# Zwiller_Lab_3

Thomas Zwiller

2025-05-08

Library Import

```r
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.4.1
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```r
library(ggraph)
```

```
## Loading required package: ggplot2
```

```r
library(ggplot2)
library(gplots)
```

```
## Warning: package 'gplots' was built under R version 4.4.1
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(ggnetwork)
```

```
nodes <- read.csv('/Users/TomTheIntern/Desktop/Mendoza/Mod 4/Networks/Lab 2/nodelist.csv')
summary(nodes)
```

```
##        ID            Name                Age           Gender
##  Min.   : 1.00   Length:12          Min.   :21.00   Length:12
##  1st Qu.: 3.75   Class :character   1st Qu.:23.00   Class :character
##  Median : 6.50   Mode  :character   Median :36.50   Mode  :character
##  Mean   : 6.50                      Mean   :38.00
##  3rd Qu.: 9.25                      3rd Qu.:45.75
##  Max.   :12.00                      Max.   :65.00
```

```
edges <- read.csv('/Users/TomTheIntern/Desktop/Mendoza/Mod 4/Networks/Lab 2/edgelist.csv')
summary(edges)
```

```
##     ego_num         alter_num          ego               alter
##  Min.   : 1.000   Min.   : 1.000   Length:40          Length:40
##  1st Qu.: 2.750   1st Qu.: 2.750   Class :character   Class :character
##  Median : 5.000   Median : 5.000   Mode  :character   Mode  :character
##  Mean   : 5.575   Mean   : 5.575
##  3rd Qu.: 9.000   3rd Qu.: 9.000
##  Max.   :12.000   Max.   :12.000
##      type              strength
##  Length:40          Min.   :1.00
##  Class :character   1st Qu.:2.00
##  Mode  :character   Median :4.00
##                     Mean   :3.45
##                     3rd Qu.:4.25
##                     Max.   :5.00
```

```
# make network object####
names(edges)
```

```
## [1] "ego_num"   "alter_num" "ego"       "alter"     "type"      "strength"
```

```
names(nodes)
```

```
## [1] "ID"     "Name"   "Age"    "Gender"
```

```
# you might need the first column in the datasets to have the same name
net <- igraph::graph_from_data_frame(edges, directed = T, vertices = nodes) # set the network to direct
#graph_from_data_frame is how we turn two separate datasets into a single network in R
net # column for ID is now called "name"
```

```
## IGRAPH 6c056ba DN-- 12 40 --
## + attr: name (v/c), Name (v/c), Age (v/n), Gender (v/c), ego (e/c),
## | alter (e/c), type (e/c), strength (e/n)
## + edges from 6c056ba (vertex names):
##  [1] 1 ->2  2 ->1  1 ->5  5 ->1  1 ->3  3 ->1  1 ->4  4 ->1  1 ->6  6 ->1 
## [11] 1 ->7  7 ->1  1 ->10 10->1  2 ->4  4 ->2  2 ->3  3 ->2  4 ->3  3 ->4 
## [21] 4 ->6  6 ->4  4 ->5  5 ->4  3 ->5  5 ->3  10->9  9 ->10 10->11 11->10
## [31] 10->12 12->10 9 ->12 12->9  11->12 12->11 6 ->7  7 ->6  7 ->8  8 ->7
```

```
is_simple(net) #does the network have loops or multiplex ties?
