Film_Markdown

Load libraries

Load function scripts and data

```
#load_functions
source("calculatecoocstats.R") #calculate co-occurrence statistics
source("grapher.R") #create graph
#Wiedemann, Gregor; Niekler, Andreas (2017): Hands-on: A five day text mining course for humanists and
source("rawcounts.R") #find raw counts of co-occurrences
source("token_filter.R") #filter tokens
```

Load token data

```
#load tokens, get it ready for analysis
load("token.all.RData")
#convert tokens to all lower
token.all <- tokens_tolower(token.all) #convert all tokens to lower
#sample based on min in a decade
token.all = tokens_sample(token.all, size = 22638, replace = FALSE, prob = NULL, by = decade)
```

Find Probability of Verbs/Adj/Noun given male/female across decades

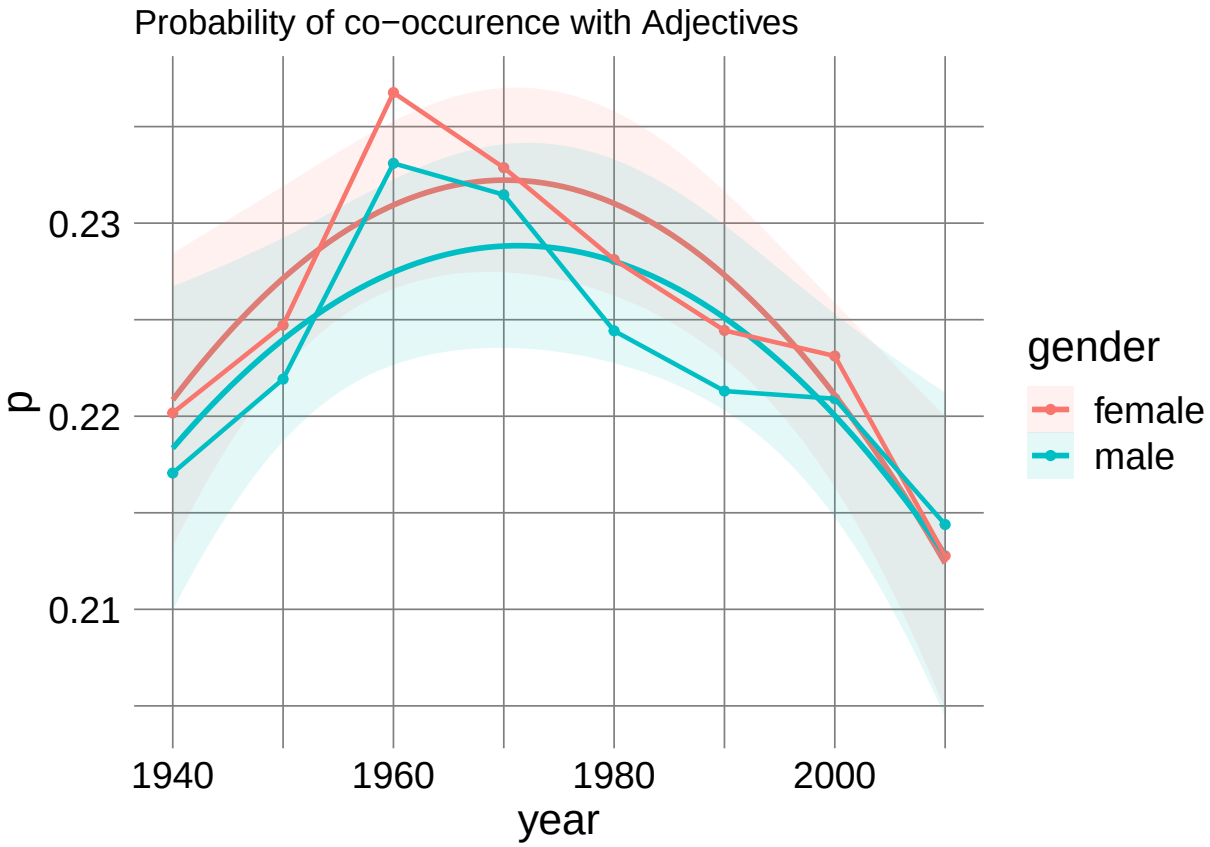
```
#create a token set with only generalized pos info
pos_replace <- function(toks.replace){
  toks.replace <- toks.replace %>%
    tokens_replace(pattern = c("*/NOUN", "*/VERB", "*/ADJ"), replacement = c("NOUN", "VERB", "ADJ"))
  return(toks.replace)
}
token.pos <- pos_replace(token.all)

p_decdat <- data.frame() #initialize data frame
pos = c('verb', 'adj', 'noun') #pos to be analysed

for(j in 0:7){ #for loop to run for each decade
  year = 1940 + j*10 #create decade variable
  pos_counts <- rawcounts(token_filter("all", year, token.pos)) #find raw co-occurrence counts
  male_pos <- pos_counts["male/characters", pos] #filter pos
  male_p <- male_pos / sum(male_pos) #find empirical probability
  male_pdat <- data.frame(pos = names(male_pos), p = male_p) #organise data frame
  male_pdat$gender = "male" #assign gender
```

