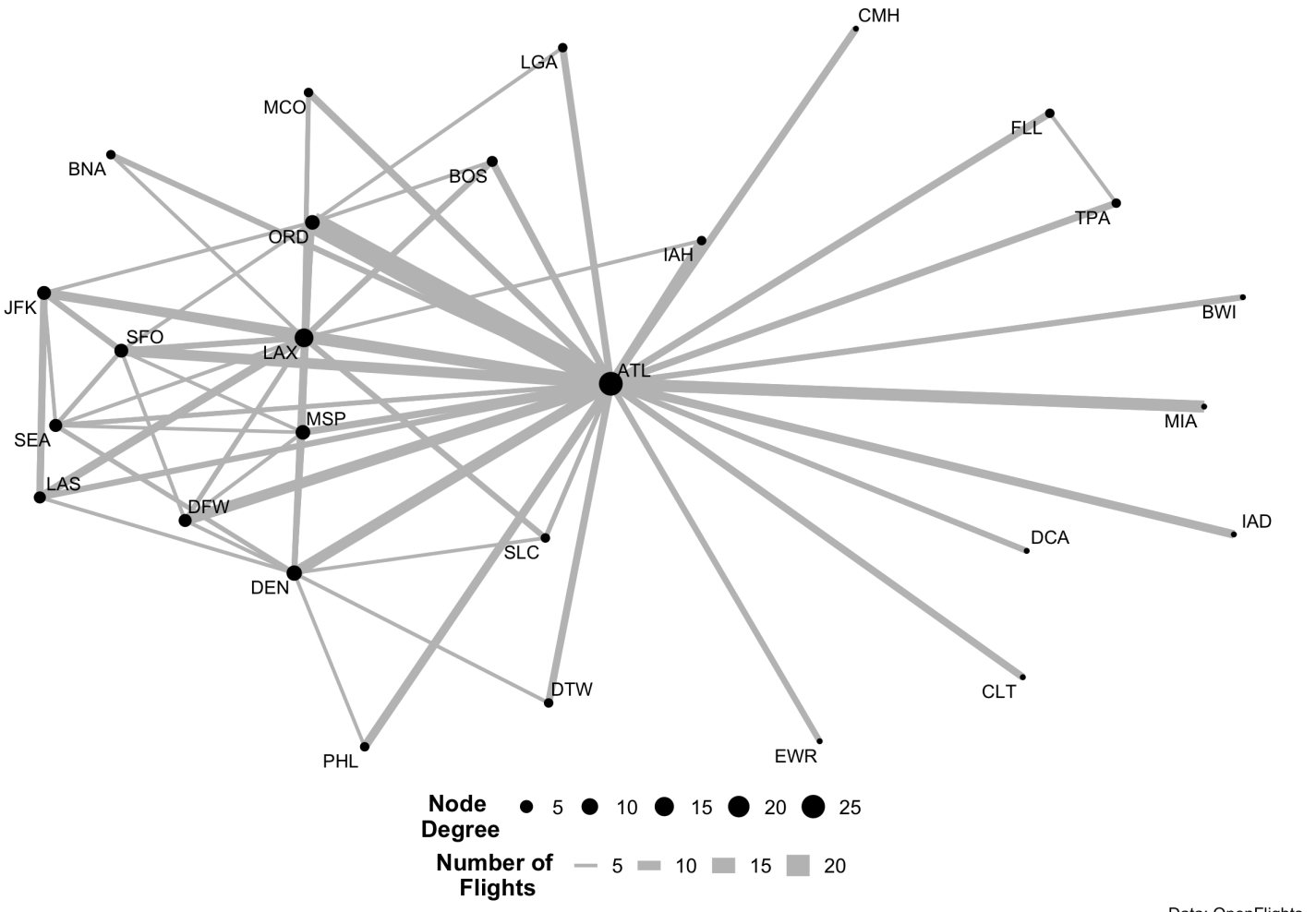
The underlying code for all of these plots is the same. The only thing that has changed is the `igraph` algorithm. See handout for more details.


Flights from Major Airports
Layout: Circle
DFW
DEN
DCA
CMH
CLT
DTW
EWR
BWI
FLL
BOS
IAD
BNA
IAH
ATL
JFK
TPA
LAS
SLC
LAX
SFO
LGA
SEA
MCO
MIA
PHL
MSP
ORD
Node Degree
5
10
15
20
25
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin


Flights from Major Airports
Layout: Fr
IAD
DCA
CMH
PHL
DTW
BWI
BNA
IAH
TPA
MCO
FLL
SLC
DEN
ATL
EWR
LAX
LAS
DFW
MSP
BOS
SEA
MIA
LGA
JFK
SFO
ORD
CLT
Node Degree
5
10
15
20
25
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin

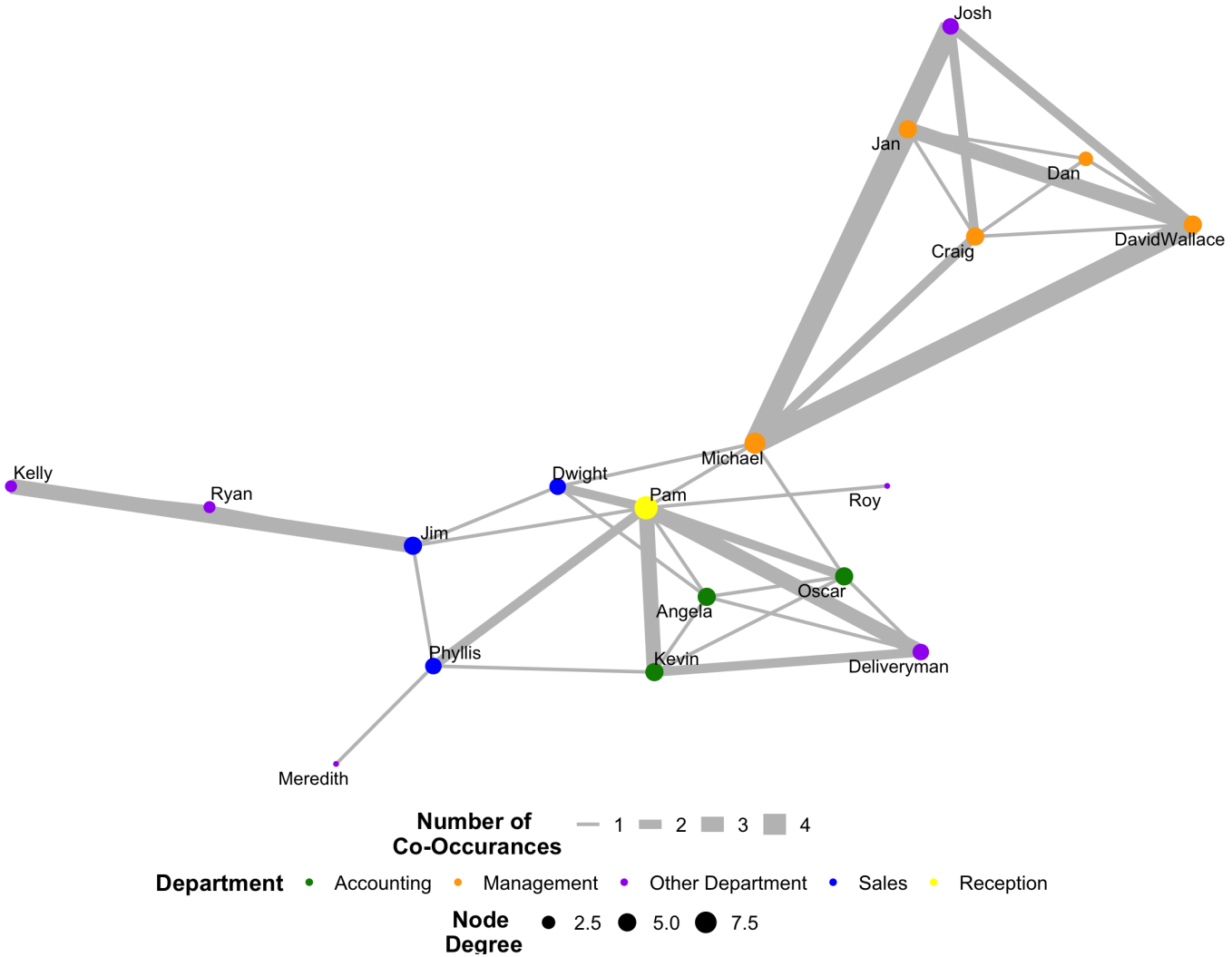

Flights from Major Airports
Layout: Gem
SLC
MCO
IAH
LAS
PHL
BNA
SEA
IAD
MSP
JFK
DFW
BOS
DTW
LAX
ATL
DEN
SFO
CMH
ORD
LGA
BWI
EWR
MIA
DCA
CLT
TPA
FLL
Node Degree
5
10
15
20
25
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin

