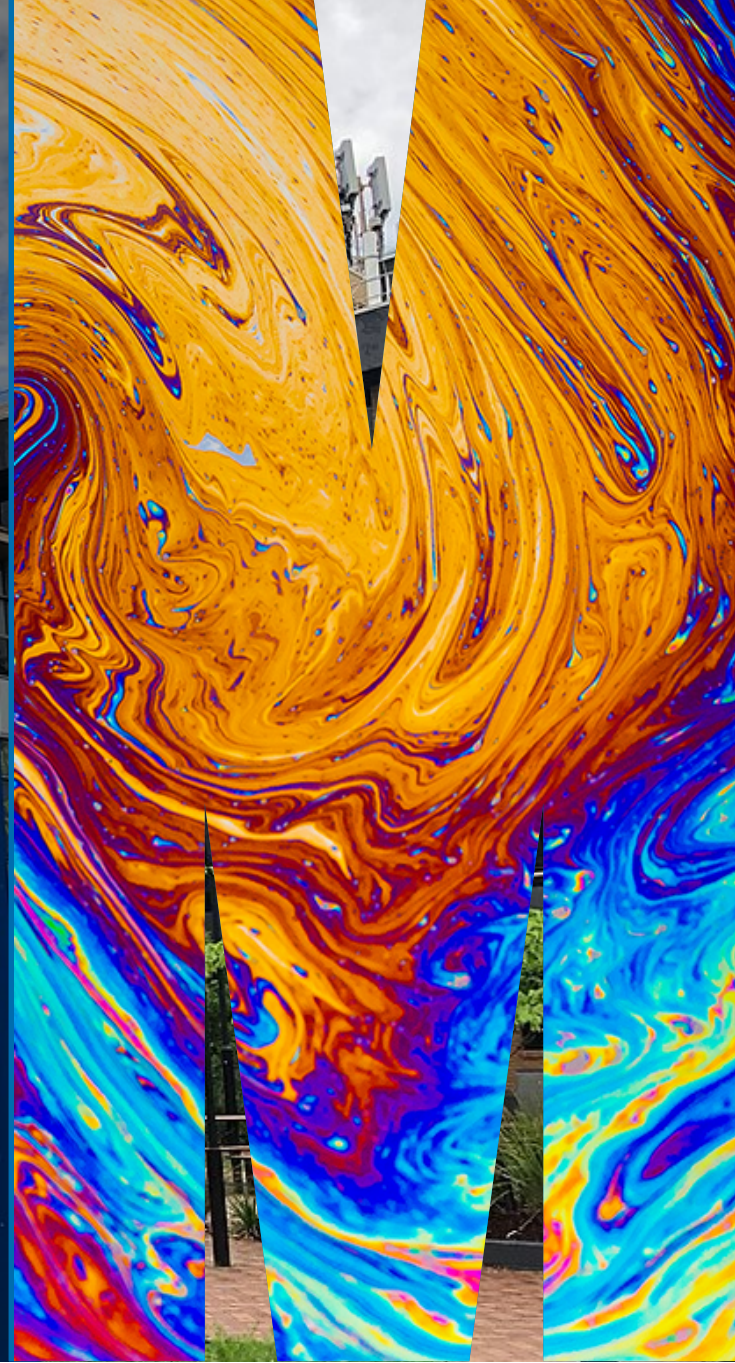
ETC5510: Introduction to Data Analysis

Week 9, part A

Networks and Graphs



Lecturer: *Nicholas Tierney & Stuart Lee*

Department of Econometrics and Business Statistics

ETC5510.Clayton-x@monash.edu

May 2020

Announcements

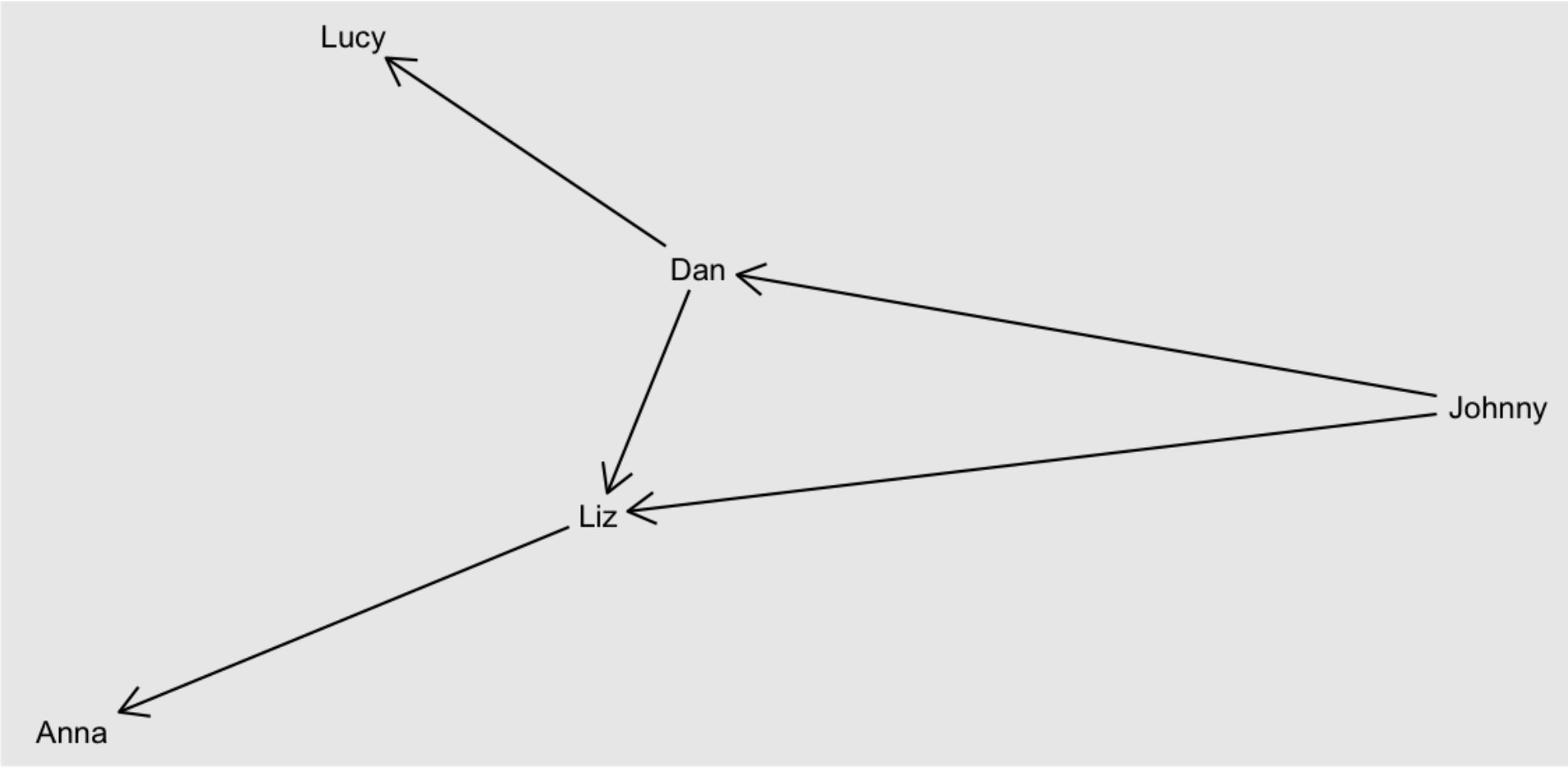
- Assignment 2: due 20th May at 5pm
- Project deadlines:
  - **Milestone 2 (20th May)** : Finalised team/team members, a paragraph about what you're going to do and links to your data sources.
  - **Milestone 3 (27th May)** : Upload data, and Rmd file, and html describing your data, the cleaning you've done and what's left.

Network analysis

A description of phone calls

- Johnny --> Liz
- Liz --> Anna
- Johnny -- > Dan
- Dan --> Liz
- Dan --> Lucy

As a graph



And as an association matrix

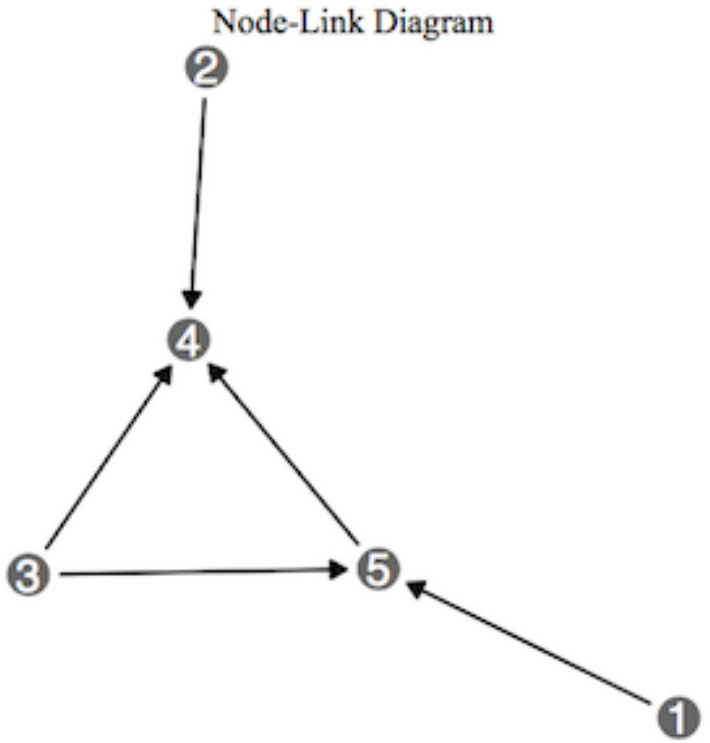
```
##        Johnny Dan Liz Lucy Anna
## Johnny      0   1   1    0    0
## Dan         0   0   1    1    0
## Liz         0   0   0    0    1
## Lucy        0   0   0    0    0
## Anna        0   0   0    0    0
```

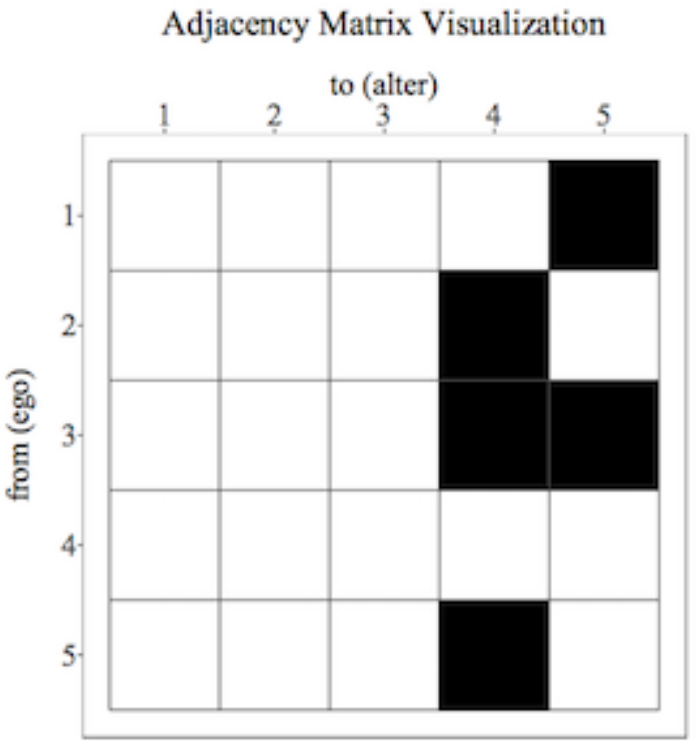
Why care about these relationships?