```

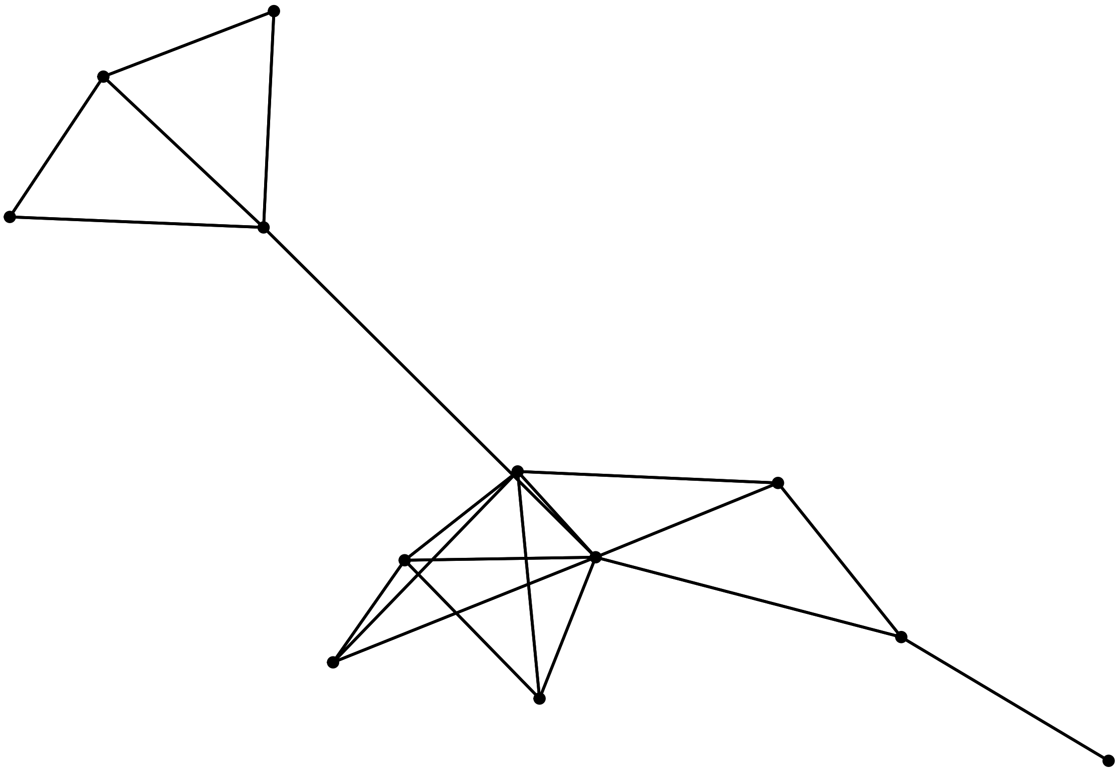
```
## [1] TRUE
```

```
is_directed(net) #does the network have directed ties?
```

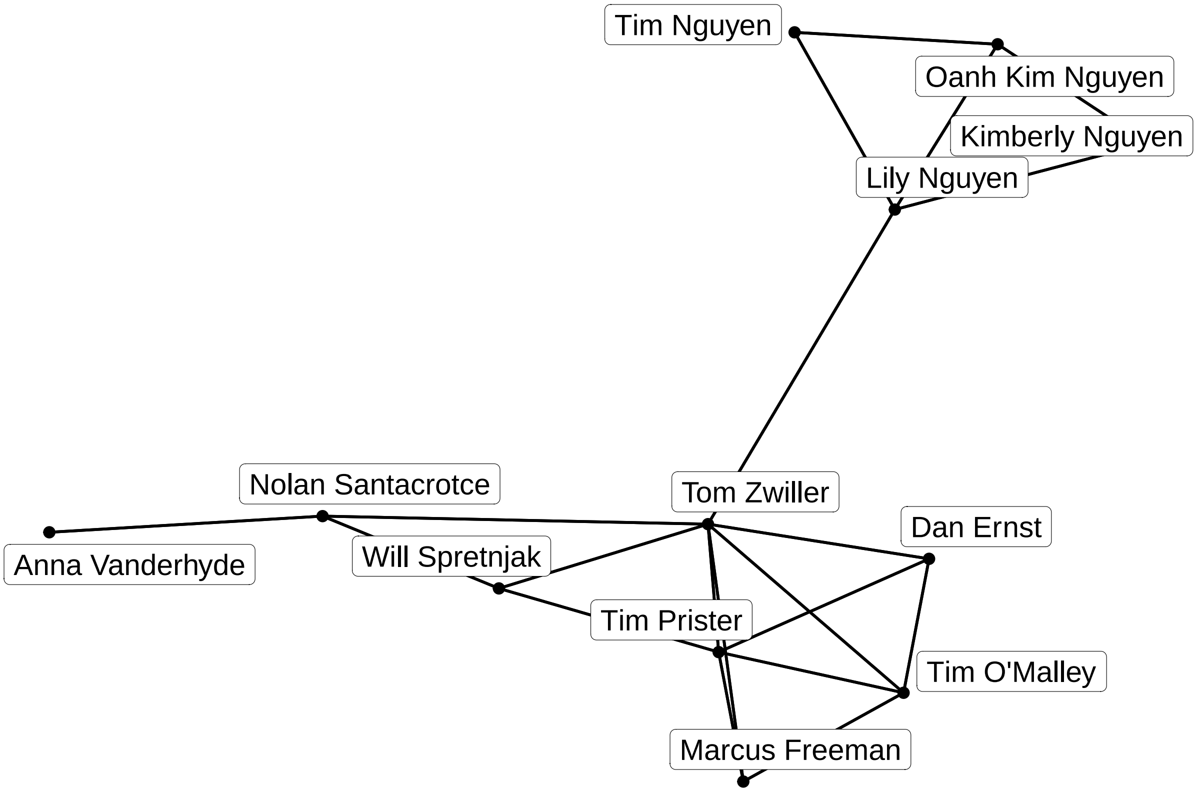
```
## [1] TRUE
```

```
#### Building a visualization####
# some layout options for ggraph:
# "stress", "star", "circle", "gem", "dh", "graphopt",
# "grid", "mds", "randomly", "fr", "kk", "drl", "lgl"
```

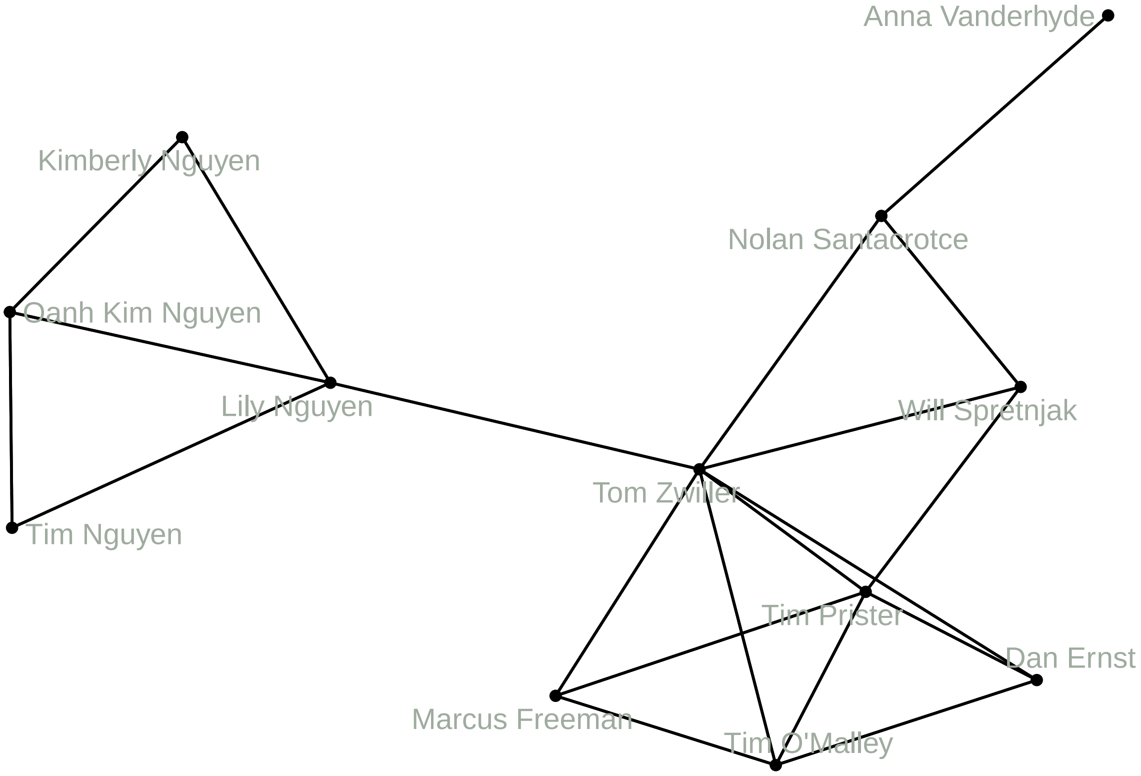
```
# 0 - default####
ggraph(net, layout = "graphopt") +
  geom_edge_link() +
  geom_node_point() +
  ggnetwork::theme_blank() #I like to add this to make the background plain white rather than grey
```



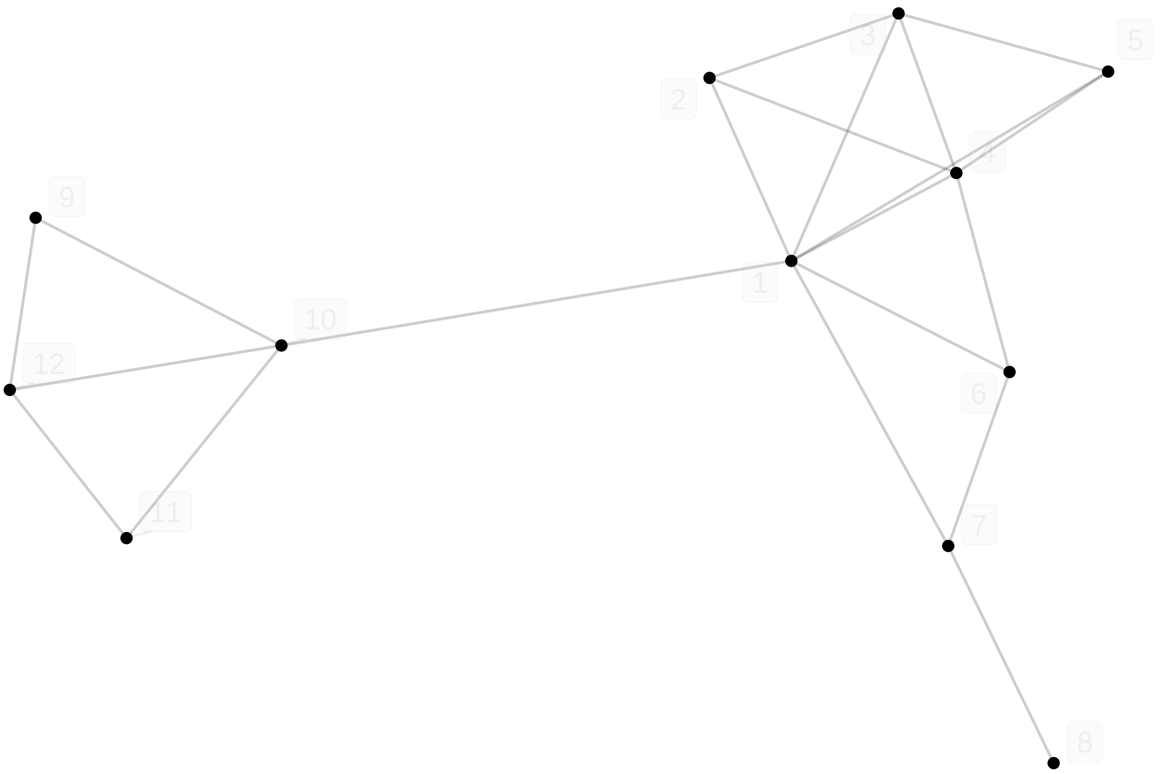
```