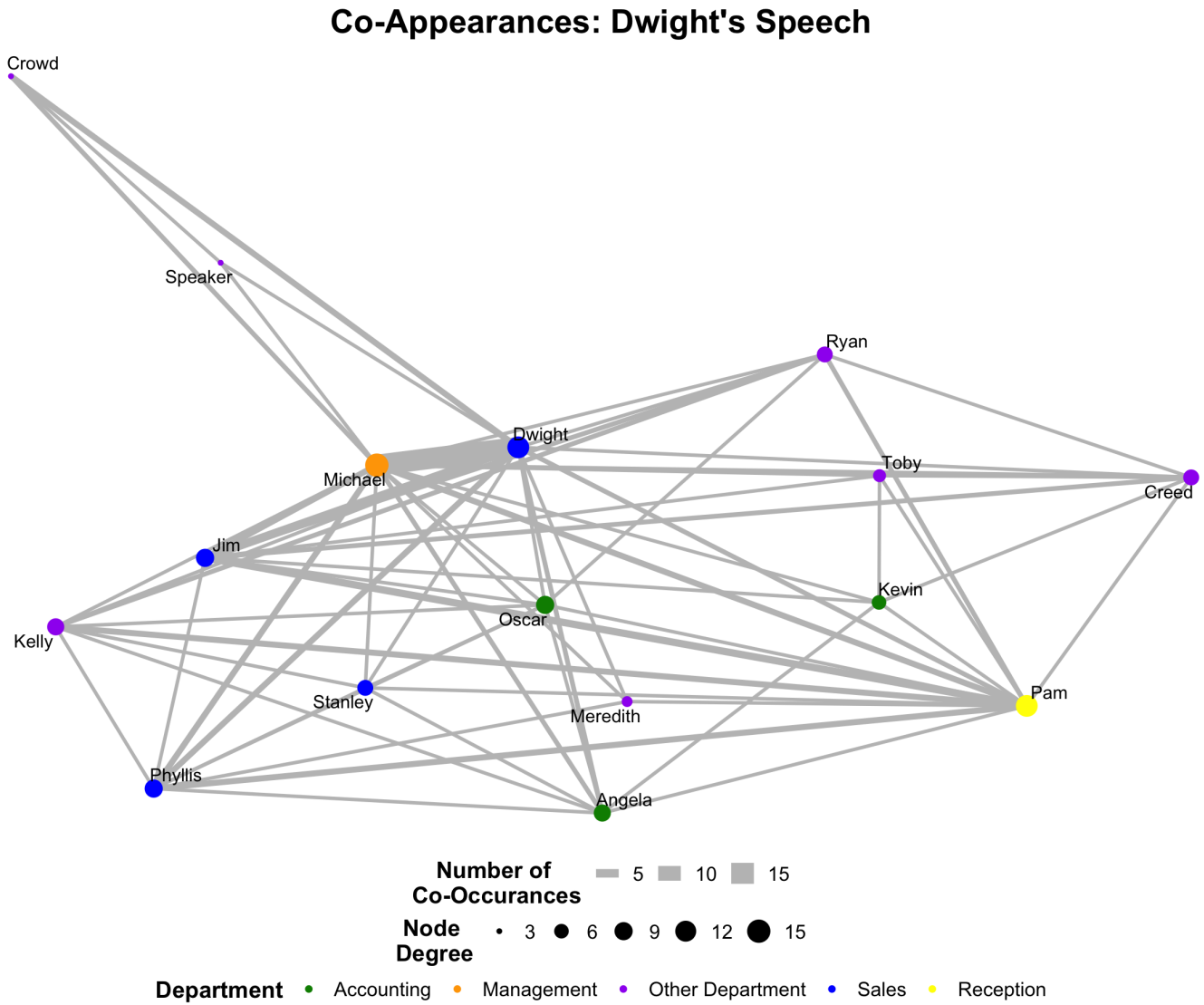

Flights from Major Airports
Layout: Kk
FLL
IAD
DTW
MIA
TPA
EWR
PHL
BNA
SLC
BWI
MCO
CLT
ATL
DEN
IAH
CMH
LAX
DCA
ORD
LAS
MSP
DFW
LGA
SEA
JFK
BOS
SFO
Node Degree
5
10
15
20
25
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin


Flights from Major Airports
Layout: Nicely
CMH
LGA
MCO
FLL
BNA
BOS
TPA
ORD
IAH
JFK
BWI
SFO
LAX
ATL
MSP
MIA
SEA
LAS
DFW
IAD
DCA
SLC
DEN
DTW
CLT
PHL
EWR
Node Degree
5
10
15
20
25
Number of Flights
5
10
15
20
Data: OpenFlights
Author: Jennifer Lin

Exercises

1. Read in the Edge Weight files for either *Valentine's Day* or *Dwight's Speech*
2. Reproduce the following networks (or generate something to a similar effect) for either episode
3. Revisit the Episode Descriptions that I include in the handout. Do your results make sense given the context of the episode? Discuss. Why or Why not?


Co-Appearances: Valentine's Day
Josh
Jan
Dan
Craig
DavidWallace
Michael
Kelly
Ryan
Dwight
Pam
Roy
Jim
Oscar
Angela
Phyllis
Kevin
Deliveryman
Meredith
Number of Co-Occurances
1
2
3
4
Department
Accounting
Management
Other Department
Sales
Reception
Node Degree
2.5
5.0
7.5


Co-Appearances: Dwight's Speech
Crowd
Speaker
Ryan
Dwight
Michael
Toby
Creed
Jim
Kevin
Oscar
Kelly
Stanley
Meredith
Pam
Phyllis
Angela
Number of Co-Occurances
5
10
15
Node Degree
3
6
9
12
15
Department
Accounting
Management
Other Department
Sales
Reception