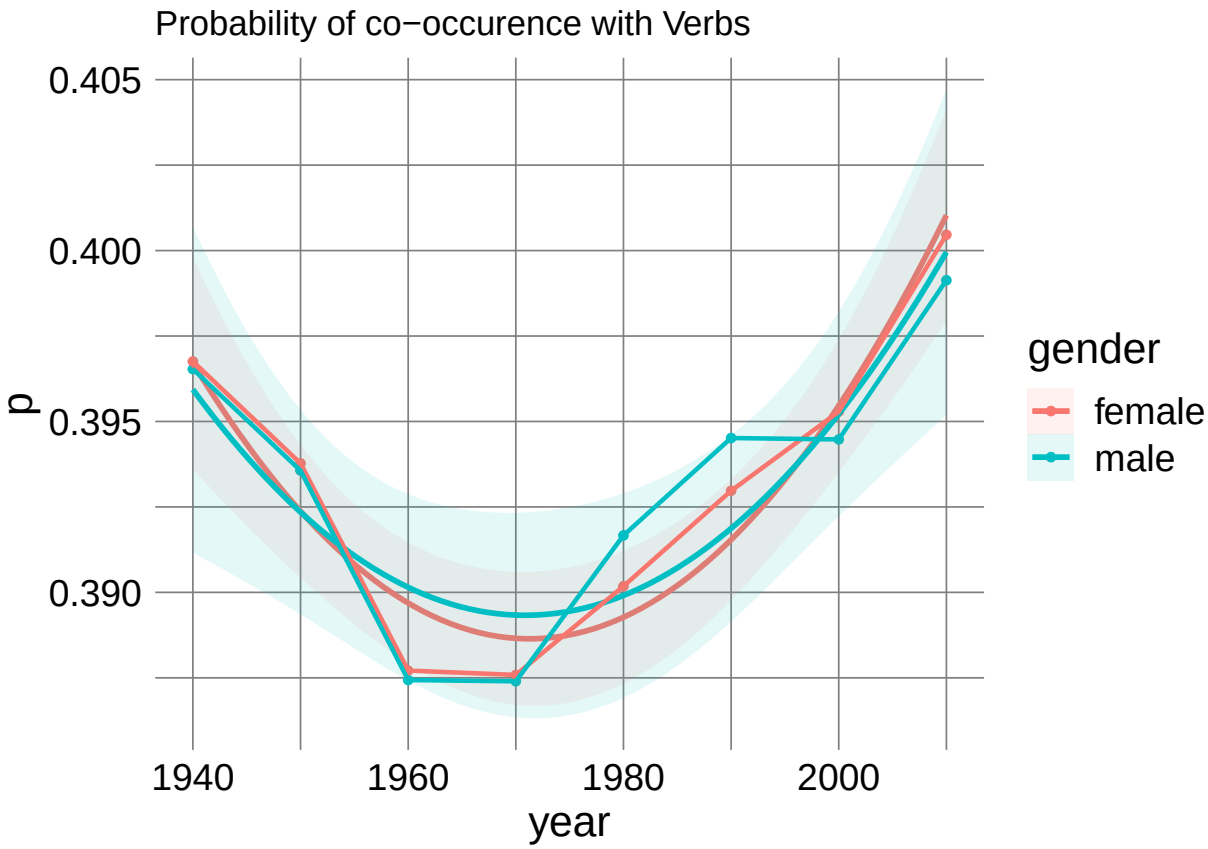
```r
  #do the same for females
  female_pos <- pos_counts["female/characters", pos]
  female_p <- female_pos / sum(female_pos)
  female_pdat <- data.frame(pos = names(female_pos), p = female_p)
  female_pdat$gender = "female"

  p_decdat.temp <- rbind(male_pdat, female_pdat) #bind gender data
  p_decdat.temp$year <- year #assign year
  p_decdat <- rbind(p_decdat, p_decdat.temp) #bind ind. decade with overall
}
```



