In [ ]:

목차

In [ ]:

01. 관계도 그리기

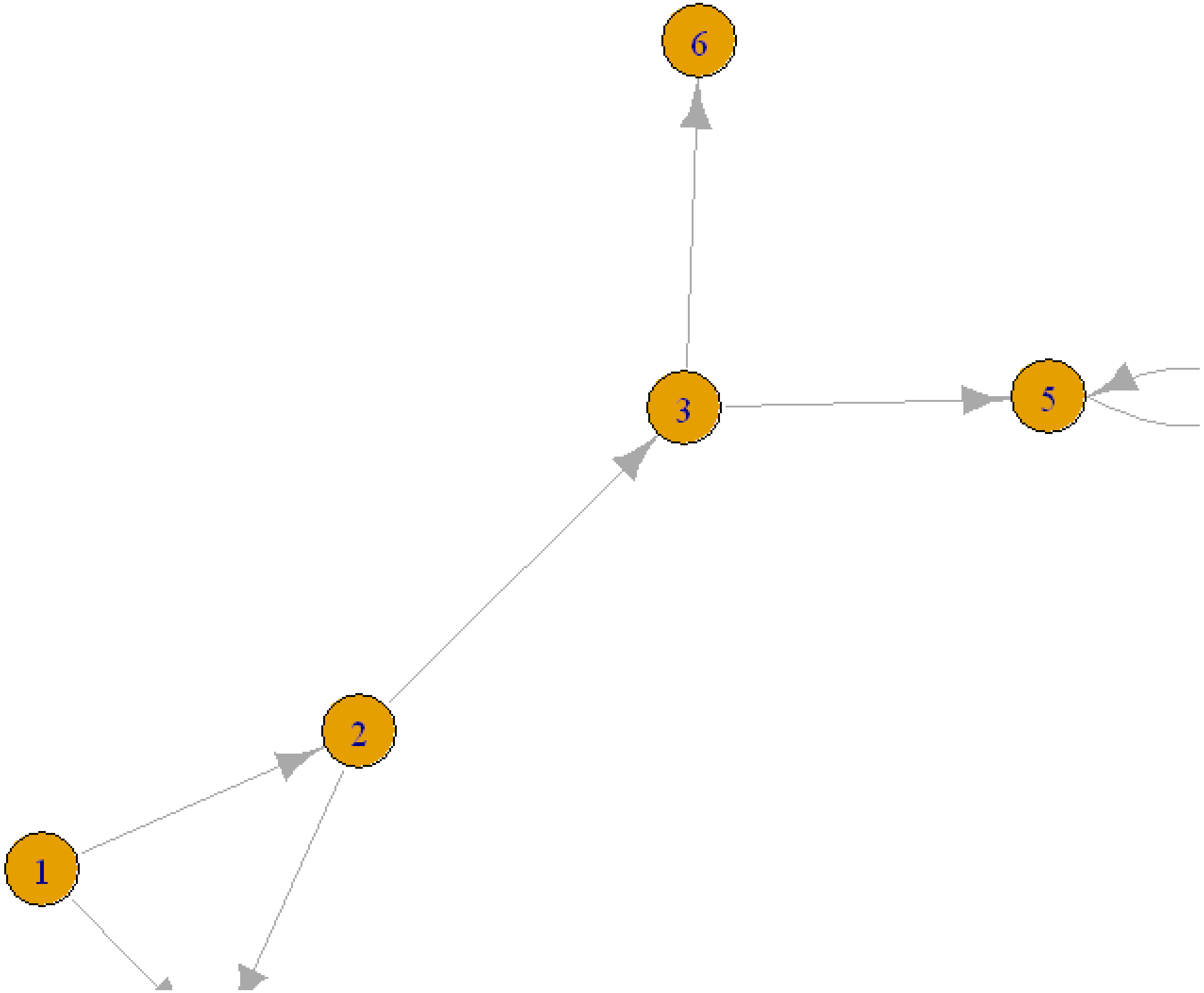
In [6]:

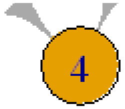
```
library(igraph)
```

```
Attaching package: 'igraph'

The following objects are masked from 'package:stats':

    decompose, spectrum

The following object is masked from 'package:base':

    union
```

