

Chapter 7 Networks and Inequality

In this section, we will look at a network analysis. Networks emerged from a combination of sociological theory, linear algebra, statistics, and graph theory. They present a totally different way of thinking about data than most of previous topics. Here, we'll cover different types of networks and methods for viewing them. Substantively, we will consider inequality in networks as this relates to disasters and changes in the environment.

7.1 What Are Networks?

Networks have two elements: nodes and edges. *Nodes*, or vertices, are points, and *edges*, or links, are the connections between the nodes. We can illustrate these with a simple network graph:

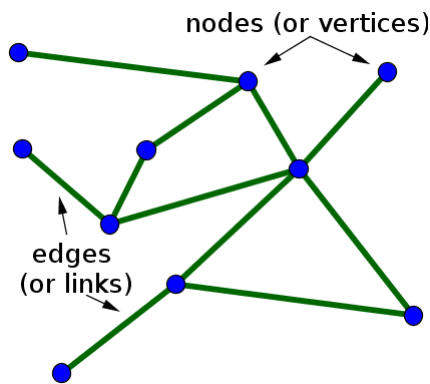


Figure 7.1: Network example. From Nykamp DQ, “An introduction to networks”.

Importantly, a network's edges can be directed or undirected. *Directed* edges constitute flows from one node to another; the order matters. These could be cash payments, academic citations, school transfers, social media follows, or anything else where the direction of the edge would matter. A directed network graph could look like the figure below. Notice how each edge is unidirectional, and that a pair of nodes can have two edges between them.

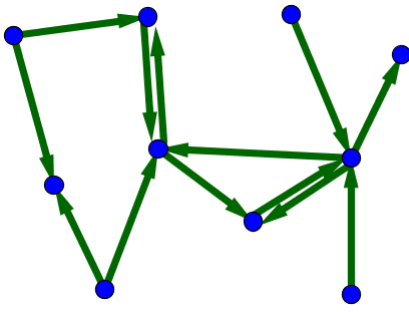


Figure 7.2: Directed Network. From Nykamp DQ, “An introduction to networks”.

Undirected edges constitute bidirectional ties, in which the order does not matter. These could be friendships or romantic partnerships, classes that students have taken together, bills that congresspeople have written together, companies that share a board member, or anything else where the connection is the same for both parties. A possible undirected network is shown here:

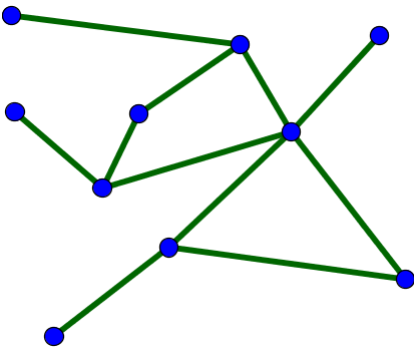


Figure 7.3: Undirected Network. From Nykamp DQ, “An introduction to networks”.

For both directed and undirected networks, we may want to modify the edges according to characteristics of the ties (for example, are these strong or weak connections?), and we may want to modify the nodes according to their characteristics as well (for example, what group is each member from?). We can do this by changing the color, shape, line width, and so on of our network, as we might do for other types of visualization.

7.2 A Broader View of Climate Change, Networks, and Inequality

Networks are relevant to the changing climate in several ways. Here, I'll focus on a few major ways in which network analysis can contribute to our understand of, and our mitigation of the climate crisis.

First, networks matter in disasters. In the book [Heat Wave](#), Eric Klinenberg notes the protective role of social networks in the 1995 Chicago Heat Wave. In [an article that we read](#), this is how Klinenberg describes a poor neighborhood in Chicago with strong social networks:

“Older people there belonged to block clubs; residents assured me they knew who they had to keep tabs on during the heat wave.”

By checking in on each other during the heat wave, the residents ensured that they were taking precautionary measures and that they would notice early warning signs. Klinenberg contrasts this to another neighborhood in which residents are far more isolated, and therefore tended to face the heat wave on their own. In the latter case, social isolation meant that residents had no one checking in on them, contributing to much higher mortality rates.

Second, social networks can explain a link between disasters and social inequalities. People affected by disasters like hurricanes often stay with family or friends in the aftermath. Families with higher socioeconomic status are more likely to be able to stay with other affluent families in regions farther from the disaster, contributing to more stability in their recovery trajectories.

[Fothergill and Peek](#) detail this in their book, *Children of Katrina*.

Finally, networks of information are vital to understanding how people think about climate change. We’ve discussed the role of [Twitter bots](#) in spreading climate misinformation. But networks of climate uncertainty go far beyond Twitter bots: [this study](#) looks at contrarian information networks across board executives, politicians, and think tanks in the U.S. It finds that “successful production and diffusion of contrarian information has a particular network structure and corporate influence.” That is, there are particular actors and elite funders that are central to disseminating contrarian information about climate change in the U.S. [This article](#) further expands on networks of the “climate change countermovement,” and its ties to oil and coal companies.

In sum, networks help to explain some of the most pressing questions relating to climate change and climate disasters: why do outcomes differ across places, and across social groups? And why does uncertainty about climate change persist, despite vast scientific evidence? In the following sections, we’ll learn how to build our own networks, and how to apply these to authorship of climate change reports and migration flows between counties.

7.3 Building And Visualizing Networks

Let's first create a network. We'll make a dataframe where every row represents an edge, and the two variables represent from which node the edge goes to (and from). You'll notice that the names can appear more than once - we can think about this as friendship ties, each person could have multiple friends, which would be represented as unique rows. We'll need to put the data in matrix form in order for our later network functions to perform correctly. We'll call this `el`, for "edge list."

```
el <- data.frame(person_1 = c("Lionel", "Frederica", "Garrett", "Lina",
                             "Sampson", "Garrett", "Frederica", "Lionel",
                             "Frederica", "Frederica", "Ahmad"),
                 person_2 = c("Lina", "Charlotte", "Lina", "Sampson",
                             "Charlotte", "Lionel", "Lina", "Sampson",
                             "Sampson", "Garrett", "Garrett")) %>%

  as.matrix()

# take a look
el
```

```
##      person_1  person_2
## [1,] "Lionel"   "Lina"
## [2,] "Frederica" "Charlotte"
## [3,] "Garrett"  "Lina"
## [4,] "Lina"     "Sampson"
## [5,] "Sampson"  "Charlotte"
## [6,] "Garrett"  "Lionel"
## [7,] "Frederica" "Lina"
## [8,] "Lionel"   "Sampson"
## [9,] "Frederica" "Sampson"
## [10,] "Frederica" "Garrett"
## [11,] "Ahmad"    "Garrett"
```