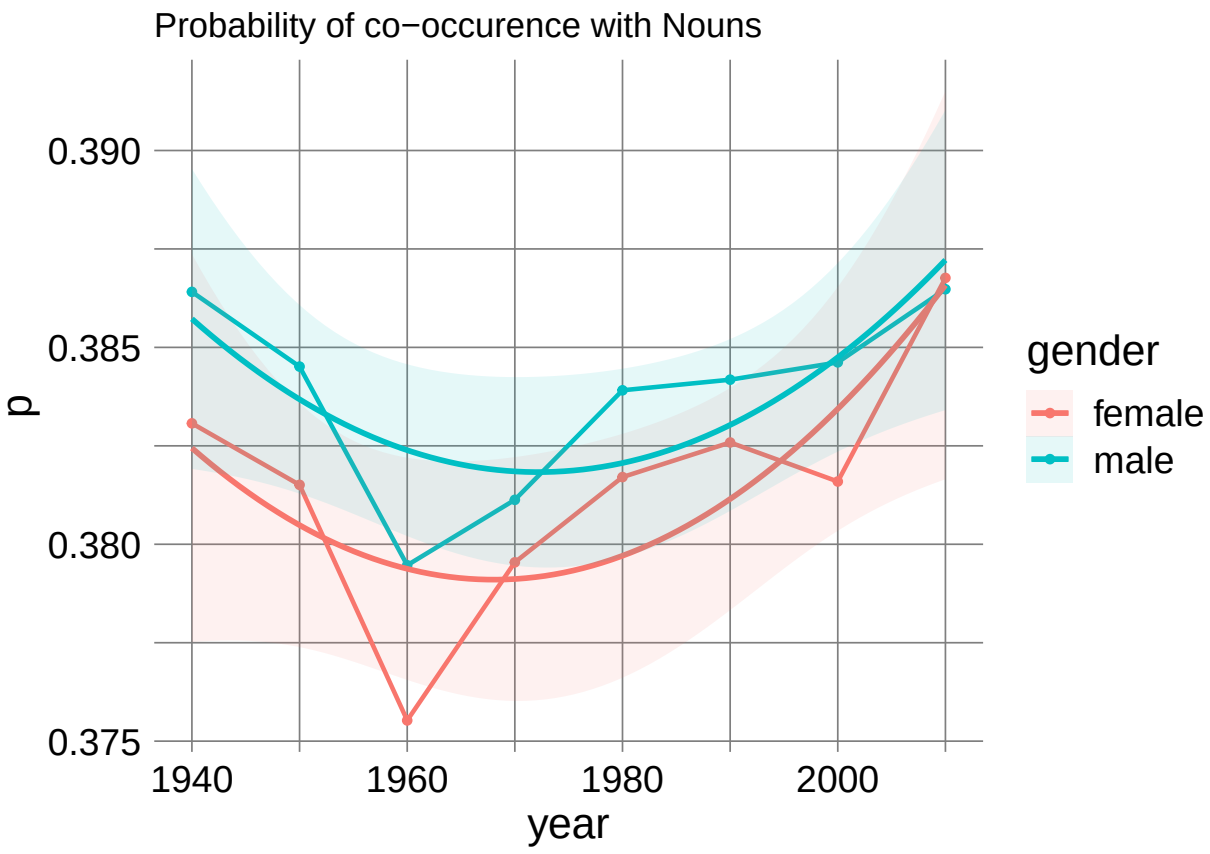
Adjectives

```
##             Df    Sum Sq   Mean Sq F value Pr(>F)
## year         1 0.0000837 8.367e-05   1.697  0.217
## gender       1 0.0000211 2.115e-05   0.429  0.525
## year:gender  1 0.0000037 3.710e-06   0.075  0.789
## Residuals   12 0.0005918 4.932e-05
```