# 1 - node labels####
ggraph(net, layout = "graphopt") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_label(aes(label = Name), repel = T, size = 4, color = "black", fill = "white", label.size =
  # NOTE: When you create the network object,
  # repel offsets the text from the node
  # size is the size of the label
  # color is the color of the label, box outline, and line connecting label to node
  # fill is the color of the inside of the box
  # label.size is the thickness of the box outline
  ggnetwork::theme_blank()
```



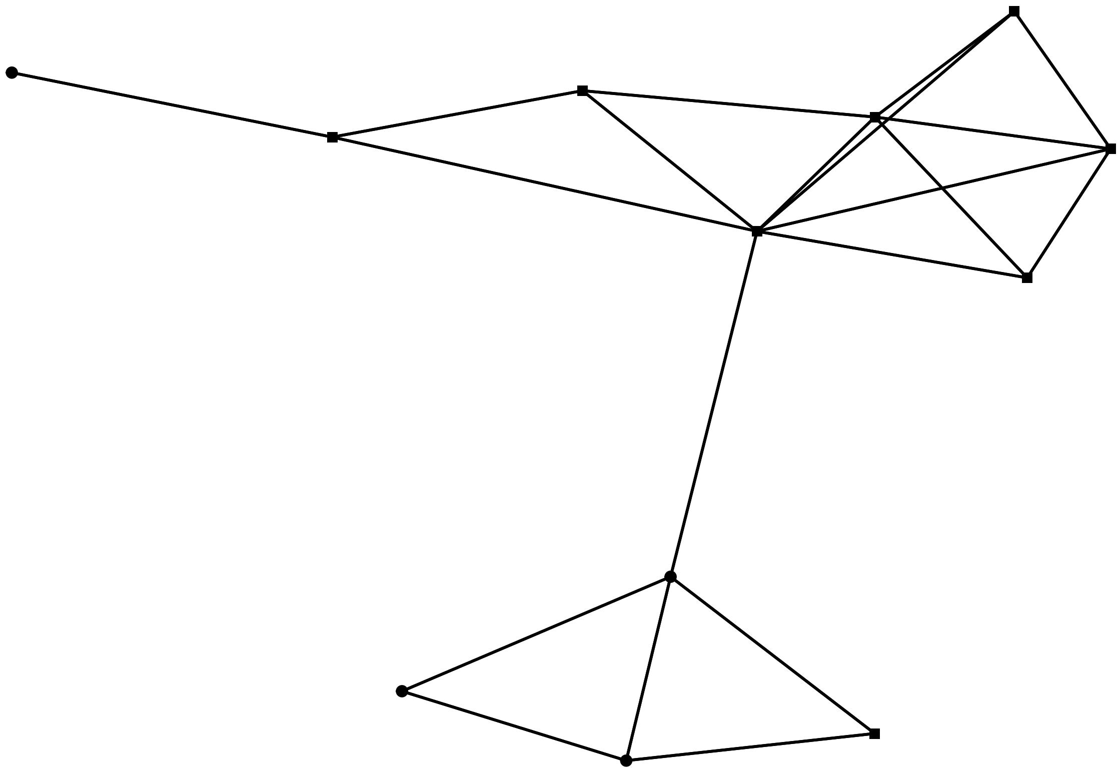
```
ggraph(net, layout = "stress") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label = Name), repel = T, size = 4, color = "#9fab9f") + #plain text
