Assignment 2 - Social Network Analysis

Part I

Start by installing the "igraph" package. Once you have installed igraph, load the package.

Now upload the data file "discipline-data.csv" as a data frame called "D1". Each row is a disciplinary action from a teacher to a student so the first line shows that teacher "E" sent student "21" to the principal. It also shows the gender of both the teacher and student and the student's main elective field of study ("major"") and the field that the teacher instructs in (".t.expertise").

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.5.2
```

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

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

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

```
D1<-read.csv("~/Desktop/master fall/hudk4050/assignment2/discipline-data.csv")
D1 <- data.frame(D1)
```

Before you proceed, you will need to change the data type of the student id variable. Since it is a number R will automatically think it is an integer and code it as such (look at the list of variables by clicking on the data frame arrow in the Data pane. Here you will see the letters "int"" next to the stid variable, that stands for integer). However, in this case we are treating the variable as a category, there is no numeric meaning in the variable. So we need to change the format to be a category, what R calls a "factor". We can do this with the following code:

```
D1$stid <- as.factor(D1$stid)
```

igraph requires data to be in a particular structure. There are several structures that it can use but we will be using a combination of an "edge list" and a "vertex list". As you might imagine the edge list contains a list of all the relationships between students and teachers and any characteristics of those edges that we might be interested in. There are two essential variables in the edge list a "from" variable and a "to" variable that descibe the relationships between vertices (a disciplinary action is given "from" and teacher "to" a student). While the vertix list contains all the characteristics of those vertices, in our case gender and major.

So let's convert our data into an edge list!

First we will isolate the variables that are of interest: tid and stid

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```
```r
D2 <- select(D1, tid, stid)
```

Since our data represnts every time a teacher sends a student to the principal there are multiple rows when
the same teacher sends the same student. We want to collapse these into a single row, with a variable that
shows how many times a teacher-student pair appears.

```r
EDGE <- count(D2, tid, stid)

names(EDGE) <- c("from", "to", "count")
EDGE
```

```
## # A tibble: 41 x 3
##     from  to    count
##     <fct> <fct> <int>
##  1 A     2         1
##  2 A     3         1
##  3 A     4         2
##  4 A     5         1
##  5 A     18        1
##  6 A     22        1
##  7 A     25        2
##  8 B     1         2
##  9 B     5         1
## 10 B     8         1
## # ... with 31 more rows
```

EDGE is your edge list. Now we need to make the vertex list, a list of all the teachers and students and their
characteristics in our network.

```r
#First we will separate the teachers from our original data frame
V.TCH <- select(D1, tid, t.gender, t.expertise)
#Remove all the repeats so that we just have a list of each teacher and their characteristics
V.TCH <- unique(V.TCH)
#Add a variable that describes that they are teachers
V.TCH$group <- "teacher"

#Now repeat this process for the students
V.STD <- select(D1, stid, s.gender, s.major)
V.STD <- unique(V.STD)
V.STD$stid <- as.factor(V.STD$stid)
V.STD$group <- "student"

#Make sure that the student and teacher data frames have the same variables names
names(V.TCH) <- c("id", "gender", "topic", "group")
names(V.STD) <- c("id", "gender", "topic", "group")

#Bind the two data frames together (you will get a warning because the teacher data frame has 5 types o
VERTEX <- bind_rows(V.TCH, V.STD)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

VERTEX

```
##    id gender   topic   group
## 1   E female     art teacher
## 2   A female biology teacher
## 3   D female     art teacher
## 4   B   male    math teacher
## 5   C   male biology teacher
## 6  21 female     art student
## 7   4   male    math student
## 8  25   male biology student
## 9  15 female english student
## 10  1   male biology student
## 11  8   male english student
## 12 10   male     art student
## 13 26 female     art student
## 14 23 female     art student
## 15  3   male     art student
## 16 27 female     art student
## 17 29 female     art student
## 18  5   male     art student
## 19 18   male biology student
## 20 28   male    math student
## 21 11   male     art student
## 22 17   male biology student
## 23  9   male biology student
## 24 12 female     art student
## 25 24   male     art student
## 26 22 female biology student
## 27  2 female    math student
## 28 19 female biology student
## 29  6   male biology student
## 30 20   male biology student
```