In [13]:

```
g1 <- graph( c(1,2,2,3,2,4,1,4,5,5,3,6,3,5) )
plot(g1)
```




4

In [14]:

```
str(g1)
```

```
List of 10
 $ :List of 1
  ..$ :Class 'igraph.vs'  atomic [1:2] 2 4
  .. .. ..- attr(*, "env")=<weakref>
  .. .. ..- attr(*, "graph")= chr "3f1e7196-73dd-11e8-acc7-2310820ee7fa"
 $ :List of 1
  ..$ :Class 'igraph.vs'  atomic [1:2] 3 4
  .. .. ..- attr(*, "env")=<weakref>
  .. .. ..- attr(*, "graph")= chr "3f1e7196-73dd-11e8-acc7-2310820ee7fa"
 $ :List of 1
  ..$ :Class 'igraph.vs'  atomic [1:2] 5 6
  .. .. ..- attr(*, "env")=<weakref>
  .. .. ..- attr(*, "graph")= chr "3f1e7196-73dd-11e8-acc7-2310820ee7fa"
 $ :List of 1
  ..$ :Class 'igraph.vs'  atomic (0)
  .. .. ..- attr(*, "env")=<weakref>
  .. .. ..- attr(*, "graph")= chr "3f1e7196-73dd-11e8-acc7-2310820ee7fa"
 $ :List of 1
  ..$ :Class 'igraph.vs'  atomic [1:1] 5
  .. .. ..- attr(*, "env")=<weakref>
  .. .. ..- attr(*, "graph")= chr "3f1e7196-73dd-11e8-acc7-2310820ee7fa"
 $ :List of 1
  ..$ :Class 'igraph.vs'  atomic (0)
  .. .. ..- attr(*, "env")=<weakref>
  .. .. ..- attr(*, "graph")= chr "3f1e7196-73dd-11e8-acc7-2310820ee7fa"
 $ :

Error in adjacent_vertices(x, i, mode = if (directed) "out" else "all"): At iterators.c:759 : Cannot create iterator, invalid vert
ex id, Invalid vertex id
Traceback:

1. str(g1)
2. str.default(g1)
3. object[[i]]
4. `[[.igraph`(object, i)
5. adjacent_vertices(x, i, mode = if (directed) "out" else "all")
```

02. 군집분석(Clustering)

In [34]:

```r
library(igraph)
df <- read.csv("D:/dataset/RLoveYou/clustering.csv")
head(df,10)
graph <- data.frame(학생=df$학생, 교수=df$교수)
```

학생	교수
S272	T1
S282	T1
S336	T1
S344	T1
S242	T2
S315	T2
S55	T2
S59	T2
S149	T2
S153	T2

In [43]:

```
library(stringr)
df <- graph.data.frame(graph,directed=TRUE)
df

gubun1 <- V(df)$name
head(gubun1,100)          # 이름 출력
gubun <- str_sub(gubun1, start=1, end=1)
table(gubun)
```