- **Telephone exchanges**: Nodes are the phone numbers. Edges would indicate a call was made betwen two numbers.
- **Book or movie plots**: Nodes are the characters. Edges would indicate whether they appear together in a scene, or chapter. If they speak to each other, various ways we might measure the association.
- **Social media**: nodes would be the people who post on facebook, including comments. Edges would measure who comments on who's posts.

Drawing these relationships out:

One way to describe these relationships is to provide association matrix between many objects.





(Image created by Sam Tyner.)

Example: Madmen

Source: wikicommons

Generate a network view

- Create a layout (in 2D) which places nodes which are most related close,
- Plot the nodes as points, connect the appropriate lines
- Overlaying other aspects, e.g. gender

introducing madmen data

```
glimpse(madmen)
## List of 2
##  $ edges   :'data.frame':    39 obs. of  2 variables:
##   ..$ Name1: Factor w/ 9 levels "Betty Draper",..: 1 1 2 2 2 2 2 2 2 2 ...
##   ..$ Name2: Factor w/ 39 levels "Abe Drexler",..: 15 31 2 4 5 6 8 9 11 21 ...
##  $ vertices:'data.frame':    45 obs. of  2 variables:
##   ..$ label : Factor w/ 45 levels "Abe Drexler",..: 5 9 16 23 26 32 33 38 39 17 ...
##   ..$ Gender: Factor w/ 2 levels "female","male": 1 2 2 1 2 1 2 2 2 2 ...
```

Nodes and edges?

Network data can be thought of as two related tables, **nodes** and **edges**:

- **nodes** are connection points
- **edges** are the connections between points

Example: Mad Men. (Nodes = characters from the series)

```
madmen_nodes
## # A tibble: 45 x 2
##    label          gender
##    <chr>          <chr>
##  1 Betty Draper   female
##  2 Don Draper     male
##  3 Harry Crane    male
##  4 Joan Holloway  female
##  5 Lane Pryce     male
##  6 Peggy Olson    female
##  7 Pete Campbell  male
##  8 Roger Sterling male
##  9 Sal Romano     male
## 10 Henry Francis  male
## # … with 35 more rows
```

Example: Mad Men. (Edges = how they are associated)

```
madmen_edges
```

```
## # A tibble: 39 x 2
##    Name1         Name2
##    <chr>         <chr>
##  1 Betty Draper Henry Francis
##  2 Betty Draper Random guy
##  3 Don Draper   Allison
##  4 Don Draper   Bethany Van Nuys
##  5 Don Draper   Betty Draper
##  6 Don Draper   Bobbie Barrett
##  7 Don Draper   Candace
##  8 Don Draper   Doris
##  9 Don Draper   Faye Miller
## 10 Don Draper   Joy
## # … with 29 more rows
```

Let's get the madmen data into the right shape

```
madmen_edges %>%
  rename(from_id = Name1, to_id = Name2)
## # A tibble: 39 x 2
##    from_id      to_id
##    <chr>        <chr>
##  1 Betty Draper Henry Francis
##  2 Betty Draper Random guy
##  3 Don Draper   Allison
##  4 Don Draper   Bethany Van Nuys
##  5 Don Draper   Betty Draper
##  6 Don Draper   Bobbie Barrett
##  7 Don Draper   Candace
##  8 Don Draper   Doris
##  9 Don Draper   Faye Miller
## 10 Don Draper   Joy
## # … with 29 more rows
```

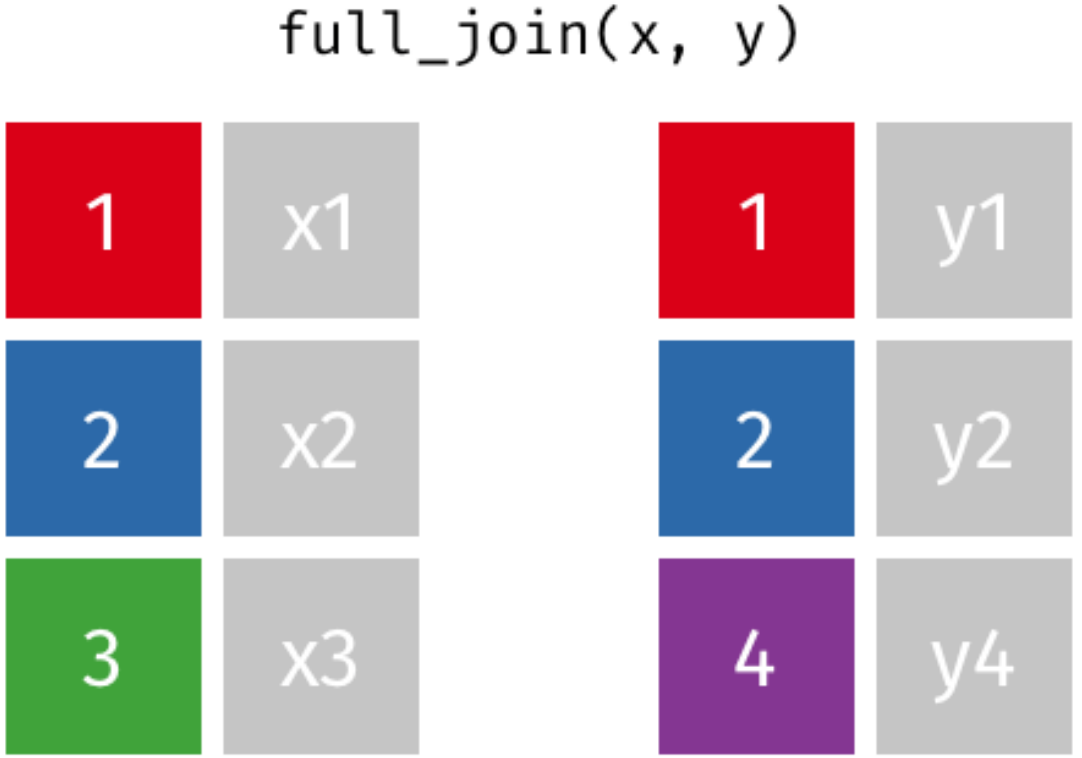
Let's get the madmen data into the right shape

```
madmen_net <- madmen_edges %>%
  rename(from_id = Name1, to_id = Name2) %>%
  full_join(madmen_nodes,
            by = c("from_id" = "label"))
```

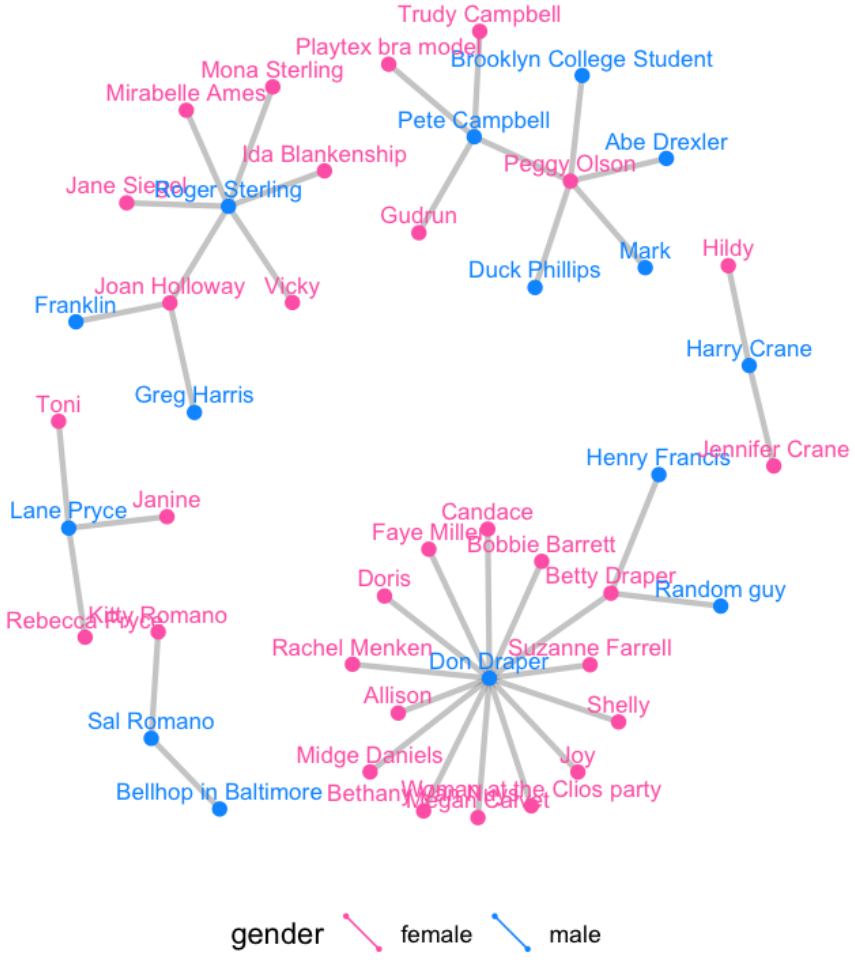
```
madmen_net
```

```
## # A tibble: 75 x 3
##    from_id      to_id            gender
##    <chr>        <chr>            <chr>
##  1 Betty Draper Henry Francis    female
##  2 Betty Draper Random guy       female
##  3 Don Draper   Allison          male
##  4 Don Draper   Bethany Van Nuys male
##  5 Don Draper   Betty Draper     male
##  6 Don Draper   Bobbie Barrett   male
##  7 Don Draper   Candace          male
##  8 Don Draper   Doris            male
##  9 Don Draper   Faye Miller      male
## 10 Don Draper   Joy              male
## # … with 65 more rows
```

Full join?