Now we have both a Vertex and Edge list it is time to plot our graph!

```
#Load the igraph package

library(igraph)

#First we will make an object that contains the graph information using our two dataframes EDGE and VER
```

```
g <- graph.data.frame(EDGE, directed=TRUE, vertices=VERTEX)
g
```

```
## IGRAPH 94b56d7 DN-- 30 41 --
## + attr: name (v/c), gender (v/c), topic (v/c), group (v/c), count
## | (e/n)
## + edges from 94b56d7 (vertex names):
##  [1] A->2  A->3  A->4  A->5  A->18 A->22 A->25 B->1  B->5  B->8  B->9
## [12] B->12 B->20 B->24 B->26 B->29 C->3  C->9  C->10 C->19 C->21 C->24
## [23] C->27 D->1  D->2  D->4  D->5  D->11 D->15 D->18 D->23 D->25 D->28
## [34] E->6  E->8  E->15 E->17 E->21 E->25 E->26 E->27
```
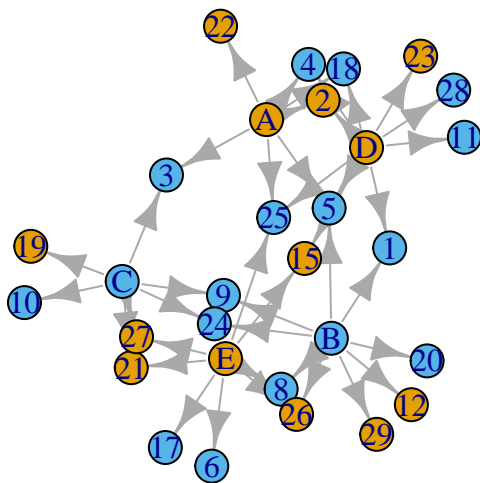
#Now we can plot our graph using the force directed graphing technique – our old friend Fruchertman–Rei

```
plot(g,layout=layout.fruchterman.reingold)
```
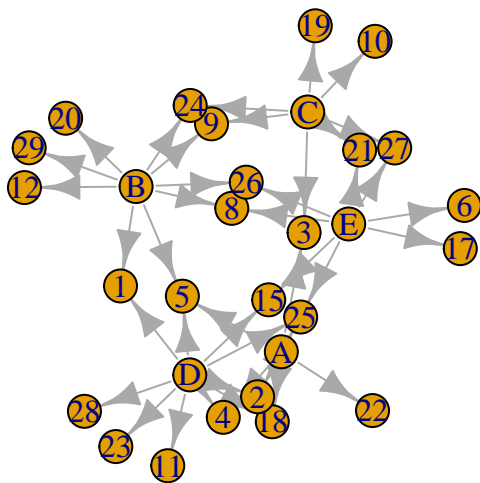

#There are many ways to change the attributes of the graph to represent different characteristics of th

```
plot(g,layout=layout.fruchterman.reingold, vertex.color=VERTEX$gender)
```


#We can change the thickness of the edge according to the number of times a particular teacher has sent

```
plot(g,layout=layout.fruchterman.reingold, vertex.color=VERTEX$gender, edge.width=EDGE$count)
```
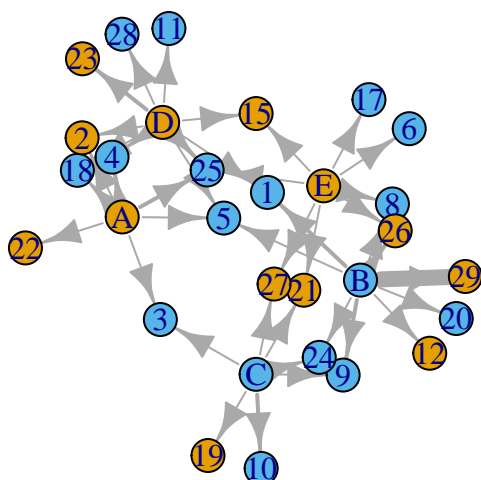
Part II

