eda

Harsh Vardhan Pachisia

2024-05-20

Selecting villages

```r
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(dplyr)
```

```r
load("data/DS0012/36151-0012-Data.rda")
villages <- da36151.0012
rm(da36151.0012)
```

```r
#selecting the columns we need
villages_sel_cols <- villages %>%
  #select(STATEID,DISTID,STATE, DISTA, VILL, starts_with("VI16")) %>%
   mutate(
    # Concatenate the columns, convert to numeric if needed
    village_id = as.numeric(paste0(STATE, DISTA, VILL))

  ) %>%
  #keeping only distinct village ids
  distinct(village_id, .keep_all = TRUE)

length(unique(villages_sel_cols$village_id))
```

```
## [1] 1381
```

```
#converting to binary
villages_sel_cols <- villages_sel_cols %>%
 mutate_at(vars(starts_with("VI16")), ~ case_when(
    str_detect(., "1") ~ 1,    # Convert '(1) Yes 1' to 1
    str_detect(., "0") ~ 0,    # Convert '(0) No 0' to 0
    TRUE ~ NA_integer_         # Handle unexpected cases
  )) %>%
  # Add a new column to sum all institution indicators
  mutate(total_institutions = rowSums(select(., starts_with("VI16")), na.rm = TRUE))

#colnames(villages_sel_cols)
```

```
villages_15 <- villages_sel_cols %>%
  select(STATEID,DISTID,STATE, DISTA, VILL, village_id, total_institutions, starts_with("VI16"), starts_
         starts_with("VH2")) %>%
  filter(total_institutions == 15)

# not selecting any of the villages in Kerela (which have 16 institutions) since not representative of
 # select(STATEID,DISTID,STATE, DISTA, VILL, starts_with("VI16"), village_id, total_institutions)
```

```
# based on religious diversity (in terms of caste and religion) and number of institutions
# selecting the village in maharashtra
mah <- villages_15 %>%
  filter(village_id == 2774)
```

importing household data

```
load("data/DS0002/36151-0002-Data.rda")
households <- da36151.0002
rm(da36151.0002)
```

```
households_select <- households %>%
  filter(STATEID == '(27) Maharashtra 27' & DISTID == 7 & PSUID == 4)
  #filter(STATEID == '(06) Haryana 06' & DISTID == 17 & PSUID == 5)
  #select(STATEID,DISTID,STATE, DISTA, VILL, starts_with("VI16")) %>%
```

```
households_select <- households_select %>%
  select(IDHH, starts_with('SN'), starts_with('ME'), ID11, starts_with('ID12'), ID13, GROUPS, ID14, sta

households_select <- households_select %>%
  mutate_at(vars(starts_with("ME")), ~ case_when(
    str_detect(., "1") ~ 1,    # Convert '(1) Yes 1' to 1
    str_detect(., "0") ~ 0,    # Convert '(0) No 0' to 0
    TRUE ~ NA_integer_         # Handle unexpected cases
  ) ) %>%
   mutate_at(vars(starts_with("SN2")), ~ case_when(
    str_detect(., "1") ~ 1,    # Convert '(1) Yes 1' to 1
    str_detect(., "0") ~ 0,    # Convert '(0) No 0' to 0
    TRUE ~ NA_integer_         # Handle unexpected cases
  ) )
```

```
library(fastDummies)
households_select <- households_select %>%
  mutate(mother_tounge = gsub("\\(\\d+\\)\\s*|\\d+", "", SN1)) %>%
  mutate(religion = gsub("\\(\\d+\\)\\s*|\\d+", "", ID11)) %>%
  mutate(caste_name = gsub("\\(\\d+\\)\\s*|\\d+", "", ID12ANM)) %>%
  mutate(sub_caste_name = gsub("\\(\\d+\\)\\s*|\\d+", "", ID12BNM)) %>%
  mutate(caste_category = gsub("\\(\\d+\\)\\s*|\\d+", "", ID13))# Remove the numeric code and parenthes
```

