

# R For SNA

## Workshop 1

---

Eric Brewe

Associate Professor of Physics at Drexel University

4 August 2020, last update: 2020-08-04

# Why shouldn't I just use Excel?

- R is a programming language
  - Data live separate from analysis (this is good)
  - Data are imported, manipulated, represented, but not changed.
  - This means you can't screw up!



Or at least it is hard to screw up

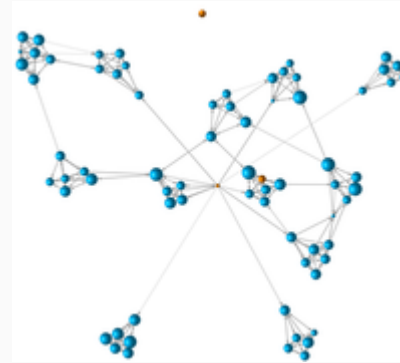
# R is good for...

- Data cleaning
- Plotting
- Summarizing
- Manipulating data
- Reproducibility
- Sharing Code



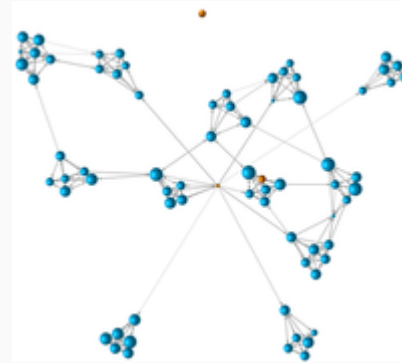
# What is a network?

- Collection of *object-like* things that are connected.
  - Nodes/actors = Object-like things (Nouns)
    - Students in a class, Words in a novel, Banks...
    - Nodes can have attributes
      - Gender,
      - Word-type,
      - Market capitalization



# What is a network?

- Collection of *object-like* things that are connected.
  - Ties/Links/Edges = Connections between nodes (Verbs)
    - Talked to each other, Are neighbors, Lend money, Sent text message...
    - Directional
    - Multiplex
    - Weighted



# Network Analysis is for the analysis of

There are four basic assumptions:

1. Nodes and interactions are interdependent\*
2. Edges allow flow between nodes
3. Network models on individuals both constrain and provide opportunity for action
4. Network models conceptualize structure as representation of lasting patterns of relations between actors

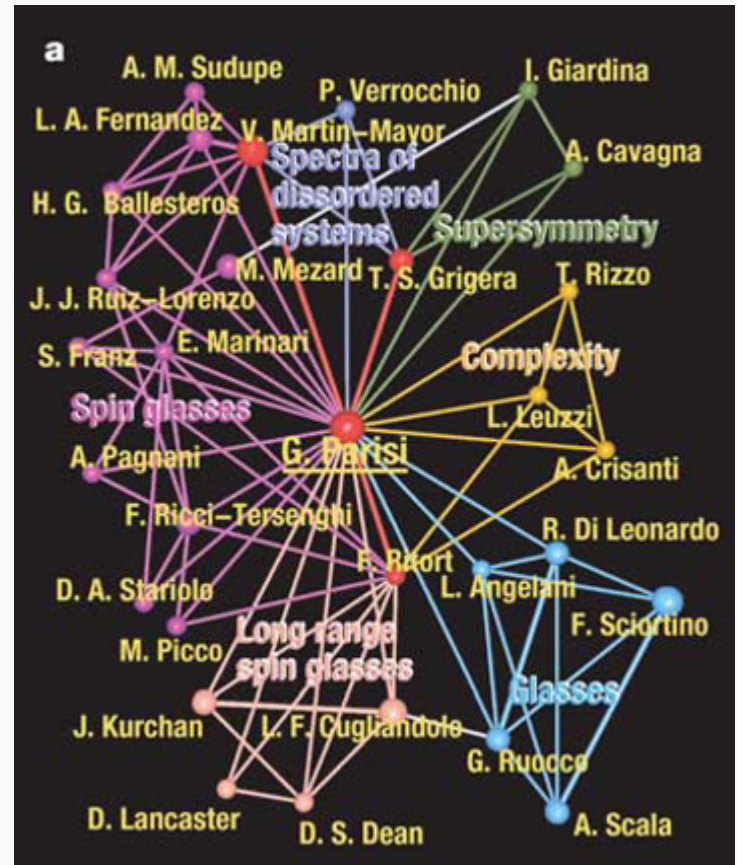
\* Violates basic assumption of inferential statistics

Wasserman, S., Faust, K. (1994). Social network analysis: Methods and applications (Vol. 8). Cambridge university press.

# What can we do with it?

## Ego-Level Analyses

- What can we know about the network of one person?
  - Ego density
  - Number of neighbors
  - Number of connected neighbors





# What can we do with it?

## Node-Level Analyses

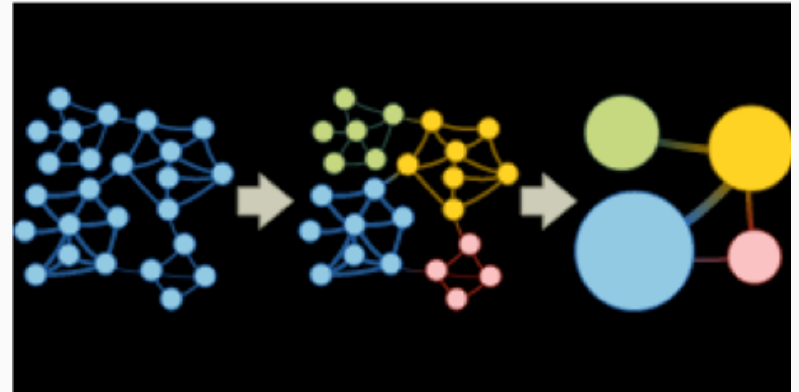
- What can we know about the position of people in a network?
  - Degree (In/Out/Total)
  - Geodesic Distance (Kevin Bacon)
  - PageRank
  - Target Entropy



# What can we do with it?

## Whole Network Analyses

- What can we say about a whole network?
  - Density, Average path length, Giant component
  - Clustering
  - Homophily
  - Modeling
    - Block models
    - Small worldness