In Part II your task is to look up in the igraph documentation and create a graph that sizes the student vertices in terms of the number of disciplinary actions they have recieved, and the teachers in terms of the number of disciplinary actions they have given out.

```
countstd <- EDGE %>% group_by(to)%>%summarise(sum(count))
counttch <- EDGE %>% group_by(from)%>%summarise(sum(count))
names(countstd) <- c("id","count")
names(counttch) <- c("id","count")
VERTEX1<- bind_rows(countstd,counttch)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
VERTEX1
```

```
## # A tibble: 30 x 2
##      id    count
##      <chr> <int>
##  1 1        3
##  2 2        2
##  3 3        2
##  4 4        3
##  5 5        3
##  6 6        1
##  7 8        2
##  8 9        3
##  9 10       2
## 10 11       1
## # ... with 20 more rows
```

```
g <- graph.data.frame(EDGE, directed=TRUE, vertices=VERTEX1)
g
```

```
## IGRAPH 2706815 DN-- 30 41 --
## + attr: name (v/c), count (v/n), count (e/n)
## + edges from 2706815 (vertex names):
##  [1] A->2  A->3  A->4  A->5  A->18 A->22 A->25 B->1  B->5  B->8  B->9
## [12] B->12 B->20 B->24 B->26 B->29 C->3  C->9  C->10 C->19 C->21 C->24
## [23] C->27 D->1  D->2  D->4  D->5  D->11 D->15 D->18 D->23 D->25 D->28
## [34] E->6  E->8  E->15 E->17 E->21 E->25 E->26 E->27
```

```r
plot(g,layout=layout.fruchterman.reingold,edge.width=EDGE$count,vertex.size = VERTEX1$count*2)
```
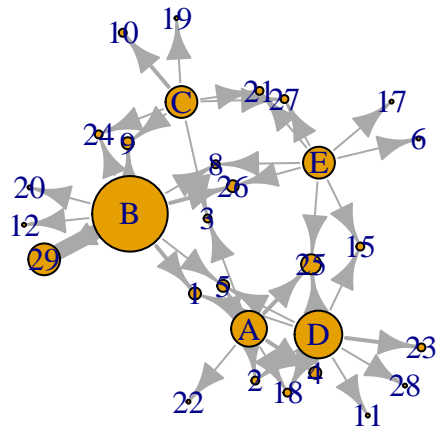

Part III

Now practice with data from our class. Please create a **person-network** with the data set hudk4050-classes.csv. To create this network you will need to create a person-class matrix using the tidyr functions and then create a person-person matrix using `t()`. You will then need to plot a matrix rather than a data frame using igraph.

Once you have done this, also look up how to generate the following network metrics: betweeness centrality and dregree. **Who is the most central person in the network?**

```r
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:igraph':
##
##     crossing
```

```r
#modify data to get revelent variables
D3 <- read.csv("~/Desktop/master fall/hudk4050/assignment2/hudk4050.classes.csv")
D3 <- data.frame(D3)
names(D3) <- c("FirstName", "LastName", "Class1", "Class2","Class3","Class4","Class5","Class6")
D4 <- select(D3,FirstName,Class1,Class2,Class3,Class4,Class5,Class6)
D6<-  gather(D4,coursenum,course,Class1,Class2,Class3,Class4,Class5,Class6)
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```r
D7 <- select(D6,FirstName,course)
#manage the course,that each student may type same course with diffterent way
D7 <- D7%>%filter(!course == "") %>% filter(course != "4050")
```

```
##      FirstName     course
## 1        Yawei  HUDK 4050
## 2      Ningyao  HUDK 4050
## 3       Qiyang  HUDK 4050
## 4      Bernell  HUDK 4050
## 5        Ruiqi   HUDM4125
## 6     Leonardo  HUDM 5122
## 7     Zhaozhuo   HUDK4050
## 8     Jiancong  HUDK 4052
## 9        ZIFAN   HUDK4050
## 10     Allison  HUDK 4050
## 11       Yiwen  HUDK 4050
## 12      Beibei  HUDK 4050
## 13      yixiao   HUDK4050
## 14       xinyi   HUDK4050
## 15      Jingru  HUDK 4050
## 16      Ziyuan   HUDK4050
## 17     Timothy  HUDK 4050
## 18          XI   HUDK4050
## 19      Chenyu  CCPX 4023
## 20        Yiyi  HUDK 4052
## 21          Di  HUDM 5026
## 22         Han  HUDK 4050
## 23     Xiaowen   HUDK4029
## 24        Anqi  HUDK 4050
## 25        Ling   HUDK4050
## 26       Yiwei  HUDK 4050
## 27      jiahao   HUDK4050
## 28    LINGLING  HUDM 4125
## 29      Shijie   HUDK4050
## 30       Yujun   HUDK4050
## 31   chaoxiong   HUDK4050
## 32     Lintong  HUDM 5026
## 33         HAN   HUDK4050
## 34    Alysandra   HUDM4122
## 35      Yixuan  HUDK 4050
## 36      XUDIAN  HUDK 4050
## 37        YAQI  HUDK 4050
## 38   Christine  BBSN 5007
## 39       David  HUDK 4050
## 40    Chengxuan   HUDK4050
## 41   Zhongyuan  HUDK 4050
## 42        Maho  HUDK 4050
## 43      Minruo   HUDK4050
## 44         Jie  HUDK 4050
## 45      INDIRA   HUDK4029
## 46 Eudora Xinyi   ITSF4090
## 47     Joellyn  HUDK 4050
## 48       Yigao   HUDK4050
## 49      Wanruo   HUDK4050
## 50       Suwon   HUDK4050
## 51        Luyi  HUDK 4050
```

```
## 52       Yawei   HUDK 4052
## 53     Ningyao   HUDM 4125
## 54      Qiyang   HUDM 4122
## 55     Bernell   HUDK 4052
## 56       Ruiqi   HUDM5026
## 57    Leonardo   HUDM 5026
## 58    Zhaozhuo   HUDK4052
## 59    Jiancong   HUDK 4050
## 60       Yiwen   HUDM 5026
## 61      Beibei   EDPE 6151
## 62      yixiao IFSF4090002
## 63       xinyi   HUDM4122
## 64      Jingru   HUDM 4125
## 65      Ziyuan   HUDM5026
## 66     Timothy   HUDK 4052
## 67          XI   HUDM4125
## 68      Chenyu   HUD 4120
## 69        Yiyi   