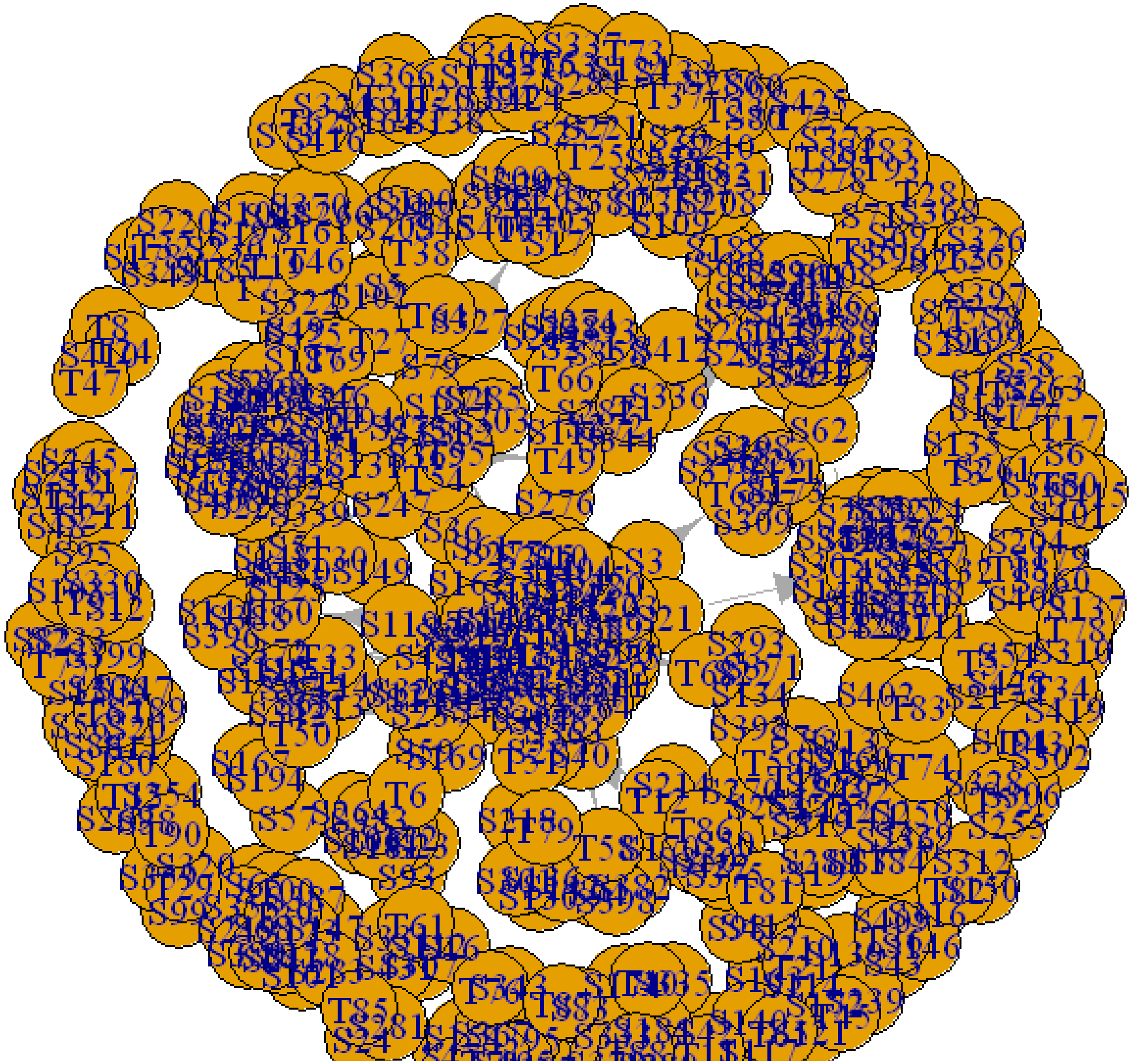
```
IGRAPH cd45da4 DN-- 526 694 --
+ attr: name (v/c)
+ edges from cd45da4 (vertex names):
 [1] S272->T1 S282->T1 S336->T1 S344->T1 S242->T2 S315->T2 S55 ->T2 S59 ->T2
 [9] S149->T2 S153->T2 S356->T2 S426->T2 S14 ->T2 S14 ->T2 S55 ->T2 S223->T2
[17] S25 ->T2 S46 ->T2 S153->T2 S169->T2 S363->T2 S423->T2 S224->T3 S135->T3
[25] S261->T3 S294->T4 S145->T4 S145->T4 S162->T4 S162->T4 S166->T4 S196->T4
[33] S196->T4 S219->T4 S237->T4 S237->T4 S242->T4 S294->T4 S294->T4 S315->T4
[41] S352->T4 S352->T4 S14 ->T4 S25 ->T4 S46 ->T4 S55 ->T4 S153->T4 S223->T4
[49] S356->T4 S423->T4 S426->T4 S54 ->T5 S111->T5 S217->T5 S428->T5 S93 ->T6
[57] S23 ->T6 S46 ->T6 S106->T6 S122->T6 S181->T6 S264->T6 S321->T6 S383->T6
+ ... omitted several edges
```

```
'S272'  'S282'  'S336'  'S344'  'S242'  'S315'  'S55'  'S59'  'S149'  'S153'  'S356'  'S426'  'S14'  'S223'  'S25'  'S46'  'S169'  'S363'
'S423'  'S224'  'S135'  'S261'  'S294'  'S145'  'S162'  'S166'  'S196'  'S219'  'S237'  'S352'  'S54'  'S111'  'S217'  'S428'  'S93'  'S23'
'S106'  'S122'  'S181'  'S264'  'S321'  'S383'  'S49'  'S185'  'S230'  'S410'  'S240'  'S188'  'S228'  'S258'  'S26'  'S66'  'S109'  'S113'
'S182'  'S208'  'S235'  'S251'  'S381'  'S334'  'S3'  'S21'  'S36'  'S40'  'S52'  'S105'  'S119'  'S129'  'S176'  'S275'  'S276'  'S277'
'S287'  'S300'  'S311'  'S314'  'S341'  'S345'  'S407'  'S414'  'S10'  'S61'  'S67'  'S78'  'S89'  'S152'  'S165'  'S198'  'S203'  'S225'
'S231'  'S304'  'S354'  'S364'  'S380'  'S385'  'S404'  'S411'  'S42'  'S243'
```

```
gubun
  S   T
432  94
```

