Example 2

Kai Chung, Ying

April 4, 2017

```
library(knitr)
setwd("/Users/KevQuant/Desktop/Depaul/csc495/wk2/ex2")
read_chunk("example02_2.R")
knitr::opts_chunk$set(echo = TRUE)
```

Working with different types of data

Step 1: Load the necessary libraries and data

```
library("ggplot2")
# Must load other packages first
library("sand")
library("intergraph")
```

Loading the Southern Women network (Davis data). GraphML format.

```
# Load data
setwd("/users/KevQuant/Desktop/Depaul/csc495/wk2/ex2")
davis<-read.graph("davis.graphml", format="graphml")
```

Step 2: Examine the bipartite network

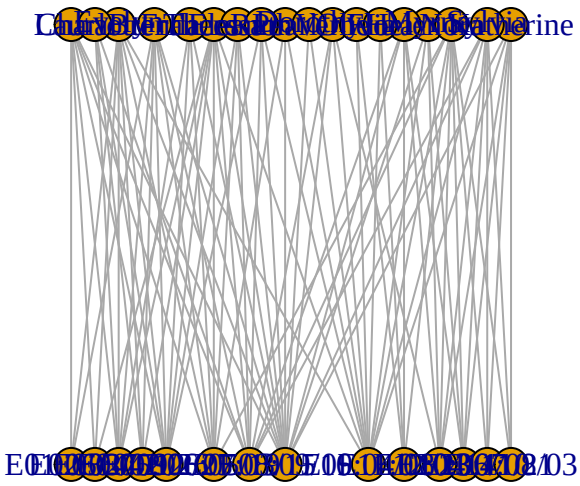
Verify that it is bipartite.

```
# Examine data
summary(davis)
```

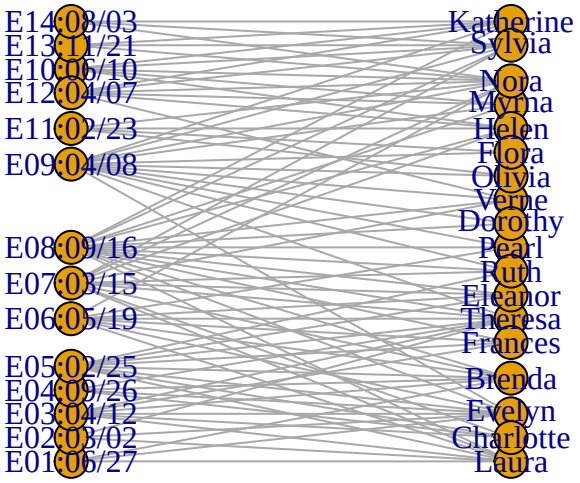
```
## IGRAPH U-WB 32 89 --
## + attr: id (v/c), x (v/n), y (v/n), type (v/l), label (v/c),
## | weight (e/n)
```

Plot using bipartite layout (better if rotated)

```
# Plot using bipartite layout
lo<-layout_as_bipartite(davis)
lo2<-cbind(lo[,2],lo[,1])
#View #1 align in Y-axis (0=bottom,1=top) by looking at the type
plot(davis,layout=layout_as_bipartite)
```



```
#For Ref, plot(davis, layout=lo)
#View #2 align in x-axis() by looking at the type
plot(davis,layout=lo2)
```



Look at labels and types

```
# Look at labels and types
#False Part
V(davis)$label[1:5]
```

```
## [1] "Evelyn"    "Laura"     "Theresa"   "Brenda"    "Charlotte"
```

```
V(davis)$type[1:5]
```

```
## [1] FALSE FALSE FALSE FALSE FALSE
```

```
#True part
V(davis)$label[20:25]
```

```
## [1] "E02:03/02" "E03:04/12" "E04:09/26" "E05:02/25" "E06:05/19" "E07:03/15"
```

```
V(davis)$type[20:25]
```

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

So, FALSE is people

Step 3: Create projection

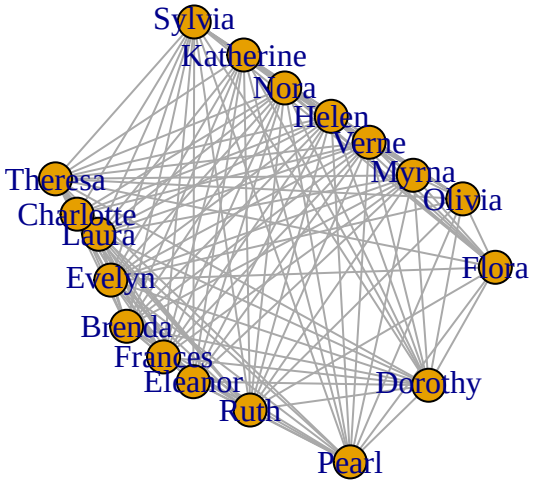
Create person-person projection

```
# Create person-person projection
#subset and transfer to single mode the network
#by the "which" argument ("TRUE" or "FALSE"or "BOTH")
davis.p2p<-bipartite_projection(davis, which="FALSE")
summary(davis.p2p)
```

```
## IGRAPH U-W- 18 139 --
## + attr: id (v/c), x (v/n), y (v/n), label (v/c), weight (e/n)
```