  ggnetwork::theme_blank()
```



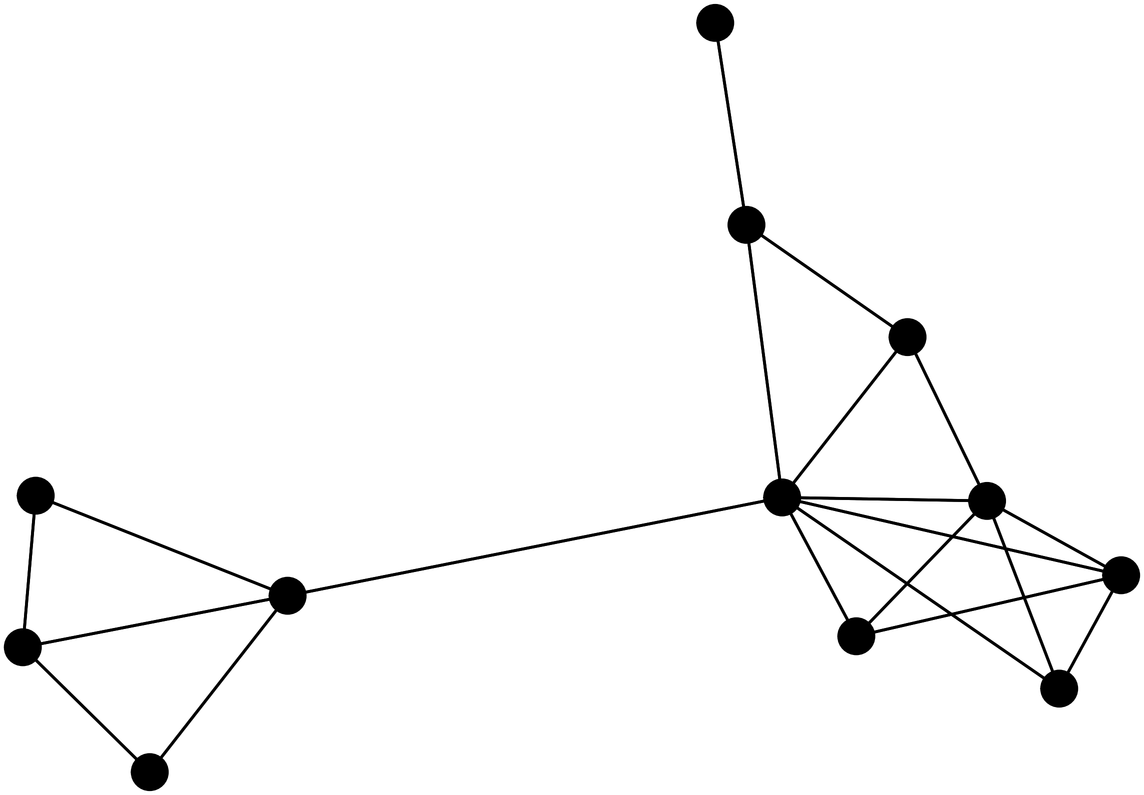
```
# 2 - transparency####
ggraph(net, layout = "graphopt") +
  geom_edge_link(alpha = 0.1) + #alpha is opacity (0.1 means 10% opaque)
  geom_node_point() +
  geom_node_label(aes(label = name), repel = T, label.size = 0.1, alpha = 0.1, color = "grey50", fill =
  ggnetwork::theme_blank()
```



```
# 3 - node shape, size, color####
ggraph(net, layout = "graphopt") +
  geom_edge_link() +
  geom_node_point(shape = ifelse(V(net)$Gender == "F", "circle", "square")) +
  #shapes can also be numeric, as shown in this chart: http://www.sthda.com/english/wiki/ggplot2-point-