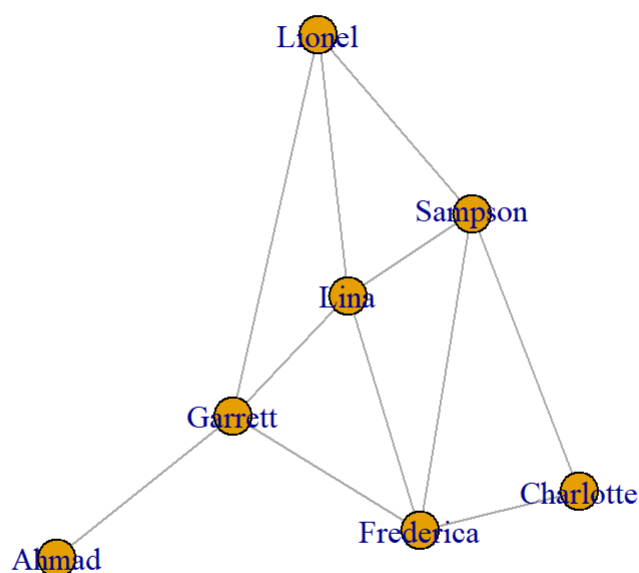
Now we can convert our data to a network with `igraph`. We can specify here whether our network is *directed* or not. This choice is meaningful: if our network simply represents friendship ties, this would best be represented as an *undirected* network (assuming the friendships are bidirectional). However, it could be that our network represents aid given to friends after a disaster (e.g. who let who stay at their home after a hurricane). In the latter case, `person_1` may represent the person who is displaced and `person_2` may represent the person hosting. For now, we will consider our edges to simply represent bidirectional friendships, so our network will be undirected.

```
library(igraph)

# create network from edge List
net <- graph_from_edgelist(el, directed = FALSE)
```

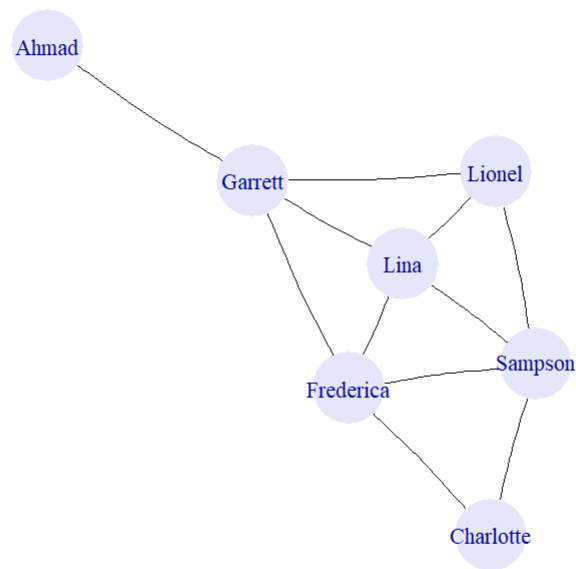
Great! Now let's try to visualize our network.

```
# plot network
plot(net)
```



We can make the graph more visually appealing by changing some of the default settings.

```
# customize network plot
plot(net, vertex.size = 30,
      vertex.color = "lavender",
      vertex.frame.color = NA,
      vertex.label.cex = .7,
      edge.curved = .05,
      edge.arrow.size = .3,
      edge.width = .7,
      edge.color = "gray1")
```

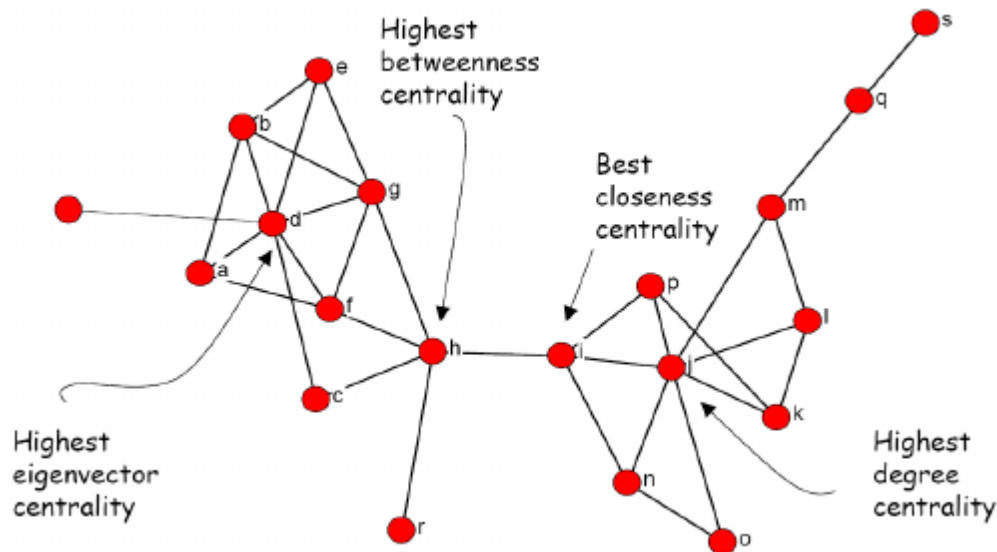


7.4 Measures of Centrality

We may be interested in measuring how central each node is to the entire network. We will focus on the following four measures of network centrality:

- Degree Centrality
- Betweenness Centrality
- Eigenvector Centrality
- Closeness Centrality

These four measures are similar, but measure slightly different concepts. These differences are illustrated here:



Degree Centrality counts how many edges each node has. The node with the most edges, therefore, is most central. This is a straightforward approach to understanding centrality that is easily communicated: in a friendship network, the most “central” actor would simply be the person with the most friends. This would capture the most popular actor, in some sense.