Choosing the required columns to keep

```
households_select <-
  households_select %>%
  select(IDHH, SN2I1, SN2I2, SN2G1, SN2G2, SN2H1, SN2H2,SN2F1, SN2F2, ME14, ME14A,
         ME2,ME3, ME4, ME6, ME8, ME9, ME11, ME12, ME13, ID14, mother_tounge, religion, caste_name, sub_
```

```
#social network characteristics
households_select <-
  households_select %>%
  mutate(
     police_links = pmax(SN2I1, SN2I2),
     pol_links = pmax(SN2G1, SN2G2, SN2H1, SN2H2),
     bue_links = pmax(SN2F1, SN2F2)
  ) %>%
  mutate(ID = row_number()) %>%
  rename(
    youth_group = ME2,
      bus_group= ME3,
      self_help_group= ME4,
      religious_group=  ME6,
      caste_group= ME8 ,
      ngo_group= ME9 ,
      pol_group= ME11 ,
      rot_group=  ME12,
      pubmeet_group= ME13,
       income = ID14
  ) %>%
  select(-c(SN2I1, SN2I2,SN2G1, SN2G2, SN2H1, SN2H2,SN2F1, SN2F2,ME14, ME14A, IDHH))
```

```
# Load necessary libraries
library(dplyr)
library(tidyr)
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.2.3
```

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

```
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
```

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

```
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
## The following object is masked from 'package:tidyr':
## 
##     crossing

## The following object is masked from 'package:tibble':
## 
##     as_data_frame

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

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

```r
# Create an edge list with all households
edge_list <- households_select %>%
  select(ID, contains("_group")) %>%
  pivot_longer(cols = -ID, names_to = "institution", values_to = "membership") %>%
  filter(membership == 1) %>%
  select(ID, institution)

# Ensure all households are included as nodes
# Ensure that 'name' column is appropriately created
all_households <- data.frame(name = as.character(households_select$ID), type = "household")

# Ensure all institutions are included as nodes (even if no household is connected)
# Make sure the institution names are treated as characters
all_institutions <- data.frame(name = as.character(unique(edge_list$institution)), type = "institution"

# Combine all nodes
# Since 'name' and 'type' columns are already present, direct rbind should work
all_nodes <- rbind(all_households, all_institutions)

# Check if all columns align correctly
print(head(all_nodes))
```

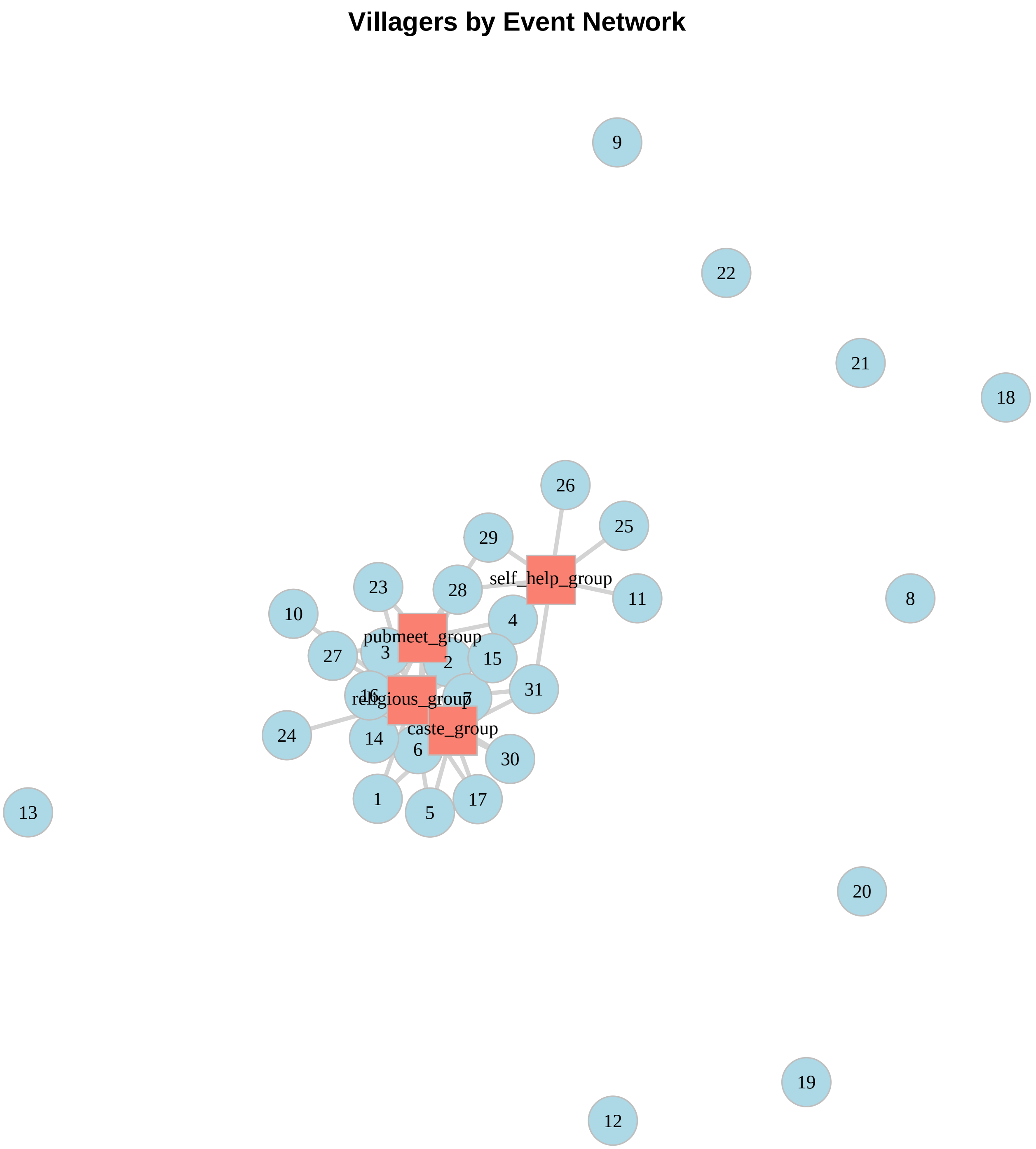
```
##   name      type
## 1    1 household
## 2    2 household
## 3    3 household
## 4    4 household
## 5    5 household
## 6    6 household
```