# Historical Foundations

Joseph Moreno & Helen Hall Jennings (1932)

- Established foundations of SNA

Quantitative Sociology/Anthropology

- Davis Southern Women's Club (1941)
- Small World Problem (1967)
- Zachary's Karate Club (1977)

Seminal Articles

- Milgram, Stanley "The small world problem" *Psychology Today* 2:1, (1967)
- Grannovetter, Mark S. "The strength of weak ties" *American Journal of Sociology* (1973)



# Modern Foundations of Network Analysis

## Sociophysics (1990s)

- Graph theory
- Information theory
- Computing
- Used to study
  - Internet
  - Power grid
  - Transportation networks

## Seminal Articles

- Watts & Strogatz "Collective dynamics of small world networks" Nature (1998)
- Page, Brin, Motwani, & Winograd "The PageRank citation ranking: Bringing order to the web" Stanford InfoLab (1999)

# Important Takeaways from History

Two main camps

Statistical -> hypothesis testing

Graph theoretic -> network models and simulation

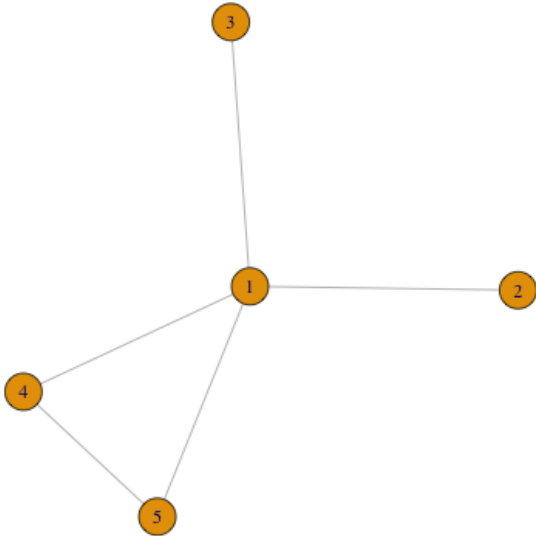
They often don't agree.

There is often distain.

They have different language, journals, conferences

# Network Data in R

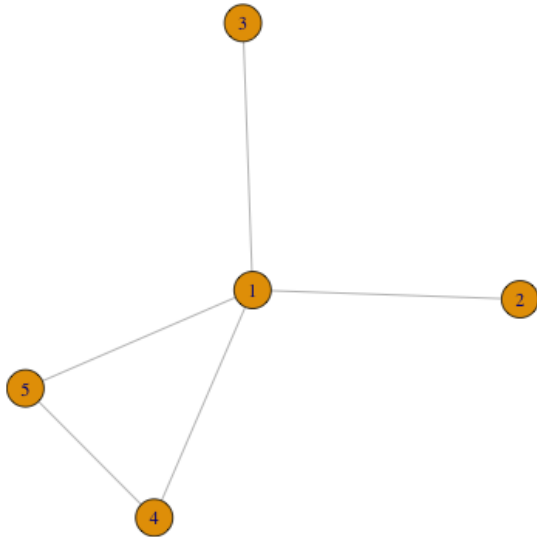
## Sociomatrix/Adjacency Matrix



```
## 5 x 5 sparse Matrix of class "dgCMatrix"
##
## [1,] . 1 1 1 1
## [2,] 1 . . . .
## [3,] 1 . . . .
## [4,] 1 . . . 1
## [5,] 1 . . 1 .
```

# Network Data in R

## Edgelist



```
## [[1]]
## + 4/5 edges from 0abd530:
## [1] 1--2 1--3 1--4 1--5
##
## [[2]]
## + 1/5 edge from 0abd530:
## [1] 1--2
##
## [[3]]
## + 1/5 edge from 0abd530:
## [1] 1--3
##
## [[4]]
## + 2/5 edges from 0abd530:
## [1] 1--4 4--5
##
## [[5]]
## + 2/5 edges from 0abd530:
## [1] 1--5 4--5
```

# What does this mean in terms of learning R?

- We need to know something about the different types of data!

## Data types (at least some of them)

- **Logical** (T/F, 1/0)
- **Integers** (whole numbers)
- **Numeric** (numbers with decimal places)
- **Complex** (I never use these)

## Data storage (at least some of them)

- **Vectors** = long columns of data (can be any type, but only one type of data)
- **Dataframes** = like Excel pages (columns can hold different types of data)
- **Matrices** = like dataframes, but they have named rows/columns
  - Adjacency Matrices are of type matrix.
- **Lists** = the junkdrawer, can hold any type of data (including dataframes or matrices)
  - in igraph, networks are stored as lists.



# What you need to know about R?

- R is the programming language.
- RStudio is the Integrated Development Environment (IDE)
- Packages: groups of functions that are developed as open source
- Base R
  - The group of packages preloaded into R
- Tidyverse
  - Family of packages that are designed with the theory that programming should be readable by humans.
- igraph
  - Package that is very useful for doing network analyses

Lets do this!

# Using R and RStudio

- Open RStudio (this will automatically open R)
- Navigate to your folder titled "RForSNA"
- In RStudio -> File -> New Project
  - Select "Existing Directory" (Unless you know Git)
- In RStudio -> File -> New File -> RMarkdown
- Save this file in the folder titled "RForSNA"

# Lets Take a Tour of RStudio IDE

# Let's Install Some Packages

You'll only need to do this once.

## In Console

To install tidyverse package...

```
install.packages("tidyverse")
```

Repeat this with the following packages:

igraph tidygraph here ggraph

# Let's Load Some Packages

You'll need to do this every time you restart R.

## In Console

To load tidyverse package...

```
library(tidyverse) #tools for cleaning data  
library(igraph) #package for doing network analysis  
library(tidygraph) #tools for doing tidy networks  
library(here) #tools for project-based workflow  
library(ggraph) #plotting tools for networks
```

Once you have done this, you will want to put include a code chunk with all of your libraries into your markdown document so that you don't have to type this every time.