We can calculate degree centrality using the `degree()` function. In our network, Lina, Frederica, Garrett, and Sampson are all tied for the highest degree centrality.

```
degree(net)
```

##	Lionel	Lina	Frederica	Charlotte	Garrett	Sampson	Ahmad
##	3	4	4	2	4	4	1

Betweenness Centrality considers how important nodes are to the flow of information (or other resources) between nodes. Betweenness similarity measures the shortest path between every combination of nodes. It then counts how many “shortest paths” each node is involved in, and the node with the most is considered most central. We might want to use this measure if we are interested in actors that connect different groups: for example, it could be that the most popular politician is also immersed in an echochamber, and we would not want to consider them central to the functioning of the entire network. In this case we could use the betweenness centrality to get a sense of who is a “bridging tie” between the most members of the entire network.

Betweenness centrality can be calculated using the `betweenness()` function. According to this metric, Garrett reigns supreme! This is mostly because Garrett has the sole tie to Ahmad, and thus is involved in every “shortest path” involving Ahmad.

```
betweenness(net)
```

```
##      Lionel      Lina Frederica Charlotte  Garrett  Sampson    Ahmad
## 0.6666667 1.0000000 3.1666667 0.0000000 5.3333333 1.8333333 0.0000000
```

Closeness Centrality similarly makes use of the “shortest paths” between nodes. Closeness centrality is measured by averaging the values of these shortest paths from one node to all others (farness), and then taking the reciprocal of this. We might want to use this if we are interested in not necessarily the flow of information, but in which members may have access to all other members of the network.

Closeness centrality can be calculated using the `closeness()` function. By this metric, Lina, Frederica, and Garrett are all at equally central positions (but not Sampson, as was the case in the degree centrality measure. This is because the shortest path from Sampson to Ahmad involves 3 edges).

```
closeness(net)
```

```
##      Lionel      Lina Frederica Charlotte  Garrett  Sampson    Ahmad
## 0.11111111 0.12500000 0.12500000 0.09090909 0.12500000 0.11111111 0.07692308
```

Eigenvector Centrality considers not only the ties of the node of interest (or ego), but also the ties of nodes that they are connected to (their alters). The Eigenvector Centrality measure rewards alters with lots of connections. Perhaps if we are looking at a network of politicians and lobbyists, we want to consider not just individual’s ties, but whether they are tied to powerful people. An influential member might only have a few ties, but each of these ties would be extremely popular.

We can calculate Eigenvector Centrality using the `evcent()` function. We also need to specify that we want the `$vector` result. According to this metric, Lina is the most central! This is because Lina is close to other popular people, who have lots of social ties themselves.

```
evcent(net)$vector
```

```
##      Lionel      Lina Frederica Charlotte      Garrett      Sampson      Ahmad
## 0.7910018 1.0000000 0.9429201 0.5334790 0.8472419 0.9304545 0.2412682
```

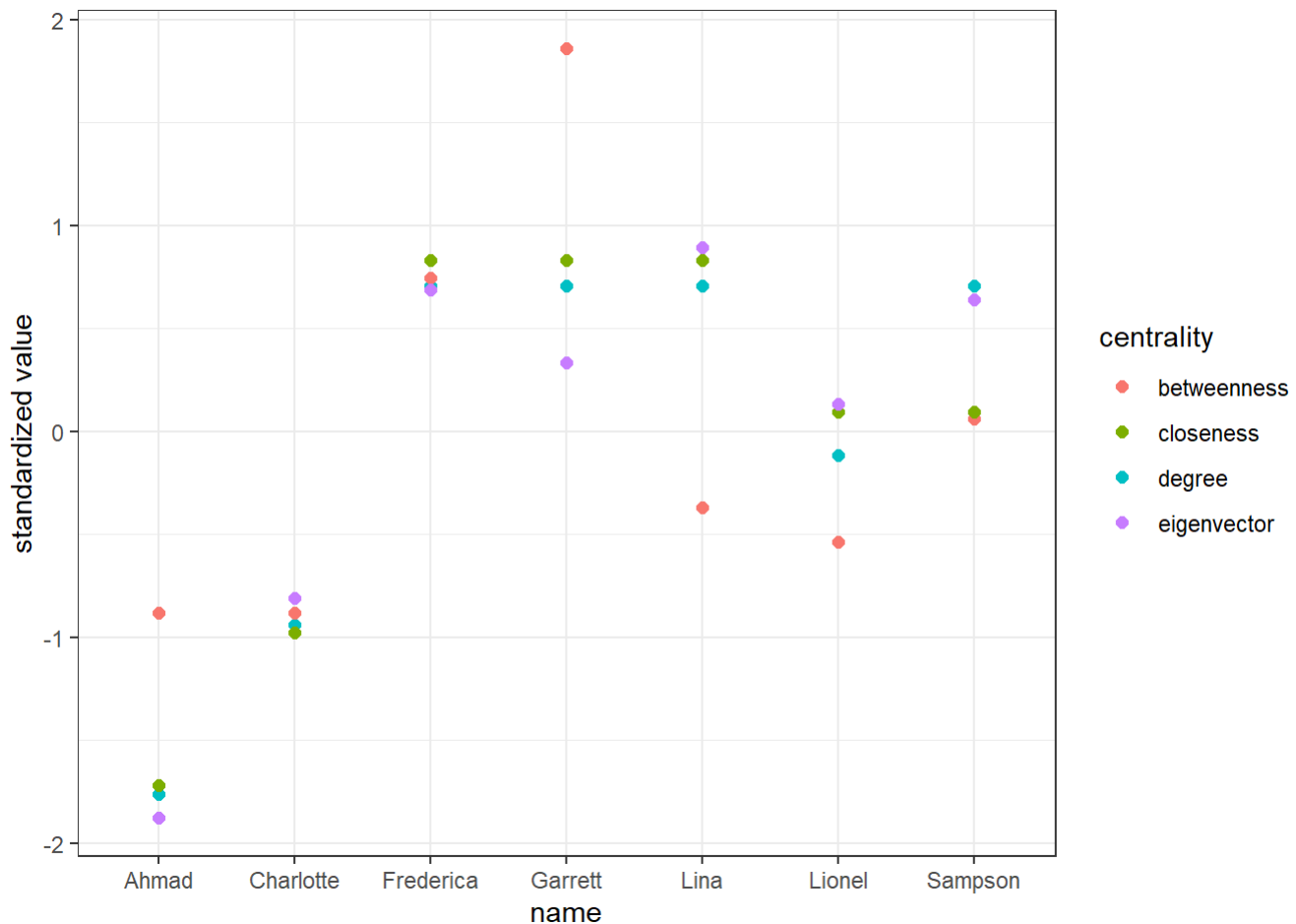
We can compare these four measures of centrality all together with the code below:

```
library(tidyr)
library(ggplot2)
# create dataframe with similarity scores
central_df <- data.frame(degree = degree(net),
                        betweenness = betweenness(net),
                        closeness = closeness(net),
                        eigenvector = evcent(net)$vector)

# all the measures are on different scales, we can standardize them with scale()
central_df %<>%
  mutate_all(scale)

# pivot so that the type of score is a variable and the value is another variable
central_df %<>%
  mutate(name = rownames(.)) %>%
  pivot_longer(-c(name),
              names_to = "centrality",
              values_to = "standardized value")

# now plot the centrality scores
ggplot(central_df, aes(name, `standardized value`))+
  geom_point(aes(col = centrality), size = 2)+
  theme_bw()
```