```r
library(igraph)
g <- graph_from_data_frame(d = edge_list,vertices = all_nodes, directed = FALSE)

V(g)$name <- as.character(V(g)$name)
V(g)$name
```

```
##  [1] "1"               "2"               "3"               "4"              
##  [5] "5"               "6"               "7"               "8"              
##  [9] "9"               "10"              "11"              "12"             
## [13] "13"              "14"              "15"              "16"             
## [17] "17"              "18"              "19"              "20"             
## [21] "21"              "22"              "23"              "24"
## [25] "25"              "26"              "27"              "28"
## [29] "29"              "30"              "31"              "religious_group"
## [33] "caste_group"     "pubmeet_group"   "self_help_group"
```

```r
# Prepare node attribute data
# Prepare node attribute data
node_attributes <- households_select %>%
  select(ID, income, mother_tounge, religion, caste_name, sub_caste_name, caste_category, police_links,
  mutate(ID = as.character(ID))  # Convert ID to character
V(g)$type <- ifelse(V(g)$name %in% node_attributes$ID, "household", "institution")
V(g)$income <- node_attributes$income[match(V(g)$name, node_attributes$ID)]
V(g)$mother_tounge <- node_attributes$mother_tounge[match(V(g)$name, node_attributes$ID)]
V(g)$religion <- node_attributes$religion[match(V(g)$name, node_attributes$ID)]
V(g)$caste_name <- node_attributes$caste_name[match(V(g)$name, node_attributes$ID)]
V(g)$sub_caste_name <- node_attributes$sub_caste_name[match(V(g)$name, node_attributes$ID)]
V(g)$caste_category <- node_attributes$caste_category[match(V(g)$name, node_attributes$ID)]
V(g)$police_links <- node_attributes$police_links[match(V(g)$name, node_attributes$ID)]
V(g)$pol_links <- node_attributes$pol_links[match(V(g)$name, node_attributes$ID)]
V(g)$bue_links <- node_attributes$bue_links[match(V(g)$name, node_attributes$ID)]
```

```r
# Set seed for reproducibility
set.seed(1)

# Assign vertex colors based on node type
V(g)$color <- ifelse(V(g)$type == "household", "lightblue", "salmon")

# Assign vertex shapes based on node type
V(g)$shape <- ifelse(V(g)$type == "household", "circle", "square")


# Assign edge color
E(g)$color <- "lightgray"

# Set vertex label options
V(g)$label.color <- "black"
V(g)$label.cex <- 1
V(g)$frame.color <- "gray"
V(g)$size <- 18

# Plot the graph with specified layout and options
plot(g, vertex.size = 10,
     vertex.label.cex = 0.8,
     edge.width = 3,
     layout = layout_with_graphopt,  # Ensure this layout function is available or use another
     main = 'Villagers by Event Network')
```



#analysis

descriptive statistics

```
#manually creating bipartite_matrix since function was giving an error
household_nodes <- all_nodes$name[all_nodes$type == "household"]
institution_nodes <- all_nodes$name[all_nodes$type == "institution"]

bipartite_matrix <- matrix(0, nrow = length(household_nodes), ncol = length(institution_nodes),
                           dimnames = list(household_nodes, institution_nodes))

for (i in seq_len(nrow(edge_list))) {
    row <- which(household_nodes == edge_list$ID[i])
    col <- which(institution_nodes == edge_list$institution[i])
    if (length(row) > 0 && length(col) > 0) {  # Check if both indices are found
        bipartite_matrix[row, col] <- 1
    }
}
```

```
bipartite_matrix
```

```
##    religious_group caste_group pubmeet_group self_help_group
## 1                1           1             0               0
## 2                1           1             1               0
## 3                1           1             1               0
## 4                0           1             1               1
## 5                1           1             0               0
## 6                1           1             1               0
## 7                1           1             1               0
## 8                0           0             0               0
## 9                0           0             0               0
## 10               1           0             0               0
## 11               0           0             0               1
## 12               0           0             0               0
## 13               0           0             0               0
## 14               1           1             1               0
## 15               1           1             1               1
## 16               1           1             1               0
## 17               1           1             0               0
## 18               0           0             0               0
## 19               0           0             0               0
## 20               0           0             0               0
## 21               0           0             0               0
## 22               0           0             0               0
## 23               1           0             1               0
## 24               1           0             0               0
## 25               0           0             0               1
## 26               0           0             0               1
## 27               1           0             1               0
## 28               1           0             1               1
## 29               0           0             1               1
## 30               1           1             0               0
## 31               1           1             0               1
```

```
t(bipartite_matrix)
```

```
##                 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
## religious_group 1 1 1 0 1 1 1 0 0  1  0  0  0  1  1  1  1  0  0  0  0  0  1  1
## caste_group     1 1 1 1 1 1 1 0 0  0  0  0  0  1  1  1  1  0  0  0  0  0  0  0
## pubmeet_group   0 1 1 1 0 1 1 0 0  0  0  0  0  1  1  1  0  0  0  0  0  0  1  0
## self_help_group 0 0 0 1 0 0 0 0 0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0
##                 25 26 27 28 29 30 31
## religious_group  0  0  1  1  0  1  1
## caste_group      0  0  0  0  0  1  1
## pubmeet_group    0  0  1  1  1  0  0
## self_help_group  1  1  0  1  1  0  1
```

household-by-household matrix

```
household_matrix_prod <- bipartite_matrix %*% t(bipartite_matrix)
diag(household_matrix_prod) <- 0 #to avoid self loops
household_matrix_prod
```

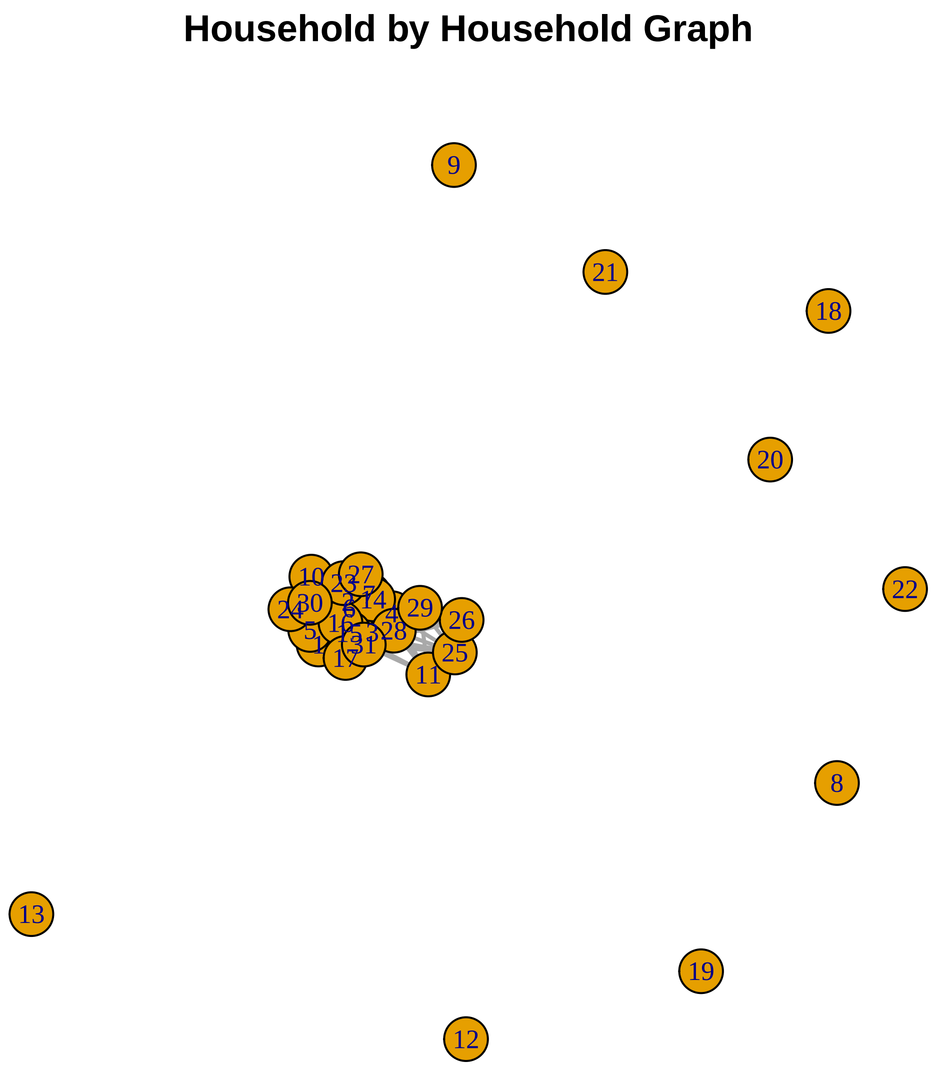
```
##    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
## 1  0 2 2 1 2 2 2 0 0  1  0  0  0  2  2  2  2  0  0  0  0  0  1  1  0  0  1  1
## 2  2 0 3 2 2 3 3 0 0  1  0  0  0  3  3  3  2  0  0  0  0  0  2  1  0  0  2  2
## 3  2 3 0 2 2 3 3 0 0  1  0  0  0  3  3  3  2  0  0  0  0  0  2  1  0  0  2  2
## 4  1 2 2 0 1 2 2 0 0  0  1  0  0  2  3  2  1  0  0  0  0  0  1  0  1  1  1  2
## 5  2 2 2 1 0 2 2 0 0  1  0  0  0  2  2  2  2  0  0  0  0  0  1  1  0  0  1  1
## 6  2 3 3 2 2 0 3 0 0  1  0  0  0  3  3  3  2  0  0  0  0  0  2  1  0  0  2  2
## 7  2 3 3 2 2 3 0 0 0  1  0  0  0  3  3  3  2  0  0  0  0  0  2  1  0  0  2  2
## 8  0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 9  0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 10 1 1 1 0 1 1 1 0 0  0  0  0  0  1  1  1  1  0  0  0  0  0  1  1  0  0  1  1
## 11 0 0 0 1 0 0 0 0 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  1  0  1
## 12 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 13 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 14 2 3 3 2 2 3 3 0 0  1  0  0  0  0  3  3  2  0  0  0  0  0  2  1  0  0  2  2
## 15 2 3 3 3 2 3 3 0 0  1  1  0  0  3  0  3  2  0  0  0  0  0  2  1  1  1  2  3
## 16 2 3 3 2 2 3 3 0 0  1  0  0  0  3  3  0  2  0  0  0  0  0  2  1  0  0  2  2
## 17 2 2 2 1 2 2 2 0 0  1  0  0  0  2  2  2  0  0  0  0  0  0  1  1  0  0  1  1
## 18 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 19 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 20 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 21 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 22 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 23 1 2 2 1 1 2 2 0 0  1  0  0  0  2  2  2  1  0  0  0  0  0  0  1  0  0  2  2
## 24 1 1 1 0 1 1 1 0 0  1  0  0  0  1  1  1  1  0  0  0  0  0  1  0  0  0  1  1
## 25 0 0 0 1 0 0 0 0 0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  1
## 26 0 0 0 1 0 0 0 0 0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  1
## 27 1 2 2 1 1 2 2 0 0  1  0  0  0  2  2  2  1  0  0  0  0  0  2  1  0  0  0  2
## 28 1 2 2 2 1 2 2 0 0  1  1  0  0  2  3  2  1  0  0  0  0  0  2  1  1  1  2  0
## 29 0 1 1 2 0 1 1 0 0  0  1  0  0  1  2  1  0  0  0  0  0  0  1  0  1  1  1  2
## 30 2 2 2 1 2 2 2 0 0  1  0  0  0  2  2  2  2  0  0  0  0  0  1  1  0  0  1  1
## 31 2 2 2 2 2 2 2 0 0  1  1  0  0  2  3  2  2  0  0  0  0  0  1  1  1  1  1  2
##    29 30 31
## 1   0  2  2
## 2   1  2  2
## 3   1  2  2
## 4   2  1  2
## 5   0  2  2
## 6   1  2  2
## 7   1  2  2
## 8   0  0  0
## 9   0  0  0
## 10  0  1  1
## 11  1  0  1
## 12  0  0  0
## 13  0  0  0
## 14   1   2   2
## 15   2   2   3
## 16   1   2   2
## 17   0   2   2
## 18   0   0   0
## 19   0   0   0
## 20   0   0   0
## 21   0   0   0
## 22   0   0   0
## 23   1   1   1
## 24   0   1   1
## 25   1   0   1
## 26   1   0   1
## 27   1   1   1
## 28   2   1   2
## 29   0   0   1
## 30   0   0   2
## 31   1   2   0
```

Institution by Institution matrix

```
group_matrix_prod <- t(bipartite_matrix) %*% bipartite_matrix
diag(group_matrix_prod) <- 0 #to avoid self-loops
group_matrix_prod
```

```
##                   religious_group caste_group pubmeet_group self_help_group
## religious_group                 0          12            10               3
## caste_group                    12           0             8               3
## pubmeet_group                  10           8             0               4
## self_help_group                 3           3             4               0
```

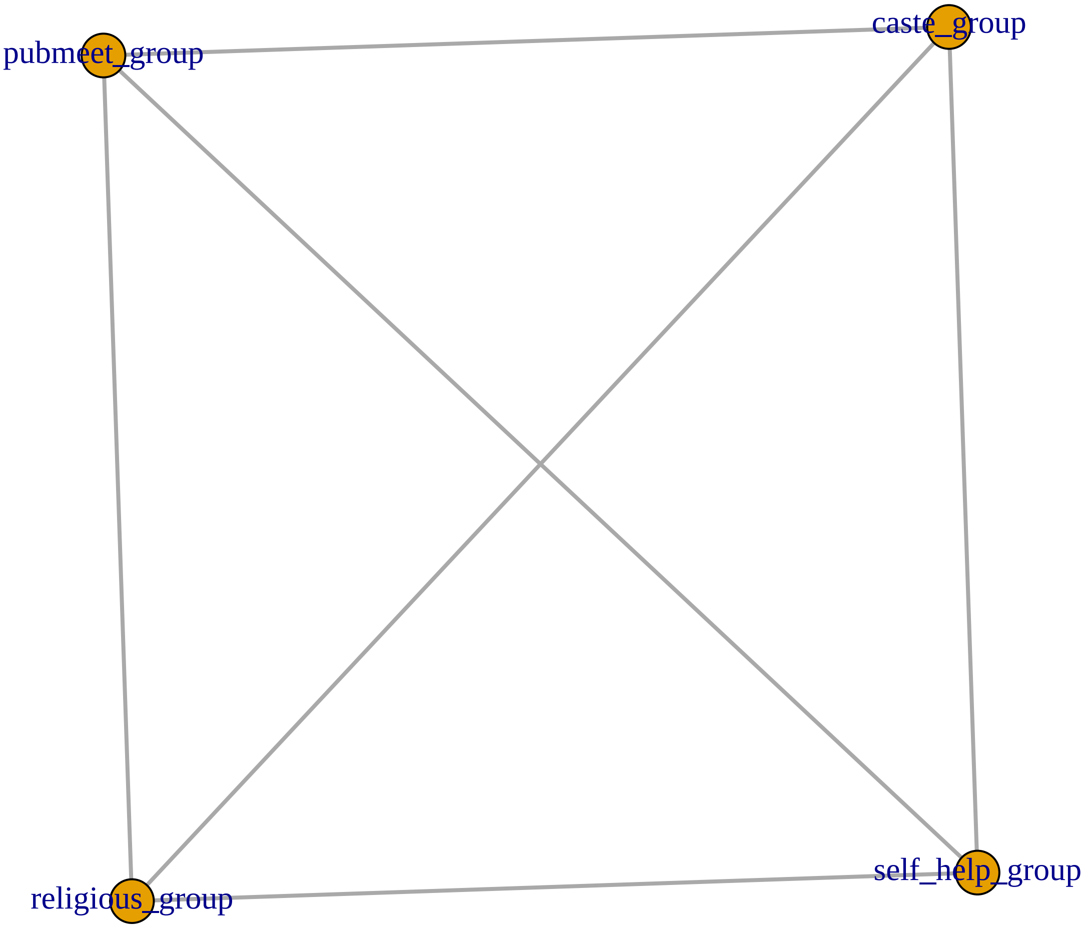
```
set.seed(1)
household_matrix_graph <- graph_from_adjacency_matrix(
  household_matrix_prod, mode = "undirected", weighted = FALSE)
plot(household_matrix_graph,
     vertex.size = 10,
     vertex.label.cex = 0.8,
     edge.width = 2,
     layout = layout_with_graphopt,
     main='Household by Household Graph')
```



Institution by Institution matrix

```
set.seed(1)
group_matrix_graph <- graph_from_adjacency_matrix(
  group_matrix_prod, mode = "undirected", weighted = TRUE)
plot(
  group_matrix_graph,
  vertex.size = 10,
  vertex.label.cex = 1,
  edge.width = 2,
  layout = layout_with_graphopt,
  main='Group by Group Graph')
```

Group by Group Graph



```
deg <- igraph::degree(g)
types <- V(g)$type
bet <- igraph::betweenness(g)

cent_df <- data.frame(types, deg,bet)
cent_df[order(
  cent_df$type, decreasing = TRUE),]
```

```
##                       types deg         bet
## religious_group institution  17 127.8604116
## caste_group     institution  13  57.7814141
## pubmeet_group   institution  12  56.7208864
## self_help_group institution   8  76.6372879
## 1                 household   2   0.5134783
## 2                 household   3   2.3923217
## 3                 household   3   2.3923217
## 4                 household   3  12.0830965
## 5                 household   2   0.5134783
## 6                 household   3   2.3923217
## 7                 household   3   2.3923217
## 8                 household   0   0.0000000
## 9                 household   0   0.0000000
## 10                household   1   0.0000000
## 11                household   1   0.0000000
## 12                household   0   0.0000000
## 13                household   0   0.0000000
## 14                household   3   2.3923217
## 15                household   4  24.4086471
## 16                household   3   2.3923217
## 17                household   2   0.5134783
## 18                household   0   0.0000000
## 19                household   0   0.0000000
## 20                household   0   0.0000000
## 21                household   0   0.0000000
## 22                household   0   0.0000000
## 23                household   2   1.0851127
## 24                household   1   0.0000000
## 25                household   1   0.0000000
## 26                household   1   0.0000000
## 27                household   2   1.0851127
## 28                household   3  16.4003841
## 29                household   2   4.5883117
## 30                household   2   0.5134783
## 31                household   3  17.9414920
```

```
centrality_households <- igraph::degree(household_matrix_graph)
centrality_households
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 28 39 39 30 28 39 39  0  0 16  7  0  0 39 46 39 28  0  0  0  0  0 27 16  7  7
## 27 28 29 30 31
## 27 34 18 28 35
```