  ggnetwork::theme_blank()
```



```
ggraph(net, layout = "graphopt") +
  geom_edge_link() +
  geom_node_point(size = 6) + #can be numeric, or based on a network variable
  ggnetwork::theme_blank()
```



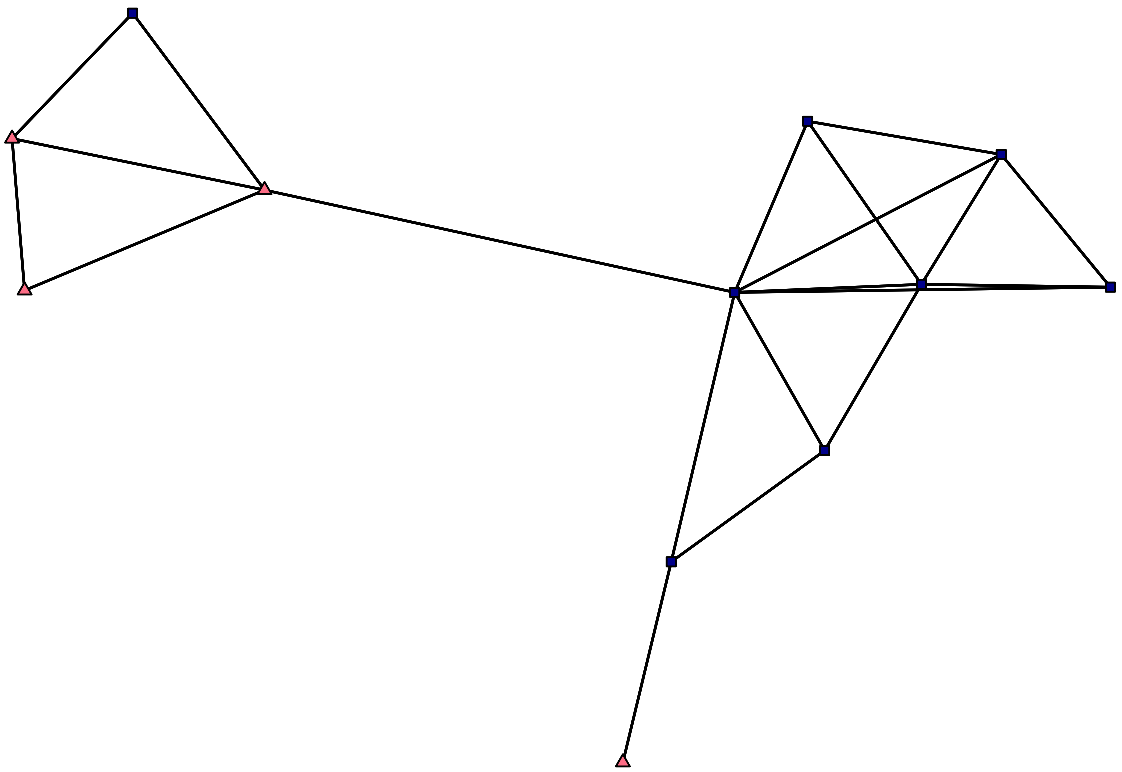
```r
table(nodes$Gender)
```

```
##
## F M
## 4 8
```

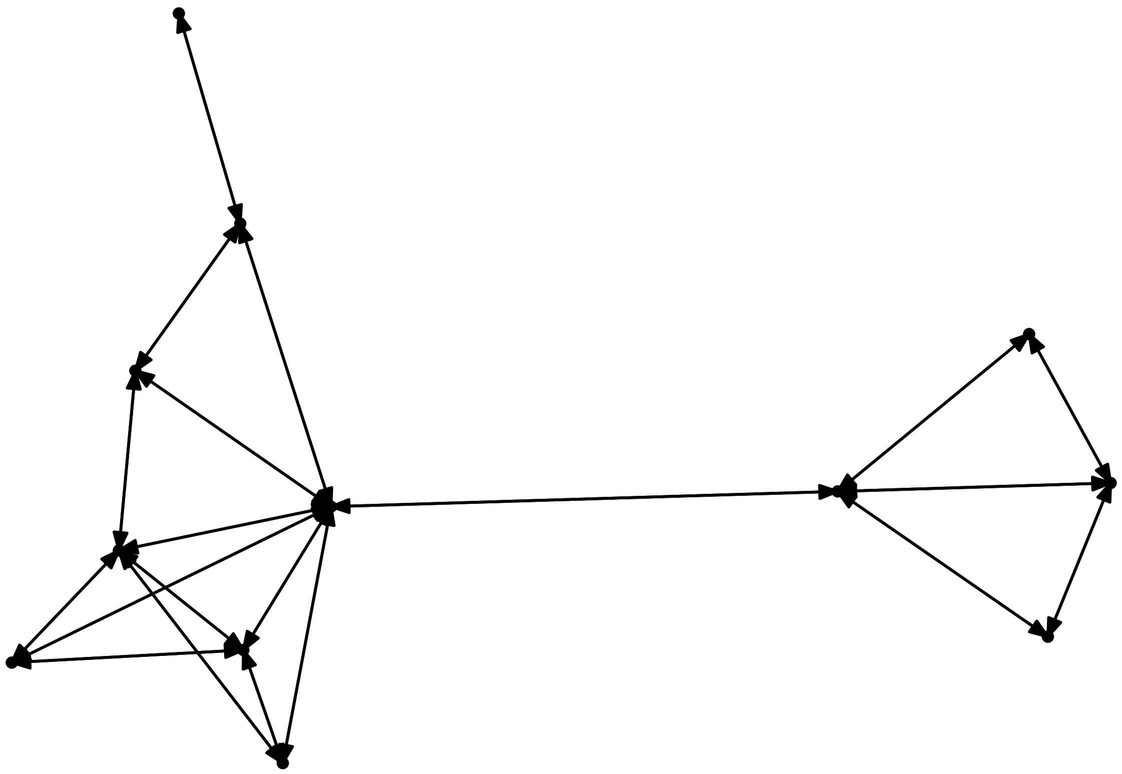
```r
mycols <- colorpanel(length(unique(nodes$class)), low = "#EFD9D3", high = "#402922")
mycols #I like using this to make sure colors are visually equidistant
```

```
## character(0)
```

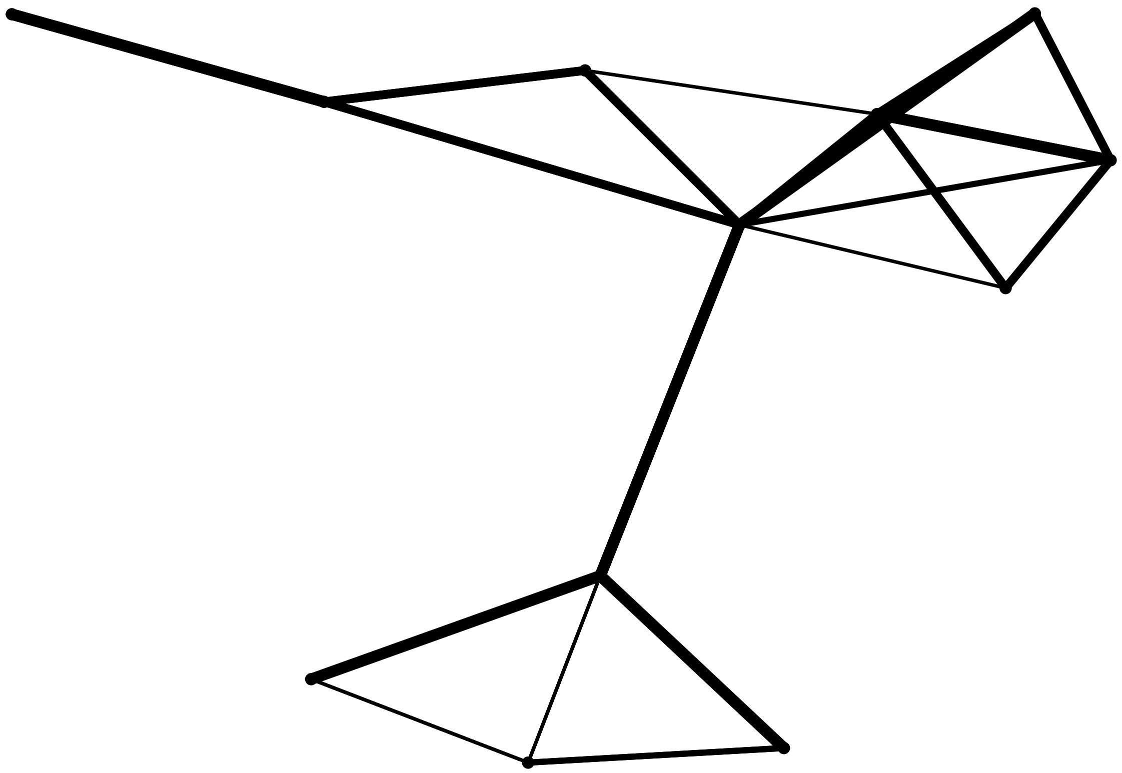
```r
ggraph(net, layout = "graphopt") +
  geom_edge_link() +
  geom_node_point(color = "black", #color is for the ring around the node