There are other measures of centrality, such as Bonacich Centrality and Google’s Page Rank, which go beyond the scope of this course. [Hoffman’s “Methods for Network Analysis” bookdown](#) has more information on these.

7.5 Co-Authorship Patterns

We’ll now turn to an example involving authors of the Intergovernmental Panel on Climate Change’s (IPCC) climate reports. The IPCC has written six assessment reports since 1990, each detailing the state of climate change and governmental actions which exacerbate or mitigate its effects. These reports garner much media attention and are influential in global plans to curtail the worst effects of the changing climate. In recent years, scholars have begun to examine geographic biases in representation in authorship of these reports, with most authors coming from the Global North.

Here, we will work with the data from [this paper](#), which analyzes networks of co-authorship in the [fifth assessment report \(AR5\)](#). Specifically, the paper examines authorship on the third working group’s (WGIII) contribution to the report, which details mitigation of climate impacts. The authors

of the study have calculated the number of co-authorships between the researchers on the WGIII (i.e. the number of times that these researchers published scientific papers together, outside of the AR5). I've added this data file to Canvas, but you can also download all the data from the paper in the [Supplementary Information](#).

```
ipcc_authors <- read_csv("Data/IPCC_co_authorship.csv",  
                          skip = 1)
```

We can then view the co-authorship:

Author	Victor	Zhou_Dadi	Ahmed	Olivier	Rogner	Yamaguchi
Abdel-Aziz	0	0	0	0	0	0
Acosta-Moreno	0	0	0	0	0	0
Acquaye	0	0	0	0	0	0
Agrawala	0	0	0	0	0	0
Ahammad	0	0	0	0	0	0
Ahmed	0	0	NA	0	0	0
Akbari	0	0	0	0	0	0
Akimoto	0	0	0	0	0	0
Allwood	0	0	0	0	0	0
Attzs	0	0	0	0	0	0
Babiker	0	0	0	0	0	0
Barreto-Gomez	0	0	0	0	0	0
Barrett	0	0	0	0	0	0
Barua	0	0	0	0	0	0
Bashmakov	0	0	0	0	0	0
Bertoldi	0	0	0	0	0	0
Bigio	0	0	0	0	0	0
Blanco_Gabriel	0	0	0	0	0	0
Blanco_Hilda	0	0	0	0	0	0
Blanford	0	0	0	0	0	0

We notice that this network data has a different structure than the dataframe that we used earlier. This structure is called an *adjacency matrix*. It is also a *sparse* matrix because most of the values are 0.

To create a network from these data, we'll transform it from a dataframe to a matrix object, and use `graph_from_adjacency_matrix()` .

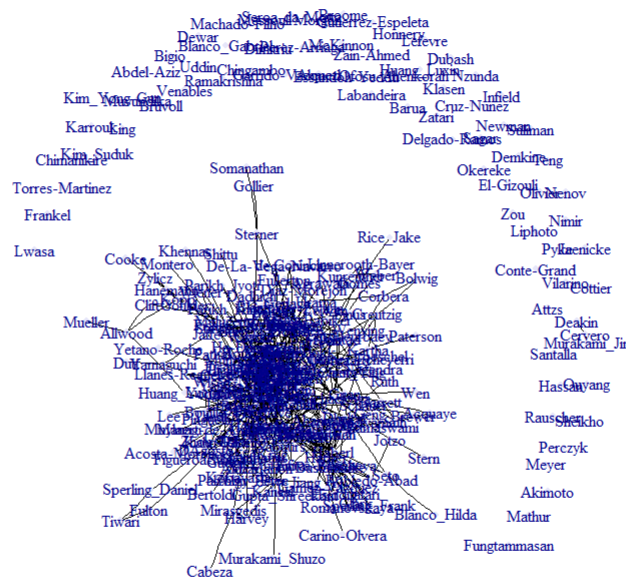
```
# reorder column names to match rows
ipcc_authors %<>%
  select(order(colnames(.)))

# turn dataframe into matrix
ipcc_authors %<>%
  select(-Author) %>%
  as.matrix()

ipcc_net <- graph_from_adjacency_matrix(ipcc_authors,
                                         mode = "undirected")
```

Great! Let's try plotting the network.

```
plot(ipcc_net, vertex.size = 2,
      vertex.color = "lavender",
      vertex.frame.color = NA,
      vertex.label.cex = .5,
      edge.curved = .05,
      edge.width = .1,
      edge.color = "gray1")
```



We notice that there are several authors who have not co-authored with anyone else in the group, and then a large clump of authors who do collaborate with each other (but it is hard to tell exactly how much or who might be central players within this group). There are also many periphery authors who may have collaborated with just one or two others.