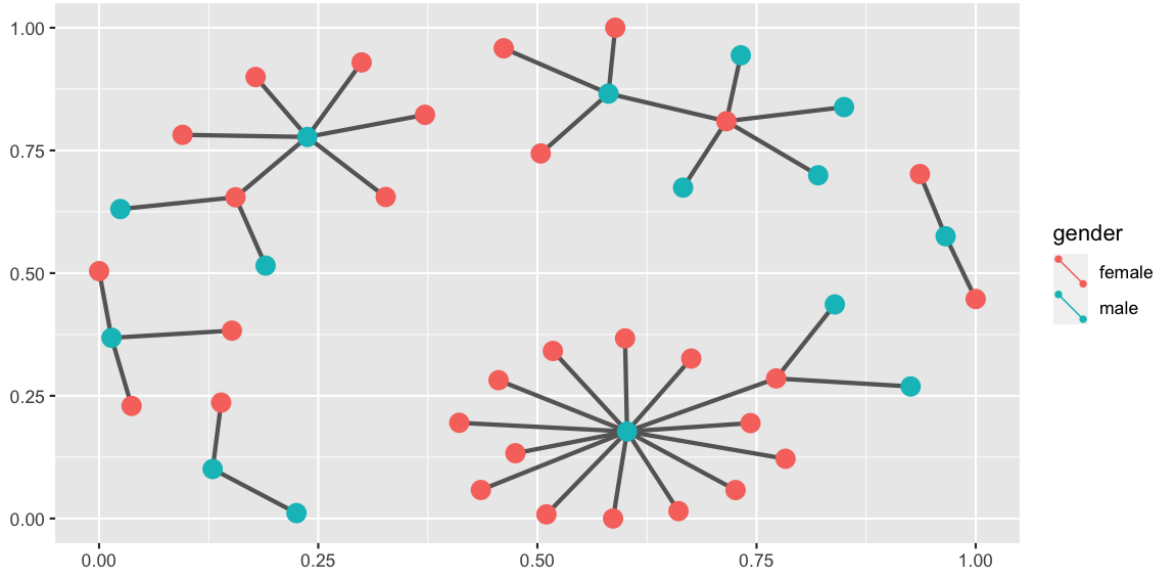


Plotting the data



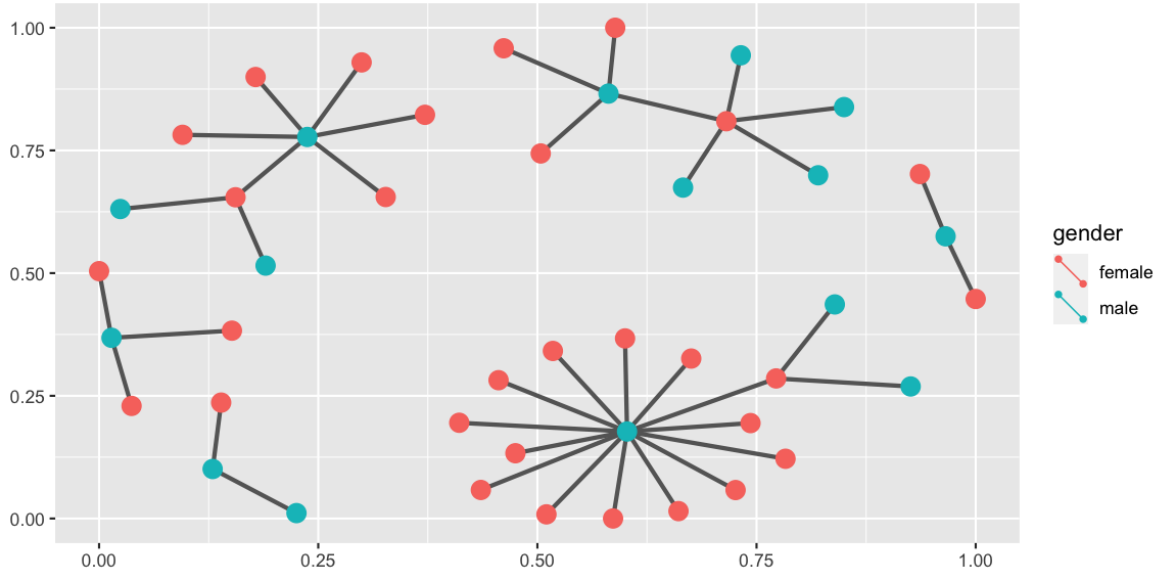
How to plot

```
set.seed(5556677)
ggplot(data = madmen_net,
       aes(from_id = from_id,
           to_id = to_id)) +
  geom_net(aes(colour = gender))
```



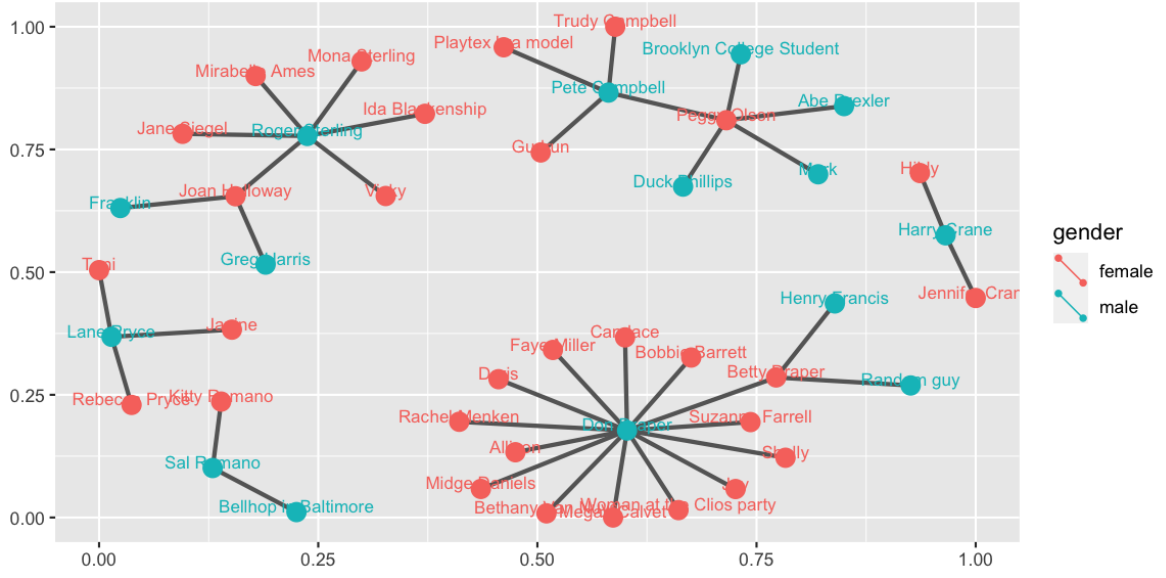
How to plot: specify the layout algorithm

```
set.seed(5556677)
ggplot(data = madmen_net,
       aes(from_id = from_id,
           to_id = to_id)) +
  geom_net(aes(colour = gender),
           layout.alg = "kamadak
```



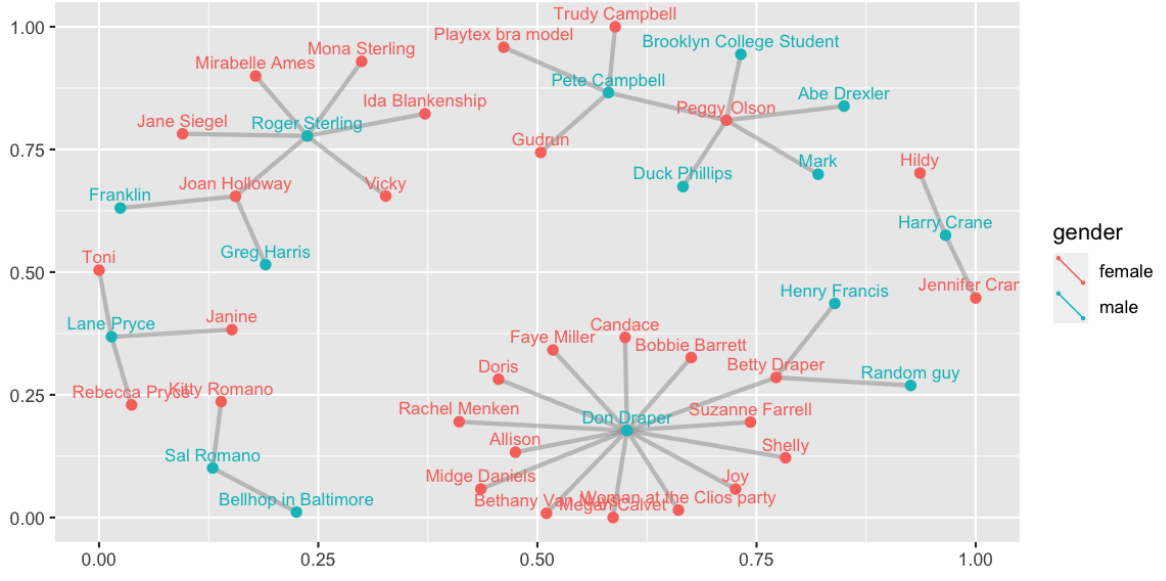
How to plot

```
set.seed(5556677)
ggplot(data = madmen_net,
       aes(from_id = from_id,
           to_id = to_id)) +
  geom_net(aes(colour = gender),
           layout.alg = "kamadak
           directed = FALSE,
           labelon = TRUE,
           fontsize = 3)
```



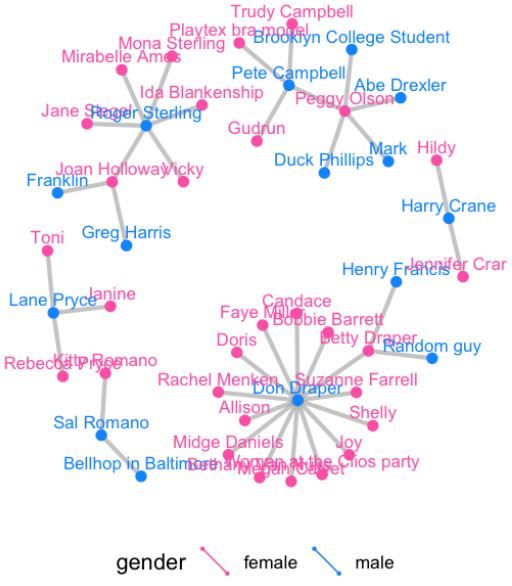
How to plot

```
set.seed(5556677)
ggplot(data = madmen_net,
       aes(from_id = from_id,
           to_id = to_id)) +
  geom_net(aes(colour = gender),
           layout.alg = "kamadak
           directed = FALSE,
           labelon = TRUE,
           fontsize = 3,
           size = 2,
           vjust = -0.6,
           ecolour = "grey60",
           ealpha = 0.5)
```



How to plot

```
set.seed(5556677)
ggplot(data = madmen_net,
       aes(from_id = from_id,
           to_id = to_id)) +
  geom_net(aes(colour = gender),
           layout.alg = "kamadak
           directed = FALSE,
           labelon = TRUE,
           fontsize = 3,
           size = 2,
           vjust = -0.6,
           ecolour = "grey60",
           ealpha = 0.5) +
    scale_colour_manual(values =
    lims(x = c(-0.05, 1.05)) +
    theme_net() +
    theme(legend.position = "bot
```



Which character was most connected?

```
madmen_edges %>%
  pivot_longer(cols = c(Name1, Name2),
               names_to = "List",
               values_to = "Name")
## # A tibble: 78 x 2
##    List  Name
##    <chr> <chr>
##  1 Name1 Betty Draper
##  2 Name2 Henry Francis
##  3 Name1 Betty Draper
##  4 Name2 Random guy
##  5 Name1 Don Draper
##  6 Name2 Allison
##  7 Name1 Don Draper
##  8 Name2 Bethany Van Nuys
##  9 Name1 Don Draper
## 10 Name2 Betty Draper
## # … with 68 more rows
```

Which character was most connected?

```
madmen_edges %>%
  pivot_longer(cols = c(Name1, Name2),
               names_to = "List",
               values_to = "Name") %>%
  count(Name, sort = TRUE)
## # A tibble: 45 x 2
##    Name               n
##    <chr>          <int>
##  1 Don Draper        14
##  2 Roger Sterling     6
##  3 Peggy Olson        5
##  4 Pete Campbell      4
##  5 Betty Draper       3
##  6 Joan Holloway      3
##  7 Lane Pryce         3
##  8 Harry Crane        2
##  9 Sal Romano         2
## 10 Abe Drexler        1
## # … with 35 more rows
```

Example: American college football

Early American football outfits were like Australian AFL today!