# Let's get data into R.

I've sent you a csv file that includes the data for workshop 1, I hope you saved this in your folder titled "data".

If you have loaded the package "here" this should just work. If you have not loaded the "here" package you will need to set the working directory.

Again, you will want to include this as a code chunk in your RMD file.

```
#This loads the csv and saves it as a dataframe titled WorkshopData
```

```
WorkshopData <- read_csv(here("data", "AnonSurveyData.csv"))
```

## Let's have a look at the data

```
glimpse(WorkshopData)
```

```
## Rows: 34
```

```
## Columns: 18
```

```
## $ ID <dbl> 5106, 6633, 7599, 4425, 2495, 6355, 8810, 3877
```

```
## $ StartDate      <dtm> 2020-07-30 12:15:20, 2020-07-30 12:18:50, 2020-07-30 12:22:20, 2020-07-30 12:25:50, 2020-07-30 12:29:20, 2020-07-30 12:32:50, 2020-07-30 12:36:20, 2020-07-30 12:39:50, 2020-07-30 12:43:20, 2020-07-30 12:46:50, 2020-07-30 12:50:20, 2020-07-30 12:53:50, 2020-07-30 12:57:20, 2020-07-30 13:00:50, 2020-07-30 13:04:20, 2020-07-30 13:07:50, 2020-07-30 13:11:20, 2020-07-30 13:14:50, 2020-07-30 13:18:20, 2020-07-30 13:21:50, 2020-07-30 13:25:20, 2020-07-30 13:28:50, 2020-07-30 13:32:20, 2020-07-30 13:35:50, 2020-07-30 13:39:20, 2020-07-30 13:42:50, 2020-07-30 13:46:20, 2020-07-30 13:49:50, 2020-07-30 13:53:20, 2020-07-30 13:56:50, 2020-07-30 14:00:20, 2020-07-30 14:03:50, 2020-07-30 14:07:20, 2020-07-30 14:10:50, 2020-07-30 14:14:20, 2020-07-30 14:17:50, 2020-07-30 14:21:20, 2020-07-30 14:24:50, 2020-07-30 14:28:20, 2020-07-30 14:31:50, 2020-07-30 14:35:20, 2020-07-30 14:38:50, 2020-07-30 14:42:20, 2020-07-30 14:45:50, 2020-07-30 14:49:20, 2020-07-30 14:52:50, 2020-07-30 14:56:20, 2020-07-30 14:59:50, 2020-07-30 15:03:20, 2020-07-30 15:06:50, 2020-07-30 15:10:20, 2020-07-30 15:13:50, 2020-07-30 15:17:20, 2020-07-30 15:20:50, 2020-07-30 15:24:20, 2020-07-30 15:27:50, 2020-07-30 15:31:20, 2020-07-30 15:34:50, 2020-07-30 15:38:20, 2020-07-30 15:41:50, 2020-07-30 15:45:20, 2020-07-30 15:48:50, 2020-07-30 15:52:20, 2020-07-30 15:55:50, 2020-07-30 15:59:20, 2020-07-30 16:02:50, 2020-07-30 16:06:20, 2020-07-30 16:09:50, 2020-07-30 16:13:20, 2020-07-30 16:16:50, 2020-07-30 16:20:20, 2020-07-30 16:23:50, 2020-07-30 16:27:20, 2020-07-30 16:30:50, 2020-07-30 16:34:20, 2020-07-30 16:37:50, 2020-07-30 16:41:20, 2020-07-30 16:44:50, 2020-07-30 16:48:20, 2020-07-30 16:51:50, 2020-07-30 16:55:20, 2020-07-30 16:58:50, 2020-07-30 17:02:20, 2020-07-30 17:05:50, 2020-07-30 17:09:20, 2020-07-30 17:12:50, 2020-07-30 17:16:20, 2020-07-30 17:19:50, 2020-07-30 17:23:20, 2020-07-30 17:26:50, 2020-07-30 17:30:20, 2020-07-30 17:33:50, 2020-07-30 17:37:20, 2020-07-30 17:40:50, 2020-07-30 17:44:20, 2020-07-30 17:47:50, 2020-07-30 17:51:20, 2020-07-30 17:54:50, 2020-07-30 17:58:20, 2020-07-30 18:01:50, 2020-07-30 18:05:20, 2020-07-30 18:08:50, 2020-07-30 18:12:20, 2020-07-30 18:15:50, 2020-07-30 18:19:20, 2020-07-30 18:22:50, 2020-07-30 18:26:20, 2020-07-30 18:29:50, 2020-07-30 18:33:20, 2020-07-30 18:36:50, 2020-07-30 18:40:20, 2020-07-30 18:43:50, 2020-07-30 18:47:20, 2020-07-30 18:50:50, 2020-07-30 18:54:20, 2020-07-30 18:57:50, 2020-07-30 19:01:20, 2020-07-30 19:04:50, 2020-07-30 19:08:20, 2020-07-30 19:11:50, 2020-07-30 19:15:20, 2020-07-30 19:18:50, 2020-07-30 19:22:20, 2020-07-30 19:25:50, 2020-07-30 19:29:20, 2020-07-30 19:32:50, 2020-07-30 19:36:20, 2020-07-30 19:39:50, 2020-07-30 19:43:20, 2020-07-30 19:46:50, 2020-07-30 19:50:20, 2020-07-30 19:53:50, 2020-07-30 19:57:20, 2020-07-30 20:00:50, 2020-07-30 20:04:20, 2020-07-30 20:07:50, 2020-07-30 20:11:20, 2020-07-30 20:14:50, 2020-07-30 20:18:20, 2020-07-30 20:21:50, 2020-07-30 20:25:20, 2020-07-30 20:28:50, 2020-07-30 20:32:20, 2020-07-30 20:35:50, 2020-07-30 20:39:20, 2020-07-30 20:42:50, 2020-07-30 20:46:20, 2020-07-30 20:49:50, 2020-07-30 20:53:20, 2020-07-30 20:56:50, 2020-07-30 21:00:20, 2020-07-30 21:03:50, 2020-07-30 21:07:20, 2020-07-30 21:10:50, 2020-07-30 21:14:20, 2020-07-30 21:17:50, 2020-07-30 21:21:20, 2020-07-30 21:24:50, 2020-07-30 21:28:20, 2020-07-30 21:31:50, 2020-07-30 21:35:20, 2020-07-30 21:38:50, 2020-07-30 21:42:20, 2020-07-30 21:45:50, 2020-07-30 21:49:20, 2020-07-30 21:52:50, 2020-07-30 21:56:20, 2020-07-30 21:59:50, 2020-07-30 22:03:20, 2020-07-30 22:06:50, 2020-07-30 22:10:20, 2020-07-30 22:13:50, 2020-07-30 22:17:20, 2020-07-30 22:20:50, 2020-07-30 22:24:20, 2020-07-30 22:27:50, 2020-07-30 22:31:20, 2020-07-30 22:34:50, 2020-07-30 22:38:20, 2020-07-30 22:41:50, 2020-07-30 22:45:20, 2020-07-30 22:48:50, 2020-07-30 22:52:20, 2020-07-30 22:55:50, 2020-07-30 22:59:20, 2020-07-30 23:02:50, 2020-07-30 23:06:20, 2020-07-30 23:09:50, 2020-07-30 23:13:20, 2020-07-30 23:16:50, 2020-07-30 23:20:20, 2020-07-30 23:23:50, 2020-07-30 23:27:20, 2020-07-30 23:30:50, 2020-07
```

```
## $ EndDate      <dtm> 2020-07-30 12:18:59, 2020-07-30 12:20:21, 2020-07-30 12:20:21
```

```
## $ Status      <chr> "IP Address", "IP Address", "IP Address", "IP
```

```
## $ Progress      <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, :
```

```
## $ `Duration (in seconds)` <dbl> 219, 91, 137, 97, 127, 233, 97, 144, 103, 231
```

```
## $ Finished      <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE
```

```
## $ RecordedDate      <dtm> 2020-07-30 12:18:59, 2020-07-30 12:20:22, 2020-07-30 12:20:22
```

```
## $ SurveyID      <chr> "R ssTGHZwy5EpQaNX", "R 2fv9VCk0tjdrDOr", "R 1
```

```
## $ ExternalReference      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
```

```
## $ DistributionChannel <chr> "email", "email", "email", "email", "email",
```

```
## $ UserLanguage      <chr> "EN", "EN", "EN", "EN", "EN", "EN", "EN", "EN"
```

```
## $ Q2 <chr> "morning person.", "morning person.", "night o
```

```
## $ Q3      <chr> "Brownies", "I don't care for dessert.", "Ice
```

```
## $ Q4 <chr> "Coffee,Water,The tears of my enemies", "Coffee"
```

```
## $ Q5 <dbl> 350, 12, 300, 0, 264, 4, 289, 550, 349, 300, 4
```

```
## $ Q6 <dbl> 350, 56, 5000, 50, 286, 250, 76, 185, 108, 220
```