To examine the network in further detail, we can increase the threshold for an edge: as is done in the Nature Climate Change paper, we will only consider two or more co-authorships to constitute edges. We can do this by changing all the 1's in our adjacency matrix to 0's. Since `ipcc_authors` is a matrix, we will do this using base R's `index` (rather than `dplyr`'s `mutate` functions, which only work with dataframes).

```
# replace 1s with 0s
ipcc_authors[ipcc_authors == 1] = 0

# replace NAs with 0s
ipcc_authors[is.na(ipcc_authors)] = 0
```

We can also remove all the authors who don't collaborate with anyone else. We'll do this by creating a vector of these indices, and then removing them from both the rows and columns of the matrix (remember that the matrix is symmetric).

```
# vector of authors who do not collaborate with others  
no_colab <- which(apply(ipcc_authors, 1, sum, na.rm = TRUE) == 0)  
  
# we can see the names of these authors by changing the margin to 2  
which(apply(ipcc_authors, 2, sum, na.rm = TRUE) == 0)
```


##	Abdel-Aziz	Acosta-Moreno	Ahammad	Ahmed
##	1	2	5	6
##	Akimoto	Attzs	Barua	Bigio
##	8	10	14	17
##	Blanco_Gabriel	Blanco_Hilda	Bouille	Brewer
##	18	19	23	24
##	Broome	Bruvoll	Cabeza	Chen_Wenying
##	25	28	30	35
##	Chimanikire	Chingambo	Clark	Clift
##	37	38	40	42
##	Conte-Grand	Cooke	Corbera	Cottier
##	43	44	45	46
##	Creutzig	Cruz-Nunez	De-La-Vega-Navarro	de Coninck
##	47	48	51	52
##	Delgado-Ramos	Demkine	Dewar	Dimitriu
##	54	55	57	60
##	Dubash	El-Gizouli	Elsiddig	Essandoh-Yeddu
##	62	66	67	68
##	Fifita	Finus	Fleurbaey	Frankel
##	71	73	76	77
##	Fullerton	Fulton	Fungtammasan	Garrido-Vazquez
##	78	79	80	84
##	Gomez-Echeverri	Gutierrez-Espeleta	Harvey	Hassan
##	89	96	104	105
##	Honnery	Huang_Luxin	Huang_Yongfu	Inaba
##	109	111	112	113
##	Infield	Jaenicke	Jafari	Karrouk
##	114	116	117	126
##	Kartha	Khennas	Kim_Suduk	Kim_Yong-Gun
##	127	128	130	131
##	Klasen	Kolstad	Labandeira	Lee
##	132	134	141	144
##	Lefevre	Ling	Liphoto	Llanes-Regueiro
##	145	147	149	150
##	Loeschel	Lwasa	Machado-Filho	Mathur

##	151	154	155	159
##	McKinnon	Messouli	Meyer	Montero
##	161	163	164	168
##	Morgan	Mulugetta	Mundaca	Murakami_Shuzo
##	169	171	172	174
##	Muvundika	Nenov	Newman	Nimir
##	175	180	181	182
##	Nzunda	Ofosu-Ahenkorah	Okereke	Olivier
##	184	185	186	187
##	Ouyang	Pan	Parikh_Jyoti	Parikh_Kirit
##	188	191	192	193
##	Paterson	Perczyk	Perez-Arriaga	Pyke
##	194	197	198	202
##	Ramakrishna	Rauscher	Robledo-Abad	Romanovskaya
##	203	205	211	213
##	Sagar	Santalla	Seroa-da-Motta	Sheikho
##	217	218	222	224
##	Shittu	Somanathan	Sperling_Daniel	Sperling_Frank
##	226	232	233	234
##	Suliman	Teng	Tiwari	Torres-Martinez
##	241	244	245	246
##	Uddin	Venables	Vilarino	Wen
##	249	254	257	259
##	Wiener	Yetano-Roche	Zain-Ahmed	Zatari
##	261	266	267	268
##	Zhou_Dadi	Zhou_Peter	Zou	Zylicz
##	270	271	272	273

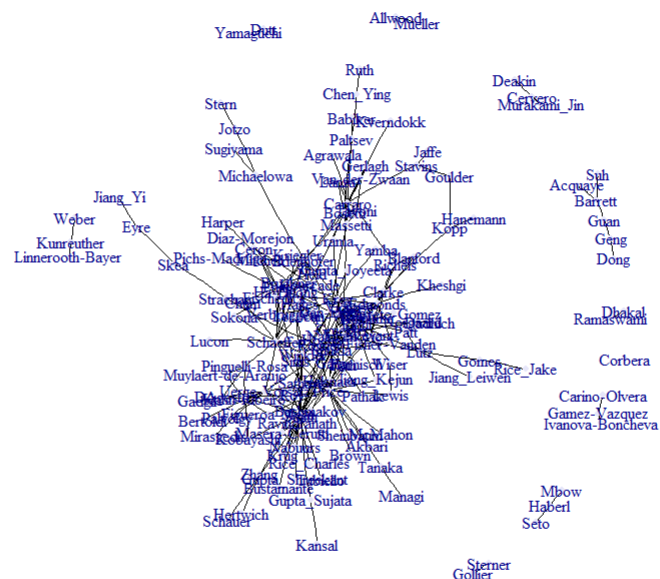
then we can remove these from ipcc_authors

```
ipcc_authors <- ipcc_authors[-no_colab,
                             -no_colab]
```

The dimensions of `ipcc_authors` have now changed from 273 x 273 to 148 x 148.

```
# create network, again
ipcc_net <- graph_from_adjacency_matrix(ipcc_authors,
                                       mode = "undirected")

plot(ipcc_net, vertex.size = 2,
     vertex.color = "lavender",
     vertex.frame.color = NA,
     vertex.label.cex = .5,
     edge.curved = .05,
     edge.width = .1,
     edge.color = "gray1")
```



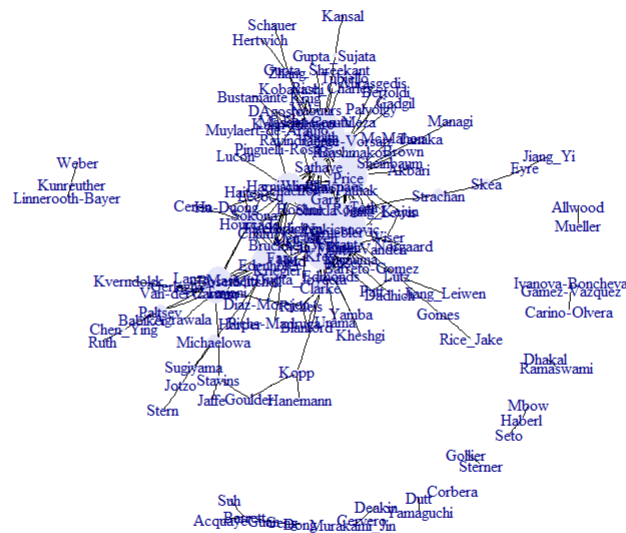
This looks a bit better. We can see that there are peripheral clusters of authorship, as well as a central cluster. However, we still can't really see who the main actors may be. To make this more visible, we'll add a variable to the vertices on our network graph with their betweenness scores.