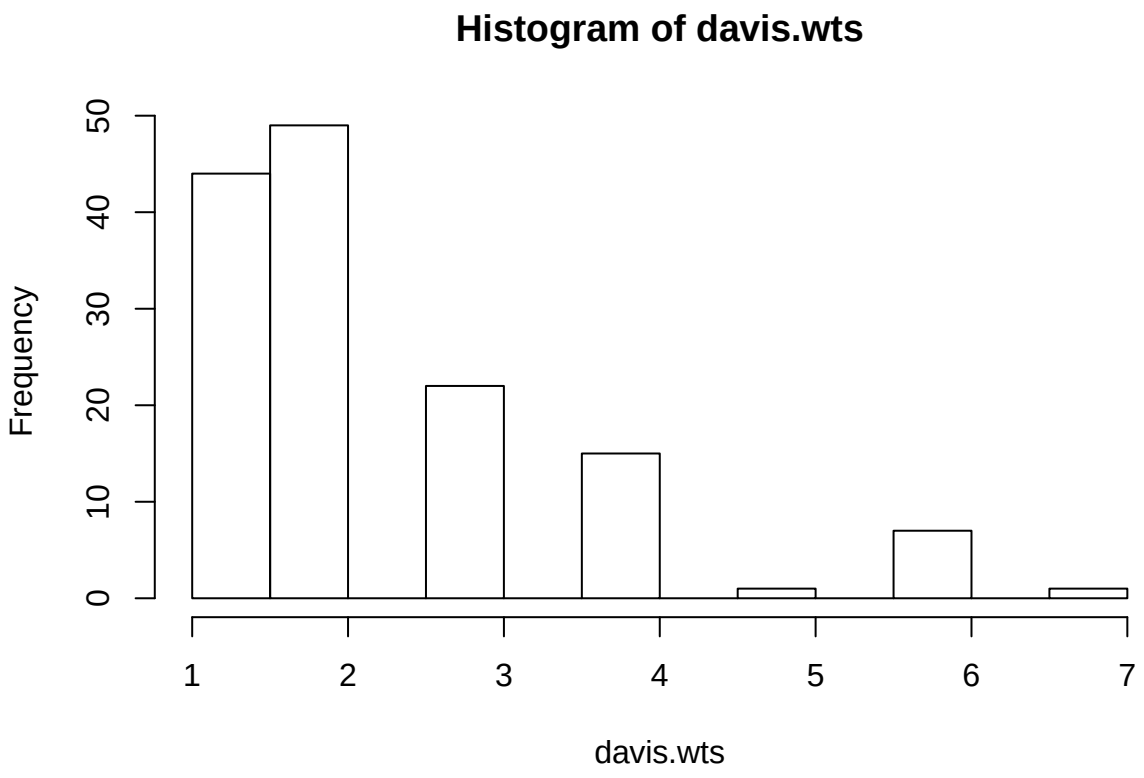
Plot the projected graph

```
# Plot the projected graph
plot(davis.p2p)
```



Plot the distribution of edge weights

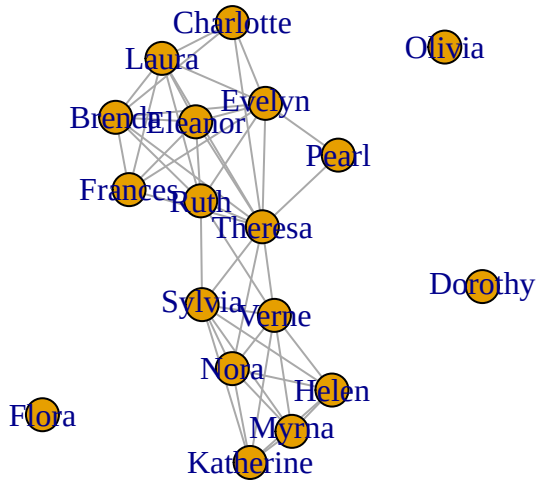
```
# Plot the distribution of edge weights
davis.wts<-E(davis.p2p)$weight
hist(davis.wts)
```



Step 4: Filtered version of the network

Create a new network removing the edges of weight 1 and 2.

```
# Create a new network removing the edges of weight 1 and 2.
davis.filt<-delete_edges(davis.p2p,E(davis.p2p)[E(davis.p2p)$weight<=2])
plot(davis.filt,layout=layout_with_kk)
```

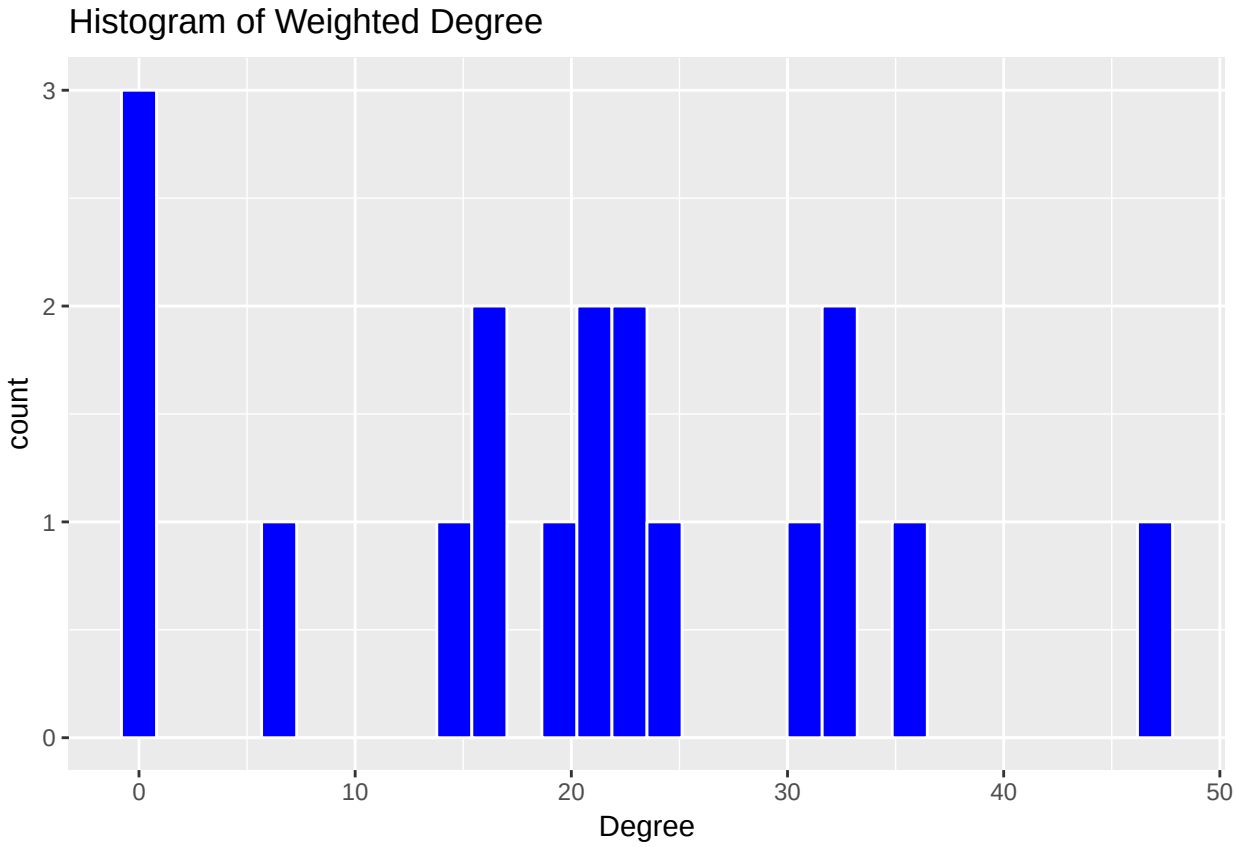


Plot the weighted degree distribution of the new network

```
# Plot the weighted degree distribution of the new network
davis.deg<-graph.strength(davis.filt)
p<-ggplot(data.frame(Degree=davis.deg),aes(x=Degree))
p<-p+geom_histogram(col="white",fill="blue")
```

```
p<-p+ggtitle("Histogram of Weighted Degree")
print(p)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Step 5: Ego networks

Extract the ego networks for Laura and Sylvia

```
# Extract the ego networks for Laura and Sylvia
V(davis.filt)$label=="Laura"
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
which(V(davis.filt)$label=="Laura")
```

```
## [1] 2
```

```
which(V(davis.filt)$label=="Sylvia")
```

```
## [1] 13
```

```
#Make ego graph with order=1 for each of nodes("Laura","Sylvia")
egos<-make_ego_graph(davis.filt,1,node=c(2,13))
egos
```

```
## [[1]]
## IGRAPH U-W- 8 24 --
```

```
## + attr: id (v/c), x (v/n), y (v/n), label (v/c), weight (e/n)
## + edges:
##  [1] 1--2 1--3 1--4 1--5 1--6 1--7 1--8 2--3 2--4 2--5 2--6 2--7 2--8 3--4
## [15] 3--5 3--6 3--7 3--8 4--5 4--6 4--7 4--8 6--7 7--8
##
## [[2]]
## IGRAPH U-W- 8 21 --
## + attr: id (v/c), x (v/n), y (v/n), label (v/c), weight (e/n)
## + edges:
##  [1] 1--2 1--3 1--6 1--7 2--3 2--6 3--4 3--5 3--6 3--7 3--8 4--5 4--6 4--7
## [15] 4--8 5--6 5--7 5--8 6--7 6--8 7--8
```

```
#Laura ego
laura<-egos[[1]]

#Sylvia ego
sylvia<-egos[[2]]
```

Plot side-by-side use layout_as_star

```
# Plot side-by-side (use par(mfrow...) to do this).
#Check labels order in their (laura,sylvia) own network
V(laura)$label
```

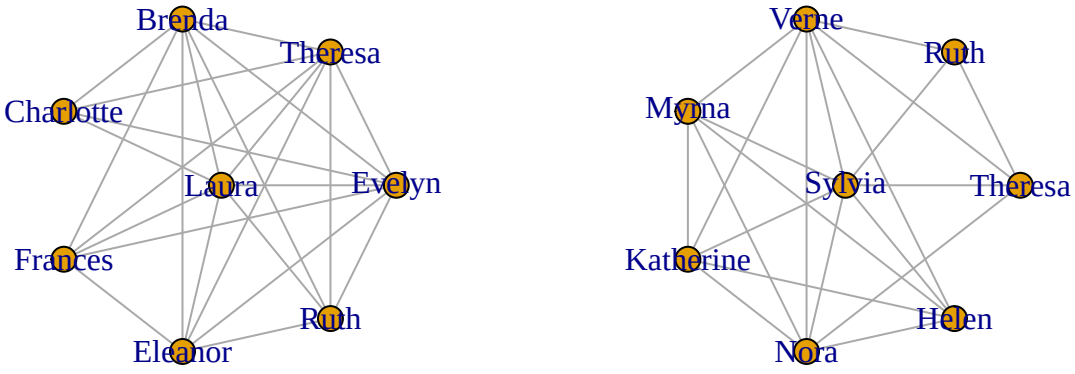
```
## [1] "Evelyn"    "Laura"     "Theresa"   "Brenda"    "Charlotte" "Frances"
## [7] "Eleanor"   "Ruth"
```

```
V(sylvia)$label
```

```
## [1] "Theresa"   "Ruth"      "Verne"     "Myrna"     "Katherine" "Sylvia"
## [7] "Nora"      "Helen"
```

```
#From above, we got the location of labels in #2(laura) and #6(sylvia)
laura.lo<-layout_as_star(laura,V(laura)[2])
sylvia.lo<-layout_as_star(sylvia,V(sylvia)[6])

par(mfrow=c(1,2))
plot(laura,layout=laura.lo)
plot(sylvia,layout=sylvia.lo)
```




```
par(mfrow=c(1,1))
```

Compare the two ego networks by weighted degree. First create data frame with the right structure.

wdeg, ego 10, Laura 24, Sylvia 34, Laura

```
# Create comparative data frame
```

Then plotting with ggplot is straightforward.

```
# Plot histogram
```

Step 6: New example: loading edge and attribute data

Load CSV files for edges and attributes. stringAsFactors is false because we can't use factors as node attributes.

```
# Edge and attribute data
edge.df <- read.csv("edgelist-sample.csv",stringsAsFactors = FALSE)
node.df <- read.csv("nodeattr-sample.csv",stringsAsFactors = TRUE)

class(edge.df)
```

```
## [1] "data.frame"
```

```
class(node.df)
```

```
## [1] "data.frame"
```

Convert to a graph from the two data frames. Note that the names of the vertices in the edge data frame have to match the first column of the vertices data frame.

```
# Convert to graph
gr<-graph.data.frame(edge.df,vertices = node.df,directed = TRUE)
summary(gr)
```

```
## IGRAPH DNW- 9 20 --
## + attr: name (v/c), Age (v/n), Sex (v/c), weight (e/n)
```

Plot

```
# Plot
plot(gr)
```