# Let's start cleaning up.

## First, we don't need most of that data

There is a ton of data there that doesn't make sense for us to keep around.

We will use the '%>%' (pipe) operator and the verb select

```
WorkshopData %>%
```

```
  select(ID,Q2:Q6) -> WorkshopData
```

```
glimpse(WorkshopData)
```

```
## Rows: 34
```

```
## Columns: 6
```

```
## $ ID <dbl> 5106, 6633, 7599, 4425, 2495, 6355, 8810, 3877, 1554, 7743, 8353, 2
```

```
## $ Q2 <chr> "morning person.", "morning person.", "night owl.", "morning person
```

```
## $ Q3 <chr> "Brownies", "I don't care for dessert.", "Ice Cream", "I don't care
```

```
## $ Q4 <chr> "Coffee,Water,The tears of my enemies", "Coffee,Tea,Water,Milk", "I
```

```
## $ Q5 <dbl> 350, 12, 300, 0, 264, 4, 289, 550, 349, 300, 424, 426, 231, 290, 10
```

```
## $ Q6 <dbl> 350, 56, 5000, 50, 286, 250, 76, 185, 108, 220, 500, 350, 412, 1135
```



# Let's start cleaning up.

Now we should actually take a look at the data

```
WorkshopData %>%  
  head()
```

```
## # A tibble: 6 x 6  
##       ID Q2                Q3                Q4  
##   <dbl> <chr>                <chr>                <chr>  
## 1  5106 morning perso... Brownies                Coffee,Water,The tears of my e...  
## 2  6633 morning perso... I don't care for dess... Coffee,Tea,Water,Milk  
## 3  7599 night owl.      Ice Cream                Fruit Juice,Tea,Water,Fizzy Wa...  
## 4  4425 morning perso... I don't care for dess... Fruit Juice,Coffee,Water  
## 5  2495 morning perso... Brownies                Coffee,Water  
## 6  6355 night owl.      Ice Cream                Fruit Juice,Tea,Water
```

# Let's check out the power of R

I am going to blast through these next slides, to show you some of the things that you might want to do with R

# Let's start visualizing the data

For categorical data, you might want to get some counts.

Here is code to do this for the question about morning or night person.

```
WorkshopData %>%  
  select(Q2) %>%  
  group_by(Q2) %>%  
  tally()
```

```
## # A tibble: 2 x 2  
##   Q2                n  
##   <chr>          <int>  
## 1 morning person.    19  
## 2 night owl.      15
```

# Let's start visualizing the data

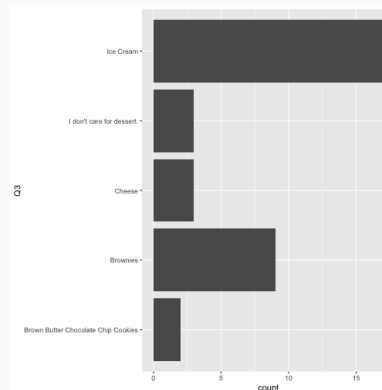
For categorical data, you might want to get a histogram.