HUDK 4050
## 70          Di   HUDM 5126
## 71         Han   HUDK 5011
## 72     Xiaowen   HUDK4050
## 73        Anqi   HUDK 4052
## 74        Ling   HUDK5053
## 75       Yiwei   HUDK 5023
## 76      jiahao   HUDM4125
## 77    LINGLING   HUDM 5026
## 78      Shijie   HUDM4125
## 79       Yujun   HUDK4052
## 80    chaoxiong   HUDK4052
## 81     Lintong   HUDK 4052
## 82        ZIMO        4125
## 83         HAN   HUDK4052
## 84    Alysandra   CCPX4023
## 85      Yixuan   HUDK 4052
## 86      XUDIAN   HUDM 4125
## 87        YAQI   EDPE 4056
## 88   Christine   HUDK 4050
## 89       David   HUDK 4029
## 90    Chengxuan   HUDM4125
## 91   Zhongyuan   HUDK 4052
## 92        Maho   MSTU 4000
## 93      Minruo   HUDM4122
## 94         Jie   HUDK 4052
## 95      INDIRA   HUDK4080
## 96 Eudora Xinyi   ITSF5008
## 97     Joellyn   QMSS 5010
## 98       Yigao   ITSF4090
## 99      Wanruo   HUDK4029
## 100      Suwon   HUDK4052
## 101       Luyi   HUDM 4125
## 102      Yawei   HUDK 5053
## 103    Ningyao   HUDM 5126
## 104     Qiyang   HUDK 4029
## 105    Bernell   HUDM 5126
```

```
## 106      Ruiqi     HUDM5126
## 107    Leonardo    HUDK 4052
## 108    Zhaozhuo    HUDM4122
## 109    Jiancong    HUDK 4029
## 110      Yiwen     HUDM 5126
## 111     Beibei     EDPE 4155
## 112     yixiao EDPS4002001
## 113      xinyi     HUDK4052
## 114     Jingru     HUDM 5126
## 115     Ziyuan     HUDM4125
## 116    Timothy       G 5067
## 117         XI     HUDM5026
## 118     Chenyu     HUDK 4050
## 119       Yiyi       B 8306
## 120         Di     HUDM 4125
## 121        Han     MSTU 5027
## 122    Xiaowen     HUDK4052
## 123       Anqi     MSTU 4052
## 124       Ling     MSTU5002
## 125      Yiwei      HUD 4120
## 126     jiahao     HUDM5126
## 127   LINGLING     HUDM 5126
## 128     Shijie     HUDM5026
## 129      Yujun     HUDK4080
## 130  chaoxiong     HUDK4029
## 131    Lintong     HUDK 4050
## 132       ZIMO          5026
## 133        HAN     HUDK4029
## 134   Alysandra    HUDK4050
## 135     Yixuan     HUDK 4029
## 136     XUDIAN     HUDM 5026
## 137       YAQI     ORLD 4085
## 138      David     BBSN 5019
## 139   Chengxuan    HUDM5126
## 140   Zhongyuan    HUDM 5122
## 141       Maho     MSTU 5003
## 142     Minruo  QMSS-G5072
## 143        Jie     HUDK 4029
## 144     INDIRA     HUDM4120
## 145 Eudora Xinyi    ITSF5035
## 146    Joellyn     QMSS 5015
## 147      Yigao     ITSF4025
## 148     Wanruo     HUDK4052
## 149      Suwon     HUDK4029
## 150       Luyi     HUDM 5026
## 151    Ningyao     HUDM 5026
## 152     Qiyang     HUDK 4080
## 153     Bernell QMSS GR5067
## 154      Ruiqi     A&HA4063
## 155    Leonardo    HUDK 4050
## 156    Zhaozhuo    HUDM4120
## 157    Jiancong    HUDM 4050
## 158      Yiwen     HUDM 4125
## 159     Beibei     EDPA 4047
```

```