In [27]:

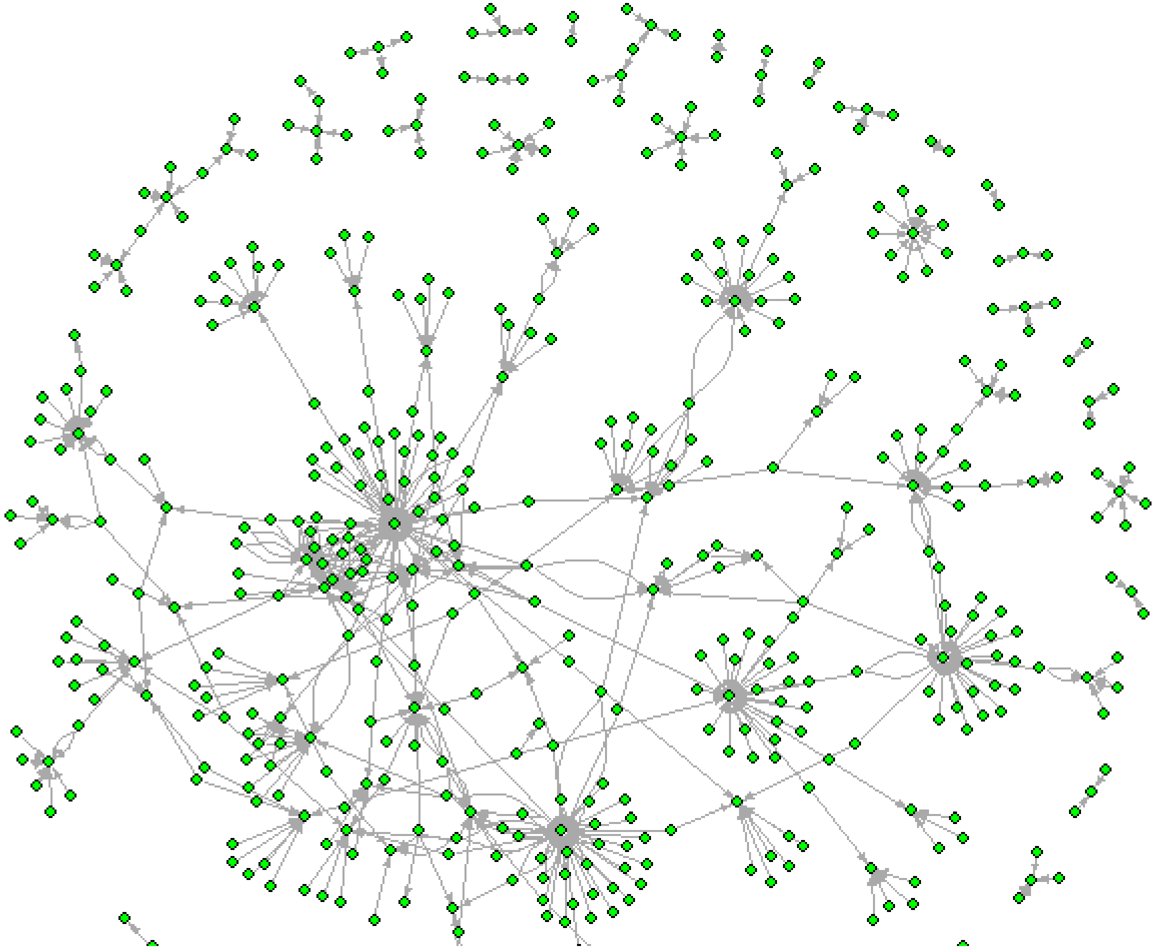
```
plot(df)
```

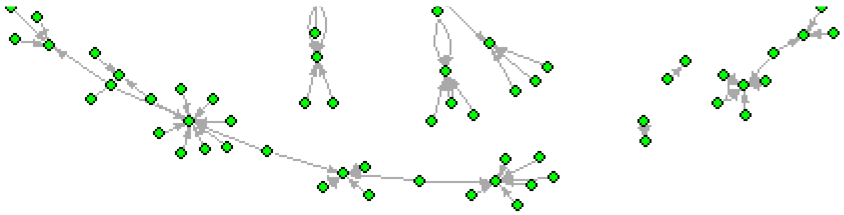


```
In [ ]:
```

In [32]:

```
plot(df, layout=layout.fruchterman.reingold,
     vertex.size=2,            # 점의 크기
     edge.arrow.size=0.2,      # 화살표 크기
     vertex.color = "green",   # 점의 색깔
     vertex.label=NA)
```