Here is code to do this for the favorite dessert type.

```
WorkshopData %>%  
  select(Q3) %>%  
  group_by(Q3) %>%  
  tally()
```

```
## # A tibble: 5 x 2  
##   Q3  
##   <chr>  
## 1 Brown Butter Chocolate Chip Cookies  
## 2 Brownies  
## 3 Cheese  
## 4 I don't care for dessert.  
## 5 Ice Cream
```

```
WorkshopData %>%  
  select(Q3) %>%  
  ggplot(aes(y = Q3)) +  
  geom_bar()
```



17

# Let's explore beverage choices

```
WorkshopData %>%  
  select(ID, Q4) %>%  
  head( )
```

```
## # A tibble: 6 x 2  
##       ID Q4  
##   <dbl> <chr>  
## 1  5106 Coffee,Water,The tears of my enemies  
## 2  6633 Coffee,Tea,Water,Milk  
## 3  7599 Fruit Juice,Tea,Water,Fizzy Water  
## 4  4425 Fruit Juice,Coffee,Water  
## 5  2495 Coffee,Water  
## 6  6355 Fruit Juice,Tea,Water
```

Notice, these are *not* tidy data, more than one variable per line

# Let's dummy code the beverage data

For **tidy** data we want one value per row

```
WorkshopData %>%  
  select(ID, Q4) %>%  
  separate_rows(Q4, sep = ",") %>%  
  head(10)
```

```
## # A tibble: 10 x 2  
##       ID Q4  
##   <dbl> <chr>  
## 1  5106 Coffee  
## 2  5106 Water  
## 3  5106 The tears of my enemies  
## 4  6633 Coffee  
## 5  6633 Tea  
## 6  6633 Water  
## 7  6633 Milk  
## 8  7599 Fruit Juice
```

We can dummy code these

```
WorkshopData %>%  
  select(ID, Q4) %>%  
  separate_rows(Q4, sep = ",") %>%  
  mutate(Checked = 1) %>%  
  pivot_wider(names_from = Q4,  
              values_from = Checked,  
              values_fill = 0)
```

```
## # A tibble: 34 x 11  
##       ID Coffee Water `The tears of m`  
##   <dbl> <dbl> <dbl> <dbl>  
## 1  5106         1         1  
## 2  6633         1         1  
## 3  7599         0         1  
## 4  4425         1         1  
## 5  2495         1         1  
## 6  6355         0         1
```

# Let's look at some quantitative data

First, let's summarize the reading data.

```
WorkshopData %>%  
  select(Q5) %>%  
  summarize(Ave = mean(Q5, na.rm = TRUE),  
            SD = sd(Q5, na.rm = TRUE))
```

```
## # A tibble: 1 x 2  
##   Ave    SD  
##   <dbl> <dbl>  
## 1  327.  247.
```

# Let's investigate groups

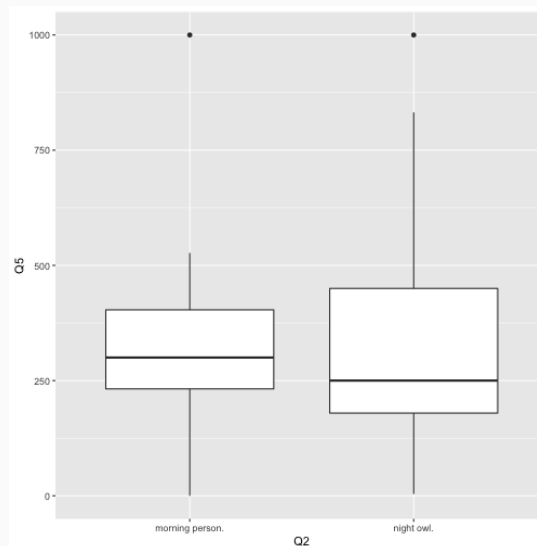
Are morning people or night owls reading longer books?

```
WorkshopData %>%  
  select(Q2, Q5) %>%  
  group_by(Q2) %>%  
  summarize(Ave = mean(Q5),  
            SD = sd(Q5))
```

```
## # A tibble: 2 x 3  
##   Q2          Ave    SD  
##   <chr>      <dbl> <dbl>  
## 1 morning person. 321  215.  
## 2 night owl.   335  289.
```

We might want to use a boxplot to display these data

```
WorkshopData %>%  
  select(Q2, Q5) %>%  
  ggplot(., aes(x = Q2, y = Q5)) +  
  geom_boxplot()
```



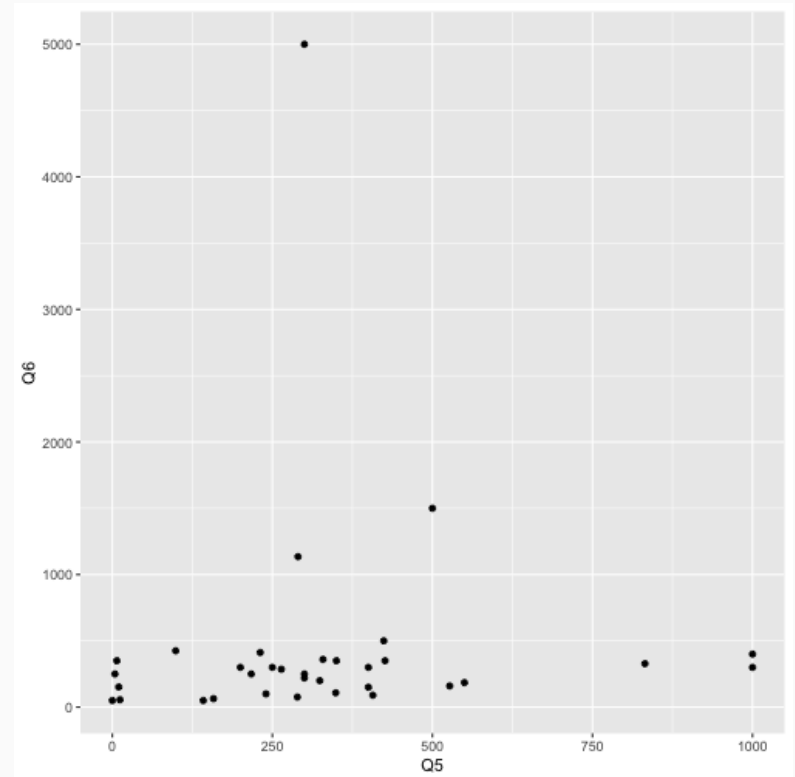


# Let's look at readers vs. blueberries

Is there a relationship between length of book and estimates on number of blueberries?