                  fill = ifelse(V(net)$Gender == 'M', "#00008b","#FC6C85"), #fill is the color for the
                  shape = ifelse(V(net)$Gender == "M", 22, 24))+ #specific shapes needed to have both c
  ggnetwork::theme_blank()
```



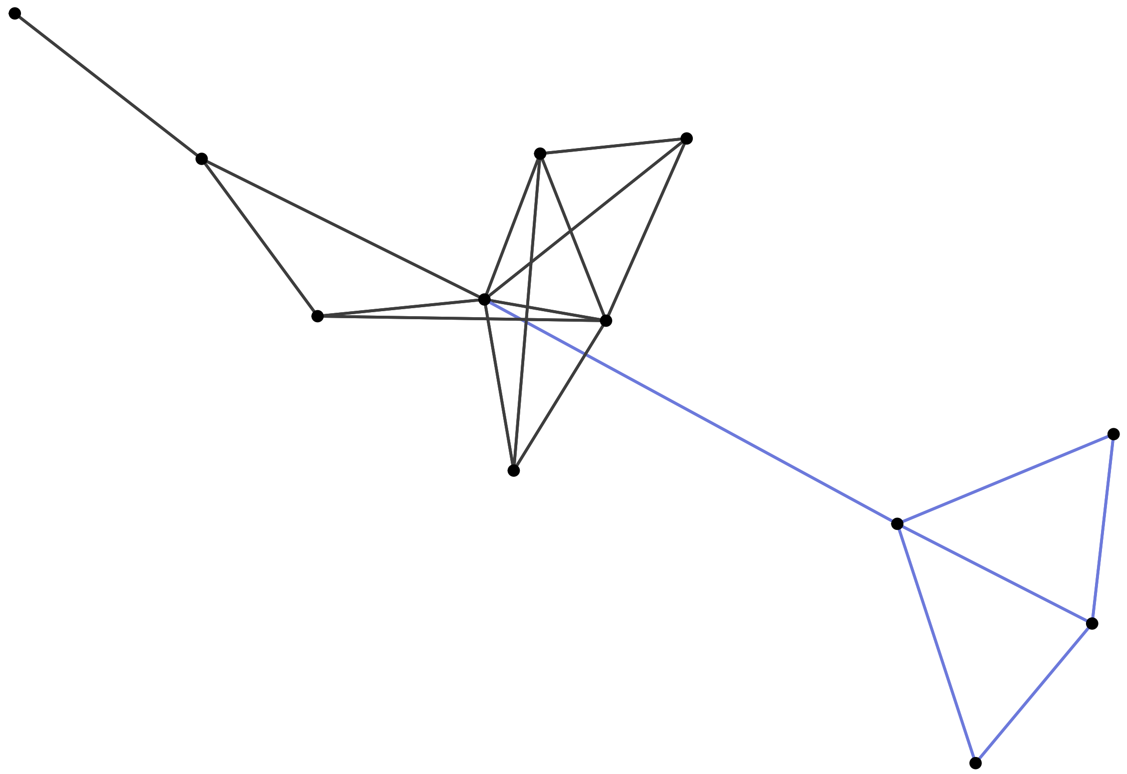
```
# 4 - edge direction, thickness, color####
ggraph(net, layout = "graphopt") +
  geom_edge_link(arrow = grid::arrow(angle = 20, #how wide should the arrow be
                                    length = unit(0.10, "in"), #how long should the arrow be
                                    type = "closed")) + #vs open arrow
  geom_node_point() +
  ggnetwork::theme_blank()