Source: wikicommons

Example: American college football

Fall 2000 Season of Division I college football.

- Nodes are the teams, edges are the matches.
- Teams are broken into "conferences" which are the primary competition, but they can play outside this group.

American college football data

Edges:

```
football$edges
##                    from
## 1          BrighamYoung         Fl
## 2                  Iowa          Ka
## 3          BrighamYoung
## 4             NewMexico
## 5           KansasState
## 6                  Iowa
## 7             PennState   Southern
## 8    SouthernCalifornia         Ar
## 9          ArizonaState        San
## 10         BrighamYoung        San
## 11            NewMexico        San
## 12            TexasTech
## 13               Baylor          I
## 14            TexasTech          I
## 15          KansasState          I
## 16    NorthernIllinois         No
```

Nodes:

```
football$vertices
##
## BrighamYoung                Brigha
## FloridaState                Florida
## Iowa
## KansasState                  Kansa
## NewMexico                     New
## TexasTech                     Texa
## PennState                     Penn
## SouthernCalifornia     SouthernCali
## ArizonaState                Arizona
## SanDiegoState              SanDiego
## Baylor                            I
## NorthTexas                    North
## NorthernIllinois       NorthernIl
## Northwestern                Northw
```

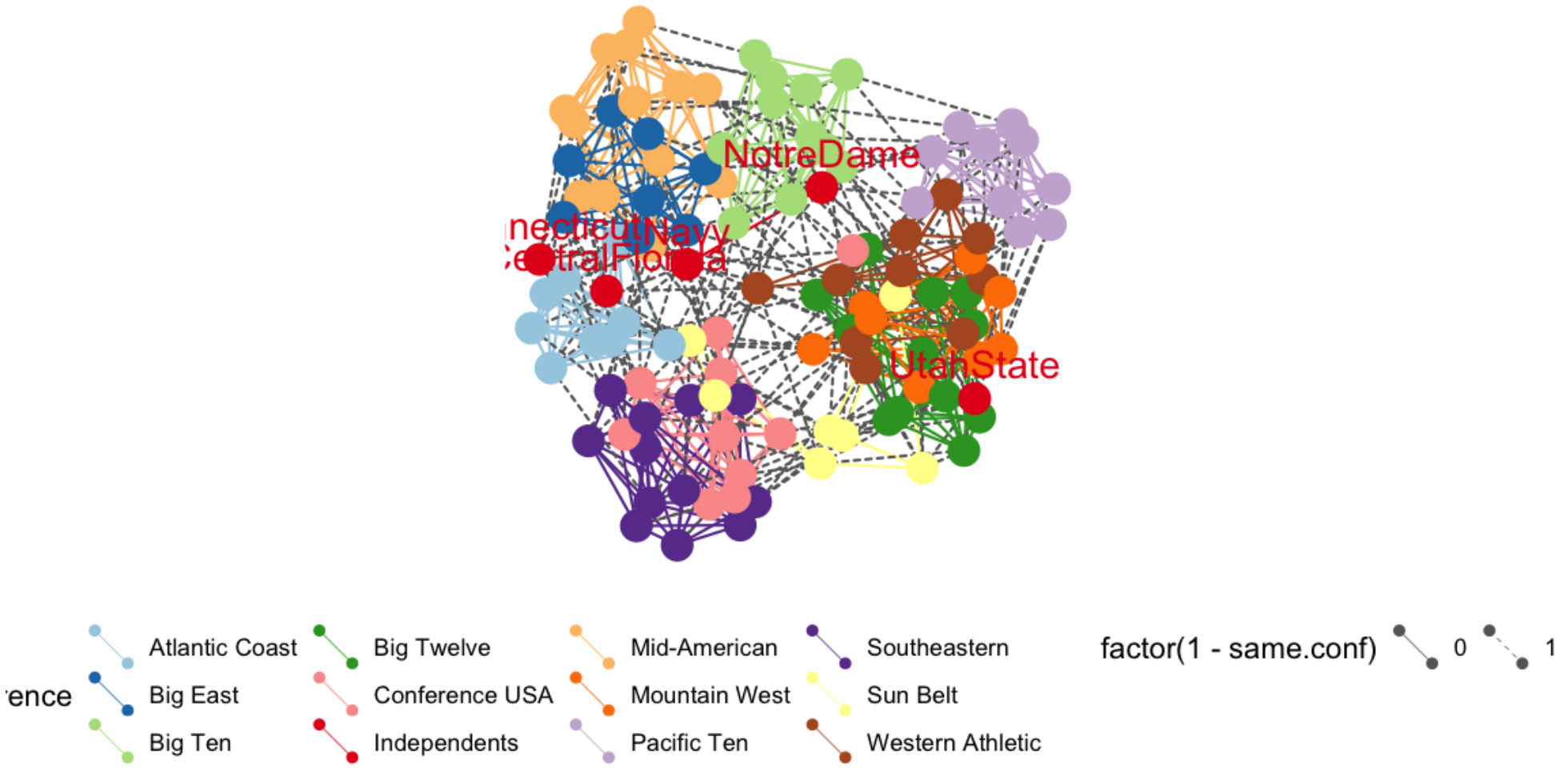
American college football: joining the data

```
# data step: merge vertices and edges
ftnet <- full_join(
  football$edges, football$vertices,
  by = c("from" = "label")
) %>%
  mutate(schools = if_else(value == "Independents", from, ""))

ftnet
##                    from                  to same.conf           value
## 1          BrighamYoung        FloridaState         0    Mountain West
## 2                  Iowa         KansasState         0          Big Ten
## 3          BrighamYoung           NewMexico         1    Mountain West
## 4             NewMexico           TexasTech         0    Mountain West
## 5           KansasState           TexasTech         1       Big Twelve
## 6                  Iowa           PennState         1          Big Ten
## 7             PennState   SouthernCalifornia         0          Big Ten
## 8     SouthernCalifornia        ArizonaState         1      Pacific Ten
## 9          ArizonaState       SanDiegoState         0      Pacific Ten
## 10         BrighamYoung       SanDiegoState         1    Mountain West
## 11            NewMexico       SanDiegoState         1    Mountain West
```

Example: American college football

```
ggplot(data = ftnet,
       aes(from_id = from, to_id = to)) +
  geom_net(
    aes(colour = value,
        group = value,
        linetype = factor(1-same.conf),
        label = schools),
    linewidth = 0.5,
    size = 5,
    vjust = -0.75,
    alpha = 0.3,
    layout.alg = 'fruchtermanreingold'
  ) +
  theme_net() +
  theme(legend.position = "bottom") +
  scale_colour_brewer("Conference", palette = "Paired")
```


NotreDame
UtahState
ence
Atlantic Coast
Big East
Big Ten
Big Twelve
Conference USA
Independents
Mid-American
Mountain West
Pacific Ten
Southeastern
Sun Belt
Western Athletic
factor(1 - same.conf)
0
1

What do we learn?

- Remember layout is done to place nodes that are more similar close together in the display.
- The colours indicate conference the team belongs too. For the most part, conferences are clustered, more similar to each other than other conferences.
- There are some clusters of conference groups, eg Mid-American, Big East, and Atlantic Coast
- The Independents are independent
- Some teams play far afield from their conference.

Example: Harry Potter characters

Source: [wikicommons]

Example: Harry Potter characters

There is a connection between two students if one provides emotional support to the other at some point in the book.

- Code to pull the data together is provided by Sam Tyner here.

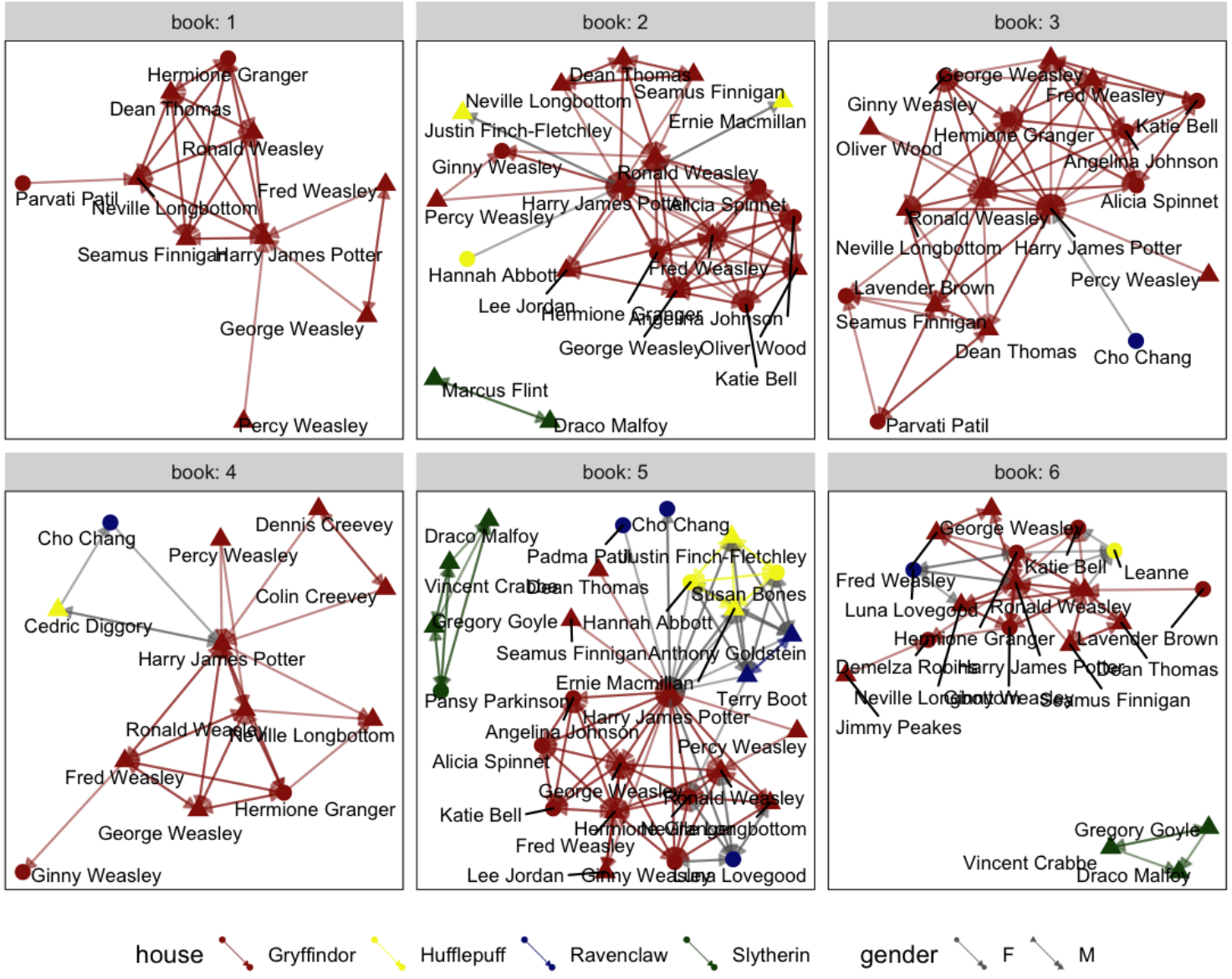
Harry potter data as nodes and edges

```
hp_all
## # A tibble: 720 x 6
##    book  from_id           to_id              schoolyear gender house
##    <chr> <chr>             <chr>                   <dbl> <chr>  <chr>
##  1 1     Dean Thomas       Harry James Potter       1991 M      Gryffindor
##  2 1     Dean Thomas       Hermione Granger         1991 M      Gryffindor
##  3 1     Dean Thomas       Neville Longbottom       1991 M      Gryffindor
##  4 1     Dean Thomas       Ronald Weasley           1991 M      Gryffindor
##  5 1     Dean Thomas       Seamus Finnigan          1991 M      Gryffindor
##  6 1     Fred Weasley      George Weasley           1989 M      Gryffindor
##  7 1     Fred Weasley      Harry James Potter       1989 M      Gryffindor
##  8 1     George Weasley    Fred Weasley             1989 M      Gryffindor
##  9 1     George Weasley    Harry James Potter       1989 M      Gryffindor
## 10 1     Harry James Potter Dean Thomas             1991 M      Gryffindor
## # … with 710 more rows
```

Let's plot the characters

```
ggplot(data = hp_all,
       aes(from_id = from_id,
           to_id = to_id)) +
  geom_net(aes(colour = house, group = house, shape = gender),
           fiteach=T,
           directed = T,
           size = 3,
           linewidth = .5,
           ealpha = .5,
           labelon = T,
           fontsize = 3,
           repel = T,
           labelcolour = "black",
           arrowsize = .5,
           singletons = FALSE) +
  scale_colour_manual(values = c("#941B08","#F1F31C", "#071A80", "#154C07")) +
  facet_wrap(~book, labeller = "label_both", ncol=3) +
  theme_net() +
  theme(panel.background = element_rect(colour = 'black'),
        legend.position="bottom")
```

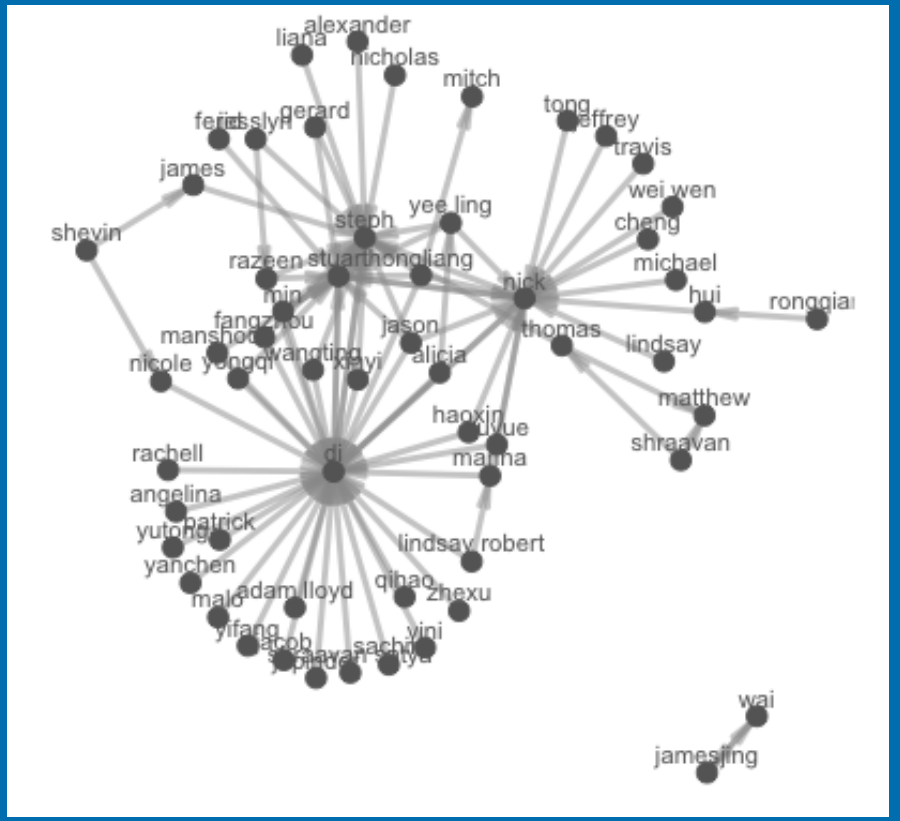
Let's plot the characters



Your turn: lab exercise

- Read in last semesters class data, which contains

  `s1_name` and `s2_name` are the first names of class members, and tutors, with the latter being the "go-to" person for the former.
- Write the code to produce a class network that looks something like below

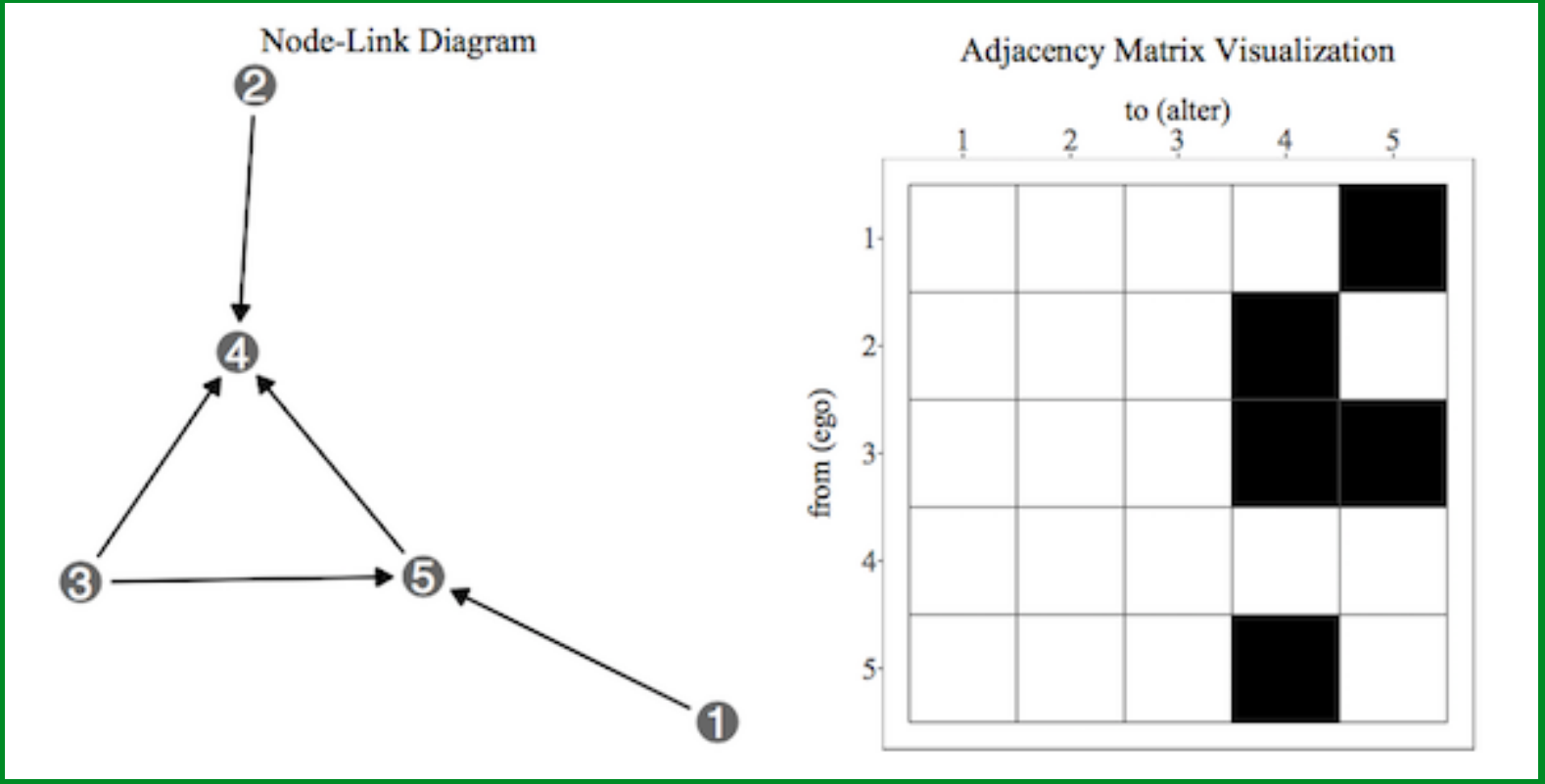


Summary

- To make a network analysis, you need:
  - an association matrix, that describes how nodes (vertices) are connected to each other
  - a layout algorithm to place the nodes optimally so that the fewest edges cross, or that the nodes that are most closely associated are near to each other.

Quantitative association matrices

Previous association matrices were black and white:



Quantitative association matrices

- You could have the association between nodes described as real numbers.
- E.g., these are the number of times that these people called each other in the last week:

	Meg	Tay	Yat	Zili	Jess
Meg	0	5	4	1	1
Tay	5	0	4	2	1
Yat	4	4	0	0	0
Zili	1	2	0	0	6
Jess	1	1	0	6	0

Quantitative association matrices

We would need to turn this into an edge data set:

```
## # A tibble: 25 x 3
##    from  to    count
##    <chr> <chr> <dbl>
##  1 Meg   Meg       0
##  2 Tay   Meg       5
##  3 Yat   Meg       4
##  4 Zili  Meg       1
##  5 Jess  Meg       1
##  6 Meg   Tay       5
##  7 Tay   Tay       0
##  8 Yat   Tay       4
##  9 Zili  Tay       2
## 10 Jess  Tay       1
## # … with 15 more rows
```

Quantitative association matrices

- We need to decide what corresponds to a "connection".
- Let's say they need to have called each other at least 4 times, to be considered connected.

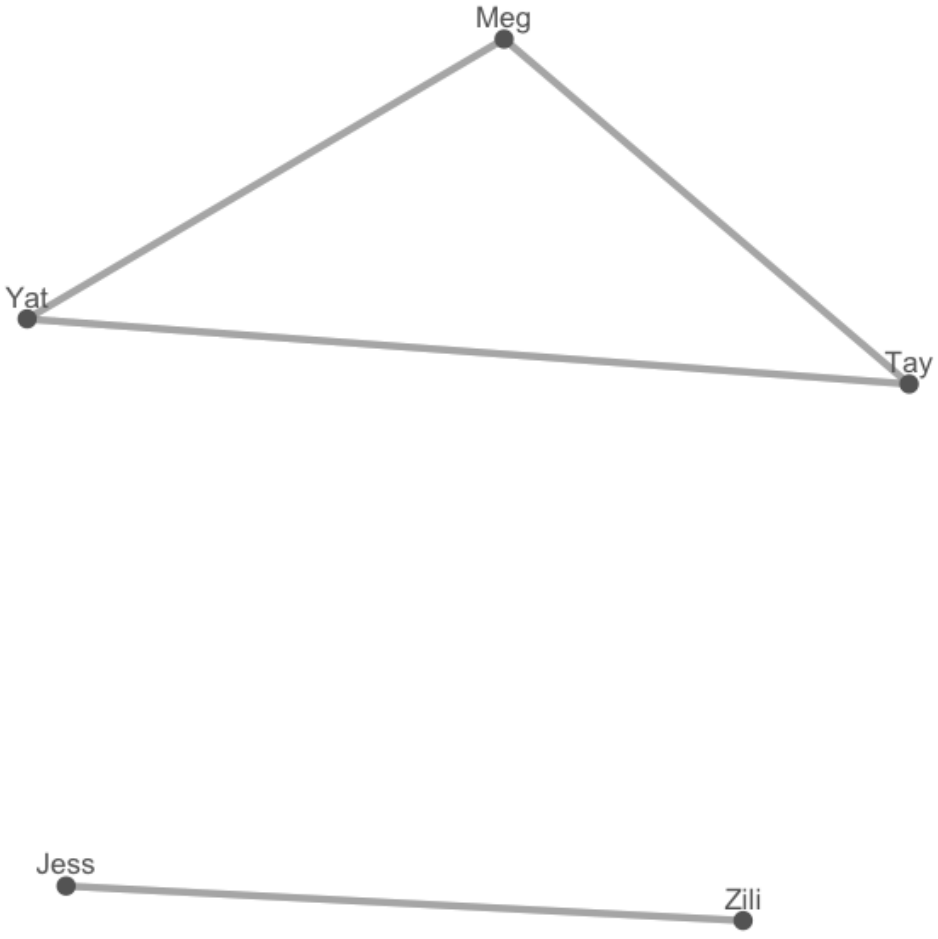
```
d_edges_filter <- d_edges %>% filter(count > 3)

d_edges_filter
## # A tibble: 8 x 3
##   from  to    count
##   <chr> <chr> <dbl>
## 1 Tay   Meg       5
## 2 Yat   Meg       4
## 3 Meg   Tay       5
## 4 Yat   Tay       4
## 5 Meg   Yat       4
## 6 Tay   Yat       4
## 7 Jess  Zili      6
## 8 Zili  Jess      6
```

Association matrices: Make the network diagram.

```
library(geomnet)
set.seed(2019-10-09)
ggplot(data = d_edges_filter,
       aes(
         from_id = from,
         to_id = to)) +
  geom_net(
    layout.alg = "kamadakawai",
    size = 2,
    labelon = TRUE,
    vjust = -0.6,
    ecolour = "grey60",
    directed =FALSE,
    fontsize = 3,
    ealpha = 0.5
    ) +
    theme_net()
```

Association matrices: Make the network diagram.



Data: Last 2018, 4 months of currency USD cross-rates

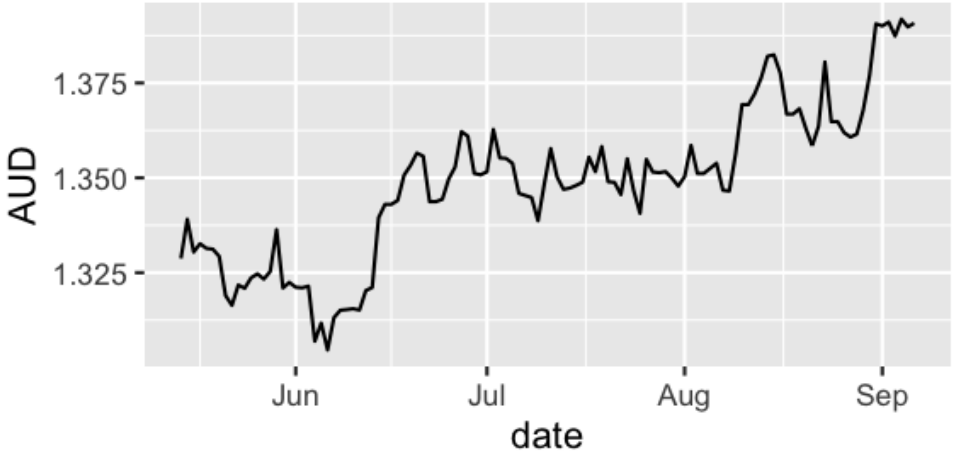
SO let's try this with cross-currency rates across the globe!

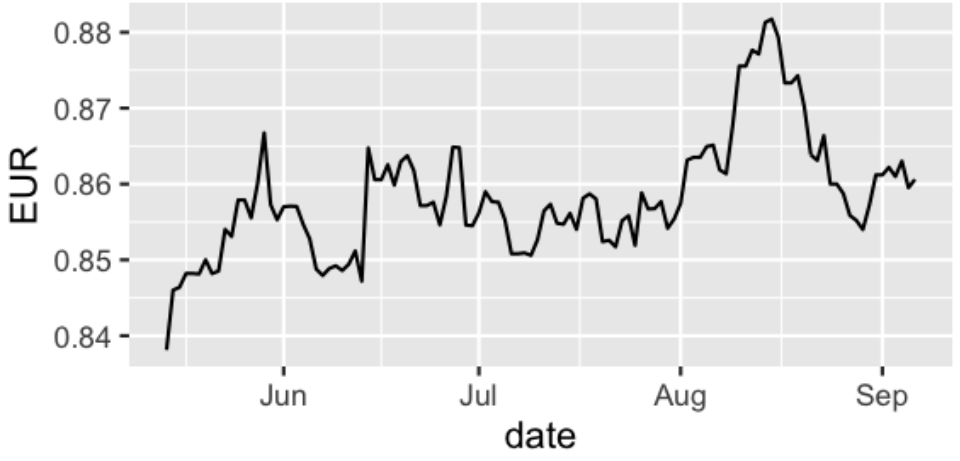
- Data extracted from http://openexchangerates.org/api/historical
- R packages `jsonlite`, processed with `tidyverse`, `lubridate`

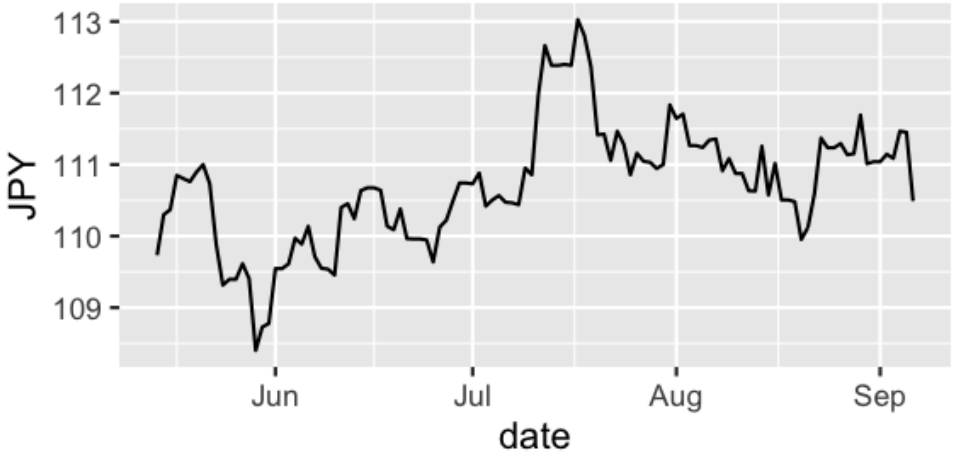
Data: Last 4 months of currency USD cross-rates

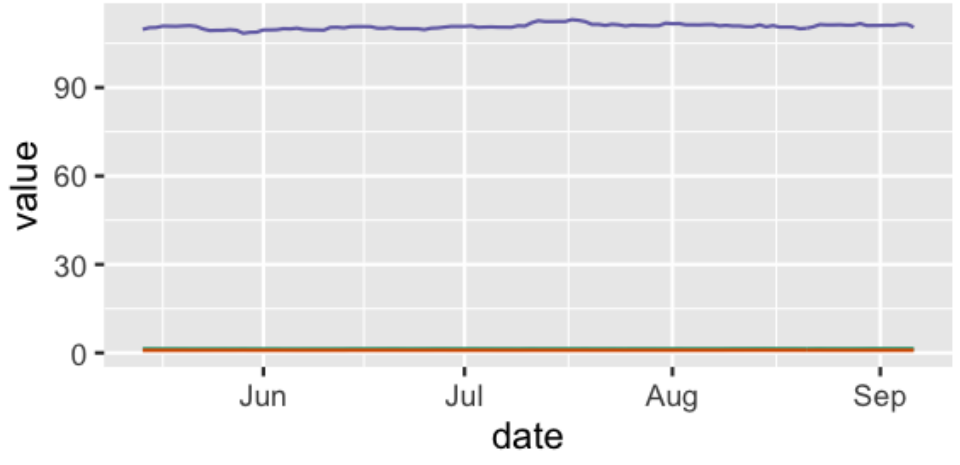
```
## # A tibble: 6 x 171
##   date         AED   AFN   ALL   AMD   ANG   AOA   ARS   AUD   AWG   AZN   BAM
##   <date>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2018-05-14  3.67  71.2  106.  485.  1.79  230.  25.0  1.33  1.79  1.70  1.63
## 2 2018-05-15  3.67  71.2  107.  485.  1.80  230.  24.1  1.34  1.79  1.70  1.64
## 3 2018-05-16  3.67  71.0  108.  484.  1.80  232.  24.3  1.33  1.79  1.70  1.66
## 4 2018-05-17  3.67  71.0  108.  483.  1.80  233.  24.3  1.33  1.79  1.70  1.66
## 5 2018-05-18  3.67  71.0  108.  483.  1.80  233.  24.4  1.33  1.79  1.70  1.66
## 6 2018-05-19  3.67  70.9  108.  482.  1.79  233.  24.4  1.33  1.79  1.70  1.66
## # … with 159 more variables: BBD <dbl>, BDT <dbl>, BGN <dbl>, BHD <dbl>,
## #   BIF <dbl>, BMD <dbl>, BND <dbl>, BOB <dbl>, BRL <dbl>, BSD <dbl>,
## #   BTC <dbl>, BTN <dbl>, BWP <dbl>, BYN <dbl>, BZD <dbl>, CAD <dbl>,
## #   CDF <dbl>, CHF <dbl>, CLF <dbl>, CLP <dbl>, CNH <dbl>, CNY <dbl>,
## #   COP <dbl>, CRC <dbl>, CUC <dbl>, CUP <dbl>, CVE <dbl>, CZK <dbl>,
## #   DJF <dbl>, DKK <dbl>, DOP <dbl>, DZD <dbl>, EGP <dbl>, ERN <dbl>,
## #   ETB <dbl>, EUR <dbl>, FJD <dbl>, FKP <dbl>, GBP <dbl>, GEL <dbl>,
## #   GGP <dbl>, GHS <dbl>, GIP <dbl>, GMD <dbl>, GNF <dbl>, GTQ <dbl>,
## #   GYD <dbl>, HKD <dbl>, HNL <dbl>, HRK <dbl>, HTG <dbl>, HUF <dbl>,
## #   IDR <dbl>, ILS <dbl>, IMP <dbl>, INR <dbl>, IQD <dbl>, IRR <dbl>,
## #   ISK <dbl>, JEP <dbl>, JMD <dbl>, JOD <dbl>, JPY <dbl>, KES <dbl>,
```

Data: Last 4 months of currency USD cross-rates





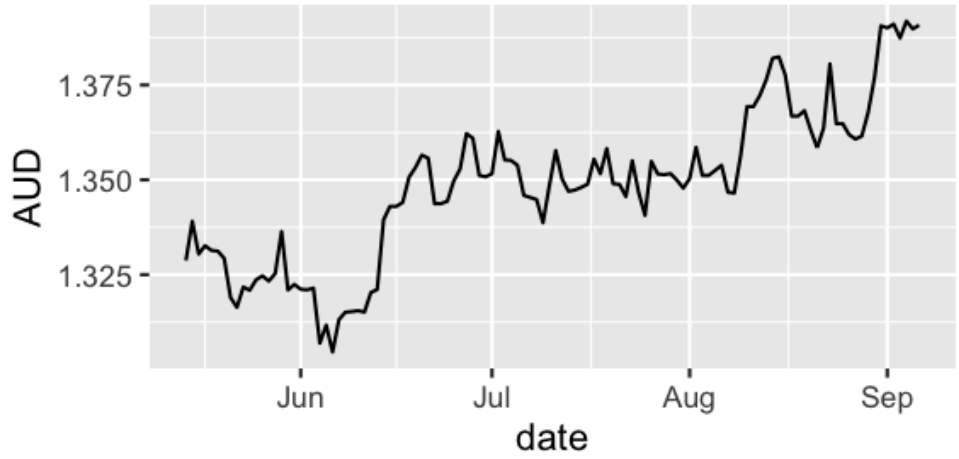


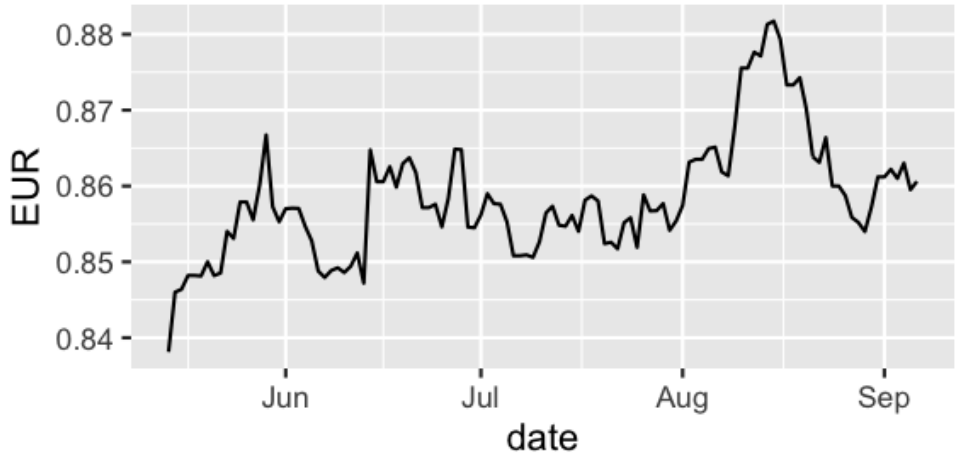


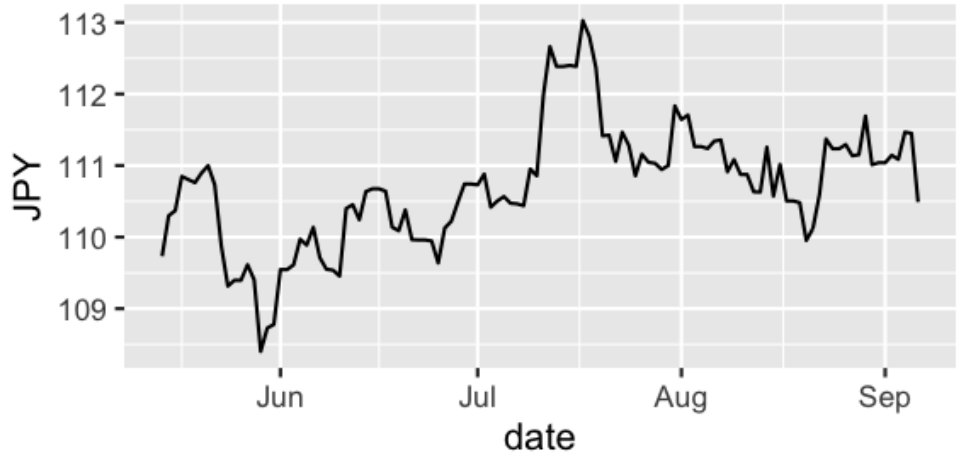
Your turn: Rstudio

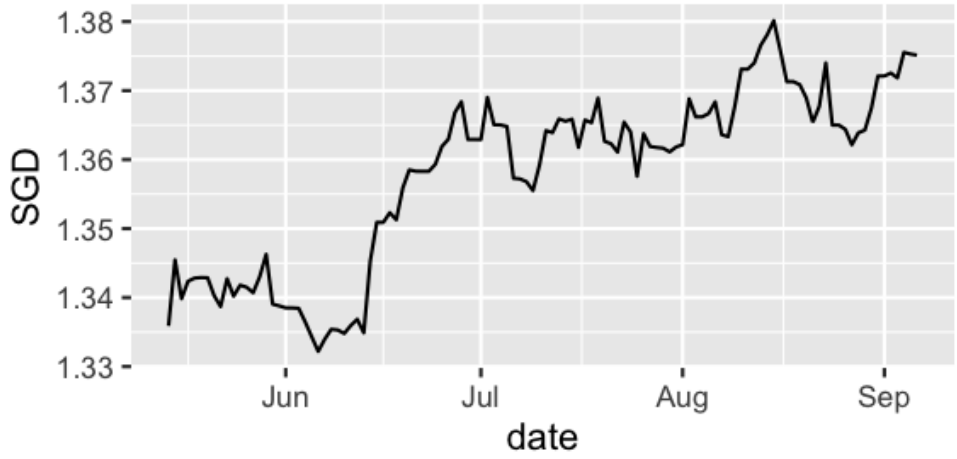
Make some plots (or google) to answer these questions

- Is the NZD more similar to AUD, EUR, or JPY? (What currency is NZD?)
- Is SGD more similar to AUD, EUR, or JPY? (What currency is SGD?)
- How many currencies are there in the British Isles?


AUD
1.375
1.350
1.325
Jun
Jul
Aug
Sep
date


EUR
0.88
0.87
0.86
0.85
0.84
Jun
Jul
Aug
Sep
date


JPY
113
112
111
110
109
Jun
Jul
Aug
Sep
date


SGD
1.38
1.37
1.36
1.35
1.34
1.33
Jun
Jul
Aug
Sep
date

Pre-processing: Keep currencies that change

- Some currencies don't change very much.
- These should be filtered from the analysis, because in a study of currency movement, if it doesn't move then there is nothing more to be said.

Pre-processing: Keep currencies that change

- To filter out these currencies we use a statistic called coefficient of variation:

$$CoefVariation = \frac{\sigma}{\mu}$$

- Measures standard deviation of currency relative to the mean.
- For high means, we expect a currency to change more.
- That is, relatively the standard deviation would be larger to consider it to be changing.

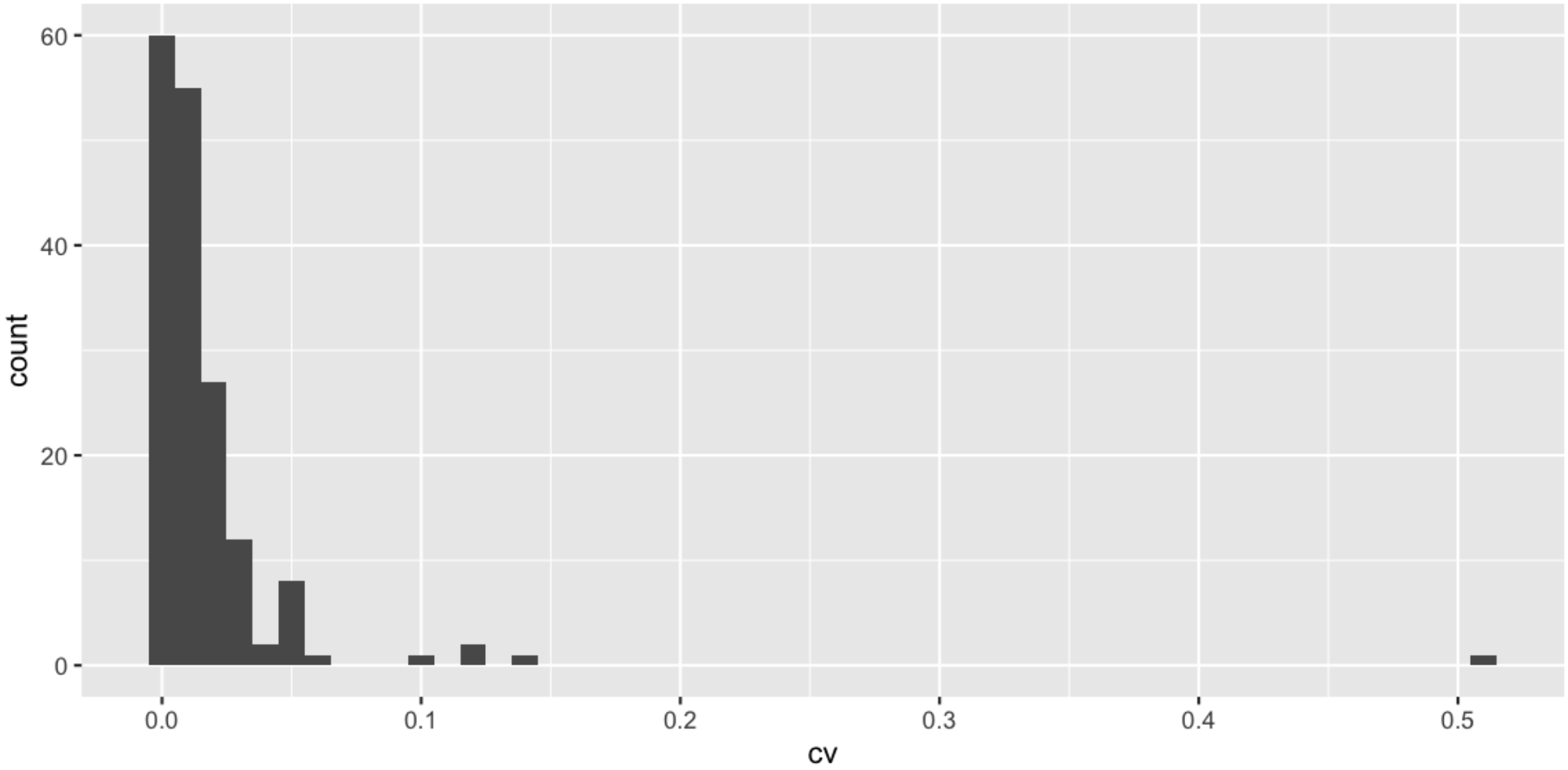
Computing CV

Stratgey pivot to long form then group and summarize currency values

```
# Compute coefficient of variation. We will only analyse
# currencies that have changes substantially over this time.
cv <- function(x){
  sd(x)/mean(x)
}

rates_cv <- rates %>%
  pivot_longer(cols = -date, names_to = "currency") %>%
  group_by(currency) %>%
  summarise(cv = cv(value))
```

Distrubtion of CV values



Filter out currencies if they have CVs below the first quantile

```
rates_stable <- rates_cv %>%
  filter(cv < quantile(cv, 0.25))
```

Filter out low cv currencies using pivot and an anti join

```
rates_sub <- rates %>%
  pivot_longer(cols = -date, names_to = "currency") %>%
  anti_join(rates_stable)
rates_sub
## # A tibble: 14,732 x 3
##    date       currency   value
##    <date>     <chr>      <dbl>
##  1 2018-05-14 AFN         71.2
##  2 2018-05-14 ALL        106.
##  3 2018-05-14 ANG          1.79
##  4 2018-05-14 AOA        230.
##  5 2018-05-14 ARS         25.0
##  6 2018-05-14 AUD          1.33
##  7 2018-05-14 BAM          1.63
##  8 2018-05-14 BDT         84.7
##  9 2018-05-14 BGN          1.64
## 10 2018-05-14 BIF       1767.
## # … with 14,722 more rows
```

Remove currencies that are not currencies

Some of the currencies ... aren't really currencies. Google these ones: XAG, XDR, XPT - what are they?

Remove currencies that are not currencies

```
# Remove non-currencies
rates_dropped <- rates_sub %>%
  filter(!currency %in% c("ALL", "XAG", "XDR", "XPT"))
```

XAG is Gold XPT is Platinum XDR is special drawing rights

Standardize the currencies

To examine overall trend regardless of actual USD cross rate, standardise the values to have mean 0 and standard deviation 1.

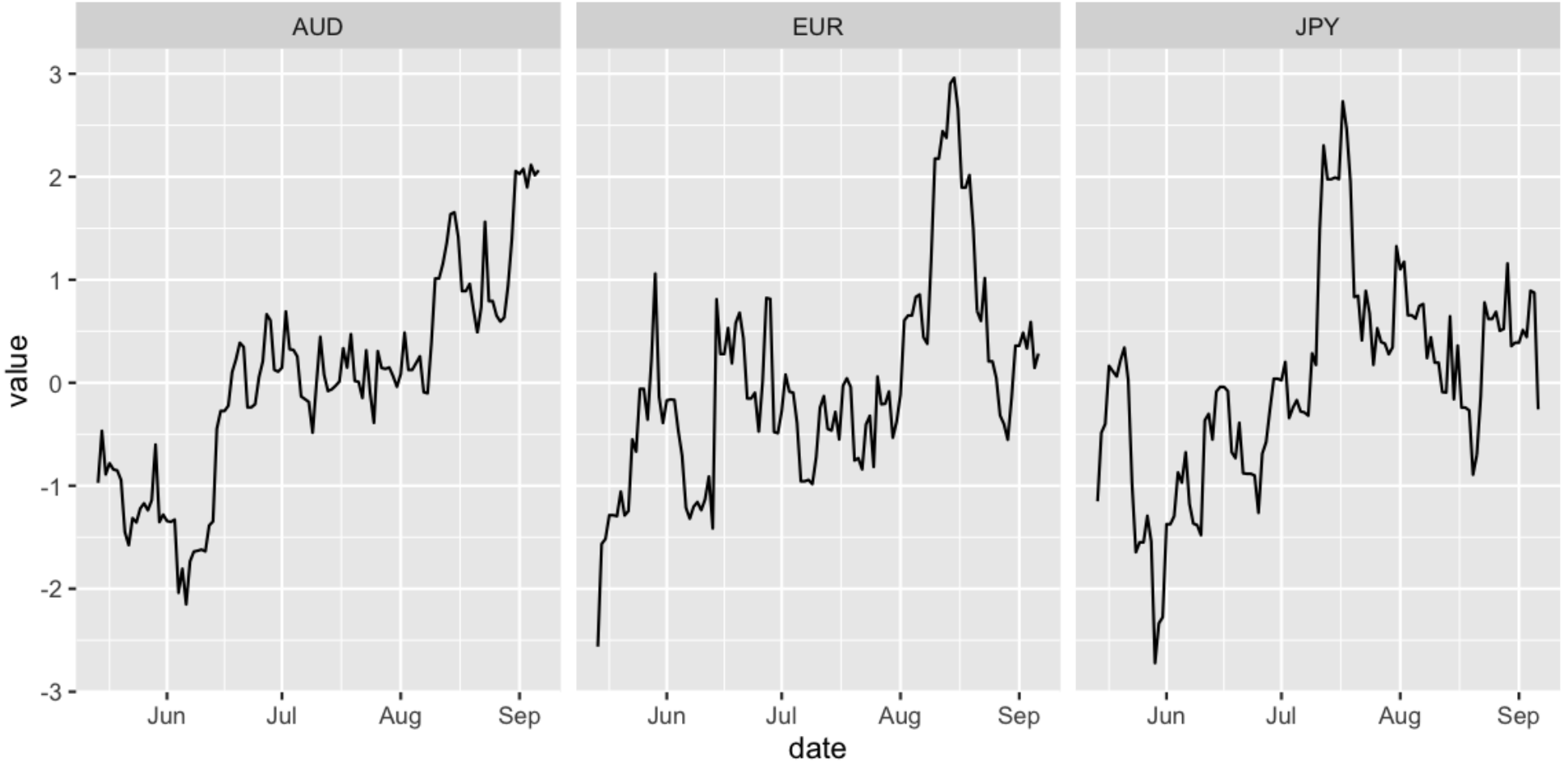
```
scale01 <- function(x) (x - mean(x)) / sd(x)
```

Rescale all values to have standardised values

Use `group_by()` plus `mutate()`!

```
rates_scaled <- rates_dropped %>%
  group_by(currency) %>%
  mutate(value = scale01(value))
```

Standardize the currencies



Compute distances between all pairs of currencies

Euclidean distance is used to compute similarity between all pairs of currencies.

$$d_{ij} = \sqrt{\sum_{i=1}^{t} (C_{1i} - C_{2i})^2}$$

Compute distances between all pairs of currencies

We need to put our data back in wide form! And then turn it into a matrix.

```
rates_wide <- rates_scaled %>%
    pivot_wider(id_cols = "date", names_from = "currency") %>%
    select(-date)

# compute distance between currencies, rows <--> columns
rates_wide_t <- t(rates_wide)
```

Use built in function to compute distance

```
currency_dist <- as.matrix(dist(rates_wide_t,
                                diag = TRUE,
                                upper = TRUE))

currency_dist[1:5, 1:5]
##           AFN      ANG      AOA      ARS      AUD
## AFN 0.000000 8.044527 7.315939 8.014165 7.970993
## ANG 8.044527 0.000000 5.628321 9.601101 7.277124
## AOA 7.315939 5.628321 0.000000 5.760894 5.299254
## ARS 8.014165 9.601101 5.760894 0.000000 5.983452
## AUD 7.970993 7.277124 5.299254 5.983452 0.000000
```

A note on distance matrices:

- A distance matrix is the inverse of an association matrix.
- A distance matrix close to 0 means the pair are most similar.
- For an association matrix far from zero means the pair are close.
- Either can be used to generate a network.

Create network: Pivot data into long form, filter based on similarity

Here only the pairs of currencies who are closer than "4" to each other are kept.

```
distance_tbl <-  currency_dist %>%
  as.data.frame() %>%
  rownames_to_column(var = "from_currency") %>%
  pivot_longer(-from_currency,
               names_to = "to_currency",
               values_to = "distance") %>%
  filter(distance < 4 ) %>%
  filter(from_currency != to_currency)
```

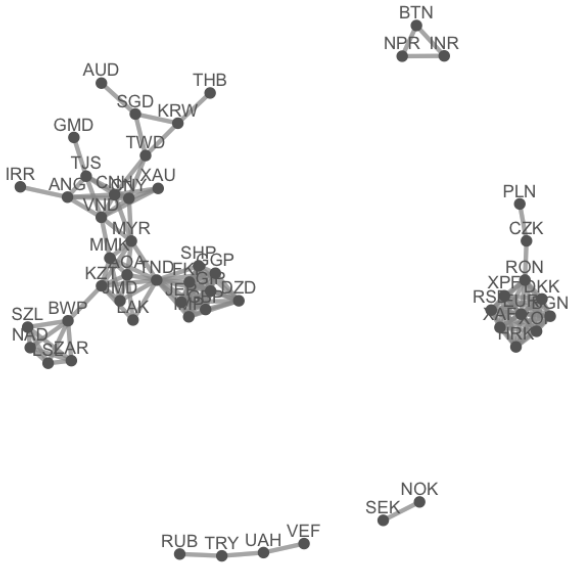
Create network: Gather data into long form, filter based on similarity

Here only the pairs of currencies who are closer than "4" to each other are kept.

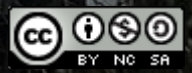
```
distance_tbl
## # A tibble: 266 x 3
##    from_currency to_currency distance
##    <chr>         <chr>           <dbl>
##  1 ANG           CNH              2.98
##  2 ANG           CNY              3.24
##  3 ANG           IRR              3.73
##  4 ANG           TJS              3.60
##  5 ANG           VND              3.42
##  6 AOA           JMD              3.66
##  7 AOA           KZT              2.11
##  8 AOA           LAK              3.55
##  9 AOA           MMK              2.19
## 10 AOA           MYR              2.17
## # … with 256 more rows
```

Network laid out

```
# Make network
library(geomnet)
set.seed(10052016)
ggplot(data = distance_tbl,
       aes(
         from_id = from_currency
         to_id = to_currency
         )) +
  geom_net(
    layout.alg = "kamadakawai",
    size = 2,
    labelon = TRUE,
    vjust = -0.6,
    ecolour = "grey60",
    directed = FALSE,
    fontsize = 3,
    ealpha = 0.5
    ) +
    theme_net() +
    theme(
```



That's it!



Lecturer: Nicholas Tierney & Stuart Lee

Department of Econometrics and Business Statistics

ETC5510.Clayton-x@monash.edu

May 2020