Could do a scatter plot

```
WorkshopData %>%  
  select(Q5:Q6) %>%  
  mutate(Q5 = as.numeric(Q5), Q6 = as.numeric(Q6))  
ggplot(aes(x = Q5, y = Q6)) +  
  geom_point()
```



# Let's look at readers vs. blueberries

Is there a relationship between length of book and estimates on number of blueberries?

Or you could do a linear model

```
summary(lm(Q6 ~ Q5, data = WorkshopData))
```

```
##  
## Call:  
## lm(formula = Q6 ~ Q5, data = WorkshopData)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -365.6  -313.8  -179.1   -98.8  4563.5   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  382.9414    249.7136   1.534    0.135      
## Q5           0.1785     0.6126   0.291    0.773
```

# Let's prep our data for SNA

We will need to prep two separate files...

1. An edgelist
2. A file of attributes of the nodes.

# Let's make an edgelist

There is an issue here.

I wanted to make these data anonymous (so we don't know who likes scotch)

But to do that I had to make the edgelist for you.

So I sent you an edgelist as a csv. Sorry.

## Check out the edgelist

```
EL = read_csv(here("data", "AnonEL.csv"))  
head(EL)
```

```
## # A tibble: 6 x 2  
##       ID Connections  
##   <dbl>         <dbl>  
## 1  5106          5106  
## 2  6633          6196  
## 3  7599          5462  
## 4  4425          7743  
## 5  2495          3940
```

# Let's assemble our node attributes

Before we can convert our Edgelist to a network, we should add in the attributes.

We have several candidate attributes:

- Morning vs. Night
- Dessert Type
- Pages in book

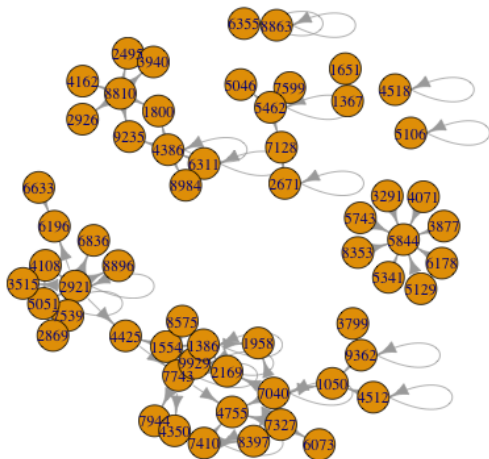
We will develop a separate dataframe for the attributes.

```
WorkshopData %>%  
  select(ID, Q2, Q3, Q5) -> AttributeDf
```

# Let's assemble our node attributes

Experience tells me that when you try to add attributes that you often make a mistake where the number attributes don't match up well to the number of nodes...but lets see.

```
gr <- graph_from_data_frame(EL, directed = TRUE)  
  
plot(gr)
```



# Let's add some attributes

So actually the easiest way to add the attributes is to add them while you make the graph.

But that isn't as easy as it seems

```
gr %>%  
  activate(nodes) %>%  
  mutate(AMPM = AttributeDf$Q2)
```

But that isn't as easy as it seems...

---

The warning was: "Input `AMPM` must be size 60 or 1, not 34."

What this means is we need to take our attributes dataframe and make sure all the nodes are listed.

To do this we need to:

- Compile a dataframe of all nodes listed in the graph
- Use a join to add attributes to this dataframe

# Let's sort out these attributes (pt 2)

```
#This will get a vector of all nodes
```

```
gr %>%  
  activate(nodes) %>%  
  as_tibble() %>%  
  transmute(ID = name) %>%  
  mutate(ID = as.numeric(ID)) -> GrNodes
```

```
#Now we pull in the attributes using a left_join
```

```
NodeAttributes = left_join(GrNodes, AttributeDf, by = "ID")
```

You should inspect Node Attributes



# Let's sort out these attributes (pt 3)

## Inspect the head

```
head(NodeAttributes)
```

```
## # A tibble: 6 x 4
##       ID Q2                Q3                Q5
##   <dbl> <chr>                <chr>                <dbl>
## 1  5106 morning person. Brownies                350
## 2  6633 morning person. I don't care for dessert.    12
## 3  7599 night owl.      Ice Cream                300
## 4  4425 morning person. I don't care for dessert.     0
## 5  2495 morning person. Brownies                264
## 6  6355 night owl.      Ice Cream                 4
```

# Let's sort out these attributes (pt 4)

## Inspect the tail

```
tail(NodeAttributes)
```

```
## # A tibble: 6 x 4
##       ID Q2      Q3      Q5
##   <dbl> <chr> <chr> <dbl>
## 1  7128 <NA>   <NA>   NA
## 2  1050 <NA>   <NA>   NA
## 3  3799 <NA>   <NA>   NA
## 4  1651 <NA>   <NA>   NA
## 5  8984 <NA>   <NA>   NA
## 6  1958 <NA>   <NA>   NA
```

# Let's finally assemble this graph.

```
gr %>%  
  as_tbl_graph() %>%  
  activate(nodes) %>%  
  mutate(AMPM = NodeAttributes$Q2) %>%  
  mutate(Dessert = NodeAttributes$Q3) %>%  
  mutate(Pages = NodeAttributes$Q5) -> gr  
  
summary(gr)
```

```
## IGRAPH ebe8c11 DN-- 60 101 --
```

```
## + attr: name (v/c), AMPM (v/c), Dessert (v/c), Pages (v/n)
```

