Homework 4. Part 2

Advanced Network Analytics and Modeling

Aleksey Shadrunov, Anastasia Podolskaya, Maksim Emelyanenko

2023-06-08

Contents

Prepare packages and load data

Install packages (uncomment if necessary):

```
# install.packages("network", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("sna", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("igraph", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("intergraph", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("knitr", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("RColorBrewer", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("bipartite", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("dendextend", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("blockmodeling", repos = "https://mirror.truenetwork.ru/CRAN/")
# install.packages("multinet", repos = "https://mirror.truenetwork.ru/CRAN/")
```

```
suppressPackageStartupMessages(library(RColorBrewer))
suppressPackageStartupMessages(library(bipartite))
suppressPackageStartupMessages(library(network))
suppressPackageStartupMessages(library(dendextend))
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(stringr))
```

The given dataset is in CSV format, which is a table with columns: Entity.A, Entity.B, Entity.A.Type, Entity.B.Type, Connection. As we have already looked through the net, we can clean up the data a bit.

```
Trump <- read.csv("/home/alex/Documents/year-3-sna/hw4/trumpworld.csv")

ncol(Trump)
```

```
## [1] 6
```

```
nrow(Trump)
```

```
## [1] 3380
```

```
Trump <- Trump[-c(6)]

Trump$Connection <- Trump$Connection %>% str_replace(" \\(as of 2016 FEC filing\\)", "")
Trump$Connection <- Trump$Connection %>% str_replace(" \\(as of 2014 FEC filing\\)", "")
Trump$Connection <- Trump$Connection %>% str_replace("Former director", "Director")
Trump$Connection <- Trump$Connection %>% str_replace("Former member", "Member")
Trump$Connection <- Trump$Connection %>% str_replace("Reported member", "Member")
Trump$Connection <- Trump$Connection %>% str_replace("Special assistant", "Assistant")
Trump$Connection <- Trump$Connection %>% str_replace("Deputy assistant", "Assistant")
Trump$Connection <- Trump$Connection %>% str_trim()
```

Create networks

Next, we want to select several Connection types to create a multiplex networks. Here we see the most common connections:

```r
cons <- data.frame(table(Trump$Connection))
kable(head(cons[order(cons$Freq, decreasing = TRUE), ], 25))
```

	Var1	Freq
654	President	495
624	Ownership	337
231	Director	197
562	Member	195
498	Investor	120
526	Listed as asset/income on Public Financial Disclosure Report	63
743	Subsidiary	55
626	Owns collateralized debt	54
767	Trustee	49
668	Provided legal services for	48
273	Donor	46
121	Board member	40
199	Consultant	39
605	NKA/FKA	36
152	Chairman	32
449	Founder	31
628	Parent/child	30
91	Assistant	28
183	Co-founder	28
308	Former board member	28
632	Partner	25
615	Official campaign committee	23
402	Former president	22
555	Managing member	22
722	Siblings	20

Also we may select rows in each type of entities.

Organization to Organization

```r
b2b <- Trump[Trump$Entity.A.Type == "Organization", ]
b2b <- b2b[b2b$Entity.B.Type == "Organization", ]

b2b_cons <- data.frame(table(b2b$Connection))
kable(head(b2b_cons[order(b2b_cons$Freq, decreasing = TRUE), ], 10))
```

	Var1	Freq
63	Ownership	336
42	Investor	93
88	Subsidiary	55
64	Owns collateralized debt	54
36	Donor	41
61	NKA/FKA	36
33	DBA	17
51	Loaned money	12

	Var1	Freq
56	Member	8
58	Mortgage	6

We may select top 2 for further analysis.

```
b2b_top_count <- 2
b2b_cons_selected <- as.vector(head(b2b_cons[order(b2b_cons$Freq, decreasing = TRUE), ], b2b_top_count)$
b2b_cons_selected
```

```
## [1] "Ownership" "Investor"
```

```
b2b_selected <- subset(b2b, Connection %in% b2b_cons_selected)

b2b_A_selected <- as.vector(unique(b2b_selected$Entity.A))
b2b_B_selected <- as.vector(unique(b2b_selected$Entity.B))
```

Person to Person

```
c2c <- Trump[Trump$Entity.A.Type == "Person", ]
c2c <- c2c[c2c$Entity.B.Type == "Person", ]

c2c_cons <- data.frame(table(c2c$Connection))
kable(head(c2c_cons[order(c2c_cons$Freq, decreasing = TRUE), ], 10))
```

	Var1	Freq
177	Parent/child	30
42	Assistant	27
216	Siblings	20
156	Married	17
161	Member of Trump's Strategic and Policy forum	7
60	Business partners	6
129	Friends	5
13	"Leader" in the transition team per CNN	4
33	A top touchstone outside the White House, according to the New York Times	3
213	Senior associate counsel	3

We may select top 4 for further analysis.

```
c2c_top_count <- 4
c2c_cons_selected <- as.vector(head(c2c_cons[order(c2c_cons$Freq, decreasing = TRUE), ], c2c_top_count)$
c2c_cons_selected
```

```
## [1] "Parent/child" "Assistant"    "Siblings"     "Married"
```

```
c2c_selected <- subset(c2c, Connection %in% c2c_cons_selected)

c2c_A_selected <- as.vector(unique(c2c_selected$Entity.A))
c2c_B_selected <- as.vector(unique(c2c_selected$Entity.B))
```

Person or Organization

```
b2c <- Trump[Trump$Entity.A.Type == "Organization", ]
b2c <- b2c[b2c$Entity.B.Type == "Person", ]
c2b <- Trump[Trump$Entity.A.Type == "Person", ]
c2b <- c2b[c2b$Entity.B.Type == "Organization", ]
mix <- rbind(b2c, c2b)

mix_cons <- data.frame(table(mix$Connection))
kable(head(mix_cons[order(mix_cons$Freq, decreasing = TRUE), ], 10))
```

	Var1	Freq
413	President	495
118	Director	196
365	Member	187
349	Listed as asset/income on Public Financial Disclosure Report	63
467	Trustee	49
424	Provided legal services for	47
44	Board member	40
103	Consultant	39
64	Chairman	32
296	Founder	31

We may select top 3 for further analysis.

```
mix_top_count <- 3
mix_cons_selected <- as.vector(head(mix_cons[order(mix_cons$Freq, decreasing = TRUE), ], mix_top_count)$
mix_cons_selected
```

```
## [1] "President" "Director"  "Member"
```

```
mix_selected <- subset(mix, Connection %in% mix_cons_selected)

mix_A_selected <- as.vector(unique(mix_selected$Entity.A))
mix_B_selected <- as.vector(unique(mix_selected$Entity.B))
```

Now we can combine all subsets together:

```
all_selected <- rbind(b2b_selected, c2c_selected, mix_selected)
all_actors_selected <- unique(c(all_selected$Entity.A, all_selected$Entity.B))
```

For creating 3-mode network (Person, Organization, Federal Agency), we should add later rows to the dataframe. Alse we may filter only those actors who are included in the rest of the network.

```
fa <- Trump[Trump$Entity.B.Type == "Federal Agency", ]
fa_selected <- fa[fa$Entity.A %in% all_actors_selected, ]
```

Multiplex network analysis

Now we may create multiplex network for each type of actors: Persons or Organization.

Organization to Organization

Plot

```
suppressPackageStartupMessages(library(multinet))

b2bm <- ml_empty()
# add two directed layers
add_layers_ml(b2bm, b2b_cons_selected, rep(c(TRUE), length(b2b_cons_selected)))
is_directed_ml(b2bm)
```

```
##      layer1    layer2 dir
## 1 Ownership Ownership   1
## 2  Investor  Investor   1
```

```
# add actors (repeat each actor on each layer)
b2b_actors_selected <- unique(c(b2b_A_selected, b2b_B_selected))
actors <- rep(b2b_actors_selected, length(b2b_cons_selected))
layers <- c()
for (con in b2b_cons_selected) {
    layers <- c(layers, rep(con, length(b2b_actors_selected)))
}
vertices <- data.frame(actors = actors, layers = layers)
add_vertices_ml(b2bm, vertices)

# add arcs
actors_from <- c()
layers_from <- c()
actors_to <- c()
layers_to <- c()
b2b_selected_for_layer <- c()

for (layer in b2b_cons_selected) {
    b2b_selected_for_layer <- b2b_selected[b2b_selected$Connection == layer, ]
    actors_from <- c(actors_from, b2b_selected_for_layer$Entity.A)
    actors_to <- c(actors_to, b2b_selected_for_layer$Entity.B)
    layers_from <- c(layers_from, b2b_selected_for_layer$Connection)
    layers_to <- c(layers_to, b2b_selected_for_layer$Connection)
}

edges <- data.frame(
    actors_from = actors_from,
    layers_from = layers_from,
    actors_to = actors_to,
    layers_to = layers_to
)

add_edges_ml(b2bm, edges)
```

Now we can plot the two-layer network to take a first glance at its properties (Figure 1).

```
suppressWarnings({
    par(mar = c(0, 0, 0, 0))
    l <- layout_multiforce_ml(b2bm, w_inter = 100, gravity = 100, iter = 100)
```

```
    plot(b2bm,
        layout = l,
        grid = c(2, 1),
        vertex.labels = "",
        vertex.labels.cex = 0.3, vertex.labels.col = "grey",
        vertex.color = "orange",
        vertex.cex = 0.5, edge.col = "grey",
    )
})
```

What we see is a very unconcise picture of the network. As both layers are little interconnected, we may plot the separately.

```
layers <- as.list(b2bm)
names(layers)
```

```
## [1] "_flat_"    "Investor"  "Ownership"
```

```
opar <- par(mfrow = c(2, 1))
par(mar = c(0, 0, 0, 0))

plot(
    layers[[2]],
    vertex.size = 6,
    edge.arrow.size = 0.2,
    edge.color = "grey",
    vertex.label.cex = 0.3,
    vertex.color = "orange",
    vertex.shape = "circle",
    vertex.frame.color = "grey",
    vertex.label.dist = 0,
    vertex.label.color = "blue"
)
plot(
    layers[[3]],
    vertex.size = 4,
    edge.arrow.size = 0.2,
    edge.color = "grey",
    vertex.label.cex = 0.2,
    vertex.color = "green",
    vertex.shape = "circle",
    vertex.frame.color = "grey",
    vertex.label.dist = 0,
    vertex.label.color = "blue"
)
```

Now the picture is clearer and we can observe, that in both layers there are no isolated vertices. The first layer (Investor) presents one central actor which is apparently a huge investing company (the name is THRIVE CAPITAL). The second layer (Ownership), in contrast, have many connected components, which are separated from each other. The most central vertices (with highest degree) are DJT HOLDINGS LLC, IT OPERATIONS MANAGING MEMBER CORP. So, we may suggest, that it is more common when one company invests in many others and not when one company owns many others.

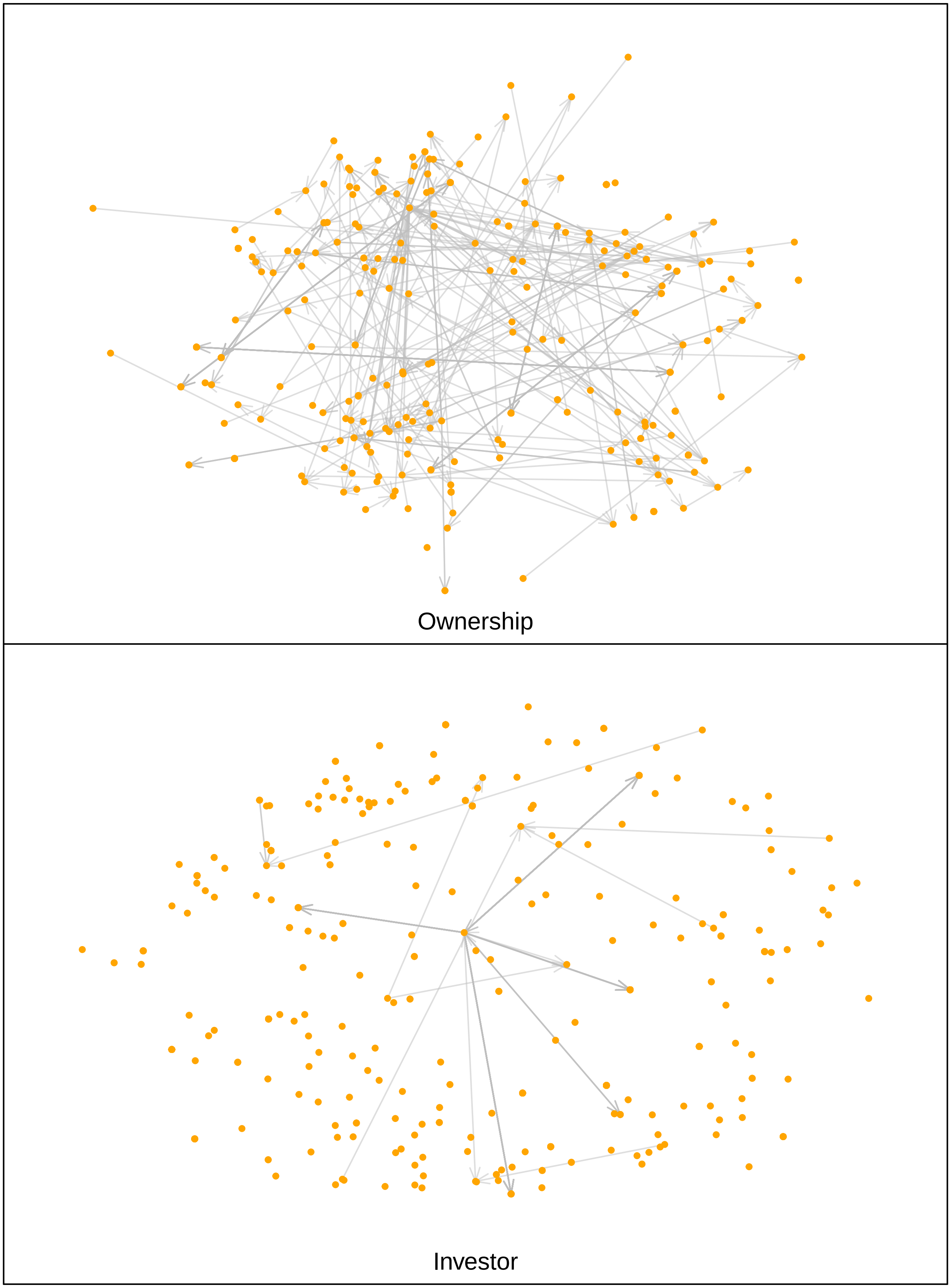


Figure 1: b2b Visualisation

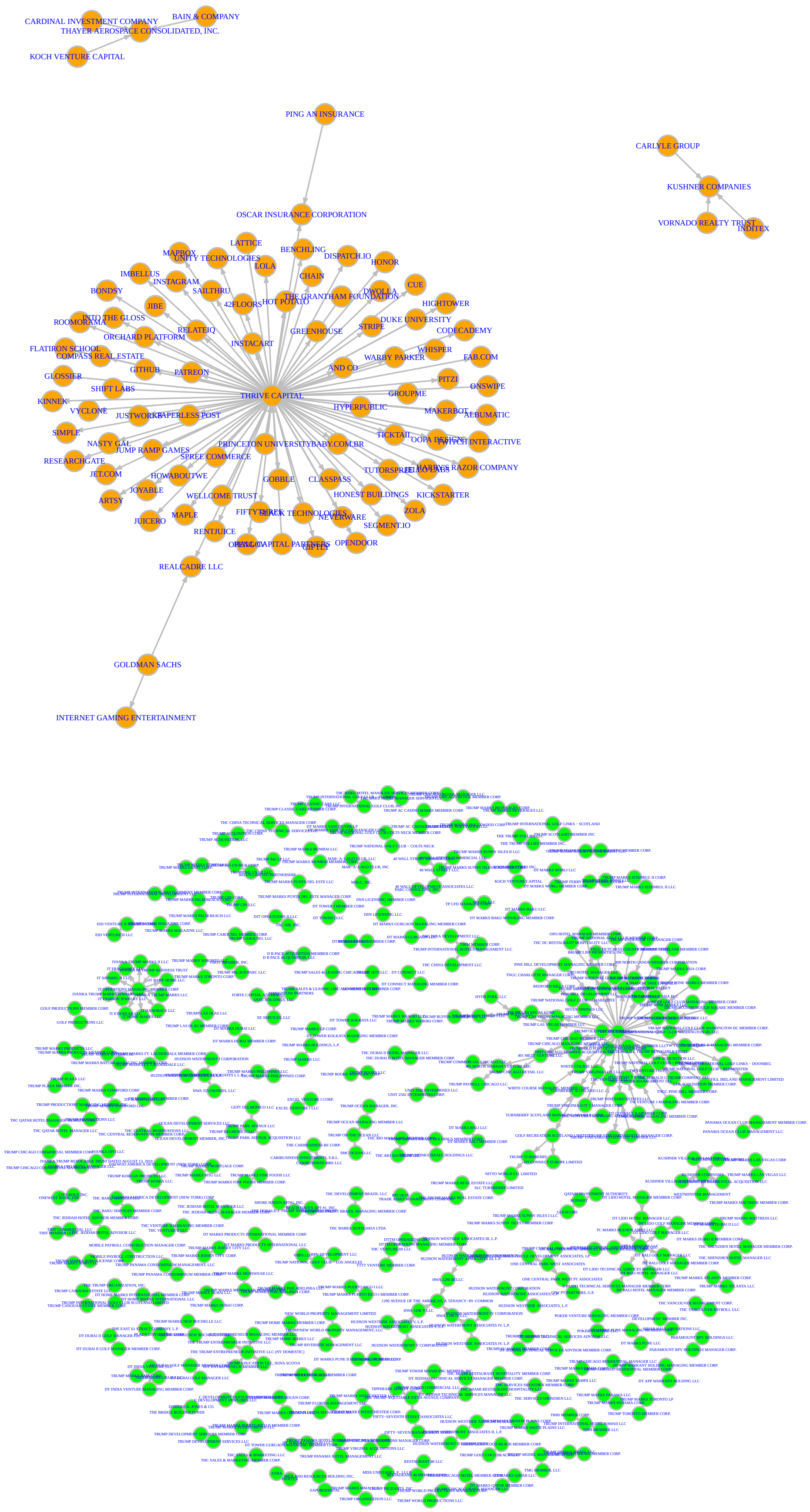


Figure 2: b2b Separate Visualisation

Statistics

Simple summary command provides us with the following Statistics: * n - number of vertices * m - number of edges * dir - directionality * nc - number of connected components (subgraphs in which any two vertices are connected to each other by paths. In case of directed layers, edge directionality will be taken into account, i.e. the number of strong components) * dens - density (the ratio between the number of edges and the number of possible edges) * cc - clustering coefficient (the ratio between the triangles and the connected triples in the layer) * apl - average path length (the average graph-distance between all pairs of vertices in the layer) * dia - diameter (the longest graph distance between any two vertices in the layer)

```
summary(b2bm)
```

```
##              n   m dir  nc slc        dens          cc      apl dia
## _flat_     592 429   1 170  88 0.001226163 0.001390821 1.685221   5
## Investor    96  93   1   3  88 0.010197368 0.000000000 1.809426   2
## Ownership 499 336   1 170  83 0.001352102 0.007481297 1.563197   5
```

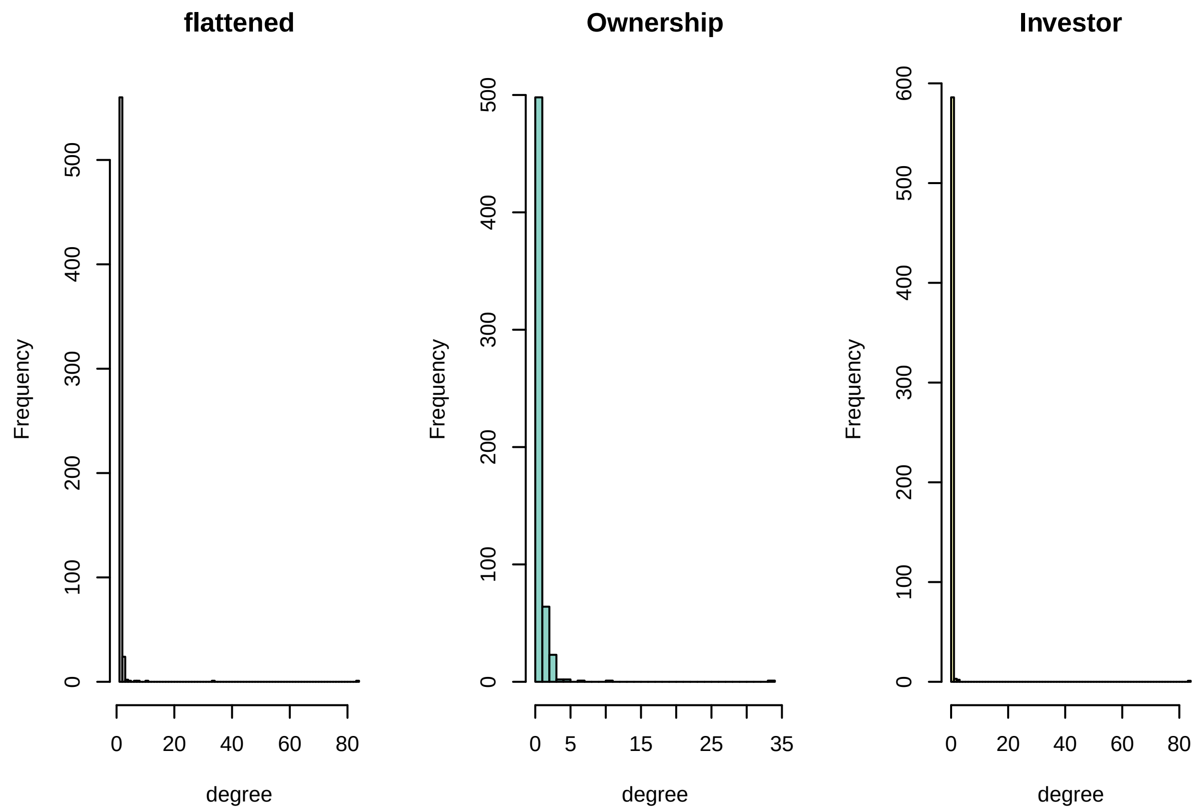
So, we see that the first layer has only 3 connected components, while the second — 170. However, the density of Investor is 10 times more than the density of Ownerships (much more ties could theoretically be created in the second layer). Curious value are assigned to the longest distance in two layers. The second one has the distance of 5, which means, that there is an ownership chain consisted of 5 companies.

Degree We continue with analysing degree for each layer.

```
color_map <- brewer.pal(num_layers_ml(b2bm), "Set3")
```

```
## Warning in brewer.pal(num_layers_ml(b2bm), "Set3"): minimal value for n is 3, returning requested pal
```

```
opar <- par(mfrow = c(1, 3))
hist(degree_ml(b2bm),
    breaks = max(degree_ml(b2bm)),
    main = "flattened",
    xlab = "degree"
)
for (i in 1:num_layers_ml(b2bm)) {
    d <- degree_ml(b2bm, layers = layers_ml(b2bm)[[i]])
    hist(d,
        breaks = max(d, na.rm = TRUE), main = layers_ml(b2bm)[[i]],
        xlab = "degree", col = color_map[i]
    )
}
```




```r
# flatten
kable(t(table(degree(layers[[1]]))), caption = "Flatten: Degree distribution")
```

Table 5: Flatten: Degree distribution

1	2	3	4	5	7	8	11	34	84
491	69	24	2	1	1	1	1	1	1

```r
# Investor
kable(t(table(degree(layers[[2]]))), caption = "Investor: Degree distribution")
```

Table 6: Investor: Degree distribution

1	2	3	84
90	3	2	1

```r
# Ownership
kable(t(table(degree(layers[[3]]))), caption = "Ownership: Degree distribution")
```

Table 7: Ownership: Degree distribution

1	2	3	4	5	7	11	34
405	64	23	2	2	1	1	1

We have already observed that most actors are of degree 1, and few possess an outstanding value (84 or 34). Let us look at those vertices:

```
# Investor
d <- data.frame(degree(layers[[2]]))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Investor: Degree top")
```

Table 8: Investor: Degree top

	Actor	Value
22	THRIVE CAPITAL	84
71	THAYER AEROSPACE CONSOLIDATED, INC.	3
77	KUSHNER COMPANIES	3
34	GOLDMAN SACHS	2
41	OSCAR INSURANCE CORPORATION	2

```
# Ownership
d <- data.frame(degree(layers[[3]]))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Ownership: Degree top")
```

Table 9: Ownership: Degree top

	Actor	Value
346	DJT HOLDINGS LLC	34
78	IT OPERATIONS MANAGING MEMBER CORP	11
477	TTTT VENTURE LLC	7
205	401 MEZZ VENTURE LLC	5
336	KUSHNER COMPANIES	5

Layer comparison We may also compare layers. However, in our case, they are not very related, so most of statistics are not useful.

We start with Pearson correlation coefficient. The smallest value (-1) indicates that high-degree actors in one layer are low-degree in the other, while the largest value (1) is returned if high-degree actors in one layer are high-degree actors in the other.

```
layer_comparison_ml(b2bm, method = "pearson.degree")
```

```
##             Ownership   Investor
## Ownership  1.00000000 -0.05915506
## Investor  -0.05915506  1.00000000
```

We see that correlation is about zero, which is expected because the actors in both layers are different.

Dyad For each layer we may calculate dyad statistics:

```
Investor <- as.igraph(b2bm, "Investor")
Ownership <- as.igraph(b2bm, "Ownership")

# types of dyads
dyad.census(Investor)
```

```
## $mut
## [1] 0
##
## $asym
## [1] 93
##
## $null
## [1] 174843
```

```
dyad.census(Ownership)
```

```
## $mut
## [1] 0
##
## $asym
## [1] 336
##
## $null
## [1] 174600
```

Dyad census says that no mutual ties are in the Ivestor layer. Ownership layer behaves similarly, but has more arcs.

```
Triads <- c("003", "012", "102", "021D", "021U", "021C", "111D", "111U", "030T", "030C", "201", "120D",
Number <- triad.census(Investor)
kable(t(data.frame(Triads, Number)), caption = "Investor: triads")
```

Triads

Table 10: Investor: triads

Triads	003	012	102	021D	021U	021C	111D	111U	030T	030C	201	120D	120U	120C	210	300
Number	34352705	47880	0	3082	18	395	0	0	0	0	0	0	0	0	0	0

```
Number <- triad.census(Ownership)
kable(t(data.frame(Triads, Number)), caption = "Ownership: triads")
```

Table 11: Ownership: triads

Triads	003	012	102	021D	021U	021C	111D	111U	030T	030C	201	120D	120U	120C	210	300
Number	34206640	196642	0	587	69	140	0	0	2	0	0	0	0	0	0	0

In Investor layer, there are plenty of A<-B->C (outcoming) triads, some A->B<-C triads, also there are transitive triads A->B->C. Ownership layer is similar, too.

Transitivity Let's calculate trads statics:

```
transitivity(Investor)
```

```
## [1] 0
```

```
transitivity(Ownership)
```

```
## [1] 0.007481297
```

In both layers transitivity is about zero, as there are not many connected ties.

```
reciprocity(Investor)
```

Reciprocity

```
## [1] 0
```

```
reciprocity(Ownership)
```

```
## [1] 0
```

In both layers reciprocity is zeroˆ there are no mutual arcs.

Centralitiy It makes sense to find the most central actors for each type of centrality. Probably the most-ranked vertices will be the same.

```
# Investor
d <- data.frame(degree(Investor, mode = "in"))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Investor: Indegree centrality")
```

Table 12: Investor: Indegree centrality

	Actor	Value
186	THRIVE CAPITAL	5
381	THAYER AEROSPACE CONSOLIDATED, INC.	3
410	KUSHNER COMPANIES	3
262	OSCAR INSURANCE CORPORATION	2

	Actor	Value
515	REALCADRE LLC	2

```
d <- data.frame(degree(Investor, mode = "out"))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Investor: Outdegree centrality")
```

Table 13: Investor: Outdegree centrality

	Actor	Value
186	THRIVE CAPITAL	79
244	GOLDMAN SACHS	2
16	THE GRANTHAM FOUNDATION	1
18	INDITEX	1
56	VORNADO REALTY TRUST	1

```
d <- data.frame(degree(Investor, mode = "total"))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Investor: Total degree centrality")
```

Table 14: Investor: Total degree centrality

	Actor	Value
186	THRIVE CAPITAL	84
381	THAYER AEROSPACE CONSOLIDATED, INC.	3
410	KUSHNER COMPANIES	3
244	GOLDMAN SACHS	2
262	OSCAR INSURANCE CORPORATION	2

```
d <- data.frame(closeness(Investor))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Investor: Closeness centrality")
```

Table 15: Investor: Closeness centrality

	Actor	Value
18	INDITEX	1
56	VORNADO REALTY TRUST	1
210	KOCH VENTURE CAPITAL	1
358	BAIN & COMPANY	1
392	CARDINAL INVESTMENT COMPANY	1

Actor	Value

```r
# Ownership
d <- data.frame(degree(Ownership, mode = "in"))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 7), caption = "Ownership: Indegree centrality")
```

Table 16: Ownership: Indegree centrality

	Actor	Value
232	401 MEZZ VENTURE LLC	3
363	TRUMP RUFFIN LLC	3
506	TRUMP PARK AVENUE LLC	3
526	TRUMP KOREA LLC	3
553	1290 AVENUE OF THE AMERICAS, A TENANCY-IN-COMMON	3
54	FIFTY-SEVENTH STREET ASSOCIATES LLC	2
63	ONE CENTRAL PARK WEST ASSOCIATES	2

```r
d <- data.frame(degree(Ownership, mode = "out"))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Ownership: Outdegree centrality")
```

Table 17: Ownership: Outdegree centrality

	Actor	Value
421	DJT HOLDINGS LLC	32
80	IT OPERATIONS MANAGING MEMBER CORP	10
570	TTTT VENTURE LLC	7
410	KUSHNER COMPANIES	5
103	THE DONALD J. TRUMP REVOCABLE TRUST	3

```r
d <- data.frame(degree(Ownership, mode = "total"))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Ownership: Total degree centrality")
```

Table 18: Ownership: Total degree centrality

	Actor	Value
421	DJT HOLDINGS LLC	34
80	IT OPERATIONS MANAGING MEMBER CORP	11
570	TTTT VENTURE LLC	7

	Actor	Value
232	401 MEZZ VENTURE LLC	5
410	KUSHNER COMPANIES	5

```
d <- data.frame(closeness(Ownership))
d <- cbind(newColName = rownames(d), d)
colnames(d) <- c("Actor", "Value")
rownames(d) <- 1:nrow(d)
kable(head(d[order(d$Value, decreasing = TRUE), ], 5), caption = "Ownership: Closeness centrality")
```

Table 19: Ownership: Closeness centrality

Actor	Value
THC CENTRAL RESERVATIONS MEMBER CORP.	1
THC DUBAI II HOTEL MANAGER MEMBER CORP.	1
THC QATAR HOTEL MANAGER MEMBER CORP.	1
THC JEDDAH HOTEL MANAGER MEMBER CORP.	1
SENTIENT JETS MEMBER CORP.	1

We may notice some centralities are consistent with previous top positions, and some are not.

```
centralization.degree(Investor)$centralization
```

Centralization

```
## [1] 0.07091998
```

```
centralization.degree(Ownership)$centralization
```

```
## [1] 0.0278515
```

The graph level centrality index is the last metric to look at. It demonstrates the measure of centrality of the whole graph, and in our case the Investor layer has a greater value. It is proven by visualisations we made previously.

Summary

The first multiplex network from Trumpworld data consists of two layers. Each of them has different structure: Investor layer is very simple with 3 components, one of them comprises almost all vertices. It is a very central actor which is connected to many other, however, only in one direction. The second layer has many small components and several comparatively large, which is built similarly to the one from Investor layer. In this component we have found a path of length 5. However, such structures are less common for layer 2.

Person to Person

Plot

```r
c2cm <- ml_empty()
# add four directed layers
add_layers_ml(c2cm, c2c_cons_selected, rep(c(TRUE), length(c2c_cons_selected)))
is_directed_ml(c2cm)
```

```
##         layer1       layer2 dir
## 1 Parent/child Parent/child   1
## 2    Assistant    Assistant   1
## 3      Married      Married   1
## 4     Siblings     Siblings   1
```

```r
# add actors (repeat each actor on each layer)
c2c_actors_selected <- unique(c(c2c_A_selected, c2c_B_selected))
actors <- rep(c2c_actors_selected, length(c2c_cons_selected))
layers <- c()
for (con in c2c_cons_selected) {
    layers <- c(layers, rep(con, length(c2c_actors_selected)))
}
vertices <- data.frame(actors = actors, layers = layers)
add_vertices_ml(c2cm, vertices)

# add arcs
actors_from <- c()
layers_from <- c()
actors_to <- c()
layers_to <- c()
c2c_selected_for_layer <- c()

for (layer in c2c_cons_selected) {
    c2c_selected_for_layer <- c2c_selected[c2c_selected$Connection == layer, ]
    actors_from <- c(actors_from, c2c_selected_for_layer$Entity.A)
    actors_to <- c(actors_to, c2c_selected_for_layer$Entity.B)
    layers_from <- c(layers_from, c2c_selected_for_layer$Connection)
    layers_to <- c(layers_to, c2c_selected_for_layer$Connection)
}

edges <- data.frame(
    actors_from = actors_from,
    layers_from = layers_from,
    actors_to = actors_to,
    layers_to = layers_to
)

add_edges_ml(c2cm, edges)
```

Now we can plot the 4-layer network to take a first glance at its properties (Figure 3).

```r
suppressWarnings({
    par(mar = c(0, 0, 0, 0))
    l <- layout_multiforce_ml(c2cm, w_inter = 100, gravity = 100, iter = 100)
    plot(c2cm,
        layout = l,
        grid = c(2, 2),
```

```
        vertex.labels = "",
        vertex.labels.cex = 0.3, vertex.labels.col = "grey",
        vertex.color = "purple",
        vertex.cex = 0.5, edge.col = "grey",
    )
})
```

On contrary with previous network, we see different pictures here. The most comprehensive is the Assistant layer. Here we may tell that there is again a very central actor which connects to many other in one-way manner.

To make picture clearer, we should perform separate visualisations.

```
layers <- as.list(c2cm)
names(layers)
```

```
## [1] "_flat_"       "Assistant"    "Married"      "Parent/child" "Siblings"
```

```
opar <- par(mfrow = c(2, 2))
par(mar = c(0, 0, 0, 0))

plot(
    layers[[2]],
    vertex.size = 8,
    edge.arrow.size = 0.3,
    edge.color = "grey",
    vertex.label.cex = 0.3,
    vertex.color = "orange",
    vertex.shape = "circle",
    vertex.frame.color = "grey",
    vertex.label.dist = 0,
    vertex.label.color = "blue"
)
plot(
    layers[[3]],
    vertex.size = 8,
    edge.arrow.size = 0.3,
    edge.color = "grey",
    vertex.label.cex = 0.3,
    vertex.color = "green",
    vertex.shape = "circle",
    vertex.frame.color = "grey",
    vertex.label.dist = 0,
    vertex.label.color = "blue"
)
plot(
    layers[[4]],
    vertex.size = 8,
    edge.arrow.size = 0.3,
    edge.color = "grey",
    vertex.label.cex = 0.3,
    vertex.color = "magenta",
    vertex.shape = "circle",
    vertex.frame.color = "grey",
```

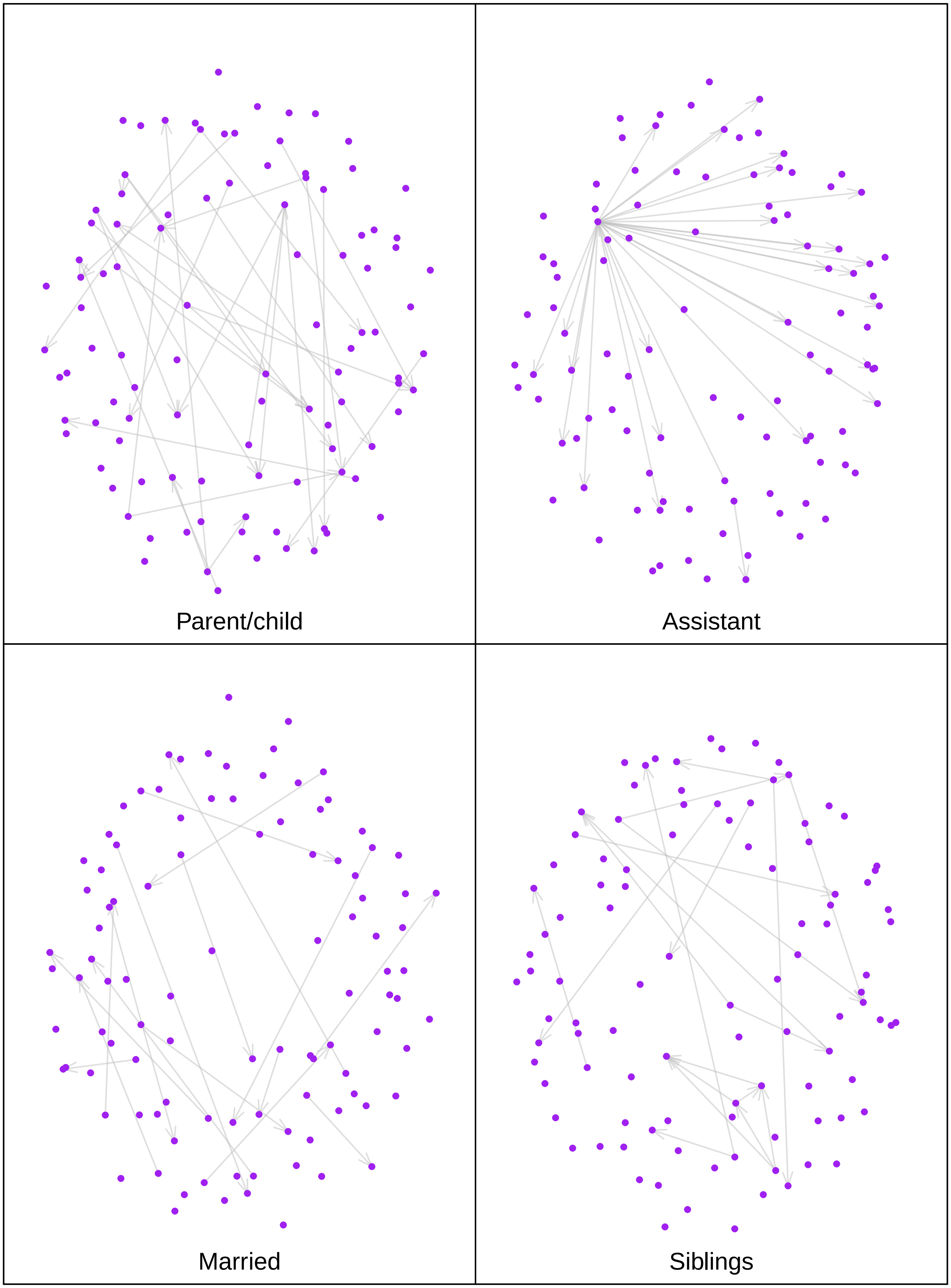


Figure 3: c2c Visualisation

```r
    vertex.label.dist = 0,
    vertex.label.color = "blue"
)
plot(
    layers[[5]],
    vertex.size = 8,
    edge.arrow.size = 0.3,
    edge.color = "grey",
    vertex.label.cex = 0.3,
    vertex.color = "red",
    vertex.shape = "circle",
    vertex.frame.color = "grey",
    vertex.label.dist = 0,
    vertex.label.color = "blue"
)
```

Now the picture is clearer (Figure 4).

The first graph (Assistant) is a very central one, with one actor (DONALD J. TRUMP) being in the centre. The single other component consists of two vertices, not linked to main subgraph (JARED KUSHNER, AVRAHM BERKOWITZ).

The second one (Married) consist only from pairs of married people. Here we observe only not reciprocal arcs. Due to its nature, the network is very unconnected, as there is no path longer 2 and no vertex with degree more than 1. So, no one had a divorce according to this picture.

The third one (Parent/child) consists of parents and their children. Here we find various figures, even consisting of 4 actors.

The fourth one (Siblings) is an interesting one. Here we have highly coupled graphs, almost a clique, and also triads.

Statistics

Simple summary gives us the following:

```r
summary(c2cm)
```

```
##                 n  m dir nc slc        dens        cc      apl dia
## _flat_        102 94   1 24  44 0.009124442 0.1089965 2.047847   6
## Assistant      29 27   1  2  27 0.033251232 0.0000000 1.480769   2
## Married        34 17   1 17   2 0.015151515       NaN 1.000000   1
## Parent/child   44 30   1 16   6 0.015856237 0.0000000 1.090909   2
## Siblings       24 20   1  9   4 0.036231884 0.9000000 1.000000   1
```

So, we see that the Assistant layer has 2 connected components, while the others have more — from 9 to 17. All layers have more or less similar density (0.015-0.036). The average path length and diameter are not significant in every layer.

Degree We continue with analysing degree for each layer.

```r
color_map <- brewer.pal(num_layers_ml(c2cm), "Set3")
opar <- par(mfrow = c(2, 2))
for (i in 1:num_layers_ml(c2cm)) {
    d <- degree_ml(c2cm, layers = layers_ml(c2cm)[[i]])
    hist(d,
        breaks = max(d, na.rm = TRUE), main = layers_ml(c2cm)[[i]],
        xlab = "degree", col = color_map[i]
```

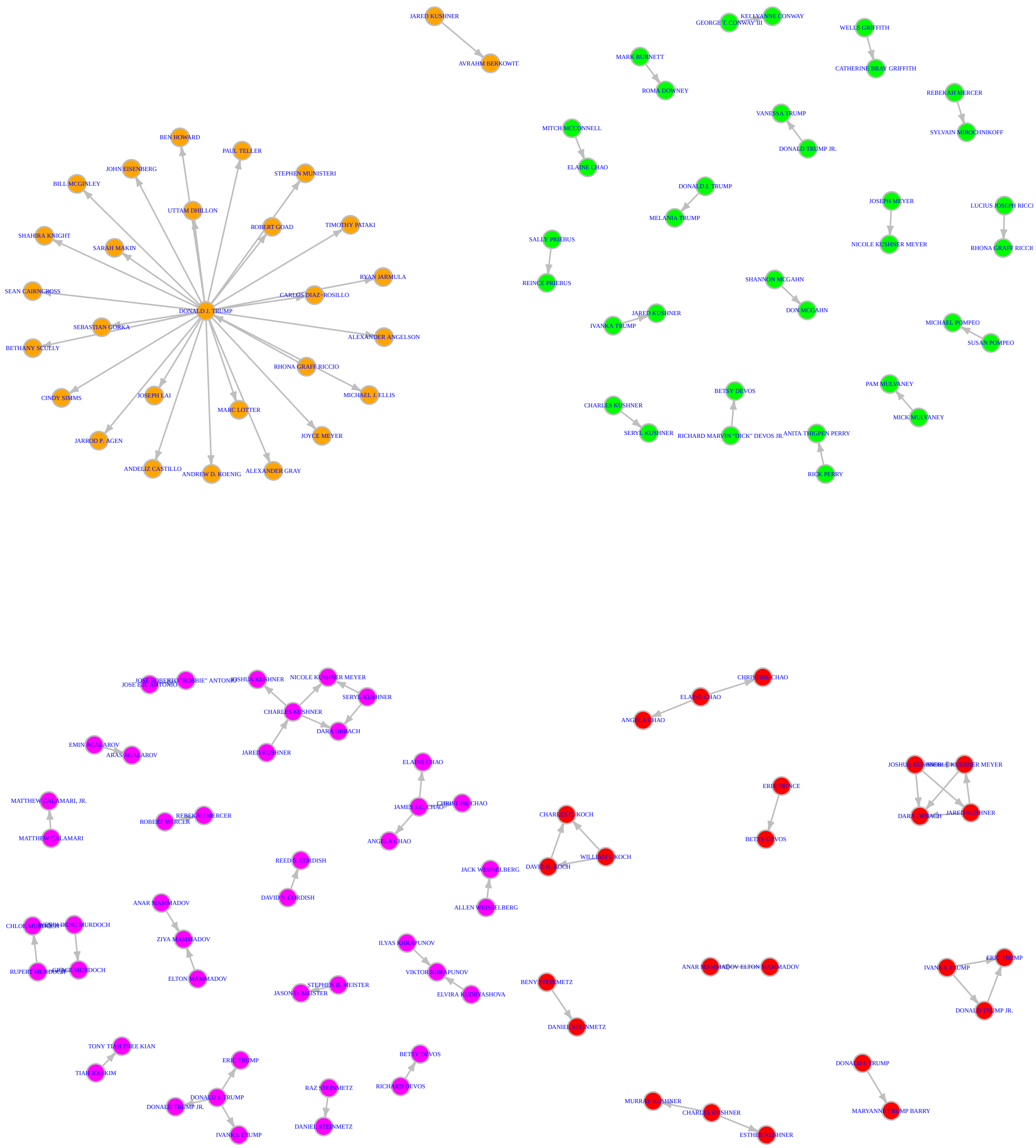
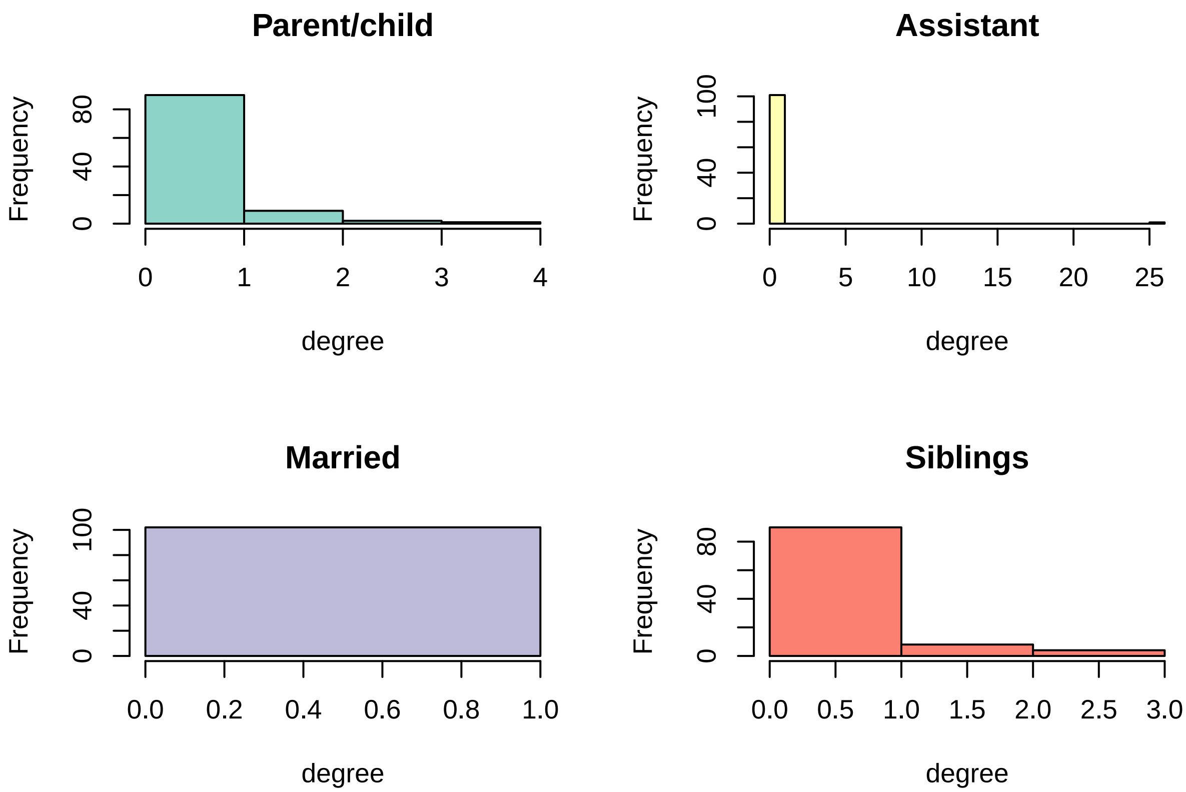


Figure 4: c2c Separate Visualisation

```
    )
}
```




```r
kable(t(table(degree(layers[[1]]))), caption = "Flatten: Degree distribution")
```

Table 20: Flatten: Degree distribution

1	2	3	4	5	6	7	31
73	16	4	4	1	2	1	1

```r
kable(t(table(degree(layers[[2]]))), caption = "Assistant: Degree distribution")
```

Table 21: Assistant: Degree distribution

1	26
28	1

```r
kable(t(table(degree(layers[[3]]))), caption = "Married: Degree distribution")
```

Table 22: Married: Degree distribution

1
34

```
kable(t(table(degree(layers[[4]]))), caption = "Parent/child: Degree distribution")
```

Table 23: Parent/child: Degree distribution

1	2	3	4
32	9	2	1

```
kable(t(table(degree(layers[[5]]))), caption = "Siblings: Degree distribution")
```

Table 24: Siblings: Degree distribution

1	2	3
12	8	4