## 160       yixiao EDPS4021001
## 161        xinyi    MSTU4039
## 162       Jingru   ORLD 4051
## 163       Ziyuan    HUDM5126
## 164      Timothy      G 5072
## 165           XI    HUDM5126
## 166       Chenyu   HUDK 4052
## 167         Yiyi   HUDM 4122
## 168           Di    HUDK4050
## 169      Xiaowen    HUDM5123
## 170         Anqi   HUDM 5122
## 171         Ling    MSTU4023
## 172       jiahao    HUDM5026
## 173     LINGLING   HUDK 4050
## 174       Shijie    HUDM5126
## 175        Yujun    HUDM5026
## 176    chaoxiong    CCPJ5056
## 177      Lintong   HUDK 4029
## 178         ZIMO        5126
## 179          HAN    MSTU4031
## 180       Yixuan   MSTU 4039
## 181       XUDIAN   HUDM 5126
## 182         YAQI   EDPA 6002
## 183        David   HUDM 5026
## 184     Chengxuan    HUDM5026
## 185    Zhongyuan   HUDM 5026
## 186         Maho   MSTU 4083
## 187       Minruo    ITSF4098
## 188          Jie   COMS 4706
## 189       INDIRA    HUDK4050
## 190 Eudora Xinyi    HUDK4050
## 191      Joellyn   QMSS 5072
## 192        Yigao    ITSF5035
## 193       Wanruo    CCPJ5062
## 194         Luyi   HUDM 5126
## 195       Chenyu   HUDM 4122
## 196           Di QMSS G 5015
## 197        Yujun    MSTU4052
## 198          HAN    HUDM5123
## 199       XUDIAN   QMSS 5015
## 200        David   BBSN 4904
## 201         Maho   MSTU 4052
## 202      Joellyn   STAT 4205
## 203       Wanruo    A&HA4063
## 204      Joellyn   QMSS 5021
```

```r
names(D7)<- c("student","course")
D7$course <- gsub(" ","",D7$course)
D7$course <- gsub("5026","HUDM5026",D7$course)
D7$course <- gsub("5126","HUDM5126",D7$course)
D7$course <- gsub("QMSS","G",D7$course)
D7$course <- gsub("GG","G",D7$course)
D7$course <- gsub("GGR","G",D7$course)
#drop hudk4050, since everyone takes hudk4050
```

```r
D7 <- D7 %>% filter(course != "HUDK4050")
#generate person-class matrix
D8<- mutate(D7,enrolled = 1)
D9 <- spread(D8,course,enrolled,0)
#generate person-person matrix
D10<-select(D9,-student)
D10 <- as.matrix(D10)
D11 <- t(D10)
D12<- D10 %*% D11
diag(D12)<-0
colnames(D12)<-D9$student
rownames(D12)<-D9$student
#plot igraph
g1 <-graph_from_adjacency_matrix(D12,mode="undirected")
V(g1)$label.cex <- seq(0.5,0.5,length.out=6)
```

```
## Warning in vattrs[[name]][index] <- value: number of items to replace is
## not a multiple of replacement length
```

```r
plot(g1,layout=layout.fruchterman.reingold,,vertex.size = 10,edge.arrow.size=0.3,vertex.label = D9$stud
```
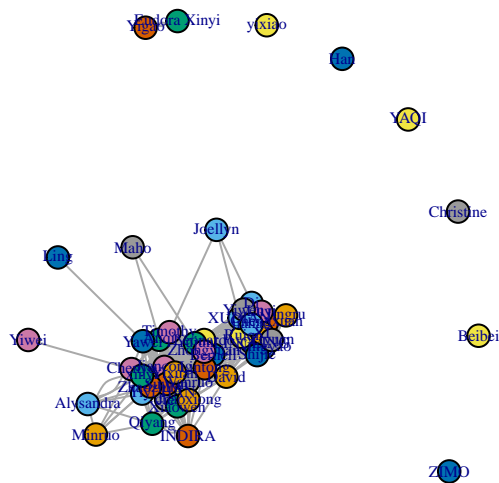

get betweeness centrality and max betweeness centrality person

```r
peoplebetween=betweenness(g1)
maxpeoplebetween <-peoplebetween[peoplebetween == max(peoplebetween)]
maxpeoplebetween
```

```
##      Yujun
## 79.03208
```

get degree and people with max degree

```r
degree <- degree(g1)
maxdegree <- degree[degree == max(degree)]
maxdegree
```

```
## Lintong
##      46
```

To Submit Your Assignment

Please submit your assignment by first "knitting" your RMarkdown document into an html file and then comit, push and pull request both the RMarkdown file and the html file.