# Let's finally assemble this graph.

Take a look at your graph

```
## # A tbl_graph: 60 nodes and 101 edges
## #
## # A directed multigraph with 8 components
## #
## # Node Data: 60 x 4 (active)
##   name  AMPM      Dessert      Pages
##   <chr> <chr>      <chr>      <dbl>
## 1 5106  morning person. Brownies      350
## 2 6633  morning person. I don't care for dessert. 12
## 3 7599  night owl.      Ice Cream      300
## 4 4425  morning person. I don't care for dessert. 0
## 5 2495  morning person. Brownies      264
## 6 6355  night owl.      Ice Cream      4
## # ... with 54 more rows
## #
## # Edge Data: 101 x 2
##   from  to
##   <int> <int>
## 1      1      1
```

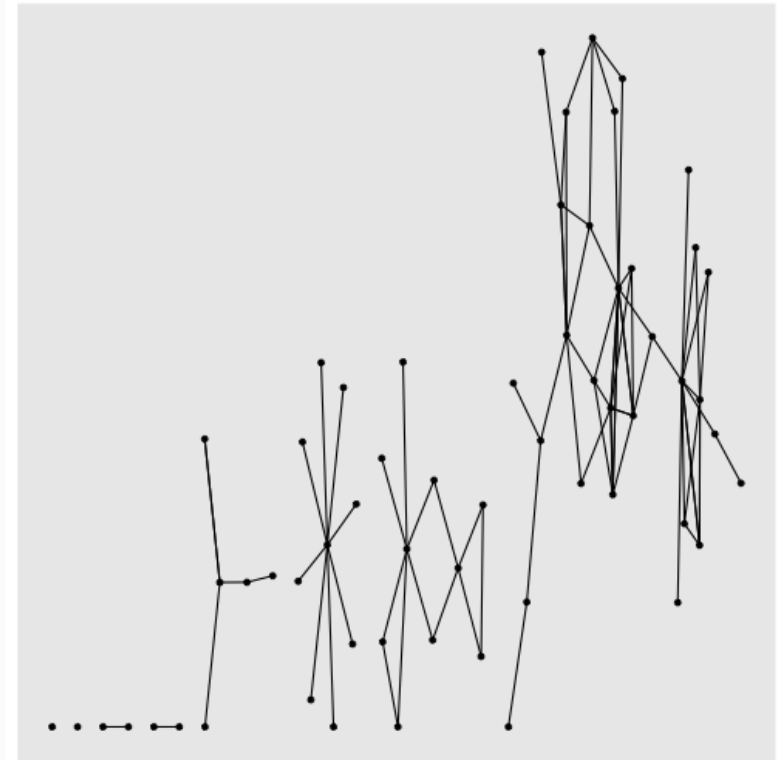
# Let's make our plots look a bit better

We can add various elements to our plot

- Color (good for grouping)
- Shape (good for grouping)
- Size (good for numeric)
- Text (good for labels)
- Layout

This is the most basic plot

```
```r
ggraph(gr) +
  geom_edge_link() +
  geom_node_point()
```
```



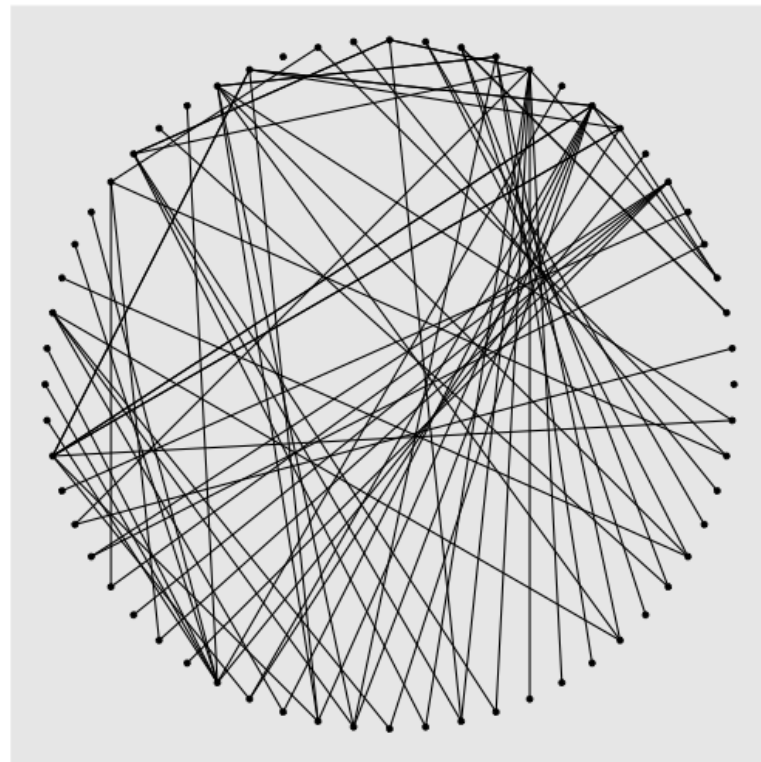
Not super pretty

# Let's make our plots look a bit better

We can add various elements to our plot

- Layout, there are lots of options -Try circle?

```
```r
ggraph(gr, layout = 'circle') +
  geom_edge_link() +
  geom_node_point()
```
```