We'll also standardize the betweenness scores by dividing all of them by the maximum and then multiplying by 20.

```
# add betweenness centrality
```

```
V(ipcc_net)$betweenness <- betweenness(ipcc_net)
```

```
plot(ipcc_net,
      vertex.size = V(ipcc_net)$betweenness/max(V(ipcc_net)$betweenness) * 20,
      vertex.color = "lavender",
      vertex.frame.color = NA,
      vertex.label.cex = .5,
      edge.curved = .05,
      edge.width = .1,
      edge.color = "gray1")
```



We see that there are several “central” authors. Still, it’s hard to see their names here. We can display the authors’ names, in descending order based on their betweenness scores, using the following code:

```
# show authors in descending order
V(ipcc_net)[order(V(ipcc_net)$betweenness, decreasing = TRUE)]

## + 148/148 vertices, named, from 44b104b:
##   [1] Nakicenovic      Smith             Shukla
##   [4] Price            Edenhofer        Schaeffer
##   [7] Krey             Bosetti          Carraro
##  [10] Riahi            Hourcade         Uerge-Vorsatz
##  [13] Rogner           Sathaye          Van-Vuuren
##  [16] Clarke           Victor           Faaij
##  [19] Lutz             Michaelowa       Paltsev
##  [22] Strachan         Edmonds          Den-Elzen
##  [25] Hoehne           Zhang            Skea
##  [28] Kopp             Stavins          Kahn-Ribeiro
## + ... omitted several vertices
```

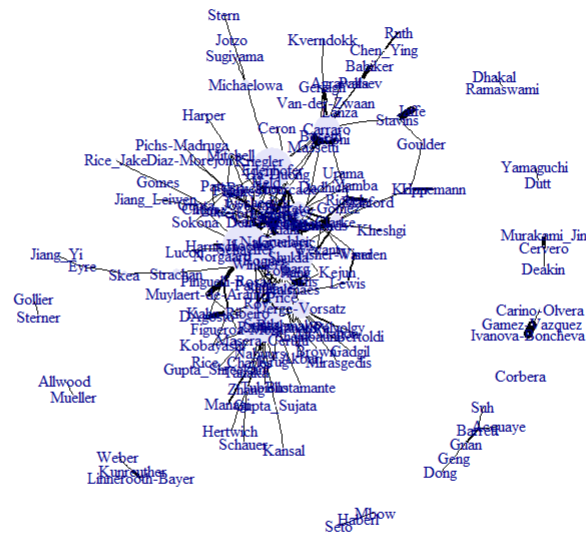
This reveals that Nakicenovic is the “most central” author, at least according to the betweenness metric.

We might want to modify our graph edges to represent the *number* of collaborations, not just the fact that two authors have collaborated. We can do this by creating a *weighted* network, and using the weights as edge widths:

```
# create network, again
ipcc_net <- graph_from_adjacency_matrix(ipcc_authors,
                                       mode = "undirected",
                                       weighted = TRUE)

# add betweenness centrality
V(ipcc_net)$betweenness <- betweenness(ipcc_net)

# plot network
plot(ipcc_net,
     edge.width=(E(ipcc_net)$weight)/5,
     vertex.size = V(ipcc_net)$betweenness/max(V(ipcc_net)$betweenness) * 20,
     vertex.color = "lavender",
     vertex.alpha = 0.5,
     vertex.frame.color = NA,
     vertex.label.cex = .5,
     edge.curved = .05,
     edge.width = .1,
     edge.color = "gray1")
```



We may also want to modify the attributes of our nodes according to their characteristics. Since geographic biases in IPCC authorship has been raised, let's start there. We can add the countries that each author represents to our data by joining the IPCC_cv data.

```
# read in cv data
ipcc_cvs <- read_csv("Data/IPCC_cv.csv")
```

Great! However, remember that we removed authors with less than 2 collaborations. So let's remove these from our CV data too. We'll use the `slice()` function, and add a negative in front of `no_colab` to specify that we *don't* want these indices.

```
# remove authors who did not collaborate
ipcc_cvs %<>%
  slice(-no_colab)
```

We notice that there are several variables around the country of origin, country of employment, and country representing at the IPCC, as well as other characteristics such as gender and institutional affiliations. We'll define a new variable indicating the *region* of the world each author is representing.

```
# define global regions
ipcc_cvs %<>%
  mutate(region = case_match(`IPCC representing country (i.e. Country of residence from IPCC)`,
    # Europe will be green
    c("Austria", "Belgium", "Denmark", "Finland",
      "France", "Germany", "Greece", "Hungary", "Italy",
      "Netherlands", "Norway", "Spain", "Sweden",
      "Switzerland", "UK") ~ "green4",
    # North America will be red
    c("Canada", "USA", "Mexico") ~ "tomato",
    # BRICS will be blue
    c("Brazil", "Russia", "India", "China",
      "South Africa") ~ "steelblue",
    # Other countries will be gold
    .default = "gold"))
```

Now let's define a variable for our vertices named `region` :

```
# define country variable
V(ipcc_net)$region <- ipcc_cvs$region
```