Verbs

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## year          1 2.984e-05 2.984e-05   1.558  0.236
## gender        1 0.000e+00 0.000e+00   0.000  0.999
## year:gender   1 4.000e-08 3.900e-08   0.002  0.965
## Residuals    12 2.298e-04 1.915e-05
```



Nouns

```
## Saving 6.5 x 4.5 in image
```

```
##             Df    Sum Sq   Mean Sq F value Pr(>F)
## year         1 1.358e-05 1.358e-05   1.705  0.216
## gender       1 2.118e-05 2.118e-05   2.659  0.129
## year:gender  1 2.990e-06 2.989e-06   0.375  0.552
## Residuals   12 9.555e-05 7.963e-06
```

Community analysis

1940 - 2020

```
load("token.all.RData")
#convert tokens to all lower
token.all <- tokens_tolower(token.all) #convert all tokens to lower
token.all = tokens_sample(token.all, size = 22638, replace = FALSE, prob = NULL, by = decade)
token.all <- token_filter2('all', 2000, 2010, token.all)
```

Load and prepare data

```r
detect_communities <- function(toks.all, gender = 'male', nn = 10){
  toks <- toks.all %>%
     tokens_select(pattern = paste(gender, '/characters', sep = ''), selection = 'remove', padding = TRU

  #filter to keep only words that occur at least 10 times
dfm <-  toks %>% dfm() %>% dfm_trim(min_termfreq = 10)
filtered = colnames(dfm)
toks <- token.all %>%
  tokens_select(pattern = filtered, selection = 'keep', padding = TRUE)

#feature co-occurrence matrix for males
fcmat = fcm(toks, context = c("window"),
                 count = c("weighted"), #words are weighted within the window
                 window = 5)

graph = graph_from_adjacency_matrix(fcmat, weighted = TRUE) #create graph from matrix
edgelist <- get.data.frame(graph)
edgelist_m <- as.matrix(edgelist[ ,c("from", "to")])

graph <- graph_from_edgelist(edgelist_m, directed = FALSE)
graph <- set.edge.attribute(graph, "weight", value = edgelist$weight)
graph = simplify(graph, remove.loops = TRUE) #remove self-looping edges

#louvian communities
louvain <- cluster_louvain(graph, weights = E(graph)$weights)#detect communities
graph$community <- louvain$membership

#most important word in each community
communities <- data.frame()

for (i in unique(graph$community)) {
  # create subgraphs for each community
  subgraph <- induced_subgraph(graph, v = which(graph$community == i))
  # get size of each subgraph
  size <- igraph::gorder(subgraph)
  # get betweenness centrality
  btwn <-  igraph::betweenness(subgraph)
  communities <- communities %>%
    dplyr::bind_rows(
      data.frame(community = i,
                 n_characters = size,
                 most_important = names(which(btwn == max(btwn)))
      )
    )
}

communities = arrange(communities, desc(n_characters))
top_comm <- communities$community[1:5]
print(communities)

#top ten in each community
top_ten <- data.frame()
```

```
n = 0
for (i in top_comm) {
  # create subgraphs for each community
  subgraph <- induced_subgraph(graph, v = which(graph$community == i))
    n = n + 1
    # get degree
    degree <-  igraph::degree(subgraph)
    # get top ten degrees
    top <- names(head(sort(degree, decreasing = TRUE), nn))
    result <- data.frame(community = i, rank = 1:nn, word = top)
    top_ten <- top_ten %>%
    dplyr::bind_rows(result)
}

print(top_ten)
#write.csv(top_ten, paste(gender, '.csv', sep = ''))
print(paste('modularity =', modularity(louvain)))

#Visualizing the communities
subgraph <- induced_subgraph(graph, v = top_ten$word)
subgraph <- simplify(subgraph)
subgraph$community
nodes = data.frame(word = names(V(subgraph)))
group = rep(1:n, each = nn)
top_ten$group = group
clusters = inner_join(nodes, top_ten)
subgraph$community <- clusters$group
#unique(subgraph$community)

# give our nodes some properties, incl scaling them by degree and coloring them by community
V(subgraph)$size <- 5
V(subgraph)$frame.color <- "white"
V(subgraph)$color <- subgraph$community
#V(male_subgraph)$label <- V(male_subgraph)$name
V(subgraph)$label.cex <- 1.8

# also color edges according to their starting node
#edge.start <- ends(subgraph, es = E(subgraph), names = F)[,1]
#E(subgraph)$color <- V(subgraph)$color[edge.start]
#E(subgraph)$arrow.mode <- 0

#plot by groups
#make clusters first
clust_obj = make_clusters(subgraph, membership = clusters$group)

# weights <- ifelse(crossing(male_clust, male_subgraph), 1, 100)
# layout <- layout_with_kk(male_subgraph, weights=weights)
# plot(male_subgraph, layout=layout)

prettyColors <- c("turquoise4", "azure4", "olivedrab","deeppink4", "blue")
communityColors <- prettyColors[membership(clust_obj)]

edge.weights <- function(community, network, weight.within = 100, weight.between = 1) {
```

```
  bridges <- crossing(communities = community, graph = network)
  weights <- ifelse(test = bridges, yes = weight.between, no = weight.within)
  return(weights)
}
E(subgraph)$weight <- edge.weights(clust_obj, subgraph)
layout <- layout_with_fr(subgraph, weights=E(subgraph)$weight)
plot(subgraph, layout=layout, col = communityColors)
}
```

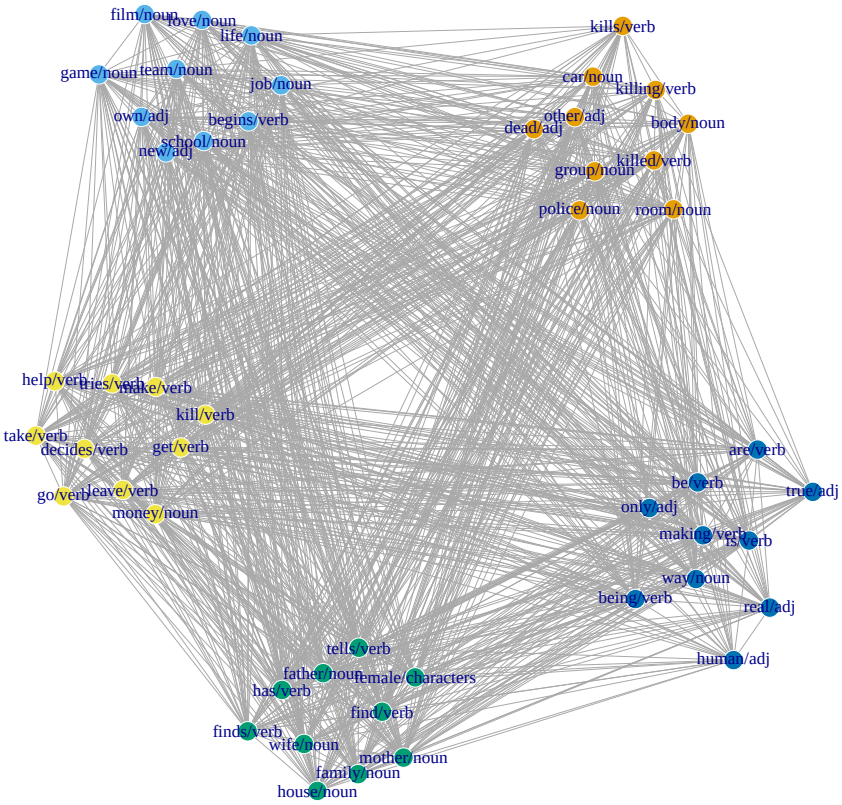
Function to detect and plor community structure

```
detect_communities(token.all, 'male', 10)
```

Male communities

```
##   community n_characters most_important
## 1         1          496   killing/verb
## 2         2          396        new/adj
## 3         9          368   becomes/verb
## 4         5          348         go/verb
## 5         4           99       only/adj
## 6         6           68    having/verb
## 7         3           17     voice/noun
## 8         7            5   suicide/noun
## 9         8            3         co/noun
##    community rank              word
## 1          1    1         other/adj
## 2          1    2          car/noun
## 3          1    3       killed/verb
## 4          1    4       police/noun
## 5          1    5        group/noun
## 6          1    6        kills/verb
## 7          1    7      killing/verb
## 8          1    8         room/noun
## 9          1    9         body/noun
## 10         1   10          dead/adj
## 11         2    1           new/adj
## 12         2    2         life/noun
## 13         2    3       school/noun
## 14         2    4         film/noun
## 15         2    5         team/noun
## 16         2    6       begins/verb
## 17         2    7           own/adj
## 18         2    8         game/noun
## 19         2    9         love/noun
## 20         2   10          job/noun
## 21         9    1 female/characters
## 22         9    2       father/noun
## 23         9    3       mother/noun
## 24         9    4        tells/verb
```

```
## 25         9    5       family/noun
## 26         9    6        finds/verb
## 27         9    7          has/verb
## 28         9    8        house/noun
## 29         9    9         find/verb
## 30         9   10         wife/noun
## 31         5    1          get/verb
## 32         5    2         take/verb
## 33         5    3           go/verb
## 34         5    4         make/verb
## 35         5    5         kill/verb
## 36         5    6        tries/verb
## 37         5    7        leave/verb
## 38         5    8        money/noun
## 39         5    9      decides/verb
## 40         5   10         help/verb
## 41         4    1           is/verb
## 42         4    2           be/verb
## 43         4    3        being/verb
## 44         4    4          are/verb
## 45         4    5          only/adj
## 46         4    6          way/noun
## 47         4    7       making/verb
## 48         4    8          real/adj
## 49         4    9          true/adj
## 50         4   10         human/adj
## [1] "modularity = 0.126887572528681"
```

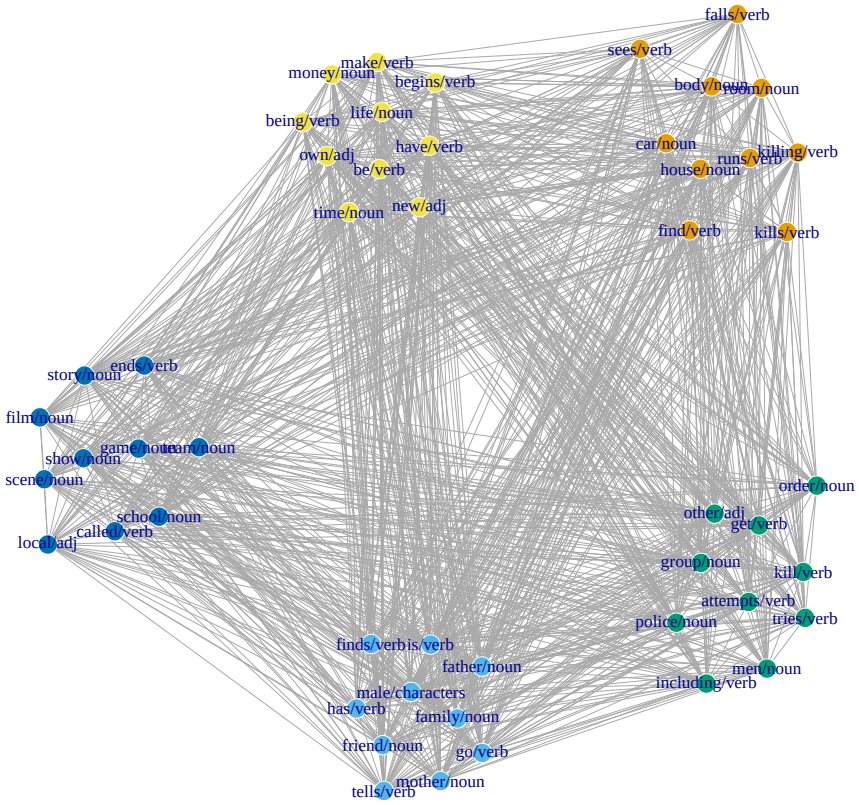



```
detect_communities(token.all, 'female', 10)
```

Female communities

```
##   community n_characters most_important
## 1         5          757   killing/verb
## 2         6          602     tells/verb
## 3         1          489      other/adj
## 4         2          446        new/adj
## 5         3          435    school/noun
## 6         4           22      gets/verb
##    community rank            word
## 1          5    1       find/verb
## 2          5    2      house/noun
## 3          5    3        car/noun
## 4          5    4       room/noun
## 5          5    5    killing/verb
## 6          5    6      kills/verb
## 7          5    7       body/noun
```

```
## 8          5    8      falls/verb
## 9          5    9       sees/verb
## 10         5   10       runs/verb
## 11         6    1 male/characters
## 12         6    2         is/verb
## 13         6    3     father/noun
## 14         6    4      tells/verb
## 15         6    5     mother/noun
## 16         6    6     family/noun
## 17         6    7        has/verb
## 18         6    8      finds/verb
## 19         6    9     friend/noun
## 20         6   10         go/verb
## 21         1    1       other/adj
## 22         1    2       kill/verb
## 23         1    3        get/verb
## 24         1    4     police/noun
## 25         1    5        men/noun
## 26         1    6      tries/verb
## 27         1    7      group/noun
## 28         1    8      order/noun
## 29         1    9  including/verb
## 30         1   10   attempts/verb
## 31         2    1         be/verb
## 32         2    2       life/noun
## 33         2    3         new/adj
## 34         2    4       have/verb
## 35         2    5       time/noun
## 36         2    6       make/verb
## 37         2    7         own/adj
## 38         2    8     begins/verb
## 39         2    9      being/verb
## 40         2   10      money/noun
## 41         3    1       film/noun
## 42         3    2     school/noun
## 43         3    3       team/noun
## 44         3    4       show/noun
## 45         3    5       ends/verb
## 46         3    6       local/adj
## 47         3    7       game/noun
## 48         3    8      scene/noun
## 49         3    9     called/verb
## 50         3   10      story/noun
## [1] "modularity = 0.108202579138032"
```


falls/verb
sees/verb
make/verb
money/noun
begins/verb
body/noun
room/noun
life/noun
being/verb
car/noun
killing/verb
own/adj
have/verb
runs/verb
be/verb
house/noun
new/adj
time/noun
find/verb
kills/verb
ends/verb
story/noun
film/noun
game/noun
team/noun
show/adj
scene/noun
order/noun
other/adj
get/verb
school/noun
called/verb
local/adj
group/noun
kill/verb
attempts/verb
police/noun
tries/verb
finds/verb
is/verb
father/noun
men/noun
including/verb
male/characters
has/verb
family/noun
friend/noun
go/verb
mother/noun
tells/verb