학생과 교수의 색상과 크기를 구분해서 출력

In [44]:

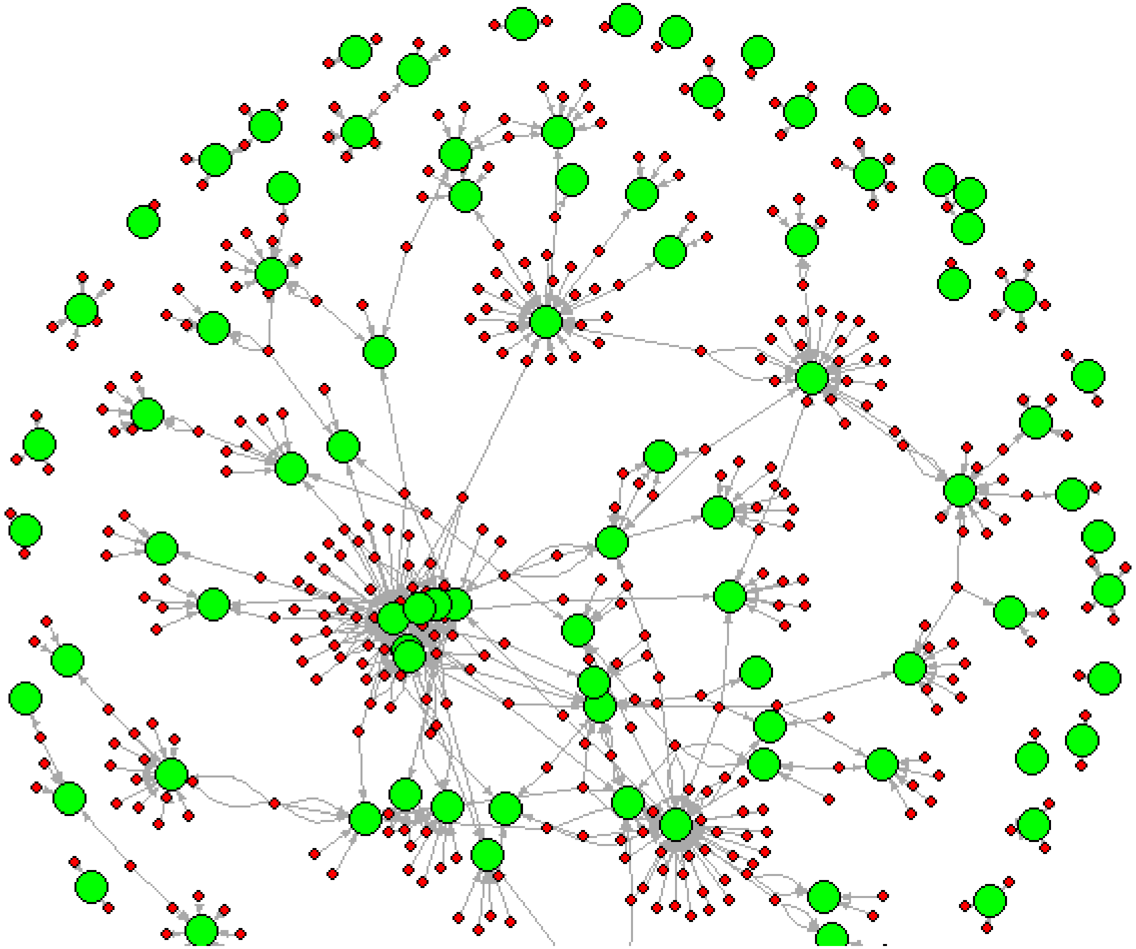
```r
colors <- c()
for (i in 1:length(gubun)) {
    if (gubun[i]=='S') {
        colors <- c(colors, "red")
    } else {
        colors <- c(colors, "green")
    }
}
```