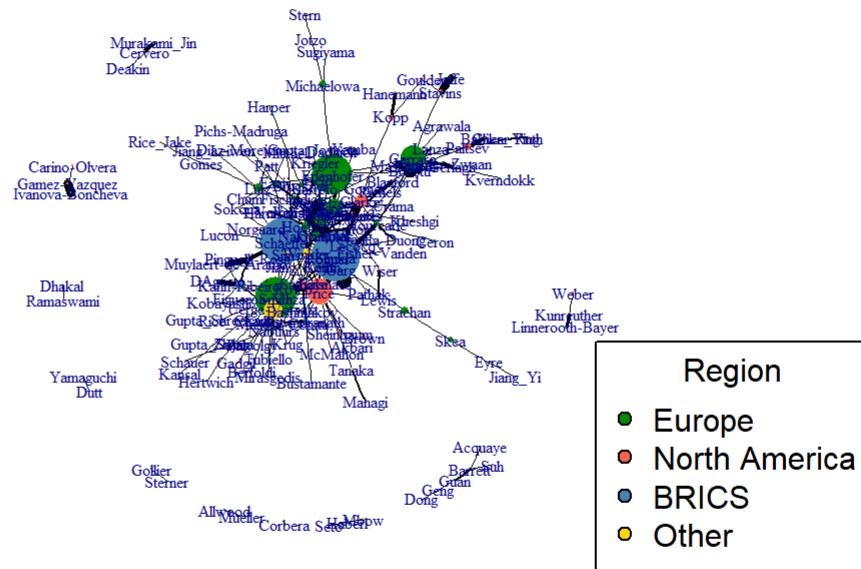
And let's try plotting our network with the nodes colored according to country:


```
# plot network
```

```
plot(ipcc_net,  
     edge.width=(E(ipcc_net)$weight)/5,  
     vertex.size = V(ipcc_net)$betweenness/max(V(ipcc_net)$betweenness) * 20,  
     vertex.color = V(ipcc_net)$region,  
     vertex.alpha = 0.5,  
     vertex.frame.color = NA,  
     vertex.label.cex = .5,  
     edge.curved = .05,  
     edge.width = .1,  
     edge.color = "gray1")
```

```
# add Legend
```

```
legend(  
  "bottomright",  
  legend = c("Europe", "North America", "BRICS", "Other"),  
  pt.bg = c("green4", "tomato", "steelblue", "gold"),  
  pch = 21,  
  cex = 1,  
  title = "Region"  
)
```



7.6 Interactive Network Visualizations

You might notice that in the previous graphs, it's still hard to read any names! If only there were a way to create an interactive chart, where one could zoom in on different parts of the network ... Here, we will use the `networkD3` package to create such an interactive visualization. The package has a `igraph_to_networkD3` function which allows us to transform our network into an object that the package can handle. We will specify the vertex feature `region` as our grouping variable, as we will want to color the nodes by this variable later (but if we were ok with the nodes all being the same color, then we wouldn't need the `group` argument).

```
library(networkD3)
```

```
# create dataframe for networkD3
```

```
ipcc_d3 <- igraph_to_networkD3(ipcc_net, group = V(ipcc_net)$region)
```

Great. Next, we'll set a color scale for our visualization. part of this code draws from JavaScript, so it looks a bit different from what we've done so far. For more information, see [here](#).

If you want to set different colors, just change the hex values in the range row. For a hex color finder, you can see [here](#). You can also label the domain row values however you like (just make sure that the number of items matches that in the range row).

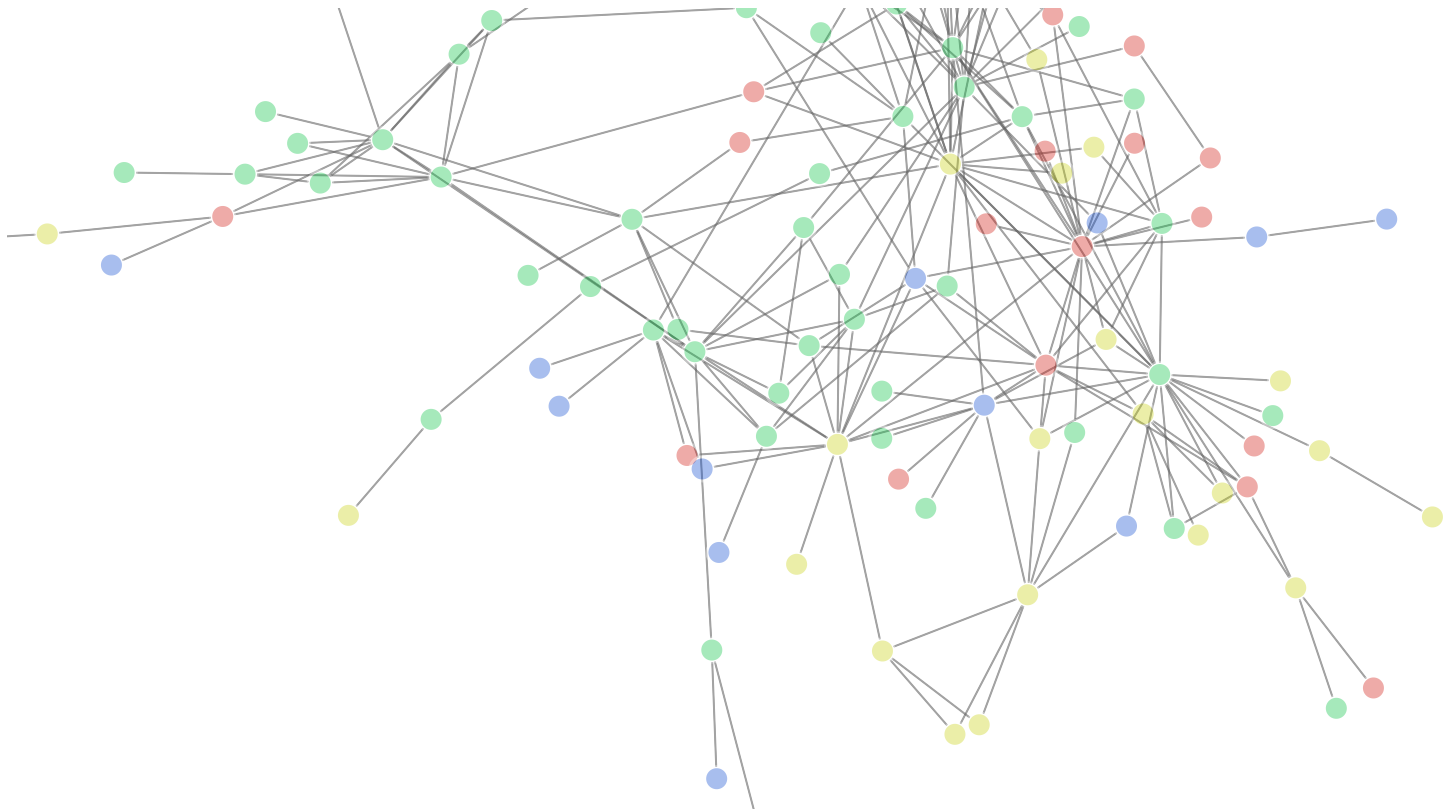
```
# set color scale
ColourScale <- 'd3.scaleOrdinal()
  .domain(["Europe", "North America", "BRICS", "Other"])
  .range(["#07C343", "#D2140C", "#0C4AD2", "#C7D20C"]);'
```

Now we can make our interactive network. We'll specify the nodes and links from our `ipcc_d3` object (this is easy since we have already converted it to `networkD3` format). we will also note our `nodeID` , "name" and our `Group` , "group." We'll make the font a bit larger than the default so we can read names, we'll add our color scale (which will be applied to the groups), and we'll set `zoom=TRUE` to enable this feature.

```
p <- forceNetwork(Links = ipcc_d3$links,
  Nodes = ipcc_d3$nodes,
  Group = "group",
  height="400px", width="800px",
  NodeID = "name",
  fontSize = 14,
  colourScale = JS(ColourScale),
  zoom = TRUE)
```

And finally, we can plot our network! Try zooming in on different parts, and highlighting the nodes to see their names.

```
# plot
p
```



7.7 Migration Flows

We can also display networks in other forms. For example, we might want to show reciprocal (or non-reciprocal) migration flows between counties. Here, we'll select Butte County, which experienced the 2018 Camp Fire, as well as the surrounding counties of Colusa, Glenn, Plumas, Sierra, Tehama, and Yuba.

The `ca_migration` data below represents migration patterns between these counties, as measured by the U.S. Census' American Community Survey. The full data can be downloaded [here](#). I use migration flows between 2016 and 2020.

We can create a chord diagram using the code below. The [R Graph Gallery](#) has similar examples of chord diagrams.

```
library(circlize)
library(stringr)
library(viridis)

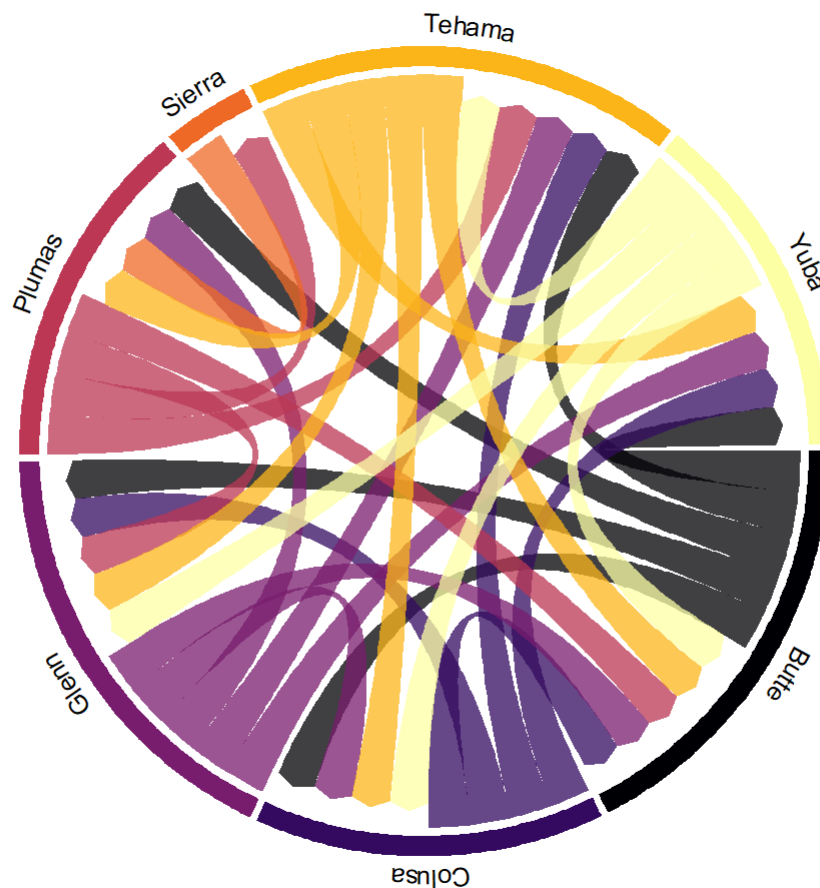
# create chord diagram
chordDiagram(ca_migration,
             transparency = 0.25,
             grid.col = inferno(7),
             directional = 1,
             direction.type = c("arrows", "diffHeight"),
             diffHeight = -0.04,
             annotationTrack = "grid",
             annotationTrackHeight = c(0.05, 0.1),
             link.arr.type = "big.arrow",
             link.sort = TRUE,
             link.largest.ontop = TRUE)

# Add text and axis
circos.trackPlotRegion(
  track.index = 1,
  bg.border = NA,
  panel.fun = function(x, y) {

    xlim = get.cell.meta.data("xlim")
    sector.index = get.cell.meta.data("sector.index")

    # Add names to the sector.
    circos.text(
      x = mean(xlim),
      y = 2.2,
      labels = sector.index,
      facing = "bending",
      cex = 0.8
    )
  }
)
```

}
)



We notice that the flows in and out of Butte County are fairly even, and similar to other counties in the region. Sierra County is smaller and more peripheral than the others, and therefore only has migration flows with Plumas County.

While we might expect to see large out-migration of Butte County, this is not visible, at least in these data. Many evacuees **fled to nearby Chico within Butte County**, and others **traveled much farther**. While we know that the Camp Fire disrupted migration patterns on various scales, we don't see strong evidence of this at the county level during the 2016-2020 period.

7.8 Problem Set 7

Recommended Resources:

Methods for Network Analysis

1. Download the “class_activities.csv” file from Canvas. Using `igraph`, create a network diagram based on who has worked with who in our class activities this quarter (you may want to use `graph_from_data_frame()`). Then plot the network.
2. Choose a centrality measure, and explain how you would interpret this in the context of the class activities data. Who is/are the most central actor(s) according to this measure?
3. Using the IPCC co-authorship data, create a network graph of all co-authors who share at least 2 publications together, excluding those with no network ties, similar to what we do in section 7.5. Graph the data using eigenvector scores, rather than betweenness scores, to plot the vertex sizes. Does this make a difference in our interpretation of the network? (Try comparing the top 5 most central authors, using the different measures of centrality).
4. Let’s say you want to give one of the IPCC authors a memo to disseminate to the others. Who would you choose? Which measure of centrality would help you arrive at your answer?
5. Finally, plot the IPCC co-authorship data, but change the color/shape of the vertices so that these correspond to authors’ gender. Describe what you see. (Note that you will need the IPCC_cv.csv dataset, which is on Canvas, to get author gender. You don’t need to remove non-collaborators from the IPCC_cv data, this has already been done).