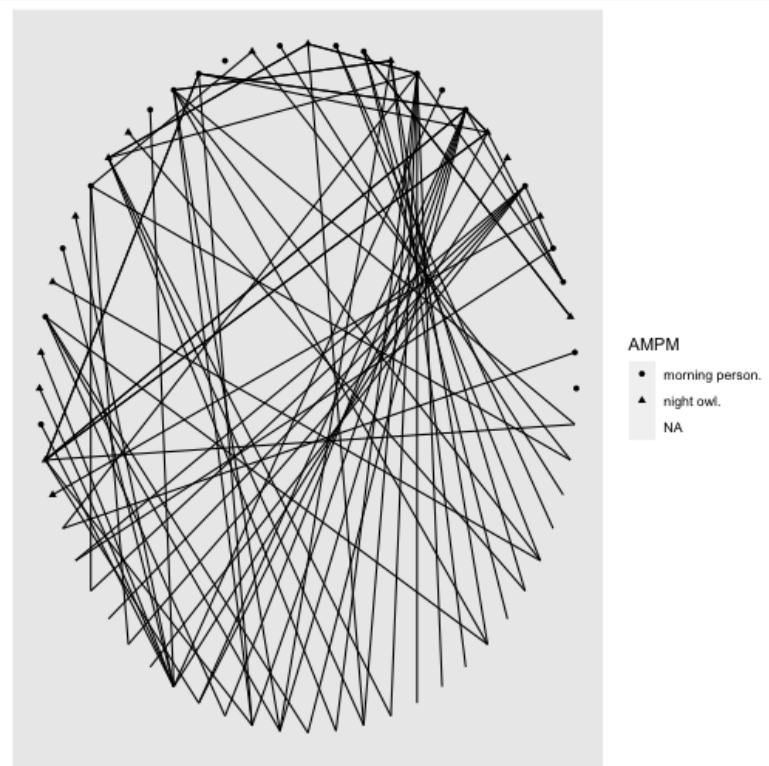


# Let's make our plots look a bit better

We can add various elements to our plot

- Shape, we can make night owls one shape and morning people a different shape...

```
```r
ggraph(gr, layout = 'circle') +
  geom_edge_link() +
  geom_node_point(aes(shape = AMPM))
```
```



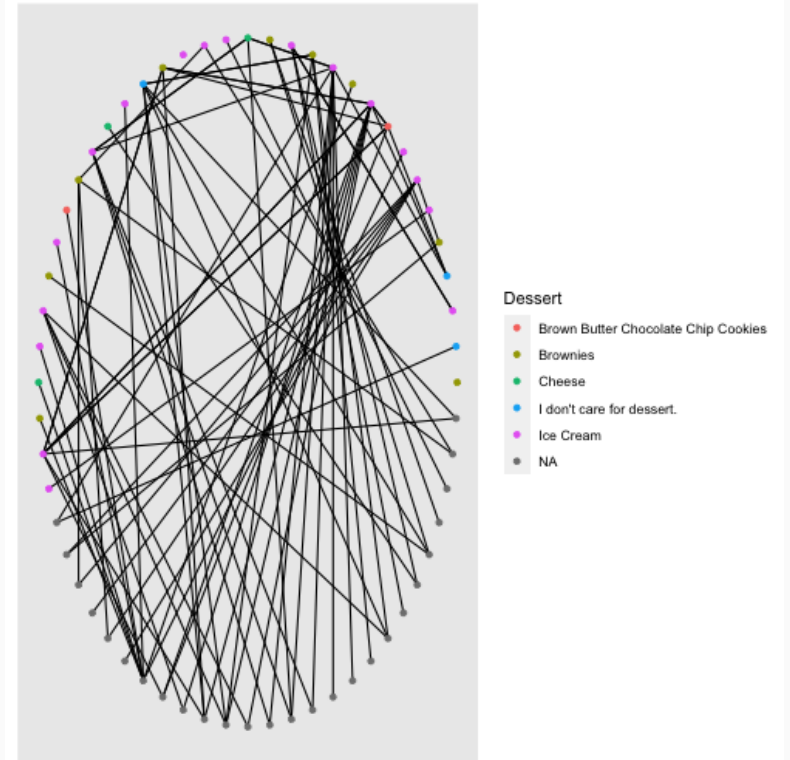
Not great

# Let's make our plots look a bit better

We can add various elements to our plot

- Color, lets use the dessert type to define a color

```
```r
ggraph(gr, layout = 'circle') +
  geom_edge_link() +
  geom_node_point(aes(color = Dessert))
```
```



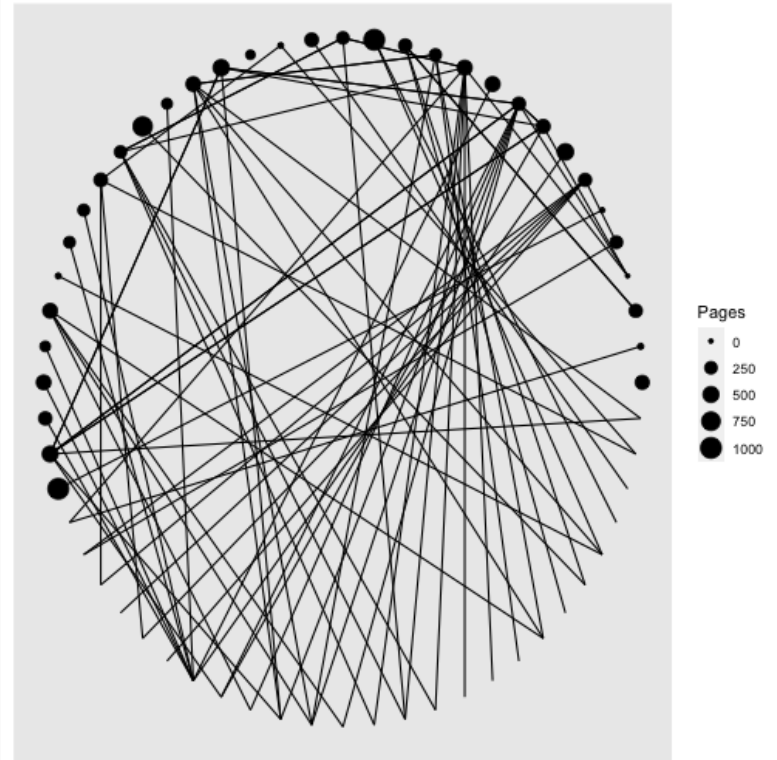


# Let's make our plots look a bit better

We can add various elements to our plot

- Size, lets make the nodes different sizes based on the number of pages in the last book read.

```
```r
ggraph(gr, layout = 'circle') +
  geom_edge_link() +
  geom_node_point(aes(size = Pages))
```
```



# Let's make our plots look a bit better

We can add various elements to our plot

- Let's jam them all together!

```
```r
ggraph(gr, layout = 'circle') +
  geom_edge_link() +
  geom_node_point(aes(shape = AMPM,
                      color = Dessert,
                      size = Pages))
```
```