```



```r
ggraph(net, layout = "graphopt") +
  geom_edge_link(aes(width = strength)) + #weight is the name of the variable for tie strength
  scale_edge_width(range = c(0.15, 2)) + #default width is VERY wide, so I recommend reducing
  geom_node_point() +
  ggnetwork::theme_blank() +
  theme(legend.position = "none") # I don't find the legend necessary in this case
```



```r
ggraph(net, layout = "graphopt") +
  geom_edge_link0(edge_color=ifelse(E(net)$type == 'Friend', "#3D3D3D", "#6C79DB")) + #friendly ties ar
  #NOTE to get colors to render you must use geom_edge_link0 rather than geom_edge_link
  geom_node_point() +
  ggnetwork::theme_blank()
```



```
# 5 - combination####
ggraph(net, layout = "graphopt") +
  geom_edge_link0(edge_color=ifelse(E(net)$type == 'Friend',"#3D3D3D", "#6C79DB"),
                  aes(width = strength), alpha = .5,
                  arrow = grid::arrow(angle = 15,
                                      length = unit(0.12, "in"),
                                      type = "closed")) +
  scale_edge_width(range = c(0.15, 1.5)) +
  geom_node_point(size = 3,
                  color = "black",
                  fill = ifelse(V(net)$Gender == 'M', "#00008b","#FC6C85"),
                  shape = ifelse(V(net)$Gender == "M", 22, 24)) +
  ggnetwork::theme_blank(legend.position = "none") #put it all together!
```