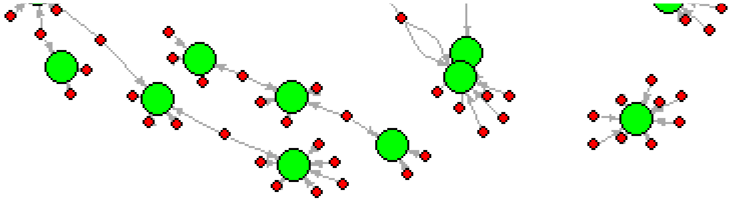
In [45]:

```r
sizes <- c()
for (i in 1:length(gubun)) {
    if (gubun[i]=='S') {
        sizes <- c(sizes,2)
    } else {
        sizes <- c(sizes,6)
    }
}
```

In [47]:

```
plot(df, layout=layout.fruchterman.reingold,
     vertex.size=sizes,            # 점의 크기
     edge.arrow.size=0.2,       # 화살표 크기
     vertex.color = colors,    # 점의 색깔
     vertex.label=NA)
```





In [1]:

```
# install.packages("networkD3")
# install.packages("d3Network")
```

```
Installing package into 'C:/Users/WITHJS/Documents/R/win-library/3.4'
(as 'lib' is unspecified)

package 'networkD3' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:₩Users₩WITHJS₩AppData₩Local₩Temp₩Rtmp00DHrv₩downloaded_packages

Installing package into 'C:/Users/WITHJS/Documents/R/win-library/3.4'
(as 'lib' is unspecified)

package 'd3Network' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:₩Users₩WITHJS₩AppData₩Local₩Temp₩Rtmp00DHrv₩downloaded_packages
```

In [2]:

```
library("networkD3")
library("d3Network")
```

In [3]:

```
source = c("A", "A", "A", "A", "B", "B", "B", "C", "C")
target = c("B", "C", "D", "J", "E", "F", "G", "H", "I")

networkData = data.frame(source, target)
networkData
```

source	target
A	B
A	C
A	D
A	J
B	E
B	F
B	G
C	H
C	I

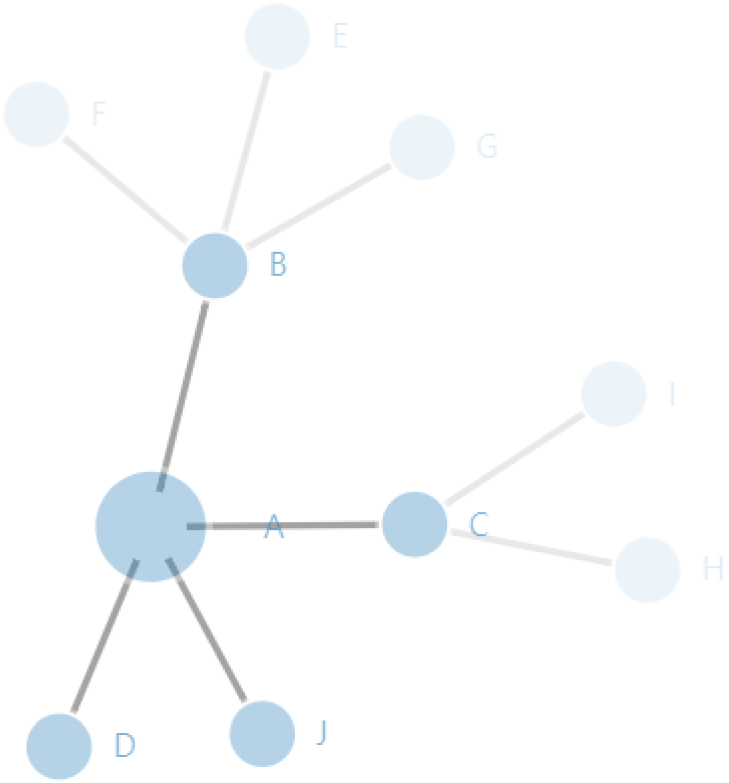
간단한 D3 자바스크립트 강제 직접 네트워크 그래프 생성 함수

In [5]:

```
simpleNetwork(networkData, fontFamily = "sans-serif")
```

```
Warning message in readChar(htmlfile, file.info(htmlfile)$size):
"UTF-8가 아닌 MBCS 로케일에서만 바이트로만 읽을 수 있습니다"
```

R Studio로 확인한 결과 이미지



실습 01

Facebook친구 목록을 확보했다.
철수는 영희, 민희, 수철이를 친구로 가지고 있고,
수철이는 수진, 철수, 희수를 친구를 가지고 있다.
기수는 영희, 수철이를 친구로 가지고 있다.
위에서 배운 내용을 이용하여 네트워크 시각화를 시켜보자.

어떤 친구가 다른 친구를 다 연결하는 친구가 되는가?

확인한 내용을 GitHub에 올려보자.

In [ ]: